

UNIVERSITÀ DEGLI STUDI DI PADOVA  
FACOLTÀ DI INGEGNERIA



*Finito di scrivere il giorno 30 marzo 2010 utilizzando LyX*

UNIVERSITÀ DEGLI STUDI DI PADOVA  
FACOLTÀ DI INGEGNERIA

—  
DIPARTIMENTO DI INNOVAZIONE MECCANICA E GESTIONALE

—  
TESI DI LAUREA SPECIALISTICA IN INGEGNERIA  
DELL'AUTOMAZIONE

CELLA DI LAVORO PER  
ASSEMBLAGGIO DI BATTERIE  
A LETTATE

RELATORE: CH.MO PROF. ALDO ROSSI

LAUREANDO: DAMIANO VANTI

ANNO ACCADEMICO 2009-2010



*Alla mia famiglia ...*



*Quelli che s'innamorano di pratica senza scienza son come il nocchiere,  
che entra in naviglio senza timone o bussola,  
che mai ha certezza su dove si vada.*

LEONARDO DA VINCI

*Quelli che s'innamorano di scienza senza pratica son come un prigioniero,  
che entra in naviglio legato e bendato,  
che mai ha certezza se si vada.*

DAMIANO VANTI





# Indice

<b>Sommario</b>	<b>XI</b>
<b>Introduzione</b>	<b>XIII</b>
<b>1 Il progetto di una cella automatizzata</b>	<b>1</b>
1.1 Lo scambiatore di calore . . . . .	1
1.2 L'attuale linea di produzione . . . . .	6
1.3 La cella in progettazione: linee guida . . . . .	7
1.3.1 Tipologie di scambiatori e limiti progettuali . . . . .	8
1.4 Obiettivi del progetto . . . . .	10
<b>2 Inizializzazione della cella</b>	<b>11</b>
2.1 Il sistema di Comunicazione . . . . .	11
2.2 Layout della cella robotizzata . . . . .	16
2.2.1 Schema Elettrico . . . . .	24
2.3 Calibrazione della telecamera . . . . .	28
2.4 Calibrazione delle pinze . . . . .	31
2.5 Interfaccia di calibrazione della cella . . . . .	33
2.5.1 GUI principale di calibrazione della cella . . . . .	33
2.5.2 GUI di calibrazione della telecamera . . . . .	35
2.5.3 GUI di calibrazione delle pinze . . . . .	37
2.5.4 GUI di settaggio dei parametri di acquisizione ed elaborazione	39
<b>3 Software di gestione della cella</b>	<b>43</b>
3.1 Interfaccia Principale . . . . .	43
3.1.1 DataBase . . . . .	45
3.1.2 Ordine di Lavoro . . . . .	49
3.1.3 Set Parametri Difetti . . . . .	53
3.2 Ciclo Principale . . . . .	57
3.2.1 Calcolo percorso ottimo . . . . .	60
3.2.2 Gestione delle Matrici Difetti . . . . .	63
<b>4 Il sistema di Visione</b>	<b>65</b>
4.1 Il sistema di illuminazione . . . . .	69
4.1.1 Identificazione dello Scambiatore . . . . .	70
4.1.2 Analisi delle Svasature . . . . .	72
4.1.3 Analisi di eventuali Curvette già inserite . . . . .	74
4.2 Strumenti per l'elaborazione immagini con MatLab® . . . . .	75
4.3 Elaborazione immagini con MatLab . . . . .	78
4.3.1 Elaborazione per l'identificazione dello Scambiatore . . . . .	79

4.3.2	Elaborazione per l'identificazione dei difetti . . . . .	87
4.3.2.1	Taglio sulla svasatura . . . . .	87
4.3.2.2	Bicchiere irregolare . . . . .	91
4.3.2.3	Svasatura insufficiente . . . . .	93
<b>5</b>	<b>Test Sperimentali</b>	<b>97</b>
5.1	Prototipo della cella . . . . .	97
5.2	Ciclo di Inserimento Curvetta . . . . .	100
5.3	Test di assemblaggio . . . . .	107
	<b>Conclusioni e Sviluppi Futuri</b>	<b>109</b>
	<b>Ringraziamenti</b>	<b>117</b>

# Sommario

Questa tesi si inserisce all'interno di un progetto finalizzato alla realizzazione di una cella robotizzata per l'assemblaggio di scambiatori di calore a pacco alettato in sviluppo presso il DIMEG (*Dipartimento di Innovazione Meccanica e Gestionale*) dell'Università di Padova.

Il progetto prevede lo studio di fattibilità e la prototipazione della cella automatizzata, con particolare riferimento al montaggio delle curve di chiusura del circuito idraulico sulla sommità dei tubi del pacco alettato seguendo percorsi ottimali che minimizzino il tempo di esecuzione del ciclo complessivo. La cella comprende un sistema di visione artificiale atto in un primo momento all'identificazione dello scambiatore posto nell'area di lavoro e successivamente utilizzato per l'individuazione dei centri-tubo e la rilevazione di eventuali difetti di mandrinatura, un robot SCARA preposto all'assemblaggio del pezzo e un software di gestione della cella e della programmazione della lavorazione (gestione ordinativi).

L'attività di studio svolta si è concentrata principalmente nel miglioramento e ottimizzazione della cella di lavoro e del software di gestione della stessa, comprendendo una vasta serie di test in laboratorio sul prototipo di cella costruito che hanno portato alla modifica di alcune parti della pinza atta alla manipolazione delle curvette e alla modifica del codice che gestisce le diverse fasi dell'assemblaggio.



# Introduzione

Il progetto che d'ora in avanti sarà descritto è nato dalla volontà di ridurre l'intervento dell'uomo all'interno di un processo industriale che prevedeva l'assemblamento di scambiatori di calore a pacco alettato<sup>1</sup>, in particolare riguardo l'inserimento delle curve di collegamento tra i tubi nella parte superiore dello scambiatore.



Figura 1: Scambiatore di Calore

Tale riduzione è stata dettata dalla ripetitività dell'operazione da svolgere (e quindi da un'alta probabilità di errore umano) e dall'evoluzione dei sistemi di visione e degli algoritmi di elaborazione dell'immagine che hanno permesso ad una lavorazione ritenuta finora esclusivamente manuale di diventare oggetto di studio per un assemblaggio automatizzato il quale, una volta identificato correttamente la posizione dello scambiatore, provveda all'inserimento delle curve secondo uno schema prestabilito.

---

<sup>1</sup>Come si vedrà meglio in seguito, per scambiatore di calore a pacco alettato intendiamo una batteria di circuiti tubolari che, inserita in un pacco di alette sovrapposte, consente di realizzare lo scambio termico tra un fluido gassoso esterno ed un fluido primario interno. Tale prodotto trova naturale applicazione in campo civile e industriale all'interno di molti sistemi di raffreddamento o riscaldamento (condizionatori, chiller, ventilconvettori).

Questa tesi di laurea specialistica in Ingegneria dell'Automazione ha visto l'analisi e l'ottimizzazione dell'intero progetto di automazione, partendo dalla realizzazione dello schema elettrico che alimenta e controlla la cella di lavoro fino alla ri-elaborazione degli algoritmi di visione; per una trattazione completa e schematica si è diviso il lavoro svolto in 4 capitoli qui in seguito brevemente descritti:

Il **primo capitolo** è introduttivo al lavoro svolto e presenta le specifiche del progetto nella sua globalità. In particolare vengono analizzate le caratteristiche tipiche degli scambiatori di calore prodotti dall'azienda committente e descritta l'attuale linea di produzione. Viene poi presentato il progetto della cella e del circuito elettrico atto alla sua alimentazione e alla gestione delle emergenze.

Il **secondo capitolo**, dopo aver esposto brevemente i fondamenti della programmazione del robot SCARA della Adept e l'architettura di comunicazione tra il controllore del robot e il pc con il software di gestione della cella, analizzerà la calibrazione della cella di lavoro e degli End-Effector.

Con il **terzo capitolo** si propone di descrivere il software di gestione nel suo insieme, dall'interfaccia grafica al database di importazione dei disegni CAD, dalla gestione degli ordini alla strategia di inserimento delle curvette con relativa pianificazione del percorso ottimo per la minimizzazione del tempo ciclo. In questo capitolo si andrà anche ad illustrare il principio dell'intero ciclo di produzione, dall'avvio dell'automazione fino all'arresto dell'impianto, con particolare riferimento alla movimentazione del robot e alla sequenza degli innesti, alla sostituzione degli end-effector in funzione dello schema e alla gestione degli eventi, allarmi ed emergenze

Il **quarto capitolo** infine descrive in modo dettagliato tutta la parte inerente alla visione, in particolare verranno analizzate le procedure e gli algoritmi per l'identificazione degli scambiatori, l'identificazione dei centri tubo e l'analisi delle svasature per un corretto inserimento delle curvette.

# Capitolo 1

## Il progetto di una cella automatizzata

La tendenza del mercato negli ultimi decenni ha evidenziato un proficuo studio nella ricerca di nuove tecniche e tecnologie che permettessero l'automatizzazione di processi produttivi, molti sono infatti gli elementi che spingono alla realizzazione di un impianto automatizzato: va senz'altro citata l'elevata produttività che i sistemi automatici garantiscono, un impianto di assemblaggio automatico permette di lavorare su più turni senza necessità di pause o periodi di malattia che invece sono caratterizzanti una linea di assemblaggio manuale; senza contare gli oneri fiscali che derivano dall'utilizzo di manodopera<sup>1</sup>. Anche gli effetti sulla manodopera risultano un aspetto notevole, in quanto l'utilizzo di sistemi di assemblaggio automatico permette di prelevare personale da quelle attività ripetitive e faticose per destinarlo a mansioni maggiormente valorizzanti.

D'altra parte sarà necessaria una opportuna programmazione degli approvvigionamenti dei componenti che alimentano l'impianto e una gestione dei fermi macchina per guasti senza contare la formazione di tecnici specializzati e l'attività di manutenzione che gli impianti automatizzati necessitano.

E' chiaro che una soluzione generale è impossibile, infatti la scelta tra una linea di produzione manuale o automatizzata è determinata dal prodotto realizzato e dalle condizioni al contorno (luogo di produzione, disponibilità di manodopera/tecnici, agevolazioni statali, ecc..). Un'analisi di questo tipo è stata fatta anche per il progetto in analisi (vedi [7] [8]), la quale ha "autorizzato" l'avvio dello studio e della realizzazione dell'automazione oggetto di questa tesi.

### 1.1 Lo scambiatore di calore

Il progetto che analizzeremo d'ora in avanti nasce all'interno di un contratto di collaborazione tra Università degli Studi di Padova ed un'azienda che si propone nel mercato con termoconvettori e scambiatori di calore.

---

<sup>1</sup>Si pensi ad esempio che il costo della manodopera per un'impresa italiana risulta spesso il doppio di ciò che il lavoratore percepisce.

Lo scambiatore di calore (Figura 1.1) è utilizzato in tutte le applicazioni civili ed industriali in cui è richiesto lo scambio termico tra aria e fluidi primari come l'acqua, gas frigoriferi e soluzioni incongelabili ed è realizzato con tubi di rame ed alette continue di alluminio o di rame. Il fluido primario scorre all'interno dei tubi di rame che costituiscono un circuito <sup>2</sup>, mentre l'aria, da riscaldare o da raffreddare, fluisce esternamente ai tubi in direzione ortogonale ad essi e attraverso il pacco alettato.

Le alette sono munite di collari ed il contatto tra questi ultimi ed i tubi è ottenuto mediante l'espansione meccanica dei tubi stessi. L'aderenza così ottenuta garantisce il massimo scambio termico tra il fluido primario e l'aria. I collari delle alette inoltre sono muniti di un bordo di altezza calibrata, il quale consente di mantenere una rigorosa spaziatura tra un'aletta e l'altra.

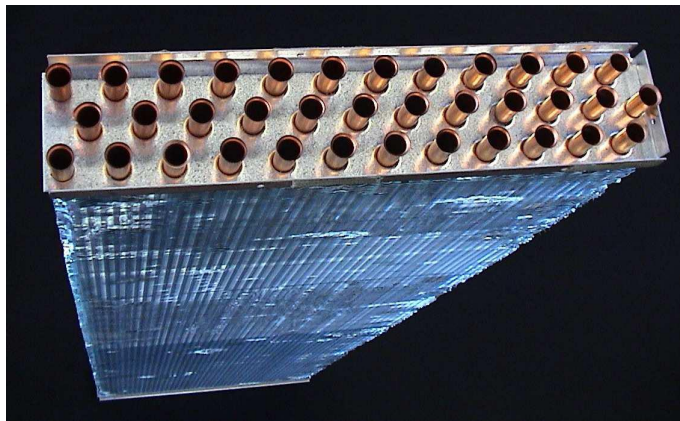


Figura 1.1: Scambiatore di calore.

Il circuito di tubi di rame inserito nel pacco alettato viene realizzato utilizzando le cosiddette "forcine" (Figura 1.2), che non sono altro che tubi di rame piegati a "U" tramite un processo automatizzato.



Figura 1.2: Forcine.

La forcina, lunga quanto lo scambiatore, viene inserita attraverso il pacco alettato e la piastra soprastante secondo schemi prestabiliti. Le forcine di un singolo circuito così inserite nel pacco vanno a costituire sulla superficie della piastra dello scambiatore un determinato **schema-tubi**, caratterizzato dal relativo numero,

<sup>2</sup>Questo circuito è prestabilito e studiato al fine di avere il massimo scambio termico tra il fluido all'interno del percorso e il fluido all'esterno del percorso.



diametro dei tubi e da una ben precisa geometria dei fori di inserimento delle forcine nel pacco.

Gli schemi-tubi presi in considerazione in questo studio sono costruiti con tubi di rame aventi il diametro esterno di  $9.52\text{mm}$  ( $3/8''$ ) oppure  $7.94\text{mm}$  ( $5/16''$ ) e gli interassi tra i tubi sono descritti dalle seguenti due geometrie (Figura 1.3):  $25\text{mm} \times 21.65\text{mm}$  (**geometria equilatera**) e  $25\text{mm} \times 19\text{mm}$  (**geometria non equilatera**). Nello scambiatore visto dall'alto é possibile distinguere un ordinamento dei tubi per righe, i cosiddetti ranghi; ad esempio lo scambiatore visibile nella figura 1.1 presenta uno schema-tubi suddiviso in tre ranghi.

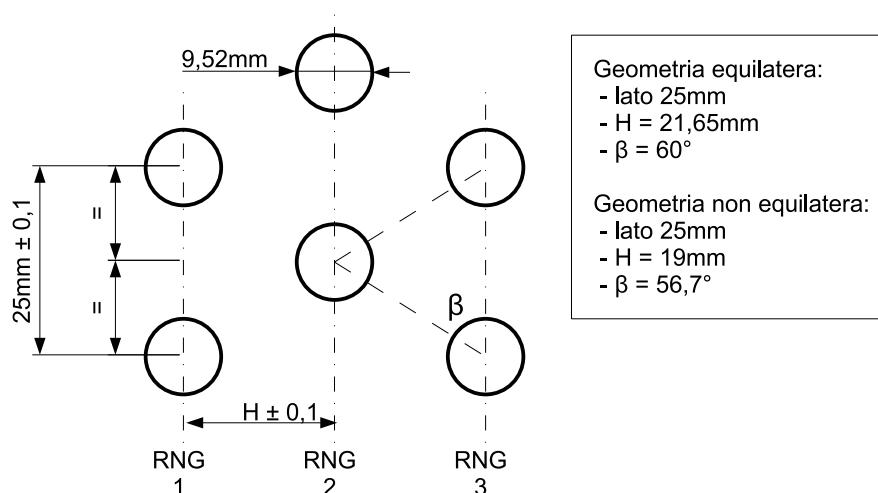


Figura 1.3: Geometria triangolare dei tubi.

Una volta inserite le forcine nel pacco alettato, per chiudere il circuito vengono utilizzate le curve. La curva (Figura 1.4) é un tratto di tubo di rame piegato a "C" avente due tratti rettilinei alle estremità e di diametro esterno coincidente con quello delle forcine pre-mandrinata. Questo oggetto funge da collegamento tra una forcina e l'altra e rappresenta proprio la parte di tubo che chiude il circuito; diversi collegamenti permettono di creare svariati circuiti chiusi che possono avere un ingresso e un'uscita oppure più ingressi e più uscite, a seconda dei tubi lasciati aperti.



Figura 1.4: Curve con e senza anello di saldatura.

Le curve possono essere provviste o meno di un anello di rame abbracciato attorno alle due estremità. L'anello, se presente, serve per gli scambiatori che vengono saldati in maniera automatizzata: l'anello infatti costituisce il materiale d'apporto per la saldo-brasatura che riscaldato andrà a collassare sul bicchiere del tubo connesso ( Figura 1.5). Le curve presentano una superficie liscia e completamente a sezione circolare, quindi molto difficile da manipolare.



Figura 1.5: Operazione di saldobrasatura delle curve.

Le curve possono avere diversi diametri e diversi interassi. Quelle che vengono prese in considerazione in questo studio sono di soli due tipi:

- interasse =  $25mm$  (utilizzabile in tutte le geometrie) con o senza anello
- interasse =  $22.743mm$  (solo per geometria  $25mm \times 19mm$ ) con o senza anello

Lo schema per mezzo del quale vengono posizionate le curve che collegano una forcina all'altra é presente nel disegno CAD (Figura 1.6) dello specifico scambiatore di calore. Ogni scambiatore di calore possiede i propri disegni costruttivi e i disegni contenenti gli schemi di montaggio delle forcine e delle curve (disegni CAD del tipo ST).

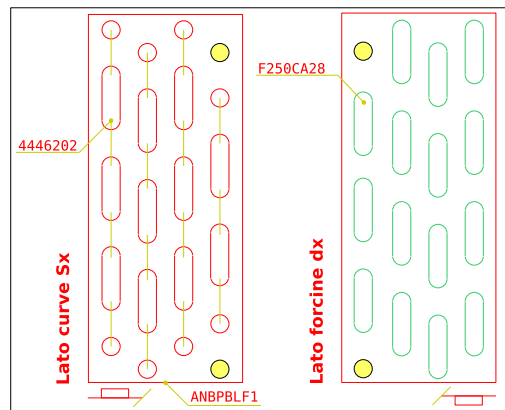


Figura 1.6: Disegno CAD di uno schema tubi a 4 ranghi.

Il bicchiere (Figura 1.7), rappresenta la sede di inserimento della curva nella forcina. Esso viene creato per mandrinatura. Sulla sommità del bicchiere viene realizzata una svasatura sul bordo con un duplice scopo: facilitare l'inserimento delle curve e soprattutto creare una zona di accumulo del materiale di apporto per la saldobrasatura delle curve sui bicchieri. La presenza della svasatura é molto utile in quanto costituisce un invito per il movimento di inserimento della curva.

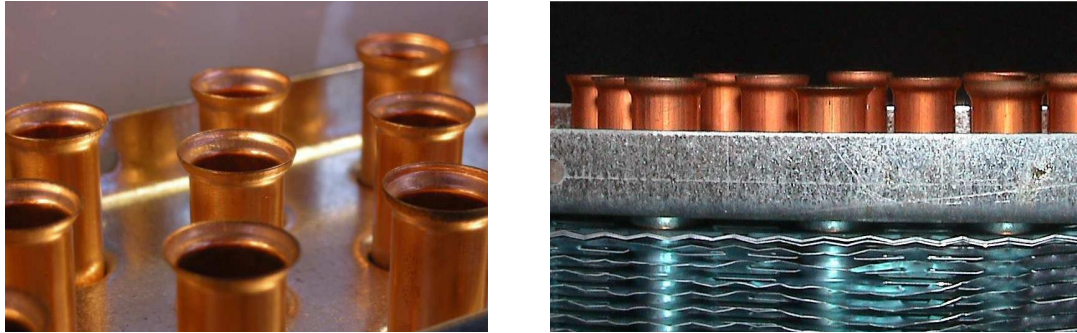


Figura 1.7: Bicchieri.

L'operazione di mandrinatura è però responsabile di numerose tipologie di difetti accidentali. In primo luogo la flessione della barra, proporzionale alla sua lunghezza, durante l'inserimento all'interno del tubo può causare un allargamento del bicchiere del tubo non uniforme correlato ad una conseguente svasatura non uniforme. Dall'altro lato una variazione di altezza delle forcine, o della loro sporgenza dalla piastra, può significare una svasatura più o meno accentuata e nel primo caso, stressando eccessivamente l'imboccatura in rame, può portarla ad aprirsi generando dei tagli sulla svasatura (Figura 1.8). L'entità delle variazioni di altezza dei bicchieri è nell'intorno dei  $2.5mm \div 4mm$ , a differenza dei disegni costruttivi indicanti tolleranze di  $\pm 1mm$  su tutte le quote del bicchiere, compresa la sporgenza dalla piastra. Questa problema non influenza gli inserimenti, tanto quanto le altre tipologie di difetti.



Figura 1.8: Taglio sulla svasatura di un bicchiere.

Si rende inoltre necessario considerare la possibilità che la mandrinatura avvenga correttamente entro i valori previsti ma che successivamente le imboccature delle batterie, particolarmente delicate visto lo spessore del rame impiegato, vengano danneggiate da colpi accidentali durante la manipolazione o il trasporto.

Tutte le considerazioni avanzate sono realmente importanti, dal momento che dalla qualità dell'operazione di mandrinatura dipende il successo della seguente operazione di inserimento delle curve. Le diverse tipologie di difetti e le relative caratteristiche saranno discusse in modo più approfondito nei prossimi capitoli.

## 1.2 L'attuale linea di produzione

La catena di produzione di uno scambiatore di calore a pacco alettato nella quale il progetto di automazione andrà ad inserirsi può essere rappresentata dallo schema di figura 1.9, in cui sono distinte le varie lavorazioni in base all'ordine di esecuzione, ai loro input ed output. La fase di lavorazione oggetto di questo studio è stata evidenziata al centro del diagramma.

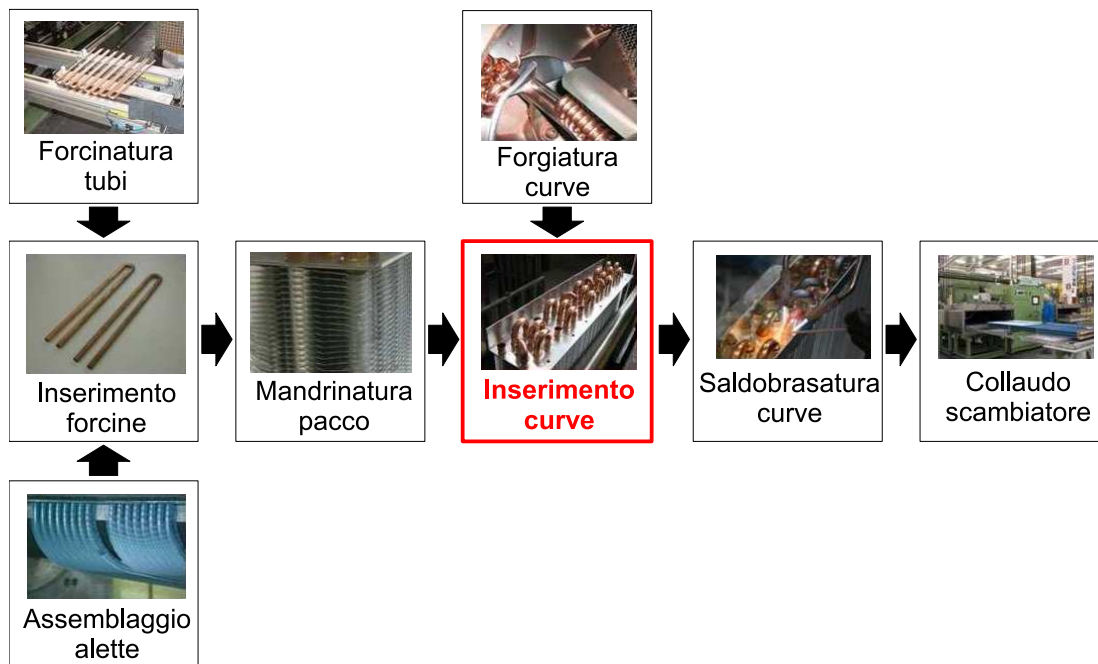


Figura 1.9: Ciclo di produzione di uno scambiatore di calore.

Allo stato attuale la procedura di inserimento delle curve avviene in forma manuale. Un operatore addetto alla catena di montaggio sistema lo scambiatore<sup>3</sup> su di un nastro trasportatore fissandolo alla base dal lato delle forcine. Tale nastro conduce molto lentamente i pezzi verso l'area dove avviene la successiva brasatura; in questo lasso di tempo un altro operatore, curve alla mano, procede al loro inserimento nella parte superiore secondo lo schema di circuitazione prefissato dopo di che, concluso l'inserimento, ribatte ogni curva nella sua sede con un martello di gomma per migliorare l'accoppiamento con i tubi. A questo punto un'altro scambiatore può essere posizionato sul nastro e l'operazione si ripete.

<sup>3</sup>In realtà lo scambiatore che più frequentemente si andrà a lavorare è formato da due schemi-tubi opportunamente accoppiati tramite rivettatura dall'operatore sopraccitato.

Una lavorazione di questo tipo ha ovviamente dei limiti e degli inconvenienti:

1. bassa velocità dell'operazione di montaggio;
2. necessaria separazione delle operazioni di inserimento e battitura;
3. possibilità di errori nell'inserimento;
4. rilevamento impreciso dei possibili difetti delle imboccature;
5. determina un carico di lavoro ripetitivo per l'operatore, dannoso sia per esso che per il processo (maggiore probabilità d'errore);

Per quanto riguarda il primo punto è stato valutato sperimentalmente che in media il tempo impiegato da un'operatore che sappia esattamente dove posizionare la curva è di circa 2 secondi. Chiaramente per definire il tempo-curva, dato dal rapporto tra il tempo totale di lavorazione e il numero di curve da inserire, deve essere sommato al tempo di inserimento il tempo per la battitura. Pertanto la velocità complessiva dell'operazione risulta scarsa. Possiamo poi ipotizzare che un'operatore allenato ed esperto sappia sempre mettere le curve nelle sedi corrispondenti allo schema di circuitazione previsto, tuttavia la probabilità di un errore esiste ed è diversa da zero. Va invece a favore dell'operatore la possibilità di manipolare con facilità curve di dimensioni e diametro del tubo diverse. I possibili difetti alle imboccature dei tubi, che come si è detto rappresentano un frequente risultato dell'operazione di mandrinatura, sono sicuramente identificabili sommariamente dall'operatore (peraltro già impegnato a montare le curve) ma non qualitativamente. Un difetto grave può infatti rendere vana la brasatura prefigurando una perdita nel circuito. Infine c'è da considerare l'importante fattore umano: una persona chiamata a svolgere un'operazione manuale ripetitiva, quindi priva di stimoli, è soggetta ad un aumento dello stress da lavoro. Tutte queste valutazioni vanno a favore della sostituzione dell'attuale modalità di lavorazione con una automatizzata, realizzata da una cella composta da un sistema di manipolazione delle curve (il braccio dell'operatore) e un sistema di visione artificiale che guidi il loro inserimento (l'occhio e il cervello).

### **1.3 La cella in progettazione: linee guida**

Lo scopo dell'intero progetto, come già accennato, è la realizzazione di una completa cella di lavoro robotizzata in grado di inserire autonomamente e intelligentemente le curve nello scambiatore, finalizzandone la struttura e le funzionalità definitive. Questa tesi si è inserita nella fase di sviluppo intermedio dell'intero progetto, il quale si è articolato tra studio iniziale, prototipazione, sviluppo software e visione fino alle sperimentazioni e alla posa in opera.

Tale cella robotizzata è rappresentabile come un sistema che ha come input il prodotto da assemblare corredato dei relativi disegni CAD contenenti le specifiche di assemblaggio, e come output, il prodotto assemblato con una serie di dati relativi all'operazione di assemblaggio effettuato (Figura 1.10). Tra questi dati, ad esempio, vi sono i dati statistici, come il numero di pezzi assemblati e non

assemblati a causa di difetti rilevati dal sistema di visione, quali e quanti di questi difetti sono stati rilevati, e la gestione degli ordini.

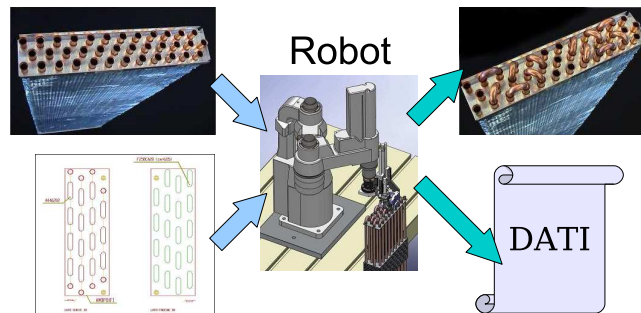


Figura 1.10: Schema di Input-Output della cella robotizzata.

### 1.3.1 Tipologie di scambiatori e limiti progettuali

I vari modelli di scambiatori prodotti dall'azienda che ha commissionato il progetto, trovando impiego in moltissime applicazioni di dimensioni e condizioni d'esercizio eterogenee, spesso realizzati secondo specifiche richieste del cliente, differiscono notevolmente tra loro per una serie di fattori quali:

- dimensioni (altezza, lunghezza, larghezza) dello scambiatore
- geometria dei fori nelle piastre
- diametro dei tubi
- numero dei tubi e dei ranghi
- numero di circuiti indipendenti
- tipo di funzionamento (fluidi)
- materiali di costruzione e trattamenti
- tipo di tubi (rigati, lisci o rigati elettrosaldati)

Attualmente la ditta committente del progetto necessita con una certa priorità di una cella di lavoro automatizzata per l'assemblaggio di un particolare modello di scambiatore (Figura 1.11). Questo particolare modello è composto da due distinti modelli di schema-tubi affiancati denominati S5/16 e S3/8; uniti in un'unica piastra di dimensioni  $170mm \times 290mm$  priva di spalle e le cui caratteristiche principali sono riassunte nella tabella 1.1. Per comodità questo modello di scambiatore lo chiameremo ST30181+S3/8.

Si è così deciso di sviluppare il progetto innanzitutto sulla base di questo particolare modello di scambiatore, cercando però di privilegiare quanto più possibile quelle scelte costruttive che consentano l'utilizzo della cella anche per l'assemblaggio di qualsiasi altro modello, o dovendo al più introdurre qualche minima modifica alla struttura della cella.

Modello	S5/16	S3/8
Tipo	non equilatero	equilatero
Numero tubi	41	41
Diametro tubi	7.94mm (5/16")	9.52mm (3/8")
Numero ranghi	6	6
Numero curve 25mm	6	20
Numero curve 22.743mm	14	0
Area dello schema	170mm × 110mm	170mm × 130mm

Tabella 1.1: Caratteristiche degli schemi-tubi modello S5/16 e S3/8.

Possiamo nel complesso restringere l'insieme di tali fattori a quelli le cui variazioni hanno rilevanza all'interno del progetto di cella: ovvero dimensioni, geometria delle piastre, numero di tubi e ranghi e diametro dei tubi. Le dimensioni degli scambiatori sono ampiamente variabili. Vengono prodotti modelli piccoli e stretti, ma anche scambiatori il cui ordine di grandezza in altezza e lunghezza è il metro, con numerosi circuiti e centinaia di tubi all'interno del pacco alettato. Considerando i numeri di produzione delle varie tipologie di batteria e coniugando valutazioni di ingombri e costi della cella con l'esigenza di renderla funzionale per il maggior numero di scambiatori possibile, risulta necessario operare un compromesso e porre un limite superiore alle dimensioni delle unità lavorabili all'interno della cella. Il limite in lunghezza stabilito è di 750mm mentre per quanto riguarda la larghezza, lo scambiatore più largo finora visto risulta appunto lo scambiatore S5/16+S3/8 di figura 1.11.



Figura 1.11: Scambiatore S5/16+S3/8.

In presenza di geometria equilatera si avranno nello schema-tubi solamente curve di interasse 25mm, orientate secondo le tre direzioni dei lati del triangolo equilatero. In presenza invece di geometria non equilatera si avranno curve ad interasse 25mm orientate lungo la direzione del lato corrispondente alla base del triangolo e curve ad interasse 22.743mm orientate secondo le direzioni date dai due lati uguali del triangolo isoscele. Un ulteriore dato di progetto è la

presenza di eventuali spalle sullo scambiatore (Figura 1.12). Queste spalle possono rappresentare un ingombro molto importante, che potrebbe limitare i movimenti del Robot nell'inserimento delle curve adiacenti alle spalle stesse o interferire con l'illuminazione del sistema di visione. Essa é costituita da una parte di lamiera che fuoriesce verticalmente dal piano orizzontale dello scambiatore. La sua altezza massima é di  $60mm$ , presa dalla piastra dalla quale fuoriescono i bicchieri. La spalla non sempre é presente ma può anche capitare che vi sia su tutti e quattro i lati della piastra.



Figura 1.12: Spalle in un modello di scambiatore.

## 1.4 Obiettivi del progetto

Il lavoro di tesi che verrà descritto nei capitoli successivi costituisce il proseguimento di una progetto che ha assorbito diverse tesi nell'arco degli ultimi due anni. Lo scopo di questa tesi è stato quello di unire e sistemare diverse parti di software implementate precedentemente, portando alla vera e propria costruzione di un prototipo di cella di lavoro e alla realizzazione dell'assemblaggio.

Si riassumono per punti le fasi principali di questo lavoro:

- definizione di uno schema elettrico e gestione delle emergenze
- miglioramento software relativo alla calibrazione della cella
- modifica degli algoritmi di visione, in particolare nell'identificazione del modello di scambiatore
- modifica del ciclo principale all'interno del software di gestione della cella, in particolare della strategia di assemblaggio e della gestione ordinativi
- modifica delle statistiche dei difetti riscontrati e della gestione allarmi verso la mandrinatrice
- ottimizzazione del codice presente sul controllore del Robot
- tests di assemblamento e robustezza del sistema realizzato.



# Capitolo 2

## Inizializzazione della cella

Il capitolo che andremo a sviluppare sarà fondato sullo start-up della cella di lavoro così da poter permettere, nei prossimi capitoli, la discussione sul ciclo principale del software, sul sistema di visione e sulla gestione degli ordini.

Cominceremo con l'introdurre il sistema di comunicazione tra il robot e il pc esterno, soffermandoci poi sulle informazioni di scambio e sul ciclo di funzionamento del codice presente nel controllore del robot. Al termine di questa parte sarà possibile trattare la calibrazione della telecamera e degli end-effector.

### 2.1 Il sistema di Comunicazione

La connessione tra il pc esterno (sul quale viene elaborato il codice di gestione della cella scritto in linguaggio Matlab) ed il controllore del robot si basa su una comune connessione Ethernet, che sfrutta il consolidato protocollo TCP/IP attraverso una gerarchia del tipo *Master/Slave*. La figura sottostante mostra molto vagamente il principio di comunicazione, principio che andremo immediatamente ad analizzare.

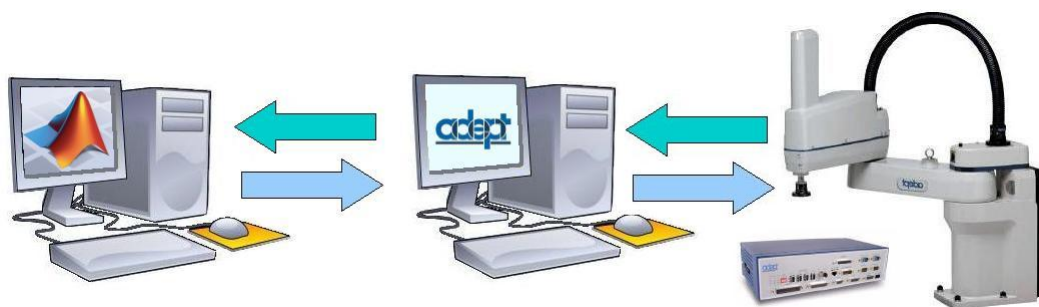


Figura 2.1: Schema di comunicazione tra robot e pc esterno.

All'inizio della giornata di lavoro vengono accesi, attraverso il quadro elettrico principale, sia il controllore del robot che il pc esterno. Entrambi questi sistemi sono impostati in modo che finita la fase interna di inizializzazione vengano caricati i rispettivi codici sviluppati (nel controllore basta che il programma sia salvato come *AUTO.V2* all'interno della RAM del controllore mentre per il pc basta inserire il file contenente il codice principale nella cartella *Esecuzione Automatica*).

Il codice presente nel pc esterno (vedi Capitolo 3) non richiede un immediato tentativo di connessione con il robot, bensì è necessario premere il pulsante *CONNECT* presente sull'interfaccia che appare sullo schermo (vedi Figura 2.2). Al contrario il codice presente sul controllore inizia subito a controllare la presenza su una porta prestabilita la presenza del pc esterno che andrà a costituire il *Master* all'interno della comunicazione (chiaramente il controllore è già impostato per occupare la posizione di *Slave*).

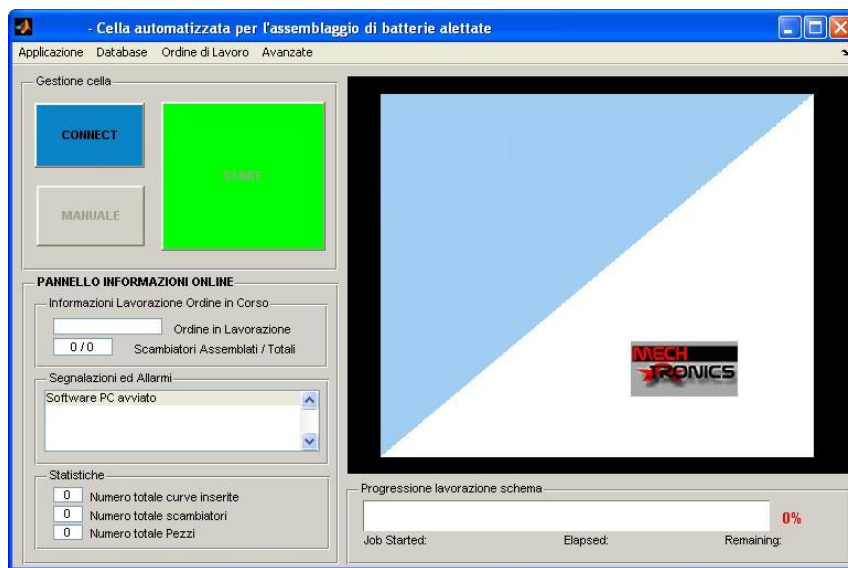


Figura 2.2: Interfaccia che appare all'avvio del pc-esterno

La gestione della comunicazione sul controllore è relativamente semplice, infatti appena instaurata la comunicazione il controllore si mette in ascolto della rete; ricevuto un pacchetto ne legge l'ID ed esegue le operazioni a cui corrisponde, al termine invia al pc esterno una notifica sull'esecuzione dell'operazione richiesta e si rimette in attesa di un nuovo pacchetto. Tale gestione può venire ben riassunta dallo schema in figura 2.3 dove si può vedere una fase di inizializzazione in cui vengono salvate come variabili globali diverse informazioni, ad esempio le porte relative ai diversi ingressi/uscite, tutti gli ID utili per la comunicazione con il pc esterno e tutte le locazioni (home e diversi magazzini pinza). Successivamente si entra nel ciclo principale del codice (dal quale vengono effettuate tutte le chiamate alle sub-routine) e si inizia con l'ascolto della rete per dare avvio alla comunicazione. Terminata questa fase si interpreta il pacchetto ricevuto dal pc esterno e a seconda dell'ID presente sull'header dello stesso si possono compiere diverse azioni (in figura 2.3 sono raggruppate nell'area di color viola) le cui principale sono riportate in figura.

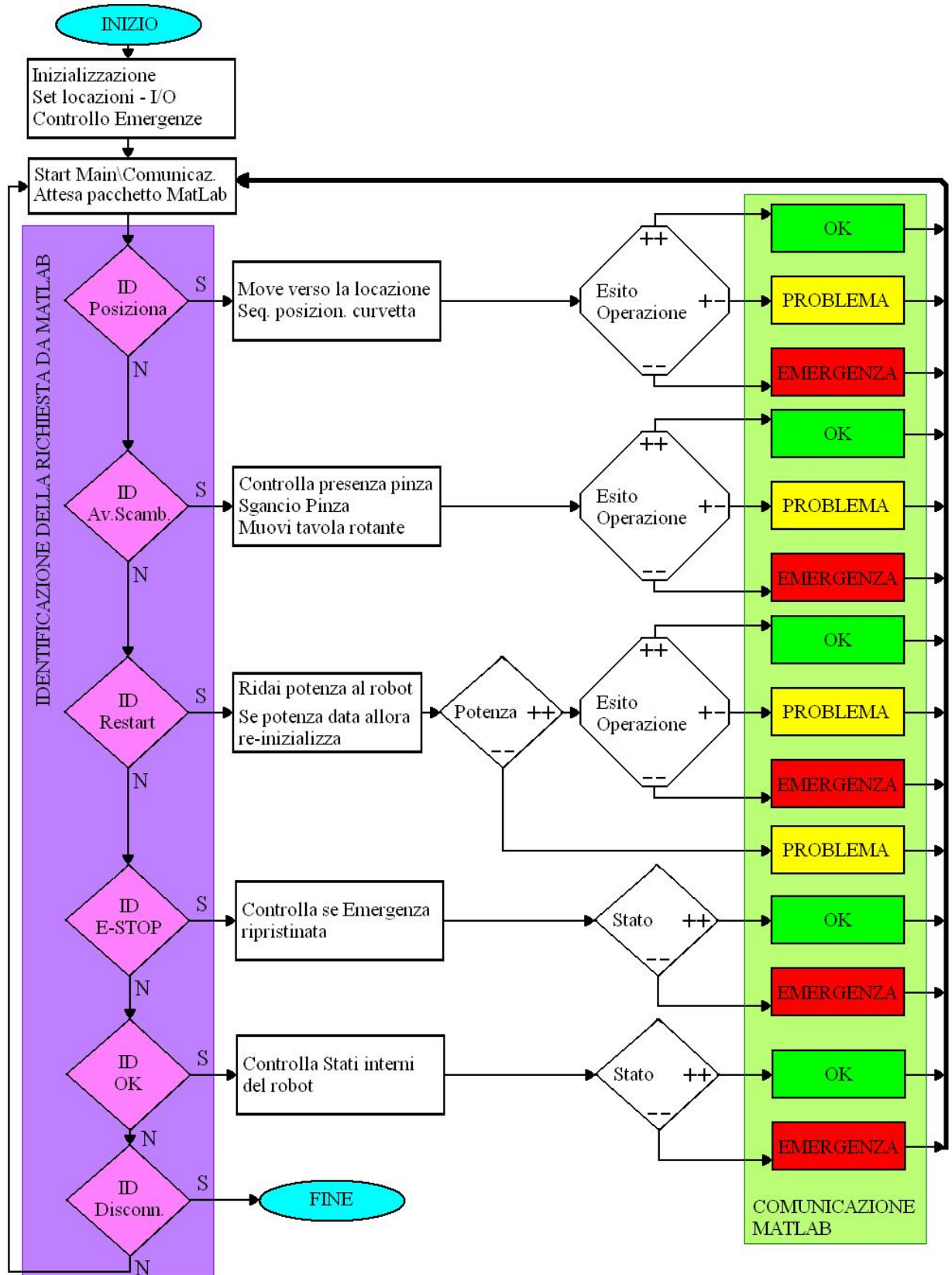


Figura 2.3: Ciclo di funzionamento del codice presente sul controllore del Robot

Con riferimento alla figura 2.3 analizziamo brevemente le operazioni che il pc esterno può richiedere al controllore del robot (sono state omesse delle operazioni meno importanti che non aiutano nella comprensione della comunicazione, ad esempio esiste un ID che ha associata una richiesta di accensione delle luci diffuse, ma in questo caso non si fa altro che attivare il segnale che comanda il relè relativo alle luci e dare la conferma di esecuzione della richiesta a MatLab). Il pacchetto con l'*ID Posiziona* richiede il vero è proprio posizionamento della testa del robot nella locazione presente nel pacchetto (il pacchetto è fatto di 8 byte, nel primo c'è l'ID e nei rimanenti c'è la locazione sopraccitata) con successiva procedura di inserimento della curvetta. Tale procedura è costituita da una sequenza ben determinata di operazioni (vedi Cap 1, End-Effector) le quali potrebbero generare diversi problemi (ad esempio un inceppamento della slitta o un bloccaggio del pistone principale, etc.). In tale situazione (rappresentata in figura 2.3 grazie all'ottagono "Esito Operazione" con la simbologia +- ) viene tolta sia la potenza al robot che la pressione all'interno del circuito pneumatico e viene inviato al pc esterno un pacchetto con un ID relativo al problema riscontrato. Se al contrario l'operazione viene eseguita perfettamente (rappresentata in figura 2.3 grazie all'ottagono "Esito Operazione" con la simbologia ++ ) allora viene inviato al pc esterno un pacchetto con un ID relativo al conseguimento con successo, infine se l'operazione genera un'emergenza (ad esempio generata da una collisione o dall'apertura durante l'operazione della protezione della cella, etc.) vengono anche in questa occasione tolte sia la potenza al robot che la pressione all'interno del circuito pneumatico e viene inviato al pc esterno un pacchetto con un ID relativo alla situazione di emergenza (*ID-ESTOP*).

L'operazione con l'*ID Avanza Scambiatore* non porta ad ulteriori informazioni che aiutino nella comprensione, per questo analizziamo il pacchetto con l'*ID Restart*. Questo pacchetto viene inviato dal pc esterno solo dopo il ripristino di una situazione di emergenza: inizialmente si tenta di ridare potenza al robot abilitando un componente esterno connesso al controllore chiamato Front Panel (vedi figura 2.4).

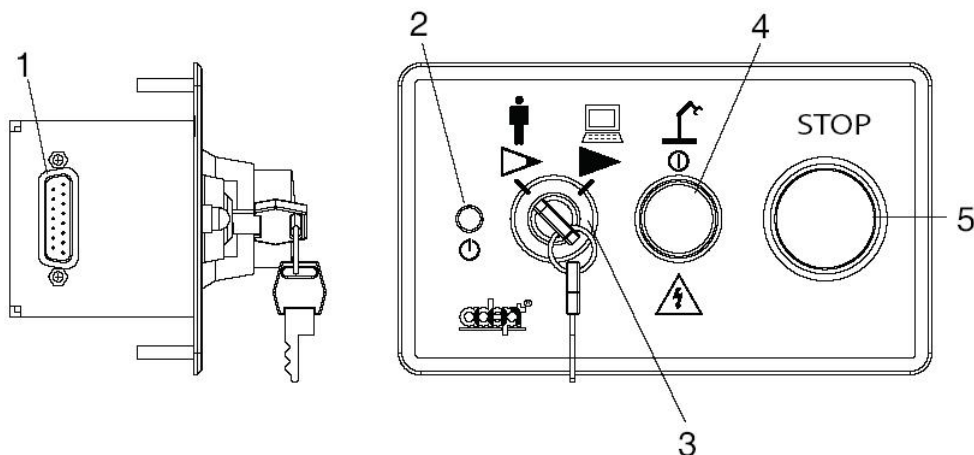


Figura 2.4: Front Panel

L'abilitazione del front panel si traduce con l'abilitazione di un pulsante dotato di una luce lampeggiante (indicata in figura 2.4 con il numero 4) che deve essere premuto entro un tempo stabilito. Se tale operazione viene effettuata con successo si procede alla successiva fase di re-inizializzazione in cui viene posata l'eventuale pinza agganciata alla flangia del robot, altrimenti si invia al pc esterno un pacchetto che si tradurrà nell'interfaccia grafica utente nel riproporre il pulsante per il Restart da software. Ipotizzando che l'operatore abbia premuto il pulsante nel tempo previsto si deve tener conto che nella fase di re-inizializzazione ci possono essere degli errori (ad esempio la sede assegnata alla pinza presente sulla flangia del robot è già occupata da un'altra pinza) i quali sono gestiti esattamente come fatto nel caso dell'*ID posiziona*.

Il pacchetto con l'*ID E-STOP* viene inviato dal pc esterno per dare conferma al robot che la situazione di emergenza (generata dallo stesso robot) è stata notificata all'utente (il controllore ed il pc continuano a mandarsi pacchetti con l'*ID E-STOP* finchè l'emergenza non viene ripristinata). Al ripristino dell'emergenza il robot invia un pacchetto con un *ID OK* al pc esterno il quale a sua volta risponde con un pacchetto con *ID OK*, tale invio di pacchetti continua esattamente come visto per l'emergenza fino a quando non intervenga una nuova emergenza oppure l'operatore richiama attraverso l'interfaccia software nuove operazioni (ad esempio premendo sull'interfaccia il pulsante RESTART).

Il pacchetto con l'*ID OK* viene inviato dal pc esterno con lo scopo di mantenere viva la comunicazione e di controllare lo stato del sistema, infatti se il robot per qualsiasi motivo va in emergenza il controllore risponde a questo pacchetto non più con un *ID OK* ma con un *ID E-STOP*.

E' bene far notare che qualsiasi sia la richiesta del pc esterno, se il controllore genera un'emergenza non viene svolta nessuna operazione di movimentazione e/o pneumatica (si ricordi che il lancio dell'emergenza provoca l'apertura dell'elettrovalvola principale del circuito pneumatico e l'interruzione dell'alimentazione al robot) e alla prima occasione viene inviato al pc esterno un pacchetto con *ID E-STOP*.

La gestione della comunicazione sul pc esterno sfrutta la libreria *pnet* fornita da Matlab, in particolare ogni volta che il codice presente sul pc esterno vuole far eseguire al controllore una certa operazione chiama la funzione *send.m* dando come parametri l'ID relativo alla operazione da eseguire.<sup>1</sup>La funzione *send.m* si mette in ascolto del controllore presente sulla rete e l'ID del pacchetto che riceverà potrà essere di due tipi: ID di emergenza o ID non di emergenza. Nel primo caso il pc esterno avvisa l'operatore del tipo di emergenza rilevata (calo di pressione, bloccaggio delle diverse componenti dell'end-effector, errori nel magazzino, etc.) e non invia al controllore il pacchetto con l'ID presente come parametro nella chiamata della funzione *send.m* ma invia un pacchetto con un ID che indica al controllore la presenza dello stato di emergenza; nel secondo caso vengono eseguite dal pc esterno le operazioni relative all'ID ricevuto e viene inviato al controllore il pacchetto con l'ID presente come parametro nella chiamata della funzione *send.m*.

---

<sup>1</sup>Tali ID sono salvati sia nel controllore che nel pc esterno nella rispettive fasi di inizializzazione.

## 2.2 Layout della cella robotizzata

La realizzazione della cella necessita innanzitutto lo studio delle operazioni che si andranno ad effettuare, operazioni che sono qui riassunte brevemente:

1. Introduzione dello scambiatore all'interno della cella.
2. Lo scambiatore entra nell'area di lavoro del robot dove un sistema di visione artificiale provvede all'acquisizione di due foto ottenute utilizzando due illuminazioni diverse: illuminazione diffusa e laterale (nel capitolo dedicato al sistema di visione si tratterà con dettaglio questa fase)
3. Dalle immagini acquisite viene identificato il modello dello scambiatore e vengono fornite al sistema robot le informazioni necessarie per l'assemblaggio delle curve. Un software di gestione della cella provvede al coordinamento delle operazioni di assemblaggio. L'assemblaggio delle curve, eseguito da un robot corredato di un apposito utensile di inserimento, avviene secondo una sequenza predeterminata. Nel caso si renda necessario un cambio di tipologia di curve da inserire, il sistema di gestione ordina al robot di effettuare il cambio utensile opportuno.
4. Al termine dell'operazione di assemblaggio viene verificato il corretto inserimento delle curve con una ulteriore acquisizione di immagini. Nel caso in cui venga rilevato un qualsiasi difetto che pregiudichi la qualità dello scambiatore, il pezzo viene segnalato all'operatore.

Il layout della cella realizzata (vedi [8] per una trattazione esaustiva) è riportato nella figura seguente:

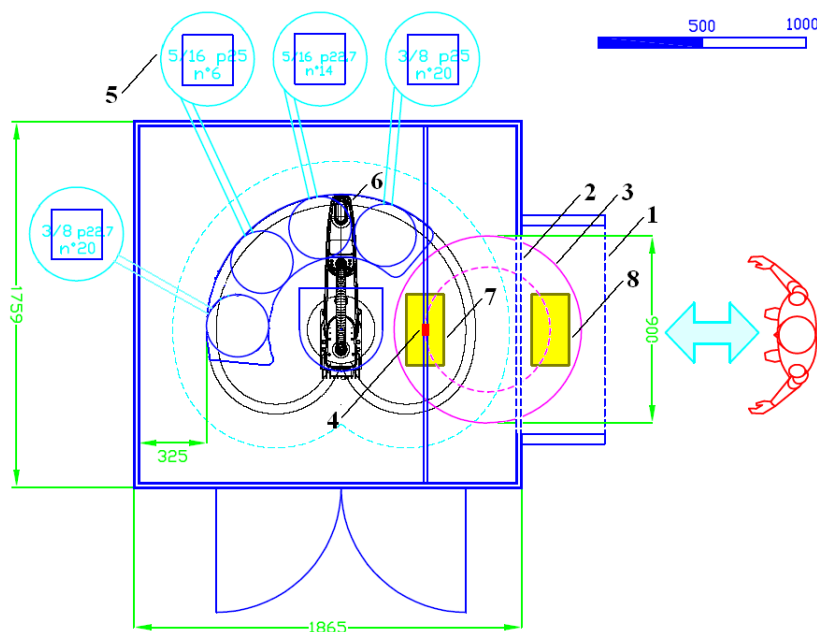


Figura 2.5: Layout della cella per assemblaggio automatizzato

Sono numerate le parti fondamentali che costituiscono la cella, in particolare abbiamo:

1. Barriera esterna: verrà utilizzata nel seguito per la gestione della movimentazione della tavola rotante, in particolare saranno posti dei sensori di presenza che indicheranno la possibilità o meno di muovere la tavola senza pericolo per l'operatore
2. Barriera interna: anch'essa dotata di sensori di presenza che indicano la presenza o meno dell'operatore all'interno dell'area riservata al robot, tale informazione sarà utile come vedremo tra breve per la gestione delle emergenze
3. Tavola rotante: comandata dal controllore del robot che permette di portare lo scambiatore inserito dall'operatore (in Figura indicato con 8) all'interno dell'area di lavoro del robot (in Figura indicato con 7)
4. Telecamera: permetterà l'acquisizione delle due immagini (una ottenuta con illuminazione diffusa e l'altra con illuminazione laterale) indispensabili per il funzionamento della cella
5. Moduli magazzino: contengono l'alloggiamento per le diverse pinze e sono direttamente connessi tramite una guida ai sistemi di alimentazione (vibroburatti)
6. Robot/End Effector: Robot SCARA a 4 gradi di libertà con annesso utensile realizzato specificatamente per maneggiare ed inserire le curvette che chiuderanno il circuito della scambiatore
7. Scambiatore in area di Lavoro: scambiatore posto nell'area di lavorazione, tale area si trova esattamente sotto il sistema di visione ed è inaccessibile all'uomo (a meno di causare uno stato di emergenza del sistema)
8. Scambiatore in area operatore: scambiatore posto nell'area accessibile all'operatore, in particolare qui avviene la prelevazione della scambiatore assemblato dal robot e il fissaggio di un nuovo scambiatore per il successivo assemblaggio

In seguito verranno brevemente descritte solo alcune delle parti sopra elencate, questo perchè le parti rimanenti avranno uno spazio a loro dedicato nel seguito della trattazione.

## Robot

La scelta del **robot** non è risultata problematica, dal momento che essendo richiesto all'*end-effector* un movimento di assemblaggio verticale dall'alto verso il basso, senza cambiamenti di orientamento, risulta sufficiente l'impiego di uno SCARA "*Selective Compliance Assembly Robot Arm*" a 4 assi, che porti la pinza sopra al punto di inserimento previsto.

Il modello di robot prescelto per l'inserimento nella cella di lavoro è un Adept Cobra s600 (Figura 2.6). I Robot Adept Cobra sono concepiti per interfacciarsi con un controller Adept MV ed il telaio di alimentazione PA-4. Il robot è collegato al suo controllore tramite cavi di segnali e di potenza. La sua programmazione e comando avviene tramite il controller Adept MV e il sistema di comando amplificatore PA-4, inglobato nel corpo centrale. Il controllore esegue un programma caricato all'avvio della cella interfacciandosi con il software di gestione sito in un pc esterno.



Figura 2.6: Robot SCARA Adept Cobra s600 con controllore Adept MV.

## End-Effector

Diverse tipologie di curvette necessitano obbligatoriamente diversi end-effector; lo studio fatto in [7] ha portato a realizzare un end-effector principale, grazie al quale il cambio del tipo di curvetta costringe a cambiare solo la parte terminale dell' end-effector, in particolare la parte costituita dalla culla che va ad avvolgere la curvetta da manipolare.



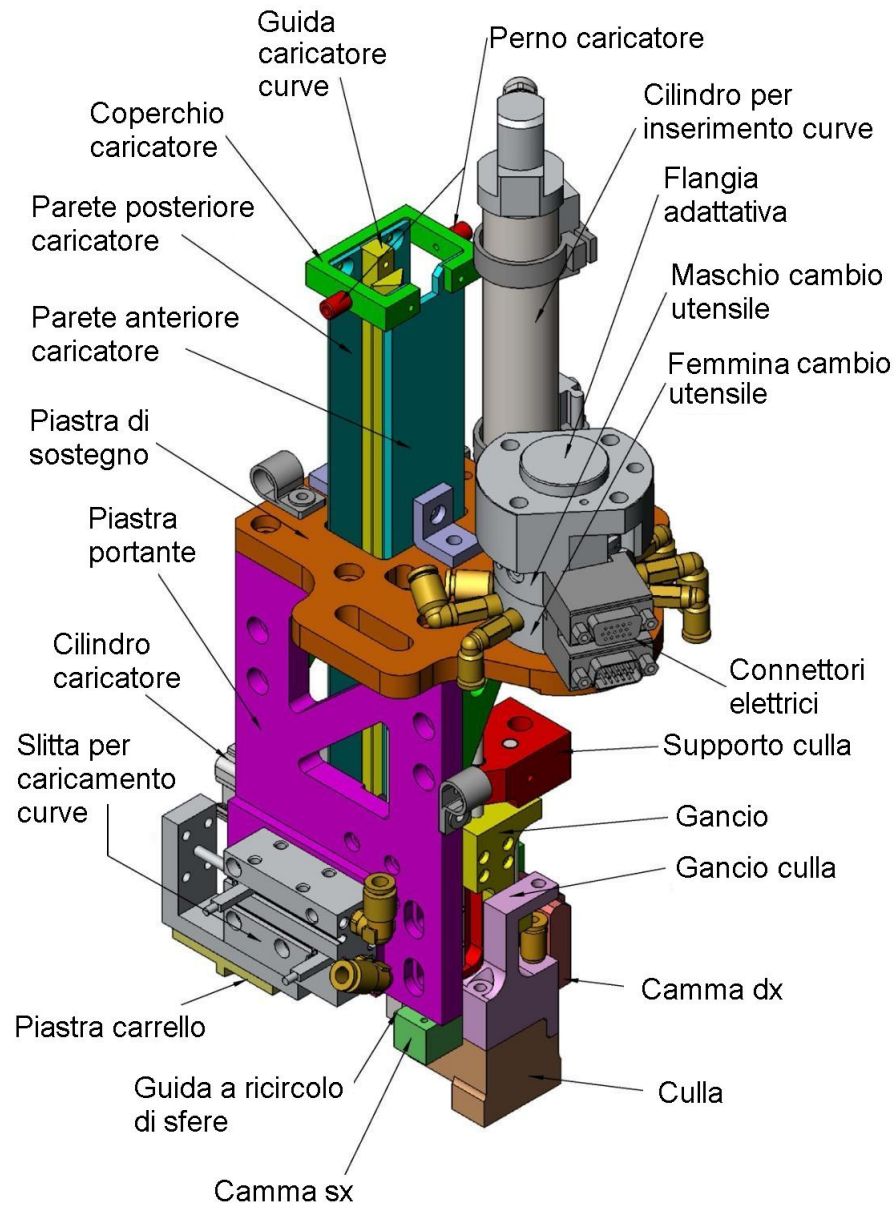


Figura 2.7: Modello CAD 3D dell'end-effector

Una rappresentazione dell'end-effector realizzato è riportata nell'immagine sopra, dove si nota il cilindro pneumatico a doppio effetto atto a portare la curvetta dal corpo del dispositivo alla sommità del pacco alettato della batteria, dove sono presenti i bicchieri delle forcine; meno visibile è la slitta laterale che permette di far scendere la curva ed inserirla nella culla, dove una ventosa a vuoto fa presa su di essa. Il pistone di inserimento verticale fa scendere la culla, la quale, a mezzo di un accoppiamento di forma con le due camme, ruota di  $90^\circ$  e si porta in posizione di inserimento. Un sistema di ganci impedisce che la culla ruoti quando si sta effettuando l'inserimento. Si rimanda comunque a [8] per maggiori dettagli su tutta la parte meccanica del progetto.

## Moduli magazzino

Tali moduli realizzano il sistema di caricamento delle curve nelle diverse pinze e vengono realizzati grazie all'impiego di tazze vibranti (vibro-buratti). L'utilizzo della tazza vibrante permette di selezionare ed orientare le curve da inserire nei diversi caricatori. La sua applicazione permette sia una lavorazione in automatico sia una più elevata produzione. Il vibratore è costituito da una base su cui è fissato, tramite appositi supporti, un elettromagnete ed una parte superiore atta a contenere o far scorrere i pezzi. Le due componenti, basamento e parte superiore, sono collegate tra loro tramite "balestre" di diverso materiale a seconda della tipologia di vibratore ed esigenze specifiche. Le tramogge vibranti, una per ogni tipologia di curva, sono dotate di un'apposita guida che alimenta il relativo utensile fermo a magazzino.

## Tavola Rotante

La tavola rotante è quel dispositivo che permette il passaggio della scambiatore dalla zona operatore (zona carico/scarico scambiatori) alla zona assemblaggio e viceversa; in questa applicazione si è deciso di utilizzare una tavola rotante ad azionamento pneumatico del tipo di quella mostrata in figura 2.8 il cui principio di funzionamento è il seguente: un cilindro a doppio effetto attiva un disco divisore che si trova sull'asse di rotazione della tavola, successivamente un posizionatore a cuneo con stantuffo fissa in posizione di arresto il disco divisore ed il piano della tavola ad esso collegato (Vedi [8] per ulteriori dettagli).

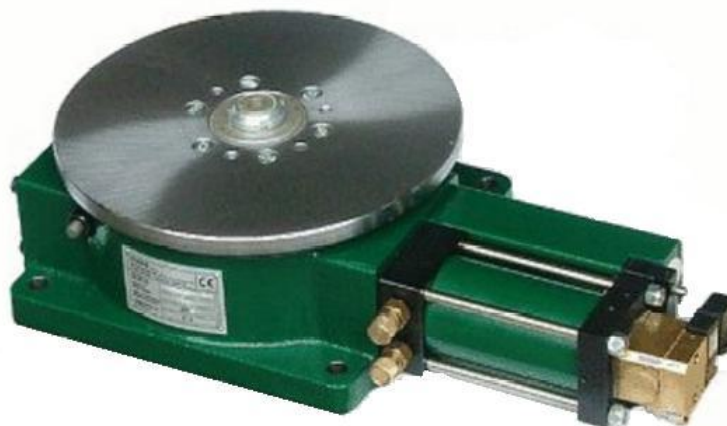


Figura 2.8: Tavola rotante

### Camera a matrice *AVT Pike 505B*

La Pike 505B, un cui esemplare dotato di ottica aggiuntiva è presentato in Figura 2.9, è una camera prodotta da Allied Vision Technology, nel settore delle camere a matrice ad alta risoluzione per applicazioni analitiche.



Figura 2.9: Camera AVT Pike 505B

La camera figura al suo interno un grande sensore CCD di tipo 2/3" (due terzi di pollice), caratterizzato da una matrice di unità fotosensibili in numero tale da consentire una risoluzione massima dell'immagine di 2452 x 2054 pixels. Il concetto di risoluzione massima è legato alla possibilità offerta dal sistema di acquisizione della camera di operare a risoluzioni inferiori unendo l'informazione catturata da più fotositi in un unico pixel. Questa tecnica, chiamata *binning*, aumentando di fatto l'area fotosensibile a cui fa riferimento il pixel, aumenta anche la sensibilità della camera. In alternativa risoluzioni inferiori possono essere ottenute tramite la definizione di una ROI di acquisizione sul sensore. Chiaramente vista la grande quantità di informazione contenuta in un'immagine digitale alla risoluzione massima indicata, con profondità a 8 o 14 bit, il relativo frame-rate, ovvero la frequenza alla quale la camera può catturare e trasmettere al sistema di elaborazione fotogrammi digitalizzati, non supera i 13 FPS (frames-per-second). Ovviamente, lavorando a risoluzioni inferiori (tramite binning o impostando una ROI sul sensore), il peso dei dati in trasmissione si riduce e possono essere raggiunti frame-rate maggiori. La camera integra al suo interno i circuiti e le funzioni del frame grabber (con codica a 8 o 14 bit), completo di memoria per la pre-elaborazione e trasmissione delle immagini acquisite. La Pike si interfaccia al sistema di elaborazione attraverso una connessione Firewire 800, utilizzando quindi il protocollo di trasmissione dati IEEE 1394 b<sup>2</sup>.

---

<sup>2</sup>Poiché l'azienda committente è provvista di un pc esterno con una connessione Firewire 400 (utilizzando quindi il protocollo di trasmissione dati IEEE 1394a) si sta testando tutt'ora la prestazione del software.

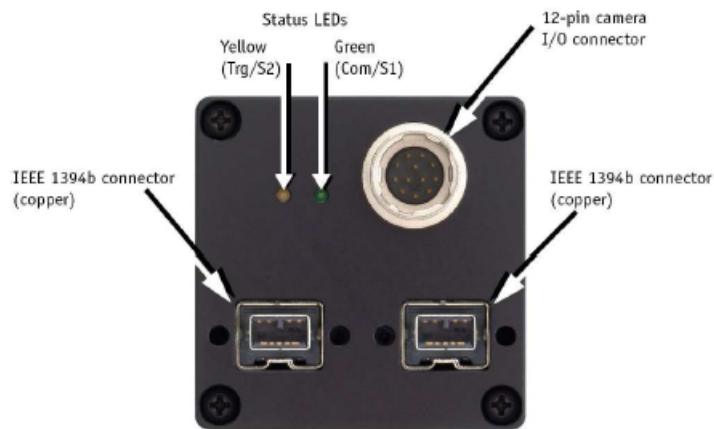


Figura 2.10: Retro della Camera AVT Pike 505B

Questo standard di comunicazione, introdotto sul mercato nel 2003, consente una velocità di trasmissione dei dati prossima agli 800 Mbit/s (precisamente 786.432 Mbit/s), e consentendo lo scambio veloce di grandi quantità di dati tra camera ed elaboratore ha promosso di fatto lo sviluppo e l'utilizzo nelle camere matriciali di sensori a più elevata risoluzione. L'interfaccia FireWire consente una comunicazione bidirezionale tra l'elaboratore e la camera: il primo può impostare i parametri di acquisizione dell'immagine (velocità otturatore, settaggio del guadagno, definizione di una ROI, ecc) inviando opportuni comandi alla camera, mentre la seconda, una volta acquisite le immagini, provvede ad inviarle all'elaboratore. La Pike prevede inoltre la possibilità di acquisire immagini in corrispondenza di segnali di trigger esterni (non provenienti dall'elaboratore via FireWire) attraverso il connettore HIROSE a 12-pin, che può gestire in alternativa al collegamento FireWire, l'alimentazione della camera e la gestione dei segnali di I/O.

Le prove di acquisizione ed elaborazione delle immagini effettuate con questa camera sono avvenute all'interno del software MATLAB, previa installazione di uno specifico driver DCAM, che consente al software di interfacciarsi e comunicare con la camera. Gli aspetti legati all'installazione di periferiche FireWire800 sotto il sistema operativo Windows XP e alla configurazione dei relativi driver sono descritti in appendice.

Le caratteristiche fondamentali della Camera *AVT Pike 505B* sono riportate nella tabella 2.1.

Caratteristica	Dettaglio
Tipo di sensore	Monocromatico a matrice Sony ICX 625 2/3", diagonale 11,016mm
Dimensioni pixel (fotosito)	3,45 $\mu$ m $\times$ 3,45 $\mu$ m
Dimensioni immagine	2452 $\times$ 2054
Frame rate (alla massima risoluzione)	13 fps
Velocità otturatore	27 $\mu$ s ... $\sim$ 67 s
Profondità conversione A/D	8 - 14 bit
Interfaccia digitale	IEEE 1394b 100-200-400-800 Mbit/s
Max camera Data Rate	62,5 MByte/s
Attacco gruppo ottico	C-mount

Tabella 2.1: Dati tecnici Camera AVT Pike 505B

### 2.2.1 Schema Elettrico

E' fondamentale a questo punto, dopo aver descritto il layout della cella, provvedere alla sua alimentazione e alla sua messa in sicurezza grazie alla gestione delle emergenze. Con alimentazione intendiamo tutti gli input ed output necessari al suo funzionamento, si pensi ad esempio all'end-effector ed agli attuatori e ai sensori alloggiati sullo stesso; mentre con messa in sicurezza intendiamo la realizzazione di un circuito per la gestione delle emergenze, in modo da fermare il funzionamento di tutta la cella in situazioni di pericolo.

Tralasciando per un attimo la prima parte (in seguito vedremo il circuito realizzato per le connessioni degli I/O) andiamo ad illustrare il funzionamento e le motivazioni che hanno portato a realizzare il circuito di emergenza in uso sulla cella.

Innanzitutto dobbiamo impedire che qualsiasi persona possa entrare nella parte riservata all'assemblaggio con il robot e/o la tavola rotante in movimento, ecco perchè sono previste delle recinzioni che ricoprono completamente il perimetro della cella (se rimosse aprono il circuito principale di emergenza) e due livelli di barriere dotate di sensori di presenza. Tali barriere funzionano nel seguente modo: una rilevazione di presenza della barriera esterna implica un arresto dell'alimentazione della tavola rotante mentre il resto della cella sarà attivo (questa situazione sarà la più frequente poichè è la normale fase di carico/scarico scambiatori), una rilevazione di presenza da parte di entrambe le barriere porta alla situazione di emergenza e quindi al blocco di tutta la cella compreso il Robot; la rilevazione di presenza da parte della sola barriera interna non provoca di per sè nessun intervento, questo sarebbe pericoloso nel caso in cui l'operatore fosse posto in qualche modo tra le due barriere e la tavola fosse in una posizione che permettesse il passaggio nella zona di assemblaggio, ma questa eventualità è impossibile poichè essere tra le due barriere comporta che la tavola sia in blocco (l'operatore è passato davanti alla barriera esterna per oltrepassarla) e che sia in posizione di carico/scarico o in posizione di lavorazione poichè solo in queste due situazioni la parte mobile della barriera interna consente di muoversi nello spazio compreso tra le due barriere. Chiaramente in questa situazione la messa in marcia della tavola rotante da parte dell'operatore deve essere vietata, ecco perchè il pulsante di marcia della tavola (compresi quello del robot e della marcia generale) sono posti all'esterno della cella, in modo tale che l'operatore debba passare la mano davanti alla barriera esterna per raggiungerli.

Prima di riportare lo schema elettrico è bene osservare che la messa in marcia della tavola non implica che la tavola si muova, infatti (vedi Figura 2.11) l'alimentazione del relativo circuito di emergenza è connessa ad un output del robot, quindi è il controllore che decide quando girare la tavola, chiaramente se questa è in marcia.

Si possono notare in Figura 2.11 le connessioni indicate con XUSR, che hanno lo scopo di rilevare anomalie da parte del controllore del robot, infatti esistono 4 piedini all'interno del connettore XUSR presente sul controllore che sono posti in corto circuito (due a due) se non sono presenti anomalie, andandando quindi a permettere il passaggio di corrente all'interno del circuito di emergenza.

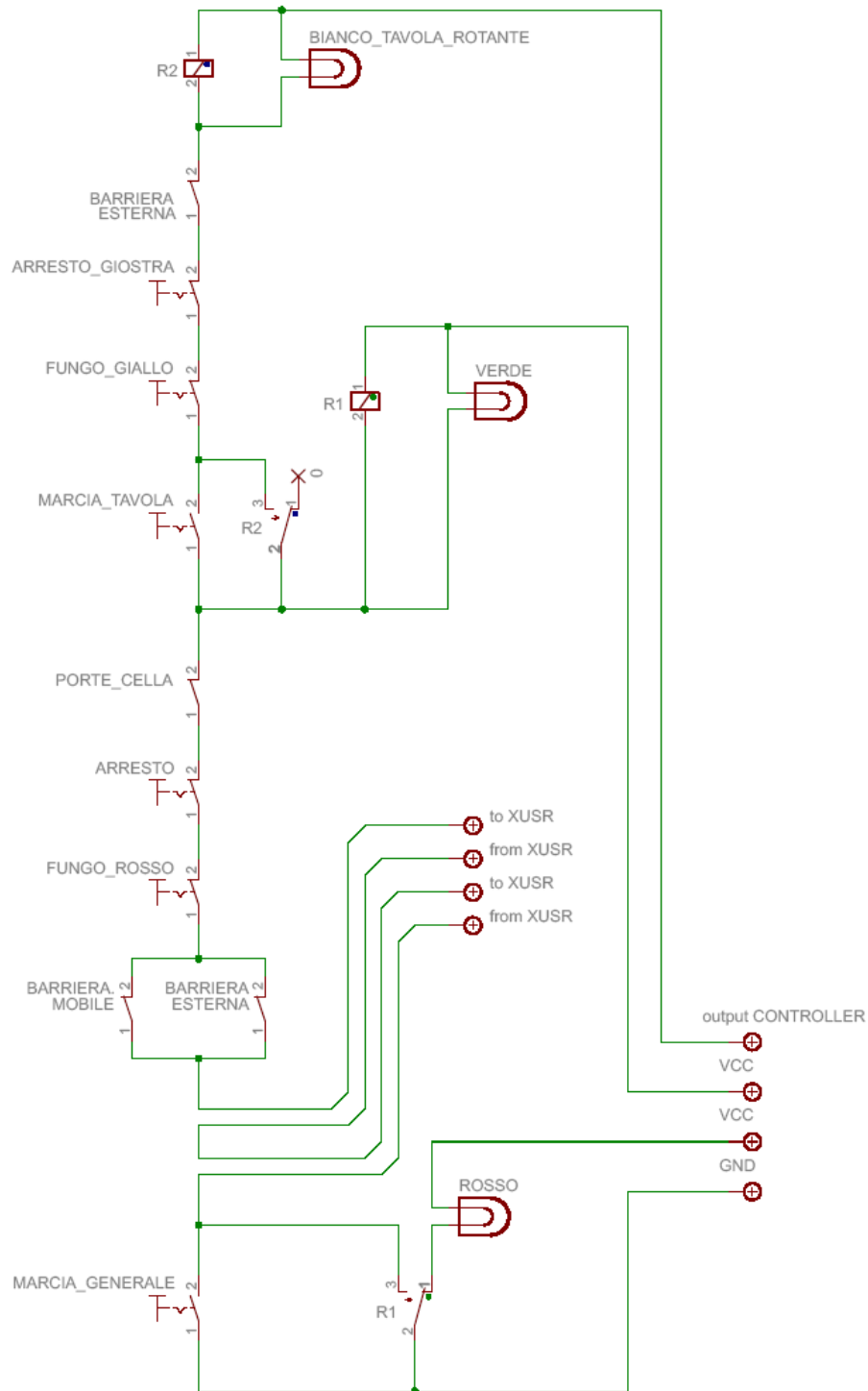


Figura 2.11: Circuito di Emergenza della Cella

L'inserimento della marcia generale fa passare corrente all'interno del relè R1 il quale, oltre ad autoalimentarsi (vedi figura) chiude il circuito di alimentazione dei vibro-buratti. L'inserimento della marcia della tavola rotante (possibile solo se la marcia generale è innestata) fa passare corrente all'interno del relè R2 il quale, oltre ad autolimentarsi (vedi figura) chiude il circuito di alimentazione della tavola rotante.

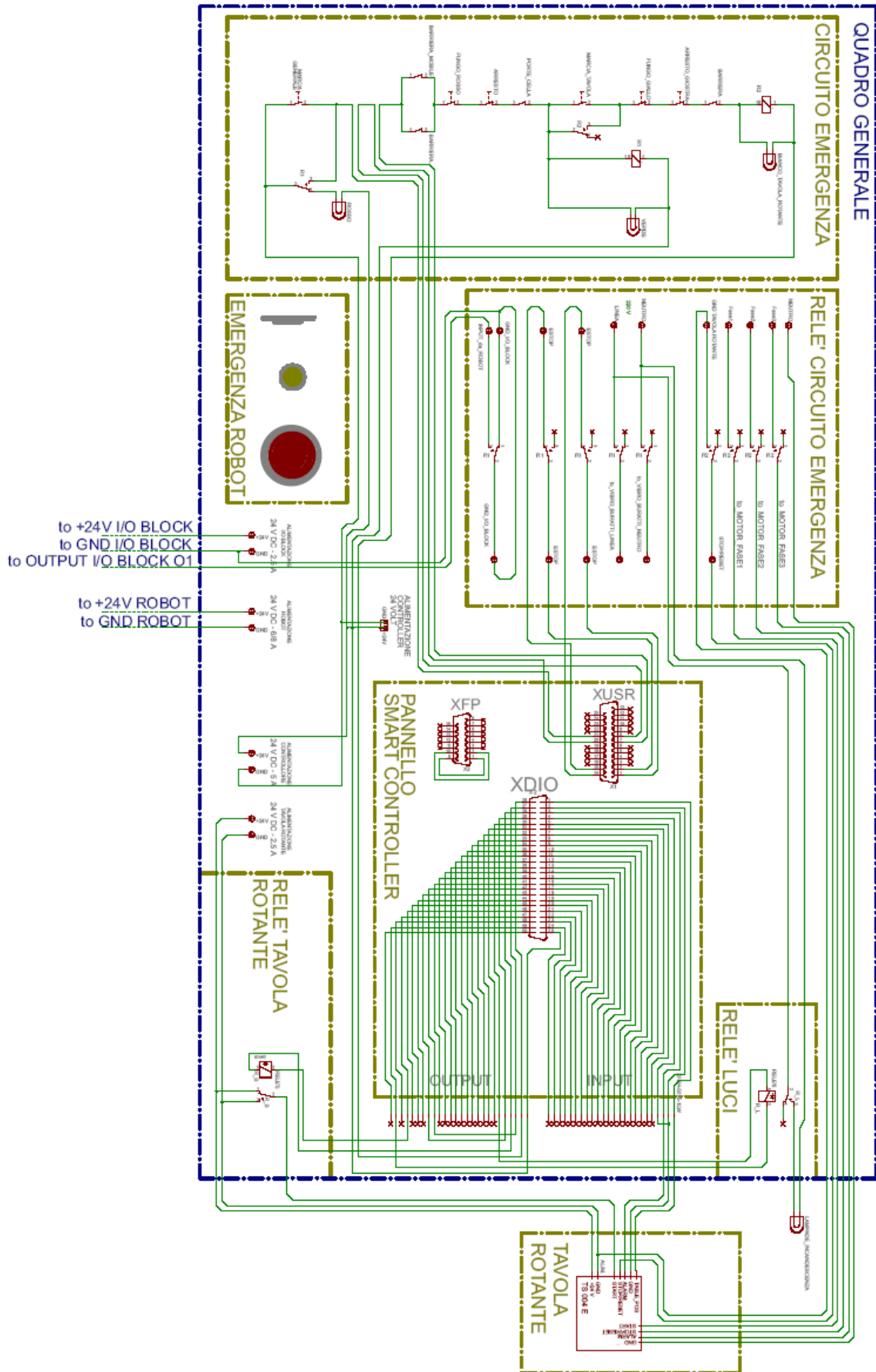


Figura 2.12: Schema I/O del quadro generale



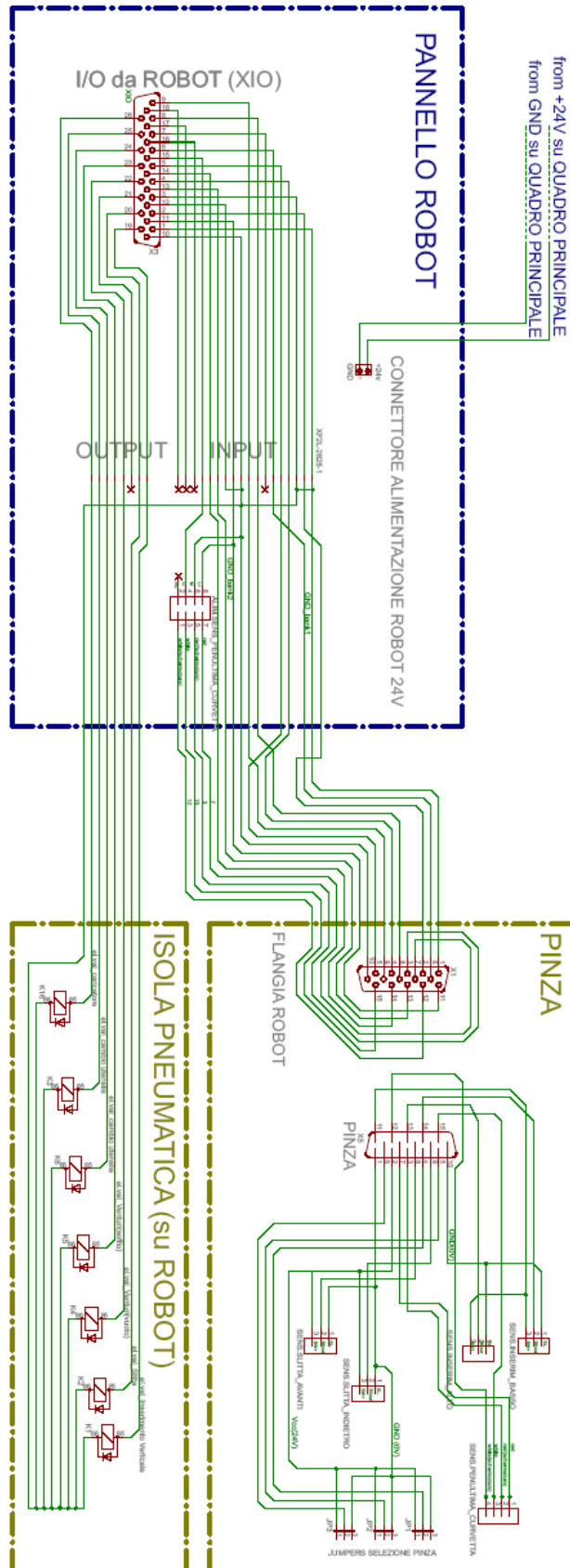


Figura 2.13: Schema I/O sul robot

## 2.3 Calibrazione della telecamera

Uno dei compiti affidati al sistema di visione artificiale è quello di identificare la posizione dei centri dei tubi sporgenti dalla piastra nel sistema di riferimento del robot. Il sistema di riferimento visione è però dato dalle coordinate riga-colonna della matrice dell'immagine acquisita dalla telecamera. Il calcolo della posizione dei centri-tubo elaborato dal software di visione avviene dunque all'interno di questo sistema di coordinate. Tali coordinate devono perciò essere successivamente rimappate nelle coordinate del sistema di riferimento del robot. La determinazione della funzione di mappatura dal sistema di riferimento della visione a quello del robot è lo scopo dell'operazione di calibrazione della telecamera.

Ricordiamo che, mentre il sistema di riferimento robot è definito nelle tre dimensioni spaziali ( $\mathbf{x}_r$ ,  $\mathbf{y}_r$  e  $\mathbf{z}_r$ ), il sistema di riferimento della telecamera è per sua natura limitato alle sole due dimensioni ( $\mathbf{x}_v$ ,  $\mathbf{y}_v$ ) che faremo coincidere con il piano medio su cui giacciono le imboccature dei tubi.

L'impiego di uno SCARA a 4 assi di movimento permette di portare la pinza sopra al punto di inserimento della curva previsto mantenendo costante l'orientazione verticale della pinza rispetto al piano di inserimento. In questo modo il movimento di inserimento di una curva consiste solamente in un movimento verticale dall'alto verso il basso dell'end-effector della pinza senza doversi preoccupare del suo orientamento.

L'ipotesi di perpendicolarità tra il piano di inserimento delle curve e l'asse verticale del robot fa sì che la funzione di mappatura dalle coordinate nel sistema di riferimento di visione a quelle del robot possa essere espressa nel seguente modo:

$$\begin{cases} x_r = f_x(x_v, y_v) \\ y_r = f_y(x_v, y_v) \\ z_r = cost \end{cases} \quad (2.1)$$

Idealmente la calibrazione della telecamera può essere vista come una particolare operazione di roto-traslazione tra i due differenti sistemi di riferimento:

$$\begin{aligned} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} &= \begin{bmatrix} \cos \alpha & -\sin \alpha & a_0 \\ \sin \alpha & \cos \alpha & b_0 \\ 0 & 0 & cost \end{bmatrix} \cdot \begin{bmatrix} x_v \\ y_v \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_v \\ y_v \end{bmatrix} + \begin{bmatrix} a_0 \\ b_0 \\ cost \end{bmatrix} \end{aligned}$$

da cui possiamo ricavare per  $f_x$  e  $f_y$  le due forme polinomiali lineari:

$$\begin{cases} x_r = a_0 + a_1 \cdot x_v + a_2 \cdot y_v \\ y_r = b_0 + b_1 \cdot x_v + b_2 \cdot y_v \end{cases} \quad (2.2)$$

Nel capitolo 2 abbiamo però visto che la lente dell'ottica montata sulla telecamera introduce nell'immagine acquisita una certa distorsione, di natura chiara-

mente non lineare, che aumenta man mano che ci si allontana dall'asse ottico e che può risultare piuttosto fastidiosa. Tale distorsione può essere compensata con buona precisione durante la fase di calibrazione della telecamera solamente attraverso un modello più complesso di tipo non lineare.

Una soluzione consiste nell'elevare il grado dei due polinomi che descrivono  $f_x$  ed  $f_y$ . Nel nostro caso abbiamo ritenuto sufficiente utilizzare come modello i polinomi quadratici:

$$\begin{cases} x_r &= a_0 + a_1 \cdot x_v + a_2 \cdot y_v + a_3 \cdot x_v^2 + a_4 \cdot y_v^2 \\ y_r &= b_0 + b_1 \cdot x_v + b_2 \cdot y_v + b_3 \cdot x_v^2 + b_4 \cdot y_v^2 \end{cases} \quad (2.3)$$

Scelto questo modello per le funzioni  $f_x(x_v, y_v)$  e  $f_y(x_v, y_v)$ , la calibrazione della telecamera può essere ridotta ad un'operazione di identificazione parametrica dei coefficienti  $\theta_x = [a_0, a_1, a_2, a_3, a_4]'$  e  $\theta_y = [b_0, b_1, b_2, b_3, b_4]'$  sui due modelli statistici lineari di misura :

$$\begin{cases} \mathbf{x}_r &= S \cdot \theta_x + \mathbf{w}_x \\ \mathbf{y}_r &= S \cdot \theta_y + \mathbf{w}_y \end{cases} \quad (2.4)$$

Secondo questa rappresentazione i vettori  $\mathbf{x}_r = [x_r(1), \dots, x_r(n)]'$  e  $\mathbf{y}_r = [y_r(1), \dots, y_r(n)]'$  possono essere visti come il risultato di  $n$  misure eseguite posizionando l'end effector del robot in punti diversi sul piano di inserimento delle curve e rilevandone le corrispondenti coordinate nel sistema di riferimento della visione, qui opportunamente registrate all'interno della matrice  $S$ :

$$S = \begin{bmatrix} 1 & x_v(1) & y_v(1) & x_v(1)^2 & y_v(1)^2 \\ & & \dots & & \\ 1 & x_v(n) & y_v(n) & x_v(n)^2 & y_v(n)^2 \end{bmatrix} \in \mathbb{R}^{n \times p}$$

mentre il vettore  $\mathbf{w} \in \mathbb{R}^n$  rappresenta il rumore di misura. Questo modello segnale+rumore è tipico nella descrizione di canali di comunicazione numerica o per misure fatte sequenzialmente nel tempo da sensori numerici nei sistemi di controllo.

A questo punto per completare la procedura di calibrazione della telecamera non resta che determinare lo stimatore di massima verosimiglianza dei parametri  $\theta_x$  e  $\theta_y \in \mathbb{R}^p$  (nel nostro caso  $p = 5$ ) nel modello lineare di misura 2.4 [11]. Per semplicità ipotizziamo che il rumore di misura  $\mathbf{w}$  sia Gaussiano a media nulla della forma  $\mathbf{w} = \sigma \mathbf{n}$ , dove  $\mathbf{n} \sim \mathcal{N}(0, I)$ ,  $I \in \mathbb{R}^{n \times n}$ , da cui deriva una matrice di covarianza del rumore:

$$\begin{aligned} R &= E[\mathbf{w} \cdot \mathbf{w}'] \\ &= \sigma^2 I \end{aligned} \quad (2.5)$$

Si può dimostrare per il teorema di Rothemberg che se nel modello 2.4 con  $n \geq p$  allora  $\theta$  è *globalmente identificabile* (cioè il vettore  $\theta$  è identificabile in modo univoco) se e solo se la matrice  $S$  ha rango pieno  $p$ , cioè se ho almeno

$p$  righe di  $S$  linearmente indipendenti. Nel nostro caso, con  $p = 5$ , risulta fondamentale eseguire durante la procedura di calibrazione della telecamera almeno cinque misurazioni in grado di fornire dei vettori  $[1, x_v(i), y_v(i), x_v(i)^2, y_v(i)^2]$  linearmente indipendenti tra di loro. Lo stimatore di massima verosimiglianza dei parametri  $\theta_x$  e  $\theta_y$ , nell'ipotesi quindi di invertibilità della matrice  $S'R^{-1}S$  risulta perciò uguale a:

$$\begin{aligned}\hat{\theta}(\mathbf{y}_r) &= [S'R^{-1}S]^{-1} S'R^{-1} \cdot \mathbf{y}_r \\ &= \left[ S' \left( \frac{1}{\sigma^2} I \right) S \right]^{-1} S' \left( \frac{1}{\sigma^2} I \right) \cdot \mathbf{y}_r \\ &= [S'S]^{-1} S \cdot \mathbf{y}_r\end{aligned}\tag{2.6}$$

Questa operazione di identificazione dei parametri  $\theta_x$  e  $\theta_y$  è stata implementata nella funzione *fittingPolinomiale2D*.

Per quanto riguarda la varianza scalare della stima, data dall'equazione:

$$\begin{aligned}\text{var}(\hat{\theta}(\mathbf{y}_r)) &= \text{Tr}(\sigma^2 [S'S]^{-1}) \\ &= \text{Tr} \left( \sigma^2 \cdot \begin{bmatrix} n & \sum_1^n x_v(i) & \sum_1^n y_v(i) & \sum_1^n x_v(i)^2 & \sum_1^n y_v(i)^2 \\ \sum_1^n x_v(i) & \sum_{i1}^n x_v(i)^2 & \dots & \dots & \dots \\ \sum_1^n y_v(i) & \dots & \sum_1^n y_v(i)^2 & \dots & \dots \\ \sum_1^n x_v(i)^2 & \dots & \dots & \sum_1^n x_v(i)^4 & \dots \\ \sum_1^n y_v(i)^2 & \dots & \dots & \dots & \sum_1^n y_v(i)^4 \end{bmatrix}^{-1} \right)\end{aligned}$$

(coincidente con il limite di Cramèr-Rao e perciò efficiente) si può constatare, come ci si poteva immaginare essendo  $\sigma^2 [S'S]^{-1}$  definita positiva, che all'aumentare del numero  $n$  di misurazioni effettuate diminuisce l'entità della varianza dello stimatore con conseguente aumento della bontà della calibrazione della telecamera. Per maggiori dettagli rimandiamo a [11].

Dal punto di vista pratico, la procedura di calibrazione della telecamera viene eseguita nel seguente modo. Sulla flangia del robot viene montato un puntale di calibrazione (Figura 2.14) di dimensioni tali da ricoprire la stessa distanza tra la flangia del robot ed il piano di inserimento delle curve durante l'operazione di assemblaggio dello scambiatore al fine di riprodurre il più fedelmente possibile le reali condizioni di lavoro. Questo puntale di calibrazione è provvisto di una serie di dischi metallici di dimensioni note che possono essere posizionati con grande precisione dal robot sulla superficie del piano di inserimento. Il disco infatti, realizzato al tornio assieme al puntale con un elevato grado di precisione, è provvisto di un foro conico al suo centro che consente il suo perfetto centraggio in corrispondenza dell'asse del puntale.

Una volta posizionato un disco in una certa posizione  $i$  sul piano di inserimento delle curve, il robot provvede ad inviare all'elaboratore le coordinate  $(x_r(i), y_r(i), z_r(i))$  della flangia espresse nel sistema di riferimento del robot. A questo punto il robot viene spostato da sopra al piano di inserimento, con l'accortezza di lasciare esattamente nello stesso punto il disco appena posizionato. Non appena il robot è fuori dalla visuale della telecamera il sistema di visione provvede ad acquisire un'immagine del disco opportunamente illuminato e a determinarne le coordinate  $(x_v(i), y_v(i))$  di posizione all'interno del sistema di riferimento dell'immagine.

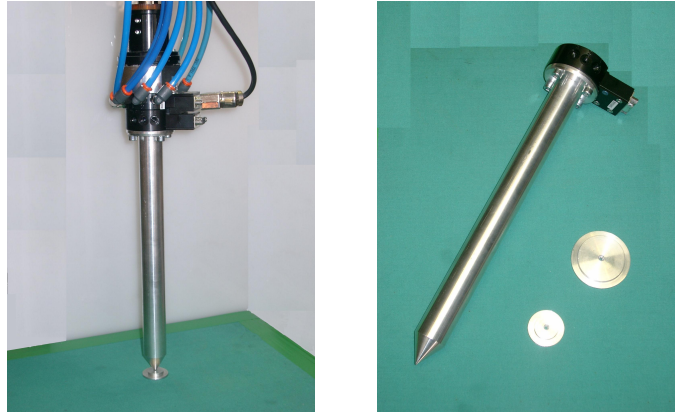


Figura 2.14: Calibrazione della telecamera in laboratorio e puntale di calibrazione con relativi dischi.

L'identificazione della posizione del disco viene effettuata con la medesima tecnica del centroide precedentemente descritta, che garantisce una precisione sub-pixel per le coordinate finali del centro del disco, oltre la risoluzione della matrice di immagine.

Una volta eseguiti un sufficiente numero di posizionamenti e relative misurazioni nei due sistemi di riferimento, i dati raccolti possono essere finalmente elaborati dalla funzione *fittingPolinomiale2D* così da ottenere la funzione di mappatura tra i due sistemi di riferimento cercata. Inoltre, conoscendo con molta precisione le dimensioni del disco di calibrazione, confrontando le sue quote reali con le misurazioni del suo diametro eseguite dal sistema di visione sulle immagini acquisite durante questa procedura di calibrazione possiamo ricavare l'effettivo rapporto pixel/mm delle immagini.

## 2.4 Calibrazione delle pinze

La locazione di inserimento di una curva viene identificata con un frame ( $\mathbf{x}_c$ ,  $\mathbf{y}_c$ ,  $\mathbf{z}_c$ ) in corrispondenza dell'end-effector del robot (la culla di alloggiamento delle curve della pinza) avente l'origine posizionata nel punto medio tra i due centri-tubo della curva, l'asse  $\mathbf{x}_c$  orientato secondo la direzione del segmento che li congiunge e l'asse  $\mathbf{z}_c$  perpendicolare al piano di inserimento ed orientato in direzione del movimento di inserimento della curva (Figura 2.15).

Abbiamo però visto che la calibrazione della telecamera ci fornisce una corrispondenza tra le coordinate nel sistema di riferimento della visione a quelle del robot e non del frame di inserimento curva. Per poter eseguire correttamente l'operazione di inserimento di una curva in una determinata locazione dobbiamo perciò conoscere la posizione e l'orientazione dell'end-effector della pinza utilizzata rispetto al sistema di riferimento del robot.

Teoricamente, note le dimensioni costruttive di ogni singola pinza, sarebbe sufficiente determinare una matrice di trasformazione che va dal sistema di del robot al frame di inserimento curva passando per il frame ( $\mathbf{x}_T$ ,  $\mathbf{y}_T$ ,  $\mathbf{z}_T$ ) del tool (Figura 2.15) posizionato in corrispondenza della flangia del robot e a noi noto:

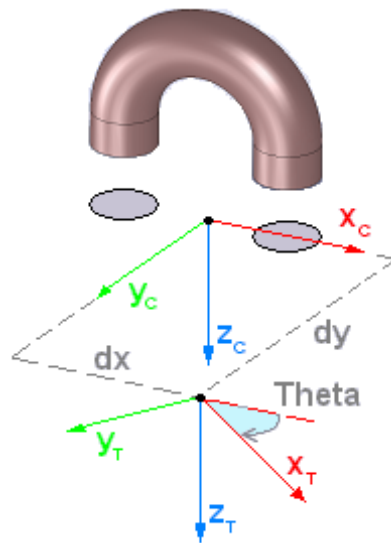


Figura 2.15: Frame robot sulla flangia rispetto al frame di inserimento curva

$$T_T^c = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & dx \\ \sin \theta & \cos \theta & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In realtà le quote dimensionali della pinza sono espresse entro certe tolleranze costruttive, le quali inoltre non tengono conto di eventuali giochi legati alla flangia adattativa del sistema di cambio dell'utensile attraverso la quale la pinza viene collegata al robot. Si preferisce così eseguire una successiva operazione di calibrazione delle pinze utilizzate al fine di determinare con maggior precisione questa matrice di trasformazione tra i due frame.

Per eseguire questa procedura di calibrazione è prevista la realizzazione di alcuni oggetti in metallo a forma di curva come in figura 2.16, una per ogni tipologia di curva al variare del diametro del tubo e dell'interasse, e da noi chiamate curve di calibrazione. Questi oggetti sono stati pensati in modo tale da poter essere posizionato dalle rispettive pinze al pari di una curva reale. La basetta inferiore ha il duplice scopo di consentire il posizionamento della curva di calibrazione sulla superficie del piano di calibrazione e di fornire al sistema di visione una sagoma facilmente identificabile. Inoltre un foro realizzato in prossimità di una delle estremità della basetta consente di identificarne in modo univoco l'orientazione.

La procedura di calibrazione delle pinze è analoga a quella di calibrazione della telecamera. Una volta posizionata una di queste curve sulla superficie del piano di calibrazione utilizzando la pinza da calibrare, viene rilevata la locazione della terna di riferimento del tool  $(x_T, y_T, z_T, \theta_T^Z)$  espressa all'interno del sistema di riferimento robot. Spostato il robot dalla visuale della telecamera, viene acquisita un'immagine della curva di calibrazione. Data la sua apposita forma, dall'elaborazione dell'immagine è possibile ricavarne la posizione del centroide  $(x_c, y_c)$  e la sua orientazione  $\theta_c$  nel sistema di riferimento della visione. Le coordi-

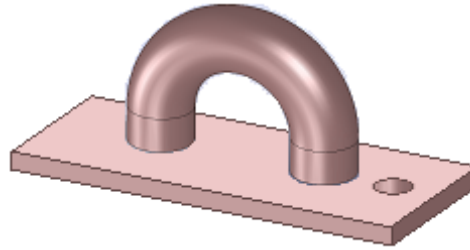


Figura 2.16: Curva di calibrazione

nate del centroide della curva, espresse nel sistema di riferimento dell'immagine, vengono riportate all'interno del sistema di riferimento del robot attraverso la mappatura determinata attraverso la precedente operazione di calibrazione della telecamera. Nota l'altezza del piano di calibrazione, supposto perpendicolare all'asse  $z$  del robot, dai dati così raccolti è immediato risalire ai parametri  $dx$ ,  $dy$ ,  $dz$  e  $\Theta$  per differenza.

## 2.5 Interfaccia di calibrazione della cella

Per agevolare all'operatore l'esecuzione della procedura di calibrazione della cella è stata realizzata una interfaccia di calibrazione utilizzando il *Graphical User Interface Development Environment* (GUIDE) di MATLAB. Questo editor di MATLAB fornisce tutta una serie di tool che consentono la semplice e rapida creazione di interfacce grafiche (GUI).

La procedura di calibrazione della cella è stata suddivisa all'interno di tre distinte GUI: un'interfaccia di calibrazione della telecamera, una di calibrazione delle pinze ed infine un'interfaccia di settaggio dei parametri di acquisizione ed elaborazione delle immagini. Queste tre distinte procedure di calibrazione possono essere richiamate dalla GUI principale di calibrazione della cella.

### 2.5.1 GUI principale di calibrazione della cella

La GUI principale di calibrazione della cella (Figura 2.17) consente di richiamare le tre diverse procedure di calibrazione della cella attraverso gli appositi pulsanti e di visualizzarne i risultati che andranno salvati nelle apposite strutture dati una volta completata la procedura di calibrazione premendo il pulsante Save.

Prima di procedere con la calibrazione della cella è necessario selezionare il modello di telecamera utilizzata nel menù a tendina.

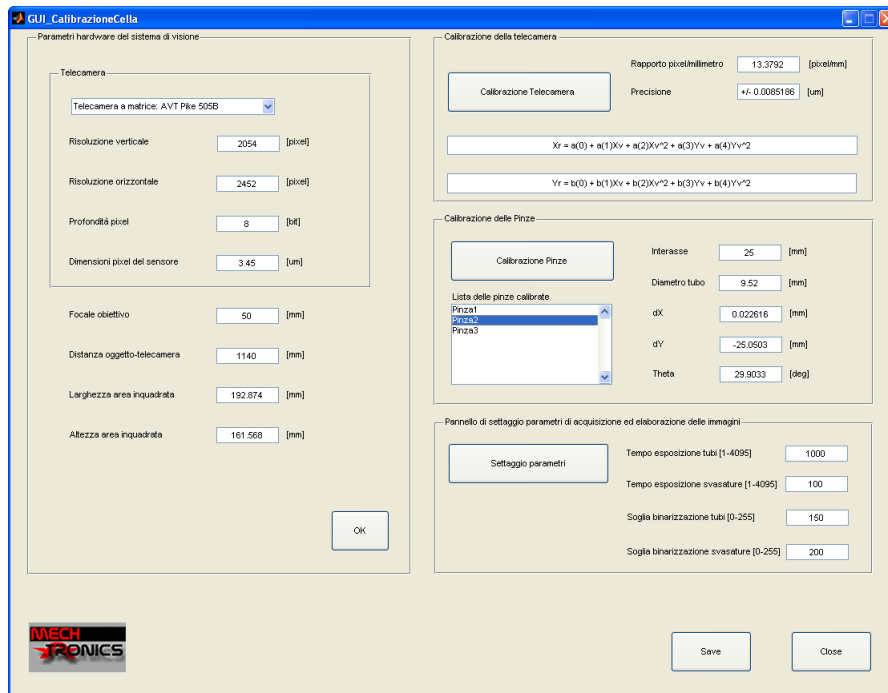


Figura 2.17: GUI principale di calibrazione della cella.

Attualmente è possibile selezionare solamente la telecamera a matrice AVT Pike 505B o la telecamera lineare Dalsa Spyder2 S2-22-04K40. È possibile inoltre specificare la distanza tra il piano di lavoro e telecamera e la lunghezza focale dell'obiettivo utilizzato per avere un'informazione preliminare delle dimensioni dell'area inquadrata e del relativo rapporto pixel/mm.

Selezionata la telecamera, la procedura di calibrazione deve essere eseguita nel seguente ordine: per prima va effettuata la calibrazione della telecamera e solo successivamente può essere eseguita la calibrazione delle pinze. Il settaggio dei parametri di acquisizione ed elaborazione può essere eseguito in qualsiasi momento.

Nel caso in cui si voglia modificare solo in parte alcuni parametri di settaggio del sistema di visione o aggiungere una nuova pinza alla lista delle pinze calibrate non è necessario rieseguire da capo la calibrazione dell'intera cella in quanto è possibile riutilizzare i dati salvati a seguito della precedente procedura di calibrazione.

Una volta terminate tutte le procedure di calibrazione della cella e di settaggio dei parametri, i risultati ottenuti vengono visualizzati all'interno dei relativi panel e se ritenuti corretti possono essere salvati nella struttura DatiCella, inserita nel file DatiCellaFile.mat all'interno della cartella DatiCalibrazioneCella, premendo il pulsante "Save".



### 2.5.2 GUI di calibrazione della telecamera

Questa interfaccia (Figura 2.18) è stata realizzata per eseguire in modo automatico la procedura di calibrazione della telecamera.

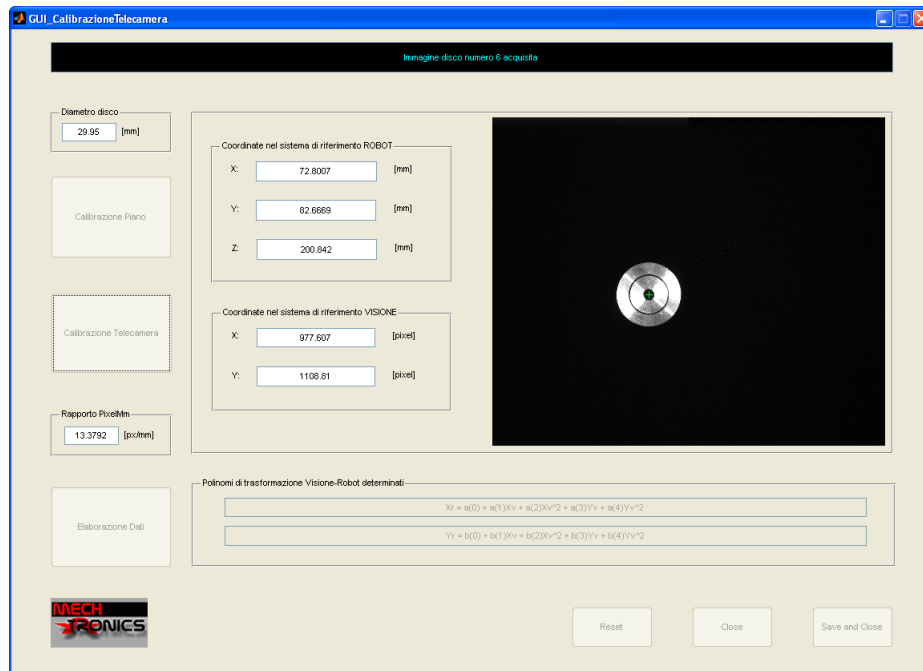


Figura 2.18: GUI di calibrazione della telecamera.

La procedura di calibrazione eseguita da questa routine è stata suddivisa in tre distinte fasi: calibrazione del piano, calibrazione della telecamera ed elaborazione dei dati. Prima di procedere con questa procedura di calibrazione è necessario fissare alla flangia del robot l'apposito puntale di calibrazione e posizionare in corrispondenza del previsto piano di inserimento delle curve un piano di calibrazione, di dimensioni pari almeno quanto l'area inquadrata dalla telecamera, sulla cui superficie poter andare a posizionare i dischi metallici utilizzati per la calibrazione della telecamera.

La prima fase di calibrazione del piano serve ad individuare l'esatta posizione del piano di calibrazione all'interno dello spazio di lavoro del robot. Essendo la successiva fase eseguita in modo automatico, per evitare di danneggiare il puntale o il piano di calibrazione durante il posizionamento dei dischi, si rende necessario per il robot conoscere l'esatta posizione del piano all'interno del suo sistema di riferimento. Per individuare la posizione del piano di calibrazione, l'operatore deve posizionare il puntale del robot tramite pendant in cinque distinti punti della superficie del piano ed inviarne la locazione all'elaboratore. Dalla rilevazione di questi cinque punti il programma è in grado di ricavare una matrice di trasformazione che va dal sistema di riferimento della flangia del robot ad un nuovo sistema di riferimento orientato in modo tale da avere l'origine giacente sul piano di calibrazione e l'asse z perpendicolare al piano stesso (Figura 2.19).

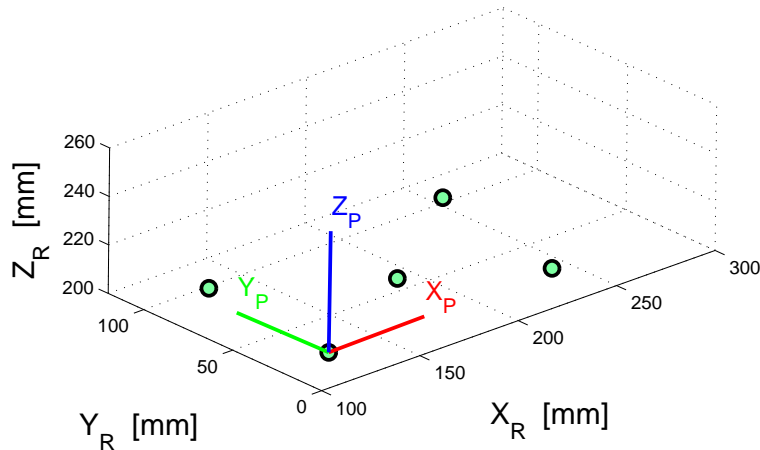


Figura 2.19: Individuazione del frame del piano di calibrazione.

La successiva fase di calibrazione della telecamera, eseguibile premendo il relativo pulsante “Calibrazione Telecamera”, prevede l’esecuzione in automatico dell’operazione di posizione e rilevamento del disco descritta nel paragrafo 4.4.1.

Inizialmente questa fase prevede che l’operatore posizioni manualmente il disco metallico di calibrazione in tre distinte posizioni sul piano di calibrazione all’interno dell’area inquadrata dalla telecamera e ne acquisisca le locazioni con la pendant mentre il sistema di visione provvede a rilevarne la posizione all’interno del sistema di riferimento dell’immagine. Questa prima parte manuale è necessaria a consentire al programma di determinare una prima mappatura approssimativa tra i pixel nell’immagine e le relative posizioni all’interno del sistema di riferimento del robot. Attraverso questa prima mappatura viene determinata una griglia di 12 punti posizionati sulla superficie del piano di calibrazione ed uniformemente disposti all’interno dell’area inquadrata dalla telecamera.

A questo punto può cominciare la procedura vera e propria di posizionamento ed acquisizione dei dischi in modo automatico. Il robot provvede in maniera automatica a trascinare il disco in una delle dodici posizioni individuate sul piano di calibrazione, successivamente lo centra e ne rileva la locazione nel sistema di riferimento del robot. Rilevata la locazione il robot si sposta in una locazione precedentemente stabilita al di fuori dell’area inquadrata dalla telecamera per permettere al sistema di visione di poter acquisire un’immagine del disco e così rilevarne la posizione nel sistema di riferimento dell’immagine. Terminata l’acquisizione il robot può riportarsi al di sopra del disco per trascinarlo nella locazione successiva.

Acquisite le coordinate nei due sistemi di riferimento del disco nelle 12 posizioni assunte, si può procedere alla terza fase di elaborazione dei dati. Premendo il pulsante “Elaborazione Dati” viene eseguita la funzione *fittingPolinomiale2D* sui dati appena raccolti così da ottenere i coefficienti dei polinomi quadratici della funzione di mappatura tra i due sistemi di riferimento cercata.

Queste tre fasi vanno eseguite in successione, ma se sono presenti i dati relativi ad una precedente calibrazione non è necessario rieffettuare l'intera procedura di calibrazione dall'inizio ma è possibile riprenderla da una fase qualsiasi. Nel caso in cui si intenda riavviare da capo la procedura di calibrazione della telecamera è sufficiente premere il tasto "Reset". Mentre le prime due fasi richiedono di instaurare una connessione con il controllore del robot, la terza fase di elaborazione dei dati può essere eseguita off-line sui dati ricavati durante la precedente operazione di calibrazione eseguita.

### 2.5.3 GUI di calibrazione delle pinze

Per poter eseguire la calibrazione delle pinze utilizzate nella cella è stata realizzata la GUI di figura 2.20.

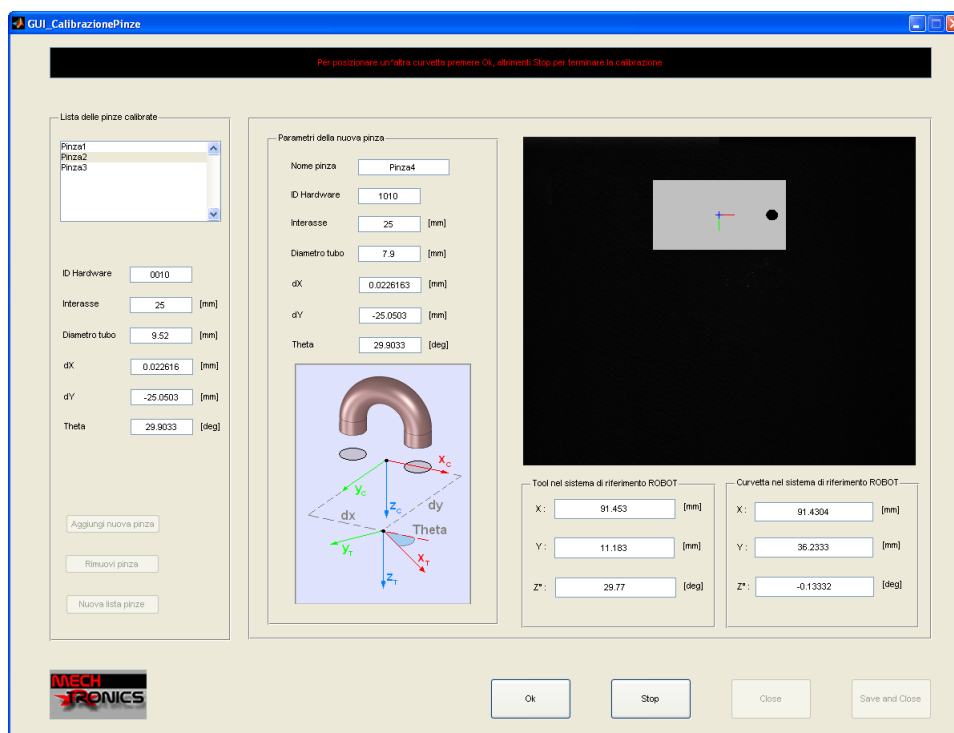


Figura 2.20: GUI di calibrazione delle pinze.

Innanzitutto questa interfaccia permette di vedere all'interno del menù a tendina l'attuale lista delle pinze finora calibrate e contenuta all'interno del file CalibrazionePinze.mat contenuto all'interno della cartella DatiCalibrazioneCella. Da questa GUI è possibile inoltre aggiungere alla lista una nuova pinza avviando così la procedura di calibrazione della stessa, eliminare dalla lista una pinza precedentemente inserita oppure creare da zero una nuova lista di pinze.

Ogni pinza calibrata è caratterizzata dai seguenti parametri:

- Nome: è una stringa di caratteri contenente il nome della pinza;

- IDHardware: costituito da 4 bit, è il codice hardware identificativo della pinza presente sull'attacco della flangia adattativa che consente al robot di riconoscerla in modo univoco;
- Interasse: è l'interasse della curva per la quale è stata creata questa pinza;
- DiametroTubo: è il diametro del tubo della curva per la quale è stata creata questa pinza;
- dx: offset lungo l'asse  $\mathbf{x}_c$  dell'origine del frame tool sulla flangia rispetto al frame di inserimento curva;
- dy: offset lungo l'asse  $\mathbf{y}_c$  dell'origine del frame tool sulla flangia rispetto al frame di inserimento curva;
- Theta: è l'angolo di rotazione sull'asse  $\mathbf{z}_c$  del frame tool sulla flangia rispetto al frame di inserimento curva.

Mentre i primi quattro parametri devono essere inseriti dall'operatore, i valori di dx, dy e Theta vengono determinati attraverso la procedura di calibrazione della pinza. A prima vista mancherebbe da determinare il parametro dz di offset lungo l'asse  $\mathbf{z}_c$  dell'origine del frame tool sulla flangia rispetto al frame di inserimento curva. In realtà non è necessario conoscere l'esatto valore di questo parametro attraverso la procedura di calibrazione in quanto il movimento verticale di inserimento eseguito dal pistone pneumatico di movimentazione della culla consente ampi margini di gioco.

La procedura di calibrazione di una nuova pinza viene avviata attraverso il pulsante "Aggiungi nuova pinza". Per prima cosa viene chiesto all'operatore di inserire nome, ID hardware, interasse e diametro tubo della nuova pinza che si intende calibrare. Per poter essere accettati questi parametri devono avere presentare nome e ID hardware diverso da tutte le altre pinze presenti nella lista.

Non appena inseriti i parametri viene chiesto all'operatore di montare sulla flangia del robot la nuova pinza da calibrare e di inserire nella culla di alloggiamento delle curve la curva di calibrazione, avendo successivamente in seguito l'accortezza di inserire la curva nella culla sempre nella stessa posizione con il foro sulla base sempre rivolto al medesimo lato.

A questo punto viene aperta la comunicazione con il robot, se non era stata già attivata precedentemente, e chiesto all'operatore di posizionare manualmente con la pendant la curva di calibrazione sulla superficie del piano di calibrazione. Rilevata la locazione nel sistema di riferimento del robot, questi viene spostato dalla visuale della telecamera ed acquisita un'immagine con il sistema di visione. Dall'elaborazione dell'immagine con la funzione *regionprops* viene calcolata la posizione e l'orientamento della curva nel sistema di riferimento della visione. Questi valori rilevati dal sistema di visione vengono successivamente riportati nel sistema di riferimento del robot attraverso la mappatura ricavata durante la precedente fase di calibrazione della telecamera. Dal confronto tra la locazione del frame sulla flangia e quello della curva di calibrazione, entrambi espressi nel sistema di riferimento del robot, viene determinata una stima dei parametri dx, dy e Theta.

Per aumentare la precisione nella determinazione di questi parametri è possibile ripetere più volte questa operazione di misura andando a posizionare più volte la curva di calibrazione in posizioni diverse: al ripetersi dell'operazione di posizionamento e ad acquisizione della curva le stime dei parametri  $dx$ ,  $dy$  e  $\Theta$  vengono aggiornate come media dei dati finora raccolti. Una volta ripetuta la di calibrazione della pinza un numero di volte ritenuto sufficiente, premendo il tasto "Stop" la procedura di calibrazione viene terminata e la nuova pinza con i relativi parametri appena determinati viene aggiunta alla lista delle pinze.

Terminato di calibrare tutte le pinze, premendo il tasto "Save" è possibile ritornare all'interfaccia principale di calibrazione della cella importandovi la lista di pinze finora manipolata ed ora pronta per essere inserita all'interno della struttura DatiVision.

#### 2.5.4 GUI di settaggio dei parametri di acquisizione ed elaborazione

Come è già stato introdotto (ne parleremo approfonditamente nella Capitolo 4) il sistema di visione ha due compiti fondamentali; l'identificazione dello scambiatore e l'analisi dei difetti. Queste due operazioni, come si vedrà in seguito, necessitano l'acquisizione di due immagini (una con un'illuminazione diffusa e l'altra con un'illuminazione laterale) e la binarizzazione delle stesse, si consideri che le due immagini possono avere soglie di binarizzazione (così come livelli di shutter) diverse.

E' indispensabile poter modificare sia in fase di calibrazione che nella fase di lavorazione vera e propria questi livelli (shutter e binarizzazione), infatti una diversa presenza di illuminazione della cella, si pensi ad esempio ad una giornata di sole o ad una giornata nuvolosa, richiederà un set di parametri diversi.

Per facilitare e velocizzare questa operazione è stata realizzata un'interfaccia di settaggio dei parametri (Figura 2.21), richiamabile dalla GUI principale di calibrazione della cella premendo l'apposito pulsante "Settaggio parametri" oppure dall GUI principale di lavoro premendo il pulsante "Avanzate" (vedi Cap. 3).

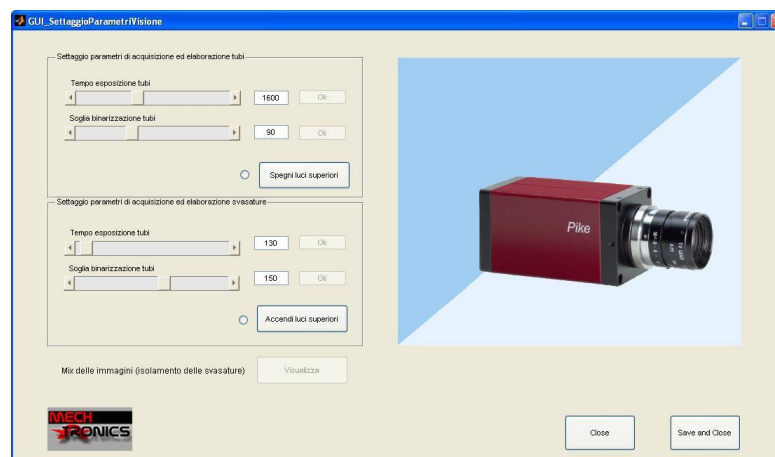


Figura 2.21: GUI di settaggio dei parametri di acquisizione ed elaborazione.

Dopo essere stata richiamata, questa routine di settaggio dei parametri consente di accendere alternativamente i due sistemi di illuminazione e di eseguire e visualizzare delle prove di acquisizione e relativa binarizzazione delle immagini al variare del tempo di esposizione e del valore di soglia.

Proviamo a dare un esempio della funzionalità della GUI: cominciamo con l'illuminazione laterale, a tal proposito dovremo premere il pulsante "Spegni Luci Superiori" presente nel riquadro chiamato "Set parametri di acquisizione ed elaborazione tubi". Variando il "Tempo di esposizione tubi" non facciamo altro che modificare lo shutter della telecamera, premendo il pulsante "OK" abilitato si acquisisce l'immagine con il tempo di esposizione desiderato (un esempio a Figura 2.22).

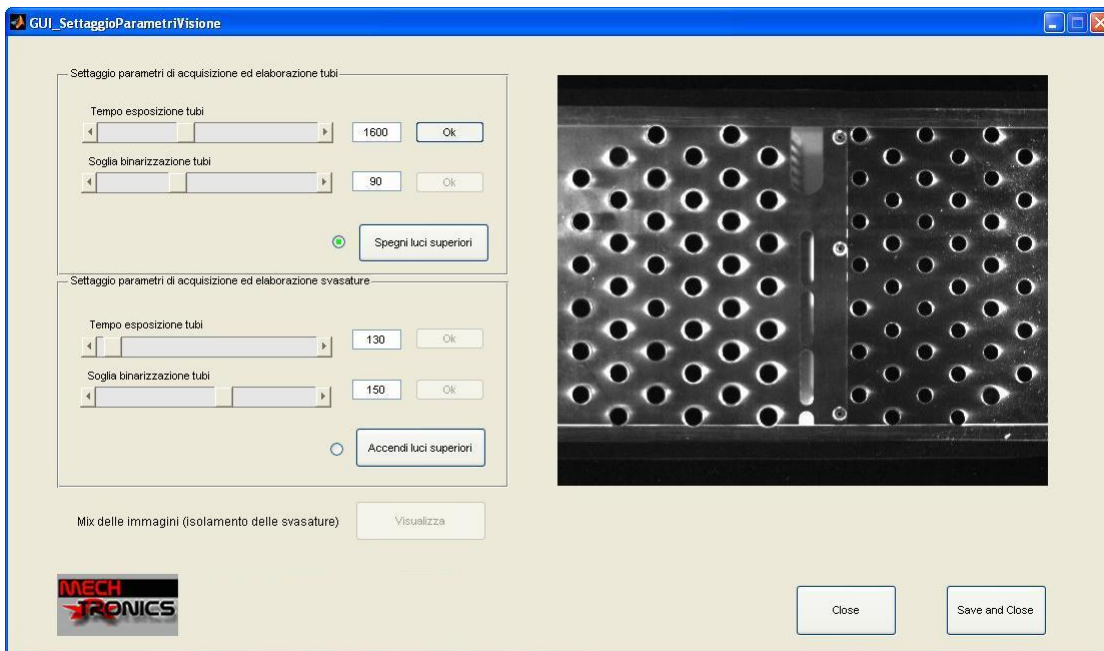


Figura 2.22: Acquisizione dell'immagine con illuminazione laterale

Ora bisogna procedere con la binarizzazione dell'immagine ottenuta, anche qui è possibile modificare la soglia di binarizzazione con lo slider chiamato "Soglia binarizzazione tubi" e premere il relativo "OK" per visualizzare il risultato ottenuto (un esempio a Figura 2.23).

A questo punto non resta che ripetere la procedura analizzata questa volta con l'illuminazione diffusa, a tal proposito si preme il pulsante "Accendi luci superiori" andando ad abilitare il tempo di esposizione e la soglia di binarizzazione per tale illuminazione. La procedura è identica alla precedente, e i risultati sono del tutto simili ai precedenti, solo che le superfici bianche risultano leggermente più ampie (infatti al contrario dell'illuminazione laterale che risalta solo il centro dei tubi, l'illuminazione diffusa esalta anche le svasature).

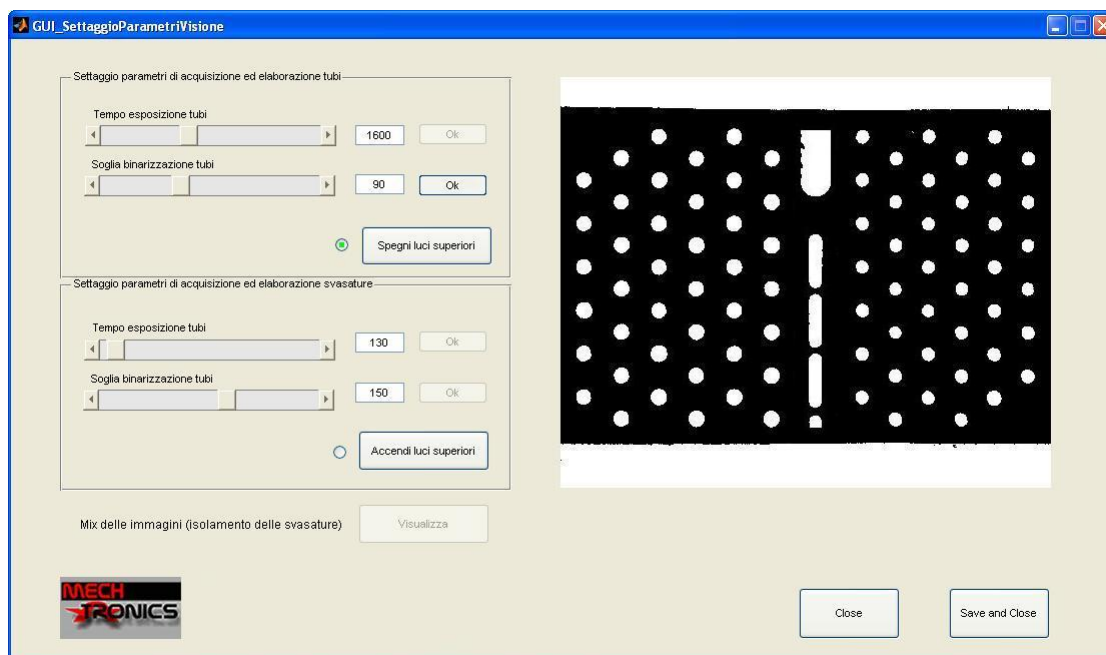


Figura 2.23: Binarizzazione dell'immagine ottenuta con illuminazione laterale

Ottenute le due immagini con le relative soglie di binarizzazione è importante avere un'anteprima della differenza delle due, infatti come si vedrà meglio nel Capitolo 4 il software di rilevazione di imperfezioni utilizza proprio queste informazioni nell'alaborazione. Basta quindi premere il pulsante "Visualizza" per ottenere il mixing desiderato (vedi Figura 2.24).

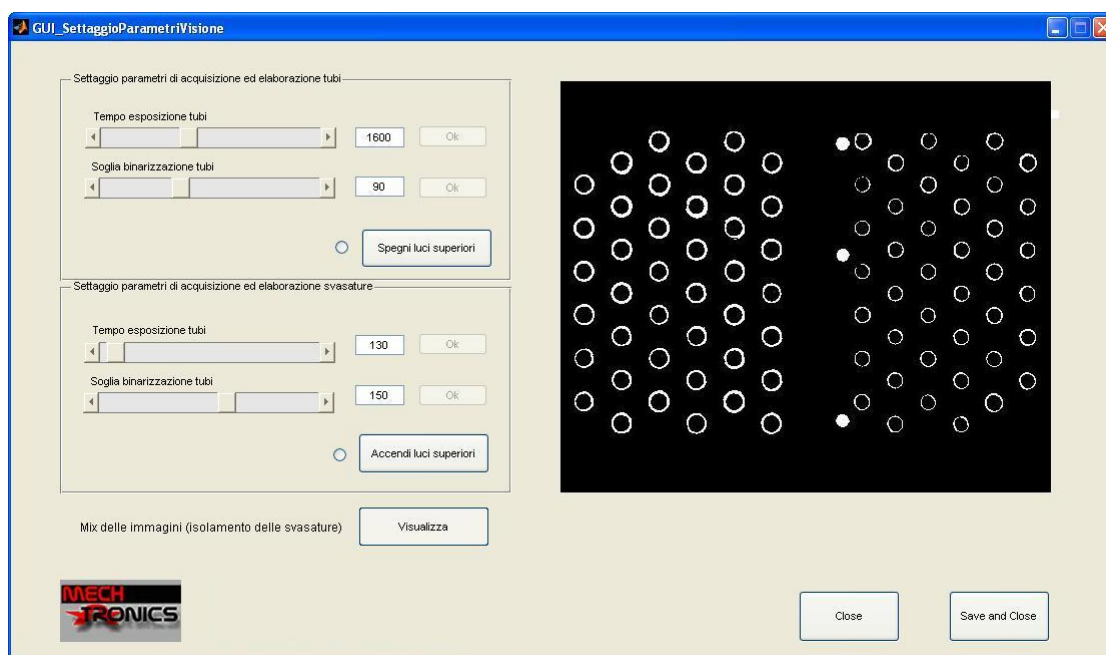


Figura 2.24: Visualizzazione delle differenze tra le due immagini binarizzate

Si osservi come tutte le svasature siano isolate dai tubi (questa informazione sarà indispensabile in seguito per determinare lo spessore medio/minimo della svasatura).

Una volta individuato il settaggio ottimo dei parametri, i valori determinati possono essere salvati all'interno di questa interfaccia per poter essere riproposti alla prossima richiesta di settaggio parametri. I dati (shutter e soglia di binarizzazione) vengono anche salvati per poter essere poi effettivamente utilizzati all'interno del ciclo principale nel momento dell'elaborazione dell'immagini.



# Capitolo 3

## Software di gestione della cella

In questo capitolo andremo ad analizzare tutta la parte software che gestisce la cella, a partire dalla gestione dei DataBase degli scambiatori e degli ordini fino all'analisi degli algoritmi che costituiscono il cuore della cella.

L'ambiente di sviluppo con il quale è stato scritto l'intero programma di gestione della cella è MatLab<sup>1</sup>, tale scelta è stata dettata dalla grande potenza di questo software, si consideri ad esempio che per l'elaborazione delle immagini MatLab fornisce una numerosa famiglia di strumenti e funzioni<sup>2</sup> (vedi Capitolo 4 per ulteriori dettagli).

### 3.1 Interfaccia Principale

Lo scambio di informazioni tra la cella e l'utente avviene grazie ad un'interfaccia principale (in seguito GULAssemblaggio) caricata automaticamente dal computer esterno al momento dell'avvio della cella la quale consente di comandare e visualizzare tutte le principali operazioni della cella. Dalla *GULAssemblaggio* (Figura 3.1) è possibile controllare direttamente l'evoluzione delle operazioni di assemblaggio degli scambiatori, dalla visualizzazione delle foto acquisite dal sistema di visione all'avanzamento della procedura di inserimenti delle curve. Da qui l'operatore è in grado di interagire con il robot richiedendo fermi macchina e ripartenze, visualizzare statistiche sull'andamento qualitativo della cella e gestire eventuali allarmi associabili a situazioni anomale nel cambio utensile, nel robot o sullo scambiatore. Dalla GULAssemblaggio è possibile richiamare altre interfacce utente (le quali consentono la gestione dei database ordini/scambiatori e il settaggio dei parametri per il riconoscimento dei difetti) grazie alla barra degli strumenti (vedi Figura 3.1).

L'organizzazione di questo capitolo prevede una discussione ed analisi delle diverse interfacce richiamabili dalla GULAssemblaggio, per poi trattare il funzionamento del ciclo principale della cella soffermandoci sul calcolo del percorso ottimo per l'inserimento delle curvette e sulla gestione delle statistiche dei difetti.

---

<sup>1</sup>Il codice presente sul controllore del robot è scritto in un linguaggio adottato da Adept denominato V+

<sup>2</sup>In questo progetto sono stati utilizzati i seguenti toolbox: *Image Acquisition Toolbox* e *Image Processing Toolbox*.

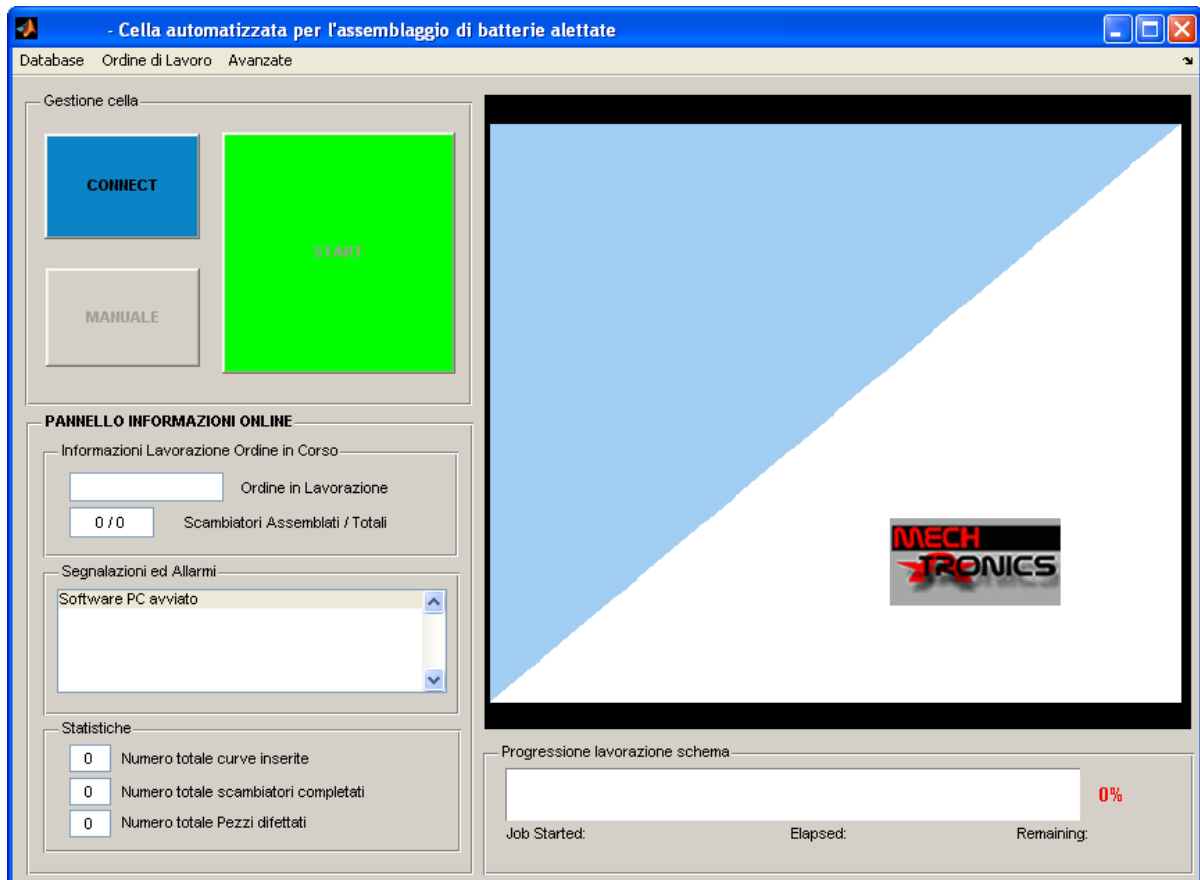


Figura 3.1: GUI\_Asemblaggio per la gestione della cella.

Come già annunciato, si possono richiamare diverse interfacce dalla barra degli strumenti, in particolare le possibilità che vengono visualizzate al click del mouse sono le seguenti:

- DataBase
  - Aggiorna
  - Mostra
- Ordine di Lavoro
  - Modifica
- Avanzate
  - Set Parametri Identificazione Difetti

Nel seguito verranno descritte tutte le possibilità riportate sopra, con lo scopo di illustrare le potenzialità offerte all'utente per osservare il funzionamento della cella e gli strumenti per interagire con la stessa.

### 3.1.1 DataBase

Selezionando “DataBase → Mostra” dalla barra degli strumenti<sup>3</sup> apparirà sullo schermo l’interfaccia riportata in figura 3.2 mentre la GULassemblaggio verrà oscurata.

L’interfaccia di gestione del DataBase permette l’acquisizione di nuovi schemi-tubi e la modifica di schemi-tubi già presenti nel database, ad esempio è possibile aggiungere o togliere le spalle presenti sullo schema.

Come da Figura 3.2 sono ben visibili due liste: sulla lista in alto, chiamata “File CAD da importare/aggiornare” sono presenti tutti gli schemi CAD non ancora importati, mentre sulla lista in basso, chiamata “Schemi presenti in DataBase” sono presenti gli schemi già importati, gli schemi quindi che insieme al file CAD hanno associato un file .m con una serie di informazioni che in seguito vedremo in dettaglio.

Per comprendere fino in fondo le funzionalità dell’interfaccia utilizziamo un caso pratico, in particolare andiamo ad utilizzare il caso visualizzato in figura 3.2 in cui sono presenti due schemi-tubi già presenti in database: “ST30180” e “ST30181”. Supponiamo di eliminare dalla lista “Schemi presenti in DataBase” lo schema-tubi ST30181 (basta premere il tasto “ELIMINA” in basso), in questo modo la scritta ST30181 comparirà nella lista “File CAD da importare/aggiornare”. Selezionando ST30181 sarà abilitato il pulsante “Importa DXF” il quale permette di caricare lo schema-tubi settando le spalle desiderate. Completata questa operazione sarà possibile visualizzare sull’interfaccia lo schema-tubi con le spalle settate e tutti i parametri salienti estratti dal file CAD come il diametro dei tubi, il numero di ranghi la geometria. Allo stesso tempo verrà abilitato il pulsante “PIANIFICA” dal quale verrà data un’anteprima del percorso previsto per l’inserimento delle curvette<sup>4</sup> (vedi Figura 3.3). Il calcolo per il percorso ottimo segue lo stesso algoritmo che verrà eseguito nella vera e propria pianificazione, algoritmo illustrato in dettaglio alla fine del capitolo.

Premendo ora il pulsante “CHIUDI” si esce dall’interfaccia andando a visualizzare nuovamente la GULAssemblaggio, tra queste operazioni viene salvato nel database in un file nominato “ST30181.m” il quale contiene al suo interno una struttura chiamata “DatiScamb” che presenta tutti i campi presenti in tabella 3.1 . In realtà i campi relativi all’output (in particolare TRJ) vengono inseriti all’inizio della lavorazione di un nuovo ordine: in tale occasione infatti vengono caricati tutti gli schemi-tubi presenti nell’ordine e viene ri-calcolato il percorso minimo (tale ricalcolo è stato pensato per motivi di sicurezza) di ognuno.

---

<sup>3</sup>Selezionando “DataBase → Aggiorna” non si fa altro che visitare le cartelle del database per controllare che ogni schema già importato abbia il relativo file CAD da cui è stato generato.

<sup>4</sup>Nell’anteprima vengono indicati con colore diverso percorsi associati a pinze diverse, e per ogni terna curvetta viene indicata anche la relativa terna inserimento (dalla quale passa chiaramente il percorso ottimo).

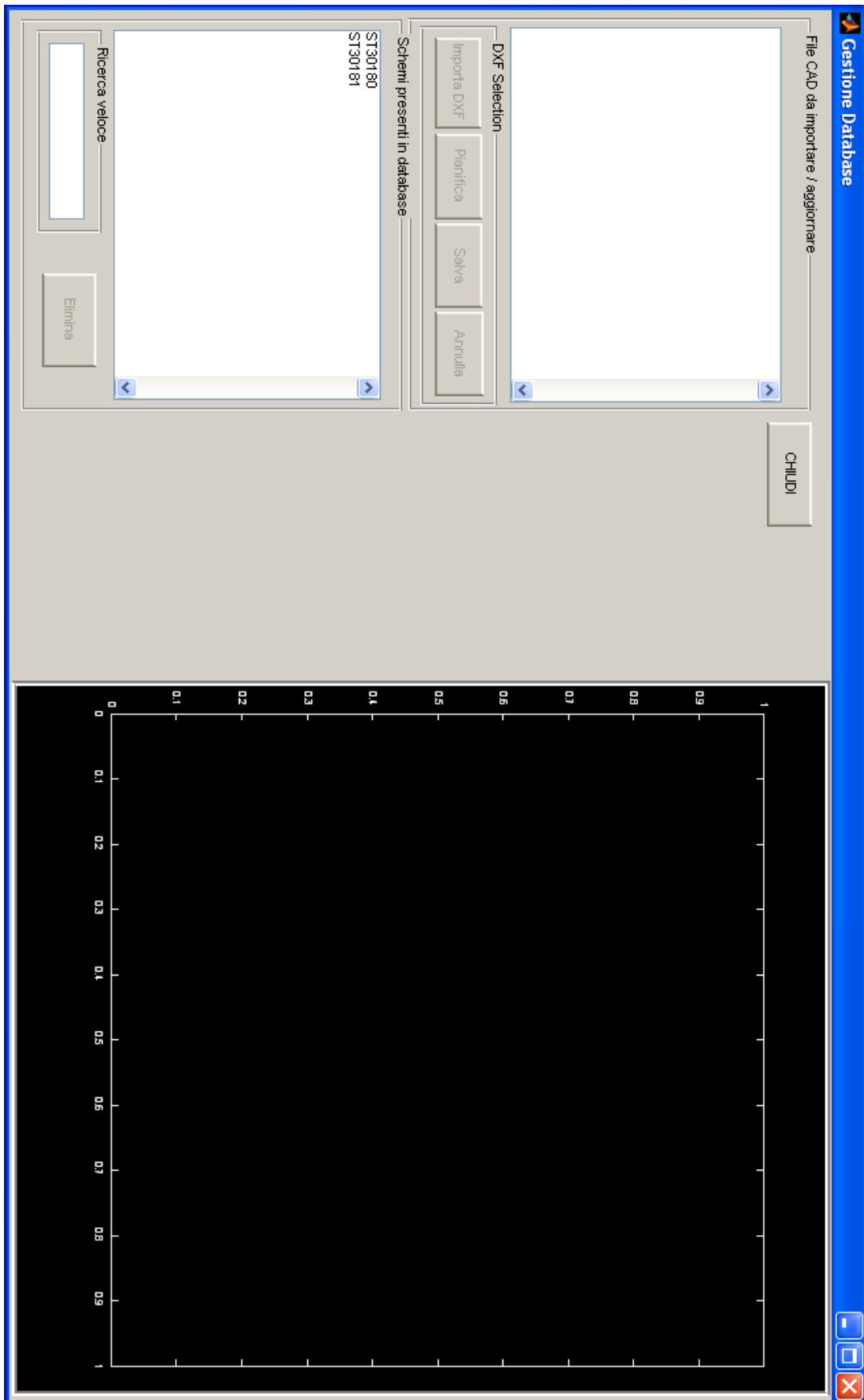


Figura 3.2: Interfaccia database

DatiScamb{i}		Descrizione	
.dati	.modello	nome del modello	
	.tipo	“equilatero” o “non equilatero”	
	.tubi	numero di tubi totale	
	.l	interassi: [25, 22.743] [mm]	
	.h	distanza tra i ranghi [mm]	
	.curv	numero di curve da 25 e 22.743mm	
	.rng	numero di ranghi	
	.tpr	numero di tubi per rango	
	.dtubo	diametro dei tubi [mm]	
	.spalla	posizione spalla nel disegno CAD	
	.hspallamax	altezza massima delle spalle [mm]	
	.M	matrice di descrizione schema-tubi: $M(i,:)=[x_{CAD}, y_{CAD}, \text{Rango}, \text{Curva}]$	
	.C	matrice di descrizione curve: $C(i,:)=[\text{Tubo1}, \text{Tubo2}, \text{Rango}, \text{Interasse}]$	
.H	matrice coordinate dei centri curve : $H(i,:)=[x_{CAD}, y_{CAD}, \text{Inclinazione}^\circ]$		
.cad	.name	nome del relativo file CAD	
	.date	data inserimento file CAD nel database	
	.bytes	dimensione del file CAD [byte]	
.output{j}	.R	matrice delle terne Robot	
	.pinza{k}	.NomePinza	nome della pinza
		.IDHardware	4bit identificativi pinza
		.Interasse	interasse curve della pinza
		.DiamTubo	diametro dei tubi delle curve della pinza
		.dx	parametro pinza
		.dy	parametro pinza
		.Theta	parametro pinza
	.T_ct	matrice trasf. curva/tool	
	.TRJ	matrice traiettoria ottima: $TRJ(i,:)=[x, y, \text{theta}, \text{curv}, R],  S, $	
	.s_tot	spazio percorso seguendo TRJ [mm]	
	.theta_tot	rotazione totale pinza seguendo TRJ [°]	
.t_tot	tempo inserimento curve [s]		

Tabella 3.1: Struttura del database DatiScamb.

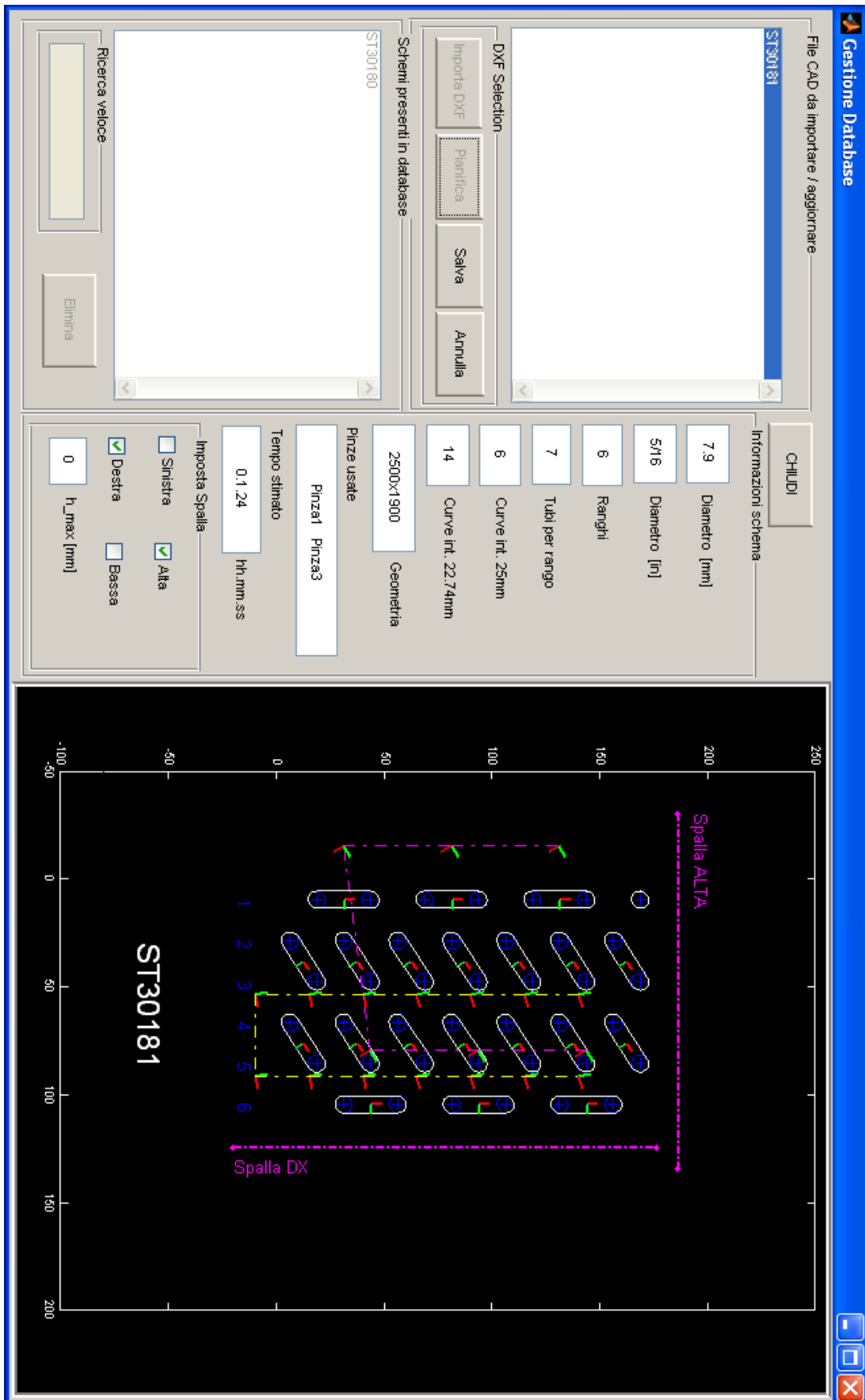


Figura 3.3: Importazione dello schema-tubi ST30181

### 3.1.2 Ordine di Lavoro

Selezionando “Ordine di Lavoro → Modifica” dalla barra degli strumenti apparirà sullo schermo l’interfaccia riportata in figura 3.4 mentre la GUI\_assemblaggio verrà oscurata.

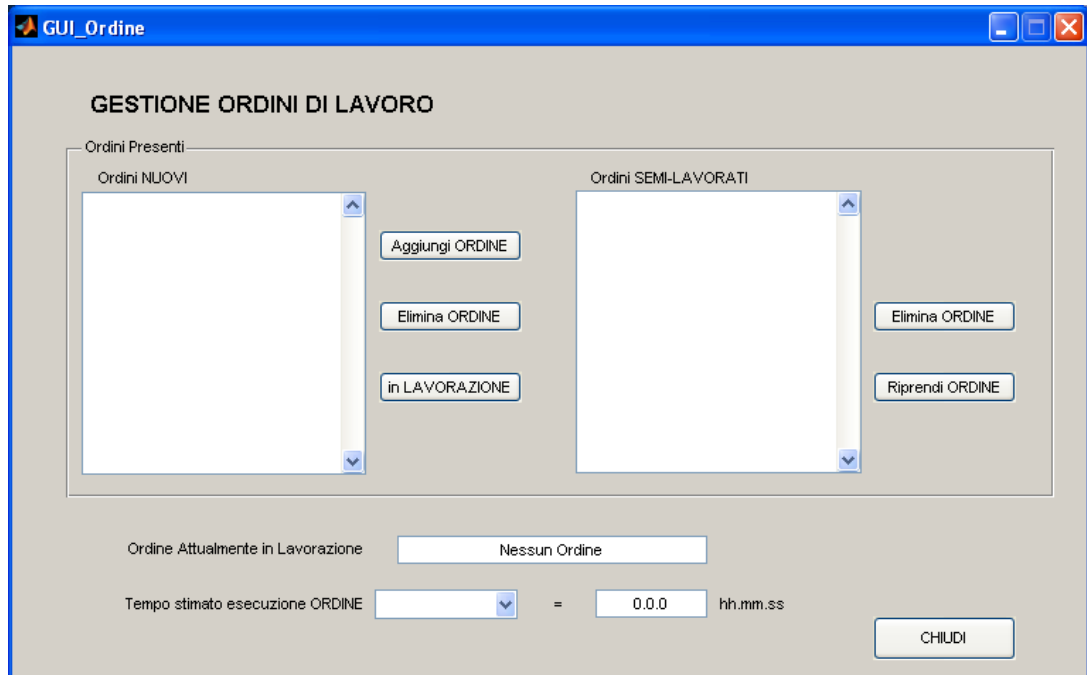


Figura 3.4: Interfaccia per la gestione degli Ordini di Lavoro

La realizzazione di questa interfaccia è nata dalla volontà dell’azienda committente di poter programmare la lavorazione assegnata<sup>5</sup> alla cella e di poter avere una stima del tempo di esecuzione delle diverse commesse.

Osservando l’interfaccia in figura sono ben distinguibili due liste (al momento vuote): Ordini NUOVI e Ordini SEMI-LAVORATI.

In Ordini NUOVI sono presenti gli ordini che non sono mai passati in lavorazione, quindi quegli ordini che sono presenti nella programmazione della lavorazione ma non sono ancora stati iniziati. Esistono due modi per inserire un nuovo ordine in questa lista: il più semplice è selezionare il pulsante “Aggiungi ORDINE” presente nell’interfaccia grazie al quale comparirà una nuova interfaccia (vedi Figura 3.5) dalla quale sarà possibile inserire il codice dell’ordine, il numero di pezzi previsti e i modelli di schema-tubi presenti nello scambiatore da assemblare.

<sup>5</sup>La richiesta è quella di impostare in gestionale un set di ordini da lavorare per poi lasciare all’utente la facoltà di scegliere la sequenza di lavorazione.

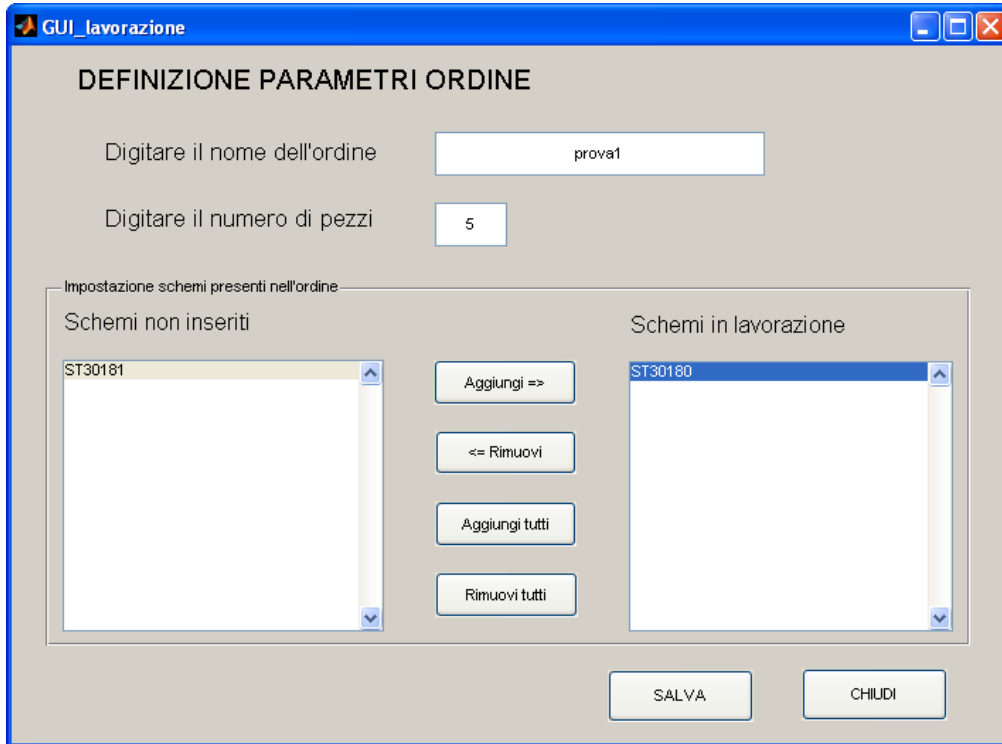


Figura 3.5: Interfaccia per la creazione di un nuovo ordine

L'altro metodo consiste nell'inserire un file di testo (estensione .txt) all'interno di una cartella messa in condivisione (con possibilità di lettura e scrittura) con il reparto gestionale, il file di testo (non ha importanza il nome del file) deve contenere al suo interno le stesse informazioni necessarie nell'altro metodo: codice dell'ordine, numero di pezzi previsti e modelli schema-tubi presenti; un esempio è riportato in figura 3.6:

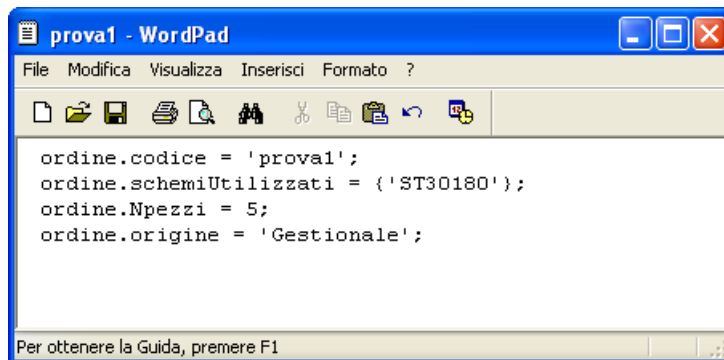


Figura 3.6: File testo per la creazione di un nuovo ordine

La differenza tra i due sistemi è l'origine dell'ordine, infatti con il primo metodo il programma oltre alle informazioni inserite aggiunge un campo chiamato "Origine" e al suo interno scrive "Non Gestionale" mentre nel secondo metodo tale campo (vedi figura 3.6) ha al suo interno il testo "Gestionale".



Se l'origine dell'ordine è "Gestionale" allora non è possibile eliminare l'ordine dall'interfaccia di figura 3.4, in caso contrario l'eliminazione da tale interfaccia è consentita. Tale funzionamento garantisce che l'operatore non possa manomettere la programmazione della lavorazione stabilita dall'ufficio gestione.

Supponiamo di avere a questo punto due ordini NUOVI, come si può facilmente vedere dalla figura 3.7 è possibile dall'interfaccia avere una stima del tempo di esecuzione degli stessi selezionando l'ordine di interesse dal menù a tendina presente in basso.

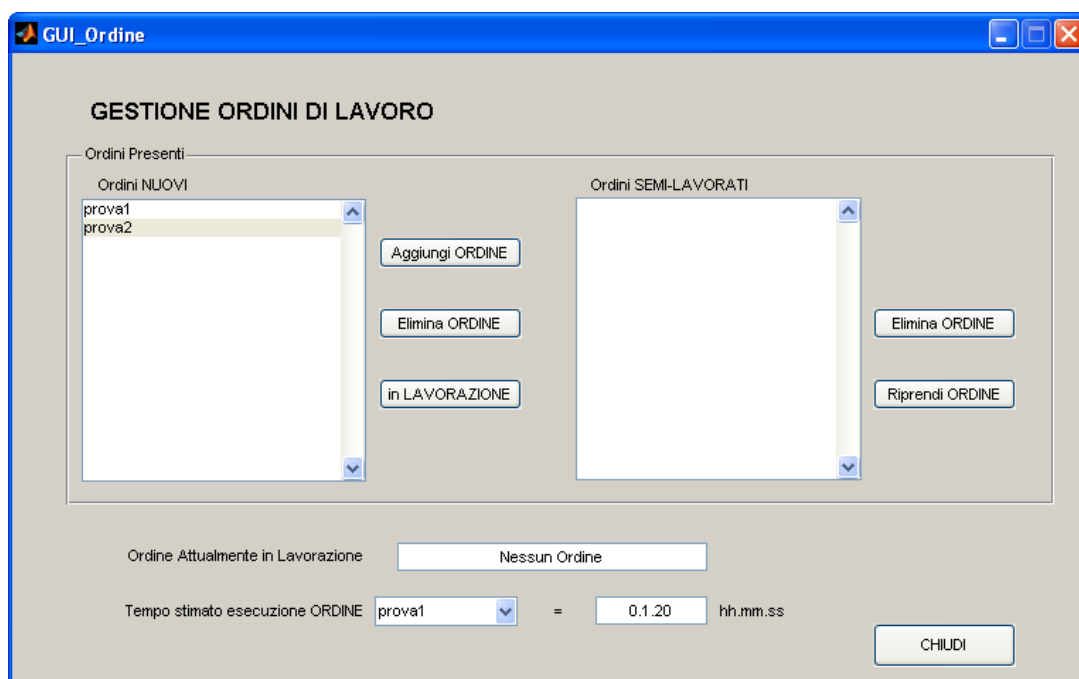


Figura 3.7: Interfaccia per la gestione degli ordini di lavoro

Dalla figura si può anche osservare come nessun ordine sia ancora in lavorazione, dobbiamo quindi selezionare uno degli ordini disponibili (*prova1* o *prova2*) e cliccare sul pulsante "In LAVORAZIONE": a questo punto un messaggio a video confermerà l'operazione eseguita e l'interfaccia sarà come riportato in figura 3.8.

Supponiamo di mettere in lavorazione *prova2*: a questo punto, cliccando sul pulsante "CHIUDI", è possibile uscire dall'interfaccia tornando alla GUI\_Inserimento iniziando la lavorazione dell'ordine *prova2*; altrimenti è possibile mettere in lavorazione l'ordine *prova1* cliccando sul pulsante "In LAVORAZIONE"; ancora, è possibile rimettere in lavorazione *prova2* selezionando tale nome e cliccando su "Riprendi ORDINE".

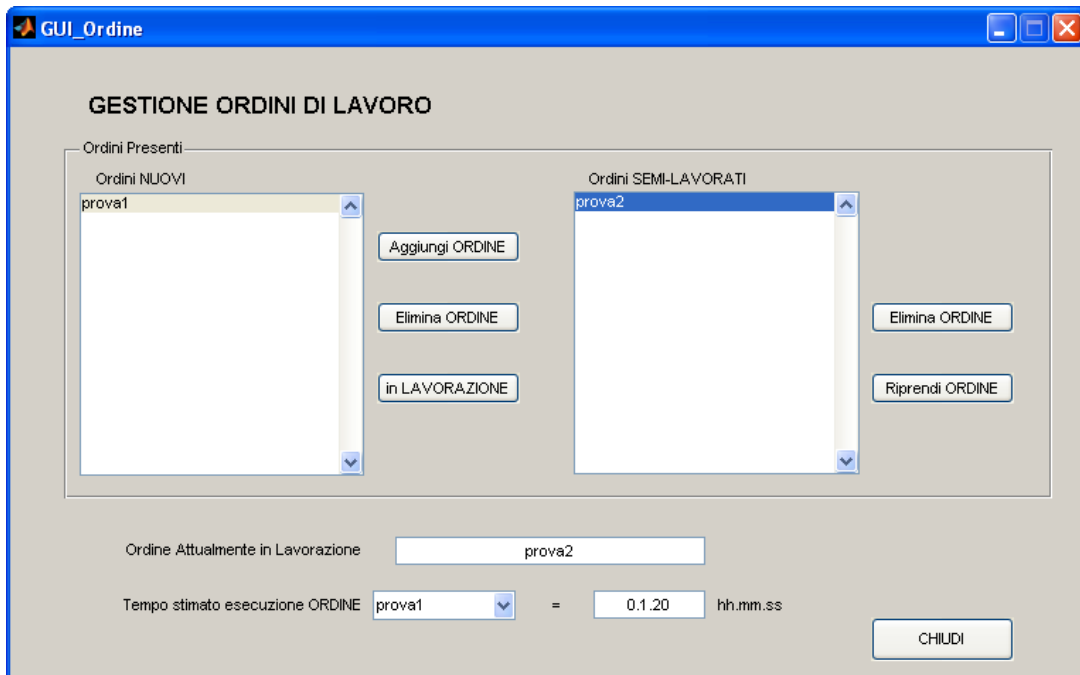


Figura 3.8: Interfaccia per la gestione degli ordini di lavoro

Chiaramente le possibilità sono infinite, per non entrare in confusione la finestra "Ordine attualmente in Lavorazione" riporta l'ordine che verrà lavorato al momento dell'uscita da questa interfaccia.

Si osservi che è possibile accedere a questa interfaccia in qualsiasi momento della lavorazione: in particolare è possibile iniziare un ordine, interromperlo e iniziarne uno nuovo: in tal caso l'ordine incompleto sarà nella lista "Ordini SEMI-LAVORATI", dalla quale scomparirà solamente al completamento dell'ultimo pezzo dell'ordine.

### 3.1.3 Set Parametri Difetti

Selezionando “Avanzate → Set Parametri Identificazione Difetti” dalla barra degli strumenti apparirà sullo schermo l’interfaccia riportata in figura 3.10 mentre la GUI\_assemblaggio verrà oscurata.

In questa interfaccia è possibile modificare i parametri di identificazione dei difetti dello scambiatore e visualizzare gli effetti di tali modifiche sul riconoscimento difetti dello scambiatore in analisi: infatti una volta settati i parametri (che è possibile riportare a valori di default) premendo sul tasto “Visualizza” si va ad eseguire esattamente la stessa parte di programma che viene eseguita nel ciclo principale al momento dell’analisi dei difetti, questo per avere un’anticipazione su quello che sarà il risultato dell’elaborazione.

Premendo il pulsante “Default” è possibile ristabilire i parametri con dei valori che in media (tests in Laboratorio su diversi scambiatori) non conducono al riconoscimento di difetti da parte del software.

Premendo il pulsante “Chiudi” si esce dall’interfaccia (un messaggio chiede la conferma di uscita senza salvare) ritornando alla GUI\_Inserimento, mentre con Save vengono salvati i parametri settati che saranno ripresentati alla prossima chiamata a questa interfaccia.

Da questa interfaccia si può selezionare (dalla barra degli strumenti) la voce “Avanzate → Set Parametri Telecamera” che richiama un’ulteriore interfaccia (Figura 3.9) dalla quale è possibile modificare i parametri della telecamera per l’acquisizione delle immagini (shutter) e per l’elaborazione delle stesse (soglia di binarizzazione). Tale interfaccia è stata già analizzata in dettaglio al Capitolo 1 parlando della calibrazione.

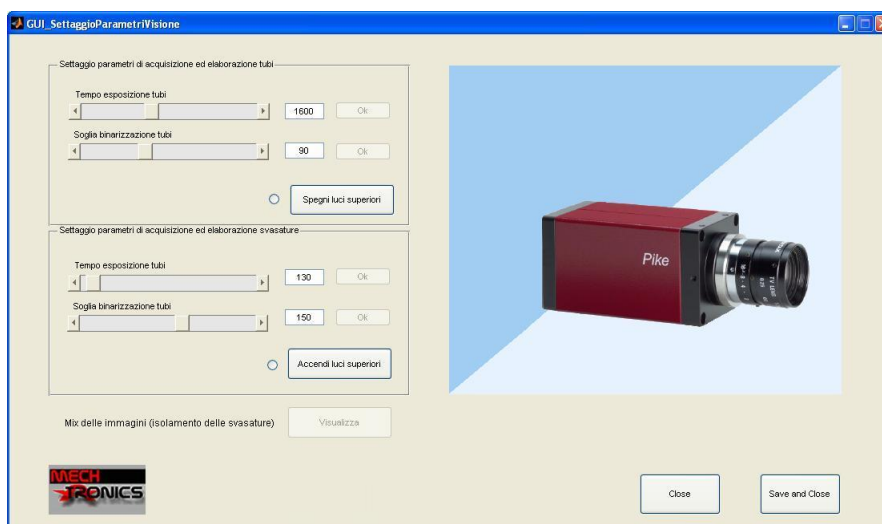


Figura 3.9: GUI di settaggio dei parametri di acquisizione ed elaborazione

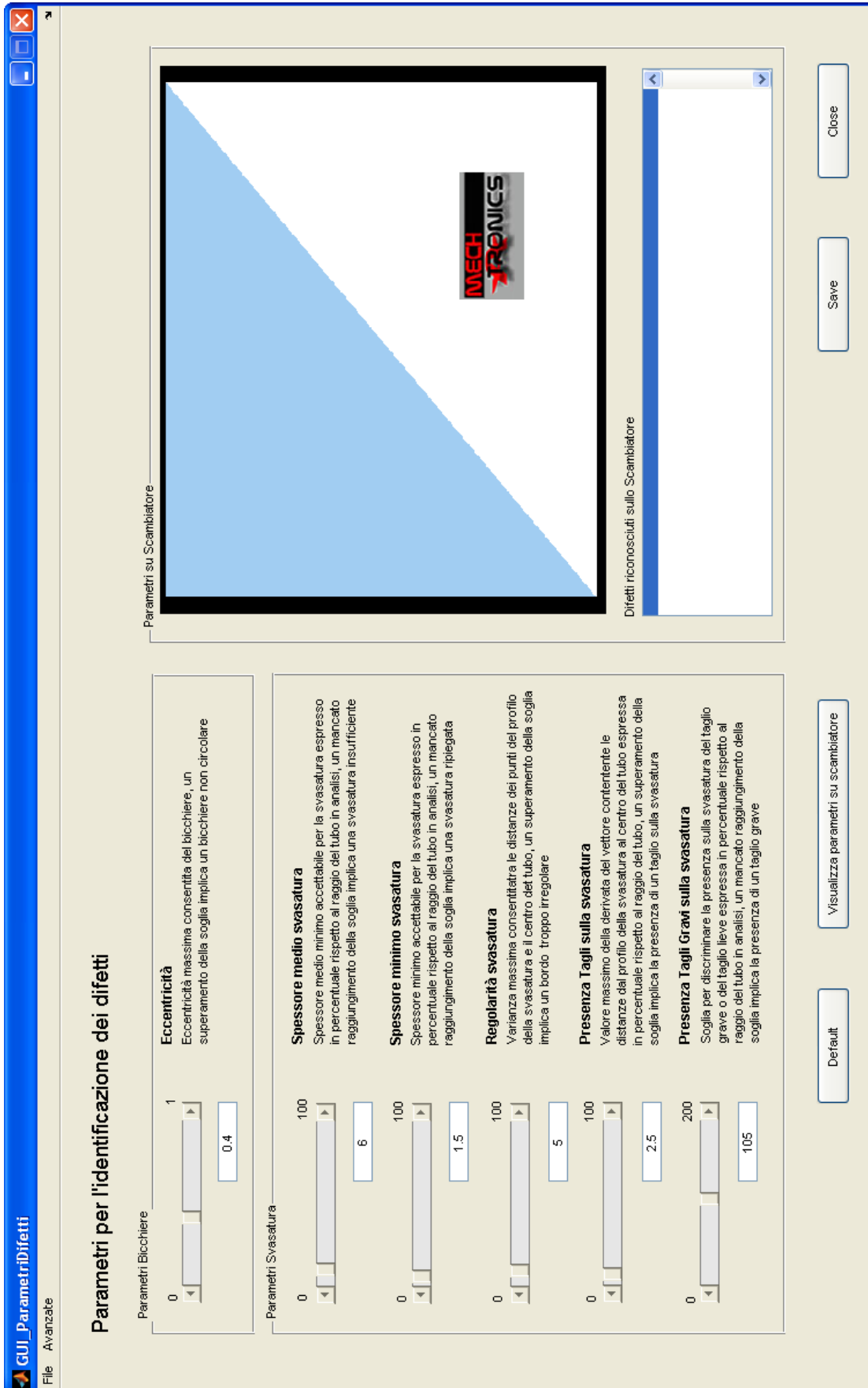


Figura 3.10: Interfaccia per il set dei parametri difetti

Prima di procedere è bene dare una spiegazione sommaria del significato dei diversi parametri presenti sull'interfaccia (una trattazione completa sarà fatta alla fine del Capitolo 4):

- *ECCENTRICITA'*:  
Eccentricità massima consentita del bicchiere (i valori estremi sono: 0 = cerchio perfetto, 1 = segmento), un superamento della soglia implica un bicchiere non circolare.
- *SPESSORE MEDIO SVASATURA*:  
Spessore medio minimo accettabile per la svasatura espresso in percentuale rispetto al raggio del tubo in analisi, un mancato raggiungimento della soglia implica una svasatura insufficiente.
- *SPESSORE MINIMO SVASATURA*:  
Spessore minimo accettabile per la svasatura espresso in percentuale rispetto al raggio del tubo in analisi, un mancato raggiungimento della soglia implica una svasatura insufficiente.
- *REGOLARITA' SVASATURA*:  
Varianza massima consentita tra le distanze dei punti del profilo della svasatura e il centro del tubo, un superamento di tale soglia implica un bordo troppo irregolare.
- *PRESENZA TAGLI SULLA SVASATURA*:  
Valore massimo della derivata del vettore contenente le distanze dal profilo della svasatura al centro del tubo espressa in percentuale rispetto al raggio del tubo, un superamento della soglia implica la presenza di un taglio sulla svasatura.
- *PRESENZA TAGLI GAVI SULLA SVASATURA*:  
Soglia per discriminare la presenza sulla svasatura del taglio grave o del taglio lieve espressa in percentuale rispetto al raggio del tubo in analisi, un mancato raggiungimento della soglia implica la presenza di un taglio grave.

In Figura 3.11 è riportato il risultato dell'elaborazione avendo abbassato il parametro "Regolarità svasatura": si noti come nella lista presente nell'interfaccia siano comparsi ora il numero dei due tubi sui quali è stato riscontrato il difetto ed il relativo schema di appartenenza.

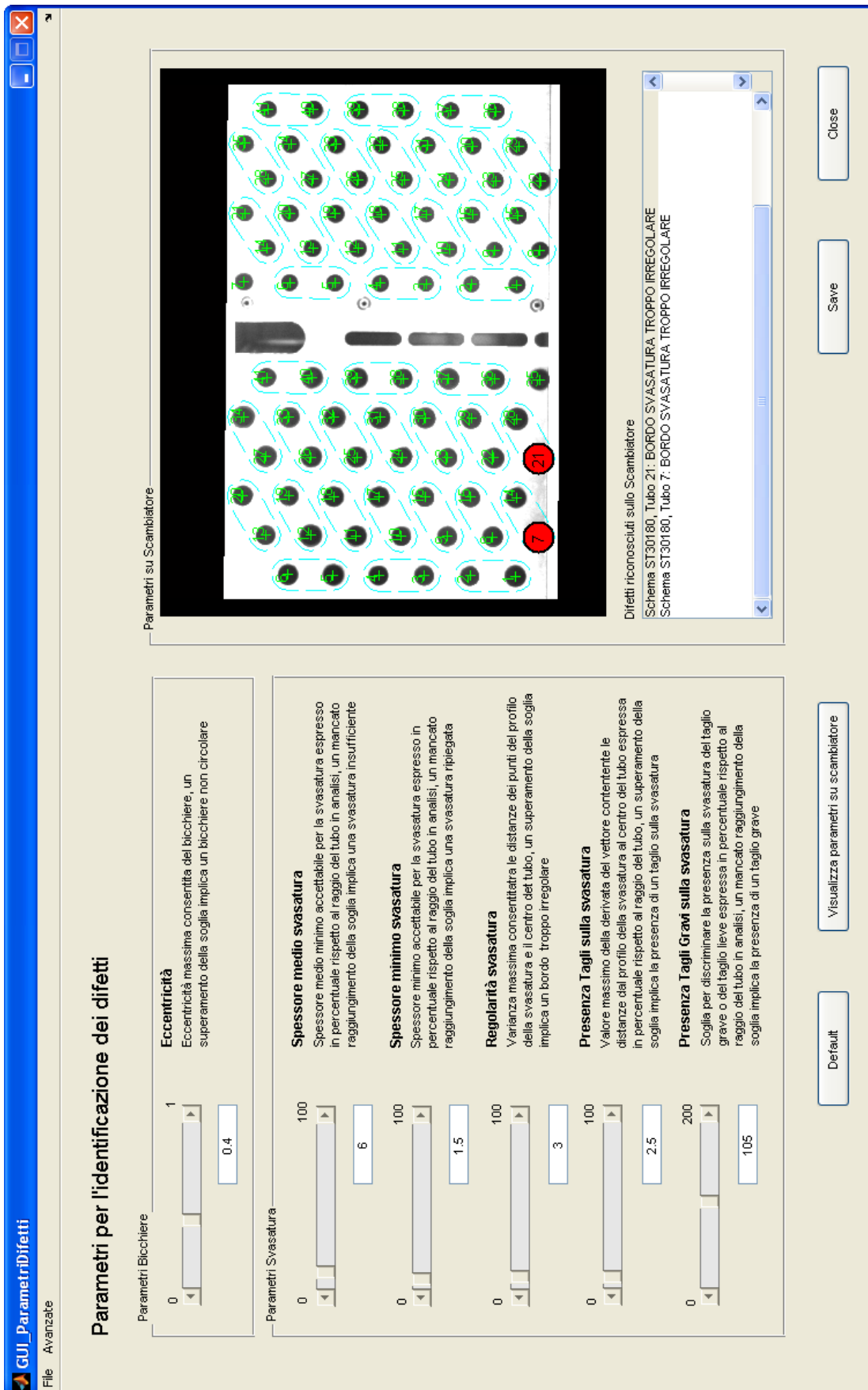


Figura 3.11: Risultato di un set dei parametri difetti

## 3.2 Ciclo Principale

Il funzionamento della cella si può dividere in due fasi che procedono di pari passo: la comunicazione con il controllore e le elaborazioni necessarie per l'assemblamento dello scambiatore. La prima parte è stata illustrata nel capitolo 2 nella parte relativa alla comunicazione tra MatLab e il controllore, proseguiamo a questo punto con l'analisi delle elaborazioni: a tal proposito è bene riassumere le fasi che identificano l'assemblaggio.

L'assemblaggio comincia con l'introduzione del pacco alettato da assemblare da parte del sistema di alimentazione della cella (tavola rotante) all'interno dello spazio di lavoro. Fissato lo scambiatore nella posizione di assemblaggio il robot viene condotto in prossimità del magazzino utensili, al di fuori dal campo visivo della telecamera, così da consentire al sistema di visione di acquisire le immagini del pezzo.

In questa fase entrambi i sistemi di illuminazione sono già attivi, e si procede quindi all'acquisizione dell'immagine (l'immagine acquisita e considerata ottenuta con l'illuminazione diffusa, infatti in presenza di tale illuminazione le luci laterali non apportano modifiche all'immagine). A questo punto viene spenta l'illuminazione diffusa e di nuovo viene acquisita l'immagine (immagine con illuminazione laterale). Prima di procedere all'elaborazione delle immagini viene ri-accesa l'illuminazione diffusa, così da permettere che le lampade siano in temperatura alla prossima acquisizione.

Grazie alle due immagini acquisite il software identifica gli schemi-tubi presenti nello scambiatore e per ogni tubo isola una ROI (Region Of Interest) che sarà necessaria in seguito per la determinazione con massima precisione dei centri-tubi<sup>6</sup>.

Riconosciuti gli schemi-tubi il sistema carica dal database la sequenza ottima di inserimento (tale sequenza è ri-calcolata all'inizio di ogni ordine, l'algoritmo utilizzato sarà presentato alla fine del capitolo).

Con sequenza intendiamo una serie di locazioni riferite ad un unico tipo di curvetta (e quindi di pinza), avere 3 sequenze significa quindi che per l'assemblaggio saranno necessarie 3 pinze. Il sistema chiede al controllore la pinza innestata sulla flangia e le curvette rimanenti: se la pinza innestata può soddisfare ad una delle sequenze programmate allora si parte da quella, altrimenti se la pinza innestata appartiene ad una sequenza programmata ma non ha abbastanza curvette per soddisfarla completamente allora tale sequenza viene iniziata per ultima (per permettere nel frattempo che la pinza si riempia di curvette), in caso contrario le sequenze vengono ordinate in base al numero di curvette da inserire (in ordine decrescente).

Selezionata una sequenza si procede ad analizzare la sua prima curvetta (verifica di inseribilità). Questa verifica viene eseguita dal software di visione andando innanzitutto a verificare che nei relativi due tubi non vi sia già inserita una curva e provvedendo successivamente ad analizzarne le ROI delle imboccature alla ricerca

---

<sup>6</sup>La fase di identificazione dello scambiatore può essere fatta sottocampionando le immagini acquisite (avendo quindi immagini con risoluzione inferiore) con lo scopo di velocizzare l'elaborazione).

di eventuali difetti. Se la curva risulta priva di gravi difetti e quindi inseribile, per poter guidare il robot nell'operazione di inserimento la visione fornisce al software di gestione della cella le coordinate esatte della posizione dei due centri-tubo espresse nel sistema di riferimento del robot (grazie alla calibrazione, Capitolo 2) dall'analisi delle ROI dei tubi acquisite con l'illuminazione laterale.

Dopo aver portato a termine o meno l'inserimento dell' $i$ -esima curva il ciclo passa alla successiva curva della sequenza di inserimento; al completamento della sequenza il sistema provvede a cambiare la pinza ed iniziare una nuova sequenza.

Una volta completato l'inserimento di tutte le curve inseribili nello schema viene fatto l'aggiornamento delle *Matrici Difetti*<sup>7</sup>, tali matrici (verranno illustrate in dettaglio nel seguito) hanno lo scopo di tenere memoria di tutti i difetti riscontrati sui diversi tubi. A questo punto vengono acquisite due ulteriori immagini (una con illuminazione laterale ed una diffusa) per verificare il corretto assemblaggio dei diversi schemi-tubi (per ogni schema-tubi si confronta il numero di curvette ideali con il numero di curvette rilevato dalle immagini acquisite). Se l'assemblaggio non è completo (ad esempio una curvetta non è stata inserita perchè i tubi relativi erano difettati) si procede con l'analisi delle Matrici Difetti per discriminare se fermare l'impianto o meno: infatti se i difetti riscontrati nell'assemblaggio sono associati a dei tubi i quali presentano troppo spesso quel tipo di difetti allora viene lanciato un allarme per avvisare l'utente di un possibile problema nella mandrinatura dei tubi.

L'assemblaggio può essere non completo anche dovuto al fatto che una curvetta è inserita in modo errato (ad esempio da parte dell'operatore prima di inserire lo scambiatore nell'area di lavoro).

L'assemblaggio può ora considerarsi concluso, viene messa in movimento la tavola rotante la quale permette l'uscita dello scambiatore assemblato dall'area di assemblaggio e ne fa entrare uno nuovo.

Il ciclo di inserimento delle curve è stato pianificato in modo tale da poter eseguire la parte di elaborazione delle ROI (analisi dei difetti e misura dei centri-tubo) in **tempo mascherato**, cioè contemporaneamente all'operazione di inserimento della curva. In questo modo, se il tempo di calcolo necessario all'elaborazione di una singola curva non supera il tempo richiesto al robot per il suo inserimento, il tempo necessario alla fase di inserimento delle curve dipende solamente dalle prestazioni del robot. Dovendo elaborare l'immagine della  $i$ -esima curva necessariamente prima di procedere al suo eventuale inserimento, per poter operare in tempo mascherato risulta inevitabile sfalsare le due fasi di elaborazione ed inserimento delle curve: mentre il robot provvede all'inserimento dell' $i$ -esima curva il sistema di visione deve già procedere con l'elaborazione della successiva curva da inserire secondo la sequenza di inserimento prestabilita.

---

<sup>7</sup>Ogni schema-tubi ha associato 2 *Matrici Difetti*: una tiene memoria dei difetti riscontrati nell'ordine di lavoro, l'altra tiene memoria dei difetti riscontrati nello scambiatore presente nel database (tale matrice può essere modificata da diversi ordini). E' proprio quest'ultima matrice che si utilizza in caso di assemblaggio scorretto per determinare se arrestare l'impianto o meno.



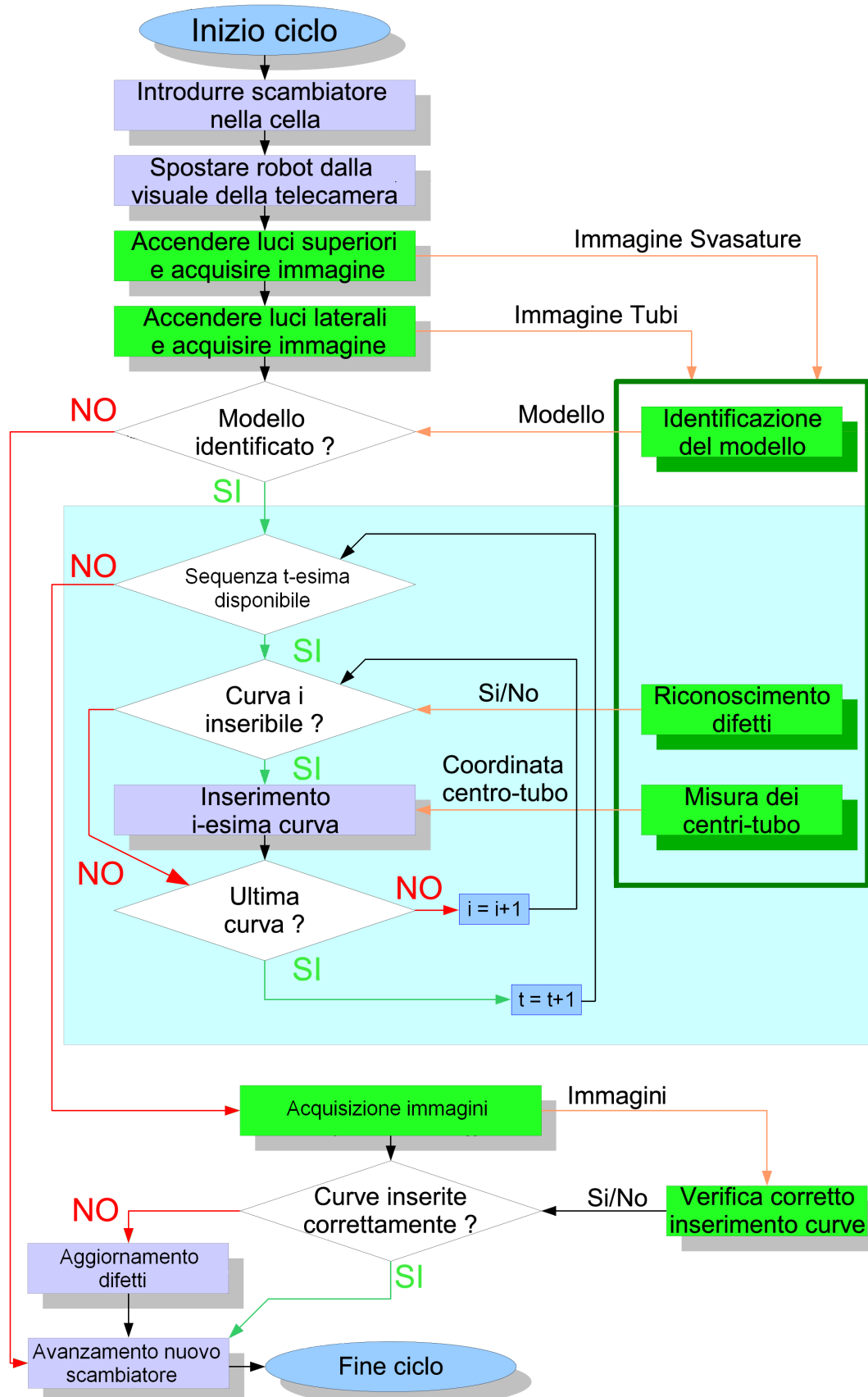


Figura 3.12: Ciclo di funzionamento per l'assemblaggio dello scambiatore

### 3.2.1 Calcolo percorso ottimo

La minimizzazione del tempo ciclo ci ha visto impegnati nella ricerca di un algoritmo per il calcolo del percorso minimo che fosse efficiente ed in particolare che non occupasse troppe risorse di memoria.

La scelta si è fermata su un algoritmo greedy che ha portato ad ottime prestazioni in termini di velocità di esecuzione (con un normale desktop siamo sotto al centesimo di secondo) e a eccellenti prestazioni in termini di risorse memoria, utilizzando infatti solo una matrice quadrata di dimensioni pari al doppio del numero delle curvette da inserire (per permettere di considerare per ogni curvetta la sua simmetrica, cioè la curvetta girata di 180°).

Andiamo a costruire la matrice delle distanze tra ogni curvetta (comprese le curvette simmetriche) e tutte le altre, tale distanza tiene conto sia della lunghezza del segmento che unisce i baricentri delle diverse curvette sia la rotazione per passare da una all'altra attraverso un indice di costo: siano ad esempio  $(x_1, y_1, \gamma_1)$  le coordinate del baricentro della  $i$ -esima curvetta e il rispettivo angolo di rotazione<sup>8</sup> e siano  $(x_2, y_2, \gamma_2)$  le coordinate della  $j$ -esima curvetta: all'interno della matrice distanze avremo in posizione  $(i, j)$  e  $(j, i)$  un numero indicante la loro distanza dato da:

$$\alpha \left( \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \right) + \beta (|\gamma_1 - \gamma_2|)$$

dove  $\alpha$  e  $\beta$  possono essere modificati per dare maggiore peso alla distanza o alla rotazione.

Procediamo con l'analisi dell'algoritmo, supponiamo di avere solo 3 curvette da inserire (la matrice distanze avrà dimensione 6 considerando le simmetriche), indicheremo attraverso un apice  $d$  la configurazione diritta e  $s$  la configurazione simmetrica. La matrice distanze sarà del tipo:

	$A_d$	$B_d$	$C_d$	$A_s$	$B_s$	$C_s$
$A_d$	0	1	2	3	4	5
$B_d$	1	0	6	4	3	2
$C_d$	2	6	0	1	5	3
$A_s$	3	4	1	0	2	6
$B_s$	4	3	5	2	0	1
$C_s$	5	2	3	6	1	0

Tabella 3.2: Esempio di Matrice distanze

Inizializziamo l'algoritmo: nessuna curvetta può muoversi verso se stessa, il che comporta che la diagonale di Tabella 3.2 viene posta ad infinito. Per la stessa ragione possiamo affermare che nessuna curvetta può muoversi verso la sua versione ruotata di 180°: per questa ragione eliminiamo, per ogni riga, l'elemento

<sup>8</sup>Si osservi che tali coefficienti sono espressi rispetto al sistema di riferimento WORLD del robot, in particolare  $x, y, \gamma$  non sono altro che la prima, la seconda e l'ultima coordinata presente nella sestupla che forma una locazione robot:  $[x y z \alpha \beta \gamma]$  essendo le altre coordinate fissate a priori.

sulla colonna relativo alla curvetta ruotata. La matrice delle distanze sarà così inizializzata:

	$A_d$	$B_d$	$C_d$	$A_s$	$B_s$	$C_s$
$A_d$	$\infty$	1	2	$\infty$	4	5
$B_d$	1	$\infty$	6	4	$\infty$	2
$C_d$	2	6	$\infty$	1	5	$\infty$
$A_s$	$\infty$	4	1	$\infty$	2	6
$B_s$	4	$\infty$	5	2	$\infty$	1
$C_s$	5	2	$\infty$	6	1	$\infty$

Tabella 3.3: Inizializzazione Algoritmo

Ora viene calcolato il minimo percorso partendo dalla curvetta  $A_d$ , poi il minimo partendo da  $B_d$  e così per tutte e 6 le curve, alla fine si confrontano i 6 diversi costi minimi (ottenuti partendo dalle diverse curve) e si sceglie il più piccolo.

Illustriamo in seguito il calcolo del minimo percorso partendo dalla curvetta  $A_d$ , il calcolo partendo dalle altre curve è del tutto identico.

Innanzitutto stiamo analizzando la curvetta  $A_d$  e quindi dobbiamo impedire che in futuro analizzando un'altra curvetta si possa ritornare su  $A_d$  (non ha senso tornare indietro e ripassare sulle curve già inserite): a tal proposito poniamo la colonna relativa ad  $A_d$  ad infinito. Per la stessa ragione  $A_s$  non verrà mai raggiunta: a tal proposito poniamo la riga e la colonna di  $A_s$  ad infinito, in modo che nessuna altra curvetta vada nel percorso minimo a selezionarla come successiva.

Prendiamo ora la prima riga (è la riga relativa ad  $A_d$ ), dobbiamo trovare il minimo: la scelta ricade su  $B_d$  (il costo è pari ad 1, il minore in tutta la riga). L'aggiornamento si conclude quindi ponendo tutti gli altri elementi sulla riga (relativa ad  $A_d$ ) ad infinito. La situazione sarà come quella riportata in Tabella 3.4

	$A_d$	$B_d$	$C_d$	$A_s$	$B_s$	$C_s$
$A_d$	$\infty$	1	$\infty$	$\infty$	$\infty$	$\infty$
$B_d$	$\infty$	$\infty$	6	$\infty$	$\infty$	2
$C_d$	$\infty$	6	$\infty$	$\infty$	5	$\infty$
$A_s$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
$B_s$	$\infty$	$\infty$	5	$\infty$	$\infty$	1
$C_s$	$\infty$	2	$\infty$	$\infty$	1	$\infty$

Tabella 3.4: Fase 1 dell'algoritmo

La seconda fase dell'algoritmo è del tutto identica. Abbiamo stabilito che la posizione attuale è  $B_d$ : poniamo ad infinito la colonna relativa (in tal modo viene cancellato l'1 colorato in blu nella tabella 3.4 per questo è bene salvare i dati sul percorso fatto: la distanza percorsa e le curve incontrate. Nella tabella lasceremo il valore in blu a scopo riepilogativo) ed eliminiamo la riga e colonna relativa a  $B_s$ .

Andiamo ora sulla riga relativa a  $B_d$  e troviamo il minimo: la scelta ricade su  $C_s$ . Come in precedenza poniamo ad infinito gli altri elementi della riga relativa a  $B_d$  diversi da  $C_s$ . La matrice a questo punto è così formata:

	$A_d$	$B_d$	$C_d$	$A_s$	$B_s$	$C_s$
$A_d$	$\infty$	1	$\infty$	$\infty$	$\infty$	$\infty$
$B_d$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	2
$C_d$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
$A_s$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
$B_s$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
$C_s$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$

Tabella 3.5: Fase 2 dell'algoritmo

A questo punto abbiamo stabilito che la posizione attuale è  $C_s$ : dopo le solite operazioni di cancellazione riga/colonna, andando a calcolare il minimo osserviamo che il risultato è infinito: questo è indice che l'algoritmo è terminato.

Abbiamo ottenuto che il percorso minimo partendo da  $A_d$  è:  $A_d \rightarrow B_d \rightarrow C_s$  comportando un costo totale di 3 (dato da 1+2). Calcoliamo ora il costo minimo partendo dalle 5 rimanenti curvette e poi scegliamo il percorso che ha generato il costo totale inferiore.

### 3.2.2 Gestione delle Matrici Difetti

Le matrici difetti non sono altro che matrici atte alla memorizzazione dei difetti incontrati lungo diversi assemblaggi. Ne esistono di due tipi: matrice difetti dell'ordine e matrice difetti dello schema-tubi. Abbiamo creato due matrici diverse per soddisfare a due problemi ben precisi: da un lato c'era la necessità di memorizzare per ogni ordine il numero di difetti incontrato sui diversi tubi, dall'altro c'era la necessità di monitorare ogni schema-tubi in modo da rilevare la presenza di difetti troppo frequenti che indicassero probabili problemi nella fase di mandrinatura dei tubi. Le matrici presentano ad ogni modo una struttura simile: ogni matrice difetti si riferisce ad un singolo schema-tubi e ha tante righe quanti sono i tubi dello schema e 5 colonne poichè sono 5 i difetti che l'analisi difetti riesce ad identificare:

- BICCHIERE NON CIRCOLARE (eccentricità)
- BORDO SVASATURA TROPPO IRREGOLARE (regolarità)
- TAGLIO GRAVE PROSSIMO AL TUBO
- SVASATURA INSUFFICIENTE
- SVASATURA RIPIEGATA

Le matrici difetti dello schema-tubi vengono create all'avvio del software e sono tante quanti sono gli schemi-tubi presenti nel database (una per schema-tubi) mentre le matrici difetti dell'ordine vengono create solo alla prima esecuzione di un ordine (una per ogni schema-tubi contenuto nell'ordine). La vera differenza tra i due tipi di matrice sta nell'aggiornamento delle stesse ma di questo parleremo in seguito nella fase relativa all'aggiornamento.

Possiamo dividere la gestione dei difetti in tre fasi principali:

1. Memorizzazione difetti
2. Aggiornamento difetti
3. Gestione emergenze relative ai difetti

La memorizzazione difetti viene svolta all'interno dell'assemblaggio, in particolare nel momento dell'analisi delle ROI relative alla curvetta da inserire vengono salvate (in un campo chiamato "note" presente in ogni tubo) eventuali informazioni sui difetti riscontrati. Come già affermato i difetti riscontrabili sono 5 (in realtà sono 6, infatti il sistema riconosce anche tagli lievi sul bordo delle svasature, ma questo tipo di difetti non viene considerato grave e quindi non è gestito).

L'aggiornamento difetti avviene al termine dell'assemblaggio, poco prima dell'acquisizione delle immagini finali per la determinazione del corretto assemblaggio. I due tipi di matrice vengono aggiornati in modo diverso: per quanto riguarda la matrice difetti dell'ordine viene incrementato il numero relativo al tubo e al difetto riscontrato di una unità, tale incremento non ha limite superiore

infatti tali matrici servono solo a scopo statistico. L'aggiornamento della matrice difetti dello scambiatore è leggermente più complesso, ogni cella è impostata (al momento della creazione della matrice) ad un valore di partenza (nel nostro caso 20) ogni volta che si analizza un tubo si procede in questo modo: per ogni tubo si analizzano tutti e 5 i tipi di difetto: se il difetto non c'è in quel tubo allora la cella relativa viene incrementata di 1 (fino ad un massimo dettato dal valore iniziale, nel nostro caso 20), in caso contrario si diminuisce il valore di tale cella di 1. In questo modo il numero presente in ogni cella cala apprezzabilmente solamente se il difetto relativo è consecutivo (non isolato).

La gestione emergenze segue immediatamente l'acquisizione ed elaborazione delle immagini per la verifica del corretto assemblamento. Se sono stati rilevati problemi (e quindi possono essere presenti dei difetti) viene fatta l'analisi delle matrici difetti degli schemi-tubi (in particolare solo quelle relativi agli schemi utilizzati nell'ordine in lavorazione); vengono quindi analizzate tutte le celle delle matrici e se il valore presente è inferiore ad una certa soglia (nel nostro caso 12) significa che c'è stato un numero pari a *valore di partenza - valore di soglia* di difetti consecutivi di quel tipo su quel tubo (nel nostro caso  $20-12 = 8$ ) e si procede quindi a lanciare un allarme a video e a mandare in *STOP* la cella. Ricevuta la notifica dall'operatore si reimposta al valore iniziale la matrice che ha generato l'errore.

# Capitolo 4

## Il sistema di Visione

Il soggetto che tratteremo in questo capitolo è il vero cuore della cella, il sistema di visione.

Lo sviluppo è l'integrazione dei sistemi di visione nell'automazione industriale ha avuto un grosso incremento nell'ultimo decennio, soprattutto grazie alla diminuzione dei costi dei componenti e alla presenza di algoritmi di elaborazione delle immagini sempre più efficienti e competitivi. I vantaggi ed i limiti che un sistema di visione sono molteplici, e dipendono per la maggior parte dal contesto operativo in cui vengono posti in essere, ad ogni modo possiamo definire delle linee guida che descrivono i pregi ed i difetti di tali sistemi rispetto ad un essere umano:

I principali vantaggi che l'adozione di un sistema di visione artificiale presenta rispetto ad un operatore umano sono:

- **La costanza delle prestazioni.** Si pensari ad un operatore umano che lavora su un nastro trasportatore per il controllo difettosità dei pezzi che vi transitano: è impossibile pretendere un controllo al 100% su tutti i pezzi e che i criteri di valutazione rimangano costanti. Lo stesso vale nel caso di più operatori che svolgano lo stesso tipo di controllo: è impossibile che tutti abbiano esattamente gli stessi criteri di valutazione.
- **La possibilità di operare in ambienti ostili.** Negli ambienti industriali esistono molte aree dove un operatore non può lavorare in condizioni di totale sicurezza e ambienti dove le condizioni ambientali sono al limite.
- **La velocità di controllo.** Un sistema di visione è in grado di svolgere operazioni di verifica in frazioni di secondo, anche su oggetti in movimento molto veloce come su nastri trasportatori. La velocità di controllo genera due importanti opportunità fra loro correlate. É possibile ripensare il sistema di controllo qualità passando da verifiche a campione a controlli totali (100% dei prodotti) con tutti i vantaggi che ne conseguono. É inoltre possibile ridisegnare i processi e le linee di produzione, introducendo sistemi di movimentazione automatici ed evitando tutte le aree intermedie dedicate ai controlli qualità e al trasferimento dei pezzi da una fase all'altra.
- **La generazione di dati sul processo.** Un sistema di visione posto in una certa fase del processo produttivo, oltre ad assolvere al suo compito

specifico, è in grado di produrre e memorizzare dati sul processo e soprattutto sulle variazioni dei parametri, consentendo di generare statistiche che permettono di individuare eventuali segnali o trend di peggioramento del processo, aiutando così i gestori della linea di produzione ad intraprendere le azioni correttive prima che il processo esca dalla soglia di tolleranza prevista.

Allo stesso tempo le limitazioni tecnico-applicative sono ancora numerose, soprattutto se il sistema non viene concepito ed installato sulla base di specifiche ben definite e stabilite a priori. Operazioni che il cervello umano è in grado di svolgere in frazioni di secondo richiedono migliaia di linee di programma e processori molto più potenti di quelli attualmente disponibili. Tipici esempi sono l'interpretazione di forme complesse o tridimensionali: ognuno di noi è in grado di riconoscere centinaia di volti in una frazione di secondo o di trovare un cacciavite a stella in una scatola degli attrezzi molto disordinata. Il nostro occhio si adatta piuttosto bene a variazioni di luminosità, variazioni della distanza del soggetto ed è in grado di mantenere l'oggetto che interessa nel campo visivo, concentrandosi sugli aspetti importanti per la funzione che stiamo svolgendo.

Tutti questi compiti non sono così scontati per un sistema automatico. In generale i sistemi di visione presentano limitazioni quando devono operare con:

- grandi varietà di pezzi da riconoscere;
- pezzi di forme molto complesse da ispezionare su più lati;
- elevato numero di punti di vista (ad esempio per riscontrare sulla stessa parte difetti individuabili solo in controluce e difetti individuabili solo in trasparenza);
- impossibilità di portare il pezzo sotto un campo visivo predeterminato;
- valutazioni e scelte sulla base di parametri non facilmente codificabili;

Ancor più problematica può rivelarsi la situazione in cui sono presenti:

- ambienti sporchi e polverosi;
- variazioni delle condizioni di illuminazione;
- variazione delle condizioni con cui l'oggetto si presenta.

Il compito del progettista del sistema di visione è sicuramente quello di ottimizzare la cella di lavoro in modo da limitare i difetti ed esaltare i pregi derivanti da tale sistema, in particolare dovrà organizzare il layout della cella in modo da enfatizzare le parti che dovranno essere analizzate: vedremo in particolare che il layout previsto per il nostro progetto è simile a quello riportato in figura 4.1, seguito da una breve descrizione delle diverse parti (per la comprensione di tutte le scelte fatte come ad esempio il tipo di telecamera si rimanda a [12], noi illustreremo solo il sistema di illuminazione essendo essenziale per la comprensione degli algoritmi sviluppati per l'elaborazione delle immagini).



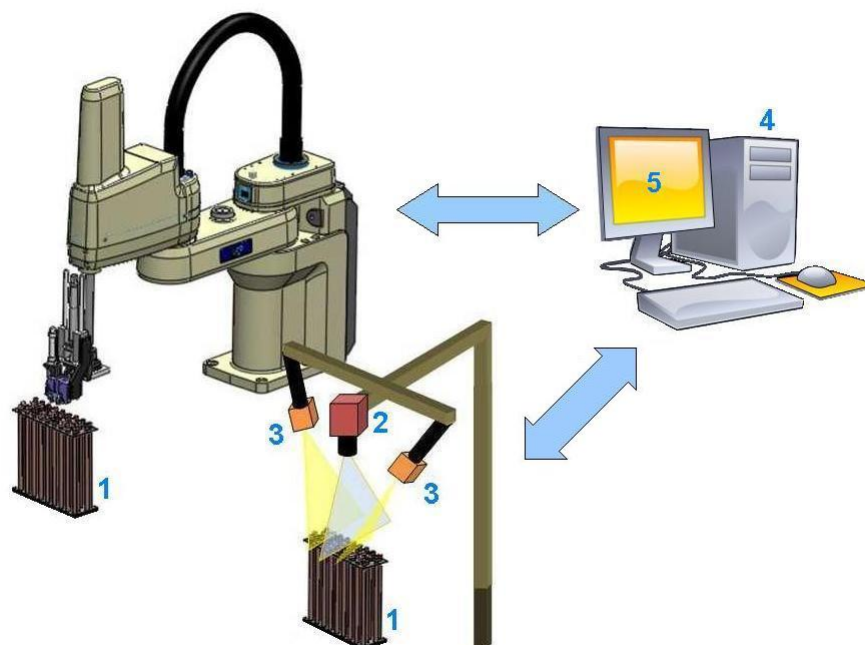


Figura 4.1: Struttura di un generico sistema di visione.

1. l'oggetto da esaminare;
2. il dispositivo di acquisizione delle immagini, composto da ottica e telecamera;
3. un sistema di illuminazione, che enfatizzi le feature dell'oggetto e ne faciliti l'identificazione;
4. un elaboratore, programmato per estrarre dalle immagini le informazioni desiderate;
5. un'interfaccia con l'utente o con il mondo esterno;

Prima di iniziare con la discussione del sistema di visione e degli algoritmi sviluppati per l'elaborazione delle immagini è opportuno (vedi [4] per ulteriori dettagli) ridare una breve illustrazione dei due diversi sistemi di riferimento (l'argomento è già stato toccato parlando della Calibrazione, al capitolo 2) che sono presenti all'interno della cella: il riferimento robot ed il riferimento visione.

L'immagine acquisita dalla telecamera è costituita da una matrice in cui ogni elemento, il pixel, costituisce la più piccola unità di informazione sulla luminosità della scena in quel punto. Il sistema di riferimento visione è pertanto dato dalle coordinate riga-colonna della matrice. È un sistema di riferimento discretizzato, basato sul numero di pixel presenti in altezza e larghezza. Il calcolo del centro avviene dunque all'interno di questo sistema di coordinate, tuttavia la precisione raggiungibile può essere maggiore della risoluzione dell'immagine e dipende dalle tecniche di elaborazione adottate. Questo è un aspetto fondamentale in quanto determina la risoluzione minima che deve avere la telecamera per poter rispettare le specifiche riguardanti la precisione nel centraggio dei tubi.

Va osservato che, mentre il sistema di riferimento robot è definito nelle tre dimensioni spaziali ( $\mathbf{x}_r$ ,  $\mathbf{y}_r$  e  $\mathbf{z}_r$ ), il sistema di riferimento della telecamera è per sua natura limitato a sole due dimensioni ( $\mathbf{x}_v$ ,  $\mathbf{y}_v$ ), che faremo coincidere con il piano su cui giacciono le imboccature dei tubi (vedi Figura 4.2).

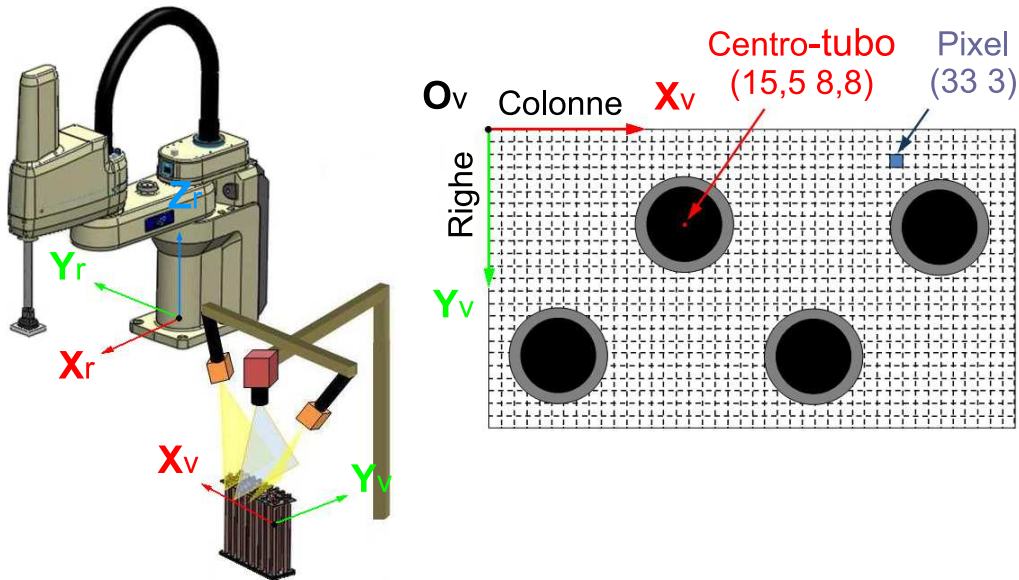


Figura 4.2: Il sistema di riferimento della Visione con coordinate dei centri-tubo.

Risulta ovvia la necessità di trovare una mappa che legghi i due sistemi di riferimento, soprattutto per diminuire i problemi dovuti alla prospettiva della scena e alle distorsioni ottiche legate alle lenti presenti nell'obiettivo. Infatti anche ottimizzando la posizione della scambiatore, posizionandolo verticalmente sotto il sistema di visione (vedi figura 4.3) in modo da consentire una valutazione dello spazio occupato dei tubi e dei difetti presenti sulle svasature, risulta evidente l'**errore prospettico** dovuto all'angolo di **parallasse** tra i vari punti dell'immagine e il centro ottico.

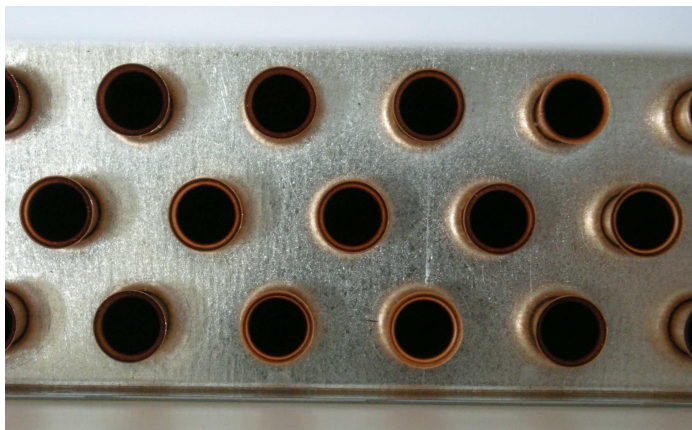


Figura 4.3: Visuale dall'alto di un scambiatore.

Tale angolo cresce all'aumentare della distanza dei punti dal centro dell'immagine, con cui è incidente l'asse ottico della camera. È evidente come l'errore di parallasse introduca una distorsione visiva progressiva allontanandosi dal centro, causando una valutazione delle forme dei tubi non corrispondenti alla realtà ed un'erronea misurazione dei loro centri. Una possibile soluzione è rappresentata dall'utilizzo di ottiche telecentriche. Purtroppo il principale svantaggio delle ottiche telecentriche sta proprio nella loro capacità discriminante sui raggi incidenti: accettando solamente raggi paralleli, l'area inquadrabile è limitata superiormente dalle dimensioni dell'obiettivo stesso quindi l'unica via percorribile per valutare correttamente i centri dei tubi, limitando l'errore introdotto dall'effetto prospettico, appare quella di spostare più in alto possibile la visuale utilizzando un teleobiettivo, riducendo così l'angolo di parallasse.

## 4.1 Il sistema di illuminazione

Risumendo quanto visto nell'introduzione del capitolo possiamo affermare che un sistema di visione artificiale dispone di meno informazioni a priori sull'oggetto da esaminare e non ha un'esperienza visiva preconstituita a cui far riferimento nell'analisi al pari di un operatore umano. Tuttavia, una volta definite con precisione le *feature* (caratteristiche) da identificare e utilizzando un sistema di illuminazione che le enfatizzi è possibile semplificare e velocizzare l'estrazione delle informazioni desiderate dalle immagini, garantendo così prestazioni superiori a quelle di un operatore umano. Si rende perciò necessario definire in modo chiaro quali informazioni vogliamo ricavare dalle immagini acquisite e per fare questo dobbiamo avere chiari gli obiettivi del sistema di visione all'interno del processo di automatizzazione in studio.

Le operazioni fondamentali della visione possono essere sintetizzate in tre punti fondamentali:

1. IDENTIFICAZIONE DELLO SCAMBIATORE: in effetti, come già visto nel capitolo 3 in merito alla "Gestione Ordini", l'informazione che a priori abbiamo sullo scambiatore che ci troveremo ad assemblare non è completa, infatti sappiamo solo che lo scambiatore sarà composto da schemi presenti nell'ordinativo, ma non sappiamo né il numero né la posizione degli stessi all'interno dello scambiatore.
2. ANALISI DELLE SVASATURE: tale fase si traduce nell'estrapolazione, dalle immagini acquisite, delle informazioni relative alle svasature, in modo da poter consentire o negare l'inserimento delle curvette. Si osserva che questa fase è volutamente svolta in tempo mascherato, quindi non viene fatta un'analisi iniziale di tutte le svasature ma mentre il robot inserisce una curvetta vengono fatte le analisi sulle svasature dei tubi appartenenti alla successiva curvetta da inserire e così via.
3. ANALISI DI EVENTUALI CURVETTE GIÀ INSERITE: tale fase ha il compito di controllare se nello scambiatore da assemblare sono già state inserite delle curvette ed in tal caso controllarne il corretto inserimento.

Non ci resta che analizzare in dettaglio ogni problema ed individuare il sistema di illuminazione che meglio lo soddisfa andando poi a trovare una soluzione generale che garantisca il soddisfacimento di tutte le richieste e quindi il corretto funzionamento della cella.

### 4.1.1 Identificazione dello Scambiatore

In questa fase le informazioni che vogliamo estrapolare dall'immagine sono costituite dal numero di tubi presenti e la loro posizione<sup>1</sup>. La svasatura, oltre ad essere un elemento variabile e potenzialmente soggetto a difetti (quindi non utilizzabile per il calcolo del centro-tubo) ci impedisce di determinare con precisione il diametro del tubo, informazione indispensabile per determinare il tipo di scambiatore in analisi (si pensi ad esempio che il modello che verrà assemblato con più frequenza è proprio formato da due schemi completamente identici per quel che riguarda il numero di tubi e il numero di ranghi) e quindi fa parte delle features da eliminare. Al contrario la parte interna dei tubi risulta essere sempre regolare e ben definita, a parte nel caso di gravi difetti al tubo per il quale non è necessaria una rilevazione precisa del centro in quanto la curva non vi verrà ovviamente inserita. Si intuisce come l'estrazione di questa feature sia legata al riconoscimento dell'area occupata dalla sola **parte interna del bicchiere**<sup>2</sup>.

L'estrazione di tale feature si ottiene filtrando la svasatura e la piastra zincata. Una tale elaborazione ottica è ottenibile facilmente combinando le proprietà di riflessione della luce dei materiali che compongono la parte superiore degli scambiatori con un'adeguata impostazione dei parametri dell'ottica (fuoco, diaframma, esposizione).

Le piastre che delimitano in altezza i pacchi alettati sono realizzate in leghe metalliche di alluminio o acciaio zincato e sono caratterizzate da un elevato indice di riflessione della luce, anche la lega di rame, che costituisce i tubi in forcina, appare lucente e riflette bene la luce.

La soluzione per separare queste due componenti è quella di posizionare in modo accurato la fonte di luce. La piastra infatti è parallela al piano immagine della camera, quindi gran parte della luce che vi incide viene riflessa nella direzione del sensore. Mentre per quanto riguarda i tubi invece, essendo posti perpendicolarmente alla piastra, la luce incidente può venire riflessa solamente dalle svasature, dal momento che quel poco di luce entrante nei tubi viene disperso al loro interno.

Queste differenze nel riflettere la luce sono state sfruttate per l'estrapolazione della feature di interesse, in particolare grazie ad una fonte luminosa posta lateralmente allo scambiatore è possibile mandare in saturazione, ovvero portare al

<sup>1</sup>La posizione è necessaria perchè ci permette in un primo luogo di definire per ogni tubo una regione che lo delimita (ROI, Region of Interest) e che verrà analizzata in tempo mascherato per determinare il vero centro del tubo e la presenza di difetti sulla svasatura, in secondo luogo perchè ci permette come vedremo in dettaglio in seguito di determinare il numero di ranghi verticali dello scambiatore)

<sup>2</sup>E' importante osservare che come centro-tubo noi intendiamo il centro presente alla sommità del tubo (senza svasatura) e non il centro presente all'altezza della piastra (il centro del foro sul pacco alettato) in modo da garantire un inserimento più sicuro (vedi [] al Cap3 per ulteriori dettagli).

livello massimo di luminosità registrabile dalla telecamera, tutto ciò che non corrisponde alla regione interna del bicchiere e che perciò non contribuisce o potrebbe falsare la misura del centro-tubo. In Figura 4.4 è riportato lo schema di disposizione ottimale delle sorgenti luminose per l'isolamento della parte interna del bicchiere.

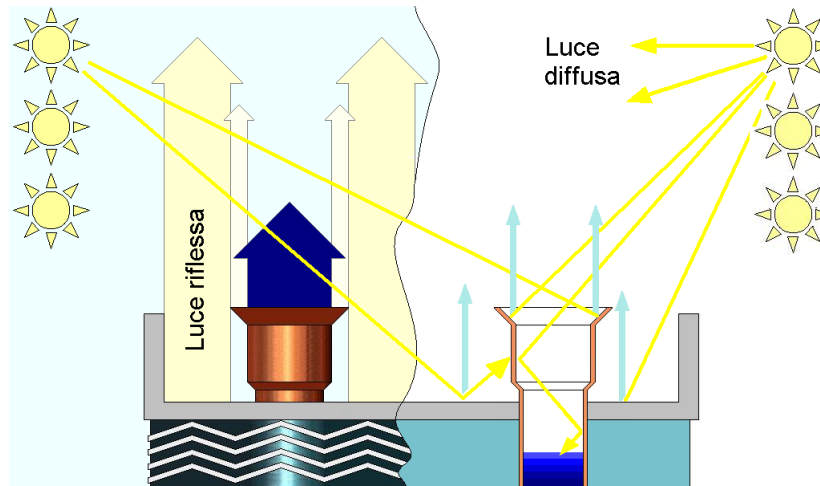


Figura 4.4: Sistema di illuminazione per l'identificazione dei centri-tubo

Giocando sul tempo di esposizione della telecamera (shutter), in particolare impostando un tempo di esposizione molto lungo, si ottiene un'immagine dove gli oggetti che anche in minima parte riflettono della luce in direzione della telecamera, come appunto la superficie della piastra zincata, le curve e le svasature dei tubi, si perdono nell'area saturata dalla luce. Al contrario l'area interna dei tubi, non riflettendo nemmeno un pò di luce a causa della sua forma e delle sue pareti perpendicolari al piano zincato, viene tradotta all'interno dell'immagine in un'area scura ben definita.

L'accurata disposizione e regolazione delle sorgenti luminose, unitamente ad un'adeguata impostazione dei parametri di acquisizione della telecamera, genera quindi l'elaborazione ottica desiderata, il cui risultato sperimentale è presentato in figura 4.5.

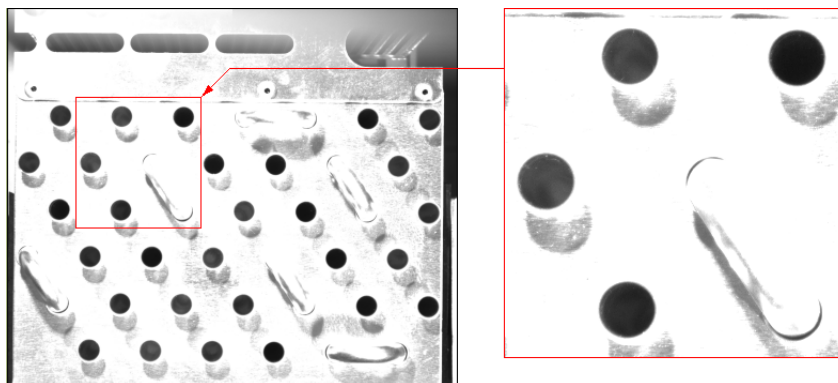


Figura 4.5: Acquisizione con illuminazione laterale e lungo tempo di esposizione.

### 4.1.2 Analisi delle Svasature

In questa fase le informazioni che vogliamo estrapolare dall'immagine sono la forma e le dimensioni delle svasature che costituiscono lo scambiatore. Come vedremo meglio alla fine del capitolo tali informazioni sono indispensabili per determinare l'eventuale difetto a cui un tubo potrebbe essere soggetto: tagli, bordi irregolari, ripiegature, ecc. Si capisce come la feature di interesse per operare un'analisi accurata sia costituita dal profilo del **bordo della svasatura** del bicchiere.

Ricordando che le immagini che abbiamo a disposizione sono in scala di grigi, l'unico modo per isolare la feature di interesse è quello di esaltare il contrasto tra il bordo della svasatura e la piastra sottostante. Si osservi a tal proposito che l'angolo di svasatura ha un valore<sup>3</sup> che si attesta attorno ai  $45^\circ$ : è noto che se si colpisce una superficie riflettente piana con un raggio di luce, più il raggio è perpendicolare alla superficie e maggiore è la quantità di luce che viene riflessa verso la sorgente (Figura 4.6).

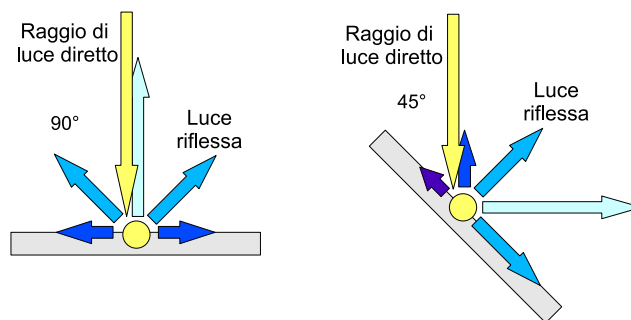


Figura 4.6: Riflessione di un raggio di luce su di un piano.

Questa differenza di inclinazione fra la superficie della piastra zincata dello scambiatore e quella delle svasature dei tubi fa sì che se le illuminiamo con una luce direzionale proveniente dall'alto, essendo la piastra perpendicolare ai raggi luminosi, questa rifletterà verso l'alto una maggiore quantità di luce rispetto alle svasature. Con questo tipo di illuminazione otteniamo però anche l'effetto indesiderato di illuminare l'interno dei tubi, che per quanto di natura poco riflettenti, possono introdurre dei disturbi nell'elaborazione dell'immagine.

Come fatto in precedenza sfruttiamo l'analisi svolta per l'estrapolazione della feature di interesse, in particolare grazie ad una fonte luminosa posta sopra allo scambiatore è possibile mandare in saturazione, ovvero portare al livello massimo di luminosità registrabile dalla telecamera, tutto ciò che non corrisponde alla totalità del tubo, ivi compresa la svasatura. In Figura 4.7 è riportato lo schema di disposizione ottimale delle sorgenti luminose per l'isolamento del contorno del bicchiere.

<sup>3</sup>Tale valore è ideale, misurazioni fatte con diversi scambiatori hanno fornito angoli compresi in media tra 30 e 50 gradi.

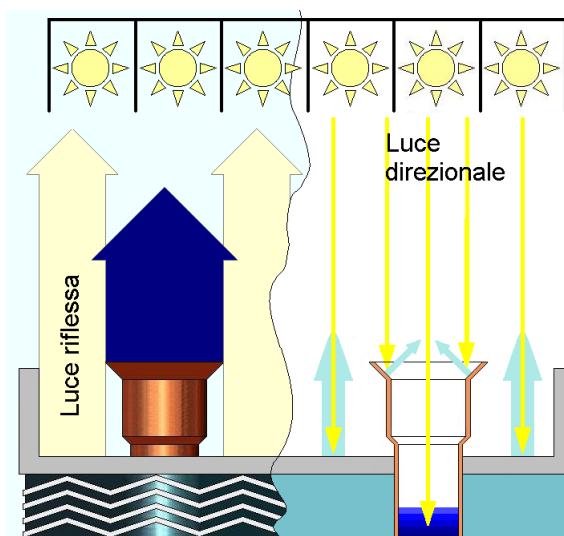


Figura 4.7: Sistema di illuminazione per l'analisi delle svasature.

Come già descritto sopra, l'utilizzo di un'illuminazione dall'alto provvede una grande quantità di luce riflessa dalla piastra zincata in direzione della telecamera, per questo il tempo di esposizione (shutter) sarà molto più breve rispetto al caso con illuminazione laterale, andando in questo modo ad ottenere un'immagine dove solo gli oggetti che riflettono gran parte della luce ricevuta in direzione della telecamera fanno parte della zona saturata dalla luce, come appunto la superficie della piastra zincata.

Al contrario l'area interna dei tubo e la svasatura, riflettendo una quantità di luce molto inferiore, vengono tradotte all'interno dell'immagine in aree scure e ben definite.

L'accurata disposizione e regolazione delle sorgenti luminose, unitamente ad un'adeguata impostazione dei parametri di acquisizione della telecamera, genera quindi l'elaborazione ottica desiderata, il cui risultato sperimentale è presentato in figura 4.8

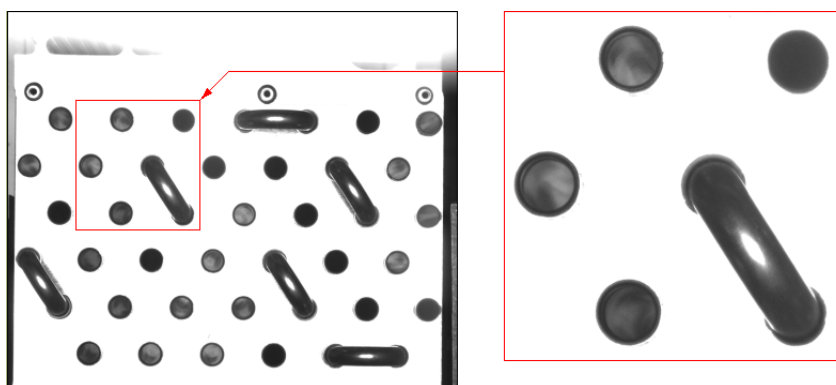


Figura 4.8: Acquisizione con illuminazione dall'alto e breve tempo di esposizione.

### 4.1.3 Analisi di eventuali Curvette già inserite

Ultimo obiettivo del sistema di visione è l'identificazione di curvette già inserite. Tale compito è fondamentale anche all'interno degli altri compiti del sistema di visione, infatti per quanto riguarda l'identificazione dello scambiatore la presenza di una curvetta implica la presenza di due tubi che nell'immagine atta a rilevare tutti i centri tubo (vedi Figura 4.5) non verrebbero altrimenti rilevati: è chiaro quindi come l'immagine atta all'analisi delle svasature (vedi Figura 4.8) verrà sfruttata per la determinazione del numero e posizione dei tubi essendo visibili in tale immagine le curvette già inserite.

Per quanto riguarda l'analisi dei difetti, la presenza di una curvetta ci invita a pensare che l'inserimento sui relativi tubi fosse possibile, quindi non si procederà per quei tubi all'analisi delle relative svasature.

Un problema ben più grave sarà la determinazione delle false curvette, a tal proposito si osservi l'immagine con illuminazione diffusa (dall'alto) dello scambiatore che maggiormente ci troveremo ad assemblare:

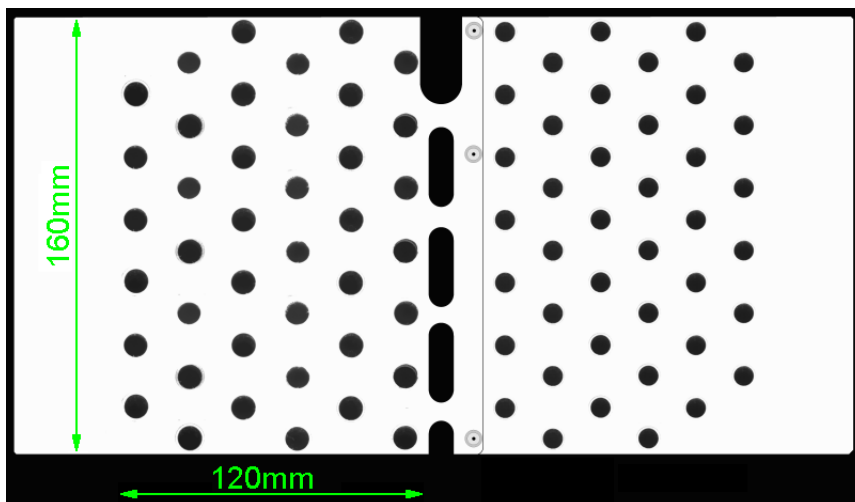


Figura 4.9: Scambiatore con illuminazione diffusa

Appaiono ben visibili nella zone centrale delle forme del tutto identiche a curvette e tubi che in realtà non sono altro che tagli fatti sulla piastra zincata. Le finte curvette sono eliminabili abbastanza facilmente, infatti tali oggetti appaiono con entrambe le illuminazioni, mentre abbiamo visto come le curvette vere appaiono solo nell'illuminazione diffusa. Basterà quindi un confronto tra le due immagini per capire quali curvette sono vere e quali no. L'eliminazione del finto tubo risulta leggermente più complessa infatti in questo caso risulta inutile il confronto tra le due immagini poichè il finto tubo compare come i tubi veri in entrambe. La strategia sarà quella di contare il numero di tubi presente su ogni rango verticale: quando la differenza del numero di tubi tra due ranghi vicini sarà superiore ad una certa soglia allora verranno eliminati tutti i tubi relativi al rango contenente il numero di tubi inferiore.



## 4.2 Strumenti per l'elaborazione immagini con MatLab®

L'operazione di elaborazione immagini è normalmente molto complessa ed onerosa in termini di calcolo e per questo si ricorre, come anche qui è stato fatto, a tecniche di illuminazione che semplifichino l'estrazione dall'immagine delle features di interesse. Prima di estrarre tali features daremo in questa sezione un breve compendio delle funzionalità del pacchetto *Image Acquisition Toolbox* di MatLab che saranno indispensabili per i nostri scopi.

La telecamera PIKE505B scelta per il progetto ritorna immagini (frames) con una profondità di 8 bit, quindi con 256 livelli di profondità. Ogni frame è rappresentato da una matrice di dimensioni pari alla risoluzione adottata e nella quale in ogni cella (pixel) è presente un numero intero compreso tra 0 (nero) e 255 (bianco) come rappresentato in Figura 4.10.

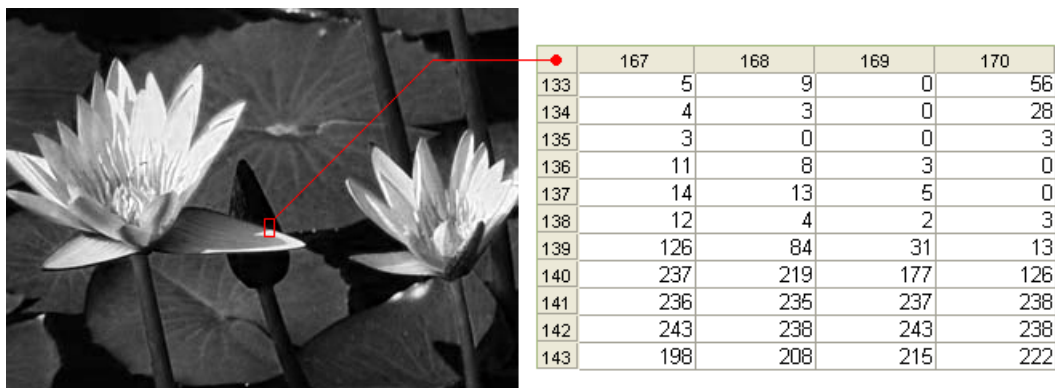


Figura 4.10: Porzione di matrice del frame di un'immagine.

L'operazione di estrazione delle features di interesse, e quindi l'elaborazione vera e propria dell'immagine viene suddivisa in 3 fasi:

1. Pre-elaborazione;
2. Segmentazione;
3. Estrazione feature.

La fase di Pre-elaborazione consiste nel migliorare i risultati ottenuti grazie all'elaborazione ottica (il sistema di illuminazione), infatti ci sono parti dell'immagine che il sistema di illuminazione non è riuscito ad esaltare o nascondere perfettamente, ecco perchè si procede con la binarizzazione dell'immagine che darà la possibilità di eliminare gli elementi di disturbo indesiderati.

Per **binarizzazione** intendiamo la conversione di ogni pixel dell'immagine da una rappresentazione a 256 livelli (scala di grigi) in una binaria a 2 livelli (bianco e nero) come riportato in Figura 4.11. La conversione avviene attraverso la funzione MATLAB *roicolor* impostando una specifica soglia di binarizzazione: tutti i pixel il cui valore di luminosità è inferiore a tale soglia vengono posti ad 1 (bianco) mentre quelli al di sopra vengono invece posti a 0 (nero) oppure viceversa a seconda delle esigenze.

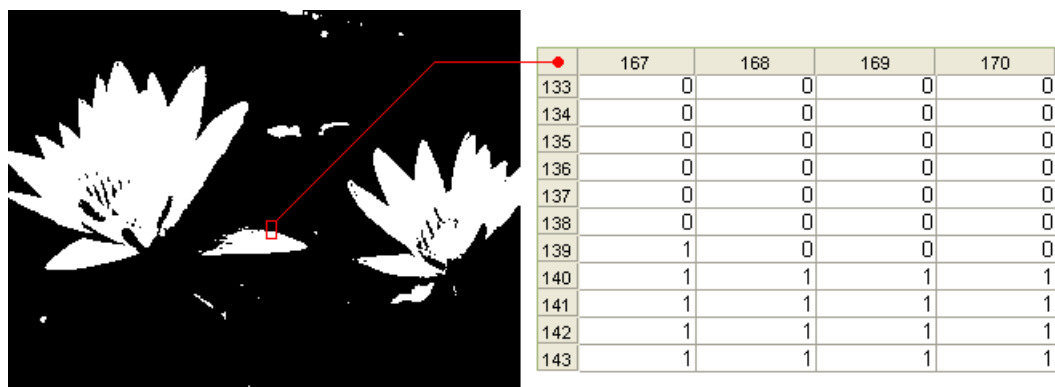


Figura 4.11: Binarizzazione dell'immagine 4.10 con un valore di soglia pari a 110.

MatLab riconosce come “oggetti” tutti gli insiemi di pixel adiacenti di valore pari a 1. Non tutti gli oggetti presenti nell'immagine corrispondono però a degli oggetti di interesse: nelle immagini degli scambiatori così come nelle Ninfee di figura 4.11 sono presenti delle “macchie bianche” dovute a limitate zone di alta riflessione che non hanno nessun legame con le features di interesse e che quindi vanno eliminate. Il pacchetto *Image Acquisition Toolbox* di MatLab fornisce due metodi principali per riuscire in tale scopo: *imopen* ed *bwareaopen*.

*Imopen*, traducibile con “apertura morfologica”, provvede alla sequenza di un'erosione e una dilatazione degli oggetti. L'erosione, attuata secondo una matrice di erosione di dimensioni e struttura impostabili, va ad erodere e quindi a cancellare i bordi di ogni regione connessa nell'immagine. In questo modo le regioni più piccole vengono perse e quelle maggiori leggermente assottigliate. Successivamente la dilatazione espande ogni pixel bianco rimasto nella regione attorno ad esso con lo stesso criterio con cui prima la stessa area era stata erosa; in questo modo le regioni sopravvissute all'erosione vengono riportate alle loro dimensioni originali.

*Bwareaopen*, traducibile con “eliminazione per dimensioni”, valuta le dimensioni di ogni regione connessa di pixel nell'immagine ed elimina gli oggetti di area inferiore ad un valore di soglia impostato.

Una altra operazione molto utile nella pre-elaborazione e quella in cui si vogliono eliminare delle “macchie nere” che inficiano la feature di interesse e che sono normalmente dovute ad un'illuminazione non ottimale o alla natura stessa dell'oggetto (si osservi come nelle Ninfee di figura 4.11 ci siano appunto delle macchie nere all'interno dell'area delimitata dai fiori). MatLab facilita questa operazione con la funzione *imfill* la quale pone ad 1 (bianco) tutti i pixel di valore 0 (nero) non raggiungibili dai bordi dell'immagine.

La pre-elaborazione fatta ci ha permesso di ottenere un'immagine a valori binari della scena nel suo complesso, in cui ogni possibile oggetto di interesse è rappresentato da una regione bianca su sfondo nero. A questo punto si può procedere con la segmentazione dell'immagine. La segmentazione è il processo che suddivide una scena nelle sue parti costituenti od oggetti. Questa operazione è necessaria per isolare il singolo pezzo da analizzare, in modo da poterne calcolare in seguito le caratteristiche desiderate. In ambiente MatLab questa operazione

viene svolta tramite la funzione *bwlabel* che, data in ingresso un'immagine binaria, produce in uscita un'immagine indicizzata. Un'immagine indicizzata altro non è che un'immagine in cui tutti i pixel appartenenti a uno stesso oggetto/regione hanno lo stesso valore, mentre pixel appartenenti a oggetti diversi hanno valore diverso. I pixel dello sfondo hanno tutti valore 0 (sfondo nero) mentre i pixel delle varie regioni hanno valore crescente (1, 2, 3, ecc) che identifica l'elemento in modo univoco. La funzione *bwlabel* analizza tutti i pixel dell'immagine e verifica, per ciascuno di essi, la presenza di valori simili nei pixel adiacenti (ossia, nel caso specifico, di pixel a valore non nullo).

Ora che sappiamo quali pixel appartengono ai singoli oggetti presenti sulla scena a seguito della segmentazione, possiamo calcolarne le caratteristiche più significative e conoscendo le caratteristiche peculiari degli oggetti si possono facilmente distinguere le label di interesse da quelle non di interesse e così procedere all'estrazione delle feature sulle label. Buona parte del lavoro in questo senso è svolta da MATLAB: data in ingresso un'immagine indicizzata, la funzione *regionprops* restituisce una vasta serie di informazioni per ogni oggetto presente sulla scena.

Tra tutte le informazioni disponibili (indicate in seguito con *descrittori regionali*) quelle che saranno più utili per i nostri scopi sono:

- *Area*, scalare; numero di pixel contenuto nella regione dell'oggetto.
- *Centroid*, vettore  $1 \times 2$ ; centro di massa (in pixel) della regione dell'oggetto.
- *Eccentricity*, scalare; eccentricità dell'ellisse avente i medesimi momenti secondi della regione dell'oggetto. L'eccentricità è il rapporto fra la distanza dei fuochi dell'ellisse e la lunghezza del suo asse maggiore. Il suo valore è compreso tra 0 e 1 (0 e 1 sono i due casi degeneri: un'ellisse con eccentricità 0 è un cerchio mentre con eccentricità 1 è un segmento).
- *EquivDiameter*, scalare; misura (in pixel) del diametro del cerchio equivalente avente la stessa area della regione dell'oggetto. Viene calcolato come  $\sqrt{\frac{4}{\pi} \cdot Area}$ .
- *MajorAxisLength*, scalare; lunghezza (in pixel) dell'asse maggiore dell'ellisse avente i medesimi momenti secondi centrali normalizzati della regione dell'oggetto.
- *MinorAxisLength*, scalare; lunghezza (in pixel) dell'asse minore dell'ellisse avente i medesimi momenti secondi centrali normalizzati della regione dell'oggetto.
- *Orientation*, scalare; l'angolo (in gradi) tra l'asse x dell'immagine e l'asse maggiore dell'ellisse avente i medesimi momenti secondi della regione dell'oggetto.

Dalle spiegazioni accanto ai diversi descrittori regionali è chiaro che stiamo parlando delle caratteristiche geometriche delle forme considerate. Altre caratteristiche possono essere derivate da quelle sopra elencate o calcolate con relativa facilità.

Spesso i descrittori regionali forniscono poco più di una descrizione sommaria della forma esaminata e sono quindi di scarso aiuto se presi singolarmente, utilizzando invece vari descrittori regionali tuttavia il loro potere discriminante aumenta.

Il grande pregio di questo tipo di descrittori è la loro semplicità: con un semplice confronto tra due scalari, operazione piuttosto leggera a livello computazionale, è possibile escludere tutti gli elementi del database che differiscono significativamente dall'oggetto da riconoscere. Nella prossima sezione in cui analizzeremo tutte diverse operazioni fatte per l'estrazione delle features di interesse mostreremo la potenza ed utilità dei descrittori regionali.

Risumendo, le operazioni di elaborazione dell'immagine sono state:

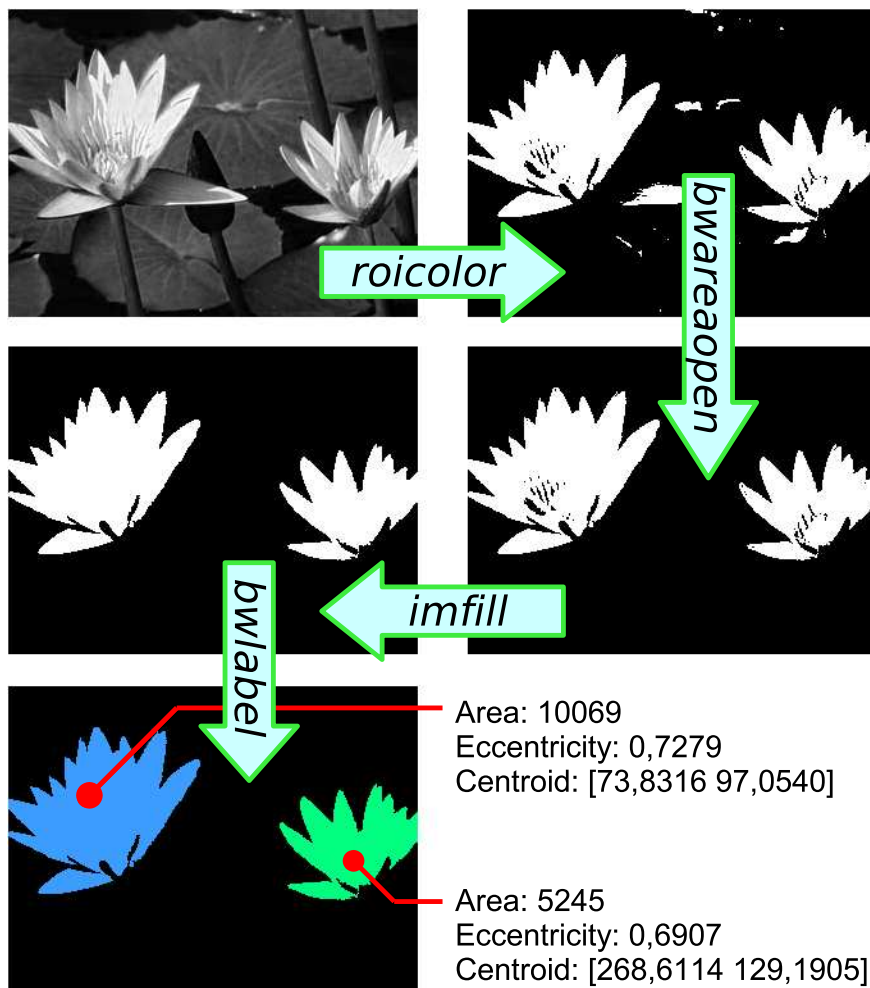


Figura 4.12: Esempio di elaborazione di un'immagine.

### 4.3 Elaborazione immagini con MatLab<sup>®</sup>

Siamo pronti ora al vero e proprio adempimento del compito del sistema di visione: l'identificazione dello scambiatore e l'analisi dei difetti (la determinazione di curvette già presenti verrà fatta all'interno della parte di identificazione).

### 4.3.1 Elaborazione per l'identificazione dello Scambiatore

L'elaborazione verrà fatta sfruttando diversi descrittori di forma combinati insieme, è per questo importante capire prima di procedere quali siano le caratteristiche che ci permettono di distinguere uno scambiatore da un'altro. Gli scambiatori sono in generale insiemi di schemi-tubi (il nostro campione principale è formato da due schemi-tubi) che si differenziano per tre caratteristiche principali:

- numero di ranghi
- numero di tubi
- diametro dei tubi

Non tutte queste caratteristiche devono cambiare da uno schema-tubi all'altro, si osservi come il nostro campione sia formato da due schemi-tubi che sono identici per quanto riguarda i primi due punti e differiscano solo per il diametro dei tubi.

La determinazione di queste tre caratteristiche ci obbliga a trovare il sistema di isolare all'interno delle immagini disponibili (si ricorda che si hanno due immagini relative a due sistemi di illuminazione diversi: illuminazione diffusa, in cui vengono evidenziate le curvette già presenti e il contorno delle svasature; illuminazione laterale, in cui scompaiono le curvette già presenti ed è invece ben visibile la parte interna dei bicchieri) le due entità fondamentali che costituiscono lo scambiatore: il tubo e la curvetta.

Come vedremo tra breve, illustrando sequenzialmente la procedura di identificazione dello scambiatore, l'isolamento dei tubi avviene grazie all'immagine con illuminazione laterale dalla quale (una volta pre-elaborata e segmentata) verranno considerati tubi tutti quegli oggetti che hanno un livello di eccentricità molto basso e un'area compresa tra l'area massima e minima dei tubi presenti nel database a nostra disposizione (considerando un certo margine dovuto alle svasature e ad eventuali irregolarità).

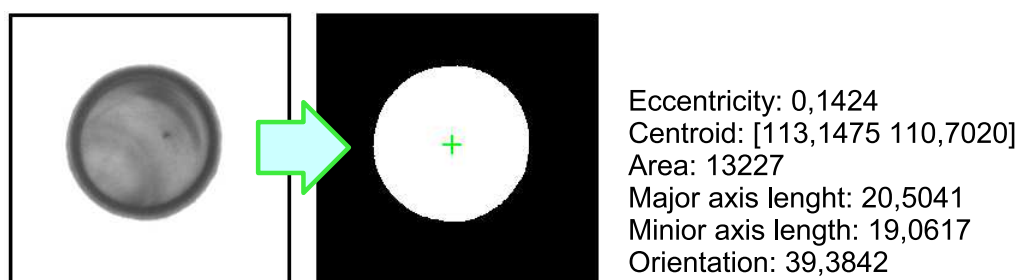


Figura 4.13: Identificazione tubo.

L'isolamento delle curvette già inserite si ottiene dall'immagine con illuminazione diffusa dalla quale (una volta pre-elaborata e segmentata) verranno considerate curvette tutti quegli oggetti che hanno i descrittori regionali di area, eccentricità, asse maggiore ed asse minore con valori che, con un certo margine, rientrano in un ellissoide che può essere associato ad una curva. Stabilita la natura dell'oggetto, la stima della posizione dei due relativi centri-tubo sottostanti

risulta immediata, infatti chiamati A e B i centri dei tubi corrispondenti ad curva già inserita (Figura 4.14) possiamo applicare il seguente calcolo<sup>4</sup> basato sull'utilizzo della posizione del centroide della curva  $(x_c, y_c)$ , della lunghezza degli assi maggiore e minore  $M$  e  $m$  e loro orientazione  $\beta$ :

$$A : \begin{cases} x_A = x_c - (M/2 - m/2) \cos(-\beta) \\ y_A = y_c - (M/2 - m/2) \sin(-\beta) \end{cases}$$

$$B : \begin{cases} x_B = x_c + (M/2 - m/2) \cos(-\beta) \\ y_B = y_c + (M/2 - m/2) \sin(-\beta) \end{cases}$$

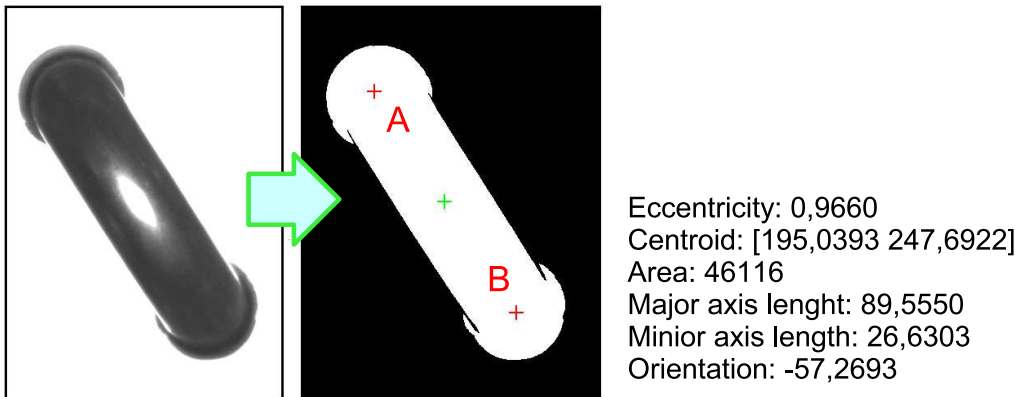


Figura 4.14: Identificazione curva.

<sup>4</sup>Si osservi che non ci interessa la posizione esatta del centro, infatti non dovremo andare ad inserire nessuna curvetta in tale locazione. La posizione dei due tubi associati alla curvetta ci serve solo per capire dove sono posti rispetto a tutti gli altri tubi, per capire quindi a che rango fanno parte.

Possiamo quindi iniziare la vera propria identificazione dello scambiatore, utilizzeremo l'immagine ottenuta con illuminazione laterale per determinare il numero e la posizione dei tubi visibili e poi utilizzeremo l'immagine con illuminazione diffusa per determinare il numero e la posizione di eventuali tubi su cui la curvetta è già stata inserita. Con un mix tra le due immagini riusciremo poi a calcolare con precisione l'orientazione dello scambiatore che ci permetterà di individuare da quanti schemi-tubi è composto lo scambiatore in analisi.

Cominciamo quindi con la determinazione del numero/posizione dei tubi visibili all'interno dell'immagine con illuminazione laterale: dopo aver pre-elaborato e segmentato l'immagine andiamo a considerare tubi quelle regioni che hanno area ed eccentricità comprese tra valori fissati a priori. Il codice MatLab relativo è riportato in seguito, insieme alle immagini ottenute dalla pre-elaborazione/segmentazione.

```
% Sogliatura e segmentazione immagine con illuminazione laterale
img2 = roicolor(img2, 0, SogliaBW1); % Binarizzo l'immagine
img2 = bwareaopen(img2, round(AreaTuboMin*0.7), 4); % Elimina le label bianche piccole
[Ilab2, Nlab2] = bwlabeln(img2,4); % Segmento l'immagine

% calcolo proprietà oggetti
LabProp2 = regionprops(Ilab2, 'Eccentricity', 'Centroid', 'Area');

% eliminazione oggetti con area troppo grande
LabProp2 = LabProp2 ([LabProp2.Area] < AreaTuboMax*1.35);

% eliminazione oggetti con eccentricità troppo alta
LabProp2 = LabProp2 ([LabProp2.Eccentricity] < 0.5);

% centri dei tubi senza le curvette
Centri = [LabProp2.Centroid]';
```

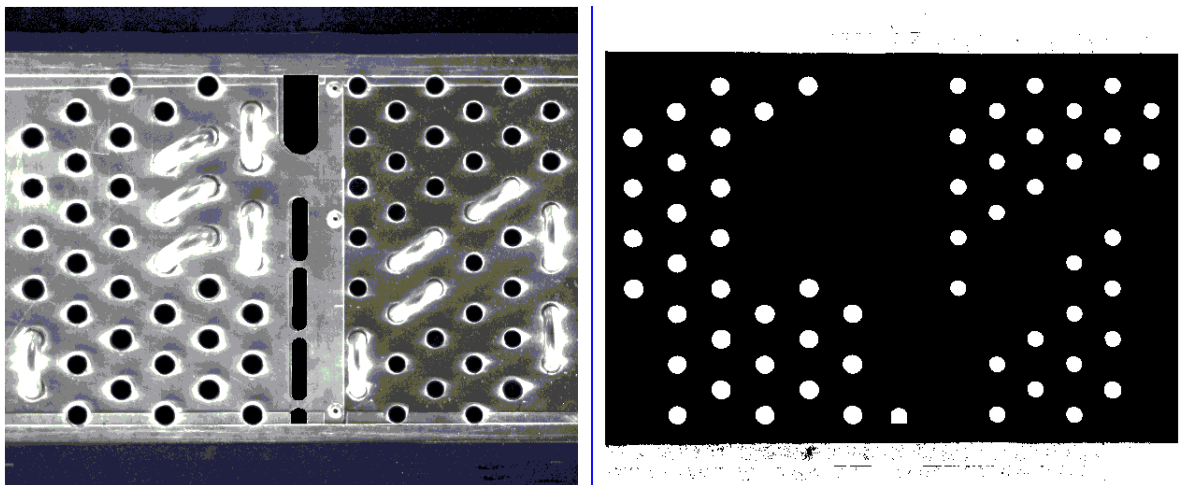


Figura 4.15: Codice per l'estrazione dei tubi visibili con immagine ad illuminazione laterale: immagine pre-elaborata (sinistra) ed elaborata (destra).

A questo punto dobbiamo calcolare la posizione dei tubi nei quali la curvetta è già stata inserita: dopo aver pre-elaborato e segmentato l'immagine con illuminazione diffusa andiamo a fare la differenza della stessa con l'immagine ad illuminazione laterale (sfruttando il fatto che le curvette si vedono solo nell'immagine ad illuminazione laterale mentre i tubi si vedono in entrambe) ottenendo un'immagine con le curvette e le svasature (vedi Figura 4.17 in basso-sinistra) ed andiamo ad isolare le sole curvette. A questo punto è immediato calcolare la posizione dei tubi con la procedura illustrata sopra. Il codice Matlab relativo e le immagini ottenute sono riportate in seguito.

```

% Sogliatura e segmentazione immagine con illuminazione diffusa
img1 = roicolor(img1, 0, SogliaBW2);           % Binarizzo l'immagine
img1 = bwareaopen(img1, round(AreaTuboMin*0.7), 4); % Elimina le label bianche piccole

% immagine differenza: rimangono solo le curvette e le svasature
img12 = img1-img2;
% isolamento delle curvette
img12 = bwareaopen(img12, 4*AreaTuboMin, 4);
img12 = imfill(img12, 'holes');               % Riempio eventuali buchi
img12 = bwlabeln(img12,8);

% calcolo proprietà oggetti
LabProp12 = regionprops(img12, 'Eccentricity', 'Centroid', 'Area', 'MajorAxisLength', ...
    'MinorAxisLength', 'Orientation');

% eliminazione oggetti con eccentricità troppo bassa
LabProp12 = LabProp12 ([LabProp12.Eccentricity]>0.6);

% eliminazione oggetti con area troppo grande
LabProp12 = LabProp12 ([LabProp12.Area]<AreaTuboMax*9);
Ncurve = size(LabProp12,1);

% Ricavo le caratteristiche dei tubi
for i = 1:Ncurve
    % Ricaviamo le coordinate dei due tubi associati a questa curva
    XY = LabProp12(i).Centroid;
    M = LabProp12(i).MajorAxisLength;
    m = LabProp12(i).MinorAxisLength;
    a = LabProp12(i).Orientation;
    CentroA = [XY(1)-(M/2-m/2)*cosd(-a), XY(2)-(M/2-m/2)*sind(-a)];
    CentroB = [XY(1)+(M/2-m/2)*cosd(-a), XY(2)+(M/2-m/2)*sind(-a)];
end

```

Figura 4.16: Codice per l'estrazione dei tubi con curvette già inserite.



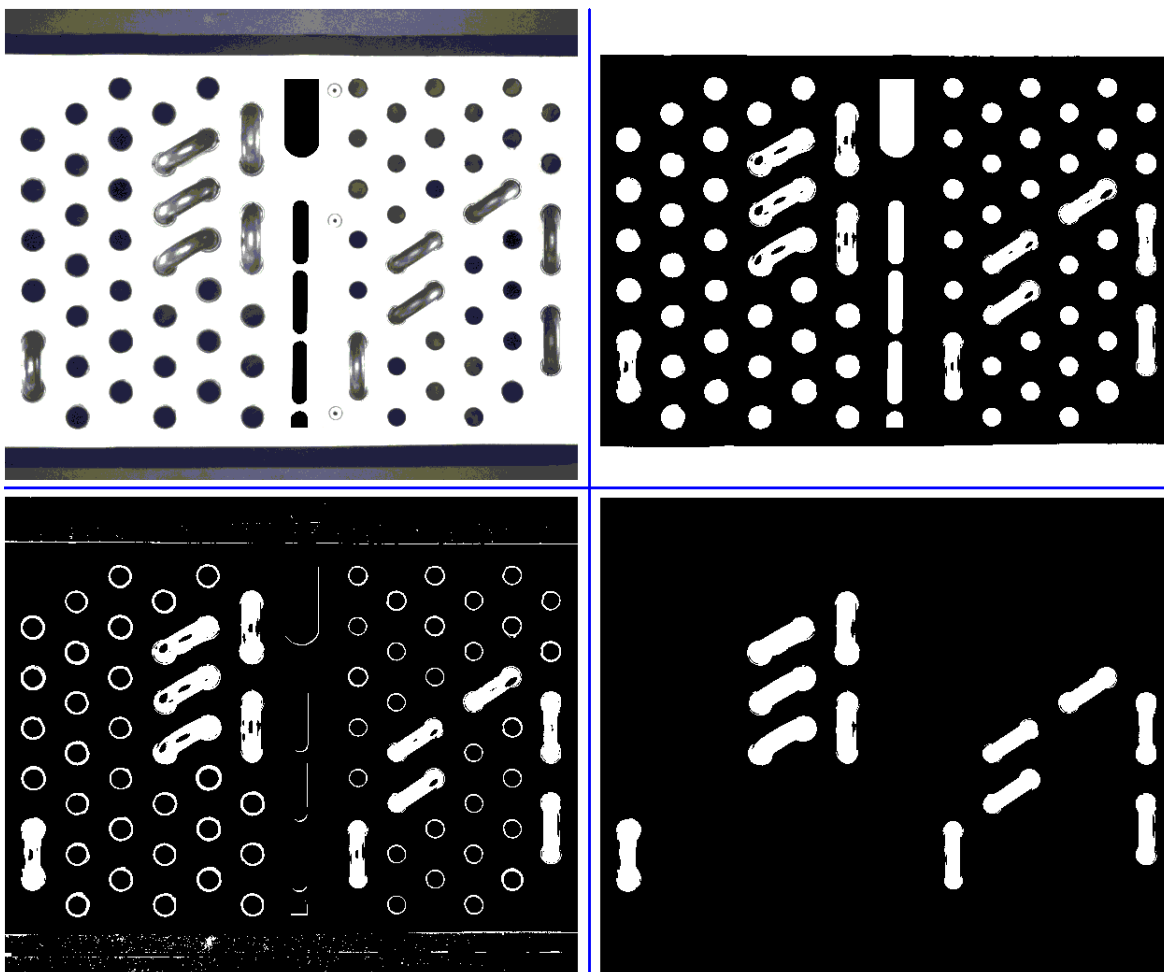


Figura 4.17: Immagini per l'estrazione dei tubi con curvette già inserite: immagine acquisita (alto-sinistra), elaborazione immagine (alto-destra), differenza con immagine ad illuminazione laterale per isolamento curvette (basso-sinistra), elaborazione immagine (basso-destra).

Come già introdotto all'inizio del capitolo dobbiamo eliminare eventuali falsi tubi che sono presenti nello scambiatore (le finte curvette sono già state eliminate nell'operazione precedente grazie alla differenza tra le due immagini (Figura 4.17, in basso-sinistra). La strategia che svilupperemo si basa sui ranghi dello scambiatore: la prima fase consiste nel costruire un frame orientato esattamente come lo scambiatore (Figura 4.18) dove non importa l'origine di tale frame poiché andremo a fare solo differenze tra diversi punti.

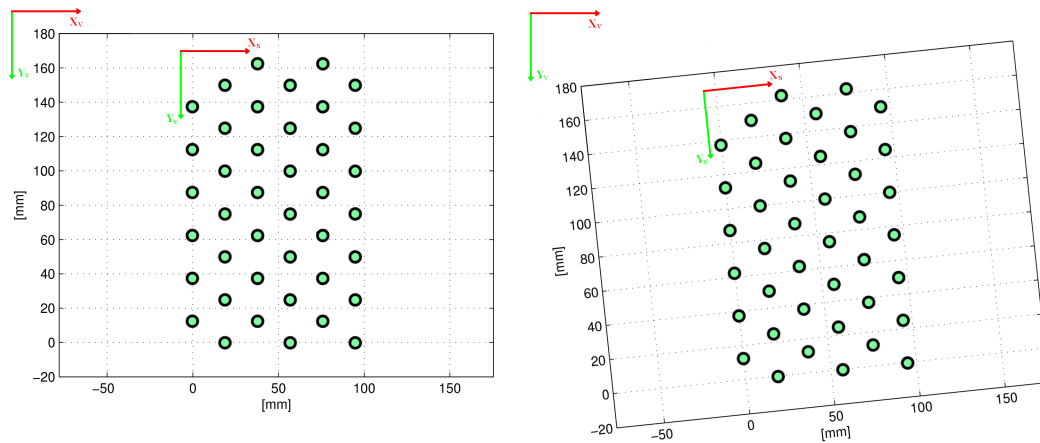


Figura 4.18: Costruzione di un frame orientato come lo scambiatore

La costruzione di tale frame è relativamente semplice, basta prendere l'immagine binarizzata ottenuta con l'illuminazione diffusa (dove si vede meglio il contorno dello scambiatore), rovesciare la binarizzazione (trasformare il bianco in nero e viceversa) e togliere tutte le regioni nere racchiuse nello spazio bianco (Figura 4.19).

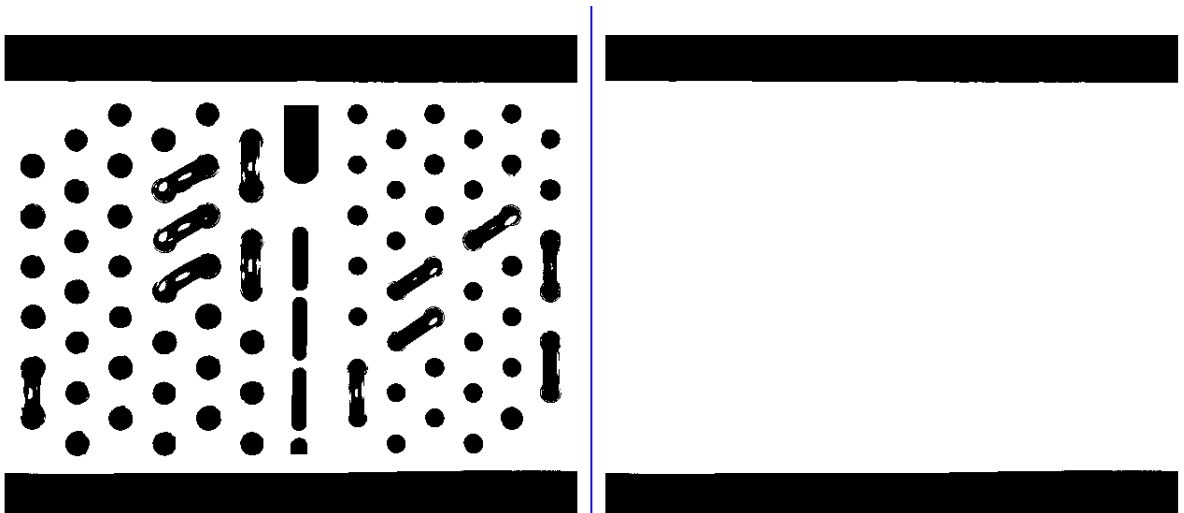


Figura 4.19: Immagini per la determinazione dell'orientazione dello scambiatore: inversione dei colori (sinistra), elaborazione immagine (destra).

Nota la corretta orientazione dello scambiatore andiamo ad ordinare in ordine crescente lungo l'asse  $x_s$  tutti i centri tubo fino ad ora rilevati (con e senza curvette). Per determinare il numero di ranghi verticali è sufficiente prendere il vettore ordinato ottenuto e fare la differenza tra un elemento ed il successivo: ogni qual volta la differenza è maggiore del diametro massimo del tubo (compresa una certa tolleranza) significa che stiamo cambiando rango. Tale procedura porta nel nostro caso al risultato rappresentato in figura 4.20, dove sono indicati i numeri di ranghi rilevati.

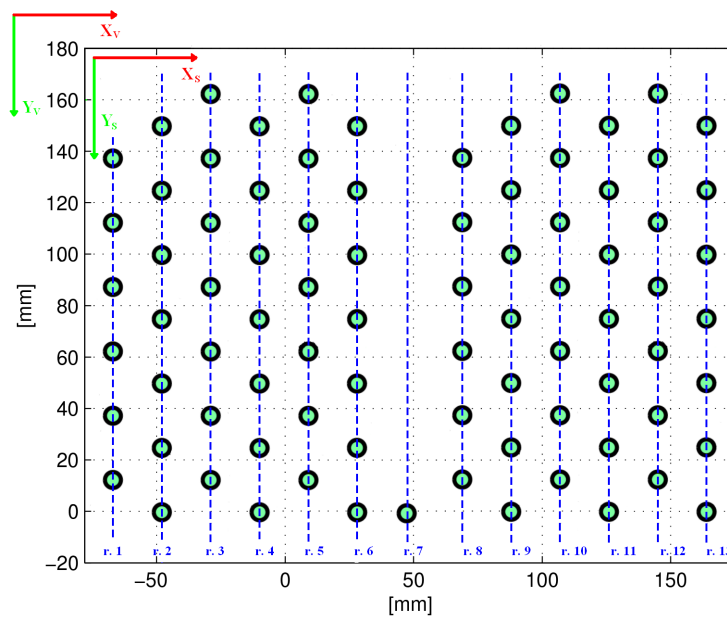


Figura 4.20: Identificazione del numero di ranghi dello scambiatore

A questo punto risulta immediato eliminare i falsi tubi, infatti basta confrontare il numero dei tubi di ogni rango con il numero di tubi del rango successivo e se la differenza è superiore ad una soglia stabilita (nel nostro caso basta 1) allora tutti i centri tubi appartenenti al rango di dimensione minore vengono eliminati dall'insieme dei centri.

```
function [indice, indfr, tubipr, dx]=ranghiCentri(xc,D)
% xc = coordinate x o y dei centri tubo
% D = distanza discriminante tra i ranghi

% ordina le coordinate dei centri (tubi di uno stesso rango hanno coordinata simile)
[XX, indice]=sort(xc,'descend');
% calcola le differenze tra le coordinate di caselle adiacenti nel vettore
dx=XX(2:end)-XX(1:end-1);
% identifica il numero dell'ultimo tubo di ciascun rango verticale
indfr = [0;find(abs(dx)>D/2);length(XX)];
% calcola il numero di tubi per ciascun rango verticale (non necessario)
tubipr = indfr(2:end)-indfr(1:end-1);
```

Figura 4.21: Codice MatLab per l'eliminazione dei falsi tubi

Questa operazione va chiaramente ripetuta anche per i ranghi orizzontali: si procede quindi con l'ordinare i centri rimasti in ordine crescente lungo l'asse  $y_s$  e ad identificare il numero di ranghi orizzontali, completata tale operazione si conclude con il confronto tra ranghi esattamente come visto per i ranghi verticali.

A questo punto non ci resta che capire dall'insieme di centri rimasto quali appartengano ad uno schema e quali ad un altro. La soluzione è identica alla determinazione dei ranghi, solo che in questo caso per determinare quando finisce un'insieme e ne inizia un altro non usiamo il diametro massimo del tubo ma utilizziamo l'interasse massimo possibile tra i ranghi. Il risultato è riportato in figura , il codice MatLab è quello riportato in figura 4.21 con la sola variazione del parametro di ingresso D.

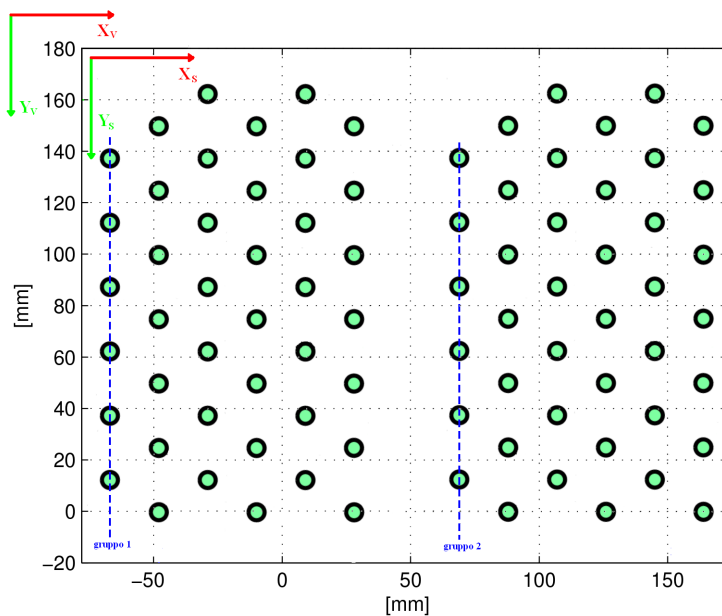


Figura 4.22: Identificazione del numero di schema-tubi

Ora non ci resta che analizzare il diametro dei tubi appartenenti ai diversi schema-tubi per avere tutte le informazioni necessarie all'identificazione. L'analisi del diametro prevede di analizzare al massimo tre tubi per ogni schema-tubi: un tubo è sicuramente senza curvetta (ogni scambiatore è dotato di un tubo destinato ad essere l'uscita del circuito di raffreddamento), poi si analizza il diametro di un secondo tubo (se presente): se il diametro è identico al primo oppure il tubo non esiste allora l'operazione termina assegnando come diametro tubi il diametro rilevato sul primo tubo analizzato altrimenti si procede con l'analisi di un terzo tubo (se esiste): se il diametro è identico ad uno dei diametri fin'ora rilevati allora si esce dalla funzione assegnando come diametro tubi il diametro rilevato sul terzo tubo analizzato, altrimenti si esce dall'operazione di assegnazione del diametro senza dato, realizzando una non identificazione dello schema-tubi.

### 4.3.2 Elaborazione per l'identificazione dei difetti

L'operazione di riconoscimento della presenza di eventuali difetti è funzionale al corretto e soprattutto sicuro inserimento delle curve da parte del robot. Dalle stesse considerazioni appena fatte nel paragrafo precedente, anche per la procedura di analisi e riconoscimento dei difetti conviene concentrarsi sull'elaborazione della singola ROI di ogni tubo anzichè dover elaborare in un sol colpo tutta l'immagine. Per questo motivo risulta opportuno eseguire questa operazione in tempo mascherato assieme alla misurazione accurata della posizione dei centri-tubo

Diversamente dall'identificazione dei centri-tubo, l'analisi dei difetti non si ferma all'estrazione di alcune feature d'interesse ma devono essere successivamente rielaborate per poter valutare la presenza di eventuali anomalie, per la maggior parte riguardanti la regione di svasatura del bicchiere. Per riconoscere e classificare i possibili difetti ognuno di questi va quindi fatto corrispondere ad una determinata fenomenologia delle feature e ad appositi indicatori su di esse. In tal modo la presenza di un difetto può essere associata al valore di questi indicatori. Si pone dunque il problema di associare ad ogni difetto una feature (o un insieme di feature) che lo identifichi adeguatamente differenziandolo dagli altri. Per predisporre adeguatamente il sistema di visione al riconoscimento dei difetti è necessario perciò definire e classificare in via preliminare le varie tipologie di difetti che il sistema dovrà essere in grado di riconoscere e quantificare.

Tratteremo separatamente le diverse tipologie di difetti da riconoscere. Di ogni tipologia verrà data una dettagliata descrizione del difetto, le cause che l'hanno generato ed il suo livello di criticità sulla successiva operazione di inserimento della curva nel tubo stesso. Da questa descrizione determineremo le feature e la relativa elaborazione attraverso cui poter rilevare tale difetto.

Tutti i seguenti algoritmi di riconoscimento dei difetti sono implementati all'interno delle funzioni *analisiBiccROI* e *analisiSvasROI*. La suddivisione di questo compito di riconoscimento dei difetti all'interno di queste due funzioni distinte dipende dal fatto che si è ritenuto opportuno dedicare, per ogni tubo da analizzare, la prima funzione all'elaborazione della sola ROI contenente l'immagine in risalto della regione interna del bicchiere ed in successione la seconda funzione all'elaborazione dell'altra ROI dedicata alla regione interna della svasatura.

#### 4.3.2.1 Taglio sulla svasatura

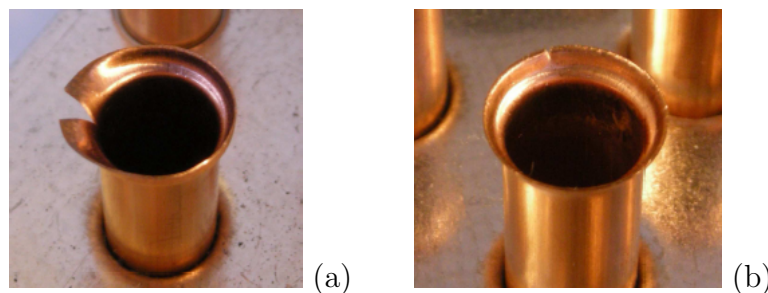


Figura 4.23: Esempi di taglio sulla svasatura: grave (a) e lieve (b).

Questo difetto è caratterizzato dalla presenza di una fenditura nella svasatura in direzione del centro-tubo. Possiamo suddividere questo difetto in due tipologie: grave, se la fenditura arriva in prossimità del tubo, altrimenti lieve.

La causa può essere imputata principalmente a due fattori:

1. Stress eccessivo del rame in fase di mandrinatura. La sporgenza eccessiva della forcina impone alla seconda ogiva una corsa più lunga, svasando eccessivamente il tubo e portando le tensioni all'interno del metallo al punto di rottura;
2. Stress del rame causato da una mandrinatura non uniforme dell'imboccatura; la barra mandrinante si flette nel tubo svasando maggiormente un lato dell'imboccatura portandolo al punto di rottura;

La criticità di questo difetto dipende dall'entità del taglio. Nel caso di taglio grave la criticità è **alta**: la presenza di un taglio prossimo al tubo può inficiare la brasatura e causare una perdita nel circuito. Nel caso invece di taglio lieve la criticità è **bassa**: le dimensioni della svasatura sono complessivamente sufficienti, il difetto non costituisce un problema per le successive fasi di lavorazione e si può perciò procedere con l'operazione di inserimento della curva.

Come possiamo subito notare dalla tipica ROI di un tubo con un taglio profondo sulla svasatura di figura 4.24, la feature utile al suo riconoscimento è chiaramente data dal contorno della svasatura che, in presenza di un taglio, si piega improvvisamente verso l'interno del tubo ed è ricavabile utilizzando la funzione *bwperim* di MATLAB.

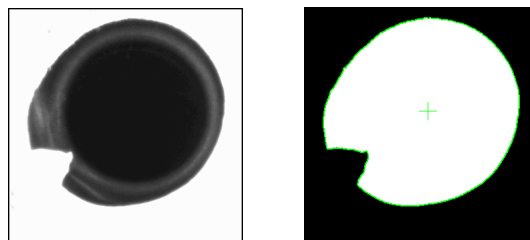


Figura 4.24: ROI di un tubo con taglio sulla svasatura.

Il primo algoritmo sviluppato per l'identificazione di questo difetto è frutto del precedente lavoro di tesi di Jacopo Vincenzi [2]. Questo algoritmo analizza il contorno percorrendolo in senso antiorario, ordinando i suoi punti secondo tale percorso ricorrendo alla funzione MATLAB *bwtraceboundary* impostando come punto iniziale il punto ad ore 3 rispetto al centro-tubo precedentemente calcolato. Per ogni pixel del contorno (percorso in senso antiorario) l'algoritmo calcola nel vettore *PtDist* le distanze dei pixel del bordo della svasatura dal centro del tubo. Dai grafici riportati in figura 4.25 si evidenzia subito il settore del profilo occupato dal taglio, che genera una rientranza a V sulla svasatura. L'obiettivo risulta dunque l'identificazione automatica del taglio e la sua localizzazione sul profilo e quindi sull'immagine originale.

L'indicatore fondamentale che consente all'elaboratore di distinguere il settore di taglio dal resto del profilo è dato dalla monotonicità crescente e decrescente

dell'andamento del profilo al suo interno. Questa caratteristica può essere chiaramente sfruttata per evidenziare e localizzare il difetto. Una derivazione sul profilo di svasatura rivela come il settore interno al taglio presenti un valore elevato di derivata con segno pressochè costante. Sommando allora per ogni punto del profilo i valori di derivata in un suo intorno, i punti appartenenti al settore di taglio risultano enfatizzati rispetto agli altri. L'ampiezza dell'intorno di somma è definita in funzione della variabilità del profilo di svasatura: maggiore è il taglio maggiore è la deviazione standard del profilo.

La stima del punto di taglio avviene attraverso un'operazione di convoluzione tra il vettore delle somme di derivata con il vettore di derivata di una funzione gaussiana di ampiezza proporzionale alla deviazione standard del profilo di svasatura e di varianza fissa. Per fare questo **PtDist** viene derivato con la funzione *diff*, ma solo dopo essere stato leggermente smussato con la funzione *smooth* al fine di avere una derivata più continua. La funzione *smooth* calcola il valore di ogni punto  $i$  del nuovo vettore smussato come media dei valori del vettore di partenza contenuti in un intorno  $[S - i : S + i]$  di ampiezza  $2S + 1$  impostabile. Prima di applicare la funzione *smooth*, per smussare in modo uniforme anche la parte iniziale e finale del segnale, il vettore **PtDist** deve essere reso circolare (cioè in capo al vettore viene aggiunto un pezzo di coda ed in coda viene aggiunto un pezzo di testa). Successivamente l'algoritmo calcola il vettore delle somme **SumDer** dei valori di derivata in un intorno di ogni punto  $i$  del vettore **DerPtDist**.

A questo punto l'algoritmo può calcolare il vettore **Conv** dato dalla convoluzione tra il vettore **SumDer** e la curva gaussiana creata. La scelta di utilizzare la derivata di una gaussiana come filtro di convoluzione nasce dalla necessità di realizzare un'operazione di derivata sul vettore in input, senza però esaltarne le alte frequenze. Dopo aver amplificato opportunamente **Conv** dove la svasatura è più sottile, questo algoritmo rileva la presenza di un taglio nel punto di massimo del vettore **Conv** se in questo punto il suo valore assoluto è superiore al valore di soglia  $T * \text{mean}(\text{abs}(\text{Conv}))$ , dove  $T$  è un parametro opportunamente settato in via sperimentale ad un valore pari a 6.

Applicando questo algoritmo alla ROI di figura 4.24 sono stati ottenuti il grafico di figura 4.25 nel quale possiamo constatare la presenza di un picco ben definito dei valori del vettore **abs(Conv)** in corrispondenza del punto di taglio.

Una volta individuato un punto di taglio, per stimarne la criticità (taglio grave o lieve) è sufficiente confrontarne la distanza del punto dal centro del tubo con il raggio del tubo stesso precedentemente misurato: se questa distanza risulta inferiore ad una certa soglia di sicurezza il taglio viene ritenuto grave. Nel caso in cui la svasatura del tubo presenti più di un taglio questo algoritmo si limita ad identificare solamente quello più profondo in quanto l'eventuale presenza di altri tagli più piccoli non modifica la criticità del difetto.

Ritenendo il precedente algoritmo un pò troppo oneroso dal punto di vista computazionale, si è ritenuto opportuno sviluppare un nuovo algoritmo di rilevazione tagli più semplice. Questo nuovo algoritmo, come il precedente, richiede per prima cosa la determinazione del vettore **PtDist** contenente le distanze dei punti del bordo della svasatura dal centro del tubo. Come già descritto, questo vettore viene prima reso circolare e poi smussato nel vettore **PtDistSmooth**. Riprendendo

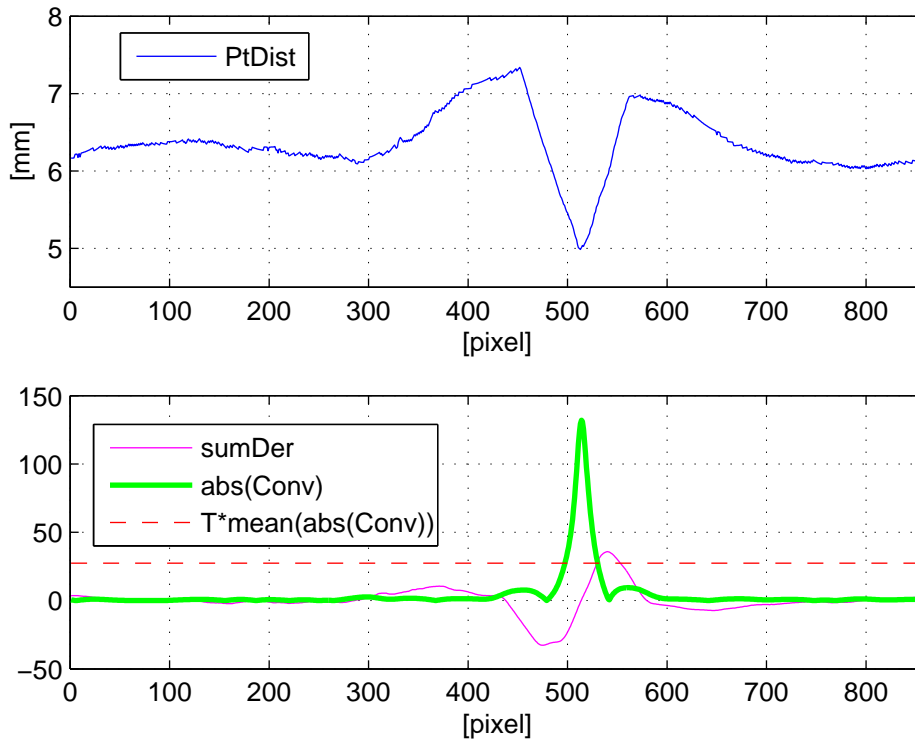


Figura 4.25: Algoritmo di riconoscimento taglio (metodo Vincenzi).

l'analisi della ROI di figura 4.24, smussando con un intervallo di ampiezza piuttosto elevato ( $2S + 1 = 51$ ) si nota che l'andamento dei valori di  $PtDistSmooth$  segue abbastanza fedelmente il profilo della svasatura contenuto nel vettore  $PtDist$ , ma in presenza di brusche variazioni delle distanze dei punti del bordo della svasatura dal centro del tubo, ad esempio a causa della presenza di un taglio, il loro andamento si discosta notevolmente (Figura 4.26 sopra).

Andando a calcolare il vettore differenza  $dPtDist = PtDistSmooth - PtDist$  possiamo osservare la presenza di un unico picco positivo (i tagli sono per definizione sempre rivolti verso l'interno del tubo) ben definito in corrispondenza del punto di taglio.

In generale questo algoritmo rileva la presenza di un taglio nel punto di massimo del vettore  $dPtDist$  se in questo punto il suo valore è superiore al valore di soglia  $T$ , il cui valore è stato determinato in modo empirico pari ad  $1/50$  del diametro del bicchiere misurato. Anche in questo algoritmo la stima della criticità del taglio individuato avviene confrontando la distanza del profilo nel punto di taglio dal centro del tubo con il raggio del tubo misurato. Abbiamo ritenuto opportuno fissare la soglia di taglio grave pari a  $0.3mm$  oltre il bordo esterno del bicchiere.

Nonostante la sua semplicità questo algoritmo risulta altrettanto robusto del precedente ma con tempi di calcolo nettamente inferiori: ad esempio utilizzando la versione R14 di MATLAB su di un processore AMD Sempron 2500+ a 1.75GHz il tempo di calcolo richiesto dal precedente algoritmo è di circa 0.16s contro i



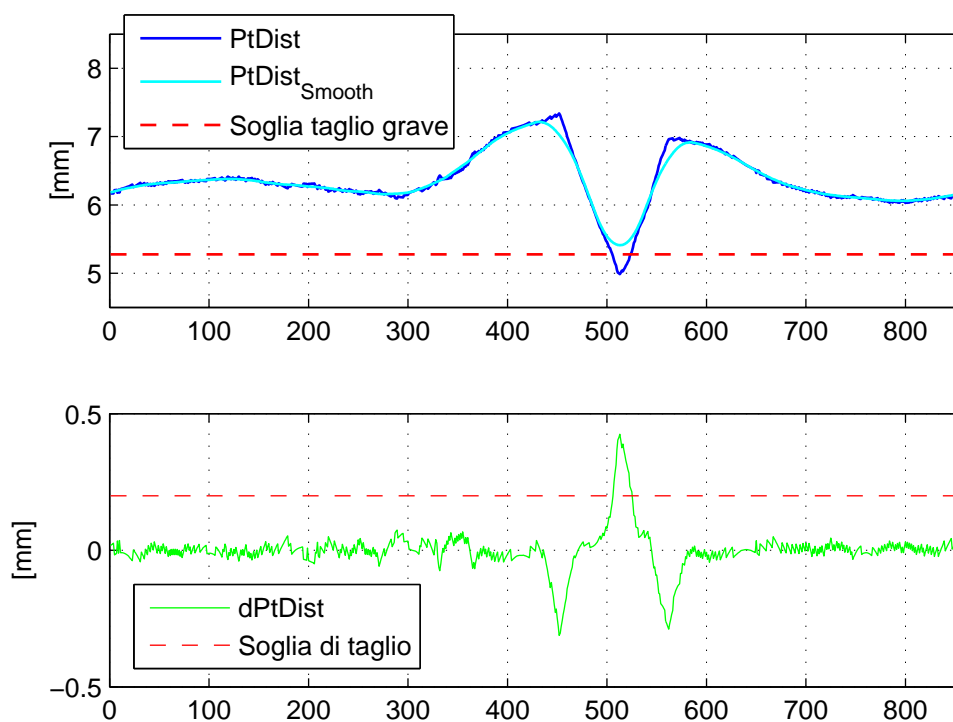


Figura 4.26: Algoritmo di riconoscimento taglio (metodo Tassarolo).

0.062s di quest'ultimo.

#### 4.3.2.2 Bicchiere irregolare

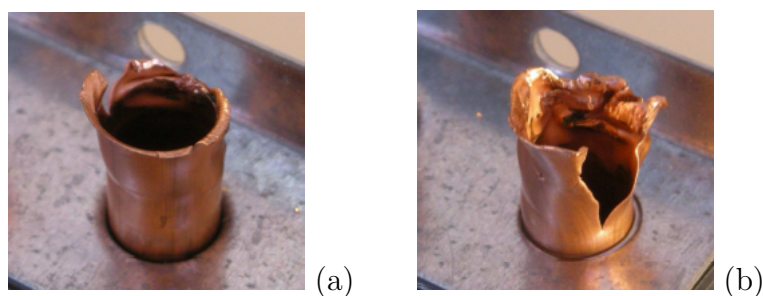


Figura 4.27: Esempi di bicchiere irregolare.

La presenza di questo difetto è caratterizzata dalla forma complessiva dell'imboccatura del bicchiere che risulta irregolare e spesso frastagliata, con numerosi tagli, sia gravi che lievi. Il tubo può essere piegato e presentare delle profonde fessure se non addirittura risultare completamente esploso.

La causa risiede nell'eccessivo stress del rame durante la fase di mandrinatura, dovuto all'alta sporgenza della forcina dalla piastra e/o alla flessione della barra mandrinante. Una causa antecedente può essere attribuita ad una forma già irregolare dell'imboccatura della forcina.

La sua criticità **alta**: in caso di tubo piegato il difetto costituisce un ostacolo all'inserimento della curva, che non può entrare correttamente nel bicchiere. Le eventuali fessure sul tubo e i tagli gravi sulla svasatura inficiano la brasatura prospettando una perdita certa nel circuito.

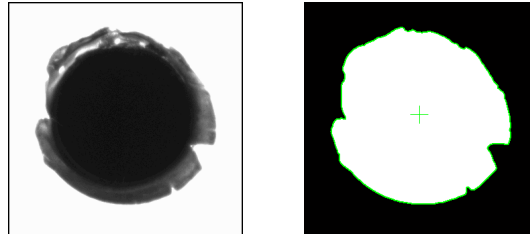


Figura 4.28: ROI di un tubo con bicchiere irregolare.

Una svasatura molto irregolare e frastagliata presenta genericamente numerosi tagli, lievi e gravi sul suo bordo. Pertanto la feature fondamentale risulta costituita dal bordo di svasatura; un esempio di questo difetto è l'immagine di figura 4.28. Un buon indicatore dell'irregolarità della svasatura è perciò sicuramente dato dalla varianza del suo profilo di ampiezza, costruito come spiegato precedentemente ordinando in senso antiorario i punti del contorno della regione di svasatura. Pertanto una prima valutazione sulla regolarità della svasatura viene da questo indicatore, a cui va legato un valore di soglia che separa la condizione di conformità da quella di anomalia.

Analogamente al secondo algoritmo visto per l'identificazione di eventuali tagli, l'algoritmo sviluppato per la rilevazione di eventuali irregolarità nel profilo del bicchiere è basato sull'analisi della differenza tra il vettore delle distanze dei punti del bordo della svasatura dal centro del tubo e la sua controparte smussata. In questo algoritmo il vettore differenza  $dPtDist = PtDistSmooth - PtDist$ , calcolato dopo aver notevolmente smussato  $PtDist$  (media su di un intervallo di ampiezza pari a 101) viene utilizzato per determinare il valore di varianza delle distanze dei punti del bordo della svasatura dal centro del tubo rispetto a quello che si potrebbe ritenere un andamento regolare di tale bordo ( $PtDistSmooth$ ). Il tubo viene ritenuto irregolare se questa varianza risulta superiore ad un certo valore di soglia da noi imposto pari a 5. Normalmente per un comune tubo privo di difetti questo valore non supera l'unità mentre nell'elaborazione della ROI di figura 4.28, la cui elaborazione è riportata in figura 4.29, la varianza di  $dPtDist$  raggiunge i 18.793.

A prima vista sembrerebbe sufficiente calcolare direttamente la varianza del vettore  $PtDist$ , in realtà tale valore non è affidabile in quanto non gestisce l'eventualità che il bordo della svasatura non sia perfettamente circolare anche se con un profilo privo di irregolarità o che la svasatura sia leggermente spostata rispetto al centro del bicchiere, lievi difetti che però possono comportare un elevato valore della varianza.

Così formulato questo algoritmo può identificare come irregolare bicchieri in realtà affetti da un taglio molto grave dal momento che un difetto di questo tipo genera chiaramente una svasatura dalla forma irregolare. Questo non è un problema in quanto in entrambi i casi la curva non va inserita. Sulla base di queste

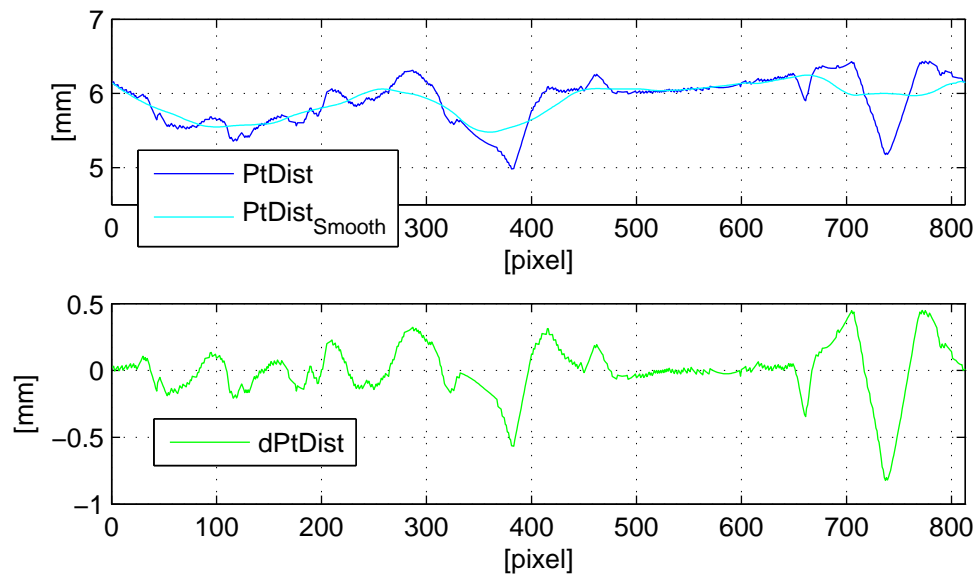


Figura 4.29: Algoritmo di riconoscimento bicchiere irregolare.

considerazioni ed essendo questo difetto considerato il più grave, se individuata la sua presenza in un tubo risulta superfluo proseguire con la ricerca di ulteriori difetti.

#### 4.3.2.3 Svasatura insufficiente

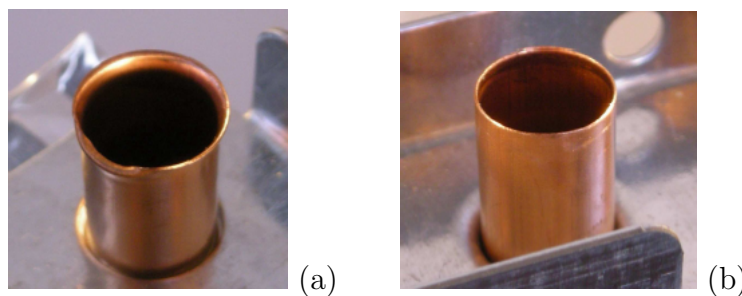


Figura 4.30: Esempi di svasatura insufficiente.

Questo difetto è caratterizzato da una svasatura all'imboccatura del bicchiere di entità molto scarsa, oltre le tolleranze previste. Questo difetto può risultare limitato ad un settore della svasatura oppure manifestarsi lungo tutto il bordo del bicchiere.

Nel caso di svasatura insufficiente limitatamente ad un settore della svasatura, di solito dovuto ad un ripiegamento verso l'interno della svasatura (Figura 4.30 a), la causa può essere imputata principalmente a due fattori:

1. Mandrinatura eseguita su di una forcina già difettata, ad esempio con l'imboccatura del tubo già irregolare o ripiegata;
2. Urto accidentale del tubo già mandrinato con un oggetto in fase di trasporto o di manipolazione;

La criticità di questa particolare tipologia di difetto è **medio/alta**: dipende dall'entità del ripiegamento verso l'interno che può impedire l'inserimento della curva.

Nel caso invece di svasatura del tutto assente (Figura 4.30 b) la causa dipende dalla mandrinatura incompleta della forcina dovuta ad una sporgenza dalla piastra molto inferiore alle sporgenze delle forcine adiacenti. La criticità di questo difetto è **alta**: se le coordinate del centro-tubo non sono precise l'inserimento della curva può essere pericoloso in quanto la curva potrebbe impuntarsi sul bordo (che non dispone di sufficiente invito conico di svasatura) con il rischio non solo di rovinare lo scambiatore ma anche di danneggiare il robot.

Entrambi i casi dipendono dallo spessore della svasatura e sono accumulati da una criticità elevata, per questo motivo vengono trattati come un unico difetto. Per la rilevazione di questo difetto risulta necessario poter valutare l'andamento dello spessore della svasatura lungo tutta la circonferenza del tubo. Osservando le ROI di figura 4.31, contenente le immagini di un tubo con un ripiegamento della svasatura, lo spessore della svasatura è dato dalla distanza tra il bordo della svasatura e la regione interna del tubo. Per poterne valutare lo spessore le feature di riferimento risultano essere il profilo della regione della svasatura ed il profilo del bicchiere.

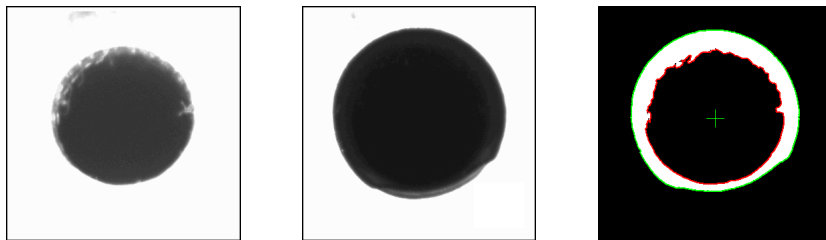


Figura 4.31: ROI di un tubo con un ripiegamento della svasatura.

Sulla base di queste considerazioni l'indicatore del difetto di svasatura assente può essere rappresentato dallo spessore medio della regione di svasatura. Pertanto, impostato un valore minimo di spessore che funga da soglia per considerare accettabile una svasatura, il riconoscimento del difetto risulta immediato.

Per quanto riguarda il difetto di ripiegatura, l'assottigliamento dello spessore della regione di svasatura non è più un fenomeno generalizzato su tutta la circonferenza ma è localizzato. Ne consegue che mentre lo spessore medio può rimanere entro valori accettabili, l'unico indicatore affidabile di questo difetto consiste nello spessore minimo della svasatura, che non deve scendere al di sotto di un certo valore di soglia. L'utilizzo di questi indicatori si basa sulla constatazione che una svasatura insufficiente è generalmente di forma regolare ed esente da tagli: difetti che possono a loro modo determinare un assottigliamento della svasatura. Per questo motivo ha senso eseguire questa analisi solamente se non sono già stati individuati precedentemente altri difetti come tagli gravi o bordo troppo irregolare.

Vediamo come opera questo algoritmo. Dopo avere estratto le due regioni binarizzate del bicchiere e della svasatura ed averne determinato per entrambe l'andamento del profilo del bordo rispetto al centro del tubo, come già visto per

l'analisi dei difetti precedenti, lo spessore della svasatura viene determinato dalla loro differenza.

Per calcolare correttamente l'andamento dello spessore della svasatura lungo la circonferenza del tubo è necessario valutare la differenza tra questi due vettori solamente nei punti in corrispondenza dello stesso angolo di percorrenza del bordo rispetto al centro. A questo punto, calcolati gli spessori medi e minimi della svasatura, se il valore di spessore medio risulta inferiore ad una certa soglia (attualmente fissata pari a  $0.7mm$ ) viene rilevato il difetto di svasatura insufficiente. Se tale difetto non viene rilevato allora l'algoritmo procede a confrontare il valore minimo di spessore della svasatura con una seconda soglia (fissata in questo caso pari a  $0.4mm$ ) per verificare la presenza di un'eventuale ripiegatura della svasatura.

Ad esempio applicando questo algoritmo alle ROI di figura 4.31, dal confronto tra valore medio dell'ampiezza della svasatura e la relativa soglia (Figura 4.32) la svasatura risulta sufficiente. Passando però al confronto tra il valore minimo e la seconda soglia rileviamo la presenza di almeno una ripiegatura (non può essere un taglio altrimenti sarebbe già stato rilevato precedentemente).

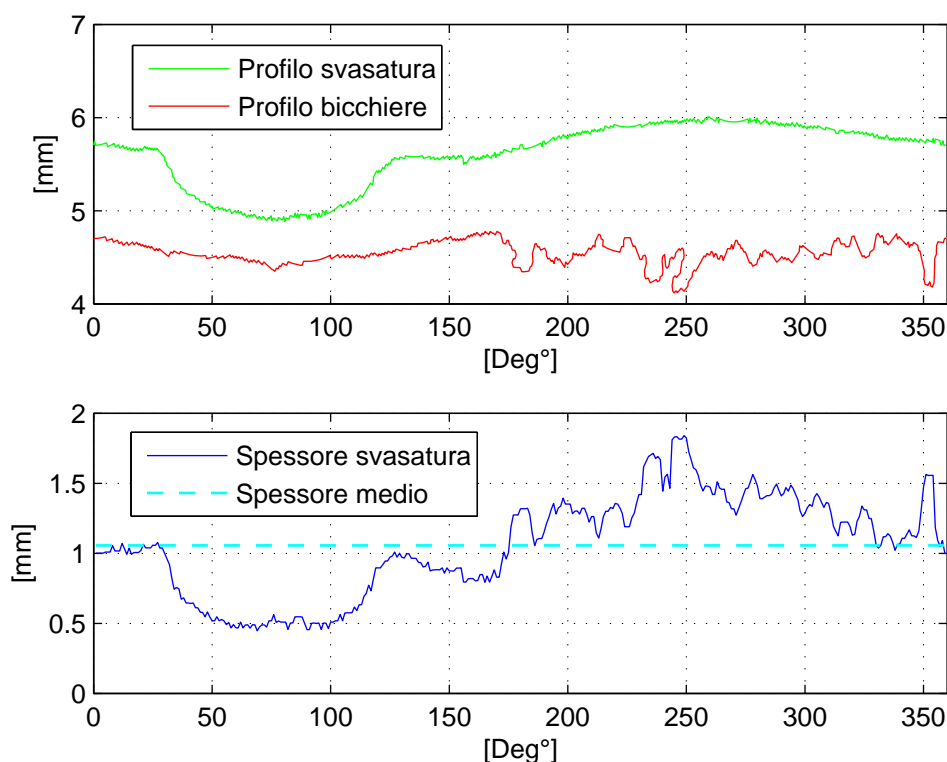


Figura 4.32: Algoritmo di riconoscimento svasatura insufficiente.



# Capitolo 5

## Test Sperimentali

In questo capitolo descriveremo brevemente tutta la fase sperimentale che ha completato e portato ad un funzionamento efficiente la cella di lavoro, in particolare andremo in un primo momento a descrivere le parti fondamentali costituenti il prototipo della cella realizzata nel laboratorio DIMEG dell'università di Padova, poi studieremo in dettaglio la parte relativa all'inserimento della curvetta analizzando le tempistiche e le problematiche associate. Andremo infine ad effettuare una serie di assemblaggi su diversi scambiatori per trarre delle statistiche globali su eventuali problematiche che possono nascere a causa della ripetizione dell'operazione di inserimento delle curvette.

### 5.1 Prototipo della cella

La fase centrale di questo lavoro di tesi ha visto la realizzazione di un prototipo di cella di lavoro per effettuare tests sull'assemblaggio degli scambiatori. Una rappresentazione della cella realizzata è visibile in Figura 5.2, dove si possono distinguere chiaramente le componenti principali che la contraddistinguono:

1. Robot: Il robot presente in Laboratorio è leggermente più piccolo del robot acquistato dall'azienda committente del progetto, in particolare i bracci del robot in laboratorio sono lunghi 550mm al posto dei 600mm del robot presente in azienda
2. Controllore: Il controllore utilizzato è piuttosto datato, in tal senso ne ha risentito la velocità di esecuzione del programma (si pensi a titolo di esempio che per stampare una scritta a video tale controllore impiega circa 3 decimi di secondo) e la disponibilità di I/O, infatti in questo controllore sono previsti solo 8 output e 12 input, problema che ci ha costretto in più occasioni e gestire input ed output manualmente.
3. Illuminazione Diffusa: Sono ben visibili i neon posti sopra lo scambiatore che realizzano l'illuminazione diffusa illustrata al capitolo 4: si osservi a tal proposito il reattore atto ad alimentare i neon ad una frequenza di 30 KHz, tale aumento di frequenza rispetto a quella di rete è indispensabile poichè l'immagine che viene acquisita ha un tempo di esposizione molto breve (ordine di centesimi di secondo) è alla frequenza di rete si avrebbero

immagini molto luminose o poco luminose a seconda dell'istante (rispetto alla tensione di forma sinusoidale) di apertura dell'obiettivo.

4. Illuminazione Laterale: Anche in questo caso sono ben visibili i neon posti al fianco dello scambiatore che realizzano l'illuminazione laterale illustrata al capitolo 4: in questo caso l'alimentazione proviene dalla rete infatti l'immagine acquisita ha un tempo di esposizione relativamente lungo (ordine di decimi di secondo) e non si risente del fenomeno di variazione della luminosità visto per illuminazione diffusa.
5. Isola pneumatica: In questa zona (si tratta di una comune tavola di legno con fissate sopra le elettrovalvole, vedi Figura 5.1) sono riunite tutte le elettrovalvole che gestiscono il funzionamento della pinza, in particolare si osservi da Figura 5.1 la condotta principale dell'aria, l'alimentatore delle elettrovalvole e i diversi output provenienti dal controllore (quando un output è abilitato il controllore mette in corto circuito i due terminali relativi, in modo che l'alimentazione possa raggiungere i due morsetti dell'elettrovalvola associata).

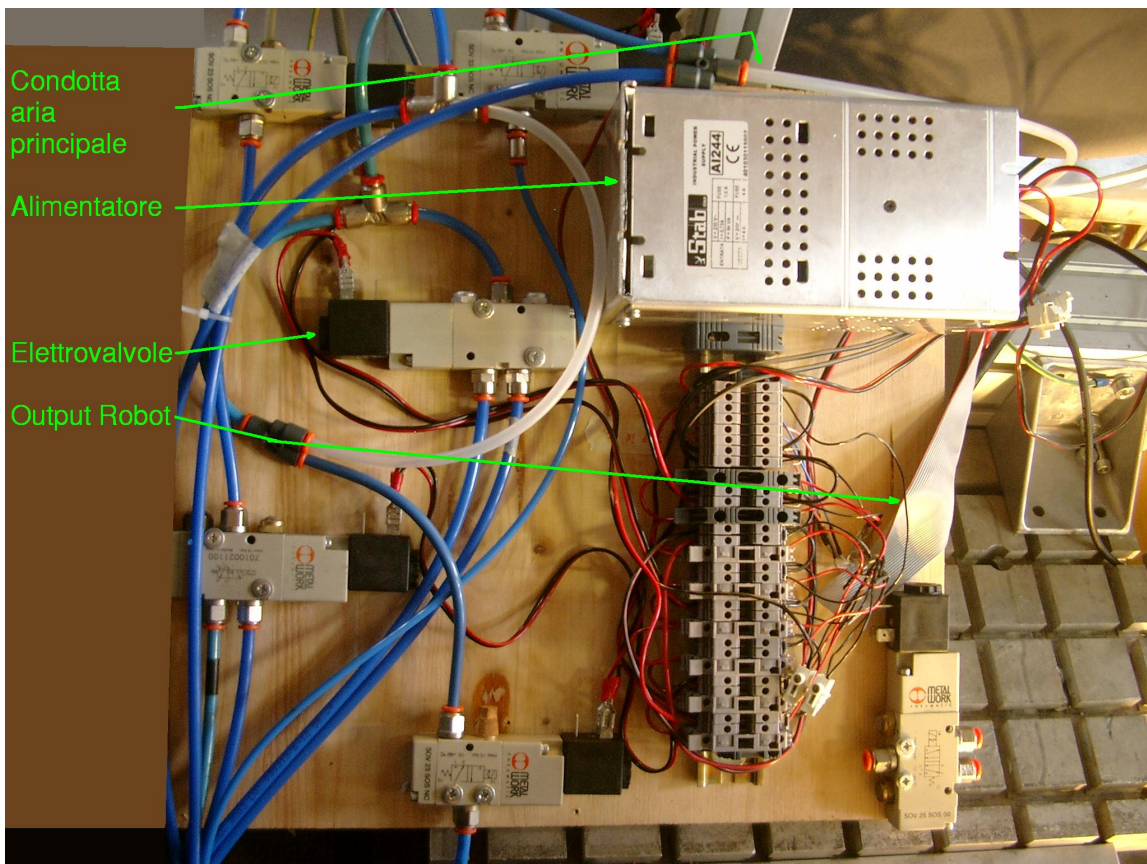


Figura 5.1: Isola pneumatica



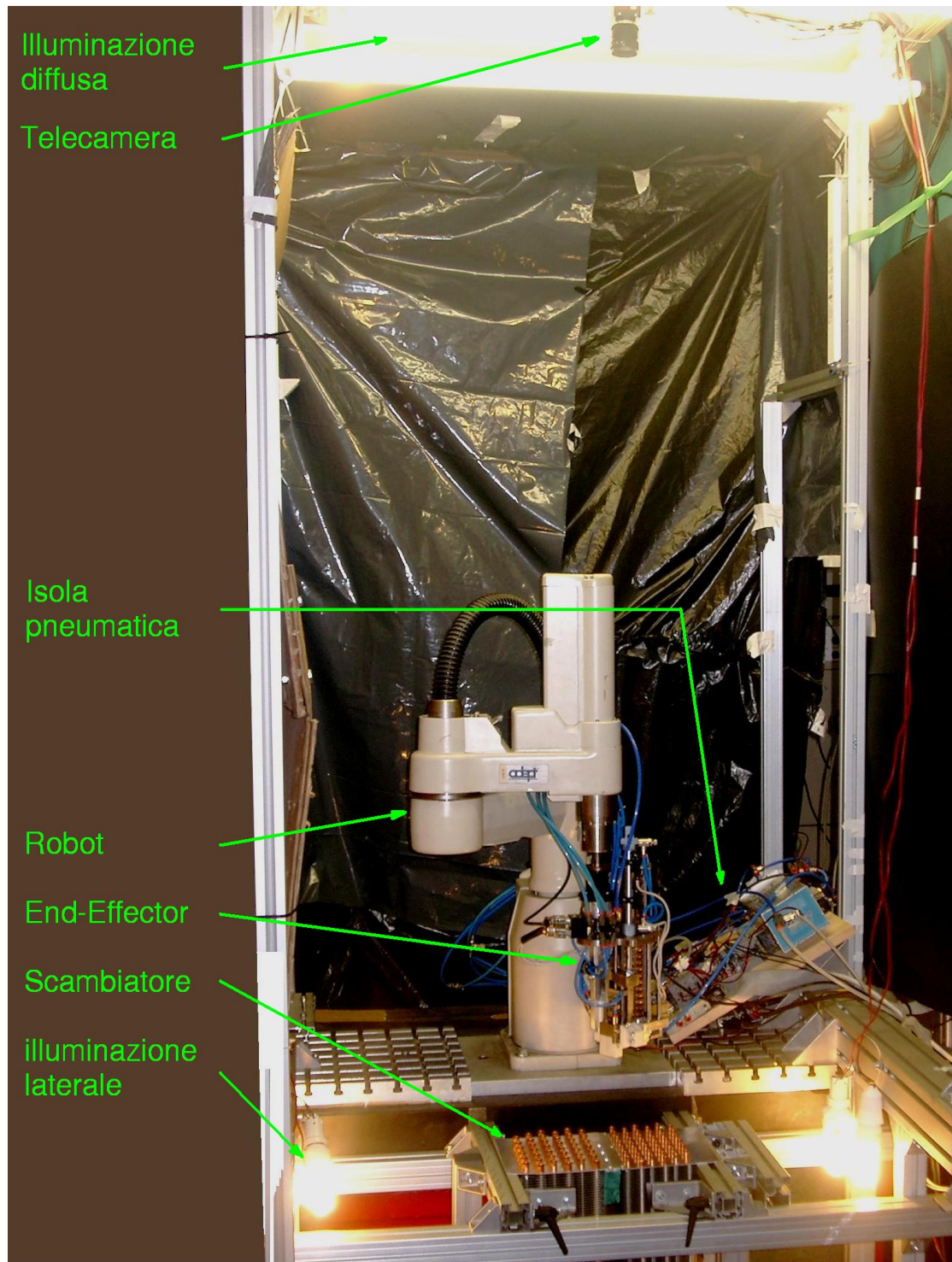


Figura 5.2: Immagine del prototipo di cella realizzato presso i laboratori DIMEG dell'università di Padova

## 5.2 Ciclo di Inserimento Curvetta

Come già introdotto, andremo ora a studiare in dettaglio il codice Adept (codice scritto in linguaggio V+ appartenente al controllore del robot della Adept Technologies Corporation) che realizza il vero e proprio inserimento di una curvetta. Si riprenda a tal proposito la comunicazione tra il controllore e MatLab vista al capitolo 2, in particolare si veda la Figura 2.3: ogni pacchetto inviato da MatLab ha come header un ID che si traduce nell'esecuzione di una particolare operazione da parte del controllore. Studiamo il caso in cui l'ID in esame sia quello relativo al posizionamento della curvetta:

```

1  VALUE (str[1] < 201) AND (str[1] > 100):
2  ;impostazione velocita'
3  SPEED (str[1]-100) ALWAYS
4  ;verifica la pinza per questo inserimento, altrimenti cambia
5  CALL checkinnesti(pz, loc_pz, matrice_trasf, cv_totali)
6  IF str[8] <> pz THEN
7      CALL prendipinza(str[8])
8  END

9  ;Controllo di blocco del cilindro principale in posizione bassa.
10 nwait = 0
11 inseff = 0
12 REACT sgn_inserim_dow, inserimentoeff
13 WHILE SIG(-sgn_inserim_up) AND (inseff = 0) AND (nwait <= 60) DO
14     nwait = nwait+1
15     WAIT.EVENT , 0.01
16 END
17 IF nwait == 60 THEN
18     CALL disabilita(lun, $p_inserim_down+"")
19     TYPE "EMERGENZA!! Bloccaggio pistone verso il basso"
20 END

21 ;settaggio quota dz per la pinza (Matlab invia una quota, se >0 la usiamo
22 ;altrimenti viene utilizzata la quota impostata in inizializzazioni
23 IF str[3] > 0 THEN
24     CALL creavettore(str, str[3])
25 ELSE
26     CALL creavettore(str, dz_pz)
27 END

28 ;verifica posizione ricevuta
29 CALL verificaemuoovi(str[],1) ; 1 indica spazio giunti (2 altrimenti)
30 CALL posiziona(lun)
31 curv_inserite = curv_inserite+1

32 ;comunicazione con MatLab
33 IF estop = 0 THEN
34     WRITE (lun) $ok
35 ELSE
36     WRITE (lun) $estop
37 END

```

Figura 5.3: Codice V+ relativo all'ID di posizionamento curvetta

Descriviamo brevemente quanto riportato nel codice: innanzitutto l'ID non è un numero ma una fascia di numeri, che possono andare da 101 a 200 in modo da poter avere un'informazione aggiuntiva sulla velocità (in percentuale rispetto alla velocità massima) con cui si vuole effettuare l'operazione di inserimento.

Dalla riga 5 alla 8 viene analizzata la pinza presente sulla flangia del robot e confrontata con la pinza indicata nell'ottavo byte della stringa inviata da MatLab: se le due pinze sono diverse si procede con il rilascio della pinza innestata e con l'aggancio della pinza desiderata. Se l'operazione non va a buon fine (emergenze) viene disabilitata la potenza al robot e tolta la pressione nel circuito dell'aria, viene settata una variabile *estop* = 1 ed il codice prosegue.

Dalla riga 9 alla 20 viene fatto un controllo sulla posizione del fusto del cilindro principale di inserimento, infatti se il fusto è fuori completamente dal cilindro un movimento del robot potrebbe portare ad una collisione con lo scambiatore. Per tale controllo sono stati posti due sensori di prossimità: uno posto nell'estremità fissa del cilindro in modo da rilevare il caso in cui il fusto è completamente dentro il cilindro (tale sensore è rappresentato nel codice dal segnale *sgn\_inserim\_up*) e l'altro in una posizione intermedia del cilindro, una posizione considerata il limite per una movimentazione sicura (quando il fusto passa da questa posizione la variabile *inseff* presente nel codice viene posta ad 1, questo grazie al react posto sul segnale associato a tale sensore<sup>1</sup>). Anche in questo caso se l'operazione non va a buon fine vengono effettuate le azioni viste precedentemente grazie al programma *disabilita*.

Dalla riga 21 alla 27 viene impostata la quota *z* di assemblaggio: tale quota è inserita nelle inizializzazioni (quota ottenuta da prove sperimentali) e viene utilizzata a meno che il terzo byte della stringa inviata da MatLab sia maggiore di 0: in tal caso si utilizza con quota *z* proprio il dato presente in quel byte.

Dalla riga 28 alla 31 dopo un controllo che la locazione inviata da MatLab sia raggiungibile e non porti a collisioni si richiama il programma per il vero e proprio inserimento che tratteremo in dettaglio in seguito.

Dalla riga 32 alla 37 viene comunicato a MatLab (come già descritto al Capitolo 2 sulla comunicazione) l'esito dell'operazione: se non sono mai intervenuti problemi si invia un OK altrimenti (in qualche fase è stata settata la variabile *estop* ad 1) si notifica a MatLab la presenza di un'emergenza. Si osservi che se sono intervenuti problemi non considerati di emergenza (ad esempio alla riga 7 la locazione magazzino della pinza da rilasciare è già occupata) viene in ogni caso tolta l'alimentazione al robot e la pressione nel circuito dell'aria<sup>2</sup> ma la variabile *estop* resta a 0 e viene notificato a MatLab il problema riscontrato: in questa condizione MatLab va nello stato *Idle*, cioè invia un pacchetto di OK. Alla riga 34 viene inviato un OK anche da Adept ed i due continuano in questo modo finché non viene premuto il pulsante RESTART presente sull'interfaccia principale.

<sup>1</sup>Il segnale associato è indicato nel codice con *sgn\_inserim\_dow*, quindi ad un fronte di salita di tale segnale viene chiamato il programma *inserimentooeff()* che non fa altro se non settare la variabile *inseff* ad 1.

<sup>2</sup>Si osservi che tutte le istruzioni successive (dalla riga 7 in poi) vengono eseguite ma non portano a nessun movimento né del robot né della pneumatica.

Andiamo ad analizzare la vera e propria sequenza di inserimento: il programma *posiziona()*: come da Figura 5.4 e 5.5 è facile osservare che le parti principali che costituiscono l'effettivo inserimento della curvetta (erano già state introdotte al Cap 2, End-Effector) sono:

- inizio vuoto + blocco caricatore
- comando slitta indietro
- controllo del blocco della slitta in avanti
- comando slitta avanti
- controllo del blocco della slitta indietro
- abbassamento pistone principale + sblocco caricatore
- verifica presenza del vuoto
- fine vuoto + inizio del soffio
- controllo di blocco del pistone principale in alto
- comando pistone principale verso l'alto
- verifica ultima curvetta

La verifica dell'ultima curvetta serve esclusivamente a portare la pinza innestata a caricare nell'eventualità che la curvetta inserita fosse l'ultima del caricatore.

E' importante osservare le pause presenti tra le varie operazioni (ad es. righe 16, 30, 35, etc.): tali pause sono servite per ottimizzare le prestazioni della fase di inserimento: si osservi a tale proposito la Figura 5.6.

```

1  .PROGRAM posiziona()
2  ;inizio vuoto + blocco caricatore
3  SIGNAL -out_soffia, out_vuoto, out_carica

4  ; slitta indietro
5  SIGNAL -out_slitta_ind

6  ; controllo di blocco della slitta in avanti
7  nwait = 0
8  WHILE SIG(-sgn_slitta_ind) AND (nwait <= 60) DO
9      nwait = nwait+1
10     WAIT.EVENT , 0.01
11  END
12  IF nwait == 60 THEN
13     CALL disabilita(lun, $p_slitta_av+"" )
14     TYPE "EMERGENZA!! Bloccaggio slitta in avanti"
15  END

```

Figura 5.4: Codice V+ relativo al programma di inserimento della curvetta (partel)

```

16  WAIT.EVENT , 0.08
17  BREAK

18  ;avanzamento slitta
19  SIGNAL out_slitta_ind

20  ;controllo di blocco della slitta indietro
21  nwait = 0
22  WHILE SIG(-sgn_slitta_av) AND (nwait <= 60) DO
23      nwait = nwait+1
24      WAIT.EVENT , 0.01
25  END
26  IF nwait == 60 THEN
27      CALL disabilita(lun, $p_slitta_ind+"")
28      TYPE "EMERGENZA!! Bloccaggio slitta indietro"
29  END

30  WAIT.EVENT , 0.1

31  ;inserimento curvetta e tic caricatore
32  SIGNAL out_inserim, -out_carica

33  ;verifica vuoto
34  CALL checkvuoto(lun)

35  WAIT.EVENT , 0.08

36  ; fine del vuoto
37  SIGNAL -out_vuoto, out_soffia

38  ; controllo di blocco del cilindro principale in posizione alta
39  nwait = 0
40  WHILE SIG(sgn_inserim_up) AND (nwait <= 60) DO
41      nwait = nwait+1
42      WAIT.EVENT , 0.01
43  END
44  IF nwait == 60 THEN
45      CALL disabilita(lun, $p_inserim_up+"")
46      TYPE "EMERGENZA!! Bloccaggio pistone di inserimento verso l'alto"
47  END
48  WAIT.EVENT , 0.05

49  ;ritorno verso l'alto
50  SIGNAL -out_inserim

51  ;controllo penultima curva
52  CALL penultimacurva(lun)
53  .END

```

Figura 5.5: Codice V+ relativo al programma di inserimento della curvetta (parte2)

EndEffector_3/8_p25	Prove a tempi non ottimizzati				Prove a tempi ottimizzati (ciclo 19 s)				Note
	frame(140 fps)	tempi assoluti[ms]	tempi relativi [ms]		frame(140 fps)	tempi assoluti[ms]	tempi relativi [ms]		
1	inizio vuoto + blocco caricatore	0	0		0	0			
2	c. slitta indietro + csa	4	30	30	4	30	30		
3	fine slitta indietro + inizio caduta curva	10	73	43	10	73	43		
4	fine caduta curva	33	237	164	27	194	120	attesa caduta curva	
5	c. slitta avanti + csi	48	344	107	28	201	7		
6	fine slitta avanti	53	380	36	32	230	29		
7	c. pistone giù + cv + cpu	65	466	86	44	315	86		
8	fine pistone giù	98	701	236	75	537	221		
9	caduta curva	116	830	129	75	537	0		
10	c. pistone su	142	1.016	186	88	630	93	attesa inserimento	
11	fine pistone su	170	1.216	200	116	830	200		
		tempo singolo inserimento [ms]		1216	tempo singolo inserimento [ms]		830		

## Note

csa = Controllo blocco Slitta Avanti  
 csi = Controllo blocco Slitta Indietro  
 cv = Controllo Vuoto  
 cpu = Controllo blocco Pistone Up

Figura 5.6: Tabella con le prestazioni ottenute nelle diverse fasi dell'inserimento.

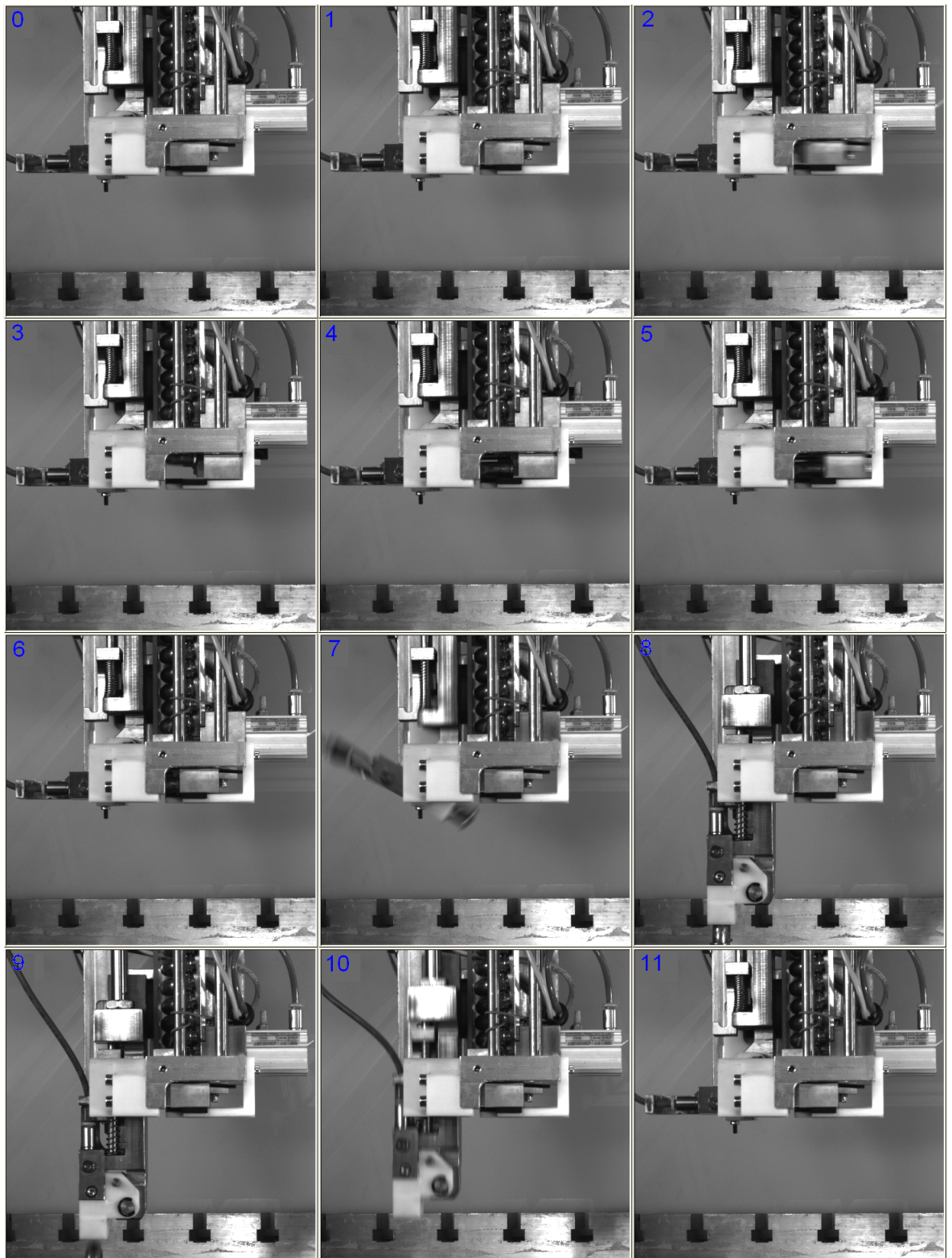


Figura 5.7: Fasi delle diverse operazioni descritte in Figura5.6.

Le prestazioni ottenute ottimizzando le pause tra le diverse operazioni in modo empirico hanno portato ai risultati visibili in Figura 5.6 sulle 3 colonne relative alla voce “Prove a tempi non ottimizzati” con i quali l’inserimento di ogni curvetta avveniva in circa 1.5 secondi (si ricordi come n operatore esperto impieghi 2 secondi a curvetta). L’utilizzo di una telecamera con un altro Frame Rate (circa 140 fps) ci ha permesso di analizzare in dettaglio i tempi morti tra le varie operazioni: in particolare si osservino i numeri in rosso presenti in Figura 5.6, tralasciando per un attimo il tempo dovuto alla caduta della curva è immediato osservare che tali tempi possono essere ridotti riducendo le pause tra le diverse operazioni ed adottando delle strategie di esecuzione del programma di posizionamento più affinate. Ad esempio il controllo del vuoto (e quindi della presenza della curvetta all’interno della culla) non è più fatto prima del comando di discesa del pistone principale (come sembrerebbe più ovvio, visto che è inutile abbassare il pistone se la curvetta è assente) ma dopo, poichè la discesa del pistone ha un suo tempo meccanico che non può essere ridotto e fare controlli all’interno di tale movimento equivale a fare controlli in tempo mascherato.

Riprendiamo brevemente il tempo dovuto alla caduta della curvetta: dalle immagini a rallentatore è stato possibile osservare un grosso spreco di tempo dovuto all’assestamento della curvetta sulla base della slitta : infatti la curvetta, avendo il baricentro spostato in avanti, tende a ruotare nel momento in cui la slitta si muove indietro: a tal proposito è stato scavato sulla slitta stessa un leggero piano inclinato (vedi Figura 5.8) che ruota la curvetta nel senso opposto in modo che arrivi a caduta terminata in posizione pressocchè orizzontale. Un altro problema erano i rimbalzi che la curvetta faceva dopo la caduta sulla slitta: a tal proposito è stato modificato il fondo della slitta (vedi Figura 5.8) con diversi materiali che hanno portato ad una riduzione significativa del tempo di attesa dovuto alla caduta della curva.

Tutte le soluzioni presentate hanno portato ad ottenere un tempo di inserimento per singola curvetta di 1 secondo.

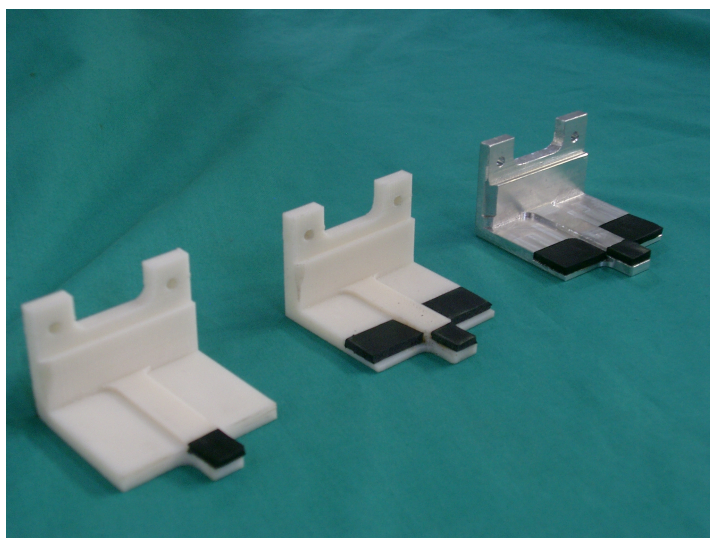


Figura 5.8: Diverse tipologie di slitte utilizzate.



## 5.3 Test di assemblaggio

In questa sezione riportiamo brevemente i problemi riscontrati nell'imponente set di test che sono stati fatti in laboratorio.

I problemi che abbiamo affrontato sono stati essenzialmente due: l'inceppamento delle curvette lungo il canale del caricatore (vedi Figura 5.9) e la caduta della curvetta dal fondo della slitta (la curvetta appena lasciata dal cariatore e pronta ad essere spinta dalla slitta all'interno della culla) in occasione di movimenti bruschi o rotazioni dovute al posizionamento del robot sulla locazione di inserimento.

Il primo problema ha obbligato ad una modifica del programma, in particolare se il canale è ostruito allora il sensore posto al termine del caricatore non rileva curvette a questo punto viene riposta la pinza in magazzino per il suo caricamento. Prelevata la pinza se il sensore ancora non da segno di presenza curvette si procede con il muovere velocemente e cambiando direzione repentinamente il robot per brevi tratti, ottenendo di fatto di sbattere la pinza. Finita questa fase se il sensore ancora non rileva curvette si procedere con l'arrestare la cella ed avvisare l'operatore.

Il secondo problema ha portato a test per il *tuning* dei parametri presenti nell'algoritmo per il calcolo del percorso ottimo, e ad un'analisi delle locazioni prima di un qualsiasi movimento: se il movimento o la rotazione da fare superano dei valori trovati sperimentalmente allora si esegue quel movimento con una velocità ridotta, per poi ripristinare la velocità precedente al termine dell'operazione di movimentazione.

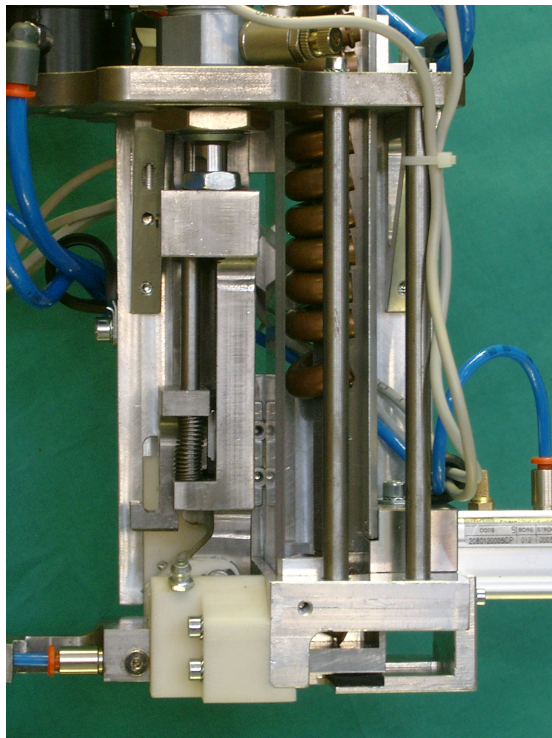


Figura 5.9: Problema di ostruzione del caricatore



# Conclusioni

In questo lavoro di tesi sono state ri-elaborate diverse parti della cella studiate in tesi precedenti: in particolare l'attività si è incentrata sulla messa in funzione della calibrazione della cella, sulle modalità di movimento e di gestione degli errori del robot, sull'attività di visione soprattutto a riguardo dell'identificazione degli scambiatori.

Diversi algoritmi sono stati elaborati, in particolare ricordiamo il calcolo del percorso ottimo di assemblaggio, la gestione dei cambi pinza e l'analisi/ memorizzazione/gestione dei difetti.

E' stata svolta in laboratorio un'attività di tests che ha portato al perfetto inserimento delle curvette da parte del robot. Resta da testare tutta la parte relativa alla gestione del magazzino: gli algoritmi sono stati tutti implementati (gestione della tavola rotante, dei cambi pinza, delle emergenze nel magazzino) ma è chiaro che piccole modifiche saranno effettuate una volta ottenuti dei risultati sperimentali.

Questo lavoro ha permesso di toccare con mano praticamente tutte le fasi della progettazione di un sistema automatizzato, dalla formulazione delle specifiche, al raggiungimento della soluzione finale. È stato di fondamentale importanza per acquisire una metodologia con cui affrontare le diverse tipologie di problemi incontrati e per apprendere le dinamiche interne allo sviluppo di un progetto dove ogni sua parte deve essere portata avanti in armonia con le altre al fine di garantire il migliore funzionamento possibile.



# Appendici

## A Utilizzo della telecamera a matrice Pike 505B

### A.1 Installazione della telecamera

La telecamera va semplicemente collegata ad un qualsiasi ingresso FireWire 800 (IEEE 1394 b) dell'elaboratore tramite l'apposito cavo di collegamento fornito.

L'azienda AVT (AlliedVisionTechnologies) produttrice della telecamera fornisce sul suo sito [www.alliedvisiontec.com](http://www.alliedvisiontec.com) il pacchetto software AVT FirePackage contenente i driver necessari alla sua installazione oltre ad una comoda interfaccia di utilizzo della telecamera. Nel caso in cui la telecamera venga utilizzata sotto Windows XP ServicePack 2 e 3 o Vista, a causa della limitazione in banda delle interfacce FireWire 800 a 100bit/s dovute al sistema operativo, è necessario installare il pacchetto driver AVT 1394 Bus Driver Package, scaricabile sempre dal sito della AVT.

### A.1 Utilizzo della telecamera con MATLAB

Utilizzando l'Image Acquisition Toolbox di MATLAB, per creare un video input object bisogna specificare il tipo di adattatore utilizzato, nel nostro caso il "dcam", il numero di identificazione del dispositivo di acquisizione utilizzato ed il formato dell'immagine da acquisire. Ad esempio, per utilizzare la Pike505B alla sua massima risoluzione ( $2452 \times 2054$ pixel) si utilizza il comando:

```
VidObj = videoinput('dcam', 1, 'F7_Y8_2452x2054');
```

Una volta creato l'oggetto video input, il settaggio dei parametri e l'esecuzione dell'acquisizione avviene come per una qualsiasi telecamera ad area descritti nel manuale dell'Image Acquisition Toolbox.

Il formato dcam non fa riferimento ai driver forniti dall'azienda produttrice ma ad un driver generico sviluppato dalla Carnegie Mellon University (CMU). Questi driver sono scaricabili dal sito [www.cs.cmu.edu/~iwan/1394](http://www.cs.cmu.edu/~iwan/1394). Purtroppo questi driver dcam di cui necessita MATLAB per comunicare con la telecamera danno dei problemi nell'utilizzo di alcuni formati. Tutto funziona perfettamente con MATLAB 2006b utilizzando i driver versione 6.3. Le versioni successive di MATLAB necessitano di driver dcam più recenti, ma si sono riscontrati dei problemi nell'utilizzo degli stessi: il formato di risoluzione voluto e per il quale la telecamera è stata scelta (F7 Y8 2452x2054) spesso non viene riconosciuto dal software al momento di creazione dell'oggetto video; tale problema si presenta in maniera e tempistica del tutto aleatorie, senza alcuna sistematicità visibile e quindi risolvibile.

A tal proposito è stato contattato un tecnico AVT tramite il forum dedicato agli utilizzatori di MATLAB, che ha confermato il disagio e garantito l'imminente emissione di un driver adaptor specifico per i prodotti AVT da utilizzare al posto di quelli standard sviluppati dalla Carnegie Mellon University. Dopo alcuni mesi dall'inizio di questo progetto la AVT ha messo a disposizione a chi ne facesse richiesta una prima versione beta di questo adaptor per MATLAB costituito da due file: AVTMatlabAdaptor.dll e AVTMatlabAdaptor.imdf, scaricabili all'indirizzo [www.alliedvisiontec.com/fileadmin/files/software/AVTMatlabAdaptor\\_v0.3.0.zip](http://www.alliedvisiontec.com/fileadmin/files/software/AVTMatlabAdaptor_v0.3.0.zip).

Per poterlo utilizzare è necessario installare prima l'AVT FirePackage versione 2.10 o superiore ed il corrispondente 1394 bus driver. A questo punto per aggiungerlo nella lista degli adaptor disponibili dell'Image Acquisition Toolbox di MATLAB è sufficiente digitare il comando *imaqregister* specificando il percorso dove è stato salvato il file AVTMatlabAdaptor.dll, d'ora in avanti digitando la funzione *imaqhwinfo* all'interno della lista degli adaptor disponibili comparirà anche il nome '*avtmatlabadaptor*'.

Per utilizzare la Pike505B con questo nuovo adaptor alla massima risoluzione ( $2452 \times 2054$  pixel) si utilizza il comando:

```
Vid=videoinput( 'avtmatlabadaptor' ,1 , 'F7M0_Mono8_2452x2054' )
```

In attesa di questo adaptor che consente un controllo della telecamera direttamente da MATLAB è stato sviluppato in linguaggio C++ un file eseguibile che provvede all'acquisizione delle immagini.

### A.2.1 Programma di acquisizione in C++

Il programma di acquisizione automatica in C++ è stato realizzato modificando un preesistente programma contenuto all'interno del pacchetto ActiveFirePackage assieme al suo codice sorgente.

Non appena lanciato il relativo file eseguibile, questo programma provvede automaticamente ad inizializzare la telecamera settando il tempo di esposizione ad un valore fisso preimpostato. Attraverso questo programma non è purtroppo possibile settare il tempo di esposizione da MATAB in quanto il file eseguibile viene compilato con già al suo interno tale valore. Per questo motivo sono stati compilati più eseguibili, uno per ogni diverso tempo di esposizione utilizzato e genericamente chiamati PikeCamXXX.exe, dove al posto di XXX viene specificato il tempo di esposizione utilizzato durante le prove di laboratorio (80, 100, 800, 1000, ecc.).

Immediatamente dopo aver inizializzato la telecamera il programma provvede ad inviare alla telecamera un segnale di trigger per l'acquisizione di un'immagine. L'immagine acquisita viene salvata nell'hard disk principale C: dell'elaboratore sottoforma di file ".bmp" con il nome PikeImageXXX.bmp.

Non appena l'immagine è stata salvata il programma termina automaticamente.

Ricorrendo a questo programma, per acquisire un'immagine con MATLAB è sufficiente lanciare uno di questi eseguibili con il comando *open*, ed attendere la comparsa su disco fisso del relativo file C:\PikeImageXXX.bmp, che deve essere letto con il comando *imread* e subito dopo eliminato.

L'utilizzo di questi eseguibili non è ottimale in quanto la telecamera viene nuovamente inizializzata con conseguente perdita di tempo ad ogni acquisizione. In aggiunta la scrittura e successiva lettura su disco fisso del file PikeImageXXX.bmp richiede dell'ulteriore tempo in più. Risulta inoltre scomodo dover compilare un nuovo file eseguibile per ogni nuovo settaggio del tempo di esposizione della telecamera che si intende utilizzare. Va detto infine che questo programma sviluppato in C++ allo stato attuale non risulta perfettamente stabile e necessiterebbe perciò di un aggiustamento del codice. Per tutti questi motivi questa soluzione va considerata provvisoria e a soli fini sperimentali in attesa dei nuovi driver che consentano l'utilizzo della telecamera Pike 505B direttamente da MATLAB.





# Bibliografia

- [1] *V+ Language Quick Reference*, Manuale di utilizzo Adept
- [2] *V+ Language User's Guide*, Manuale di utilizzo Adept
- [3] J.Vincenzi. *Automazione di sistema di assemblaggio tramite sistema di visione*, Tesi di laurea, Padova, 2008
- [4] G.Boothroyd. *Assembly Automation and Product Design, Second Edition*, Boca Raton (etc.) : Taylor & Francis, 2005
- [5] J.Craig. *Introduction to Robotics, Mechanics and Control, Third Edition*, Upper Saddle River, NJ : Pearson Education, 2005
- [6] A. Rossi, *Dispensa delle lezioni di Meccanica dei Robot*, Padova, 2005
- [7] J.Vincenzi. *Automazione di sistema di assemblaggio tramite sistema di visione*, Tesi di laurea, Padova, 2008
- [8] A.Carli. *Progetto di cella robotizzata per l'assemblaggio di batterie alettate*, Tesi di laurea, Padova, 2009
- [9] D.Cattai. *Sistema robotizzato di assemblaggio per scambiatori*, Appunti di Tesi di laurea, Padova, 2009
- [10] M.Fardin. *Sistemi di visione per l'automazione industriale*, Tesi di laurea, Padova, 2009
- [11] G. Picci. *Metodi statistici per l'identificazione di sistemi lineari*, Dispense, Padova, 2007
- [12] I.Tessarolo. *Sistema di visione per cella automatizzata*, Tesi di laurea, Padova, 2010



# Ringraziamenti

Desidero innanzitutto ringraziare il Professor Aldo Rossi ed il Professor Giulio Rosati per avermi dato la possibilità di partecipare ad un importante progetto, ambizioso e costruttivo. Inoltre, ringrazio sentitamente il Dr Stefano Cenci ed il Dr Andrea Carli che sono stati sempre disponibili a dirimere i miei dubbi durante il corso di questo lavoro.

Ringrazio i miei compagni di corso, in particolare Marco, Damiano, Spola, Matteo e Daniele per le piacevoli discussioni e per tutte le collaborazioni durante questi anni all'università.

Con immensa gioia vorrei dire grazie a tutti i miei amici del collegio "GREGORIANUM", che mi hanno insegnato ad apprezzare e vivere la carriera universitaria: grazie soprattutto al direttore Scarpis, a Don Leonardo, al mitico Cisco, al glorioso Decano, all'infallibile Dott. Bo, al Batt ed i suoi prelibati caffè con la crema, a Valerio e alle sue interminabili conquiste, a Dennis per le notti passate a discutere, al Peri per le sue teorie, a Pilon per le Philip Morris condivise e a Palli per la sua simpatia.

Finalmente ho l'occasione di ringraziare con affetto i miei genitori per il sostegno ed il grande aiuto che mi hanno dato non solo con l'università, ma in tutta la mia vita: esempi di instancabile volontà e determinazione da cui non smetterò mai di imparare.

Grazie.