

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA

Scuola di Ingegneria e Architettura

Corso di Laurea in Ingegneria dell'Automazione

**Progettazione e sviluppo del software di controllo
di un sistema di trascinamento di matasse di filo
guainato**

Tesi di Laurea

in

Sistemi di Elaborazione e Controllo M

Candidato:
Lorenzo Galasso

Relatore:
Prof. Eugenio Faldella

Anno Accademico 2017 / 2018

*Alla mia famiglia,
per il sostegno donatomi durante il mio percorso*

*"La volontà ha una funzione direttiva e regolatrice;
rimette in equilibrio ed utilizza costruttivamente tutte le altre
attività ed energie senza reprimerne nessuna" Roberto Assagioli*

INDICE

INTRODUZIONE	XI
CAPITOLO 1: SISTEMI DI AUTOMAZIONE.....	13

1.1 SISTEMI DI PRODUZIONE AUTOMATIZZATI	13
1.1.2 CLASSIFICAZIONE DEI PROCESSI PRODUTTIVI.....	15
.....	15
1.2 CARATTERISTICHE DEI SISTEMI DI AUTOMAZIONE	18
1.3 CIM: COMPUTER INTEGRATE MANUFACTURING	21
1.3.1 DESCRIZIONE LIVELLI DEL MODELLO PIRAMIDALE DEL CIM	26
1.3.2 TIPI DI CONTROLLO NELLA CIM	31
CAPITOLO 2.....	33
DESCRIZIONE DELL'IMPIANTO	33
2.1 SISTEMI DI MOVIMENTAZIONE.....	33
2.2 DESCRIZIONE DELL'IMPIANTO	36
.....	43
2.2.2 SISTEMA DI TRASCINAMENTO	43
2.2.3 SISTEMA DI LEGATURA MATASSA.....	46
CAPITOLO 3.....	50
AMBIENTE DI PROGRAMMAZIONE: CODESYS	50
3.1 IL MODELLO SOFTWARE	51
3.1.2 I LINGUAGGI DI PROGRAMMAZIONE.....	54

3.1.3 CODESYS: STRUTTURA E PROGRAMMAZIONE	55
3.1.4 CODESYS: L'INTERFACCIA GRAFICA	60
3.1.4.1 L'EDITOR GRAFICO	61
CAPITOLO 4.....	67
PROGETTAZIONE SOFTWARE DI UN SISTEMA DI TRASCINAMENTO MATASSE.....	67
4.1 DESCRIZIONE DEL SISTEMA DI TRASCINAMENTO	68
4.2 DESCRIZIONE DEL FUNZIONAMENTO NOMINALE DEL SISTEMA	71
4.3 COMANDO DI ARRESTO	71
4.3.1	74
4.4.1 MAIN	77
4.4.3 GENERIC DEVICE	82
4.4.4 SIMULAZIONE DEL PLANT	87
4.4.5 COMUNICAZIONE TRA I VARI MODULI	89
CAPITOLO 5.....	92
CONCLUSIONI	92
ELENCO DELLE FIGURE.....	94
BIBLIOGRAFIA	97

RINGRAZIAMENTI

Ringrazio il *Prof. Faldella*, per avermi dato la possibilità di usufruire dei suoi insegnamenti su come approcciarsi alla modellazione del software per le macchine automatiche, per la sua pronta disponibilità a ricevermi al fine di chiarire ogni dubbio manifestato durante lo svolgimento dell'elaborato e per la sua grande pazienza, strumento necessario per raggiungere i propri obiettivi.

Un particolare ringraziamento al mio amico e ex- collega universitario Antonio Genovese che mi ha affiancato sin dall'inizio della tesi, offrendomi: consigli utili su come assimilare i concetti utilizzati nell'elaborato, tramite tanti esempi tangibili e chiari che possono verificarsi nella vita quotidiana; consigli sulla gestione del tempo in modo efficace tramite la stesura di una tabella di marcia quotidiana con degli obiettivi da raggiungere; la pazienza, la collaborazione e la determinazione nel raggiungere il risultato prefissatoci all'inizio dei nostri incontri serali nella modellazione del progetto software; per il sostegno morale e per l'affetto che ti può donare solo una persona con il cuore grande.

Ringrazio per il sostegno economico che mi ha dato la mia famiglia, mio padre Massimino, senza il quale non avrei potuto affrontare il corso di studi; ma soprattutto per l'affetto e la fiducia concessa anche da mia sorella Agnese, fondamentale soprattutto nei momenti di grande stanchezza prima di raggiungere il traguardo.

Un grazie particolare alla mia cara amica Eliana, per la sua disponibilità e pazienza infinita ad ascoltare i miei sfoghi, sempre pronta al confronto e a illuminarmi grazie ai suoi consigli utili e pratici per affrontare in modo delicato, le gioie e i dolori degli eventi che si presentano nella vita.

Un grazie ai miei amici Eliseo, Francesco, Valentina, Chiara, Marco, Toni, Alessandra, Antonio, i miei cugini Mirko e Katia che mi hanno accompagnato durante tutto il percorso di studi e al grande affetto scambiati che rende la nostra relazione stabile e indissolubile nel tempo.

INTRODUZIONE

L'obiettivo della tesi è di implementare un'architettura software per controllare in modo automatico il posizionamento di una matassa di filo guinato e coordinare il funzionamento delle macchine dell'impianto con l'obiettivo di realizzare a partire da un avvolto di filo la legatura dello stesso.

Il progetto oltre a realizzare la logica di controllo del sistema di trascinamento della matassa, si estende nella simulazione e visualizzazione dell'intero sistema tramite interfaccia grafica integrata nell'ambiente di sviluppo utilizzato. Inoltre, è stato gestito il comportamento del sistema, simulando le possibili anomalie che possono verificarsi durante il funzionamento dei dispositivi dell'impianto e durante l'ARRESTO DI EMERGENZA.

Nel capitolo 1 viene data una definizione e una classificazione dei processi produttivi presenti nel mondo industriale al fine di individuare la famiglia di appartenenza del processo realizzato dai dispositivi dell'impianto studiato. È stato presentato il modello CIM analizzando la funzione volta dai vari livelli e i relativi tipi di controllo, ponendo attenzione ai livelli e ai controlli d'interesse che sono stati trattati nell'elaborato.

Il capitolo 2 espone l'importanza dei sistemi di movimentazione nel ciclo di produzione e prosegue con la descrizione dei sotto-sistemi che costituiscono l'impianto proponendo oltre alle configurazioni presenti sul mercato anche possibili implementazioni che potrebbero essere adottate. Il capitolo 3 mostra una panoramica dell'ambiente di sviluppo del software per sistemi di automazione utilizzato, Codesys, presentando le applicazioni in cui viene utilizzato, il modello software standardizzato

dalla norma IEC, gli oggetti del ‘browser di progetto’ necessari per implementare un’applicazione per PLC, i programmi principali che realizzano un’applicazione di controllo e l’interfaccia grafica resa disponibile dal tool di sviluppo.

Il capitolo 4 presenta le caratteristiche dei dispositivi che compongono il sistema di trascinamento; descrive il funzionamento nominale e in caso di ARRESTO DI EMERGENZA o di FAULTS del dispositivo, utilizzando sia il modello grafico, ‘automi a stati finiti’, che la relativa traduzione nel codice eseguibile corrispondente, scritto in linguaggio ‘testo strutturato’. La trattazione si conclude mostrando l’interfaccia grafica dove in base ai comandi impartiti dall’utente viene visualizzata la relativa risposta del sistema.

CAPITOLO 1

SISTEMI DI AUTOMAZIONE

In questo capitolo viene data una definizione di processo produttivo descrivendo l'insieme delle operazioni che lo caratterizzano. Dopo aver presentato la classificazione dei sistemi produttivi, viene identificato il processo discreto a lotti come il tipo di processo che caratterizza il flusso operativo dell'impianto studiato. Inoltre, si procede a stilare una classifica dei sistemi di automazione e in particolare si descrivono i livelli della piramide del modello CIM, evidenziando gli strati d'interesse in questo lavoro di tesi.

1.1 SISTEMI DI PRODUZIONE AUTOMATIZZATI

Un *processo di produzione automatizzato* è composto da [1]:

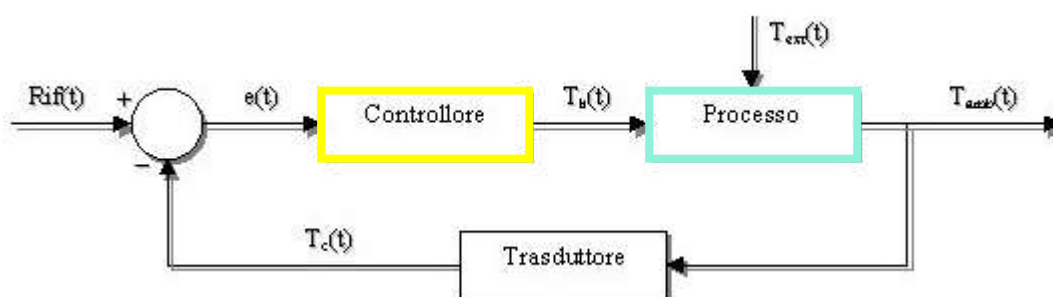


fig. 1 schema a blocchi di un sistema di controllo ad anello chiuso

- **un processo produttivo** costituito dalla combinazione di operazioni (movimentazioni, lavorazioni meccaniche) e trasformazioni fisico/chimiche che permettono di ottenere un prodotto finito da materie prime.

- **un sistema di controllo** (o controllore) rappresentato da un dispositivo (o un insieme di dispositivi interconnessi) che scambia informazioni e azioni con il processo produttivo per modificarne il comportamento (in modo desiderato) senza o con ridotto intervento umano.

Il sistema di controllo interagisce con il processo attraverso:

- dispositivi per acquisire *misure* dello stato dell'impianto. Tali misure sono rilevate da *sensori* di variabili fisiche (meccaniche, elettriche...) e *trasduttori* che convertono l'informazione veicolata dai sensori in un segnale di natura differente (tipicamente elettrica) seguiti da un condizionamento opportuno (per amplificare il livello, filtrare il rumore etc.);
- dispositivi che trasmettono *comandi* all'impianto. Tali sono rappresentati dagli *attuatori* che modificano il valore della variabili di controllo del processo per influenzarne lo stato. Gli attuatori sono preceuti da convertitori D/A la cui uscita viene amplificata fino alla necessaria potenza richiesta dagli attuatori;
- dispositivi di *elaborazione delle informazioni* (secondo opportuni algoritmi ad esempio PLC) e di *interfaccia* con operatori umani (ad esmpio HMI, Human Machine Interface).

Un processo produttivo è composto da una sequenza di operazioni elementari:

- di lavorazione: utilizzo dell'energia per alterare le proprietà dei materiali (trasformazioni)
- di assemblaggio: insieme di operazioni per unire più parti per formare un'unica entità
- di trasporto e stoccaggio: movimentazione e stoccaggio di parti e prodotti

- di test: verifica e collaudo del prodotto finale e delle sue funzionalità
- di coordinamento e controllo: coordinamento e regolazione delle operazioni anche a livello di gestione della produzione

In genere si hanno diversi tipi di processi produttivi come ad esempio, il processo di *produzione di energia* che trasforma le materie prime rappresentate da combustibile fossile e ossigeno in energia (prodotto finito), il *processo di assemblaggio* che consiste nel realizzare un prodotto finito tramite l'assemblaggio di diverse componenti elementari. Un *impianto* industriale è composto da macchinari, strutture e componenti di varia natura finalizzati alla realizzazione di un processo produttivo.

1.2 CLASSIFICAZIONE DEI PROCESSI PRODUTTIVI

Un processo produttivo, in base alla natura intrinseca del materiale che *trasformano*, si può classificare in [1]:

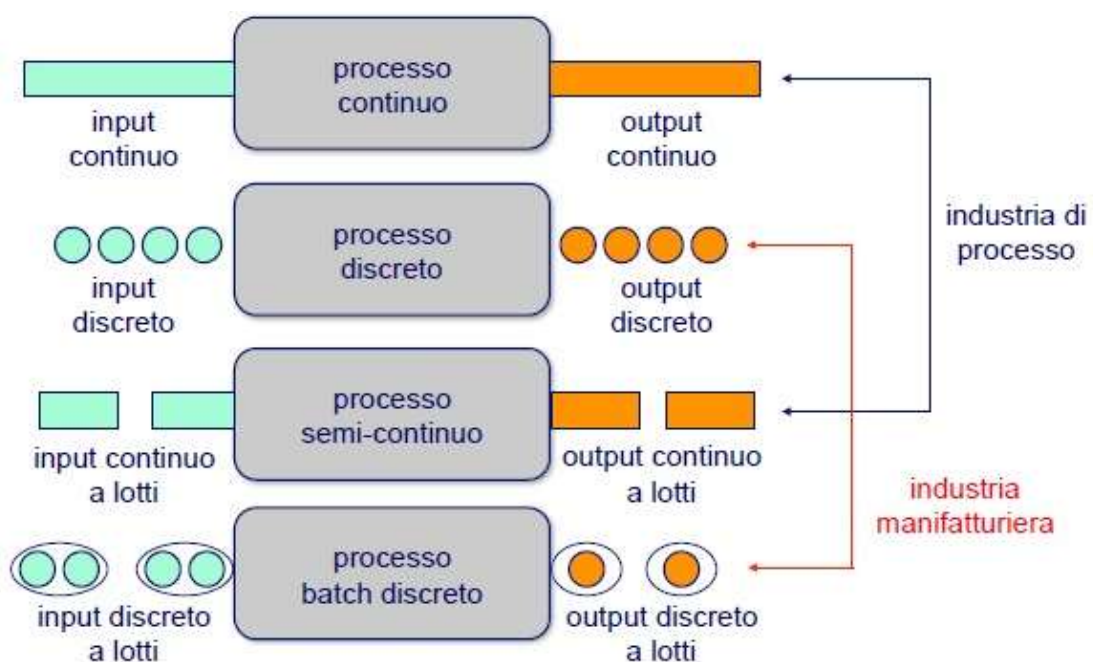


fig. 2 classificazione dei processi produttivi

Il **processo continuo**: coinvolge trasformazioni continue di massa, energia e quantità di moto su flussi continui di materiale.

Nel funzionamento a regime, l'obiettivo è di ottenere un prodotto dalla qualità uniforme nel tempo indipendentemente dalla durata del periodo in cui il processo è rimasto attivo.

Questo tipo di processo può durare giorni o settimane e viene interrotto solo in seguito ad operazioni di pulizia o manutenzione.

Tra le tipologie di impianti che realizzano il processo continuo si hanno: gli impianti per la produzione di energia, impianti per l'estrazione del petrolio, impianti per la produzione di vetro, ceramica, carta, cemento.

Tra le componenti principali si hanno: pompe, compressori, valvole, scambiatori di calore, pistoni e cilindri idraulici e pneumatici.

Possono riscontrarsi problemi di controllo del livello del liquido, della pressione, della portata o per le sequenze di avviamento/spegnimento dell'impianto (non si possono spegnere/avviare singole componenti) e nell'analisi della convenienza di spegnimento in caso di guasto.

Il **processo batch-discreto**: quantità finite di prodotto finale ottenute da quantità finite di materiali grezzi, processati secondo un insieme ordinato di attività e in un periodo di tempo finito.

I prodotti vengono lavorati in quantità di dimensioni non fisse a priori (lotti), ma *scalabili*. La sequenza di lavorazione (ricetta) viene fissata ma non la quantità di prodotto finale. Il processo è *interrotto* dopo la lavorazione di un lotto e è *ripreso* con quello successivo.

Le apparecchiature sono riutilizzabili per prodotti differenti ma con fase di pulizia intermedia.

I prodotti vengono realizzate secondo diverse formule ma in piccole quantità come prodotti farmaceutici, detersivi etc, prodotti che richiedono tempi di maturazione e fermentazione regolati come vino, birra e prodotti relativi ai settori della chimica industriale e alimentare (packaging).

I componenti principali sono: i sistemi di trasporto(pompe, valvole, collettori)e i sistemi di lavorazione(reattori, serbatoi, tubi, switch, allarmi,).

I problemi in questo tipo di processi riguardano, la definizione delle *ricette di lavorazione* (passi logici che descrivono il processo produttivo) con le relative operazioni sul sistema e la gestione dei lotti, più lotti di un'unica ricetta o più ricette possono essere contemporaneamente presenti nell'impianto.

Gli obiettivi del processo – batch sono garantire un uso corretto delle risorse dell'impianto, realizzare la ripetitività delle operazioni e aumentare la flessibilità di conduzione dell'impianto quando si introducono algoritmi sofisticati.

Il **processo semi-continuo** ha caratteristiche comuni sia con il processo continuo che con il processo a lotti.

Tale processo si presta per realizzare applicazioni specifiche e ripetitive come il filtraggio di gas/liquidi, trattamento delle acque, etc.

La caratteristica di tale processo è di separare componenti in un flusso continuo. Il componente separato viene accumulato in un'unità di processo fino a riempimento (processo continuo), a riempimento completato, il processo viene interrotto per svuotar e ripulire i serbatoi e successivamente il processo viene riavviato (processo - batch).

Il **processo discreto**: caratterizzato da cicli di lavorazione su singole parti o singole unità di prodotto. I materiali di partenza e i prodotti finali sono numerabili, basti pensare ai sistemi manifatturieri per lavorazione (tornitura, fresatura, etc).

Le tipologie di operazioni eseguite nei processi discreti sono: *lavorazioni* (meccaniche, per deformazione plastica, assemblaggio, saldatura, controllo qualità), *trasporto* (nastri, rulli, carrelli a guida automatica AGV, carri-ponte), *immagazzinamento* (buffer per accumulo di prodotti, grezzi, semi-lavorati, finiti, gestisce code in ingresso a una macchina e assorbe diverse velocità di lavorazione).

Il funzionamento del sistema oggetto di studio di questa tesi viene realizzato da un processo discreto caratterizzato dalle operazioni di assemblaggio e trasporto.

I controllori di processi industriali combinano in generale il controllo *logico* con il controllo *diretto* (analogico o digitale). Nel caso di un processo produttivo discreto, il *controllo logico* rappresenta il controllo delle sequenze di lavoro delle singole macchine dell'impianto, la supervisione con l'avviamento/spegnimento e la gestione dei guasti e delle emergenze, il *controllo diretto* viene espletato ad esempio da servo-azionamenti che realizzano il controllo ad anello chiuso della posizione, velocità, coppia degli attuatori (generalmente motori elettrici).

1.3 CARATTERISTICHE DEI SISTEMI DI AUTOMAZIONE

La scelta tecnologica del sistema di automazione da utilizzare per la produzione di un prodotto dipende dalle caratteristiche (chimiche, fisiche, strutturali, scomponibile o non scomponibile) e dalla collocazione nel mercato della domanda dello stesso.

Durante l'evoluzione tecnologica, nel mondo industriale si sono sviluppate tre tipi di sistemi di automazione [2]:

1. AUTOMAZIONE RIGIDA
2. AUTOMAZIONE PROGRAMMABILE

3. AUTOMAZIONE FLESSIBILE (FMS)

AUTOMAZIONE RIGIDA

Caratteristiche dei processi (industria di processo): produzione di un volume elevato di prodotti che presentano sostanzialmente le stesse caratteristiche tecnologiche. Il flusso delle operazioni sequenziali si ripete ciclicamente.

Architettura delle macchine: si tratta di macchine mono-scopo altamente specializzate per operare una sola tipologia di prodotto con ottimizzazione dei tempi di produzione.

Il sistema di controllo è deicato, realizzato tramite PLC o con tecniche cablate con modesta o nulla capacità di riprogrammare il flusso operativo per realizzare altri tipi di prodotti.

Un esempio di Automazione Rigida è rappresentato dall'industria automobilistica dove vengono impiegate *linee transfer*, costituita da numerose stazioni collegate rigidamente da un sistema di movimentazione, ad esempio nastro trasportatore, che eseguono delle lavorazioni sui motori e sui sistemi di trasmissione.

L'automazione rigida presenta due rischi: il primo è costituito dal pericolo che il volume di produzione realizzato sia inferiore a quello preventivato, questo comporta un aumento del costo unitario e il secondo è che terminato il ciclo di vita del prodotto, le linee di produzione risultano obsolete. Pertanto per prodotti con ciclo di vita breve, l'automazione rigida è rischiosa.

AUTOMAZIONE PROGRAMMABILE

Caratteristiche dei processi (industria manifatturiera): produzione di piccoli o medi lotti di produzione con caratteristiche variabili. Il flusso delle operazioni varia in base al tipo di lotto da realizzare.

Architettura delle macchine: i mezzi produttivi vengono progettati per adattarsi alle variazioni del prodotto. Questa caratteristica è ottenuta facendo lavorare le macchine sotto il "controllo" di un programma di istruzioni preparato per la realizzazione di un particolare lotto. Per passare da un lotto ad un altro è necessario fermare la macchina e caricare il programma che realizza il lotto desiderato.

Dal punto di vista economico, il costo dell'automazione programmabile può essere ripartito su un gran numero di prodotti anche se differenti. Grazie alle caratteristiche di programmabilità e adattabilità delle macchine si possono realizzare molti tipi di prodotti in piccoli lotti.

Il sistema di controllo è programmabile e/o adattativo.

AUTOMAZIONE FLESSIBILE (Flexible Manufacturing System)

Caratteristiche dei processi: la produzione è realizzata su richiesta di lotti di dimensioni variabili e diversificati. Sono richiesti elevati ritmi di produzione e le lavorazioni richiedono tecnologie sofisticate.

Architettura delle macchine: architettura altamente versatile in grado di operare su oggetti caratterizzate da differenti tecnologie.

Il sistema di controllo riprogrammabile durante il funzionamento dell'impianto consente di aumentare l'efficienza produttiva dell'impianto, tale versatilità ottenuta anche dalla capacità del calcolatore di controllo rappresenta il punto di forza dell'FMS rispetto agli altri sistemi di automazione.

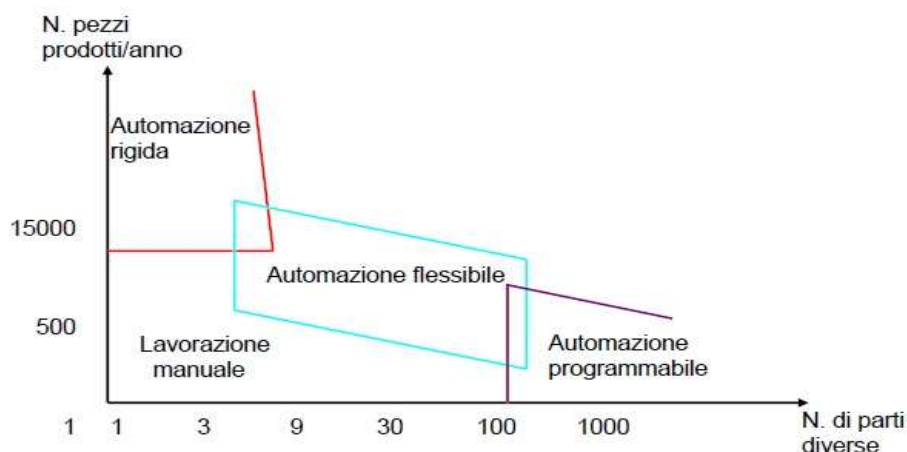


fig. 3 tipologie di sistemi di automazione [2]

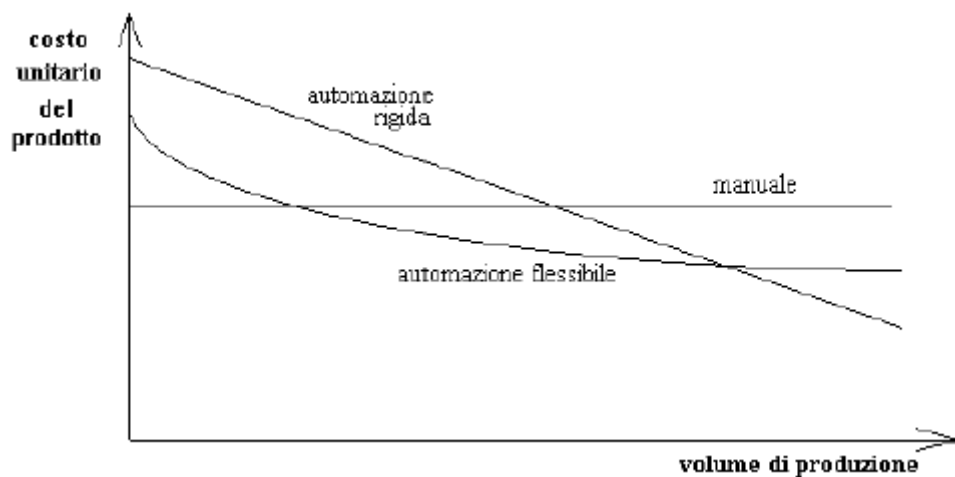


fig. 4 confronto dei sistemi di automazione industriale nel grafico c.u/ v [2]

Dalla figura 3 è possibile notare la posizione intermedia dell'Automazione Flessibile poiché è possibile riprogrammare la macchina per diverse configurazioni di prodotto ma la varietà delle configurazioni è limitata rispetto a quella che si ottiene nell' Automazione Programmabile.

Gli FMS consistono in una serie di stazioni di lavoro connesse da sistemi di manipolazione e accumulo materiale; un calcolatore coordina le varie attività che si svolgono nel sistema indirizzando opportunamente i prodotti alle stazioni controllando le operazioni programmate.

1.4 CIM: COMPUTER INTEGRATE MANUFACTURING

La fabbricazione meccanica, intesa come il processo che dall'idea di progettazione del prodotto, dalla domanda del mercato, da materiali grezzi conduce al prodotto finito può essere sintetizzata dalla figura 5

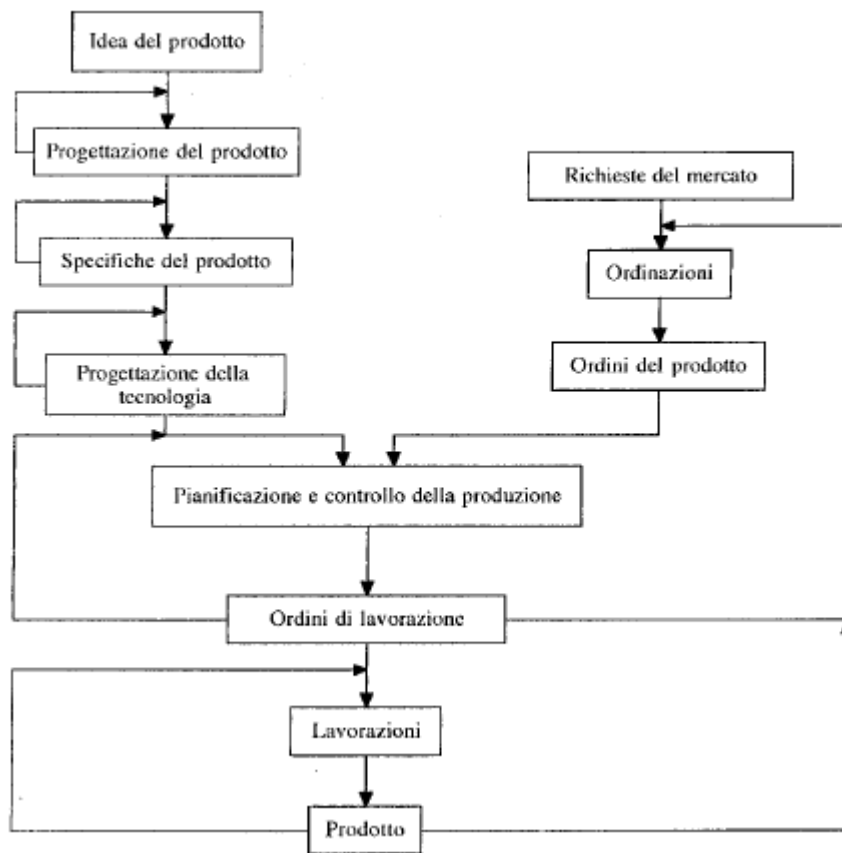


fig. 5 processo di produzione del prodotto [2]

Nell'analisi del processo di produzione si possono individuare alcune *fasi operative*, fasi in cui sono necessari svolgere una serie di calcoli e operazioni per ottimizzare il processo sia dal punto di vista della qualità del prodotto, sia dal punto di vista della quantità, in particolar modo, dei tempi di produzione.

Le fasi sono le seguenti:

- 1) Progettazione del prodotto: concezioni, calcoli, disegno di insieme, distinta base;
- 2) Progettazione della tecnologia: ciclo di lavorazione, metodo, attrezzi, programma macchina, tempo di lavorazione;
- 3) Pianificazione e controllo della produzione: verifica dei carichi di lavorazione, schedulazione, lancio e controllo della produzione;
- 4) Lavorazione: gestione magazzini, trasporti e movimentazione, assemblaggio, collaudo.

Queste fasi rappresentano l'insieme delle attività che non riguardano in senso stretto la produzione rappresentano l'insieme di attività secondarie (dette attività di supporto alla produzione) fondamentali per completare il sistema di produzione.



fig. 6 attività di supporto alla produzione [1]

Negli anni sono stati sviluppati software che automatizzano le attività di supporto alla produzione.

I software di supporto alle attività di business:

4. ERP (Enterprise Resource Planning): insieme di applicazioni volte all'automazione di attività di amministrazione, gestione della produzione e risorse umane;

I software di supporto alle attività di progettazione:

5. CAD (Computer Aide Design): insieme di software tools che assistono i progettisti nelle attività di progettazione;
6. CAE (Computer Aide Engineering): software tools per la verifica delle funzionalità del progetto (Es. AutoCad)

I software di supporto alle attività di planning e controllo:

7. CAM(Computer Aide Manufacturing): software che permette di automatizzare le prove di fattibilità del processo di produzione e organizzare la produzione stessa
8. CAPP(Computer Aide Process Planning): software che permette di automatizzare e ottimizzare la produzione pianificata.

Il CIM (Computer Integrate Manufacturing) rappresenta un modello teorico che consente l'integrazione informatica dell'automazione delle attività di supporto alla produzione e del processo di produzione.

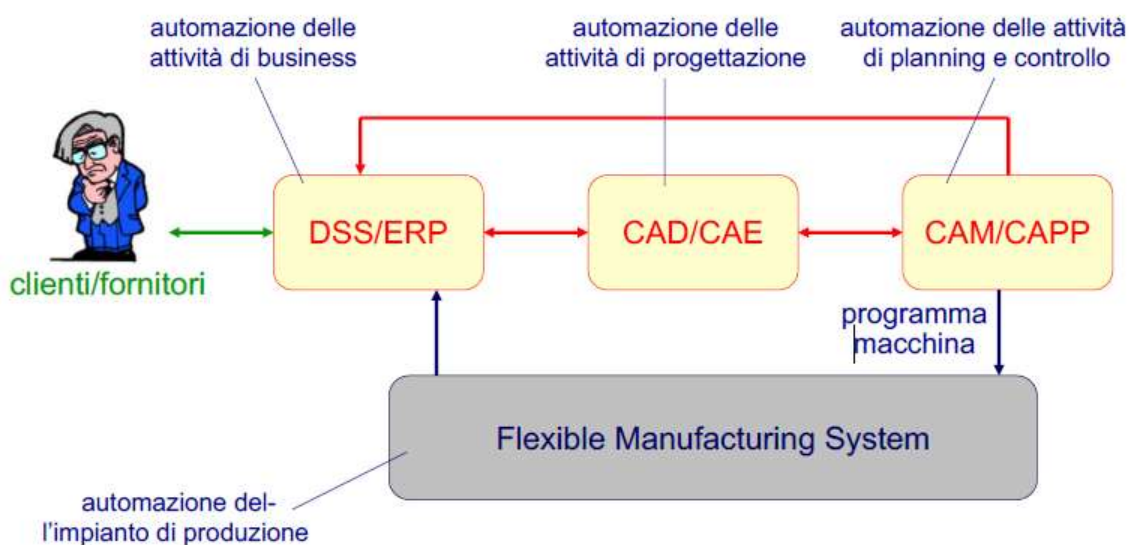


fig. 7 relazione tra CIM, FMS, e clienti/fornitori [1]

L'utilizzo di questo modello di produzione consente di avere una serie di vantaggi competitivi tra cui il miglioramento della qualità di produzione, riduzioni di tempo e costi, aumento della flessibilità della produzione, riduzione degli scarti, riduzione dell'impatto energetico e ambientale.

Il modello CIM è piramidale (fig. 8): suddiviso in cinque livelli, ciascun livello acquisisce informazioni dal livello sottostante, elabora strategie e invia i comandi elaborati al livello sottostante.

Il modello CIM ha una struttura modulare:



fig. 8 CIM: modello piramidale [1]

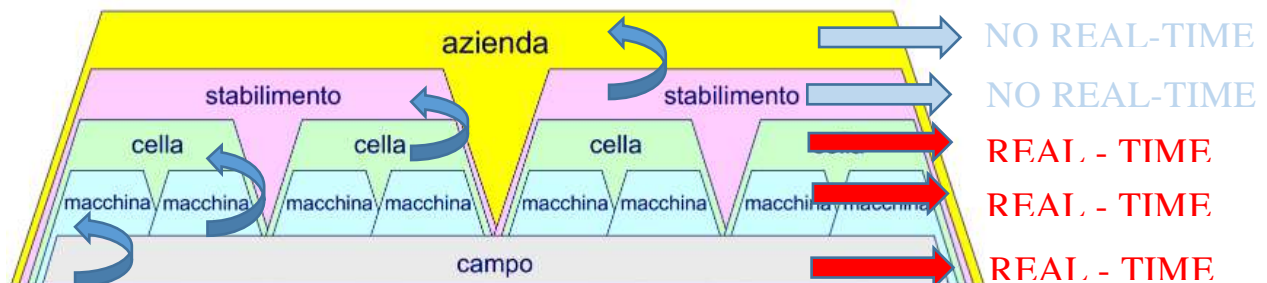


fig. 9 CIM: architettura modulare [1]

1.4.1 DESCRIZIONE LIVELLI DEL MODELLO PIRAMIDALE DEL CIM

Per i livelli di campo, macchina e cella serve hardware di controllo diretto o logico, deicato, con sistemi digitali a microprocessore e con vincoli real-time. Un algoritmo di controllo per essere real-time deve essere logicamente e temporalmente corretto. Nei sistemi real-time la presenza di vincoli temporali stringenti complica la corretta esecuzione dei task; è compito dello scheuler che in base a determinate politiche, seleziona in ogni istante di tempo quali task mandare in esecuzione tale da terminarli entro la deadline preventivata.

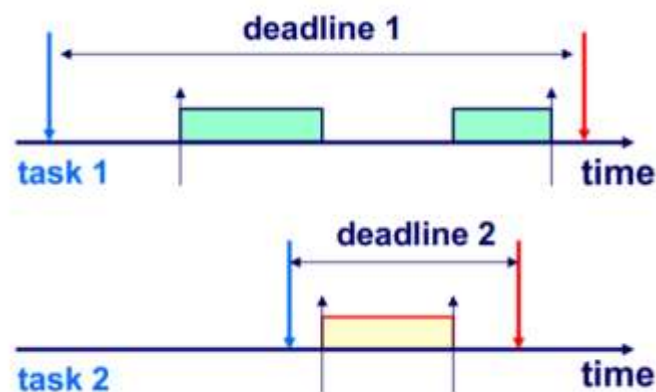


fig. 10 tasks real-time [1]

- Livello di campo: è il livello più basso della gerarchia e comprende i dispositivi che eseguono le attività di produzione e il loro controllo:

- sensori, attuatori;
- segnali i/o interfacciati con il livello superiore al livello fisico;

- sensori e attuatori dotati di "intelligenza", cioè in grado di pre-processare l'informazione, gestire l'interfaccia di comunicazione e di effettuare l'auto-diagnosi dei guasti;
- hardware di controllo dedicato, real-time con sistemi di controllo a microprocessore (sistemi embee)



fig. 11 dispositivi del livello "campo" [1]

- Livello macchina: insieme di elementi del livello di campo raggruppati nel livello superiore che forniscono determinate funzionalità:

- macchina utensile, robot
- gli elementi appartenenti a questo livello sono gestiti da un sistema di controllo che consente:
 - regolazione delle grandezze analogiche;
 - realizzazione delle operazioni sequenziali;
- il controllo a questo livello è necessario per pianificare il movimento che l'attuatore deve svolgere e per il coordinamento con gli altri eventuali attuatori presenti nel sistema
- hardware di controllo dedicato e real-time: PLC e sistemi embee.



fig. 12 dispositivi del livello "macchina" [1]

- Livello di cella: gli elementi del livello macchina vengono raggruppati al livello superiore formando le celle di produzione. Gli elementi che caratterizzano questo livello sono:

insieme di macchine interconnesse fisicamente da un sistema di locale di trasporto materiali e controllate in modo coordinato al fine di portare a termine un ben definito processo produttivo;

I sistemi di controllo supervisionano il funzionamento coordinato di tutte le macchine appartenenti alla stessa cella; le operazioni analoghe a quelle svolte a livello macchina ma più complesse;

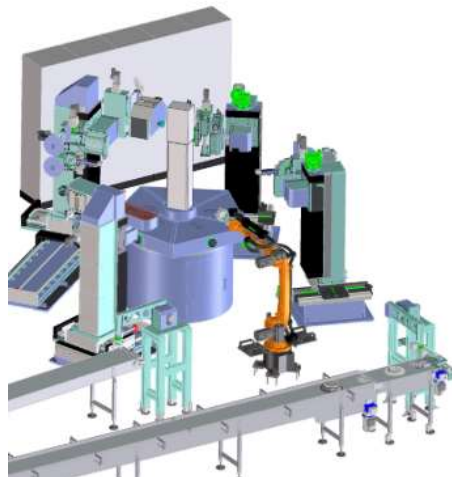


fig. 13 esempio di cella di produzione [1]

- Livello di stabilimento: racchiude tutte le celle o linee facenti parti di un impianto industriale; riceve le istruzioni dal livello gestionale (Es. gestione degli ordini) e le attua sotto forma di piani di produzione. Tale livello si caratterizza:

- il sistema di controllo è costituito da un sistema di supervisione, controllo e acquisizione dati (SCADA – Supervisory Control And Data Acquisition);
- workstation con struttura client – server;
- i requisiti di elaborazione real-time non sono stringenti

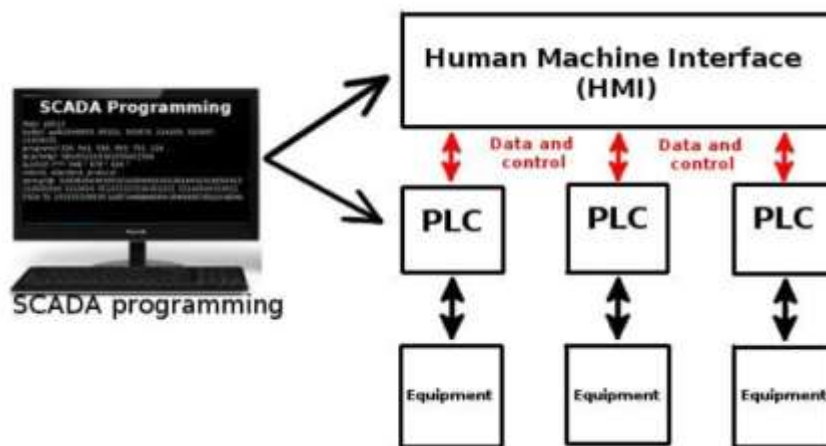


fig. 14 software del livello "stabilimento" [1]

- Livello di azienda: rappresenta il livello più alto della gerarchia dove nascono tutti i processi gestionali di supporto ai livelli inferiori. Questo livello è caratterizzato:

- non si parla di sistema di controllo ma sistema decisionale
- infrastruttura software basata su client – server
- non esiste il concetto di vincolo real-time;

I sistemi di controllo che realizzano i sistemi di automazione ai vari livelli del modello CIM rappresentano una struttura gerarchica costituita, andando dal basso verso l'alto da:

- controllo di campo
- controllo di procedure
- controllo di coordinamento

Nella figura sottostante è riportata secondo lo standard ANSI/ISA-S 88.01-1995 la corrispondenza del particolare tipo controllo con il livello/i nel quale lo stesso viene espletato.

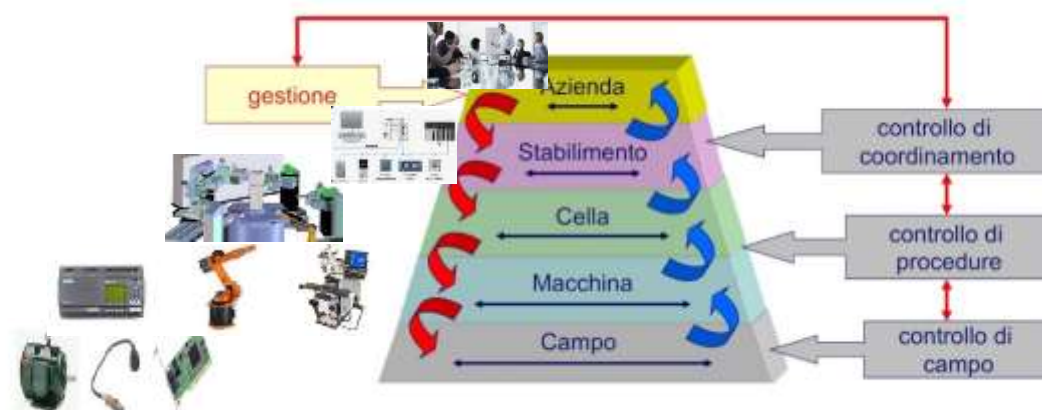


fig. 15 tipi di controllo per i livelli del modello CIM [1]

1.4.2 TIPI DI CONTROLLO NELLA CIM

Controllo di campo: comprende il sistema di controllo dei singoli componenti del campo; agisce su variabili continue e è implementato su dispositivi dedicati (Es. scheda per il controllo assi per applicazioni di motion) con vincoli real-time.

Controllo di proceure: si colloca tra i livelli macchina e cella della piramide CIM e riguarda il controllo di gruppi strutturati di componenti del livello di campo.

Il controllo è di due tipi:

- DIRETTO: viene applicato a livello macchina (es. tuning adattativo dei parametri del sistema);
- LOGICO: riguarda il coordinamento degli elementi di campo sulla base di operazioni sequenziali che caratterizzano il programma di lavorazione (es. macchina a stati che descrive il funzionamento logico-sequenziale del sistema);
- AUTO-DIAGNOSTICA e funzioni complesse di gestione automatica dei malfunzionamenti
- PLC o PC INDUSTRIALI si adattano bene per questo tipo di controllo
- VINCOLI REAL – TIME stringenti

Controllo di coordinamento: si pone in alto alla piramide del CIM e riguarda il coordinamento e la gestione delle varie celle di produzione. Questo controllo realizza la gestione dei sistemi di controllo di proceure tramite metodi implementati nel campo della "ricerca operativa". Tale controllo si caratterizza per la bassa frequenza di intervento e vincoli temporali molto estesi.

Il lavoro di tesi è basato sull'implementazione del software di controllo di un sistema simulato, quindi i livelli gerarchici della piramide CIM di interesse sono il livello macchina e il livello cella.

Avendo simulato l'impianto, il controllo di campo non è presente; il livello di controllo d'interesse è il controllo di procedure.

CAPITOLO 2

DESCRIZIONE DELL'IMPIANTO

In questa sezione viene evidenziata l'importanza dei sistemi di movimentazione all'interno del ciclo di produzione. Successivamente vengono presentati i sistemi di movimentazione che generalmente possono trovarsi all'interno di un sistema di automazione.

La trattazione procede tramite la descrizione delle macchine che costituiscono l'intero impianto studiato: il sistema di trascinamento, la macchina legatrice e la ribobinatrice. Si analizzano le varie soluzioni presenti sul mercato e delle ultime due si presenta una possibile configurazione che potrebbe essere adottata per l'applicazione studiata.

2.1 SISTEMI DI MOVIMENTAZIONE

Tra i fattori determinanti il successo della FMS (Flexible Manufacturing System) ci sono i sistemi di movimentazione automatizzati dei prodotti. Infatti la definizione di FMS recita:

"gruppo di macchine automatizzate, collegate da un sistema automatico di movimentazione e controllato da un calcolatore di processo atto a produrre una grande varietà di pezzi con una sequenza qualunque" [3].

La movimentazione automatizzata ha portato ad una riduzione del tempo della realizzazione del prodotto e un rapido adattamento del sistema alle variazioni delle quantità da produrre.

Come si nota dalla fig.16, il tempo impiegato per la lavorazione vera e propria rappresenta una percentuale minima rispetto al tempo totale per la pro-

duzione del prodotto stesso. È quindi chiaro che per aumentare la produttività è necessario ridurre sia il tempo necessario per trasportare il prodotto da una stazione a un'altra che per alimentare le stazioni dai rispettivi magazzini.

Quindi l'aumento della produttività dell'impianto dipende fortemente dall'efficienza dei sistemi di movimentazione che asservono le macchine dell'impianto stesso.

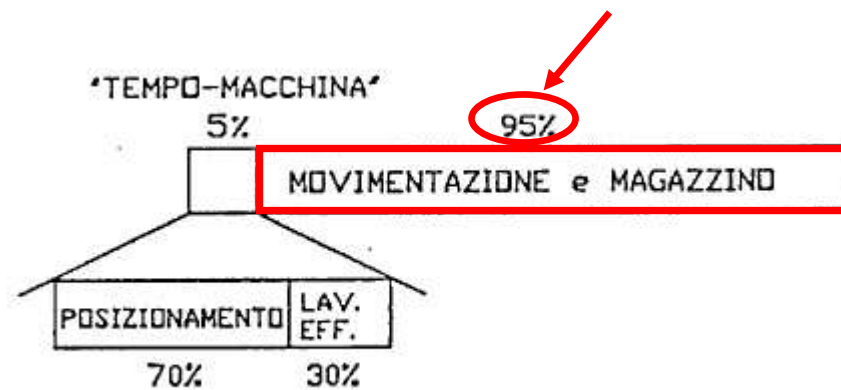


fig. 16 i tempi della produzione tradizionale [2]

I sistemi di movimentazione (o di asservimento) prodotto che si trovano nelle macchine automatiche sono rappresentati da:

- 1) ROBOT [4]



2) NASTRI TRASPORTATORI [4]



3) CARRELLI AGV [4]



Tra i vari compiti di trasporto eseguiti dal ROBOT si hanno:

- 4) pick-and-place del prodotto da movimentare tra due stazioni di lavorazione o tra il magazzino e una stazione di lavoro;
- 5) pallettizzazione (disposizione di oggetti preordinata su un opportuno supporto raccoglitore);
- 6) movimentazione e logistica dei pallet;
- 7) carico e scarico di magazzini;
- 8) carico e scarico di macchine operatrici e macchine utensili.



fig. 17 esempio di un sistema pick-and-place [4]

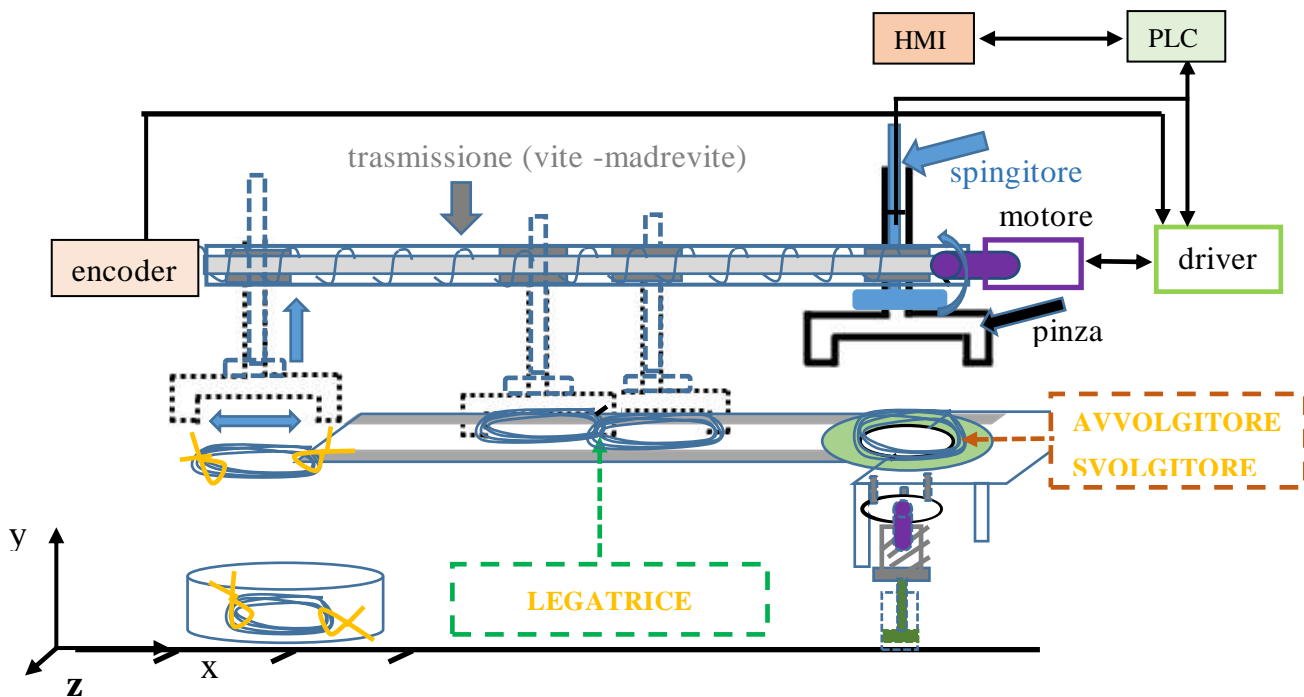


fig. 18 impianto: sistema di asservimento, avvolgitore-svolgitore, legatrice

2.2 DESCRIZIONE DELL'IMPIANTO

In questa sezione vengono presentati e contestualizzati i campi di applicazione dei sottosistemi che costituiscono l'impianto: la macchina "avvolgitrice-svolgitrice" (o sbobinatrice) di cavo guainato utilizzato nel cablaggio dei quadri elettrici, la "macchina legatrice" che si occupa di legare la matassa creata dalla sbobinatrice e il "sistema di trascinamento" (o asservimento) che movimentata la matassa in uscita dalla sbobinatrice, la serve alla legatrice che lega entrambi i lati e infine la matassa legata viene trasportata nell'apposito contenitore. In fig.18 viene data una rappresentazione di come potrebbe essere realizzato l'impianto evidenziando le interconnessioni tra i diversi componenti elettromeccanici (Es. motore -driver) e software di controllo (es.PLC).

2.2.1 MACCHINA AVVOLGITRICE - SVOLGITRICE

Per rispondere all'esigenza di ottenere, a partire da una matassa (madre) di materiale avvolto caratterizzata da un volume e massa considerevole, matasse di dimensioni ridotte e diversificate in base al tipo di materiale e al modo di utilizzo del cliente finale, sono state ideate le macchine avvolgitrici – svolgitrici.

Nella letteratura tale trasformazione prende il nome di "converting" e il sistema che la esegue è conosciuto come "ribobinatrice" (fig.19).

Tale esigenza nasce in diversi contesti di automazione industriale come l'industria della carta, l'industria metalmeccaniche, etc.

Tra i diversi tipi di materiale che si presta ad essere avvolto come la carta e i diversi tipi di filo, l'attenzione sarà incentrata sul filo guainato utilizzato nel cablaggio dei quadri elettrici per macchine automatiche.

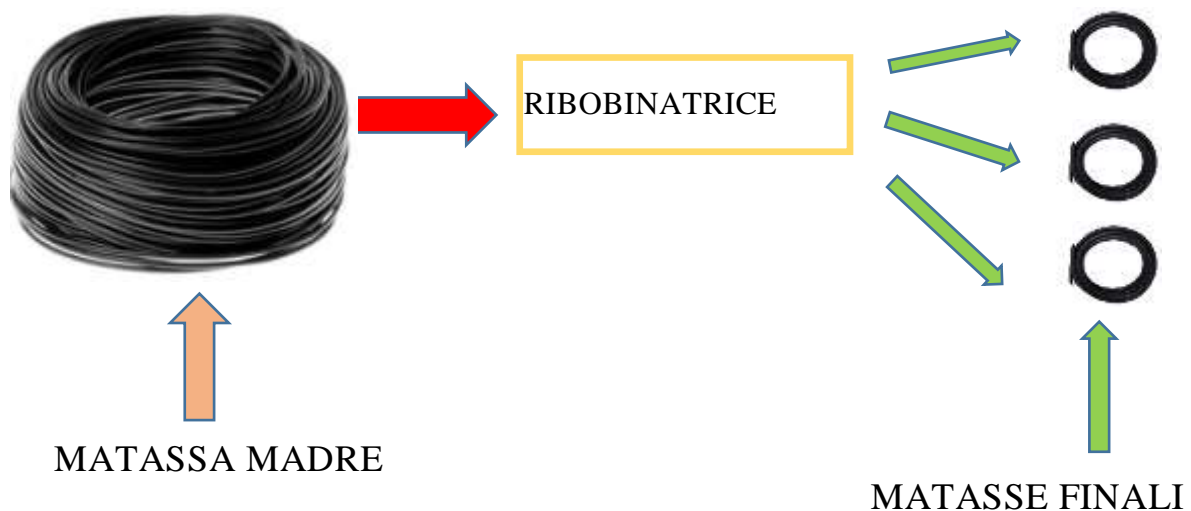


fig. 19 ribobinatrice (o avvolgitore-svolgitore) [9]

È possibile operare una distinzione tra i diversi tipi di macchine ribobinatrici che segue criteri differenti:



fig. 20 classificazione delle macchine ribobinatrici

SBOBINATORE MANUALE – AUTOMATICO

Le prime macchine introdotte sul mercato erano azionate manualmente, infatti la movimentazione del filo avveniva tramite l'azionamento manuale di una "manovella" posta sull'albero di rotazione che ruotando consente lo svolgimento del filo dalla "bobina madre" alla "matassa finale".

Il filo durante l'operazione di converting scorre su un piano dove è posizionata la taglierina che sarà azionata (sempre manualmente) quando la matassa raggiunge le dimensioni desiderate.

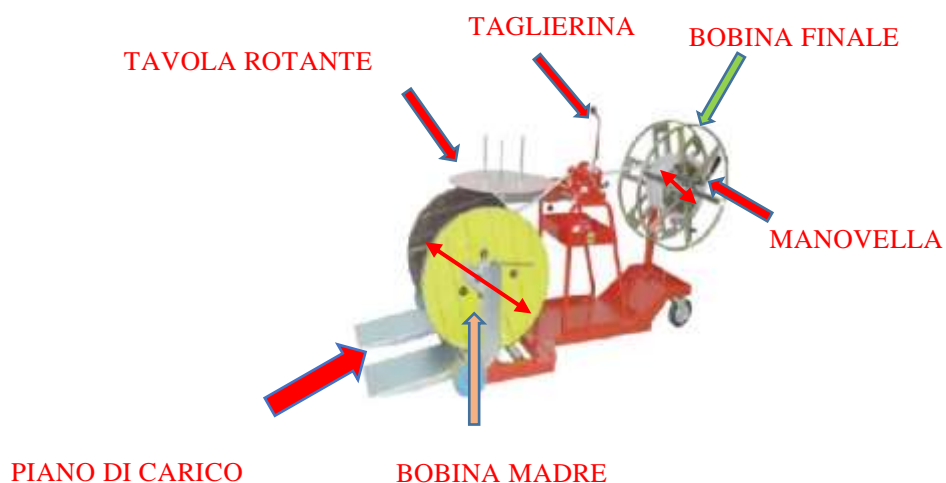


fig. 21 avvolgitore-svolgitore manuale [9]

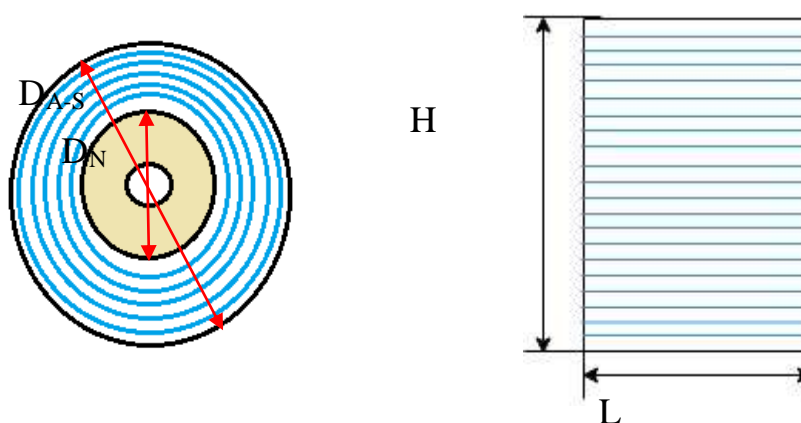
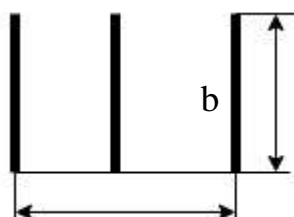


fig. 22 vista dall'alto e laterale di una bobina madre

La fig.22 rappresenta un avvolgitore – svolgitore manuale con diametro massimo (D_{A-S}) pari a 1200 mm. La taglierina consente il taglio di cavi con sezione fino a 20 mm. Le dimensioni massime dell'ingombro $H = 1700$ mm, $L = 750$ mm. Il peso della matassa da svolgere può raggiungere 116 Kg con un diametro del nocciolo variabile da 320 mm a 620 mm. Le dimensioni dello svolgitore vanno da 80 a 120 mm. La dimensione frontale dello svolgitore (a) può variare da 80 a 320 mm, mentre $b = 250$ mm.



*fig. 23 dimensioni dello
svolgitore*

Il diametro della tavola rotante è di 800 mm. La massa della bobina avvolta può raggiungere 100 Kg.

L'aumento della domanda di bobine di filo utilizzato nel cablaggio dei quadri elettrici delle macchine automatiche ha portato alla necessità di dover automatizzare l'operazione di riboninatura delle matasse.

Questo ha portato alla realizzazione della macchina completamente automatizzata tramite software di controllo caricato su plc in cui l'operatore può interfacciarsi tramite un pannello hmi.

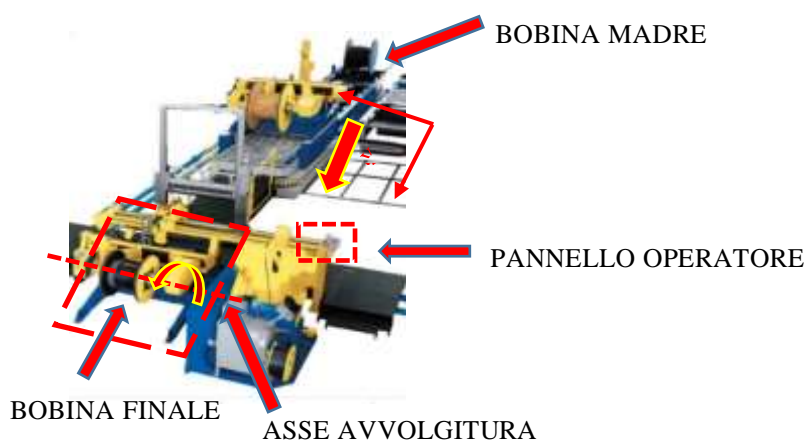


fig. 24 avvolgitore-svolgitore ad asse parallelo [10]

SBOBINATORE A RASTRELLIERA – A BOBINA CARICATA

L'operazione di converting può avvenire sia da matassa a bobina che da bobina a bobina o da matassa a matassa.

La bobina da srotolare può essere disposta a rastrelliera (fig. 25) su cui sono montati dei supporti che sostengono matassa di dimensioni e peso variabili.

Nel secondo caso la bobina madre ruota insieme alla tavola rotante su cui giace, dotata di aste verticali che svolgono la funzione del nocciolo (fig. 24).

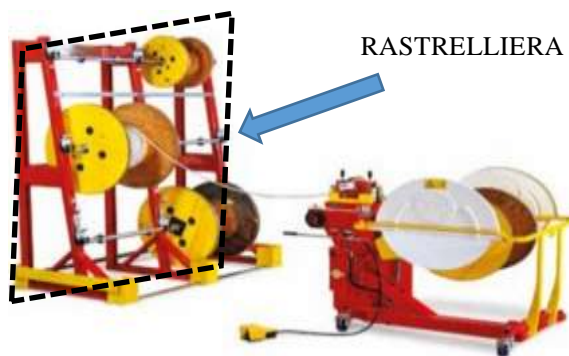


fig. 25 ribobinatrice a rastrelliera [8]

SBOBINATORE CON ASSE // AL PIANO DELLA MACCHINA – CON ASSE PERPENDICOLARE

Le dimensioni geometriche del luogo in cui deve essere collocata la macchina impongono il layout della stessa. In fig. 24 è possibile notare che l'asse di rotazione della "bobina formatrice" è parallelo rispetto al piano su cui giace la macchina.

In altri casi, come quello rappresentato in fig. 26, l'asse di rotazione dello svolgitoro giace su un piano perpendicolare al piano di appoggio della matassa.

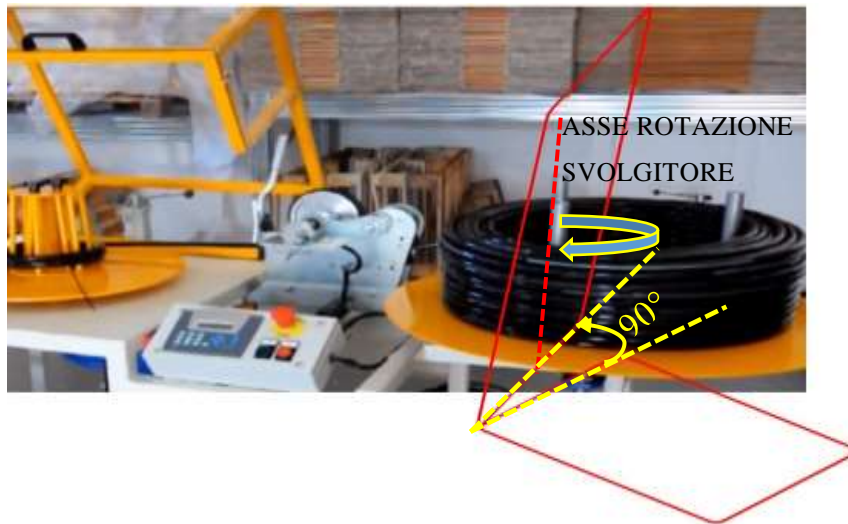


fig. 26 sbobinatrice ad asse perpendicolare [11]

In figura 27 viene mostrata la configurazione di una ribobinatrice che potrebbe essere adottata nell'impianto studiato.

La sequenza delle operazioni della macchina ribobinatrice può essere descritta come segue:

Dopo che il filo (la prima volta manualmente) è stato srotolato dalla matassa madre e viene adagiato sulla guida:

- La pinza scende e afferra il filo (1);
- la pinza srotola il filo lungo la guida (2) e rimane in attesa nella posizione "A" fino alla ricezione del segnale dal sistema di controllo che comunica la presenza e la disponibilità da parte della "tavola avvolgitrice";
- se la tavola avvolgitrice è presente, la pinza inserisce il filo nel gancio solidale con una delle tre aste della tavola (3);
- la pinza si apre e torna in 1 (4);
- tramite la rotazione dell'albero solidale alla tavola rotante il filo si avvolge attorno alle tre aste della tavola (5);

- quando la matassa ha raggiunto la lunghezza desiderata, il filo viene sezionato da una taglierina (7) dopo essere stato afferrato dalla pinza (6);
- la tavola avvolgitrice scende (8) tramite un pistone spingitore, liberando la zona d'ingombro e rendendo la matassa disponibile al sistema di trascinamento.

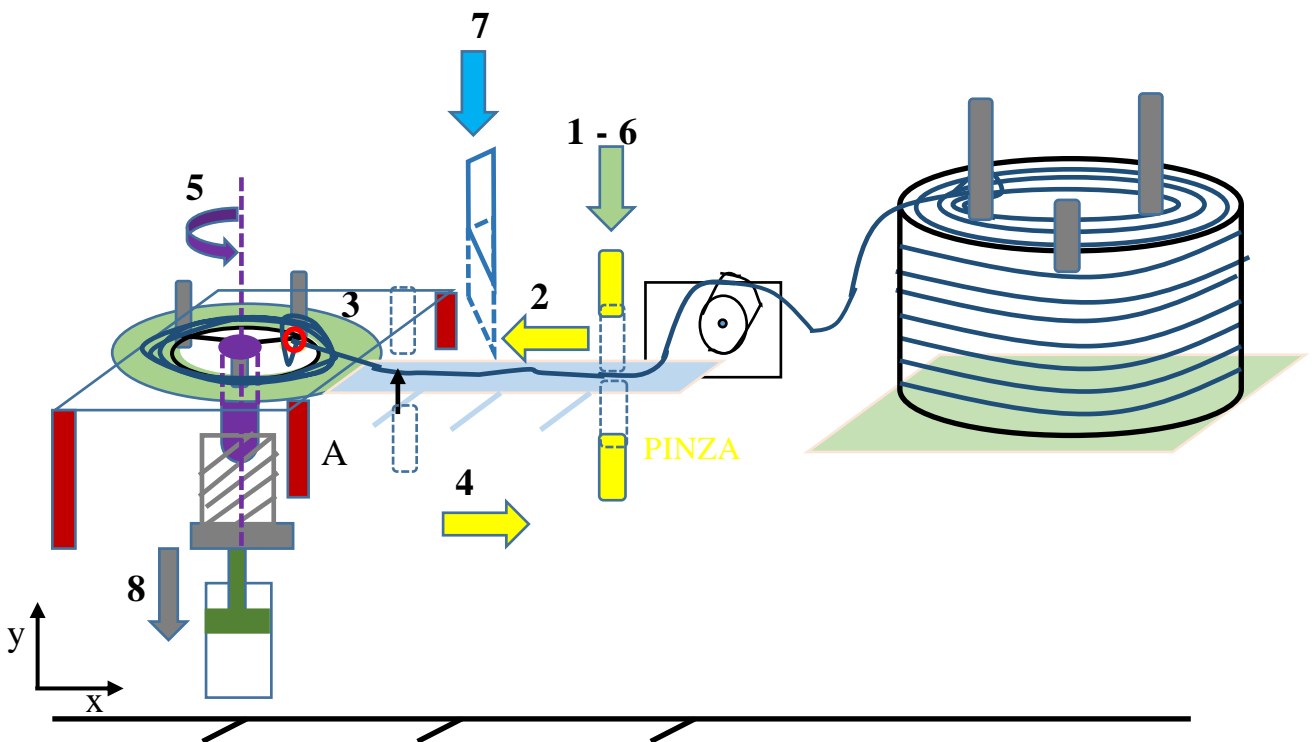


fig. 27 fasi di lavoro della macchina "ribobinatrice"

2.2.2 SISTEMA DI TRASCINAMENTO

Il sistema di automatizzazione coinvolto nel trasporto di un prodotto in fase di lavorazione, da una macchina a monte a una macchina a valle, è detto sistema di asservimento.

Nel campo dell'automazione industriale esistono svariati sistemi di asservimento.

Nella figura 28 è rappresentato un sistema di asservimento costituito da un robot antropomorfo che prelevando dei pezzi da un nastro trasportatore li unisce con i restanti di completamento che si trovano su una tavola rotante.

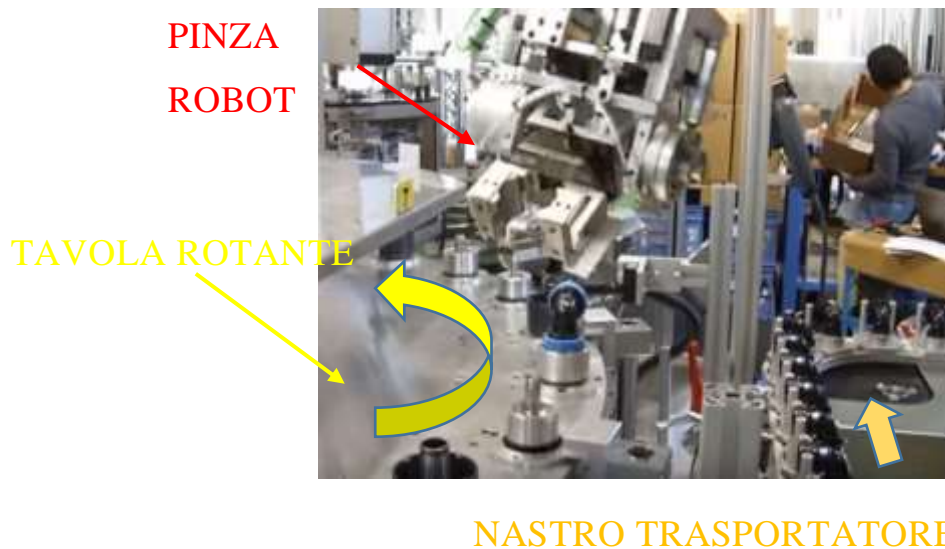


fig. 28 sistema di asservimento: nastro, robot e tavola rotante [12]

Nella figura 29 si nota che il nastro 2 serve i listelli di cartone al robot, che con le quattro pinze li preleva e li deposita all'interno dei rispettivi contenitori vuoti; tali vengono serviti in posizione di carico tramite un movimento alternato del nastro 1 su cui sono adagiati. Si hanno due forme di asservimento:

- nastro 2 *asserve* la pinza robot;
- la pinza robot *asserve* il nastro 1.

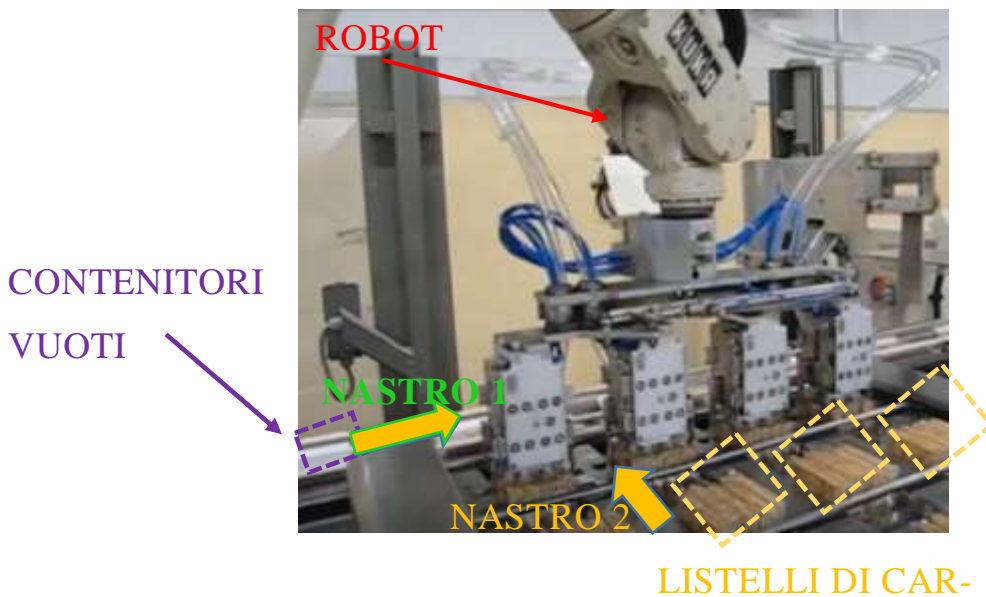


fig. 29 sistema di asservimento: nastro2, robot e nastro1 [13]

I vantaggi dell'asservimento automatizzato possono essere riassunti come segue:

- aumento della velocità di produzione: riduzione dei tempi e costi operativi;
- aumento della flessibilità lavorativa: cambio formato rapido;
- aumento della competitività: ampia fetta del mercato di settore;
- riduzione operazioni pesanti, pericolose e ripetitive per l'operatore: migliora la qualità del lavoro;

Nel progetto, il sistema di asservimento (fig.18), che trasporta la matassa nelle posizioni in cui viene effettuata la legatura fino alla posizione di scarico, è realizzato da una pinza pneumatica che afferra la matassa, uno

spintore che movimentata la pinza in direzione verticale (asse y). Lo spostamento nella direzione longitudinale (asse x) della pinza-spintore, avviene tramite la conversione del moto rotatorio generato da un servo-azionamento, nel moto traslatorio realizzato tramite un sistema di trasmissione vite-madrevite.

Il servo-azionamento garantisce il controllo in posizione del sistema di trascinamento.

La legatura della matassa viene resa possibile tramite il funzionamento sincronizzato del *sistema di trascinamento* con la *legatrice*.

Alternativamente al sistema di trascinamento sopra descritto sarebbe stato possibile utilizzare un sistema di asservimento pick-and-place che serve la matassa a un nastro servo-controllato.

Questo sistema oltre ad essere più costoso (dovuto al nastro servo-controllato necessario per il posizionamento della matassa nelle posizioni di legatura e scarico) risulta efficiente perché non sarebbe assicurata una presa "sicura" considerando la matassa non avvolta.

2.2.3 SISTEMA DI LEGATURA MATASSA

Di seguito viene riportata la sequenza operativa che caratterizza la configurazione della macchina ideata per la legatura della matassa di filo guainato.

Il filo utilizzato per la legatura ha una consistenza simile al filo di ferro plastificato usato per chiudere i sacchetti alimentari.

Le tre macro-operazioni sono:

- *stesura del filo e sezionamento*
- *formatura* del filo usato per legatura

- *legatura*

STESURA E SEZIONAMENTO DEL FILO

Per garantire la presa del filo che srotolato sulla guida (A-B-C) verrà sezionato da una taglierina, è necessario che la *pinza* afferra il filo.

La sequenza di operazioni di questa fase si può descrivere:

- la pinza si porta in posizione "a" e afferra il filo (1);
- la pinza traslando lungo l'asse x ha srotolato il filo per una lunghezza superiore a quella della guida;
- le linguette si chiudono garantendo la presa del filo, la pinza si apre (2), torna nella posizione iniziale (3);
- quando la pinza è in posizione "a", si chiude (4);
- la taglierina scende e seziona il filo; dopo il taglio la taglierina torna nella posizione iniziale (5).

- FORMATURA DEL FILO E LEGATURA

Dopo aver ottenuto, la dimensione desiderata, il filo deve essere *piegato* fino a formare una forma a "ferro di cavallo" ottenendo i due lembi necessari per realizzare la legatura.

La sequenza di fasi dell'operazione di formatura si riassume:

- tramite dei giunti rotoidali le aste della guida possono compiere rotazioni intorno all'asse perpendicolare al giunto (asse uscente dal foglio);
- 6.1, ruota l'asta C di 90 ° in senso antiorario;
- 6.2, ruota l'asta A di 90 ° in senso orario;
- la guida (a forma di ferro di cavallo) viene movimentata da uno "spintore" per una corsa tale che i lembi del filo si trovino ad una quota di una decina di centimetri superiore alla quota del piano che contiene la matassa;

- dopo che i due lembi siano stati grippati da due pinze (una per ogni lembo, fig.31), le linguette possono aprirsi. A questo punto la guida, tramite lo spintore su cui è appoggiata la riporta nella posizione iniziale e le linguette si aprono.
- le pinze ruotando una con senso opposto all'altra realizzano la legatura (fig.32).

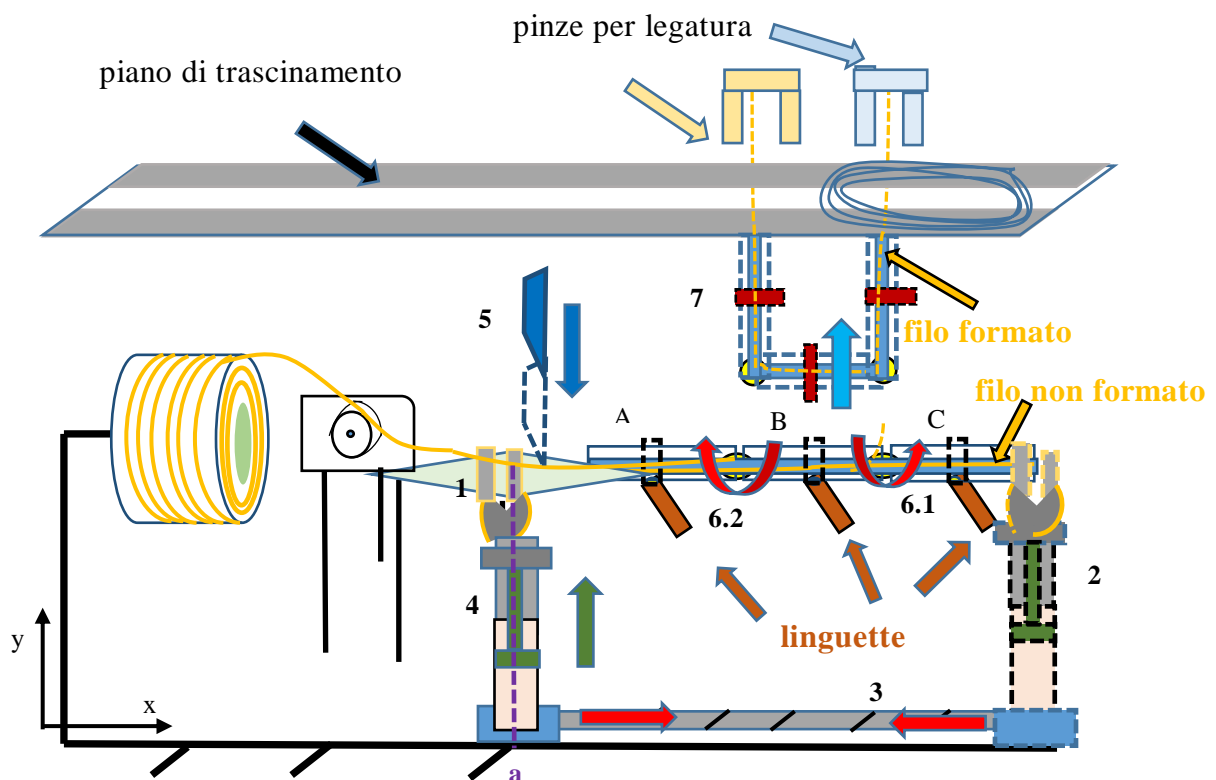


fig. 30 fasi di lavoro della macchina "legatrice"

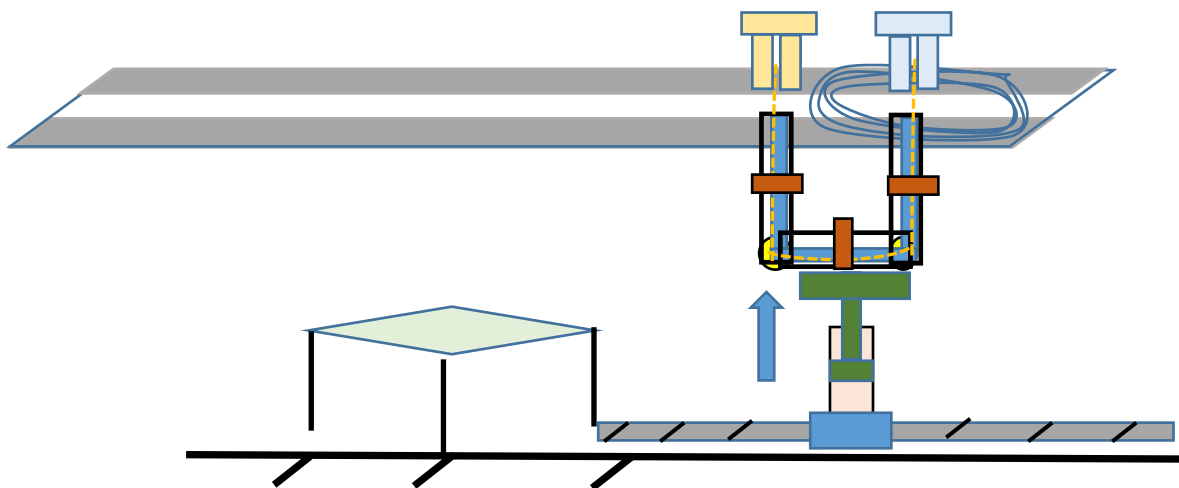


fig. 31 spintore serve il filo formato alle pinze della "legatrice"

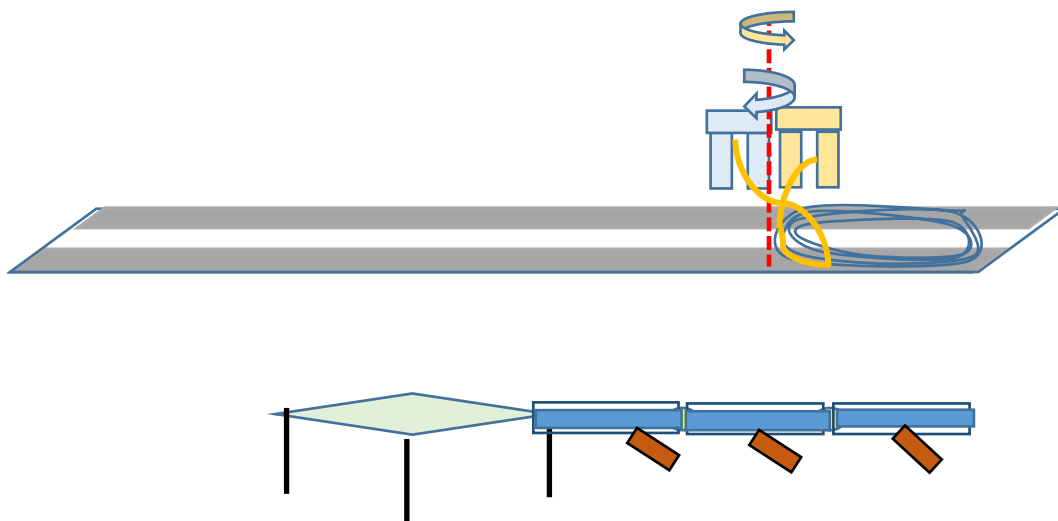


fig. 32 rotazione delle pinze per effettuare l'operazione di legatura

CAPITOLO 3

AMBIENTE DI PROGRAMMAZIONE: CODESYS

L'ambiente di sviluppo Codesys (Controlle Development System) realizzato dall'azienda 3S Software, permette di tradurre il modello "automi a stati finiti", utilizzato per descrivere la logica delle transizioni degli stati della macchina, in uno dei linguaggi definiti dalla norma IEC 61131-3.

La versione demo dell'ambiente di sviluppo è disponibile sul sito www.3s-software.com e presenta tutte le componenti della versione completa. Tuttavia, la versione demo presenta limitazioni, come il numero ridotto delle librerie fornite.

Il tool di sviluppo viene utilizzato per due tipologie di applicazioni [5]:

- 1) per programmare e eseguire su PC, controllori per macchine automatiche, per queste applicazioni Codesys funziona come compilatore dei programmi scritti nei linguaggi previsti dalla norma IEC 61131-3 e genera un programma eseguibile per l'ambiente real-time Codesys RTE;
- 2) per programmare sistemi di controllo industriali (come i PLC), per queste applicazioni il costruttore di PLC acquista il programma Codesys con il compilatore e l'interfaccia personalizzata per la comunicazione tra il PLC e il compilatore.

Dopo che il programmatore ha realizzato il codice di controllo, Codesys mette a disposizione una modalità di funzionamento "Simulation mode", che permette di eseguire il programma implementato senza avere, né l'ambiente real-time né il PLC.

È possibile creare un programma per simulare il funzionamento dell'impianto che interagendo tramite un'interfaccia di comunicazione con il software di controllo consente di verificare l'esecuzione del codice di controllo. Inoltre, Codesys consente di creare, un pannello operatore che consente l'interazione dell'utente con l'applicazione in esecuzione.

3.1 IL MODELLO SOFTWARE

L'IEC (International Electrotechnical Commission) ha elaborato un modello software per il sistema di controllo. Considerando le interazioni fra il programma e l'ambiente operativo del sistema di controllo si è giunti alla realizzazione di un modello software stratificato e gerarchico, in cui gli elementi appartenenti ad un livello possiedono anche le funzioni svolte dai livelli sottostanti e non viceversa.

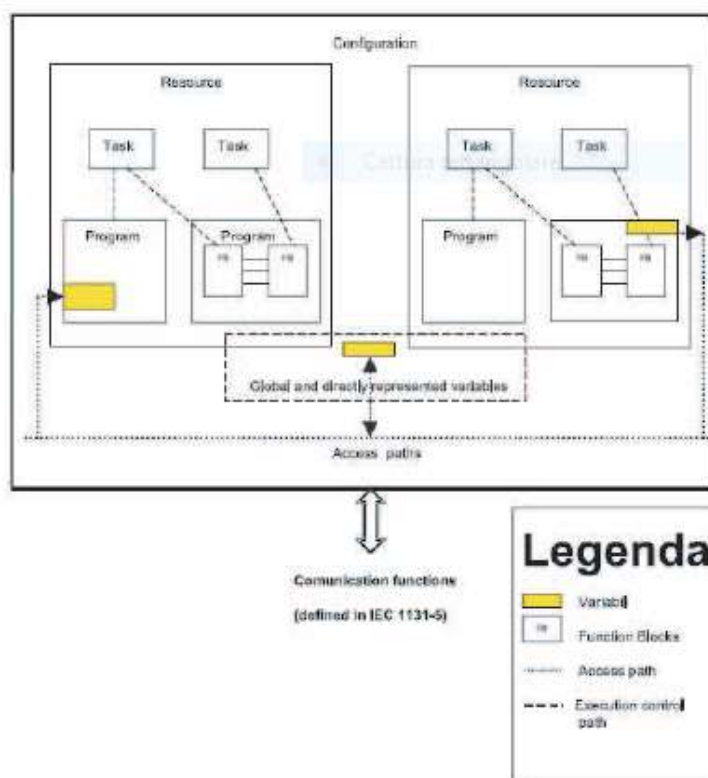


fig. 33 modello software [5]

Le varie componenti del modello (fig. 33) sono:

Configuration: è l'elemento che contiene tutti gli altri elementi e corrisponde, con l'intero sistema controllore programmabile, generalmente ad un PLC. In un sistema di controllo possono essere presenti più "configuration" che comunicano secondo le modalità definite dalla norma.

Resource: sono entità software che contengono i tasks, rappresenta un'interfaccia software fra il programma e gli i/o del PLC e tra il PLC e l'uomo, tramite terminali di programmazione.

Le Resource sono componenti del modello software autonome e comunicano tra loro, essendo definite a livello di configuration, tramite le Global Variables e alle Directly Represente Variables.

Program: lo standard IEC 61131-3 definisce un programma: "l'insieme logico di tutti gli elementi di programmazione e i costrutti necessari per la funzionalità di elaborazione dei segnali richiesta ad un sistema controllore programmabile. I Program contengono i costrutti con le istruzioni realmente eseguibili. L'esecuzione di ogni Program viene gestita da uno o più tasks.

Task: entità software che controlla l'esecuzione di uno o più programmi interi o una parte di essi. Il task può essere attivato ciclicamente o al verificarsi di un evento.

Local and Global Variables: identificativi delle variabili che indicano la locazione di memoria delle variabili dichiarate. Le variabili locali sono accessibili solo nelle unità in cui sono dichiarate, mentre le variabili globali sono accessibili nelle unità in cui sono dichiarate e in tutti gli oggetti in esse contenute.

Directly represented variable: variabili che si riferiscono a indirizzi di memoria del PLC.

Function Blocks: un Function Block è composto da due componenti:

- DATI: insieme di variabili utilizzate all'interno del Function Block (parametri d'ingresso, uscita, variabili interne);
- ALGORITMO: insieme delle istruzioni che vengono eseguite ogni volta che il Function Block viene richiamato.

A partire da un set di valori delle variabili d'ingresso l'algoritmo elabora dei valori delle variabili interne e di uscita, tali variabili possono essere locali o globali.

Oltre ai Function Block resi disponibili dalla libreria "Standard.LIB" (es temporizzatori, contatori), si possono creare i Function Block che eseguono il codice di controllo che risponde alle specifiche dell'applicazione desiderata. È utile avere i Function Block come gli "oggetti" utilizzati nella programmazione ad oggetti: infatti per utilizzare un Function Block è necessario: *creare* il Function Block, note le variabili d'ingresso, di uscita, di input-output, implementare l'algoritmo che l'oggetto deve eseguire; *istanziarlo* nel livello gerarchico più alto del modello software implementato; *chiamare* il Function Block. La chiamata è anticipata dall'assegnamento dei parametri alle variabili d'ingresso e viene seguita dalla scrittura sulle variabili di uscita.

Functions: sono moduli di programmi anch'essi riutilizzabili, che ricevono un set di valori in ingresso e producono solo un valore. La differenza tra le Functions e i Function Block è che i primi non hanno una struttura dati allocata in memoria quindi in corrispondenza della stessa configurazione d'ingresso forniscono sempre lo stesso risultato, cosa non vera nel secondo caso perché il risultato dipende dallo stato interno del blocco.

Access path: rappresenta l'ultimo delle componenti del modello software introdotto dallo standard. Poiché un sistema di controllo può essere costituito da diverse Configuration le quali devono scambiarsi informazioni è necessario gestire la comunicazione tramite opportuni protocolli.

Quindi Access Path sono dei percorsi di comunicazione fra le varie Configuration che avviene tramite delle variabili speciali dichiarate usando il costrutto VAR_ACCESS.

3.1.2 I LINGUAGGI DI PROGRAMMAZIONE

Per soddisfare tutte le necessità che possono emergere nello sviluppo di un programma di controllo, la norma IEC 61131-3 ha definito i seguenti linguaggi:

- Instruction List (IL): linguaggio di basso livello generalmente utilizzato per piccole applicazioni.
- Sequential Function Chart (SFC): è un linguaggio grafico molto espressivo. L'utilità di questo linguaggio consiste nel rappresentare il flusso di esecuzione del programma tramite una sequenza di stati consecutivi.
- Structure Text (ST): è un linguaggio testuale di alto livello, simile al C, creato appositamente per la programmazione automatizzata di processi. La potenzialità di questo linguaggio consiste nella possibilità che offre per implementare procedure complesse non facilmente esprimibili utilizzando linguaggi grafici.
- Ladder Diagram (LD): linguaggio grafico che implementa la logica booleana riferendosi agli schemi elettrici.
- Function Block Diagram (FBD): linguaggio grafico, basato sull'interpretazione del comportamento del sistema rappresentando il flusso dei segnali tra gli elementi del processo.

3.1.3 CODESYS: STRUTTURA E PROGRAMMAZIONE

Il tool di sviluppo utilizzato per implementare il codice di controllo, per simulare il plant e per realizzare l'interfaccia grafica uomo-macchina è Codesys V2.3. Questo ambiente di programmazione comprende:

- Il programma *Configuration* che serve per la configurazione dei server OPC e DDE necessaria per la comunicazione tra diversi PLC e tra PLC e applicazioni remote;
- L'interfaccia *ENI* che consente lo scambio di informazioni tra l'ambiente Codesys e un database esterno, in modo che i dati possono essere condivisi eventualmente da più server;
- Il sistema runtime *HMI* per le esecuzioni grafiche create in ambiente Codesys
- Il programma *Codesys SP RTE* utilizzabile per esecuzioni real-time

Il sistema runtime HMI e Codesys SP RTE non è disponibile nella versione demo come quella utilizzata per realizzare l'applicazione in questo lavoro di tesi.

In Codesys con il termine **Progetto** si intendono tutti gli "oggetti" necessari per realizzare l'applicazione per PLC. Gli oggetti sono (fig.36):

- **POU (Program Organization Unit)**: elenco delle unità di programma utilizzate, distinte in Program (PRG), Function (FC) e Function Block (FB). Ciascuna POU consiste in una parte, "intestazione", la dichiarazione delle variabili e una parte detta "corpo" che rappresenta l'algoritmo scritto in uno dei linguaggi previsti dalla norma (fig.36). Ai fini di realizzare un'applicazione real-time, requisito indispensabile per le applicazioni di automazione,

una POU può richiamare un'altra POU ma è non ammessa ricor-
sione.

- **Data Types:** elenco dei tipi di dato definiti dall'utente (es. Strut-
ture dati, tipi enumerativi).
- **Visualization:** elenco dei pannelli grafici di visualizzazione.
- **Resource:** elenco dei menù di configurazione degli elementi del
progetto (es. variabili globali, tasks di configurazione, PLC Tar-
get, etc).

Lanciando il programma Codesys, per creare un nuovo progetto è neces-
sario cliccare su *File*, selezionare *New* e quando appare *Target settings*,
selezionare "none" se desidera utilizzare l'ambiente di sviluppo per la
programmazione del software di controllo di un sistema simulato altri-
menti il nome del PLC sul quale caricare l'applicazione di controllo. A
questo punto compare una finestra come quella rappresenta in fig.34 che
indica il nome da dare al programma (PRG) e in quale linguaggio scri-
verlo.

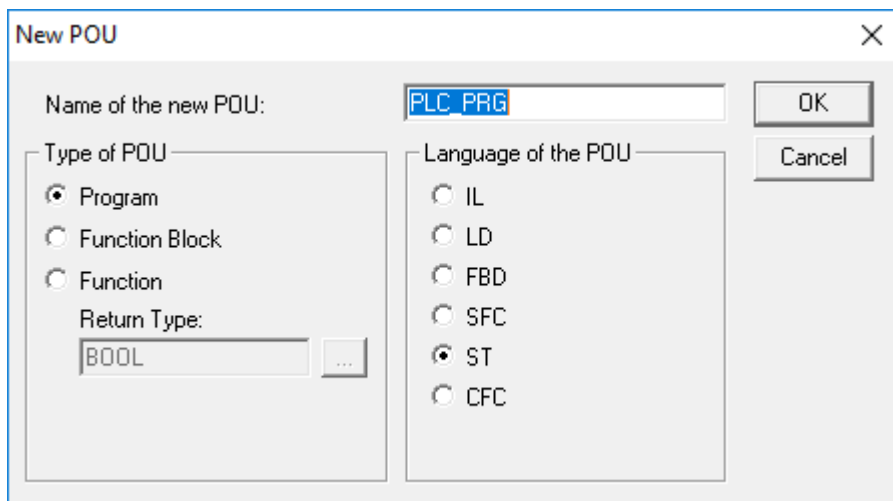


fig. 34 nuova POU

Per mandare in esecuzione il programma PLC_PRG è necessario creare un nuovo task, impostare la scelta del programma da eseguire e inserire il programma/i che il task manda in esecuzione (fig.35).

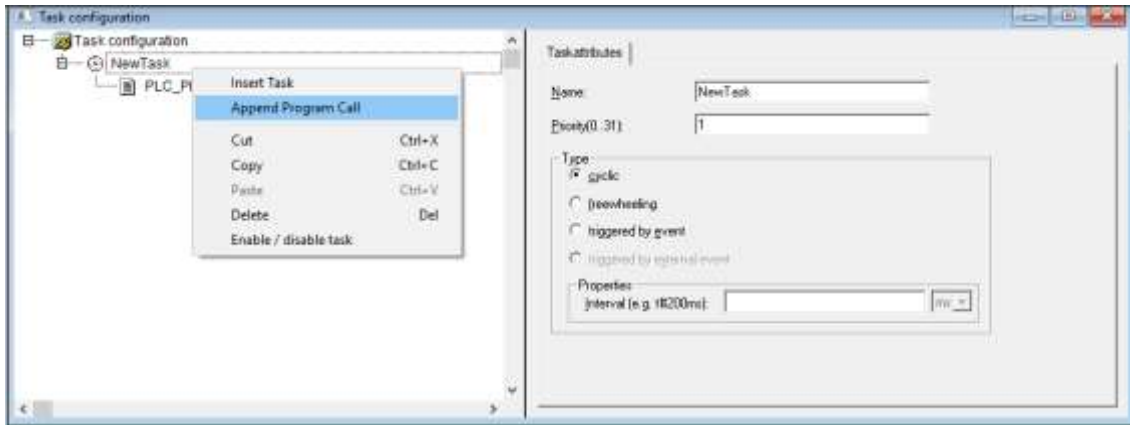


fig. 35 inserimento di un programma all'interno di un task

Una volta creato il progetto o aperto uno già esistente File → Open Codesys, si presenta una schermata come in fig.36. La schermata è divisa in due parti, la parte di destra rappresenta la finestra collegata con l'elemento selezionato nella parte sinistra.

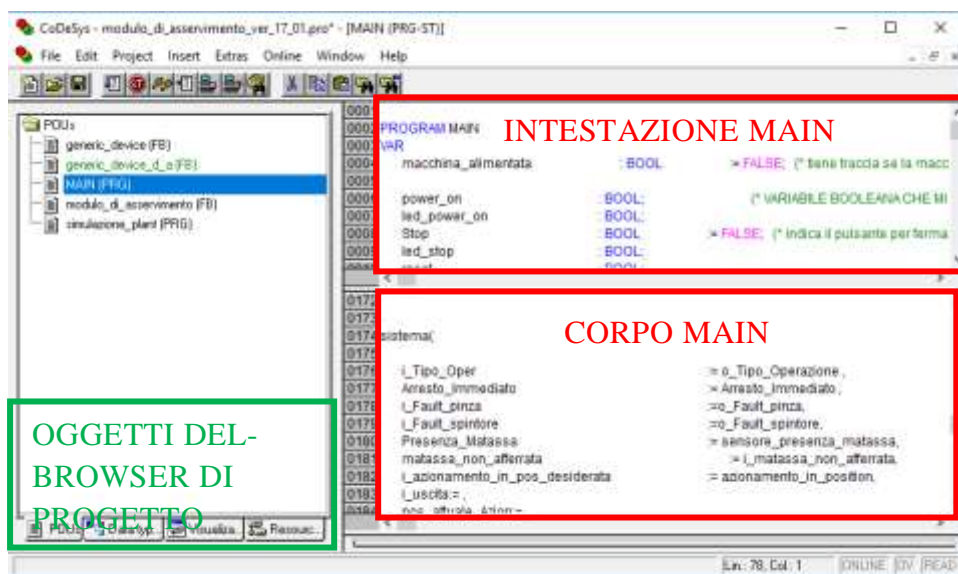


fig. 36 progetto

In basso a sinistra viene visualizzato il **Browser di Progetto**, cui è possibile navigare selezionando le sotto-schermate di cui è composto.

Per scrivere i programmi utilizzando i linguaggi previsti dalla norma suddetta è necessario cliccare la finestra **Resource** e dopo aver cliccato due volte si apre **Library Manager**, a questo punto con il tasto dx del mouse aggiungere la libreria Standard.LIB (fig.37).

Come si nota sotto Library Manager si è aperta una schermata che mostra l'elenco delle POU programmate nella libreria Standard.LIB che si possono utilizzare nel progetto (timer, contatori, etc). Inoltre, è possibile inserire altre librerie nel progetto, selezionando all'interno della cartella Additional Library scegliendo un file .LIB.

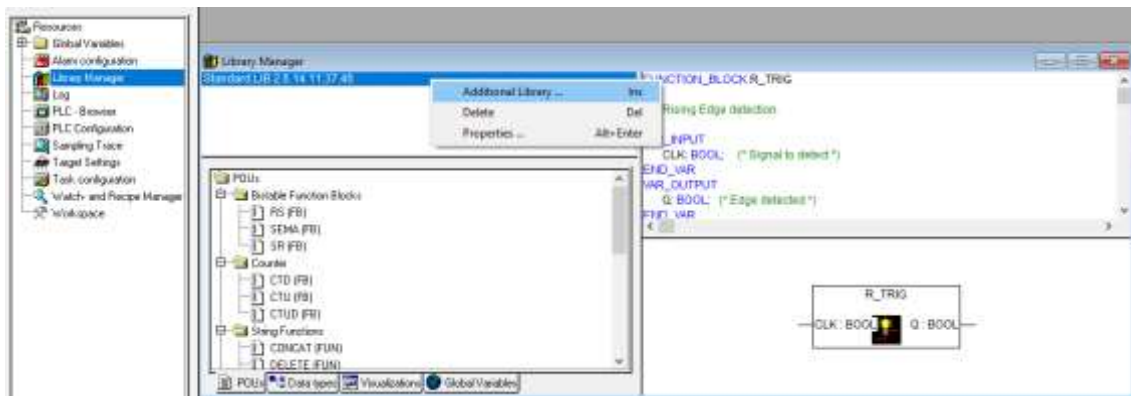


fig. 37 inserimento libreria: "Library Manage"

Il progetto per lo sviluppo di un sistema di controllo in ambiente Code-sys può essere pensato composto da due programmi principali (PRG): uno che realizza il controllo logico del sistema che si vuole controllare, CONTROLLO, l'altro programma simula l'impianto nella sua struttura, SIMULATORE (fig.38).

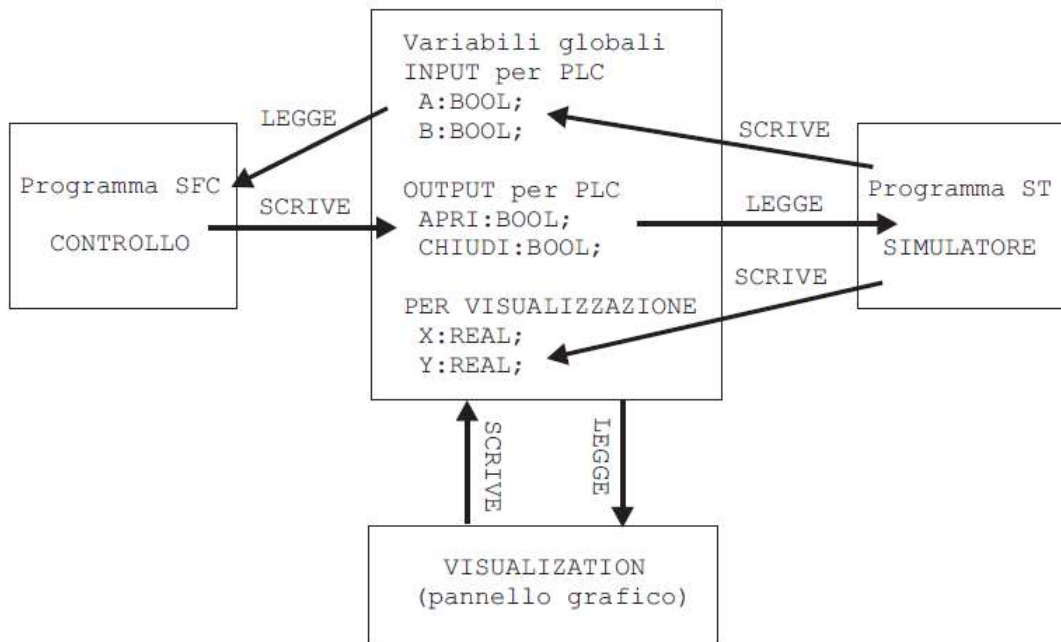


fig. 38 schema di funzionamento di Codesys [5]

Di solito il programma di controllo viene scritto in linguaggio SFC, mentre il programma di simulazione dell'impianto in linguaggio strutturato. I due programmi comunicano tra loro tramite gli ingressi e uscite del PLC.

Il programma di controllo legge gli ingressi del PLC (es. legge il valore dei sensori), esegue l'algoritmo implementato e scrive il valore delle variabili di uscita (es. invia i comandi agli attuatori) del controllore;

il programma di simulazione legge lo stato delle variabili scritte dal programma di controllo, comandi agli attuatori, e dopo aver eseguito il codice, modifica lo stato dei sensori e delle variabili utilizzate nel pannello grafico in modo coerente con il funzionamento dell'impianto.

I due programmi dovranno interagire tra loro, quindi le variabili utilizzate per leggere o scrivere gli ingressi e le uscite del sistema dovranno essere definite globali e definire le variabili locali quelle utilizzate all'interno di una POU.

Per eseguire entrambi i programmi è necessario inserirli in **Task Configuration** (fig.36) tramite il comando *Append program call* e selezionare i due programmi (PRG).

Il tool di sviluppo consente di simulare il progetto in tutte le sue componenti; per utilizzare Codesys in modalità di simulazione (cioè che simula il funzionamento del PLC) è necessario impostare l'opzione *Simulation Mode* dal menù *Online*, tale opzione viene impostata di default se viene impostato come target di progetto, *none*.

Terminata la stesura del codice di tutti i blocchi del progetto, tramite il comando *Build* (selezionabile dal menù *Project*), il sistema compila il codice e nel caso di errore si apre una finestra in cui viene visualizzato la descrizione e il numero delle righe di codice imputate.

Se la compilazione non riporta errori allora il codice può essere eseguito scegliendo dal menù *Online* il comando *Login* e successivamente il comando *Run*.

In questo modo il programma in esecuzione sul PC simula l'esecuzione del programma caricato su PLC.

3.1.4 CODESYS: L'INTERFACCIA GRAFICA

L'ambiente di sviluppo Codesys contiene un editor che consente all'utente di creare oggetti grafici collegati al programma al fine di realizzare un'interfaccia grafica per la simulazione dell'applicazione.

La visualizzazione può essere eseguita in quattro modi:

- Direttamente dal programma, quando si opera in *simulation mode* è possibile visualizzare direttamente l'interfaccia grafica collegata al controllore senza aver bisogno di altri tool.

- Tramite il programma Codesys HMI, che è il sistema run time necessario per la visualizzazione;
- Via web, Codesys genera una descrizione in formato XML che viene caricato nel controllore insieme ad un Java-Applet e che può essere visualizzato tramite protocollo TCP/IP su un browser.
- Per controllori con display integrato la visualizzazione dei dati può essere caricata insieme all'applicazione sul sistema, questa soluzione può essere semplicemente applicata su qualunque dispositivo programmabile in Codesys.

Tramite la Visualization, selezionabile nel browser di progetto, è possibile dare una rappresentazione grafica delle variabili di progetto, quindi collegare gli ingressi del PLC (sensori) alla tastiera e visualizzare le variabili di uscita (es. azioni di comando che generano il movimento). Le proprietà di ogni singolo elemento delle finestre di configurazione.

3.1.4.1 L'EDITOR GRAFICO

Per accedere all'editor grafico di Codesys è necessario cliccare la sotto-finestra **Visualization** presente nel browser di progetto. Per creare un nuovo oggetto è sufficiente cliccare con il tasto dx **Visualization** e selezionare *Add object*. Dopo aver inserito il nome del *object* creato, a dx compare un pannello in cui inserire le forme degli oggetti selezionabili nella barra degli strumenti (fig.39).

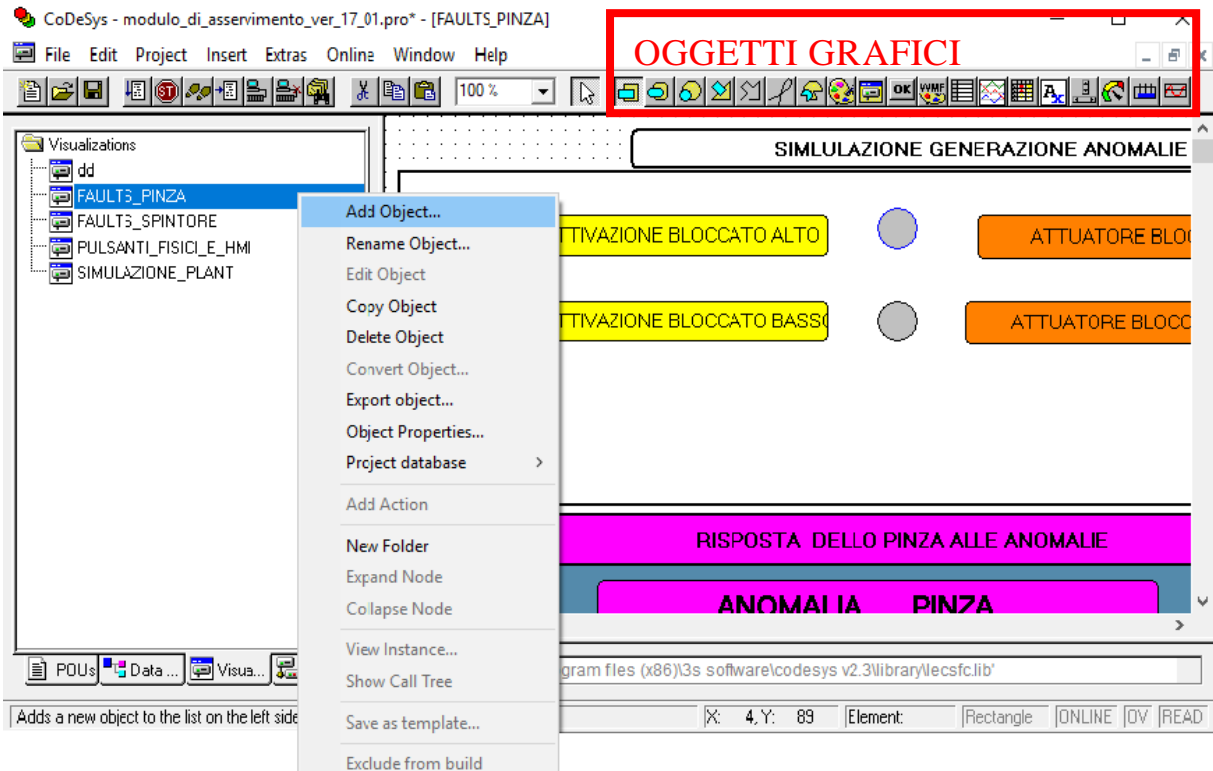


fig. 39 creazione di uno nuovo Object (Visualization)

Dal menù *Extras* selezionando il comando ‘Element list’ compare una finestra che rappresenta la lista degli elementi contenuti nell’object selezionato. Gli elementi sono numerati e nel caso risultano sovrapposti verrà mostrato in primo piano l’elemento identificato dal numero maggiore. Tuttavia, è possibile, utilizzando i comandi posti a lato della finestra (es. “To back”) cambiare il “piano” di visualizzazione degli elementi (fig. 40

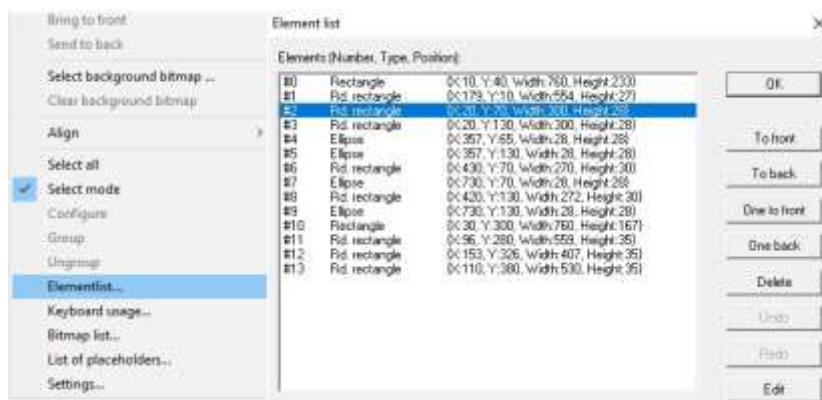


fig. 40 Element list

Per impostare le proprietà che caratterizzano ogni elemento dell'object creato, è necessario cliccare due volte sull'elemento, si apre una finestra in cui sotto la voce Category è possibile selezionare la categoria e impostare i parametri associati che l'elemento deve possedere coerentemente con la funzione espletata (fig.41).

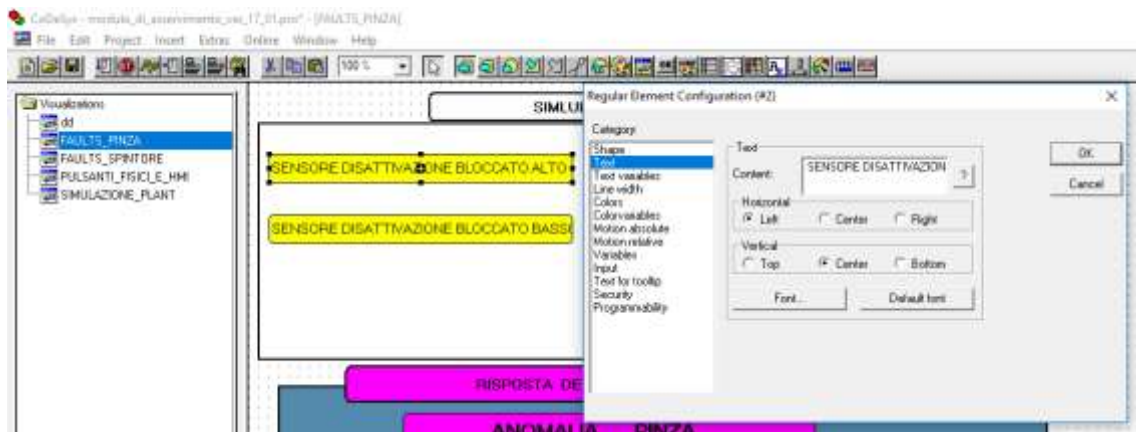


fig. 39 finestra di configurazione dell'elemento visualizzato (categoria 'Text')

Si descrivono alcune delle categorie necessarie per impostare le proprietà degli elementi utilizzati nella visualizzazione del progetto realizzato in questo lavoro di tesi.

Color: inserita la variabile booleana nel campo 'Change Color' della categoria *Variables*, l'elemento assumerà il colore impostato nel campo 'Color' della categoria *Colors* quando la variabile è false altrimenti verrà visualizzato il colore scelto nel campo 'Alarm Color'.

Motion absolute: il valore della variabile inserita in questa configurazione consente, nella modalità on line, lo spostamento dell'intero elemento lungo gli assi x-y (rispettivamente campo X- Offset e Y-Offset). Il valore della variabile inserito nel campo 'Scale' (fattore di scala) varierà la dimensione dell'elemento. Il campo 'Angle' determina una rotazione dell'elemento intorno al 'tuning point', rappresentato da un piccolo cerchio nero con una croce centrale.

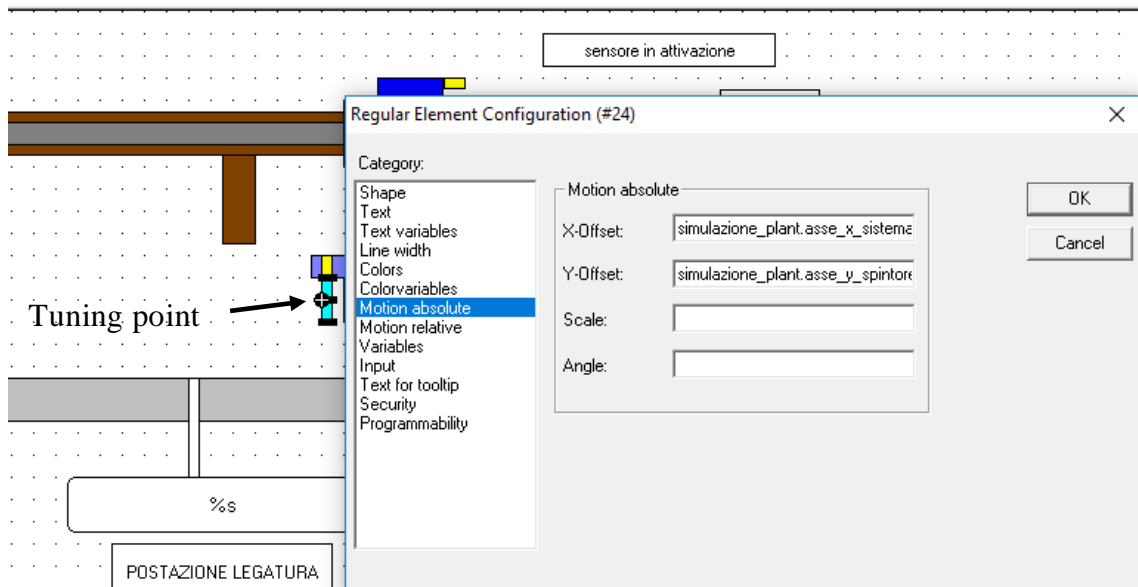


fig. 40 finestra configurazione dell'elemento visualizzato (categoria Motion Absolute)

Variables: le variabili allocate ai campi di questa categoria caratterizzano l'aspetto e le proprietà dell'elemento in modalità on line. Il campo 'invisible' consente di visualizzare l'elemento se il valore logico della variabile inserita nel campo è FALSE altrimenti l'elemento sarà invisibile;

'Disable Input' se la variabile inserita è TRUE allora tutte le impostazioni associate alla categoria 'Input' verranno ignorate; 'Textdisplay': se viene inserito un "%s" nel campo "Content" della categoria 'Text', allora

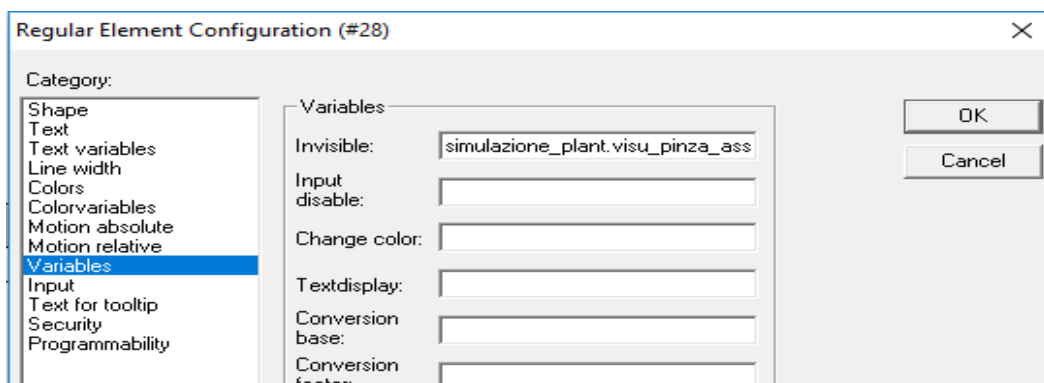


fig. 41 finestra configurazione dell'elemento visualizzato (categoria Variables)

il valore della variabile inserita nel campo "Textdisplay" verrà visualizzato, nella modalità on line al posto di %s.

Input: in questa categoria vengono inserite tutte le variabili il cui valore viene modificato, in on line mode cliccando sull'elemento. Il campo "Toggle Variable" realizza la funzione di un interruttore bistabile, consente di modificare il valore della variabile inserita, cliccando sull'elemento il valore passa da FALSE a TRUE e rimane a questo valore logico, finchè non sarà nuovamente cliccato. Il campo "Tap Variable" realizza invece la funzione di un interruttore monostabile, il valore della variabile booleana cambierà per un durata pari al tempo cui viene cliccato l'elemento, appena viene rilasciato il pulsante, viene ripristinato il valore iniziale. Di default il valore iniziale della variabile è FALSE se Tap False è disattiva, viceversa è TRUE.

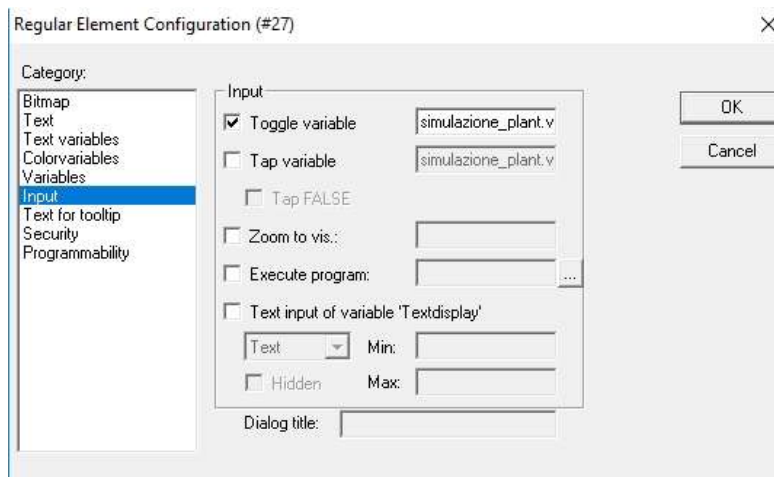


fig. 42 finestra configurazione dell'elemento selezionato (categoria 'Input')

Security: questa categoria è utile se chi crea l'elemento vuole garantire differenti modalità di accesso e visualizzazione ai diversi User Group definiti. Tramite il comando 'User Group Password' selezionabile dal menù 'Project' si accede ad una finestra cui è possibile creare un proprio user-group protetto da Password (Menù Project User Group Password)

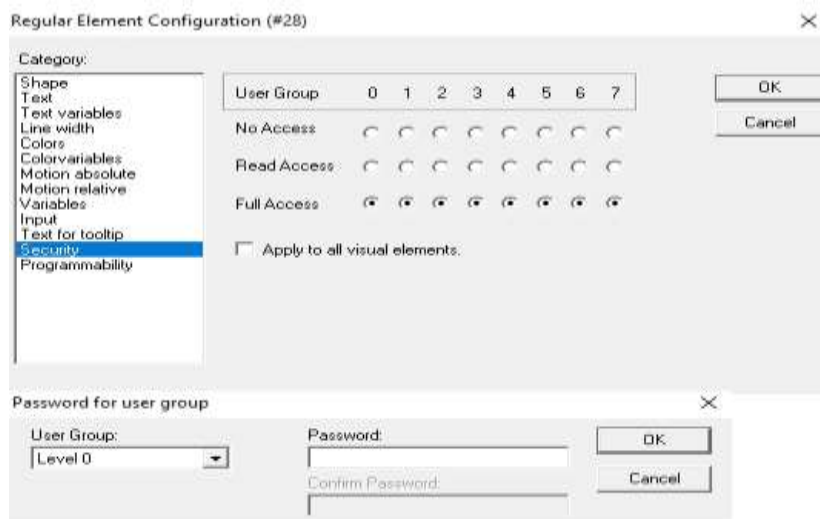


fig. 43 finestra configurazione dell'elemento selezionato (categoria Security)

CAPITOLO 4

PROGETTAZIONE SOFTWARE DI UN SISTEMA DI TRASCINAMENTO MATASSE

In questo capitolo si entra nel vivo del lavoro e del contributo che l'elaborato si propone di offrire. Vengono descritte le caratteristiche dei dispositivi che costituiscono il sistema di trascinamento. Si presenta la descrizione del funzionamento nominale e del funzionamento nel caso in cui si verifica l'evento di ARRESTO DI EMERGENZA o di FAULTS dei dispositivi. Nel caso di ARRESTO DI EMERGENZA si differenzia il comportamento del sistema nel caso in cui la matassa viene rimossa dal caso in cui continua ad essere presente; nel caso di FAULT, se il sistema di controllo del dispositivo dispone di un'informazione sensoriale completa si differenziano i FAULTS in condizioni statiche dai FAULT in condizioni dinamiche; altrimenti nel caso di un dispositivo con a bordo uno dei due sensori si presentano solo i FAULTS in condizioni dinamiche.

La trattazione prosegue analizzando le funzioni e la comunicazione tra i moduli utilizzati nell'architettura del progetto software che è stato implementato in ambiente Codesys.

Il progetto software oltre a controllare il funzionamento del sistema, consente la simulazione e visualizzazione tramite un'interfaccia grafica resa disponibile dall'ambiente di sviluppo.

Inoltre, è stato evidenziato la generalità del modello *generic device* che si presta a descrivere il controllo di qualunque dispositivo indipendentemente dal funzionamento oltre ad incapsulare la diagnostica dello stesso.

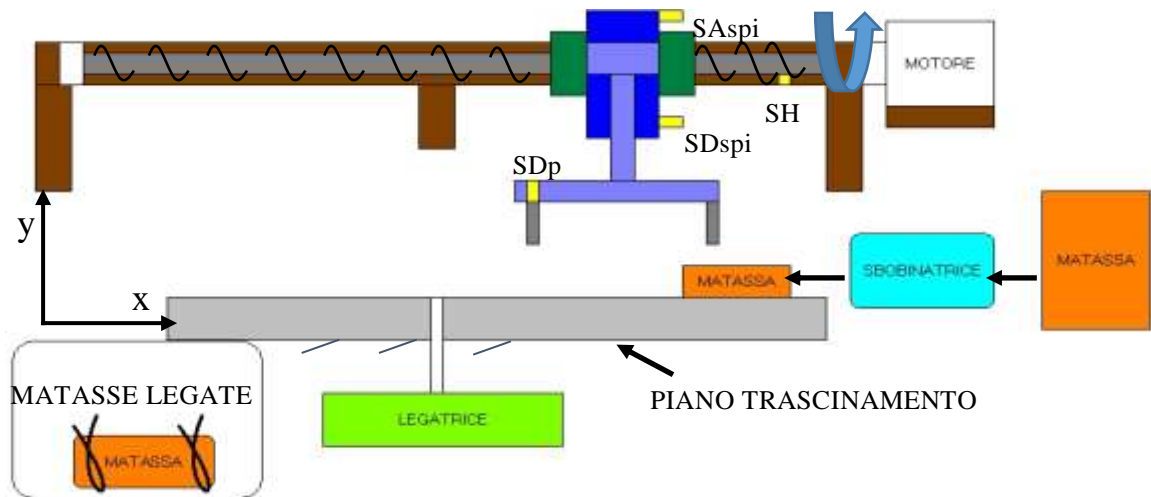


fig. 44 simulazione di un sistema di trascinamento matasse

4.1 DESCRIZIONE DEL SISTEMA DI TRASCINAMENTO

L'esigenza di avere matasse di filo con dimensioni variabili (sezione sia del cavo che dell'isolante, peso e lunghezza) dipende dal particolare impiego del conduttore e dall'organizzazione logistica dell'azienda destinataria.

Basti pensare che un cavo di potenza ha caratteristiche costruttive molto diverse rispetto ad un cavo di segnale di segnale, a parità di lunghezza la matassa nel primo caso avrà un peso e un ingombro maggiore rispetto al secondo caso.



fig. 45 cordina, cavo di segnale,
da 2.5 mm, il peso è di 15 Kg / Km



fig. cavo di potenza diametri da 6
mm il peso è di 90 Kg/ Km [14]

Dal punto di vista logistico risulta più agevole disporre di matasse di dimensioni variabili per poterne cambiare la disposizione in base alle esigenze che possono sorgere durante il cablaggio dei quadri rispetto ad avere un'unica matassa molto pesante che non è possibile muovere.

Inoltre, considerando la rigidezza 'bassa' del materiale isolante che ricopre il conduttore, la matassa non essendo afferrata da alcun sostegno potrebbe svolgersi, occupando uno spazio maggiore al caso in cui fosse legata. Quindi per ottimizzare lo spazio negli scaffali dei magazzini molte aziende che realizzano quadri elettrici richiedono matasse legate.

L'obiettivo della tesi è automatizzare un sistema di trascinamento di matasse di filo guainato e simulare l'impianto stesso.

Il sistema di trascinamento viene alimentato da un avvolto di matassa che rappresenta l'uscita di una macchina *sbobinatrice*, trascina la matassa fino alla 'postazione di legatura' in corrispondenza della quale una macchina *legatrice* realizza la legatura su entrambi i lati dell'avvolto; terminata la legatura, la matassa viene trasportata in un deposito di raccolta e il sistema di trascinamento ritorna in posizione iniziale.

Il sistema di trascinamento è costituito da uno *spintore*, rappresentato da un pistone il cui stelo spinge un'asta trasversale dove è disposta una *pinza*.

Il movimento del pistone consente la traslazione della pinza lungo la direzione individuata dall'asse y, la pinza traslando lungo l'asse x realizza la presa della matassa.

Lo spintore e la pinza sono collegate a una trasmissione vite-madrevite che trasforma il moto rotatorio generato da un motore in un moto traslatorio che realizza il trascinarsi della matassa nelle posizioni desiderate.

Lo spintore viene modellato come un pistone (generalmente pneumatico) che riceve dal sistema di controllo un segnale di comando (singola attuazione) e fornisce due segnali di retroazione (SAspi sensore in attivazione e SDspi, sensore in disattivazione). Quando il comando di attivazione del pistone è TRUE, viene portata in pressione la camera con sezione inferiore (lato stelo) che spinge lo stelo fino a raggiungere la posizione di massima quota, viceversa quando il comando diventa FALSE, la camera con sezione maggiore è in pressione esercitando una forza tale da far scendere lo stelo fino alla quota minima.

La pinza è un pistone(pneumatico) a singola attuazione e singolo segnale in retroazione (SDp, sensore disattivazione pinza). Se il comando di attivazione della pinza è TRUE, la pinza si chiude afferrando la matassa, viceversa si apre.

In genere per il controllo in posizione si utilizza un servo-azionamento brushless.

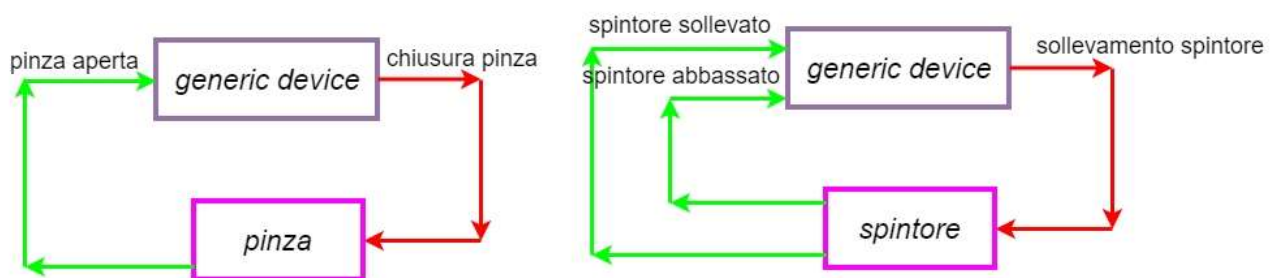


fig. 47 generic device di un dispositivo a singolo segnale di retroazione

fig. 46 generic device di un dispositivo con doppio segnale di retroazione

4.2 DESCRIZIONE DEL FUNZIONAMENTO NOMINALE DEL SISTEMA

All'avviamento della macchina, dopo aver eseguito la 'procedura di homing, il sistema si porta in posizione, spintore a quota massima e pinza aperta, di attesa matassa, in presenza della quale lo spintore si porta in corrispondenza della quota in cui è presente la matassa e la pinza si chiude afferrando la stessa. A questo punto si attiva il motore, che trascina l'avvolto fino alla posizione di legatura, posizione in cui il motore si disattiva per un tempo pari all'operazione di legatura. Al termine della legatura del primo lato della matassa, la macchina legatrice si disattiva e il motore si riattiva trascinando la matassa fino a far corrispondere l'altro lato da legare nella posizione di legatura e si riattiva l'operazione di legatura.

Terminata la legatura del secondo lato della matassa il motore si riattiva trascinando la matassa fino alla posizione di scarico in corrispondenza della quale, il motore si disattiva, la pinza si apre, lo spintore risale e la matassa legata cade nel deposito delle matasse legate. A questo punto il motore si riattiva e il sistema si riporta nella posizione di partenza ripetendo ciclicamente la sequenza descritta.

4.3 COMANDO DI ARRESTO

Durante il funzionamento di una macchina automatica, le norme di sicurezza prevedono dei dispositivi di sicurezza, (es. FUNGO DI EMERGENZA fig.46), che attivati da parte dell'operatore azzerano nel minor tempo possibile la presenza di pericolo di una macchina. La funzione di arresto di emergenza è destinata ad evitare o ridurre, al loro sorgere, i pericoli per le persone e i danni alle macchine o lavorazioni in corso.



*fig. 48 dispositivo di sicurezza:
fungo di emergenza*

L'arresto è un comando tramite il quale si ottiene il fermo di una macchina o una parte di essa. Ogni macchina deve essere munita di almeno un dispositivo di comando che consenta l'arresto generale in condizioni di sicurezza. In presenza di più postazioni di lavoro, ognuna di questa deve essere munita di un dispositivo di comando di sicurezza che in relazione ai rischi, consenta di arrestare l'intera macchina o una parte di essa garantendo la sicurezza.

Esistono tre categorie di arresto [6]:

- Categoria 0, ARRESTO NON CONTROLLATO (O DI EMERGENZA) attraverso il quale si interrompe la fonte di energia (tramite il fungo di emergenza) che alimenta gli organi pericolosi della macchina.
- Categoria 1, ARRESTO CONTROLLATO, tramite il quale vengono fermati gli elementi pericolosi della macchina mantenendo attiva la fonte di energia che sarà rimossa ad arresto avvenuto.
- Categoria 2, ARRESTO CONTROLLATO, uguale alla Categoria 1 solo che la fonte di energia rimane attiva anche dopo l'arresto.

La scelta della categoria di arresto della macchina viene intrapresa sulla base della valutazione del rischio e in relazione ai requisiti funzionali della macchina.

Ogni macchina deve essere munita di uno o più dispositivi di arresto di emergenza e ogni dispositivo deve essere attivabile mediante una singola azione umana. I dispositivi di arresto di emergenza devono possedere le seguenti caratteristiche [7]:

1. deve essere chiaramente individuabile e rapidamente accessibile;
2. una volta azionato, deve restare inserito;
3. deve essere possibile disinserirlo mediante una adeguata manovra(riarmo);
4. il riarmo dell'arresto di emergenza non deve avviare la macchina, ma solo consentirne il riavvio mediante apposito comando (es. pulsante di RESET e dopo MARCIA);
5. l'azionamento del comando provoca l'arresto del processo pericoloso;
6. L'ordine di arresto della macchina deve essere prioritario rispetto agli ordini di avviamento;
7. non deve essere possibile avviare il moto pericoloso fino a che tutti gli attuatori di comando azionati non sono stati ripristinati manualmente, singolarmente e intenzionalmente;

Il comando di arresto di emergenza, come ogni altro dispositivo di protezione, ha come caratteristica fondamentale il ripristino manuale; cioè viene attivato volutamente da un operatore e non con un comando automatico. La sua funzione non è sostitutiva di altre misure di sicurezza, come dispositivi automatici di sicurezza o protezione (MODULI SAFETY) ma deve essere considerata come supplementare; infatti la norma EN 292/2 punto 6 dichiara che in presenza di rischi residui o condizioni anomale di funzionamento è opportuno usufruire di un dispositivo che se

attivato riduca automaticamente nel miglior modo possibile con soluzioni diverse a seconda dei rischi, dei cicli funzionale e del tipo di macchina.

Nella progettazione dello sviluppo del codice di controllo del sistema di trascinamento matasse è stato ritenuto opportuno adottare un arresto di Categoria 0, attivabile tramite pressione del fungo di emergenza disponibile sulla pulsantiera fisica.

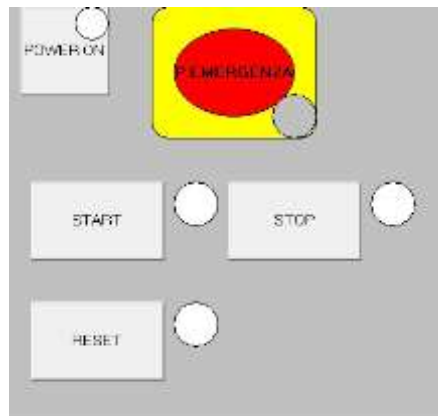


fig. 49 pulsantiera fisica del sistema di trascinamento

4.3.1 DESCRIZIONE DEL FUNZIONAMENTO IN CASO DI EMERGENZA

Durante il funzionamento nominale può verificarsi un'anomalia dell'impianto, quindi per operare in condizioni di sicurezza l'operatore preme il fungo e la macchina frena nel minor tempo possibile.

Nel caso del sistema di trascinamento coerentemente alla definizione di arresto in Categoria 0, se durante il trascinamento della matassa l'operatore preme il fungo, viene interrotta solo l'alimentazione dei dispositivi in questione. Quindi lo spintore rimane disattivo (basso), la pinza rilascia la matassa e il servo-azionamento si ferma. L'impianto può essere

riattivato solo dopo che in seguito all'intervento dell'operatore venga eseguita la sequenza delle operazioni manuali:

- 1) rilascio del fungo di emergenza;
- 2) attivazione del comando di RESET;
- 3) attivazione del comando di START.

Nel progetto del software è stato differenziato il comportamento della macchina nel caso in cui in seguito all'ARRESTO IMMEDIATO l'operatore ritiene necessario rimuovere la matassa in seguito ad un danneggiamento, dal caso in cui la matassa rimane.

Nel primo caso lo spintore con la pinza aperta si solleva e il sistema ritorna nella posizione di attesa matassa; nel secondo caso il sistema riparte dallo stato in cui si era arrestato, la pinza si richiude e la macchina evolve nella sequenza delle operazioni del funzionamento nominale.

4.4 ARCHITETTURA SOFTWARE DEL PROGETTO

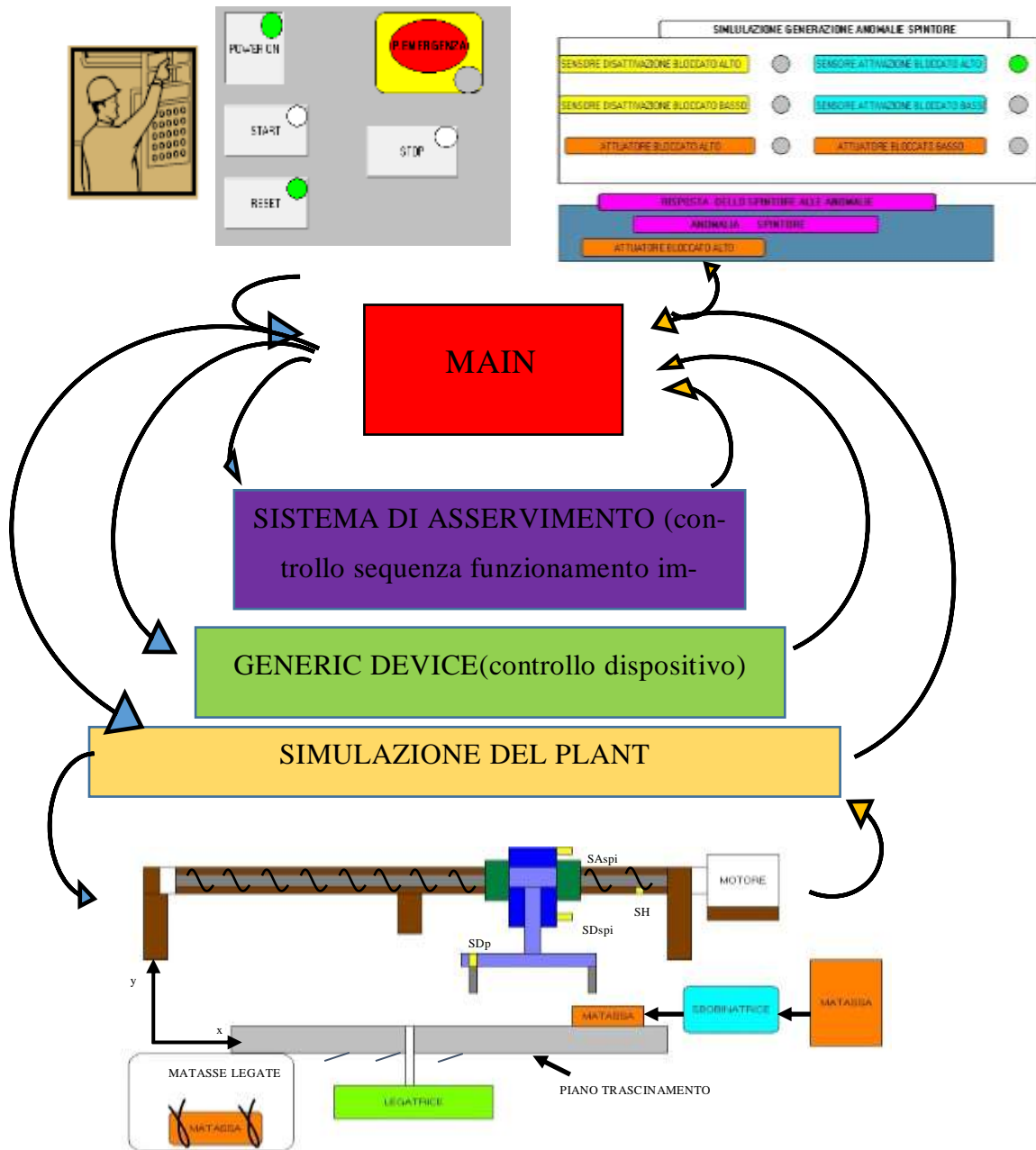


fig. 50 architettura del software del progetto

Come si nota dalla fig.48 il modello del software è caratterizzato da un'architettura piramidale in cui la comunicazione tra i moduli avviene tramite un'interfaccia software tra il modulo MAIN e i moduli comunicanti.

Si procede con una descrizione relativa alla funzione svolta da ogni modulo tramite la rappresentazione del modello 'automi a stati finiti', poi la traduzione in linguaggio 'testo strutturato' dello stesso e infine viene rappresentata la comunicazione tra i vari blocchi tramite degli esempi.

Viene evidenziato il disaccoppiamento tra i moduli in particolare tra il modulo MAIN, il modulo di asservimento e i *generic devices* che è alla base della riutilizzabilità del codice.

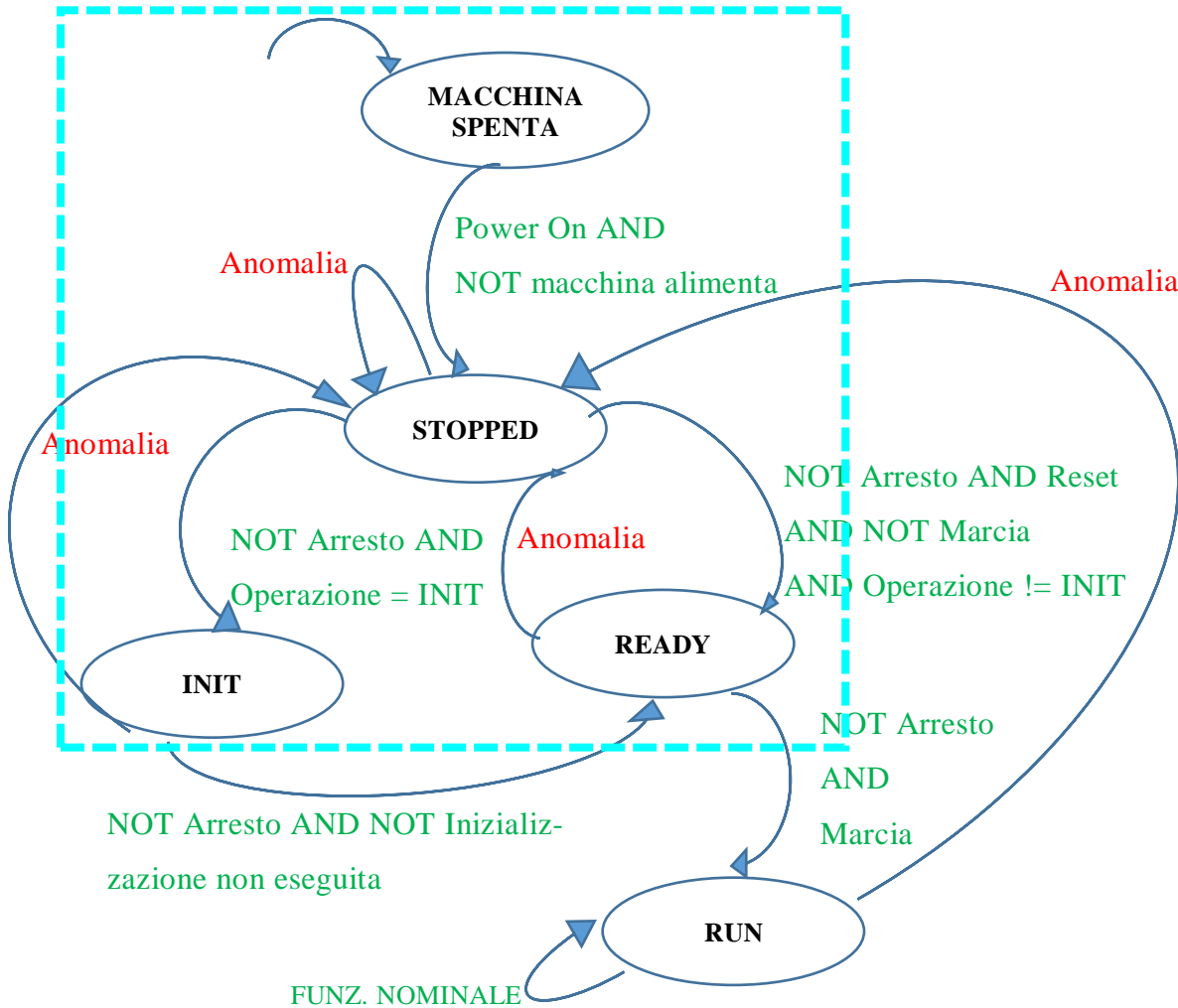
Infine, viene mostrata l'implementazione dell'*identificazione dei FAULTS*, il reset degli stessi e il riavvio della macchina.

4.4.1 MAIN

Il modulo MAIN gestisce il funzionamento di un generico impianto di automazione. Viene descritto l'automa di fig. 49 considerando prima il funzionamento nominale e dopo in caso di arresto o FAULTS.

Appena l'operatore preme il pulsante 'Power On', la macchina viene alimentata e si porta nello stato STOPPED, la prima volta che si trova in tale stato viene eseguita l'operazione di inizializzazione (o homing) quindi 'Operazione = INIT'. Dopo che viene eseguita l'inizializzazione di tutti e tre i dispositivi il sistema evolve nello stato 'READY' e appena l'operatore abilita il consenso di marcia, l'automa transita nello stato 'RUN' e lì permane se il funzionamento del sistema risulta essere nominale. In qualunque stato, a meno del primo poiché la macchina non risulta essere ancora alimentata, al verificarsi dell'ARRESTO DI EMERGENZA o di un FAULT dello spintore o della pinza, l'automa si porta

nello stato 'STOPPED' e conseguentemente i dispositivi dell'impianto si arrestano (vedi modulo di asservimento).



NOT Arresto:= NOT Arresto AND NOT Faults Pistone AND Not Faults Pinza

Anomalia:= Arresto Emr OR Faults Pinza OR Faults Pistone

fig. 51 automi a stati finiti del modulo 'Main'

La fig. 50 rappresenta la traduzione in linguaggio testo strutturato di una parte dell'automa (parte tratteggiata) di fig. 49.

```

CASE (stato_macchina) OF
(*0*) stato_speinta:
    led_MARCIA           => FALSE;
    led_stop             => FALSE;
    led_reset            => FALSE;
    stato_impianto       => 'attesa switch on potenza azionamenti';
    macchina_alimentata  => FALSE;
    IF power_on
    AND
    NOT macchina_alimentata
    THEN
        macchina_alimentata => TRUE;
        stato_macchina      => stato_stopped;
    END_IF
(*1*) stato_stopped:
    led_MARCIA           => FALSE;
    led_stop             => FALSE;
    led_reset            => TRUE;
    stato_impianto       => 'attesa comando di reset';
    o_Tipo_Operazione    := 30; (*RICHIESTA DI ARRESTO AL MODULO DI ASSERVIMENTO; NON CAPISCO PERCHE' NON MI PRENDE IL NOME stop*)
    IF NOT Arresto_Immediato AND reset AND NOT o_Fault_spintore AND NOT o_Fault_pinza AND NOT MARCIA
    THEN
        IF Operazione_prima_Allarme = RUN (*NEL CASO IN CUI IL SISTEMA ERA IN MARCIA*)
        THEN
            stato_macchina := stato_ready;
        ELSE
            o_Tipo_Operazione := INIT;
            stato_macchina := stato_init;
        END_IF
    END IF

```

fig. 50 parte del codice del modulo 'Main'

4.4.2 MODULO: 'SISTEMA'

Coerentemente con la scelta di far comunicare tutti i blocchi tramite il modulo "MAIN" sono stati implementati due *oggetti* (Function Block): il "Sistema di Asservimento" e il "*generic device*" istanziati nel MAIN.

L'oggetto "Sistema di Asservimento" che rappresenta il *tipo* della variabile 'Sistema' descrive la logica di funzionamento dell'impianto.

L'oggetto "*generic device*" implementa il sistema di controllo di un dispositivo; quindi per modellare il funzionamento della pinza e dello spintore è stato necessario istanziare l'oggetto due volte.

Il generale funzionamento del dispositivo, si specializza nell'istanza creata in base ai parametri d'ingresso che riceve dal MAIN.

Viene data una rappresentazione dell'automa completo che descrive il "Sistema di Trascinamento", rappresentando prima il funzionamento nominale (fig.52) poi il funzionamento nel caso di FAULT o ARRESTO DI EMERGENZA (fig.53).

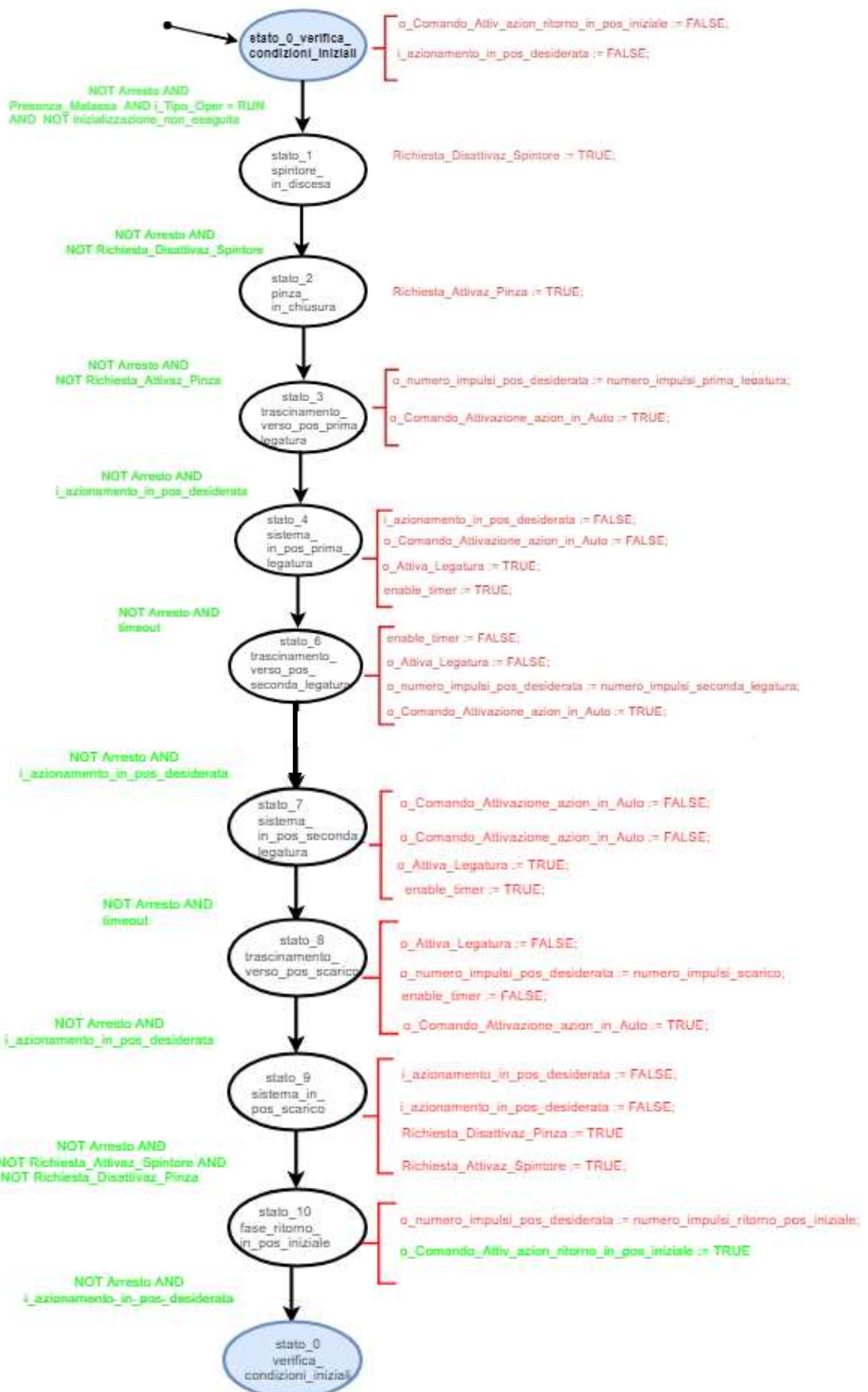


fig. 53 macchina a stati del modulo "sistema" (funzionamento nominale)

A questo punto se la matassa è presente, l'automa evolve nello stato 12, viene attivata (cioè si chiude) la pinza e quando la pinza è chiusa, la macchina a stati ritorna nello stato 3 e il sistema continua nel funzionamento nominale; altrimenti dallo stato 12 passa allo stato 9 e il sistema ritorna in posizione iniziale.

4.4.3 GENERIC DEVICE

Nella scrittura del software con l'aumentare della complessità si sviluppati modelli che andavano oltre il concetto di funzione (classico) con l'obiettivo di creare *oggetti base* che possono essere **riutilizzati** in più software. Questi modelli sono alla base della *programmazione ad oggetti*.

Questo concetto può essere utilizzato anche nel campo controllistico in cui è necessario dividere la logica di controllo (automa di fig.52 e fig.53, 'SISTEMA') delle sequenze operative del sistema dal funzionamento di ogni dispositivo (*generic device*).

La logica di controllo chiede, tramite delle richieste, l'attivazione/disattivazione di un servizio al sistema sottostante che notifica il completamento del servizio annullando la richiesta.

In letteratura questo tipo di *generic device* è noto come DO – DONE in cui la logica di controllo attiva (DO) l'oggetto e quando questo è attivo, notifica che è attivato (DONE) che si differenzia dal tipo START – STOP perché in quest'ultimo la logica di controllo attiva e disattiva l'oggetto.

L'astrazione è evidente perché alla logica di controllo non interessano le caratteristiche dell'oggetto sottostante (tipo di sensori e attuatori) ma solo che venga soddisfatta la richiesta inviata.

Il *generic device* implementato nel progetto di questa tesi è di tipo DO-DONE.

La fig.53 rappresenta la prima formulazione della macchina a stati che implementa il funzionamento base del '*generic device*', mentre la fig.54 rappresenta l'estensione del primo, considera tutte le transizioni che possono verificarsi in un caso reale e è orientata alla diagnostica tramite l'introduzione del 'tempo'.

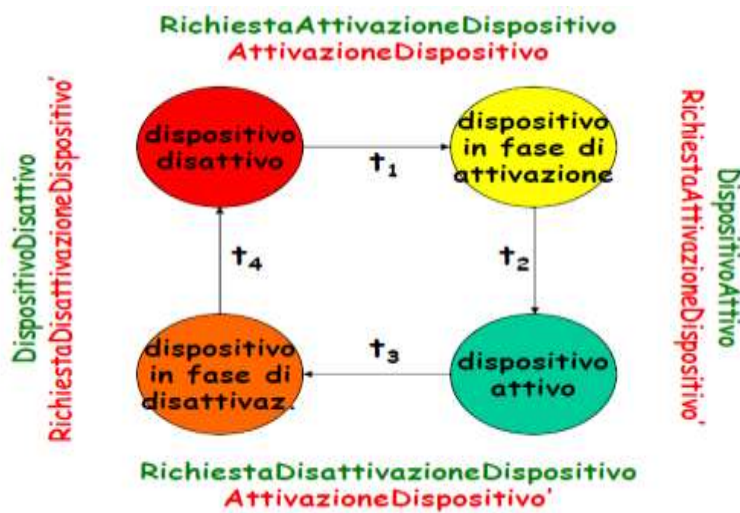


fig. 55 prima formulazione dell'automa del generic device

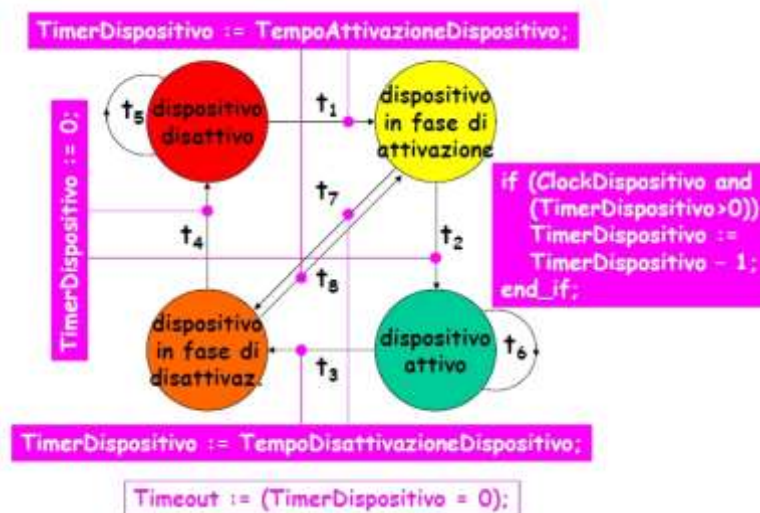


fig. 56 estensione dell'automa del generic device

```

stato_10_dispositivo_disattivo:
  IF L_tipo_operaz = INIT
  THEN
    Dispositivo_non_inizializzato = FALSE; (*dispositivo inizializzato con "stato_dispositivo_attivo"*)
  ELSE
    Richiesta_Disattivazione_Dispositivo = FALSE;
    IF Richiesta_Attivazione_Dispositivo
    THEN Comando_Attiv_Dispositivo = TRUE;
        enable_timer_attivazione_TON = TRUE;
        stato_dispositivo = stato_20_in_fase_di_attivazione;
    END_IF
  END_IF
stato_20_in_fase_di_attivazione:
  IF (Richiesta_Disattivazione_Dispositivo)
  THEN Richiesta_Attivazione_Dispositivo = FALSE;
        Comando_Attiv_Dispositivo = FALSE;
        enable_timer_attivazione_TON = FALSE;
        enable_timer_disattivazione_TON = TRUE;
        stato_dispositivo = stato_40_in_fase_di_disattivazione;
  ELSEIF (sensore_in_attivazione) (* OR time_attivazione *)
  THEN
    Richiesta_Attivazione_Dispositivo = FALSE;
    enable_timer_attivazione_TON = FALSE; (*davo bloccare il conteggio del timer_attivazione*)
    stato_dispositivo := stato_30_dispositivo_attivo;
  END_IF
stato_30_dispositivo_attivo:
  IF L_tipo_operaz = INIT
  THEN
    Dispositivo_non_inizializzato := FALSE; (*dispositivo inizializzato con "stato_dispositivo_attivo"*)
  ELSE
    Richiesta_Attivazione_Dispositivo := FALSE; (*se il dispositivo è già nello stato di "dispositivo attivo" abbasso la richiesta di attivazione*)
    IF (Richiesta_Disattivazione_Dispositivo)

```

fig. 57 Traduzione in codice dell'automata del generic device

In generale lo stato iniziale dell'automata che rappresenta il controllo del dispositivo (fig. 53 e fig.54) non coincide con lo stato iniziale del dispositivo fisico. L'operazione che consente la sincronizzazione dello stato dell'automata con lo stato del dispositivo fisico è l'inizializzazione (INIT). Il frammento di codice (fig.55) descrive il caso in cui se viene richiesto che lo stato iniziale del dispositivo sia 'attivo' in base all'informazione sensoriale (sensore attivazione) lo stato dell'automata si porta in fase_di_attivazione o coincide con lo stato richiesto.

```

CASE (stato_dispositivo) OF
stato_100_init: (*SECONDO ME, IN QUESTO STATO INVOCHEREI UNA FUNZIONE CHE SI OCCUPADI INIZIALIZZARE L'AUTOMA DEL G.D.*)
  Richiesta_Attivazione_Dispositivo := FALSE;
  Richiesta_Disattivazione_Dispositivo := FALSE;
  IF Stato_iniziale_dispositivo_attivo
  THEN
    Comando_Attiv_Dispositivo := TRUE; (*A=1*)
    enable_timer_attivazione_TON := TRUE;
    (*SECONDO LA MIA INTERPRETAZIONE*)
    IF NOT sensore_in_attivazione (*a = 0---- considero i casi di porta complet. chiusa a=0 e d =1 o porta dischiusa a = 0, d=0*)
    THEN
      stato_dispositivo := stato_20_in_fase_di_attivazione ;
    ELSE
      enable_timer_attivazione_TON := FALSE; (*il dispositivo è già attivo, quindi scarico il timer*)
      stato_dispositivo := stato_30_dispositivo_attivo;
    END_IF

```

fig. 58 parte di codice del generic device: inizializzazione

Nel caso in cui a bordo di un dispositivo non siano presenti entrambi i sensori o siano presenti uno dei due è necessaria la ricostruzione dell'informazione sensoriale associata al sensore mancante.

Tale ricostruzione viene realizzata utilizzando, quando disponibile, l'informazione del sensore presente, la nozione di *stato* e di *tempo*.

Se ad esempio consideriamo come dispositivo una 'porta' e lo 'stato dispositivo disattivo' coincide con *porta chiusa* mentre stato 'dispositivo attivo' con *porta aperta*. Ipotizzando che sia presente solo il *sensore di disattivazione*, partendo dallo 'stato dispositivo disattivo' in seguito ad un *comando di apertura*, la porta si considera aperta (*sensore_in_attivazione* = TRUE) quando la porta non è chiusa e dopo un intervallo di tempo necessario per aprirsi e lo stato dell'automa si trova in 'stato dispositivo in fase di attivazione' oppure sia trova in 'stato dispositivo attivo' (parte di codice evidenziata in fig.55).

```
(*-----GESTIONE-----DEI-----FAULTS-----*)

(**individuazione sensori**)
IF i_tipo_dispositivo = Dispositivo_con_retroazione_in_disattivazione
THEN
    sensore_in_attivazione := NOT sensore_in_disattivazione AND (stato_dispositivo=stato_30_dispositivo_attivo
                                                                OR stato_dispositivo= stato_20_in_fase_di_attivazione
                                                                AND timeout_attivazione) ;
    (*sensore_in_attivazione := NOT sensore_in_disattivazione AND stato_dispositivo=stato_20_in_fase_di_attivazione; *)
ELSIF i_tipo_dispositivo = Dispositivo_con_retroazione_in_attivazione
THEN
    sensore_in_disattivazione := NOT sensore_in_attivazione AND (stato_dispositivo= stato_10_dispositivo_disattivo
                                                                OR stato_dispositivo= stato_40_in_fase_di_disattivazione AND timeout_disattivazione) ;

ELSIF i_tipo_dispositivo = Dispositivo_senza_retroazione
THEN
    sensore_in_attivazione := stato_dispositivo=stato_30_dispositivo_attivo OR stato_dispositivo= stato_20_in_fase_di_attivazione AND timeout_attivazione ;
    sensore_in_disattivazione := stato_dispositivo= stato_10_dispositivo_disattivo OR stato_dispositivo=stato_40_in_fase_di_disattivazione AND timeout_disattivazione ;
END_IF
```

fig. 59 parte di codice del generic device: ricostruzione dell'informazione sensoriale

Nel codice che implementa il *generic device* è incapsulata la diagnostica dei FAULTS. È possibile segnalare in modo puntuale la causa del FAULT (identificando il livello logico dello stato dell'attuatore o del sensore che presenta un'anomalia) solo nel caso in cui si dispone della completa informazione sensoriale utilizzando la nozione di stato e di tempo, nel caso in cui si dispone di un'informazione sensoriale parziale è possibile dare una segnalazione generica di guasto al dispositivo, altrimenti quando non si dispone di alcuna informazione sensoriale non è possibile individuare una qualunque situazione di FAULT (fig.56). Dopo aver segnalato il FAULT, l'operatore interviene per riparare l'eventuale componente segnalato e dopo aver premuto il pulsante RESET, il programma resetta le segnalazioni pendenti ripristinando la condizione di funzionamento nominale (fig.57).

```

(SEGNALAZIONE ----- FAULTS ----- SENSORE ----- ATTIVAZIONE*)
IF i_tipo_dispositivo = Dispositivo_con_doppia_retroazione
THEN
  IF (NOT Comando_Activ_Dispositivo AND sensore_in_disattivazione AND sensore_in_attivazione) AND timeout_disattivazione
  THEN
    Fault_sensore_in_attivazione_alto := TRUE;
    (*ATTENZIONE QUANDO DOPO IL RESET, LO SPINTORE TORNA IN ALTO SI ATTIVA Fault_sensore_in_attivazione_basso*)
  ELSIF (Comando_Activ_Dispositivo AND NOT sensore_in_disattivazione AND NOT sensore_in_attivazione) AND timeout_attivazione
  THEN
    Fault_sensore_in_attivazione_basso := TRUE;
  END_IF
  IF Fault_sensore_in_attivazione_alto OR Fault_sensore_in_attivazione_basso
  THEN
    Fault_sensore_in_attivazione := TRUE;
  END_IF
|
ELSIF i_tipo_dispositivo = Dispositivo_con_retroazione_in_disattivazione
THEN
  IF (Comando_Activ_Dispositivo AND sensore_in_disattivazione AND NOT sensore_in_attivazione) AND timeout_attivazione
  OR
  ( NOT Comando_Activ_Dispositivo AND NOT sensore_in_disattivazione AND NOT sensore_in_attivazione) AND timeout_disattivazione
  THEN
    Fault_Dispositivo := TRUE;
  END_IF

```

fig. 60 parte di codice del generic device: segnalazione faults

```

RESET          SEGNALAZIONE          FAULTS          SENSORE          ATTIVAZIONE
IF i_tipo_dispositivo = Dispositivo_con_doppia_retroazione
THEN
  IF
    Fault_Dispositivo AND Fault_sensore_in_attivazione
  THEN
    IF Fault_sensore_in_attivazione_basso AND NOT simulazione_plantvisu_sensore_attivazione_spintore_bloccato_basso AND MAIN.reset
    THEN
      Fault_sensore_in_attivazione_basso := FALSE;
      Fault_sensore_in_attivazione := FALSE;
      Fault_Dispositivo := FALSE;

      ELSIF Fault_sensore_in_attivazione_alto AND NOT simulazione_plantvisu_sensore_attivazione_spintore_bloccato_alto AND MAIN.reset
      THEN
        Fault_sensore_in_attivazione_alto := FALSE;
        Fault_sensore_in_attivazione := FALSE;
        Fault_Dispositivo := FALSE;
      END_IF
    END_IF
  END_IF
ELSIF i_tipo_dispositivo = Dispositivo_con_retroazione_in_disattivazione
THEN
  IF
    Fault_Dispositivo AND NOT simulazione_plantvisu_attuatore_pinza_bloccato_basso (* OR NOT simulazione_plantvisu_Fault_attuatore_pinza
    OR NOT simulazione_plantvisu_Fault_sensore_disattivazione_pinza_bloccato_alto OR NOT simulazione_plantvisu_sen
    AND MAIN.reset
  THEN
    Fault_attuatore_alto := FALSE;
    Fault_attuatore_basso := FALSE;
    Fault_sensore_in_disattivazione_alto := FALSE;
    Fault_sensore_in_disattivazione_basso := FALSE;
    Fault_Dispositivo := FALSE;
  END_IF
END_IF

```

fig. 61 parte di codice del generic device: reset segnalazioni faults

4.4.4 SIMULAZIONE DEL PLANT

Il modulo "simulazione_plant" simula il funzionamento di tutti i dispositivi del sistema di trascinamento. Tramite i comandi che riceve dall'istanza del Generic Device o dall'azionamento, incrementa/decrementa alcune variabili che descrivono il moto del particolare dispositivo in questione. Quando il dispositivo raggiunge la posizione di fine corsa, il modulo imposta a TRUE/FALSE il relativo sensore e tale informazione viene veicolata alla relativa istanza del *generic device* (fig.60).

Inoltre, rappresenta uno strato software in cui vengono dichiarati le variabili associate agli "elementi" che si desidera visualizzare.

Nel progetto sono stati realizzati quattro 'Object' che hanno consentito di interagire con i moduli del progetto in base alle variabili di visualizzazione di ingresso e di uscita.

Gli oggetti 'FAULTS_PINZA' e 'FAULTS_SPINTORÈ' sono suddivisi in due parti, la parte superiore che rappresenta una serie di pulsanti collegati alle variabili d'ingresso che premuti simulano il verificarsi di un'anomalia (es. sensore in disattivazione del pistone bloccato alto) e la parte inferiore che segnala il FAULTS del dispositivo relativo.

L'object 'PULSANTI_FISICI_E_HMI' rappresenta la pulsantiera fisica presente in ogni macchina automatica; mentre l'object 'simulazione _plant' simula il sistema di trascinamento (fig.61).

```
(*-----COMANDI-----SPINTORE-----*)
IF visu_comando_attivazione_spintore AND NOT visu_attuatore_spintore_bloccato_basso
  AND
  asse_y_spintore > -32
  THEN
    asse_y_spintore := asse_y_spintore -3;
  ELSIF NOT visu_comando_attivazione_spintore AND NOT visu_attuatore_spintore_bloccato_alto
  AND
  asse_y_spintore < 40
  THEN
    asse_y_spintore := asse_y_spintore + 3;
  END_IF
(*-----SENSORI-----SPINTORE-----*)
IF asse_y_spintore <= -31(* OR visu_sensore_attivazione_spintore_bloccato_alto *)
  THEN
    visu_sensore_in_attivazione_spintore := TRUE;
  ELSIF asse_y_spintore >= -28 (*AND NOT visu_sensore_attivazione_spintore_bloccato_alto *)
  THEN
    visu_sensore_in_attivazione_spintore := FALSE;
  END_IF
IF asse_y_spintore >= 39
  THEN
    visu_sensore_in_disattivazione_spintore := TRUE;
  ELSIF asse_y_spintore <= 35
  THEN
    visu_sensore_in_disattivazione_spintore := FALSE;
  END_IF
```

fig. 62 modulo: simulazione_plant

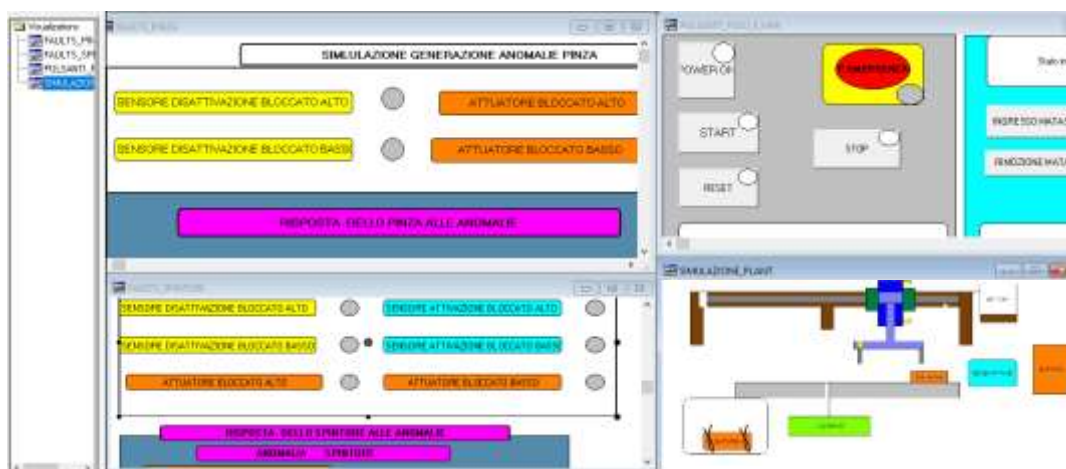


fig. 63 objects 'Visualizations'

4.4.5 COMUNICAZIONE TRA I VARI MODULI

Per comprendere lo scambio di dati tra i vari moduli utilizzati nel realizzare il progetto di questa tesi consideriamo ad esempio la fase di inizializzazione. Dopo aver mandato in esecuzione (RUN) il progetto e aver premuto il pulsante di POWER_ON, il modulo 'MAIN' rimane in attesa che venga premuto il pulsante di RESET (fig.62)

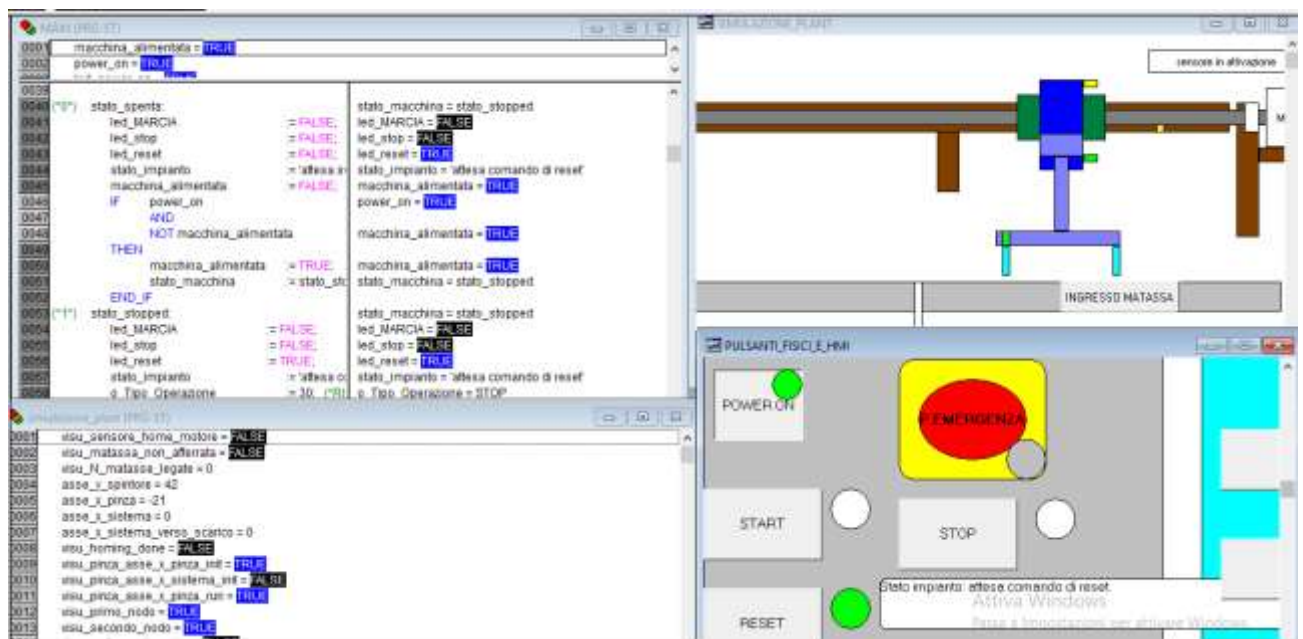


fig. 64 impianto in attesa di RESET

Appena viene premuto RESET, il MAIN transita nello *stato_init*, ogni modulo esegue l'operazione di inizializzazione; il modulo 'Sistema' si porta nello *stato_0* e dopo aver inviato il comando di attivazione dell'attuatore al modulo "simulazione_plant", al termine dell'inizializzazione notifica al MAIN che l'inizializzazione è conclusa. Le istanze dell'oggetto "generic device" si sincronizzano con lo stato attivo nel caso dello spintore e con lo stato disattivo nel caso della pinza. Nel primo caso viene inviato il comando di attivazione dello spintore, nel secondo il comando di disattivazione della pinza (fig.63).

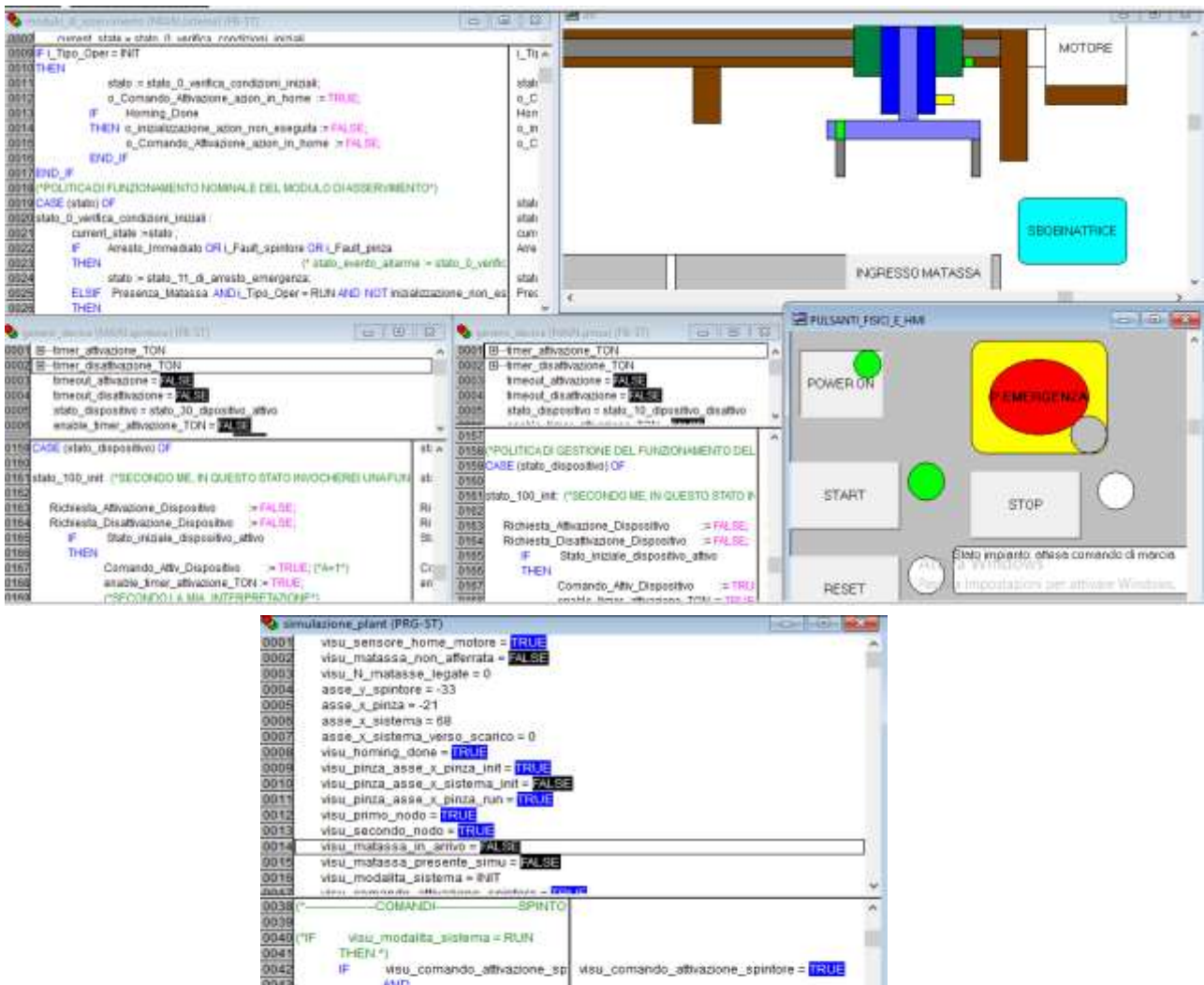


fig. 65 impianto in attesa di START (inizializzazione completata)

Gli automi di ogni modulo risultano nello stato iniziale; in seguito alla pressione del pulsante di 'START' il sistema evolve nel funzionamento nominale.

CAPITOLO 5

CONCLUSIONI

L'elaborato di questa tesi mi ha consentito di apprendere l'efficacia degli strumenti necessari per la realizzazione del software di controllo e simulazione del sistema di un sistema di automazione industriale.

Per descrivere l'evoluzione del flusso sequenziale delle operazioni seguendo una logica coerente con le specifiche del problema, è stato utilizzato il modello *automi a stati finiti*. Tale formulazione risulta una modellazione di alto livello la cui traduzione in linguaggio testo strutturato risulta essere immediata.

La struttura modulare del codice consente la riutilizzabilità dei moduli in qualunque altra applicazione nell'ambito controllo di macchine automatiche; in particolare si evidenzia la generalità del modulo di gestione del funzionamento di un dispositivo, *generic device*, modello per il controllo di un dispositivo indipendentemente dalle sue caratteristiche (attuatori e sensori). Inoltre, nel *generic device* è incapsulata la gestione (segnalazione e reset) dei FAULTS che possono verificarsi durante il funzionamento nominale dei dispositivi a bordo macchina.

Durante lo sviluppo del software l'utilizzo di elementi grafici è stato utile, specie nel caso di elementi collegati con variabili di moto, perché hanno consentito di avere un riscontro veloce sulla correttezza del codice e nel caso contrario intervenire sulle variabili collegate agli elementi stessi.

ELENCO DELLE FIGURE

fig. 1 schema a blocchi di un sistema di controllo ad anello chiuso	13
fig. 2 classificazione dei processi produttivi	15
fig. 3 tipologie di sistemi di automazione [2]	20
fig. 4 confronto dei sistemi di automazione industriale nel grafico c.u/ v [2]	21
fig. 5 processo di produzione del prodotto [2]	22
fig. 6 attività di supporto alla produzione [1]	23
fig. 7 relazione tra CIM, FMS, e clienti/fornitori [1]	24
fig. 8 CIM: modello piramidale [1]	25
fig. 9 CIM: architettura modulare [1]	25
fig. 10 tasks real-time [1]	26
fig. 11 dispositivi del livello " campo " [1]	27
fig. 12 dispositivi del livello " macchina " [1]	28
fig. 13 esempio di cella di produzione [1]	28
fig. 14 software del livello " stabilimento " [1]	29
fig. 15 tipi di controllo per i livelli del modello CIM [1]	30
fig. 16 i tempi della produzione tradizionale [2]	34
fig. 17 esempio di un sistema pick-and-place [4]	35
fig. 18 impianto: sistema di asservimento, avvolgitore-svolgitore, legatrice	36
fig. 19 ribobinatrice (o avvolgitore-svolgitore) [9]	37
fig. 20 classificazione delle macchine ribobinatrici	38
fig. 21 avvolgitore-svolgitore manuale [9]	39
fig. 22 vista dall'alto e laterale di una bobina madre	39
fig. 23 dimensioni dello svolgitore	40
fig. 24 avvolgitore-svolgitore ad asse parallelo [10]	40
fig. 25 ribobinatrice a rastrelliera [8]	41
fig. 26 sbobinatrice ad asse perpendicolare [11]	42
fig. 27 fasi di lavoro della macchina " ribobinatrice "	43
fig. 28 sistema di asservimento: nastro, robot e tavola rotante [12]	44
fig. 29 sistema di asservimento: nastro2, robot e nastro1 [13]	45
fig. 30 fasi di lavoro della macchina " legatrice "	48
fig. 31 spintore serve il filo formato alle pinze della " legatrice "	49
fig. 32 rotazione delle pinze per effettuare l'operazione di legatura	49
fig. 33 modello software [5]	51
fig. 34 nuova POU	56
fig. 35 inserimento di u programma all'interno di untask	57

fig. 36 progetto	57
fig. 37 inserimento libreria	58
fig. 38 schema di funzionamento di Codesys [5]	59
fig. 39 finestra di configurazione dell'elemento visualizzato (categoria 'Text')	Errore. Il segnalibro non è definito.
fig. 40 finestra configurazione dell'elemento visualizzato(categoria Motion Absolute)	Errore. Il segnalibro non è definito.
fig. 41 finestra configurazione dell'elemento visualizzato(categoria Variables)	Errore. Il segnalibro non è definito.
fig. 42 finestra configurazione dell'elemento selezionato (categoria'Input')	Errore. Il segnalibro non è definito.
fig. 43 finestra configurazione dell'elemento selezionato (categoria Security)	Errore. Il segnalibro non è definito.
fig. 44 simulazione di un sistema di trascinamento matasse	Errore. Il segnalibro non è definito.
fig. 45 cordina, cavo di segnale, da 2.5 mm, il peso è di 15 Kg / Km [14]	Errore. Il segnalibro non è definito.
fig. 46 dispositivo di sicurezza: fungo di emergenza	Errore. Il segnalibro non è definito.
fig. 47 pulsantiera fisica del sistema di trascinamento	Errore. Il segnalibro non è definito.
fig. 48 architettura del software del progetto	Errore. Il segnalibro non è definito.
fig. 49 automi a stati finiti del modulo 'Main'	Errore. Il segnalibro non è definito.
fig. 50 Automa a stati finito modulo 'Main'	Errore. Il segnalibro non è definito.
fig. 51 macchina a stati del modulo " sistema" (funzionamento nominale)	Errore. Il segnalibro non è definito.
fig. 52 macchina a stati del modulo " sistema" (funzionamento nel caso di anomalia)	Errore. Il segnalibro non è definito.
fig. 53 prima formulazione dell'automa del Generic Device	Errore. Il segnalibro non è definito.
fig. 54 estensione dell'automa del Generic Device	Errore. Il segnalibro non è definito.
fig. 55 Traduzione in codice dell'automa del Generic Device	Errore. Il segnalibro non è definito.
fig. 56 parte di codice del Generic Device: inizializzazione	Errore. Il segnalibro non è definito.
fig. 57 parte di codice del generic Device: ricostruzione dell'informazione sensoriale	Errore. Il segnalibro non è definito.
fig. 58 parte di codice del Generic Device: segnalazione FAULTS	Errore. Il segnalibro non è definito.
fig. 59 parte di codice del Generic Device: reset segnalazioni FAULTS	Errore. Il segnalibro non è definito.
fig. 60 modulo: simulazione_plant	Errore. Il segnalibro non è definito.
fig. 61 objects 'Visualizations'	Errore. Il segnalibro non è definito.
fig. 62 impianto in attesa di RESET	Errore. Il segnalibro non è definito.
fig. 63 impianto in attesa di START (inizializzazione completata)	Errore. Il segnalibro non è definito.

BIBLIOGRAFIA

- [1 A. D. Luca, «Processi Industriali,» La Sapienza - Roma, 2017.
]
- [2 V. P. Castelli, «Automazione Industriale,» Bologna.
]
- [3 P. G. Dini, «Sistemi di Produzione,» slide 9, Università di Pisa,
] 2011.
- [4 C. D. R. "E.Piaggio", «Robotica e Automazione,» slide 9.
]
- [5 M. Sartini, «manuale codesys,» 10 2 2009. [Online]. Available:
] <http://www-lar.deis.unibo.it/people/msartini/Files/ITSC/Codesys/ManualeCodesys.pdf>. [Consultato il giorno 22 01 2019].
- [6 I. s. -. metalmeccanica, 27 12 2011. [Online]. Available:
] <http://www.sabbatiniconsulting.com/DOCUMENTAZIONE/DOC1/05%20Macchine/Dispositivi%20di%20comando.pdf>. [Consultato il giorno 10 1 2018].
- [7 p. sicuro, «sicurezza della macchina: i dispositivi di comando,» 27
] 12 2011. [Online]. Available: <https://www.puntosicuro.it/sicurezza-sul-lavoro-C-1/tipologie-di-rischio-C-5/attrezzature-macchine-C-45/la-sicurezza-delle-macchine-i-dispositivi-di-interblocco-AR-17197/>. [Consultato il giorno 10 12 2018].
- [8 N. c. g. 2018, «pag.18,» 2018. [Online]. Available:
] file:///C:/Users/Lorenzo/Downloads/NICOLETTI_catalogo_generale_2018_light.pdf. [Consultato il giorno 12 2018].

- [9 N. c. generale, «pag.37,» 2018. [Online]. Available:
] https://www.meccanicoletti.com/modules/lofdownload/view.php?id_download=2. [Consultato il giorno 12 2018].
- [1 Nicoletti, «Automat - 300,» Meccanica Nicoletti, [Online].
0] Available:
https://www.meccanicoletti.com/category.php?id_category=18.
- [1 G. Bandinelli, «Avvolgitrice - Svolgitrice "Elecktra",» 11 12 2013.
1] [Online]. Available:
<https://www.youtube.com/watch?v=bRu46zNDeKo>.
- [1 G. Automation, 12 2012. [Online]. Available:
2] https://www.youtube.com/watch?v=YS5somm_XwU. [Consultato il
giorno 10 2018].
- [1 d. channel, «Patreider, macchine speciali per automazione
3] industriale,» 5 10 2011. [Online]. Available:
https://www.youtube.com/watch?v=lZG_kpEWd2I. [Consultato il
giorno 6 10 2018].
- [1 bricomano-Bologna, «cavi elettrici,» [Online]. Available:
4] <https://www.bricoman.it/n/elettricit%C3%A0-illuminazione/cavi-elettrici/cavi-elettrici/>.