



UNIVERSITÀ DEGLI STUDI DI PADOVA

SCUOLA DI INGEGNERIA
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA MECCATRONICA

TESI DI LAUREA MAGISTRALE

**PRESA AL VOLO DI OGGETTI IN
MOVIMENTI DA NASTRO TRASPORTATORE
CIRCOLARE**

Relatore: Boschetti Giovanni

Correlatore:

Laureando: Fochesato Manuele
1019798-IMC

ANNO ACCADEMICO: 2012-13

SOMMARIO

La presa al volo di oggetti in movimento da nastro trasportatore circolare è una tecnica innovativa che è stata applicata per ridurre gli sprechi in termini di tempo e costo. A livello industriale la presa degli oggetti avviene arrestando ogni qual volta il nastro. Quindi lo scopo di tale progetto è di realizzare un metodo in modo che il sistema lavori senza nessuna interruzione. Viene utilizzata una telecamera per raccogliere le informazioni sulla posizione degli oggetti. Unendo le informazioni ricevute dalla telecamera e dalla lettura dell'angolo encoder del nastro conosco con precisione dove si trova il pezzo e riesco così a seguirlo con il manipolatore. Questo ci porta ad avere il risultato sperato.

RINGRAZIAMENTI

Desidero innanzitutto ringraziare il mio relatore, l'Ingegnere Boschetti Giovanni, per avermi dato la possibilità di svolgere un lavoro nel settore che di più mi affascina e in cui spererei di fare carriera nel prossimo domani.

Inoltre, ringrazio l'Ingegnere Aldo per avermi aiutato a realizzare il setup di partenza.

Vorrei anche ringraziare i miei compagni di corso per i bellissimi cinque anni trascorsi insieme, in particolare Alberto che mi ha sempre aiutato nei momenti di bisogno.

Infine, ho desiderio di ringraziare con affetto i miei genitori, in particolare la mamma Rita, per il sostegno e il grande aiuto che mi hanno dato, senza dimenticare gli amici e mio fratello.

INDICE

1	INTRODUZIONE	1
2	IL NASTRO TRASPORTATORE CIRCOLARE	3
2.1	Introduzione	3
2.1.1	Cosa può alimentare	4
2.1.2	Come si collega al manipolatore	4
2.1.3	La retroilluminazione e la superficie rotante	4
2.1.4	Come lavora il flexibowl	5
2.1.5	Come interfacciarla	5
2.1.6	Vantaggi	5
3	ADEPTSIGHT: SOFTWARE DI VISIONE	7
3.1	Introduzione	7
3.2	Il progetto visione	8
3.2.1	Sequence Manager	8
3.2.2	System Devices Manager	9
3.3	Sequence Editor	10
3.4	strumenti di adeptsight	13
3.4.1	Acquisizioni d'immagini	14
3.4.2	Strumenti di movimento	14
3.4.3	Strumenti di cerca e individuazione	14
3.4.4	Strumenti di colore	15
3.4.5	Strumenti di elaborazione delle immagini	15
3.4.6	Strumenti d'ispezione	15
3.4.7	Altri strumenti	16
3.5	Strumenti usati per realizzare il progetto di tesi	16
3.5.1	Strumento Acquire Image	16
3.5.2	Strumento Locator	18
4	PRINCIPIO DI FUNZIONAMENTO	27
4.1	Introduzione	27
4.2	Spazio di lavoro	30
4.3	calibrazione del sistema	30
5	RICERCA OGGETTI E VERIFICHE	35
5.1	Introduzione	35
5.2	Ricerca degli Oggetti	36
5.3	Verifica della possibilità di presa	37
5.3.1	Rotazione irregolare del flexibowl	38
5.4	Raccolta di informazioni in un database	41
6	LA PRESA DI OGGETTI IN MOVIMENTO	43
6.1	Introduzione	43
6.2	Tecnica presa al volo	43
6.2.1	Dove e come avviene la presa	44
6.2.2	Metodo per svolgere la traiettoria circolare	44
6.3	spazio di lavoro del manipolatore	46

6.4	Inseguimento in feedforward	50
	Conclusioni	53
	APPENDIX	55
A		57
	BIBLIOGRAFIA	65

ELENCO DELLE FIGURE

Figura 1	Setup di Sperimentazione.	1
Figura 2	Nastro Trasportatore Circolare.	3
Figura 3	Contenuti e Relazioni di un Progetto.	7
Figura 4	Vision Project Interface.	8
Figura 5	Vision Project Toolbar.	9
Figura 6	System Devices Toolbar.	10
Figura 7	Sequence Editor.	11
Figura 8	Sequence Editor Toolbar.	11
Figura 9	Display Toolbar.	12
Figura 10	Results Tool Locator	13
Figura 11	Acquire Image Tool Interface.	16
Figura 12	Icons Acquire Image.	17
Figura 13	Acquire Image Tool Interface.	17
Figura 14	Locator Toll Interface.	19
Figura 15	Model Edition.	21
Figura 16	Gestione Modelli.	23
Figura 17	Range di Rotazione.	24
Figura 18	Location Proprieties.	26
Figura 19	Zone di Lavoro sul Nastro Circolare.	27
Figura 20	Schema Funzionamento Programma Principale.	28
Figura 21	Spazio di Lavoro.	30
Figura 22	Sistemi di Riferimento.	31
Figura 23	Sistema di Riferimento Robot e Telecamera dei due Oggetti.	32
Figura 24	Metodo per Trovare il Centro del Flexibowl.	33
Figura 25	Schema Funzionamento Ricerca Oggetti.	35
Figura 26	Schema Funzionamento Ricerca Oggetti.	36
Figura 27	Informazioni Relative all'Oggetto.	37
Figura 28	Limiti.	38
Figura 29	Rotazione Irregolare.	39
Figura 30	Spazio di Lavoro Telecamera Modificato.	39
Figura 31	Stesso Oggetto.	40
Figura 32	Schema di Funzionamento Presa Oggetti.	43
Figura 33	Schema Funzionamento Presa.	44
Figura 34	Parametri Oggetto allo Scatto della Foto.	45
Figura 35	Parametri Oggetto dopo che il Nastro si è spostato.	46
Figura 36	Traiettoria di Presa.	47
Figura 37	Traiettoria Circolare di Presa.	47
Figura 38	Problema Limite Superiore Manipolatore.	48

Figura 39	Limite Statico.	49
Figura 40	Limite Dinamico.	50
Figura 41	Traiettoria di Presa errata.	51
Figura 42	Inseguimento FeedForward.	52

ELENCO DELLE TABELLE

Tabella 1	Esempio di un Registro	42
-----------	------------------------	----

INTRODUZIONE



Figura 1: Setup di Sperimentazione.

Oggi giorno le industrie sono alla ricerca di soluzioni lavorative sempre più efficaci in modo da ottimizzare tempo e costi di produzione. Le operazioni di pick & place svolte su nastri trasportatori lineari sono già implementate a livello industriale mentre per quanto riguarda quelle su nastri circolari non ancora. Il progetto in considerazione prevede la presa di oggetti da un nastro trasportatore circolare denominato flexibowl. A livello industriale il suo utilizzo è però scarsamente sfruttato nel senso che per prelevare oggetti sul suo piatto il sistema deve essere fermato per permettere al manipolatore di operare, questo ci fa intuire uno spreco notevole di tempo che si traduce direttamente in costi di produzione maggiori. Ecco che la soluzione adatta a questo problema è di considerare la presa al volo degli oggetti, una tecnica innovativa con una serie di vantaggi di notevole interesse. Questo metodo permette di svolgere le operazioni in modo continuo, senza interruzioni delle parti coinvolte, con consistenti pro-

fitti proprio in termini di tempo e costi. Tutto ciò è stato realizzato usando semplicemente un flexibowl, un robot scara e una telecamera. Il compito della telecamera è veramente importante perché permette di dialogare fra le altre due parti. Il manipolatore riceve le informazioni sulla posizione degli oggetti dalla elaborazione della foto scattata. Le informazioni sono ricavate sfruttando la combinazione di due software AdeptSight che gestisce la visione e AdeptDesktop che gestisce la movimentazione del robot e del flexibowl. Inoltre, è di notevole importanza conoscere la posizione angolare di avanzamento del nastro che ci è fornita dall'encoder montato su di esso. In questo modo so con precisione dove si trova l'oggetto. Quindi il lavoro svolto ci ha permesso di creare un sistema di produzione che lavora senza soste e in modo autonomo. Per fare sì che il sistema funzioni con estrema precisione devo definire dei campi di lavoro per la telecamera, il robot e inserimento oggetti. Grazie a questa netta separazione delle zone di lavoro il sistema lavora senza che le parti si ostacolino. La zona di lavoro è fondamentale perché riesco così a svolgere contemporaneamente le operazioni di visione e presa di oggetti con il vantaggio di riuscire sempre a prelevare pezzi e non di aspettare il giro successivo. Passando ora a presentare la struttura della tesi, essa si articola in cinque capitoli legati tra loro, che seguono l'ordine temporale e intuitivo delle azioni svolte per raggiungere gli scopi prefissati

- Capitolo 2 descrive il nastro trasportatore circolare (flexibowl).
- Capitolo 3 descrive il software AdeptSight.
- Capitolo 4 descrive il principio di funzionamento.
- Capitolo 5 descrive il metodo di ricerca degli oggetti e dettagli aggiuntivi.
- Capitolo 6 descrive il metodo di presa degli oggetti e la soluzione ai possibili rallentamenti riscontrati.

IL NASTRO TRASPORTATORE CIRCOLARE

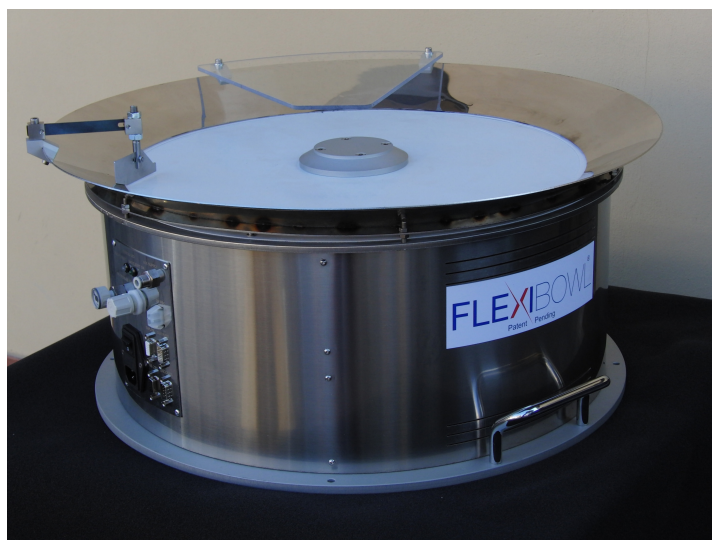


Figura 2: Nastro Trasportatore Circolare.

2.1 INTRODUZIONE

Il flexibowl è un dispositivo ideato e sviluppato per alimentare in maniera flessibile particolari destinati al montaggio industriale abbinabile a un qualsiasi robot e sistema di visione artificiale. Il sistema è costituito sostanzialmente da una tazza rotante dotata di motore direct-drive e da un dispositivo generatore di impulsi perpendicolare al piano di rotazione; il dispositivo permette di separare e variare opportunamente la giacitura dei pezzi in modo tale che questi possano essere visti e riconosciuti da una telecamera e quindi prelevati da un robot dotato di un adeguato organo di presa. Non sono richieste attrezzature dedicate. Il sistema può essere facilmente riprogrammato per produzioni diverse. Velocità, accelerazione e direzione del movimento di rotazione, così come pure la frequenza degli impulsi, possono essere variati da programma in funzione della geometria e dell'attrito superficiale dei pezzi, per determinare il risultato migliore. Il sistema completo comprende alimentatore, robot e sistema di visione.

2.1.1 *Cosa può alimentare*

Il flexibowl è adatto ad alimentare pezzi di dimensioni piccole. In particolare è indicato per oggetti che non si prestano ad essere alimentati con sistemi vibranti tradizionali. Grazie alla sua costruzione e funzionamento, Il flexibowl è in grado di sbrogliare pezzi che altrimenti in sistemi tradizionali si incastrerebbero, causando un fermo di produzione. Inoltre, non è influenzato da eventuali variazioni di dimensione dei pezzi o dalla presenza di bave o residui di lavorazione. E' possibile alimentare anche più di due tipologie di pezzi anche se è sconsigliabile perché alimentando due o più tipologie in un unico flexibowl potrebbero presentarsi problemi di cadenza nella sequenza di montaggio. In altre parole è una soluzione tecnicamente possibile, ma da valutare attentamente di volta in volta. La cadenza di alimentazione varia molto in funzione del tipo di pezzo e della giacitura che si considera ottimale per la presa.

2.1.2 *Come si collega al manipolatore*

Il flexibowl è stato progettato per essere interfacciato a qualunque robot presente sul mercato. Il collegamento avviene indifferentemente via Ethernet o tramite I/O digitali. La differenza fra le due modalità operative sta tutta nella possibilità o meno di impostare i parametri di movimento della tazza. Con il collegamento ethernet (protocollo UDP) i parametri possono essere impostati direttamente dal controller del robot, consentendo quindi la loro variazione in funzione del tipo di pezzi attualmente in produzione. Con il collegamento tramite I/O digitali, i parametri possono essere impostati da un pc ed un programma fornito collegandosi via ethernet, mentre direttamente dal controller del robot via I/O è possibile dare solo i comandi di movimentazione.

2.1.3 *La retroilluminazione e la superficie rotante*

Come fornitura standard viene fornito un backligh ad infrarossi. L'illuminazione ad infrarossi consente di essere più facilmente filtrata, consentendo un maggiore isolamento dalla luce ambiente, che può causare problemi nel riconoscimento dei pezzi. Sono comunque disponibili in alternativa backligh a luce bianca o rossa. Grazie al metodo di fissaggio estremamente semplice, è possibile cambiare in pochi secondi la superficie rotante. Ci sono vari tipi di superfici con diversi colori e rugosità, ma nulla vieta che il cliente possa trovare qualcosa di più adatto alle proprie esigenze e ne ritagli un pezzo per poterlo utilizzare sul flexibowl.

2.1.4 *Come lavora il flexibowl*

I movimenti possibili sono:

1. avanzamento
2. shake, scuotimento avanti ed indietro in rapida successione
3. flip, azionamento del pistone sotto il tappeto per far saltare e girare i pezzi

L'azione combinata dei movimenti avanzamento e flip ci permettono di ottenere un maggiore effetto nel fare saltare i pezzi. Di solito la tazza viene consegnata con l'opzione flip già installata. E' possibile avere anche come alternativa l'opzione "blow". Si tratta di un soffio d'aria esterno posizionabile a piacere dall'utente. Viene utilizzato per separare ulteriormente i pezzi e per ridirigerli verso il centro del nastro circolare, contrastando così l'azione della forza centrifuga. Il flip o il blow possono essere azionati indipendentemente dal comando di avanzamento, o durante lo stesso.

2.1.5 *Come interfacciarla*

L'interfacciamento con il manipolatore avviene tramite l'uso del software AdeptDesktop nel quale è fornita una libreria in codice "V+" per poter utilizzare il flexibowl. Viene inoltre fornito un applicativo per pc per poter variare i parametri di movimento e l'indirizzo IP. Il dispositivo prevede la possibilità di connettere un'encoder, quindi è possibile il prelievo di parti in movimento continuo, in altre parole un inseguimento circolare (circular conveyor tracking), permettendo una più elevata cadenza produttiva. In questo modo si evita di arrestarlo ad ogni avanzamento.

2.1.6 *Vantaggi*

Fra i vantaggi più importanti offerti da questa soluzione si segnala:

- elevata efficienza e autonomia di funzionamento
- nessun degrado di prestazione nel tempo
- possibilità di riattrezzare facilmente il sistema e di recuperare la maggior parte dell'investimento in caso di cambi di produzione
- beneficia di una costruzione semplice, affidabile e di una bassa rumorosità
- grazie alla forma rotonda il ricircolo dei pezzi diventa automatico

ADEPTSIGHT: SOFTWARE DI VISIONE

3.1 INTRODUZIONE

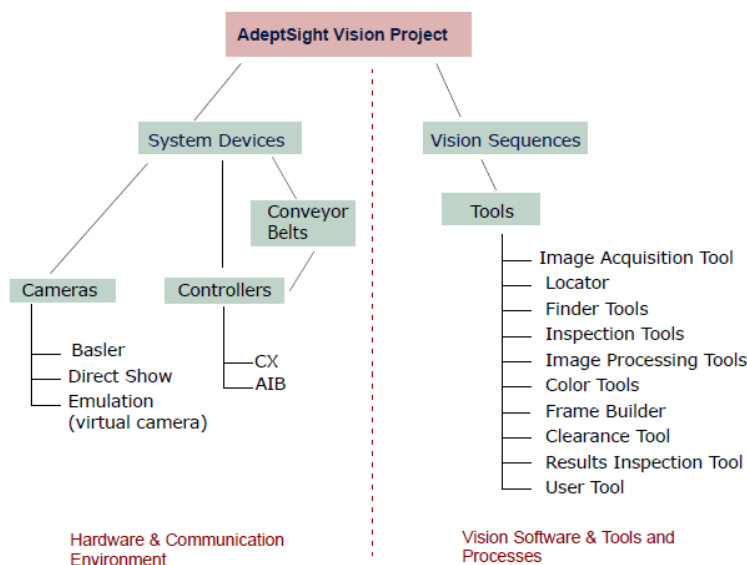


Figura 3: Contenuti e Relazioni di un Progetto.

AdeptSight è un programma che permette la gestione della telecamera ovvero la gestione della visione tramite PC ed è incorporato nell'ambiente di sviluppo di Adept DeskTop. Il software di visione fornisce una gamma di strumenti ad alta prestazione, integrate a sistemi robotici Adept, e garantisce precisione, robustezza e velocità nella capacità di ricerca e d'ispezione. Inoltre permette interazioni con i robot e controller dal PC, lo sviluppo di applicazioni robotiche in V+, procedure guidate di calibrazione e un set completo di strumenti di controllo e di elaborazione delle immagini. In AdeptSight, il PC agisce come server visione. Utilizzando le immagini da una o più telecamere, che sono collegate al PC, il software esegue un'approfondita ricerca, ispezione ed elaborazione delle immagini. I risultati di visione, come le posizioni degli oggetti, sono inviati al controllore. Ogni progetto Visione, che è il nome specifico di un'applicazione di AdeptSight, contiene la configurazione degli strumenti di visione e la configurazione dei dispositivi connessi all'applicazione di visione. Attraverso il progetto visione è anche possibile calibrare le telecamere, e calibrare le telecamere degli altri dispositivi utilizzati dall'applica-

zione visione cioè robot, controller e nastri trasportatori. La figura 3 illustra i contenuti e le relazioni di un progetto in AdeptSight Vision.

3.2 IL PROGETTO VISIONE

Le applicazioni di visione sono fatte e configurate attraverso la finestra di dialogo del progetto Vision, chiamata anche Vision Project manager, vedi figura 4. Un progetto Vision è costituito da una o più

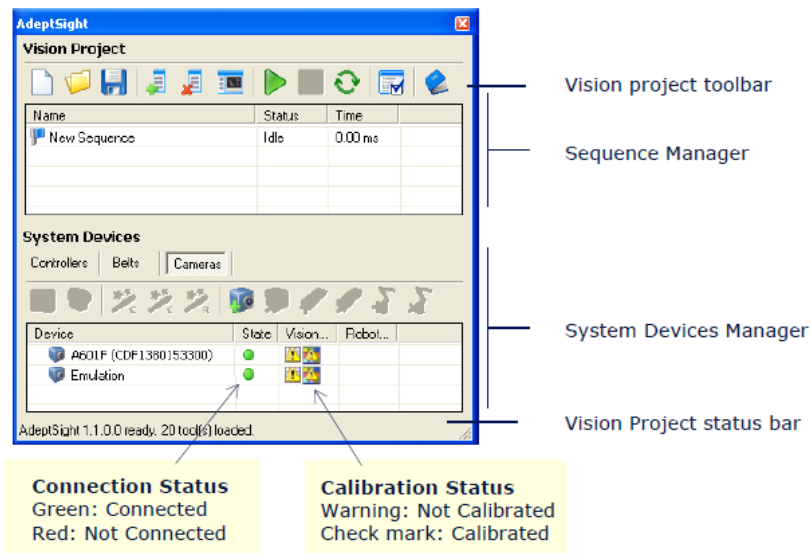


Figura 4: Vision Project Interface.

sequenze di visione e dai dati di configurazione per i dispositivi utilizzati dall'applicazione quindi si tratta di una o più sequenze di strumenti con i relativi dati di configurazione delle periferiche utilizzate dal sistema visione. Questa interfaccia è divisa in due sezioni:

- Il Sequence Manager consente di gestire e modificare le sequenze di strumenti che si trovano nel progetto di visione le quali sono create e modificate nel Sequence Editor.
- Il System Devices Manager consente di impostare e configurare i dispositivi necessari per il progetto visione.

Le funzioni disponibili nella barra degli strumenti superiore sono legati alla gestione del progetto e delle sue sequenze, vediamo qui di seguito quali sono, vedi figura 5.

3.2.1 Sequence Manager

Il Sequence Manager è l'area dell'interfaccia progetto visione che consente di gestire e modificare le sequenze che fanno parte del progetto Vision. Una sequenza è una serie di processi. Quando si esegue una












	Create Project	Starts a new vision project - Clears all the current sequences and all System Devices settings from the Vision Project.
	Load Project	Starts a new vision project - Clears all the current sequences and all System Devices settings from the Vision Project.
	Save Project	Saves all the current sequences and System Devices settings to a Vision Project file (*.hsproj).
	Add Sequence	Adds a new, blank sequence to the Vision Project.
	Remove Sequence	Removes the selected sequence from the Vision Project.
	Edit Sequence	Opens the Sequence Editor for the selected sequence. This enables you to add and configure the vision tools in the sequence.
	Execute Sequence	Runs the vision sequence. Depending on the state of the Continuous Loop icon, Execute Sequence runs a single iteration, or runs the sequence continuously until stopped.
	Stop Sequence	Stops the execution of a sequence that is running.
	Continuous Loop	Run the selected sequence in loop mode (continuous execution), until the execution is stopped by the Stop Sequence icon.
	Project Properties	Opens the Environment Options dialog, for viewing and editing of properties that apply to the AdeptSight environment.
	Help	Opens the AdeptSight online help.

Figura 5: Vision Project Toolbar.

sequenza, ciascuno degli strumenti è eseguito nell'ordine in cui appaiono in essa. Un progetto visione può contenere una o più sequenze, che sono gestite ed eseguite dalla sezione Sequence Manager dell'interfaccia progetto visione. Le sequenze sono costruite e configurate attraverso l'interfaccia Sequence Editor vedi paragrafo 3.3.

3.2.2 System Devices Manager

Il System Devices Manager (gestore delle periferiche) è l'area dell'interfaccia progetto visione che consente di gestire e modificare i dispositivi connessi all'applicazione. Un dispositivo di sistema è un qualsiasi dispositivo come fotocamera, controllore robot o un nastro trasportatore utilizzato da AdeptSight per compiere le operazioni definite da un progetto visione. Il tipo e il numero di dispositivi che possono essere aggiunti a un progetto visione può essere limitato dal tipo di licenza AdeptSight. L'interfaccia mette a disposizione tre schede per diversi tipi di dispositivi di sistema: macchina fotografica, nastri trasportatori e controller. Attraverso il gestore Periferiche di sistema è possibile: aggiungere dispositivi, assegnare i controller e nastri trasportatori a una macchina fotografica e avviare procedure guidate di calibrazione per i dispositivi. In ogni caso l'unico dispositivo assolutamente necessario per un progetto visione è una telecamera, nel caso non fosse disponibile, è possibile utilizzare il dispositivo di emulazione per simulare ingresso della telecamera. Vediamo ora in figura 6 le icone per i dispositivi del sistema di gestione e calibrazione

del gestore delle periferiche.



















	Edit Camera Properties	Opens the properties window for the selected camera. If the selected camera is an emulation device, this icon opens the Emulation Properties dialog.
	Live Display	Displays the live images provided by the selected camera. This is useful for visualizing the effect of camera settings, such as brightness, focus, aperture, white balance, etc.
	Add Camera	Opens the Add a Camera dialog, allowing you to select and name the camera to add.
	Remove Camera	Removes the selected camera from the vision project.
	Add Belt	In the Belts tab: Adds a belt to the list. In the Cameras tab: Opens the Select a Belt dialog, allowing you to select and assign a conveyor belt to a camera.
	Remove Belt	Removes the selected camera from the vision project.
	Add Robot	Opens the Select a Robot dialog, allowing you to select and assign a robot to a camera.
	Remove Robot	Removes the selected robot from the camera.
	Add Controller	Opens a dialog that allows you to select and add a controller to the devices list or to assign a controller to a selected conveyor belt.
	Remove Controller	Removes the selected controller from the vision project.
	Calibrate Camera	Launches the 2D Vision Calibration Wizard, to calibrate the camera for perspective and lens distortion.
	Calibrate Color Camera	Launches the Color Calibration Wizard, to calibrate the colors rendered by a color camera.
	Calibrate Vision-to-Robot	Launches the Vision to Robot Calibration Wizard that enables you to calibrate the setup of devices for your camera Calibration: cameras, robots, controllers and conveyor belts.
	"Not Calibrated" status icon	The device is not calibrated.
	"Calibrated" status icon	The device is calibrated.
	"Color Not Calibrated" status icon	The camera color is not calibrated.
	"Color Calibrated" status icon	The camera color is calibrated for color.
	Connection state icons for system devices	Green: The device is connected Red: The device is not connected.

Figura 6: System Devices Toolbar.

3.3 SEQUENCE EDITOR

Come abbiamo già detto una sequenza è una serie di processi che sono eseguiti da diversi strumenti i quali sono aggiunti e configurati nel Sequence Editor. L'interfaccia di questo editor è rappresentata dalla figura 7. Gli strumenti per la sequenza sono aggiunti, gestiti e configurati nel settore Process Manager. L'ordine degli strumenti è molto

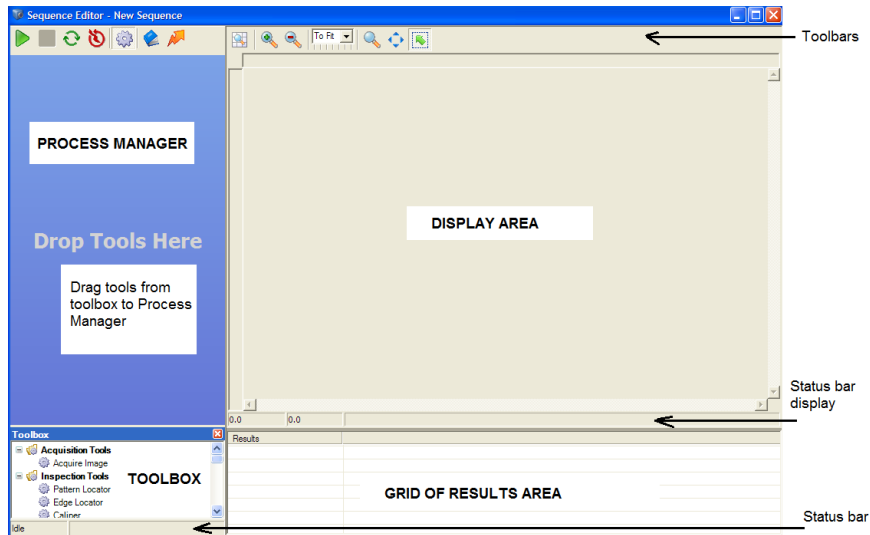


Figura 7: Sequence Editor.

importante perché una volta eseguita la sequenza gli strumenti sono lanciati in ordine di creazione. E' anche possibile eseguire un utensile singolarmente, ciò è utile per testare la configurazione di uno strumento, è possibile eseguire ripetutamente l'utensile sulla stessa immagine per visualizzare l'effetto di una variazione dei parametri. Il primo strumento che ci dovrebbe essere è uno di acquisizione d'immagine cioè l'Acquire Image tool, che fornisce le immagini d'input per tutti i successivi strumenti di visione. Inoltre ogni strumento riceve input solo da quello che lo precede e mai da quello che lo segue. In figura 8 vediamo gli strumenti del Sequence Editor. Tuttavia non








	Execute Sequence	Runs the vision sequence. Depending on the state of the Continuous Loop icon, Execute Sequence runs a single iteration, or runs the sequence continuously until stopped.
	Stop Sequence	Stops the execution of the vision sequence.
	Continuous Loop	Run the vision sequence in loop mode (continuous execution), until the execution is stopped by the Stop Sequence or Reset Sequence icon.
	Reset Sequence	Stops the execution of the vision sequence and resets the Overlap Tool, the Communication Tool, and the Acquire Image tool.
	Toolbox	Shows/Hides the toolbox. Tools can be added to the sequence by dragging them from the toolbox to the Process Manager area. Tools can also be added from the context menu that is displayed by right-click in the Process Manager area.
	Help	Opens the AdeptSight online help.
	Collapse All	Collapses all tool interfaces in the Sequence Editor.

Figura 8: Sequence Editor Toolbar.

è sempre così perché ci sono degli strumenti che non elaborano le

immagini ma che richiedono come input una struttura (frame) necessaria per posizionare l'utensile rispetto ad un sistema di riferimento. Il display del Sequence Editor è uno strumento multifunzione che consente di visualizzare le immagini fornite dallo strumento Acquire Image, l'ingresso in diretta dalla fotocamera, come immagini continue o come una singola immagine statica, e una rappresentazione visiva dei risultati di uno strumento. Fornisce anche un'interfaccia interattiva di modifica o configurazione di alcuni strumenti, consente di posizionare all'utente la regione d'interesse di alcuni strumenti e anche un'interfaccia interattiva per la creazione e modifica di modelli. In figura 9 gli strumenti del display. La barra di stato del display ci

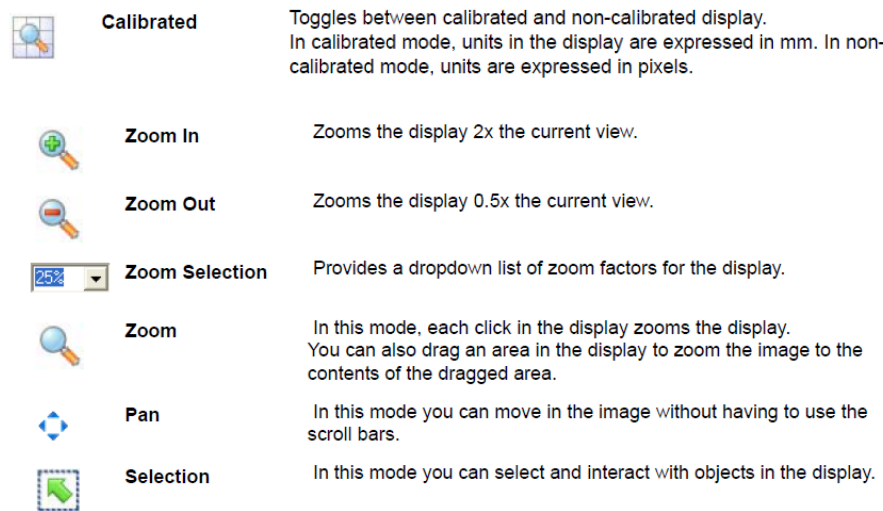


Figura 9: Display Toolbar.

da informazioni della posizione del cursore (mouse) nell'immagine. Le informazioni fornite sono le coordinate XY in due possibili unità di misura, millimetri se la fotocamera è calibrata altrimenti in pixel e il valore del livello di grigio in corrispondenza della posizione corrente del cursore. La griglia dei risultati, sotto il display, mostra i valori dei risultati dello strumento specifico, ad esempio in figura 10 quello del Locator, solitamente come valori numerici e possono anche essere salvati in un file di testo chiamato il results log. Di utile interesse è capire come sono visualizzati i risultati dello strumento Locator, essi sono riferiti rispetto ad un sistema di coordinate ed esistono quattro diverse modalità:

1. World, sistema di coordinate del mondo. I risultati sono visualizzate rispetto al sistema di coordinate dell'immagine di input. Se la telecamera attuale è stata calibrata con il 2D Vision Calibration Wizard, l'origine (0,0) è al centro dell'immagine invece se è stata calibrata attraverso il Vision to Robot calibration, l'origine coincide con quello Robot.

2. Object, sistema di coordinate oggetto. I risultati, le uscite, sono visualizzati rispetto al modello telaio di riferimento.
3. Image, sistema di coordinate immagine. I risultati sono visualizzati in pixel. Non vengono calcolati i seguenti parametri: scala, rotazione, fattore di qualità e risultati di simmetria. Sono espressi in unità calibrate rispetto al sistema di coordinate oggetto, se e solo se lo strumento è telaio base. Il sistema di coordinate oggetto è il sistema di coordinate del modello.
4. Tool, sistema di coordinate utensile. In questo sistema il riferimento di posizione è espresso rispetto alla regione d'interesse Locator, dove (0,0) è il centro del rettangolo che definisce la regione. I risultati sono restituiti in valori pixel. Vengono calcolati i seguenti parametri: scala, rotazione, qualità e risultati di simmetria.



Figura 10: Results Tool Locator

La barra di stato (Status bar) mostra il tempo di esecuzione dell'intera sequenza o di un unico strumento. Infine c'è l'area Toolbox dove ci sono tutti i strumenti messi a disposizione da AdeptSight all'utente per costruire la sequenza.

3.4 STRUMENTI DI ADEPTSIGHT

AdeptSight fornisce un insieme completo di strumenti di visione base per applicazioni complesse. Gli strumenti sono aggiunti in una disposizione che costituisce una sequenza. Un'applicazione di visione può contenere un qualsiasi numero di sequenze e all'interno di ognuna gli strumenti sono eseguiti in ordine. L'ordine degli strumenti nella sequenza è importante perché l'uscita di un determinato strumento può essere utilizzato come ingresso solo da un altro posto in successione. Vediamo qui di seguito in dettaglio gli strumenti utilizzati da AdeptSight presenti nel Toolbox del Sequence Editor.

3.4.1 *Acquisizioni d'immagini*

Ogni sequenza inizia con uno strumento d'acquisizione delle immagini, che fornisce le figure d'ingresso per altri strumenti di visione. Ci può essere più di uno strumento d'acquisizione d'immagini in sequenza.

- *Acquire Image*, fornisce immagini che sono acquisite da una fotocamera compatibile, o da un database d'immagini fornite da un dispositivo di emulazione. Per le applicazioni di monitoraggio di trasporto, questo strumento fornisce parametri di acquisizione che permettono il "soft" o "hard" di monitoraggio del nastro, e la visione della posizione di aggancio al volo.

3.4.2 *Strumenti di movimento*

Gli strumenti di movimento forniscono il modo di funzionare richiesto per la comunicazione tra l'applicazione di visione e i dispositivi di movimento: controller, robot, nastro trasportatore.

- *Overlap tool*, filtra istanze che sono già state rilevate dallo strumento *Locator*, in modo che il controller del robot non può tentare di prendere/controllare/gestire un oggetto più di una volta.
- *Communication toll*, gestisce ed invia le istanze trovate dallo strumento *Locator* a una coda sul controller.

3.4.3 *Strumenti di cerca e individuazione*

Questi strumenti creano una descrizione vettoriale di oggetti o delle loro caratteristiche, sono più veloci, più affidabili e più precisi rispetto a quelli che utilizzano un controllo su scala di grigi, nella maggior parte delle situazioni.

- *Locator*, trova e individua le istanze degli oggetti definiti nel modello. I modelli sono creati e modificati tramite l'editor del modello *locator*. E' uno strumento ideale per il controllo del posizionamento degli oggetti.
- *Arc Finder*, trova individua le caratteristiche circolari sugli oggetti e restituisce le coordinate del centro dell'arco e il raggio.
- *Line Finder*, trova e individua le caratteristiche lineari sugli oggetti e restituisce l'angolo della linea e le coordinate del punto.
- *Point Finder*, trova e individua le caratteristiche del punto sugli oggetti e restituisce l'angolo così come le coordinate del punto trovato.

3.4.4 *Strumenti di colore*

- Color Matching Tool, filtra e analizza le aree di colore specificato, o intervalli di colore nelle immagini RGB.

3.4.5 *Strumenti di elaborazione delle immagini*

Questi strumenti forniscono varie operazioni e funzioni per l'analisi e l'elaborazione delle immagini.

- Image Processing Tool, elabora le immagini in scala di grigi attraverso applicazioni aritmetiche, di assegnazione, logiche, di filtraggio, operatori morfologici o istogramma. Gli utenti possono definire gli operatori di filtraggio personalizzati.
- Image Sharpness Tool, lo strumento di nitidezza dell'immagine, calcola la nitidezza dei bordi preponderanti in una regione definita dall'utente d'interesse.
- Image Histogram tool, calcola statistiche a livello di grigio all'interno di una regione definita dall'utente d'interesse.
- Sampling tool, è usato per estrarre un'area da un'immagine e l'output è rappresentato da un'immagine separata.

3.4.6 *Strumenti d'ispezione*

Gli strumenti d'ispezione/controllo sono comunemente usati in applicazioni di visione per ispezionare gli oggetti e le parti, solitamente trovate da uno strumento Locator. Si basano sull'analisi delle informazioni in pixel, e non creano descrizioni vettoriali degli oggetti, come fanno gli strumenti di cerca e individuazione.

- Blob Analyzer, trova e individua "macchie" e restituisce risultati diversi per ogni tipo di "macchia" trovata.
- Capiler, trova e individua una o più coppie di bordo e misura le distanze tra i due bordi all'interno di ogni coppia.
- Arc Capiler, trova e individua una o più coppie di bordo su una superficie a forma di arco o circolare e misura le distanze tra i due bordi all'interno di ogni coppia.
- Edge Locator, trova e individua un bordo o un'insieme di archi che soddisfano i criteri definiti dall'utente.
- Arc Edge Locator, trova e individua un bordo o un'insieme di bordi in una zona a forma di arco o circolare.
- Pattern Locator, trova e individua le istanze di un modello in scala di grigi.

3.4.7 Altri strumenti

- Results Inspection Tool, filtra i risultati, da altri strumenti, che soddisfano criteri specifici. Operatori logici AND e OR sono applicati a un insieme di condizioni che si applicano ai risultati di altri strumenti in una sequenza visione.
- Frame Builder Tool, permette all'utente di creare sistemi di riferimento personalizzati che possono essere utilizzati in applicazioni di visione AdeptSight.

3.5 STRUMENTI USATI PER REALIZZARE IL PROGETTO DI TESI

Analizziamo ora gli strumenti utilizzati nel progetto per chiarire in dettaglio come funzionano. Sostanzialmente sono due e sono l'acquire image tool e il locator tool. Il loro utilizzo è fondamentale per lo scopo che ci siamo prefissati; il primo strumento raccoglie le informazioni scattando una foto sul piatto girevole del flexibowl nel quale grazie al secondo strumento riusciamo ad analizzare gli oggetti presenti ed averne un elenco preciso in base alla forma stessa del pezzo.

3.5.1 Strumento Acquire Image

Questo strumento acquisisce immagini da una fotocamera o da un database, attraverso il sistema di emulazione. Per il nostro scopo deve essere il primo strumento perché ci fornisce le immagini su cui operare con il successivo. In figura 11 vediamo l'interfaccia proposta da AdeptSight. L'elenco a discesa fotocamera consente di seleziona-

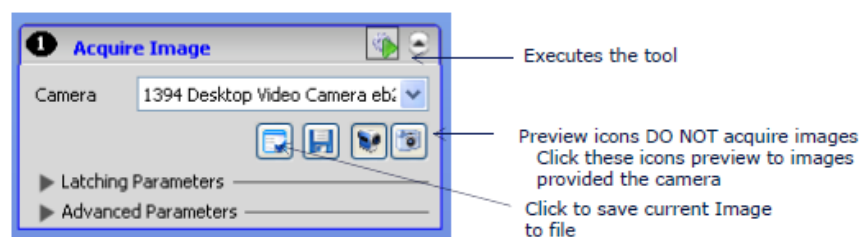


Figura 11: Acquire Image Tool Interface.

re la telecamera in grado di fornire le immagini per la sequenza. La fotocamera Basler viene fornita con AdeptSight ed è identificata dal nome del modello e dall'id, comunque si può utilizzare anche altre di tipo compatibile oppure il dispositivo di emulazione. Le immagini fornite dallo strumento sono visualizzate nel display del Sequence Editor come singola o in modalità live. Le funzioni disponibili delle icone dello strumento sono visualizzate in figura 12. I parametri di aggancio (Lanching Parameters) forniscono opzioni di aggancio fra il







	Save Current Image	Saves the current image to specified file and image format. Various standard formats are available. An additional format: <i>hig</i> , allows you to store the calibration format in the image, for reuse in other AdeptSight applications.
	Camera Properties	Opens the properties window for the selected camera. If the selected camera is an emulation device, this opens the Emulation Properties dialog.
	Live Mode	Provides a continuous display of live images being taken by the camera. This is useful for visualizing the effect of camera settings, such as brightness, focus, aperture, white balance, etc.
	Image Preview	Provides a single image preview of an image taken by the camera. A new image is displayed each time the icon is clicked.
	Execute Tool	Executes the Acquire Image tool, and makes images available for other tools in the sequence. Does not execute other tools in the sequence.
	Grab Failed	Indicates that tool execution failed. Often, this occurs because the selected camera is unavailable, either because it is disconnected, non-existent, or incorrectly configured.

Figura 12: Icons Acquire Image.

robot e le posizioni dei nastri trasportatori. Esistono due tipi di aggancio “Hard Latching” e “Soft Latching”. Il primo si riferisce a una configurazione in cui un cavo trasmette segnali analogici dalla fotocamera al controller. Ogni volta che viene acquisita un’immagine, viene inviato un segnale al controllore. Il quale legge la posizione di un dispositivo al momento dello scatto della foto, e salva questa informazione in un buffer. Invece per quanto riguarda il secondo si riferisce a una configurazione in cui non c’è nessun vero aggancio della posizione. Nel momento in cui è acquisita l’immagine, si chiede al controller per la posizione del dispositivo (robot o nastro trasportatore). Perché c’è un ritardo tra il momento di acquisizione dell’immagine e il tempo che la posizione del dispositivo venga letta. E’ d’obbligo che il sistema (robot o nastro) non sia in movimento durante l’esecuzione dello strumento d’acquisizione immagine. I “Belt Latching Paramete-

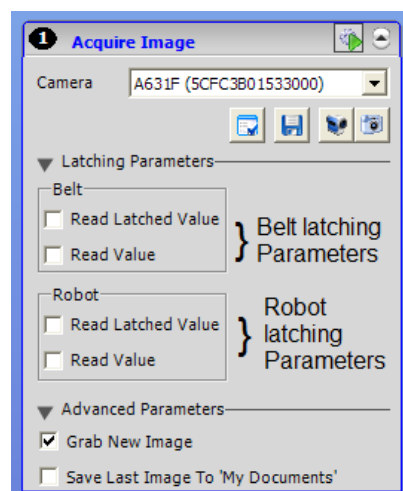


Figura 13: Acquire Image Tool Interface.

res“ definiscono la modalità che verrà utilizzata per bloccare i segnali encoder di applicazioni monitoraggio cinghia. L'impostazione predefinita è scegliere “Read Latched Value“ che è la modalità consigliata per la maggior parte delle situazioni, come visibile in figura 13, che è l'equivalente del hard Latching. L'altra opzione, “Read Value“, si comporta come il soft lanching. Questa ha il vantaggio di utilizzare nessun cavo per l'aggancio del segnale però può essere utilizzata solo nei casi in cui il nastro trasportatore sia fermo completamente quando la fotocamera sta acquisendo oppure quando il movimento del nastro trasportatore è molto lento e un leggero errore di posizione non è critico per l'applicazione. I “Robot Latching Parameters“ definiscono la modalità che verrà utilizzata per bloccare la posizione del robot. L'impostazione predefinita è scegliere “Read Latched Value“ che è la modalità consigliata per la maggior parte delle situazioni, come visibile in figura 13, perché permette il blocco del robot nel momento dello scatto della foto. Non posso utilizzare questa opzione se la telecamera è montata sul braccio o sull'utensile e quando è rivolta verso l'alto e le parti in movimento attraversano il campo di visione. L'opzione “Read Value“ è valida solo se il robot è fermo o in movimento molto lentamente quando la foto viene scattata; si comporta come il soft latching. Il vantaggio di utilizzare questa modalità è che non vi è nessun cavo richiesto per l'aggancio del segnale però può essere utilizzato solo nei casi in cui il robot sia fermo quando la telecamera sta acquisendo un'immagine o quando il movimento del robot è molto lento e un leggero errore di posizione non è critico per l'applicazione.

3.5.2 *Strumento Locator*

IL locator è un potente strumento d'ispezione nel senso che ha il compito principale di trovare gli oggetti presenti nell'immagine acquisita da una telecamera basandosi su modelli. Quest'ultimi descrivono la geometria con estrema precisione degli oggetti da catturare nella foto. Caratterizzato da una ottima velocità, precisione e robustezza. Offre anche elaborazioni basate su colori sui sistemi che dispongono di una licenza AdeptSight colore. Vediamo ora quali sono le operazioni fondamentali nella configurazione di uno strumento Locator:

1. Selezionare lo strumento in grado di fornire immagini in ingresso al Locator (Acquire Image Tool).
2. Posizionare lo strumento Locator.
3. Creare i modelli che verranno utilizzati dal Locator per trovare e individuare gli oggetti.
4. Configurare i parametri di ricerca, in base alle esigenze.
5. Eseguire lo strumento e verificare i risultati.

6. Configurare i parametri avanzati, in base alle esigenze.

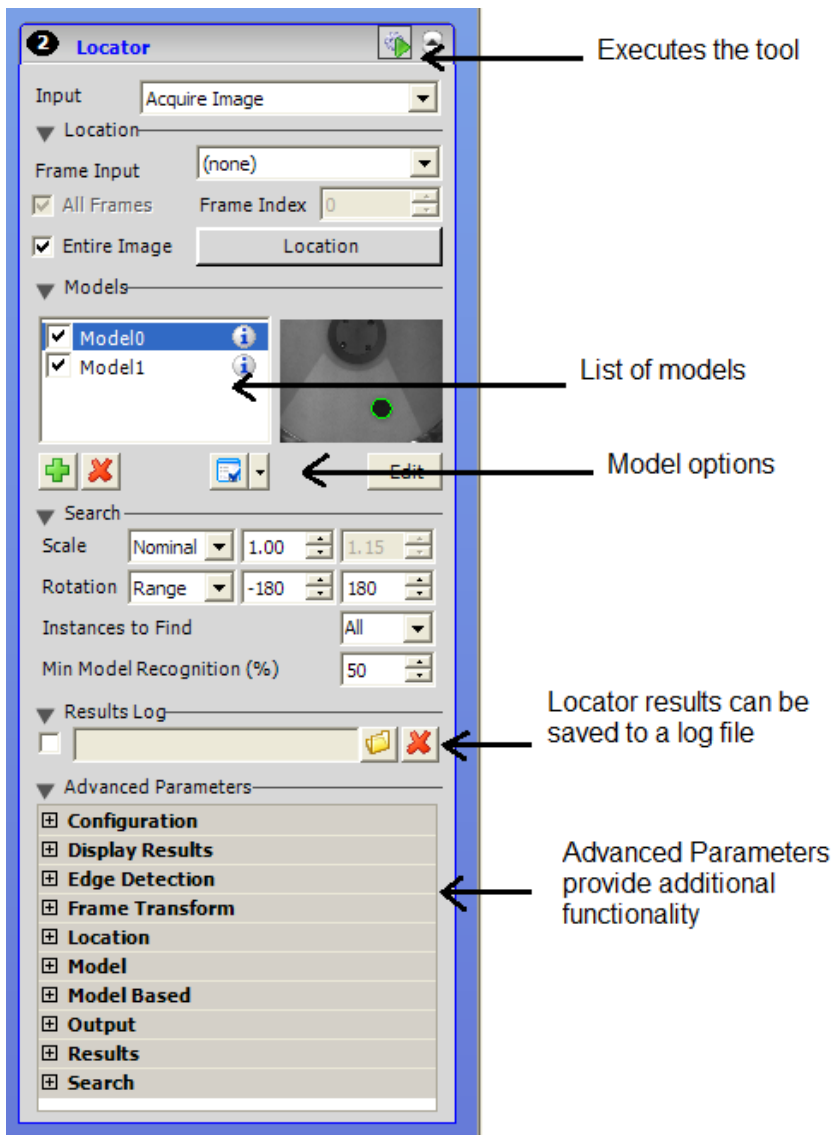


Figura 14: Locator Tool Interface.

L'interfaccia dello strumento locator, vedi figura 14, è suddivisa in sei sezioni:

- Input, seleziona lo strumento che fornisce immagini in ingresso.
- Location, posiziona la regione dello strumento di interesse per il processo di ricerca.
- Models, gestisce i modelli e offre una modalità Model Edition per la creazione e modifica di essi.
- Search, consente di impostare i parametri di base utilizzati dal Locator.

- Result Log, consente il salvataggio dei risultati dello strumento in un file di log.
- Advanced Parameters, fornisce parametri avanzati per la configurazione del Locator.

Analizziamo qui di seguito, in dettaglio, le varie sezioni dell'interfaccia dello strumento Locator. Da esso possiamo ricavare le informazioni utili per prelevare dal nastro trasportatore gli oggetti in movimento.

3.5.2.1 *Input*

Il Locator richiede l'immissione di un'immagine, in genere fornita dallo strumento Acquisisci immagine. Ci sono anche dei casi in cui le immagini di input sono fornite da altri strumenti di AdeptSight che hanno come loro uscita dei risultati di immagini, un esempio è l'Image Processing Tool. Il locator per funzionare si basa proprio sull'ispezione di ciò che la telecamera cattura e quindi è di notevole interesse avere come ingresso direttamente la visione della camera.

3.5.2.2 *Location*

I parametri di Location definiscono dove la regione di interesse dello strumento è posizionata. La regione d'interesse è l'area in un'immagine in cui lo strumento effettua il suo processo. Ci sono due tipi di posizionamento per lo strumento:

- Image-based, la regione di interesse è sempre posta sulla stessa area dell'immagine. Spesso, questa zona è l'intera immagine e il tipo di posizionamento è statico. Per scegliere questa opzione nel menu a tendina del Locator impostare il Frame Input su none, se deve operare su tutta l'area dell'immagine selezionare la casella Entire Image invece se si vuole restringere la zona d'interesse in un'area più piccola cliccare su Location. Vedi figura 14.
- Frame-based, lo strumento è posizionato rispetto a "un'istanza risultato frame" fornito da un'altro utensile, chiamato il frame-provider. Questo tipo di posizionamento è dinamico, la posizione dipende dalla posizione del frame-provider. Se nessuna istanza di frame-provider è presente in una immagine, nessuna istanza dello strumento viene applicata. Per scegliere questa opzione nel menu a tendina del Locator impostare il Frame Input su frame-provider. L'ingresso Frame può essere fornito solo da uno strumento che è al di sopra di quello corrente nella sequenza. Di default la casella All Frames è attivata. Se lo strumento corrente deve essere utilizzato solo per un frame specifico, disattivare la casella All Frames e selezionare il frame desiderato,

il valore predefinito è 0. Poi cliccare su Location per settare la regione d'interesse, modifiche dei valori di grandezza dell'area avvengono in modo dinamico sul display. Vedi figura 14.

3.5.2.3 Models

La sezione Models serve allo strumento Locator per trovare e individuare gli oggetti che sono definiti da un modello. Il Locator fornisce una modalità Model Edition in cui l'utente può creare nuovi modelli e modificare quelli esistenti. Per realizzare un modello cliccare sull'icona aggiungi modello nella sezione dedicata, da qui si apre una finestra, il Model Edition, nell'area display vado a selezionare l'oggetto d'interesse racchiudendolo all'interno di un rettangolo verde come si vede in figura 15. A questo punto se si vuole cambiare i parametri

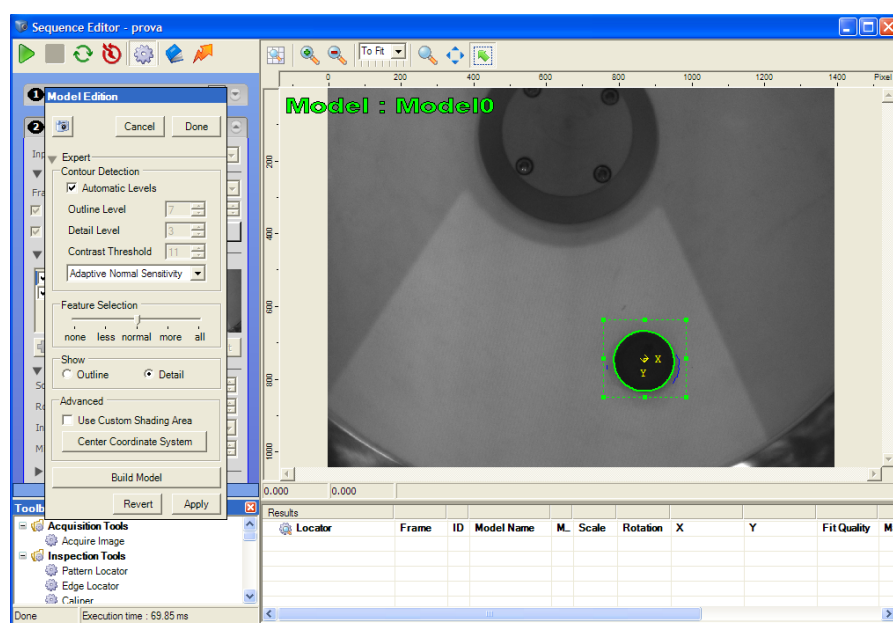


Figura 15: Model Edition.

che influenzano la creazione del modello è necessario accedere alla modalità di correzione Expert. La finestra Model Edition Expert si suddivide in quattro sotto sezioni:

- Contour Detection, di solito i valori predefiniti di rilevamento dei profili sono settati di default. Quando la casella di controllo Automatic Levels è attivata, il sistema ottimizza automaticamente l'Outline Level e il Detail Level. Questa è la modalità consigliata. Per impostare manualmente questi parametri basta disattivare questa casella. L'Outline Level fornisce un livello più grossolano dei contorni con diretta conseguenza che sarà più grossolana anche la risoluzione del profilo. Il Detail Level conferma l'identificazione dell'istanza oggetto e perfeziona la sua posizione all'interno dell'immagine. Minore è l'imposta-

zione (minimo 1) più fine è la risoluzione del profilo. Il valore del Detail Level è sempre inferiore o uguale all'Outline Level. Contrast Threshold imposta la sensibilità al contrasto che viene utilizzato per generare contorni. L'Adaptive Normal Sensitivity è consigliata per la maggior parte delle applicazioni. L'Adaptive High Sensitivity accetta più bassi contrasti. L'Adaptive Low Sensitivity mantiene un contrasto alto in modo da rimuovere quelli inferiori, come quelli causate dal rumore e ombre. Fixed Value richiede di impostare manualmente il valore di soglia del contrasto in step di livello di grigio che varia da 0 a 255.

- Feature Selection, utilizzando il dispositivo di scorrimento si può aumentare o diminuire la quantità di caratteristiche che vengono selezionate dai profili. Se si vogliono aggiungere manualmente posizionare il cursore su none (cioè tutto a sinistra).
- Show, ha due livelli l'Outline che è utilizzato dallo strumento Locator per identificare rapidamente potenziali istanze dell'oggetto. Deve contenere almeno un numero minimo di caratteristiche che consentono al Locator di generare ipotesi affidabili. Invece il Detail è utilizzato dallo strumento Locator per raffinare la posa e la posizione di un oggetto. Dovrebbe contenere caratteristiche non sfocate.
- Advanced, se si attiva la casella Use Custom Shading Area si può definire manualmente un'area che il modello utilizza per l'analisi di sfumature/ombre, se questa non è attivata automaticamente l'analisi è fatta su tutta la regione d'interesse. L'utilizzo di questa opzione ha senso solo se si utilizza una telecamera a colori altrimenti il suo beneficio è assai poco efficiente. Inoltre si può centrare il sistema di riferimento del modello creato cliccando sul bottone Center Coordinate System. Questa cosa è molto utile perché ci fornisce già un'idea dell'orientazione del pezzo.

Facendo clic su Build Model inizia un nuovo processo di creazione del modello. Si cancella il modello che è attualmente in Editor, rileva i contorni di origine utilizzando le nuove impostazioni, selezionare le caratteristiche dei profili di origine in base all'impostazione del cursore di selezione delle funzioni. Questo non apporta alcuna modifica al sistema di coordinate. Infine cliccare su Apply se si vuole salvare le modifiche altrimenti su Revert se si vuole annullarle. In sostanza la gestione dei modelli in AdeptSight è molto facile e comprensibile all'utente. Essi vengono memorizzati in un apposito database e possono essere abilitati o disabilitati in base alle esigenze, questo ci comporta una grande flessibilità nella gestione della funzione cerca e individuazione, in figura 16 è possibile vedere in dettaglio questa sezione.

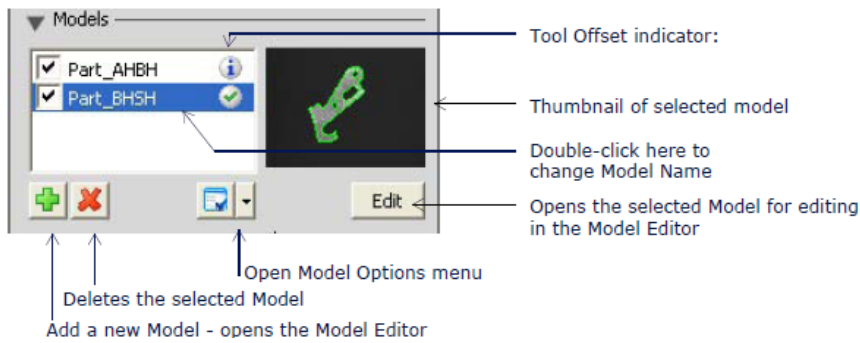


Figura 16: Gestione Modelli.

3.5.2.4 Search

Nella sezione dedicata alla configurazione dei parametri di ricerca si vuole fornire alcuni vincoli per limitare il processo di ricerca Locator. Esso si suddivide in quattro sottosezioni:

- **Scale**, la scala degli oggetti da individuare può essere imposta ad un valore nominale fisso (default), o come un intervallo di valori. Posso scegliere Nominal per la ricerca di un specifico fattore di scala; lo strumento Locator non calcola la scala reale delle istanze e queste vengono posizionate rispetto a quella nominale ed è anche ciò che restituiscono. E' utilizzato nella maggior parte dei casi. Se un valore nominale viene utilizzato con oggetti che presentano una leggera variazione di scala, questi possono eventualmente essere riconosciuti e posizionati con qualità ridotta perché il loro fattore di scala vero non sarà misurato. In tal caso è preferibile configurare un intervallo ristretto di scala, ad esempio $\pm 2\%$, invece del valore nominale. Attenzione: questo comando deve essere utilizzato con cura perché ha il maggiore impatto sulla velocità di ricerca, quindi bisogna includere solo fattori di scala che sono effettivamente previsti per quella data applicazione.
- **Rotation**, può essere impostata come un intervallo di valori orientamento fisso, di default, al valore nominale. L'impostazione predefinita permette una ricerca da -180° a $+180^\circ$ e quindi il Locator cercherà oggetti in tutti i possibili orientamenti. Il campo di rotazione va dall'angolo minimo a quello massimo come illustrato in figura 17.
- **Instances to Find**, permette di impostare il numero massimo di istanze che si possono trovare. Selezionando all il numero di oggetti che si possono cercare è praticamente illimitato. Tuttavia ha senso impostare questo valore per ottimizzare i tempi di ricerca. Se il numero di oggetti è maggiore del valore settato

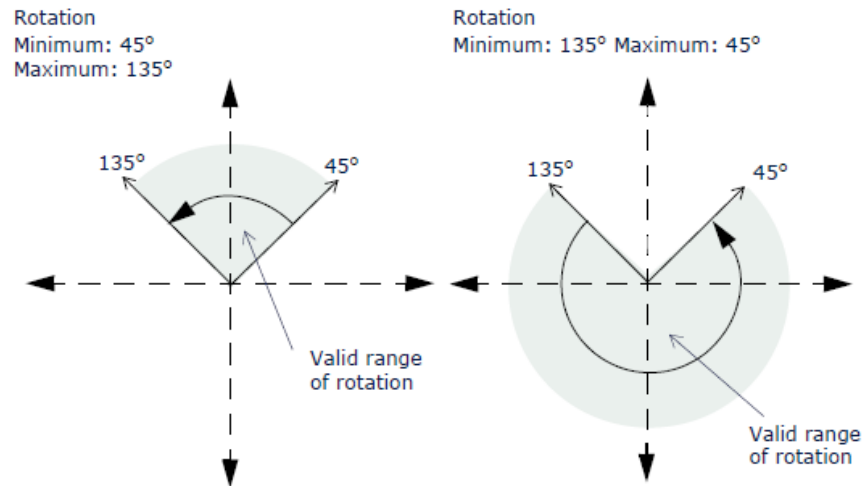


Figura 17: Range di Rotazione.

lo strumento Locator si ferma al raggiungimento di tale limite, cioè non si preoccupa di visualizzarli tutti.

- **Min Model Recognition**, imposta il livello massimo di riconoscimento di un'istanza. Un valore basso di tale parametro fa aumentare il riconoscimento di istanze occluse, ma può anche portare a falsi riconoscimenti. Invece un valore più elevato può contribuire a eliminare i casi in cui gli oggetti si sovrappongono.

3.5.2.5 *Result Log*

Ci permette di salvare in un file log le impostazioni dello strumento Locator, in questo modo posso avere per ogni sequenza più strumenti Locator settati per ogni mia esigenza senza andare a modificare l'unico presente.

3.5.2.6 *Advanced Parameters*

Questa sezione dell'interfaccia consente di accedere ai parametri avanzati e alle proprietà del Locator, vediamo come è articolato:

- **Configuration**, c'è un Processing Format che definisce il formato richiesto per elaborare le immagini fornite dalla telecamera. Quando è attivo hsNative il Locator utilizza il formato nativo per trasmettere le immagini. Quando è attivo il hsGreyScale, il Locator elabora solo le informazioni in scala di grigi dell'immagine in ingresso indipendentemente dal formato in cui sono previste. Questo può ridurre il tempo di esecuzione in cui l'elaborazione del colore non è richiesta.
- **Display Results**, specifica in che modo i risultati del Locator sono rappresentati nell'area del display. Ci sono diversi tipi di

visualizzazione: Display Frame che visualizza ogni istanza con un indicatore di assi XY. Display Instances che mostra i risultati delle istanze nella barra dei risultati del display. Display Grey-Scale Image mostra l'ultimo ingresso in scala di grigi nella barra dei risultati del display. Display Detail Scene mostra tutti i profili a livello di Detail che sono stati trovati nell'immagine in ingresso e vengono visualizzati sul display dei risultati. Display Outline Scene mostra tutti i profili a livello di Outline che sono stati trovati nell'immagine in ingresso e vengono visualizzati sul display dei risultati.

- Edge Detection, Il Locator rileva i bordi nelle immagini d'ingresso e utilizza loro per generare una descrizione vettorizzata dell'immagine, chiamata scena. Vengono generati su due livelli di grossolanità: Outline e Detail. Il Contrast Threshold imposta il contrasto minimo necessario per rilevare un bordo. Il valore di soglia esprime i livelli di grigio necessari per rilevare i contorni. Valori elevati di questo parametro riducono la sensibilità al contrasto e il rumore. Valori bassi invece generano effetti contrari causando anche false rilevazioni e rallenta il processo di ricerca.
- Location, lo strumento Locator svolge i suoi processi all'interno di una regione d'interesse. Per impostazione predefinita, questa regione è l'immagine d'ingresso. Tuttavia può essere ridotta a un'area più piccola, per escludere determinate aree di lavoro o per ridurre il tempo di esecuzione. Qui ho accesso al parametro CalibratedUnitsEnabled che ci restituisce i risultati in millimetri se è impostato su vero altrimenti in pixel. Poi ci sono i parametri altezza, angolo di rotazione, larghezza, coordinata X e coordinata Y per definire la regione d'interesse; in ultima c'è anche un comando UseEntireImage che ha lo scopo di fornire l'intera immagine se è impostato su vero altrimenti ci mostra solo la regione settata dall'utente per svolgere la ricerca. In figura 18 è possibile vedere come sono applicati questi parametri nella regione d'interesse.
- Model, esiste anche la possibilità di individuare con precisione una caratteristica su un oggetto padre, questo è detto Model-Based. Richiede l'input da un altro strumento Locator nell'applicazione. Questo tipo di opzione ci permette di settare già qui gli intervalli di rotazione e la scala. Si può anche intervenire sui parametri di rotazione se si volesse trovare una caratteristica secondaria su un oggetto padre.

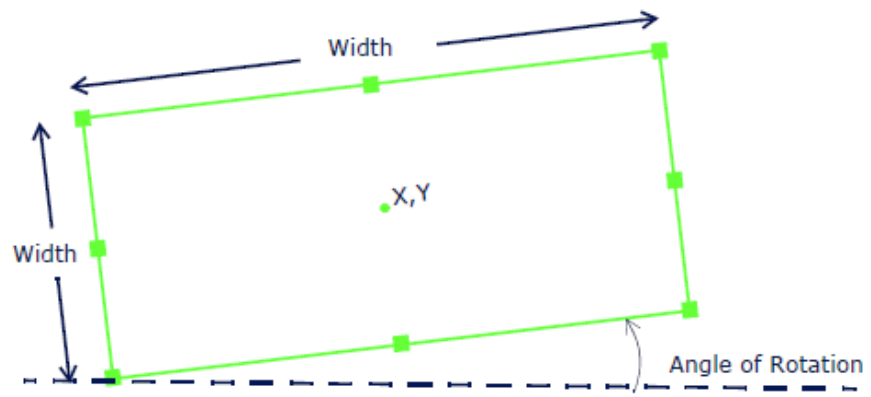


Figura 18: Location Proprieties.

PRINCIPIO DI FUNZIONAMENTO

4.1 INTRODUZIONE

Lo sviluppo di un sistema che agisce in modo continuo e automatico è al giorno d'oggi un'esigenza richiesta. Per far sì che il progetto in questione operi in tale modo si deve focalizzare in dettaglio quali sono le funzioni principali che il sistema deve svolgere. Nel nostro caso sono essenzialmente due:

- la ricerca di oggetti presenti sulla superficie rotante del nastro.
- il metodo di presa di taluni oggetti.

Queste mansioni devono essere gestite separatamente anche se fra loro possono comunicare per poter svolgere il loro compito ininterrottamente senza nessun fermo della macchina. Importante è capire come funziona il meccanismo per la presa degli oggetti, come è possibile vedere in figura 19 il nastro circolare è suddiviso in tre aree che rappresentano ciascuna un determinato compito. Come si può vedere le

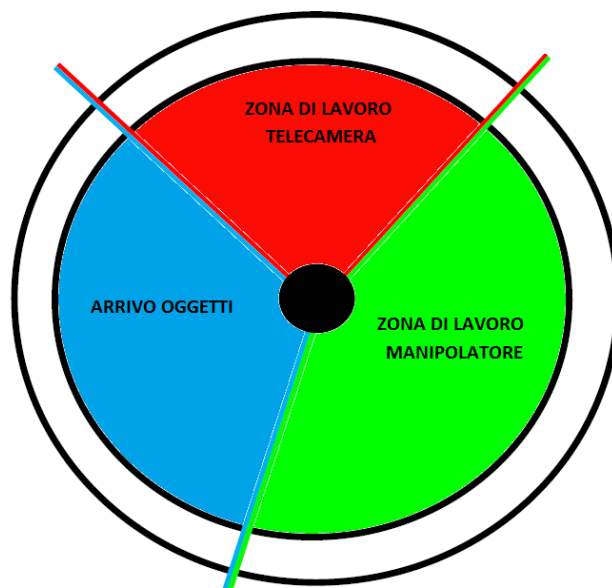


Figura 19: Zone di Lavoro sul Nastro Circolare.

tre zone sono:

- zona di lavoro della telecamera.

- zona di lavoro del manipolatore
- zona di arrivo degli oggetti.

Queste possono avere anche dimensioni diverse e disposte in modo totalmente a piacere dell'utente. Come si può capire in ciascuna area è svolta un'unica operazione e non ci sono aree in comune. Questo garantisce che le parti coinvolte non si ostacolino permettendo così un funzionamento contemporaneo delle stesse. Il funzionamento del sistema si avvale di un programma principale il quale gestisce più programmi in esecuzione. In figura 20 è possibile vedere schematicamente le operazioni svolte. In codice "V+" questo si traduce con un

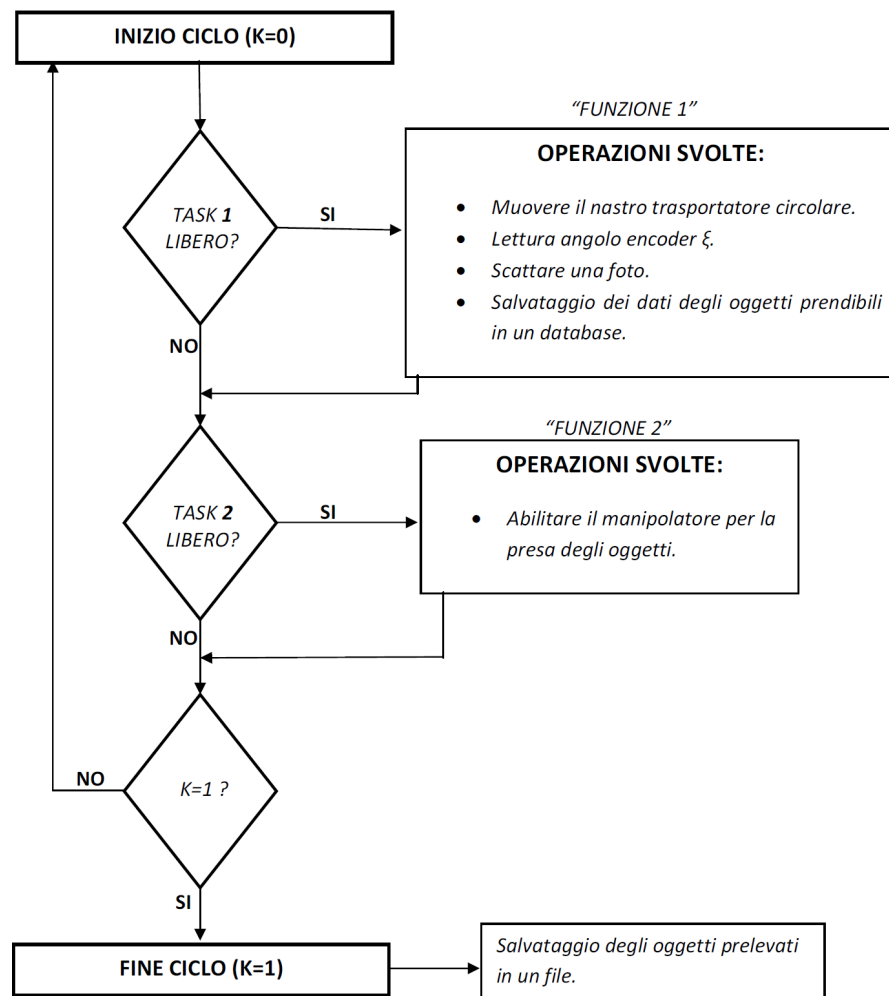


Figura 20: Schema Funzionamento Programma Principale.

ciclo while do infinito che nel nostro caso può essere interrotto solo se si rende falso l'argomento. Rendere non vero l'argomento ci porta ad arrestare la macchina e questo può essere utile se nella tazza non ci sono più pezzi da prelevare. In laboratorio si è ipotizzato il fermo del dispositivo se in un giro completo del flexibowl non ci sono oggetti, tuttavia questa opzione può essere cambiata e anche tolta secondo le

esigenze dell'utente. Dentro al loop troviamo due blocchi di istruzioni che svolgono i compiti descritti all'inizio del capitolo. Queste due funzioni agiscono indipendentemente l'una dall'altra e anche dal programma principale che ha solo il compito di eseguirle se sono libere. In altre parole il programma interroga il task dove è lanciata questa funzione se la risposta è vera svolge le istruzioni all'interno di essa, se è falsa prosegue avanti. Una volta terminato il ciclo esso ri-inizia, si ferma solo se l'argomento è diventato falso. Analizziamo ora in particolare questi due blocchi:

- Il primo blocco ha il compito di eseguire su un'altro task (il numero 1, ricordando che il task 0 è il primo task ed è occupato dal programma principale) il programma test. Questo ha il semplice compito di azionare il nastro trasportatore; al suo interno ci sono tutti i parametri necessari per la configurazione di essa, come la velocità di rotazione, l'accelerazione ecc... Dopo di che andiamo a salvare in una variabile l'angolo encoder che è un dato particolarmente essenziale perché senza di esso non sapremmo mai dov'è fisicamente posto un'oggetto sulla tazza. A questo punto può iniziare la ricerca degli oggetti sulla superficie rotante. Gli oggetti trovati vengono salvati in un database per rendere più facile al manipolatore la loro presa. Queste operazioni vengono eseguite da altri due programmi che sono eseguiti qui tramite una chiamata (call); questa interrompe il programma in esecuzione per eseguire l'altro. Ci sono due chiamate una per cercare e selezionare gli oggetti prendibili e una per salvarli in una matrice secondo un formato più congeniale per la presa. Termina così il primo blocco di istruzioni che come abbiamo appena descritto svolge due compiti che sono fra di loro collegati cioè azionare il flexibowl e contemporaneamente ricercare gli oggetti salvando anche la posizione d'avanzamento del nastro.
- Il secondo blocco abilita il manipolatore per la presa al volo degli oggetti. Ovviamente ha senso eseguire questa operazione in un'altro task, per precisione il numero 2, in modo da poterlo interrogare; se questo non è in esecuzione allora lo abilito.

Una volta usciti dal ciclo infinito può aver senso andare a salvare in un file tutti gli oggetti prelevati dal flexibowl. In questo modo si può fare un'analisi concreta del lavoro svolto. Lo scopo di tale file è di avere una documentazione di ciò che si è preso foto per foto. Da questa è possibile fare degli accorgimenti per cercare una soluzione più ottima in termini di tempi e costi. Essenzialmente si può notare che se in una determinata successione di foto ci sono pochi oggetti o nessuno avrebbe senso velocizzare la rotazione del piatto e quindi nello stesso tempo avrei la possibilità di fare più prelievi.

4.2 SPAZIO DI LAVORO

La telecamera, come per il manipolatore, ha un proprio spazio di lavoro dove svolgere le sue attività sul nastro. La regione è estremamente essenziale perché ci permette di lavorare senza fermi. Quindi riesco a operare in modo continuo senza che la telecamera debba aspettare che il manipolatore li lasci lo spazio per scattare una foto e viceversa cioè aspettare la telecamera che scatti una foto per andare a prendere gli oggetti. Come si vede in figura 21 vengono fissati dei limiti inferiori e superiori che possono essere anche cambiati a piacere dell'utente. Bisogna però ricordarci che il manipolatore scara ha anche un suo spazio di lavoro e quindi la scelta del suo limite superiore all'interno della tazza deve essere sensato. Lo spazio di lavoro è fondamentale per operare senza che si presentino situazioni di intralcio fra le due parti in questo modo garantisco una fluidità nel svolgere le loro mansioni. Inoltre, sempre in figura 21 è possibile vedere lo zero del sistema di riferimento che è stato usato per leggere l'angolo encoder (ξ) e l'angolo di presa dell'oggetto (θ).

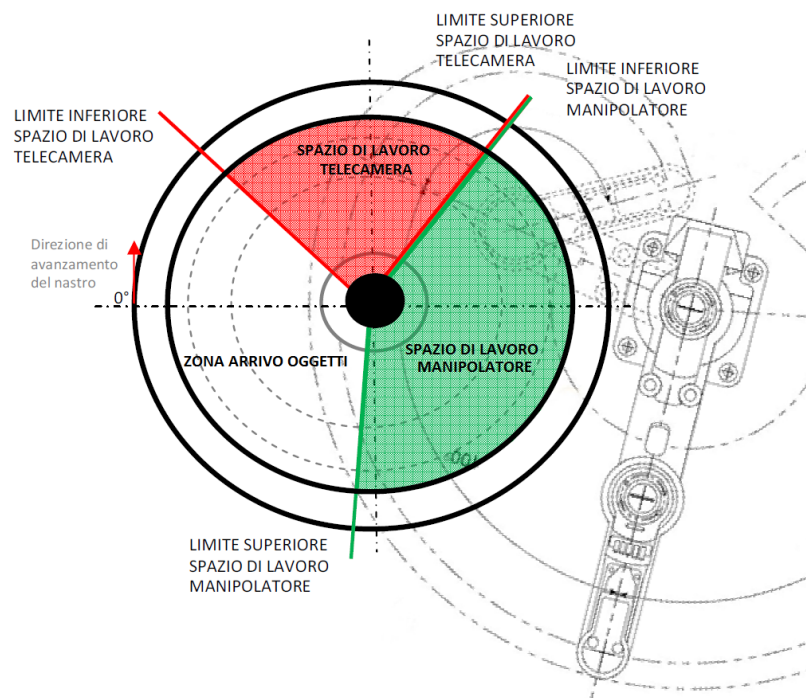


Figura 21: Spazio di Lavoro.

4.3 CALIBRAZIONE DEL SISTEMA

Prima di procedere in dettaglio su come si è realizzato il progetto ha senso concentrarsi un attimo sui sistemi di riferimento adottati. La telecamera, come pure per il manipolatore, lavora secondo il suo siste-

ma di riferimento, vedi figura 22, dove (X_R, Y_R) rappresenta il sistema di riferimento robot, (X_T, Y_T) rappresenta il sistema di riferimento telecamera e (X_C, Y_C) il sistema di riferimento centro flexibowl rispetto a quello della telecamera. La calibrazione serve proprio per unificare il

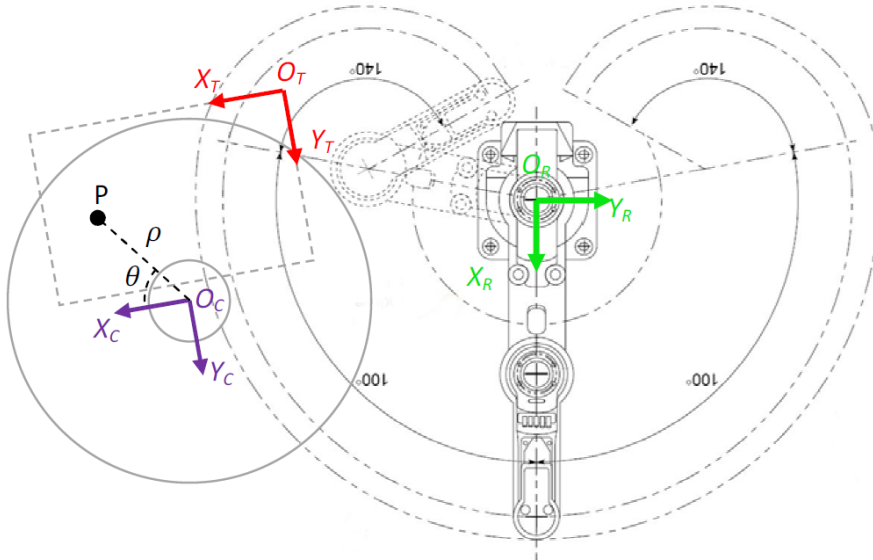


Figura 22: Sistemi di Riferimento.

tutto rispetto al sistema di riferimento robot e inoltre per trovare alcuni dati che ci serviranno per raggiungere lo scopo prefissato. Innanzitutto ho cercato il rapporto di conversione pixel-millimetri e l'angolo tra il sistema robot e telecamera. Le informazioni fornite dal software AdeptSight sono espresse in pixel, si possono comunque avere in millimetri se si fa la calibrazione automatica messa a disposizione da Adept. Per trovare il rapporto di conversione e l'angolo fra i due sistemi di riferimento (β) si è proceduti nel seguente modo: posiziono un oggetto sul nastro trasportatore e ne scatto una foto per avere le informazioni desiderate (coordinata X,Y) dopo di che lo vado a posizionare in un'altro punto del nastro e ne rifaccio la foto, vedi figura 23. Ora vado a calcolare la distanza fra la posizione iniziale e finale dell'oggetto in coordinate robot e quelle date dal software di visione e facendo il rapporto ne trovo il coefficiente di conversione. Dalle informazioni ricevute da AdeptSight mi trovo l'angolo β facendo l'arcotangente del rapporto fra la distanza in Y dei i due punti e quella in X e sommandogli 90° per portarmi nel sistema di riferimento robot. Ora è possibile esprimere anche il sistema di riferimento telecamera rispetto a quello robot. Per la presa dell'oggetto è congeniale usare le coordinate polari riferite al centro del piatto rotante e quindi devo conoscere tale punto. Per trovare il centro si sono sfruttate considerazioni geometriche: il centro di una circonferenza si trova nell'incrocio degli assi di due corde qualsiasi della circonferenza stessa. Per avere due corde abbiamo bisogno di tre punti. Questi li abbiamo ot-

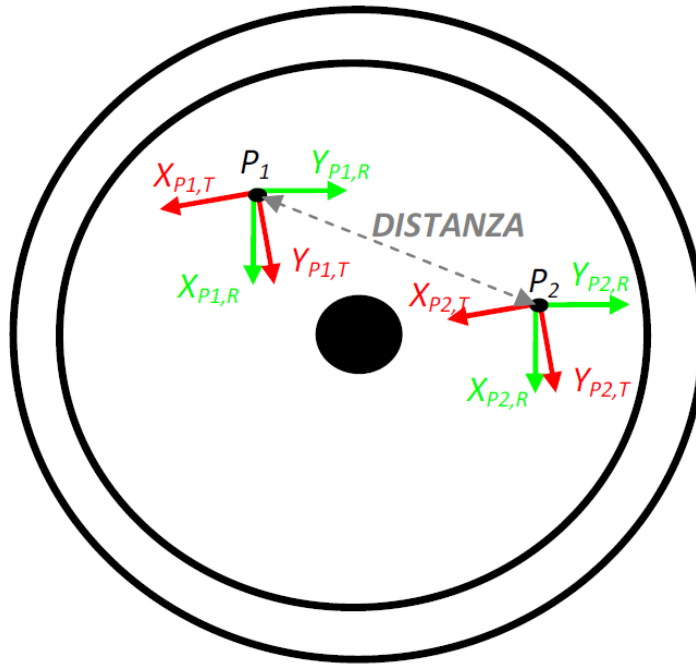


Figura 23: Sistema di Riferimento Robot e Telecamera dei due Oggetti.

tenuti memorizzando la posizione del centro di ogni oggetto che si muove sul piatto in tre istanti temporali successivi, ruotando il nastro sempre di 90° . L'oggetto descrive una traiettoria che non è altro che una circonferenza concentrica a quella del flexibowl. Quindi mi basta trovare i segmenti passanti per i centri e da questi mi trovo le normali le quali si incroceranno in un punto che coincide proprio con il centro del flexibowl, vedi figura 24. In definitiva le coordinate dell'oggetto rispetto al sistema di riferimento robot sono date dal prodotto di due matrici di trasformazione che non sono altro le operazioni svolte nelle righe precedenti. Come si vede dall'equazione 1 la prima trasformazione ci permette di passare dal sistema telecamera a quello robot invece la seconda dal centro del flexibowl a quello della telecamera. Il prodotto di queste due matrici per le coordinate X, Y dell'oggetto riferite al centro della tazza ci fornisce le nuove coordinate nel sistema robot.

$$\begin{Bmatrix} X_P \\ Y_P \end{Bmatrix}_R = [T_{T,R}] * [T_{C,T}] * \begin{Bmatrix} X_P \\ Y_P \end{Bmatrix}_C \quad (1)$$

L'equazione 2 mostra le coordinate X, Y dell'oggetto riferite al centro del flexibowl.

$$\begin{cases} X_P = \rho * \cos\theta \\ Y_P = \rho * \sin\theta \end{cases} \quad (2)$$

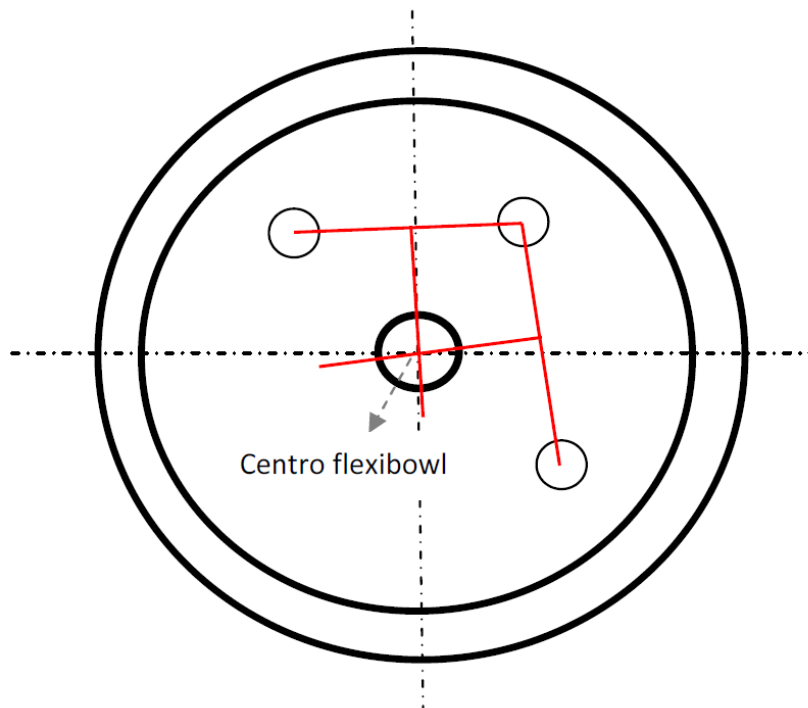


Figura 24: Metodo per Trovare il Centro del Flexibowl.

RICERCA OGGETTI E VERIFICHE

5.1 INTRODUZIONE

Il manipolatore per poter prelevare gli oggetti presenti sulla superficie rotante del flexibowl deve sapere la loro posizione. Per ottenere ciò è stata utilizzata una telecamera che dialoga con il robot e gli fornisce tutti i dati necessari per svolgere tale mansione. Nel diagramma in figura 25 sono elencati i passi svolti per ottenere da una foto tutte le informazioni relative agli oggetti presenti.



Figura 25: Schema Funzionamento Ricerca Oggetti.

5.2 RICERCA DEGLI OGGETTI

Il programma che ricerca gli oggetti salva in opportune variabili le informazioni sulla loro posizione, visibile in forma di schema nella figura 26. Tramite il software AdeptSight conosco il numero di pezzi

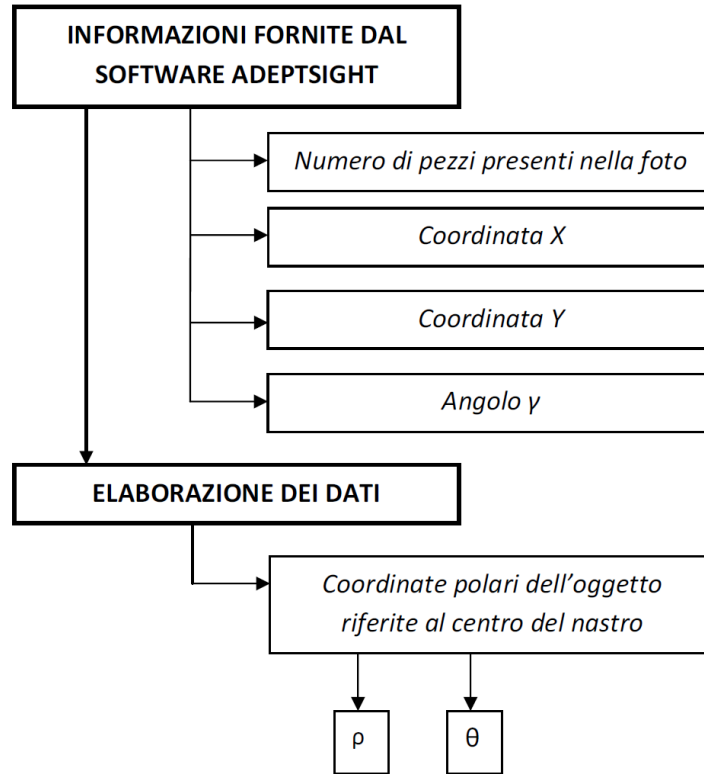


Figura 26: Schema Funzionamento Ricerca Oggetti.

presenti, le coordinate X, Y e l'angolo di rotazione γ di ogni oggetto rispetto il sistema di riferimento della telecamera. Da queste informazioni mi ricavo due grandezze: la distanza dell'oggetto dal centro del flexibowl (ρ) e l'angolo effettivo (θ) rispetto allo zero della tazza, vedi in figura 27. In questo modo ho espresso la posizione dell'oggetto in coordinate polari riferite al centro del flexibowl, vedi equazione 3.

$$\begin{cases} X_P = \rho * \cos\theta \\ Y_P = \rho * \sin\theta \end{cases} \quad (3)$$

Dove ρ e θ si sono ottenuti dalle seguenti equazioni:

$$\begin{cases} \rho = \sqrt{(X_c - X_P)^2 + (Y_c - Y_P)^2} \\ \theta = \text{atan2}((X_c - X_P)/(Y_c - Y_P)) - \beta \end{cases} \quad (4)$$

Invece γ rappresenta l'angolo di rotazione dell'organo di presa, in questo modo conosco con precisione come viene afferrato l'oggetto. Ora ci rimane di ordinarli rispetto l'angolo trovato in ordine decrescente. Questo ha senso perché il flexibowl ruota in senso orario e i

pezzi saranno presi con giudizio e non arbitrariamente con notevole vantaggio di ridurre i tempi di spreco che il manipolatore commette fra il rilascio del pezzo e la nuova presa.

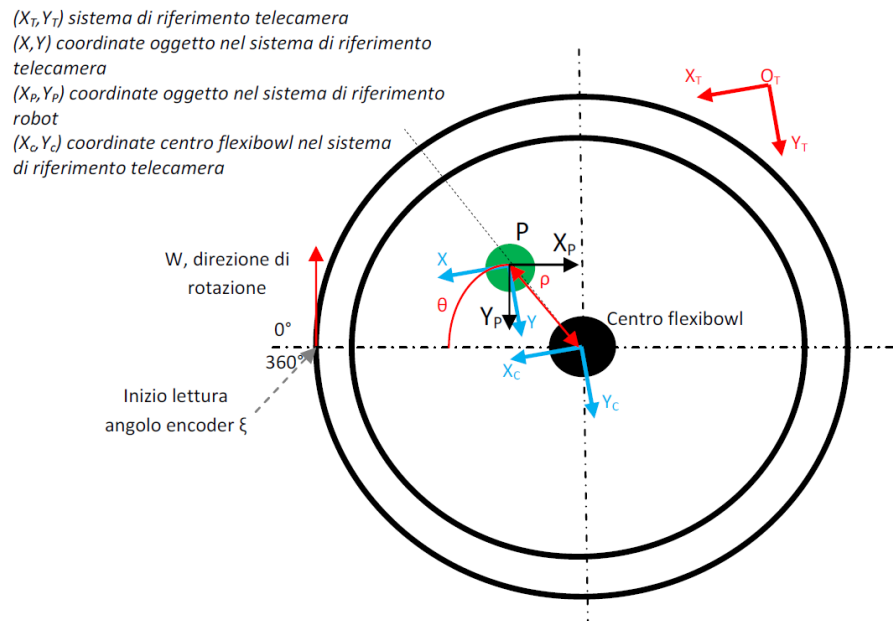


Figura 27: Informazioni Relative all'Oggetto.

5.3 VERIFICA DELLA POSSIBILITÀ DI PRESA

Siamo giunti ora ad avere una foto nella quale conosco il numero di oggetti presenti e i dati necessari per il loro prelievo. Questo, però non è sufficiente in quanto si possono manifestare delle situazioni che sono indesiderate come ad esempio che il manipolatore durante la presa vada a colpire con la pinza degli ostacoli. Fra un pò sarà tutto più chiaro. Dalla figura 2 si può notare che il flexibowl è caratterizzata da un bordo esterno e da un disco posto in centro rialzato, quindi queste due sporgenze possono essere facilmente cause di urto con l'organo di presa del robot. Inoltre non mi va nemmeno bene la presa di oggetti troppo vicini sempre per lo stesso motivo. Insomma ci sono tre limiti e sono: `limitdist2obj`, `limitdistcentre` e `limitdistboard`; visibili in figura 28. Il primo limite si riferisce alla distanza fra due oggetti (`limitdist2obj`), con questo limite intendo che non tollero la configurazione di due pezzi troppo vicini. Il secondo limite come il terzo (`limitdistcentre` e `limitdistboard`) riguarda una distanza di sicurezza dal centro e dal bordo del nastro. La ricerca del vettore contenente solo oggetti prendibili deve prendere in considerazione i tre limiti appena descritti e il risultato sarà un array di dimensione minore o uguale a quello che ci fornisce il programma di ricerca degli oggetti. Inoltre non ci fermiamo assolutamente qui con il controllo perché com'è già visto bisogna rispettare anche i rispettivi spazi di lavoro.

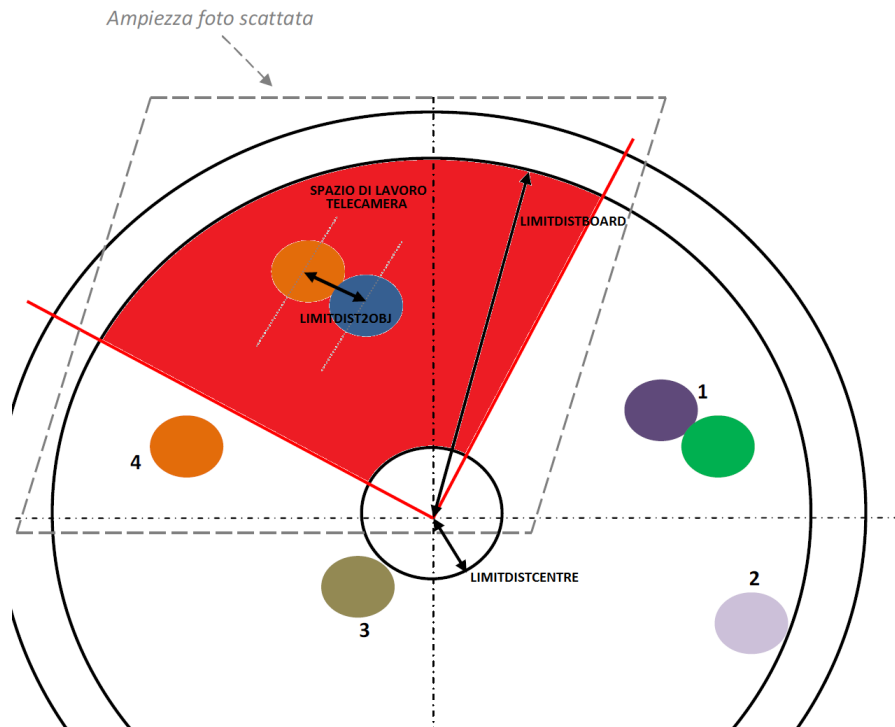


Figura 28: Limiti.

Creo così una finestra fittizia per delineare qual'è lo spazio permesso per raccogliere le informazioni dalla foto. Scarto tutti i pezzi al di fuori di tale regione. Per cui gli oggetti 1,2,3,4 di figura 28 non saranno prelevati dal manipolatore perché non soddisfano i criteri appena analizzati. Tutte queste operazioni sono necessarie per poter lavorare in modo automatico senza l'intervento dell'uomo. Il quale per evitare queste spiacevoli situazioni dovrebbe arrestare la macchina il che implica un aumento dei costi e dei tempi di produzione. Tuttavia il sistema opera così in sicurezza evitando infortuni agli addetti al lavoro e il danneggiamento delle parti che lavorano.

5.3.1 Rotazione irregolare del flexibowl

Finora si è considerato che il sistema funzioni alla perfezione. Questo ragionamento è valido solo dal punto di vista teorico e non pratico. Idealmente il sistema lavora senza commettere errori però, nella realtà non è così, cioè possono succedere dei fatti che vanno a influire sul risultato finale. Nel progetto si è deciso di scattare una foto ogni 90° di rotazione e quindi quattro foto ogni giro completo. Può succedere però, che il flexibowl, pur essendo stato programmato per ruotare di 90° , ruoti per una quantità superiore o inferiore causando due problemi che portano come risultato negativo o la presa a vuoto oppure la perdita dell'oggetto (ovvero non viene fotografato e il manipolatore non può così andarlo a prelevare); vedi figura 29. Quest'ultimo si può

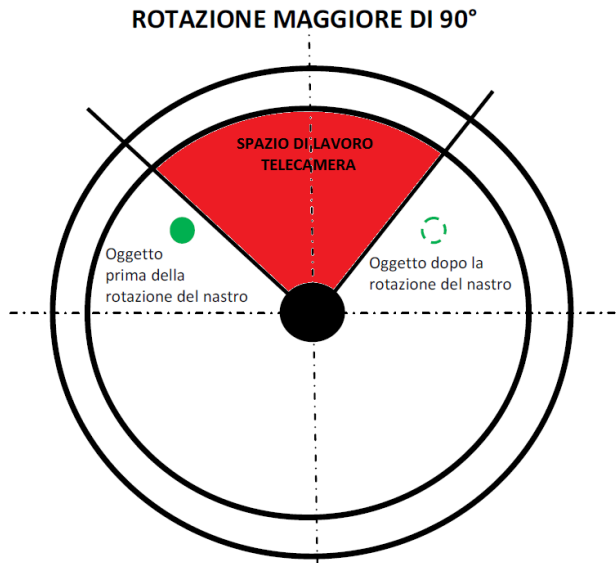


Figura 29: Rotazione Irregolare.

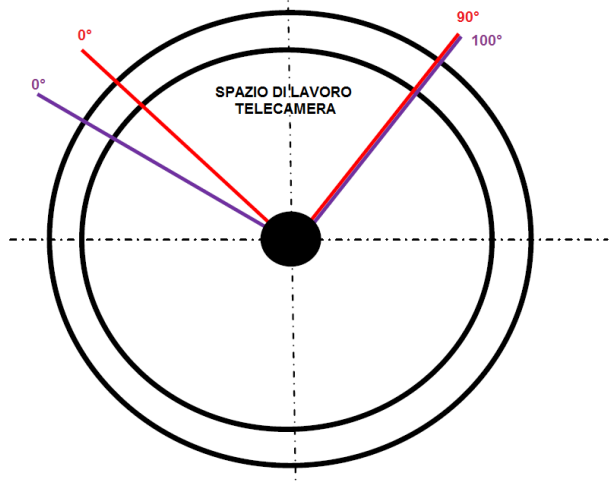


Figura 30: Spazio di Lavoro Telecamera Modificato.

risolvere andando ad allargare il campo di visione ovvero lo spazio di lavoro della telecamera. Passiamo così dai 90° di partenza ad una finestra di 100° , vedi figura 30. Naturalmente la dimensione totale è a piacere dell'utente. Inoltre può succedere che in due foto successive compaiano degli oggetti che sono in realtà gli stessi con il risultato che il manipolatore la prima volta prende il pezzo e la seconda volta fa l'azione per prendere l'oggetto ma non preleva niente. Questo non deve succedere e deve essere risolto per agevolare un funzionamento il più corretto possibile. Il problema è stato risolto facendo un controllo sulla posizione effettiva del pezzo prima e dopo la rotazione vedi figura 31. Mi ricavo le coordinate X, Y dell'oggetto nel seguente modo:

- P oggetto allo scatto della foto
- P' oggetto traslato di 90° in avanti per mostrare che P e D sono lo stesso oggetto
- D oggetto allo scatto della foto successiva

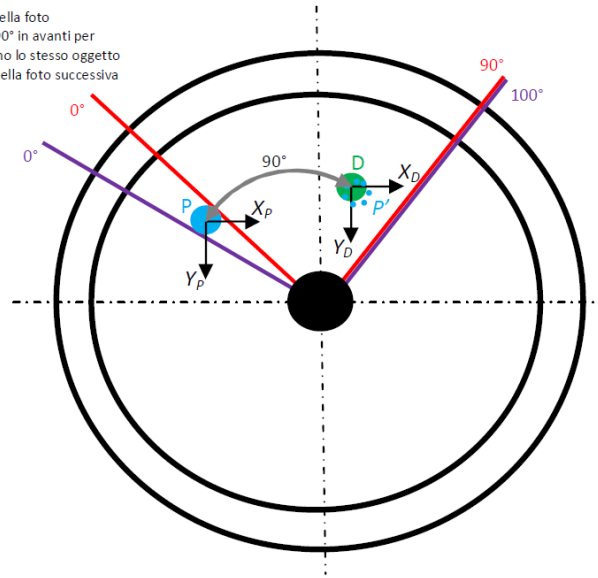


Figura 31: Stesso Oggetto.

$$\begin{cases} X_P = \rho_P * \cos(\theta_P + 90) \\ Y_P = \rho_P * \sin(\theta_P + 90) \end{cases} \quad (5)$$

$$\begin{cases} X_D = \rho_D * \cos\theta_D \\ Y_D = \rho_D * \sin\theta_D \end{cases} \quad (6)$$

Usando l'equazione 5 mi trovo le coordinate dell'oggetto riferito alla foto precedente ruotato di 90° in avanti, invece usando la 6 mi trovo le coordinate del pezzo della foto successiva. Dalla conoscenza di questi dati posso calcolarmi la distanza fra i due oggetti e verificare se è minore di due volte il raggio, se risulta minore allora significa che sono lo stesso oggetto, vedi equazione 7, e quindi non lo prendo.

$$\sqrt{(X_P - X_D)^2 + (Y_P - Y_D)^2} < 2 * r \quad (7)$$

Il controllo è fatto in una zona specifica che è vicino al limite superiore della regione d'azione della telecamera, ovvero ad una distanza pari all'incremento della finestra. In fine ottengo un array contenente solo gli oggetti che possono essere presi, privandomi di tutti quelli che non soddisfano i limiti impostati.

5.4 RACCOLTA DI INFORMAZIONI IN UN DATABASE

In definitiva ho un vettore che contiene tutti i pezzi prendibili ordinati in base all'angolo di presa (θ). Ora ha senso inserire i dati di ciascun oggetto all'interno di una matrice che servirà al manipolatore per leggere le informazioni. E' ovvio che se in una foto non ho nessun oggetto non vado a scrivere niente. Innanzitutto la matrice ha lo scopo di raggruppare in base al numero di foto tutti gli oggetti che sono accessibili, in questo modo il manipolatore sa quanti oggetti ci sono in ogni determinata foto e conosce anche l'angolo encoder ξ_0 per poterli prendere. Se così non fosse ad ogni nuova foto, e quindi lettura dell'angolo encoder, (ricordo che queste due operazioni sono svolte dal primo blocco di istruzioni del main, vedi capitolo 4) il robot continuerebbe a prelevare gli oggetti in ordine di arrivo, ma l'angolo encoder, un dato essenziale per la presa degli oggetti, varierebbe e quindi non darebbe il tempo al manipolatore di terminare la presa dei pezzi relativi alla foto precedente. Questo causa degli errori che si traducono in un errato inseguimento degli stessi. Quindi la creazione di un registro ha proprio lo scopo di far lavorare in simultanea la telecamera e il robot senza che ci siano scambi di informazioni direttamente. Uno lavora in scrittura e l'altro in lettura, seguendo sempre questo ordine. Così il manipolatore legge nel database le informazioni foto per foto e non avrò mai che l'arrivo di nuovi dati possa influenzare il lavoro attuale. Sono riuscito così a separare le due cose. La tabella 1 ci mostra un esempio del database. Dove ρ è la distanza fra il centro dell'oggetto e quello del flexibowl, θ l'angolo di presa, ξ_0 è l'angolo encoder, γ è l'angolo di rotazione dell'organo di presa e Coordinatede X, Y sono le coordinate del pezzo in millimetri nel sistema di riferimento della telecamera.

DATI	FOTO 1	FOTO 1	FOTO 1	FOTO 2	FOTO 2
Foto	1	1	1	2	2
ρ	110,2251	130.9987	150.1201	92.72939	160.2482
θ	120,2545°	102,3698°	82.5654°	64.71265°	44.48133°
ξ_0	9000	9000	9000	9090	9090
γ	-143.1108	-93.3696	105.2041	60.76089	81.2216
X	690.1828	1199.191	943.7101	860.5833	705.9198
Y	475.4227	492.3663	620.7101	320.4446	758.85187

DATI	FOTO 3	FOTO 4	FOTO 4	FOTO 4	FOTO 4
Foto	3	4	4	4	4
ρ	160.2482	104.0329	196.8232	82,2251	131.5485
θ	44.48133°	108.9277°	91.363°	88.2258°	56.76744°
ξ_0	9180	9270	9270	9270	9270
γ	81.2216	-93.3696	105.2041	43.1108	-58.8748
X	705.9198	552.7993	1152.587	719.2814	800.2333
Y	758.85187	386.1624	655.0406	315.7678	402.7414

Tabella 1: Esempio di un Registro

LA PRESA DI OGGETTI IN MOVIMENTO

6.1 INTRODUZIONE

D'ora in poi ci occupiamo di descrivere il processo di presa di oggetti in movimento da nastri trasportatori circolari. Ha senso andare a sviluppare una tecnica di presa che non obblighi l'arresto della macchina, che in questo caso si tratterebbe del nastro circolare. Per realizzare ciò è necessario che il manipolatore si muova in modo solidale con la tazza rotante. Inoltre sono stati risolti alcuni potenziali rallentamenti dovuti al manipolatore. In figura 32 è visibile uno schema illustrativo.

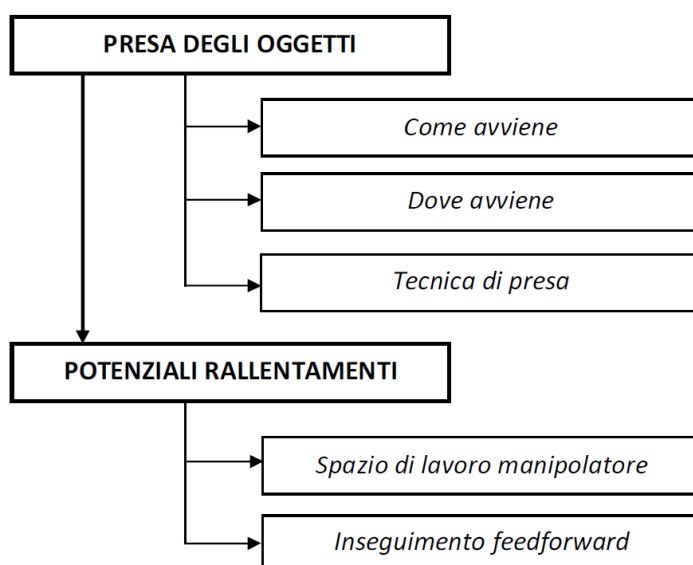


Figura 32: Schema di Funzionamento Presa Oggetti.

6.2 TECNICA PRESA AL VOLO

La presa degli oggetti sulla superficie rotante non è una cosa assai banale. Facilmente si può pensare che una volta trovate le coordinate del pezzo si vada a fermare il nastro trasportatore per permettere al manipolatore di prelevarlo. Questo è uno spreco di tempo con un notevole costo di produzione. Per risolvere tale problema è congeniale andare a prendere l'oggetto senza fermarlo. La tecnica utilizzata prevede infatti che il manipolatore si posizioni sopra l'oggetto e poi compia tre azioni: discesa sull'oggetto, inseguimento dell'oggetto e

risalita con l'oggetto. Questo prende il nome di "circular conveyor tracking".

6.2.1 Dove e come avviene la presa

Il codice "V+" implementato per la presa prevede che il manipolatore possa operare solamente all'interno del suo spazio di lavoro, come si vede dalla figura 21. Se non ci sono oggetti da prelevare esso rimane nella posizione di home o in quella di rilascio del pezzo (target). Non appena arrivano pezzi il robot è attivato per andarli a prendere. Se il pezzo si trova ancora all'interno dello spazio di lavoro della telecamera il manipolatore si posiziona sul suo limite inferiore e non appena l'angolo di rotazione dell'oggetto (φ , angolo che mi dice dove si trova in ogni istante l'oggetto) eguaglia tale limite inizia a svolgere la presa. Invece se il pezzo è già all'interno dello spazio di lavoro del robot posso già eseguire l'operazione di prelievo senza aspettare nessun vincolo.

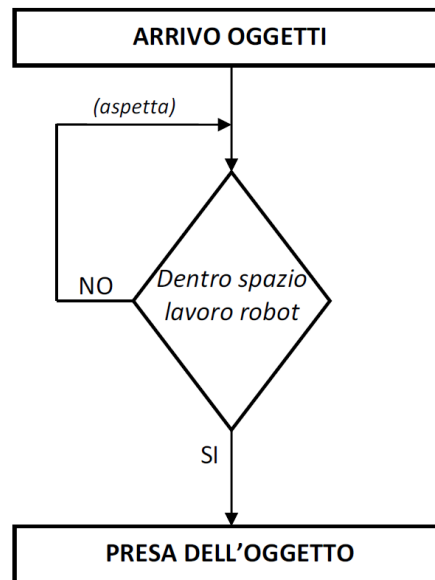


Figura 33: Schema Funzionamento Presa.

6.2.2 Metodo per svolgere la traiettoria circolare

Il manipolatore per svolgere una traiettoria circolare deve sapere molte informazioni e legarle fra loro. Infatti la presa dell'oggetto viene eseguita tramite un comando (MOVE) e una serie di prodotti matriciali, vedi equazione 8.

$$[T_{\text{goal}}] = [T_{\text{centro}}] * [R\varphi] * [T\rho] * [T_h] * [R\gamma] \quad (8)$$

Innanzitutto deve conoscere il centro del flexibowl rispetto il sistema di riferimento robot. Poi gli devo passare l'angolo di rotazione

dell'oggetto (φ), il quale è espresso sempre rispetto al sistema di riferimento robot, che rappresenta la direzione radiale. Questo angolo è la somma di tre termini, vedi l'equazione 9.

$$\varphi = \xi + \theta + \beta \quad (9)$$

$$\xi = \xi_i - \xi_o \quad (10)$$

Attenzione che non è fisso perché il nastro non si ferma mai. Devo quindi conoscere: l'avanzamento del nastro ξ (che è dato dalla differenza della lettura dell'encoder nell'istante attuale e di quello allo scatto della foto, vedi equazione 10), la coordinata polare (θ) del pezzo e l'angolo per portarlo nel sistema di riferimento robot (β). Quest'ultimo angolo vale circa 90° , vedi capitolo 4.3. E' possibile ve-

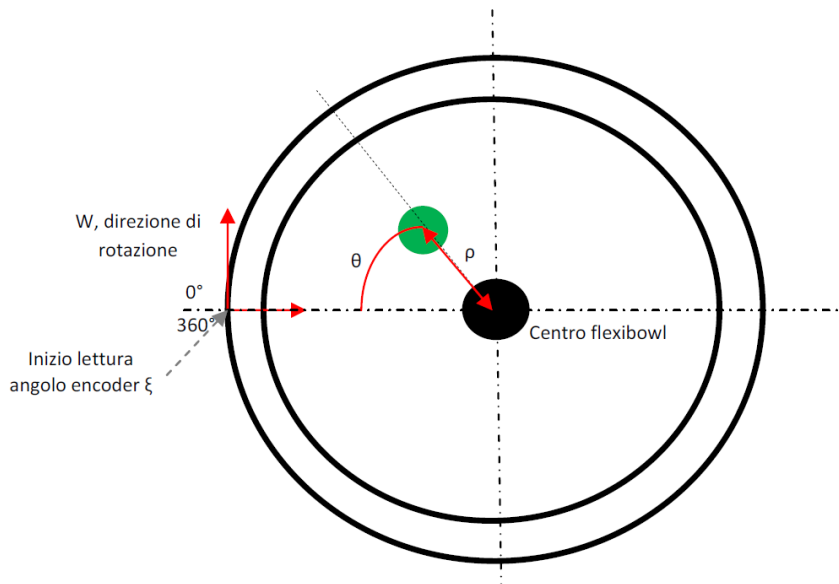


Figura 34: Parametri Oggetto allo Scatto della Foto.

dere in figura 34 e 35 questi angoli. Questo fa sì che il manipolatore si muova in modo solidale con il nastro trasportatore. Ora ci rimane solo da capire come prelevare l'oggetto. Gli devo passare altre tre informazioni. La prima è la distanza tra il centro dell'oggetto e il centro del flexibowl (ρ), vedi figura 34. In questo modo ho che il robot è posto sopra l'oggetto e lo segue percorrendo la stessa traiettoria circolare. La seconda è la coordinata zeta del robot (h) che è usata per compiere l'azione di prelievo e rappresenta la quota di differenza tra il piatto e la pinza. Tramite un ciclo posso svolgere le tre fasi per la presa dell'oggetto, in accordo con la figura 36, e sono:

- Diminuisco di una piccola quantità la coordinata zeta fino ad avere l'organo di presa a livello del baricentro dell'oggetto.

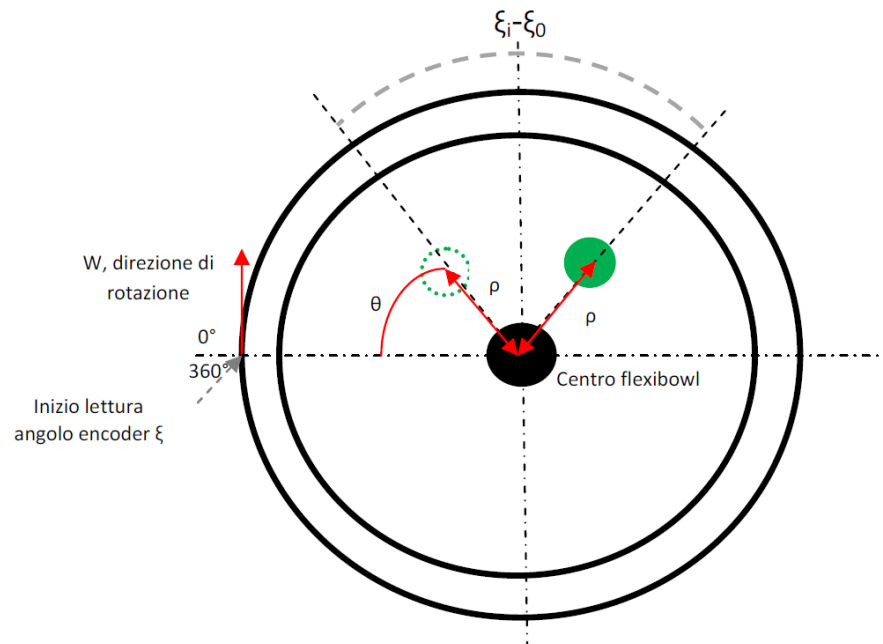


Figura 35: Parametri Oggetto dopo che il Nastro si è spostato.

- Inseguo l'oggetto e chiudo le pinze.
- Aumento della stessa quantità la coordinata zeta per salire con l'oggetto.

La terza è l'angolo γ del robot che ci serve per capire come orientare l'organo di presa per prelevare l'oggetto. Questo è importante perché pezzi di dimensioni diverse non sono prelevati allo stesso modo. Bisogna vedere come sono fatti perché se sono tondi non ho nessun problema, ma se sono rettangolari ha senso dirgli prendilo per il lato lungo o corto. Naturalmente questo valore di angolo ci è fornito dal software AdeptSight che noi abbiamo chiamato γ nel capitolo 5.2. Questo ci ha permesso di prelevare l'oggetto percorrendo la traiettoria circolare dello stesso come mostrato in figura 37, che è ottenuta plottando in un grafico le coordinate X,Y,Z del robot.

6.3 SPAZIO DI LAVORO DEL MANIPOLATORE

Come già descritto nel paragrafo 4.2 il manipolatore agisce in una determinata area del nastro. E' importante sottolineare la presenza del limite inferiore e superiore che rappresentano rispettivamente l'inizio e la fine dello spazio di lavoro concesso al robot sul nastro per fare la presa degli oggetti. Quindi ho la necessità di fare un controllo ogni qual volta che vado a prendere un pezzo all'interno della tazza. Questo serve per evitare che il manipolatore si blocchi nel limite superiore e non porti così a termine il prelievo di un oggetto. In termini più chiari il robot quando comincia a svolgere le sue operazioni si

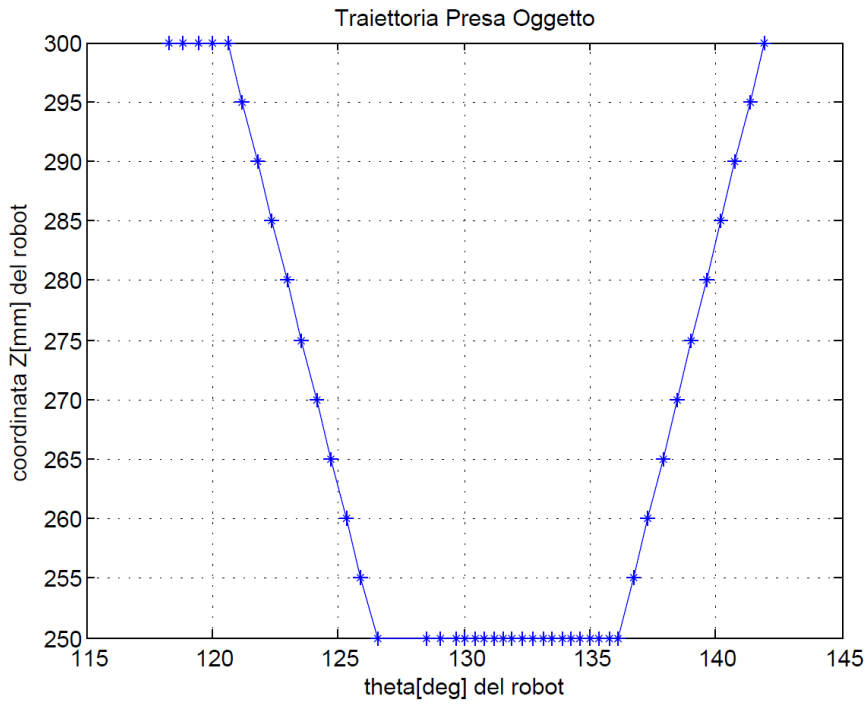


Figura 36: Traiettorie di Presa.

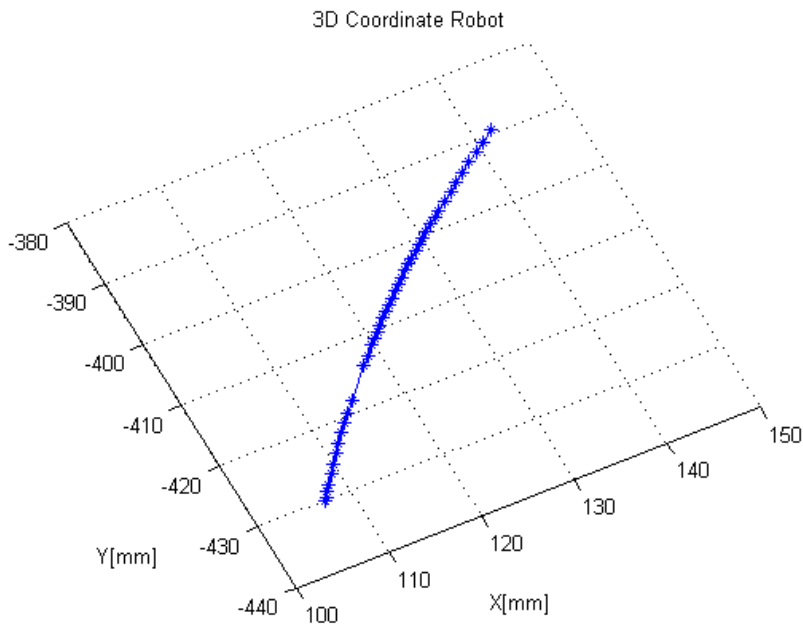


Figura 37: Traiettorie Circolare di Presa.

trova all'interno del nastro circolare e nel frattempo viene monitorata la zona di lavoro con lo spostamento del nastro. Infatti questo è un angolo e coincide con quello della relazione 9. Se esso eguaglia il valore del limite superiore il manipolatore si blocca in questa posizione senza terminare la presa dell'oggetto e aspetta le nuove istruzioni, vedi figura 38. Come si può capire questa cosa non è accettabile. E' quindi opportuno fare in modo che il manipolatore si fermi e aspetti di prendere l'oggetto il giro successivo. La soluzione al problema

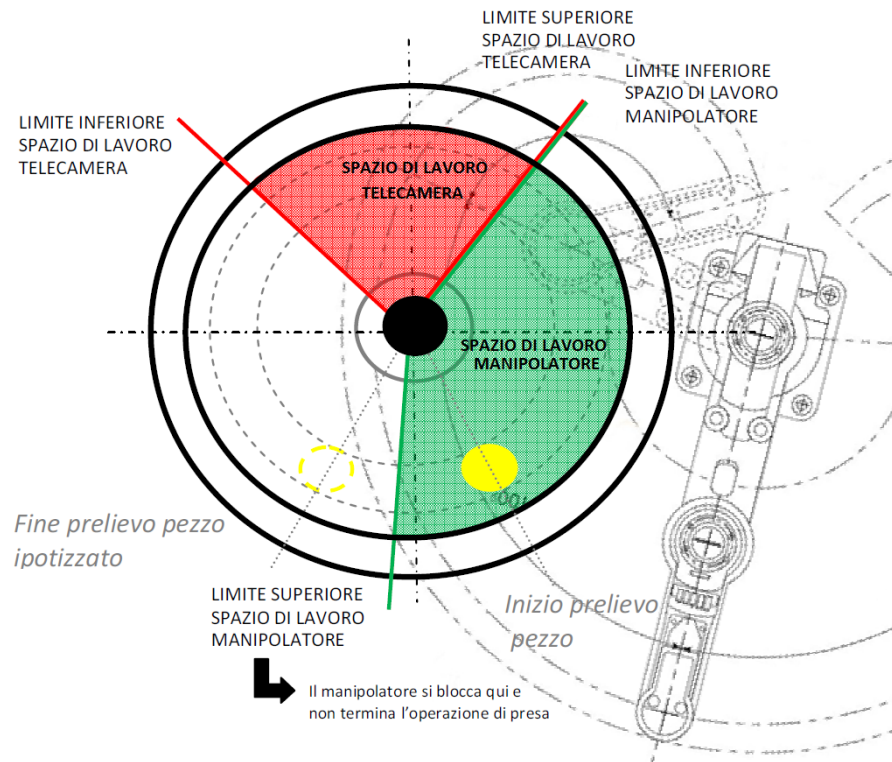


Figura 38: Problema Limite Superiore Manipolatore.

prevede l'inserimento di un controllo di tipo dinamico. Non avrebbe senso uno statico, che si traduce in una linea immaginaria nella quale il manipolatore oltre non può più lavorare, perché in base alla velocità del robot questa cambierebbe, vedi figura 39. E' logico pensare che se il robot è impostato alla velocità massima questa linea fittizia sarebbe più piccola, ma se il robot lavora alla metà della massima velocità questa sarebbe più grande. In definitiva il suo settaggio ci comporterebbe solamente uno spreco di tempo. Fare un controllo dinamico è di grande aiuto e ci risolve i problemi appena citati. Naturalmente è sempre un angolo e quindi si ottiene semplicemente lavorando sui dati che ho a disposizione. Il controllo prevede inoltre la conoscenza della velocità di rotazione del nastro trasportatore e del tempo d'impiego per fare un prelievo. La velocità di rotazione della tazza essendo fissa si può calcolare facilmente andando a vedere quanto impiega a fare un giro completo (360°), così ricavo la velocità in $[\circ/s]$. Il tempo per pre-

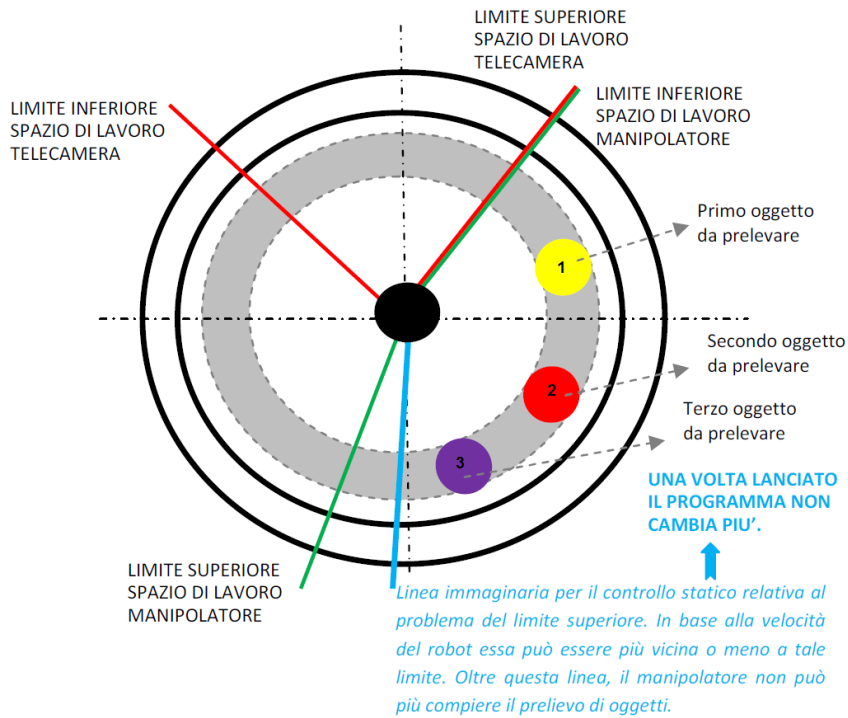


Figura 39: Limite Statico.

levare un pezzo lo misuro solamente una volta, quando eseguo per la prima volta il programma principale. All'inizio non ho nessun problema perché sono certo della riuscita del prelievo; in qualsiasi posto l'oggetto si trovi nello spazio riservato alla telecamera quando entra nello spazio dedicato alla presa riesco sempre a prelevarlo. Con i dati a disposizione posso calcolarmi l'angolo (α) e vedere se è maggiore o minore di zero. Per trovarlo mi basta sottrarre al limite superiore dello spazio di lavoro del robot l'angolo di rotazione (φ), i gradi necessari per la presa di un oggetto (li ottengo moltiplicando la velocità in $[\text{°}/\text{s}]$ della tazza per il tempo in $[\text{s}]$ impiegato per fare il prelievo) ed eventualmente l'angolo ϑ (vedi capitolo 6.4). Se è maggiore allora vuol dire che ho il tempo e lo spazio per la riuscita del prelievo altrimenti no. Il controllo dinamico consiste in un calcolo di questa linea immaginaria che cambia ogni qual volta che si va a prelevare un oggetto, vedi figura 40. Il vantaggio che offre il limite dinamico rispetto a quello statico è che non è inserito manualmente dall'utente nel programma, ma è calcolato e aggiornato direttamente dal software. La soluzione comporta un miglioramento della prestazione offerta dal manipolatore perché nel caso di oggetti fuori range il programma che si dedica alla presa termina liberando così il task. In questo modo è già pronto ad andare a prelevare pezzi relativi alla foto successiva senza perdite di tempo. In questo modo evito due scocciature. Una riguarda la non riuscita terminazione dell'operazione di presa e l'altra una perdita di tempo. Essendo nell'ambito industriale un fattore

determinante è per noi essenziale fare questo controllo.

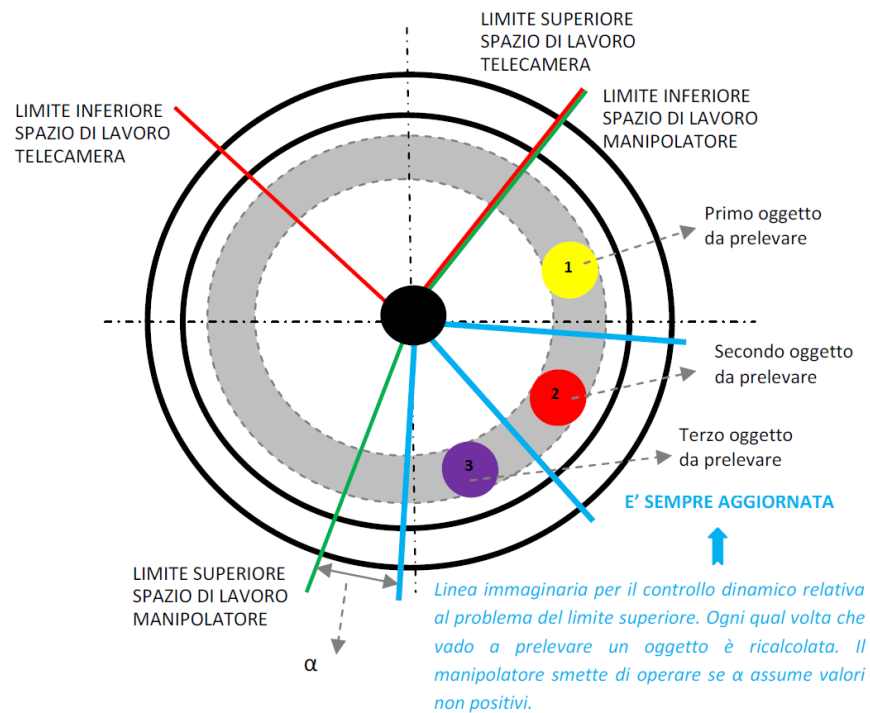


Figura 40: Limite Dinamico.

6.4 INSEGUIMENTO IN FEEDFORWARD

La presa degli oggetti avviene all'interno dello spazio di lavoro del manipolatore perciò se il pezzo fotografato è ancora nell'area riservata alla telecamera il robot si posiziona nel suo limite inferiore, vedi figura 21. Non appena l'oggetto raggiunge tale posizione il programma permette al manipolatore di eseguire la presa al volo. In questa situazione ho che la presa è eseguita senza che siano commessi degli errori perché andando a visualizzare su di un grafico l'andamento della coordinata zeta del robot e l'angolo di percorrenza dello stesso ottengo un andamento lineare in accordo con la figura 36. Tutt'altro si verifica se l'oggetto è già entrato nel campo d'azione del robot. In questo caso ho che il manipolatore si trova o nella posizione di home o in quella di rilascio del pezzo (target). Nel frattempo che il robot si posizioni sopra l'oggetto per iniziare la presa ho che il pezzo si è spostato. Si ricordi che il nastro trasportatore circolare è sempre in movimento e non viene mai fermato. Quindi ho che il manipolatore inizia la fase di discesa sull'oggetto prima ancora di raggiungerlo. Questo succede perché il programma prevede che quando l'oggetto entra nel raggio d'azione del robot questo possa iniziare a eseguire la presa al volo. Essendo più veloce l'elaborazione di calcolo del controller rispetto a quella di movimentazione del manipolatore avrò questo

fatto indesiderato. Il risultato può essere osservato in figura 41. Co-

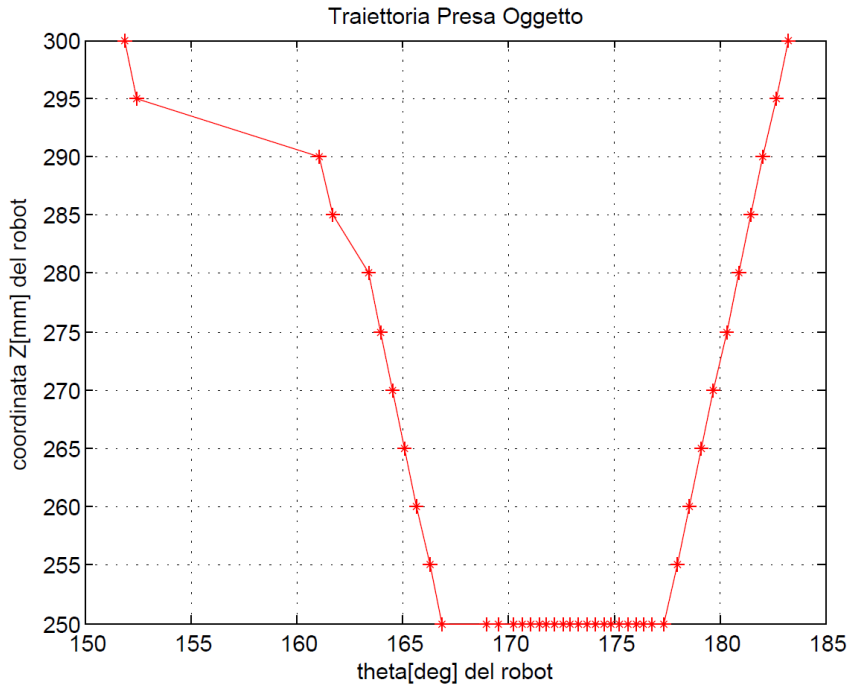


Figura 41: Traiettorie di Presa errata.

me si può osservare nel tratto iniziale di discesa il manipolatore sta correggendo la sua posizione per essere proprio sopra il pezzo. Però, ho che ci arriva con la pinza che si sta già abbassando. Non mi va bene perché non è bello da vedere, ma soprattutto potrebbe andare ad urtare l'oggetto d'interesse o altri pezzi vicini che sono più grandi in altezza. Questo provocherebbe solamente un disastro con l'obbligo da parte dell'operatore di arrestare la macchina per evitare danni. L'unica soluzione possibile per sfuggire a ciò è quella di mandare il manipolatore in anticipo rispetto all'oggetto. In questo modo sposto il limite inferiore del robot in un punto più avanti, vedi figura 42. Così ottengo il funzionamento desiderato cioè che il manipolatore si posizioni sopra l'oggetto e poi inizia le tre fasi per la presa. Tutto ciò ha senso se questo spostamento del limite non comporti un aumento dei tempi di lavoro. Ovviamente il calcolo del nuovo limite è fatto in modo che sia l'oggetto e il manipolatore arrivino nel punto d'incontro allo stesso istante. Per riuscire a ottenere questo risultato devo conoscere alcuni dati assai importanti che sono: la distanza del manipolatore dall'oggetto nell'istante in cui devo andarlo a prelevare (d), la velocità di rotazione del nastro e la velocità di lavoro del robot.

$$d = \sqrt{(x_{\text{target}} - x_i)^2 - (y_{\text{target}} - y_i)^2} \quad (11)$$

In accordo con la figura 42 si deve cercare il valore di angolo ϑ da sommare all'angolo di rotazione dell'oggetto (φ), (vedi espressione 9

per φ), per ottenere il risultato sperato. La relazione 12 mostra come si ottiene il nuovo limite inferiore relativo allo spazio di lavoro del manipolatore.

$$\mu = \varphi + \vartheta \quad (12)$$

Per il calcolo dell'angolo ϑ mi sono basato su una stima del tempo

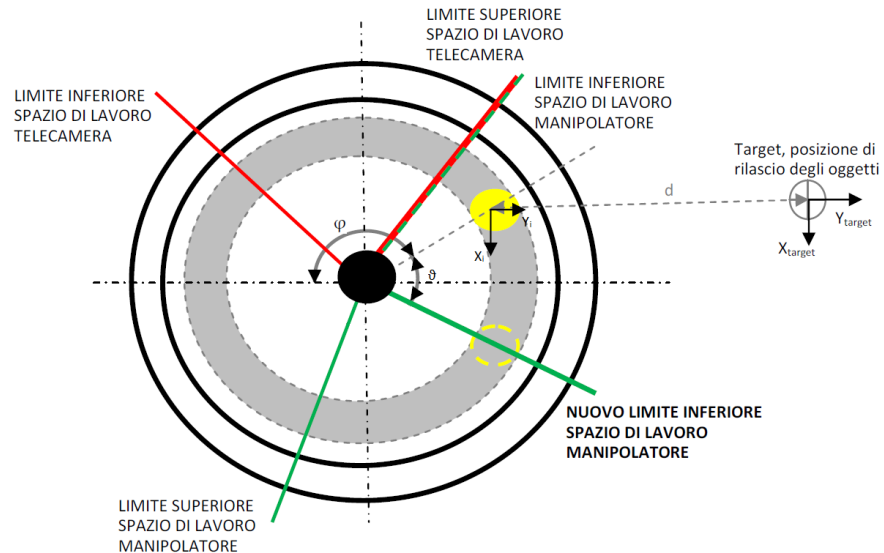


Figura 42: Inseguimento FeedForward.

che impiega il manipolatore per raggiungere l'oggetto da dove esso si trova. Naturalmente questo calcolo deve tenere conto che il manipolatore può operare a velocità diverse e quindi devo cercare di rendere flessibile questo conto. Combinando le informazioni in possesso cioè, la conoscenza della velocità di rotazione del nastro trasportatore ($12^\circ/s$) e il tempo impiegato dal robot per percorrere tale distanza mi trovo l'angolo desiderato direttamente in gradi. Con questa operazione risolvo il problema e ciò che ottengo è comparabile con quello che si vede nella figura 36.

CONCLUSIONI

Il progetto realizzato ha centrato in pieno l'obiettivo prefissato. In questo modo le aziende oltre a prelevare oggetti da nastri trasportatori lineari lo possono fare anche da quelli circolari. Inoltre l'utilizzo di un manipolatore scara ci ha permesso di ottenere dei risultati abbastanza buoni, dai test effettuati è emerso che compie l'operazione di pick & place con una continuità di 26 pezzi al minuto. Si può quindi intuire che il codice scritto dato in pasto ad un Adept 4 o a qualsiasi altro robot di nuova generazione avrà sicuramente dei risultati migliori. Il lavoro svolto non puntava sulla quantità e quindi sul numero di pezzi in grado di prelevare il manipolatore, ma bensì sullo sviluppo di una tecnica innovativa in grado di superare le problematiche riscontrate nell'ambito lavorativo. Questo ci ha permesso di creare un sistema che montato in una catena produttiva può funzionare in modo autonomo assieme agli altri organi della catena.

APPENDIX

La relazione 13 mette in evidenza le analogie utilizzate nel testo per i parametri usati nel codice. In questo modo chi consulterebbe il codice potrà avere una leggenda per comparare le nozioni con il testo di riferimento.

$$\begin{aligned}
 \varphi &= a \\
 \xi &= c - \text{absang} = \text{BELT}(\%belt1,0) * \text{conver} - \text{absang} \\
 \theta &= \text{vang} \\
 \beta &= \text{aaa} \\
 \rho &= \text{rdist} \\
 \gamma &= \text{rotazione} \\
 \alpha &= \text{ang}_s\text{ufficiente} \\
 \vartheta &= \text{feedfwd} \\
 d &= \text{distance}
 \end{aligned}
 \tag{13}$$

Di seguito sono visibili le parti salienti del codice V+ per raggiungere lo scopo prefissato. I commenti sono rappresentati tramite un punto e virgola.

Listing 1: Programma Principale

```

WHILE (NOT k) DO

    ; Attivazione FlexiBOWL e Abilitazione alla ricerca
    degli oggetti

    IF STATUS("test") == -1 THEN

        EXECUTE 1 test()
        absang = BELT(%.belt1,0)*conver
        indice.test = indice.test+1
        CALL selezionaoggetto()
        ;EXECUTE 3 registro()
        CALL rregistro()

    END

    ; Abilitazione alla presa degli oggetti

    IF (indice.test > 1) THEN

        IF TASK(1,2) <> 4 THEN

```

```

EXECUTE 2 vpick.rregistro()

END

END

END ;while

```

Listing 2: Ricerca Oggetti

```

$ip = "147.162.95.34"

VRUN $ip, 1
VWAITI (1) $ip, 0
instance_count = VRESULT($ip, 1, 2, 1, 1310) ;numero
oggetti trovati

FOR i = 1 TO instance_count

    rotazione[i] = VRESULT($ip, 1, 2, i, 1314)
    x = VRESULT($ip, 1, 2, i, 1315)
    y = VRESULT($ip, 1, 2, i, 1316)
    y = 1040-y
    valueang = VRESULT($ip, 1, 2, i, 1314)
    vdist = SQRT((xc-x)*(xc-x)+(yc-y)*(yc-y))
    rdist[i] = vdist*mmpx
    vang[i] = ATAN2(x-xc,y-yc)-aaa
    xx[i] = x*mmpx
    yy[i] = y*mmpx

    IF vang[i] < 0 THEN

        vang[i] = vang[i]+360

    END

END

;Ordino gli angoli in modo decrescente

FOR ord.i = 1 TO instance_count-1

    FOR ord.j = ord.i+1 TO instance_count

        IF vang[ord.i] < vang[ord.j] THEN

            ord.temp1 = vang[ord.i]
            vang[ord.i] = vang[ord.j]
            vang[ord.j] = ord.temp1

            ord.temp2 = rdist[ord.i]

```

```

        rdist[ord.i] = rdist[ord.j]
        rdist[ord.j] = ord.temp2

        ord.temp3 = xx[ord.i]
        xx[ord.i] = xx[ord.j]
        xx[ord.j] = ord.temp3

        ord.temp4 = yy[ord.i]
        yy[ord.i] = yy[ord.j]
        yy[ord.j] = ord.temp4

        ord.temp5 = rotazione[ord.i]
        rotazione[ord.i] = rotazione[ord.j]
        rotazione[ord.j] = ord.temp5

    END

END

END

```

Listing 3: Seleziona Oggetti

```

limitdist2obj = 50
limitdistcentre = 100 ;limite distanza dal centro della
    flb
limitdistboard = 200 ;limite distanza dal bordo della flb

CALL search()

;l'istruzione seguente serve per uscire dal while del
    programma principale se non trovo in un giro completo
    nessun oggetto

IF (instance_count == 0) THEN

    t = t+1

    IF (t == 3) THEN

        k = 1

    END

ELSE

    t = 0

END

```

```

count = 0 ;contatore per le posizioni corrette degli
oggetti (li voglio sempre ordinati)
elementi = 0

;trova oggetti da non prendere

IF (instance_count == 1) THEN

    count = count+1

    IF (rdist[count] < limitdistcentre) OR (rdist[count]
        > limitdistboard) THEN

        obj_no[count] = rdist[count] ;oggetto non
            prendibile

    ELSE

        obj_no[count] = 0 ;oggetto prendibile

    END

END

FOR i = 1 TO instance_count-1

    count = count+1

    FOR l = i+1 TO instance_count

        dist2obj = SQRT(SQR(xx[i]-xx[l])+SQR(yy[i]-yy[l]))
            ) ;distanza fra due oggetti

        IF (dist2obj < limitdist2obj) THEN

            obj_no[count] = rdist[i]
            count = l
            obj_no[count] = rdist[l]
            count = i
            elementi = l

        ELSE

            IF (i == 1) AND (l == i+1) THEN ;serve
                solo nel caso il primo oggetto è
                prendibile

                obj_no[i] = 0
                obj_no[l] = 0

            ELSE

```

```

        obj_no[l] = 0      ;oggetto l è prendibile

    END

END

;per verificare l'ultimo oggetto

IF (rdist[instance_count] < limitdistcentre) OR (
    rdist[instance_count] > limitdistboard) THEN

    obj_no[instance_count] = rdist[instance_count
    ]

END

END

;per verificare gli oggetti da i a i-1

IF (rdist[i] < limitdistcentre) OR (rdist[i] >
    limitdistboard) THEN

    obj_no[count] = rdist[i]

END

END

;oggetti effettivi che posso prendere

contatore = 0

FOR i = 1 TO instance_count

    IF (rdist[i] <> obj_no[i]) AND (vang[i] < 120) AND (
        vang[i] > 30) THEN ;l'istruzione dopo l'and è per
        fare una finestra di foto

        contatore = contatore+1
        vettdist[contatore] = rdist[i]
        vettang[contatore] = vang[i]
        vettrotaz[contatore] = rotazione[i]
        vettxx[contatore] = xx[i]
        vettyy[contatore] = yy[i]

    END

END

FOR i = 1 TO contatore

```

```

    rdist[i] = vettdist[i]
    vang[i] = vettang[i]
    rotazione[i] = vettrotaz[i]
    xx[i] = vettxx[i]
    yy[i] = vettyy[i]
    TYPE ""
    TYPE "x:", rdist[i]
    TYPE "alfa:", vang[i]
    TYPE "rotazione:", rotazione[i]
    TYPE "xx:", xx[i], " yy:", yy[i]

END

IF (indice.test > 1) AND (contatore <> 0) THEN

    foto = foto+1          ;solo quelle con oggetti
                          prendibili

END

```

Listing 4: Registro degli Oggetti

```

;costruisco una matrice che salva gli oggetti foto per foto

IF (indice.test == 1) THEN

    contatore = 0

ELSE

    regcontnuovo = cont.succ+contatore

END

IF (indice.test > 1) THEN

    FOR iva = cont.succ+1 TO regcontnuovo

        rdisto[iva] = rdist[iva-cont.succ]
        vango[iva] = vang[iva-cont.succ]
        rotazioneo[iva] = rotazione[iva-cont.succ]
        absango[iva] = absang
        fotoo[iva] = foto
        xx[iva] = xx[iva-cont.succ]
        yy[iva] = yy[iva-cont.succ]
        TYPE "iva:", iva
        TYPE "registro.dist:", rdisto[iva]
        TYPE "registro.ang:", vango[iva]
        TYPE "registro.foto:", fotoo[iva] ;foto solo
            quelle con oggetti prendibili
        TYPE "registro.rotazione:", rotazioneo[iva]
    
```



```
END

END

IF (indice.test > 1) THEN

  FOR i = cont.succ+1 TO regcontnuovo
    m[1,i] = foto ;o[i]
    m[2,i] = rdist[i-cont.succ] ;rdisto[i]
    m[3,i] = vang[i-cont.succ] ;vango[i]
    m[4,i] = absang ;o[i]
    m[5,i] = rotazione[i-cont.succ]
    m[6,i] = xx[i-cont.succ]
    m[7,i] = yy[i-cont.succ]
  END
END

IF (indice.test > 1) THEN

  cont.ctrl.foto = cont.succ
  cont.succ = cont.succ+contatore

END
```


BIBLIOGRAFIA

- [1] Adept. URL <http://www.adept.it>.
- [2] ARS. URL <http://www.flexibowl.it>.
- [3] Chiara Passarini. Programmazione e Ottimizzazione di un Sistema Robotizzato per Applicazioni di "Pick & Place".
- [4] Giulia Tomaello. Calibrazione di una Cella per la Presa di Oggetti in Movimento da un Nastro Trasportatore Circolare.

