

به نام خالق بی همتا



کنترل کننده منطق فازی برای سیستم آبیاری خودکار گیاهان

درس

هوش محاسباتی

استاد

دکتر حسین کارشناس

دستیاران آموزشی

رضا برزگر

علی شاه زمانی

آرمان خلیلی

نویسندگان

محمد امین نصیری

سید حسین حسینی دولت آبادی

دانشکده مهندسی کامپیوتر

دانشگاه اصفهان

بهار ۱۴۰۴

فهرست

۲	معرفی پروژه.....
۲	اهداف پروژه.....
۲	تحلیل مسأله.....
۲	ساختار کلی سیستم.....
۲	قوانین فازی.....
۳	تحلیل پیاده‌سازی.....
۳	نمایش تصویری.....
۴	پیشنهادات توسعه‌ای.....
۴	شبیه‌سازی ۱۰ روزه.....
۴	تحلیل و بررسی.....
۴	قبل از افزایش قوانین.....
۴	پس از افزایش قوانین.....
۴	جمع‌بندی.....
۵	نتیجه‌گیری.....
۵	لینک‌های مرتبط.....
۵	منابع.....

مستندات پروژه

معرفی پروژه

HydroWizard یک سیستم آبیاری هوشمند مبتنی بر منطق فازی است. این سیستم به طور خود کار و با توجه به شرایط محیطی مانند رطوبت خاک و وضعیت آب و هوا، میزان مناسب آبیاری را تعیین می کند. هدف آن کاهش مصرف آب و افزایش بهره وری کشاورزی است.

اهداف پروژه

- بهینه سازی مصرف آب در کشاورزی
- کاهش وابستگی به نیروی انسانی
- ایجاد زیرساخت برای آبیاری مبتنی بر داده
- سازگاری با سیستم های اینترنت اشیا

تحلیل مسأله

در روش های سنتی، آبیاری به صورت زمان بندی شده یا بر اساس تجربه انجام می شود که ممکن است منجر به هدررفت منابع یا آسیب به گیاه شود. HydroWizard با استفاده از منطق فازی و ورودی های محیطی تصمیم گیری دقیق تری ارائه می دهد.

ساختار کلی سیستم

سیستم از دو ورودی و یک خروجی تشکیل شده است:

ورودی ها:

- ✓ رطوبت خاک (Dry, Medium, Wet)
- ✓ وضعیت آب و هوا (Sunny, Cloudy, Rainy)

خروجی:

- ✓ میزان آبیاری (None, Low, Medium, High)

بر اساس قوانین فازی، خروجی مناسب بر حسب ورودی ها تعیین می شود.

قوانین فازی

قوانین فازی به صورت IF-THEN تعریف می شوند. نمونه ها:

- اگر رطوبت خشک و هوا آفتابی باشد ← آبیاری زیاد
- اگر رطوبت مرطوب و هوا بارانی باشد ← بدون آبیاری
- اگر رطوبت متوسط و هوا ابری باشد ← آبیاری متوسط

تحلیل پیاده‌سازی

این پروژه با زبان Python و کتابخانه scikit-fuzzy پیاده‌سازی شده است. مراحل کلی:

1. تعریف متغیرهای ورودی/خروجی و دامنه‌ها بین ۰ تا ۱۰۰
 2. تعریف توابع عضویت فازی با *trimf* و *trapmf*
 - a. با توجه به مقالات ارائه شده در بخش منابع [1], [2] برای مقادیر *wet* و *dry* در متغیر ورودی و مقدار *meduim* در متغیر خروجی از توابع دوزنقه‌ای و برای بقیه‌ی مقادیر از توابع عضویت مثلثی استفاده شده است.
 - b. برای تعریف ورودی آب و هوا ابتدا از ۳ مقدار قطعی ۰، ۵۰ و ۱۰۰ استفاده شد که هر کدام به ترتیب برای *cloudy*, *sunny* و *rainy* دارای درجه عضویت یک می‌باشند. سپس این مقادیر بعنوان میانگین برای ایجاد توابع عضویت فازی استفاده شد.
 3. تعریف قوانین فازی
 4. ایجاد سیستم کنترل فازی
 5. شبیه‌سازی با مقادیر ورودی و مشاهده خروجی (انواع خروجی‌ها قابل مشاهده است که هر کدام در یک سامانه کاربرد دارد مانند مرکز ثقل و یا مقدار بیشینه و ...)
- کتابخانه matplotlib نیز برای نمایش نمودارها استفاده شده است.

نمایش تصویری

در طول اجرای نوت‌بوک، نمودارهای عضویت برای متغیرها و همچنین خروجی نهایی به صورت گرافیکی نمایش داده می‌شوند که درک بهتری از عملکرد سیستم ارائه می‌دهند.

مزایا

- + تصمیم‌گیری هوشمند و منعطف
- + امکان توسعه و افزودن پارامترهای جدید
- + ترکیب منطق انسانی با سیستم ماشینی

پیشنهادهای توسعه‌ای

- افزودن پارامترهایی مانند دمای هوا، نوع خاک یا گیاه
- پیاده‌سازی رابط کاربری گرافیکی
- اتصال به سنسورها در بستر IoT
- تست در شرایط واقعی و میدانی

شبیه‌سازی ۱۰ روزه

تنظیمات اولیه و لیست آب و هوای اولیه طبق فایل راهنما ایجاد کرده و طبق فرمول
$$\text{new_moisture} = \text{old_moisture} + (\text{irrigation_amount} / 10) + \text{evap}$$
 رطوبت جدید را بدست می‌آوریم. که رطوبت خاک در روزهای پایانی این ده روز در حدود ۴۰ درصد باقی می‌ماند.

تحلیل و بررسی

قبل از افزایش قوانین

در نسخه اولیه، سیستم با رطوبت خاک ۱۵ درصد کار خود را شروع کرد و تا روز دهم به حدود ۴۱.۴۵ درصد رسید. در نگاه اول شاید به نظر برسد که این روند خوب پیش رفته، اما واقعیت این است که در تمام این ده روز، رطوبت خاک حتی یک بار هم وارد محدوده‌ی بهینه‌ی ۴۰ تا ۸۰ درصد نشد. آبیاری انجام می‌شد، اما نه آنقدر مؤثر که بتواند نیاز واقعی گیاه را برطرف کند. سیستم نسبت به تغییرات آب‌وهوا واکنش نشان می‌داد، اما این واکنش‌ها کند و گاهی ناکارآمد بود. درواقع سیستم بیشتر از آن که هوشمند عمل کند، محتاط و محدود ظاهر شد.

پس از افزایش قوانین

وقتی پایگاه قوانین فازی گسترش پیدا کرد، همه‌چیز شروع به بهتر شدن کرد. همان رطوبت اولیه‌ی ۱۵ درصد، این بار در پایان روز دهم به ۴۶.۷۳ درصد رسید. مهم‌تر از آن، در سه روز از ده روز، رطوبت خاک وارد محدوده‌ی بهینه شد. این یعنی سیستم نه تنها سریع‌تر عمل کرد، بلکه توانست سطح رطوبت را بهتر کنترل کند. میزان آبیاری در این نسخه کمی بیشتر بود، اما به جای مصرف بی‌هدف، هدفمند و با واکنش مناسب به شرایط روزهای آفتابی، ابری یا بارانی انجام شد. در نتیجه خاک زودتر به سطح مطلوب رسید و گیاهان شانس بیشتری برای رشد بهتر پیدا کردند.

جمع‌بندی

واضح است که افزایش قوانین فازی عملکرد سیستم را متحول کرد. سیستم از حالت محافظه‌کار و نسبتاً غیرفعال، به سیستمی واکنش‌گرا، دقیق و هوشمند تبدیل شد. در محیط‌هایی مثل گلخانه که دقت در آبیاری حیاتی است، چنین تغییری می‌تواند تفاوت زیادی در رشد و سلامت گیاهان ایجاد کند. پس اگر هدف طراحی یک سیستم آبیاری واقعاً هوشمند است، داشتن پایگاه قوانین قوی و کامل، نه تنها مفید بلکه ضروری است.

نتیجه گیری

HydroWizard گامی مؤثر در راستای استفاده از فناوری‌های نوین برای مدیریت منابع آبی است. این پروژه نمونه‌ای از کاربرد منطق فازی در حل مسائل دنیای واقعی به شمار می‌آید.

نکته) لازم به ذکر است کلیه بخش‌های پروژه اعم از اجباری و اختیاری پیاده سازی شده است و مصورسازی تمامی توابع عضویت و خروجی سیستم در نوت‌بوک پروژه آورده شده است.

لینک‌های مرتبط

نوت‌بوک در گوگل کولب:

https://colab.research.google.com/github/Sayed-Hossein-Hosseini/HydroWizard_Intelligent_Fuzzy_Logic_Irrigation/blob/master/HydroWizard_Intelligent_Fuzzy_Logic_Irrigation.ipynb

ریپازیتوری گیت‌هاب:

https://github.com/Sayed-Hossein-Hosseini/HydroWizard_Intelligent_Fuzzy_Logic_Irrigation.git

منابع

- [1] K., Amthal & Al-Janabi, Muayad & Abdullah, Mohammed. (2014). Fuzzy based Decision Support Model for Irrigation System Management. International Journal of Computer Applications. 104. 14-20. 10.5120/18230-9177 <https://www.researchgate.net/publication/284367853>
- [2] Neugebauer, M., Akdeniz, C., Demir, V. et al. Fuzzy logic control for watering system. Sci Rep 13, 18485 (2023) <https://doi.org/10.1038/s41598-023-45203-2>
- [3] <https://github.com/xbeat/Machine-Learning/blob/main/Introduction%20to%20Fuzzy%20Logic%20in%20Python.md>
- [4] ChatGPT