

**DRAWING ROBOT CONTROLLED THROUGH THE KINECT SENSOR****ROBOT DIBUJANTE CONTROLADO MEDIANTE EL SENSOR KINECT**

**MSc(c). Edgar Andrés Gutierrez Cáceres\***, **MSc(c) Camilo Ernesto Pardo Beainy\*\***  
**PhD. Luis Fredy Sosa Quintero\***, **MSc. Fabián Rolando Jiménez López\*\***

\* **Universidad Santo Tomas**, Facultad de Ingeniería Electrónica, GIDINT.  
Calle 19 No 11 – 64 Centro Tunja, Boyacá, Colombia.  
Teléfono: (57) (8) 7440404 Ext. 31057.

E-mail: {edgar.gutierrez, camilo.pardo, luis.sosa}@usantoto.edu.co.

\*\* **Universidad Pedagógica y Tecnológica de Colombia**, Facultad de Ing. Electrónica.  
Ave. Central del Norte 39-115, Tunja, Boyacá, Colombia.  
Teléfono: (+578) 7405626 - 7430953 Ext: 2521-2523-2527.  
E-mail: fabian.jimenez@uptc.edu.co

**Abstract:** In this paper we present the implementation of a robot 3 DOF which is controlled by processing digital imaging, the sensor used comprises an RGB camera and infrared emitter that projects a dot pattern a variable intensity light having the ability to hand scanning and extract the coordinates x, y. passing a processing scheme and converting the data into a number of rotations of the robot servo motors. Describes generalities and results obtained in the investigation performed.

**Keywords:** arm, coordinates, images, processing, robots servomotors.

**Resumen:** En este artículo se presenta la implementación de un brazo de 3 DOF planar que es controlado mediante procesamiento digital de imágenes, el sensor utilizado consta de una cámara RGB y un emisor de infrarrojos que proyecta un patrón de puntos de luz con sus coordenadas x, y, pasándolas a un esquema de procesamiento y convirtiendo los datos en una serie de rotaciones en los servomotores del robot. Se describen las generalidades y resultados obtenidos dentro de la investigación realizada

**Keywords:** arm, coordinates, images, processing, robots servomotors.

## 1. INTRODUCCIÓN

Un aporte importante dentro de los avances tecnológicos en las últimas décadas es el procesamiento digital de imágenes, que han traído consigo diversas aplicaciones en diferentes áreas de interés, el ejemplo más conocido y básico en la vida cotidiana es el Kinect de Microsoft, el cual permite hacer una interacción con la consola de video juegos sin necesidad de un contacto físico con ningún tipo de controlador, por tal razón tiene diversas utilidades no solamente en este sector sino también aplicado en áreas como la robótica,

industria e investigación (Berkeley & Urtasun, 2010), (Liu, *et al.*, 2008), (Oh, *et al.*, 2010). En este orden de ideas se busca realizar el control de un brazo robótico o manipulador robótico que tenga la capacidad de imitar los movimientos del brazo de una persona al ejecutar un dibujo, obteniendo los datos de la imagen mediante el sensor Kinect (Demirdjian, *et al.*, 2003), (Jiang, *et al.*, 2010), (Tran, *et al.*, 2008), (Wang & Chung, 2010).

Este tipo de sensor adquiere la caracterización de una imagen mediante la proyección de una nube de puntos infrarrojos o rejilla de puntos infrarrojos

que no son visibles para el ser humano y que hacen referencia a una serie de datos de un espacio 2D/3D y corresponden a la textura, forma, color, profundidad, del medio que se está capturando, esto se logra gracias a una hardware que obtiene la imagen en color y adicionalmente tienen la función de medir la distancia a la que se encuentran los elementos en el campo de visión del Kinect (Oh, Islam, & Lee, 2010; Salem, Kopp, Wachsmuth, & Joublin, 2010; Suma, Lange, Rizzo, & Krum, n.d.).

Los datos de movimiento obtenidos por el sensor son enviados como una serie de coordenadas X, Y a un algoritmo que se realiza en Arduino para que puedan ser evaluadas, procesadas y así controlar cada uno de los servomotores asociados al brazo robótico

## 2. DISEÑO DEL BRAZO

El objetivo del diseño y construcción de este manipulador es crear cierta similitud con la anatomía de las extremidades superiores del cuerpo humano, por tal razón el brazo robótico también recibe nombres de términos como muñeca, hombro, codo, antebrazo y brazo.

El mecanismo mecánico se compone de una parte fija donde se encuentra la placa Arduino, posee tres articulaciones o servomotores que permiten el desplazamiento o giro de los dos enlaces correspondientes al brazo y antebrazo y en la parte final se encuentra una pinza que es la encargada de sostener el lápiz que corresponde a la muñeca del manipulado (Fossati, Berkeley, & Fua, 2009; Sapp, Weiss, & Taskar, n.d.; Sheikh, 2008; Takahashi, Nagasawa, & Hashimoto, 2008; Wu & Huang, n.d.).

### 2.1 Cinemática

La Cinemática estudia las propiedades geométricas del movimiento de los puntos fijos en enlaces sin tener en cuenta sus masas o de las fuerzas que actúan sobre ellos.

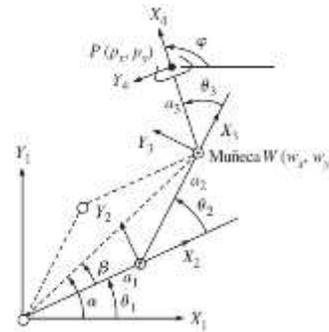


Fig. 1. Brazo 3 DOF planar consideraciones cinemática.

Como se muestra en la Fig.1. Se realiza un modelo del mecanismo con todos los elementos que lo componen, la posición y distancia en la cual se encuentran cada una de las articulaciones, mostrando las características de posición angular con respecto a los ejes de coordenadas cartesianas, esto con el fin de hacer un estudio detallado de los posibles movimientos que pueda obtener el mecanismo. El resultado de esto es una serie de matrices que definen la localización espacial del brazo ya que el movimiento relativo en las articulaciones junto con los enlaces resulta en el movimiento de cada uno de los elementos que posicionan todo el brazo en una orientación deseada.

El problema de cinemática inversa en el caso de un brazo robótico se divide en dos sectores, el primero sector se analiza en la parte superior del brazo que comprende la zona del hombro del mismo y el brazo hasta el codo, el segundo sector comprende desde el codo, antebrazo y la muñeca, que en nuestro caso será la zona de objeto de estudio, por tal razón sólo hay una configuración cinemática del brazo y por lo tanto una solución del problema de cinemática inversa, por ejemplo el método de solución geométrica. Teniendo en cuenta la Fig.1. Se obtiene la Ecuación (1):

$$W_x^2 + W_y^2 = a_1^2 + a_2^2 - 2a_1a_2 \cos(\pi - \theta_2) \quad (1)$$

Para obtener los ángulos se determina que:

$$\theta_2 = \cos^{-1}(c_2) \quad (2)$$

$$\theta_1 = \alpha \pm \beta \quad (3)$$

Donde,  $\alpha$  y  $\beta$  :

$$\alpha = a \tan 2(W_y, W_x) \quad (4)$$

$$\beta = \cos^{-1} \left( \frac{W_x^2 + W_y^2 + a_1^2 + a_2^2}{2a_1a_2\sqrt{W_x^2 + W_y^2}} \right) \quad (5)$$

Finalmente se tiene que  $\theta_3$ , además se tiene en cuenta que  $\varphi$  es la orientación final deseada:

$$\theta_3 = \varphi - \theta_1 - \theta_2 \quad (6)$$

### 3. PROCESAMIENTO DE LOS DATOS

#### 3.1 Procesamiento digital de imágenes

Por medio del procesamiento digital de imágenes en el software *Processing* se realiza el escaneo de la posición del brazo de la persona mediante el sensor Kinect, el sensor emite un haz láser infrarrojo que proyecta un patrón de puntos sobre los cuerpos llamada nube de puntos. Una cámara infrarroja capta este patrón y el procesador utiliza algoritmos para capturar las imágenes, el programa desarrollado en *Processing* proporciona información acerca de los puntos que corresponden a las articulaciones del brazo de la persona, al captar un desplazamiento en cada uno de los puntos aparece un  $\Delta x$  proporcional al movimiento, de esta manera, es posible obtener la información posicional en tiempo real del brazo.

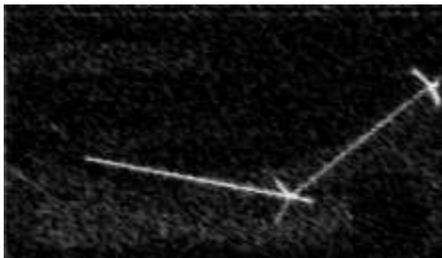


Fig. 2. Esqueleto del brazo con sus respectivas articulaciones.

Al obtener dicha información se traduce en movimientos de servomotores para poder imitar los movimientos captados por el sensor a través de la plataforma Arduino como lo muestra el siguiente diagrama.

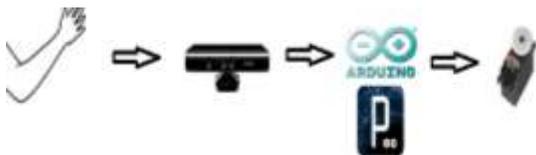


Fig. 3. Diagrama Procesamiento de los datos.

El Sensor Kinect se debe colocar en la parte superior del brazo de forma perpendicular a la superficie alineándolo con la base del brazo, como se muestra en a siguiente figura.

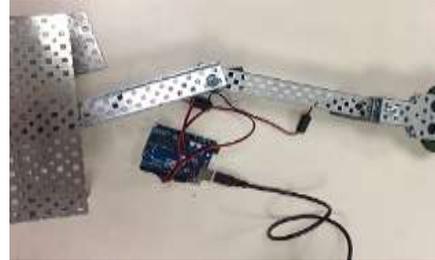


Fig. 4. Fotografía tomada desde el Kinect.

Esto se hace con el fin de que el campo de visión del Kinect apunte al entorno de trabajo del brazo, para poder aplicar los diferentes movimientos en las coordenadas del plano x, y. A este punto donde se logra la alineación y la perpendicularidad se le va a dar el nombre de (0,0).

Para poder lograr esto es necesario calibrar la nube de puntos, esto quiere decir que se realiza una rotación de la nube de puntos de tal forma que la mesa se vea en un determinado Angulo (de lado), con el fin de poder tener una captura del origen del brazo robótico, el objetivo de esto es hacer que el punto nublar coexista en el mismo sistema de coordenadas que el robot.

Luego se debe realizar un filtrado de la nube de puntos, ya que se está observando la escena desde un lado, puede verse otros objetos en el entorno, elementos que no son necesarios para el seguimiento como la misma superficie de la mesa, fondos, u otros objetos, todo esto debe suprimirse con el objetivo de que en la nube de puntos solamente se pueda observar el brazo de la persona cuando pase por el frente del sensor como una serie de puntos negros al tener un área de trabajo definida (Castro *et al.*, 2013).

Dentro de la realización del dibujo un de los movimientos más importantes es el módulo de la muñeca que controla la pinza, por tal razón el sensor debe detectar los dedos de la mano por medio de una región espacial en toda la nube de puntos, cuando se realiza un dibujo los dedos que sostienen el lápiz es el punto más bajo por encima de la mesa, considerando la mano exclusivamente como se muestra en la siguiente imagen.



Punto Mas bajo

Fig. 5. Posición de la mano al realizar un dibujo

Por tal razón se debe encontrar este punto en el área de escaneo del sensor, que matemáticamente hablando es el punto más bajo de la coordenada  $z$ . Dentro del filtrado hay que tener en cuenta una condición muy importante, cuando el robot detecte la mano él se va a mover de la misma manera que la persona lo haga, si se realizan movimientos rápidos, repentinos o por fuera de ciertas regiones de área donde el brazo no debería estar, por ejemplo hacia arriba, dentro del entorno de visualización del sensor, se tendrán problemas o daños de tipo mecánicos o físicos con el brazo, por tanto se debe demarcar una región o área donde el brazo funcione correctamente y otra área donde el robot haga movimientos suavizados o cortar con la nube de puntos ciertos espacios o ciertas alturas para evitar este tipo de inconvenientes.

### 3.2 Procesamiento de datos

Arduino es una plataforma abierta, lo que corresponde con la categoría de Hardware *OpenSource*, posee diferentes librerías que funcionan de la mano de *processing*, en el procesamiento de datos en Arduino se extrae las coordenadas de la nube de puntos  $x$ ,  $y$  de los sectores donde se hace el movimiento y se convierten en posiciones que son llevadas a los servomotores de tal forma que correspondan a la misma ubicación del brazo de la persona.

Como se dijo anteriormente el brazo posee tres módulos, cada uno de estos accionado por un servomotor diferente que pueden rotar 180 grados, los movimientos que son captados se relacionan directamente con un valor en coordenadas polares de longitud y un valor Angular que se mide desde la base y las articulaciones con respecto el punto final del brazo.

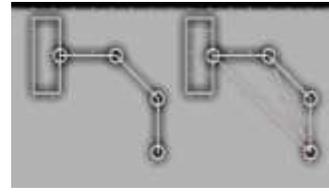


Fig. 6. Proyecciones de longitud de la base y cada una de las articulaciones con respecto a la posición final del brazo.

Teniendo en cuenta que cada una de las articulaciones y enlaces que conforman el brazo constituyen triángulos con proyecciones de líneas imaginarias que se realizan desde el inicio del brazo hasta el punto final del mismo. Aplicando relaciones trigonométricas podemos hallar los diferentes valores angulares y de longitud con el fin de conocer las coordenadas polares correspondientes (Pardo et al., 2013).

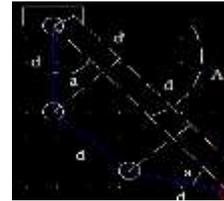


Fig. 7. Proyecciones de distancias y ángulos de las articulaciones con respecto a la posición de final del brazo.

Se conoce con precisión cuantos grados deben moverse los servomotores para llegar a una posición indicada. Esta conversión de coordenadas se puede hacer siempre y cuando los puntos se encuentren dentro del alcance angular de cada servomotor de  $-90$  a  $90$  grados. Ya que se pueden realizar movimientos en los cuales los servomotores no respondan de una forma apropiada. Teniendo en cuenta el desplazamiento del brazo este puede llegar a una misma posición de diferentes maneras como se muestra en la siguiente imagen (Peña; 2005).

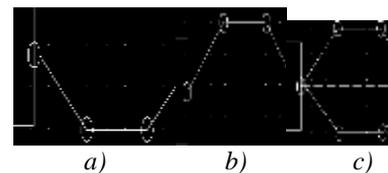


Fig. 8. Trayectorias posibles del brazo para terminar en una misma posición.

Por tanto si se desea que el brazo realice el movimiento de la figura a y pase a realizar una trayectoria como la de la figura 8.b, este no conseguirá hacerlo debido a la restricción de los

grados de libertad que poseen los servomotores, a este cambio de trayectorias con respecto a un punto se le conoce como traslación, por esta razón el brazo robótico se utilizara en un solo lado o una sola trayectoria, por tanto se gira la base 45 grados, con el fin de brindarle más espacio de trabajo al brazo y correcto funcionamiento de los servomotores.

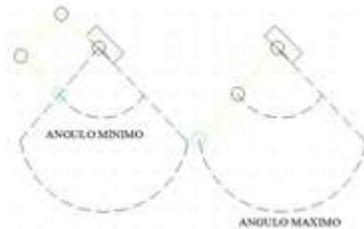


Fig. 9. Espacio de trabajo que puede alcanzar el brazo con la rotación de la base.

Se puede observar la región de trabajo que comprende un valor máximo o mínimo dependiendo de la posición que el brazo tenga sin tener ningún tipo de restricción con los servomotores.

#### 4. CONCLUSIONES

Uno de los parámetros más importantes en el diseño, construcción y calibración del brazo es el espacio de trabajo en el cual va a realizar el dibujo, ya que este está condicionado tanto por las dimensiones del modelo mecánico, como por las dimensiones del filtrado en la nube de puntos. Se realizó un análisis de los datos entregados por la nube de puntos aportados por el sensor Kinect usando reglas geométricas para traducir dichas coordenadas en movimientos mecánicos en el servomotor. El brazo robótico debe tener una calibración correcta que se logra después de varias pruebas con el fin de tener una buena coordinación de las lecturas de la exploración del Kinect y el posicionamiento del brazo.

#### RECONOCIMIENTO

Los autores agradecen a la Universidad Santo Tomas – Tunja, Boyacá, Colombia, En especial a la Facultad de Ingeniería Electrónica y a la Unidad de Investigaciones CIUSTA, quienes apoyan el proyecto Diseño de Hardware y Software para la estimación de patrones en imágenes con fines de esquelización y detección de pose de GIDINT.

#### REFERENCIAS

- Berkeley, U. C., & Urtasun, R. (2010). Combining Discriminative and Generative Methods for 3D Deformable Surface and Articulated Pose Reconstruction, pp. 647–654.
- Castro, Gonzalez & Lengerke (2013). Fuzzy Speed Control for Trajectory Follow-Up of a Robot Moving on Vertical Surface. Revista Colombiana de Tecnologías de Avanzada. Vol. 2, No. 22, pp. 34 – 41.
- Pardo, Gutierrez, Jimenez, Sosa & Martinez (2013). Diseño e Implementacion de un Prototipo de Sistema de Identificacion y Clasificacion de Piezas por Tratamiento Digital de Imagenes con Acceso a Ethernet y Comunicacion Inalambrica. Revista Colombiana de Tecnologías de Avanzada. Vol. 2, No. 22, pp. 49 – 58.
- Peña (2005). Sistema de Visión Artificial para Asistir la Conduccion de una Silla de Rueda. Revista Colombiana de Tecnologías de Avanzada. Vol. 2, No. 22, pp. 34 – 41.
- Demirdjian, D., Ko, T., & Darrell, T. (2003). Constraining Human Body Tracking.
- Fossati, A., Berkeley, U. C., & Fua, P. (2009). Observable Subspaces for 3D Human Motion Recovery, pp. 1137–1144.
- Jiang, M., Tang, J., Chen, L., Shang, T., Gan, Z., Liu, X., & Xu, Q. (2010). Articulated human body pose tracking by suppression based immune particle filter. College of Computer Science and Technology, pp. 3273–3276.
- Liu, F., Ding, G., Deng, X., & Xu, Y. (2008). Vision-based 3D Articulated Pose Tracking Using Particle Filtering and, 965–970. doi:10.1109/SITIS.2007.35
- Oh, C., Islam, Z., & Lee, C. (2010). Pictorial Structures-based Upper Body Tracking and Gesture Recognition.
- Oh, C., Islam, Z., Park, J., & Lee, C. (2010). A Gesture Recognition Interface with Upper Body Model-based Pose Tracking, pp. 531–534.
- Salem, M., Kopp, S., Wachsmuth, I., & Joubin, F. (2010). Generating Robot Gesture Using a Virtual Agent Framework, pp. 3592–3597.
- Sapp, B., Weiss, D., & Taskar, B. (n.d.). Parsing Human Motion with Stretchable Models.
- Sheikh, Y. A. (2008). On the Sustained Tracking of Human Motion.
- Suma, E. A., Lange, B., Rizzo, A. S., & Krum, D. M. (n.d.). FFAST : The Flexible Action and Articulated Skeleton Toolkit, pp. 247–248.
- Takahashi, K., Nagasawa, Y., & Hashimoto, M. (2008). Remarks on Markerless Human Motion

- Capture from Voxel Reconstruction with Simple Human Model, pp. 22–26.
- Tran, C., Trivedi, M. M., Vision, C., & Diego, S. (2008). Human body modeling and tracking using volumetric representation: selected recent studies and possibilities for extensions.
- Wang, Z., & Chung, R. (2010). Articulated Human Body: 3D Pose Estimation using a Single Camera. doi:10.1109/ICPR.2010.918
- Wu, Y., & Huang, T. S. (n.d.). Capturing Articulated Human Hand Motion: A Divide-and-Conquer Approach.