Recibido: 7 de septiembre 2018. Aceptado: 2 de noviembre 2018.

Sistema de control de mano robótica replicadora de movimientos para manipular sustancias químicas

Control system for robotic hand to replicate movements to manipulate chemical substances

Natalia Leonila Quiroga Pérez & Edwin Calla Durandal

Departamento de Ciencias Exactas e Ingeniería, Universidad Católica Boliviana San Pablo.

natyquiroga_p@hotmail.com

Resumen: El presente proyecto ayuda al desempeño en laboratorios químicos, con el fin de reducir accidentes laborales y enfermedades a corto y largo plazo. El proceso de realización del robot consta de cuatro partes fundamentales. El sistema mecánico, el sistema electrónico, el sistema de reconocimiento de movimiento de manos y el sistema de control.

El objetivo de este proyecto es realizar un sistema de control para el robot replicador de movimientos para manipular sustancias químicas. Por lo tanto, se analizó los movimientos que puede realizar una mano, concluyendo que existen cuatro tipos de sujeción fundamentales. El modelo matemático de cada tipo de agarre se determinó con ayuda del *software* PIPIMC y Matlab. Por lo tanto, se logró estimar las funciones de transferencia y realizar los cuatro controladores proporcionales – integrales IMC (PI-IMC), a partir de la técnica de control IMC (*internal model controller*). Finalmente, el controlador fue incluido en el programa principal del robot. De esta manera se mejoró la precisión y tiempo de respuesta del robot.

Palabras clave: Robot, mano, sistema de control, controlador, replicar, PI-IMC.

Abstract: This project helps performance in chemical laboratories, in order to reduce work accidents and diseases in the short and long term. The process of realization of the robot consists of four fundamental parts. The mechanical system, the electrical system, the hand movement recognition system and the control system.

The objective of this project is to make a control system for the robot replicator of movements to manipulate chemicals. Therefore, we analyzed the movements that can be made by a hand, concluding that there are four fundamental types of subjection. The mathematical model of each type of equipment was determined with the help of the PIPIMC and Matlab software. Therefore, the transfer functions are calculated and perform the four proportional - integral IMC controllers (PI-IMC), from the IMC control technique (internal model controller). Finally, the controller was included in the robot's main program. This improved the accuracy and response time of the robot.

Key words: Robot, hand, control system, controller, replicate, PI-IMC.

1 Introducción

La ingeniería se encarga de entender y controlar los materiales y fuerzas de la naturaleza en beneficio de la humanidad. Es por eso que los ingenieros se preocupan de comprender y controlar segmentos de su entorno más comúnmente llamados sistemas de control, con el fin de proporcionar productos económicos y útiles para la sociedad. El control efectivo de los sistemas requiere entendimiento y modelamiento, se deben considerar paralelamente el control de sistemas conocidos, como los sistemas robóticos como base teórica (BISHOP & DORF, 2008).

Desde hace algunos años, los robots son utilizados para realizar trabajos especialmente peligrosos, moverse por zonas inhóspitas, contacto con ambientes nocivos y peligrosos para seres humanos (CONSEJO DE SEGURIDAD NUCLEAR, 2014). Un ejemplo claro son los manipuladores robóticos capaces de realizar gran variedad de tareas a distintas velocidades y precisiones, en muchos casos superan a los operadores humanos. Son usados en procesos de fabricación como la soldadura por puntos o administración de fármacos, para realizar las tareas con fiabilidad y precisión (FADALI, 2009).

Según la asociación mundial de robótica, un robot industrial es un manipulador reprogramable multifuncional diseñado para mover materiales, partes, herramientas o dispositivos especiales (ALVAREZ, 1999). Los productos químicos poseen un riesgo potencial para la salud, por fugas o derrames, ya sea por una situación accidental en las plantas de procesamiento o por errores humanos (MARCO DE RESPUESTA A EMERGENCIAS, 2013). Se estima que durante 1998 y 2007, cerca de 3 200 accidentes tecnológicos afectaron a dos millones de personas, con un aproximado de 100 000 pérdidas humanas (WORLD HEALTH ORGANIZATION, 2009).

En el presente trabajo se propone un sistema de control de mano robótica replicadora de movimientos para manipular sustancias químicas, entre otros elementos electrónicos y mecánicos para su funcionamiento, además de realizar la descripción del *software* y *hardware* de apoyo para el funcionamiento.

2 Identificación del Problema

En Bolivia según el ministerio de trabajo, empleo y previsión social con datos hasta el 30 de junio de 2012, se tiene que 7 384 personas en promedio sufrieron accidentes laborales entre los años 2009 y 2012. Existen muy pocos datos acerca de accidentes con productos químicos en el país, por lo tanto, de acuerdo con el Ministerio de Ambiente y Desarrollo Sostenible de Colombia, el año 2009, 19 723

personas sufrieron de alguna clase de intoxicación por sustancias químicas. Un total de 4 578 personas tuvieron una emergencia relacionada con tóxicos.

En el país, la industria química es muy limitada y no se cuenta con programas para hacer frente a los accidentes que podrían ocurrir en exposición con sustancias químicas. Por ello se considera importante realizar diferentes acciones como nuevos equipos de manipuleo (NACIONES UNIDAS PARA LA FORMACIÓN PROFESIONAL E INVESTIGACIÓN, 2008).

3 Objetivo

El objetivo de la investigación es implementar un sistema de control para un robot replicador manipulador de sustancias químicas comandado por un sensor de visión y sensores resistivos de fuerza. Debido a que diariamente se manipulan gran variedad de sustancias peligrosas en laboratorios, durante esta manipulación ocurren accidentes y la exposición puede causar daños a corto y largo plazo. Los elementos más peligrosos son los explosivos o inflamables, ácidos y peróxidos, como ser el ácido nítrico, ácido fluorhídrico, ácido tricloroacético, ácido clorhídrico, peróxido de hidrógeno, entre otros.

El prototipo ayuda a evitar el contacto directo con sustancias tóxicas, evitando accidentes y enfermedades a corto y largo plazo. Por otro lado, el precio de un robot con estas características asciende a los miles de dólares, el diseño y la tecnología a ser usada en el presente proyecto, permite reducir el costo haciéndolo significativamente más económica y accesible, sin perder la fiabilidad o eficiencia del mismo.

4 Sustento teórico

4.1 Sustancias químicas y enfermedades

Existe una gran variedad de sustancias químicas, cada una tiene propiedades y composiciones diferentes (PATNAIK, 2007). Las sustancias peligrosas son elementos los cuales al ser liberados son capaces de afectar a la salud pública o al medio ambiente. Esta peligrosidad se debe a alguna propiedad corrosiva, reactiva, explosiva, tóxica o inflamable. (YARTO, IZE, & GAVILÁN, 2003).

Los tóxicos pueden dañar uno o más órganos específicos del cuerpo, que incluyen el hígado, los riñones, las vías respiratorias, el sistema inmunitario, la piel, el sistema nervioso, los ojos, el sistema reproductor y los sistemas cardiovasculares (PATNAIK, 2007). Un carcinógeno es una sustancia capaz de causar cáncer en mamíferos. Los productos químicos, la radiación y ciertos virus son agentes causantes de cáncer. Las rutas de exposición, la cantidad de dosis, la absorción y la distribución del producto químico dentro del cuerpo (PATNAIK, 2007).

4.2 Identificación de sistemas

En un proceso de control, un sistema se denota como un objeto o una colección de objetos, que tiene un objetivo y cuyas propiedades son estudiadas. En la Figura 1: se muestran los principales elementos que componen un sistema de control. Donde u(.) es considerada la entrada del sistema, h(.) es el sistema e y(.) es la salida (RODRIGUES & DOS SANTOS, 2004).

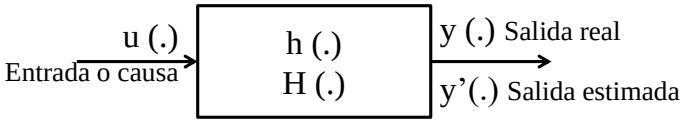

Figura 1: Componentes de un sistema de control

Para la identificación del sistema h(.), es necesario conocer la entrada u(.) y la salida y(.), de modo que y'(.) tienda a y(.). Donde y(.) es la salida real del sistema o la leída e y'(.) es la salida estimada (RODRIGUES & DOS SANTOS, 2004).

4.3 Modelo matemático de un sistema

Los modelos matemáticos de los sistemas físicos son fundamentales en el diseño y análisis de los sistemas de control. El comportamiento dinámico generalmente se describe mediante ecuaciones diferenciales ordinarias ya sea para sistemas mecánicos, eléctricos, hidráulicos, etc. debido a que los sistemas físicos no son lineales, se realizan aproximaciones mediante la transformada de Laplace. Posteriormente, se obtiene la relación entre la entrada y la salida en forma de función de transferencia (BISHOP & DORF, 2008).

Se entiende por modelaje e identificación a la determinación del modelo matemático de un sistema representado por sus aspectos esenciales de forma que se enfoque a una utilización en particular. Se divide en los siguientes subprocesos (RODRIGUES & DOS SANTOS, 2004).

- Análisis físico-matemático: se basa en las leyes físicas que caracterizan un sistema en particular.
- Análisis experimental: se basan en las medidas y observaciones del sistema.

Estos subprocesos facilitan la obtención de modelos matemáticos de un sistema. El propósito de los modelos matemáticos son en particular, la previsión de estados futuros, análisis y diseño del sistema de control para proporcionar algoritmos de control adaptativo y predictivo, supervisión para analizar las características operacionales del sistema y optimización de los sistemas (RODRIGUES & DOS SANTOS, 2004).

4.4 Identificación experimental por computadora

El modo de identificación del modelo tipo *off-line,* recolecta una cierta cantidad de datos de entrada y salida del sistema, que están almacenados. Después de la recolección, estas medidas son procesadas para estimar los parámetros del modelo. Por otro lado, la identificación *on-line,* no almacena ningún dato del proceso, un algoritmo de estimación recursivo es utilizado para ajustar los parámetros en cada periodo de muestreo, el tiempo de procesamiento del algoritmo necesario para ajustar el modelo está en función al periodo de muestreo (RODRIGUES & DOS SANTOS, 2004).

4.5 Modelo de un proceso de primer orden

Un modelo paramétrico de dinámica de un proceso común está caracterizado por la función de transferencia mostrada en la ecuación (1). Donde podemos identificar a K como la ganancia, τ es la constante de tiempo y θ es el atraso (RODRIGUES & DOS SANTOS, 2004).

$$G_p(s) = \frac{K}{\tau s+1} e^{-\theta s} = \frac{Y(s)}{U(s)} \qquad (1)$$

La respuesta de primer grado de un modelo matemático está representada por la ecuación (2). Con esta base en la respuesta temporal es posible dimensionar puntos para la aplicación de diferentes métodos de estimación. Los métodos más usados para determinar los parámetros de la ecuación (1), son Ziegler Nichols y Hagglund (RODRIGUES & DOS SANTOS, 2004).

$$y(t) = K\,(1-e^{-(T-\theta)/\tau}) \qquad (2)$$

4.6 Los métodos de PIPIMC

PIPIMC es una herramienta computacional, la cual sirve para la enseñanza de la identificación del modelo FOPDT (*First-Order Plus Dead-Time*) y proyectar el controlador PI-IMC (*Proportional Integral -Internal Model Controller*). PIPIMC utiliza distintos métodos para hallar los valores de K, τ y θ de la ecuación (1), como ser: Ziegler-Nichols, Hagglund, Smith-Corripio, Sundaresand-Krishnawamy y el método de áreas o Nishikawa (ARAÚJO y otros, 2015).

Para poder la herramienta PIPIMC es necesario que el sistema tenga las siguientes características: ser estable y convergente en estado estacionario (ARAÚJO y otros, 2015). Los valores del MVI (*Model Validation Index*) y del MCI (*Model Caracterization Index*), determinan: la consistencia del sistema y la característica del modelo de proceso y se calculan mediante las ecuaciones (3) y (4). Donde Y(t) es el valor real del vector de salida, $Y_s(t)$ es el valor simulado del vector de salida. Se debe considerar un valor de MVI > 0,9 para demostrar un sistema consistente. Por otro lado, para el MCI se divide en: $0 \leq MCI \leq 0{,}2$ para *lag-dominated dynamic*, $0{,}2 < MCI$

< 0,7 para *balanced* y por último 0 ,7≤ MCI ≤ 1 para *delay-dominated* (ARAÚJO y otros, 2015).

$$MVI = 1 - \frac{|Y_s(t) - Y(t)|}{|Y(t)|} \qquad (3)$$

$$MCI = \frac{\theta}{\tau + \theta} \qquad (4)$$

En la Tabla 1, se muestran las reglas de sintonización PI-IMC para los modelos simplificados que utilizan la función de transferencia FOPDT (ARAÚJO y otros, 2015).

Tabla 1. PI-IMC *tuning* mediante modelos simplificados

Modelo de 1er Orden	PI-IMC Tuning	
$\frac{K}{\tau s+1} e^{-\theta s} \approx \frac{K(-\theta s+1)}{\tau s+1}$	$K_c = \frac{\tau}{K(\lambda + \theta)}$	$T_i = \tau$
$\frac{K}{\tau s+1} e^{-\theta s} \approx \frac{K}{\tau s+1}$	$K_c = \frac{\tau}{K \lambda}$	$T_i = \tau$
$\frac{K}{\tau s+1} e^{-\theta s} \approx \frac{K}{(\tau+\theta)s+1}$	$K_c = \frac{\tau+\theta}{K \lambda}$	$T_i = \tau + \theta$
$\frac{K}{\tau s+1} e^{-\theta s} \approx \frac{K}{(\tau+\theta/2)s+1}$	$K_c = \frac{2\tau+\theta}{2 K \lambda}$	$T_i = \tau + \theta/2$
$\frac{K}{\tau s+1} e^{-\theta s} \approx \frac{\bar{K}}{s} e^{-\theta s} \approx \frac{\bar{K}}{s}(-\theta s+1)$ $\bar{K} = \frac{K}{\tau}$	$K_c = \frac{2\tau+\theta}{\bar{K}(\lambda + \theta)^2}$	$T_i = 2\lambda + \theta$

Para ajustar la constante de tiempo λ se procede de la siguiente manera: $\lambda > 0{,}1\tau$ o $0{,}8\theta$ en caso de contar con la respuesta de bucle con 10% a 15% de sobre-impulso, $\lambda > 0{,}5\tau$ o 4θ en caso de contar con la respuesta de bucle sin sobre-impulso, según el Chien y Fruehauf $\theta < \lambda < \tau$ y según Skogestad $\lambda = \theta$.

4.7 Controladores

La mayoría de los procesos encontrados en la práctica tienen carácter estocástico. Los cambios de parámetros son causados por cambios en el proceso, en la naturaleza de los materiales de entrada, combustible, desgaste, etc. es por eso que la técnica de autoajuste de los parámetros de control es indispensable (BOBAL, BOHM, FESSL, & MACHACEK, 2005).

Alrededor del 90% de los controladores actualmente utilizados en la industria son controladores de tipo PID porque, siempre que estén bien ajustados, muestran resultados de control muy buenos, son fáciles de usar porque son simples,

generalmente bien conocidos y fáciles de implementar (BOBAL, BOHM, FESSL, & MACHACEK, 2005).

4.7.1 Controladores PID

Los miembros de la familia de controladores PID, incluyen tres acciones: proporcional (P), integral (I) y derivativo (D). Los controladores más usados con esta denominación son: P, I, PI, PD y PID. A continuación, se muestra la definición del PI y el PID (MAZZONE, 2002).

El PI o acción de control proporcional-integral, se define mediante la ecuación (5). Donde Ti se denomina tiempo integral y es quien ajusta la acción integral. La función de transferencia resultante se expresa en la ecuación (6) (MAZZONE, 2002).

$$u(t) = K_c e(t) + \frac{K_c}{T_i}\int_0^t e(\tau)d\tau \qquad (5)$$

$$C_{PI}(s) = K_c(1+\frac{1}{T_i\, s}) \qquad (6)$$

Con un control proporcional, es necesario que exista error para tener una acción de control distinta de cero. Con acción integral, un error pequeño positivo siempre dará una acción de control creciente, y si fuera negativo será decreciente. Este razonamiento sencillo muestra que el error en régimen permanente será siempre cero. Muchos controladores industriales tienen solo acción PI. Se puede demostrar que un control PI es adecuado para todos los procesos donde la dinámica es esencialmente de primer orden. Para el PI, la ley de control discretizada es la mostrada en la ecuación (7), donde q_0 y q_2 son parámetros constantes calculados en la discretización (MAZZONE, 2002).

$$u(k) = u(k\text{-}1) + q_0 e(k) + q_2 e(k\text{-}1) \qquad (7)$$

4.8 Robots manipuladores

Un robot manipulador es un brazo articulado, es un elemento programable multifuncional diseñado para mover objetos, piezas, materiales, etc. El sistema mecánico está compuesto por distintas articulaciones, las principales son el brazo y el efector terminal (OLLERO, 2001).

Para realizar tareas con precisión y fiabilidad, las posiciones y velocidades del manipulador se controlan digitalmente. Todos los movimientos son coordinados por una computadora supervisora para lograr los resultados deseados. Los algoritmos de control se descargan de la computadora de supervisión a los microprocesadores (FADALI, 2009).

Los robots de laboratorio son robots colaborativos, o cobots y están destinados a la interacción con el ser humano. Sus funciones principales son: evitar lesiones ergonómicas y realizar trabajos de forma precisa y rápida (PESHKIN, y otros, 2001).

4.9 Cylon

Cylon es un *framework* JavaScript, osea una plataforma basada en JavaScript especialmente diseñado para la robótica, la computación física y el internet de las cosas. Soporta un gran número de protocolos y dispositivos, cuenta con una API sencilla desde la cual se puede manejar un *hardware* como Arduino, Beaglebone, entre otros (VEGA, 2016). Cylon usa el protocolo firmata para comunicarse con Arduino va través del puerto serie. Hace uso de Node.js para ejecutar órdenes a través del puerto serie. Establece una comunicación maestro-esclavo, donde Arduino hace todo aquello que el PC le manda (VEGA, 2016).

4.10 Arduino

Arduino es una herramienta que permite controlar el mundo físico desarrollado por medio de programas en un computador. Consta de una placa que incluye un microcontrolador y un entorno de desarrollo de *software* que permite comunicación con el computador y otros programas de interés. Arduino tiene gran variedad de productos y modelos que pueden ser utilizados de acuerdo a la magnitud de cada proyecto a ser realizado. En general cada placa Arduino tiene las especificaciones técnicas generales para su uso mostradas en la Tabla 2 (ARDUINO, 2017).

Tabla 2. Especificaciones técnicas de Arduino

Microcontrolador	ATmega328
Voltaje de operación	5 Voltios
Voltaje de entrada (recomendado)	7-12 Voltios
Voltaje de entrada (limites)	6-20 Voltios
Pines digitales de entrada y salida	14 (6 tienen salida PWM)
Pines de entrada analógica	6
Corriente DC para pines de entrada y salida	40 mA
Corriente DC para pines el pin de 3.3 V	50 mA
Velocidad del reloj	16 MHz

4.11 Sensor resistivo de fuerza (FSR)

El sensor resistivo de fuerza FSR (*Force Sensitive Resistor*) es una película de polímero (PTF) el cual reacciona resistivamente frente a la fuerza aplicada sobre la superficie activa, la resistencia es inversamente proporcional a la fuerza aplicada (CARRION, OCHOA, & VALVERDE, 2009).

Las características más importantes del FSR-402 según su hoja de datos son las siguientes:

- Tamaño: 12,7 [mm] de diámetro de área activa por 0,05 [mm] de espesor.
- Rango de resistencia: sin presión a 100 [KΩ] y 200 [Ω] a presión máxima.
- Rango de fuerza: 0 a 20 lb. (0 a 100 [Newtons]).
- Fuente de alimentación: Utilizar menos de 1 [mA] de corriente.

La forma de medir un sensor resistivo es conectar un extremo a *power* y el otro a una resistencia desplegable a tierra. Luego, el punto entre la resistencia fija desplegable y el la resistencia FSR variable debe estar conectada a la entrada analógica de un microcontrolador (LADYADA, 2013).

4.12 Sensor Leap Motion

Leap Motion es un sensor equipado con dos cámaras monocromáticas IR y tres leds infrarrojos. Es una herramienta tecnológica enfocada al control de movimientos, es precisa y robusta (LEAP MOTION, 2017).

La función principal de Leap Motion es detectar como se mueven naturalmente las manos, permite apuntar, agitar, alcanzar, agarrar, recoger cosas y moverlas. Es ligero y pequeño ya que mide solo 1,27 x 3,05 x 7,62 [cm]. Su campo de visión es de 150º y todo un eje para la profundidad. Para usarlo es necesario únicamente conectarlo a una computadora y descargar el *software* Orion. El sensor es capaz de rastrear movimientos a una velocidad de más de 200 fotogramas por segundo (LEAP MOTION, 2017).

4.13 Servomotores

El servomotor es un actuador eléctrico que consta de un motor eléctrico al que se puede controlar la velocidad y la posición de eje giratorio por lo general de 0 a 180 grados, aunque algunos servomotores giran 360 grados. El servomotor se alimenta con 4 a 8 [V] dependiendo del modelo. Es controlado por pulsos eléctricos durante un tiempo determinado. El periodo del pulso es constante y tiene un valor de 20 [ms]. Para regular la posición del servomotor la parte del pulso alto debe durar entre 1 a 2 [ms]. Los pulsos deben ser generados por un microcontrolador o con un circuito analógico con un *timer* (DIGEST, 2017).

4.14 Estructura del robot InMoov

La estructura del robot InMoov está impresa originalmente en plástico ABS (LANGEVIN, 2012). El plástico PLA es otro material el cual también se usa para la impresión 3D. PLA es un polímero (ácido poliláctico) que al contrario del ABS es biodegradable ya que es fabricado a base de almidón de maíz. Es un material no

tóxico generalmente utilizado en la fabricación de objetos que entran en contacto con alimentos. Es más fácil de manipular que el ABS por su elevada velocidad de enfriamiento y solidificación (SANCHEZ, 2016).

5 Metodología

Las siguientes actividades a realizar, son fundamentales para la conclusión satisfactoria del proyecto. Primeramente, establecer los alcances y limitaciones tanto de la estructura y del sistema de control. Realizar la selección de la estructura y los materiales de construcción. Realizar la impresión 3D y correspondiente armado del robot. Diseñar e implementar un sistema de reconocimiento de movimiento de las manos y procesar las señales leídas para interpretar la información. Diseñar la parte electrónica del dispositivo, determinar los materiales, determinar las conexiones, enlaces, empalmes, soldaduras, etc. Proyectar el sistema de control del robot en función a las señales de presión y analizar el comportamiento del robot. Integrar los sistemas eléctrico, mecánico, de reconocimiento de las manos y de control. Realizar pruebas y ajustes del sistema.

6 Resultados

6.1 Sistema mecánico

Se utilizó la impresora Witbox 2, para realizar la impresión de la estructura con una calidad de impresión de 0,2 [mm], el tipo de relleno (sólido), en material PLA, el cual tiene la característica de ser flexible, biodegradable, resistente y tiene buena solidificación. Una vez impresas las piezas se lijan y pulen, de esta manera el acabo de las piezas es más uniforme, se eliminan asperezas que pueden causar fricción durante el movimiento, en la Figura 2:, se muestra la mano armada.

Figura 2: Mano robótica

La estructura seleccionada es el brazo del robot InMoov. Debido a que el movimiento de la estructura funciona a base de poleas es necesario seleccionar hilos adecuados. Los hilos de pesca ofrecen gran resistencia, son utilizados con más de una aplicación debido a sus características. El monofilamento, tiene características favorables para el proyecto como gran resistencia a los enredos y resistencia a la tensión. La poliamida seleccionada es de 1 [mm] de diámetro, este hilo resiste hasta 45 [Kg].

6.1.1 Análisis de esfuerzos de la estructura

El fabricante del robot InMoov garantiza que el brazo del mismo resiste hasta un peso de 3 [Kg]. Tomando en cuenta esta referencia, se realizó el análisis de esfuerzos de la estructura para la presente aplicación. En la Figura 3:, se muestra el robot ensamblado en el software Fusion 360, el cual está desarrollado por Autodesk. Por motivos de disponibilidad, se usó plástico ABS con calidad de 0,2 [mm].

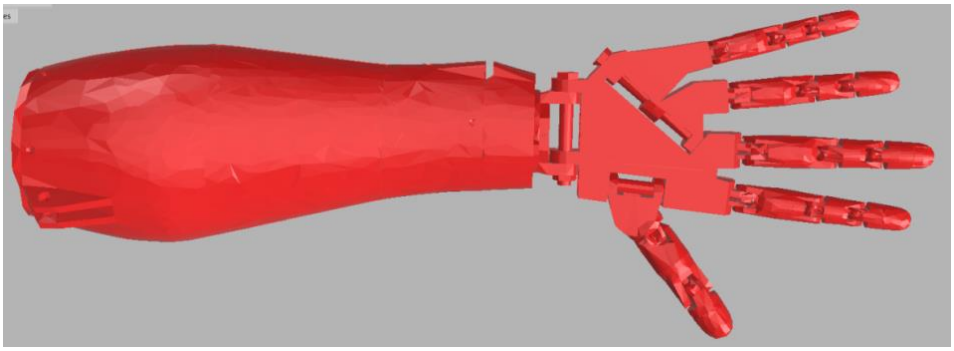

Figura 3: Estructura ensamblada en Fusion 360

Posteriormente, se procedió a realizar dos análisis de esfuerzos; el primero aplicando una fuerza puntual en la palma de 100[N], el segundo análisis aplicando fuerzas puntuales de 100[N] a cada uno de los dedos, se selecciona este valor debido a que es la fuerza máxima sensada por el FSR, y por tanto la fuerza límite de agarre. Se observó el resultado de la Figura 4:(a), Figura 4: (b) y Figura 4: (c). El *software* consta de una escala propia, la cual se muestra en la Figura 4: (c). Es por este motivo que recomienda tener un factor de seguridad mínimo de 3, ya que con eso se garantiza la durabilidad de las piezas. En la primera prueba, no existe deformidad en la estructura y no se esperan puntos de quiebre, se determina el factor de seguridad de 5,248. Por otro lado, en la segunda prueba, el factor de seguridad reduce a 2,287. El *software* indica que con esta fuerza la estructura puede sufrir deformaciones pero no rupturas a menos que se excedan las fuerzas por factores externos.

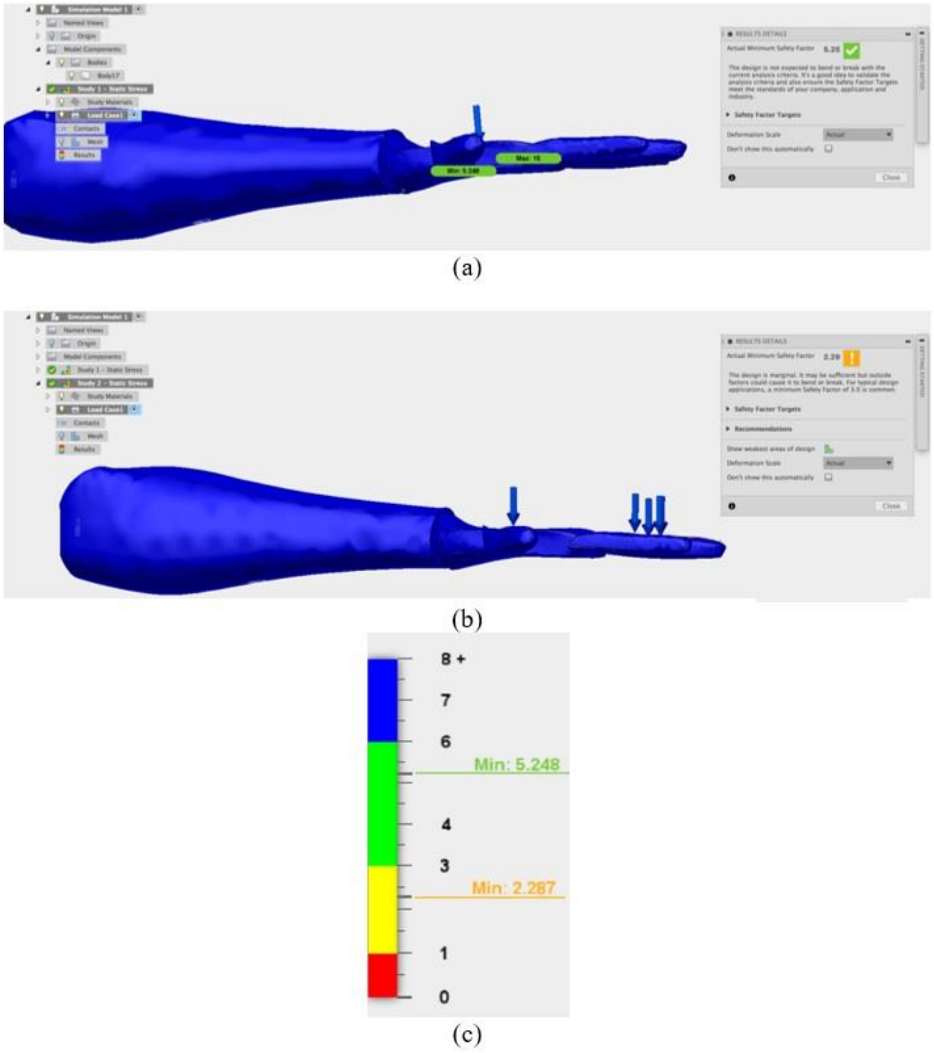

Figura 4: Análisis de esfuerzos en Fusion 360

6.2 Sistema electrónico

6.2.1 Simulaciones

El modelo de sensor de presión seleccionado fue el FSR-402, debido a que cuenta con las mejores prestaciones. El microcontrolador con mejores características es Arduino, por la cantidad de puertos PWM y el lenguaje de programación. Existe una gran gama de microcontroladores Arduino, el más adecuado según las características proporcionadas por el fabricante, es el modelo Arduino UNO.

La carencia de precisión en el los movimientos podría causar accidentes de gravedad tanto para el usuario como para el robot. El servomotor modelo Mg 996R,

es el que tiene mejores prestaciones para cumplir con los requerimientos del proyecto debido a que su velocidad de reacción y el material de los engranajes garantizan buen funcionamiento. En la Tabla 3, se muestran las características del servomotor seleccionado.

Tabla 3. Características del servomotor MG 996R

Modelo de Servomotor	Precio [Bs]	Torque [Kgf*cm]	Voltaje de operación [V]	Velocidad de reacción sin carga [Seg/60º]	Material engranaje
Mg 996R	70	9,4 a 11	4,8 a 6	0,19-0,15 Seg/60º.	Metal

Una vez seleccionados los componentes electrónicos, es necesario determinar la interacción entre ellos. En la Figura 5:, se muestra la conexión entre los FSR, Arduino y los servomotores. Esta simulación se realizó en el *software* PROTEUS.

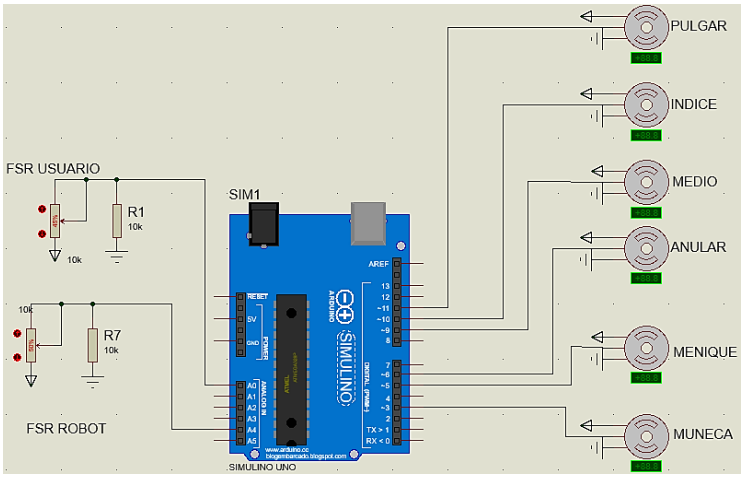

Figura 5: Diagrama de conexiones simulado en Proteus

El diagrama de conexiones se muestra en la Figura 6:, donde el sensor Leap Motion, cuenta con un cable de conexión USB para conectarlo directamente con la computadora. Para mover los actuadores es necesario tener una interfaz entre la computadora y el sensor, por esta razón se cuenta con la placa Arduino, la cual estará directamente conectada con la computadora para recibir la información procesada mediante el puerto serial. Debido a que esta placa no es capaz de proporcionar corriente eléctrica a seis servomotores de forma simultánea, se cuenta con una fuente de alimentación externa a la que están conectados todos los motores. Los FSRs, al ser sensores resistivos no tienen un consumo de corriente significante por lo que será conectado directamente al Arduino.

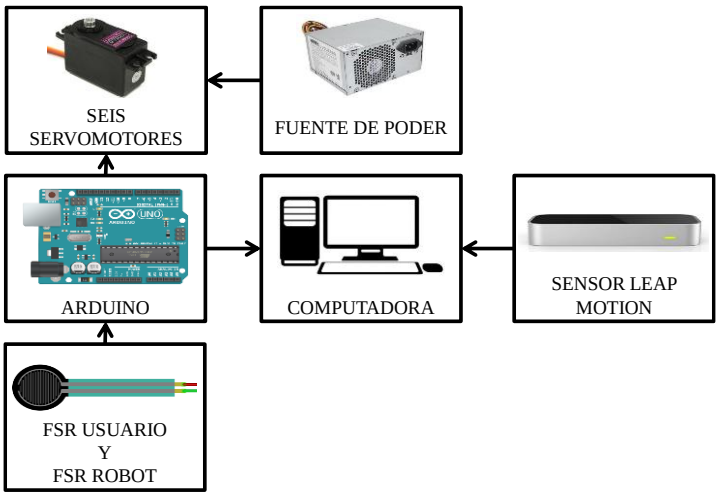

Figura 6: Diagrama de bloques del sistema eléctrico

Por otro lado, los FSRs tanto del usuario como del robot están posicionados en el pulgar de ambos elementos tal como se muestra en la Figura 7:. El FSR_u, está fijado sobre un guante de tela común con el fin de mantenerlo en una posición estable mientras se realizan tareas de sujeción o movimientos, además de proteger la parte plástica del sensor y los cables conectados al sensor. De la misma manera se fijaron los cables del sensor al guante mediante hilos para que este no se mueva durante la actividad.

Figura 7: Posición de los FSRs en el guante del usuario y en la mano robótica.

6.2.2 Cálculo de los límites de funcionamiento del sistema

Para calcular los límites de sistema, se realizaron pruebas midiendo la variación del voltaje de alimentación de los motores. Estas mediciones se realizaron con el sistema sin movimiento, con movimiento de un solo grado de libertad y con presión

en 2, 3, 4 y 5 dedos. Posteriormente se realizaron cálculos de torque, tiempo de reacción, momento y velocidad angular, de los motores.

Los datos del servomotor MG996R se describen en la Tabla 4. Estos valores son proporcionados por el fabricante. Respecto a esta tabla se realizó la aproximación del torque (T) ecuación (8) y tiempo de reacción (T_R), ecuación (9), donde V es el voltaje de alimentación. Posteriormente se transformó las unidades del torque a [Nm] y se calculó la velocidad angular (ω) respecto al tiempo de reacción del motor, según la ecuación (10). Los resultados de los cálculos, fueron almacenados en la Tabla 5.

Tabla 4. Límites de funcionamiento del servomotor MG996R

	Voltaje [V]	Torque [Kgf*cm]	Tiempo [seg/60º]	Corriente [A]
Valores mínimos	4,8	9,4	0,19	0,5
Valores máximos	7	11	0,15	2,5

$$T = 3+\frac{4}{3}V \qquad (8)$$

$$T_R = 0{,}35 - \frac{V}{30} \qquad (9)$$

$$\omega = \frac{\pi}{3\,T_R} \qquad (10)$$

Tabla 5. Parámetros de funcionamiento calculados (1)

Cantidad de motores sin carga	Cantidad de motores con carga	Parámetros calculados				
		Voltaje [V]	Torque [Kgf*cm]	Tiempo [seg/60º]	Momento [N*m]	Velocidad angular [rad/seg]
6	0	5,16	9,88	0,178	0,968897	5,88313231
5	1	5,14	9,85333333	0,1786667	0,966282	5,86118032
4	2	5,13	9,84	0,179	0,964974	5,85026565
3	3	5,11	9,81333333	0,1796667	0,962359	5,8285578
2	4	5,1	9,8	0,18	0,961052	5,81776417
1	5	5,09	9,78666667	0,1803333	0,959744	5,80701045

Posteriormente, se prosigue a calcular la potencia de los servomotores mediante la ecuación (11). Y la corriente consumida por cada servomotor mediante la ecuación (12).

$$P = M*\omega \qquad (11)$$

$$A = \frac{P}{V} \qquad (12)$$

La corriente de un servomotor sin carga es aproximadamente 0,9 [A]. Por lo tanto, la corriente total será la cantidad de motores con carga por la corriente con carga más la cantidad de motores sin carga por la corriente sin carga. Los parámetros calculados se muestran en la Tabla 6.

Tabla 6. Parámetros de funcionamiento calculados (2)

Cantidad de motores sin carga	Cantidad de motores con carga	Parámetros calculados			
		Potencia [Watts]	Corriente con carga [A]	Corriente sin carga [A]	Corriente Total [A]
6	0	5,70015	1,10468011	0,9	5,4
5	1	5,66355	1,10185847	0,9	5,601858
4	2	5,64536	1,10045933	0,9	5,800919
3	3	5,60917	1,09768425	0,9	5,993053
2	4	5,59117	1,09630826	0,9	6,185233
1	5	5,57324	1,09493994	0,9	6,3747

La corriente máxima alcanzada por los servomotores es 1,1047 [A]. Lo cual es menor a 2,5 [A], límite establecido por el fabricante. Por otro lado, es importante hallar el torque máximo que puede ejercer el servomotor a 2,5 [A]. En la ecuación (13) se realiza este cálculo.

$$T_{2,5} = \frac{9,88 * 2,5}{1,1047} \qquad (13)$$

El valor del torque calculado es 22,359 [Kgf*cm]. Este torque debe ser menor al valor del torque de ruptura de los hilos. Para calcular el valor, se tomó en cuenta la masa máxima que soporta el hilo de pesca, este es 45 Kg. Por lo tanto el peso máximo soportado por el hilo se da por medio de la ecuación (14). Donde 9,77 [m/s^2] es la el valor de la aceleración de la gravedad de Cochabamba.

$$Peso = 45*9,77 \qquad (14)$$

El valor del peso es 439,65 [N]. Para calcular el torque máximo, es necesario medir la distancia desde el eje del servomotor hasta el punto del contacto del hilo. Este valor es 1,35 [cm]. Por lo tanto, el torque máximo que se puede ejercer al hilo es 5,94 [Nm], el cual es calculado mediante la ecuación (15).

$$T_{45} = 439,65*0,0135 \qquad (15)$$

Realizando la conversión a kilogramos fuerza, se obtiene 60,5232 [Kgf*cm]. El cual es menor al torque máximo que ejercen los motores, por esta razón se tiene un margen de seguridad de 38,164 [Kgf*cm].

6.3 Sistema de reconocimiento de las manos

6.3.1 Plataforma Orion

Orion es el programa de instalación de Leap Motion, gracias a esta plataforma es posible interpretar los datos obtenidos por el sensor. En el presente proyecto se trabaja con la versión Orion 3.2. Para entender los datos obtenidos mediante la API (*Application Programming Interface*) del sensor, es necesario tomar en cuenta el sistema de coordenadas de la Figura 8:.

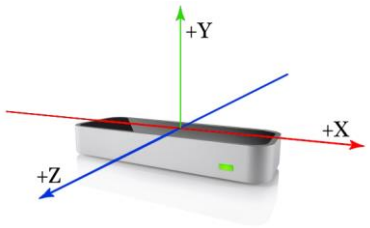

Figura 8: Sistema de coordenadas del sensor

La API, cuenta con una estructura de clases bien definida, la cual se muestra en la Figura 9:. Los objetos son *frame, hand, pointable, arm, finger* y *bone.* El objeto *frame* es esencialmente la raíz del modelo de datos y proporciona acceso a todas las entidades rastreadas por el sensor. El objeto *hand* describe la posición y la orientación de una mano, rastrea su movimiento y contiene listas de los dedos asociados con esa mano. El objeto *pointable* define las características básicas comunes a los dedos. El objeto *finger* se extiende a través de *pointeble* con información adicional específica para cada dedo. El objeto *bone* y *arm* no son usados en esta aplicación.

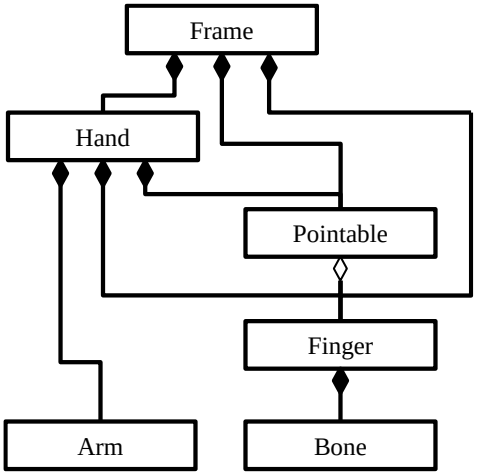

Figura 9: Diagrama de clases de la API de Leap Motion

6.3.2 Estándar Firmata

Dentro de la librería Firmata, se encuentra el protocolo Firmata genérico para la comunicación con microcontroladores desde *software* instalado en un ordenador. Este protocolo se puede implementar en cualquier arquitectura de microcontroladores, así como en cualquier paquete de *software*. El objetivo de Firmata es permitir controlar completamente Arduino desde *software* instalado en un ordenador, sin escribir una sola línea de código de Arduino.

6.3.3 Diagrama de flujo del sistema

Para realizar la programación del sistema es necesario realizar el diagrama de flujo previo donde se especifican todas las funciones e instrucciones necesarias para el funcionamiento del robot. En la Figura 10: se presenta el diagrama o línea principal del sistema.

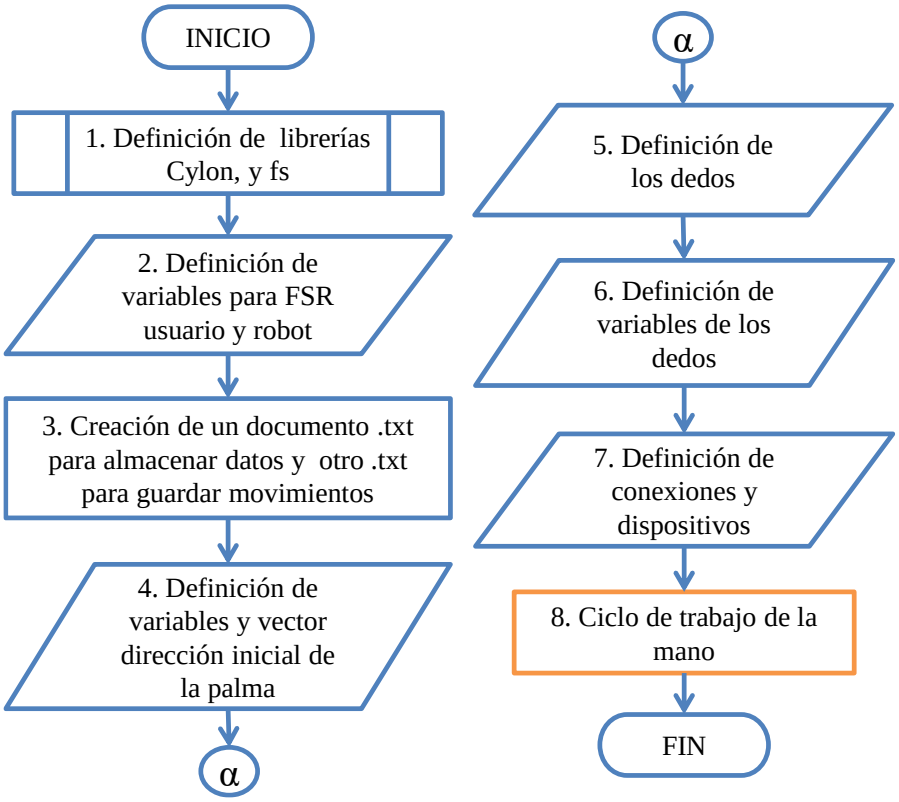

Figura 10: Diagrama principal del sistema

El ciclo de trabajo de la mano consta de un bucle infinito que termina cuando el usuario detiene el sistema, en la Figura 11:, se describe este algoritmo con cada uno de los pasos respectivos.

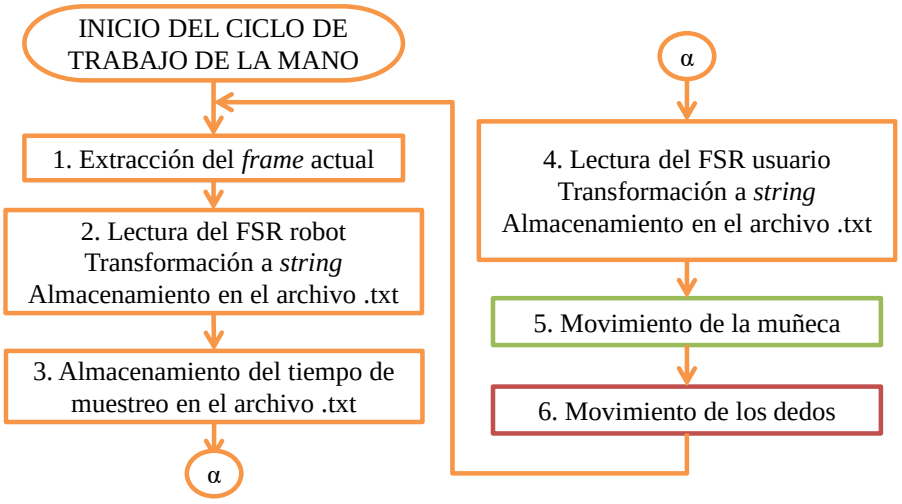

Figura 11: Diagrama del ciclo de trabajo de la mano

El movimiento de cada dedo se describe en el diagrama de la Figura 12:, este proceso se realiza para cada uno de los cinco dedos. En el bloque 3, la distancia medida (entre la punta del dedo y el centro de la palma) se reemplaza en la ecuación respectiva de cada uno de los dedos para hallar el equivalente en grados, para el dedo pulgar ecuación (16), dedo índice ecuación (17), dedo medio ecuación (18), dedo anular ecuación (19) y dedo meñique ecuación (20).

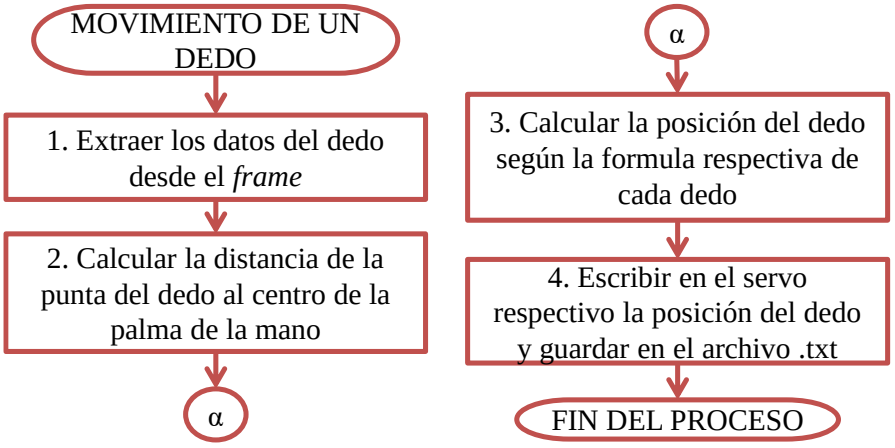

Figura 12: Diagrama del movimiento de un dedo o grado de libertad

Las ecuaciones (16), (17), (18), (19) y (20), tienen la siguiente forma: a = (d - m) * f. Dónde:

- a es el ángulo que debe girar el servomotor para replicar el movimiento de cada dedo.
- d es la distancia actual entre la punta del dedo y el centro de la palma.
- m es la distancia mínima entre la punta del dedo y el centro de la palma.
- M es la distancia máxima entre la punta del dedo y el centro de la palma.
- f es el factor de proporción, el cual es equivalente a M/m.

Los valores a, d, m, M y f son variables que dependen del tamaño de la mano del usuario. Por esta razón las siguientes ecuaciones deben ser calculadas para cada persona que use el robot.

$$\text{angPulgar} = (\text{distanciaPulgar} - 49) * 4.2861 \quad (16)$$

$$\text{angIndice} = (\text{distanciaIndice} - 42) * 3.3634 \quad (17)$$

$$\text{angMendio} = (\text{distanciaMedio} - 38) * 2.8595 \quad (18)$$

$$\text{angAnular} = (\text{distanciaAnular} - 36) * 3.0031 \quad (19)$$

$$\text{angMenique} = (\text{distanciaMenique} - 36) * 3.6416 \quad (20)$$

El movimiento de la muñeca se describe en el diagrama de la Figura 13:. En el bloque cuatro se calcula el ángulo entre el vector de referencia $u = (u_1, u_2, u_3)$ y el vector normal a la palma $v = (v_1, v_2, v_3)$, según la ecuación (21), la cual da como resultado el ángulo entre dos vectores en el espacio. Posteriormente, en el bloque cinco se realiza la conversión de radianes a grados.

$$\cos\alpha = \frac{u_1 * v_1 + u_2 * v_2 + u_3 * v_3}{\sqrt{u_1^2 + u_2^2 + u_3^2} * \sqrt{v_1^2 + v_2^2 + v_3^2}} \quad (21)$$

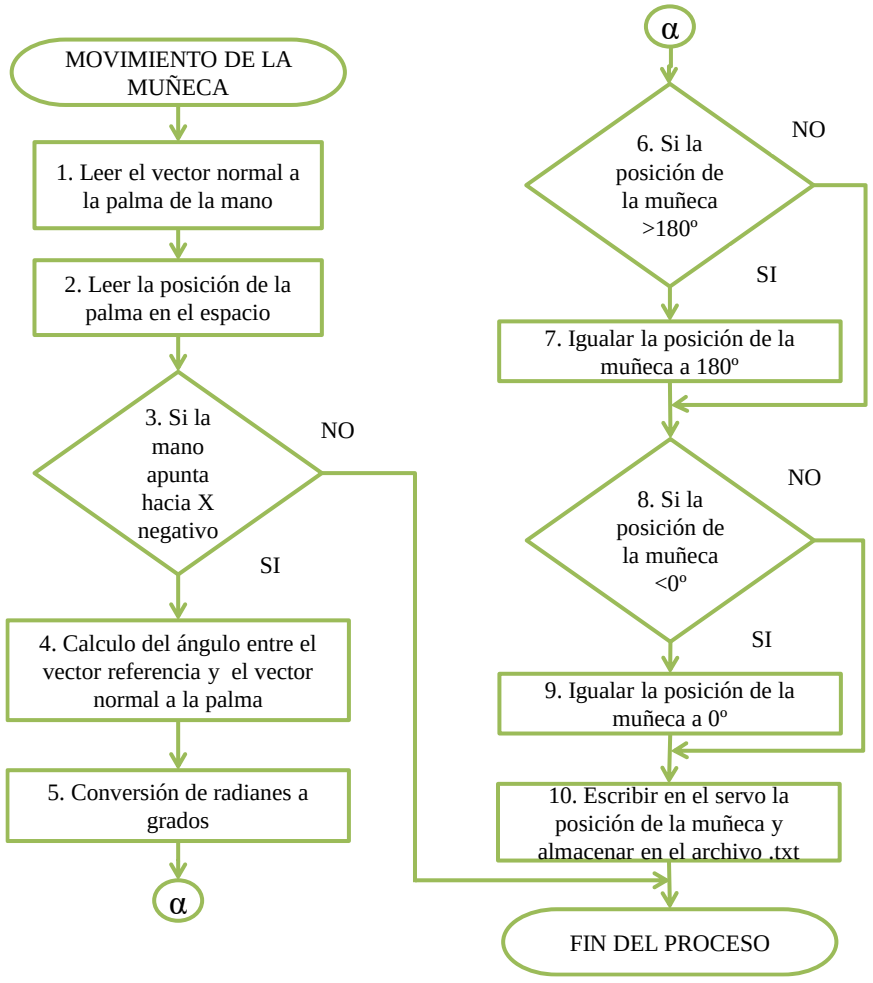

Figura 13: Diagrama del movimiento de la muñeca

6.4 Sistema de control

6.4.1 Diagrama de proceso

El primer paso de este diagrama de la Figura 14:, es la obtención de datos del sensor Leap Motion y del FSR_u, posteriormente se procesan estos datos y se realizan los cálculos respectivos. Estos datos son procesados por el controlador el cual se encarga de modificar la posición de los actuadores, esta diferencia en la posición crea movimiento en el robot. Por último se realiza la lectura del FSR_r y se retroalimenta el sistema hasta que se llega a la presión especificada por el usuario.

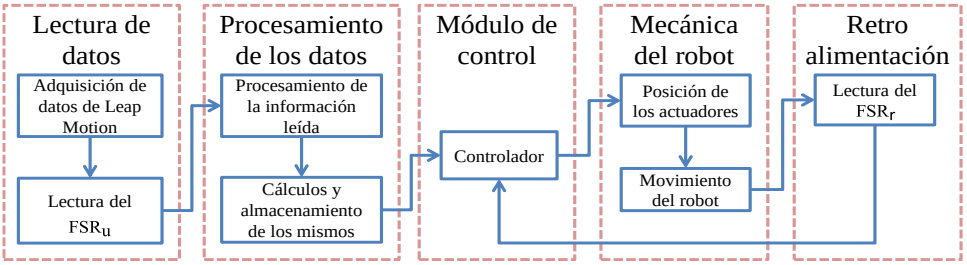

Figura 14: Diagrama de proceso a lazo cerrado

Por otro lado, este sistema MIMO se representará por cuatro sistemas SISO en paralelo, mediante cuatro modelos matemáticos y cuatro controladores, los cuales dependen de la cantidad de dedos que están en contacto con el objeto que se sujeta en un determinado tiempo. El diagrama de este proceso se muestra en la Figura 15:.

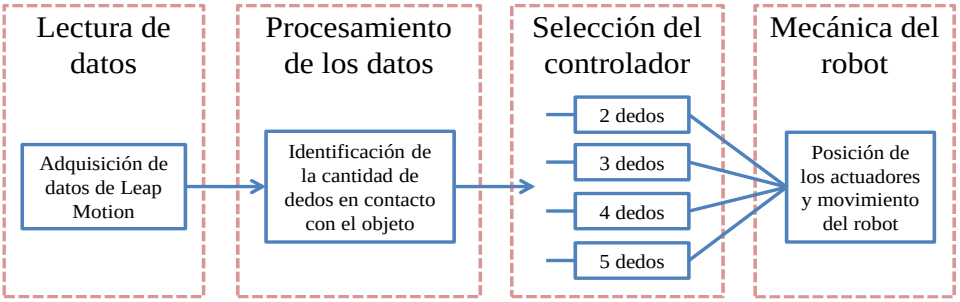

Figura 15: Sistema MIMO representado por 4 sistemas SISO en paralelo

6.4.2 Cálculo del *setpoint*

Para el uso de PIPIMC es necesario determinar el valor del *setpoint*. Para el robot, este valor está dado por el FSR_u al ser la presión de referencia. Para cada objeto que se sostiene este valor cambia en un rango de 0 a 1 000 (adimensional), lo cual representa una fuerza de presión entre 0 y 100 Newtons. Dado que esta variable cambia para cada instrumento de laboratorio, se toma en cuenta la Tabla 7, para determinar un valor promedio.

Tabla 7. Presión en instrumentos de laboratorio.

Herramienta	Presión en Newtons	Valor del sensor
Bureta	30,514	305,143
Probeta con 50 ml de agua	76,522	765,217
Matraz aforado	48,092	480,917
Matraz Erlenmeyer	70,294	702,938
Piseta con 1 L de agua	86,4	864
Crisol de porcelana	73,149	731,485
Pipeta	30	299,375

El cálculo del *setpoint* (Sp), se da en la ecuación (22). Por lo tanto el valor de Sp es 426,695. Es importante tomar en cuenta que este valor está calculado mediante la columna del valor del sensor y no así por la columna en Newtons de la Tabla 7.

$$Sp = \frac{305,143+765,217+480,917+702,938+864+731,485+299,375}{7} \qquad (22)$$

6.4.3 Pruebas con PIPIMC

Una vez calculado el *setpoint,* se realizaron pruebas con el robot y una frecuencia de muestreo de 0,1 [s/muestra]. Estas pruebas consisten en agarrar un objeto de laboratorio con 2, 3, 4 y 5 dedos, para ver el comportamiento del sistema en cada uno de estos casos.

El procedimiento de uso del *software* PIPIMC es el mismo para todos los tipos de agarre. En cada prueba se deben ajustar los valores λ y Ts, en la ventana de Tuning PI-IMC, verificando para qué valores se tienen mejor respuesta del sistema.

El proceso de modelamiento FOPDT del sistema con dos dedos, dio como resultado los valores mostrados en la Figura 16:, donde se identifica a la curva azul como la respuesta real y a la curva roja como el valor estimado. Se usó el tipo de proceso en circuito abierto, el modelo de Nishikawa debido a que está enfocado en sistemas de circuito abierto. Posteriormente se ejecutó la ventana Tuning-IMC como se muestra en la Figura 17:, donde se seleccionó el modelo de primer orden aproximado 3 de la Tabla 1.

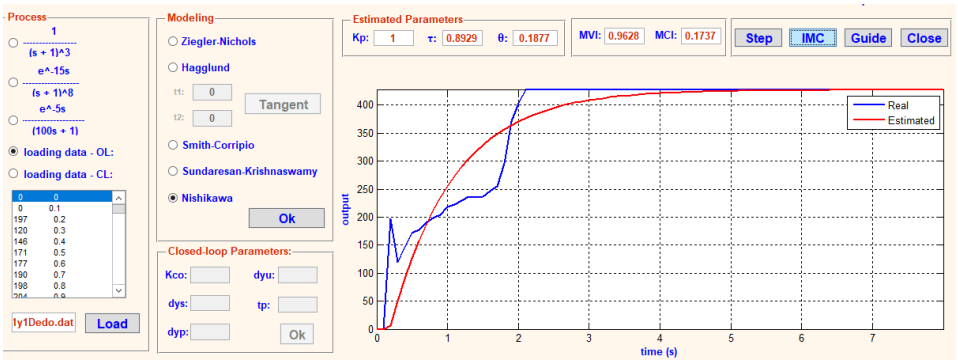

Figura 16: Proceso de modelamiento FOPDT del sistema con 2 dedos

En la Figura 17: se muestra la respuesta del sistema al escalón unitario, a un escalón de valor 3 y finalmente al escalón unitario nuevamente. Donde la señal constante negra es referencia, la señal de salida es roja y la señal de control es azul punteada.

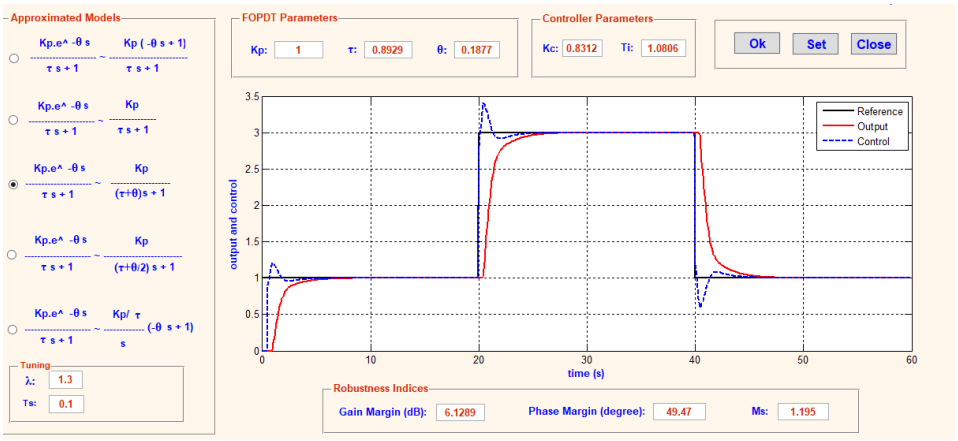

Figura 17: Tuning-PI-IMC para agarre con 2 dedos

En la Figura 18: se muestra el proceso de modelamiento FOPDT del sistema con tres dedos. Donde se identifica a la curva azul como la respuesta real y a la curva roja como el valor estimado. Esta respuesta tiene menos ruido que la respuesta con dos dedos de agarre.

En la Figura 19: se muestra la ventana Tuning-IMC del sistema con tres dedos. Donde la señal constante negra es referencia, la señal de salida es roja y la señal de control es azul punteada. Esta respuesta tiene mayor oscilación y retraso que la respuesta con dos dedos de agarre.

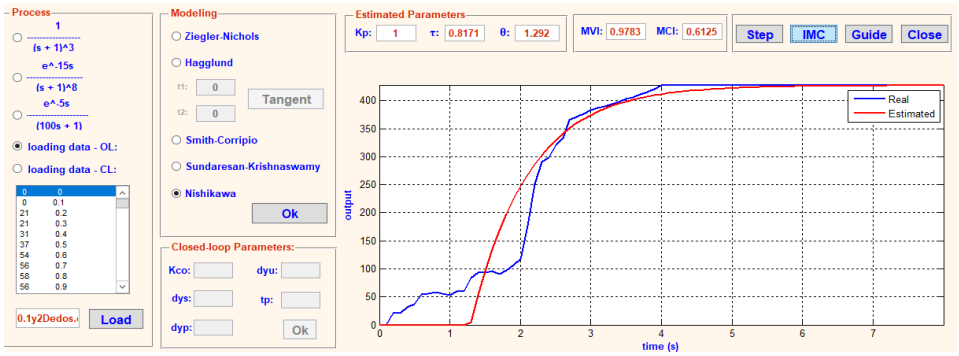

Figura 18: Proceso de modelamiento FOPDT del sistema con 3 dedos

En la Figura 20: se muestra el proceso de modelamiento FOPDT del sistema con cuatro dedos. Donde se identifica a la curva azul como la respuesta real y a la curva roja como el valor estimado. Esta respuesta tiene un retraso considerable en comparación con los otros tipos de agarre. En la Figura 21: se muestra la ventana Tuning-IMC del sistema con cuatro dedos.

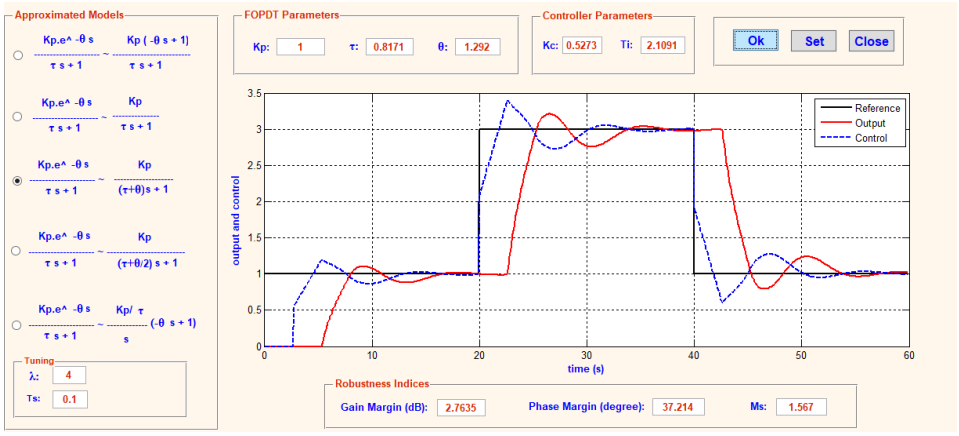

Figura 19: Tuning-PI-IMC para agarre con 3 dedos

En la Figura 22: se muestra el proceso de modelamiento FOPDT del sistema con cinco dedos. Donde se identifica a la curva azul como la respuesta real y a la curva roja como el valor estimado. Esta respuesta no cuenta con retraso considerable. En la Figura 23: se muestra la ventana Tuning-IMC del sistema con cinco dedos. Donde la señal constante negra es referencia, la señal de salida es roja y la señal de control es azul punteada Se muestra que el tiempo de reacción y la oscilación tienden a cero.

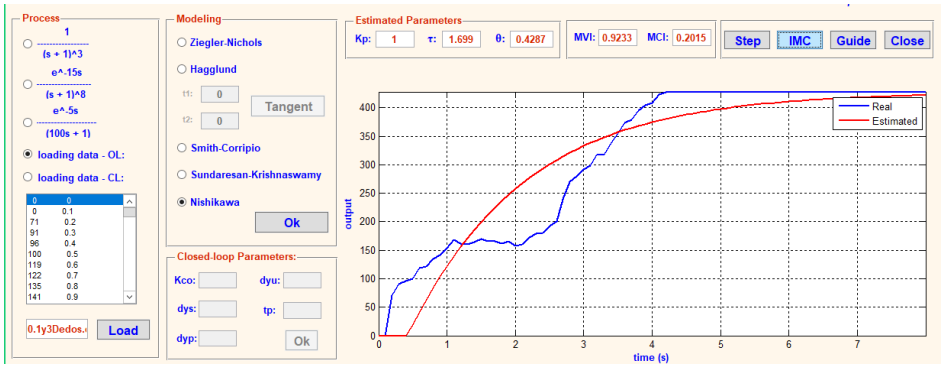

Figura 20: Proceso de modelamiento FOPDT del sistema con 4 dedos

Todos los parámetros estimados tanto del modelamiento FOPDT como del Tuning PI-IMC están clasificados en las Tabla 8 y 9 respectivamente, se analizaron los valores de MVI, los cuales demuestran la consistencia de los sistemas. Por otro lado, el valor de MCI identificó el tipo de modelo de cada uno de los sistemas.

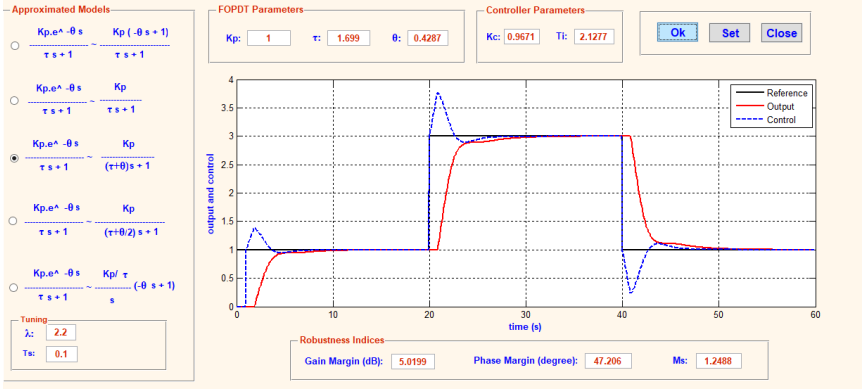

Figura 21: Tuning-PI-IMC para agarre con 4 dedos

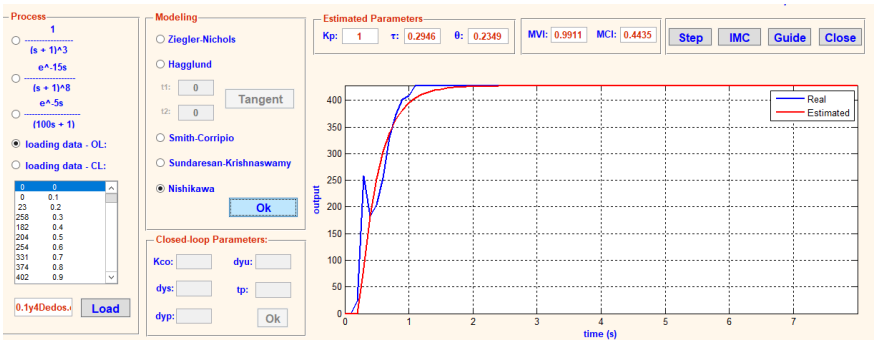

Figura 22: Proceso de modelamiento FOPDT del sistema con 5 dedos

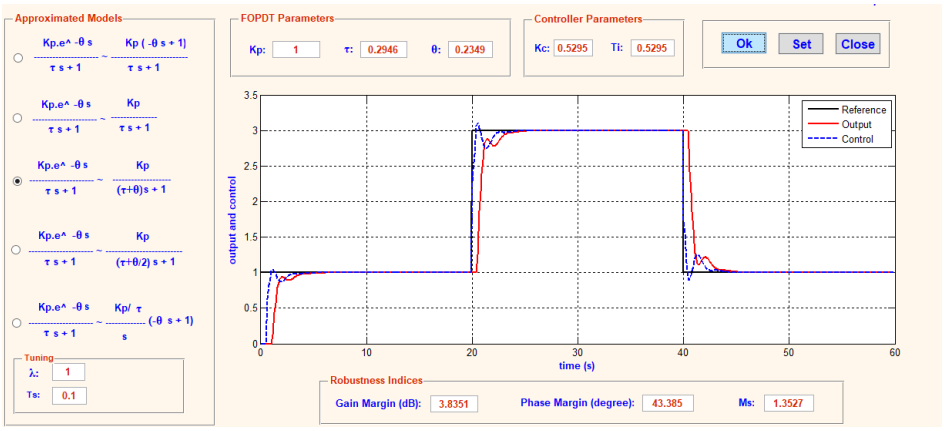

Figura 23: Tuning-PI-IMC para agarre con 5 dedos

Tabla 8. Parámetros estimados FOPDT para cada tipo de agarre

Cantidad de dedos	Parámetros estimados FOPDT					
	Kp	τ [s]	θ [s]	MVI	MCI	Tipo de MCI
2	1	0,8929	0,1877	0,9628	0,1737	*lag-dominated*
3	1	0,8171	1,292	0,9783	0,6125	*Balanced*
4	1	1,699	0,4287	0,9233	0,2015	*Balanced*
5	1	0,2946	0,2349	0,9911	0,4435	*Balanced*

Reemplazando los datos de la Tabla 8 en el modelo equivalente 3, da como resultado el modelo de primer orden de la ecuación (23), resultante para el agarre con dos dedos, la ecuación (24) es la resultante para el agarre con tres dedos, la ecuación (25) es la resultante para el agarre con cuatro dedos, la ecuación (26) es la resultante para el agarre con cinco dedos.

$$\frac{e^{-0,1877s}}{0,8929s+1} = \frac{1}{1,0806s+1} \qquad (23)$$

$$\frac{e^{-1,292s}}{0,8171s+1} = \frac{1}{2,1091s+1} \qquad (24)$$

$$\frac{e^{-0,4287s}}{1,699s+1} = \frac{1}{2,1277s+1} \qquad (25)$$

$$\frac{e^{-0,2349s}}{0,2946s+1} = \frac{1}{0,5295s+1} \qquad (26)$$

En la Tabla 9, se clasifican los parámetros de PI-IMC, en todos los casos se tomó una frecuencia de muestreo de 0,1 [seg/muestra], por otro lado se determinó

λ para cada uno de los casos, este valor determina la robustez del sistema, mientras más inestable es la respuesta del sistema este número aumenta, como en el agarre con tres dedos, por otro lado, el valor es menor cuando el sistema no presenta retraso u oscilaciones considerables.

Tabla 9. Parámetros de PI-IMC para cada tipo de agarre

Nº de dedos	Parámetros de control		Tuning	
	Kc	Ti	λ	Ts [s]
2	0,8312	1,0806	1,3	0,1
3	0,5273	2,1091	4	0,1
4	0,9671	2,1277	2,2	0,1
5	0,5295	0,5295	1	0,1

Según la aproximación número 3 de la Tabla 1. Se obtiene la ecuación (27).

$$C_{PI}(s) = \left(\frac{\tau+\theta}{K\lambda}\right)\left(1+\frac{1}{(\tau+\theta)\,s}\right) \qquad (27)$$

Realizando operaciones aritméticas sobre la ecuación (27), se llega a la ecuación (28), donde se reemplazaran los valores estimados de τ, θ, K y λ, para cada uno de los casos.

$$C_{PI}(s) = \frac{(\tau+\theta)s+1}{K\lambda\, s} \qquad (28)$$

Reemplazando los datos de la Tabla 24, se obtienen las siguientes ecuaciones (29) a (32), que representan la estructura del PI para agarres con 2, 3, 4 y 5 dedos.

$$C_{PI\text{-}2Dedos}(s) = \frac{1{,}0806s+1}{1{,}3s} \qquad (29)$$

$$C_{PI\text{-}3Dedos}(s) = \frac{2{,}1091s+1}{4s} \qquad (30)$$

$$C_{PI\text{-}4Dedos}(s) = \frac{2{,}12775s+1}{2{,}2s} \qquad (31)$$

$$C_{PI\text{-}5Dedos}(s) = \frac{0{,}5295s+1}{s} \qquad (32)$$

Posteriormente, se realizó la discretización de estas ecuaciones con ayuda de Matlab. Obteniendo como resultado las ecuaciones (33) a (36).

$$C_{z\text{-}2Dedos}(z) = \frac{1.001\,z - 0.912}{z - 1} \qquad (33)$$

$$C_{z\text{-}3Dedos}(z) = \frac{0.5398\ z - 0.5148}{z - 1} \qquad (34)$$

$$C_{z\text{-}4Dedos}(z) = \frac{0.9899\ z - 0.9445}{z - 1} \qquad (35)$$

$$C_{z\text{-}5Dedos}(z) = \frac{0.5795\ z - 0.4795}{z - 1} \qquad (36)$$

A continuación, se reemplazan las ecuaciones en la siguiente igualdad $u(k) = C_{z\text{-}iDedos}(z) * e(k)$, obteniendo la ecuación genérica (37). Mediante operaciones aritméticas se llega a la ecuación (38). Finalmente, se reemplazan valores numéricos en la ecuación (38) para obtener las ecuaciones (39) a (42), las cuales tienen la misma forma que la ecuación (7). Estas ecuaciones están discretizadas, por lo tanto es posible incluirlas en el algoritmo de control.

$$u(k) = \frac{az - b}{z - 1} e(k) \qquad (37)$$

$$u(k) = u(k\text{-}1) + a\ e(k) - b\ e(k\text{-}1) \qquad (38)$$

$$u_{2Dedos}(k) = u_{2Dedos}(k\text{-}1) + 1.0010\ e(k) - 0.9120\ e(k\text{-}1) \qquad (39)$$

$$u_{3Dedos}(k) = u_{3Dedos}(k\text{-}1) + 0.5398\ e(k) - 0.5148\ e(k\text{-}1) \qquad (40)$$

$$u_{4Dedos}(k) = u_{4Dedos}(k\text{-}1) + 0.9899\ e(k) - 0.9445\ e(k\text{-}1) \qquad (41)$$

$$u_{5Dedos}(k) = u_{5Dedos}(k\text{-}1) + 0.5795\ e(k) - 0.4795\ e(k\text{-}1) \qquad (42)$$

6.5 Implementación del control

A continuación, en la Figura 24:, se muestra el desarrollo del algoritmo de control, representador por un diagrama de flujo implementado para un grado de libertad.

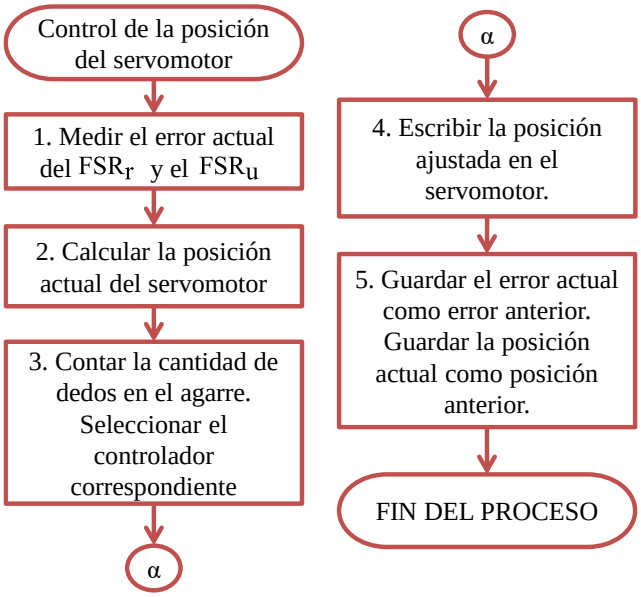

Figura 24: Diagrama de flujo del sistema de control

7 Conclusiones

El sistema es capaz de replicar los movimientos de la mano. El sistema de control mejora el funcionamiento del robot en comparación con el sistema en lazo abierto y es capaz de manipular sustancias químicas en cantidades menores a 500 mL.

El algoritmo de reconocimiento de manos es capaz de realizar 50 ciclos de trabajo por segundo, es por esta velocidad de procesamiento que tiene una reacción de 40 milisegundos de retraso, este tiempo es casi imperceptible para el ojo humano, por lo que puede considerarse tiempo real.

Los datos obtenidos mediante el sistema de reconocimiento de la mano, son modelos matemáticos de alto orden que fueron relacionados con la señal de presión, por esta razón se vio la necesidad de levantar modelos de orden reducido equivalentes de los mismos. En los cuales se determinó si eran *lag-dominated, balanced* o *delay-dominated*. Los datos fueron almacenados en vectores de 400 (equivalente a ocho segundos) para poder analizar y clasificar las señales.

El sistema MIMO se reemplazó por cuatro sistemas SISO en paralelo. Para cada sistema SISO se proyectó un controlador a partir del modelo matemático de orden reducido FOPDT obtenido del *software* PIPIMC, donde se identificó el sistema de primer orden con la técnica de Nishikawa, y posteriormente se proyectó el controlador con la técnica de IMC para sintonizar un PI. Dicha técnica aumenta la robustez del controlador a partir de la variable lambda.

Referencias Bibliográficas

[1] ALVAREZ, J. (1999). Técnicas de control para robots manipuladores con flexibilidad en la acticulación. Nuevo Leon: Universidad Autónoma de Nuevo Leon.

[2] ARDUINO. (2017). Arduino. Obtenido de Arduino: https://www.arduino.cc/

[3] BISHOP, R., & DORF, R. (2008). Modern Control Systems Eleventh edition. United States: Pearson Education International.

[4] BOBAL, V., BOHM, J., FESSL, J., & MACHACEK, J. (2005). Digital Self-tuning Controllers. República Checa: Springer.

[5] CARRION, L., OCHOA, D., & VALVERDE, J. (2009). Análisis del funcionamiento del sensor resistivo (FSR) con LabVIEW. Cuenca: NATIONAL INSTRUMENTS.

[6] CONSEJO DE SEGURIDAD NUCLEAR. (2014). Robots al rescate. Revista de seguridad nuclear y protección radiológica, 4-5.

[7] DIGEST, C. (13 de Octubre de 2017). CIRCUIT DIGEST. Obtenido de https://circuitdigest.com/article/servo-motor-basics

[8] FADALI, S. (2009). Digital control engineering. Burlington: Elsevier Inc.

[9] LADYADA. (2013). Force Sensitive Resistor (FSR). Adafruit learning system.

[10] LANGEVIN, G. (Enero de 2012). iinmoov. Obtenido de inmoov: http://inmoov.fr/

[11] LEAP MOTION. (2017). Leap Motion Datasheet. LEAP MOTION, INC.

[12] MARCO DE RESPUESTA A EMERGENCIAS. (2013). Accidentes químicos. Washington, DC: World Health Organization.

[13] MAZZONE, V. (2002). Controladores PID. Quilmes: Universidad Nacional de Quilmes.

[14] NACIONES UNIDAS PARA LA FORMACIÓN PROFECIONAL E INVESTIGACIÓN. (2008). Perfil Nacional de Sustancias Químicas en Bolivia. La Paz: SAICM.

[15] OLLERO, A. (2001). Robótica, manipuladores y robots moviles. Barcelona: Marcombo S.A.

[16] PATNAIK, P. (2007). A comprehensive guide to the hazardous properties of chemical substances. New Jersey: John Wiley & Sons, Inc.

[17] PESHKIN, M., COLGATE, E., WANNASUPHOPRASIT, W., MOORE, C., GILLESPIE, B., & AKELLA, P. (2001). Cobot Architecture. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, 377-389.

[18] RODRIGUES, A., & DOS SANTOS, L. (2004). Indentificación de sistemas dinámicos lineales. Florianópolis: Universidad Federal de Santa Catarina.

[19] SANCHEZ, S. (15 de Julio de 2016). Plasticos de la impresión 3D. Obtenido de 3D natives: https://www.3dnatives.com/es/plasticos-impresion-3d-22072015/

[20] VEGA, R. (15 de Septiembre de 2016). Cylon.js. Obtenido de https://ricveal.com/blog/cylon-js/

[21] WORLD HEALTH ORGANIZATION. (2009). Manual for the public health management of chemical incidents. Geneva: World Health Organization.

[22] YARTO, M., IZE, I., & GAVILÁN, A. (2003). "El universo de las sustancias químicas peligrosas y su regulación para un manejo adecuado". Gaceta ecológica, 57-66.