

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI
INDUSTRIALI
DIPARTIMENTO DI INGEGNERIA INDUSTRIALE
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA
MECCATRONICA

TESI DI LAUREA MAGISTRALE

**Sviluppo software per il controllo di un
manipolatore collaborativo impiegato
come pallettizzatore**

Relatore: Prof. Giovanni Boschetti

Laureando: Matteo Poli
2061927

ANNO ACCADEMICO: 2023-24

SOMMARIO

Lo scopo del seguente elaborato è lo sviluppo di un software di controllo per un manipolatore che funzioni da pallettizzatore senza la necessità di impiegare barriere protettive che lo separino dagli operatori umani.

Questo robot deve quindi essere in grado, in collaborazione con un controllore PLC, di compiere operazioni di pick and place in maniera ciclica fino a quando non comunicato diversamente dal controllore stesso e deve essere in grado di lavorare senza causare pericolo per gli operatori.

La tesi è stata svolta nell'azienda TechPa S.p.A impiegando il manipolatore FANUC CRX-25iA, un manipolatore collaborativo della Fanuc che implementa al suo interno delle funzioni di sicurezza che gli permettono di lavorare a stretto contatto con gli operatori umani.

Nel seguente elaborato verranno dapprima introdotti i robot e i manipolatori collaborativi per poi introdurre i PLC. Successivamente si parlerà nello specifico del manipolatore in esame, andandone ad analizzare la scheda tecnica per poi passare, infine, all'analisi del codice sviluppato, sia lato robot che PLC.

INDICE

1	Introduzione	1
2	Robotica e industria	3
2.1	La storia della robotica	3
2.2	Definizione di Robot	4
2.2.1	Struttura dei robot	5
2.2.2	I robot collaborativi, CoBot	7
2.2.3	Analisi CoBot presenti in commercio	10
3	PLC e Automazione	17
3.1	Storia del PLC	17
3.2	Funzionamento dei PLC	18
3.2.1	Programmazione	18
3.2.2	Software	19
3.3	PLC in commercio	20
4	FANUC CRX-25 <i>iA</i> e Gripper a vuoto	27
4.1	Descrizione	27
4.2	Caratteristiche tecniche	29
4.2.1	Controller	31
4.2.2	Sicurezza	33
4.2.3	Sensori	34
4.2.4	Polso	36
4.2.5	Base	37
4.2.6	Braccio Robotico	37
4.3	Programmazione	39
4.3.1	Dispositivi per la programmazione	39
4.3.2	Analisi codice	42
4.4	Analisi cinematica	50
4.4.1	Convenzione di Denavit-Hartenberg	51
4.4.2	Analisi cinematica con Denavit-Hartenberg	52
4.4.3	Singularità	53
4.5	Utensile	54
5	Sviluppo del software di controllo	57
5.1	Robot	57
5.1.1	Preparazione iniziale	57
5.1.2	Codice	60
5.2	PLC	67
5.2.1	Codice	68
	Conclusioni	75

ELENCO DELLE FIGURE

Figura 2.1	Robot UNIMATE	3	
Figura 2.2	Catena cinematica	6	
Figura 2.3	YuMi	11	
Figura 2.4	GoFa CRB 15000	11	
Figura 2.5	Universal Robots	13	
Figura 2.6	COMAU Robots	15	
Figura 2.7	Modello CRX-5iA	16	
Figura 3.1	PLC Allen Bradley	17	
Figura 3.2	Esempio di Logica Ladder	19	
Figura 3.3	Esempio di Testo Strutturato	19	
Figura 3.4	Esempio controllori SIEMENS	23	
Figura 3.5	Esempio controllori Rockwell Automation	24	
Figura 3.6	Esempio controllori MELSEC	26	
Figura 4.1	CoBot FANUC CRX-25iA	29	
Figura 4.2	Scheda tecnica	30	
Figura 4.3	R-30iB Mini Plus visione esterna	32	
Figura 4.4	R-30iB Mini Plus visione interna	33	
Figura 4.5	Polso	37	
Figura 4.6	Diagramma payload polso	38	
Figura 4.7	Connessione cavi CRX-25iA	39	
Figura 4.8	Configurazione unità meccanica	40	
Figura 4.9	Esempio di programma	40	
Figura 4.10	Tastierino per programmazione robot FANUC	41	
Figura 4.11	Tablet per la programmazione robot FANUC	41	
Figura 4.12	Definizione rack e segnali digitali	44	
Figura 4.13	Istruzione di movimento	45	
Figura 4.14	Esempio modalità di posizionamento	47	
Figura 4.15	Rappresentazione giunti e relativi sistemi di riferimento	52	
Figura 4.16	Tipi di singolarità cinematiche	54	
Figura 4.17	Gripper KVG60	54	
Figura 5.1	A00_AvvioRobot	68	
Figura 5.2	A04_Pinza	69	
Figura 5.3	A08_GestioneDeposito	70	
Figura 5.4	A09_GestionePrelievo	71	
Figura 5.5	Esempio collegamento uscita PLC e ingresso manipolatore	72	
Figura 5.6	Esempio collegamento ingresso PLC e uscita manipolatore	72	
Figura 5.7	A98_ScambioDatiFanuc	72	

ELENCO DELLE TABELLE

Tabella 4.1	Specifiche Robot, Velocità, Range movimento	29
Tabella 4.2	Parametri di Denavit-Hartenberg del FANUC CRX-25iA	53
Tabella 5.1	Digital Input	57
Tabella 5.2	Digital Output	57
Tabella 5.3	DI Utilizzati	58
Tabella 5.4	DO Utilizzati	58
Tabella 5.5	Group Input	59
Tabella 5.6	Group Output	59
Tabella 5.7	GI Utilizzati	60

INTRODUZIONE

Con l'avvento dei principi dell'Industria 4.0, che pone al centro l'automazione, sempre più aziende stanno adottando robot per ottimizzare e rendere più sicuri i luoghi di lavoro.

Tra i vari robot presenti sul mercato si possono distinguere due categorie principali di robot

- Robot Industriali
- Robot Collaborativi

Una delle principali differenze tra le due tipologie di robot è che, mentre i robot industriali richiedono la presenza di barriere e protezioni così da separare l'operatore umano dal robot e riuscirne a garantire la sicurezza, quelli collaborativi, anche noti come *co-robot* o *cobot*, sono robot antropomorfi con movimenti su sei assi che vengono progettati per rispettare determinati criteri di sicurezza, flessibilità e compattezza che garantisce loro di lavorare a stretto contatto con l'operatore senza necessitare di barriere di protezione.

In particolare i CoBot sono dotati di una serie di **safety function** che permettono all'operatore di lavorare in piena sicurezza affianco al robot e sono solitamente equipaggiabili con una serie di sensori, ad esempio laser e scanner, che permettono di andare a registrare la presenza dell'operatore controllando così movimenti e tempo di reazione del cobot aumentandone la sicurezza.

La tesi è stata svolta presso l'azienda Tech.Pa SpA e lo scopo è quello di andare a programmare un robot collaborativo, in particolare il robot *Fanuc CRX-25iA*, per essere impiegato come un pallettizzatore senza la necessità di andare ad installare barriere di protezione.

Nell'elaborato si andrà dapprima a parlare dei robot concentrandosi principalmente sui manipolatori collaborativi e sulla loro integrazione nell'industria.

Successivamente si andrà a parlare dei PLC concentrandosi sulla loro storia, sul loro funzionamento e si analizzeranno alcune opzioni presenti in commercio.

Si passerà poi al quarto capitolo, dove andremo ad analizzare più nello specifico il manipolatore impiegato e l'utensile utilizzato.

Infine analizzeremo il software sviluppato sia lato robot che PLC.

ROBOTICA E INDUSTRIA

I robot sono macchine che hanno suscitato interesse generale nel corso della storia e possono essere descritte come macchine o dispositivi in grado di operare autonomamente o controllate da remoto.

Uno dei settori dove trovano principale applicazione è quello industriale, dove hanno il compito di svolgere quelle mansioni che sarebbero troppo pericolose, troppo ripetitive o troppo inefficienti se svolte dall'operatore umano.

2.1 LA STORIA DELLA ROBOTICA

Sebbene l'essere umano cercò da sempre di creare dei dispositivi in grado di supportare o perfino sostituire i lavoratori, fu solo con il progredire dell'industrializzazione che comparve il primo vero e proprio robot programmabile. Infatti grazie al lavoro di due ingegneri americani, *George Devol* e *Joseph Engelberger*, nel 1954 nacque un braccio meccanico dotato di pinza in grado di manipolare oggetti (Figura 2.1).

Grazie al fatto che poteva essere programmato tramite controllo numerico, *UNIMATE* (questo il nome del robot), divenne molto popolare, soprattutto nell'industria automobilistica, in quanto poteva essere utilizzato in un gran numero di scenari e, se dotato della giusta attrezzatura, era in grado di saldare, verniciare e impilare parti metalliche.



Figura 2.1: Robot UNIMATE

Successivamente i robot iniziarono ad essere apprezzati non solo nell'ambito automobilistico, iniziando così ad essere inseriti in maniera massiccia in tutti i processi produttivi.

Così cominciò la diffusione dei robot nelle aziende e iniziarono a entrare nell'industria della robotica altre aziende come *KUKA*, che nel 1973 sviluppò il primo robot a sei assi, *FANUC*, *Yaskawa*, *ASEA* che venne successivamente acquistata dalla *ABB* e *COMAU*.

Da allora l'evoluzione dei robot è avvenuta in maniera molto rapida. Mentre le prime macchine erano in grado di compiere semplici operazioni ripetitive senza avere possibilità di adattamento alle variazioni dell'ambiente circostante, ora esistono macchine che possono compiere lavori molto complessi e che, grazie alla presenza dei sensori, sono a conoscenza dell'ambiente esterno e quindi in grado di adattarsi ai cambiamenti delle condizioni operative, come una variazione nella posizione dell'oggetto target.

L'evoluzione dei robot ha portato alla comparsa dei *CoBot* che hanno subito avuto un grande successo in quanto in grado di condividere lo spazio di lavoro con l'operatore umano e di supportarlo nel suo lavoro senza mettere a repentaglio la sua incolumità.

2.2 DEFINIZIONE DI ROBOT

Possiamo trovare diverse definizioni di robot che, seppur diverse, ne racchiudono alcune peculiarità che possono aiutare a definirlo.

ROBOTIC INSTITUTE OF AMERICA (RIA): Un robot è un manipolatore multifunzionale e riprogrammabile, progettato per muovere materiali, parti, attrezzi o dispositivi specialistici attraverso movimenti programmati variabili, per l'esecuzione di una varietà di compiti. Un robot acquisisce inoltre informazioni dall'ambiente e si muove in modo intelligente di conseguenza

SOCIETÀ ITALIANA DI ROBOTICA INDUSTRIALE (SIRI): Il robot è un manipolatore programmabile multiscopo per la movimentazione di materiali, di attrezzi e altri mezzi di produzione, capace di interagire con l'ambiente nel quale si svolge il ciclo tecnologico di trasformazione relativo all'attività produttiva

FONDATORE DEL ROBOTICS RESEARCH GROUP: La robotica è la connessione intelligente tra percezione ed azione

Da queste definizioni possiamo dedurre come un robot è qualcosa che reagisce in modo intelligente a una situazione ambientale che rileva attraverso un sistema di sensori e questa sua reazione serve a raggiungere un determinato scopo.

2.2.1 *Struttura dei robot*

Un sistema robotico consiste in un insieme complesso rappresentato da più sottosistemi:

SISTEMA MECCANICO composto da tutti quegli organi di locomozione e di manipolazione che consentono al robot di muoversi e intervenire sugli oggetti nell'ambiente circostante

SISTEMA DI ATTUAZIONE, ovvero quel sistema che permette al robot di eseguire una determinata azione

SISTEMA SENSORIALE a cui è affidato il compito di acquisire informazioni sia sullo stato interno del robot, che esterno dell'ambiente

SISTEMA DI GOVERNO che permette di connettere in maniera intelligente azione e percezione che nella robotica industriale risiede solitamente nell'unità di controllo

La struttura meccanica di un robot è anche detta manipolatore ed è costituita da un insieme di corpi rigidi interconnessi tra di loro tramite dei giunti. Possiamo individuare tre componenti essenziali

STRUTTURA PORTANTE, che conferisce mobilità al sistema e gli permette di muoversi nello spazio

POLSO, che conferisce destrezza al robot

ORGANO TERMINALE, che ha il compito di eseguire il compito per cui il robot viene impiegato

2.2.1.1 *Struttura portante*

La struttura portante può essere di due tipi:

CATENA CINEMATICA APERTA Quando è presente una sola sequenza di bracci che connette i due estremi della catena (Figura 2.2a)

CATENA CINEMATICA CHIUSA Quando la sequenza di bracci forma un anello (Figura 2.2b)

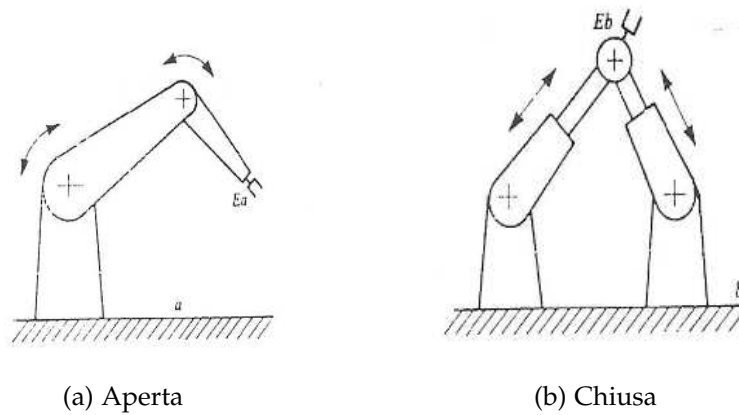


Figura 2.2: Catena cinematica

A seconda della catena cinematica presente nel robot è possibile andare a distinguere l'architettura del manipolatore. Possiamo individuare tre tipi di architetture

SERIALE costituita da un'unica catena cinematica aperta (per esempio i robot antropomorfi)

PARALLELA costituita da molteplici catene cinematiche aperte che ne vanno a costituire una chiusa in quanto collegate ad un'unico membro

MISTA che è un'unione delle due sopra descritte

I giunti servono a garantire la mobilità del manipolatore e, a seconda del moto da realizzare, possono essere di due tipi

GIUNTO PRISMATICO (P) che conferiscono il moto relativo di traslazione

GIUNTO ROTOIDALE (R) Che conferiscono moto relativo di rotazione

Il numero di giunti presenti in un manipolatore dipende dalla sua catena cinematica, infatti mentre con una catena cinematica aperta ogni giunto garantirà un grado di libertà, con quella chiusa dovranno essere in numero maggiore rispetto ai gradi di libertà effettivi del robot. I gradi di libertà dovranno poi essere distribuiti in maniera appropriata in modo tale da essere in numero sufficiente per garantire l'esecuzione di un dato compito.

A seconda del tipo e della successione dei gradi di libertà si possono individuare cinque diversi tipi di manipolatori

MANIPOLATORE CARTESIANO: composto da soli accoppiamenti prismatici

MANIPOLATORE CILINDRICO: come il manipolatore cartesiano, solo che il giunto alla base diventa un giunto rotoidale modificando quindi il suo movimento nello spazio

MANIPOLATORE SFERICO: composto da due giunti rotoidali e uno prismatico. Il suo spazio di lavoro diventa una sfera cava

MANIPOLATORE SCARA: possiamo trovare due configurazioni, *SCARA classico* composto da un giunto prismatico alla base e due giunti rotoidali e *SCARA modificato* dove il giunto prismatico si sposta al polso e viene sostituito alla base da un rotoidale

MANIPOLATORE ANTROPOMORFO: composto da soli accoppiamenti rotoidali

2.2.1.2 *Polso*

I restanti gradi di libertà sono fortini dal polso, che può essere a uno o tre gradi di libertà. Tra quelli a tre gradi di libertà possiamo individuare due categorie principali

POLSO SFERICO: caratterizzato dal fatto che i tre assi di rotazione di intersecano in un punto. Questa forma è quella che garantisce maggiore destrezza. Fondamentale di questa tipologia è il disaccoppiamento che si ha tra posizione e orientamento dell'organo terminale, infatti il posizionamento del punto degli assi di polso è affidato alla struttura portante, mentre l'orientamento dell'organo terminale è affidato al polso stesso

POLSO NON SFERICO: rispetto a quello sferico è meno complesso dal punto di vista meccanico a discapito, però, di una maggiore complessità per quanto riguarda il posizionamento e orientamento in quanto interagiscono tra di loro complicando il coordinamento della struttura portante e del polso stesso per la realizzazione del moto assegnato.

2.2.2 *I robot collaborativi, CoBot*

I CoBot sono manipolatori in grado di lavorare a stretto contatto con l'operatore umano senza necessitare di barriere o protezioni che li separi, è quindi in grado di condividere lo spazio di lavoro con il lavoratore senza essere un rischio per la sua incolumità. Grazie a questa loro caratteristica hanno avuto un grande successo in ambito industriale e ha aiutato a far sì che si diffondessero in settori come l'industria automobilistica, chimica, elettronica e delle materie plastiche.

2.2.2.1 Storia

I CoBot vennero inventati nel 1996 da J. Edward Colgate e Michael Peshkin a seguito di un'iniziativa del 1994 della General Motors e da una ricerca sovvenzionata dalla stessa nel 1995, finalizzata a trovare un modo per rendere i robot o le attrezzature simili ai robot sicure al punto da poter collaborare con le persone.

I primi CoBot erano sicuri per l'essere umano in quanto non avevano generazione interna di forza motrice ma era l'operatore umano stesso a fornirla, lasciando al CoBot il compito di re-indirizzare o guidare il carico aiutando il lavoratore. Negli anni a seguire anche i CoBot sono stati in grado di generare forza motrice autonomamente consentendo loro di muoversi autonomamente nello spazio.

Nel 2004 l'azienda tedesca KUKA lancia sul mercato il suo primo CoBot, *LBR3*, un robot leggero controllato da un computer risultato di una lunga collaborazione con l'istituto Aerospaziale Tedesco.

Decisivo però per la diffusione dei robot collaborativi nel mercato dell'automazione è stato il CoBot *UR5* sviluppato da Universal Robots uscito nel 2008.

A seguire altre aziende cominciarono a produrre CoBot, tra le quali possiamo trovare nomi come ABB e FANUC.

2.2.2.2 Differenza tra CoBot e Robot Industriali

La principale differenza tra i due manipolatori è che il robot industriale tradizionale viene impiegato per lavorare su alti volumi e di solito viene mantenuto in una posizione fissa, mentre il robot collaborativo è compatto e occupa poco spazio il che permette di collocarlo facilmente in luoghi diversi. Altra differenza è dovuta al fatto che i CoBot hanno capacità di carico inferiori rispetto i robot industriali e che sono progettati per lavorare e interagire con le persone.

Questa loro caratteristica è dovuta all'implementazione delle cosiddette *Safety Native*, ovvero delle caratteristiche di sicurezza, sia hardware che software, integrate con lo specifico scopo di permettere al manipolatore di lavorare senza dover fare affidamento su barriere fisiche o dispositivi di sicurezza esterni.

Alcune delle Safety Native più comuni sono

SENSORE DI FORZA E COPPIA: che consentono loro di rilevare contatti con gli esseri umani o con oggetti e reagire adeguatamente per evitare danni

SISTEMI DI VISIONE AVANZATA: che consente loro di percepire e riconoscere l'ambiente circostante, contribuendo a

evitare collisioni e garantire una collaborazione sicura con gli operatori umani

CONTROLLO DI VELOCITÀ E FORZA: che permette loro di regolare dinamicamente la velocità e forza in risposta alle condizioni circostanti

Altra caratteristica molto importante di questi manipolatori è rappresentata dal loro modo di programmarli in modo semplice e intuitivo così che la maggior parte degli operatori che lavorano con loro possano dare le relative indicazioni.

2.2.2.3 *Caratteristiche e ambiti applicativi*

I CoBot sono robot antropomorfi progettati per imitare le capacità umane. Spesso si configurano come uno o due bracci robotici con almeno sei gradi di libertà permessi dalla presenza di giunti, principalmente rotoidali.

Un braccio CoBot è costituito da una base, fissa e ancorata a una piastra-base che contiene i connettori.

I connettori sono poi collegati, tramite appositi fili, a una *unità di controllo* che monitora e gestisce il movimento degli assi attraverso appositi motori e un encoder ad alta risoluzione.

I CoBot sono regolati dalla specifica tecnica **ISO/TS 15006**, ovvero delle norme che forniscono delle linee guida per la progettazione, le misure protettive e le informazioni di utilizzo dei robot.

Queste norme stabiliscono che debbano essere presenti i seguenti sistemi

STOP DI SICUREZZA MONITORATO, ovvero un fermo-robot alla presenza dell'operatore

GUIDA MANUALE, che deve avere in dotazione stop di emergenza, dispositivo di attivazione, riduzione di velocità e movimento

MONITORAGGIO POSIZIONE E VELOCITÀ, che mantenga una distanza predefinita con l'operatore e che effettui una eventuale riduzione della velocità

LIMITAZIONE POTENZA E VELOCITÀ sia in fase di progettazione che attraverso un sistema di controllo che valuta la soglia di pressione sui corpi esposti al pericolo di contatto

I CoBot hanno capacità di carico che variano solitamente tra i 0.5kg e i 110kg, pesano tra i 2kg e i 1000kg e hanno un'estensione del braccio che va dai 50cm ai 3m.

Quindi sono generalmente leggeri, compatti e flessibili in quanto sono riprogrammabili facilmente.

Per queste caratteristiche si dimostrano particolarmente adatti per le piccole-medie produzioni soggette a frequenti variazioni nella lunghezza del lotto e del mix.

I CoBot vengono impiegati principalmente nella produzione manifatturiera, nella logistica e nel medicale.

Nell'industria manifatturiera questi manipolatori permettono di automatizzare le operazioni di assemblaggio di precisione (come avvitatura, saldatura, inserimento/messa in posizione di parti), verniciatura, lucidatura, incollaggio e dosatura. Viene anche impiegato per le operazioni di *Machine Tending*, ovvero tutte quelle operazioni di asservimento macchine e per le operazioni di picking e confezionamento, ad esempio il funzionamento come pallettizzatore.

I CoBot vengono quindi impiegati in quelle applicazioni meccaniche e ripetitive, potenziandone le capacità, recuperando efficienza nelle prestazioni e riducendo i rischi relativi alla salute.

2.2.3 *Analisi CoBot presenti in commercio*

Sul mercato oggi ci sono numerosi robot collaborativi provenienti da diverse case costruttrici, ognuno con le sue caratteristiche e specializzato in determinati campi applicativi. Lo scopo di questa sezione è quello di andare a vedere alcuni dei CoBot presenti in commercio e andare ad analizzarne le caratteristiche.

2.2.3.1 *ABB*

Tra i robot presenti in casa ABB possiamo trovare

- Dual-arm YuMi IRB 14000
- Single-arm YuMi IRB 14050
- GoFa CRB 15000

Il primo robot collaborativo con marchio ABB è lo **Dual-arm YuMi IRB** (Figura 2.3a) un CoBot che, come indicato dal nome, possiede due bracci robotici indipendenti.

Questo manipolatore è stato progettato per andare incontro a processi produttivi che necessitano di flessibilità e agilità per l'assemblaggio di piccoli componenti. E' costituito da uno scheletro leggero e rigido in magnesio coperto da una plastica morbida e imbottita che aiuta ad assorbire gli urti accidentali e inaspettati. Questi aspetti, insieme al suo design arrotondato, aiutano ad aumentare la sicurezza per gli operatori umani.

Con un payload massimo di $500g$, un'estensione massima di $559mm$ e una ripetibilità di $0.02mm$ presenta un'ottima soluzione per quelle situazioni in cui lo spazio di lavoro non è troppo grande ed è richiesta un'alta precisione per la movimentazione di piccoli componenti non troppo pesanti.

Versione molto simile al Dual-arm è lo **Single-arm YuMi** (Figura 2.3b), dove l'unica differenza è il fatto che è composto da un solo braccio robotico invece che due.

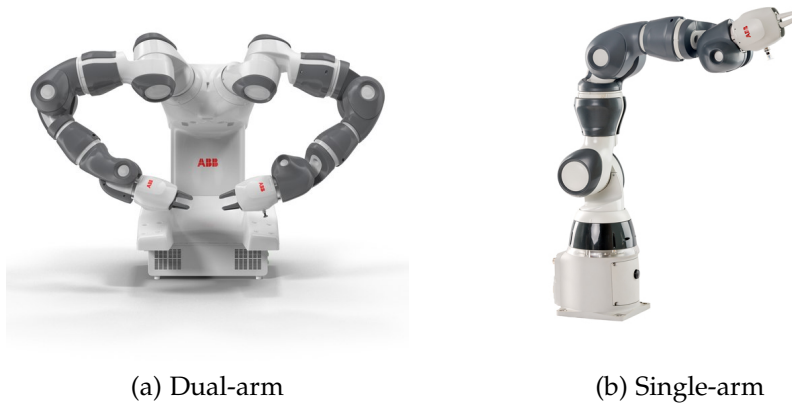


Figura 2.3: YuMi

Un altro robot collaborativo che possiamo trovare è il **GoFa CRB 15000** (Figura 2.4).

Ultimo CoBot prodotto da ABB, il GoFa è stato concepito per la collaborazione stretta e sicura con l'uomo. E' stato progettato per la movimentazione di carichi fino a $5kg$ e integra sensori di coppia in ognuno dei suoi 6 giunti. Consente inoltre una velocità massima pari a $2.2m/s$ e uno sbraccio di $0.95m$.



Figura 2.4: GoFa CRB 15000

2.2.3.2 *Universal Robots*

Universal Robots è un'azienda di origine Danese specializzata nella produzione di robot collaborativi. I CoBot UR sono utilizzati sia nelle piccole e medie imprese che nelle grandi società in settori diversi tra i quali quello automobilistico, elettronico, metalmeccanico, farmaceutico e manifatturiero.

I prodotti di Universal Robots consistono nei CoBot *UR3e*, *UR5e*, *UR10e*, *UR16e*, *UR20*, *UR30*, classificati in base alla loro capacità di carico.

Andandone ad esaminare alcuni abbiamo

UR3e (Figura 2.5a) è il robot più piccolo della gamma, è ultraleggero e compatto ideale per applicazioni da tavolo. Questo CoBot ha un peso molto ridotto pari a 11kg , ma riesce ad offrire comunque un payload massimo di 3kg . Ogni giunto di cui è composto è in grado di effettuare una rotazione di $\pm 360^\circ$, mentre il polso presenta rotazione infinita il che lo rende particolarmente adatto per applicazioni di assemblaggio e avvitatura.

UR10e (Figura 2.5b) robot collaborativo molto versatile che offre un payload di 12.5kg e un raggio di azione di 1300mm . E' quindi in grado di coprire grandi spazi di lavoro, senza però andare a perdere precisione o ridurre le performance di carico. Viste le sue caratteristiche risulta adatto a ricoprire un'ampia gamma di operazioni, quali asservimento macchine, pallettizzazione e packaging.

UR30e (Figura 2.5c) può sollevare carichi pesanti mantenendo un ingombro ridotto in un ambiente collaborativo. Ha una capacità di sollevamento pari a 30kg e un raggio di 1300mm che gli permette di gestire macchine più grandi, pallettizzare prodotti pesanti e supportare efficacemente l'avvitatura a coppia elevata.

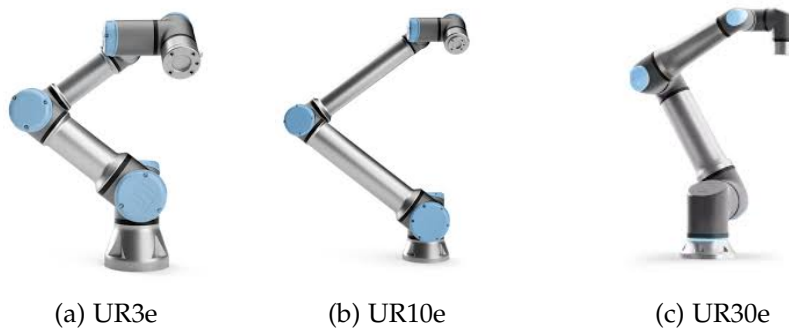


Figura 2.5: Universal Robots

2.2.3.3 COMAU

Con ben 45 anni di esperienza in robotica e automazione in vari settori, Comau oggi propone anche soluzioni collaborative e collabora con molte attività industriali. Collaborando con le aziende sviluppa soluzioni collaborative che siano in grado di soddisfare le diverse esigenze produttive.

Le due principali soluzioni che si possono trovare sono

AURA (Advance Use Robotic Arm) (Figura 2.6a) ricoperto da una speciale pelle protettiva e dotato di sensori di contatto e prossimità lo rendono in grado di lavorare a stretto contatto con l'uomo in totale sicurezza. Può inoltre fare affidamento su un sistema di laser scanner.

Oltre ai sistemi di sensoristica è anche dotato di un sistema di monitoraggio della velocità e della distanza minima con persone/oggetti, in maniera tale che il manipolatore possa cominciare a muoversi solo qualora la distanza minima sia rispettata, e di una limitazione della potenza e della forza derivante da specifiche progettuali o da controlli intrinseci che consentono al robot, anche in caso di impatto, di limitare potenza e forza a livelli non dannosi. E' inoltre progettato per essere facile da programmare e supporta la guida manuale. Grazie a questa caratteristica lo si può programmare insegnandogli i movimenti da eseguire muovendolo a mano, oppure spostarlo in una determinata posizione semplicemente toccandolo.

Supporta due modalità di lavoro, *modalità collaborativa* e *modalità non collaborativa*. La prima, contraddistinta da una luce led accesa, va a limitare la velocità lineare massima a 500mm/s , mentre nella modalità non collaborativa, contraddistinta dal led spento e attivabile solo che è presente un sistema di laser scanner, questa viene alzata fino a 2000mm/s . Il led viene inoltre utilizzato anche per anda-

re ad identificare lo stato del robot, ovvero a seconda del colore del led sarà possibile capire se il robot si trova in *manual restart* in quanto ha percepito una forza molto elevata e si è bloccato, *automatic restart*, nel caso percepisca una leggera forza, *stato di allarme* dove il robot si ferma in quanto ci sono degli stati di allarme attivo.

Il robot ha inoltre tre modalità di funzionamento, *Program Mode*, *Automatic Mode* e *Remote mode*. Nella prima modalità l'utente può programmare il movimento del robot manualmente, nella seconda modalità il robot esegue automaticamente il programma senza la necessità di intervento diretto da parte dell'operatore, la terza modalità invece consente al robot di essere controllato da remoto attraverso, per esempio, una connessione di rete.

Ulteriore caratteristica è che questo manipolatore riesce ad unire le elevate capacità di carico dei robot industriali con le caratteristiche dei robot collaborativi, in quanto riesce ad offrire uno dei payload maggiori tra i CoBot in commercio (170kg) per una estensione di 2790mm. Questa capacità di sollevamento elevata lo rende perfetto per lavorare in settori dove la capacità di carico richiesta è elevata, avendo però una ripetibilità di $\pm 0.1mm$ può essere applicato anche in applicazioni dove è richiesta precisione.

RACER-5-0.80 COBOT (Figura 2.6b) ultimo CoBot di casa COMAU, è stato progettato per essere in grado di passare automaticamente da velocità di un robot industriale alla velocità collaborativa quando sente la presenza di un operatore umano nella sua area di lavoro. In modalità non collaborativa può infatti raggiungere i 6m/s che vengono invece limitati a 500mm/s in modalità collaborativa.

Grazie alla sua flessibilità, ripetibilità (di $\pm 0.03mm$) e precisione ad alte prestazioni permette di ridurre i tempi ciclo garantendo una migliore produttività.

Offre inoltre un payload di 5kg per uno sbraccio di 809mm, il che lo rende adatto a lavorare in quelle situazioni dove lo spazio di lavoro disponibile è limitato ma è richiesta una elevata ripetibilità e il payload non è troppo elevato.

Anche per questo manipolatore sono presenti dei led che aiutano ad identificare velocemente se si trova in modalità collaborativa o meno.

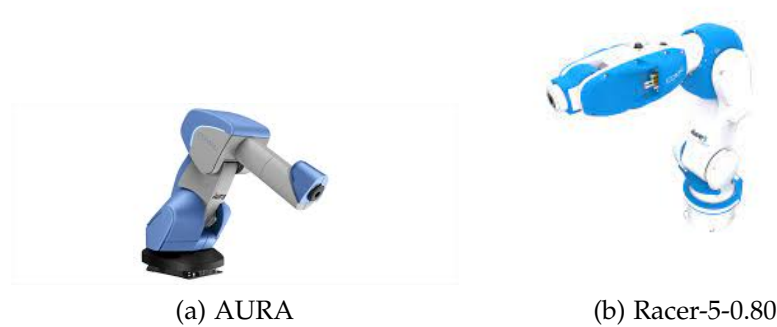


Figura 2.6: COMAU Robots

2.2.3.4 FANUC

Forte di una grande esperienza in ambito della robotica e dell'automazione industriale, Fanuc si lancia nel mondo dei CoBot con una prima gamma di robot i robot collaborativi CR. La categoria CR era composta dai seguenti modelli: CR-4iA, CR-7iA, CR-7iA/L, CR-14iA/L, CR-15iA, CR-35iB ed erano tutti caratterizzati da una copertura in gomma.

Questi modelli sono poi stati sostituiti da una categoria più aggiornata, la gamma CRX. In questa categoria possiamo trovare il

CRX-5iA con un carico utile pari a 5kg, uno sbraccio di 994mm e un peso di 25kg questo CoBot si colloca bene in un'ampia gamma di applicazioni, come l'assemblaggio, l'asserimento macchina o la movimentazione di materiali.

CRX-10iA avente carico utile di 10kg, uno sbraccio di 1249mm e un peso di 40kg

CRX-10iA/L avente carico utile pari a 10kg, uno sbraccio di 1418mm e un peso di 39kg

CRX-20iA/L con un carico utile pari a 20kg, uno sbraccio di 1418mm e un peso di 41kg questo CoBot si colloca bene in un'ampia gamma di applicazioni, come carico e scarico delle macchine, movimentazione dei materiali o pallettizzazione.

CRX-25iA con un carico utile pari a 25 – 30kg, uno sbraccio di 1756 – 1889mm e un peso di 135kg questo CoBot può gestire un'ampia gamma di applicazioni, come gestione delle macchine, movimentazione dei materiali, pallettizzazione e imballaggio.

Le caratteristiche principali della gamma CRX sono

FUNZIONE DI ARRESTO DA CONTATTO grazie alla presenza di sensori ad alta sensibilità che attivano un arresto in sicurezza immediato in caso di contatto con un corpo umano.

FACILE DA CONNETTERE dispone di semplici plugin per poter installare le interface di utilizzo delle periferiche quali pinze, telecamere, ecc.

PROGRAMMAZIONE TRAMITE MOVIMENTAZIONE MANUALE grazie alla funzione di programmazione manuale è possibile insegnare una posizione al robot spostandolo manualmente in quella posizione e poi salvarla



Figura 2.7: Modello CRX-5iA

Nell'esperienza descritta in questo elaborato si è utilizzato il *CRX-25iA* e lo si è programmato per funzionare come pallettizzatore.

I **PLC**, acronimo di *Programmable Logic Controller*, sono dispositivi progettati per monitorare e regolare macchine e processi di produzione in ambienti industriali che utilizzano tecnologie di automazione. Questi possono operare come sistemi autonomi per ottimizzare i processi in modo intelligente e indipendente.

La sua funzione è quella di andare ad eseguire ed elaborare i segnali digitali e analogici provenienti da sensori e diretti agli attuatori presenti in un impianto industriale. Ci permette quindi di far comunicare tra loro le varie periferiche e ci consente di comandarle e controllarle nella maniera più appropriata. La sua struttura è adattata a in base al processo da automatizzare.

Il PLC è quindi, essenzialmente, un sistema elettronico a funzionamento digitale dotato di una memoria programmabile. Questa memoria viene utilizzata per salvare le istruzioni di controllo che consentono al dispositivo di eseguire determinate funzioni e compiti. Attraverso moduli di uscita digitali e analogici vengono quindi controllati vari tipi di macchine e processi.



Figura 3.1: PLC Allen Bradley

3.1 STORIA DEL PLC

Prima dell'avvento dei PLC la logica di funzionamento nei macchinari elettromeccanici era gestita da relè. Questi apparecchi presentavano, tuttavia, l'inconveniente per cui dopo molte ore di lavoro si rompevano in quanto si tratta di oggetti meccanici continuamente sollecitati. Ulteriore inconveniente nell'utilizzo di questi dispositivi era la quantità di cablaggi necessari per il loro funzionamento, il che rendeva estremamente comples-

sa la ricerca e risoluzione di eventuali guasti o la modifica dei circuiti.

Si cominciò quindi a cercare un modo per rendere più semplice ed efficiente il lavoro, cercando di spostarsi verso computer e semiconduttori a stato solido. Fu così che, nel 1968, l'ingegnere *Dick Morley* scrisse le prime specifiche per un apparecchio PLC.

Questo apparecchio doveva permettere un processo continuo e quindi stopparsi solo se voluto da programma. Decise inoltre di che l'accesso alla mappa della memoria del controllore dovesse essere diretto in maniera tale da permettere un modo veloce di programmare tabelle dati e informazioni nella cpu. Doveva infine avere un linguaggio di programmazione proprietario e in particolare inventò la *logica ladder*.

Fu così che nel 1969 i primi PLC furono acquistati dalla *General Motors*, che era alla ricerca di un modo per ammodernare le proprie fabbriche, seguiti successivamente dalla *General Electric*.

3.2 FUNZIONAMENTO DEI PLC

I PLC sono dispositivi che possono essere collegati alla rete e programmati per rispondere a determinati input con determinati output. Possono essere programmati digitalmente e possono svolgere compiti di controllo anche molto complessi, per questo sono molto utilizzati nelle industrie 4.0.

3.2.1 Programmazione

Esistono diversi modelli di PLC che possono supportare diversi tipi di programmazione. Le opzioni di programmazione più comuni sono

- ST: Structured Text
- FBD: Function Block Diagram
- LAD: Logical Ladder
- STL: Statement List
- SCL: Structured Control Language

Tra queste opzioni la *Logica Ladder* (Figura 3.2) è di sicuro la più utilizzata in quanto, essendo basata su immagini, permette una più facile comprensione di quello che sta succedendo nel PLC. È un linguaggio di programmazione molto semplice che si avvale di simboli grafici che vengono combinati assieme all'interno di due linee verticali, rappresentanti l'alimentazione. Questi simboli vengono inseriti in delle righe orizzontali

che congiungono queste verticali e che vengono chiamate *rung*. Ogni rung è idealmente diviso in due parti, la parte sinistra, chiamata zona di test, contiene le variabili di ingresso o variabili interne, la parte di destra, detta zona di azione, che comprende le uscite esterne o interne, nonché i blocchi di funzione avanzata. L'energia può fluire solo da sinistra verso destra.

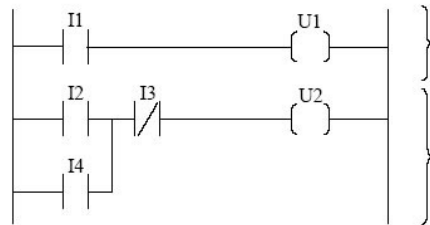


Figura 3.2: Esempio di Logica Ladder

Lo *Structured Text* (Figura 3.3) viene invece usato in quelle situazioni dove le operazioni da eseguire sono complesse e quindi risulterebbe impossibile scrivere il programma con la sola logica ladder. Questo è un linguaggio di programmazione molto simile ai linguaggi di programmazione classici. Questo linguaggio ci permette di andare a scrivere le istruzioni in un modo più simile al nostro modo di pensare e parlare. Seppur risulti meno immediata la comprensione di quello che sta succedendo nel PLC, il testo strutturato ci permette di andare ad usare strumenti molto potenti come i *cicli FOR* che ci permette di eseguire delle istruzioni un numero di volte da noi deciso.

```
// Structured Text Program
ST_ON_TIMER.PRE := 10000;
ST_OFF_TIMER.PRE := 10000;

IF ST_ON_TIMER.TI THEN ST_OUTPUT := 1;
ELSE ST_OUTPUT := 0;
END_IF;

IF ST_TRIGGER THEN ST_ON_TIMER.TimerEnable := 1;
ELSE ST_ON_TIMER.TimerEnable := 0;
END_IF;

TONR(ST_ON_TIMER);
TONR(ST_OFF_TIMER);

ST_OFF_TIMER.TimerEnable := ST_ON_TIMER.DN;
ST_ON_TIMER.Reset := ST_OFF_TIMER.DN;
```

Figura 3.3: Esempio di Testo Strutturato

3.2.2 Software

Il programma, una volta scritto su PC tramite il programma applicativo, viene caricato sul controllore tramite l'interfaccia e va così a definire le relazioni tra i dispositivi di ingresso e di uscita.

Il funzionamento vero e proprio del PLC è quindi indipendente dal PC, che invece funge come mera interfaccia.

Il controllore può includere diverse funzioni specifiche, tra le quali

- Controllo del flusso
- Controllo dei collegamenti
- Funzioni basate su dati numerici

Il software dei PLC possono essere classificati secondo diverse categorie

PLC A FUNZIONAMENTO CICLICO: questi controllori seguono il principio di base dell'elaborazione dei dati con input, elaborazione e output. Gli input vengono chiamati con una query e il controllo viene trasferito al programma utente. Dopo che i segnali di controllo sono stati trasferiti alle uscite, il processo riparte dall'inizio.

PLC CICLICI CON INTERRUPT: questi PLC sono programmati per eseguire un loop di programmazione aggiuntivo che, in caso di cambiamento di stato del sensore collegato, ne adatta il funzionamento a una situazione diversa. Una volta completato il ciclo aggiuntivo, il programma principale continuerà la normale esecuzione.

PLC CONTROLLATI DA EVENTI: questi controllori sono progettati per elaborare specifiche attività pre-programmate a seguito di un cambiamento di stato del sensore o dei sensori collegati.

Sarà quindi compito del programmatore e progettista andare ad individuare quale PLC risulti essere più adatto per l'applicazione sotto esame.

3.3 PLC IN COMMERCIO

Andando ad analizzare il mercato dei PLC possiamo notare come siano tre i principali produttori

- Siemens
- Rockwell Automation (sotto nome Allen Bradley)
- Mitsubishi Electric

Queste aziende da sole si spartiscono la maggiore fetta del volume globale, infatti possiamo trovare i loro prodotti in qualsiasi soluzione di automazione di ogni dimensione e per ogni tipo di industria.

3.3.0.1 Siemens

Siemens è un'azienda di origine tedesca fondata nel 1847, con sede a Monaco di Baviera, ed è una delle maggiori fornitrici di PLC a livello globale.

Tra i prodotti che offre possiamo distinguere quattro prodotti principali: il SIMATIC S7-1200, adatto per le piccole applicazioni, il SIMATIC S7-1500, S7-400 e S7-300, adatti per le grandi applicazioni. Gli ultimi due sono versioni più vecchie rispetto al S7-1500, ma sono ancora ampiamente utilizzate.

Ognuna delle famiglie di cui sopra viene offerta con diverse opzioni. Per quanto riguarda le unità centrali si possono scegliere le dimensioni della memoria e quali porte di comunicazione avere a bordo, oltre ad avere moduli per le applicazioni di sicurezza.

SIMATIC s7-1200 Controller programmabile utilizzato per automatizzare processi industriali e controllare macchinari (Figura 3.4a).

È caratterizzato da un'*architettura modulare*, ovvero permette l'aggiunta e la rimozione di moduli input/output in base alle esigenze dell'applicazione.

È un controllore compatto che è comunque in grado di offrire una notevole potenza di calcolo per eseguire i vari compiti di controllo, che siano questi semplici o complessi. È infatti dotato di una CPU che esegue il software di automazione e gestisce il controllo delle periferiche.

Dispone di diverse *porte di comunicazione* che consentono il collegamento con altri dispositivi o sistemi.

È progettato con *funzionalità di sicurezza integrate* per garantire un funzionamento affidabile e sicuro. Tra queste possiamo trovare: protezioni contro sovratensioni, surriscaldamento e cortocircuiti.

SIMATIC s7-1500 Controllore programmabile avanzato progettato per fornire elevate prestazioni, affidabilità e flessibilità nelle applicazioni industriali (Figura 3.4b).

È caratterizzato da una *potenza di calcolo avanzata* in quanto dotato di una CPU potente che offre elevate capacità di elaborazione.

Ha un'*architettura modulare e scalabile* in quanto permette l'aggiunta e la rimozione di moduli input/output in base alle esigenze dell'applicazione, oltre al fatto che possiede una piattaforma scalabile, fornendo quindi agli utilizzatori la possibilità di espandere e aggiornare il sistema secondo necessità.

Possiede *connessioni di rete avanzate*, tra cui connessione Ethernet, PROFINET, PROFIBUS, Modbus, TCP e altre

E' progettato con *funzionalità di sicurezza integrata* per garantire un funzionamento sicuro delle applicazioni industriali, oltre a supportare l'integrazione di *moduli di sicurezza dedicati* per tutte quelle applicazioni che richiedono funzionalità di sicurezza avanzate.

SIMATIC s7-300 Controllore usato per l'automazione industriale (Figura 3.4c) caratterizzato da un'elevata *versatilità e compattezza* che lo rende ideale per applicazioni industriali di medie dimensioni.

Possiede un'*architettura modulare* che consente l'aggiunta e la rimozione di moduli I/O in base alle esigenze specifiche dell'applicazione, oltre a permettere una *facile espansione* dei moduli presenti nel sistema.

SIMATIC s7-400 Controllore usato per l'automazione industriale (Figura 3.4d) caratterizzato da *elevate prestazioni* di elaborazione, il che lo rende adatto per quelle applicazioni industriali complesse che richiedono tempi di risposta rapidi e una grande quantità di I/O.

Possiede un'*architettura modulare* che consente l'aggiunta e la rimozioni di moduli I/O.

E' un controllore progettato per garantire un'*elevata disponibilità e affidabilità* del sistema in quanto possiede funzionalità come la duplicazione della CPU, la ridondanza dell'alimentazione e dei moduli I/O e la possibilità di "hot swapping" per la sostituzione dei moduli senza la necessità di interrompere il funzionamento del sistema.

Supporta, inoltre, una vasta gamma di funzionalità, tra cui la gestione di reti di comunicazione complesse, il controllo di processi di automazione avanzati e l'integrazione con sistemi di visualizzazione e controllo.



(a) S7-1200

(b) S7-1500

(c) S7-300

(d) S7-400

Figura 3.4: Esempio controllori SIEMENS

3.3.0.2 Rockwell Automation (Allen Bradley)

Azienda fondata nel 1903 da Lynde Bradley e Stanton Allen e acquistata nel 1985 da Rockwell Automation, questa azienda rappresenta la seconda maggiore fornitrice di PLC a livello globale.

Tra le sue proposte possiamo trovare sistemi di controllo di grandi e piccole dimensioni oltre che a quelli micro. Alcuni dei controllori che offrono sono

COMPACTLOGIX 5380 (Figura 3.5a) Controllore progettato per lavorare in applicazioni di automazione industriale che richiedono elevate prestazioni, affidabilità e flessibilità.

Sono caratterizzati da un'*architettura potente* che offre un'elevata velocità di elaborazione, una *connessione di rete avanzata* in grado di supportare diverse opzioni di connettività di rete, *flessibilità di configurazione* che permette all'utilizzatore di aggiungere/rimuovere moduli I/O o altri dispositivi per renderlo più adatto all'applicazione specifica e *sicurezza integrata* che garantisce un funzionamento sicuro delle applicazioni industriali come, per esempio, protezione contro gli accessi non autorizzati, monitoraggio delle prestazioni del sistema e diagnostiche di errore avanzate.

COMPACTLOGIX 5370 (Figura 3.5b) E' un controllore progettato per lavorare in applicazioni di automazione industriale molto simile al CompactLogix 5380 se non per delle differenze dal punto di vista delle caratteristiche di connettività e prestazioni.

Questo controllore offre delle *prestazioni di elaborazione* minori, minori opzioni di *connessioni di rete avanzate*, capacità di integrare *funzioni di sicurezza* meno complesse.

Il CompactLogix 5380 offre quindi delle prestazioni superiori rispetto il CompactLogix 5370, questo però non rende quest'ultimo inutile in quanto il controllore verrà scelto in maniera opportuna a seconda dell'applicazione.

MICRO850 (Figura 3.5c) Controllore progettato per funzionalità di controllo in quelle applicazioni di automazione di piccole e medie dimensioni.

E' caratterizzato da *dimensioni compatte* che lo rendono ideale per l'installazione in spazi ridotti, *elevata flessibilità di I/O* che consente agli utenti di adattare il sistema alle esigenze che l'applicazione richiede, *connessioni di rete* che consentono il collegamento a dispositivi e sistemi di automazione in un ambiente industriale e *funzionalità di sicurezza integrate* che gli consentono un funzionamento sicuro.



(a) CompactLogix 5380



(b) CompactLogix 5370



(c) Micro850

Figura 3.5: Esempio controllori Rockwell Automation

3.3.0.3 Mitsubishi Electric

La Mitsubishi Electric è un'ulteriore grande fornitrice di PLC a livello globale. Anche in questo caso l'azienda offre un'ampia varietà di controllori così da poter trovare quello più adatto all'applicazione specifica, alcuni esempi sono

MELSEC SERIE iQ-R (Figura 3.6a) Famiglia di controllori programmabili che offre funzionalità avanzate per l'automazione industriale e si distingue per la sua potenza di calcolo, flessibilità e affidabilità.

Questa serie offre *elevate prestazioni di calcolo* in quanto dotati di una CPU ad alte prestazioni che offre, quindi, un'elevata velocità di elaborazione.

E' caratterizzata da un'*architettura modulare* che consente di aggiungere e rimuovere facilmente moduli I/O per adattare il controllore alle esigenze dell'applicazione.

Offre un'*ampia gamma di opzioni di comunicazione* tra cui Ethernet, Profibus, Profinet, Modbus TCP/IP e molti altri.

Integra *funzioni di sicurezza* come controllo degli accessi, monitoraggio degli ingressi di sicurezza e possibilità di integrare moduli di sicurezza esterni per applicazioni che richiedono funzionalità avanzate di sicurezza.

MELSEC FAMIGLIA FX (Figura 3.6b) Linea di controllori che offre funzionalità per l'automazione industriale di piccole e medie dimensioni.

Questa serie è caratterizzata da *dimensioni compatte* che li rende ideali per l'installazione in spazi ristretti o applicazioni in cui lo spazio è limitato.

Offre un'*ampia gamma di I/O e flessibilità di espansione* attraverso l'aggiunta di moduli aggiuntivi o altri dispositivi per adattarsi all'applicazione interessata.

MELSEC SYSTEM Q (Figura 3.6c) Controllori progettato per offrire un controllo preciso e una gestione efficiente dei processi.

Questo PLC offre un'*architettura modulare* per poterlo adattare all'applicazione specifica, *elevate prestazioni di elaborazione* per gestire quelle applicazioni che richiedono tempi di risposta rapidi e un'elevata capacità di elaborazione dati.

Consente inoltre un'*ampia gamma di opzioni di comunicazione, funzionalità di sicurezza avanzate* anche implementabili qualora fosse richiesto in applicazioni che richiedono elevati livelli di sicurezza.



(a) Serie iQ-R



(b) Famiglia FX



(c) System Q

Figura 3.6: Esempio controllori MELSEC

FANUC CRX-25iA E GRIPPER A VUOTO

4.1 DESCRIZIONE

Tra i robot collaborativi presenti sul mercato oggi giorno possiamo trovare il CRX-25iA (Figura 4.1), CoBot sviluppato dalla FANUC.

Questo manipolatore appartiene alla gamma dei CRX che, insieme a quella dei CR, compone la gamma di robot collaborativi più ampia e più completa presente sul mercato.

Il CRX-25iA è un robot antropomorfo a sei assi, progettato per una vasta gamma di applicazioni industriali, il cui design è stato ottimizzato per garantire prestazioni elevate, precisione e affidabilità. La struttura principale è realizzata in leghe leggere ad alta resistenza che gli conferiscono una buona resistenza mantenendo il contenuto il peso. Questa sua caratteristica gli permette di muoversi con agilità e rapidità.

I giunti e le articolazioni sono costruiti con materiali resistenti all'usura e agli agenti ambientali, conferendogli una lunga durata operativa. Questo manipolatore è infatti dotato di protezione IP67 contro polveri e schizzi d'acqua, il che consente anche di lavorare in quegli ambienti più difficili della fabbrica e in condizioni gravose.

Il braccio è composto da sei giunti motorizzati che gli conferiscono sei gradi di libertà per il movimento, proprio per questo il manipolatore è in grado di eseguire una vasta gamma di operazioni, comprese le azioni di presa, sollevamento, posizionamento e assemblaggio.

Il CRX-25iA è equipaggiato con sensori che monitorano costantemente la sua posizione, velocità e carico che consentono al robot di adattarsi dinamicamente alle variazioni dell'ambiente di lavoro e, così facendo, di evitare le collisioni con ostacoli o operatori umani. Qualora avvenisse invece un contatto accidentale con un corpo esterno, grazie a questi sensori il robot sarebbe in grado di percepirlo e quindi di eseguire immediatamente un arresto in sicurezza.

La parte terminale del braccio può essere configurata con una grande varietà di strumenti, come pinze, ventose e dispositivi di saldatura a seconda dell'esigenza dell'applicazione. L'applicazione di questi dispositivi è facilitata dalla presenza di porte di connessione sulla parte terminale.

Possiede un sistema di controllo altamente sofisticato che utilizza algoritmi avanzati per pianificare e eseguire i movimenti del robot con elevata precisione. Questo sistema è inoltre in grado di integrarsi facilmente con altri dispositivi e sistemi di automazione.

La struttura di questo manipolatore è caratterizzata da forme arrotondate che svolgono un ruolo molto importante nella promozione della sicurezza degli operatori. Queste forme gli garantiscono infatti

MINIMIZZAZIONE DEI RISCHI DI IMPATTO, infatti le forme arrotondate gli consentono di ridurre i rischi di lesioni in caso di contatto accidentale con il robot.

FACILITÀ DI MANOVRA, le forme arrotondate gli consentono infatti di muoversi in modo più fluido e naturale all'interno dell'ambiente di lavoro conferendogli la facoltà di evitare ostacoli con maggiore facilità

INTEGRAZIONE CON DISPOSITIVI DI SICUREZZA E' possibile infatti andare ad installare ulteriori sistemi di sicurezza, come gusci protettivi o barriere luminose. Questa operazione è resa più semplice grazie alle forme arrotondate del manipolatore.

ASPETTO VISIVO RASSICURANTE Aspetto da non trascurare è il fatto che, grazie a queste sue forme arrotondate, viene visto meno come una minaccia da parte degli operatori.

Questo manipolatore è stato inoltre pensato per avere una facile installazione, in quanto possiede un design compatto e leggero che lo rende adatto per essere integrato facilmente in qualsiasi area di lavoro o sistema esistente. E' inoltre caratterizzato da un'interfaccia di programmazione facile e intuitiva in quanto il tastierino da utilizzare per la programmazione implementa una nuova funzione di drag and drop dove per inserire le funzioni basta selezionare quella desiderata e trascinarla sullo schermo.



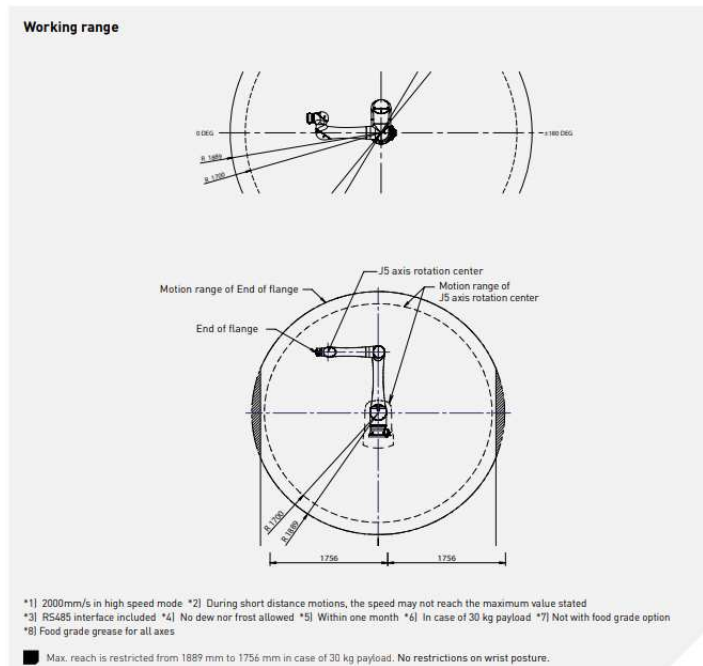
Figura 4.1: CoBot FANUC CRX-25iA

4.2 CARATTERISTICHE TECNICHE



Andando ad analizzare la scheda tecnica del manipolatore troviamo quanto riportato in tabella 4.1 e nelle figure 4.2a e 4.2b

Robot Specifications		Motion range [°]		Motion speed [° /s]	
Axes	6	J1	360	J1	80
Repeatability [mm]	±0.05	J2	360	J2	80
Mechanical weight [kg]	135	J3	540	J3	120
Payload Max [kg]	25/30	J4	380	J4	180
Max Reach [mm]	1889/1756	J5	360	J5	180
Max Linear Speed[mm/s]	1000	J6	450	J6	180

Tabella 4.1: Specifiche Robot, Velocità, Range movimento



(a) Area lavoro

 Robot	CRX-25iA
Robot footprint [mm]	283
Mounting position Floor	•
Mounting position Upside down	•
Mounting position Angle	•
 Controller	R-30iB Plus
Mini Plus	•
Dimensions [mm]	410 x 277 x 370
Tablet TP	•
Electrical connections	
Voltage 50/60Hz 3phase [V]	-
Voltage 50/60Hz 1phase [V]	200-230
Average power consumption [kW]	0.4
Integrated services	
Integrated signals on upper arm In/Out	5/6 *3
Integrated air supply	1
Environment	
Acoustic noise level [dB]	<70
Ambient temperature [° C]	0-45
Ambient humidity [%]	≤ 75 *4
Ambient humidity - Short term [%RH]	≤ 95 *5
Vibration acceleration [m/s ²]	≤ 4.9 [0.5G]
Protection	
Body standard/optional	IP67
Wrist & J3 arm standard/optional	IP67
Cleanroom class [ISO Class 2]	○ *7
Food grade / mechanical options	
Food grade grease	○ *8
White epoxy paint	○

(b) Controller, Alimentazione

Figura 4.2: Scheda tecnica

Andando ad analizzare il sistema nel suo complesso possiamo andare ad individuare i seguenti elementi:

- Base, fornisce stabilità al manipolatore. Sulla base sono presenti diversi connettori tra cui quello di alimentazione e di collegamento al controller esterno.
- Braccio robotico, costituito da una serie di giunti.
- Polso, dove sono posizionati diversi connettori per facilitare il montaggio di diversi gripper.
- Sensori di sicurezza, tra cui quello di forza che è integrato nella struttura e altri sensori esterni per il rilevamento di persone.
- Controller, esterno al robot, consente il controllo del robot e il collegamento via ethernet a sistemi di controllo esterni.

4.2.1 Controller

Il controller è un componente essenziale del robot che funge da controllore e gestore di tutte le operazioni del robot.

Il controllore non è integrato nel robot ma è posto esternamente ed è collegato tramite dei cavi che fungono anche da canale di comunicazione tra robot e controllore.

Il controller installato su questo manipolatore è il *FANUC R-30iB Mini Plus* che è la versione più compatta e avanzata del controller R-30iB.

Nonostante si tratta di un controller compatto ha una potente unità di elaborazione che gestisce tutte le operazioni del CoBot.

Altra caratteristica è che è in grado di offrire diverse opzioni di connettività per l'integrazione con altri dispositivi e sistemi di automazione. Ciò include connessioni Ethernet, USB e altre interfacce standard dell'industria.

E' dotato inoltre di funzionalità di sicurezza avanzate per garantire la collaborazione sicura tra il CoBot e gli operatori umani. Queste funzionalità includono la gestione dei sensori di sicurezza e la programmazione di zone di sicurezza per evitare collisioni e lesioni.

Andandolo ad analizzare esternamente (Figura 4.3) possiamo notare come questo sia composto da:

SELETTORE ON/OFF: Permette di accendere il robot. E' stato inserito un sistema di sicurezza a leva che permette di ruotare il selettore solo con la porta del controller chiusa.

SERRATURA: Necessaria per aprire la porta del controllore e avere accesso alla componente elettronica.

COMPARTO VENTOLE: Mantengono raffreddate le componenti elettroniche poste all'interno del controller.

CONNETTORE DI POTENZA PRIMARIA: Necessaria per alimentare il controller e il robot.

CONNETTORE PER IL CAVO DI TEACH PENDANT: E' un connettore che permette di collegare il tastierino. Qua'ora non fosse connesso il tastierino risulterebbe un circuito aperto e quindi va chiuso con un tappo apposito.

PORTA USB: Permette di inserire una chiavetta USB per effettuare eventuali backup.

CONNETTORE PER CAVO DI CONNESSIONE CON ROBOT: Permette di connettere il controller al robot e quindi di alimentarlo.

APERTURA PASSAGGIO CAVI: Spazio apposito per il passaggio dei cavi da collegare direttamente alle schede inserite dentro al controllore, come per esempio quello Ethernet per la connessione con il PLC.

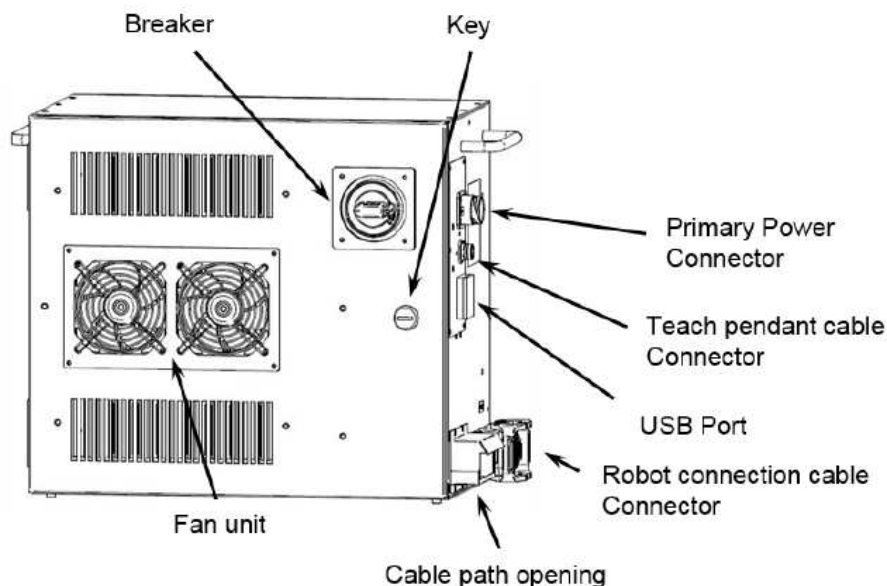


Figura 4.3: R-30iB Mini Plus visione esterna

Andandolo a vedere internamente (Figura 4.4) invece possiamo individuare i seguenti elementi

BACKPLANE BOARD: Serve a connettere la scheda principale a quelle opzionali.

BATTERIA AUSILIARIA: Ha diversi utilizzi, serve come *Backup di emergenza*, così da mantenere attive alcune funzionalità critiche del controller in caso di interruzione dell'alimentazione principale, *Avvio rapido*, *Fornitura di energia*, infatti può fornire energia aggiuntiva per dispositivi o componenti che richiedono un'alimentazione ausiliaria, *Gestione della sicurezza*, per mantenere attive alcune procedure di emergenza anche in caso di interruzione di corrente.

SIDE BOARD: Connette la scheda principale con la servo amplifier unit.

6-AXIS SERVO AMPLIFIER: Servono per governare i servomotori del robot.

UNITÀ DEGLI INGRESSI: Contiene la scheda degli input, l'unità di alimentazione principale e l'unità di interruzione dell'alimentazione.

La scheda degli input contiene un assorbitore di sovratensioni, un filtro EMC e un contatto magnetico.

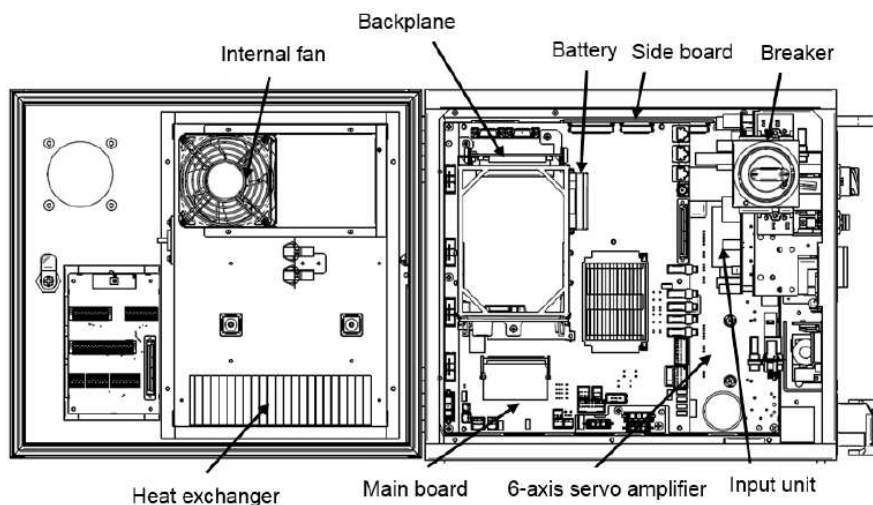


Figura 4.4: R-30iB Mini Plus visione interna

4.2.2 Sicurezza

Il robot FANUC CRX-25iA è progettato con diversi sistemi di sicurezza così da garantire un ambiente di lavoro sicuro sia per gli operatori umani che per l'attrezzatura circostante.

Tra i sistemi di sicurezza implementati nel manipolatore possiamo trovare

FRENO DI SICUREZZA Freno che impedisce al robot di muoversi quando viene rilevata un'anomalia, come una caduta di tensione o un guasto improvviso.

SISTEMA DI RILEVAMENTO COLLISIONI Il manipolatore è equipaggiato con sensori di collisione che rilevano la presenza di ostacoli nell'area circostante.

Questi sistemi possono essere configurati per andare a stoppare o rallentare il manipolatore in caso di rilevamento di una potenziale collisione.

ZONA DI SICUREZZA MONITORATA Consente di andare a definire delle zone entro le quali il manipolatore può lavorare senza rappresentare una fonte di rischio per gli operatori umani.

Se un operatore entra dentro una di queste zone il robot, a seconda della zona interessata, si ferma o rallenta in maniera tale da evitare il rischio di collisione.

SISTEMA DI CONTROLLO DEL LIMITE DI COPPIA Questo sistema serve a limitare la forza esercitata dal robot durante le operazioni di movimento. Questo sistema consente di andare a limitare lesioni o danni in caso di contatto accidentale.

DISPOSITIVI DI ARRESTO D'EMERGENZA Consentono agli operatori di fermare immediatamente il movimento del robot in caso di emergenza o pericolo imminente.

RICONOSCIMENTO DEGLI OGGETTI E' un sistema di sicurezza implementato da alcuni manipolatori che consente di andare a identificare oggetti all'interno dell'area di lavoro così che il robot possa adattare il proprio comportamento così da diminuire il rischio di collisioni accidentali.

Questi sono solo alcuni dei sistemi di sicurezza che vengono implementati nel CRX-25iA. Gli effettivi sistemi di sicurezza implementati nel manipolatore dipendono dal caso specifico e dalle normative vigenti.

4.2.3 *Sensori*

La scelta sei sensori da installare sul robot dipende dalle esigenze dell'applicazione. Possiamo tuttavia individuare una serie di sensori che vengono montati di base, ovvero gli *Encoder rotativi* e i *Sensori di coppia*.

Gli **encoder rotativi** è un dispositivo elettromeccanico in grado di convertire il movimento rotazionale in segnali digitali o

analogici. Questi dispositivi possono essere classificati in due categorie principali

ENCODER ROTATIVI INCREMENTALI producono una serie di impulsi elettrici che corrispondono a una quantità di movimento. La loro uscita non indica una posizione assoluta, ma piuttosto quanto si è mosso l'asse dalla posizione precedente.

ENCODER ROTATIVI ASSOLUTI forniscono un codice univoco per ogni posizione angolare, permettendo la determinazione immediata della posizione assoluta dell'asse rotante senza necessità di trovare un punto di riferimento.

Il principio di funzionamento di un encoder rotativo si basa sulla lettura di dischi che ruotano con l'asse. Questi dischi presentano una serie di tracce, generalmente realizzate in materiale ottico o magnetico, che vengono lette da sensori per generare i segnali di uscita.

Grazie a questi segnali di uscita riusciamo quindi ad ottenere un feedback di posizione indispensabile per il controllo preciso del movimento del robot e vengono spesso utilizzati anche per la sua calibrazione.

I **sensori di forza/coppia** sono invece dei sensori montati sui motori che servono a misurare la forza esercitata durante le operazioni di movimento.

Questi sensori si basano su celle torsionometriche, ovvero delle celle costituite da elementi piezoelettrici, strain gauge o altri elementi sensibili alla deformazione, montati in modo da rilevare la torsione o la flessione indotta dalla coppia. Quando viene applicata una forza o una coppia il materiale si deforma. Questa deformazione viene convertita in un segnale elettrico misurabile utilizzando i principi piezoelettrici o resistivi¹. A questo punto il segnale viene letto da un circuito di acquisizione dati o un modulo di interfaccia elettronica e successivamente viene elaborato dal sistema di controllo del robot.

Mediante l'utilizzo di queste celle il manipolatore è in grado di tradurre la coppia meccanica in un segnale di uscita digitale proporzionale alla coppia applicata. Questo garantisce una risposta molto rapida del CoBot in quanto si affida alla rilevazione diretta della forza o della coppia, invece che derivarne il valore con calcoli complessi che sfruttano le correnti del motore.

Questi misuratori monitorano la coppia meccanica lungo i vari giunti del CoBot e rileva la coppia applicata all'interno

¹ Se per esempio utilizziamo gli Strain Gauge, la deformazione modifica la resistenza elettrica del materiale generando così una variazione di tensione proporzionale alla forza applicata.

del motore e del riduttore. I dati ricavati vengono poi utilizzati dal sistema di controllo per disattivare il giunto evitando, così, danni al braccio robotico e agli operatori umani.

4.2.4 Polso

Il polso di questo manipolatore rappresenta una componente essenziale del braccio robotico, situata alla fine del segmento del braccio e responsabile di fornire movimenti precisi e versatili. È caratterizzato da un design robusto e compatto in grado di adattarsi a una vasta gamma di applicazioni industriali.

La struttura meccanica del polso è costruita con materiali che consentano di ridurre l'inerzia e migliorarne la manovrabilità. I giunti articolati da cui è composto gli consentono movimenti multidirezionali.

All'interno del polso sono integrati attuatori e motori ad alte prestazioni che forniscono la potenza necessaria per eseguire movimenti rapidi e controllati. Questi componenti sono gestiti da un sistema di controllo elettronico che interpreta i comandi provenienti dal sistema di controllo principale del robot e genera i segnali necessari per azionare i motori e controllare i movimenti del robot.

Il polso è stato inoltre progettato per facilitare il collegamento di end effector e gripper. Infatti, come possiamo notare dalla figura 4.5, sono presenti dei connettori, in particolare troviamo

EE INTERFACE 1 E 2, connettori in grado di fornire alimentazione e segnali di controllo all'end effector

CAMERA CABLE INTERFACE, connettore necessario nel caso in cui si vada a collegare una telecamera

USER AIR, connettore per il passaggio dell'aria

Questi connettori sono collegati direttamente alla base del robot, dove sono presenti altri connettori, permettendo così di poter effettuare i collegamenti necessari con l'end effector senza avere cavi che possano disturbare il movimento del manipolatore.

Al polso è associato il carico che può sostenere il braccio del CoBot. Come si può notare da quanto riportato in figura 4.6 più il *Tool Center Point* (TCP) si allontana dallo zero del polso più il carico che è in grado di movimentare il robot diminuisce.

Proprio per questo motivo risulta di particolare importanza andare a capire, a seconda dell'applicazione, quale end effector risulta ottimale per l'applicazione di interesse.

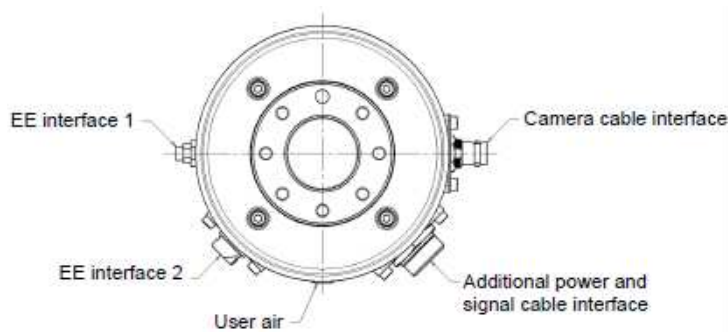


Figura 4.5: Polso

4.2.5 Base

La base del manipolatore è una componente molto importante del sistema robotico in quanto gli conferisce stabilità nelle operazioni di manipolazione. Questa può essere montata, a seconda delle esigenze imposte dall'applicazione, sia direttamente sul pavimento della fabbrica che su una piattaforma sollevata.

Sulla base del manipolatore sono presenti inoltre dei connettori che consentono il collegamento del robot con il controller e con altri dispositivi esterni. Tra questi possiamo trovare, come mostrato in figura 4.7:

- Connettore per alimentazione e trasmissione segnali
- Connettore aggiuntivo per alimentazione e trasmissione segnali
- Connettore per terminale di terra
- Connettore per cavo telecamera
- Connettore per aria

Questi ingressi sono direttamente collegati con i connettori posti sul polso, garantendo quindi il collegamento con eventuali end effector che necessitano di tali collegamenti senza però rischiare di andare a creare ingombro o impedimenti a movimento del manipolatore stesso.

4.2.6 Braccio Robotico

Nel braccio del robot possiamo trovare, come mostrato in figura 4.8, sei giunti a cui sono associati sei servomotori che conferiscono al manipolatore la capacità di eseguire i movimenti assegnati.

I giunti sono di tipo rotoidale così da consentire al manipolatore di muoversi nello spazio tridimensionale e quindi di offrire

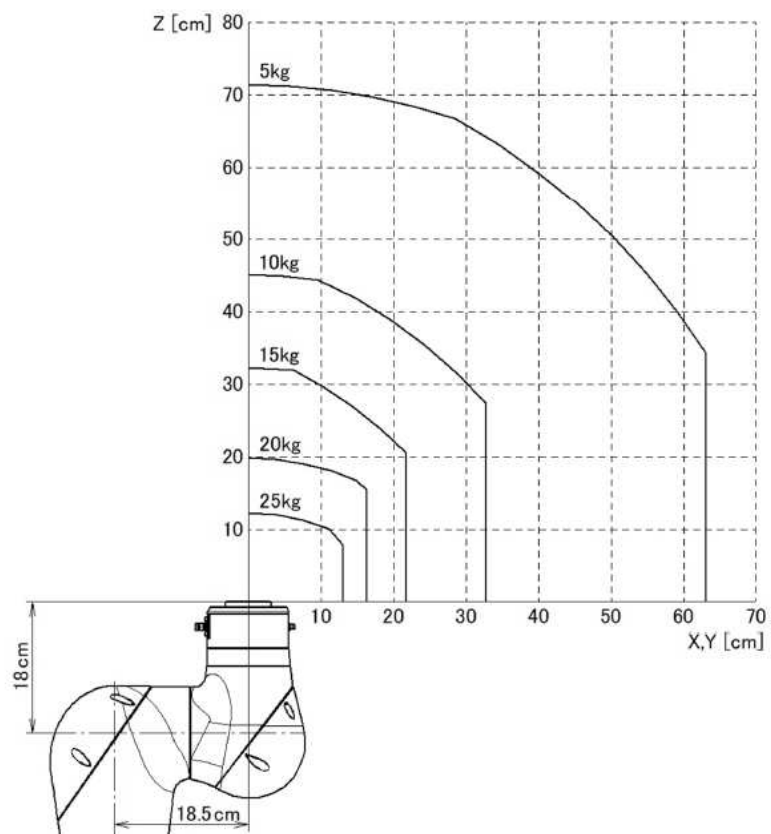


Figura 4.6: Diagramma payload polso

una vasta gamma di movimenti che possono essere coordinati per eseguire un'ampia varietà di compiti industriali.

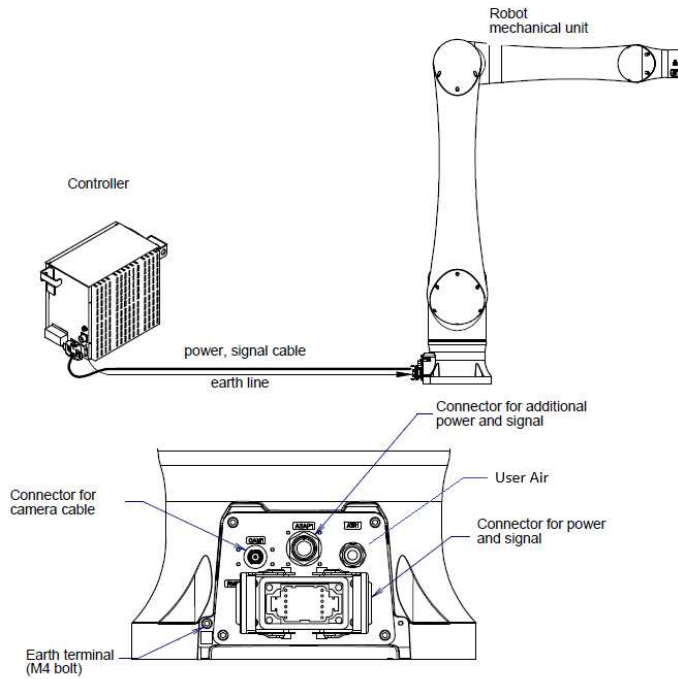


Figura 4.7: Connessione cavi CRX-25iA

Nel giunto J2 è possibile inoltre trovare le batterie di backup che consentono di mantenere salvata in memoria la posizione degli assi.

4.3 PROGRAMMAZIONE

Un programma contiene istruzioni di movimento, istruzioni di I/O, istruzioni registro e istruzioni condizione. Il robot eseguirà ogni istruzione in modo sequenziale così da completare il lavoro assegnatogli.

Per scrivere o correggere un programma viene usato il *Teach Pendant (TP)*.

In figura 4.9 possiamo vedere un esempio di programma in cui possiamo trovare istruzioni di movimento e macro istruzioni, ovvero istruzioni che richiamano ad altri programmi.

4.3.1 Dispositivi per la programmazione

I manipolatori della FANUC possono essere programmati in diversi modi, tra cui

TASTIERINO / TEACH PENDANT Metodo più comune usato per programmare un manipolatore FANUC.

Il tech pendant è un dispositivo portatile collegato al robot che consente agli operatori di inserire manualmente i

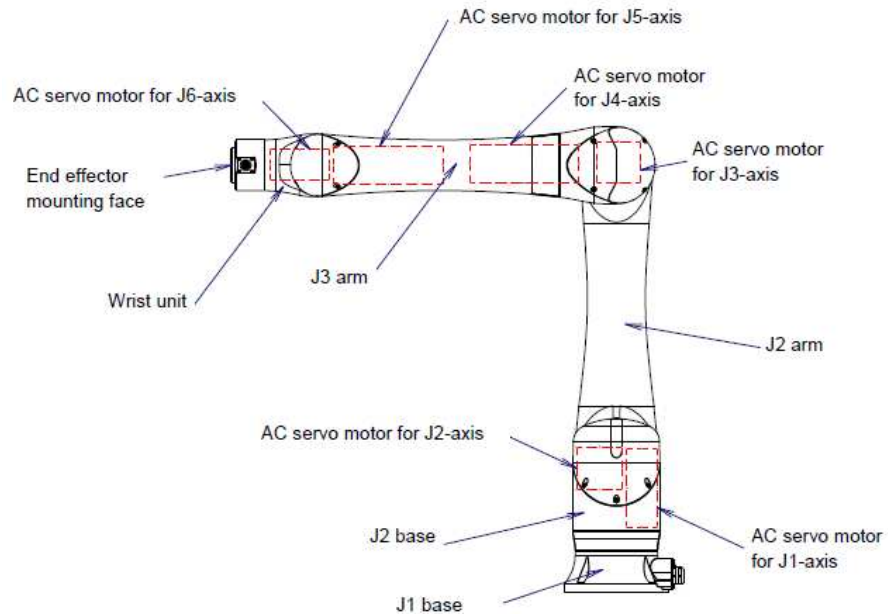


Figura 4.8: Configurazione unità meccanica

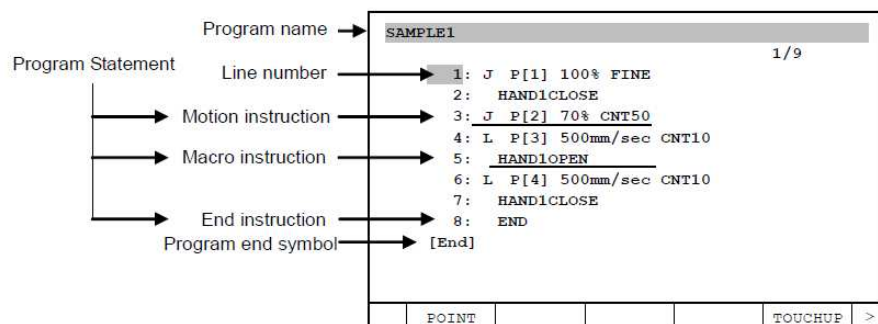


Figura 4.9: Esempio di programma

comandi e i movimenti del robot utilizzando un tastierino (Figura 4.10).

OFFLINE Attraverso l'utilizzo di un software dedicato il programmatore sviluppa o modifica un programma su un computer senza andare a interagire direttamente con il manipolatore. Una volta completato il programma, può essere caricato sul robot per l'esecuzione.

ISTRUZIONI DI ALTO LIVELLO Viene utilizzato un linguaggio di programmazione di alto livello, ad esempio KAREL o TP, che consente ai programmatori di scrivere script o sequenze di comandi più complessi per automatizzare determinate operazioni o processi.

Novità della serie CRX è legata alla possibilità di usare un tablet come tastierino per la programmazione del manipolatore (Figura 4.11). Con questa nuova possibilità è stata inserita anche



Figura 4.10: Tastierino per programmazione robot FANUC

la modalità di *drag and drop* delle istruzioni, ovvero permette di programmare la logica di movimento del robot andando a trascinare le funzioni sul programma.



Figura 4.11: Tablet per la programmazione robot FANUC

4.3.2 *Analisi codice*

Il programma di un robot Fanuc è costituito da una serie di istruzioni che indicano al robot come eseguire specifiche operazioni o compiti. Tra le istruzioni più comuni possiamo trovare

- Istruzioni I/O, gestiscono l'input e l'output di segnali dal robot
- Istruzioni di movimento, indicano al robot come muoversi nello spazio
- Istruzioni di controllo, gestiscono il flusso del programma come comandi per l'iterazione, la sezione condizionale e la chiamata di sottoprogrammi
- Funzioni di utilità, forniscono funzionalità aggiuntive al programma come comandi per calcolare e trasformare dati
- Commenti, utilizzati per documentare il programma e aggiungere annotazioni che facilitano la comprensione del codice da parte di operatori o dei programmatori

4.3.2.1 *Istruzioni di I/O*

I segnali di I/O sono segnali elettrici che permettono al controllore di comunicare con robot, end effector, equipaggiamento esterno e altri dispositivi periferici del sistema.

I segnali sono divisi in due gruppi: *I/O a uso generico* e *I/O specializzati*

I/O A USO GENERICO: sono segnali che può definire, se necessario, il programmatore.

In questo gruppo possiamo trovare:

- Digital I/O, ovvero un singolo bit
- Group I/O, ovvero 16 bit
- Analog I/O

I/O SPECIALIZZATI: sono segnali già definiti.

In questo gruppo possiamo trovare:

- I/O periferici
- I/O da pannello operatore
- I/O da robot

I segnali di I/O, siano questi a uso generico o specializzato, sono chiamati *Segnali Logici* e sono sottoposti, nel controller, a elaborazione del segnale. Invece i segnali derivanti da effettivi dispositivi di I/O sono chiamati *Segnali Fisici*. I segnali fisici specificano un dispositivo utilizzando un rack e uno slot e specificano ciascun segnale utilizzando il suo numero di segnale nel dispositivo.

Il *Rack* indica il tipo di modulo di I/O. La definizione dei rack si può trovare in figura 4.12a.

Oltre al rack troviamo anche lo *Slot*. Questo indica i numeri dei moduli I/O che compongono il rack.

Successivamente possiamo trovare il *Numero Fisico* che indica il numero del segnale nel modulo di I/O. Per controllare i segnali dei dispositivi di I/O nel controller del robot è necessario stabilire un'associazione tra segnali fisici e segnali logici. Questa associazione è chiamata assegnazione I/O. Generalmente questa associazione viene stabilita automaticamente, ma in alcuni casi questa può essere cambiata. In particolare abbiamo che per gli I/O digitali, analogici, group e periferici le porte logiche possono essere mappate sulle porte fisiche e possono essere ridefinite, mentre i numeri fisici degli I/O del robot e quelli del pannello operatore sono sempre gli stessi dei numeri logici e non possono essere cambiati.

Quando l'assegnazione I/O viene cancellata e viene alimentato il controller i dispositivi I/O connessi vengono riconosciuti e viene eseguita automaticamente una corretta assegnazione I/O.

In figura 4.12b si possono vedere i segnali digitali in ingresso e in uscita che sono già assegnati e che non possono essere cambiati con il loro numero fisico.

- 0 = Process I/O board, I/O link connection unit
- 1 to 15 = I/O Unit-MODEL A / B
- 32 = I/O link slave interface
- 33 = PMC (internal I/O assignment)
- 34 = Flags, Markers
- 35 = Always ON
- 36 = DCS Safe I/O (Non-safety function)
- 48 = R-30iB Mate main board (CRMA15, CRMA16)
- 66 = PROFIBUS-DP (master)
- 67 = PROFIBUS-DP (slave)
- 68 = FL-net (variety 1)
- 69 = FL-net status (variety 1)
- 75 = FIPIO (slave)
- 81 = DeviceNet (board 1)
- 82 = DeviceNet (board 2)
- 83 = DeviceNet (board 3)
- 84 = DeviceNet (board 4)
- 87 = RoboWeld
- 88 = Ethernet Global Data
- 89 = EthernetIP
- 90 = ArcLink
- 91 = WTC welder
- 92 = CC-Link
- 93 = InterBus (master)
- 94 = InterBus (slave)
- 95 = InterBus (CMD mode)
- 96 = Modbus TCP (This doesn't relate to R553 "HMI DEVICE COMMUNICATION" function. This is for R800 "Modbus TCP Interface".)
- 98 = InterBus slave only
- 99 = PROFINET I/O controller
- 100 = PROFINET I/O device
- 101 = Dual Channel PROFINET I/O controller
- 102 = Dual Channel PROFINET I/O device
- 103 = FL-net (variety 2)
- 104 = FL-net status (variety 2)
- 105 = CC-Link IE Field
- 106 = EtherCAT
- 107 = Tool Interface

(a) Definizione dei rack

Physical number	R-30iB maintenance manual
in 1	*IMSTP
in 2	*HOLD
in 3	*SFSPD
in 4	CSTOPI
in 5	FAULT RESET
in 6	START
in 7	HOME
in 8	ENBL
in 9	RSR1/PNS1
in 10	RSR2/PNS2
in 11	RSR3/PNS3
in 12	RSR4/PNS4
in 13	RSR5/PNS5
in 14	RSR6/PNS6
in 15	RSR7/PNS7
in 16	RSR8/PNS8
in 17	PNSTROBE
in 18	PROD START
in 19	DI01
?	
in N	DI(N-18)
?	
out 1	CMDENBL
out 2	SYSRDY
out 3	PROGRUN
out 4	PAUSED
out 5	HELD
out 6	FAULT
out 7	ATPERCH
out 8	TPENBL
out 9	BATALM
out 10	BUSY
out 11	ACK1/SNO1
out 12	ACK2/SNO2
out 13	ACK3/SNO3
out 14	ACK4/SNO4
out 15	ACK5/SNO5
out 16	ACK6/SNO6
out 17	ACK7/SNO7
out 18	ACK8/SNO8
out 19	SNACK
out 20	RESERVED
out 21	DO01
?	
out N	DO(N-20)
?	

(b) Assegnazione Digital I/O

Figura 4.12: Definizione rack e segnali digitali

4.3.2.2 Istruzioni di movimento

Un'istruzione di movimento muove l'utensile del robot in un specifico punto all'interno dell'area di lavoro con una specifica velocità e con un movimento specifico.

Per andare ad utilizzare queste istruzioni bisogna andare a specificare i seguenti elementi:

- Tipo di movimento, specifica come eseguire il movimento per raggiungere una posizione specifica
- Posizione obiettivo, specifica quale posizione deve raggiungere il robot
- Velocità, indica la velocità di movimento
- Modalità di posizionamento, specifica se posizionare il manipolatore in un punto specifico

- Specifiche di movimento aggiuntive, che permettono di andare a far eseguire al robot lavori particolari

Un'istruzione di movimento avrà la forma riportata in figura 4.13

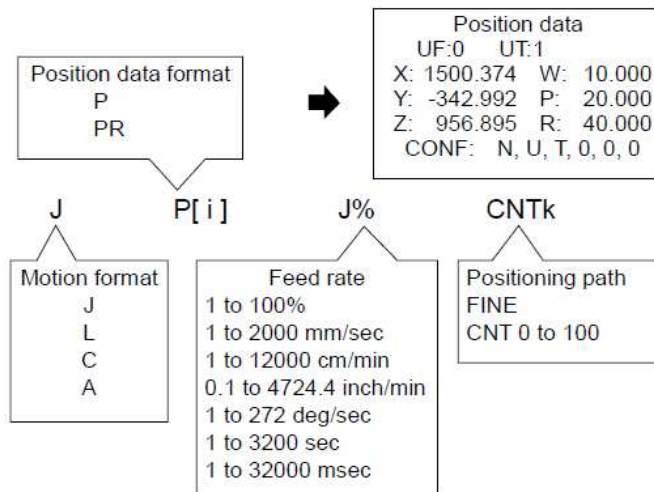


Figura 4.13: Istruzione di movimento

Per quanto riguarda il **tipo di movimento**, possiamo distinguere di quattro tipologie

JOINT MOTION (J): movimento base per muovere il robot in una specifica posizione. E' un movimento di tipo non lineare. La velocità può essere specificata come valore percentuale della velocità massima o andando a dare un tempo massimo (in secondi o millisecondi) per l'esecuzione del movimento. L'orientamento del tool non è controllato.

LINEAR MOTION (L): movimento che serve per controllare il percorso eseguito dal TCP che si muoverà linearmente. La velocità di movimento può essere fornita in *mm/s*, *cm/min*, *inch/min*, *inch/s*, *inch/ms*. L'orientamento del tool è gestito andando a vedere l'orientamento al punto iniziale e quello specificato per il punto finale. Possono essere anche compiute rotazioni sul posto, ovvero movimenti dove il TCP rimane fermo e si sposta solo il braccio robotico, in questo caso andremo a specificare la velocità di movimento come *deg/s*.

CIRCULAR MOTION (C): controlla il movimento del TCP da un punto iniziale ad un punto finale passando per un punto intermedio. Sia il punto intermedio che il punto finale sono specificati nella stessa riga di istruzione. La velocità di movimento può venire fornita in *mm/s*, *cm/min*, *inch/min*, secondi e millisecondi. Come per il movimento

lineare, l'orientazione del tool viene distinta andando a confrontare quella iniziale e quella obbiettivo.

CIRCLE ARC MOTION (A): a differenza del circular motion, nel circle arc motion abbiamo che i vari punti per cui deve passare il TCP sono specificati in linee diverse. In questo caso, quindi, l'istruzione viene eseguita andando a connettere le varie "righe di istruzioni" andando a generare il movimento ad arco specificato da programma.

La **posizione obbiettivo** include sia la posizione finale che l'orientamento del manipolatore stesso.

Possiamo distinguere due tipologie di dati di posizione: *coordinate dei giunti*, dove viene indicata la posizione che devono raggiungere singolarmente i vari giunti e *coordinate cartesiane*, dove viene indicata la posizione dell'utensile e la sua orientazione nell'area di lavoro.

COORDINATE CARTESIANE: sono caratterizzate dalla presenza di due elementi principali: *posizione*, che rappresenta la posizione del TCP rispetto l'origine del sistema di riferimento utilizzato, *orientazione*, ovvero l'inclinazione attorno all'asse X, Y, Z.

COORDINATE DEI GIUNTI: in questo caso viene fornita la posizione angolare dei giunti. E' formata da nove elementi, i primi sei sono le posizioni dei sei assi (tre della struttura base e tre del polso) e tre sono degli eventuali assi aggiuntivi.

Queste posizioni possono essere salvate in delle *variabili di posizione (P)* o in dei *registri di posizione (PR)*.

Una variabile di posizione è una variabile di tipo numerico utilizzata per memorizzare temporaneamente le coordinate di posizione durante l'esecuzione di un programma. Può essere usata per calcoli intermedi o per gestire dinamicamente le posizioni del robot all'interno del programma.

Una registro di posizione è, invece, un'area di memoria dedicata che viene utilizzata per memorizzare permanentemente le coordinate di posizione. Questi registri vengono usati solitamente per memorizzare punti specifici.

La **modalità di posizionamento** definisce il metodo con cui il robot deve raggiungere la posizione obbiettivo. Possiamo distinguere due modalità:

FINE, il robot raggiunge la posizione obbiettivo e si ferma un attimo prima di ripartire.

CNT, il robot approssia la posizione obiettivo ma, invece che fermarsi in quel punto, inizia a muoversi subito verso quello successivo. Quanto si deve avvicinare al punto obiettivo prima di passare a quello successivo viene definito attraverso un numero posto subito dopo la dicitura "CNT". Il valore che possiamo inserire va da 0 a 100, dove zero va a indicare che deve raggiungere esattamente quel punto prima di iniziare a muoversi per quello successivo e 100 indica che può passare il più lontano possibile da quel punto.

Un esempio delle due modalità è riportato in figura 4.14.

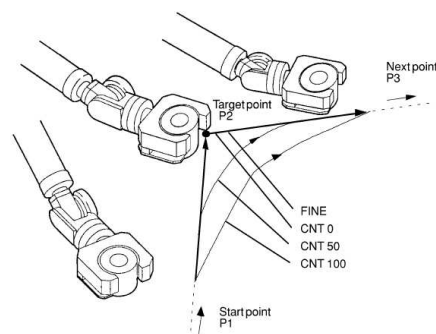


Figura 4.14: Esempio modalità di posizionamento

4.3.2.3 Istruzioni di controllo

Con istruzioni di controllo di intendono quelle istruzioni che vanno a modificare il flusso di esecuzione di un programma. Sono quindi quelle istruzioni che causano un salto da una linea del programma ad un'altra.

Possiamo individuare quattro tipi di istruzioni di controllo:

- Istruzioni label
- Istruzioni di fine programma
- Istruzioni di salto incondizionato
- Istruzioni di salto condizionato

Le **istruzioni label** vengono usate per specificare in che punto del programma saltare. Questa istruzione può essere usata sia per salti condizionati che per quelli incondizionati a seconda della necessità.

Le **istruzioni di fine programma** servono per indicare la fine di un programma. Quando questa istruzione viene eseguita il programma che stiamo eseguendo viene terminato e, se si tratta

di un programma richiamato dal main, ripassiamo il controllo al main.

Le **istruzioni di salto incondizionato** sono istruzioni che causano, incondizionatamente, un salto da una linea del programma ad un'altra. Possiamo trovare due tipi di istruzioni di salto incondizionato, *jump* e *call*.

L'istruzione JUMP causa un salto da una riga del programma ad un'altra specifica indicata dall'istruzione label.

Il comando per usare questa istruzione è

JMP LBL[i]

dove "i" indica a quale istruzione label deve saltare.

L'istruzione CALL serve invece per richiamare un altro programma caricato nel controller. Una volta richiamato il controllo viene passato a questo programma che lo ripasserà al main program una volta terminate le sue operazioni.

Il comando per usare questa istruzione è

CALL (Program Name)

Le **istruzioni di salto condizionato** causano un salto da una linea del programma ad un'altra qualora una determinata condizione venga soddisfatta. Possiamo individuare due istruzioni di questo tipo, *istruzione di comparazione* e *istruzione di selezione*.

L'istruzione di *comparazione* causa un salto a un specifico label o richiama un determinato programma se una determinata condizione è verificata. Questa istruzione può essere usata sia con i registri sia con gli I/O.

Nel primo caso avremo la seguente struttura

IF R[i] (operatore) (valore) (istruzione da eseguire)

ovvero va a confrontare il valore nel registro con il valore che abbiamo impostato e, se la condizione indicata dall'operatore è soddisfatta, andrà ad eseguire l'istruzione.

Nel secondo caso avremo una struttura del tipo

IF (variabile) (operatore) (valore) (istruzione da eseguire)

Come nel caso precedente verranno confrontati variabile, che questa volta sarà un valore di I/O, con il valore impostato e, se la condizione è soddisfatta, andrà ad eseguire l'istruzione.

E' possibile concatenare anche più condizioni insieme usando gli operatori logici "AND" e "OR".

L'istruzione di *selezione* consiste, invece, in diverse istruzioni di confronto tra registri. Si presenta in questa forma

```

SELECT R[i] = (valore1) (processo1)
            = (valore2) (processo2)
            = (valore3) (processo3)
            ELSE (processo4)

```

L'istruzione compara il valore del registro con i valori inseriti dal programmatore ed esegue l'istruzione della condizione che trova verificata.

4.3.2.4 Istruzioni di utilità

Con istruzioni di utilità si intendono tutte quelle funzioni che forniscono delle funzioni aggiuntive.

Tra quelle più importanti possiamo trovare

- Istruzione di wait
- Istruzione FOR/ENDFOR
- Istruzione payload
- Istruzione di pausa
- Istruzione di abort

L'**istruzione di wait** è usata per stoppare l'esecuzione del programma per uno specifico periodo di tempo o fino a quando una condizione specifica non è rispettata. Possiamo quindi distinguere due tipi di istruzioni di wait, *wait temporale* e *wait condizionale*.

Il *wait temporale* si presenta in questo modo

WAIT (time)

dove "time" può essere una costante o un valore salvato in un registro. Il programma, una volta arrivato a questa istruzione, si fermerà e attenderà il numero di secondi indicato.

Il *wait condizionale* si presenta invece come

WAIT (condizione) (istruzione da eseguire)

In questo caso, quando il programma arriva a questa istruzione, verifica che la condizione sia verificata e, se la verifica ha esito positivo, viene eseguita l'istruzione indicata.

La condizione viene scritta in questo modo

(variabile) (operatore) (valore)

dove la variabile può essere: un valore salvato in registro, una variabile di sistema o un segnale di I/O.

L'**istruzione FOR/ENDFOR** serve per andare a ripetere, per un numero di volte prestabilito, determinate operazioni.

Si presenta nella seguente forma

```
FOR (Contatore loop) = (Valore iniziale) TO (Valore target)
    (istruzioni)
ENDFOR
```

Quando il programma arriva alla linea del FOR il contatore loop viene settato al valore iniziale e vengono eseguite le istruzioni poste tra FOR e ENDFOR. Quando arriva a ENDFOR il programma verifica se il contatore loop è uguale al valore target. Se questo è minore allora gli viene incrementato il valore di uno e viene rieseguito il loop, altrimenti il programma esce dal loop ed esegue le istruzioni successive.

L'**istruzione payload** viene utilizzate in quelle situazioni in cui il payload montato sul robot cambi durante l'esecuzione del programma (per esempio in quelle situazioni in cui si ha il carico/scarico dei componenti). Questa istruzione permette quindi di andare a richiamare uno dei payload precedentemente salvati nella memoria del robot.

L'**istruzione di pausa** stoppa l'esecuzione del programma andando così a far rallentare e fermare il robot. Se il robot sta eseguendo un'operazione il programma si ferma prima che questa venga completata e, una volta che il programma viene riavviato, questo andrà ad eseguire l'operazione immediatamente successiva. Questa istruzione stoppa anche l'avanzamento del timer, qualora fosse attivo, per poi riattivalo al riavvio del programma.

L'**istruzione si abort** provoca l'interruzione dell'esecuzione del programma in corso, andando a causare il rallentamento e successivo arresto del manipolatore. Se il manipolatore sta eseguendo un'operazione questa viene interrotta prima del suo completamento. Quando viene utilizzata questa istruzione, l'esecuzione del programma non può essere ripresa.

4.4 ANALISI CINEMATICA

L'analisi cinematica di un robot è fondamentale per comprendere e controllare il movimento dello stesso.

Lo scopo di questa analisi è infatti quello di andare a determinare la relazione esistente tra i movimenti dei giunti del robot e la posizione/orientazione dell'end effector del robot nello spazio circostante.

Possiamo fare una distinzione tra *analisi cinematica diretta* e *analisi cinematica inversa*.

La prima risolve il problema dell'andare a determinare la posizione del robot per raggiungere un determinato punto. Questo problema ha sempre soluzione in quanto si conoscono i valori delle variabili di giunto e le dimensioni dei bracci per cui la soluzione è certa.

La seconda, invece, risolve il problema dell'andare a determinare la posizione dei giunti conoscendo quella che è la posizione dell'end effector. In questo caso la soluzione non sarà più certa, ma anzi potremmo trovarci con situazioni in cui la posizione dell'end effector sia raggiungibile da diverse posizioni del robot o problemi legati alle singolarità².

4.4.1 Convenzione di Denavit-Hartenberg

La **convenzione di Denavit-Hartenberg** è un metodo standardizzato utilizzato nella cinematica dei robot per descrivere la posizione e l'orientamento dei giunti e dei link di un manipolatore. Questa convenzione prevede l'utilizzo di quattro parametri per descrivere la trasformazione tra due link consecutivi, che sono:

α : Angolo di rotazione attorno all'asse x del link precedente

a : Lunghezza del link precedente misurata lungo l'asse x del link

d : Distanza lungo l'asse z tra i due link

θ : Angolo di rotazione attorno all'asse z del link successivo

Utilizzando questi quattro parametri è possibile andare a rappresentare la trasformazione da un sistema di riferimento al link precedente a quello del link successivo utilizzando una matrice di trasformazione omogenea, rappresentate la posizione e l'orientamento del link successivo rispetto al link precedente.

Per poter descrivere la trasformazione è per prima cosa necessario definire i sistemi di riferimento seguendo le regole riportate di seguito. Considerando due giunti consecutivi:

- l'asse z_{n-1} si sceglie coincidente con l'asse del giunto $i - 1$, mentre z_n coincidente con l'asse del giunto i
- l'asse x_{n-1} può essere scelto liberamente, ma è conveniente porlo in direzione del giunto successivo. Questo si interseca con z_{n-1} in corrispondenza del giunto $i - 1$. L'asse x_n corre lungo la normale comune fra gli assi z_{n-1} e z_n

² Con singolarità ci si riferisce a una situazione in cui il manipolatore perde uno o più dei suoi gradi di libertà.

- gli assi y_{n-1} e y_n sono scelti in modo da completare le terne seguendo la regola della mano destra

Riprendendo quindi i parametri sopra descritti avremo che:

- d sarà la distanza dell'asse z_{n-1} dalla normale comune
- θ sarà l'angolo di rotazione intorno all'asse z_{n-1} necessario per allineare x_{n-1} e x_n
- α sarà l'angolo di rotazione attorno alla normale comune per allineare z_{n-1} e z_n
- a sarà la distanza minima fra gli assi z_{n-1} e z_n

Ad ogni parametro è associata una matrice da cui si ottiene

$$T_{i,i-1} = R_x(\alpha_{i-1})T_x(a_{i-1})R_z(\theta_i)T_z(d_i) \quad (4.1)$$

dove con R si indica la matrice di rotazione e con T la matrice di traslazione.

4.4.2 Analisi cinematica con Denavit-Hartenberg

Per applicare Denavit-Hartenberg sul robot dobbiamo innanzi tutto andare a rappresentare i giunti e andare a capire come posizionare i vari sistemi di riferimento. Possiamo andare a trovare quanto necessario nelle figure 4.15a e 4.15b.

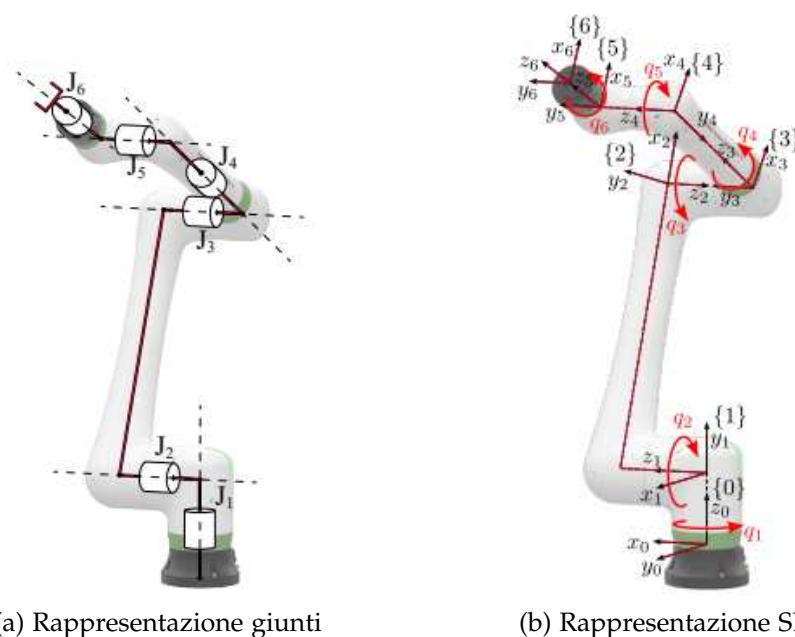


Figura 4.15: Rappresentazione giunti e relativi sistemi di riferimento

Link	$T_{i,i-1}$	α_{i-1} [rad]	a_{i-1} [mm]	θ_i [rad]	d_i [mm]
1	$T_{1,0}$	$\pi/2$	0	q_1	277.5
2	$T_{2,1}$	$-\pi$	950	q_2	274.2
3	$T_{3,2}$	$-\pi/2$	0	q_3	274.2
4	$T_{4,3}$	$-\pi/2$	0	q_4	750
5	$T_{5,4}$	$\pi/2$	0	q_5	185
6	$T_{6,5}$	0	0	q_6	180

Tabella 4.2: Parametri di Denavit-Hartenberg del FANUC CRX-25iA

Andando ora ad applicare la convenzione otteniamo quanto scritto in tabella 4.2

Grazie a questi parametri sarà possibile andare a risolvere il problema cinematico sia diretto che inverso.

4.4.3 Singolarità

Si parla di singolarità del manipolatore quando questo si trova in una situazione che causa la perdita di uno o più dei suoi gradi di libertà. Si verifica, per esempio, quando il manipolatore si trova in una configurazione che va a limitare la sua capacità di movimento.

Possiamo distinguere tre tipi di singolarità

SINGOLARITÀ CINEMATICHE: si verifica quando il robot raggiunge una configurazione in cui la matrice jacobiana del sistema diventa singolare o non invertibile. Questo implica che il robot ha perso la sua capacità di muoversi lungo una certa direzione nello spazio senza muoversi lungo un'altra, ovvero ha perso la capacità di raggiungere determinate posizioni o orientamenti.

SINGOLARITÀ DI MANIPOLABILITÀ: si riferisce alla perdita di capacità del robot di muoversi in modo efficiente in tutte le direzioni nello spazio. In una singolarità di manipolabilità il robot può continuare ad avere tutti i gradi di libertà, ma alcuni movimenti possono diventare molto inefficienti o impossibili.

SINGOLARITÀ FISICA: si verifica quando il robot raggiunge una configurazione in cui le sue parti meccaniche entrano in collisione tra loro o con l'ambiente circostante. In una singolarità fisica il robot non può muoversi senza causare danni a se stesso o agli oggetti circostanti.

Le singolarità di maggiore interesse sono quelle cinematiche in quanto vanno a rendere infinite le soluzioni della cinema-

tica inversa, oltre al fatto che causano una diminuzione della mobilità del manipolatore.

Possiamo distinguere tre tipi di singolarità cinematiche

SINGOLARITÀ DI GOMITO: si verifica quando il braccio si trova completamente disteso e il centro di polso sferico (CpS) si trova nel punto più distante, chiamato anche frontiera dello spazio di lavoro (figura 4.16a).

SINGOLARITÀ DI SPALLA: si verifica quando il centro di polso sferico si trova sul primo asse cinematico (figura 4.16b).

SINGOLARITÀ DI POLSO: avviene quando l'asse 4 e 6 sono coincidenti, ovvero quando i due assi dei giunti in considerazione sono sovrapposti (figura 4.16c).

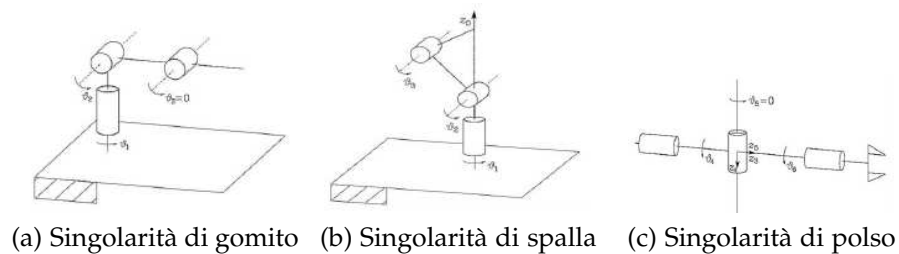


Figura 4.16: Tipi di singolarità cinematiche

Mentre le singolarità di spalla e gomito sono più facili da gestire in quanto evitabili con una corretta pianificazione della traiettoria, quella di polso è più complessa.

4.5 UTENSILE

L'utensile utilizzato per lo svolgimento della tesi è il **Gripper a vuoto Kenos - KVG60** (figura 4.17).

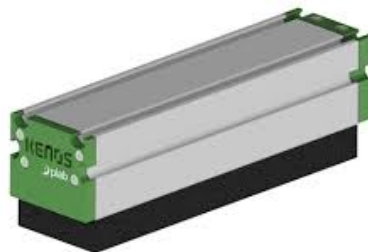


Figura 4.17: Gripper KVG60

Questo gripper è un dispositivo sviluppato dalla Piab, un'azienda specializzata nella fornitura di soluzioni di presa e mo-

vimentazione di materiali. Utilizza il principio del vuoto per afferrare e sollevare oggetti di varie forme e dimensioni in modo efficiente e sicuro.

E' caratterizzato da un design modulare che gli consente di essere facilmente integrato con il robot e macchinari esterni.

E' dotato inoltre di un efficiente sistema di vuoto che gli permette di creare una forte presa sull'oggetto da afferrare, anche se caratterizzato superfici irregolari o porose e si adatta facilmente a diversi materiali il che lo rende molto versatile.

Un aspetto caratteristico di questo gripper è legato al fatto che è dotato di valvole autoescludenti. Queste valvole consentono l'uso di pompe per vuoto più piccole, pur mantenendo il livello di vuoto anche se l'intera superficie del gripper non è in presa.

Le applicazioni più comuni sono la pallettizzazione e la movimentazione di materiali.

SVILUPPO DEL SOFTWARE DI CONTROLLO

Per controllare il manipolatore e permettergli di funzionare come pallettizzatore avremo bisogno della collaborazione tra due dispositivi, il robot e il PLC.

Di seguito sarà trattato e riportato il codice sviluppato sia lato del robot che lato del PLC.

5.1 ROBOT

Il lavoro del robot è quello di rimanere in attesa fino a quando il PLC non gli comunica la presenza di una scatola da prelevare nella zona di prelievo e, a quel punto, prelevarla. Successivamente si deve mettere in attesa che il PLC gli dia l'abilitazione al deposito. Una volta depositato il manipolatore deve tornare nella posizione di attesa.

5.1.1 Preparazione iniziale

Prima di procedere con la scrittura del codice bisogna andare a mettere il manipolatore nelle condizioni di comunicare con il PLC.

Per fare questo bisogna andare a impostare *Digital Input* (tab 5.1), *Digital Output* (tab 5.2), *Group Input* (tab 5.5), *Digital Output* (tab 5.6).

Gli ingressi e le uscite digitali sono state impostate come

#	RANGE	RACK	SLOT	START
1	DI[1 – 1024]	89	1	1
2	DI[1025 – 1052]	48	1	1
3	DI[1053 – 1104]	0	0	1

Tabella 5.1: Digital Input

#	RANGE	RACK	SLOT	START
1	DO[1 – 1008]	89	1	1
2	DO[1009 – 1024]	48	1	1
3	DO[1025 – 1104]	0	0	1

Tabella 5.2: Digital Output

Dove con *RANGE* si indica quanti bit andare ad utilizzare per quel rack. Il rack che verrà utilizzato è il numero 89 ovvero quello dedicato alla porta Ethernet.

Gli altri due range indicati servono come predisposizioni per eventuali modifiche future.

Per la scrittura di questo programma sono stati usati i seguenti DI (tab 5.3) e DO (tab 5.4)

ADDRESS	COMMENT
DI[1]	IMSTP (Stop Immediato del robot)
DI[2]	HOLD (Fermata in pausa)
DI[3]	SFSPD (Velocità controllata)
DI[4]	Cycle STOP
DI[5]	Fault RESET
DI[8]	Enable
DI[21]	Abilitazione ciclo MAN
DI[22]	Abilitazione ciclo AUT
DI[23]	PickCommand
DI[24]	PlaceCommand
DI[26]	PayloadConfirm

Tabella 5.3: DI Utilizzati

ADDRESS	COMMENT
DO[23]	ReadyToPick
DO[24]	ReadyToPlace
DO[25]	ACKPick
DO[26]	ACKPlace
DO[27]	Robot_In_Routine
DO[30]	EV_Presa

Tabella 5.4: DO Utilizzati

Per quanto riguarda i GROUP invece:

Quando si vanno a definire i *GROUP* non si definiscono più i range, ma si vanno a definire i singoli gruppi, da che posizione del rack partono e quanti bit sono dedicati a quel group. In questo caso avremo 34 GROUP INPUT, di cui alcuni messi

#	RACK	SLOT	START PT	NUM PTS
1	89	1	97	16
2	89	1	113	16
3	89	1	129	16
...	89	1	...	16
34	89	1	625	16

Tabella 5.5: Group Input

#	RACK	SLOT	START PT	NUM PTS
1	89	1	97	16
2	89	1	113	16
3	89	1	129	16

Tabella 5.6: Group Output

come predisposizione per modifiche future, ognuno da 16 bit tutti nel rack ethernet, mentre di GROUP OUTPUT ne avremo solo 3 sempre da 16 bit ciascuno e sempre sul rack 89.

In particolare sono stati definiti i GI riportati in tabella 5.7

Successivamente si procede alla definizione dei *PAYLOAD*. I payload sono importanti in quanto segnalano al manipolatore che condizioni deve aspettarsi e quindi adattare tutte le routine di sicurezza.

In questa fase sono stati definiti due payload diversi. Il primo payload, che è stato chiamato *Vuoto*, si riferisce alla situazione in cui abbiamo la presenza della sola ventosa. Il secondo payload, chiamato *Carico*, fa riferimento alla situazione in cui abbiamo sia la pinza che lo scatolone.

Infine si procede alla definizione di uno *USERFRAME* aggiuntivo, ovvero un sistema di riferimento diverso da quello originale del robot.

Questa operazione si fa per la zona di deposito così da poter dare le quote di posizionamento delle scatole riferite a questo nuovo sistema e non rispetto quello originale del manipolatore. Questo rende anche più facile andare a modificare le quote in caso di modifica della zona di deposito.

Per la zona di prelievo invece si è optato per il salvataggio della posizione nella memoria del robot.

ADDRESS	COMMENT
GI[1]	Velocità Robot Vuoto
GI[2]	Velocità Robot Carico
GI[4]	Offset X Prelievo
GI[5]	Offset Y Prelievo
GI[6]	Offset Z Prelievo
GI[8]	Offset X Deposito 1
GI[9]	Offset Y Deposito 1
GI[10]	Offset Z Deposito 1
GI[14]	TempoPrelievo
GI[15]	TempoDeposito
GI[16]	PickElevation
GI[17]	PlaceElevation
GI[28]	c1CountToEncoder
GI[29]	c1CountToEncoder
GI[33]	IndexPrelScatola
GI[34]	IndexDepScatola

Tabella 5.7: GI Utilizzati

5.1.2 Codice

Il software si compone di diversi programmi, con task diverse, che lavorano insieme per permettere al robot di compiere il lavoro desiderato.

Il software del manipolatore si compone di sette programmi. Tra questi troviamo un **main**, che rappresenterà il programma principale, nel quale saranno richiamati, con una certa logica, gli altri sotto programmi.

5.1.2.1 Programma MAIN

Il programma **main** sarà così strutturato

```

1: PAYLOAD[1:Vuoto] ;
2: OVERRIDE=100% ;
3: ;
4: !Posizione Home ;
5: CALL HOME_PALLETIZZATORE ;
6: ;
7: ;
8: !Main loop ;
9: LBL[1] ;

```

```

10: ;
11: ;
12: R[3:AtHome]=0 ;
13: !Ready to pick ;
14: WAIT DI[23:PickCommand]=OFF TIMEOUT,LBL[1] ;
15: DO[25:ACKPick]=OFF ;
16: DO[23:ReadyToPick]=ON ;
17: ;
18: ;
19: WAIT DI[23:PickCommand]=ON TIMEOUT,LBL[2] ;
20: ;
21: !Set Trigger ;
22: CALL OFFSET_PICK_CALCULATION ;
23: DO[23:ReadyToPick]=OFF ;
24: ;
25: !pick fisso ;
26: CALL PICK_FIX ;
27: ;
28:L PR[9:pTempDep] R[2:Vel_carico]mm/sec CNT100 ;
29: ;
30: !Ready to drop ;
31: LBL[3] ;
32: WAIT DI[24:PlaceCommand]=OFF TIMEOUT,LBL[3] ;
33: DO[26:ACKPlace]=OFF ;
34: DO[24:ReadyToPlace]=ON ;
35: WAIT DI[24:PlaceCommand]=ON ;
36: ;
37: CALL OFFSET_PLACE_CALCULATION ;
38: DO[24:ReadyToPlace]=OFF ;
39: ;
40: !Place fisso ;
41: CALL PLACE_FIX ;
42: ;
43: UFRAME_NUM=0 ;
44:L PR[1:pPrelievo] R[1:Vel_a_vuoto]mm/sec FINE
    Offset,PR[10:PickElevation] ;
45: ;
46: LBL[2] ;
47: !Verifica abilitazione ciclo ;
48: !lavoro ;
49: IF DI[22:AbilitazioneCiclo_AUT]=ON,
    JMP LBL[1] ;
50: ;
51: ;
52: !Posizione Home ;
53: CALL HOME_PALLETIZZATORE ;

```

All'accensione del manipolatore viene, come prima cosa, impostato il payload caricato e l'override. Queste due operazioni sono necessarie in quanto con il payload sbagliato il manipolatore percepirebbe una discrepanza tra quello che viene riferito dai suoi sensori e quello che viene riportato dal programma causando la comparsa di un errore e quindi il fermo immediato del manipolatore, mentre l'override è necessario in quanto permette al manipolatore di andare ad usare la velocità massima impostata dall'utente tramite PLC.

Fatte queste operazioni andiamo a chiamare la routine di *home* che porta il manipolatore in una posizione iniziale standard definita dal programmatore.

A questo punto comincia il *main loop*, ovvero quella serie di operazioni che il manipolatore dovrà compiere ciclicamente fino a quando non gli viene comunicato dal controllore di smettere. La condizione che determina la ripetizione del ciclo è la presenza o meno del segnale DI[22: AbilitazioneCiclo AUT], ovvero un segnale inviato da PLC.

La prima cosa che viene fatta una volta entrati nel ciclo è quella di andare a porre a zero il valore R[3:AtHome] che serve per andare ad indicare se il manipolatore si trova o meno nella posizione di home.

Successivamente il manipolatore viene posto in attesa che il segnale DI[23: PickCommand] venga abbassato così da evitare possibili situazioni in cui il segnale rimanga alzato anche se non è presente alcun pacco da prelevare.

A questo punto il manipolatore abbassa il DO[25: ACKPick] e alza il segnale DO[23: ReadyToPick]. In questo modo comunica al controllore che è libero per iniziare a lavorare.

Il robot rimane poi in attesa del comando di pick. Qualora questo non arrivasse entro un certo periodo di tempo il programma salterebbe alla linea 46 e andrebbe a verificare se il plc lo abilitasse ancora al lavoro o meno.

Nel momento in cui il manipolatore riceve il comando di pick chiama la routine di *OFFSET_PICK_CALCULATION*, ovvero una routine che serve al manipolatore per andare a calcolare gli offset mandati dal controllore, abbassa il segnale DO[23: ReadyToPick] e chiama la routine di *PICK_FIX*, ovvero una routine contenente tutte le istruzioni necessarie al manipolatore per lavorare correttamente.

Una volta prelevato il pacco il manipolatore si sposta in una posizione temporanea in attesa che il segnale di place, DI[24: PlaceCommand], venga prima abbassato e successivamente, dopo che il manipolatore abbia comunicato al controllore di essere libero per il deposito, venga alzato nuovamente.

A questo punto, dopo il calcolo degli offset e aver comunicato al PLC di non essere più libero, chiama la routine di place.

Una volta depositato il pacco viene cambiato il sistema di riferimento, modificato dentro la routine di place, impostandolo nuovamente con quello standard del robot e viene fatto muovere il manipolatore nella posizione di attesa. La posizione di attesa è stata impostata come la posizione di prelievo solamente che è stato impostato un offset in Z in maniera tale che non causi ingombro con l'arrivo dei pacchi.

Una volta arrivato alla posizione di attesa viene controllato se la condizione per la ripetizione del main loop è verificata. Nel caso questa fosse verificata allora il programma tornerebbe alla riga 9 e ripeterebbe le operazioni sopra descritte, altrimenti verrebbe chiamata nuovamente la routine di home e usciremmo dal programma.

5.1.2.2 *Programma HOME_PALLETIZZATORE*

Il programma di homing è così strutturato

```

1: D0[27:Robot_In_Routine]=ON ;
2: !Settaggio tool e frame ;
3: UFRAME_NUM=0 ;
4: PAYLOAD[1:Vuoto] ;
5: D0[30:EV_Presa]=OFF ;
6: ;
7: IF R[3:AtHome]=1,JMP LBL[1] ;
8: !Go home ;
9: PR[5:pTemp]=LPOS ;
10: ;
11: ;
12: ;
13: ;
14: PR[5,3:pTemp]=250 ;
15: ;
16: ;
17: ;
18: !----- ;
19:L PR[5:pTemp] R[1:Vel_a_vuoto]mm/sec FINE ;
20:L PR[6:pHome] R[1:Vel_a_vuoto]mm/sec FINE ;
21: ;
22: R[3:AtHome]=1 ;
23: LBL[1] ;
24: D0[27:Robot_In_Routine]=OFF ;

```

Dopo aver comunicato al controllore che il manipolatore è entrato in una routine e che quindi non può ricevere altri comandi, andiamo a cambiare il sistema di riferimento e il payload.

Queste due istruzioni sono state implementate in quanto se il manipolatore viene spento durante una routine che prevede un sistema di riferimento diverso o un payload diverso il robot si comporterebbe in maniera inaspettata o si bloccherebbe. Per lo stesso motivo viene anche posto a OFF il DO[30: EV_Presa] così da comunicare al PLC di spegnere l'aria nel caso in cui questo la stia tenendo ancora accesa.

Fatte queste operazioni, che servono per portare il manipolatore in una condizione di inizio o fine ciclo di lavoro nota, viene verificato se il manipolatore si trova già nella posizione di home (usando il valore in registro R[3: AtHome]). Se il valore salvato in registro è pari a 1 allora il manipolatore esce dalla routine, se invece è pari a 0 procede con le operazioni che lo porteranno a raggiungere la posizione desiderata.

Come prima operazione andiamo a salvare nel PR[5:pTemp] la posizione in cui si trova il manipolatore in questo momento. Vengono salvate le coordinate cartesiane e non le posizioni dei giunti in quanto la posizione di home viene fornita come una posizione cartesiana.

Alla posizione temporanea andiamo a modificare il valore della coordinata Z così da portarci ad una altezza che ci garantisca di riuscirci a muovere senza rischiare collisioni e successivamente ci muoviamo rispetto quella posizione modificata con velocità impostata dall'utente.

Una volta raggiunta questa posizione ci possiamo infine muovere verso la posizione di home e, una volta arrivati, impostare il valore nel registro R[3: AtHome] a 1 e andare a segnalare al controllore l'uscita dalla routine abbassando l'apposito segnale.

5.1.2.3 Programma OFFSET_PICK_CALCULATION

La routine di pick è così strutturata

```

1:  ! OFFSET X PRELIEVO ;
2:  R[50]=GI[4:OffsetXPrelievo]    ;
3:  IF (R[50]<32767),JMP LBL[1] ;
4:  R[4:OffsetXPrelievo]=R[50] - 65536    ;
5:  JMP LBL[2] ;
6:  LBL[1] ;
7:  R[4:OffsetXPrelievo]=R[50]    ;
8:  LBL[2] ;
9:  !OFFSET Y PRELIEVO ;
10: R[50]=GI[5:OffsetYPrelievo]    ;
11: IF (R[50]<32767),JMP LBL[3] ;
12: R[5:OffsetYPrelievo]=R[50] - 65536    ;
13: JMP LBL[4] ;
14: LBL[3] ;

```

```

15: R[5:OffsetYPrelievo]=R[50] ;
16: LBL[4] ;
17: !OFFSET Z PRELIEVO ;
18: R[50]=GI[6:OffsetZPrelievo] ;
19: IF (R[50]<32767),JMP LBL[5] ;
20: R[6:OffsetZPrelievo]=R[50]-65536 ;
21: JMP LBL[6] ;
22: LBL[5] ;
23: R[6:OffsetZPrelievo]=R[50] ;
24: LBL[6] ;
25: !OFFSET rZ PRELIEVO ;
26: R[50]=GI[7:RzPrelievo] ;
27: IF (R[50]<32767),JMP LBL[7] ;
28: R[7]=R[50]-65536 ;
29: JMP LBL[8] ;
30: LBL[7] ;
31: R[7]=R[50]/10 ;
32: LBL[8] ;
33: PR[3,1:OffsetPrelievo]=R[4:OffsetXPrelievo] ;
34: PR[3,2:OffsetPrelievo]=R[5:OffsetYPrelievo] ;
35: PR[3,3:OffsetPrelievo]=R[6:OffsetZPrelievo] ;
36: PR[3,6:OffsetPrelievo]=R[7] ;
37: !OFFSET Z PRELIEVO ;
38: R[50]=GI[16:PickElevation] ;
39: IF (R[50]<32767),JMP LBL[9] ;
40: R[16]=R[50]-65536 ;
41: JMP LBL[10] ;
42: LBL[9] ;
43: R[16]=R[50] ;
44: LBL[10] ;
45: PR[10:PickElevation]=PR[3:OffsetPrelievo] ;
46: PR[10,3:PickElevation]=R[16] ;

```

In questo programma andiamo determinare il valore di offset che andiamo a passare dal controllore. Nei controllori numerici della fanuc il numero massimo rappresentabile è 32767, quindi possiamo andare da un valore negativo pari a -32767 a un massimo positivo di 32767. Nel caso avessimo valori maggiori dovremmo implementare una routine che ce lo trasformi nel rispettivo valore negativo così che il robot possa usarlo. Per fare questo viene implementata una routine dove se il valore mandato da PLC è minore di 32767 verrà usato senza modifiche, se invece è maggiore gli verrà sottratto 65536 così da ottenere il corrispettivo valore negativo.

Questa operazione verrà fatta per gli offset di x, y, z e rotazione attorno a z.

La routine di *OFFSET_PLACE_CALCULATION* sarà uguale se non per le variabili di offset utilizzate.

5.1.2.4 Programma *PICK_FIX*

La routine di pick è così strutturata

```

1: DO[27:Robot_In_Routine]=ON ;
2: UFRAME_NUM=0 ;
3: PAYLOAD[1:Vuoto] ;
4: ;
5: !approccio ;
6: !Mi porto sopra pPick con un ;
7: !certo offset ;
8:L PR[1:pPrelievo] R[1:Vel_a_vuoto]mm/sec FINE
   Offset,PR[10:PickElevation] ;
9: ;
10: DO[30:EV_Presa]=ON ;
11: ;
12: !pPrelievo ;
13:L PR[1:pPrelievo] R[1:Vel_a_vuoto]mm/sec FINE
   Offset,PR[3:OffsetPrelievo] ;
14: ;
15: !Attesa tempo di presa ;
16: WAIT R[14] ;
17: PAYLOAD[2:Carico] ;
18: ;
19: !retreat ;
20:L PR[1:pPrelievo] R[2:Vel_carico]mm/sec CNT100
   Offset,PR[10:PickElevation] ;
21: DO[25:ACKPick]=ON ;
22: DO[27:Robot_In_Routine]=OFF ;

```

Una volta arrivato il comando di presa entriamo nella routine di pick.

Per la presa del pezzo ci spostiamo dapprima nella posizione di presa con un certo offset in z, comunichiamo al controllore di accendere l'aria e iniziamo a scendere verso la posizione di presa vera e propria. Una volta raggiunta questa posizione attendiamo un certo periodo di tempo, impostato da plc, per essere sicuri che la presa sia avvenuta correttamente e cambiamo il payload impostato.

A questo punto facciamo l'operazione di ritirata, dove semplicemente ci alziamo di un certo valore di offset.

Infine diamo conferma al plc di aver finito l'operazione di pick e usciamo dalla routine.

5.1.2.5 Programma PLACE_FIX

La routine di place è molto simile alla routine di pick, ma presenta delle piccole modifiche

```

1: D0[27:Robot_In_Routine]=ON ;
2: PAYLOAD[2:Carico] ;
3: ;
4: UFRAME_NUM=2 ;
5: ;
6: !approccio ;
7:L PR[2:pDeposito] R[2:Vel_carico]mm/sec FINE
   Offset,PR[11:PlaceElevation] ;
8: ;
9: ;
10: !Set Trigger ;
11: D0[24:ReadyToPlace]=OFF ;
12: ;
13: D0[30:EV_Presa]=OFF ;
14: ;
15: !pDeposito ;
16:L PR[2:pDeposito] R[2:Vel_carico]mm/sec FINE
   Offset,PR[4:OffsetDeposito] ;
17: ;
18: !Attesa tempo di deposito ;
19: WAIT R[15] ;
20: D0[26:ACKPlace]=ON ;
21: PAYLOAD[1:Vuoto] ;
22: ;
23: !retreat ;
24:L PR[2:pDeposito] R[1:Vel_a_vuoto]mm/sec CNT100
   Offset,PR[11:PlaceElevation] ;
25: D0[27:Robot_In_Routine]=OFF ;

```

Come si può notare dal codice sopra riportato andremo ad usare un payload diverso, in quanto in questo caso avremmo il peso dato dalla ventosa più quella dello scatolone, inoltre andremmo ad usare uno USERFRAME diverso, in quanto ne andremmo ad usare uno specifico della baia di deposito.

Il payload verrà nuovamente cambiato una volta che l'oggetto sarà stato depositato.

5.2 PLC

Il PLC ha il compito di gestire tutta la logica della funzione di pallettizzatore. Il programma che dovrà svolgere dovrà quindi contenere la logica per: avvio, marcia e start e stop del robot, lo

scambio dati con il manipolatore, la gestione del prelievo e del deposito delle scatole e la gestione della ventosa.

In questo capitolo andremo ad analizzare il codice implementato sul controllore.

5.2.1 Codice

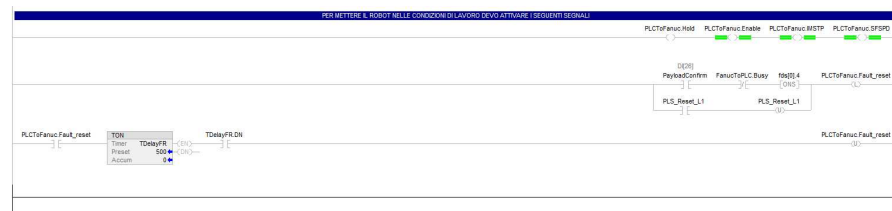
Nel codice possiamo distinguere due sezioni, la sezione *main task* e la sezione *robots*.

La prima sezione contiene tutte le funzioni inerenti la gestione del robot come pallettizzatore, la seconda invece serve per andare ad abilitare il manipolatore ad eseguire o meno il suo programma.

5.2.1.1 MAIN TASK: A00_AvvioRobot



(a) AvvioRobot prima parte



(b) AvvioRobot seconda parte

Figura 5.1: A00_AvvioRobot

In questa routine andiamo a gestire la fase di accensione del manipolatore.

Durante l'accensione il manipolatore richiede, per poter lavorare, la conferma del payload. Questa conferma viene gestita dal PLC tramite questa la logica riportata nelle figure 5.1a e 5.1b.

Oltre alla logica di conferma del payload possiamo trovare la logica per l'accensione di segnali essenziali per il funzionamento del manipolatore e quella di reset degli errori.

Il reset degli errori dovrà avvenire alla prima accensione del manipolatore dopo la conferma del payload in quanto la richie-

sta di conferma del payload genera un errore che impedisce al manipolatore di funzionare senza un reset degli stessi.

5.2.1.2 MAIN TASK: A04_Pinza

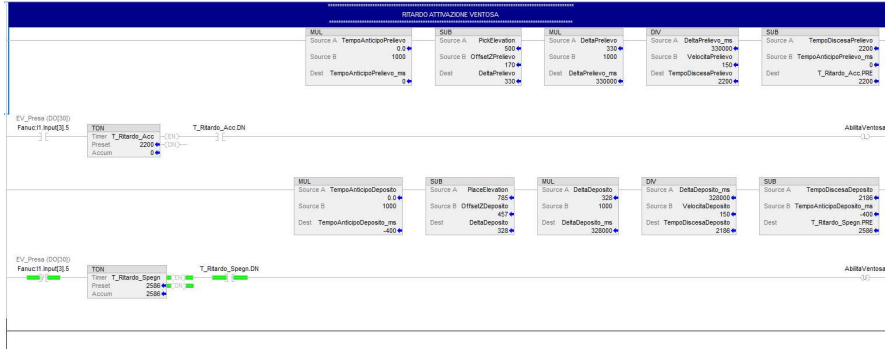


Figura 5.2: A04_Pinza

Questa routine si occupa della gestione delle ventose, in particolare va a gestira l'anticipo e il ritardo di spegnimento e accensione aria.

Quando il manipolatore manda il segnale che determina l'accensione o lo spegnimento dell'aria, il plc anticipa o ritarda l'operazione basandosi sul tempo indicato dall'operatore e in base alla velocità del manipolatore.

5.2.1.3 MAIN TASK: A08:GestioneDeposito

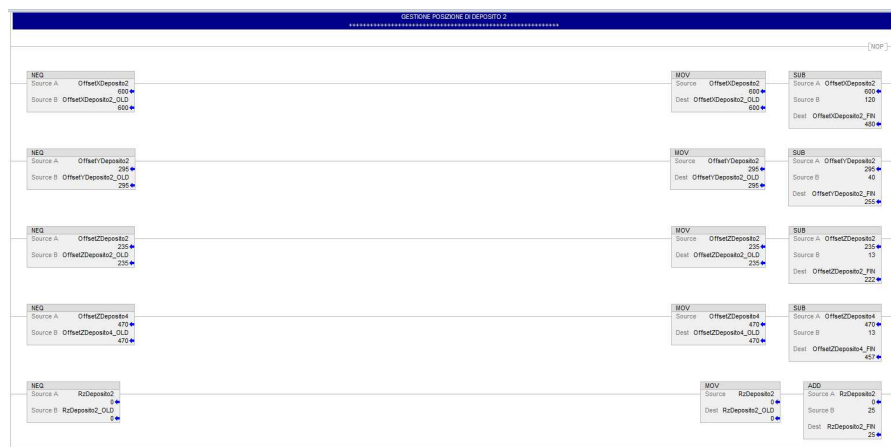
Con questa sezione si vuole andare a gestire l'offset del deposito da passare al manipolatore. Nel deposito verranno posizionate quattro scatole, due scatole affiancate alla base e due posizionate sopra quelle precedenti.

Con questo programma si vanno a impostare le posizioni delle zone di deposito (fig 5.3a e 5.3b) e si va a implementare una logica che, a seconda della scatola da depositare, passerà la posizione corretta al manipolatore (fig 5.3c).

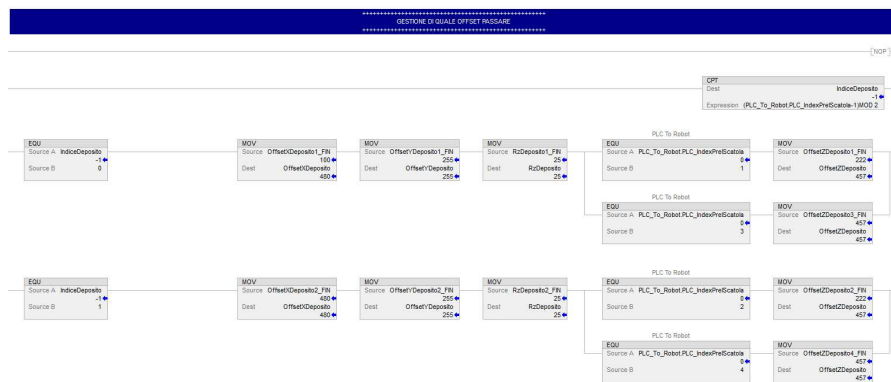
E' stata inoltre implementata la routine per il passaggio della PlaceElevation (fig 5.3d).



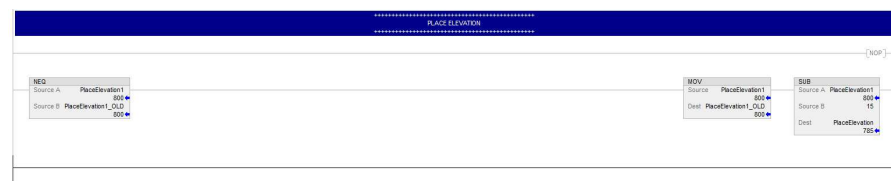
(a) GestioneDeposito prima parte



(b) GestioneDeposito seconda parte



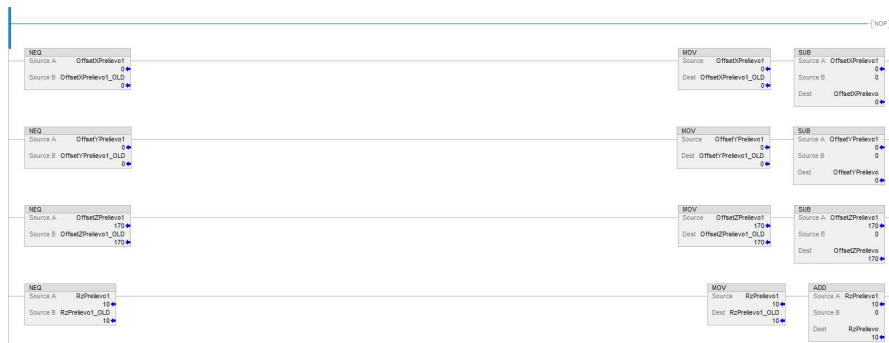
(c) GestioneDeposito terza parte



(d) GestioneDeposito quarta parte

Figura 5.3: A08_GestioneDeposito

5.2.1.4 MAIN TASK: A_09:GestionePrelievo



(a) Gestione prelievo prima parte



(b) Gestione prelievo seconda parte

Figura 5.4: A09_GestionePrelievo

In maniera analoga a quanto fatto per il programma di gestione dei depositi, è stata scritta la routine di gestione del prelievo e della *PickElevation*.

La differenza rispetto alla situazione di deposito risiede nel fatto che il punto di prelievo è unico e fisso, quindi viene passata solo una serie di offset x, y, z e rotazione attorno a z.

Come si può vedere dall'immagine 5.4a gli unici valori che vengono mandati sono quelli riguardanti la z e la rotazione attorno a z. Questo è dovuto al fatto che essendo il punto di prelievo un punto fisso lo si è preferito salvare nel manipolatore come una posizione registrata, andando quindi a fornire solo gli offset necessari per gli aggiustamenti necessari.

5.2.1.5 MAIN TASK: A_98:ScambioDatiFanuc

In questa sezione del programma sono presenti i collegamenti tra ingressi/uscite digitali del PLC con uscite/ingressi del manipolatore. Questo collegamento è necessario per permettere il trasferimento dei dati da un sistema all'altro. Per fare questo è necessario collegare, mediante l'utilizzo di indirizzi dipendenti dal cablaggio effettuato, i segnali interessati alle rispettive porte di input/output.

Un esempio di collegamenti tra uscita del PLC e ingresso del manipolatore è riportato in figura 5.5

Mentre un esempio di collegamento tra ingresso del PLC e uscita del manipolatore è riportato in figura 5.6



Figura 5.5: Esempio collegamento uscita PLC e ingresso manipolatore



Figura 5.6: Esempio collegamento ingresso PLC e uscita manipolatore

In questa sezione sono state riportate inoltre la logica di arresto del manipolatore nel caso in cui questo non riesca più a comunicare con il PLC (figura 5.7a) sia la logica che permette di attivare la ventosa nel momento in cui arriva il segnale di accensione da parte del manipolatore (figura 5.7b)



(a) Logica controllo connessione Ethernet



(b) Logica azionamento ventosa

Figura 5.7: A98_ScambioDatiFanic

Nel caso di errore di connessione Ethernet abbiamo che, nel momento in cui il PLC non riesce più a comunicare con il manipolatore, viene alzato il segnale di errore e questo va a togliere l'abilitazione al ciclo di lavoro automatico.

La logica di controllo della ventosa invece consente la sua attivazione solo se la comunicazione sta avvenendo correttamente e se arriva il segnale di richiesta di attivazione dell'aria da parte del manipolatore.

5.2.1.6 ROBOTS: A01_StartStop

In questa routine andiamo a impostare una logica che abiliti o disabiliti il manipolatore a eseguire il suo ciclo di lavoro.

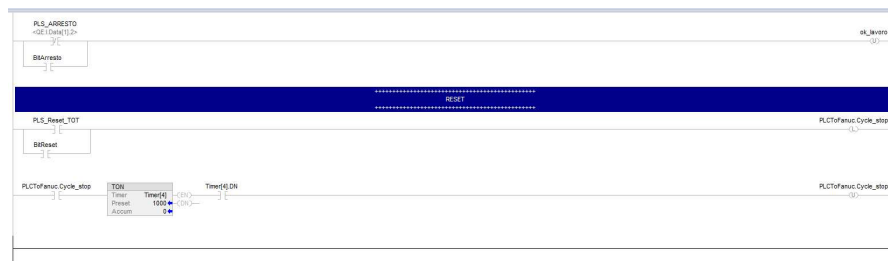
In figura 5.8a possiamo vedere la logica che permette l'avvio del manipolatore. Per permettere l'avvio del manipolatore si dovrà andare a premere il pulsante di marcia per due secondi. Questo, qualora tutte le ulteriori condizioni (sicurezze tutte attive, robot non in errore, pulsante di arresto non premuto e abilitazione al lavoro alzata) permetterà di alzare il bit di *start* del manipolatore. Lo start andrà poi ad alzare il bit di *ok_lavoro* che andrà a selezionare il ciclo di lavoro automatico.



(a) StartStop prima parte



(b) StartStop seconda parte



(c) StartStop terza parte

Figura 5.8: A01_StartStop

In figura 5.8b e inizio 5.8c possiamo vedere la routine di arresto del manipolatore. Nel caso in cui si verificasse un fault del manipolatore o non ci sia l'enable generale verrà alzato il bit di *Cycle_stop* che impone al manipolatore lo stop immediato. Verrà inoltre abbassato il bit di *ok_lavoro* così da impedire al manipolatore di ripartire una volta resettati gli errori a meno che non venga premuto nuovamente il pulsante di marcia.

E' stata inserita inoltre la logica per il pulsante di arresto. Nel caso venga premuto questo pulsante il verrà abbassato il bit di *ok_lavoro* e il manipolatore, una volta terminato il suo ciclo, si fermerà.

In figura 5.8c è stata inserita la logica di *abort* del programma. L'abort del programma è essenziale quelle volte in cui il manipolatore si ferma per degli errori o quando si vuole far ricominciare da zero il ciclo di lavoro del manipolatore. Per fare questo quando premiamo il pulsante di reset totale alziamo il bit di *cycle_stop*. Dopo un secondo questo viene riabbassato e procede all'abort del programma. Il bit viene tenuto alzato un secondo in quanto se così non fosse sarebbe un segnale troppo veloce per essere letto dal PLC.

CONCLUSIONI

I robot sono stati sviluppati per aiutare i lavoratori a svolgere le operazioni rischiose e gravose per l'essere umano. Mentre inizialmente erano limitati a lavorare circondati da barriere e protezioni per non causare pericolo per gli operatori, con lo sviluppo tecnologico sono nati i robot collaborativi che non necessitano più di barriere che li separino dall'uomo ma, grazie a una serie di sensori e funzioni di sicurezza integrate, sono in grado di lavorare a stretto contatto con esso.

In commercio sono presenti un gran numero di cobot che possono spaziare nelle più disparate applicazioni.

In questo elaborato si è andato ad analizzare il manipolatore CRX-25iA della Fanuc andando a vedere come questo possa essere utilizzato come pallettizzatore.

Andando ad analizzare la scheda tecnica si è visto come questo manipolatore abbia una capacità di carico di 25kg che lo rende idoneo ad essere impiegato come pallettizzatore. Inoltre si è potuto vedere come questo abbia un'area di lavoro abbastanza ampia che gli permette di raggiungere agevolmente anche le posizioni più lontane, rendendo quindi più flessibile la sua applicazione.

Si è inoltre visto come il controller del manipolatore supporti una porta per la connessione ethernet che permette di comunicare con un PLC.

Nel PLC vengono implementate tutti i controlli principali per permettere al manipolatore di funzionare in piena sicurezza, mentre nel robot sono inseriti i comandi di movimento che gli consentono di andare a svolgere il lavoro prefissato.

In questo elaborato sono stati riportati i controlli inerenti al solo movimento del robot senza andare quindi ad analizzare tutti i sistemi di sicurezza aggiuntivi implementati o il controllo di ulteriori dispositivi esterni come: sensori di rilevamento scatola nella zona di prelievo, sensori che segnalano se la baia di deposito è piena e gestione dei motori per il movimento di un nastro trasportatore che porta il pacco nella posizione di prelievo che verrebbero implementati lato PLC.

Andando quindi ad utilizzare il manipolatore e il PLC si è in grado di andare a scrivere un programma che sia in grado di andare a comandare il manipolatore stesso per funzionare come pallettizzatore.

Questa applicazione è solo una delle numerose possibili e, grazie allo sviluppo tecnologico e all'impegno dei programma-

tori, sarà possibile andare a compiere operazioni sempre più complesse.

BIBLIOGRAFIA

Giovanni Boschetti, "Appunti del corso di Robotica Industriale", Dispensa 2022.

Roberto Caracciolo, "Appunti del corso di Logica e Programmazione di macchine automatiche", Appunti 2023.

Bruno Siciliano Lorenzo Sciavicco Luigi Villani Giuseppe Oriolo, *Robotics: Modelling, Planning and Control*, Springer-Verlag London Limited, 2009, ISBN: 978-1-84628-641-4.

FANUC, CRX-5iA_10iA_10iAL_20iAL_25iA_Operator_Manual_[B-84194EN_03], 2022

FANUC, R-30iB_Plus_basic_operator_manual_alarm_code_[B-83284EN-1_08], 2021

Origine e storia dei robot: <https://it.wikipedia.org/wiki/Robot>
<https://www.trebi-bs.com/robot-industriali-storia/>

Robot collaborativi: <https://www.universal-robots.com/it/robot-collaborativi-o-cobot-cosa-sono-la-guida-definitiva/>
<https://www.internet4things.it/iot-library/cobot-cosa-sono-e-quali-sono-i-vantaggi-dei-robot-collaborativi/>

Gripper: <https://www.piab.com/it-it/promotion-pages/kenos/>

Storia dei PLC: <https://it.rs-online.com/web/content/discovery-blog/idee-suggerimenti/guida-plc-controllori-logici-programmabili>

Scheda tecnica Fanuc CRX-25iA: <https://www.fanuc.eu/it/it/robot/robot-filter-page/robot-collaborativi/crx-25ia>

