DISEÑO E IMPLEMENTACION DE CONTROLADORES INTELIGENTES PARA LA UNIDAD AIR *FLOW TEMPERATURE CONTROL SYSTEM*

PAUL ADRIAN CABEZAS YÁNEZ

DANIEL ALEJANDRO CAPELO PARREÑO

Facultad de Ingeniería Eléctrica y Electrónica, Escuela Politécnica del Ejército

Av. El Progreso S/N, Sangolquí, Ecuador

adriancabezasy@hotmail.com

daniel_capelo13@hotmail.com

***Resumen.*- El presente artículo describe la aplicación de las teorías de control inteligente para el control de temperatura de la unidad Air Flow Temperature Control System, que integra la adquisición de datos y el software Matlab.**

Los controladores inteligentes implementados se basan en la teoría de la lógica difusa, redes neuronales y predictivo. Este proyecto pretende brindar más información sobre el diseño e implementación de los controladores inteligentes mencionados en un sistema real de temperatura, los modelos de simulación y los modelos experimentales se desarrollan en la herramienta Simulink de Matlab.

Palabras clave. Unidad Air Flow Temperature Control System, controladores inteligentes, Lógica difusa, redes neuronales, control predictivo, modelo matemático.

I. INTRODUCCIÓN

Los controladores más usados a nivel industrial son los controladores PID y si bien representan un método eficiente y de bajo costo, requiere un amplio trabajo de ingeniería. Los métodos de control inteligente representan un método alternativo de solución de problemas de control.

Actualmente el software MATLAB es comúnmente utilizado como simulador para el diseño de sistemas de control por su gran capacidad de cálculo, versatilidad de librerías y porque dispone de una amplia colección de funciones aplicables al campo de control inteligente, que mediante la herramienta Real-Time Workshop (RTW), permite probar los algoritmos modelados en tiempo real.

Con el fin de contribuir al desarrollo de investigación del Departamento de Eléctrica y Electrónica, el proyecto pretende consolidar la teoría impartida en la cátedra de Control Inteligente, mediante la creación de un entorno para realizar prácticas en los laboratorios propios del Departamento. Así se podrá analizar pruebas en tiempo real de los controladores diseñados.

II. MARCO TEORICO

Redes Neuronales

Las redes neuronales en el ambiente industrial se los utilizan como identificadores de procesos o como sistemas que optimizan el funcionamiento de un controlador.

En los controladores neuronales existen varias maneras de realizar el control, así como también el entrenamiento. Dos posibles esquemas son:

- Control neuronal indirecto
- Control neuronal directo

En este proyecto se diseñaron e implementaron dos tipos de redes neuronales, el controlador neuronal indirecto por modelo inverso, donde puede ser entrenada para imitar el comportamiento de la planta pero de forma inversa. La entrada a la red es

la salida del proceso y la salida de la red es la entrada del proceso y el controlador neuronal directo por modelo de referencia que consiste en diseñar un sistema que modifique el comportamiento natural de la planta con el objetivo que se aproxime a la respuesta que tiene un modelo de referencia establecido [6].

Control Difuso

Esta estrategia de control se basa en la lógica difusa, es decir, busca imitar la forma en que una persona tomaría decisiones según lo relativo de lo observado o experimentado. La lógica difusa proporciona la facilidad de resolver situaciones complejas y dinámicas, que son más fácilmente expresadas con palabras que con modelos matemáticos.

Un controlador difuso contiene un algoritmo que convierte una estrategia de control lingüística en una de control automático, mediante una base de reglas que genera un conjunto de expresiones del tipo:

Si <Condiciones> Entonces <Acciones>

Control Predictivo

El control predictivo basado en modelo o MPC, es una técnica de control que mediante la utilización de un modelo matemático interno del proceso, predice la evolución de las variables a controlar para calcular con un procedimiento de optimización, las señales de control que lleven al proceso a los objetivos deseados.

En cada instante t de tiempo, se predicen las salidas futuras para un horizonte de predicción N, usando un modelo del proceso a controlar. Estas salidas $\hat{y}(t+k|t)$ para k = 1, 2,…, N dependen de las entradas y salidas pasadas, cuyos valores son conocidos y las señales de control futuras $u(t+k|t)$ para k=1,2,…,N-1 que son las que se calcularán y se enviaran al sistema.

Las señales de control futuras serán calculadas por medio de la función de optimización, la cual se ajustara lo más cercano posible el proceso a la referencia dada $w(t+k|t)$.

La señal de control $u(t|t)$ se envía al proceso, mientras que las demás señales de secuencia de control se eliminan. En el siguiente instante de muestreo se calcula la secuencia actualizada, esto se realiza porque ya se conoce el y$(t+1)$ y se obtiene $u(t+1|t+1)$, haciendo uso del concepto de horizonte deslizante.

III. CARACTERÍSTICAS DE LA UNIDAD AIR FLOW TEMPERATURE CONTROL SYSTEM

La Figura 1 muestra un diagrama del módulo de temperatura, a continuación se describen sus componentes:

- **Elemento primario y de medición:** consta de un sensor-transmisor de temperatura tipo termistor. Toma la temperatura a la salida del calentador (20 – 70 [ºC]), y envía una señal analógica de voltaje (0 – 5 [V]) hacia el controlador.

- **Conducto de flujo:** mediante un ventilador, renueva el aire que se encuentra en su interior.

- **Niquelina:** se encuentra en la entrada del conducto, es la encargada de calentar el aire que ingresa mediante una señal de voltaje de 0 a 10 [V]
- **Placa Metálica:** posee cuatro orificios, que determinan la cantidad de aire que ingresará al proceso.

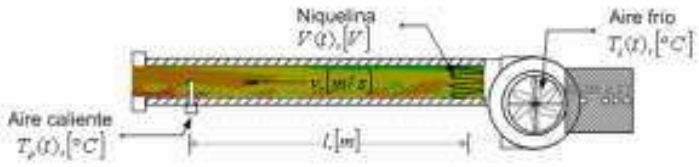

Figura 1. Unidad Air Flow Temperature Control System

Para asegurar que se mida una solución homogénea, el punto de supervisión se localiza a cierta distancia del punto de mezcla.

IV. DESARROLLO DEL MODELO MATEMATICO

El flujo de calor en el sistema de temperatura se da por convección, es decir, se transfiere energía entre una superficie, que es la niquelina, y un fluido que se mueve sobre esta que en este caso es el aire.

Mediante el ventilador se fuerza al aire a fluir a través de la niquelina y el conducto, a esto se lo denomina convección interna forzada y la transferencia de calor viene descrita por la ley de Newton del enfriamiento:

$$Q = h\,As\,(T\infty - T) \quad (1)$$

Donde:

h = Coeficiente de de transferencia de calor por convección

As = Área superficial a través de la cual tiene lugar la transferencia de calor.

T∞ = Temperatura de la niquelina

T = Temperatura del Aire

La transferencia de calor del sistema es en régimen transitorio, es decir, la temperatura no cambia sólo con la posición dentro del cilindro, sino que también cambia con el tiempo.

El calor acumulado en el cilindro viene dado por la siguiente fórmula:

$$Q = m\,Cp\,\frac{dT}{d\theta} \quad (2)$$

m = Masa del fluido

Cp = Calor especifico del fluido (aire)

T = Temperatura del fluido

θ = tiempo

Igualamos las ecuaciones 1 con 2 y resolvemos para llegar a la ecuación que describe el sistema transiente de temperatura:

$$h\,As\,(T\infty - T) = m\,Cp\,\frac{dT}{d\theta}$$

$$m\,Cp\,\frac{dT}{d\theta} - h\,As\,(T\infty - T) = 0$$

$$t = T\infty - T \;;\; \frac{dT}{d\theta} = \frac{dt}{d\theta}$$

$$\frac{dt}{d\theta} + \frac{h\,As}{m\,Cp}t = 0$$

$$\frac{dt}{d\theta} = -\frac{h\,As}{m\,Cp}t$$

$$\int \frac{dt}{t} = -\int \frac{h\,As}{m\,Cp}d\theta$$

$$\ln t = -\frac{h\,As}{m\,Cp}\theta + \ln a1$$

$$\ln(T\infty - T) = -\frac{h\,As}{m\,Cp}\theta + \ln a1$$

$$a1 = (T\infty - T)$$

$$Condiciones\ Iniciales:\ \theta = 0\,;T = To$$

$$\ln(T\infty - T) = -\frac{h\,As}{m\,Cp}\theta + \ln(T\infty - To)$$

$$\ln\left(\frac{T\infty - T}{T\infty - To}\right) = -\frac{h\,As}{m\,Cp}\theta$$

$$\frac{T\infty - T}{T\infty - To} = e^{-\left(\frac{h\,As}{m\,Cp}\right)t}$$

$$T = T\infty - (T\infty - To)e^{-\left(\frac{h\,As}{m\,Cp}\right)t} \quad (3)$$

La ecuación 3 describe el comportamiento de la temperatura en función del tiempo del sistema como se observa en la figura 2:

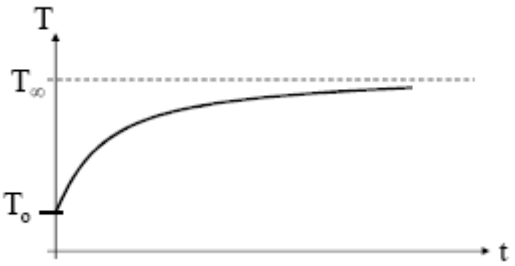

Figura 2. Comportamiento del la temperatura en función del tiempo.

La constante que determina el retardo de la función de temperatura se calcula de:

$$\frac{h\,As}{m\,Cp}$$

Los valores de coeficiente de convección (h), área superficial (As), masa (m) y calor específico del aire (Cp) son conocidos, pero para un mejor ajuste del modelo matemático con la respuesta real del sistema, la constante de retardo se la determinó mediante identificación como se explica a continuación.

V. IDENTIFICACION

Para la identificación de la constante de retardo del modelo que es el parámetro desconocido es necesario conocer el comportamiento real de la planta de temperatura, en Simulink se diseño un simple diagrama para que con la tarjeta de adquisición PCI 6221 guardar los datos de respuesta del sistema en lazo abierto.

Este procedimiento se lo realizó para 10 voltajes de entrada (1-10V), podemos observar las señales de respuesta en grados centígrados en la Figura 3.

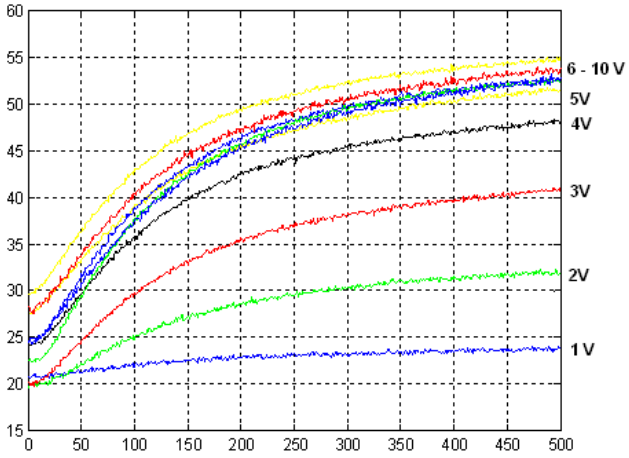

Figura 3. Respuestas de la planta de temperatura para diferentes voltajes.

Se observó que para voltajes de entrada de 6 a 10 voltios la respuesta del sistema fue casi la misma, por lo que se redujo el rango de voltajes de 0 a 6 voltios.

La ecuación 3 que caracteriza al sistema expresada como función de transferencia en diagrama de bloques se aprecia en la Figura 4, donde la entrada es el voltaje de 0 a 6 voltios, la salida la temperatura en grados centígrados, K la ganancia del sistema y R la constante de retardo que será identificada.

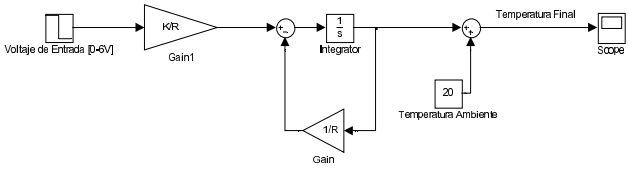

Figura 4. Diagrama de simulación del modelo matemático

La ganancia del sistema (K) relaciona el voltaje de entrada con la temperatura que es la salida del modelo. Por ejemplo para 2 voltios la temperatura alcanzó 31.7 °C (Figura 4), la temperatura subió 11.7 grados la temperatura ambiente (20°C) por lo que la ganancia para este valor es de 5.8.

La ganancia para cada voltaje de entrada se obtuvo usando el máximo valor que alcanzó la temperatura para cada muestra que se tomó, estos datos se los muestra en la Tabla 1

Voltios	Ganancia (K)
1	3,57
2	5,87
3	7,06
4	6,94
5	6,29
6	5,43

Tabla 1 Valores de ganancia para cada voltaje de entrada

Podemos ver en la Figura 5, la variación de la ganancia con respecto al voltaje de entrada.

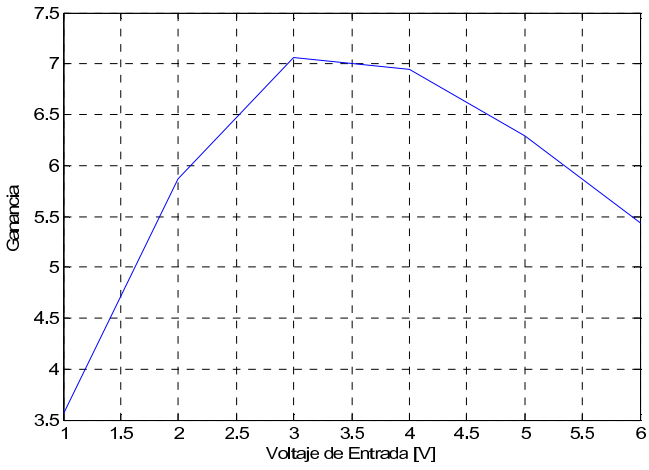

Figura 5. Gráfica de ganancia del Modelo con respecto al voltaje de entrada

Para introducir la variación de la ganancia en el modelo, se ajustó esta curva a un polinomio de segundo orden y se creó una función en Matlab que recibe el voltaje de entrada y nos devuelve el valor de la ganancia para este valor.

Para identificar el parámetro desconocido que es la constante de retardo, se desarrollo un programa que mediante el cálculo integral del error cuadrático, nos

entrega el valor de la constante que más ajusta las respuestas del modelo de simulación con los datos guardados de la planta. Así por ejemplo para 3 voltios de entrada, la constante de retardo resultó 13, con este valor en la Figura 6 podemos observar un comportamiento similar entre respuesta de la planta (color azul), con la gráfica que resulta del modelo de simulación (color rojo).

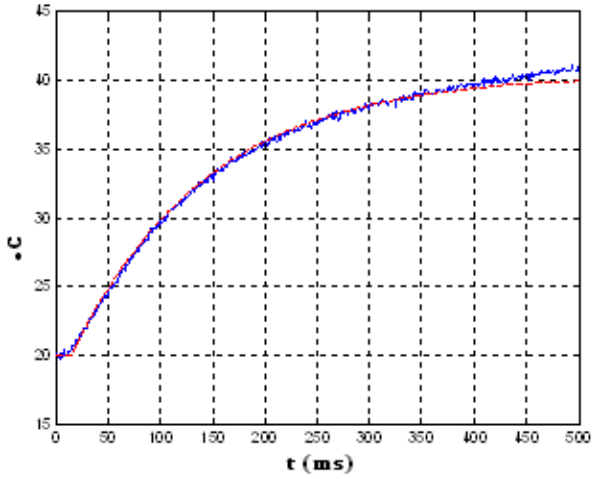

Figura 6. Comparación de respuestas del modelo de simulación con la planta para un voltaje de entrada de 3V.

Este procedimiento se lo realizó para cada muestra obtenida en la recolección de datos, la constante de retardo es el promedio de la identificada para cada voltaje de entrada como se indica en la Tabla 2:

Vin [V]	R
1	14
2	14
3	14
4	12
5	13
6	14
Promedio	13.5

Tabla 1 Valores de la constante de retardo para cada valor de voltaje de entrada

De esta manera completamos el modelo de simulación como se muestra en la Figura 7:

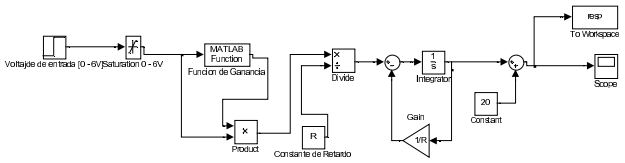

Figura 7 Diagrama de simulación del modelo matemático con la función de la ganancia y la constate de retardo identificada.

Este modelo se usó en este proyecto para la simulación de los controladores desarrollados como se puede apreciar en los siguientes capítulos.

VI. IMPLEMENTACIÓN DEL CONTROLADOR DIFUSO

En este capítulo se describe el diseño de un controlador difuso en Matlab, el objetivo es estabilizar la temperatura del sistema a los valores deseados utilizando descripciones lingüísticas en el rango de 20 a 70°C, este principio es mucho más sencillo que los métodos de control clásico.

El control difuso para el sistema de temperatura funciona en base a un voltaje de referencia, a partir de este voltaje el controlador lo aumenta o disminuye según la respuesta de la planta, es decir, el voltaje de referencia con la salida del controlador forman la señal de control (Figura 8).

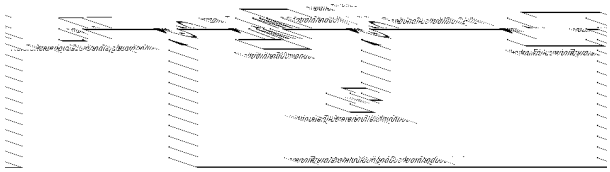

Figura 8. Diagrama del Controlador Difuso

En principio el voltaje de referencia que se usó fue el resultado de la siguiente relación lineal:

$$0 - 10\ [V]$$
$$20 - 70\ [°C]$$

De lo que resulta la ecuación:

$$Voltaje\ de\ referencia = \frac{(Temperatura - 20)}{5}\ [v] \quad (3)$$

Al usar esta relación, el voltaje de referencia resultante generaba una temperatura mucho mayor a la deseada, esto se debe a que la planta de temperatura describe un sistema no lineal, como se explicó en el Capitulo 3, por lo que se restó 2 voltios

a esta relación para disminuir la temperatura generada, así la relación usada nos queda:

$$Voltaje\ de\ referencia = \frac{(Temperatura - 20)}{5} - 2 \quad [v] \tag{4}$$

Con este valor se logra un control de lazo abierto adecuado.

Para diseñar el controlador difuso se utilizó el FIS Editor de Matlab (Figura 9).

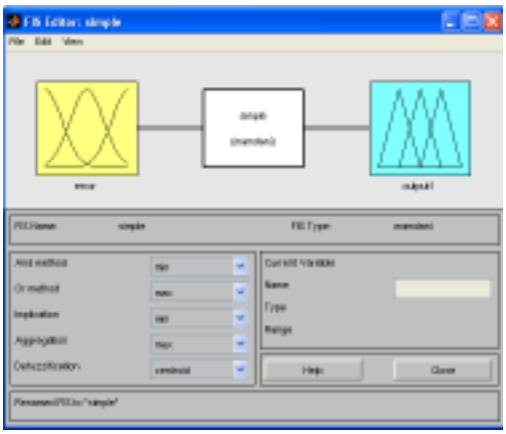

Figura 9. FIS Editor de Matlab

Rango de Operación de las Variables de Entrada y Salida:

En este controlador solo se tiene una variable de entrada que es el error, los límites de operación: -5 a 5 grados. La variable de salida que se sumará con el voltaje de referencia es de -5 a 5 Voltios.

Definición de las Funciones de Pertenencia Fuzzificación:

El controlador tiene dos funciones de pertenencia, el error y el voltaje de salida.

El error se define en cinco valores lingüísticos:

- Grande positivo (GP)
- Pequeño positivo (PP)
- Cero (Z)
- Pequeño negativo (PN)
- Grande negativo (GN)

Cuando la temperatura medida es igual a la deseada, el error tendrá un valor de 0, correspondiente a la variable lingüística Cero.

Cuando la temperatura medida supere a la deseada, el error será negativo, y podría pertenecer a las variables lingüísticas: Cero, Pequeño negativo o Grande negativo según su valor.

Si la temperatura medida es menor a la deseada, el error será positivo, y podría pertenecer a las variables lingüísticas: Cero, Pequeño positivo o Grande positivo según su valor.

En la Figura 10 se puede apreciar la función de pertenencia del error.

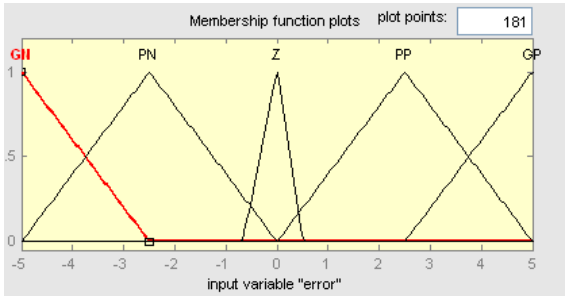

Figura 10. Función de pertenencia del error

La variable de salida del controlador se define en cinco valores lingüísticos:

- Calentar mucho (CM)
- Calentar poco (CP)
- Cero (Z)
- Enfriar poco (EP)
- Enfriar mucho (EM)

La variable de salida del controlador modificará el voltaje de referencia para controlar la temperatura del sistema.

En la Figura 11 se puede apreciar la función de pertenencia del la salida del controlador.

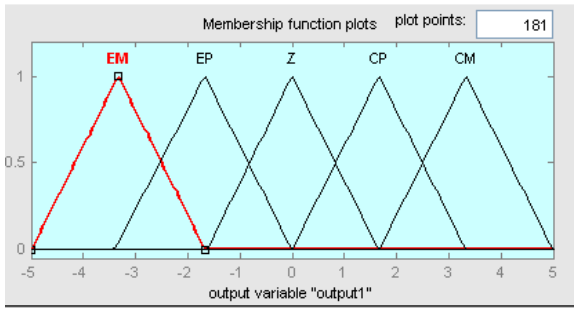

Figura 11 Función de pertenencia del la salida del controlador.

Desarrollo de la base de reglas

La base de reglas consta de cinco reglas donde se relacionan las premisas de las variables de entrada con las premisas de la variable de salida, como por ejemplo: *Si el error es grande positivo, la salida del controlador será Calentar mucho.*

En la tabla 3 se muestra la base de reglas

N°	Regla
1	Si (Error es grande positivo) entonces (Salida es Calentar mucho)
2	Si (Error es pequeño positivo) entonces (Salida es Calentar poco)
3	Si (Error cero) entonces (Salida es cero)
4	Si (Error es pequeño negativo) entonces (Salida es Enfriar poco)
5	Si (Error es Grande Negativo) entonces (Salida es Enfriar mucho)

Tabla 3 Base de reglas del controlador difuso

Para la implementación del controlador difuso se utiliza el modulo real de la planta de temperatura, esta es la única diferencia con el controlador simulado, en vez de utilizar el modelo matemático se utiliza la entrada analógica de la tarjeta de adquisición para obtener la temperatura del sistema y la salida analógica para enviar la señal de control, la temperatura del sistema se recibe en valores de 0 a 5 voltios por lo que se realiza su respectiva transformación a grados centígrados. Este diagrama se lo puede observar en la Figura 12.

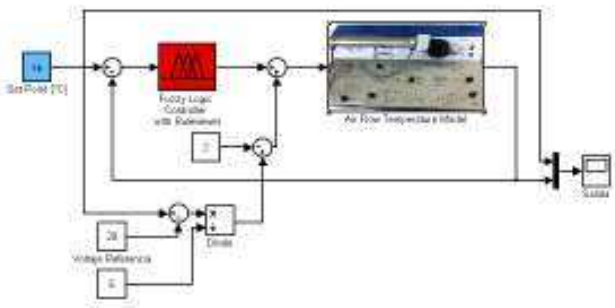

Figura 12. Diagrama del Controlador Difuso Implementado

La figura 13 nos muestra el desempeño del controlador difuso para tres diferentes valores de referencia en 350 segundos.

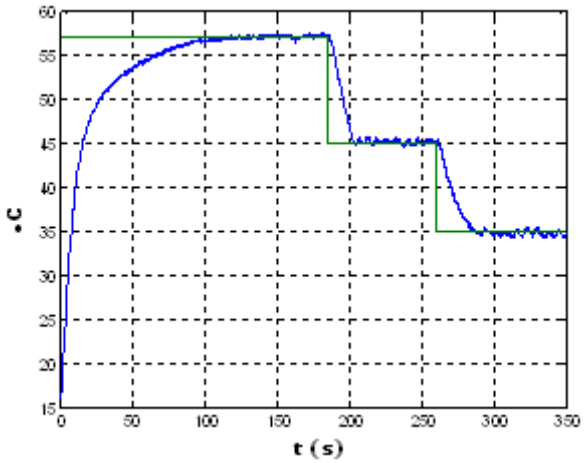

Figura 13. Desempeño del controlador difuso con la planta de temperatura real para diferentes valores de referencia

VII. IMPLEMENTACIÓN DEL CONTROLADOR PREDICTIVO

El objetivo de este capítulo es detallar el procedimiento de diseño de un controlador predictivo en Matlab para la unidad *air flow temperature control system*. Con esta técnica de control se pretende optimizar las señales de control que lleven al proceso a los objetivos deseados, mediante el concepto predicción del comportamiento de la planta.

El control predictivo funciona en base a la función de transferencia del proceso, por lo que se debe linealizar el sistema a un punto de operación. Para obtener la función de transferencia de la planta de temperatura se utilizó la herramienta Ident de Matlab con los valores de respuesta guardados del modelo matemático para un voltaje de entrada de 5 V con un tiempo de muestreo de 0.1 segundos, el diagrama para la obtención de estos datos se puede ver en la Figura 14.

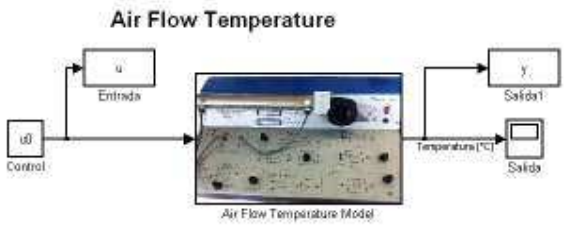

Figura 14. Diagrama para registrar la respuesta del modelo matemático a 5 V.

Para importar los datos obtenidos a la herramienta Ident realizamos los siguientes pasos (Figura 6.2):

En la ventana de comandos de Matlab digitamos:

data=iddata(y,u,Ts)

Donde *y* es la salida del modelo, *u* la entrada y *Ts* el tiempo de muestreo en segundos.

En la herramienta Ident en la Pestaña "Import data" seleccionamos: "Time domain data"

Posteriormente se abre la ventana "Import Data" y en la opción "Object" escribimos el nombre de la variable que en nuestro caso es "data".

Damos click en el botón Import y luego Close.

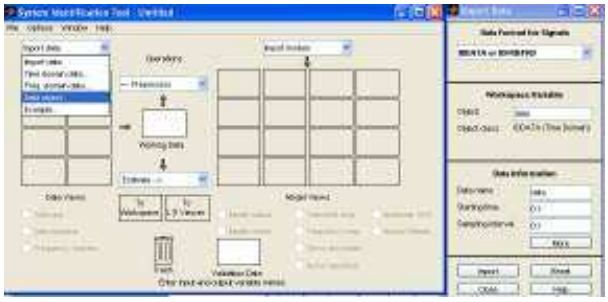

Figura 15. Carga Del Datos En La Herramienta Ident

Con los datos importados se utilizó la opción de estimación: "Process Model", exportamos la función de transferencia identificada al espacio de trabajo arrastrando al cuadro "To Worksapce".

La función transferencia identificada es:

$$Gs = \frac{10.614}{1 + 13.5\,s}$$

Para diseñar el controlador en la herramienta MPCTOOL es necesario realizar los siguientes pasos:

Importar la función de transferencia de la planta de temperatura

Para importar función de transferencia anteriormente identificada digitamos en la ventana de comandos de Matlab:

Gs= tf([10.614],[1 13.5])

Ingresar datos de la planta de temperatura

En la tabla 4 se muestra las características de la planta que se deben ingresar:

Nombre	Tipo	Descripción	Unidad	Valor Nominal
Voltaje	Variable Manipulada	Entrada	V	0
Temperatura	Variable Medida	Salida	°C	0

Tabla 0 Datos de la entrada y salida de la planta de temperatura

Diseñar el control predictivo

Los parámetros del control predictivo que se deben ingresar en la herramienta MPCTOOL se describen en la tabla 5.

Parámetro	Detalle	Valor
Modelo y Horizonte	Intervalo de Control	0.1
	Horizonte de Predicción	10
	Horizonte de Control	2
Restricciones	Entrada : Voltaje	Min: 0 Max: 6
	Salida: Temperatura	Min: 20 Max: 70
Pesos	Overall	0.8
	Varianza del peso	0.1
	Temperatura	1

Tabla 5 Datos de los parámetros del controlador Predictivo

Para la implementación del controlador predictivo se utiliza el módulo real de la planta de temperatura, esta es la única diferencia con el controlador simulado, en vez de utilizar el modelo matemático se utiliza la entrada analógica de la tarjeta de adquisición para obtener la temperatura del sistema y la salida analógica para enviar la señal de control, la temperatura del sistema se recibe en valores de 0 a 5 voltios por lo que se realiza su respectiva transformación a grados centígrados. Este diagrama se lo puede observar en la Figura 16.

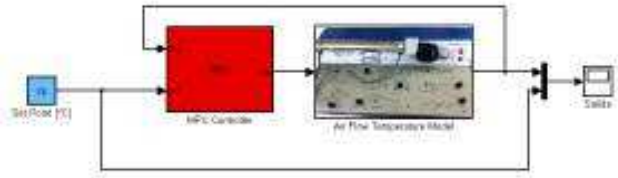

Figura 16. Diagrama del controlador predictivo implementado

En la figura 17 podemos ver el desempeño del controlador predictivo implementado en la planta real para diferentes valores de referencia en 350 segundos.

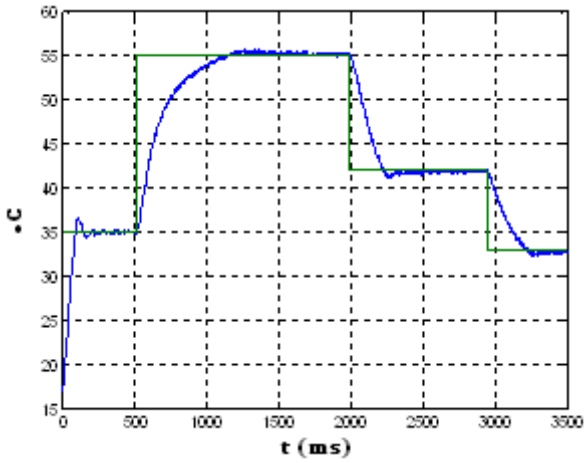

Figura 17 Desempeño del controlador predictivo en la planta real de temperatura para diferentes valores de referencia

VIII. IMPLEMENTACIÓN DEL CONTROLADOR CON REDES NEURONALES POR MODELO INVERSO

El objetivo es generar un sistema de control mediante una red neuronal inversa. Para generar el controlador, se debe entender que dado un estado inicial *To* (Temperatura inicial) se genera una señal de control *u* para con esto obtener las variaciones de estado deseadas.

El controlador de red neuronal inverso actúa en forma muy similar a la planta pero en forma inversa, es por esto que en el controlador tenemos como entrada la variación de estado y como salida la señal de control.

Lo que se desea es que la entrada *To*, que es el estado o temperatura inicial coincida después del procesamiento con la temperatura de referencia, es decir que la variación de estado deseada será:

$$\Delta t = t_{ref} - t_0 \quad (5)$$

Donde:

Δt	Variación de estado
t_{ref}	Temperetura de referencia
t_0	Estado inicial

Obtención de los patrones de entrenamiento

Previo a realizar la red de control se debe identificar la planta, para esto se debe utilizar el modelo matemático en lazo abierto para obtener los patrones de entrenamiento.

La red neuronal de control tiene como objetivo mantener la temperatura, manteniendo la variable de estado en el valor deseado, para esto se debe describir posibles valores que puede tomar tanto el estado como el valor de referencia:

```
%RANGOS DE LAS VARIABLES DEL SISTEMA
entrada=0:0.2:6;    %Temperatura de referencia
estado=20:1:60;     %Estado incial
```

Lo siguiente será crear una variable Pm que tendrá todas las posibles combinaciones entre el valor de referencia y el estado para así simular en el sistema dinámico con una variación de tiempo de simulación (tiempo de paso) igual a 2.

```
Pm = combvec(estado,entrada);
timestep = 2;
```

A continuación se debe crear la variable Tm, este vector contiene la variación que se produce en la variable de estado ante las condiciones de entrada de los estados actuales.

```
Q = length(Pm);
Tm = [];
%Identificacion
for i=1:Q
```

```
    entrada = Pm(2,i);
    to = Pm(1,i)-20;
    to1 = Pm(1,i);
    sim('subingpro',[0 timestep]);   %modelo no lineal
    dt = tf-to1;
    Tm=[Tm [dt]];
end
```

Entrenamiento de la red neuronal

Los patrones de entrenamiento para la red de control deben tener en consideración que el controlador debe funcionar en forma inversa a la planta. En el modelo matemático de la planta se tiene como entradas el estado inicial y el voltaje de entrada o señal de control, mediante estos valores de entrada obtenemos una variable de salida que es temperatura o una variación de temperatura.

Para empezar el aprendizaje, se debe volver a ordenar los vectores Pm y Tm, así el vector Pm deberá contener las variables que se desea que sean entradas y el vector Tm contendrá las variables que se desea que sean salidas en el controlador, y se las describe mediante código de la siguiente manera:

```
Targ=Pm(2,:);
Pm=Pm(1,:);
Pm=[Pm;Tm];
Tm=Targ;
```

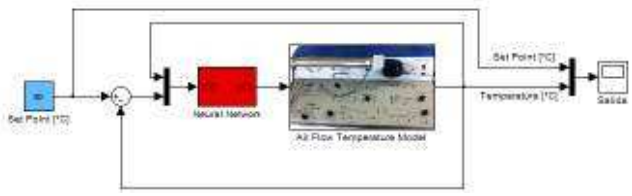

Figura 18 Diagrama para la implementación del control neuronal inverso en la planta

Se realizó varias pruebas usando Simulink con nuestro modelo de implementación como se muestra en la figura, cada una de ellas con diferentes temperaturas de referencia, pero además con un diferente tiempo de paso, con valores de tiempo de paso de 1, 2, 5, 10 y 20 segundos. Con estos resultados se observó cuál fue la mejor identificación de nuestro sistema y si nuestro controlador inverso fue el más adecuado, y en este caso el tiempo de paso más adecuado fue de dos segundos (Ver Figura 19).

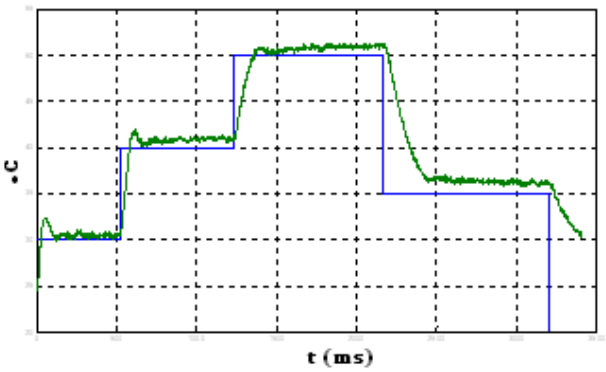

Figura 19 Prueba de la implementación del controlador neuronal inverso con un tiempo de paso de dos segundos.

IX. IMPLEMENTACIÓN DEL CONTROLADOR CON REDES NEURONALES POR MODELO DE REFERENCIA CON ACCION INTEGRAL

El objetivo es generar un control por medio de redes neuronales usando como referencia un modelo propio.

El control neuronal por modelo de referencia consiste en un sistema que controla el comportamiento natural de la planta con el objetivo de que se aproxime a la respuesta que tiene un modelo de referencia establecido.

Para encontrar el controlador, antes se debe identificar la planta, como lo antes realizado en el control neuronal inverso, así esta identificación formará parte de la red de control total.

En el entrenamiento del control neuronal los pesos y las bias se ajustarán gracias al error que se produce entre el modelo de referencia lineal y el sistema de control representado por la red neuronal de control e identificación. De este proceso se extrae la red neuronal de control, que se encargara de controlar la planta de temperatura.

La red se compone por cuatro capas, las dos primeras capas para el control (“cnet”) y las dos últimas capas para la identificación (“mnet”) como se observa en la Figura 20.

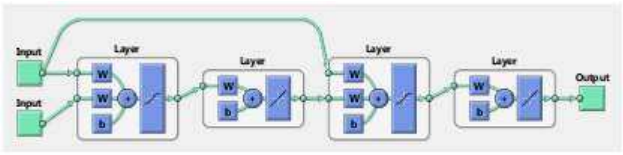

Figura 20. Filosofía del diseño de la red neuronal por modelo de referencia

La red de control viene junto a una acción integral que decrementa el tiempo de subida y tiene el efecto de eliminar el error de estado estable; es una acción totalmente necesaria para nuestra red para poder tener un control completo, aunque se tienen desventajas ya que también se incrementa el sobre impulso y puede empeorar la respuesta transiente.

Es decir se obtendrá una mayor desviación del set point, la respuesta será más lenta y el periodo de oscilación será mayor, pero tendremos un error bastante bajo.

Obtención de la red neuronal identificación

En este paso se realiza la identificación de la planta de temperatura en lazo abierto.

La red de identificación debe seguir la misma estructura de la red de control, por esto se tiene como entrada el voltaje *V1* y su estado inicial *to*. Esta red nos entregara como salida la variación de temperatura *dt.*

```
entrada=0:0.2:6;
estado=20:1:60;
Pm = combvec(estado,entrada);
```

Los patrones de entrenamiento están conformados por la matriz de patrones *Pm* y la matriz objetivo *Tm*, que representan la entrada y la salida de la red neuronal respectivamente.

```
timestep = 1;
Q = length(Pm);
Tm = [];
for i=1:Q
    entrada = Pm(2,i);
    to = Pm(1,i)-20;
    to1 = Pm(1,i);
    sim('subingpro',[0 timestep]);
    dt = tf-to1;
    Tm=[Tm [dt]];
end
```

Gracias a las múltiples variaciones de los valores patrones, se pudo obtener la red neuronal de identificación, donde a continuación se muestra la comparación con el modelo matemático (Figura 21), cabe recalcar que los dos sistemas estarán con iguales condiciones de estado inicial y entradas.

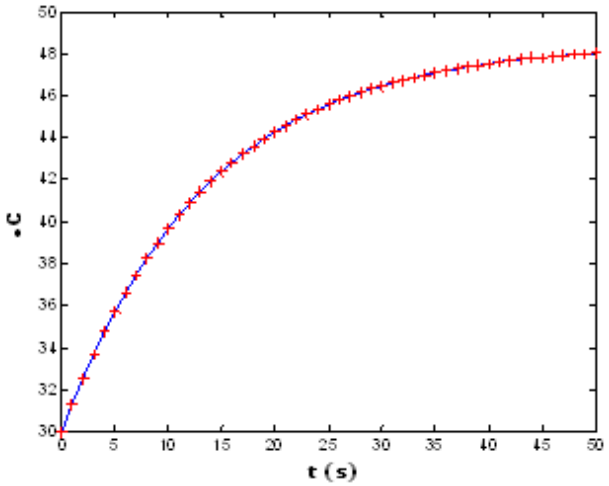

Figura 21 Comparación de la red neuronal identificación con el modelo matemático propuesto.

Obtención de los patrones de entrenamiento con el modelo de referencia lineal

El modelo de referencia lineal escogido es definido por la siguiente ecuación:

$$T = \frac{1}{1 + \tau s}$$

Que imita un comportamiento similar a un sistema de primer orden como el de la planta, donde podemos controlar su tiempo de establecimiento.

Se definió una constante de retardo de tiempo en este modelo de referencia de *Tau* igual a 10 segundos; después de varias pruebas con diferentes valores, se determino este valor.

El modelo de referencia ayuda a obtener los parámetros *Pc* y *Tc* para la red total *"tnet"*

```
demanda=25:1:40;
estado=25:1:40;
Pc=combvec(estado,demanda);
timestep = 1;
tao=10;
Q = length(Pc);
Tc = [];      %matriz de objetivos "Tc".
for i=1:Q
      demanda = Pc(2,i);
      to = Pc(1,i);
      sim('mmref',[0 timestep]);
      dt = tf-to;
      Tc=[Tc [dt]];
```

Obtención de la red neuronal de control

El control neuronal por modelo de referencia para el sistema de temperatura es una red tansig, purelin, que tiene nueve neuronas en la capa oculta y una salida. El algoritmo de entrenamiento es de tipo trainlm.

```
S1 = 9;

cnet = newff(minmax(Pc),[S1 1],{'tansig' 'purelin'});
```

Entrenamiento de la red neuronal

La red total que se obtiene al final está compuesta por la red del modelo y la red de control.

Una vez conformada la red total "tnet" con sus respectivos pesos y bias se procede al entrenamiento de la red total.

```
[tnet,tr] = train(tnet,Pc,Tc);
```

Finalmente se extrae la red de control "cnet" de la red total, es decir actualizando los pesos y bias entrenados.

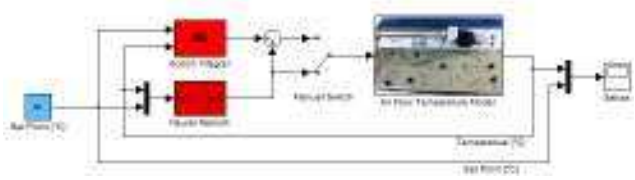

Figura 22 Modelo en Simulink para la implementación del control neuronal por modelo en referencia sin acción integral en la planta

Para la implementación del control se realizo pruebas con y sin la acción integral, ya que después de obtener el controlador lo último es ajustar y sincronizar la acción integral.

Se realizó la implementación de la red neuronal por modelo de referencia sin acción integral usando el modelo en Simulink del software como muestra la Figura 22, para esto se realizó pruebas con diferentes entradas de temperatura 35°C, 40°C y 50°C, a continuación la Figura 23 respuesta del controlador sin acción integral.

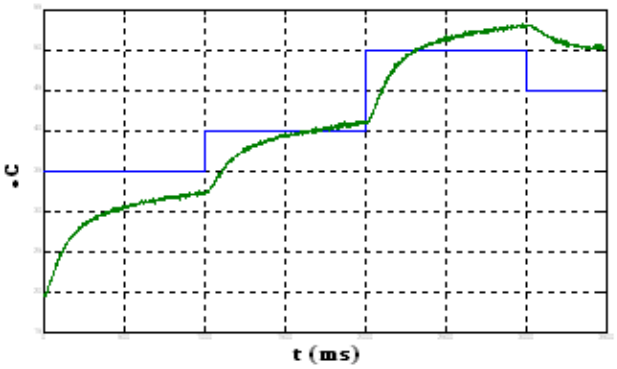

Figura 23 Respuesta del controlador neuronal con modelo de referencia sin acción integral

Como se observa es totalmente necesaria una corrección, ya que se tiene un error grande en este control y esto se compensa con la acción integral.

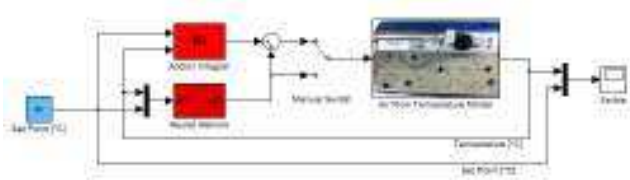

Figura 24 Modelo en Simulink para la implementación del control neuronal por modelo en referencia con acción integral en la planta

La siguiente prueba se realiza con el control neuronal por modelo de referencia y con la acción integral usando el modelo en Simulink como se muestra en la Figura 24, para empezar, se necesita sincronizar la acción integral.

Se realizo cuatro pruebas con diferentes valores de la ganancia integral, en la siguiente Figura, se puede observar la respuesta del sistema.

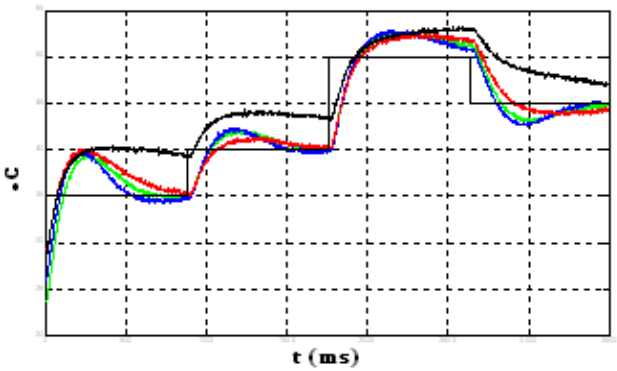

Figura 25 Respuestas de la implementación con diferentes ganancias de la acción integral.

Color	Ganancia Ki
Verde	0.008
Azul	0.01
Rojo	0.005
Negro	0.001

Tabla 6 Diferentes valores en la ganacia Ki para la sintonización de la acción integral en el controlador neuronal por modelo de referencia.

Al observar los resultados del comportamiento de la planta con la acción integral en la Tabla 6 y en la Figura 25, se puede concluir que el valor más adecuado para la acción integral es 0.008.

X. CONCLUSIONES

Para la implementación de los controladores neuronales y predictivo es necesario un modelo matemático que describa la dinámica del sistema real. En el caso de las redes neuronales se utiliza este modelo en la etapa de entrenamiento de los controladores y en el caso del control predictivo se requiere del modelo para optimizar los valores de la señal de control que ajuste la salida del proceso a la referencia ingresada.

Para el controlador difuso el modelo matemático solo se utiliza en la simulación, una ventaja de este control difuso es que simplemente con el conocimiento del funcionamiento de la planta se puede implementar un controlador.

En el control por redes neuronales se utilizó dos métodos de entrenamiento: modelo de referencia y red neuronal inversa, la diferencia de estos controladores principalmente se nota en los diferentes procedimientos de entrenamiento y desempeño en tiempo real. El controlador neuronal más eficiente fue la red neuronal inversa.

Para el control neuronal por modelo de referencia se logro el objetivo de controlar el sistema con la ayuda de una acción integral K_i, la planta contiene demasiados parámetros que no pueden incorporarse en el modelo matemático, es por esto que la red neuronal no logró eliminiar el error de estado al sistema, este problema se compensó con el efecto de una acción integral.

REFERENCIAS

[1] Bonifacio Martin del Brio, Alfredo Sanz Molina. *Redes Neuronales y Sistemas Difusos*, 2da Edicion, Alfaomega Ra-Ma

[2] Burns Roland S., *Advanced Control Engineering,* Butterworth Heinemann

[3] Rodriguez Ramírez Daniel, Bordóns Alba Carlos, *Apuntes de Ingeniería de Control.*

[4] Elena López Guillen Ma., *Identificación de Sistemas*

[5] *Yoshikuni Okawa, Introduction to Embedded Programming Using MATLAB, Simulink.*

[6] Ochoa José Bernardo, *Control Clásico y Control Inteligente.*

[7] MATLAB & Simulink Based Books.

[8] Ibarra Alexander, Acuña Alejandro, Diseño e Implementacion de un Sistema Controlador de Temperatura PID para la unidad Air Flow Temperature Control System mediante la utilización de la herramienta RTW (Real Time Workshop) de Matlab, 2010.

[10] Yunus A., Cengel, *Transferencia de Calor,* Segunda edición, Mc Graw Hill, 2004.

[11] Ati Patricia, *Análisis, Diseño e Implementación de Controladores Predictivos,* 2011.

[12] Armijos Luis, Chicaiza Dario, *Desarrollo e Implementación de Controladores Inteligentes en la Planta Multi-Tanque del Laboratorio de Servomecanismos.*

[13] Incropera Frank, DeWitt David, *Fundamentos de Transferencia de Calor,* 1996

[14] Ortiz Paula, Arias Alexander, *Control de una Planta No Lineal de Temperatura con Redes Neuronales,* 2009.

BIOGRAFÍA DEL AUTOR

Daniel Alejandro Capelo Parreño

Nació el 7 de enero de 1987 en la ciudad de Quito, provincia de Pichincha-Ecuador.

Tiene la edad de 25 años, realizo sus estudios primarios en la Colegio Guadalupano, obtuvo su título de Bachiller en el Colegio William Shakespeare School y su especialidad es en Ciencias.

Obtuvo su título de ingeniería en Electrónica Automatización y Control en él año 2012.

Uno de sus campos de interés destaca en el control de procesos industrial, y técnicas de control inteligente.

Adrian Paul Cabezas Yanez

Nació el 19 de diciembre de 1987 en la ciudad de Quito, provincia de Pichincha-Ecuador.

En el año 2005 concluye el bachillerato en el Colegio San Luis Gonzaga y su especialidad es de Físico Matemático.

Obtuvo su título de ingeniería en Electrónica Automatización y Control él año 2012.