

DESARROLLO E IMPLEMENTACION DE UN CONTROL GESTUAL PARA UN ROBOT DE 4 GRADOS DE LIBERTAD

Olivares Borja Eduardo, Macías Solís Francisco, Fernández Ramírez Arnoldo, García Andrade Roxana¹, Martínez Reyes Armando¹, Rincón Martínez Ernesto

Instituto Tecnológico Nacional de México / Instituto Tecnológico de Nuevo León

División de Estudios de Posgrado e Investigación

Departamento de Ingeniería Eléctrica y Electrónica¹

Av. Eloy Cavazos # 2001 Col. Tolteca C.P. 67170 Guadalupe Nuevo León México

Tel.+ 81 8157 0500 ext 159

olivareseduardo97@gmail.com

RESUMEN.

El presente proyecto de tele-operación describe la implementación de un control gestual usando el sensor Leap Motion, planteando el uso de un sistema que permite gestionar la posición y orientación de un robot de 4 grados de libertad de manera remota con movimientos de las manos. De esta forma será posible generar una interfaz gráfica de usuario más intuitiva e interactuar con el robot donde se encuentre con solo mover alguna extremidad del cuerpo en este caso la mano derecha. Esta interacción se podrá realizar con todos los beneficios de un robot como por ejemplo hacer tareas peligrosas y mayor fuerza.

Así mismo, se analizan proyectos existentes cuyas líneas de investigación se entrelazan con la aquí establecida, además de permitirnos formar las bases conceptuales y poder concretar el proyecto.

Palabras Clave: control gestual, teleoperacion, interacción, robot.

ABSTRACT.

The present teleoperation project describes the implementation of a gestural control using the Leap Motion sensor, proposing the use of a system that allows to manage the position and orientation of a robot with 4 degrees of freedom remotely with manual movements. In this way, it will be possible to generate a more intuitive graphical user interface and interact with the robot where it is with just moving one limb of the body in this case the right hand. This interaction can be done with all the benefits of a robot, such as carrying out dangerous tasks and greater strength. Likewise, the existing projects whose lines of research intertwine with those established here are analyzed, and we are allowed to form the conceptual bases and be able to define the project.

Keywords: gesture control, teleoperacion, interaction, robot.

1. INTRODUCCIÓN

En los últimos años se han incrementado la utilización de dispositivos que permiten la interacción hombre-computadora mediante la interpretación de gestos. En robótica el reconocimiento de gestos es usado por la capacidad de explotar todo el rango de movimiento en 3D para controlar robots de manera simple basada en movimientos que una mano puede realizar (traslación y rotación) [1], [2].

Los diferentes dispositivos actuales en el mercado como *Intellect Motion*, *Kinect*, *MYO*, *Airy*, han encontrado diversas aplicaciones en el campo de la ingeniería ya que existen muchas circunstancias en las cuales no es conveniente emplear personas para la realización de algunas labores, como por

ejemplo en el área de Seguridad e higiene industrial manejar elementos peligrosos o realizar maniobras sin riesgo. Aplicaciones cotidianas, manejar dispositivos electrónicos que conforman nuestra vida diaria, por ejemplo, televisiones, computadoras, consolas de video juegos. Aplicaciones médicas, como la generación de aplicaciones o programas de entrenamiento para personas con discapacidad o capacidades diferentes, exoesqueletos y dispositivos robóticos para rehabilitación física. Aplicaciones de aprendizaje como un sistema interactivo en donde nuestras acciones determinan el trascurso de la acción. Pero la efectividad de algunos dispositivos ya mencionados es poca precisa aumentando que la interacción pueda ser incomoda con el tiempo, teniendo la posibilidad de no reconocer el gesto y fatigar a las personas.

El objetivo de este trabajo es desarrollar un sistema de control gestual de un robot de 4 grados de libertad, que mediante el reconocimiento de gestos corporales permita representar un movimiento, ver figura 1.

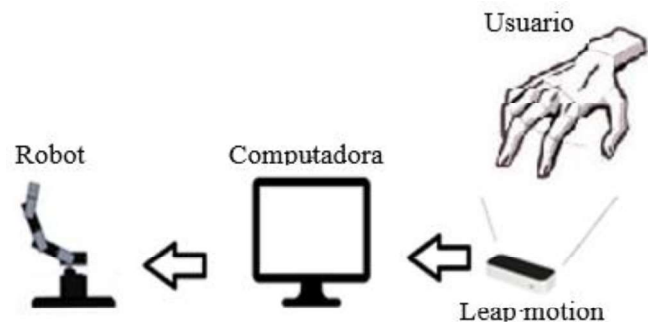


Figura 1 Esquema general de la aplicación.

Este trabajo está organizado de la siguiente manera, en la sección 2 se detalla todos los antecedentes de la investigación, posteriormente en la sección 3 se observará la metodología utilizada para el diseño, el desarrollo y la integración del sistema que describen los dispositivos que componen la arquitectura del sistema. En la sección 3, se realiza la validación de pruebas, aquí se validará si el robot es capaz de realizar un movimiento a través de lenguaje corporal. Se finaliza con las conclusiones de la aplicación desarrollada y las referencias consultadas.

2. ESTADO DEL ARTE

En esta sección se llevó a cabo el estudio de los artículos publicados en años anteriores sobre los temas tales como: dispositivos de movimiento, Leap Motion, robótica, interacción hombre – computadora, donde han participado diferentes autores que trabajaron y analizaron las distintas aplicaciones y los métodos de interacción para el desarrollo de las mismas, buscando que fueran más intuitivas y que ayuden a los sistemas de control gestual.

De esta manera, en [1] se habla sobre la investigación que impulsa la búsqueda de la mejora de técnicas basadas en sensores, que utilizan los datos como entrada para algunos algoritmos que puedan estimarse en tiempo real tanto la orientación y posición de los miembros del cuerpo humano.

Por otra parte, en [2], se discute sobre la interacción hombre – computadora, aplicando un sistema de control gestual para un robot de 4 grados de libertad donde se describen los dispositivos que componen la arquitectura física del sistema, el desarrollo y modelos geométricos para calcular la cinemática directa e inversa y la estructura de comunicación de la lógica del sistema. De igual forma en [3], las interfaces humano-máquina y la evolución constantemente son presentadas. Dentro de este grupo, las interfaces basadas en gestos naturales de los usuarios que permiten convertir estos movimientos en comandos para un sistema informático, sin contacto con ninguna superficie.

Encontramos en [4], un desarrollo que presenta la aplicación del brazalete mioeléctrico (*Myo*) para el control de movimiento de una mano robótica. La adquisición y el procesamiento de los datos se realizan utilizando el programa *MATLAB* y la aplicación *Classification Learner*, los datos procesados son utilizados para mover servomotores que a su vez accionan los dedos de la prótesis con una tarjeta *Arduino uno*, pero según [5], indica en su trabajo que la eficiencia de algunos dispositivos de movimiento, dentro de los cuales, se encuentra el brazalete mioeléctrico (*Myo*) donde se comprobó que tenía una eficiencia del 80% en el reconocimiento de gestos, comparados con otros dispositivos como *Leap Motion* y *Kinect*. [6] Este artículo presenta una interfaz de usuario de objetos 3D (tridimensionales) utilizando el dispositivo *Leap Motion* como un sistema de capacitación médica, por otra parte [7] propone un diseño de mano de robot antropomórfico utilizando veinte servomotor para veinte grados de libertad que puede reproducir el gesto exacto para tareas diestras como cirugías llevándolo todo a una simulación sin implementarlo a un sistema mecatrónico.

3. METODOLOGIA.

3.1. Diseño.

En la figura 1 se muestra un esquema general de la aplicación. La integración de la arquitectura de sistema se hace bajo la plataforma de Windows, *Arduino* y *Processing*. Se utilizan cada una de las herramientas de software proveídas por el fabricante.

- Sensor de movimiento *Leap Motion*.

Es un pequeño dispositivo con dimensiones muy reducidas, tan solo mide 8 cm de largo, 3 cm de ancho y 1.5 cm de alto, Como se puede ver en la (figura.2), *Leap Motion* cuenta con dos cámaras, tres leds y un microcontrolador y su campo de visión es de 150°. Este dispositivo obtiene la posición, orientación y velocidad del objeto rastreado a una velocidad de hasta 200 cuadros por segundo.

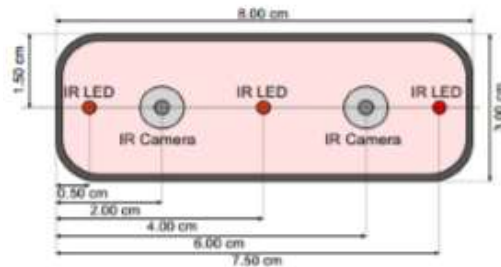


Figura 2 Representación esquemática de *Leap Motion Controller*.

Leap Motion cuenta con su propio SDK la cual permite acceder a las diferentes funciones que retornan los datos obtenidos por el sensor. Debes saber que *Leap Motion* contiene varias librerías para programar en los siguientes lenguajes de programación y plataformas de desarrollo: Java, Python, C++, C#, Unity, JavaScript.

- *Arduino Uno*.

Es una placa electrónica basada en el microcontrolador *ATmega328*. Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas. Además, incluye un resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteo, ver figura.3.

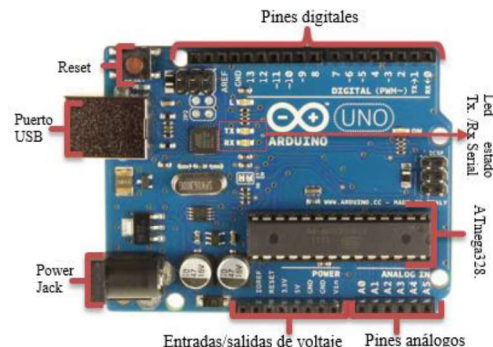


Figura 3 Ilustración representativo de *Arduino*

- Robot



Figura 4 Mini-robot antropomórfico de cuatro grados de libertad.

- Actuadores GO TECK.

Dispositivo que tiene un sistema de corrección con retroalimentación para el control de un parámetro específico, ver figura 5, los datos técnicos se muestran en la Tabla.1



Figura 5 Actuador GO TECK.

Tabla 1 Especificaciones técnicas del actuador.

Datos técnicos:	
Dimensiones(LxWxH)=	22.0x11.5x27mm (0.86x0.45x1.0 in)
Peso	13 gramos
Peso con cable y conector	10.6 gramos
Torque 4.8 volts	16.7oz/in o 1.2kg/cm
Voltaje de operación	4.0 a 7.2 volts
Velocidad de giro a 4.8 volts	0.12seg/60°

3.2. Modelado cinemático directo de posición.

El modelo directo, es la relación que permite determina el vector “x” de coordenadas de operacionales del robot correspondiente a una configuración dada q. Este modelo se expresa como en la siguiente figura 6:

$$\mathbf{x} = \mathbf{f}(\mathbf{q})$$

Figura 6 ecuación de movimiento

A continuación, se muestra las tablas de parámetros de D-H del mini-robot.

Tabla 2 Tabla de parámetros de D-H para el mini robot

Eslabón i	α_i	d_i	θ_i	r_i
1	0	0	θ_1	0
2	90°	0	θ_2	0
3	0	D3	θ_3	0
4	0	D4	θ_4	0
5	0	D5	0°	0

En la figura 7 se muestra las matrices de trasformación del mini-robot para $q_1=q_2=q_3=q_4=q_5=0^\circ$, D3= 4mm, D4= 4mm y D5= 3.5mm.

$${}^0T_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, {}^1T_2 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^2T_3 = \begin{pmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, {}^3T_4 = \begin{pmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^4T_5 = \begin{pmatrix} 1 & 0 & 0 & 3.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figura 7 matrices de trasformación

3.3. Desarrollo e Integración del sistema.

A partir de la información obtenida se desarrolló un algoritmo que permite darle movimiento al robot a través de señales de la mano. El sistema fue desarrolla en una computadora de 64 bits con sistema operativo *Windows* donde se conecta el dispositivo *Leap Motion* y la placa electrónica *Arduino uno*, ver figura 8. Cada actuador del robot se conecta a los pines digitales de la placa *Arduino uno*.

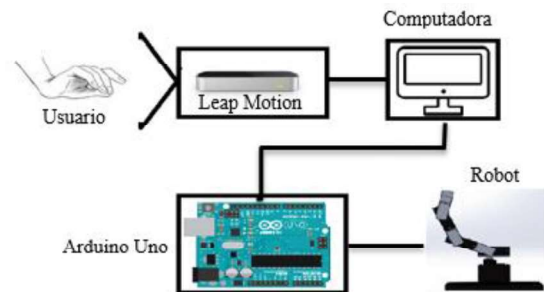


Figura 8 Esquema del sistema físico.

El movimiento de los servomotores, se realiza cuando se ejecuta el sketch de *Processing* y el sensor de movimiento detecta la mano, haciendo uso de la tarjeta *Arduino uno* se decide el número de grados que debe girar cada actuador y se envían las señales a cada uno de los servomotores para que alcancen la posición deseada, usando una señal PWM. A continuación, se muestra en la figura 8, el diagrama de flujo de la aplicación desarrollada en *Arduino* y en las Tablas 3 y 4 en la siguiente página, se observan fragmentos del código desarrollado.

velocidad de las manos, el movimiento de izquierda a derecha o viceversa da lecturas diferentes.[8]

Tabla 3 Código de extensión y flexión.

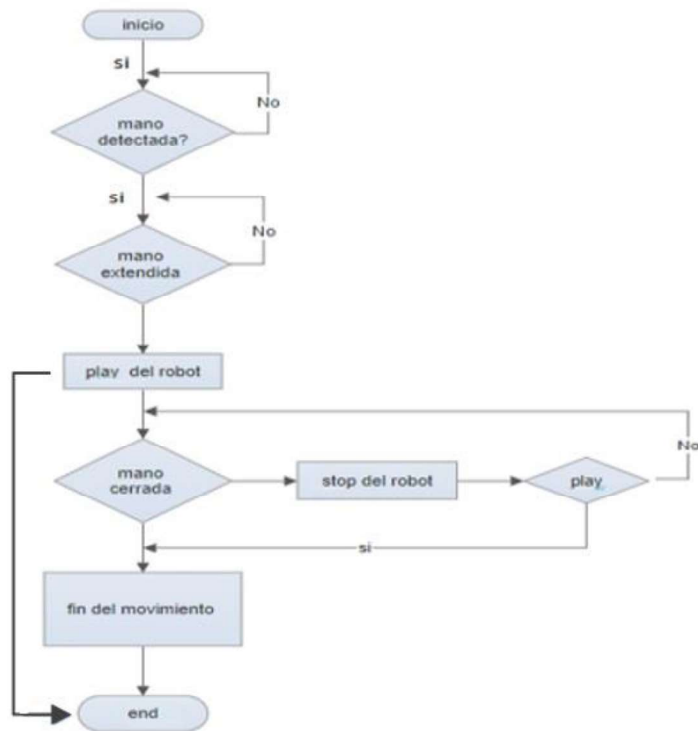


Figura 8 Diagrama de flujo del programa.

El tipo de gesto utilizado tiene el nombre de “Swipe”. Este es un movimiento en línea de la mano con los dedos extendidos.

Este tipo de movimiento se detecta por dedo, por tal motivo, esto te permite detectar si la mano completa se desliza de derecha a izquierda o viceversa, de igual manera de arriba hacia abajo o viceversa.

Los casos anteriores solo hacen mención a dos dimensiones, en el plano “X” (derecha - izquierda) y el plano “Y” (arriba - abajo).” Mediante la siguiente función llamada “posición ()”. En este caso, para diferenciar el tipo de deslizamiento se pueden comparar los valores de “X” o “Y” en sus respectivos valores absolutos, además entre estos mismos para saber si el desplazamiento fue horizontal o vertical, y se puede observar en dentro de la interfaz.

La información de cada gesto, viene dado por el tipo de información que entran a las cámaras. Tenemos que la

```

#include <VarSpeedServo.h>
VarSpeedServo falange1;
VarSpeedServo falange2;
VarSpeedServo falange3;

int num;
int mov;

// constructores
void setup () {
  Serial.begin(9600);
  falange1.attach(9);
  falange1.write(85,150) ;// motor de abajo

  falange2.attach(10) ;//motor del medio
  falange2.write(150,150);

  falange3.attach(13) ;// motor de arriba
  falange3.write(90,150);
;
}

// funciones miembro
void loop () {
  if (Serial.available()>0) {
    num = Serial.read();
    if (num >=0 && num <110) {
      mov = map (num,0, 109,90, 110);
      falange1.write(mov);
    }
    if (num >=0 && num <150) {
      mov = map (num, 0, 149, 90, 150);
      falange2.write(mov);
    }
    if (num >=0 && num <166) {
      mov= map (num, 0, 165, 80, 166);
      falange3.write(mov);
    }
  }
}

```


Tabla4 Código de Abducción y Aducción.

```
#include <VarSpeedServo.h>
VarSpeedServo falange4;
int num;
int mov;

// constructores

void setup () {
    Serial.begin(9600);

    falange4.attach(8) ;// motor de base
    falange4.write(0);
}

// funciones miembro

void loop () {
    if (Serial.available() > 0) {
        num = Serial.read();
        if (num >= 0 && num < 110) {
            mov = map (num, 0, 109, 90, 110);
            falange4.write(mov);
        }
    }
}
```

En el código del programa *Arduino*, se incluye la librería *VarSpeedServo.h*, esta biblioteca permite que una placa *Arduino uno* controle la velocidad y posición de los servomotores.

Se declara un objeto o variable de la librería del *VarSpeedServo.h* y se hace uso de los métodos *attach*, que permite indicar en que pin digital se tiene conectado el servo, y por otro lado el método *write*, el cual indica el ángulo en que se desea posicionar los servomotores, además de la velocidad que se desea adquirir.

Después en el void loop, se declara la condición que debe verificarse y la ejecuta, utilizando previamente la función *map* (), que permite re-mapear valores de un rango de entrada, a valores de un rango de salida.

La interface consta de una ventana, la cual incluye un área gráfica que muestra la simulación del robot de cuatro grados de libertad, representada con polígonos regulares. Aquí se puede ver el robot en 3 dimensiones además de acceder a una página de internet que detecte las manos por medio de un botón que se encuentra en la parte superior izquierda de la ventana con tan solo accionar el mouse. Ver figura 9.

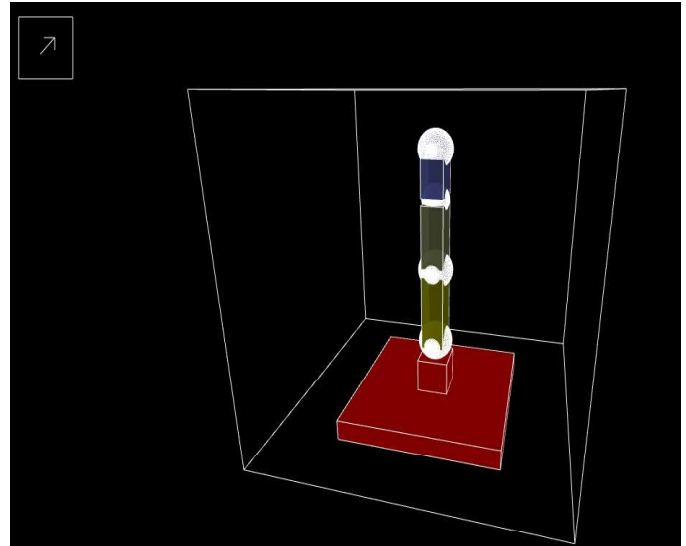


Figura 9 Interface gráfica del usuario.

4. VALIDACIÓN Y PRUEBAS.

El operador se posiciona frente el sensor de movimiento *Leap Motion*. Se realiza los primeros movimientos del robot cuando el operador tenga la palma de la mano abierta frente el campo de visión del dispositivo y el sujeto empiece a mandar gestos que por medio de un algoritmo el dispositivo reconoce: arriba, abajo, izquierda, derecha, puño cerrado. Empezando a mandar señales a la placa de programación *Arduino uno* obteniendo la posición que deben realizar los 4 servomotores para que el robot genere los movimientos específicos, ver figura.10, y figura. 11.

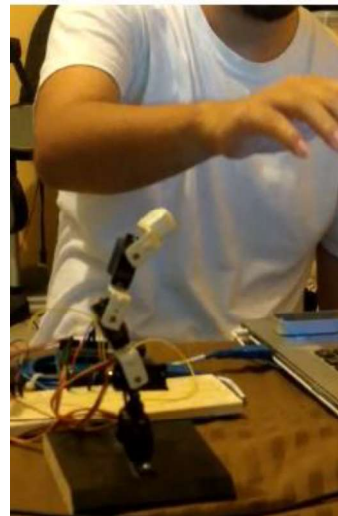
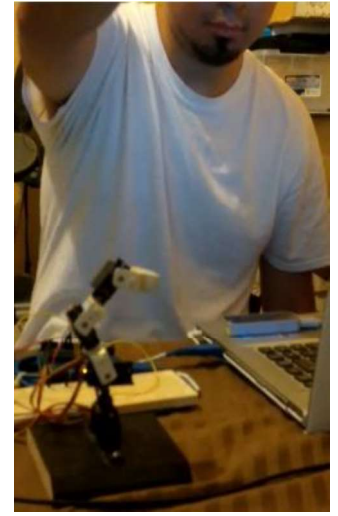


Figura10 A) Extensión



B) Flexión

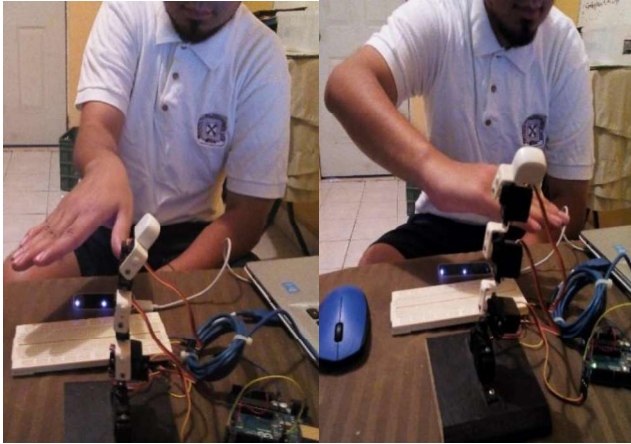


Figura 11 A) Aducción B) Abducción

5. CONCLUSION

El proyecto surge de la necesidad de la interacción humano – robot, donde sea demostrado que existe cierta discrepancia hacia los robots. esta investigación busca reducir ésta dificultades de operación, implementando un sistema de control para manipular un robot de 4 grados de libertad sin utilizar dispositivos de mando manual, solo mediante el reconocimiento visual de movimientos de extremidades superiores del operador como medio de comunicación brindando facilidades de operación a las personas que deban operar a distancia por aspectos de seguridad.

En conclusión, se puede lograr la realización de aplicaciones intuitivas, que puedan lograr generar una interacción de tele-presencia entre receptor y emisor empleando cualquier tipo de protocolo multimedia.

El trabajo futuro abordará la calibración conjunta de dos dispositivos de movimiento con el fin de calcular nuevas características basadas en la combinación de las posiciones 3D y una mejora al diseño de robot de cuatro grados de libertad.

Este trabajo ha sido desarrollado con financiamiento del consejo nacional de ciencia y tecnología, CONACYT en el Tecnológico Nacional de México / Tecnológico de Nuevo León.

6. REFERENCIAS

- [1] E. W. G. Clua and I. C. Medialab, "Sensor Data Fusion for full Arm Tracking using Myo Armband and Leap Motion Sensor Data Fusion : Kalman Filter," pp. 3–8, 2015.
- [2] O. Justinico, P. Cárdenas, and J. S. Rodríguez, "Control gestual de robot de 4 GDL con sensor Leap Motion," *Congr. Int. 206 Ing. Mecatrónica Y Autom.*, no. 1, pp. 206–214, 2015.
- [3] P.-J. Gonzalo and A. Holgado-Terriza Juan, "Control of home devices based on hand gestures," *2015 IEEE 5th Int. Conf. Consum. Electron. - Berlin*, pp. 510–514, 2015.
- [4] M. Martínez, Vargas-Pérez, E. Gómez-Merlín, M. Arias-Montiel, E. Lugo-González, and R. Miranda-Luna, "Control de Movimiento de una Mano Robótica Mediante Señales Electromiográficas," pp. 73–76, 2016.
- [5] L. Vokorokos, J. Mihal'ov, and E. Chovancova, "Motion sensors: Gesticulation efficiency across multiple platforms," *INES 2016 - 20th Jubil. IEEE Int. Conf. Intell. Eng. Syst. Proc.*, pp. 293–298, 2016.
- [6] M. A. Seif, R. Umeda, and H. Higa, "Training System Using Leap Motion," pp. 159–162, 2017.
- [7] L. Heisnam and B. Suthar, "20 DOF Robotic Hand for Tele-operation : - Design , Simulation , Control and Accuracy test with Leap Motion," pp. 5–9, 2016.
- [8] E. Juan, "Control por gestos usando Leap Motion Instituto Tecnológico y de Estudios," 2016.
- [6] A. B. M. Anigstein, "Implementaci ´ on de un Control Gestual para Robots," pp. 1–6.
- [7] Y. Xu, C. Yang, P. Liang, L. Zhao, and Z. Li, "Development of a hybrid motion capture method using MYO armband with application to teleoperation," *2016 IEEE Int. Conf. Mechatronics Autom. IEEE ICMA 2016*, pp. 1179–1184, 2016.
- [8] W.-J. Li, C.-Y. Hsieh, L.-F. Lin, and W.-C. Chu, "Hand Gesture Recognition for Post-stroke Rehabilitation Using Leap Motion," no. c, pp. 978–1, 2017.
- [9] S. Pareek, "Development of CAD Interface Using Leap Motion."
- [10] P. Ponsa Asensio, C. Urbina, C. Manresa-Yee, and R. Vilanova Arbós, "FAZ revista de diseño de interacción.," *Faz*, no. 8, pp. 99–119, 2015.
- [11] J. L. Montañés, "Development of a Robotic Arm and implementation of a control strategy for gesture recognition through Leap Motion device .," 2016.
- [12] Arduino, "Aurduino Mega 2560," <https://www.arduino.cc/>, vol. XXXIII, no. 2. pp. 81–87, 2014.
- [13] "Libraries \ Processing.org." [Online]. Available: <https://processing.org/reference/libraries/>. [Accessed: 26-May-2018].
- [14] "Leap Motion." [Online]. Available: <https://www.leapmotion.com/en/>. [Accessed: 13-Jun-2018]