



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



Reporte Práctica 4 “Lógica Difusa en un sistema de control para lavado de ropa”

Integrantes: Cruz Barragan Ryan Nathanael - Rodríguez Vázquez Joshua Levi - Torres
Abonce Luis Miguel

Correos: RNCB0963@gmail.com – joshualevirv@gmail.com –
luiseishon9@gmail.com

I. Resumen

El programa fue desarrollado en Python. Su objetivo es simular el control del lavado de ropa utilizando lógica difusa, incluyendo las reglas definidas. Con base a los factores involucrados (químico, mecánico, térmico, tiempo de lavado), a las consideraciones que plantea el problema y el comportamiento del programa, da los resultados de las cantidades necesarias de otros factores esperados para un buen lavado, mediante reglas establecidas y ponderaciones que permiten dar un mejor resultado.

II. Palabras clave

Lógica difusa, reglas difusas, fuzzificación, desfuzzificación, conjunto difuso, Python.

III. Introducción

Una de las grandes diferencias que existe entre el cerebro humano y la inteligencia artificial (IA) es el manejo de la información inexacta, ya que los sistemas computacionales aún no han logrado resolver este problema de manera eficiente. Es ahí cuando inicia el estudio de la lógica difusa, con bases de la lógica multivaluada (S. XX, Jan Lukasiewicz). La lógica difusa es una herramienta vital para la IA que permite tomar decisiones con base a las reglas que le proveen.

También conocida como lógica borrosa, la lógica difusa permite que las variables tomen valores continuos dentro de un rango definido. De esta manera, se obtiene un comportamiento más flexible que otro sistema

informático convencional con la capacidad de trabajar con la imprecisión de las decisiones.

Esta ventaja que tiene la ha convertido en una técnica muy utilizada en diversos aspectos de control: navegación, localización, construcción de mapas, etc. Además, cabe aclarar que este tipo de lógica lleva un orden para convertir los datos cuantificables dentro de un conjunto difuso (fuzzificación) y después de haber hecho un determinado proceso de análisis de datos y aplicación de las reglas difusas devuelve un valor dentro del conjunto difuso y lo convierte nuevamente en un valor que puede ser medido (desfuzzificación).

El trabajo presentado a continuación trata de presentar el funcionamiento de la lógica difusa, mediante la simulación del proceso de selección de lavado. La implementación de las reglas difusas permite crear un sistema mucho más flexible y adaptable, donde se ajusta automáticamente a los parámetros de lavado, según las especificaciones.

IV. Desarrollo

La práctica fue desarrollada en Python, dentro del cual se consideraron tres características fundamentales para su implementación: entradas, salidas y reglas de difusión (conjuntos difusos).

A. Entradas

Las entradas son los datos que introduce el usuario.

1. **tipo_tela:** indica el tipo de material con el que está hecho la prenda. En donde la escala va de 0 a 10 con las siguientes descripciones: delicada (0), normal (5) y resistente (10).
2. **cantidad_ropa:** indica la cantidad de ropa que se va a utilizar. En donde la escala va de 0 a 10 con las siguientes descripciones: poca (0), media (5) y mucha (10).
3. **grado_suciedad:** indica qué tan sucia está la ropa. En donde la escala va de 0 a 10 con las siguientes descripciones: ligera (0), media (5) y pesada (10).

Cabe aclarar que los datos introducidos son cantidades cuantificables, es decir, números que pueden ser medidos dentro de un rango especificado.

B. Salidas

Las salidas son los resultados que el sistema brinda de acuerdo a las entradas introducidas y las reglas de difusión definidas.

1. **cantidad_agua:** indica la cantidad de agua necesaria para lavar la ropa. En donde la escala va de 0 a 100 con las siguientes descripciones: poca (0), media (50) y mucha (100).
2. **temperatura_agua:** indica a qué temperatura debe estar el agua al momento de lavar la ropa. En donde la escala va de 0 a 100 con las siguientes descripciones: fría (0), tibia (50) y caliente (100).
3. **tiempo_lavado:** indica cuánto tiempo debe dejarse lavando la ropa. En donde la escala va de 0 a 100 con las siguientes descripciones: poco (0), medio (50) y mucho (100).

Cabe aclarar que la combinación de las entradas y de las funciones de difusión crean un nuevo valor que debe ser convertido nuevamente a una cantidad cuantificable para que sea amigable hacia el usuario.

Cada entrada y salida es un conjunto difuso, en el cual tiene las descripciones que lo definen (Figura 1).

```
# Funciones de membresía
tipo_tela.automf(3, names=['delicada', 'normal', 'resistente'])
cantidad_ropa.automf(3, names=['poca', 'media', 'mucha'])
grado_suciedad.automf(3, names=['ligera', 'media', 'pesada'])

cantidad_agua.automf(3, names=['poca', 'media', 'mucha'])
temperatura_agua.automf(3, names=['fría', 'tibia', 'caliente'])
tiempo_lavado.automf(3, names=['corto', 'medio', 'largo'])
detergente.automf(3, names=['poco', 'medio', 'mucho'])
```

Figura 1. Conjuntos difusos de las entradas y salidas del sistema.

C. Funciones de difusión

Las funciones de difusión son un conjunto de 13 reglas que definen la relación que existe entre las entradas y las salidas (Figura 2). Con base a las entradas introducidas, el programa mediante condicionales y posibles escenarios que pueden ocurrir, determina las salidas factibles. Cada regla es elegida (dependiendo del escenario en el que se encuentre) mediante un sistema de control (Figura 3).

```

# Reglas de inferencia
regla1 = ctrl.Rule(tipo_tela['resistente'] & grado_suciedad['pesada'],
                  [detergente['mucho'], tiempo_lavado['largo'], temperatura_agua['caliente'], cantidad_agua['mucho']])
regla2 = ctrl.Rule(tipo_tela['delicada'] & grado_suciedad['ligera'],
                  [detergente['poco'], tiempo_lavado['corto'], temperatura_agua['fria'], cantidad_agua['poca']])
regla3 = ctrl.Rule(cantidad_ropa['mucho'] & grado_suciedad['media'],
                  [detergente['medio'], tiempo_lavado['medio'], temperatura_agua['tibia'], cantidad_agua['media']])
regla4 = ctrl.Rule(tipo_tela['normal'] & cantidad_ropa['poca'] & grado_suciedad['ligera'],
                  [detergente['poco'], tiempo_lavado['corto'], temperatura_agua['fria'], cantidad_agua['poca']])
regla5 = ctrl.Rule(tipo_tela['delicada'] & cantidad_ropa['mucho'] & grado_suciedad['pesada'],
                  [detergente['mucho'], tiempo_lavado['largo'], temperatura_agua['caliente'], cantidad_agua['mucho']])
regla6 = ctrl.Rule(tipo_tela['normal'] & cantidad_ropa['mucho'] & grado_suciedad['ligera'],
                  [detergente['medio'], tiempo_lavado['medio'], temperatura_agua['tibia'], cantidad_agua['media']])
regla7 = ctrl.Rule(tipo_tela['resistente'] & cantidad_ropa['mucho'] & grado_suciedad['pesada'],
                  [detergente['mucho'], tiempo_lavado['largo'], temperatura_agua['caliente'], cantidad_agua['mucho']])
regla8 = ctrl.Rule(tipo_tela['resistente'] & cantidad_ropa['poca'] & grado_suciedad['media'],
                  [detergente['medio'], tiempo_lavado['medio'], temperatura_agua['tibia'], cantidad_agua['media']])
regla9 = ctrl.Rule(tipo_tela['delicada'] & cantidad_ropa['poca'] & grado_suciedad['ligera'],
                  [detergente['poco'], tiempo_lavado['corto'], temperatura_agua['fria'], cantidad_agua['poca']])
regla10 = ctrl.Rule(tipo_tela['normal'] & grado_suciedad['media'],
                  [detergente['medio'], tiempo_lavado['medio'], temperatura_agua['tibia'], cantidad_agua['media']])
regla11 = ctrl.Rule(tipo_tela['resistente'] & grado_suciedad['ligera'],
                  [detergente['poco'], tiempo_lavado['corto'], temperatura_agua['fria'], cantidad_agua['poca']])
regla12 = ctrl.Rule(tipo_tela['delicada'] & cantidad_ropa['media'],
                  [detergente['medio'], tiempo_lavado['medio'], temperatura_agua['tibia'], cantidad_agua['media']])
regla13 = ctrl.Rule(tipo_tela['normal'] & cantidad_ropa['media'] & grado_suciedad['pesada'],
                  [detergente['mucho'], tiempo_lavado['largo'], temperatura_agua['caliente'], cantidad_agua['mucho']])

```

Figura 2. Reglas de difusión con base a las entradas para determinar las salidas.

```

# Sistema de control
control_system = ctrl.ControlSystem([regla1, regla2, regla3, regla4, regla5, regla6, regla7, regla8, regla9, regla10, regla11, regla12, regla13])
return ctrl.ControlSystemSimulation(control_system)

```

Figura 3. Sistema de control para determinar qué regla utilizar.

V. Resultados

La salida del sistema se divide en tres partes: pantalla de inicio, ventana de ayuda y ventana de entrada de datos y visualización de salidas. Cada una tiene su propio objetivo y funcionamiento, sin embargo, las tres en conjunto crean el sistema de simulación de control de lavado de ropa.

A. Pantalla de inicio

En la pantalla de inicio (Figura 4) únicamente indica datos sobre el sistema (nombre y logo), así como dos botones para direccionar a la ventana de ayuda o empezar a introducir los datos.

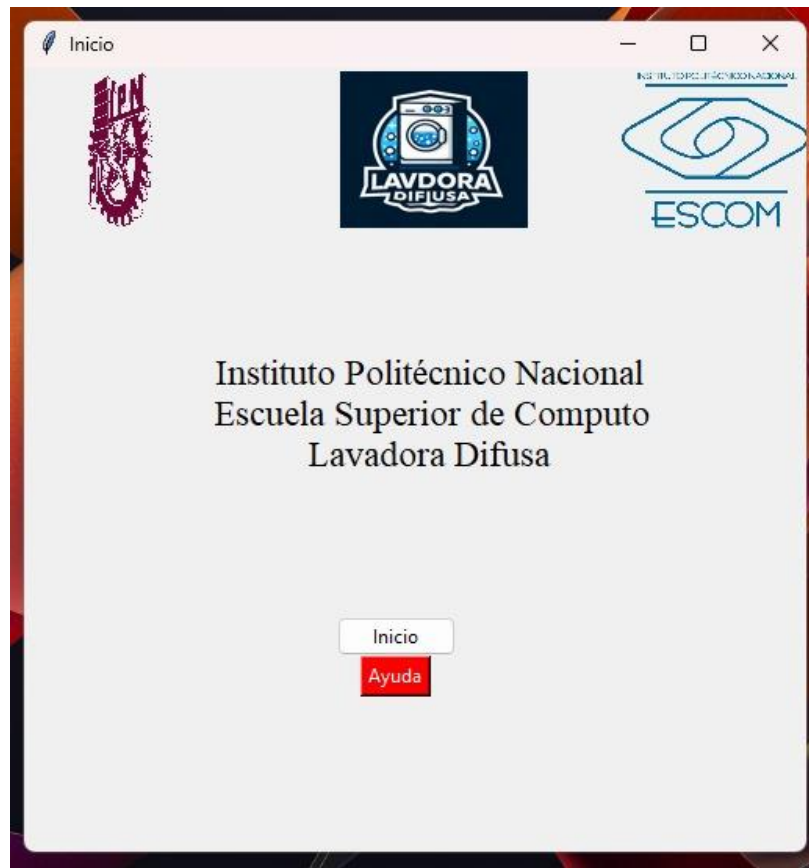


Figura 4. Pantalla de inicio.

B. Ventana de ayuda

La ventana de ayuda (Figura 5) le permite al usuario tener un manual para empezar a utilizar el sistema. Le muestra una serie de pasos para conocer su funcionamiento, así mismo, despliega una tabla con la información de las categorías (entradas y salidas) y su descripción.

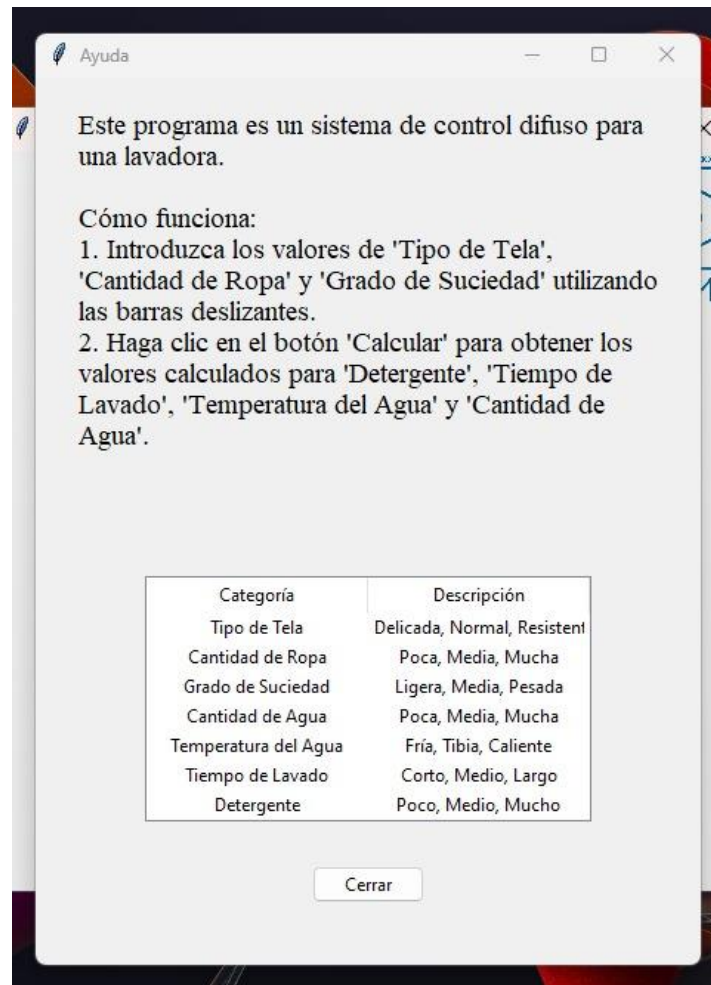


Figura 5. Ventana de ayuda.

C. Sistema de Control para Lavado de Ropa

En la ventana del sistema de control para el lavado de la ropa es donde el usuario introduce las especificaciones iniciales mediante una barra deslizante para elegir dentro del rango especificado (Figura 6), además muestra las salidas después de haberle dado en el botón "Calcular" (Figura 7), de esta manera muestra los datos con los que se debe configurar la lavadora para que la ropa quede de la mejor manera, contemplando las especificaciones dadas por el usuario.

Sistema de Control para Lavado de Ropa

Regresar

Tipo de Tela (0-Delicada, 10-Resistente):

Cantidad de Ropa (0-Poca, 10-Mucha):

Grado de Suciedad (0-Ligera, 10-Pesada):

Calcular

Detergente: 0

Tiempo de Lavado: 0

Temperatura del Agua: 0

Cantidad de Agua: 0

Figura 6. Ventana del Sistema de Control para Lavado de Ropa

Sistema de Control para Lavado de Ropa

Regresar

Tipo de Tela (0-Delicada, 10-Resistente):

Cantidad de Ropa (0-Poca, 10-Mucha):

Grado de Suciedad (0-Ligera, 10-Pesada):

Calcular

Detergente: 51.41

Tiempo de Lavado: 61.70

Temperatura del Agua: 51.41

Cantidad de Agua: 51.41

Figura 7. Ingreso de datos y visualización de resultados.

VI. Conclusión

Con el desarrollo de la práctica, las investigaciones previas y durante el desarrollo se observó la funcionalidad y utilidad de la lógica difusa. Nos permitió crear el diseño y control del proceso de lavado a través de reglas difusas y con los datos de entrada, generando así nuevos datos de salida. Esto permitió una optimización de los parámetros variables de lavado, permitiendo adaptar los resultados conforme a las especificaciones iniciales.

Por otro lado, la implementación de los conjuntos difusos permitió separar una categoría de otra, con sus respectivas características. Además, las reglas difusas nos permitieron tener una diversidad de posibles escenarios donde dependiendo de las combinaciones de las entradas dentro del rango, permitieron obtener cifras para llevar un control sobre el lavado de la ropa.

Entonces, podemos concluir que la lógica difusa dentro de un sistema es muy confiable si se trabaja con valores variables, ya que permite trabajar con valores con cierto grado de incertidumbre, pero al final devuelve un valor confiable, llevando un proceso complejo puede recibir diferentes entradas a la vez y mediante las reglas permite obtener una salida coherente.

VII. Bibliografía

- [1] Hurtado Palacio, J., & Buitrago Giraldo, J. (2014). LÓGICA DIFUSA: PERSPECTIVA Y APLICACIONES (1.a ed., Vol. 1) [Facultad de Ciencias Básicas y Tecnologías].
<https://bdigital.uniquindio.edu.co/bitstream/handle/001/5517/MONOGRAF%C3%8DA-JUAN%20PABLO%20HURTADO%20PALACIO.pdf>
- [2] I. Pérez, B. León, Logica difusa para principiantes. Caracas: Universidad Catolica Andres, 2007.
- [3] Smith, L. (n.d.). Programming Fuzzy Systems in Python.
https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_tipping_problem_newapi.html
- [4] Warner, J. (2018). Scikit-Fuzzy: Fuzzy Logic Toolbox for Python. Zenodo.
doi:10.5281/zenodo.1205566