

POLITECNICO DI TORINO

Corso di Laurea magistrale in

Ingegneria Meccanica

Tesi di Laurea Magistrale

**Analisi e sperimentazione della cinematica inversa
di bracci robotici 6R e 7R**



Relatore/i:

Prof. Stefano Paolo Pastorelli

Ing. Leonardo Sabatino Scimmi

Candidato:

Didier Squinabol

Anno Accademico 2018/2019

Il progresso è la capacità dell'uomo di complicare la semplicità.

(Thor Heyerdahl)

Ai miei nonni

che, nella loro genuina e preziosa semplicità,

hanno sempre creduto nel mio progresso.

INDICE

INDICE	I
INDICE DELLE FIGURE	III
INDICE DELLE TABELLE.....	VII
INTRODUZIONE.....	1
CAPITOLO 1: Robot ABB IRB120	3
1.1 Modellizzazione del robot.....	3
1.2 Cinematica inversa	6
1.3 Analisi della molteplicità di soluzioni di braccio	10
1.4 Analisi della destrezza.....	16
1.5 Pianificazione di traiettoria Resolved Rate Motion Control	30
CAPITOLO 2: Robot Kinova Jaco 2 7DOF	41
2.1 Modellazione del robot.....	43
2.2 Rappresentazione virtuale del robot.....	46
2.3 Analisi del Workspace	50
2.4 Modello semplificato di un manipolatore a 7 gradi di libertà.....	51
2.5 Sistema di riferimento per lo studio della ridondanza.....	54
2.6 Cinematica inversa	63
2.7 Limiti di giunto.....	77
CAPITOLO 3: Analisi di traiettorie	83
CAPITOLO 4: Sperimentazione.....	98
4.1 Metodi di controllo del Kinova Jaco 2	98
4.2 Setup per la sperimentazione.....	101
4.3 Funzioni utilizzate	103
4.4 Comando in posizione.....	105

4.5. Comando in velocità.....	109
CONCLUSIONE	116
RINGRAZIAMENTI.....	117
BIBLIOGRAFIA	118

INDICE DELLE FIGURE

FIGURA 1.1 ROBOT ABB IRB120 [14].	3
FIGURA 1.2 SISTEMI DI RIFERIMENTO DENAVIT-HARTENBERG.	4
FIGURA 1.3 ANDAMENTO DEL NUMERO DI SOLUZIONI NELLO SPAZIO OPERATIVO (A) E VARIANDO q_2 E q_3 (B) CON $q_1 = 0^\circ$.	11
FIGURA 1.4 DUE CONFIGURAZIONI DEL ROBOT PER RAGGIUNGERE LA MEDESIMA POSIZIONE DEL CENTRO POLSO.	11
FIGURA 1.5 ANDAMENTO DEL NUMERO DI SOLUZIONI NELLO SPAZIO OPERATIVO (A) E VARIANDO q_2 E q_3 (B) CON $q_1 = 90^\circ$.	12
FIGURA 1.6 SOVRAPPOSIZIONE DELLA DISTRIBUZIONE DEL NUMERO DI SOLUZIONI PER DUE CONFIGURAZIONI DIAMETRALMENTE OPPOSTE.	13
FIGURA 1.7 QUATTRO CONFIGURAZIONI DEL ROBOT PER RAGGIUNGERE LA MEDESIMA POSIZIONE DEL CENTRO POLSO.	13
FIGURA 1.8 DISTRIBUZIONE DEL MASSIMO NUMERO DI SOLUZIONI AL VARIARE DI q_1 .	14
FIGURA 1.9 POSIZIONI POSSIBILI DELL'END EFFECTOR DATA UNA DETERMINATA POSA DEL CENTRO POLSO.	16
FIGURA 1.10 ANDAMENTO DEL VOLUME E DELLA SUPERFICIE NORMALIZZATI IN FUNZIONE DI q_5 .	18
FIGURA 1.11 VARIAZIONE DELLA CALOTTA SFERICA AL VARIARE DI q_5 .	18
FIGURA 1.12 SOVRAPPOSIZIONE DEI PUNTI EFFETTIVAMENTE RAGGIUNGIBILI DALL'END EFFECTOR LEGATA AD UNA DIVERSA SOLUZIONE DI BRACCIO PER PORTARE IL CENTRO POLSO NELLA MEDESIMA POSIZIONE.	19
FIGURA 1.13 CASO PARTICOLARE IN CUI LA SOVRAPPOSIZIONE GENERA UNA SFERA COMPLETA.	19
FIGURA 1.14 DUE CONFIGURAZIONI CHE PRODUCONO LA SFERA DI DESTREZZA VISIBILE IN FIGURA 1.13.	20
FIGURA 1.15 PUNTI RAGGIUNGIBILI DALL'END EFFECTOR DATA UNA POSIZIONE DEL CENTRO POLSO CON INDICAZIONE DI DUE SISTEMI DI RIFERIMENTO DI INTERESSE.	22
FIGURA 1.16 SOVRAPPOSIZIONE DEI PUNTI EFFETTIVAMENTE RAGGIUNGIBILI DALL'END EFFECTOR LEGATI ALL'ESISTENZA DI DUE DIVERSE CONFIGURAZIONI DI BRACCIO CHE PRODUCONO LA STESSA POSIZIONE DEL CENTRO POLSO.	23
FIGURA 1.17 SOVRAPPOSIZIONE DEI PUNTI EFFETTIVAMENTE RAGGIUNGIBILI DALL'END EFFECTOR LEGATI ALL'ESISTENZA DI TRE DIVERSE CONFIGURAZIONI DI BRACCIO CHE PRODUCONO LA STESSA POSIZIONE DEL CENTRO POLSO.	24
FIGURA 1.18 GRAFICO 3D (A) E MAPPA COLORATA (B) CON ANDAMENTO INDICE DI DESTREZZA AL VARIARE DEI GRADI DI LIBERTÀ q_2 E q_3 E MANTENENDO IL GRADO DI LIBERTÀ $q_1 = 0^\circ$.	26
FIGURA 1.19 GRAFICO 3D (A) E MAPPA COLORATA (B) CON ANDAMENTO INDICE DI DESTREZZA AL VARIARE DELLA POSIZIONE DEL CENTRO POLSO SU UN PIANO CON $q_1 = 0^\circ$.	27
FIGURA 1.20 GRAFICO 3D (A) E MAPPA COLORATA (B) CON ANDAMENTO INDICE DI DESTREZZA AL VARIARE DEI GRADI DI LIBERTÀ q_2 E q_3 E MANTENENDO IL GRADO DI LIBERTÀ $q_1 = 90^\circ$.	28
FIGURA 1.21 GRAFICO 3D (A) E MAPPA COLORATA (B) CON ANDAMENTO INDICE DI DESTREZZA AL VARIARE DELLA POSIZIONE DEL CENTRO POLSO SU UN PIANO CON $q_1 = 90^\circ$.	29
FIGURA 1.22 SCHEMA SIMULINK DELL'ALGORITMO RRM C LOGICA CLOSED.	30
FIGURA 1.23 TRAIETTORIA END EFFECTOR DESIDERATA.	31
FIGURA 1.24 CONFRONTO TRA LA TRAIETTORIA OTTENUTA (IN BLU) E LA TRAIETTORIA DESIDERATA (IN ROSSO).	31
FIGURA 1.25 ANDAMENTO DETERMINANTE MATRICE JACOBIANA.	32
FIGURA 1.26 TRAIETTORIA END EFFECTOR DESIDERATA PER PRODURRE SINGOLARITÀ.	32
FIGURA 1.27 ANDAMENTO DEI VALORI DI GRADI DI LIBERTÀ NEI GIUNTI (A) E DELLE VELOCITÀ (B) NEL CASO DI SINGOLARITÀ.	33
FIGURA 1.28 ANDAMENTO DETERMINANTE MATRICE JACOBIANA.	34
FIGURA 1.29 SCHEMA SIMULINK DELL'ALGORITMO RRM C CLOSED LOOP CON LIMITI DI VELOCITÀ NEI GIUNTI.	35
FIGURA 1.30 ANDAMENTO DELLE VELOCITÀ NEI GIUNTI NEL CASO DI SINGOLARITÀ IN CUI SI CONSIDERANO I LIMITI DI VELOCITÀ DEI GIUNTI.	35

FIGURA 1.31 ANDAMENTO DEI GRADI DI LIBERTÀ NEI GIUNTI NEL CASO DI SINGOLARITÀ IN CUI SI CONSIDERANO I LIMITI DI VELOCITÀ DEI GIUNTI.....	36
FIGURA 1.32 CONFRONTO TRA LA TRAIETTORIA OTTENUTA (IN BLU) E LA TRAIETTORIA DESIDERATA (IN ROSSO) NEL CASO DI SINGOLARITÀ IN CUI SI CONSIDERANO I LIMITI DI VELOCITÀ NEI GIUNTI.....	36
FIGURA 1.33 CONFRONTO TRA DUE CONFIGURAZIONI NELLO STESSO ISTANTE NEL CASO IN CUI NON SI CONSIDERANO I LIMITI DI VELOCITÀ (A) E NEL CASO IN CUI SI CONSIDERANO TALI LIMITI (B) IN PRESENZA DI SINGOLARITÀ DI BRACCIO.	37
FIGURA 1.34 TRAIETTORIA END EFFECTOR DESIDERATA PER PRODURRE SINGOLARITÀ.....	37
FIGURA 1.35 ANDAMENTO DETERMINANTE MATRICE JACOBIANA SINGOLARITÀ DI POLSO.	38
FIGURA 1.36 ANDAMENTO DEI VALORI DI GRADI DI LIBERTÀ NEI GIUNTI NEL CASO DI SINGOLARITÀ DI POLSO.	38
FIGURA 1.37 ANDAMENTO DELLE VELOCITÀ NEI GIUNTI NEL CASO DI SINGOLARITÀ DI POLSO.	39
FIGURA 1.38 ANDAMENTO DELLE VELOCITÀ NEI GIUNTI NEL CASO DI SINGOLARITÀ DI POLSO IN CUI SI CONSIDERANO I LIMITI DI VELOCITÀ DEI GIUNTI.....	39
FIGURA 1.39 CONFRONTO TRA LA TRAIETTORIA OTTENUTA (IN BLU) E LA TRAIETTORIA DESIDERATA (IN ROSSO) NEL CASO DI SINGOLARITÀ DI POLSO IN CUI SI CONSIDERANO I LIMITI DI VELOCITÀ NEI GIUNTI.	40
FIGURA 1.40 CONFRONTO TRA DUE CONFIGURAZIONI NELLO STESSO ISTANTE NEL CASO IN CUI NON SI CONSIDERANO I LIMITI DI VELOCITÀ (A) E NEL CASO IN CUI SI CONSIDERANO TALI LIMITI (B) IN PRESENZA DI SINGOLARITÀ DI POLSO.	40
FIGURA 2.1 KINOVA JACO 2 7DOF [6].....	41
FIGURA 2.2 QUOTE GEOMETRICHE, ASSI E DIREZIONE DI ROTAZIONE GIUNTI KINOVA JACO 2 [6].	42
FIGURA 2.3 SISTEMI DI RIFERIMENTO DENAVIT-HARTENBERG.	43
FIGURA 2.4 MODELLO CAD DEI LINK [13] CON RISPETTIVA MATRICE DI TRASFORMAZIONE TRA SISTEMA DI RIFERIMENTO DENAVIT-HARTENBERG E SISTEMA DI RIFERIMENTO CAD.	48
FIGURA 2.5 RAPPRESENTAZIONE REALISTICA DEL ROBOT TRAMITE AMBIENTE SIMULINK IN DUE CONFIGURAZIONI DIVERSE.	49
FIGURA 2.6 POSIZIONI RAGGIUNGIBILI DAL CENTRO POLSO DEL ROBOT FACENDO VARIARE q_2 E q_4 MANTENENDO $q_1 = 0^\circ$	50
FIGURA 2.7 POSIZIONI RAGGIUNGIBILI DALL'END EFFECTOR DATA UNA POSIZIONE FISSA DEL CENTRO POLSO.	50
FIGURA 2.8 RAPPRESENTAZIONE SCHEMATICA DEL MANIPOLATORE.	52
FIGURA 2.9 VISTE SU DUE PIANI ORTOGONALI DEL MODELLO GEOMETRICO DEL ROBOT.	53
FIGURA 2.10 RAPPRESENTAZIONE DEL SISTEMA DI RIFERIMENTO AUSILIARIO R.	54
FIGURA 2.11 TRASLAZIONE DEL SISTEMA DI RIFERIMENTO PER OTTENERE IL SISTEMA DI RIFERIMENTO $X^1-Y^1-Z^1$	55
FIGURA 2.12 ROTAZIONE DEL SISTEMA DI RIFERIMENTO $X^1-Y^1-Z^1$ PER OTTENERE IL SISTEMA DI RIFERIMENTO $X^2-Y^2-Z^2$	56
FIGURA 2.13 ROTAZIONE DEL SISTEMA DI RIFERIMENTO $X^2-Y^2-Z^2$ PER OTTENERE IL SISTEMA DI RIFERIMENTO $X^3-Y^3-Z^3$	57
FIGURA 2.14 ROTAZIONE DEL SISTEMA DI RIFERIMENTO $X^3-Y^3-Z^3$ PER OTTENERE IL SISTEMA DI RIFERIMENTO R CERCATO.....	60
FIGURA 2.15 RAPPRESENTAZIONE STICK FIGURE DEL ROBOT CON CIRCONFERENZA DI SWIVEL PER UNA DATA POSA DELL'END EFFECTOR.	61
FIGURA 2.16 RAPPRESENTAZIONE TRAMITE V-REP DEL ROBOT CON CIRCONFERENZA DI SWIVEL PER UNA DATA POSA DELL'END EFFECTOR.	62
FIGURA 2.17 SCHEMATIZZAZIONE DEL BRACCIO DEL ROBOT CON VERSORI DEI SISTEMI DI RIFERIMENTO DEL LINK 3 E DEL LINK 4.	65
FIGURA 2.18 OTTO CONFIGURAZIONI DEL ROBOT CON STESSA POSA END EFFECTOR E STESSA POSIZIONE DEL GOMITO E VALORI DEI GRADI DI LIBERTÀ AL VARIARE DELL'ANGOLO DI SWIVEL.	76
FIGURA 2.19 GRAFICO 2D (A) E GRAFICO 3D (B) CON I VALORI DEGLI ANGOLI DI SWIVEL FISICAMENTE OTTENIBILI PER UNA DATA POSA DELL'END EFFECTOR.....	77
FIGURA 2.20 POSSIBILI POSIZIONI DEL GOMITO PER OGNUNA DELLE 8 SOLUZIONI ALLA CINEMATICA INVERSA.	78

FIGURA 2.21 ANDAMENTO DEL NUMERO DI SOLUZIONI DI BRACCIO OTTENUTE DALLA CINEMATICA INVERSA FISSANDO LA POSIZIONE DEL GOMITO LUNGO LA CIRCONFERENZA DI SWIVEL NELLO SPAZIO OPERATIVO (A) E AL VARIARE DEI GRADI DI LIBERTÀ q_2 E q_4 (B).....	80
FIGURA 2.22 GRAFICO 3D (A) E MAPPA COLORATA (B) CON PERCENTUALE DI CIRCONFERENZA DI SWIVEL SU CUI IL GOMITO PUÒ FISICAMENTE TROVARSI AL VARIARE DEI GRADI DI LIBERTÀ q_2 E q_4	81
FIGURA 2.23 GRAFICO 3D (A) E MAPPA COLORATA (B) CON PERCENTUALE DI CIRCONFERENZA DI SWIVEL SU CUI IL GOMITO PUÒ FISICAMENTE TROVARSI AL VARIARE DELLA POSIZIONE DEL CENTRO POLSO.	82
FIGURA 3.1 PERCORSO DELL'END EFFECTOR NELLO SPAZIO OPERATIVO.	83
FIGURA 3.2 PERCORSO DEL GOMITO E DELL'END EFFECTOR CON $q_3 = \text{COST}$	84
FIGURA 3.3 PIANO DI RIFERIMENTO RAPPRESENTANTE IL MURO E PERCORSO DEL GOMITO.	85
FIGURA 3.4 ANDAMENTO DEI GRADI DI LIBERTÀ LUNGO LA TRAIETTORIA CON $q_3 = \text{COST}$	85
FIGURA 3.5 SEQUENZA DEI MOVIMENTI DEL ROBOT SIMULATI CON V-REP LUNGO LA TRAIETTORIA CON $q_3 = \text{COST}$ IN PRESENZA DI UN MURO.....	86
FIGURA 3.6 PERCORSO DEL GOMITO E DELL'END EFFECTOR SFRUTTANDO LA RIDONDANZA.	88
FIGURA 3.7 PIANO DI RIFERIMENTO RAPPRESENTANTE IL MURO E PERCORSO GOMITO.	88
FIGURA 3.8 ANDAMENTO DEI GRADI DI LIBERTÀ NEI GIUNTI SFRUTTANDO LA RIDONDANZA CON LA PRESENZA DI UN MURO NELLO SPAZIO DI LAVORO DEL ROBOT.	89
FIGURA 3.9 SEQUENZA DEI MOVIMENTI DEL ROBOT SIMULATI CON V-REP LUNGO LA TRAIETTORIA SFRUTTANDO LA RIDONDANZA IN PRESENZA DI UN MURO.....	90
FIGURA 3.10 PERCORSO DELL'END EFFECTOR NELLO SPAZIO OPERATIVO.	91
FIGURA 3.11 PERCORSO DEL GOMITO CON $q_3 = \text{COST}$	91
FIGURA 3.12 PERCORSO DEL GOMITO E DELL'END EFFECTOR CON $q_3 = \text{COST}$	92
FIGURA 3.13 PERCORSO DEL GOMITO E DELL'END EFFECTOR CON $q_3 = \text{COST}$ IN PRESENZA DI UN OSTACOLO SFERICO.	92
FIGURA 3.14 SEQUENZA DEI MOVIMENTI DEL ROBOT SIMULATI CON V-REP LUNGO LA TRAIETTORIA CON $q_3 = \text{COST}$ IN PRESENZA DI UN OSTACOLO SFERICO.	93
FIGURA 3.15 ANDAMENTO DELL'ANGOLO DI ROTAZIONE δ LUNGO LA TRAIETTORIA	94
FIGURA 3.16 SCHEMATIZZAZIONE ADOTTATA PER IL CALCOLO DELLE POSIZIONI DEL GOMITO DEL MANIPOLATORE.	95
FIGURA 3.17 PERCORSO DEL GOMITO E DELL'END EFFECTOR SFRUTTANDO LA RIDONDANZA IN PRESENZA DI UN OSTACOLO SFERICO.	95
FIGURA 3.18 CONFRONTO TRA GLI ANDAMENTI DEI GRADI DI LIBERTÀ NEI GIUNTI NEL CASO IN CUI SI UTILIZZI IL ROBOT COME UN 6DOF (A) O NEL CASO IN CUI SI SFRUTTI LA RIDONDANZA PER EVITARE COLLISIONE CON OSTACOLO SFERICO (B).	96
FIGURA 3.19 SEQUENZA DEI MOVIMENTI DEL ROBOT SIMULATI CON V-REP LUNGO LA TRAIETTORIA SFRUTTANDO LA RIDONDANZA IN PRESENZA DI UN OSTACOLO SFERICO.	97
FIGURA 4.1 RAPPRESENTAZIONE DEL JOYSTICK DI COMANDO DEL KINOVA JACO 2 [6].....	98
FIGURA 4.2 RAPPRESENTAZIONE DELLA BASE DEL KINOVA JACO 2 CON INDICAZIONE CIRCA LE PORTE DI COLLEGAMENTO E I PULSANTI CHE SI TROVANO SU DI ESSA [7].	99
FIGURA 4.3 INTERFACCIA GRAFICA DEL DEVELOPEMENT CENTER.....	100
FIGURA 4.4 INTERFACCIA GRAFICA DEL TORQUE CONTROL.....	100
FIGURA 4.5 BANCO SPERIMENTALE (A) E SCHEMATIZZAZIONE COLLEGAMENTI (B).	101
FIGURA 4.6 ANDAMENTO DEI GRADI DI LIBERTÀ DEI GIUNTI PERCHÈ IL GOMITO PERCORRA TUTTA LA CIRCONFERENZA DI SWIVEL.....	108
FIGURA 4.7 ANDAMENTO DEI GRADI DI LIBERTÀ NEI GIUNTI CHE PRODUCE IL MEDESIMO EFFETTO DI QUELO IN FIGURA 4.6 SENZA AVERE DEI SALTI GENERATI DALLA SOLUZIONE NUMERICA DELL'ALGORITMO DI CINEMATICA INVERSA.....	108
FIGURA 4.8 CONFRONTO TRA PARAMETRO DI CURVA S CON LEGGE LINEARE E A TRAPEZIO DI VELOCITÀ.	111
FIGURA 4.9 ANDAMENTO DEI GRADI DI LIBERTÀ IN FUNZIONE DEL PARAMETRO DI CURVA S CON LEGGE A TRAPEZIO DI VELOCITÀ.	111
FIGURA 4.10 ANDAMENTO DELLE VELOCITÀ NEI GIUNTI IN FUNZIONE DEL TEMPO.	112

FIGURA 4.11 TRAIETTORIE CON OSTACOLO SIMULATO DALLA MANO NEL CASO DI ROBOT UTILIZZATO COME UN 6DOF (PRIMA COLONNA) O SFRUTTANDO LA RIDONDANZA (SECONDA COLONNA).....	113
FIGURA 4.12 TRAIETTORIE IN PRESENZA DI UN MURO SIMULATO DA UN FOGLIO NEL CASO DI ROBOT UTILIZZATO COME UN 6DOF (PRIMA COLONNA) O SFRUTTANDO LA RIDONDANZA (SECONDA COLONNA).	114
FIGURA 4.13 FOTOGRAMMI PRESI LUNGO LA CIRCONFERENZA DI SWIVEL.	115

INDICE DELLE TABELLE

TABELLA 1.1 PARAMETRI DI DENAVIT-HARTEMBERG	4
TABELLA 1.2 INDICE PERCENTUALE DELLA DISTRIBUZIONE DELLE SOLUZIONI	15
TABELLA 2.1 LIMITI DI GIUNTO KINOVA JACO 2	41
TABELLA 2.2 PARAMETRI DI DENAVIT-HARTEMBERG	44
TABELLA 2.3 ANGOLI DI ROTAZIONE PER OTTENERE IL SISTEMA DI RIFERIMENTO R	59

INTRODUZIONE

Lo scopo principale di questo elaborato è l'analisi di alcuni aspetti relativi alla cinematica inversa di un robot articolato facendo riferimento, in particolare, alla tematica della ridondanza. Tale studio è stato svolto, nello specifico, sul robot collaborativo Jaco 2 prodotto dalla compagnia canadese Kinova. Preliminarmente, per approcciarsi al problema, si è studiato un classico robot antropomorfo a sei gradi di libertà cercando di valutare aspetti connessi alla molteplicità cinematica anche in una struttura che non presenta gradi di libertà addizionali. Per questo tipo di analisi è stato modellizzato il manipolatore IRB120 prodotto dalla ABB. La scelta di partire dallo studio di un robot a sei gradi di libertà trova giustificazione nel fatto che la sua cinematica è più semplice da analizzare ma allo stesso tempo la struttura di un robot ridondante, qualora a quest'ultimo venisse bloccato un grado di libertà nei giunti, è facilmente riconducibile a quella di un braccio robotico 6R. Dopo aver modellizzato l'IRB120 attraverso il Robotics Toolbox sviluppato da Peter Corke in ambiente MATLAB, si è passati alla risoluzione della cinematica inversa in modo analitico. Una volta noto il fatto che, per una determinata posa dell'End Effector esistono 8 diverse soluzioni alla cinematica inversa, è stata svolta un'analisi volta a capire quante e quali di esse siano valide. Considerando, infatti, i limiti imposti sui gradi di mobilità e i problemi legati ad interferenze tra i link contigui è possibile che più di una delle 8 soluzioni generate dal processo di cinematica inversa non sia fisicamente realizzabile. Tale analisi è stata divisa in due parti. Preliminarmente, si è studiato il problema di posizionamento del centro polso legato ai giunti di braccio e successivamente si è studiato il problema di orientazione dell'End Effector. Per quanto riguarda il posizionamento, sono state realizzate mappe colorate che forniscono indicazioni sul reale numero di configurazioni di braccio in grado di portare il centro polso nella posizione desiderata nel rispetto dei limiti di giunto. Successivamente, per studiare il problema di orientamento, è stata svolta un'analisi sulla destrezza del polso. In questo ambito, si è studiato un indice in grado di stimare l'effettiva destrezza del robot, diversa da quella conferita solo dal polso, per ogni posizione del centro polso all'interno dello spazio di lavoro. Parallelamente a tale attività è stato anche implementato un algoritmo di pianificazione della traiettoria del tipo Resolved Rate Motion Control con logica closed ed è stato testato su traiettorie che portano il robot a transitare in configurazioni di singolarità. Dopo aver compreso a fondo la cinematica inversa del

manipolatore antropomorfo a sei gradi di libertà, si è passati alla modellizzazione del robot a 7 gradi di libertà e, anche in questo caso, è stata studiata nel dettaglio la cinematica inversa. Come noto, il principale vantaggio, nonché problema, di un robot ridondante risiede nel fatto che esistono un numero infinito di configurazioni in grado di portare l'End Effector nella posa desiderata. In questo elaborato, dopo aver creato un modello geometrico semplificato del manipolatore, si è individuato un unico parametro, chiamato angolo di Swivel, tramite il quale è possibile conoscere la configurazione in cui si trova il robot conoscendone la posa dell'End Effector. Grazie all'utilizzo di tale parametro è stato possibile trovare una soluzione analitica alla cinematica inversa.

Successivamente, si è cercato di comprendere come sfruttare la ridondanza del robot. Per fare ciò si è studiato il comportamento del braccio robotico mentre il suo End Effector si muove lungo un percorso rettilineo. La medesima traiettoria viene fatta percorrere al manipolatore usandolo come fosse un robot a 6 gradi di libertà o sfruttando tutti i suoi giunti, immaginando la presenza all'interno del workspace di alcuni ostacoli.

Infine, è stata svolta una parte di sperimentazione sul robot Kinova Jaco 2 nei laboratori del DIMEAS del Politecnico di Torino. La sperimentazione è stata svolta con il duplice scopo di visualizzare realmente le traiettorie studiate in modo teorico tramite MATLAB e di comprendere come poter comandare e comunicare con il robot tramite l'uso del medesimo software di calcolo.

CAPITOLO 1: Robot ABB IRB120

Il robot preso in esame è il modello IRB 120 prodotto dalla ABB. Si tratta di un robot a 6 assi, con struttura a braccio articolato (3 gradi di libertà) e polso sferico (3 gradi di libertà). In figura 1.1 viene riportata la rappresentazione del robot in proiezione ortogonale con tutte le quote utili per la sua modellizzazione.

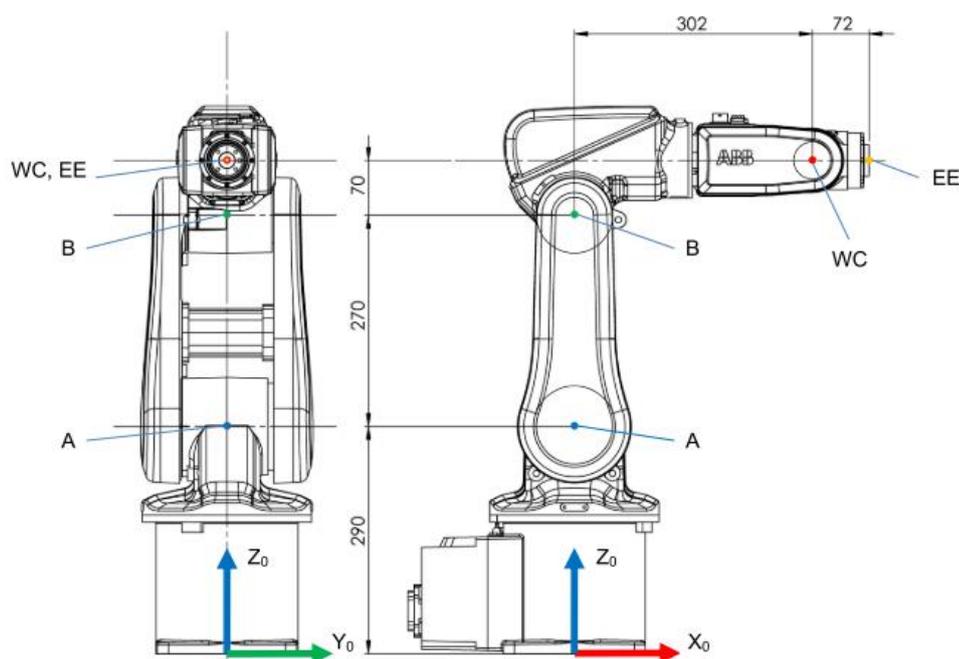


Figura 1.1 Robot ABB IRB120 [14].

1.1 Modellizzazione del robot

Partendo dalle informazioni di base di cui sopra, applicando la convenzione di Denavit-Hartenberg (d'ora in avanti indicata come convenzione di D-H) standard si sono assegnati i sistemi di riferimento a ogni link del manipolatore.

In figura 1.2 sono riportati i sistemi di riferimento di D-H di ogni link nella configurazione di zero del robot, mentre nella tabella 1.1 sono riportati i valori dei parametri di D-H.

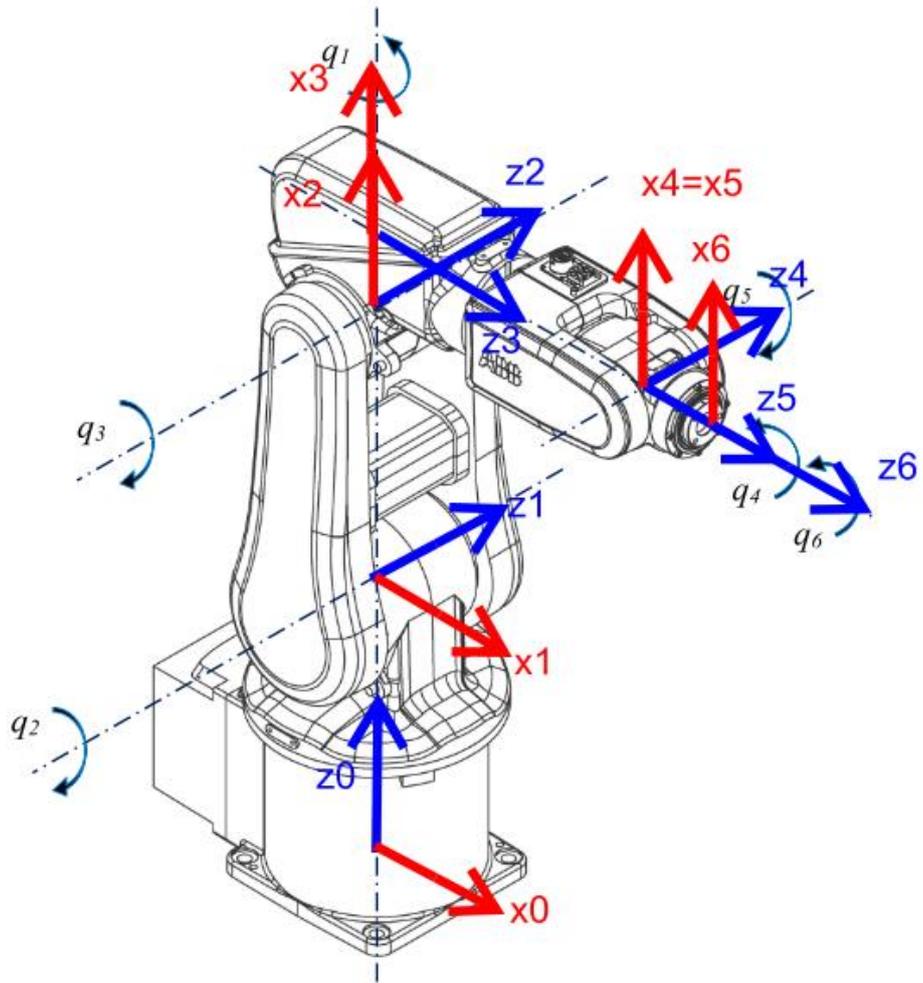


Figura 1.2 Sistemi di riferimento Denavit-Hartenberg.

Tabella 1.1 Parametri di Denavit-Hartenberg ABB IRB120.

Link	α (deg)	a (mm)	d (mm)	ϑ (deg)
1	-90	0	290	q_1
2	0	270	0	$q_2 - 90$
3	-90	70	0	q_3
4	+90	0	302	q_4
5	-90	0	0	q_5
6	0	0	72	q_6

Le matrici di trasformazione omogenee sono le seguenti:

$${}^0\hat{A}_1 = \begin{bmatrix} \cos(q_1) & 0 & -\sin(q_1) & 0 \\ \sin(q_1) & 0 & \cos(q_1) & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.1)$$

$${}^1\hat{A}_2 = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & a_2\cos(q_2) \\ \sin(q_2) & \cos(q_2) & 0 & a_2\sin(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.2)$$

$${}^2\hat{A}_3 = \begin{bmatrix} \cos(q_3) & 0 & -\sin(q_3) & a_3\cos(q_3) \\ \sin(q_3) & 0 & \cos(q_3) & a_3\sin(q_3) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.3)$$

$${}^3\hat{A}_4 = \begin{bmatrix} \cos(q_4) & 0 & \sin(q_4) & 0 \\ \sin(q_4) & 0 & -\cos(q_4) & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.4)$$

$${}^4\hat{A}_5 = \begin{bmatrix} \cos(q_5) & 0 & -\sin(q_5) & 0 \\ \sin(q_5) & 0 & \cos(q_5) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.5)$$

$${}^5\hat{A}_6 = \begin{bmatrix} \cos(q_6) & -\sin(q_6) & 0 & 0 \\ \sin(q_6) & \cos(q_6) & 0 & 0 \\ 0 & -1 & 0 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.6)$$

1.2. Cinematica inversa

Per la soluzione della cinematica inversa, vista la struttura a polso monocentrico del manipolatore, si procede dividendo il problema in due parti. Si è, infatti, prima affrontato il problema del posizionamento del centro polso, che è determinato dall'azione dei soli giunti di spalla e gomito, poi il problema dell'orientazione, legata ai giunti di polso.

Si parte calcolando la posizione del centro polso (coincidente con l'origine del sistema di riferimento del link 4) rispetto al sistema di riferimento 0 nel seguente modo:

$${}^0\underline{p}_4 = {}^0\underline{p}_6 - d_6 {}^0\underline{k}_6 = [x_4 \ y_4 \ z_4]^T \quad (1.7)$$

Dove il vettore ${}^0\underline{k}_6$ è ricavato dall'ultima colonna della matrice di orientazione ${}^0\hat{A}_6$ da cui si parte conoscendo la posa desiderata.

Si procede poi con il calcolo della posizione del centro polso rispetto al sistema di riferimento del link 1:

$$\begin{aligned} {}^1\underline{\hat{p}}_4 &= {}^1\hat{A}_2 {}^2\hat{A}_3 {}^3\underline{\hat{p}}_4 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_3 & 0 & -s_3 & a_3 c_3 \\ s_3 & 0 & c_3 & a_3 s_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ d_4 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} a_3 c_2 c_3 - d_4 s_2 c_3 + a_2 c_2 \\ d_4 c_2 c_3 + a_3 s_2 c_3 + a_2 s_2 \\ 0 \\ 1 \end{bmatrix} \end{aligned} \quad (1.8)$$

$${}^1\underline{\hat{p}}_4 = {}^0\hat{A}_1^{-1} {}^0\underline{\hat{p}}_4 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_4 \\ y_4 \\ z_4 \\ 1 \end{bmatrix} = \begin{bmatrix} x_4 c_1 + y_4 s_1 \\ d_1 - z_4 \\ y_4 c_1 - x_4 s_1 \\ 1 \end{bmatrix} \quad (1.9)$$

Uguagliando le due espressioni si ottiene il seguente sistema:

$$\begin{cases}
x_4 c_1 + y_4 s_1 = a_3 c_{23} - d_4 s_{23} + a_2 c_2 & (1) \\
d_1 - z_4 = d_4 c_{23} + a_3 s_{23} + a_2 s_2 & (2) \\
y_4 c_1 - x_4 s_1 = 0 & (3)
\end{cases} \quad (1.10)$$

Dall' equazione (3) del sistema 1.10:

$$\begin{cases}
l = \sqrt{x_4^2 + y_4^2} \\
\gamma = \text{atan2}(y_4, x_4)
\end{cases} \quad \begin{cases}
x_4 = l c_\gamma \\
y_4 = l s_\gamma
\end{cases} \quad (1.11)$$

$$\begin{aligned}
(c_1 s_\gamma - s_1 c_\gamma) l &= 0 \\
\sin(\gamma - q_1) l &= 0 \\
\gamma - q_1 &= \text{atan2}(0, \pm 1) \text{ per } l \neq 0 \\
q_1 &= \text{atan2}(y_4, x_4) - \text{atan2}(0, \pm 1)
\end{aligned} \quad (1.12)$$

Sommando il quadrato delle tre equazioni:

$$\begin{aligned}
x_4^2 + y_4^2 + (z_4 - d_1)^2 &= a_2^2 + 2a_3 a_2 c_3 - 2a_2 d_4 s_3 + a_3^2 + a_4^2 \\
a_3 c_3 - d_4 s_3 &= \frac{x_4^2 + y_4^2 + (z_4 - d_1)^2 - a_3^2 - a_4^2 - a_2^2}{2a_2} = B
\end{aligned} \quad (1.13)$$

$$\begin{cases}
e = \sqrt{a_3^2 + d_4^2} \\
\xi = \text{atan2}(a_3, d_4)
\end{cases} \quad \begin{cases}
a_3 = e s_\xi \\
d_4 = e c_\xi
\end{cases} \quad (1.14)$$

$$\begin{aligned}
(c_3 s_\xi - s_1 c_\xi) e &= B \\
\cos(q_3 - \xi) e &= B
\end{aligned} \quad (1.15)$$

$$\begin{cases}
\cos(q_3 - \xi) = \frac{B}{e} \\
\sin(q_3 - \xi) = \sqrt{1 - \left(\frac{B}{e}\right)^2}
\end{cases} \quad (q_3 - \xi) = \text{atan2}\left(\pm \sqrt{1 - \left(\frac{B}{e}\right)^2}, \frac{B}{e}\right) \quad (1.16)$$

$$q_3 = \text{atan2}(a_3, d_4) + \text{atan2}\left(\pm\sqrt{1 - \left(\frac{B}{e}\right)^2}, \frac{B}{e}\right) \quad (1.17)$$

Risolvendo le espressioni (1) e (2) del sistema 1.10 si ottengono i valori di s_2 e c_2 e infine

$$q_2 = \text{atan2}(s_2, c_2) \quad (1.18)$$

Possono essere calcolati quattro valori di q_2 , uno per ogni paio di valori (q_1, q_3) .

Avendo determinato il valore dei primi tre gradi di libertà, è possibile calcolare la matrice ${}^0\hat{A}_3$:

$${}^0\hat{A}_3 = {}^0\hat{A}_1 {}^1\hat{A}_2 {}^2\hat{A}_3 = \begin{bmatrix} c_{23}c_1 & s_1 & -s_{23}c_1 & c_1(a_3c_{23} + a_2c_2) \\ c_{23}s_1 & -c_1 & -s_{23}s_1 & s_1(a_3c_{23} + a_2c_2) \\ -s_{23} & 0 & -c_{23} & d_1 - a_3s_{23} - a_2s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.19)$$

Conoscendo tale matrice è possibile calcolare i valori della matrice ${}^3\hat{A}_6$:

$${}^3\hat{A}_6 = {}^0\hat{A}_3^{-1} {}^0\hat{A}_6 = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \quad (1.20)$$

Avendo calcolato però le varie matrici di orientazione ${}^{i-1}\hat{A}_i$.

$${}^3\hat{A}_4 = \begin{bmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.21)$$

$${}^4\hat{A}_5 = \begin{bmatrix} c_5 & 0 & -s_5 & 0 \\ s_5 & 0 & c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.22)$$

$${}^5\hat{A}_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.23)$$

È possibile ricavare la matrice ${}^3\hat{A}_6$:

$${}^3\hat{A}_6 = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -s_4c_6 - c_4c_5s_6 & -s_5c_4 & -d_6s_5c_4 \\ c_4s_6 + s_4c_5c_6 & c_4c_6 - s_4c_5s_6 & -s_5s_4 & -d_6s_4s_5 \\ c_6s_5 & -s_5s_6 & c_5 & d_4 + d_6c_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.24)$$

Uguagliando tale matrice alla matrice dei valori noti calcolata precedentemente è possibile ottenere i valori dei gradi di libertà nei giunti:

$$q_5 = \pm \arccos(A_{33}) \quad (1.25)$$

$$q_4 = \text{atan2}\left(-\frac{A_{23}}{\sin(q_5)}, -\frac{A_{13}}{\sin(q_5)}\right) \quad (1.26)$$

$$q_6 = \text{atan2}\left(-\frac{A_{32}}{\sin(q_5)}, \frac{A_{31}}{\sin(q_5)}\right) \quad (1.27)$$

Se $\sin(q_5) = 0$:

$$q_5 = 0 \quad q_4 + q_6 = \text{atan2}(-A_{12}, A_{11}) \quad (1.28)$$

$$q_5 = \pi \quad q_4 - q_6 = \text{atan2}(-A_{12}, -A_{11}) \quad (1.29)$$

Si ottengono, in definitiva, due set di soluzioni per i gradi di libertà del polso e 4 set di soluzioni per i gradi di libertà di braccio. In totale, il processo di cinematica inverso applicato al nostro manipolatore a 6 gradi di libertà produce 8 soluzioni distinte.

1.3. Analisi della molteplicità di soluzioni di braccio

Nel momento in cui vengono generate tali soluzioni è necessario capire quante e quali di esse siano valide. Considerando, infatti, i limiti imposti sui gradi di mobilità e i problemi legati ad interferenze tra i link è possibile che più di una delle 8 soluzioni generate dal processo di cinematica inversa non siano fisicamente realizzabili.

Per considerare tale fenomeno si è deciso di analizzare l'andamento del numero di soluzioni fisicamente ottenibili al variare della posa dell'End Effector. Per avere la certezza che la posa richiesta si trovasse all'interno del workspace del robot, essa non è stata imposta, ma calcolata tramite un processo di cinematica diretta partendo dai valori dei gradi di libertà nei giunti

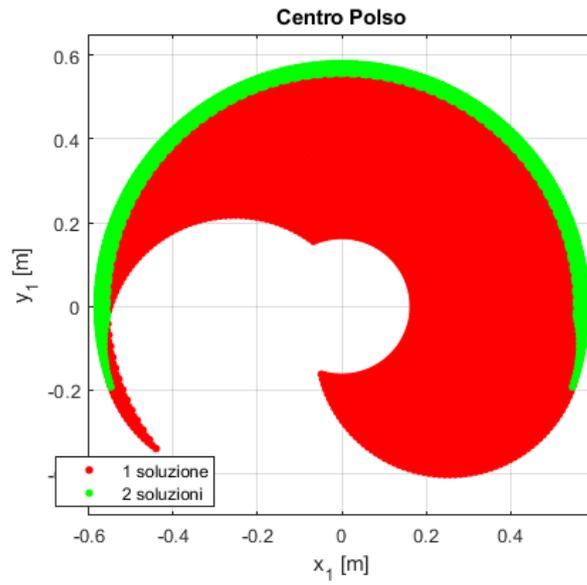
Come prima cosa si è svolta un'analisi relativa alla sola posizione del centro polso. Tenendo il valore del giunto q_1 fisso e andando a variare il valore dei gradi di libertà q_2 e q_3 si è analizzato qual è la distribuzione del numero di soluzioni ottenibili risolvendo la cinematica inversa per ogni posizione del centro polso.

Imponendo il valore del grado di libertà $q_1 = 0^\circ$ l'andamento del numero di soluzioni che si ottiene è rappresentato in figura 1.3.

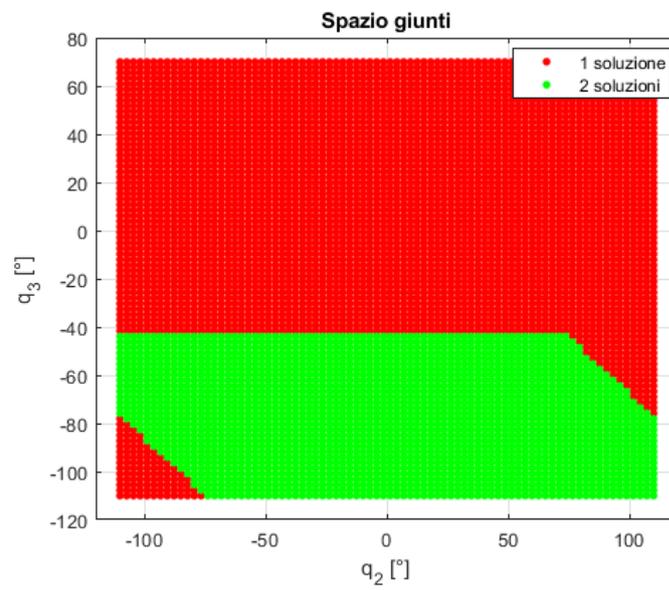
Dove l'area colorata in rosso rappresenta una zona del workspace che il robot può raggiungere solo attraverso una configurazione mentre l'area colorata in verde rappresenta una zona del workspace raggiungibile in due modi diversi. Il fatto che il massimo numero di soluzioni ottenibili con il valore di $q_1=0^\circ$ è pari a due dipende dai limiti imposti sulla rotazione dell'asse 1. Infatti, essendo i limiti sul giunto 1 pari a $\pm 165^\circ$, con un valore di $q_1 = 0^\circ$ le due soluzioni legate alla rotazione del braccio di 180° sono precluse.

Nella figura 1.4 sono riportate le 2 configurazioni del robot possibili per raggiungere la medesima posizione del centro polso.

Andando, invece, a variare il valore del grado di libertà q_1 e ponendolo pari a 90° la mappa colorata che si ottiene è rappresentata in figura 1.5.

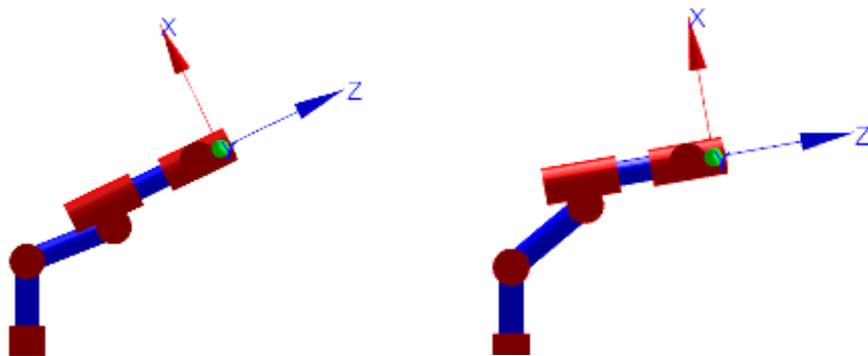


(a)



(b)

Figura 1.3 Andamento del numero di soluzioni nello spazio operativo (a) e variando q_2 e q_3 (b) con $q_1 = 0^\circ$.



(a) Right handed-Elbow down

(b) Right handed-Elbow up

Figura 1.4 Due configurazioni del robot per raggiungere la medesima posizione del centro polso.

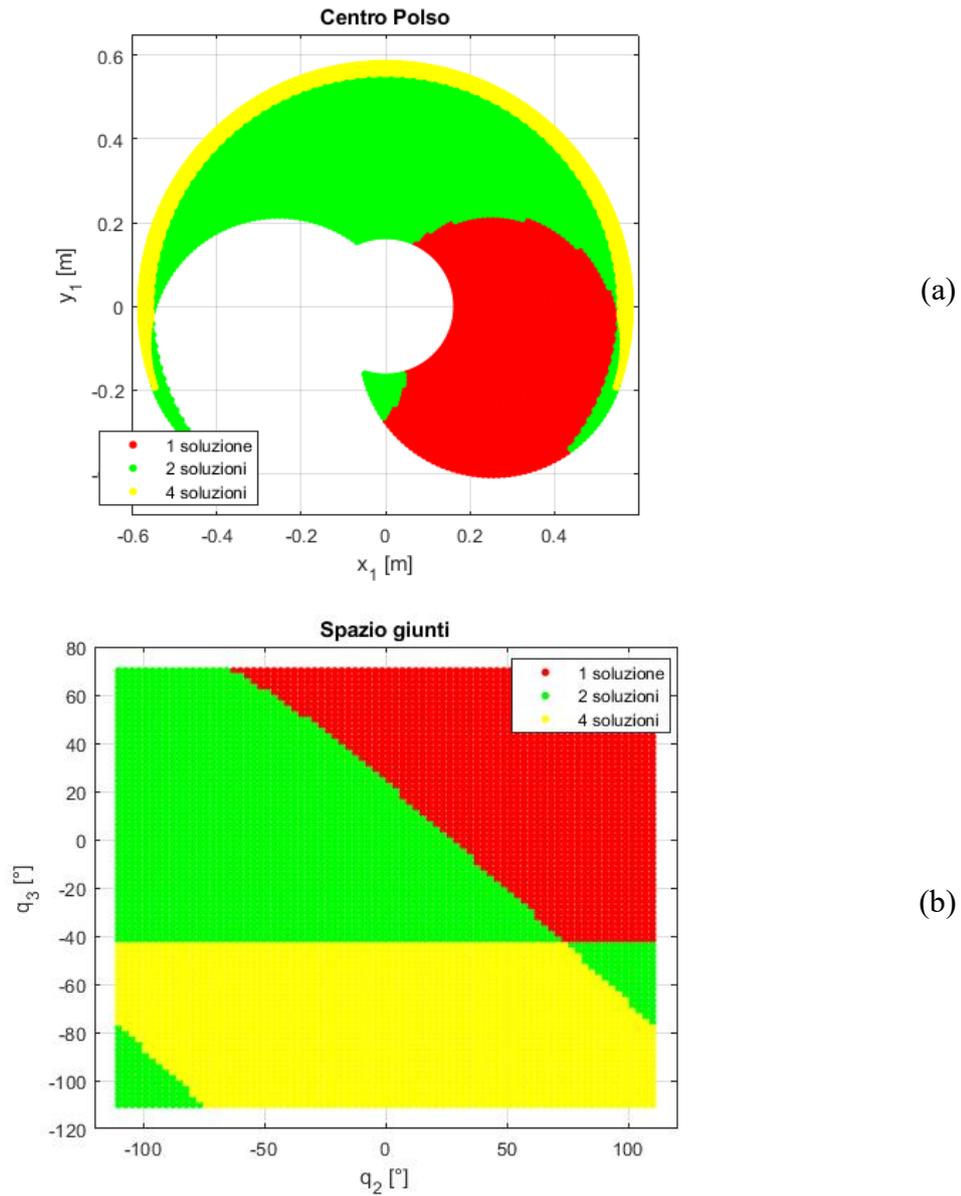


Figura 1.5 Andamento del numero di soluzioni nello spazio operativo (a) e variando q_2 e q_3 (b) con $q_1 = 90^\circ$.

In questo caso ci sono delle aree in cui tutte e 4 le soluzioni ottenibili attraverso il processo di cinematica inversa sono fisicamente accettabili. Ciò deriva dal fatto che, ruotando di 180° il giunto 1, il robot si trova all'interno dei limiti del giunto. La presenza di una zona in cui c'è solo una soluzione è legata al fatto che, pur potendo il robot ruotare di 180° sul suo primo asse, esso non ha comunque a disposizione dei valori di q_2 e q_3 tali da poterlo portare nella posizione richiesta. Per comprendere tale fatto è sufficiente considerare la mappatura che si ottiene imponendo un valore di $q_1 = -90^\circ$. Provando a sovrapporre la rappresentazione dei due spazi di lavoro si nota come le aree rosse corrispondano a punti

esterni al workspace della configurazione diametralmente opposta; tale rappresentazione è visibile in figura 1.6.

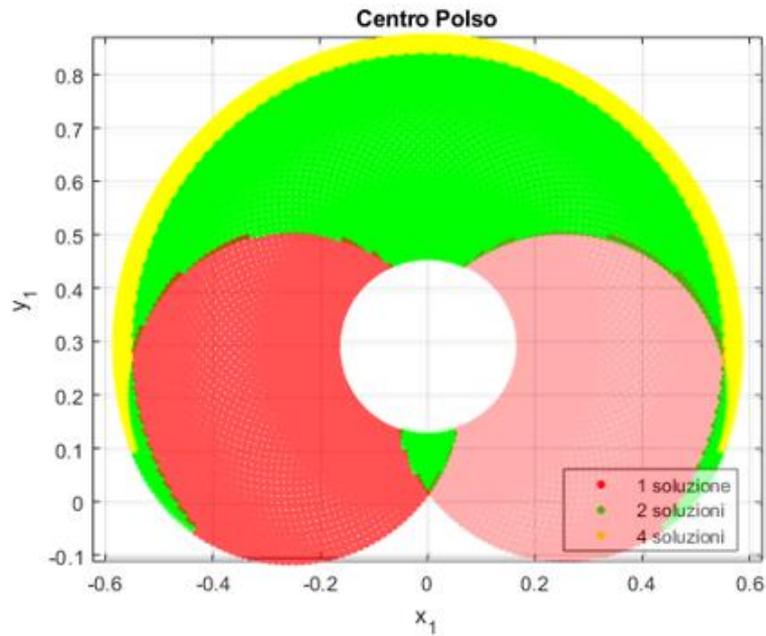


Figura 1.6 Sovrapposizione della distribuzione del numero di soluzioni per due configurazioni diametralmente opposte.

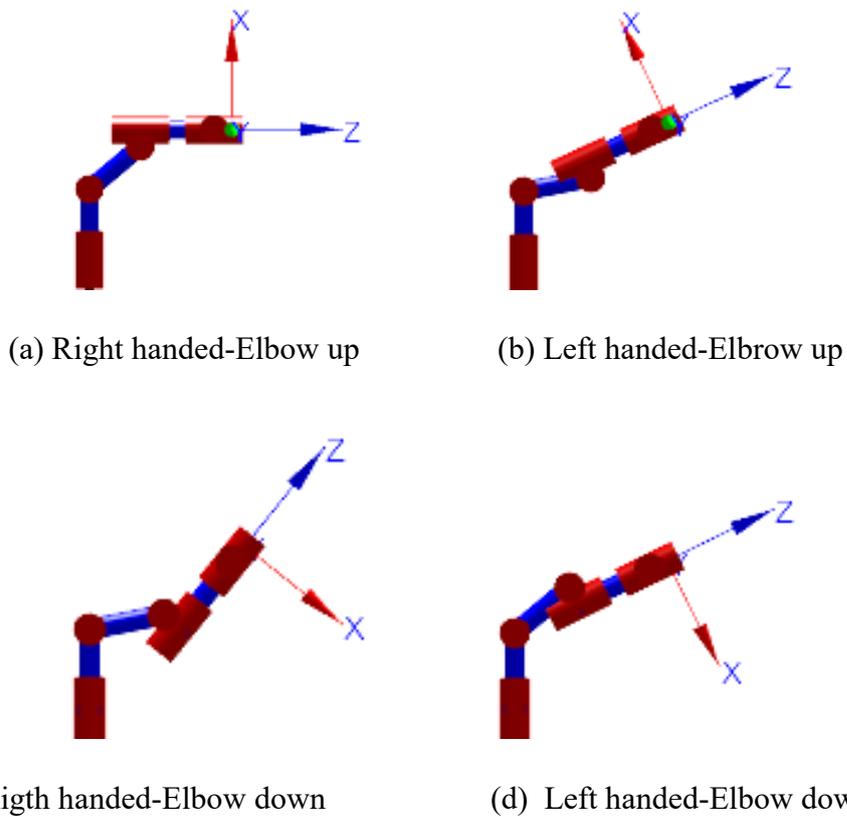


Figura 1.7 Quattro configurazioni del robot per raggiungere la medesima posizione del centro polso.

In figura 1.7 sono riportate le 4 configurazioni del robot possibili per raggiungere la medesima posizione del centro polso quando i gradi di libertà di braccio valgono $[90^\circ \ 50^\circ \ -50^\circ]$. Andando a visualizzare lo spazio di lavoro del manipolatore dall'alto, esso può quindi essere sinteticamente diviso in una zona in cui sono possibili al massimo 2 soluzioni e una zona in cui sono possibili al massimo 4 soluzioni; tale rappresentazione è visibile in figura 1.8.

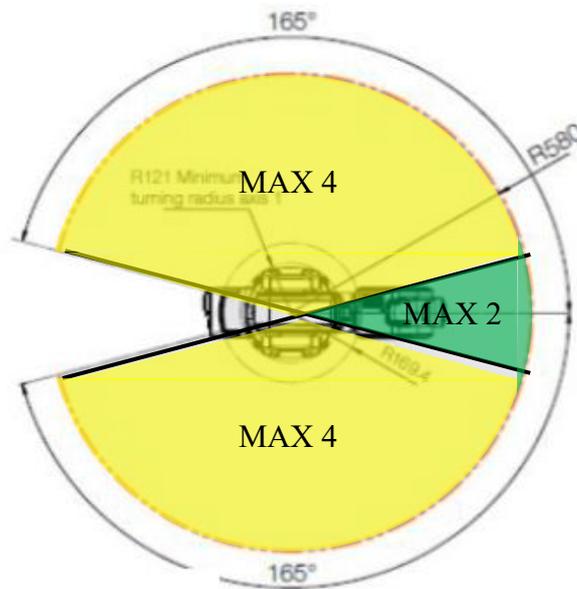


Figura 1.8 Distribuzione del massimo numero di soluzioni al variare di q_1 .

Per avere una stima sulla distribuzione del numero di soluzioni all'interno del workspace si sono calcolate tutte le posizioni raggiungibili dal manipolatore facendo variare anche il valore del giunto 1 e si è stimato il seguente indice percentuale:

$$\text{indice} = \frac{\text{posizioni CP con } n \text{ soluzioni possibili}}{\text{posizioni CP totali calcolate}} \cdot 100 \quad (1.30)$$

I valori di tale indice sono riassunti nella tabella 1.2.

Tabella 1.2 Indice percentuale della distribuzione delle soluzioni

Numero di soluzioni possibili	Indice
1	31.5%
2	38.3%
4	30.2%

Si può notare come, sebbene a livello teorico esistono 4 diverse configurazioni di braccio in grado di portare il centro polso nella posizione richiesta, in realtà, considerando solo i limiti di giunto e non altre problematiche quali la compenetrazione tra i link, la molteplicità di soluzioni nello spazio di lavoro è molto più ridotta.

1.4. Analisi della destrezza

Una volta studiato il problema di posizione del centro polso si passa all'analisi del problema di orientazione. Con il termine destrezza si definisce proprio la capacità del polso di fornire all'organo terminale l'orientamento richiesto.

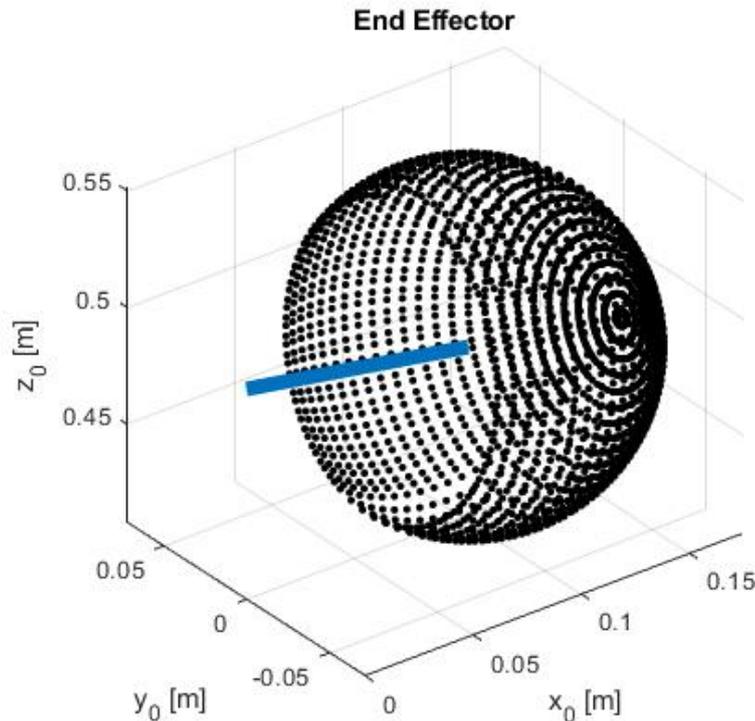


Figura 1.9 Posizioni possibili dell'End Effector data una determinata posa del centro polso.

Prendendo una posizione qualsiasi del centro polso all'interno dello spazio di lavoro e facendo variare i gradi di libertà q_4 e q_5 di polso tra i loro valori limite si ottengono tutti i punti raggiungibili dall'End Effector. Essendo il manipolatore a polso sferico, tali punti descrivono nello spazio la porzione di sfera visibile in figura 1.9. Tale porzione di sfera, essendo definita dai limiti di giunto sul quarto e quinto asse, avrà le medesime dimensioni per tutte le posizioni del centro polso che si considerano, ciò che cambierà sarà solamente la sua orientazione spaziale.

È interessante studiare ciò che succede alla superficie e al volume di tale calotta sferica al variare del valore di q_5 . Per svolgere tale analisi si è partiti dalle formule per il calcolo del volume e della superficie di una calotta sferica che sono le seguenti:

$$S = 2\pi rh \quad (1.31)$$

$$V = \pi h^2 \left(r - \frac{h}{3} \right) \quad (1.32)$$

Dove r è il raggio della sfera, nel nostro caso pari alla distanza tra il punto centro polso e il punto End Effector, e h è l'altezza della calotta sferica e nel nostro caso vale:

$$h = r(1 - \cos(q_5)) \quad (1.33)$$

In figura 1.10 si mostra l'andamento di volume e superficie della calotta sferica, normalizzati rispetto al volume e alla superficie della sfera completa, al variare del valore di q_5 .

Si nota come, poiché il valore limite di q_5 è pari a 120° , la superficie massima della calotta sferica è pari al 75% della superficie totale della sfera. Tale valore però non è rappresentativo dell'effettiva destrezza del polso. Infatti, è importante notare, come non tutte le posizioni all'interno del workspace avranno la stessa capacità di orientare l'End Effector. Come visto dall'analisi illustrata nel capitolo 1.3, potendo molti punti dello spazio di lavoro essere raggiunti secondo 2 o 4 configurazioni di braccio diverse, per una posizione specifica del centro polso ci sono 2 o 4 porzioni di sfera con il medesimo centro ma un'orientazione diversa. Per avere una stima della destrezza totale in un punto si deve quindi prendere in considerazione la sovrapposizione di tali sfere. Un esempio di tale sovrapposizione è visibile in figura 1.12.

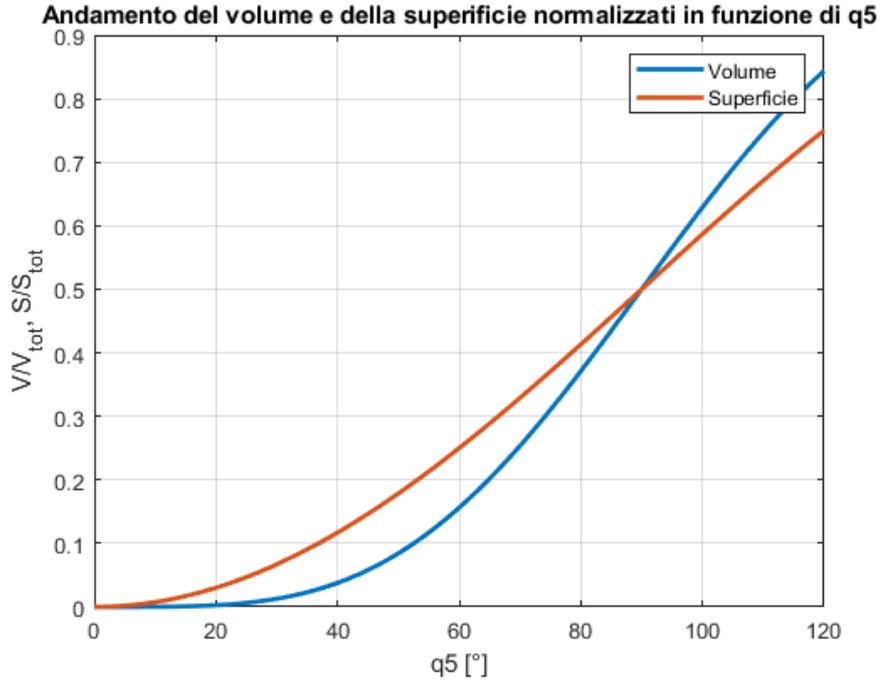


Figura 1.10 Andamento del volume e della superficie normalizzati in funzione di q_5 .

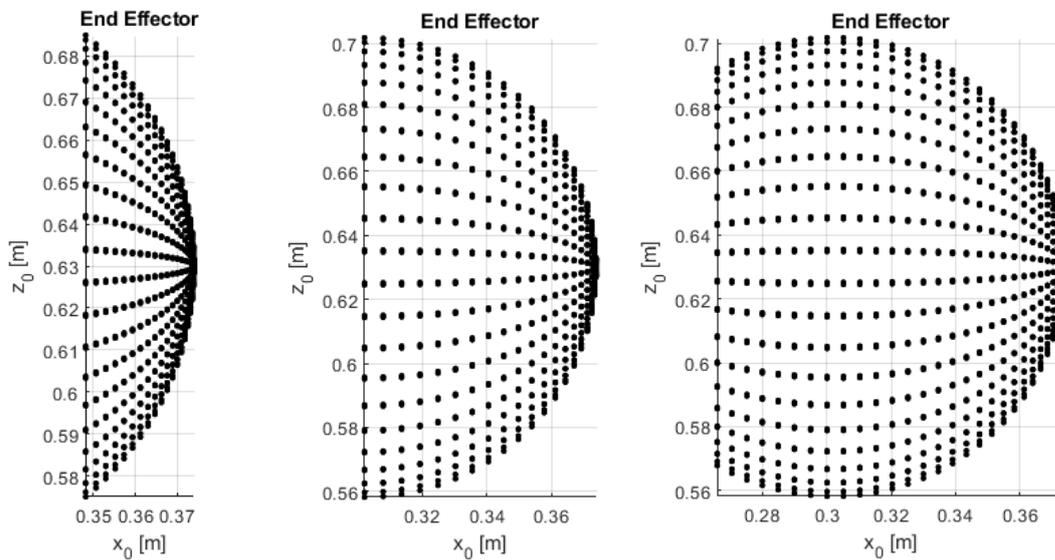


Figura 1.11 Variazione della calotta sferica al variare di q_5 .

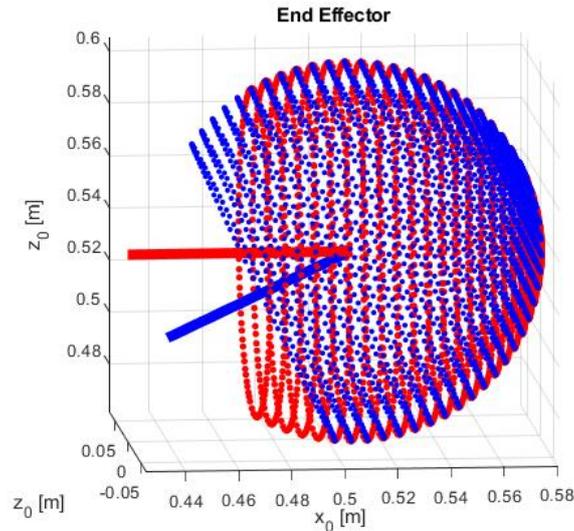


Figura 1.12 Sovrapposizione dei punti effettivamente raggiungibili dall'End Effector legata ad una diversa soluzione di braccio per portare il centro polso nella medesima posizione.

Dalla figura 1.12 si può vedere come i punti effettivamente raggiungibili dal robot quando il suo centro polso si trova nella configurazione considerata sono quelli dati dalla sovrapposizione della porzione di sfera rossa e di quella blu. La destrezza del manipolatore in tale punto sarà perciò maggiore di quella che si ha in un punto in cui esiste una sola configurazione di braccio possibile.

Esistono alcune configurazioni particolari in cui la sovrapposizione delle due porzioni di sfera permette di avere la massima destrezza possibile; in figura 1.13 si riporta, come esempio, il grafico della destrezza del manipolatore quando i gradi di libertà di braccio valgono $[90^\circ \ 110^\circ \ 70^\circ]$. Le due configurazioni del robot sono riportate in figura 1.14.

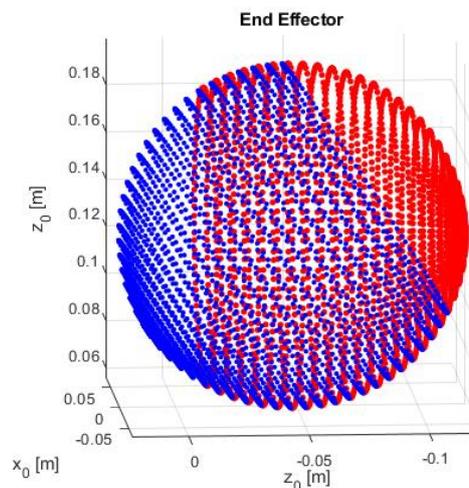


Figura 1.13 Caso particolare in cui la sovrapposizione genera una sfera completa.



(a) Right handed-Elbow up

(b) Left handed-Elbow up

Figura 1.14 Due configurazioni che producono la sfera di destrezza visibile in figura 1.13.

È importante notare come, benché i valori dei gradi di libertà dei giunti si trovano all'interno dei limiti, queste due configurazioni non sono fisicamente ottenibili in quanto generano delle compenetrazioni tra i link. In questa analisi tale fatto non è stato considerato. È quindi possibile, come in questo caso, che laddove si sono considerate come ammissibili una, due o quattro soluzioni di braccio alla cinematica inversa, il numero di configurazioni effettivamente raggiungibili sia inferiore o addirittura nullo.

Una volta presa conoscenza di questo fatto si vuole studiare un indice in grado di dare una rappresentazione di quella che è l'effettiva destrezza del manipolatore all'interno del workspace. Tale indice è di natura percentuale e viene calcolato in modo discreto come rapporto tra i punti in cui è stata discretizzata la sfera effettivamente raggiungibili dall'End Effector e i punti totali costituenti la sfera.

$$indice = \frac{sup\ raggiungibile\ sfera}{sup\ totale\ sfera} = \frac{punti\ raggiungibili}{punti\ totali\ sfera} \cdot 100 \quad [\%] \quad (1.34)$$

La valutazione di tale indice nel caso in cui solo una soluzione alla cinematica inversa di braccio è possibile risulta essere molto semplice. Il valore dell'indice in questi punti, infatti, è calcolato esattamente dalla relazione 1.31 e vale 75%. Per fare in modo che tale valore combaci con quello ricavabile in maniera discreta attraverso la formula 1.34, è necessario effettuare un'opportuna discretizzazione della sfera. La distribuzione dei punti che permette di avere una stima corretta dell'indice è quella che ad ogni punto associa un pezzo di superficie uguale. Per fare ciò si generano dei paralleli tali per cui l'angolo che formano

al centro della sfera sia costante. Se l'angolo al centro è sempre uguale così sarà anche per la lunghezza dell'arco tra un parallelo e l'altro. Per avere anche la seconda dimensione del rettangolino sferico costante è necessario che su ogni parallelo ci sia un numero diverso di punti. Se il numero di punti presenti su ogni parallelo fosse costante risulterebbe un addensamento di punti in prossimità dei due poli che genererebbe una discretizzazione non corretta. La legge usata per conoscere il numero di punti da distribuire lungo ogni parallelo, una volta imposto una lunghezza dell' arco $\delta = \text{costante}$ e noto il diametro della circonferenza d_i descritta dal parallelo i -esimo, è la seguente:

$$\frac{\pi d_i}{\delta} = n \quad (1.35)$$

Il valore del numero di punti ricavato dalla relazione 1.35 difficilmente sarà un numero intero. Esso viene quindi arrotondato all'intero maggiore più vicino, commettendo un piccolo errore sul risultato finale.

Una volta che si è realizzata la suddetta discretizzazione si può procedere con il calcolo dell'indice di interesse. Per i punti centro polso raggiungibili attraverso una sola configurazione di braccio i punti della sfera effettivamente raggiungibili dall'End Effector sono solo quelli tali per cui q_4 e q_5 si trovano all'interno dei loro valori limite.

Nel caso in cui la posizione del centro polso considerata può essere raggiunta tramite due diverse configurazioni di braccio il processo diventa più articolato. Come prima cosa è necessario, prendendo una posizione qualunque del centro polso, far variare i gradi di libertà q_4 e q_5 all'interno dei loro range ammissibili e applicare un processo di cinematica diretta in modo da conoscere le matrici di orientazione dell'End Effector (le coordinate dei punti costituenti la sfera).

Dal processo di cinematica diretta si calcola anche la matrice ${}^0\hat{A}_{5(zero)}$ ovvero la matrice omogenea del sistema di riferimento di Denavit-Hartenberg del link 5 nella configurazione in cui $q_4 = q_5 = 0$.

In figura 1.15 è visibile una rappresentazione di tali sistemi di riferimento.

Una volta note tali matrici si calcola, per ogni combinazione di q_4 e q_5 , la matrice di orientazione ${}^5(zero)\hat{A}_6$ nel modo seguente:

$${}^5(zero)\hat{A}_6 = {}^0\hat{A}_5(zero)^{-1} {}^0\hat{A}_6 = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \quad (1.36)$$

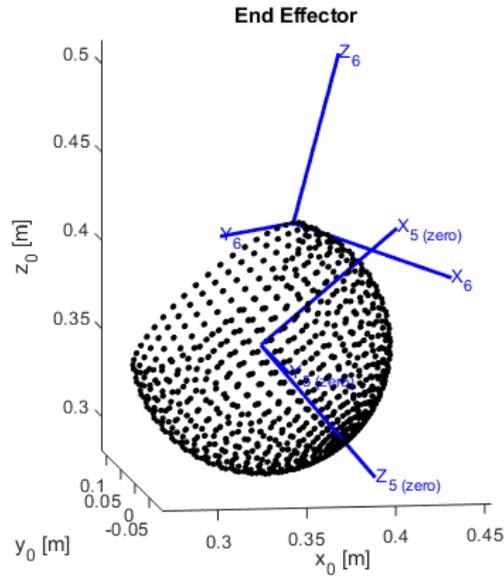


Figura 1.15 Punti raggiungibili dall'End Effector data una posizione del centro polso con indicazione di due sistemi di riferimento di interesse.

L'elemento A_{34} di tale matrice rappresenta la coordinata z del sistema di riferimento 6 rispetto al sistema di riferimento 5 (zero) fisso.

$${}^5(zero)z_6 = A_{34} \quad (1.37)$$

Una volta noti i valori di A_{34} per tutti i punti della porzione di sfera si calcola il seguente parametro:

$$z_{min} = \min({}^5(zero)z_6) \quad (1.38)$$

Tale parametro rappresenta la quota, rispetto al sistema di riferimento 5 (zero), dell'ultima fascia di punti effettivamente raggiungibili dall'End Effector sulla sfera e ha un valore fisso.

Per conoscere il numero di punti, sui punti totali costituenti la sfera, effettivamente raggiungibili dall'End Effector bisogna sommare al numero di punti associati alla prima configurazione di braccio, il numero di punti della seconda configurazione che rispettano le seguenti condizioni:

- valori dei gradi di libertà di q_4 e q_5 all'interno dei range ammissibili.
- ${}^5(\text{zero})z_6 < z_{min}$

La freccia nera in figura 1.16 evidenzia i punti della seconda configurazione che rispettano tali condizioni e che quindi devono essere sommati ai punti associati alla prima configurazione (blu) per avere il numero di punti totali effettivamente raggiungibili dall'End Effector.

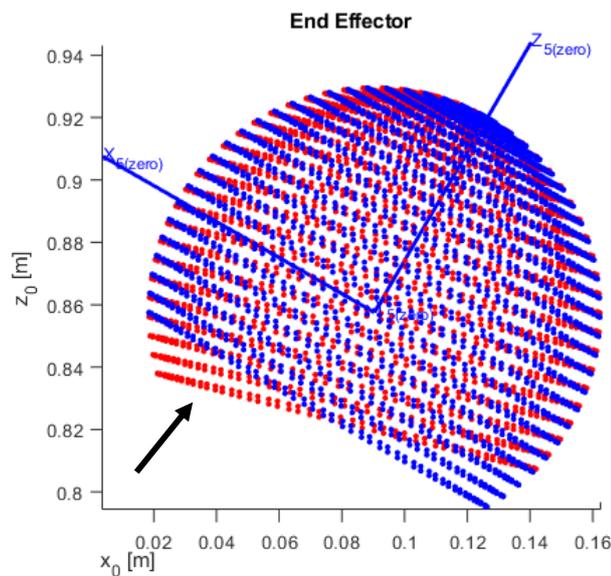


Figura 1.16 Sovrapposizione dei punti effettivamente raggiungibili dall'End Effector legati all'esistenza di due diverse configurazioni di braccio che producono la stessa posizione del centro polso.

Nel caso in cui le soluzioni alla cinematica inversa di braccio siano 4 il processo è simile al caso in cui le soluzioni sono due. Per conoscere il numero di punti, sui punti totali costituenti la sfera, effettivamente raggiungibili dall'End Effector bisogna sommare al numero di punti associati alla prima configurazione di braccio, oltre al numero di punti della seconda configurazione che rispettano le medesime condizioni del caso precedente, anche alcuni punti della terza e quarta configurazione.

Per individuare tali punti è necessario calcolare la matrice di orientazione del sistema di riferimento di Denavit-Hartenberg del link 5 nella configurazione in cui $q_4 = q_5 = 0$ per la seconda e terza configurazione di braccio che porta il centro polso nel punto considerato.

Tali matrici sono state chiamate rispettivamente ${}^0\hat{A}_5(zero)$ e ${}^0\hat{A}_5(zero'')$.

Applicando le equazioni 1.37 e 1.38 si ottengono i valori di ${}^5(zero)z_6$ e ${}^5(zero'')z_6$.

I punti della terza configurazione da sommare ai punti ottenuti dalle prime due configurazioni devono rispettare tali condizioni:

- valori dei gradi di libertà di q_4 e q_5 all'interno dei range ammissibili.
- ${}^5(zero)z_6 < z_{min}$
- ${}^5(zero'')z_6 < z_{min}$

Nella figura 1.17 sono mostrati in verde, e indicati dalla freccia nera, i punti della terza configurazione che rispettano tali condizioni.

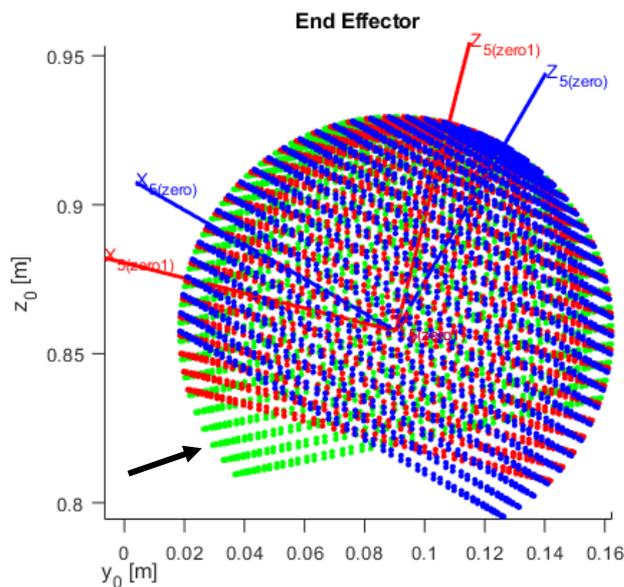


Figura 1.17 Sovrapposizione dei punti effettivamente raggiungibili dall'End Effector legati all'esistenza di tre diverse configurazioni di braccio che producono la stessa posizione del centro polso.

Infine, i punti della quarta configurazione da sommare ai punti ottenuti dalle prime tre devono rispettare tali condizioni:

- valori dei gradi di libertà di q_4 e q_5 all'interno dei range ammissibili.
- ${}^{5(zero)}Z_6 < z_{min}$
- ${}^{5(zero')}Z_6 < z_{min}$
- ${}^{5(zero'')}Z_6 < z_{min}$

Una volta noto il numero di punti raggiungibili dall'End Effector per ogni posizione del centro polso, si calcola l'indice di destrezza usando l'equazione 1.34.

In figura 1.18 viene mostrato l'andamento di tale indice al variare del valore di q_2 e q_3 su un piano con $q_1 = 0^\circ$.

Dall'analisi svolta nel capitolo 1.3 si è capito come su tale piano siano possibili solo 1 o 2 soluzioni di braccio alla cinematica inversa per ogni singola posizione del centro polso. Si può infatti vedere dal grafico come l'indice sia costante e pari a circa il 75% laddove solo una soluzione è possibile e sia maggiore laddove vi sono due soluzioni alla cinematica inversa di braccio.

Nel caso in cui si considera un piano generato dalla variazione dei valori di q_2 e q_3 e mantenendo il valore di $q_1 = 90^\circ$, l'andamento dell'indice che si ottiene è rappresentato in figura 1.20 e figura 1.21.

Anche in questo caso si può notare come l'indice sia costante e pari a circa il 75% laddove solo una soluzione è possibile e sia maggiore laddove vi sono due o quattro soluzioni alla cinematica inversa di braccio.

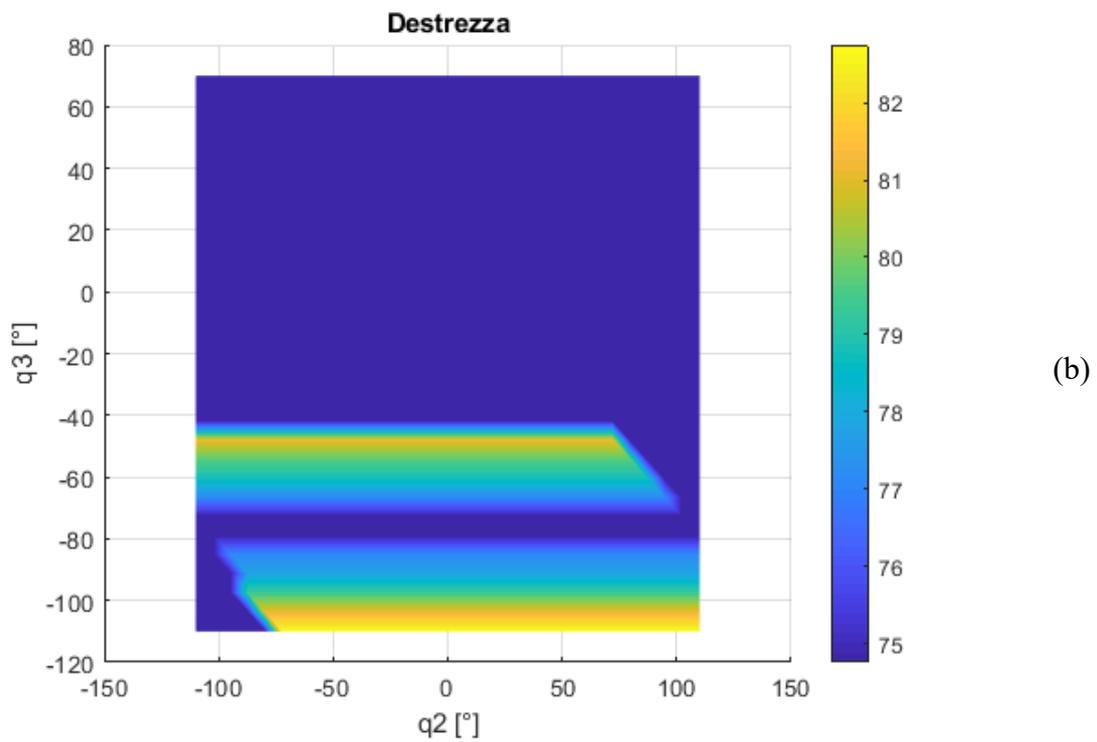
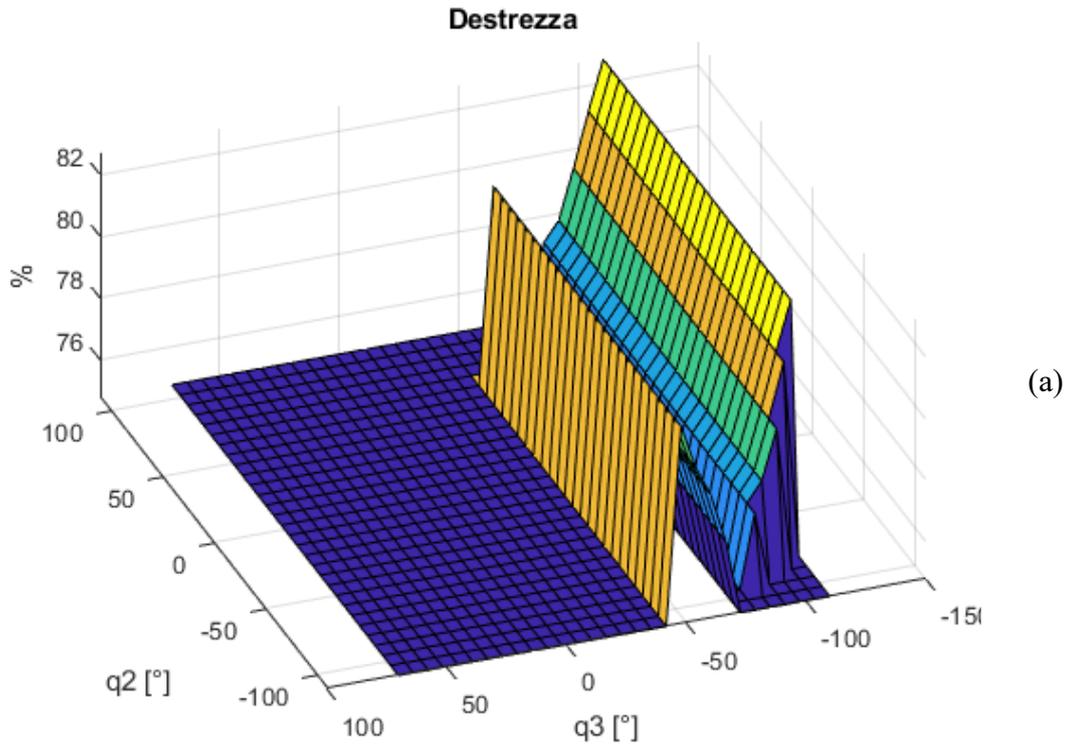


Figura 1.18 Grafico 3D (a) e mappa colorata (b) con andamento indice di destrezza al variare dei gradi di libertà q_2 e q_3 e mantenendo il grado di libertà $q_1 = 0^\circ$.

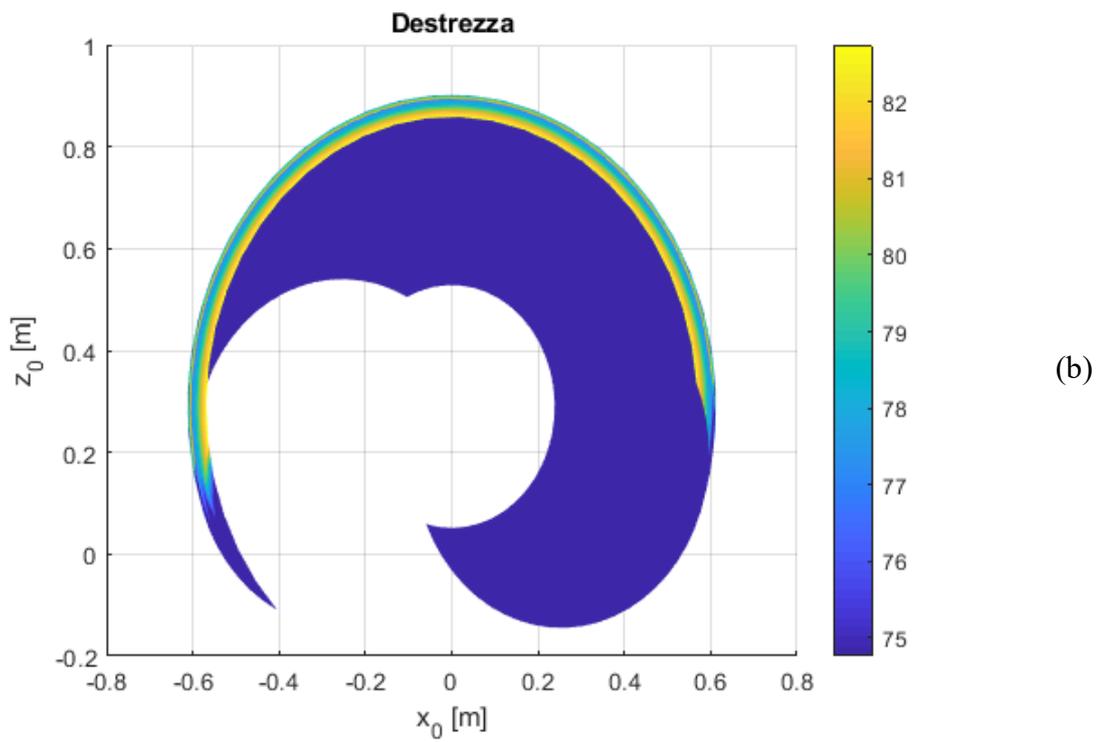
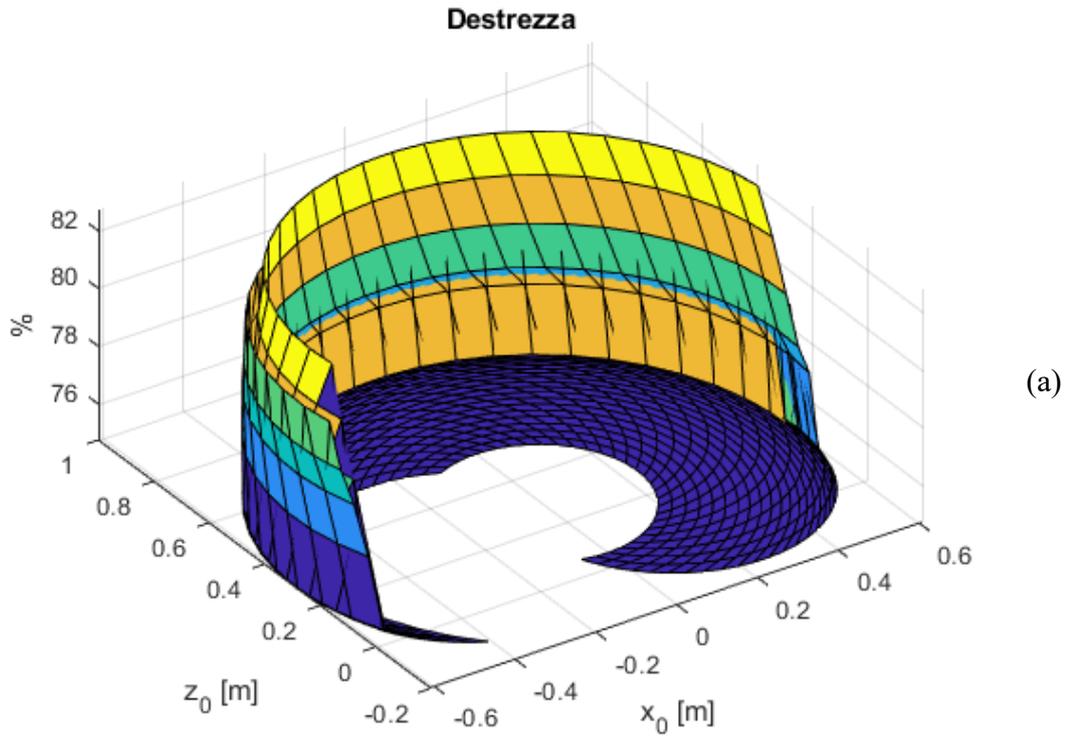


Figura 1.19 Grafico 3D (a) e mappa colorata (b) con andamento indice di destrezza al variare della posizione del centro polso su un piano con $q_1 = 0^\circ$.

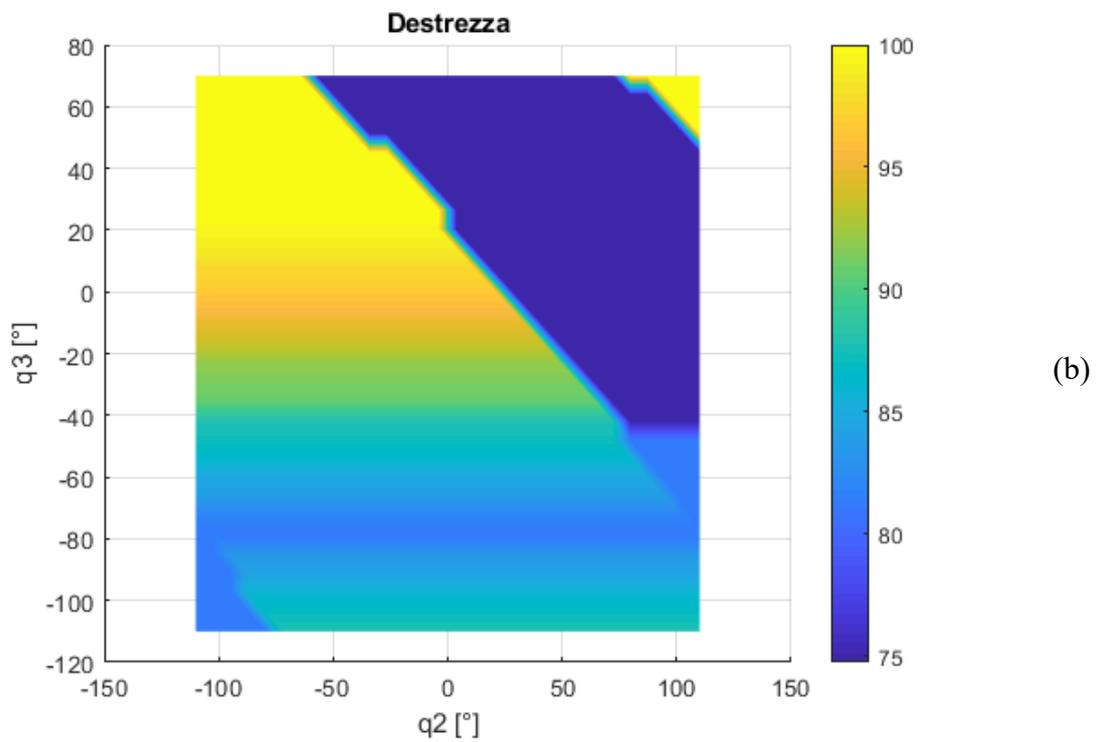
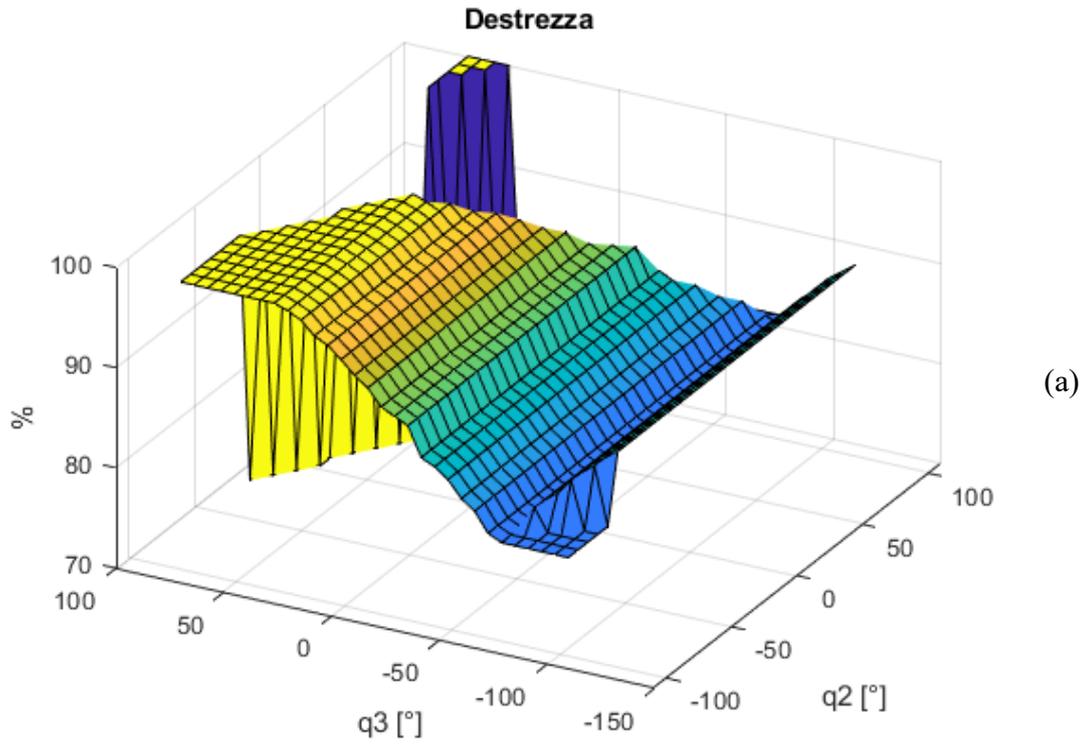


Figura 1.20 Grafico 3D (a) e mappa colorata (b) con andamento indice di destrezza al variare dei gradi di libertà q_2 e q_3 e mantenendo il grado di libertà $q_1 = 90^\circ$.

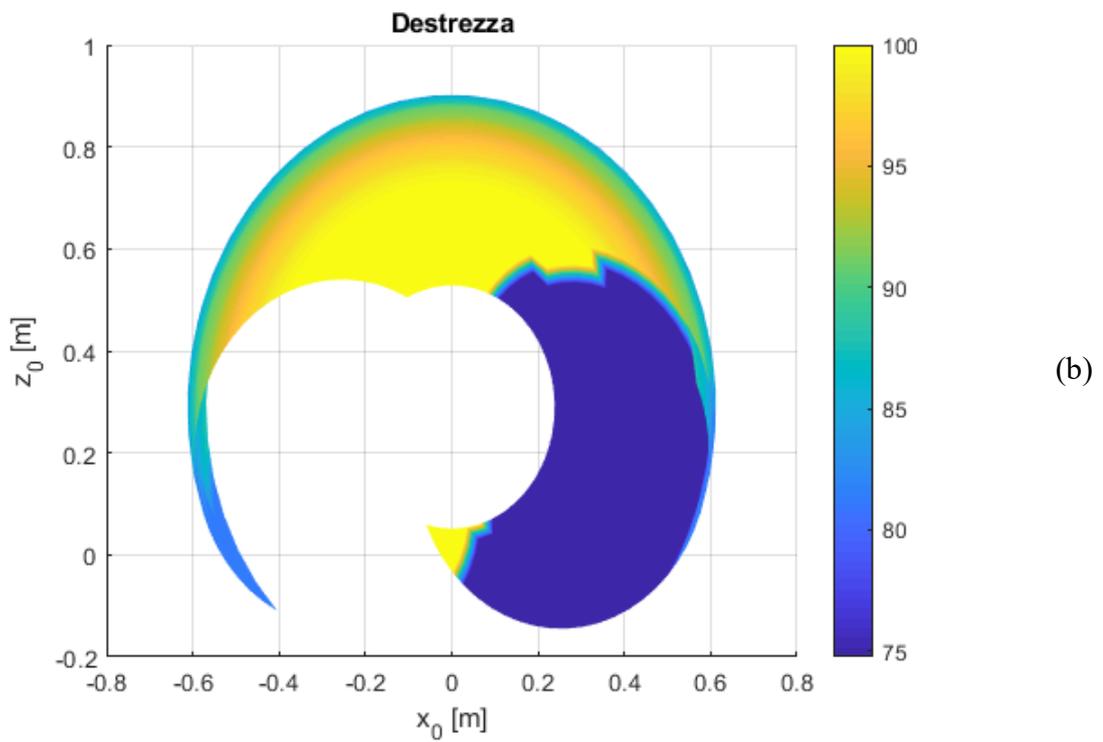
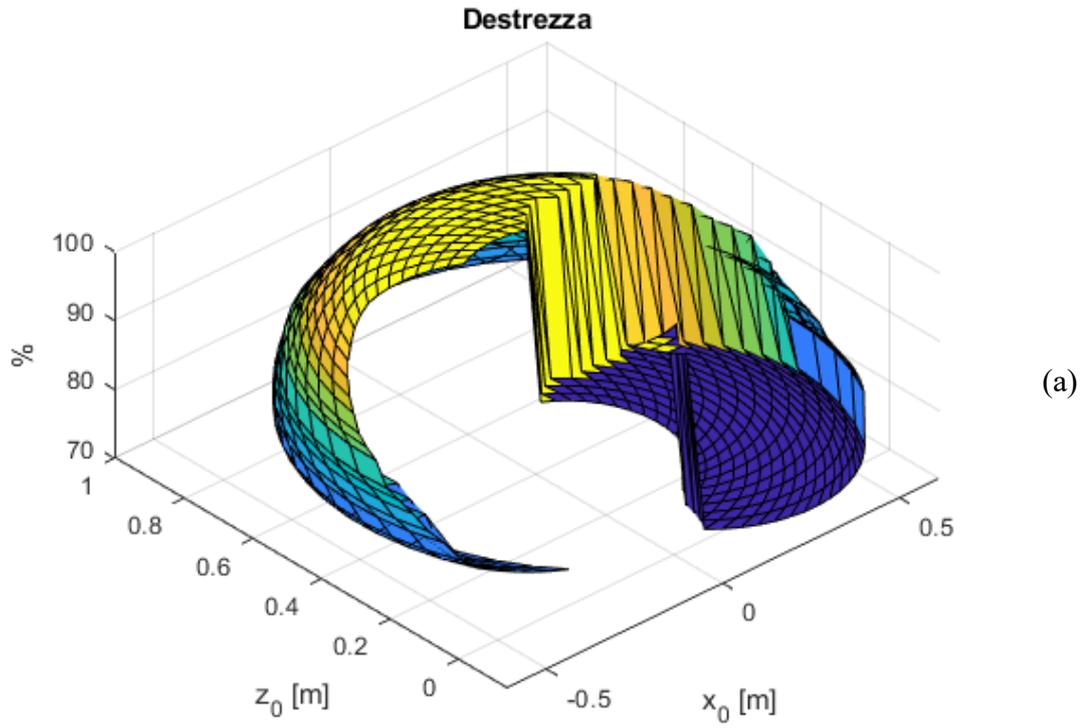


Figura 1.21 Grafico 3D (a) e mappa colorata (b) con andamento indice di destrezza al variare della posizione del centro polso su un piano con $q_1 = 90^\circ$.

1.5. Pianificazione di traiettoria Resolved Rate Motion Control

In questa sezione si è analizzato un algoritmo di pianificazione della traiettoria del tipo RRMC con logica closed e si è studiato, in particolare, il comportamento di tale algoritmo nel caso in cui il robot, durante la traiettoria, transiti in una condizione di singolarità.

L'algoritmo RRMC con logica closed consiste nel confrontare la posa stimata con quella desiderata. Questo metodo si basa sul calcolo della differenza tra la posa desiderata e la posa attuale del robot. La differenza tra le due pose è un'informazione di spostamento generalizzato che si richiede alla posa per potersi sovrapporre a quella desiderata. Il robot viene comandato non in base alla velocità generata dal pianificatore ma in base alla velocità generata dall'inseguimento della posa di riferimento [9]. In figura 1.22 è presentato lo schema dell'algoritmo realizzato tramite Simulink.

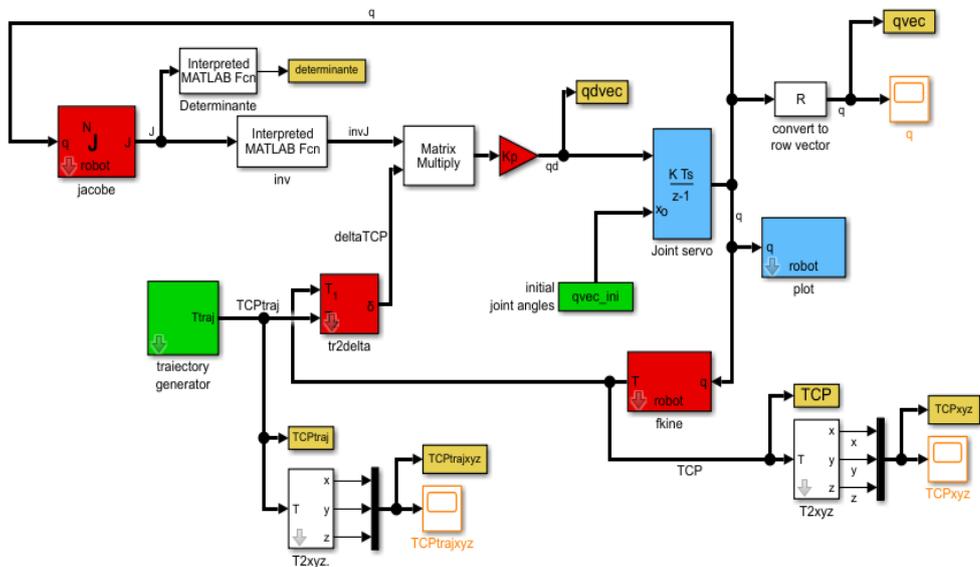


Figura 1.22 Schema Simulink dell'algoritmo RRMC logica closed.

L'algoritmo così implementato viene testato dapprima sulla traiettoria rettilinea dell'End Effector rappresentata in figura 1.23. In figura 1.24, invece, è riportato un grafico di confronto tra la traiettoria desiderata e quella ottenuta. Analizzando la traiettoria che si ottiene lungo l'asse y, si nota come, in corrispondenza di un tempo pari a circa 1 s, si hanno delle oscillazioni della soluzione. Tali oscillazioni sono generate dal fatto che in

quell'istante il robot sta transitando in una configurazione di singolarità (il centro polso si trova sull'origine degli assi) e quindi la matrice Jacobiana risulta essere mal condizionata. L'algoritmo, ad ogni modo non genera errori nel calcolo, in quanto, lavorando per istanti di tempo discreto, non calcola l'inversa della Jacobiana esattamente nel punto critico. Per essere sicuri di essere in condizione di singolarità è sufficiente analizzare l'andamento del determinante della Jacobiana, rappresentato in figura 1.25 e vedere come esso si annulli ad un tempo di circa 1s.

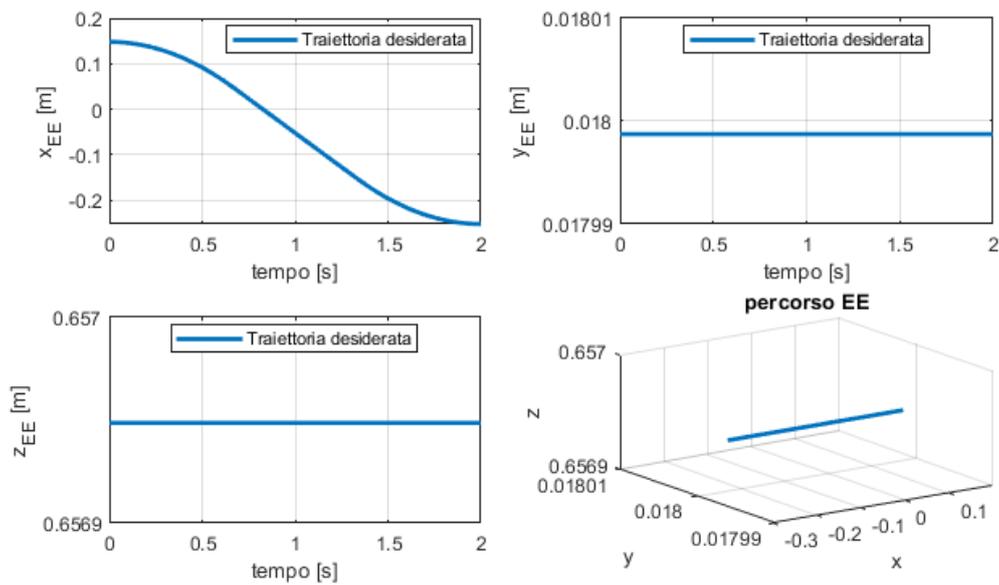


Figura 1.23 Traiettorie End Effector desiderate.

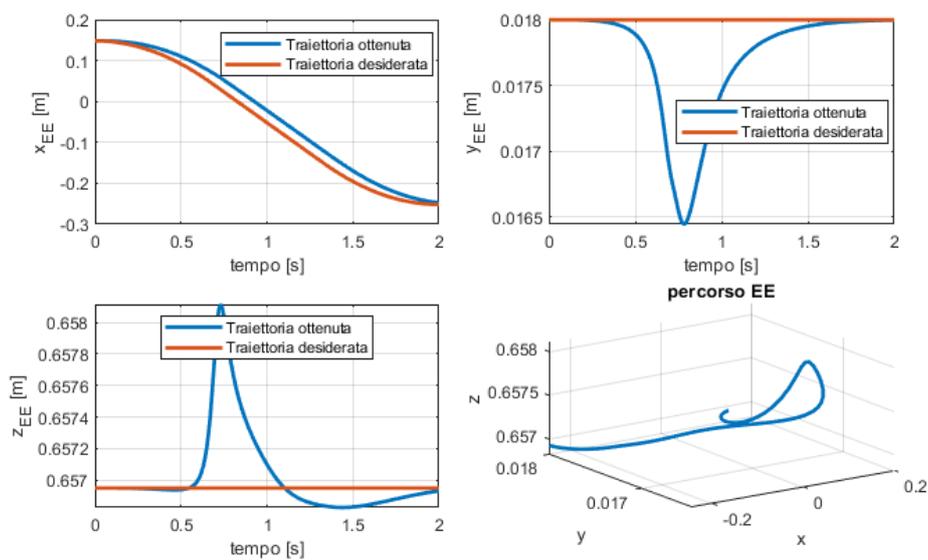


Figura 1.24 Confronto tra la traiettoria ottenuta (in blu) e la traiettoria desiderata (in rosso).

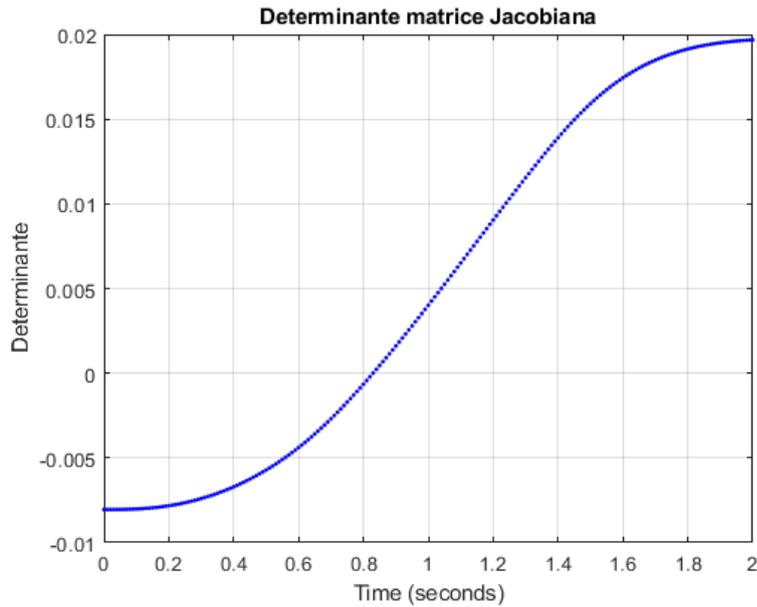


Figura 1.25 Andamento determinante matrice Jacobiana.

Per studiare meglio ciò che accade in singolarità, si analizza una traiettoria che porta l'End Effector a transitare nell'origine del sistema di riferimento fisso richiedendo, in quell'istante, un valore di velocità nella direzione y che il robot non è in grado di fornire. In figura 1.26 è visibile la traiettoria analizzata, mentre gli andamenti dei valori dei gradi di libertà e di velocità nei giunti sono rappresentati in figura 1.27.

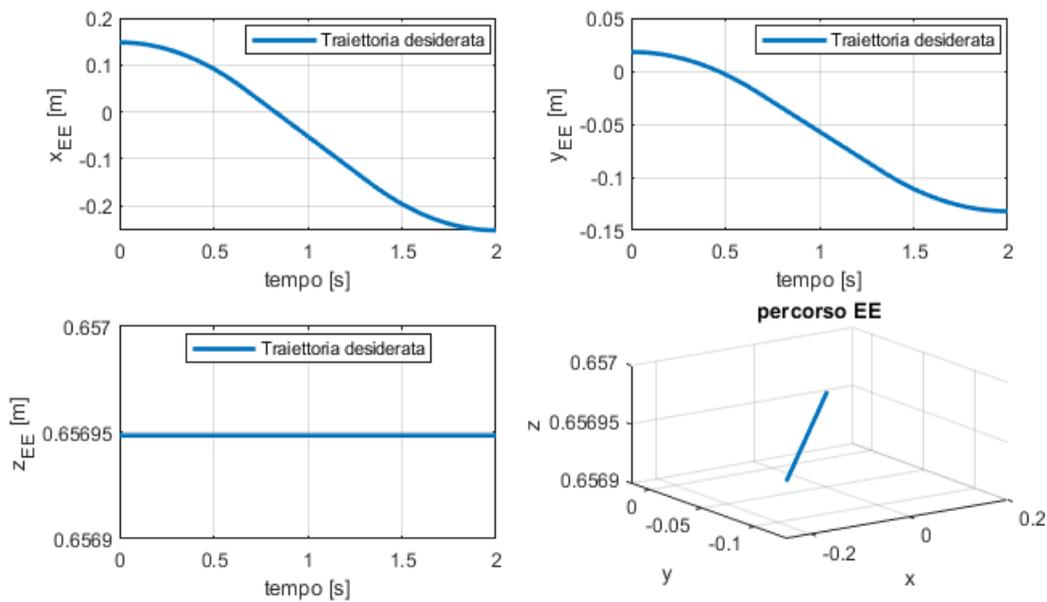
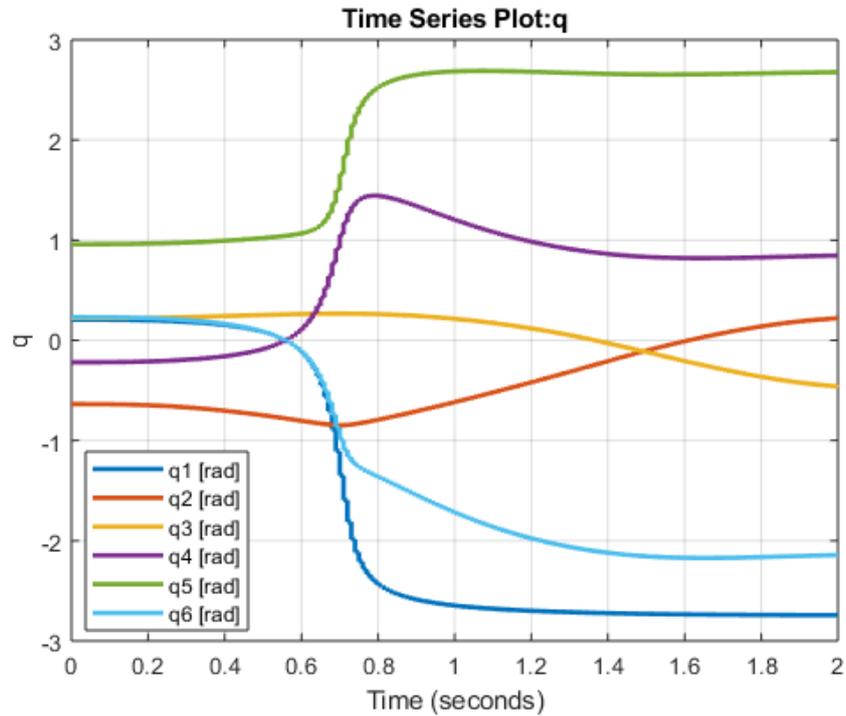
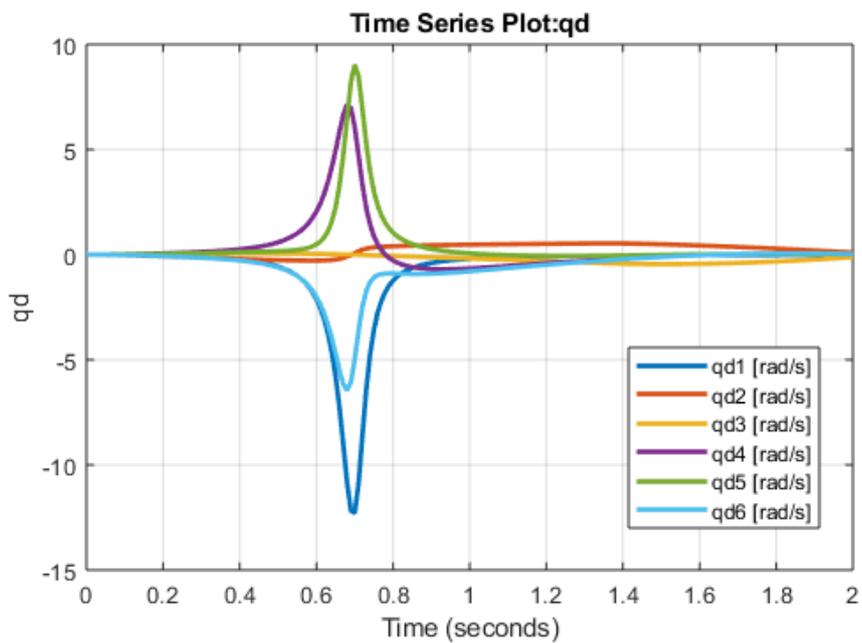


Figura 1.26 Traiettoria End Effector desiderata per produrre singolarità.



(a)



(b)

Figura 1.27 Andamento dei valori di gradi di libertà nei giunti (a) e delle velocità (b) nel caso di singolarità.

È possibile notare come dopo un tempo pari a circa 0.75 s ci sia una brusca variazione del valore dei gradi di libertà nei giunti, ovvero sia richiesta una elevata velocità ai motori. Questo risultato è indice del fatto che il robot si trova in condizioni prossime alla singolarità e che per compiere la traiettoria desiderata si sta richiedendo una velocità nello spazio

operativo con una componente nella direzione verso la quale si ha perso il grado di libertà. In figura 1.28 è riportato l'andamento del determinante della matrice Jacobiana. Si nota come esso tende a zero proprio nel medesimo istante in cui è richiesto il picco di velocità al robot.

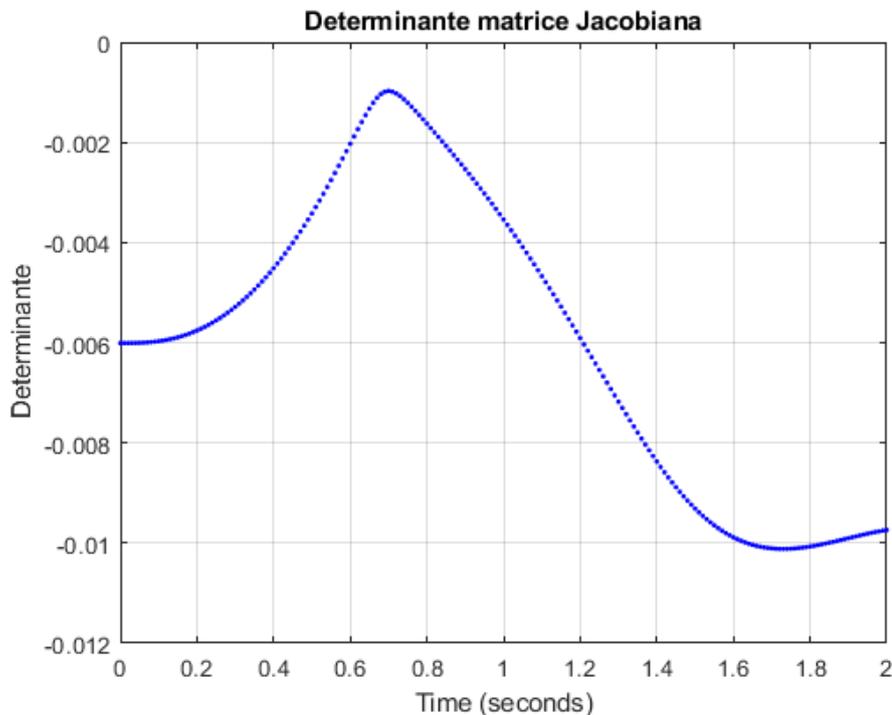


Figura 1.28 Andamento determinante matrice Jacobiana.

Confrontando i valori di velocità con i limiti massimi di velocità forniti a catalogo si può vedere come questi siano superiori. Per tenere conto di tale fatto si procede con attuare una modifica all'algoritmo inserendo un ulteriore blocco Simulink. In figura 1.29 è visibile lo schema Simulink con il blocco per tenere conto dei limiti di velocità (tale blocco è evidenziato dalla freccia rossa).

Applicando tale algoritmo si hanno dei nuovi andamenti dei valori dei gradi di libertà nei giunti e delle rispettive velocità. Tali andamenti sono riportati in figura 1.30. Poiché i valori dei gradi di libertà, nel momento in cui si considerano i limiti di velocità dei giunti, sono diversi da quelli necessari per avere la traiettoria desiderata, il risultato finale sarà quello di avere, come si può vedere in figura 1.31, una traiettoria ottenuta molto diversa da quella richiesta.

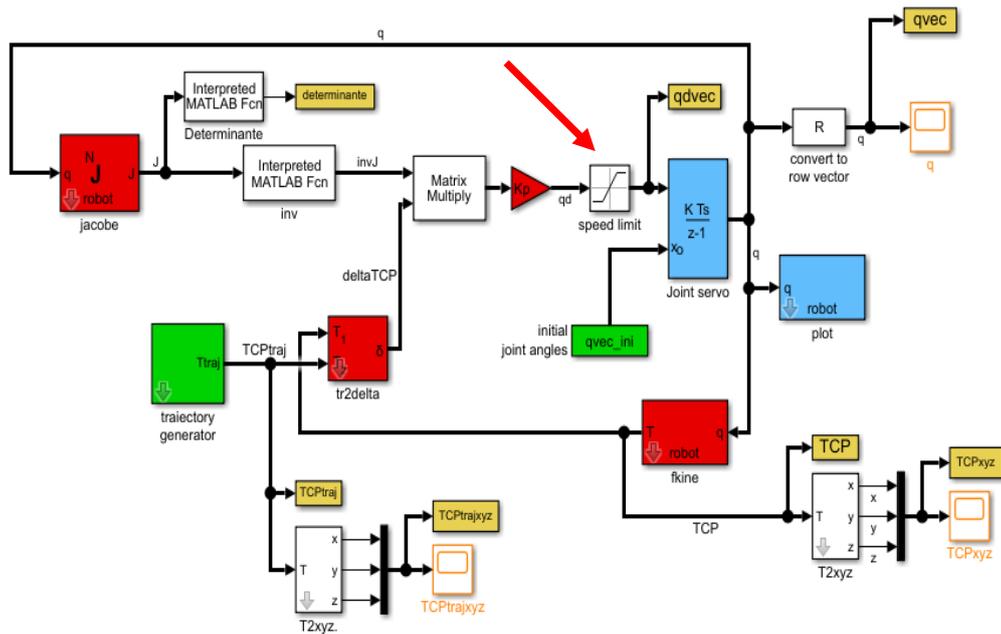


Figura 1.29 Schema Simulink dell'algoritmo RRMC closed loop con limiti di velocità nei giunti.

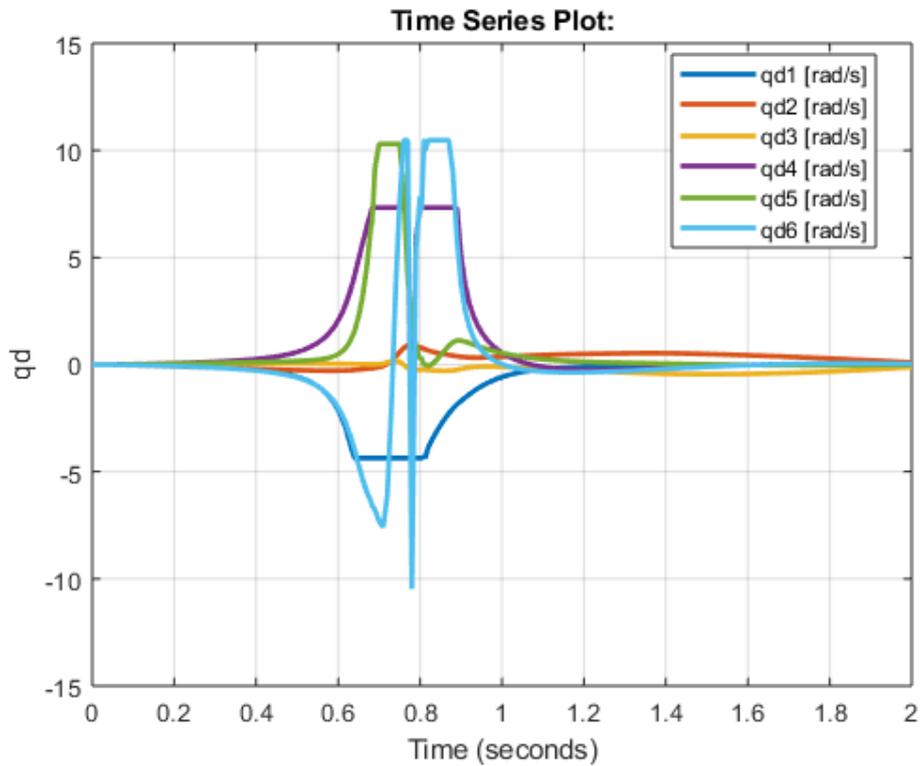


Figura 1.30 Andamento delle velocità nei giunti nel caso di singolarità in cui si considerano i limiti di velocità dei giunti.

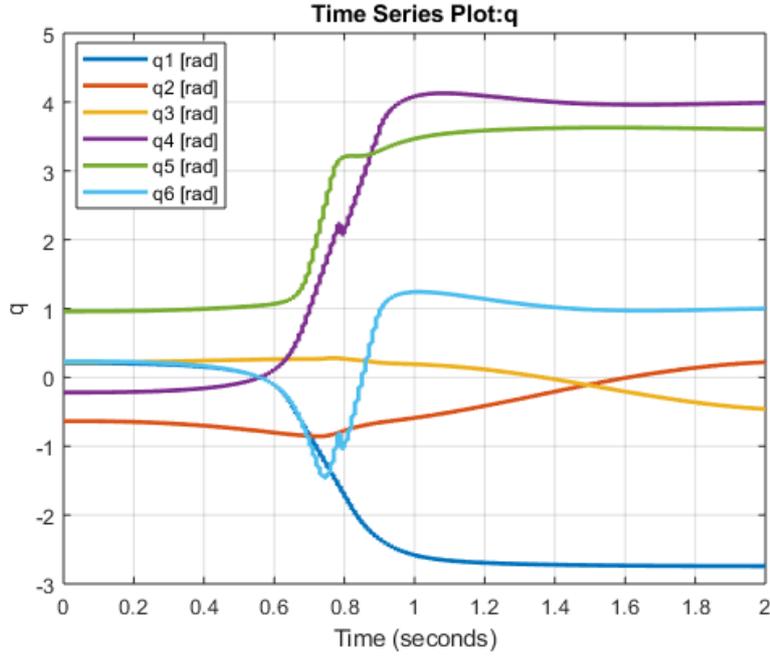


Figura 1.31 Andamento dei gradi di libertà nei giunti nel caso di singolarità in cui si considerano i limiti di velocità dei giunti.

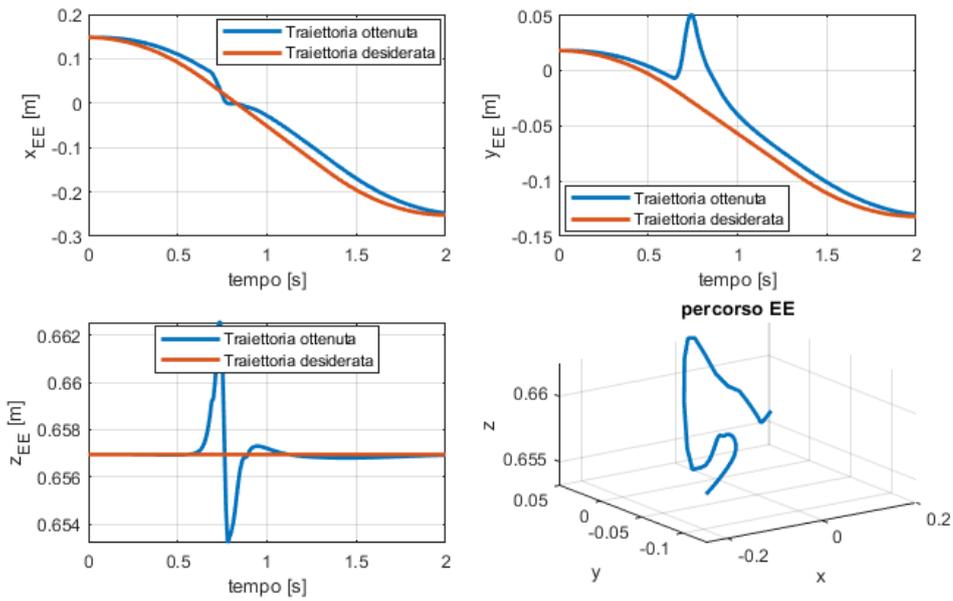


Figura 1.32 Confronto tra la traiettoria ottenuta (in blu) e la traiettoria desiderata (in rosso) nel caso di singolarità in cui si considerano i limiti di velocità nei giunti.

In figura 1.33 si riporta il confronto tra due configurazioni del robot all'istante di tempo 0.8 s nel caso in cui si considerano o meno i limiti di velocità nei giunti. Si nota come le due configurazioni siano notevolmente diverse.

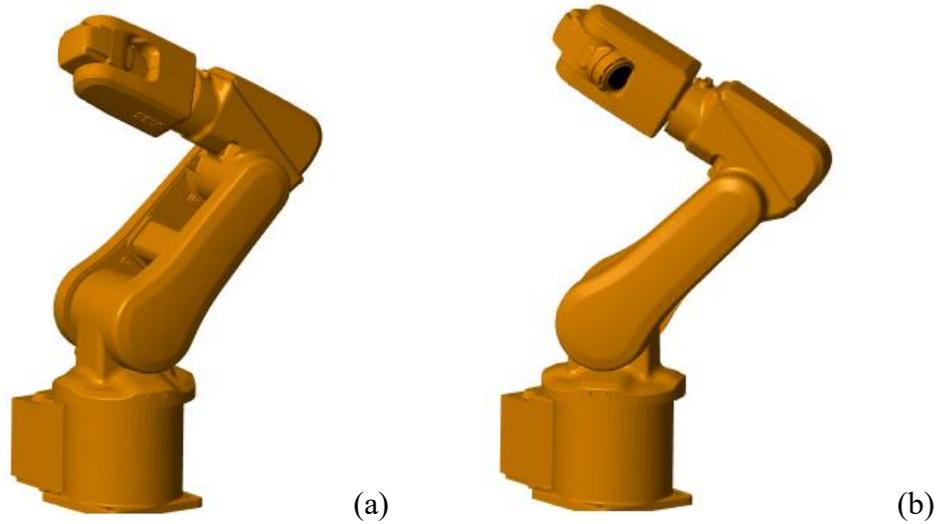


Figura 1.33 Confronto tra due configurazioni all'istante di tempo 0.8 s nel caso in cui non si considerano i limiti di velocità (a) e nel caso in cui si considerano tali limiti (b) in presenza di singolarità di braccio.

Per comprendere meglio il fenomeno delle singolarità è opportuno analizzare un caso più semplice, ovvero una situazione in cui la singolarità sia puramente di polso.

Per studiare il caso di singolarità di polso è stata realizzata una traiettoria opportuna, tale da portare il manipolatore a transitare in una configurazione in cui $q_5 = 0^\circ$ provocando l'allineamento degli assi 4 e 6. La suddetta traiettoria è riportata in figura 1.34.

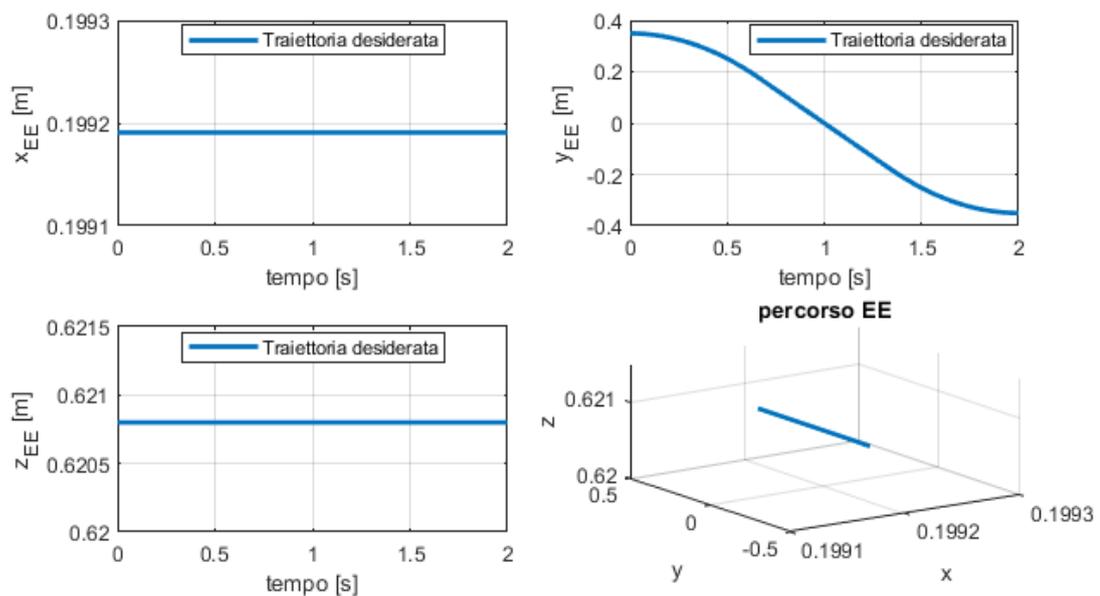


Figura 1.34 Traiettoria End Effector desiderata per produrre singolarità.

Per verificare che il manipolatore transita in una configurazione di singolarità si analizza l'andamento del determinante della matrice Jacobiana. Si nota che, per un tempo pari a $t=1.1s$, tale grandezza assume un valore molto prossimo allo zero. L'andamento del valore del determinante della matrice Jacobiana è riportato in figura 1.35.

In figura 1.36 e 1.37, invece, si riportano gli andamenti dei gradi di libertà nei giunti e delle relative velocità. È possibile notare come, in prossimità della condizione di singolarità, vi sia un aumento incontrollato di velocità dei giunti 4 e 6 corrispondente a una rotazione istantanea di tali assi, nei due versi opposti, di un angolo pari a 180° . Il movimento a elevate velocità di giunti non coinvolti nel moto è proprio il tipico fenomeno che si osserva nel momento in cui un manipolatore transita in una configurazione di singolarità.

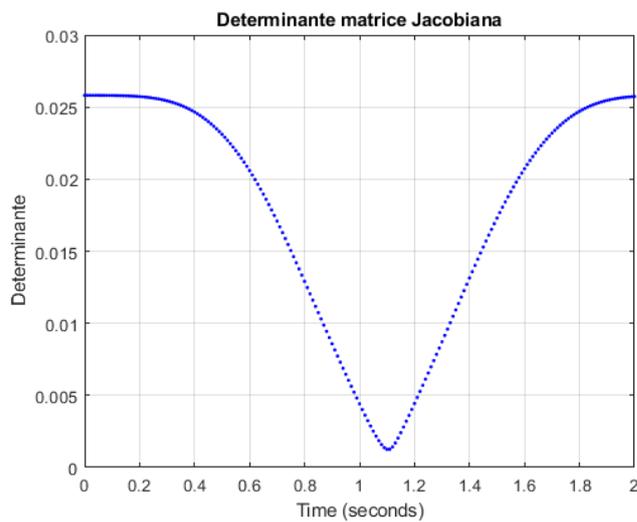


Figura 1.35 Andamento determinante matrice Jacobiana singolarità di polso.

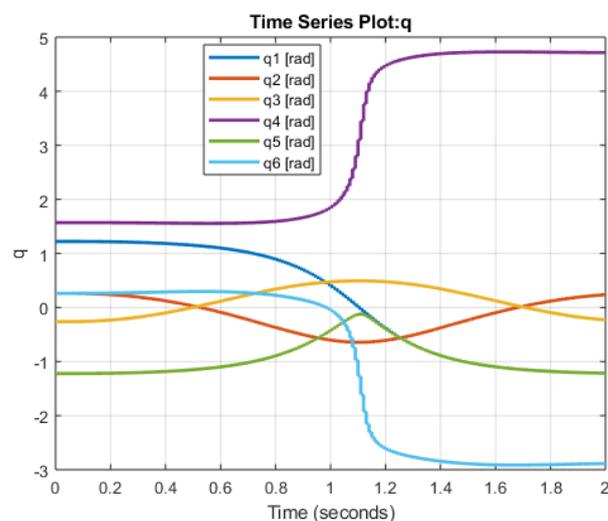


Figura 1.36 Andamento dei valori di gradi di libertà nei giunti nel caso di singolarità di polso.

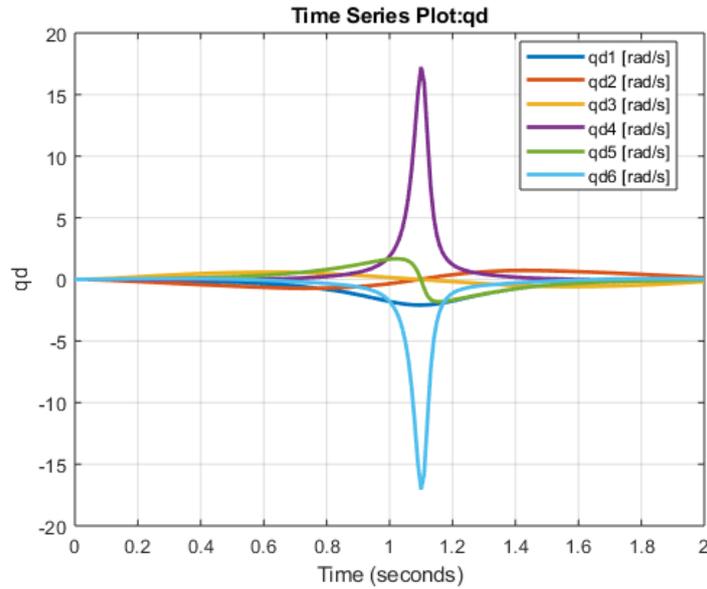


Figura 1.37 Andamento delle velocità nei giunti nel caso di singolarità di polso.

Andando a imporre i limiti di velocità sui giunti l'andamento delle velocità nei giunti varia ed è riportato in figura 1.38.

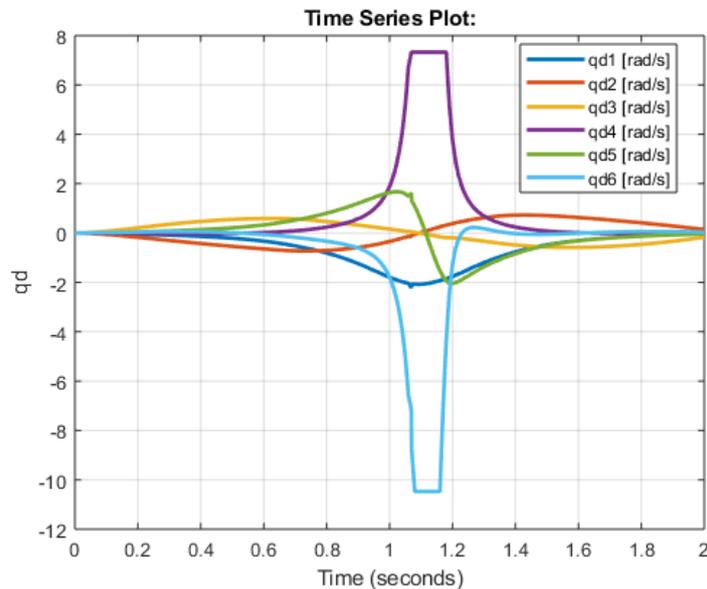


Figura 1.38 Andamento delle velocità nei giunti nel caso di singolarità di polso in cui si considerano i limiti di velocità dei giunti.

Andando a limitare le velocità nei giunti l'effetto che si ottiene è quello di un errore sulla traiettoria desiderata maggiore rispetto al caso in cui non si considerano tali limiti. In figura

1.38 viene mostrato il confronto tra la traiettoria desiderata e quella ottenuta. Si possono notare sia sull'asse X che sull'asse Z delle importanti deviazioni.

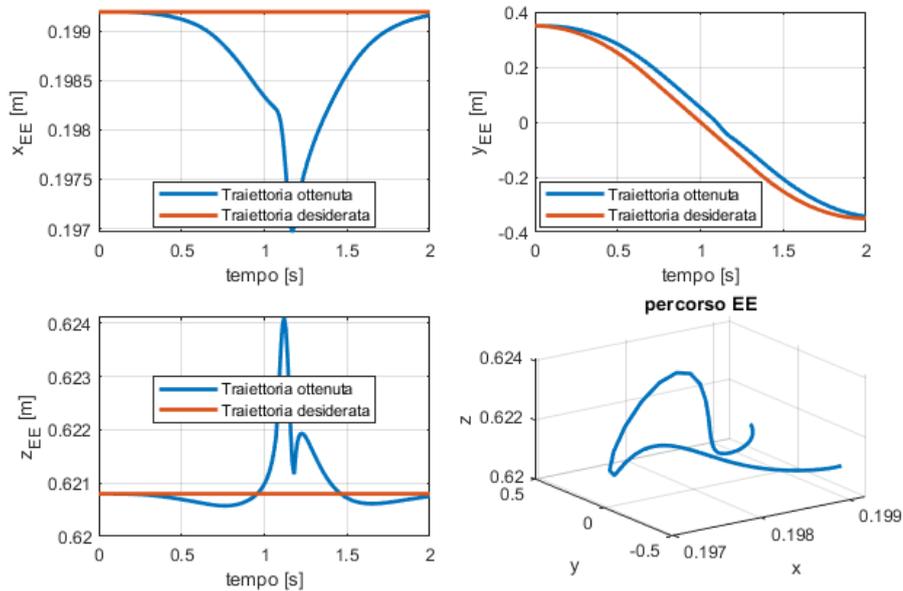


Figura 1.39 Confronto tra la traiettoria ottenuta (in blu) e la traiettoria desiderata (in rosso) nel caso di singolarità di polso in cui si considerano i limiti di velocità nei giunti.

In figura 1.40 si mostra il confronto tra due configurazioni all'istante di tempo 1.2 s nel caso in cui si considerano o meno i limiti di velocità nei giunti. Si nota anche in questo caso una differenza tra le due configurazioni anche se meno evidente rispetto al caso di singolarità di braccio analizzato in precedenza.

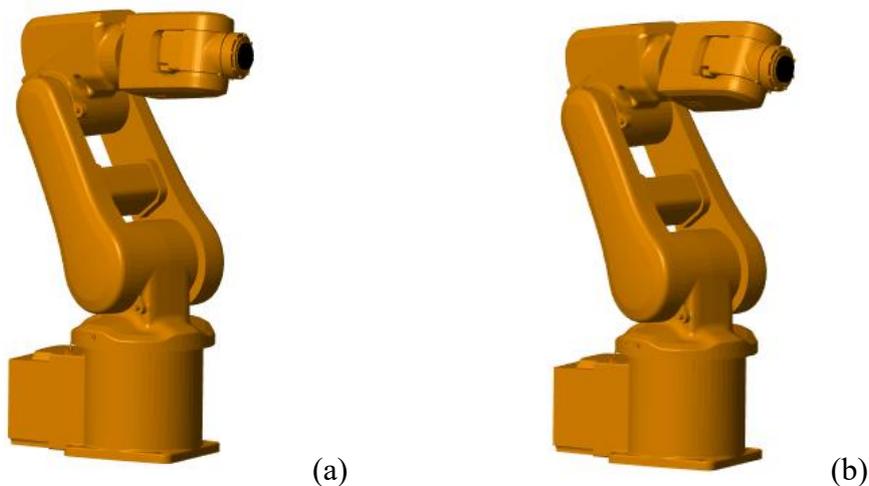


Figura 1.40 Confronto tra due configurazioni all'istante di tempo 1.2 s nel caso in cui non si considerano i limiti di velocità (a) e nel caso in cui si considerano tali limiti (b) in presenza di singolarità di polso.

CAPITOLO 2: Robot Kinova Jaco 2 7DOF

Il robot oggetto di studio è il Kinova Jaco2. Si tratta di un braccio robotico di tipologia collaborativa una volta ridondante nello spazio in quanto dotato di 7 giunti. In figura 2.1 è rappresentato il robot con indicazioni circa i nomi e la posizione dei link e dei giunti (attuatori).

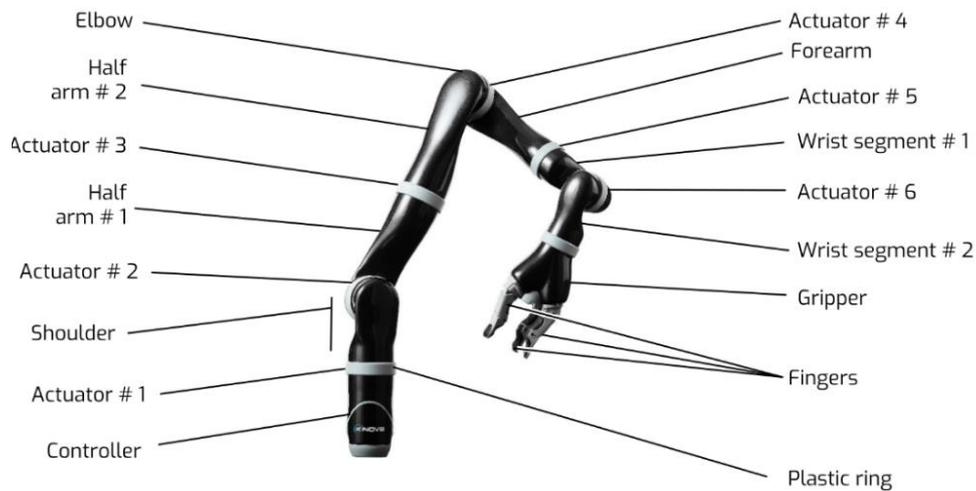


Figura 2.1 Kinova Jaco 2 7DOF [6].

Tabella 2.1 Limiti di giunto Kinova Jaco 2 [6].

Giunto	Limite minimo	Limite massimo
1	-10000°	10000°
2	47°	313°
3	-10000°	10000°
4	30°	330°
5	-10000°	10000°
6	65°	295°
7	-10000°	10000°

Nella figura 2.2 viene, invece, rappresentato il robot in una configurazione con il braccio completamente steso e sono riportate le quote geometriche dei vari componenti, gli assi di

rotazione dei vari giunti e il loro senso di rotazione. Mentre, per completare il set di informazioni iniziali utili alla modellizzazione e analisi del robot si riportano, nella tabella 2.1, anche i valori dei limiti di giunto.

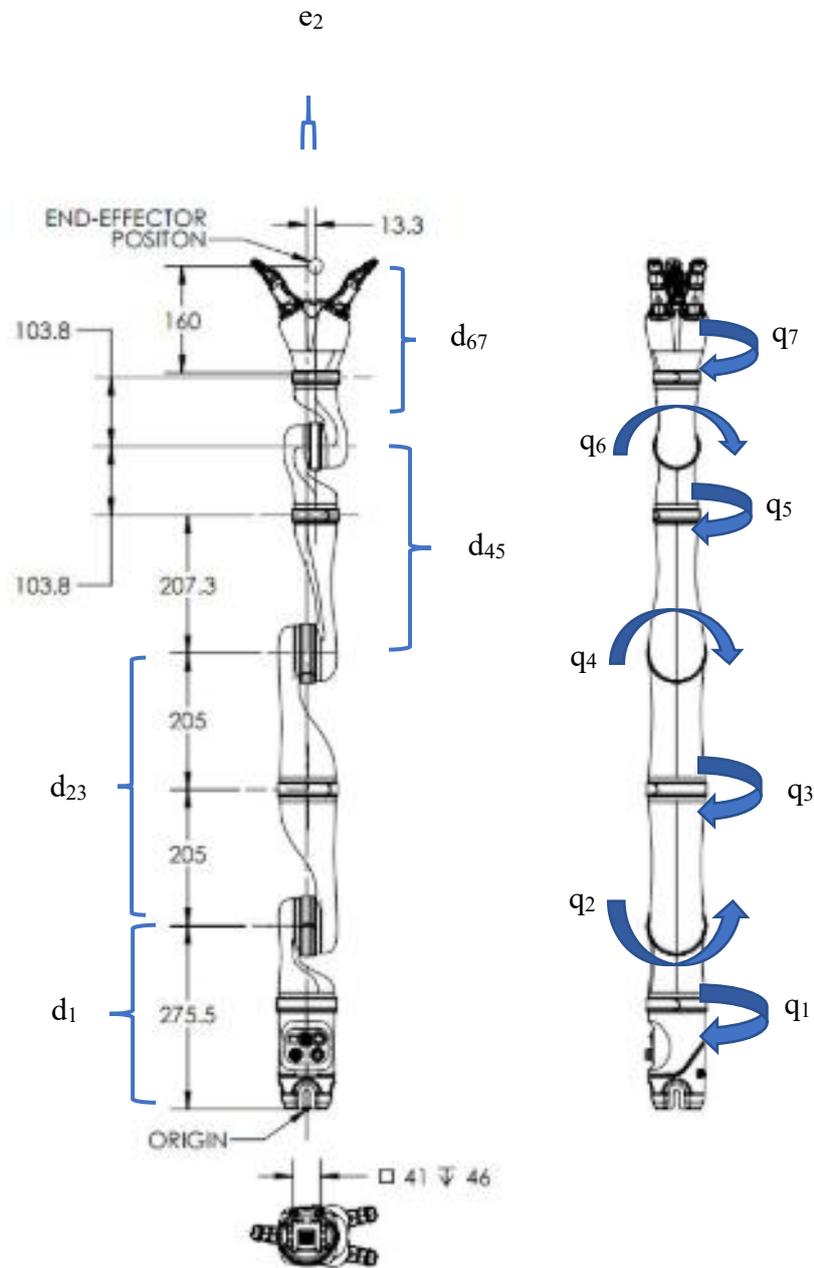


Figura 2.2 Quote geometriche, assi e direzione di rotazione giunti Kinova Jaco 2 [6].

2.1. Modellazione del robot

Partendo dalle informazioni di base di cui sopra, applicando la convenzione di D-H standard si sono assegnati i sistemi di riferimento a ogni link del manipolatore. In realtà, la configurazione rappresentata nella figura 2.2 non è la configurazione di zero del robot, la quale risulta essere molto complessa da rappresentare in quanto il braccio si trova del tutto ripiegato su sé stesso. Inoltre, analizzando i valori dei limiti di giunto, si nota come tale configurazione non sia fisicamente raggiungibile dal nostro manipolatore.

In figura 2.3 sono riportati i sistemi di riferimento di D-H tramite una rappresentazione stick figure del robot in cui tutti i gradi di libertà sono posti pari a 180° .

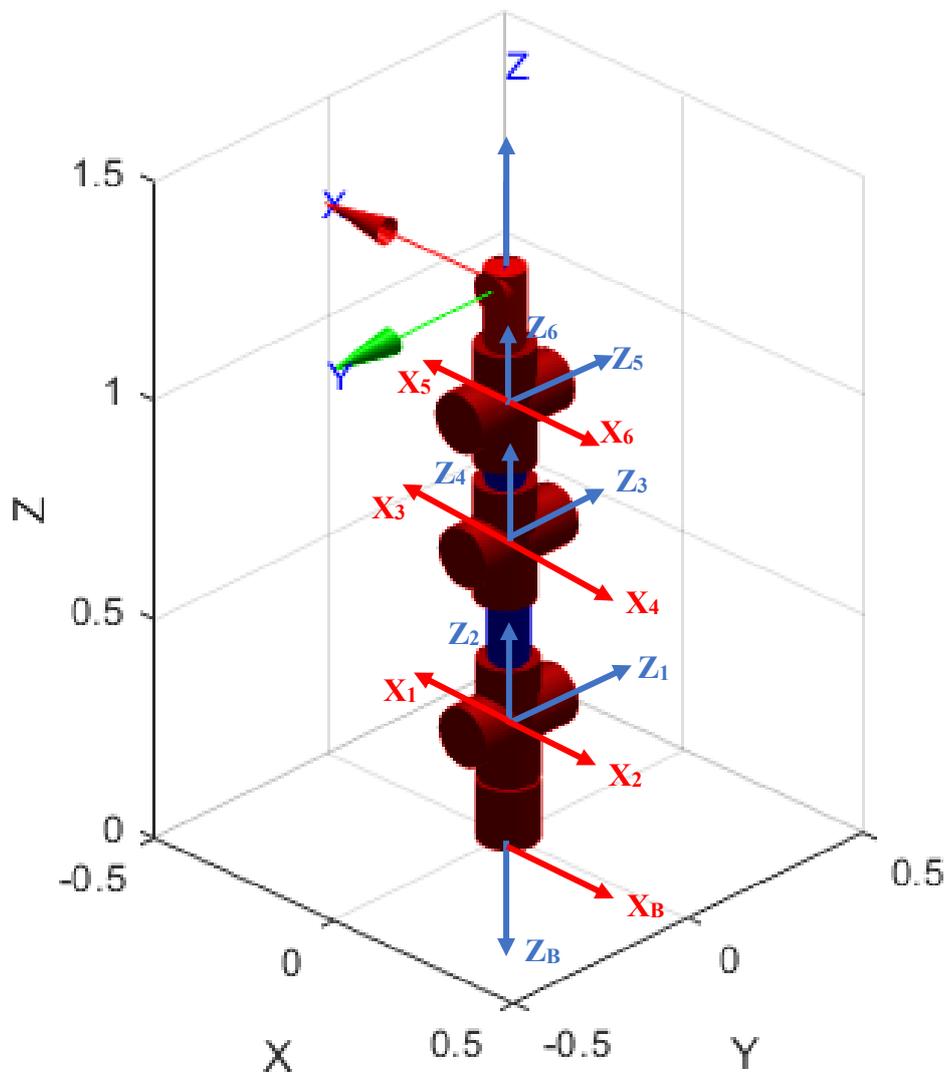


Figura 2.3 Sistemi di riferimento Denavit-Hartenberg.

Nella tabella 2.2 sono riportati i valori dei parametri di D-H.

Tabella 2.2 Parametri di Denavit-Hartenberg Kinova Jaco 2.

Link	α (deg)	a (mm)	d (mm)	ϑ (deg)
1	-90	0	$-d_1=-275.5$	q_1
2	-90	0	0	q_2
3	-90	0	$-d_{23}=-410$	q_3
4	-90	0	$-e_2=-9.8$	q_4
5	-90	0	$-d_{45}=-311.1$	q_5
6	-90	0	0	q_6
7	180	0	$-d_{67}=-263.8$	q_7

Le matrici di trasformazione omogenee risultano essere le seguenti:

$${}^B\hat{A}_1 = \begin{bmatrix} \cos(q_1) & 0 & -\sin(q_1) & 0 \\ \sin(q_1) & 0 & \cos(q_1) & 0 \\ 0 & -1 & 0 & -d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$${}^1\hat{A}_2 = \begin{bmatrix} \cos(q_2) & 0 & -\sin(q_2) & 0 \\ \sin(q_2) & 0 & \cos(q_2) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

$${}^2\hat{A}_3 = \begin{bmatrix} \cos(q_3) & 0 & -\sin(q_3) & 0 \\ \sin(q_3) & 0 & \cos(q_3) & 0 \\ 0 & -1 & 0 & -d_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

$${}^3\hat{A}_4 = \begin{bmatrix} \cos(q_4) & 0 & -\sin(q_4) & 0 \\ \sin(q_4) & 0 & \cos(q_4) & 0 \\ 0 & -1 & 0 & -e_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

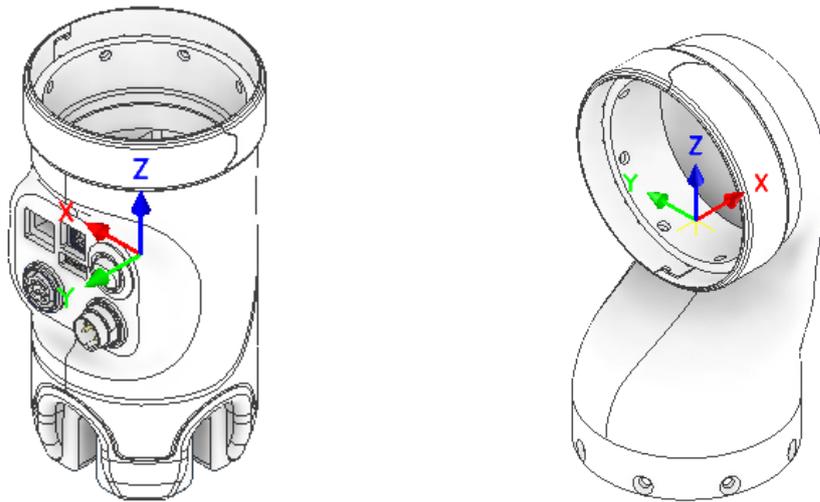
$${}^4\hat{A}_5 = \begin{bmatrix} \cos(q_5) & 0 & -\sin(q_5) & 0 \\ \sin(q_5) & 0 & \cos(q_5) & 0 \\ 0 & -1 & 0 & -d_{45} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$${}^5\hat{A}_6 = \begin{bmatrix} \cos(q_6) & 0 & -\sin(q_6) & 0 \\ \sin(q_6) & 0 & \cos(q_6) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

$${}^6\hat{A}_7 = \begin{bmatrix} \cos(q_7) & \sin(q_7) & 0 & 0 \\ \sin(q_7) & -\cos(q_7) & 0 & 0 \\ 0 & 0 & -1 & -d_{67} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

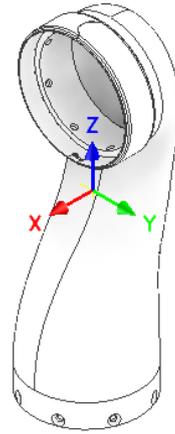
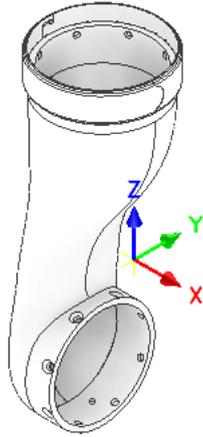
2.2. Rappresentazione virtuale del robot

Per poter avere una visualizzazione grafica migliore rispetto alla sola rappresentazione stick figure del robot è stato utilizzato l'ambiente di simulazione Simulink. Dato l'assieme Solidworks dello Jaco 2 fornito dal sito della casa costruttrice, esso è stato scomposto e, per ogni link (convertito in formato ".stl"), si è individuata la terna di riferimento CAD. Successivamente, tramite delle misure sul CAD, si è studiato il legame tra quest'ultima e le terne di riferimento scelte precedentemente con la convenzione di Denavit-Hartenberg. In figura 2.4 si riporta la posizione delle terne CAD per ogni link e le relative matrici di trasformazione.



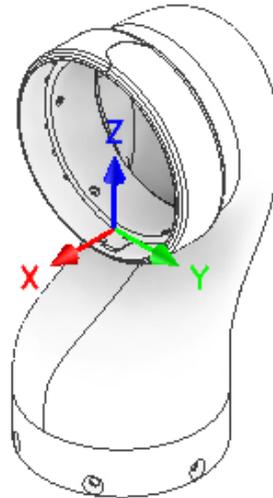
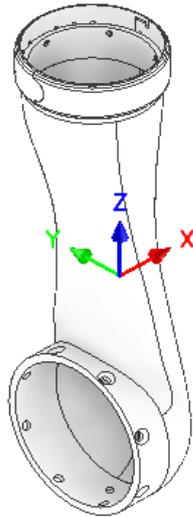
$${}^B \hat{A}_{CAD} = \begin{bmatrix} 1 & 0 & 0 & 0.003611 \\ 0 & -1 & 0 & -0.001667 \\ 0 & 0 & -1 & -0.082874 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1 \hat{A}_{CAD} = \begin{bmatrix} 0 & 1 & 0 & 0.004722 \\ 0 & 0 & 1 & -0.030342 \\ 1 & 0 & 0 & 0.010994 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



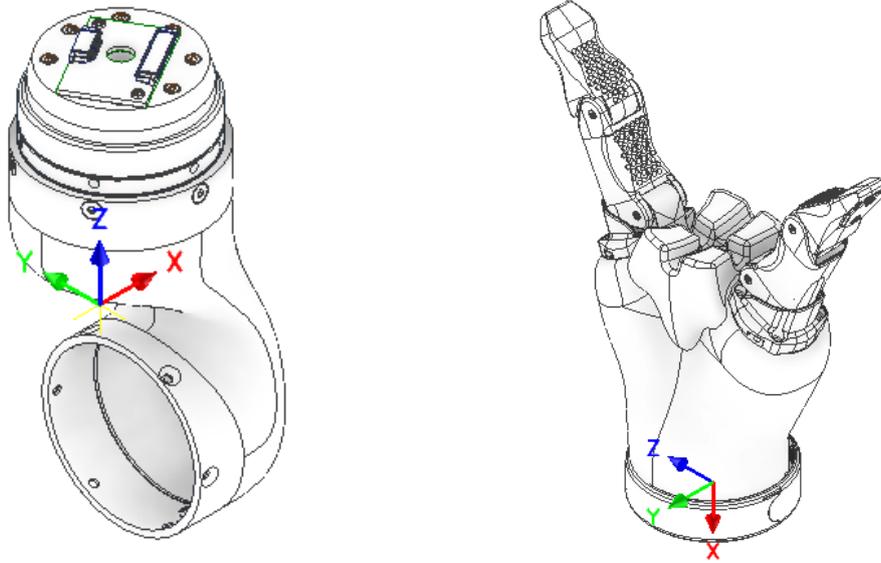
$${}^2\hat{A}_{CAD} = \begin{bmatrix} 0 & -1 & 0 & -0.00287 \\ -1 & 0 & 0 & -0.022194 \\ 0 & 0 & -1 & -0.081301 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3\hat{A}_{CAD} = \begin{bmatrix} 0 & -1 & 0 & 0.002315 \\ 0 & 0 & 1 & -0.073467 \\ -1 & 0 & 0 & 0.006694 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$${}^4\hat{A}_{CAD} = \begin{bmatrix} 0 & 1 & 0 & -0.005185 \\ 1 & 0 & 0 & 0.01213 \\ 0 & 0 & -1 & -0.08155 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^5\hat{A}_{CAD} = \begin{bmatrix} 0 & -1 & 0 & 0.006204 \\ 0 & 0 & 1 & -0.034496 \\ -1 & 0 & 0 & 0.003281 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

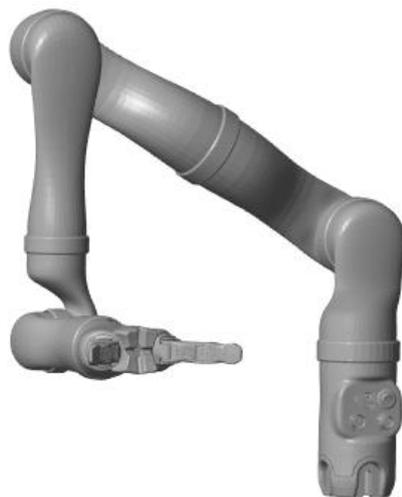


$${}^6\hat{A}_{CAD} = \begin{bmatrix} 0 & 1 & 0 & -0.001667 \\ 1 & 0 & 0 & -0.010337 \\ 0 & 0 & -1 & -0.051683 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^7(E E)\hat{A}_{CAD} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0.160 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 2.4 Modello CAD dei link [13] con rispettiva matrice di trasformazione tra sistema di riferimento Denavit-Hartenberg e sistema di riferimento CAD.

Nella figura 2.5 viene riportata una rappresentazione del robot, ottenuta tramite Simulink, in due configurazioni diverse.



(a)

[180° 240° 0° 60° 90° 90° 180°]



(b)

[180° 180° 180° 180° 180° 180° 180°].

Figura 2.5 Rappresentazione realistica del robot tramite ambiente Simulink in due configurazioni diverse.

2.3. Analisi del Workspace

Facendo variare il valore dei gradi di libertà q_2 e q_4 all'interno dei loro range possibili si ottiene una rappresentazione delle posizioni effettivamente raggiungibili dal centro polso del robot. Tale rappresentazione è visibile in figura 2.6.

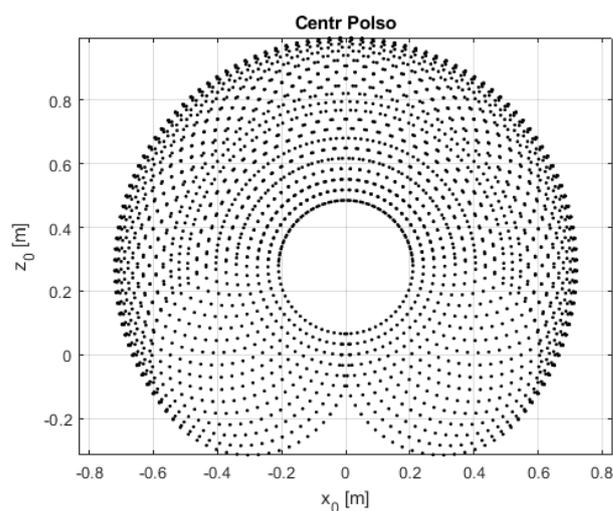


Figura 2.6 Posizioni raggiungibili dal centro polso del robot facendo variare q_2 e q_4 mantenendo $q_1 = 0^\circ$.

Facendo, invece, variare il valore dei gradi di libertà q_5 e q_6 all'interno dei loro range possibili si ottiene una rappresentazione delle posizioni effettivamente raggiungibili dall'End Effector del robot data una posizione del centro polso. Tale rappresentazione è visibile in figura 2.7.

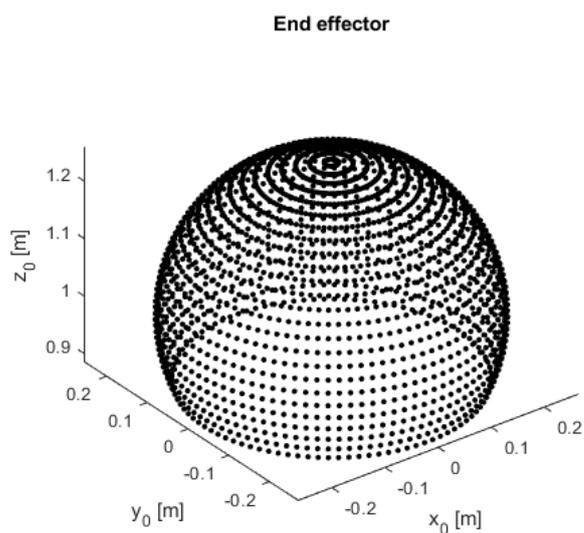


Figura 2.7 Posizioni raggiungibili dall'End Effector data una posizione fissa del centro polso.

2.4. Modello semplificato di un manipolatore a 7 gradi di libertà

L'obiettivo della seguente sezione è quello di creare un modello semplificato del manipolatore a 7 gradi di libertà oggetto di studio. In questo modo è possibile analizzare, attraverso l'utilizzo di un solo parametro, le infinite configurazioni che il manipolatore può assumere data una certa posa dell'End Effector. Tale parametro prende il nome di angolo di Swivel [2].

Analizzando nel dettaglio la geometria del robot si possono notare diverse peculiarità e individuare dei punti notevoli all'interno della sua struttura. I punti in questione sono 3:

- **Punto di centro spalla P_s** : punto in cui si intersecano gli assi dei primi tre gradi di mobilità. La posizione di tale punto sarà sempre fissa al variare della configurazione dei giunti.
- **Punto di centro polso P_w** : punto in cui si intersecano gli assi degli ultimi tre gradi di mobilità. La posizione di tale punto sarà sempre fissa per una fissata posizione dell'End Effector
- **Punto di centro gomito P_e** : punto individuato dall'origine del sistema di riferimento di D-H del link 3.

Nella figura 2.8 viene riportata una rappresentazione schematica del manipolatore in cui vengono rappresentati i tre punti appena citati e alcune grandezze di significativo valore per lo sviluppo successivo del modello.

Dove con L_1 si definisce la distanza tra centro spalla e centro gomito. Tale distanza è fissa e pari alla lunghezza del link d_{23} . Anche la distanza L_2 è fissa e pari a:

$$L_2 = \sqrt{d_{45}^2 + e_2^2} \quad (2.8)$$

La distanza L_{ws} tra centro spalla e centro polso è invece calcolabile dalla seguente relazione:

$$L_{ws} = \left\| {}^0\underline{p}_w - {}^0\underline{p}_s \right\| \quad (2.9)$$

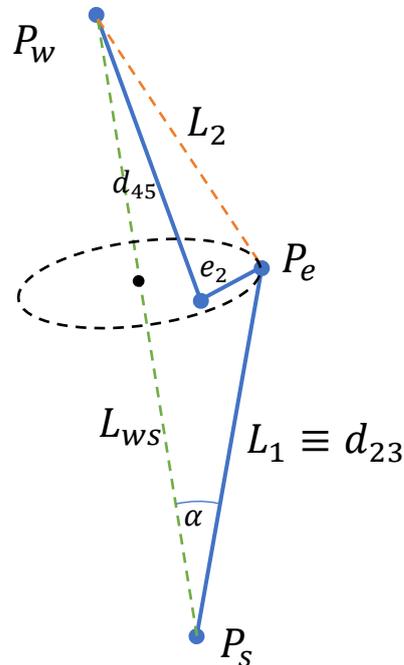


Figura 2.8 Rappresentazione schematica del manipolatore.

Dove la posizione del centro spalla rispetto al sistema di riferimento 0 è il vettore:

$${}^0\underline{p}_s = \begin{bmatrix} 0 \\ 0 \\ -d_1 \end{bmatrix} \quad (2.10)$$

Mentre la posizione del centro polso è calcolabile a partire dalla posizione dell'End Effector come specificato nel capitolo 2.6.

Per avere una visione più chiara della geometria del manipolatore nella figura 2.9 si riportano le viste del modello su due piani perpendicolari tra di loro.

Con il termine L_{WSL} si definisce la distanza tra centro polso e centro spalla proiettata su un piano perpendicolare al segmento e_2 ed è pari a:

$$L_{WSL} = \sqrt{L_{ws}^2 + e_2^2} \quad (2.11)$$

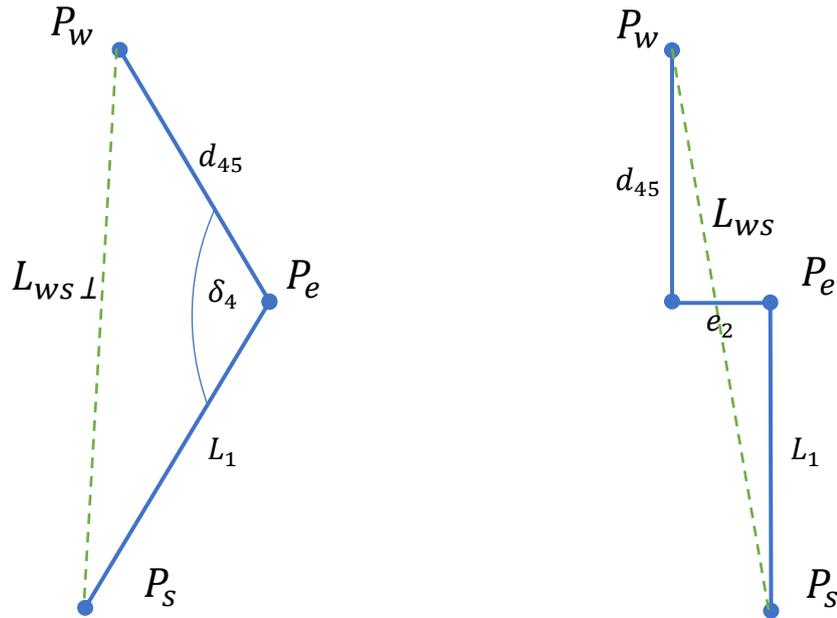


Figura 2.9 Viste su due piani ortogonali del modello geometrico del robot.

Conoscendo i valori delle grandezze precedentemente calcolate, applicando il teorema del coseno, è possibile ricavare il valore del coseno dell'angolo δ_4 , il quale sarà costante per tutte le infinite posizioni del centro gomito per una data posizione dell'End Effector.

$$\cos\delta_4 = \frac{L_1^2 + d_{45}^2 - L_{WS}^2}{2L_1 d_{45}} \quad (2.12)$$

Si può quindi affermare che, le infinite posizioni che il gomito può assumere si trovano su una circonferenza giacente su un piano perpendicolare alla congiungente centro polso-centro spalla e con centro su tale segmento. Il modo per determinare l'equazione di tale circonferenza verrà descritto di seguito.

Infine, sempre tramite il teorema del coseno, è possibile calcolare anche il valore del coseno dell'angolo α compreso tra L_{WS} e L_1 .

$$\cos\alpha = \frac{L_1^2 + L_{WS}^2 - L_2^2}{2L_1 L_{WS}} \quad (2.13)$$

2.5. Sistema di riferimento per lo studio della ridondanza

Per poter definire in modo più agevole la circonferenza di Swivel e le posizioni che il gomito può assumere nello spazio data una determinata posa dell'End Effector si introduce un sistema di riferimento ausiliario r . L'origine di tale sistema di riferimento si trova nel punto di centro spalla, l'asse x è rivolto lungo la congiungente centro spalla-centro polso, l'asse y è diretto dal centro spalla verso la posizione del gomito presa come riferimento e l'asse z è diretto in modo da formare una terna destrorsa di assi. È necessario precisare che la posizione del gomito di riferimento $P_{e,rif}$, è la posizione del gomito nella configurazione scelta per attribuire il valore zero all'angolo di Swivel.

Nella figura 2.10 si riporta una schematizzazione di tale sistema di riferimento.

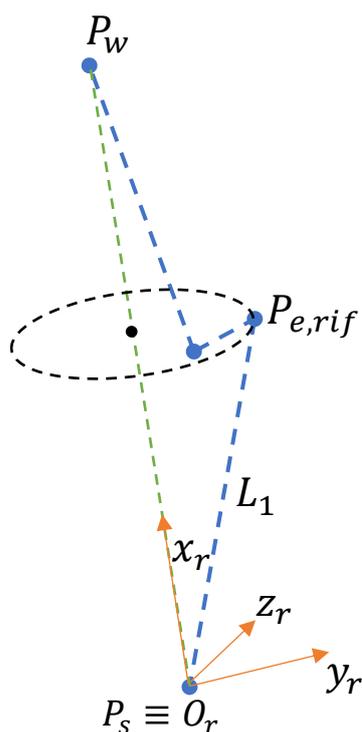


Figura 2.10 Rappresentazione del sistema di riferimento ausiliario r .

Per ottenere tale sistema di riferimento si parte da un sistema di riferimento ausiliario sovrapposto al sistema di riferimento 0, trasladandolo e ruotandolo opportunamente.

Come prima operazione si applica al sistema di riferimento ausiliario una traslazione nello spazio in modo da portare l'origine O_r a coincidere con il punto di centro spalla. La matrice di trasformazione che applicherà tale traslazione è la seguente:

$$\text{Trâs}(\underline{p}_s) = \begin{bmatrix} 1 & 0 & 0 & p_s(x) \\ 0 & 1 & 0 & p_s(y) \\ 0 & 0 & 1 & p_s(z) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

Applicando tale trasformazione si ottiene un nuovo sistema di riferimento $x'-y'-z'$ orientato come il sistema di riferimento 0 ma centrato nel centro spalla.

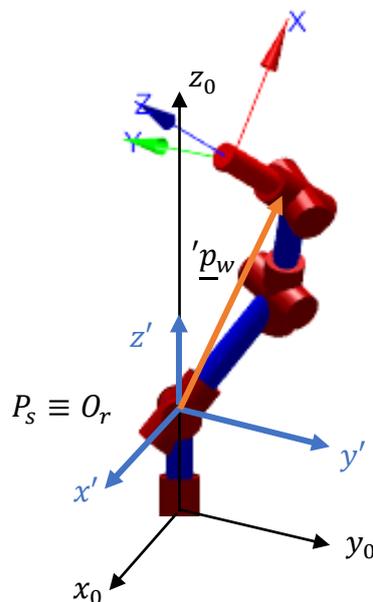


Figura 2.11 Traslazione del sistema di riferimento per ottenere il sistema di riferimento $x'-y'-z'$.

È utile, per le trasformazioni successive, esprimere la posizione del centro polso rispetto al nuovo sistema di riferimento.

$$\underline{{}'p}_w = \text{Trâs}(\underline{p}_s)^{-1} \underline{{}^0p}_w \quad (2.15)$$

Tale vettore, come detto, è utile per calcolare il valore degli angoli di rotazione da applicare al nuovo sistema di riferimento affinché esso coincida con il sistema di riferimento desiderato.

La prima rotazione sarà intorno all'asse $z=z'$ di un angolo pari a:

$$\lambda = \text{atan2}(\underline{p}_w(y'), \underline{p}_w(x')) \quad (2.16)$$

Con il valore di tale angolo si costruisce la matrice di rotazione $R\hat{t}(z', \lambda)$, applicando la quale, si ottiene un nuovo sistema di riferimento $x''-y''-z''$.

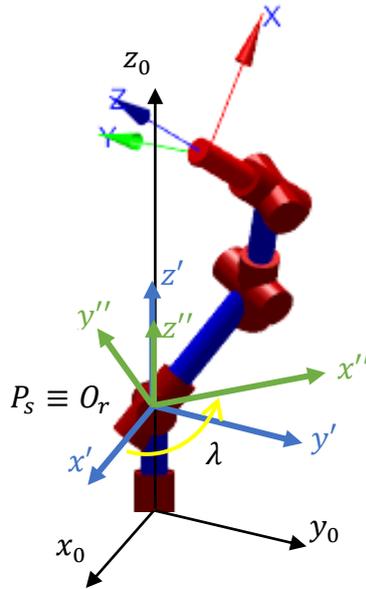


Figura 2.12 Rotazione del sistema di riferimento $x'-y'-z'$ per ottenere il sistema di riferimento $x''-y''-z''$.

A tale sistema di riferimento si applica poi una rotazione intorno all'asse y'' di un angolo pari a:

$$\mu = 2\pi - \text{atan2}\left(\underline{p}_w(z'), \sqrt{\underline{p}_w(x')^2 + \underline{p}_w(y')^2}\right) \quad (2.17)$$

Da cui si ricava la matrice $R\hat{t}(y'', \mu)$.

L'ultima rotazione necessaria per far coincidere il sistema di riferimento ausiliario con il sistema di riferimento r cercato è quella intorno all'asse x''' . L'angolo di rotazione sarà pari a:

$$\psi = \psi_0 + \varphi \quad (2.18)$$

Dove φ e varia da 0° a 360° al variare della posizione di gomito sulla circonferenza, mentre ψ_0 ha un valore fisso e indica l'angolo di cui bisogna ruotare il sistema di riferimento $x'''-y'''-z'''$ perché esso coincida con la configurazione di riferimento, ovvero la configurazione in cui l'angolo di Swivel vale zero.

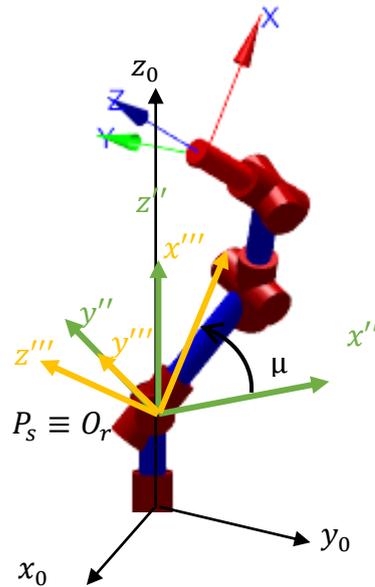


Figura 2.13 Rotazione del sistema di riferimento $x''-y''-z''$ per ottenere il sistema di riferimento $x'''-y'''-z'''$

Dove φ e varia da 0° a 360° al variare della posizione di gomito sulla circonferenza, mentre ψ_0 ha un valore fisso e indica l'angolo di cui bisogna ruotare il sistema di riferimento $x'''-y'''-z'''$ perché esso coincida con la configurazione di riferimento, ovvero la configurazione in cui l'angolo di Swivel vale zero.

La configurazione di riferimento coincide con la configurazione in cui la proiezione del versore perpendicolare al piano passante per P_s, P_w, P_e sul piano x_0-y_0 coincide con l'asse y_0 . Poiché il punto di centro gomito non è fisso ma varia lungo una circonferenza si potrà individuare un fascio di infiniti piani passanti per la retta che congiunge il centro polso con il centro spalla.

Partendo dall'equazione di un generico piano espressa nel sistema di riferimento fisso 0:

$$\pi: a_\pi x_0 + b_\pi y_0 + c_\pi z_0 + d_\pi = 0 \quad (2.19)$$

E conoscendo le coordinate dei tre punti per cui tale piano deve passare, si possono calcolare i vari coefficienti risolvendo la seguente relazione:

$$\det \begin{pmatrix} x - p_s(x) & y - p_s(y) & z - p_s(z) \\ p_e(x) - p_s(x) & p_e(y) - p_s(y) & p_e(z) - p_s(z) \\ p_w(x) - p_s(x) & p_w(y) - p_s(y) & p_w(z) - p_s(z) \end{pmatrix} = 0 \quad (2.20)$$

Conoscendo i valori dei coefficienti risulta immediato trovare un vettore perpendicolare al piano considerato in quanto esso è proprio formato dai coefficienti a_π, b_π, c_π .

$$\underline{n}_\pi = \begin{bmatrix} a_\pi \\ b_\pi \\ c_\pi \end{bmatrix} \quad (2.21)$$

La proiezione di tale vettore sul piano x_0-y_0 sarà lo stesso vettore ma con la terza componente pari a zero. Tale vettore verrà chiamato $\underline{n}_{\pi 0}$ e per semplicità viene normalizzato.

$$\underline{n}_{\pi 0} = \frac{\begin{bmatrix} a_\pi \\ b_\pi \\ 0 \end{bmatrix}}{\left\| \begin{bmatrix} a_\pi \\ b_\pi \\ 0 \end{bmatrix} \right\|} \quad (2.22)$$

Ora, come detto, la configurazione scelta come quella in cui l'angolo di Swivel vale zero è quella in cui il versore $\underline{n}_{\pi 0}$ è coincidente con l'asse y_0 , ovvero vale:

$$\underline{n}_{\pi 0}^* = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (2.23)$$

Uguagliando le due espressioni trovate, si calcolano i coefficienti e, di conseguenza, l'equazione del piano passante per $P_s, P_w, P_{e,rif}$.

$$\begin{cases} \underline{n}_{\pi 0}^*(x_0) = 0 \\ \underline{n}_{\pi 0}^*(y_0) = 1 \\ \underline{n}_{\pi 0}^*(z_0) = 0 \end{cases} \quad (2.24)$$

Tale uguaglianza non può essere verificata quando il punto di centro spalla e il punto di centro polso si trovano alla medesima quota ovvero il segmento L_{WS} è parallelo al piano x_0 - y_0 . In questo caso, non esistendo nessun vettore parallelo all'asse y_0 , si sceglie come configurazione per cui l'angolo di Swivel vale zero quella che presenta il gomito con la coordinata z_0 inferiore.

Una volta individuata la posizione del gomito di riferimento è possibile calcolare il valore dell'angolo ψ_0 che risulta pari a:

$$\psi_0 = \text{atan2} \left(P_{e,rif}(z'''), P_{e,rif}(y''') \right) \quad (2.25)$$

Da cui si ricava la matrice di rotazione $R\hat{ot}(x''', \psi)$.

In totale, la matrice di trasformazione da sistema di riferimento fisso 0 a sistema di riferimento r è la seguente:

$${}^0\hat{T}_r = \text{Tr}\hat{a}s(\underline{p}_s) R\hat{ot}(z', \lambda) R\hat{ot}(y'', \mu) R\hat{ot}(x''', \psi) \quad (2.26)$$

Dove i tre angoli di rotazione sono riassunti nella tabella 2.3.

Tabella 2.3 Angoli di rotazione per ottenere il sistema di riferimento r .

λ	$\text{atan2}(\underline{p}_w(y'), \underline{p}_w(x'))$
μ	$2\pi - \text{atan2} \left(\underline{p}_w(z'), \sqrt{\underline{p}_w(x')^2 + \underline{p}_w(y')^2} \right)$
ψ	$\text{atan2} \left(P_{e,rif}(z'''), P_{e,rif}(y''') \right) + \varphi$

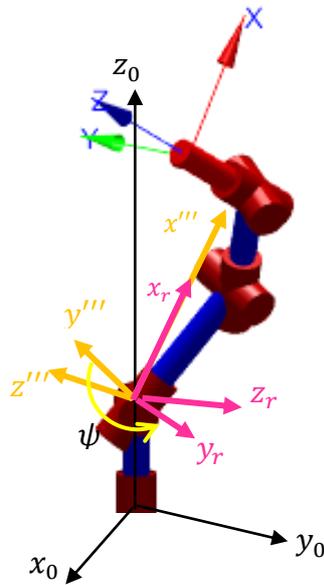


Figura 2.14 Rotazione del sistema di riferimento $x''-y''-z''$ per ottenere il sistema di riferimento r cercato.

Il vettore che individua la posizione del gomito nel nuovo sistema di riferimento dipenderà solo dall'angolo di Swivel e sarà il seguente:

$${}^r \underline{P}_e = \begin{bmatrix} L_1 \cos(\alpha) \\ L_1 \sin(\alpha) \cos(\varphi) \\ L_1 \sin(\alpha) \sin(\varphi) \end{bmatrix} \quad (2.27)$$

Dove l'angolo α è stato definito in precedenza come l'angolo tra L_{WS} e L_1 e φ è l'angolo di Swivel variabile da 0 a 360°.

La posizione del centro della circonferenza è anch'essa facilmente esprimibile ed è individuata dal vettore:

$${}^r \underline{P}_c = \begin{bmatrix} L_1 \cos(\alpha) \\ 0 \\ 0 \end{bmatrix} \quad (2.28)$$

Tali grandezze appena calcolate nel sistema di riferimento r possono essere espresse anche nel sistema di riferimento 0 applicando le seguenti trasformazioni di coordinate:

$${}^0\hat{p}_e = {}^0\hat{T}_r \quad {}^r\hat{p}_e \quad (2.29)$$

$${}^0\hat{p}_c = {}^0\hat{T}_r \quad {}^r\hat{p}_c \quad (2.30)$$

Nella figura 2.15 è rappresentata la circonferenza di Swivel nel caso in cui si richieda all'End Effector la posizione $[-0.1 \ 0.1 \ 0.9]$ e un'orientazione pari a $[0^\circ \ 0^\circ \ 90^\circ]$ adottando una convenzione RPY di angoli di Eulero.

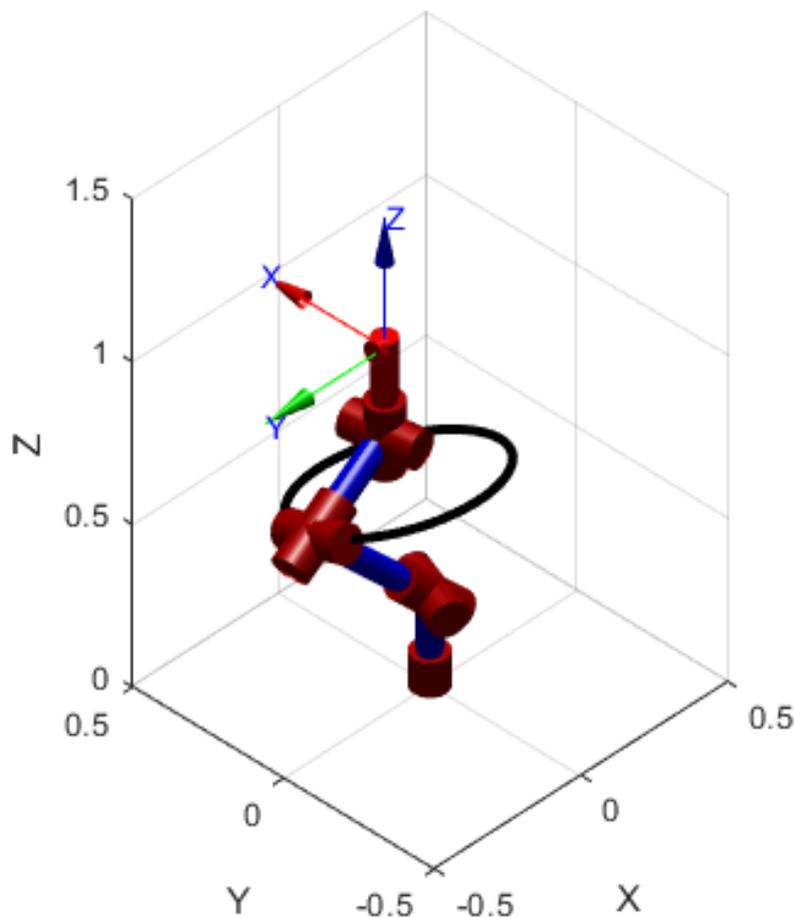


Figura 2.15 Rappresentazione stick figure del robot con circonferenza di Swivel per una data posa dell'End Effector.

Nella figura 2.16 è rappresentata la medesima circonferenza di Swivel ma attraverso una rappresentazione più realistica del manipolatore ottenuta attraverso l'ambiente di simulazione V-rep.

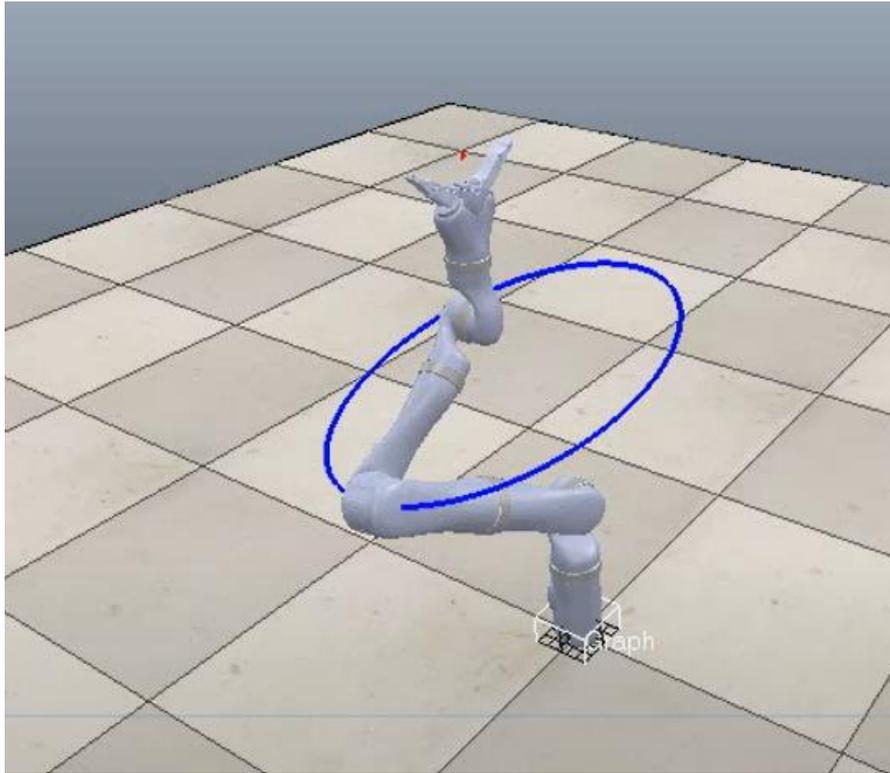


Figura 2.16 Rappresentazione tramite V-rep del robot con circonferenza di Swivel per una data posa dell'End Effector.

2.6. Cinematica inversa

Il problema si affronta dividendolo in due sezioni:

- soluzione di braccio (primi 4 giunti)
- soluzione di polso (ultimi 3 giunti)

Come prima cosa, data la posizione dell'End Effector contenuta nell'ultima colonna della matrice ${}^0\hat{A}_7$, è possibile ottenere la posizione del centro polso come:

$${}^0\underline{p}_w = {}^0\underline{p}_{EE} - {}^0\underline{k}_7 d_{67} \quad (2.31)$$

Dove ${}^0\underline{k}_7$ è il versore dell'asse z del sistema di riferimento 7, ovvero la terza colonna della matrice ${}^0\hat{A}_7$.

Avendo definito, tramite l'angolo di Swivel, la posizione del gomito ${}^0\underline{p}_e$ e, conoscendo l'espressione della matrice di orientazione ${}^0\hat{A}_3$, pari a :

$$\begin{aligned} {}^0\hat{A}_3 = {}^0\hat{A}_1 {}^1\hat{A}_2 {}^2\hat{A}_3 &= \begin{bmatrix} s_1 s_3 + c_1 c_2 c_3 & c_1 s_2 & s_1 c_3 - c_1 c_2 s_3 & d_{23} c_1 s_2 \\ c_1 s_3 - s_1 c_2 c_3 & -s_1 s_2 & c_1 c_3 + s_1 c_2 s_3 & -d_{23} s_1 s_2 \\ s_2 c_3 & -c_2 & -s_2 s_3 & d_1 - c_2 d_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \end{aligned} \quad (2.32)$$

Poiché l'elemento A_{34} rappresenta la componente lungo z del vettore ${}^0\underline{p}_e$, dalla seguente relazione:

$${}^0\underline{p}_e(z) = d_1 - c_2 d_{23} \quad (2.33)$$

Si calcola:

$$\cos(q_2) = \frac{d_1 - {}^0\underline{p}_e(z)}{d_{23}} \quad (2.34)$$

Da cui si ottiene il valore del grado di libertà q_2 pari a:

$$q_{2,1} = \text{atan2}(\sqrt{1 - \cos^2(q_2)}, \cos(q_2)) \quad (2.35)$$

$$q_{2,2} = \text{atan2}(-\sqrt{1 - \cos^2(q_2)}, \cos(q_2)) \quad (2.36)$$

Conoscendo, adesso il valore di q_2 , dai termini A_{14} e A_{24} è possibile estrarre i valori rispettivamente del coseno e del seno di q_1 .

$$q_{1,1} = \text{atan2}\left(-\frac{{}^0\underline{p}_e(y)}{\sin(q_{2,1}) d_{23}}, \frac{{}^0\underline{p}_e(x)}{\sin(q_{2,1}) d_{23}}\right) \quad (2.37)$$

$$q_{1,2} = \text{atan2}\left(-\frac{{}^0\underline{p}_e(y)}{\sin(q_{2,2}) d_{23}}, \frac{{}^0\underline{p}_e(x)}{\sin(q_{2,2}) d_{23}}\right) \quad (2.38)$$

Mentre il valore di q_4 si ottiene a partire dal valore del coseno di δ_4 calcolato in precedenza:

$$q_{4,1} = \text{atan2}(\sqrt{1 - \cos^2(\delta_4)}, \cos(\delta_4)) \quad (2.39)$$

$$q_{4,2} = -\text{atan2}(\sqrt{1 - \cos^2(\delta_4)}, \cos(\delta_4)) \quad (2.40)$$

Per completare la soluzione di braccio rimane da calcolare il valore del giunto q_3 . Esso, come si vede dalla schematizzazione in figura, deve essere tale per cui la direzione del versore ${}^0\underline{k}_4$ coincida con quella della congiungente tra il punto P_e' e il punto P_w .

L'espressione di P_e' calcolata nel sistema di riferimento 0 è la seguente:

$${}^0\underline{p}_{e'} = {}^0\underline{p}_e - e_2 {}^0\underline{k}_3 \quad (2.41)$$

Dove ${}^0\underline{k}_3$ si può estrarre dalla 3 colonna della matrice ${}^0\hat{A}_3$ vista in precedenza.

Come detto in precedenza, il vettore ${}^0\underline{k}_4$ deve coincidere con la congiungente tra $P_{e'}$ e P_w e può, quindi, essere espresso come:

$${}^0\underline{k}_4 = \frac{{}^0\underline{p}_{e'} - {}^0\underline{p}_w}{\|{}^0\underline{p}_{e'} - {}^0\underline{p}_w\|} = \frac{{}^0\underline{p}_{e'} - {}^0\underline{p}_w}{d_{45}} \quad (2.42)$$

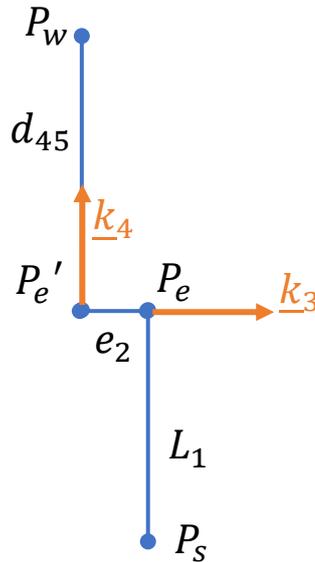


Figura 2.17 Schematizzazione del braccio del robot con vettori dei sistemi di riferimento del link 3 e del link 4.

L'informazione relativa al vettore ${}^0\underline{k}_4$ è contenuta anche nella seguente matrice:

$${}^0\hat{A}_4 = {}^0\hat{A}_3 {}^3\hat{A}_4 = \begin{bmatrix} c_4(s_1s_3 + c_1c_2c_3) + c_1s_2s_4 & c_1c_2s_3 - s_1c_3 & c_1s_2c_4 - s_4(s_1c_3 - c_1c_2s_3) & d_{23}c_1s_2 - e_2(s_1c_3 - c_1c_2s_3) \\ c_4(c_1s_3 - s_1c_2c_3) - s_1s_2s_4 & -c_1c_3 - s_1c_2s_3 & -s_1s_2c_4 - s_4(c_1c_3 + s_1c_2s_3) & -d_{23}s_1s_2 - e_2(c_1c_3 + s_1c_2s_3) \\ c_3s_2c_4 - c_2s_4 & s_2s_3 & -c_1c_4 - c_3s_2s_4 & d_1 - c_2d_{23} + e_2s_2s_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \quad (2.43)$$

Uguagliando, quindi, i due termini si ottiene il seguente sistema di equazioni:

$$\begin{cases} c_1 s_2 c_4 - s_4 (s_1 c_3 - c_1 c_2 s_3) = \frac{{}^0 p_e(x) - e_2 (s_1 c_3 - c_1 c_2 s_3) - {}^0 p_w(x)}{d_{45}} \\ -s_1 s_2 c_4 - s_4 (c_1 c_3 + s_1 c_2 s_3) = \frac{{}^0 p_e(y) - e_2 (c_1 c_3 + s_1 c_2 s_3) - {}^0 p_w(y)}{d_{45}} \\ -c_1 c_4 - c_3 s_2 s_4 = \frac{{}^0 p_e(z) - e_2 s_2 s_3 - {}^0 p_w(z)}{d_{45}} \end{cases} \quad (2.44)$$

Le uniche incognite del sistema sono il seno e il coseno del grado di libertà q_3 . Risolvendo, quindi, il sistema in tali variabili è possibile determinare i 4 valori di q_3 possibili. Si ottengono 4 valori combinando i due valori di q_2 con i due valori di q_4 ottenuti in precedenza.

$$\begin{aligned} \cos(q_3) = & \left(e_2 \underline{p}_e(y) \sin(q_2) - e_2 \underline{p}_w(y) \sin(q_2) \right. \\ & - d_{45} \underline{p}_e(z) \cos(q_1) \sin(q_4) + d_{45} \underline{p}_w(z) \cos(q_1) \sin(q_4) \\ & + e_2 \underline{p}_e(z) \cos(q_2) \sin(q_1) \\ & - e_2 \underline{p}_w(z) \cos(q_2) \sin(q_1) \\ & - \cos(q_1) \cos(q_2) \cos(q_4) \sin(q_4) d_{45}^2 \\ & \left. + \cos(q_2)^2 \cos(q_4) d_{45} e_2 \sin(q_1) + \cos(q_4) d_{45} e_2 \sin(q_1) \sin(q_2)^2 \right) \\ & / \left(\cos(q_1) \sin(q_2) \cdot (d_{45}^2 \sin(q_4)^2 + e_2^2) \right) \end{aligned} \quad (2.45)$$

$$\begin{aligned} \sin(q_3) = & - \left(e_2 \underline{p}_e(z) \cos(q_1) - e_2 \underline{p}_w(z) \cos(q_1) \right. \\ & + d_{45} \underline{p}_e(y) \sin(q_2) \sin(q_4) - d_{45} \underline{p}_w(y) \sin(q_2) \sin(q_4) \\ & + d_{45}^2 \cos(q_2)^2 \cos(q_4) \sin(q_1) \sin(q_4) \\ & + d_{45}^2 \cos(q_4) \sin(q_1) \sin(q_2)^2 \sin(q_4) \\ & + \cos(q_1) \cos(q_2) \cos(q_4) d_{45} e_2 \\ & \left. + \cos(q_2) d_{45} \underline{p}_e(z) \sin(q_1) \sin(q_4) \right. \\ & \left. - \cos(q_2) d_{45} \underline{p}_w(z) \sin(q_1) \sin(q_4) \right) \\ & / \left(\cos(q_1) \sin(q_2) \cdot (d_{45}^2 \sin(q_4)^2 + e_2^2) \right) \end{aligned} \quad (2.46)$$

$$q_3 = \text{atan2}(\sin(q_3), \cos(q_3)) \quad (2.47)$$

Si sono così ottenute le 4 possibili combinazioni in grado di portare il centro polso nella posizione desiderata:

$$\begin{aligned} & [q_{1,1}, q_{2,1}, q_{3,1}, q_{4,1}]; [q_{1,2}, q_{2,2}, q_{3,2}, q_{4,2}] \\ & [q_{1,1}, q_{2,1}, q_{3,3}, q_{4,2}]; [q_{1,2}, q_{2,2}, q_{3,4}, q_{4,1}] \end{aligned} \quad (2.48)$$

Per completare la soluzione del problema di cinematica inversa rimangono da determinare i valori dei gradi di libertà di polso responsabili della corretta orientazione dell'End Effector.

È possibile calcolare la matrice ${}^4\hat{A}_7$:

$$\begin{aligned} {}^4\hat{A}_7 &= {}^4\hat{A}_5 {}^5\hat{A}_6 {}^6\hat{A}_7 \\ &= \begin{bmatrix} s_5s_7 + c_5c_6c_7 & c_5c_6s_7 - c_7s_5 & c_5s_6 & d_{67}c_5s_6 \\ s_5c_6c_7 - c_5s_7 & s_5c_6s_7 - c_7c_5 & s_5s_6 & d_{67}s_5s_6 \\ -s_6c_7 & -s_6s_7 & c_6 & c_6d_{67} - d_{45} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (2.49)$$

Poiché i valori dei primi 4 giunti sono già stati calcolati, la stessa matrice può essere calcolata nel seguente modo:

$${}^4\hat{A}_7 = {}^0\hat{A}_4^{-1} {}^0\hat{A}_7 = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \quad (2.50)$$

Uguagliando le due matrici è possibile calcolare il valore di q_6 .

$$q_{6,1} = \text{atan2}(\sqrt{1 - A_{33}^2}, A_{33}) \quad (2.51)$$

$$q_{6,2} = \text{atan2}\left(-\sqrt{1 - A_{33}^2}, A_{33}\right) \quad (2.52)$$

Conoscendo il valore di q_6 è possibile calcolare anche quello di q_5 e di q_7 .

$$q_{5,1} = \text{atan2}\left(\frac{A_{23}}{\sin(q_{6,1})}, \frac{A_{13}}{\sin(q_{6,1})}\right) \quad (2.53)$$

$$q_{5,2} = \text{atan2}\left(\frac{A_{23}}{\sin(q_{6,2})}, \frac{A_{13}}{\sin(q_{6,2})}\right) \quad (2.54)$$

$$q_{7,1} = \text{atan2}\left(-\frac{A_{33}}{\sin(q_{6,1})}, -\frac{A_{31}}{\sin(q_{6,1})}\right) \quad (2.55)$$

$$q_{7,2} = \text{atan2}\left(-\frac{A_{32}}{\sin(q_{6,2})}, -\frac{A_{31}}{\sin(q_{6,2})}\right) \quad (2.56)$$

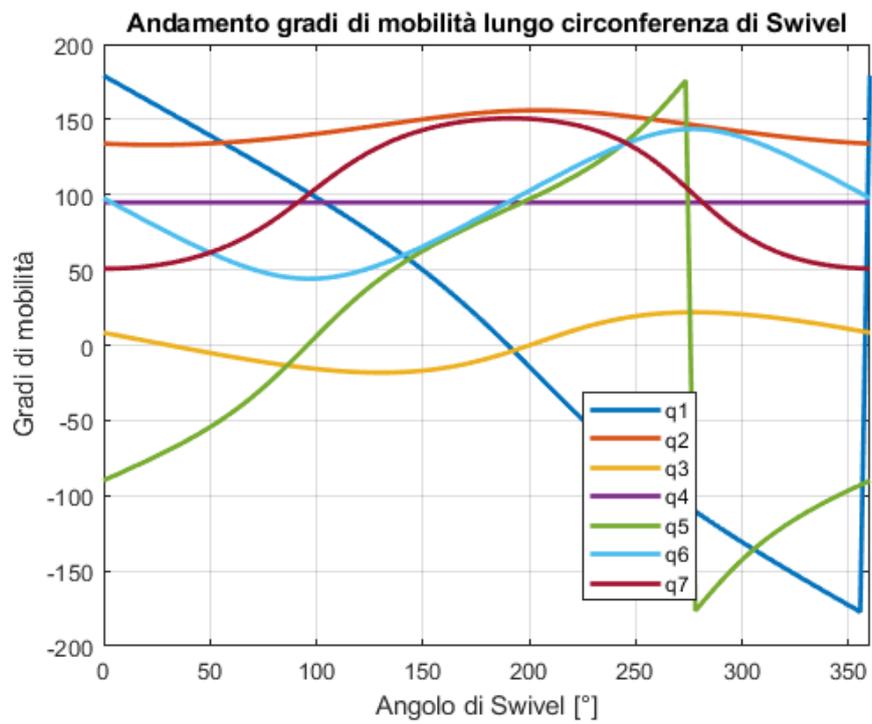
Quindi, assegnata una certa configurazione dei primi quattro giunti, si hanno due possibili terne di soluzioni per i giunti di polso.

$$[q_{5,1}, q_{6,1}, q_{7,1}]; [q_{5,2}, q_{6,2}, q_{7,2}] \quad (2.57)$$

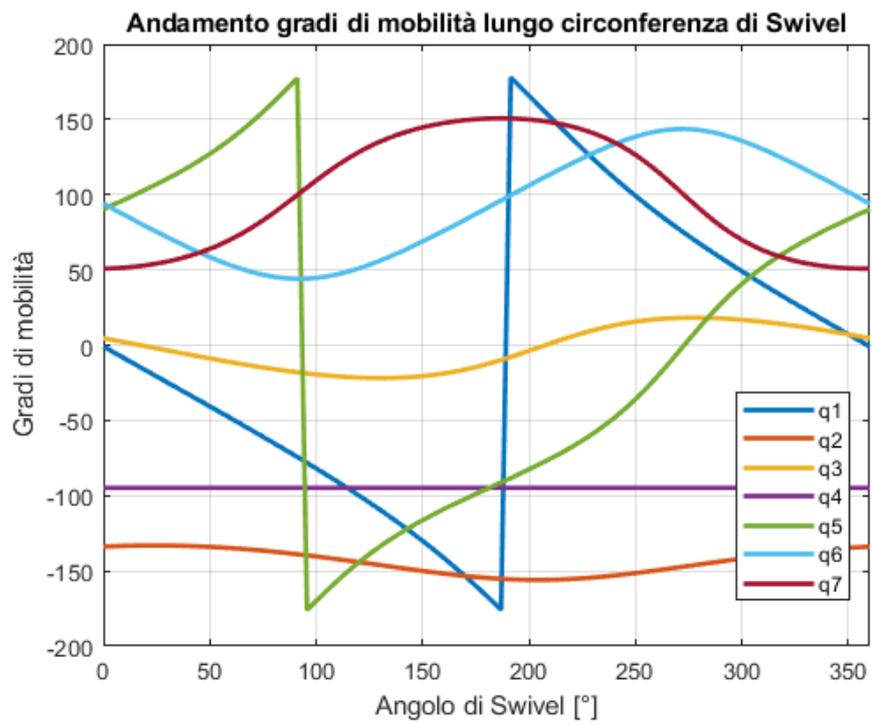
In totale, per ogni posizione del gomito, esistono 8 configurazioni diverse che generano la medesima posa dell'End Effector.

Chiedendo, per esempio, al robot di portare l'End Effector in posizione $[-0.1 \ -0.3 \ 0.8]$ e con un'orientazione pari a $[90^\circ \ 0 \ 0]$ le 8 configurazioni che si ottengono sono rappresentate in figura 2.18 insieme all'andamento dei gradi di libertà nei giunti al variare dell'angolo di Swivel.

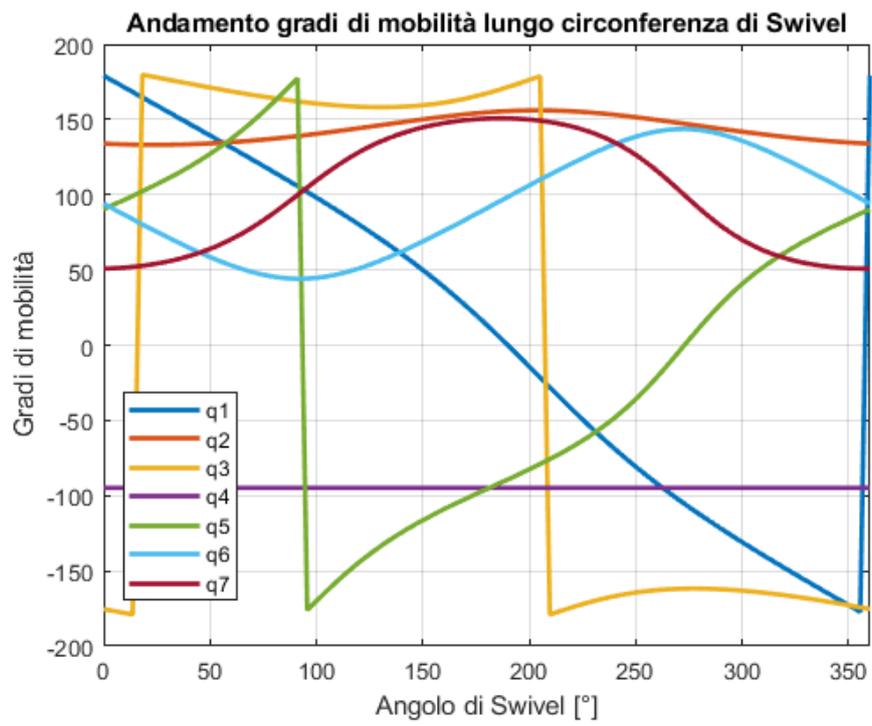
CONFIGURAZIONE 1



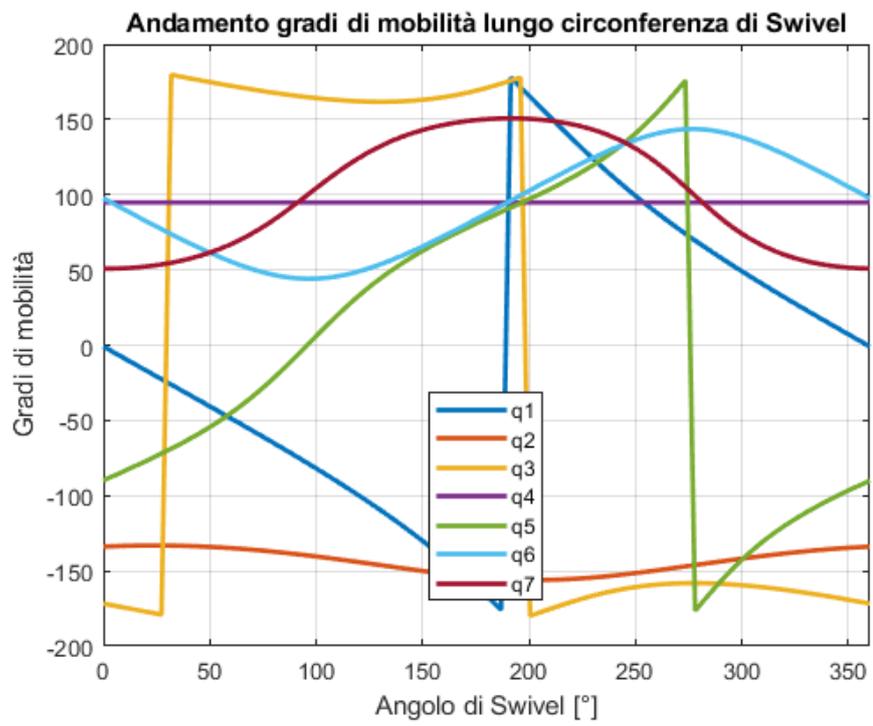
CONFIGURAZIONE 2



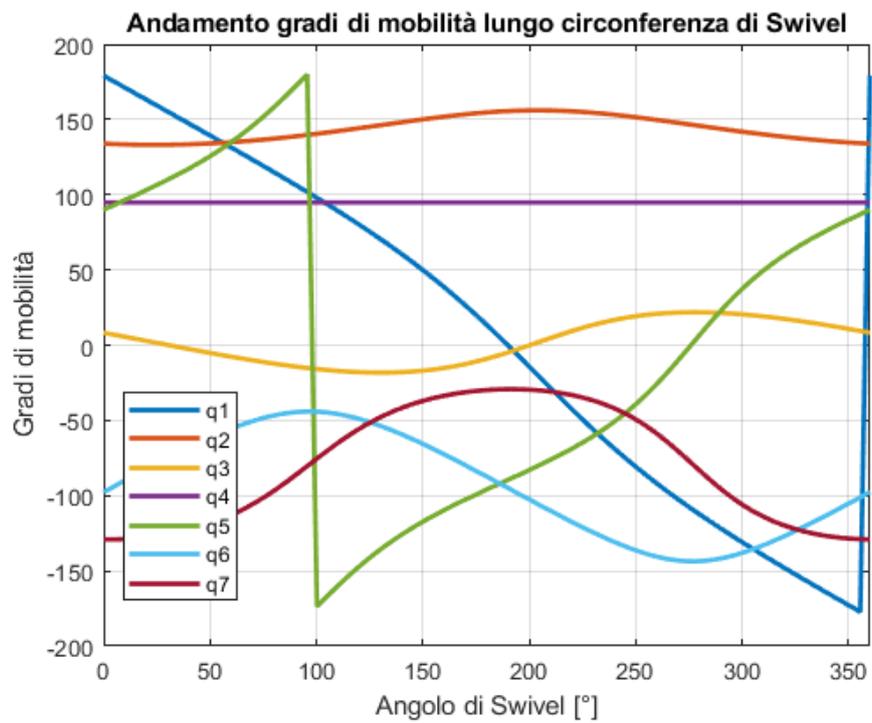
CONFIGURAZIONE 3



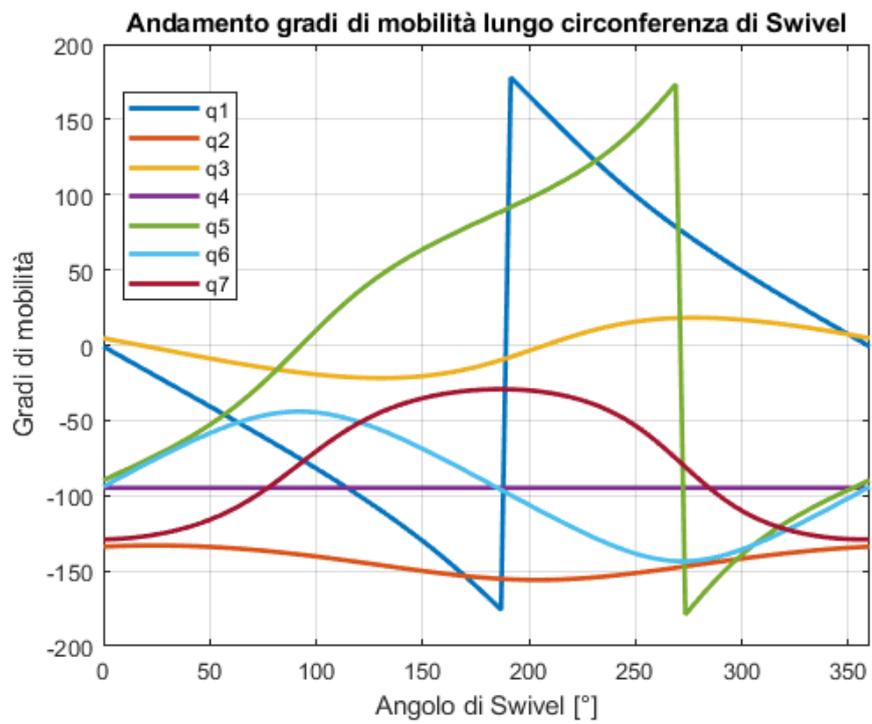
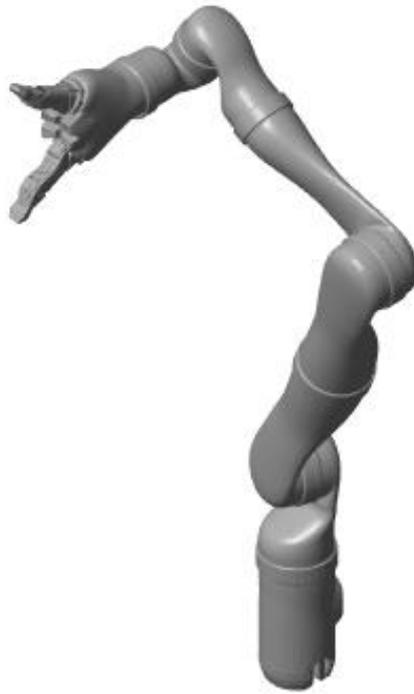
CONFIGURAZIONE 4



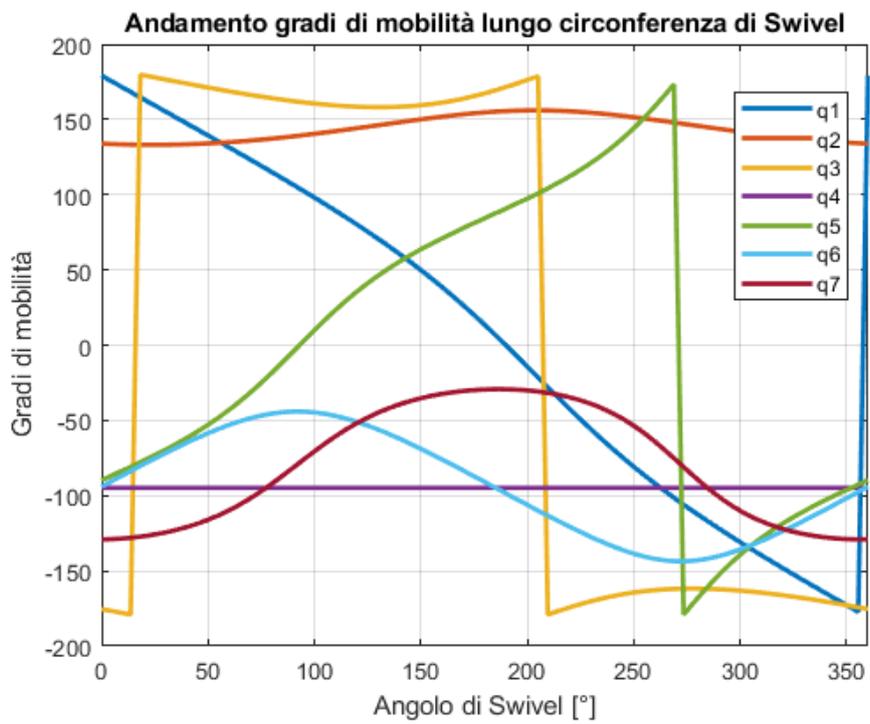
CONFIGURAZIONE 5



CONFIGURAZIONE 6



CONFIGURAZIONE 7



CONFIGURAZIONE 8

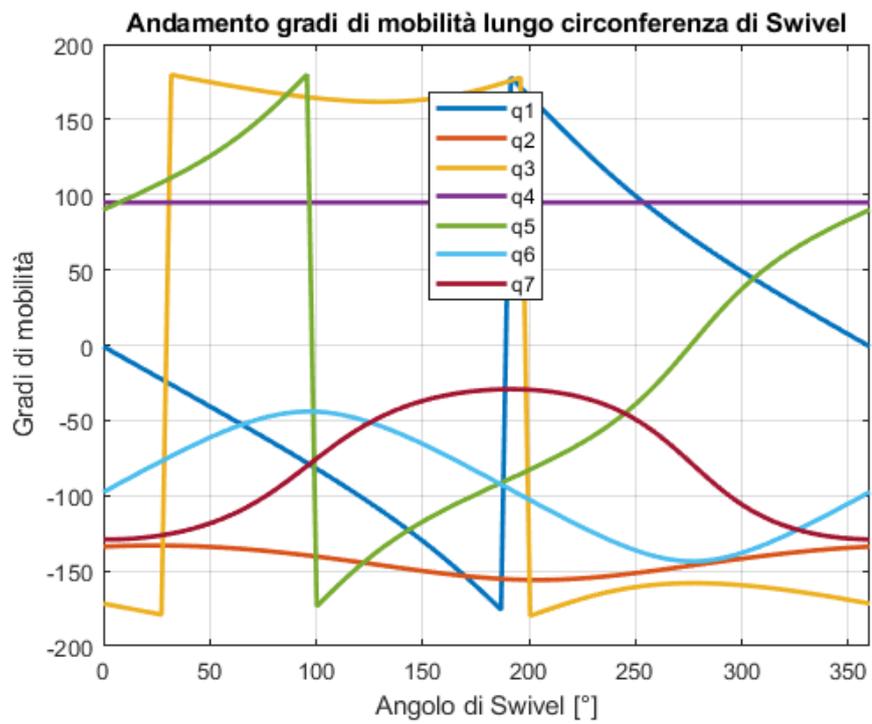
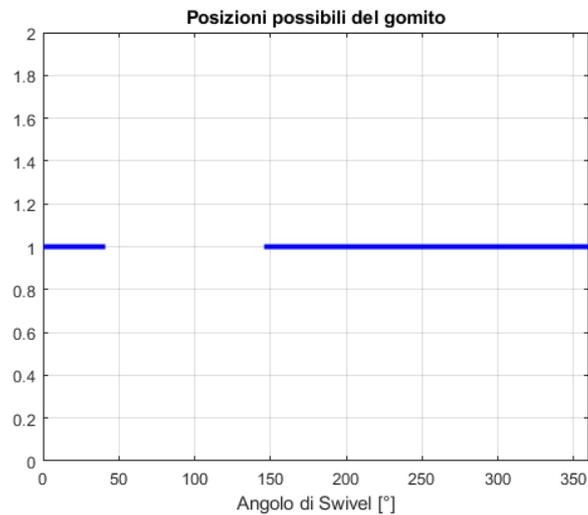


Figura 2.18 Otto configurazioni del robot con stessa posa End Effector e stessa posizione del gomito e valori dei gradi di libertà al variare dell'angolo di Swivel.

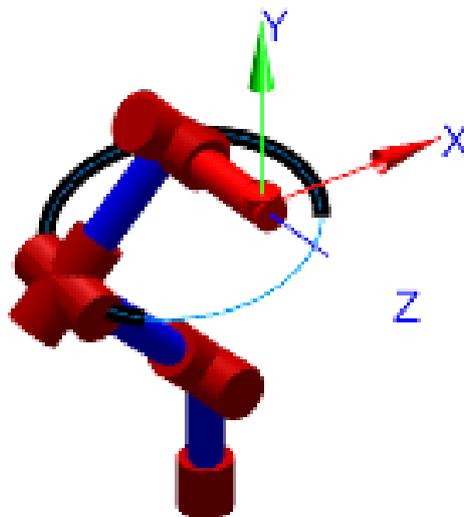
2.7. Limiti di giunto

Poiché esistono dei limiti sui gradi di libertà dei giunti, per una determinata posa dell'End Effector, non è detto che tutte le possibili posizioni del gomito lungo la circonferenza di Swivel siano fisicamente ammissibili. Ciò è dovuto al fatto che uno o più giunti per portare il robot in tale configurazione dovrebbero assumere dei valori che si trovano al di fuori del loro campo di variabilità [2].

In figura 2.19 vengono mostrati i valori di angolo di Swivel fisicamente possibili considerando il caso in cui la posizione dell'End Effector sia $[-0.1 \ -0.3 \ 0.8]$ e la sua orientazione sia pari a $[90^\circ \ 0^\circ \ 0^\circ]$ adottando una convenzione RPY di angoli di Eulero.



(a)



(b)

Figura 2.19 Grafico 2D (a) e grafico 3D (b) con i valori degli angoli di Swivel fisicamente ottenibili per una data posa dell'End Effector.

Nella figura 2.19 sono rappresentate tutte le posizioni possibili del gomito, ovvero tutte quelle posizioni in cui almeno una delle 8 configurazioni è fisicamente realizzabile. Nella figura 2.20, invece, viene riportato un grafico in cui sono rappresentate le possibili posizioni del gomito per ognuna delle 8 soluzioni alla cinematica inversa.

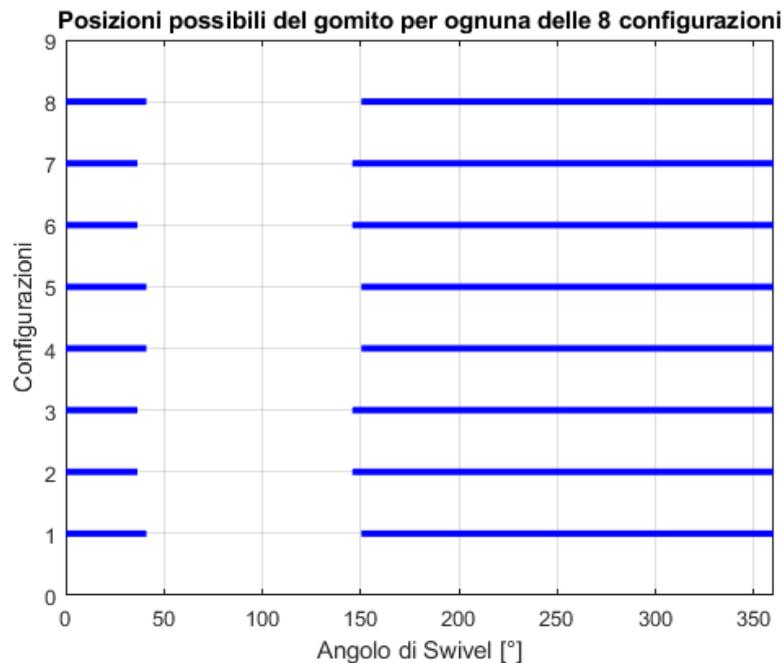


Figura 2.20 Possibili posizioni del gomito per ognuna delle 8 soluzioni alla cinematica inversa.

Si può notare come, le posizioni possibili del gomito lungo la circonferenza di Swivel sono le medesime per tutte e 8 le configurazioni. Tale fenomeno trova giustificazione nel fatto che i due valori dei giunti q_2, q_4, q_6 generati dalla cinematica inversa hanno medesimo modulo ma segno opposto. Poiché i giunti q_1, q_3, q_5, q_7 non hanno limiti alla rotazione e i limiti dei giunti q_2, q_4, q_6 presentano una simmetria rispetto al valore nullo, gli angoli di Swivel per cui il valore dei gradi di libertà supera tali limiti è sempre lo stesso.

Per capire in modo più generale l'influenza dei limiti sul numero di soluzioni fisicamente ottenibili al variare della posa richiesta all'End Effector è stata svolta un'analisi su tutti i punti del workspace del manipolatore.

Per avere la certezza che la posa richiesta si trovasse all'interno del workspace del robot, essa non è stata imposta, ma calcolata tramite un processo di cinematica diretta partendo dai valori dei gradi di libertà nei giunti

Anzitutto si è svolta un'analisi relativa alla sola posizione del centro polso. Tenendo il valore del giunto q_1 e q_3 fisso e andando a variare il valore dei gradi di libertà q_2 e q_4 si è analizzato qual è la distribuzione del numero di soluzioni ottenibile risolvendo la cinematica inversa per ogni posizione. Poiché il manipolatore è una volta ridondante nello spazio il numero di soluzioni per ogni posizione dell'End Effector è infinito, motivo per cui si è fissata anche la posizione del gomito, analizzando quindi, per ogni posizione dell'End Effector solo la soluzione con angolo di Swivel nullo. In questo modo il numero massimo di soluzioni ottenibili per ogni posizione dell'End Effector è 4. Nella figura 2.21 si riporta l'andamento del numero di soluzioni che si ottiene:

Dall'analisi dei grafici si nota come per una data posizione del gomito per ogni posizione dell'End Effector vi siano regioni in cui ci sono o 4 soluzioni oppure 0. Le zone con 0 soluzioni sembrano senza senso in quanto, avendo determinato la posizione dell'End Effector tramite un processo di cinematica diretta a partire da valori di gradi di mobilità accettabili, ci si aspetta che almeno una soluzione in tali regioni sia possibile. In effetti la ragione per cui in certe aree vi sono 0 soluzioni è legata al fatto di aver fissato la posizione del gomito. Infatti, in tali aree esisteranno delle soluzioni alla cinematica inversa ma non per la posizione imposta del gomito.

Per visualizzare tale fenomeno si è proceduto a far variare per ogni posizione la posizione del gomito lungo tutta la circonferenza di Swivel.

Nella figura 2.22, si riporta un grafico in cui viene rappresentata, per ogni posizione dell'End Effector, determinata dalla variazione di q_2 e q_4 , la percentuale di circonferenza di Swivel fisicamente percorribile dal gomito. Si nota come, giustamente, non c'è nessuna regione in cui non vi sono configurazioni possibili.

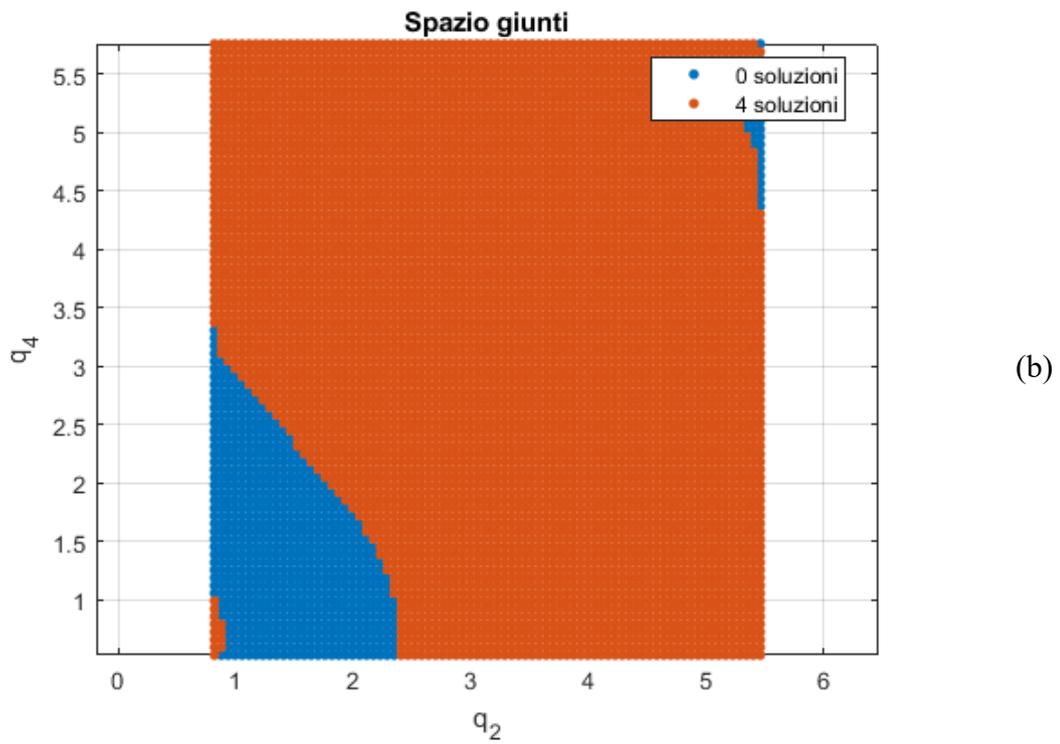
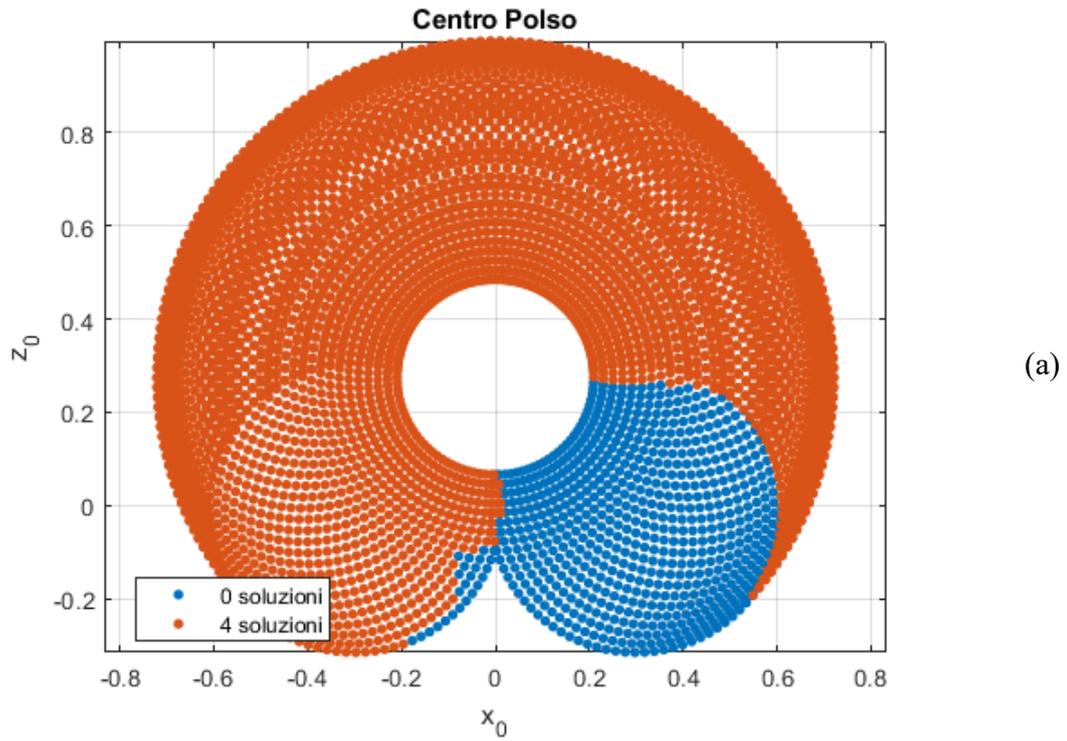


Figura 2.21 Andamento del numero di soluzioni di braccio ottenute dalla cinematica inversa fissando la posizione del gomito lungo la circonferenza di Swivel nello spazio operativo (a) e al variare dei gradi di libertà q_2 e q_4 (b).

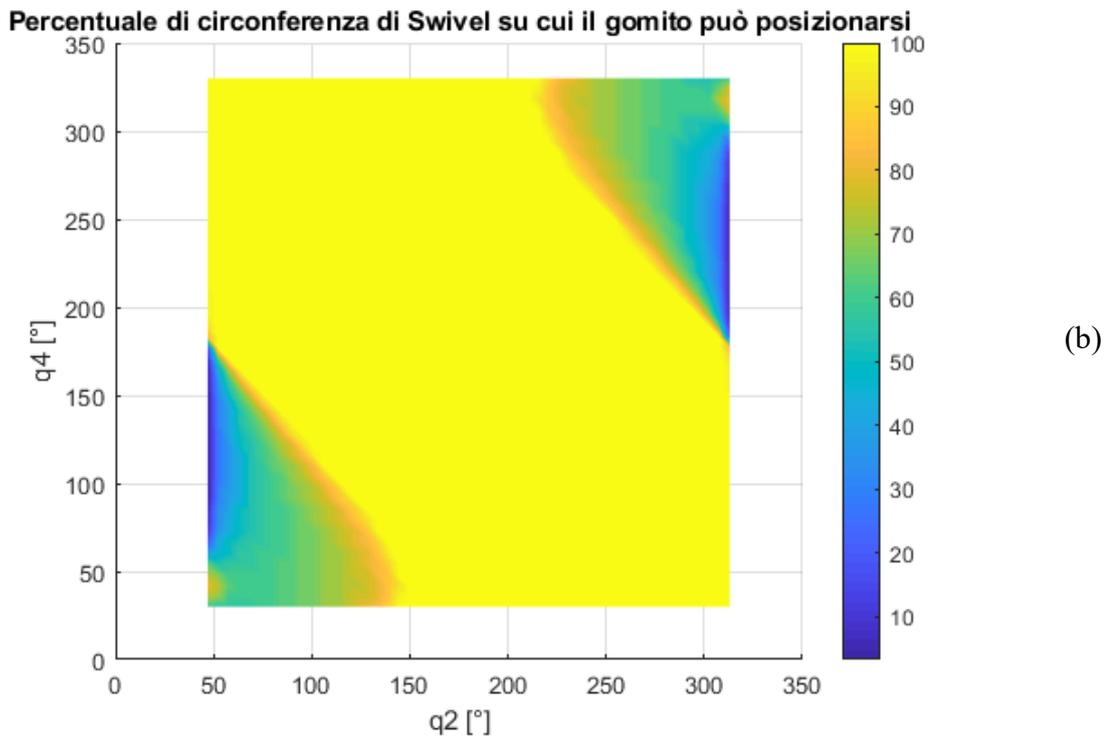
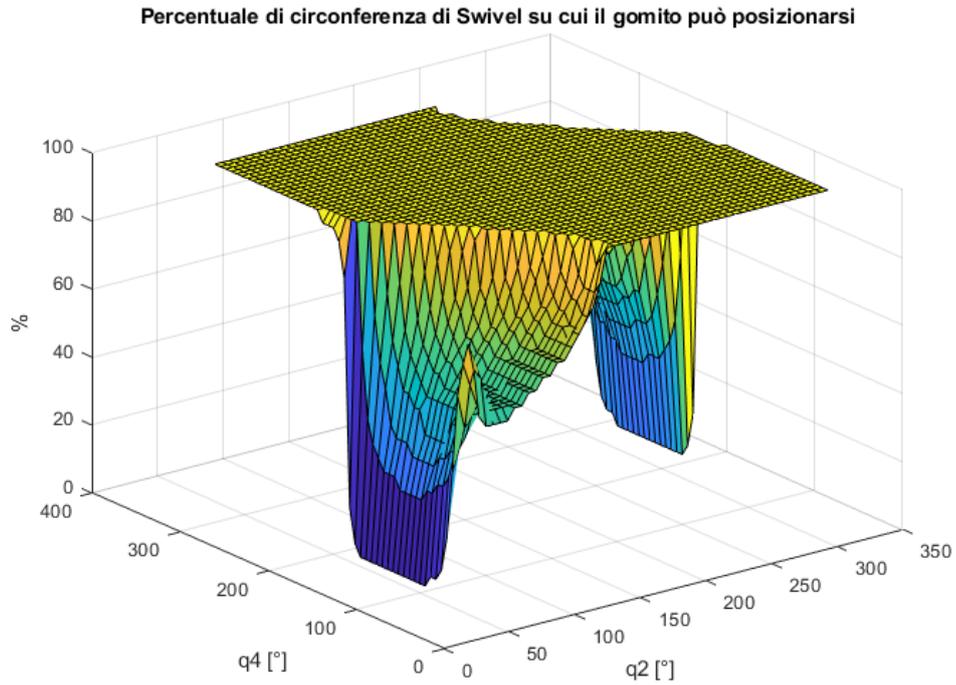
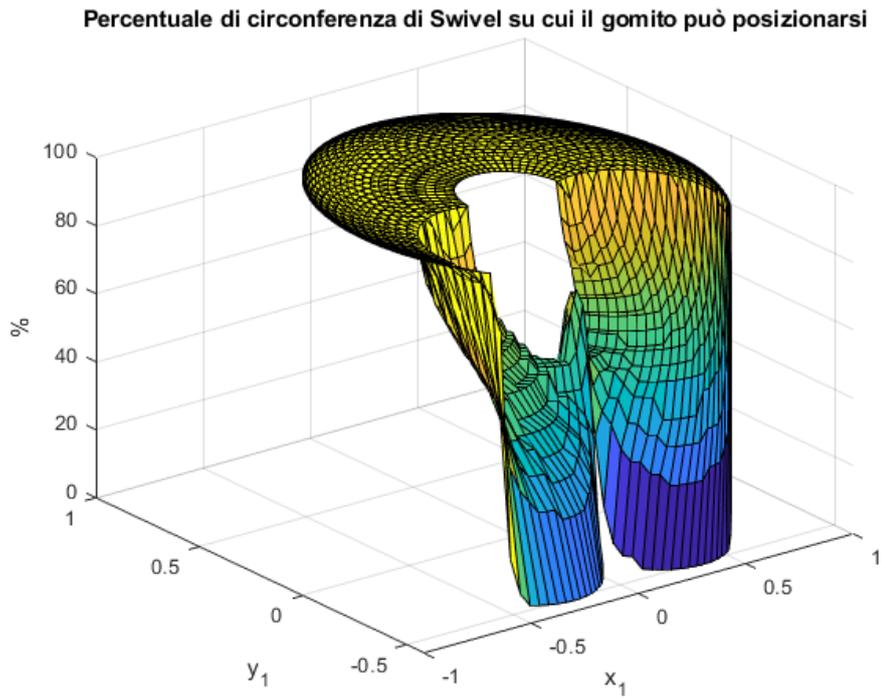
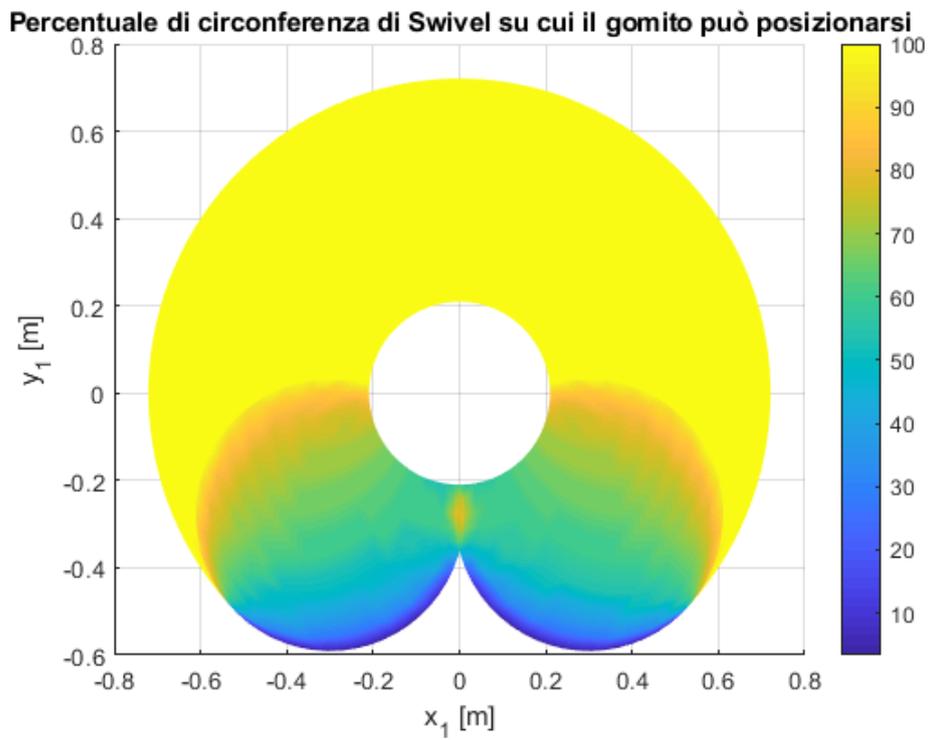


Figura 2.22 Grafico 3D (a) e mappa colorata (b) con percentuale di circonferenza di Swivel su cui il gomito può fisicamente trovarsi al variare dei gradi di libertà q_2 e q_4 .



(a)



(b)

Figura 2.23 Grafico 3D (a) e mappa colorata (b) con percentuale di circonferenza di Swivel su cui il gomito può fisicamente trovarsi al variare della posizione del centro polso.

CAPITOLO 3: Analisi di traiettorie

Una volta studiata la cinematica inversa e compresa la possibilità di ottenere, a partire da una posa dell'End Effector, infinite configurazioni del manipolatore in grado di produrla, è necessario capire in che modo sfruttare tale proprietà. Per fare ciò si è studiato il comportamento del braccio robotico mentre il suo End Effector si muove lungo un percorso rettilineo. La medesima traiettoria viene fatta percorrere al manipolatore usandolo come fosse un robot a 6 gradi di libertà o sfruttando tutti i suoi giunti.

Per ridurre il manipolatore ridondante a un semplice robot a 6 gradi di libertà è stato bloccato un grado di mobilità ad un valore fisso. Il grado di libertà ridondante che si è scelto di bloccare è q_3 , così da ottenere la struttura cinematica di un robot antropomorfo.

All'End Effector del robot viene richiesto un movimento rettilineo tra il punto $[0.65 \ 0.15 \ 0.15]$ e il punto $[0.25 \ 0.15 \ 0.15]$ mantenendo lungo tutto il movimento la stessa orientazione, che utilizzando la convenzione RPY degli angoli di Eulero, è pari a $[0^\circ \ 180^\circ \ 0^\circ]$.

Nella figura 3.1 è rappresentato il percorso dell'End Effector nello spazio operativo.

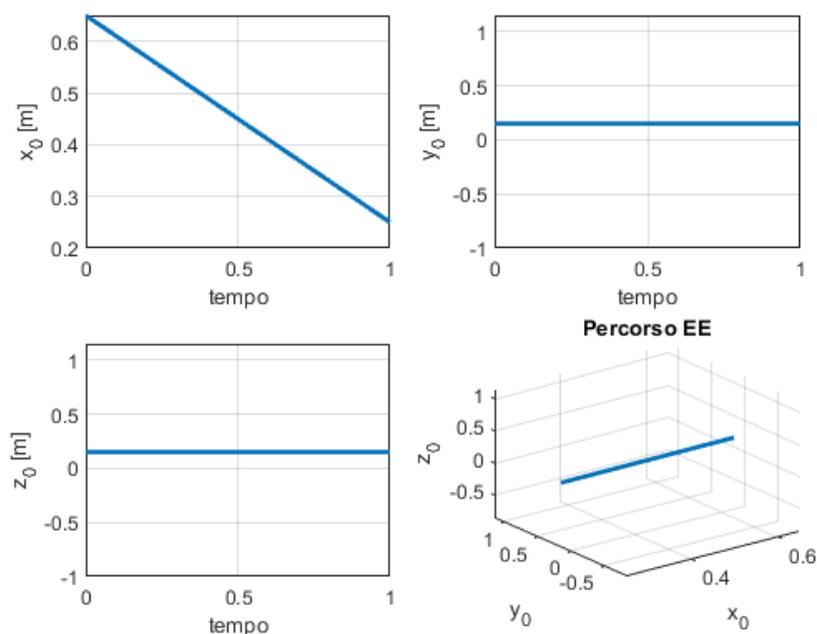


Figura 3.1 Percorso dell'End Effector nello spazio operativo.

Tale traiettoria, come detto, viene fatta percorrere al robot mantenendo inizialmente fisso il grado di libertà q_3 . In questo caso, per ogni posizione richiesta all'End Effector durante il movimento, ci saranno solo 8 soluzioni alla cinematica inversa e non vi sarà quindi la possibilità di controllare la posizione del gomito lungo la circonferenza di Swivel. Infatti, come si può vedere nella figura 3.2, il percorso seguito dall'End Effector è quello richiesto, mentre il percorso del gomito ha un andamento del tutto casuale nello spazio di lavoro.

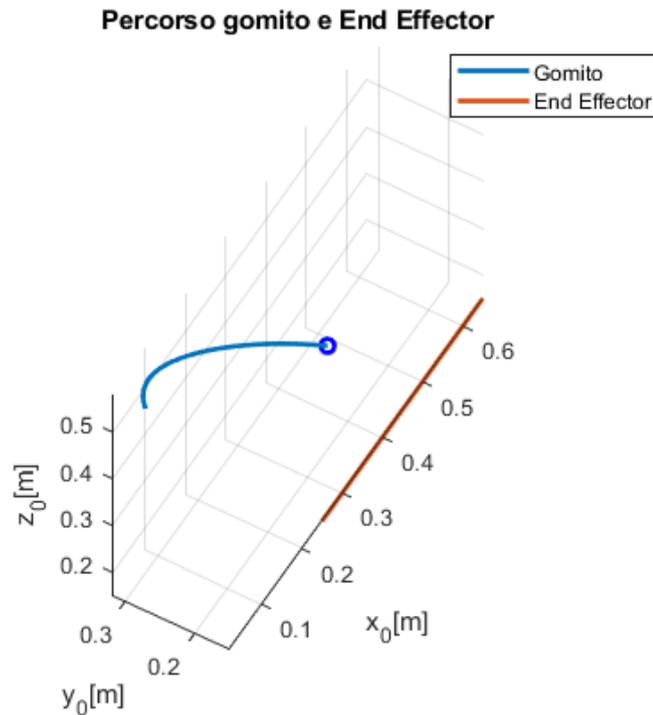


Figura 3.2 Percorso del gomito e dell'End Effector con $q_3 = \text{cost}$.

Il fatto che il gomito segua un percorso qualsiasi all'interno dello spazio di lavoro può rivelarsi un problema nel momento in cui il robot si trova in ambienti confinati o in presenza di ostacoli.

Per esempio, in questo caso, si immagina la presenza di un muro nei pressi del robot che viene simulato da un piano avente equazione:

$$\pi: y = 0.3 \quad (3.1)$$

Dalla figura 3.3 si può vedere come la traiettoria del gomito attraversi il piano, il che significa che il robot entra in collisione con il muro.

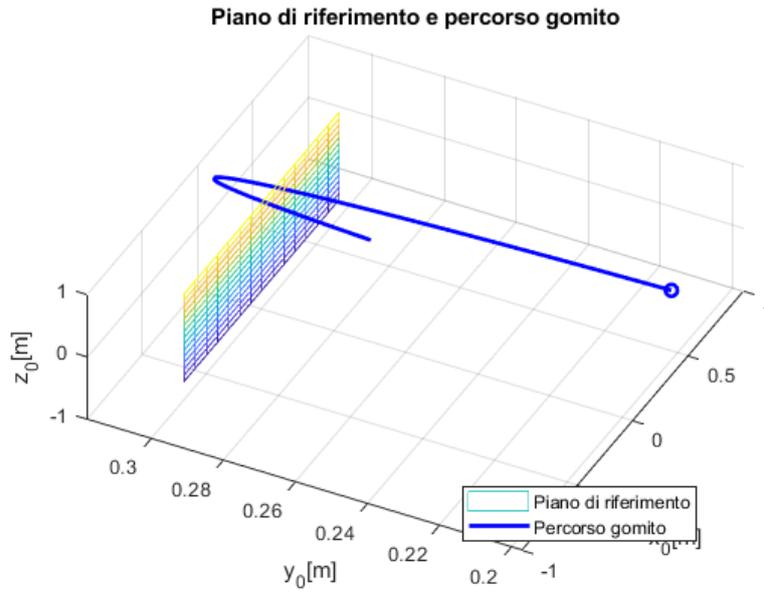


Figura 3.3 Piano di riferimento rappresentante il muro e percorso del gomito.

Nella figura 3.4 è riportato l'andamento dei gradi di libertà dei giunti ottenuti considerando la prima delle 8 possibili soluzioni della cinematica inversa.

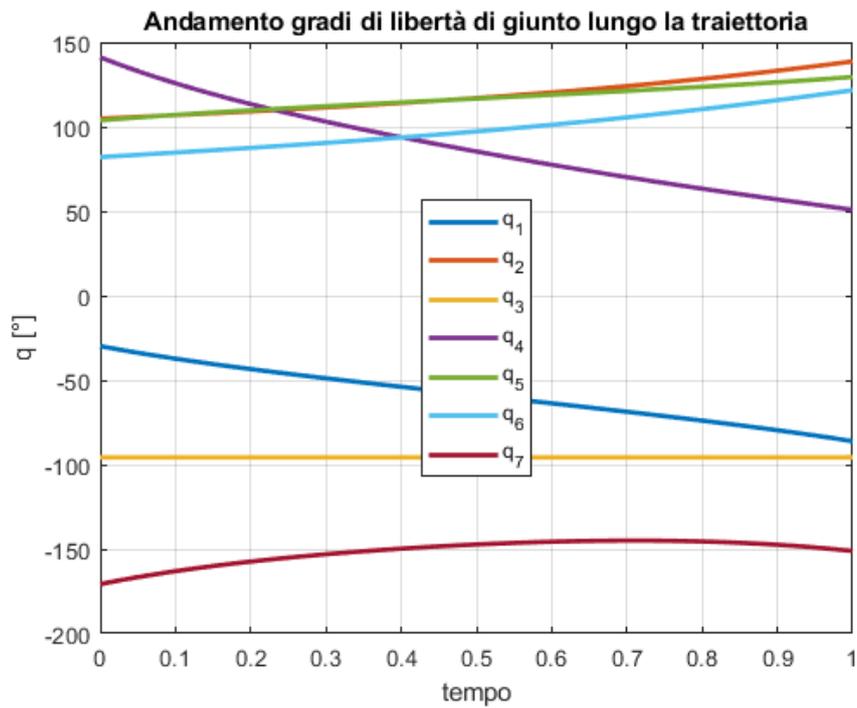


Figura 3.4 Andamento dei gradi di libertà lungo la traiettoria con $q_3 = \text{cost.}$

Nella figura 3.5 si riporta la sequenza dei movimenti del manipolatore simulati nell'ambiente virtuale V-rep riportando in rosso la traiettoria dell'End Effector e in blu quella del gomito.

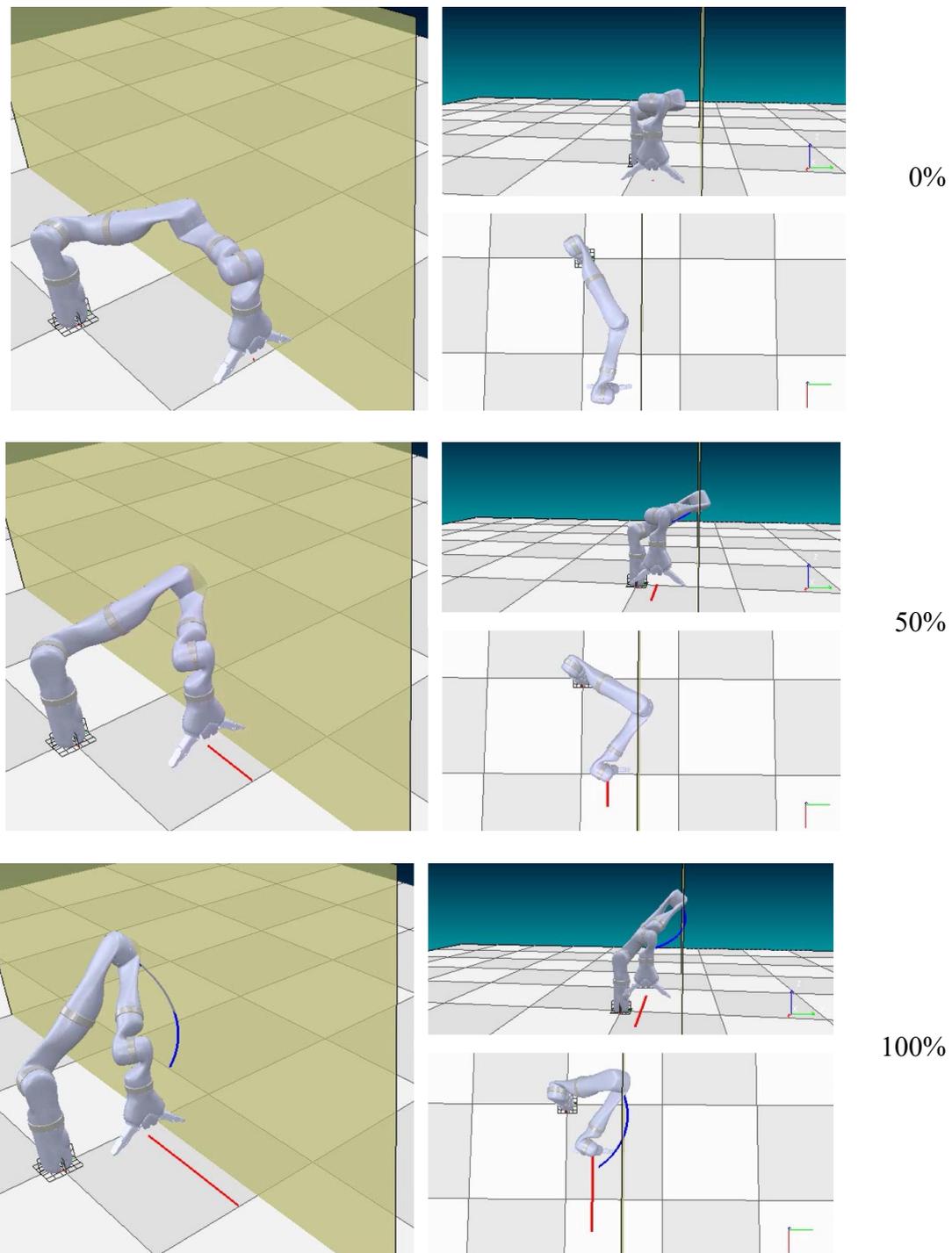


Figura 3.5 Sequenza dei movimenti del robot simulati con V-rep lungo la traiettoria con $q_3 = \text{cost}$ in presenza di un muro.

Nel momento in cui, invece, si vanno ad utilizzare tutti e 7 i gradi di libertà del manipolatore è possibile effettuare la medesima traiettoria scegliendo tra un'infinità di configurazioni diverse e quindi andando a variare la posizione del gomito in modo da evitare il contatto con il muro.

Nel caso in esame, sfruttando la ridondanza dello Jaco 2, è stato chiesto al manipolatore di effettuare la medesima traiettoria dell'End Effector ma mantenendosi a una distanza di sicurezza di 5 cm dal muro. Il primo pezzo della traiettoria è uguale a quello percorso nel caso di braccio a sei gradi di libertà. Però, nel momento in cui il gomito raggiunge la distanza critica viene attivata la ridondanza per fare in modo che tra la posizione del gomito e il muro si mantenga sempre la stessa distanza. Per calcolare le coordinate dei punti di gomito richiesti è stato sufficiente studiare l'intersezione tra l'equazione della circonferenza di Swivel e l'equazione del piano a y costante. Nello specifico per definire l'equazione della circonferenza di Swivel nello spazio è stato necessario introdurre l'equazione del piano su cui essa giace, della sfera avente centro nel centro della circonferenza di Swivel e raggio pari al raggio di Swivel e farne l'intersezione. In definitiva, si ottiene il seguente sistema di tre equazioni in tre incognite che permette di calcolare le coordinate dei punti del gomito.

$$\begin{cases} ap_e(x) + bp_e(y) + cp_e(z) + d = 0 \\ (p_e(x) - p_c(x))^2 + (p_e(y) - p_c(y))^2 + (p_e(z) - p_c(z))^2 = r_c^2 \\ p_e(y) = 0.25 \end{cases} \quad (3.2)$$

Dove i coefficienti a, b, c e d sono stati calcolati imponendo il passaggio del piano per il centro della circonferenza di Swivel e per due punti di gomito ad essa appartenenti tramite l'equazione 2.20.

Tale sistema, presentando un'equazione di secondo grado, individua due soluzioni nello spazio. È necessario, quindi, scegliere quella che non genera una discontinuità con la traiettoria di avvicinamento del robot al muro.

Nella figura 3.6 vengono mostrate la traiettoria dell'End Effector e quella del gomito; la prima risulta essere la medesima del caso precedente.

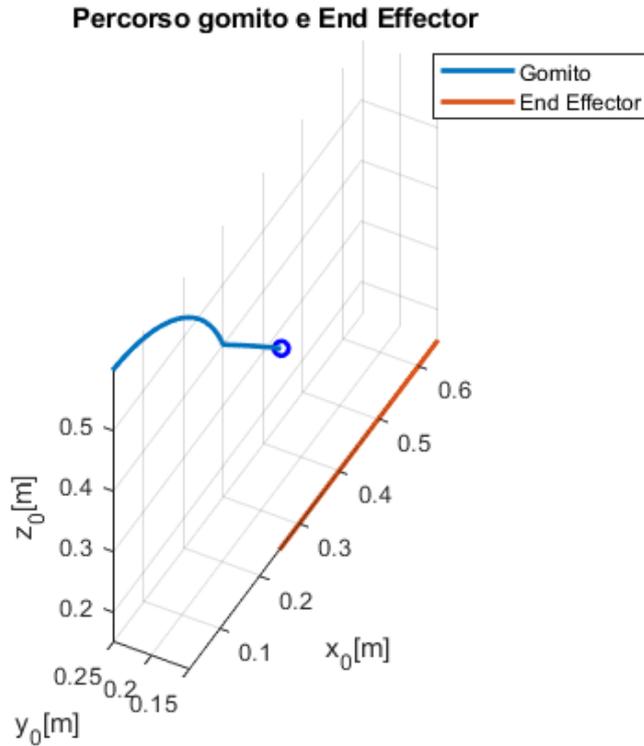


Figura 3.6 Percorso del gomito e dell'End Effector sfruttando la ridondanza.

Se si considera il medesimo ostacolo planare considerato in precedenza, in questo caso, il gomito si troverà al di fuori da una condizione di interferenza; una schematizzazione del piano e del percorso del gomito è rappresentata in figura 3.7.

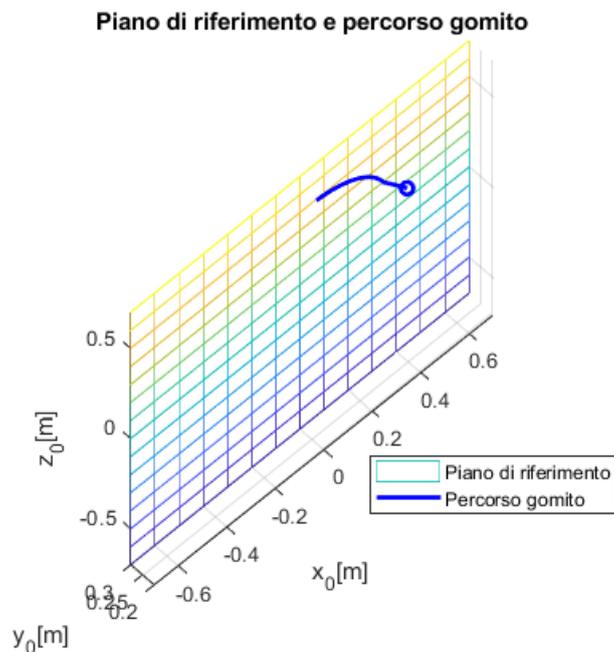


Figura 3.7 Piano di riferimento rappresentante il muro e percorso gomito.

Nella figura 3.8 è riportato l'andamento dei gradi di libertà dei giunti imponendo la giacenza del gomito sul piano.

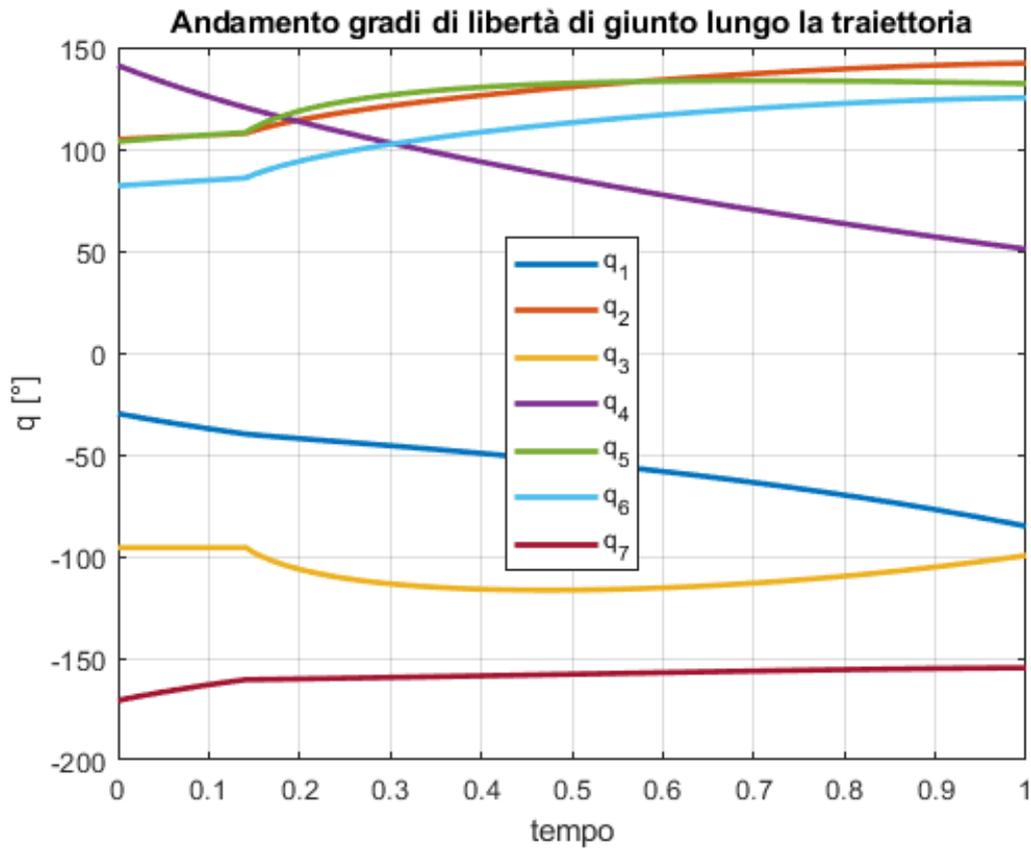


Figura 3.8 Andamento dei gradi di libertà nei giunti sfruttando la ridondanza con la presenza di un muro nello spazio di lavoro del robot.

Nella figura 3.9 si riporta la sequenza dei movimenti del manipolatore simulati nell'ambiente virtuale V-rep riportando in rosso la traiettoria dell'End Effector e in blu quella del gomito.

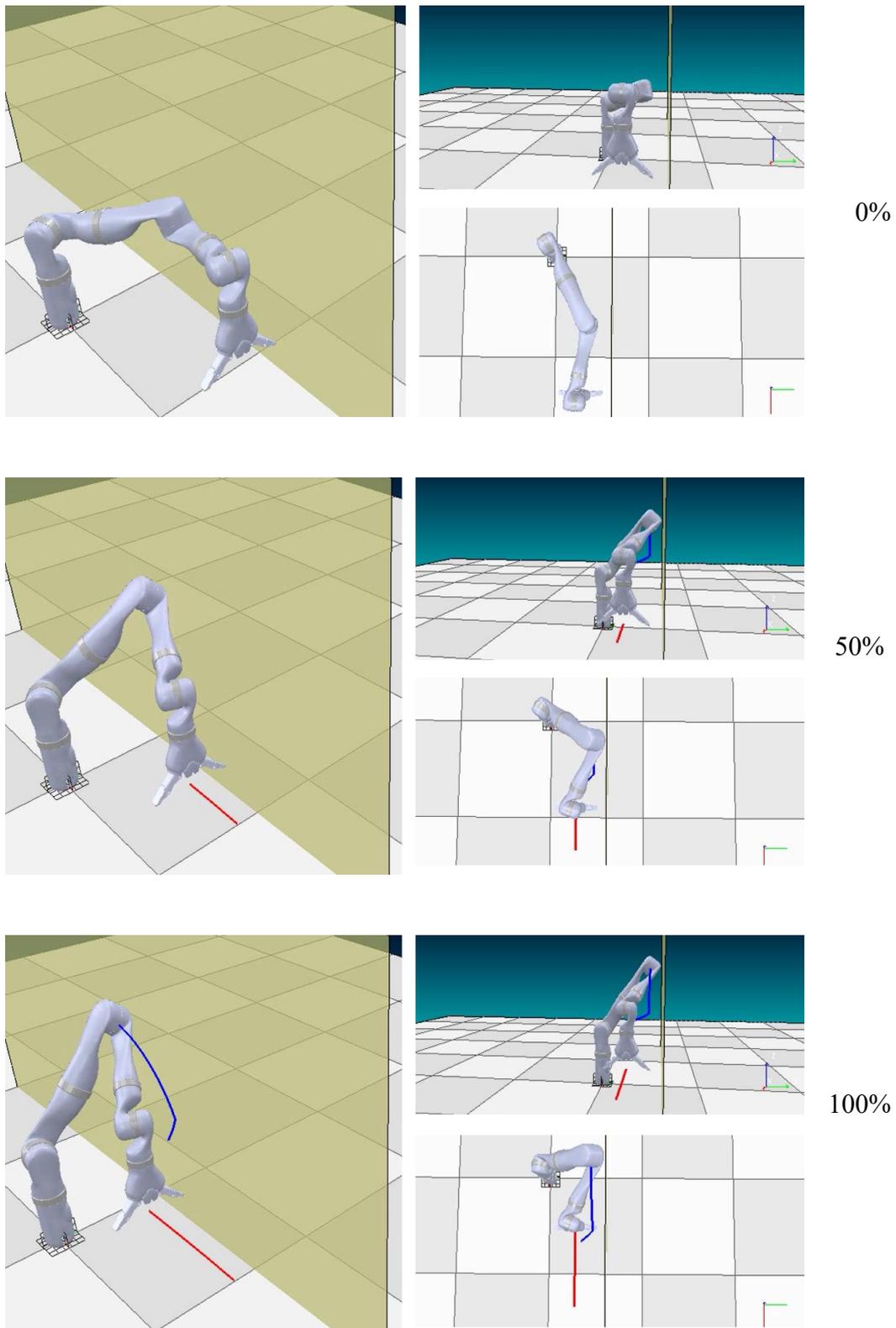


Figura 3.9 Sequenza dei movimenti del robot simulati con V-rep lungo la traiettoria sfruttando la ridondanza in presenza di un muro.

La proprietà di un manipolatore ridondante di poter raggiungere lo stesso punto nello spazio in un numero infinito di configurazioni è, come detto, molto importante per esempio per evitare interferenze tra il robot e oggetti che ne ostacolano il movimento. Per comprendere meglio tale fatto si considera la traiettoria che porta l'End Effector dalla posizione $[0.30 \ 0 \ 0.15]$ alla posizione $[0.65 \ 0 \ 0.15]$ con un'orientazione costante, che utilizzando la convenzione RPY degli angoli di Eulero, è pari a $[0^\circ \ 180^\circ \ 0^\circ]$. Nella figura 3.10 è rappresentato il percorso dell'End Effector.

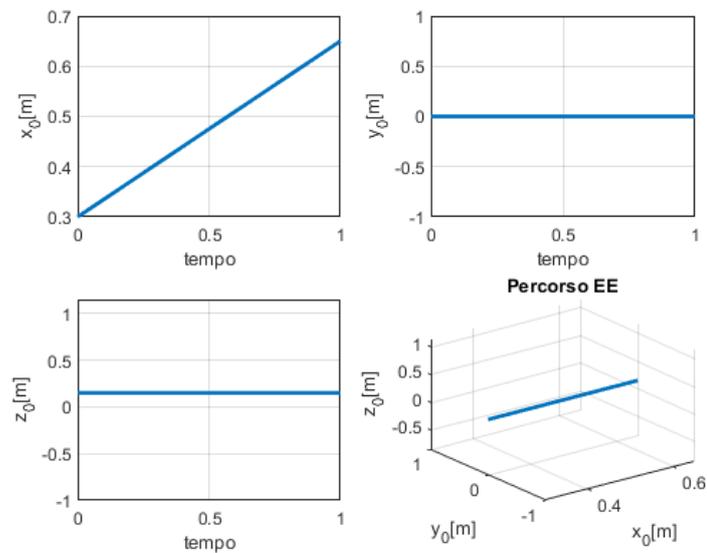


Figura 3.10 Percorso dell'End Effector nello spazio operativo.

Nella figura 3.11 e 3.12 sono rappresentati il percorso dell'End Effector e del gomito che si ottengono bloccando il grado di libertà q_3 del manipolatore.

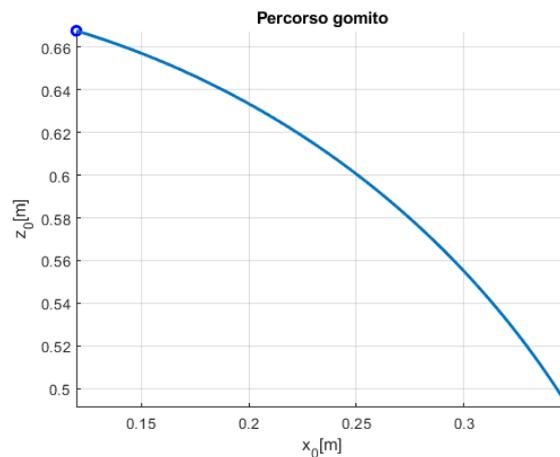


Figura 3.11 Percorso del gomito con $q_3 = \text{cost.}$

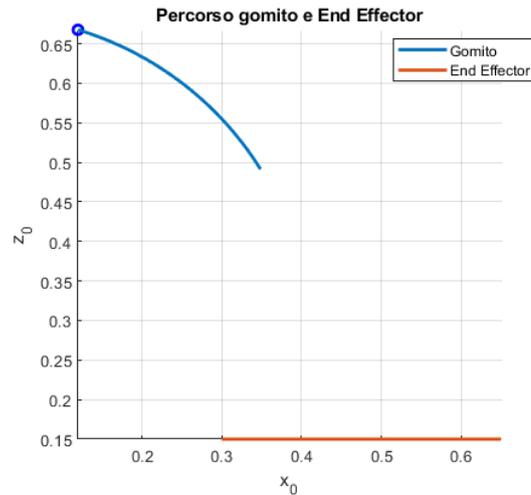


Figura 3.12 Percorso del gomito e dell'End Effector con $q_3 = \text{cost}$.

Se si considera adesso la presenza all'interno del workspace del manipolatore di un ostacolo sferico, di raggio 0.08 m e centro in $[0.295 \ 0 \ 0.68]$, è possibile notare come, durante il movimento, il gomito entri in collisione con tale oggetto. In figura 3.13 sono rappresentati i percorsi di gomito, End Effector e l'ostacolo sferico.

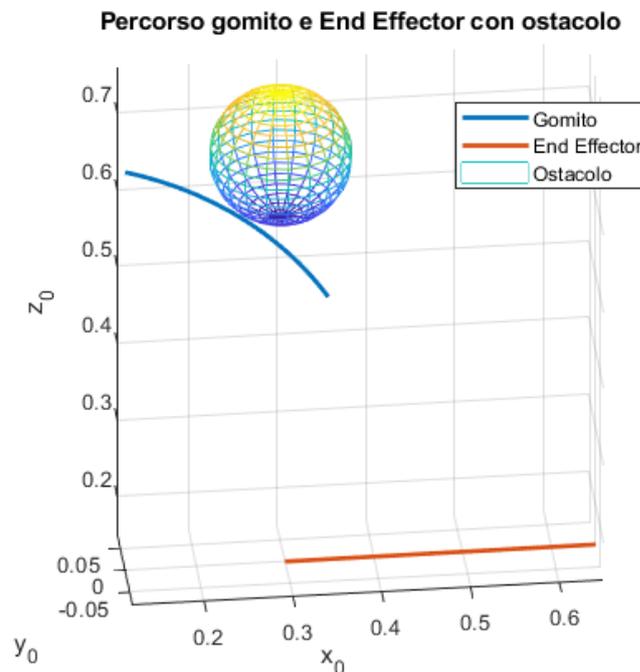
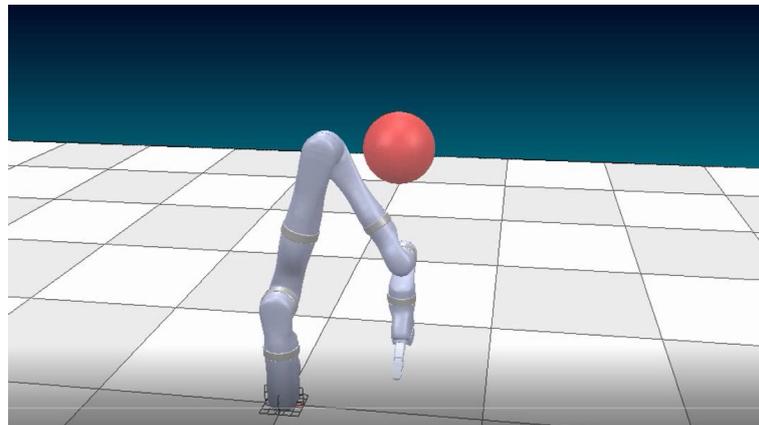


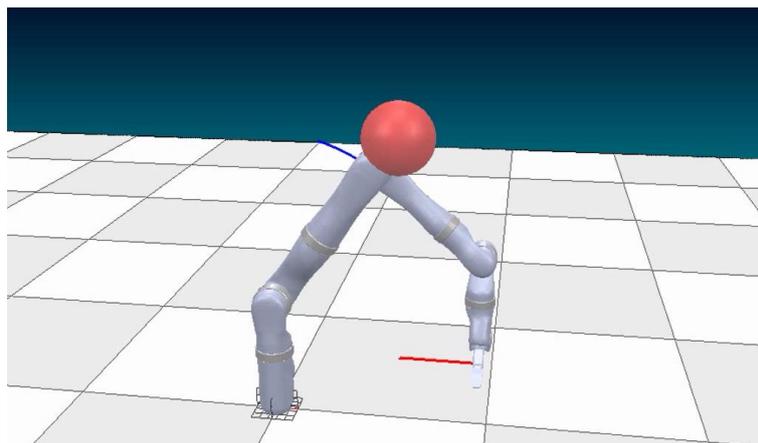
Figura 3.13 Percorso del gomito e dell'End Effector con $q_3 = \text{cost}$ in presenza di un ostacolo sferico.

Dall'analisi della figura 3.13 sembrerebbe che non vi sia collisione tra gomito e sfera ma così non è. Infatti, in tale rappresentazione non si considerano gli ingombri effettivi del

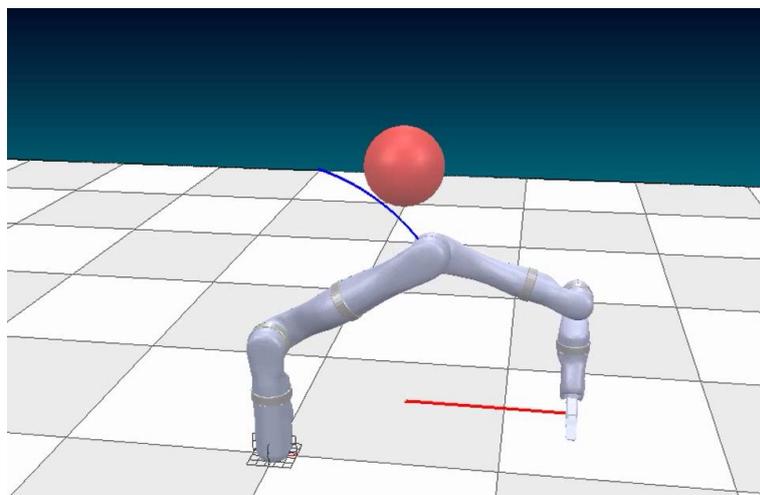
braccio robotico. Per visualizzare meglio tale fenomeno in figura 3.14 si riporta la sequenza dei movimenti del manipolatore simulati nell'ambiente virtuale V-rep individuando in rosso la traiettoria dell'End Effector e in blu quella del gomito.



0%



50%



100%

Figura 3.14 Sequenza dei movimenti del robot simulati con V-rep lungo la traiettoria con $q_3 = \text{cost}$ in presenza di un ostacolo sferico.

Per evitare la collisione è possibile sfruttare la ridondanza del robot. Utilizzando tutti e sette i gradi di libertà dei giunti del manipolatore è possibile portare la traiettoria del gomito fuori dal campo di interferenza della sfera evitando quindi il contatto tra il gomito e l'ostacolo.

Per fare ciò si è definito un angolo di rotazione δ che è necessario imporre al piano passante per centro polso, centro spalla e gomito in modo tale che il robot non interferisca con l'oggetto ostacolante. L'obiettivo è quello di avere la configurazione iniziale e finale del manipolatore uguale a quella del caso precedente, per cui il valore dell'angolo di rotazione δ dovrà valere zero agli estremi della traiettoria. L'andamento imposto al valore dell'angolo di rotazione è rappresentato in figura 3.15.

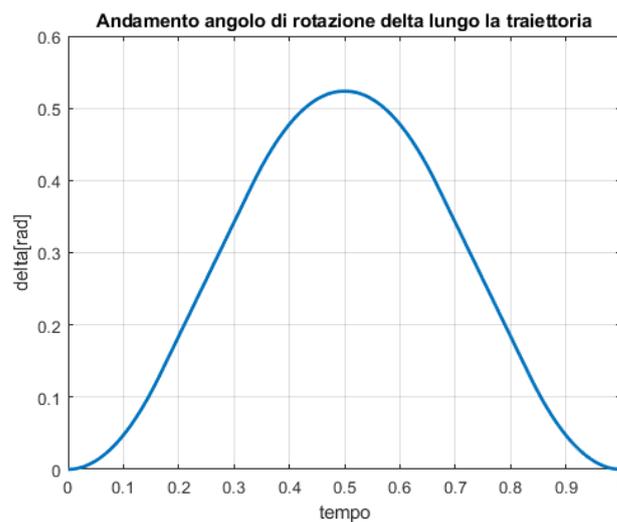


Figura 3.15 Andamento dell'angolo di rotazione δ lungo la traiettoria

Una volta che si conosce, istante per istante, il valore di tale angolo è necessario determinare la posizione del gomito. Dato che, istante per istante, è nota la posizione del centro della circonferenza di Swivel, è possibile conoscere anche il vettore che collega tale punto con il punto individuato dal gomito nel caso in cui $\delta = 0^\circ$. Tale vettore dovrà essere ruotato dell'opportuno angolo δ intorno all'asse passante per centro polso e centro spalla. Poiché tali punti sono noti si ricava la matrice di rotazione:

$$Rot(\underline{h}, \delta) = \begin{bmatrix} h_x^2(1 - \cos\delta) + \cos\delta & h_x h_y(1 - \cos\delta) - h_z \sin\delta & h_x h_z(1 - \cos\delta) + h_y \sin\delta \\ h_x h_y(1 - \cos\delta) + h_z \sin\delta & h_y^2(1 - \cos\delta) + \cos\delta & h_y h_z(1 - \cos\delta) - h_x \sin\delta \\ h_x h_z(1 - \cos\delta) - h_y \sin\delta & h_y h_z(1 - \cos\delta) + h_x \sin\delta & h_z^2(1 - \cos\delta) + \cos\delta \end{bmatrix} \quad (3.3)$$

Con il vettore h che vale:

$$\underline{h} = {}^0\underline{p}_w - {}^0\underline{p}_s \quad (3.4)$$

Moltiplicando la matrice per il vettore individuato precedentemente si ottiene un nuovo vettore con punto di applicazione e modulo invariati (centro e raggio della circonferenza di Swivel) ma con direzione diversa passante per il punto di gomito cercato. In figura 3.16 è riportata una schematizzazione di tale processo.

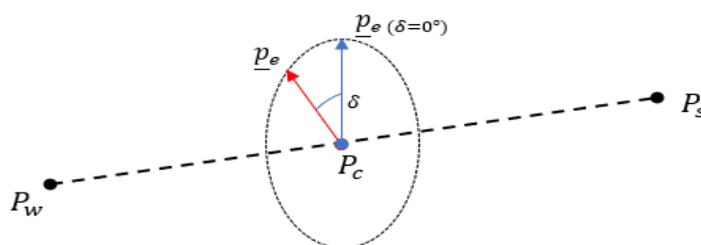


Figura 3.16 Schematizzazione adottata per il calcolo delle posizioni del gomito del manipolatore.

Nella figura 3.16 viene riportato un grafico contenente l'ostacolo sferico, il percorso dell'End Effector e il percorso del gomito. Si può notare come, a differenza del caso precedente, sebbene la configurazione iniziale e finale del braccio siano le stesse, non vi è interferenza tra il braccio robotico e l'oggetto esterno.

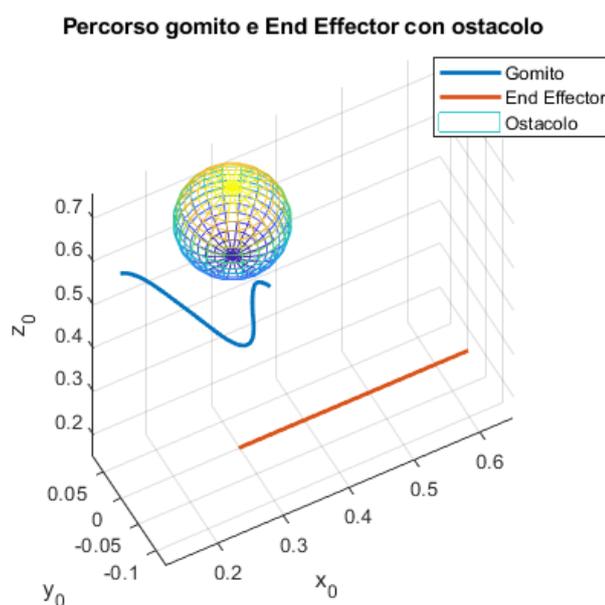


Figura 3.17 Percorso del gomito e dell'End Effector sfruttando la ridondanza in presenza di un ostacolo sferico.

Nella figura 3.17 si confrontano gli andamenti dei gradi di libertà nei giunti nei due casi. Nel primo caso si nota la costanza del valore di q_3 , mentre nel secondo si nota un andamento diverso sebbene i gradi di libertà delle due configurazioni iniziali e finali siano gli stessi.

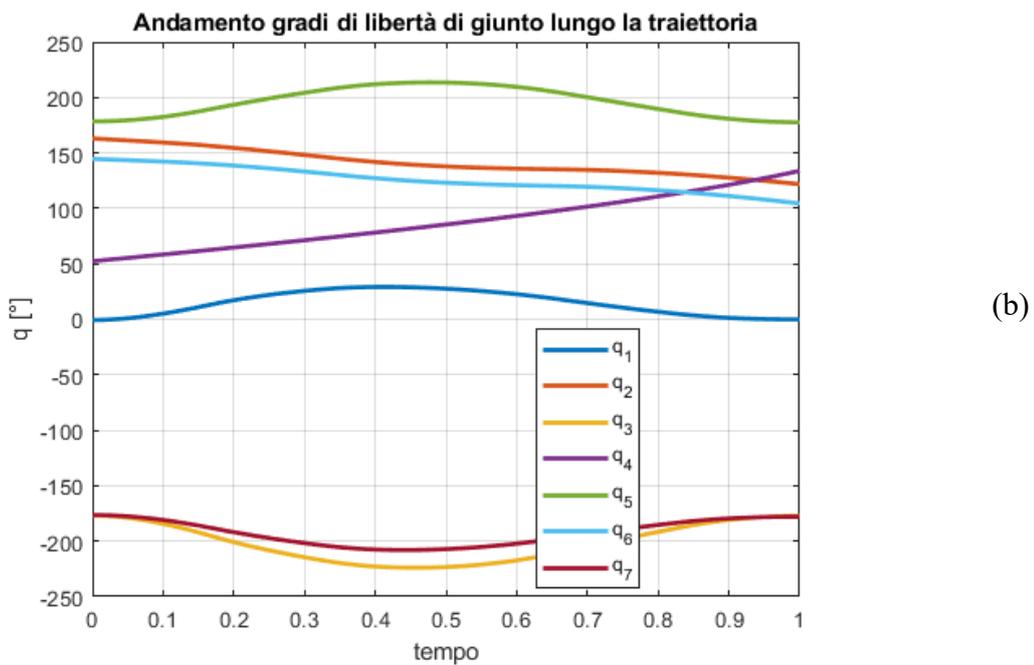
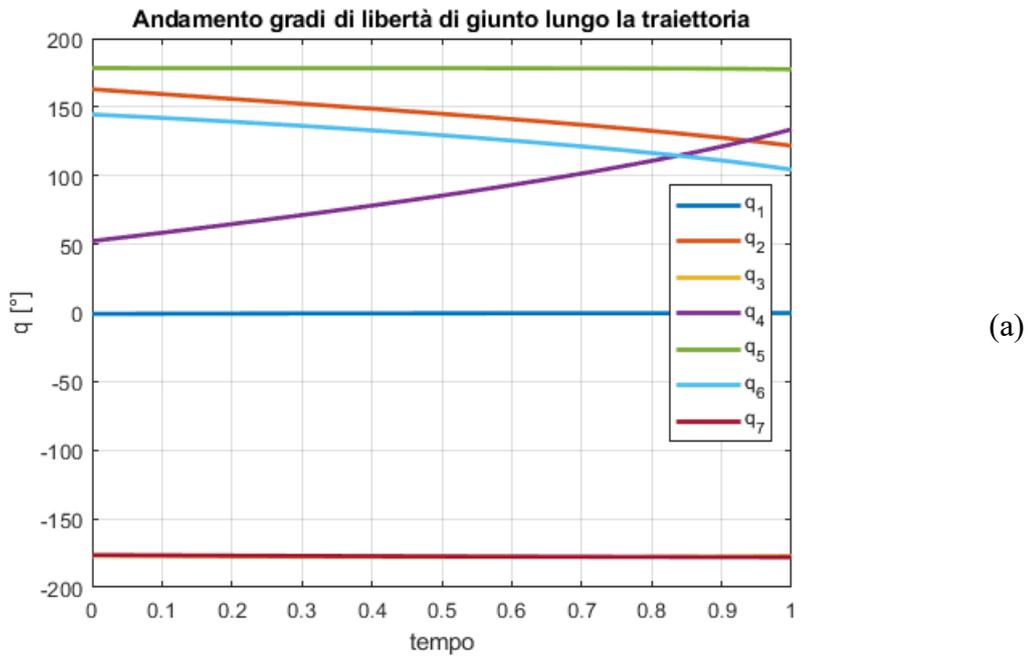
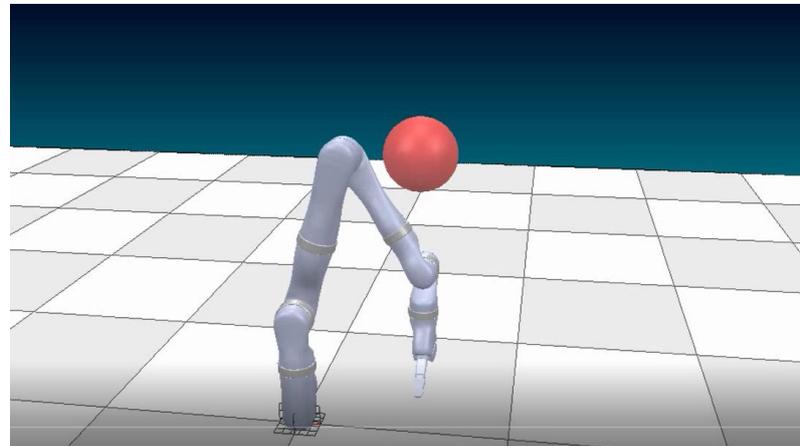
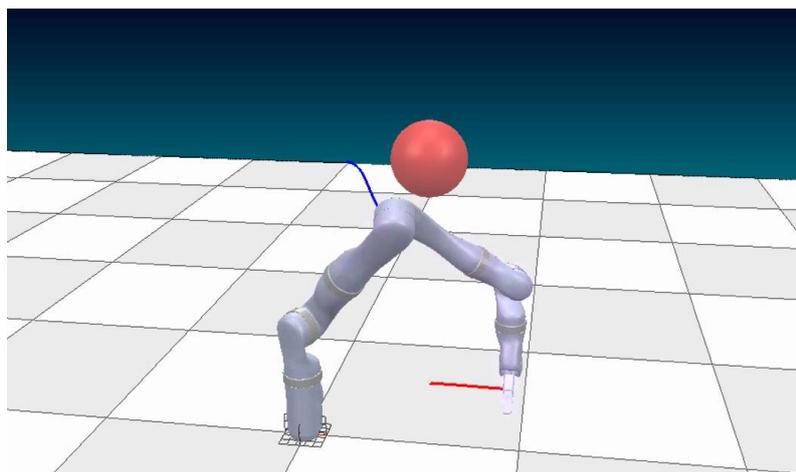


Figura 3.18 Confronto tra gli andamenti dei gradi di libertà nei giunti nel caso in cui si utilizzi il robot come un 6DOF (a) o nel caso in cui si sfrutti la ridondanza per evitare collisione con ostacolo sferico (b).

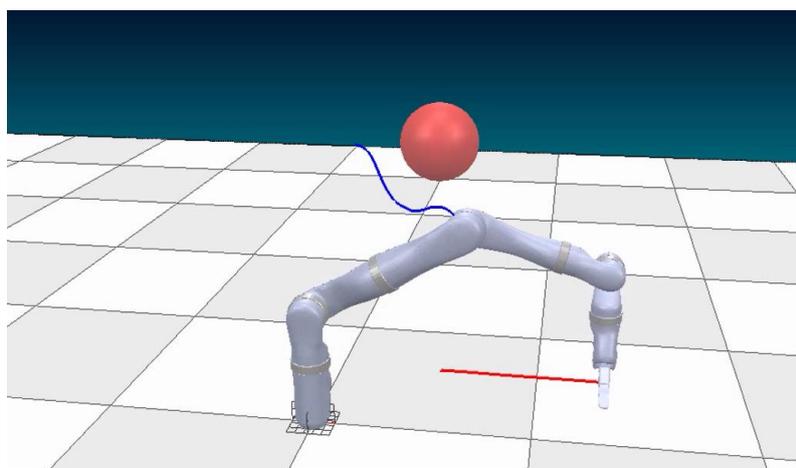
Nella figura 3.18 viene riportata la sequenza dei movimenti del manipolatore simulati nell'ambiente virtuale V-rep individuando in rosso la traiettoria dell'End Effector e in blu quella del gomito.



0%



50%



100%

Figura 3.19 Sequenza dei movimenti del robot simulati con V-rep lungo la traiettoria sfruttando la ridondanza in presenza di un ostacolo sferico.

CAPITOLO 4: Sperimentazione

Per dare maggiore completezza al lavoro di tesi si è proceduto con l'andare a visualizzare gli aspetti trattati tramite l'analisi teorica e le simulazioni sul software di calcolo, sul robot vero e proprio. L'obiettivo di tale sezione è quindi stato quello di capire come far comunicare MATLAB con il manipolatore Kinova Jaco 2 in modo da poterlo comandare tramite il software di calcolo.

4.1. Metodi di controllo del Kinova Jaco 2

Il braccio robotico oggetto di studio può essere comandato in diversi modi. Il più immediato e semplice è tramite l'apposito Joystick, di cui vediamo una rappresentazione in figura 4.1, che viene fornito assieme al robot e che viene collegato al braccio tramite l'apposita porta di connessione situata sulla base dello Jaco 2 e visibile in figura 4.2. In questo modo è possibile controllare il robot nello spazio cartesiano (più intuitivo) e nello spazio giunti (tale opzione deve essere attivata tramite il software di controllo da pc) [6].

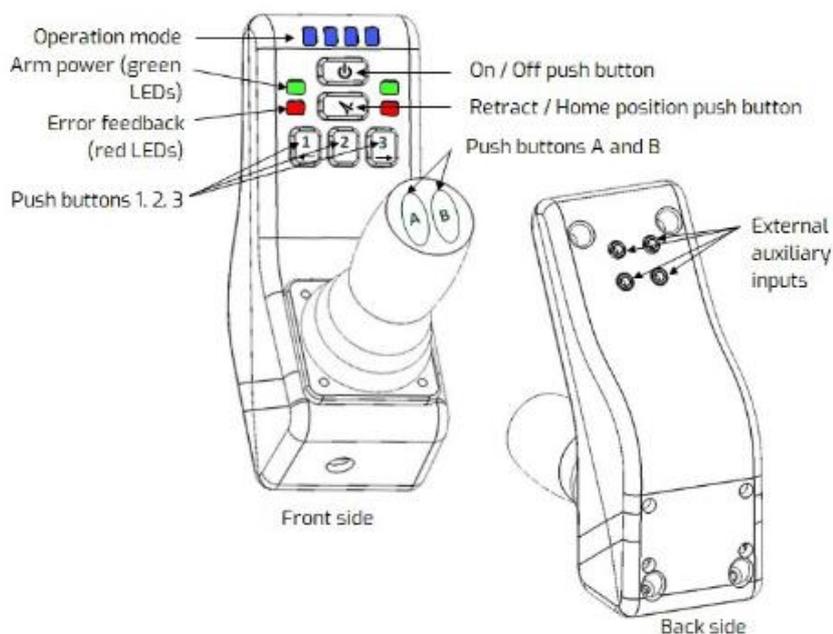


Figura 4.1 Rappresentazione del joystick di comando del Kinova Jaco 2 [6].

Un altro metodo per poter comandare il braccio robotico è attraverso l'utilizzo di un PC e dei due software di controllo sviluppati dalla Kinova Robotics:

- Development Center
- Torque Control

Per poter comandare il manipolatore tramite il PC è necessario, preliminarmente, connettere i due dispositivi tramite l'apposito cavo USB; la porta per la connessione del cavo USB al robot è situata sulla base ed è visibile in figura 4.2.

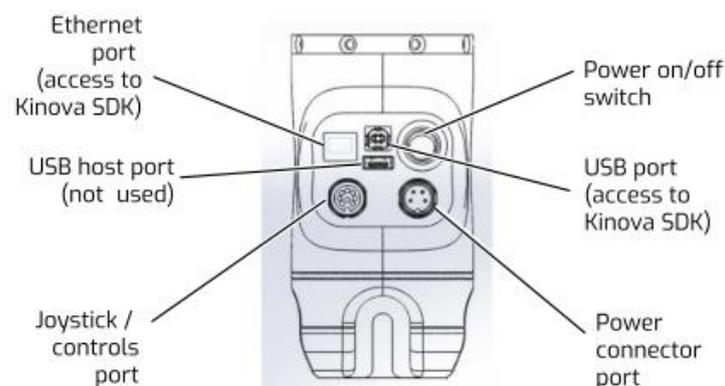


Figura 4.2 Rappresentazione della base del Kinova Jaco 2 con indicazione circa le porte di collegamento e i pulsanti che si trovano su di essa [7].

Tramite questi due software di controllo è possibile, attraverso l'uso del computer, comandare il braccio robotico in posizione e velocità, assegnare e pianificare delle traiettorie, controllare il robot in forza o coppia, ricevere informazioni come: posizione temperatura, intensità di corrente elettrica e coppia nei giunti [6]. In figura 4.3 e 4.4 si riporta l'interfaccia grafica dei due software di controllo Kinova.

L'ultimo metodo di controllo è quello meno immediato e più complesso e consiste nell'utilizzo di funzioni scritte in linguaggio C++ per comunicare con il manipolatore. La libreria di tali funzioni è definita Kinova API. Vista la complessità della scrittura di funzioni in linguaggio C++ la Kinova Robotics ha inoltre creato e messo a disposizione degli utenti un Wrapper MATLAB. Tramite l'utilizzo di questo Wrapper è possibile controllare il robot tramite degli script MATLAB. Per fare questo sono state create una serie di funzioni

MATLAB a cui sono associati dei programmi scritti in linguaggio C++ che vengono inviati al controllore del robot e da esso sono poi interpretati.

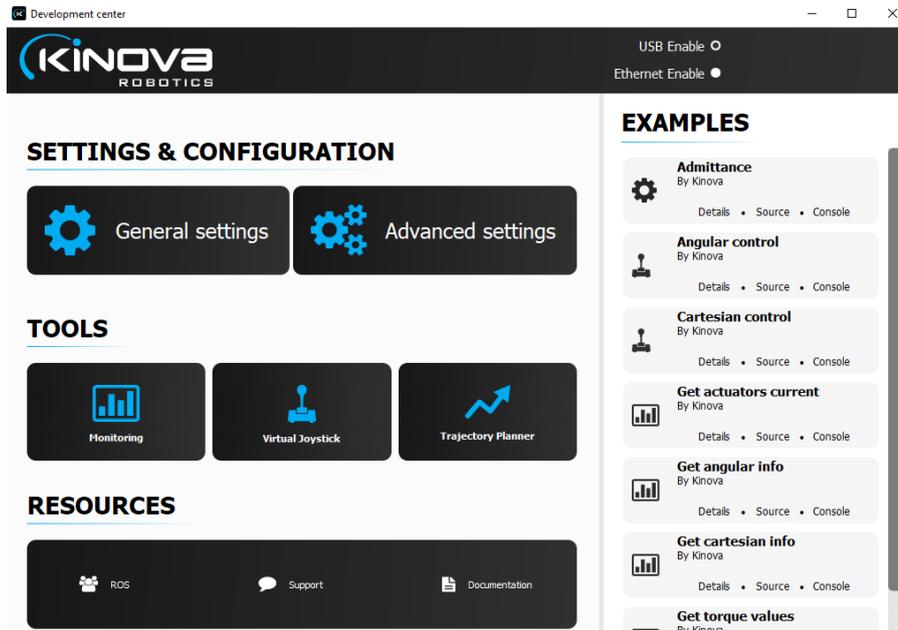


Figura 4.3 Interfaccia grafica del Development Center.

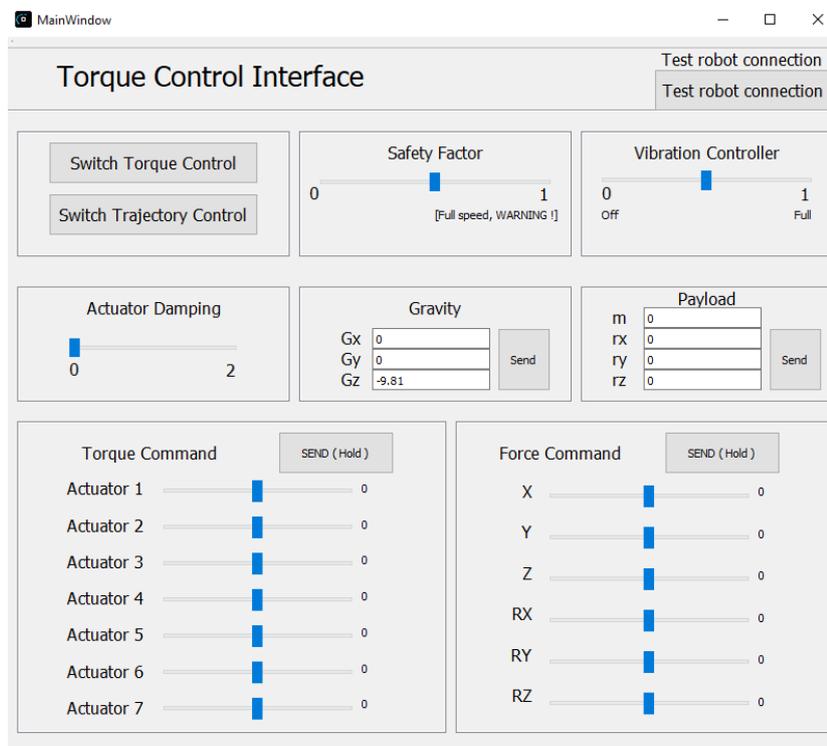
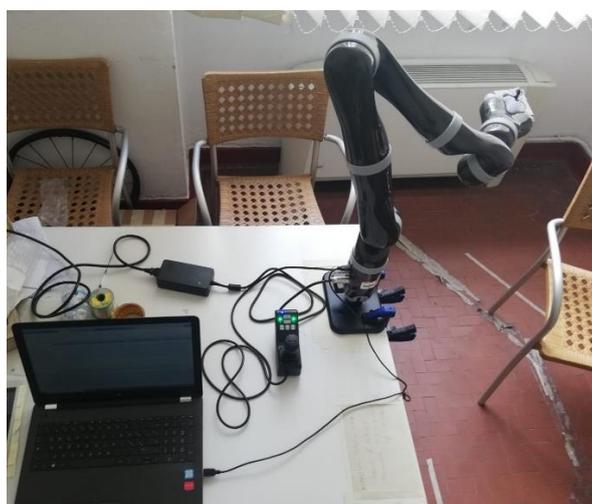


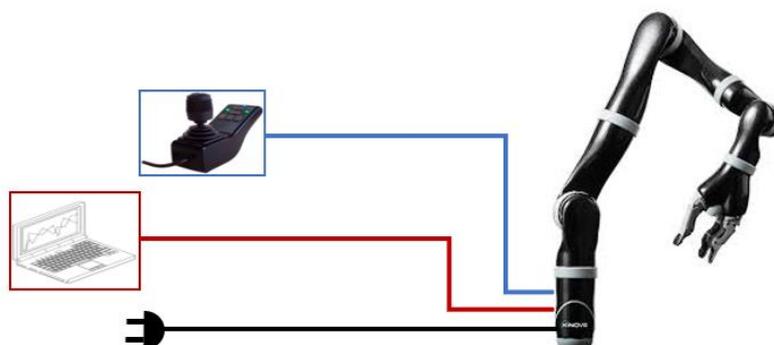
Figura 4.4 Interfaccia grafica del Torque Control.

4.2. Setup per la sperimentazione

La prima operazione da svolgere consiste nella corretta installazione del robot. Una volta verificato il corretto fissaggio del robot sulla base di lavoro è necessario procedere con il collegamento dei vari cavi. Come prima cosa si procede con la connessione del cavo di alimentazione e del Joystick al robot, si procede poi con il collegare l'alimentazione alla rete elettrica. Una volta effettuato il collegamento, premendo il tasto di accensione del robot situato sulla base e visibile in figura 4.2, si vedranno comparire delle luci verdi lampeggianti sul Joystick di comando. Una volta che queste luci verdi passeranno da essere lampeggianti a fisse significa che il robot è pronto all'uso. Al momento, in realtà, esso si trova ancora in posizione di Recovery, è necessario, per renderlo operativo, portarlo nella posizione di Home tenendo premuto l'apposito pulsante sul Joystick [6]. Per poter comandare il manipolatore tramite computer è, inoltre, necessario collegare, con l'apposito cavo, il PC alla base del robot. In figura 4.5 si può vedere una foto del banco sperimentale e dei collegamenti effettuati.



(a)



(b)

Figura 4.5 Banco sperimentale (a) e schematizzazione collegamenti (b).

In seguito, come detto in precedenza, per poter comandare il robot tramite MATLAB è necessario scaricare il Kinova MATLAB Wrapper dalla libreria degli AddOns del software. Aprendo la cartella scaricata si trova un file chiamato 'testJacoComm.m' in cui è possibile vedere e testare le varie funzioni MATLAB implementate. La cosa importante per il corretto funzionamento dei programmi tramite i quali si vuole comandare il robot è che venga aggiunta al percorso delle cartelle di lavoro di MATLAB la cartella del Wrapper con tutte le sottocartelle contenute in essa e che gli script vengano eseguiti avendo come 'Current folder' la cartella JACO2SDK contenuta a sua volta nella cartella JACO2Communicator. Di seguito è riportato l'intero percorso dati per lavorare nella corretta cartella file:

C:\Users\Documents\MATLAB\Add-Ons\Collections\Kinovarobotics_matlab_Kinovaapi_wrapper\code\Kinovarobotics_matlab_Kinovaapi_wrapper-e51b887\JACO2Communicator\JACO2SDK

Se il robot è collegato in modo corretto e ci si trova nella suddetta cartella, eseguendo i file MATLAB opportunamente scritti, il robot risponderà ai comandi dati. È possibile che ci sia ancora la necessità di correggere un errore che impedisce il corretto funzionamento del software. Tale errore consiste nel fatto che, eseguendo il programma dopo aver lanciato un'altra analisi su MATLAB, il PC non riconosce il robot e non riesce a comunicare con esso. Per porre rimedio a tale problema è necessario ricompilare i file MEX che serve per rendere disponibile in ambiente MATLAB delle funzioni in C/C++ indicate dall'utente. Come prima cosa, è necessario, qualora non lo si possieda già, installare sul computer una versione compatibile con la versione MATLAB utilizzata del programma Microsoft Visual C++ Professional. Dopo di che, bisogna selezionare la versione scelta di Microsoft Visual C++ Professional come compilatore all'interno dell'ambiente MATLAB, scrivendo all'interno della Command Windows i seguenti comandi:

```
>> mex -setup C++
```

Bisogna poi cancellare il file JacoMexInterface.mexw64 dalla cartella Source contenuta a sua volta nella cartella JACO2Communicator del pacchetto Kinova scaricato. Successivamente, è necessario cambiare all'interno del file compileMexInterface.m, il percorso dati contenuto nelle righe di codice inserendo il percorso del proprio PC. Infine, bisogna eseguire tale file in modo che venga generato il nuovo file JacoMexInterface.mexw64.

4.3. Funzioni utilizzate

In questa sezione si riportano indicazioni circa le principali funzioni MATLAB utilizzate per mettere in comunicazione il robot con il PC. Poiché lo scopo dell'attività è quello di far muovere il robot dando in ingresso al controllore un'informazione sui gradi di libertà nei giunti in termini di posizione o velocità, le funzioni studiate sono solo quelle utili a tale scopo.

Tabella 4.1 Principali funzioni utilizzate per comandare il robot.

<code>jc=JacoComm</code>	Comando per creare un oggetto <code>jc</code> della classe <code>JacoComm</code> a cui sono associate sia la struttura dati del robot che le procedure che operano su di esso
<code>connect(jc)</code>	Comando per connettere il PC con il robot
<code>setPositionControlMode(jc);</code>	Comando per attivare il controllo in posizione dei giunti (è già attivato di default)
<code>setVelocityControlMode(jc)</code>	Comando per attivare il controllo in velocità dei giunti
<code>sendJointPositionCommand(jc,jntCmd)</code>	Comando per inviare al robot i valori dei gradi di libertà nei giunti (<code>jntCmd</code>).

<code>sendFingerPositionCommand(jc, fCmd)</code>	Comando per inviare al robot i valori della posizione dei finger del gripper (fCmd)
<code>sendJointAndFingerPositionCommand(jc, jntCmd, fCmd)</code>	Comando per inviare al robot i valori dei gradi di libertà nei giunti (jntCmd) e della posizione dei finger dell'End Effector (fCmd)
<code>sendJointVelocityCommand(jc, jntVel)</code>	Comando per inviare al robot i valori delle velocità ai giunti (jntVel).
<code>getJointAndFingerPos(jc)</code>	Comando per visualizzare i gradi di libertà nei giunti e nelle dita del gripper
<code>Disconnect(jc)</code>	Comando per disconnettere il PC dal robot

4.4. Comando in posizione

Di seguito si riporta lo script MATLAB implementato per far muovere il robot inviando i valori dei gradi di libertà.

```
clear all
clc
```

```
%% Connessione al robot
jc = JacoComm
connect(jc)
```

```
%% Caricamento dei gradi di libertà del movimento desiderati
load trajectory_obstacle_q3cost.mat
```

```
%% Controllo dei gradi di libertà dei giunti
setPositionControlMode(jc);
```

```
%% Posizione iniziale della traiettoria
jntCmd=Sol(1,1:7)';
sendJointPositionCommand(jc,jntCmd);
pause(5);
```

```
%% Posizioni lungo la traiettoria
for i=2:size(Sol,1)
    jntCmd=Sol(i,1:7)';
    sendJointPositionCommand(jc,jntCmd);
end
```

```
%% Disconnessione del robot
disconnect(jc)
```

La prima sezione esegue il collegamento del PC con il robot creando l'oggetto `jc`. Nella seconda sezione si procede con il caricamento della matrice formata da 7 colonne contenenti i 7 gradi di libertà di giunto e tante righe quanti sono i punti in cui è stata divisa la traiettoria. La funzione utilizzata nella terza sezione imposta il metodo di controllo in controllo posizione e potrebbe anche non essere implementata in quanto tale metodo di controllo è già attivo di default all'avvio del robot. Nella quarta sezione si procede con l'invio al robot della configurazione iniziale. Grazie a questo comando il robot si porta dalla configurazione di Home a quella iniziale della traiettoria pianificata. Dopo un tempo scelto di 5 secondi si attiva la quinta sezione in cui, tramite un ciclo `for`, si procede con l'inviare in sequenza tutti i gradi di libertà necessari perché il robot esegua la traiettoria desiderata. Durante

l'esecuzione di tale comando si ha il movimento del robot lungo la traiettoria. Al termine, con la sesta sezione, si procede alla disconnessione del robot dal computer.

È necessario precisare come, comandando il robot in posizione e non in velocità è opportuno inviare al controllore delle traiettorie campionate molto fittamente in modo da non vedere scatti o discontinuità durante il movimento. Nonostante si infittisca molto la traiettoria però, si notano comunque delle vibrazioni indotte da un movimento a scatti da parte del braccio robotico. Inoltre, con questo genere di comando non si ha nessuna possibilità di controllo sul tempo di percorrenza della traiettoria. Infatti, il robot si sposta da una configurazione all'altra secondo una sua determinata legge di velocità che si è verificato essere di tipo trapezoidale ed è, quindi, impossibile conoscere a priori il tempo Δt che passa tra una configurazione e quella successiva. Il valore di tale intervallo di tempo, oltre a essere del tutto ignoto cambia anche lungo la traiettoria.

Sono state svolte diverse prove di movimento in controllo di posizione con lo Jaco 2. Il primo test di movimento è stato svolto sulla circonferenza di Swivel. È stata scelta una traiettoria che mantiene fisso l'End Effector facendo variare la posizione del gomito lungo tutta la circonferenza di Swivel. È stato individuato un punto tale per cui i limiti di giunto permettessero di avere la circonferenza completa. Grazie all'analisi svolta nella sezione 2.7. di questo elaborato la determinazione di un punto con tale caratteristiche è risultata immediata. Per ottenere un movimento fluido del gomito lungo la circonferenza di Swivel è importante verificare che durante il processo di cinematica inversa la risoluzione numerica non abbia prodotto degli andamenti dei gradi di libertà con dei salti tra -180° e $+180^\circ$. Se ciò succede è necessario correggere gli andamenti facendo traslare i valori dei gradi di libertà di roll di 360° ; essendo i limiti su tali giunti pari a $\pm 10000^\circ$, con questa operazione non si incorre nel rischio di avere dei valori di gradi di libertà fuori dai range accettabili. In figura 4.6 e 4.7 sono rappresentati gli andamenti dei gradi di libertà nei giunti prima e dopo l'operazione di correzione.

È, infine, utile precisare come nel momento in cui si invia la posizione iniziale con il comando `sendJointPositionCommand` il robot si porta dalla configurazione in cui si trova a quella desiderata. La traiettoria che il robot deve seguire per compiere questo movimento è così definita nello spazio giunti; ciò comporta che la traiettoria del robot nello spazio operativo non è nota a priori. Questa condizione potrebbe quindi portare il robot ad

urtare il piano di lavoro o oggetti intorno ad esso. Per evitare tale fenomeno si è implementato l'algoritmo seguente.

```
J_pos_start=jc.JointPos; %Posizione attuale
jntCmd=Sol(1,1:7)'; %Posizione richiesta

J_pos_start(2)=pi;
J_pos_start(4)=pi/2;
sendJointPositionCommand(jc,J_pos_start)
J_pos_start(1)=jntCmd(1);
sendJointPositionCommand(jc,J_pos_start)

for i=5:7
    J_pos_start(i)=jntCmd(i);
    sendJointPositionCommand(jc,J_pos_start);
end
pause
J_pos_start(3)=jntCmd(3);
J_pos_start(4)=jntCmd(4);
sendJointPositionCommand(jc,J_pos_start)
J_pos_start(2)=jntCmd(2);
sendJointPositionCommand(jc,J_pos_start)
```

La prima operazione consiste nel ruotare il secondo giunto di 180° e il quarto giunto di 90° in modo da portare il robot a lavorare su un piano sopraelevato rispetto al livello del piano d'appoggio. Si procede ruotando i tre giunti del polso del valore richiesto, successivamente si azionano a cascata il giunto 4, 3 e 2 per portare il robot nella configurazione desiderata evitando contatti con il piano di lavoro e oggetti intorno ad esso.

Le altre prove che si sono realizzate hanno avuto l'obiettivo di riprodurre le traiettorie studiate nel capitolo 3. Durante l'esecuzione di tutte queste traiettorie si è notato, come detto precedentemente, un movimento non fluido del manipolatore. Per cercare di migliorare il movimento del braccio robotico si è cercato di infittire le traiettorie inviando i valori dei gradi di libertà di giunto in più configurazioni intermedie. Il risultato è stato quello di un miglioramento della fluidità ma anche di un rallentamento nella percorrenza della traiettoria. Per muovere il manipolatore in modo migliore e potendo controllare anche il tempo si è passati al controllo in velocità.

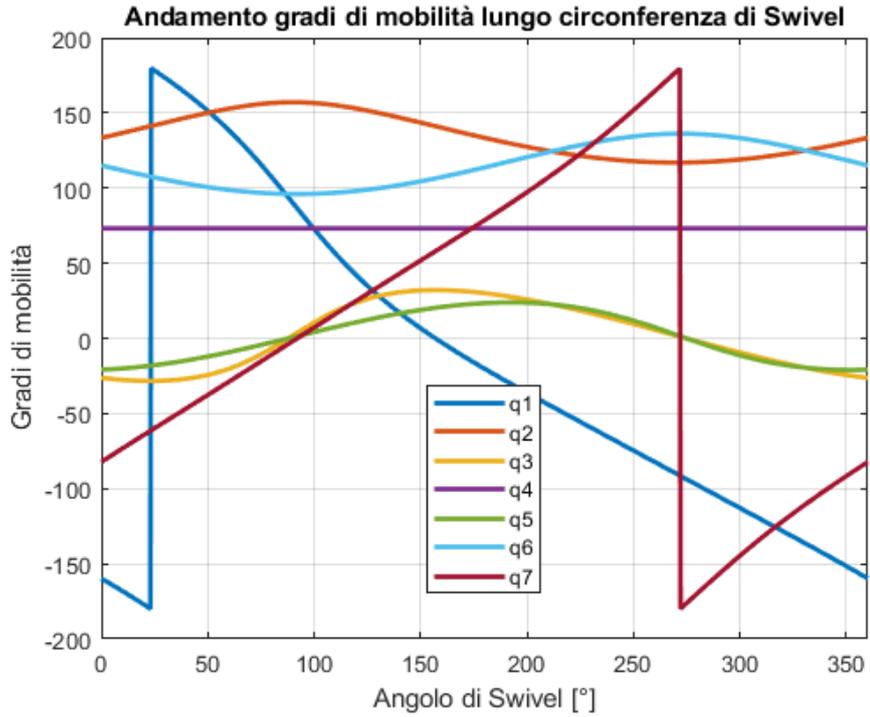


Figura 4.6 Andamento dei gradi di libertà dei giunti perchè il gomito percorra tutta la circonferenza di Swivel.

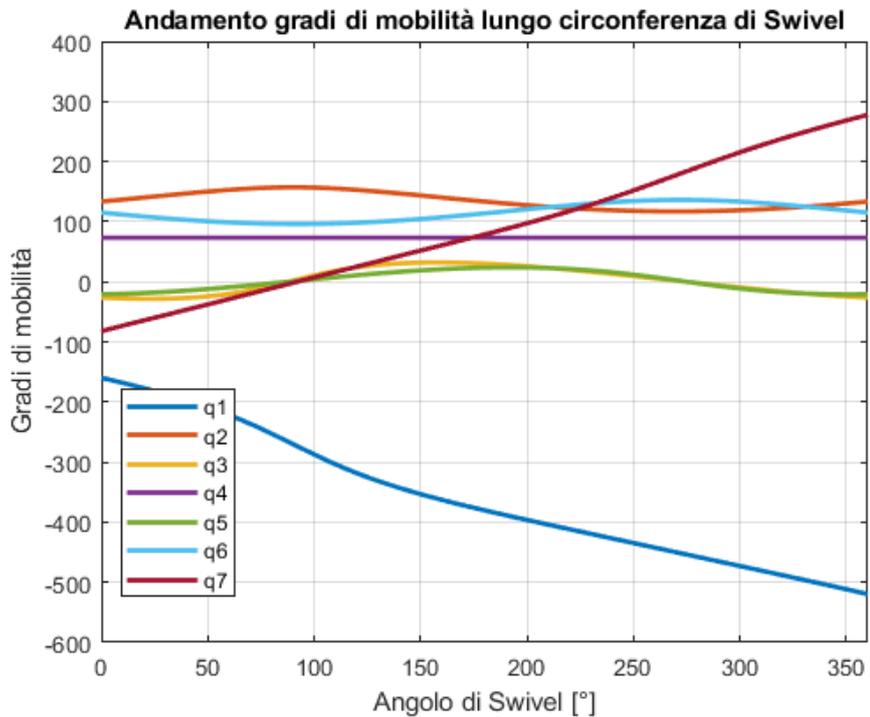


Figura 4.7 Andamento dei gradi di libertà nei giunti che produce il medesimo effetto di quello in figura 4.6 senza avere dei salti generati dalla soluzione numerica dell' algoritmo di cinematica inversa.

4.5. Comando in velocità

Di seguito si riporta lo script MATLAB implementato per far muovere il robot inviando i valori di velocità nei giunti.

```
t_freq=0.01; % s
t_cycle=zeros(length(Vel),1);
param_pause=1.02;
```

```
%% Connessione al robot
jc = JacoComm
connect(jc)
```

```
%% Posizione iniziale della traiettoria
jntCmd=Sol(1,1:7)';
sendJointPositionCommand(jc,jntCmd);
pause
```

```
%% Comando per attivare controllo in velocità
setVelocityControlMode(jc);
```

```
%% Velocità lungo la traiettoria
tic
for i=1:length(Vel)

    tStart=tic;
    vel_jc=Vel(i,:)';
    sendJointVelocityCommand(jc,vel_jc);

    t_cycle(i,1)=toc(tStart);

    if t_cycle(i,1)<t_freq
        differenza=(t_freq-t_cycle(i,1))/param_pause;
        java.lang.Thread.sleep(differenza*1000)
        %pause(differenza)
    end
end
toc
```

```
%% Disconnessione del robot
disconnect(jc)
```

Nella prima sezione si dichiarano delle variabili la cui utilità verrà mostrata nel seguito del paragrafo. Dopo di che, come nel caso di comando in posizione, si procede con l'eseguire la connessione tra il PC e il robot. Nella terza sezione si inviano i valori dei gradi di libertà associati alla posizione iniziale lungo la traiettoria (anche in questo caso, in realtà, il comando `sendJointPositionCommand` è stato sostituito con l'algoritmo presentato alla fine del paragrafo precedente). Successivamente, si passa in controllo di velocità. Nella quinta sezione si procede con l'invio delle velocità nei giunti al robot tramite un ciclo `for`.

All'interno di tale ciclo è necessario inserire un controllo temporale per avere la certezza che il robot effettui la traiettoria desiderata e nel tempo richiesto. Infatti, è noto che il valore di velocità inviato deve essere mantenuto per 0.01s (periodo di elaborazione del controllore del manipolatore) per avere il corretto movimento del robot, il PC, d'altro canto, elabora il ciclo `for` con una frequenza maggiore. Il risultato di questo fenomeno è che, poiché il ciclo `for` viene compilato in meno tempo rispetto a quello che il controllore impiega ad elaborare i dati in ingresso, il robot percorre solo un pezzo della traiettoria. È necessario quindi, inserire una pausa all'interno del ciclo `for` in modo da permettere la sincronia tra compilazione dello script ed elaborazione da parte del controllore del dato inviato. Per fare questa operazione, si inserisce la funzione MATLAB `tic toc` per misurare il tempo che il computer utilizza per elaborare un ciclo `for`. Si confronta poi tale tempo con il periodo associato alla frequenza di lavoro del manipolatore e si inserisce all'interno della routine una pausa pari alla differenza tra questi due valori. Si può notare come, nel nostro caso, la differenza tra queste due grandezze sia poi stata divisa per un parametro leggermente maggiore di 1. Tale parametro, ricavato sperimentalmente, serve per compensare i piccoli errori temporali sulla pausa, in modo da avere il tempo totale di movimento molto prossimo a quello richiesto. È utile precisare come, trattandosi di tempi inferiori al centesimo di secondo la funzione `pause` di MatLab non è abbastanza precisa ma è necessario richiamare la funzione Java- `java.lang.Thread.sleep()` in cui il tempo in input è da inserire in millisecondi. Infine, nella sesta sezione, si procede con la disconnessione del robot dal computer.

Una volta noto il modo di comunicare in velocità con il robot, risulta utile precisare come si sono ricavati gli andamenti delle velocità nei giunti. Nella pianificazione di traiettoria utilizzata per comunicare con il robot in posizione il parametro di curva viene fatto variare da 0 a 1 in modo lineare, ovvero con una velocità costante. In questo caso, invece, si è definito un parametro di curva S il cui valore varia da 0 a 1 con una legge a trapezio di velocità visibile in figura 4.8. Il risultato di questa parametrizzazione è quello di creare una partizione non equispaziata dei punti di calcolo dei gradi di libertà nei giunti. Dal punto di vista pratico i valori dei gradi di libertà nei giunti nei nuovi punti non sono stati ricalcolati da zero ma tramite la funzione `interp1` di MATLAB sono stati ricavati per interpolazione a partire dai valori usati per comandare il robot in posizione. Una volta ottenuti tali valori, le velocità sono state calcolate approssimando la derivata facendo uso della definizione di rapporto incrementale. È quindi stata calcolata la differenza dei gradi

di libertà istante dopo istante e tale differenza è stata divisa per l'intervallo di tempo corrispondente.

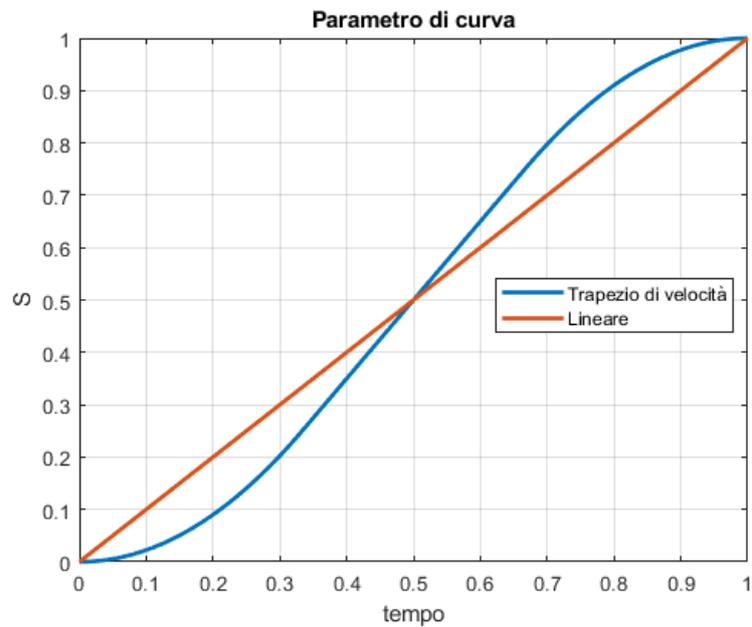


Figura 4.8 Confronto tra parametro di curva S con legge lineare e a trapezio di velocità.

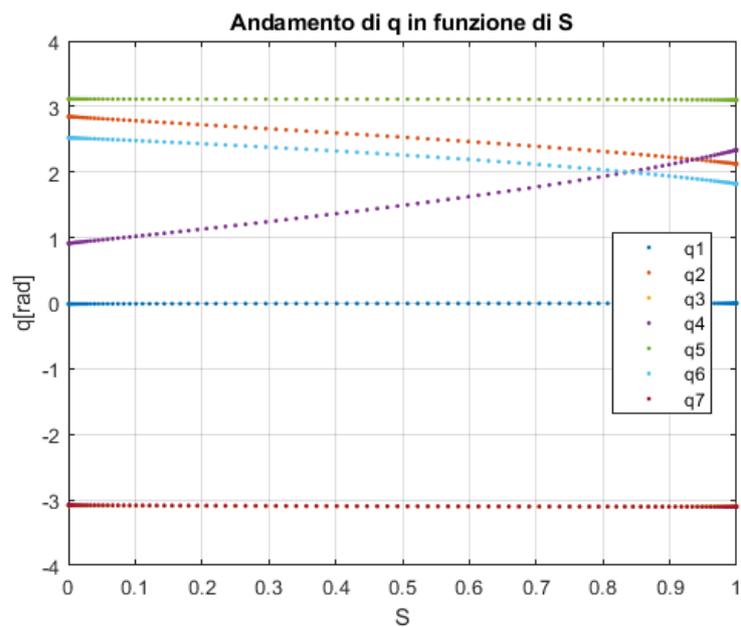


Figura 4.9 Andamento dei gradi di libertà in funzione del parametro di curva S con legge a trapezio di velocità.

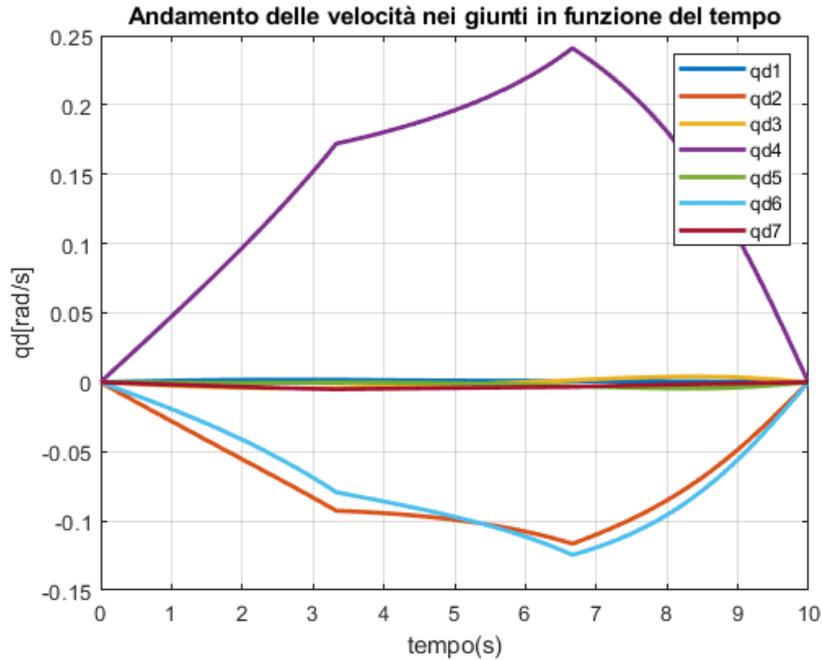


Figura 4.10 Andamento delle velocità nei giunti in funzione del tempo.

Con questo metodo di controllo sono state svolte le medesime prove effettuate in controllo posizione. La prima traiettoria implementata in controllo velocità è stata la circonferenza di Swivel. Tale traiettoria è stata fatta percorrere al robot variando il tempo totale di movimento finché non si è trovato il giusto compromesso tra velocità di esecuzione e non superamento dei limiti di velocità di giunto. Tale operazione è stata fatta sperimentalmente, in quanto non c'è modo di conoscere tramite le funzioni MATLAB, il valore dei limiti di velocità. Nonostante nessun limite venisse saturato durante l'esecuzione della traiettoria si è notato che l'End Effector non riuscisse esattamente a mantenere la posa richiesta. La causa di questo fenomeno trova spiegazione nel fatto che il robot è controllato in anello aperto senza nessuna retroazione in grado di confrontare la posa richiesta con quella attuale del manipolatore. Per migliorare l'esecuzione alla traiettoria sarebbe quindi opportuno passare a un controllo in anello chiuso e inserire un termine di velocità correttivo per compensare la deriva nella posa dell'End Effector.

Dopo la circonferenza di Swivel si sono realizzate le traiettorie studiate nel capitolo 3 di questo elaborato. A queste traiettorie sono state aggiunti, prima e dopo, degli opportuni movimenti del braccio e di apertura e chiusura delle pinze del gripper per simulare delle operazioni di pick and place. Nel caso specifico, a causa di un guasto del gripper, i

movimenti di apertura e chiusura pinza non si sono potuti realizzare. Di seguito si riportano alcune immagini del robot durante l'esecuzione dei vari movimenti implementati.



0%



50%



100%

Figura 4.11 Traiettorie con ostacolo simulato dalla mano nel caso di robot utilizzato come un 6DOF (prima colonna) o sfruttando la ridondanza (seconda colonna).



0%



50%



100%

Figura 4.12 Traiettorie in presenza di un muro simulato da un foglio nel caso di robot utilizzato come un 6DOF (prima colonna) o sfruttando la ridondanza (seconda colonna).

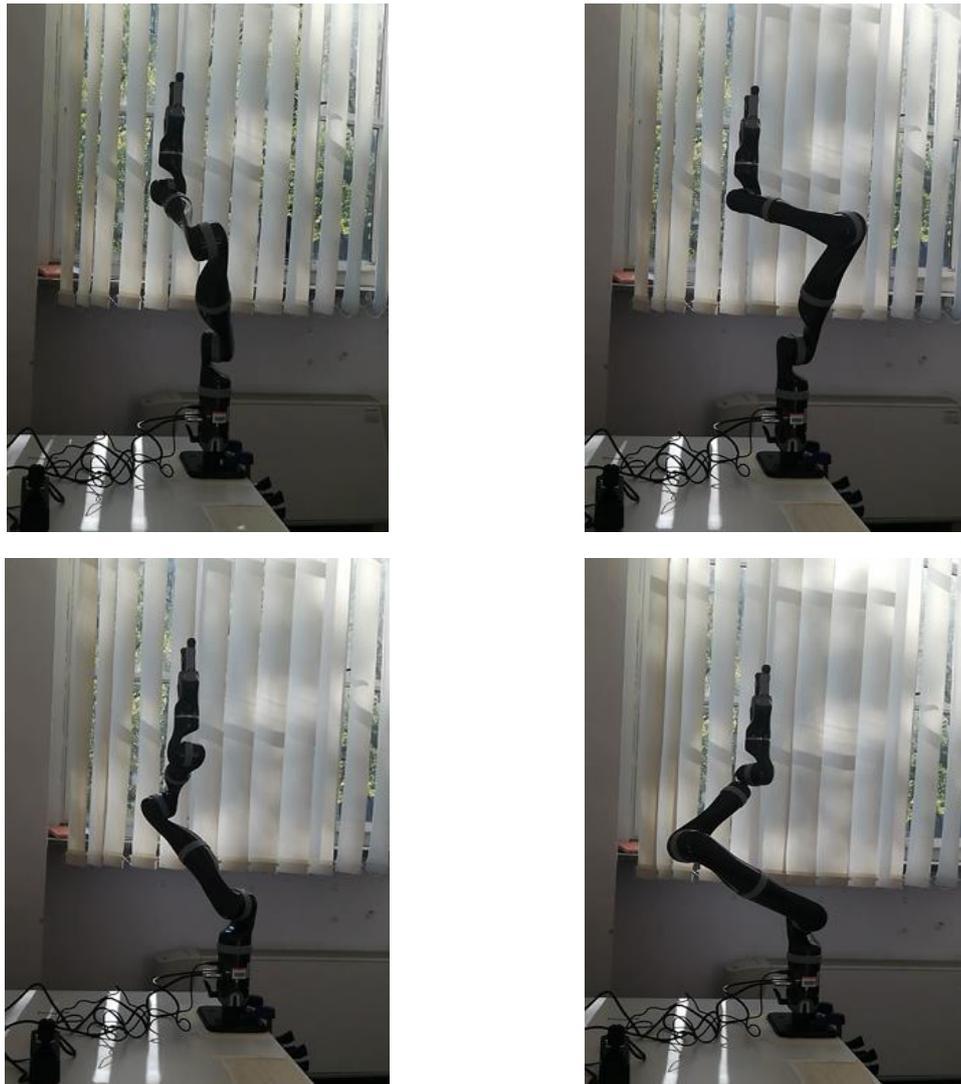


Figura 4.13 Fotogrammi lungo la circonferenza di Swivel.

A seguito di tale sperimentazione si è notato come il metodo di controllo del robot tramite MATLAB sia ancora molto basico e debba ancora essere sviluppato. Come detto in precedenza, infatti, non c'è modo di conoscere i limiti di velocità i quali risultano essere più bassi di quelli che si hanno lavorando tramite il Development Center. Inoltre, non c'è nessuna conoscenza circa l'accelerazione; non si conosce il suo valore limite e non si conosce la legge di variazione di questa grandezza, anche se, tramite alcune prove, si è notato che durante il comando in posizione il robot sfrutta per ogni giunto sempre il valore massimo di velocità e accelerazione. Allo stato attuale, per avere un controllo migliore sul robot è necessario usare direttamente il linguaggio C++ senza passare attraverso il Wrapper MATLAB che risulta ancora limitante.

CONCLUSIONE

Nel presente elaborato si sono studiati nel dettaglio alcuni aspetti della cinematica inversa di manipolatori concentrando particolarmente l'attenzione sul tema della ridondanza. Il primo aspetto di interesse rilevato è legato allo studio dei manipolatori a sei gradi di libertà. Grazie all'analisi sulla distribuzione del numero di soluzioni all'interno del workspace è stato possibile mostrare come, sebbene la cinematica inversa garantisca l'esistenza di 4 configurazioni di braccio per ogni posizione richiesta del centro polso, in realtà, le soluzioni effettivamente ottenibili sono, quasi sempre, molte meno. I risultati di questo studio permettono di conoscere, attraverso la mappatura colorata realizzata, come il numero delle soluzioni alla cinematica inversa è distribuito nello spazio di lavoro. Un secondo aspetto che è stato messo in luce circa i robot a sei gradi di libertà è la possibilità di individuare un indice in grado di fornire indicazioni circa la reale destrezza del manipolatore. Si è infatti mostrato, come essa non è solo legata alla destrezza del polso ma dipende dalle varie configurazioni di braccio in grado di portare il centro polso nella posizione desiderata.

Grazie allo studio sul robot a sette gradi di libertà Kinova Jaco 2 è poi stato realizzato un algoritmo in grado di calcolare analiticamente le soluzioni alla cinematica inversa di un manipolatore ridondante che presenta un offset sul gomito, conoscendo la posa dell'End Effector. Sempre attraverso lo studio di tale robot, si è poi messo in luce quali sono le potenzialità di un manipolatore ridondante in termini di non collisione con oggetti presenti nello spazio di lavoro. Infine, attraverso la sperimentazione, si è realizzata una sorta di piccolo manuale per poter comandare in posizione o velocità il robot Kinova Jaco 2 tramite MATLAB.

A partire da questo elaborato si potrebbe ampliare lo studio relativo alla ridondanza nei robot a sei gradi di libertà andando a considerare, oltre ai limiti di giunto, anche problemi di compenetrazione tra i link. Per quanto riguarda lo Jaco 2, invece, se da un lato questa tesi ne illustra nel dettaglio la cinematica inversa, dall'altro rappresenta solo uno studio preliminare di come la sua struttura possa essere sfruttata e di come esso possa essere controllato tramite computer. Sviluppare tale aspetto potrebbe portare tale robot, nato per effettuare operazioni di assistenza medica ai disabili comandato tramite joystick, a diventare fruibile in applicazioni di tutt'altra natura.

RINGRAZIAMENTI

Al prof. S.Pastorelli, grazie per avermi guidato con grande disponibilità ed attenzione nell'elaborazione della tesi.

All'Ing. L.Scimmi, grazie per la disponibilità, per la pazienza e per la generosità con la quale ha condiviso con me tutte le informazioni già in suo possesso sul robot oggetto di studio.

Alla mia famiglia, grazie per essermi stata sempre vicino, per i consigli, per avermi permesso di raggiungere questo obiettivo e per avermi appoggiato e incoraggiato in ogni scelta fatta.

A Sylvie, grazie per esserci sempre stata, per avermi ascoltato e per essere stata un appoggio fondamentale in questi anni.

Ai ragazzi di Via Brunetta, grazie per avermi permesso di vivere questi 5 anni in un ambiente sereno, grazie per la complicità, per le risate e per l'affetto dimostrato. Gli anni trascorsi con voi rimarranno sempre un piacevole ricordo.

A tutti i miei amici, grazie per avermi accompagnato in questo percorso e per aver reso questo cammino meno duro e faticoso.

BIBLIOGRAFIA

- [1] A. Campeau-Lecours, H. Lamontagne, S. Latour, P. Fauteux, V. Maheu, F. Boucher, C. Deguire, L. J. Caron L'Ecuyer, «Kinova Modular Robot Arms for Service Robotics Applications», *International Journal of Robotics Applications and Technologies*, Vol.5, Issue 2, July-December 2017.
- [2] F. Capetti, «Analisi della ridondanza nei manipolatori collaborativi: studio di indici caratteristici di configurazione », *Tesi di Laurea Magistrale*, Politecnico di Torino, 2018.
- [3] Z. Chen, Z. Zeng, G. Shu, Q.Chen, «Kinematic solution and singularity analysis for 7-DOF redundant manipulators with offsets at the elbow», *Proceedings - 2018 IEEE Industrial Cyber-Physical Systems*, pp. 422-427, 2018.
- [4] P. Corke, «Robotics, Vision and Control, Fundamental algorithms in MATLAB», Springer, 2017.
- [5] P. Corke, «Robotics Toolbox for MATLAB Release 10», 2017.
- [6] Kinova Robotics Inc., «Kinova Ultra lightweight robotic arm user guide», 2018.
- [7] Kinova Robotics Inc., «Kinova Controller user guide», 2018.
- [8] Kinova Robotics Inc., «Kinova software development kit», 2018.
- [9] S. Pastorelli, *Slides corso Dispositivi e sistemi robotici*, Politecnico di Torino, 2018.
- [10] B. Siciliano, L. Sciavicco, L. Villani e G. Oriolo, «Robotics Modelling, Planning and Control », Springer, 2009.
- [11] M. Shimizu, H. Kakuya, W. Yoon, K. Kitagaki, K. Kosuge, «Analytical inverse kinematic computation for 7-DOF redundant manipulators with joint limits and its application to redundancy resolution », *IEEE Transactions on Robotics*, Vol.24, Issue 5, pp. 1131-1142, 2008.
- [12] C. Yu, M. Jin, H. Liu, «An analytical solution for inverse kinematic of 7-DOF redundant manipulators with offset-wrist», *2012 IEEE International Conference on Mechatronics and Automation*, pp.92-97, 2012.
- [13] www.kinovarobotics.com
- [14] www.abb.com/robotics