

LAUREA SPECIALISTICA IN INGEGNERIA  
DELL'AUTOMAZIONE

TESI DI LAUREA

UTILIZZO IN TEMPO REALE DI  
INFORMAZIONI VISIVE STEREO PER IL  
CONTROLLO IN SICUREZZA DI UN  
MANIPOLATORE ROBOTICO IN PRESENZA  
DI OPERATORI UMANI

CANDIDATO  
FABRIZIO FLACCO

RELATORI

PROF. ANTONIO BICCHI

PROF. ANDREA CAITI

ING. RICCARDO SCHIAVI

4 DICEMBRE 2007

*Ai miei genitori che mi hanno sempre sostenuto.*

*A mio fratello ed i miei amici che mi hanno  
incoraggiato nei momenti difficili.*

*Alla mia ragazza che mi è stata  
sempre vicino al di là  
delle distanze.*

---

## Sommario

In questa Tesi presentiamo un sistema per ottenere la *piena coesistenza* fra un manipolatore robotico ed un operatore umano che condividono lo stesso spazio di lavoro. Con piena coesistenza intendiamo la possibilità per l'operatore ed il manipolatore di condividere lo stesso spazio di lavoro nello stesso momento senza dover limitare (se non minimamente) il proprio *task*; né in termini di spazio né di prestazioni; considerando sempre prioritaria la sicurezza dell'operatore. Manipolando le informazioni date da una coppia di camere in configurazione stereo costruiamo una rappresentazione degli ostacoli seguendo la teoria del  $\mathcal{C}$ -Space. Quindi, utilizzando un sistema di forze repulsive da noi ideato che lavora interamente sul  $\mathcal{C}$ -Space, riusciamo a far proseguire il task assegnato al manipolatore evitando gli ostacoli (operatore umano).

## Abstract

In this thesis we present a system for obtaining *full coexistence* between robotic manipulators and human operators who share the same workspace. Whith full coexistence we mean the opportunity for the operator and the manipulator to share the same workspace at the same time without having to restrict (if not least) his task; nor in terms of space or performance; operator's safe is ever priority. Using the information of a stereo vision system the algorithm build a representation of obstacles following the  $\mathcal{C}$ -Space's theory. Then, using a system of repulsive forces we created, working entirely on  $\mathcal{C}$ -Space, we continue the manipulator's assigned task avoiding obstacles (incuding human operators).

# Indice

<b>1</b>	<b>Manipolatore Robotico</b>	<b>11</b>
1.1	Struttura . . . . .	11
1.1.1	Aumento della destrezza . . . . .	12
1.2	Cinematica . . . . .	13
1.2.1	Parametri di Denavit-Hartenberg . . . . .	14
1.2.2	Cinematica diretta . . . . .	14
1.2.3	Workspace . . . . .	17
1.3	Attuatori . . . . .	18
1.3.1	Muscoli artificiali . . . . .	19
1.3.2	Controllo attuatori . . . . .	19
1.4	Sensori . . . . .	21
1.4.1	Calibrazione sensori . . . . .	22
1.5	Controllo . . . . .	22
1.5.1	Controllo alla <i>Arimoto</i> . . . . .	23
1.5.2	Realizzazione discreta . . . . .	24
1.5.3	Riferimento Smorzato . . . . .	25
1.5.4	Simulazione . . . . .	25
<b>2</b>	<b>Visione Stereo</b>	<b>29</b>
2.1	Videocamera: funzionamento e modello matematico . . . . .	29
2.1.1	Parametri Intrinseci . . . . .	30
2.1.2	Parametri Estrinseci . . . . .	32
2.1.3	Matrice di Proiezione . . . . .	33
2.2	Geometria Visione Stereo . . . . .	33
2.2.1	Geometria Epipolare . . . . .	33
2.2.2	La Matrice Fondamentale . . . . .	35
2.2.3	Condizione di Corrispondenza . . . . .	36
2.3	Autocalibrazione . . . . .	36
2.3.1	Matrici di calibrazione in forma canonica . . . . .	37
2.3.2	Forma canonica con riduzione della distorsione . . . . .	39
2.3.3	Spazio di Proiezione, Metrico ed Euclideo . . . . .	39

2.3.4	Rettificazione Metrica . . . . .	40
2.4	Triangolazione . . . . .	43
<b>3</b>	<b>C-Space</b>	<b>45</b>
3.1	Manifolds . . . . .	45
3.2	$\mathcal{C}$ -Space, $\mathcal{C}$ -Free e $\mathcal{C}$ -Obstacle . . . . .	46
3.3	Costruzione $\mathcal{C}$ -Obstacle per un manipolatore multi-link . . . . .	47
<b>4</b>	<b>Server di Visione</b>	<b>49</b>
4.1	Equalizzazione dell'immagine . . . . .	49
4.2	Background Subtraction . . . . .	52
4.3	Area oggetti e filtraggio . . . . .	53
4.4	Ricostruzione dei punti 3D . . . . .	55
4.4.1	Estrazione delle <i>feature</i> . . . . .	55
4.4.2	Triangolazione <i>feature</i> . . . . .	56
4.5	Creazione del $\mathcal{C}$ -Space . . . . .	57
4.5.1	Metodo A . . . . .	57
4.5.1.1	Contorno e Zone d'Ombra . . . . .	57
4.5.1.2	Costruzione Modello 3D degli Oggetti . . . . .	58
4.5.1.3	Costruzione Modello 3D del Manipolatore . . . . .	60
4.5.1.4	Calcolo del $\mathcal{C}$ -Space . . . . .	60
4.5.1.5	Considerazioni sul Metodo A . . . . .	63
4.5.2	Metodo B . . . . .	63
4.5.2.1	Mappa delle Profondità . . . . .	64
4.5.2.2	Calcolo del $\mathcal{C}$ -Space . . . . .	65
4.5.2.3	Considerazioni e Confronto con il Metodo A . . . . .	66
<b>5</b>	<b>Obstacle Avoidance : <math>\mathcal{C}</math>-Force</b>	<b>68</b>
5.1	Forza Repulsiva . . . . .	69
5.2	Movimento dell'Ostacolo . . . . .	71
5.3	Movimento del Manipolatore . . . . .	74
5.4	Manovra Evasiva senza il calcolo dei Baricentri . . . . .	75
5.5	$\mathcal{C}$ -Force: Simulazioni . . . . .	76
5.5.1	Simulazione 1 . . . . .	77
5.5.2	Simulazione 2 . . . . .	77
5.5.3	Simulazione 3 . . . . .	78
5.5.4	Simulazione 4 . . . . .	78
<b>6</b>	<b>Risultati</b>	<b>82</b>
6.1	Test 1 . . . . .	83
6.2	Test 2 . . . . .	85

6.3	Commenti ai Risultati . . . . .	85
<b>7</b>	<b>Conclusioni e Sviluppi Futuri</b>	<b>87</b>
<b>A</b>	<b>Comunicazione :</b>	
	<b>Server Visione <math>\leftrightarrow</math> Controllo</b>	<b>88</b>
A.1	Comunicazione TCP . . . . .	88
<b>B</b>	<b>Complementi algebrici</b>	<b>91</b>
B.1	Geometria Omogenea . . . . .	91
B.1.1	Coordinate omogenee . . . . .	91
B.1.2	Retta omogenea su un piano . . . . .	92
B.1.2.1	Retta fra due punti . . . . .	92
B.1.2.2	Intersezione fra due rette . . . . .	92
B.1.2.3	Distanza minima fra una retta ed un punto . . . . .	92
B.1.3	Piano omogeneo nello spazio . . . . .	92
B.1.3.1	Piano passante per tre punti . . . . .	92
B.1.3.2	Piano costruito minimizzando la distanza fra più punti . . . . .	93
B.1.3.3	Intersezione fra un piano ed il raggio di proiezione di un punto sul piano immagine . . . . .	93
B.2	Soluzione di un sistema di equazioni lineari . . . . .	93
B.3	Matrici skew simmetrica . . . . .	94
B.4	Norma di Frobenius . . . . .	94
<b>C</b>	<b>Manipolatore <i>SoftArm</i></b>	<b>96</b>
C.1	Cinematica diretta . . . . .	96
C.2	Cinematica differenziale . . . . .	97
C.2.1	Jacobiano di posizione . . . . .	97
C.2.2	Jacobiano di orientazione . . . . .	97
C.2.3	Inversione Jacobiano di posizione . . . . .	97
C.3	Singolarità cinematiche . . . . .	97
C.4	Dinamica . . . . .	98
C.4.1	Matrice $B$ . . . . .	98
C.4.2	Matrice $C$ . . . . .	99
C.4.3	Vettore $G$ . . . . .	102

# Elenco delle tabelle

1.1	Masse e lunghezze dei link . . . . .	12
1.2	Dimensione delle pulegge ai giunti . . . . .	12
1.3	Parametri di DH nell'ordine di applicazione delle trasformazioni rigide . . . . .	14
A.1	Messaggi inviati dal server di rete al client . . . . .	90

# Elenco delle figure

1.1	Schema del manipolatore . . . . .	11
1.2	Offset di spalla: foto e schema. . . . .	12
1.3	Gomito: foto e schema. . . . .	13
1.4	Proiezioni di una configurazione del <i>SoftArm</i> . . . . .	15
1.5	Rendering del workspace ottenuti con $q_3$ variabile in 5, 9, 33 e 129 passi. . . . .	17
1.6	Da sinistra a destra, visione del workspace dall'alto e dal basso. . . . .	18
1.7	Proiezioni del workspace sul piano $x-y$ dall'alto (a sinistra) e dal basso (a destra). . . . .	18
1.8	Alloggiamento dei muscoli e regolatori elettropneumatici. . . . .	19
1.9	Configurazione agonista-antagonista. . . . .	20
1.10	Collocazione potenziometro. . . . .	21
1.11	Relazione angolo tensione dei 3 potenziometri. . . . .	22
1.12	Schema controllo alla <i>Arimoto</i> . . . . .	24
1.13	Andamento della funzione 1.18. . . . .	25
1.14	Schema Simulink <sup>®</sup> : dinamica del manipolatore. . . . .	26
1.15	Schema Simulink <sup>®</sup> del controllore. . . . .	27
1.16	Risposta a gradino. . . . .	27
1.17	Simulazione: configurazione variabile. . . . .	28
2.1	Geometria del modello Pinhole. . . . .	30
2.2	Coordinate nel piano immagine. . . . .	31
2.3	Rototraslazione data dai parametri Estrinseci. . . . .	33
2.4	Stereo Visione. . . . .	34
2.5	Piano Epipolare. . . . .	34
2.6	Geometria Epipolare. . . . .	35
2.7	Spazi di ricostruzione e ambiguita. . . . .	40
2.8	Absolute Dual Quadric. . . . .	41
2.9	Triangolazione. . . . .	44
3.1	Esempio di <i>Manifold</i> . . . . .	45
3.2	Tipi di contatto. . . . .	48



4.1	Schema di esecuzione del server di visione . . . . .	50
4.2	Equalizzazione dell'istogramma . . . . .	51
4.3	Regolazione luminosità e contrasto . . . . .	51
4.4	Sottrazione del <i>background</i> . . . . .	52
4.5	Contorno dell'oggetto . . . . .	54
4.6	Matching delle <i>feature</i> . . . . .	56
4.7	Triangolazione. . . . .	56
4.8	Contorno dell'oggetto, convex hull . . . . .	58
4.9	Posizione 3D dei punti del contorno. . . . .	59
4.10	Punti ricostruiti nello spazio. . . . .	59
4.11	Modello 3D dell'oggetto . . . . .	60
4.12	Modello 3D del <i>SoftArm</i> . . . . .	61
4.13	Metodo della sfera . . . . .	61
4.14	Metodo delle disequazioni del piano . . . . .	62
4.15	<i>C-Space</i> : risoluzione 3 gradi. . . . .	63
4.16	Mappa delle profondità . . . . .	65
4.17	Metodo B: confronto profondità. . . . .	66
4.18	<i>C-Space</i> ottenuto con il metodo B. . . . .	66
4.19	Confronto fra il metodo A ed il metodo B . . . . .	67
5.1	Modello a molla compressa . . . . .	69
5.2	Forza respingente . . . . .	69
5.3	Forza respingente: <i>C-Space</i> discreto . . . . .	70
5.4	Forza respingente: <i>C-Space</i> 3 dimensioni . . . . .	71
5.5	Variazione baricentri <i>C-Obstacle</i> . . . . .	72
5.6	Variazione baricentri <i>C-Obstacle</i> . . . . .	73
5.7	Movimento Manipolatore. . . . .	74
5.8	Simulazione 1 . . . . .	77
5.9	Simulazione 2: <i>C-Force</i> eq.5.3 . . . . .	78
5.10	Simulazione 2: <i>MDC-Force</i> . . . . .	79
5.11	Simulazione 2: <i>PDC-Force</i> . . . . .	79
5.12	Simulazione 3: <i>MDC-Force</i> . . . . .	80
5.13	Simulazione 3: <i>PDC-Force</i> . . . . .	80
5.14	Simulazione 4: <i>PDC-Force</i> . . . . .	81
5.15	Simulazione 4: <i>MDC-Force</i> . . . . .	81
6.1	Test 1. . . . .	82
6.2	Test 1. . . . .	84
6.3	Test 2. . . . .	85
A.1	Winsock . . . . .	89

## Introduzione

La coesistenza di operatori umani e manipolatori robotici nello stesso spazio di lavoro ha avuto, negli ultimi anni, una crescente importanza, soprattutto in campo industriale. Conferma di tale tendenza è l'enorme incremento della ricerca in questo campo.

La priorità nella coesistenza tra manipolatori ed operatori è la sicurezza di quest'ultimo. In campo industriale vengono utilizzati manipolatori robotici capaci di esercitare grandi forze e di lavorare con enormi carichi a velocità elevate. Tali caratteristiche rendono impraticabile la coesistenza nello stesso spazio di lavoro di operatori e manipolatori, a causa della limitata sicurezza. La sicurezza dell'operatore solitamente viene garantita utilizzando dei sensori di prossimità (es. fasci laser) per verificare la presenza di un operatore nello spazio di lavoro; al verificarsi di tale condizione il manipolatore viene quindi fermato finché lo spazio di lavoro non viene ad essere nuovamente libero. Consideriamo ad esempio una catena di montaggio. La presenza di un operatore umano per controllo qualità, manutenzione o semplicemente pulizia fermerebbe il manipolatore portando al blocco parziale o totale della catena.

In questa Tesi abbiamo sviluppato un sistema che, utilizzando le informazioni provenienti dalla stereovisione, riesce ad ottenere la completa coesistenza fra operatore umano e manipolatore robotico. La procedura sviluppata utilizza le informazioni ricevute dalla visione stereo per costruire una rappresentazione degli ostacoli seguendo la teoria dei  $\mathcal{C}$ -Space, quindi utilizzando un sistema di forze repulsive da noi ideato ( $\mathcal{C}$ -Force) che lavora interamente sul  $\mathcal{C}$ -Space, se pur in presenza di ostacoli il manipolatore continua il task assegnato modificando la traiettoria in base al campo di potenziale artificiale.

Nel primo capitolo viene presentato il manipolatore che è stato utilizzato per le prove pratiche, soffermandoci in particolar modo sul controllo utilizzato (1.5.1). Nel capitolo 2 viene illustrata la teoria di base della *Computer Vision* in particolar modo della stereovisione, infine nel paragrafo 2.3 viene introdotta una metodologia di autocalibrazione sviluppata in questa tesi. Nel capitolo 3 viene sinteticamente trattata la teoria dello spazio delle configurazioni ( $\mathcal{C}$ -Space). Nel quarto capitolo viene visionato il server di visione sviluppato in questa tesi, introducendo due diversi metodi per il calcolo del  $\mathcal{C}$ -Space. Nel quinto capitolo viene presentato il metodo per l'*obstacle avoidance* da noi creato ( $\mathcal{C}$ -Force), concludendo con una serie di simulazioni che ne provano la correttezza. In fine nel settimo capitolo vengono presentati i risultati ottenuti da simulazioni in laboratorio.

## Stato dell'arte

In passato sono state discusse varie metodologie per garantire la sicurezza di operatori umani che condividano lo spazio di lavoro con manipolatori robotici. Molti di questi utilizzano sensori che forniscono solo informazioni locali, ad esempio [8] e [9] utilizzano dei sensori che si attivano se un ostacolo è a meno di 400mm, quando il sensore si attiva la velocità del manipolatore viene ridotta fino a fermarsi. In [10] viene utilizzata una videovideocamera per effettuare il *path planning* però viene presupposto l'utilizzo di sensori di prossimità e la conoscenza dell'immagine che si otterrà al raggiungimento del *target*. In [11] la collisione viene evitata attraverso una forza repulsiva dipendente dalla distanza fra l'*end-effector* e l'ostacolo, tale soluzione però non è efficace a meno di porre dei pesanti vincoli sulle possibili configurazioni. In [12] viene utilizzato un sistema di più videocamere disposte opportunamente per verificare la distanza fra il manipolatore e l'ostacolo, si deve notare come non sia proposta una manovra evasiva ma semplicemente si rallenti il manipolatore fino a farlo fermare. Con la stessa configurazione di videocamere del metodo precedente in [13] viene proposta una manovra evasiva, in particolare dato un task da svolgere viene proiettato sul piano immagine il manipolatore nelle configurazioni che dovrà assumere. Andando a valutare la sovrapposizione dell'immagine del manipolatore e di quella dell'ostacolo per ogni videocamera il metodo valuta se il percorso è libero o meno. Non avendo però nessuna informazione sulla profondità degli oggetti, tale metodo è efficace solo utilizzando molte videocamere, in modo da avere immagini da più angolazioni. In [14] è utilizzata la stereovisione e, come nel nostro sistema, le videocamere sono solidali al manipolatore. Gli ostacoli vengono elusi efficacemente a patto che siano posizionati su un tavolo bianco davanti al manipolatore; sono evitati aggirandoli sulla terza dimensione.

In definitiva esistono molti metodi efficaci da un punto di vista di funzionamento. Ognuna di queste soluzioni pone però dei vincoli, quali ad esempio la posizione dell'ostacolo, o prevede delle limitazioni, come un numero elevato di videocamere fisse nello spazio.

Il metodo che presenteremo è in grado di evitare ostacoli senza imporre vincoli di alcun genere (posizione, forma, dimensioni. . .). Vedremo inoltre come con le opportune modifiche che realizzeremo nell'immediato futuro la nostra metodologia possa essere esportata su manipolatori mobili (spazio di lavoro variabile).

# Capitolo 1

## Manipolatore Robotico

In questo capitolo verrà descritto il manipolatore robotico utilizzato per le prove pratiche. Verrà introdotta la struttura del manipolatore e le matrici della cinematica, poi verranno presentate le caratteristiche degli attuatori e dei sensori; In fine verrà esposto il controllo adottato.

Il manipolatore utilizzato è il *SoftArm* realizzato all'interno del Centro di Ricerca interdipartimentale "E. Piaggio" da *Giovanni Tonietti e Stefano Lodi Rizzini*.

### 1.1 Struttura

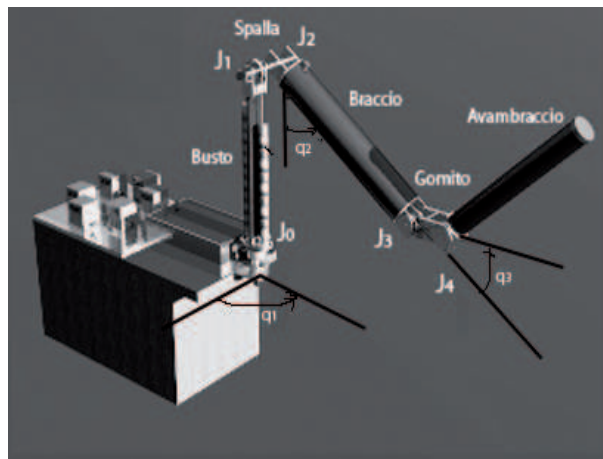


Figura 1.1: Schema del manipolatore

Il *SoftArm* è un manipolatore antropomorfo seriale a tre gradi di libertà, realizzato in Alluminio C15.

$J_0$ ,  $J_1$  e  $J_3$  (figura 1.1) sono gli unici giunti attuati e sensorizzati,  $J_2$  è un giunto fisso, mentre  $J_4$  è un giunto passivo (paragrafo 1.1.1). Ogni giunto è attuato da una coppia di muscoli McKibben, disposti in configurazione agonista-antagonista.

In tabella 1.1 si riportano lunghezze e masse<sup>1</sup> dei vari link.

	<i>Busto</i>	<i>Spalla</i>	<i>Braccio</i>	<i>Gomito</i>	<i>Avambraccio</i>
masse [Kg]	0.5	0.16	0.44	0.27	0.35
lunghezze [cm]	38.5	9	33	6	22.5

Tabella 1.1: Masse e lunghezze dei link

In tabella 1.2 si riportano i diametri delle pulegge che costituiscono i vari giunti:

	$J_0$	$J_1$	$J_2$	$J_3$	$J_4$
diametri [cm]	5	5	4	3	3

Tabella 1.2: Dimensione delle pulegge ai giunti

### 1.1.1 Aumento della destrezza

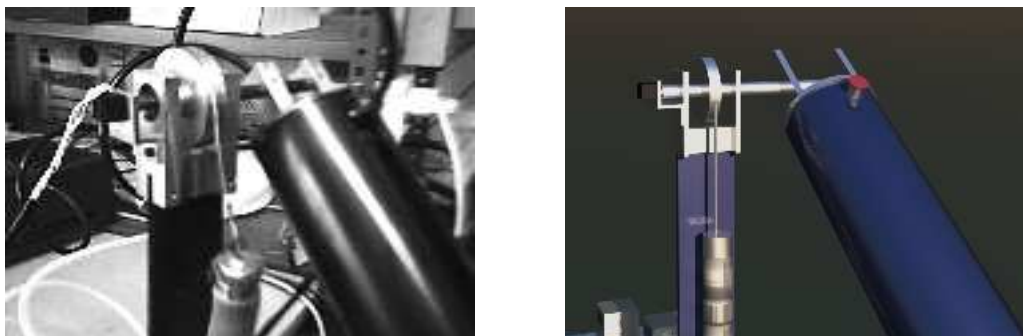


Figura 1.2: Offset di spalla: foto e schema.

Il link di spalla è stato introdotto al fine di aumentare la destrezza del manipolatore, infatti la sua presenza consente al giunto di spazzare un angolo

<sup>1</sup>Nella massa dei link si considera anche quella dell'apparato di attuazione

di  $180^\circ$ ;  $J_2$  è puramente passivo essendo l'angolo tra il link di spalla ed il link di braccio sul piano y-z di  $90^\circ$ , mentre il valore a riposo di  $q_2$  è regolabile agendo sulla puleggia.

Come il link di spalla anche quello del gomito è stato inserito per aumentare

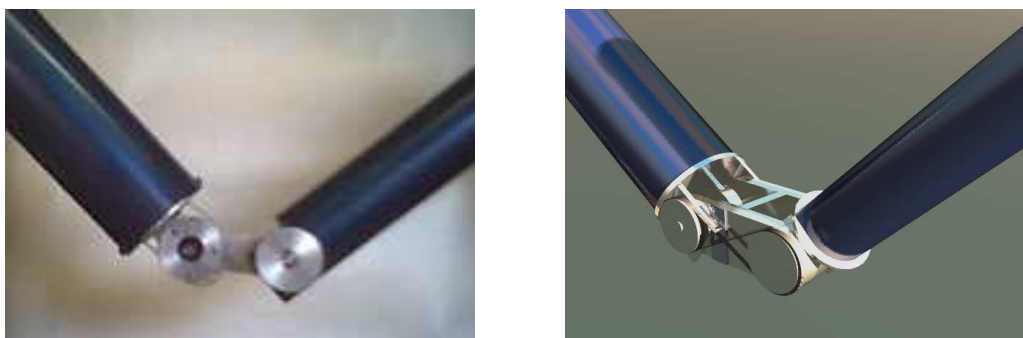


Figura 1.3: Gomito: foto e schema.

la destrezza del manipolatore.  $J_3$  è attuato tramite una coppia di muscoli alloggiati all'interno del link di busto, mentre  $J_4$  è azionato tramite un tendine in acciaio dello spessore di 0.25 mm fissato alle pulegge per mezzo di una coppia di viti. E' stato scelto l'acciaio in quanto, essendo inestensibile, smorza notevolmente eventuali moti oscillatori che complicherebbero il modello.

Con lo scopo di realizzare un moto angolare concorde per  $J_3$  e  $J_4$ , si è scelto di dare al cavo di acciaio la forma di un otto, per realizzare la quale è stata praticata un'apertura nella parte centrale ed un restringimento all'estremità, come si vede dalla figura 1.3.

Le due pulegge, in  $J_3$  e  $J_4$ , hanno lo stesso raggio  $\Delta\theta_3 = \Delta\theta_4$ . Indicando con  $R_1$   $R_2$  il raggio delle pulegge,  $\Delta\theta_1$   $\Delta\theta_2$  l'angolo compiuto e  $\Delta x_1$   $\Delta x_2$  l'arco corrispondente si può vedere che:  $\Delta x_1 = R_1 \Delta\theta_1$  e  $\Delta x_2 = R_2 \Delta\theta_2$  ma  $\Delta x_1 = \Delta x_2$  e quindi  $R_1 \Delta\theta_1 = R_2 \Delta\theta_2$  da cui imponendo  $R_1 = R_2$ , si ottiene:  $\Delta\theta_1 = \Delta\theta_2$ . Per questo motivo entrambi gli angoli sono stati definiti come  $q_3$ , e non si è introdotta nella trattazione la variabile di giunto  $q_4$ .

## 1.2 Cinematica

In questa sezione introduciamo il modello cinematico del manipolatore, ovvero quel modello che tiene conto dei parametri geometrici del robot, come le lunghezze dei giunti, la loro disposizione reciproca e gli eventuali offset tra un link e l'altro.

### 1.2.1 Parametri di Denavit-Hartenberg

Il *SoftArm* è un manipolatore a tre gradi di libertà, ma il particolare tipo di gomito richiede la specifica di un quarto link nella descrizione di Denavit-Hartenberg. In effetti però l'attuazione degli ultimi due (gomito e avambraccio) non è indipendente quindi a tutti gli effetti appariranno solo tre variabili di giunto (paragrafo 1.1.1):  $q_1, q_2$  e  $q_3$ . la descrizione dei parametri DH è la seguente mostrata nella tabella 1.3

link	d	$\theta$	a	$\alpha$
1	$L_1$	$\theta_1$	0	$\pi/2$
2	$-L_{12}$	$\theta_2 - \frac{\pi}{2}$	$L_2$	0
3	0	$\theta_3$	$L_{23}$	0
4	0	$\theta_3$	$L_3$	0

Tabella 1.3: Parametri di DH nell'ordine di applicazione delle trasformazioni rigide

Possiamo scrivere la matrice di transizione associata a questa descrizione. Sapendo che

$$\text{DH}\{\text{Link}_{i-\text{esimo}}\} \rightarrow {}^{i-1}R_i = \begin{pmatrix} C_{\theta_i} & -S_{\theta_i}C_{\alpha_i} & S_{\theta_i}S_{\alpha_i} & a_iC_{\theta_i} \\ S_{\theta_i} & C_{\theta_i}C_{\alpha_i} & -C_{\theta_i}S_{\alpha_i} & a_iS_{\theta_i} \\ 0 & S_{\alpha_i} & C_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}^2.$$

### 1.2.2 Cinematica diretta

La cinematica diretta consente di trovare la posizione e l'orientazione dell'end-effector in funzione delle variabili di giunto. Questo risultato è semplice da raggiungere nel caso di manipolatori seriali come il *SoftArm* e deriva direttamente dalla  ${}^0T_E$  di Denavit-Hartenberg. In effetti  ${}^0T_E[1..3, 4]$  è proprio la posizione dell'end-effector, così come  ${}^0T_E[1..3, 1..3]$  ne rappresenta l'orientazione rispetto al riferimento base.

<sup>2</sup>Con  $C_{xyz}$  indichiamo  $\cos(x + y + z)$ , e con  $S_{xyz}$  indichiamo  $\sin(x + y + z)$

$$\begin{aligned}
{}^0T_E &= \prod_{i=1}^n {}^{i-1}R_i \text{ avremo} \\
{}^0T_E &= \begin{pmatrix} C_1 S_{233} & C_1 C_{233} & S_1 & -L_{12} S_1 + (L_2 S_2 + L_{23} S_{23} + L_3 S_{233}) C_1 \\ S_1 S_{233} & S_1 C_{233} & -C_1 & L_{12} C_1 + (L_2 S_2 + L_{23} S_{23} + L_3 S_{233}) S_1 \\ -C_{233} & S_{233} & 0 & L_1 - L_2 C_2 - L_{23} C_{23} - L_3 C_{233} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\
&= \left( \begin{array}{c|c} \textit{rotazione} & \textit{traslazione} \\ \hline \textit{prospettiva} & \textit{scala} \end{array} \right). \tag{1.1}
\end{aligned}$$

Può essere d'aiuto per una maggiore comprensione il gruppo di proiezioni in figura 1.4 <sup>3</sup>:

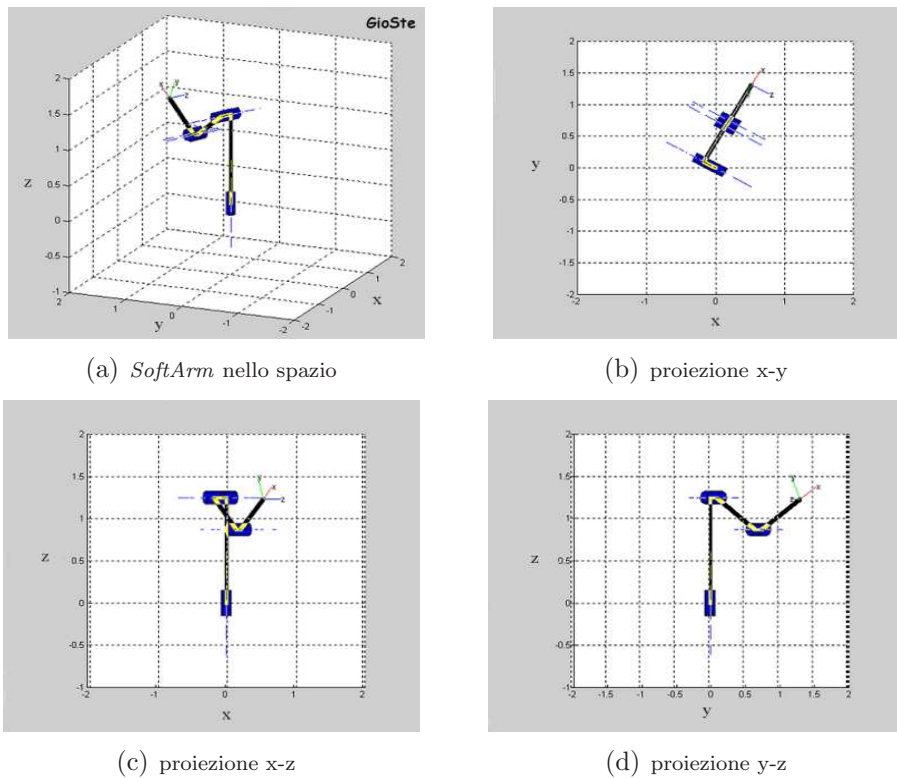


Figura 1.4: Proiezioni di una configurazione del *SoftArm*

Applicando una alla volta le matrici di trasformazione ottenuti dalle righe della tabella 1.3 otteniamo le seguenti matrici.

<sup>3</sup>ottenute con *Robotics Toolbox* di Matlab<sup>©</sup>



- link 1: L1

$$p1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \lambda_1 \end{bmatrix} \quad (1.2)$$

- offset: L12

$$p12 = \begin{bmatrix} x_{12} \\ y_{12} \\ z_{12} \end{bmatrix} = \begin{bmatrix} -\sin(q_1)\lambda_{12} \\ \cos(q_1)\lambda_{12} \\ L_1 \end{bmatrix} \quad (1.3)$$

- link 2: L2

$$p2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} -\sin(q_1)L_{12} + \lambda_2 \sin(q_2) \cos(q_1) \\ \cos(q_1)L_{12} + \lambda_2 \sin(q_2) \sin(q_1) \\ L_1 - \lambda_2 \cos(q_2) \end{bmatrix} \quad (1.4)$$

- link 3: L23

$$p23 = \begin{bmatrix} x_{23} \\ y_{23} \\ z_{23} \end{bmatrix} = \begin{bmatrix} -S_1 L_{12} + L_2 S_2 C_1 + \lambda_{23} S_{23} C_1 \\ C_1 L_{12} + L_2 S_2 S_1 + \lambda_{23} S_{23} S_1 \\ L_1 - L_2 C_2 - \lambda_{23} C_{23} \end{bmatrix} \quad (1.5)$$

- link 4: L3

$$\begin{aligned} p3 &= \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \\ &= \begin{bmatrix} -S_1 L_{12} + [L_2 S_2 + L_{23} S_{23} + \lambda_3 S_{233}] C_1 \\ C_1 L_{12} + [L_2 S_2 + L_{23} S_{23} + \lambda_3 S_{233}] S_1 \\ L_1 - L_2 C_2 - L_{23} C_{23} - \lambda_3 C_{233} \end{bmatrix} \end{aligned} \quad (1.6)$$

Le matrici ottenute saranno utili per conoscere la posizione dei punti che compongono il manipolatore al variare della configurazione; ad esempio assumendo che l'end-effector si trovi all'estremità del quarto link basterà sostituire nella 1.6  $\lambda_3$  con  $L_3$  ovvero:

$$X_{EE} = p3|_{\lambda_3=L_3} = \begin{bmatrix} -S_1 L_{12} + [L_2 S_2 + L_{23} S_{23} + L_3 S_{233}] C_1 \\ C_1 L_{12} + [L_2 S_2 + L_{23} S_{23} + L_3 S_{233}] S_1 \\ L_1 - L_2 C_2 - L_{23} C_{23} - L_3 C_{233} \end{bmatrix}, \quad (1.7)$$

che corrisponde alla 1.1

### 1.2.3 Workspace

lo spazio di lavoro (*workspace*) è composto dall'insieme dei punti che l'*end-effector* può assumere. Riportiamo varie immagini ottenute con Matlab<sup>®</sup> e l'aiuto del *Robotics Toolbox*. Ciascuna di esse è ottenuta come sovrapposizione di più superfici in  $R^3$ . Ciascuna superficie rappresenta l'area "spazzata" dall'end-effector al variare di  $q_1, q_2$  e  $q_3$  nei loro intervalli di definizione.

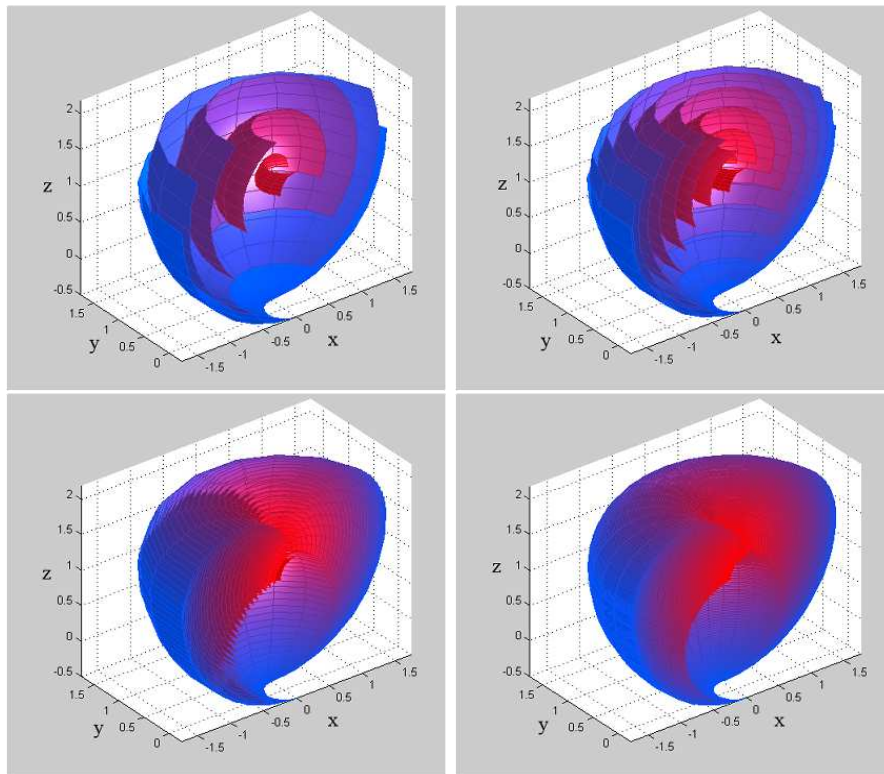


Figura 1.5: Rendering del workspace ottenuti con  $q_3$  variabile in 5, 9, 33 e 129 passi.

Il Workspace rappresentato in figura 1.5, figura 1.6 e figura 1.7 tiene conto dei limiti cinematici dei giunti. come vedremo nel paragrafo 1.3.1 i giunti non riescono a raggiungere i limiti cinematici, dunque il Workspace risulta molto più piccolo.

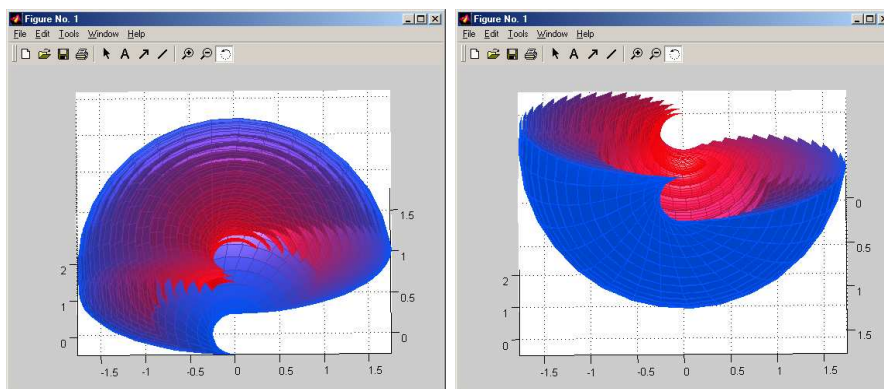


Figura 1.6: Da sinistra a destra, visione del workspace dall'alto e dal basso.

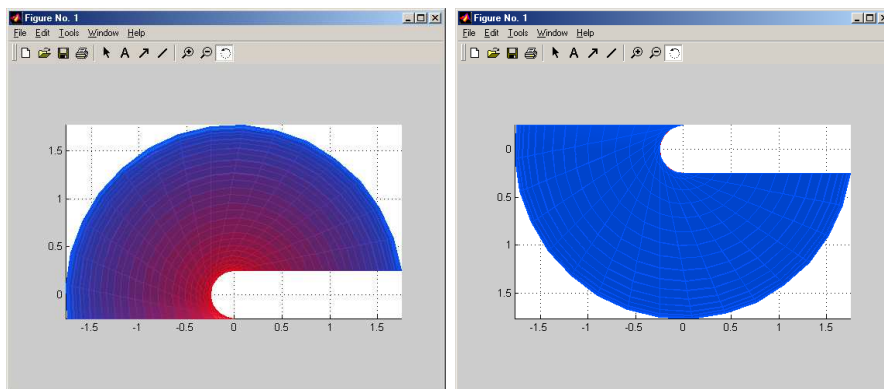


Figura 1.7: Proiezioni del workspace sul piano  $x-y$  dall'alto (a sinistra) e dal basso (a destra).

## 1.3 Attuatori

I giunti sono attuati tramite una coppia di muscoli artificiali per ogni giunto attuato, la pressione all'interno di ognuno dei quali è stabilita da un regolatore elettropneumatico *ITV2000* di *SMC*, figura 1.8.

La caratteristica fondamentale di questi regolatori è la linearità tra segnale applicato in ingresso e pressione prodotta in uscita.

Le elettrovalvole vengono controllate da un calcolatore utilizzando la scheda NI PCI 6704.



Figura 1.8: Alloggiamento dei muscoli e regolatori elettropneumatici.

### 1.3.1 Muscoli artificiali

I muscoli artificiali utilizzati per l'attuazione del *SoftArm* sono composti ciascuno da una guaina in gomma ricoperta da una rete a maglie incrociate in materiale plastico inestensibile.

L'inestensibilità del filo che compone il rivestimento conferisce al muscolo stiffness infinita al raggiungimento della massima estensione, in quanto fissa la massima lunghezza conseguibile, inoltre controbilancia l'allungamento dovuto alla forza causata dalla pressione alle estremità.

Il modello matematico da noi adottato per questo tipo di attuatori è quello sviluppato da Ching-Ping Chou e Blake Hannaford, riportato in [6]:

$$F = K_M(L^2 - L_{min}^2), \quad (1.8)$$

dove  $F$  rappresenta la forza sviluppata dal muscolo,  $K_M$  è un parametro che dipende da vari fattori quali ad esempio spessore ed estensione del filo ed il diametro del muscolo,  $L$  è la lunghezza del muscolo mentre  $L_{min}$  è la minima lunghezza teoricamente raggiungibile.

### 1.3.2 Controllo attuatori

La configurazione agonista - antagonista utilizzata per l'attuazione del giunto è mostrata in figura 1.9. Il controllo di un giunto viene realizzato variando opportunamente la pressione nel muscolo agonista e in quello antagonista. Considerando la figura 1.9 avremo

$$\tau = (F_1 - F_2)r. \quad (1.9)$$

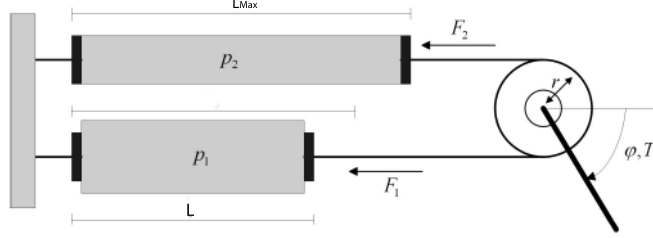


Figura 1.9: Configurazione agonista-antagonista.

Partiamo dal caso di un manipolatore con un solo giunto rotoidale e attuato da una coppia di muscoli. Con riferimento al modello di Chou-Hannaford (1.8), la relazione tra la coppia  $\tau$  e le due pressioni,  $p_1$  e  $p_2$ , nei due muscoli è

$$\tau = k_1 \phi_1 p_1 - k_2 \phi_2 p_2, \quad (1.10)$$

dove  $k_1$  e  $k_2$  sono le costanti elastiche dei muscoli e  $\phi_1$  e  $\phi_2$  sono funzioni della sola  $q$  date dalla equazione 1.8. Assumiamo che sia valida l'approssimazione

$$k_1 \simeq k_2 \triangleq k$$

e valgano le relazioni

$$\begin{cases} p_1 + p_2 \triangleq p_{MAX} \\ p_1 \triangleq p \\ p_2 = p_{MAX} - p_1 = p_{MAX} - p. \end{cases}$$

È importante notare che  $p_{MAX}$ , pressione massima nei muscoli, viene impostata dall'utente e non va inserita tra i parametri da stimare. Adesso possiamo riscrivere la (1.10) come

$$\begin{aligned} \tau &= k \phi_1 p - k \phi_2 (p_{MAX} - p) = \\ &= k (\phi_1 + \phi_2) p - k \phi_2 p_{MAX} = \\ &= k \omega(q) p - k \phi_2 p_{MAX}, \end{aligned} \quad (1.11)$$

dove  $\omega(q) = \phi_1 + \phi_2$ . Portiamo il termine  $-k \phi_2 p_{MAX}$  al primo membro e dividiamo per  $k \omega(q)$ . Si ottiene così

$$\frac{\tau}{k \omega(q)} + \frac{\phi_2}{\omega(q)} p_{MAX} = p.$$

La precedente operazione di divisione è stata possibile in virtù del fatto che valgono sempre le relazioni

$$\begin{cases} k \neq 0 \\ \omega(q) = \phi_1 + \phi_2 \neq 0. \end{cases}$$

Vediamo ora di estendere la trattazione al caso di  $n$  giunti, ciascuno dei quali viene attuato da una coppia di muscoli pneumatici. A partire dalla (1.11) si deduce che la relazione tra la coppia  $\tau_i$  al giunto  $i$  e le rispettive pressioni è data da

$$\tau_i = k_i \omega_i(q_i) p_i - \phi_{i2} p_{iMAX}.$$

In forma compatta le  $n$  equazioni si scrivono come

$$\tau = \mathbb{K} \Omega(q) p - \phi_2 p_{MAX},$$

avendo definito

$$\mathbb{K} = \begin{bmatrix} k_1 & 0 & \cdots & 0 \\ 0 & k_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & k_n \end{bmatrix}, \quad \Omega(q) = \begin{bmatrix} \phi_{11} + \phi_{12} & 0 & \cdots & 0 \\ 0 & \phi_{21} + \phi_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \phi_{n1} + \phi_{n2} \end{bmatrix},$$

dove

$$\begin{aligned} p &= (p_1, \dots, p_n)^T, & p_{MAX} &= (p_{1MAX}, \dots, p_{nMAX})^T, \\ \phi_2 &= (\phi_{21}, \dots, \phi_{2n})^T, & \tau &= (\tau_1, \dots, \tau_n)^T. \end{aligned}$$

## 1.4 Sensori



Figura 1.10: Collocazione potenziometro.

Per ottenere la configurazione attuale del manipolatore ogni giunto attuato è stato dotato di un potenziometro (figura 1.10). Per campionare il segnale di tensione in uscita dai potenziometri è stata usata una scheda di acquisizione dati NI PCI 6024E di *National Instruments*.

### 1.4.1 Calibrazione sensori

La caratteristica angolo-tensione dei potenziometri, di norma, non è lineare. Per ottenere una stima di tale caratteristica abbiamo valutato la tensione al variare dell'angolo. I risultati di tale prova sono rappresentati in figura 1.11. Come si vede dalla figura 1.11 un'approssimazione lineare della caratteristica è fattibile, in particolare si è assunto  $\theta_1 \in [-60, 60]$  gradi,  $\theta_2 \in [0, 55]$  e  $\theta_3 \in [0, 50]$ ; le prove sono state effettuate con una risoluzione di 5 gradi.

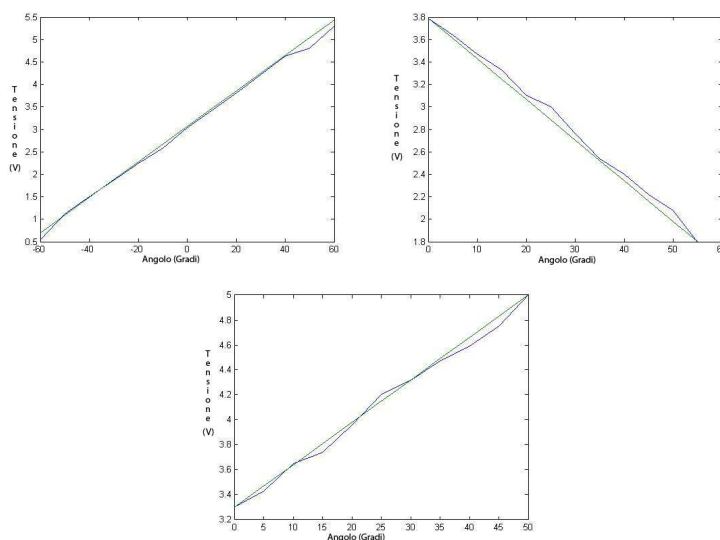


Figura 1.11: Relazione angolo tensione dei 3 potenziometri.

## 1.5 Controllo

Per il controllo del manipolatore abbiamo scelto di puntare sulla semplicità computazionale dell'algoritmo in modo da non occupare tempo di CPU eccessivo per i calcoli del controllore. A tal proposito abbiamo scelto di utilizzare un controllo alla *Arimoto*. L'algoritmo è stato prima simulato su Simulink<sup>®</sup> ed in seguito realizzato in C++.

### 1.5.1 Controllo alla *Arimoto*

Consideriamo la forma standard della dinamica di un manipolatore

$$\tau = H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q), \quad (1.12)$$

Nel controllo alla *Arimoto* si va a valutare solo gli effetti dinamici dovuti alla gravità, si considerano quindi le stime

$$\bar{H} = I, \bar{C} = 0, \bar{G} = G \quad (1.13)$$

e si applica un controllo punto-punto, cioè per cui  $\dot{\hat{q}} = \ddot{\hat{q}} = 0$  dove  $\hat{q}$  è la posizione desiderata. Applicando un controllo a *coppia calcolata* [4] usando le stime 1.13 si ha

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = K_p e - K_v \dot{q} + G(q) \quad (1.14)$$

dove  $e = \hat{q} - q$ . Studiamo la stabilità mediante la funzione di *Lyapunov*

$$V(q, \dot{q}) = \frac{1}{2} \dot{q}^T H \dot{q} + \frac{1}{2} e^T K_p e \quad (1.15)$$

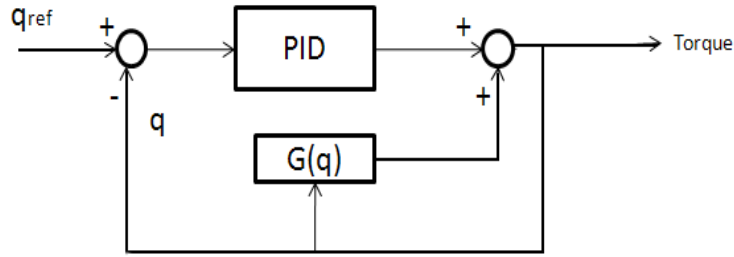
definita positiva; la sua derivata sarà

$$\begin{aligned} \dot{V} &= \dot{q}^T H \ddot{q} + \frac{1}{2} \dot{q}^T \dot{H} \dot{q} - e^T K_p \dot{q} = \\ &= \dot{q}^T K_p e - \dot{q}^T K_v \dot{q} - \dot{q}^T C \dot{q} + \frac{1}{2} \dot{q}^T \dot{H} \dot{q} - \dot{e}^T \dot{q} = \\ &= -\dot{q}^T K_v \dot{q} + \frac{1}{2} \dot{q}^T (\dot{H} - 2C) \dot{q} = \\ &= -\dot{q}^T K_v \dot{q} \leq 0. \end{aligned}$$

la derivata della funzione di Liapunov 1.15 è semidefinita negativa. Utilizzando il teorema di *Krasowsky* possiamo affermare che essendo  $\dot{V} = 0$  solo per  $\dot{q} = 0$  (escludendo  $K_v = 0$  che non ha interesse pratico) il sistema è ASINTOTICAMENTE STABILE. Praticamente si inserisce un controllore PD su ogni giunto.

Se  $\bar{G} \neq G$  si deve inserire un'azione integrante per eliminare l'errore a regime, per cui il controllore utilizzato sarà un PID (schema 1.12).



Figura 1.12: Schema controllo alla *Arimoto*.

## 1.5.2 Realizzazione discreta

Come discusso nel paragrafo 1.5.1 il controllore utilizzato sarà un PID più il recupero della gravità. Il controllo verrà effettuato da un calcolatore quindi dovrà lavorare a tempo discreto. Consideriamo la realizzazione base si un controllore PID :

$$y(t) = K_p e(t) + K_d \frac{de(t)}{dt} + K_i \int_{-\infty}^t e(t) dt \quad (1.16)$$

Dove  $e(t) = q_{ref} - q(t)$

Considerando la filosofia del minimo carico della CPU per il calcolo del controllore abbiamo deciso di realizzare il derivatore del primo ordine e l'integratore di tipo trapezoidale.

- *derivatore*

il derivatore è troncato al primo ordine, da cui

$$\frac{de(t)}{dt} \Rightarrow \frac{e(k) - e(k-1)}{T_s}$$

- *integratore*

l'integratore è di tipo trapezoidale

$$\int_{-\infty}^t e(t) dt \Rightarrow I(k) = I(k-1) + \frac{T_s}{2} (e(k) + e(k-1))$$

dove  $T_s$  è il tempo di campionamento.

il controllore discretizzato è :

$$I(0) = \langle \text{stato iniziale} \rangle$$

$$I(k) = I(k-1) + \frac{T_s}{2} (e(k) + e(k-1))$$

$$\text{Torque}(k) = K_p e(k) + K_d \frac{e(k) - e(k-1)}{T_s} + K_i I(k) + g_i(q, k), \quad (1.17)$$

dove  $g_i(q, k)$  rappresenta la componente  $i$ -esima del vettore di gravità (appendice C.8)

### 1.5.3 Riferimento Smorzato

Grazie alla prontezza di risposta delle elettrovalvole il *SoftArm* riesce a muoversi velocemente nello spazio di lavoro. Tale velocità è però incompatibile con i tempi caratteristici dell'algoritmo di visione. Per rallentare la dinamica del manipolatore il riferimento da seguire viene smorzato secondo l'equazione :

$$q_{ref}(t) = q_{voluta} - \Delta q e^{-|\frac{k}{\Delta q}|(t-\bar{t})}, \quad (1.18)$$

dove  $q_{ref}(t)$  è l'angolo di riferimento al tempo  $t$ ,  $q_{voluta}$  è il valore richiesto,  $\Delta q$  è data da  $\Delta q = q(\bar{t}) - q_{voluta}$  dove  $q(\bar{t})$  è l'angolo che si aveva al tempo  $\bar{t}$ ;  $k$  è un parametro che varia la ripidità della funzione. Un possibile andamento

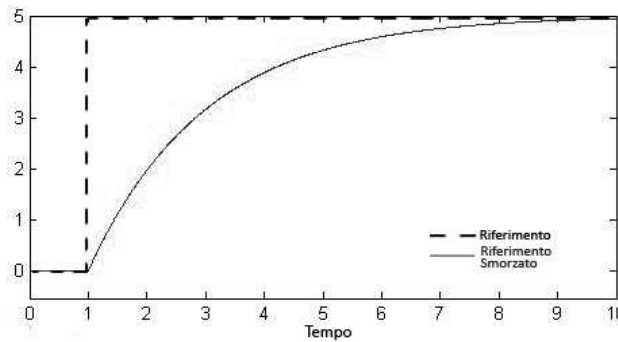


Figura 1.13: Andamento della funzione 1.18.

della funzione 1.18 è mostrato in figura 1.13.

### 1.5.4 Simulazione

Il controllore descritto al paragrafo 1.5.1 è stato simulato utilizzando il pacchetto Simulink<sup>®</sup> di Matlab<sup>®</sup>. Per le simulazioni abbiamo utilizzato un modello del *SoftArm* che realizza la dinamica del manipolatore (appendice C.4) includendo il modello dei muscoli 1.3.2. Lo schema utilizzato per le simulazioni è mostrato in figura 1.14. Il controllore è stato realizzato utilizzando il modello discretizzato del controllo alla *Arimoto* (paragrafo 1.5.2). In figura 1.15 è mostrato lo schema Simulink<sup>®</sup> del controllore.

Le simulazioni sono state effettuate considerando un tempo di campionamento per il controllore pari a  $1ms$  ( $1KHz$ ). La calibrazione dei parametri

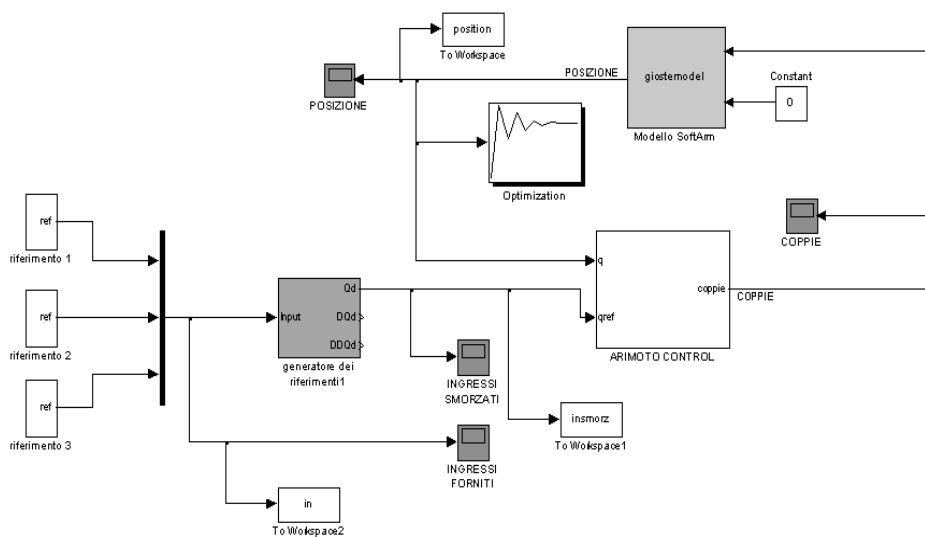


Figura 1.14: Schema Simulink<sup>®</sup>: dinamica del manipolatore.

dei tre PID è stata effettuata utilizzando il *Response Optimization<sup>®</sup> Toolbox* di Simulink<sup>®</sup>. In figura 1.16 è mostrato il risultato della simulazione effettuata utilizzando come riferimento un gradino unitario (1 radiante). Il riferimento viene smorzato secondo la funzione 1.18. In figura 1.16 si nota l'efficacia del controllo adottato. In figura 1.17 è mostrato il risultato della simulazione ottenuto cambiando la configurazione di riferimento in tempi finiti. I risultati delle simulazioni hanno dimostrato come il controllo adottato consente di ottenere ottimi risultati, nonostante la sua semplicità.

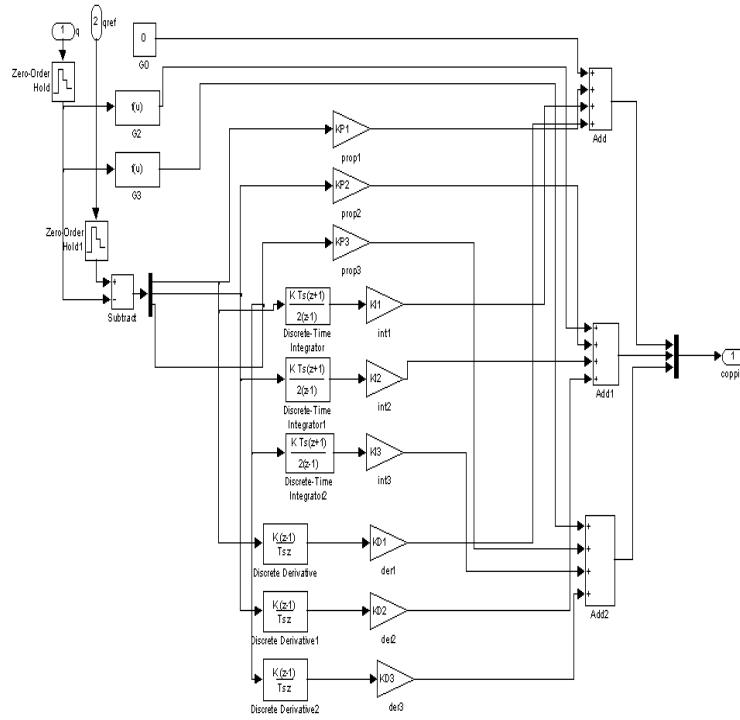


Figura 1.15: Schema Simulink<sup>®</sup> del controllore.

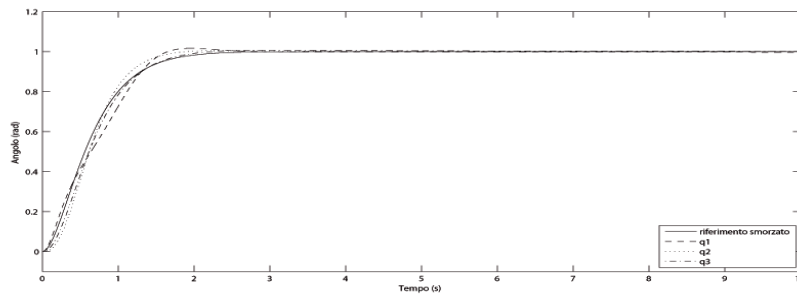


Figura 1.16: Risposta a gradino.

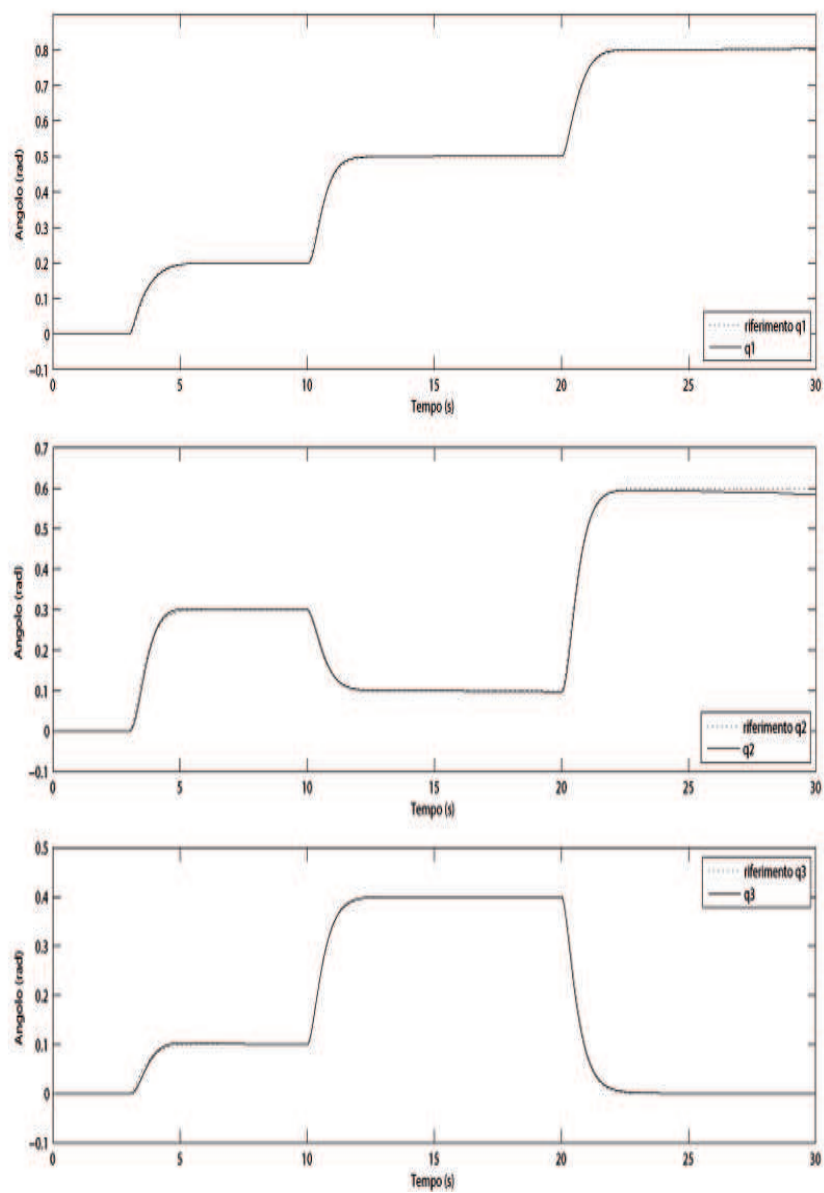


Figura 1.17: Simulazione: configurazione variabile.

# Capitolo 2

## Visione Stereo

In questo capitolo verrà trattata brevemente la teoria di base della visione, partendo dal modello matematico di una videocamera, proseguendo con la geometria epipolare di un sistema stereo, per concludere con il sistema di autocalibrazione a videocamere semicalibrate da noi sviluppato.

### 2.1 Videocamera: funzionamento e modello matematico

La *Computer Vision* si basa sulla manipolazione delle immagini catturate da videocamere. Una videocamera è un trasduttore che converte i segnali luminosi provenienti dall'esterno in una matrice di pixel, in cui ogni pixel contiene una rappresentazione digitale dell'intensità luminosa del rispettivo raggio di luce. In una videocamera monocromatica la matrice di pixel sarà ad una sola dimensione, mentre in una videocamera a colori la matrice è a più dimensioni, solitamente tre e contiene ad esempio l'intensità, del rosso, verde e blu (RGB). La scelta di una videocamera per applicazioni in tempo reale deve basarsi su :

- **risoluzione**, dimensione in pixel della suddetta matrice;
- **frame rate**, numero di frame che riesce a catturare in un secondo;
- **tempo di esposizione**, tempo per cui l'otturatore resta aperto, ovvero il tempo per cui viene catturato il raggio luminoso.

Di massima importanza per la *Computer Vision* è sapere quale è il raggio luminoso che corrisponde ad un pixel, ovvero sapere come la videocamera rappresenta geometricamente lo spazio Euclideo catturato.

### 2.1.1 Parametri Intrinseci

Esistono vari modelli matematici che descrivono come la videocamera proietti i raggi luminosi nel piano immagine. Il più semplice, ed anche più utilizzato, è il modello *Pinhole*. Nel modello *Pinhole* si ipotizza che tutti i raggi passano per il medesimo punto  $C$ , chiamato *centro di proiezione* (figura 2.1) (oppure centro ottico). Di conseguenza gli unici punti che contribuiscono a irradiare il punto nell'immagine con coordinate  $\mathbf{x} = [x, y]^T$  si trovano su una linea passante per  $C$ . Se un punto  $\mathbf{p}$  ha coordinate  $\mathbf{X} = [X, Y, Z]^T$  relative ad un

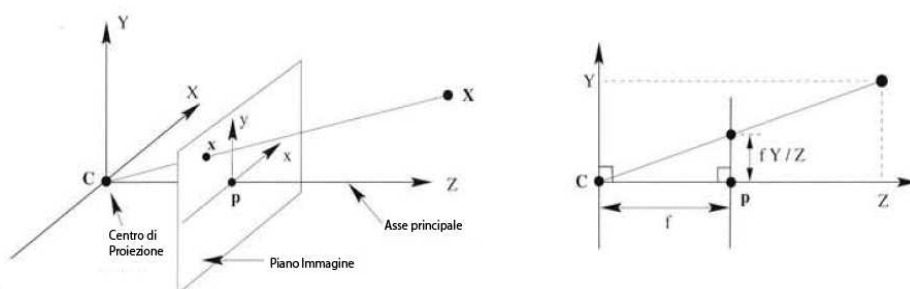


Figura 2.1: Geometria del modello Pinhole.

sistema di riferimento centrato sul *centro di proiezione*, come in figura 2.1, le coordinate di  $\mathbf{p}$  e la sua immagine  $\mathbf{x}$ , sono date dalla relazione

$$x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z}, \quad (2.1)$$

dove  $f$  è chiamata la *lunghezza focale*.

Possiamo dire che la proiezione è una mappa  $\pi$  :

$$\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2; \quad \mathbf{X} \rightarrow \mathbf{x}. \quad (2.2)$$

possiamo anche scrivere  $\mathbf{x} = \pi(\mathbf{X})$ . Da notare che ogni altro punto che si trova nella linea passante per  $C$  e  $\mathbf{p}$  mi da lo stesso punto  $\mathbf{x}$  nell'immagine. riscrivendo l'equazione 2.1 in forma matriciale avremo :

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix}$$

che in coordinate omogenee (appendice B.1.1) diventa

$$Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}, \quad (2.3)$$

che può essere riscritta come

$$Z\mathbf{x} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{X}. \quad (2.4)$$

Il modello con equazione 2.4 è relativo ad una scelta molto particolare del sistema di riferimento. In pratica, quando viene catturata un'immagine con una videocamera digitale le misure sono ottenute in termini di pixel  $(i, j)$ , con l'origine tipicamente nell'angolo superiore sinistro dell'immagine. Per rendere il modello 2.4 utilizzabile dobbiamo specificare la relazione fra le coordinate del piano immagine e la matrice dei pixel. Il primo passo consiste nello specificare l'unità lungo l'asse  $x$  ed  $y$ : se  $(x, y)$  sono specificati in termini di unità metriche (es. millimetri), e  $(x_s, y_s)$  sono le versioni scalate che corrispondono alle coordinate del pixel, quindi la trasformazione può essere descritta da una matrice

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \quad (2.5)$$

che dipende dalla dimensione dei pixel (in unità metriche) lungo la direzione  $x$  e  $y$ . Quando  $s_x = s_y$  i pixel sono quadrati. I punti  $x_s$  e  $y_s$  sono ancora relativi al *punto principale* (punto di intersezione tra il piano immagine e l'asse  $z$ ), quindi devono essere traslati in modo da portare l'origine nell'angolo in alto a sinistra dell'immagine, come mostrato in figura 2.2.

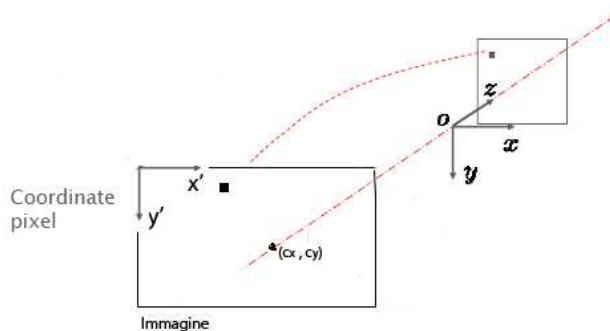


Figura 2.2: Coordinate nel piano immagine.

Siano  $(c_x, c_y)$  le coordinate (in pixel) del punto principale, relative al



sistema di riferimento dell'immagine, avremo la trasformazione

$$\mathbf{x}' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & c_x \\ 0 & s_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (2.6)$$

Tralasciando altri fattori come il *fattore di skew* e la *distorsione delle lenti*, otteniamo la matrice

$$K = \begin{bmatrix} \alpha_x & 0 & c_x \\ 0 & \alpha_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.7)$$

dove  $\alpha_x = fs_x$  ed  $\alpha_y = fs_y$ .

La matrice  $K \in \mathbb{R}^{3 \times 3}$  è chiamata *matrice dei parametri intrinseci*. Dato un punto  $\mathbf{X}$  rispetto al sistema di riferimento della videocamera, il punto corrispondente nell'immagine sarà

$$\mathbf{x} \simeq K\Pi_0\mathbf{X} = \begin{bmatrix} \alpha_x & 0 & c_x \\ 0 & \alpha_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{X}. \quad (2.8)$$

La matrice costante  $\Pi_0$  rappresenta la proiezione. Al posto del simbolo di uguaglianza abbiamo usato il simbolo  $\simeq$  per indicare che l'uguaglianza vale a meno di un fattore di scala. Tale convenzione sarà utilizzata nel resto della tesi.

## 2.1.2 Parametri Estrinseci

Generalmente, i punti nello spazio vengono espressi relativamente ad un diverso sistemi di riferimento  $O$  (sistema globale), rispetto a quello delle videocamera. Il sistema di riferimento della videocamera è legato al sistema di riferimento  $O$  da una trasformazione. vedi figura 2.3.

Sia  $\tilde{\mathbf{X}}$  un vettore di coordinate non omogenee espresso rispetto al sistema di coordinate  $O$ , e  $\tilde{\mathbf{X}}_{\text{cam}}$  lo stesso punto espresso rispetto al sistema di riferimento della videocamera, possiamo scrivere  $\tilde{\mathbf{X}}_{\text{cam}} = \mathbf{R}(\tilde{\mathbf{X}} - \tilde{\mathbf{C}})$  dove  $\tilde{\mathbf{C}}$  rappresenta la coordinata del centro del sistema di riferimento della videocamera rispetto al sistema di riferimento  $O$ , e  $\mathbf{R}$  è una matrice di rotazione, che rappresenta l'orientazione del sistema di riferimento della videocamera rispetto al sistema di riferimento  $O$ . La matrice di trasformazione omogenea

$$\mathcal{E} = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0} & 1 \end{bmatrix} \quad (2.9)$$

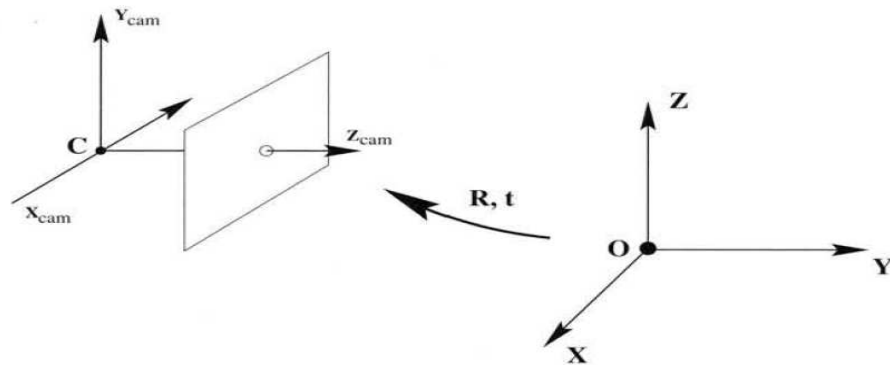


Figura 2.3: Rototraslazione data dai parametri Estrinseci.

è chiamata *matrice dei parametri estrinseci*.

### 2.1.3 Matrice di Proiezione

Per ottenere la proiezione di un punto  $\mathbf{X}$ , espresso nel sistema di riferimento globale  $O$ , nell'immagine dobbiamo prima trasformare le coordinate del punto rispetto al sistema di riferimento videocamera usando i parametri estrinseci (equazione 2.9) e poi proiettarlo nell'immagine con i parametri intrinseci (equazione 2.8).

Da cui :

$$\mathbf{x} \simeq \mathcal{P}\mathbf{X} = \mathbf{K}\mathbf{R}[\mathbf{I} \mid -\mathbf{K}\tilde{\mathbf{C}}] \quad (2.10)$$

la matrice  $\mathcal{P}$  è chiamata *matrice di proiezione* della videocamera.

## 2.2 Geometria Visione Stereo

Un sistema di visione stereo è caratterizzato da una coppia di videocamere (figura 2.4) che catturano immagini dello stesso spazio da punti diversi. Come per la visione umana questo consente di stimare la profondità degli oggetti.

### 2.2.1 Geometria Epipolare

La *geometria epipolare* tra due videocamere è essenzialmente la geometria dell'intersezione dei piani immagine con l'insieme dei piani che hanno come asse la *baseline*<sup>1</sup>.

<sup>1</sup>Linea che congiunge i centri delle due videocamere.



Figura 2.4: Stereo Visione.

Consideriamo un punto  $\mathbf{X}$  nello spazio, questo sarà proiettato in un punto  $\mathbf{x}$  nella prima immagine, ed in un punto  $\mathbf{x}'$  nella seconda. Come si vede in figura 2.5(a) i due punti  $\mathbf{x}$  e  $\mathbf{x}'$ , ed i centri di proiezione delle videocamere sono coplanari, chiamiamo questo piano  $\pi$ . chiaramente anche i raggi di proiezione dei due punti giacciono su  $\pi$ . Tale proprietà è essenziale quando si vogliono trovare le corrispondenze fra i punti delle due immagini.

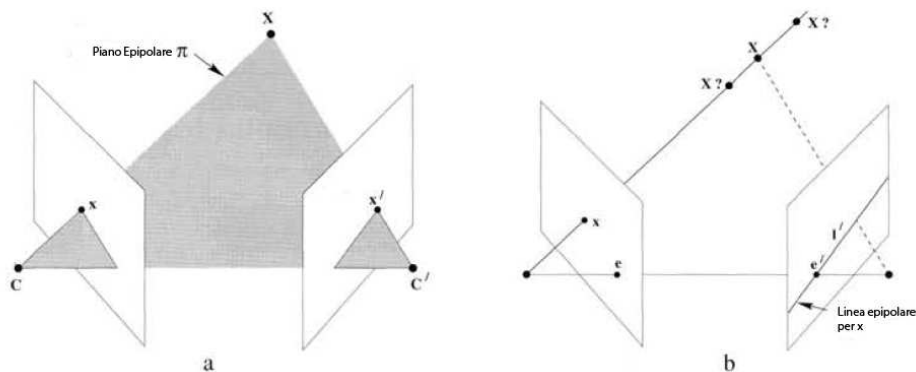


Figura 2.5: Piano Epipolare.

Supponiamo di conoscere solo la proiezione  $\mathbf{x}$ , da questa informazione vogliamo sapere dove cercare il punto corrispondente nella seconda immagine (figura 2.5(b)). Essendo la *baseline* fissa, conosciamo il piano  $\pi$ . Dovendo, anche  $\mathbf{x}'$ , essere sul piano  $\pi$  possiamo restringere la ricerca alla sola linea data dall'intersezione fra il piano immagine della seconda videocamera ed il piano  $\pi$ . Questa linea  $l'$  non è altro che la proiezione del raggio di  $\mathbf{x}$  sul piano immagine della seconda videocamera ed è chiamata la *linea epipolare* corrispondente ad  $\mathbf{x}$ .

Le entità geometriche che entrano in gioco nella geometria epipolare sono mostrate in figura 2.6.

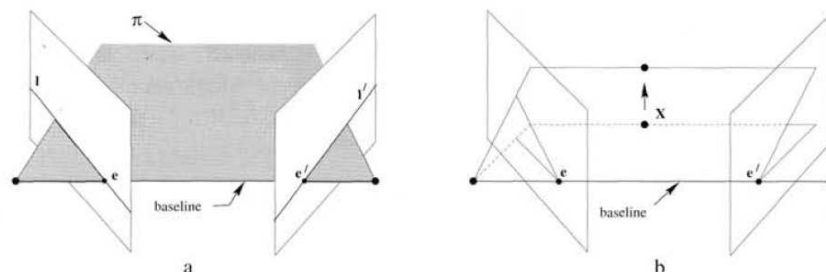


Figura 2.6: Geometria Epipolare.

La terminologia è :

- l'**epipolo** di una immagine è il punto di intersezione tra la *baseline* ed il piano immagine;
- il **piano epipolare** è una famiglia di piani che dipendono da un solo parametro, passanti per la *baseline*;
- la **linea epipolare** è l'intersezione del piano epipolare con il piano immagine;

## 2.2.2 La Matrice Fondamentale

La *matrice fondamentale* è la rappresentazione algebrica della geometria epipolare.

Data una coppia di immagini, per ogni punto  $\mathbf{x}$  nella prima immagine esiste una linea epipolare  $l'$  corrispondente, nella seconda immagine. Un punto  $\mathbf{x}'$  della seconda immagine, che corrisponde al punto  $\mathbf{x}$  deve trovarsi sulla linea epipolare corrispondente  $l'$ .

Quindi deve esistere una mappa

$$\mathbf{x} \mapsto l'$$

da un punto in un'immagine alla linea epipolare corrispondente nell'altra.

Date le matrici di proiezione delle due videocamere  $\mathcal{P}$   $\mathcal{P}'$ , possiamo derivare per via algebrica la matrice fondamentale. La proiezione alla rovescia del

raggio del punto  $\mathbf{x}$  è ottenuta risolvendo l'equazione  $\mathcal{P}\mathbf{X} = \mathbf{x}$ . Avremo una famiglia di soluzioni con un parametro

$$\mathbf{X}(\lambda) = \mathcal{P}^\dagger \mathbf{x} + \lambda C \quad (2.11)$$

dove  $\mathcal{P}^\dagger$  è la matrice pseudoinversa di  $\mathcal{P}$ , e  $C$  il suo vettore nullo chiamato centro di proiezione, definito da  $\mathcal{P}C = 0$ . Il raggio è parametrizzato dallo scalare  $\lambda$ . In particolare due punti del raggio saranno  $\mathcal{P}\mathbf{x}$  ( $\lambda=0$ ) ed il centro di proiezione della prima videocamera  $C$  ( $\lambda = \infty$ ). Questi due punti sono visti dalla seconda videocamera in  $\mathcal{P}'\mathcal{P}^\dagger \mathbf{x}$  e  $\mathcal{P}'C$ . La linea epipolare passerà per questi due punti  $l' = \mathcal{P}'C \times \mathcal{P}'\mathcal{P}^\dagger \mathbf{x}$  il punto  $\mathcal{P}'C$  è l'epipolo della seconda immagine, corrispondente alla proiezione del centro della prima videocamera, e d'ora in poi lo chiameremo  $e'$ . da cui  $l' = [e']_\times \mathcal{P}'\mathcal{P}^\dagger \mathbf{x} = \mathbf{F}\mathbf{x}^2$ .

Dove  $\mathbf{F}$  è la matrice

$$\mathbf{F} = [e']_\times \mathcal{P}'\mathcal{P}^\dagger \quad (2.12)$$

ed è chiamata *matrice fondamentale* del sistema stereo.

### 2.2.3 Condizione di Corrispondenza

Fino a questo punto abbiamo considerato la mappa  $\mathbf{x} \mapsto l'$  definita da  $\mathbf{F}$ . Andiamo ora a vedere la proprietà di base della matrice fondamentale. Per ogni coppia di corrispondenza  $\mathbf{x} \leftrightarrow \mathbf{x}'$  vale la relazione

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0. \quad (2.13)$$

Questo perché se  $\mathbf{x}$  e  $\mathbf{x}'$  corrispondono allora  $\mathbf{x}'$  giace sulla linea  $l' = \mathbf{F}\mathbf{x}$  corrispondente al punto  $\mathbf{x}$ . In altre parole  $0 = \mathbf{x}'^T l' = \mathbf{x}'^T \mathbf{F}\mathbf{x}$ .

L'importanza della relazione 2.13 è di permettere una caratterizzazione della matrice fondamentale senza l'utilizzo delle matrici di proiezione delle videocamere, per esempio usando le corrispondenze di *features*<sup>3</sup> fra le due immagini.

## 2.3 Autocalibrazione di un sistema stereo con videocamere semicalibrate

Dato un sistema di videocamere stereo per ottenere una ricostruzione metrica (dipendente solo da un fattore di scala) si devono conoscere le matrici di calibrazione di entrambe le videocamere. Avere le matrici di calibrazione di

<sup>2</sup>Con  $[e']_\times$  indichiamo la matrice skew simmetrica derivata da  $e'$  vedi appendice B.3

<sup>3</sup>Punti dell'immagine con un altro contenuto informativo

entrambe le videocamere corrisponde a conoscere sia i parametri intrinseci che estrinseci. I parametri intrinseci, che possono essere stimati utilizzando dei pattern, rappresentano come punti nello spazio vengono proiettati sul piano immagine. I parametri estrinseci rappresentano la posizione nello spazio delle videocamere. Considerare conosciuti i parametri estrinseci di entrambe le videocamere rende troppo rigido il sistema, dato che una piccola variazione nella posizione relativa delle due videocamere genera una variazione della geometria epipolare. L'utilizzo di matrici di calibrazione che non rispettano la reale geometria epipolare delle videocamere rende praticamente errata la ricostruzione ottenuta con la triangolazione.

Si vuole quindi considerare conosciuti i parametri intrinseci delle videocamere e stimare la posizione e l'orientamento relativo fra le due videocamere.

### 2.3.1 Matrici di calibrazione in forma canonica

Data la corrispondenza di almeno 8 punti fra le immagini delle due videocamere  $\mathbf{x} \leftrightarrow \mathbf{x}'$  è possibile stimare la matrice fondamentale  $\mathbf{F}$  [3]. Utilizzando il metodo *RANSAC* (*Random Sample Consensus*) si possono eliminare gli *outlier*<sup>4</sup> e rendere più robusta la stima. Una volta stimata la matrice fondamentale è possibile risalire alle matrici di proiezione che la generano.

**Teorema 2.1** *Sia  $\mathbf{H}$  una matrice  $4 \times 4$  che rappresenta una trasformazione di proiezione nello spazio 3D allora la matrice fondamentale corrispondente alla coppia di matrici di proiezione  $(\mathcal{P}, \mathcal{P}')$  sarà equivalente a quella ottenuta dalla coppia  $(\mathcal{P}\mathbf{H}, \mathcal{P}'\mathbf{H})$*

**Dimostrazione.** Si può osservare che  $\mathcal{P}\mathbf{X} = (\mathcal{P}\mathbf{H})(\mathbf{H}^{-1}\mathbf{X})$  e similmente per  $\mathcal{P}'$ . Se  $\mathbf{x} \leftrightarrow \mathbf{x}'$  sono punti di corrispondenza per le videocamere con matrice di proiezione  $(\mathcal{P}, \mathcal{P}')$ , corrispondente ad un punto  $\mathbf{X}$ , allora saranno anche i punti di corrispondenza per una coppia di videocamere con matrici di proiezione  $(\mathcal{P}\mathbf{H}, \mathcal{P}'\mathbf{H})$ , del punto  $\mathbf{H}^{-1}\mathbf{X}$ .

□

Data questa ambiguità è comune usare una *forma canonica* in cui si considera come sistema di riferimento il *frame*<sup>5</sup> della videocamera corrispondente alla matrice

$$\mathcal{P} = [\mathbf{I} \mid \mathbf{0}], \quad (2.14)$$

<sup>4</sup>Corrispondenze affette da un errore elevato.

<sup>5</sup>Con *frame* intendiamo un sistema di riferimento solidale ad un corpo.

dove  $\mathbf{I}$  è la matrice d'identità  $3 \times 3$  e  $\mathbf{0}$  è un vettore nullo di 3 componenti. Data la scelta della forma canonica 2.14 per la prima videocamera la matrice di proiezione della seconda videocamera può essere trovata utilizzando i seguenti teoremi

**Teorema 2.1** *una matrice  $\mathbf{F}$  non nulla è la matrice fondamentale di una coppia di matrici  $\mathcal{P}$   $\mathcal{P}'$  se e solo se  $\mathcal{P}^T \mathbf{F} \mathcal{P}$  è skew simmetrica (appendice B.3).*

**Dimostrazione.** La condizione che  $\mathcal{P}'^T \mathbf{F} \mathcal{P}$  sia skew simmetrica equivale a  $\mathbf{X} \mathcal{P}'^T \mathbf{F} \mathcal{P} \mathbf{X} = 0$  per ogni  $\mathbf{X}$ . Facendo il cambio di variabili  $\mathbf{x}' = \mathcal{P}' \mathbf{X}$  e  $\mathbf{x} = \mathcal{P} \mathbf{X}$ , avremo  $\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0$ , la quale è l'equazione 2.13.

□

Sia  $\mathbf{S}$  una matrice  $3 \times 3$  skew simmetrica, una possibile coppia di matrici di calibrazione in forma canonica è ottenuta del seguente teorema

**Teorema 2.1** *Sia  $\mathbf{F}$  una matrice fondamentale ed  $\mathbf{S}$  una qualsiasi matrice skew simmetrica. Possiamo definire una coppia di matrici di proiezione*

$$\mathcal{P} = [\mathbf{I} \mid \mathbf{0}] \text{ e } \mathcal{P}' = [\mathbf{S}\mathbf{F} \mid e'], \quad (2.15)$$

dove  $e'$  è l'epipolo per cui  $e'^T \mathbf{F} = \mathbf{0}$ , e ipotizzando che  $\mathcal{P}'$  sia una matrice di proiezione valida (con rango 3). allora  $\mathbf{F}$  è la matrice fondamentale corrispondente alla coppia  $(\mathcal{P}, \mathcal{P}')$ .

**Dimostrazione.** considerando il teorema 2.1 possiamo verificare semplicemente che

$$[\mathbf{S}\mathbf{F} \mid e']^T \mathbf{F} [\mathbf{I} \mid \mathbf{0}] = \begin{bmatrix} \mathbf{F}^T \mathbf{S}\mathbf{F} & \mathbf{0} \\ e'^T \mathbf{F} & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{F}^T \mathbf{S}\mathbf{F} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix}. \quad (2.16)$$

La quale è skew simmetrica.

□

Una possibile soluzione è data da

$$\mathcal{P} = [\mathbf{I} \mid \mathbf{0}] \text{ e } \mathcal{P}' = [[e']_{\times} \mathbf{F} \mid e']. \quad (2.17)$$

La matrice  $\mathcal{P}'$ , in questa forma, è dipendente dai parametri intrinseci delle due videocamere e dalla posizione ed orientazione relativa tra le due videocamere. La matrice  $\mathcal{P}'$  codifica tutto l'errore dovuto alla non conoscenza dei parametri intrinseci ed estrinseci, per questo porta ad una ricostruzione altamente distorta.

### 2.3.2 Forma canonica con riduzione della distorsione

La forma delle matrici nella formula 2.17 rappresenta una particolare scelta in cui il centro della videocamera giace sul piano ad infinito, tale condizione non è necessaria, quindi è possibile scegliere un qualsiasi  $\mathcal{P}'$  dato da

$$\mathcal{P}' = [v_4 [e']_{\times} \mathbf{F} + e' \mathbf{v}^T \mid e'], \quad (2.18)$$

dove  $v_4$  è uno scalare e  $\mathbf{v}$  è un vettore di 3 elementi. La matrice  $\mathcal{P}'$  può essere vista come  $\mathcal{P}' = \tilde{K}[\mathbf{R} \mid \mathbf{t}]$  dove  $[\mathbf{R} \mid \mathbf{t}]$  è la rotazione e la traslazione relativa fra le due videocamere e  $\tilde{K}$  dipende dai parametri intrinseci delle due videocamere. Sfruttando la conoscenza dei parametri intrinseci delle videocamere si può eliminare  $\tilde{K}$  in modo che  $\mathcal{P}'$  sia composta da una rotazione pura ed una traslazione. Per ridurre la distorsione si possono scegliere i parametri  $(\mathbf{v}, v_4)$  in modo che  $\mathbf{R}$  rappresenti una rotazione con angolo più piccolo possibile.

Sia  $\mathbf{K}$  la matrice dei parametri intrinseci della prima videocamera e  $\mathbf{K}'$  della seconda, posso “normalizzare” i punti sul piano immagine come

$$\begin{aligned} \hat{\mathbf{x}} &= \mathbf{K}^{-1} \mathbf{x} \\ \hat{\mathbf{x}}' &= \mathbf{K}'^{-1} \mathbf{x}'. \end{aligned} \quad (2.19)$$

Data la corrispondenza di punti  $\mathbf{x} \leftrightarrow \mathbf{x}'$  ottengo la corrispondenza  $\hat{\mathbf{x}} \leftrightarrow \hat{\mathbf{x}}'$  dalla quale risalgo alla stima della matrice fondamentale  $\hat{\mathbf{F}}$ .

Le corrispondenti matrici di calibrazione in forma canonica sono

$$\begin{aligned} \hat{\mathcal{P}} &= [\mathbf{I} \mid \mathbf{0}] & \hat{\mathcal{P}}' &= [\mathbf{R} \mid \hat{e}'] \\ \mathbf{R} &= v_4 [\hat{e}']_{\times} \hat{\mathbf{F}} + \hat{e}' \mathbf{v}^T. \end{aligned} \quad (2.20)$$

Per ridurre la distorsione voglio  $\mathbf{R} = \mathbf{I}$ , ottenendo un sistema di 9 equazioni nelle quattro incognite  $(\mathbf{v}, v_4)$  che può essere “risolto” utilizzando la scomposizione ai valori singolari (appendice B.2).

### 2.3.3 Spazio di Proiezione, Metrico ed Euclideo

Ottenute le matrici  $(\hat{\mathcal{P}}, \hat{\mathcal{P}}')$  si può ottenere una ricostruzione 3D utilizzando le relazioni

$$\begin{aligned} \hat{\mathbf{x}} &\simeq \hat{\mathcal{P}} \mathbf{X}_{\mathbf{P}} \\ \hat{\mathbf{x}}' &\simeq \hat{\mathcal{P}}' \mathbf{X}_{\mathbf{P}}. \end{aligned} \quad (2.21)$$



Ambiguità	GdL	Trasformazione	Invarianti
Proiezione	15	$T_P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix}$	cross-ratio
Affine	12	$T_A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$	Distanza relativa lungo una direzione; parallelismi; piano ad infinito.
Metrica	7	$T_M = \begin{bmatrix} \sigma r_{11} & \sigma r_{12} & \sigma r_{13} & t_x \\ \sigma r_{21} & \sigma r_{22} & \sigma r_{23} & t_y \\ \sigma r_{31} & \sigma r_{32} & \sigma r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$	Distanze relative; Angoli; absolute conic.
Euclidea	6	$T_E = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$	Distanze assolute.

Figura 2.7: Spazi di ricostruzione e ambiguita.

La ricostruzione ottenuta con le relazioni 2.21 è affetta da una distorsione di proiezione, che è stata ridotta ma non eliminata. In letteratura esistono molti metodi che usano una rettificazione stratificata. Tale rettificazione si basa sul trasformare lo spazio di proiezione in uno affine, utilizzando il piano ad infinito, e da quello affine allo spazio metrico utilizzando la conoscenza dei parametri intrinseci. Le caratteristiche dei spazi citati sono riportate nella tabella 2.7.

Il metodo che abbiamo adottato permette il passaggio diretto dallo spazio di proiezione a quello metrico.

### 2.3.4 Rettificazione Metrica

Sia data la corrispondenza di punti  $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$  e la matrice fondamentale  $\hat{\mathbf{F}}$  legate in modo univoco dalla relazione epipolare

$$\hat{\mathbf{x}}_i'^T \hat{\mathbf{F}} \hat{\mathbf{x}}_i = 0, \quad (2.22)$$

per ogni  $i$ , eccetto che per gli epipoli.

Come dimostrato in [3] Siano  $[\hat{\mathcal{P}}_1, \hat{\mathcal{P}}'_1, \{\mathbf{X}_{1i}\}]$  e  $[\hat{\mathcal{P}}_2, \hat{\mathcal{P}}'_2, \{\mathbf{X}_{2i}\}]$  due ricostruzioni

ottenute dalle corrispondenze  $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ , allora esisterà una matrice non singolare  $\mathbf{H}$  per cui  $\hat{\mathcal{P}}_2 = \hat{\mathcal{P}}_1 \mathbf{H}$ ,  $\hat{\mathcal{P}}'_2 = \hat{\mathcal{P}}'_1 \mathbf{H}$  e  $\mathbf{X}_{2i} = \mathbf{H}^{-1} \mathbf{X}_{1i}$ .

Si avrà una ricostruzione metrica se il piano ad infinito ( $\pi_\infty$ ) e l'IAC (*Image of the Absolute Conic*  $\Omega$ ) della ricostruzione coincideranno con quelli reali (figura 2.8).

È possibile dimostrare [3] che data una ricostruzione di proiezione ottenuta con matrici di calibrazione in forma canonica ( $\hat{\mathcal{P}}, \hat{\mathcal{P}}'$ ), la matrice di trasformazione  $\mathbf{H}_{\mathbf{PM}}$ , che mi permette il passaggio dallo spazio di proiezione a quello metrico, è data da

$$\mathbf{H}_{\mathbf{PM}} = \begin{bmatrix} \hat{\mathbf{K}} & \mathbf{0} \\ \mathbf{p}^T \hat{\mathbf{K}} & 1 \end{bmatrix}.$$

Dove  $\hat{\mathbf{K}}$  è la matrice dei parametri intrinseci della prima videocamera e  $(\mathbf{p}^T, 1)$  sono le coordinate del piano ad infinito rispetto alla ricostruzione di proiezione.

Dato che le matrici di calibrazione ( $\hat{\mathcal{P}}, \hat{\mathcal{P}}'$ ) sono state normalizzate rispetto ai parametri intrinseci si ha che  $\hat{\mathbf{K}} = \mathbf{I}$ , da cui

$$\mathbf{H}_{\mathbf{PM}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{p}^T & 1 \end{bmatrix}. \quad (2.23)$$

Per ottenere la matrice di trasformazione  $\mathbf{H}_{\mathbf{PM}}$  abbiamo utilizzato le relazioni della Absolute Dual Quadric.

L' *Absolute Dual Quadric* ( $\mathbf{Q}_\infty^*$ ), è una quadrica duale degenera rappresen-

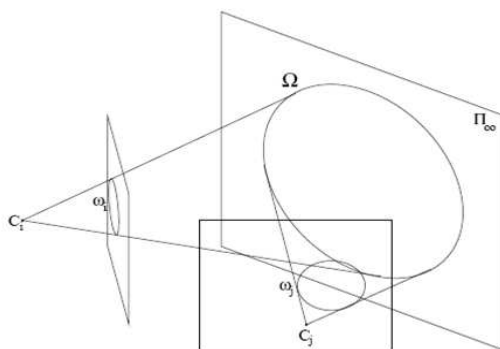


Figura 2.8: Absolute Dual Quadric.

tata da una matrice omogenea 4 x 4 con rango 3. Geometricamente  $\mathbf{Q}_\infty^*$  è il piano tangente a  $\Omega$ . La sua importanza sta nel codificare sia  $\Omega$  che  $\pi_\infty$ , in

particolare il piano ad infinito è il vettore nullo di  $\mathbf{Q}_\infty^*$  ( $\mathbf{Q}_\infty^* \pi_\infty = 0$ ), e la sua proiezione sul piano immagine è legata dalla relazione algebrica

$$\omega_i^* = \mathcal{P}_i \mathbf{Q}_\infty^* \mathcal{P}_i^T. \quad (2.24)$$

Dove  $\omega_i^*$  è il DIAC (*Dual Image of the Absolute Conic*) ed è dato da  $\omega_i^* = \mathbf{K}\mathbf{K}^T$ . Un'importante proprietà di  $\mathbf{Q}_\infty^*$  è che in un frame Euclideo assume la forma  $\mathbf{Q}_\infty^* = \tilde{\mathbf{I}}$ , dove

$$\tilde{\mathbf{I}} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}.$$

Per quanto detto nel paragrafo 2.3.3 in un frame Metrico la relazione diventa  $\mathbf{Q}_\infty^* \simeq \tilde{\mathbf{I}}$ .

La proiezione di  $\mathbf{Q}_\infty^*$  sarà data da

$$\mathbf{Q}_\infty^* \mathbf{H}_{\text{PA}} \simeq \tilde{\mathbf{I}} \mathbf{H}_{\text{PA}}^T. \quad (2.25)$$

La calibrazione con  $\mathbf{Q}_\infty^*$  si basa sul trasferire i vincoli su  $\omega^*$  in vincoli su  $\mathbf{Q}_\infty^*$  attraverso le matrici di calibrazione usando l'equazione 2.24. Quindi, stimata la  $\mathbf{Q}_\infty^*$  decomporla secondo la relazione 2.25 per ottenere la matrice di rettificazione metrica  $\mathbf{H}_{\text{PM}}$ .

Riconsiderando le matrici di calibrazione in forma canonica con distorsione ridotta ( $\hat{\mathcal{P}}, \hat{\mathcal{P}}'$ ) avremo

$$\omega_i^* = \hat{\mathbf{K}} \hat{\mathbf{K}}^T = \mathbf{I},$$

e considerando le equazioni 2.23 e 2.25 avremo  $\mathbf{Q}_\infty^*$  nella forma

$$\mathbf{Q}_\infty^* = \begin{bmatrix} \mathbf{I} & \mathbf{p} \\ \mathbf{p}^T & a \end{bmatrix}. \quad (2.26)$$

Per mantenere il vincolo sul rango di  $\mathbf{Q}_\infty^*$  deve valere

$$a = \mathbf{p}^T \mathbf{p}, \quad (2.27)$$

così da avere il determinante nullo.

Trasferendo i vincoli su  $\omega^*$  in vincoli su  $\mathbf{Q}_\infty^*$  attraverso le matrici di calibrazione usando la relazione 2.24 si ottiene

$$\hat{\mathcal{P}} \mathbf{Q}_\infty^* \hat{\mathcal{P}}^T = \mathbf{I}, \quad (2.28)$$

$$\hat{\mathcal{P}}' \mathbf{Q}_\infty^* \hat{\mathcal{P}}'^T = \mathbf{I}. \quad (2.29)$$

Data la forma canonica di  $\hat{\mathcal{P}} = [\mathbf{I} \mid \mathbf{0}]$  la 2.28 è sempre verificata e quindi non contribuisce nel fornire vincoli per il calcolo della  $\mathbf{Q}_\infty^*$ . Questo avviene

chiaramente perché tutta la distorsione è stata mappata in  $\hat{\mathcal{P}}'$ .  
 Portando la 2.29 nella forma  $\mathbf{A}\mathbf{p} = \mathbf{I}$  con  $\mathbf{A} 9 \times 3$  trovo la stima di  $\mathbf{p} = \mathbf{A}^\dagger$ .  
 Ottengo la stima della  $\mathbf{Q}_\infty^*$  usando l'equazione 2.26 ed il vincolo 2.27.

La stima ottenuta viene utilizzata come punto di partenza per un algoritmo iterativo che minimizza la funzione di costo

$$\|\mathbf{I} - \hat{\mathcal{P}}' \mathbf{Q}_\infty^* \hat{\mathcal{P}}'^T\|_F^2.$$

Dove la norma usata è quella di Frobenius (appendice B.4) e sia  $\hat{\mathcal{P}}' \mathbf{Q}_\infty^* \hat{\mathcal{P}}'^T$  che  $\mathbf{I}$  vengono normalizzate per avere norma di Frobenius unitaria in modo da eliminare il fattore di scala.

Ottenuta la stima ottimale della  $\mathbf{p}$  usando la formula 2.23 ottengo la rettificazione metrica  $\mathbf{H}_{\mathbf{PM}}$ .

Siano  $\mathbf{X}_{\mathbf{pi}}$  i punti nello spazio 3D di proiezione ottenuti triangolando le corrispondenze  $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$  i corrispondenti punti rettificati in uno spazio 3D metrico sono dati da

$$\hat{\mathbf{X}}_{Mi} = \mathbf{H}_{\mathbf{PM}}^{-1} \hat{\mathbf{X}}_{\mathbf{pi}}. \quad (2.30)$$

Una volta ottenuta la matrice di rettificazione metrica l'utilizzo della forma canonica non è più necessaria; inoltre l'applicazione della corrispondenza 2.19 per la normalizzazione su tutte le corrispondenze è un *overhead* che può essere eliminato. Si ha quindi

$$\begin{aligned} \mathcal{P} &= \mathbf{K} \hat{\mathcal{P}} \\ \mathcal{P}' &= \mathbf{K}' \hat{\mathcal{P}}'. \end{aligned} \quad (2.31)$$

Dalla condizione epilolare  $\hat{\mathbf{x}}'_i \hat{\mathbf{F}} \hat{\mathbf{x}}_i$  si ottiene  $\mathbf{F} = \mathbf{K}'^{-T} \hat{\mathbf{F}} \mathbf{K}^{-1}$ .

Una volta ottenuto  $\mathbf{H}_{\mathbf{PM}}$  per eliminare il fattore di scala bisogna avere delle informazioni metriche. Ad esempio la posizione nello spazio di un punto, la dimensione di un oggetto o la distanza di due punti.

Sia  $k$  il fattore di scala stimato si ha  $\mathbf{X}_{\mathbf{E}} = k \mathbf{X}_{\mathbf{Mi}} = k \mathbf{H}_{\mathbf{PM}}^{-1} \mathbf{X}_{\mathbf{pi}}$

## 2.4 Triangolazione

Data la corrispondenza di punti nelle due immagini  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$  posso ottenere il punto 3D che le ha generate usando le proiezioni

$$\mathbf{x} \simeq \mathcal{P} \mathbf{X}_{\mathbf{p}}, \quad \mathbf{x}' \simeq \mathcal{P}' \mathbf{X}_{\mathbf{p}} \quad (2.32)$$

A causa del rumore la corrispondenza non soddisfa la condizione epipolare  $\mathbf{x}_i^T \mathbf{F} \mathbf{x}_i = 0$  per cui i due raggi di proiezione (rette passanti per il centro di proiezione ed il punto nel piano immagine) non si intersecano. Quindi una soluzione al problema di triangolazione in generale non esiste. Una buona

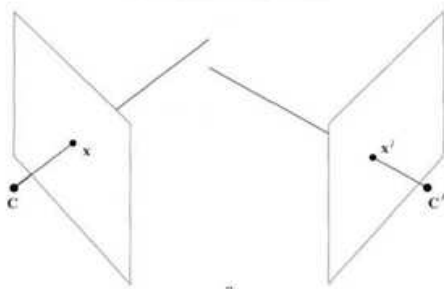


Figura 2.9: Triangolazione.

approssimazione è ottenuta con il seguente metodo lineare. Consideriamo la formula 2.32, è possibile eliminare entrambi i fattori di scala effettuando il prodotto vettoriale  $\mathbf{x} \times \mathcal{P} \mathbf{X}_p$  e  $\mathbf{x}' \times \mathcal{P}' \mathbf{X}_p$ , ottenendo così tre equazioni di cui due linearmente indipendenti; per esempio per la prima videocamera si ha

$$\begin{cases} x(\mathcal{P}^{3T} \mathbf{X}_p) - (\mathcal{P}^{1T} \mathbf{X}_p) = 0 \\ y(\mathcal{P}^{3T} \mathbf{X}_p) - (\mathcal{P}^{2T} \mathbf{X}_p) = 0 \\ x(\mathcal{P}^{2T} \mathbf{X}_p) - y(\mathcal{P}^{1T} \mathbf{X}_p) = 0. \end{cases}$$

Dove  $\mathbf{x} = [x, y, 1]^T$  e  $\mathcal{P}^{iT}$  rappresenta la  $i$ -esima riga di  $\mathcal{P}$ . Possiamo quindi ottenere il sistema lineare  $\mathbf{A} \mathbf{X}_p = 0$  dove

$$\mathbf{A} = \begin{bmatrix} x\mathcal{P}^{3T} - \mathcal{P}^{1T} \\ y\mathcal{P}^{3T} - \mathcal{P}^{2T} \\ x'\mathcal{P}'^{3T} - \mathcal{P}'^{1T} \\ y'\mathcal{P}'^{3T} - \mathcal{P}'^{2T} \end{bmatrix}. \quad (2.33)$$

Una volta ottenuto  $\mathbf{X}_p$  possiamo ottenere il punto in un frame metrico utilizzando la matrice di rettificazione metrica

$$\mathbf{X}_M = \mathbf{H}_{PM}^{-1} \mathbf{X}_p.$$

E la ricostruzione nello spazio 3D Euclideo sarà

$$\mathbf{X}_E = k \mathbf{X}_M.$$

# Capitolo 3

## C-Space

In questo capitolo verrà introdotta brevemente la teoria sui *Manifold*, per poi prendere in esame il *Manifold C-Space*. Inizialmente vedremo cosa rappresenta il *C-Space*, per poi trattare la sua costruzione per un manipolatore multi-link.

### 3.1 Manifolds

Un *Manifold* è uno spazio matematico astratto localmente Euclideo, nel senso che attorno ad ogni punto ci sono dei vicini che topologicamente sono come una sfera unitaria in  $\mathbb{R}^N$ . Per illustrare tale idea consideriamo la visione della terra che si aveva in passato. Fino al XV secolo si credeva che la terra fosse piatta, contrariamente alla provata visione sferica che abbiamo oggi. Tale convinzione era generata dal fatto che, nella piccola scala con cui vediamo (rispetto alle dimensioni della Terra), la Terra appare effettivamente piatta (figura 3.1). In generale ogni oggetto visto con una scala relativamente

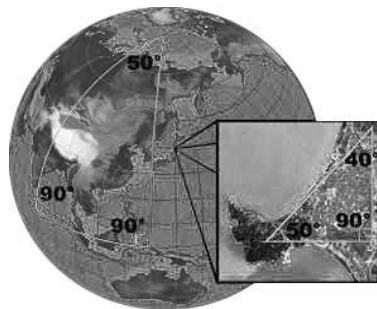


Figura 3.1: Esempio di *Manifold*

piccola può essere considerato piatto, tale visione è un *Manifold* che mi rap-

presenta parte dell'oggetto, tale rappresentazione è valida finché si mantiene il fattore di scala (es. una linea che congiunge due paesi vista dalla terra resta una linea, vista dallo spazio diventa una curva).

Per rappresentare il globo terrestre vengono usate delle mappe piatte, riunite in atlanti. Similmente, un *Manifold* può essere descritto da mappe matematiche, chiamate, *mappe delle coordinate*, e riunite in un atlante matematico.

Una *mappe delle coordinate*, o più semplicemente *mappa*, di un *Manifold* è una mappa invertibile fra un sottoinsieme del *Manifold* ed un semplice spazio in modo che entrambi, la mappa e la sua inversa, mantengano la struttura desiderata.

Nel nostro caso siamo interessati a mantenere la struttura topologica, tale struttura è preservata da un *Omeomorfismo*.

Una trattazione matematica dei *Manifold* esula dagli scopi di questa tesi è può essere trovata in [1].

### 3.2 $\mathcal{C}$ -Space, $\mathcal{C}$ -Free e $\mathcal{C}$ -Obstacle

Consideriamo un oggetto rigido  $\mathcal{A}$  che si muove in uno spazio  $\mathbb{R}^2$ , per esempio su un piano  $x$ - $y$ . La configurazione attuale dell'oggetto dipende da tre parametri, le posizioni rispetto a  $x$  e ad  $y$ , e la rotazione attorno all'asse  $z$ . Lo spazio delle configurazioni di  $\mathcal{A}$  è dato dallo spazio  $\mathcal{C}$  ( $\mathcal{C}$ -Space) di tutte le possibili configurazioni di  $\mathcal{A}$ , sarà quindi di dimensione  $\mathbb{R}^3$ .

Lo spazio delle configurazioni  $\mathcal{C}$  di un oggetto rigido  $\mathcal{A}$  che si muove (liberamente) in uno spazio di lavoro di dimensioni  $\mathbb{R}^N$  sarà un *Manifold* di dimensioni

$$m = \frac{1}{2} N ( N + 1 ). \quad (3.1)$$

Da notare che  $m$  è pari ai *gradi di libertà* dell'oggetto  $\mathcal{A}$ .

Consideriamo la topologia un generico spazio delle configurazioni  $\mathcal{C}$  di dimensione  $\mathbb{R}^N$  indotto dalla topologia di uno spazio metrico Euclideo. Essendo  $\mathcal{C}$  un *Manifold* manterrà le proprietà metriche dello spazio Euclideo di partenza, in particolare le distanze. Cioè la distanza di due configurazioni  $q$  e  $q'$  tenderà a zero quando le regioni  $\mathcal{A}(q)$  e  $\mathcal{A}(q')$  tenderanno a sovrapporsi. Tale proprietà sarà determinante per i metodi che vedremo in seguito.

Consideriamo un generico ostacolo fisso  $\mathcal{B}$ , all'interno dello spazio di lavoro di  $\mathcal{A}$ , questo mappa in  $\mathcal{C}$  una regione

$$\mathcal{CB} = \left\{ q \in \mathcal{C} \mid \mathcal{A}(q) \cap \mathcal{B} \neq \emptyset \right\}. \quad (3.2)$$

la regione  $\mathcal{CB}$  è chiamata  $\mathcal{C}$ -Obstacle, e rappresenta la regione dello spazio delle configurazioni  $\mathcal{C}$  in cui  $\mathcal{A}$  e  $\mathcal{B}$  collidono. Dati  $K$  ostacoli  $\mathcal{B}_i$  avrò un  $\mathcal{C}$ -Obstacle dato dall'unione dei  $\mathcal{CB}_i$ .

$$\mathcal{C}\text{-Obstacle} = \bigcup_{i=1}^K \mathcal{CB}_i. \quad (3.3)$$

Il  $\mathcal{C}$ -Free è la regione dello spazio delle configurazioni in cui  $\mathcal{A}$  non collide con alcun ostacolo:

$$\mathcal{C}\text{-Free} = \left\{ q \in \mathcal{C} \mid \mathcal{A}(q) \cap \bigcup_{i=1}^K \mathcal{CB}_i = \emptyset \right\}. \quad (3.4)$$

### 3.3 Costruzione $\mathcal{C}$ -Obstacle per un manipolatore multi-link

Consideriamo ora il  $\mathcal{C}$ -Space associato al *SoftArm*. Secondo l'equazione 3.1, essendo il manipolatore composto da 5 link il  $\mathcal{C}$ -Space associato avrà dimensione  $5_{\text{link}} \times 6_{\text{dim configurazioni}} = 30$ . Dobbiamo però tener conto che 2 link sono completamente vincolati (spalla e gomito) mentre gli altri 3 possono solo ruotare attorno al giunto, quindi  $30 - 2 \times 6 - 3 \times 5 = 3$ ; Non sorprende che la dimensione del  $\mathcal{C}$ -Space è uguale ai gradi di libertà del manipolatore.

Consideriamo un ostacolo  $\mathcal{B}$  nello spazio tridimensionale, rappresentato come un poliedro convesso, ed anche ogni link ( $\mathcal{A}_i$ ) del manipolatore come un poliedro convesso (es. parallelepipedo). Il  $\mathcal{C}$ -Obstacle corrispondente all'ostacolo  $\mathcal{B}$  è in generale una regione di  $\mathcal{C}$  a sei dimensioni per ogni link, ogni regione a sei dimensioni è limitata da cinque superfici in  $\mathcal{C}$  generata da cinque vincoli.

I vincoli in  $\mathcal{C}$  sono dovuti a tre diversi tipi di contatti fra le superfici di  $\mathcal{A}_i$  e  $\mathcal{B}$ . Mantenendo la simbologia del Latombe [1] definiamo tre tipi di contatti:

- **tipo A**, contatto fra una faccia di  $\mathcal{A}$  e un vertice di  $\mathcal{B}$  (figura 3.2(A));
- **tipo B**, contatto fra un vertice di  $\mathcal{A}$  e una faccia di  $\mathcal{B}$  (figura 3.2(B));
- **tipo C**, contatto fra un bordo di  $\mathcal{A}$  e un bordo di  $\mathcal{B}$  (figura 3.2(C));

La regione nello spazio  $\mathcal{C}$  sarà data dalle intersezioni delle superfici data dai tre tipi di contatti.

Come abbiamo già asserito, il  $\mathcal{C}$ -Space del *SoftArm* ha dimensione tre. Da



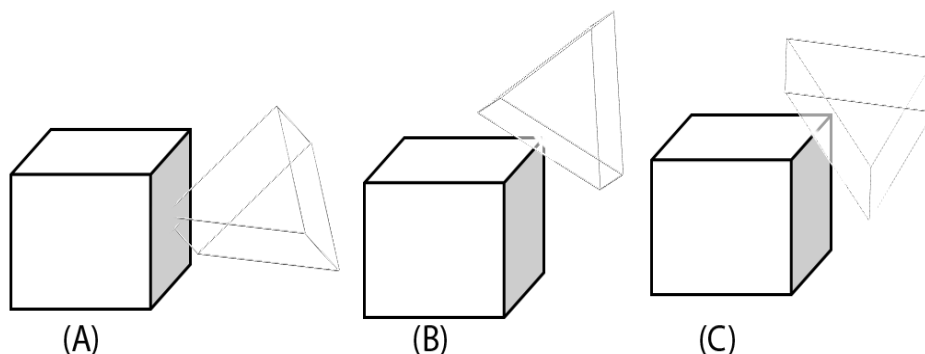


Figura 3.2: Tipi di contatto.

un punto di vista matematico possiamo considerare l'iperspazio  $\mathcal{C}$  di dimensione 30 ( $6 \times 5$  link) ed in esso un  $\mathcal{CB}$  dovuto all'intersezione delle superfici prodotta ai contatti di tipo A, B e C. Andando ora ad imporre anche i vincoli cinematici del *SoftArm* otterremo una regione che giace interamente in un iperpiano di dimensione 3.

Chiaramente conviene tener conto dal principio dei vincoli cinematici, e lavorare su un  $\mathcal{C}$ -Space di dimensione 3.

Da un punto di vista pratico basta verificare le configurazioni del manipolatore per cui, almeno uno dei parallelepipedi che rappresentano i link, collide con il poliedro rappresentante l'ostacolo.

Si ha collisione quando almeno una delle condizioni di contatto, di tipo A, B o C, è verificata. Tali problematiche verranno affrontati in maniera approfondita nel paragrafo 4.5.1.4.

# Capitolo 4

## Server di Visione

In questo capitolo verrà illustrato il *server di visione* da noi sviluppato. Il server di visione è quel modulo del sistema che si occupa della manipolazione delle immagini per ottenere le informazioni che verranno in seguito utilizzate dagli altri moduli del sistema.

Nel nostro sistema il server di visione avrà il compito di manipolare la coppia di immagini provenienti dalle videocamere per ottenere una ricostruzione del *C-Space* (capitolo 3). A tale scopo verrà illustrato un metodo per isolare nell'immagine solo gli ostacoli (paragrafo 4.2), Poi verranno presentati due diversi metodi per la costruzione del *C-Space* (paragrafo 4.5).

In figura 4.1 sono raffigurate schematicamente le varie fasi che compongono il nostro server di visione, questo schema verrà utilizzato come riferimento per l'intero capitolo.

Gli algoritmi che vedremo nei prossimi paragrafi sono sviluppati utilizzando le librerie OpenCV<sup>©</sup> di Intel ed ERSP<sup>©</sup> di Evolution Robotics.

### 4.1 Equalizzazione dell'immagine

Le tecniche che vedremo nei prossimi paragrafi sono tutte sensibili ai valori di luminosità e contrasto delle immagini catturate, tale sensibilità si è riscontrata in particolar modo nelle funzioni della libreria OpenCV<sup>©</sup>; la libreria ERSP<sup>©</sup> in teoria non è sensibile a variazioni di luminosità, ma nella versione attuale variazioni di luminosità sono ancora fonte di errori. In questo paragrafo introduciamo due metodi per l'*Enhancement* dell'immagine da noi utilizzati.

Innanzitutto la scelta da noi fatta è di utilizzare immagini monocromatiche (scala di grigi). Questa scelta è stata fatta perché gli algoritmi da noi utiliz-

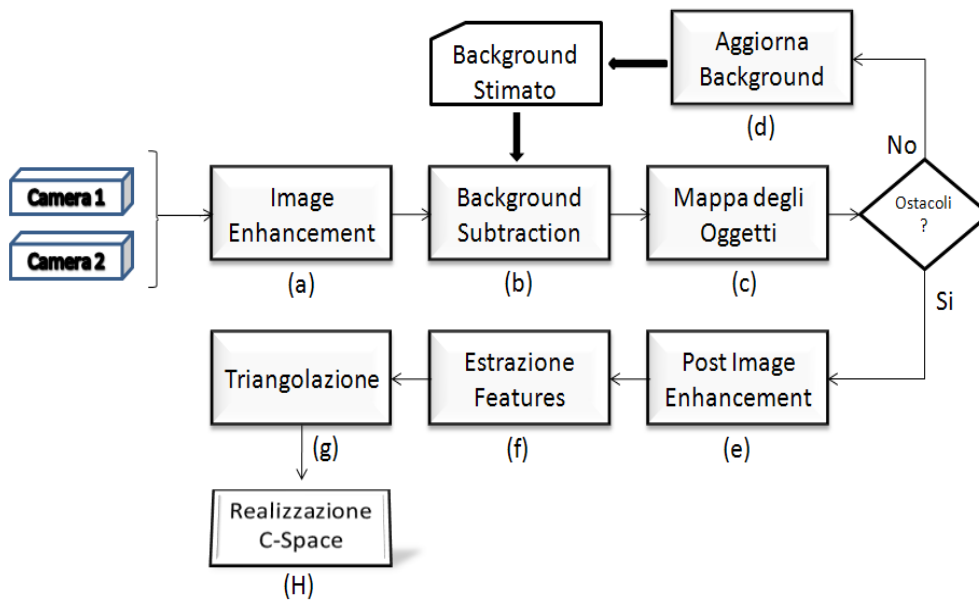


Figura 4.1: Schema di esecuzione del server di visione

zati e sviluppati possono lavorare anche su immagini monocromatiche. L'utilizzo della sola componente monocromatica rende più veloce l'esecuzione del server di visione, dovendo lavorare su un canale cromatico invece di lavorare su tre canali.

Algoritmi per l'*Image Enhancement*:

- **Equalizzazione dell'istogramma** : Uno degli strumenti più utili nell'elaborazione delle immagini è l'istogramma dei livelli di grigio contenuti in una immagine. Il calcolo dell'istogramma è molto semplice ed è quindi anche molto utilizzato per ottenere miglioramento del contrasto e quindi della visualizzazione dell'immagine.

L'istogramma è la rappresentazione grafica della quantità di pixel presenti nell'immagine per ciascun livello di grigio (figura 4.2). L'asse orizzontale X rappresenta i valori dei grigi. L'asse verticale Y rappresenta il numero di pixel per ogni livello. L'istogramma fornisce una raffigurazione sintetica delle caratteristiche di luminosità dell'immagine, trascurando però ogni informazione relativa alla posizione dei singoli pixel.

Rendendo più uniforme i livelli dell'istogramma si ottiene un'immagine con un con caratteristiche cromatiche migliori (figura 4.2).

- **Regolazione della Luminosità e del Contrasto** Al contrario della tecnica precedente, in questo caso le variazioni sono di tipo assoluto e

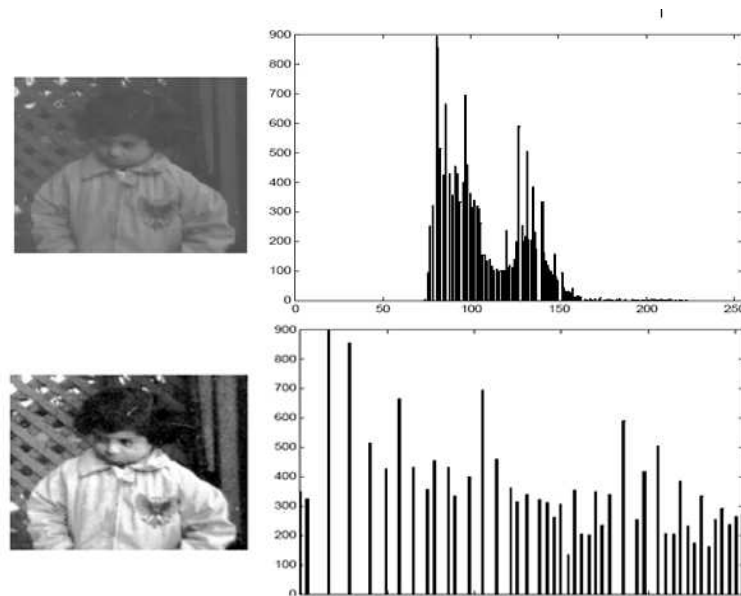


Figura 4.2: Equalizzazione dell'istogramma

non relativo. Abbiamo utilizzato un algoritmo sviluppato da *Werner D. Streidt* (<http://visca.com/ffactory/archives/5-99/msg00021.html>) in cui, date le variazioni di luminosità e contrasto voluti si realizza una tabella di corrispondenza (luminosità pixel originale  $\rightarrow$  luminosità risultante), con cui l'immagine viene modificata.

Un esempio dell'uso di tale tecnica è riportato in figura 4.3.



Figura 4.3: Regolazione luminosità e contrasto

Nel nostro server di visione entrambe le tecniche per l'*enancement* dell'immagine esposti possono essere attivate o disattivate ed eseguite nell'ordine desiderato (es. prima si aumenta la luminosità e poi equalizzazione dell'istogramma), a seconda delle caratteristiche dell'ambiente in cui sono svolte le prove.

Nello schema di riferimento (immagine 4.1) troviamo l'*Image Enancement*

al blocco (a), e poi lo troviamo anche nel blocco (e) con il prefisso *Post*; Nel blocco (e) le funzioni svolte sono le stesse, però la sua attivazione viene decisa dal server stesso durante l'esecuzione nel caso le caratteristiche delle immagini diventino scadenti. La necessità di avere una ripetizione delle funzioni che vengono già eseguite al blocco (a) diventeranno chiare dopo il paragrafo 4.2.

## 4.2 Background Subtraction

Questa è una tecnica che permette di isolare l'immagine di oggetti che “entrano” nell'immagine di partenza. Data una stima del *background*<sup>1</sup>(figura 4.4(A)) l'immagine di un oggetto che prima non era presente (per noi è un possibile ostacolo) è data dai pixel che verificano la relazione

$$|frame(x, y) - background(x, y)| < Th, \quad (4.1)$$

dove *Th* è un valore di soglia.

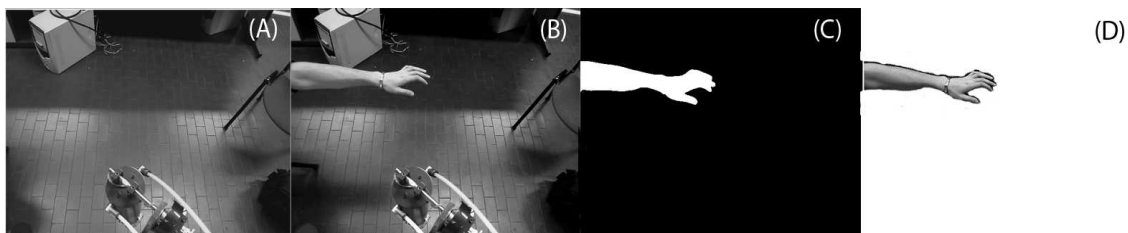


Figura 4.4: Sottrazione del *background*.

La stima del *background* viene effettuata in due punti differenti del server di visione. Nella fase di *set-up* vengono catturati *N* frame del *background*(chiaramente in questa fase lo spazio di lavoro del manipolatore deve essere libero) quindi viene fatta la media

$$background(x, y) = \frac{1}{N} \sum_{i=1}^N frame_i(x, y). \quad (4.2)$$

Durante l'esecuzione del server di visione la stima del *background* viene aggiornata quando si riscontra la non presenza di oggetti nello spazio di lavoro. La stima viene aggiornata dando un peso maggiore al nuovo frame secondo la formula 4.3.

$$background_{i+1} = (1 - \alpha)background_i + \alpha frame_i. \quad (4.3)$$

<sup>1</sup>Il *background* per noi è l'immagine vista dalle videocamere quando nessun oggetto si trova nello spazio di lavoro

I passi con cui otteniamo il *backgroundsubtraction* data la stima del *background* del nuovo frame sono:

- si realizza la sottrazione aritmetica della luminosità di ogni pixel del frame con il corrispettivo del *background*;
- si applica un filtro con effetto *smooth* per rendere più omogeneo il risultato del passo precedente;
- si realizza una maschera con i pixel che hanno dopo i passaggi precedenti una luminosità superiore alla soglia (figura 4.4(C));
- si applica la maschera al frame di partenza ottenendo così un'immagine con i soli oggetti isolati (figura 4.4(D)).

In (figura 4.4) si può vedere il risultato del *background Subtraction*.

Abbiamo sviluppato anche altre tecniche, in cui oltre alla media del *background* si stima anche la varianza, per poi usarla come valore di soglia. A dispetto della sua semplicità il metodo da noi usato si è rivelato più affidabile e più veloce.

Il *Background Subtraction* viene realizzato solo sul frame proveniente dalla camera principale<sup>2</sup>, farlo su entrambe sarebbe superfluo, dato che non utilizziamo tale informazioni.

Nello schema di riferimento (figura 4.1) il *Background Subtraction* viene eseguito nel blocco (b), mentre l'aggiornamento della stima del *background* durante l'esecuzione del server di visione è eseguita nel blocco (d).

Si deve considerare che il manipolatore si muove nello spazio di lavoro. La sottrazione del *background* considera lo stesso manipolatore come un ostacolo. Conoscendo la configurazione attuale possiamo calcolare la regione del piano immagine in cui viene proiettato il manipolatore applicando le matrici della cinematica e la matrice di proiezione della videocamera. Il manipolatore viene considerato parte del *background* sottraendo all'immagine la regione occupata dal manipolatore

## 4.3 Area oggetti e filtraggio

Ottenuta un'immagine in cui siano presenti solo gli oggetti che non facevano parte del *background* (più chiaramente il rumore dovuto a variazioni di luce) vogliamo:

---

<sup>2</sup>Nel nostro sistema la camera principale (quella con matrice di proiezione  $\mathcal{P} = [\mathbf{I} \mid \mathbf{0}]$  è la camera di sinistra.)

- filtrare il rumore;
- valutare il numero di oggetti;
- creare una mappa in cui per ogni pixel dell'immagine nella quale indico se un dato pixel fa parte di un'oggetto e se sì di quale.

Consideriamo l'immagine di un oggetto formato da un'insieme di pixel adiacenti non vuoti<sup>3</sup>. Possiamo costruire una struttura dati in cui per ogni oggetto memorizzo i pixel che lo compongono. Data questa struttura posso calcolare l'area di ogni oggetto (numero di pixel che lo compongono). Posso quindi filtrare i risultati andando ad eliminare gli oggetti che hanno un'area minore ad una soglia.

Dopo aver effettuato i passi precedenti si realizza una mappa, della stessa dimensione dell'immagine, in cui ogni elemento sarà: 0 se il pixel non fa parte di un oggetto;  $i$  se il pixel fa parte dell'oggetto  $i$ -esimo.

In figura 4.5 è mostrato il risultato della ricerca del contorno che poi viene utilizzata per la realizzazione della mappa.

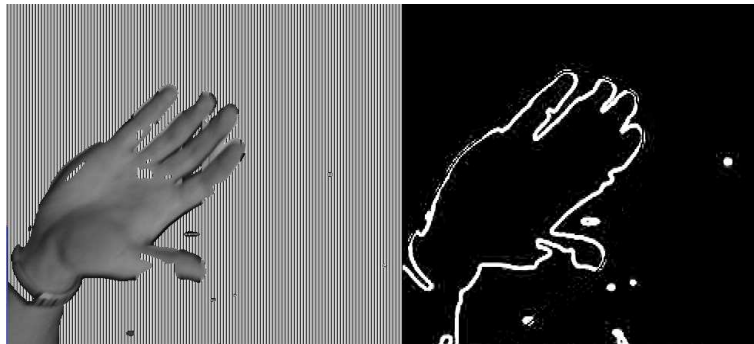


Figura 4.5: Contorno dell'oggetto

Ricapitolando i passaggi per ottenere la *mappa degli oggetti* sono:

1. si trovano i contorni;
2. si associa ad ogni oggetto i pixel che lo compongono;
3. si filtra gli oggetti che hanno l'area inferiore ad una certa soglia;
4. si crea la *mappa degli oggetti*.

Riferendoci allo schema 4.1, le operazioni illustrate in questo paragrafo sono svolte nel blocco (c).

<sup>3</sup>Con luminosità maggiore della soglia durante il *Background Subtraction*.

## 4.4 Ricostruzione dei punti 3D

Per ottenere la posizione 3D di un punto tramite la visione stereo, come illustrato nel paragrafo 2.4, si deve conoscere la posizione del punto nelle due immagini. La ricerca della corrispondenza dei punti all'interno di due immagini è un argomento ampiamente trattato in letteratura. I vari metodi hanno tutti in comune come procedura di base l'estrazione dalle immagini dei pixel con alto contenuto informativo (*feature*) e la ricerca di corrispondenze fra le *feature* estratte.

### 4.4.1 Estrazione delle *feature*

Per l'estrazione e il *matching* delle *feature* abbiamo usato il metodo delle *SIFT* (*Scale-Invariant Feature Transform*) fornito dalle librerie ERSP<sup>©</sup> di *Evolution Robotics*. *SIFT* è un algoritmo pubblicato da *David Lowe* [5] nel 2004. Le *feature* estratte risultano essere invarianti alla scala dell'immagine, alla rotazione, alla traslazione e parzialmente invarianti al cambio del punto di vista e di illuminazione. I passi principali dell'algoritmo di estrazione delle *SIFT* sono i seguenti:

- **Individuazione degli estremi locali nello *scale-space*:** si cercano punti interessanti su tutte le scale e posizioni dell'immagine utilizzando una funzione *DoG* (*Difference of Gaussian*). L'approccio utilizzato è quello del filtraggio in cascata, che consente di determinare le posizioni e la scala delle *feature* candidate ad essere punti chiave e che, in un secondo momento, vengono caratterizzate con maggior dettaglio;
- **Localizzazione dei *keypoint*:** per ciascun punto candidato viene costruito un modello dettagliato per determinarne posizione e scala. I punti vengono inoltre selezionati secondo misure di stabilità;
- **Generazione delle orientazioni:** per ottenere l'invarianza rotazionale, ad ogni punto chiave (*keypoint*) vengono associate una o più orientazioni calcolate in base ai gradienti locali dell'immagine;
- **Generazione del descrittore:** a partire dai gradienti locali dell'immagine, alla scala selezionata e nell'intorno del punto chiave, viene costruito il descrittore.

Ad ogni punto individuato viene quindi associato un descrittore che consente di individuarlo in immagini diverse.

Nel nostro server di visione il *matching* viene fatto tra l'immagine della camera di sinistra a cui è stato sottratto il *background* e l'immagine della



camera di destra. Così facendo avremo la corrispondenza delle *feature* solo per gli oggetti individuati (figura 4.6).

Riferendoci allo schema 4.1 questa fase è eseguita nel blocco (f).

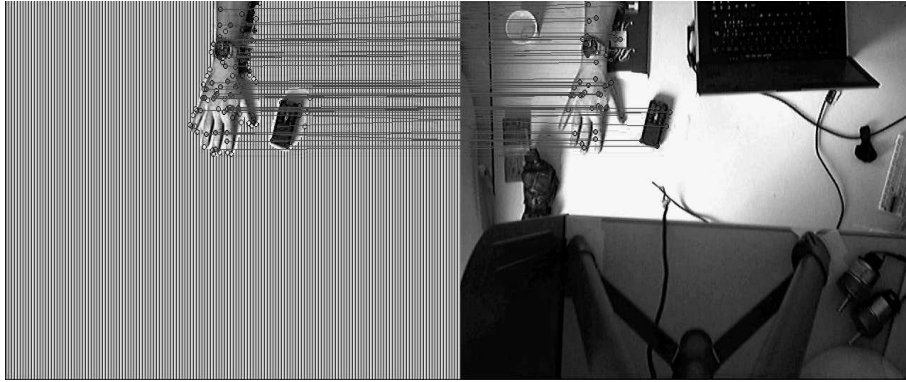


Figura 4.6: Matching delle *feature*.

#### 4.4.2 Triangolazione *feature*

Una volta ottenuta la corrispondenza delle *feature* possiamo ottenere la posizione 3D del punto tramite l'equazione 2.32.

Prima di procedere alla triangolazione possiamo ottenere un filtraggio delle *feature* trovate utilizzando la geometria epipolare. Consideriamo quanto detto nel paragrafo 2.2.1. La corrispondenza delle due *feature* è corretta se soddisfa la condizione di corrispondenza (equazione 2.13). A causa del rumore e della discretizzazione effettuata dalle camere, la condizione di corrispondenza difficilmente viene soddisfatta.

Prendiamo come riferimento la figura 4.7, se la condizione di corrispondenza

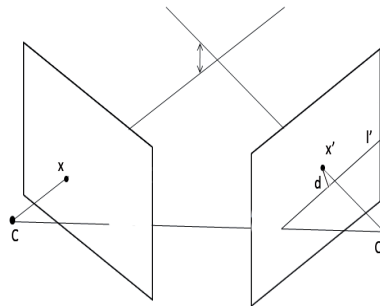


Figura 4.7: Triangolazione.

non è soddisfatta il punto  $\mathbf{x}'$  non si troverà sulla linea epipolare  $l'$ . Volendo

calcolare la distanza (minima) fra il punto  $\mathbf{x}'$  e la linea  $l'$  dovrei calcolare  $d = \mathbf{x}'^T l'$  (appendice B.1.2.3), dato che  $l' = \mathbf{F} \mathbf{x}$  avremo

$$d = \mathbf{x}'^T \mathbf{F} \mathbf{x}. \quad (4.4)$$

Quindi per ogni corrispondenza calcolo la distanza dalla linea epipolare con la formula 4.4, se questa distanza è maggiore di una certa soglia si considera la corrispondenza errata e quindi viene scartata.

Riferendoci allo schema 4.1 questa fase è eseguita nel blocco (g).

## 4.5 Creazione del $\mathcal{C}$ -Space

In questo paragrafo verranno illustrati due diversi metodi da noi sviluppati per la realizzazione del  $\mathcal{C}$ -Space. Il primo genera una ricostruzione degli ostacoli nello spazio 3D relativo al sistema di riferimento globale per poi verificare la collisione con un modello 3D del manipolatore. Il secondo metodo proietta i punti del manipolatore nel piano immagine della camera principale, e poi usando una mappa che chiameremo *mappa delle profondità* verifica le collisioni. Riferendoci allo schema 4.1 questa fase è eseguita nel blocco (H).

### 4.5.1 Metodo A

Come illustrato nel paragrafo 3.3 il  $\mathcal{C}$ -Space (Spazio delle Configurazioni) è un *Manifold* in cui ogni punto rappresenta una configurazione del manipolatore. Un ostacolo nello spazio di lavoro del manipolatore è mappato in una regione del  $\mathcal{C}$ -Space chiamato  *$\mathcal{C}$ -Obstacle*.

Nel metodo A viene prima realizzata una ricostruzione nello spazio 3D degli oggetti trovati e poi si valuta le configurazioni in cui il modello 3D dell'oggetto collide con il modello 3D del manipolatore.

#### 4.5.1.1 Contorno e Zone d'Ombra

L'algoritmo *SIFT* per il matching delle *feature* consente di trovare le corrispondenze fra le immagini catturate dalle due camere. Le corrispondenze trovate sono nella maggior parte dei casi interni all'oggetto. Per realizzare il modello 3D dell'oggetto dobbiamo conoscere anche la posizione 3D dei punti che compongono il contorno.

Per prima cosa individuiamo i punti nell'immagine che compongono il contorno degli oggetti. Considerando l'immagine della camera principale a cui è stata applicata la sottrazione del *background*; i punti che compongono il contorno dell'oggetto possono essere trovati applicando l'algoritmo di *Canny*

[7]. L'angoritmo di *Canny* trova tutti i punti dell'immagine che compongono un contorno (figura 4.5). Il numero dei punti appartenenti al contorno trovati è elevato; applicare l'algoritmo che consente di stimare la posizione 3D dei punti del contorno su tutti i punti trovati richiede un tempo computazionale elevato. Per diminuire il numero dei punti che compongono il contorno utilizziamo un algoritmo che realizza il *convex hull* contenente tutti i punti trovati (figura 4.8).

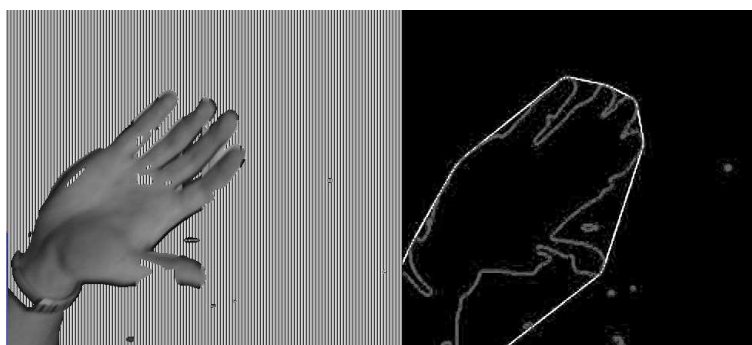


Figura 4.8: Contorno dell'oggetto, convex hull

Tutti i punti nello spazio che si trovano sullo stesso raggio di proiezione di un punto dell'immagine appartenente ad un oggetto ma con profondità<sup>4</sup> maggiore fanno parte della *zona d'ombra*. La *zona d'ombra* è quindi composta da punti occlusi alla visione da un oggetto individuato. I punti appartenenti alla *zona d'ombra* di un oggetto vengono considerati parte di esso.

#### 4.5.1.2 Costruzione Modello 3D degli Oggetti

La proiezione 3D dei punti del contorno non può essere fatta con la triangolazione perché non conosciamo i punti corrispondenti nell'immagine di destra. Ipotizziamo che la proiezione 3D di un punto del contorno giaccia sullo stesso piano dei punti 3D conosciuti a lui più vicini.

Per ogni punto del *convex hull* vengono considerati tutte le *feature* di cui conosciamo la posizione 3D presenti in un intorno. Se sono presenti nell'intorno almeno 4 *feature* costruiamo il piano nello spazio 3D che minimizza la distanza fra il piano e i punti 3D corrispondenti alle *feature* considerate (figura 4.9)(appendice B.1.3.2).

<sup>4</sup>Distanza sull'asse z rispetto al sistema di riferimento della camera principale.

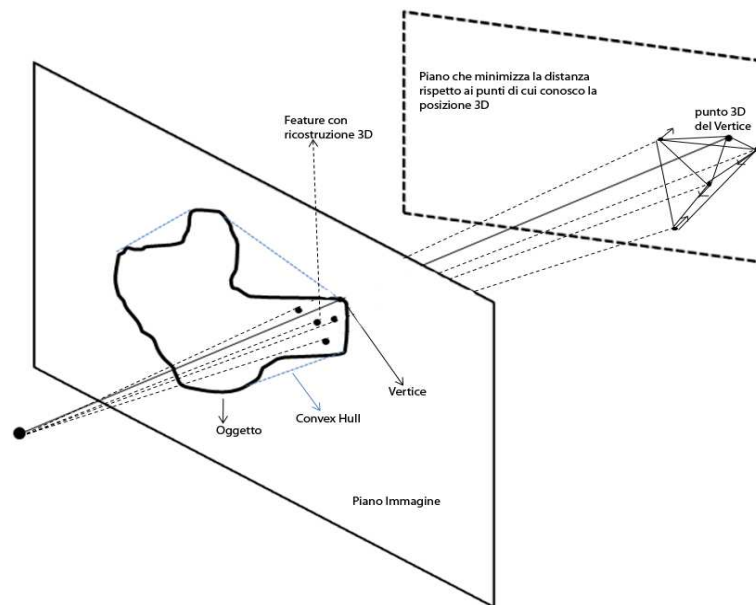


Figura 4.9: Posizione 3D dei punti del contorno.

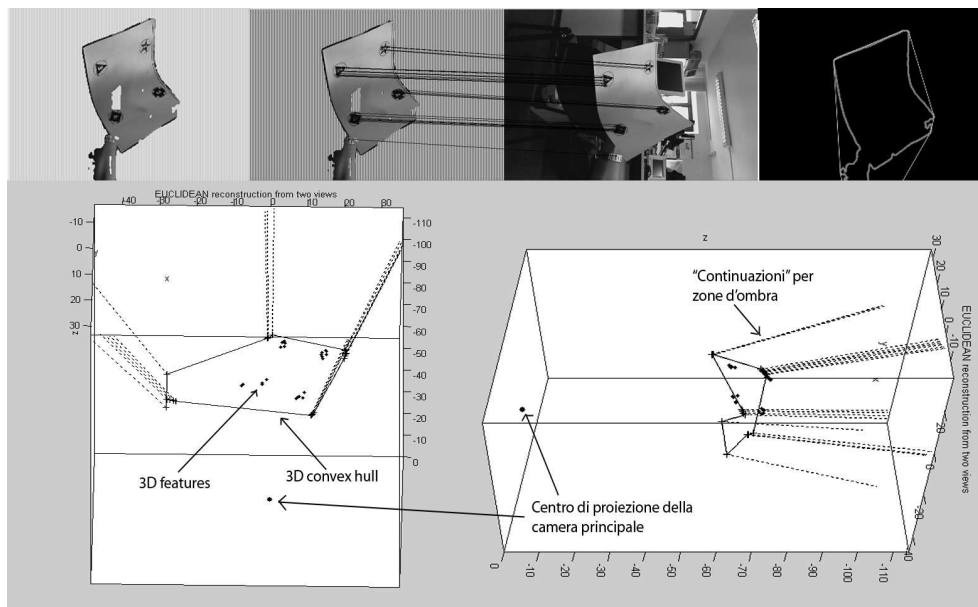


Figura 4.10: Punti ricostruiti nello spazio.

La posizione nello spazio del punto del contorno è data dall'intersezione fra il raggio di proiezione del punto del contorno ed il piano costruito (figura 4.9).

Vengono considerati almeno 4 punti perché con 3 punti l'algoritmo risulta troppo soggetto ai rumori<sup>5</sup>.

Se nell'intorno del punto del contorno sono presenti meno di quattro *feature* di cui conosciamo la posizione 3D, viene ipotizzato che la proiezione del punto del contorno sia alla stessa profondità della *feature* a lui più vicina.

Per la costruzione del modello 3D vengono considerati i punti nello spazio dati da: corrispondenza di *feature*; contorni; proiezione dei contorni che rappresentano le *zone d'ombra* (figura 4.10).

Il modello 3D dell'oggetto è dato dal poliedro convesso che inscrive i punti nello spazio citati (figura 4.11).

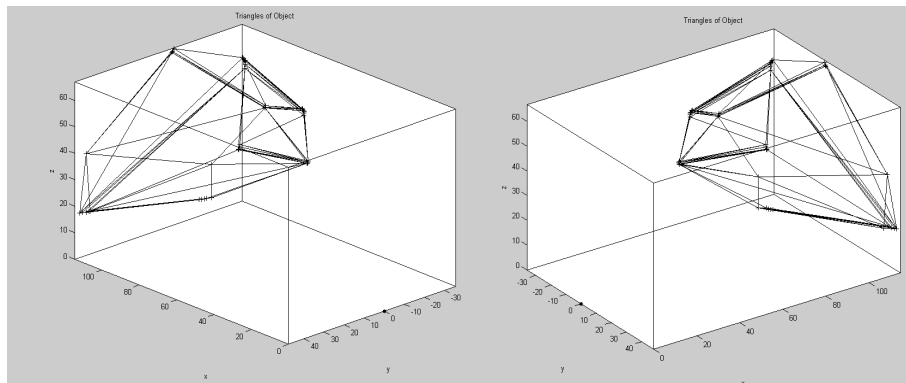


Figura 4.11: Modello 3D dell'oggetto

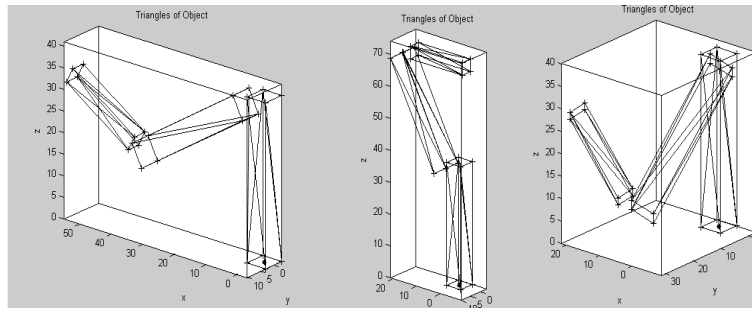
#### 4.5.1.3 Costruzione Modello 3D del Manipolatore

Ogni link del manipolatore viene rappresentato nello spazio dal parallelepipedo che lo inscrive (figura 4.12). Il parallelepipedo viene costruito a partire dalla conoscenza dei due punti estremi del link nello spazio. Data una configurazione del manipolatore i due punti estremi per ogni link vengono calcolati utilizzando la cinematica diretta del manipolatore.

#### 4.5.1.4 Calcolo del *C-Space*

Per costruire il *C-Space* dobbiamo considerare tutte le configurazioni del manipolatore. Per ogni configurazione dobbiamo verificare se il modello 3D del manipolatore collide (interseca) il modello 3D di uno degli oggetti ricostruiti. Data una configurazione, se la collisione fra il modello 3D dell'oggetto e quella

<sup>5</sup>Dovuti al non corretto rispetto della geometria epipolare delle *feature*.

Figura 4.12: Modello 3D del *SoftArm*

del manipolatore è verificata tale configurazione fa parte del  $\mathcal{C}\text{-Obstacle}$  altrimenti fa parte del  $\mathcal{C}\text{-Free}$ .

Per verificare la collisione abbiamo realizzato tre metodi:

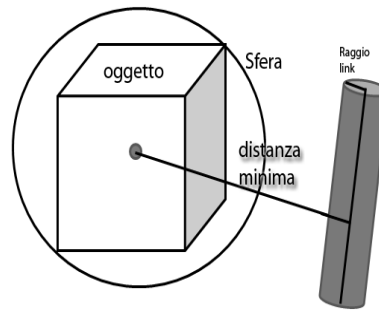


Figura 4.13: Metodo della sfera

- **Metodo della Sfera.** Condizione necessaria. Per ogni oggetto costruisco la sfera a minimo raggio che inscrive il poliedro che rappresenta l'oggetto. Se la distanza fra il link ed il centro della sfera meno il raggio del link è minore del raggio della sfera la collisione è possibile (figura 4.13). Con tale tecnica si possono eliminare velocemente tutte le configurazioni che si trovano sicuramente nel  $\mathcal{C}\text{-Free}$ .

$$d(\text{link}, C_{\text{sfera}}) - \text{raggio}_{\text{link}} \leq \text{raggio}_{\text{sfera}} \iff \text{collisione}$$

- **Metodo delle disequazioni del piano.** Condizione sufficiente ma non necessaria. Ogni link è rappresentato dagli 8 vertici del parallelepipedo che lo inscrive. Se uno di questi 8 vertici si trova all'interno del poliedro che rappresenta l'oggetto la collisione è sicura. Per verificare se il punto è interno al poliedro dobbiamo valutare ogni triangolo che compone il poliedro, consideriamo l'equazione del piano su

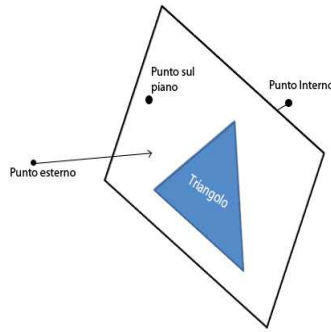


Figura 4.14: Metodo delle disequazioni del piano

cui giace il triangolo (figura 4.14). Un punto sarà interno al poliedro se verifica tutte le disequazioni. Sia  $P_j$  il punto del link che vado a considerare. Il piano su cui giace il triangolo sarà dato dall'equazione  $\pi(x, y, z) = ax + by + cz + d = 0$ .

Avremo:

$$\begin{aligned}\pi(P_j) < 0 &\Leftrightarrow \text{punto interno} \\ \pi(P_j) = 0 &\Leftrightarrow \text{punto sul piano} \\ \pi(P_j) > 0 &\Leftrightarrow \text{punto esterno}\end{aligned}$$

$$\exists j \in [\text{punti link}] : \forall i \in [\text{triangoli}] \pi_i(P_j) \leq 0 \implies \text{collisione}$$

Tale metodo non dà una condizione necessaria perché non considera le intersezioni date dai bordi e dalle facce dei link.

- **Metodo dei triangoli.** Condizione necessaria e sufficiente. Consideriamo ogni link rappresentato dai 12 (oppure 8 escludendo le basi) triangoli che compongono il parallelepipedo che lo inscrive (figura 4.12). Andando a verificare l'intersezione fra i triangoli del link e quelli del poliedro verifico con certezza se tale configurazione appartiene al  $\mathcal{C}$ -*Obstacle* o no.

$$\begin{aligned}\exists i \in [\text{triangoli link}], j \in [\text{triangoli poliedro}] : \\ \text{intersezione}(\text{triangolo}_i, \text{triangolo}_j) \iff \text{collisione}\end{aligned}$$

I tre metodi possono essere combinati per ottenere i risultati di velocità e approssimazione<sup>6</sup> voluti. Dalle prove fatte abbiamo riscontrato una buona velocità combinando il metodo della sfera con il metodo dei triangoli. Con il metodo della sfera riusciamo ad eliminare velocemente le configurazioni che sono sicuramente nel  $\mathcal{C}$ -Free, per le altre configurazioni utilizziamo il metodo dei triangoli.

Il risultato della costruzione del  $\mathcal{C}$ -Space per la figura 4.10 è mostrato in figura 4.15.

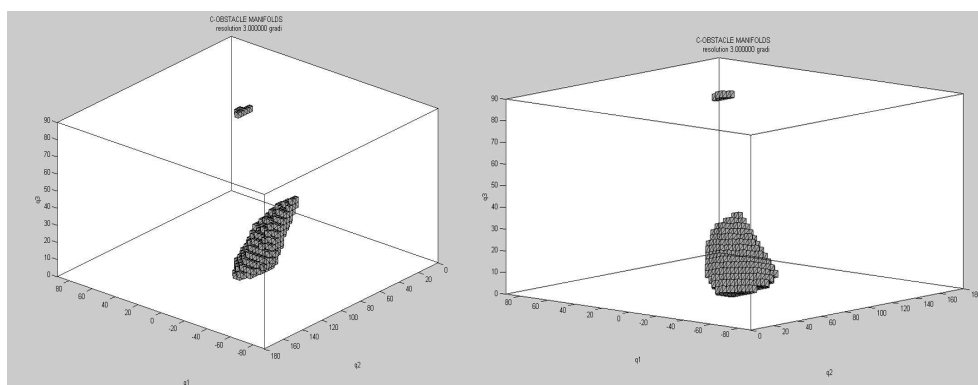


Figura 4.15:  $\mathcal{C}$ -Space: risoluzione 3 gradi.

#### 4.5.1.5 Considerazioni sul Metodo A

Prove effettuate utilizzando il metodo A hanno riscontrato come con tale metodo la stima del  $\mathcal{C}$ -Space venga effettuata correttamente, ma il tempo richiesto per il calcolo delle collisioni per ogni configurazione è eccessivo. Dei miglioramenti sono stati ottenuti precaricando in memoria le informazioni sul modello 3D del manipolatore nelle varie configurazioni. Anche con queste modifiche il tempo per il calcolo del  $\mathcal{C}$ -Space si aggira fra i 9 ed i 20 secondi. Visti i risultati incompatibili con le richieste di tempo reale abbiamo ideato un secondo metodo (che chiameremo Metodo B).

### 4.5.2 Metodo B

In questo secondo metodo viene utilizzato un procedimento alternativo da noi sviluppato. Al posto di realizzare la ricostruzione 3D dell'ostacolo realizziamo una mappa che chiameremo *mappa delle profondità*. Dal confronto fra tale *mappa delle profondità* e le proiezione sul piano immagine dei punti del manipolatore riusciamo a costruire il  $\mathcal{C}$ -Space.

<sup>6</sup>Possiamo sovrastimare o sottostimare il  $\mathcal{C}$ -Obstacle.



### 4.5.2.1 Mappa delle Profondità

La *mappa delle profondità* contiene la profondità<sup>7</sup> stimata dell'oggetto per ogni pixel dell'immagine<sup>8</sup>.

La *mappa delle profondità* viene creata partendo dalla profondità dei punti 3D ottenuti tramite triangolazione delle corrispondenze di *feature* (paragrafo 4.4.2). Consideriamo un punto  $\mathbf{x}$  dell'immagine; per quanto detto nel paragrafo 4.3 sappiamo se il punto  $\mathbf{x}$  fa parte di un oggetto. Se il punto  $\mathbf{x}$  non appartiene a nessun oggetto allora fa parte del *background* quindi consideriamo la sua profondità infinita<sup>9</sup>.

Se invece fa parte di un oggetto si stima la sua profondità come media pesata dall'inverso della distanza delle profondità dei punti di cui conosco la proiezione 3D, appartenenti allo stesso oggetto.

I passi per la costruzione della *mappa delle profondità* sono quindi:

Per ogni punto dell'immagine  $\mathbf{p}$

1. verifichiamo se appartiene ad un oggetto;
2. se non appartiene ad un oggetto profondità =  $\infty$ , prossimo punto;
3. altrimenti, consideriamo le profondità dei punti 3D ottenuti dalla triangolazione  $X_{z_i}$  delle corrispondenze di *feature*  $(\mathbf{x}_i, \mathbf{x}'_i)$  ;
4. calcoliamo la profondità come

$$\frac{\sum_{i=1}^{N_P} X_{z_i} \frac{1}{d(\mathbf{p}, \mathbf{x})^k}}{\sum_{i=1}^{N_P} \frac{1}{d(\mathbf{p}, \mathbf{x})^k}},$$

dove  $N_P$  è il numero di punti di cui conosco la posizione 3D appartenenti allo stesso oggetto;  $k$  è un parametro che caratterizza il peso della distanza sul calcolo della media.

La rappresentazione di una *mappa delle profondità* ottenuta da immagini catturate è rappresentata in figura 4.16, dove la gradazione di grigio rappresenta la profondità : più la gradazione di grigio è scura e più la profondità è maggiore.

<sup>7</sup>Coordinata  $Z$  nel sistema di riferimento della camera principale.

<sup>8</sup>Consideriamo sempre quella catturata dalla camera di sinistra su cui è stato applicato la sottrazione del *background*

<sup>9</sup>È ad una profondità che il manipolatore non può raggiungere.

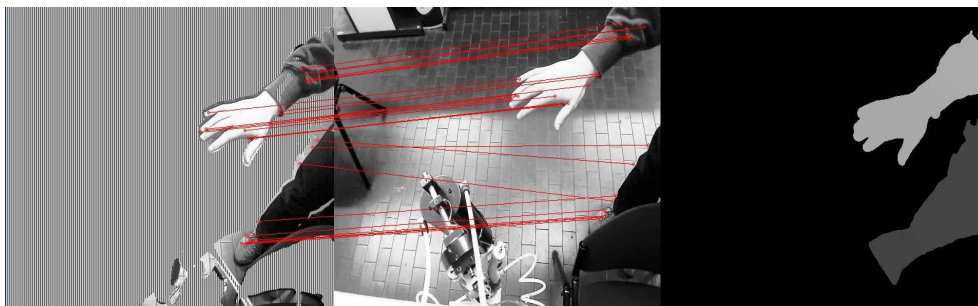


Figura 4.16: Mappa delle profondità

#### 4.5.2.2 Calcolo del $\mathcal{C}$ -Space

Data una configurazione  $\mathbf{q}$  del manipolatore, per verificare se tale configurazione appartiene al  $\mathcal{C}$ -Obstacle utilizzando la *mappa delle profondità* devo conoscere i punti nell'immagine su cui viene proiettato il manipolatore e la loro profondità.

I punti nello spazio, rispetto al riferimento globale (paragrafo 2.1.2), assunti dal manipolatore nella configurazione  $\mathbf{q}$  sono calcolati utilizzando la cinematica diretta. Per ogni link consideriamo un numero finito di punti  ${}^G\mathbf{P}_i$ <sup>10</sup>.

Utilizzando i parametri estrinseci della camera principale  $[\mathbf{R} \mid \mathbf{t}]$  (paragrafo 2.1.2) ottengo la posizione dei punti  ${}^G\mathbf{P}_i$  rispetto al sistema di riferimento della camera principale (riferimento di visione) come

$${}^V\mathbf{P}_i = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} {}^G\mathbf{P}_i. \quad (4.5)$$

La coordinata  $Z$  di  ${}^V\mathbf{P}_i$  è la profondità associata al punto.

I punti nel piano immagine vengono calcolati utilizzando i parametri intrinseci ( $\mathbf{K}$ ) della camera principale (paragrafo 2.1.1)

$$\mathbf{p}_i = [\mathbf{K} \mid \mathbf{0}] {}^V\mathbf{P}_i. \quad (4.6)$$

I punti  $\mathbf{p}_i$  sono punti nel piano immagine, ed hanno ognuno un punto corrispondente nella *mappa delle profondità*  $\hat{\mathbf{p}}_i$ <sup>11</sup> (figura 4.17).

La condizione di appartenenza al  $\mathcal{C}$ -Obstacle sarà:

$$\exists i : {}^V\mathbf{P}_{i_z} - \text{profondit\`a}(\hat{\mathbf{p}}_i) > r, r \in \mathbb{R}^+ \implies \mathbf{q} \in \mathcal{C}\text{-Obstacle} \quad (4.7)$$

Dove  $\text{profondit\`a}(\hat{\mathbf{p}}_i)$  è l'operatore che mi restituisce i valori della *mappa delle profondità* ed  $r$  è un valore di soglia che tiene conto delle varie approssimazioni<sup>12</sup> fatte.

<sup>10</sup> ${}^G P$  punto in coordinate globali  ${}^V P$  in coordinate di visione.

<sup>11</sup>Il pixel corrispondente al punto nell'immagine

<sup>12</sup>Ad esempio prendendo solo un numero finito di punti per link.

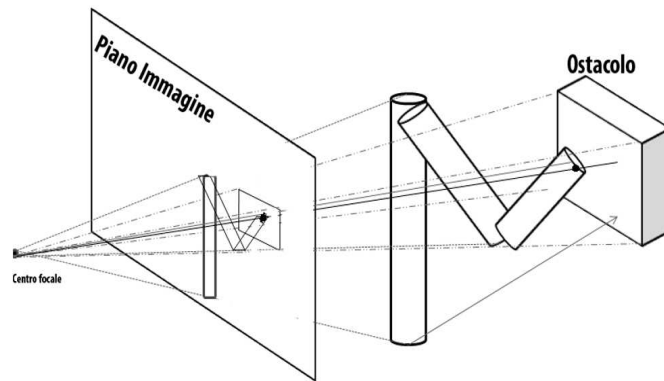


Figura 4.17: Metodo B: confronto profondità.

Un esempio di  $C$ -Space calcolato con il metodo B è riportato in figura 4.18.

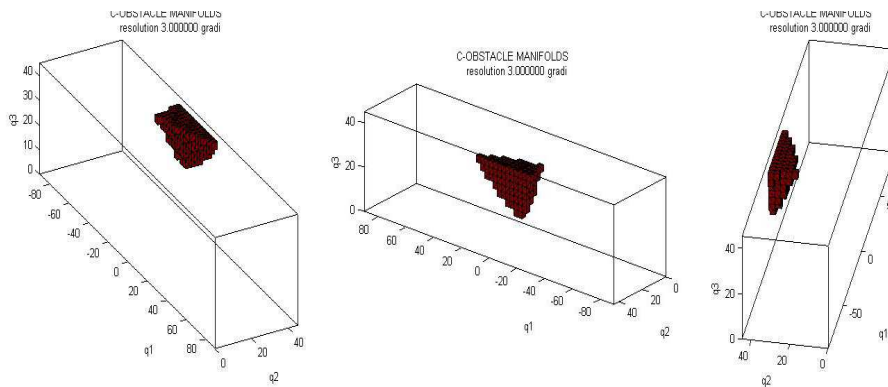


Figura 4.18:  $C$ -Space ottenuto con il metodo B.

### 4.5.2.3 Considerazioni e Confronto con il Metodo A

Nel paragrafo 4.5.1.5 abbiamo discusso dei problemi di performance del metodo A. Al contrario il metodo B riesce a fornire risultati equivalenti in qualche decimo di secondo (invece di decine di secondi). In figura 4.19 vengono mostrati i passaggi effettuati dai due diversi metodi. Questo nuovo metodo si è rivelato molto veloce ed affidabile e sarà sicuramente al centro di sviluppi futuri.

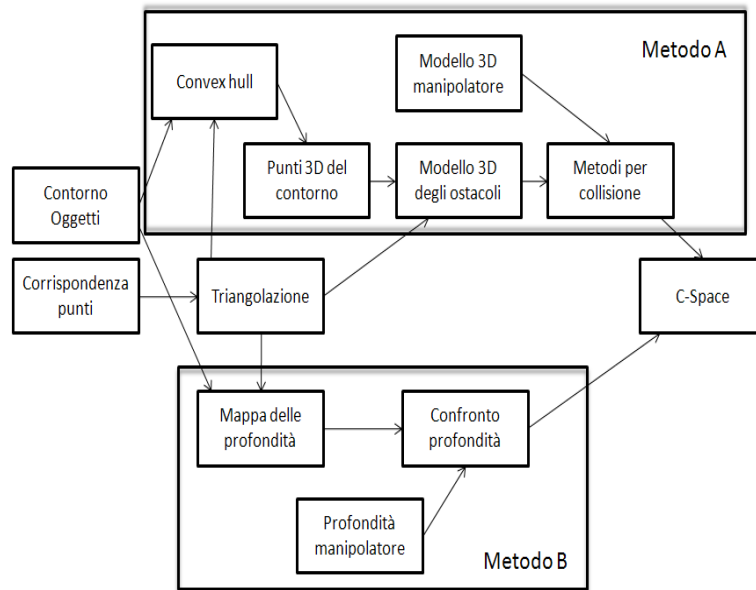


Figura 4.19: Confronto fra il metodo A ed il metodo B

# Capitolo 5

## Obstacle Avoidance : *C-Force*

In questo capitolo vedremo il metodo da noi sviluppato per ottenere una traiettoria del manipolatore che non entri in collisione con l'operatore. L'idea di base è quella delle forze repulsive<sup>1</sup>; la novità è che queste forze si basano completamente sul *C-Space*, in questo modo otteniamo dei valori direttamente utilizzabili nel controllo. Per le sue caratteristiche abbiamo chiamato questo metodo *C-Force* (forze nello spazio delle configurazioni).

Il metodo proposto è chiaramente ispirato al *potential field (PF)*[1]. Nel *PF* viene creata una mappa in cui il *targhet* ha un potenziale negativo ed gli ostacoli un potenziale positivo; viene quindi creato un percorso (che segue il gradiente) per raggiungere il minimo assoluto. La differenza principale è che il *PF* è un algoritmo di *path planning* mentre il *C-Force* si basa solo su un intorno; questo potrebbe sembrare un punto a sfavore, ma va considerato che il *PF* deve ricalcolare interamente il *path* ogni volta che il *C-Space* viene aggiornato; questo è incompatibile con le caratteristiche di tempo reale da noi volute. Come vedremo in questo capitolo la caratteristica del *C-Force* che lo rende preferibile al *PF* è la possibilità di tener conto anche della velocità degli oggetti e delle velocità del manipolatore, senza aumentare la complessità computazionale. Al contrario nel *PF* vengono considerati *C-Space* tempo varianti<sup>2</sup> [1] aumentando notevolmente sia il calcolo del *C-Space* che la realizzazione del *path*.

---

<sup>1</sup>Parliamo di forze perché usando la teoria delle forze si riesce a descrivere efficacemente il metodo utilizzato. Fisicamente sarebbe più corretto parlare di coppie repulsive.

<sup>2</sup>*C-Space* con una dimensione in più rappresentante il tempo.

## 5.1 Forza Repulsiva

L'idea di base del  $\mathcal{C}$ -Force è di applicare al manipolatore una forza repulsiva tale da permettere al manipolatore il passaggio solo su configurazioni appartenenti al  $\mathcal{C}$ -Free.

Possiamo considerare senza perdita di generalità un  $\mathcal{C}$ -Space di dimensione 2. La configurazione attuale del braccio è mappata come un punto nel  $\mathcal{C}$ -Space. La presenza di un  $\mathcal{C}$ -Obstacle imprime sul punto una forza inversamente proporzionale alla distanza ( $D$ ) fra il punto ed il  $\mathcal{C}$ -Obstacle e proporzionale ad una costante  $K$  (elastica). Tale rappresentazione può essere vista come una molla compressa (figura 5.1).

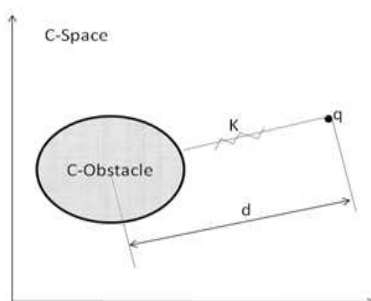


Figura 5.1: Modello a molla compressa

Consideriamo una porzione infinitesimale del  $\mathcal{C}$ -Obstacle ( $dC_O$ ), questa imprimerà una forza respingente  $dF$  con direzione data dalla retta (con vettore  $\vec{r}$ ) che congiunge  $dC_O$  con  $q^3$  (figura 5.2).

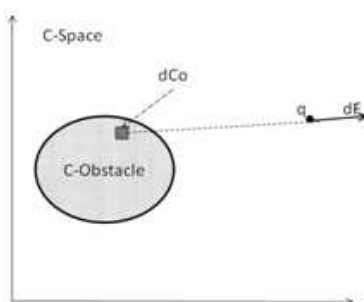


Figura 5.2: Forza respingente

$$dF = \frac{K}{D} dC_O \vec{r}, \quad (5.1)$$

<sup>3</sup>Punto che rappresenta la configurazione attuale.

da cui la forza repulsiva

$$F = \int_{C_o} dF. \quad (5.2)$$

Consideriamo la discretizzazione del C-Space effettuata<sup>4</sup>. Consideriamo un intorno della configurazione attuale ( $q$ ). Ogni Cella elementare di un *C-Obstacle* presente nell'intorno imprime una forza inversamente proporzionale alla distanza fra la cella ed il punto  $q$ ; con direzione data dalla retta che li congiunge. Facendo la somma di questi vettori ottengo la forza repulsiva (figura 5.3).

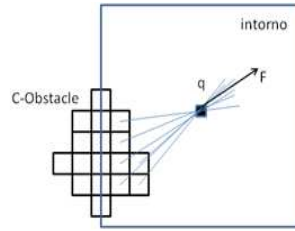


Figura 5.3: Forza respingente: *C-Space* discreto

Viene scelto solo un intorno della configurazione attuale perché :

- data la caratteristica della forza repulsiva le celle distanti sono trascurabili;
- mi consente di costruire il C-Space solo per un intorno<sup>5</sup> di  $q$  risparmiando tempo di calcolo ;
- semplifico il calcolo della Forza repulsiva.

Consideriamo in *C-Space* a 3 dimensioni<sup>6</sup> (figura 5.4). La forza repulsiva è data da

$$\begin{aligned} \mathbf{F}_R(\mathbf{q}) &= K_1 \sum_{i=1}^{N_C} \frac{1}{d(\mathbf{C}_i, \mathbf{q})} \frac{\mathbf{q} - \mathbf{C}_i}{\|\mathbf{q} - \mathbf{C}_i\|} = K_1 \sum_{i=1}^{N_C} \frac{\mathbf{q} - \mathbf{C}_i}{\|\mathbf{q} - \mathbf{C}_i\|^2} \\ &= F_1 \vec{q}_1 + F_2 \vec{q}_2 + F_3 \vec{q}_3. \end{aligned} \quad (5.3)$$

Dove  $N_C$  è il numero di celle nell'intorno,  $d(,)$  è l'operatore di distanza ed il termine  $\frac{\mathbf{q} - \mathbf{C}_i}{\|\mathbf{q} - \mathbf{C}_i\|}$  è la direzione fra la cella e la configurazione attuale.  $K_1$  è un fattore proporzionale.

È importante notare che la "*C-Forza*"<sup>7</sup>  $\mathbf{F}_R(\mathbf{q})$  è composta da tre componenti

<sup>4</sup>Avendo valutato un numero finito di configurazioni del manipolatore.

<sup>5</sup>Intorno condizionato dall'intorno per il calcolo della forza, dalla frequenza del ciclo di visione e dalla velocità in cui si può muovere il braccio.

<sup>6</sup>Dimensione del *C-Space* del manipolatore da noi utilizzato

<sup>7</sup>Forza nello stazio delle configurazioni.

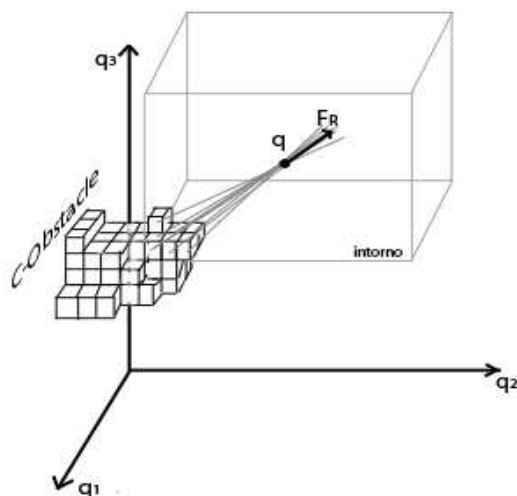


Figura 5.4: Forza respingente:  $C\text{-Space}$  3 dimensioni

$F_1$ ,  $F_2$  ed  $F_3$ ; dove  $F_1$  è la  $C$ -Forza che devo applicare al giunto  $J_0$ <sup>8</sup>,  $F_2$  quella al giunto  $J_1$  ed  $F_3$  al giunto  $J_3$ .

Sia  $\tau$  la coppia che applico ai giunti, questa sarà

$$\tau = \tau_C + \mathbf{F}_R(\mathbf{q}) \quad (5.4)$$

Dove  $\tau_C$  è la coppia data dal controllore (paragrafo 1.5.1).

## 5.2 Movimento dell'Ostacolo

Nel controllo appena presentato non si tiene conto della velocità con cui si muove l'ostacolo. Se l'ostacolo si muove verso il braccio ad una velocità superiore ad una certa soglia (relativamente piccola) non si riuscirà ad evitare l'ostacolo. Per stimare come si muove l'ostacolo esistono varie tecniche:

- **Flusso immagine:** Studiando le variazioni del flusso delle immagini (sottraggo ad ogni immagine la precedente) si può identificare il movimento dell'oggetto. Le informazioni che si riescono ad ottenere sono rispetto al piano immagine, ma non si riescono ad ottenere informazioni sul movimento nello spazio tridimensionale;

<sup>8</sup>Essendo il giunto rotoidale la  $C$ -Forza rappresenta fisicamente la coppia che devo applicare al giunto.



- **Collineazione:** Conoscendo almeno 5 punti 3D (nello spazio di proiezione) dell'oggetto, in 2 istanti diversi, posso stimare la collineazione ( $H_{12}$ ) fra i due istanti. Dalla collineazione posso ricavare la trasformazione  $T_{12}$  come

$$H_{12} = \alpha H_{PE}^{-1} T_{12} H_{PE},$$

dove  $\alpha = \text{sign}(\text{tr}(H_{12})) \sqrt[4]{\det(H_{12})}$  ed  $H_{PE}$  è la matrice di calibrazione metrica calcolata nella fase iniziale (paragrafo 2.3.4). Dato  $T_{12}$  e conoscendo l'intervallo di tempo fra i due istanti posso ottenere informazioni sulla velocità. Tale metodo però, è valido solo se l'oggetto è rigido (tutti i punti sono soggetti a  $T_{12}$ ) e quindi non si può applicare se l'oggetto che si muove è una persona.

- **Spostamento 3D dei punti** Facendo il matching fra 2 coppie di immagini stereo posso valutare lo spostamento dei singoli punti 3D fra due istanti. Considerando ogni punto a se stante posso valutare la velocità di ogni punto. Tale metodo è però molto sensibile ai rumori dati dal processo di visione; risulta quindi inadatto.

Tutti i metodi visti non consentono di ottenere efficacemente la stima voluta. Il metodo proposto si basa sul considerare gli spostamenti dei  $\mathcal{C}$ -Obstacle. Ogni oggetto viene mappato nel  $\mathcal{C}$ -Space da uno o più "agglomerati"<sup>9</sup>. Di ogni agglomerato posso calcolare Baricentro. Supponiamo che l'oggetto non si muova; a causa del rumore si avranno delle variazioni nella rappresentazione dell'ostacolo, da cui delle variazioni del  $\mathcal{C}$ -Obstacle. È ragionevole supporre che i baricentri del  $\mathcal{C}$ -Obstacle siano poco soggetti al rumore (figura 5.5(A)). Se l'oggetto si muove cambierà la dimensione, la forma e la posizione del  $\mathcal{C}$ -Obstacle, di conseguenza varia la posizione dei baricentri (figura 5.5(B)).

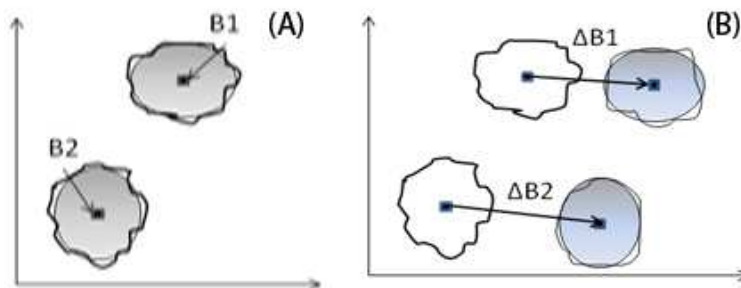


Figura 5.5: Variazione baricentri  $\mathcal{C}$ -Obstacle.

<sup>9</sup>Porzione del  $\mathcal{C}$ -Space composta da celle appartenenti al  $\mathcal{C}$ -Obstacle adiacenti.

Sia  $\Delta \mathbf{B}_i$  il vettore di variazione della posizione del baricentro dell' $i$ -esimo oggetto. Se la variazione è avvenuta in un tempo  $\Delta t$  si avrà la velocità

$$\mathbf{V}_i = \frac{\Delta \mathbf{B}_i}{\Delta t}.$$

Tale velocità rappresenta come si sposta il  $\mathcal{C}$ -Obstacle nel  $\mathcal{C}$ -Space, e quindi se ipotizzo la velocità del baricentro costante fra due stime consecutive del  $\mathcal{C}$ -Space, mi consente di fare una predizione delle prossime configurazioni occupate, ovvero le configurazioni che il manipolatore non può assumere nell'immediato futuro.

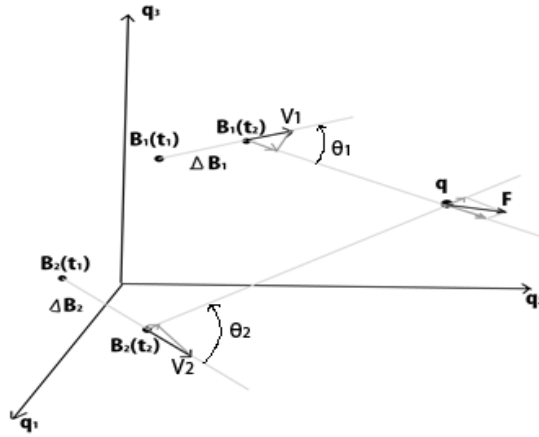


Figura 5.6: Variazione baricentri  $\mathcal{C}$ -Obstacle

Riferendoci alla figura 5.6; consideriamo il baricentro  $i$ -esimo con velocità  $\mathbf{V}_i$  e posizione attuale  $\mathbf{B}_i$ . La configurazione attuale è  $\mathbf{q}$ , il versore di direzione da  $\mathbf{q}$  a  $\mathbf{B}_i$  sarà

$$D\vec{I}R = \frac{\mathbf{q} - \mathbf{B}_i}{\|\mathbf{q} - \mathbf{B}_i\|}$$

Sia  $\theta_i$  l'angolo che formano  $\mathbf{V}_i$  e  $\mathbf{B}_i$  sul piano comune. La componente di  $\mathbf{V}_i$  che agisce su  $\mathbf{q}$  sarà

$$|\mathbf{V}_i| \cos \theta_i = \langle \mathbf{V}_i, D\vec{I}R \rangle \quad (5.5)$$

Vogliamo avere una forza repulsiva proporzionale alla componente 5.5, in modo da avere un *manovra evasiva*.

In totale avremo

$$\mathbf{F}_R(\mathbf{q}) = K_1 \sum_{i=1}^{N_C} \frac{\mathbf{q} - \mathbf{C}_i}{\|\mathbf{q} - \mathbf{C}_i\|^2} + K_2 \sum_{j=1}^{N_A} \frac{\langle \frac{\Delta \mathbf{B}_j}{\Delta t}, \mathbf{q} - \mathbf{B}_j \rangle \mathbf{q} - \mathbf{B}_j}{\|\mathbf{q} - \mathbf{B}_j\|^2}, \quad (5.6)$$

dove  $N_A$  è il numero di agglomerati considerati.

### 5.3 Movimento del Manipolatore

Fino ad ora si è considerata solo la configurazione statica del braccio  $\mathbf{q}$ , è interessante andare a considerare anche la  $\dot{\mathbf{q}}$ <sup>10</sup>. Facendo un ragionamento pratico si può dire che: se la configurazione del braccio si muove nello stesso verso con cui si muove il baricentro dell'agglomerato, la forza respingente necessaria per la *manovra evasiva* sarà minore di quella considerata nel caso standard; sarà maggiore se il verso è opposto. Per coerenza calcolo  $\dot{\mathbf{q}}$  come  $\frac{\Delta \mathbf{q}}{\Delta t}$ , dove  $\Delta t$  è il tempo impiegato dal ciclo di visione per effettuare la stima del *C-Space*, al posto di usare il valore calcolato nel ciclo di controllo.

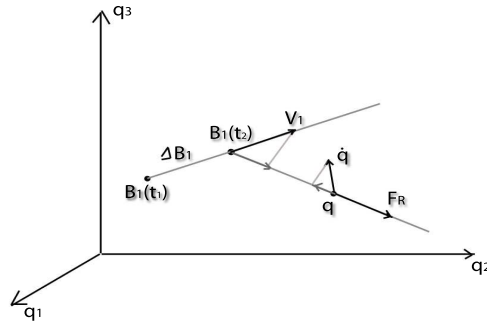


Figura 5.7: Movimento Manipolatore.

Sia  $\varphi_i$  l'angolo fra  $\dot{\mathbf{q}}$  e  $D\vec{I}R$  sul piano comune (figura 5.7), la velocità relativa sarà

$$|\mathbf{V}_{\text{REL}_i}| = \left| \frac{\Delta \mathbf{B}_i}{\Delta t} \right| \cos \theta_i + \left| \frac{\Delta \mathbf{q}}{\Delta t} \right| \cos \varphi_i, \quad (5.7)$$

la forza totale sarà

$$\mathbf{F}_R(\mathbf{q}) = K_1 \sum_{i=1}^{N_C} \frac{\mathbf{q} - \mathbf{C}_i}{\|\mathbf{q} - \mathbf{C}_i\|^2} + K_2 \sum_{j=1}^{N_A} |\mathbf{V}_{\text{REL}_i}| \frac{\mathbf{q} - \mathbf{B}_j}{\|\mathbf{q} - \mathbf{B}_j\|}. \quad (5.8)$$

La 5.8 può essere scritta come

$$\mathbf{F}_R(\mathbf{q}) = K_1 \sum_{i=1}^{N_C} \frac{\mathbf{q} - \mathbf{C}_i}{\|\mathbf{q} - \mathbf{C}_i\|^2} + K_2 \sum_{j=1}^{N_A} \frac{\langle \Delta \mathbf{B}_j - \Delta \mathbf{q}, \mathbf{q} - \mathbf{B}_j \rangle}{\|\mathbf{q} - \mathbf{B}_j\|^2} \mathbf{q} - \mathbf{B}_j. \quad (5.9)$$

<sup>10</sup>Variazione della configurazione nel tempo.

In teoria  $K_2$  può essere scomposto in modo da dare un peso differente alla velocità del braccio e a quella dell'oggetto. Si avrà con velocità del braccio e dell'oggetto di

- **verso concorde.** la forza per la manovra evasiva diminuisce<sup>11</sup>;
- **verso opposto.** la forza per la manovra evasiva aumenta

Se l'oggetto è fermo fa in modo che la velocità del manipolatore non sia eccessiva nella direzione del  $\mathcal{C}$ -Obstacle.

Nella pratica la condizione di verso concorde viene annullata per non rallentare la manovra evasiva.

## 5.4 Manovra Evasiva senza il calcolo dei Baricentri

La forza repulsiva generata dalla velocità dell'oggetto e dalla velocità della configurazione del manipolatore viene calcolata riferendosi al baricentro dell'agglomerato. Il calcolo del baricentro è computazionalmente dispendioso ed è quindi incompatibile con la ricerca di performance fatte fino ad ora.

Nel paragrafo precedente abbiamo considerato la forza repulsiva formata dal modulo  $|\mathbf{V}_{\mathbf{REL}_i}|$  e la direzione  $\frac{\mathbf{q}-\mathbf{B}_j}{\|\mathbf{q}-\mathbf{B}_j\|}$ ; se al posto del baricentro di tutto l'agglomerato considero solo il baricentro delle celle in un intorno di  $\mathbf{q}$  avrò la direzione  $\sum_{i=1}^{N_C} \frac{\mathbf{q}-\mathbf{C}_i}{\|\mathbf{q}-\mathbf{C}_i\|}$  che abbiamo già utilizzato e quindi non comporta ulteriori calcoli.

Per quanto riguarda la velocità relativa questa è pari alla variazione della distanza; consideriamo la distanza media nell'intorno

$$d_M = \sum_{i=1}^{N_C} \frac{d(\mathbf{q}, \mathbf{C}_i)}{N_c},$$

dove  $d(, )$  è l'operatore di distanza. La variazione di tale distanza sarà

$$\frac{d_M(t) - d_M(t + \Delta t)}{\Delta t}.$$

---

<sup>11</sup>Rispetto al caso in cui non valuto la velocità del manipolatore

Da cui

$$\begin{aligned} \mathbf{F}_R(\mathbf{q}) = & K_1 \sum_{i=1}^{N_C} \frac{\mathbf{q} - \mathbf{C}_i}{\|\mathbf{q} - \mathbf{C}_i\|^2} + \\ & + K_2 \frac{1}{N_c \Delta t} \left\langle \mathbf{d}_M(t) - \mathbf{d}_M(t - \Delta t), \sum_{i=1}^{N_C} \frac{\mathbf{q} - \mathbf{C}_i}{\|\mathbf{q} - \mathbf{C}_i\|} \right\rangle \sum_{i=1}^{N_C} \frac{\mathbf{q} - \mathbf{C}_i}{\|\mathbf{q} - \mathbf{C}_i\|}. \end{aligned} \quad (5.10)$$

Chiamiamo questo metodo *MDC-Force* (*Mean Distance C-Force*).

Un approccio alternativo consiste nel considerare la forza repulsiva dovuta alla distanza fra la configurazione attuale ed il *C-Obstacle* (paragrafo 5.1) e la sua derivata temporale. La forza repulsiva dovuta alla distanza aumenta rapidamente se il *C-Obstacle* e la configurazione attuale si stanno avvicinando rapidamente.

Sia

$$\mathbf{f}_d(\mathbf{q}, t) = \sum_{i=1}^{N_C} \frac{\mathbf{q} - \mathbf{C}_i}{\|\mathbf{q} - \mathbf{C}_i\|^2} \quad (5.11)$$

la forza repulsiva dovuta alla distanza. Posso calcolare la forza repulsiva totale come:

$$\mathbf{F}(\mathbf{q}) = K_P \mathbf{f}_d(\mathbf{q}, t) + K_D \frac{\mathbf{f}_d(\mathbf{q}, \tilde{t}) - \mathbf{f}_d(\mathbf{q}, \tilde{t} - \Delta t)}{\Delta t}; \quad (5.12)$$

il *C-Obstacle* ha un tempo di aggiornamento pari a  $\Delta t$ . Il valore  $\tilde{t}$  rappresenta il tempo a cui è stato effettuato l'ultimo aggiornamento del *C-Space*. In pratica la prima parte di  $\mathbf{F}(\mathbf{q})$  viene calcolata ad ogni ciclo di controllo, mentre la seconda parte viene ricalcolata ogni volta che il *C-Space* viene aggiornato. Fra due aggiornamenti del *C-Space* viene considerata costante la velocità relativa fra la configurazione attuale ed il *C-Obstacle*. Abbiamo chiamato questa tecnica *PDC-Force* (*Proportional Derivative C-Force*).

## 5.5 *C-Force*: Simulazioni

Al fine di provare il corretto funzionamento degli algoritmi esposti in questo capitolo abbiamo realizzato un simulatore utilizzando Matlab<sup>®</sup>. Il simulatore del metodo *C-Force* considera un *C-Space* a due dimensioni. La simulazione inizia da una configurazione di partenza (un punto del *C-Space*); viene interrotta quando si raggiunge la configurazione desiderata (target) oppure

se avviene una collisione (configurazione attuale entra nel  $\mathcal{C}$ -Obstacle). Il  $\mathcal{C}$ -Obstacle viene rappresentato da un ellisse e può essere statico o può muoversi durante la simulazione. Il simulatore considera che non si conosce la posizione del  $\mathcal{C}$ -Obstacle ad ogni istante, ma soltanto ogni  $\Delta t$ ; cioè considera il tempo impiegato dal ciclo di visione per calcolare il nuovo  $\mathcal{C}$ -Space; ogni volta che la posizione del  $\mathcal{C}$ -Obstacle viene aggiornata il  $\mathcal{C}$ -Obstacle viene rappresentato tratteggiato.

### 5.5.1 Simulazione 1

La prima simulazione è eseguita mantenendo il  $\mathcal{C}$ -Obstacle statico; come si vede da figura 5.8 l'utilizzo del  $\mathcal{C}$ -Force permette di raggiungere la configurazione desiderata (targhet) evitando il  $\mathcal{C}$ -Obstacle.

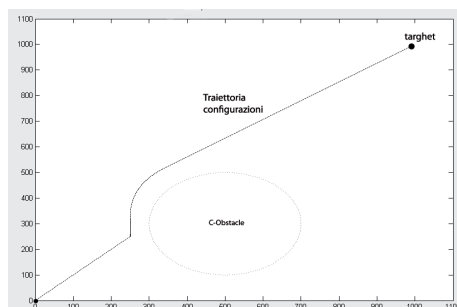


Figura 5.8: Simulazione 1

### 5.5.2 Simulazione 2

La seconda simulazione considera un  $\mathcal{C}$ -Obstacle in movimento. In figura 5.9 è rappresentati il risultato della simulazione considerando la  $\mathcal{C}$ -Forza data solo dall'equazione 5.3, senza tener conto del movimento del  $\mathcal{C}$ -Obstacle. In questo caso la  $\mathcal{C}$ -Forza non è sufficiente ad evitare il  $\mathcal{C}$ -Obstacle. Consideriamo lo stesso scenario, ma applichiamo la  $MDC$ -Forza; come si vede in figura 5.10 si riesce ad evitare il  $\mathcal{C}$ -Obstacle ottenendo una manovra evasiva<sup>12</sup>. Se viene utilizzata la  $PDC$ -Forza sullo stesso scenario si ottiene il risultato mostrato in figura 5.11. In figura 5.11 si vede come la  $PDC$ -Forza ha un comportamento più “brusco” rispetto alla  $MDC$ -Forza; questo perché dipende dall'inverso di una distanza invece che da una distanza, quindi quando la distanza è piccola la  $\mathcal{C}$ -Forza generata è molto alta.

<sup>12</sup>Il manipolatore si muove in una direzione “opposta” a quella desiderata per evitare la collisione.

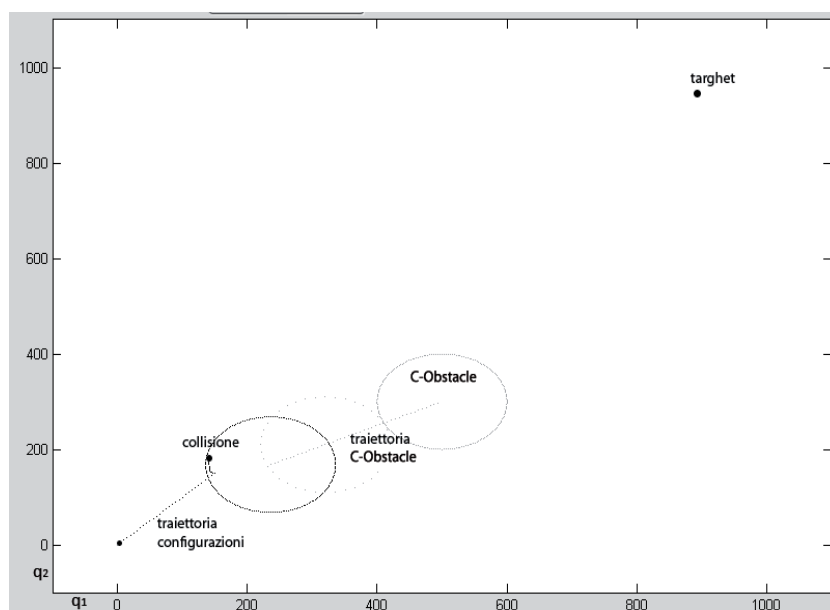


Figura 5.9: Simulazione 2:  $C$ -Force eq.5.3

### 5.5.3 Simulazione 3

In questa terza prova viene aumentata la velocità del  $C$ -Obstacle (rispetto alla simulazione 2). in figura 5.12 ed in figura 5.13 viene mostrato come sia la  $MDC$ -Force che la  $PDC$ -Force riescono anche in questo caso a evitare il  $C$ -Obstacle.

### 5.5.4 Simulazione 4

Questa terza prova è stata opportunamente dimensionata per mostrare una caratteristica della  $PDC$ -Force. Come si vede in figura 5.14 a volte la  $C$ -Forza data dalla  $PDC$ -Force è molto elevata, questo avviene quando si ha un aggiornamento del  $C$ -Obstacle con la configurazione attuale molto vicina. In pratica il manipolatore riceve un'accelerazione improvvisa che lo porta ai limiti dello spazio di lavoro. Pur evitando comunque la collisione con il  $C$ -Obstacle tale comportamento non è desiderabile. In figura 5.15 si vede come utilizzando nello stesso scenario la  $MDC$ -Force si ottiene una traiettoria decisamente migliore.

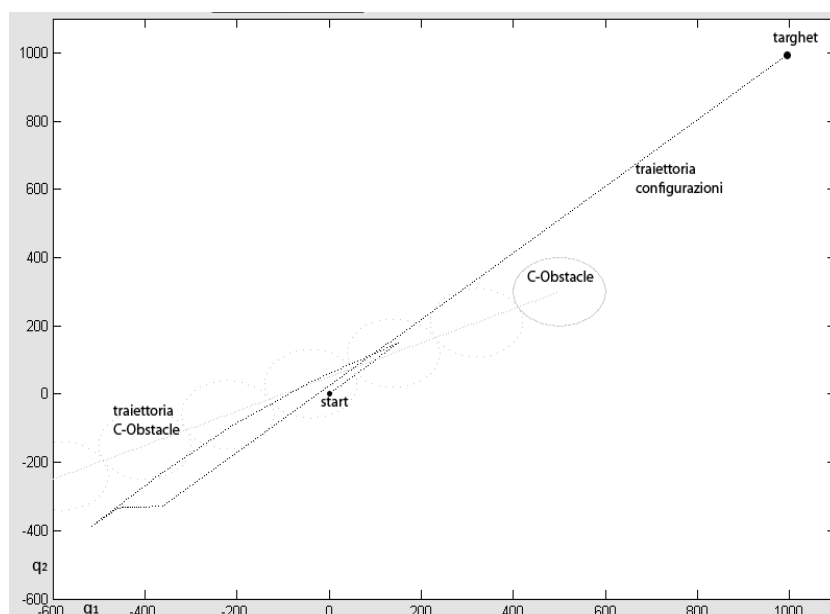


Figura 5.10: Simulazione 2: MDC-Force

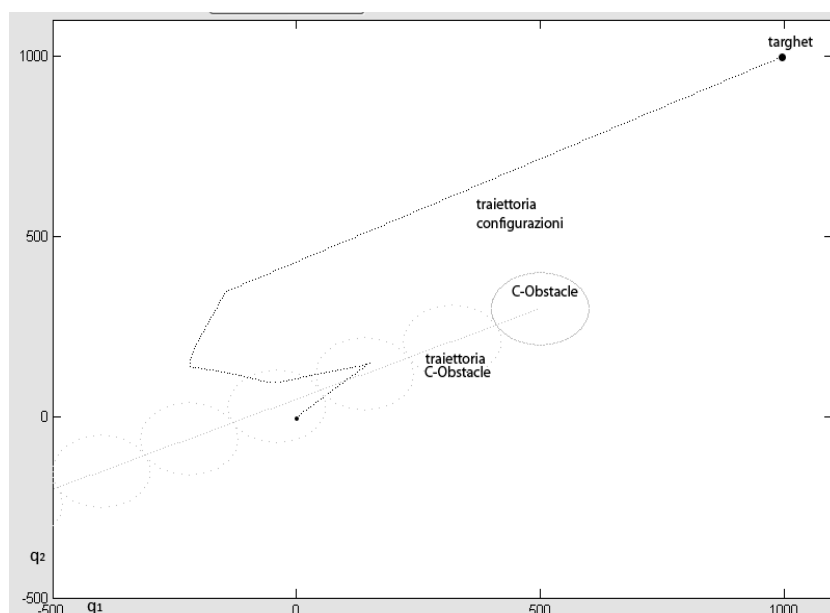


Figura 5.11: Simulazione 2: PDC-Force



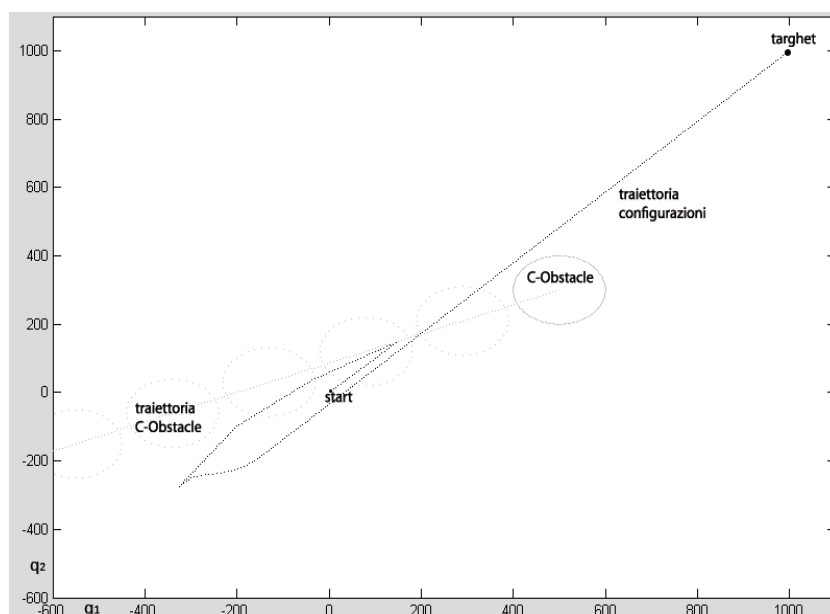


Figura 5.12: Simulazione 3: MDC-Force

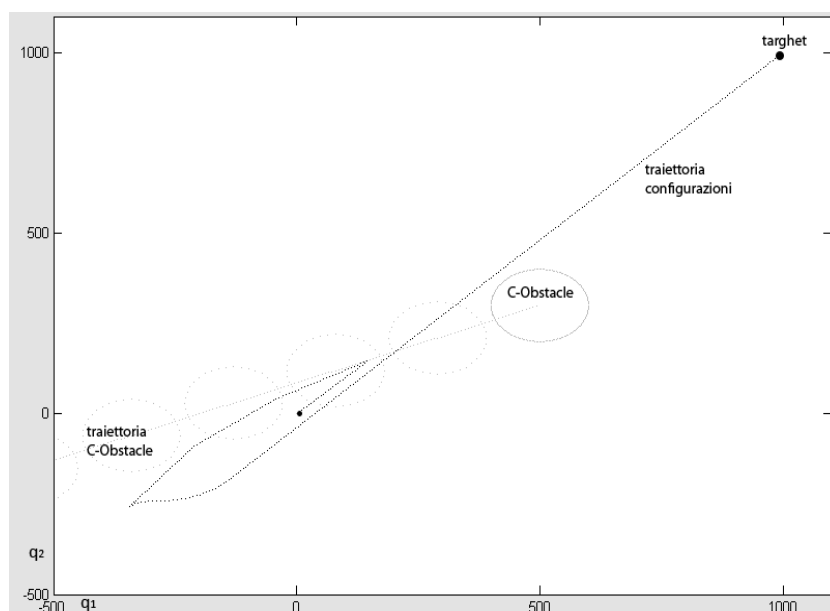


Figura 5.13: Simulazione 3: PDC-Force

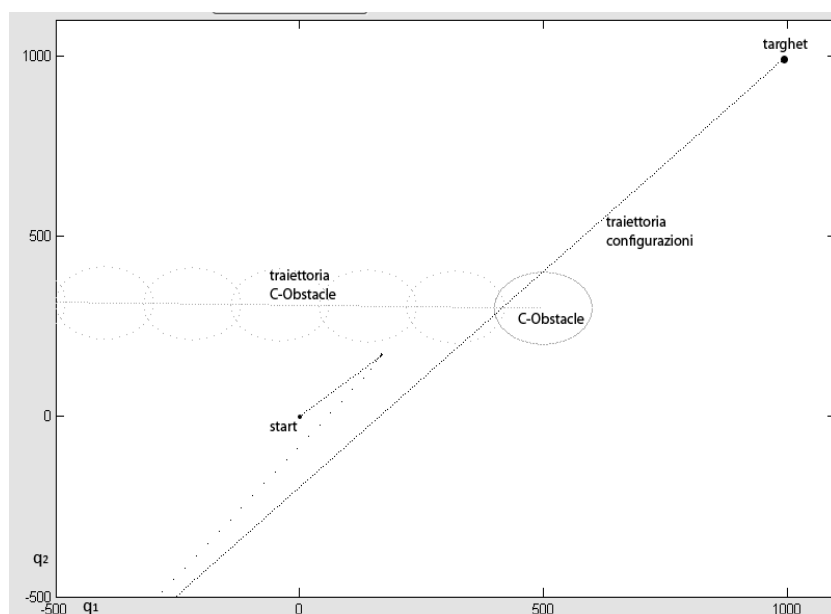


Figura 5.14: Simulazione 4: PDC-Force

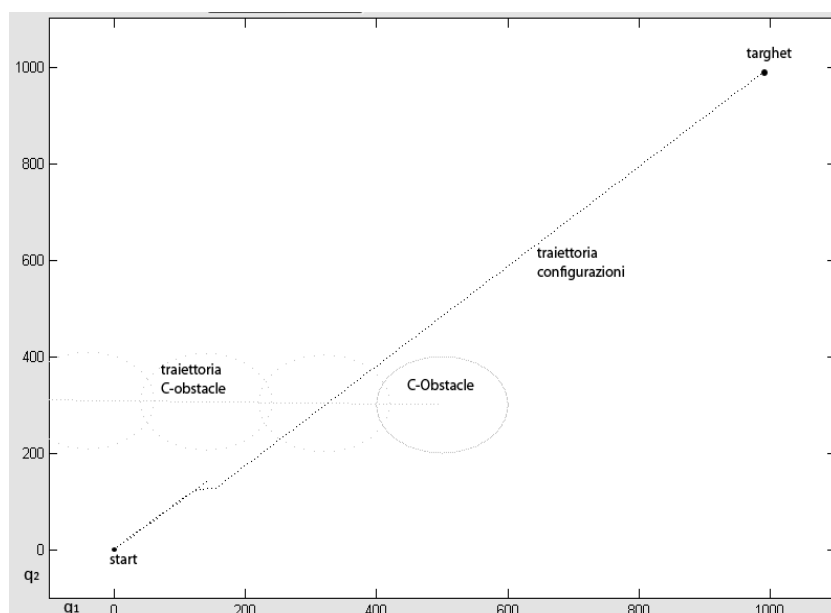


Figura 5.15: Simulazione 4: MDC-Force

# Capitolo 6

## Risultati



Figura 6.1: Test 1.

In questo capitolo illustreremo una serie di prove effettuate in laboratorio e commenteremo i risultati ottenuti. L'architettura utilizzata per effettuare le prove è composta dal manipolatore *SoftArm* e da una coppia di camere montate ad una altezza (coordinata  $z$ ) di  $80\text{cm}$  rispetto al sistema di riferimento del manipolatore (figura 6.1). Il sistema è controllato da due calcolatori. Un calcolatore gestisce il server di visione e l'altro controlla il manipolatore. I due calcolatori comunicano utilizzando il protocollo TCP (appendice A).

## 6.1 Test 1

Consideriamo costante la configurazione di riferimento del manipolatore, cioè vogliamo che il manipolatore resti fermo in una posizione prestabilita. In questa condizione inseriamo un ostacolo, nel caso specifico la mano dell'operatore. Quando l'ostacolo si avvicina al manipolatore il sistema *C-Force* previene la collisione generando delle forze repulsive. In figura 6.2 possiamo vedere una serie di immagini prodotte sovrapponendo i frame del video realizzato durante l'esecuzione del test 1. La realizzazione di questo primo test ha mostrato come l'integrazione fra il server di visione e il *C-Force* è concepita in maniera corretta e permette di ottenere i risultati attesi. Il tempo di reazione<sup>1</sup> varia tra i 2 ed i 5 secondi, a seconda del numero di volte che il metodo delle SIFT fallisce. Le prestazioni in tempo reale sono inferiori a quelle desiderate.

---

<sup>1</sup>Tempo che intercorre da quando si inserisce l'ostacolo a quando si ha la reazione corrispondente da parte del manipolatore.



Figura 6.2: Test 1.

## 6.2 Test 2

In questo secondo test vogliamo che il manipolatore esegui un determinato *task*. Durante l'esecuzione del *task* l'operatore entra nel *task space* del manipolatore. In figura 6.3 è mostrato il risultato del secondo test come sovrapposizione dei frame del video girato durante il test. Il secondo test conferma i risultati ottenuti con le simulazioni. Utilizzando il nostro sistema si riesce ad ottenere l'*obstacle avoidance* in modo efficace. I limiti di prestazione in real time riscontrati nel primo test si presentano chiaramente anche nel secondo test. I limiti di prestazione si riflettono in limiti di velocità per l'operatore.



Figura 6.3: Test 2.

## 6.3 Commenti ai Risultati

Gli obiettivi che ci eravamo prefissi sono stati conseguiti. Il sistema realizzato garantisce la piena coesistenza fra l'operatore ed il manipolatore utilizzando solo le informazioni provenienti dalla stereovisione. Durante i test si sono riscontrati dei limiti nelle prestazioni in tempo reale, questi limiti non invalidano la correttezza del nostro sistema, ma comportano delle limitazioni sulla velocità dei movimenti dell'operatore. La causa dei limiti di prestazioni è da attribuire alla scarsa potenza di calcolo a disposizione ed alla presenza di algoritmi che richiedono un tempo di calcolo elevato individuati all'interno del server di visione; in particolare l'estrazione e matching delle *feature* e la gestione dei contorni costituiscono il collo di bottiglia dell'intero sistema.

La creazione di un metodo alternativo per la costruzione della *mappa delle profondità* che elimini i passi che costituiscono il collo di bottiglia è già in fase di sviluppo.

# Capitolo 7

## Conclusioni e Sviluppi Futuri

Attraverso esperimenti condotti in laboratorio abbiamo dimostrato per via sperimentale la possibilità di ricostruire il *C-Space* associato ad una coppia di immagini. Abbiamo inoltre dimostrato l'effettività del controllo *C-Force* proposto in questo lavoro. Nel funzionamento a ciclo chiuso abbiamo ottenuto risultati molto incoraggianti. Il sistema funziona correttamente ma le prestazioni in tempo reale sono al di sotto di quelle desiderate. Questo è dovuto alla bassa velocità del ciclo di visione (in media circa 0.5 Hz), imposto soprattutto dalla limitata potenza di calcolo in nostro possesso.

Per migliorare le prestazioni in tempo reale del nostro sistema stiamo ideando un nuovo algoritmo per la costruzione della *mappa delle profondità*; questo nuovo metodo non utilizzerà le *SIFT* (che non sono sempre affidabili) ed il *background subtraction* (che limita l'utilizzo ad uno spazio di lavoro fisso). Secondo le nostre previsioni, l'utilizzo di questo nuovo metodo unito ad un aumento della potenza di calcolo a disposizione consentirà di ottenere le prestazioni in tempo reale da noi desiderate.

Il sistema realizzato (con gli sviluppi citati) esula da un'applicazione specifica e rappresenta un metodologia per l'*Obstacle Avoidance* con visione stereoscopica adatta ad ogni applicazione, ad esempio potrà essere utilizzata con manipolatori mobili (spazio di lavoro variabile) o direttamente come sistema di *Obstacle Avoidance* per veicoli.

Si ringrazia il Prof. Antonio Bicchi per l'opportunità data.

Si ringrazia inoltre tutti i membri del Centro di Ricerca Interdipartimentale "E.Piaggio", in primis l'Ing. Riccardo Schiavi che ha seguito questa tesi in tutte le sue fasi.



# Appendice A

## Comunicazione : Server Visione ↔ Controllo

In questa appendice vengono esposte le soluzioni adottate per realizzare il nostro sistema su una piattaforma che non supporta il *real time*.

Per la realizzazione del nostro sistema abbiamo utilizzato una piattaforma *Windows XP*®; la scelta della piattaforma è stata dettata dall'utilizzo di alcune librerie disponibili solo per piattaforme *Windows*® e dalla compatibilità di alcuni driver.

Il sistema operativo *Windows XP*® consente l'utilizzo del *multitasking* ma il *tick*<sup>1</sup> consentito è di *1ms*. Creando un task per il ciclo di controllo sotto queste condizioni si avrà sempre un controllo con banda minore ad *1KHz*. Il manipolatore utilizzato per le prove pratiche deve essere controllato ad una frequenza maggiore ad *1KHz* per garantire la stabilità.

Per superare questa incompatibilità abbiamo deciso di utilizzare due calcolatori; uno per il server di visione ed un altro per il controllo del manipolatore. I due calcolatori si scambiano le informazioni utilizzando il protocollo TCP.

### A.1 Comunicazione TCP

I due calcolatori sono collegati con un cavo *ethernet* incrociato in modo da creare una rete punto-punto. La comunicazione avviene utilizzando il protocollo TCP. Il calcolatore che gestisce il server di visione è anche il server della rete. Le operazioni per inizializzare la comunicazione sono mostrate in figura A.1.

La comunicazione avviene attraverso lo scambio di messaggi. I messaggi vengono sempre coordinati dal server di rete. Il server invia al client messaggi

---

<sup>1</sup>Unità temporale minima con cui sono gestiti i task.

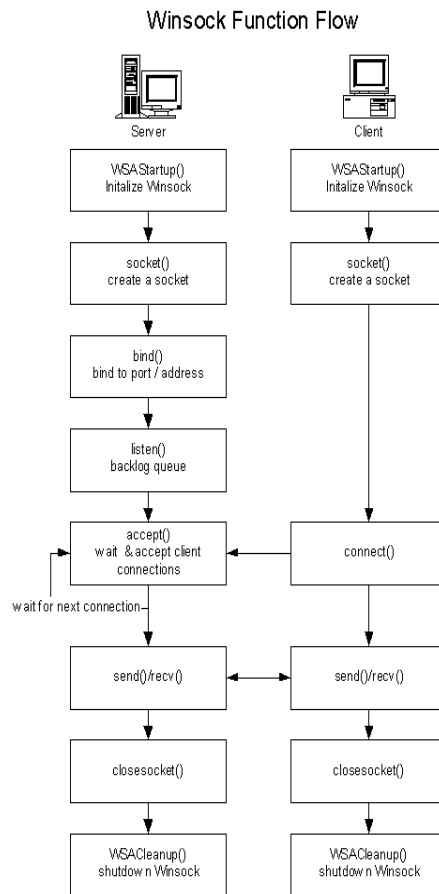


Figura A.1: Winsock

composti da un comando, e secondo la tipologia di comando il messaggio può contenere dati aggiuntivi. Nella tabella A.1 sono raccolti i messaggi che compongono il protocollo utilizzato.

Comando	Operazione richiesta	Operazioni successive
START	Abilita il controllo del manipolatore	-
STOP	Disabilita il controllo del manipolatore	-
NEW REF	Informa dell'invio di un nuova configurazione di riferimento	Invia la configurazione di riferimento
REQ POS	Richiede la configurazione attuale del manipolatore	Attende la ricezione della configurazione
C-SPACE SEND	Informa dell'invio di un nuovo <i>C-Space</i>	Invia il <i>C-Space</i>
C-SPACE RESET	Resetta il <i>C-Space</i> : nessun oggetto presente	-

Tabella A.1: Messaggi inviati dal server di rete al client

# Appendice B

## Complementi algebrici

In questa appendice presentiamo sinteticamente alcuni strumenti e proprietà matematiche che sono state utilizzati nella tesi.

### B.1 Geometria Omogenea

La geometria omogenea è stata ideata da August Ferdinand Möbius intorno al 1837. La geometria omogenea introduce strumenti per descrivere i punti nella geometria proiettiva. Utilizzando le coordinate omogenee per esprimere i punti nello spazio, la concatenazione di trasformazioni si riduce al prodotto di matrici di trasformazione omogenee; l'uso delle coordinate omogenee non è solo un comodo artificio di calcolo, ma ha un significato geometrico molto più profondo, legato al concetto di proiezione e a quello di "punti all'infinito".

#### B.1.1 Coordinate omogenee

Chiameremo coordinate omogenee di un punto  $\mathbf{p} = (x, y)$  del piano una qualsiasi terna ordinata  $(x', y', w)$  di numeri reali tali che  $w \neq 0$ . Qualsiasi punto  $(x, y, 1)$  rappresenta lo stesso punto sul piano del punto con coordinate  $(rx, ry, r) : r \neq 0$ . Allo stesso modo posso esprimere un punto nello spazio  $\mathbf{P} = (x, y, z)$  con le coordinate omogenee  $(x', y', z', w)$ .

I punti con coordinata  $w = 0$  rappresentano i *punti ad infinito*. Un punto ad infinito rappresenta il punto dato dall'intersezione fra le proiezioni su un piano di due rette parallele. Questa proprietà permette di esprimere dei punti che fisicamente non avrebbero alcun senso.

### B.1.2 Retta omogenea su un piano

Una retta omogenea su un piano viene espressa con un vettore di tre elementi  $\mathbf{l} = [a, b, c]^T$ . Tale retta è composta dai punti con coordinate omogenee  $\mathbf{p} = (x, y, w)$  che soddisfano la condizione

$$ax + by + cw = 0 \implies \mathbf{p}^T \times \mathbf{l} = 0$$

#### B.1.2.1 Retta fra due punti

La retta passante fra due punti con coordinate omogenee  $\mathbf{p}, \mathbf{p}'$  sarà data da

$$\mathbf{l} = \mathbf{p} \times \mathbf{p}'$$

#### B.1.2.2 Intersezione fra due rette

Date due rette omogenee  $\mathbf{l}$  ed  $\mathbf{l}'$  il punto sul piano dato dalla loro intersezione sarà:

$$\mathbf{p} = \mathbf{l} \times \mathbf{l}'$$

#### B.1.2.3 Distanza minima fra una retta ed un punto

Dato un punto in coordinate omogenee  $\mathbf{p}$  ed una retta  $\mathbf{l}$  la loro distanza sarà:

$$d = \mathbf{p}^T \mathbf{l}$$

### B.1.3 Piano omogeneo nello spazio

Un piano omogeneo nello spazio viene espresso con un vettore di quattro elementi  $\pi = [a, b, c, d]^T$ . Tale retta è composta dai punti con coordinate omogenee  $\mathbf{p} = (x, y, z, w)$  che soddisfano la condizione

$$ax + by + cz + dw = 0 \implies \mathbf{p}^T \times \pi = 0$$

#### B.1.3.1 Piano passante per tre punti

Dati tre punti omogenei nello spazio  $\mathbf{P}_1, \mathbf{P}_2$  e  $\mathbf{P}_3$  non colineari<sup>1</sup> possiamo costruire il piano risolvendo l'equazione

$$\begin{bmatrix} \mathbf{P}_1^T \\ \mathbf{P}_2^T \\ \mathbf{P}_3^T \end{bmatrix} \pi = 0,$$

<sup>1</sup>Che non si trovano sulla stessa linea

Una soluzione in forma chiusa può essere ottenuta utilizzando le coordinate non omogenee dei punti. Sia  $\mathbf{P}_i = [\tilde{\mathbf{P}}_i, 1]^T$ , il piano sarà dato da

$$\pi = \begin{bmatrix} (\tilde{\mathbf{P}}_1 - \tilde{\mathbf{P}}_3) \times (\tilde{\mathbf{P}}_2 - \tilde{\mathbf{P}}_3) \\ -\tilde{\mathbf{P}}_3(\tilde{\mathbf{P}}_1 \times \tilde{\mathbf{P}}_2) \end{bmatrix}$$

### B.1.3.2 Piano costruito minimizzando la distanza fra più punti

Dato un insieme di  $N$  punti  $\mathbf{P}_i$  il piano costruito utilizzando questi punti è dato dalla soluzione dell'equazione

$$\begin{bmatrix} \mathbf{P}_1^T \\ \mathbf{P}_2^T \\ \vdots \\ \mathbf{P}_N^T \end{bmatrix} \pi = 0.$$

Il sistema non ha soluzioni se  $N > 3$ . Una soluzione che minimizza la distanza euclidea fra i punti ed il piano viene trovata utilizzando la scomposizione ai valori singolari.

### B.1.3.3 Intersezione fra un piano ed il raggio di proiezione di un punto sul piano immagine

Dato un punto sul piano immagine  $\mathbf{p} = (x, y)$  il raggio di proiezione è dato da

$$\mathbf{A} = \begin{bmatrix} x\mathcal{P}^{3T} - \mathcal{P}^{1T} \\ y\mathcal{P}^{3T} - \mathcal{P}^{2T} \end{bmatrix},$$

dove  $\mathcal{P}^{iT}$  è la  $i$ -esima riga della matrice di proiezione della camera. Dato il piano nello spazio  $\pi$ . Il punto  $\mathbf{P}$  dato dall'intersezione fra il piano  $\pi$  e la proiezione del punto nel piano immagine  $\mathbf{p}$  è dato dalla soluzione del sistema di equazioni

$$\begin{bmatrix} \mathbf{A} \\ \pi^T \end{bmatrix} \mathbf{P} = 0.$$

## B.2 Soluzione di un sistema di equazioni lineari

Dato un sistema di equazioni

$$\mathbf{A}x = 0 \tag{B.1}$$

con  $\mathbf{A} \in \mathbb{R}^{n \times m}$ ; con  $n > m$ . Il sistema ha più equazioni che incognite, se le equazioni linearmente indipendenti sono maggiori di  $m$  il sistema non ha soluzioni. Consideriamo soluzione del sistema B.1 il valore  $\tilde{x}$  per cui  $\mathbf{A}\tilde{x}$  è minimizzato secondo la norma euclidea. La soluzione  $\tilde{x}$  viene trovata utilizzando la *scomposizione ai valori singolari* (SVD) della matrice  $\mathbf{A}$ .

La scomposizione ai valori singolari di  $\mathbf{A}$  sarà

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

dove  $\mathbf{U}$  è una matrice  $m \times n$  con colonne ortogonali,  $\mathbf{\Sigma}$  è una matrice  $n \times n$  diagonale e  $\mathbf{V}$  è una matrice  $n \times n$  ortogonale. La matrice  $\mathbf{\Sigma}$  contiene sulla diagonale i valori singolari della matrice  $\mathbf{A}$  in ordine decrescente.

La soluzione  $\tilde{x}$  del sistema B.1 è data dall'ultima colonna della matrice  $\mathbf{V}$  [3], associata al valore singolare più piccolo.

## B.3 Matrice skew simmetrica

Una matrice  $\mathbf{A}$  è detta *skew-simmetrica* se  $\mathbf{A}^T = -\mathbf{A}$ .

Di particolare interesse sono le matrici  $3 \times 3$  skew-simmetriche; sia  $\mathbf{a} = (a_1, a_2, a_3)$  un vettore di dimensione 3 la matrice skew-simmetrica generata dal vettore  $\mathbf{a}$  è data da

$$[\mathbf{a}]_{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

Dato un vettore di dimensione 3 ( $\mathbf{b}$ ) il prodotto vettoriale  $\mathbf{a} \times \mathbf{b}$  è dato da  $[\mathbf{a}]_{\times} \mathbf{b}$

## B.4 Norma di Frobenius

La norma di Frobenius è una norma di matrici  $m \times n$  ( $\mathbf{A}$ ) definita come

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}. \quad (\text{B.2})$$

La norma di Frobenius può anche essere considerata una norma di vettori

$$\|\mathbf{A}\|_F = \sqrt{\text{trace}(\mathbf{A}\mathbf{A}^H)}, \quad (\text{B.3})$$

dove  $\mathbf{A}^H$  è la matrice *hermitiana*<sup>2</sup>. La norma di Frobenius sarà anche uguale a

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{\min(m,n)} \sigma_i^2}, \quad (\text{B.4})$$

---

<sup>2</sup>trasposta coniugata



# Appendice C

## Manipolatore *SoftArm*

In questa appendice presentiamo sinteticamente le matrici che rappresentano la cinematica e la dinamica del manipolatore utilizzato per le prove pratiche.

### C.1 Cinematica diretta

Consente di conoscere la posizione dell'*end-effector* data una configurazione ( $\mathbf{q}$ ).

$$\begin{aligned} {}^0T_E &= \begin{pmatrix} C_1S_{233} & C_1C_{233} & S_1 & -L_{12}S_1 + (L_2S_2 + L_{23}S_{23} + L_3S_{233})C_1 \\ S_1S_{233} & S_1C_{233} & -C_1 & L_{12}C_1 + (L_2S_2 + L_{23}S_{23} + L_3S_{233})S_1 \\ -C_{233} & S_{233} & 0 & L_1 - L_2C_2 - L_{23}C_{23} - L_3C_{233} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\ &= \left( \begin{array}{c|c} \textit{rotazione} & \textit{traslazione} \\ \hline \textit{prospettiva} & \textit{scala} \end{array} \right) \end{aligned} \quad (\text{C.1})$$

## C.2 Cinematica differenziale

### C.2.1 Jacobiano di posizione

$$\mathcal{J}_p = \left( \begin{array}{c|c|c} -L_{12}C_1 - \mu_1S_1 & \mu_2C_1 & \mu_3C_1 \\ -L_{12}S_1 + \mu_1C_1 & \mu_2S_1 & \mu_3S_1 \\ \hline 0 & \mu_1 & \mu_4 \end{array} \right) \quad (C.2)$$

in cui

$$\begin{aligned} \mu_1 &= [L_2S_2 + L_{23}S_{23} + L_3S_{233}] \\ \mu_2 &= [L_2C_2 + L_{23}C_{23} + L_3C_{233}] \\ \mu_3 &= [L_{23}C_{23} + 2L_3C_{233}] \\ \mu_4 &= [L_{23}S_{23} + 2L_3S_{233}] \end{aligned}$$

### C.2.2 Jacobiano di orientazione

$$\mathcal{J}_o = \begin{pmatrix} 0 & \sin(q_1) & 2\sin(q_1) \\ 0 & -\cos(q_1) & -2\cos(q_1) \\ 1 & 0 & 0 \end{pmatrix} \quad (C.3)$$

### C.2.3 Inversione Jacobiano di posizione

$$\mathcal{J}_p^{-1} = \begin{pmatrix} -\frac{S_1}{\mu_1} & \frac{C_1}{\mu_1} & 0 \\ \frac{\mu_4L_{12}S_1 - \mu_4\mu_1C_1}{-\mu_1\mu_2\mu_4 + \mu_3\mu_1^2} & \frac{-\mu_4L_{12}C_1 - \mu_4\mu_1S_1}{-\mu_1\mu_2\mu_4 + \mu_3\mu_1^2} & \frac{\mu_3}{-\mu_2\mu_4 + \mu_3\mu_1} \\ \frac{-L_{12}S_1 + \mu_1C_1}{-\mu_2\mu_4 + \mu_3\mu_1} & \frac{L_{12}C_1 + \mu_1S_1}{-\mu_2\mu_4 + \mu_3\mu_1} & -\frac{\mu_2}{-\mu_2\mu_4 + \mu_3\mu_1} \end{pmatrix} \quad (C.4)$$

## C.3 Singularità cinematiche

Le singularità cinematiche del *SoftArm* sono .

- ▷  $q_3 = \pi$ : non è significativo perché il *SoftArm* ha un'escursione massima per  $q_3$  pari a  $\frac{\pi}{2}$ , configurazione in cui il braccio è piegato su se stesso.
- ▷  $q_3 = 0$ : è interessante perché si tratta di una singularità ai confini dello spazio di lavoro raggiungibile. Questo punto nello spazio delle variabili di giunto non costituisce però un problema a livello pratico,

purché si assuma che  $q_3$  non possa diventare negativo. In ogni caso la manipolabilità del braccio, nell'intorno di questa configurazione, sarà molto scarsa.

- ▷ termini del tipo  $q_k = \arctan(\mathbf{a} + j\mathbf{b})$ : non hanno evidentemente un significato fisico.
- ▷  $q_2 = \pm \arctan \left\{ \frac{[L_{23} + 2L_3 \cos(q_3)] \sin(q_3)}{2L_3 \sin^2(q_3) - L_2 - L_{23} \cos(q_3) - L_3} \right\}$ : singolarità di spalla tipica dei manipolatori antropomorfi, che si verifica quando l'end-effector giace sull'asse di rotazione del primo link.

## C.4 Dinamica

Viene considerata la dinamica del manipolatore espressa nella forma

$$B(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \tau_{nc}, \quad (\text{C.5})$$

### C.4.1 Matrice $B$

$$B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{12} & b_{22} & b_{23} \\ b_{13} & b_{23} & b_{33} \end{pmatrix}. \quad (\text{C.6})$$

$$\begin{aligned} b_{11} = & m_2 L_{12}^2 + m_{ee} L_{12}^2 + m_3 L_3 S_{233} L_2 S_2 + m_3 L_{12}^2 + 2m_3 L_2 S_2 L_{23} S_{23} + \\ & + \frac{2}{3} m_{12} L_{12}^2 - m_3 L_{23}^2 C_{22}^2 - m_3 L_2^2 C_2^2 - \frac{2}{3} m_3 L_3^2 C_{233}^2 + 2m_{ee} L_2 S_2 L_{23} S_{23} + \\ & + \frac{2}{3} m_2 L_2^2 + m_3 L_3 S_{233} L_{23} S_{23} + 2m_{ee} L_{23} S_{23} L_3 S_{233} + 2m_{ee} L_2 S_2 L_3 S_{233} + \\ & + m_3 L_2^2 + \frac{2}{3} m_3 L_3^2 + m_3 L_{23}^2 - \frac{2}{3} m_2 L_2^2 C_2^2 + m_{ee} L_2^2 + m_{ee} L_{23}^2 + \\ & - m_{ee} L_2^2 C_2^2 - m_{ee} L_{23}^2 C_{22}^2 - m_{ee} L_3^2 C_{233}^2 + m_{ee} L_3^2 - m_{23} L_{23}^2 C_{2233} + \\ & - \frac{1}{2} m_{23} L_2^2 C_{22} + \frac{1}{4} m_{23} L_{23} L_2 C_{22-3} - \frac{1}{4} m_{23} L_{23} L_2 C_{3-2-2} + m_{23} L_{12}^2 + \\ & + \frac{1}{6} m_{23} L_{23}^2 + \frac{1}{2} m_{23} L_2^2 - \frac{1}{2} m_{23} L_{23} L_2 C_{223} + \frac{1}{2} m_{23} L_{23} L_2 C_3 \end{aligned}$$

$$b_{12} = -m_{23} L_{12} L_2 C_2 - \frac{1}{2} m_{23} L_{23} L_{12} C_{22} - \frac{1}{2} L_{12} L_2 C_2 m_2 - \frac{1}{2} m_3 L_3 C_{233} L_{12} + \\ -m_3 L_2 C_2 L_{12} - m_3 L_{23} C_{22} L_{12} - m_{ee} L_2 C_2 L_{12} - m_{ee} L_{23} C_{22} L_{12} \\ -m_{ee} L_3 C_{233} L_{12}$$

$$b_{13} = -m_3 L_3 C_{233} L_{12} - m_3 L_{23} C_{22} L_{12} - \frac{1}{2} m_{23} L_{23} L_{12} C_{22} - m_{ee} L_{23} C_{22} L_{12} + \\ -2 m_{ee} L_3 C_{233} L_{12}$$

$$b_{21} = b_{12}$$

$$b_{22} = \frac{2}{3} m_3 L_3^2 + \frac{2}{3} m_{23} L_{23}^2 + \frac{1}{2} m_{23} L_2^2 + \frac{2}{3} L_2^2 m_2 + m_{23} L_{23} L_2 C_3 + 2m_{ee} L_2 S_2 L_{23} S_{23} + \\ +2m_{ee} L_2 S_2 L_3 S_{233} + 2m_{ee} L_{23} S_{23} L_3 S_{233} + 2m_3 L_2 S_2 L_{23} S_{23} + m_3 L_3 S_{233} L_2 S_2 + \\ +m_3 L_3 S_{233} L_{23} S_{23} + m_{ee} L_3^2 + \frac{1}{2} m_{ee} L_{23}^2 + 2m_3 L_{23} C_{22} L_2 C_2 + \\ +2m_{ee} L_2 C_2 L_3 C_{233} + m_3 L_3 C_{233} L_{23} C_{22} + m_3 L_3 C_{233} L_2 C_2 + \\ +2m_{ee} L_{23} C_{22} L_2 C_2 + m_{ee} L_{23} C_{22} L_3 C_{233} + m_3 L_{23}^2 + m_3 L_2^2 + m_{ee} L_2^2$$

$$b_{23} = \frac{1}{3} m_{23} L_{23}^2 + \frac{1}{2} m_{23} L_{23} L_2 C_3 + \frac{2}{3} m_3 L_3^2 + 2 m_{ee} L_3 S_{233} L_2 S_2 + m_3 L_{23} S_{23} L_2 S_2 + \\ +m_{ee} L_{23} S_{23} L_2 S_2 + 3 m_{ee} L_{23} S_{23} L_3 S_{233} + m_3 L_3 S_{233} L_2 S_2 + \\ +m_{ee} L_{23}^2 + 2 m_{ee} L_3^2 + m_3 L_3 C_{233} L_2 C_2 + 3 m_{ee} L_{23} C_{22} L_3 C_{233} + \\ +m_{ee} L_{23} C_{22} L_2 C_2 + m_3 L_{23}^2 + m_3 L_{23} C_{22} L_2 C_2 + \frac{3}{2} m_3 L_3 L_{23} C_{22} C_{233} + \\ +2 m_{ee} L_3 C_{233} L_2 C_2 + \frac{3}{2} m_3 L_3 L_{23} S_{23} S_{233}$$

$$b_{31} = b_{13}$$

$$b_{32} = b_{23}$$

$$b_{33} = m_3 L_{23}^2 + 2m_3 L_3 L_{23} S_{23} S_{233} + 4 m_{ee} L_{23} S_{23} L_3 S_{233} + \frac{4}{3} m_3 L_3^2 + m_{ee} L_{23}^2 + \\ +4 m_{ee} L_3^2 + \frac{2}{3} m_{23} L_{23}^2 + 4 m_{ee} L_{23} C_{22} L_3 C_{233} + 2m_3 L_3 L_{23} C_{22} C_{233}$$

## C.4.2 Matrice C

$$C = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{12} & c_{22} & c_{23} \\ c_{13} & c_{23} & c_{33} \end{pmatrix}. \quad (\text{C.7})$$

$$\begin{aligned}
c_{11} = & \left( \frac{1}{4} m_3 L_{23}^2 S_{2233} + \frac{1}{12} m_3 L_3^2 S_{223333} + \frac{1}{2} m_{ee} L_{23} L_3 S_{2233} + \frac{1}{2} m_{ee} L_3 L_2 S_{2233} + \right. \\
& + \frac{1}{2} m_{ee} L_{23} L_2 S_{223} + \frac{1}{12} m_{23} L_{23}^2 S_{2233} - \frac{1}{8} m_{23} L_{23} L_2 S_{22-3} + \\
& - \frac{1}{8} m_{23} L_{23} L_2 S_{3-2-2} + \frac{1}{4} m_{23} L_{23} L_2 S_{223} + \frac{1}{4} m_3 L_2^2 S_{22} + \frac{1}{4} m_{23} L_2^2 S_{22} + \\
& + \frac{1}{2} m_3 L_{23} L_2 S_{223} + \frac{1}{12} m_2 L_2^2 S_{22} + \frac{1}{4} m_{ee} L_2^2 S_{22} + \frac{1}{4} m_{ee} L_{23}^2 S_{2233} + \\
& + \frac{1}{4} m_3 L_3 L_{23} S_{2233} + \frac{1}{4} m_3 L_3 L_2 S_{2233} + \frac{1}{4} m_{ee} L_3^2 S_{223333} \Big) \dot{q}_2 + \\
& + \left( \frac{1}{4} m_{ee} L_{23}^2 S_{2233} + \frac{1}{2} m_{ee} L_3^2 S_{223333} + \frac{1}{3} m_3 L_3^2 S_{223333} + \frac{1}{4} m_3 L_{23}^2 S_{2233} + \right. \\
& + \frac{1}{4} m_3 L_{23} L_2 S_{223} - \frac{1}{4} m_3 L_{23} L_2 S_3 - \frac{1}{2} m_{ee} L_3 L_2 S_{33} + \frac{1}{2} m_{ee} L_3 L_2 S_{2233} + \\
& + \frac{1}{4} m_{ee} L_{23} L_2 S_{223} - \frac{1}{4} m_{ee} L_{23} L_2 S_3 - \frac{1}{8} m_3 L_3 L_{23} S_3 + \frac{3}{8} m_3 L_3 L_{23} S_{2233} + \\
& - \frac{1}{8} m_{23} L_{23} L_2 S_3 - \frac{1}{4} m_3 L_3 L_2 S_{33} + \frac{1}{4} m_3 L_3 L_2 S_{2233} + \frac{1}{12} m_{23} L_{23}^2 S_{2233} + \\
& + \frac{1}{16} m_{23} L_{23} L_2 S_{22-3} + \frac{1}{16} m_{23} L_{23} L_2 S_{3-2-2} + \frac{1}{8} m_{23} L_{23} L_2 S_{223} + \\
& \left. - \frac{1}{4} m_{ee} L_{23} L_3 S_3 + 3/4 m_{ee} L_{23} L_3 S_{2233} \right) \dot{q}_3 \\
c_{12} = & \left( \frac{1}{4} m_3 L_{23}^2 S_{2233} + \frac{1}{12} m_3 L_3^2 S_{223333} + \frac{1}{2} m_{ee} L_{23} L_3 S_{2233} + \frac{1}{2} m_{ee} L_3 L_2 S_{2233} + \right. \\
& + \frac{1}{2} m_{ee} L_{23} L_2 S_{223} + \frac{1}{12} m_{23} L_{23}^2 S_{2233} - \frac{1}{8} m_{23} L_{23} L_2 S_{22-3} - \frac{1}{8} m_{23} L_{23} L_2 S_{3-2-2} + \\
& + \frac{1}{4} m_{23} L_{23} L_2 S_{223} + \frac{1}{4} m_3 L_2^2 S_{22} + \frac{1}{4} m_{23} L_2^2 S_{22} + \frac{1}{2} m_3 L_{23} L_2 S_{223} + \frac{1}{12} m_2 L_2^2 S_{22} + \\
& + \frac{1}{4} m_{ee} L_2^2 S_{22} + \frac{1}{4} m_{ee} L_{23}^2 S_{2233} + \frac{1}{4} m_3 L_3 L_{23} S_{2233} + \frac{1}{4} m_3 L_3 L_2 S_{2233} + \\
& + \frac{1}{4} m_{ee} L_3^2 S_{223333} \Big) \dot{q}_1 + \frac{1}{2} L_{12} (2 m_{23} L_2 S_2 + m_{23} L_{23} S_{23} + L_2 S_2 m_2 + m_3 L_3 S_{233} + \\
& + 2 m_3 L_2 S_2 + 2 m_3 L_{23} S_{23} + 2 m_{ee} L_2 S_2 + 2 m_{ee} L_{23} S_{23} + 2 m_{ee} L_3 S_{233}) \dot{q}_2 + \\
& + \frac{1}{2} L_{12} (2 m_3 L_3 S_{233} + 2 m_3 L_{23} S_{23} + m_{23} L_{23} S_{23} + 2 m_{ee} L_{23} S_{23} + 4 m_{ee} L_3 S_{233}) \dot{q}_3
\end{aligned}$$

$$\begin{aligned}
c_{13} &= \left( \frac{1}{4} m_{ee} L_{23}^2 \mathbf{S}_{2233} + \frac{1}{2} m_{ee} L_3^2 \mathbf{S}_{223333} + \frac{1}{3} m_3 L_3^2 \mathbf{S}_{223333} + \frac{1}{4} m_3 L_{23}^2 \mathbf{S}_{2233} + \right. \\
&+ \frac{1}{4} m_3 L_{23} L_2 \mathbf{S}_{223} - \frac{1}{4} m_3 L_{23} L_2 \mathbf{S}_3 - \frac{1}{2} m_{ee} L_3 L_2 \mathbf{S}_{33} + \frac{1}{2} m_{ee} L_3 L_2 \mathbf{S}_{2233} + \\
&+ \frac{1}{4} m_{ee} L_{23} L_2 \mathbf{S}_{223} - \frac{1}{4} m_{ee} L_{23} L_2 \mathbf{S}_3 - \frac{1}{8} m_3 L_3 L_{23} \mathbf{S}_3 + 3/8 m_3 L_3 L_{23} \mathbf{S}_{2233} + \\
&- \frac{1}{8} m_{23} L_{23} L_2 \mathbf{S}_3 - \frac{1}{4} m_3 L_3 L_2 \mathbf{S}_{33} + \frac{1}{4} m_3 L_3 L_2 \mathbf{S}_{2233} + \frac{1}{12} m_{23} L_{23}^2 \mathbf{S}_{2233} \\
&+ \frac{1}{16} m_{23} L_{23} L_2 \mathbf{S}_{22-3} + \frac{1}{16} m_{23} L_{23} L_2 \mathbf{S}_{3-2-2} + \frac{1}{8} m_{23} L_{23} L_2 \mathbf{S}_{223} - \frac{1}{4} m_{ee} L_{23} L_3 \mathbf{S}_3 \\
&+ \frac{3}{4} m_{ee} L_{23} L_3 \mathbf{S}_{2233} \Big) \dot{q}_1 + \frac{1}{2} L_{12} (2 m_3 L_3 \mathbf{S}_{233} + 2 m_3 L_{23} \mathbf{S}_{23} + m_{23} L_{23} \mathbf{S}_{23} + \\
&+ 2 m_{ee} L_{23} \mathbf{S}_{23} + 4 m_{ee} L_3 \mathbf{S}_{233}) \dot{q}_2 + \frac{1}{2} L_{12} (4 m_3 L_3 \mathbf{S}_{233} + 2 m_3 L_{23} \mathbf{S}_{23} + \\
&+ m_{23} L_{23} \mathbf{S}_{23} + 2 m_{ee} L_{23} \mathbf{S}_{23} + 8 m_{ee} L_3 \mathbf{S}_{233}) \dot{q}_3 \\
c_{21} &= \left( -\frac{1}{4} m_3 L_{23}^2 \mathbf{S}_{2233} - \frac{1}{12} m_3 L_3^2 \mathbf{S}_{223333} - \frac{1}{2} m_{ee} L_{23} L_3 \mathbf{S}_{2233} - \frac{1}{2} m_{ee} L_3 L_2 \mathbf{S}_{2233} + \right. \\
&- \frac{1}{2} m_{ee} L_{23} L_2 \mathbf{S}_{223} - \frac{1}{12} m_{23} L_{23}^2 \mathbf{S}_{2233} + \frac{1}{8} m_{23} L_{23} L_2 \mathbf{S}_{22-3} + \\
&- \frac{1}{4} m_{23} L_{23} L_2 \mathbf{S}_{223} - \frac{1}{4} m_3 L_2^2 \mathbf{S}_{22} - \frac{1}{4} m_{23} L_2^2 \mathbf{S}_{22} - \frac{1}{2} m_3 L_{23} L_2 \mathbf{S}_{223} + \\
&- \frac{1}{12} m_2 L_2^2 \mathbf{S}_{22} - \frac{1}{4} m_{ee} L_2^2 \mathbf{S}_{22} - \frac{1}{4} m_{ee} L_{23}^2 \mathbf{S}_{2233} - \frac{1}{4} m_3 L_3 L_{23} \mathbf{S}_{2233} + \\
&- \frac{1}{4} m_3 L_3 L_2 \mathbf{S}_{2233} - \frac{1}{4} m_{ee} L_3^2 \mathbf{S}_{223333} \Big) \dot{q}_1 + \frac{1}{8} m_{23} L_{23} L_2 \mathbf{S}_{3-2-2} \\
c_{22} &= \left( -\frac{1}{4} m_{23} L_{23} L_2 \mathbf{S}_3 - \frac{1}{2} m_{ee} L_{23} L_2 \mathbf{S}_3 - m_{ee} L_3 L_2 \mathbf{S}_{33} - \frac{1}{2} m_{ee} L_{23} L_3 \mathbf{S}_3 + \right. \\
&- \frac{1}{2} m_3 L_{23} L_2 \mathbf{S}_3 - \frac{1}{4} m_3 L_3 L_{23} \mathbf{S}_3 - \frac{1}{2} m_3 L_3 L_2 \mathbf{S}_{33} \Big) \dot{q}_3 \\
c_{23} &= \left( -\frac{1}{4} m_{23} L_{23} L_2 \mathbf{S}_3 - \frac{1}{2} m_{ee} L_{23} L_2 \mathbf{S}_3 - m_{ee} L_3 L_2 \mathbf{S}_{33} + \right. \\
&- \frac{1}{2} m_{ee} L_{23} L_3 \mathbf{S}_3 - \frac{1}{2} m_3 L_{23} L_2 \mathbf{S}_3 - \frac{1}{4} m_3 L_3 L_{23} \mathbf{S}_3 - \frac{1}{2} m_3 L_3 L_2 \mathbf{S}_{33} \Big) \dot{q}_2 + \\
&+ (-4 m_{ee} L_3 L_2 \mathbf{S}_{33} - m_3 L_{23} L_2 \mathbf{S}_3 - m_{ee} L_{23} L_2 \mathbf{S}_3 - 3 m_{ee} L_{23} L_3 \mathbf{S}_3 + \\
&- 2 m_3 L_3 L_2 \mathbf{S}_{33} - \frac{3}{2} m_3 L_3 L_{23} \mathbf{S}_3 - \frac{1}{2} m_{23} L_{23} L_2 \mathbf{S}_3) \dot{q}_3
\end{aligned}$$

$$\begin{aligned}
c_{31} &= \left( -\frac{1}{4} m_{ee} L_{23}^2 \mathbf{S}_{2233} - \frac{1}{2} m_{ee} L_3^2 \mathbf{S}_{223333} - \frac{1}{3} m_3 L_3^2 \mathbf{S}_{223333} - \frac{1}{4} m_3 L_{23}^2 \mathbf{S}_{2233} + \right. \\
&\quad -\frac{1}{4} m_3 L_{23} L_2 \mathbf{S}_{223} + \frac{1}{4} m_3 L_{23} L_2 \mathbf{S}_3 + \frac{1}{2} m_{ee} L_3 L_2 \mathbf{S}_{33} - \frac{1}{2} m_{ee} L_3 L_2 \mathbf{S}_{2233} + \\
&\quad -\frac{1}{4} m_{ee} L_{23} L_2 \mathbf{S}_{223} + \frac{1}{4} m_{ee} L_{23} L_2 \mathbf{S}_3 + \frac{1}{8} m_3 L_3 L_{23} \mathbf{S}_3 - 3/8 m_3 L_3 L_{23} \mathbf{S}_{2233} + \\
&\quad + \frac{1}{8} m_{23} L_{23} L_2 \mathbf{S}_3 + \frac{1}{4} m_3 L_3 L_2 \mathbf{S}_{33} - \frac{1}{4} m_3 L_3 L_2 \mathbf{S}_{2233} - \frac{1}{12} m_{23} L_{23}^2 \mathbf{S}_{2233} + \\
&\quad - \frac{1}{16} m_{23} L_{23} L_2 \mathbf{S}_{22-3} - 1/16 m_{23} L_{23} L_2 \mathbf{S}_{3-2-2} - \frac{1}{8} m_{23} L_{23} L_2 \mathbf{S}_{223} + \\
&\quad \left. + \frac{1}{4} m_{ee} L_{23} L_3 \mathbf{S}_3 - \frac{3}{4} m_{ee} L_{23} L_3 \mathbf{S}_{2233} \right) \dot{q}_1 \\
c_{32} &= \left( \frac{1}{4} m_{23} L_{23} L_2 \mathbf{S}_3 + \frac{1}{2} m_{ee} L_{23} L_2 \mathbf{S}_3 + m_{ee} L_3 L_2 \mathbf{S}_{33} + \frac{1}{2} m_{ee} L_{23} L_3 \mathbf{S}_3 + \right. \\
&\quad \left. + \frac{1}{2} m_3 L_{23} L_2 \mathbf{S}_3 + \frac{1}{4} m_3 L_3 L_{23} \mathbf{S}_3 + \frac{1}{2} m_3 L_3 L_2 \mathbf{S}_{33} \right) \dot{q}_2 \\
c_{33} &= \left( -\frac{1}{2} m_3 L_3 L_{23} \mathbf{S}_3 - m_{ee} L_{23} L_3 \mathbf{S}_3 \right) \dot{q}_3
\end{aligned}$$

### C.4.3 Vettore G

$$\begin{aligned}
g_1 &= 0 \\
g_2 &= \left( \frac{1}{2} m_2 L_2 \mathbf{S}_2 + m_{23} (L_2 \mathbf{S}_2 + \frac{1}{2} L_{23} \mathbf{S}_{23}) + m_3 (L_2 \mathbf{S}_2 + L_{23} \mathbf{S}_{23} + \frac{1}{2} L_3 \mathbf{S}_{233} + \right. \\
&\quad \left. + m_{ee} (L_2 \mathbf{S}_2 + L_{23} \mathbf{S}_{23} + L_3 \mathbf{S}_{233}) \right) g \\
g_3 &= \left( \frac{1}{2} m_{23} L_{23} \mathbf{S}_{23} + m_3 (L_{23} \mathbf{S}_{23} + L_3 \mathbf{S}_{233} + m_{ee} (L_{23} \mathbf{S}_{23} + 2 L_3 \mathbf{S}_{233})) \right) g
\end{aligned}$$

# Bibliografia

- [1] Jean-Claude Latombe, *Robot Motion Planning*, Kluwer Academic Publishers
- [2] Y.Ma, S.Soatto, J.Košěná, S.S.Sastry *An Invitation to 3-D Vision From Image to Geometric Models*, Springer
- [3] Hartley R.,Zisserman *Multiple View Geometry in Computer Vision*,2nd Edition Cambridge University Press
- [4] Sciavicco, Siciliano, *Robotica Industriale, Modellistica e controllo dei manipolatori*, McGraw-Hill.
- [5] Lowe, D. G. *Distinctive Image Features from Scale- Invariant Keypoints*. International Journal of Computer Vision Vol. 60(2), p.9. 2004.
- [6] *Measurement and modeling of McKibben pneumatic artificial muscles*, IEEE, Transactions on robotic and automation, pag. 90-102, vol. 12, No 1, Febbraio 1996.
- [7] Gerhard X. Ritter , Joseph N. Wilson *Handbook of Computer Vision Algorithms in Image Algebra*, 2nd Edition CRC Press
- [8] Novak J.L., Feddema J.T.: *A Capacitance-Based Proximity Sensor for Whole Arm Obstacle Avoidance.*, IEEE Proceedings of the Intl. Conf. on Robotics and Automation, pp. 1307-1314, 1992.
- [9] Feddema J.T., Novak J.L.: *Whole Arm Obstacle Avoidance for Teleoperated Robots.*, IEEE Robotics and Automation Proceedings, pp.3303 - 3309, 1994.
- [10] Noborio H., Nishino Y.: *Image-based Path-Planning Algorithm on the Joint Space.*, IEEE International Conference on Robotics and Automation, pp. 1180- 1187,Seoul, 2001.



- 
- [11] jih-Gau Juang : *Application of Repulsive Force and Generic Algorithm to Multi-manipulator Collision Avoidance*, National Taiwan Ocean University
- [12] Kuhn S., Gecks T. and Henrich D.: *Velocity control for safe robot guidance based on fused vision and force/torque data*, IEEE Conference on Multisensor Fusion and Integration, September 3-6, 2006, Heidelberg, Germany
- [13] Gecks T., Henrich D.: *Human-robot cooperation: Safe Pick-and-Place Operations*, IEEE International Workshop on Robots and Human Interactive Communication, 2005
- [14] Iossifidis I., Schöner G. *Dynamic System Approach for the Autonomous Avoidance of Obstacle and Joint-limit for an Redundant Robot Arm*, in proceeding of the IEEE International Conference on Intelligent Robots and System, Beijing, China, 2006