

“Interfaz de Calibración entre el simulador COSIMIR Educational y el brazo robótico RV-2AJ.”

Wilbert Armando Zapata Jerónimo [eldar_illiande@hotmail.com]

M.C.David Flores Granados

4 de Diciembre del 2010.

Universidad del Caribe, Cancún Quintana Roo, México.

Resumen.

La Universidad del Caribe cuenta con un robot de carácter industrial con modelo Rv-2aj, que puede ser operado por medio de un control de mando o por software desde una computadora, el manejo por software es donde ubicamos la problemática de este proyecto, ya que la programación de rutinas es errónea, esto se debe a que las medidas usadas en el software de las articulaciones difieren de las medidas reales del robot, lo cual lleva a la generación de posiciones fuera del rango del área de trabajo del robot, esto se puede combatir haciendo uso de los métodos que usa el robot internamente para conseguir la configuración ideal, los cuales son la cinemática inversa y directa, al conocer estos procedimientos, la manera en que opera el robot y las medidas reales, tanto como las virtuales erróneas, se construyó una herramienta la cual aplica una corrección y genera posiciones correctas dentro de la burbuja operacional del robot, el resultado de este software es la concordancia entre el simulador y el robot y la posibilidad de crear las rutinas y operar al robot desde una computadora.

1. Planteamiento del Problema

El laboratorio de manufactura y automatización de la Universidad del Caribe cuenta con un brazo robótico de alta precisión modelo RV-2AJ de la empresa FESTO, el cual cuenta con 5 ejes de libertad, una pinza de presión para poder sostener y operar objetos; además es parte de un conjunto de módulos operacionales los cuales tienen como fin al trabajar en conjunto poder brindar una estación completa de automatización de una línea producción.

El robot está inmerso en una burbuja que conforma su espacio real de trabajo que se puede representar como un sistema de coordenadas. El brazo puede ser operado de 2 maneras: una por medio de un control que permite programar los movimientos y operaciones a realizar, la segunda forma es por medio de descarga de archivos desde una computadora que contiene el conjunto de instrucciones a ser ejecutadas, cada archivo está compuesto por las 3 coordenadas donde se va a posicionar el brazo así como la orientación lo cual le da una posición real en su espacio de trabajo.

El brazo robótico industrial sigue la analogía de un brazo humano tiene articulaciones que le permiten quedar en una posición específica, el cual puede ser utilizado para diversas tareas, como son reposicionamiento de ciertas piezas, muestreo de ciertos químicos entre otras muchas actividades en diferentes campos de la industria

La posición real de cada articulación del robot está descrita por los ángulos que marcan su desplazamiento en base a una posición origen, electrónicamente es calculado por un circuito llamado Encoder. El conjunto de posicionamientos de todas las articulaciones componen la posición final del robot que describe una ubicación en su espacio de trabajo bajo los 3 ejes euclidianos que maneja (X, Y, Z). En robótica existen 2 métodos llamados Cinemática Directa y Cinemática Inversa para la obtención de la posición final del robot en base a los ángulos de cada articulación y para el cálculo de los ángulos de las articulaciones en base a una posición final respectivamente.

El software COSIMIR es una herramienta de simulación y modelado para robots industriales de Mitsubishi, dentro de su contenido podemos hacer uso de la simulación de un gran número de modelos diferentes de robots de operación industrial, con la facilidad de poder crear rutinas completas, modelar y simular en su totalidad espacios de trabajo, líneas de producción etc. El software contiene dentro de sus librerías las cinemáticas directas e inversas de cada modelo a simular, siendo los mismos métodos que usa internamente cada robot para hacer sus cálculos.

COSIMIR te permite crear las rutinas y generar los archivos resultantes de estas simulaciones, permitiendo descargar dichos programas en el robot y poder observar con seguridad que el robot fue programado correctamente. Aquí es donde ubicamos la problemática que este proyecto trata de combatir, para poder llevar a cabo los cálculos de cinemática directa e inversa COSIMIR cuenta con las medidas exactas de las articulaciones de cada modelo, para que todos los cálculos sean correctos. Las medidas del robot con el que cuenta la Universidad difieren con las que COSIMIR calcula los movimientos, esto imposibilita el uso de dicha herramienta de simulado para la creación de rutinas, ya que al ser medidas diferentes la posición en la que se tiene que colocar el brazo para ubicarse en una coordenada no es la misma en el mundo real y en el espacio simulado, por consiguiente los grados que se debe mover cada articulación difieren considerablemente e incluso llegan a salirse de la burbuja de trabajo sobre la cual actúa el robot.

2. Justificación.

El modelado/simulado es una herramienta muy poderosa y nos brinda una gran utilidad en el ámbito educativo, ha de notarse que al contar con un solo robot de éste estilo en una Universidad donde se encuentran 2 carreras que pueden hacer uso de él en sus asignaturas nos lleva a tener que dividir mucho las ventanas de tiempo en las que cada alumno, profesor o becario puede operar con el robot. Lo cual es poco eficiente si se considera que se puede tener una población de hasta 30 alumnos para una sola asignatura de robótica por carrera, dejar practicas donde los alumnos se tengan que involucrar con el robot se vuelve un problema con los tiempos tanto para los alumnos como al momento de calificar para un profesor. Un simulador funcional es la respuesta obvia para este problema, ya que lo que nos brinda una herramienta de éste estilo es un espacio libre de riesgos por ser una recreación de las condiciones reales de trabajo del robot, esto permite crear rutinas completas de trabajo que podrían darse por correctas y solo esperar a descargarlas en el robot y verlo operar, de ahí la importancia en una institución educativa de poder contar con este tipo de herramientas.

COSIMIR es el simulador hecho para esta tarea pero no es funcional específicamente en la problemática que quiere combatir este documento por los detalles ya mencionados, al corregirse ese error entonces se podría tener el simulador 100% funcional que opere con el robot.

Las diferencias entre el espacio virtual que maneja el simulador y el espacio real de trabajo del robot imposibilitan en su totalidad la creación de las ya mencionadas rutinas operacionales, debido a que el posicionamiento de las articulaciones del robot se ve afectado produciendo posiciones diferentes además de colocándolo en ubicaciones que no son contempladas en el simulador, otro caso es que se traten de alcanzar coordenadas que el robot no tienen dentro de su burbuja de acción terminando toda actividad de manera terminal como producto de la protección.

3. Propuesta.

La propuesta del proyecto es crear una interfaz invisible para el usuario final que corrija estas diferencias espaciales entre el simulador y el robot, esto proporcionaría un simulador útil al 100% para la programación controlada de carácter académico necesaria para las materias de robótica de la Universidad del Caribe, dicha interfaz hará uso de los conocimientos de Cinemática y Cinemática Inversa así como geometría Euclidiana tridimensional aplicada a la robótica para poder reproducir el movimiento del simulador en una escala correcta para nuestro Brazo robótico de precisión.

El software tendrá como datos de entrada 2 posibles opciones, la coordenada con sus 3 componentes (X, Y, Z) o los ángulos de cada articulación, datos necesarios para llevar a cabo ya sea la Cinemática Inversa o la Cinemática Directa, el programa devolverá un archivo con todas las coordenadas adaptadas para sus medidas y así poder tener una rutina que coincida al 100% con el simulador.

A lo largo de este documento se desarrollaran las 2 aproximaciones la cinemática directa y la cinemática inversa y la manera en que fueron adaptadas en el lenguaje C++ para poder resolver la problemática.

3.1 Cinemática Directa.

La cinemática directa es una técnica que es usada para obtener el posicionamiento de cada una de las partes de una estructura articulada, tomando en cuenta un sistema de coordenadas base, de igual manera se tiene que las articulaciones están encadenadas entre ellas, o en el último de los casos de la base.

Se debe de tomar en consideración que en nuestro caso buscamos analizar primero la cinemática directa del Robot RV-2AJ y después reproducir el procedimiento para generar una posición correcta en el espacio de trabajo real basándonos en ángulos que nos brinda el software de simulación COSIMIR. Teniendo claro entonces que nuestros datos de entrada son los ángulos que marcan el desplazamiento de cada articulación la variable de salida será una única coordenada sobre los 3 ejes euclidianos por los cuales puede marcar su desplazamiento el robot.

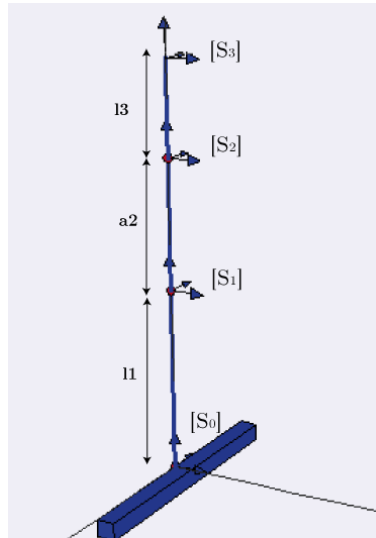


Figura 1: Posicionamiento en el Origen.

La figura 1 nos muestra una representación de la ubicación y la relación directa que hay entre el posicionamiento de cada una de las articulaciones, siendo S0 la Base, S1 Shoulder, S2 Elbow, S3 Pitch.

El desplazamiento de una articulación lo podemos entender como una rotación dentro de uno de los ejes que componen la coordenada que describe su posición. En la geometría euclidiana la rotación de un punto se describe como la multiplicación de sus componentes por la matriz rotacional respectiva, esto dependiendo sobre que eje se requiere su rotación, en caso de que se rote en más de un eje a la vez simplemente se hace una operación de multiplicación de matrices.

A continuación se describen las matrices rotacionales para los ejes de nuestro espacio Euclidiano tridimensional:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\text{Ang}) & -\sin(\text{Ang}) \\ 0 & \sin(\text{Ang}) & \cos(\text{Ang}) \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos(\text{Ang}) & 0 & \sin(\text{Ang}) \\ 0 & 1 & 0 \\ -\sin(\text{Ang}) & 0 & \cos(\text{Ang}) \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos(\text{Ang}) & -\sin(\text{Ang}) & 0 \\ \sin(\text{Ang}) & \cos(\text{Ang}) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Siendo Ang el valor en grados que se quiere rotar.

Por consiguiente teniendo una coordenada base $P = [X, Y, Z]$ podemos hacer una multiplicación con alguna de las matrices anteriores dependiendo de en que eje quiera uno rotar y obtenemos una coordenada resultante rotada Ang grados Por ejemplo:

$P = [0 \ 0 \ 905]$ y lo queremos rotar 90 grados sobre el eje de las Ys, entonces tenemos lo siguiente:

$$\text{RotatedP} = P * R_y = [0 \ 0 \ 905] * R_y = [\cos(90) \ 0 \ \sin(90); 0 \ 1 \ 0; -\sin(90) \ 0 \ \cos(90)] = [905 \ 0 \ 0].$$

$$\text{RotatedP} = [905 \ 0 \ 0].$$

Si se quiere rotar en ejes diferentes por ejemplo el Eje Y y El eje X la operación quedaría descrita de la siguiente manera:

$$\text{RotatedP} = (P * R_y) * R_x.$$

Conociendo las operaciones necesarias para poder ubicar la posición final del robot, tenemos que comprender que un brazo articulado tiene cierto número de articulaciones, este número indica que tantos grados de libertad tiene para moverse y alcanzar la posición deseada, en otras palabras para nuestro caso particular tenemos lo siguiente $\text{PosicionFinal} = \text{PosShoulder} + \text{PosElbow} + \text{PosPitch}$. De lo cual podemos sacar una conclusión si obtenemos por separado la rotación de cada articulación con respecto a su coordenada origen y las sumamos podemos obtener la posición final, pero para poder llevar a cabo ese conjunto de operaciones se necesita como dato de entrada la medida de cada articulación, para que entonces se pueda crear una coordenada o trama que describa a 1 y solo 1 articulación en particular, rotarla Ang Grados y sumar este resultado a los obtenidos de aplicar esa misma operación al resto de las articulaciones.

Para poder obtener las medidas de cada articulación se llevó a cabo el siguiente procedimiento:

Obtenemos la siguiente coordenada = $[X=0, Y=0, Z=905]$ esto ubica al brazo extendido todo hacia el eje de las Z hacia arriba.

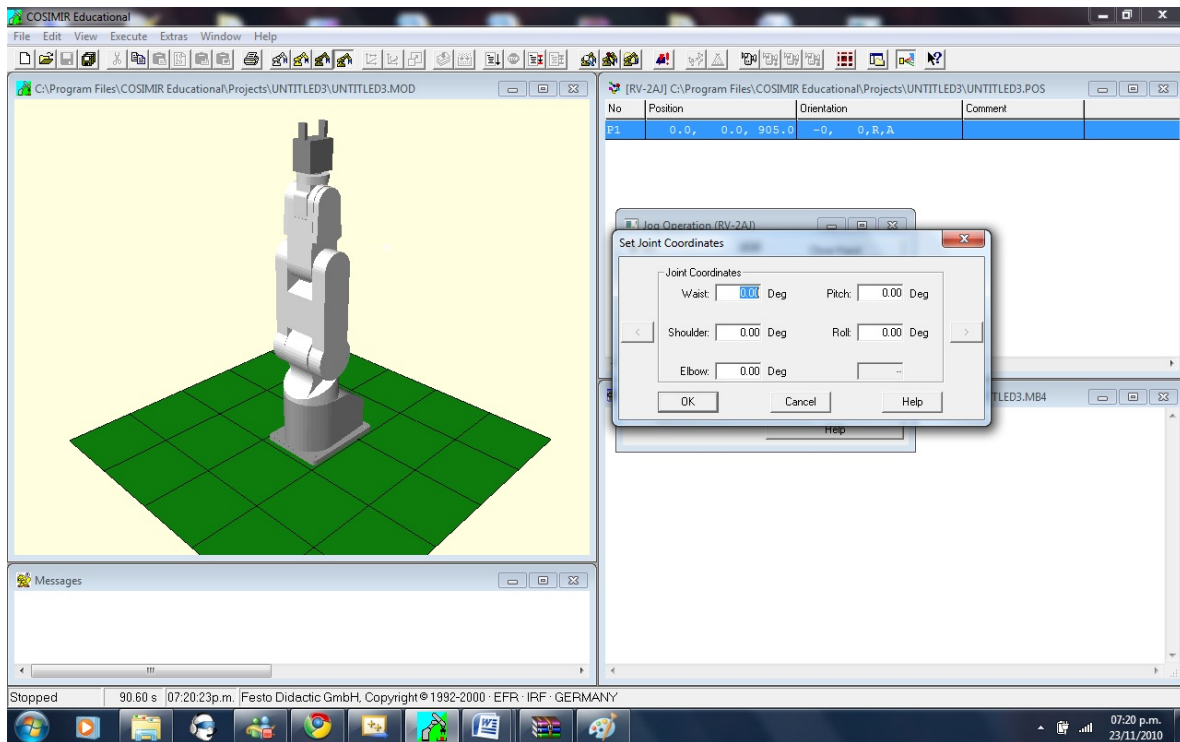


Figura 2: Posición Origen.

Se puede observar en la figura 2 que el largo total de las articulaciones + la base era 905.

Si se coloca Shoulder = 90, el brazo robótico se inclina completamente para verse Extendido en el eje de las X y se obtiene la siguiente coordenada:

X = 605

Y = 0

Z = 300

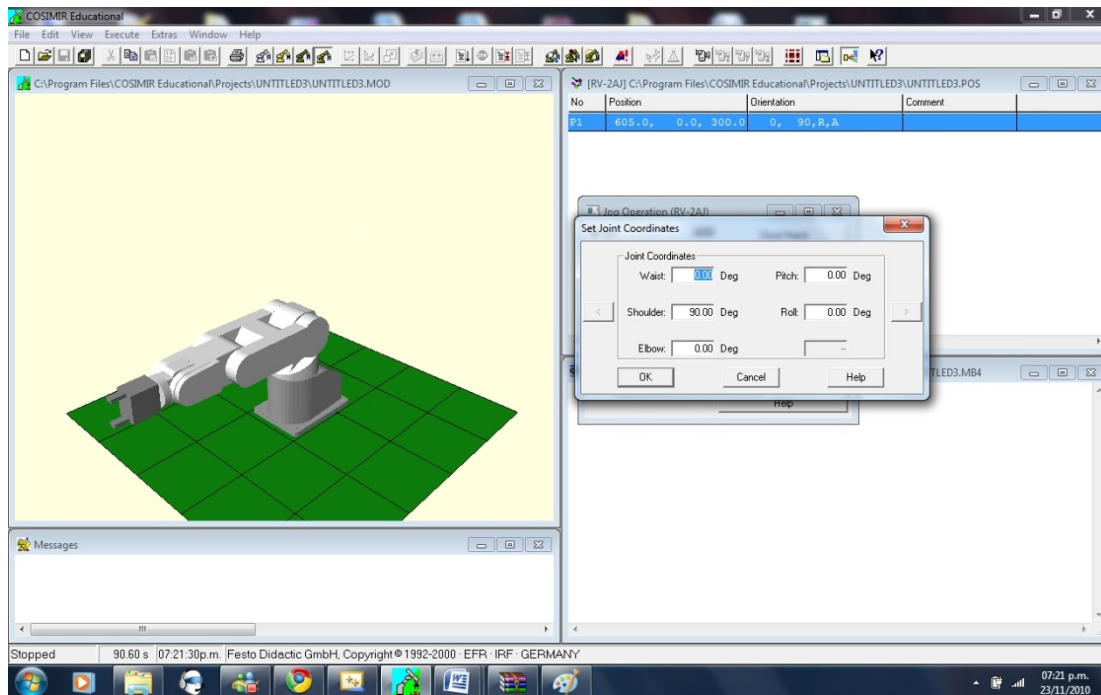


Figura 3.

Así se obtiene el valor de la base, de ahí se ubico el ángulo del Pitch = -90 lo cual arrojo la siguiente coordenada:

X= 410

Y = 0

Z = 495

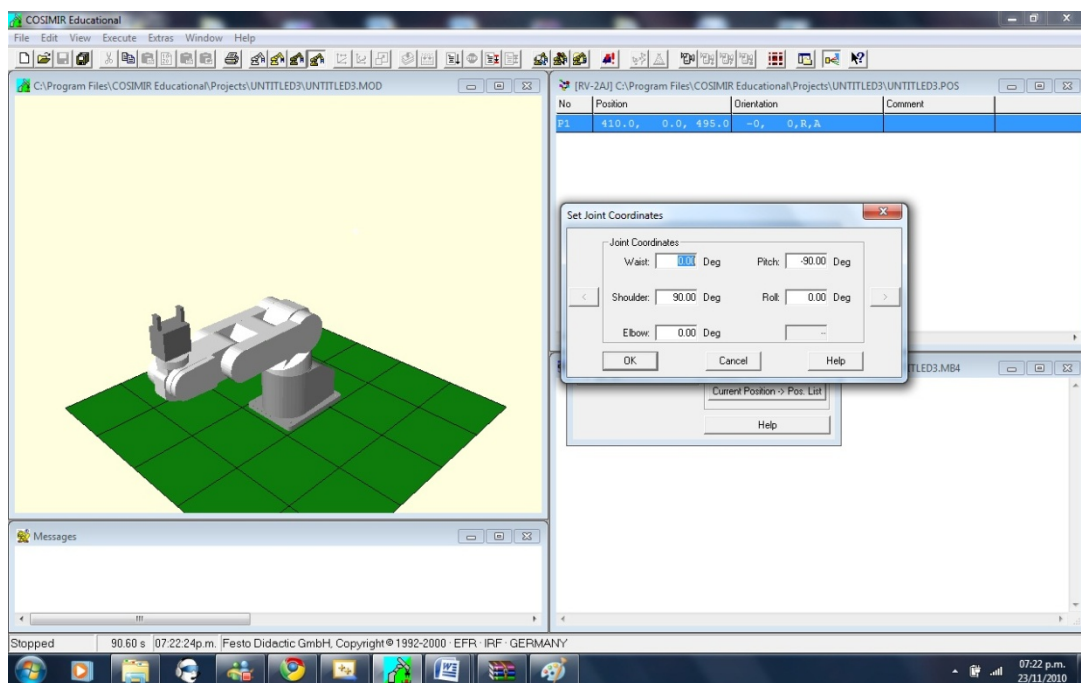


Figura 4.

Como se observa que la base incremento en +195, y X decremento en -195, por lo cual se sobre entiende que la medida del Pitch = 195.

Se coloco de nuevo el valor del Pitch = 0 y ahora coloque el Elbow = -90 lo cual arrojo la siguiente coordenada:

X = 250

Y = 0

Z = 655

Con lo que se concluye que el Valor del Shoulder = 250, y Elbow = 160.

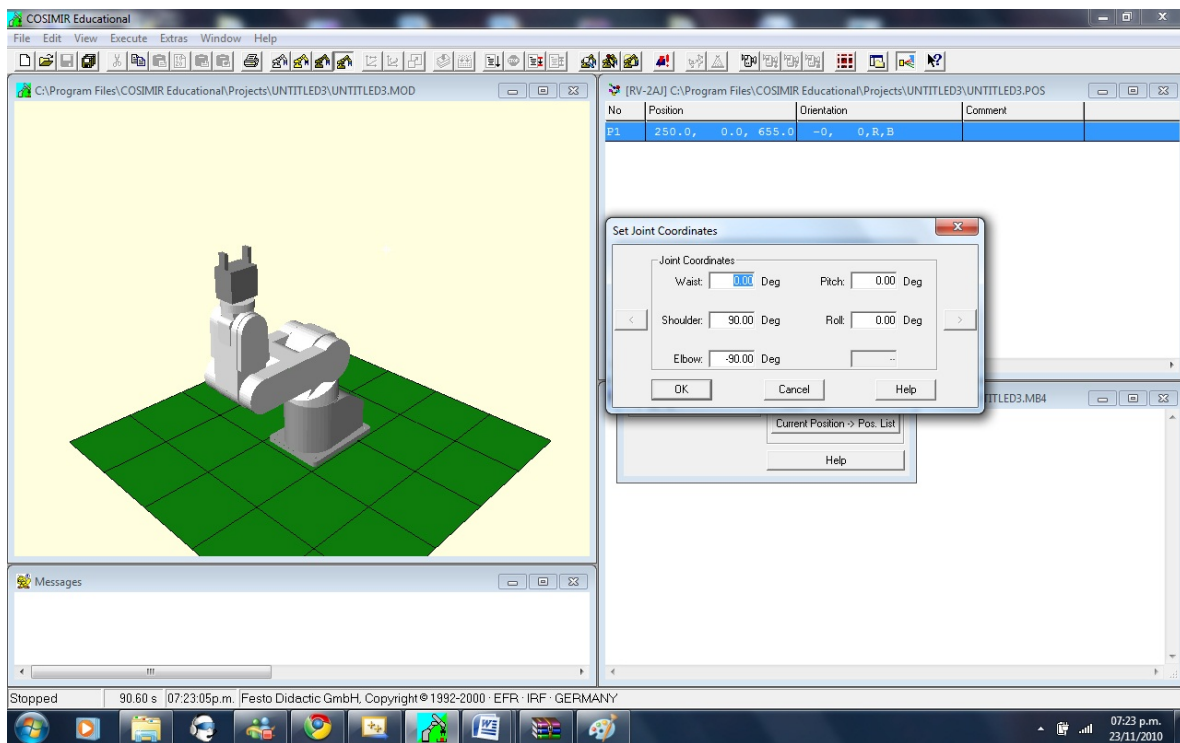


Figura 5.

Con las medidas de cada articulación se puede entonces crear una coordenada que describa a cada articulación y así poder rotarla cuanto sea necesario, una vez hecho esto para todas sumarlas y obtener la posición final del brazo. Las coordenadas descriptivas quedarían de la siguiente manera:

Pitch = [0 0 195]

Elbow = [0 0 160]

Shoulder = [0 0 250]

Entonces se pueden rotar esas coordenadas Ang grados, sumarmas y obtenemos la posición final, ahora bien algo que se debe tomar en cuenta que se menciono al inicio del reporte , es que cada articulación depende de la art. Inferior a ella, por lo consiguiente sus ejes X, Y y Z dependen de la posición de la articulación inferior. Teniendo en cuenta que el Shoulder depende de la base, Elbow depende del Shoulder y el Pitch a si mismo del Shoulder, por ejemplo; el shoulder se desplaza 30' grados, el Elbow 15' grados y el Pitch 25' grados, el shoulder al depender de la base, la cual no puede rotar siempre tendrá su rotación descrita de la siguiente manera, $AngShoulder = AngShoulder + AngBase$, dado que base es igual a 0 tenemos que:

$$AngShoulder = AngShoulder$$

$$AngElbow = AngElbow + AngShoulder$$

$$AngPitch = AngPitch + AngElbow$$

Si sustituimos por los valores tendremos que:

$$AngShoulder = 30$$

$$AngElbow = 15 + 30 = 45$$

$$AngPitch = 25 + 45 = 70$$

Y esos son los grados que reciben las matrices de rotación, al multiplicarse por su coordenada al origen correspondiente tenemos el desplazamiento real de cada articulación habiendo respetado el desplazamiento de la inferior.

Hay otras maneras de resolver la cinemática directa, por medio de ecuaciones de senos y cosenos, pero eran aproximaciones complejas, este es un enfoque sencillo y funcional que fue probado con Matlab, COSIMIR y una hoja de cálculo en Excel durante la fase de pruebas, cuenta con un margen de error de .1 en el peor de los casos, lo cual es completamente despreciable ya que no se posiciona el brazo fuera de la ubicación esperada al menos no que el ojo humano pueda apreciar o afectar la rutina.

3.2 Cinemática Inversa.

La cinemática inversa es el procedimiento contrario a la cinemática directa, mediante este método nosotros podemos obtener los ángulos que marcan el desplazamiento de cada articulación que conforma nuestro cuerpo para ubicarlo en la posición final de la pinza, la cual es nuestro único parámetro de entrada.

Una de las principales características que diferencia a este método es que no existe una solución única o una formula general que nos pueda solucionar la problemática, cada robot tiene sus propias ecuaciones fuertemente ligadas a sus configuraciones, además que pueden existir N configuraciones diferentes que pueden tomarse como opciones validas para alcanzar la coordenada deseada. El enfoque utilizado durante este proyecto fue tratar de buscar una solución geométrica para poder obtener la configuración angular de las articulaciones para poder llegar a la coordenada final de la muñeca que fue nuestro parámetro de entrada.

Se uso un acercamiento geométrico y se describe a continuación:

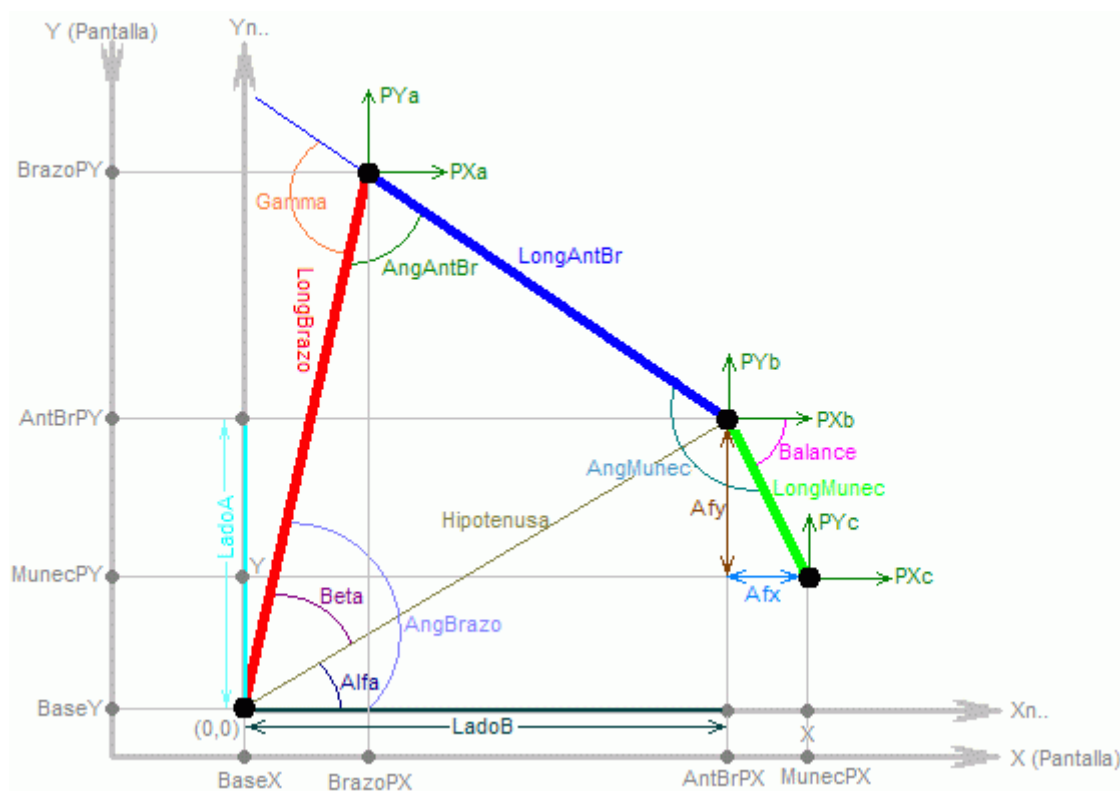


Figura 6.

La idea principal es trazar triángulos, debido a que por las propiedades de los triángulos nosotros podemos obtener longitudes, y los ángulos que lo conforman, se usan los únicos parámetros de entrada (los cuales son las coordenadas en los 3 ejes) para ir obteniendo poco a poco los ángulos que describen el desplazamiento de cada articulación. La serie de pasos necesarios vienen descritos en pseudocódigo:

```
/* Medidas del Robot */  
float Base = 300;  
float Shoulder = 250;  
float Elbow = 160;  
float Pitch = 195;  
/* Medidas del Robot */
```

Las medidas del robot son extremadamente necesarias, esto es para poder hacer los cálculos de longitudes necesarios.

```
/* Posición Inicial del robot*/  
float X = 250;  
float Y = 0;  
float Z = 550;  
float Roll = 0; // es el Balance, pero COSIMIR no te brinda este dato y es  
extremadamente necesario  
/* Posición Inicial del robot*/
```

Nuestros únicos parámetros de entrada son la posición final de la muñeca.

```
/* Variables auxiliares */  
float Pi = atan(1.0)*4; // esto es lo mismo que 3.1416  
float Rad = Pi/180;  
float Grad = 180/Pi;  
/* Variables auxiliares */
```

```
/* Calculo de la Cinematica Inversa */  
float AngGiro = (atan2(Y,X))*Grad;  
//float AngGiro = atan(Y/X)*Grad;
```

El Angulo de Giro (J1) es el más sencillo de calcular ya que al usar la propiedad geométrica del Arco tangente de el eje Y con respecto al Eje X podemos obtener el desplazamiento de la herramienta, Grad como podemos observar es una variable que usamos para convertir un resultado de Radianes a grados, por eso se hace esa multiplicación.

```
float Modulo = sqrt(abs(X^2)+abs(Y^2));  
float XPrima = Modulo;  
float YPrima = Z;
```

El siguiente paso es calcular el modulo, este dato nos sirve para poder obtener la Xprima y Yprima, las cuales son los elementos básicos para poder encontrar los valores de los catetos A y B que vemos en la figura2.

```

float Afx = cos(Rad*Roll)*Pitch;
float LadoB = XPrima - Afx;
float Afy = sin(Rad*Roll)*Pitch;
float LadoA = YPrima - Afy - Base;

```

El cálculo del LadoA y LadoB queda descrito como la diferencia entre las X y Y primas de las variables Afx y Afy.

```

float Hipotenusa = sqrt((LadoA^2)+(LadoB^2));

```

Lo siguiente es calcular la hipotenusa que une al LadoA con el LadoB, valor que nos será útil inmediatamente para poder obtener los 2 ángulos que conforman nuestro Angulo del Brazo (AngShoulder o J2).

```

float Alfa = atan2(LadoA,LadoB);
float Beta = acos(((Shoulder^2)-
(Elbow^2)+(Hipotenusa^2))/(2*Shoulder*Hipotenusa));

```

Alfa esta descrito como el Arco Tangente de LadoA respecto a LadoB y Beta queda descrito como el arco coseno de la diferencia de cuadrados del Brazo y Antebrazo, mas el cuadrado de la Hipotenusa entre 2 veces la longitud del Brazo Por la Hipotenusa.

```

float AngShoulder = (Alfa+Beta)*Grad;

```

El Angulo del Brazo como se menciono previamente es la suma de Alfa y Beta, el resultado de esta adición es en radianes por lo consiguiente se necesita multiplicar por nuestra variable Grad para transformarlo a Grados.

```

float Gamma = acos(((Shoulder^2)+(Elbow^2)-
(Hipotenusa^2))/(2*Shoulder*Elbow));

```

El siguiente paso es calcular el valor de Gamma, valor que nos es útil para poder obtener el desplazamiento del Ante brazo, y queda descrito como el arco coseno de la suma de cuadrados de la longitud del brazo y ante brazo menos el cuadrado de la hipotenusa entre 2 veces la longitud del Brazo por la longitud del Ante brazo.

```

float AngElbow = -((180*Rad)-Gamma)*Grad;

```

El Angulo del antebrazo queda descrito como la diferencia entre Pi y Gamma por menos 1, multiplicado por Grad para obtener el valor en grados;

```

float AngPitch = Roll-AngShoulder-AngElbow;

```

El ángulo de la Muñeca se obtiene fácilmente restando el valor de los 2 ángulos obtenidos previamente del valor que te da COSIMIR que indica la suma total de los 3 ángulos.

4. Resultados.

Una vez terminados los algoritmos se procedió a probar su efectividad; tenemos como datos de entrada las medidas de las articulaciones en el simulador y en el robot:

Simulador.	
Medidas del Robot.	
Pitch	195
Elbow	160
shoulder	250
base	300

Tabla 1.

Robot Físico	
Medidas del Robot.	
Pitch	72
Elbow	160
shoulder	250
base	300

Tabla 2.

De igual manera nosotros contamos con la coordenada que describe la posición final de la pinza de nuestro robot, en base a esta posición nosotros obtenemos la configuración ideal. COSIMIR nos brinda el simulado del posicionamiento que adopta el robot para alcanzar nuestra coordenada destino, se toma esa información y se utiliza el procedimiento de cinemática inversa para obtener los ángulos:

Test 1:

Coordenada y cálculo de ángulos:

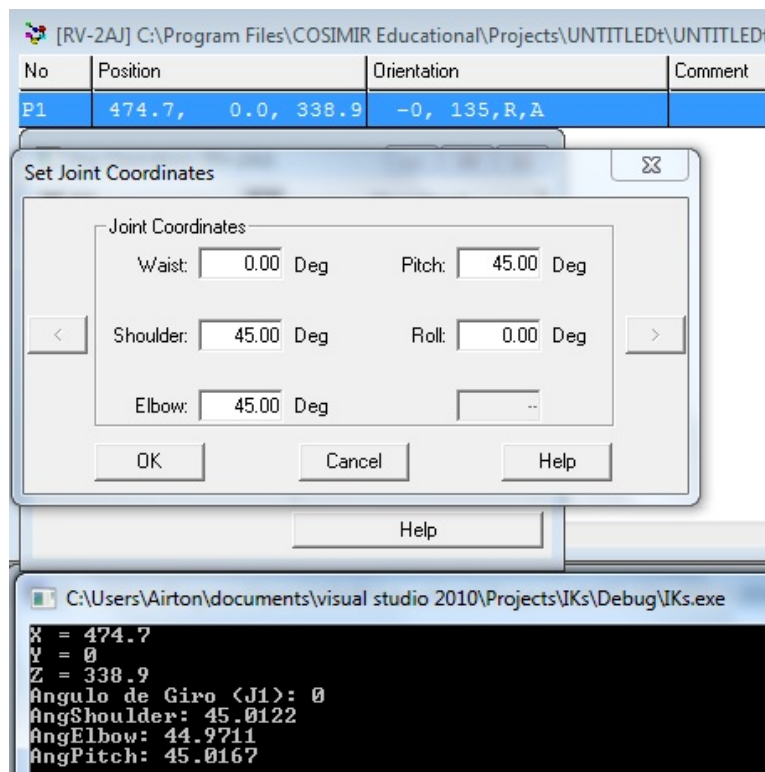


Figura 7.

En la parte superior de la figura 1 encontramos la información que nos brinda COSIMIR, en la parte inferior en la pantalla de fondo negro encontramos los datos que nos arroja el Script de Cinemática Inversa.

Con los ángulos obtenidos se hace el cálculo de la cinemática Directa, calculando la rotación de cada una de las articulaciones con las medidas reales del robot, la suma de todas estas rotaciones nos brinda la posición preliminar a la cual se le ha de sumar cualquier constante de movimiento, en caso de que haya sido desplazado de la posición origen sobre uno o más de sus ejes, siendo el caso de las últimas pruebas el robot fue utilizado por alumnos y por consiguiente quedo desplazado 50 unidades en el eje de las Xs y 100 en el eje de las Ys, teniendo en cuenta esto, estas cantidades son adicionadas a las posiciones preliminares y así se calcula el posicionamiento correcto del robot.

Cinemática Directa:

	45'	45'	45'				
Eje	Shoulder	Elbow	Pitch	Total	Base	Subtotal	Total
X	176.7767	160	50.9117	387.6884		387.6884	437.6884
Y	0	0	0	0		0	100
Z	176.776	0	-50.9117	125.8643	300	425.8643	425.8643

Tabla 3.

Posicionamiento del robot:

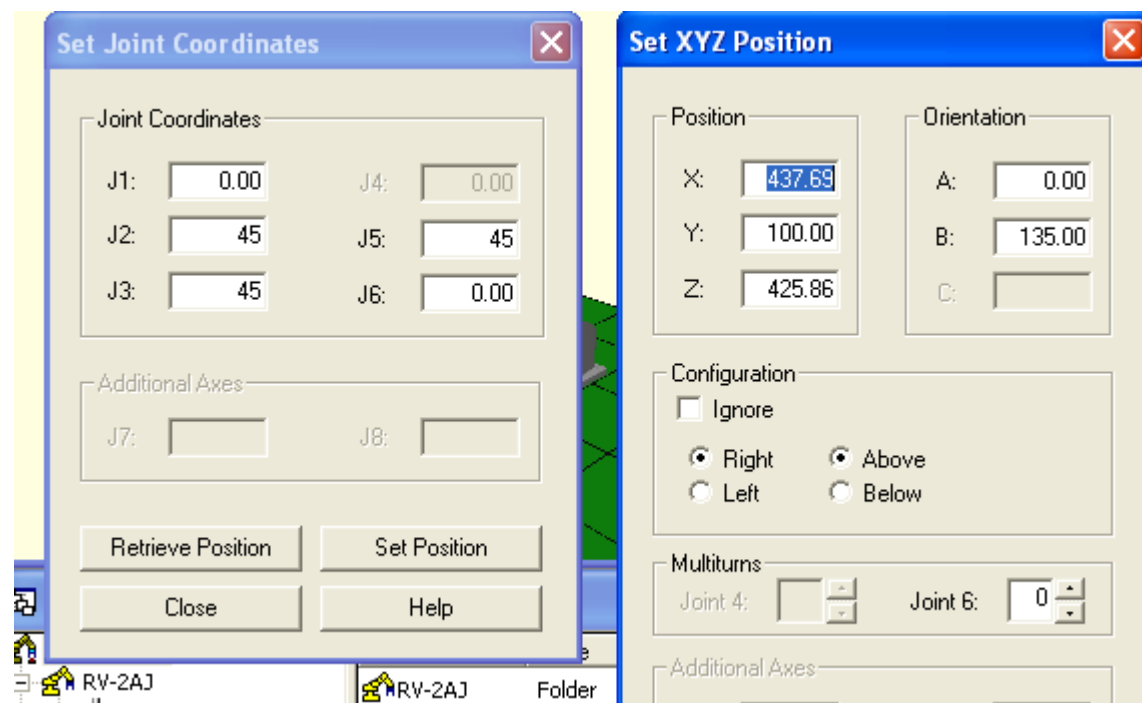


Figura 8.

Posicionamiento del simulador:

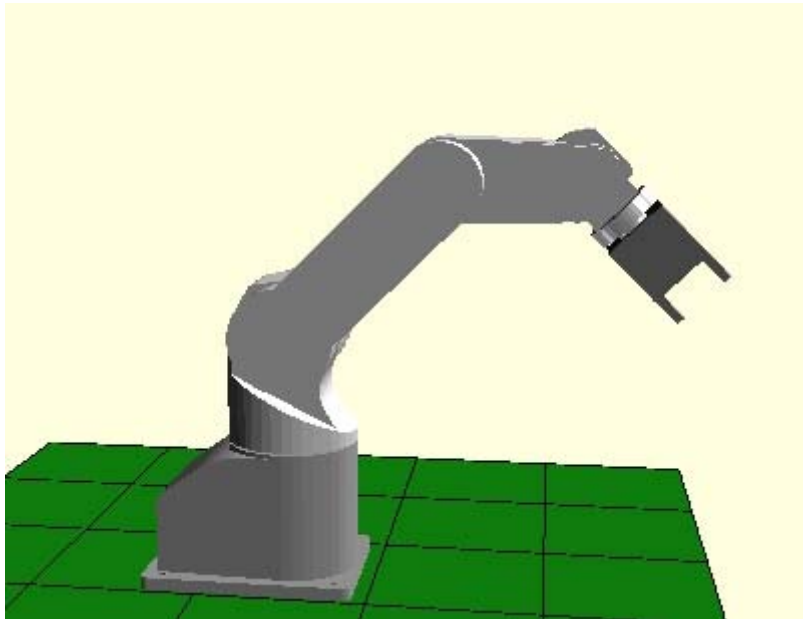


Figura 9.

Test 2:

Coordenada y cálculo de ángulos:

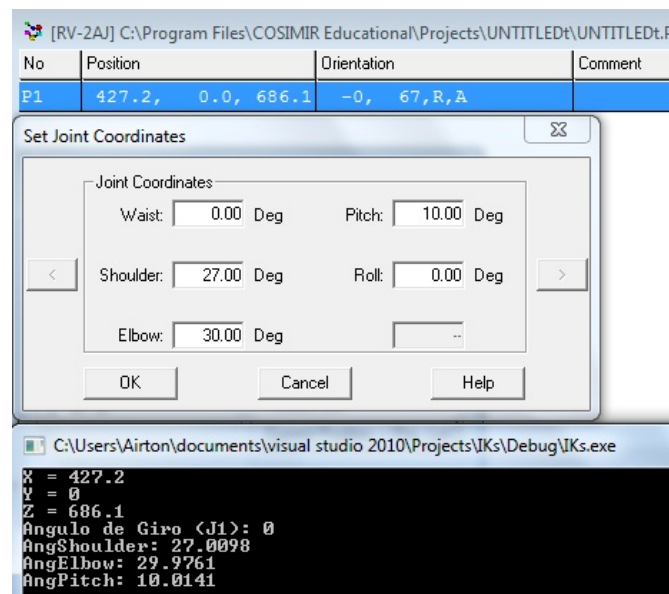


Figura 10.

Cinemática Directa:

	27	30	10				
Eje	Shoulder	Elbow	Pitch	Total	Base	Subtotal	Total
X	113.4976	134.1873	66.2763	313.9612		313.9612	363.9612
Y	0	0	0	0		0	100
Z	222.7516	87.1422	28.1326	338.0264	300	638.0264	638.0264

Tabla 4.

Posicionamiento del robot:

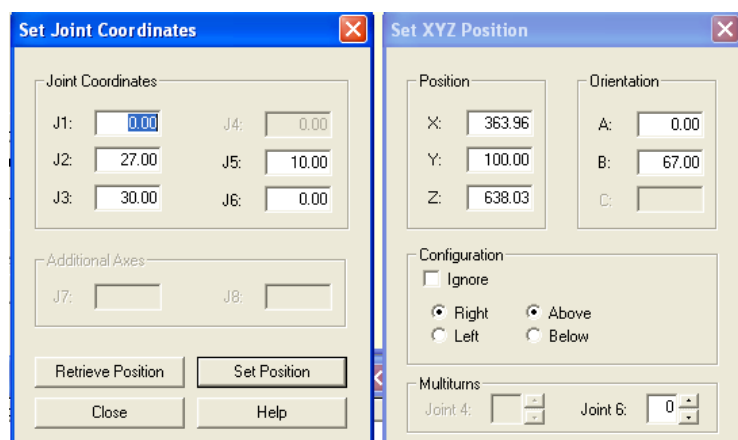


Figura 11.

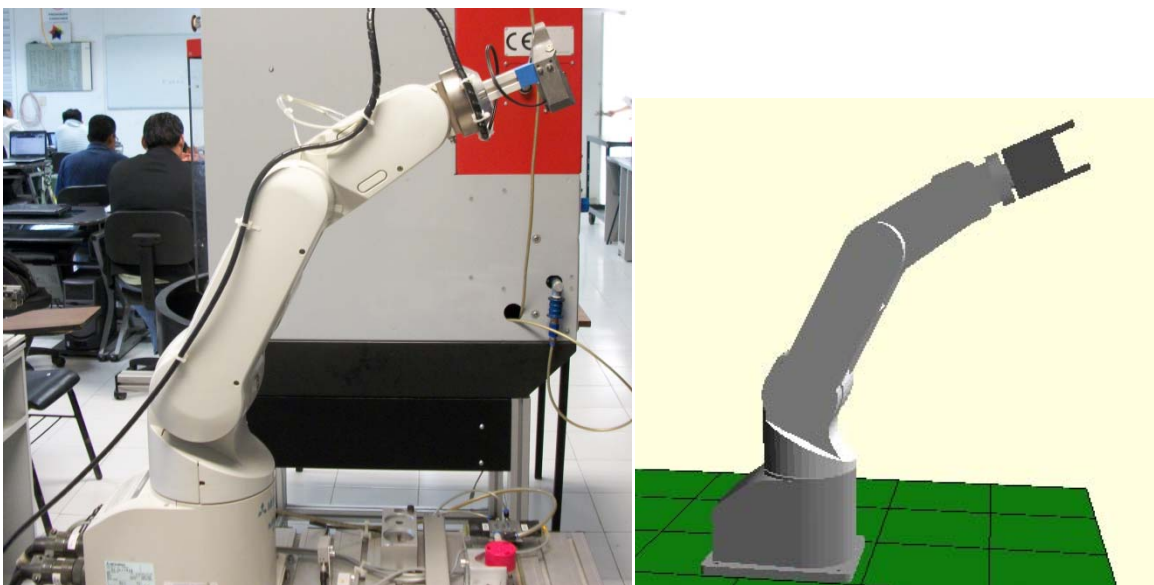


Figura 12.

La figura 12 nos muestra la comparativa del posicionamiento virtual y real.

Test 3:

Coordenada y cálculo de ángulos:

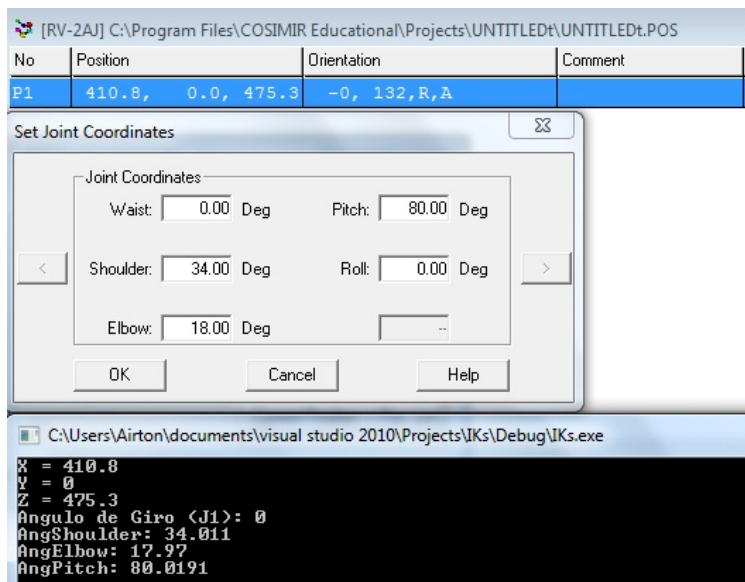


Figura 13.

Cinemática Directa:

	34	18	80				
Eje	Shoulder	Elbow	Pitch	Total	Base	Subtotal	Total
X	139.7982	126.0817	53.5064	319.3863		319.3863	369.3863
Y	0	0	0	0		0	100
Z	207.2594	98.5058	-48.1774	257.5878	300	557.5878	557.5878

Tabla 5.

Posicionamiento del robot:

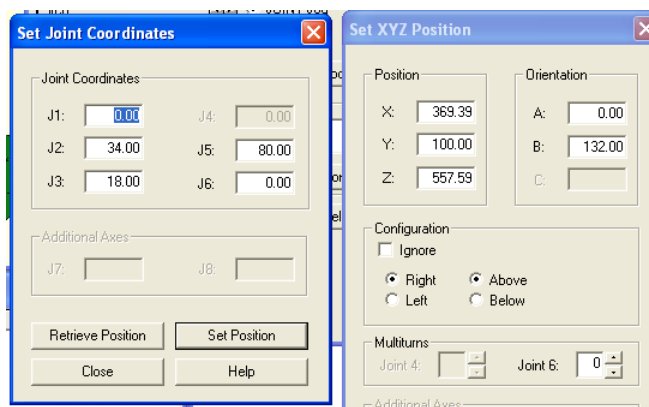


Figura 14.

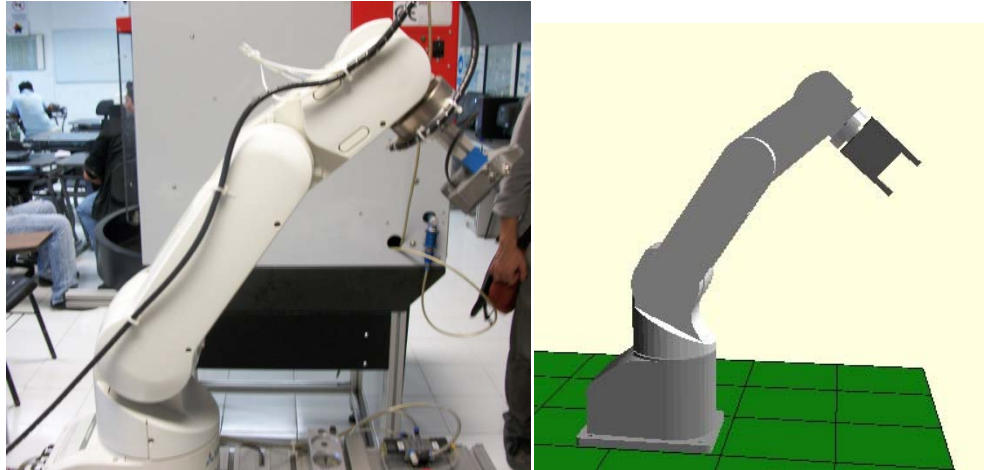


Figura 15.

5. Conclusiones.

La diferencia entre las medidas de las articulaciones que maneja el simulador COSIMIR, y las medidas del robot real de la Universidad del Caribe, causa un problema de discordancia espacial entre la herramienta de simulado/programación y el brazo de precisión. Dicho problema puede ser combatido hasta casi nulificar el porcentaje de error con los métodos de cinemática inversa y directa, estos procedimientos son los que usa internamente para hacer sus cálculos y llevar al dispositivo a la configuración ideal para alcanzar el objetivo, con los scripts descritos en este documento usando como base la información arrojada por el simulador se generan los datos necesarios para que el robot se aproxime lo más posible a la ubicación deseada.

Los resultados nos arroja información que nos permite estimar que tanto se pudo disminuir este porcentaje de error, durante la primera fase se produce la cinemática inversa, en base a una coordenada final calculamos los ángulos que pertenecen al desplazamiento de cada articulación, los ángulos obtenidos mediante el script "InverseKinematics" no están redondeados hacia arriba mostrando solamente los primeros 2 decimales lo cual produce que estos sean diferentes a los mostrados por COSIMIR esto nos lleva a decir que el script tiene un 99% de efectividad, una vez aplicado este método los ángulos serán idénticos, la fase 2 es el cálculo de la cinemática directa la cual si presenta un margen de error decimal muy pequeño que no tiene directamente que ver con algún tipo de redondeo por lo cual este script tiene un porcentaje de efectividad de 98% el cual ya no puede incrementarse mas, de igual manera cabe mencionar que este porcentaje no disminuye incluso si la rutina tiene 100 movimientos, ya que el error no se arrastra y no se va sumando hasta llegar a un punto donde el resultado sea totalmente erróneo, esto es porque cada posición es calculada desde la posición inicial del robot, y no desde su ultima ubicación, por lo cual el porcentaje de efectividad de toda la interfaz es de 98%.

El uso de los scripts permite crear posiciones correctas para que el robot se pueda mover y alcanzar las configuraciones ideales para ubicarse de manera idéntica a como el simulador nos

indica, lo cual permite el uso de esta herramienta para una programación correcta y sin problemas del Rv-2aj.

6. Referencias.

- [1] "INSTRUCTION MANUAL, Detailed explanations of functions and operations", Mitsubishi Electric Corporation, Japan, 2001
- [2] "INSTRUCTION MANUAL, Troubleshooting", Mitsubishi Electric Corporation, Japan, 2001
- [3] "GETTING STARTED", Mitsubishi Electric Corporation, www.festo.com/didactic/, Germany, November 2000.
- [4] "Mitsubishi RV-2AJ Industrial Robot Programming", The University of Plymouth, <http://www.swrtec.de/swrtec/industrial/Mitsubishi%20LabNotes%20v0.6.pdf>, November 2005, Version 0.6.
- [5] "Alphabetical Overview of Melfa-Basic-IV commands", <http://dmi.uib.es/~jguerrero/instMelfa.pdf>, November 2004.
- [6] "Programación Melfa Iv", Ruben Loredó, http://www.slideshare.net/ruben_loredo/programacion-melfa-iv, January 2010.
- [7] <http://sites.google.com/site/proyectosroboticos/cinematica-inversaii>
- [8] http://www.aurova.ua.es/robolab/EJS2/RRR_Intro_3.html
- [9] http://freespace.virgin.net/hugo.elias/models/m_ik.htm
- [10] http://freespace.virgin.net/hugo.elias/models/m_ik2.htm