



دانشگاه اصفهان
دانشکده مهندسی کامپیوتر

هوش محاسباتی
گزارش کتبی تمرین دوم
بخش دوم: کنترل کننده منطق فازی برای سیستم آبیاری خودکار گیاهان

اعضای گروه:

رضا چراخ

فرناز موحدی

اردیبهشت ۱۴۰۴

فهرست مطالب

عنوان	صفحه
فصل اول توابع عضویت	2
۱-۱- مقدمه	2
۲-۱- توابع عضویت فازی زیر را با استفاده از زبان پایتون به طور واضح و دقیق تعریف و پیاده‌سازی کنید.	2
۱-۲-۱- ورودی-سطح رطوبت خاک: خشک، متوسط، مرطوب	2
۲-۲-۱- ورودی-شرایط آب و هوا: آفتابی، ابری، بارانی	3
۳-۲-۱- خروجی-مقدار آبیاری: بدون آب، کم، متوسط، زیاد	4
۳-۱- توابع عضویت را ترسیم و بصری‌سازی کنید.	4
فصل دوم قواعد کنترلی و غیرفازی‌سازی	6
۱-۲- مقدمه	6
۲-۲- قواعد منطق فازی زیر را پیاده‌سازی کنید.	6
۳-۲- با استفاده از روش مرکز ثقل (Centroid)، خروجی فازی را به یک مقدار واضح (Crisp) برای میزان آب تبدیل کنید.	8
۴-۲- حداقل با سه روش دیگر یا بیشتر، خروجی فازی را به یک مقدار واضح برای میزان آب تبدیل کنید. سپس نتایج را برای هر یک مقایسه کنید. (اختیاری).	8
فصل سوم شبیه‌سازی و تحلیل	9
۱-۳- مقدمه	9
۲-۳- کنترل‌کننده فازی طراحی‌شده را در یک بازه زمانی با شرایط متغیر آب‌وهوایی شبیه‌سازی کنید. در هر گام زمانی، رطوبت خاک را بر اساس تصمیم کنترل‌کننده به روز کنید.	9
۳-۳- ترسیم نمودار و بررسی آن	11
۱-۳-۳- تغییرات رطوبت خاک در طول زمان	11
۲-۳-۳- میزان آب دهی تعیین شده توسط کنترل‌کننده فازی در طول زمان	12
۴-۳- نتایج را تحلیل و بررسی کنید. کنترل‌کننده فازی طراحی‌شده تا چه حد در حفظ سطح بهینه رطوبت خاک موفق است؟	13
منابع	14

فصل اول

توابع عضویت

۱-۱- مقدمه

سیستم‌های آبیاری خودکار می‌توانند با در نظر گرفتن شرایط مختلف محیطی مانند رطوبت خاک و وضعیت آب و هوا، مقدار مناسب آب را به گیاهان ارائه دهند. کنترل‌کننده‌های مبتنی بر منطق فازی، با استفاده از استدلال تقریبی به جای آستانه‌های دقیق، برای مدیریت چنین سیستم‌هایی و تصمیم‌گیری در شرایط عدم قطعیت مناسب هستند. فصل اول گزارش، به طراحی و پیاده‌سازی توابع عضویت مربوط به ورودی‌ها و خروجی‌ها می‌پردازد.

۱-۲- توابع عضویت فازی زیر را با استفاده از زبان پایتون به طور واضح و دقیق تعریف و پیاده‌سازی کنید.

۱-۲-۱- ورودی-سطح رطوبت خاک: خشک، متوسط، مرطوب

رطوبت خاک در این پروژه به صورت یک متغیر پیوسته در بازه ۰ تا ۱۰۰ درصد در نظر گرفته شده است. این متغیر به سه مجموعه فازی تقسیم شده که به شکل توابع مثلثی (Triangular) تعریف شده‌اند:

• خشک: (Dry)

تابع عضویت Dry در بازه‌ی [0, 0, 30] تعریف شده است. این تابع در مقدار رطوبت 0 دارای بیشترین درجه عضویت (1) بوده و با افزایش رطوبت تا مقدار 30، عضویت آن به صفر می‌رسد. این ناحیه نشان‌دهنده خاک کاملاً خشک است که نیاز فوری به آبیاری دارد.

• متوسط: (Medium)

تابع عضویت Medium در بازه‌ی [25, 40, 55] تعریف شده است. بیشینه عضویت در نقطه 40 اتفاق می‌افتد و دو طرف آن به تدریج کاهش می‌یابد. این ناحیه بیانگر وضعیت بهینه

رطوبت خاک برای گیاهان معمولی است که نیاز به آبیاری متوسط دارد.

- **مرطوب: (Wet)**

تابع Wet در بازه $[50, 65, 100]$ تعریف شده و بیانگر رطوبت بالا در خاک است. در این حالت از رطوبت (به‌ویژه بالای 65 درصد)، خاک مرطوب محسوب می‌شود و سیستم نباید آبیاری انجام دهد.

```
# Soil Moisture (0-100%) - Best is 20-60% range
soil_moisture = np.arange(0, 101, 1)
soil_moisture_dry = fuzz.trimf(soil_moisture, [0, 0, 30])
soil_moisture_medium = fuzz.trimf(soil_moisture, [25, 40, 55])
soil_moisture_wet = fuzz.trimf(soil_moisture, [50, 65, 100])
```

۱-۲-۲- ورودی-شرایط آب و هوا: آفتابی، ابری، بارانی

برای ساده‌سازی مدل، شرایط آب‌وهوا به‌صورت یک متغیر گسسته با سه مقدار ممکن در نظر گرفته شده که با توابع مثلثی فازی‌سازی شده‌اند:

- **آفتابی: (Sunny)**

تابع Sunny در بازه $[0, 0, 1]$ تعریف شده و برای مقدار صفر (کاملاً آفتابی) بیشترین عضویت را دارد. این حالت نمایانگر روزهایی با تبخیر بالا است که نیاز به آبیاری بیشتر را ایجاد می‌کند.

- **ابری: (Cloudy)**

تابع Cloudy در بازه $[0, 1, 2]$ قرار دارد و بیشترین عضویت در مقدار 1 است. این وضعیت تأثیر متوسطی بر رطوبت خاک دارد.

- **بارانی: (Rainy)**

تابع Rainy در بازه $[1, 2, 2]$ تعریف شده است و بیانگر شرایطی است که در آن بارش باران اتفاق می‌افتد. در این حالت، خاک به‌صورت طبیعی مرطوب شده و نیازی به آبیاری از سوی سیستم وجود ندارد.

```
# Weather (0: Sunny, 1: Cloudy, 2: Rainy)
weather = np.arange(0, 3, 0.01)
weather_sunny = fuzz.trimf(weather, [0, 0, 1])
weather_cloudy = fuzz.trimf(weather, [0, 1, 2])
weather_rainy = fuzz.trimf(weather, [1, 2, 2])
```

۱-۲-۳-مقدار آبیاری: بدون آب، کم، متوسط، زیاد

خروجی سیستم فازی نیز به صورت یک متغیر پیوسته بین ۰ تا ۱۰۰ تعریف شده است. این متغیر نشان‌دهنده شدت نسبی آبیاری (نه مقدار واقعی آب) است و به چهار مجموعه فازی تقسیم شده:

• None

تابع None در بازه $[0, 0, 10]$ قرار دارد و برای مواقعی استفاده می‌شود که خاک کاملاً مرطوب بوده یا شرایط آب‌وهوا بارانی است. این تابع خروجی صفر یا بسیار کم تولید می‌کند.

• Low

تابع Low در بازه $[5, 25, 45]$ تعریف شده و بیانگر آبیاری سبک برای شرایط نسبتاً مرطوب یا ابری است.

• Medium

تابع Medium در بازه $[40, 60, 80]$ قرار دارد و برای شرایط متوسط (مثل رطوبت متوسط در هوای آفتابی) استفاده می‌شود.

• High

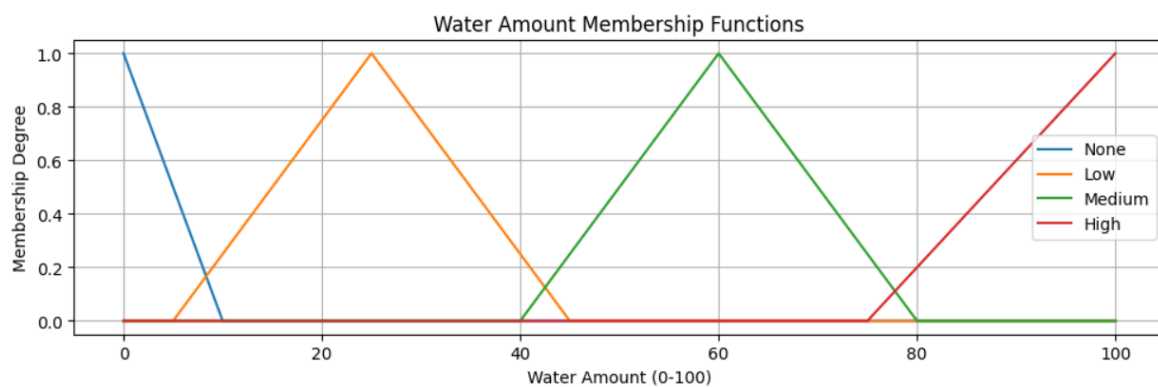
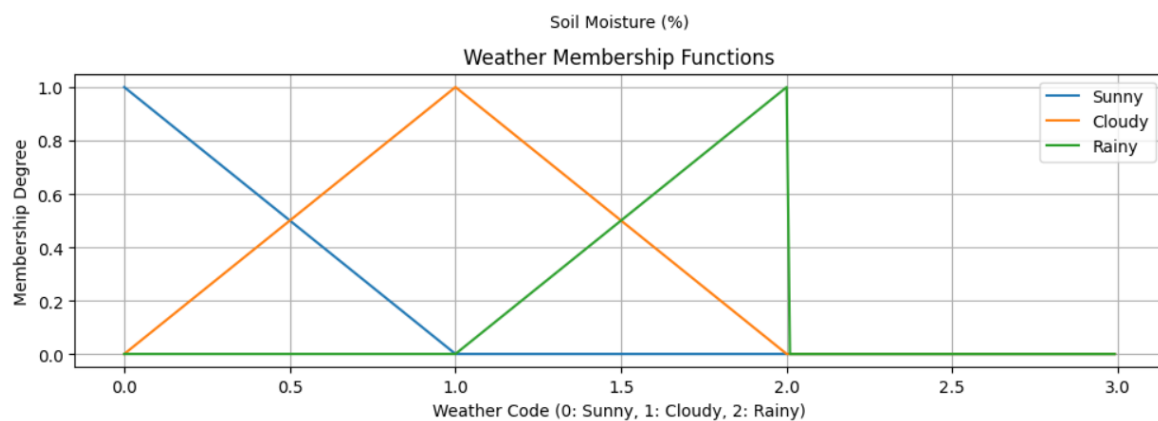
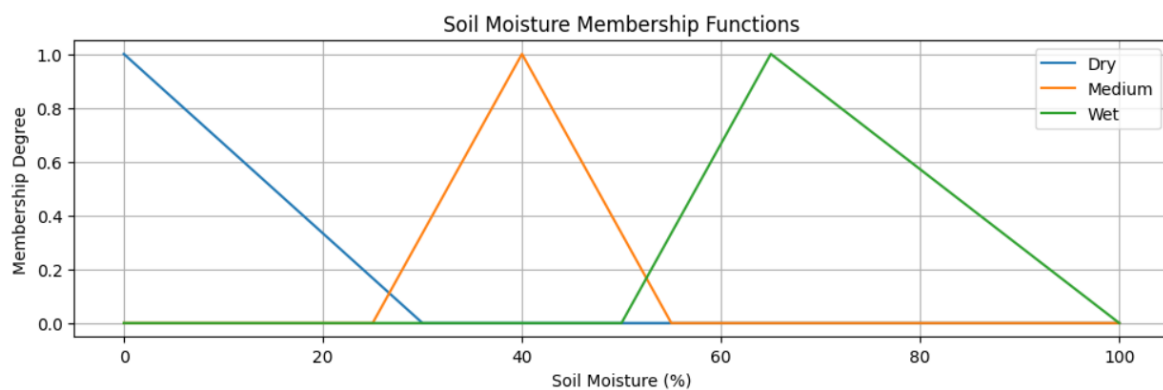
تابع High در بازه $[75, 100, 100]$ تعریف شده و برای شرایط بحرانی که خاک خشک و هوا آفتابی است استفاده می‌شود. این تابع منجر به بیشترین میزان آبیاری خواهد شد.

```
# Water Amount (0-100 units)
water = np.arange(0, 101, 1)
water_none = fuzz.trimf(water, [0, 0, 10])
water_low = fuzz.trimf(water, [5, 25, 45])
water_medium = fuzz.trimf(water, [40, 60, 80])
water_high = fuzz.trimf(water, [75, 100, 100])
```

۱-۳-توابع عضویت را ترسیم و بصری‌سازی کنید.

تمام توابع عضویت به صورت مثلثی (Triangular) تعریف شده‌اند زیرا:

- محاسبه ساده دارند.
- پیاده‌سازی آن‌ها در محیط فازی سریع و مؤثر است.
- برای اغلب کاربردهای کنترل فازی (مثل آبیاری خودکار)، کفایت می‌کنند.



فصل دوم

قواعد کنترلی و غیرفازی سازی

۲-۱- مقدمه

فصل دوم گزارش، به پیاده سازی قواعد کنترلی در پایتون و همچنین غیرفازی سازی آنها جهت خودکارسازی فرآیند آبیاری گیاهان بر اساس میزان رطوبت خاک و شرایط آب و هوایی می پردازد.

۲-۲- قواعد منطق فازی زیر را پیاده سازی کنید.

اگر خاک خشک است و هوا آفتابی است، آبیاری زیاد باشد.

اگر خاک خشک و هوا ابری است، آب دهی متوسط باشد.

اگر خاک خشک و هوا بارانی است، آب دهی کم.

اگر خاک متوسط و هوا آفتابی است، مقدار آب متوسط.

اگر خاک متوسط و هوا ابری است، مقدار آب کم.

اگر خاک متوسط و هوا بارانی است، نیازی به آب نیست.

اگر خاک مرطوب و هوا آفتابی است، مقدار آب کم.

اگر خاک مرطوب و هوا ابری است، نیازی به آب نیست.

اگر خاک مرطوب و هوا بارانی است، نیازی به آب نیست.

این قوانین در تابع `fuzzy_inference()` پیاده سازی شده اند. این تابع با دریافت دو مقدار ورودی (رطوبت خاک و وضعیت آب و هوا)، تصمیم می گیرد که چه میزان آب به گیاه داده شود.

```
def fuzzy_inference(soil_value, weather_value):
    # Fuzzification
    dry = fuzz.interp_membership(soil_moisture, soil_moisture_dry, soil_value)
    medium = fuzz.interp_membership(soil_moisture, soil_moisture_medium, soil_value)
    wet = fuzz.interp_membership(soil_moisture, soil_moisture_wet, soil_value)

    sunny = fuzz.interp_membership(weather, weather_sunny, weather_value)
    cloudy = fuzz.interp_membership(weather, weather_cloudy, weather_value)
    rainy = fuzz.interp_membership(weather, weather_rainy, weather_value)

    # Rules
    rule1 = np.fmin(dry, sunny) # High
    rule2 = np.fmin(dry, cloudy) # Medium
    rule3 = np.fmin(dry, rainy) # Low
    rule4 = np.fmin(medium, sunny) # Medium
    rule5 = np.fmin(medium, cloudy) # Low
    rule6 = np.fmin(medium, rainy) # Low
    rule7 = np.fmin(wet, sunny) # None
    rule8 = np.fmin(wet, cloudy) # None
    rule9 = np.fmin(wet, rainy) # None
```

در بخش اول این تابع (fuzzification)، مقادیر واقعی رطوبت خاک یا شرایط آب و هوایی به درجه عضویت در هر یک از نمودارهای مربوط به ورودی‌ها تبدیل می‌شود. برای مثال مقدار واقعی رطوبت خاک (مثلاً ۴۲٪) به درجه عضویت در Dry، Medium و Wet تبدیل می‌شود. تابع interp_membership بررسی می‌کند که عدد ورودی چقدر در هر تابع عضویت صدق می‌کند.

در بخش دوم این تابع (rules)، قوانین نوشته می‌شوند. تا اینجای کار ورودی‌ها مدل شده‌اند. تابع np.fmin مینیمم بین دو درجه عضویت را می‌گیرد (منطق AND فازی).

```
# Aggregation
activation_high = np.fmax(rule1, np.zeros_like(water))
activation_medium = np.fmax(rule2, rule4)
activation_low = np.fmax(rule3, np.fmax(rule5, rule6))
activation_none = np.fmax(rule7, np.fmax(rule8, rule9))

aggregated = np.fmax(
    np.fmax(activation_none * water_none, activation_low * water_low),
    np.fmax(activation_medium * water_medium, activation_high * water_high)
)
```

در بخش نهایی این تابع (aggregation)، قواعدی که نتیجه‌شان مشابه است (منظور از نتیجه همان سطح آبیاری است)، با هم تجمیع می‌شوند. مثلاً اگر Rule 2 و Rule 4 هر دو منجر به "Medium" شوند، با استفاده از fmax ترکیب می‌شوند.

در نهایت خروجی‌های فازی از چهار دسته (None, Low, Medium, High) با وزن‌گیری حاصل از قواعد قبلی ترکیب می‌شوند و یک تابع خروجی فازی ترکیبی ایجاد می‌کنند.

۲-۳- با استفاده از روش مرکز ثقل (Centroid)، خروجی فازی را به یک مقدار واضح (Crisp) برای میزان آب تبدیل کنید.

```
output = fuzz.defuzz(water, aggregated, 'centroid')
return output, active_rule
```

تابع defuzz با روش مرکز ثقل (centroid) خروجی فازی را به یک مقدار عددی دقیق تبدیل می‌کند که همان میزان نهایی آبیاری است (مثلاً 42.7 واحد).

خروجی‌های تابع fuzzy_inference():

- Output: مقدار عددی آبیاری (واقعی)
- Active_rule: شماره قانون فعال شده با بیشترین اثر (برای تحلیل)

۲-۴- حداقل با سه روش دیگر یا بیشتر، خروجی فازی را به یک مقدار واضح برای میزان آب تبدیل کنید. سپس نتایج را برای هر یک مقایسه کنید. (اختیاری)

سه روش دیگر انتخاب شده شامل Bisector، MoM و SoM است.

- Bisector: نقطه‌ای که مساحت زیر تابع را به دو قسمت مساوی تقسیم کند.
- MoM: میانگین تمام نقاطی که بیشترین درجه عضویت را دارند.
- SoM: کم‌ترین مقدار x که بیشترین درجه عضویت را دارد.

Bisector تعادل بین دو طرف را برقرار می‌کند اما ممکن است رفتار نامنظم داشته باشد. MoM ساده و سریع است اما دقت کافی را ندارد. SoM برای تصمیمات محتاطانه بهتر است اما ممکن است مقدار حداقلی بدهد. در نهایت Centroid دقیق و استاندارد است. برخی به دلیل محاسبات سنگین تر آن این روش را کنار می‌گذارند. یک نمونه خروجی:

```
-----
Day 6:
Weather: Rainy 🌧️
Soil Moisture Before: 52.14%
Activated Rule: Rule 6
Defuzzified Outputs:
- Centroid: 21.81
- Mean of Max (MoM): 25.00
- Smallest of Max (SoM): 25.00
- Bisector: 23.24
Water Given (simulation): 21.81 units
-----
Day 7:
Weather: Rainy 🌧️
Soil Moisture Before: 57.97%
Activated Rule: Rule 9
Defuzzified Outputs:
- Centroid: 3.33
- Mean of Max (MoM): 0.00
- Smallest of Max (SoM): 0.00
- Bisector: 2.93
Water Given (simulation): 3.33 units
```

فصل سوم شبیه‌سازی و تحلیل

۳-۱- مقدمه

فصل سوم گزارش، به طراحی و پیاده‌سازی یک شبیه‌سازی ده روزه برای کنترل‌کننده منطق فازی می‌پردازد. در انتهای فصل هم به تحلیل و بررسی کنترل‌کننده فازی با استفاده از نمودارهای مربوطه خواهیم پرداخت.

۳-۲- کنترل‌کننده فازی طراحی‌شده را در یک بازه زمانی با شرایط متغیر آب‌وهوایی شبیه‌سازی کنید. در هر گام زمانی، رطوبت خاک را بر اساس تصمیم کنترل‌کننده به روز کنید.

شبیه‌سازی در تابع `time_simulation()` انجام شده است. این تابع مسئول شبیه‌سازی تغییرات رطوبت خاک در طی ده روز با توجه به شرایط آب‌وهوایی و تصمیم‌های سیستم فازی برای آبیاری است. در شرایط واقعی بهترین درصد رطوبت خاک بین ۲۰ تا ۶۰ درصد است و سیستم فازی در تلاش است که رطوبت را در این بازه حفظ کند. مقدار اولیه رطوبت خاک برای اینکه وسط این بازه باشد، مقدار ۴۰ تعریف شده است.

```
def time_simulation(initial_soil_moisture, weather_list):
    efficiency = 0.8
    max_effective_water = 10 # maximum real-world effect of water on soil moisture
    moisture_log = [initial_soil_moisture]
    water_log = []
    rule_log = []
    soil = initial_soil_moisture

    for day, weather_today in enumerate(weather_list):
        water_amount, rule_used = fuzzy_inference(soil, weather_today)

        # Weather effect
        if weather_today == 0: effect = -5
        elif weather_today == 1: effect = -2
        elif weather_today == 2: effect = +5

        # Normalize fuzzy output before applying to soil
        water_amount_norm = (water_amount / 100) * max_effective_water
        gain = water_amount_norm * (1 - (soil / 100)) * efficiency
        soil = max(0, min(100, soil + gain + effect))

        moisture_log.append(soil)
        water_log.append(water_amount)
        rule_log.append(rule_used)

    return moisture_log, water_log, rule_log
```

efficiency: بازدهی جذب آب در خاک (یعنی چند درصد از آب واقعاً جذب می‌شود).

max_effective_water: حداکثر اثری که سیستم آبیاری می‌تواند روی رطوبت خاک بگذارد.

این تابع چند مرحله اصلی دارد:

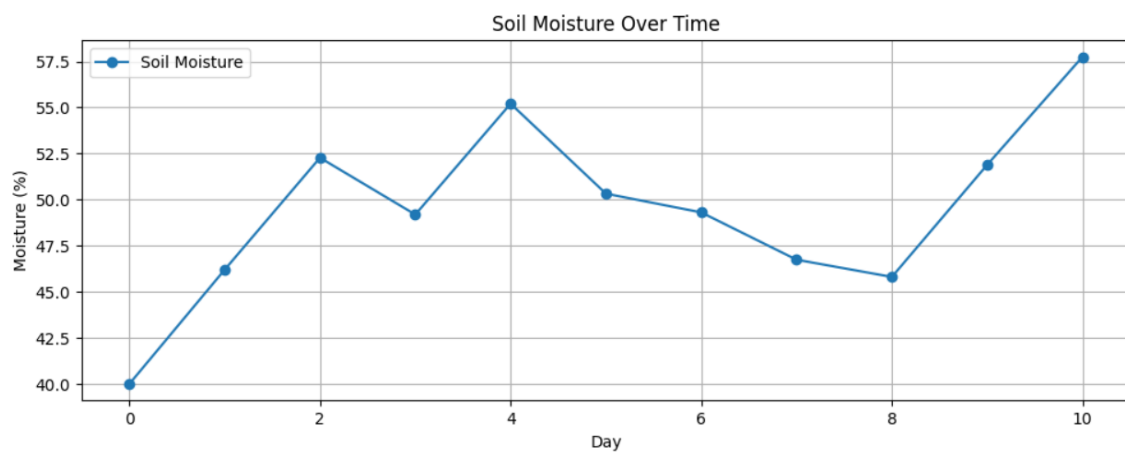
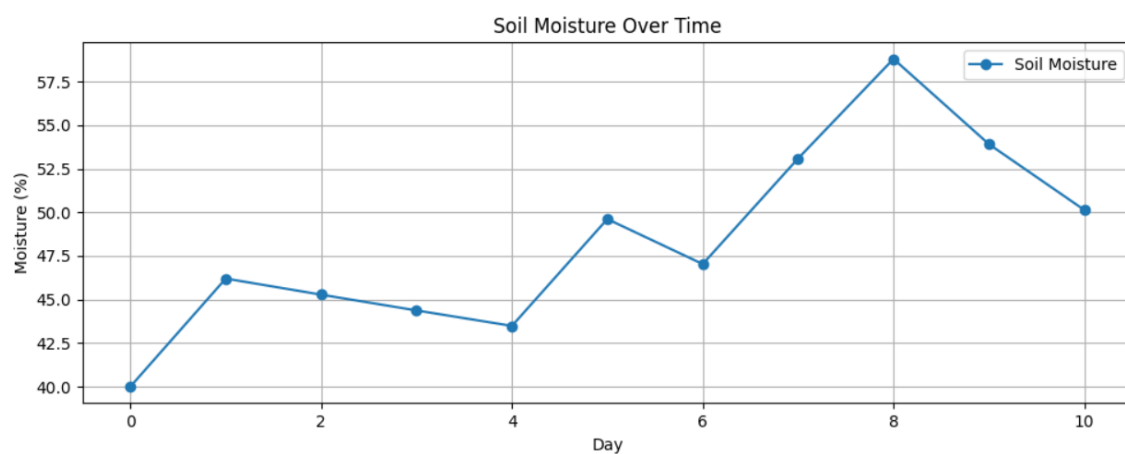
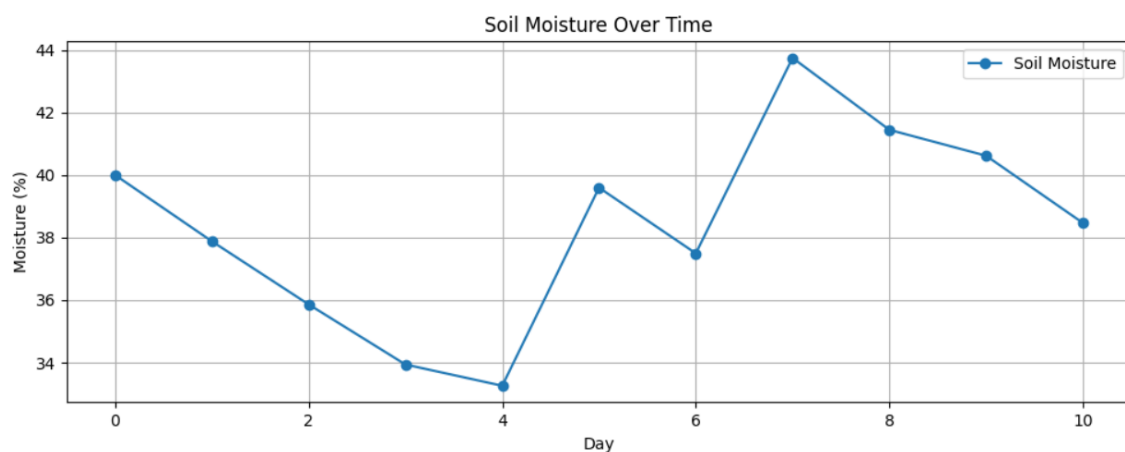
- **مرحله ۱: تصمیم‌گیری فازی:** سیستم فازی بر اساس رطوبت فعلی و وضعیت هوا، میزان آب‌دهی (water_amount) و قانون فعال را برمی‌گرداند.
- **مرحله ۲: تأثیر شرایط آب‌وهوا:** تأثیر این شرایط به گونه‌ای است که برای هوای آفتابی ۵ درصد کاهش، برای هوای ابری ۲ درصد کاهش و برای هوای بارانی ۵ درصد افزایش رطوبت داریم.
- **مرحله ۳: اعمال تأثیر آبیاری بر رطوبت خاک:** ابتدا خروجی فازی نرمال‌سازی می‌شود. water_amount_norm: مقدار آب‌دهی فازی به بازه واقعی (0 تا 10) نرمال می‌شود. gain: مقدار واقعی‌ای که رطوبت خاک افزایش می‌یابد. (گر خاک خیلی مرطوب باشد، این مقدار کم خواهد بود. اگر خاک خشک باشد، مقدار بیشتری از آب جذب خواهد شد).
- **مرحله ۴: بروزرسانی رطوبت خاک:** اثر آبیاری و شرایط آب‌وهوا با هم جمع می‌شوند و سپس این مقدار به بازه [0, 100] محدود می‌شود.

در نهایت هم نتایج روزانه ذخیره می‌شوند.

۳-۳- ترسیم نمودار و بررسی آن

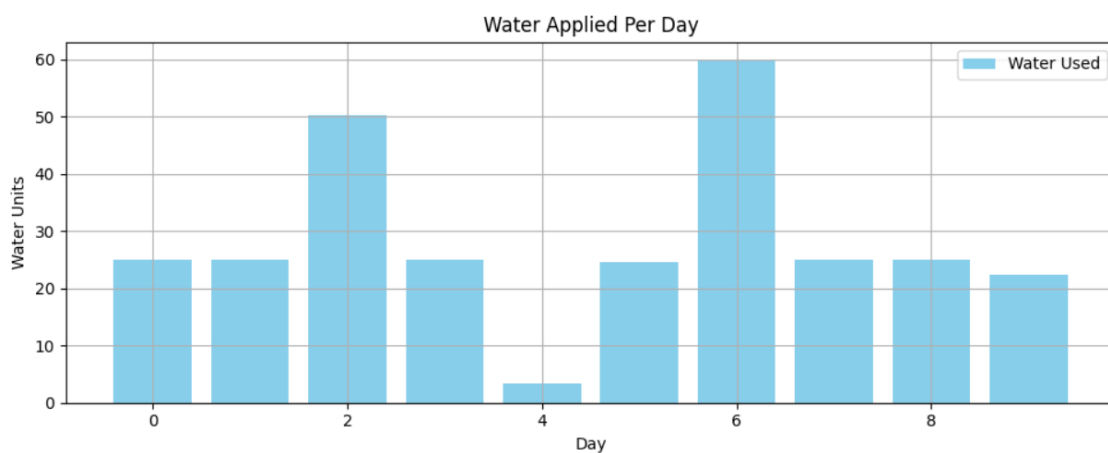
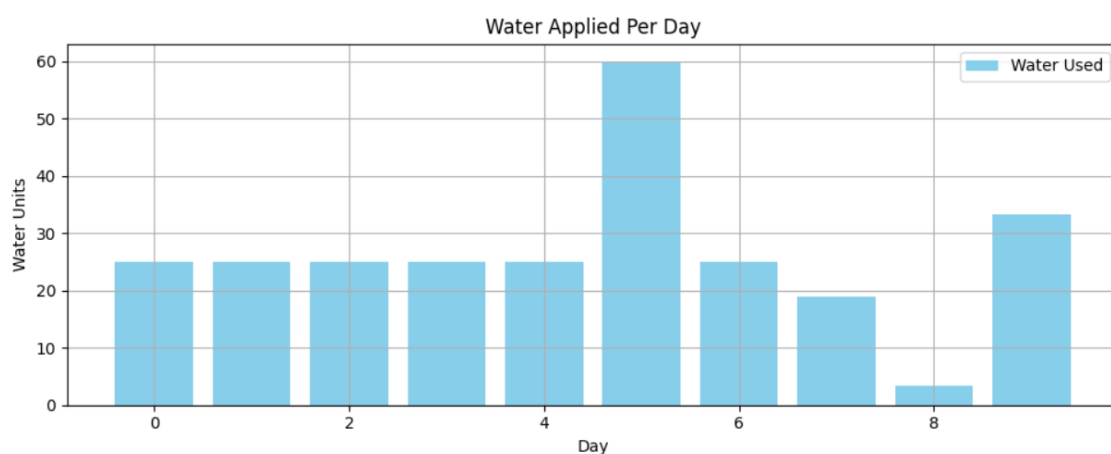
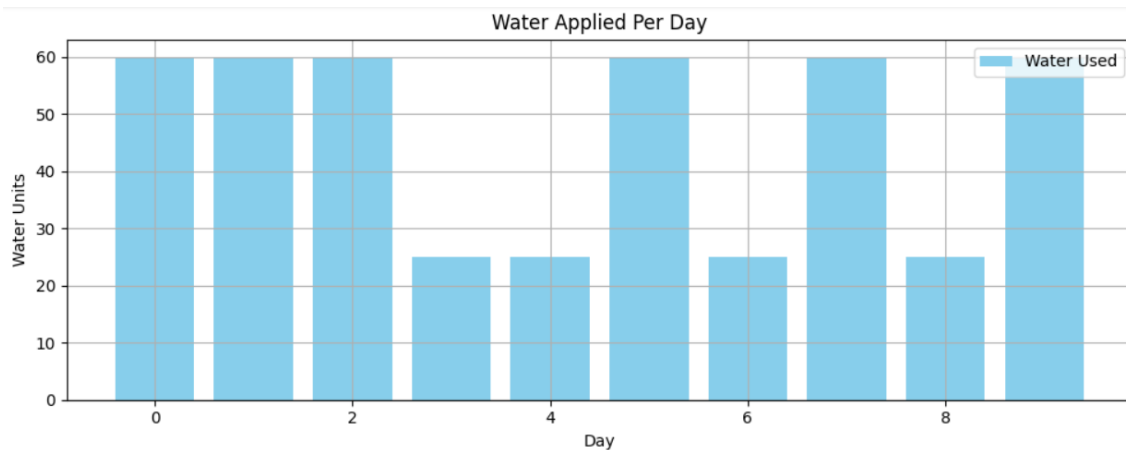
۳-۳-۱- تغییرات رطوبت خاک در طول زمان

سه اجرا با شرایط آب و هوایی متفاوت:



۳-۲- میزان آب دهی تعیین شده توسط کنترل کننده فازی در طول زمان

میزان آب دهی برای همان سه اجرای قبل:



نمونه خروجی:

```
Day 1:
Weather: Sunny ☀️
Soil Moisture Before: 40.00%
Water Given: 60.00 units
Activated Rule: Rule 4
-----
Day 2:
Weather: Sunny ☀️
Soil Moisture Before: 37.88%
Water Given: 60.00 units
Activated Rule: Rule 4
-----
Day 3:
Weather: Sunny ☀️
Soil Moisture Before: 35.86%
Water Given: 60.00 units
Activated Rule: Rule 4
-----
```

۳-۴- نتایج را تحلیل و بررسی کنید. کنترل کننده فازی طراحی شده تا چه حد در حفظ سطح بهینه رطوبت خاک موفق است؟

با توجه به اجراهای متفاوت و بررسی تغییرات رطوبت خاک در طی ده روز، این کنترل کننده سطح رطوبت خاک را به صورت حدودی در بازه ۲۰ تا ۶۰ که مقدار بهینه است، نگه می‌دارد.

- [1] "chatgpt," [Online]. Available: <https://chatgpt.com/>.
- [2] Kruse, Mostaghim, Borgelt, Braune and Steinbrecher, Computational intelligence: A methodological introduction, 2022.
- [3] youtube. [Online]. Available:
<https://www.youtube.com/watch?v=XuWPdFteIkQ>.
- [4] "youtube," [Online]. Available:
https://www.youtube.com/watch?v=__0nZuG4sTw.