# Controladores de lógica difusa para el funcionamiento de una bicicleta inteligente

Huerta Emanuel, Segura Daniel, Ochoa Andrés, Zúñiga David, Jiménez Jaime

*INSTITUTO Tecnológico y de Estudios Superiores de Monterrey*
*Ciudad de México, México*

A01653094@itesm.mx
A01653823@itesm.mx
A01651898@itesm.mx
A01378821@itesm.mx
A01371929@itesm.mx

***Abstract*— Las bicicletas son un medio de transporte utilizado en todas partes del mundo y con distintos tipos de aplicaciones. El presente trabajo desarrolla un sistema de navegación basado en lógica difusa enfocado a una bicicleta eléctrica inteligente para dos tipos de controladores, uno crucero y otro multivariable, ambos implementados en el software MATLAB-Simulink, LabVIEW y arduino. Este trabajo se basa en ajustar una señal PWM (Pulse Width Modulation) para controlar la velocidad de un motor DC así como la intermitencia de las luces traseras que estén sometidas a variaciones de diversos factores al momento de manejar una bicicleta.**

## I. INTRODUCCIÓN

Con los grandes saltos en avances tecnológicos de los últimos años, el control de máquinas eléctricas se ha vuelto un tema crítico para el funcionamiento de todos nuestros sistemas operando en el mundo. Cada año, la necesidad de nuevos instrumentos de control incrementa en demanda debido a las aplicaciones de precisión requeridas para el correcto funcionamiento de los procesos y operaciones ingenieriles.

Existe una gran variedad de controladores de máquinas eléctricas que permiten obtener o aproximar el valor deseado de referencia de un motor DC. Estos controladores son la base fundamental del funcionamiento de prácticamente todos nuestros sistemas.

Contrario a los controladores clásicos proporcional-integral-derivativos (PID), los controladores de lógica difusa no requieren modelos matemáticos del sistema para poder ser controlado. La toma de decisión de estos controladores no es más que la de un experto, lo que significa que las máquinas están programadas para tomar decisiones similares a las de los humanos mediante el uso de lógica difusa y la teoría de conjuntos difusos.

Debido a su adaptabilidad en la naturaleza, el control difuso ha ganado gran importancia a lo largo de los años para controlar tanto sistemas lineales como no lineales, aunque se puede decir que nada es lineal en la vida real, simplemente se linealiza utilizando métodos de aproximación. Sin embargo, cualquier proceso de linealización reduce la precisión del sistema, lo que lleva a errores operativos con los controladores PID que requieren modelos matemáticos para su diseño.

Una de las ventajas de los controladores difusos es que los errores dados por la linealización no afectan en su desempeño, por el contrario, lo incrementa mientras simplifica y reduce el costo del sistema, razón por la cual se optó por diseñar los dos controladores de la bicicleta inteligente mediante el método de lógica difusa.

La intención de este trabajo es simular el funcionamiento de una bicicleta inteligente en diferentes estados de operación con el objetivo de realizar los controladores de lógica difusa e implementarlos mediante los softwares MATLAB-Simulink y LabVIEW y así apreciar las ventajas del mismo.

El diseño de un control PID para un sistema como una bicicleta inteligente puede ser complejo debido a la gran variedad de factores que afectan su comportamiento. A partir de variables de interés, el control lógico difuso fue realizado para dos casos mediante los cuales la bicicleta inteligente puede ser controlada de una forma adecuada.

## II. MARCO TEÓRICO

Los componentes principales para la elaboración del presente proyecto son: un motor de corriente directa de 3 volts, una tarjeta Arduino y un puente H (L298N) cuyo funcionamiento y forma de implementar es de suma importancia clarificar. Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para distintos tipos de desarrollos.

La placa de Arduino está basada en microcontroladores en los que se puede grabar instrucciones con lenguaje de programación, el microcontrolador de Arduino posee una interfaz de entrada para conectar en la placa diferentes tipos de periféricos. También cuenta con una interfaz de salida, la cual se encarga de llevar la información que se ha procesado en el arduino a otros periféricos como pantallas, altavoces, entre otros. Está compuesta por puertos de entrada y de salida que pueden ser tanto analógicos como digitales.

Las salidas digitales de tipo PWM fueron utilizadas en este proyecto para controlar la velocidad de un motor de 3 volts DC y a su vez, controlar la intermitencia de un LED. A continuación, se muestra un diagrama de la estructura de una tarjeta de Arduino UNO y los puertos que la componen, se tuvo presente que algunos son especialmente para utilizar señales PWM.

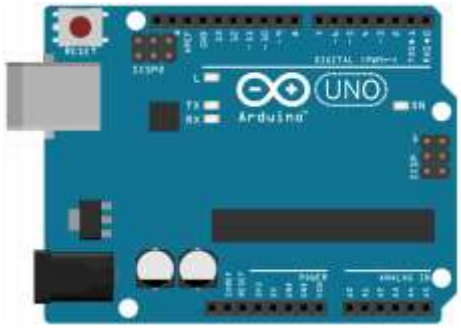


Fig. 1 Arquitectura de la tarjeta Arduino. [1]

La tarjeta Arduino se acopló a un módulo puente H (L298N), el cual es una tarjeta empleada para el cambio de giro de motores de corriente directa, motores a pasos, solenoides y en general cualquier otra carga inductiva. Esta dispone en su interior de dos puentes H independientes con capacidad de conducir dos amperios constantes.

La tarjeta expone las conexiones hacia el motor a través de bloques de terminales, mientras que las entradas de control y habilitación del puente H son accesibles a través de headers macho estándar. Es necesario conocer su estructura para poder realizar las conexiones correctas.

El diagrama de pines y entradas del puente H se puede observar a continuación, en el cual se debe aclarar que los jumpers de corto circuito en ENA y ENB, fueron removidos para poder introducir la señal PWM. Fig. 2 muestra la arquitectura del puente H donde se puede apreciar el puerto de entrada de 12 volts y GND que fueron utilizados. Por otro lado, *IN3* e *IN4* funcionan para controlar la velocidad del motor de corriente directa conectado en las terminales Out 3 y Out 4.

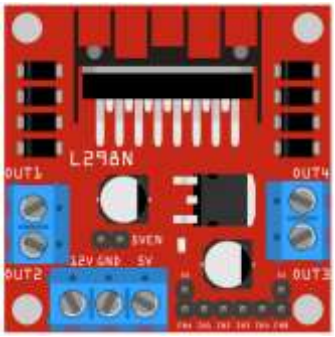


Fig. 2 Arquitectura del puente H (L298N) [2]

Existen dos tipos de motores, de corriente directa y de corriente alterna. Los motores de corriente directa o DC, son máquinas versátiles y útiles para la transformación de energía, estas máquinas transforman energía eléctrica en mecánica. Los motores eléctricos se componen principalmente de dos partes; *el rotor y estator*. La primera es la parte que gira mientras que el estator es la que se mantiene estática en todo momento. El objetivo de cualquier motor es inducir fuerza eléctrica magnética en el rotor para poder generar un par. Los motores de corriente directa usualmente poseen un anillo con el cual conmuta la corriente en los devanados del rotor de la máquina. Fig. 3 muestra las distintas partes de la máquina de corriente directa.

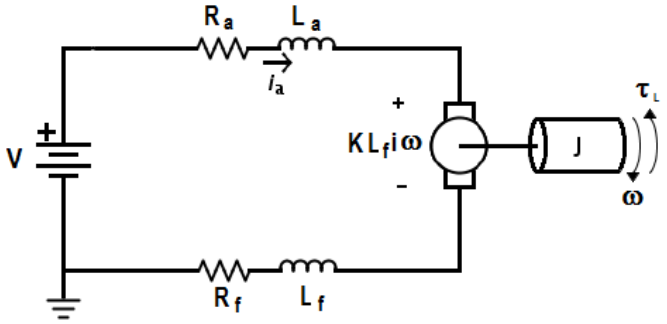


Fig. 3 Diagrama de corriente directa. [3]

La diferencia entre las máquinas de corriente continua y los transformadores es el movimiento, además, los motores usualmente están creados a partir de hierro, esto con la finalidad de concentrar el campo magnético y evitar pérdidas. Otra característica de los motores de corriente directa es que el rotor puede estar construido mediante imanes permanentes, en lugar de los devanados por donde circula una corriente eléctrica. [4]

Los motores de corriente continua se dividen en 5 tipos los cuales son: excitación separada, en derivación o *Shunt*, en serie, compuesto e imanes permanentes. Para el presente proyecto se utiliza un motor de corriente directa de imanes permanentes de 3 volts.

La bicicleta es un medio de transporte eficaz y económico, es utilizado por gran parte de la sociedad en diferentes naciones y ha ido evolucionando a lo largo de la historia, hoy en día podemos encontrar bicicletas eléctricas de distintos tipos que son cada vez más utilizadas. [5] La bicicleta eléctrica es aquella que tiene todas las partes de cualquier bicicleta común, la única variante es que está equipada con un motor eléctrico, una batería de la cual se alimenta y un controlador.

Los motores implementados en estas bicicletas tienen una potencia que varía entre 250 y 700 watts. Las baterías más comunes que se utilizan son de litio, níquel e hidruro metálico y plomo. Los controladores son de suma importancia en las bicicletas, ya que tienen la finalidad de suministrar de manera correcta la corriente y voltaje para distintas situaciones de uso. [6]

Para comprender los conceptos de lógica difusa que hay en los softwares MATLAB-Simulink y LabVIEW, se explicarán a continuación conceptos importantes que se emplean a lo largo del trabajo.

Los conjuntos difusos son un elemento central al momento de diseñar controladores. Los conjuntos difusos se caracterizan por funciones de membresía, estas funciones de membresía son una especie de números difusos. Por ejemplo, dos colores definidos en el universo de colores se mezclan y se muestran en la Fig. 4, primero el color es blanco, luego se cambia a negro sobre una región de transición, de modo que se vuelve gris claro, gris, gris oscuro y negro a medida que nos movemos de izquierda a derecha y no es un sólo color claro durate la transición, también incluye tonos tanto de blanco como de negro y no se puede distinguir un color del otro porque la parte de transición es difusa. [7]

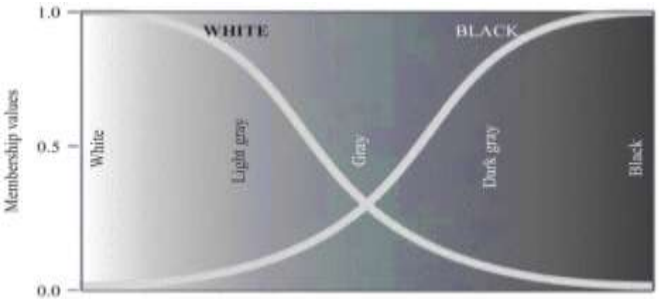

Fig. 4 Representación de conjuntos difusos mediante colores.[7]

El concepto de difusividad proviene de la incertidumbre, si los datos no son nítidos y sólidos para distinguir uno del otro, se representan mediante grados de inclusión en categorías relacionadas.

Las funciones de membresía o pertenencia difusa pueden considerarse como un puente entre datos inciertos y un mundo difuso. El nítido universo de datos difusos se divide en subsecciones y está representado por funciones de membresía difusas. Un universo de cualquier variable se puede dividir en subsecciones de alguna característica de ella. Como se aprecia en la Fig. 5, las funciones de membresía difusa se definen en un intervalo que va de 0 a 1, el valor más bajo (cero) significa que no hay inclusión en el conjunto difuso relacionado, mientras que el más alto (valor de uno) quiere decir inclusión, total.

Fig. 5 representa cinco funciones de pertenencia de tipo triangular y se utilizan cinco subconjuntos para definir los tonos de gris. Los triángulos se utilizan como funciones de pertenencia a los límites para representar la parte inferior (blanca) y límites superiores (negros) de tonos de gris. Los triángulos equiláteros se utilizan como funciones de membresía para los otros tonos que varía de blanco a negro. No es correcto decir que donde un subconjunto termina, empieza el otro, ya que se observa cómo comparten una sección del subconjunto. [7]

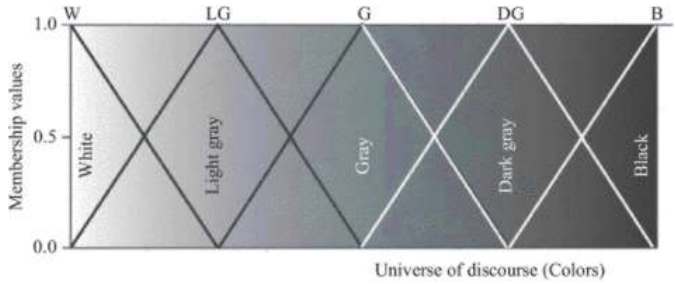

Fig. 5 Representación de subconjuntos difusos en una escala de colores. [7]

Una de las funciones de membresía más empleadas en el trabajo es la de tipo trapezoidal, este tipo de funciones se obtiene de la combinación de ecuaciones de línea como en el tipo triangular. Las funciones de membresía trapezoidal difieren de los tipos triangulares con su región en punta en la parte superior, ya que estas tienen una parte superior plana que no es difusa. Por su parte, las funciones de membresía gaussiana se asemejan en forma con las de tipo triangular, con la diferencia que la forma de la campana se hace cada vez más estrecha.

## III. DESCRIPCIÓN DE ACTIVIDADES

La primera actividad engloba toda la planeación del control lógico difuso para el control de crucero de la bicicleta inteligente. Se describen las variables de entrada de interés, las reglas o etiquetas de la matriz de asociación difusa y la respuesta correspondiente del controlador. Posteriormente se muestran los prototipos del control realizados en SIMULINK y en LabVIEW, donde se implementa dicho controlador a un motor de 3 volts DC a partir de una tarjeta Arduino.

La segunda actividad engloba el desarrollo de un controlador con múltiples variables de entrada y salida asociadas con los fenómenos físicos que puede encontrar el usuario al momento de manejar dicho vehículo, al igual que la primera parte, se describen las variables de interés con sus correspondientes reglas obtenidas de la matriz de asociación difusa para su posterior implementación.

El propósito de tener dos controladores para la bicicleta inteligente es simular dos estados de operación y extender la aplicabilidad del controlador difuso. Con ello, se pretende alcanzar un sistema de control en la bicicleta donde el usuario pueda dictar las variables principales correspondientes y con ello obtener una respuesta óptima bajo el estado de operación actual.

### *A. Controlador Crucero*

El primer controlador busca asistir al ciclista manipulando la velocidad de un motor DC acoplado a la transmisión del vehículo. Se toman en cuenta cinco diferentes tipos de variables de entrada las cuales tienen distintos estados. A partir de estas entradas se define una sola salida PWM para el motor DC de asistencia.

Señales de Entrada

#### *1. Type of Bike (Tipo de bicicleta)*

Dentro de esta variable de entrada se encuentran dos tipos de bicicletas, *Road bike* y *Mountain bike.* Este tipo de variable resulta importante para el controlador ya que las bicicletas de ruta tienden a ser mucho más ligeras a comparación de las de montaña. El torque de control para el motor DC puede ser de menor magnitud para llevar al ciclista de ruta a la velocidad deseada. A pesar de que las bicicletas de montaña son más pesadas y requieren un mayor torque para incrementar su velocidad, dichas bicicletas cuentan con suspensión, por lo que la velocidad de salida puede ser aumentada sin preocupación de irregularidades en la superficie. Por otro lado, el rango entre estas variables va de 0 a 1, donde 0 es *Full road bike*, 1 es *Full mountain bike* y los valores intermedios corresponden a bicicletas de otro tipo. [8]

#### *2. Rider's Weight (Peso del usuario)*

En el análisis de la dinámica de un vehículo, la masa equivalente es una de las variables más importantes que dictan el rendimiento de dicho vehículo. Por esta razón, la masa del ciclista es una variable de entrada fundamental para el controlador, en el cual se considera la masa de la bicicleta a partir de la variable anterior. Esta variable de entrada está dividida en tres categorías; *Lightweight, Middleweight* y *Heavyweight* las cuales varían en un rango de 0 a 120 kilogramos.

#### *3. Type of Tyre (Tipo de llanta)*

Otra variable a considerar es el tipo de llanta utilizada en una bicicleta. El rendimiento de una bicicleta varía a partir del tipo de llanta que se monta y la superficie en la que se utiliza. Es común observar bicicletas de ruta utilizando llantas lisas, sin embargo, existen llantas dentadas para superficies sueltas. Esta variable de entrada es crucial ya que en una llanta lisa se debe limitar la salida PWM del motor para evitar deslizamiento y además son susceptibles a punturas dependiendo de la superficie presente. Por el contrario, cuando se utiliza una llana dentada en una superficie suelta es posible aplicar una salida al motor de mayor magnitud ya que dichas llantas poseen mayor adherencia y generalmente son menos susceptibles a perforaciones. Se utilizó un rango de 0 a 1 donde 0 es *Smooth* y 1 es *Grooved.*

### 4. *Time of Day (Tiempo del día)*

Para fines de seguridad y protección del ciclista, el controlador capta como variable de entrada la iluminación del medio. Esta variable afecta el control de crucero ya que durante la noche el ciclista pierde visibilidad y es posible que no pueda corregir su curso en presencia de un obstáculo. El rango varía de 0 a 100, donde 0 representa un medio completamente oscuro y 100 representa un medio con iluminación máxima.

### 5. *Surface (Superficie)*

Finalmente, una variable importante a considerar para el rendimiento de la bicicleta es el tipo de superficie en la que opera. En esta variable se determinaron 3 tipos de superficie: *Gravel road*, *Tarmac street* y *Highway.* El rango varía de 0 a 100, donde 0 es una superficie de grava y 100 es una superficie pavimentada. Los valores intermedios corresponden a variaciones entre las superficies mencionadas.

La tabla 1 muestra las cinco variables de entrada del controlador crucero.

TABLA I
VARIABLES DE ENTRADA PARA CONTROL CRUCERO

| Entrada | Nombre | Valores |
|---|---|---|
| 1 | *Type of bike* | Road<br>Mountain |
| 2 | *Rider's weight* | Lightweight<br>Middleweight<br>Heavyweight |
| 3 | *Tyre* | Smooth<br>Grooved |
| 4 | *Time of day* | Daytime<br>Nighttime |
| 5 | *Surface* | Gravel road<br>Tarmac street<br>Highway |

## Señal de Salida

La señal de salida de control para el motor DC es una salida PWM con un rango que va de 0 a 255 que es la duración de pulso para el duty cycle de la tarjeta arduino. Se definen nueve diferentes pasos de salida, variando desde un valor mínimo llamado *Minor* a un valor máximo llamado *Ultra, e*stos valores de salida PWM se muestran en la tabla 2.

TABLA II
VARIABLES DE SALIDA PARA CONTROL CRUCERO

| Paso | Nombre |
|---|---|
| 1 | Minor |
| 2 | Very low |
| 3 | Low |
| 4 | Med-Low |
| 5 | Med |
| 6 | Med-High |
| 7 | High |
| 8 | Very high |
| 9 | Ultra |

A partir de las variables mencionadas se realiza la matriz de asociación difusa para determinar las reglas o etiquetas del control difuso. Es importante recordar que el control difuso depende de una base de datos para su funcionamiento, la cual es proporcionada por un experto a partir de estas reglas definidas por la matriz de asociación. Las reglas fueron dictadas con base en los factores y características de las variables de entrada descritas anteriormente. Todas las variantes que corresponden a las reglas establecidas se observan en la tabla 3.

TABLA III
MATRIZ DE ASOCIACIÓN CON REGLAS DEFINIDAS

| Type of Bike | Rider's weight | Type of tyre | Time of day | Surface | PWM |
|---|---|---|---|---|---|
| Road | Lightweight | Smooth | Daytime | Gravel road | Very low |
| | | | | Tarmac street | Low |
| | | | | Highway | Med-Low |
| | | | Nighttime | Gravel road | Minor |
| | | | | Tarmac street | Very low |
| | | | | Highway | Low |
| | | Grooved | | Gravel road | Low |
| | | | Daytime | Tarmac street | Med-Low |
| | | | | Highway | Med |
| | | | | Gravel road | Very low |
| | | | Nighttime | Tarmac street | Low |
| | | | | Highway | Med-Low |
| | | | | Gravel road | Low |
| | Middleweight | Smooth | Daytime | Tarmac street | Med-Low |
| | | | | Highway | Med |
| | | | | Gravel road | Very low |
| | | | Nighttime | Tarmac street | Low |
| | | | | Highway | Med-Low |
| | | | | Gravel road | Med-Low |
| | | Grooved | Daytime | Tarmac street | Med |
| | | | | Highway | Med-high |
| | | | | Gravel road | Low |
| | | | Nighttime | Tarmac street | Med-Low |
| | | | | Highway | Med |
| | | | | Gravel road | Med-Low |
| | Heavyweight | Smooth | Daytime | Tarmac street | Med |
| | | | | Highway | Med-high |
| | | | | Gravel road | Very low |
| | | | Nighttime | Tarmac street | Low |
| | | | | Highway | Med-Low |
| | | | | Gravel road | Med |
| | | Grooved | Daytime | Tarmac street | Med-high |
| | | | | Highway | High |
| | | | | Gravel road | Med-Low |
| | | | Nighttime | Tarmac street | Med |
| | | | | Highway | Med-high |
| | | | | Gravel road | Med-Low |
| Mountain | Lightweight | Smooth | Daytime | Tarmac street | Med |
| | | | | Highway | Med-High |
| | | | | Gravel road | Med |
| | | | Nighttime | Tarmac street | Med-High |
| | | | | Highway | High |
| | | | | Gravel road | Med-Low |
| | | Grooved | Daytime | Tarmac street | Med-High |
| | | | | Highway | High |
| | | | | Gravel road | Med-Low |
| | | | Nighttime | Tarmac street | Med |
| | | | | Highway | Med-High |
| | | | | Gravel road | Med |
| | Middleweight | Smooth | Daytime | Tarmac street | High |
| | | | | Highway | Very high |
| | | | | Gravel road | Med |
| | | | Nighttime | Tarmac street | Med-High |
| | | | | Highway | High |
| | | Grooved | Daytime | Gravel road | Med-High |
| | | | | Tarmac street | High |

| | | | | | |
|---|---|---|---|---|---|
| | | | | Highway | Very high |
| | | | | Gravel road | Med |
| | | | Nighttime | Tarmac street | Med-High |
| | | | | Highway | High |
| | | | | Gravel road | High |
| | | | Daytime | Tarmac street | Very high |
| | | Grooved | | Highway | Ultra |
| | | | | Gravel road | Med-High |
| | | | Nighttime | Tarmac street | High |
| | | | | Highway | Very high |

Una vez definida la matriz de asociación, la implementación en MATLAB-Simulink se llevó a cabo. El primer paso es utilizar el *Diseñador Lógico Difuso* incluido en la librería de MATLAB. En dicho diseñador, tanto las variables de entrada como de salida, la función de membresía y las reglas previamente definidas son determinadas y editadas para establecer el comportamiento de la salida PWM a partir de estas funciones. Dentro de este diseñador, cinco variables de entrada y una de salida deben ser definidas. La estructura fundamental de este primer controlador se observa en Fig. 6

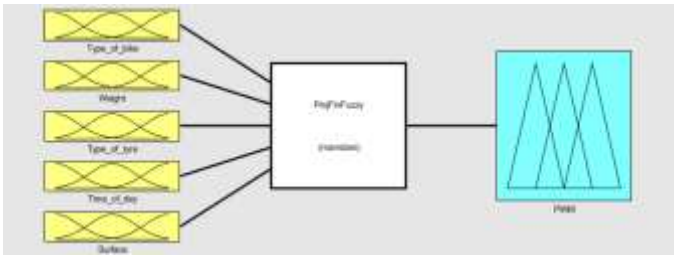

Fig. 6 Estructura de lógica difusa para control crucero

Con la estructura fundamental del controlador difuso, las funciones de membresía de cada variable fueron editadas. La variable de entrada *Type of bike*, corresponde a una función de membresía lineal tipo *trapmf*. La variable *Rider 's weight*, corresponde a las funciones de membresía para los diferentes tipos de pesos las cuales fueron de tipo *gaussmf* con distribuciones normales. Para la tercera variable *Type of tyre*, las funciones de membresía para ambos tipos de llanta fueron de tipo lineal *trapmf* donde existe una superposición entre *Smooth y Grooved tyres.* En la variable *Time of day*, la función de membresía determinada para ambos casos *Daytime* y *Nighttime* fue de tipo *gabellmf* para así poder aproximar las curvas de declinación solar de una forma más fidedigna. La función de membresía de la última variable de entrada *Surface*, corresponde a tipo *gabellmf*, la cual fue utilizada nuevamente para los tres casos correspondientes. Finalmente, para la variable de salida *PWM*, todos los casos fueron determinados con función de membresía lineal tipo *trimf*. [9]
Fig. 7 muestra las variables con sus funciones de membresía correspondientes.

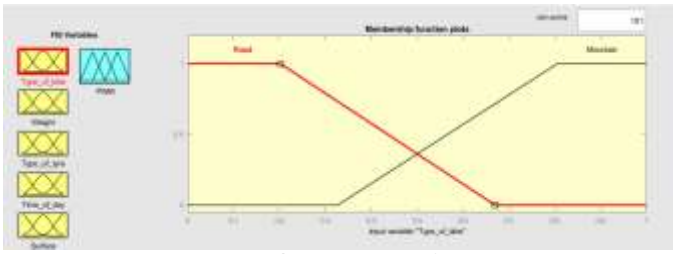

Fig. 7.1 Funciones de membresía para control difuso en MATLAB - control crucero para Type of bike

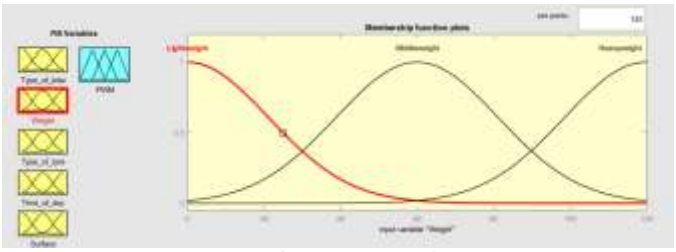

Fig. 7.2 Funciones de membresía para control difuso en MATLAB - control crucero para Rider's weight

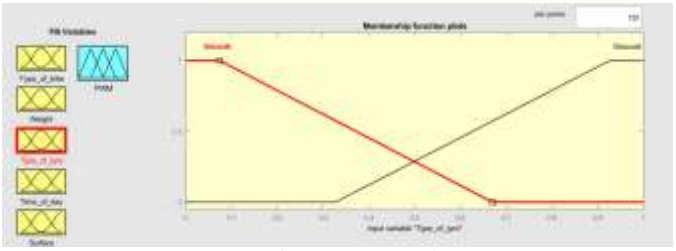

Fig. 7.3 Funciones de membresía para control difuso en MATLAB - control crucero Type of tyre

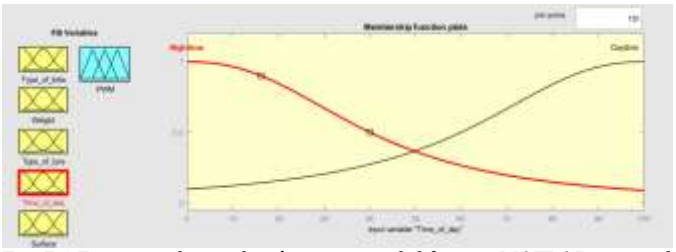

Fig. 7.4 Funciones de membresía para control difuso en MATLAB - control crucero para Time of the day

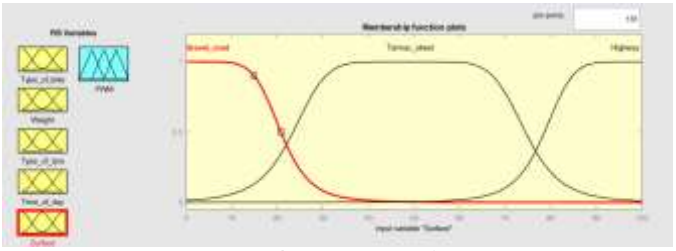

Fig. 7.5 Funciones de membresía para control difuso en MATLAB - control crucero para Surface

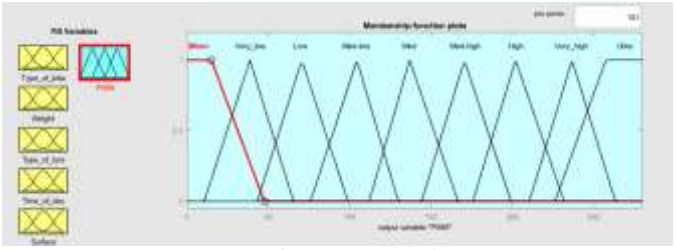

Fig. 7.6 Funciones de membresía para control difuso en MATLAB - control crucero para PWM

Con el control lógico difuso y sus funciones de membresía determinadas, la implementación en MATLAB-Simulink se realiza para aplicar el controlador a un motor DC real. Por otro lado, la

tarjeta Arduino y el Puente H fueron utilizados para acoplar el controlador con el sistema, los cuales se tomaron en consideración para el prototipo de controlador en Simulink. Cabe mencionar que para el diseño de este controlador el *Paquete para Hardware Arduino* fue instalado con el propósito de establecer comunicación entre el programa y la tarjeta. Dentro del programa se generan los siguientes bloques para definir el sentido de giro del motor a partir de las terminales *IN4* e *IN5* del Puente H y dos señales de alimentación, las cuales son salidas digitales de los puertos 7 y 8 de la tarjeta Arduino.

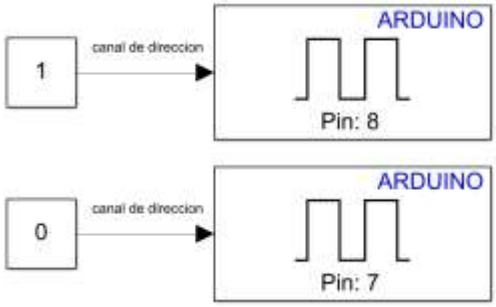


Fig. 8 Canal de dirección PWM

Para incorporar el controlador lógico difuso, cinco constantes de entrada deben ser proporcionadas al bloque *Fuzzy Logic Controller* a través de un *MUX*. La variable de salida, el bloque *PWM* de Arduino debe ser colocado a la salida del *Fuzzy Logic Controller* en el cual se debe determinar un puerto de salida digital *PWM* para la tarjeta física de Arduino En este caso, la salida PWM se asignó al puerto 6 en el que, adicionalmente, se incluyó un bloque de saturación previo para limitar el valor de PWM entre 0 y 255.

Fig. 9 muestra el prototipo de controlador crucero en Simulink en el cual se documentan los rangos de cada variable de entrada y donde es posible modificar dichas variables a través de *knobs*. Estos *knobs* representan valores dados por distintos tipos de sensores. El valor resultante del controlador se despliega en un *display* y un indicador en forma de tacómetro.

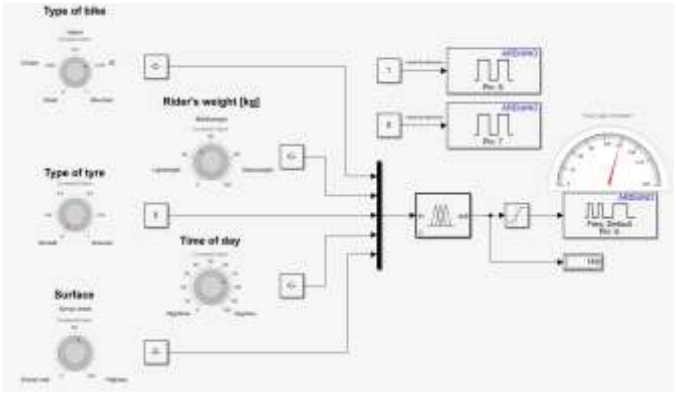


Fig. 9 Diagrama de control difuso, bicicleta crucero en MATLAB - Simulink

Para alimentar el motor de 3 volts, se utilizó una batería de 9 volts, mientras que, para la etapa de control, la alimentación se dió a través del cable de conexión de Arduino. Las conexiones fueron realizadas con base en el controlador diseñado en MATLAB-Simulink. Fig. 10 muestra el diagrama de conexión entre la tarjeta Arduino, el Puente H, la batería y el motor DC.

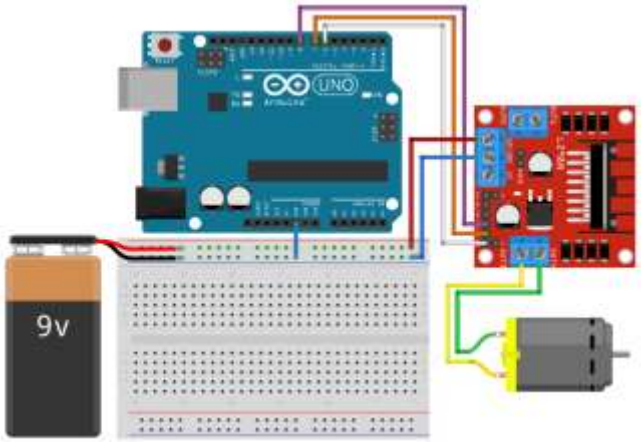


Fig. 10 Diagrama de conexiones para control crucero. [2]

Fig. 11 muestra el ensamble físico del diagrama esquemático anterior.

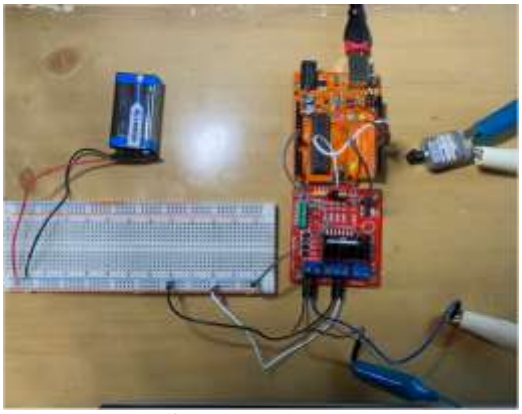

Fig. 11 Diagrama físico de conexiones para control crucero

Una vez ensamblado el circuito, se plantearon los siguientes casos para observar el comportamiento del control crucero.

Para darle un mejor enfoque al análisis del trabajo y demostrar su funcionamiento, se establecieron diferentes escenarios independientes a cada controlador. Cada uno de los escenarios ejemplifica la funcionalidad real de la bicicleta inteligente en sus diferentes casos o variables de entrada. Es importante señalar que los escenarios son ficticios y únicamente son utilizados con fines explicativos.

*Caso 1*

*Debido a la poca preparación de los competidores por la pronta recuperación de la pandemia en el tour de Francia, cada participante contará con una bicicleta con motor que le ayude a soportar todo el recorrido. Los argumentos de la carrera están dirigidos a la incorporación de nuevas tecnologías. No existen limitaciones con respecto a la incorporación de controladores en la carrera, por lo que uno de los competidores usará un controlador fuzzy, en caso de que éste requiera descansar un poco y pueda ser utilizado en diferentes escenarios.*

*Al ser una carrera, el tipo de bicicleta buscará ser la más ligera y tratará de exigirle menos consumo al motor, por lo que se*

*utilizará el tipo de bicicleta "Road bike" y un tipo de llanta que sea intermedia entre lisa y rugosa, debido al tipo de superficies que se pueden encontrar. Se espera que el consumo en este caso sea intermedio referente al tipo de llanta.*

*Empieza la carrera. El motor se encuentra encendido. Una vez que el participante se sube a la misma, el motor comienza a exigir mayor consumo hasta llegar al peso deseado con la intención de ayudar al competidor de no cansarse tanto. Conforme pasa la carrera, el pavimento comienza a hacerse más rugoso y peligroso, por lo que se espera que el motor disminuya su consumo y no dañe el neumático. Empieza a anochecer y cada vez se ve menos. El competidor ya está cansado y busca usar su motor para ir más rápido, sin embargo, el controlador disminuye el rendimiento del motor para tratar de evitar accidentes por la pérdida de luz. El competidor llega al final y se sube de la bicicleta haciendo que el motor ya no consuma la misma energía.*

*Caso 2*

*Para el segundo escenario, el competidor hizo lo mismo, pero en su bicicleta de montaña y la usará para probarla. Al ser una bicicleta de montaña, se espera que el motor consuma mayor energía, de esa manera subirá sin problema cualquier pendiente o cada piedra y rama que se encuentre en el camino.*

*En este caso, el tipo de rueda será menos lisa, ya que necesitará mayor adherencia en superficies rugosas. El conductor está por comenzar la ruta, por lo que enciende el motor y pronto comienza a generar mayor consumo por el peso del mismo. Conforme va por el camino, el conductor pasa por diferentes superficies que hacen que el motor se encuentre en constante cambio de consumo hasta estabilizarse en un nivel medio. Al final, se empieza a oscurecer, y el motor comienza a bajar su energía para evitar cualquier tipo de accidente.*

Los resultados obtenidos para estos dos escenarios se discuten en el análisis de resultados.

Adicional a la implementación de este controlador en MATLAB-Simulink, se realizó la simulación del mismo controlador en el programa LabVIEW de National Instruments. Una de las ventajas de realizar la simulación en este software es que el panel de control se puede visualizar de una manera más interactiva. Gracias a la estructura de diagrama de bloques, es posible manipular las variables de una forma más intuitiva y representativa.

Se utilizó el *Diseñador Lógico Difuso* de LabVIEW de manera similar al de MATLAB por lo que el proceso consta de los mismos pasos: definir entradas y salidas, determinar las funciones de membresía para cada variable y establecer las reglas o etiquetas que dictan el comportamiento de la salida PWM.

Fig.12 muestra las funciones de membresía de las variables de entrada y variables de salida de este controlador, dichas funciones son similares a las creadas en MATLAB-Simulink. Las reglas establecidas son las mismas observadas en la matriz de asociación difusa mostrada en la tabla 3.

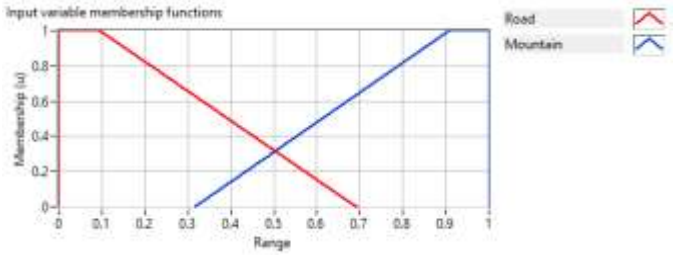


Fig. 12.1 Funciones de membresía para la variable Type of *bike* en LabVIEW control crucero

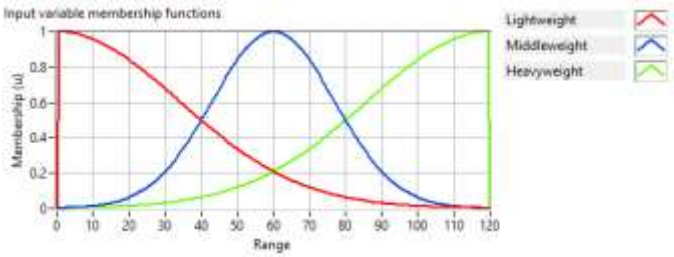


Fig. 12.2 Funciones de membresía para la variable Rider's weight en LabVIEW control crucero

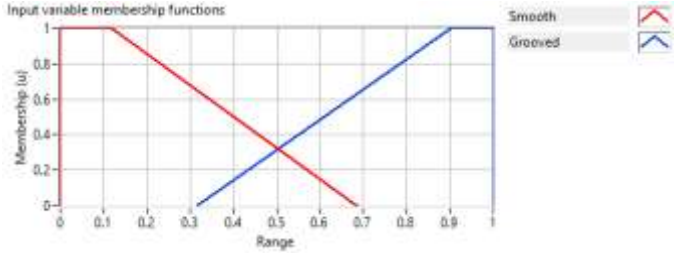


Fig. 12.3 Funciones de membresía para la variable Type of tyre en LabVIEW control crucero

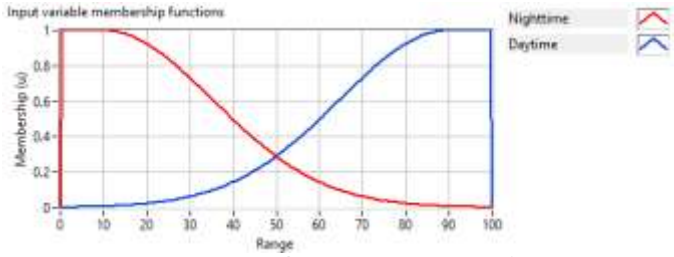


Fig. 12.4 Funciones de membresía para la variable Time of day en LabVIEW control crucero

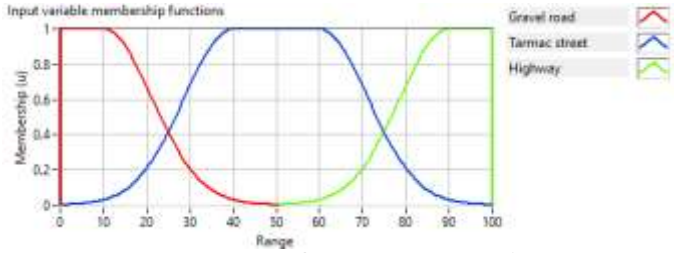


Fig. 12.5 Funciones de membresía para la variable Surface en LabVIEW control crucero

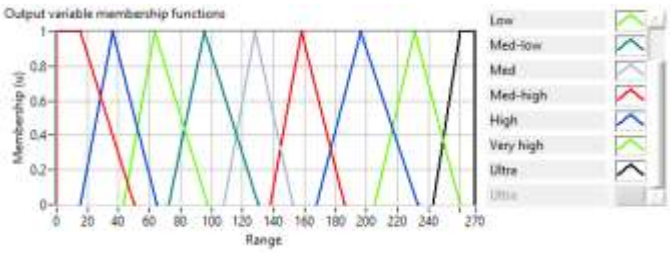


Fig. 12.6 Funciones de membresía para la variable PWM en LabVIEW control crucero

Con las funciones de membresía definidas y todas las reglas determinadas a partir de la matriz de asociación, se exportaron las reglas de lógica difusa a un archivo independiente para ser usado en el diagrama de bloques de LabView. El diagrama de este controlador es similar a la estructura de control de Simulink, sin embargo, en LabVIEW es necesario comenzar con un comando *While* para mantener el programa corriendo.

Los bloques principales que se usan son: *Fuzzy Controller (MISO)* que proporciona múltiples entradas y salidas, *Load Fuzzy System* que permite importar las reglas difusas previamente creadas, *File Path* que establece el directorio del archivo fussy y un *Build Array* que actúa como un MUX de MATLAB, sirve para introducir todas las variables de entrada al *Fuzzy System*.

Con los bloques anteriores se construye el diagrama del controlador utilizando una estructura *Case* para incorporar un sistema de encendido de motor y controlador como se muestra en Fig. 13.

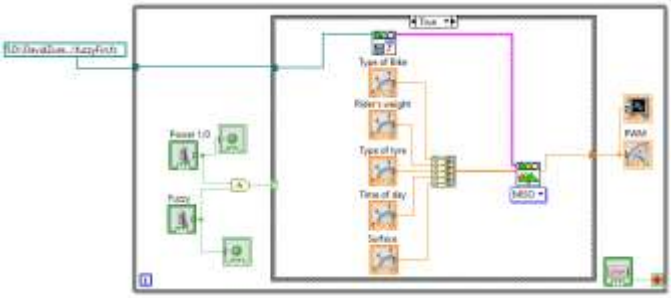


Fig. 13 Diagrama de bloques control lógico difuso, control crucero en LabVIEW

Finalmente, un panel de control se construye para simular la variación de entradas del sistema y su reflejo en el comportamiento de la salida PWM.

Fig. 14 muestra dicho panel de control en el cual se pueden observar los controladores de variables de entrada y el sistema de prendido de motor y controlador fuzzy.

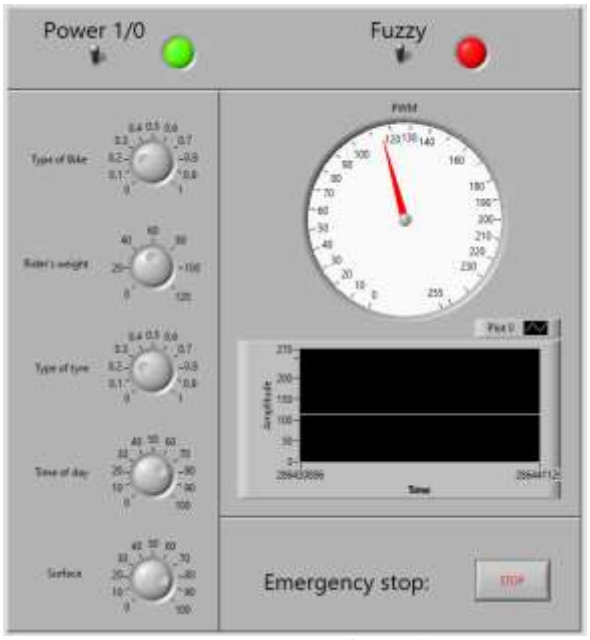


Fig. 14 Panel de control para controlador lógico difuso, control crucero en LabVIEW

### *A. Controlador con Múltiples Variables de Entrada y Salida*

Se planeó y diseñó un segundo controlador crucero con múltiples variables de entrada y salida. Se busca que este asista al usuario al modificar la velocidad del motor de la bicicleta inteligente acoplada a la transmisión de la misma, además de proveer un segundo control sobre las luces traseras y delanteras de la bicicleta para brindar mayor seguridad durante todo el día, haciendo que la luz sea tenue durante el día y más brillosa a medida que oscurece en el entorno. En este controlador se toman en cuenta cuatro variables de entrada que se consideran importantes al momento de manejar una bicicleta, por otro lado, el controlador tiene como salida dos variables, siendo una señal PWM para el motor DC del vehículo y una segunda señal PWM para modificar la intensidad de las luces LED acopladas a la bicicleta. A continuación, se presentan las variables antes mencionadas y sus características.

Señales de Entrada

#### *1. Rugosity Surface (Rugosidad de superficie)*

Se tomó en cuenta un amplio rango de rugosidad para distintos tipos de terreno que van desde césped hasta pavimento, contando con un rango medible de un milímetro hasta un metro de rugosidad, esto con el propósito de permitir al vehículo una conducción asistida en distintos tipos de terreno. La variable de rugosidad se divide en *High, Medium y Low rugosity.* [10]

#### *2. Rider's Weight (Peso del conductor)*

Así como en el primer controlador, se toma como variable la masa del usuario, para que el motor controlado pueda satisfacer la tracción al momento en que se desee avanzar de cero, esta variable tiene un gran impacto en el rendimiento del vehículo.

El rango empleado va desde los 20 hasta los 100 kilogramos diseñado de tal forma que la bicicleta inteligente sea

operada desde niños con edad de 6 años con un peso promedio de 20 kg hasta adultos que no excedan con el límite máximo del rango especificado, el peso se divide en la matriz de asociación como *Slim, Medium y Weight.*

### 3. *Inclination Angle (Ángulo de inclinación)*

Otra variable importante a considerar es el ángulo de inclinación al que está sometido el sistema. No se tomaron en cuenta ángulos negativos, sin embargo, pueden ser implementados posteriormente. En cambio, se utilizan ángulos que parten desde 0° (superficie horizontal) hasta una inclinación de 45°, dividiéndose en la matriz como inclinación *Low, Medium y High.*

### 4. *Luminous Intensity (Intensidad lumínica)*

Para la última variable de entrada ligada al control de la señal PWM de las luces LED, se diseña un rango de nivel de iluminación que abarca desde el cielo nocturno iluminado únicamente por la luna en una noche despejada hasta la luz solar a mediodía con un cielo despejado. Se busca ampliar la seguridad del usuario y al mismo tiempo prolongar la duración de la batería al utilizar la potencia necesaria para iluminar adecuadamente el camino a lo largo del transcurso del día.

El rango medible utilizado en esta variable va desde los 0 luxes hasta los 120000, comprendiendo una división en la matriz de asociación de *Dark, Opaque y Shiny* para denotar los tres distintos grados de iluminación.

Con el propósito de visualizar las variables antes mencionadas con sus respectivos valores, se muestra la tabla 4 con sus características fundamentales.

TABLA IV
VARIABLES DE ENTRADA PARA CONTROL MULTIVARIABLE

| Entrada | Nombre | Valores |
|---|---|---|
| 1 | *Rugosity surface* | High rugosity<br>Medium rugosity<br>Low rugosity |
| 2 | *Rider's weight* | Weight<br>Medium<br>Slim |
| 3 | *Inclination angle* | High angle<br>Medium angle<br>Low angle |
| 4 | *Luminous intensity* | Shiny<br>Opaque<br>Dark |

## Señales de Salida

Las señales de salida para este controlador son dos PWM con un rango que va de 0 a 255. Una para el control de la velocidad del motor de 3 volts DC y otra para el control de la intensidad lumínica de las luces LED de la bicicleta.

Para el motor, se tienen cinco pasos de salida que varían desde el valor mínimo llamado *Low* hasta el máximo nombrado como *High*, mientras que la variable de salida que regula la intensidad lumínica, sólo cuenta con tres pasos de salida: *Dark, Dim y Bright.*

En las tablas 5 y 6 se ejemplifican las variables de salida con sus respectivas variaciones.

TABLA V
VARIABLE DE SALIDA VELOCIDAD PARA CONTROL MULTIVARIABLE

| Paso | Nombre |
|---|---|
| 1 | Low |
| 2 | Med-Low |
| 3 | Med |
| 4 | Med-High |
| 5 | High |

TABLA VI
VARIABLE DE SALIDA INTENSIDAD LUMÍNICA PARA CONTROL MULTIVARIABLE

| Paso | Nombre |
|---|---|
| 1 | Bright |
| 2 | Dim |
| 3 | Dark |

Una vez establecidas las variables de entrada y salida, se crea una matriz de asociación difusa de la que se extrajeron las reglas del controlador basado en lógica difusa. La matriz fue construida directamente desde la herramienta Excel y se presenta a continuación en la tabla 7.

TABLA VII
MATRIZ DE ASOCIACIÓN DIFUSA CONTROL MULTIVARIABLE

| Rugosity Surface | Rider's Weight | Luminous Intensity | Inclination Angle | Velocity | Light |
|---|---|---|---|---|---|
| | | | Low | High | Dark |
| | | Shiny | Medium | High | Dark |
| | | | High | Med-High | Dark |
| | | | Low | High | Dim |
| | Weight | Opaque | Medium | High | Dim |
| | | | High | Med-High | Dim |
| | | | Low | High | Bright |
| | | Dark | Medium | High | Bright |
| | | | High | Med-High | Bright |
| | | | Low | High | Dark |
| | | Shiny | Medium | Med-High | Dark |
| | | | High | Med | Dark |
| | | | Low | High | Dim |
| High | Medium | Opaque | Medium | Med-High | Dim |
| | | | High | Med | Dim |
| | | | Low | High | Bright |
| | | Dark | Medium | Med-High | Bright |
| | | | High | Med | Bright |
| | | | Low | Med | Dark |
| | | Shiny | Medium | Med-Low | Dark |
| | | | High | Low | Dark |
| | | | Low | Med | Dim |
| | Slim | Opaque | Medium | Med-Low | Dim |
| | | | High | Low | Dim |
| | | | Low | Med | Bright |
| | | Dark | Medium | Med-Low | Bright |
| | | | High | Low | Bright |
| | | | Low | High | Dark |
| | | Shiny | Medium | Med-high | Dark |
| | | | High | Med | Dark |
| | | | Low | High | Dim |
| | Weight | Opaque | Medium | Med-high | Dim |
| | | | High | Med | Dim |
| | | | Low | High | Bright |
| | | Dark | Medium | Med-high | Bright |
| | | | High | Med | Bright |
| | | | Low | Med-High | Dark |
| | | Shiny | Medium | Med | Dark |
| | | | High | Med-Low | Dark |
| | | | Low | Med-High | Dim |
| Medium | Medium | Opaque | Medium | Med | Dim |
| | | | High | Med-Low | Dim |
| | | | Low | Med-High | Bright |
| | | Dark | Medium | Med | Bright |
| | | | High | Med-Low | Bright |
| | | | Low | Med-low | Dark |
| | | Shiny | Medium | Med | Dark |
| | | | High | Low | Dark |
| | | | Low | Med-low | Dim |
| | Slim | Opaque | Medium | Med | Dim |
| | | | High | Low | Dim |
| | | | Low | Med-low | Bright |
| | | Dark | Medium | Med | Bright |
| | | | High | Low | Bright |
| | | | Low | Med-High | Dark |
| | | Shiny | Medium | Med | Dark |
| | | | High | Med | Dark |
| | | | Low | Med-High | Dim |
| | Weight | Opaque | Medium | Med | Dim |
| Low | | | High | Med | Dim |
| | | | Low | Med-High | Bright |
| | | Dark | Medium | Med | Bright |
| | | | High | Med | Bright |
| | Medium | Shiny | Low | Med | Dark |

| | | | | | |
|---|---|---|---|---|---|
| | | | Medium | Med-low | Dark |
| | | | High | Med-low | Dark |
| | | | Low | Med | Dim |
| | | Opaque | Medium | Med-low | Dim |
| | | | High | Med-low | Dim |
| | | | Low | Med | Bright |
| | | Dark | Medium | Med-low | Bright |
| | | | High | Med-low | Bright |
| | | | Low | Med | Dark |
| | | Shiny | Medium | Med-Low | Dark |
| | | | High | Low | Dark |
| | | | Low | Med | Dim |
| | Slim | Opaque | Medium | Med-Low | Dim |
| | | | High | Low | Dim |
| | | | Low | Med | Bright |
| | | Dark | Medium | Med-Low | Bright |
| | | | High | Low | Bright |

Fig. 15 muestra la estructura fundamental del segundo controlador implementado en MATLAB-Simulink.

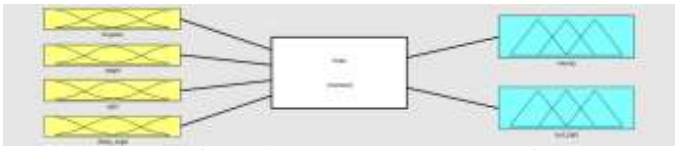

Fig. 15 Estructura de lógica difusa para control crucero de múltiples variables de salida

A partir de la estructura fundamental del segundo controlador, se configuran todas las funciones de membresía de las distintas variables. Al inicio y al final de los rangos de las cuatro variables de entrada, se utilizan funciones lineales trapezoidales de tipo *trapmf*. En el rango intermedio se utiliza una función de membresía triangular de tipo *trimf* donde existe una superposición entre la parte intermedia para considerar que la variable se puede localizar entre esas divisiones del rango. En cuanto a las variables de salida, al inicio y final del rango se diseñaron con funciones de tipo *trapmf* y en los puntos medios con funciones triangulares tipo *trimf.*

Fig. 16 muestra las funciones de membresía de las variables de entrada y salida de este controlador.

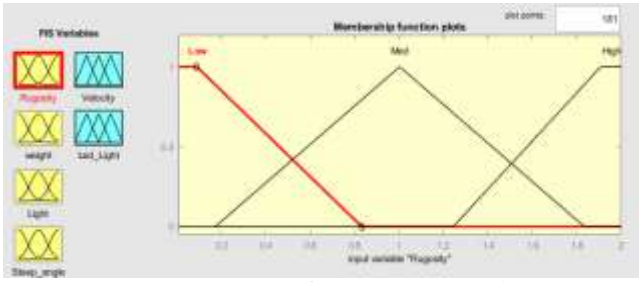


Fig. 16.1 Funciones de membresía para Rugosity surface en MATLAB, control de múltiples variables de entrada y salida

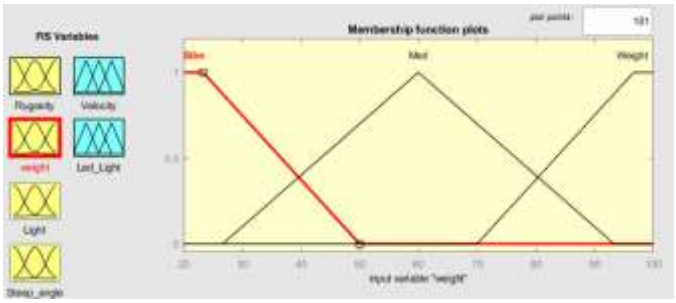


Fig. 16.2 Funciones de membresía para Rider's weight en MATLAB, control de múltiples variables de entrada y salida

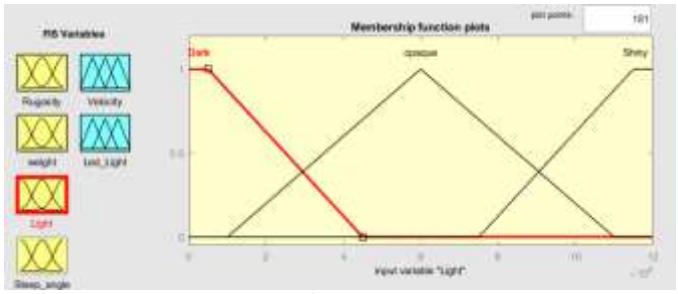


Fig. 16.3 Funciones de membresía para Inclination angle en MATLAB, control de múltiples variables de entrada y salida

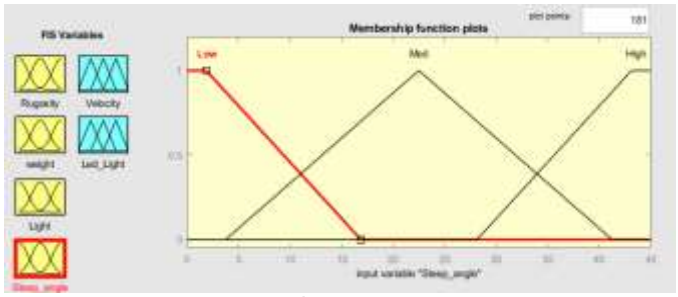


Fig. 16.4 Funciones de membresía para Luminous intensity en MATLAB, control de múltiples variables de entrada y salida

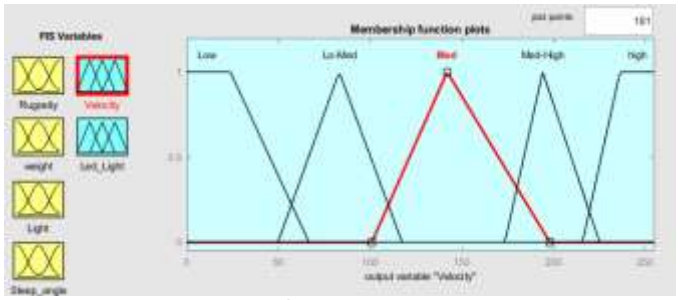


Fig. 16.5 Funciones de membresía para PWM motor en MATLAB, control de múltiples variables de entrada y salida

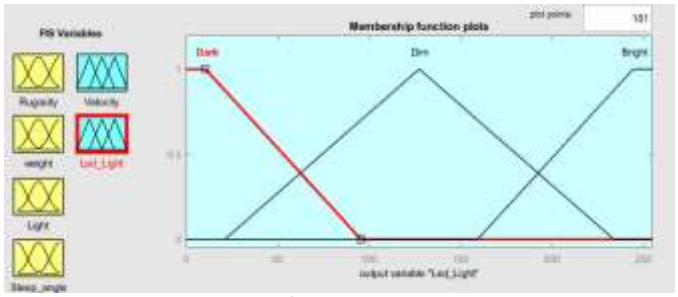


Fig. 16.6 Funciones de membresía para PWM LED en MATLAB, control de múltiples variables de entrada y salida

Se utilizan los mismos componentes para llevar a cabo la implementación del control de velocidad que en el control crucero, al igual que los pines utilizados en el Puente H y las señales de dirección de giro del motor, únicamente se utilizó un programa distinto que se explicará a continuación.

Se necesitan cuatro constantes de entradas cuya función es proporcionar los valores a las distintas variables de entrada del bloque *Fuzzy Logic Controller*, el orden de las cuatro variables de forma descendente es la variable *Rugosity Surface*, seguido por *Rider's weight*, *Luminous intensity* y por último *Inclination angle.*

Después de la salida del bloque de control, se coloca un bloque de saturación para filtrar los valores que no estén dentro del rango indicado entre 0 y 255. Por último, se colocan dos puertos de salida digital PWM dirigidos a la tarjeta física de Arduino. El primero que se muestra en orden descendente es la señal PWM de la velocidad que está conectado al Pin 11 mientras que el segundo corresponde a la señal PWM del LED conectada al Pin 12 de la tarjeta Arduino.

Fig. 17 muestra el prototipo de simulación para un controlador fuzzy de múltiples variables de entrada y salida de la bicicleta inteligente.

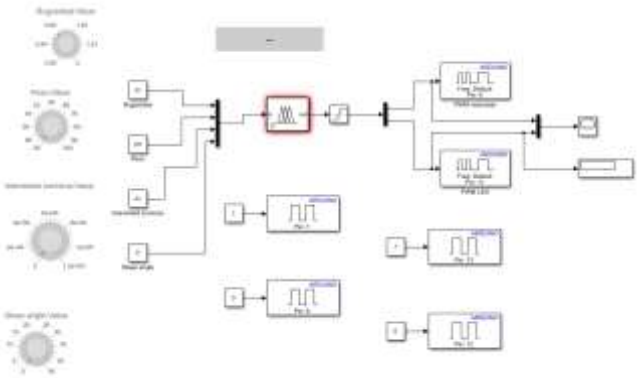

Fig. 17 Diagrama de control difuso, bicicleta con control multivariables en MATLAB-Simulink

Se realizan las conexiones físicas en un motor de 3 volts DC alimentado por una batería de 9 volts a través de la tarjeta Arduino y Puente H (L298N). Las conexiones del circuito se muestran en Fig. 18. Se añadió un foco LED para la variable de salida de luz intermitente de la bicicleta, se conecta como si fuera un segundo motor en el puente H, de esta forma al momento de cambiar la polaridad en Matlab Simulink la luz empezará a parpadear.

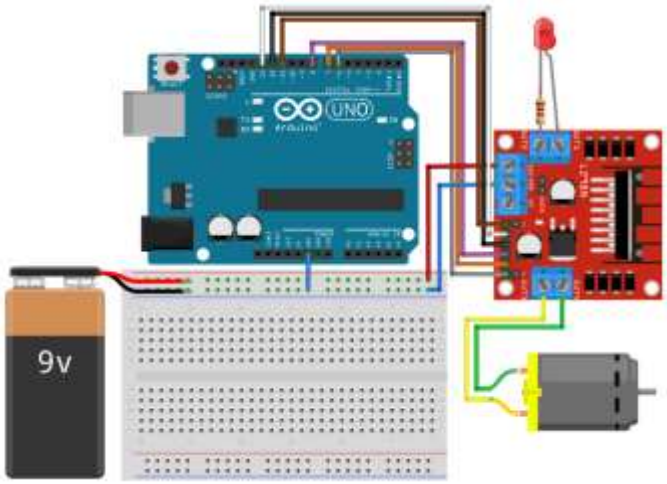


Fig. 18 Diagrama de conexiones para control multivariables [2]

El circuito ensamblado físicamente tanto del motor como de la luz LED se muestran a continuación en la Fig. 19.

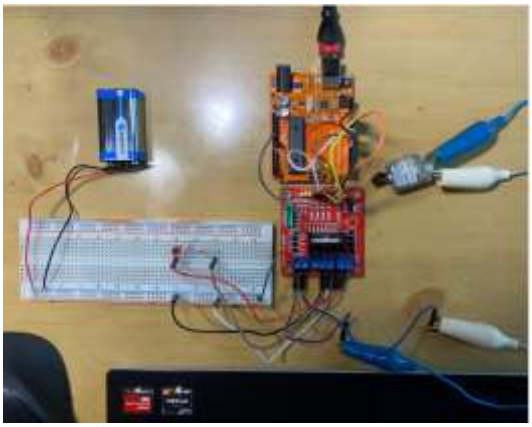

Fig. 19 Diagrama físico de conexiones para control multivariables

Los casos correspondientes a cada una de las variables son los siguientes:

*Caso 1*

*Debido a un aumento de carga el motor consume más energía por la necesidad de aumentar el torque para llegar a una velocidad deseada.*

*Caso 2*

*Se conserva la misma masa y el camino se pone más rugoso disminuye la velocidad del motor para evitar deslizamiento.*

*Caso 3*

*Si la bicicleta empieza a irse por una pendiente y comienza a aumentar su inclinación, el motor necesitará mayor torque, por lo tanto, mayor consumo de energía.*

*Caso 4*

*Si cada vez hay más luz solar, la intensidad del LED comienza a disminuir. Conforme pasa el día y se va oscureciendo, comienza a aumentar la luminosidad del mismo.*

Adicional a la implementación en el software MATLAB-Simulink, se llevó a cabo la simulación en el software LabVIEW de National Instruments.

Se utiliza de nueva cuenta el *Diseñador Lógico Difuso*, definiendo las entradas y salidas, así como las funciones de membresía para cada variable estableciendo todas las reglas de la matriz de asociación difusa del controlador.

Fig. 20 muestra las funciones de membresía de las variables de entrada y salida. Se usaron las mismas funciones de MATLAB.

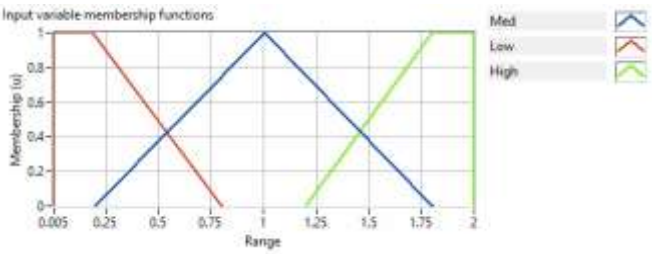

Fig. 20.1 Funciones de membresía para Rugosity surface en LabVIEW, control multivariable

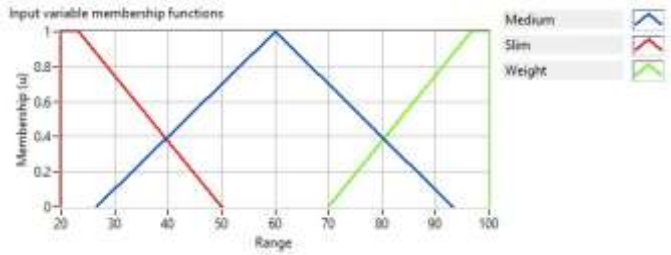


Fig. 20.2 Funciones de membresía para Rider's weight en LabVIEW, control multivariable

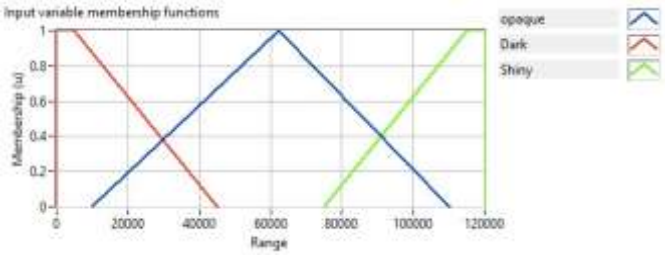


Fig. 20.3 Funciones de membresía para Luminous intensity en LabVIEW, control multivariable

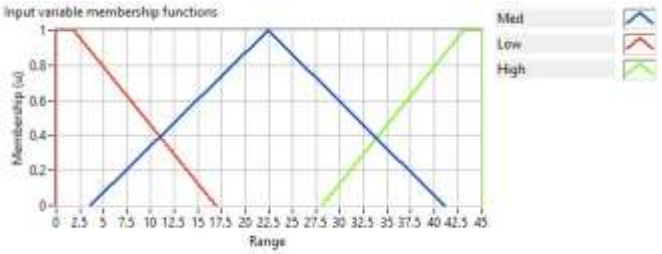


Fig. 20.4 Funciones de membresía para Inclination angle en LabVIEW, control multivariable

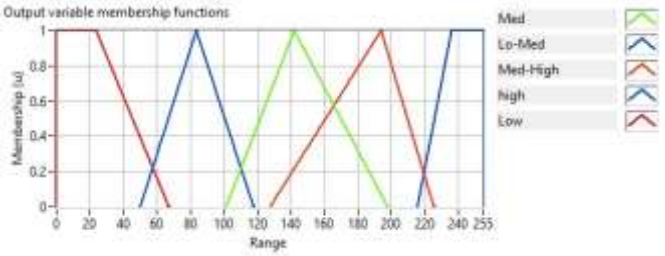


Fig. 20.5 Funciones de membresía para PWM motor en LabVIEW, control multivariable

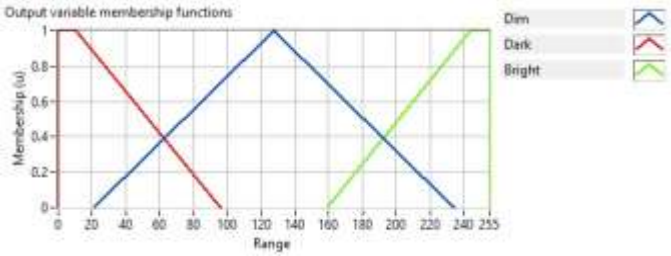


Fig. 20.6 Funciones de membresía para PWM LED en LabVIEW, control multivariable

Se introducen a detalle cada una de las funciones para que coincidan con los valores y rangos exactos utilizados en la lógica difusa de MATLAB. Una vez establecidas todas las funciones de membresía de las variables de entrada y salida, se exporta el control lógico difuso para su uso en el diagrama de bloques de LabVIEW.

Como se mencionó anteriormente, el diagrama y estructura de bloques es muy similar al de MATLAB. Primero se comienza con una estructura *While* para mantener al programa corriendo hasta que se ejecute un comando de *Stop*.

Se utilizan los bloques: *Fuzzy Controller (MISO)* que permite introducir múltiples entradas y salidas, *Load Fuzzy* para importar el programa de lógica difusa donde se establecieron todas las reglas, así como funciones de membresía, se agregó un *File Path* para localizar el archivo de lógica difusa, por último, se utilizó un *Build Array* que permite introducir todas las variables de entrada al *Fuzzy System*. El diagrama de programación implementado en LabVIEW se observa en Fig. 21.

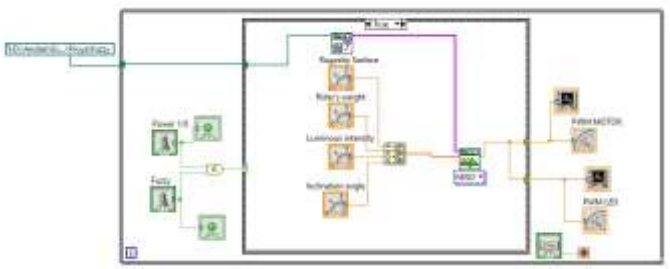

Fig. 21 Diagrama de bloques control lógico difuso, control multivariables en LabVIEW

Fig. 22 muestra un panel de control que representa el sistema con el controlador. Se observan dos indicadores tipo tacómetro, el del lado izquierdo indica el comportamiento de la velocidad mientras que el de la derecha el comportamiento de la luz LED del sistema. También muestra las variables de entrada en forma de *Knobs* para introducir los valores de entrada al controlador y dos indicadores que representan los valores de salida con respecto al tiempo.

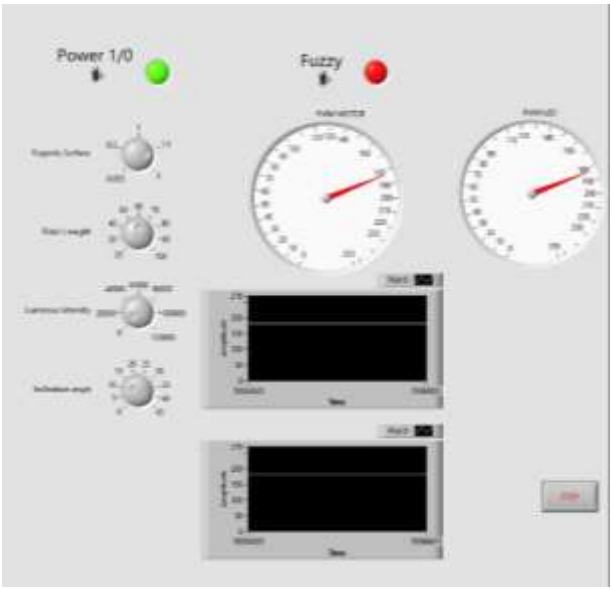


Fig. 22 Panel de control para controlador lógico difuso, control multivariables en LabVIEW

## IV. Análisis, Interpretación y Discusión de Resultados

Para el *Caso 1* del primer controlador, se considera un tipo de bicicleta *Road bike* y un tipo de llanta dentro de un balance entre lisa y rugosa (entre *Smooth* y *Grooved*); esto permite pronosticar un consumo medio por parte del motor. Por otro lado, el peso ligero del ciclista (*Lightweight*) no interfiere de gran manera con el consumo del motor.

Considerando que comienza a anochecer, es de esperarse que la velocidad del motor disminuya, con el fin de reducir la velocidad de la bicicleta y evitar algún accidente.

Además, se contempla que la superficie por la que se viaja, pase de ser pavimento firme a un terreno irregular de grava, ocasionando que el motor disminuya la velocidad dadas las condiciones del terreno y también para evitar algún accidente.

Para visualizar de mejor forma los resultados mencionados, se presentan en la tabla 8.

Tabla VIII
Resultados de las Variables Consideradas en el Caso 1

| **Tipo de entradas** | **Valor inicial** | **Valor cambiado** | **Salidas de matriz difusa** | **Resultados físicos obtenidos** |
|---|---|---|---|---|
| Tipo de bicicleta | Road bike | - | Motor más lento | Motor más lento |
| Peso del conductor | Light weight | Lightweight & Heavyweight | Motor más rápido | Motor más rápido |
| Tipo de llanta | Smooth & Grooved | - | Rango intermedio | Rango intermedio |
| Tiempo de día | Daytime | Nighttime | Motor más lento | Motor más lento |
| Superficie | Highway | Gravel road | Motor más lento | Motor más lento |

Se observa de la tabla 8 que las salidas planteadas de la matriz difusa coinciden con los resultados físicos obtenidos de la implementación del motor de 3 volts DC.

Para el *Caso 2* del primer controlador, se considera que el tipo de bicicleta cambia a *Mountain bike*, lo que se traduce en aumento de velocidad del motor.

Para el tipo de llanta, se considera que conforme avanzara, cambiaría de un tipo lisa (*Smooth*) a rugosa (*Grooved*), lo que implica una disminución en la velocidad del motor.

Además, la intensidad lumínica aumenta conforme al paso del tiempo provocando que el motor aumente su velocidad ya que los riesgos por colisión y derrapamiento disminuyen considerablemente durante el día (o con mejor iluminación).

Los únicos parámetros que se mantuvieron respecto al caso anterior, fueron el peso ligero del usuario y la superficie sobre la que andaría el ciclista.

Tabla IX
Resultados de las Variables Consideradas en el Caso 2

| **Tipo de entradas** | **Valor inicial** | **Valor cambiado** | **Salidas de matriz difusa** | **Resultados físicos obtenidos** |
|---|---|---|---|---|
| Tipo de bicicleta | Mountain bike | - | Motor más rápido | Motor más rápido |
| Peso del conductor | Lightweight | Lightweight & Heavy weight | Motor más rápido | Motor más rápido |
| Tipo de llanta | Smooth | Grooved | Motor más lento | Motor más lento |
| Tiempo de día | Night time | Daytime | Motor más rápido | Motor más rápido |
| Superficie | Highway | Gravel road | Motor más lento | Motor más lento |

Nuevamente, se observa de la tabla 9 que el comportamiento físico del motor de 3 volts DC es gobernado por las reglas establecidas en la matriz de asociación difusa, lo que implica el funcionamiento óptimo del controlador el cual se ajusta a los distintos escenarios en los que se desempeña la bicicleta.

Para el análisis del segundo controlador, es decir, el multivariable, se extrajo la siguiente información a partir de las pruebas realizadas de forma física.

Tabla X
Resultados de los Diferentes Casos del Controlador 2

| **Casos** | **Variable** | **Argumento** | **Salidas Esperadas por matriz difusa** | **Resultados físicos obtenidos** |
|---|---|---|---|---|
| Caso 1 | Rider's weight | Aumenta | Motor más rápido | Motor más rápido |
| Caso 2 | Rugosity surface | Aumenta | Motor más lento | Motor más lento |
| Caso 3 | Inclination angle | Aumenta | Motor más rápido | Motor más rápido |
| Caso 4 | Luminous | Aumenta | Led pierde | Led pierde |

| | | | | |
|---|---|---|---|---|
| | intensity | | luz | luz |

Como se puede observar de la tabla 10, en el primer caso, la variable *Rider's weight* aumenta cuando el conductor se sube a la bicicleta, ambos resultados muestran que el motor comienza a consumir mayor energía para así poder avanzar más rápido.

En el segundo caso, el pavimento comienza a hacerse más rugoso, y, en consecuencia, el motor empieza a alentarse, tal comportamiento se comprueba en la implementación física.

Para el caso 3, mientras aumenta el ángulo de la superficie asumiendo lugares montañosos o altas pendientes, el motor tiende a comportarse más rápido debido a que requiere mayor torque y en consecuencia mayor consumo de energía a pesar de que la bicicleta podría ir más lenta.

El caso 4 presenta un comportamiento similar al del primer controlador, con la excepción que el controlador multivariable tiene un LED que simula una lámpara que se prenderá y apagará con el pasar del día y dependiendo de la luz solar.

La siguiente liga muestra un video de la implementación de ambos controladores tanto para el primer escenario del primer controlador como los diferentes casos del segundo controlador: https://www.youtube.com/watch?v=jzDKi79Eu8I&feature=youtu.be&ab_channel=DanielSegura

## V. Conclusiones

La lógica difusa permite un manejo amplio de variables en comparación de un controlador de tipo PID, su versatilidad concede un cambio sencillo de las reglas de la matriz de asociación difusa para su diseño y operación, a diferencia del PID que requiere del modelo matemático y en los casos presentados resultaría sumamente complejo.

La aplicación de dos controladores basados en lógica difusa de este trabajo para una bicicleta inteligente, refleja de forma exitosa diferentes variables que se desean controlar.

En el primer controlador, las respuestas obtenidas de la implementación física, fueron las mismas que se establecieron en la programación de la lógica difusa, por lo que se concluye que el controlador es congruente en su funcionamiento. También se demuestra con los resultados del controlador multivariable que tener distintas salidas u objetos a controlar, no afecta el desempeño del controlador, ambos satisfacen las necesidades propuestas en cada uno de los casos.

Gracias a la implementación de un controlador difuso, el motor de 3 volts y luces LED garantizan mayor seguridad al ciclista y a su vez un mayor rendimiento.

## Referencias


[1] The Arduino website. [Online]. Available: https://www.arduino.cc/en/Main/Create

[2] The Fritzing website. [Online]. Available: https://forum.fritzing.org/t/h-bridge-with-l298n-motor-driver/7711/2

[3] J. L. Barahona-Avalos. (2015) *Control de velocidad de un motor de CD con conexión en serie mediante rechazo activo de perturbaciones.* [Online]. Available: https://www.researchgate.net/figure/Circuito-electrico-equivalente-del-motor-de-CD-en-conexion-en-serie-Un-sistema-no-lineal_fig1_282862157

[4] S. J. Chapman, *Motores y generadores de corriente directa*. 5th. ed. Boston: McGraw-Hill, 2012. p.345

[5] P. Navarro, J. R. Wamba, A. Fernández , O. Altisench, C. Carcía, J. Julia, M. Rui, *La ingeniería de la bicicleta.* Madrid España: Fundación ESTEYCO, 2010.

[6] P. Gianfranco. (2014) *Lithium-Ion Batteries Advances and Applications* . Elsevier. [Online]. Available: https://app.knovel.com/hotlink/toc/id:kpLIBAA003/lithium-ion-batteries/lithium-ion-batteries

[7] I. H. Altas, *Fuzzy Logic Control in Energy Systems - With Design Applications in MATLAB/Simulink*. Institution of Engineering and Technology, 2017. pp. 3-4.

[8] K.T. Chau, *Energy Systems for Electric and Hybrid Vehicles Induction Machine.* Institution of Engineering and Technology, 2016.

[9] D.A. Tibaduiza, I. Amaya, S. Rodríguez, N. Mejia, M. Flórez, "Implementación de un control fuzzy para el control cinemático directo en un robot manipulador", *Ingeniare. Rev.chil*, vol.19, no.3, pp.312-322.

[10] F. Pasquill, *Atmospheric diffusion* . 2nd. ed. UK: Horwood, 1974.

[11] The Mathworks website. [Online]. Available: https://www.mathworks.com/products/matlab.html