

UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA MECCANICA

DIPARTIMENTO DI INNOVAZIONE MECCANICA E GESTIONALE

TESI DI LAUREA

**STUDIO E REALIZZAZIONE DI UN JOYSTICK RETROAZIONATO
IN CONTROLLO DI FORZA**

Relatore: Ch.mo Prof. Aldo Rossi

Laureando: Fabio Antoniazzi

ANNO ACCADEMICO 1999-2000

INDICE

SOMMARIO	1
INTRODUZIONE	3
CAP. 1: RICERCA ATTUALE	7
1.1 PERCHÉ LA RETROAZIONE	7
1.2 METODI DI RETROAZIONE.....	8
1.2.1 <i>Controllo di posizione</i>	8
1.2.1.1 Un metodo per aumentare il guadagno.....	10
1.2.2 <i>Controllo di velocità</i>	11
1.2.2.1 Retroazione di forza	11
1.2.2.2 Retroazione di rigidezza.....	12
1.3 UN SISTEMA PER MUOVERSI IN AMPI SPAZI DI LAVORO (NTRFC).....	12
CAP. 2: IL PROGETTO JOYFORCE	17
2.1 CONSIDERAZIONI INIZIALI.....	17
2.2 DUE TIPI DI CONTROLLO	19
2.2.1 <i>Il controllo di velocità</i>	20
2.2.2 <i>Il controllo di forza</i>	22
2.3 DEFINIZIONE DEL SISTEMA.....	24
2.3.1 <i>Progetto del joystick</i>	25
2.3.1.1 Scelta della configurazione e dei componenti.....	26
2.3.1.2 Meccanica del joystick	31
2.3.1.3 Scheda di interfaccia	33
2.3.2 <i>Il microcontrollore Atmel AT90S4434</i>	38
2.3.2.1 Lettura dell'encoder	39
2.3.2.2 Ingressi analogici.....	43
2.3.2.3 Comunicazione con il DAC	46
2.3.2.4 Implementazione del controllo del motore.....	46
2.3.2.5 Implementazione del comando tradizionale.....	56
2.4 LA SIMULAZIONE DELL'AMBIENTE ESTERNO.....	60
2.4.1 <i>La scheda di acquisizione National Instruments PCI-6024E</i>	63
2.4.1.1 Programmazione dell'input/output.....	64
2.4.2 <i>Routine di movimento</i>	65
2.4.2.1 Comando di velocità.....	67
2.4.2.2 Comando di forza	69
2.4.3 <i>Salvataggio dei dati</i>	70
CAP. 3: RISULTATI	73
3.1 INSEGUIMENTO DI TRAIETTORIA	74
3.2 VELOCITÀ DI ESECUZIONE DELLA MANOVRA	77
3.2.1 <i>Confronto con l'opzione di non reversibilità</i>	79
3.3 INSEGUIMENTO DI POSIZIONE.....	80
CAP. 4: SVILUPPI FUTURI	83
CONCLUSIONI	85
APPENDICE A: DISEGNI COSTRUTTIVI	87
APPENDICE B: CODICE ASSEMBLER PER AT90S4434	95

APPENDICE C: ROUTINE DI SIMULAZIONE VISUAL BASIC	111
APPENDICE D: COSTO DEI COMPONENTI	117
BIBLIOGRAFIA	121

SOMMARIO

È stato affrontato il problema del comando di organi meccanici con la necessità di avere una elevata sensibilità dell'operazione, in particolare quando i volumi dello spazio di lavoro tra il comando e l'end-effector differiscono grandemente. Dapprima sono stati analizzati i sistemi esistenti e, considerando i limiti intrinseci, è stato definito un nuovo metodo di controllo, in cui il comando avviene misurando la forza esercitata dall'operatore e la retroazione avviene sulla posizione dell'end-effector.

Le configurazioni possibili del modello ideato e la sua realizzabilità sono state studiate con *Simulink* di *Matlab*.

Per verificarne l'efficacia è stato progettato e realizzato il prototipo di un joystick ad un solo asse, pensato per l'eventuale impiego nel comando di cilindri idraulici, per esempio in una macchina escavatrice.

I componenti principali sono un torsionmetro per la rilevazione della forza esercitata dall'operatore, e un azionamento costituito da un motoriduttore in corrente continua accoppiato ad un encoder per la lettura della posizione. Il controllo è effettuato da un microcontrollore ad 8 bit. Le caratteristiche principali sono una forza massima nominale sulla leva di 20 N ed una corsa utile di 30 cm su un arco di $\pm 60^\circ$.

L'ambiente esterno e l'interazione dell'attuatore con esso sono stati simulati tramite un PC dotato di una scheda di acquisizione, programmato in linguaggio *Visual Basic 6.0*.

Sono stati implementati due diversi tipi di controllo e di attuatori, uno in controllo di velocità, tipico sei sistemi idraulici, e l'altro in controllo di forza, come nel caso degli azionamenti elettrici.

Le caratteristiche dell'hardware a disposizione hanno posto il problema di adattare il modello di funzionamento ad un sistema con ritardo, fornendo, di conseguenza, lo spunto per un eventuale utilizzo nei sistemi di teleoperazione.

INTRODUZIONE

Il joystick retroazionato che è stato sviluppato potrebbe trovare immediata applicazione nel controllo di macchine ad azionamento oleodinamico per impieghi pesanti, utilizzate in ambito costruttivo, nelle opere di escavazione e nella silvicoltura, con l'obiettivo di aumentare la sensibilità dell'operatore.

Nel comando tradizionale, l'operatore manovra un joystick assimilabile ad una molla avente una predeterminata caratteristica elastica, nel quale una forza applicata corrisponde ad una posizione assunta. Questa posizione, attraverso sistemi diversi, rappresenta la velocità impartita all'attuatore idraulico. Un sistema così definito è detto in controllo di velocità ed è impiegato vantaggiosamente quando le velocità sono ridotte, perché consente di effettuare posizionamenti precisi anche quando la corsa di lavoro è molto estesa: un eventuale comando di posizione comporterebbe errori elevati per la presenza del rapporto esistente tra la corsa di lavoro e di comando, che è abitualmente almeno di un ordine di grandezza.

Inoltre, gli attuatori idraulici si prestano particolarmente ad un comando di velocità perché, l'eventuale valvola idraulica proporzionale o pompa a portata variabile di alimento, può variare la portata d'olio e non la pressione.

Per contro, l'operatore non ha nessun controllo sulla forza che la macchina sta esercitando, e l'unica stima che può fare deriva dalle vibrazioni e dal rumore degli organi sotto sforzo. L'utilizzo di queste macchine richiede, pertanto, una notevole abilità da parte dell'operatore, che può essere sviluppata soltanto con l'esperienza e un certo tempo di apprendimento.

In ogni caso la stima della forza applicata rimane una valutazione personale, e l'imprecisione intrinseca si riflette in manovre errate e sovraccarichi della macchina.

E' evidente che la possibilità dell'operatore di "sentire quello che sta facendo" avrebbe vantaggi sui tempi di apprendimento, sulla velocità e precisione di esecuzione, nonché sulla riduzione dei guasti causati dal sovraccarico.

Esistono poi delle situazioni in cui eseguire il compito prefissato risulta ancora più difficoltoso, perché non è nemmeno possibile vedere direttamente l'end-effector: è il caso della teleoperazione, ad esempio dei robot subacquei, preposti a lavorare in condizioni estreme, comandati dalla superficie.

Gli sforzi possono essere dedotti soltanto dalle immagini inviate da una telecamera, per cui ogni movimento richiede un aggravio di cautela e di tempo.

Stabilito il vantaggio di far pervenire all'operatore la consapevolezza tattile dell'operazione in corso, si rende necessario rilevare le grandezze che caratterizzano l'interazione dell'end-effector con l'ambiente. Questo si può attuare con un apparato sensoriale, tanto più complesso quanto maggiori sono i gradi di libertà e le grandezze che si vogliono rilevare nel sistema, e che nella presente trattazione, relativa ad un solo asse, deve essere in grado di misurare forza e posizione dell'attuatore.

Pensando all'applicazione su un cilindro idraulico, i trasduttori principali per misurare la forza applicata sono tre:

- attraverso una cella di carico fissata sullo stelo
- tramite un perno di fissaggio forato e dotato di estensimetri per rilevarne la deformazione
- con sensori di pressione applicati alle due camere del cilindro.

Gli ultimi due metodi sono applicabili senza particolari interventi su una macchina preesistente, con la differenza che un perno fornisce una misura più precisa, mentre i sensori di pressione sono più economici, ma necessitano di un'elaborazione che consideri gli attriti interni.

Per la posizione sono in commercio degli attuatori idraulici che prevedono già un sistema interno di rilevazione (di seguito, dal catalogo "PARKER Electrohydraulic Actuators").

Fondamentalmente si basano su due diversi principi.

Il primo, induttivo, non ha nessun contatto strisciante e può dare in uscita un segnale analogico con risoluzione infinita e una non-linearità di circa 0.02 %.

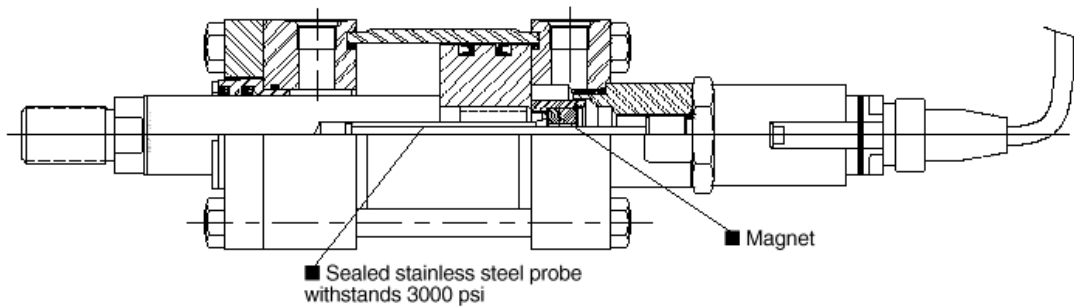


Figura 1: sensore di posizione induttivo

Il funzionamento è dato dall'induzione di un impulso elettromagnetico sullo stelo, e dalla misura del tempo che trascorre nell'aspettare il segnale di ritorno. Questo tempo è proporzionale alla distanza da percorrere e quindi all'estensione dello stelo. Con la tecnica sotto rappresentata è possibile ricavare un segnale PWM oppure, con un modulo aggiuntivo, un segnale analogico (in questo caso c'è la possibilità di avere anche il segnale di velocità).

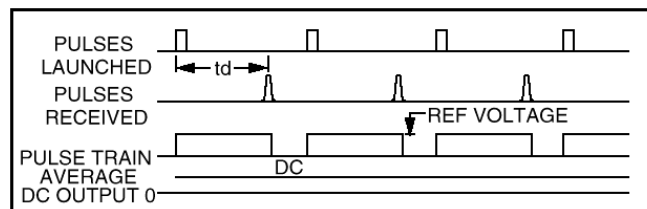


Figura 2: segnale di uscita

Il secondo, di tipo resistivo, ha invece una non linearità dello 0.1 % su un fondo scala di 1 m, ed ha una ripetibilità di 0.2 mm. Opera come un partitore di tensione con l'alimentazione che può essere compresa tra 5 e 50 Vdc.

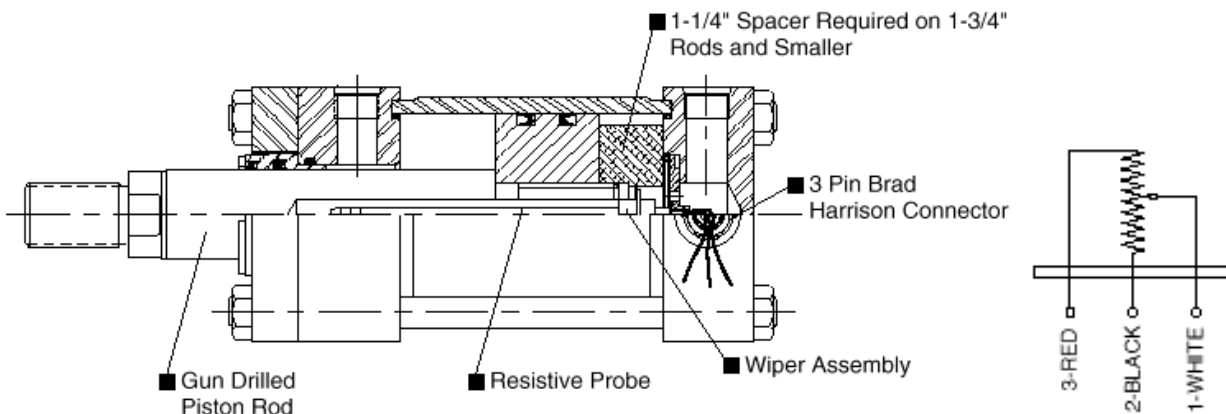


Figura 3: sensore di posizione resistivo

Una volta rilevate le misure di forza e posizione, rimane il problema di utilizzarle in modo da conferire al joystick il requisito di sensibilità desiderato; questo compito è proprio del sistema di controllo, a cui si dedica il primo capitolo con l'analisi delle soluzioni più comuni e interessanti.

Questo studio si è prefisso come obiettivo quello di indagare le possibili alternative nel gestire i segnali provenienti dai sensori, e alla luce di questo, sviluppare un nuovo sistema di controllo in grado di soddisfare sia la precisione nel movimento, che la capacità di riprodurre sul joystick le sollecitazioni dell'ambiente.

Anche se si è sempre tenuta particolare attenzione all'eventuale impiego su una macchina ad azionamento oleodinamico, si è voluto affrontare il problema indipendentemente da un'applicazione specifica.

Pertanto, proprio per questo requisito di generalità, fin dalle prime fasi il sistema è stato pensato per lavorare in molteplici modalità, e questo lo ha reso molto flessibile, capace di adattarsi ad ambiti diversi, e a cambiare logica di funzionamento con estrema facilità.

Cap. 1: RICERCA ATTUALE

Per quanto concerne l'ambito delle macchine ad azionamento oleodinamico per impieghi pesanti, allo scopo di aumentare la sensibilità dell'operatore, si può trovare un riferimento nella ricerca fatta da S.E. Salcudean [1] [2] [3] [4].

Più precisamente, Salcudean si è occupato di migliorare il comando di macchine adibite al taglio e movimentazione di tronchi albero, attraverso uno schema innovativo con retroazione di rigidità. Il risultato, applicato ad un CAT 215 per il caricamento di tronchi, è stato una riduzione del 10 % nel carico di lavoro.

Un campo di ricerca più generale, al quale si può ricondurre lo studio in esame, è invece il controllo di manipolatori. I problemi fondamentali che si presentano sono il controllo in campo libero e durante il contatto con l'ambiente. Il controllo di velocità presenta dei vantaggi considerevoli, primo tra tutti la linearità dell'algoritmo di soluzione, ma non è stato utilizzato nell'industria e nella teleoperazione per la sua inefficienza nel gestire il contatto con un ostacolo: le forze esercitate assumono, infatti, valori inaccettabili.

Le strade seguite per superare questo limite sono state diverse.

Craig e Raibert (1981) [7] hanno presentato un sistema ibrido in cui solo alcuni assi sono controllati in velocità mentre gli altri rimangono controllati in forza; il sistema non è così propriamente controllato in velocità perché bisogna imporre per ciascun asse sia la forza che la velocità.

Colbaugh (1993) [8] ha presentato uno schema adattivo per controllare l'impedenza dell'end-effector di un robot manipolatore nel contatto; in ogni caso è richiesto di effettuare un cambio di modalità per passare al movimento libero.

Hyde e Cutkosky (1994) [9] hanno indagato sperimentalmente alcuni metodi per controllare il passaggio tra il movimento libero ed il contatto.

Yao e Tomizuka (1995) [10] hanno presentato un sistema adattivo che manifestava un'incertezza sia nel movimento del robot che nell'applicazione della forza; Vukobratovic (1996) [11] si è occupato di considerarne la stabilizzazione nell'istante successivo al contatto.

Tarn (1996) [12] ha invece sviluppato una strategia basata su un evento per passare ad una delle due modalità di controllo.

Williams II (1998) [5] ha sviluppato un sistema in cui il passaggio avviene con continuità ed in modo proporzionale sia al comando dell'operatore, sia alla forza esercitata dall'end-effector; ha inoltre studiato le caratteristiche che devono avere il sistema fisico ed il controllo per soddisfare alla condizione di stabilità.

1.1 PERCHÉ LA RETROAZIONE

In un sistema tradizionale il controllo avviene a senso unico dalla parte di comando verso l'end-effector. Retroazione vuol dire, invece, dotare l'end-effector di sensori, che possono essere di forza, posizione, velocità ed accelerazione, in modo da renderlo un elemento attivo, capace di misurare l'interazione con l'ambiente esterno.

Questi segnali possono essere utilizzati per condizionarne direttamente il movimento, ed il sistema si dice adattivo perché è in grado di variare il comportamento in base agli stimoli esterni, oppure per pilotare l'organo di comando, come nel caso della retroazione di forza.

In quest'ultimo caso l'intento è di fornire all'operatore la sensibilità sull'operazione che sta svolgendo; sarà la sua reazione alla percezione degli stimoli esterni a stabilire un'eventuale modifica al comando.

Per contro il sistema acquista una maggiore complessità perché è lo stesso operatore ad entrare nell'anello di controllo, ed è quindi necessario introdurre il modello del corpo umano per prevederne la risposta alle sollecitazioni. È evidente che le caratteristiche si discostano in funzione della persona, e che tale modello non può essere univoco.

In ogni caso la presenza della retroazione modifica la funzione di trasferimento del sistema di controllo, che deve perciò essere rianalizzato per verificarne il funzionamento ed i possibili problemi di stabilità.

1.2 METODI DI RETROAZIONE

Si vuole ora descrivere brevemente l'effetto della retroazione sui sistemi più comuni di controllo, con particolare interesse alla sensazione fornita alla mano dell'operatore.

Innanzitutto si stabiliscono i modelli semplificati della mano (h), del joystick (j) e dell'ambiente esterno (E).

Per questo tipo di analisi si ritiene sufficiente rappresentare la mano come un sistema molla-smorzatore, in cui il comando viene impartito con la posizione (x_h), e la sensazione tattile viene percepita attraverso la forza (f_h).

Per il joystick si prende il modello classico costituito da massa-molla-smorzatore, al quale viene aggiunta la forza attuabile da un azionamento (f_j); i sistemi mano e joystick sono connessi direttamente.

L'ambiente esterno viene rappresentato genericamente con un sistema massa-molla-smorzatore, in cui la forza esercitata dall'attuatore (f_E) e la posizione (x_E) sono grandezze rilevate da sensori dedicati.

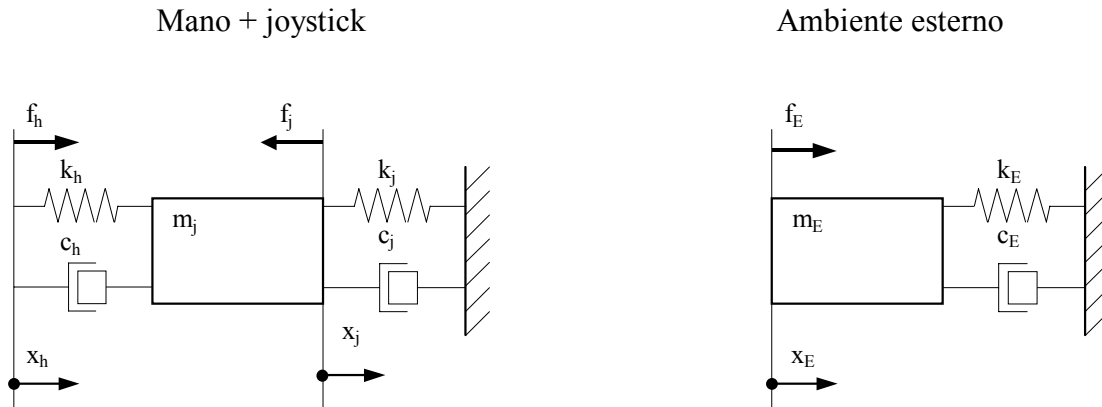


Figura 4: modellizzazione dei sistemi mano-joystick e ambiente esterno

La funzione di trasferimento dell'attuatore è approssimabile con un sistema del primo o secondo ordine con ritardo, ed è la maggiore causa dei problemi di stabilità. Successivamente verrà completamente trascurata, considerando il caso ideale in cui è unitaria, perché l'obiettivo è di analizzare il comportamento nei confronti della sensibilità; pertanto non si avranno informazioni inerenti all'effettiva stabilità del sistema.

1.2.1 Controllo di posizione

Rappresenta il modo più naturale per applicare la retroazione di forza: visto che l'attuatore segue la posizione impartita con il joystick, e la forza rilevata dall'attuatore viene trasmessa alla mano, l'operatore dovrebbe avere la sensazione di manovrare direttamente l'end-effector.

Questo schema (fig. 5) viene riportato solo a titolo di riferimento perché è possibile impiegarlo solo nel caso il joystick abbia a disposizione uno spazio di lavoro maggiore o uguale a quello dell'end-effector (interfacce aptiche); altrimenti, come nel nostro caso, la precisione del posizionamento viene ad essere pesantemente compromessa, perché l'errore sul comando viene moltiplicato per il fattore di scala k_{pos} .

Tenendo presente una scala diversa per il joystick e l'end-effector, le coordinate dei due sistemi sono legate dalla relazione:

$$x_E = k_{\text{pos}} \cdot x_j \quad (1)$$

Allo stesso modo è necessario un termine per riportare la forza sul joystick:

$$f_j = k_{\text{for}} \cdot f_E \quad (2)$$

In questa trattazione si esegue la semplificazione di considerare le funzioni di trasferimento del gruppo di controllo e dell'attuatore unitarie:

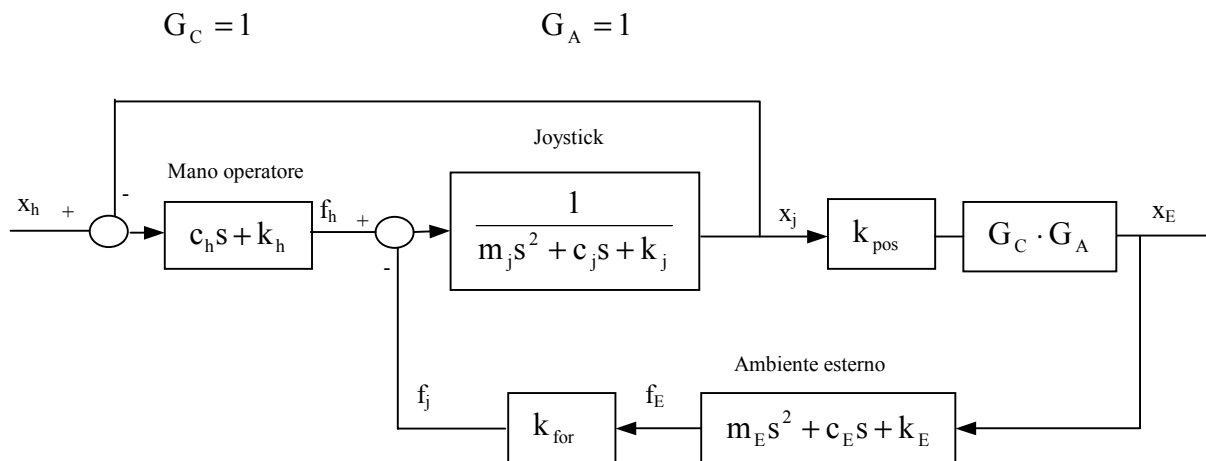


Figura 5: retroazione di forza nel controllo di posizione

Svolgendo la riduzione del sistema si trova la funzione di trasferimento che lega la posizione dell'end-effector (x_E) alla posizione impostata dalla mano (x_h):

$$\frac{\overline{x_E}}{\overline{x_h}} = \frac{k_{\text{pos}} \cdot (c_h \cdot s + k_h)}{(m_j + k_{\text{pos}} \cdot k_{\text{for}} \cdot m_E) \cdot s^2 + (c_h + c_j + k_{\text{pos}} \cdot k_{\text{for}} \cdot c_E) \cdot s + (k_h + k_c + k_{\text{pos}} \cdot k_{\text{for}} \cdot k_E)}$$

È l'equivalente di un altro sistema massa-molla-smorzatore in cui il termine $k_{\text{pos}} \cdot k_{\text{for}}$ determina l'avvicinamento al sistema ambiente. Anche se, con le semplificazioni assunte, il sistema risulta sempre stabile, si può notare come, aumentando il guadagno di retroazione k_{for} , i coefficienti di smorzamento del joystick e della mano perdano di efficacia, e nell'eventualità di contatto con superfici dure (k_E elevato) possano insorgere fenomeni di instabilità.

Per la rigidità tipica di un braccio di un robot industriale, è riportato in letteratura un guadagno di forza massimo di 1/10, senza perdere stabilità.

Si vuole ora determinare la condizione per la quale la forza percepita sulla mano corrisponde a quella, in scala, sull'end-effector, ovvero:

$$f_h = k_{for} \cdot f_E \quad (3)$$

Le due forze si esprimono con le seguenti equazioni differenziali:

$$f_h = m_j \cdot \ddot{x}_j + c_j \cdot \dot{x}_j + k_j \cdot x_j + f_j \quad (4)$$

$$f_E = m_E \cdot \ddot{x}_E + c_E \cdot \dot{x}_E + k_E \cdot x_E \quad (5)$$

Introducendo l'eq. (2) nella (4) si ha:

$$f_h = m_j \cdot \ddot{x}_j + c_j \cdot \dot{x}_j + k_j \cdot x_j + k_{for} \cdot f_E \quad (6)$$

Confrontando questa espressione con la (3) si comprende l'importanza di aumentare il guadagno di forza (k_{for}): solo se massa, coeff. di smorzamento e di rigidità del joystick sono trascurabili rispetto alla forza di retroazione ($f_j = k_{for} \cdot f_E$), la forza sulla mano (f_h) sarà in grado di riprodurre la forza esercitata dall'end-effector sull'ambiente (f_E) e l'operatore avrà la sensibilità sul compito in esecuzione.

1.2.1.1 Un metodo per aumentare il guadagno

Partendo dal presupposto che l'instabilità è legata all'elevata rigidità del sistema perché comporta picchi elevati della forza nei contatti, si riporta in questo paragrafo un sistema proposto dal *NASA Jet Propulsion laboratory, Pasadena, California*, atto a ridurre la rigidità intrinseca del sistema fisico.

La retroazione non viene più compiuta direttamente sulla forza, ma sulla differenza tra la posizione impartita con il joystick e quella assunta dall'end-effector. Per far questo è necessario creare un ulteriore anello di controllo sull'attuatore in modo che la forza rilevata influenzi la posizione raggiunta: questo si ottiene molto semplicemente con l'anello superiore della fig. 6. Il filtro passa basso deve essere in grado di tagliare le frequenze proprie della struttura, che altrimenti presenterebbe fenomeni di instabilità.

Con questo schema si ha l'effetto di mettere una molla di minore rigidità in serie al sistema reale, così che è possibile arrivare ad un guadagno di forza pari a 2 o 3, molto maggiore che con una retroazione diretta.

Per contro, l'effetto di maggiore elasticità sarà percepita anche sulla mano dell'operatore, che avrà quindi una sensazione alterata dell'ambiente in cui sta operando.

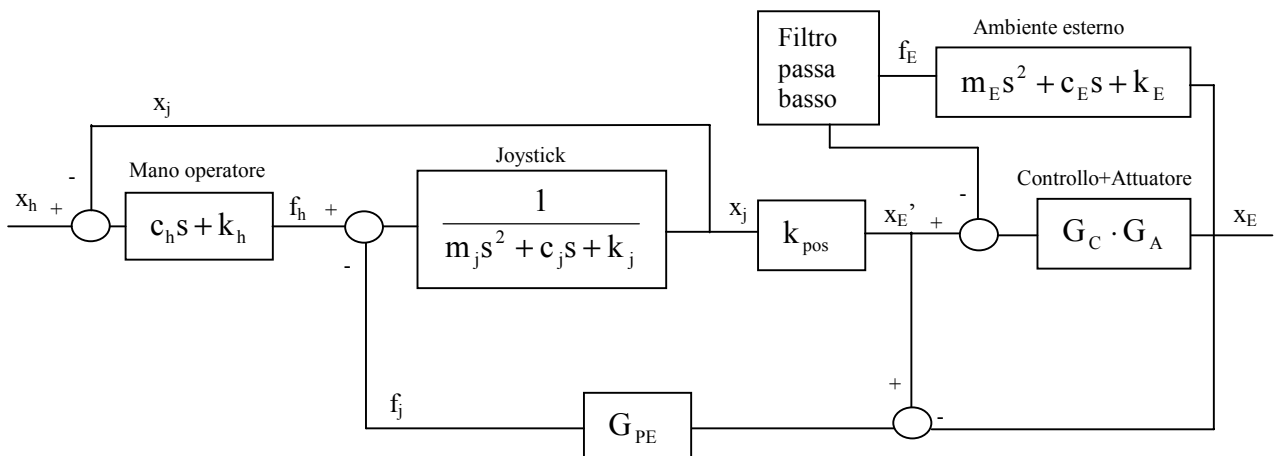


Figura 6: retroazione di forza ad elevato guadagno

1.2.2 Controllo di velocità

Il controllo di velocità è utilizzato per ottenere un posizionamento preciso quando si è in presenza di un campo di lavoro esteso. L'introduzione della retroazione di forza comporta però dei problemi, in primo luogo perché l'end-effector è fermo solo quando il joystick è in posizione centrale, inoltre, come sarà illustrato nei paragrafi successivi, si pone il problema di dare una sensibilità che conservi l'informazione dell'operazione reale.

1.2.2.1 Retroazione di forza

Se si utilizza la retroazione per produrre direttamente la forza sul joystick (f_j), si ottiene lo schema di fig. 7:

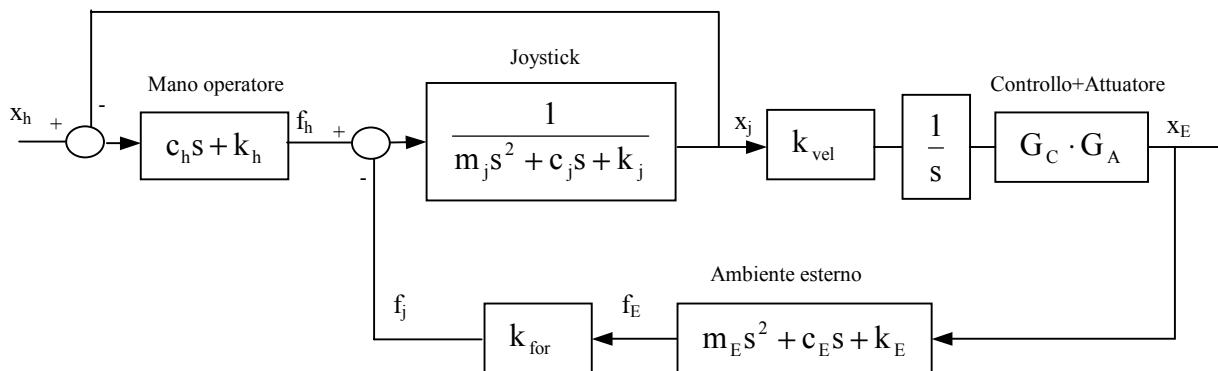


Figura 7: retroazione di forza nel controllo di velocità

La funzione di trasferimento del sistema completo è:

$$\frac{\overline{x_E}}{\overline{x_h}} = \frac{k_{vel} \cdot (c_h \cdot s + k_h)}{m_j \cdot s^3 + (c_h + c_j + k_{vel} \cdot k_{for} \cdot m_E) \cdot s^2 + (k_h + k_j + k_{vel} \cdot k_{for} \cdot c_E) \cdot s + k_{vel} \cdot k_{for} \cdot k_E}$$

Nonostante questo sia un modello semplificato, dal criterio di Routh-Hurwitz si ricava la seguente condizione di stabilità:

$$m_E \cdot c_E \cdot K^2 + [c_E \cdot (c_h + c_j) + m_E \cdot (k_h + k_j) - m_j \cdot k_E] \cdot K + (c_h + c_j) \cdot (k_h + k_j) > 0$$

In cui si è posto $K = k_{vel} \cdot k_{for}$

Ciò significa che il guadagno di forza, nel controllo di velocità, dovrà sottostare anche a questo ulteriore vincolo.

Analizzando, invece, la sensibilità derivante da questo schema, si osserva che k_{vel} non è adimensionale per la presenza del gruppo integratore successivo, e che quindi, come risulta anche nelle equazioni appena scritte, la massa del sistema ambiente (m_E) va ad influire sulla caratteristica di viscosità del joystick e, per lo stesso motivo, la viscosità viene sentita come una costante elastica. Il risultato è che con questo sistema non è possibile avere una percezione accettabile dell'ambiente esterno. Per ovviare a questo problema si può pensare di sostituire la relazione usata:

$$f_j = k_{for} \cdot f_E \quad \text{con:} \quad f_j = s \cdot k_{for} \cdot f_E$$

per riportare le variabili esterne nell'ambito corretto.

In questo caso la sensibilità dovrebbe riacquistare maggiore naturalezza, però rimangono dei limiti inaccettabili: basti pensare che in questo modo una forza costante non ha alcuna influenza sulla retroazione.

1.2.2.2 Retroazione di rigidità

Un nuovo schema in controllo di rigidità è stato sviluppato da S.E. Salcudean [1] per aggirare i problemi visti sopra. La forza di retroazione viene utilizzata per variare la costante elastica del joystick: il risultato è un sistema non lineare difficile da esaminare, ma che ha dato prestazioni soddisfacenti nelle prove sperimentali [2].

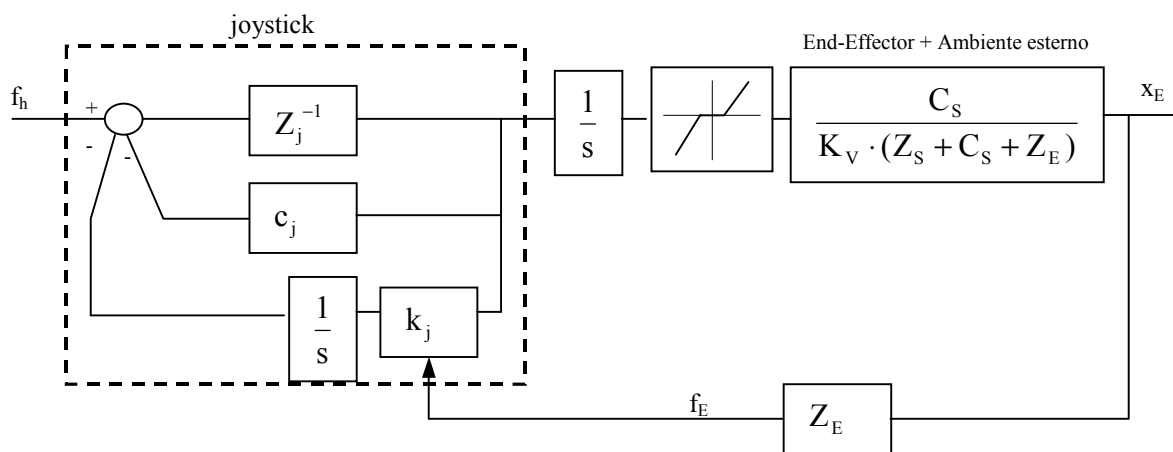


Figura 8: retroazione di rigidità nel controllo di velocità

Come per tutti i sistemi in cui la posizione del joystick determina una velocità, anche in questo caso il punto di equilibrio si raggiunge solo per $x_j = 0$. In presenza di una forza di comando (f_h) costante, non è possibile avere equilibrio perché qualsiasi valore di k_j comporta un valore di x_j diverso da zero.

Per ovviare a questo è stata introdotta una zona morta, rappresentata nello schema a blocchi, all'interno della quale la velocità dell'end-effector rimane nulla. Questa complicazione non ha permesso di trovare una soluzione analitica alla stabilità, ma la simulazione numerica ha mostrato di convergere per una vasta varietà dei parametri caratteristici.

Con questo schema l'operatore avverte i carichi e le sollecitazioni agenti sull'end-effector, e quindi è in grado di reagire di conseguenza, ma ciò avviene in modo artificioso, che non mantiene la corrispondenza delle variabili fisiche. Pertanto, sebbene le prove sperimentali abbiano dimostrato un miglioramento nelle operazioni eseguite, rimane un dubbio sull'effettiva sensibilità fornita all'operatore.

1.3 UN SISTEMA PER MUOVERSI IN AMPI SPAZI DI LAVORO (NTRFC)

Quando si ha la necessità di eseguire un'operazione in uno spazio di lavoro ampio, il controllo più indicato è quello di velocità, d'altro canto si è visto che il contatto con una superficie può essere gestito solo da un controllo di forza. Per combinare queste due esigenze si deve pertanto pensare ad un sistema in grado di passare alla modalità più appropriata in base alle sollecitazioni esterne.

Il sistema che viene qui riportato, "Naturally-transitioning rate-to-force control" proposto da R.L. Williams II, è stato studiato per l'impiego nella telemanipolazione e consente di spostarsi gradualmente tra le due modalità senza interventi artificiali.

Nel movimento libero, quando non viene rilevata alcuna forza, il comando è puramente in controllo di velocità, mentre, in presenza di un ostacolo, la forza esercitata sul joystick corrisponde a quella, in scala, sull'end-effector; nelle situazioni intermedie il controllo risulta misto.

Questo viene ottenuto con l'anello più interno dello schema di fig. 9, in cui la forza rilevata costituisce un termine sottratto al comando di velocità impartito con la posizione del joystick.

La retroazione sul joystick, invece, non costituisce una parte fondamentale del sistema, tanto da poter essere omessa, ma si rifà, tale e quale, alla retroazione vista nel controllo di posizione, con gli stessi effetti e gli stessi limiti.

La caratteristica principale di questo schema si può riassumere nella capacità di adattare il segnale di comando agli stimoli esterni, per produrre un movimento che riduce le sollecitazioni da contatto: l'impiego più naturale è quello della telemanipolazione, dove un eventuale disallineamento viene corretto automaticamente.

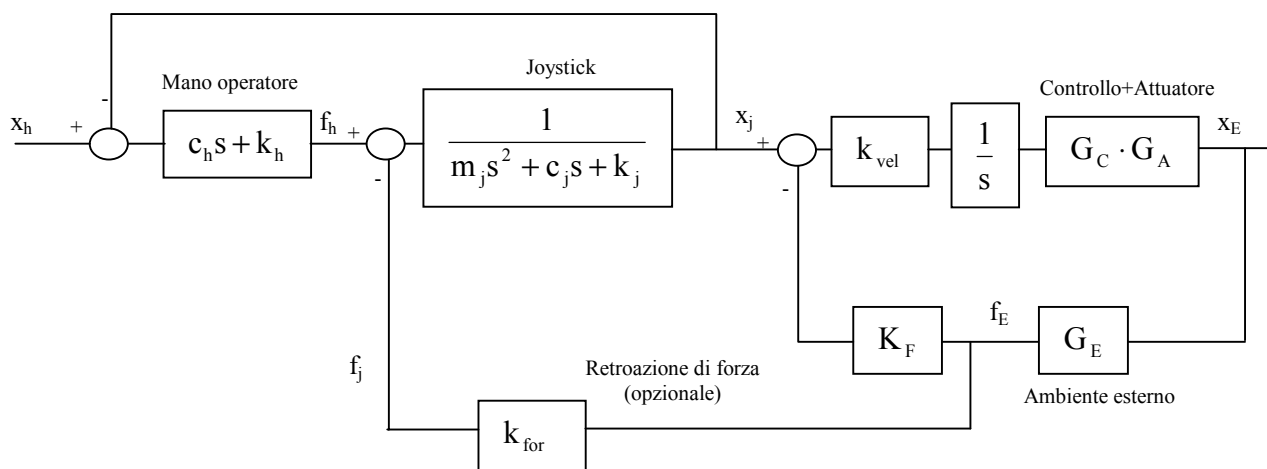


Figura 9: Naturally-transitioning rate-to-force control

Per studiare il transitorio tra il movimento libero ed il contatto con un ostacolo, si prende il seguente modello di attuatore e ambiente esterno:

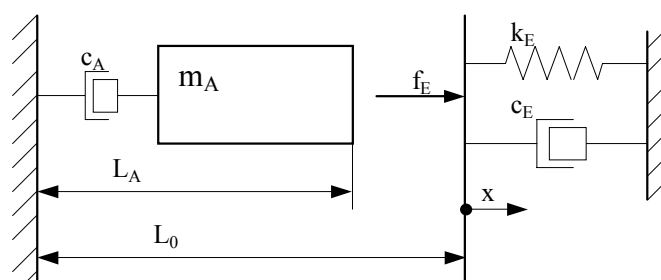


Figura 10: modello dell'attuatore per il contatto con un ostacolo

In fig. 11 si riporta lo schema a blocchi del solo anello interno riprodotto con Simulink, nel quale sono state definite le seguenti funzioni di trasferimento:

- Controllo: $G_C = k_D \cdot s + k_P$

- Attuatore: $G_A = \frac{1}{m_A \cdot s^2 + c_A \cdot s}$

- Ambiente esterno: $G_E = c_E \cdot s + k_E$

I valori dei parametri fisici e delle costanti di controllo utilizzati nella simulazione sono i seguenti:

- Attuatore: $m_A = 1 \text{ kg}$ $c_A = 0.4 \text{ N/(m/s)}$
 - Ambiente esterno: $L_0 = 0.1 \text{ m}$ $k_E = 1000 \text{ N/m}$ $c_E = 1 \text{ N/(m/s)}$
 - Controllo: $k_{vel} = 10 \text{ (m/s)/m}$ $k_F = 0.02 \text{ (m/s)/N}$
 $k_P = 500 \text{ N/m}$ $k_D = 75 \text{ N/(m/s)}$

Si impone una posizione fissa per il joystick ($x_j = 0.05 \text{ m}$), e si osserva il comportamento di posizione e forza dell'attuatore.

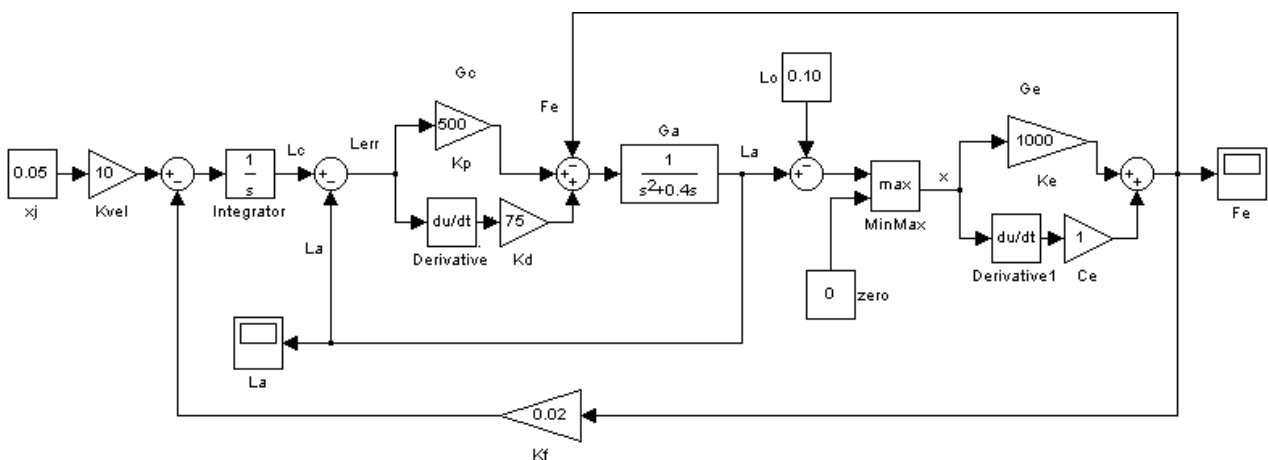


Figura 11: simulazione con Simulink del contatto con un ostacolo

Si riportano i diagrammi della forza esercitata dall'attuatore e della relativa posizione in funzione del tempo:

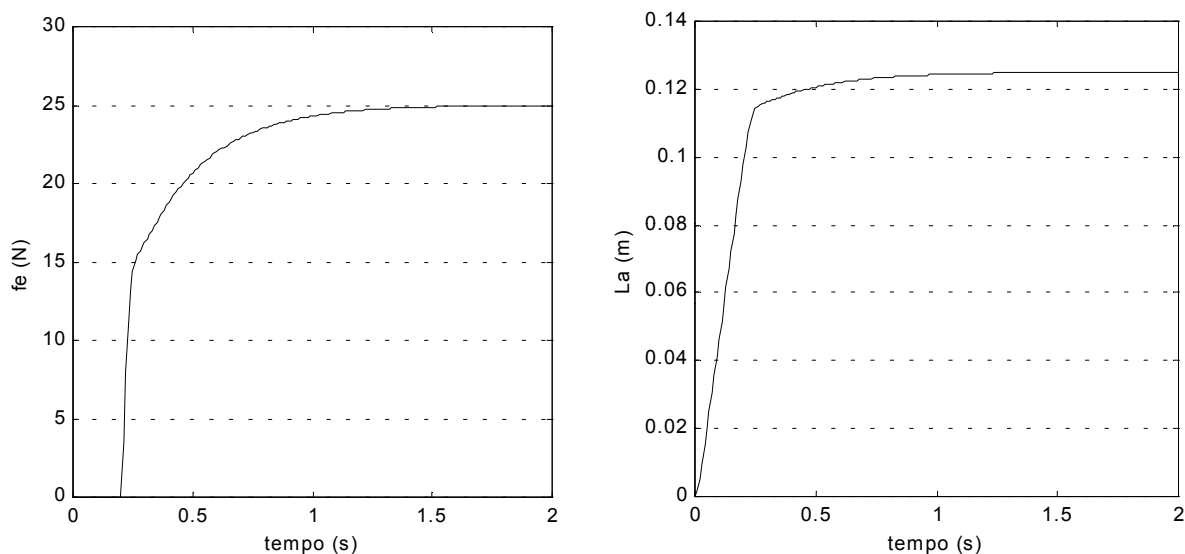


Figura 12: diagrammi di forza e posizione relativi all'attuatore

Nel grafico di destra della fig. 12 si vede che nel primo tratto, in corsa libera, l'attuatore si muove a velocità costante fino a raggiungere l'ostacolo. Poi la forza di contatto sale rapidamente (grafico di sinistra), per assestarsi, infine, ad un valore corrispondente alla posizione del joystick; allo stesso tempo la velocità diminuisce fino ad annullarsi con il raggiungimento della posizione di equilibrio. Il tutto avviene in modo progressivo e senza oscillazioni.

Questo risultato è stato ottenuto calcolando opportunamente le costanti del controllo e della retroazione (k_F), e dimostra l'efficacia di questo schema nel gestire il passaggio dal controllo di velocità a quello di forza in modo continuo.

L'operatore, quindi, non ha la sensibilità sull'operazione, ma è in grado di eseguire la manovra più velocemente e con minori forze di contatto perché è l'attuatore stesso ad "adattarsi" al contesto ambientale.

Cap. 2: IL PROGETTO JOYFORCE

2.1 CONSIDERAZIONI INIZIALI

L'idea principale da cui si è partiti è che, per dare la reale sensibilità all'operatore su quello che sta eseguendo l'end-effector, è necessaria la corrispondenza tra le posizioni. Si è pensato di ottenere questa relazione, in via del tutto generale, rovesciando la logica di funzionamento di uno schema tradizionale e facendo in modo che sia il joystick a seguire la posizione raggiunta dall'end-effector.

Così facendo, però, si perde completamente la variabile che normalmente rappresenta l'output del sistema di controllo, ossia la posizione impostata con la mano sul joystick, e si rende necessario rilevare un'altra grandezza fisica per impartire il comando: si è deciso, pertanto, di utilizzare la forza esercitata dalla mano, misurabile attraverso l'installazione di un apposito sensore.

La configurazione del joystick deve mutare considerevolmente rispetto ad un comando tradizionale perché completamente diversa è la sua funzionalità, e necessita, quindi, di una radicale riprogettazione. In un joystick tradizionale, infatti, la forza esercitata dall'operatore deve essere messa in relazione univoca con la posizione assunta dalla leva, ed è quindi importante ridurre gli attriti al fine di rendere la caratteristica elastica della molla di ritorno più costante possibile.

Nella nuova definizione di joystick, invece, si è in presenza di un posizionatore le cui qualità principali devono essere la velocità e la stabilità con cui raggiunge la posizione data in ingresso, senza essere influenzato da disturbi esterni, tra i quali è preponderante la forza esercitata con la mano.

A questo punto si deve definire il modo di rendere il sistema capace di interagire con l'ambiente esterno: se si utilizzasse la forza misurata sulla leva per comandare direttamente l'end-effector, si sarebbe in presenza di un sistema "cieco". Il comando viene allora elaborato combinando assieme sia la forza esercitata sul joystick che quella rilevata sull'attuatore.

Si rappresenta uno schema generale d'insieme di quanto descritto:

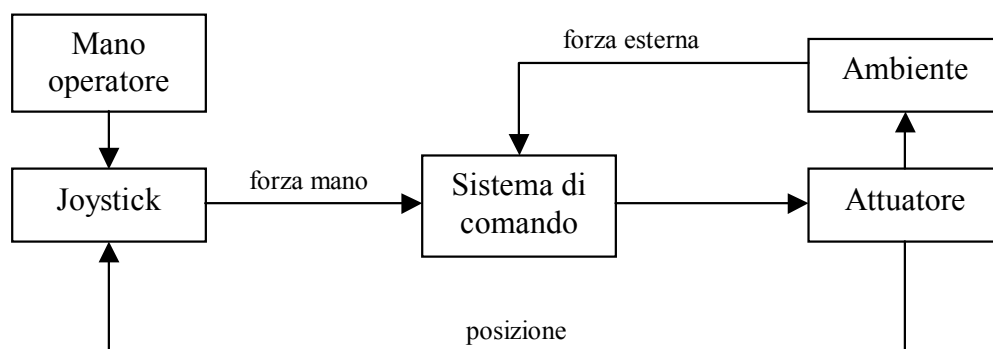


Figura 13: schema d'insieme del sistema JoyForce

Per effettuare un'analisi più dettagliata si stabilisce, intanto, la modellizzazione del sistema mano-joystick (fig. 14), ricordando, per quanto sopra descritto, che non è più prevista la molla di richiamo:

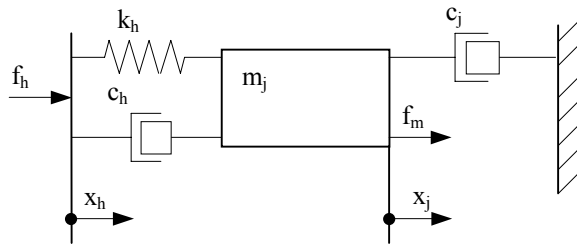


Figura 14: modello del sistema mano-joystick

Con f_m si è denotata la forza prodotta dall'azionamento del joystick.
 Nella fig. 15 si rappresenta lo schema a blocchi del sistema di posizionamento del joystick in cui x_E è la posizione dell'attuatore.

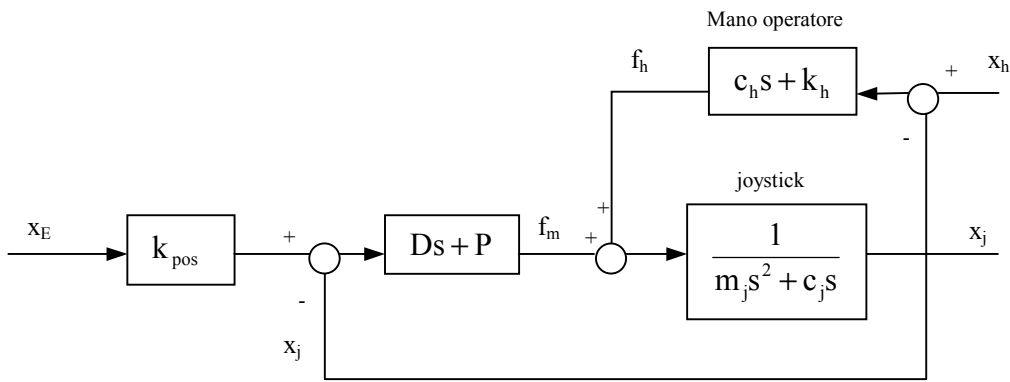


Figura 15: schema a blocchi del controllo del joystick

Il blocco $(Ds+P)$ costituisce la funzione di trasferimento del controllo della posizione, le cui costanti saranno determinate più avanti per conferire la risposta voluta.
 Qui sotto si riporta invece lo schema del sistema completo (fig. 16), che verrà successivamente ripreso nei dettagli per i diversi tipi di controllo.

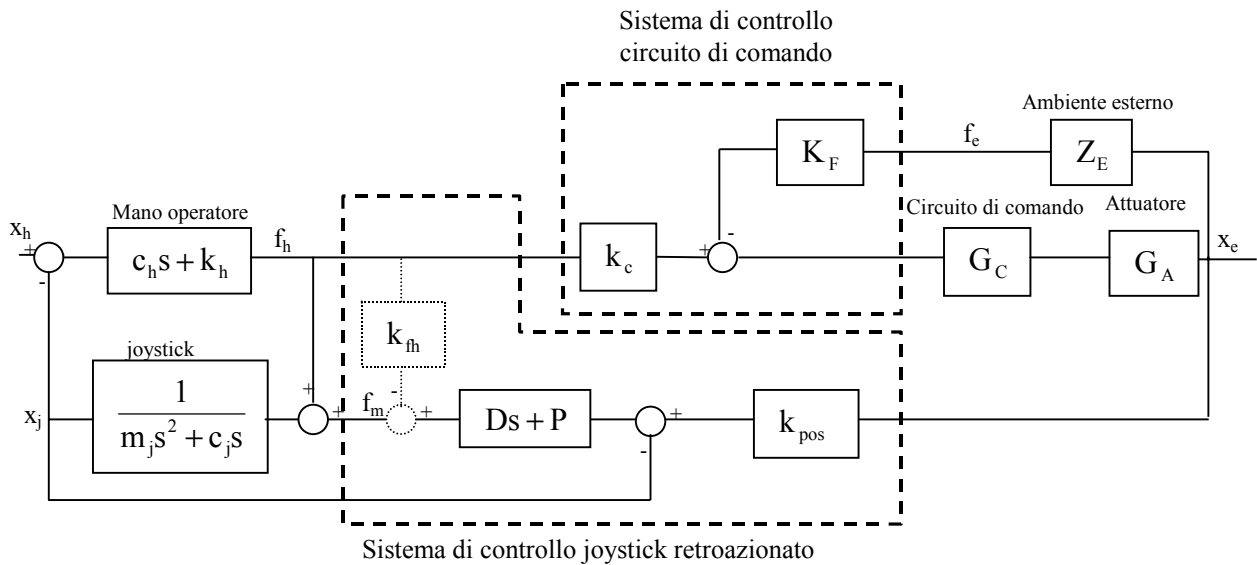


Figura 16: schema a blocchi del sistema JoyForce

Si possono distinguere due sottosistemi distinti, racchiusi a tratteggio:

- il sistema di controllo del joystick retroazionato è lo stesso visto in fig. 15, nel quale è stata inclusa la possibilità (a tratteggio) di compensare la forza esercitata dalla mano: questa opportunità sarebbe da valutare nel caso si avesse un errore statico rilevante, perché il controllo, di tipo proporzionale-derivativo, non è in grado di gestirlo.
Costituisce parte integrante del joystick, ed essendo attraversato dai segnali di input/output, funge da interfaccia per le connessioni. Per ottenere un movimento fluido del joystick la sua frequenza di aggiornamento deve essere elevata.
- Il sistema di controllo del circuito di comando si occupa della gestione dell'attuatore in funzione degli stimoli esterni.
Questo anello deve essere chiuso in prossimità dell'attuatore perché, anche in caso di telemanipolazione, deve intervenire con tempestività alle sollecitazioni provenienti dall'esterno.

I due sottosistemi, ciascuno dei quali deve essere collocato in prossimità del sistema controllato, sono parte di un anello più ampio che può funzionare a frequenze molto minori. Infatti, i due rami che lo costituiscono sono formati, superiormente sullo schema dalla forza applicata con la mano, ed inferiormente dalla posizione raggiunta dall'attuatore.

Queste due grandezze riguardano entrambe il sistema uomo, incapace di impartire comandi al di sopra di una frequenza di 10 Hz, e di distinguere ritardi sulla posizione dello stesso ordine di grandezza.

Con degli accorgimenti sul sistema di controllo, studiati in modo da mantenere la stabilità, è quindi possibile utilizzare una frequenza sull'anello ampio relativamente bassa, senza compromettere la sensibilità fornita all'operatore.

L'insieme dei due sottosistemi definisce completamente il funzionamento dell'intero sistema, e la possibilità di comunicazione tra i due permette di ridefinirne la modalità di funzionamento. Tale peculiarità potrebbe essere utilizzata per implementare la commutazione dello stesso joystick tra più logiche di controllo.

Un'osservazione importante è che nei due anelli di retroazione si eseguono operazioni tra grandezze omogenee, confermando la sensazione di naturalezza che è in grado di restituire il sistema.

2.2 DUE TIPI DI CONTROLLO

Nello studiare più in dettaglio lo schema presentato, ci si è posti il problema di definire il modello dell'attuatore. Fin dalla prima analisi, pur volendo mantenere la massima semplificazione, ci si è accorti che era indispensabile differenziare un sistema idraulico da uno elettrico.

L'apparato di regolazione di un cilindro idraulico, infatti, può essere costituito da una valvola elettroidraulica oppure, per migliorare il rendimento energetico, da una pompa a portata variabile; in tutti e due i casi, però, la regolazione viene effettuata sulla portata di fluido che, trasferita nell'ottica dell'attuatore, si traduce in un controllo di velocità. Pensando poi alla caratteristica di una pompa volumetrica per fluidi incompressibili, nel caso ideale, si capisce anche che il movimento non è in nessun modo influenzato dalle forze resistenti incontrate. In questo caso, allora, il modello più semplice che si è pensato, senza tener conto della massa e delle accelerazioni massime raggiungibili, è stato quello di un sistema di comando capace di definire direttamente la velocità dell'attuatore (**controllo di velocità**).

In un azionamento elettrico, invece, pensato come un sistema movimentato da un motore, la caratteristica principale è determinata dalla presenza di una massa (o un momento d'inerzia), e la grandezza che si è normalmente in grado di controllare è la coppia, che va ad incidere sulla forza sviluppata dall'attuatore.

Il modello che si è preso è stato dunque quello di una massa e di una forza ad essa applicabile (**controllo di forza**).

2.2.1 Il controllo di velocità

Per quanto assunto fino ad ora, lo schema si può rappresentare nel seguente modo:

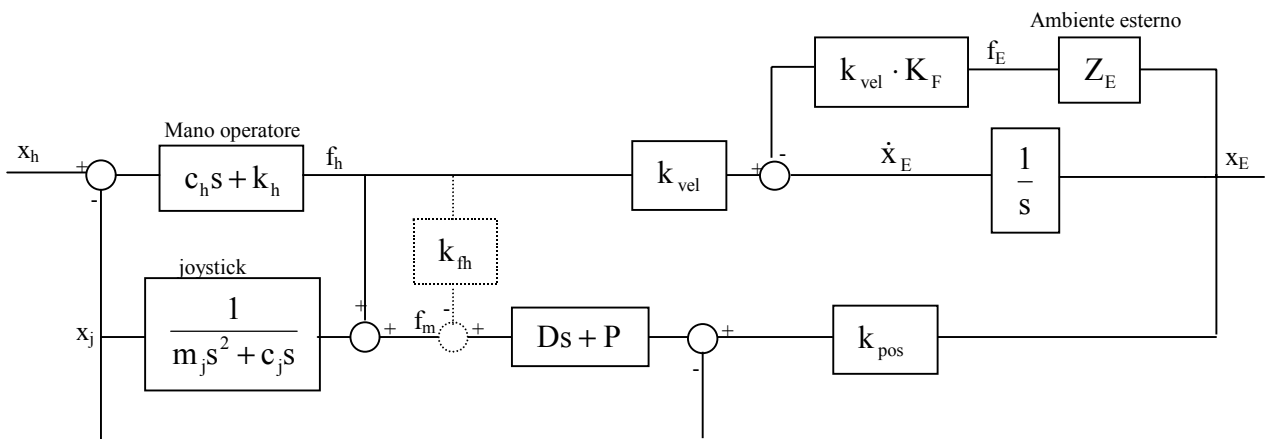


Figura 17: JoyForce applicato ad un controllo di velocità (attuatori idraulici)

Si vuole porre l'attenzione, adesso, sugli effetti nella stabilità derivanti dalla presenza della mano nell'anello esterno di retroazione. Si ipotizza pertanto, alla luce della maggiore frequenza di funzionamento e, di conseguenza, al minore tempo di assestamento, che il joystick segua istantaneamente la posizione in ingresso ($x_j = x_E$).

Si pone inoltre il caso in cui il movimento avvenga in campo libero, e quindi con forza esterna nulla ($f_E = 0$).

Si può allora schematizzare il sistema senza i due anelli interni di retroazione:

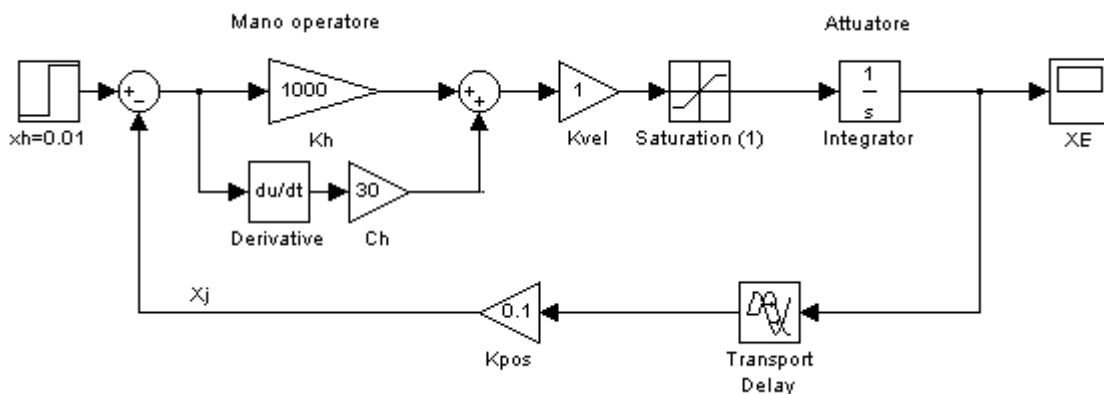


Figura 18: simulazione del solo anello esterno nel controllo di velocità

Sono stati fissati i seguenti valori delle costanti caratteristiche:

- mano operatore:	$k_h=1000 \text{ N/m}$	$c_h=30 \text{ N/(m/s)}$
- costanti del sistema:	$k_{vel}=1 \text{ (m/s)/N}$	$k_{pos}=0.1 \text{ (fattore di scala)}$

Inoltre è stata posta la velocità massima dell'attuatore pari a 1 m/s.

È stato scelto un segnale d'ingresso a gradino per la posizione della mano: $x_h=1 \text{ cm}$ all'istante $t=0.1$ secondi.

Nell'anello di retroazione è stato introdotto un blocco di ritardo per includere, in primo luogo, il ritardo tipico di tutti i sistemi idraulici, ed in modo più esteso il ritardo derivante da un'applicazione di teleoperazione.

Nella fig. 19 sono presentati i diagrammi della risposta, a sinistra senza ritardo e a destra con un ritardo di 0.02 secondi.

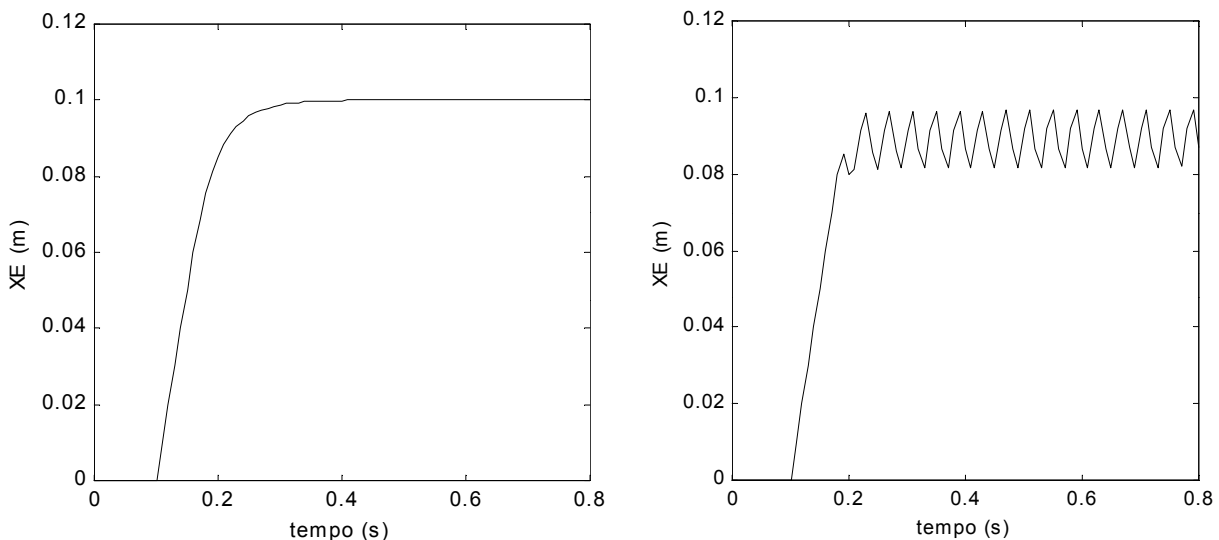


Figura 19: diagrammi della posizione senza ritardo e con ritardo di 0.02 secondi

In assenza di ritardo, la posizione dell'attuatore si avvicina a quella imposta dalla mano, dapprima con la velocità massima, e poi in modo asintotico: si può dire essere la miglior risposta auspicabile.

Introducendo il ritardo, però, si vede che la posizione finale non è stabile ma continua ad oscillare: questo comportamento si spiega perché quando l'attuatore passa per la posizione di equilibrio, il joystick si trova ancora in una posizione precedente, e quindi la forza applicata dalla mano è diversa da zero. Questo porterà a superare detta posizione e a riprendere il ciclo.

Il sistema appena descritto è stato il primo ad essere implementato sperimentalmente e, come si è visto nel grafico del modello matematico, in alcune circostanze entrava in oscillazione. Il fenomeno era meno evidente rispetto alla simulazione, e ciò si spiega perché dipende prevalentemente dalle caratteristiche della mano.

Nella realtà, infatti, il modello è notevolmente più complesso e varia considerevolmente, oltre che tra una persona e l'altra, anche dal modo in cui viene impugnata la leva.

Lo studio eseguito è stato utile perché ha permesso di capire le cause, e si è potuto intervenire per risolverle.

Il sistema è stato modificato inserendo un filtro passa basso immediatamente prima del comando dell'attuatore: in termini meccanici questo significa smorzare il segnale di ingresso, che diventa meno sensibile alle oscillazioni.

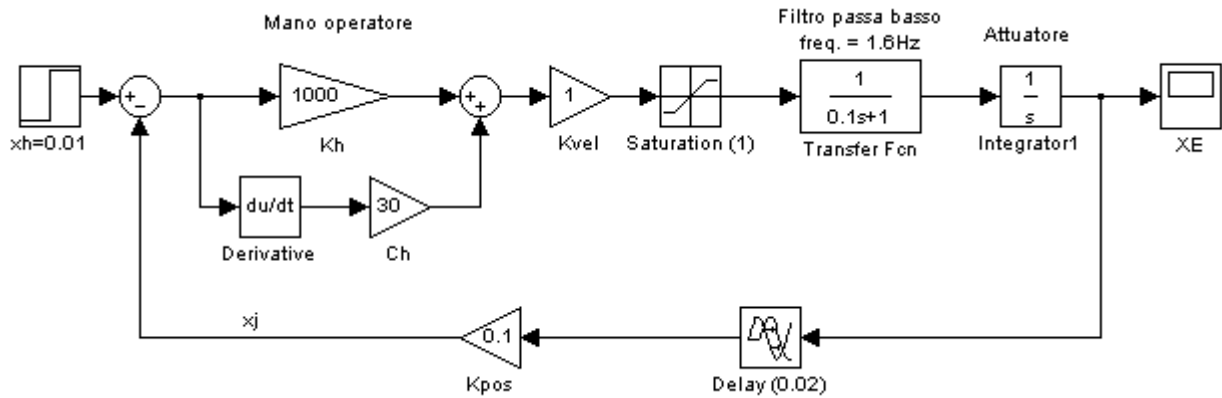


Figura 20: controllo di velocità stabilizzato con l'introduzione del filtro passa-basso

Sono stati mantenuti i valori delle costanti caratteristiche visti in precedenza, con il ritardo di 0,02 secondi, e si è ottenuto il seguente diagramma della risposta:

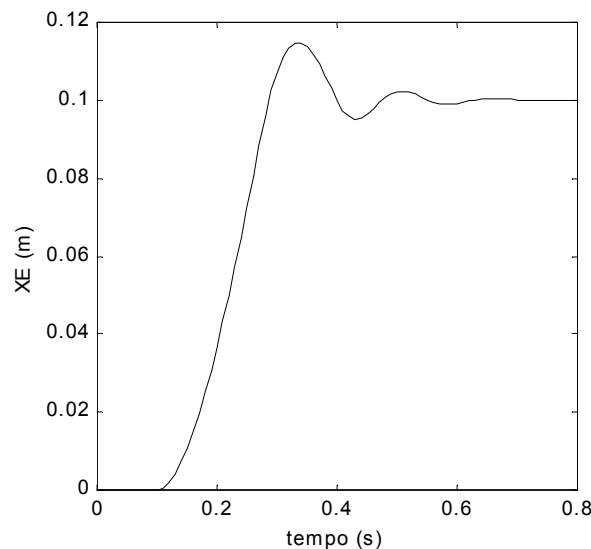


Figura 21: risposta con ritardo di 0.02 secondi e filtraggio del segnale

Il sistema presenta ora un comportamento più progressivo e, a fronte di un minimo aumento del tempo di salita, riesce ad assestarsi sulla posizione di equilibrio.

2.2.2 Il controllo di forza

In questo schema la funzione di trasferimento dell'attuatore è quella di una massa.

La forza impartita con la mano f_h costituisce il valore di riferimento per il modulo di controllo dell'attuatore.

In caso di movimento in campo libero, però, quando non si hanno forze di contatto, l'attuatore si muoverebbe in solo controllo di accelerazione, con il pericolo avere velocità e tempi di oscillazione troppo elevati.

Attraverso il sistema di controllo, allora, si è aggiunto un termine proporzionale alla velocità, che si traduce, nel sistema fisico, nell'introduzione di una viscosità fittizia.

In questo modo aumenta lo smorzamento della funzione di trasferimento complessiva e la risposta diviene più stabile.

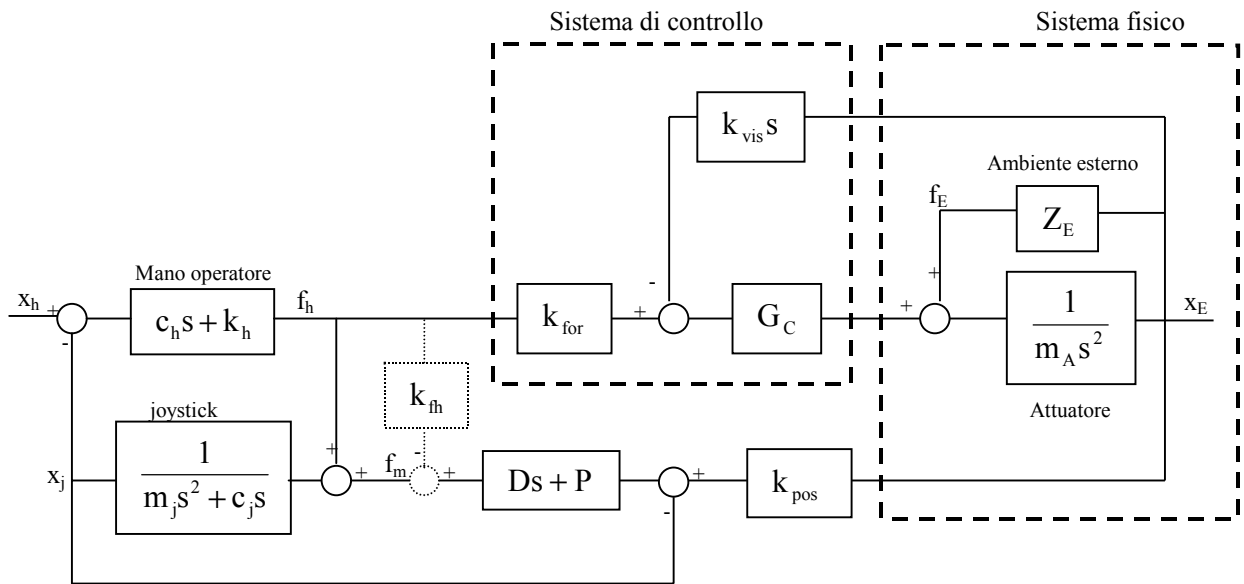


Figura 22: JoyForce applicato ad un controllo di forza (attuatori elettrici)

Come nel caso precedente, si è voluto analizzare il comportamento in corsa libera ed in presenza di un ritardo nell'anello ampio di retroazione.

Le costanti ed i valori assunti per le grandezze del sistema si rifanno a quelli stabiliti in precedenza, con la comparsa dei nuovi termini:

- massa equivalente dell'attuatore: $m_A = 100 \text{ kg}$
- costante di scala della forza: $k_{\text{for}} = 1000$
- costante di smorzamento: $k_{\text{vis}} = 1000 \text{ N/(m/s)}$

Lo schema studiato con Simulink è il seguente:

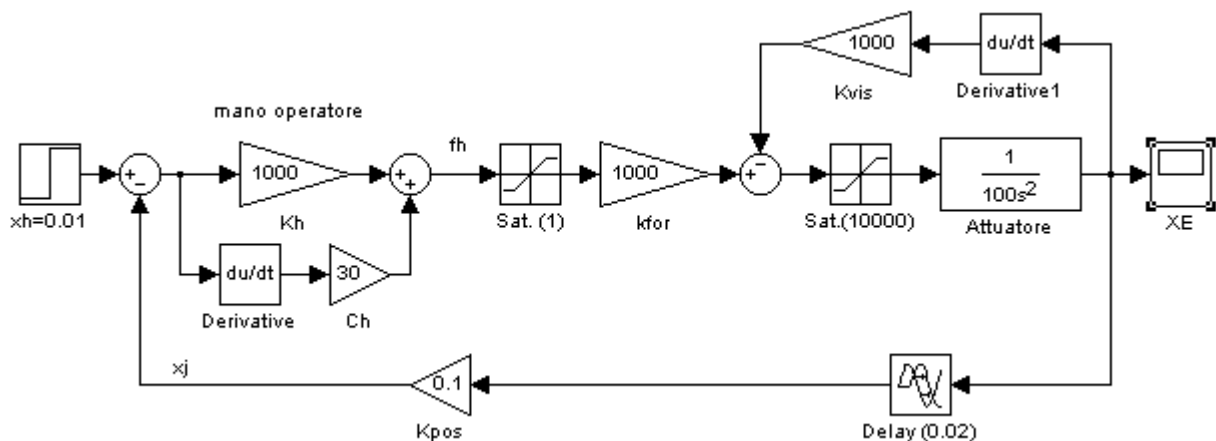


Figura 23: simulazione del solo anello esterno nel controllo di forza

Sono stati inseriti due blocchi di saturazione: il primo rappresenta il campo di misura del sensore di forza posto sul joystick, ed il secondo considera la coppia massima erogabile dal motore.

Nella fig. 24 si riportano i diagrammi della risposta, con l'intento di valutare l'efficacia della viscosità fittizia introdotta.

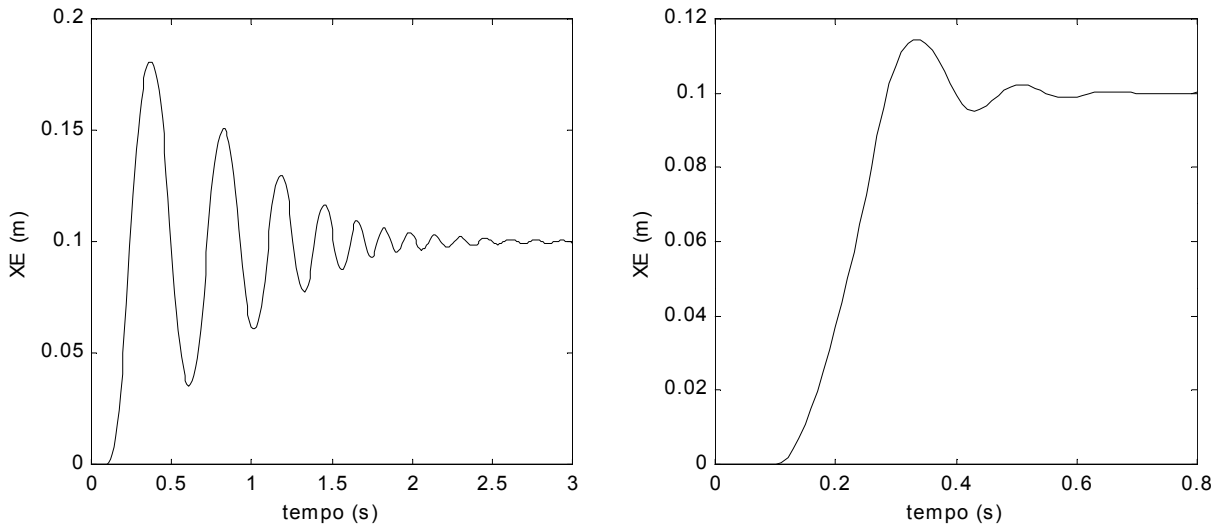


Figura 24: diagrammi della risposta prima e dopo l'introduzione della viscosità fittizia nella retroazione

Il primo grafico è relativo all'assenza della retroazione ($k_{vis} = 0$), e presenta una risposta che si assesta sulla posizione di equilibrio dopo molte oscillazioni.

Il grafico di destra, invece, mette in risalto i miglioramenti derivanti dall'aumento della viscosità complessiva. Si ribadisce che ciò è stato ottenuto intervenendo esclusivamente sul sistema di controllo.

2.3 DEFINIZIONE DEL SISTEMA

Dopo che lo studio del modello matematico ha assicurato un funzionamento stabile ed efficace, si è passati alla fase sperimentale.

L'obiettivo prefisso è stato quello di indagare le prestazioni che il sistema è in grado di fornire, principalmente in relazione alla precisione ed alla sensibilità di esecuzione del compito.

In assenza delle attrezzature necessarie per svolgere le prove, si è pensato di passare alla realizzazione del joystick, comprensiva dell'interfaccia di controllo, e di simulare la rimanente parte del sistema tramite l'ausilio di un computer.

La necessità di chiudere l'anello di retroazione dell'attuatore con una elevata frequenza di aggiornamento per conferirgli tempestività di intervento, ha condizionato la scelta di implementare via software anche il controllo del circuito di comando.

In una eventuale applicazione reale, invece, dovrebbe essere costituito da un'unità autonoma, collocata in prossimità della centralina di comando dell'attuatore, tutt'al più avente l'opportunità di scambiare dati con l'interfaccia joystick, per permettere un funzionamento flessibile.

Questa configurazione, rispetto al sistema reale, è più penalizzante perché comporta una discretizzazione delle variabili ed introduce un tempo di ritardo per le operazioni di input/output.

Proprio quest'ultimo effetto, nel caso specifico, è stato utile perché ha permesso di verificare la validità del sistema anche nelle operazioni di telemanipolazione.

Nella fig. 25 si riporta lo schema del controllo di velocità, con la suddivisione effettuata nella definizione del sistema.

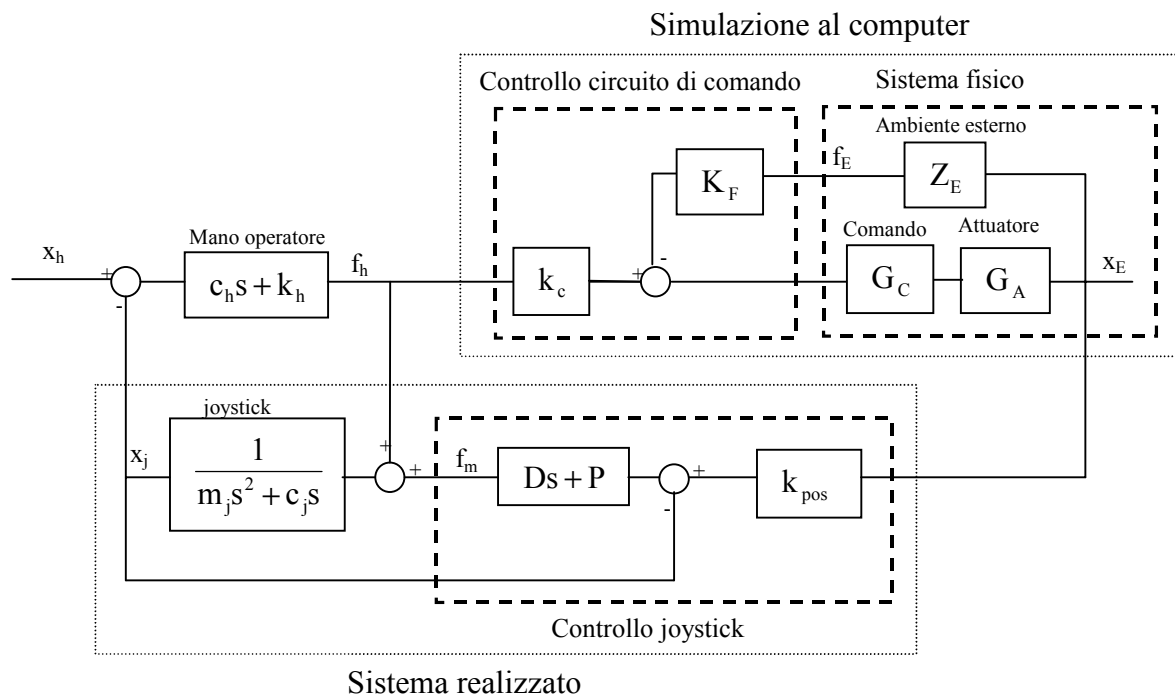


Figura 25: layout del progetto JoyForce

Il capitolo prosegue occupandosi dei seguenti argomenti:

- realizzazione del joystick
- realizzazione dell'interfaccia di controllo
- Implementazione software di:
 - Controllo circuito di comando
 - Sistema fisico: comando, attuatore, ambiente esterno

2.3.1 Progetto del joystick

D'ora in avanti, con il termine joystick, ci si riferirà, in un'accezione più ampia, al sottosistema costituito dalla parte meccanica, dai sensori e dall'interfaccia di controllo.

Questa non è una mera definizione formale, ma l'indicazione dell'intento di costituire, mediante la scelta di componenti omogenei, un'unità autonoma, trasportabile ed utilizzabile in ambiti diversi.

Sono stati fissati i seguenti dati di progetto:

- massima forza sulla leva: 20 N
- braccio della leva: 150 mm
- ampiezza della corsa: $\pm 60^\circ$

Si calcola la coppia nominale:

$$C_{\text{nom}} = F_{\text{max}} \cdot b = 20 \cdot 0.15 = 3 \text{ N} \cdot \text{m}$$

Si riporta lo schema funzionale dell'insieme:

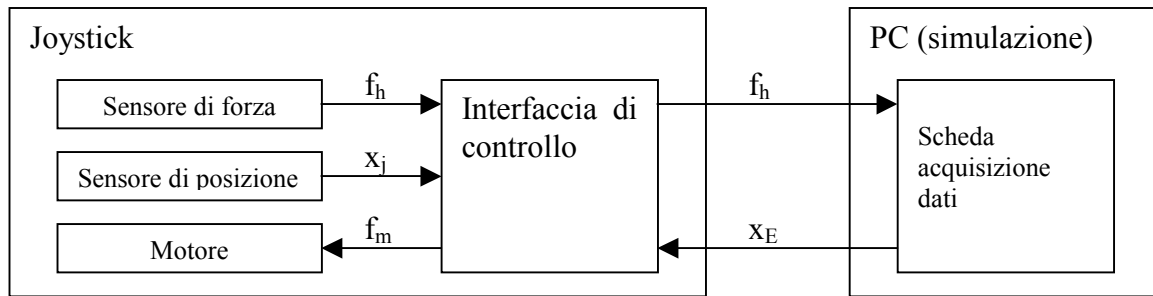


Figura 26: schema funzionale del progetto JoyForce

2.3.1.1 Scelta della configurazione e dei componenti

Stabiliti i termini del progetto si è passati alla definizione della configurazione d'insieme, ben consci della stretta connessione con la tipologia dei componenti. Gli elementi fondamentali della parte meccanica e di quella riguardante i sensori, sono stati così individuati:

- Motore elettrico: le possibili alternative sono un motore in corrente continua, un passo-passo oppure un brushless. Visto che non erano richiesti particolari requisiti si è optato per la soluzione più economica del motore in corrente continua.
- Trasmissione: l'alternativa può essere tra una cinghia dentata e un riduttore ad ingranaggi. La prima soluzione non presenta giochi, mentre la seconda consente un rapporto di riduzione più elevato. Un comando in presa diretta non è nemmeno stato preso in considerazione perché, vista la coppia nominale di $3 \text{ N} \cdot \text{m}$, avrebbe richiesto un motore eccessivamente ingombrante.
- Sensore di posizione: può essere di tipo resistivo (potenziometro), induttivo (resolver), oppure può essere costituito da un encoder (segnale digitale).
- Sensore di forza: la tecnologia disponibile è quella piezoelettrica oppure estensimetrica. Il confronto con la disponibilità offerta dal mercato ha determinato l'utilizzo di un torsionmetro. Il segnale di coppia è riconducibile alla forza applicata sulla leva perché le due grandezze si trovano in rapporto di proporzionalità diretta.

La prima configurazione ipotizzata prevedeva una trasmissione a cinghia ed un potenziometro connesso sull'albero della leva: in questo modo non ci sarebbero stati giochi, e l'elasticità della cinghia non avrebbe avuto influenza sul posizionamento. Qui sotto se ne rappresenta lo schema.

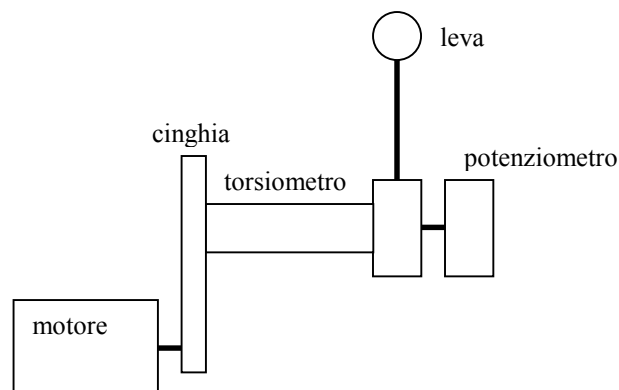


Figura 27: disposizione d'insieme nella soluzione con trasmissione a cinghia

La trasmissione a cinghia, però, non permette rapporti di riduzione superiori a 10:1, cosicché sarebbe stato necessario montare un motore di costo ed ingombri non compatibili con il progetto. Inoltre, rimaneva qualche dubbio sulla risoluzione del potenziometro e sulle difficoltà che si sarebbero incontrate nel comandare il motore. Infatti in questa configurazione, tra il motore e il sensore di posizione è interposta una trasmissione elastica e, di conseguenza, non è possibile conoscere in ogni istante la posizione esatta del motore. Più precisamente, si è in presenza di un sistema semi-definito, le cui estremità oscillano in opposizione di fase. Se il controllo ha una frequenza di aggiornamento in grado di campionare la frequenza naturale del sistema ($> 4 \cdot \omega_n$), il potenziometro rileva la posizione sfasata di 180° : questo significa che, quando il motore si trova nell'intorno della posizione di equilibrio, in relazione a questa, il sensore restituisce una misura opposta di quella reale. In questa situazione il controllo del motore si comporta nel modo opposto al raggiungimento dell'equilibrio, ed il sistema, che è fisicamente smorzato, continua ad oscillare attorno alla posizione di riferimento. Pertanto, potrebbe essere necessario prevedere un ulteriore sensore di posizione connesso al motore.

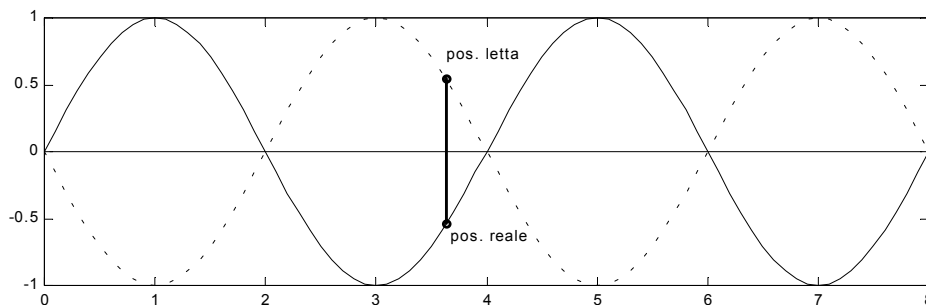


Figura 28: oscillazione nella trasmissione elastica

Anche per questa ragione, comunque, si è scelta una configurazione diversa, basata sull'utilizzo di un motoriduttore in gruppo unico con l'encoder.

Questa soluzione, oltre ad essersi dimostrata più economica, ha semplificato notevolmente l'assemblaggio dei vari componenti ed ha permesso di ottenere un insieme molto compatto.

Per contro, la posizione dell'encoder e la presenza degli ingranaggi hanno introdotto un gioco sulla leva, ritenuto accettabile per l'applicazione che si stava realizzando.

Si riporta lo schema:

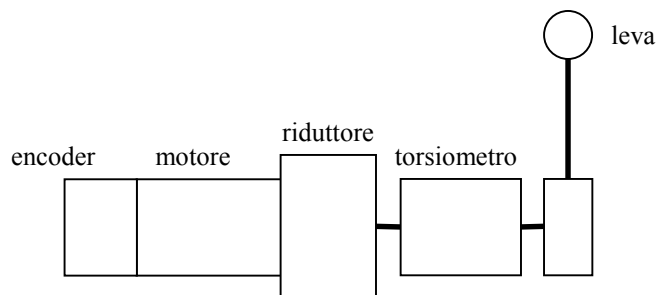


Figura 29: disposizione d'insieme nella soluzione con motoriduttore

Dati caratteristici del complesso motoriduttore-encoder:

Modello:	Pittman GM9236S027
Rapporto di riduzione:	$\tau = 65.5:1$
Rendimento della trasmissione:	$\eta_\tau = 0.77$
Coppia max continua:	$T_C = 3.39 \text{ N} \cdot \text{m}$
Velocità a vuoto:	$S_0 = 7.4 \text{ rad/s}$

Costante di coppia:	$K_T = 45.8 \cdot 10^{-3} \text{ N} \cdot \text{m} / \text{A}$ (riferita all'albero motore)
Costante di F.C.E.M. indotta:	$K_E = 45.8 \cdot 10^{-3} \text{ V}/(\text{rad}/\text{s})$ (riferita all'albero motore)
Resistenza:	$R_T = 2.49 \Omega$
Induttanza:	$L = 2.63 \text{ mH}$
Corrente a vuoto:	$I_{NL} = 0.16 \text{ A}$
Tensione di alimentazione:	$E = 24 \text{ V}$
Encoder:	512 tacche per giro

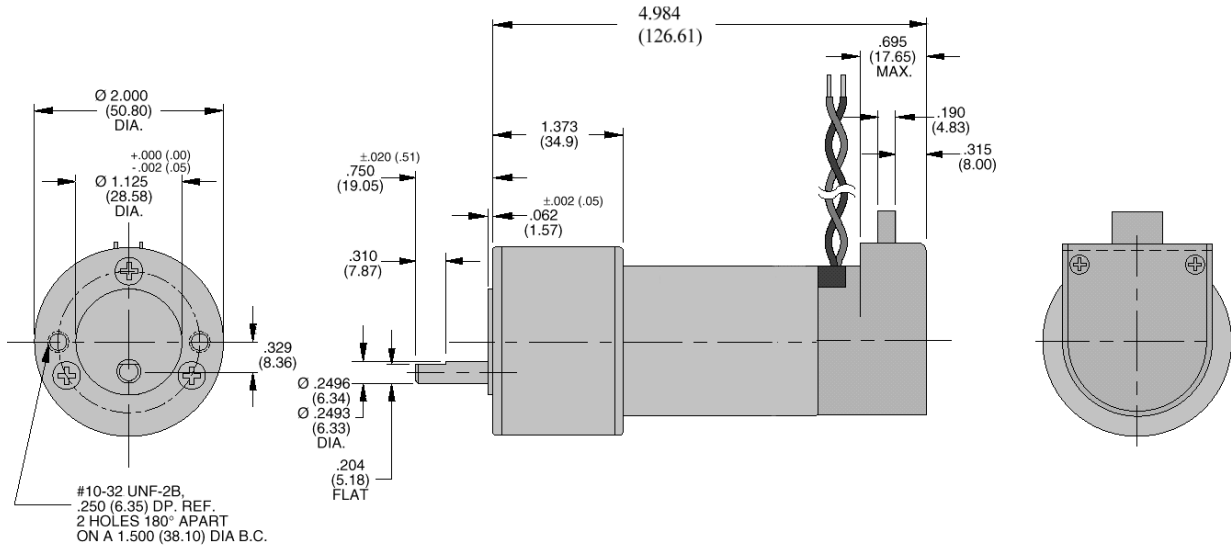


Figura 30 gruppo motoriduttore-encoder Pittman GM9236S027

Specifiche del torsionmetro (riferite al fondo scala):

Modello:	FUTEK T5160
Coppia nominale:	6 N · m
Costante di rigidezza:	1128 (N · m)/rad
Uscita nominale:	2 mV/V
Sovraccarico massimo:	150 %
Scostamento dallo zero:	±1 %
Tensione di alimentazione max:	18 V (continui o alternati)
Resistenza del ponte:	1000 Ω (nominale)
Non linearità:	±0.2 %
Isteresi:	±0.2 %
Errore di ripetibilità:	±0.05 %
Temperatura di esercizio:	da -50 a 93 °C
Peso:	65 g
Materiale:	Alluminio

Al torsionmetro è stato applicato il seguente modulo di amplificazione:

Modello:	FUTEK JM-2
Alimentazione:	da 12 a 24 Vdc
Tensione di alimentazione del ponte:	10 Vdc oppure 5 Vdc (selezionabile)
Segnale di uscita:	±5 Vdc
Risposta in frequenza:	1000 Hz (filtro a 2 poli)
Temperatura di lavoro:	da 0 a 70 °C

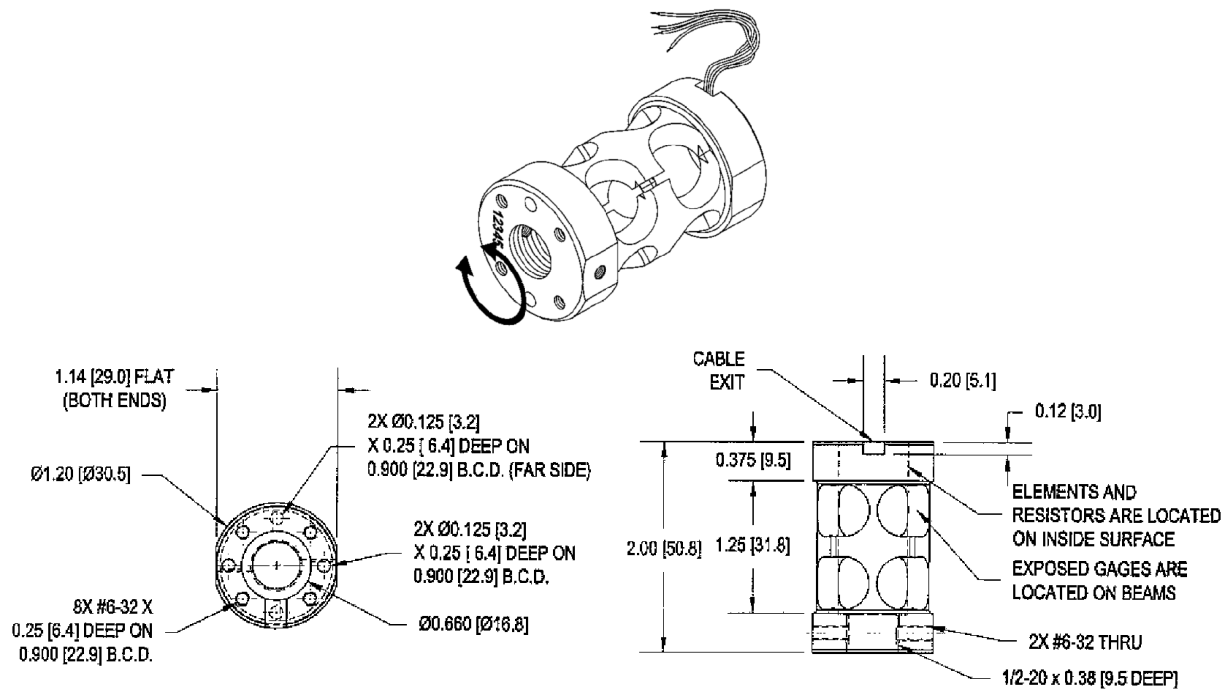


Figura 31: torsionmetro FUTEK T5160

Il principio di funzionamento del torsionmetro è quello del ponte ad estensimetri.

La forma con cui è costruito lo rende assimilabile a quattro elementi trave, con il vincolo di incastrato alle due estremità. L'applicazione di una coppia determina una deformata a S, come si rappresenta nello schema sottostante:

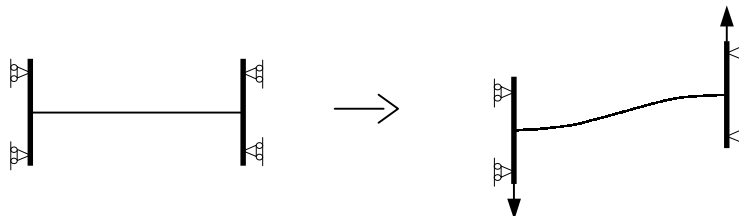


Figura 32: principio di funzionamento del torsionmetro

Gli estensimetri sono applicati nella zona in cui c'è la massima flessione.

La forma particolare conferita alle singole travi ha lo scopo di limitare la deformazione ai soli punti in cui viene rilevata, in modo da conservare un'elevata rigidità del complesso.

Per il controllo si è deciso di adottare un microcontrollore perché rappresenta la scelta più economica e flessibile. I requisiti sono stati determinati in funzione dei compiti che doveva svolgere, così riassunti:

- lettura dell'encoder
- lettura del segnale analogico della posizione di riferimento (x_E)
- lettura del segnale analogico del torsionmetro
- elaborazione dei dati
- uscita segnale per comandare il motore
- uscita segnale di forza (f_h)

Per pilotare il motore si è deciso di utilizzare un ponte a transistor (ponte a H), sia per la maggior efficienza rispetto ad un amplificatore analogico, sia perché direttamente gestibile dal microcontrollore con un segnale “Pulse Wave Modulation” (PWM).

La corrente richiesta dal motore, limitata a meno di 2 A, ha permesso di impiegare un elemento integrato (National Semiconductor LMD 18200), di cui si riporta lo schema funzionale:

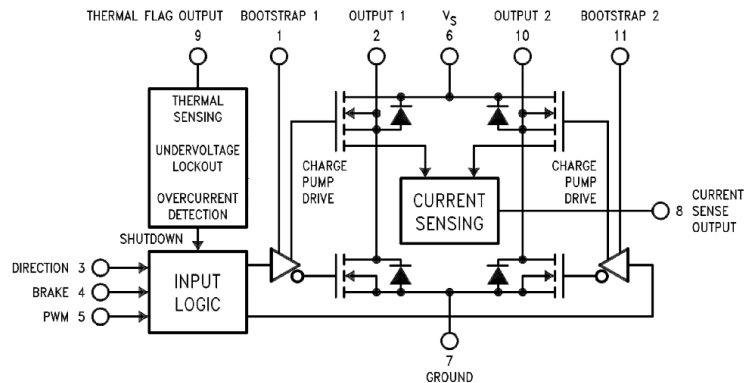


Figura 33: ponte a H (LMD18200)

L'uscita analogica (f_h) è stata ottenuta con un convertitore digitale-analogico (DAC) a 12 bit. Si è cercato un componente con un registro di memoria per poter accedere alla scrittura senza influire sull'uscita; la scelta, dettata anche dalla disponibilità di mercato, è caduta sull'integrato Analog Devices AD7396.

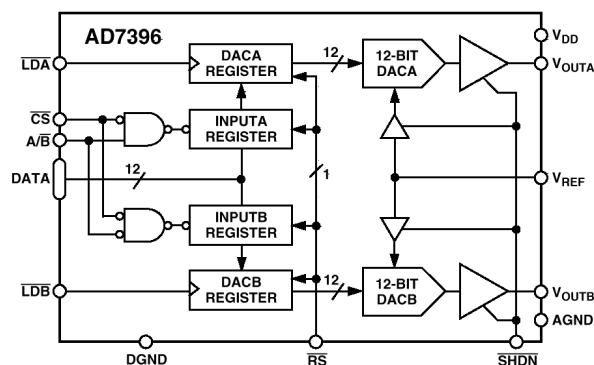


Figura 34: convertitore digitale-analogico (AD7396)

Alla luce degli elementi adottati, le caratteristiche individuate per il microcontrollore si possono riassumere nei seguenti termini:

- Memoria programmabile Flash
- 2 ingressi analogici
- 2 interrupt esterni
- 3 pin di input
- 1 timer
- 1 uscita PWM
- 2 porte di output (16 pin)

È stato scelto il microcontrollore ATMEL AVR AT90S4434, che soddisfa a tutte le richieste di progetto, e di cui si parlerà in dettaglio nel capitolo 2.3.2.

L'alimentazione del motore avviene tramite un alimentatore non controllato con tensione nominale di 24 V e corrente massima di 5 A.

2.3.1.2 Meccanica del joystick

Il progetto meccanico del joystick si è occupato di definire gli elementi di connessione tra i vari componenti, in modo da permetterne la migliore funzionalità.

In relazione a questo, il torsionmetro è stato posto tra due bicchieri, dei quali uno collegato all'uscita del motoriduttore, e l'altro a diretto sostegno della leva del joystick.

Per i due supporti sono stati utilizzati, da una parte, il cuscinetto interno al motoriduttore, e dall'altra un cuscinetto (SKF 61807) di diametro interno maggiore del torsionmetro, in modo da poter essere montato vicinissimo all'asse della leva. Il carico viene così quasi completamente sostenuto da quest'ultimo e il complesso risulta estremamente compatto. Questa configurazione garantisce anche una rilevazione accurata della forza, perché l'unico elemento di disturbo è rappresentato dall'attrito, impercettibile, del cuscinetto a sfere.

Il tutto è stato fissato ad un piano mediante una flangia per il motore ed un supporto per il cuscinetto.

Tutti i pezzi sono stati realizzati in alluminio dal laboratorio del DIMEG dell'Università di Padova. Qui sotto si riporta il disegno del modello tridimensionale, mentre i disegni costruttivi sono raccolti nell'appendice A.

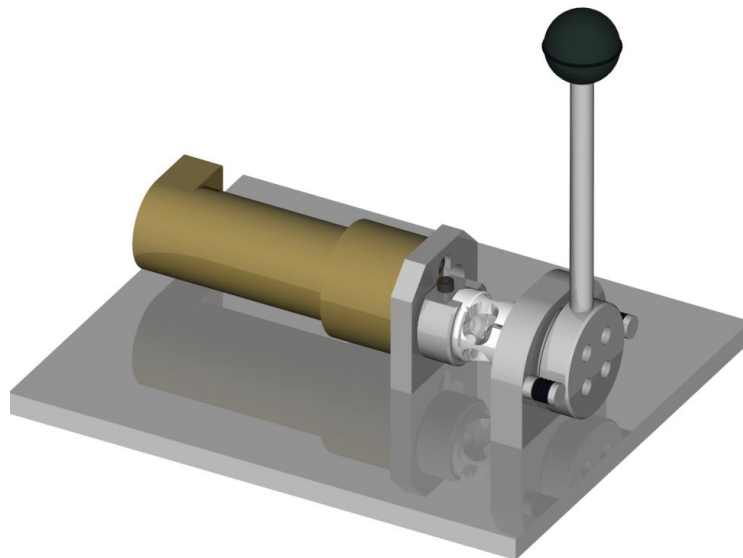


Figura 35: modello tridimensionale del complessivo

L'unico elemento che è stato sottoposto a verifica di resistenza è stata la base della leva.

Questa presenta un diametro $d = 10 \text{ mm}$ ed ha un braccio, rispetto a dove viene applicata la forza, di 125 mm . Si calcola il momento flettente per la forza nominale $F_{\text{nom}} = 20 \text{ N}$:

$$M_f = F_{\text{nom}} \cdot b = 20 \cdot 125 = 2'500 \text{ N} \cdot \text{m}$$

Il modulo di resistenza della sezione è:

$$W_f = \frac{\pi}{32} \cdot d^3 = \frac{\pi}{32} \cdot 10^3 = 98 \text{ mm}^3$$

La tensione massima sulla sezione risulta:

$$\sigma_{\text{nom}} = \frac{M_f}{W_f} = \frac{2500}{98} = 25.5 \text{ N/mm}^2$$

Questo risultato, considerato come materiale in esame dell'alluminio generico, senza fare uno studio più approfondito, non costituisce un valore preoccupante.

Come già visto in precedenza, il modello meccanico del joystick comprende una massa ed una viscosità: di seguito si illustra come sono state calcolate partendo dalle caratteristiche del motore elettrico e dei componenti meccanici.

Dal catalogo del motore si sono trovati i valori:

- Momento d'inerzia: $J_M = 7.06 \cdot 10^{-6} \text{ kg} \cdot \text{m}^2$
- Smorzamento viscoso: $K_D = 8.42 \cdot 10^{-4} \text{ N} \cdot \text{m}/(\text{rad}/\text{s})$

Si è scelto di riportare tali grandezze all'albero di uscita del motoriduttore, così sono stati calcolati il momento d'inerzia e la viscosità equivalenti del motore per il rapporto di trasmissione $\tau = 65.5$.

$$J_M^* = J_M \cdot \tau^2 = 7.06 \cdot 10^{-6} \cdot 65.5^2 = 3.03 \cdot 10^{-2} \text{ kg} \cdot \text{m}^2$$

$$K_D^* = K_D \cdot \tau^2 = 8.42 \cdot 10^{-4} \cdot 65.5^2 = 3.61 \text{ N} \cdot \text{m}/(\text{rad}/\text{s})$$

Il momento d'inerzia dei componenti meccanici in rotazione è stato calcolato in modo automatico dal modello tridimensionale. I componenti interessati all'analisi sono:

- bicchieri di destra e di sinistra
- torsionometro
- leva del joystick
- pomello sferico
- 8 viti 6-32 UNC lunghezza 1/2 pollice
- 1 vite M4x12



Figura 36: modello tridimensionale delle masse rotanti

Il momento d'inerzia complessivo dei componenti è:

$$J_C = 1.25 \cdot 10^{-3} \text{ kg} \cdot \text{m}^2$$

Si calcola, allora, il momento d'inerzia totale equivalente:

$$J_T = J_M^* + J_C = 3.03 \cdot 10^{-2} + 1.25 \cdot 10^{-3} = 3.15 \cdot 10^{-2} \text{ kg} \cdot \text{m}^2$$

2.3.1.3 Scheda di interfaccia

Nel progetto definitivo la scheda di interfaccia dovrebbe comprendere l'unità di controllo, ovvero il microcontrollore, tutti i componenti necessari al condizionamento dei segnali, e il circuito di alimentazione del motore.

Il microcontrollore dovrebbe essere programmato mediante una scheda (Atmel STK 200, collegata alla porta seriale del PC), e poi inserito definitivamente in uno zoccolo della scheda di interfaccia.

In questa fase di prototipo, invece, dovendo intervenire spesso sul codice del microcontrollore, si è reputato più pratico mantenerlo sulla scheda di programmazione e utilizzare dei connettori per trasferire le porte di input/output alla scheda di interfaccia.

In fig. 37 si riporta lo schema a blocchi delle funzioni svolte dalla scheda di interfaccia:

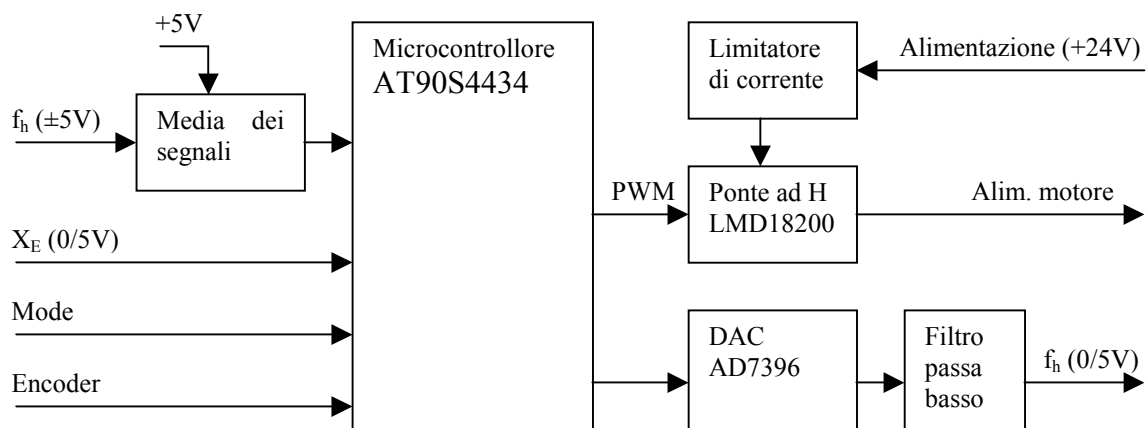


Figura 37: schema funzionale della scheda di interfaccia

Rispetto agli schemi visti in precedenza è stato aggiunto l'ingresso Mode (2 bit), il quale stabilisce la modalità di funzionamento.

L'AT90S4434 ha quattro porte di input/output, ciascuna formata da 8 pin. Ogni piedino possiede delle funzionalità particolari, per le quali si rimanda al datasheet; di seguito si riporta l'assegnazione della piedinatura:

Porta A (convertitore analogico-digitale a 8 canali):

- PA0: ingresso torsiometro
- PA1: ingresso posizione dell'attuatore (x_E)
- PA2-PA7: non utilizzati

Le porte B e C sono state dedicate alla comunicazione con il DAC (si veda il capitolo 2.3.2.3 per i dettagli)

Porta D:

- PD0: non utilizzato
- PD1: Encoder, ch. B
- PD2: Encoder, ch. A (attiva l'interrupt in salita)
- PD3: Encoder, ch. A (attiva l'interrupt in discesa)
- PD4: PWM
- PD5: Mode (bit 0)
- PD6: Mode (bit 1)
- PD7: non utilizzato

Il circuito che esegue la media trasforma il segnale proveniente dal modulo di amplificazione del torsionometro, in uno compatibile con il convertitore analogico-digitale (ADC) interno al chip.

Per trasformare un segnale variabile in un campo da -5 a $+5$ V in uno che andasse da 0 a 5 V, è stata utilizzata l'operazione di media aritmetica con un valore fisso di $+5$ V.

Visto che la resistenza degli ingressi analogici del microcontrollore è molto elevata ($100\text{ M}\Omega$), si è potuta eseguire questa operazione molto semplicemente con un partitore di tensione. I due segnali di input (quello variabile e quello fisso di $+5$ V) sono stati posti ai capi di due resistenze uguali ($10\text{ k}\Omega$), e dal punto situato nel mezzo è stato prelevato il segnale di uscita. Siccome le resistenze adoperate sono $10'000$ volte più piccole di quella del microcontrollore, l'assorbimento di corrente è trascurabile rispetto alla corrente che attraversa il partitore, e quindi la tensione non ne subisce l'influsso.

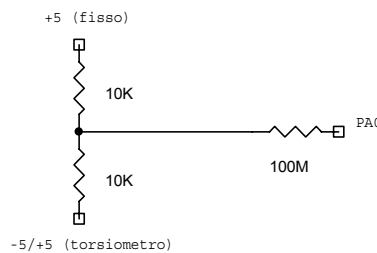


Figura 38: circuito per ottenere la media di due segnali

Nel circuito di alimentazione del motore è stato inserito un limitatore di corrente.

Questa scelta è stata fatta per motivi di sicurezza, perché in un motore a corrente continua la coppia è legata linearmente alla corrente, ed il superamento del valore massimo sopportabile dalla trasmissione porterebbe ad una rottura.

Il segnale in uscita dal microcontrollore (PWM), attraverso il ponte a H, è in grado di regolare la tensione dell'alimentazione e non la corrente. Le due grandezze sono legate da una relazione in cui è determinante la velocità di rotazione:

$$E = I \cdot R_T + K_E \cdot \omega + L \frac{dI}{dt}$$

E = tensione applicata

I = corrente

R_T = resistenza degli avvolgimenti

K_E = tensione indotta

L = induttanza degli avvolgimenti

ω = velocità di rotazione

Quando si vuole controllare la coppia ci sono due alternative: la prima è di creare un anello di retroazione con la misura della corrente assorbita, mentre la seconda si basa sulla misura di velocità per stimare la forza contro elettromotrice indotta, e calcolare la tensione di conseguenza.

In questo progetto si è optato per il secondo metodo, perché comporta meno problemi di stabilità, e la velocità può essere dedotta dalla lettura della posizione dell'encoder in due istanti successivi di tempo, senza l'impiego di ulteriori ingressi.

Per contro rappresenta un sistema in anello aperto, che non reagisce per compensare i disturbi sulla grandezza in uscita, ovvero il controllo non garantisce il non superamento del valore massimo di corrente (e di coppia). Il compito del limitatore di corrente è perciò quello di intervenire nei transitori, o nell'errato funzionamento del controllo.

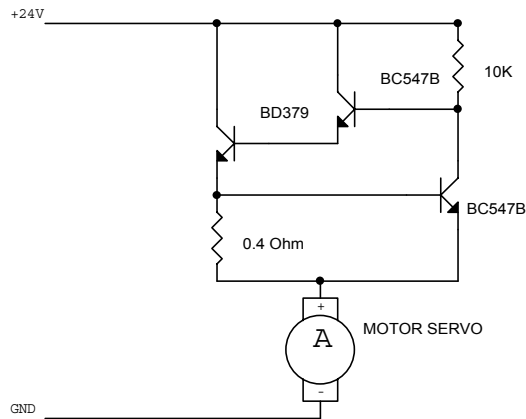


Figura 39: circuito limitatore di corrente

Il limitatore di corrente è stato costruito con un transistor di potenza (BD379) e due di controllo (BC547B) posti in connessione Darlington. Quando la tensione ai capi della resistenza di 0.4Ω supera il valore di 0.6 V (tensione di inizio conduzione tra base ed emettitore), il primo transistor di controllo entra in conduzione e manda in interdizione prima l'altro transistor di controllo, e poi quello di potenza. Il limite di corrente è così fissato a:

$$I_{\text{lim}} = \frac{0.6}{0.4} = 1.5 \text{ A}$$

che corrisponde ad una coppia sull'albero del motore di:

$$C_{\text{lim}} = (I_{\text{lim}} - I_{\text{NL}}) \cdot K_T = (1.5 - 0.16) \cdot 0.046 = 0.0616 \text{ N} \cdot \text{m}$$

in cui si è indicato con I_{NL} la corrente a vuoto.

Questa coppia viene riportata all'albero del joystick attraverso il rapporto di riduzione ed il rendimento della trasmissione:

$$C_{\text{lim(joy)}} = C_{\text{lim}} \cdot \tau \cdot \eta_{\tau} = 0.0616 \cdot 65.5 \cdot 0.77 = 3.11 \text{ N} \cdot \text{m}$$

Quindi la coppia limite si mantiene al di sotto di quella massima ammessa.

Il motore è alimentato tramite un integrato (LMD18200) che è un ponte a H con una corrente massima continua di 3 A . In questa applicazione, il segnale ricevuto in ingresso (PWM) ha il compito di alternare la conduzione dei transistor opposti, in modo da invertire il senso della tensione di alimentazione ad una frequenza costante: la tensione ai capi ha sempre lo stesso valore assoluto (pari a quello di alimentazione) ed è l'induttanza del motore a limitare la corrente. Infatti, nel motore scorre una corrente alternata limitata dall'elevata impedenza degli avvolgimenti, mentre il valore continuo dipende dalla tensione media di alimentazione, controllata dal rapporto tra il tempo in cui la conduzione avviene in ciascuno dei due sensi (fig. 40).

Questo tipo di funzionamento dell'integrato è indicato quando si debba invertire di continuo il senso di rotazione, come appunto nel caso di un posizionatore.

Nella fig. 40 si riportano il diagramma del segnale PWM per comandare il ponte a H, e lo schema di collegamento.

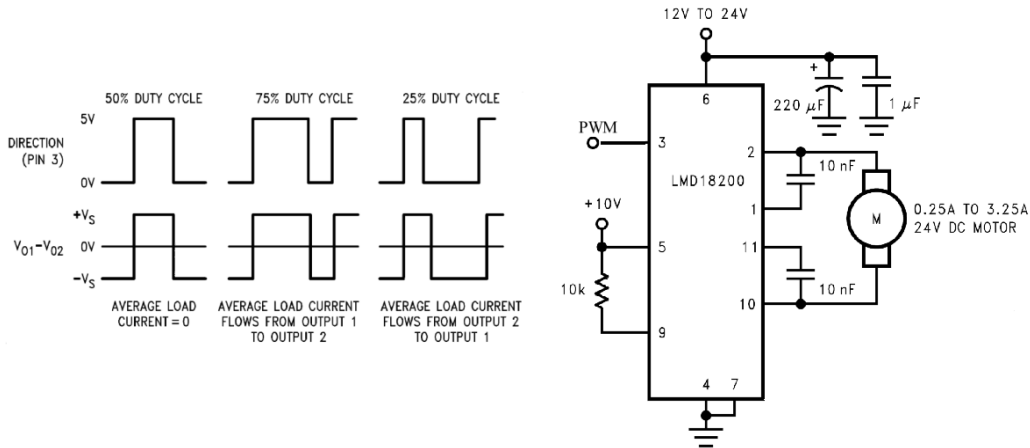


Figura 40: schema di collegamento del ponte a H (LMD18200)

Il convertitore digitale-analogico (AD7396) ha una risoluzione di 12 bit ed è collegato al microcontrollore con le porte B e C. Tutta la porta B più i primi 4 bit della porta C sono stati utilizzati per trasmettere i dati, mentre i secondi 4 bit della porta C sono stati utilizzati per le funzioni di controllo. Questo componente ha la possibilità di gestire due canali (A e B) e, anche se nel caso in esame se ne è utilizzato solo uno, nel definire il layout si è comunque pensato a rendere disponibili entrambi per eventuali sviluppi futuri.

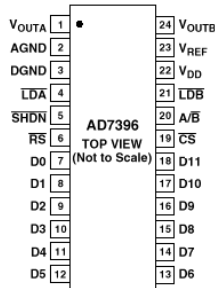


Figura 41: piedinatura del convertitore digitale-analogico

Conessioni:

- | | |
|-------------|---|
| (AT90S4434) | (AD7396) |
| - PB0-PB7 | D0-D7 |
| - PC0-PC3 | D8-D11 |
| - PC4 | $\overline{\text{LDB}}$: Trasferisce il registro INPUTB nel registro DACB |
| - PC5 | $\overline{\text{LDA}}$: Trasferisce il registro INPUTA nel registro DACA |
| - PC6 | $\overline{\text{A/B}}$: Seleziona l'input per il registro INPUTA (1) o INPUTB (0) |
| - PC7 | $\overline{\text{CS}}$: Chip Select |

L'uscita del DAC (V_{OUTA}) è stata collegata ad un filtro passa basso per evitare errori di aliasing nell'acquisizione del segnale.

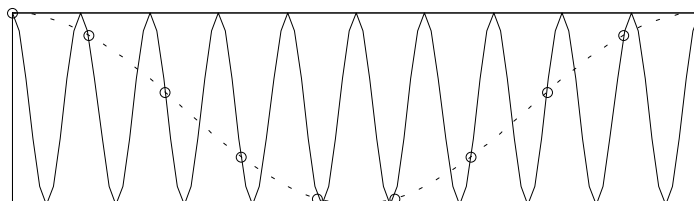


Figura 42: fenomeno dell'aliasing

Questo fenomeno (fig. 42) si presenta quando la frequenza di campionamento è inferiore alla frequenza del segnale, perché vengono rilevati come adiacenti dei punti che si trovano, in realtà, ad una distanza maggiore di una semi-onda. Il risultato è che il segnale campionato ha una frequenza di oscillazione molto minore di quella reale, difficilmente eliminabile per via numerica.

La frequenza di campionamento è di 1 kHz e si è posta la frequenza di taglio del filtro a 159 Hz.

$$f = \frac{1}{2 \cdot \pi \cdot R \cdot C} = \frac{1}{2 \cdot \pi \cdot 100 \cdot 10^3 \cdot 10 \cdot 10^{-9}} = 159 \text{ Hz}$$

Si rappresenta lo schema elettrico completo della scheda di interfaccia:

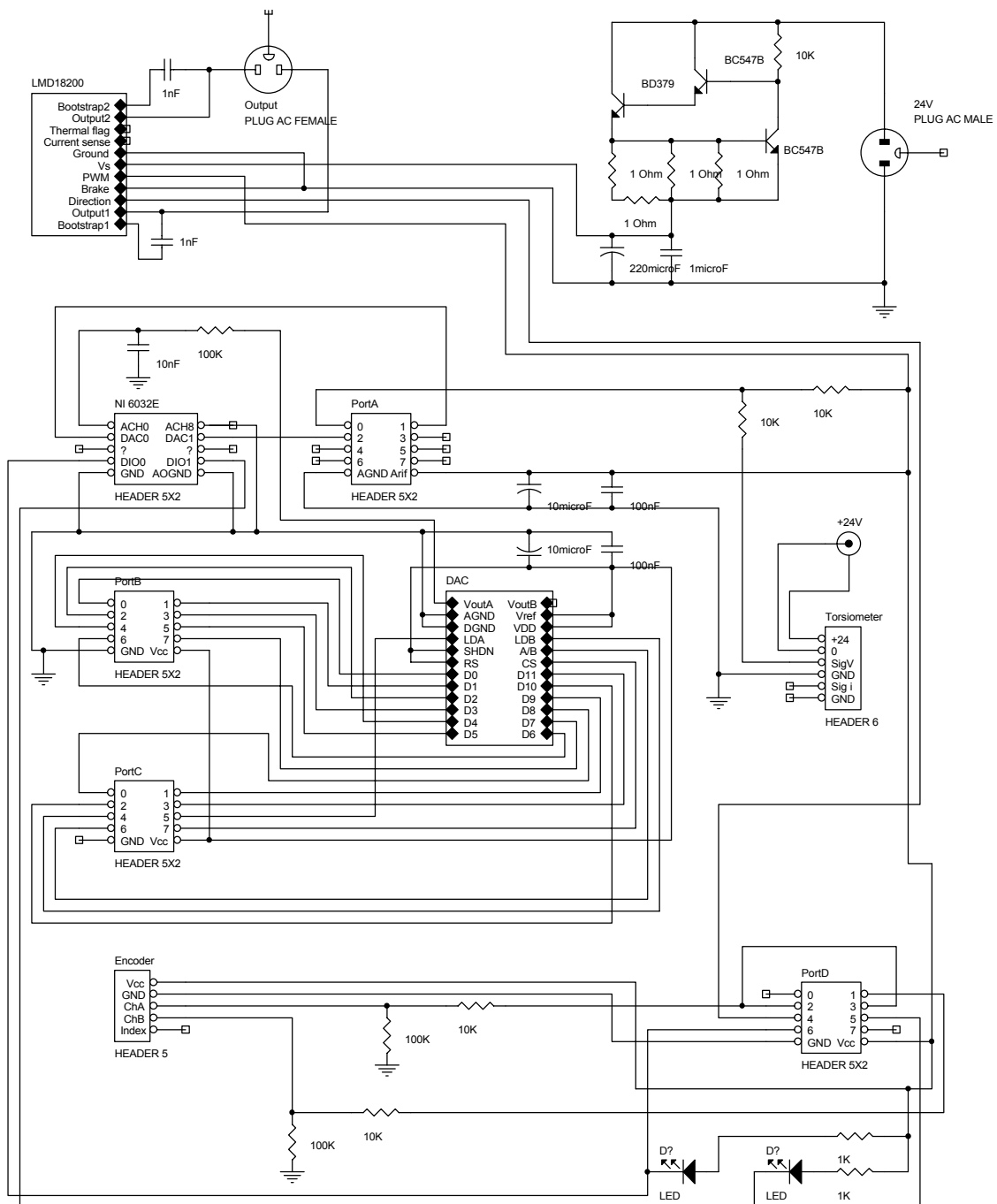


Figura 43: schema elettrico della scheda di interfaccia

2.3.2 Il microcontrollore Atmel AT90S4434

L'Atmel AT90S4434 è un microcontrollore ad 8 bit della serie AVR con architettura RISC. Si riportano le caratteristiche fondamentali interessate in questa applicazione:

- Set di istruzioni formato da 118 operazioni (molte eseguibili in un ciclo di clock)
- 32 registri¹ di lavoro ad 8 bit
- 4 kbyte di memoria programmabile Flash (possibilità di riscriverla 1'000 volte)
- 256 byte di memoria EEPROM (riscrivibile per 100'000 volte)
- 256 byte di memoria di lavoro SRAM
- Convertitore digitale-analogico a 10 bit ed 8 canali
- 2 timer/contatori ad 8 bit programmabili
- 1 timer/contatore a 16 bit programmabile, con funzione di 8, 9 o 10 bit PWM
- 2 interrupt esterni
- 32 linee di input/output programmabili (4 porte a 8 bit)
- Package: 40 pin PDIP
- Tensione operativa: $V_{CC} = 4.0-6.0$ V
- Frequenza di clock: 0-8 MHz (sorgente interna a 8 MHz)

La programmazione, in linguaggio assembler, è stata effettuata avvalendosi dell'ambiente di sviluppo *AVR Studio 3.21*, reperibile liberamente dal sito dell'Atmel (<http://www.atmel.com>). Questo programma comprende un editor integrato, un debugger ed un compilatore assembler, per cui si riesce, senza utilizzare programmi diversi, a scrivere il codice, a correggerlo e ad ottenere il file compilato.

Per trasferire il codice sulla memoria Flash del microcontrollore sono stati utilizzati una schedina di programmazione (Atmel STK200) connessa al PC con la porta seriale, ed il programma Atmel *AVR ISP Version 2.70 (beta1)*. Questa scheda, molto economica, è la più semplice in commercio e permette di eseguire solo le operazioni fondamentali sulla memoria flash del microcontrollore (lettura, cancellazione, scrittura), senza consentire un debug in real-time. A questo scopo sono in commercio altre schede, ma a costi ben maggiori.

Inoltre, è stata riscontrata l'incompatibilità del programma *AVR ISP* con il sistema operativo *Windows 2000*: non è stata rilasciata una versione più aggiornata perché le nuove schede commercializzate hanno abbandonato la porta seriale a favore della porta USB. Di fatto questo ha causato la necessità di utilizzare il sistema operativo *Windows 98*.

Prima di passare alla descrizione delle routine fondamentali implementate, si ritengono necessarie alcune considerazioni generali.

Il microcontrollore è stato impostato per lavorare con la frequenza interna di 8 MHz, che equivale ad un tempo di clock:

$$CK = \frac{1}{f} = \frac{1}{8 \cdot 10^6} = 125 \text{ ns}$$

Le istruzioni di calcolo non comprendono la moltiplicazione ed avvengono tra numeri interi ad 8 bit, pertanto si è deciso di utilizzare un'aritmetica a 16 bit a virgola fissa. Questa esigenza ha

¹ I 32 registri di lavoro hanno delle peculiarità: i primi 16 (R0-R15) non supportano le istruzioni SBCI, SUBI, CPI, ANDI, ORI, LDI e gli ultimi 6 registri vengono utilizzati per gli indirizzamenti in memoria (R26-R31). Inoltre i registri R24, R26, R28, R30 sono gli unici che supportano l'istruzione ADIW per sommare direttamente una costante ad una variabile a 16 bit.

condizionato l'implementazione di tutte le operazioni, riviste in modo da sfruttare al meglio la risoluzione disponibile.

In fig. 44 si rappresenta lo schema principale del programma del microcontrollore, in cui, per eseguire le routine di calcolo ad intervalli di tempo regolari, è stato inserito un ciclo di loop controllato dal timer.

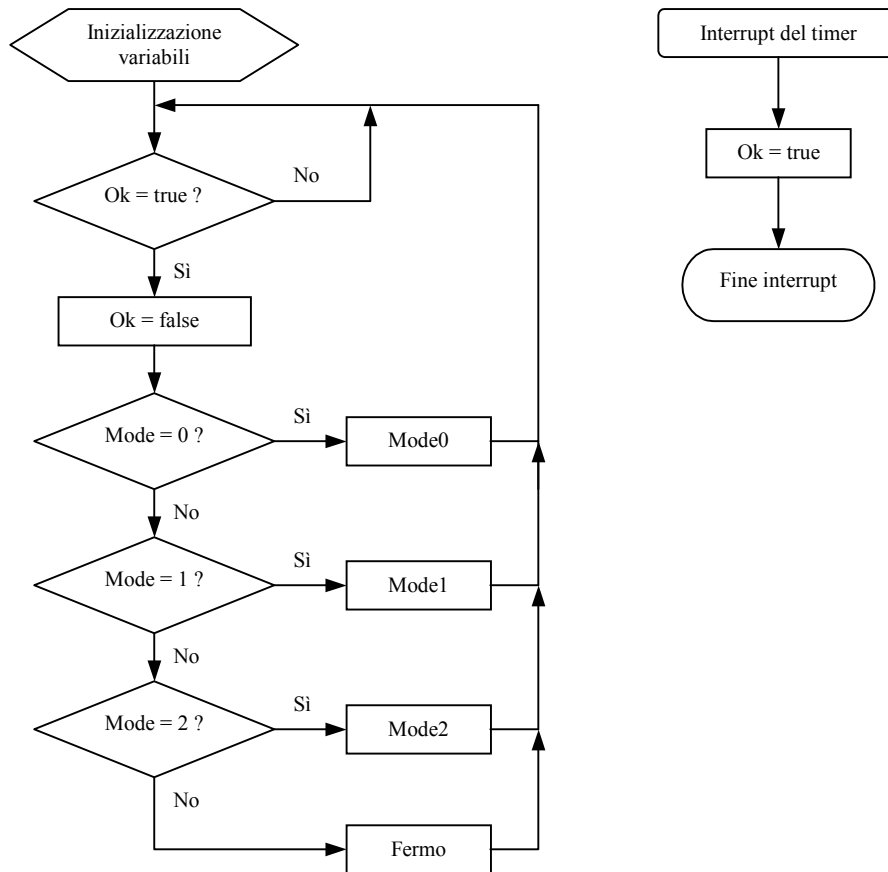


Figura 44: struttura principale del programma di controllo

La modalità di funzionamento viene letta dai pin 5 e 6 della porta D (PD5 e PD6) e determina l'esecuzione della routine corrispondente; nel caso sia stata impostata una modalità non prevista, viene richiamata una routine che toglie l'alimentazione al motore elettrico.

2.3.2.1 Lettura dell'encoder

L'encoder è un sensore di rilevazione ottico nel quale un intero giro è suddiviso in un determinato numero di tacche (zone opache); tra due tacche successive è interposto uno spazio trasparente della stessa larghezza della tacca. Normalmente l'uscita è formata da due canali, sfasati di 90°, ciascuno dei quali assume valore logico opposto in corrispondenza che sia presente la zona opaca oppure quella trasparente.

La lettura della posizione ha comportato l'esigenza di riuscire a stabilire la posizione assoluta del joystick, avendo a disposizione un sensore di posizione relativa come l'encoder. Questo si è tradotto nella necessità, prima, di fissare una posizione di partenza e di riuscire, poi, ad effettuare la lettura senza perdere tacche. Più precisamente, non è sufficiente fare in modo che ogni tacca venga conteggiata, ma deve esserci corrispondenza nella somma anche durante le inversioni del moto.

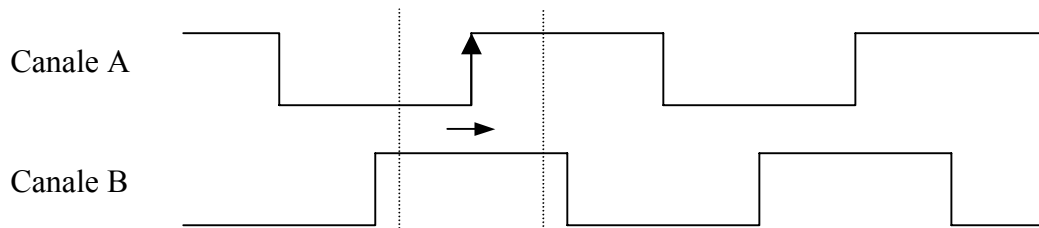


Figura 45: lettura dell'encoder con il gradino di salita

Ad esempio, un metodo usato abitualmente per leggere l'encoder, è quello di osservare quando il segnale del primo canale (A) effettua un gradino in salita. In quel momento si esegue la lettura del secondo canale (B) e in base al fatto che abbia un valore alto oppure basso si esegue un incremento o una sottrazione del conteggio (fig. 45).

Questa tecnica va bene solo fintanto siano richieste misure di velocità, oppure quando si debba misurare la distanza relativa tra due posizioni muovendosi in un solo verso. Infatti, quando ci si trova ad invertire più volte la posizione a cavallo di un gradino del canale A, il conteggio avviene, ogni volta, in un solo senso, mentre la posizione rimane, in realtà, invariata.

Si è reso indispensabile pensare ad un metodo alternativo che ovviasse a questo limite.

Il problema è stato risolto osservando sia quando il segnale del canale A effettua un gradino in salita, sia quando lo effettua in discesa, ed in corrispondenza si valuta lo stato del canale B in modo opposto: se nel gradino in salita il valore logico alto del canale B è associato ad un incremento del conteggio, nel caso di un gradino in discesa lo stesso valore logico alto corrisponde a un decremento (fig. 46).

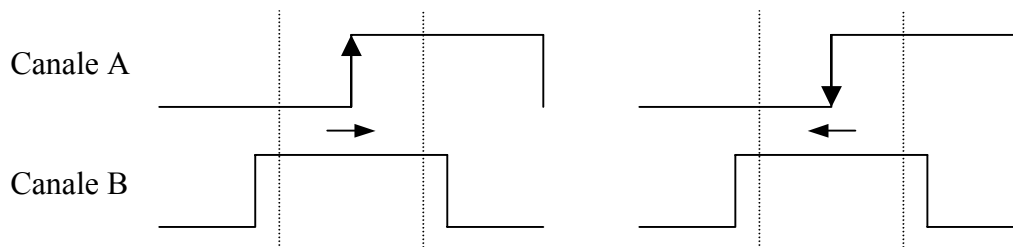


Figura 46: lettura dell'encoder con i gradini di salita e di discesa

Con questa tecnica si ha anche il raddoppio della risoluzione perché, se con la tecnica precedentemente descritta il contatore, in un giro, viene incrementato dello stesso numero delle tacche, in questo modo, contando sia le salite che le discese, viene incrementato di un numero pari al doppio delle tacche.

L'encoder ha 512 tacche per giro che corrispondono, per quanto detto, a 1024 conteggi (count) per giro. Per avere una variabile unsigned del contatore si è fissato lo zero per la posizione orizzontale; in realtà è un'assunzione formale perché il fermo ne impedisce comunque il raggiungimento.

La posizione verticale (90°) ha quindi il valore:

$$P_{90} = 1'024 \cdot \tau \cdot \frac{90}{360} = 1'024 \cdot 65.5 \cdot \frac{90}{360} = 16'768 \text{ count}$$

A questo punto sono stati determinati i valori di fine-corsa in modo da coniugare due esigenze: la prima è che la corsa totale doveva essere di circa 120° come stabilito nelle specifiche, e la seconda è che il valore del conteggio fosse calcolabile dalla moltiplicazione del valore letto sull'ADC con un

numero intero. L'ADC, si ricorda, ha una risoluzione di 10 bit e, quindi, la posizione verticale corrisponde ad un valore di 511.

La prima esigenza si traduce nella differenza, approssimativa, che intercorre per il contatore tra la posizione verticale e quella all'estremità della corsa:

$$P_{60} = P_{90} \cdot \frac{60}{90} = 11'179 \text{ count}$$

Da questo si è trovato il numero intero che deve costituire il fattore di moltiplicazione:

$$m = \frac{P_{60}}{511} = \frac{11'179}{511} = 21.88 \quad \text{da cui} \quad m = 22$$

Si è ricalcolato il numero per raggiungere le estremità della corsa:

$$\Delta P = m \cdot 511 = 22 \cdot 511 = 11'242 \text{ count}$$

Con questo si è potuto determinare l'angolo di corsa effettivo e il valore del contatore agli estremi:

$$\Delta\alpha = 360 \cdot \frac{\Delta P}{\tau \cdot 1'024} = 360 \cdot \frac{11'242}{65.5 \cdot 1'024} = 60.34^\circ$$

Il contatore assume i valori:

$$P_{\text{inf}} = P_{90} - \Delta P = 16'768 - 11'242 = 5'526 \text{ count}$$

$$P_{\text{sup}} = P_{90} + \Delta P = 16'768 + 11'242 = 28'010 \text{ count}$$

In fig. 47 si riporta la rappresentazione schematica delle posizioni principali nelle tre coordinate: gradi, ADC, contatore dell'encoder.

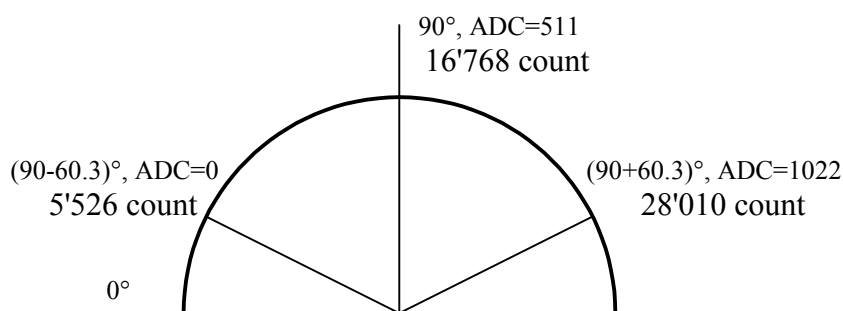


Figura 47: posizioni caratteristiche della leva del joystick nei tre sistemi di misura: gradi, ADC, count

Per l'implementazione, il segnale del canale A è stato collegato ai due pin degli interrupt esterni (PD2 e PD3) abilitando, nel codice di programma, il primo (INT0) in salita ed il secondo (INT1) in discesa. Il canale B è stato collegato al pin PD1, settato come input.

Nella fig. 48 si riporta lo schema a blocchi delle operazioni eseguite dalle routine di gestione dell'encoder.

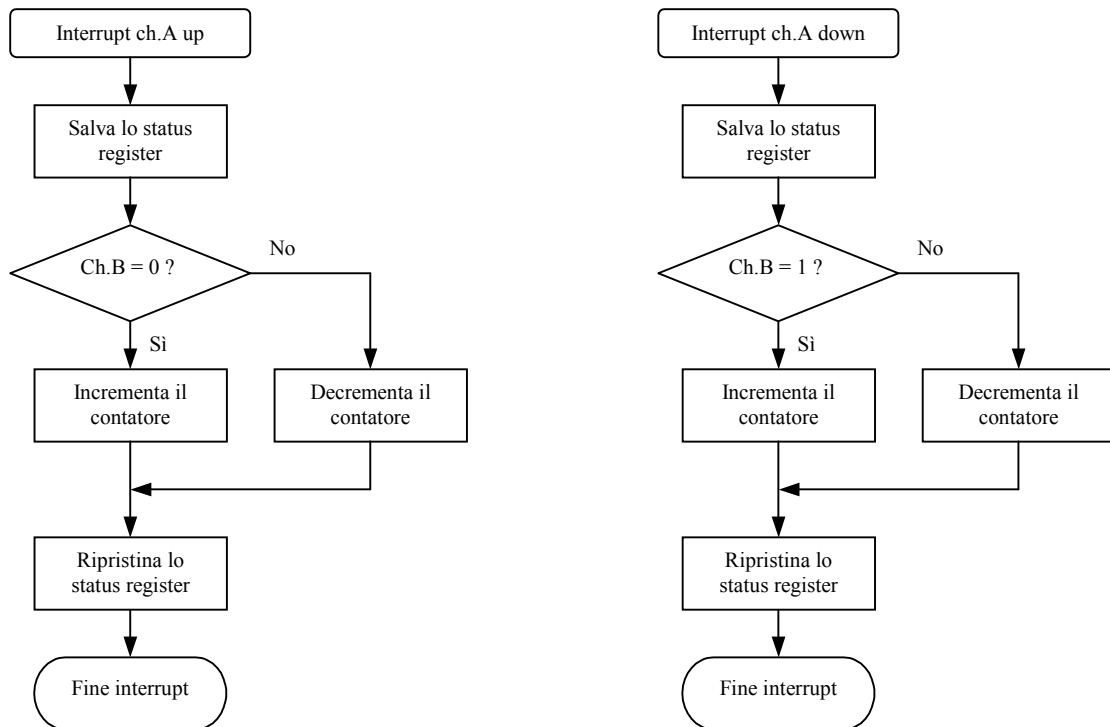


Figura 48: routine di gestione dell'encoder

Per non perdere conteggi è altresì necessario che le routine di interrupt vengano eseguite prima di una ulteriore chiamata, la quale non verrebbe rilevata. Si è allora eseguito il calcolo del tempo che intercorre tra due chiamate successive, quando il motore gira alla massima velocità.

Nota la velocità massima all'uscita del motoriduttore ($\omega_j = 7.4 \text{ rad/s}$), la si riporta all'albero del motore:

$$\omega_m = \omega_j \cdot \tau = 7.4 \cdot 65.5 = 484.7 \text{ rad/s}$$

In termini di frequenza

$$f_m = \frac{\omega_m}{2 \cdot \pi} = \frac{484.7}{2 \cdot \pi} = 77.18 \text{ Hz}$$

Sapendo che in un giro vengono effettuate 1'024 chiamate agli interrupt, si calcola la frequenza di queste:

$$f_{\text{int}} = f_m \cdot 1'024 = 79'032 \text{ Hz}$$

Ovvero il tempo minimo tra due chiamate successive è:

$$t_{\text{int}} = \frac{1}{f_{\text{int}}} = 12'653 \text{ ns}$$

il quale corrisponde a circa 101 cicli di clock.

Questo è un tempo accettabile per eseguire le istruzioni necessarie a leggere il canale B e ad aggiornare il contatore di conseguenza.

Le routine di interrupt implementate occupano, nel caso peggiore, 17 cicli di clock, con il risultato di avere un'occupazione massima della CPU del 17 %.

Pertanto si è riusciti a gestire l'encoder direttamente con il microcontrollore e senza l'utilizzo di componenti esterni.

Il riferimento fisso per la posizione assoluta viene rilevato al momento dell'accensione del microcontrollore, in cui la leva del joystick deve trovarsi nella posizione centrale e al contatore viene attribuito il valore corrispondente di 16'768 count.

2.3.2.2 Ingressi analogici

I due ingressi analogici sono costituiti dalla posizione dell'attuatore (x_E) e dalla forza del torsiometro (f_h). In più è stato aggiunto il segnale di velocità dell'attuatore, poi non utilizzato, ma disponibile per uno sviluppo futuro: solo per questo motivo sono stati attivati 3 canali.

Il convertitore analogico-digitale (ADC, interno al microcontrollore) è stato impostato in modo da richiamare una routine di interrupt non appena terminata la conversione, la quale provvede a salvare il dato nella ram, ad impostare il canale successivo e ad iniziare la nuova conversione (fig. 49).

È stata utilizzata la modalità a singola conversione perché nel datasheet del microcontrollore è riportato un bug nella modalità free-run, quando si debba usare più di un canale.



Figura 49: routine di interrupt per il campionamento dei segnali

L'ADC deve lavorare ad una frequenza compresa tra 50 kHz e 200 kHz. A questo scopo è presente un divisore sulla frequenza di clock principale (prescaler), che è stato impostato su un valore di 64. La frequenza di lavoro dell'ADC è pertanto:

$$f_{\text{ADC}} = \frac{f_{\text{micro}}}{\text{prescaler}} = \frac{8 \cdot 10^6}{64} = 125 \text{ kHz}$$

Ogni conversione impiega 14 cicli per essere eseguita; essendoci la scansione su 3 canali, le variabili sono aggiornate con la frequenza:

$$f_{\text{var}} = \frac{f_{\text{ADC}}}{n_{\text{cicli}} \cdot n_{\text{ch}}} = \frac{125 \cdot 10^3}{14 \cdot 3} \cong 3 \text{ kHz}$$

Nelle specifiche è riportata un'incertezza tipica dell'ADC di 1 bit, e massima di 2 bit su quelli meno significativi; questi valori sono riferiti ad una applicazione del microcontrollore che segua gli accorgimenti per limitare i disturbi elettromagnetici provenienti dalla parte digitale. Invece, nella scheda di programmazione dov'è inserito il chip, questo aspetto è stato sacrificato per altre esigenze e, congiuntamente all'impiego di cavi per la connessione con la scheda di interfaccia, si è registrata un'incertezza che coinvolgeva costantemente gli ultimi 3 bit meno significativi, con picchi di 4 bit. Si è allora provveduto ad effettuare una schermatura dei disturbi ponendo le due schede sopra un piano metallico, al quale sono stati collegati i punti di ground.

In questo modo è stata ripristinata l'incertezza riportata nelle specifiche, ma si sono dovuti comunque adottare altri accorgimenti per aumentare la risoluzione.

Per il segnale proveniente dal torsionometro, si è posto il problema di passare dalla risoluzione dell'ADC (10 bit) alla risoluzione del DAC (12 bit), eliminando i gradini che si sarebbero generati con una semplice moltiplicazione (x4).

La soluzione è stata di calcolare la media di 16 valori letti con l'ADC, così da avere il dato da scrivere sul registro del DAC esterno.

I 16 valori vengono salvati in una tabella *tabtors* aggiornata all'interno della routine di calcolo che, come riportato in seguito, gira ad una frequenza $f_{\text{calcolo}} = 10 \text{ kHz}$.

Ogni singolo valore, *ctors*, viene però aggiornato con una frequenza più bassa di 3 kHz. Questo vuol dire che nella tabella entra il valore di un nuovo campionamento ogni 3.33 cicli della routine di calcolo, e che la tabella contiene in realtà solo $16/3.33 = 4.8$ valori distinti; la presenza di 16 valori mantiene invece l'effetto di passare da un valore all'altro con maggiore gradualità.

Questo risultato è compatibile con le caratteristiche del segnale di forza impressa dall'operatore, che non è in grado di superare i 10 Hz.

L'implementazione di quanto descritto, facendo uso dell'aritmetica intera a 16 bit, si è tradotta nel prevedere una variabile *tors16* nella quale è conservata la somma dei 16 valori registrati in *tabtors*.

Questo è possibile perché *ctors* è un numero a 10 bit e quindi *tors16* è composto da $10+4 = 14$ bit.

La tabella prevede un indice di avanzamento che viene incrementato ad ogni passaggio; quando raggiunge la fine della tabella viene riportato all'inizio in modo da costituire una struttura circolare.

Ad ogni ciclo viene sottratto a *tors16* il valore indicato dall'indice nella tabella, e viene poi aggiunto il nuovo valore di *ctors*, che va anche a rimpiazzare quello preesistente in tabella.

Si riporta una rappresentazione schematica:

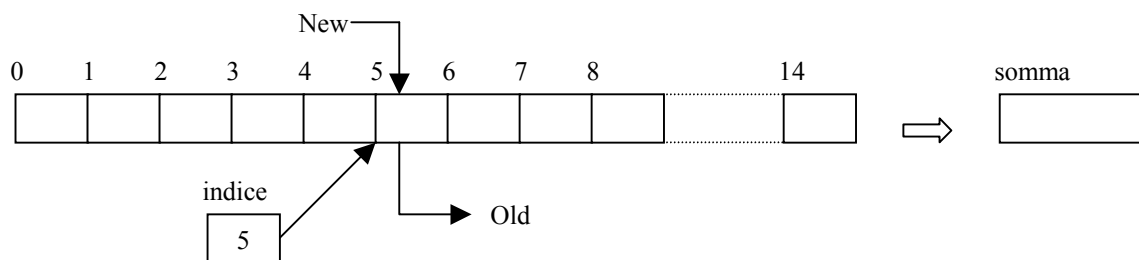


Figura 50: tabella per il calcolo della media mobile

Il dato da mandare al DAC deve essere a 12 bit, e si ottiene agevolmente dividendo la somma *tors16* per 4, che corrisponde a spostare i bit a destra per due volte.

Per quanto riguarda la misura della posizione esterna x_E (corrispondente alla variabile del campionamento *posa*), è stata impiegata una tecnica analoga, adoperando una tabella da 44 valori (*posaqq*) ed un ulteriore accorgimento.

In questo caso la variabile deve essere moltiplicata per una costante ($m = 22$) per costituire il riferimento da raggiungere con il contatore dell'encoder. Nel caso questo valore risulti oscillante per dei disturbi, l'intero sistema di controllo verrebbe compromesso ed il motore, invece di raggiungere stabilmente la posizione impostata, continuerebbe a vibrare attorno a detta posizione.

Anche l'incremento di risoluzione ha assunto maggiore importanza, perché permette il passaggio da una suddivisione della semi-corsa della leva del joystick di 511 valori ad una di 11'242 valori.

Nel primo caso, cioè, si sarebbe avuta la seguente discretizzazione espressa in gradi:

$$d\alpha = \frac{\Delta\alpha}{511} = \frac{60.34}{511} = 0.12^\circ$$

che, riportato ad una misura lineare sull'arco descritto dalla mano, sarebbe stato:

$$ds = d\alpha \cdot \frac{2 \cdot \pi}{360} \cdot b = 0.12 \cdot \frac{2 \cdot \pi}{360} \cdot 150 = 0.314 \text{ mm}$$

il quale costituisce una misura fastidiosamente percettibile sulla leva.

Un valore di 22 volte più piccolo non rappresenta invece una misura facilmente rilevabile dalla sensibilità umana.

L'aggiornamento della tabella *tabposa* e della somma *posaqq* viene effettuato ogni 6 cicli di calcolo, ovvero con la frequenza:

$$f_{\text{posa}} = \frac{f_{\text{calcolo}}}{6} = \frac{10 \cdot 10^3}{6} = 1.67 \text{ kHz}$$

Questa è una frequenza inferiore all'aggiornamento delle variabili ($f_{\text{var}} = 3 \text{ kHz}$), perciò viene letto ogni volta un nuovo valore di *posa*.

La necessità di ottenere un andamento regolare del riferimento ha assunto un'importanza fondamentale per il corretto funzionamento dell'intero sistema, e ha portato ad avvalersi di una considerazione molto semplice: conoscendo la velocità massima con cui può variare la grandezza esaminata, è possibile limitarne la variazione a questo valore.

Invece di prendere il valore di *posa* tale e quale per inserirlo in tabella e sommarlo a *posaqq*, viene prima elaborato nel seguente modo: si mantiene memoria del valore precedentemente inserito in tabella (*oldposa*), e si esegue il confronto con il nuovo valore; se quello nuovo è maggiore, allora si attribuisce a *posa* il valore di quello vecchio più uno, se è minore gli si attribuisce il valore di quello vecchio meno uno.

I valori registrati in tabella possono avere una differenza massima con quelli attigui di una unità, e pertanto la somma può avere la variazione massima in un ciclo di 44 unità.

Si calcola, allora, la massima variazione permessa alla somma in un secondo:

$$\Delta\text{posaqq} = 44 \cdot f_{\text{posa}} = 44 \cdot 1'670 = 73'480 \text{ unità/s}$$

Le unità dell'encoder (count) si ricavano dalla divisione per due:

$$\Delta\text{posa22} = \frac{\Delta\text{posaqq}}{2} = 36'740 \text{ count/s}$$

Sapendo che l'intera corsa è composta da $2 \cdot \Delta P = 22'484$ count, si ricava che può essere percorsa nel tempo di:

$$t_{\min} = \frac{2 \cdot \Delta P}{\Delta\text{posa22}} = \frac{22'484}{36'740} = 0.62 \text{ s}$$

Questo valore è compatibile con le specifiche di progetto che prevedono un tempo minimo di un secondo per percorrere l'intera corsa.

2.3.2.3 Comunicazione con il DAC

La connessione elettrica tra il microcontrollore ed il DAC è già stata illustrata nel capitolo 2.3.1.3, qui si vuole chiarire l'implementazione software della comunicazione.

Si riporta lo schema dell'associazione delle porte B e C con il DAC:

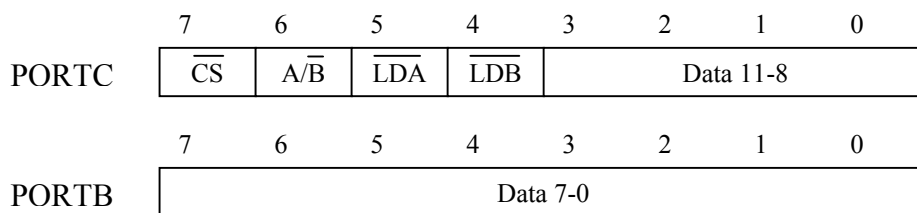


Figura 51: funzioni delle porte B e C nel controllo dell'ADC

Le operazioni che vengono eseguite sono, in sequenza:

- Selezione del chip e del registro INPUTA: $\overline{\text{CS}} = 0, \text{A}/\overline{\text{B}} = 1$
- Scrittura della parte bassa del dato su INPUTA: Data 7-0
- Scrittura della parte alta del dato su INPUTA: Data 11-8
- Trasferimento del registro INPUTA sul registro DACA: $\overline{\text{LDA}} = 0$
- Deselezione del chip: $\overline{\text{CS}} = 1$
- Chiusura del trasferimento del registro: $\overline{\text{LDA}} = 1$

Per cambiare il valore di un solo bit all'interno di un byte è stata utilizzata una maschera: nel caso si voglia portarlo a zero la maschera deve essere combinata con l'operatore logico AND, mentre per portarlo ad 1 si deve utilizzare l'OR.

2.3.2.4 Implementazione del controllo del motore

Il microcontrollore deve provvedere a far raggiungere alla leva del joystick la posizione di riferimento, acquisita con il segnale analogico x_E . Si è già visto, nella parte riguardante gli ingressi analogici, il processo per campionare tale grandezza e come viene convertita in conteggi dell'encoder.

Il microcontrollore comanda direttamente il ponte a H con un segnale PWM, che viene prodotto dal timer/counter 1 e può essere settato con una risoluzione di 8, 9 o 10 bit.

I registri del contatore sono a 16 bit (TCNT1H e TCNT1L) e, ad ogni ciclo di clock, vengono incrementati oppure decrementati di una unità per formare una rampa di salita seguita da una di discesa, con un valore all'apice pari alla risoluzione impostata per il PWM.

Il valore del contatore viene confrontato con i registri OCR1AH e OCR1AL nei quali viene posta la misura dell'ampiezza desiderata (questa variabile è stata chiamata *E256*).

Pertanto, la frequenza del segnale PWM è direttamente collegata alla sua risoluzione e vale:

Risoluzione PWM	Frequenza PWM
8 bit	$f/510$
9 bit	$f/1022$
10 bit	$f/2046$

La risoluzione del PWM è stata impostata ad 8 bit per avere la frequenza più alta possibile con questo microcontrollore, che risulta essere:

$$f_{\text{PWM}} = \frac{f}{510} = \frac{8\text{MHz}}{510} = 15.7 \text{ kHz}$$

Questo valore non è al di sopra della frequenza udibile, e comporta la percezione di un leggero fischio; per quanto riguarda la funzionalità risulta, invece, sufficiente.

La tensione ai capi del motore dipende dalla percentuale, in un ciclo, in cui il PWM è alto, che è a sua volta controllata dalla variabile di programma *E256* (8 bit).

E256 (8 bit)	PWM (%)	E (V)
0	0	-24
127	50	0
255	100	+24

Quello che verrà illustrato in questo paragrafo è il modello, in anello aperto, per controllare la coppia del motore, la definizione del sistema di controllo, ed il procedimento per implementarlo nell'aritmetica intera a 16 bit del microcontrollore.

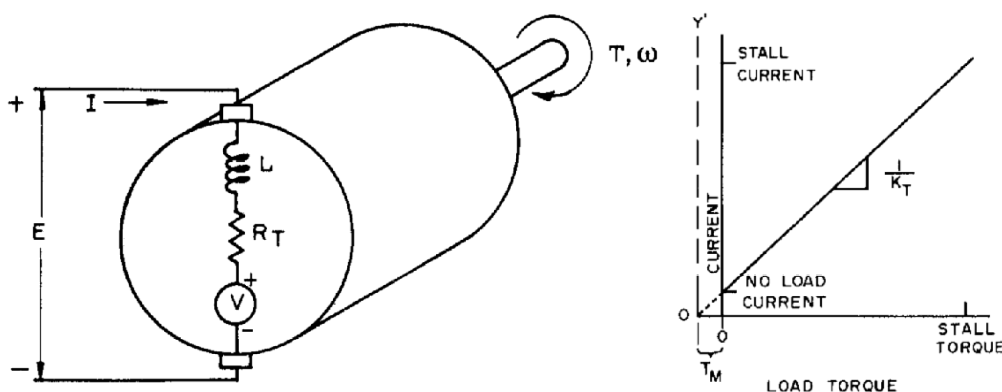


Figura 52: modello di un motore a corrente continua e relativo diagramma coppia-corrente

Si comincia col presentare il modello del motore che è costituito da un'induttanza, una resistenza ed una forza contro elettromotrice generata dalla rotazione dell'albero. Questi parametri sono legati dall'equazione:

$$E = I \cdot R_T + K_E \cdot \omega + L \frac{dI}{dt}$$

In questa analisi il contributo dell'induttanza è ininfluente e sarà pertanto trascurato, ottenendo la relazione:

$$E = I \cdot R_T + K_E \cdot \omega \quad (1)$$

Quando il segnale PWM ha un'ampiezza del 50 %, la tensione E risulta nulla, e l'eventuale velocità angolare ω determina la comparsa di una corrente indotta:

$$\frac{I_{\text{ind}}}{\omega} = \frac{K_E}{R_T} = \frac{45.8 \cdot 10^{-3}}{2.49} = 0.0184 \text{ A/(rad/s)}$$

Siccome la coppia del motore è legata alla corrente, si ha che la rotazione dell'albero motore viene contrastata da una coppia di verso contrario; allora è possibile ricavare la costante caratteristica che lega coppia e velocità, e rappresenta, pertanto, una viscosità:

$$K_D = \frac{I_{\text{ind}}}{\omega} \cdot K_T = 0.0184 \cdot 45.8 \cdot 10^{-3} = 8.42 \cdot 10^{-4} \text{ N} \cdot \text{m}/(\text{rad/s})$$

Questo valore è quello riportato a catalogo e costituisce, di fatto, il termine di viscosità dell'intero sistema.

Per ragioni che saranno chiarite in seguito, si è pensato di compensare parzialmente questa viscosità applicando una tensione proporzionale alla velocità di rotazione: nella variabile *E256* è stato previsto un contributo denominato *Etw*.

La velocità di rotazione viene misurata dal microcontrollore in termini di differenza del contatore dell'encoder tra due cicli di calcolo successivi; la variabile differenza è stata chiamata *deltatacche*.

L'espressione di compensazione usata è:

$$Etw = 4 \cdot \text{deltatacche}$$

Per riportare questa relazione alle unità di misura del sistema fisico (N, m, rad, s) bisogna utilizzare il fattore di conversione:

$$K_{SI} = \frac{1024}{2 \cdot \pi} \cdot \frac{1}{\frac{128}{24} \cdot f_{\text{calcolo}}} = 3.06 \cdot 10^{-3} \text{ V/(rad/s)}$$

La tensione di compensazione risulta essere:

$$\frac{V_{\text{comp}}}{\omega} = 4 \cdot K_{SI} = 4 \cdot 3.06 \cdot 10^{-3} = 1.22 \cdot 10^{-2} \text{ V/(rad/s)}$$

Si ricalcola la corrente indotta dalla velocità, dopo aver adottato questo artificio:

$$\left(\frac{I_{\text{ind}}}{\omega} \right)' = \frac{K_E - \frac{V_{\text{comp}}}{\omega}}{R_T} = \frac{45.8 \cdot 10^{-3} - 1.22 \cdot 10^{-2}}{2.49} = 1.35 \cdot 10^{-2} \text{ A/(rad/s)}$$

Si ottiene la viscosità fittizia:

$$K'_D = \left(\frac{I_{\text{ind}}}{\omega} \right)' \cdot K_T = 1.35 \cdot 10^{-2} \cdot 45.8 \cdot 10^{-3} = 6.18 \cdot 10^{-4} \text{ N} \cdot \text{m}/(\text{rad}/\text{s})$$

Questo valore viene riportato all'albero del joystick:

$$c_j = K'_D \cdot \tau^2 = 6.18 \cdot 10^{-4} \cdot 65.5^2 = 2.65 \text{ N} \cdot \text{m}/(\text{rad}/\text{s})$$

Anche se è una viscosità angolare, per generalità di rappresentazione, il termine è stato denominato come una grandezza lineare.

Lo stesso è stato fatto anche per il momento d'inerzia, denotato con m_j :

$$m_j = J_T = 3.15 \cdot 10^{-2} \text{ kg} \cdot \text{m}^2$$

Ritornando al modello del motore, la corrente può essere espressa tramite la costante di coppia K_T , in cui si considerino separatamente la coppia effettivamente trasmessa al carico T_L e quella persa internamente T_M . L'espressione (1) diventa:

$$E = \left(\frac{T_L + T_M}{K_T} \cdot R_T \right) + K_E \cdot \omega \quad (2)$$

La coppia persa internamente si può desumere dalla corrente a vuoto I_{NL} :

$$T_M = I_{NL} \cdot K_T = 0.16 \cdot 45.8 \cdot 10^{-3} = 7.3 \cdot 10^{-3} \text{ N} \cdot \text{m}$$

Anche se rappresenta circa il 10 % della coppia massima ($T_{\text{Calb}} = 67.1 \cdot 10^{-3} \text{ N} \cdot \text{m}$), T_M viene trascurata in questa analisi, perché è un termine dissipativo che aiuta la stabilità.

Si riscrive l'equazione (2) con questa semplificazione e riportando coppia e velocità angolare all'albero del joystick:

$$E = \frac{T_{L(\text{joy})}}{\tau \cdot \eta_\tau} \cdot \frac{R_T}{K_T} + K_E \cdot \tau \cdot \omega_{\text{joy}} = \frac{T_{L(\text{joy})}}{65.5 \cdot 0.77} \cdot \frac{2.49}{45.8 \cdot 10^{-3}} + 45.8 \cdot 10^{-3} \cdot 65.5 \cdot \omega_{\text{joy}}$$

da cui si ricava:

$$E = 1.08 \cdot T_{L(\text{joy})} + 3 \cdot \omega_{\text{joy}} \quad (3)$$

La tensione applicata ai capi del motore è formata da due termini che contribuiscono, l'uno, alla coppia sul carico, e l'altro alla velocità di rotazione.

Il modello è stato definito in modo da considerare il primo termine come la coppia che deve impartire il controllo per governare la posizione del joystick, ed il secondo, parzialmente compensato, come la viscosità propria del sistema.

Con questa assunzione si procede alla definizione del sistema di controllo in cui massa (m_j) e viscosità fittizia (c_j) sono le caratteristiche fisiche, la posizione del joystick (x_j) è la grandezza

controllata, la posizione di riferimento ($k_{\text{pos}} \cdot x_E$) è l'ingresso controllante, e la forza (f_m) è il parametro di intervento (grandezza manipolata). Lo schema a blocchi è riportato in fig. 53.

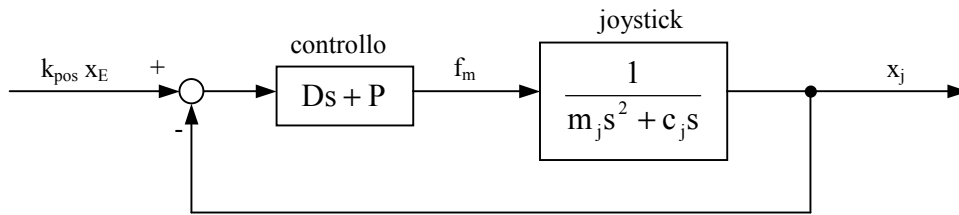


Figura 53: sistema di controllo della posizione del joystick

Si scrive la funzione di trasferimento completa:

$$W(s) = \frac{D \cdot s + P}{m_j \cdot s^2 + (c_j + D) \cdot s + P}$$

La caratteristica della risposta dipende dal polinomio di secondo grado a denominatore. Si procede allora ad analizzare l'equazione:

$$m_j \cdot s^2 + (c_j + D) \cdot s + d = 0$$

che può essere scritta nella forma:

$$s^2 + \frac{c_j + D}{m_j} \cdot s + \frac{P}{m_j} = 0$$

rappresentabile con il polinomio:

$$s^2 + a_1 \cdot s + a_0 = 0 \quad (4)$$

in cui i coefficienti a_1 e a_0 valgono rispettivamente:

$$a_1 = \frac{c_j + D}{m_j} \quad (5)$$

$$a_0 = \frac{P}{m_j} \quad (6)$$

Un sistema di secondo grado, riportato ai parametri del fattore di smorzamento ζ e di pulsazione naturale ω_n , si scrive nella forma:

$$s^2 + 2 \cdot \zeta \cdot \omega_n \cdot s + \omega_n^2 = 0 \quad (7)$$

Confrontando le espressioni (4) e (7), si ha:

$$a_0 = \omega_n^2 \quad (8)$$

$$a_1 = 2 \cdot \zeta \cdot \omega_n \quad (9)$$

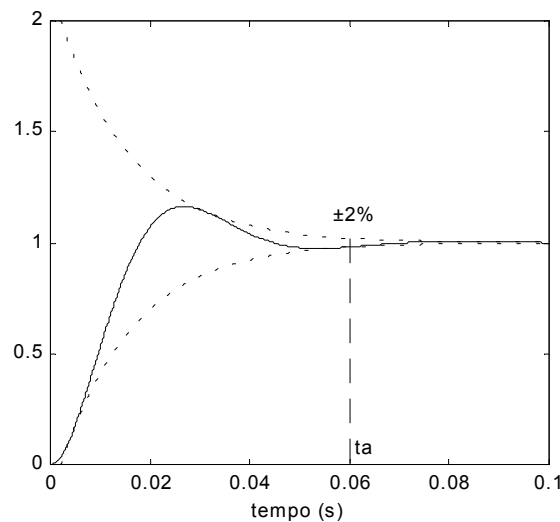


Figura 54: risposta ad un gradino di un sistema del secondo ordine e determinazione del tempo di assestamento

Per definire il sistema di controllo sono stati fissati il fattore di smorzamento ed il tempo di assestamento t_a : quest'ultimo è definito come il tempo impiegato dal sistema, in risposta ad un ingresso a gradino, per rientrare definitivamente all'interno di una fascia compresa tra +2 % e -2 % dell'ingresso ricevuto.

Dunque, sono stati fissati:

$$\zeta = 0.5 \quad (10)$$

$$t_a = 0.06 \text{ s} \quad (11)$$

Dovendosi determinare il polinomio (7), è stato necessario stabilire la pulsazione naturale che deve avere il sistema per rispettare questi due parametri; a questo scopo si è analizzata l'equazione della risposta di un sistema di secondo grado ad un gradino unitario, in cui posizione e velocità all'istante iniziale siano nulle.

$$y(t) = 1 - e^{-\zeta \omega_n t} \cdot \left[\cos(\omega_n \cdot \sqrt{1 - \zeta^2} \cdot t) + \frac{\zeta}{\sqrt{1 - \zeta^2}} \cdot \sin(\omega_n \cdot \sqrt{1 - \zeta^2} \cdot t) \right]$$

In questa espressione si vede che la funzione sinusoidale dell'oscillazione viene moltiplicata per un termine esponenziale che la porta asintoticamente ad annullarsi nel tempo. Quindi, partendo dalla considerazione che la funzione è sempre compresa all'interno di due esponenziali, si può determinare il tempo di assestamento imponendo la condizione che la curva di delimitazione intersechi la fascia contraddistinta dal $\pm 2\%$ del valore di riferimento unitario.

L'espressione delle due esponenziali di delimitazione è:

$$y_{\text{lim}}(t) = 1 \pm \frac{1}{\sqrt{1 - \zeta^2}} \cdot e^{-\zeta \omega_n t}$$

La condizione di intersezione si esprime invece con l'equazione:

$$\frac{1}{\sqrt{1-\zeta^2}} \cdot e^{-\zeta \cdot \omega_n \cdot t_a} = \frac{2}{100} \quad (12)$$

Dalla quale si ricava la pulsazione naturale:

$$\omega_n = -\frac{\ln \frac{2}{100} + \ln \sqrt{1-\zeta^2}}{\zeta \cdot t_a} = \frac{\ln \frac{2}{100} + \ln \sqrt{1-0.5^2}}{0.5 \cdot 0.06} = 135 \text{ rad/s}$$

Dall'eq. (8) è stato calcolato il coefficiente a_0 :

$$a_0 = \omega_n^2 = 135^2 = 18'225$$

Si è determinato il parametro di controllo P dall'eq. (6):

$$P = m_j \cdot a_0 = 3.15 \cdot 10^{-2} \cdot 18'225 = 574 \text{ N} \cdot \text{m} / \text{rad}$$

Questo valore è stato riportato ad uno implementabile nell'aritmetica intera a 16 bit; a tal proposito si è calcolata, dapprima, la costante di trasformazione della posizione dall'unità di misura in radianti, all'unità di conteggio dell'encoder:

$$K_{\text{count}} = \frac{1'024}{2 \cdot \pi} \cdot \tau = \frac{1'024}{2 \cdot \pi} \cdot 65.5 = 10'680 \text{ count/rad}$$

Si è calcolato poi, riprendendo l'espressione (3), la costante che lega la tensione applicata (espressa nel formato a 8 bit della variabile del PWM) alla coppia T_L :

$$K_{T256} = 1.08 \cdot \frac{128}{24} = 5.71 \frac{1}{\text{N} \cdot \text{m}}$$

Con questa si è espresso P con il sistema di misura adottato nei calcoli del microcontrollore:

$$P'_{256} = P \cdot \frac{K_{T256}}{K_{\text{count}}} = 574 \cdot \frac{5.71}{10'680} = 0.307$$

L'aritmetica deve essere intera a 16 bit, per cui una costante si può esprimere come numero intero oppure frazionario; in questo caso, per velocizzare i calcoli è stato posto:

$$P'_{256} = 0.307 = \frac{1}{3.25} \qquad P_{256} = \frac{1}{4}$$

Dopo questa assunzione è stato ricalcolato il corrispondente valore nel sistema di misura fisico:

$$P = P_{256} \cdot \frac{K_{\text{count}}}{K_{T256}} = \frac{1}{4} \cdot \frac{10'680}{5.71} = 468 \text{ N} \cdot \text{m} / \text{rad}$$

Con l'eq. (6) si è trovato l'effettivo valore di a_0 :

$$a_0 = \frac{P}{m_j} = \frac{468}{3.15 \cdot 10^{-2}} = 14'825$$

Con l'eq. (8) si è calcolata la nuova pulsazione naturale:

$$\omega_n = \sqrt{a_0} = \sqrt{14'825} = 122 \text{ rad/s}$$

Con l'eq. (9) si è proceduto calcolando il coefficiente a_1 :

$$a_1 = 2 \cdot \zeta \cdot \omega_n = 2 \cdot 0.5 \cdot 122 = 122$$

Dall'eq. (5) si è calcolato, infine, il valore da attribuire al parametro di controllo D:

$$D = m_j \cdot a_1 - c_j = 3.15 \cdot 10^{-2} \cdot 122 - 2.65 = 1.19 \text{ N} \cdot \text{m}/(\text{rad}/\text{s})$$

Analogamente a quanto fatto prima per il parametro P, si è riportato D al sistema di misura di calcolo, con la differenza che in questo caso deve essere eseguita anche la conversione della grandezza temporale:

$$D'_{256} = D \cdot \frac{f_{\text{calcolo}} \cdot K_{T256}}{K_{\text{count}}} = 1.19 \cdot \frac{10'000 \cdot 5.71}{10'680} = 6.36$$

È stato quindi ricondotto al valore intero:

$$D_{256} = 6$$

Determinati i parametri di controllo per l'implementazione sul microcontrollore, è stata eseguita la verifica delle caratteristiche effettivamente attribuite, dopo le approssimazioni di adattamento al sistema di calcolo.

Si è riportato D_{256} al sistema fisico:

$$D = D_{256} \cdot \frac{K_{\text{count}}}{f_{\text{calcolo}} \cdot K_{T256}} = 6 \cdot \frac{10'680}{10'000 \cdot 5.71} = 1.12 \text{ N} \cdot \text{m}/(\text{rad}/\text{s})$$

Con l'eq. (5) si è ricalcolato il coefficiente a_1 :

$$a_1 = \frac{c_j + D}{m_j} = \frac{2.65 + 1.12}{3.15 \cdot 10^{-2}} = 120$$

Con l'eq. (9) si è trovato l'effettivo valore del fattore di smorzamento:

$$\zeta = \frac{a_1}{2 \cdot \omega_n} = \frac{120}{2 \cdot 122} = 0.49$$

Ed infine, con l'eq. (12), è stato calcolato il tempo di assestamento:

$$t_a = -\frac{\ln \frac{2}{100} + \ln \sqrt{1 - \zeta^2}}{\zeta \cdot \omega_n} = -\frac{\ln \frac{2}{100} + \ln \sqrt{1 - 0.49^2}}{0.49 \cdot 122} = 0.0677 \text{ s}$$

Questi due valori costituiscono una caratteristica della risposta accettabile.

Per quanto detto finora, la variabile che comanda il PWM assume la forma:

$$E_{256} = P_{256} \cdot \text{Err} + D_{256} \cdot \text{deltaerr} + 4 \cdot \text{deltatacche}$$

$$E_{256} = \frac{\text{Err}}{4} + 6 \cdot \text{deltaerr} + 4 \cdot \text{deltatacche}$$

Un accorgimento per evitare il superamento della coppia massima del motore interviene sui primi due termini, che costituiscono il controllo della coppia vero e proprio.

Per questo motivo vengono sommati assieme nella variabile E_{tl} , sulla quale è poi eseguita una sogliatura per farla rientrare nell'intervallo compreso tra -20 e $+20$.

Questo valore corrisponde ad una coppia, con il motore fermo, pari a:

$$T_{\max} = \frac{E_{tl_{\max}}}{K_{T_{256}}} = \frac{20}{5.71} = 3.50 \text{ N} \cdot \text{m}$$

In realtà T_{\max} , oltre alla coppia trasmessa al carico, comprende anche la coppia persa internamente T_M , per cui il valore della coppia in uscita è:

$$T_{L(\text{joy})_{\max}} = T_{\max} - \tau \cdot T_M = 3.50 - 65.5 \cdot 7.3 \cdot 10^{-3} = 3.02 \text{ N} \cdot \text{m}$$

Questo valore risulta inferiore a quello massimo nominale di $3.39 \text{ N} \cdot \text{m}$.

La sogliatura, però, limita anche la velocità massima che può raggiungere il motore.

Come già accennato in precedenza, la presenza del termine di compensazione parziale, consente di pensare ad un'alterazione della forza contro elettromotrice indotta, che assume la dimensione:

$$K'_E = K_E - \frac{V_{\text{comp}}}{\omega} = 45.8 \cdot 10^{-3} - 1.22 \cdot 10^{-2} = 3.36 \cdot 10^{-2} \text{ V}/(\text{rad/s})$$

Riportata all'albero del joystick diventa:

$$K'_{E(\text{joy})} = K'_E \cdot \tau = 3.36 \cdot 10^{-2} \cdot 65.5 = 2.20 \text{ V}/(\text{rad/s})$$

Quella che si vuole calcolare è la velocità massima del joystick in assenza di carico; questo significa che si tratta di una condizione nella quale c'è un minimo assorbimento di corrente, e siccome si è preso un alimentatore non regolato, sul ponte a H non arriva la tensione nominale di 24 V , bensì le misure hanno rivelato una tensione di circa 30 V . La stima della velocità deve essere eseguita, pertanto, con questo valore.

$$\omega_{\max(\text{joy})} = Etl_{\max} \cdot \frac{30}{128} \cdot \frac{1}{K'_{E(\text{joy})}} = 20 \cdot \frac{30}{128} \cdot \frac{1}{2.20} = 2.13 \text{ rad/s}$$

L'intera corsa, espressa in radianti, ha la seguente ampiezza:

$$\alpha_{\text{joy}} = 2 \cdot \Delta\alpha \cdot \frac{\pi}{180} = 2 \cdot 60.34 \cdot \frac{\pi}{180} = 2.11 \text{ rad}$$

Per cui può essere coperta in un tempo pari a:

$$t_{\text{corsa}} = \frac{\alpha_{\text{joy}}}{\omega_{\max(\text{joy})}} = \frac{2.11}{2.13} = 0.99 \text{ s}$$

Questo risultato rispetta la specifica fissata di 1 secondo per coprire l'intera corsa.

L'ultima verifica che si intende eseguire è la determinazione dell'errore sulla posizione quando viene applicata sulla leva la forza massima nominale. Il controllo, infatti, è di tipo proporzionale-derivativo ed abbisogna di un certo errore per poter reagire ai disturbi esterni, in questo caso rappresentati dalla forza esercitata dalla mano sul joystick.

L'errore si desume molto semplicemente dal parametro proporzionale P:

$$d\alpha' = \frac{F_{\text{nom}} \cdot b}{P} = \frac{20 \cdot 0.15}{468} = 6.41 \cdot 10^{-3} \text{ rad}$$

Questo angolo corrisponde ad uno spostamento sulla leva equivalente a:

$$ds' = d\alpha' \cdot b = 6.41 \cdot 10^{-3} \cdot 150 = 0.96 \text{ mm}$$

Tale valore si ritiene accettabile perché non compromette la percezione tattile; inoltre, si nota che il parametro P (468 N·m/rad) costituisce una rigidezza confrontabile con quella fisica del torsiometro (1'128 N·m/rad), al di sopra della quale non è possibile arrivare.

Nella fig. 55 si riporta il diagramma a blocchi della simulazione effettuata con Simulink del controllo implementato sul microcontrollore, completo della discretizzazione eseguita nelle variabili intere a 16 bit.

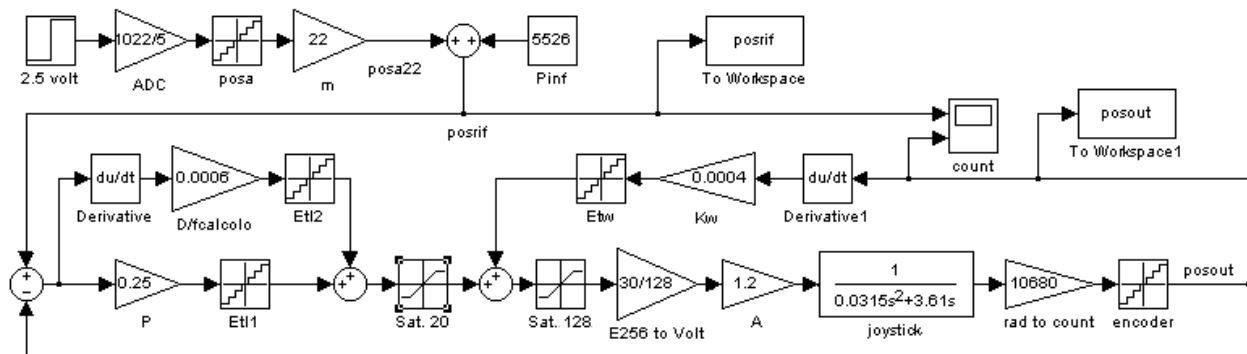


Figura 55: simulazione del controllo implementato sul microcontrollore

In fig. 56 si riportano due grafici relativi alla simulazione con un ingresso a gradino di diversa ampiezza:

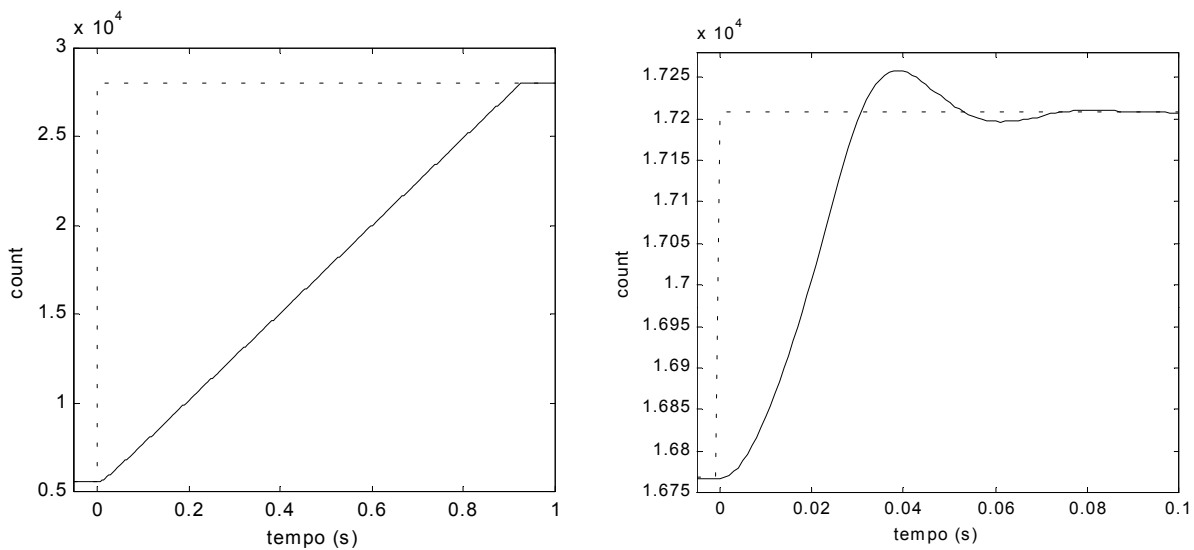


Figura 56: risposta a due gradini di diversa ampiezza: 22'484 e 440 count

Il grafico di sinistra riporta la risposta del sistema, inizialmente posizionato all'estremità inferiore della corsa (pari a 5'526 count), ad un segnale a gradino di 5 V (corrispondente a 28'010 count). L'effetto principale, denotato nella risposta, è la presenza della sogliatura che rende il raggiungimento della posizione pressoché un movimento a velocità costante. Il tempo di salita è prossimo a quello stimato di 0.99 secondi

Nel grafico di destra si è voluta mettere in risalto la risposta del controllo senza l'intervento della sogliatura. Si è pertanto utilizzato un gradino più piccolo di 0.1 V sul segnale di ingresso a partire dalla posizione centrale di 2.5 V; ciò equivale, nel sistema di misura di calcolo, ad un gradino di 440 count a partire dalla posizione di 16'768 count.

La caratteristica segue i valori calcolati per il fattore di smorzamento $\zeta = 0.49$ e per il tempo di assestamento $t_a = 0.0677$ secondi; si registra un overshoot di 50 count, pari all'11 % dell'ampiezza del gradino in ingresso.

In entrambi i casi la simulazione ha confermato la rispondenza alle specifiche, e quindi la validità della conversione del controllo nell'aritmetica intera a 16 bit stabilita per il microcontrollore.

2.3.2.5 Implementazione del comando tradizionale

La necessità di avere un raffronto con un joystick tradizionale, ha portato alla decisione di programmare questa modalità di funzionamento.

Un joystick "classico" ricava la grandezza di comando dalla posizione assunta dalla leva, mentre la grandezza che fornisce sensibilità è la forza. La caratteristica elastica associa univocamente le due grandezze e rappresenta, pertanto, il parametro fondamentale di questo tipo di joystick; inoltre, assieme alla viscosità c_j , definisce completamente il modello assunto.

La viscosità, come già visto, dipende dalle caratteristiche del motore elettrico, ed anche in questo caso, essendo troppo elevata, si è attuata una compensazione parziale.

Nell'implementazione è stata utilizzata la seguente relazione:

$$E_{tw} = 6 \cdot \text{deltatacche}$$

Riportata alle unità di misura del sistema fisico, la tensione di compensazione è pari a:

$$\left(\frac{V_{\text{comp}}}{\omega}\right)^* = 6 \cdot K_{SI} = 6 \cdot 3.06 \cdot 10^{-3} = 1.84 \cdot 10^{-2} \text{ V}/(\text{rad/s})$$

La corrente indotta dalla velocità, vale allora:

$$\left(\frac{I_{\text{ind}}}{\omega}\right)^* = \frac{K_E - \left(\frac{V_{\text{comp}}}{\omega}\right)^*}{R_T} = \frac{45.8 \cdot 10^{-3} - 1.84 \cdot 10^{-2}}{2.49} = 1.10 \cdot 10^{-2} \text{ A}/(\text{rad/s})$$

Si ottiene il fattore di viscosità:

$$K_D^* = \left(\frac{I_{\text{ind}}}{\omega}\right)^* \cdot K_T = 1.10 \cdot 10^{-2} \cdot 45.8 \cdot 10^{-3} = 5.05 \cdot 10^{-4} \text{ N} \cdot \text{m}/(\text{rad/s})$$

Questo valore viene riportato all'albero del joystick attraverso il rapporto di riduzione:

$$c_j^* = K_D^* \cdot \tau^2 = 5.05 \cdot 10^{-4} \cdot 65.5^2 = 2.17 \text{ N} \cdot \text{m}/(\text{rad/s})$$

La definizione della caratteristica elastica ha comportato uno studio preliminare per stabilirne l'andamento più appropriato.

La relazione più semplice, quella lineare, ottempera ai requisiti di proporzionalità e stabilità, ma risulta del tutto inadeguata per il tempo di richiamo: la pendenza, infatti, determina la costante di rigidità e deve essere tarata in funzione della forza massima esercitabile a fine corsa, ma in questo modo, in prossimità della posizione centrale, produce una forza di richiamo troppo esigua. Nella fig. 57 se ne riporta il diagramma, corrispondente a quello di una molla semplice.

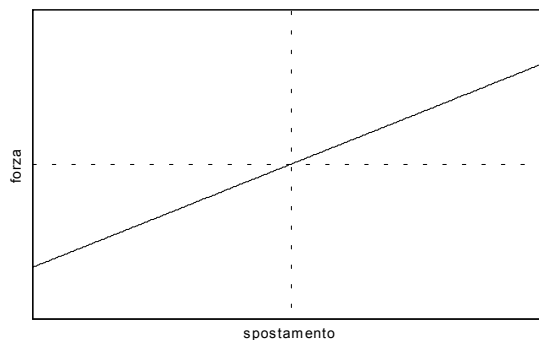


Figura 57: caratteristica elastica di una molla lineare

Il problema si risolve con una caratteristica analoga a quella di due molle precaricate, ciascuna riservata all'azione su una sola semi-corsa. La forza di richiamo minima viene stabilita dall'entità del precarico e la rigidità può essere stabilita in modo indipendente.

Nella fig. 58 si può osservare come, in corrispondenza della posizione centrale, è presente una discontinuità.

In un sistema fisico sarebbe impossibile realizzare una caratteristica con un passaggio netto tra i due valori di precarico a causa dell'imperfezione degli accoppiamenti e dell'elasticità propria di ogni elemento meccanico, ma dovendo definire la caratteristica per essere implementata su un sistema di controllo, questa discontinuità provocherebbe un'oscillazione attorno alla posizione di equilibrio.

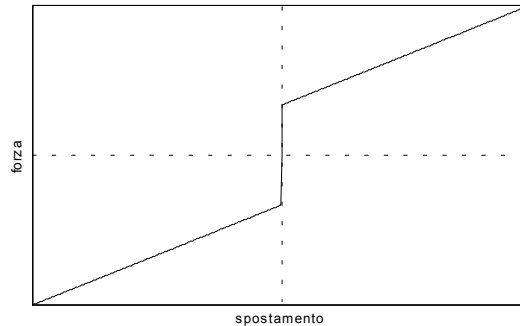


Figura 58: caratteristica elastica di due molle precaricate e collocate in opposizione

È necessario, allora, ripristinare la continuità raccordando i due valori di precarico con una funzione lineare.

Si ottiene il grafico di fig. 59, in cui sono rispettati tutti i requisiti posti dall'applicazione.

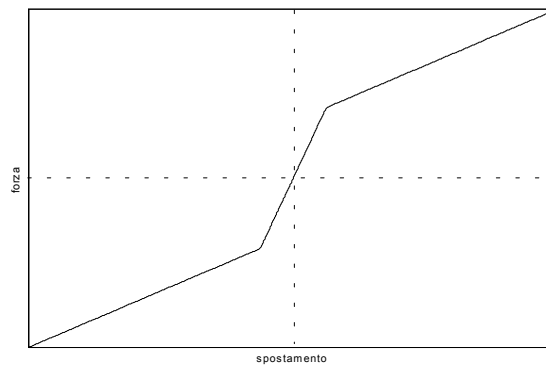


Figura 59: caratteristica elastica con precarico e linea di raccordo

Nel codice di programma si è voluto riprodurre questa caratteristica avendo come dato di input la posizione della leva espressa in conteggi dell'encoder, e l'output rappresentato dalla variabile di comando, ad 8 bit, del PWM.

In assembler la divisione tra due numeri richiede una routine di calcolo molto dispendiosa in termini di tempo, ma la divisione per una potenza di 2 può essere eseguita molto agevolmente traslando a destra i bit della variabile un numero di volte pari all'esponente. Nel caso si lavori con numeri dotati di segno, nel bit più significativo deve essere ricopiato il valore preesistente.

Segue la rappresentazione della divisione per due di un numero 16 bit signed:

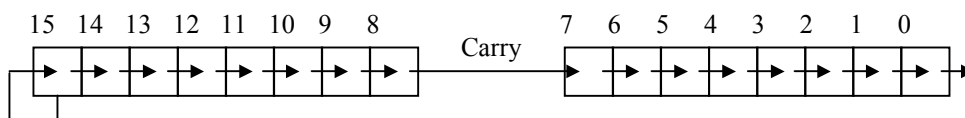


Figura 60: divisione per 2 di un numero con segno

Il risultato che si ottiene è l'intero arrotondato verso meno infinito.

Se si vuole ottenere, invece, il numero arrotondato all'intero più vicino, è sufficiente sommare un valore pari alla metà del divisore prima di effettuare l'operazione.

La variabile di input si chiama *posout* ed identifica la posizione centrale per un valore di 16'768 count (P90). Si calcola *pos0* considerando di dover poi effettuare una divisione per 1'024:

$$\text{pos0} = \text{posout} - 16'768 + 512$$

Da questo valore si ricava la tensione da fornire al motore:

$$dE = \frac{\text{pos0}}{1'024}$$

Viene aggiunto il valore di precarico:

$$\text{per } dE > 0 \quad \text{si ha} \quad dE = dE + 4$$

$$\text{per } dE < 0 \quad \text{si ha} \quad dE = dE - 4$$

Per $dE = 0$ si rientra invece in una zona morta avente l'ampiezza di 1'024 count: è l'intervallo entro il quale la caratteristica deve essere raccordata con una retta a pendenza maggiore.

Si riprende la variabile *pos0* incrementata di 64 count in previsione di una divisione per 128:

$$\text{pos0x} = \text{pos0} + 64$$

Si effettua tale divisione:

$$dEx = \frac{\text{pos0x}}{128}$$

Dopo aver sottratto 4 unità, corrispondenti alla somma precedente ($4 = 512/128$), si ottiene la tensione:

$$dE = dEx - 4$$

Si riporta il diagramma dell'implementazione, nell'intorno della posizione centrale, in cui in ascissa è stato posto il conteggio dell'encoder, ed in ordinata la tensione espressa in unità di controllo del segnale PWM.

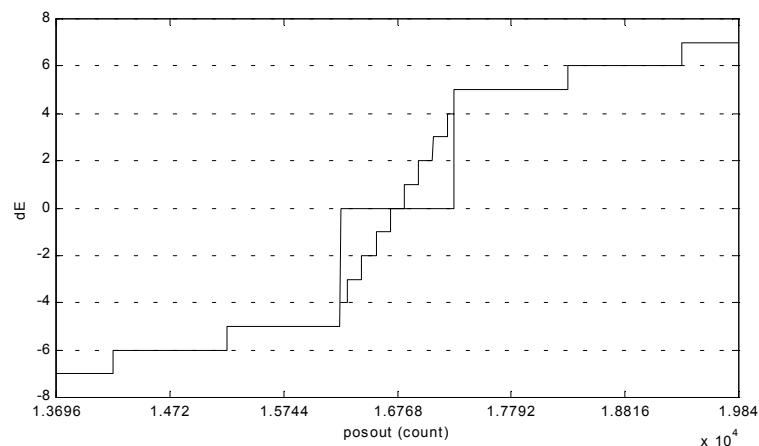


Figura 61: caratteristica elastica implementata nel microcontrollore

La rigidezza è stata stabilita con il fattore di divisione ($2^{10} = 1024$).

Per eseguire la verifica sulla coppia esercitata a fine corsa si calcola il valore di dE corrispondente a $posout = 28'010$:

$$dE = \frac{posout - 16'768 + 512}{1'024} + 4 = \frac{28'010 - 16'768 + 512}{1'024} + 4 = 15$$

Attraverso la costante di trasformazione K_{T256} si ricava la coppia:

$$T_{fc} = \frac{dE}{K_{T256}} = \frac{15}{5.71} = 2.63 \text{ N} \cdot \text{m}$$

Questo è un valore accettabile perché inferiore alla coppia massima nominale del joystick.

Il segnale di output viene ricavato dalla posizione, dividendo $posout$ per 8 in modo da riportarlo all'unità di misura del DAC a 12 bit. Per ripristinare la coincidenza con i valori centrali viene successivamente sottratto il valore di 48:

$$Data_{(DAC)} = \frac{posout}{8} - 48$$

I valori di tensione sul segnale di uscita, in corrispondenza alle due estremità della corsa, sono:

$$V_{inf} = \left(\frac{P_{inf}}{8} - 48 \right) \cdot \frac{5}{4'094} = \left(\frac{5'526}{8} - 48 \right) \cdot \frac{5}{4'094} = 0.78 \text{ V}$$

$$V_{sup} = \left(\frac{P_{sup}}{8} - 48 \right) \cdot \frac{5}{4'094} = \left(\frac{28'010}{8} - 48 \right) \cdot \frac{5}{4'094} = 4.22 \text{ V}$$

2.4 LA SIMULAZIONE DELL'AMBIENTE ESTERNO

Per la verifica sperimentale del joystick costruito, si è presentata la necessità di avere a disposizione un sistema da comandare. Non essendo direttamente disponibile in laboratorio un sistema reale, si è pensato di crearlo attraverso una simulazione, con il vantaggio aggiuntivo di poter effettuare le prove su sistemi diversi, implementati via software.

La scelta è stata di utilizzare lo stesso PC impiegato per la programmazione del microcontrollore, un *Pentium II* a 400 MHz con il sistema operativo *Windows 98*, così da ottenere la massima compattezza del sistema, oltre ad un indubbio risparmio di risorse.

Su questo computer è stata installata la scheda di acquisizione National Instruments PCI-6024E, in modo da avere a disposizione l'entrata e l'uscita analogiche per collegarsi al joystick.

Una valutazione degli obiettivi della simulazione e dell'hardware a disposizione, ha determinato la scelta del linguaggio di programmazione *Visual Basic 6.0*: infatti, non era richiesta una elevata efficienza del codice, mentre l'impiego di un linguaggio ad alto livello e dell'interfaccia grafica hanno permesso di ridurre al minimo i tempi di sviluppo del software.

Nella fig. 62 è riportata la schematizzazione delle funzioni delegate all'ambiente di simulazione: come già accennato nella definizione del sistema, oltre al movimento e all'interazione dell'attuatore con l'ambiente esterno, la simulazione si occupa anche della retroazione sul sistema di comando.

Questa funzione, che nella realtà dovrebbe essere svolta da un sistema di controllo autonomo, è stata inglobata nella simulazione per fornirle la necessaria tempestività di intervento nei confronti delle sollecitazioni esterne, rilevate dal sensore di forza posto sull'attuatore, altrimenti non realizzabile a causa del ritardo di comunicazione presente nella scheda di acquisizione.

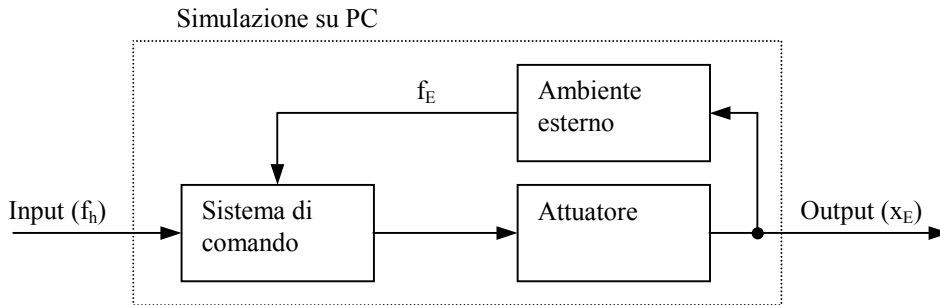


Figura 62: funzioni svolte dalla simulazione su PC

L'ambiente esterno doveva essere in grado di riprodurre le situazioni tipiche incontrabili in un ambiente di lavoro reale.

Si è pensato di rappresentare genericamente queste situazioni con due soli elementi distinti: posto il sistema in verticale, nella parte inferiore è stato simulato un sistema massa-molla-smorzatore a cui ricondurre il contatto con un elemento elastico, e nella parte superiore è stato collocato un carico sostenuto da due appoggi. Quest'ultimo permette di verificare il comportamento nel contatto con un corpo rigido e, una volta sollevato, restituisce una misura della sensibilità nella manovra di carichi sospesi.

Nella fig. 63 è riportata la schematizzazione dell'ambiente riprodotto nella simulazione; dell'attuatore è stata rappresentata solo l'estremità (con un cerchio) perché l'unica parte interessata alla simulazione è quella a contatto con l'ambiente esterno.

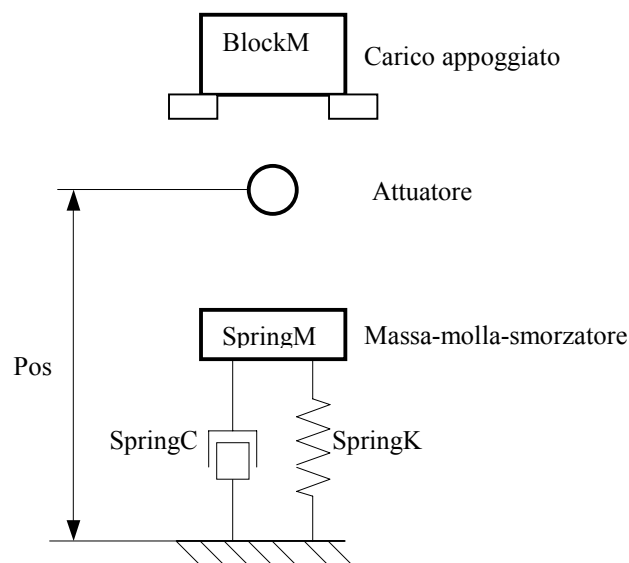


Figura 63: modello dell'ambiente esterno

Per ciascun elemento sono stati definiti i seguenti parametri, impostabili all'interno di un certo range:

Attuatore:

Forza massima sviluppabile:	$F_{max} = 10 - 20'000 \text{ N}$
Velocità massima:	$V_{max} = 0.1 - 1 \text{ m/s}$
Massa (utilizzata nei contatti)	$Mass = 10 - 5'000 \text{ kg}$

Carico appoggiato:

Posizione appoggi:	$BlockX0 = 0.5 - 1 \text{ m}$
Massa:	$BlockM = 10 - 10'000 \text{ kg}$

Sistema massa-molla-smorzatore:

Posizione:	$SpringX = 0 - 0.5 \text{ m}$
Massa:	$SpringM = 10 - 10'000 \text{ kg}$
Rigidità:	$SpringK = 0 - (F_{max} \cdot 1'000) \text{ N/m}$
Viscosità:	$SpringC = (SpringM/10) - (F_{max} \cdot 0.9) \text{ N/(m/s)}$

La corsa dell'attuatore è stata fissata pari ad 1 metro.

Nella finestra del programma (fig. 64), durante la simulazione, la configurazione dei vari elementi è riprodotta attraverso un'animazione grafica.

Le variabili fisiche di maggiore interesse sono visualizzate con 4 cursori a barra, nell'ordine:

- IN: è il segnale di input che, in dipendenza della modalità selezionata, corrisponde alla forza prodotta dalla mano, oppure alla posizione del joystick.
- Vel: è la velocità dell'attuatore
- Pos: è la posizione dell'attuatore
- Force: è la forza di contatto dell'estremità dell'attuatore con l'ambiente esterno.

È possibile scegliere tre diverse modalità di funzionamento:

0 - *Velocity*: è il controllo di velocità di JoyForce

1 - *No feedback*: è il controllo di velocità senza la retroazione sul joystick. Il comando viene impartito con la posizione della leva ed è comunque sempre presente la retroazione sul controllo dell'attuatore.

2 - *Force*: è il controllo di forza di JoyForce

La modalità viene comunicata al microcontrollore attraverso due uscite digitali della scheda di acquisizione; il joystick utilizza la logica di funzionamento in inseguimento di posizione nelle modalità 0 e 2, mentre nella modalità 1 riproduce la caratteristica elastica di un joystick tradizionale.

Nelle modalità 0 e 1 è possibile selezionare l'opzione di non reversibilità di movimento dell'attuatore nei confronti del comando. Questo può essere utile nel caso si debbano sollevare carichi sospesi, perché si ha la necessità che rimangano fermi anche in assenza del comando dell'operatore.

Questa opzione interviene nel sistema di controllo dell'attuatore disabilitando la retroazione sulla forza di contatto rilevata, quando è discorde rispetto al comando proveniente dal joystick.

La casella contrassegnata con “*Ghost*” imposta i riferimenti utilizzati nelle prove per confrontare le tre modalità di funzionamento, mentre la casella “*IN/OUT*” permette di disabilitare la scheda di acquisizione e di provare il programma di simulazione con il comando di input proveniente dal mouse. È anche possibile, attivando il mouse e mantenendo attiva la scheda di acquisizione, verificare l’inseguimento di posizione attuato dal controllo del joystick.

I tasti *Start* e *Reset* danno l’avvio e fermano la simulazione; il tasto *Save data* dà la possibilità di salvare la registrazione delle grandezze di simulazione in un file di testo (*.txt) oppure in un file di Matlab (*.m). La registrazione conserva la memorizzazione degli ultimi 100 secondi di simulazione, campionati con la risoluzione di un centesimo di secondo, delle seguenti grandezze:

- 1- segnale di input
- 2- posizione dell’attuatore
- 3- velocità dell’attuatore
- 4- forza rilevata sull’attuatore
- 5- posizione di riferimento (*Ghost*)

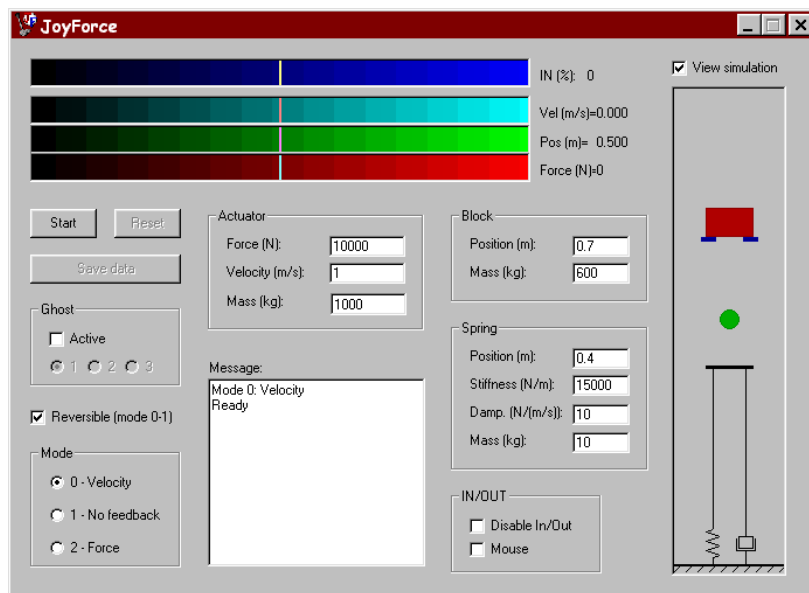


Figura 64: finestra grafica del programma di simulazione

2.4.1 La scheda di acquisizione National Instruments PCI-6024E

Per poter ricevere il segnale di comando proveniente dal joystick e mandare in uscita il segnale di posizione dell’attuatore, è stata utilizzata una scheda di acquisizione multifunzione a basso costo, della quale si riportano le caratteristiche principali:

Modello: National Instruments PCI-6024E

INPUT:

- 16 canali analogici di ingresso con una risoluzione di 12 bit
- frequenza di campionamento max: 200 kS/s
- input range: da ± 0.05 a ± 10 V
- Tolleranza per una tensione nominale di 5 V:
 - o assoluta sul fondoscala: 5.263 mV
 - o risoluzione: 2.95 mV

OUTPUT:

- 2 uscite analogiche con una risoluzione di 12 bit
- frequenza di aggiornamento max: 10 kS/s
- output range: ± 10 V
- 8 ingressi/uscite digitali
- 2 counter/timer a 24 bit
- Triggers digitali

Questa scheda è stata configurata e programmata all'interno della simulazione scritta in *VisualBasic 6.0*, utilizzando la libreria di funzioni NI-DAQ 6.9, reperibile gratuitamente nel sito del produttore (<http://www.ni.com>).

2.4.1.1 Programmazione dell'input/output

Senza entrare nei dettagli, si vuole descrivere com'è stata strutturata la comunicazione tra il joystick e la scheda di acquisizione.

Un sistema real-time, per poter funzionare correttamente, deve poter fare riferimento ad un sistema operativo specifico, che permetta di stabilire la priorità delle operazioni e che sia quindi in grado di eseguire le routine di calcolo e di input/output entro gli intervalli di tempo prestabiliti.

In questa applicazione, invece, sia il sistema operativo che la scheda di acquisizione non erano stati concepiti per questo tipo di utilizzo, cosicché si è dovuto ricorrere ad alcuni espedienti nella programmazione, e sono rimaste, comunque, delle limitazioni non superabili.

Il compito del programma di simulazione è di ricevere il segnale di input, elaborarlo, e mandare in uscita l'output. In questa applicazione è presente, come in tutti i sistemi real-time, la necessità di far trascorrere il minor tempo possibile tra l'acquisizione dell'input e la scrittura in uscita del segnale elaborato: il ritardo, infatti, costituisce uno sfasamento con il reale trascorrere del tempo e fa discostare la simulazione dal sistema fisico simulato.

La scheda di acquisizione registra il segnale campionato nella memoria locale, che viene successivamente trasferita in blocco nella memoria del PC con una chiamata specifica ad una funzione della libreria NI-DAQ.

La frequenza di campionamento determina la risoluzione del segnale, mentre il tempo che trascorre tra due trasferimenti successivi condiziona il ritardo della simulazione.

Si è fissata la frequenza di campionamento a 1 kHz, maggiore di almeno un ordine di grandezza sia alla capacità dell'operatore di impartire il comando, sia alla frequenza di taglio del filtro passa basso inserito in entrata, così da evitare il problema dell'aliasing. La frequenza di trasferimento è invece stata posta pari al massimo valore supportabile dalla scheda: questa caratteristica non è riportata nelle specifiche ed è stato trovato sperimentalmente un valore di 100 Hz.

Questa modalità di funzionamento della scheda, che permette il trasferimento dei dati mentre è in corso l'acquisizione, è denominata double-buffer perché viene allocato un buffer di dimensioni doppie rispetto al blocco di memoria trasferito: in questo modo, mentre è in corso la scrittura dei dati su una parte del buffer, è possibile trasferire l'altra metà senza provocare interruzioni.

In questa applicazione, per quanto detto, il blocco di memoria risulta composto da un array di 10 elementi, che viene trasferito nella memoria del PC ogni 0.01 secondi.

Come si vede nella rappresentazione di fig. 65, a trasferimento avvenuto, comincia la fase di elaborazione dei dati relativi all'intervallo di tempo precedente. L'elaborazione è costituita principalmente dalle routine di movimento, e deve essere terminata prima dell'inizio del trasferimento successivo.

L'output del segnale è stato impostato con gli stessi parametri dell'acquisizione, perché funziona in modo del tutto analogo, ed è stato così possibile sincronizzare i due trasferimenti (input ed output) nella stessa routine, che viene richiamata da un evento prodotto direttamente dalla scheda di acquisizione.

Indipendentemente dal tempo di calcolo (che deve sottostare solo al limite massimo di 0.01 secondi), allora, i dati elaborati vengono mandati in uscita nel trasferimento successivo.

Dunque, lo sfasamento temporale è pari a due intervalli di tempo, ovvero 0.02 secondi.

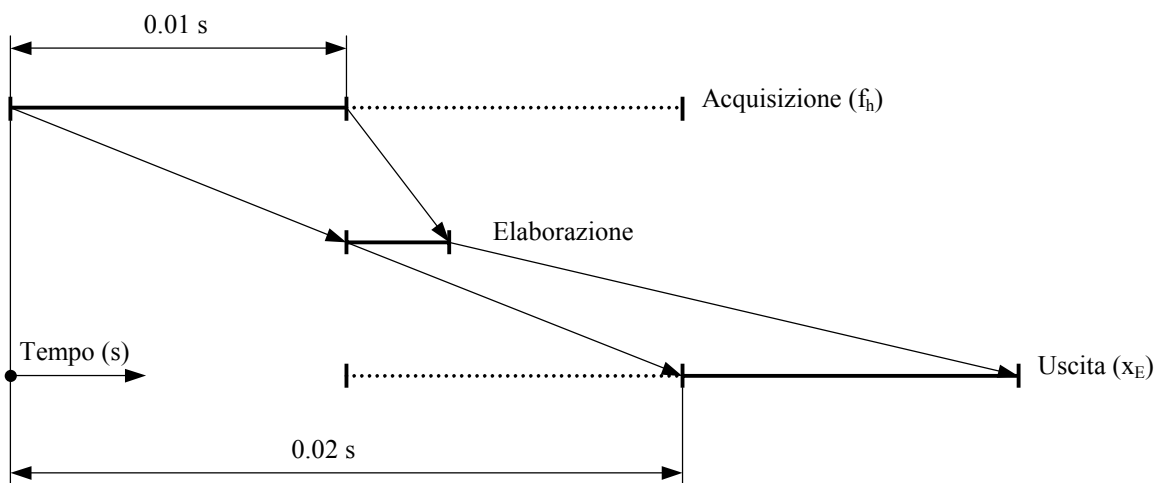


Figura 65: sfasamento temporale della simulazione

Il tempo di sfasamento, pur essendo il minimo permesso dalla scheda utilizzata, risulta sensibile nei confronti del controllo (vedi capitolo 2.2.1) e per questo ha rappresentato, da un lato, un problema aggiuntivo nei confronti di un sistema reale, e dall'altro ha dato l'opportunità di verificare la possibilità di impiego del sistema nelle applicazioni aventi questa caratteristica, prima fra tutte, la telemanipolazione.

2.4.2 Routine di movimento

Nella simulazione, il calcolo delle grandezze fisiche è stato eseguito con il metodo degli intervalli di tempo finiti, prendendo come tempo base (dt) lo stesso utilizzato nel campionamento del segnale di ingresso (1/1000 di secondo).

In questo metodo viene applicata una discretizzazione, in termini di tempo, delle variabili del sistema: si assume, pertanto, che il loro valore rimanga costante all'interno di ciascun intervallo dt .

Le equazioni differenziali che descrivono il sistema reale si riducono a delle equazioni lineari, che possono essere calcolate molto velocemente dalla routine di calcolo.

La routine può così essere richiamata con una frequenza elevata ed il tempo diviso in intervalli ridotti, che rende, di fatto, questa approssimazione aderente al modello continuo.

Per avere un'idea dell'errore commesso, si è provato a confrontare la soluzione del modello discretizzato con la soluzione esatta della posizione, in funzione del tempo, di un sistema massa-molla-smorzatore sottoposto ad oscillazione.

Sono stati fissati i parametri caratteristici:

costante di rigidezza:	$k = 20'000 \text{ N/m}$
massa:	$m = 3 \text{ kg}$
costante di smorzamento:	$c = 30 \text{ N/(m/s)}$

e le condizioni iniziali:

posizione iniziale:	$x_0 = 1 \text{ m}$
velocità iniziale:	$\dot{x}_0 = 0 \text{ m/s}$

Calcolati i coefficienti:

$$\omega_n = \sqrt{\frac{k}{m}} \quad \text{e} \quad \zeta = \frac{\omega_n \cdot c}{2 \cdot k}$$

l'espressione esatta è data da:

$$y(t) = e^{-\zeta \cdot \omega_n \cdot t} \cdot \left[\frac{\dot{x}_0 + \zeta \cdot \omega_n \cdot x_0}{\omega_n \cdot \sqrt{1 - \zeta^2}} \cdot \sin(\omega_n \cdot \sqrt{1 - \zeta^2} \cdot t) + x_0 \cdot \cos(\omega_n \cdot \sqrt{1 - \zeta^2} \cdot t) \right]$$

La soluzione approssimata è invece stata calcolata ad intervalli di tempo $dt = 0.001$ secondi, con le seguenti espressioni:

$$F = -c \cdot \dot{x}_0 - k \cdot x_0$$

$$\dot{x}'_0 = \dot{x}_0 + \frac{F}{m} \cdot dt$$

$$x'_0 = x_0 + \dot{x}'_0 \cdot dt$$

Nel grafico di sinistra della fig. 66, gli andamenti delle due curve risultano sovrapposti; nel grafico di destra è stato allora diagrammato il valore dell'errore, che è risultato rimanere sempre al di sotto del 4 % rispetto al valore di oscillazione massimo.

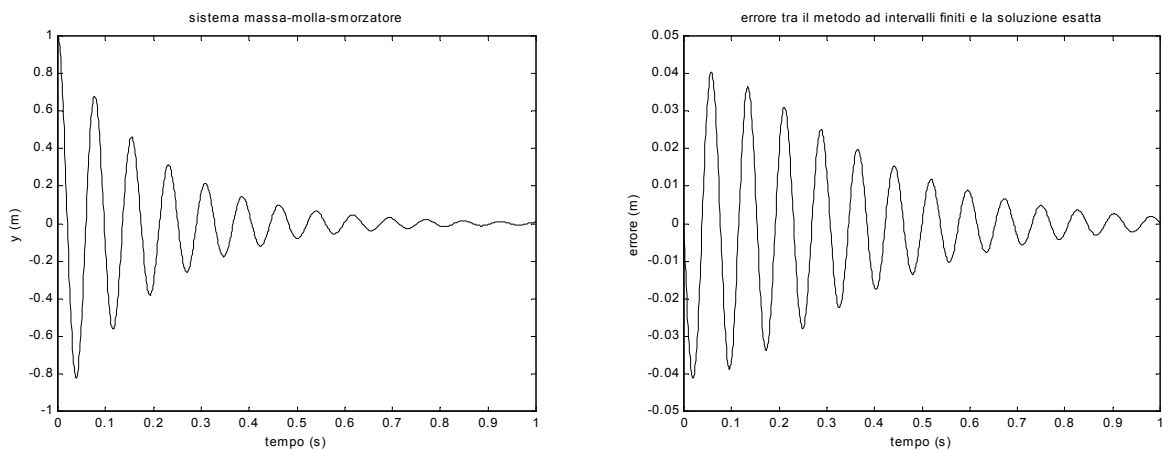


Figura 66: confronto tra la soluzione esatta ed il metodo agli intervalli di tempo finiti

Un requisito fondamentale perché i risultati ottenuti con questo metodo siano attendibili, è che la frequenza degli intervalli sia di almeno un ordine di grandezza superiore alla frequenza naturale del sistema esaminato.

Nei due paragrafi successivi vengono illustrate le routine di movimento per i due controlli distinti di velocità e di forza; si riportano, qui sotto, le variabili fisiche definite per ciascuno dei tre elementi del sistema:

Attuatore:

massa	Mass
velocità	Vel
posizione	Pos
forza esterna	Forza

Carico sospeso (blocco)

posizione appoggi	BlockX0
massa	BlockM
velocità	Blockv
posizione	Blockp

Massa-molla-smorzatore (molla)

posizione a riposo	SpringX
massa	SpringM
costante elastica	SpringK
coeff. di smorzamento	SpringC
velocità	Springv
posizione	Springp

Infine, con IN1 è stato indicato il segnale di input, che in funzione della modalità può essere la forza esercitata dalla mano dell'operatore oppure la posizione imposta alla leva del joystick.

2.4.2.1 Comando di velocità

In questo tipo di comando si presuppone che l'attuatore sia in grado di muoversi con la velocità impartita, indipendentemente dalla resistenza incontrata.

Nella routine di movimento, la prima cosa che viene fatta è una copia dei valori di ingresso delle variabili degli elementi attuatore, blocco e molla.

Poi, sia il blocco che la molla vengono fatti muovere come se fossero isolati; per il blocco si usano le espressioni:

$$\text{Blockv} = \text{Blockv} - g \cdot dt$$

$$\text{Blockp} = \text{Blockp} + \text{Blockv} \cdot dt$$

Per la molla si utilizzano delle espressioni analoghe, previo aver calcolato la forza relativa all'azione della costante elastica e della viscosità nell'istante precedente:

$$\text{Springf} = -\text{SpringK} \cdot (\text{Springp} - \text{SpringX}) - \text{SpringC} \cdot \text{Springv}$$

$$\text{Springv} = \text{Springv} + \frac{\text{Springf}}{\text{SpringM}} \cdot dt$$

$$\text{Springp} = \text{Springp} + \text{Springv} \cdot dt$$

Per il blocco viene eseguito un controllo: se si trova al di sotto degli appoggi viene riportato sulla posizione minima; lo stesso viene fatto per la molla se è uscita dal campo inferiore.

Viene calcolata la velocità di riferimento dell'attuatore: questa è la velocità derivante dalla retroazione sul controllo dell'attuatore e deve essere successivamente filtrata per ripristinare la stabilità nei sistemi con ritardo (vedi capitolo 2.2.1). L'espressione è la seguente:

$$V_{\text{rif}} = V_{\text{max}} \cdot \left(\text{IN1} - \frac{\text{Forza}}{F_{\text{max}}} \right)$$

Il filtro passa basso è stato implementato riportandolo al sistema meccanico equivalente: si può pensare ad un sistema molla-smorzatore in cui V_{rif} è la grandezza in ingresso x_i della fig. 67.

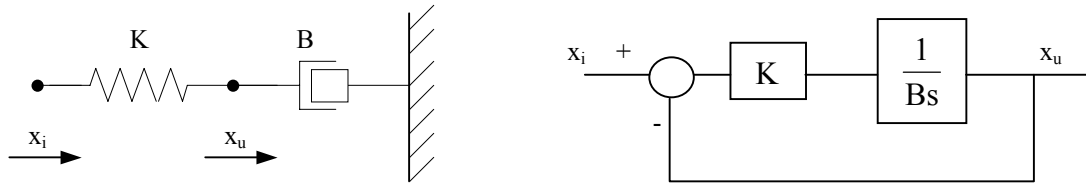


Figura 67: schema meccanico di un filtro passa basso

La funzione di trasferimento completa è:

$$\frac{x_u}{x_i} = \frac{K}{B \cdot s} \cdot \frac{1}{1 + \frac{K}{B \cdot s}} = \frac{1}{\frac{B}{K} \cdot s + 1}$$

e quindi la costante di tempo è data da:

$$\tau = \frac{B}{K}$$

Nella routine di calcolo il valore di ingresso, invece di una posizione, è una velocità, e perciò il sistema è stato pensato come uno smorzatore collegato ad una massa; la massa è stata considerata unitaria e si è preso il coefficiente di smorzamento $\text{GainVirt} = 10$. La costante di tempo vale:

$$\tau = \frac{1}{\text{GainVirt}} = \frac{1}{10} = 0.1 \text{ s}$$

La frequenza di taglio risulta:

$$f_{\tau} = \frac{1}{2 \cdot \pi \cdot \tau} = \frac{1}{2 \cdot \pi \cdot 0.1} = 1.59 \text{ Hz}$$

Nel metodo degli intervalli di tempo finiti questo sistema è stato tradotto nelle due equazioni:

$$F_{virt} = Gain_{Virt} \cdot (V_{rif} - Vel)$$

$$Vel = Vel + F_{virt} \cdot dt$$

La routine procede con il calcolo della posizione dell'attuatore, effettuata con la stessa tecnica:

$$Pos = Pos + Vel \cdot dt$$

A questo punto vengono attuati i controlli sulla collisione dell'attuatore con uno degli altri due elementi: nel caso si verifichi una di queste condizioni, l'elemento interessato viene posto ai medesimi valori di velocità e posizione dell'attuatore, secondo le ipotesi iniziali.

Rimane da calcolare la forza di contatto, che viene ricavata dall'accelerazione dell'elemento interessato.

Per il carico sospeso vale la relazione:

$$Forza = BlockM \cdot \left(g + \frac{Blockv - oldBlockv}{dt} \right)$$

Mentre per massa-molla-smorzatore viene utilizzata l'espressione:

$$Forza = SpringM \cdot \frac{Springv - oldSpringv}{dt} + SpringC \cdot Springv + SpringK \cdot (Springp - SpringX)$$

Sulla forza così calcolata viene eseguita una sogliatura con il valore massimo, in modo da preparare il valore per il ciclo di calcolo successivo.

L'opzione di non reversibilità del comando è stata ottenuta con una condizione, posta subito dopo il calcolo della velocità di riferimento V_{rif} , in cui vengono confrontati il segno del segnale di input con quello della forza rilevata sull'attuatore.

2.4.2.2 Comando di forza

Il comando di forza intende simulare il comportamento di un attuatore elettrico in cui la regolazione viene effettuata sulla coppia del motore, e di conseguenza sulla forza sviluppata all'esterno. Quindi, rispetto alla routine vista per il controllo di velocità, le differenze riguardano il controllo dell'attuatore e la dinamica dei contatti con gli elementi esterni.

Come analizzato nel capitolo 2.2.2, in questo tipo di comando è stata inserita una viscosità fittizia nell'anello di retroazione del controllo dell'attuatore; nell'implementazione è stata tradotta nelle seguenti due espressioni:

$$Forza = F_{max} \cdot (IN1 - visc \cdot Vel)$$

$$Vel = Vel + \frac{Forza}{Mass} \cdot dt$$

La posizione è stata calcolata allo stesso modo visto in precedenza, mentre, nel caso sia verificata la condizione di collisione, questa volta è stata assunta l'ipotesi di urto completamente anelastico.

Pertanto, il calcolo della velocità dell'attuatore, uguale a quella dell'elemento con cui si è verificato il contatto, viene eseguito nella condizione di conservazione della quantità di moto.

In realtà, dopo aver calcolato la collisione, è necessario anche aggiornare le variabili con quanto accade nell'intervallo di tempo in corso; le espressioni utilizzate per il carico sospeso sono:

$$\text{Acc} = \frac{\text{Forza} - \text{BlockM} \cdot g}{\text{BlockM} + \text{Mass}}$$

$$\text{Vel} = \frac{\text{oldBlockv} \cdot \text{BlockM} + \text{oldVel} \cdot \text{Mass}}{\text{BlockM} + \text{Mass}}$$

$$\text{Vel} = \text{Vel} + \text{Acc} \cdot dt$$

$$\text{Pos} = \text{oldPos} + \text{Vel} \cdot dt$$

Mentre, per il sistema massa-molla-smorzatore, sono state utilizzate le equazioni:

$$\text{Acc} = \frac{\text{Forza} - \text{SpringK} \cdot (\text{oldSpringp} - \text{SpringX}) - \text{SpringC} \cdot \text{oldSpringv}}{\text{SpringM} + \text{Mass}}$$

$$\text{Vel} = \frac{\text{oldSpringv} \cdot \text{SpringM} + \text{oldVel} \cdot \text{Mass}}{\text{SpringM} + \text{Mass}}$$

$$\text{Vel} = \text{Vel} + \text{Acc} \cdot dt$$

$$\text{Pos} = \text{oldPos} + \text{Vel} \cdot dt$$

2.4.3 Salvataggio dei dati

I dati vengono registrati, durante l'esecuzione del programma, in una tabella di memoria con 5 colonne di 10'000 elementi ciascuna, una per ogni grandezza registrata:

- 1 - segnale di input
- 2 - posizione dell'attuatore
- 3 - velocità dell'attuatore
- 4 - forza rilevata sull'attuatore
- 5 - posizione di riferimento (*Ghost*)

La routine di calcolo, per ogni trasferimento di output, elabora i dati relativi al campionamento di 10 istanti distinti, perché, come si è visto, il campionamento avviene ad una frequenza 10 volte superiore alla frequenza di trasferimento. Il dato registrato è la media dei valori assunti dalla grandezza in quei 10 istanti, così, se il campionamento avviene ad 1 kHz, la registrazione fa riferimento ad una frequenza di 100 Hz. Quindi, i 10'000 elementi a disposizione sono sufficienti a memorizzare un tempo massimo di 100 secondi.

Alla tabella è stato associato un indice che individua la posizione progressiva e, quando raggiunge la fine, viene riportato sul primo elemento (tabella circolare).

In questo modo, trascorso il tempo massimo di memorizzazione, i primi dati registrati cominciano ad essere sovrascritti, e vengono sempre conservati gli ultimi 100 secondi di simulazione.

Al momento del salvataggio, un flag indica se è stata utilizzata tutta la tabella: nel qual caso i dati vengono salvati sul file prendendo prima la parte di tabella successiva alla posizione dell'indice, e poi la rimanente parte iniziale, in modo da ristabilire l'ordine temporale.

Terminata la simulazione, con il pulsante *Save data* è possibile salvare i dati immagazzinati in memoria in un file di testo (*.txt) oppure in un file leggibile direttamente da Matlab (*.m).

Nel primo caso i valori sono salvati in formato ascii, incolonnati nello stesso ordine citato all'inizio e separati tra di loro da una virgola; questo formato è importabile, per esempio, in un foglio Excel.

Nel caso dell'm-file viene aggiunta una riga di intestazione ($y=[$), e una di chiusura ($]$); è sufficiente digitare il nome del file sulla riga di comando della shell di Matlab per caricare la matrice completa (chiamata y).

Cap. 3: RISULTATI

La valutazione della sensibilità fornita dal sistema JoyForce è stata effettuata confrontando le due modalità di funzionamento previste (in controllo di velocità e di forza) con quella senza retroazione sul joystick. È da tener presente che anche quest'ultima modalità rappresenta un'evoluzione rispetto ad un joystick tradizionale perché viene eseguita la retroazione sul sistema di controllo dell'attuatore descritta nel capitolo 1.3 (NTRFC).

Quindi, il confronto non riguarda la capacità del sistema di ridurre le forze negli impatti e di evitare i sovraccarichi (di cui si trovano dei raffronti in altre ricerche [5]), ma è mirato esclusivamente a quantificare il vantaggio offerto dalla sensibilità restituita alla mano dell'operatore, in termini di tempestività e precisione di manovra.

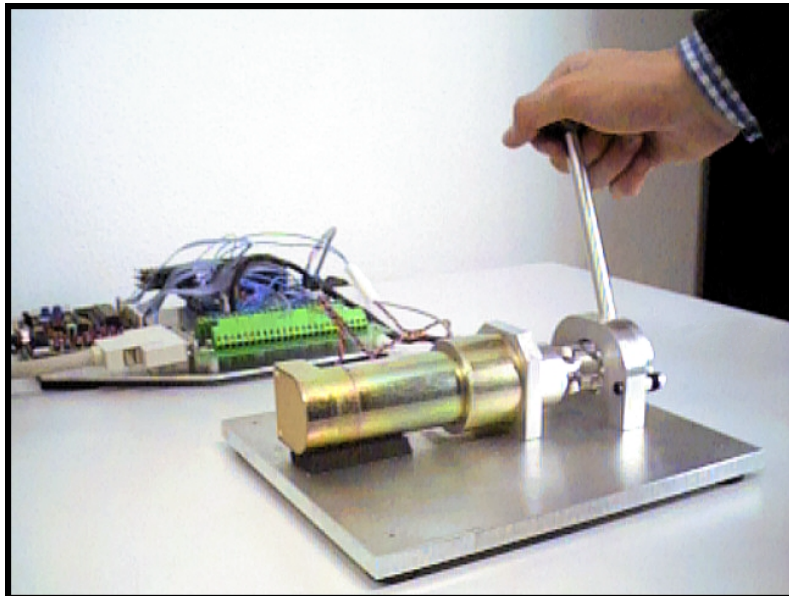


Figura 68: fotografia del prototipo realizzato

Le prove sono state eseguite chiedendo ad una stessa persona, priva di esperienza specifica, di eseguire delle manovre standard nell'ambiente simulato al PC, alternando le diverse modalità di controllo.

Sono stati definiti tre tipi di manovre standard, ciascuna atta ad evidenziare un singolo parametro di valutazione, selezionabili con l'opzione *Ghost* del programma di simulazione.

Quando *Ghost* è attivo, compare sull'animazione grafica del sistema fisico simulato un segnalino mobile, che rappresenta, per chi sta operando la manovra, il riferimento da raggiungere con l'attuatore nel minor tempo possibile e con la massima precisione di cui è capace.

Nei paragrafi successivi si descrivono i tre tipi di manovre utilizzate e si riportano i rilevamenti sperimentali effettuati.

3.1 INSEGUIMENTO DI TRAIETTORIA

In questa prova il segnalino di riferimento si sposta su posizioni successive con una velocità prefissata, e in ogni posizione rimane fermo per qualche secondo.

La caratteristica analizzata è l'attitudine del controllo ad impostare sia la velocità che la posizione dell'attuatore desiderate dall'operatore.

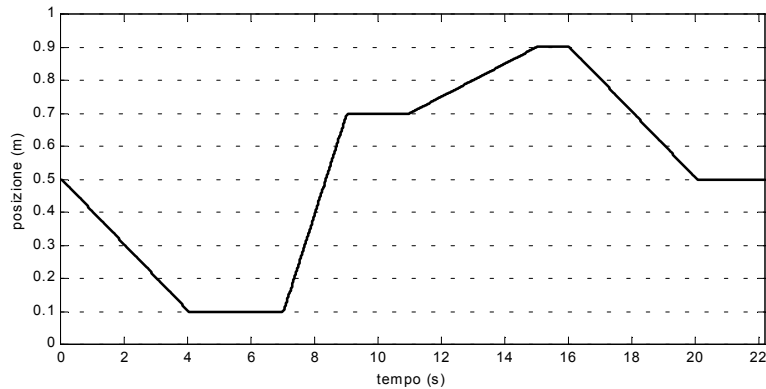


Figura 69: traiettoria di riferimento

La traiettoria seguita dal riferimento è tracciata nel grafico di fig. 69, e rappresenta le seguenti operazioni:

- Deve essere raggiunta la posizione di 0.1 m alla velocità di 0.1 m/s; durante il movimento avviene il contatto con il sistema massa-molla-smorzatore. La posizione rimane costante per 3 secondi.
- Fase di risalita fino alla posizione corrispondente agli appoggi del carico sospeso alla velocità di 0.3 m/s. Si arriva quindi a toccare il blocco senza sollevarlo, e si rimane fermi per 2 secondi.
- Si opera il sollevamento lento della massa (velocità di 0.05 m/s) fino alla posizione di 0.9 m, e poi si rimane immobili per 1 secondo.
- Si ritorna alla posizione iniziale con la velocità di 0.1 m/s.

Nella fig. 70 sono stati riportati i diagrammi relativi al controllo di velocità.

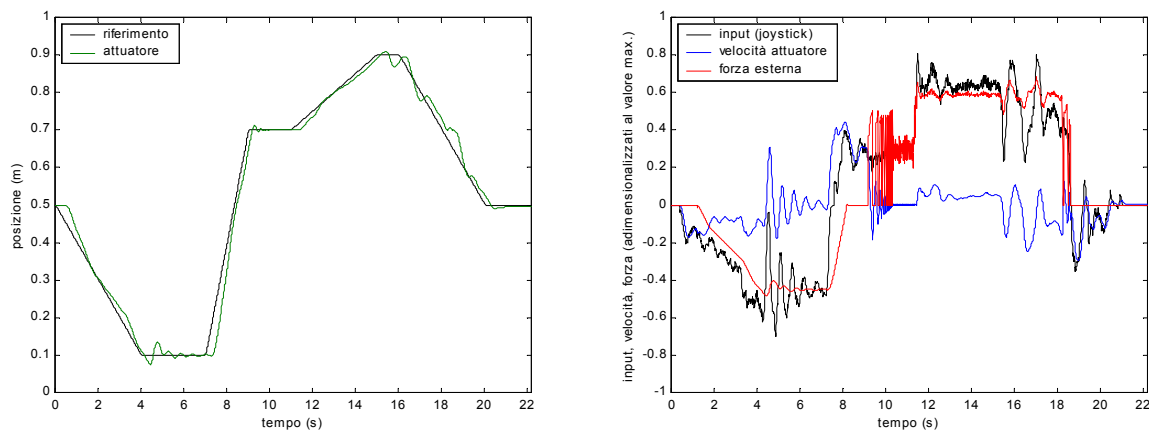


Figura 70: inseguimento di traiettoria con il controllo di velocità

Dal grafico della posizione si vede che, dopo il tempo di reazione, l'operatore è in grado di raggiungere il riferimento e di seguirlo con la stessa velocità. Nei punti di sosta è presente

un'oscillazione che va smorzandosi, perché il sistema con cui l'attuatore si trova in contatto possiede una determinata massa, percepita sul joystick, la quale rende impossibile seguire la discontinuità della velocità del riferimento. In quella fase, si nota che la forza esercitata sul joystick compie delle oscillazioni per riportare l'attuatore alla posizione corretta.

Il contatto con il carico sospeso non crea nessun problema, e anche il sollevamento avviene in modo molto regolare. La fase di discesa accusa, invece, qualche oscillazione ed è presente uno scostamento dalla linea ideale proprio nel momento in cui il carico ritorna sugli appoggi: questo è dovuto all'elevata variazione della forza esercitata dall'operatore, che d'improvviso sente mancare il peso da sostenere.

La fig. 71 riporta la prova eseguita con la modalità senza retroazione.

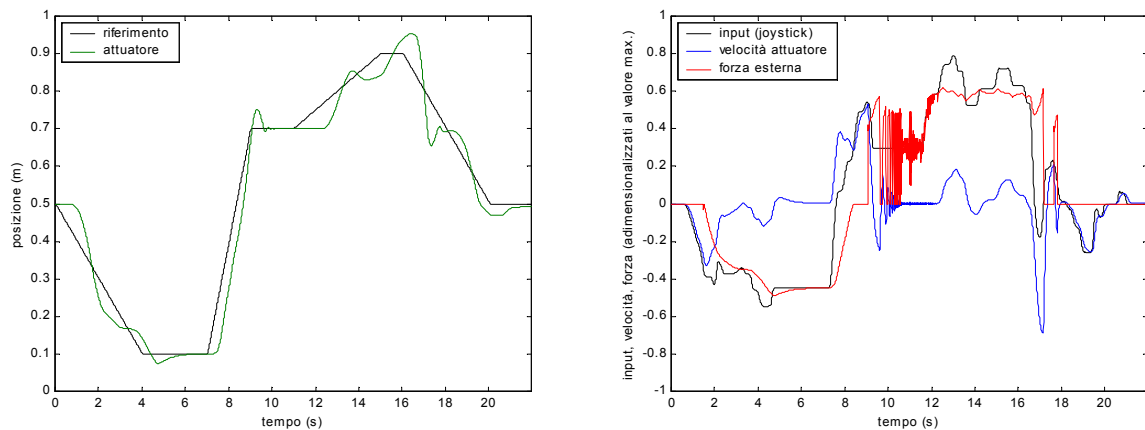


Figura 71: inseguimento di traiettoria con il controllo senza retroazione

In questo caso si nota una maggiore difficoltà nel seguire il movimento del riferimento, dovuto presumibilmente alla minore velocità con cui il segnale di input può variare: si ricorda che in questa modalità l'input è costituito dalla posizione della leva, a differenza delle altre due modalità in cui rappresenta la forza applicata alla leva del joystick. Nella corsa di risalita c'è un maggiore ritardo e nel contatto con il carico si verifica un overshoot elevato.

Per quanto riguarda il sollevamento, questo metodo di controllo si dimostra completamente inadatto perché non riesce nemmeno a raggiungere la posizione stabile, e la corsa di ritorno appare incontrollata.

Per ultima si riporta la modalità in controllo di forza, rappresentata nella sottostante fig. 72:

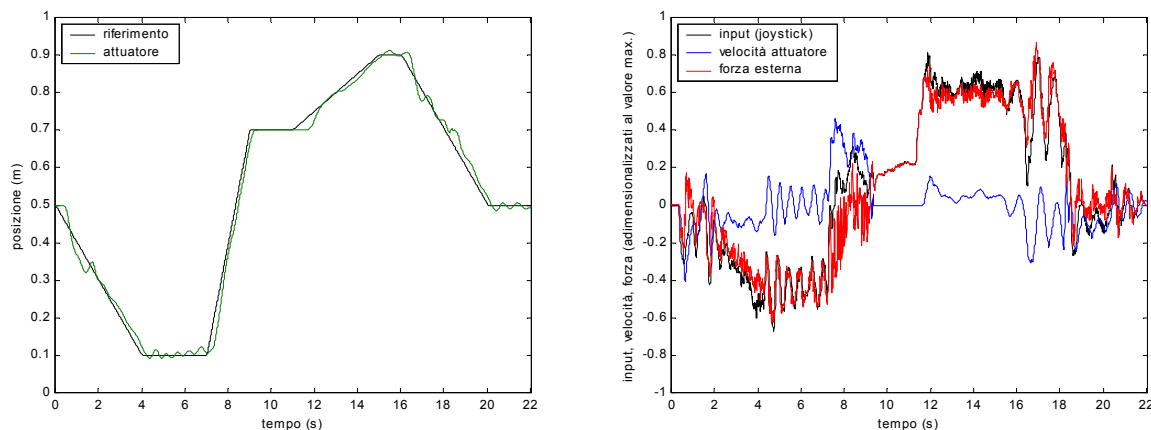


Figura 72: inseguimento di traiettoria con il controllo di forza

L'andamento della posizione è simile a quello visto per il controllo di velocità, con la differenza che l'attuatore si muove con maggiore tempismo e le oscillazioni hanno minore ampiezza, ma persistono maggiormente.

Nel grafico di destra spicca la sovrapposizione, anche alle frequenze maggiori, tra il segnale di input e la forza dell'attuatore: la spiegazione è data dalla corrispondenza diretta che hanno in questo tipo di controllo.

Dopo aver analizzato singolarmente le tre modalità, la fig. 73 raggruppa i diagrammi delle posizioni per poter effettuare un confronto diretto.

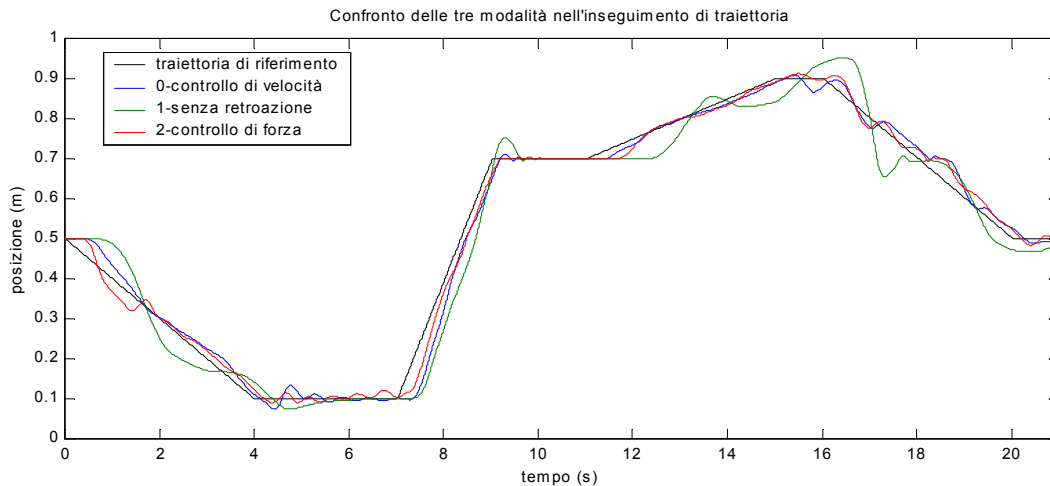


Figura 73: confronto tra le traiettorie percorse con i 3 metodi di controllo

Il grafico conferma la similitudine tra il comportamento del controllo di velocità con quello di forza, che si discostano invece dal controllo senza retroazione.

La vicinanza delle curve non permette ulteriori considerazioni e, per mettere in maggiore risalto le differenze, si riporta il diagramma degli errori tra la traiettoria seguita e quella di riferimento (fig. 74):

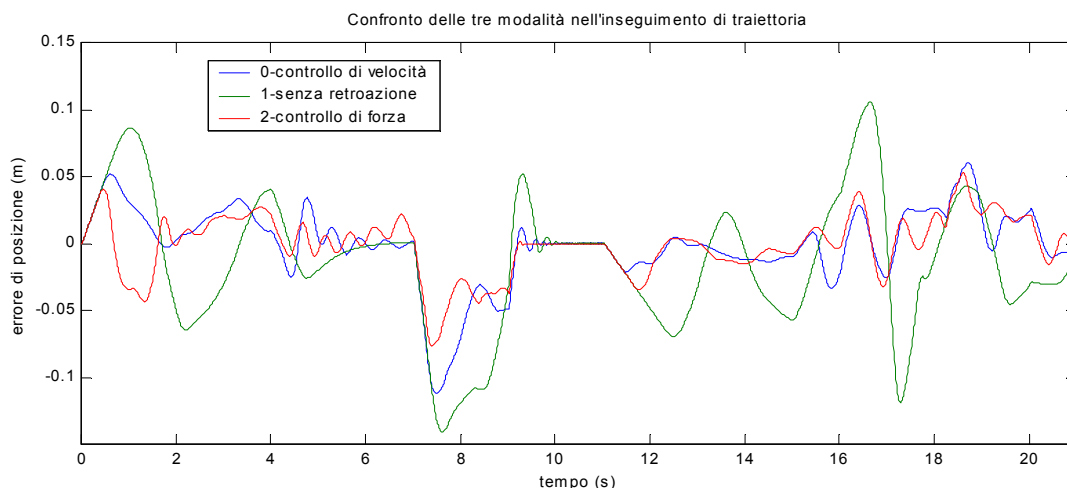


Figura 74: errore dalla traiettoria di riferimento dei tre metodi di controllo

In tutta la durata della prova, si vede che il controllo di forza è quello che reagisce prima ai cambi di direzione, seguito dal controllo di velocità e poi da quello senza retroazione.

L'errore rimane contenuto, con ampiezze simili, per i controlli di velocità e forza, mentre ha dei picchi circa doppi per il controllo senza retroazione.

Una misura oggettiva della capacità di seguire la traiettoria è data dal valore dell'errore medio assoluto, calcolato con la formula:

$$e_{m,a} = \frac{\int_0^T |e| \cdot dt}{T}$$

Riferito in percentuale alla corsa totale di 1 metro, è risultato essere per i tre controlli:

Modalità di funzionamento	Errore medio assoluto (%)	Variazione %
0 - Controllo di velocità (JoyForce)	1.85 %	-51.3 %
1 – Controllo senza retroazione	3.80 %	(valore max.)
2 – Controllo di forza (JoyForce)	1.60 %	-57.9 %

I valori calcolati rispettano le impressioni avute dalla lettura dei diagrammi, con un errore medio assoluto del controllo senza retroazione maggiore del doppio rispetto a quello degli altri due controlli.

Il valore più basso è stato ottenuto dal controllo di forza con l'1.60 % riferito all'intera corsa.

3.2 VELOCITÀ DI ESECUZIONE DELLA MANOVRA

Questa prova è stata definita con l'intento di valutare la capacità dell'operatore di posizionare l'attuatore su delle posizioni prefissate nel minor tempo possibile.

Il riferimento viene posto su una posizione che l'attuatore deve cercare di raggiungere con la tolleranza dell'1 %, sia sulla posizione che sulla velocità. Una volta soddisfatta tale condizione, il segnalino si sposta sul riferimento successivo.

In tutto sono state definite 6 posizioni diverse, che interessano il campo del sistema massa-molla-smorzatore e del carico sospeso.

Sono state cronometrate 5 prove per ognuno dei controlli del joystick, senza effettuare test preliminari (è per questa ragione che si notano dei miglioramenti nelle ultime prove).

I risultati ottenuti sono riportati nella tabella sottostante:

Modalità di funzionamento	Tempo (secondi)				
	1°	2°	3°	4°	5°
0 - Controllo di velocità (JoyForce)	15.35	10.60	10.69	8.80	8.79
1 – Controllo senza retroazione	26.25	16.22	16.03	14.07	15.10
2 – Controllo di forza (JoyForce)	14.90	12.34	9.60	8.86	8.93

La valutazione è stata eseguita sul tempo medio delle 5 prove effettuate, e su questo tempo è stata calcolata la variazione in percentuale rispetto al tempo medio più alto, ottenuto con il controllo senza retroazione. La seguente tabella raccoglie i risultati.

Modalità di funzionamento	Tempo medio (s)	Variatione %
0 - Controllo di velocità (JoyForce)	10.85	-38.1 %
1 – Controllo senza retroazione	17.53	(tempo max.)
2 – Controllo di forza (JoyForce)	10.93	-37.7 %

Il tempo più basso è stato registrato dal controllo di velocità, con un risparmio di tempo del 38.1 % rispetto al controllo senza retroazione; il controllo di forza ha dimostrato un miglioramento analogo con il 37.7 % di tempo in meno.

Segue l'analisi delle migliori prove ottenute con ciascuno dei tre controlli. Nella fig. 75 è rappresentata la prova n° 5 del controllo di velocità:

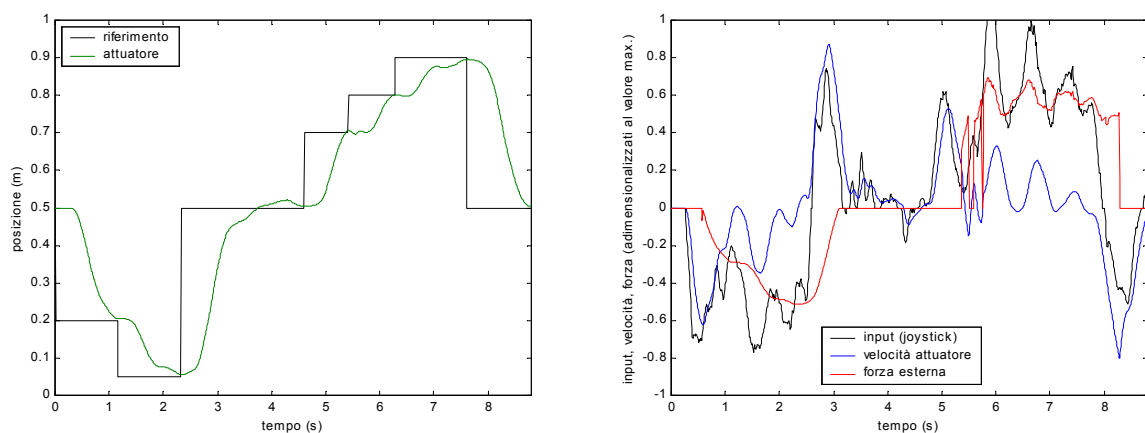


Figura 75: miglior rilevamento del controllo di velocità

Dai grafici si vede che l'attuatore arriva sulle posizioni di riferimento in modo puntuale e con velocità nulla; solo in un caso è presente un overshoot che costringe l'operatore a ritornare indietro. Si rileva un'elevata omogeneità di comportamento, sia in movimento libero che nei contatti con il sistema massa-molla-smorzatore e il carico sospeso.

Nella fig. 76 si riportano i diagrammi della 4° prova con il controllo senza retroazione:

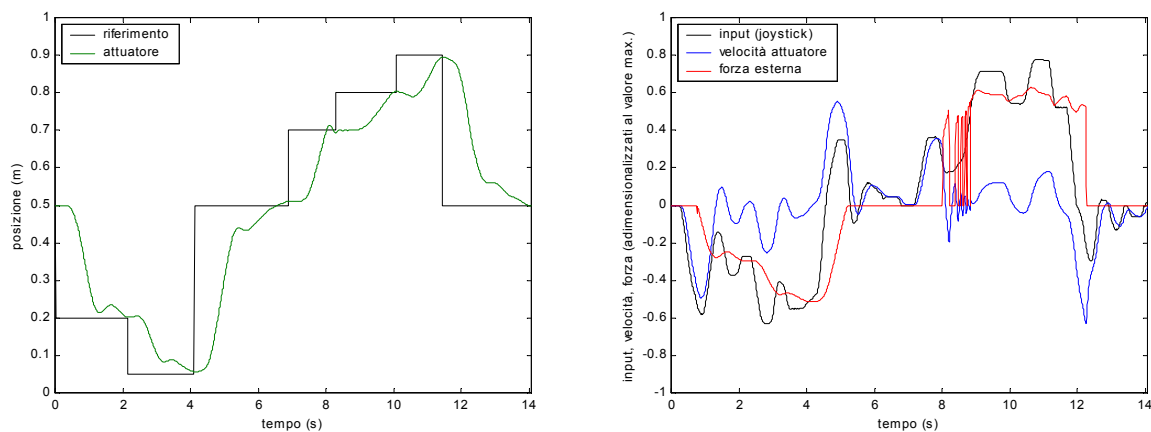


Figura 76: miglior rilevamento del controllo senza retroazione

In questo caso l'operatore si avvicina per tentativi e sono sempre presenti delle correzioni prima del raggiungimento della posizione di riferimento. Non sono presenti differenze rilevanti tra le diverse situazioni di carico.

Seguono, nella fig. 77, i risultati della 4° prova del controllo di forza:

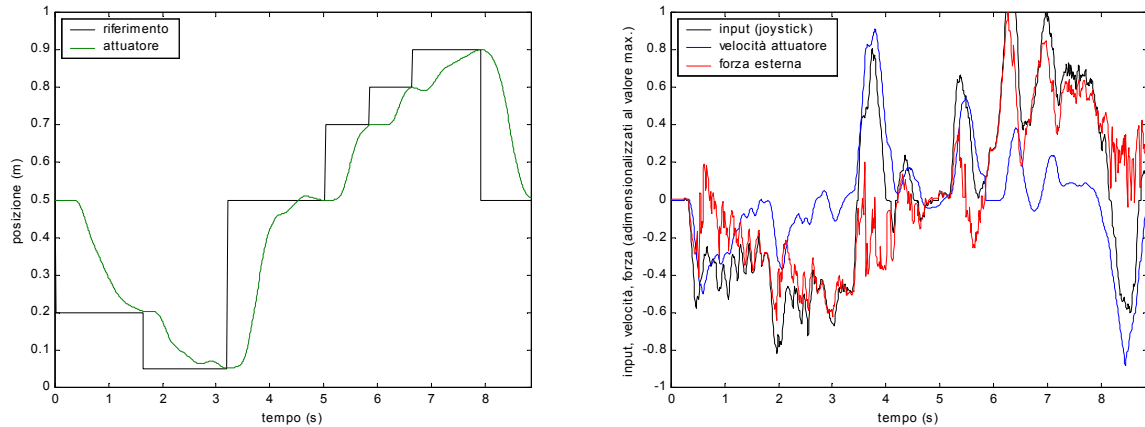


Figura 77: miglior rilevamento del controllo di forza

I grafici denotano un comportamento del tutto simile a quello registrato nel controllo di velocità.

3.2.1 Confronto con l'opzione di non reversibilità

Le prime due modalità, in controllo di velocità con e senza retroazione, sono state sottoposte alla stessa prova con l'opzione di non reversibilità attivata.

Nella fig. 78 si riportano i diagrammi del controllo retroazionato (JoyForce):

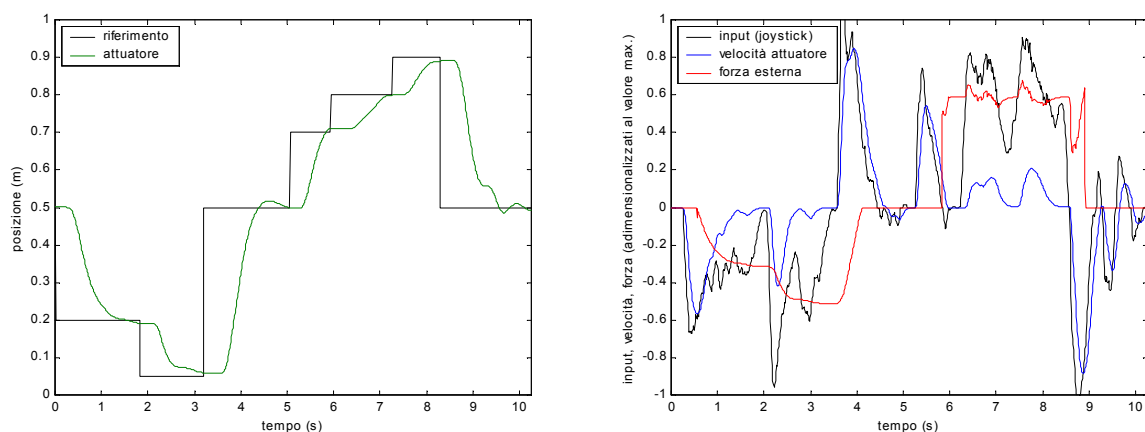


Figura 78: controllo di velocità non reversibile

Questa opzione interviene quando è presente una forza esterna, ed infatti si vedono delle differenze nelle situazioni in cui è presente il contatto con gli elementi esterni.

Più precisamente la posizione viene raggiunta con una curva più piatta e senza avere alcuna oscillazione successiva; il tempo di 10.25 secondi risulta maggiore di quello registrato precedentemente, a causa dei maggiori tempi di reazione nel cercare il riferimento successivo.

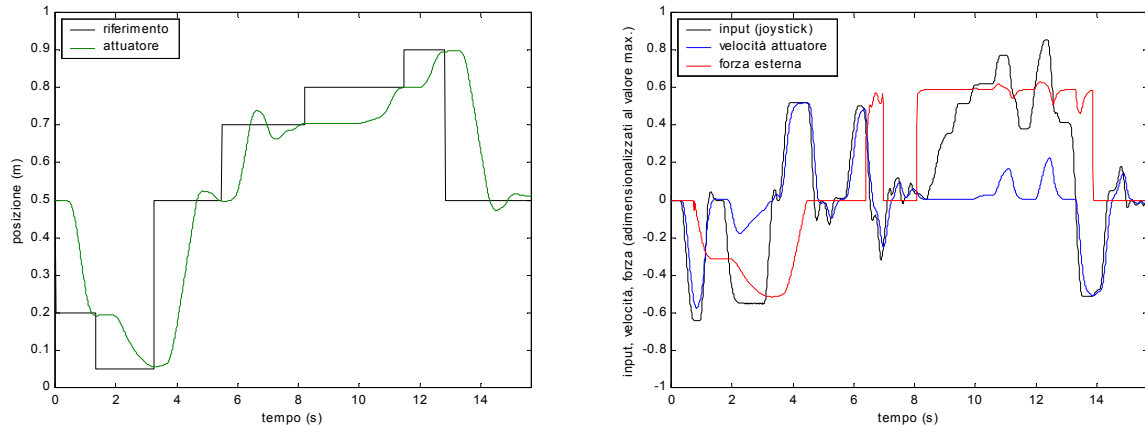


Figura 79: controllo senza retroazione e non reversibile

La fig. 79 riporta la prova del sistema non retroazionato.

Il tempo di esecuzione è pari a 15.66 secondi ed è in linea con quelli fatti registrare dallo stesso sistema reversibile. Dal grafico appare una maggiore stabilità di movimento, che non accusa più le elevate oscillazioni nel sollevamento del carico, ma rimane un ritardo importante nel valutare il comando da imprimere per compensare le forze esterne.

Il confronto tra i due sistemi di controllo stabilisce un vantaggio di tempo del 35 % per il sistema retroazionato (JoyForce).

3.3 INSEGUIMENTO DI POSIZIONE

In questa prova il segnalino, ad intervalli di 4 secondi, passa in successione su delle posizioni di riferimento predefinite, come illustrato nella fig. 80.

In questo modo viene messa in evidenza la capacità di raggiungere tempestivamente una posizione e di mantenerla in modo stabile, in più, facendo riferimento ad una traccia costante è possibile effettuare un confronto diretto dei risultati, come avvenuto per l'inseguimento di traiettoria.

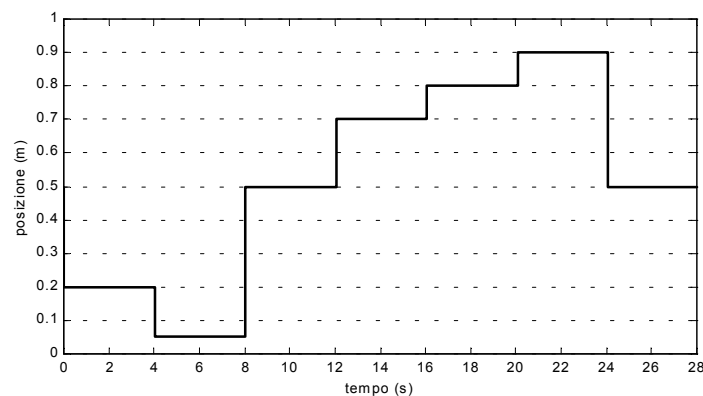


Figura 80: posizioni di riferimento

La fig. 81 riporta i diagrammi del controllo di velocità, in cui si vede che il tempo di reazione è molto breve, cosicché la posizione viene raggiunta tempestivamente. Anche l'overshoot presenta un valore minimo, ma, con l'eccezione del riferimento in campo libero, permane un'oscillazione della posizione.

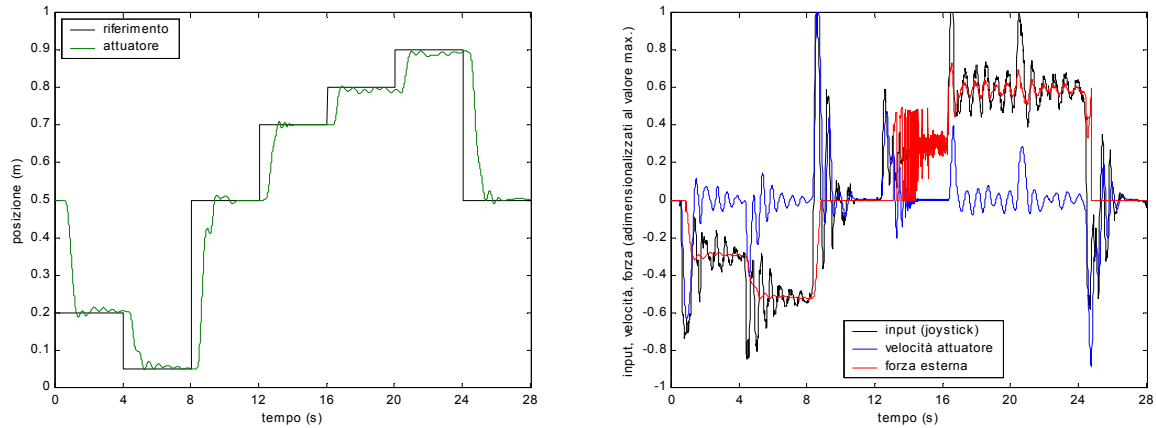


Figura 81: inseguimento di posizione con il controllo di velocità

Nella fig. 82 si trovano i grafici della prova eseguita con il controllo senza retroazione:
 Le caratteristiche, evidenziate anche nelle altre prove, sono una risposta più lenta, denotata anche dai minori valori di velocità dell'attuatore, ed una elevata difficoltà nel mantenere il carico sospeso all'altezza desiderata.

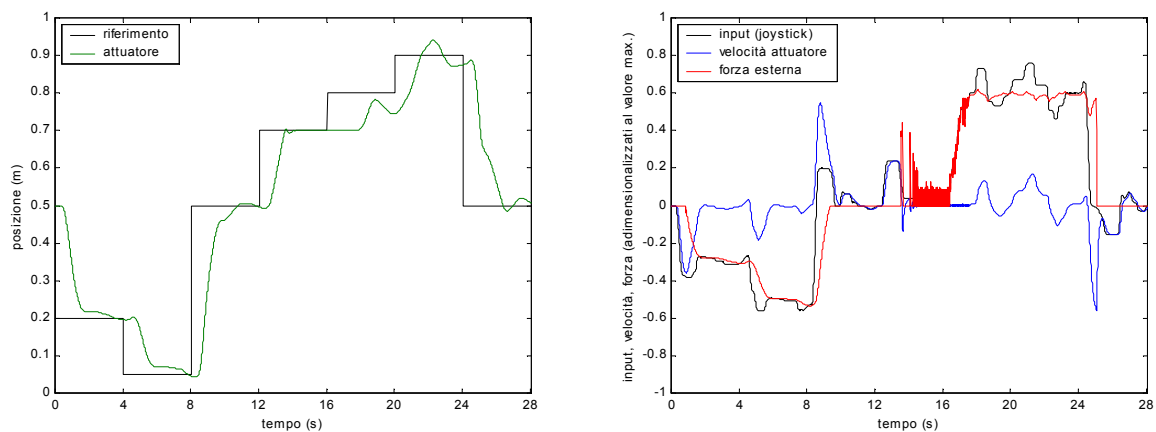


Figura 82: inseguimento di posizione con il controllo senza retroazione

La fig. 83, relativa al controllo di forza, presenta un andamento della posizione simile al controllo di velocità, con delle oscillazioni di ampiezza minore, ma aventi una frequenza più elevata.

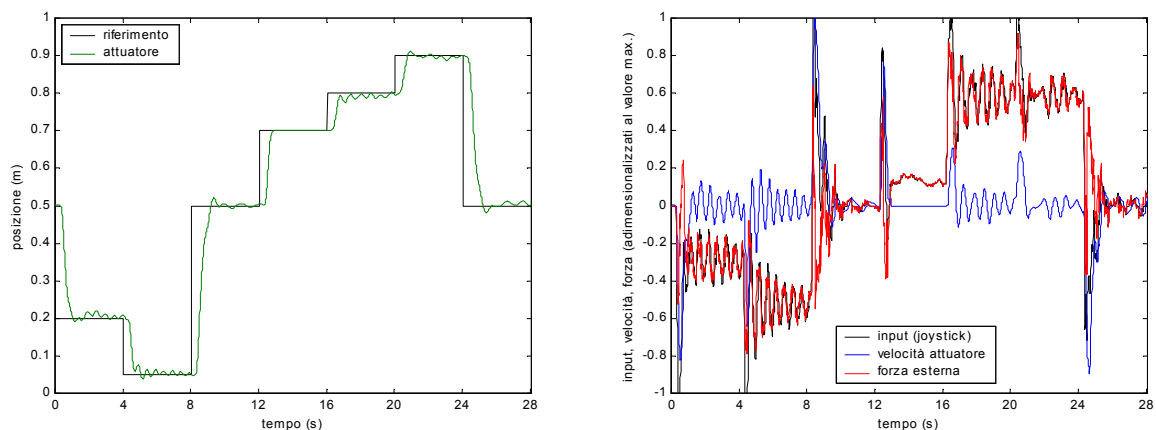


Figura 83: inseguimento di posizione con il controllo di forza

Nel grafico di fig. 84 sono stati sovrapposti i tre diagrammi della posizione.

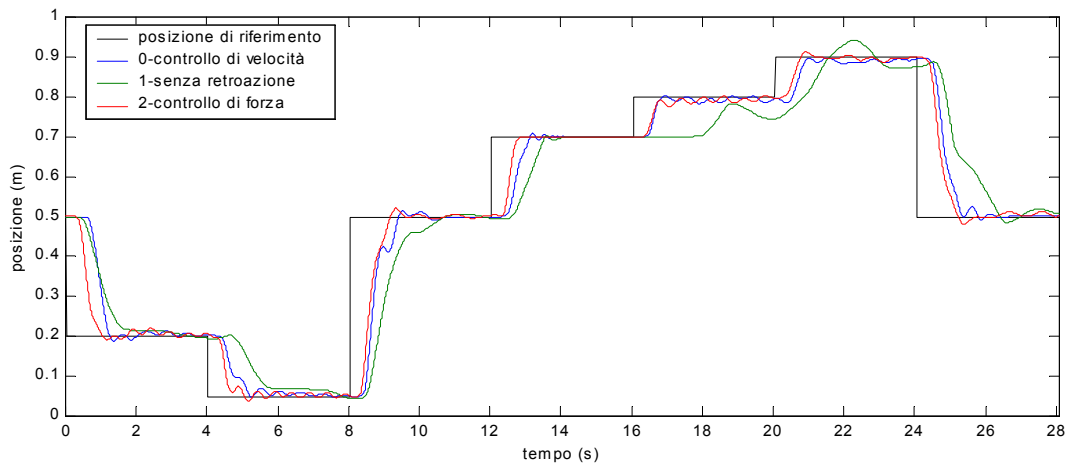


Figura 84: confronto nell'inseguimento di posizione con i 3 metodi di controllo

Il confronto delle curve stabilisce un vantaggio nel tempo di risposta per il controllo di posizione, subito seguito dal controllo di velocità.

Il controllo senza retroazione risulta particolarmente carente nella fase di sollevamento del carico.

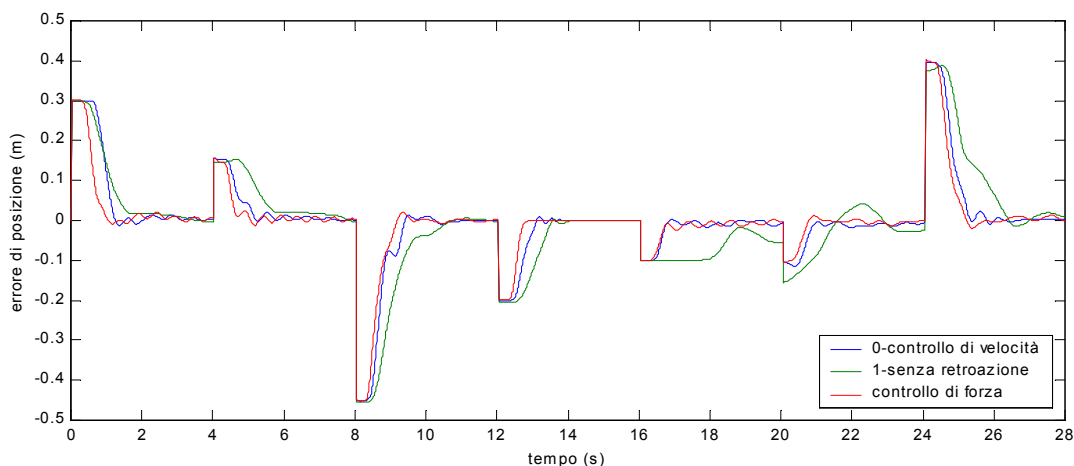


Figura 85: errore nell'inseguimento di posizione con i 3 metodi di controllo

La fig. 85 riporta il grafico degli errori della posizione rispetto al riferimento.

Nella tabella sottostante sono stati calcolati gli errori medi assoluti dei tre controlli durante l'esecuzione della prova, espressi in percentuale sull'intera corsa:

Modalità di funzionamento	Errore medio assoluto (%)	Variazione %
0 - Controllo di velocità (JoyForce)	5.06 %	-36.9 %
1 – Controllo senza retroazione	8.02 %	(valore max.)
2 – Controllo di forza (JoyForce)	3.98 %	-50.4 %

Anche in questo caso i due sistemi retroazionati hanno fatto registrare un errore medio assoluto considerevolmente minore del sistema non retroazionato, con un vantaggio del controllo di forza.

Cap. 4: SVILUPPI FUTURI

In questo progetto è stato definito un nuovo sistema di controllo, è stato realizzato un prototipo del joystick di comando, e lo si è provato in un ambiente simulato al computer.

Lo sviluppo futuro dovrebbe prevedere, come prima necessità, l'applicazione del joystick in un sistema fisico, che potrebbe essere rappresentato da un banco di prova oleodinamico, dotato dei sensori di posizione e forza richiesti dal sistema.

Si avrebbe il duplice risultato di eliminare il ritardo nella retroazione di 0.02 secondi, e di ottenere un riscontro reale delle potenzialità del sistema.

La fase successiva potrebbe essere quella di un miglioramento del joystick, alla luce dell'esperienza maturata con la costruzione del prototipo, in particolare sui seguenti punti:

- La parte di controllo del joystick, divisa in più moduli per facilitare la programmazione del microcontrollore ed eventuali modifiche, dovrebbe essere raggruppata su un'unica scheda con una sorgente comune di alimentazione, al posto degli attuali 3 alimentatori separati. In questo modo la parte analogica potrebbe essere debitamente schermata nei confronti della parte digitale, così da attenuare i disturbi elettromagnetici ed ottimizzare il funzionamento del convertitore analogico-digitale. Sempre in questo ambito, dovrebbe essere rivista la procedura di accensione del sistema e dovrebbe essere introdotta la gestione del funzionamento in emergenza.
- La parte meccanica del prototipo realizzato ha dimostrato i maggiori limiti nella presenza della trasmissione ad ingranaggi: sulla leva del joystick è infatti presente un certo gioco che, in caso di applicazioni diverse dal comando di cilindri idraulici, potrebbe compromettere la sensibilità. Un primo intervento per limitarne l'entità potrebbe essere quello di recuperare il gioco avvalendosi di un freno meccanico e programmando il microcontrollore di conseguenza: la presenza di un attrito statico sull'albero di uscita del motoriduttore consentirebbe di rilevare, con il torsionmetro, l'inversione della forza applicata alla leva, e di compensare, via software, il conteggio della posizione di riferimento. Un intervento più radicale, che risolverebbe anche il problema della disomogeneità di funzionamento degli ingranaggi, sarebbe quello di impiegare un riduttore armonico di precisione, oppure una cinghia dentata.
- In alternativa al microcontrollore Atmel AT90S4434, per aumentare la precisione e la complessità del controllo, si potrebbe utilizzarne uno a 16 bit, con un set di istruzioni più esteso che includa la moltiplicazione. Sarebbe auspicabile, inoltre, poter avere a disposizione un'uscita PWM con una maggiore risoluzione e con una frequenza superiore ai 22 kHz, in modo da eliminare ogni disturbo acustico.

JoyForce, proprio per essere un sistema estremamente flessibile, gestito da un microcontrollore riprogrammabile in-system, potrebbe essere sviluppato come joystick destinato ad un impiego generico, adattabile alla singola esigenza dall'utente finale. La sua caratteristica, infatti, è di costituire un'unità autonoma, applicabile ad una svariata schiera di situazioni, e di essere scarsamente influenzato dai parametri del sistema controllato, rapporto di scala compreso.

Dall'altra parte, invece, potrebbe essere ottimizzato per alcuni impieghi specifici, esaltando, in ogni ambito, alcune sue peculiarità.

Applicato in una macchina movimento terra, per esempio, tutto il sistema dovrebbe essere volto a poter sviluppare forze elevate, e la sensibilità potrebbe essere in parte sacrificata a vantaggio di un controllo più stabile.

L'impiego nella telemanipolazione e nei robot subacquei richiederebbe, invece, come principale prerogativa, la capacità di restituire all'operatore le sensazioni tattili del lavoro svolto dall'end-effector: il parametro interessato, in questo caso, sarebbe la tempestività di risposta del joystick.

Un'ultima ipotesi potrebbe essere quella di un utilizzo nelle applicazioni medicali, in cui si voglia sostituire, per motivi di precisione, l'intervento diretto della mano del chirurgo con un robot comandato a distanza. Il joystick ad un asse, allora, potrebbe essere parte di un sistema più complesso al quale sarebbe delegato il compito di coordinare i movimenti.

In neurochirurgia, ad esempio, dove la delicatezza della zona di intervento pone il massimo problema, un sistema di posizionamento potrebbe guidare l'esatta traiettoria dello strumento chirurgico, mentre il joystick potrebbe amplificare la precisione del movimento attivo, restituendo al chirurgo la sensazione di resistenza esercitata dai tessuti.

Un'applicazione così specifica richiederebbe una approfondita rivisitazione dello studio presentato, per poter garantire gli standard di precisione ed affidabilità richiesti.

CONCLUSIONI

Questa trattazione si è occupata di migliorare il controllo di un generico organo meccanico da parte di un operatore, indipendentemente dalla dimensione dello spazio di lavoro in cui opera.

Il presupposto di partenza è stato quello della necessità di riuscire a restituire all'operatore l'effettiva sensibilità dell'azione che sta compiendo.

A questo scopo sono stati analizzati i sistemi con retroazione già esistenti e si è passati allo sviluppo di un nuovo sistema, in cui, per restituire una sensazione naturale, ci fosse corrispondenza tra posizione e forza applicate al joystick, rispetto alle stesse grandezze rilevate sull'attuatore.

Sono stati studiati due modelli diversi, applicabili, il primo, ad un attuatore controllato in velocità, ed il secondo ad un attuatore controllato in forza.

Per ciascuno dei due modelli sono state effettuate delle simulazioni con Simulink di Matlab e, dopo aver risolto il problema dell'oscillazione dovuta all'introduzione di un tempo di ritardo nell'anello di retroazione, sono stati ottenuti due modelli stabili nei confronti di un ingresso a gradino.

È stato realizzato un prototipo di un joystick ad un solo asse, mentre l'ambiente di lavoro è stato implementato in un programma di simulazione al PC.

Nel joystick e nel programma di simulazione sono state previste le due modalità di funzionamento inizialmente modellizzate, in controllo di velocità e di forza, alla quale è stata aggiunta un'ulteriore modalità, senza retroazione, con cui compararle.

Sono state stabilite tre prove diverse, ciascuna atta a far risaltare una diversa caratteristica del controllo, con le quali è stato effettuato il confronto tra le tre modalità di funzionamento.

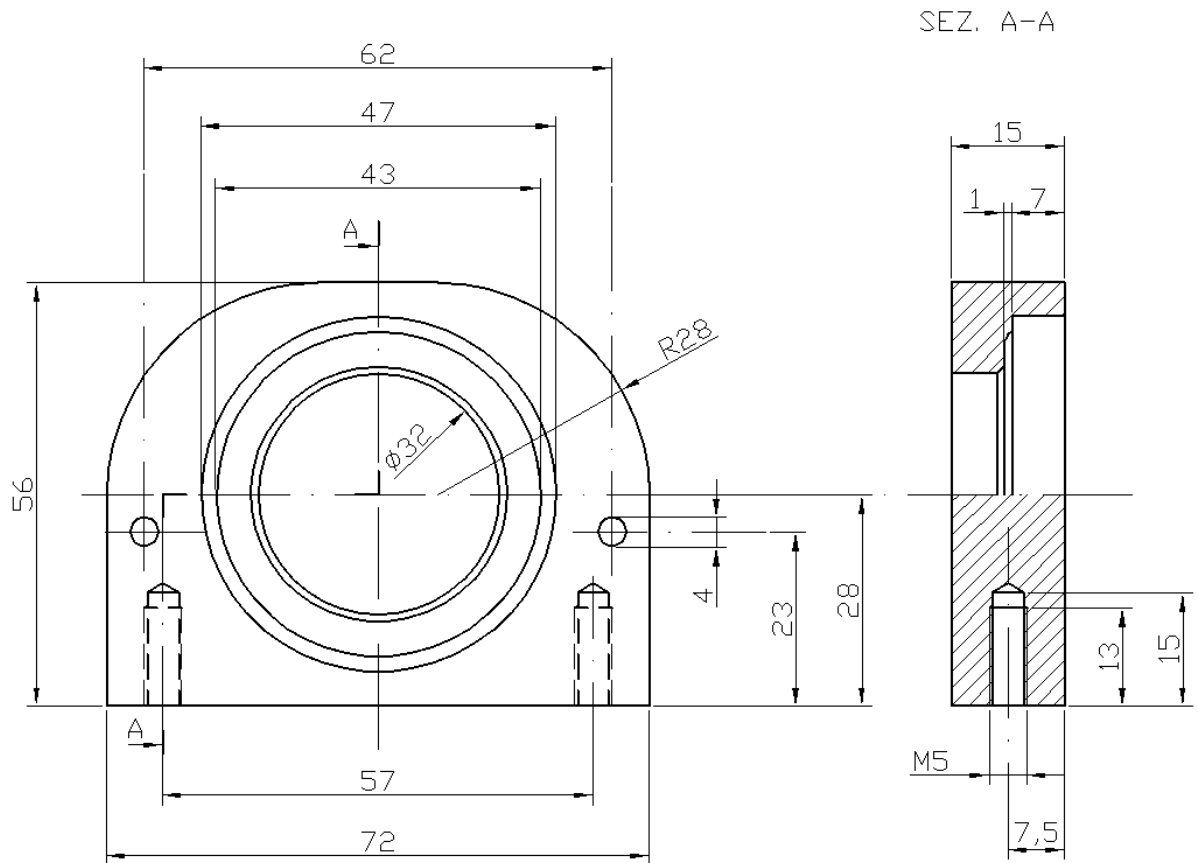
La prima prova, in inseguimento di traiettoria, è stata concepita per verificare l'attinenza ad impostare i comandi di velocità e di posizione desiderati dall'operatore.

La valutazione è avvenuta sull'errore medio assoluto rispetto alla traiettoria di riferimento: nel controllo di forza si è avuto un errore inferiore del 57.9 % rispetto a quello registrato senza retroazione, mentre nel controllo di velocità si è avuto un miglioramento del 51.3 %.

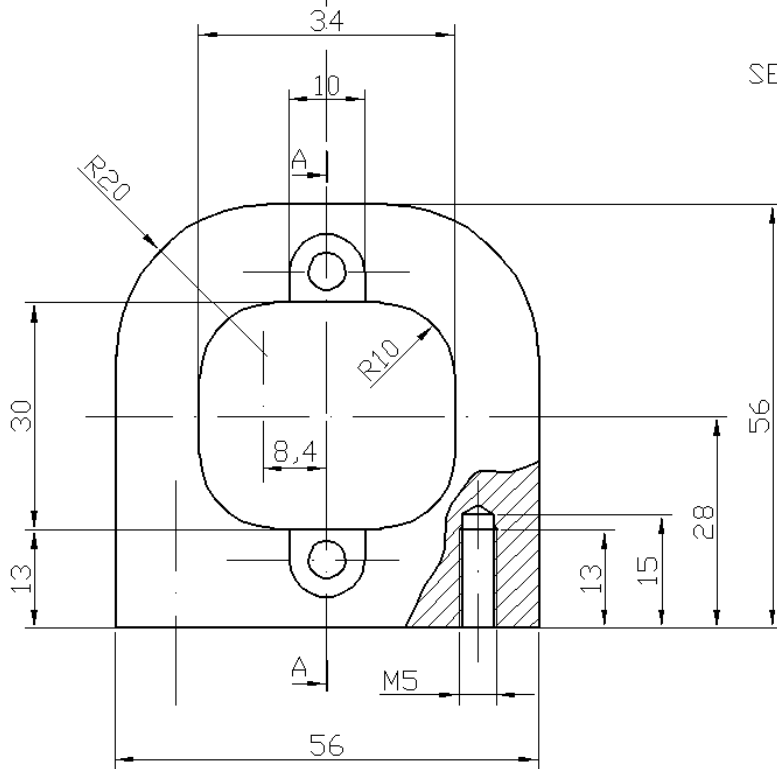
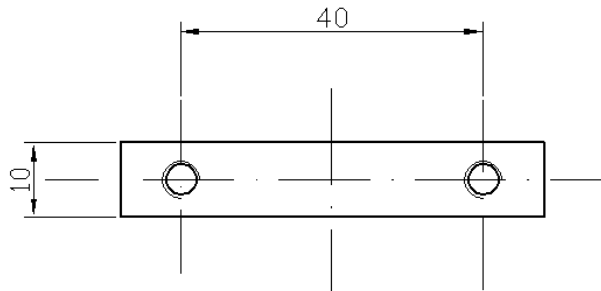
Nella seconda prova è stata misurata la velocità di esecuzione di un compito prefissato, consistente nel raggiungimento statico di 6 posizioni successive. La media di 5 prove ha rivelato un risparmio di tempo del 38.1 % per il controllo di velocità, e del 37.1 % per il controllo di forza.

Infine, la terza prova, simile alla prima, è stata utilizzata per valutare la tempestività nel raggiungere la posizione e a mantenerla stabilmente. In questo caso è stato registrato un miglioramento sull'errore medio assoluto del 50.4 % per il controllo di forza, e del 36.9 % per il controllo di velocità, nei confronti del controllo senza retroazione.

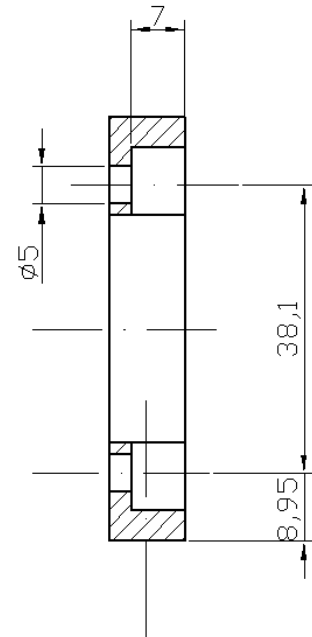
APPENDICE A: Disegni costruttivi



D.I.M.E.G. Università di Padova	Scala 1:1	
Materiale: Alluminio	PORTA CUSCINETTO	
Antoniazzi Fabio		



SEZ. A-A



D.I.M.E.G. Università di Padova

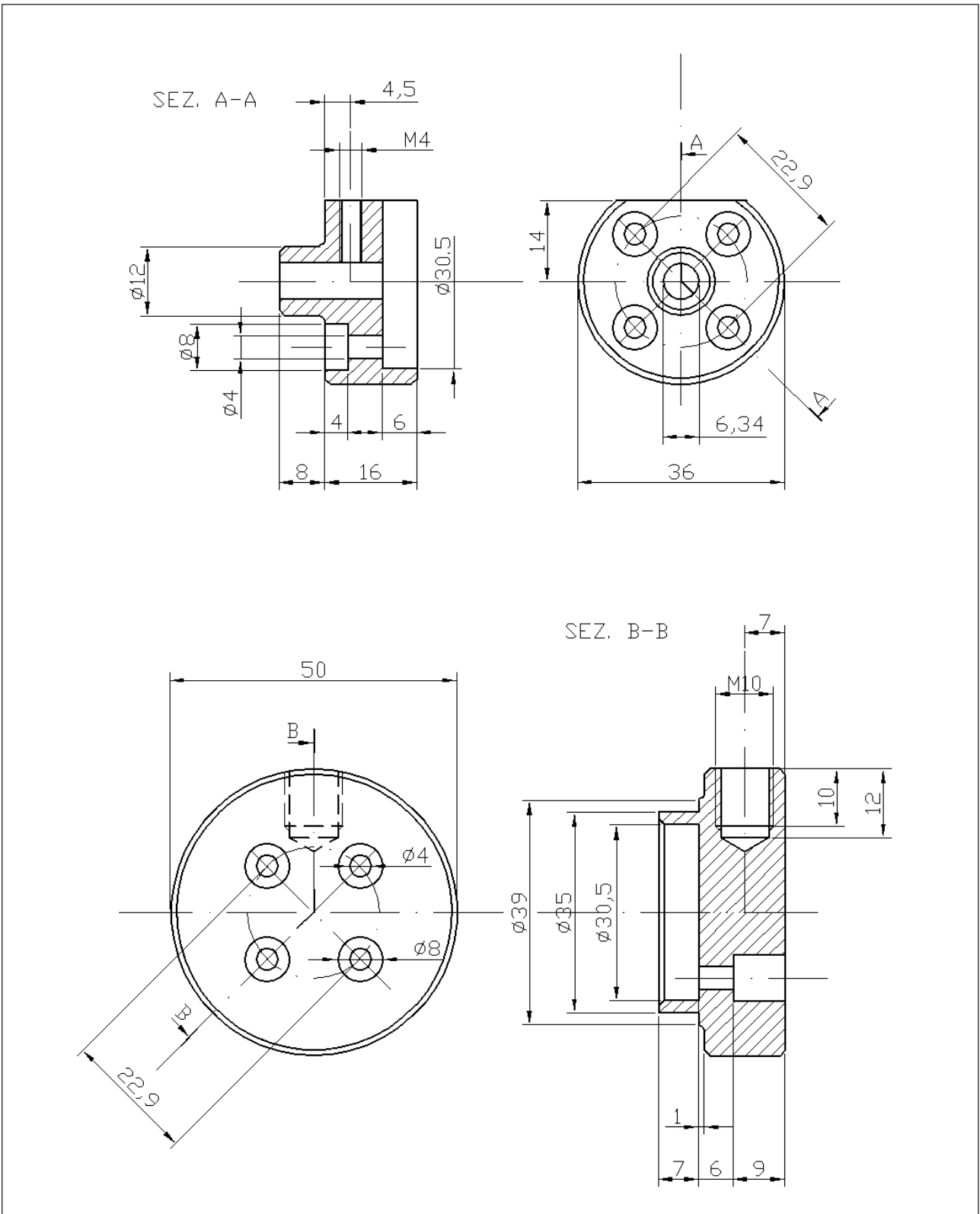
Scala 1:1

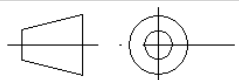


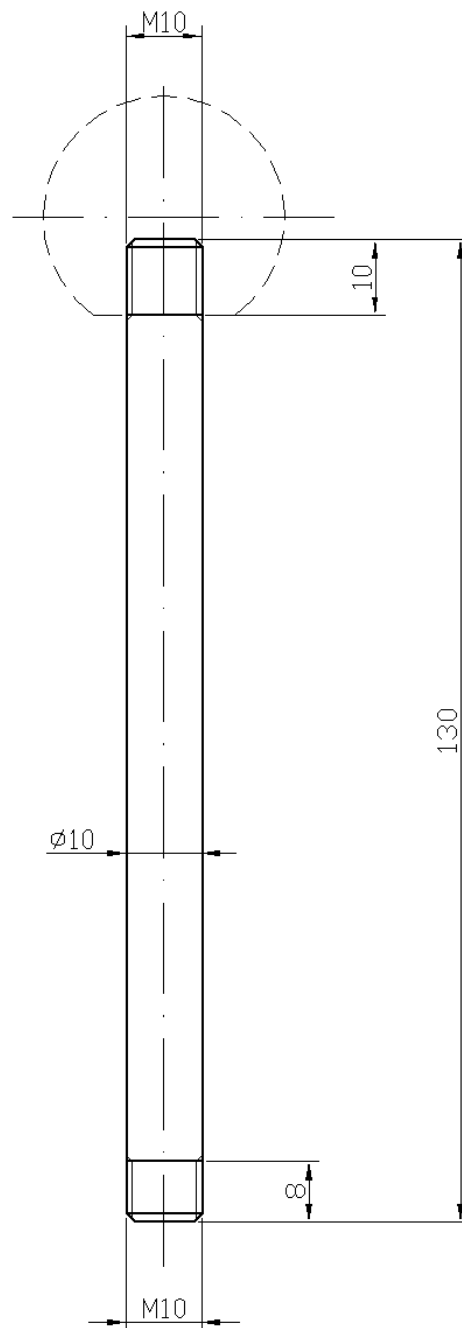
Materiale: Alluminio

FLANGIA MOTORE

Antoniazzi Fabio



D.I.M.E.G. Università di Padova	Scala 1:1	
Materiale: Alluminio	BICCHIERI DESTRO E SINISTRO	
Antoniuzzi Fabio		



D.I.M.E.G. Università di Padova

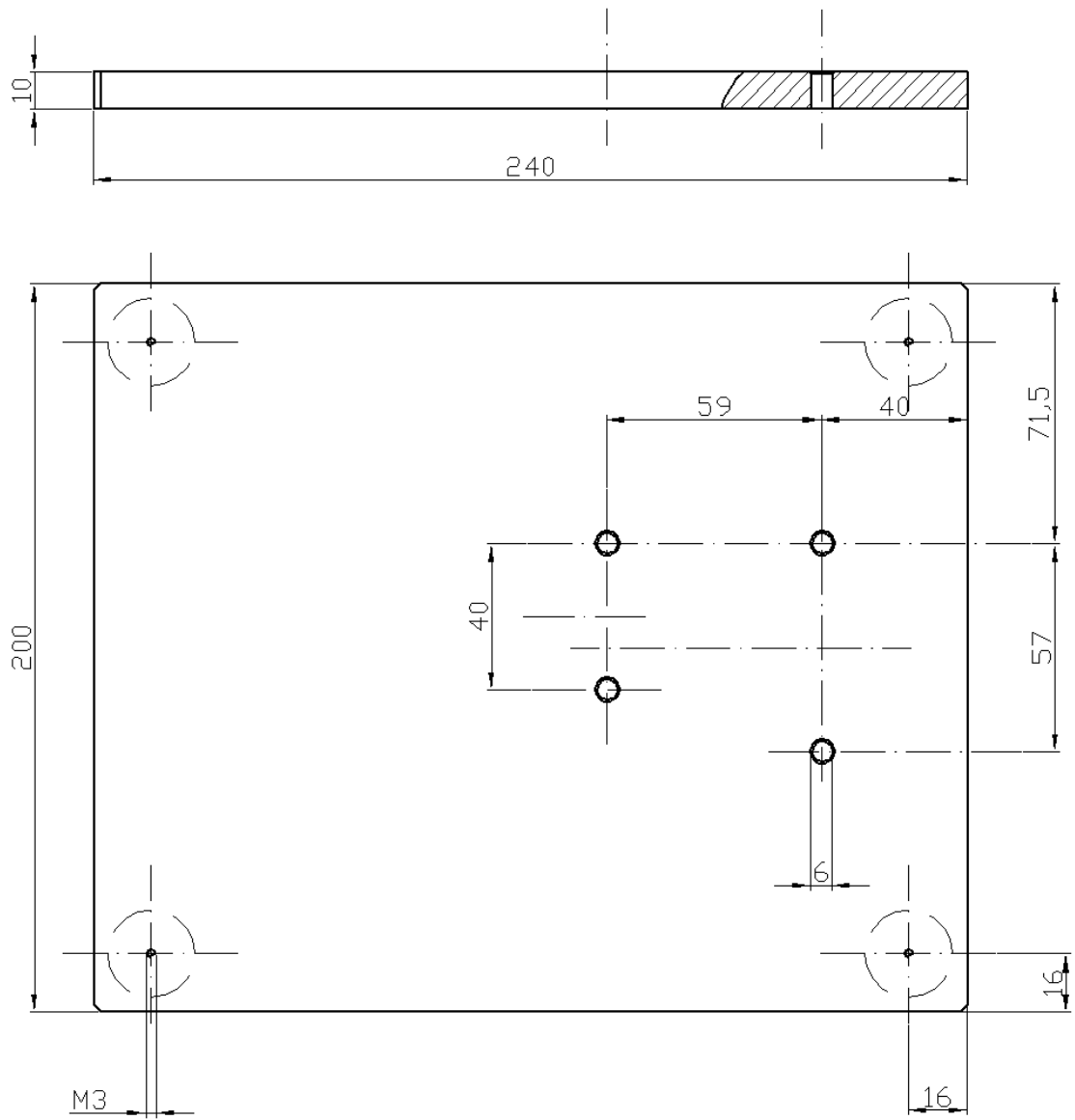
Scala 1:1



Materiale: Alluminio

Antoniazzi Fabio

LEVA DEL JOYSTICK



D.I.M.E.G. Università di Padova

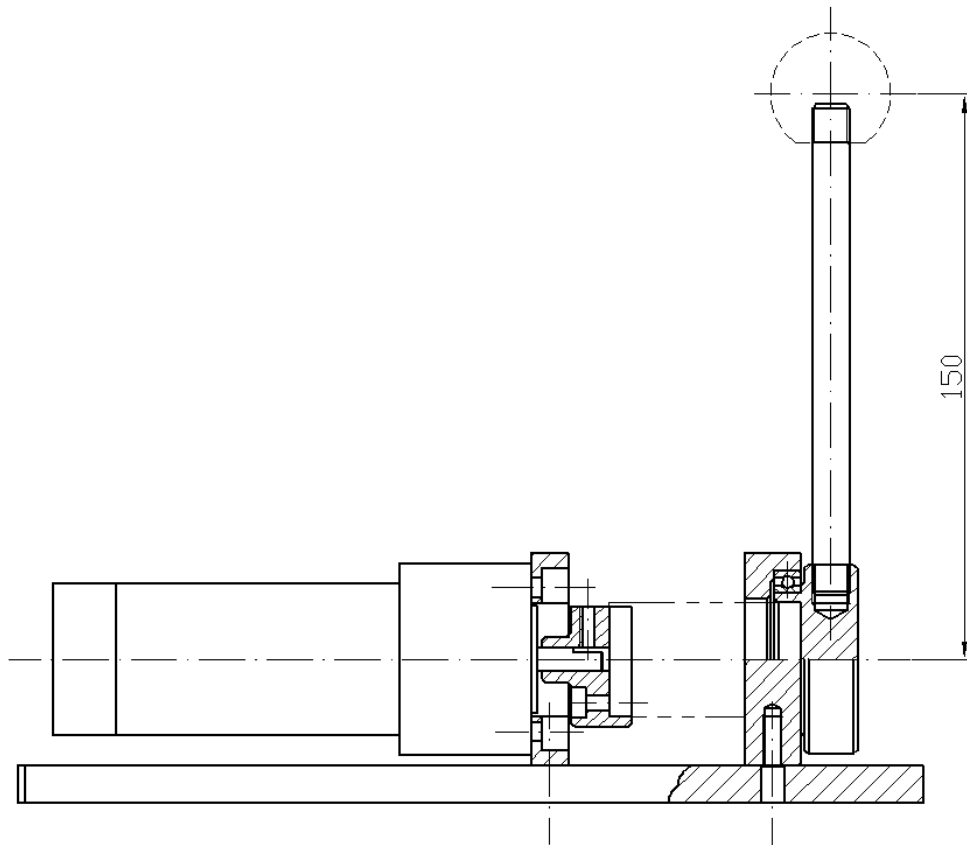
Scala 1:2



Materiale: Alluminio

PIASTRA DI BASE

Antoniazzi Fabio



D.I.M.E.G. Università di Padova

Scala 1:2



Materiale: Alluminio

COMPLESSIVO JOYSTICK

Antoniazzi Fabio

APPENDICE B: Codice assembler per AT90S4434


```

; PROGETTO: JOYFORCE
; Autore: Fabio Antoniazzi
; Data: 14 Marzo 2001
; DIMEG
; Facoltà di Ingegneria Meccanica dell'Università di Padova
;*****

.include "4434def.inc"

;*****

RESET:
.def sregister = r0      ;status register per int
.def tmpi =r23          ;temporanea utilizzata negli int
.def ok =r16            ;serve per avere intervalli di tempo
                        ;costanti: viene settata dall'int OC2

.def temp =r17          ;temporanea
.def temp2 =r18         ;temporanea2
.def contal =r24        ;contatore a 16 bit (encoder)
.def contah = r25       ;occupa 2 byte(r24:r25)

.def a1l=r19            ;(a1) utilizzo nei calcoli
.def a1h=r20
.def a2l=r21            ;(a2)
.def a2h=r22
.def a3l=r26            ;(a3) sarebbe il registro X
.def a3h=r27           ;che non uso per indirizzare

;il registro Z è dedicato all'indirizzamento da programma,
;il registro Y è dedicato agli interrupt.

;*****

.DSEG
  chnum: .byte 1      ;canale analogico selezionato
  ctors: .byte 2      ;coppia torsiometro (3 var consecutive)
  posa: .byte 2       ;posizione attuatore
  vela: .byte 2       ;velocità attuatore
  ctorsoff:.byte 2    ;valore di offset per il torsiometro

  posout:.byte 2
  oldposout:.byte 2
  deltatacche:.byte 2
  olderr:.byte 2
  oldposa:.byte 2

  posaqq:.byte 2      ;somma di 44 valori di posa
  tabposa:.byte 88    ;44*2
  indposa:.byte 2     ;indirizzo tabposa
  tors16: .byte 2     ;somma di 16 valori di t1024
  tabtors: .byte 32   ;16*2
  indtors: .byte 2    ;indirizzo tabtors
  Esomma:.byte 2
  tabE:.byte 32       ;32 di 1 byte
  indE:.byte 2
  indstep: .byte 1    ;lettura di posa ogni 6 cicli

;*****

.CSEG

.org 0x00              ;fissa l'origine del programma
    rjmp main

```

```

.org int0addr          ;fissa la chiamata alla routine di int
    rjmp  chAup

.org int1addr
    rjmp  chAdown

.org OC2addr
    rjmp  daiok

.org ADCCaddr          ;interrupt di avvenuta conversione
    rjmp  adcint

;*****
;   COSTANTI
.equ  posoff=5526
.equ  contacentro=16768
.equ  etlmax=20
.equ  Eoff=127
.equ  Emax=255

;*****
;interrupt di avvenuta conversione
adcint:  in   sregister,SREG    ;salva lo status register
        lds  tmpi,chnum
        ldi  YL,low(ctors)
        ldi  YH,high(ctors)
        add  YL,tmpi
        add  YL,tmpi
        in   tmpi, ADCL        ;salva il dato dell'ADC nella
        std  Y+1,tmpi         ;variabile passata
        in   tmpi, ADCH        ;la parte bassa si trova a destra
        st   Y, tmpi          ;(indirizzo successivo)

        lds  tmpi,chnum
        inc  tmpi
        cpi  tmpi,3
        brne seguente
        ldi  tmpi,0
seguinte: out  ADMUX,tmpi      ;setta il canale successivo
        sts  chnum,tmpi      ;salva la variabile
        ldi  tmpi,0b11001111 ;Start conversion
        out  ADCSR,tmpi
        out  SREG,sregister   ;ripristina lo status register
        reti                 ;ritorna dall'interrupt

;-----

daiok:   ldi  ok,1            ;routine dell'int OC2
        reti

;-----

chAup:   in   tmpi,PIND        ;routine dell'int INT0 (ext)
        in   sregister,SREG    ;salva lo status register
        bst  tmpi,PD1         ;carica nel T flag il bit PD1
        brtc skip1           ;se T=0 (PD1=0) skip
        sbiw contal,1
ret1:    out  SREG,sregister   ;ripristina lo status register
        reti                 ;ritorna dall'interrupt

skip1:   adiw contal,1        ;incrementa il contatore a 16bit
        rjmp ret1

;-----

```

```

chAdown: in  tmpi,PIND      ;routine dell'INT1 (ext)
         in  sregister,SREG ;salva lo status register
         bst tmpi,PD1
         brts skip1
         sbiw contal,1     ;decrementa il contatore a 16bit
         rjmp ret1

;***** MAIN *****

main: ldi temp,low(RAMEND)
      out SPL,temp
      ldi temp,high(RAMEND)
      out SPH,temp      ;init Stack Pointer

;---porte
      ser temp          ;porta temp a $FF
      out DDRB,temp     ;porta B abilitata in uscita
      out DDRC,temp     ;porta C abilitata in uscita
      ldi temp,0b10010000 ;porta D: PD7:(?out),
      out DDRD,temp     ;PD65:Mode in PD4:PWM ,PD23:ChA,
                        ;PD1:ChB PD0:Set mode
      ldi temp,(1<<INT0)+(1<<INT1)
      out GIMSK,temp     ;abilita gli int 0 e 1 ext
      ldi temp,(1<<ISC01)+(1<<ISC00)+(1<<ISC11)
      out MCUCR,temp     ;int0 in up e int1 in down

      ldi temp,(1<<INTF0)+(1<<INTF1) ;mette a zero i flag
      out GIFR,temp     ;degli int ext

;---timer
      clr temp          ;timer 2
      out TCNT2,temp    ;resetta il contatore
      ldi temp,0b00001010 ;disattiva il pin OC2 in uscita,
                        ;attiva il reset dopo il compare
                        ;e setta PCK2/8
      out TCCR2,temp
      ldi temp,0b10000000
      out TIMSK,temp    ;abilita int OCF2 (Output Compare Flag2)
      out TIFR,temp     ;mette a zero il flag OCF2
      ldi temp,100      ;tempo di 800 CK (100micros,10kHz)
      out OCR2,temp     ;setta il registro da comparare

;---ADC
      ldi temp, 0
      sts chnum, temp   ;seleziona il canale 0
      out ADMUX,temp    ;setta il canale 0 (PD0=torsiometro)
      ldi temp,0b11001110 ;Abilita ADC, Start conversion(dummy)
      out ADCSR,temp    ;Single, Int flag, Int enable,
                        ;Prescaler (125KHz)

;---PWM
      ldi temp,0b00110001 ;init invert PWM (8 bit, PD4)
      out TCCR1A,temp
      ldi temp,high(127)
      out OCR1BH,temp    ;PWM=50%
      ldi temp,low(127)
      out OCR1BL,temp
      ldi temp,0b00000001 ;prescaler=1 (f=15.7kHz)
      out TCCR1B,temp

;inizia var
      ldi temp,0        ;***** inizializza variabili
      sts olderr+1,temp ;olderr=0
      sts olderr,temp
      sts indstep,temp  ;indstep=0

```

```

ldi a1,low(511)
ldi ah,high(511)
sts oldposa+1,a1      ;salvo oldposa=511
sts oldposa,ah

ldi contal,low(contacentro)      ;inizializza il contatore
ldi contah,high(contacentro)    ;sulla pos centrale
sts oldposout+1,contal          ;oldpososout=contacentro
sts oldposout,contah
ldi ok,0

;inizializzo tabposa=0
ldi a1,0                    ;(a1)posa=0
ldi ah,0
sts posaqg+1,a1            ;salvo posa44=posa=0
sts posaqg,ah

ldi ZL,low(tabposa)        ;(Z)carica indirizzo 0 tab
ldi ZH,high(tabposa)
looptab: st  Z+,ah          ;salva il dato e incrementa
st  Z+,a1                 ;parte alta e poi bassa
cpi ZL,low(tabposa+88)    ;confronto con la fine tab
ldi temp,high(tabposa+88)
cpc ZH,temp
brne looptab             ;se non è arrivato alla fine continua

ldi ZL,low(tabposa)        ;(Z)carica indirizzo 0 tab
ldi ZH,high(tabposa)
sts indposa+1,ZL          ;salva indirizzo 0 tab
sts indposa,ZH           ;in indposa
;fine inizializzazione tabposa

;inizializzo tabE=0
ldi a1,0                    ;(a1)E256=0
ldi ah,0
sts Esomma+1,a1          ;salvo Esomma=E256=0
sts Esomma,ah

ldi ZL,low(tabE)          ;(Z)carica indirizzo 0 tab
ldi ZH,high(tabE)
looptab2:st  Z+,a1         ;salva il dato e incrementa (8bit)
cpi ZL,low(tabE+32)      ;confronto con la fine tab
ldi temp,high(tabE+32)
cpc ZH,temp
brne looptab2           ;se non è arrivato alla fine continua

ldi ZL,low(tabE)          ;(Z)carica indirizzo 0 tab
ldi ZH,high(tabE)
sts inde+1,ZL            ;salva indirizzo 0 tab
sts inde,ZH             ;in inde
;fine inizializzazione tabE

;inizializzo tabtors
ldi a1,low(8176)          ;(a1)=511*16
ldi ah,high(8176)
sts tors16+1,a1          ;salvo tors16=8176
sts tors16,ah
ldi a1,low(511)          ;(a1)t1024=511
ldi ah,high(511)

ldi ZL,low(tabtors)      ;(Z)carica indirizzo 0 tab
ldi ZH,high(tabtors)

```



```

looptab3:st  Z+,alh          ;salva il dato e incrementa
           st  Z+,alh          ;parte alta e poi bassa
           cpi ZL,low(tabtors+32) ;confronto con la fine tab
           ldi temp,high(tabtors+32)
           cpc ZH,temp
           brne looptab3      ;se non è arrivato alla fine continua

           ldi ZL,low(tabtors) ;(Z)carica indirizzo 0 tab
           ldi ZH,high(tabtors)
           sts indtors+1,ZL    ;salva indirizzo 0 tab
           sts indtors,ZH     ;in indtors
           ;fine inizializzazione tabtors

           sei                  ;abilita gli interrupt

;-----serve per memorizzare i valori di offset
           ldi a2l,0           ;(a2)=0
           ldi a2h,0
ritinizio:cpi  ok,1
           brne  ritinizio    ;loop di attesa
           ldi  ok,0
           subi a2l,low(-1)    ;(a2)=(a2)+1
           sbci a2h,high(-1)
           cpi  a2l,low(10000) ;1 secondo
           ldi  temp,high(10000)
           cpc  a2h,temp
           brne ritinizio

           cli
           lds a1l,ctors+1    ;(a1)=ctors
           lds a1h,ctors
           sei
           subi a1l,low(511)  ;(a1)ctorsoff=ctors-511
           sbci a1h,high(511)
           sts  ctorsoff+1,a1l ;salva ctorsoff
           sts  ctorsoff,a1h

;***** INIZIO LOOP *****
loop:  cpi  ok,1              ;attende l'interrupt
       brne loop            ;del timer
       ldi  ok,0

       ;legge la modalità
       in  temp,PIND         ;legge mode sui bit 6,5
       swap temp
       lsr temp
       com temp              ;i led sono invertiti
       andi temp, 0b00000011 ;maschera
       cpi  temp,0
       brne else1
       rjmp mode0
else1: cpi  temp,1
       brne else2
       rjmp mode1
else2: cpi  temp,2
       brne else3
       rjmp mode2
else3: rjmp fermo

```

```

;***** MODE 0
mode0:
;tabtors
  lds a2l,tors16+1 ;(a2)=tors16 precedente
  lds a2h,tors16
  lds ZL,indtors+1 ;carica l'indirizzo dell'indice
  lds ZH,indtors
  ldd a1l,Z+1 ;(a1)=oldtors
  ld ah,Z
  sub a2l,a1l ;(a2)tors16=tors16-oldposa
  sbc a2h,alh

  cli
  lds a1l,ctors+1 ;(a1)=ctors
  lds ah,ctors
  sei
  lds a3l,ctorsoff+1 ;(a3)=ctorsoff
  lds a3h,ctorsoff ;è la differenza da 511
  sub a1l,a3l ;(a1)t1024=ctors-ctorsoff
  sbc ah,a3h ; (è in nuovo valore)

  add a2l,a1l ;(a2)tors16=tors16+t1024
  adc a2h,alh
  sts tors16+1,a2l ;salva new tors16
  sts tors16,a2h

  st Z+,alh ;salva il dato e incrementa
  st Z+,a1l ;parte alta e poi bassa
  cpi ZL,low(tabtors+32) ;confronto con la fine tab
  ldi temp,high(tabtors+32)
  cpc ZH,temp
  brne conttors ;se non è arrivato alla fine continua
  ldi ZL,low(tabtors) ;(Z)carica indirizzo 0 tab
  ldi ZH,high(tabtors)
conttors:sts indtors+1,ZL ;salva l'indice
  sts indtors,ZH

  lsr a2h ;1 (a2)tors4=tors16/4 (unsigned)
  ror a2l ;div 2 per 2 volte
  lsr a2h ;2
  ror a2l
;fine tabtors

;DAC
  ldi temp,0b01110000 ;metto CS=L
  out PORTC,temp ;(a2)=tors4
  out PORTB,a2l ;scrivo la parte bassa
  mov temp,a2h
  ori temp,0b01100000
  out PORTC,temp
  andi temp,0b11011111
  out PORTC,temp
  ori temp,0b10000000
  out PORTC,temp
  ori temp,0b00100000
  out PORTC,temp
;fine DAC

;-----calcoli
  cli
  sts posout+1,contal
  sts posout,contah ;salvo il valore del contatore
  sei

```

```

;tabposa
  lds a2l,posaqq+1      ;(a2)=posa44 precedente
  lds a2h,posaqq

  lds temp2,indstep
  subi temp2,(-1)      ;incrementa indstep
  cpi temp2,6          ;confronta con 6
  brne saltaposa      ;se non è uguale salta
  ldi temp2,0          ;mette temp=0

  lds ZL,indposa+1    ;carica l'indirizzo dell'indice
  lds ZH,indposa
  ldd a1l,Z+1          ;(a1)=oldposa
  ld  a1h,Z
  sub a2l,a1l          ;(a2)posa44=posa44-oldposa
  sbc a2h,a1h

  cli
  lds a1l,posa+1      ;carico il valore della posizione
  lds a1h,posa        ;(a1)=posa
  sei

  lds a3l,oldposa+1   ;(a3)=oldposa
  lds a3h,oldposa
  cp  a1l,a3l          ;compare posa oldposa
  cpc a1h,a3h
  breq fineif         ;se è uguale salta
  mov a1l,a3l          ;(a1)=oldposa
  mov a1h,a3h
  brlo negif          ;se è minore
  subi a1l,low(-1)    ;altrimenti vuol dire
  sbci a1h,high(-1)  ;che è maggiore
  rjmp fineif         ;(a1)posa=oldposa+1

negif:  subi a1l,low(1) ;(a1)posa=oldposa-1
        sbci a1h,high(1)

fineif: sts oldposa+1,a1l ;salvo il nuovo oldposa
        sts oldposa,a1h

        add a2l,a1l      ;(a2)posa44=posa44+posa
        adc a2h,a1h
        sts posaqq+1,a2l ;salva new posa44
        sts posaqq,a2h

        st  Z+,a1h      ;salva il dato e incrementa
        st  Z+,a1l      ;parte alta e poi bassa
        cpi ZL,low(tabposa+88) ;confronto con la fine tab
        ldi temp,high(tabposa+88)
        cpc ZH,temp
        brne continua  ;se non è arrivato alla fine continua
        ldi ZL,low(tabposa) ;(Z)carica indirizzo 0 tab
        ldi ZH,high(tabposa)
continua:sts indposa+1,ZL ;salva l'indice
          sts indposa,ZH
        ;fine tabposa

saltaposa: sts indstep,temp2 ;salva indstep

        lsr a2h          ;(a2)posa22=posa44/2
        ror a2l

        subi a2l,low(-posoff) ;sommo 5526

```

```

    sbci a2h,high(-posoff)
                                ;(a2)=posrif
    lds  a1l,posout+1           ;carico in a1 posout
    lds  a1h,posout
    sub  a2l,a1l                 ;(a2)err=posrif-posout
    sbc  a2h,a1h

    mov  a3l,a2l
    mov  a3h,a2h                 ;copio err in a3
    lds  a1l,olderr+1          ;carico olderr in a1
    lds  a1h,olderr
    sub  a2l,a1l                 ;(a2)deltaerr=err-olderr
    sbc  a2h,a1h
    sts  olderr+1,a3l           ;salvo olderr
    sts  olderr,a3h

    clt                          ;T=0 (se deltaerr è positivo)
    brpl positivo
    com  a2l                      ;(a2)abs(deltaerr)
    com  a2h
    subi a2l,low(-1)
    sbci a2h,high(-1)
    set                          ;T=1 (risultato negativo)

; multiplico (a1)Etl2=deltaerr*6
positivo:mov a1l,a2l             ;copio 110
          mov  a1h,a2h
          lsl  a1l                 ;shifto sx
          rol  a1h
          add  a1l,a2l             ;sommo
          adc  a1h,a2h
          lsl  a1l                 ;shifto sx
          rol  a1h
          ;fine moltiplicazione

          brtc prox0              ;se è positivo salta(T=0)
          com  a1l                 ;altrimenti nega
          com  a1h
          subi a1l,low(-1)
          sbci a1h,high(-1)

;divido
prox0:   asr  a3h                 ;1 (a3)Etl1=err/4
          ror  a3l                 ;div 2 volte per 2
          asr  a3h                 ;2
          ror  a3l
          ;fine divisione

          add  a1l,a3l             ;(a1)Etl=etl1+etl2
          adc  a1h,a3h

;sogliatura
          cpi  a1l,low(etlmax)
          ldi  temp,high(etlmax)
          cpc  a1h,temp
          brlt prox1              ;se è < salta
          ldi  a1l,low(etlmax)    ;altrimenti (a1)etl=etlmax
          ldi  a1h,high(etlmax)
          rjmp prox2
prox1:   cpi  a1l,low(-etlmax)
          ldi  temp,high(-etlmax)
          cpc  a1h,temp
          brge prox2              ;se è >= salta

```

```

        ldi a1l,low(-etlmax);altrimenti (a1)etl=-etlmax
        ldi a1h,high(-etlmax)
;fine sogliatura

prox2:  lds a2l,oldposout+1 ;carico oldposout in a2
        lds a2h,oldposout
        lds a3l,posout+1   ;carico posout in a3
        lds a3h,posout
        sts oldposout+1,a3l ;salvo olposout for next
        sts oldposout,a3h
        sub a3l,a2l         ;(a3)deltatacche=posout-oldposout
        sbc a3h,a2h

        clt                 ;T=0 (se positivo)
        brpl posit2
        com a3l              ;(a3) abs(deltatacche)
        com a3h
        subi a3l,low(-1)
        sbci a3h,high(-1)
        set                  ;T=1 (negativo)
;multiplica          (a2)Etw= deltatatacche*4
posit2:  mov a2l,a3l         ;copio      4 -> 100
        mov a2h,a3h
        lsl a2l              ;shifto sx
        rol a2h
        lsl a2l              ;shifto sx
        rol a2h
;fine multiplica

        brtc near           ;se è positivo salta
        com a2l              ;altrimenti nega
        com a2h
        subi a2l,low(-1)
        sbci a2h,high(-1)

near:    add a1l,a2l         ;(a1)Etlw=etl+Etw
        adc a1h,a2h

        subi a1l,low(-Eoff) ;(a1)E256=127+Etlw
        sbci a1h,high(-Eoff)

;sogliatura
        cpi a1l,0
        ldi temp,0
        cpc a1h,temp
        brge prox3         ;se è >= salta
        ldi a1l,0          ;altrimenti E256=0
        ldi a1h,0
        rjmp prox4
prox3:   cpi a1l,low(Emax)
        ldi temp,high(Emax)
        cpc a1h,temp
        brlt prox4         ;se è < salta
        ldi a1l,low(Emax)  ;altrimenti E256=255
        ldi a1h,high(Emax)
;fine sogliatura          (a1)=E256
prox4:   ;tabE
        lds a2l,Esomma+1   ;(a2)=Esomma
        lds a2h,Esomma
        lds ZL,indE+1      ;carica l'indice
        lds ZH,indE
        ld a3l,Z           ;(a3)= oldE256

```

```

sub a2l,a3l          ;(a2)=Esomma-oldeE256
sbci a2h,0          ;la parte alta è nulla
add a2l,a1l         ;(a2)=Esomma+E256
adc a2h,a1h
sts Esomma+1,a2l    ;salva Esomma
sts Esomma,a2h
st Z+,a1l          ;salva il dato(E256) e inc
cpi ZL,low(tabE+32) ;confronto con fine tab
ldi temp,high(tabE+32)
cpc ZH,temp
brne contE
ldi ZL,low(tabE)
ldi ZH,high(tabE)
contE: sts indE+1,ZL    ;salva l'indice
      sts indE,ZH

      lsr a2h          ;1 Emedio=Esomma/32
      ror a2l          ;div 2 per 5 volte
      lsr a2h          ;2
      ror a2l
      lsr a2h          ;3
      ror a2l
      lsr a2h          ;4
      ror a2l
      lsr a2h          ;5
      ror a2l
;finetabE

      out OCR1BH,a2h    ;aggiorna il PWM
      out OCR1BL,a2l
      rjmp loop

;***** MODE 1
model: nop
      cli
      mov a2l,contal    ;(a2)=posout
      mov a2h,contah
      sei
      nop
      sts posout+1,a2l  ;salva per dopo
      sts posout,a2h

      asr a2h          ;1 posout/8
      ror a2l
      asr a2h          ;2
      ror a2l
      asr a2h          ;3
      ror a2l

      subi a2l,low(48)  ;tolgo 48
      sbci a2h,high(48)
;DAC
      ldi temp,0b01110000 ;metto CS=L
      out PORTC,temp      ;(a2)=posout/8-48
      out PORTB,a2l      ;scrivo la parte bassa
      mov temp,a2h
      ori temp,0b01100000
      out PORTC,temp
      andi temp,0b11011111
      out PORTC,temp
      ori temp,0b10000000
      out PORTC,temp
      ori temp,0b00100000

```

```

    out  PORTC,temp
;fine DAC

;-----calcoli
    nop
    lds  a1l,posout+1    ; (a1)=posout
    lds  a1h,posout
    subi a1l,low(contacentro-512) ;pos0=posout-16768+512
    sbci a1h,high(contacentro-512)

    mov  a3l,a1l        ; (a3)=pos0
    mov  a3h,a1h

    asr  a1h            ;1    dE=pos0/1024
    ror  a1l
    asr  a1h            ;2    div2 x10
    ror  a1l
    asr  a1h            ;3
    ror  a1l
    asr  a1h            ;4
    ror  a1l
    asr  a1h            ;5
    ror  a1l
    asr  a1h            ;6
    ror  a1l
    asr  a1h            ;7
    ror  a1l
    asr  a1h            ;8
    ror  a1l
    asr  a1h            ;9
    ror  a1l
    asr  a1h            ;10
    ror  a1l

    cpi  a1l,0          ;dE=0 ?
    breq uguale         ;si trova nella zona morta
    brlt meno
    subi a1l,-4         ;somma 4
    rjmp lascia
meno:  subi a1l,4       ;sottrai 4
    rjmp lascia
;zona morta
uguale: mov a1l,a3l    ; (a1)=pos0
    mov  a1h,a3h
    subi a1l,low(-64)  ; (a1)pos0x=pos0+64
    sbci a1h,high(-64) ;
    asr  a1h            ;1    dEx=pos0x/128
    ror  a1l
    asr  a1h            ;2    div2 x7
    ror  a1l
    asr  a1h            ;3
    ror  a1l
    asr  a1h            ;4
    ror  a1l
    asr  a1h            ;5
    ror  a1l
    asr  a1h            ;6
    ror  a1l
    asr  a1h            ;7
    ror  a1l
    subi a1l,low(4)    ;dE=dEx-4
    sbci a1h,high(4)
;fine zona morta      (a1)=dE=(pos0+64)/128-4

```

```

lascia: com a1l                ;dE=-dE
        com a1h
        subi a1l,low(-1)
        sbci a1h,high(-1)
        lds a2l,oldposout+1;carico oldposout in a2
        lds a2h,oldposout
        lds a3l,posout+1    ;carico posout in a3
        lds a3h,posout
        sts oldposout+1,a3l;salvo olposout for next
        sts oldposout,a3h
        sub a3l,a2l          ;(a3)deltatacche=posout-oldposout
        sbc a3h,a2h

        clt                  ;T=0 (se positivo)
        brpl posit3
        com a3l              ;(a3) abs(deltatacche)
        com a3h
        subi a3l,low(-1)
        sbci a3h,high(-1)
        set                  ;T=1 (negativo)

        ;moltiplica          (a2)Etw= deltatatacche*6
posit3: mov a2l,a3l          ;copio      6 -> 110
        mov a2h,a3h
        lsl a2l              ;shifto sx
        rol a2h
        add a2l,a3l          ;sommo
        adc a2h,a3h
        lsl a2l              ;shifto sx
        rol a2h
        ;fine moltiplica

        brtc near2          ;se è positivo salta
        com a2l              ;altrimenti nega
        com a2h
        subi a2l,low(-1)
        sbci a2h,high(-1)

near2:  add a1l,a2l          ;(a1)Etlw=dE+Etw
        adc a1h,a2h

        subi a1l,low(-Eoff) ;E256=127+Etlw
        sbci a1h,high(-Eoff)

        ;sogliatura
        cpi a1l,low(0)
        ldi temp,high(0)
        cpc a1h,temp
        brge prox5          ;se è >= salta
        ldi a1l,low(0)      ;altrimenti E256=0
        ldi a1h,high(0)
        rjmp prox6

prox5:  cpi a1l,low(Emax)
        ldi temp,high(Emax)
        cpc a1h,temp
        brlt prox6          ;se è < salta
        ldi a1l,low(Emax)  ;altrimenti E256=255
        ldi a1h,high(Emax)
        ;fine sogliatura    (a1)=E256

```



```

prox6:
    ;tabE
    lds a2l,Esomma+1    ;(a2)=Esomma
    lds a2h,Esomma
    lds ZL,indE+1      ;carica l'indice
    lds ZH,indE
    ld a3l,Z           ;(a3)= oldE256
    sub a2l,a3l        ;(a2)=Esomma-oldE256
    sbci a2h,0         ;la parte alta è nulla
    add a2l,a1l        ;(a2)=Esomma+E256
    adc a2h,a1h
    sts Esomma+1,a2l   ;salva Esomma
    sts Esomma,a2h
    st Z+,a1l         ;salva il dato(E256) e inc
    cpi ZL,low(tabE+32) ;confronto con fine tab
    ldi temp,high(tabE+32)
    cpc ZH,temp
    brne contE2
    ldi ZL,low(tabE)
    ldi ZH,high(tabE)
contE2: sts indE+1,ZL   ;salva l'indice
        sts indE,ZH

        lsr a2h         ;1 Emedio=Esomma/32
        ror a2l         ;div 2 per 5 volte
        lsr a2h         ;2
        ror a2l
        lsr a2h         ;3
        ror a2l
        lsr a2h         ;4
        ror a2l
        lsr a2h         ;5
        ror a2l
    ;finetabE

    out OCR1BH,a2h    ;aggiorna il PWM
    out OCR1BL,a2l
    rjmp loop

;***** MODE 2
mode2:  nop
        rjmp mode0
        rjmp loop

;***** FINE MODE

fermo:  nop
        ldi a2l,low(127)    ;50%
        ldi a2h,high(127)
        out OCR1BH,a2h     ;aggiorna il PWM
        out OCR1BL,a2l
        rjmp loop

;***** FINE LOOP *****

```


APPENDICE C: Routine di Simulazione Visual Basic


```
Private Sub MuoviMode0(IN1 As Double)
```

```
Dim Oldpos As Double
```

```
Dim Oldvel As Double
```

```
Dim oldBlockv As Double
```

```
Dim oldBlockp As Double
```

```
Dim Acc As Double
```

```
Dim dx As Double
```

```
Dim springf As Double
```

```
Dim oldspringp As Double
```

```
Dim oldspringv As Double
```

```
Static OldIN As Double
```

```
Static Fblock As Double
```

```
Dim Vrif As Double
```

```
Dim Fvirt As Double
```

```
Const GainVirt = 10
```

```
Oldvel = Vel
```

```
Oldpos = Pos
```

```
oldBlockv = Blockv
```

```
oldBlockp = Blockp
```

```
oldspringp = Springp
```

```
oldspringv = Springv
```

```
'----- muovi block
```

```
Blockv = Blockv - g * 0.001
```

```
Blockp = Blockp + Blockv * 0.001
```

```
If Blockp < BlockX0 Then
```

```
Blockp = BlockX0
```

```
Blockv = 0
```

```
End If
```

```
'----- fine block
```

```
'-----spring
```

```
springf = -SpringK * (Springp - SpringX) - SpringC * Springv
```

```
Springv = Springv + springf / SpringM * 0.001
```

```
Springp = Springp + Springv * 0.001
```

```
If Springp < 0 Then
```

```
Springp = 0
```

```
Springv = 0
```

```
End If
```

```
'-----fine spring
```

```
Vrif = Vmax * (IN1 - Forza / Fmax)
```

```
If Reverse.value = False Then
```

```
    If (Sgn(Vrif) * Sgn(IN1)) < 1 Then Vrif = 0
```

```
    If (Sgn(Forza) * Sgn(IN1)) = -1 Then Vrif = Vmax * IN1
```

```
End If
```

```
'opzione irreversibile
```

```
'non concordi: il segnale e la vel elaborata
```

```
'discordi: ricalcolo la velocita trascurando la forza
```

```
Fvirt = GainVirt * (Vrif - Vel)
```

```
Vel = Vel + Fvirt * 0.001 'Mvirt=1
```

```

If Vel > Vmax Then Vel = Vmax
If Vel < -Vmax Then Vel = -Vmax
Pos = Pos + Vel * 0.001
If Pos < 0 Then
  Pos = 0
  Vel = 0
Elseif Pos > 1 Then
  Pos = 1
  Vel = 0
End If

```

```

Forza = 0
If Pos > Blockp Then
  Blockp = Pos
  Blockv = Vel
  Forza = BlockM * (g + (Blockv - oldBlockv) * 1000)
End If

```

```

If Pos < Springp Then
  Springp = Pos
  Springv = Vel
  Forza = SpringM * ((Springv - oldspringv) * 1000) + SpringC * Springv + SpringK * (Springp - SpringX)
End If

```

```

If Forza > Fmax Then Forza = Fmax
If Forza < -Fmax Then Forza = -Fmax

```

End Sub

```

Private Sub MuoviMode2(IN1 As Double)

```

```

  Dim Oldpos As Double
  Dim Oldvel As Double
  Dim oldBlockv As Double
  Dim Acc As Double
  Dim dx As Double
  Dim springf As Double
  Dim oldspringp As Double
  Dim oldspringv As Double
  Static OldIN As Double

```

```

  Oldvel = Vel
  Oldpos = Pos
  oldBlockv = Blockv
  oldspringp = Springp
  oldspringv = Springv
  '----- muovi block
  Blockv = Blockv - g * 0.001
  Blockp = Blockp + Blockv * 0.001
  If Blockp < BlockX0 Then
    Blockp = BlockX0

```

```

Blockv = 0
End If
'----- fine block

'-----spring
springf = -SpringK * (Springp - SpringX) - SpringC * Springv
Springv = Springv + springf / SpringM * 0.001
Springp = Springp + Springv * 0.001
If Springp < 0 Then
  Springp = 0
  Springv = 0
End If
'-----fine spring

If Abs(IN1 - OldIN) > 0.0001 Then
  If visc < 1 Then visc = visc + 0.1
Else
  If visc >= 0.3 Then visc = visc - 0.1
End If
OldIN = IN1

Forza = Fmax * (IN1 - visc * Vel)
Vel = Vel + Forza / Mass * 0.001
If Vel > Vmax Then Vel = Vmax
If Vel < -Vmax Then Vel = -Vmax
Pos = Pos + Vel * 0.001
If Pos < 0 Then
  Pos = 0
  Vel = 0
Elseif Pos > 1 Then
  Pos = 1
  Vel = 0
End If

If Pos > Blockp Then
  Acc = (Forza - BlockM * g) / (BlockM + Mass)
  Vel = (oldBlockv * BlockM + Mass * Oldvel) / (BlockM + Mass)
  Vel = Vel + Acc * 0.001
  If Vel > Vmax Then Vel = Vmax
  If Vel < -Vmax Then Vel = -Vmax
  Pos = Oldpos + Vel * 0.001
  If Pos < 0 Then
    Pos = 0
    Vel = 0
  Elseif Pos > 1 Then
    Pos = 1
    Vel = 0
  End If
  If Pos < BlockX0 Then
    Pos = BlockX0
    Vel = 0
  End If

```

```
End If
Blockp = Pos
Blockv = Vel
End If

If Pos < Springp Then
  Acc = (Forza - SpringK * (oldspringp - SpringX) - SpringC * oldspringv) / (SpringM + Mass)
  Vel = (oldspringv * SpringM + Mass * Oldvel) / (SpringM + Mass)
  Vel = Vel + Acc * 0.001
  If Vel > Vmax Then Vel = Vmax
  If Vel < -Vmax Then Vel = -Vmax
  Pos = Oldpos + Vel * 0.001
  If Pos < 0 Then
    Pos = 0
    Vel = 0
  Elself Pos > 1 Then
    Pos = 1
    Vel = 0
  End If
  Springp = Pos
  Springv = Vel
End If

End Sub
```


APPENDICE D: Costo dei componenti

ELENCO DEI PRINCIPALI FORNITORI E DEL COSTO DEI COMPONENTI UTILIZZATI NEL PROGETTO:

Fornitore: DSPM industria srl, Padova

Torsiometro Futek 5160	747.00 Euro
Condizionatore di segnale Futek JM-2	483.00 Euro
Alimentatore di rete	77.44 Euro

Fornitore: RS-Components Spa, Vimodrone (PD)

Alimentatore non stabilizzato E124	141.46 Euro
Driver motore LMD18200T	17.86 Euro
Convertitore digitale-analogico AD7396	16.31 Euro

Fornitore TECNOPOLIS, Padova

Microcontrollore Atmel AT90S4434	9.81 Euro
Starter kit Atmel STK200	93.96 Euro

Fornitore: Rappr. National Instruments per la provincia di Padova, Ing. Parere

Scheda acquisizione National Instruments PCI-6024E	860.00 Euro
Cavo di connessione	80.00 Euro
Scheda connettori	115.00 Euro

Fornitore: PITTMAN, Harleysville, PA19438 U.S.A.

Motoriduttore con encoder GM9236S027	190.00 Euro
--------------------------------------	-------------

Fornitori vari:

Componenti elettronici	30.00 Euro
Componenti meccanici	50.00 Euro

TOTALE:

2'911.84 Euro

BIBLIOGRAFIA

- [1] N.R: Parker, S.E. Salcudean and P.D: Lawrence, “**Application of force feedback to heavy duty hydraulic machines**”. Presented at the IEEE Robotica and Automation Conference, May 2, 1993: Atlanta, Georgia.
- [2] P.D. Lawrence, S.E. Salcudean, N. Sepehry, “**Coordinated and Force-Feedback Control of Hydraulic Excavators**”. Symposium on Experimental Robotica, ISER’95, June 30 – July 2, 1995, Stanford, California.
- [3] S. Tafazoli, P:D: Lawrence, S.E. Salcudean, D. Chan, S. Bachmann and C.W. de Silva, “**Parameter Estimation and Actuator Friction Analisis for a mini Excavator**”. University of British Columbia, Vancouver, Canada
- [4] S.E. Salcudean, S. Tafazoli, P:D: Lawrence, I. Chau, “**Impedence Control of a Teleoperated Mini Excavator**”. University of British Columbia, Vancouver, Canada
- [5] Robert L. Williams II, Jason M. Henry, Mark A. Murphy, “**Naturally-Transitioning Rate to Force Control in free and constrained motion**”. Journal of dynamic systems, measurement, and control. Transactions of the ASME, vol.121, No.3, pp.425-432, 1999.
- [6] D.E. Whitney, 1985, “**Historical Perspective and State of the Art in Robot Force Control**”, IEEE International Conference on Automation and Robotics, 262-268.
- [7] J.J. Craig, M. Raibert, 1981, “**Hybrid Position/Force Control of Manipulators**”, ASME Journal of Dynamic System, Measurement, and Control, 103(2): 126-133.
- [8] R. Colbaugh, H. Serali, K. Glass, 1993, “**Direct Adaptive Impedence Control of Robot Manipulators**”, Journal of Robotica System, 10(2): 217-248.
- [9] J.M. Hide, M.R. Cutcosky, 1994, “**Controlling Contact Transition**”, IEEE Control System Magazine, 14(1): 25-30.
- [10] B. Yao, M. Tomizuka, 1995, “**Adaptive Control of Robot Manipulators in Constrained Motion – Controller Design**”, ASME Journal of Dynamic System, Measurement and Control, 117(3): 320-328
- [11] M. Vokobratovic, R. Stojic, 1996, “**On Position/Force of Robot Interacting with Dynamic Environment in Cartesian Space**”, ASME ASME Journal of Dynamic System, Measurement, and Control, 118(1): 187-192.
- [12] T.J. Tarn, Y. Wu, N. Xi, A.Isidori, 1996, “**Force Regulation and Contact Transition Control**”, IEEE Control System Magazine, 16(1): 32-40.