Desarrollo e Implementación de un Modelo Determinístico en Matlab para el Control de un Elevador Mediante Lógica Difusa y el Uso de Diagrama de Bloques

Primer Autor: I. **Aldair** Morales C. Segundo Autor: F. **Alfredo** Forero. G
Estudiantes de Ingeniería de Sistemas
Universidad de la Amazonia
Florencia – Caquetá
i.morales@udla.edu.co | forerogomez4@gmail.com

RESUMEN

El siguiente trabajo se basa en los conceptos de lógica difusa y el manejo de diagramas de bloques, en donde se presenta un problema de control, la tarea está en simular mediante el desarrollo y la implementación de un modelo matemático y el uso de la lógica difusa, mediante un controlador difuso controlar un ascensor, en donde se incluyen variables de modelado determinístico.

Dicho problema de control mediante la aplicación de la lógica difusa nos permite introducir nuevas conceptualizaciones de esquemas referentes al control, aplicados en el campo de la ingeniería, además del manejo de herramientas de programación matemática y modelamiento de este como lo es Matlab.

Palabras clave: Lógica difusa, controladores, modelo matemático, Matlab, programación, diagramas de bloques, transformada de Laplace.

I. Introducción

Este articulo nos proporciona bases fundamentales para poder desarrollar e implementar mediante el uso del entorno de desarrollo integrado Matlab un modelo determinístico para el control de un ascensor, además de ello basado en lógica difusa y la utilización de diagramas de bloques, en donde podemos tener un enriquecimiento de conocimiento en cuanto a los mecanismos de control.

El desarrollo que se va a estar realizando representa un problema básico de modelos determinísticos, además basándonos en la lógica difusa para poder crear un controlador difuso e implementarlo sobre un diagrama de bloques bien estructurado con el sistema de control de un ascensor.

El problema planteado en este artículo está basado en la asignatura de Inteligencia Artificial, orientada por el docente **Milher Fabian Tovar Rubiano** del programa de Ingeniería de Sistemas, más exactamente en la asignatura de Inteligencia Artificial.

La asignatura como tal, presenta el adelanto de las bases sólidas para lo que se introduce a la Inteligencia Computacional, en donde los estudiantes ponen en práctica los conocimientos adquiridos en base a redes neuronales, algoritmos evolutivos, genéticos y para este caso en específico las bases fundamentales de la lógica difusa.

En este artículo, se presentan varias secciones en donde veremos un enfoque a lo que queremos realizar mediante el uso de la lógica difusa, posteriormente la identificación de los factores matemáticos (desarrollo del modelo matemático) y lo que se refiere a toda la programación que se realiza directamente en Matlab.

Además de ello nos basaremos fundamentalmente en teoremas matemáticos implementados en el mundo de la electrónica, específicamente en el campo de circuitos eléctricos para poder tener una estructuración concreta del sistema de control haciendo uso de la transformada de Laplace.

.

II. PRIMER ACERCAMIENTO AL DESARROLLO E IMPLEMENTACIÓN DEL MODELO MATEMÁTICO (DIAGRAMACIÓN)

Lo que queremos realizar en este trabajo, ya anteriormente lo habíamos mencionado, es desarrollar el sistema de control de un ascensor mediante el uso de la lógica difusa, además de ello poder tener una señal de control que se estabilice y podamos determinar que el sistema funcionar correctamente, así poder hacerlo implementando un modelo matemático de tipo determinístico, además de ello que este realizado en Matlab mediante diagramas de bloques, más ampliamente usando Simulink.

El primer acercamiento que podemos hacer en este caso es tener en cuenta como es el funcionamiento del ascensor.

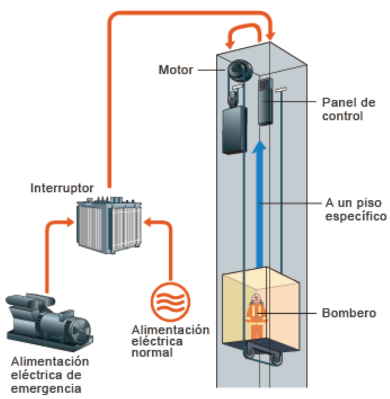

Figura 1: Estructura de un Ascensor[1]

En este caso nos centraremos en aplicar nuestros conocimientos al desarrollo del sistema de control del ascensor, en donde tendremos en cuenta la parte de alimentación eléctrica, motor, panel de control.

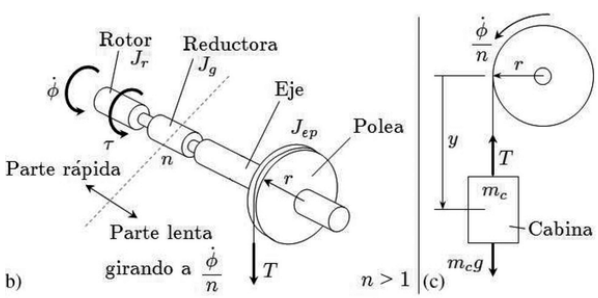

Figura 2: (b) Partes rotativas del sistema del Ascensor (c) Fuerzas que actúan sobre la cabina del Ascensor[2].

Teniendo en cuenta lo anterior, nuestro sistema tendrá una entrada, proceso y una salida, además de una retroalimentación por medio de un sensor.

- ✓ Entrada: Voltaje.
- ✓ Salida: Velocidad Angular.
- ✓ Proceso: Velocidad Angular / Voltaje.
- ✓ Sensor: El sensor que es un control adicional que podemos implementar lo dejamos como opcional.

Diagrama 1: Sistema de control con su entrada, proceso y salida[3].

Entonces lo que se quiere obtener con esto es una función de transferencia, la cual se define como:

$$Función\ de\ Transferencia = \frac{Salida\ \dot{\theta}(t)}{Entrada\ e} \qquad (1)$$ [4]

III. DESARROLLO DEL MODELO MATEMÁTICO APLICANDO TRANSFORMACIÓN DE LAPLACE

Para poder implementar el modelo matemático y posteriormente hacer una transformada de Laplace, se deben afianzar varios conceptos previos como lo es el tener en cuenta sobre el rotor de un motor y el funcionamiento de un motor eléctrico.

Ahora, desarrollar el modelo matemático implica tener en cuenta el diagrama de bloques, entonces hacemos referencia al diagrama de circuitos proporcionado por el docente Milher Fabian y trabajar con el desarrollo del modelo matemático.

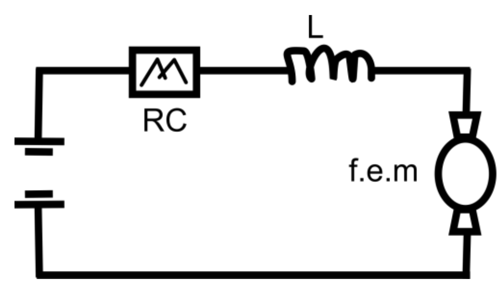

Figura 3: Circuito del sistema de control del Ascensor[5].

En base al circuito anterior, podemos plantear el modelo matemático determinístico de la siguiente manera, teniendo en cuenta los procedimientos matemáticos pertinentes.

$$\dot{\theta} = velocidad\ angular\ (\frac{\partial\theta}{\partial t})$$

[1] Figura 1: Estructura de un Ascensor, imagen tomada de http://www.mitsubishielectric.com/elevator/es/products/basic/elevators/nexiez_mrl/index_safety.html teniendo en cuenta seguridad en ascensores y escaleras.

[2] Figura 2: (b) Partes rotativas del sistema del Ascensor c) Fuerzas que actúan sobre la cabina del Ascensor, tomado de https://www.researchgate.net/figure/Partes-del-modelo-dinamico-del-ascensor-a-Circuito-equivalente-del-motor-DC-b_fig8_324114705

[3] Diagrama del proceso del sistema de control con su entrada, proceso y salida.

[4] Función de Trasferencia.

[5] Circuito del sistema de control del ascensor.

$$-e + v_r + v_l + v_{f.e.m} = 0 \qquad (2)$$

Con respecto a las anteriores ecuaciones tenemos:

$$v_r = iR$$
$$v_i = \frac{\partial i}{\partial t} L$$
$$v_{f.e.m} = k_c \dot{\theta}(t)$$

Resolviendo (2)

$$-e(t) + i(t)R + \frac{\partial i}{\partial t} L + k_c \frac{\partial \theta}{\partial t} = 0 \qquad (3)$$

Con respecto a la ecuación anterior, podemos pensar en la forma de implementar o hacer uso de una función de transferencia que anteriormente se había descripto.

$$F.T = \frac{(\dot{\theta}(t))}{(e)}$$

Es Así como podemos implementar una función de torque que se genera a partir de la correspondencia entre el engranaje y la polea que hace ascender y descender el elevador.

$\sum T = J\ddot{\theta}$ Desarrollando la ecuación tendríamos

$$\sum T = J \frac{\partial^2 \theta(t)}{\partial t^2}$$
$$Tm = J\ddot{\theta}(t)$$
$$Tm = K_{c1} i(t)$$

Entonces se puede decir que la función de Torque (T) la podemos resolver y obtener:

$$K_{c1} i(t) = J \frac{\partial^2 \theta}{\partial (t)^2} \qquad (4)$$

Y a esta ecuación (4) le aplicamos la Transformada de Laplace para obtener una función de transferencia con valores complejos.

$$L\{f(t)\} = F(s) \qquad (5)^{6}$$

$$L^{-1}\{F(s)\} = f(t) \qquad (6)^{7}$$

Para dar solución y poder encontrar los valores de la función de transferencia, se tendrás las condiciones iniciales = 0.

Resolviendo (5)

$$L\left\{\frac{Kdf(t)}{dt}\right\} = Ks + F(s) + F(s_0)$$

$$L\left\{\frac{Kd^2 f(t)}{dt^2}\right\} = Ks^2 + (s) + SF(s_0) - F(s_0)$$

Entonces

$$L\left\{-e(t) + i(t)R + \frac{\partial i(t)}{\partial (t)} L + K_c \frac{\partial (\theta)}{\partial (t)}\right\} = L\{0\}$$

$$L\{-e(t)\} + L\{i(t)R\} + L\left\{\frac{\partial i(t)}{\partial (t)} L\right\} + L\left\{K_c \frac{\partial (\theta)}{\partial (t)}\right\} = L\{0\}$$

$$-e(s) + i(s)R + Lsi(s) + Kcs\theta(s) = 0 \qquad (7)$$

Entonces obtenemos una función de la forma:

$$-e(s) + i(s)[R + Ls] + Kcs\theta(s) = 0 \qquad (8)$$

$$L\{Kc1 * i(t)\} = L\left\{J \frac{\partial^2 \theta(t)}{\partial t^2}\right\}$$

$$Kc1 * i(s) = Js^2\theta(s) \qquad (9)$$

Ahora despejando y resolviendo (9)

$$i(s) = \frac{Js^2\theta(s)}{Kc1}$$

Teniendo en cuenta la ecuación (8)

$$-e(s) + i(s)[R + Ls] + Kcs\theta(s) = 0$$

$$-e(s) + s\theta(s)\left[\frac{J}{Kc1} s[R + Ls] + Kc\right] = 0$$

$$e(s) = w(s)\left[\frac{J}{Kc1} s[R + Ls] + Kc\right] = 0$$

Donde

- ✓ $e(s)$ es la entrada
- ✓ $w(s)$ *es la salida*

Entonces

$$1 = \frac{w(s)}{e(s)}\left[\frac{J}{Kc1} s[R + Ls] + Kc\right]$$

Con la anterior ecuación, ya podemos hacernos una idea de la función de transferencia, la cual queda de la siguiente manera:

$$\frac{w(s)}{e(s)} = \frac{1}{\frac{J}{Kc1} s[R + Ls] + Kc} \qquad (10)^{8}$$

IV. DIAGRAMA DE BLOQUES EN MATLAB

Para el cálculo de nuestra función de transferencia tenemos tres valores por defecto, que son:

- ✓ $J = 1$
- ✓ $Kc1 = 0.8$
- ✓ $Kc = 1.2$

[6] Función de la Transformada de Laplace. Función tomada de https://es.wikipedia.org/wiki/Transformada_de_Laplace Enciclopedia Libre.

[7] Función inversa a la Transformada de Laplace

[8] Función de transferencia encontrada a partir de la transformación que se hizo por medio de Laplace.

Pero este último Kc se diferencia a cada grupo de trabajo, en nuestra tabla corresponde a los valores de la $f.e.m$

- ✓ $f.e.m\ (fuerza\ electromotriz) = 1.5$

Así mismo tenemos los valores de R y L:

- ✓ $R = 2$
- ✓ $L = 0.2$

Haciendo los cálculos obtenemos:

$$\frac{w(s)}{e(s)} = \frac{1}{\frac{(1)(0.2)}{0.8}s^2 + \frac{(1)(2)}{0.8}s + 1.5}$$

$$\frac{w(s)}{e(s)} = \frac{1}{0.25s^2 + 2.5s + 1.5}$$

Obtenida nuestra función de transferencia lo que hacemos en implementarla en Matlab, para este caso trabajaremos bajo la versión Matlab R2015a.

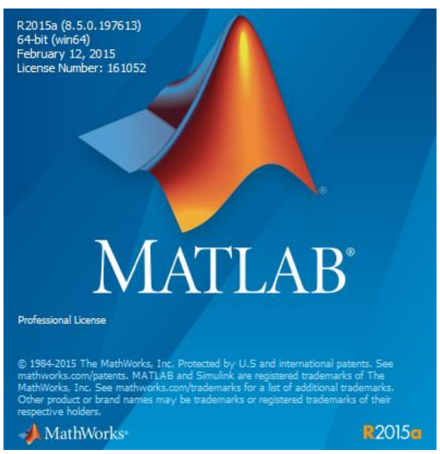

Figura 4: Pantalla de inicialización de Matlab en Windows 8.1 [9]

Para poder diseñar mediante Simulink nuestro diagrama de bloques, lo que hacemos es seleccionar la ubicación de trabajo (current folder) y posteriormente damos clic en la herramienta de Simulink o simplemente en la línea de funciones de Matlab $fx \gg$ escribimos la palabra Simulink.

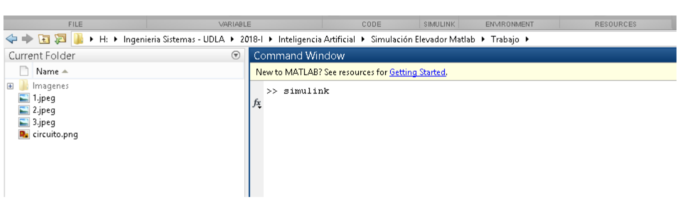

Figura 5: Current Folder (Carpeta actual) en donde vamos a guardar todos nuestros archivos creamos en Matlab.

El diagrama de bloques que se implementó en el Entorno de Desarrollo Integrado matemático Matlab en su versión R2015a, es el siguiente:

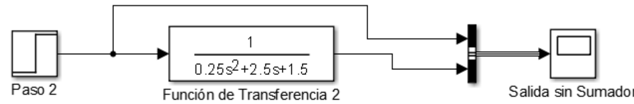

Figura 6: Función de Transferencia [10]

En donde la función de transferencia de calcula mediante los valores obtenidos anteriormente, teniendo como grafica la siguiente:

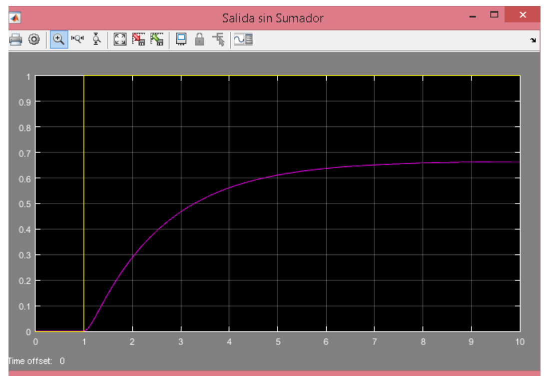

Figura 7: Grafica de salida del sistema de control que se realizó en Simulink mediante diagrama de bloques.

V. DESARROLLO E IMPLEMENTACIÓN DEL CONTROLADOR DIFUSO EN MATLAB

Para la implementación del controlador difuso, lo que debemos de hacer en Matlab, es irnos a la ventana principal y buscar dentro de sus Apps el diseñador de lógica difusa (Fuzzy Logic Designer) y dar clic, en el vamos a configurar nuestro controlador difuso, además de ello debemos de tener en cuenta que va a ser de tipo determinístico SUGENO.

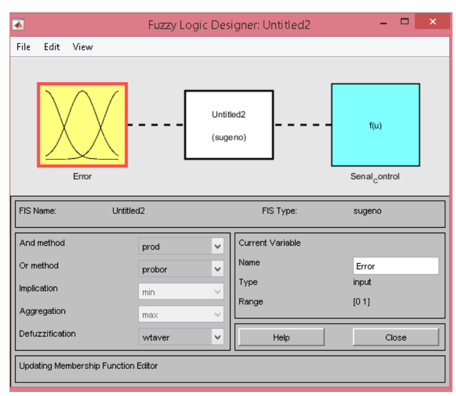

Figura 7: Primer acercamiento al diseño de nuestro controlador Difuso.

[9] Pantalla de inicialización de Matlab R2015a – Pantallazo tomado de ordenador propio.

[10] Figura 6: Función de transferencia, y se conoce como la primera función del sistema retroalimentado, es decir, que el sistema sufre modificaciones.

Se puede observar que hemos modificado lo que corresponde a la entrada y la salida, es decir, asignamos de nombre Error (entrada) y Señal de Control (salida).

Posteriormente para poder configurar mi universo difuso y tener en cuenta las reglas de inferencia para el desarrollo de mi controlador difuso que se va a implementar en el diagrama de bloques del ascensor, tenemos que dar doble clic sobre el Error y allí configurar el controlador.

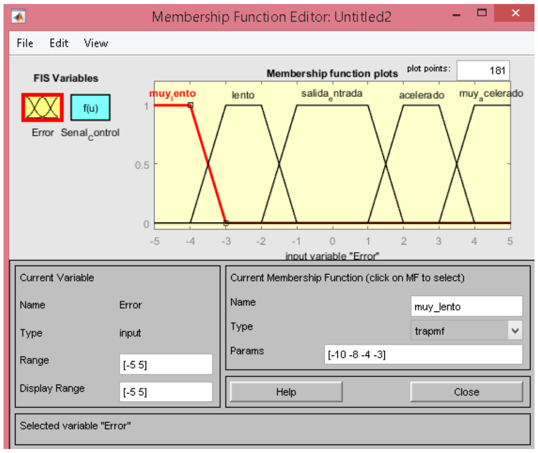

Figura 8: Configuración de las funciones que va a tener nuestro controlador difuso

Teniendo en cuenta lo que se conoce como validación horizontal de los datos, hemos tomado los rangos desde -4, -3, -2, -1, 0, 1, 2, 3, 4, de tal forma que podamos implementar nuestros conjuntos difusos en el controlador.

Además de ello debemos tener en cuenta que estamos trabajando sobre una función lineal, por tanto, debemos manejar valores para controlar nuestro sistema cuando se encuentre en cualquiera de los cinco estados que tenemos.

La configuración de los valores que vamos a tener en cuenta, lo hacemos mediante las reglas de inferencia que podemos manejar en nuestro controlador difuso.

Para poder asignar los valores de control, lo debemos de hacer en la señal de control que tenemos como salida, además de ello poder tener en cuenta que estos valores son pertenecientes a lo que declaramos en el mismo conjunto difuso, es decir, cuando este en la zona de muy lento, este debe de acelerarse más, pero esto lo vamos a ver con mayor detalle en las reglas de inferencia.

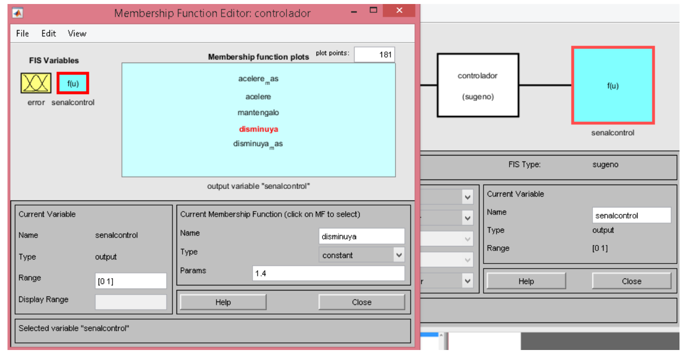

Figura 9: Conjuntos difusos con valores con respecto a la entrada.

Una vez tengamos estos valores definidos en orden de 1 hasta 5, entonces lo que hacemos es ir a las reglas de inferencia para poder aplicar las respectivas condiciones.

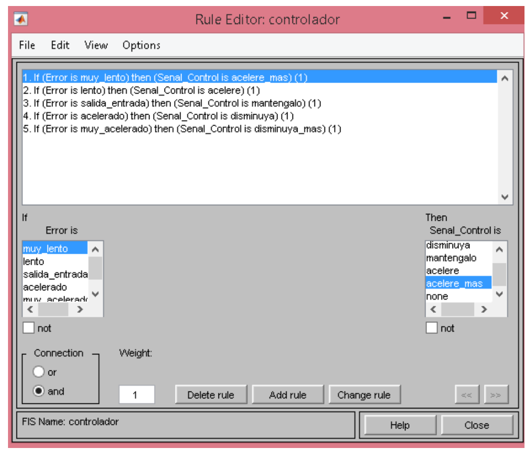

Figura 10: Reglas de inferencia

Una vez hecho esto, se puede simular mediante View > Rules, las reglas de inferencia que acabamos de declarar.

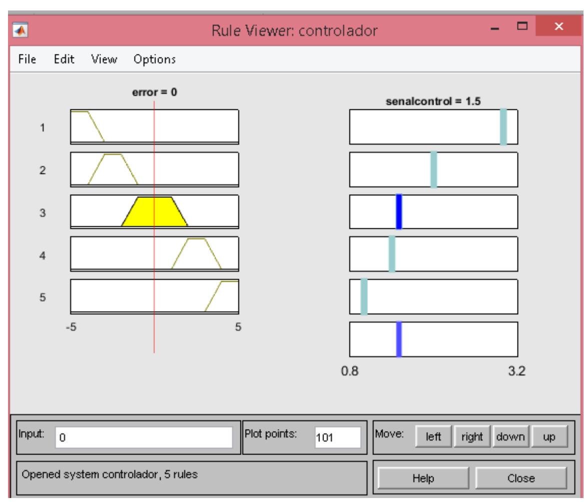

Figura 11: Simulación de reglas de inferencia

Lo que tenemos que hacer a continuación es montar nuestro controlador difuso construido mediante la App de Fuzzy Logic Designer en nuestro diagrama de bloques, y para ello agregamos a al mismo un controlador difuso desde la librería de Matlab en Simulink, buscando dentro de las opciones de Fuzzy Logic Toolbox, y seleccionamos el de nombre “Fuzzy Logic Controller”.

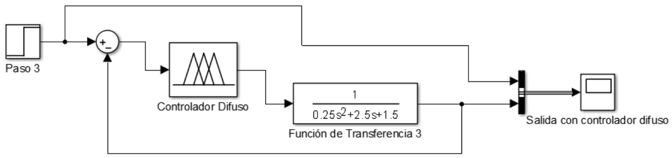

Figura 11: Diagrama de bloques con el controlador difuso.

Para poder ejecutar nuestro sistema de control, debemos de exportar al Workspace nuestro controlador con la extensión .fis, es decir, tenemos que incorporarlo a nuestro entorno de trabajo, para ello damos clic en File>export>to Workspace

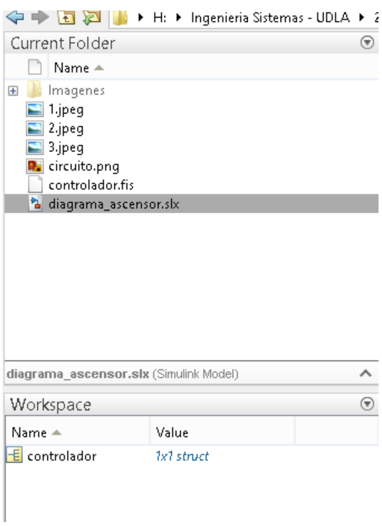

Figura 12: Exportación al Workspace de nuestro controlador difuso con extensión .fis.

Una vez ejecutamos la simulación de nuestro sistema de control, se observa que la línea de demarcación del error está estabilizada, es por ello por lo que se evidencia un buen funcionamiento del sistema.

Además, el resultado de la función es en términos generales positivo porque no tenemos casi nada de margen de error en cuanto a la estabilización del sistema.

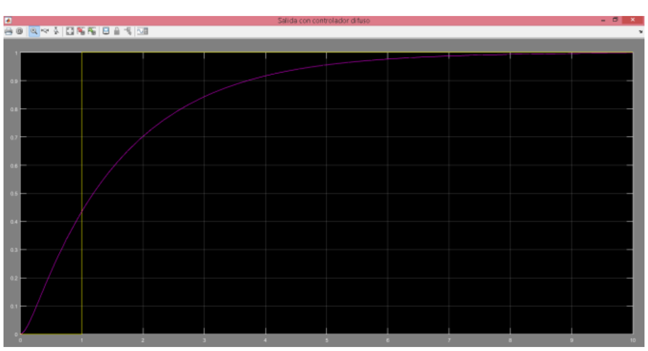

Figura 13: Gráfica final de nuestro sistema de control, implementando el controlador difuso.

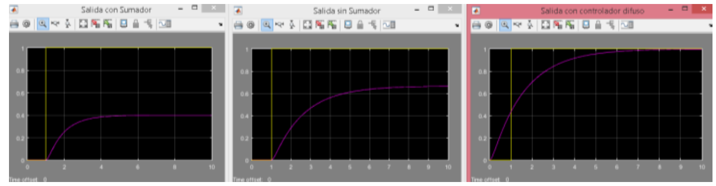

Figura 14: Comparación de los tres diagramas de bloques implementados, en donde el ultimo representa la implementación del controlador difuso.

VI. CONCLUSIONES

De acuerdo con el planteamiento del problema y a su respectiva solución mediante un sistema de control de un ascensor que tenia como objetivo estabilizar la señal de control, se evidencio que mediante el uso de diagrama de bloques y la implementación de la lógica difusa en cuanto a términos computacionales, se pudo resolver el problema de controlar el sistema, en este caso el ascensor.

Así mismo, se debe de aclarar que todo sistema por muy robusto que sea, tiene errores, lo que se puede hacer ante cualquier caso es optimizar mediante el uso de herramientas, para nuestro problema ya se había mencionado el uso de la lógica difusa aplicándola en los diagramas de bloques.