

IMPLEMENTACIÓN DE ROBOT SCORBOT ER-VPLUS, PARA DOCENCIA, EN REALIDAD VIRTUAL

Juan Carlos Parra Márquez y Karina Pilar Cid Cifuentes
Académicos de la Universidad del Bío-Bío
jparra@ubiobio.cl
kcid@alumnos.ubiobio.cl

IMPLEMENTACIÓN DE ROBOT SCORBOT ER-VPLUS, PARA DOCENCIA, EN REALIDAD VIRTUAL

Resumen

La simulación de maquinarias, en un entorno 3D, es cada vez más común en la formación de ingenieros y en la formación, o capacitación, en la industria, y sus principales ventajas son: no depreciación de la máquina simulada, reducción de costos de aprendizaje, resguardo físico del usuario, entre otras. En particular, en nuestra Universidad del Bío-Bío, se cuenta con laboratorios de formación en robótica pero no se posee todas las unidades robotizadas (principalmente, Scorbott ER-Vplus) necesarias para el alumnado y, por otra parte, el costo de mal uso, por inexperiencia inicial de las mismas, es muy alto. La simulación del funcionamiento del robot, como alternativa, en un entorno de Realidad Virtual crea el ambiente propicio para que los alumnos aprendan a interactuar correctamente con la botonera de aprendizaje sin dañar la estructura real cuando intentan llegar a posiciones físicas inalcanzables. Finalmente, para el desarrollo de la aplicación se fusionaron conceptos claves relacionados con Robótica y Realidad Virtual, tales como: resolución de ecuaciones para la cinemática y trayectoria del robot, diseño y construcción del sistema bajo el software 3D Webmaster y programación en el lenguaje SCL, entre otros.

Palabras clave: Realidad Virtual, Robótica, Enseñanza, Informática.

Developmet of robot Scorbott ER-Vplus, for learning, in Virtual Reality

Abstract

3D machinery simulation is a common engineering activity both at pre-grade and training at the work place in the industry. Its main advantages are the non-depreciation of the simulated machinery, lower learning costs, and user physical safeguards among others. At Universidad del Bío-Bío there is a laboratory for Robotics Control but without the necessary robotised units (mainly Scorbott ERVplus) for the students to handle, whose cost increases because of misuse on their part. The alternative solution of robot simulation, in a virtual reality environment, allows a proper situation for the interaction between students and robotics learning without damaging the real physical structure. Finally, for the development of the application, key concepts such as Equations Resolutions for Cinematic and Robot Trajectory, Software design and Implementation under Programming in 3D Webmaster and Programming in SCL language, related to Robotics and Virtual Reality, have been blend.

Key words: Virtual Reality, Robotic, Teaching, Informatics.

Introducción

El Laboratorio de Sistemas Automatizados de Producción (Figura 1) de la Universidad del Bío-Bío, CIMUBB [1], está constituido por máquinas CNC, sistemas de control de calidad, estaciones de trabajo CAD/CAM/CAE y de simulación. Estas tecnologías se integran para permitir la automatización de un proceso de producción completo, desde su diseño hasta la fabricación misma del producto. Como líneas de investigación de dicho Laboratorio encontramos: Procesamiento digital de señales e imágenes (Automatización de la producción, CAD/CAM/CAE y Robótica).



Actualmente este laboratorio cuenta con dos las células de trabajo, cada una posee un robot SCORBOT-ER Vplus programable y manipulado por alumnos que realizan actividades, o proyectos, que apoyan su proceso de enseñanza. Durante el proceso de aprendizaje, en la utilización del robot, muchas veces se instruye al robot para moverse a lugares donde no tiene alcance, forzando el trabajo de sus motores al máximo o colisionando su estructura con algún objeto; este problema desgasta la estructura física del robot aumentando la depreciación de la máquina. Otro aspecto es la alta demanda de los estudiantes, situación que se complica cuando el laboratorio es ocupado como aula de clases y principalmente cuando varios alumnos deben desarrollar tareas con el robot, o en un proyecto de investigación.

Finalmente, por lo anteriormente descrito, el objetivo central fue el diseñar e implementar un sistema de simulación en 3D, para el robot SCORBOT-ER Vplus, que apoyará la labor formativa y orientará a los alumnos que trabajan con el robot en el Laboratorio CIMUBB, mediante la ejecución de los comandos propios del lenguaje reconocidos por el controlador real del robot. Para ello, se ha diseñado el robot, con el software de realidad virtual 3D Webmaster, con el cual se ha programado la generación de trayectorias, base de los movimientos del robot, e interactuando con su botonera virtual.

Conceptos

En este proyecto, es clave tener presente ciertos conceptos básicos; estos son: realidad virtual, robot, cinemática, entre otros.

Realidad Virtual (Figura 2): Consiste en "La simulación de medios ambientes y de los mecanismos sensoriales del hombre por computadora, de tal manera que se busca proporcionar al usuario la sensación de inmersión y la capacidad de interacción con medios ambientes artificiales"¹.



¹ Definición de RV obtenida en http://www.network-press.org/?realidad_virtual

En la realidad virtual se identifican tres conceptos bases: simulación, interacción y percepción [7], estas son las condiciones necesarias que se deben cumplir para que el sistema que estamos modelando pueda ser clasificado como aplicación de realidad virtual.

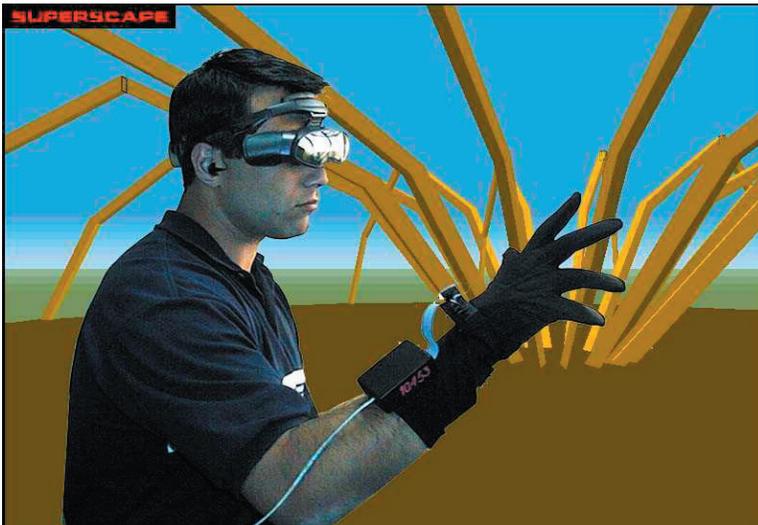


Figura 2. Realidad Virtual.

Robot: Es un dispositivo compuesto por sensores y actuadores [2]. Por medio de los sensores se reciben datos de entrada que, dependiendo de la inteligencia del robot, son analizados para luego efectuar la acción programada por medio de dichos actuadores. En particular, un robot manipulador es un mecanismo compuesto generalmente por elementos en serie, articulados o deslizantes, uno con respecto al otro, cuyo objetivo es recoger y desplazar objetos a determinado lugar, de acuerdo con los grados de libertad que posea el robot.

Una característica relevante de un robot es la morfología (Figura 3), es decir, su estructura constituida de articulaciones y elementos o eslabones. Los elementos son las partes, que unidas por medio de articulaciones y que terminan en una pieza final llamado *efector o muñeca*, son diseñadas para manipular objetos. Por cada articulación puede existir más de un movimiento independiente, llamado *Grado de libertad*.

En particular, el robot Scorbot ER Vplus [5] está compuesto por un brazo robot, una botonera y un controlador (Figura 4). Usualmente es parte de una célula de trabajo y lo podemos ver interactuando con otras células por medio de un PC, con la finalidad de realizar tareas integradas de manufactura. El PC se comunica con el controlador del robot, ya sea para enviar instrucciones o recibir comandos de control por medio de alguna interfaz de programación. Con la botonera de enseñanza se practican los primeros movimientos del robot y se envían instrucciones básicas al controlador; éstos constituyen los primeros pasos que deben dar los estudiantes para entender y puedan interactuar de manera segura con el robot.

Los movimientos que se generan a partir de la estructura son detallados en Tabla1.

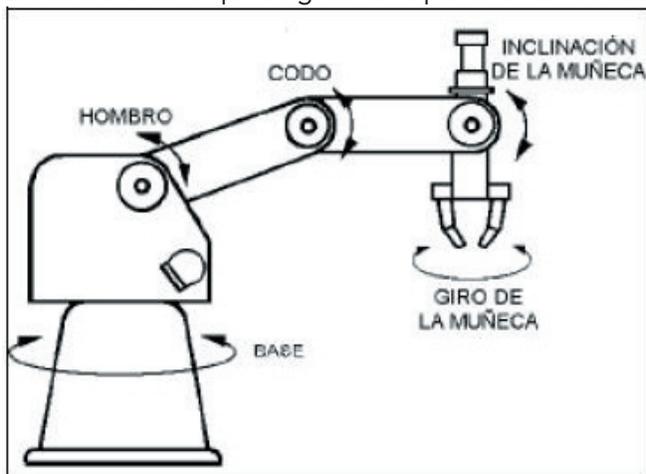


Figura 3. Estructura de un robot.

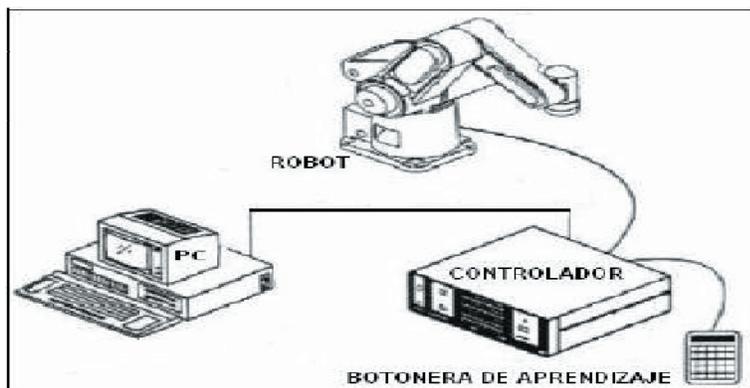


Figura 4. Célula de trabajo.

Nº EJE	NOMBRE	MOVIM.	RANGOS
1	Base	Rotación del cuerpo	0°/310°
2	Hombro	Sube y baja el brazo	+130°/-35°
3	Codo	Sube y baja el antebrazo	±130°
4	Elevación	Sube y baja la inclinación de la pinza	±130°
5	Giro	Gira la herramienta final (pinza)	Sin límite mecánico, eléctricamente ±570°.

Este robot puede ser programado y operado por medio del lenguaje de programación ACL [4] que es reconocido por el controlador del robot, interfase ATS para el usuario, y por medio de la botonera; estos medios de programación se usan tanto en forma independiente como en combinación. Esta botonera, posee una pantalla de cristal líquido (LCD) que permite visualizar 2 líneas y 32 caracteres. La pantalla muestra el estado del controlador, la orden del usuario y mensajes del sistema de control.

Cinemática: La cinemática [6] es el estudio del movimiento. La estructura de los robots manipuladores se mueve con el objetivo de que su efector logre una posición deseada; para ello, cada articulación alcanza una posición angular (coordenadas angulares o ángulos de rotación de cada eje con respecto a la articulación anterior) de modo que el efector alcance las coordenadas x,y,z en el espacio. En particular, la denominada cinemática directa calcula las coordenadas x,y,z a partir de las coordenadas angulares ($q_0,q_1,..q_n$) ya conocidas para cada articulación. Por otra parte, la cinemática inversa consiste en obtener las coordenadas angulares ($q_0,q_1,..q_n$), de todas las articulaciones, conociendo solamente la posición x,y,z del efector en el espacio.

Trayectoria: Es el camino que debe recorrer el robot para llegar de un punto a otro, si hablamos de la trayectoria (Figura 5) que sigue el efector a través de una línea recta, una curva o cualquier otra figura geométrica. En general, la trayectoria se presenta en coordenadas cartesianas (x,y,z).

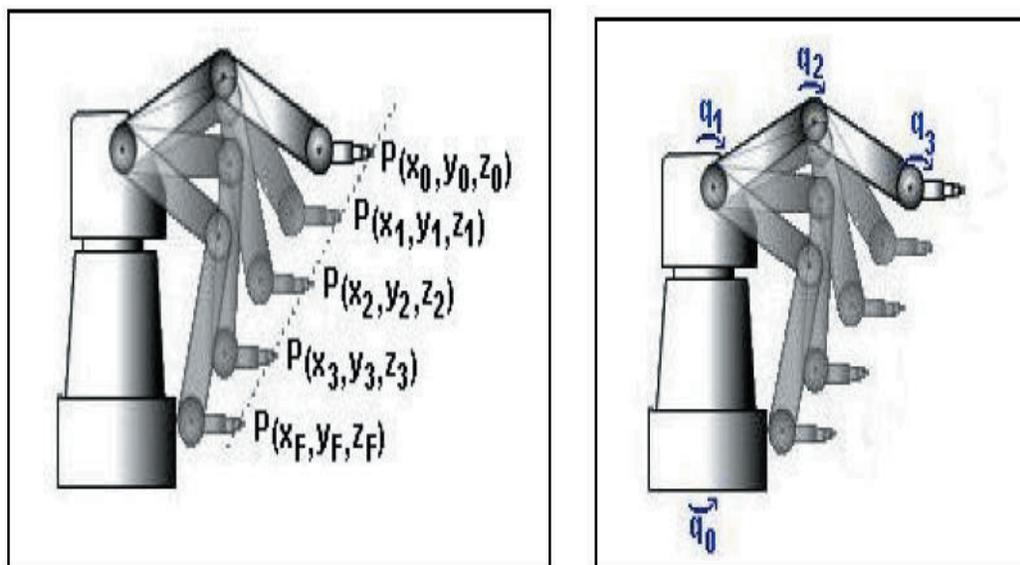


Figura 5. Trayectorias cartesiana y polar.

Interpolación: En cada trayectoria, es necesario definir una cantidad de puntos intermedios por los que debe pasar el efector. Al método que genera estos puntos se le llama interpolación. En el caso de la trayectoria lineal hay que definir los puntos x, y, z sobre la línea, dividiendo el trayecto completo en tramos. En el caso de la trayectoria angular, aunque todos los ángulos evolucionan de forma independiente, por lo general, todos comienzan y terminan al mismo tiempo por lo que el trayecto completo también se puede dividir en tramos, pero en este caso los ángulos de cada articulación varían en distintas magnitudes. Debemos hacer notar que para la trayectoria lineal es necesario utilizar la cinemática inversa, puesto que para cada punto x, y, z de la interpolación lineal es necesario calcular las posiciones angulares que debe alcanzar cada articulación. En general, el robot Scorbot ER Vplus trabaja en dos sistemas de coordenadas: en el sistema de coordenadas de ejes, JOINTS, y en el sistema de coordenadas cartesianas XYZ. *El sistema de coordenadas cartesianas* es un sistema geométrico, usado para representar la posición x, y, z del punto central de la pinza por medio de la definición de distancia desde el punto de origen, que se encuentra en el centro de la base, hacia los ejes a lo largo de los tres ejes coordenados. Cuando se ejecuta un movimiento en modo Joints, los ejes se mueven individualmente.

Fundamentos matemáticos: Las ecuaciones obtenidas por la cinemática directa y la cinemática inversa [3] deben ser coherentes con el modelo virtual diseñado en 3D. Cuando el robot mueve su estructura, en modo JOINTS, está constantemente utilizando cinemática directa; es decir, a partir de los ángulos ya conocidos incrementa los valores de sus articulaciones q_0, q_1, q_2, q_3 (Figura 7) para mover el efector (o pinza). Si el robot se mueve en modo XYZ primero se realizan los cálculos de cinemática directa para conocer el punto (x_p, y_p, z_p) actual de la pinza, luego se incrementa uno o todos los valores del punto (x_p, y_p, z_p) para mover la pinza en línea recta, y con el nuevo punto se calcula, por medio de cinemática inversa, los valores q_0, q_1, q_2, q_3 que corresponden a ese punto; finalmente, sólo queda asignar los nuevos q_0, q_1, q_2, q_3 a las articulaciones para llegar al punto esperado. La cinemática directa calcula la posición (x_m, y_m, z_m) de la muñeca con los valores angulares q_0, q_1, q_2 ; luego se calcula q_3' y, conociendo la posición de la muñeca, se puede calcular el punto (x_p, y_p, z_p) que define en coordenadas XYZ la punta de la pinza.

Las expresiones matemáticas necesarias, para obtener (x_p, y_p, z_p) , en función de q_0, q_1, q_2, q_3 y q_3' , son:

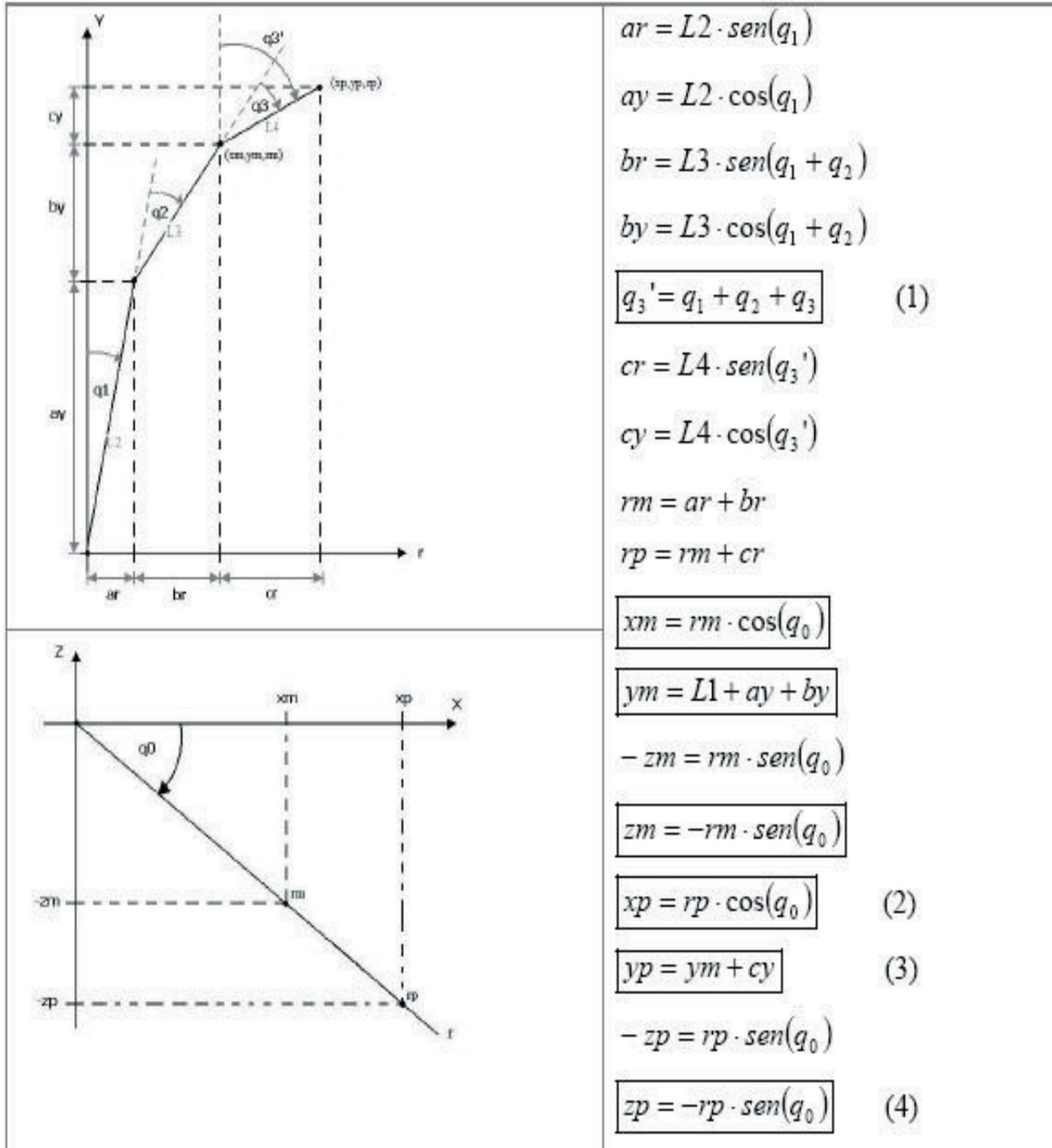


Figura 7. Cinemática directa.

Para la cinemática, tanto en la realidad como en la simulación, se conoce en todo momento los ángulos de apertura de cada articulación, por lo que se podría pensar que no se necesita calcular cinemática inversa; pero cuando se debe generar una trayectoria lineal sólo se conocen las coordenadas angulares q_0, q_1, q_2, q_3 (Figura 8) del punto actual (inicial) y del punto final, así que es necesario calcular por cinemática directa el punto x_p, y_p, z_p inicial y final e interpolarlos (al igual que para el ángulo q_3'). Para estos puntos interpolados es necesario calcular, por cinemática inversa, las posiciones angulares que debe tomar cada articulación para asegurarse que la punta de la pinza pase por los puntos interpolados sobre la recta. En la cinemática inversa la pinza se considera por separado del resto de las articulaciones, ya que gira sobre su propio eje

creando el ángulo q_3' . Así tendremos solamente el punto (x_p, y_p, z_p) y q_3' como datos conocidos para calcular todas las articulaciones. Inicialmente se toma el ángulo q_3' para calcular el punto de la muñeca (x_m, y_m, z_m) , y con este punto y q_3' es posible calcular por cinemática inversa (q_0, q_1, q_2, q_3) .

En función de los datos ya conocidos x_p, y_p, z_p y q_3' obtenemos:

	$q_3' = q_1 + q_2 + q_3$ $y_m = y_p - L4 \cdot \cos(q_3')$ $drp = L4 \cdot \sin(q_3')$ $r_m = r_p - drp$
	$q_0 = \text{Arc tan} \left(\frac{-z_p}{x_p} \right)$ $-z_m = -z_p - drp \cdot \sin(q_0)$ $z_m = z_p + drp \cdot \sin(q_0)$ $x_m = x_p - drp \cdot \cos(q_0)$ $r_p = \sqrt{(x_p)^2 + (z_p)^2}$

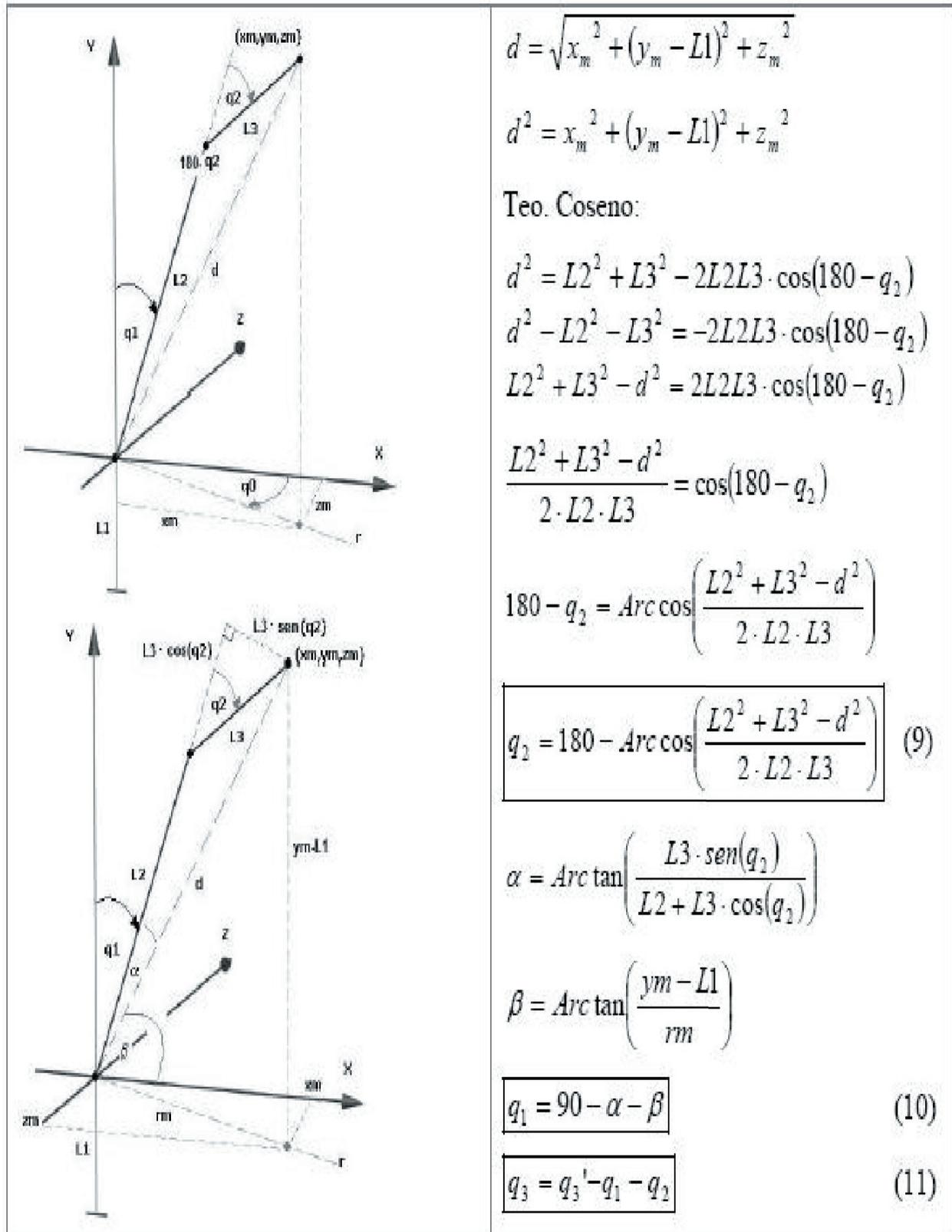


Figura 8 Cinemática Inversa.

Desarrollo

El sistema a modelar: El robot SCORBOT ER Vplus fue diseñado en un ambiente 3D incluyendo: brazo robot, botonera de enseñanza y controlador. El robot se opera por medio de su botonera de enseñanza, por lo que se implementaron las funciones de cada uno de sus botones en el lenguaje de programación SCL, que es lenguaje de programación para el software 3D Webmaster. El usuario puede interactuar con el entorno 3D por medio del teclado y mouse conectados al PC donde se visualiza la aplicación. Con la botonera el usuario puede: mover la estructura del robot en sus 2 sistemas de coordenadas: cartesiano (X,Y,Z) y de ejes (JOINTS), grabar posiciones y mover el punto central de la pinza a las posiciones grabadas, elegir la velocidad con la cual se producen los movimientos, abrir y cerrar la pinza del robot, habilitar y deshabilitar el controlador y ejecutar programas. Por otra parte, se realizaron los cálculos de cinemática directa e inversa para el robot SCORBOT ER-Vplus considerando las medidas reales de los eslabones L1, L2, L3, L4 en centímetros y transformadas a unidades de 3D, u3D (Figura 9), según la escala de conversión usada para diseñar las piezas del robot Scorbobot en 3D (Tabla2).

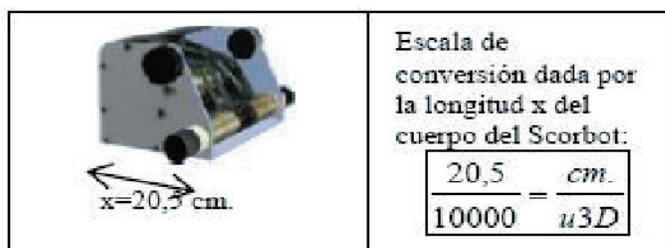


Figura 9. Escala de conversión a robot.

	L1	L2	L3	L4
cm.	36,5	21,95	22,17	13,55
u3D	17809	10712	10818	6613

Tabla 2. Conversiones de dimensiones del robot a aplicación 3D.

Metodología de diseño: La construcción del sistema se ha realizado en el software 3D Webmaster. Este software permite crear un ambiente de realidad virtual en donde podemos interactuar con los objetos creados por medio del teclado, mouse u otro dispositivo. Los objetos se crean a partir de una forma previamente diseñada en un Editor de Formas (Shape Edit), para luego agregarlos como objetos en el Editor de Mundos (World Editor). Cada objeto tiene sus propios atributos, los cuales indican las capacidades que tendrá el objeto en el mundo virtual, entre estas capacidades (propiedades) encontramos: asignación de movimientos, velocidades angulares, gravedad, detección de colisiones, texturas, sonidos, visibilidad o invisibilidad del objeto, rotaciones, código ejecutable, etc. Para el diseño y construcción del sistema de simulación del robot se realizaron: esquema conceptual del sistema (Figura 10), diseño de formas en el Editor de Objetos del software 3D Webmaster (Figura 11), construcción de objetos, aplicación de texturas a los objetos, asignación de atributos, y asignación de código en el lenguaje SCL (lenguaje propio del software).

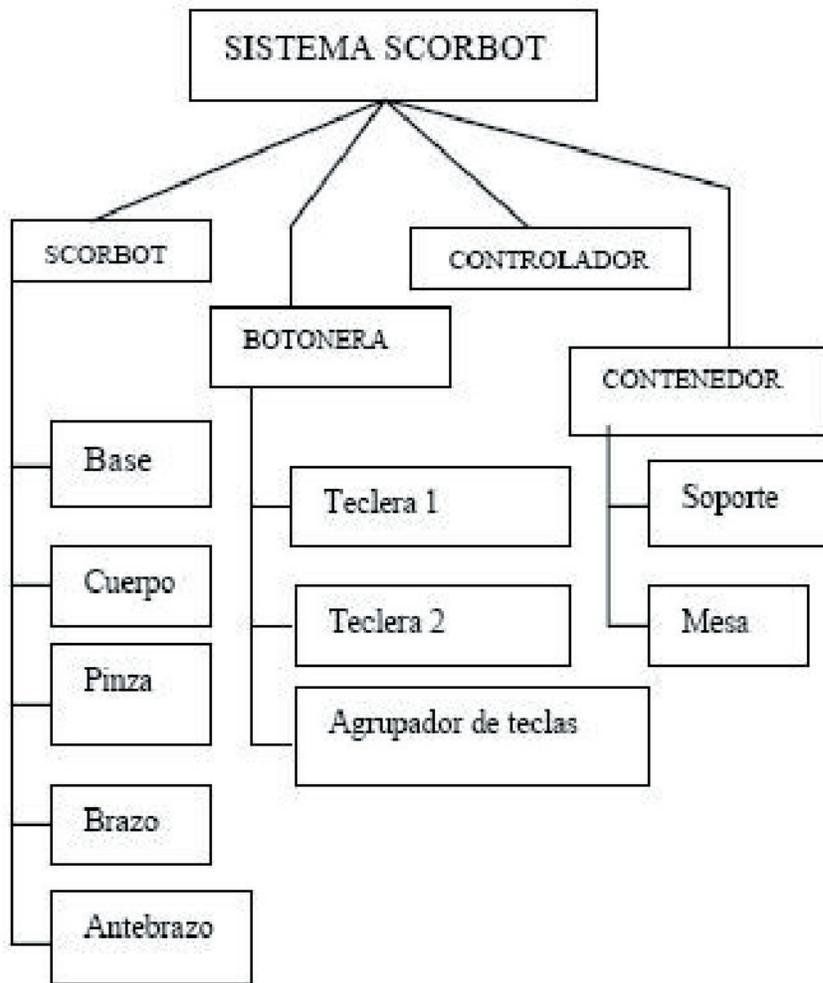


Figura 10. Esquema de componentes 3D.

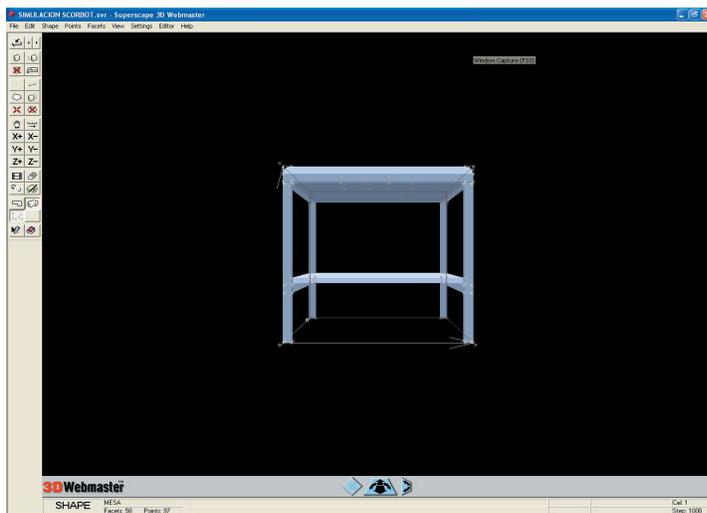


Figura 11. Editor de Objetos (Shape Editor).

Definición de objetos: se definieron todos los objetos a partir del esquema conceptual del sistema; en particular, para la botonera o teclera (Figura 13) tenemos la Figura 12 que nos muestra su esquema conceptual.

La pantalla de la botonera muestra los comandos Speed, Run, Go Position o Record Position, que se han presionado para combinar con dígitos (teclas numéricas), también muestra cuando se ha deshabilitado o habilitado el Control de la botonera con los comandos: Control Disabled y Control Enabled, y cuando se ha abortado la acción (Aborted). Además muestra en el extremo inferior izquierdo el grupo de ejes que estamos ocupando, por defecto corresponde al grupo A 2, y en el extremo inferior derecho muestra si los movimientos ocurren en modo Joints o XYZ.

Escala de conversión dada por la longitud x del cuerpo del Scorbot:

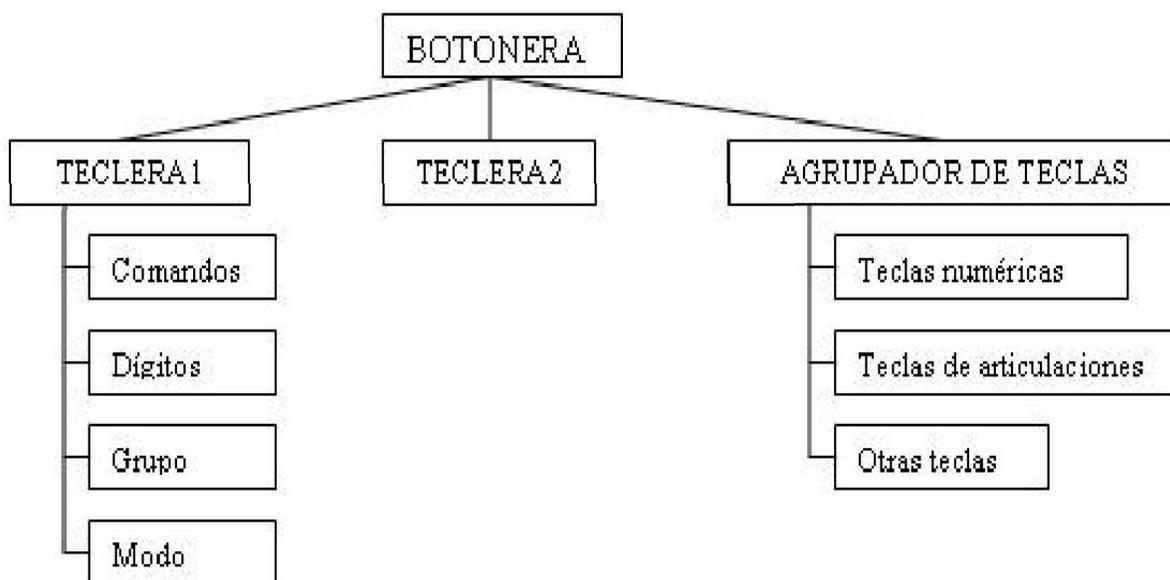


Figura12. Esquema conceptual de la botonera



Figura 13. Detalle del objeto Botonera.

Por otra parte, los eslabones del robot SCORBOT se agrupan para mover las articulaciones base, hombro, codo, inclinación y rotación de la pinza. El movimiento de la articulación base agrupa: *cuerpo, brazo, antebrazo, y pinza*; la articulación hombro agrupa: *brazo, antebrazo, pinza*; la articulación codo agrupa: *antebrazo y pinza*.

Los objetos, en la aplicación virtual, deben ser definidos tanto en posición y tamaño dentro del mundo virtual; por ejemplo, atributos del controlador y el objeto contenedor (Tabla 3); estos valores le indican, al software 3D Webmaster, en qué ubicación estará el objeto, y su tamaño, con respecto a los demás objetos y con respecto a los ejes coordenados principales del mundo virtual. Además, cada objeto debe tener definidos los atributos de objetos que le indican a la aplicación que características posee éste. Ellas son:

s: código SCL
a: velocidad angular

2 El grupo B tendría validez si se agregara un eje más a la estructura del Scorbót, como por ejemplo: una línea de desplazamiento para la Base del robot.

r: rotación

Nº	Nombre objeto	Posición			Tamaño			Atrib.
		X	y	z	x	y	z	
1	controlador	2010595	15888	1991330	23901	7316	21706	Txi
14	mesa	2000499	0	1974825	43901	43901	43901	Txi
15	sopORTE	2037750	43901	1977817	1950	46827	1950	Txi

Tabla 3. Ejemplo de atributos objeto para controlador y objetos contenedores.

Método de programación: La programación del sistema se realiza en el lenguaje SCL (del software 3DWebmaster) seleccionando la opción Edit SCL. Cualquier código debe asociarse a uno de los objetos creados en el mundo virtual, y éste puede influir en otros objetos modificando atributos como rotación, posición, velocidades angulares, texturas e invisibilidad.

Similar al Lenguaje C, el lenguaje SCL (Figura 14) permite declarar variables, asignar valores a las variables definidas, modificar y reutilizar variables e imprimir mensajes en pantalla, además cuenta con comandos de control como if-else, Do, while, Switch; funciones para manejo de strings y funciones matemáticas tradicionales y trigonométricas. SCL tiene algunas restricciones con respecto al lenguaje C, entre ellas se pueden mencionar las estructuras y los arreglos multidimensionales que no son soportados, también algunas operaciones como los incrementos y decrementos no se encuentran definidos. La ventaja es que el lenguaje SCL agrega muchas funciones para facilitar la manipulación de objetos virtuales, y así es como existen funciones para: verificar que un objeto ha sido activado, modificar rotaciones y velocidades angulares, cambiar texturas, hacer objetos invisibles y visibles, cambiar posiciones, agrupar objetos, etc. El sistema mantiene la programación en un objeto que agrupa las teclas de la botonera, y su función es llevar a cabo cada instrucción de la botonera que se activa al presionar una tecla. La activación de los botones producen el movimiento de las articulaciones del robot Scorbót en modo XYZ y en modo 14 -xx

En el mundo virtual se puede navegar libremente, y se pueden visualizar los objetos desde cualquier ángulo y con la cercanía que el usuario desee. Podemos volver siempre a cargar el mundo desde el inicio y elegir algunas opciones para visualizar, como por ejemplo ver los ejes cartesianos de acuerdo a la vista en que estamos, grupos de objetos, colisiones, etc.

El usuario interactúa directamente con la botonera que se encuentra al lado derecho; se pueden ver los comandos presionados en la pantalla de la botonera, y a la derecha se puede ver el robot moviéndose.

Caso de Uso: Los casos de uso reflejan todas las posibles maneras en que puede ser utilizado el sistema desde el punto de vista del usuario. En particular, se definieron 20 casos (Figura 16); para estos se contemplaron las combinaciones posibles que podemos tener presionando los botones de la teclera, de modo de cubrir su completo funcionamiento.

<p>Nombre: Asignar velocidad.</p> <p>Actor: Usuario.</p> <p>Función: Definir la velocidad (en porcentaje) con que se realizarán los movimientos.</p> <p>Descripción: Determina la velocidad del movimiento para el grupo actual de control de ejes(A, B ó C). La velocidad se define de 1 a 100, teniendo el 100% como velocidad máxima.</p> <p>Combinación de teclas:</p> <div style="text-align: center;"><p>El diagrama muestra tres cuadros rectangulares dispuestos horizontalmente. El primer cuadro contiene el texto 'Speed'. El segundo cuadro contiene el texto 'Tecla numérica'. El tercer cuadro contiene el texto 'Enter'.</p></div>

Figura 16. Ejemplo de caso de uso.

4 Conclusiones

Primeramente, el sistema es óptimo debido a que las pruebas realizadas al comprobar los movimientos del brazo robot, los que resultaron prácticamente idénticos al real. Además, los botones de la teclera de aprendizaje son totalmente funcionales, por lo que el aprendizaje y/o entrenamiento, con el sistema, a superado toda expectativa. Se cubrieron casi todos los casos especiales de restricción de movimientos articulares y lineales para posiciones inalcanzables por la arquitectura del robot. Sin embargo, aún existen ciertas situaciones particulares que pueden resolverse en el futuro como mejoras del sistema.

La gran dificultad se presentó al momento de adaptar el sistema de referencia teórico del brazo robot Scrobot al sistema de referencia del software 3D Webmaster, así como también el sentido de algunas referencias angulares.

Finalmente, las ecuaciones teórico-matemáticas han sido fielmente reflejadas en la aplicación virtual lo que conlleva a concluir el gran nivel de realismo que se logra a partir de la realidad virtual, y en particular con el software 3DWebMaster. Sólo se vio limitada por la capacidad máxima de definición de variables que el lenguaje SCL soporta por lo que fue necesario utilizar eficientemente las variables definidas para la aplicación

Referencias

- [1] CIMUBB. <http://www.ubiobio.cl/cimubb/>
- [2] Craig J. *Introduction to robotics: mechanics and control*. Addison-Wesley, 1989.
Figura 16. Ejemplo de caso de uso.
- [3] Groover M., Weiss M., Nagel R. and Odrey N. *Industrial Robotics: Technology, programming and applications*. Mc Graw Hill, New York, 1986.
- [4] *Guía de Referencia ACL Lenguaje de Control Avanzado*. Eshed Robotec, 1982.
- [5] *Manual de Usuario SCORBOT-ER Vplus*. Eshed Robotec, 1982.
- [6] Ollero A. *Robótica: Manipuladores y robots móviles*. Alfaomega-Marcombo, Barcelona, 2001.
- [7] Parra J.C., García R., Santelices I. **Introducción Práctica a la Realidad Virtual**. Ediciones Universidad del Bío-Bío, 2005.