

به نام خدا



گزارش تحلیلی پروژه دوم منطق فازی

موضوع :

پیاده سازی کنترل کننده منطق فازی برای سیستم آبیاری خودکار گیاه

درس: هوش محاسباتی

استاد: دکتر حسین کارشناس

دستیاران آموزش :

رضا برزگر - علی شاه زمانی - آرمان خلیلی

اعضای گروه :

یونس ایوبی راد

پویا اسفندانی

Fuzzy Variables

```
In [3]: # Defining the scope of variables
soil_moisture_range = np.arange(0, 101, 1)
weather_range = np.arange(0, 11, 1)
irrigation_range = np.arange(0, 11, 1)

# Antecedents (Inputs) - Used by all systems initially
soil_moisture_orig = ctrl.Antecedent(soil_moisture_range, 'soil_moisture_orig')
weather_orig = ctrl.Antecedent(weather_range, 'weather_orig')

# Consequent (Output) - Using Centroid for the main simulation (Original System)
irrigation_centroid = ctrl.Consequent(irrigation_range, 'irrigation_centroid', defuzzify_method='centroid')
```

Membership Functions

```
In [4]: # Membership Functions for Soil Moisture (Original)
soil_moisture_orig['dry'] = fuzz.trimf(soil_moisture_orig.universe, [0, 0, 50])
soil_moisture_orig['medium'] = fuzz.trimf(soil_moisture_orig.universe, [20, 50, 80])
soil_moisture_orig['moist'] = fuzz.trimf(soil_moisture_orig.universe, [50, 100, 100])

# Membership Functions for Weather (Original)
weather_orig['sunny'] = fuzz.trimf(weather_orig.universe, [0, 0, 4])
weather_orig['cloudy'] = fuzz.trimf(weather_orig.universe, [2, 5, 8])
weather_orig['rainy'] = fuzz.trimf(weather_orig.universe, [6, 10, 10])

# Membership Functions for Irrigation Amount (shared shape for all methods initially)
irrigation_mfs_orig = {
    'no_water': fuzz.trimf(irrigation_range, [0, 0, 2]),
    'low': fuzz.trimf(irrigation_range, [1, 3, 5]),
    'medium': fuzz.trimf(irrigation_range, [4, 6, 8]),
    'high': fuzz.trimf(irrigation_range, [7, 10, 10])
}

# Assign MFs to the centroid consequent
for label, mf in irrigation_mfs_orig.items():
    irrigation_centroid[label] = mf
```

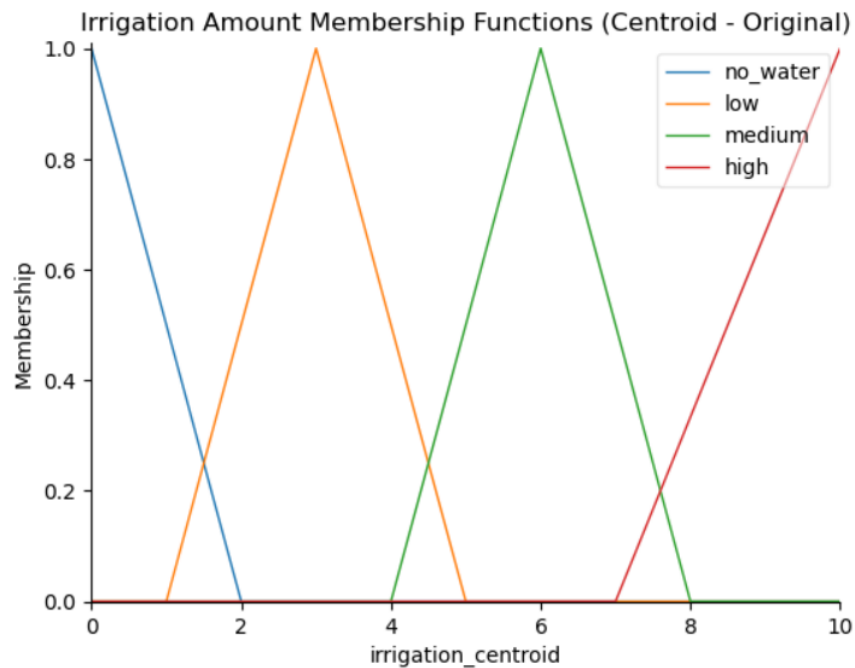
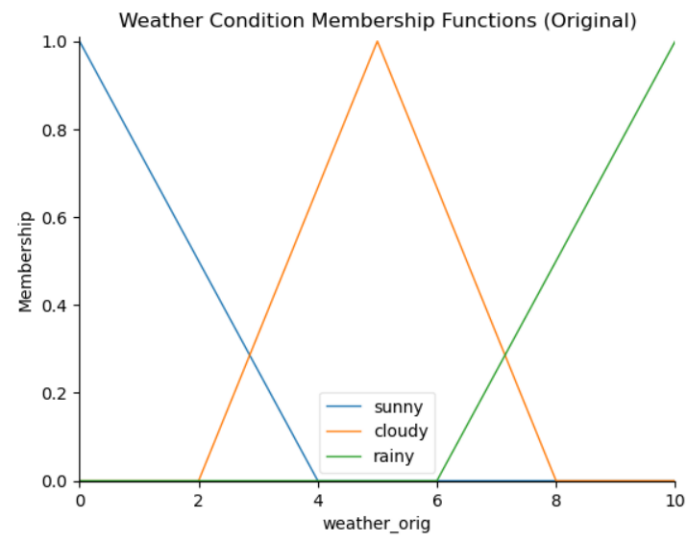
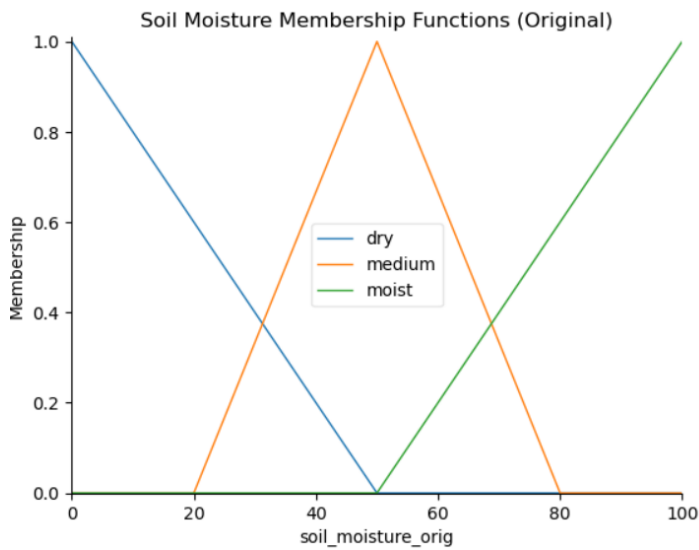
تعریف متغیرهای فازی و توابع عضویت

در این بخش از پروژه، متغیرهای فازی مربوط به ورودی‌ها و خروجی سیستم کنترل آبیاری تعریف شده‌اند. سه متغیر اصلی مورد استفاده قرار گرفته‌اند:

1. رطوبت خاک – (Soil Moisture) به عنوان یک متغیر ورودی با دامنه $[0, 100]$
2. وضعیت آب‌وهوا – (Weather) به عنوان ورودی دیگر با دامنه $[0, 10]$
3. مقدار آبیاری – (Irrigation Amount) به عنوان خروجی سیستم با دامنه $[0, 10]$ و روش واسازی (defuzzification) از نوع *centroid*

برای هر یک از این متغیرها، توابع عضویت مثلثی (Triangular Membership Functions) تعریف شده‌اند که ویژگی‌های زبانی مختلف را مدل می‌کنند. برای مثال، رطوبت خاک به صورت «خشک»، «متوسط» و «مرطوب» دسته‌بندی شده

است، و آب و هوا شامل «آفتابی»، «ابری» و «بارانی» می باشد. همچنین مقدار آبیاری با چهار برچسب زبانی «بدون آب»، «کم»، «متوسط» و «زیاد» مدل سازی شده است.



Control Rules

- Define Control Rules (Original - 9 Rules)
- Create Control System for Centroid

```
: rule1 = ctrl.Rule(soil_moisture_orig['dry'] & weather_orig['sunny'], irrigation_centroid['high'])
rule2 = ctrl.Rule(soil_moisture_orig['dry'] & weather_orig['cloudy'], irrigation_centroid['medium'])
rule3 = ctrl.Rule(soil_moisture_orig['dry'] & weather_orig['rainy'], irrigation_centroid['low'])
rule4 = ctrl.Rule(soil_moisture_orig['medium'] & weather_orig['sunny'], irrigation_centroid['medium'])
rule5 = ctrl.Rule(soil_moisture_orig['medium'] & weather_orig['cloudy'], irrigation_centroid['low'])
rule6 = ctrl.Rule(soil_moisture_orig['medium'] & weather_orig['rainy'], irrigation_centroid['no_water'])
rule7 = ctrl.Rule(soil_moisture_orig['moist'] & weather_orig['sunny'], irrigation_centroid['low'])
rule8 = ctrl.Rule(soil_moisture_orig['moist'] & weather_orig['cloudy'], irrigation_centroid['no_water'])
rule9 = ctrl.Rule(soil_moisture_orig['moist'] & weather_orig['rainy'], irrigation_centroid['no_water'])

rules_original = [rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8, rule9]

irrigation_ctrl_original = ctrl.ControlSystem(rules_original)
irrigation_simulation_original = ctrl.ControlSystemSimulation(irrigation_ctrl_original)
```

تعریف قواعد فازی (Fuzzy Rules)

در این بخش، مجموعه‌ای از ۹ قاعده فازی طراحی شده که رفتار سیستم آبیاری را بر اساس ترکیب ورودی‌های «رطوبت خاک» و «وضعیت آب‌وهوا» تعیین می‌کند. هدف از این قواعد، تعیین مقدار مناسب آبیاری در شرایط مختلف محیطی است.

هر قاعده به صورت منطقی (AND) بین دو متغیر ورودی تعریف شده است و بر اساس آن، یک مقدار خروجی مشخص برای آبیاری تعیین می‌گردد. به عنوان نمونه:

- اگر خاک خشک و هوا آفتابی باشد، مقدار آبیاری باید زیاد (high) باشد.

- اگر خاک مرطوب و هوا بارانی باشد، آبیاری نیاز نیست. (no_water)

این قواعد تمام حالت‌های ممکن بین سه سطح رطوبت و سه نوع وضعیت آب‌وهوا را پوشش می‌دهند.

در پایان، این قواعد در قالب یک سیستم کنترل فازی پیاده‌سازی شده‌اند و با استفاده از آن، شبیه‌سازی سیستم آبیاری فازی اصلی انجام خواهد شد.

Alternative Defuzzification Methods

- Implement and Compare Alternative Defuzzification Methods
- Using the ORIGINAL 9-Rule structure for this comparison
- bisector - mom (Mean of Maximum) - lom (Largest of Maximum)

پیاده‌سازی و مقایسه روش‌های مختلف واسازی (Defuzzification)

در این بخش از پروژه، چهار روش مختلف برای واسازی مقدار خروجی آبیاری مورد بررسی قرار گرفته‌اند. روش‌ها عبارتند از:

• Centorid (مرکز ثقل)

• Bisector (نیم ساز)

• Mean of Maximum (MoM) (میانگین مقادیر بیشینه)

• Left of Maximum (LoM) (سمت چپ بیشینه)

برای هر یک از این روش‌ها، متغیر خروجی جدیدی تعریف شده و توابع عضویت اولیه روی آن‌ها اعمال شده‌اند. سپس با استفاده از تابع کمکی، همان قواعد فازی اولیه برای هر متغیر خروجی تولید و سیستم کنترل متناظر ساخته شده است.

به‌منظور مقایسه خروجی این روش‌ها، چند سناریوی آزمایشی با مقادیر مختلفی از رطوبت خاک و وضعیت آب‌وهوا تعریف و شبیه‌سازی شده‌اند. نتایج به‌صورت عددی بین ۰ تا ۱۰ برای هر روش چاپ شده‌اند تا تفاوت‌های احتمالی در نتایج واسازی مشخص شود.

این مقایسه می‌تواند برای تحلیل دقت، حساسیت و عملکرد عملی هر روش در سیستم‌های تصمیم‌یار فازی مفید باشد.

Comparing Defuzzification Outputs for Test Inputs:

Scenario	Centroid	Bisector	MoM	LoM
Dry Soil, Sunny Weather	8.93	9.02	10.00	10.00
Dry Soil, Cloudy Weather	4.61	4.74	6.00	7.00
Medium Soil, Cloudy Weather	3.00	3.00	3.00	3.00
Moist Soil, Rainy Weather	0.78	0.75	0.50	1.00
Medium Soil, Sunny Weather	5.42	5.67	6.00	7.00

1. همگرایی نتایج در شرایط خاص:

- در سناریوی "*Medium Soil, Cloudy Weather*"، هر چهار روش دقیقاً مقدار یکسانی (۳.۰۰) ارائه داده‌اند. این نشان می‌دهد که در این حالت، ناحیه فعال تابع عضویت نسبتاً متقارن و تک‌پیک است، بنابراین تفاوتی بین روش‌ها ایجاد نمی‌شود.

2. تفاوت چشمگیر در مقادیر حداکثری:

- در حالت "*Dry Soil, Sunny Weather*"، روش‌های MoM و LoM مقدار ۱۰.۰۰ را تولید کرده‌اند، در حالی که روش‌های Centroid و Bisector به مقادیر کمی پایین‌تر (حدود ۹) رسیده‌اند. این نشان می‌دهد که MoM و LoM بیشتر به نقاط بیشینه تابع عضویت وابسته‌اند و گرایش به برآورد "حداکثر ممکن" دارند.

3. رفتار متفاوت در مقادیر پایین:

- در "*Moist Soil, Rainy Weather*"، مقادیر بسیار پایین بوده‌اند که منطقی است (خاک مرطوب و باران نیازی به آبیاری ندارد). اما روش‌های مختلف باز هم تفاوت اندکی دارند؛ Centroid و Bisector به میانگین‌ها نزدیک شده‌اند، در حالی که MoM به مقدار ۰.۵۰ و LoM به ۱.۰۰ تمایل نشان داده‌اند.

4. تفاوت متوسط در شرایط میانی:

- در "*Dry Soil, Cloudy Weather*" و "*Medium Soil, Sunny Weather*"، مشاهده می‌شود که Centroid و Bisector مقادیر میانی‌تر و هموارتری ارائه داده‌اند، در حالی که MoM و LoM مقداری بالاتر هستند. این نشان می‌دهد روش‌های Centroid و Bisector معمولاً پاسخ‌های نرم‌تر و دقیق‌تری برای شرایط ناپایدار فراهم می‌کنند.

نتیجه :

Centroid و Bisector برای شرایط میانی عملکرد نرم‌تری دارند و مناسب تحلیل‌هایی هستند که نیاز به دقت پیوسته دارند.

MoM و LoM به دلیل وابستگی به بیشینه، در تصمیم‌گیری‌های سریع یا بحرانی می‌توانند مفید باشند، اما ممکن است منجر به واکنش بیش از حد در برخی شرایط شوند.

انتخاب روش مناسب وابسته به هدف نهایی سیستم (دقت مقابل سرعت واکنش) است.

پیاده سازی نسخه دوم پروژه با 15 ویژگی:

در این پروژه، دو نسخه از یک سیستم آبیاری هوشمند مبتنی بر منطق فازی مورد بررسی قرار گرفته‌اند. نسخه اولیه دارای ۹ قانون و نسخه بهبود یافته دارای ۱۵ قانون فازی است که با استفاده از مجموعه‌های عضویت دقیق‌تر و تفکیک‌پذیری بیشتر ورودی‌ها طراحی شده‌اند. هدف، مقایسه میزان اثربخشی هر سیستم در حفظ رطوبت خاک در محدوده بهینه طی یک بازه ۱۰ روزه است.

Define Enhanced System (15 Rules)

```
soil_moisture_enhanced = ctrl.Antecedent(soil_moisture_range, 'soil_moisture_enhanced')
weather_enhanced = ctrl.Antecedent(weather_range, 'weather_enhanced')
irrigation_enhanced = ctrl.Consequent(irrigation_range, 'irrigation_enhanced', defuzzify_method='centroid')

# Enhanced Membership Functions
soil_moisture_enhanced['very_dry'] = fuzz.trimf(soil_moisture_enhanced.universe, [0, 0, 30])
soil_moisture_enhanced['dry'] = fuzz.trimf(soil_moisture_enhanced.universe, [10, 30, 50])
soil_moisture_enhanced['medium'] = fuzz.trimf(soil_moisture_enhanced.universe, [30, 50, 70])
soil_moisture_enhanced['moist'] = fuzz.trimf(soil_moisture_enhanced.universe, [50, 70, 90])
soil_moisture_enhanced['saturated'] = fuzz.trimf(soil_moisture_enhanced.universe, [70, 100, 100])

# Reusing original weather MFs for simplicity, assigning to new variable
weather_enhanced['sunny'] = weather_orig['sunny'].mf
weather_enhanced['cloudy'] = weather_orig['cloudy'].mf
weather_enhanced['rainy'] = weather_orig['rainy'].mf

irrigation_enhanced['no_water'] = fuzz.trimf(irrigation_enhanced.universe, [0, 0, 2])
irrigation_enhanced['very_low'] = fuzz.trimf(irrigation_enhanced.universe, [1, 2, 3])
irrigation_enhanced['low'] = fuzz.trimf(irrigation_enhanced.universe, [2, 4, 5])
irrigation_enhanced['medium'] = fuzz.trimf(irrigation_enhanced.universe, [4, 6, 7])
irrigation_enhanced['high'] = fuzz.trimf(irrigation_enhanced.universe, [6, 8, 9])
irrigation_enhanced['very_high'] = fuzz.trimf(irrigation_enhanced.universe, [8, 10, 10])
```

در این بخش، نسخه پیشرفته‌ای از سیستم فازی طراحی شده که با تعریف دقیق‌تر توابع عضویت، دقت مدل افزایش یافته است.

طراحی نسخه پیشرفته سیستم فازی (Enhanced Fuzzy System)

در این مرحله، یک مدل پیشرفته از سیستم آبیاری فازی طراحی شده که شامل جزئیات بیشتر در ورودی و خروجی‌ها می‌باشد. این کار با هدف بهبود دقت تصمیم‌گیری سیستم انجام شده است.

بهبود در ورودی‌ها: (Antecedents)

- رطوبت خاک (Soil Moisture) به جای سه حالت قبلی، اکنون در پنج سطح تعریف شده است:

○ very_dry

○ dry

medium ○

moist ○

saturated ○

○ این تفکیک ریزتر امکان مدل سازی دقیق تری از شرایط خاک را فراهم می کند.

- آب و هوا (Weather) همچنان در سه حالت (sunny, cloudy, rainy) باقی مانده و از توابع عضویت نسخه قبلی استفاده شده تا ساختار حفظ شود.

بهبود در خروجی: (Consequent)

- مقدار آبیاری نیز با شش سطح مختلف توصیف شده است:

no_water ○

very_low ○

low ○

medium ○

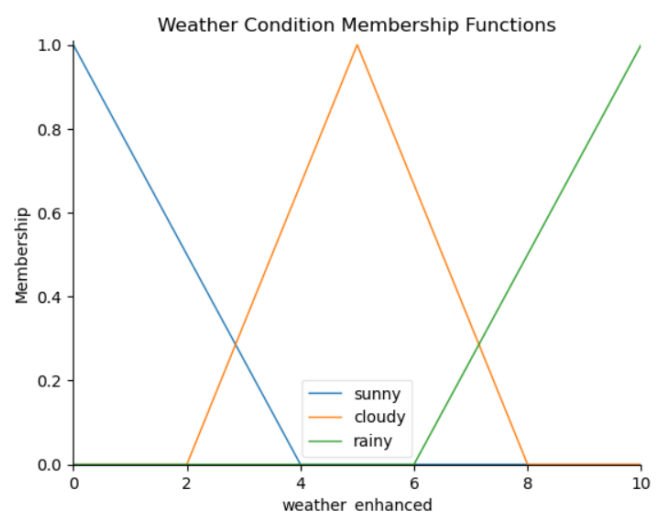
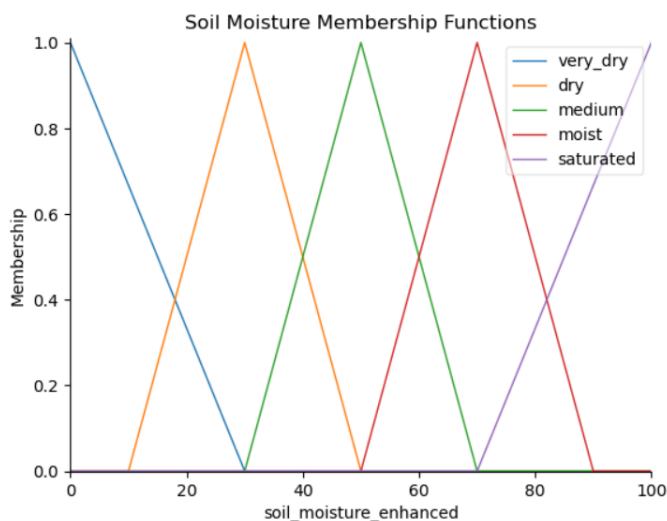
high ○

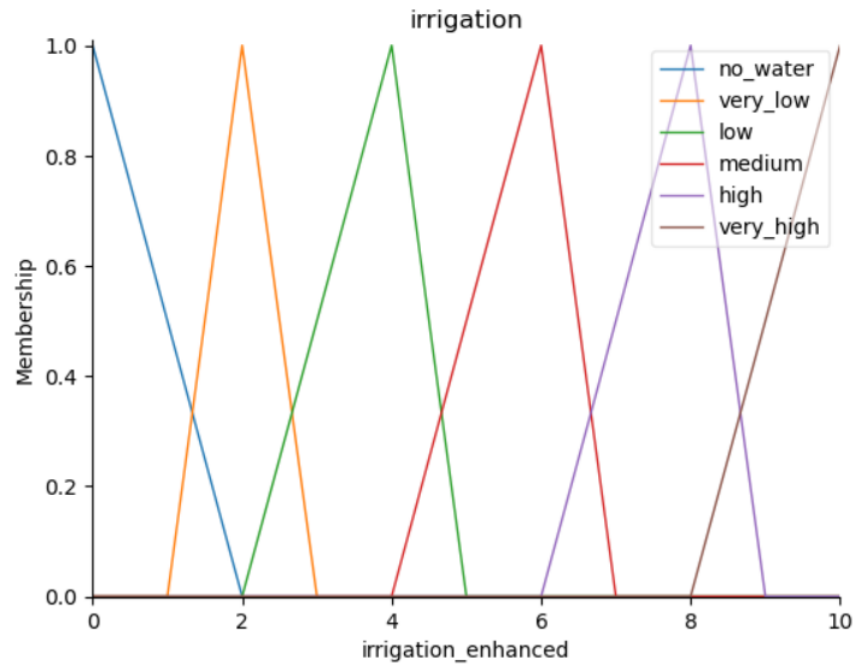
very_high ○

این دسته بندی دقیق تر، امکان اعمال تنظیمات متنوع تر آبیاری را بر اساس شرایط محیطی فراهم می سازد.

نتیجه:

این طراحی پیشرفته پایه ای مناسب برای ایجاد قواعد فازی غنی تر و انعطاف پذیرتر است که می تواند دقت تصمیم گیری در سیستم های آبیاری هوشمند را بهبود دهد.





```
# Define individual rules for the enhanced system
rule_e1 = ctrl.Rule(soil_moisture_enhanced['very_dry'] & weather_enhanced['sunny'], irrigation_enhanced['very_high'])
rule_e2 = ctrl.Rule(soil_moisture_enhanced['very_dry'] & weather_enhanced['cloudy'], irrigation_enhanced['high'])
rule_e3 = ctrl.Rule(soil_moisture_enhanced['very_dry'] & weather_enhanced['rainy'], irrigation_enhanced['low'])
rule_e4 = ctrl.Rule(soil_moisture_enhanced['dry'] & weather_enhanced['sunny'], irrigation_enhanced['high'])
rule_e5 = ctrl.Rule(soil_moisture_enhanced['dry'] & weather_enhanced['cloudy'], irrigation_enhanced['medium'])
rule_e6 = ctrl.Rule(soil_moisture_enhanced['dry'] & weather_enhanced['rainy'], irrigation_enhanced['very_low'])
rule_e7 = ctrl.Rule(soil_moisture_enhanced['medium'] & weather_enhanced['sunny'], irrigation_enhanced['medium'])
rule_e8 = ctrl.Rule(soil_moisture_enhanced['medium'] & weather_enhanced['cloudy'], irrigation_enhanced['low'])
rule_e9 = ctrl.Rule(soil_moisture_enhanced['medium'] & weather_enhanced['rainy'], irrigation_enhanced['no_water'])
rule_e10 = ctrl.Rule(soil_moisture_enhanced['moist'] & weather_enhanced['sunny'], irrigation_enhanced['low'])
rule_e11 = ctrl.Rule(soil_moisture_enhanced['moist'] & weather_enhanced['cloudy'], irrigation_enhanced['very_low'])
rule_e12 = ctrl.Rule(soil_moisture_enhanced['moist'] & weather_enhanced['rainy'], irrigation_enhanced['no_water'])
rule_e13 = ctrl.Rule(soil_moisture_enhanced['saturated'] & weather_enhanced['sunny'], irrigation_enhanced['no_water'])
rule_e14 = ctrl.Rule(soil_moisture_enhanced['saturated'] & weather_enhanced['cloudy'], irrigation_enhanced['no_water'])
rule_e15 = ctrl.Rule(soil_moisture_enhanced['saturated'] & weather_enhanced['rainy'], irrigation_enhanced['no_water'])

# Collect the individual rules into the rules_enhanced list
rules_enhanced = [
    rule_e1, rule_e2, rule_e3, rule_e4, rule_e5, rule_e6,
    rule_e7, rule_e8, rule_e9, rule_e10, rule_e11, rule_e12,
    rule_e13, rule_e14, rule_e15
]

# Create Control System for Enhanced System
irrigation_ctrl_enhanced = ctrl.ControlSystem(rules_enhanced)
irrigation_simulation_enhanced = ctrl.ControlSystemSimulation(irrigation_ctrl_enhanced)
```

ساختار قوانین:

قوانین به صورت ترکیبی از دو ورودی (سطح رطوبت خاک و نوع آبوهوا) و یک خروجی (میزان آبیاری مناسب) تعریف شده‌اند. هدف از طراحی این قوانین، افزایش دقت پاسخ‌دهی سیستم با در نظر گرفتن حالات مختلف محیطی با جزئیات بیشتر نسبت به نسخه ساده قبلی است.

- در رطوبت‌های بسیار کم (very_dry)، حتی در شرایط بارانی نیز میزان آبیاری در حداقل ممکن لحاظ شده (اما حذف نشده) تا کمبود رطوبت خاک جبران شود.
- در رطوبت‌های اشباع (saturated)، فارغ از وضعیت آبوهوا، خروجی آبیاری همیشه no_water است، که منطقاً درست است.
- تطابق تدریجی بین شرایط و خروجی به وضوح در قوانین رعایت شده است؛ برای مثال:
 - از very_high تا very_low برای حالت‌های خشک بسته به نوع آبوهوا.
 - از medium تا no_water برای خاک متوسط.
 - استفاده از درجات ملایم‌تر مثل very_low برای انتقال نرم بین مقادیر آبیاری.

نتیجه:

این مجموعه قوانین باعث می‌شود رفتار سیستم فازی نرم‌تر و واکنش‌پذیرتر نسبت به شرایط دقیق خاک و هوا باشد. همچنین می‌توان انتظار داشت که نسبت به نسخه اولیه، خروجی‌ها طبیعی‌تر و بهتر منطبق با نیازهای واقعی گیاهان باشند.

Simulation and Analysis

- Comparing Original 9-Rule vs Enhanced 15-Rule

Simulation Parameters

```
initial_moisture = 25
days = 10
weather_sequence_continuous = np.array([0.5, 1.5, 4.0, 6.0, 8.5, 9.5, 7.0, 3.0, 1.0, 5.5])
weather_points = [0.0, 5.0, 10.0]
effect_points = [-5.0, -2.0, 5.0]
optimal_range_lower = 35
optimal_range_upper = 65
```

```
# Calculates the effect of weather on soil moisture using interpolation.
def calculate_weather_effect(weather_value):
    return np.interp(weather_value, weather_points, effect_points)
```

برای ارزیابی عملکرد دو سیستم فازی با ۹ و ۱۵ قانون، یک شبیه‌سازی ۱۰ روزه با رطوبت اولیه خاک برابر با ۲۵٪ انجام شد. در این شبیه‌سازی، تأثیر آب‌وهوا به‌صورت پیوسته و با استفاده از درون‌یابی بین سه نقطه‌ی کلیدی (آفتابی، ابری، بارانی) بر رطوبت خاک اعمال گردید. همچنین محدوده مطلوب رطوبت خاک بین ۳۵٪ تا ۶۵٪ در نظر گرفته شد تا عملکرد کنترلی سیستم‌ها در حفظ این بازه سنجیده شود.

عملکرد تابع `calculate_weather_effect`:

این تابع مقدار اثر وضعیت آب‌وهوا را با استفاده از درون‌یابی خطی بین نقاط مرجع محاسبه می‌کند:

- آفتابی (۰) → بیشترین کاهش رطوبت (-5.0)

- ابری (۵) → کاهش کمتر (-2.0)

- بارانی (۱۰) → افزایش رطوبت (+5.0)

بنابراین، با مقداردهی پیوسته به آب‌وهوا (مثلاً 6.7)، سیستم قادر به شبیه‌سازی رفتار محیط واقعی با دقت بیشتری است.

```
moisture_levels_original = [initial_moisture]
irrigation_amounts_original = []
current_moisture = initial_moisture

print(f"Day 0: Initial Moisture = {current_moisture:.2f}%")
print("-" * 70)
print(f"{'Day':>3} | {'Weather Val':>11} | {'Moisture In':>12} | {'Irrigation':>10} | {'Weather Effect':>15} | {'Moisture Out':>13}")
print("-" * 70)

for day in range(days):
    current_weather_continuous = weather_sequence_continuous[day]
    moisture_before_irrigation = current_moisture

    irrigation_simulation_original.input['soil_moisture_orig'] = current_moisture
    irrigation_simulation_original.input['weather_orig'] = current_weather_continuous

    irrigation_simulation_original.compute()
    irrigation_output = irrigation_simulation_original.output['irrigation_centroid']

    irrigation_amounts_original.append(irrigation_output)
    weather_effect_value = calculate_weather_effect(current_weather_continuous)

    current_moisture = current_moisture + (0.6 * irrigation_output) + weather_effect_value
    current_moisture = np.clip(current_moisture, 0, 100)
    moisture_levels_original.append(current_moisture)

    print(f"{'day + 1':>3} | {'current_weather_continuous':>11.1f} | {'moisture_before_irrigation':>12.2f} | {'irrigation_output':>10.2f} | "
          f"{'weather_effect_value':>15.1f} | {'current_moisture':>13.2f}")

print("-" * 70)
```

در این قسمت از شبیه‌سازی، رطوبت خاک برای هر روز با استفاده از ورودی‌های سیستم فازی (رطوبت خاک و وضعیت آب‌وهوا) محاسبه می‌شود. در ابتدا، رطوبت اولیه خاک برابر با ۲۵٪ تنظیم می‌شود و سپس برای هر روز شبیه‌سازی، مقدار آبیاری بر اساس وضعیت آب‌وهوا و رطوبت خاک محاسبه می‌گردد. در نهایت، اثر وضعیت آب‌وهوا بر تغییرات رطوبت خاک (با استفاده از تابع درون‌یابی) محاسبه شده و رطوبت خاک بعد از آبیاری و اثرات آب‌وهوا به‌روز رسانی می‌شود.

نتایج شبیه‌سازی:

برای هر روز شبیه‌سازی، مقادیر زیر محاسبه و گزارش می‌شوند:

- مقدار آب‌وهوا: مقدار پیوسته وضعیت آب‌وهوا (آفتابی، ابری، بارانی) برای هر روز
 - رطوبت قبل از آبیاری: میزان رطوبت خاک قبل از اعمال آبیاری
 - مقدار آبیاری: میزان آب اضافه شده به خاک از سیستم فازی
 - اثر آب‌وهوا: تغییرات رطوبت خاک ناشی از وضعیت آب‌وهوا
 - رطوبت نهایی خاک: میزان رطوبت خاک پس از آبیاری و اثر آب‌وهوا
- این فرایند برای ۱۰ روز متوالی انجام می‌شود و رطوبت خاک برای هر روز محاسبه و به‌روز رسانی می‌گردد.

نتیجه شبیه‌سازی سیستم 9 قانونه :

Day 0: Initial Moisture = 25.00%

Day	Weather Val	Moisture In	Irrigation	Weather Effect	Moisture Out
1	0.5	25.00	7.85	-4.7	25.01
2	1.5	25.01	7.85	-4.1	25.62
3	4.0	25.62	5.08	-2.6	26.07
4	6.0	26.07	5.02	-0.6	28.48
5	8.5	28.48	2.50	+2.9	32.88
6	9.5	32.88	2.22	+4.3	38.51
7	7.0	38.51	3.72	+0.8	41.54
8	3.0	41.54	5.08	-3.2	41.39
9	1.0	41.39	6.50	-4.4	40.89
10	5.5	40.89	3.78	-1.3	41.86

نتیجه شبیه سازی سیستم 15 قانونه :

Day 0: Initial Moisture = 25.00%

Day	Weather Val	Moisture In	Irrigation	Weather Effect	Moisture Out
1	0.5	25.00	7.88	-4.7	25.81
2	1.5	25.81	7.84	-4.1	27.21
3	4.0	27.21	5.94	-2.6	28.76
4	6.0	28.76	5.78	-0.6	32.21
5	8.5	32.21	1.81	+2.9	36.38
6	9.5	36.38	1.59	+4.3	41.79
7	7.0	41.79	3.63	+0.8	45.12
8	3.0	45.12	5.36	-3.2	45.67
9	1.0	45.67	6.19	-4.4	45.61
10	5.5	45.61	4.19	-1.3	47.24

مقایسه و تحلیل نتایج شبیه سازی برای سیستم های 9 Rule , 15 Rule :

سیستم 9 Rule:

1. رطوبت خاک در ابتدای شبیه سازی:

○ هر دو سیستم با رطوبت خاک 25٪ شروع می کنند.

2. مقدار آبیاری:

○ در سیستم 9 Rule مقدار آبیاری در روزهای ابتدایی به طور مداوم در حال کاهش است. از 7.85 واحد در روز

اول به 2.22 واحد در روز ششم می رسد.

○ مقدار آبیاری در روزهای اول نسبت به روزهای آخر بیشتر است. در این سیستم، آبیاری در روزهای با وضعیت

آفتابی (مثلاً روزهای 1، 2 و 4) بالاتر است، اما در روزهای با وضعیت ابری یا بارانی (مثلاً روزهای 5 و 9) مقدار

آبیاری کاهش می یابد.

3. اثر وضعیت آب و هوا:

○ اثر وضعیت آب و هوا در سیستم 9 Rule به طور متوسط منفی است. به عنوان مثال، در روزهای با وضعیت

آب و هوا آفتابی، اثر منفی بیشتری مشاهده می شود.

- این سیستم به دلیل استفاده از تعداد محدود قاعده‌ها، نمی‌تواند به‌طور دقیق‌تر واکنش نشان دهد و به همین دلیل بعضی تغییرات وضعیت آب‌وهوا را کمتر در نظر می‌گیرد.

4. رطوبت خاک نهایی:

- رطوبت خاک در پایان روز دهم در سیستم 9 Rule به 41.86٪ می‌رسد که نشان‌دهنده افزایش تدریجی رطوبت خاک است.

سیستم 15 Rule:

1. رطوبت خاک در ابتدای شبیه‌سازی:

- همانطور که در سیستم اول، رطوبت خاک در ابتدای شبیه‌سازی 25٪ است.

2. مقدار آبیاری:

- در سیستم 15 Rule ، مقدار آبیاری در مقایسه با سیستم 9 Rule به طور کلی بیشتر است. برای مثال، در روز اول، مقدار آبیاری 7.88 واحد است که کمی بیشتر از 7.85 واحد سیستم 9 Rule است.

- در ادامه، این سیستم آبیاری بیشتری در روزهای مختلف می‌دهد و مقدار آبیاری در روزهای بعدی، به‌ویژه روزهای با وضعیت آفتابی و ابری، کمی بیشتر از سیستم 9 Rule است.

3. اثر وضعیت آب‌وهوا:

- اثر وضعیت آب‌وهوا در سیستم 15 Rule طور مشابه با سیستم 9 Rule منفی است، اما به دلیل استفاده از 15 قاعده، این سیستم قادر است به واکنش‌های پیچیده‌تری در برابر تغییرات وضعیت آب‌وهوا پاسخ دهد.
- این سیستم با داشتن قواعد بیشتر، در تطابق با وضعیت‌های مختلف، به‌ویژه در روزهای با وضعیت آفتابی یا بارانی، بهتر عمل می‌کند.

4. رطوبت خاک نهایی:

- رطوبت خاک در پایان روز دهم در سیستم 15 Rule به 47.24٪ می‌رسد که این نشان‌دهنده بهبود تدریجی رطوبت خاک نسبت به سیستم 9 Rule است.

نتیجه‌گیری کلی:

سیستم 15 Rule نسبت به سیستم 9 Rule عملکرد بهتری دارد و رطوبت خاک بالاتری را در پایان دوره شبیه‌سازی حفظ می‌کند.

آبیاری در سیستم 15 Rule بیشتر است، که نشان‌دهنده واکنش دقیق‌تر به تغییرات وضعیت آب‌وهوا است.

سیستم 9 Rule با تعداد قاعده‌های کمتری که دارد، ممکن است در برخی شرایط نتواند به خوبی به وضعیت‌های پیچیده آب‌وهوا واکنش نشان دهد، در حالی که سیستم 15 Rule با داشتن قواعد بیشتر، می‌تواند تصمیمات دقیق‌تری در زمینه آبیاری بگیرد.

Comparison Metrics (9 Rules vs 15 Rules):

Metric	Original (9R)	Enhanced (15R)
Avg Moisture (%)	34.22	37.58
Total Irrigation (unit)	49.60	50.20
Days in Optimal	5	6
Percent in Optimal (%)	50.0	60.0

- میانگین رطوبت: سیستم 15 قاعده‌ای میانگین رطوبت بالاتری را حفظ کرده است (+3.36٪)، که نشان‌دهنده آبیاری مؤثرتر و کنترل بهتر رطوبت در طول دوره است.
 - مجموع آبیاری: هر دو سیستم تقریباً مقدار مشابهی آبیاری انجام داده‌اند، اما سیستم 15 قاعده‌ای با مصرف تنها 0.6 واحد بیشتر، عملکرد بهتری در حفظ رطوبت داشته است. این نشان می‌دهد که مدیریت منابع در این سیستم بهینه‌تر بوده است.
 - روزهای در محدوده بهینه: سیستم پیشرفته یک روز بیشتر را در محدوده بهینه رطوبت گذرانده است، که از نظر سلامت گیاه و کارایی سیستم اهمیت دارد.
 - درصد حضور در محدوده بهینه: سیستم 15 قاعده‌ای 60٪ از روزها را در محدوده بهینه نگه داشته، در حالی که سیستم 9 قاعده‌ای فقط 50٪ موفق به این کار شده است.
- نتیجه‌گیری کلی:

سیستم 15 قاعده‌ای با تنها اندکی افزایش در مصرف آب، توانسته عملکرد بهتری از نظر حفظ رطوبت خاک و قرارگیری در محدوده بهینه داشته باشد. این نتایج بیانگر دقت بالاتر و تطابق بهتر این سیستم با شرایط متغیر محیطی است.

نمایش بصری سیستم‌های 9 Rule , 15 Rule :

Fuzzy Control System Performance Comparison (9 vs 15 Rules)

