POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Meccanica



Tesi di Laurea Magistrale

Applicazione di un sistema di una camera di profondità per manipolatori mobili

Relatori:

Candidato:

Prof. Giuseppe QUAGLIA

Lorenzo BAGLIERI

Ing. Luca CARBONARI

Anno Accademico 2020/2021

Sommario

Italiano

In questo documento di tesi viene presentata un'applicazione di robotica collaborativa. L'obiettivo perseguito da questo elaborato è in linea con l'attuale stato dell'arte nel quale la robotica si prefigge di svolgere compiti di assistenza, di supporto o sostituivi all'operato umano. La finalità del lavoro di tesi consiste nello stabilire un'interazione fra un braccio robotico a 6 gradi di libertà ed una camera di profondità equipaggiata con un sensore lidar affinché si possa condurre l'organo terminale del braccio robotico nel posto e con la posa ottenuta tramite i dati raccolti ed elaborati dalla camera di profondità. Per la pianificazione del moto del braccio robotico è stato utilizzato il pianificatore di movimenti OMPL tramite il pacchetto MoveIt presente fra gli altri disponibili in ROS. Grazie ad esso è possibile condurre la posa dell'organo terminale dalla posizione attuale alla posizione specificata. Inoltre è in grado di pianificare traiettorie che evitano gli ostacoli specificati nello spazio di lavoro e pianificano il movimento dei giunti secondo una legge del moto nota di volta in volta e che tiene conto dei vincoli cinematici sia dei giunti sia dei vincoli dello spazio esterno. I dati forniti dal sensore lidar della camera sono utilizzati per ottenere le distanze, le posizioni e le pose degli oggetti che vengono inquadrati. Per assicurarsi della bontà dei dati acquisiti è stata svolta una campagna di test sulle posizioni catalogate dalla camera per conoscere la tolleranza con cui lavora. La matrice RGB acquisita dal sensore video della camera di profondità è stato post-processato tramite l'utilizzo di OpenCV, una libreria di visione artificiale, che consente di riconoscere con un un certo grado di affidabilità un determinato set di oggetti tra tutti quelli che vengono inquadrati. La commistione di tutte le informazioni estrapolate dalla camera sono usato in questa modalità: si riconosce l'oggetto target e viene calcolata posizione e posa rispetto al sistema di riferimento della camera. Infine si è risolta l'interazione fra la camera di profondità e il braccio robotico tramite una matrice di calibrazione, la quale effettua una trasformazione dal sistema di riferimento del giunto in cui è ospitata la camera al sistema di riferimento di essa. Grazie a questa matrice di trasformazione è stato possibile trasformare nel sistema di riferimento solidale con

il robot le informazioni di distanza e posa acquisite dalla camera. In ultima analisi sono state svolti dei dimostrativi sperimentali per validare la bontà e la correttezza del grado di interazione raggiunto fra camera e robot.

English

In this master thesis is presented an application of collaborative robot. The aim pursued in this document is aligned with the actual robotic's state of art in which the presetted task are assistance, support or substituting of human actions. The aims of the works during the master thesis consists into establish an interaction between a robotic arm with 6 degree of freedom and a depth camera in orther that is possible to lead the robot's end-effector in the place and with the pose obtained from the data extrapoleted by the depth camera. For the move planning of the robotic arm it is used the OMPL planner through the MoveIt package present among the ROS' package. With it is possible move the end-effector from the actual pose to the desired position and orientation. Furthermore this package has the peculiarity of planning trajectories that ensure collision avoidance of the specified obstacle in the work space and plan the joint movements in order to follow a certain law of motion that respects the kinematic constraints both of the joints and of the free work space. The depth sensor of the depth camera is used to acquire information about distance, posistion and orientation of the framed objects. To ensure of the reliability of the acquired data a test campaign was carried out on the cataloged position by the camera to know the tolerance with which the camera works. The RGB matrix acquired from video sessor of depth camera was post-processated thanks the use of OpenCV, an artificial vision library, that allows to recognise with a certain degree of reliability a known set of objects among all that have been framed. The mixture of all these information extracted from the depth camera have been used in this way: once identified the target object and calculated position and orientation respect the camera reference system. Finally thanks to using a calibration matrix was possible put in contact the depth camera and the robotics arm, the matrix is used to obtain a transformation from the reference system of the joint who holds the camera to camera reference system. Thank to this transformation matrix it was possible to transfrom in the reference system solidal to the robot the information of distance and orientation acquired by the camera. At the end have been carried out some experimental demostratives to check the accuracy of the degree of interaction reached between camera and robot.

Ringraziamenti

Ringrazio il professor Giuseppe Quaglia per avermi dato l'opportunità di studiare su questa tematica così affascinante e innovativa. Inoltre lo ringrazio per il proficuo e costante supporto e scambio di opinioni durante i mesi della tesi.

Ringrazio Luca e Luigi per il loro aiuto, i loro consigli e l'esperienza che mi hanno trasmesso durante lo svolgimento dei vari compiti e dei diversi step che si sono susseguiti durante tutto il processo di ricerca e scrittura. Li ringrazio inoltre per il bel rapporto di amicizia che si è instaurato fra di noi nei vari giorni in cui ci siamo ritrovati fianco a fianco.

Ringrazio Andrea, Giovanni e Paride i quali hanno indubbiamente contribuito nella mia crescita formativa grazie ai consigli e scambi di pareri che si sono susseguiti di giorno in giorno. Anche a loro va un affettuoso pensiero per il sentimento di amicizia che si è creato fra le innumerevoli settimane in cui siamo stati insieme.

Infine ringrazio i miei familiari, affetti ed amici i quali non hanno mai fatto mancare amore, affetto, amicizia, supporto e sprono sia in questi mesi in cui sono stato impegnato con la tesi, sia in tutti gli anni universitari che ho trascorso.

Indice

El	enco	delle tabelle	VI
\mathbf{E} l	enco	delle figure	VIII
1	Intr	roduzione	1
	1.1	Classificazione ed evoluzione dei bracci robotici	3
	1.2	Stato dell'arte dei manipolatori articolati	7
		1.2.1 Applicazioni industriali	8
		1.2.2 Applicazioni sanitarie	13
		1.2.3 Applicazioni di assistenza	15
	1.3	Classificazione ed evoluzione delle camere di profondità	19
	1.4	Stato dell'arte delle applicazioni per camere di profondità	21
	1.5	Descrizione del sistema in analisi	22
2	Mai	nipolatore Articolato	24
	2.1	Descrizione del manipolatore articolato	24
		2.1.1 Giunti e link	25
		2.1.2 Sensoristica	27
		2.1.3 Spazio di lavoro	30
	2.2	Cinematica del manipolatore articolato	32
		2.2.1 Cinematica diretta di posa	32
		2.2.2 Cinematica inversa di posa	34
		2.2.3 Cinematica diretta di velocità	38
		2.2.4 Dualità cineto-statica	39
	2.3	Interfaccia software con il manipolatore articolato	40
		2.3.1 Logica dell'interfaccia con il manipolatore articolato	41
		2.3.2 Strumento ROS per la movimentazione: MoveIt	45
3	Can	nera di profondità	50
	3.1	Descrizione della camera di profondità	50
		3.1.1 Sistema di riferimento della camera	51

	3.1.2	Campo visivo della camera	52
	3.1.3	Funzionamento del sensore di profondità	53
3.2	Interfa	ccia software con la camera di profondità	55
	3.2.1	Inizializzazione della camera di profondità	56
	3.2.2	Descrizione delle funzioni principali	59
3.3	Test p	reliminari	64
	3.3.1	Set-up sperimentale dell'ambiente per il test	65
	3.3.2	Test sperimentali e risultati ottenuti	67
	3.3.3	Commento dei risultati ottenuti	71
Imp	lement	tazione del sistema: manipolatore articolato e sistema	
di v	isione		73
4.1	Definiz	zione del problema	73
	4.1.1	Confronto fra camera eye-in-hand e eye-to-hand	74
	4.1.2	Configurazione adottata	75
4.2	Proced	lura di calibrazione della camera	77
	4.2.1	Strumenti utilizzati per la calibrazione	78
	4.2.2	Processo di calibrazione	80
	4.2.3	Algoritmo di ricerca della matrice di calibrazione	81
4.3	Analisi	i critica della procedura implementata	86
Test	sperii	mentali dimostrativi del sistema studiato	88
5.1			88
5.2	Analisi	i delle prove svolte	91
	5.2.1	Prima prova: afferrare un oggetto con input di alto livello	
		definito dall'utente	91
	5.2.2	Seconda prova: afferrare un oggetto con procedura automatica	92
5.3	Presen	tazione delle prove proposte	93
	5.3.1	Effettuare un pick-and-place con un oggetto riconosciuto	
		dall'utente o automaticamente	93
	5.3.2	Utilizzare una pulsantiera con input di alto livello definito	
		dall'utente o in maniera automatica	94
Con	clusior	ne	95
bliog	rafia		97
	3.3 Imp di v 4.1 4.2 4.3 Test 5.1 5.2	3.1.3 3.2 Interfa 3.2.1 3.2.2 3.3 Test p 3.3.1 3.3.2 3.3.3 Implement di visione 4.1 Definiz 4.1.1 4.1.2 4.2 Proced 4.2.1 4.2.2 4.2.3 4.3 Analis Test sperif 5.1 Obiett 5.2 Analis 5.2.1 5.2.2 5.3 Presen 5.3.1 5.3.2	3.1.3 Funzionamento del sensore di profondità 3.2 Interfaccia software con la camera di profondità 3.2.1 Inizializzazione della camera di profondità 3.2.2 Descrizione delle funzioni principali 3.3 Test preliminari 3.3.1 Set-up sperimentale dell'ambiente per il test 3.3.2 Test sperimentali e risultati ottenuti 3.3.3 Commento dei risultati ottenuti 3.3.3 Commento dei risultati ottenuti Implementazione del sistema: manipolatore articolato e sistema di visione 4.1 Definizione del problema 4.1.1 Confronto fra camera eye-in-hand e eye-to-hand 4.1.2 Configurazione adottata 4.2 Procedura di calibrazione della camera 4.2.1 Strumenti utilizzati per la calibrazione 4.2.2 Processo di calibrazione 4.2.3 Algoritmo di ricerca della matrice di calibrazione 4.2.3 Analisi critica della procedura implementata Test sperimentali dimostrativi del sistema studiato 5.1 Obiettivo delle prove sperimentali 5.2 Analisi delle prove svolte 5.2.1 Prima prova: afferrare un oggetto con input di alto livello definito dall'utente 5.2.2 Seconda prova: afferrare un oggetto con procedura automatica 5.3 Presentazione delle prove proposte 5.3.1 Effettuare un pick-and-place con un oggetto riconosciuto dall'utente o automaticamente 5.3.2 Utilizzare una pulsantiera con input di alto livello definito dall'utente o in maniera automatica

Elenco delle tabelle

2.1	Tabella riassuntiva delle lunghezze dei vari link del manipolatore	
	articolato	27
2.2	Parametri di Denavit-Hartemberg con cui si caratterizza il passaggio	
	fra i giunti del robot Kinova Gen3 Lite	35
3.1	Accuratezza e deviazione standard forniti dal costruttore	64
3.2	Sunto delle distanze utilizzate per spaziare la griglia di nove punti	
	usata le la validazione della camera	65
3.3	Sintesi delle distanze prescelte con relativa impostazione del sensore	
	di profondità e tipo di griglia di punti utilizzata per le acquisizioni .	68
3.4	Sintesi dei valori di accuratezza e deviazione standard ottenuti dalla	
	misura della griglia di punti con la camera	72

Elenco delle figure

1.1	Rappresentazione grafica dei Sustainable Development Goals secondo	
	l'ONU[4]	3
1.2	Cobot che lavora a stretto contatto con l'operatore umano	6
1.3	Robot non collaborativo con la struttura metallica grigliata che	
	delimita la sua area di lavoro	6
1.4	Rappresentazione dei diversi gradi di collaborazione che un robot	
	può offrire[5]	7
1.5	Applicazione industriale in abito di sicurezza, EOD and HazMat	9
1.6	Applicazione industriale in abito di test, mantenimento e ispezione .	10
1.7	Braccio robotico con applicazioni nella logistica	11
1.8	Robotica applicata nel campo dell'agricoltura	12
1.9	Applicazioni fotovoltaiche robotiche	13
1.10	Robot applicato in sala operatoria con finalità chirurgiche	14
1.11	Robot con applicazione nell'industria farmaceutica	15
1.12	Robot per l'assistenza montato su carrozzina	16
1.13	Robot per l'assistenza montato su tavolo	18
1.14	Braccio e struttura robotica per mangiare	18
1.15	Principio di funzionamento delle camere di profondità[12]	19
1.16	Evoluzione delle camere di profondità[12]	20
1.17	Manipolatore articolato utilizzato in questo lavoro di tesi	23
1.18	Camera di profondità utilizzata in questo lavoro di tesi	23
2.1	Immagine con nomenclatura dei vari link che compongono il mani-	
	polatore articolato	26
2.2	Dimensioni nominali del robot in millimetri	27
2.3	Esempi di diverse posizioni assumibili dal manipolatore per afferrare oggetti differenti	30
2.4	Rappresentazione di una sezione bidimensionale dello spazio di lavoro	90
۷.٦	nominale in millimetri	31
2.5	Sistema di riferimento e nomenclatura per la risoluzione della cine-	IJΙ
۵.0	matica diretta secondo la convenzione di Denavit-Hartemberg	34
	manda diretta secondo la convenzione di Denavit-Hartemberg	94

2.6	Rappresentazione del funzionamento della convenzione di Denavit- Hartemberg che permette di descrivere un sistema di rifermento i	
2.7	rispetti al sistema di riferimento i-1 [20]	35
	matica inversa secondo la convenzione di Denavit-Hartemberg	37
2.8	Schematizzazione del processo di pianificazione del moto	45
2.9	Logica di funzionamento del pianificatore di moto	46
2.10	Esempio di pianificazione dei movimenti tramite il pianificatore OMPL	47
	Esempio di pianificazione dei movimenti tramite il pianificatore PILZ Esempio di pianificazione dei movimenti tramite il pianificatore PILZ	49
	con trapezio degenerato in triangoli	49
3.1	Raffigurazione delle dimensioni più importanti della camera	51
3.2 3.3	Raffigurazione del sistema di riferimento della camera	52
3.3	di profondità	53
3.4	Schematizzazione dell'effetto del tipo di superficie sulla riflessione	
	del segnale ad infrarosso emesso	54
3.5	Esempio di un sistema di riferimento posto su un ArUco Marker	63
3.6	Esempio di una bottiglia riconosciuta tramite OpenCV	63
3.7	Schema grafico della disposizione degli oggetti atti ad acquisire le	
	misure per validare la camera	66
3.8	Rappresentazione della griglia di nove punti usati per la misura	
	dell'accuratezza della camera	66
3.9	Immagini tratte dalla prova sperimentale dove si apprezza il set-up	67
3.10	sperimentale	67
5.10	m, in modalità LONG_RANGE e con la griglia dei punti grande	69
3 11	Rappresentazione del punto medio e della deviazione standard di	0.0
0.11	tutte le misure effettuate fra i 0.5 e i 1.5 m , in modalità LONG_RANGE	
	e con la griglia dei punti grande	70
3.12	Rappresentazione della posizione media della distanza rispetto alla	
	camera della griglia di punti acquisiti. Configurazione con griglia	
	grande, a $0.5 m$ dalla camera e preset LONG_RANGE	70
3.13	Rappresentazione della deviazione standard della distanza rispetto	
	alla camera della griglia di punti acquisiti. Configurazione con griglia	
	grande, a 0,5 m dalla camera e preset LONG_RANGE	71
4.1	Schema comparativo fra una sensore visivo montato in configurazione	
1.1	eye-to-hand ed un sensore visivo montato in configurazione eye-in-	
	hand rispetto al medesimo manipolatore articolato	75

4.2	Immagine dove è mostrata la posizione della camera rispetto all'or-	
	gano terminale del manipolatore articolato	76
4.3	Immagine dove è mostrata l'intera configurazione del robot	77
4.4	Schema del tabellone impiegato per la calibrazione della camera, viene esplicitato il sistema di riferimento del robot e del Marker usato come target. In grigio è rappresentata l'area coperta dalle misure di posa effettuate per la calibrazione	79
4.5	Rappresentazione grafica delle matrici presenti nel processo di ricerca della matrice di calibrazione: la matrice di posa del giunto 5, la matrice di calibrazione, la matrice di posa del Marker rispetto alla	0.0
4.0	camera e rispetto al braccio	82
4.6 4.7	Legenda dei vari blocchi utilizzati nel diagramma di flusso Schema di flusso logico che rappresenta il funzionamento dell'algoritmo di ricerca del minimo obiettivo al fine di trovare la matrice di calibrazione	83 85
5.1	Schematizzazione del funzionamento del sistema oggetto di studi: nello schema viene esplicata come avviene l'interazione fra la camera di profondità e il manipolatore articolato	89
5.2	Sequenza di immagini che mostra la manipolazione della bottiglia. la sequenza da a) a c) è rappresentativa della procedura con input dall'utente, mentre la sequenza da a) a d) è rappresentativa della procedura automatica. L'immagine a) mostra la posizione reciproca fra organo terminale del manipolatore, bottiglia da manipolare e	
	tavolo che funge da ostacolo	92

Capitolo 1

Introduzione

La richiesta e le applicazioni robotiche nei paesi più industrializzati si incrementano di anno in anno nei diversi settori della vita e della produttività. La robotica più di avanguardia negli ultimi anni è la robotica collaborativa la quale punta ad una collaborazione uomo-macchina attiva dove il robot sopperisce alle mancanze umane e viceversa. D'altro canto la robotica tipica del processo di automazione è caratterizzata da un braccio robotico che svolge azioni al posto dell'essere umano, ma è rinchiuso in uno spazio limitato e interagisce solo con i pochi oggetti che trova nel suo spazio di azione.

Due settori emergenti per le applicazioni della robotica sono l'assistenza e la sanità nei quali si assiste ad una progressiva domanda di personale per ottemperare alle richieste di assistenza alle persone, sia presso il domicilio sia in sede ospedaliera. Considerato il trend crescente di impiego di questi robot, lo sviluppo di una robotica sanitaria appare indispensabile nei paesi occidentali poiché la quantità di pazienti che ha bisogno di assistenza e di un monitoraggio continuo è in costante aumento. Correlato all'aumento di pazienti c'è anche un aumento di prodotto interno lordo da investire in ambito medico e di personale sanitario da reperire per fare fronte a queste esigenze[1]; d'altronde entrambe le risorse non sono facili da reperire subito ed in quantità massive, come già è emerso in questi mesi di lotta contro il Covid-19. Secondo l'autore dell'articolo precedente[1], l'ausilio fornito dai robot e dai cobot¹ può essere incisivo per riuscire a garantire una giusta assistenza a tutti in ambito sanitario. Infatti grazie a robot e cobot si possono assistere e monitorare molte più persone contemporaneamente aiutando anche il paziente superare eventuali disabilità grazie ad un robot dedicato per questo tipo di applicazioni.

¹Un cobot o co-robot (derivante da "collaborative robot") è un robot concepito per interagire fisicamente con l'uomo in uno spazio di lavoro Ciò trasmette un contrasto della maggior parte dei robot industriali adottati fino al 2008, i quali erano progettati per operare in maniera autonoma o con una guida limitata e protetti da barriere.

In ambito produttivo, sia nel settore primario[2] che nel secondario[3], si rende sempre più indispensabile l'impiego di robot per svolgere lavori che richiedono una manodopera non specializzata la quale a volte manca (a causa del calo demografico dei paesi occidentali) a volte non è economicamente sostenibile pagarla e in altre applicazioni non è etico impiegarla a causa dei rischi collegati alla mansione proposta. Grazie ai cobot si possono impiegare meno operai per svolgere le stesse mansioni con una maggiore efficienza. Questo tipo di approccio è classico dell'Industria 4.0 nel quale si predilige l'impiego di pochi addetti altamente specializzati che vengono assisti dai cobot nel svolgere la loro mansione, piuttosto che stipendiare tanti operai poco specializzati che svolgono il lavoro del robot.

È importante notare che i temi di assistenza in cui si possono impiegare i cobot, sia in ambito sanitario che in ambito produttivo, vanno a ricoprire le tematiche tre, nove e dieci dei Suistainable Developement Goals delle Nazioni Unite[4] mostrati in Figura 1.1, ovvero:

- Good Helath and Well-Being;
- Industry, Innovation and Infrastructure;
- Reduced Inequalities.

In particolare l'assistenza di un braccio robotico sul posto di lavoro aiuta ad avere condizioni di lavoro più agevoli e confortevoli ed anche più sicure in base a quale compito il robot va ad affiancare o sostituire l'essere umano. Nell'ambito dell'assistenza il braccio aiuta sicuramente a ridurre le ineguaglianze di persone non autosufficienti e/o anziane che possono diminuire il loro bisogno di aiuto e la dipendenza da un altro essere umano per compiere determinate azioni, traendo dunque un grosso vantaggio da questo tipo di applicazione robotica.

Dunque lo scopo di questa tesi è produrre un prototipo di un sistema robotico che possa essere applicato sia in ambito sanitario che produttivo poiché i due sono due facce della stesso prodotto. Infatti è molto interessante pensare ad un generico braccio robotico equipaggiato con una stereocamera, che identifica gli oggetti nel raggio d'azione del robot restituendo al robot le coordinate di tali oggetti. Inoltre un AGV², che ne estende il raggio di azione spostando il braccio robotico sul suolo.

²AGV è l'acronimo per Automated/Automatic Guided Vehicle (dall'inglese: Veicolo a guida automatizzata/automatica) e identifica dei veicoli a guida autonoma per la movimentazione di prodotti all'interno di uno spazio chiuso. Esistono comunque anche veicoli atti a lavorare all'esterno, anche se molto meno utilizzati.



Figura 1.1: Rappresentazione grafica dei Sustainable Development Goals secondo l'ONU[4].

1.1 Classificazione ed evoluzione dei bracci robotici

I robot sono stati sviluppati a partire dall'automazione industriale che nel XX secolo ha sicuramente mostrato i suoi vantaggi, per citarne alcuni: affidabilità, velocità e ripetitività. Una prima possibile suddivisione dei manipolatori robotici può seguire le funzioni per cui sono stati realizzati e quindi impiegati:

- Robot per applicazioni di automazione: dove il robot sostituisce completamente il lavoro di un essere umano. Tipicamente si tratta di lavori alienanti, logoranti e/o rischiosi, inoltre il robot va ad occuparsi di un compito semplice e ben preciso, non riesce ad eseguire compiti con movimenti complessi, evitare gli ostacoli o percepire che oggetti lo circondano.
- Robot per applicazioni collaborative: sono l'evoluzione dei robot per l'automazione, dotati di molti più sensori per aumentare le informazioni ricevute ed elaborate dal robot e quindi la complessità delle azioni che può compiere. Questi robot lavorano a stretto contatto con un operatore, percepiscono la posizione gli oggetti che devono manipolare, gli ostacoli e la posizione dell'operatore riuscendo a prevedere ed evitare le collisioni.

In sintesi la differenza tra i due tipi di robot è la dotazione elettronica ed informatica che ne determina la capacità di elaborare informazioni e di comprendere

dove sono gli umani, quali sono gli oggetti da manipolare e quali da evitare. Il cobot, Figura 1.2, è corredato da sensori atti a recepire il contatto con altri oggetti e persone, inoltre grazie anche a sensori visivi è in grado di prevedere il contatto e mettere in atto pratiche di collision avoidance. Nonostante ciò i cobot di grosse taglie non esistono poiché le elevate masse e le elevate inerzie in gioco non garantirebbero l'incolumità umana in caso di urti. Mentre il robot in Figura 1.3 è utilizzabile solo in ambienti chiusi e protetti dove si ha la certezza che nulla di imprevisto ostacoli i movimenti del robot, i quali sono stati pianificati e definiti in precedenza. Essendo ambienti protetti e privi di esseri umani non si hanno problemi legati all'incolumità umana e quindi totale libertà sulla taglia dei robot.

Esistono diversi gradi di collaborazione fra robot ed operatori, il discriminante fra i vari gradi è rappresentato dai sensori in dotazione al robot come illustrato in *Figura 1.4*. Di seguito sono elencati brevemente gli insiemi in cui è possibile raggruppare i robot con le loro principali caratteristiche:

- Fenced Robot: il robot viene posto in un ambiente chiuso e protetto affinché si abbia la certezza che mentre opera non urti nessun oggetto o essere umano che si trova di passaggio. Sul braccio non sono installati sensori per verificare la presenza o meno di corpi estranei nella traiettoria dei suoi movimenti. La traiettoria viene programmata a priori e tipicamente è sempre la stessa. Questi robot non sono flessibili poiché non utilizzabili in presenza di un ambiente di lavoro che varia o di persone nello spazio di lavoro. Il robot è cieco a ad ogni variazione e per evitare di andare a sbattere su persone o cose ne andrebbe riprogrammata la traiettoria.
- Coexistence: il robot può essere installato senza gabbia di sicurezza grazie ad opportuni sensori di sicurezza che rallentano o fermano il moto se un oggetto o una persona entra nella zona operativa del robot. Questo tipo di robot ha con sé solo dei sensori adeguati al riconoscimento delle persone o degli oggetti estranei nel suo spazio di lavoro. Tuttavia non è sufficientemente evoluto per riprogrammare la traiettoria nel caso in cui un impedimento si trovi in essa ed allora non può fare altro che rallentare o bloccare il moto finché la traiettoria non si libera nuovamente.
- Sequential Collaboration: il robot, oltre a riconoscere la presenza umana o di un oggetto che ne ostruisce la traiettoria nel suo spazio di lavoro, condivide lo spazio di lavoro con gli esseri umani svolgendo degli obiettivi non in contemporanea con l'operatore. Questo tipo di robot riesce a garantire più flessibilità del precedente adattandosi ai cambiamenti dello spazio di lavoro dovuti alla presenta di una persona, tuttavia non è ancora abbastanza equipaggiato di sensori da potersi interfacciare direttamente con una persona.

- Cooperation: il robot viene controllato in corrente (proporzionale alla coppia che il robot esprime) affinché si possa arrestare non appena i sensori di forza o di corrente percepiscono un segnale sopra una soglia di sicurezza prefissata. Robot ed operatore possono lavorare contemporaneamente a diversi oggetti nello stesso spazio di lavoro. La sicurezza in questo tipo di applicazione è molto elevata grazie sia ai sensori di coppia che monitorano la forza che il robot imprime a tutti i corpi con cui viene in contatto ed anche grazie ad altri sensori, come quelli visivi, che aiutano a predire il contatto e ad evitarlo. Tuttavia per massimizzare la sicurezza dell'applicazione è necessario limitare la taglia di questo tipo di robot, quindi limitando le masse e le inerzie in movimento, al fine di assicurare l'incolumità fisica della persona in un eventuale urto.
- Responsive Collaboration: il robot viene monitorato in corrente affinché si possa arrestare non appena i sensori di forza percepiscono un segnale sopra una soglia di sicurezza prefissata, robot ed operatore lavorano contemporaneamente allo stesso oggetto condividendo lo stesso spazio di lavoro. La sicurezza in questo tipo di applicazione è molto elevata grazie sia ai sensori di coppia che monitorano la forza che il robot imprime a tutti i corpi con cui viene in contatto ed anche grazie ad altri sensori, come quelli visivi, che aiutano a predire il contatto e ad evitarlo. Tuttavia per massimizzare la sicurezza dell'applicazione è necessario limitare la taglia di questo tipo di robot, quindi limitando le masse e le inerzie in movimento, al fine di assicurare l'incolumità fisica della persona in un eventuale urto.



Figura 1.2: Cobot che lavora a stretto contatto con l'operatore umano



Figura 1.3: Robot non collaborativo con la struttura metallica grigliata che delimita la sua area di lavoro

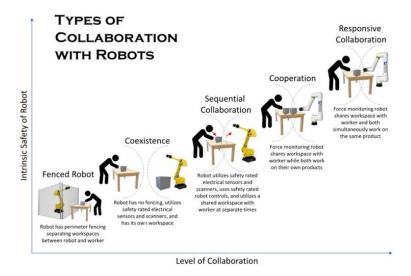


Figura 1.4: Rappresentazione dei diversi gradi di collaborazione che un robot può offrire[5].

1.2 Stato dell'arte dei manipolatori articolati

In questa sezione vengono presentate le soluzioni attualmente commercializzate e/o ideate inerenti bracci collaborativi. Grazie all'utilizzo di robot collaborativi si può sfruttare in pieno l'interazione uomo macchina e tutti i vantaggi che derivano dalla massimizzazione dell'interazione uomo macchina. Applicazioni di maggiore interesse, poiché più simili al prototipo pensato per la tesi, consistono in bracci accoppiati a delle piattaforme autonome in maniera tale da poter estendere l'area di impiego del robot.

Di seguito verranno analizzate solo delle applicazioni che per portata, taglia, potenza espressa e potenza assorbita sono simili all'applicazione oggetto di questa tesi di laurea. Le informazioni sono state prese dalle pagine web di Kinova[6], KUKA[7], Staubli[8], Panasonic[9] e Kawasaki[10]. I bracci robotici possono svolgere molteplici compiti con un equipaggiamento simile. Infatti variando di poco gli accessori e i sensori in dotazione è possibile passare da una applicazione ad un'altra. Un conciso elenco delle possibili applicazioni è:

- Manipolatore autonomo;
- Manipolatore teleguidato;
- Manipolatore per assistenza al personale medico;

- Manipolatore per assistenza al paziente;
- Manipolatorie per il lavoro agricolo;
- Manipolatore rivolto all'industria e al settore secondario.

1.2.1 Applicazioni industriali

In questa sotto sezione sono stati raccolti alcuni esempi di robot con le loro rispettive applicazioni industriali di assistenza all'operato umano o di supplenza all'operato umano. Da questa raccolta sono state tralasciate le tipiche applicazioni robotiche nel settore secondario come i robot per l'industria metallica, automobilistica, alimentare che svolgono operazioni di saldatura, stampa, ispezione, verniciatura, lavorazioni puntuali sia di asportazione materiale grossolano che di finitura oltre che il trasporto di prodotti da un nastro ad un altro.

Security, EOD and HazMat

In Figura 1.5 è mostrato come esempio un braccio robotico fissato ad un AGV dotato di diversi sensori visivi al fine di svolgere compiti del tipo EOD³ e HazMat⁴. Grazie ai sensori visivi e all'AGV ha la possibilità di essere tele-operato o di operare in autonomia. I sensori di visione coadiuvati da software di AI riescono ad operare anche senza operatore, mantenendo l'addetto ad una distanza di sicurezza opportuna.

L'obiettivo dell'applicazione è quello di sondare ambienti pericolosi e svolgere compiti che metterebbero a rischio l'incolumità umana come per esempio la manipolazione ed il disinnesco di esplosivi, la manipolazione di corpi radioattivi, cancerogeni o altamente tossici. La piattaforma dell'AGV al quale è fissato il braccio robotico ha anche lo scopo di trasportare il materiale rischioso. Questo tipo di manipolatori sono sviluppati per assicurare un intervento umano nel controllo del robot, integrando l'esperienza e la conoscenza umana con le capacità del robot.

³EOD è l'acronimo inglese che sta per Explosive Ordnance Disposal, è un ramo delle forze dell'ordine che si occupa del disinnesco delle bombe e di ordigni simili (munizioni e ordigni artigianali) che possono minacciare l'incolumità dei civili. Sono inclusi anche bombe o armi chimiche, biologiche, incendiarie, radiologiche e nucleari.

⁴Hazmat è una abbreviazione che sta per "Hazardous Material", ovvero sostanze pericolose



Figura 1.5: Applicazione industriale in abito di sicurezza, EOD and HazMat

Test, mantenimento e ispezione

L'applicazione mostrata in Figura 1.6 è in grado di svolgere dei test, della manutenzione o delle ispezioni in maniera automatica. Se l'oggetto ha più di una parte su cui operare, o se sono presenti più oggetti su cui lavorare, il robot è comunque in grado di portare a termine il test. Inoltre il robot è capace di svolgere lo stesso compito in differenti tipi di ambienti e circostanze. Le peculiarità principali che rendono possibile tale applicazione risiedono in un hardware robusto ed un potente software, il quale permette di adattare e configurare il braccio robotico per compiere le azioni richieste (test, mantenimento e ispezione). Dal punto di vista dei sensori e della robotica il sistema non richiede delle specifiche tecniche molto elevate o ricercate, è sufficiente l'impiego dei classici bracci e dei classici sensori in uso nella robotica tradizionale.

L'obiettivo di questo braccio robotico è quello di effettuare senza interruzioni operazioni a basso o nullo valore aggiunto liberando manodopera e dunque aumentando la produttività. Si possono così impiegare gli operai per ottenere un'ottimizzazione delle performance tramite l'analisi dei dati oppure impiegando manodopera in compiti a più valore aggiunto.



Figura 1.6: Applicazione industriale in abito di test, mantenimento e ispezione

Logistica e Magazzini

L'esempio di applicazione mostrato in Figura~1.7 è un braccio robotico impiegato nella logistica. La logistica è un ambiente in cui attualmente regna l'automazione e la standardizzazione. I bracci robotici che attualmente lavorano nella logistica tipicamente eseguono solo operazioni predefinite come:

- pick and place;
- imballaggio;
- pallettizzazione;
- depallettizzazione;
- trasporto delle merci.

Mentre grazie all'introduzione della robotica ci si può approcciare ad una logistica più flessibile, adatta ad esempio per dei piccoli trasporti di tipologie merceologiche differenti.

L'uso bracci robotici collaborativi, quindi equipaggiati di sensori, è opportuno quando si lavora in ambienti non standard poiché riescono a garantire flessibilità come in questo caso d'immagazzinamento e di logistica ottimizzata. Questa tipologia

di bracci robotici è facilmente integrabile con un supporto mobile, come un AGV, il quale può aumentare la mobilità del braccio robotico. Inoltre tramite l'ausilio dei controllori è possibile offrire una più ampia possibilità di comando, di flessibilità ed una più ampia differenza rispetto alla classica automazione da logistica.



Figura 1.7: Braccio robotico con applicazioni nella logistica

Robotica agricola

In Figura 1.8 è rappresentata un'applicazione agricola per un robot, il quale tramite un AGV riesce a far spostare il braccio robotico lungo un campo in modo da automatizzare il lavoro delle serre e dei campi in modo da incrementare: l'efficienza, la quantità di prodotti e la qualità.

Le principali caratteristiche che un robot agricolo deve possedere affinché esplichi la sua funzione sono: un buon rapporto carico utile/peso del robot, un basso consumo di energia e un controllore integrato per poterlo usare sia autonomamente che teleguidato. L'AGV attualmente è la piattaforma che consente al sistema robotico di avere un funzionamento autonomo o teleguidato su piattaforma mobile. Con la tecnologia attuale è plausibile riuscire ad ottenere un buon robot agricolo poiché le caratteristiche sopra elencate ormai sono tipicamente presenti in molti bracci robotici in commercio. Infine è importante dotare il modulo robotico di un adeguato sensore visivo 2D o 3D per permettere un riconoscimento degli oggetti che lo circondano, i quali possono essere sia ostacoli che obiettivi.



Figura 1.8: Robotica applicata nel campo dell'agricoltura

Robotica fotovoltaica

In Figura 1.9 è mostrato un esempio tratto da una gamma completa di soluzioni robotiche specializzate per la produzione solare e fotovoltaica. I bracci robotici in oggetto possono essere a 4 o 6 assi a seconda dell'affidabilità, precisione, destrezza richiesta nei movimenti e produttività richiesti; la scelta degli assi influisce di conseguenza anche sui costi di installazione e produzione.

I robot fotovoltaici sono caratterizzati da alcune peculiarità tipiche di tutta la robotica grazie alle quali è possibile ottemperare ad alcune specifiche richieste in ambito fotovoltaico, come:

- la struttura rigida per il braccio robotico, la quale assicura una maggiore precisione e affidabilità nelle misure;
- il braccio robotico chiuso, il quale assicura la pulizia e la non contaminazione con elementi esterne;
- l'alta velocità, la quale assicura una elevata produttività;
- l'elevata precisione, la quale assicura il mantenimento degli standard di qualità con cui si vuole operare;

• il movimento fluido del robot, grazie al quale il robot riesce a muoversi e a compiere traiettorie con maggiore destrezza.

Le applicazioni fotovoltaiche per cui questi robot sono pensati e realizzati sono:

- produzione di lingotti e mattoni: tramite operazioni di manipolazione per il taglio, la misurazione, la rettifica e incollaggio;
- produzione di wafer e di celle solari: tramite operazioni di controllo qualità wafer, caricamento su banco bagnato, caricamento del rivestimento antiriflesso, caricamento serigrafia, smistamento e confezionamento delle cellule;
- produzione di moduli C-Si: tramite operazioni di movimentazione del vetro per lavatrice, applicazione Tabber e Stringer, posizionamento delle corde, montaggio e posizionamento della scatola di giunzione e montaggio di lampeggiatore e tester per modulo.



Figura 1.9: Applicazioni fotovoltaiche robotiche

1.2.2 Applicazioni sanitarie

Robot medico chirurgici

Si tratta di robot intuitivi, configurabili e pensati appositamente per lavorare a fianco di un chirurgo e del suo staff nella sala operatoria. Il vantaggio di avere un robot chirurgico in una sala operatoria in alcuni casi è notevole proprio per la differenza di impieghi che può svolgere rispetto ad un umano. Un esempio di tale applicazione è mostrata in Figura 1.10.

I robot che lavorano a fianco con i medici sono molto personalizzati, dipendono sia dall'applicazione che andranno a svolgere sia dalle esigenze dettate dal personale sanitario e dalla struttura sanitaria. Per questo motivo questi robot vengono sempre sviluppati a stretto contatto con l'equipe medica, personalizzandoli il più possibile. Le personalizzazioni eseguite su un robot con applicazioni mediche sono innumerevoli e possono riguardare il modus operandi dello staff medico, l'applicazione che il robot andrà a ricoprire oppure il tipo di aiuto che il robot presterà al personale sanitario. Talvolta i robot sono personalizzati per svolgere più operazioni insieme, soprattutto quelle semplici in modo da lasciare il personale sanitario ad occuparsi dei compiti che richiedono capacità critiche o doti umane. Le applicazioni in ambito medico in cui attualmente sono uso robot sono:



Figura 1.10: Robot applicato in sala operatoria con finalità chirurgiche

- Applicazioni speciali di riabilitazione;
- Chirurgia mini-invasiva;
- Radioterapia intraoperatoria;
- Migliore diagnosi del tumore al seno;
- Chirurgia ossea e trattamento osseo senza contatto;
- Chirurgia ortopedica;
- Sistemi per angiografia e sala operatoria ibrida;
- Assemblaggio e confezionamento di cartucce di criogas;
- Soluzioni robotiche per mantenere elevati standardi di igiene;
- Robot nella bonifica delle ossa femorali dell'anca per allotrapianti;

- Manipolazione delle protesi dentarie;
- Chirurgia estetica.

Robot farmaceutici

In Figura 1.11 è mostrato un esempio di un'applicazione farmaceutica della robotica. Nell'ambito farmaceutico le caratteristiche molto richieste sono: precisione, pulizia, ripetibilità e flessibilità. La giusta commistione di queste caratteristiche è la chiave per avere un più ampio successo economico ed organizzativo nella vendita.

Le applicazioni in cui l'utilizzo del robot è più diffuso sono: produzione di vaccini, produzione e ispezione di siringhe o altro materiale asettico, produzione e riempimento di flaconi di gel, ispezioni e applicazioni asettiche in generale.



Figura 1.11: Robot con applicazione nell'industria farmaceutica

1.2.3 Applicazioni di assistenza

L'assistenza è un ambito molto sentito in robotica poiché per il paziente può rappresentare il raggiungimento di alcuni standard di autonomia ed autosufficienza che gli sono stati preclusi a causa di qualche handicap momentaneo o permanente. In questo ambito più la disabilità da fronteggiare è complessa e più la sfida robotica si fa ardua. Ad esempio di seguito saranno mostrare alcune applicazioni industriali per la movimentazione degli arti, tuttavia esistono dei brevetti in fase embrionale di sviluppo che tentano di fare il medesimo movimento tramite l'utilizzo di un esoscheletro[11].

Le applicazioni di assistenza sono le più emergenti nell'ambito della robotica. É uno degli ambiti più innovativi della robotica, infatti le uniche soluzioni in commercio sono ancora dedicate a coadiuvare le persone al fine di superare le piccole disabilità. Tutto il resto invece è in fase sperimentale sia presso le università,

ma anche presso aziende che studiano come commercializzare nuove applicazioni di questo tipo.

É molto importate sottolineare come questo tipo di innovazione debba procedere con i giusti standard di sicurezza poiché sono robot che spesso operano direttamente con gli esseri umani muovendo attivamente i loro arti o manipolando oggetti vicino ad organi molto delicati, compito che attualmente rappresenta uno dei limiti tecnologici. Anche questo tipo di accorgimenti, dovuti e necessari, richiede molto sforzo e molto lavoro per creare dei protocolli di lavoro che non mettano a rischio l'incolumità del paziente in caso di errore, malfunzionamento o guasto del sistema robotico.

Braccio dinamico di supporto per utenti di carrozzine motorizzate

In Figura 1.12 è mostrato una braccio robotico per l'assistenza a persone non autosufficienti in carrozzina motorizzata. Le funzionalità di questo braccio robotico permettono di redistribuire fra paziente e braccio la forza necessaria per i moti più articolati, ciò consente all'assistito di poter compiere più movimenti in modo sicuro e preciso. I punti di forza di questa applicazione robotica sono rappresentati dall'essere un dispositivo leggero, compatto e progettato per utenti di carrozzina motorizzata. Riassumendo alcune caratteristiche su questo sistema si ha che:



Figura 1.12: Robot per l'assistenza montato su carrozzina

- Adattato per utilizzare l'alimentazione della carrozzina motorizzata;
- Utilizzabile sia da persone mancine che da persone destrorse;

- Elevata adattabilità alla forza del braccio e alla morfologia dell'utente;
- Segue il movimento naturale del braccio dell'utente;
- Ogni braccio ha una portata fino a 4.5 kg.

Braccio dinamico di supporto per applicazioni su tavolo

Lo scopo del braccio robotico mostrato in *Figura 1.13* è quello di far ottenere ai pazienti che lo usano l'autosufficienza e l'indipendenza nei compiti giornalieri. Le funzionalità di questo braccio robotico, analoghe al precedente, permettono di redistribuire fra paziente e braccio la forza necessaria per i moti più articolati, ciò consente all'assistito di poter compiere più movimenti in modo sicuro e preciso.

Le peculiarità di questo braccio robotico sono simili al precedente braccio robotico da carrozzina motorizzata, dunque è un braccio robotico leggero, compatto, facile da usare e facilmente trasportabile. Riassumendo alcune caratteristiche su questo sistema si ha che:

- Adattato per essere montato su ogni superficie piana;
- Utilizzabile sia da persone mancine che da persone destrorse;
- Manualmente aggiustabile secondo le esigenze personali e puntuali;
- Segue il movimento naturale del braccio dell'utente;
- Ogni braccio ha una portata fino a 5.5 kg.

Supporto robotico per mangiare

In Figura 1.14 è mostrato un braccio robotico il cui scopo è quello di aiutare le persone affette da disabilità nel ritrovare la piena autonomia ed indipendenza durante la consumazione dei pasti a condizione che l'assistito abbia la facoltà di masticare e deglutire da solo, senza bisogno di aiuto.

Il robot è pensato come un braccio sostitutivo a quello del paziente che prende il cibo da un contenitore solidale con la struttura del braccio stesso e lo imbocca all'assistito. Per portare a compimento questa operazione il robot è dotato di una opportuna sensoristica per essere consapevole di dove sia il volto e la bocca dell'assistito.

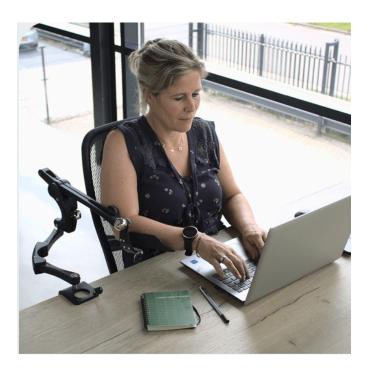


Figura 1.13: Robot per l'assistenza montato su tavolo



 ${\bf Figura~1.14:~Braccio~e~struttura~robotica~per~mangiare}$

1.3 Classificazione ed evoluzione delle camere di profondità

Questo tipo di camere è caratterizzato da un sensore RGB che svolge le classiche funzioni di una camera, ovvero acquisisce le varie immagini che vengono inquadrate dal sensore, e da un sensore di profondità il quale misura lo spazio presente fra il dispositivo e l'oggetto da misurare calcolando il tempo intercorso fra il segnale emesso e quello acquisito. Nello specifico il tempo viene calcolato tramite la misura dello sfasamento fra le due onde. Il sensore di profondità inoltre è in grado di acquisire la misura della distanza su una griglia di pixel equivalente a quella del sensore RGB, grazie a questa funzione si riesce ad allineare tramite software un'informazione all'altra per ottenere un'immagine con le varie informazioni di profondità al suo interno. In Figura 1.15 viene mostrata una rappresentazione schematica di come tale misura viene acquisita ed utilizzata al fine di ottenere le distanze.

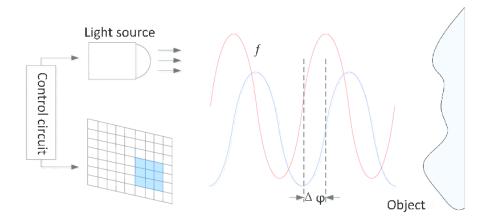


Figura 1.15: Principio di funzionamento delle camere di profondità[12]

L'invenzione della camere di profondità risale al 1977 quando per la prima volta è ideata una procedura che ha permesso di ottenere efficacemente la distanze utilizzando le informazioni provenienti dai segnali elettromagnetici emessi da una sorgente luminosa. Tuttavia finché questa tecnica non diventa più maneggevole ed efficace le applicazioni non sono state molteplici. Quando nel 1990 è stata inventata la tecnica lock-in CCD è stata possibile una più facile ricezione e campionamento dei segnali elettromagnetici ricevuti dal sensore. Da questo momento è stato possibile l'utilizzo delle onde elettromagnetiche per misurare le distanze fra gli oggetti e il sensore. Come si vede in Figura 1.16 accoppiare la misura delle distanze con l'acquisizione della matrice RGB è stato un obiettivo perseguito dagli albori di questa tecnologia. Dopo il 2010 sono comparsi sul mercato i primi dispositivi che

sono riusciti a conciliare efficacemente il sensore RGB con quello laser i quali hanno messo insieme efficacemente il funzionamento dei due sensori in spazi ridotti e con performance sempre più avanzate. Si prende come esempio una camera di ultima generazione che esplica le funzioni sopracitate, ovvero la Kinect di Microsoft. Nel datasheet[13] della camera si possono apprezzare le performance che riesce a garantire il sensore di profondità:

- un frame rate fino a 30 FPS;
- una risoluzione fino a 1024x1024 pixel;
- un'apertura del cono visivo fino a $120^{\circ} \times 120^{\circ}$;
- un range operativo di misura con estremi massimi compresi tra i 0.25 e i 5.46 m.

Le peculiarità della camera RGB sono paragonabili ad una qualsiasi ottica compatta presente attualmente sul mercato, con rapporti di visione pari a 16:9, full HD, 30 FPS.

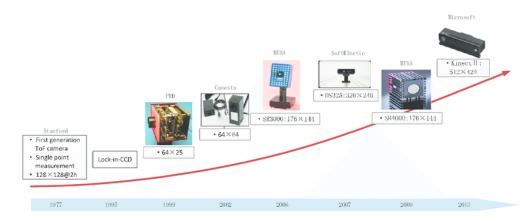


Figura 1.16: Evoluzione delle camere di profondità[12]

La classificazione che può essere fatta su questo tipo di camere è inerente solo al tipo di tecnologia con il quale vengono acquisite le misure, il quale può influenzare alcuni parametri statistici. I principali sistemi usati per l'acquisizione di questo tipo di dato sono:

- tramite un segnale pulsante;
- tramite un segnale continuo;
- tramite un numero pseudo-casuale e il rilevamento compresso.

1.4 Stato dell'arte delle applicazioni per camere di profondità

Le camere di profondità hanno principalmente lo scopo di acquisire un segnale visivo ed un segnale di distanza dunque accoppiare i due segnali in modo da produrre due matrici con la stessa disposizione dei pixel, ovvero in modo da avere i pixel delle due matrici allineati. Il vantaggio di avere i pixel delle due matrici allineati è che per ogni pixel si può conoscere il colore che lo caratterizza e la distanza fra il sensore e quell'oggetto.

Una volta acquisite le due matrici allineate si può pensare di utilizzarle per estrapolare informazioni utili allo scopo per cui si è istallata la camera di profondità. Le varie applicazioni che si possono ottenere con questo tipo di camere si realizzano sviluppando codici ed algoritmi che manipolano ed elaborano correttamente i dati che sono forniti dalla matrice RGB e dalla matrice di profondità.

A titolo di esempio sono riportate alcune applicazioni che sono state realizzate con le camere di profondità che da un lato rappresentano i vari ambiti a cui attualmente questa tecnologia si applica, dall'altro lato danno un'idea delle potenzialità di sviluppo di questa tecnologia:

- Algoritmi per il riconoscimento e la mappatura degli ambienti indoor[14]. Sono algoritmi molto utili in accoppiata ad algoritmi di guida autonoma e collision avoidance per una robotica mobile autonoma indoor.
- Algoritmi per il tracciamento e l'acquisizione del corpo umano [15]. Questo funzione viene effettuata tramite diverse camere diverse camere disposte sopra, sotto e ai lati del corpo umano. In particolare questo algoritmo è capace di costruire una mesh efficace delle superfici del corpo umano che sta inquadrando dando l'opportunita di ottenere un modello tridimensionale dello stesso.
- Algoritmi per il riconoscimento dei gesti delle mani[16]. In questo caso l'algoritmo è in grado di riconoscere alcune pose specifiche delle mani. Ad esempio grazie a questa capacità di riconoscimento è possibile utilizzare l'avvenuto riconoscimento di una gesture per comandare un altro sistema.
- Algoritmi per il riconoscimento della posa della testa di una persona[17]. Grazie a questo algoritmo è possibile riconoscere efficacemente la testa umana a prescindere dalla posizione in cui si trova.

Le applicazioni viste precedentemente sono solo un esempio di quanto si può fare con le camere di profondità, tuttavia come principio generale, allenando il giusto algoritmo di riconoscimento immagine si può ottenere il riconoscimento di qualsiasi caratteristica presente nell'inquadratura della camera.

1.5 Descrizione del sistema in analisi

Il sistema che questa tesi si prefigge di studiare è composto da un manipolatore articolato a sei gradi di libertà, si tratta del Kinova Gen 3 Lite mostrato in *Figura 1.17*, in accoppiata con esso viene usata la camera di profondità Intel RealSense L515, mostrata in *Figura 1.18*.

Si è studiata l'interazione tra questi due attori principali del sistema affinché potesse avvenire una esatta e corretta comunicazione di segnali fra i due dispositivi e si potessero manipolare gli oggetti riconosciuti e misurati tramite la camera. Per tale scopo si è suddiviso il lavoro di tesi in tre macro problemi:

- La pianificazione delle traiettorie e del moto del manipolatore articolato. L'obiettivo è quello di mandare il manipolatore nella posa specificata in input facendo anche in modo che eviti gli ostacoli che possono presentarsi fra la sua posa attuale e quella finale.
- Il riconoscimento degli oggetti e delle distanze. Utilizzare entrambi i sensori della camera di profondità sia per riconoscere gli oggetti da manipolare fra quelli inquadrati, sia per misurare la posa degli oggetti da manipolare rispetto alla camera.
- Il passaggio di informazioni fra camera e manipolatore, per far si che le informazioni di posa acquisite dalla camera siano utilizzabili dal manipolatore. É necessario che il manipolatore sia a conoscenza della posizione della camera rispetto al suo sistema di riferimento fisso (camera eye-to-hand) o rispetto ad uno dei suoi giunti (camera eye-in-hand).

Una volta ottenuta un'efficace interazione tra camera e manipolatore si ha un sistema capace di manipolare gli oggetti che vengono riconosciuti e di cui è possibile misurare la posa. Lo scopo di tale prototipo è quello di inserirsi nel settore ospedaliero o in quello dell'assistenza a persone abili o disabili in linea con i Sustainaible Developement Goals dell'ONU precedentemente menzionati. Inoltre porre il sistema di manipolatore e camera su una piattaforma mobile a guida autonoma aumenterebbe di molto le distanze fra gli oggetti che si possono manipolare, ciò è in linea con l'evoluzione dei vari sistemi di manipolatori, i quali non si limitano più a manipolare oggetti vicini ma anche lontani raggiungendo elevati gradi di autonomia ed integrazione con l'ambiente circostante. Pertanto lo scopo ultimo del lavoro è quello di prototipare un sistema in grado di manipolare un determinato campione di oggetti che lui stesso riconosce in un vasto ambiente di lavoro ausiliato da una piattaforma mobile a guida autonoma. Il prototipo così sviluppato può essere impiegato per alleviare il carico di lavoro nel settore industriale o nel settore ospedaliero e contemporaneamente fornire assistenza agli essere umani, pienamente abili o meno, sia in ambito sanitario che non.



Figura 1.17: Manipolatore articolato utilizzato in questo lavoro di tesi



Figura 1.18: Camera di profondità utilizzata in questo lavoro di tesi

Capitolo 2

Manipolatore Articolato

In questo capitolo viene approfondito il funzionamento del manipolatore articolato, sia da un punto di vista fisico e cinematico che da un punto di vista informatico ed elettronico, dove le due rappresentazioni sono complementari. Nella rappresentazione cinematica si approfondisce il motivo per cui il manipolatore articolato può compiere determinati movimenti, la spiegazione del movimento, i vincoli fisici e i sistemi di riferimento cartesiano che sono presenti. Il manipolatore articolato utilizzato in questo lavoro di tesi è il Kinova Gen3 Lite[18] dal quale manuale tecnico sono estratti i dati del robot riportati in questo capitolo.

2.1 Descrizione del manipolatore articolato

In questa sezione si approfondiscono e si dettagliano le specifiche tecniche del robot per una più accurata comprensione dei limiti e delle potenzialità. In particolare verranno approfonditi gli aspetti legati alle dimensioni del manipolatore articolato, all'attuazione del manipolatore articolato e ai sensori presenti sul braccio.

2.1.1 Giunti e link

Il manipolatore articolato è complessivamente composto da un primo manipolatore antropomorfo con polso sferico sulla sommità. I manipolatori antropomorfi sono composti da tre giunti rotoidali, come anche il polso sferico, per un totale di sei giunti rotoidali che vanno ad incernierare i sei link¹ che compongono il manipolatore mobile. Da ciò si deduce che il manipolatore articolato utilizzato possiede sei gradi di libertà. In Figura 2.1 è mostrato il manipolatore articolato con i vari attuatori che movimentano un link rispetto ad un altro.

I primi tre attuatori, che collegano tra di loro i primi tre link, formano la struttura del manipolatore antropomorfo la quale è caratterizzata dal primo asse di rotazione perpendicolare agli altri due. Per la spiccata somiglianza fra il manipolatore antropomorfo e un braccio umano il secondo giunto ed il terzo giungo vengono rispettivamente chiamati giunto di spalla e giunto di di gomito. I giunti quattro, cinque e sei, con i rispettivi attuatori, sono articolati tra di loro con altri 3 giunti rotoidali e fanno parte della struttura del polso. Un contributo importante dato dal polso sferico è il disaccoppiamento fra posizionamento e orientamento dell'organo terminale, difatti i tre giunti del braccio antropomorfo si occupano del posizionamento mentre i tre giunti del polso si occupano dell'orientamento. L'unica eccezione nel disaccoppiamento è presente nel bordo estremo dell'area di lavoro del robot sia interna che esterna dove l'orientamento del polso è soggetto ai vincoli fisici legati alle dimensioni dei link.

I parametri fisici che caratterizzano il manipolatore articolato in oggetto rappresentano un dato importante per realizzare calcoli e simulazioni sia a livello cinematico e sia a livello fisico. In *Figura 2.2* sono riportate le misure del robot dal quale sono state dedotte le lunghezze effettive dei vari link che compongono il manipolatore articolato e sono riportate in *Tabella 2.1*.

Dall'analisi dei giunti si nota come:

- il primo link con la base siano coassiali e di dimensioni confrontabili ed è
 permesso il movimento di rotazione intorno a se stessi come l'articolazione
 della spalla umana;
- il secondo link è il più lungo e corrisponde al link che svolge la funzione di braccio;
- terzo, quarto e quinto link hanno di nuovo dimensioni nuovamente confrontabili e si occupano di gestire i movimenti del polso garantendo la rotazione su assi perpendicolari.

¹Il link è un elemento del meccanismo che trasporta la cinematica da un elemento di accoppiamento al successivo. Nel caso di un manipolatore articolato è la porzione rigida del braccio presente fra un giunto attuato e il successivo

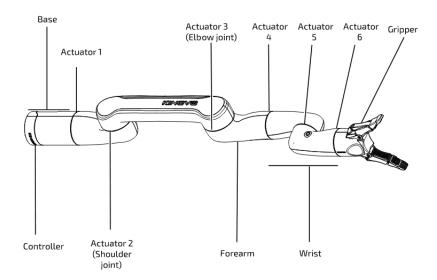


Figura 2.1: Immagine con nomenclatura dei vari link che compongono il manipolatore articolato

In Tabella 2.1 sono riportati i dati relativi agli attuatori montati sui vari link. Essendo gli attuatori montati sul giunto che collega due link è importante evidenziare che il primo attuatore collega la base al primo link e così via fino all'attuatore della pinza che non è collegato con nessuno. Si nota dalle dimensioni degli attuatori che i primi tre link muovendo gran parte della massa del manipolatore hanno una taglia più grande degli ultimi tre e quindi una potenza assorbita maggiore. Inoltre l'attuatore che collega primo con secondo giunto è il più grande di tutto il robot per venire incontro alla più elevata inerzia generata dalla lunghezza del link due.

Infine in *Tabella 2.1* sono anche riportati il modulo dell'offset che il giunto in esame ha rispetto al successivo, naturalmente la base e l'ultimo giunto sono esenti da questa considerazione. Si nota subito come la quasi totalità dei link presenta un offset fra il giunto precedente e quello successivo, l'unico che non lo presenta è il giunto maggiore, ovvero il secondo link che presenta due giunti rivolti con la stessa orientazione al fine di ottimizzarne l'inerzia riducendolo a compiere una rotazione lungo un solo asse e quindi un solo tipo sollecitazione.

Tutti i link che sono stati descritti precedentemente rappresentano la struttura portante del manipolatore, ciò che fisicamente mantiene tutto il peso. Essa rappresenta lo scheletro del manipolatore ed è l'elemento che mantiene e assicura la rigidità all'intero manipolatore articolato, provvede a svolgere funzione di telaio per tutti gli altri organi ed elementi che inevitabilmente sono montati su di esso. Per quanto riguarda l'aspetto estetico sullo scheletro e sui vari organi montati su di esso è montato un rivestimento che contribuisce alla protezione degli organi e inoltre conferisce l'aspetto esteriore del robot.

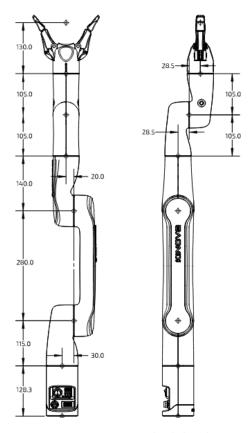


Figura 2.2: Dimensioni nominali del robot in millimetri

Componente	Taglia attuatore	Lung. [mm]	Offset fra giunti [mm]
Base	Media potenza	128,3	
Primo link	Grande potenza	115,0	30
Secondo link	Media potenza	280,0	0
Terzo link	Piccola potenza	114,0	20
Quarto link	Piccola potenza	105,0	28,5
Quinto link	Piccola potenza	105,0	28,5
Pinza	Attuatore pinza	113,0	

Tabella 2.1: Tabella riassuntiva delle lunghezze dei vari link del manipolatore articolato

2.1.2 Sensoristica

La sensoristica rappresenta l'oggetto fisico grazie al quale si ha una retroazione e una misura dello stato del manipolatore. Essa è montata dentro la struttura portante, la quale all'interno è cava e permette l'alloggiamento dei cavi utili per alimentare

e ricevere segnali di output dai vari motori e sensori montati sul manipolatore articolato.

I principali sensori presenti nel manipolatore articolato servono a leggere i dati di feedback utili per effettuare controlli di sicurezza e per retro-azionare efficacemente ed in tempo reale il movimento del manipolatore articolato. Di seguito vengono raccolti alcuni tipi di sensori presenti sul robot con le loro principali applicazioni:

- Sensori di temperatura: non sono molto importanti dal punto di vista delle applicazioni del manipolatore articolato, tuttavia sono fondamentali da un punto di vista della sicurezza del manipolatore articolato.
- Sensori di corrente: sono sensori importanti sia dal punto di vista dell'utilizzo del braccio nelle sue applicazioni, sia dal punto di vista della sicurezza. Nell'ambito del controllo il sensore di corrente è utilizzato in luogo del sensore di coppia, per comprendere che forza sta esercitando il braccio. Nell'ambito della sicurezza conoscere la corrente in transito nel circuito robotico è importante per evitare surriscaldamenti del braccio.
- Sensori di posizione assoluti e relativi: sono sensori utilizzati per ottenere informazioni utili al processo di retroazione del robot (posizione assoluta e velocità). In aggiunta svolge funzioni inerenti alla sicurezza se vengono delimitate delle aree nello spazio di lavoro del robot nel quale non si può accedere o si può accedere a velocità ridotta.
- Accelerometro o giroscopio: necessario per capire l'orientamento del robot infatti sono presenti solo nella base e nel gripper, per conoscerne il reciproco orientamento.
- Voltmetro: per assicurare un corretto funzionamento al sistema robotico, alla corretta tensione.

Di seguito sono riassunti tutti i sensori presenti nel manipolatore articolato:

Sensori della base

• Tensione;

- Corrente;
- Temperatura;
- Sensore di temperatura della CPU;
- Accelerometro o Giroscopio a sei assi.

Sensori dell'attuatore Sensori del gripper

- Sensore di corrente del motore;
- Sensore di temperatura del motore;
- Sensore angolare ad effetto Hall;
- Sensore angolare magnetico;
- Sensore di controllo per la tensione a 24V;

- Due sensori capacitivi;
- Micro-controllore di temperatura;
- Micro-controllore della corrente di bordo;
- Giroscopio a sei assi o accelerometro;
- Sensore di corrente del motore del gripper;
- Sensore di temperatura del motore del gripper;
- Sensore di tensione del motore;
- Sensore di posizione e velocità del motore che attua il gripper.

Gli attuatori dei bracci robotici presentano degli ingranaggi e dei riduttori per collegare i motori elettrici degli attuatori con gli organi dei bracci robotici. Tutti gli attuatori meno che quello grande presentano un riduttore epicicloidale ad alta efficienza, mentre l'attuatore grande ha un ingranaggio armonico al fine di ridurre i giochi ed i contraccolpi. L'insieme di questi attuatori sono tutti collegati tra di loro tramite cavi che portano l'alimentazione e la connessione ethernet, collegamenti che quindi garantiscono il movimento al robot.

Il gripper presenta un attuatore lineare che si occupa dell'apertura e della chiusura della pinza dell'organo terminale. Il gripper è composto da: una plastica flessibile ma robusta che garantisce integrità strutturale e resistenza e una gomma plastica per avere un miglior grip sugli oggetti da manipolare. In Figura 2.3 sono mostrate diverse configurazioni che possono essere assunte dalla pinza del manipolatore articolato per assicurare la presa dell'oggetto. Grazie alle due falangi

in cui è diviso il gripper della pinza esso può articolarsi in movimenti e prese più complesse. L'attuatore e le falangi permettono al gripper di sotto-auttare i movimenti che le pinze terminali possono fare, dunque è grazie alla sotto-attuazione è possibile avere diverse configurazioni delle pinze dell'organo terminale.

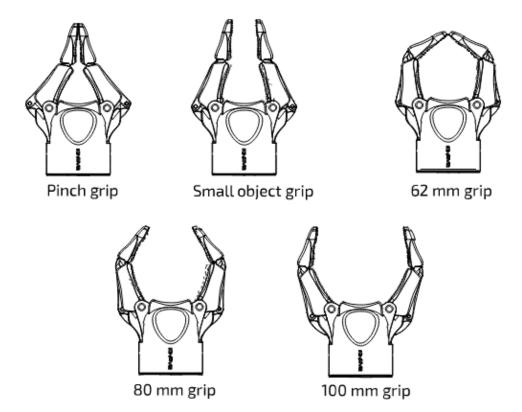


Figura 2.3: Esempi di diverse posizioni assumibili dal manipolatore per afferrare oggetti differenti

2.1.3 Spazio di lavoro

Lo spazio di lavoro è lo spazio tridimensionale nel quale il robot è libero di muoversi e posizionare il centro del suo organo terminale. É influenzato principalmente da fattori come:

- lunghezza dei link;
- numero dei link;
- il raggio d'azione dei giunti;
- la forma dei link.

In base al tipo di spazio di lavoro a cui si è interessanti esistono due diversi tipi di definizioni, una più stringente ed una più generale o inclusiva:

- Spazio di lavoro nominale o raggiungibile: è l'insieme di tutte le posizioni nello spazio tridimensionale che sono raggiungibili dall'organo terminale attraverso una opportuna combinazione di posizione ed orientazione dell'organo terminale. In Figura2.4 è mostrato una sezione dello spazio di lavoro nominale del manipolatore articolato. Si nota che lo spazio di lavoro nominale corrisponde approssimativamente ad una sfera di raggio 760 mm centrata sul giunto di spalla del robot.
- Spazio di lavoro destro: è il sottoinsieme dello spazio di lavoro nominale nel quale l'organo terminale ha ancora la piena libertà di muoversi in tutti e sei i gradi di libertà.

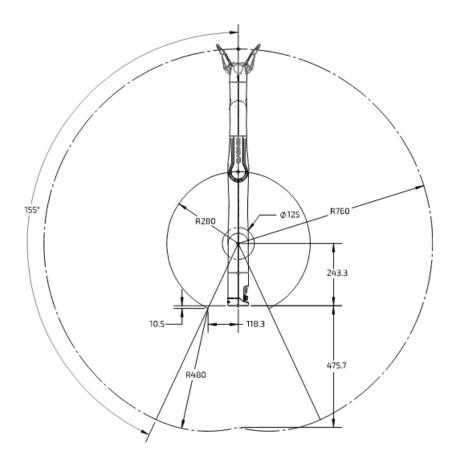


Figura 2.4: Rappresentazione di una sezione bidimensionale dello spazio di lavoro nominale in millimetri

2.2 Cinematica del manipolatore articolato

In questa sezione si tratta della cinematica del robot, viene spiegato nel dettaglio il legame fra le varie posizioni dei giunti del manipolatore articolato e la posizione assunta dall'organo terminale rispetto al sistema di riferimento cartesiano ortogonale avente origine nella base del robot.

In dettaglio verranno illustrate la cinematica diretta sia di posa che di velocità e la cinematica inversa di posa. La cinematica diretta di posa serve a ricavare la posizione dell'organo terminale note le posizioni delle coordinate di giunto, analogo discorso per la cinematica diretta di velocità che serve a ricavare la velocità dell'organo terminale note le velocità dei giunti. La cinematica inversa di posa serve a calcolare le posizioni dei giunti nota la posizione dell'organo terminale rispetto alla terna cartesiana solidale con la base. Il calcolo della cinematica inversa è molto utilizzato quando si vuole pianificare un moto passante per dei punti di cui sono note le coordinate cartesiane rispetto alla terna cartesiana solidale con la base robotica.

2.2.1 Cinematica diretta di posa

I bracci robotici sono composti da un insieme di corpi rigidi, i link, connessi in cascata tramite coppie cinematiche o giunti, i quali nel caso del manipolatore articolato sono tutti rotoidali. Essendo tutti i link collegati da cerniere ogni link avrà un grado di libertà che viene tipicamente associato all'articolazione di giunto e costituirà una variabile di giunto atta a descrivere la postura del manipolatore articolato.

Risolvere la cinematica diretta di posa del robot significa fare in modo che le informazioni locali, relative alle variabili di giunto della singola articolazione, si trasformino efficacemente in una informazione globale: la posizione dell'organo terminale rispetto al sistema di rifermento cartesiano fisso e solidale con la base del robot.

Viene scelto di utilizzare la notazione di Denavit-Hartemberg modificata [19] mostrata in Figura~2.5 con i vari sistemi di riferimento coerenti utili per passare dal sistema di riferimento di un giunto al successivo. La scelta di opportuni sistemi di riferimento è utilizzata per descrivere il sistema di riferimento i+1-esimo rispetto al sistema di riferimento precedente i-esimo, un esempio schematico di questo passaggio viene mostrato in Figura~2.6. Il passaggio da un sistema di riferimento al sistema successivo si ha tramite due traslazioni e due rotazioni. La logica[20] che guida il passaggio dal sistema di riferimento in un giunto al successivo è la seguente:

• si sceglie l'asse z_i giacente lungo l'asse del giunto i+1;

- si individua O_i all'intersezione dell'asse z_i con la normale comune agli assi z_{i-1} e z_i e con $O_{i'}$ si indica l'intersezione della normale con z_{i-1} ;
- si sceglie l'asse x_i diretto lungo la normale comune agli assi z_{i-1} e z_i con verso positivo dal giunto i al giunto i+1;
- si sceglie l'asse y_i in modo da completare la terna.

Dunque le trasformazioni geometriche utilizzate per realizzare il passaggio fra un sistema di riferimento ed il successivo, in accordo con la convenzione di Denavit-Hartemberg modificata sono riassunte di seguito:

- una rotazione attorno all'asse x_{i-1} , ovvero l'asse del sistema di riferimento precedente, di α_i la quale rotazione permette di allineare la z_{i-1} con la z_i .
- una traslazione di a_i , ovvero la distanza fra gli assi dei due giunti, il giunto i-1 ed il giunto i.
- una traslazione di d_i lungo l'asse z_i che porta la terna del sistema di riferimento laddove passa la normale ai due assi, z_i e z_{i+1} .
- una rotazione di θ_i lungo z_i al fine di allineare l'asse x_i con il link i-esimo.

Tramite le equazioni equivalenti (2.1) e (2.1) si descrivono in termini matematici le rotazioni e le traslazioni elencate sopra. In particolare la (2.1) rappresenta in maniera compatta le istruzioni che vengono eseguite per passare da un giunto al successivo, mentre la (2.2) rappresenta in forma matriciale le medesime informazioni.

$$^{\mathbf{i-1}}\mathbf{T_{i}} = Rot\left(x_{i-1}, \alpha_{i}\right) Trans\left(x_{i-1}, a_{i}\right) Trans\left(z_{i}, d_{i}\right) Rot\left(z_{i}, \theta_{i}\right)$$
(2.1)

$$\mathbf{i}^{-1}\mathbf{T}_{\mathbf{i}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_{i} & -\sin\alpha_{i} & 0 \\ 0 & \sin\alpha_{i} & \cos\alpha_{i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_{i} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_{i} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_{i} & -\sin\theta_{i} & 0 & 0 \\ \sin\theta_{i} & \cos\theta_{i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(2.2)

Visto che la (2.1) e la (2.2) descrivono il passaggio da un link al successivo si deduce l'equazione generale che permette di scrivere la posizione di un generico giunto rispetto al sistema di riferimento fisso, solidale con il robot. Infatti dato un sistema di riferimento fisso, detto $\{0\}$, solidale al giunto di base del robot si può scrivere attraverso la (2.3) la posizione di un generico giunto rispetto al sistema di riferimento fisso, ovvero solidale con il link di base del robot. É facile capire come la moltiplicazione matriciale fra tutte le matrici dei sei giunti possa portare ad ottenere la cinematica diretta del robot.

$${}^{0}\mathbf{T_{i}} = {}^{0}\mathbf{T_{1}}...{}^{i-2}\mathbf{T_{i-1}}{}^{i-1}\mathbf{T_{i}}$$
 (2.3)

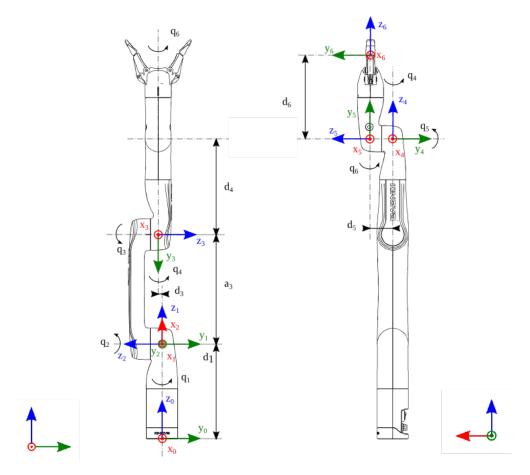


Figura 2.5: Sistema di riferimento e nomenclatura per la risoluzione della cinematica diretta secondo la convenzione di Denavit-Hartemberg

Infine i quattro parametri presenti nelle equazioni (2.2) e (2.1) sono necessari in ogni trasformazione per passare dal giunto precedente al successivo. Vengono raccolti per tutti i giunti e costituiscono i parametri caratteristici di Denavit-Hartemberg per ogni trasformazione: due rotazioni α e θ e due traslazioni a e d. In Tabella 2.2 sono riportati tutti i parametri di Denavit-Hartemberg del robot Kinova Gen3 Lite attraverso i quali si possono ricavare tutte le matrici di trasformazione del robot fra i vari giunti.

2.2.2 Cinematica inversa di posa

La cinematica inversa ha il compito di calcolare il set di angoli di giunto \mathbf{q} tali che si possa raggiungere la posa desiderata dell'end effector, ovvero posizione ed

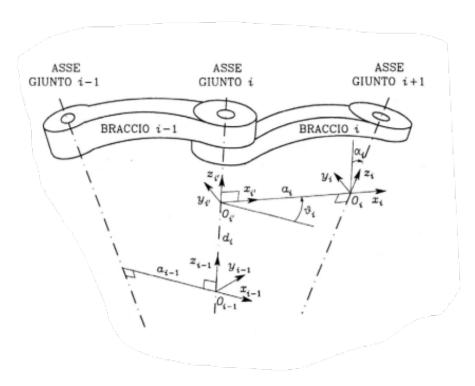


Figura 2.6: Rappresentazione del funzionamento della convenzione di Denavit-Hartemberg che permette di descrivere un sistema di rifermento i rispetti al sistema di riferimento i-1 [20]

Link i	$\alpha_{\mathbf{i}} [rad]$	$\mathbf{a_i} \ [m]$	$\theta_{\mathbf{i}} [m]$	$\mathbf{d_i} [rad]$
1	0	0	0.243	q_{i}
2	$\pi/2$	0	0	$\pi/2 + q_2$
3	π	0.280	-0.010	$\pi/2 + q_3$
4	$\pi/2$	0	0.245	$\pi/2 + q_4$
5	$\pi/2$	0	0.057	q_5
6	$-\pi/2$	0	0.235	q_6

Tabella 2.2: Parametri di Denavit-Hartemberg con cui si caratterizza il passaggio fra i giunti del robot Kinova Gen3 Lite

orientamento scelti dall'utente (ad esempio in termini di x, y, z, θ_x , θ_y e θ_z rispetto al sistema di riferimento di base). Di solito il problema non è univoco, ma si ottengono più set di giunti che possono essere soluzione del problema di cinematica inversa dunque la scelta di un set di giunti piuttosto che di un'altra va fatta in base ad ulteriori considerazioni sulle performance del robot, sullo spazio da percorrere

per raggiungere tale configurazione e su eventuali ostacoli contro cui il robot può urtare. Scelto il set di angoli di giunto \mathbf{q} si utilizza la cinematica diretta per portare il gripper nella posa desiderata.

In Figura 2.7 è mostrata la nomenclatura ed i sistemi di riferimento usati per risolvere la cinematica inversa del robot. É mostrata la posa finale dell'end effector con rispettivo sistema di riferimento: $\{EE\}$. Sono mostrati anche dei sistemi di riferimento intermedi che sono utili per la scrittura delle equazioni risolutive della cinematica inversa, questi sistemi di riferimento sono: $\{s\}$, $\{1\}$ e $\{2\}$. Le matrici di trasformazione che collegano i sistemi di riferimento $\{s\}$ con $\{1\}$ e i sistemi di riferimento $\{1\}$ con $\{2\}$ sono le $\{2.4\}$ e la $\{2.5\}$

$${}^{\mathbf{s}}\mathbf{T}_{1} = \begin{bmatrix} Rot(x, \Phi_{1}) & [0\ 0 - d_{3}]^{T} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Rot(z, \Phi_{2}) & [x\ y\ z]^{T} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(2.4)

$${}^{\mathbf{1}}\mathbf{T_{2}} = \begin{bmatrix} Rot (y, \Phi_{3}) & [0 - d_{4} \ 0]^{T} \\ 0 & 0 & 1 \end{bmatrix}$$
 (2.5)

Al fine di risolvere la cinematica inversa si scrivono le equazioni dei punti A, B, C, D ed E. La posizione del punto A rispetto al sistema di riferimenti fisso è nota, coincide infatti con la posizione dell'end effector ${}^{s}\mathbf{T}_{\mathbf{EE}}$. Il punto B è scritto (2.6) come una traslazione lungo $\hat{\mathbf{k}}_{\mathbf{EE}}$ di una quantità pari a $-d_6$ e come traslazione di d_5 lungo $\hat{\mathbf{k}}_2$. La posizione del punto C (2.7) equivale alla posizione del punto 2, quindi viene scritta come moltiplicazione fra le due matrici di trasformazione: ${}^{s}\mathbf{T}_1$ e ${}^{1}\mathbf{T}_2$. La posizione del punto D è scritta in riferimento alle coordinate cartesiane che occupa rispetto ad s (2.8). Infine la posizione del punto E viene descritta tramite i movimenti della spalla del giunto, secondo la (2.9).

$$\mathbf{p_A} - d_6 \hat{\mathbf{k}}_{\mathbf{EE}} = \begin{pmatrix} \mathbf{s} \mathbf{T_1^1} \mathbf{T_2} \begin{bmatrix} 0 \\ 0 \\ d_5 \\ 1 \end{bmatrix} \end{pmatrix}_{(3,:)}$$
 (2.6)

$$\mathbf{p_C} = {}^{\mathbf{s}}\mathbf{T_1} \, {}^{\mathbf{1}}\mathbf{T_2} \tag{2.7}$$

$$\mathbf{p_D} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.8}$$

$$\mathbf{p_E} = \begin{bmatrix} Rot(x, \Phi_1) & \begin{bmatrix} 0 & 0 & -d_3 \end{bmatrix}^T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
 (2.9)

La soluzione del problema di cinematica inversa si ottiene mettendo a sistema (2.10) le equazioni finora ottenute o risultati che derivano da esse. Le prime tre

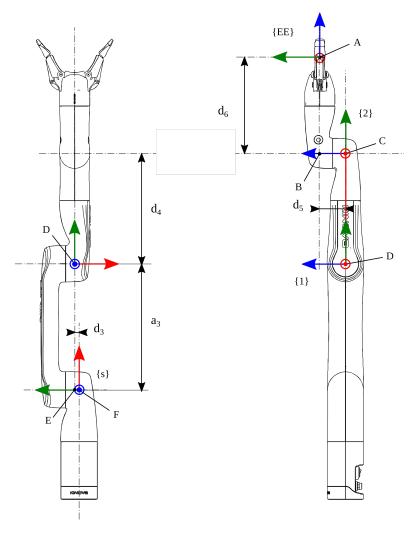


Figura 2.7: Sistema di riferimento e nomenclatura per la risoluzione della cinematica inversa secondo la convenzione di Denavit-Hartemberg

equazioni del sistema $(eq_{1,2,3})$ derivano dalla scrittura del punto B secondo le due configurazioni: partendo dal punto A o partendo dal punto C. L'equazione (eq_4) impone la perpendicolarità tra il versore $\hat{\mathbf{k}_2}$ ed il versore $\hat{\mathbf{k}_{EE}}$. L'equazione (eq_5) impone al punto D l'appartenenza ad una sfera di raggio $\sqrt{d_3^2 + a_3^2}$ centrata in F. L'equazione (eq_6) impone che il punto D appartenga ad un sottoinsieme di tale sfera in modo tale che la proiezione di $\mathbf{p_D}$ lungo la z del riferimento $\{1\}$ sia in modulo uguale alla lunghezza d_3 .

Dalla risoluzione del sistema di equazioni (2.10) si ottengono le 16 configurazioni nelle quali si possono disporre i punti di interesse del robot rispettando i vincoli imposti. Al fine di ottenere tali configurazioni in funzione degli angoli di spazio

giunti bisogna scrivere le relazioni che legano gli angoli di giunto alla posizione relativa dei punti di interesse, in modo da ottenere gli angoli di giunto delle 16 configurazioni possibili.

$$\begin{cases}
\mathbf{p_{A}} - d_{6} \hat{\mathbf{k}}_{\mathbf{EE}} = \begin{pmatrix} \mathbf{s} \mathbf{T_{1}^{1}} \mathbf{T_{2}} \begin{bmatrix} 0 \\ 0 \\ d_{5} \\ 1 \end{bmatrix} \end{pmatrix}_{(3,:)} & (eq_{1,2,3}) \\
[Rot(x, \Phi_{1}) Rot(z, \Phi_{2}) Rot(y, \Phi_{3})]_{(1:3,3)}^{T} \mathbf{s} \mathbf{T_{EE}} (1:3,3) = 0 & (eq_{4}) \\
\left\| \begin{bmatrix} x \ y \ z \end{bmatrix}^{T} \right\|^{2} = (\mathbf{p_{D}} - \mathbf{p_{F}})^{2} & (eq_{5}) \\
\left\| \begin{bmatrix} x \ y \ z \end{bmatrix}^{T} \hat{\mathbf{k}_{1}} \right\|^{2} = (\mathbf{p_{E}} - \mathbf{p_{F}})^{2} & (eq_{6})
\end{cases}$$

2.2.3 Cinematica diretta di velocità

Lo scopo di questa sottosezione è quello di scrivere il twist di velocità associato all'end-effector in funzioni delle velocità di giunto. L'equazione (2.11) rappresenta il legame tra le velocità di giunto e la velocità dell'end-effector sia come velocità di traslazione che di rotazione.

$$\mathbf{V_{EE}} = \begin{bmatrix} \mathbf{v_n} \\ \omega_n \end{bmatrix} = \mathbf{J_{EE}}(\mathbf{q}) \, \dot{\mathbf{q}}$$
 (2.11)

Dove $\mathbf{J_{EE}}$, avendo usato la notazione di Denavit-Hartemberg Modificata, è una matrice $6 \times n$ composta da due sottomatrici $3 \times n$ che descrivono il cambiamento della velocità $\mathbf{J_{P}}$ e dell'orientamento $\mathbf{J_{O}}$ (2.12). Mentre $\dot{\mathbf{q}}$ è composto dal vettore delle velocità di giunto $\left[\dot{q}_{1}\ \dot{q}_{2}\ \dot{q}_{3}\ \dot{q}_{4}\ \dot{q}_{5}\ \dot{q}_{6}\right]^{T}$. Un'altra rappresentazione della matrice dello Jacobiano $\mathbf{J_{EE}}$ è mostrata tramite l'equazione (2.13) dove viene esplicitato il contributo colonna per colonna della matrice del Jacobiano, in particolare viene evidenziato il contributo i-esimo della colonna dello Jacobiano. Il valore di δ_{i} varia tra 0 ed 1 a seconda se il giunto i-esimo è prismatico o è rotoidale. Infine è importante notare che il vettore $\mathbf{p_{i}}$ è l'origine del sistema di riferimento i-esimo e $\hat{\mathbf{k_{i}}}$ è il versore dell'asse z del sistema di riferimento, che per via della notazione scelta coincide in direzione e verso con l'asse giunto.

$$\mathbf{J}_{\mathbf{E}\mathbf{E}}(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_{\mathbf{P}} \\ \mathbf{J}_{\mathbf{O}} \end{bmatrix} \tag{2.12}$$

$$J_{i} = \begin{bmatrix} \delta_{i} \hat{\mathbf{k}}_{i} + (1 - \delta_{i}) \hat{\mathbf{k}}_{i} \times (\mathbf{p}_{i} - \mathbf{p}_{i-1}) \\ (1 - \delta_{i}) \hat{\mathbf{k}}_{i} \end{bmatrix}$$
(2.13)

2.2.4 Dualità cineto-statica

La matrice dello Jacobiano precedentemente presentata e ricavata si presta anche ad alcune considerazioni statiche rispetto alla relazione che lega le forze esplicitate dall'organo terminale e le coppie applicate ai giunti. Viene usato il principio dei lavori virtuali per ricavare la sopracitata relazione, in particolare essendo i manipolatori dotati di vincoli olonomi 2 indipendenti dal tempo le cui configurazioni assunte dipendono solo dalle variabili di giunto \mathbf{q} . Da questa considerazione è possibile affermare che gli spostamenti virtuali coincidono con gli spostamenti elementari.

Di seguito viene esplicitata, nell'equazione (2.14), la relazione che esprime il lavoro dei giunti rispetto alle coppie applicate e agli spostamenti compiuti.

$$dW_{\tau} = \boldsymbol{\tau}^T d\mathbf{q} \tag{2.14}$$

Dove τ rappresenta il vettore (6 × 1) delle coppie applicate ai giunti. Per quanto riguarda le forze esplicate dall'organo terminale in prima battuta esse possono essere espresse secondo la (2.15) dove viene separato rispetto all'orgine della terna utensile il contributo di forza, \mathbf{f}_e , da quello di momento $\boldsymbol{\mu}_e$.

$$dW_{\gamma} = \mathbf{f}_{e}^{T} d\mathbf{p}_{e} + \boldsymbol{\mu}_{e}^{T} \boldsymbol{\omega}_{e} dt \tag{2.15}$$

Dove $d\mathbf{p}_e$ rappresenta uno spostamento lineare e $\boldsymbol{\omega}_e dt$ rappresenta uno spostamento angolare. Al fine di esprimere anche la (2.15) in funzione di $d\mathbf{q}$ vengono sfruttate le proprietà viste nelle matrici(2.12) e (2.11) per il calcolo della velocità anche per implicazioni riguardanti la posizione ottenendo la (2.16) che esprime in modo più compatto il lavoro nell'organo terminale del manipolatore.

$$dW_{\gamma} = \mathbf{f}_{e}^{T} \mathbf{J}_{P}(\mathbf{q}) d\mathbf{q} + \boldsymbol{\mu}_{e}^{T} \mathbf{J}_{O}(\mathbf{q}) d\mathbf{q} = \boldsymbol{\gamma}_{e}^{T} \mathbf{J}_{EE}(\mathbf{q}) d\mathbf{q}$$
(2.16)

Dove si ha che $\gamma_e = \left[\mathbf{f}_e^T \boldsymbol{\mu}_e^T\right]^T$. Seguendo la teoria dei lavori virtuali si impone l'eguaglianza dei lavori virtuali per ogni $\delta \mathbf{q}$ fra giunti e organo terminale al fine di garantire l'equilibrio statico. L'equazione che si ottiene è la (2.17). Questa equazione appena ricavata mette in relazione, tramite la trasposta dello Jacobiano, le forze esplicitate dall'organo terminale con le coppie ai giunti.

$$\tau = \mathbf{J}^{\mathbf{T}}(\mathbf{q})\gamma_e \tag{2.17}$$

 $^{^2}$ In meccanica è un vincolo di posizione, cioè quello caratterizzato analiticamente dall'annullarsi di una funzione delle sole coordinate generalizzate. In particolare, per un punto di coordinate x, y, z, in funzione del tempo t risulta: f(x, y, z, t)=0. In particolare il vincolo non dipende dalla velocità

L'analogia cineto-statica si basa sulla somiglianza fra la (2.11) e la (2.17). Infatti nella prima si esplica, tramite lo Jacobiano, la relazione fra velocità ai giunti e velocità espressa dall'organo terminale. Nella seconda viene espressa la relazione fra forze espresse dall'organo terminale e coppie ai giunti, con la trasposta dello Jacobiano a collegare i due termini. Da ciò si deduce il motivo per cui sia lo Jacobiano che la sua trasposta sono d'interesse per il calcolo di grandezze fisiche legate al manipolatore articolato. Inoltre diverse grandezze, come le velocità angolari dei giunti e le forze/coppie esplicate dall'organo terminale, si possono calcolare grazie all'inversa dello Jacobiano quindi esso rappresenta un tema molto sentito in letteratura scientifica.

2.3 Interfaccia software con il manipolatore articolato

Esistono diverse tipologie di interfacce informatiche che permettono di comunicare con i driver del manipolatore e controllarne i movimenti. Di seguito sono elencate quelle che sono state valutate in questo lavoro di tesi:

- Utilizzando le API³;
- Utilizzando le API del robot tramite ROS[21];
- Utilizzando le API del robot tramite il pacchetto MoveIt[22] di ROS.

Utilizzare le sole API è il metodo più facile e veloce per interfacciarsi col robot, tuttavia è anche il metodo meno consigliato poiché i comandi vengono impartiti con una logica a basso livello. Inoltre sono presenti due tipi di logiche: una più a basso livello e una più ad altro livello che integra una libreria di sicurezza che evita al robot movimenti che possono provocare collisioni contro se stesso. Lavorare con le sole API presenta problematicità quando si lavora con input forniti in spazio operativo poiché la loro capacità di risolvere la cinematica in prossimità delle singolarità e la loro discretizzazione della traiettoria da compiere non è eccellente.

Utilizzare le API tramite ROS presenta le stesse criticità di usare le API da sole. Il vantaggio fornito da ROS è la facilità di interagire con altri sistemi, ad esempio in questa applicazione la camera, e riuscire ad unire i due in un unico applicativo più agevolmente.

³Con Application Programming Interface (API) si indica un insieme di procedure, in genere raggruppate per strumenti specifici, atte all'espletamento di un dato compito. Ad esempio tale termine designa le librerie software di un linguaggio di programmazione. Nel caso del robot le API rappresentano le librerie software necessarie ad interfacciare il robot con un elaboratore, come il pc, da cui riceve gli input e a cui consegna i feedback.

Il pacchetto di MoveIt su ROS permette una programmazione del moto ad alto livello. É in grado di suddividere la traiettoria da compiere in vari punti di passaggio (waypoints). La scomposizione oculata della traiettoria in punti, evitando le singolarità, ed il calcolo delle coordinate e delle velocità di giunto che si devono manifestare in ogni punto in cui è suddiviso la traiettoria permettono di sfruttare al meglio la risoluzione degli algoritmi di cinematica inversa. Tramite l'utilizzo di MoveIt si ha la certezza di far compiere un movimento fluido al robot che rispetti anche i vincoli cinematici ed i vincoli spaziali imposti rispettivamente dal robot e dall'ambiente.

Fra le tre alternative presentate in questo lavoro di tesi è stato scelto di movimentare il manipolatore articolato usando MoveIt poiché offre un migliore controllo sulla traiettoria la quale viene pianificata secondo leggi del moto note di volta in volta. Inoltre permette anche pratiche più avanzate ed efficienti di collision avoidance utili sia per la salvaguardia del robot stesso, ma soprattutto per fare operare il robot in contesti di assistenza fra le persone.

2.3.1 Logica dell'interfaccia con il manipolatore articolato

A prescindere che si programmi il moto del manipolatore articolato in C++ o in Python o che si usino le sole API del robot o che si impartiscano comandi attraverso ROS la struttura dei codici con cui si impartiscono le istruzioni presenta una logica sempre simile. Nella prima parte del codice si trova la definizione delle funzioni atte a interfacciarsi con il robot e a fargli eseguire i vari comandi. Tali dichiarazioni di funzioni rappresentano la struttura fissa del codice poiché descrivono le istruzioni atte a compiere i vari movimenti del manipolatore articolato. Un elenco delle definizioni che si possono trovare sono:

- muovere il robot con un input in spazio operativo (usando le coordinate cartesiane e l'orientamento dell'organo terminale);
- movimento del robot usando le coordinate di giunto;
- muovere il gripper;
- muovere il robot in posizioni prestabilite (esempio: posizione home, posizione retract, altre posizioni costruite dall'utente);
- ottenere il feedback della posizione dei giunti;
- ottenere il feedback della posizione cartesiana e dell'orientamento rispetto il sistema di riferimento fisso.

Queste funzioni sopra elencate sono alla base del movimento del robot, infatti possono essere visti come i tasselli grazie ai quali si costruisce un movimento completo ed articolato del manipolatore articolato. Il vantaggio di averle come funzioni è la versatilità: così possono essere richiamate nella funzione principale in modo da comporre un movimento più complesso composto da più movimenti in successione.

Algoritmo per il movimento del robot usando le coordinate di giunto

L'Algoritmo 1 si occupa del movimento tramite input in coordinate giunto, una per una ogni coordinata di giunto che il robot ha a disposizione. Nel caso più complesso, ovvero il caso di MoveIt, le librerie richiamate svolgono i controlli affinché non ci siano collisioni con la struttura stessa del robot e con gli oggetti che vengono specificati informaticamente nello spazio di lavoro e rispettare i vincoli cinematici imposti ai giunti. Se tutti i precedenti criteri vengono rispettati i giunti vengono mandati alla posizione desiderata.

Algorithm 1 Algoritmo di una funzione per la movimentazione con input in coordinate giunto

```
Require: robot\_interface, joints vector and tollerance (eventualy)

Ensure: movement of robot at joint coordinates

joint\_goal[0:5] \leftarrow joints[0:5] \triangleright \text{Fill the joint goal with the desired joint set}

robot.set\_tolerance(tollerance) \triangleright \text{If not passed or not as input tollerance is a}

fix number in the function

done \leftarrow robot\_interface.move(joint\_goal)

if done == 1 then

control of movement success & return 1

else

if done == 0 then

control of movement failure & return 0

end if

end if
```

Algoritmo per il movimento in spazio operativo

L'algoritmo 2 si occupa del movimento del robot in spazio operativo, è strutturato in modo tale che prenda in input la posizione e l'orientamento della posa da raggiungere e svolga le istruzioni necessarie a compiere il movimento. Nel caso più complesso, ovvero il caso di MoveIt, vengono richiamate le librerie che si occupano di: pianificare la traiettoria, evitare di far collidere il robot con se stesso e con gli oggetti specificati informaticamente nello spazio di lavoro e rispettare i vincoli cinematici imposti ai giunti. Il movimento risultante se rispetta tutti i vari criteri

imposti in precedenza viene attuato per raggiungere la posizione imposta. Inoltre l'algoritmo è in grado di calcolare gli spostamenti mantenendo un orientamento del polso fisso in un modo, se si specifica un determinato orientamento come vincolo in input.

Algorithm 2 Algoritmo di una funzione per la movimentazione con input in spazio operativo

```
Require: robot\_interface, x, y, z, \theta_x, \theta_y, \theta_z and tollerance (eventualy),
  constraint (eventually)
Ensure: movement of robot at specified pose
  catesian\_pose \leftarrow robot\_interface.feedback
                                                          \triangleright Set cartesian\_pose as I/O
  struct
  cartesian\_pose.x \leftarrow x
  cartesian\_pose.y \leftarrow y
  cartesian pose.z \leftarrow z
  cartesian\_pose.theta\_x \leftarrow \theta_x
  cartesian\_pose.theta\_y \leftarrow \theta_y
  cartesian\_pose.theta\_z \leftarrow \theta_z
                                       ▶ Replace the feedback with new position and
  orientation
  robot.set \ tolerance(tollerance) > If not passed or not as input tollerance is a
  fix number in the function
  if constraints is present then
      robot.set\_set_path_constraints(constraints)
  end if
  done \leftarrow robot\_interface.move(cartesian\_pose)
  if done == 1 then
      control of movement success & return 1
  else
      if done == 0 then
          control of movement failure & return 0
      end if
  end if
```

Algoritmo per muovere il robot in posizioni prestabilite

Questi tipi di algoritmi rappresentano dei casi particolari dell'*Algoritmo 1 e 2* poiché si compongono della stessa struttura e sequenza di informazioni ma senza richiedere in input una posizione da raggiungere. Infatti dentro l'algoritmo è già stabilita la posizione che deve essere raggiunta, tipicamente come posizione giunti in quanto rappresenta una posa univoca del robot. Inoltre ci sono due posizioni

prestabilite dal costruttore del robot che possono essere raggiunte chiedendo al robot di raggiungere la posa chiamata HOME o RETRACT, al posto di passargli una posa obiettivo qualsiasi. Tuttavia queste due pose prestabilite in produzione del braccio, per quanto efficaci nono sono state utilizzate in questo elaborato.

Algoritmo per ottenere feedback dei giunti o della posizione e della posa

Sia l'algoritmo per ottenere il feedback rispetto alle posizioni dei giunti sia l'algoritmo per ottenere il feedback rispetto alla posa ed all'orientamento dell'organo terminale sono molto semplici. Spesso si tratta solo di una istruzione dell'interfaccia del robot che va a leggere lo stato dei giunti o della posa del robot e popola un opportuno vettore o una opportuna struttura di dati.

Algoritmo per il movimento del gripper

L'Algoritmo 3 per movimentare il gripper risulta più snello e semplice rispetto a quelli per movimentare il manipolatore articolato. Il gripper viene comandato tramite la percentuale di apertura e chiusura che si vuole ottenere, usando i dati di massima e minima apertura possibile.

Algorithm 3 Algoritmo di una funzione per la movimentazione del gripper

```
Require: robot interface, relative position
Ensure: movement of robot gripper
  gripper\_joint \leftarrow robot\_interface.get\_joint\_name(gripper\_name)
  Max \ bound \leftarrow gripper \ joint.max \ bound
  Min\_bound \leftarrow gripper\_joint.min\_bound
                                                   ⊳ gripper_joint contain also the
  information about the movement of the robot
  move \ value \leftarrow relative \ position \cdot (Max \ bound - Min \ bound) + Min \ bound
  done \leftarrow robot \ interface.move(move \ value)
  if done == 1 then
     control of movement success & return 1
  else
     if done == 0 then
         control of movement failure & return 0
     end if
  end if
```

2.3.2 Strumento ROS per la movimentazione: MoveIt

Si discute in dettaglio il funzionamento del pacchetto MoveIt di ROS. Data in input una posa, che è composta da una posizione dell'end-effector p_{EE} e da un orientamento Φ_{EE} , permette all'organo terminale di raggiungere tale posizione. Come mostrato in Figura 2.8 l'input in spazio operativo viene dato in pasto all'algoritmo di pianificazione del moto, da questo si ricava un set di posizioni di giunto e di velocità di giunto. Il feedback è un'importante ingresso nell'algoritmo di pianificazione, poiché in base alla posizione attuale del robot cambia conseguentemente la pianificazione del moto che bisogna compiere per portare l'organo terminale nella posizione specificata dall'input. La sequenza di coordinate e velocità di giunto rappresenta la legge del moto che ogni giunto deve rispettare affinché avvenga il movimento dalla posizione attuale a quella specificata in input, ad esempio il giunto uno all'istante N ha pianificata una velocità \dot{q}_1 e una posizione angolare q_1 . I driver del robot rappresentano l'interfaccia fra il l'insieme di comandi impartiti e il movimento dei giunti, sono dunque le API del robot che permettono di fare dialogare gli attuatori dei giunti con l'informatica più di alto livello.



Figura 2.8: Schematizzazione del processo di pianificazione del moto

In Figura 2.9 è mostrata la logica che governa l'algoritmo di pianificazione del moto. Esso prende in input il set cioè la posa che si vuole raggiungere in termini di posizione ed orientamento, ed il feedback cioè la posa in cui attualmente si trova il robot. Avendo a disposizione il punto iniziale ed il punto finale della traiettoria è possibile calcolarsi una possibile traiettoria e discretizzarla in punti. I punti in cui è stata suddivisa la traiettoria sono utili al calcolo della cinematica inversa in modo tale da ottenere una successione di coordinate di giunto e velocità di giunto che rappresentano la successione di movimenti con rispettive velocità angolari che i giunti del robot compiranno. La successione di posizioni e velocità angolari che i giunti andranno a ricoprire vengono prima controllate dal pianificatore rispetto ai vincoli imposti. I vincoli possono essere dovuti a diversi fattori, ad esempio:

- Evitare che il robot vada a sbattere su se stesso;
- Evitare che il robot vada a sbattere su altri oggetti che sono stati specificati nello spazio di lavoro;

• Assicurarsi che le specifiche del costruttore vengano rispettate in termini di: posizione, velocità ed accelerazione massima che i giunti possono sopportare.

Se il set di posizioni e velocità di giunto calcolati dal pianificatore rispettano tutti i vincoli imposti al pianificatore essi diventano l'output del pianificatore, altrimenti è necessario ripetere il processo da capo: calcolandosi una nuova traiettoria, dei nuovi punti e una nuovo set di punti di posizione e velocità angolare.

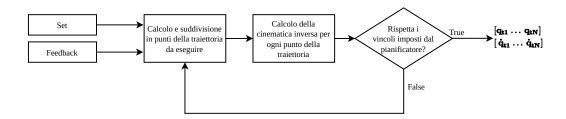


Figura 2.9: Logica di funzionamento del pianificatore di moto

Il pacchetto MoveIt presenta diversi tipi di algoritmi per la pianificazione del moto. In questo elaborato sono stati presi in considerazione solo il pianificatore OMPL[23] e il pianificatore PILZ[24]. Fra i due è stato scelto OMPL poiché riesce a ottimizzare le sue traiettorie tramite teorie matematiche molto più robuste e rappresenta lo stato dell'arte della pianificazione dei movimenti.

Pianficatore OMPL

OMPL, acronimo che sta per Open Motion Planning Library, è un algoritmo di programmazione del moto anche per manipolatori mobili sviluppato da diversi ricercatori. É un pianificatore open source molto potente poiché è in grado di gestire traiettorie di manipolatori che presentano diversi gradi di ridondanza oltre che presentare applicazioni in grado di gestire traiettorie per la guida dei veicoli autonomi. OMPL è un pianificatore probabilistico ovvero cerca una fa le probabili soluzioni al problema, quindi non è per nulla assicurato che produca la stessa traiettoria dell'organo terminale e i medesimi movimenti di giunto nel pianificare il moto per due medesime pose.

Per quanto concerne la pianificazione del moto del manipolatore articolato a 6 gradi di libertà presente in questo elaborato OMPL ha ricoperto il compito di trovare la traiettoria ottimale da far compiere ai 6 giunti robotici affinché l'organo terminale del manipolatore articolato si potesse spostare dalla posa attuale, letta tramite feedback, alla posa richiesta in input.

In Figura 2.10 viene mostrato un esempio di legge del moto di velocità che viene passata ai giunti. Si nota che la legge del moto nei vari giunti che vengono

movimentati è simile, ma scalata in base a quanto è ampio o meno il movimento Δq che interesserà quel giunto. Si nota che il pianificatore fa in modo di far coincidere l'inizio e la fine del moto di ogni giunto quindi identifica il giunto che deve compiere la massima rotazione angolare e lo cercherà di muovere alla massima velocità angolare, a tutti gli altri giunti applicherà la stessa legge del moto del giunto più veloce ma con i valori scalati al fine di ottenere un minore spostamento nello stesso spazio. Da ciò si deduce che OMPL non lavora con la logica dei tempi di movimento diversificati tali da permettere ad ogni giunto di potersi muovere alla massima velocità.

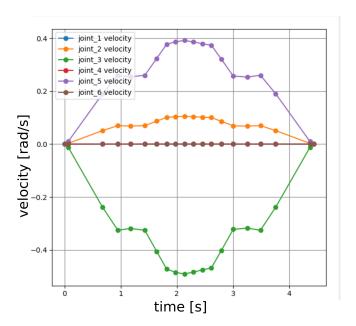


Figura 2.10: Esempio di pianificazione dei movimenti tramite il pianificatore OMPL

Infine OMPL presenta diverse metodologie di ottimizzazione della traiettoria. In letteratura sono presenti diversi parametri di ottimizzazione come: il lavoro compiuto dai giunti, lo spazio percorso o l'energia assorbita. In base a quale di questi parametri si sceglie per effettuare l'ottimizzazione della traiettoria si otterranno traiettorie differenti. OMPL presenta una grande quantità di parametri e routine come alternativa al pianificatore standard consigliato di default.

Pianficatore PILZ

PILZ, sviluppato dalla omonima azienda di automazione, è un pianificatore di movimenti industriale con una struttura più semplice rispetto a quella di OMPL. PILZ privilegia ottimizzare la traiettoria così da ottenere il movimento più rapido possibile fra due punti in spazio operativo. É implementato in modo tale da iniziare e finire il movimento di tutti i giunti simultaneamente. PILZ è un pianificatore deterministico, questo rappresenta un'altra grande differenza da OMPL, quindi date le medesime pose di partenza e fine lui programmerà la medesima traiettoria dell'organo terminale con il medesimo movimento dei giunti.

La legge del moto trapezoidale è una caratteristica di questo pianificatore. Infatti ottenute tutte le coordinate di giunto si assegna alla coordinata di giunto che cambia maggiormente l'accelerazione massima fino al raggiungimento della velocità massima che viene mantenuta per tutto il resto della traiettoria fino in prossimità del fine movimento dove si decelera con decelerazione massima. Tutte le altre leggi dei giunti che hanno una variazione di giunto inferiore seguiranno la stessa legge del moto ma scalata affinché tutti i giunti possano iniziare e finire il loro movimento nello stesso istante.

In Figura 2.11 è mostrato un esempio di legge del moto data con PILZ. Si nota come la discretizzazione del pianificatore lavori su più punti rispetto ad OMPL, inoltre si nota il profilo di velocità trapezoidale tipico di PILZ. Tuttavia se il limite di velocità massima è troppo elevato non si arriva mai a saturare la velocità del giunto. Questo è il caso della Figura 2.12 dove si passa dalla fase di accelerazione a quella di decelerazione e quindi la forma di velocità trapezoidale degenera in una forma di velocità triangolare.

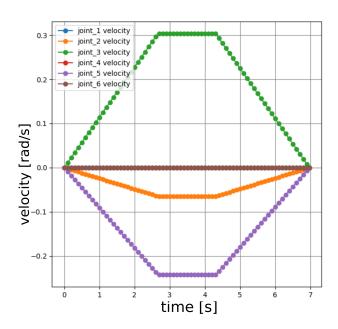


Figura 2.11: Esempio di pianificazione dei movimenti tramite il pianificatore PILZ

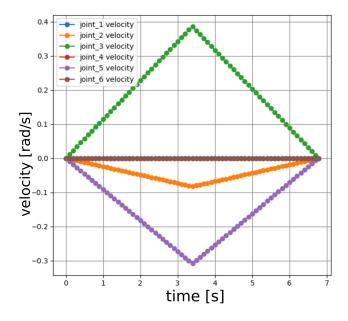


Figura 2.12: Esempio di pianificazione dei movimenti tramite il pianificatore PILZ con trapezio degenerato in triangoli

Capitolo 3

Camera di profondità

In questo capitolo è descritto il funzionamento della camera di profondità, viene fatto il punto sui sensori che la compongono e su come avviene la raccolta dei dati in output dalla camera. É spiegata la logica dell'inizializzazione informatica della camera grazie al quale si riescono ad ottenere ed estrapolare le immagini. Inoltre è stata condotta una campagna di test sull'accuratezza di misura della camera al fine di comprendere la tolleranza che ci si può aspettare dalle sue misure. La camera di profondità utilizzata in questo lavoro di tesi è la Intel L515[25].

3.1 Descrizione della camera di profondità

In Figura~3.1 è mostrata una rappresentazione schematica della camera di profondità, nell'immagine vengono evidenziati i tre principali componenti che caratterizzano la camera:

- Il sensore RGB: il quale acquisisce le immagini come un qualsiasi obiettivo fotografico;
- L'emettitore di onde infrarosso: il quale emette le onde infrarosse ad una specificata lunghezza d'onda compresa fra i $844 875 \ nm$;
- Il ricevitore di onde infrarosse: il cui obiettivo è quello di acquisire le onde infrarosse emesse dall'emettitore in modo da calcolare la distanza degli oggetti.

L'area di acquisizione della matrice RGB e del sensore di profondità sono delle piramidi a base rettangolare (quadrangolare nel caso del sensore di profondità). L'output ottenuto dalla camera sono due matrici:

• La matrice RGB che presenta tre valori, uno per ogni colore, e rappresenta il dato visivo catturato.

• La matrice di profondità presenta un solo valore che descrive pixel per pixel la distanza dal centro di riferimento della camera di quel punto.

Il punto di forza di questa camera di profondità è avere un software interno che si occupa di allineare le due matrici fornite da due sensori che acquisiscono da posizioni che sono necessariamente in posti diversi nella camera. L'allineamento è fatto in modo da avere come sistema di riferimento l'emettitore di onde infrarosse della camera e da ottenere due matrici analoghe: una contenente le informazioni sul colore ed una contenente le informazioni sulla profondità.

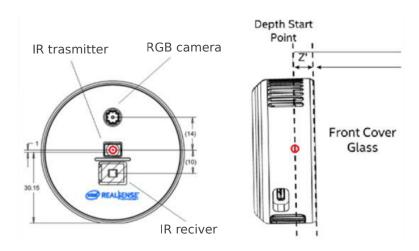


Figura 3.1: Raffigurazione delle dimensioni più importanti della camera

3.1.1 Sistema di riferimento della camera

In Figura 3.2 viene mostrata la disposizione degli assi del sistema di riferimento della camera, esso presenta la particolarità di essere collocato al centro geometrico della camera ad una distanza Z' pari a 4,5 mm dalla superficie esterna in vetro della camera. É facile notare che il sistema di riferimento della camera è posto in maniera tale da presentare l'asse \mathbf{Z} uscente dalla camera, l'asse \mathbf{Y} rivolto verso il basso ovvero verso l'interfaccia filettata della camera utile per fissarla e l'asse \mathbf{X} va di conseguenza a puntare verso il lato destro della camera (se la si guarda rivolti verso la direzione dell'asse \mathbf{Z} uscente).

La conoscenza di come è disposto il sistema di riferimento della camera è molto importante per interpretare il dato della distanza ottenuta in output. Una delle rappresentazioni più efficaci delle misure ottenute dalla camera consiste nel fornire le coordinate del punto rispetto a questo sistema di riferimento solidale con la camera.

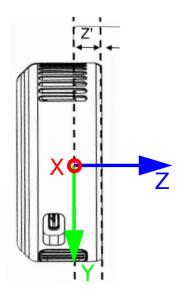


Figura 3.2: Raffigurazione del sistema di riferimento della camera

3.1.2 Campo visivo della camera

In Figura 3.3 è schematizzato il range d'azione dentro al quale è possibile ottenere una misura tramite la camera di profondità. Dalla figura si nota che il campo massimo di acquisizione del sensore di profondità equivale ad un angolo sferico con centro sull'origine del sistema di riferimento della camera. Tuttavia poiché si è interessati ad avere una matrice di misure è importante che si possa avere una misura su tutto un piano, quindi si prende come FOV¹ la maggiore distanza in cui è possibile ottenere su tutti i punti della matrice una distanza. Per la camera in esame viene specificata che il massimo range per il FOV è di 9 m. Infatti eccedendo sopra i 9 m si ottiene solo una misura parziale concentrata tutta su un intorno circolare di punti concentrici al centro dell'immagine, questa misura va a degenerare in un punto nel caso in cui si voglia misurare un oggetto ad una distanza pari al Max Range. Nel caso della camera in oggetto, la quale presenta un angolo di apertura del cono di 70^o , quindi un semi-angolo di apertura di 35^o il Max Range viene calcolato grazie alla relazione trigonometrica che lo lega al FOV: $\frac{9}{\cos 35^o} = 10,98 \ m$.

Analogamente all'occhio umano che non riesce a lavorare correttamente sotto un certo range di misura a causa della distanza fisica fra i due nervi ottici, anche la camera di profondità presenta un limite sulla minima distanza sotto la quale non riesce a prendere le misure. Il minimo range a cui è utilizzabile la camera di

¹Field Of View: ovvero campo di visione, consiste nella visione angolare di una determinata porzione di schermo inquadrato dalla camera

profondità è influenzato dal fatto che il sensore visivo e il sensore di profondità hanno fra di loro una certa distanza fisica. Per la camera di profondità il minimo range è compreso fra i $0.25\ m$ e i $0.2\ m$ con accuratezza sulla misura ottenuta decrescente rispetto al massimo.

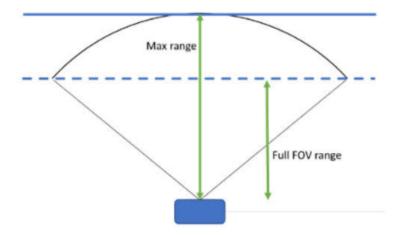


Figura 3.3: Rappresentazione del massimo campo di acquisizione della camera di profondità

3.1.3 Funzionamento del sensore di profondità

Il funzionamento della camera è simile a quello di un sonar: l'emettitore infrarossi produce un insieme di onde elettromagnetiche nel cono dove vengono prese le misure, il ricevitore calcola il tempo trascorso tra quando il segnale è stato mandato e quando è stato ricevuto. Analogamente al sonar, sapendo che la velocità delle onde elettromagnetiche è una costante, il tempo intercorso tra l'emissione e la ricezione dell'onda è correlato con la distanza fra la camera e l'oggetto che ha riflesso l'onda elettromagnetica.

In Figura 3.4 è mostrato un esempio di come il sensore laser della camera funzioni con due tipi di superfici: una liscia ed una rugosa. In entrambi i casi è possibile apprezzare il funzionamento simil sonar che caratterizza questo tipo di applicazione, sono infatti evidenziate le onde emesse dal sensore e le onde riflesse dalla superficie. Da questa immagine si nota come la presenza di una superficie rugosa rende più probabile che le onde elettromagnetiche riflesse siano più diffuse, quindi più omnidirezionali, viceversa una superficie completamente liscia o trasparente vedrà le sue onde riflesse o rifratte ma tutte nella stessa direzione. Queste considerazioni sono alla base dei differenti livelli di rendimento del sensore di misura, infatti per intercettare il riflesso dell'onda elettromagnetica su una superficie è necessario che

l'angolo di incidenza fra la superficie e l'onda emessa sia molto minore rispetto a quanto ci si può permettere con una superficie rugosa. La conclusione di questa comparazione è che una superficie rugosa è più facile da misurare rispetto ad una superficie liscia e la relativa accuratezza sulla misura ottenuta sarà anche maggiore.

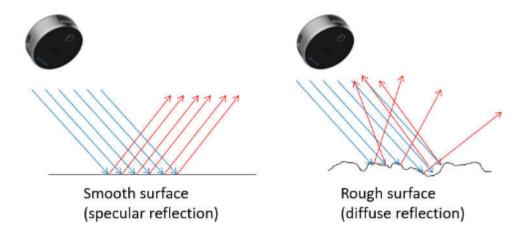


Figura 3.4: Schematizzazione dell'effetto del tipo di superficie sulla riflessione del segnale ad infrarosso emesso

Affinché sia possibile utilizzare la camera in diversi contesti di illuminazione ed in diversi contesti di spazi ambientali, che possono essere ristretti o meno, la camera di profondità presenta diverse modalità di funzionamento. Scegliendo fra una delle diverse modalità in cui si può regolare la camera è possibile ottenere una migliore resa del dispositivo. Di seguito l'elenco delle modalità con cui la camera può essere usata:

- Max Range: questa modalità è utile quando si è in presenza di grandi ambienti non illuminati direttamente dalla luce del sole. Questa modalità impone la potenza del laser e la potenza del ricevitore al massimo, con l'obiettivo di ottimizzare l'emissione e la ricezione di onde elettromagnetiche in un posto non soleggiato.
- Short Range: questa modalità è utile quando la camera interagisce con oggetti a distanza ravvicinata, tuttavia questa impostazione non risulta ottima se oltre agli oggetti vicini si vuole avere una misura accurata anche su quelli più distanti. Questa modalità si occupa di diminuire la potenza di laser e ricevitore in modo tale che gli oggetti vicini non vengano sovra-saturati e quindi non misurati o misurati male.
- No Ambient Light: modalità di utilizzo analoga a Max Range, si rivela utile quando non c'è luce solare nell'ambiente. La differenza tra questa modalità e

Max Range è che qui la potenza di emissione del laser è minore per evitare false distanze segnate da oggetti che sono oltre il limite del visibile della camera.

• Low Ambient Light: è una modalità di utilizzo adatta per ambienti caratterizzati da poca luce solare. É una modalità simile a Max range tuttavia presenta solo l'emettitore di luce infrarossa alla massima potenza, mentre il guadagno del ricevitore è limitato ad un valore più basso per evitare una saturazione dovuta all'effetto combinato di laser e luce solare. Questa modalità è raccomandata anche se si vuole avere a che fare con oggetti vicini, sotto i 0,5 m

3.2 Interfaccia software con la camera di profondità

La camera di profondità svolge un compito che, a differenza del braccio robotico, risulta essere sempre lo stesso a prescindere dall'interfaccia con cui ci si collega. Infatti gli unici input che la camera di profondità riceve sono quelli dovuti alle regolazioni che vengono fatte sulla camera come le dimensioni in pixel dell'immagine oppure le regolazioni per ottimizzare il sensore di profondità in base alla luce e all'ambiente in cui si opera.

I possibili modi di comunicare con la camera sono tramite le API o tramite ROS. Sia le API che ROS possono presentare un'interfaccia sia attraverso il codice scritto in C++ che attraverso il codice scritto in Python. Di seguito sono riportati i vantaggi e gli svantaggi nell'utilizzo dei vari metodi di interfaccia prima elencati:

- Interfacciarsi alla camera in C++, a prescindere che si usino le API o ROS, è sicuramente il metodo con cui si ottiene la maggiore velocità di esecuzione. Tuttavia è anche il metodo più complesso dal lato della programmazione, infatti i codici in C++ richiedono una sintassi più complicata sia per l'interfaccia con la camera sia per la manipolazione delle immagini.
- Interfacciarsi alla camera in Python, a prescindere che lo si faccia attraverso ROS o attraverso le API, è il modo con cui si ottiene una maggiore semplicità nel codice. Infatti da un punto di vista di complessità il codice Python riesce ad essere più compatto, sintetico ed intuitivo di C++. Inoltre Python riesce a gestire in maniera più efficace gli output visivi poiché presenta sviluppate al suo interno diverse librerie per la manipolazione delle matrici, anche matrici di pixel. Tuttavia i codici in Python presentano una velocità di esecuzione leggermente minore rispetto ai codici in C++.
- Usando C++ o Python si può scegliere di elaborare l'output della camera di profondità direttamente tramite le API oppure passando per ROS. La

principale differenza in termini di utilizzo fra questi due metodi è che nel primo caso l'output resta circoscritto al codice che si è implementato, mentre nel secondo caso la camera pubblica il suo output come nodo di ROS quindi tutti gli altri dispositivi presenti nella medesima sessione ROS possono usufruire delle informazioni pubblicate dalla camera. Ciononostante inserire la camera di profondità su ROS risulta un processo complesso che può portare con se complicazioni e talvolta maggiori instabilità nel funzionamento.

Durante il lavoro di tesi sono stati usati come codici alcune delle alternative proposte sopra. In particolare la camera è stata utilizzata in due differenti modi, sfruttando tipologie differenti di codice. Il primo utilizzo è stato per verificare l'affidabilità delle misure prodotte dalla camera: si è utilizzato un codice d'interfaccia con le API, scritto in C++ che elabora i dati in output dalla camera. Lo scopo del codice è quello di produrre e porre un cursore sull'immagine ottenuta il quale viene poi usato per ottenere le coordinate di quel punto indicato dal cursore. Il secondo utilizzo è volto ad ottenere la posa, intesa come posizione ed orientamento, di un oggetto inquadrato dalla camera. Questa applicazione è stata implementata in Python al fine di sfruttare la maggiore maneggevolezza delle matrici intrinseca al linguaggio di programmazione in uso. Infatti affinché si possa ottenere la posa è necessario conoscere tre punti, ovvero altri due punti oltre a quello che si sta scegliendo. Inoltre grazie alle librerie di computer vision, come OpenCV, è possibile selezionare l'oggetto di cui misurare la posa fra quelli riconosciuti dal software di riconoscimento visivo. Anche questa applicazione è stata sviluppata in Python interfacciandosi direttamente con le API della camera. Tuttavia, nell'ottica di costruire un'interfaccia fra camera e manipolatore è stato, creato un nodo ROS al fine di pubblicare la posa dell'oggetto affinché il braccio robotico ne possa disporre.

3.2.1 Inizializzazione della camera di profondità

A prescindere dal linguaggio di programmazione che si utilizza la logica e le procedure di inizializzazione della camera, affinché produca l'output richiesto sono sempre simili.

Nell'Algoritmo 4 è mostrata la logica dietro a tutte le istruzioni preliminari per far sì che la camera si avvii e si mantenga in funzione con le impostazioni stabilite dall'utente come la risoluzione dell'immagine e la modalità con cui usare il sensore di profondità in base alle condizioni di lavoro. Inoltre vengono acquisite le informazioni utili a convertire sia le misure acquisite dal sensore di profondità in misure in metri sia le informazioni caratteristiche della camera utili a trasporre i pixel in punti aventi le tre coordinate: X, Y e Z. In particolare nella prima parte del codice si crea, si configura e si inizializza l'interfaccia con la camera; nella seconda parte del codice vengono specificate ed acquisite tutte le informazioni secondarie fra le quali la maggior parte è riguardante il sensore di profondità.

Algorithm 4 Logica dell'algoritmo per l'inizializzazione della camera

```
Require: the API library of depth camera, math and matrix manipulating
Ensure: configure the video stream from camera with an RGB and a depth matrix
  pipeline \leftarrow camera\_interface.pipeline
                                              ▷ Create a pipeline for stream image
  config \leftarrow camera\_interface.config  \triangleright Create a config struct to configure the
  pipeline
  config.enable\_stream(camera\_interface.depth\_sensor, res.width,
                          res.height, camera interface.format, frame rate)
  config.enable\_stream(camera\_interface.color\_sensor, res.width,
                          res.height, camera\_interface.format, frame\_rate)
  profile \leftarrow pipeline.start(confiq)
                                           > Start streaming the pipeline with the
  configuration choosen
  align to \leftarrow camera interface.stream.color
  align \leftarrow camera \quad interface.align(align \quad to)
                                                      by this option is used to align
  the depth sensor to the color sensor, in this way all the RGB pixel will have the
  correct depth pixel
  depth\ sensor \leftarrow profile.get\_device().first\_depth\_sensor()
                                                                     ▶ Initialize the
  variables that interfaces with configuration of depth sensor
  depth\_sensor.set\_option(camera\_interface.visual\_preset,
                          SHORT\_RANGE)
                                                   ▶ In order to put the camera in
  short vision mode
  depth\_scale \leftarrow depth\_sensor.get\_depth\_scale
  intrd \leftarrow profile.qet \ stream(camera \ interface.stream.color).qet \ intrinsics
  ▶ In order to achieve information useful to obtain the true depth of the depth
  sensor, throught the depth scale, and useful to deproject the pixel to a point
  with coordinates
```

Una volta inizializzata la camera non resta che eseguire un ciclo che acquisisce sempre la nuova immagine, la processa e la mostra. Nell'*Algoritmo 5* è mostrata la logica di un algoritmo che si occupa di acquisire il dato più nuovo in uscita dalla camera e mostrarlo a video. Infatti dopo aver acquisito le nuove informazioni l'algoritmo si occupa di post-processarle, ovvero:

 Allineare l'informazione della distanza con quella del colore come specificato nelle impostazioni iniziali;

- Separare in variabili diverse l'informazione sul colore e l'informazione sulla distanza, inoltre sistemare le informazioni in strutture matriciali che possono essere meglio manipolate;
- Moltiplicare la misura grezza ottenuta dal sensore di profondità con l'opportuna scala al fine di ottenere misure di profondità misurabili in metri;
- Mostrare in una finestra del sistema operativo l'acquisizione ottenuta in termini di matrice.

Si fa notare che in questo esempio si è voluto mostrare l'output ottenuto dal sensore visivo, mantenendo nascosto l'output ottenuto dal sensore di profondità, ma si poteva benissimo fare l'opposto. Inoltre pur mostrando solo l'output visivo l'informazione sulla profondità resta accessibile a condizione di sapere qual è il pixel o l'insieme di pixel di cui si vuole sapere la profondità

Algorithm 5 Algoritmo che permette di mostrare sempre la nuova immagine acquisita

Require: activated and configured pipeline of camera's API, OpenCV library Ensure: obtain a video stream from camera with an RGB matrix and a depth matrix

```
while not shutdown do
```

 $frames \leftarrow pipeline.wait_for_frames$ \triangleright Get the newest frames of color and depth matrix

 $aligned_framse \leftarrow align.process(frames) \triangleright Align the color and the depth frames on the base of the previous settings$

```
depth\_frame \leftarrow aligned\_frames.get\_depth\_frame \\ color\_frame \leftarrow aligned\_frames.get\_color\_frame \rhd \text{Get the two aligned frames} frames
```

```
depth\_image \leftarrow matrix\_array(depth\_frame.get\_data) depth \leftarrow depth\_image * depth\_scale color\_image \leftarrow matrix\_array(color\_frame.get\_data) \rhd \text{Color} \text{ and depth} information acquired in matrix form
```

```
cv2.nameWindow('L515\ Video') \\ cv2.imshow('L515\ Video', color\_image) \qquad \triangleright \ Print \ an \ image \ with \ OpenCV \\ library
```

end while

pipeline.stop()

3.2.2 Descrizione delle funzioni principali

Nel ciclo while della funzione che permette di mostrare sempre il nuovo set di dati, ne viene mostrato un esempio nell'*Algoritmo 5*, è possibile richiamare e inserire altre funzioni al fine di utilizzare i dati appena acquisiti. Le funzioni che sono state utilizzate in questo lavoro di tesi si possono riassumere in tre tipi:

- Acquisire le coordinate cartesiane di un punto;
- Sfruttare tre punti in coordinate cartesiane per ottenere la posa di un corpo;
- Usare librerie per il computer vision al fine di riconoscere oggetti noti nell'inquadratura della camera.

La posizione di queste funzioni e istruzioni dipende molto dai dati che necessitano in input. Per ottenere la posizione o la posa di un corpo è necessario conoscere un pixel significativo che viene occupato da questo corpo, di solito uno dei pixel centrali raffiguranti il corpo, inoltre è necessario avere i dati ottenuti dalla matrice di profondità e i dati intrinseci della camera. Viceversa per usare un algoritmo di computer vision è necessario solamente avere la matrice RGB in output dalla camera e post-processarla con una opportuna rete neurale, la rete neurale va caricata una sola volta prima del ciclo while.

Algoritmo per ottenere le coordinate e la posa di un corpo

Lo scopo di questo algoritmo è quello di ottenere delle coordinate cartesiane di un punto ricevendo in input la posizione di un pixel, la distanza associata a quel pixel e le informazioni intrinseche delle camera.

Per ottenere la coordinata del pixel in input all'algoritmo:

- viene usato l'output delle funzioni di computer vision, i quali restituiscono una regione di pixel dove è presente l'oggetto di interesse, di cui viene preso il pixel centrale;
- è presente una funzione di callback che accetta in input il click del mouse sul frame e restituisce come variabile l'avvenuta pressione dello specifico pulsante del mouse e le coordinate in pixel su cui ci si trovava.

La funzione di callback non viene descritta in questo elaborato poiché è una funzione propria del sistema operativo e della libreria di riconoscimento immagine, in questo ambito viene solo usata ma senza entrare nel merito del funzionamento.

Nell'Algoritmo 6 è mostrato il modo in cui si passa da un'informazione in pixel e distanza in una in coordinate cartesiane rispetto il sistema di riferimento della camera. Si ottiene velocemente la misura della distanza lungo le due coordinate

X ed Y dal centro della camera. Per ottenere ciò viene usata una formula matematica suggerita dal costruttore della camera [26], in tale formula sono usate le caratteristiche intrinseche della camera e i dati misurati dai sensori stesi della camera:

- Le lunghezze focali: fx ed fy;
- Le coordinate in pixel del centro del frame: ppx e ppy;
- La distanza in metri misurata fra la camera ed il pixel: depth;
- Le coordinate *pixel_x* e *pixel_y* del punto di cui si vogliono sapere le coordinate.

Algorithm 6 Funzione che permette di calcolare le coordinate di un punto

 $\textbf{Require:}\ pixel_x,\ pixel_y,\ depth,\ camera_intr$

Ensure: Return the X, Y, Z coordinates of pixel point

 $X = (pixel_x - camera_intr.ppx)/camera_intr.fx * depth$

 $Y = (pixel_y - camera_intr.ppy)/camera_intr.fy * depth$

▶ This

matematical equation are the one who describes the X and Y position according to the constructor website [26]

return X, Y, depth

L'algoritmo che permette di ottenere la posa di un oggetto si ottiene come un evoluzione dell'algoritmo che permette di ottenere le coordinate cartesiane di un punto. Vengono selezionati altri due punti oltre a quello indicato in input, i due punti sono scelti in maniera tale da formare due segmenti ortogonali tra di loro che rappresentano rispettivamente il versore \mathbf{x} ed il versore \mathbf{y} del sistema di riferimento dell'oggetto di cui si vuole conoscere la posa. In questa tipologia di lavoro ci si è imbattuti in due diversi tipologie di definizione del sistema di riferimento dell'oggetto osservato. Si può infatti avere un sistema di riferimento che ruota con l'oggetto o un sistema di riferimento fisso.

Il sistema di riferimento che ruota con l'oggetto è possibile solo se la libreria di computer vision sta riconoscendo un oggetto di forma univoca di cui può facilmente riconoscere tutti i bordi e quindi ha una certezza sull'orientamento dell'oggetto riconosciuto. In quel caso, affidandosi ai punti riconosciuti dal software, se ne selezionano tre in modo tale che si ottengano due segmenti con angolo retto. Questa metodologia è stata adottata nel caso degli ArUco Markers mostrati in Figura 3.5. In questo caso gli assi \mathbf{x} e \mathbf{y} sono stati messi in modo tale che l'asse \mathbf{z} risultasse uscente.

Il sistema di riferimento fisso si usa nel caso in cui si ha una libreria di computer vision che riconosce gli oggetti solo se disposti in una direzione, quella in cui si è più frequentemente avvezzi a vederli, come in Figura 3.6; oppure se si utilizza il mouse per specificare l'oggetto e quindi il pixel su cui misurare la posa. Ciò è stato possibile poiché se l'oggetto viene riconosciuto dal software di computer vision allora esso si trova con un sistema di riferimento molto simile a quello fisso usato per la misura della posa, invece se è una persona a riconoscere l'oggetto si farà anche carico di selezionare gli oggetti che possono essere compatibili con la posa misurata. Il sistema di riferimento in questo caso viene scelto con l'asse x rivolto verso sinistra e l'asse y rivolto verso l'alto in modo da presentare l'asse z rivolto verso l'interno dell'oggetto da manipolare. Nel capitolo precedente si è visto che il manipolatore del robot presenta l'asse z uscente (Figura 2.5), dunque per motivi di semplicità si è scelto di misurare una posa con un asse z coerente con quello del manipolatore.

Nell'Algoritmo 7è mostrato un esempio di come si presenta la logica di un codice che calcola la posa di un corpo partendo da un punto dato in input e utilizzando un sistema di riferimento fisso per il cacolo della posa. L'algoritmo si riduce al calcolo delle coordinate dei tre punti, che vengono inseriti nei vettori Center, Left e Top. Ricavate le coordinate dei tre punti per differenza si calcolano i vettori X ed Y. Tramite l'opportuno prodotto vettoriale si ottiene pure il vettore Z. Infine i vettori acquisiti vengono normalizzati al fine di rappresentare il prodotto scalare tra i vettori del sistema di riferimento da misurare e quello fisso della camera che misura. Una volta normalizzati vanno a confluire nella matrice di trasformazione, popolando la matrice di rotazione, insieme alle coordinate del centro del sistema di riferimento della posa appena misurato ovvero il vettore Center.

Algoritmi di computer vision

Esistono diversi algoritmi di computer vision i quali si occupano di ottenere approssimativamente lo stesso risultato, ma con procedimenti diversi. In questo elaborato è stato scelto di utilizzare OpenCV[27][28] poiché presenta i seguenti vantaggi rispetto alle librerie con scopi analoghi presenti sul mercato:

- É una libreria open source;
- Assicura il funzionamento real-time, con un numero di frame per secondo elaborati superiore a 24(numero minimo sopra il quale al nostro occhio appare come un unico filmato);
- Assicura una facilità di utilizzo notevole.

Proprio per la facilità di utilizzo non è il caso di presentare una logica di funzionamento di questo algoritmo, tuttavia di seguito sono stati riportati i pochi passaggi

Algorithm 7 Funzione che permette di calcolare la posa di un corpo

Require: pixel_x, pixel_y, depth(whole matrix), camera_intr

Ensure: Return the pose matrix of the object

 $Center \leftarrow calculate_coord(pixel_x, pixel_y, depth(pixel_x, pixel_y),$

 $camera_intr)$

 $Left \leftarrow calculate_coord(pixel_x - dist_pix, pixel_y,$

 $depth(pixel_x - dist_pix, pixel_y), camera_intr)$

 $Top \leftarrow calculate_coord(pixel_x, pixel_y - dist_pix,$

 $depth(pixel_x, pixel_y - dist_pix), camera_intr)$

▶ This is an example of calculanting the coordinates of tre fixes point for obtain pose

 $X \leftarrow Left - Center$

 $Y \leftarrow Top - Center$

 $Z \leftarrow X \times Y$ \triangleright In this way the tree vector of reference system are obtained

 $X \leftarrow \frac{X}{|X|}$

 $Y \leftarrow \frac{Y}{|Y|}$

 $Z \leftarrow \frac{Z}{|Z|}$ > Normalizing the tre vector in order to obtain the versors. The versors are usefull to represent the orientation of this pose vectors respect to the versor of the camera reference system

$$\begin{aligned} Matrix \leftarrow \left[\begin{array}{cccc} X & Y & Z & Center \\ 0 & 0 & 0 & 1 \end{array} \right] \\ return \ Matrix \end{aligned}$$

che servono per ottenere le informazioni processate da un software di computer vision.

Nel caso di una rete neurale qualsiasi prima del ciclo while, dove si interroga la camera costantemente per ottenere il dato più aggiornato possibile, viene caricata e specificata la tipologia di rete neurale con cui si ha a che fare. Nei casi di reti neurali più complesse viene specificato:

- Il nome degli oggetti che possono essere riconosciuti;
- I due file weightsPath e configPath che sono gli attori principali per il riconoscimento degli oggetti;
- Altre piccole impostazioni che sono consigliate nel forum del sito ufficiale.

Nel ciclo while viene inserita l'istruzione utile a confrontare l'immagine acquisita dalla camera con il set di immagini della rete neurale caricata, il confronto tra le due permette di effettuare il riconoscimento di immagini. Questa funzione

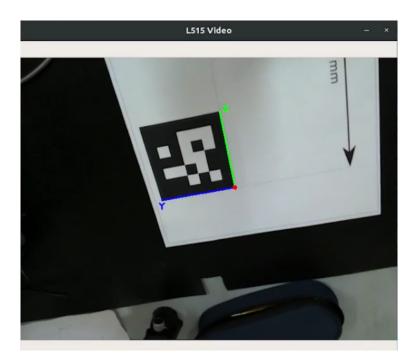


Figura 3.5: Esempio di un sistema di riferimento posto su un ArUco Marker



Figura 3.6: Esempio di una bottiglia riconosciuta tramite OpenCV

preposta al riconoscimento prende in input l'immagine uscente dalla camera e la soglia di affidabilità con la quale stabilire se un oggetto inquadrato corrisponde o meno a qualcuno del set che ha con se la rete neurale. Dopo aver effettuato la comparazione la funzione dà in output: l'identificativo dell'oggetto riconosciuto, il grado di affidabilità del riconoscimento e gli angoli in pixel dove l'oggetto è contenuto. Dall'identificativo si ottiene il tipo di oggetto che è stato riconosciuto, mentre dalla posizione in pixel si ottiene un punto in pixel utile per la misura della posa. In Figura 3.6 è mostrato un esempio di una bottiglia riconosciuta tramite questo tipo di rete neurale.

Nel caso si debba identificare un ArUco Marker l'algoritmo di identificazione non prevede il caricamento di una libreria, quindi risulta sprovvisto delle istruzioni da compiere prima del ciclo while. Questo perché nativamente OpenCV ospita un algoritmo per il riconoscimento di questa tipologia di immagini. Dunque occore specificare all'interno del ciclo while:

- il tipo di ArUco Marker bisogna identificare;
- in quale immagine cercare il Marker, in questo caso i frame in uscita dalla camera.

Terminato il processo di identificazione la libreria restituisce in output le coordinate in pixel che racchiudono il marker, l'identificativo e l'affidabilità su quanto identificato. In *Figura 3.5* è mostrato un esempio di un ArUco Marker riconosciuto tramite questo tipo di rete neurale.

3.3 Test preliminari

Sono state condotte delle prove sperimentali per verificare l'affidabilità delle misure della camera di profondità. Per ottenere ciò sono stati presi ripetutamente e a diverse distanze le coordinate di nove punti noti per confrontarli con i punti acquisiti. Nella *Tabella 3.1* sono riportati i dati di accuratezza media e deviazione standard della misura forniti dal costruttore, questi dati sono stati confrontati con quelli acquisiti nella campagna di prove.

Tipo di misura	Valore numerico
Accuratezza media nella misura (@ 1 m) [mm]	5
Deviazione standard della misura (@ 1 m) $[mm]$	2,5

Tabella 3.1: Accuratezza e deviazione standard forniti dal costruttore

3.3.1 Set-up sperimentale dell'ambiente per il test

Al fine di poter effettuare il test in maniera certa e ripetibile sono stati utilizzati i seguenti strumenti:

- camera di profondità;
- livella planare;
- treppiedi;
- schermo con i punti noti da acquisire.

In Figura 3.7 è visualizzata una rappresentazione schematica di come è stato impostato l'ambiente sperimentale al fine di ottenere l'insieme di misure che compongono il campione da validare. La camera è mantenuta parallela al suolo grazie all'utilizzo della livella planare ed è posizionata all'altezza del centro dello schermo grazie all'utilizzo di un treppiedi. La distanza fra la camera e lo schermo viene variata al fine di investigare l'accuratezza della camera all'aumentare della distanza dall'obiettivo da misurare. Inoltre al variare della distanza è stata adeguata la dimensione dello schermo, quindi al fine di avere un set di nove punti facilmente distinguibili nell'inquadratura ad una certa distanza si sono raddoppiate tutte le misure. In Figura 3.8 è mostrata la disposizione della griglia dei nove punti con anche le misure che li caratterizzano, in particolare la dimensione del quadrato resta costante mentre la distanza fra due quadrati è funzione della dimensione dello schermo. Infatti al variare della distanza della camera dallo schermo dove sono contenuti la griglia di nove punti che fa da obiettivo vengono distanziati pure i nove punti stessi, in Tabella 3.2 sono specificate le due coppie di distanze possibili a seconda che si tratti di uno schermo grande o di uno schermo piccolo.

In Figura 3.9 sono mostrate delle foto di come realmente è stato predisposto il set-up sperimentale. Si nota la camera di profondità fissata sopra la livella planare e montata a sua vola sopra il treppiedi, in fondo si vede lo schermo con la griglia di punti noti. In questo caso si sta lavorando con una distanza sotto il mezzo metro, quindi si ha il preset SHORT RANGE e la griglia di misura piccola.

Tipo griglia	Distanza [m]	a [mm]	b [mm]
Piccola	≤ 0.5	105	148.5
Grande	≥ 0.5	210	297

Tabella 3.2: Sunto delle distanze utilizzate per spaziare la griglia di nove punti usata le la validazione della camera

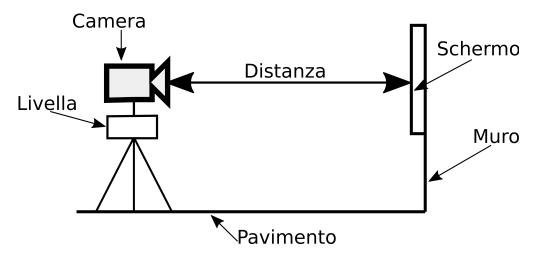


Figura 3.7: Schema grafico della disposizione degli oggetti atti ad acquisire le misure per validare la camera

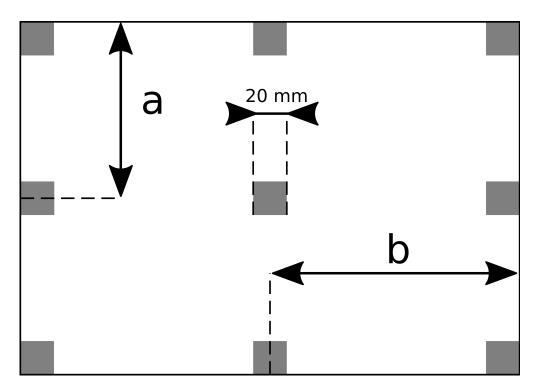


Figura 3.8: Rappresentazione della griglia di nove punti usati per la misura dell'accuratezza della camera



Figura 3.9: Immagini tratte dalla prova sperimentale dove si apprezza il set-up sperimentale

3.3.2 Test sperimentali e risultati ottenuti

La modalità con cui sono state eseguite le misurazioni è stata pensata con lo scopo di evitare di ripetere un set di misure due volte nella stessa posizione. Questa attenzione è stata pensata per evitare di inficiare la statistica mantenendo lo stesso errore di montaggio della camera rispetto allo schermo su due acquisizioni diverse di misure.

La campagna di acquisizione dei dati si è svolta secondo le seguenti indicazioni:

- Si posiziona la camera in una delle distanze prescelte;
- Si mette in bolla la livella planare;
- Si controlla che la camera sia allineata con il centro della griglia dei nove punti tramite la vista del cursore dalla finestra di interfaccia;
- Si acquisiscono le coordinate dei nove punti;
- Si ripete la procedura per tutte le distanze prestabilite.

Questa procedura, che descrive come acquisire i nove punti della griglia per tutte le distanze prescelte, è stata ripetuta 10 volte al fine di avere un campione statistico affidabile.

L'unica variazione degna di nota fra le varie procedure di acquisizione delle misure è dovuta alla impostazione del sensore di profondità, infatti ogni volta si attraversa la distanza di $0.5\ m$ è necessario modificare la sensibilità del sensore per

ottenere una migliore lettura dei dati. Nel dettaglio per distanze superiori al $0,5\ m$ è stato utilizzato l'impostazione LONG_RANGE, viceversa per distanze inferiori al $0,5\ m$ è l'impostazione utilizzata è stata di SHORT_RANGE. Avendo notato che quando la camera è a mezzo metro di distanza dal suo obiettivo si riesce agevolmente ad usare come obiettivo sia lo schermo grande, sia lo schermo piccolo e che si possono acquisire misure affidabili con il preset di LONG_RANGE ma anche con SHORT_RANGE dunque sono state condotte più misurazioni a questa distanza al fine di apprezzare meglio la differenza fra i vari tipi di misura.

In *Tabella 3.3* sono riassunte le varie distanze prescelte per la misura, l'impostazione utilizzata per acquisire le informazioni e la dimensione della griglia dei punti utilizzata (le dimensioni effettive sono riportate nella *Tabella 3.2*). Essendo le acquisizioni effettuate con iterazioni di 10, per ogni riga della tabella sono presenti 10 misure uguali per ogni punto della griglia.

Distanza [m]	Metodo utilizzato	Griglia punti
1,5	LONG_RANGE	Grande
1	LONG_RANGE	Grande
0,5	LONG_RANGE	Grande
0,5	SHORT_RANGE	Grande
0,5	SHORT_RANGE	Piccola
0,4	SHORT_RANGE	Piccola
0,3	SHORT_RANGE	Piccola

Tabella 3.3: Sintesi delle distanze prescelte con relativa impostazione del sensore di profondità e tipo di griglia di punti utilizzata per le acquisizioni

I dati acquisiti vengono elaborati sotto forma di grafici significativi per estrapolarne al meglio le informazioni ottenute. Per ogni tipologia di misura effettuata si è mostrata la dispersione dei vari punti acquisiti, poi viene calcolato un punto medio fra tutte le misure acquisite e lo si confronta con la misura corretta. In Figura 3.10 è mostrato un esempio di come vengono rappresentate le dieci posizioni acquisite con relativa media e posizione corretta, questo tipo di visualizzazione è stata presentata per ognuna della tipologia di misure acquisite in Tabella 3.3. É importante notare che viene selezionata la vista nel piano xy poiché rappresenta lo schermo dove è posta la griglia da misurare.

Successivamente i dati sono stati raggruppati in modo da visualizzare efficacemente la media per ogni set di misura e la deviazione standard sui vari assi. In $Figura\ 3.11$ sono rappresentati i dati delle acquisizioni fatte con preset LONG_RANGE, la griglia di punti grande e le distanze fra camera e schermo di 0.5, $1\ e\ 1.5\ m$. Da questa immagine si nota come le distanze minori presentino una minore deviazione standard sul piano xy ovvero quello della griglia di misura, si nota ancora che

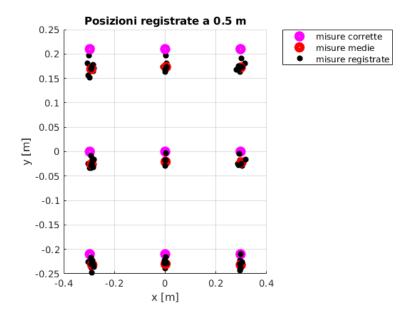


Figura 3.10: Grafico che mostra le posizioni registrate con una acquisizione a 0.5 m, in modalità LONG RANGE e con la griglia dei punti grande

i punti misurati lateralmente hanno degli errori statistici più rilevanti di quello centrale. Infine in questo set di dati si nota un lieve errore sistematico compiuto durante l'acquisizione poiché le coordinate medie risultano tutte spostate verso il basso. Anche gli altri set di dati acquisiti si prestano a considerazioni analoghe e soprattutto si prestano a questo tipo di elaborazione.

Le considerazioni su media della posizione e deviazione standard sono state effettuate anche per le misure ottenute sull'asse \mathbf{z} . In Figura 3.12 è mostrata l'interpolazione delle misure medie acquisite lungo l'asse \mathbf{z} per una misura effettuata a 0,5 m con il preset LONG_RANGE e la griglia di misura grande. Anche in questo caso si nota una ottima accuratezza sulla zona centrale, mentre nei bordi la misura si fa meno accurata, l'errore resta comunque nell'ordine di grandezza dei millimetri. Questo trend di scarsa accuratezza può anche essere dovuto ad un errore sistematici di parallasse fra la camera e la griglia delle misure infatti in altre elaborazione si ripresenta simile.

L'analogo all'elaborazione del piano medio sull'asse \mathbf{z} è stato effettuato anche con la deviazione standard. In Figura 3.13 è mostrata un'interpolazione delle deviazioni standard ottenute su un una griglia di misura grande, ad una distanza di 0.5~m dalla camera e con un preset LONG_RANGE. Anche in questa elaborazione resta costante il trend per cui al centro le misure sono più precise e ripetibili, mentre ai bordi queste caratteristiche vanno scemando. Queste considerazioni, come nel caso delle precedenti analisi, sono state estese a tutti i set di dati acquisiti.

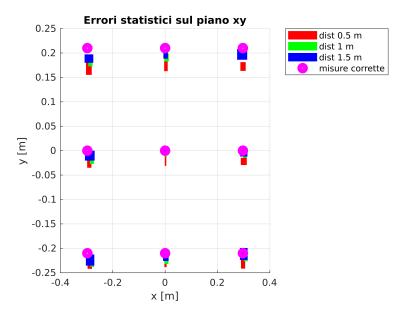


Figura 3.11: Rappresentazione del punto medio e della deviazione standard di tutte le misure effettuate fra i 0.5 e i 1.5 m, in modalità LONG_RANGE e con la griglia dei punti grande

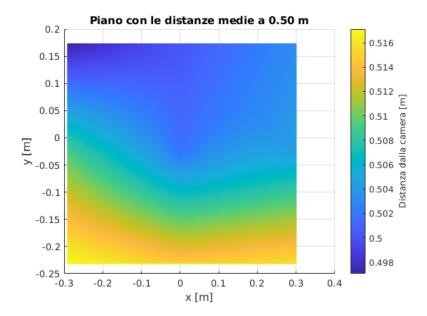


Figura 3.12: Rappresentazione della posizione media della distanza rispetto alla camera della griglia di punti acquisiti. Configurazione con griglia grande, a $0.5\ m$ dalla camera e preset LONG_RANGE

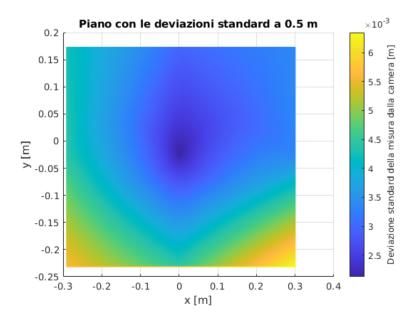


Figura 3.13: Rappresentazione della deviazione standard della distanza rispetto alla camera della griglia di punti acquisiti. Configurazione con griglia grande, a 0.5 m dalla camera e preset LONG_RANGE

3.3.3 Commento dei risultati ottenuti

In *Tabella 3.4* sono sintetizzati i dati ottenuti come accuratezza e deviazione standard delle varie prove eseguite. Nella comparazione fra questi dati e quelli dichiarati dal costruttore in *Tabella 3.1*.

Si nota che rispetto a quanto dichiarato dal costruttore l'accuratezza ottenuta dalle misure sperimentali risulta mediamente di un ordine di grandezza superiore, alcune misure arrivano ad essere quattro volte meno accurate di quanto dichiarato dal costruttore. Una delle possibili cause di questa mancanza di accuratezza è da ricercare sicuramente negli errori sistematici compiuti durante l'acquisizione sperimentale delle misure e sicuramente evitati dal costruttore.

Per quanto riguarda la deviazione standard invece si nota che il dato è più in linea con quanto dichiarato dal costruttore, quasi tutte le misure sono in quell'ordine di grandezza eccetto una, mentre un'altra misura è simile al dato ricavato dal costruttore. Da questo si deduce che il set di dati acquisiti ha una buona ripetibilità, ma si conferma l'errore di accuratezza ottenuto precedentemente il quale è da ricercarsi fra gli errori sistematici presenti nella prova sperimentale.

Complessivamente si può constatare che il dato fornito dal costruttore sia in linea con quanto acquisito sperimentalmente, certo è che le misure dichiarate dalle prove fatte dal fabbricatore sono ripetibili solo in condizioni severamente controllate,

mentre per un uso più sperimentale fanno sicuramente fede le misure ottenute in laboratorio.

Dist [m]	Metodo	Griglia	Accuratezza [mm]	Dev std [mm]
1,5	LONG_RANGE	Grande	12	9
1	LONG_RANGE	Grande	18	6
0,5	LONG_RANGE	Grande	18	6
0,5	SHORT_RANGE	Grande	20	2,6
0,5	SHORT_RANGE	Piccola	7	5,5
0,4	SHORT_RANGE	Piccola	6	6
0,3	SHORT_RANGE	Piccola	10	10

Tabella 3.4: Sintesi dei valori di accuratezza e deviazione standard ottenuti dalla misura della griglia di punti con la camera

Capitolo 4

Implementazione del sistema: manipolatore articolato e sistema di visione

In questo capitolo è descritto come avviene l'interazione ed il montaggio reciproco fra la camera di profondità e il manipolatore articolato. Sono spiegati i motivi che hanno portato ad adottare una soluzione di montaggio piuttosto che un'altra. É anche spiegato il metodo di calibrazione grazie al quale si riesce a trasferire l'informazione acquisita dalla camera in una informazione utilizzabile dal manipolatore articolato. Il processo di calibrazione si prefigge di cercare e trovare la migliore matrice di trasformazione che riesca ad esprimere efficacemente il passaggio dal sistema di riferimento della camera al sistema di riferimento del manipolatore. Attraverso questo processo viene calcolata la matrice di trasformazione dal sistema di riferimento del sistema visivo al sistema di riferimento del manipolatore, il quale permette l'interazione e lo scambio di informazioni efficace tra i due quindi permettendo di lavorare effettivamente come un unico sistema.

4.1 Definizione del problema

Montare un dispositivo di visione su un manipolatore articolato si presta a diverse considerazioni riguardo il posizionamento relativo fra camera e braccio. In generale lo scopo per cui si ricerca l'integrazione di un dispositivo di visione con un dispositivo di manipolazione consiste nel manipolare oggetti che vengono riconosciuti dal sensore visivo. Il modus operandi prevede che prima si identifica l'oggetto, poi

si misura la posizione rispetto alla camera e infine la posizione viene trasformata secondo il sistema di riferimento del manipolatore articolato così che può essere trasmessa, tramite interfaccia informatica, al manipolatore articolato. Per questo fine viene utilizzata la matrice di calibrazione, ovvero il mezzo atto a trasformare la posizione relativa dal sistema di riferimento della camera al sistema di riferimento del giunto del manipolatore a cui la camera viene fissata. Tuttavia la matrice di calibrazione rappresenta il fattore di maggiore incertezza nel posizionamento relativo fra camera e manipolatore articolato, poiché la posizione del giunto rispetto alla base del robot e la posizione dell'oggetto da manipolare rispetto alla camera sono più affidabili poiché testati anche dai rispettivi costruttori. Le variazioni a cui è soggetta la suddetta matrice sono: la posizione relativa sul giunto e l'orientamento fra il giunto e la camera. Entrambe queste variazioni vengono misurate e tenute in considerazione nel processo di calibrazione in modo tale da ottenere una matrice di trasformazione quanto più accurata possibile. Esplicando i fattori da cui le variazioni nel montaggio della camera dipendono si ha che il primo è la configurazione scelta per il montaggio della camera rispetto al braccio e il secondo dipende da piccole variazioni nel montaggio della camera stessa. Una volta ottenuta una matrice di calibrazione dall'algoritmo si ha che un'ulteriore variazione genera ripercussioni che inficiano il risultato precedentemente ottenuto nella matrice di calibrazione. Pertanto è opportuno avere un algoritmo che calcola la matrice di calibrazione in modo veloce, efficiente e facilmente utilizzabile in modo da poter ricalibrare ogni qual volta ci si rende conto che il sensore visivo ha subito una qualche piccola variazione che invalida la matrice di calibrazione precedentemente calcolata.

Per quanto riguarda le diverse configurazioni che si possono ottenere cambiando la disposizione fra manipolatore e sensore visivo è fattuale che ognuna di esse presenta differenti peculiarità. In base al tipo d'impiego che si vuole ottenere dal sistema sono state vagliate diverse posizioni in cui montare la camera rispetto al manipolatore. Sebbene sono possibili vari modi di posizionare il sensore visivo rispetto al manipolatore, questi si possono tuttavia classificare in due grandi famiglie in cui è possibile montare la camera rispetto al manipolatore: eye-in-hand e eye-to-hand.

4.1.1 Confronto fra camera eye-in-hand e eye-to-hand

Montando la camera in configurazione eye-in-hand, come mostrato a sinistra nello schema di Figura 4.1, il sistema si presenta con la camera montata su uno dei giunti del polso del manipolatore articolato. Grazie a questo tipo di montaggio si ha una visione precisa su quello che si sta manipolando, tuttavia è imparziale sull'insieme della scena. Ad esempio ipotizzando un task di manipolazione con ricerca dell'oggetto da manipolare in configurazione eye-in-hand si deve ricorrere ad alcuni movimenti del manipolatore articolato al fine di osservare tutti gli

oggetti presenti nella scena, una volta visto l'oggetto da manipolare l'organo terminale della camera ci si può avvicinare e può manipolarlo. Avvicinando l'organo terminale alla camera si avvicina anche il sistema di visione alla camera con conseguente miglioramento delle informazioni ottenute dal sensore, ciò permette una manipolazione più fine e precisa.

Utilizzando la configurazione di eye-to-hand, mostrata a destra nello schema di Figura 4.1, il sistema si presenta con la camera montata esternamente alla struttura del manipolatore articolato. Grazie a questo tipo di configurazione si può avere una migliore percezione globale della scena in cui il manipolatore si sta muovendo, tuttavia si ha una minore risoluzione delle informazioni nei movimenti di precisione per i compiti di manipolazione. Ad esempio ipotizzando un task di manipolazione con ricerca dell'oggetto da manipolare in configurazione eye-to-hand si ha una più ampia vista della scena in cui il manipolatore articolato opera con una facilità di riconoscimento degli oggetti da manipolare. Tuttavia avvicinando l'organo terminale alla camera l'inquadratura resta fissa, quindi molto globale, questo non aiuta nel miglioramento delle informazioni ottenute dal sensore per effettuare una manipolazione più fine e precisa.

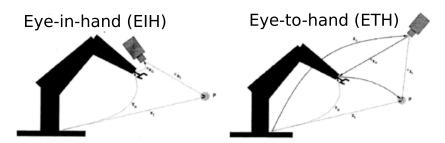


Figura 4.1: Schema comparativo fra una sensore visivo montato in configurazione eye-to-hand ed un sensore visivo montato in configurazione eye-in-hand rispetto al medesimo manipolatore articolato

4.1.2 Configurazione adottata

In Figura 4.2 è mostrata la configurazione che è stata attuata nel sistema oggetto del lavoro di tesi. Il sensore visivo è stato montato utilizzando una configurazione del tipo eye-in-hand al fine di avere i vantaggi di visione accurata durante la manipolazione dei corpi tramite l'organo terminale. Tuttavia sono state prese altre accortezze le quali sono figlie della configurazione del polso che è montato sul manipolatore articolato in uso.

Dalla Figura 4.2 si nota che il canotto dove è alloggiata la camera di profondità è alloggiato sul penultimo link lasciando così libero il grado di libertà di rotazione dell'organo terminale, ovvero il giunto sei, che permette alle pinze di ruotare su sé

stesse senza far ruotare anche la camera. Un'altra scelta di montaggio effettuata consiste nella posizione del link cinque in cui viene alloggiato il canotto. Infatti il link cinque si estende fra il giunto cinque e il giunto sei, in questa applicazione è stato scelto di montare la camera di profondità sul link cinque in prossimità del giunto sei. La vicinanza al giunto sei permette di mantenere una visuale ravvicinata con l'organo terminale e quindi l'oggetto da manipolare in ottica di un montaggio eye-in-hand.

In Figura 4.3 è mostrata la configurazione adottata del manipolatore con i vari sistemi di riferimento notevoli usati successivamente per la calibrazione e la movimentazione:

- il sistema riferimento solidale al giunto 5;
- il sistema di riferimento fisso solidale col manipolatore;
- il sistema solidale all'oggetto da manipolare o rispetto a cui calibrare la camera;
- i sistema solidale alla camera di profondità.

Nella prossima sezione viene esplicata la logica con cui è stato realizzato l'algoritmo che stima la matrice di trasformazione fra il giunto cinque e la camera. Questa stima è fondamentale per passare le informazioni acquisite dalla camera al manipolatore



Figura 4.2: Immagine dove è mostrata la posizione della camera rispetto all'organo terminale del manipolatore articolato

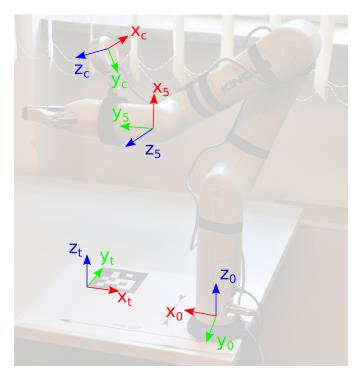


Figura 4.3: Immagine dove è mostrata l'intera configurazione del robot

4.2 Procedura di calibrazione della camera

In questa sezione viene illustrata la procedura di calibrazione della camera rispetto al braccio robotico. Lo scopo finale è quello di ottenere una matrice che sappia esprimere la posizione della camera rispetto al giunto 5 in modo tale da poter trasmettere l'informazione dalla camera al sistema di riferimento solidale alla base del robot.

La procedura di calibrazione è composta da una fase di acquisizione dei dati e da una fase di elaborazione dei dati. L'acquisizione dei dati consiste nel misurare la posa di un punto noto con la camera e contemporaneamente misurare la posizione del giunto 5 del braccio robotico rispetto al sistema di riferimento fisso sulla base del robot. I dati così salvati vengono poi processati al fine di trovare la matrice di calibrazione che riesca a minimizzare l'errore di misura fra quella nota e quella misurata in cui ci si frappone l'incertezza del posizionamento della camera rispetto al giunto 5.

4.2.1 Strumenti utilizzati per la calibrazione

Nella calibrazione fra camera e manipolatore articolato è stato impiegato un tabellone opportunamente costruito, mostrato in Figura~4.4. Il sistema di riferimento mostrato sul tabellone corrisponde al sistema di riferimento fisso del manipolatore articolato, ovvero quello solidale con la base. Attorno al sistema di riferimento fisso sono presenti quattro circoletti neri utili per centrare ed orientare i fori presenti sulla base del robot in modo da assicurarsi che il reale sistema di riferimento del manipolatore articolato sia ben allineato con quello raffigurato. Dal sistema di riferimento fisso viene misurato posizione ed orientamento dell'angolo in alto a destra dell'ArUco Marker. L'angolo di cui viene presa la misura presenta un secondo sistema di riferimento, con pedice t, riportante la posa del punto che viene misurato. Tuttavia essendo il tabellone bidimensionale l'unica dimensione che non può essere efficacemente riportata è quella riguardante la posizione reciproca dei due sistemi di riferimento lungo l'asse \mathbf{z} , essa dipende naturalmente solo dallo spessore del supporto della base, che pone il sistema di riferimento del Marker di 6 mm più in basso rispetto a quello del manipolatore articolato.

La (4.1) mostra la matrice di posa del sistema di riferimento del manipolatore articolato rispetto al sistema di riferimento posto sull'ArUco Marker. É una matrice di posa piuttosto semplice, è solo stato posto il segno meno sul primo e secondo elemento della diagonale poiché l'asse \mathbf{x} e l'asse \mathbf{y} di manipolatore e marker hanno direzioni rispettivamente opposte. Infine i primi tre elementi della quarta colonna rappresentano la distanza fra i due sistemi di riferimento secondo il sistema di riferimento del manipolatore articolato. Le misure che rappresentano le distanze sono espresse in metri al fine di produrre matrici omogenee a quelle della camera e dal manipolatore articolato. Si fa notare come nella nomenclatura adottata lo zero corrisponda al sistema di riferimento fisso del manipolatore articolato, mentre la t corrisponda al sistema di riferimento solidale con l'obiettivo, ovvero l'elemento di cui si vuole sapere la posa: il target.

$${}^{\mathbf{0}}\mathbf{T_{t}} = \begin{bmatrix} -1 & 0 & 0 & 0.29\\ 0 & -1 & 0 & -0.087\\ 0 & 0 & 1 & -0.06\\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$(4.1)$$

É interessante notare che il tabellone per la calibrazione può essere utilizzato anche invertendo la posizione del manipolatore articolato, il quale non si troverà più con l'asse \mathbf{x} e \mathbf{y} in direzione discorde rispetto a quelli dell'ArUco Marker, ma concordi. Anche in questa configurazione la calibrazione è agevole e fattibile, tuttavia bisogna modificare opportunamente la matrice di posa dell'ArUco Marker

rispetto alla base fissa del manipolatore articolato come mostrato dalla (4.2)

$${}^{\mathbf{0}}\mathbf{T_{t}} = \begin{bmatrix} 1 & 0 & 0 & -0.29 \\ 0 & 1 & 0 & 0.087 \\ 0 & 0 & 1 & -0.06 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
 (4.2)

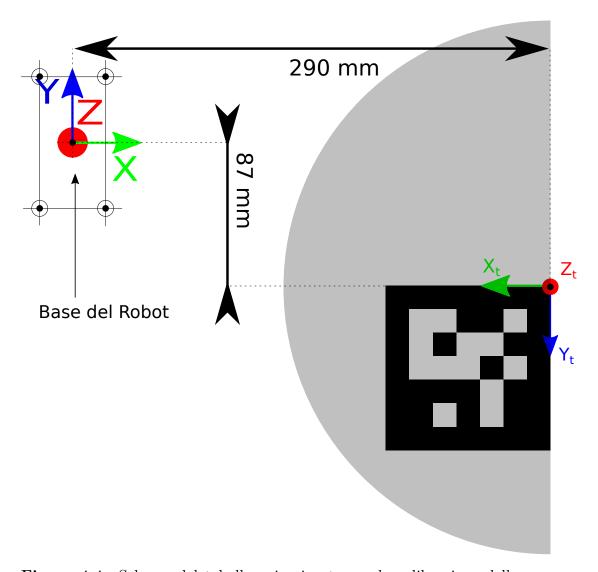


Figura 4.4: Schema del tabellone impiegato per la calibrazione della camera, viene esplicitato il sistema di riferimento del robot e del Marker usato come target. In grigio è rappresentata l'area coperta dalle misure di posa effettuate per la calibrazione

4.2.2 Processo di calibrazione

Affinché si possa svolgere correttamente e velocemente il processo di calibrazione è necessario avere un insieme di pose, ovvero un insieme di misure di posizione ed orientamento, acquisite. Inoltre nel processo di calibrazione messo in atto viene utilizzata anche una posa certa che viene usata come dato in ingresso: la posizione e l'orientamento di un ArUco Marker rispetto al sistema di riferimento solidale con la base del robot come illustrato in *Figura 4.4* e dalla (4.1).

L'insieme di pose acquisite deve essere un valido campione di dati acquisiti, quindi deve presentare una discreta varietà di misure della posa dell'ArUco Marker acquisite da diversi punti di vista al fine di caratterizzare bene quante più possibili configurazioni in cui il manipolatore articolato e la camera di profondità si possono trovare.

In Figura 4.4 è raffigurata l'area in cui viene posta la camera di profondità per acquisire le varie pose dell'ArUco Marker. Tutte le possibili posizioni in cui la camera viene posta per acquisire innumerevoli volte le pose del Marker ricadono dentro un un semi-cono, ovvero rispetto al sistema di riferimento posto sul Marker in Figura 4.4 il cono esiste solo per le ascisse positive. Il cono presenta un'apertura massima della falda di 30°, l'asse di simmetria del cono coincidente con l'asse z del sistema di riferimento posto sul Marker e il vertice del cono giace sul piano avente coordinata z nulla rispetto al sistema di riferimento solidale con la base del manipolatore articolato.

Lo spazio in cui la camera viene posta per acquisire le misure delle pose è una diretta conseguenza dell'algoritmo di calibrazione utilizzato per svolgere le acquisizioni. L'algoritmo di default ha con sé la posizione dell'asse di simmetria del cono ed ha anche due valori con cui equi-spaziare rispettivamente l'angolo di 30° della falda del cono e l'angolo di 180° del semi-cono. Grazie a questi valori è possibile: scegliere in quanti punti suddividere lo spazio di acquisizione in maniera tale da ottenere più o meno acquisizioni, scegliere se il sistema di riferimento ha gli assi \mathbf{x} e \mathbf{y} del sistema di riferimento del robot allineati o meno con quelli dell'ArUco Marker ed infine si può specificare una posizione dell'asse di simmetria del cono diversa rispetto a quella passante per l'asse \mathbf{z} dell'ArUco Marker.

Affinché sia possibile il riconoscimento della posa dell'ArUco Marker viene utilizzato OpenCV, un algoritmo di computer vision, tramite il quale si riesce a riconoscere la posizione e l'orientamento dei Marker. L'algoritmo processa l'immagine e si occupa di restituire, fra le varie informazioni, i 4 vertici del quadrato in cui è contenuto il Marker. Inoltre riconoscendo sia la posizione che l'orientamento del Marker le informazioni dei vertici vengono fornite sulla base di un orientamento fisso predefinito dentro l'algoritmo di riconoscimento immagine. Tramite l'utilizzo di questi quattro vertici è sempre possibile porre l'origine e gli assi del sistema di riferimento nella posizione voluta, ovvero secondo l'immagine in Figura 4.4. Questo

poiché i 4 vertici sono stati identificati in maniera univoca.

4.2.3 Algoritmo di ricerca della matrice di calibrazione

L'algoritmo di calibrazione si occupa nella sua prima fase di acquisire le matrici di posa delle varie posizioni dell'ArUco Marker rispetto alla camera e del giunto cinque rispetto al sistema di riferimento fisso solidale con la base del manipolatore articolato. Lo spazio di acquisizione rappresenta l'insieme di punti in cui l'algoritmo è in grado di posizionare la camera, più intervalli di acquisizione sono specificati in input più punti dello spazio di acquisizione sono utilizzati nel processo di acquisizione delle pose del Marker.

L'acquisizione delle matrici di posa sia della camera che del giunto fa sì che ogni volta che il manipolatore articolato si trovi in una posizione designata per l'acquisizione venga letto il feedback dei giunti del robot, ovvero la posizione dei giunti, così viene costruita la matrice di posa del giunto cinque rispetto al sistema di riferimento solidale con la base. Contemporaneamente al raggiungimento da parte del manipolatore articolato della posizione di acquisizione viene mandato un segnale alla camera affinché salvi la posa del sistema di riferimento posto sull'ArUco Marker. I due feedback, quello della posa del giunto cinque e quello della posa del Marker, vengono salvati consecutivamente per ogni posizione in un file di testo per ogni feedback (Robot_fb.txt e Camera_fb.txt) formando un unico file con tutte le matrici acquisite per il giunto o per il Marker.

Alla fine del processo di acquisizione si hanno a disposizione le seguenti matrici che esprimono tre pose differenti:

- diverse matrici di posa del giunto cinque rispetto al sistema di riferimento fisso solidale con il manipolatore articolato $({}^{0}T_{5})$;
- diverse matrici di posa dell'ArUco Marker rispetto al sistema di riferimento solidale con la camera di profondità (${}^{c}T_{t}$);
- la matrice che esprime la posizione dell'ArUco Marker rispetto al sistema di riferimento fisso solidare con il manipolatore articolato $({}^{0}\mathbf{T}_{t})$;

In Figura 4.5 sono mostrate le quattro matrici di trasformazione che vengono usate nel processo di calibrazione. L'unica matrice ignota delle tre è la matrice di calibrazione stessa, ovvero la ${}^{5}\mathbf{T_{c}}$, che si occupa di descrivere la posa della camera secondo il sistema di riferimento solidale con i giunto cinque.

Le quattro matrici del processo di calibrazione possono essere usate secondo la (4.3) per rappresentare due modi equivalenti di esprimere la posa dell'ArUco Marker rispetto al sistema di riferimento fisso solidale con il manipolatore articolato. Il primo modo è tramite la misura diretta la quale dipende da come è stato costruito il tabellone (Figura 4.4), esso rappresenta una misura nota ed è un dato

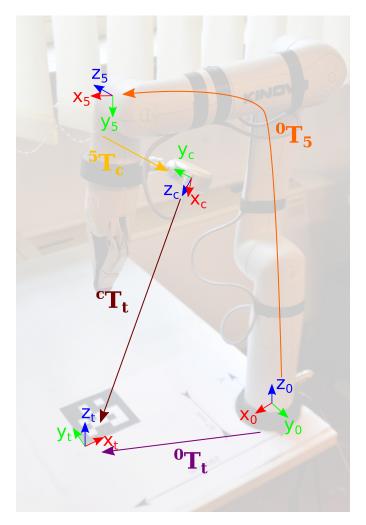


Figura 4.5: Rappresentazione grafica delle matrici presenti nel processo di ricerca della matrice di calibrazione: la matrice di posa del giunto 5, la matrice di calibrazione, la matrice di posa del Marker rispetto alla camera e rispetto al braccio

certo del problema. Il secondo modo è tramite la chiusura dell'anello che passa rispettivamente dal manipolatore articolato fino al giunto cinque, poi attraverso il collegamento fra manipolatore e camera ed infine alla misura effettuata con la camera dell'ArUco Marker.

$${}^{0}\mathbf{T_{t}} = {}^{0}\mathbf{T_{5}} {}^{5}\mathbf{T_{c}} {}^{c}\mathbf{T_{t}}$$
 (4.3)

L'equazione risolutiva si ottiene scambiando i membri e pre-moltiplicando la (4.3) per ${}^{0}\mathbf{T}_{5}^{-1}$ e post-moltiplicando per ${}^{c}\mathbf{T}_{t}^{-1}$, in questo modo a primo membro resta solo la matrice di calibrazione, che esprime la posizione della camera di profondità

rispetto al giunto cinque, mentre a secondo membro si trova una implementazione per calcolarla.

$${}^{5}\mathbf{T_{c}} = {}^{0}\mathbf{T_{5}^{-1}} {}^{0}\mathbf{T_{t}} {}^{c}\mathbf{T_{t}^{-1}}$$
 (4.4)

Al fine di evitare errori di cancellazione numerica durante l'inversione delle matrici sono state sfruttate le proprietà di simmetria e ortonormalità della matrice di rotazione, insieme alla conoscenza su come è strutturata una matrice di posa. Da ciò ne deriva che partendo da una matrice di posa generica ${}^{\mathbf{a}}\mathbf{T}_{\mathbf{b}}$ la sua inversa si può esprimere secondo la (4.5).

$${}^{\mathbf{a}}\mathbf{T}_{\mathbf{b}} = \begin{bmatrix} {}^{\mathbf{a}}\mathbf{T}_{\mathbf{b}} [1:3,1:3]' & -{}^{\mathbf{a}}\mathbf{T}_{\mathbf{b}} [1:3,1:3]' & {}^{\mathbf{a}}\mathbf{T}_{\mathbf{b}} [1:3,4] \\ 0 \ 0 \ 0 & 1 \end{bmatrix}$$
(4.5)

Successivamente al processo di acquisizione dei dati l'algoritmo di calibrazione si occupa della ricerca della matrice di calibrazione tramite una procedura di ricerca del minimo errore. Partendo dai risultati che sono stati precedentemente presentati si compone l'algoritmo di ricerca della funzione di ricerca del minimo errore per ottenere la matrice di calibrazione.

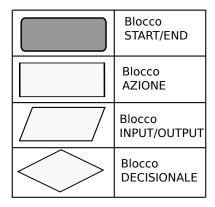


Figura 4.6: Legenda dei vari blocchi utilizzati nel diagramma di flusso

In Figura 4.7è mostrato lo schema di flusso che mostra la logica di funzionamento dell'algoritmo di ricerca della matrice di calibrazione, mentre in Figura 4.6 è mostrata la legenda dei vari blocchi del diagramma. Partendo dai dati noti e dai dati acquisiti, ovvero le matrici ${}^{0}\mathbf{T}_{t}$, ${}^{0}\mathbf{T}_{5}$ e ${}^{c}\mathbf{T}_{t}$, si crea una matrice di primo tentativo che approssima la matrice di posizione fra camera e giunto cinque, ovvero la ${}^{5}\mathbf{T}_{c}$ di primo tentativo. Successivamente sia dalla matrice di primo tentativo che dalla matrice ottenuta dai dati misurati si estraggono le coordinate del punto (${}^{5}p_{c_dati}$ e ${}^{5}p_{c_tent}$) e la matrice di Rotazione (${}^{5}R_{c_dati}$ e ${}^{5}R_{c_tent}$). L'estrazione dei due vettori e delle due matrici è finalizzata a creare un vettore di errore che raccoglie le differenze fra la matrice di calibrazione di tentativo e le matrici di

calibrazione ottenute dai dati misurati. Dunque il vettore di errore è somma tutti gli errori fra tutte le configurazioni misurate che danno la loro matrice di calibrazione e la matrice di calibrazione di tentativo. Per popolare il vettore di errore viene usata la (4.6): dove i primi tre elementi sono la differenza tra il vettore posizione calcolato tramite i dati misurati e il vettore posizione tentativo, mente gli ultimi tre elementi sono i valori assoluti della matrice emi-simmetrica che rappresentano le differenze fra gli angoli della configurazione misurata e della configurazione di tentativo. Lo scopo di moltiplicare tra loro le due matrici di rotazione, di cui la prima trasposta, e sottrarre la matrice identità è diretta conseguenza della suddetta moltiplicazione di matrici di rotazione di tentativo con la matrice di rotazione dei dati acquisiti: essa genera una diagonale popolata di uno e con una triangolare superiore ed inferiore popolata di numeri che rappresentano la rotazione fra le due matrici, come mostrato dall'equazione (4.7), quindi nel calcolo dell'errore fra le due configurazioni si pone l'accento nel trovare la configurazione che minimizza i vari δ i quali rappresentano la differenza di rotazione fra le due matrici.

$$error = \sum_{c_dati} \left\| vex \left({}^{5}\mathbf{R'_{c_tent}} {}^{5}\mathbf{R_{c_dati}} - \mathbf{Eye(3)} \right) \right\| \le toll$$
 (4.6)

$${}^{\mathbf{5}}\mathbf{R}_{\mathbf{c_tent}}' {}^{\mathbf{5}}\mathbf{R}_{\mathbf{c_dati}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -\delta_z & \delta_y \\ \delta_z & 0 & -\delta_x \\ -\delta_y & \delta_x & 0 \end{bmatrix}$$
(4.7)

I diversi set di dati misurati produrranno diverse matrici di calibrazione quindi l'obiettivo del processo di calibrazione è trovare la matrice di calibrazione che minimizza gli errori posti in essere dalle discrepanze fra le varie matrici di calibrazione calcolate dai dati acquisiti. Per questo motivo il processo di ricerca del minimo errore viene reiterato finché non si trova una matrice tentativo la cui somma degli errori fra i veri set di misura effettuati sta sotto una certa soglia di tolleranza, la quale quindo sarà la matrice di calibrazione cercata.

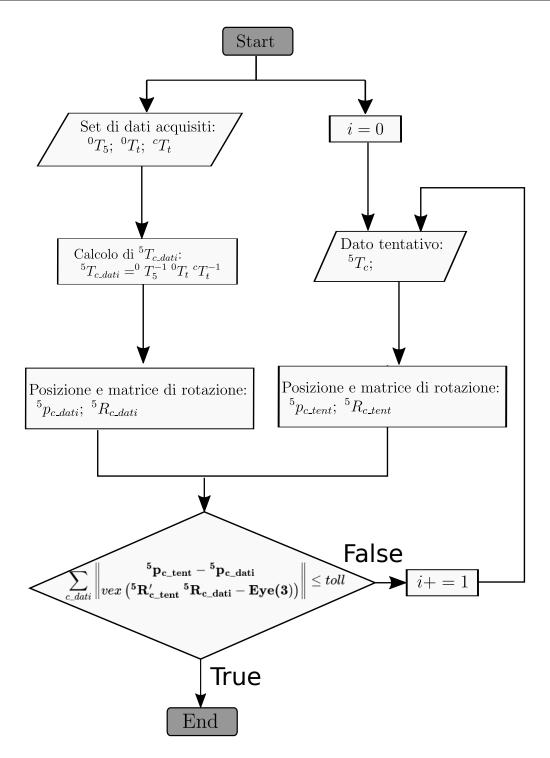


Figura 4.7: Schema di flusso logico che rappresenta il funzionamento dell'algoritmo di ricerca del minimo obiettivo al fine di trovare la matrice di calibrazione

4.3 Analisi critica della procedura implementata

La procedura di calibrazione implementata è una procedura semplice ed efficace poiché si basa sulla chiusura di un anello di misure e la ricerca del minimo errore dalle differenze fra le misure di questo anello. La distinzione fra questo sistema di calibrazione e quelli più diffusi attualmente in circolazione è la conoscenza della posa del target che si sta misurando grazie all'utilizzo di un preciso sistema di posizionamento e riferimento della base del manipolatore articolato rispetto alla posa dell'ArUco Marker.

La conoscenza della posa del target che si sta misurando rispetto al sistema di riferimento del manipolatore articolato permette di mantenere una migliore accuratezza nella stima della matrice di calibrazione, ciò si traduce sia in una misura di calibrazione più precisa sia in una stima che necessita un minore numero di acquisizioni. Per come è strutturato il problema esso ammette una soluzione univoca poiché si ha una equazione in una incognita, tuttavia nella procedura implementata si sono registrate delle acquisizioni in più per essere certi di ottenere una matrice di calibrazione che tiene conto anche di eventuali errori di acquisizione in fatto di incertezza nella misura, range di tolleranza, ripetibilità et similia. Tramite l'utilizzo di questa metodologia non è necessario ricorrere alla risoluzione in maniera iterativa del classico algoritmo di ottimizzazione dell'equazione matriciale (4.8). L'equazione da risolvere iterativamente presenta i due sistemi A e B che sono rispettivamente il manipolatore articolato e la camera di profondità, mentre la X rappresenta la matrice di trasformazione tra i due incognita.

Tuttavia la conoscenza della posa del target da misurare per calibrare la camera rappresenta un limite per la calibrazione poiché si è vincolati a montare il tabellone della calibrazione e quindi ad effettuare la calibrazione solo in determinate configurazioni e bisogna avere a disposizione determinati spazi. Infatti gli algoritmi che presentano la risoluzione della (4.8) presentano il vantaggio di poter piazzare il Marker in una generica posa, sarà poi compito dell'algoritmo di calibrazione identificare questa posa e successivamente riferirla rispetto al manipolatore articolato.

Per ovviare al problema dello spazio imposto dalla procedura di calibrazione si è strutturato l'algoritmo di calibrazione in maniera tale che si possa invertire il sistema di riferimento del manipolatore articolato e produrre una calibrazione efficace sia che il Marker sia posizionato nel semipiano delle ascisse positive del manipolatore articolato, sia che sia posizionato nel semipiano negativo (Figura 4.4). Un'altra misura adottata per ovviare al problema dei vincoli spaziali è poter spostare il centro del sistema di riferimento del Maker attorno al quale la camera si costruisce il semi-cono per le acquisizioni delle misure, questo consente di fare degli aggiustaggi nel caso in cui lo spazio a disposizione non permette una calibrazione nel punto passato di default all'algoritmo.

Complessivamente si può affermare che questo algoritmo cerca di unire, cercando

un compromesso, il meglio delle due configurazioni: la certezza della posa dell'ArUco Marker, che garantisce migliori performance di calibrazione, con la libertà di poter posizionare il Marker in posti diversi, tutti comunque giacenti a superfici normali all'asse ${\bf z}$ del robot.

$$\mathbf{AX} = \mathbf{XB} \tag{4.8}$$

Capitolo 5

Test sperimentali dimostrativi del sistema studiato

In questo capitolo sono descritti gli esperimenti dimostrativi compiuti atti a validare l'efficacia e la buona riuscita della calibrazione fra camera di profondità e manipolatore articolato. Viene specificato lo scopo per cui vengono condotti questi esperimenti e ne viene inoltre commentato l'esito, per comprendere se è in linea con quanto si voleva ottenere o meno.

5.1 Obiettivo delle prove sperimentali

Le prove sperimentali sono condotte al fine di validare se il sistema composto dal manipolatore articolato e dalla camera di profondità in coppia con l'infrastruttura informatica riesce ad esprimere risultati notevoli e riesce a fare apprezzare una efficace ed efficiente interazione fra la camera di profondità e il manipolatore articolato. Lo scopo finale della tesi infatti è quello di integrare il manipolatore articolato con un sistema di visione e di misura delle pose degli oggetti che lo circondano, l'integrazione è finalizzata a muovere il manipolatore articolato sulla base degli input prodotti dai dati derivanti dalla camera di profondità. Nella fattispecie è indispensabile che tutte i vari compiti in cui viene suddivisa l'azione complessiva siano coerenti e coordinati tra di loro al fine di garantire al sistema di compiere un'azione completa e di senso compiuto. Un esempio di alcuni compiti in cui viene scomposta l'azione sono:

• riconoscimento immagine;

- misura della distanza;
- acquisizione della posa del corpo misurato;
- trasformazione della posa dal sistema di riferimento della camera a quello del manipolatore articolato;
- posizionare l'organo terminale del manipolatore articolato in una posa prestabilita.

In Figura 5.1 è mostrata la schematizzazione dei vari blocchi che compongono il sistema con le rispettive funzioni svolte. Dallo schema si nota quanti e quali funzioni sono da coordinare per esprimere un'azione che porti ad un movimento del manipolatore generato da un input visivo. Di seguito è spiegato come si svolge l'interazione fra le varie funzioni, il corretto funzionamento delle singole funzioni e il risultato complessivo nel loro insieme rappresentano l'obiettivo della validazione delle prove sperimentali.

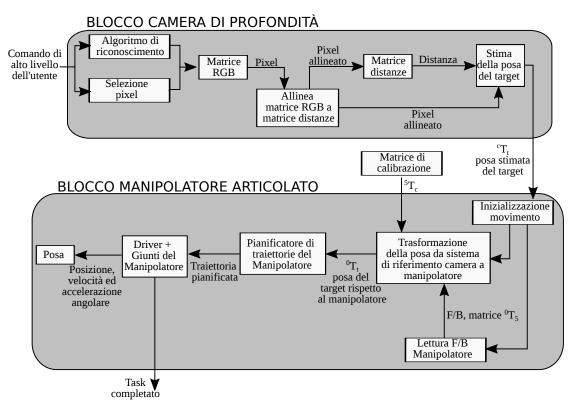


Figura 5.1: Schematizzazione del funzionamento del sistema oggetto di studi: nello schema viene esplicata come avviene l'interazione fra la camera di profondità e il manipolatore articolato

Tutte le prove sperimentali condotte sono casi particolari e applicativi dell'algoritmo schematizzato in *Figura 5.1*. L'algoritmo generico prende in input un comando ad alto livello dall'utente che entra nel blocco di competenza della camera di profondità al quale può essere attribuita una duplice interpretazione:

- Solo segnale di inizio procedura se nell'algoritmo è presente la libreria di computer vision che si occupa di riconoscere l'oggetto da manipolare e di cui acquisire la posa;
- Sia segnale di inizio procedura che pixel nel quale acquisire la posa dell'oggetto da manipolare.

Entrambi i comandi lavorano sulla matrice RGB, la matrice che permette sia all'utente che all'algoritmo di computer vision di visualizzare e riconoscere gli oggetti presenti nell'inquadratura. Ad ogni nuova acquisizione dei dati di output della camera la matrice RGB viene allineata con la matrice delle distanze affinché si possa usare l'informazione del pixel allineato per acquisire il valore della distanza corrispondente al pixel identificato nella matrice RGB, nel caso applicativo in esame il pixel allineato rappresenta la posizione in cui mandare l'organo terminale del manipolatore articolato. Per ottenere la posa a partire dal pixel allineato si costruisce il sistema di riferimento solidale con l'oggetto da manipolare: scegliendo due punti che rappresentano uno la direzione dell'asse \mathbf{x} rispetto al centro ed uno la direzione dell'asse y da cui poi per prodotto vettoriale si ricava anche la direzione dell'asse z. É importante l'orientazione scelta per il sistema di riferimento dell'oggetto da manipolare poiché l'organo terminale del manipolatore articolato afferra l'oggetto in base all'orientazione di quest'ultimo sistema riferimento in quanto non sono previste matrici di rotazione fra i due sistemi di riferimento, quello dell'oggetto e quello dell'organo terminale. Per questo motivo di solito si cerca di rivolgere il sistema di riferimento dell'oggetto in una configurazione utile per la manipolazione rispetto al sistema di riferimento dell'organo terminale.

Una volta acquisita la posa dell'oggetto da manipolare rispetto alla camera di profondità essa viene elaborata dal blocco di funzioni di competenza del manipolatore articolato. Il primo passo per movimentare il manipolatore articolato verso l'oggetto da manipolare è quello di trasformare la posa acquisita dal sistema di riferimento della camera a quello del manipolatore. A tal scopo si utilizza la matrice di trasformazione precedentemente calcolata e la matrice di posa del giunto cinque rispetto al sistema di riferimento fisso del manipolatore per effettuare la conversione della posa registrata dal sistema di riferimento della camera a quello del manipolatore. Avendo a disposizione la posa dell'oggetto da manipolare secondo il sistema di riferimento fisso solidale con il manipolatore si esegue l'algoritmo pianificatore di traiettorie che si occupa di pianificare una traiettoria in spazio operativo dalla posizione attuale del manipolatore alla posizione imposta dalla

posa. La traiettoria pianificata viene passata ai driver del manipolatore, i quali si occupano di mandare i riferimenti corretti ai giunti al fine di percorrere la traiettoria fino alla posa voluta con le corrette posizioni, velocità ed accelerazioni angolari di giunto. Una volta che la posa è stata raggiunta i driver del robot si occupano anche di notificare la fine del processo di movimentazione dei giunti.

In tutti gli esempi a seguire lo schema di funzionamento generale resta sempre fedele a questo, viene solo declinato alle varie applicazioni, l'obiettivo dunque è quello di dimostrare l'efficacia di funzionamento logico e fisico di questo algoritmo al fine di poterlo validare.

5.2 Analisi delle prove svolte

Una volta approntato l'algoritmo di calibrazione e di movimentazione del braccio robotico sono state condotte alcune prove con lo scopo di validare se le scelte progettuali adottate e i risultati ottenuti sono congeniali alla visione dell'oggetto, acquisizione della posa, trasformazione della posa acquisita secondo il sistema di riferimento del manipolatore ed infine pianificazione del moto.

5.2.1 Prima prova: afferrare un oggetto con input di alto livello definito dall'utente

Questa prima prova è volta a validare soprattutto la robustezza dell'algoritmo di trasferimento informazioni dalla camera al manipolatore articolato, ovvero la trasformazione dalla matrice di posa dell'oggetto da afferrare da riferito alla camera a riferito al manipolatore. Inoltre vengono anche testate la capacità di pianificazione del moto e collision avoidance del pianificatore.

La configurazione della prova compiuta è mostrata in Figura 5.2. Si osserva la posa iniziale del manipolatore articolato rispetto all'oggetto che deve manipolare, ovvero la bottiglia, e rispetto all'oggetto che deve evitare, ovvero il tavolo. Durante la prova l'utente ha il compito di selezionare il pixel in cui l'organo terminale del manipolatore va ad afferrare la bottiglia. Una volta preso questo input l'algoritmo segue lo schema precedentemente discusso mostrato in Figura 5.1.

Rispetto allo schema generale dell'algoritmo la prova eseguita dipende dall'input dell'utente anche per l'apertura o chiusura del gripper, inoltre accetta come input anche un comando per ritornare nella posizione di riposo. Queste aggiunte sono state fatte per rendere la prova ripetibile più e più volte senza soluzione di continuità. Poiché si possono manipolare oggetti sempre diversi, ritornare nella posizione di riposo e posarli aprendo l'organo terminale.

Complessivamente la prova valida l'algoritmo in uso e l'interazione fra manipolatore e camera. Anche dopo diverse manipolazioni la presa avviene sempre

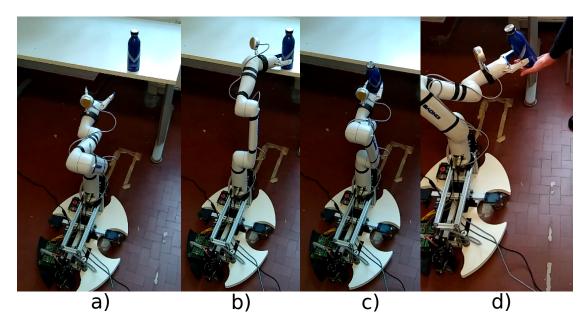


Figura 5.2: Sequenza di immagini che mostra la manipolazione della bottiglia. la sequenza da a) a c) è rappresentativa della procedura con input dall'utente, mentre la sequenza da a) a d) è rappresentativa della procedura automatica. L'immagine a) mostra la posizione reciproca fra organo terminale del manipolatore, bottiglia da manipolare e tavolo che funge da ostacolo

efficacemente nel punto indicato. L'algoritmo riesce efficacemente anche a evitare l'ostacolo una volta che gli si vengono specificati gli ingombri e la posizione. L'unica pecca riscontrata nella prova è inerente alla visuale della camera che viene parzialmente ostruita dall'ingombro dell'oggetto manipolato.

5.2.2 Seconda prova: afferrare un oggetto con procedura automatica

In questa seconda prova per il riconoscimento viene usata una libreria di computer vision, precisamente viene sfruttata la capacità di riconoscere bottiglie e persone. Dunque si è testata l'efficacia e l'accuratezza con cui vengono riconosciuti gli oggetti di interesse, inoltre viene anche testata l'interazione fra la posa acquisita e la facilità con cui essa rende manipolabile l'oggetto. Infine, come nella precedente prova, si va anche a testare l'efficacia della pianificazione rispetto all'ostacolo da evitare.

Anche per questa prova è stata mostrata la configurazione mostrata in *Figura* 5.2 dove il manipolatore è messo nelle condizioni di afferrare la bottiglia, ma deve necessariamente evitare la collisione col tavolo. Stavolta l'utente ha solo il compito di dare l'input di avvio poiché ci si affida alla libreria di riconoscimento immagini.

Rispetto all'algoritmo generale, mostrato in *Figura 5.1*, viene talvolta chiesto un input dell'utente come conferma di azione avvenuta con successo.

Il comportamento dell'algoritmo è tale per cui riesce a riconoscere efficacemente l'oggetto da manipolare e la posa mandata al manipolatore è sempre compatibile con la manipolazione, le volte in cui non è compatibile è a causa dei limiti fisici del manipolatore stesso. Inoltre nonostante la visuale parzialmente ostruita dalla bottiglia la libreria di computer vision riesce sempre a riconoscere una persona ogni qual volta si presentano nell'inquadratura dei dettagli caratterizzanti inequivocabilmente una persona. Si può concludere validando la logica dell'algoritmo utilizzato, l'unica anomalia riscontrata rispetto al codice implementato riguarda la misura della posa della persona, infatti il manipolatore non riesce a portare la bottiglia alla persona una volta che viene riconosciuta. Le cause di questa anomalia ricadono tipicamente fra queste tre:

- la persona riconosciuta è fuori dall'area di lavoro del robot;
- la bottiglia intralcia la visuale del sensore di profondità che non riesce ad acquisire bene la misura;
- la superficie offerta dalla persona non è adatta ad ottenere una misura esatta o corretta.

Una possibile soluzione potrebbe essere rappresentata dal fare acquisire al manipolatore la posizione della persona prima che afferri l'oggetto e successivamente mandare direttamente l'organo terminale con la bottiglia verso la persona. Questa soluzione implica che la persona non cambi radicalmente posizione durante tutto il processo di manipolazione.

5.3 Presentazione delle prove proposte

Visti i risultati raggiunti con l'interazione fra il manipolatore e la camera di profondità si è ipotizzato di svolgere altre prove di complessità simile o inferiore che possono essere portate a compimento con successo dalla medesima struttura logica dell'algoritmo e del sistema.

5.3.1 Effettuare un pick-and-place con un oggetto riconosciuto dall'utente o automaticamente

Si tratta di modificare le precedenti prove di manipolazione con e senza l'ausilio della libreria di riconoscimento immagini affinché il manipolatore posizioni l'oggetto afferrato sopra una porzione di spazio definito a priori e lo depositi lì aprendo le pinze. Si tratta di una aggiunta poco rilevante in termini di complessità del codice

ma che permette una facile e veloce raccolta in un unico posto degli oggetti che il manipolatore afferra.

Questa modifica risulta decisiva per tutte quelle operazioni in cui è necessario effettuare una raccolta automatica di determinati oggetti. Viste le precedenti performance nel riconoscere, afferrare e manipolare oggetti questa è sicuramente un'applicazione che mette a frutto il sistema fisico e la struttura informatica del robot.

5.3.2 Utilizzare una pulsantiera con input di alto livello definito dall'utente o in maniera automatica

La procedura con input definita dall'utente non necessita della scrittura di un nuovo algoritmo, necessita solamente dell'utilizzo in maniera differente del medesimo algoritmo utilizzato per afferrare un oggetto con input definito dall'utente. Sfruttando le logiche già presenti nell'algoritmo si può facilmente aprire e chiudere l'organo terminale, muovere l'organo terminale verso una posa ben definita e ritrarlo nella configurazione iniziale. L'utente dunque è in grado di chiudere il gripper e selezionare il pulsante a cui accostarlo tramite input sull'immagine a video. Il sistema si farà carico di prendere l'input di alto livello e portare il manipolatore con l'organo terminale chiuso a contatto con il pulsante indicato al fine di azionarlo.

Nel caso invece si voglia fare la medesima operazione ma in modo automatico bisogna utilizzare una libreria di riconoscimento immagine tale da riconoscere il pulsante desiderato ed acquisirne la posa. Una volta acquisitane la posa la procedura di pianificazione del moto è analoga a quella vista per afferrare un oggetto solo che bisogna utilizzare l'organo terminale con le pinze chiuse al fine di interagire efficacemente col pulsante. La parte del precedente algoritmo che si occupava di passare ad una persona l'oggetto afferrato non viene utilizzato in questa applicazione e viene quindi rimosso.

Un importante accorgimento che va fatto ad entrambi gli algoritmi che si propongono di utilizzare l'organo terminale del manipolatore per azionare un pulsante è quello di sottrarre lo spazio presente fra il centro del sistema di riferimento dell'organo terminale e l'estremità in maniera tale da non farlo urtare. Si può anche pensare di lasciare un millimetro al fine di mandare il gripper in interferenza col pulsante per ottenere la pressione necessaria ad azionare il tasto, ma non di più altrimenti si può incorrere incorre nel cedimento dell'organo terminale.

Capitolo 6

Conclusione

Il progetto di tesi affrontato ha l'obiettivo di trovare una configurazione efficace per il sistema composto da un manipolatore articolato a 6 gradi di libertà ed una camera di profondità. Nell'interazione fra questi due attori principali bisogna strutturare un algoritmo tale per cui i dati acquisiti dalla camera siano opportunamente elaborati ed utilizzabili dal manipolatore articolato al fine di consentirne il moto verso il punto acquisito dalla camera.

É stato cercato un metodo efficace per la movimentazione dei giunti del manipolatore articolato che permettesse sia di muovere l'organo terminale del manipolatore in spazio operativo, sia che si occupasse di evitare gli ostacoli che vengono specificati all'algoritmo di pianificazione del moto. A tale fine sono stati vagliati diversi tipi di pianificatori già sviluppati e presenti sulla letteratura scientifica o sul mercato. Qualsiasi pianificatore per essere utilizzato presenta una richiesta di input in termini di posa da raggiungere ed eventualmente orientamento da mantenere sull'organo terminale durante la traiettoria, l'algoritmo di pianificazione si farà carico di leggere la posa attuale del robot e di stabilire il corretto movimento dei giunti per muoversi verso la posa indicata. Il pianificatore scelto è probabilistico e cerca di far convergere la pianificazione verso il percorso più economico in termini di traiettorie percorse dai giunti o di traiettoria percorsa dall'organo terminale o di traiettoria con il minor dispendio energetico.

Della camera di profondità sono stati sfruttati sia le informazioni ricavate dal sensore di profondità sia le informazioni ricavate dalla matrice RGB. Grazie ad un'istruzione di allineamento fra i pixel della matrice di profondità e quelli della matrice RGB è stato possibile risalire alla distanza che i vari oggetti avevano rispetto al sistema di riferimento solidale con la camera i profondità. Sono state sviluppate due funzioni utili alla trasmissione dei dati dalla camera al manipolatore: la prima ha permesso di scomporre la distanza nelle tre coordinate rispetto agli assi \mathbf{x} , \mathbf{y} e \mathbf{z} e la seconda ha permesso di ricavare la posa di un sistema di riferimento posto sull'oggetto di interesse rispetto al sistema di riferimento della camera.

Infine è stata ideata una procedura rapida e veloce per permettere la calibrazione fra la camera di profondità e il manipolatore articolato. L'algoritmo di calibrazione utilizzato ha il pregio di utilizzare una tecnica di risoluzione del problema non iterativa, ma certa, basata su una equazione matriciale con una sola incognita, che quindi permette in poche iterazioni di raggiungere una calibrazione della camera affidabile. La matrice di calibrazione rappresenta un tassello molto importante poiché permette di riferire efficacemente l'informazione della posa dell'oggetto inquadrato rispetto al sistema di riferimento solidale al giunto cinque del manipolatore articolato, dal quale poi è semplice riferire la stesa posa rispetto al sistema di riferimento fisso solidale con la base del manipolatore stesso.

Complessivamente le procedure ideate ed impiegate nel manipolatore articolato, la matrice di calibrazione per collegare i sistemi di riferimento dei due sistemi e l'algoritmo sviluppato per fare interagire i due sistemi efficacemente ha prodotto buoni risultati. Sono state condotte diverse prove sperimentali le quali hanno ottenuto tutti esiti positivi: si è sempre riusciti a manipolare gli oggetti desiderati evitando gli ostacoli che si frapponevano fra il manipolatore e l'oggetto stesso. Questo dimostra che la logica applicata alla creazione del sistema è sicuramente valida ed utile nella manipolazione robotica. Il grado di destrezza, efficienza ed autonomia raggiunto è paragonabile a quello attualmente presente nello stato dell'arte, quindi si può affermare che il sistema realizzato sia comparabile con l'attuale tecnologia presente in mercato e rappresenta una notevole punto di partenza per ulteriori sviluppi più sofisticati.

Come futura miglioria auspicabile da produrre vi è quella di un algoritmo per la pianificazione del moto che sia sviluppato in proprio, non preso dalla produzione scientifica mondiale, in modo tale che si possano scegliere efficacemente tutte le leggi del moto del giunto al fine di ottenere alcuni movimenti meno bruschi e sussultori. Un'altra implementazione utile consiste nel riuscire a leggere efficacemente la nuvola di punti prodotta dalla camera di profondità al fine di dare in pasto al pianificatore tutti gli ostacoli che sono presenti nell'inquadratura della camera in modo da evitare urti con oggetti. Infine dotare il manipolatore articolato di una piattaforma mobile ad esso integrato in maniera tale da riuscire a coordinare fra la piattaforma e il manipolatore il movimento di avvicinamento ad un corpo estenderebbe il campo di lavoro dell'applicazione studiata notevolmente, rendendolo ancora più versatile ed indipendente dall'uomo.

Bibliografia

- [1] Mohammed Owais Qureshi e Rumaiya Sajjad Syed. «The Impact of Robotics on Employment and Motivation of Employees in the Service Sector, with Special Reference to Health Care». In: Safety and Health at Work 5.4 (2014), pp. 198-202. ISSN: 2093-7911. DOI: https://doi.org/10.1016/j.shaw.2014.07.003. URL: https://www.sciencedirect.com/science/article/pii/S2093791114000511 (cit. a p. 1).
- [2] Spyros Fountas, Borja Espejo-García, Aikaterini Kasimati, Nikolaos Mylonas e Nicoleta Darra. «The Future of Digital Agriculture: Technologies and Opportunities». In: *IT Professional* 22.1 (2020), pp. 24–28. DOI: 10.1109/MITP.2019.2963412 (cit. a p. 2).
- [3] Astrid Weiss, Ann-Kathrin Wortmeier e Bettina Kubicek. «Cobots in Industry 4.0: A Roadmap for Future Practice Studies on Human–Robot Collaboration». In: *IEEE Transactions on Human–Machine Systems* 51.4 (2021), pp. 335–345. DOI: 10.1109/THMS.2021.3092684 (cit. a p. 2).
- [4] ONU. Sustainable Development Goals. 2015. URL: https://sdgs.un.org/goals (cit. alle pp. 2, 3).
- [5] Emily Minton. Working with Robots: A Guide to the Levels of Collaboration Between Humans and Robots. Ott. 2020. URL: https://www.zetagroupengineering.com/working-with-robots-a-guide-to-the-levels-of-collaboration-between-humans-and-robots/ (cit. a p. 7).
- [6] Kinova. Kinova Web Page. URL: https://www.kinovarobotics.com/en (cit. a p. 7).
- [7] KUKA. KUKA Web Page. URL: https://www.kuka.com/it-it (cit. a p. 7).
- [8] Staubli. Staubli Web Page. URL: https://www.staubli.com/it-it/ (cit. ap. 7).
- [9] Panasonic. Panasonic Web Page. URL: https://industrial.panasonic.com/ww/products/pt/industrial-robots (cit. a p. 7).
- [10] Kawasaki Kawasaki Web Page. URL: https://robotics.kawasaki.com/en1/index.html?language_id=2 (cit. a p. 7).

- [11] R. A. R. C. Gopura e Kazuo Kiguchi. «Mechanical designs of active upper-limb exoskeleton robots: State-of-the-art and design difficulties». In: 2009 IEEE International Conference on Rehabilitation Robotics. 2009, pp. 178–187. DOI: 10.1109/ICORR.2009.5209630 (cit. a p. 15).
- [12] Ying He, Bin Liang, Yu Zou, Jin He e Jun Yang. «Depth Errors Analysis and Correction for Time-of-Flight (ToF) Cameras». In: Sensors 17 (gen. 2017), p. 92. DOI: 10.3390/s17010092 (cit. alle pp. 19, 20).
- [13] Microsoft. Microsoft Kinect official website datasheet. URL: https://docs.microsoft.com/en-us/azure/Kinect-dk/hardware-specification (cit. a p. 20).
- [14] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren e Dieter Fox. «RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments». In: *The International Journal of Robotics Research* 31.5 (2012), pp. 647–663. DOI: 10.1177/0278364911434148. eprint: https://doi.org/10.1177/0278364911434148. URL: https://doi.org/10.1177/0278364911434148 (cit. a p. 21).
- [15] Jing Tong, Jin Zhou, Ligang Liu, Zhigeng Pan e Hao Yan. «Scanning 3D Full Human Bodies Using Kinects». In: *IEEE Transactions on Visualization and Computer Graphics* 18.4 (2012), pp. 643–650. DOI: 10.1109/TVCG.2012.56 (cit. a p. 21).
- [16] Dina Satybaldina, Gulziya Kalymova e Natalya Glazyrina. «Application Development for Hand Gestures Recognition with Using a Depth Camera». In: *Databases and Information Systems*. A cura di Tarmo Robal, Hele-Mai Haav, Jaan Penjam e Raimundas Matulevičius. Cham: Springer International Publishing, 2020, pp. 55–67. ISBN: 978-3-030-57672-1 (cit. a p. 21).
- [17] Anh Tuan Nghiem, Edouard Auvinet e Jean Meunier. «Head detection using Kinect camera and its application to fall detection». In: 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA). 2012, pp. 164–169. DOI: 10.1109/ISSPA.2012.6310538 (cit. a p. 21).
- [18] Kinova Kinova Gen3 Lite Technical Resources. URL: https://www.kinovarobotics.com/en/resources/gen3-lite-technical-resources (cit. a p. 24).
- [19] P. Corke. Robotics, vision and control. fundamental algorithms in MATLAB® second, completely revised. 2^a ed. Vol. 118. Springer International Publishing, 2017, pp. XIX, 693. ISBN: 978-3-319-54413-7 (cit. a p. 32).
- [20] Luigi Villani Bruno Siciliano Lorenzo Sciavicco. *Robotica. Modellistica, pianificazione e controllo.* 3ª ed. McGraw-Hill Education, 2008, p. 640. ISBN: 883866322X (cit. alle pp. 32, 35).

- [21] ROS. ROS official website. URL: https://www.ros.org/(cit. a p. 40).
- [22] MoveIt. MoveIt official website. URL: https://moveit.ros.org/ (cit. a p. 40).
- [23] Ioan Sucan, Mark Moll e E.E. Kavraki. «The Open Motion Planning Library». In: Robotics & Automation Magazine, IEEE 19 (dic. 2012), pp. 72–82. DOI: 10.1109/MRA.2012.2205651 (cit. a p. 46).
- [24] Pilz GmbH & Co. KG. PILZ Industrial Motion Planner Ros package. URL: https://www.pilz.com/en-INT/products/robotics/ros-modules (cit. a p. 46).
- [25] Intel. Inel L515 reference page. URL: https://www.intelrealsense.com/lidar-camera-1515/(cit. a p. 50).
- [26] Intel. IntelRealsense deproject pixel to point method. URL: https://github.com/IntelRealSense/librealsense/wiki/Projection-in-RealSense-SDK-2.0?fbclid=IwAR3gogVZe824YUps88Dzp02AN_XzEm1BDb0UbmzfoYvn1qDFb7KzbIz9twU#point-coordinates (cit. a p. 60).
- [27] OpenCV. OpenCV official website. URL: https://opencv.org/ (cit. a p. 61).
- [28] Ivan Culjak, David Abram, Tomislav Pribanic, Hrvoje Dzapo e Mario Cifrek. «A brief introduction to OpenCV». In: 2012 Proceedings of the 35th International Convention MIPRO. Mag. 2012, pp. 1725–1730 (cit. a p. 61).