

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-27

CONTROL DIFUSO DE MONÓXIDO DE CARBONO Y TEMPERATURA EN MINERÍA SUBTERRÁNEA DE CARBÓN

Diego Fernando Usquiano Sánchez

Ingeniería Mecatrónica

Juliana Valencia Aguirre

INSTITUTO TECNOLÓGICO METROPOLITANO

2016

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RESUMEN

La regulación y monitoreo de gases tóxicos y altas temperaturas en las minas subterráneas de carbón del país no es muy común, medir y controlar niveles de gas y temperatura al interior de la mina es necesario para garantizar atmosferas sanas y seguras.

En este trabajo se propone el diseño de un sistema de control automático basado en lógica difusa, que garantice los niveles admisibles de temperatura y concentración de monóxido de carbono (CO) en minería subterránea de carbón a través de la variación de la velocidad de los motores asociados al sistema de ventilación. El proceso comienza con la medición de temperatura y CO dentro de la mina, estas señales son enviadas a un Arduino donde se acondicionan y se convierten en los valores de entrada del controlador difuso. El valor de salida del control se convierte en una señal que modifica de manera análoga los parámetros de un variador de velocidad que activa el motor asociado al sistema de ventilación.

Se consigue controlar la temperatura y el CO en minería subterránea de carbón y así garantizar una atmósfera más limpia y un entorno más seguro.

Palabras clave: Control difuso, minería subterránea de carbón, ventilación, temperatura, monóxido de carbono.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RECONOCIMIENTOS

Quiero expresar mis agradecimientos a Juliana Valencia mi asesora, por su dedicación y acertados consejos en la elaboración de este proyecto. A Norma Guarnizo por su apoyo y por ser una guía tanto en el crecimiento profesional como personal.

Agradezco a los laboratorios del ITM por su disponibilidad y a sus encargados por su eficiencia. A mis compañeros de carrera que aportaron ideas en el desarrollo de este proyecto.

Finalmente, a mi familia que con su continuo aliento y apoyo lograron mantenerme con la fuerza necesaria para culminar este pregrado.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

ACRÓNIMOS

CO Monóxido de Carbono

PPM Partículas Por Millón

NIOSH Instituto Nacional de Salud y Seguridad Ocupacional

ACGIH Higienistas Industriales de la Conferencia Americana Gubernamental

OSHA Administración Seguridad y Salud Ocupacional

TWA Media Ponderada en el Tiempo

TLV Valor Umbral Límite

VLP Valor Límite Permisible

IDLH Inmediato Peligro para la Salud o Vida

REL Límite de Exposición Máximo

MF Función de Membresía

TRIMF Función de Membresía Triangular

TRAPMF Función de Membresía Trapezoidal

RRG Grupo de Investigación Robótico

PWM Modulación por Ancho de Pulso

ICSP Programación es Circuito Serial

VDF Variador de Frecuencia

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

TABLA DE CONTENIDO

1. INTRODUCCIÓN	6
2. MARCO TEÓRICO	8
3. METODOLOGÍA.....	15
4. RESULTADOS Y DISCUSIÓN.....	27
5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO	38
REFERENCIAS	40
APÉNDICE.....	42

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

1. INTRODUCCIÓN

En minería subterránea, la ventilación es necesaria para obtener un entorno libre de gases perjudiciales y temperaturas confortables, en general para garantizar una atmósfera segura. En la mayoría procesos mineros de la actualidad colombiana es notable la ausencia de la implementación de sistemas automáticos y controlados, aún sigue primando el uso de técnicas manuales a lo largo del proceso productivo.

La implementación de un control difuso en el sistema de ventilación brindaría diferentes ventajas como las siguientes:

- La regulación del consumo energético de los motores ya que en la actualidad se usan controles *on/off*.
- El conocimiento en tiempo real de las condiciones del aire.
- La posibilidad de registrar dichos datos automáticamente a lo largo del tiempo, obteniendo así mayor confiabilidad debido a que el registro actual se hace manualmente con intervalos hasta de 8 horas.

Por otra parte, la calidad del aire al interior de la mina debe cumplir los requerimientos mínimos establecido por el ministerio de minas y energía en el código de minas, al conseguir dichos niveles se puede optar por la certificación del ministerio.

En este trabajo se propone controlar la temperatura y los niveles de CO por medio de lógica difusa automatizando el proceso de ventilación de una mina subterránea de carbón, esto con el fin de garantizar un entorno saludable y seguro para los operarios de las minas. Para ello se cumplirán los siguientes objetivos específicos:

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Diseñar un controlador usando lógica difusa para garantizar la temperatura adecuada y los niveles mínimos de monóxido de carbono, según lo establecido por Ministerio de Minas y Energía.
- Generar el proceso de comunicación entre un Arduino y un variador de velocidad y finalmente implementar y simular las condiciones de la mina, la respuesta de control y la forma de realizar la integración de equipos.

Este documento se encuentra organizado de la siguiente manera: En [2] se encuentra el marco teórico en donde se hallan todos los conceptos usados en la elaboración del proyecto, en [3] está la metodología en la cual se describe detalladamente y en orden todos los pasos que se siguieron para obtener los objetivos propuestos.

En el numeral [4] se dan los resultados con sus respectivas discusiones, en [5] se presentan las conclusiones, recomendaciones y el trabajo futuro y finalmente se encuentran las referencias y los apéndices correspondientes a los distintos códigos de programación usados en la implementación.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2. MARCO TEÓRICO

En la minería actual del país, los sistemas de control y monitoreo de variables como la temperatura y concentraciones de gas son deficientes. En la mayoría, el seguimiento de estos datos se hace de forma manual y los procesos de ventilación se encuentran totalmente aislados a las alteraciones que dichas variables puedan presentar. La poca regulación que se tiene de la calidad del aire al interior de la mina representa un problema tanto de seguridad como de salud para los operarios y demás personal que trabaja bajo tierra (Vargas, 2015).

Algunos de los avances que se han hecho en el país en materia de control, se enfocan exclusivamente en la concentración de gas. En (Carreño, 2010) se propone un sistema que se fundamenta en la utilización de sensores dedicados a la detección de gases nocivos y niveles de oxígeno, estos niveles son reportados al sistema desarrollado en la plataforma LabVIEW a través de una tarjeta de adquisición de datos. Dependiendo del nivel detectado se toma la decisión de activar la ventilación o alarma, según sea el caso. La herramienta funciona de modo automático o con personal vinculado. En el sistema se definen un umbral el cual activa o desactiva la ventilación y un rango que al ser superado genera una alarma, además posee la capacidad de almacenar datos de cada tipo de gas medido.

En otra investigación se plantea y desarrolla un aplicativo que permite definir atmósferas explosivas subterráneas utilizando los métodos de Coward y Bureau of Mines. Restrepo (2012) afirma que: *“En la actualidad, el 95% de las minas subterráneas de carbón en Colombia no cuenta con un sistema de detección de las atmosferas en tiempo real”* y por ello define basado en la medición de cuatro diferentes gases la peligrosidad del ambiente.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

En un panorama internacional se pueden destacar desarrollos como los que se muestran a continuación, empezando con la empresa chilena VDM INGENIERÍA Y CONSTRUCCIÓN LTDA que realiza una propuesta en automatización de sistemas de ventilación en minas subterráneas enfocándose en el monitoreo de las condiciones de la mina y en el control remoto de las operaciones de ventilación principal, se implementa un control *on/off* de ventiladores y el control de los estados de las puertas batientes de ventiladores (abiertas-cerradas). Esta operación se controlaría de manera remota desde una sala de control descentralizada (SCC) y las decisiones tomadas por el operario se hacen con base a los datos obtenidos en tiempo real por medio de la instalación de varios sensores en la mina (Betancur, 2011).

En Eynez una mina de carbón subterránea situada en Soma, Turquía, es utilizado un sistema denominado SENTURION-600 MODEL. Este sistema consta de dos secciones principales, un ordenador principal y otro auxiliar que sirve de soporte en caso de que el primero entre en conflicto. El sistema primario está diseñado para procesar de manera preliminar los datos y transmitirlos al sistema de manera comunitaria. Los datos de gases peligroso obtenidos en mina son enviados al ordenador principal y luego a otro ordenador multipropósito para sistematizarlos entre otras actividades. La evaluación de datos de gases peligrosos es comparada con los niveles teóricos peligrosos y allí se toman decisiones como la evacuación de personal si sobrepasa los valores de CO (Pamukcu, 2007).

Además de lo antes mencionado, indicando la importancia del manejo óptimo de los residuos gaseosos y de partículas de polvo producto de la actividad en una mina subterránea a través de mecanismos autónomos, también se presenta la importancia de disminuir el consumo de energía en el sistema de ventilación mediante implementación de un método que se conjuga perfectamente con las necesidades de automatización y monitoreo constante antes mencionados. El sistema de Ventilación Bajo Demanda propone trabajar mediante tres principales estaciones o pasos: entrada de datos, sistema de control

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

y dispositivos controlados. El sistema permite tomar una decisión inteligente con respecto a la demanda de flujo de aire por zonas, esto se logra tomando el control de los dispositivos disponibles tales como ventiladores, persianas y puertas para enviar aire directamente a la ubicación del objetivo (Allen & Tran, 2008).

Para el completo entendimiento del desarrollo de este proyecto es necesario conocer los conceptos en los que se fundamente, los principales son los siguientes:

Lógica Difusa:

Es una forma de lógica basada en conjuntos difusos, cada elemento del conjunto posee un grado de pertenencia que se expresa en un rango de valores que va desde 0 a 1 lo que representa un rango de "definitivamente no en el conjunto" hasta "parcialmente en el conjunto" a "totalmente en el conjunto". En un sentido más amplio la lógica difusa se identifica matemáticamente como un grupo de clases que no tiene límites claros o específicos. (Sangali, 2014).

Los conjuntos difusos son una generalización de los conjuntos ordinarios, y se pueden combinar mediante operaciones como la unión, intersección y complemento. Sin embargo, algunas propiedades de las operaciones de conjuntos ordinarios ya no son válidas para los conjuntos difusos.

Un conjunto difuso se define matemáticamente como:

$$A = \{x, \mu_A(x) | x \in U\}$$

Donde $\mu_A: x \rightarrow [0,1]$ es la función de pertenencia, $\mu_A(x)$ es el grado de pertenencia de la variable x y U es el dominio de la aplicación, también llamado el Universo del Discurso.

Una función de membresía que también se conoce como función de pertenencia es la que permite obtener el grado de pertenencia de un elemento al conjunto difuso. De estas

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

existen distintos tipos como la función triangular, trapezoidal, gaussiana, campana, S, Z, sigmoïdal. En particular, para el desarrollo de este proyecto fueron utilizadas las siguientes funciones de membresía:

Función Triangular:

La forma más común de detallar una función de este tipo, es a través de la identificación de los tres puntos del triángulo, $Tri[a, b, c](x)$ con $a \leq b \leq c$. En este caso la función de membresía vendrá dada por la expresión:

$$\mu_{Tri}[a, b, c](x) = \begin{cases} \frac{x - a}{b - a} & \text{si } a < x \leq b \\ \frac{c - x}{c - b} & \text{si } b < x < c \\ 0 & \text{en otro caso} \end{cases}$$

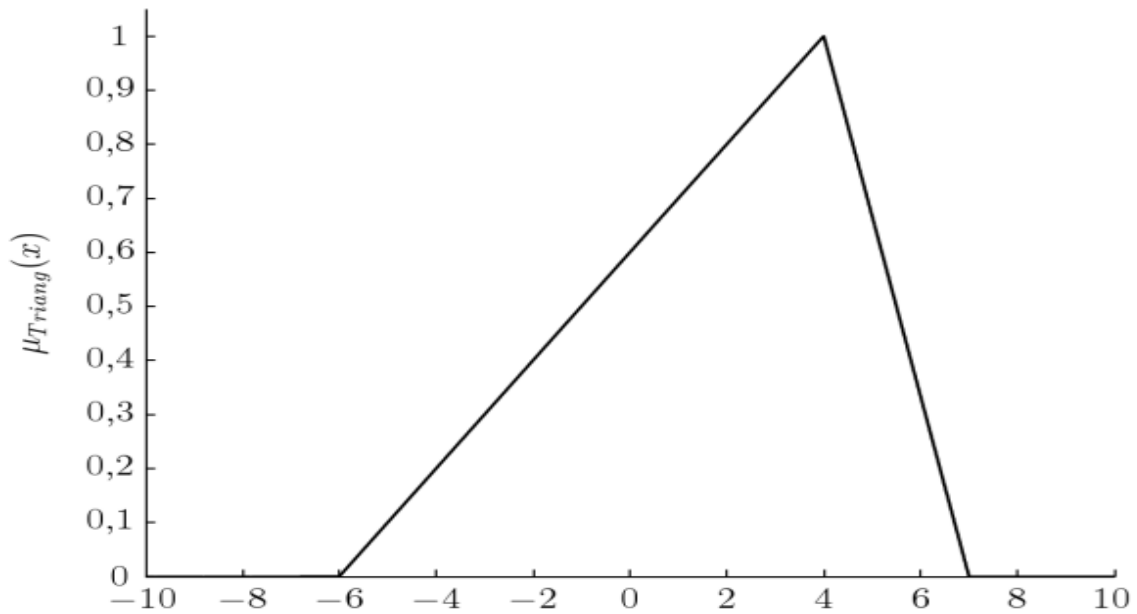


Figura 1. Función de membresía triangular. Tomada de Síntesis de Sistemas de Control Borrosos Estables por Diseño

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

En la figura 1 se muestra la función de membresía correspondiente a $Tri[-6, 4, 7](x)$ que tiene como dominio $X \in [-10, 10]$. El soporte de esta función en específico es de $(-6, 7)$ y el valor de su núcleo es igual a 4.

Función Trapezoidal:

Este tipo de función de membresía al igual que el anterior se puede definir por los puntos del trapecio así: $Trap[a, b, c, d](x)$ con $a \leq b \leq c \leq d$. De esta manera la función estará dado por:

$$\mu_{Trap}[a, b, c, d](x) = \begin{cases} \frac{x - a}{b - a} & \text{si } a < x < b \\ 1 & \text{si } b \leq x \leq c \\ \frac{d - x}{d - c} & \text{si } c < x < d \\ 0 & \text{en otro caso} \end{cases}$$

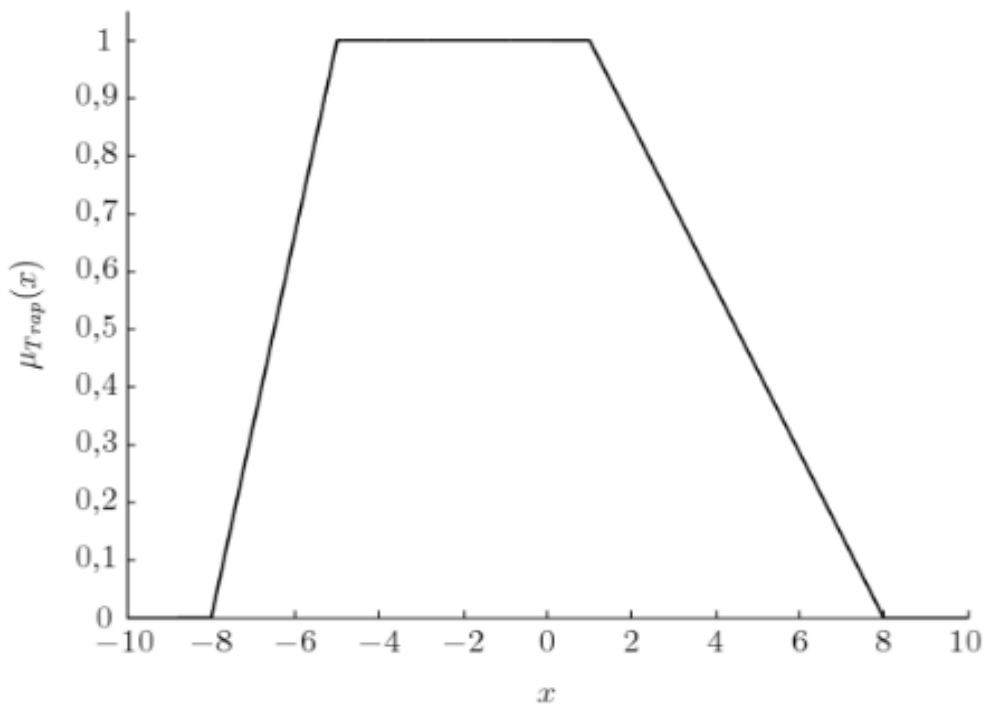


Figura 2. Función de Membresía Trapezoidal. Tomada de Síntesis de Sistemas de Control Borrosos Estables por Diseño

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

En la figura 2 se encuentra la función de membresía correspondiente a $Trap[-8, -5, 1, 8](x)$, con soporte en el intervalo $(-8, 8)$ y núcleo en $[-5, 1]$. (Barragán, 2009).

Arduino:

Arduino es una plataforma de programación de código abierto que permite leer entradas y transformarlas en la salida que sea deseada. Para este proyecto se usa como controlador principal, ya que en él se monta todo el control difuso desarrollado.

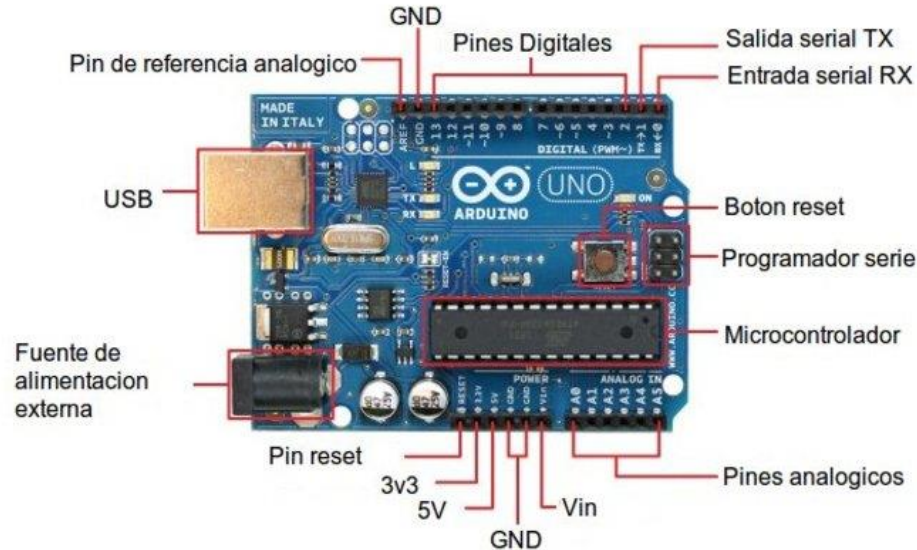


Figura 3. Arduino Uno y sus partes principales. Tomado de comohacer.eu/analisis-comparativo-placas-arduino-oficiales-compatibles

El Arduino Uno es una placa basada en el microcontrolador ATmega328P. Cuenta con 14 pines digitales de entrada/salida (de los cuales 6 se pueden usar como salidas PWM), 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP y un botón de reset.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Cuenta con varias entradas análogas por donde ingresan las señales correspondientes a las variables a controlar. Con ellas se dispone en el controlador principal el inicio del sistema de control difuso que con ayuda de una librería específica procede a entregar una salida análoga que regulará la velocidad de la ventilación.

Este sistema proporciona numerosas ventajas como lo son la facilidad de programación debido a su similitud con C++, su versatilidad en cuanto al gran número de librerías para acondicionamiento de sensores como de temperatura y gases, por ejemplo y finalmente se hace necesario mencionar su bajo costo.

Variador de Velocidad:

Un variador de velocidad usa el concepto del inversor de frecuencia transformando la corriente alterna con la que se alimenta a corriente directa, cuenta con rectificador de diodos, un contactor interno, resistencias y capacitores que en conjunto logran obtener una corriente directa más estable. Luego la corriente directa se transforma a alterna de nuevo con una frecuencia de 60 ciclos por segundo; dicha variación de frecuencia regulará la velocidad del motor a solicitud del variador.

Para conseguir la conversión de corriente directa a alterna, *internamente en el Inversor existen dos tarjetas electrónicas, una de control que tiene un procesador similar al de una computadora y que es el cerebro del aparato quien mandará a otra tarjeta electrónica llamada de disparo o de potencia que regulará la operación de un circuito de transistores de potencia IGBT quienes son los últimos encargados de formar la nueva corriente alterna de la salida hacia el motor mediante el sistema PWM, "Pulse Width Modulation" ("Modulación por ancho de pulso").* (Escobedo, 2011).

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3. METODOLOGÍA

Para la realización de proyecto se tienen principalmente tres etapas: El diseño del controlador difuso, la implementación del control y sus respectivas variables en el Arduino y finalmente la comunicación de este con el variador de velocidad.

Diseño del sistema de control difuso:

En la elaboración de este proyecto se implementa lógica difusa ya que posee la ventaja de que al trabajar con variables en términos lingüísticos no se hace necesario el desarrollo de complejos modelos matemáticos para definir el comportamiento de las condiciones del aire al interior de la mina de carbón, simplificando así el procesamiento de las señales.

Inicialmente las variables de entrada son los niveles de CO medidos en PPM y temperatura medida en °C. Para cada una de estas variables es necesario definir rangos que cuantifican los niveles en los que puede encontrarse en algún instante las entradas del sistema de control. Para definir dichos niveles se tuvo en cuenta los parámetros definidos por el Ministerio de Minas y Energía en el decreto número 1886 de 21 de septiembre de 2015 en el cual se establece el Reglamento de Seguridad en las Labores Mineras Subterráneas.

En el artículo 39 se establecen los valores límites permisibles para gases contaminantes, de allí y considerando lo establecido por el Instituto Nacional de Salud y Seguridad Ocupacional (NIOSH), Los Higienistas Industriales la Conferencia Americana Gubernamental (ACGIH) y la Administración Seguridad y Salud Ocupacional se establecen los siguientes niveles para el CO (Ministerio de Minas y Energía, 2015):

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Nivel de CO	Valor en PPM
Permisible	Menor a 15
Prevención	Entre 0 y 25
Límite Permisible	Entre 20 y 40
Peligro Inmediato	Mayor a 25

Tabla 1. Definición de niveles de CO difusos.

En el Capítulo III se definen las condiciones de temperatura, con esta información se definen los siguientes rangos:

Temperatura	Valor en °C
Baja	Menor a 20
Media	Entre 17 y 25
Óptima	Entre 21 y 29
Alta	Entre 24 y 32
Evacuar	Mayor a 29

Tabla 2. Definición de niveles de temperatura difusa.

De esta manera quedan definidos los valores lingüísticos de las dos variables de entrada, se definen los valores de la variable de salida que es la velocidad del motor asociado al sistema de ventilación, este dato de frecuencia en Hz es el que recibe el variador de velocidad. Conociendo que el rango que admite el variador de velocidad va desde 0 Hz hasta 60 Hz se genera la siguiente tabla:

Velocidad	Valor en Hz
Mínima	Menor a 15
Moderada	Entre 10 y 30
Media	Entre 25 y 45
Alta	Entre 38 y 48
Máxima	Mayor a 55

Tabla 3. Definición de rangos de Velocidad difusa.

Partiendo de esta información se procede a generar el sistema de control, para ello se usa el toolbox Fuzzy Logic del software Matlab, se define un control difuso con dos entradas y una salida de la siguiente forma:

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

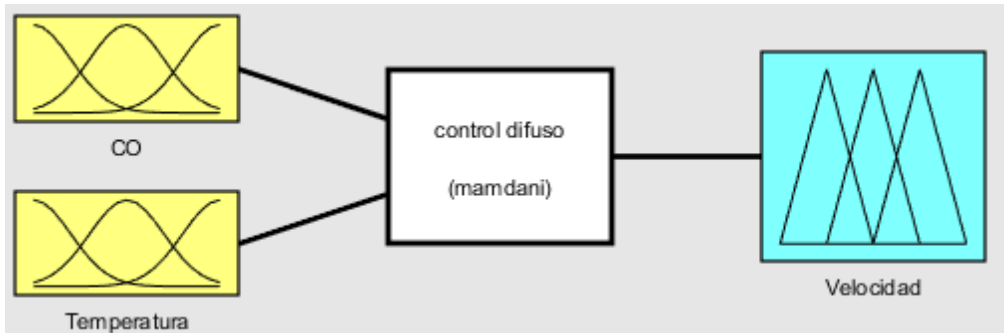


Figura 4. Esquema del controlador difuso.

Luego se define cada una de las entradas (CO y temperatura) y las salidas (velocidad) teniendo en cuenta los valores considerados en las tablas anteriores así:

Monóxido de Carbono CO:

Para el CO se escogen funciones de membresía trapezoidales (TRAPMF) para los niveles permisible y peligro inmediato que se encuentran en los extremos, esto con el fin de tener continuidad en el control en caso de presentarse valores que no se encuentren en el rango mostrado. Para los niveles de prevención y límite permisible se establecen funciones de membresía triangulares (TRIMF). Las geometrías se establecen de esta forma para que los grados de pertenencia que se asignan a un determinado valor de CO concuerden con su importancia real.

Los parámetros de cada función de membresía (MF) se definieron de la siguiente manera:

- Permisible [0 0 8 15]
- Prevención [0 15 25]
- Límite Permisible [20 25 40]
- Peligro Inmediato [25 50 200 1200]

En la última MF correspondiente a “Peligro Inmediato” se escoge el valor 200 ya que es un límite de exposición que nunca debe excederse sin importar 8 horas de TWA y el valor 1200 ha sido designado por NIOSH como de Inmediato para la Salud o Vida y así armar el trapecio

con valores que den sentido a la prolongación de la MF a límites que no debería llegar nunca.

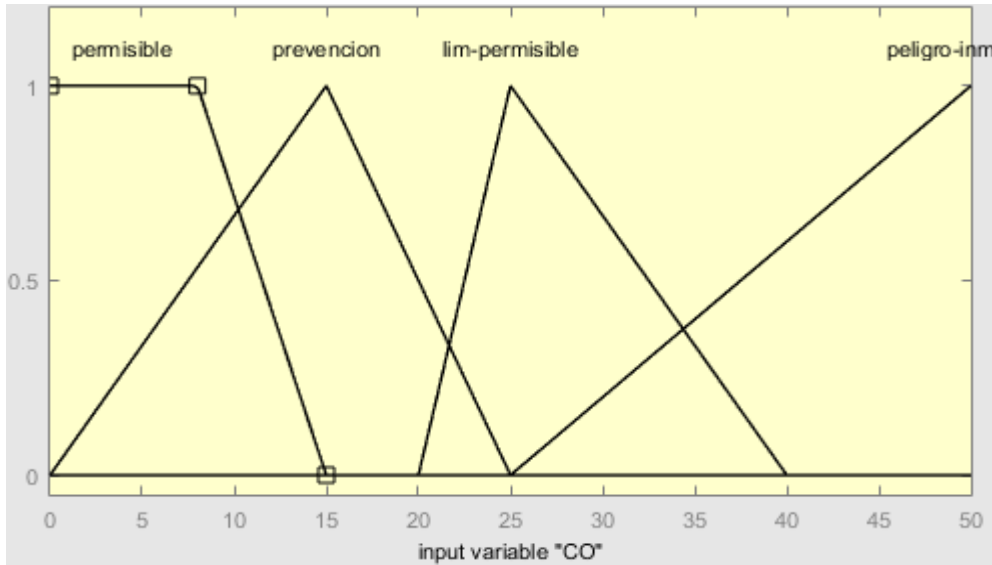


Figura 5. Funciones de Membresía para la variable de entrada CO

Temperatura:

Para la temperatura se usan TRAPMF para los niveles Baja y Evacuar con la misma finalidad que en el CO, los otros niveles usan TRIMF. Los parámetros de cada MF se muestran a continuación

- Baja [0 0 18 20]
- Media [17 21 25]
- Óptima [21 25 29]
- Alta [24 28 32]
- Evacuar [29 32 90 90]

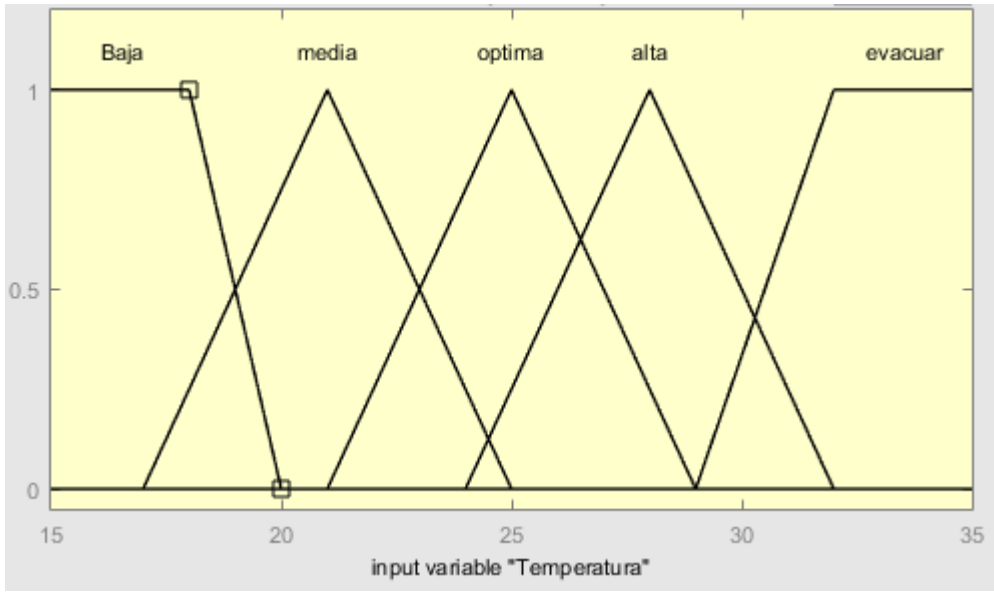


Figura 6. Funciones de Membresía para la variable de entrada Temperatura.

Velocidad:

Finalmente se crean las funciones de membresía para la variable de salida, en este caso Velocidad. Se hace la misma selección en cuanto al tipo de funciones de membresía que en las variables de entrada, esto debido a que la librería de lógica difusa que se emplea en el Arduino solo permite el uso de estos dos tipos de MF. La creación de otro tipo de MF aún no se encuentra definido para este dispositivo.

Aclarado esto, se generan los siguientes parámetros:

- Mínima [0 0 10 15]
- Moderada [10 20 30]
- Media [25 35 45]
- Alta [38 48 58]
- Máxima [55 60 60 60]

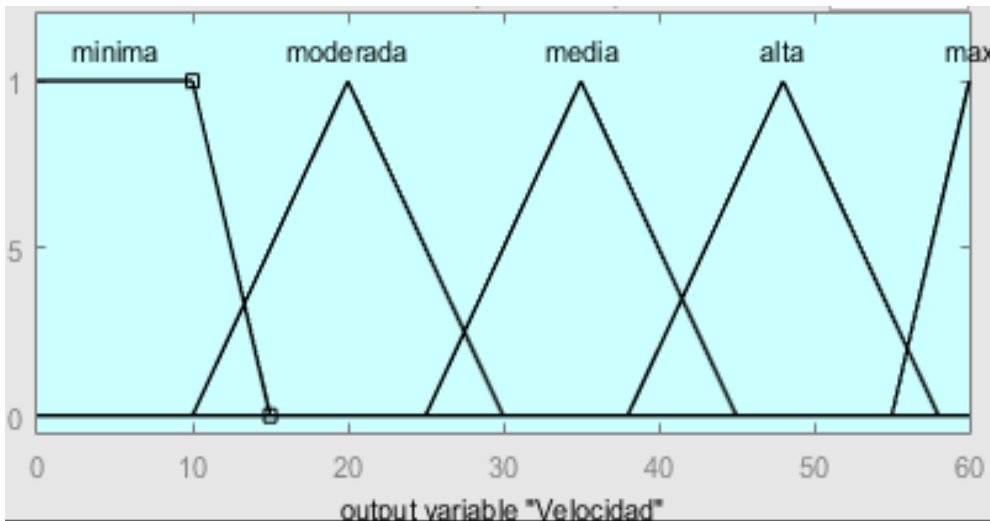


Figura 7. Funciones de Membresía para la variable de salida Velocidad.

Reglas:

Al haberse establecido todos estos parámetros, funciones de membresía y conjuntos difusos, se procede a generar las reglas. Con estas reglas se debe garantizar el cubrimiento total de cualquier situación que pudiese ocurrir, de esta manera el controlador encuentre una respuesta adecuada sin importar las condiciones de entrada.

La generación de estas reglas debe ser cuidadosa en el sentido que crear un número exagerado de reglas con el fin de cubrir n posibles estados podría incurrirse en redundancia y disminuir la eficiencia del algoritmo al momento de ejecutarse.

Las reglas generadas fueron las siguientes:

- Si (CO es Permisible) y (Temperatura es Baja) entonces (Velocidad es Mínima)
- Si (CO es Permisible) y (Temperatura es Media) entonces (Velocidad es Moderada)
- Si (CO es Prevención) y (Temperatura es Media) entonces (Velocidad es Moderada)
- Si (CO es Límite Permisible) y (Temperatura es Óptima) entonces (Velocidad es Alta)
- Si (CO es Peligro Inmediato) o (Temperatura es Evacuar) entonces (Velocidad es Máxima)
- Si (CO es Permisible) y (Temperatura es Óptima) entonces (Velocidad es Moderada)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Si (CO es Límite Permissible) y (Temperatura es Media) entonces (Velocidad es Alta)
- Si (CO es Límite Permissible) y (Temperatura es Baja) entonces (Velocidad es Media)
- Si (CO es Prevención) y (Temperatura es Óptima) entonces (Velocidad es Media)
- Si (CO es Permissible) y (Temperatura es Evacuar) entonces (Velocidad es Máxima)
- Si (CO es Permissible) y (Temperatura es Alta) entonces (Velocidad es Alta)
- Si (CO es Prevención) y (Temperatura es Alta) entonces (Velocidad es Alta)
- Si (CO es Prevención) y (Temperatura es Baja) entonces (Velocidad es Mínima)

Todas las reglas se definieron con el mismo peso, y en solo una se eligió el conector or.

Implementación en Arduino:

En primer lugar, es necesario el uso de la librería eFLL (Librería de Lógica Difusa Embebida) desarrollada por Grupo de Investigación Robótica de la Universidad Estatal de Piauí.

En la creación del código hay varias etapas importantes, la primera de ellas hace referencia a las señales de entrada que corresponden a los valores de CO y Temperatura, ambas señales son análogas que ingresan al Arduino por esta misma clase de puertos.

En seguida se debe crear el control difuso, para ello se sigue una estructura muy similar a la descrita anteriormente en el diseño del controlador. Primero se especifica cada una las entradas y la respectiva salida, en ellas se define cada conjunto difuso con sus respectivos valores. Cabe resaltar que esta librería solo admite dos distintos tipos de MF; TRIMF y TRAPMF.

La siguiente parte del código hará referencia a las reglas, 13 en total para este caso. Se sigue la misma estructura ya mostrada. En cada regla se evalúan dos parámetros ya definidos en las entradas difusas, y se asocian a una salida difusa.

Finalmente, la salida difusa se guarda en una variable de tipo *float*, esta señal se envía por salida de PWM donde el valor entre [0, 60] Hz se ha escalizado entre [0, 255] que equivale

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

a una señal en voltaje que varía entre 0v y 5v. Este valor es el que será interpretado por el variador de velocidad.

Se usó un Arduino UNO provisto por el laboratorio de Mecatrónica del ITM.

Comunicación con Variador de Velocidad:

La conexión con el variador de velocidad será análoga para ello se deben configurar los siguientes parámetros:

- P0700: Selección de fuente de órdenes. Se fija en 2, terminal
- P1000: Selección de frecuencia del setpoint. Se fija en 2, setpoint análogo

Adicionalmente, debido a que la entrada análoga del variador está establecida entre 0v a 10v y la salida del Arduino varía entre 0v y 5v se debe escalar los valores de entrada, para ello se modifican los siguientes parámetros:

- P0756: Tipo de ADC. Se fija en 0, entrada de voltaje unipolar
- P0757: Valor x1 de la escalización ADC. Se fija en 0, equivale a 0%
- P0758: Valor y1 de la escalización ADC. Se fija en 0, equivale a 0V

El P0757 y P0758 son correspondientes, es decir, el 0V equivaldrá a 0%. (O sea 0Hz por defecto).

- P0759: Valor x2 de la escalización ADC. Se fija en 100, equivale a 100%
- P0760: Valor y2 de la escalización ADC. Se fija en 5, equivale a 5V

El P0759 y P0760 son correspondientes, es decir, el 5V equivaldrá a 100%. (O sea 60Hz por defecto).

Luego se hace la conexión de las entradas análogas del variador de velocidad así:

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Número de terminal	Terminal del Variador	Conectado a:
2	+ entrada análoga o referencia análoga	Pin PWM del Arduino, donde se encuentra la salida de control.
3	- entrada análoga o tierra de la referencia	Tierra del Arduino
5	Controla el encendido del Variador	24V fuente externa
6	Cambio de giro del Variador	No se conecta
28	Masa del variador	Tierra de la fuente externa

Tabla 4. Conexión de las terminales del variador de velocidad.

Así la salida final de control estaría comunicada con el variador de velocidad y este finalmente a un motor que simula la ventilación de la mina subterránea de carbón.

Los equipos a los que se hace referencia son:

- Variador de velocidad Siemens Micromaster 440
- Motor Siemens de 0,5 HP a 60Hz
- Fuente de voltaje variable

Todos estos provistos por el laboratorio de Automatización Industrial del ITM.

Implementación del prototipo:

Después de haber diseñado el controlador difuso e implementarlo en Arduino y tener preparada la conexión con el variador de velocidad para realizar la comunicación, se hace la integración de todos los equipos para la puesta en marcha.

Para conseguir la completa funcionalidad de todos los elementos del proyecto, se sigue el esquema mostrado en la figura 8, donde se muestra de manera general y ordenada la disposición de cada uno de los elementos:

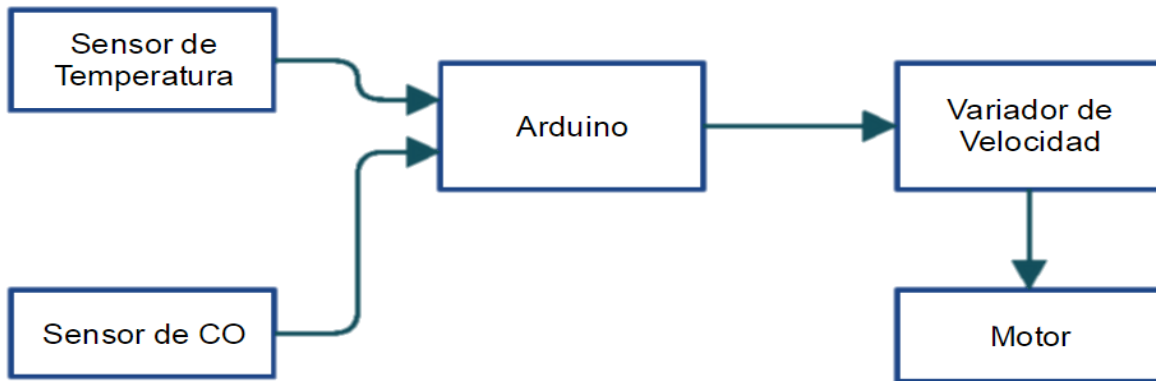


Figura 8. Diagrama de bloques de la implementación del prototipo.

Cada una de las partes del prototipo se explica a continuación:

Sensor de Temperatura:

Este sensor es el encargado de transformar en una señal eléctrica la temperatura del ambiente minero, se ha usado para esta finalidad un sensor BMP180 en oportunidades anteriores. El BMP180 mide temperatura por radiación, y sería ideal la implementación de sensores de este tipo ya que bajo tierra se requiere una medición en amplios volúmenes y no en sitios puntuales.

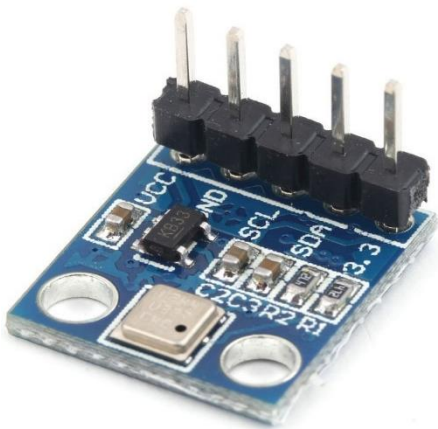


Figura 9. Sensor BMP180 con el módulo para conexión a Arduino. Tomado del blog <http://41j.com/blog/>

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Conocida la respuesta de este sensor y para efectos del prototipo se simuló usando un potenciómetro como divisor de tensión, enviando directamente esta señal al Arduino

Sensor de CO:

Este sensor se encarga de medir el CO en PPM y convertirlo en una señal eléctrica. Para dicha tarea en trabajos anteriores se ha usado el sensor MQ-9, dispositivo sensible al CO como gas inflamable.



Figura 10. Sensor de CO MQ-9 con el módulo para conexión a Arduino. Tomado de <https://altronics.cl/sensor-gas-combustible-mq-9>

La señal en voltaje que entrega este sensor fue simulada en la realización del montaje al igual que el anterior con un potenciómetro como divisor de tensión, el comportamiento ya había sido conocido y para efecto del prototipo se recrearon de igual forma.

Arduino y Variador de Velocidad:

El Arduino toma las señales que se definieron anteriormente y las ingresa al controlador difuso, obteniendo una salida que se envía al variador de velocidad. La implementación del controlador en Arduino y la parametrización del variador de velocidad se encuentra detallado cada uno en su respectiva sección.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Motor:

En esta parte se implementa un Motor Siemens de 0,5 HP a 60Hz, la función de este elemento en el prototipo es la de emular la entrada de movimiento que se asociaría al sistema de ventilación de la mina. Este sería el elemento final control del sistema y en él se evidencia los cambios en la frecuencia en respuesta a los valores definidos por el controlador difuso.

4. RESULTADOS Y DISCUSIÓN

Después de tener en consideración todos los aspectos mencionados en la metodología se obtiene un sistema de control difuso de CO y temperatura que está formado principalmente por el diseño del control, la implementación en Arduino y su conexión de variador de velocidad. Todos los resultados en conjunto con el análisis de sus ventajas o desventajas y la implementación final que muestra el correcto desarrollo del proyecto se enseña a continuación.

Primero se genera la siguiente superficie de control (Figura 11), en ella se puede apreciar el valor en Hz que se obtendría para las entradas de CO y temperatura, los rangos son [0, 50] y [15, 35] respectivamente, se muestran estos valores debido a que son los de mayor impacto en el control.

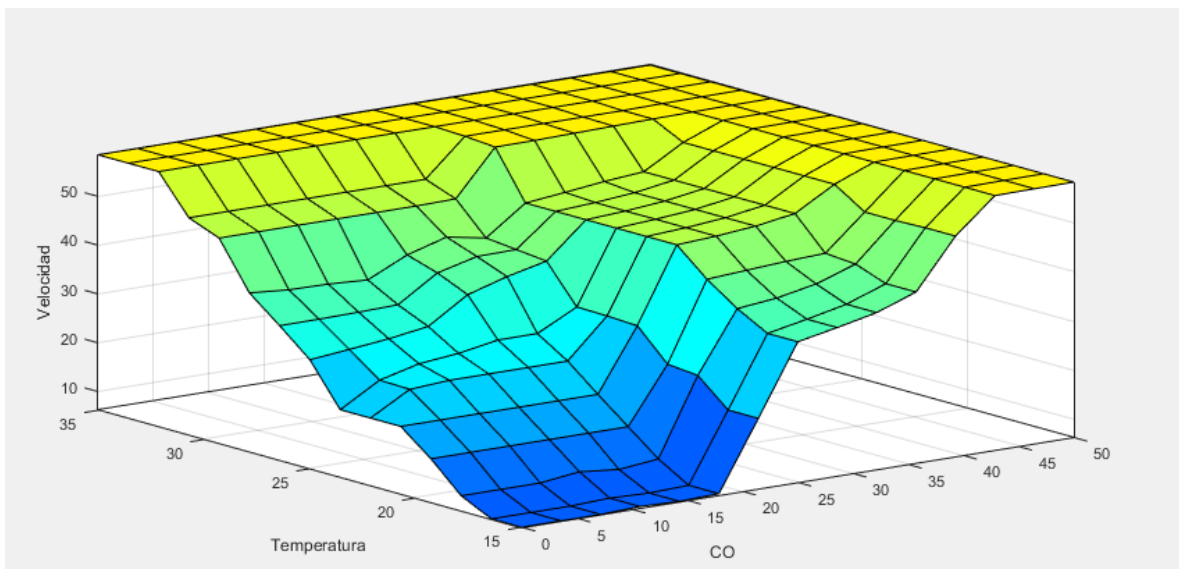


Figura 11. Superficie del controlador difuso de CO y Temperatura.

Esta superficie de control muestra el aumento en la velocidad de ventilación a medida que los valores de CO y Temperatura aumentan. Así, sin la necesidad de realizar cálculos

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

matemáticos complejos para modelar el comportamiento del CO y la Temperatura en una mina subterránea de carbón, se obtiene un controlador difuso que parte de la definición de variables de forma lingüística.

En la figura 12 se muestra la variación de la velocidad con respecto al CO mientras que la figura 13 es posible ver la variación de la velocidad con respecto a la temperatura. En estas gráficas se destaca que la curva del CO es más suave si es comparada con la de la temperatura, esto se debe a que en la realidad el CO es más peligroso. El CO es un gas inodoro, incoloro y altamente tóxico, su presencia en una mina no es de fácil detección por el personal que se pueda encontrar al interior de la mina, siendo más vulnerable a sus efectos.

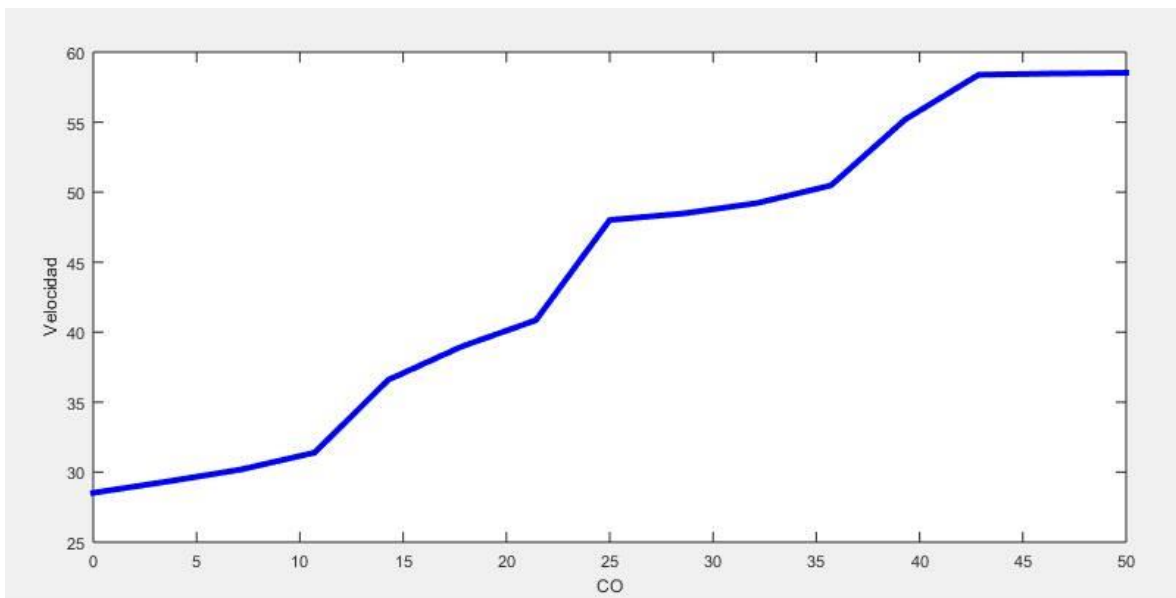


Figura 12. Velocidad con respecto al CO

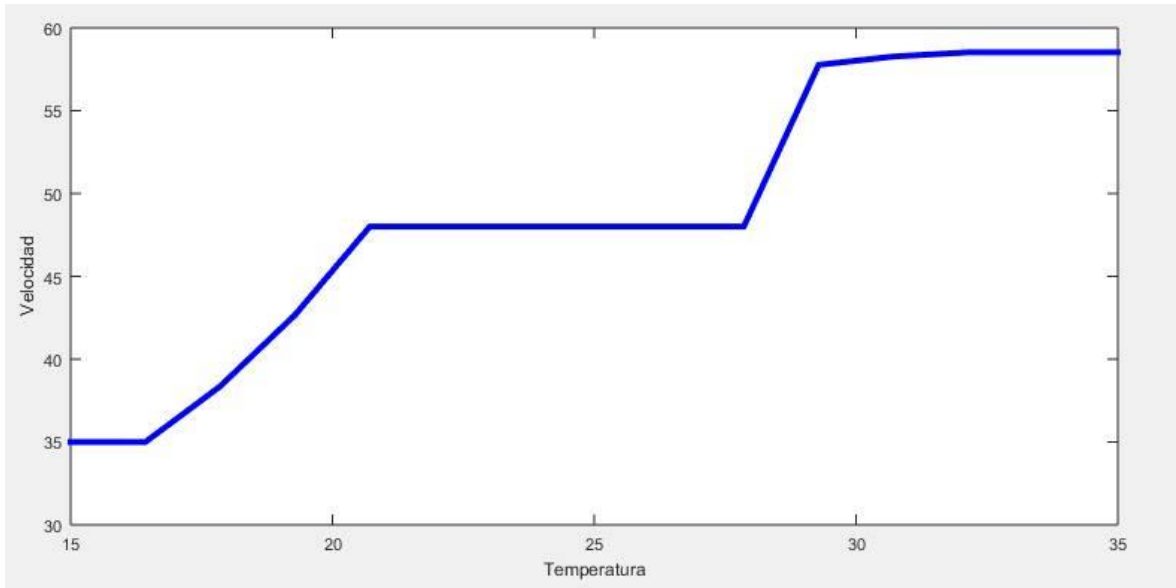


Figura 13. Velocidad con respecto a la Temperatura.

Como se ha mencionado en ocasiones anteriores el controlador difuso evalúa todos los parámetros al mismo tiempo, esto implica que para la obtención de las gráficas fue necesario dejar una señal constante y variar la otra. Para evidenciar la forma en que se comporta el controlador se elaboraron las gráficas de las figuras 14 y 15, en ellas se muestra la salida deseada en comparación con la salida del sistema.

En la figura 14 se puede observar que la salida real del sistema sigue adecuadamente la salida esperada. Para obtener esta gráfica se mantuvo constante la temperatura en un valor de 15°C tanto en la simulación como en la implementación y se modificaron los valores del CO de 0 PPM hasta 50 PPM.

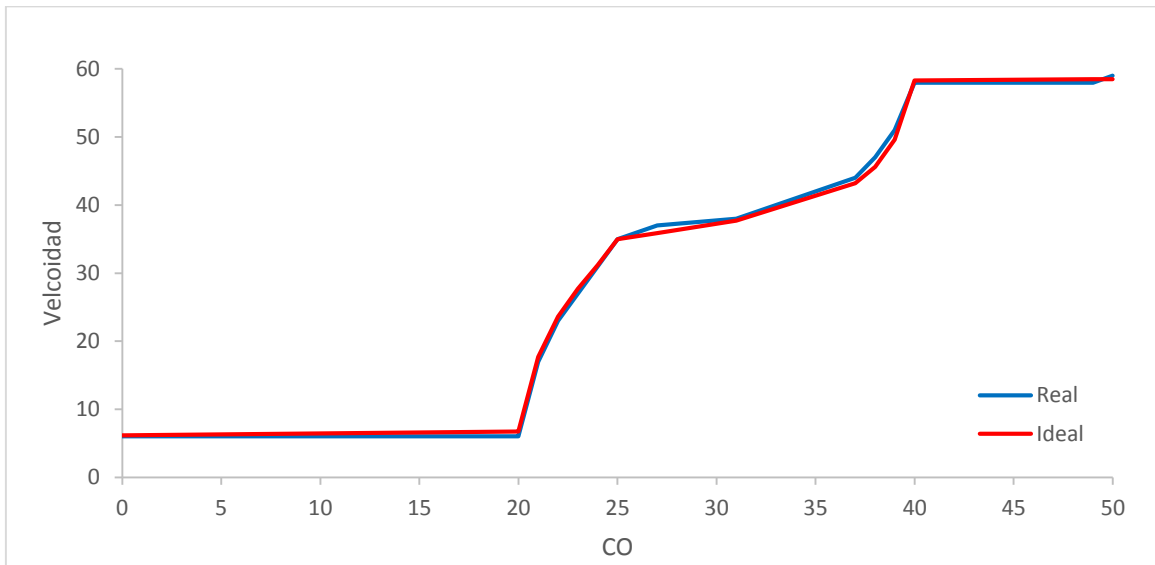


Figura 14. Salida ideal comparada con la salida real del sistema para el CO

En la figura 15 se hace la comparación del valor esperado y el valor obtenido de la velocidad con la variación de la temperatura, para conseguir está gráfica se mantuvo el CO constante en un valor igual a 0 PPM.

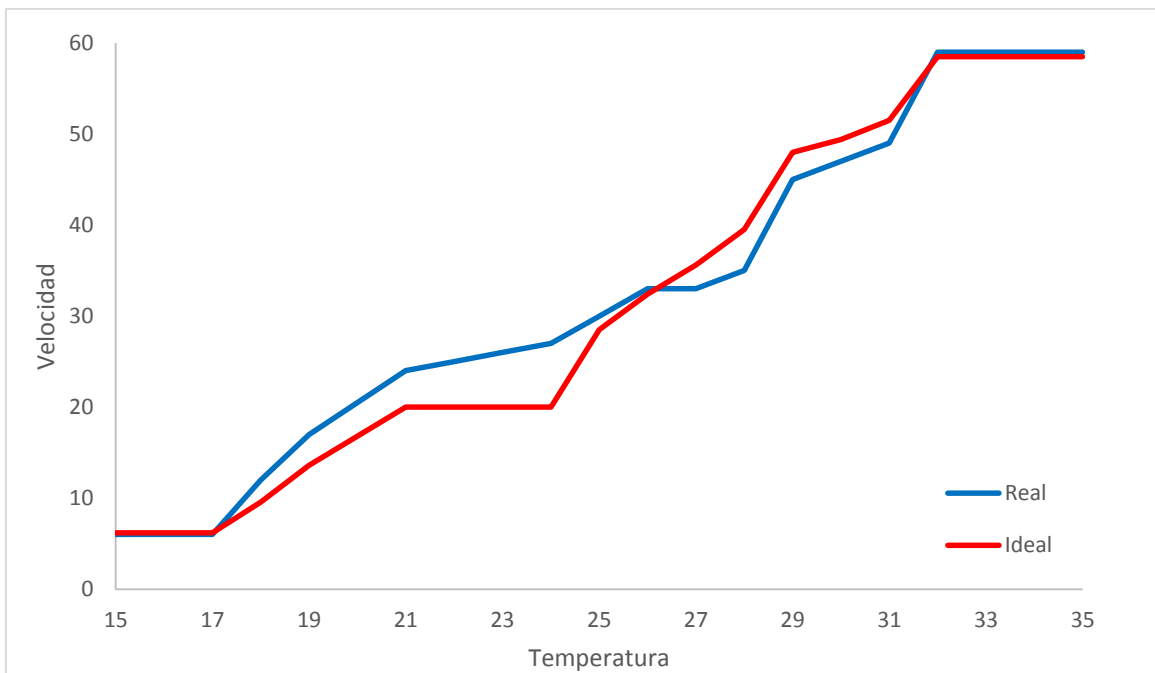


Figura 15. Salida ideal comparada con la salida real del sistema para la Temperatura

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Los humanos son más sensibles a los cambios de temperatura y podrían detectar que sus niveles se encuentran en aumento antes de encontrarse en fatal riesgo para su vida. Justificando así la razón por la cual una de las variables es más riesgosa y se tuvo mayor cuidado en el establecimiento de los parámetros y su significancia en los resultados totales del sistema de control difuso.

Otro de los logros obtenidos fue la implementación del controlador difuso diseñado en la plataforma de Arduino, el desarrollo del código se muestra en el Apéndice B de este documento. Al usar la librería eFLL que ha sido desarrollada recientemente se presentan algunas limitaciones, por ejemplo, la librería restringe el uso de tipos de funciones de membresía a solo dos: TRIMF y TRAPMF. Existen muchos más tipos de MF que podrían o no arrojar resultados mejores, aunque la actual salida del control es satisfactoria no se puede comparar con otras debido a esta restricción.

Los resultados fueron satisfactorios, la salida de sistema control sigue aproximadamente el valor que se esperaba. Comparado con la simulación que se hacía en software Matlab los valores del Arduino eran similares.

En la figura 16 se muestra la simulación que hace el software Matlab, en ella se ingresa un valor de CO igual a 11 PPM y una temperatura de 25 °C, el controlador difuso evalúa ambas variables en las distintas reglas al mismo tiempo y se obtiene una salida equivalente a una velocidad de 31,6 Hz.

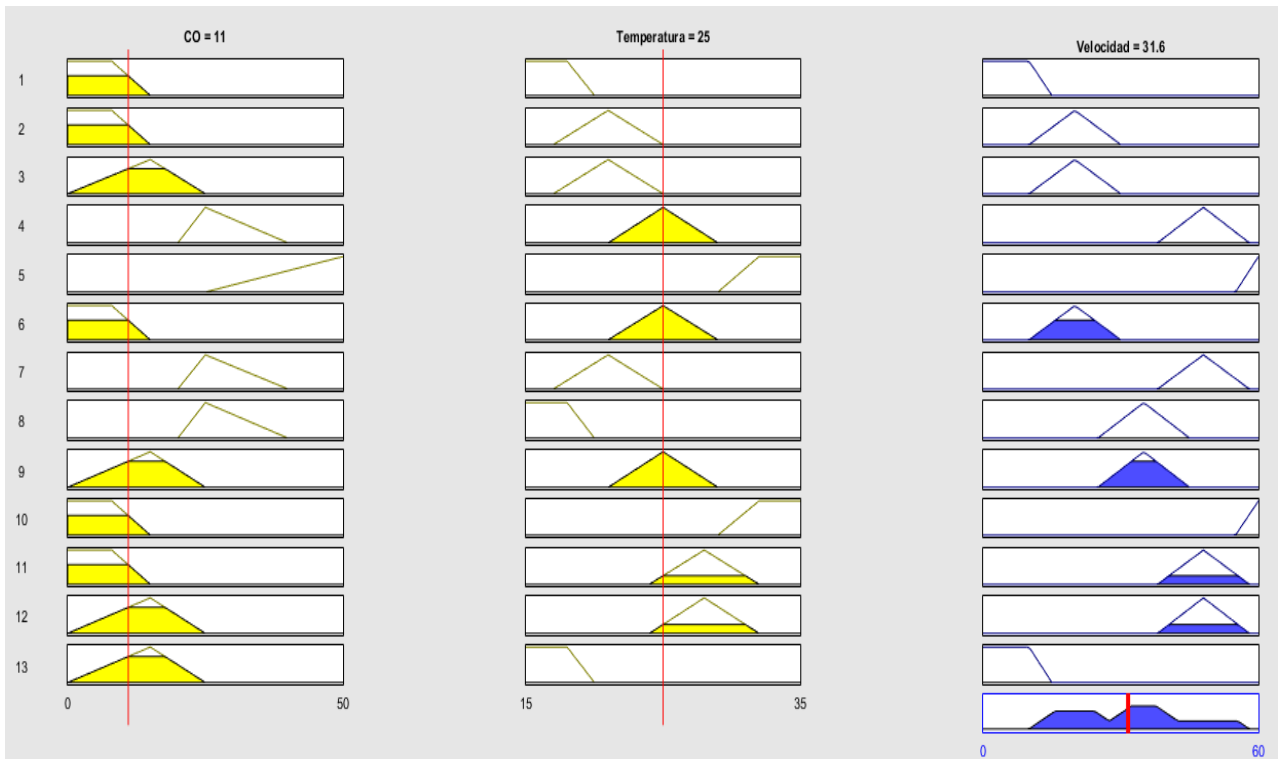


Figura 16. Simulación en el software Matlab.

En la figura 17 se puede ver los valores ingresados al controlador implementado en el Arduino que son respectivamente una temperatura de 25 °C y el CO igual a 11 PPM, esto con el fin del compararlos con la simulación en Matlab. El valor de salida obtenido es de 33Hz. También se observa que al Variador de Velocidad le llegan 32.90 Hz al momento de tomar la imagen y como se muestra en el Video 1, esta señal tiene una variación de aproximadamente +/- 1Hz.

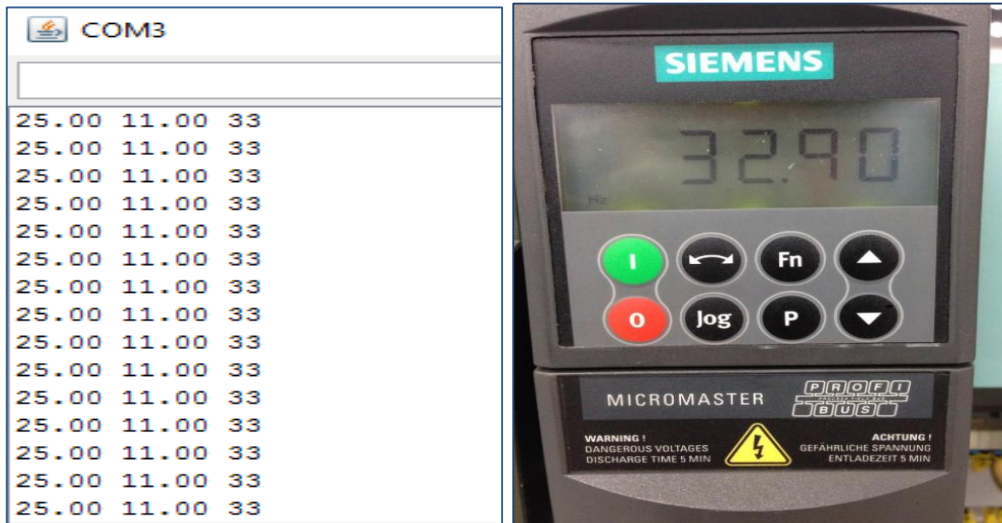


Figura 17. Visualización por el COM de las señales del Arduino y valor en el variador de frecuencia.

En consecuencia, se obtiene finalmente la implementación de todo el sistema, la conexión entre equipos y la puesta en marcha de todo el sistema de control difuso como se muestra a continuación:

La fuente de alimentación externa que fue usada para entregar los 24V que controlan encendido del Variador de Velocidad y el montaje en Arduino son mostrados en la figura 18.

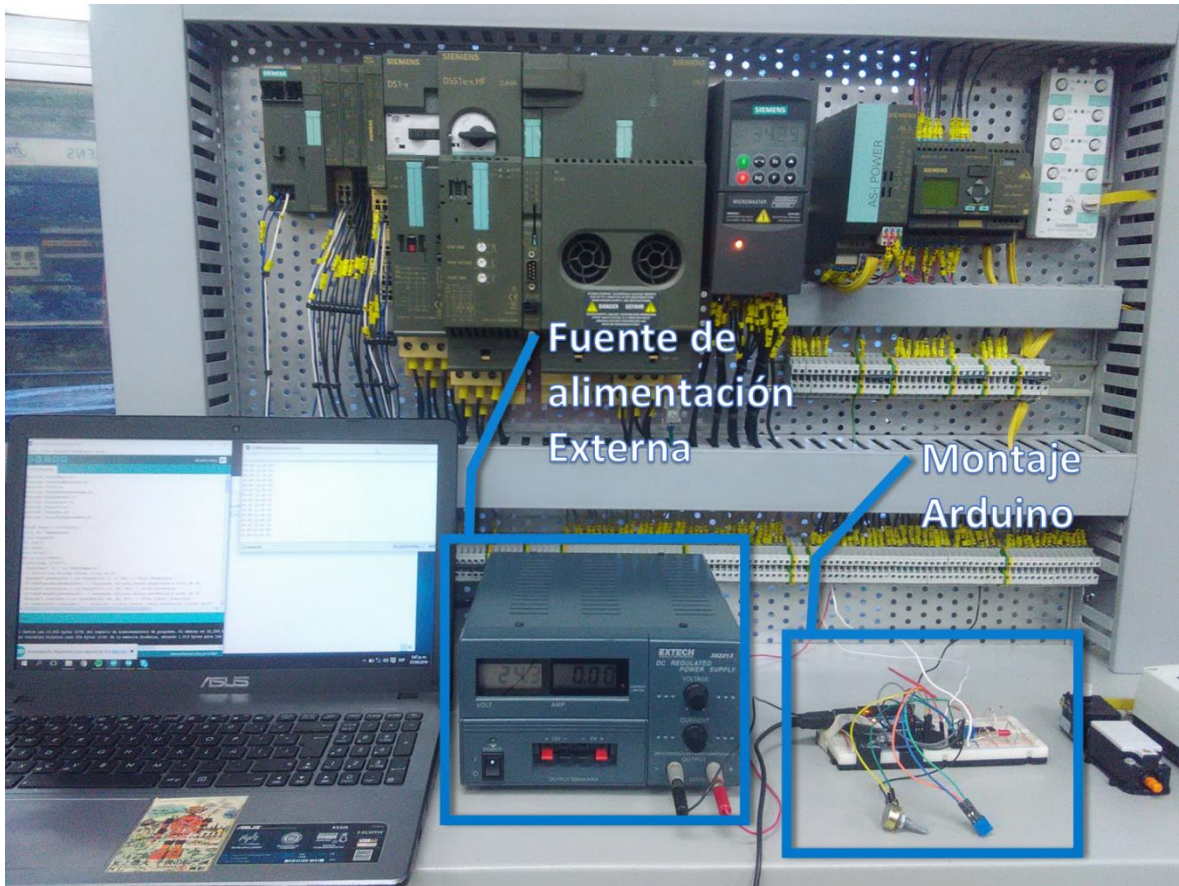


Figura 18. Implementación Final del sistema de control difuso.

En la figura 19 se muestra la visualización de señales de entrada y salida del Arduino, el variador de velocidad y el motor que es el elemento final de control. Por otra parte, en la figura 20 se puede ver la comunicación entre el Arduino y el Variador de Velocidad a través de la conexión análoga.

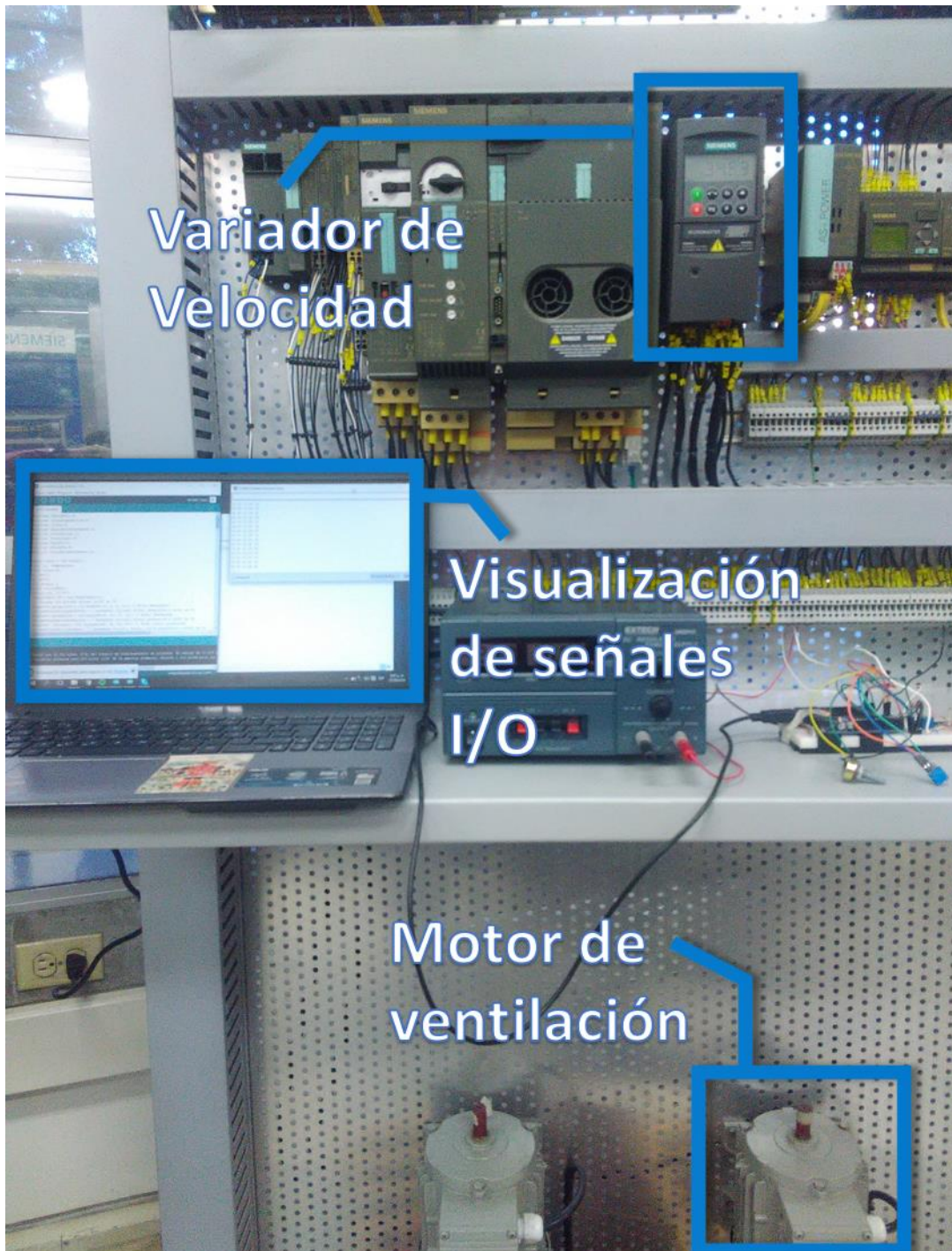


Figura 19. Implementación Final del sistema de control difuso.

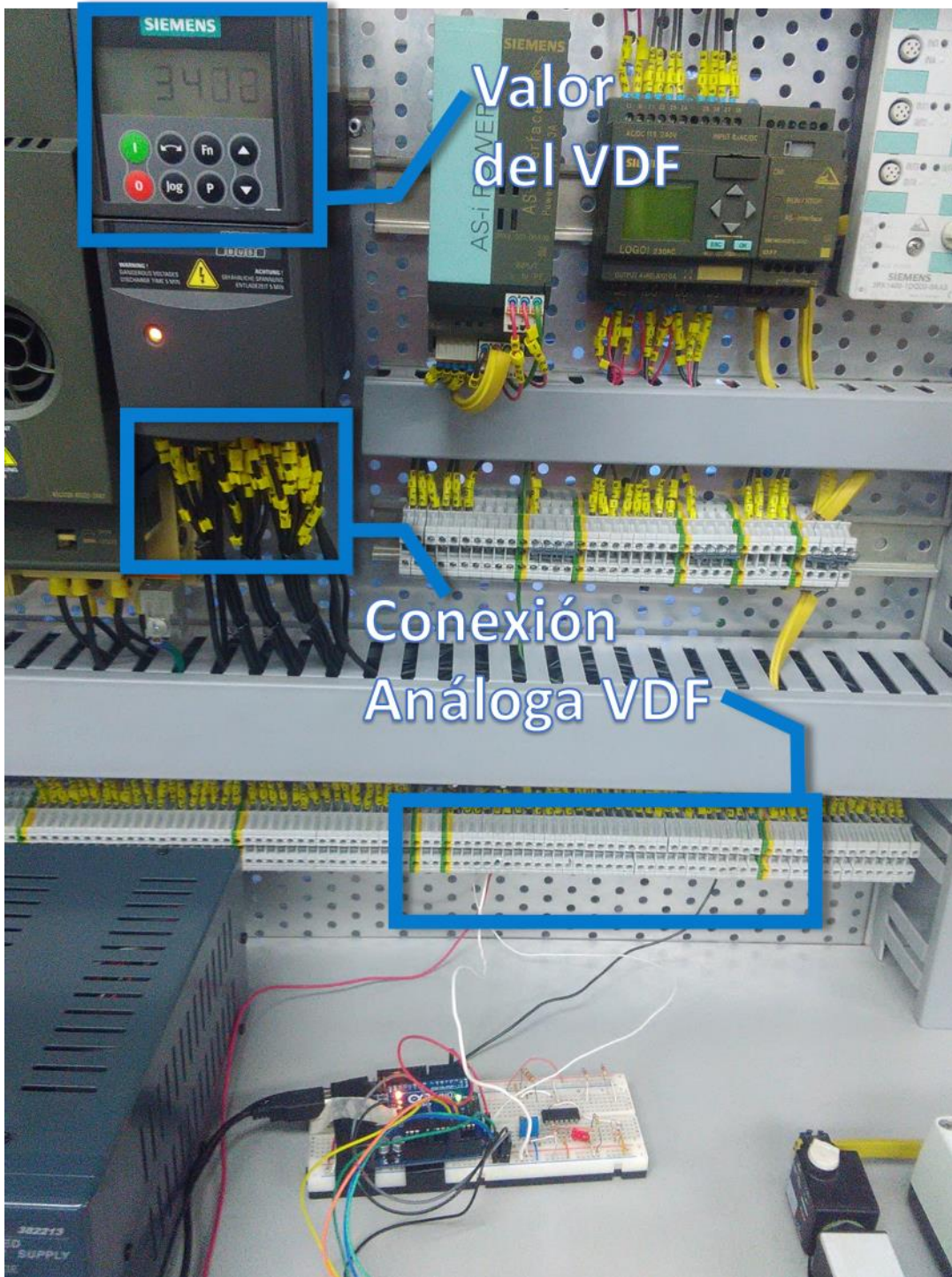


Figura 20. Implementación Final del sistema de control difuso.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Para tener una mejor perspectiva de la puesta en marcha del proyecto se anexa el siguiente vídeo en el que se aprecia el correcto funcionamiento para unos valores simulados de Temperatura y CO.



Vídeo 1. Implementación final del sistema de control.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

- En las minas subterráneas donde la ventilación no es controlada, sus operarios son expuestos a condiciones riesgosas debido al desconocimiento de niveles de gases tóxicos. Para ello se garantiza un entorno saludable y seguro en una mina subterránea de carbón por medio de un control difuso de los niveles de CO y temperatura, regulando de manera automática los motores asociados al sistema de ventilación.
- Realizar el modelo matemático de una mina subterránea de carbón con el fin de realizar un control es muy complejo, en consecuencia, es posible usar lógica difusa para diseñar un controlador eficiente que regule la temperatura y los niveles mínimos de CO y así cumplir lo decretado por el Ministerio de Minas y Energía
- Se consigue montar el sistema de control, en el cual se simulan las condiciones de una mina real y se logra la integración de equipos necesarios para obtener la respuesta deseada.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Recomendaciones

Probar distintas MF para las distintas variables (entrada y salida) y comparar los resultados entre ellos. Así podría encontrarse una MF que mejore la salida o que suavice la superficie de control.

Buscar alguna otra librería que no restrinja el tipo de MF en el control difuso, o probar con otro controlador en el que se pueda cargar un control difuso sin importar las funciones de membresía que se hayan definido en el mismo.

Trabajo Futuro

Para futuras aplicaciones, se podría pensar en usar otros gases contaminantes (Dióxido de carbono, metano, ácido sulfúrico, etc.) como variables de entrada del control. Así se puede garantizar de una manera más amplia la buena calidad del aire al interior de la mina.

Este proyecto se basa en el control difuso de CO y temperatura, pero se puede adicionar todo un sistema de monitoreo, alarmas, manipulación remota, que brinde al encargado directo de la vigilancia de la seguridad en la mina un conjunto de herramientas para generar soluciones inmediatas a emergencias presentadas.

También se podría generar un sistema de registro que almacene en una base de datos las medidas que se han hecho a lo largo del tiempo, esta información es necesaria para uso interno de la mina, como para entidades externas o gubernamentales que vigilen la seguridad de explotaciones mineras de este tipo.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

REFERENCIAS

- Vargas, E. (2015). Ventilación de Minas. Curanilahue, Chile.
- Carreño, C. (2010). Sistema de Control y Monitoreo Automatizado para Gases en Minas de Carbón. Duitama, Colombia.
- Restrepo, D. Ríos, S. Jiménez, J. (2012). Detección y control de atmósferas explosivas en minas subterráneas de carbón usando programación estructurada. Revista Educación en Ingeniería Vol. 7, N°. 14, pp. 10-21. Medellín, Colombia.
- Allen, C. L., & Tran, T. T. (2008). VENTILATION-ON-DEMAND CONTROL SYSTEM'S IMPACT. Ontario, Canadá.
- Betancur, M. J. (2011). Estudio de automatización de procesos industriales en el sector minero, (May).
- Pamukcu, C. (2007). Remote control, signalization and communication system at Eynes fully mechanized underground Colliery, Turkey. Journal of Mining Science, 43.
- Sangalli, A. (2014). Fuzzy logic. Encyclopædia Britannica.
- Barragán, A. (2009). Síntesis de Sistemas de Control Borrosos Estables por Diseño
- Escobedo, J. (2011). Tableros de Control. Universidad Tecnológica de Querétaro.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Ministerio de Minas y Energía, (2015). Decreto número 1886 de 21 de septiembre de 2015 en el cual se establece el Reglamento de Seguridad en las Labores Mineras Subterráneas.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

APÉNDICE

APÉNDICE A

En este apéndice se encuentra el código de programación del controlador difuso, que se puede cargar desde el *workspace* usando el *toolbox* “fuzzy” en el software MatLab. El archivo original se encuentra en un extensión .fis. El código es el siguiente:

```
[System]
Name='co'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=1
NumRules=13
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='CO'
Range=[0 50]
NumMFs=4
MF1='permisible':'trapmf',[-35.85 0 8 15]
MF2='prevencion':'trimf',[0 15 25]
MF3='lim-permisible':'trimf',[20 25 40]
MF4='peligro-inmediato':'trapmf',[25 50 200 200]

[Input2]
Name='Temperatura'
Range=[15 35]
NumMFs=5
MF1='Baja':'trapmf',[-5 -5 18 20]
MF2='optima':'trimf',[21 25 29]
MF3='evacuar':'trapmf',[29 32 90 90]
```

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```
MF4='media': 'trimf', [17 21 25]
```

```
MF5='alta': 'trimf', [24 28 32]
```

```
[Output1]
```

```
Name='Velocidad'
```

```
Range=[0 60]
```

```
NumMFs=5
```

```
MF1='minima': 'trapmf', [-21.6 -2.4 10 15]
```

```
MF2='alta': 'trimf', [38.01 48.01 58.01]
```

```
MF3='max': 'trapmf', [55 60 60 81.6]
```

```
MF4='moderada': 'trimf', [10 20 30]
```

```
MF5='media': 'trimf', [25 35 45]
```

```
[Rules]
```

```
1 1, 1 (1) : 1
```

```
1 4, 4 (1) : 1
```

```
2 4, 4 (1) : 1
```

```
3 2, 2 (1) : 1
```

```
4 3, 3 (1) : 2
```

```
1 2, 4 (1) : 1
```

```
3 4, 2 (1) : 1
```

```
3 1, 5 (1) : 1
```

```
2 2, 5 (1) : 1
```

```
1 3, 3 (1) : 1
```

```
1 5, 2 (1) : 1
```

```
2 5, 2 (1) : 1
```

```
2 1, 1 (1) : 1
```

APÉNDICE B

En este apéndice se encuentra el código de programación usado en el Arduino, en este se genera el sistema de control difuso con las señales de entradas y la salida que se conectará posteriormente al variador de velocidad.

El archivo original se encuentra en una extensión .ino, el código es el siguiente:

```
#include <FuzzyRule.h>
#include <FuzzyComposition.h>
#include <Fuzzy.h>
#include <FuzzyRuleConsequent.h>
```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

#include <FuzzyOutput.h>
#include <FuzzyInput.h>
#include <FuzzyIO.h>
#include <FuzzySet.h>
#include <FuzzyRuleAntecedent.h>

Fuzzy* fuzzy = new Fuzzy();
float co, Temperatura;
float output=0;
int led=3;
int outhz;
void setup() {
  Serial.begin(9600);
  pinMode(led, OUTPUT);
  FuzzyInput* CO = new FuzzyInput(1);
  // Creando una entrada difusa, nivel de CO
  FuzzySet* permisible = new FuzzySet(0, 0, 8, 15); // Nivel Permisible
  CO->addFuzzySet(permisible); // Agregando conjunto difuso permisible a nivel de
CO
  FuzzySet* prevencion = new FuzzySet(0, 15, 15, 25); // Nivel Prevención
  CO->addFuzzySet(prevencion); // Agregando conjunto difuso prevención a nivel de
CO
  FuzzySet* limiteper = new FuzzySet(20, 25, 25, 40); // Nivel Límite Permisible
  CO->addFuzzySet(limiteper); // Agregando conjunto difuso límite permisible a
nivel de CO
  FuzzySet* peligro = new FuzzySet(25, 50, 200, 1200); // Nivel Peligro Inmediato
  CO->addFuzzySet(peligro); // Agregando conjunto difuso peligro inmediato a
nivel de CO

  fuzzy->addFuzzyInput(CO); //Agregando la entrada al objeto Fuzzy

  FuzzyInput* Temp = new FuzzyInput(2);
  // Creando una entrada difusa, temperatura
  FuzzySet* baja = new FuzzySet(0, 0, 18, 20); // Temperatura Baja
  Temp->addFuzzySet(baja); // Agregando conjunto difuso baja a temperatura
  FuzzySet* media = new FuzzySet(17, 21, 21, 25); // Temperatura Media
  Temp->addFuzzySet(media); // Agregando conjunto difuso media a temperatura
  FuzzySet* optima = new FuzzySet(21, 25, 25, 29); // Temperatura Óptima
  Temp->addFuzzySet(optima); // Agregando conjunto difuso óptima a temperatura
  FuzzySet* alta = new FuzzySet(17, 21, 21, 25); // Temperatura Alta

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

Temp->addFuzzySet(alta); // Agregando conjunto difuso alta a temperatura
FuzzySet* evacuar = new FuzzySet(21, 25, 25, 29); // Temperatura Evacuar
Temp->addFuzzySet(evacuar); // Agregando conjunto difuso evacuar a temperatura

fuzzy->addFuzzyInput(Temp); // Agregando la entrada al objeto Fuzzy

// Creando salida difusa velocidad
FuzzyOutput* Vel = new FuzzyOutput(1);
// Creando los conjuntos difusos de la salida Velocidad
FuzzySet* Min = new FuzzySet(0, 0, 10, 15); // Velocidad Mínima
Vel->addFuzzySet(Min); // Agregando conjunto difuso mínima a velocidad
FuzzySet* moderada = new FuzzySet(10, 20, 20, 30); // Velocidad Moderada
Vel->addFuzzySet(moderada); // Agregando conjunto difuso moderada a velocidad
FuzzySet* Med = new FuzzySet(25, 35, 35, 45); // Velocidad media
Vel->addFuzzySet(Med); // Agregando conjunto difuso media a velocidad
FuzzySet* Alt = new FuzzySet(38, 48, 48, 58); // Velocidad Moderada
Vel->addFuzzySet(Alt); // Agregando conjunto difuso moderada a velocidad
FuzzySet* Max = new FuzzySet(55, 60, 60, 62); // Velocidad Máxima
Vel->addFuzzySet(Max); // Agregando conjunto difuso máxima a velocidad

fuzzy->addFuzzyOutput(Vel); // Agregando la salida al objeto Fuzzy

//----- Montando as regras Fuzzy
// Regra 1
FuzzyRuleAntecedent* COpermisibleandTempbaja = new FuzzyRuleAntecedent(); //
Instanciando um Antecedente para a expresso
COpermisibleandTempbaja->joinWithAND(permisible, baja); // Adicionando o
FuzzySet correspondente ao objeto Antecedente
FuzzyRuleConsequent* Velmin = new FuzzyRuleConsequent(); // Instancinado um
Consequente para a expressao
Velmin->addOutput(Min); // Adicionando o FuzzySet correspondente ao objeto
Consequente
// Instanciando um objeto FuzzyRule
FuzzyRule* fuzzyRule1 = new FuzzyRule(1, COpermisibleandTempbaja, Velmin); //
Passando o Antecedente e o Consequente da expressao
fuzzy->addFuzzyRule(fuzzyRule1); // Adicionando o FuzzyRule ao objeto Fuzzy

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

// Regra 2
FuzzyRuleAntecedent* COpermisibleandTempmedia = new FuzzyRuleAntecedent(); //
Instanciando um Antecedente para a expresso
COpermisibleandTempmedia->joinWithAND(permisible, media); // Adicionando o
FuzzySet correspondente ao objeto Antecedente
FuzzyRuleConsequent* Velmoderada = new FuzzyRuleConsequent(); // Instancinado
um Consequente para a expressao
Velmoderada->addOutput(moderada); // Adicionando o FuzzySet correspondente ao
objeto Consequente
// Instanciando um objeto FuzzyRule
FuzzyRule* fuzzyRule2 = new FuzzyRule(2, COpermisibleandTempmedia,
Velmoderada); // Passando o Antecedente e o Consequente da expressao
fuzzy->addFuzzyRule(fuzzyRule2); // Adicionando o FuzzyRule ao objeto Fuzzy

// Regra 3
FuzzyRuleAntecedent* COpvencionandTempmedia = new FuzzyRuleAntecedent(); //
Instanciando um Antecedente para a expresso
COpvencionandTempmedia->joinWithAND(vencion, media); // Adicionando o
FuzzySet correspondente ao objeto Antecedente
FuzzyRuleConsequent* Velmoderada2 = new FuzzyRuleConsequent(); // Instancinado
um Consequente para a expressao
Velmoderada2->addOutput(moderada); // Adicionando o FuzzySet correspondente ao
objeto Consequente
// Instanciando um objeto FuzzyRule
FuzzyRule* fuzzyRule3 = new FuzzyRule(3, COpvencionandTempmedia,
Velmoderada2); // Passando o Antecedente e o Consequente da expressao
fuzzy->addFuzzyRule(fuzzyRule3); // Adicionando o FuzzyRule ao objeto Fuzzy

// Regra 4
FuzzyRuleAntecedent* COLimiteperandTempoptima = new FuzzyRuleAntecedent(); //
Instanciando um Antecedente para a expresso
COLimiteperandTempoptima->joinWithAND(limiteper, optima); // Adicionando o
FuzzySet correspondente ao objeto Antecedente
FuzzyRuleConsequent* VelAlt = new FuzzyRuleConsequent(); // Instancinado um
Consequente para a expressao
VelAlt->addOutput(Alt); // Adicionando o FuzzySet correspondente ao objeto
Consequente
// Instanciando um objeto FuzzyRule

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

FuzzyRule* fuzzyRule4 = new FuzzyRule(4, COlimiteperandTempoptima, VelAlt); //
Passando o Antecedente e o Consequente da expressao
fuzzy->addFuzzyRule(fuzzyRule4); // Adicionando o FuzzyRule ao objeto Fuzzy

// Regra 5
FuzzyRuleAntecedent* COpeligroorTempevacuar = new FuzzyRuleAntecedent(); //
Instanciando um Antecedente para a expresso
COpeligroorTempevacuar->joinWithOR(peligro, evacuar); // Adicionando o FuzzySet
correspondente ao objeto Antecedente
FuzzyRuleConsequent* VelMax = new FuzzyRuleConsequent(); // Instancinado um
Consequente para a expressao
VelMax->addOutput(Max); // Adicionando o FuzzySet correspondente ao objeto
Consequente
// Instanciando um objeto FuzzyRule
FuzzyRule* fuzzyRule5 = new FuzzyRule(5, COpeligroorTempevacuar, VelMax); //
Passando o Antecedente e o Consequente da expressao
fuzzy->addFuzzyRule(fuzzyRule5); // Adicionando o FuzzyRule ao objeto Fuzzy

// Regra 6
FuzzyRuleAntecedent* COpermisibleandTempoptima = new FuzzyRuleAntecedent(); //
Instanciando um Antecedente para a expresso
COpermisibleandTempoptima->joinWithAND(permisible, optima); // Adicionando o
FuzzySet correspondente ao objeto Antecedente
FuzzyRuleConsequent* Velmoderada3 = new FuzzyRuleConsequent(); // Instancinado
um Consequente para a expressao
Velmoderada3->addOutput(moderada); // Adicionando o FuzzySet correspondente ao
objeto Consequente
// Instanciando um objeto FuzzyRule
FuzzyRule* fuzzyRule6 = new FuzzyRule(6, COpermisibleandTempoptima,
Velmoderada3); // Passando o Antecedente e o Consequente da expressao
fuzzy->addFuzzyRule(fuzzyRule6); // Adicionando o FuzzyRule ao objeto Fuzzy

// Regra 7
FuzzyRuleAntecedent* COlimiteperandTempmedia = new FuzzyRuleAntecedent(); //
Instanciando um Antecedente para a expresso
COlimiteperandTempmedia->joinWithAND(limiteper, media); // Adicionando o
FuzzySet correspondente ao objeto Antecedente

```

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

FuzzyRuleConsequent* VelAlt2 = new FuzzyRuleConsequent(); // Instancinado um
Consequente para a expressao
VelAlt2->addOutput(Alt); // Adicionando o FuzzySet correspondente ao objeto
Consequente
// Instanciando um objeto FuzzyRule
FuzzyRule* fuzzyRule7 = new FuzzyRule(7, ColimiteperandTempmedia, VelAlt2); //
Passando o Antecedente e o Consequente da expressao
fuzzy->addFuzzyRule(fuzzyRule7); // Adicionando o FuzzyRule ao objeto Fuzzy

// Regra 8
FuzzyRuleAntecedent* ColimiteperandTempbaja = new FuzzyRuleAntecedent(); //
Instanciando um Antecedente para a expresso
ColimiteperandTempbaja->joinWithAND(limiteper, baja); // Adicionando o FuzzySet
correspondente ao objeto Antecedente
FuzzyRuleConsequent* VelMed = new FuzzyRuleConsequent(); // Instancinado um
Consequente para a expressao
VelMed->addOutput(Med); // Adicionando o FuzzySet correspondente ao objeto
Consequente
// Instanciando um objeto FuzzyRule
FuzzyRule* fuzzyRule8 = new FuzzyRule(8, ColimiteperandTempbaja, VelMed); //
Passando o Antecedente e o Consequente da expressao
fuzzy->addFuzzyRule(fuzzyRule8); // Adicionando o FuzzyRule ao objeto Fuzzy

// Regra 9
FuzzyRuleAntecedent* COpvencionandTempoptima = new FuzzyRuleAntecedent(); //
Instanciando um Antecedente para a expresso
COpvencionandTempoptima->joinWithAND(prevencion, optima); // Adicionando o
FuzzySet correspondente ao objeto Antecedente
FuzzyRuleConsequent* VelMed2 = new FuzzyRuleConsequent(); // Instancinado um
Consequente para a expressao
VelMed2->addOutput(Med); // Adicionando o FuzzySet correspondente ao objeto
Consequente
// Instanciando um objeto FuzzyRule
FuzzyRule* fuzzyRule9 = new FuzzyRule(9, COpvencionandTempoptima, VelMed2);
// Passando o Antecedente e o Consequente da expressao
fuzzy->addFuzzyRule(fuzzyRule9); // Adicionando o FuzzyRule ao objeto Fuzzy

// Regra 10

```


 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

FuzzyRuleAntecedent* COpermisibleandTempevacuar = new FuzzyRuleAntecedent(); //
Instanciando um Antecedente para a expresso
COpermisibleandTempevacuar->joinWithAND(permisible, evacuar); // Adicionando o
FuzzySet correspondente ao objeto Antecedente
FuzzyRuleConsequent* VelMax2 = new FuzzyRuleConsequent(); // Instancinado um
Consequente para a expressao
VelMax2->addOutput(Max); // Adicionando o FuzzySet correspondente ao objeto
Consequente
// Instanciando um objeto FuzzyRule
FuzzyRule* fuzzyRule10 = new FuzzyRule(10, COpermisibleandTempevacuar,
VelMax2); // Passando o Antecedente e o Consequente da expressao
fuzzy->addFuzzyRule(fuzzyRule10); // Adicionando o FuzzyRule ao objeto Fuzzy

// Regra 11
FuzzyRuleAntecedent* COpermisibleandTempalta = new FuzzyRuleAntecedent(); //
Instanciando um Antecedente para a expresso
COpermisibleandTempalta->joinWithAND(permisible, alta); // Adicionando o
FuzzySet correspondente ao objeto Antecedente
FuzzyRuleConsequent* VelAlt3 = new FuzzyRuleConsequent(); // Instancinado um
Consequente para a expressao
VelAlt3->addOutput(Alt); // Adicionando o FuzzySet correspondente ao objeto
Consequente
// Instanciando um objeto FuzzyRule
FuzzyRule* fuzzyRule11 = new FuzzyRule(11, COpermisibleandTempalta, VelAlt3);
// Passando o Antecedente e o Consequente da expressao
fuzzy->addFuzzyRule(fuzzyRule11); // Adicionando o FuzzyRule ao objeto Fuzzy

// Regra 12
FuzzyRuleAntecedent* COpvencionandTempalta = new FuzzyRuleAntecedent(); //
Instanciando um Antecedente para a expresso
COpvencionandTempalta->joinWithAND(vencion, alta); // Adicionando o
FuzzySet correspondente ao objeto Antecedente
FuzzyRuleConsequent* VelAlt4 = new FuzzyRuleConsequent(); // Instancinado um
Consequente para a expressao
VelAlt4->addOutput(Alt); // Adicionando o FuzzySet correspondente ao objeto
Consequente
// Instanciando um objeto FuzzyRule
FuzzyRule* fuzzyRule12 = new FuzzyRule(12, COpvencionandTempalta, VelAlt4);
// Passando o Antecedente e o Consequente da expressao

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

fuzzy->addFuzzyRule(fuzzyRule12); // Adicionando o FuzzyRule ao objeto Fuzzy

// Regra 13
FuzzyRuleAntecedent* COpvencionandTempbaja = new FuzzyRuleAntecedent(); //
Instanciando um Antecedente para a expresso
COpvencionandTempbaja->joinWithAND(preveccion, baja); // Adicionando o
FuzzySet correspondente ao objeto Antecedente
FuzzyRuleConsequent* VelMin2 = new FuzzyRuleConsequent(); // Instancinado um
Consequente para a expressao
VelMin2->addOutput(Min); // Adicionando o FuzzySet correspondente ao objeto
Consequente
// Instanciando um objeto FuzzyRule
FuzzyRule* fuzzyRule13 = new FuzzyRule(13, COpvencionandTempbaja, VelMin2);
// Passando o Antecedente e o Consequente da expressao
fuzzy->addFuzzyRule(fuzzyRule13); // Adicionando o FuzzyRule ao objeto Fuzzy

}

//-----
// Main Loop
//-----
void loop()
{
  unsigned long Time5;
  int n = 0;
  Time5 = millis();

  while (millis() < Time5 + 100) {
    co = analogRead(0);
    Temperatura = analogRead(1);

    n++;
  }

  co = map(co, 0, 256, 0, 50);
  Temperatura = map(Temperatura, 0, 256, 0, 40);
  fuzzy->setInput(1, co);

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

fuzzy->setInput(2, Temperatura);
fuzzy->fuzzify();

output = fuzzy->defuzzify(1);
outhz=output;
output = map(output, 0, 60, 0, 256);
analogWrite(led, output );
Serial.print(Temperatura); Serial.print(" "); Serial.print(co); Serial.print("
"); Serial.print(outhz); Serial.print(" ");
    Serial.println();
}

```

FIRMA ESTUDIANTES DIEGO USQUIANO S.

FIRMA ASESOR Juliana Valencia A

FECHA ENTREGA: _____

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD _____

RECHAZADO___ ACEPTADO___ ACEPTADO CON MODIFICACIONES___

ACTA NO. _____

FECHA ENTREGA: _____

FIRMA CONSEJO DE FACULTAD _____

ACTA NO. _____

FECHA ENTREGA: _____