

# Programma di pallettizzazione

Una delle operazioni più conosciute e sviluppate nell'industria è la pallettizzazione. Consiste nel raggruppare o immagazzinare pezzi o prodotti in modo organizzato per l'ottimizzazione degli spazi e dei tempi. Nella figura è mostrato un esempio reale. Nella sezione precedente sono stati confezionati un paio di programmi semplificati di pallettizzazione di pezzi utilizzando il linguaggio VAL. Esistono sul mercato costruttori

che sviluppano robot e strumenti speciali per ottimizzare i compiti di pallettizzazione. Kawasaki ha sviluppato il robot UD100 che è capace di pallettizzare 1.000 scatole con peso di 100 Kg ognuna. Nella successione delle figure sono mostrati gli stadi da seguire per implementare un processo di pallettizzazione e che sono i seguenti:

1°. Ingresso dei dati, quali la dimensione dei pezzi, tipo di pallet da organizzare, ecc.  
2°. Registrazione dei punti di

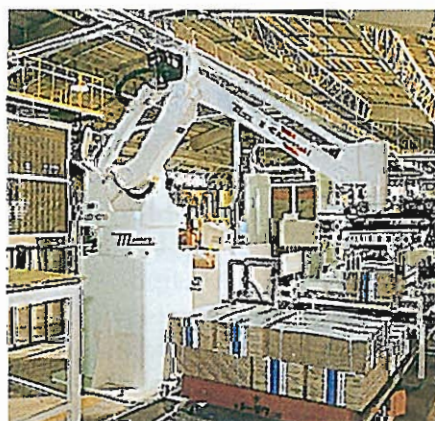
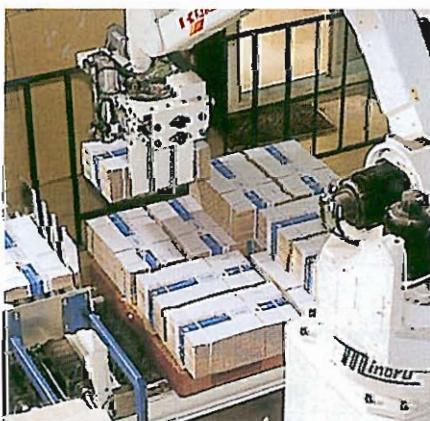
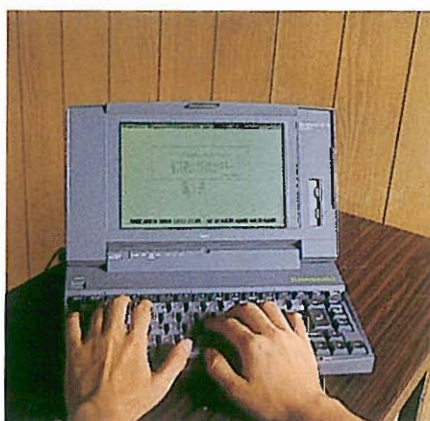
acquisizione. Mediante la pistola manuale si registrano i punti in cui devono essere presi i pezzi.

3°. Registrazione dei punti di destinazione.

4°. Manipolatori speciali adattati ai prodotti maneggiati. Kawasaki, come altri costruttori di robot, dispone di tutti i tipi di manipolatori per adattarsi a qualsiasi pezzo da pallettizzare.

5°. Lavoro reale di pallettizzazione.

6°. Risultati finali.



Fasi del progetto di un processo di pallettizzazione utilizzando il robot UD100 di Kawasaki.



## Programma di pallettizzazione con due alimentatori

In precedenza avevamo proposto e risolto un programma in linguaggio VAL per un sistema di pallettizzazione semplificato, la cui principale caratteristica era il funzionamento con un solo alimentatore; in questo caso il sistema dispone di due alimentatori, uno che lascia i pezzi nella zona A, e l'altro nella zona B. Il nuovo programma da costruire dovrà tenere conto dei due alimentatori e si chiamerà PUMA2. Come ricorderete c'era un "sensore di pezzo disponibile" collegato all'ingresso 1. Ora esiste anche un altro sensore collegato all'ingresso 2 che inserisce un

| Programma PUMA2 |                         |
|-----------------|-------------------------|
|                 | OPEN                    |
| 10              | MOVE RIPOSO             |
|                 | WAIT 1                  |
|                 | WAIT -1                 |
|                 | IFSIG 2,,, THEN 500     |
|                 | MOVE AP1                |
|                 | MOVES P1                |
|                 | GOTO 510                |
| 500             | SHIFT P1 BY 150, 0, 0   |
|                 | SHIFT AP1 BY 150, 0, 0  |
|                 | MOVE AP1                |
|                 | MOVES P1                |
|                 | SHIFT AP1 BY -150, 0, 0 |
|                 | SHIFT P1 BY -150, 0, 0  |
| 510             | DELAY 0,5               |
|                 | CLOSE                   |
|                 | DELAY 0,5               |
|                 | MOVE RIPOSO             |
|                 | MOVE AP2                |
|                 | MOVES P2                |
|                 | DELAY 0,5               |
|                 | OPEN                    |
|                 | DELAY 0,5               |
|                 | MOVES AP2               |
|                 | GOTO 10                 |

livello basso quando il pezzo rilevato è nella posizione A, mentre se è nella posizione B il livello che produce il sensore è alto. Nel programma PUMA2 l'istruzione IFSIG 2,,,THEN 500, testa lo stato della linea di ingresso 2. Se è a livello alto salta alla linea etichettata con il numero 500. Le tre virgole che seguono il 2 significano che è possibile testare contemporaneamente quattro ingressi con la stessa istruzione. L'istruzione IFSIG, nel caso non si compia la condizione specifica, non ferma l'esecuzione del programma, come l'istruzione WAIT, ma lascia che questo continui normalmente.

L'istruzione SHIFT somma o sottrae, secondo il segno, una distanza espressa in millimetri alle coordinate X, Y, Z. Non origina il movimento del manipolatore al punto dato che semplicemente modifica il valore dato al punto. Siccome gli alimentatori A e B si trovano separati a una distanza di 150 mm, sull'asse X si spostano i punti AP1 e P1 a questa distanza nel senso positivo dell'asse X.

## Programma migliorato PUMA2M

La soluzione precedente può essere migliorata utilizzando un altro tipo di istruzione del linguaggio VAL. In questo nuovo programma si utilizzano subroutines per differenziare il trattamento di ogni alimentatore. Le subroutines sono programmi che si chiamano con GOSUB

e terminano con RETURN, che restituisce il controllo al programma principale da dove è partita la chiamata. Utilizzeremo anche le istruzioni DEPART/APRO e DEPARTS/APROS, con le quali si risparmiano due passi del programma. L'istruzione APRO avvicina la pinza al punto indicato dalla distanza che appare come argomento.

L'istruzione DEPART fa il contrario, ossia, lo allontana. Le istruzioni APROS e DEPARTS realizzano le stesse funzioni però eseguendo i movimenti in linea retta.

| Programma PUMA2M  |                        |
|-------------------|------------------------|
|                   | OPEN                   |
| 10                | MOVE RIPOSO            |
|                   | WAIT 1                 |
|                   | WAIT -1                |
|                   | IFSIG 2,,, THEN 500    |
|                   | GOSUB PEZZOA           |
|                   | GOTO 510               |
| 500               | GOSUB PEZZOB           |
| 510               | DELAY 0,5              |
|                   | CLOSE                  |
|                   | DELAY 0,5              |
|                   | MOVE RIPOSO            |
|                   | APPRO P2,100           |
|                   | MOVES P2               |
|                   | DELAY 0,5              |
|                   | OPEN                   |
|                   | DELAY 0,5              |
|                   | DEPARTS 100            |
|                   | GOTO 10                |
| Subroutine PEZZOA |                        |
|                   | APPRO P1,100           |
|                   | MOVES P1               |
|                   | RETURN                 |
| Subroutine PEZZOB |                        |
|                   | SHIFT P1 BY 150, 0, 0  |
|                   | APPRO P1,100           |
|                   | MOVES P1               |
|                   | SHIFT P1 BY -150, 0, 0 |
|                   | RETURN                 |