

# TareaRiegoDifuso

November 17, 2018

CONTROL INTELIGENTE

**Maestro:** Dr. Agustín Flores Novelo

**Autor:** Ing. José Alfonso Ureña Pajón

## 1 Objetivo:

Familiarizarse con la Lógica Difusa aplicada al control.

## 2 Desarrollo:

Implementar un control difuso para un sistema clásico (Temperatura)

1. Generar un control difuso para un sistema de temperatura con las siguientes características:  
Entradas: \* La temperatura del aire exterior \* Humedad del suelo  
Salida: \* La duración del agua
2. Implementar el sistema de control difuso anterior. (para este ejercicio se implementa en python)

## 3 Definición de las variables de entrada/salida y de reglas difusas

Ejercicio de sistema de riego desarrollado por medio de Lógica Difusa y utilizando la librería de python [SciKit-Fuzzy](#).

Recordando que los sistemas difusos consisten en:

- **si** premisa (antecedente) **entonces** conclusión (consecuencia)

**Los antecedentes (variables de entrada) son:**

1. Temperatura
  - Universo: temperatura en grados celcius de 5 a 50 grados
  - Conjunto difuso: fresco, templado y caliente.
2. Humedad
  - Universo: Humedad del suelo, medida con sensor de humedad generico para arduino, medido de 0 a 880 según medición de salida del [Sensor de Humedad de Sparkfun](#).

- Conjunto difuso: seco, humedo y mojado.

Los consecuentes (variables de salida) son:

#### 1. Tiempo

- Universo: tiempo que permanece abierta de una valvula de flujo de agua, este tiempo se define en minutos y va de 0 a 30 minutos.
- Conjunto difuso: Nada, Muy poco, Poco, Medio, Mucho y Muchisimo tiempo tiempo de riego.

#### Reglas difusas

- **si** el suelo esta seco **y** la temperatura es fresca **entonces** el tiempo de riego es mucho.
- **si** el suelo esta seco **y** la temperatura es templada **entonces** el tiempo de riego es mucho.
- **si** el suelo esta seco **y** la temperatura es caliente **entonces** el tiempo de riego es muchisimo.
- **si** el suelo esta humedo **y** la temperatura es fresca **entonces** el tiempo de riego es muy poco.
- **si** el suelo esta humedo **y** la temperatura es templada **entonces** el tiempo de riego es poco.
- **si** el suelo esta humedo **y** la temperatura es caliente **entonces** el tiempo de riego es medio.
- **si** el suelo esta mojado **entonces** el tiempo de riego es nada.

## 4 Implementación del sistema de control difuso

```
In [1]: # Se importan las librerias principales Numpy y Skfuzzy y
# se especifica que las gráficas son locales
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import skfuzzy as fuzz
```

```
In [2]: #Condiciones actuales de temperatura y humedad
CURRENT_TEMP = 12
CURRENT_HUM = 500

# Generación del universo de las funciones
temperatura = np.arange(5, 51, 1)
humedad = np.arange(0, 881, 1)
riego = np.arange(0, 31, 1)

# Función característica para la Temperatura
t_fresca = fuzz.trapmf(temperatura, [5, 5, 10, 18])
t_templada = fuzz.trimf(temperatura, [15, 20, 27])
t_caliente = fuzz.trapmf(temperatura, [24, 35, 50, 51])
```

```

# Función característica para la Humedad
h_seco = fuzz.trimf(humedad, [0, 0, 150, 250])
h_humedo = fuzz.trimf(humedad, [200, 440, 680])
h_mojado = fuzz.trimf(humedad, [600, 700, 880, 880])

# Función característica para el tiempo de riego
r_nada = fuzz.trimf(riego, [-.5, 0, 0])
r_muy_poco = fuzz.trimf(riego, [0, 5, 10])
r_poco = fuzz.trimf(riego, [5, 10, 15])
r_medio = fuzz.trimf(riego, [10, 15, 20])
r_mucho = fuzz.trimf(riego, [15, 23, 30])
r_muchisimo = fuzz.trimf(riego, [25, 30, 30])

```

## 4.1 Visualización de las funciones características

```

In [3]: # Funciones características para la humedad
fig, (ax0, ax1, ax2) = plt.subplots(nrows=3, figsize=(8, 9))

ax0.plot(humedad, h_seco, 'r', linewidth=1.5, label='Seco')
ax0.plot(humedad, h_humedo, 'm', linewidth=1.5, label='Humedo')
ax0.plot(humedad, h_mojado, 'b', linewidth=1.5, label='Mojado')
ax0.set_ylabel('Sets Difusos')
ax0.set_xlabel('Nivel de humedad')
ax0.legend()
ax0.set_title('Humedad')

# Funciones características para la temperatura
ax1.plot(temperatura, t_fresca, 'b', linewidth=1.5, label='Fresca')
ax1.plot(temperatura, ttemplada, 'm', linewidth=1.5, label='Templada')
ax1.plot(temperatura, t_caliente, 'r', linewidth=1.5, label='Caliente')
ax1.set_ylabel('Sets Difusos')
ax1.set_xlabel('Temperatura (Celcius)')
ax1.legend()
ax1.set_title('Temperatura')

# Funciones características para el tiempo de riego
ax2.plot(riego, r_nada, 'gray', linewidth=1.5, label='Sin riego')
ax2.plot(riego, r_muy_poco, 'firebrick', linewidth=1.5, label='Muy poco riego')
ax2.plot(riego, r_poco, 'r', linewidth=1.5, label='Poco Riego')
ax2.plot(riego, r_medio, 'g', linewidth=1.5, label='Riego medio')
ax2.plot(riego, r_mucho, 'royalblue', linewidth=1.5, label='Mucho riego')
ax2.plot(riego, r_muchisimo, 'navy', linewidth=1.5, label='Muchisimo riego')
ax2.set_ylabel('Sets Difusos')
ax2.set_xlabel('Tiempo de riego')
ax2.legend()
ax2.set_title('Tiempo de Riego')

# Editar Ejes

```

```

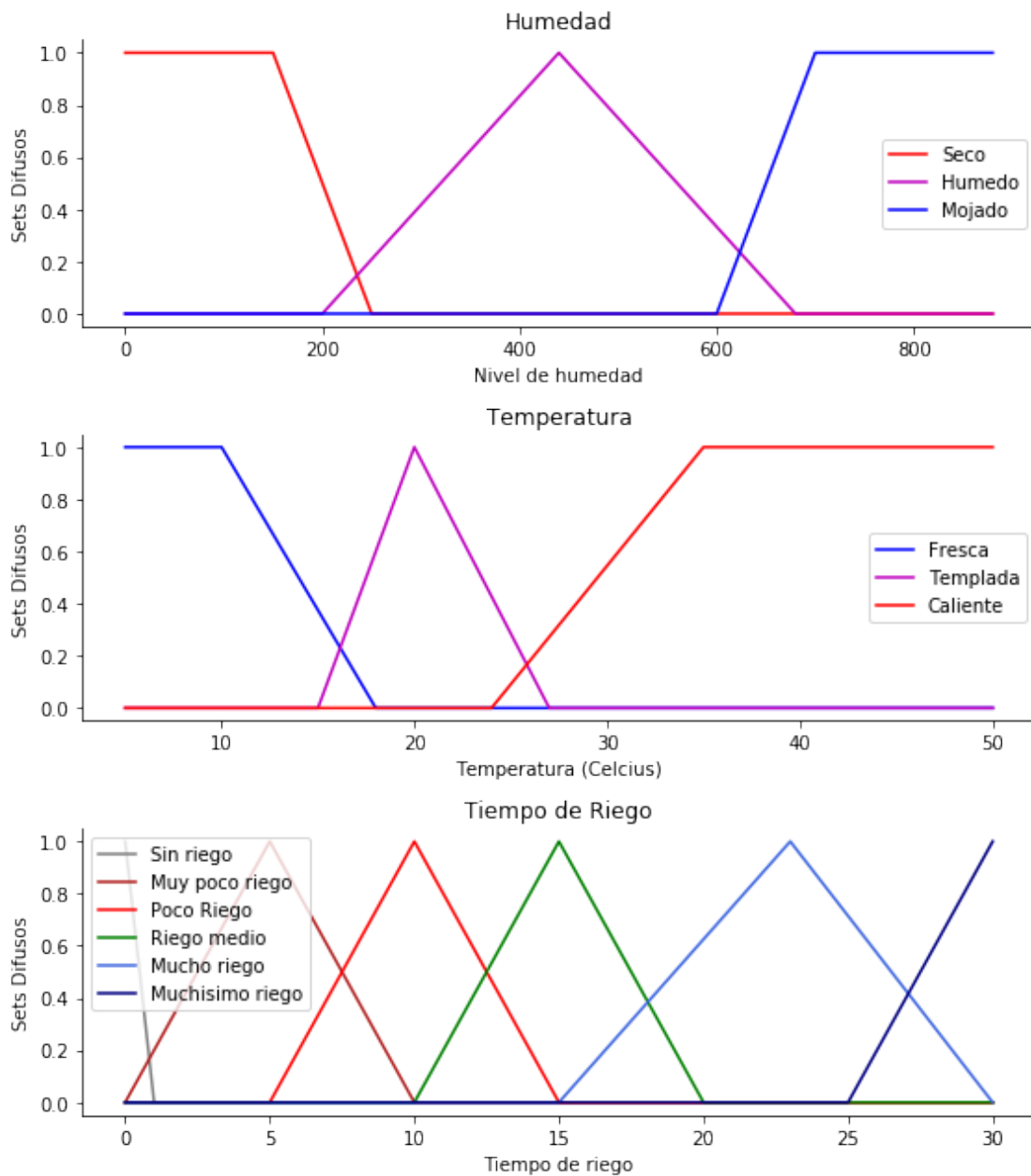
for ax in (ax0, ax1, ax2):
    ax.spines['top'].set_visible(False)
    ax.spines['right'].set_visible(False)
    ax.get_xaxis().tick_bottom()
    ax.get_yaxis().tick_left()

```

```

plt.tight_layout()

```



## 5 Interpolación de datos y Creación de las relaciones difusas

```
In [4]: #Interpolación de datos para saber los valores correspondientes
hum_level_seco = fuzz.interp_membership(humedad, h_seco, CURRENT_HUM)
hum_level_humedo = fuzz.interp_membership(humedad, h_humedo, CURRENT_HUM)
hum_level_mojado = fuzz.interp_membership(humedad, h_mojado, CURRENT_HUM)

temp_level_fresco = fuzz.interp_membership(temperatura, t_fresca, CURRENT_TEMP)
temp_level_templado = fuzz.interp_membership(temperatura, t_templada, CURRENT_TEMP)
temp_level_caliente = fuzz.interp_membership(temperatura, t_caliente, CURRENT_TEMP)

# Relaciones difusas
R1 = np.fmin( np.fmin(hum_level_seco, temp_level_fresco), r_mucho)
R2 = np.fmin( np.fmin(hum_level_seco, temp_level_templado), r_mucho)
R3 = np.fmin( np.fmin(hum_level_seco, temp_level_caliente), r_muchisimo)
R4 = np.fmin( np.fmin(hum_level_humedo, temp_level_fresco), r_muy_poco)
R5 = np.fmin( np.fmin(hum_level_humedo, temp_level_templado), r_poco)
R6 = np.fmin( np.fmin(hum_level_humedo, temp_level_caliente), r_medio)
R7 = np.fmin(hum_level_mojado, r_nada)
```

## 6 Defusificación

```
In [5]: # Combinacion de todas la reglas
R_combined = np.fmax(R1, np.fmax(R2, np.fmax(R3, np.fmax( \
    R4,np.fmax(R5, np.fmax(R6,R7))))))

#Defuzzificación
riego_salida = fuzz.defuzz(riego, R_combined, 'centroid')
riego_activacion = fuzz.interp_membership(riego, R_combined, riego_salida)
```

## 7 Visualización de resultados

```
In [6]: # Visualización
riego0 = np.zeros_like(riego)
fig, ax0 = plt.subplots(figsize=(8, 3))

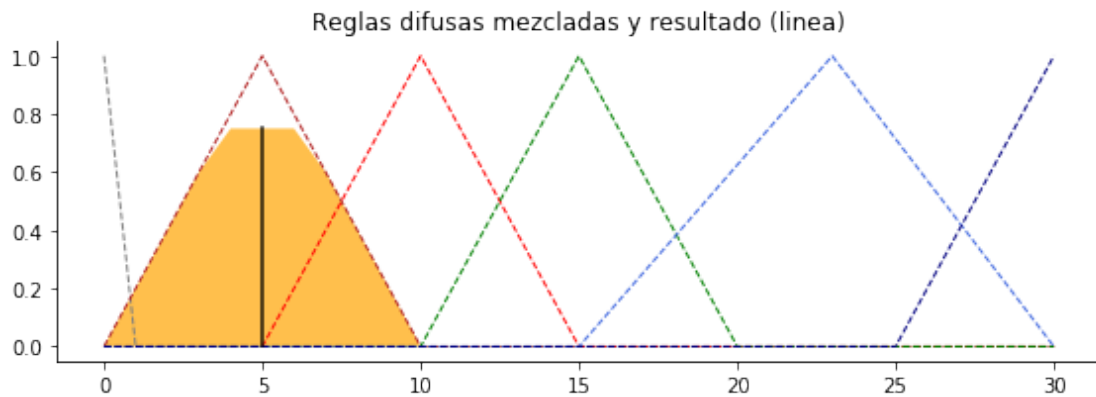
ax0.plot(riego, r_nada, 'gray', linewidth=1, linestyle='--', )
ax0.plot(riego, r_muy_poco, 'firebrick', linewidth=1, linestyle='--')
ax0.plot(riego, r_poco, 'r', linewidth=1, linestyle='--')
ax0.plot(riego, r_medio, 'g', linewidth=1, linestyle='--')
ax0.plot(riego, r_mucho, 'royalblue', linewidth=1, linestyle='--')
ax0.plot(riego, r_muchisimo, 'navy', linewidth=1, linestyle='--')
ax0.fill_between(riego, riego0, R_combined, facecolor='Orange', alpha=0.7)
ax0.plot([riego_salida, riego_salida], [0, riego_activacion], 'k', \
    linewidth=1.5, alpha=0.9)
ax0.set_title('Reglas difusas mezcladas y resultado (linea)')
```

```

# Quitar ejes
for ax in (ax0,):
    ax.spines['top'].set_visible(False)
    ax.spines['right'].set_visible(False)
    ax.get_xaxis().tick_bottom()
    ax.get_yaxis().tick_left()

plt.tight_layout()

```



Para las condiciones iniciales el tiempo en minutos que debe permanecer encendido del sistema de riego es:

```
In [7]: print("Tiempo de riego calculado: %.1f minutos" % riego_salida)
```

Tiempo de riego calculado: 5.0 minutos