



UNIVERSITÀ DI BRESCIA
FACOLTÀ DI INGEGNERIA
Dipartimento di Ingegneria dell'Informazione

Laboratorio di Robotica Avanzata
Advanced Robotics Laboratory

Corso di Robotica Mobile
(Prof. Riccardo Cassinis)

**Controllo di posizione di un robot
articolato mediante Kinect**

Elaborato di esame di:

**Francesco Compagnoni, Roberto
Rossini, Gianluca Rossi**

Consegnato il:

17 luglio 2014



Controllo di posizione di un robot articolato mediante Kinect by Francesco Compagnoni, Roberto Rossini, Gianluca Rossi is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

Sommario

Questo elaborato ha come finalità il controllo remoto del manipolatore industriale fisso Kawasaki RS03N mediante l'utilizzo di Microsoft Kinect. Per fare ciò verrà implementato un software in grado di sfruttare il Microsoft Software Development Kit per permettere a Kinect di acquisire i movimenti del braccio umano e controllare in questo modo il manipolatore. Dopo una breve panoramica della strumentazione necessaria al raggiungimento dello scopo prefissato, vengono descritte le criticità e le problematiche incontrate durante i lavori. Successivamente viene descritta la soluzione adottata e la modalità operativa con cui si è giunti all'interazione tra il braccio umano ed il manipolatore Kawasaki RS03N. Per fare ciò si trova in allegato tutto il materiale e le istruzioni necessarie a chiunque voglia cimentarsi nell'analisi o nell'ampliamento del lavoro svolto.

1. Introduzione

In questo capitolo vengono presentati gli strumenti di lavoro e l'ambiente software utilizzati per controllare da remoto il robot industriale Kawasaki RS03N.

1.1. Microsoft Kinect



Fig. 1 - Dispositivo Microsoft Kinect

Il Kinect è un accessorio originariamente pensato come dispositivo di controllo per la console Microsoft Xbox 360. Col passare del tempo si è intensificato l'interesse verso questa apparecchiatura tanto da suscitare l'interesse di molti ingegneri per lo sviluppo di diverse applicazioni grazie alla diffusione dei driver per l'utilizzo della videocamera RGB e dei sensori di profondità. Infatti il Kinect offre la possibilità, tramite cavo USB, di esser connesso ad un normale PC.

La struttura del Kinect si presenta come segue:

- **Video camera:** si tratta di una comune video camera in grado di fornire immagini con risoluzione 640x480 a colori;
- **Sensori di profondità:** rappresentano uno dei punti di forza del dispositivo e sono in grado di recuperare un "immagine" in cui sono riportati i dati di distanza di ciò che appare davanti al Kinect;

- **Batteria di microfoni:** si tratta di 4 microfoni in grado di fornire funzionalità di pulitura suono, posizionamento della sorgente sonora, cancellazione dell'eco;
- **Inclinazione motorizzata:** il dispositivo può essere brandeggiato (da codice) con un angolo verticale di ± 23 gradi rispetto al piano orizzontale.

1.2. Microsoft SDK

Per poter sviluppare applicazioni che sfruttano le potenzialità del Kinect è necessario scaricare il Software Development Kit (d'ora in poi SDK) che possiamo trovare all'indirizzo <http://www.microsoft.com/en-us/download/details.aspx?id=36996>.

L'SDK fornisce una libreria in grado di recuperare e gestire flussi di dati provenienti dai sensori e può essere utilizzata sia in C++ che con altri linguaggi compatibili, come il C#.

I requisiti hardware da soddisfare sono i seguenti:

- Processore dual-core con cpu a 2.66GHz o superiore;
- Almeno 2GB di RAM;
- Scheda grafica compatibile con Windows 7 che supporti DirectX 9.0;
- Kinect (ovviamente) e cavo USB.

I requisiti software utilizzati invece sono:

- Visual Studio Express 2010

Dopo aver installato l'SDK, è possibile collegare il dispositivo Kinect al PC tramite l'apposito cavo USB. La prima volta che si collega il Kinect, si noterà che il led diventerà rosso ed il sistema inizierà l'installazione automatica dei driver su PC.

1.3. Kawasaki

Il Kawasaki RS03N è un manipolatore industriale fisso a sei gradi di libertà ed è composto da tre parti fondamentali :

1. Braccio (Fig. 2 -
2. Controller
3. Teach Pendant

1.3.1. Braccio

Il braccio rappresenta la parte più caratteristica di un robot industriale in quanto è quella che fisicamente si muove e interagisce direttamente con il mondo fisico eseguendo vari compiti. Il braccio ha sei giunti, tanti quanti i gradi di libertà.



Fig. 2 - Braccio del manipolatore articolato Kawasaki

Le modalità principali di movimento sono con le posizioni dei giunti (JOINT) o con le coordinate spaziali rispetto alla base XYZOAT (BASE).

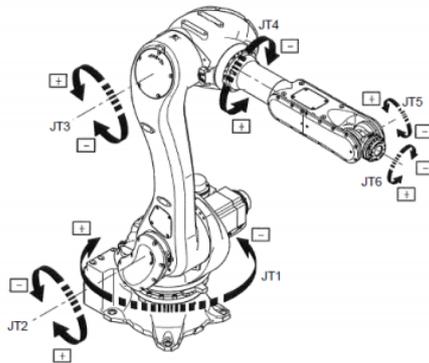


Fig. 3 - Movimento in base alla posizione dei giunti

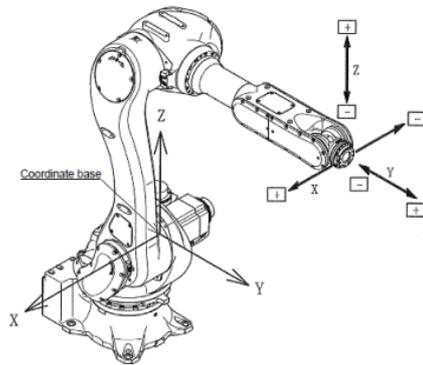


Fig. 4 - Le coordinate base di riferimento

In Fig. 3 - si controllano direttamente i movimenti dei singoli giunti, mentre nella Fig. 4 - modalità di controllo dei movimenti secondo le coordinate di base, o XYZOAT, è il controller a preoccuparsi dello spostamento dei giunti in modo da permettere lo spostamento del polso nelle direzioni desiderate.

1.3.2. Controller

Il controller (Fig. 5 - comprende:

1. Interruttore generale necessario per avviare il Robot
2. Il pannello Operativo dal quale è possibile scegliere tre diverse configurazioni in cui far partire il manipolatore :
 - a. 100% : modalità di verifica rapida del programma per la ripetizione automatica;
 - b. TEACH : modalità che permette all'utente tramite l'utilizzo del Teach Pendant , che spiegheremo nel paragrafo successivo, di insegnare al robot come compiere determinate operazioni;
 - c. REPEAT: modalità che permette di far eseguire in maniera automatica i programmi presenti.

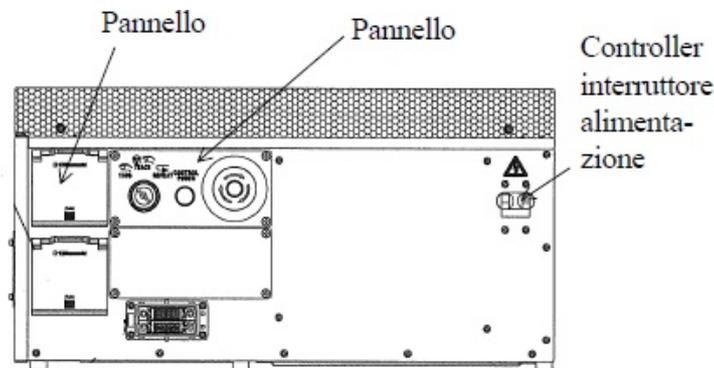


Fig. 5 - Controller

1.3.3. Teach Pendant

Il Teach Pendant (Fig. 6 - è un dispositivo che viene utilizzato dagli utenti per muovere il manipolatore in maniera manuale.

Questo dispositivo è composto da :

1. Uno schermo touchscreen
2. Tastiera Hardware
3. Pulsante per arrestare tutte le operazioni che sta compiendo il robot in caso di emergenza
4. Pulsante blocco insegnamento (T.LOCK)
5. Interruttore di sicurezza

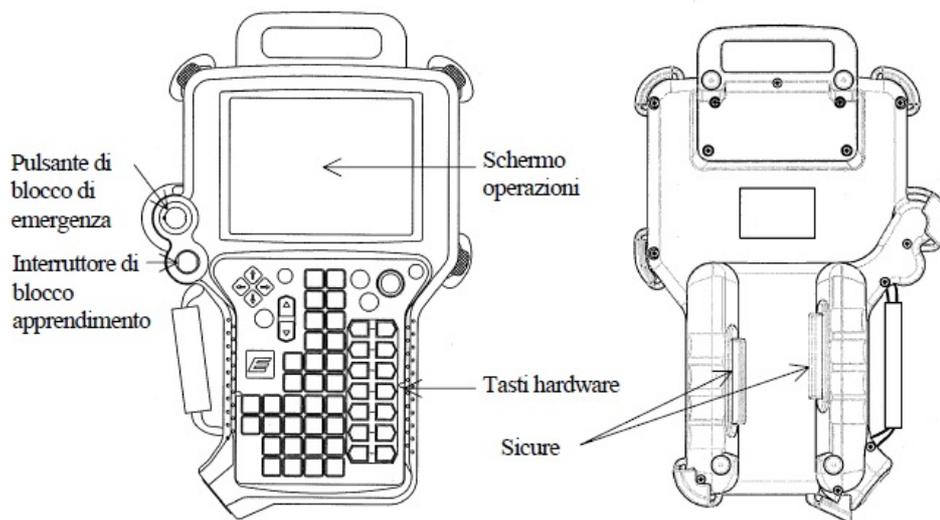


Fig. 6 - Teach Pendant

1.4. Visual Studio Express 2010

L'ambiente di sviluppo utilizzato per l'implementazione del software è Microsoft Visual Studio 2010, che supporta attualmente diversi tipi di linguaggio quali C, C++, C#, F#, Visual Basic .NET e ASP .NET, permettendo la realizzazione di applicazioni, siti e servizi web.

La scelta di Visual Studio Express è dovuta in particolare al completo supporto per il linguaggio C# ed alla ottima gestione dell'aspetto grafico degli applicativi grazie al designer visuale.

Il linguaggio scelto per l'implementazione è il C#, un linguaggio di programmazione object oriented sviluppato da Microsoft all'interno dell'iniziativa .NET; questa scelta è dovuta, oltre che alla potenza computazionale, all'alto grado di integrazione con Visual Studio e con l'SDK ufficiale per Kinect.

2. I problemi riscontrati

In questo paragrafo vengono elencate le difficoltà riscontrate durante il controllo remoto e soprattutto la fase di calcolo di coordinate che collegano il nostro braccio umano, rilevato dal Kinect attraverso gli appositi sensori, con i limiti massimi raggiungibili dal manipolatore fisso Kawasaki.

2.1. Il problema dei giunti

Questo è stato il primo ostacolo incontrato durante lo svolgimento dell'elaborato. L'anatomia del braccio umano e l'anatomia del manipolatore sono assai differenti:

1. Il braccio umano ha sette gradi di libertà (Fig. 7 - :
 - a. Tre gradi di libertà sono forniti dal giunto della spalla
 - b. Due gradi di libertà sono forniti dal giunto del gomito
 - c. Due gradi di libertà sono forniti dal giunto del polso

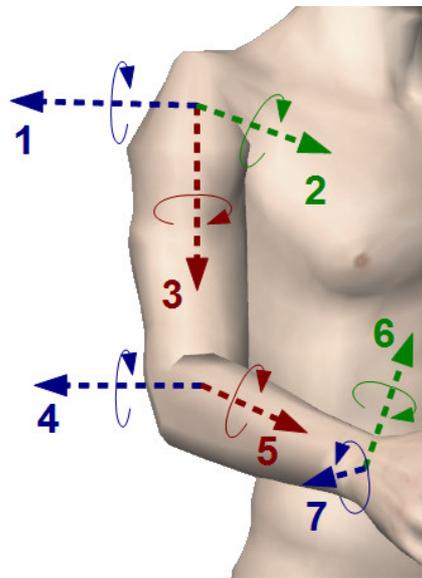


Fig. 7 - Gradi di libertà braccio umano

2. Il manipolatore industriale ha invece sei gradi di libertà, forniti da sei giunti rotatori. Tuttavia con una certa approssimazione è possibile raggruppare questi giunti ottenendo secondo un modello spalla-gomito-polso, ripartendo i gradi di libertà nel seguente modo:
 - a. Due gradi di libertà sono forniti dalla spalla
 - b. Un grado di libertà è fornito dal gomito
 - c. Tre gradi di libertà sono forniti dal polso

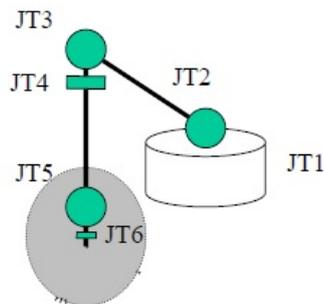


Fig. 8 - Giunti astratti del robot Kawasaki

Il primo problema dunque è trovare il modo di mappare la posizione assunta dai giunti del braccio umano in modo che il manipolatore possa replicare in maniera sufficientemente fedele la posizione assunta dal braccio: infatti anche solo considerando i giunti della spalla e del gomito si nota che il manipolatore non sarà mai in grado di assumere tutte le posizioni assunte dal braccio umano.

2.2. Il problema dell'orientazione

Anche supponendo che il manipolatore sia in grado di assumere tutte le posizioni assunte dal braccio umano, un secondo problema è dato da come il manipolatore è montato nel laboratorio: gli assi di rotazione dei giunti J1, J2 e J3 (relativi a spalla e gomito umano) del manipolatore risultano essere ruotati di -90° rispetto all'asse Y del nostro sistema di riferimento preso direttamente dal Kinect.

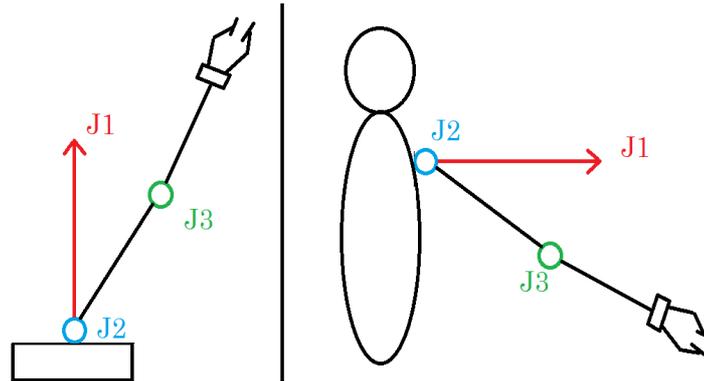


Fig. 9 - Riferimento dei giunti presi in considerazione

Questo problema non ha particolari ripercussioni ai fini dell'implementazione del programma per controllare il manipolatore, si tratta infatti di un problema che rende l'interazione operatore-manipolatore meno intuitiva.

2.3. I limiti del braccio umano ed i limiti del manipolatore

Un ulteriore problema è dato dal fatto che manipolatore e braccio umano hanno limiti di rotazione differenti. Basti pensare al gomito umano ed al gomito del manipolatore: quello umano può muoversi da un angolo di circa 0° (avambraccio e bicipite a contatto) fino ad un angolo di circa 180° (braccio completamente esteso), mentre il manipolatore può muoversi di circa 270°.

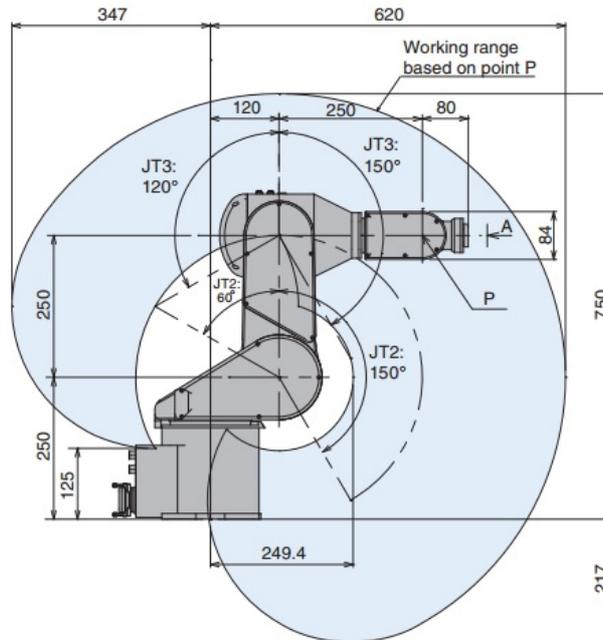


Fig. 10 - Angoli di lavoro del robot Kawasaki

RS003N SPECIFICATIONS			
Type	Articulated		
Degrees of Freedom	6 axes		
Payload	3 kg		
Horizontal Reach	620 mm		
Vertical Reach	967 mm		
Repeatability	±0.02 mm		
Maximum Speed	6,000 mm/s		
Work Envelope (degrees) & Maximum Speed (degrees/s)	Axis	Motion Range	Maximum Speed
	JT1	±160°	360°/s
	JT2	+150° ~ -60°	250°/s
	JT3	+120° ~ -150°	225°/s
	JT4	±360°	540°/s
	JT5	±135°	225°/s
JT6	±360°	540°/s	

Fig. 11 - Tabella delle specifiche robot Kawasaki

Anche questo problema non ha particolare ripercussioni ai fini dell'implementazione, ma sicuramente limita le possibilità dell'operatore.

2.4. Rilevamento della posizione dei giunti del braccio

Mentre i precedenti problemi sono più di carattere teorico, questo è il primo problema di carattere pratico: bisogna infatti misurare gli angoli dei giunti del braccio umano per poi poterli replicare.

2.5. Comunicazione Programma-Manipolatore

Anche questo problema è di carattere pratico: come passare al robot Kawasaki le istruzioni che gli permettono di muoversi?

3. La soluzione adottata

3.1. Il problema dei giunti

Per risolvere questo problema ci siamo concentrati sui primi tre giunti del manipolatore: J1, J2 e J3. Come anticipato prima è possibile assimilare i giunti:

1. J1 e J2 alla spalla umana, come mostrato nelle Fig. 12 - Fig. 14 - . Purtroppo non è stato possibile associare ad alcun giunto il terzo movimento di rotazione della spalla umana

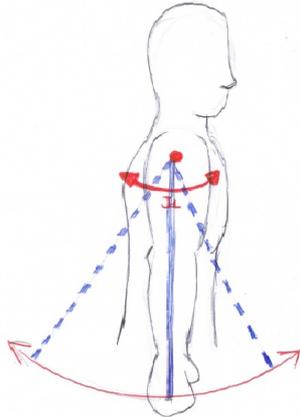


Fig. 12 - Movimento del giunto umano J1

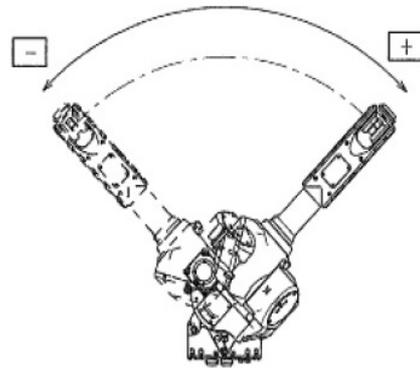


Fig. 13 - J1: Movimento girevole del braccio a sinistra e d destra

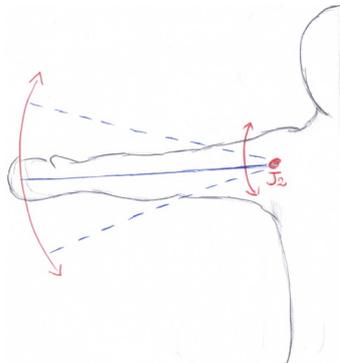


Fig. 14 - Movimento del giunto umano J2

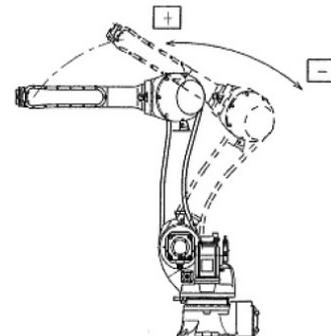


Fig. 15 - J2: Movimento del braccio avanti e indietro

2. J3 al gomito umano, come mostrato in Fig. 16

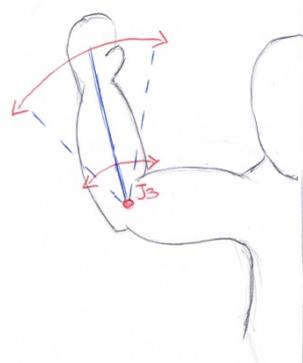


Fig. 16 - Movimento del giunto umano J3

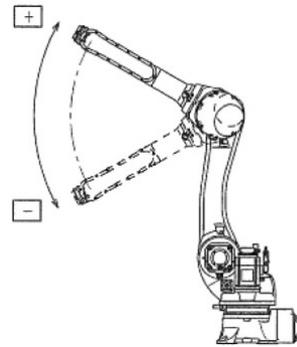


Fig. 17 - J3: Movimento del braccio alto-basso

I giunti J4, J5, e J6 non sono stati presi in considerazione in quanto nella nostra ottica sono associati al polso umano, problematica che non abbiamo potuto affrontare. Infatti gestire il movimento del polso con il Kinect non è stato possibile.

3.2. Il problema dell'orientazione

Questo problema non è stato affrontato, o meglio la soluzione consiste nel chiedere all'operatore una certa capacità di astrazione.

3.3. I limiti del braccio umano ed i limiti del manipolatore

Questo problema è stato risolto empiricamente: una volta realizzato il programma che permetteva di rilevare gli angoli assunti dai giunti del braccio, abbiamo trovato quei valori limite entro cui il braccio umano può arrivare.

3.4. Rilevamento della posizione dei giunti del braccio

Per il calcolo degli angoli assunti dai vari giunti del braccio abbiamo sviluppato un'applicazione C# che utilizza le librerie Kinect (Fig. 19 -) per rilevare la posizione nello spazio cartesiano dei tre elementi fondamentali del braccio (spalla, gomito e polso) per poi ricavarne algebricamente gli angoli.

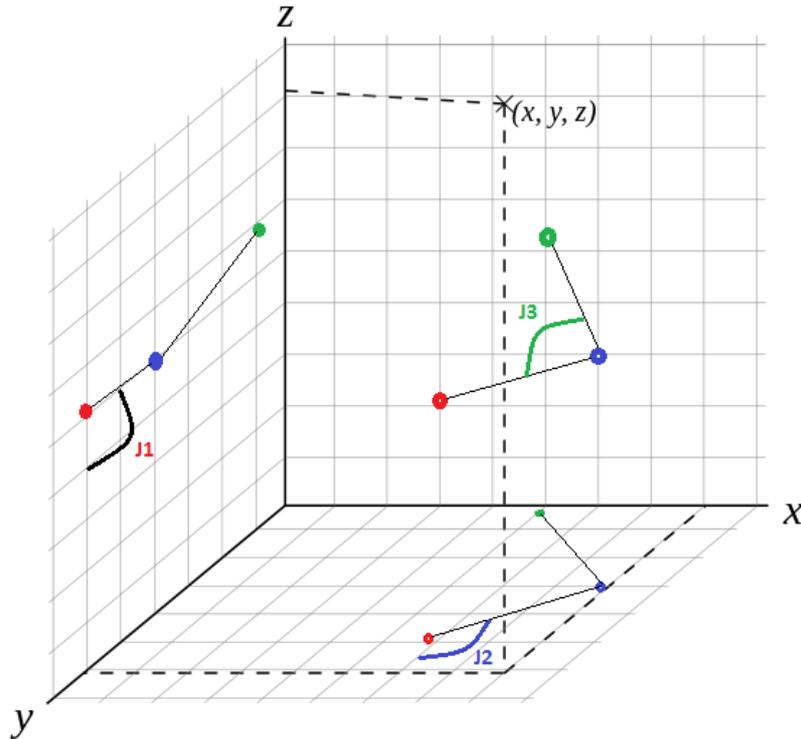


Fig. 18 - Proiezione e calcolo angoli

```
float m1 = 0;
float m2 = 0;

if (elbowP.X - shoulderP.X != 0)
    m1 = (float)(elbowP.Y - shoulderP.Y) / (elbowP.X - shoulderP.X);
else
    m1 = 99999999;
A1 = -(int)((Math.Atan(m1)) * 180 / Math.PI);

m2 = (elbowSP.Y - shoulderSP.Y) / ((float)Math.Sqrt((elbowSP.Z - shoulderSP.Z) * (elbowSP.Z - shoulderSP.Z) +
    (elbowSP.Y - shoulderSP.Y) * (elbowSP.Y - shoulderSP.Y)));
A2 = (int)((Math.Asin(m2)) * 180 / Math.PI);
```

Fig. 19 - Calcolo degli angoli di J1 e J2

Il calcolo di questi due angoli è abbastanza semplice, abbiamo utilizzato un po' di trigonometria per derivare gli angoli partendo dalla posizione nello spazio cartesiano dei punti individuati dal Kinect. La scelta di arcotangente ed arcseno è stata una scelta empirica: infatti abbiamo notato che mentre per l'angolo del giunto J1 l'arcotangente fornisce un angolo più accurato, per quanto riguarda l'angolo di J2 invece l'arcseno è risultato essere migliore.

Siccome, a differenza degli angoli J1 e J2, l'angolo J3 non è calcolato basandosi sulla proiezione del braccio umano su dei piani, non è stato possibile utilizzare una coppia di dimensioni. Abbiamo quindi cercato online una soluzione a questo problema. Dopo vari tentativi siamo arrivati alla conclusione che il codice presentato in Fig. 20 - risolve correttamente questo problema.

```

/* Il calcolo dell'angolo del giunto J3 è più complesso perché non considera la proiezione del braccio su dei piani,
ma calcola l'angolo nello spazio */

public static double AngleBetweenJoints(Joint j1, Joint j2, Joint j3)
{
    double Angulo = 0;
    double shrhX = j1.Position.X - j2.Position.X;
    double shrhY = j1.Position.Y - j2.Position.Y;
    double shrhZ = j1.Position.Z - j2.Position.Z;
    double hsl = vectorNorm(shrhX, shrhY, shrhZ);
    double unrhX = j3.Position.X - j2.Position.X;
    double unrhY = j3.Position.Y - j2.Position.Y;
    double unrhZ = j3.Position.Z - j2.Position.Z;
    double hul = vectorNorm(unrhX, unrhY, unrhZ);
    double mhshu = shrhX * unrhX + shrhY * unrhY + shrhZ * unrhZ;
    double x = mhshu / (hul * hsl);
    if (x != Double.NaN)
    {
        if (-1 <= x && x <= 1)
        {
            double angleRad = Math.Acos(x);
            Angulo = angleRad * (180.0 / Math.PI);
        }
        else
            Angulo = 0;
    }
    else
        Angulo = 0;

    return Angulo;
}

```

Fig. 20 - Calcolo di J3

Tramite l'applicazione sviluppata, si sono misurati gli intervalli entro i quali la Kinect riesce a rilevare correttamente la posizione dei giunti del braccio umano. Nella tabella sottostante, sono riportati questi valori:

	Braccio Umano	Braccio Robotico
J1	[-60°, 30°]	[-160°, 160°]
J2	[-60°, 90°]	[150°, -60°]
J3	[-100°, 0°]	[120°, -150°]

3.5. Comunicazione Programma-Manipolatore

Per capire come comunicano programmi come Krterm con il manipolatore abbiamo registrato una comunicazione tra Krterm con il robot Kawasaki utilizzando Wireshark. Abbiamo scoperto che si tratta di una semplice sessione Telnet, quindi abbiamo ricreato un programma in grado di stabilire una sessione Telnet connettendosi all'indirizzo del manipolatore e di inviare le istruzioni necessarie.

L'unica istruzione utilizzata per far muovere il manipolatore è:

```
DO JMOVE #PPOINT(A1, A2, A3)
```

che ordina al manipolatore di portare i giunti J1, J2 e J3 negli angoli A1, A2 e A3 passati come argomento attraverso un movimento di interpolazione giunti.

Il controllo del manipolatore segue ciclicamente questa serie di eventi:

1. Il Kinect registra le posizioni cartesiane di spalla, gomito e polso
2. Il programma calcola gli angoli A1, A2 e A3
3. SE
 - a. A1, A2 e A3 sono troppo vicini ai valori calcolati nel ciclo precedente (ovvero la differenza in valore assoluto tra A_i e A_{i_old} è inferiore ad una certa soglia per ogni $i=1,2,3$) non viene mandato nessun comando al manipolatore e si torna al punto 1.
 - b. Altrimenti si può proseguire col punto 4.
4. Il programma ordina al manipolatore di interrompere ogni movimento attraverso un ABORT e gli passa le nuove coordinate da raggiungere con il comando:
DO JMOVE #PPOINT(A1, A2, A3)
5. Infine si riparte dal punto 1.

In questo modo (se il movimento del braccio non è rilevate) non vengono inviate troppe istruzioni al manipolatore costringendolo a muoversi in maniera poco fluida.

4. Modalità operative

In questo paragrafo verrà illustrato passo passo come utilizzare il robot articolato Kawasaki RS03N attraverso il dispositivo microsoft Kinect.

4.1. Componenti necessari

Hardware:

1. Il manipolatore Kawasaki RS03N, disponibile nel laboratorio di robotica nell'università
2. Un qualsiasi PC dotato di scheda ethernet e porta USB
3. Un dispositivo Microsoft Kinect

Software:

1. Il programma "KKController.exe" da noi scritto
2. Microsoft Kinect SDK installato sul PC dal quale si vuole operare (scaricabile al link: <http://www.microsoft.com/en-us/download/details.aspx?id=36996>)

4.2. Modalità di installazione

1. Collegare il Kinect al PC
2. Collegare il cavo ethernet collegato al controller del robot Kawasaki al PC
3. Accendere il manipolatore in modalità Repeat
4. Avviare il programma "KKController.exe", allegato alla relazione
5. Il programma è pronto per essere utilizzato, si ricorda che il programma risponde solo al movimento del braccio destro.

5. Conclusioni e sviluppi futuri

Con questo lavoro siamo riusciti ad implementare un primitivo sistema di controllo di un manipolatore attraverso il riconoscimento dei movimenti del braccio dell'operatore. Al momento non si può dire che il lavoro sia concluso, infatti è possibile governare solo tre dei sei giunti del braccio meccanico. Tuttavia crediamo che questo lavoro possa fornire ai colleghi che si cimenteranno in problematiche simili in futuro una base di partenza solida da cui partire. Un possibile sviluppo futuro potrebbe essere governare i restanti tre giunti del manipolatore attraverso l'utilizzo di un Nintendo Wii-mote in modo che l'operatore possa sfruttare a pieno tutte le potenzialità del Kawasaki RS03N.

Bibliografia

- [1] Jana, A. (2012). *Kinect for Windows SDK Programming Guide*. Birmingham - Mumbai: Packt Publishing Ltd.
- [2] Kawasaki Heavy Industries. (2014). *RS03N Robot*. Tratto da www.kawasakirobotics.com: <http://www.kawasakirobotics.com/Robots/RS03N>
- [3] Microsoft. (s.d.). *Get started building Kinect for Windows apps and experiences*. Tratto da www.microsoft.com: <http://www.microsoft.com/en-us/kinectforwindowsdev/Start.aspx>
- [4] Petre, D. (a.a. 2010/2011). Guida introduttiva all'utilizzo del robot Kawasaki RS03N.

Indice

SOMMARIO.....	1
1. INTRODUZIONE.....	1
1.1. Microsoft Kinect	1
1.2. Microsoft SDK	2
1.3. Kawasaki	2
1.3.1. Braccio	3
1.3.2. Controller	4
1.3.3. Teach Pendant	5
1.4. Visual Studio Express 2010	5
2. I PROBLEMI RICONTRATI.....	6
2.1. Il problema dei giunti	6
2.2. Il problema dell'orientazione	7
2.3. I limiti del braccio umano ed i limiti del manipolatore	2
2.4. Rilevamento della posizione dei giunti del braccio	2
2.5. Comunicazione Programma-Manipolatore	2
3. LA SOLUZIONE ADOTTATA.....	3
3.1. Il problema dei giunti	3
3.2. Il problema dell'orientazione	2
3.3. I limiti del braccio umano ed i limiti del manipolatore	2
3.4. Rilevamento della posizione dei giunti del braccio	2
3.5. Comunicazione Programma-Manipolatore	4
4. MODALITÀ OPERATIVE	5
4.1. Componenti necessari	5
4.2. Modalità di installazione	5
5. CONCLUSIONI E SVILUPPI FUTURI	5
BIBLIOGRAFIA.....	5
INDICE.....	6