



**UNIVERSITÀ DI BRESCIA**  
**FACOLTÀ DI INGEGNERIA**  
Dipartimento di Elettronica per l'Automazione

**Laboratorio di Robotica Avanzata**  
**Advanced Robotics Laboratory**

Corso di Robot Industriali e di Servizio  
(Prof. Riccardo Cassinis)

**Implementazione programma  
dimostrativo per robot antropomorfo  
kawasaki RSo3N**

**Elaborato di esame di:**

**Marco Tebaldi, Eros Testi**

Consegnato il:

**09 Luglio 2012**



# 1. Introduzione

In questo elaborato verrà descritto il lavoro svolto per la programmazione del robot Kawasaki RS03N presente nel Laboratorio di Robotica Avanzata della Facoltà di Ingegneria dell'Università di Brescia.

L'obiettivo è quello di implementare un software dimostrativo in grado di mettere in luce le principali capacità di lavoro del robot antropomorfo RS03N.

Per fare questo si è deciso di far eseguire al robot delle operazioni di Pick & Place, molto diffuse in ambiente industriale, che consistono nel prelevare un oggetto da una posizione e nel depositarlo in un'altra.

Il robot lavora con palline di legno, posizionate sopra a tappi di bottiglia equispaziati, questi ultimi sono fissati su una tavola di legno per rendere il tutto più stabile (l'ambiente in cui lavora il robot sarà descritto più dettagliatamente nel paragrafo 3.1)

Di seguito una breve descrizione delle operazioni svolte dal programma:

- Spostamento di una colonna di 4 palline nella colonna adiacente.
- Ripetizione dell'operazione precedente fino al raggiungimento dell'ultima colonna disponibile.
- Costruzione di una piramide di cinque palline al centro del pallet.
- Ritorno nella posizione home.

Per svolgere il lavoro è stato necessario fare un breve studio introduttivo per quanto riguarda il linguaggio di programmazione del robot, viene usato infatti il linguaggio AS di proprietà dell'azienda Kawasaki, creato per la programmazione dei robot del marchio.

## 2. Il Robot RS03N

In questo paragrafo verranno illustrate le principali caratteristiche fisiche del robot.



Fig.1-Foto del robot RS03N

Il robot Kawasaki RS03N è di tipo antropomorfo Fig.1, è il più piccolo della serie RS offerta da Kawasaki, ma nonostante questo ha già capacità di movimento sovrapponibili a quelle dei suoi fratelli maggiori, rendendolo ideale per scopi didattici o per lavori in cui non siano richiesti spazi di lavoro di grandi dimensioni e forze di lavoro di elevata entità.

Questo robot è dotato di 6 assi, può quindi fare movimenti anche molto complessi, è inoltre munito di encoder assoluti che controllano la posizione dei motori non rendendo più necessarie fasi di taratura degli encoder come invece avviene per il robot IBM presente nel laboratorio di robotica.

Viene fornito dall'azienda corredato del suo calcolatore di controllo che serve per comunicare con il robot e per programmarlo.



Fig.2-Teach pendant

Il calcolatore permette inoltre di controllare e programmare il robot anche con un Teach Pendant Fig.2

Questo dispositivo consente la programmazione "Teach In" ossia la possibilità di muovere manualmente il robot, registrarne i movimenti e in seguito riprodurli in modalità automatica.

In questo lavoro il Teach Pendant è stato utilizzato solo per il salvataggio delle pose d'interesse (come meglio sarà spiegato nel paragrafo 3.2) e non per la programmazione effettiva del robot che è invece stata realizzata utilizzando il linguaggio AS.

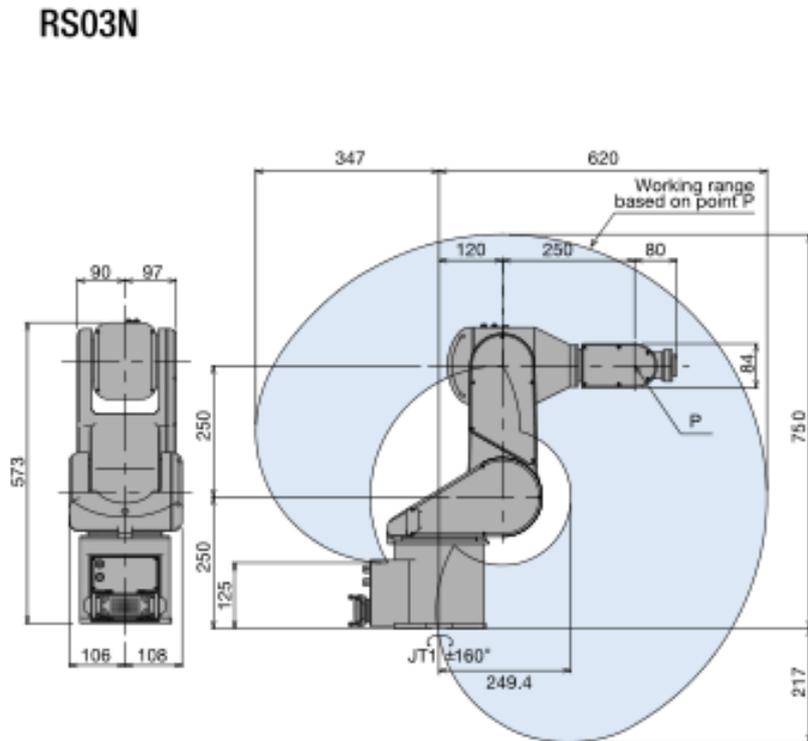
Per una comprensione maggiore delle possibilità offerte dal Teach Pendant si rimanda al manuale del robot [1] o per un trattazione più rapida alla relazione di Daniele Petre [2]).

Di seguito sono riportate le principali caratteristiche del robot risultate di interesse per il progetto.

- Numero assi	6	
- Carico utile	3	kg
- Max distanza raggiungibile orizzontalmente	620	mm
- Max distanza raggiungibile verticalmente	967	mm
- Precisione nella ripetibilità delle pose	$\pm 0.02$	mm
- Velocità massima	6000	mm/s

Per un elenco più completo si rimanda al manuale del robot [1]).

In Fig.3 è riportato lo spazio di lavoro totale nel quale il robot può operare.



**Fig.3- Spazio di lavoro del robot RS03N**

La massima distanza raggiungibile orizzontalmente tuttavia è ridotta nel caso si voglia utilizzare la pinza sempre ortogonalmente al piano.

Questo è stato il principale problema nella definizione dell'area di lavoro, tale limitazione infatti ci ha costretti a far lavorare il robot solo in una parte della tavola preparata per l'esperimento e in particolare solo sulle prime quattro righe in quanto la quinta non è raggiungibile in ogni sua posizione, ma solo a centro tavola.

### 3. Programma

#### 3.1. Schematizzazione del piano di lavoro

Come già accennato precedentemente per la realizzazione del programma si è utilizzata una tavola di legno sulla quale sono stati fissati dei tappi di bottiglia equispaziati di 70mm, che facilitano le operazioni di presa e rilascio delle palline.

In Fig.4 è visibile una foto dell'ambiente di lavoro del robot.



Fig.4- Ambiente di lavoro del robot all'interno dell' A.R.L

Per meglio comprendere la spiegazione delle operazioni svolte dal programma in Fig.5 viene mostrata una schematizzazione del piano di lavoro.

Si noti come il pallet risulti posizionato parallelamente al sistema di coordinate XY di base del Robot e costituito da 4 righe (quelle cioè effettivamente utilizzate dal programma per le ragioni già spiegate precedentemente) e 7 colonne di supporti equispaziati di 70mm.

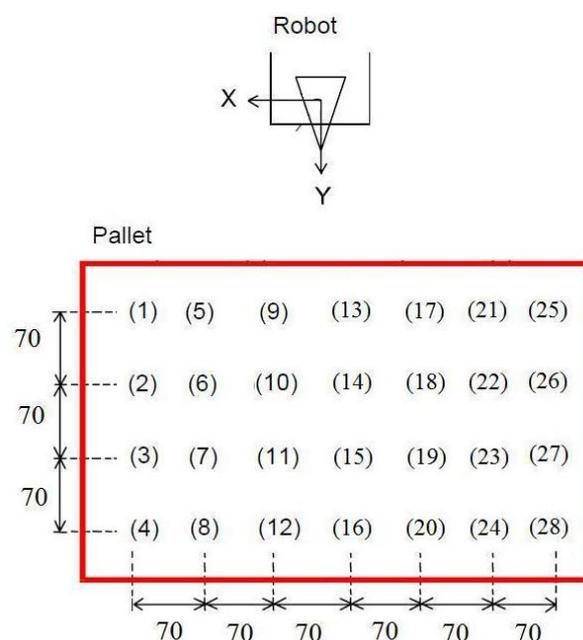


Fig.5- Schema del piano di lavoro

### 3.2. Definizione Pose

Per rendere la programmazione indipendente dalla posizione del pallet rispetto al robot, si è deciso di utilizzare la trasformazione composta conosciuta anche come trasformazione relativa.

Una posa può essere definita relativamente ad un punto di riferimento: in questo modo la relazione fra la posa e il punto di riferimento resta invariata anche se il punto di riferimento è ridefinito.

In altre parole, definendo come punto di riferimento un vertice del pallet e riferendo a quest'ultimo tutte le altre pose, il funzionamento del programma è garantito anche a seguito di un riposizionamento della tavola semplicemente modificando la posa di riferimento.

Questo risulta essere molto vantaggioso in quanto non risulta necessario modificare il software e ridefinire tutte le pose utilizzate bensì è sufficiente aggiornare la posizione che il pallet possiede rispetto alla base del robot.

Si è quindi proceduto definendo come posa di riferimento:

- Plate1 (posizione 25) che contiene le coordinate del pallet rispetto a quelle di base del robot

e come posa relativa:

- Object a che contiene le coordinate del punto di presa della prima pallina (posizione 1) rispetto a Plate1.

Per meglio comprendere il concetto in Fig.6 e Fig.7 viene illustrato lo schema esplicativo:

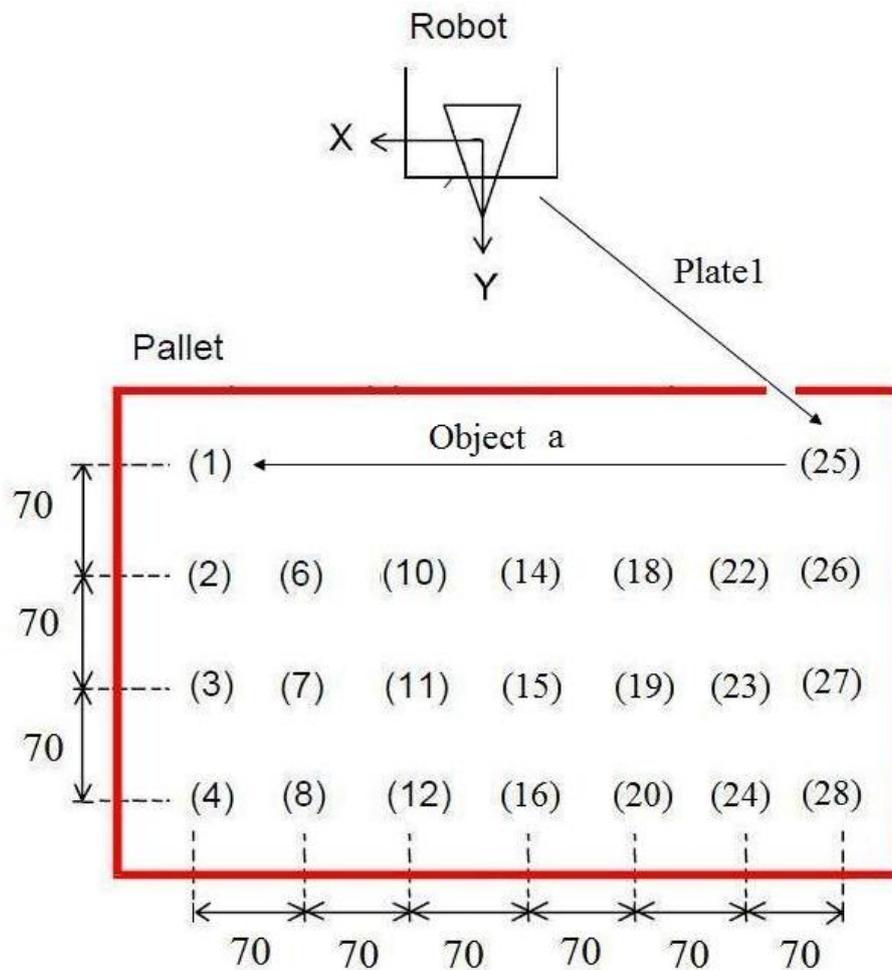
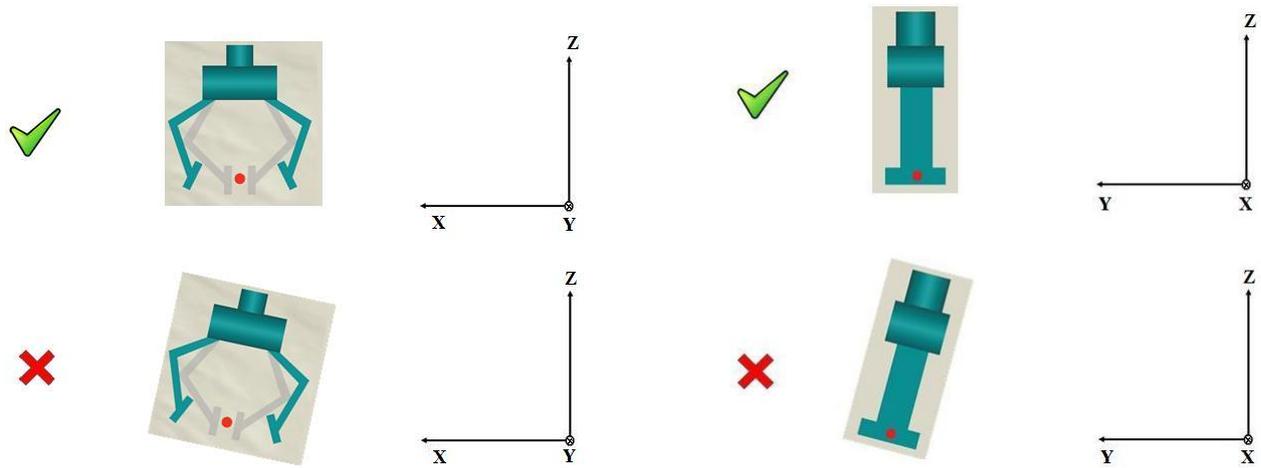


Fig.6-Schema pose



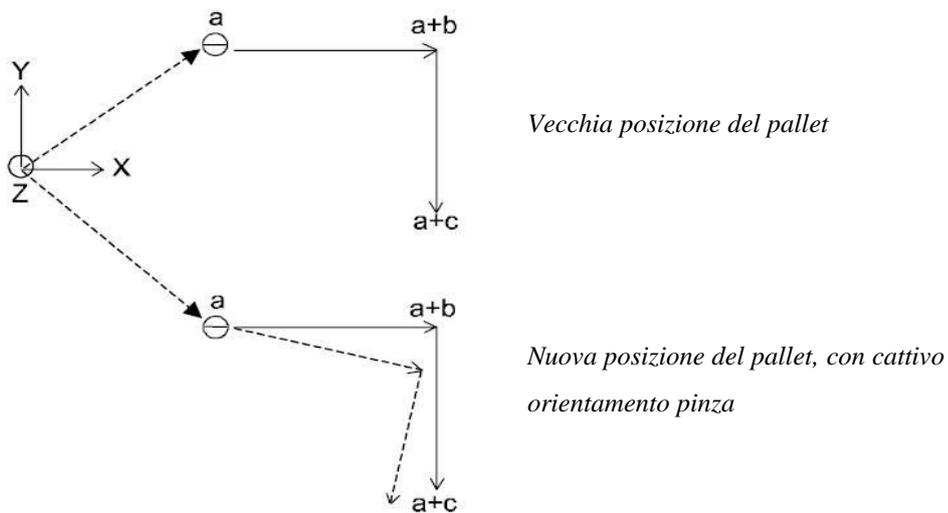
E' tuttavia necessario porre attenzione in fase di assegnamento delle nuove coordinate alla variabile plate che la pinza sia orientata correttamente che risulti cioè parallela agli assi x-y del pallet come mostrato in Fig.8



**Fig.8-Posizionamento Pinza**

Se questo non accade il robot successivamente si muoverà in maniera errata come meglio si comprende dalla Fig.9 nella quale vengono mostrati i movimenti del robot (linea tratteggiata) a seguito della riassegnazione della posa di riferimento (a) con la pinza orientata in modo scorretto.

In particolare si nota come le generiche pose relative (a+b) e (a+c) risultino essere spostate rispetto a quanto invece ci si aspetterebbe, andando ad alterare il corretto funzionamento del software.



**Fig.9-Movimenti robot**

Nel elaborato si è pertanto posta molta attenzione al corretto orientamento della pinza in fase di definizione della posa Plate1.

Come sviluppo futuro è possibile risolvere questo inconveniente rendendo la riassegnazione indipendente dalla postura del polso utilizzando la funzionalità FRAME che risulta spiegata nel manuale [2].

### 3.3. Spiegazione delle operazioni svolte dal programma

Il programma realizzato sposta le 4 palline presenti sulla prima colonna a sinistra nella colonna adiacente e prosegue fino ad averle spostate nell'ultima a destra Fig.10.

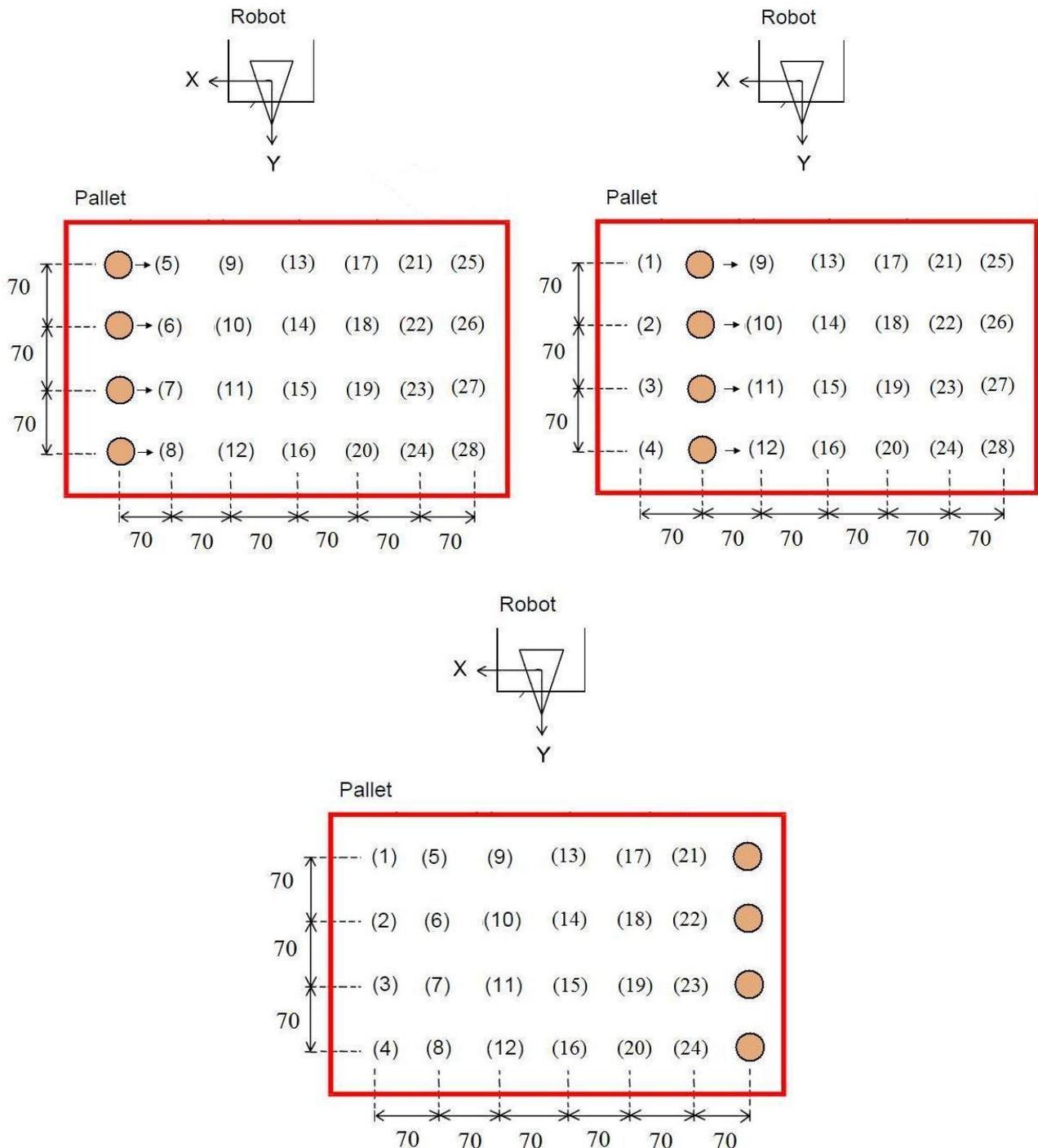


Fig.10-Evolversi delle posizioni delle palline durante il programma

In particolare prende la pallina in posizione (1) e la pone in posizione (5), poi preleva la pallina in posizione (2) e la sposta in posizione (6) e procede in questo modo fino a completare la colonna.

A questo punto sposta la pallina in posizione (5) nella posizione (9) e completa la colonna.

Arrivato in fondo costruisce con cinque palline una piramide al centro della tavola e termina.

In particolare nel pallet sono stati aggiunti due tappi corrispondenti alle posizioni (29) e (30) della Fig.11 i quali sono distanziati di 35 mm lungo l'asse x e lungo l'asse y che consentono di costruire la base della piramide.

La posizione (31) indica invece la posizione della quinta pallina, si ricorda che la quinta riga non è coperta completamente dallo spazio di lavoro del robot se la pinza rimane verticale, ma nella posizione (31) non si hanno problemi.

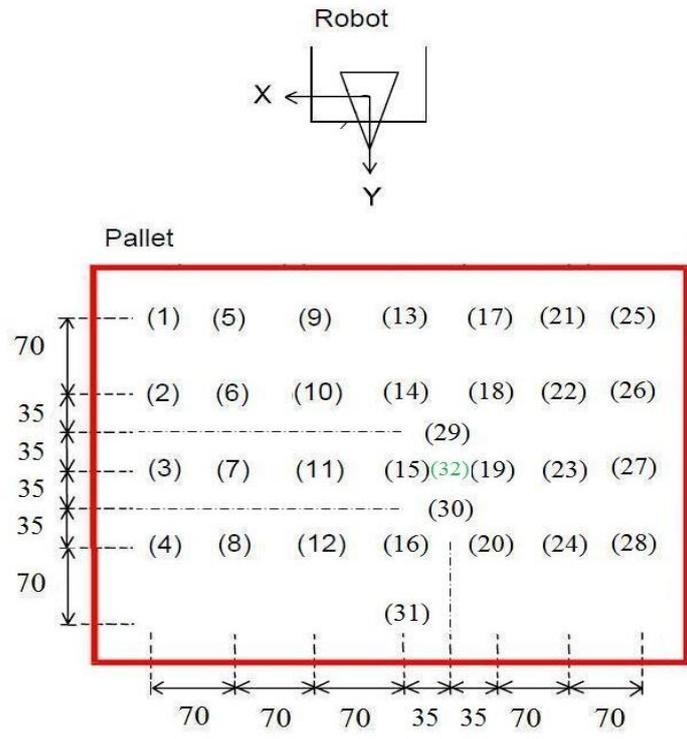


Fig.11- Posizioni per costruzione piramide

Nella Fig.12 viene mostrato il procedimento seguito dal robot nella realizzazione della piramide.

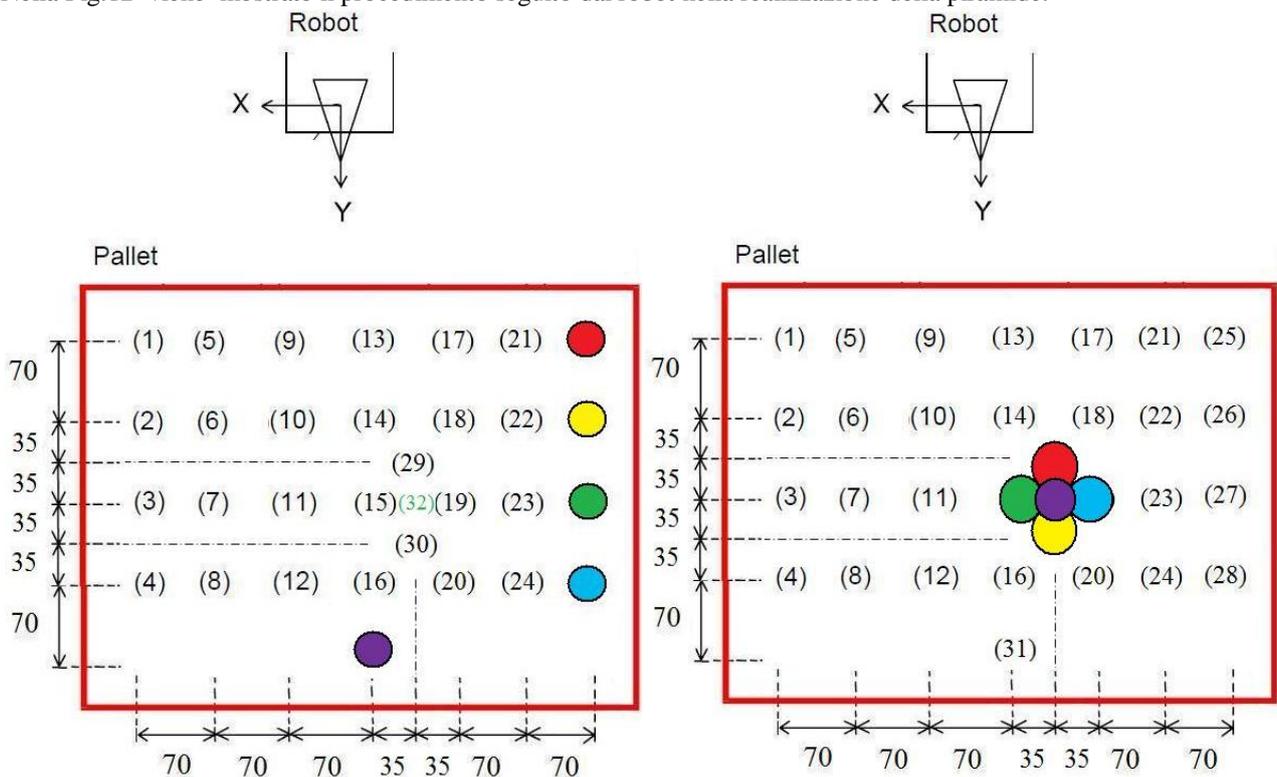


Fig.12-Posizione palline per costruzione piramide

### 3.4. Il codice

L'idea generale sulla quale si basa la realizzazione del programma è di sfruttare le posizioni fisse del pallet, cioè le posizioni dei tappi i quali risultano distanziati tra loro (come già mostrato nelle figure precedenti) di 70mm lungo entrambi gli assi.

In questo modo risulta possibile descrivere completamente le posizioni dell'intero pallet semplicemente sommando all'unica posa plate1+objecta gli incrementi  $x_s=70\text{mm}$  e  $y_s=70\text{mm}$ , in questo modo è anche garantita la relatività delle pose.

L'istruzione che consente quanto detto è la seguente:

**SHIFT**( *nome variabile a cui sommare gli incrementi* **BY** *Incremento X, Incremento Y, Incremento Z*)

Si mostra un esempio per comprenderne meglio il funzionamento:

**esempio**

Data la variabile "x" i cui valori di trasformazione sono (200,150,100,10,20,30), a seguito dell'istruzione: `POINT y=SHIFT(x BY 5,-5,10)`

I valori di "x" diventano(205,145,110,10,20,30) e vengono assegnati alla variabile "y" tramite l'istruzione POINT.

Si è definita per comodità la variabile locale "put" indicante la posizione che il robot di volta in volta deve raggiungere, quest'ultima viene inizializzata alla posizione (1) del pallet nel seguente modo:

`POINT put = plate1+objecta`

La fase di spostamento delle palline dalla prima all'ultima colonna è stata realizzata mediante due cicli for annidati, uno per scorrere le righe ed uno per scorrere le colonne (fare riferimento al software mostrato in Fig.25).

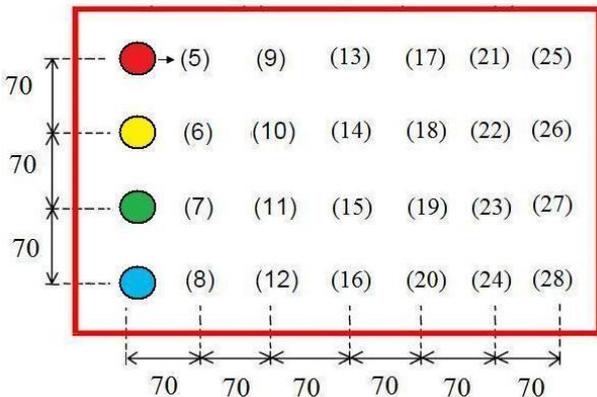


Fig.13

Durante la prima iterazione il robot si porta sulla pallina rossa indicata da "put", preleva la pallina, calcola il nuovo valore di "put":

`POINT put = SHIFT(put By -xs,0,0)`

indicante la posizione (5), si sposta a "put" e rilascia la pallina Fig.13.

Procede poi con l'aggiornare nuovamente "put":

`POINT put = SHIFT(put By xs,ys,0)`

in modo da preparare per l'iterazione successiva del for che scorre le righe la posizione sulla pallina gialla cioè sulla posizione (2) Fig.14.

A questo punto all'iterazione successiva del ciclo delle righe il robot si porterà sulla pallina gialla, la preleverà, calcolerà il nuovo valore di "put" indicante la posizione (6) esattamente come fatto prima, si sposterà e rilascerà la pallina.

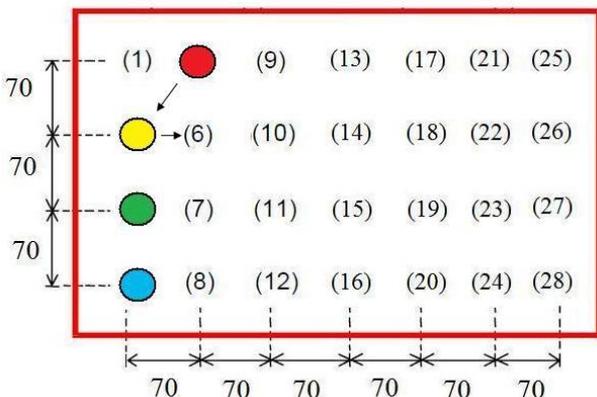


Fig.14

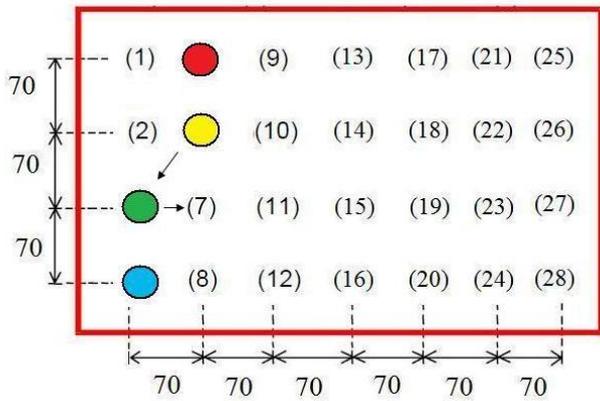


Fig. 15

Poi nuovamente verrà preparata la variabile “put” per l’iterazione successiva del for delle righe indicante la pallina verde Fig.15

Il ciclo procede in questo modo fino a quando tutte le palline sono state spostate nella seconda colonna Fig.16, Fig.17.

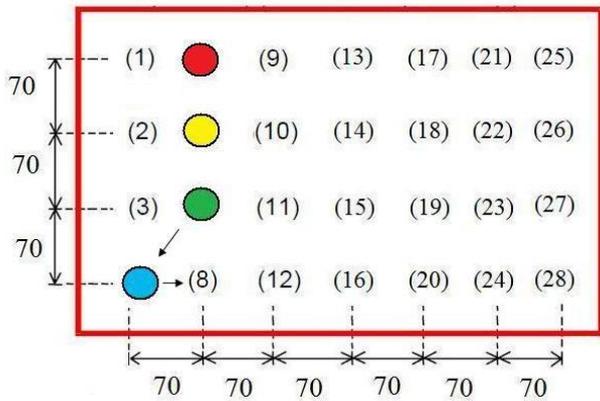


Fig. 16

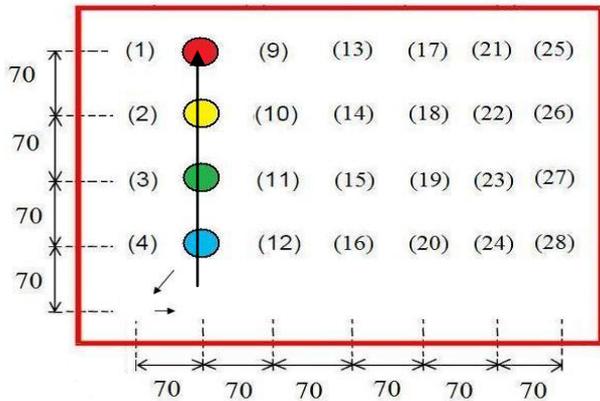


Fig. 17

Uscito dal ciclo for che scorre le righe, la variabile “put” indicherà il primo elemento della quinta riga pertanto il robot aggiorna tale variabile con la seguente istruzione:

`POINT put = SHIFT(put By -xs,-4*ys,0)`

dopo la quale la posizione indicata è quella corrispondente alla pallina rossa Fig.17

Alla successiva iterazione il robot si porterà sulla pallina rossa e procederà nello spostare l’intera colonna in quella adiacente ripetendo le stesse operazioni appena spiegate.

Una volta spostate le palline nell’ultima colonna anche il ciclo for delle colonne può terminare, la variabile “put” a questo punto indicherà la posizione (25) cioè la pallina rossa Fig.18

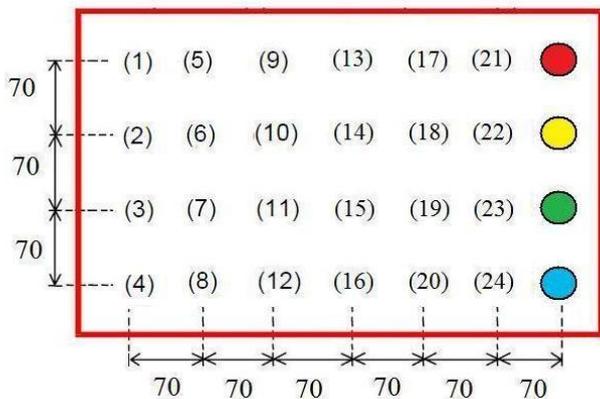


Fig. 18



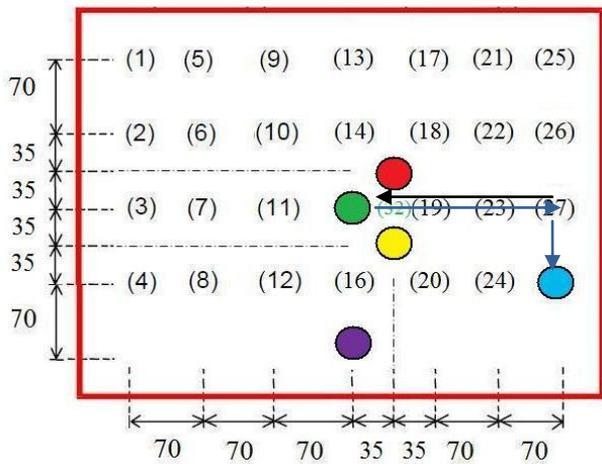


Fig. 22

Il programma procede allo stesso modo spostando le palline come mostrato nelle figure Fig.22, Fig.23, Fig.24 fino a completare la piramide.

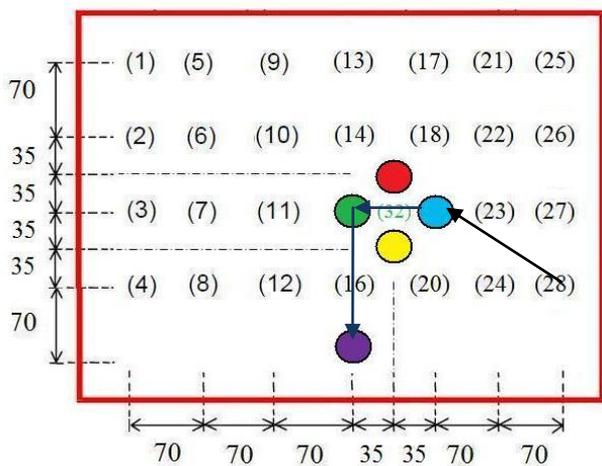


Fig. 23

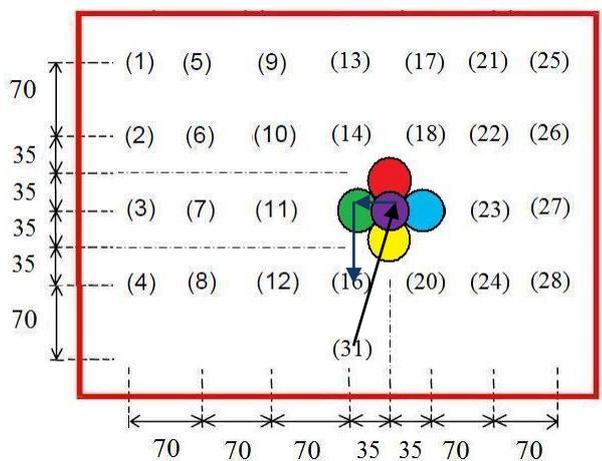


Fig.24

Viene di seguito mostrato Fig.25 il codice in linguaggio AS che permette al robot di effettuare le operazioni spiegate; lo analizzeremo con un metodo step by step per rendere il più chiaro possibile il lavoro svolto.

```

1 PROGRAM ttpick3()#10
2 HOME
3
4 OPENI
5 row.max = 4
6 col.max = 6
7 xs = 70
8 ys = 70
9 OPENI
10 SPEED 40 ALWAYS
11
12 POINT put = plate1+objecta
13
14 FOR col = 1 TO col.max
15 FOR row = 1 TO row.max
16 OPENI
17 JAPPRO put,60
18 LMOVE put
19 CLOSEI
20 LDEPART 60
21 POINT put = SHIFT(put BY -xs,0,0)
22 JAPPRO put,60
23 LMOVE put
24 OPENI
25 LDEPART 60
26 POINT put = SHIFT(put BY xs,ys,0)
27 END
28 POINT put = SHIFT(put BY -xs,-4*ys,0)
29 END
30 xd = 35
31 yd = 35
32 i = 0
33
34 WHILE i<5 DO
35 OPENI
36 JAPPRO put,60
37
38 LMOVE put
39 CLOSEI
40 LDEPART 60
41 i = i+1
42 IF i==1 THEN
43 POINT put = SHIFT(put BY 5*xd,3*yd,0)
44 JAPPRO put,60
45 LMOVE put
46 OPENI
47 LDEPART 60
48 POINT put = SHIFT(put BY -5*xd,-yd,0)
49 END
50 IF i==2 THEN
51 POINT put = SHIFT(put BY 5*xd,3*yd,0)
52 JAPPRO put,60
53 LMOVE put
54 OPENI
55 LDEPART 60
56 POINT put = SHIFT(put BY -5*xd,-yd,0)
57 END
58 IF i==3 THEN
59 POINT put = SHIFT(put BY 6*xd,0,0)
60 JAPPRO put,60
61 LMOVE put
62 OPENI
63 LDEPART 60
64 POINT put = SHIFT(put BY -6*xd,ys,0)
65 END
66 IF i==4 THEN
67 POINT put = SHIFT(put BY 4*xd,-ys,0)
68 JAPPRO put,60
69 LMOVE put
70 OPENI
71 LDEPART 60
72 POINT put = SHIFT(put BY xs,2*ys,0)
73 END
74 IF i==5 THEN
75 POINT put = SHIFT(put BY -xd,-2*ys,0)
76 JAPPRO put,60
77 LAPPRO put,50
78 OPENI
79 LDEPART 60
80 POINT put = SHIFT(put BY xd,ys,0)
81 END
82 HOME
83 .END

```

```

%nome del programma
%richiamo il robot nella posizione di home,
completamente esteso verso l'alto
%apro la pinza per sicurezza
%definisco variabile locale con num max righe
%definisco variabile locale con num max colonne
%definisco variabile locale con spaziatura tappi lungo x
%definisco variabile locale con spaziatura tappi lungo y
%mi assicuro che la pinza sia aperta
%taro la velocità al 40% per non sollecitare troppo il tavolo
su cui è fissato il robot
%alla variabile locale put assegno il punto in cui si trova
la prima pallina posizione(1)
%imposto il for per le colonne
%imposto il for per le righe
%apro la pinza(operazione necessaria dal 2^ ciclo in poi)
%mi porto con traiettoria interpolata sul punto put +60(in Z)
%mi muovo linearmente verso il basso fino a put(punto presa oggetto)
%chiudo la pinza
%mi alzo linearmente lungo Z di 60
%aggiorno put, stessa riga si incrementa la colonna
%mi porto con traiettoria interpolata sul punto put +60(in Z)
%mi muovo linearmente verso il basso fino a put(punto rilascio oggetto)
%apro la pinza e rilascio la palla
%mi alzo linearmente lungo Z di 60
%aggiorno put, torno alla colonna precedente ma incremento la riga
%chiudo ciclo for delle righe lungo y, colonna spostata
%agg. put,colonna corrente riga 1, per spostare la successiva colonna
%chiudo ciclo delle colonne lungo x, palline nell'ultima colonna
%definisco spostamento dimezzato in x per piramide
%definisco spostamento dimezzato in y per piramide
%inizializzo variabile di appoggio per controllo cicli
%controllo se sono all'ultimo ciclo se si mi fermo
%apro la pinza
%mi porto con traiettoria interpolata sopra il punto put +60(in Z)
la prima volta quindi sopra la posizione (25)
%mi muovo linearmente verso il basso fino a put(punto presa pallina)
%chiudo la pinza
%mi alzo linearmente lungo Z di 60
%incremento il contatore
%controllo contatore per verificare a che punto della piramide sono
%aggiorno put con il nuovo punto cioè la posizione (29)
%mi porto con traiettoria interpolata sul punto put +60(in Z)
%mi muovo linearmente verso il basso fino a put(punto presa oggetto)
%apro la pinza e rilascio la pallina
%mi alzo linearmente lungo Z di 60
%aggiorno put al nuovo punto, ultima colonna seconda riga pos(26)
%controllo contatore per verificare a che punto della piramide sono
%aggiorno put con il nuovo punto cioè la posizione (30)
%aggiorno put al nuovo punto, ultima colonna terza riga pos(27)
%controllo contatore per verificare a che punto della piramide sono
%aggiorno put con il nuovo punto cioè la posizione (15)
%aggiorno put al nuovo punto, ultima colonna quarta riga pos(28)
%controllo contatore per verificare a che punto della piramide sono
%aggiorno put con il nuovo punto cioè la posizione (19)
%aggiorno put al nuovo punto, posizione (31) quinta pallina
%controllo contatore per verificare a che punto della piramide sono
%aggiorno put con il nuovo punto cioè la posizione (32)
%aggiorno put al nuovo punto, posizione (16) centro tavola
%rimando il robot a casa

```

Fig.25- Codice commentato step by step

## 4. Bibliografia

- [1] Kawasaki heavy industries, *Manuale operativo, robot kawasaki controller Serie E*, prima edizione ed., Marzo 2010.  
URL: <http://www.ing.unibs.it/~arl/docs/documentation/Kawasaki.html>
- [2] Kawasaki heavy industries, *Manuale di riferimento del linguaggio AS, robot kawasaki controller Serie E*, prima edizione ed., Marzo 2010.  
URL: <http://www.ing.unibs.it/~arl/docs/documentation/Kawasaki.html>
- [3] Daniele Petre, "Guida introduttiva all'utilizzo del robot Kawasaki RS03N," facolta di ingegneria di Brescia, 2010/2011.  
URL: [http://www.ing.unibs.it/~arl/docs/projects/Doc\\_07.pdf](http://www.ing.unibs.it/~arl/docs/projects/Doc_07.pdf)

# Indice

1. Introduzione .....	2
2. Il Robot RS03N.....	2
3. Programma .....	4
3.1. Schematizzazione del piano di lavoro .....	4
3.2. Definizione Pose .....	5
3.3. Spiegazione delle operazioni svolte dal programma .....	8
3.4. Il codice .....	10
4. Bibliografia .....	15