

**UNIVERSITÀ DEGLI STUDI DI BRESCIA**

**FACOLTÀ DI INGEGNERIA**

CORSO DI LAUREA SPECIALISTICA IN INGEGNERIA INFORMATICA

CORSO DI ROBOTICA INDUSTRIALE



**Guida introduttiva all'utilizzo del robot Kawasaki RS03N**

Studente:

**Daniele Petre**

Matricola : **77686**

Professore:

**Prof. Riccardo Cassinis**

ANNO ACCADEMICO 2010/2011

## Indice

1	Introduzione.....	3
2	Il robot.....	4
2.1	Braccio.....	4
2.1.1	Sistemi di coordinate .....	5
2.2	Controller .....	8
2.3	Teach Pendant o Unità Portatile .....	9
3	Funzioni di base.....	14
3.1	Accensione .....	14
3.2	Spegnimento .....	14
3.3	Insegnamento .....	14
3.4	Esecuzione programmi.....	15
3.4.1	TP.....	15
3.4.2	AS .....	15
3.4.3	AS + TP.....	15
4	Insegnamento tramite Teach Pendant .....	17
4.1	Formato delle istruzioni .....	17
4.1.1	Campi .....	18
4.2	Programma di esempio con TP .....	19
4.3	Check del programma passo-passo .....	20
5	Insegnamento tramite console e linguaggio AS.....	22
5.1	Collegamento al controller via ethernet con KRterm .....	22
5.2	Linguaggio AS .....	25
5.2.1	Comandi principali .....	25
5.3	Programma di esempio con AS .....	28
6	Risorse per approfondire .....	31
7	Tabella delle figure.....	32

## 1 Introduzione

Questo documento mira ad introdurre gli interessati all'utilizzo del robot Kawasaki RS03N che si trova nel laboratorio di robotica dell'Università. In particolare, dopo brevi cenni riguardo alle caratteristiche del robot e del suo controller, vengono presentati gli strumenti di controllo e infine vengono proposti due piccoli tutorial di utilizzo pratico del robot. Il primo riguarda l'insegnamento di pose e piccoli programmi con l'impiego del teach pendant (o unità portatile) in teach mode. Mentre il secondo mira ad introdurre i rudimenti per la programmazione del robot a PC utilizzando il linguaggio AS (di proprietà Kawasaki). Si fa notare che le informazioni qui riportate sono volutamente non esaustive. Per maggiori informazioni, dettagli e procedure più avanzate è necessario fare riferimento al manuale operativo e al manuale del linguaggio AS che è possibile trovare in lingua, sia inglese sia italiana, in formato PDF sul PC portatile dedicato all'utilizzo del robot (se non si dovesse ritrovare chiedere al prof. Cassinis una copia). Nel medesimo formato e posizione è possibile trovare il manuale Antinfortunistica che presenta le basilari regole di sicurezza nell'utilizzo del robot. In questo documento, i riferimenti ai manuali sono alle versioni italiane se non diversamente specificato.

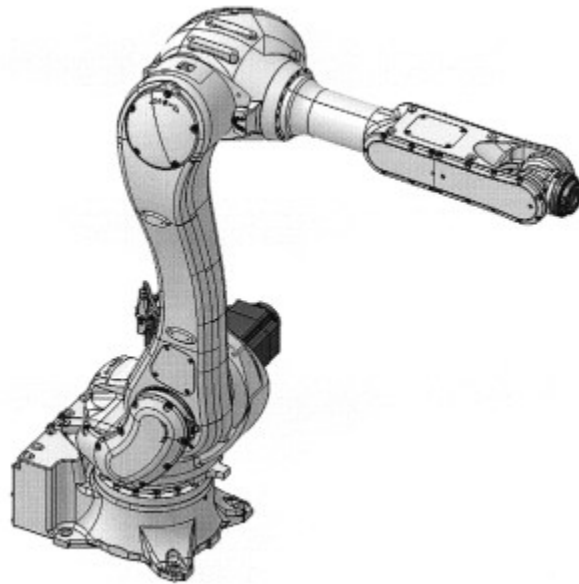


Figura 1: Robot Kawasaki serie RS

## 2 Il robot

Il nuovo robot Kawasaki entrato nella dotazione del laboratorio di robotica avanzata dell'Università di Brescia è un piccolo robot industriale fisso di tipo articolato.

Il robot è composto da tre parti fondamentali:

1. Braccio (Figura 1);
2. Controller (Figura 7);
3. Teach pendant (Figura 9).

In seguito vengono velocemente introdotte tutte e tre queste componenti.

### 2.1 Braccio

Il braccio rappresenta la parte più caratteristica di un robot industriale in quanto è quella che fisicamente si muove e interagisce direttamente con il mondo fisico eseguendo vari compiti (manipolazione, saldatura, verniciatura, ecc. ). Nel braccio RS03N il numero di giunti è pari a sei, così come i gradi di libertà. Non ci sono quindi particolari vincoli nelle posizioni, o pose, raggiungibili con il polso salvo l'estensione massima del braccio, come visibile in Figura 2.

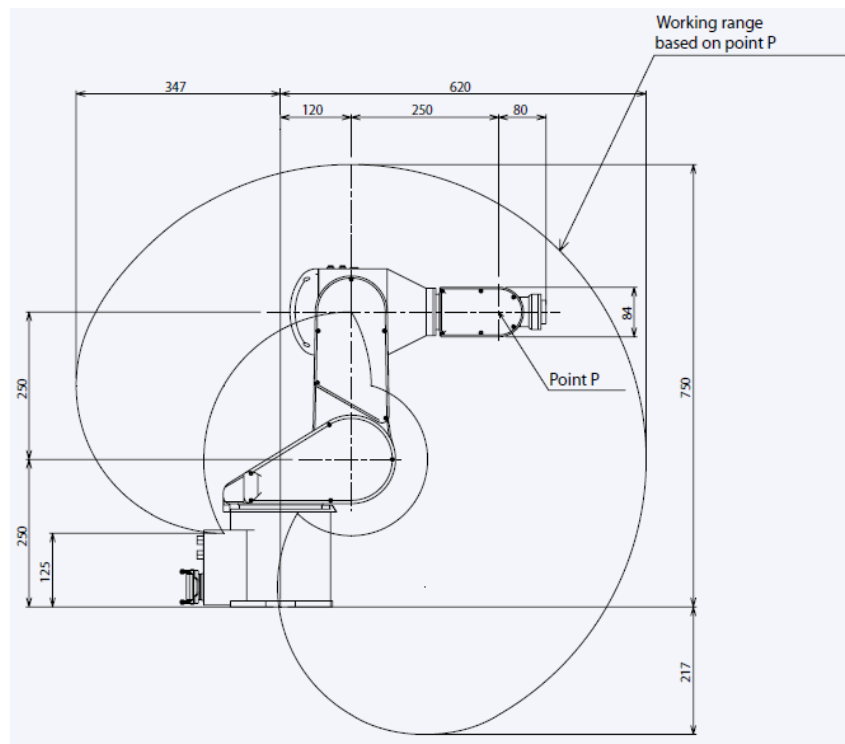


Figura 2: Spazio di lavoro del robot RS03N

Il carico massimo utile, cioè il peso massimo manipolabile a livello del polso, è di 3Kg, utensile compreso, e la precisione di ripetibilità dei movimenti è nell'ordine dei centesimi di millimetro. Si tratta quindi di un ottimo strumento per lo studio nella pratica di un robot industriale moderno.

### 2.1.1 Sistemi di coordinate

Le modalità principali di movimento sono con le posizioni dei giunti (JOINT) o con le coordinate spaziali rispetto alla base XYZOAT (BASE). Nel primo caso, si controllano direttamente i movimenti dei singoli giunti e i movimenti eseguibili saranno quindi quelli in Figura 5. Per comodità i giunti vengono numerati e denominati JT.

E' da notare che è possibile utilizzare le due tipologie insieme in un medesimo programma senza limitare quindi la scelta. Sarà quindi opportuno utilizzare movimenti di tipo JOINT oppure BASE in base alle necessità. La misurazione della posizione dei giunti avviene con degli encoder assoluti.

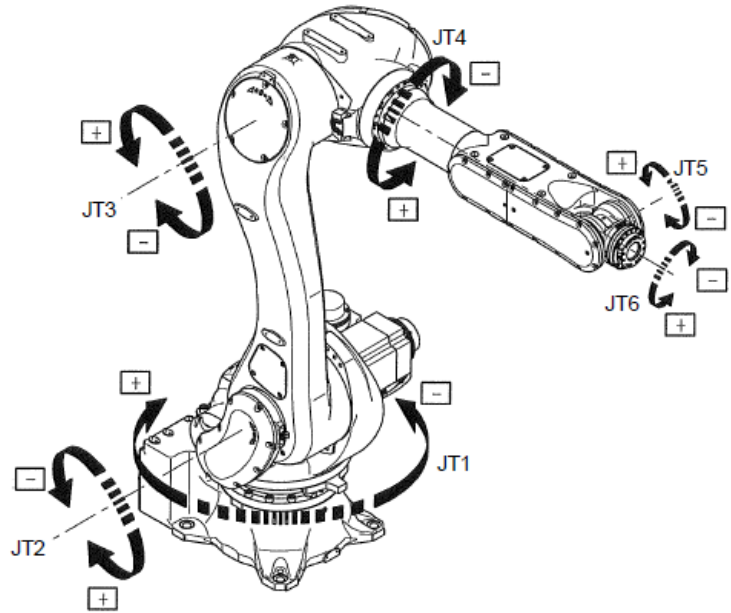


Figura 3: Movimento in base alla posizione dei giunti

Nella modalità di controllo dei movimenti secondo le coordinate di base, o XYZOAT, è il controller a preoccuparsi dello spostamento dei giunti in modo da permettere lo spostamento del polso nelle direzioni desiderate.

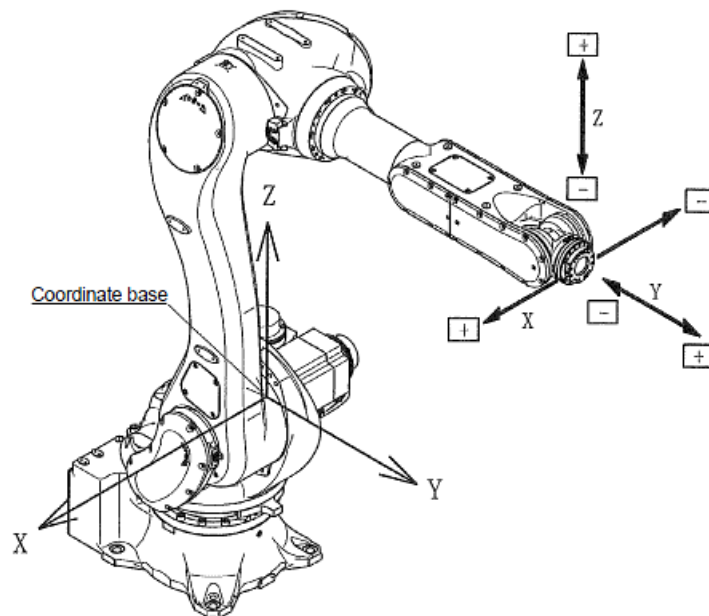
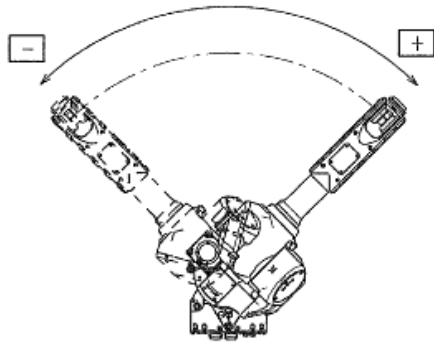
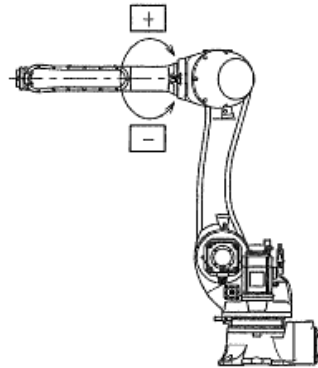


Figura 4: Le coordinate di base di riferimento

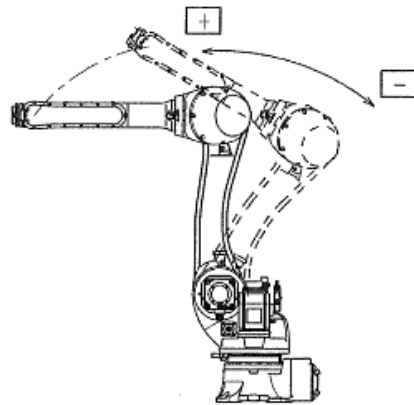
JT1: Movimento girevole del braccio a sinistra e a destra



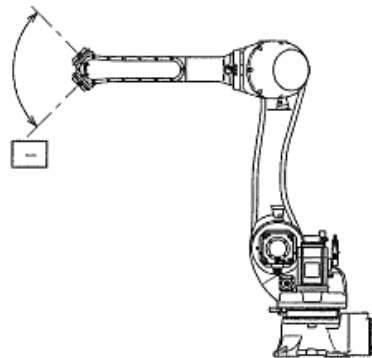
JT4: Movimento girevole del polso (1)



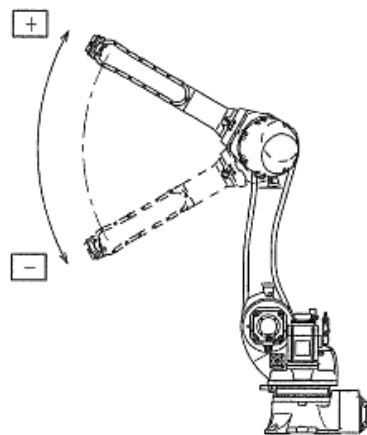
JT2: Movimento del braccio avanti e indietro



JT5: Movimento girevole del polso (2)



JT3: Movimento del braccio alto-basso



JT6: Movimento girevole del polso (3)

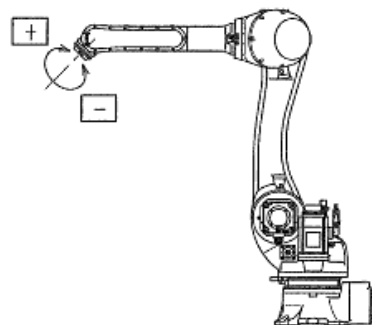
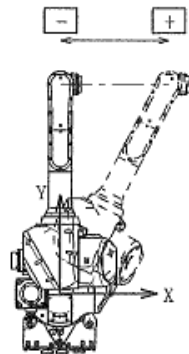
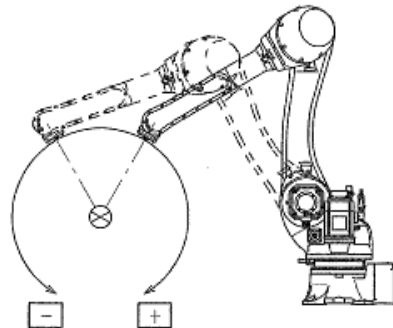


Figura 5: Movimenti controllando JT1-6

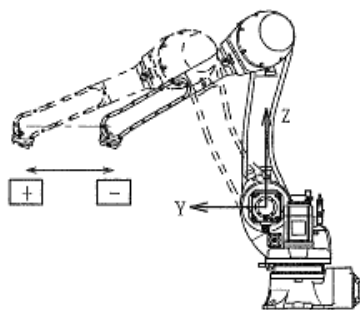
X: Movimento del braccio a sinistra e destra parallelo alle coordinate X della base (L'orientamento del polso è costante.)



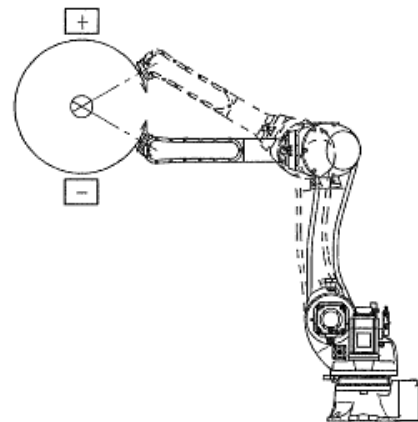
RX Rotazione attorno alle coordinate X della base ( Direzione - (negativa) rispetto al lettore.) (L'origine delle coordinate utensile nullo (TCP) non si muove.)



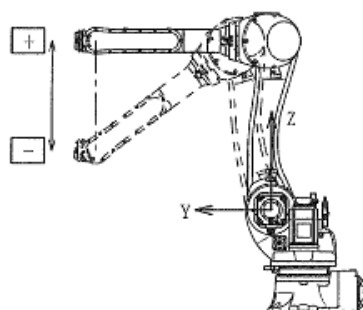
Y: Movimento del braccio avanti e indietro parallelo alle coordinate Y della base (L'orientamento del polso è costante.)



RY Rotazione attorno alle coordinate Y della base (Direzione + (positiva) verso JT2) (Il TCP non si muove.)



Z: Movimento del braccio alto - basso parallelo alle coordinate Z della base (L'orientamento del polso è costante.)



RZ Rotazione attorno alle coordinate Z della base (Direzione +(positivo) rispetto al lettore) (Il TCP non si muove.)

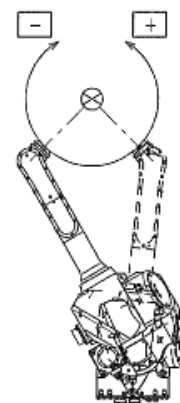


Figura 6: Possibili movimenti lungo le 6 coordinate di base

## 2.2 Controller

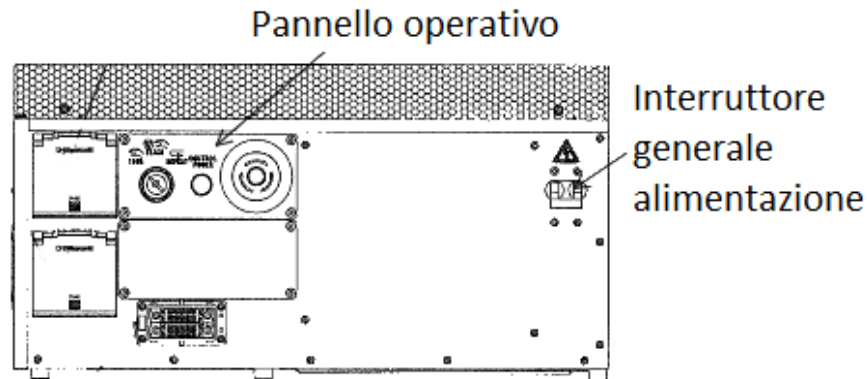


Figura 7: Vista frontale del controller

Sul frontale del controller è posizionato il suo componente principale, cioè il pannello operativo, mostrato in

Figura 8. Esso è composto da tre componenti:

1. Interruttore a chiave per la selezione della modalità. Le modalità di utilizzo del robot disponibili sono:
  - 100%: modalità di verifica rapida del programma impostato per la ripetizione automatica;
  - TEACH: modalità per l'insegnamento;
  - REPEAT: modalità di ripetizione automatica e indipendente di programmi.
2. Control power: la luce si accende quando l'alimentazione è su ON.
3. Pulsante rosso: da utilizzare in caso di emergenza e prima di spegnere il controller, interrompe l'alimentazione dei motori e attiva i freni, bloccando il robot. L'alimentazione generale resta però attiva (il sistema non si spegne). Per sbloccarlo ruotare il pulsante in direzione oraria fino al suo rilascio.

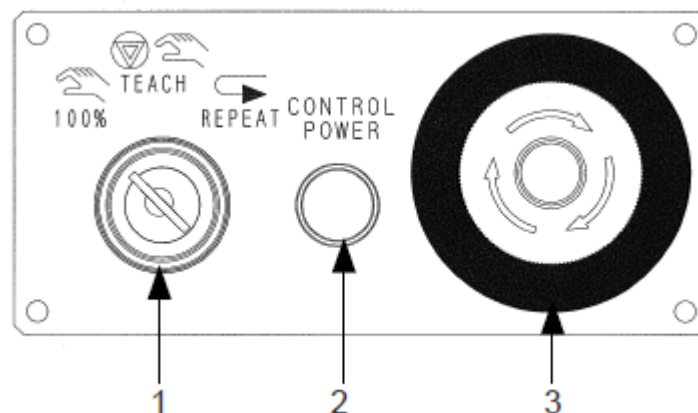


Figura 8: Pannello operativo del controller



## 2.3 Teach Pendant o Unità Portatile

Al controller è collegato il cosiddetto Teach Pendant, o unità portatile, che rappresenta il principale sistema di insegnamento al robot, in particolare di pose.

Il TP, visibile in Figura 9, è composto da:

- Schermo touchscreen, dettaglio in Figura 12;
- Tastiera hardware, dettaglio in Figura 11;
- Pulsante arresto di emergenza, interrompe l'alimentazione ai motori e arresta il robot;
- Pulsante blocco insegnamento, in teach mode è da posizionare su ON per abilitare le operazioni manuali e le operazioni di verifica, in repeat mode è da posizionare su OFF per eseguire le operazioni di ripetizione automatica;
- Interruttori di sicurezza sul retro, dettaglio in Figura 10, è un interruttore di abilitazione del robot, ed è necessario premerne leggermente uno dei due disponibili per poter manovrare il robot. Una pressione a fondo equivale al rilascio dell'interruttore, cioè disabilita il robot (in inglese vengono detti deadman switch e sul manuale italiano la traduzione è "interruttore uomo morto").

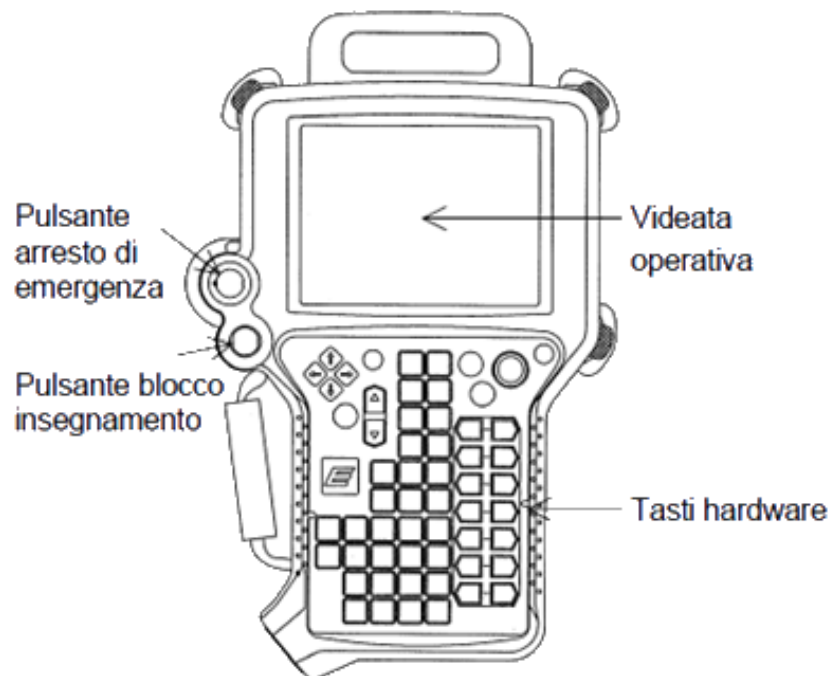


Figura 9: Il Teach Pendant o Unità Portatile

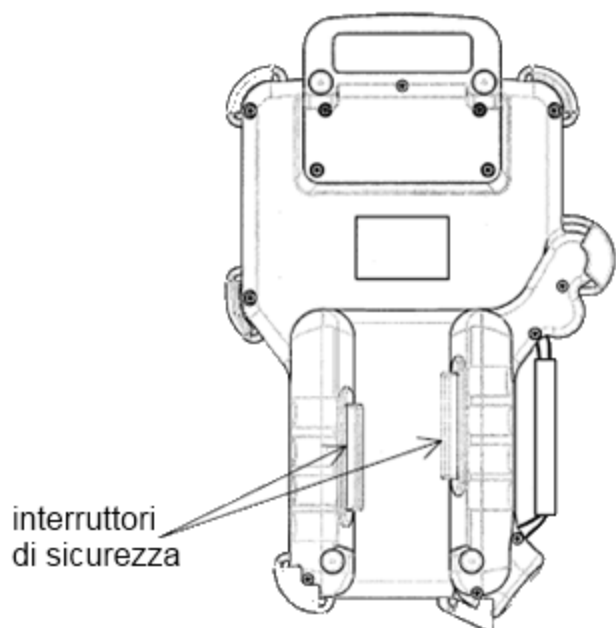


Figura 10: Retro del Teach Pendant

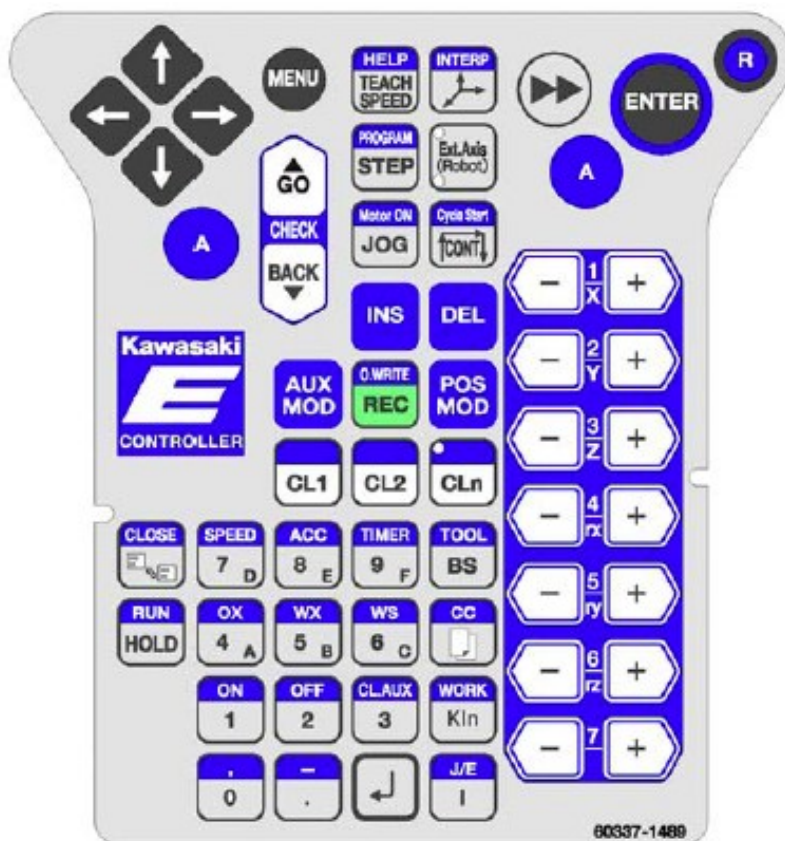








Figura 11: Pulsantiera del TP

Di seguito vengono descritti i bottoni più utilizzati presenti sul TP (Figura 11):

	<p>Tenendo premuto questo bottone vengono abilitate le <b>funzioni alternative</b>, visibili in blu, degli altri bottoni (come il tasto MAIUSC/SHIFT sulle normali tastiere dei computer).</p>
	<p>Funzione <b>Base</b>: Cicla attraverso i diversi sistemi di coordinate, nell'ordine: JOINT → Base → Tool</p>
	<p>Funzione <b>Base</b>: Mentre è premuto, in modalità insegnamento o verifica, aumenta temporaneamente la velocità del robot.</p> <p>Funzione <b>Alternativa</b>: Attiva/Disattiva l'alimentazione dei motori. Se i motori non sono attivi il robot non si muoverà.</p>
	<p>Funzione <b>Base</b>: Arresta il robot.</p> <p>Funzione <b>Alternativa</b>: Abilita il robot. Necessario per farlo muovere (insieme a MOTOR ON).</p>
	<p>Muove il robot. A seconda del tipo di sistema di coordinate impostato muoverà il giunto associato oppure la coordinata associata.</p> <p>Esempio: in modalità JOINT il [+] associato a 1 X muoverà il giunto JT1 in senso orario. Esempio: in modalità BASE il [-] associato a 3 Z muoverà il polso, mantenendone l'orientamento costante, verso il basso lungo la coordinata Z.</p>
	<p>Funzione <b>Base</b>: Imposta la modalità di ripetizione del programma attuale. Possibili opzioni: ripetizione singola o ripetizione continua.</p> <p>Funzione <b>Alternativa</b>: Avvia il ciclo di esecuzione automatica del programma attuale in modalità ripetizione.</p>



Funzione **Base**: aggiunge l'istruzione semplice al passo dopo quello corrente.

Funzione **Alternativa**: Sovrascrive il passo corrente con l'istruzione semplice appena creata.

Per la lista completa delle funzionalità di tutti i bottoni consultare il manuale operativo alle pagine da 2-6 a 2-11.

Molte delle funzionalità attivabili dei bottoni del TP sono accessibili direttamente attraverso il touchscreen avendo quindi un accesso più immediato.



Per esempio le funzioni del bottone qui a lato sono: STEP, selezionare un passo del programma attuale (funzione base) o PROGRAM, cambiare programma (funzione alternativa). Le medesime funzioni possono essere attivate premendo lo schermo nelle zone Program e STEP in alto nella Sezione A come visibili in Figura 12.

In particolare si consiglia l'uso dello schermo per l'impostazione più pratica delle velocità del robot (REP. SPD e MAN.SPEED, rispettivamente velocità di movimento in modalità di verifica e velocità di movimento negli spostamenti manuali con i tasti [+] e [-] del TP).

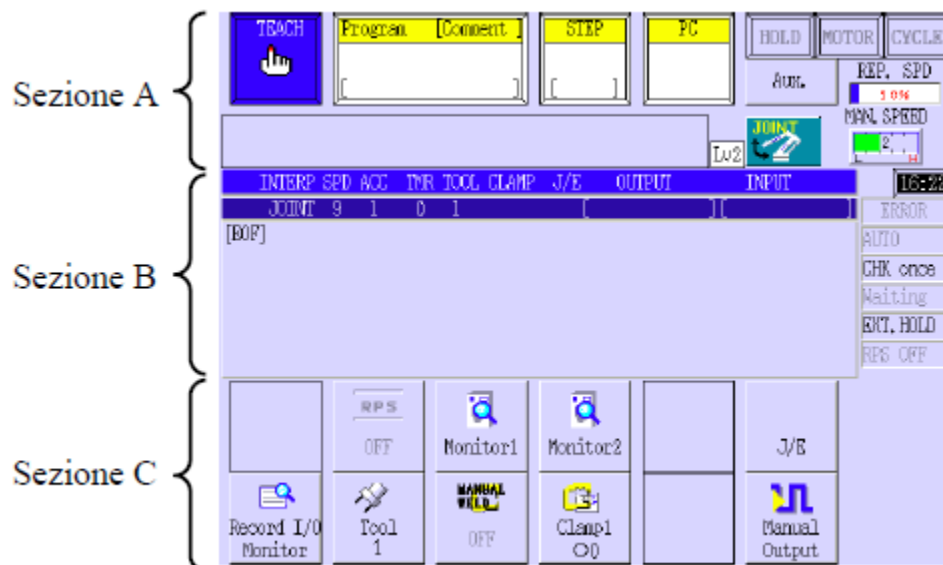


Figura 12: Generica schermata del TP in teach mode

Lo schermo offre molte altre funzionalità. È possibile trovarle descritte approfonditamente nel manuale operativo dalla pagina 2-12 alla 2-69.

Una nota particolare deve essere posta alle schermate di errore. L'errore più comune nell'utilizzo del robot consiste nel tentare di muovere il polso verso una posizione al di fuori dell'area di lavoro. In casi

come quello appena descritto verrà visualizzato sullo schermo un messaggio con la descrizione dell'errore come in Figura 13. Per continuare è obbligatorio il reset dell'errore. Nel caso si preme il tasto "Close" l'errore viene "messo da parte" come in Figura 14 (nda: si riferisce ad un errore differente rispetto alla Figura 13) ma sarà comunque necessario il reset dell'errore per muovere il robot. Nell'appendice del manuale operativo è presente la lista dei possibili errori e relativi codicie errore(da A-1 a A-38).

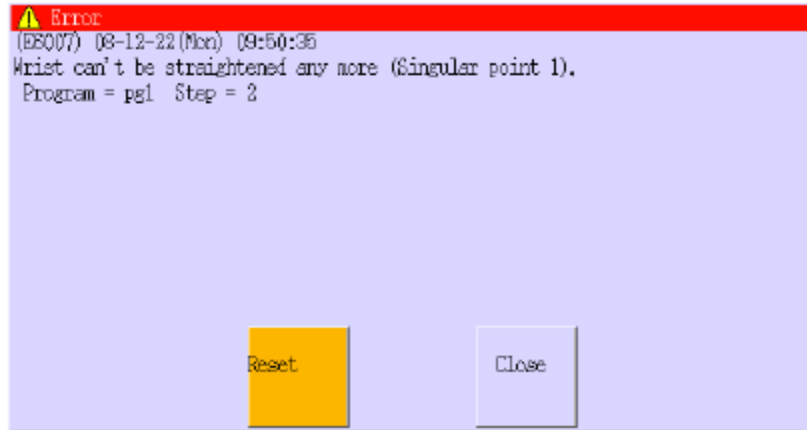


Figura 13: Esempio di schermata di errore visualizzata sul TP

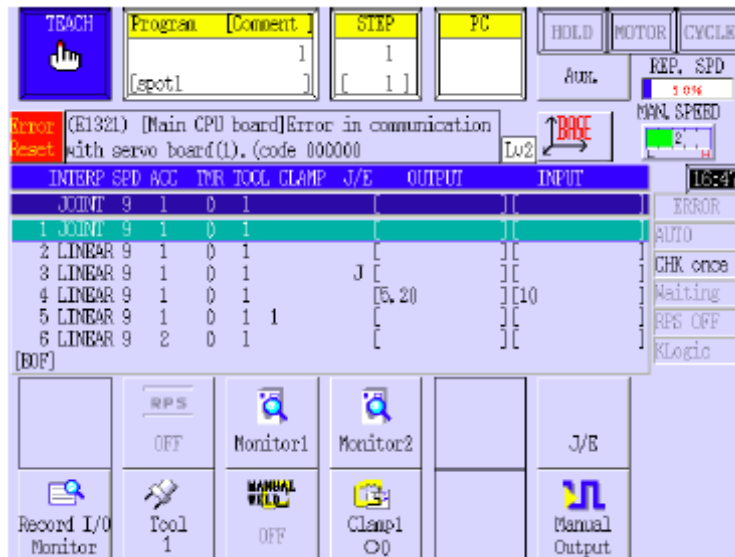


Figura 14: Attenzione: se non si resettano gli errori sarà impossibile continuare l'insegnamento

## 3 Funzioni di base

### 3.1 Accensione

Per accendere il robot è sufficiente attivare l'interruttore di alimentazione presente sulla destra nella parte frontale del controller, visibile in Figura 7. Per il completo avvio del sistema è necessario attendere il caricamento del sistema operativo che verrà visualizzato sul teach pendant. La modalità di funzionamento è modificabile anche quando il sistema è già acceso. Controllare che i pulsanti di emergenza sul pannello operativo e sul Teach Pendant siano entrambi disattivati.

#### 3.1.1.1 Accensione teach mode

1. Selezionare la modalità **teach** sul pannello operativo del controller;
2. Attivare lo switch dell'alimentazione sul controller;
3. Attendere il caricamento del sistema operativo;
4. Verificare che il robot sia in stato RUN (in alto a destra sullo schermo del TP);
5. Accendere i motori premendo A+Motor ON oppure A + pressione sul bottone MOTOR sul touchscreen;
6. Spostare l'interruttore blocco insegnamento su **ON**;
7. A questo punto il robot è pronto a ricevere comandi, tenere premuto uno dei due interruttori di sicurezza sul retro del TP per sbloccare i freni e azionare il robot.

#### 3.1.1.2 Accensione repeat mode

1. Selezionare la modalità **repeat** sul pannello operativo del controller;
2. Attivare lo switch dell'alimentazione sul controller;
3. Attendere il caricamento del sistema operativo;
4. Verificare che il robot sia in stato RUN (in alto a destra sullo schermo del TP);
5. Accendere i motori premendo A+Motor ON oppure A + pressione sul bottone MOTOR sul touchscreen;
6. Spostare l'interruttore blocco insegnamento su **OFF**;
7. A questo punto il robot è pronto a eseguire programmi o ricevere comandi tramite la console AS.

### 3.2 Spegnimento

Per spegnere il sistema, premere il pulsante rosso sul pannello operativo e successivamente disattivare l'interruttore di alimentazione.

### 3.3 Insegnamento

La programmazione, o insegnamento, del robot può essere effettuata utilizzando una delle seguenti due modalità:

- tramite teach pendant;
- utilizzando il linguaggio AS.

Essendo l'insegnamento la funzionalità più complessa nell'utilizzo del robot entrambi i sistemi vengono introdotti successivamente con l'ausilio di programmi di esempio sotto forma di tutorial passo-passo.

### 3.4 Esecuzione programmi

Al termine della fase di insegnamento il robot è pronto per l'esecuzione in serie dei programmi.

Così come l'insegnamento, anche la fase di ripetizione automatica è controllabile sia da TP sia tramite il sistema AS. Di seguito le due procedure.

#### 3.4.1 TP

1. Accendere il sistema in repeat mode
2. Selezionare il programma con il tasto PROGRAM sul TP oppure il bottone PROGRAM sul touchscreen;
3. Automaticamente verrà selezionato il passo 1;
4. Impostare, in base alle necessità, velocità, continuità degli step, continuità della ripetizione del programma;
5. Premere A+Cycle Start sul TP oppure A+ CYCLE sul touchscreen.

In caso di ripetizione continua il programma verrà eseguito ciclicamente più volte. Per bloccare l'esecuzione premere HOLD sul TP. In un comportamento inaspettato o di emergenza premere il pulsante rosso sul controller o sul TP.

#### 3.4.2 AS

1. Accendere il sistema in repeat mode;
2. Collegarsi alla console del sistema AS;
3. Impostare la velocità adeguata in percentuale della velocità massima, esempio SPEED 10;
4. Eseguire direttamente il programma con EXECUTE nome\_programma, N<sup>1</sup>.

Nota: i programmi creati con il TP che hanno come nome un numero N tra 0 e 999 vengono salvati con nome pgN nel sistema AS, sono quindi richiamabili con EXECUTE pgN.

Durante l'esecuzione del programma è possibile inserire comandi monitor, esempio:

- ABORT/HOLD per fermare l'esecuzione dopo il passo corrente;
- CONTINUE per riprenderla;
- WHERE per conoscere la posizione corrente dei giunti.

#### 3.4.3 AS + TP

1. Accendere il sistema in repeat mode;
2. Collegarsi alla console del sistema AS;
3. Selezionare il programma con PRIME nome\_programma, N<sup>2</sup>;

---

<sup>1</sup> N è un parametro opzionale, rappresenta il numero di volte che si vuole eseguire nome\_programma. Se N=-1 l'esecuzione proseguirà indefinitamente.

4. A questo punto è possibile procedere come descritto nell'esecuzione tramite TP dal punto 4 in poi.

---

<sup>2</sup> Come per EXECUTE con la differenza che sul TP deve essere selezionato CYCLE CONT. Altrimenti il programma verrà eseguito sempre e solo una volta, indipendentemente dal valore di N.



## 4 Insegnamento tramite Teach Pendant

Utilizzare esclusivamente il TP rappresenta la forma più semplice di insegnamento per questo robot. È possibile creare programmi, sia semplici sia molto complessi. Questi ultimi però, per la natura dello strumento risultano più difficili da produrre poiché presenta delle limitazioni pratiche. Un esempio lampante è la tastiera sullo schermo touchscreen in cui i tasti sono molto piccoli e quindi praticamente inutilizzabili per lunghe digitazioni. È da notare come sia possibile, sempre tramite TP, utilizzare il linguaggio AS, che verrà introdotto in seguito. Per lo stesso motivo appena detto però risulta una strada abbastanza ostica. Le funzioni del TP sono già state introdotte in precedenza quindi è possibile passare direttamente al formato delle istruzioni che formano un programma creato tramite TP.

### 4.1 Formato delle istruzioni

Le istruzioni, dette elementari, utilizzano un formato fisso, in Figura 17 viene riportata la tabella con l'elenco dei campi che formano ogni singola istruzione inserita tramite Teach Pendant. L'attenzione verrà qui posta sui campi principali e obbligatori.

Una premessa fondamentale: ad ogni istruzione è automaticamente assegnata la posa in cui si trova il robot mentre l'istruzione stessa viene salvata all'interno del programma. Per questo motivo, l'inserimento delle istruzioni avviene generalmente seguendo la procedura rappresentata in Figura 15.

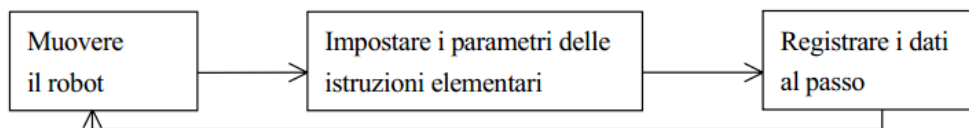


Figura 15: Passi per la creazione di un programma tramite TP

Un esempio di programma potrebbe essere quello che muove il robot lungo il percorso rappresentato in Figura 16. Il primo passo consiste quindi nel posizionare il robot nella posa del Punto1, come punto di inizio. Nello spostamento iniziale generalmente ci sono meno vincoli sul percorso che il polso deve seguire, si può quindi scegliere uno spostamento di tipo giunto. Il secondo passo consiste nel posizionare il robot nella posa del Punto2. Il percorso desiderato per il polso, per passare dal Punto1 al Punto2, è lineare, si imposta quindi questa informazione e si prosegue via via per concludere tutto il giro.

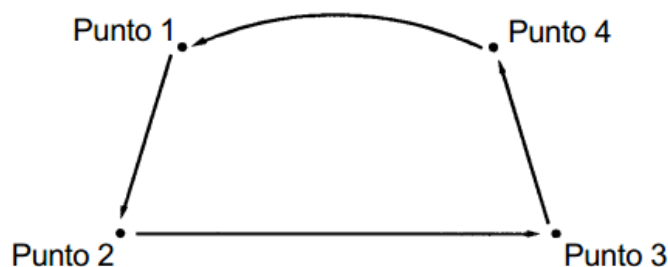


Figura 16: Esempio di percorso composto da tre movimenti lineari e uno circolare

Istruzione elementare	Interpolation (Interpolazione)	Velocity (Velocità)	Accuracy (Precisione)	Timer	Tool (Utensile)	Clamp (Serraggio)	WK (Pezzo) *	J/E (salto) / (fine)	Output	Input
Valore di Parametro	JOINT(giunto) / LINEAR(lineare)/LIN2/ CIR1/CIR2/ FLinear(FLineare)/ FCIR1/FCIR2/XLIN	0-9	0-4	0-9	1-9	Senza visualizzazione, 1-2	Senza visualizzazione, C	J, E	1-64 o 1-96	1-64 o 1-96
Tasti	<b>A+</b> <b>INTERP</b>	<b>A+</b> <b>SPD</b>	<b>A+</b> <b>ACC</b>	<b>A+</b> <b>TMR</b>	<b>A+</b> <b>TOOL</b> o <Tool>	<b>CL 1</b> / <b>CL 2</b>	<b>A+</b> <b>&lt;Work&gt;</b>	<b>A+</b> <b>J/E</b> o <b>&lt;J/E&gt;</b>	<b>A+</b> <b>OX</b>	<b>A+</b> <b>WX</b>

Figura 17: Formato delle istruzioni dei programmi inseriti tramite TP

#### 4.1.1 Campi

Di seguito i campi di maggiore interesse. Per modificare i campi utilizzare i corrispettivi tasti del TP riportati in Figura 17. Maggiori informazioni sono reperibili sul manuale operativo alle pagine dalla 5-5 alla 5-16.

##### 4.1.1.1 Interpolazione

Indica quale percorso il robot debba far seguire al polso muovendosi dalla posa di partenza a quella di arrivo. Le opzioni più utilizzate sono:

- JOINT, il percorso non è importante mentre il movimento risulta più veloce poichè lo spazio percorso sarà minore;
- LINEAR, il movimento sarà una linea retta tra le due pose;
- CIRCULAR, specificando una terza posa si identifica una singola traiettoria circolare possibile che passa dalla posa iniziale e porta in quella finale.

##### 4.1.1.2 Velocità

Indica a quale velocità viene spostato il polso lungo il percorso scelto, con due possibili metriche:

- Valore costanti da 0 a 9, le cui velocità effettive sono impostate nelle opzioni del robot (attenzione: le velocità potrebbero non essere in ordine crescente);
- Valore diretto, inserendo 10 viene successivamente chiesto il valore di velocità effettivamente desiderato. Questo può essere in secondi, in caso di interpolazione JOINT, oppure in mm/s, nel caso LINEAR.

##### 4.1.1.3 Precisione

Indica qual'è la tolleranza con cui il polso viene considerato nella posa insegnata per il passo corrente.

Così come per la velocità viene utilizzato un valore fittizio da 0 a 4, il cui reale valore di precisione è impostato nelle opzioni. Salvo particolari necessità è generalmente consigliato il valore 0, di default.

#### 4.1.1.4 Timer

Indica qual'è il tempo che deve passare successivamente al raggiungimento della posa per considerare questa corretta e proseguire quindi con l'istruzione successiva.

Anche in questo caso il valore è fittizio, con impostazione nelle opzioni, e può essere valorizzato da 0 a 9. Di default l'impostazione è in tempo crescente: 0 tempo corto, 9 tempo lungo.

#### 4.1.1.5 Serraggio

Indica quale operazione di serraggio effettuare quando il robot si trova nella posa di arrivo. Il robot in laboratorio, al momento, è dotato di pinza a dita, quindi l'unico serraggio possibile consiste nell'apertura e chiusura delle dita ed è impostato sul serraggio numero 1 (CL1).

Nota: durante l'insegnamento potrebbe essere necessario effettuare piccoli aggiustamenti alla posa per far sì che questa sia particolarmente precisa nel prendere o lasciare gli oggetti. Per attivare il serraggio, piuttosto che modificare solamente il campo dell'istruzione, utilizzare il tasto TP A+CL1 sia per aprire sia per chiudere la pinza.

## 4.2 Programma di esempio con TP

Seguendo la procedura descritta per la creazione di un programma tramite TP viene di seguito riportato un semplice esempio in cui:

1. viene portato il robot in una posa iniziale in cui si trova una pallina;
2. viene presa la pallina;
3. viene portato il robot in una seconda posa, con movimento lineare;
4. viene depositata la pallina.

I passi da eseguire per la creazione del programma sono:

1. Accendere il controller;
2. Attendere la completa accensione del sistema;
3. Per creare un nuovo programma premere A+PROGRAM (o PROGRAM sul touchscreen) ed inserire un valore tra 0 e 999. Se un programma al numero selezionato è già esistente, verrà caricato e mostrato sullo schermo. Cercare una posizione libera.
4. Abilitare il robot (RUN) e i motori (MOTOR ON);
5. Abilitare la modalità TEACH ON;
6. Per iniziale assicurarsi che il robot sia in una posa "sicura", cioè senza il rischio di collidere con altri oggetti (generalmente HOME, tutti i giunti in posizione zero);
7. Spostare il robot nella prima posa desiderata (iniziale), cioè sopra la pallina;
8. Impostare e salvare, premendo il tasto REC, l'istruzione nel programma utilizzando interpolazione JOINT;
9. Abbassare il robot lungo l'asse z in modo che le dita si trovino ai lati della pallina (assicurarsi preventivamente che la pinza sia aperta!);
10. Verificare la corretta chiusura della pinza e presa della pallina con A+CL1;
11. Impostare e salvare l'istruzione nel programma utilizzando interpolazione LINEAR;

12. Alzare il robot lungo l'asse z tornando all'incirca alla posa iniziale;
13. Impostare e salvare l'istruzione nel programma utilizzando interpolazione LINEAR;
14. Spostare il robot nella seconda posa desiderata (finale), lasciando invariata l'altezza dal piano;
15. Impostare e salvare l'istruzione nel programma utilizzando interpolazione LINEAR;
16. Abbassare il robot lungo l'asse z in modo che la pallina si appoggi leggermente sul piano;
17. Verificare il corretto rilascio della pallina nella posizione desiderata aprendo la pinza con A+CL1;
18. Impostare e salvare l'istruzione nel programma utilizzando interpolazione LINEAR;
19. Alzare il robot lungo l'asse z tornando all'incirca alla posa finale;
20. Impostare e salvare l'istruzione nel programma utilizzando interpolazione LINEAR;
21. Opzionalmente: riportare il robot nella posa HOME con interpolazione JOINT;

Nota: nei movimenti verticali di avvicinamento e allontanamento dalla pallina è consigliabile utilizzare una velocità moderata.

In Figura 18 è visibile il listato del programma prodotto durante il test della procedura appena descritta. Da notare tra parentesi quadra i valori di giunto delle pose utilizzate (lo 0 finale è il valore JT7 di un ulteriore giunto attualmente non presente fisicamente).

```
>list/p pg1
PROGRAM pg1()
1 JOINT SPEED3 ACCU1 TIMER9 TOOL1 WORK0 CLAMP1 (OFF,0,0,0) 2 (OFF,0,0
,O) OX= WX= #[96.527,18.349,-125.13,-3.3552,-37.886,41.954,0] ;
2 LINEAR SPEED1 ACCU1 TIMER9 TOOL1 WORK0 CLAMP1 (ON,0,0,0) 2 (OFF,0,0
,O) OX= WX= #[96.527,34.382,-125.42,-5.6027,-21.605,44.515,0] ;
3 LINEAR SPEED1 ACCU1 TIMER9 TOOL1 WORK0 CLAMP1 (ON,0,0,0) 2 (OFF,0,0
,O) OX= WX= #[96.527,21.914,-125.63,-3.7026,-33.83,42.381,0] ;
4 LINEAR SPEED3 ACCU1 TIMER9 TOOL1 WORK0 CLAMP1 (ON,0,0,0) 2 (OFF,0,0
,O) OX= WX= #[78.869,45.766,-82.087,-1.9324,-54.041,58.1,0] ;
5 LINEAR SPEED1 ACCU1 TIMER9 TOOL1 WORK0 CLAMP1 (OFF,0,0,0) 2 (OFF,0,
O,O) OX= WX= #[78.869,53.23,-81.937,-2.1473,-46.732,58.439,0] ;
6 LINEAR SPEED1 ACCU1 TIMER9 TOOL1 WORK0 END CLAMP1 (OFF,0,0,0) 2 (OF
F,0,0,0) OX= WX= #[78.869,44.644,-81.901,-1.9021,-55.346,58.052,0] ;
.END
>
```

Figura 18: Listato del programma inserito tramite TP

### 4.3 Check del programma passo-passo

Successivamente alla creazione di un programma è fondamentale verificarne la corretta implementazione rispetto alle intenzioni originali. Questo è possibile farlo procedendo un passo alla volta, per evitare situazioni di rischio per persone o cose, come descritto in seguito.

I passi da eseguire, supponendo il controller sia già acceso in teach mode, con teach lock ON e il robot abilitato a muoversi:

1. Selezionare il programma da controllare, se è appena stato creato questo passo è superfluo;
2. Selezionare il passo da cui si vuole partire con la verifica (tasto STEP oppure A+freccia su/giu);
3. Controllare che la velocità impostata (MAN. SPD) sia adeguatamente bassa;
4. Tenendo premuto l'interruttore di sicurezza (deadman switch), sul retro del TP, si può procedere alla verifica dei passi del programma.

5. Se è impostata la ripetizione singola (STEP ONCE): premere A+GO per eseguire il passo corrente, premerlo nuovamente per proseguire;
6. Se viene impostata la ripetizione continua (STEP CONT): tenendo premuto GO vengono eseguiti i passi in maniera continua, se il tasto viene rilasciato il robot si arresta immediatamente;
7. Procedere fino al passo desiderato.

Nota: in caso che il movimento del robot non sia quello desiderato rilasciare l'interruttore di sicurezza sul retro del TP oppure premere i bottoni rossi per un arresto di emergenza.

Nota: così come durante gli spostamenti manuali, è possibile velocizzare l'esecuzione di ogni passo tenendo premuto il tasto JOG.

## 5 Insegnamento tramite console e linguaggio AS

I robot Kawasaki sono pilotati da un sistema software denominato AS. Nel sistema AS è possibile inserire comandi o eseguire programmi usando il linguaggio AS. Il sistema AS viene avviato automaticamente quando viene alimentato il controller attendendo poi l'inserimento di comandi da parte dell'utente. L'utilizzo del sistema AS può avvenire collegandosi fisicamente al controller tramite porta COM oppure Ethernet. Per accedere poi alla console viene utilizzato il programma KRterm, fornito da Kawasaki e già installato sul pc in laboratorio.

Il sistema AS può trovarsi in diverse modalità di funzionamento (ling AS 2-1):

- monitor, è la modalità di base ed è quella in cui ci si trova al collegamento con KRterm. In essa è possibile eseguire comandi monitor o passare alla modalità editor e playback;
- editor, è la modalità che permette la creazione e modifica di programmi, gli unici comandi eseguibili sono quelli di tipo editor;
- playback, si passa in questa modalità durante l'esecuzione di un programma, in questa modalità si possono usare alcuni dei comandi di controllo.

### 5.1 Collegamento al controller via ethernet con KRterm

Di seguito viene mostrata la procedura passo passo per il collegamento.

Supponendo che il sistema sia già acceso e il cavo di rete collegato.

1. Aprire KRterm, si aprirà la schermata principale come in Figura 19;

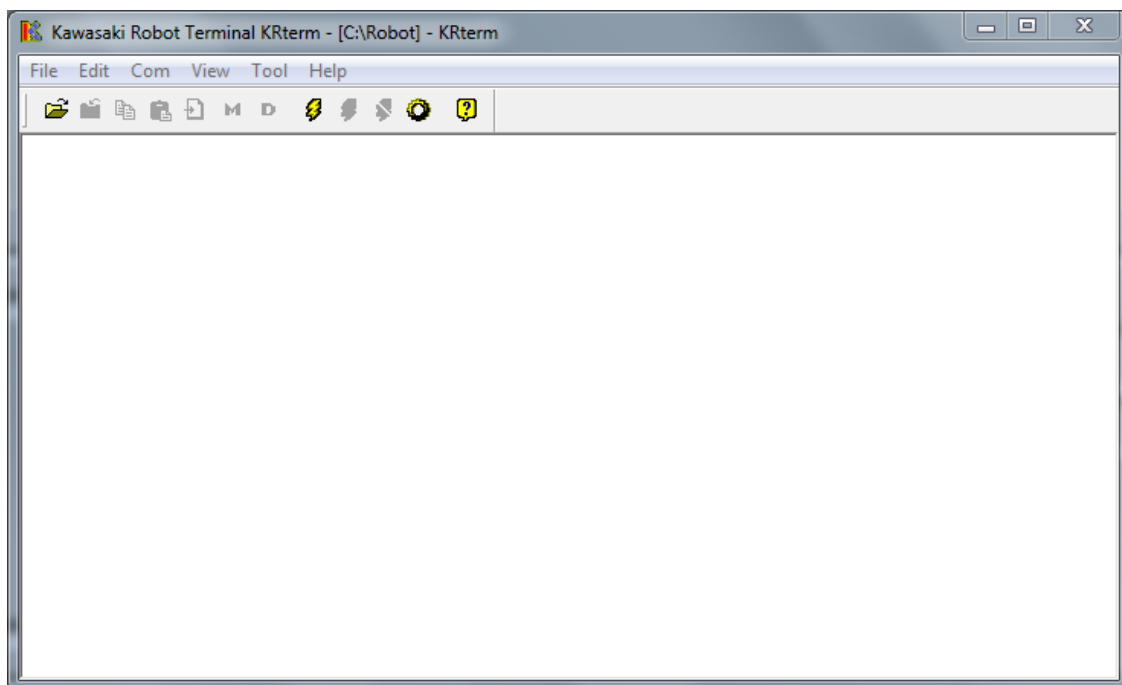


Figura 19: Schermata iniziale di KRterm

2. Dal menu Com selezionare Connect by List (Figura 20), verrà aperto il dialog Select Connection come mostrato in Figura 21;

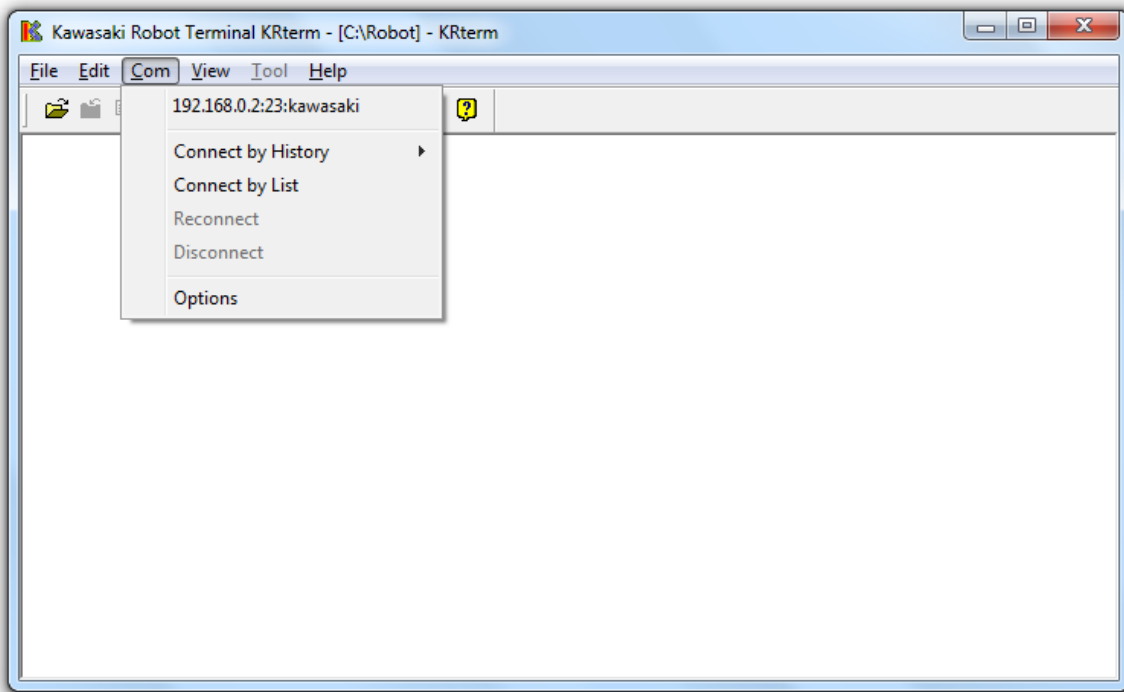


Figura 20: Menu Com di KRterm

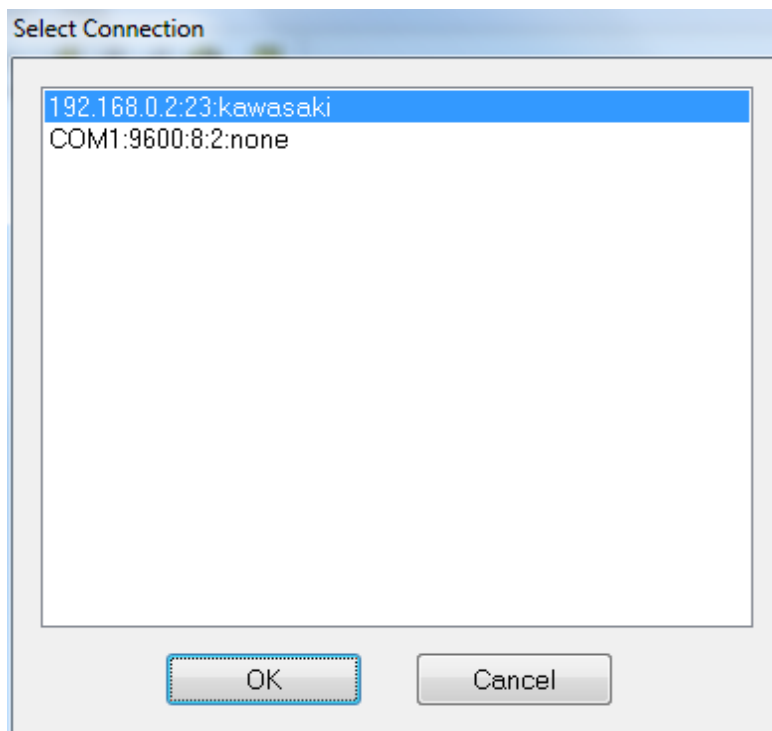


Figura 21: Finestra di dialogo per la scelta della connessione

3. Infine selezionare la voce 192.168.0.x:23:Kawasaki e premere OK;

Nota: l'ip attualmente viene assegnato dal DHCP interno del controller ma in futuro il sistema dovrebbe essere connesso alla rete universitaria per consentire un accesso remoto da altri laboratori quindi l'ip sarà diverso, regolarsi di conseguenza.

4. La console mostrerà una richiesta di login, digitare "as" e premere invio;
5. In Figura 22 viene mostrato il risultato di una connessione avvenuta correttamente;
6. A questo punto il sistema è in modalità monitor.

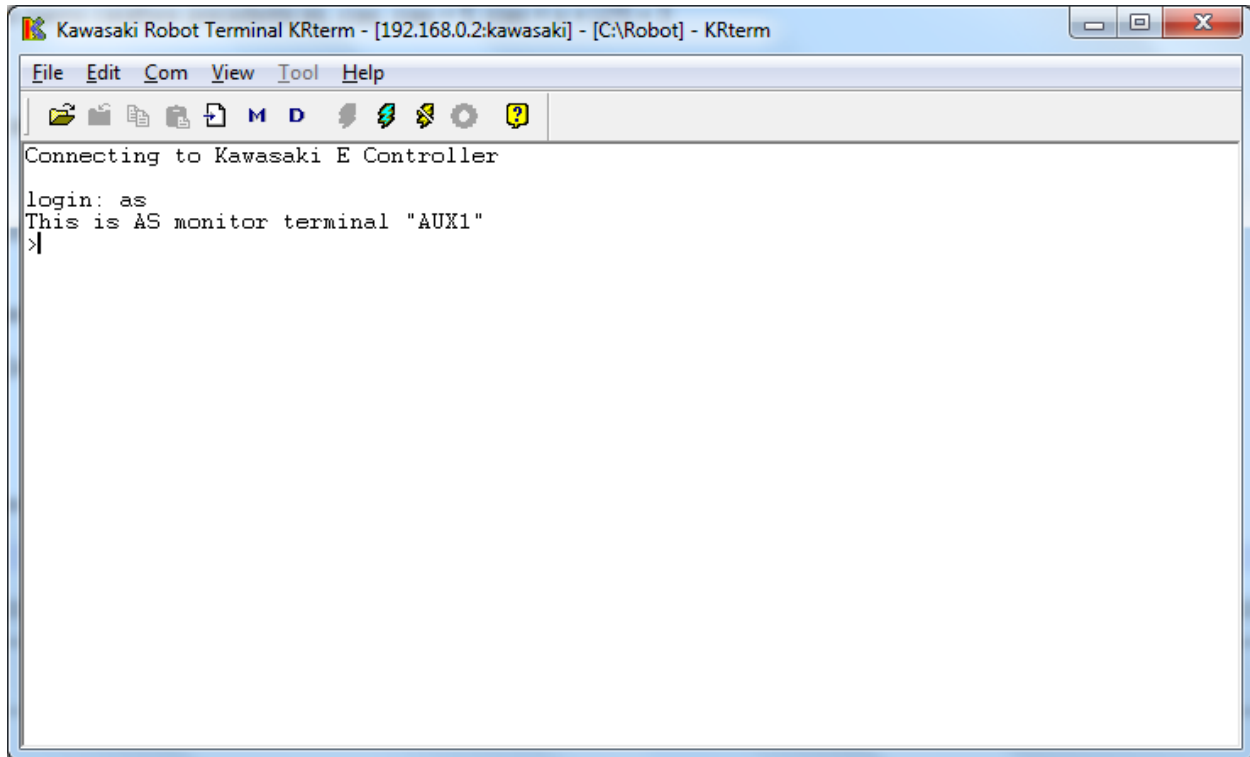


Figura 22: Connessione correttamente eseguita dopo il login

Nota: è possibile collegarsi alla console AS in modalità sia teach sia repeat. In base alla modalità saranno eseguibili alcune operazioni piuttosto di altre. Il sistema segnalerà con un messaggio l'impossibilità di eseguire un comando nella modalità corrente suggerendo di cambiarla.

Per scollegarsi dalla console è sufficiente selezionare dal menu Com la voce Disconnect.



## 5.2 Linguaggio AS

Come già detto il linguaggio AS è il linguaggio utilizzato per l'esecuzione di comandi nella console del sistema AS e per la scrittura di programmi per il robot.

I comandi disponibili sono divisibili in due macrosezioni principali:

- Comandi monitor, sono comandi che vengono utilizzati nella modalità monitor del sistema AS;
- Comandi programma, sono comandi che vengono utilizzati per la programmazione del robot.

### Convenzioni

Il linguaggio AS ha delle particolari caratteristiche e convenzioni. Di seguito viene riportato un breve elenco comprendente alcune delle più particolari:

- Lunghezza comandi massimo 128 caratteri;
- Nomi variabili massimo 15 caratteri case insensitive;
- Dimensione massima vettori 9999;
- \$ indica variabili stringa es. \$prova, \$prova = "hello world";
- Le variabili reali non hanno carattere precedente es. ciao, ciao = 5, ciao = x + L[1] + 3;
- Nelle variabili posa usare # per valori spostamento giunto, altrimenti vengono usati valori di trasformazione XYZOAT;

Tutte le informazioni approfondite su convenzioni, tipologia di variabili e operatori si trovano alle pagine dalla 3-1 alla 3-24 del manuale del linguaggio AS.

### 5.2.1 Comandi principali

In questa sezione vengono riportati alcuni dei comandi di utilizzo più frequente. Inizialmente viene indicata la pagina del manuale a cui fare riferimento per una descrizione dettagliata. In grassetto viene riportato il comando seguito dai parametri, tra parentesi quadre quelli opzionali, e infine una breve descrizione della funzione del comando stesso.

Oltre ai comandi qui presentati ne esistono molti altri. Inoltre, nei programmi, è possibile utilizzare delle funzioni utili consultabili alle pagine del manuale 9-2, 9-30, 9-46, 9-49. Infine è da notare che tutti i comandi possono essere abbreviati, per comodità di utilizzo, è però sconsigliato usare la forma abbreviata nei programmi poiché ne viene fortemente diminuita la leggibilità. Nell'appendice 5 del manuale è possibile trovare la lista di tutti i comandi comprensivi di breve descrizione e relativa abbreviazione.

### Comandi modalità monitor

I seguenti comandi possono essere utilizzati quando il sistema è in modalità monitor e servono per il controllo del robot e del sistema AS.

- 5-16 **USB\_FDIR** [nome\_cartella], visualizza tutti i file salvati sulla memoria USB, o in una specifica cartella.
- 5-17 **LIST** [nome\_programma], visualizza tutti i dati il programma, visualizza tutti i dati;

- LIST/P [nome\_programma], visualizza solo il programma;
  - LIST/L [variabile\_posa], visualizza solo le variabili di posa;
  - LIST/R [variabile\_reale], visualizza solo le variabili reali;
  - LIST/S [variabile\_stringa], visualizza solo le variabili stringa.
- 5-19 **USB\_FDEL** nome\_file, cancella i file specificati dalla memoria USB.
  - 5-29 **SAVE** nome\_file [=nome\_programma], salva i dati sulla memoria del PC (se connesso), o solo il programma indicato.
  - 5-29 **USB\_SAVE** nome\_file [=nome\_programma], salva i dati sulla memoria USB, o solo il programma indicato.
  - 5-33 **LOAD** nome\_file, carica i dati del file indicato dalla memoria del PC connesso.
  - 5-33 **USB\_LOAD** nome\_file, carica i dati del file indicato dalla memoria usb.
  - 5-37 **SPEED** velocità\_monitor, imposta la velocità monitor in percentuale della velocità massima.
  - 5-39 **EXECUTE** [nome\_programma], [cicli], [numero\_passo], esegue il programma.
  - 5-40 **STEP** [nome\_programma], [cicli], [numero\_passo], esegue un passo del programma.
  - 5-41 **ABORT**, arresta l'esecuzione al termine del passo corrente.
  - 5-41 **HOLD**, arresta l'esecuzione.
  - 5-42 **CONTINUE [NEXT]**, riprende l'esecuzione.
  - 5-44 **DO** [istruzione], esegue una singola istruzione di programma (non tutte sono eseguibili).
  - 5-46 **HERE** variabile\_posa, definisce una variabile di posa con il valore della posa attuale.
  - 5-47 **POINT** variabile\_posa 1 [= variabile\_posa2], definisce una variabile di posa.
  - 5-53 **STATUS**, visualizza lo stato del sistema e il programma corrente del robot.
  - 5-55 **WHERE** [modalità], visualizza i dati della posa attuale del robot, default in valori dei giunti (JT1,...JT6).

>where					
JT1	JT2	JT3	JT4	JT5	JT6
97.675	34.180	-126.466	-2.079	-19.595	-46.850
X[mm]	Y[mm]	Z[mm]	O[deg]	A[deg]	T[deg]
339.777	-46.770	-109.047	-115.870	179.266	112.995

Figura 23: Output di esempio del comando WHERE

- 5-82 **HELP**, visualizza un elenco dei comandi/istruzioni del linguaggio AS.
- 5-111 **PRINT/TYPE** [numero dispositivo :] dati di stampa, stampa sul dispositivo indicato i dati di stampa (PRINT "Hello World").

### Comandi modalità editor

Nella seguente tabella vengono riportati i principali comandi per l'utilizzo della modalità editor per la scrittura di programmi in linguaggio AS tramite console.

EDIT home del programma, passo	Avvia la modalità editor.
Program instructions	Sostituisce il passo corrente con una nuova istruzione.
tasto ENTER (↵)	Va al passo successivo senza modificare il passo corrente.
D numero di passi	Cancella il numero specificato di passi di programma.
E	In modo editor torna al modo monitor.
F stringa di caratteri	Ricerca di caratteri e visualizzazione della linea. (Find - Trova)
I	Inserisce un nuovo passo.
L	Visualizza il passo precedente. (Last - Ultimo)
M /caratteri esistenti /caratteri nuovo	Sostituisce il carattere esistente con un nuovo carattere. (Modify - Modifica)
O	Posiziona il cursore nel passo corrente per l'editing. (One line - Una linea)
P numero di passi	Visualizza il numero di passi specificato. (Print - Stampa)
R stringa di caratteri	Sostituisce caratteri all'interno di un passo.
S numero di passi	Seleziona passi del programma. (Step - Passo)
XD	Taglia il o i passi selezionati e li memorizza negli appunti.
XY	Copia il o i passi selezionati e li memorizza negli appunti.
XP	Incolla il contenuto degli appunti.
XQ	Incolla il contenuto degli appunti nell'ordine inverso.
XS	Mostra il contenuto degli appunti.
T	Assegna istruzioni di movimento mentre è in modo editor. (opzione)

Figura 24: Tabella dei comandi dell'editor

### Comandi programma

I seguenti comandi possono essere utilizzati per la programmazione del robot nei programmi AS.

- 6-3 **JMOVE** variabile\_posa [, numero\_serraggio], muove il robot verso la posa indicata con un movimento ad interpolazione giunti.
- 6-3 **LMOVE** variabile\_posa [, numero\_serraggio], muove il robot verso la posa indicata con un movimento ad interpolazione lineare.
- 6-5 **JAPPRO** variabile\_posa, distanza, muove il robot verso la posa indicata con un movimento ad interpolazione giunti ma arrivando alla distanza indicata lungo l'asse Z.
- 6-5 **LAPPRO** variabile\_posa, distanza, muove il robot verso la posa indicata con un movimento ad interpolazione lineare ma arrivando alla distanza indicata lungo l'asse Z.

- 6-7 **HOME** numero\_posa\_home, muove con un movimento ad interpolazione giunti nella posa HOME o HOME2.
- 6-8 **DRIVE** numero\_giunto, spostamento [, velocità], muove il giunto indicato.
- 6-9 **DRAW** [traslazione X] , [traslazione Y] , [traslazione Z] , [rotazione X] , [rotazione Y] , [rotazione Z] [, velocità], muove il robot con movimento lineare nel sistema di coordinate BASE dalla posa attuale alla posa alla distanza specificata (NB: è possibile omettere i parametri ma sono comunque necessarie le virgole fino all'ultimo parametro che viene specificato, esempio: DRAW 50,,,-30 muove il robot di 50 lungo l'asse X e di -30 lungo l'asse Z).
- 6-17 **SPEED** velocità, velocità di rotazione [,ALWAYS], imposta la velocità per il successivo movimento, se ALWAYS imposta velocità fino al successivo comando SPEED.
- 6-36 **OPEN** [numero\_serraggio], apre la mano del robot all'inizio del movimento successivo.
- 6-36 **OPENI** [numero\_serraggio], apre la mano del robot alla fine del movimento successivo.
- 6-37 **CLOSE** [numero\_serraggio], chiude la mano del robot all'inizio del movimento successivo.
- 6-37 **CLOSEI** [numero\_serraggio], chiude la mano del robot alla fine del movimento successivo.
- 6-46 **GOTO** etichetta [IF condizione], salta al passo del programma con l'etichetta indicata, eventualmente in base alla condizione specificata.
- 6-48 **CALL** nome\_programma, sospende l'esecuzione del programma corrente e salta al programma indicato come se fosse un sottoprogramma.
- 6-48 **RETURN**, termina l'esecuzione del sottoprogramma e torna al programma in cui c'è stata la chiamata CALL.
- 6-49 **WAIT** condizione, mette in attesa il programma finchè la condizione non è verificata.
- 6-50 **TWAIT** tempo, mette in attesa il programma per il numero di secondi indicato.
- 6-59 **IF** condizione **THEN** istruzioni [**ELSE** istruzioni] **END**, classica struttura if-then-else (NB: ELSE e END devono sempre comparire in una riga a sè stante).
- 6-61 **WHILE** condizione **DO** istruzioni **END**, classica struttura ciclica while.
- 6-63 **DO** istruzioni **UNTIL** condizione, classica struttura ciclica do-until.
- 6-65 **FOR** variabile\_ciclo = valore\_iniziale **TO** valore\_finale [**STEP**] valore\_incremento istruzioni **END**, classica struttura ciclica for.
- 6-67 **CASE**, classica struttura case/switch con variabile reale.
- 6-69 **SCASE**, classica struttura case/switch con variabile stringa.
- 6-94 **PRINT/TYPE** [numero dispositivo :] dati di stampa, stampa sul dispositivo indicato i dati di stampa (PRINT "Hello World").
- 6-105 **HERE** variabile\_posa, assegna a variabile\_posa la posa corrente del robot.
- 6-106 **POINT** variabile\_posa1 [= variabile\_posa2], definisce una variabile di posa.

### 5.3 Programma di esempio con AS

Per fare un esempio pratico con il linguaggio AS è stato creato un programma molto simile a quello creato precedentemente con il TP. In questo caso però vengono utilizzati alcuni dei vantaggi dell'AS quali, ad esempio, le variabili di posa. Certamente i comandi usati non sono gli unici utilizzabili o i più appropriati. Si è cercato semplicemente di avere un minimo di varietà essendo il fine ultimo puramente didattico.

Preventivamente alla creazione del programma si è optato per la definizione di una variabile di posa globale in formato JOINT denominata #tutorial\_centro<sup>3</sup>. Questo perchè potrebbe risultare utile anche in altri programmi, essendo una posa potenzialmente importante nel setup attuale del robot.

I passi per la definizione della variabile di posa con coordinate JOINT sono i seguenti:

1. In teach mode, portare il polso nella posa desiderata con il TP;
2. Opzionalmente digitare il comando WHERE per visualizzare i valori delle coordinate JOINT e XYZOAT;
3. Digitare il comando HERE #tutorial\_centro (se la variabile è già esistente verrà sovrascritta);
4. Il sistema propone un cambio manuale dei valori delle coordinate;
5. Premere Return per ignorare la richiesta e confermare i valori della posa;
6. Per verificare i dati della posa appena creata utilizzare il comando POINT #tutorial\_centro oppure list/l che stampa tutte le variabili di posa globali.

```
>list/l
Location
centro      339.775   -46.772  -109.049  -115.918   179.266   112.949    0.000
home        0.002    119.993   580.000    89.990     0.000   -89.999    0.000
#centro     97.678     34.180  -126.465    -2.078   -19.595   -46.851    0.000
>here #tutorial_centro
      JT1      JT2      JT3      JT4      JT5      JT6      CONV
      97.675    34.180  -126.466    -2.079   -19.595   -46.850    0.000
Change? (If not, Press RETURN only.)
>
```

Figura 25: Lista delle pose salvate e aggiunta della nuova posa #tutorial\_centro

Una volta definita la variabile di posa è possibile scrivere il programma desiderato, il cui listato è il seguente:

1. OPENI
2. HOME
3. JAPPRO #tutorial\_centro, 100
4. LMOVE #tutorial\_centro
5. CLOSEI
6. JAPPRO #tutorial\_centro, 100
7. DRAW 68,68,0,0,0,0
8. DRAW 0,0,-100,0,0,0
9. OPENI
10. DRAW 0,0,100,0,0,0
11. HOME

I passi per l’inserimento del programma nel sistema AS tramite KRterm sono i seguenti:

<sup>3</sup> Era già presente una variabile denominata semplicemente “centro”, è stato quindi scelto questo nome per una maggiore congruenza con le immagini mostrate.

1. Entrare nella modalità editor digitando EDIT tutorial;
2. Digitare la prima istruzione del programma elencato sopra (OPENI);
3. Premere RETURN;
4. Tornare al punto 2 e inserire le rimanenti istruzioni fino al punto 11;
5. Digitare e per terminare la creazione e uscire dalla modalità editor.

Per verificare il programma appena creato digitare list/p tutorial che mostrerà l'output di Figura 26.

Per modificare o aggiungere passi al programma è possibile utilizzare una variante del comando EDIT in cui si indica il numero del passo da cui iniziare le modifiche: EDIT tutorial, N. Scrivendo dunque un nuovo comando si andrà a sovrascrivere quello precedente. Se invece si vuole inserir un passo prima di quello corrente è possibile premere "i" per entrare nella modalità inserimento da cui si potrà poi uscire premendo "e".

```
list/p tutorial
PROGRAM tutorial()
1      OPENI
2      HOME
3      JAPPRO #tutorial_centro,100
4      LMOVE #tutorial_centro
5      CLOSEI
6      JAPPRO #tutorial_centro,100
7      DRAW 68,68,0,0,0,0
8      DRAW 0,0,-100,0,0,0
9      OPENI
10     DRAW 0,0,100,0,0,0
11     HOME
      .END
>
```

Figura 26: Listato del programma "tutorial"

## 6 Risorse per approfondire

Sul pc portatile dedicato all'utilizzo del robot si possono trovare i seguenti manuali in formato PDF:

- Manuale Operativo: manuale dettagliato di tutte le operazioni eseguibili sul robot
  - 90203-1104DIA Manuale operativo.pdf
  - 90203-1104DEC\_Operation Manual (E series).pdf
- Manuale Linguaggio AS: manuale approfondito del linguaggio proprietario Kawasaki e dei suoi comandi.
  - 90209-1022DIA Linguaggio AS.pdf
  - 90209-1022DEB\_AS Language Manual (E series).pdf
- Manuale Antinfortunistica: manuale di base in materia di sicurezza delle persone durante l'utilizzo del robot in laboratorio.
  - 90202-1001DIF Antinfortunistica.pdf

## 7 Tabella delle figure

Figura 1: Robot Kawasaki serie RS.....	3
Figura 2: Spazio di lavoro del robot RS03N.....	4
Figura 3: Movimento in base alla posizione dei giunti .....	5
Figura 4: Le coordinate di base di riferimento .....	5
Figura 5: Movimenti controllando JT1-6.....	6
Figura 6: Possibili movimenti lungo le 6 coordinate di base.....	7
Figura 7: Vista frontale del controller.....	8
Figura 8: Pannello operativo del controller .....	8
Figura 9: Il Teach Pendant o Unità Portatile.....	9
Figura 10: Retro del Teach Pendant.....	10
Figura 11: Pulsantiera del TP.....	10
Figura 12: Generica schermata del TP in teach mode .....	12
Figura 13: Esempio di schermata di errore visualizzata sul TP.....	13
Figura 14: Attenzione: se non si resettano gli errori sarà impossibile continuare l'insegnamento .....	13
Figura 15: Passi per la creazione di un programma tramite TP.....	17
Figura 16: Esempio di percorso composto da tre movimenti lineari e uno circolare .....	17
Figura 17: Formato delle istruzioni dei programmi inseriti tramite TP.....	18
Figura 18: Listato del programma inserito tramite TP .....	20
Figura 19: Schermata iniziale di KRterm.....	22
Figura 20: Menu Com di KRterm.....	23
Figura 21: Finestra di dialogo per la scelta della connessione.....	23
Figura 22: Connessione correttamente eseguita dopo il login .....	24
Figura 23: Output di esempio del comando WHERE.....	26
Figura 24: Tabella dei comandi dell'editor .....	27
Figura 25: Lista delle pose salvate e aggiunta della nuova posa #tutorial_centro.....	29
Figura 26: Listato del programma "tutorial" .....	30