

Diseño e implementación de un sistema de control para el robot móvil P3-AT por medio de gestos manuales

Manuel Veintimilla, Rodolfo Gordillo, Paúl Ayala

Resumen—Este documento detalla como se construyó una relación entre un usuario y un robot móvil a través de una red cliente servidor inalámbrica utilizando el sensor Kinect para la respectiva adquisición de datos. El robot se mueve con los gestos indicados por la persona, ya sea que se necesite explorar o simplemente trasladar el robot. El usuario debe desarrollar la destreza para manipular el robot siguiendo el manual de usuario e iniciar con el simulador, así con este procedimiento se evita tener errores desde la conexión con el robot hasta la ejecución de los comportamientos. Los comportamientos se ejecutan con botones en la interfaz del cliente, y cada uno de forma exclusiva.

Index Terms—Kinect, simulador.

I. INTRODUCCIÓN

EL siguiente trabajo sirve de base para proyectos a futuro, debido a su configuración se puede utilizar en actividades exploratorias y militares más concretamente; está diseñado para que a través de una red informática cliente-servidor el robot móvil interactúe con una persona quien va a comandar los movimientos del mismo. Dentro de la red el cliente estará conectado a un sensor Kinect, que permitirá receptor los distintos gestos de una persona, los mismos que se transmitirán hacia el robot, no solo se enviarán estos gestos corporales sino que también se enviarán diferentes comandos para realizar tareas diversas como eventos de parada, o avanzar a cierta posición ya sea de frente o en sentido opuesto, todas estas comportamientos serán descritos en detalle en la configuración de comportamientos del robot. En concreto la función principal del proyecto será la de manejar el robot móvil por medio de gestos realizados con las manos, dentro de esta área estará comprendido el avanzar de frente, parada, giros y retroceder. Cada conducta del robot, desde el manejo por gestos manuales así como las introducidas por comandos deberán ser exclusivas, es decir, no se podrán realizar dos comportamientos a la vez para evitar errores graves, que afectarían la funcionalidad del robot o incluso la de todo el proyecto. El servidor finalmente será el encargado de receptor todos los mensajes enviados por el cliente a través de la red, principalmente debe evitar que efectúen errores en la ejecución de las órdenes del cliente y también permitir la fluidez en la

manipulación del robot, impidiendo los choques o roces con objetos en el campo.

II. FUNDAMENTOS

Kinect fue presentado por Microsoft como el dispositivo complementario para la consola de videojuegos Xbox 360®. Es un controlador de juego que utiliza un sensor de movimiento para controlar todo el cuerpo, su interfaz natural reconoce gestos, comandos de voz y objetos o imágenes, su lanzamiento comercial se realizó en el 2010. La utilización de este dispositivo supone grandes avances en la tecnología que presenta diferentes oportunidades para la investigación en diversas áreas tanto científicas como médicas y hasta espaciales[1].

Tanto el sensor Kinect como el SDK, nos permiten realizar a una amplia gama de aplicaciones como captura de video en tiempo real, realizar análisis y modelación en tercera dimensión, reconocimiento de voz humana y desarrollo de aplicaciones de manos libres, que se puedan controlar mediante la voz. Podemos construir una serie de aplicaciones para el mundo real que dependiendo del área de desempeño nos abren las puertas de diferentes estudios:

- Salud: Se puede construir diferentes aplicaciones para la salud, tales como la medición de ejercicio, los pacientes de control, sus movimientos corporales, y entre otras.
- Robótica: Se puede utilizar como un sistema de navegación para los robots, ya sea mediante el seguimiento de los gestos humanos, comandos de voz, o por los movimientos corporales humanos.
- Educación: Se puede construir diversas aplicaciones para los estudiantes y niños para educar y ayudar a que aprendan las materias, por medio de sus gestos y comandos de voz.
- Sistema de seguridad: Kinect se puede usar para el desarrollo de los sistemas de seguridad donde se puede realizar un seguimiento de los movimientos del cuerpo humano o de la cara y enviar las notificaciones.
- Realidad virtual: Con la ayuda de la tecnología 3D y de seguimiento de gestos humanos, muchas aplicaciones de realidad virtual se puede construir utilizando el sensor de Kinect.
- Entrenamiento: Se puede utilizar como un entrenador mediante la medición de los movimientos de las articulaciones del cuerpo humano, proporcionando retroalimentación en directo a los usuarios, si las articulaciones

Manuel Veintimilla Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador, e-mail: manuavc@gmail.com.

Rodolfo Gordillo Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador

Paúl Ayala, Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador, jpayala@espe.edu.ec

se mueven de una manera apropiada mediante la comparación de los movimientos con los datos almacenados previamente.

- Militar: Se puede usar para manipular aviones no tripulados inteligentes para espiar a las líneas enemigas

II-A. Funcionamiento del sensor

El sensor Kinect tiene la capacidad de capturar una imagen 3D primitiva de los objetos en frente, sin importar las condiciones de iluminación de la habitación. Utiliza un emisor infrarrojo y un receptor de profundidad infrarrojo que es un sensor CMOS monocromo (semiconductor complementario de óxido de metal). La columna vertebral detrás de esta tecnología es del chip PrimeSense, la figura(1) muestra el procedimiento del sensor para la captura de datos.

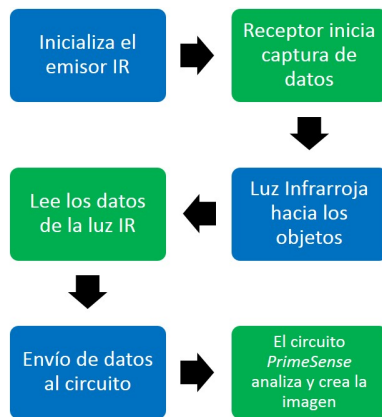


Figura 1. Procesamiento de datos de profundidad

II-B. Funcionamiento de Skeleton Traking

El sensor Kinect devuelve los datos de profundidad de los que podemos identificar fácilmente los píxeles que representan a los individuos. El “skeleton traking” no sólo es sobre la detección de las articulaciones mediante la lectura de la información de los individuos; más bien, se trata de la lectura del movimiento del cuerpo completo. En tiempo real se plantea que el reconocimiento es difícil y desafiante debido a las diferentes formas, tamaños, alturas y posturas del ser humano. Para superar estos problemas y para realizar una localización de diferentes articulaciones independientemente de la pose del cuerpo, Kinect utiliza el “rendering pipeline”, que típicamente, recibe la representación de una escena tridimensional como entrada y genera una imagen en 2 dimensiones como salida.

Lo que sigue en el reconocimiento inmediato de poses, es etiquetar las partes del cuerpo mediante la creación de segmentos, esta creación segmentos se realiza haciendo coincidir los datos probables con similitudes. Kinect utiliza una estructura de árbol de decisiones para que coincida con los datos de un determinado tipo de cuerpo humano, este procedimiento se conoce como “Decision Forrest”[2]. Son varios los algoritmos y actividades que se realiza para usar el skeleton tracking:

- Los datos de profundidad entran en el proceso de “rendering pipelines” y los hace coincidir con los datos

etiquetados del “decision forrest” y genera los segmentos corporales deducidos.

- Una vez que todas las piezas están identificadas basados en los datos etiquetados, el sensor identifica las articulaciones del cuerpo.
- El sensor calcula entonces la vista 3D desde la parte superior, frontal y la izquierda de las articulaciones propuestas.
- Finalmente, el sensor inicia el seguimiento del movimiento del esqueleto y del cuerpo humano en base a los puntos de articulación propuestos y la vista 3D.

III. ARQUITECTURA DEL SISTEMA

Existen dos partes principales en este proyecto, el cliente y el servidor, el cliente-PC está conectado a un sensor de Kinect. Con la ayuda de este sensor se reconocerán varios gestos diferentes. El cliente está conectado a través de Wi-Fi a un servidor que procesa los comandos y los movimientos, en este proyecto el servidor es una pc montada sobre la plataforma del robot móvil ya que la comunicación inalámbrica de los robots no está disponible. Esta aplicación se puede utilizar en cualquiera de los robots móviles del laboratorio pues el proyecto se enfoca en el movimiento de los robots y no en las características individuales de cada uno, es decir se podrá utilizar tanto en los robots Pioneer 3-DX como con el Pioneer P3-AT, a excepción de elementos propios de cada modelo como gripper o bumpers. El usuario escoge el modo de conducir, el sensor percibe los movimientos y los transmite al cliente, luego éste calcula el respectivo ángulo entre las manos, y lo envía como texto al servidor por medio de la red previamente configurada, el servidor realiza comparaciones con la información enviada y ejecuta los movimientos, la figura (2) describe la configuración del sistema.

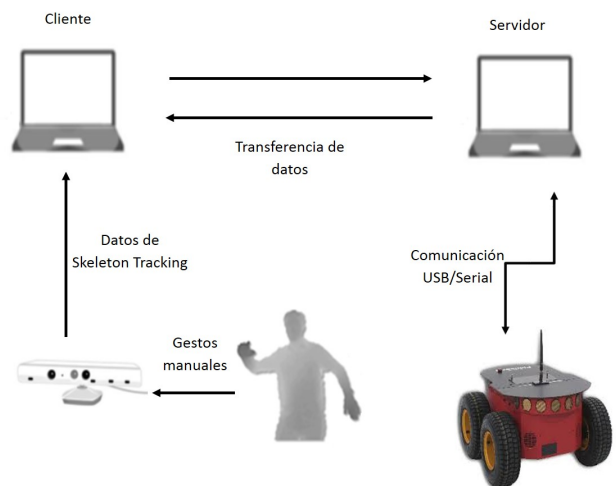


Figura 2. Arquitectura de comunicación y adquisición de datos

El cliente se desempeña a través de una interfaz de usuario, con las respectivas órdenes y/o comandos a ejecutar, al ponerse una persona frente al sensor, este comienza inmediatamente a detectar los movimientos, realiza un modelo del esqueleto de la persona en frente y posiciona las articulaciones, estos

datos son transmitidos al cliente para realizar operaciones. La interfaz de usuario se encarga de inicializar las funciones del Kinect y del socket de comunicación, la figura(3) muestra el diagrama de las operaciones del cliente, se busca conectar con el servidor, luego se puede escoger un modo de operación hasta que el usuario opte por salir, en tal caso se finaliza la aplicación.

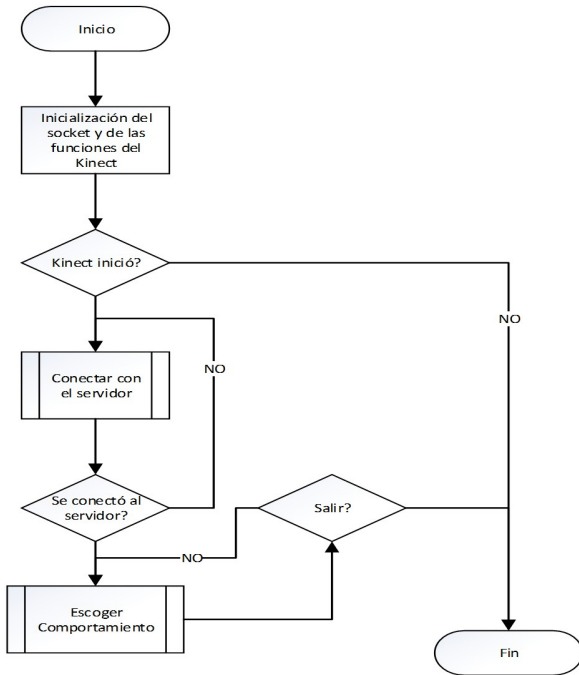


Figura 3. Diagrama de flujo del funcionamiento del programa cliente

Después de ingresar la dirección IP y el puerto de comunicación el cliente espera por la réplica del servidor, si se concreta la conexión entonces se muestra el mensaje recibido. Si se selecciona el modo conducir, el método lanza un proceso asíncrono, para enviar el ángulo de rotación del robot. Una vez que se calcula el ángulo el proceso asíncrono toma ese dato y lo envía por el socket hacia el servidor, luego actualiza el ángulo tomado de las posiciones de las manos y se vuelve a enviar, mientras no se cancele la acción mandar el ángulo el proceso se mantiene. Todo este proceso se mantiene independiente del programa principal, así se ahorra tiempo de procesamiento, y es más eficiente el uso de la memoria del computador.

La aplicación del servidor se desarrolla en el robot Pioneer P3-AT, es necesario montar una PC tipo laptop en la plataforma del robot, en esta computadora se instala el servidor del proyecto, y se comunica con el robot por medio de puerto USB, cabe aclarar que para dicha comunicación es necesario utilizar un cable de transmisión USB a serial, porque la plataforma del Pioneer no cuenta con este puerto. La aplicación del servidor con la programación de sus comportamientos en lenguaje C++, se puede realizar en el sistema operativo que se desee, en este caso se usó Windows, y se debe incluir la librería ARIA, exclusiva para la programación de los robots Pioneer. Una vez que se logra la comunicación entre el robot y la PC servidor, ya se puede interactuar con el cliente y así completar

la comunicación total, desde el usuario hasta el robot. El P3-AT se moverá dependiendo de las acciones que envíe el cliente, cada uno de los comportamientos está configurado en el programa servidor, cuando se pida realizar una acción, ésta se realizará excluyendo a las demás, la figura(4) muestra la configuración del servidor.

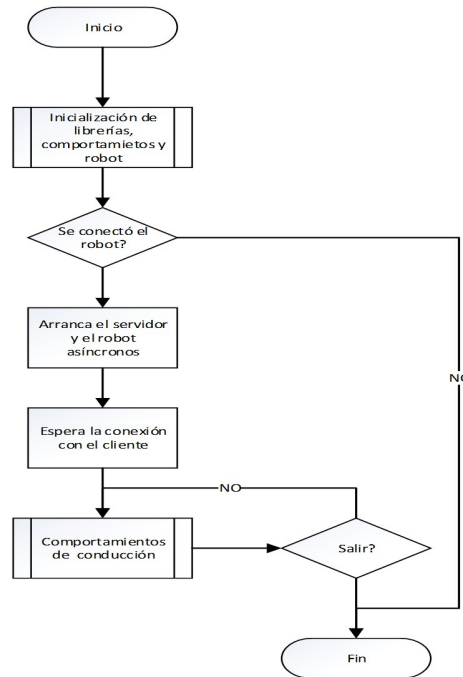


Figura 4. Diagrama de flujo del funcionamiento del programa servidor

El primer paso es la inicialización de las librerías y se crea el grupo de acciones configuradas para cada comportamiento del robot. Al final el servidor siempre está esperando una respuesta del cliente si se envía "salir" termina la aplicación. El subproceso inicialización carga las librerías necesarias y crea los objetos robot y sonar, este último es agregado al robot y sirven para evitar las colisiones en cada comportamiento. La selección de comportamientos se realiza en un lazo donde cada dato recibido se compara para determinar la acción solicitada por el cliente, y se mantiene en este lazo hasta que se pide salir. Este proyecto propone una serie de comportamientos para el robot, a continuación se muestra el conjunto de comportamientos que se permitirán para el robot móvil:

- Comportamiento paseo: Realiza los movimientos y las funciones del programa se aseguran que el robot no colisione con ningún obstáculo. Además se deben evitar los obstáculos que están más lejos, finalmente cuando no hay ningún objeto frente al robot, éste se mueve con una velocidad constante de 250 mm/s.
- Comportamiento distancia: Permite conducir el robot en una distancia específica, el robot se desplaza hacia adelante si el parámetro es positivo y hacia atrás si es negativo.
- Comportamiento conducir: Este comportamiento se ejecuta directamente desde el cliente, para lo cual se requiere de dos parámetros, ángulo y velocidad. En el

proceso de conducir por gestos, el dato que se recibe principalmente es el ángulo, la velocidad es constante para evitar que robot acelere indiscriminadamente, esta es una forma de limitar los posibles errores que el usuario puede cometer cuando se conduce el robot, dentro de este comportamiento también se puede ejecutar las acciones de parar por completo al robot o conducir en retroceso.

- Comportamiento atrás: Este modo de operación permite al robot la conducción en retroceso, con una velocidad constante de 150 mm/s., así se evita que el usuario conduzca el robot hacia un obstáculo.

III-A. Interfaz de usuario

La interfaz gráfica es el programa que interactúa directamente con el usuario y que le permite realizar todas las tareas definidas en este proyecto, aquí se valida todo aquello que el cliente emite al servidor, para que este último se dedique específicamente a ejecutar los comportamientos del robot descritos anteriormente. El cliente debe empezar la conexión con el servidor, el mismo que debe estar previamente inicializado y en comunicación con el robot o el simulador, el usuario ingresa la dirección IP y el puerto de comunicación, así el servidor envía un mensaje “Hola Cliente” reconociendo la conexión con el usuario, como se muestra en la figura(5).



Figura 5. Interfaz de usuario, ventana de configuración y conexión

En la sección de configuración se puede observar el título y los campos a ingresar, seguido de respuesta de reconocimiento del servidor, en ese instante se puede seleccionar la pestaña “Manejo”, donde se opta por los distintos comportamientos. La figura(6) muestra la pestaña manejo que permite escoger los comportamientos a ejecutar y se visualiza los gestos realizados, el dato enviado al servidor, la ilustración del “skeleton tracking” con el ángulo obtenido entre las manos del usuario y los botones para cada comportamiento.

IV. ANÁLISIS Y PRUEBAS DEL SISTEMA

El sistema está diseñado para que ejecute el servidor, estando éste conectado al robot o al simulador, luego se espera a que el cliente se comunique para iniciar la aplicación. Si no

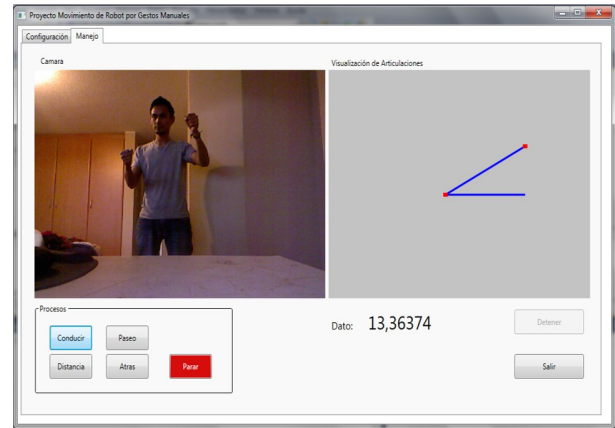


Figura 6. Interfaz de usuario, ventana de manejo por medio de gestos y comportamientos

es posible la comunicación con el robot se lanza una excepción informando un error y saliendo del programa, una vez que se establece la comunicación el programa se dispone a esperar al usuario, la comunicación se establece principalmente entre el servidor que opera directamente el robot y el cliente que envía las instrucciones de movimiento ya sea por comandos o por gestos manuales. Estas pruebas están realizadas tanto para el simulador como para el robot P3-AT. Para iniciar la aplicación completa es necesario que el servidor esté conectado con robot y el cliente debe estar conectado con el sensor, una vez hecho esto ya se puede manejar el robot.

El programa controla el robot por medio de comandos o por gestos manuales el cual es la aplicación principal, para el correcto funcionamiento, el robot toma los valores de velocidad y ángulo determinados por la posición de las manos. El rango de ángulos que se puede ingresar va de -90° a 90° , aunque no es necesario llegar a estos valores máximos pues el robot gira con una velocidad de rotación y no con un ángulo directamente. Para obtener el ángulo es necesario conseguir las posiciones de las manos, luego se traza una línea horizontal imaginaria, desde la mano izquierda hasta la posición de X_D de la mano derecha, esta línea será la referencia para marcar el ángulo de giro de las manos y posteriormente del robot, la posición Y en ambas manos es igual así se mantiene una línea horizontal, por tanto $Y_I = Y_D$. Una vez que se tiene la línea de referencia se traza una línea dada por la posición de las manos como se muestra en la figura(7) la cual determina el ángulo que será enviado al robot.

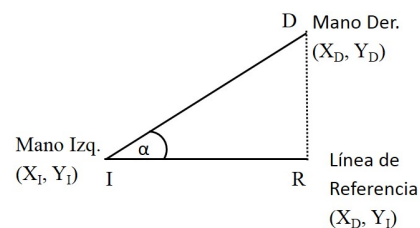


Figura 7. Ángulo entre las manos y línea de referencia, con sus respectivas coordenadas

Para obtener el ángulo α se utiliza la definición trigonométrica de coseno, donde el coseno de un ángulo está determinado por el cociente entre el cateto adyacente \overline{IR} y la hipotenusa \overline{ID} (3). Ahora obtenemos la dimensión de la hipotenusa y del cateto, como se usa en (1) y (2):

$$(\overline{ID}) = \sqrt{((X_D - X_I)^2 + (Y_D - Y_I)^2)} \quad (1)$$

$$(\overline{IR}) = \sqrt{((X_D - X_I)^2)} \quad (2)$$

$$\alpha = \cos^{-1} \frac{\overline{IR}}{\overline{ID}} \quad (3)$$

Cuadro I
ÁNGULOS ENVIADOS AL SERVIDOR

	α	$\frac{\alpha}{2}$	$\frac{\alpha}{3}$	$\frac{\alpha}{5}$
1	03.1233	00.0140	00.6020	00.1307
2	05.1826	03.4693	00.7434	01.6580
3	19.2739	07.9349	01.4244	04.0390
4	33.0039	10.2589	06.1634	05.6904
5	42.2090	13.8514	08.5950	07.2268
6	51.3436	17.4088	12.3703	08.4278
7	54.13.54	24.5446	14.7429	09.3981
8	53.8911	22.9922	18.3775	11.0203
9	48.4466	24.5446	19.5532	12.0223
10	35.1111	26.1011	19.5532	13.3193
11	04.5604	27.7545	18.1034	12.8708
12	03.2308	28.6312	12.1036	09.2864
13	-06.4376	27.6133	08.9370	06.0075
14	-18.8650	20.2385	06.4740	03.3045
15	-26.7977	13.0157	04.5478	01.9365
16	-35.1878	09.2853	02.4528	00.6450
17	-41.4424	05.3992	-00.4931	-00.5881
18	-42.4420	02.9769	-03.1681	-02.3996
19	-43.1771	-01.4422	-08.4258	-05.8059
20	-40.5505	-05.8563	-12.8674	-06.4522
21	-19.5209	-08.7922	-14.6944	-05.6465

La prueba se realizó desde una posición donde el ángulo es cercano a cero, luego se gira lentamente los brazos hacia arriba para tomar ángulos positivos y después se gira hacia abajo para tomar los ángulos negativos, cada adquisición se hizo dividiendo el ángulo obtenido para un número determinado. El robot tiene en su programación interna un método para girar `setRotVel()`, pero este dato es la velocidad de rotación y no el ángulo de rotación que se está obteniendo, por ello cuadro(1) muestra algunos resultados de los valores dados al método `setRotVel()`, para escoger el mejor desempeño.

Como se ve en la figura(8) si se envía directamente el ángulo dado por el sensor, el robot haría giros demasiado bruscos, lo cual puede llegar a ser peligroso es por eso que se opta por dividir el ángulo para que la tasa de cambio sea menor, que si se envía el dato sin cambios. Para esta configuración es preferible escoger un flujo de datos que no de cambios bruscos, por ello es que `Ang/2`(donde `Ang = \alpha`) cumple con estos requerimientos mínimos

V. CONCLUSIONES

El diseño del sistema permitió el manejo del robot móvil con gestos manuales, y a través de botones en la interfaz del cliente, con los que se controla acciones aún más específicas.

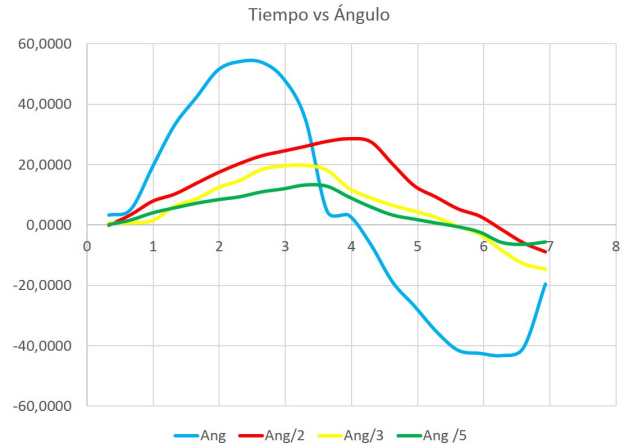


Figura 8. Tasa de cambio de los ángulos enviados al servidor

La utilidad de los sensores de profundidad, es muy amplia, no solo se puede manejar un robot móvil por medio de gestos sino que también, se pueden diseñar sistemas donde se manipule robots en áreas de riesgo de forma sencilla, al imitar los movimientos humanos.

La comunicación entre cliente y servidor se estableció correctamente, y es necesario que el envío de datos del sensor se haga en un subproceso asíncrono de la aplicación principal, así se pudo evitar que el modo "conducir" interfiera con los procesos de la aplicación.

La cámara y el sensor de profundidad son eficientes, pero sin embargo tiene limitantes ya que si el usuario se encuentra muy cerca de él o muy lejos, los datos obtenidos serán erróneos, haciendo que el robot se mueva de forma muy irregular y errática.

El ángulo de rotación tuvo que ser reducido al 50 %, pues la función que permite girar al robot admite una velocidad de rotación y no directamente un ángulo específico, así el ángulo α se divide para dos, porque con este valor se obtienen mejores resultados en cuanto a la respuesta del sistema.

La utilización de la clase `ArActionGroup`, facilita la programación de los comportamientos del robot haciendo que todos ellos se desarrollen en un solo grupo de acciones, permitiendo que el servidor pueda pasar de un comportamiento a otro inmediatamente, con el llamado de funciones.

Una ventaja de agrupar los comportamientos es que se puede ampliar la interacción del cliente con el robot programando movimientos adicionales, que complementen las utilidades de la aplicación.

REFERENCIAS

- [1] Abhijit Jana, Kinect for Windows SDK Programming Guide. Birmingham, U.K.: Packt Publishing Ltd, 2012
- [2] John MacCormick. (2015) How does the Kinect work? [En línea]. <http://www.cs.bham.ac.uk/~vjk201/Teach/Graphics/kinect.pdf>
- [3] Gabriela Andaluz, "Modelación Identificación y Control de Robots Móviles," Escuela Politécnica Nacional, Quito, Tesis de Ingeniería en Electrónica y Control 2011.
- [4] Aníbal Ollero, ROBÓTICA Manipuladores y robots móviles. Barcelona, España: Marcombo, 2001.
- [5] Lukas Häfele. (2012, Marzo) Gesture Control. [En línea]. <http://www.gesture-control.com.eu/>

- [6] MobileRobots Inc., Pioneer 3 Operations Manual with MobileRobots Exclusive Advanced Robot Control & Operations Software, 2006.
- [7] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, and Mark Finocchio. (2011) Real-Time Human Pose Recognition in Parts from Single Depth. [En línea]. <http://research.microsoft.com/pubs/145347/BodyPartRecognition.pdf>
- [8] Jarrett Webb and James Ashley, Beginning Kinect Programming with the Microsoft Kinect SDK. Nueva York, N. Y.: Apress, 2012.