

# **DISEÑO DE UN CONTROLADOR PD DIFUSO Y ESTUDIO COMPARATIVO CONTRA UN CONTROL PID CLÁSICO PARA EL MODELO 3D DE UN ROBOT HÁPTICO**

*FUZZY PD CONTROLLER DESIGN AND COMPARATIVE STUDY WITH A CLASSIC PID CONTROLLER FOR A 3D MODEL OF AN HAPTIC ROBOT*

**Jaime Jalomo Cuevas**

Tecnológico Nacional de México / IT de Cd. Guzmán, México  
*jaime.jc@cdguzman.tecnm.mx*

**José Luis Cortés Mendoza**

Tecnológico Nacional de México / IT de Cd. Guzmán, México  
*josem20291055@cdguzman.tecnm.mx*

**Jorge Gudiño Lau**

Universidad de Colima, México.  
*jgalu@ucol.mx*

**Sergio Sandoval Pérez**

Tecnológico Nacional de México / IT de Cd. Guzmán, México  
*ssandoval@cdguzman.tecnm.mx*

**José de Jesús García Cortes**

Tecnológico Nacional de México / IT de Cd. Guzmán, México  
*jesusgc@cdguzman.tecnm.mx*

**Recepción:** 29/agosto/2021

**Aceptación:** 24/noviembre/2021

## **Resumen**

Este trabajo presenta el rediseño y optimización de un dispositivo háptico mediante las herramientas de SolidWorks, para su exportación a Simulink, mediante la herramienta Simmechanics segunda generación. Se crea un diagrama a bloques para la simulación del comportamiento de los motores de cada articulación, mediante el uso de una función de transferencia obtenida a partir de los parámetros de un motor de características similares. Se crea un controlador PD (proporcional-derivativo) difuso, sintonizado heurísticamente a partir de las características del sistema y la respuesta deseada. El controlador diseñado prueba ser eficiente ante los movimientos programados en las articulaciones. La presente metodología

muestra ser una alternativa para la enseñanza de los conceptos básicos del control difuso en las instituciones de educación superior de recursos limitados, suministrando la posibilidad de realizar prácticas sobre dispositivos virtuales, eliminando los costos de equipo físico y conservando el valor didáctico de la visualización del comportamiento en tiempo real del controlador.

**Palabras Clave:** Controlador PD difuso, Simmechanics, SolidWorks.

## **Abstract**

*This paper shows a haptic device's redesign and optimization through the tools provided by SolidWorks and its exportation to Simulink using the Simmechanics 2nd generation's resources. Parting on a similar motor's parameters a transfer function is obtained and then added to a block diagram created to simulate the 3-degree freedom device's behavior. Fuzzy PD (proportional-derivative) controller is created and heuristically tuned, considering system's characteristics and the desired response. The controller proves to be efficient in every movement programmed for all three articulations. Developed methodology proved to be a good alternative to teach fuzzy control basics in low resources universities, permitting them to direct laboratory practices using virtual devices, removing the need for physical equipment but keeping the didactic benefits that comes from watching a controller's real time behavior.*

**Keywords:** Fuzzy PD controller, Simmechanics, SolidWorks.

## **1. Introducción**

Cuando se refiere a educación superior, en México hay una oferta variada de universidades que ofrecen sus servicios, sin embargo, existe una gran diferencia entre los recursos a los que pueden acceder las universidades de iniciativa privada y a los que se acceden en las subsidiadas por el gobierno. En la rama de las ingenierías, el equipo a disposición para prácticas de laboratorio es de un gran valor didáctico para el estudiante, particularmente cuando son dispositivos con los que podrá interactuar en su etapa de egresado, por lo que carecer del contacto con ellos supondrá una clara desventaja. Al hacer una simple comparativa entre los equipos

dentro de una institución privada, contra los que hay en una universidad pública, es fácil notar la diferencia entre las prestaciones a las que el estudiante de cada universidad podrá acceder, ya sea en cuanto a equipo de cómputo, laboratorios o tecnología en general, esta realidad priva todos los días a miles de estudiantes de alcanzar su máximo potencial y puede complicar su capacidad de obtener mejores puestos de trabajo. Por lo expuesto anteriormente es que dentro de las instituciones se ha comenzado a incentivar la reparación y reingeniería de equipo que se compra a un menor precio a ciertas empresas o universidades, con el fin de poner a disposición de los alumnos, una experiencia real con tecnología de época. Este trabajo forma parte de un proyecto de varias etapas que tiene por finalidad abrir el código y hacer la reparación de un dispositivo háptico de la marca Phantom, a fin de utilizar ese código para que mediante el uso de este robot pueda enriquecerse la experiencia de los estudiantes del Instituto Tecnológico de Ciudad Guzmán y la Universidad de Colima. Podemos ver en la figura 1 una foto del dispositivo real, el cual es la parte principal de este proyecto.



Figura 1 Robot háptico Phantom.

En esta segunda etapa del proyecto de reingeniería del dispositivo mostrado anteriormente, buscamos crear una plataforma que apoye el aprendizaje de los fundamentos del control difuso mediante un modelo 3D del dispositivo y el uso de

un diagrama a bloques de Simulink. Para lograr lo anterior primero se hizo un proceso de optimización sobre el modelo de SolidWorks original para reducir la carga que significa el procesamiento del sistema digital, mediante la fusión de partes y la supresión de elementos innecesarios dentro del modelo 3D. Por otro lado, mediante el uso de una función de transferencia, obtenida a partir de los parámetros de un motor de características similares, se añadió la dinámica de comportamiento propia del sistema dentro del marco de la simulación que se realiza dentro de Simulink. Por último, se realizó la labor de diseño y puesta en marcha de un controlador PD difuso, el cual fuera capaz de alcanzar estándares de desempeño dignos de un controlador lo suficientemente robusto y lo suficientemente rápido para competir con otro tipo de controladores, con la ventaja de que, al ser más tolerante al error, las reglas de diseño y su comportamiento resulta menos complejo que sus contrapartes que a menudo dependen de reglas matemáticas y un entendimiento de conceptos más avanzados.

La identificación de los parámetros de los motores del robot mediante un método probado, así como la implementación del control en tiempo real sobre el dispositivo en físico y su comparación con la simulación, se abordará en un estudio posterior.

## **2. Métodos**

La primera parte del trabajo fue el diseño del modelo digitalizado del robot Phantom mediante una plataforma CAD (Diseño Asistido por Computadora), en este caso el software seleccionado fue SolidWorks, una plataforma sobre la cual podemos hacer el modelo 3D de las piezas que juntas, conforman el robot en torno al que se hizo el resto del trabajo. Cabe señalar que una de las razones para optar por usar SolidWorks es la herramienta que nos ofrece para importar el modelo 3D a Simulink mediante la librería Simmechanics, de esta forma podemos aprovechar las prestaciones de Simulink para hacer el control necesario sobre el robot. Ahora bien, no fue necesario hacer un diseño del modelo 3D desde cero, ya que se utilizó el modelo desarrollado en un trabajo previo sobre este mismo robot [Jaime Jalomo, 2020], sin embargo, si se realizaron algunas modificaciones a este modelo por razones relacionadas con la optimización del diagrama de bloques exportado a

Simulink. Al hacer una réplica 3D de un dispositivo mediante un programa de computadora se debe considerar que cada pieza que se agrega tiene un impacto sobre el procesamiento que la computadora tiene que hacer para simular movimientos, y sobre la complejidad del diseño que será exportado a la interfaz de Simulink [Bethune, 2017], esto debido a que dentro del programa se le asigna un bloque a cada pieza que se diseña. Por esta razón, se simplificó el diseño fusionando partes que normalmente habrían de estar unidas mediante tornillos, así como elementos que, aunque son diseñados individualmente, en realidad se comportan como una sola pieza. Esta parte del proceso es muy importante cuando llega el momento de exportar el modelo, ya que así como se le asigna un bloque a cada pieza, a cada pieza también se le asigna un movimiento específico con respecto a las piezas con las que está unida en el ensamblaje, lo cual es un problema si son demasiadas piezas, ya que esto significa tener bastantes movimientos posibles en piezas que no son relevantes, cuando en realidad estamos hablando de un robot que solo tiene 3 grados de libertad, o dicho de otra forma, un robot del cual solo nos interesa saber el comportamiento que tiene en 3 movimientos específicos. Al terminar este proceso pasamos de tener un ensamblaje de 25 piezas, a un ensamblaje de solo 6, figura 2.

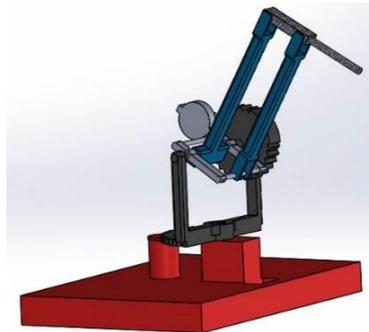


Figura 2 Modelo 3D en SolidWorks

Una vez creado el modelo 3D adecuado continuamos con la exportación del modelo mediante la herramienta Simscape Multibody Link, un proceso que genera un archivo extensión .xml, el cual contiene las instrucciones de ensamblaje que se leerán mediante MATLAB para la creación de un modelo en Simmechanics, cabe

señalar que contamos con la posibilidad de exportar el modelo mediante la primera o segunda generación de la herramienta Simscape, en este caso en particular se usó la segunda generación, lo cual nos ofrece más prestaciones y la gran ventaja de tener una animación de los movimientos generados durante la simulación, que podemos repetir las veces que sean necesarias, sin necesidad de ejecutar la simulación una y otra vez.

El diagrama a bloques del modelo 3D que se generó mediante simmechanics no se muestra para mantener el enfoque de presente proyecto sobre el control del mismo, sin embargo, el modelo 3D creado se muestra en la figura 3.

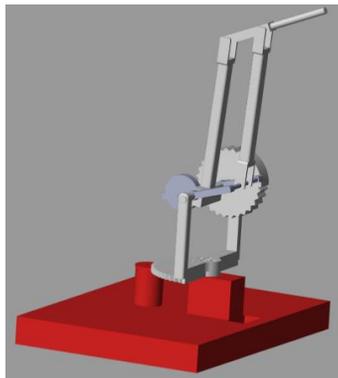


Figura 3 Modelo 3D en MATLAB

El modelo 3D de Simulink originalmente no cuenta con las facilidades para definir posiciones o movimientos a cada una de las piezas, sino que es nuestra labor añadir bloques de control y de sensado de movimiento sobre los bloques que representan los movimientos que nos interesan, en este caso, los movimientos tipo revoluta del medio engrane de la cintura, del brazo y del antebrazo [Dwiputra, 2014]. Cada uno de estos bloques fueron configurados para depender de la posición (rad), aceleración ( $\text{rad/s}^2$ ) y velocidad angular (rad/s), para así simular la dinámica de movimiento deseada. Los bloques de sensado fueron configurados para proporcionar la posición en radianes.

Ahora bien, puesto que la posición de cada una de las partes de este robot depende del control de un motor CD, fue necesario introducir la dinámica de este motor al modelo para controlarlo. Ni el fabricante del robot ni el de los motores nos

proporcionan muchos de los parámetros de los motores que el robot utiliza para moverse, por lo cual, para obtener una función de transferencia que describa exactamente el comportamiento de sus motores hubiera sido necesario hacer una estimación de los parámetros, lo cual es un proceso de varias fases [Vallejo, 1997], sin embargo, esa labor será abordada en un trabajo diferente, ya que debido a la complejidad de la tarea es necesario explicar en detalle cada uno de los pasos. Para este trabajo se optó por usar la función de transferencia de un motor de características similares a las de los motores del robot, con la diferencia de que el fabricante si nos proporciona toda la información necesaria acerca del motor, como para obtener una función de transferencia a partir de ella.

La función de transferencia a partir de la cual obtuvimos nuestro modelo es la propuesta por [Phillips, 1989], la cual podemos ver en la ecuación 1.

$$\frac{\Theta(s)}{E(s)} = \frac{\frac{K_T}{R_a}}{Js^2 + \left[\frac{BR_a + K_T K_b}{R_a}\right]s} \quad (1)$$

En tabla 1 se muestran valores de cada una de las variables.

Tabla 1 Parámetros del motor

Descripción del parámetro	Valor
$\Theta(s)$ es la posición en grados	--
$E(s)$ es el voltaje aplicado	--
$K_T$ es la constante de torque	0.052
$K_b$ es la constante de fuerza contraelectromotriz	0.052
$J$ es el momento de inercia del rotor	$11.6 \times 10^{-5}$
$B$ es el coeficiente de fricción viscosa	$2 \times 10^{-6}$
$R_a$ es la resistencia terminal	10.61

A partir de este punto, se diseñó un controlador PD difuso, el cual consta de dos entradas, el error y la derivada del error de posición, ambas expresadas en radianes, y una salida en voltaje, el cual estará limitado a un rango de -15 a +15 V.

El proceso de diseño de un controlador difuso, a diferencia de otros tipos de control, está basado en la forma en que los humanos pensamos y permite cierto grado de imprecisión para alcanzar una eficiencia superior en algunas aplicaciones [Ross,

2010], debido a esto, para su diseño no se hizo uso de fórmulas ni reglas, más bien se optó por un proceso heurístico para definir las funciones de membresía de entrada y de salida, así como las reglas que las relacionan. Una vez diseñado el controlador, se evaluó el desempeño del mismo, en términos del error de estado estable, el sobreimpulso y el tiempo de respuesta.

### 3. Resultados

#### Diagrama a bloques en Simulink

La figura 4 muestra el diagrama a bloques del modelo final, el cual ha sido simplificado mediante la encapsulación en subsistemas. Para ejecutar cualquier movimiento es necesario administrar una entrada a la articulación en cuestión, ya sea a la cintura, al brazo o al antebrazo. Los valores que reciben estos elementos deben estar expresados en radianes, y su magnitud debe estar dentro de los márgenes de movimiento permitidos por cada articulación.

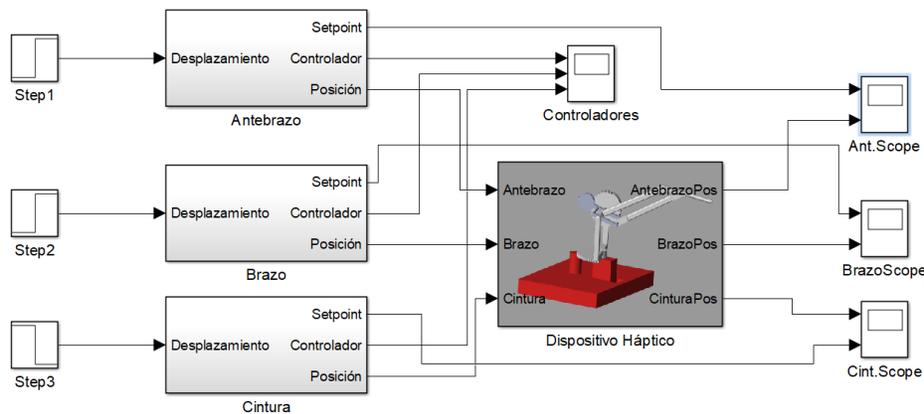


Figura 4 Diagrama a bloques en Simulink

Los movimientos fueron descritos con un escalón unitario con una magnitud igual al desplazamiento deseado y un tiempo de retardo definido por nosotros en base al orden que queremos ver en los movimientos del robot.

El controlador PD difuso fue diseñado para trabajar con dos entradas, una que evalúe el error multiplicado una ganancia (ganancia proporcional) y otra que evalúe la derivada del error multiplicada por una ganancia (ganancia derivativa). Si bien el

error se obtiene fácilmente, restando la posición real a la del setpoint, para obtener la derivada del error usamos un bloque de derivación, en la figura 5, en el subsistema principal, podemos observar la estructura del controlador PD difuso, así como la función de transferencia del motor y el bloque de conversión de grados a radianes a la salida de la planta. El subsistema secundario (Initial Setup) fue diseñado para poder establecer una posición inicial de acuerdo a nuestras necesidades, el mismo solo trabaja durante los primeros 3 segundos de la simulación, después de eso no afecta en lo absoluto al comportamiento del robot.

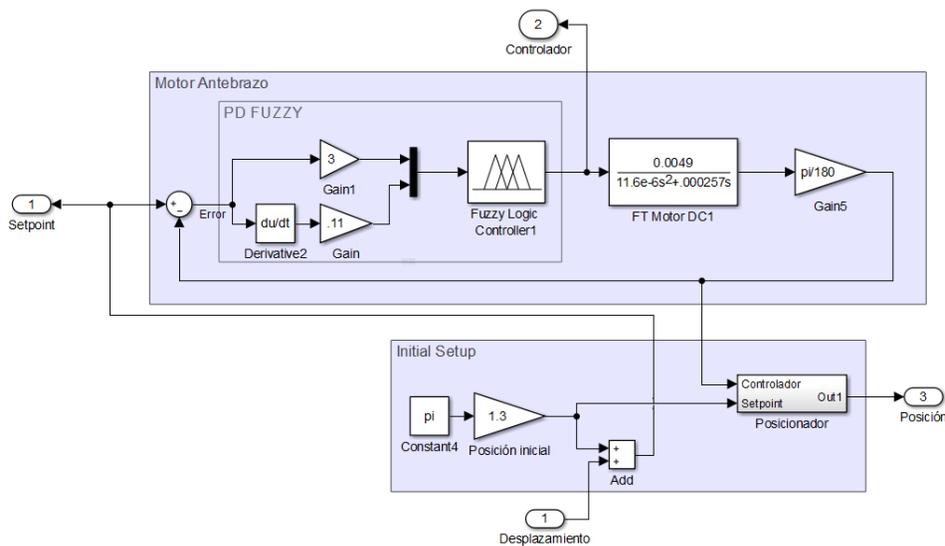


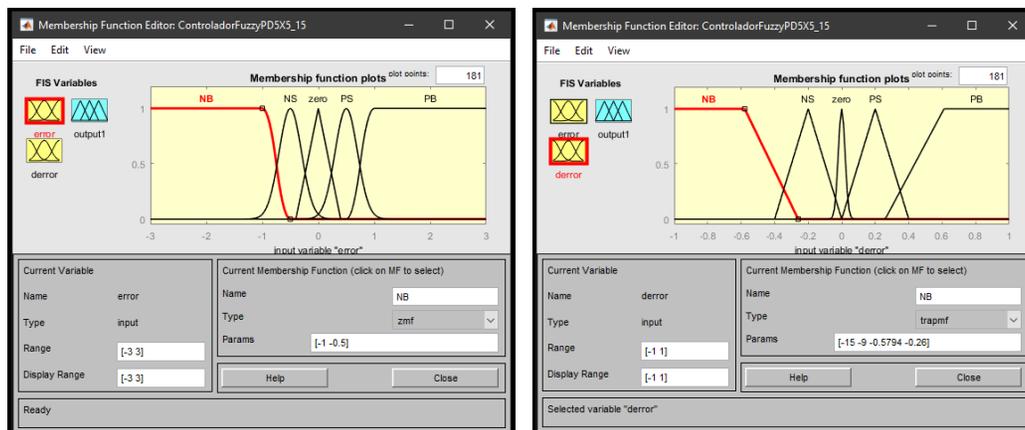
Figura 5 Subsistema con planta y controlador

### Controlador difuso

Dentro del controlador, la entrada “error” trabaja en base al rango de movimiento posible de la articulación que controla, por ejemplo, en el caso de la cintura, al ser medio engrane y tener como posición inicial el centro del engrane, solo le es posible moverse .5 radianes hacia la derecha o hacia la izquierda, por lo que el error máximo posible es de  $\pm\pi$  radianes, este dato entonces se utiliza para delimitar el rango de la función de membresía de la articulación.

A continuación, se muestran las funciones de membresía de uno de los tres controladores, en el entendido que todos son iguales, cambiando únicamente los valores límite de las funciones de membresía.

En la figura 6 se observa la gráfica con las funciones de membresía de la entrada “error” y la entrada “derror”, del controlador diseñado para la cintura. Los valores que se asignan dentro de la entrada de la derivada del error, dependen directamente de la ganancia que se define en el bloque de Simulink, en este caso particular, al haber definido una ganancia derivativa de 0.1 y hacer una simulación para visualizar el valor máximo y mínimo de la curva que describe la derivada del error, no eran de esperarse valores fuera de  $\pm 1$ , por lo que defino el rango de trabajo dentro de ese margen.



a) Error

b) Derror

Figura 6 Funciones de membresía de las entradas

Las dos entradas, categorizan el tamaño del error y su signo, considerando errores de tipo, grande negativo (NB), pequeño negativo (NS), pequeño positivo (PS), grande positivo (PB) y cero (zero), esto fue diseñado así debido a que conforme el error se reduce y se va llegando al setpoint es necesario desacelerar la pendiente para evitar un sobreimpulso y obtener una curva suave, por lo que si se detecta que el error ha disminuido considerablemente, la acción de control cambia a una magnitud menor. Asimismo, se espera también que el error llegue a ser cero en algún momento, en cuyo caso se suspende cualquier acción de control, esto evita que haya oscilaciones al llegar a la referencia, por lo que a su vez reduce el tiempo de estabilización del sistema.

Lo último con respecto al controlador tiene que ver con las funciones de membresía de su salida, las cuales se muestran en la figura 7. Como se dijo anteriormente, las

salidas están limitadas a trabajar dentro de un rango de  $\pm 15$  V, sin embargo, se separan en 7 categorías, negativas grandes (NB), negativas medianas (NM), negativas pequeñas (NS), cero (zero), positivas pequeñas (PS), positivas medianas (PM) y positivas grandes (PB).

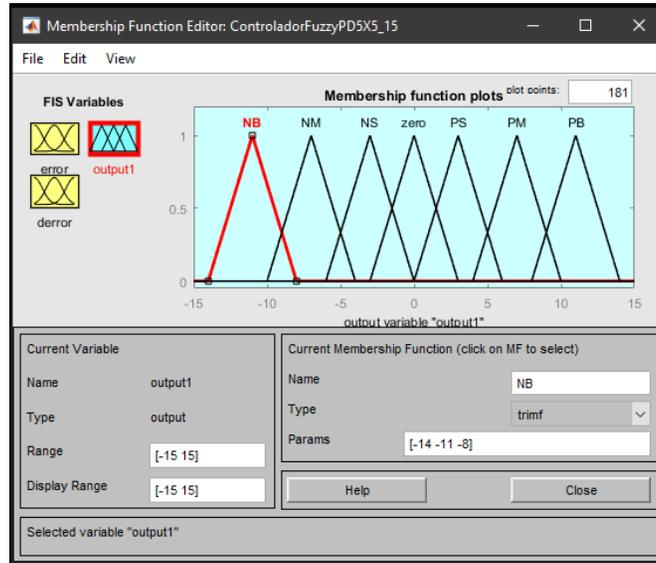


Figura 7 Función de membresía de la salida

Las reglas del controlador difuso se muestran en la tabla 2.

Tabla 2 Tarjeta de reglas del controlador PD difuso

SALIDA		ERROR				
		NB	NS	Z	PS	PB
DERROR	NB	NB	NM	Z	PM	PB
	NS	-	NS	Z	PS	-
	Z	NB	NM	Z	PM	PB
	PS	-	NS	Z	PS	-
	PB	NB	NM	Z	PM	PB

Las figuras 9, 10 y 11, presentan la respuesta en la cintura, el antebrazo y el brazo respectivamente, ante un movimiento de 0.5, 0.3 y 0.2 radianes respectivamente. El controlador fue comparado con un controlador tipo PID que actúa sobre una planta idéntica, el mismo fue sintonizado utilizando las utilidades que MATLAB provee, siendo las ganancias  $K_p=38.6$ ,  $K_d=1.49$  y  $K_i=86.3$ .

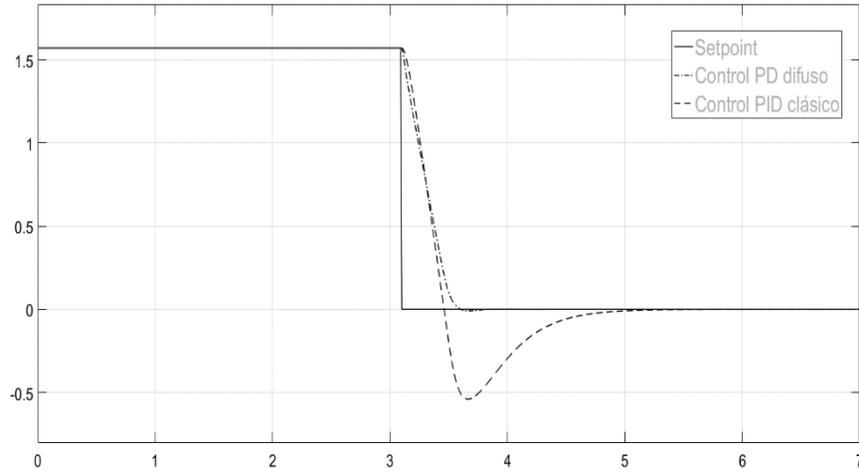


Figura 9 Respuesta de la cintura ante un desplazamiento de 0.5 radianes.

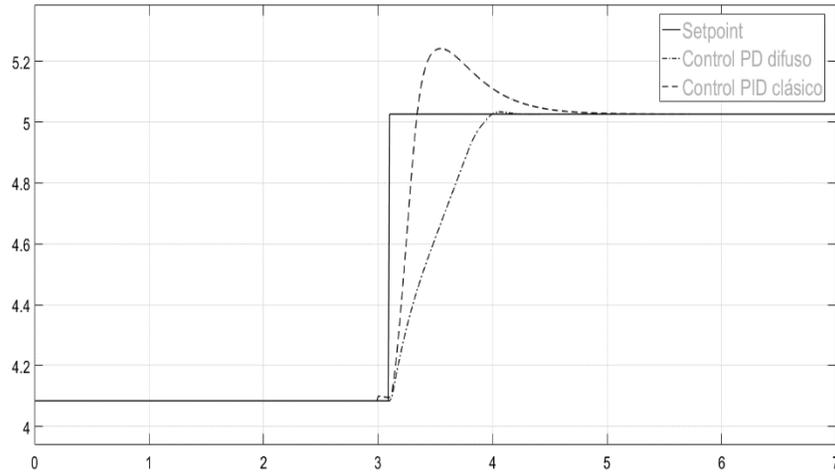


Figura 10 Respuesta del antebrazo ante un desplazamiento de 0.3 radianes.

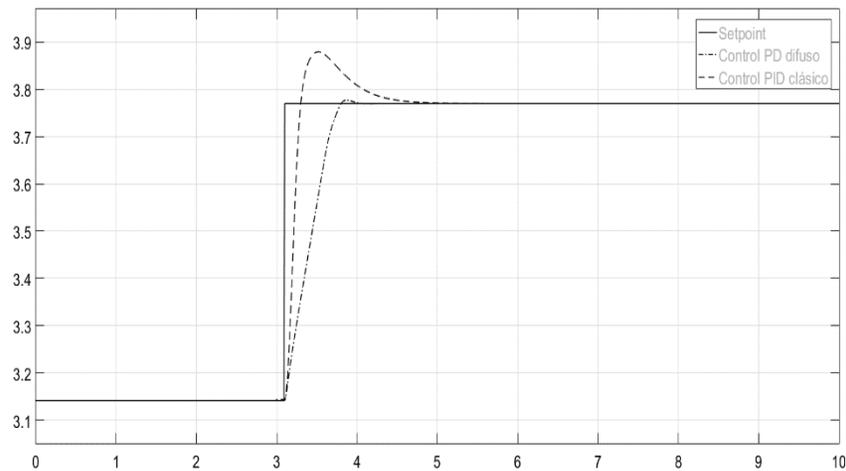


Figura 11 Respuesta del brazo ante un desplazamiento de 0.2 radianes

La figura 12 muestra la posición inicial del robot y su posición al final de los movimientos programados.

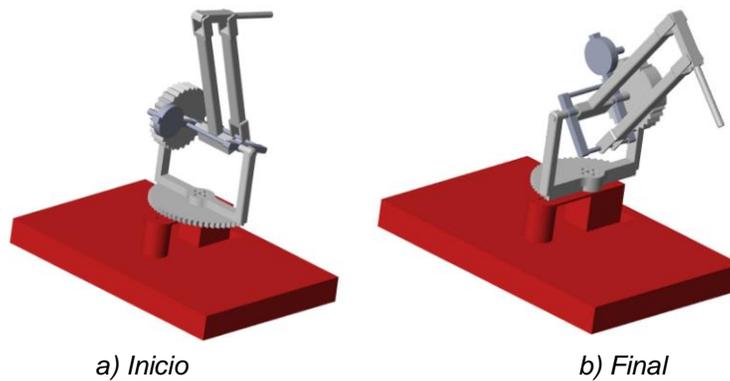


Figura 12 Posición inicial y final del dispositivo háptico.

#### **4. Discusión**

El trabajo con programas de computadora para simular el comportamiento de modelos 3D de piezas y ensamblajes digitales, ha resultado bastante práctico para este caso en particular, donde requerimos explorar el comportamiento de un controlador sobre un sistema físico que aún está identificándose, y sobre el que aún no es posible realizar modificaciones. Lo anterior es importante cuando se diseñan herramientas para la enseñanza, ya que se busca transmitir conceptos clave al estudiante mediante la experimentación, pero también se intenta optimizar los recursos empleados y evitar riesgo de daños sobre los dispositivos con que se cuenta.

En cuanto al controlador diseñado, el mismo demostró un comportamiento estable y relativamente rápido, mientras que su diseño no requirió de fórmulas o consideraciones especiales que retrasarían su diseño, sino que, mediante un diseño basado en lo empírico, se logró un controlador que fácilmente compite con otro que se diseñara para la misma actividad, esto se sustenta en las características de las respuestas observadas en las figuras 9, 10 y 11, sobre las cuales haremos un análisis a continuación. Para empezar, podemos hablar del tiempo que tarda el controlador en alcanzar la referencia, el cual va de 0.8 a 1.5 segundos, dependiendo del desplazamiento programado, un tiempo de respuesta lo suficientemente rápido,

considerando que la curva que describe la respuesta se caracteriza por conservar un incremento suave y constante, también es de notarse el tiempo de estabilización, el cual no es muy superior al tiempo que le toma el sistema alcanzar la referencia, lo que nos habla de lo rápido que el sistema alcanza un error de estado estable de 0. La gráfica no muestra ningún tipo de sobreimpulso que sea siquiera visible sin hacer un considerable aumento sobre la señal en cuestión, tampoco se aprecia una oscilación previa a la estabilización del sistema, algo que para nada sería deseable en un manipulador como el dispositivo háptico con que estamos trabajando. Por otro lado, el controlador PID clásico demostró ser útil solo hasta cierto punto, pues si bien no se hizo un estudio sobre su sintonización, con los parámetros asignados por el mismo Simulink la respuesta no fue superior en ninguno de los casos a la respuesta de su contraparte difusa, es interesante notar que en las tres articulaciones se observa un sobreimpulso considerable, pues excede demasiado el setpoint y se mantiene por casi por un segundo dependiendo del desplazamiento. Tomando entonces en consideración los resultados obtenidos, se puede conjeturar el haber diseñado un controlador robusto y bastante competitivo para su aplicación.

## **5. Conclusiones**

El uso de las paqueterías para el modelado 3D fue sin duda una de las piedras angulares de este proyecto, en el que se demostró muy bien que tan útiles pueden resultar para simulaciones de sistemas reales sobre los cuales no puede existir una manipulación por diversas razones. Además, Simulink demostró ser una poderosa herramienta no solo para la simulación de movimiento de modelos 3D, sino también como plataforma de control. En este trabajo se logró idear una interfaz en la cual, mediante modificaciones simples, se puede enseñar los diferentes factores que intervienen dentro de un controlador PD difuso, lo que se potencializa al contar con una simulación del movimiento del dispositivo estudiado, lo que de seguro es de gran valor didáctico. Por otro lado, la posibilidad de hacer un comparativo con un control conocido como lo es el PID clásico, permite visualizar de forma sencilla las diferencias entre ambos controles en términos de funcionamiento, lo que a su vez hace más sencilla la identificación de las fortalezas y limitaciones del controlador

PD difuso, a la vez que se ofrece la posibilidad de interactuar con el controlador, modificándolo a antojo sin poner en riesgo ningún equipo, para al final lograr comprender la naturaleza y el comportamiento del mismo.

Durante el desarrollo de este trabajo quedó demostrado que en controles difusos es posible hacer un diseño basado en la heurística sin comprometer la eficiencia del mismo, basta con tener ciertos conceptos clave acerca de la naturaleza del controlador para garantizar un funcionamiento óptimo.

Para trabajos posteriores queda obtener una función de transferencia exacta de los motores del dispositivo, de forma que la simulación de computadora se apegue más a la realidad, lo que será de especial utilidad cuando el proyecto llegue a la fase de diseñar un controlador real y sea necesaria hacer una comparativa entre los resultados reales y los simulados. También se puede explorar la posibilidad de integrar más tipos de control al programa creado en Simulink, a fin de ofrecer una experiencia más completa en término de cantidad de controladores disponibles para el usuario, en este aspecto, un controlador PID difuso puede ser la alternativa inmediata más próxima.

## **6. Bibliografía y Referencias**

- [1] Bethune, J. (2017). *Engineering Design and Graphics with SolidWorks 2016*. Pearson, EUA.
- [2] Dwiputra, R. (2014). Modelica model for the youbot manipulator. *Proceedings of the 10th International Modelica Conference*, 1205-1212.
- [3] Jaime Jalomo, A. L. (2020). Diseño de un Robot Háptico de Tres Grados de Libertad con Solidworks y Control Mediante Pid Clásico en Matlab. *Pistas Educativas*, 326-336.
- [4] Phillips, C. L. (1989). *Digital control system analysis and design*. Prentice-Hall, Inc.
- [5] Ross, T. J. (2010). *Fuzzy logic with engineering applications*. Wiley, United Kingdom.
- [6] Vallejo, E. (1997). Identificación paramétrica de sistemas dinámicos. *Revista Científica Ingeniería y Desarrollo*, 10-22