

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA - CCET
DEPARTAMENTO DE ENGENHARIA ELÉTRICA - DEE

GENIVAL DE MELO JUNIOR

ANÁLISE E COMPARAÇÃO DE RESULTADOS DE UM
CONTROLADOR USANDO LÓGICA FUZZY E
OUTRAS FORMAS DE CONTROLE PARA
IRRIGAÇÃO.

SÃO CARLOS -SP
2024

GENIVAL DE MELO JUNIOR

ANÁLISE E COMPARAÇÃO DE RESULTADOS DE UM CONTROLADOR USANDO
LÓGICA FUZZY E OUTRAS FORMAS DE CONTROLE PARA IRRIGAÇÃO.

Trabalho de conclusão de curso apresentado ao Departamento de Engenharia Elétrica da Universidade Federal de São Carlos, para obtenção do título de bacharel em Engenharia Elétrica.

Orientador: Prof.Dr. Celso Ap.de França

São Carlos-SP
2024

AGRADECIMENTO

A Deus por sempre me conceder forças, a Nossa Senhora por sua intercessão em momentos importantes, à minha família, em especial meu falecido pai, minha mãe, irmãos e irmã por sempre me apoiarem. Ao Prof. Dr. Celso Aparecido de França pela disponibilidade, paciência e atenção em todo o processo de elaboração deste trabalho.

A todos os professores que dedicaram seu tempo e disponibilidade para ensinar e formar, aos autores usados como referência que foram fundamentais para que este trabalho tenha sido realizado, e a todos que fizeram parte direta ou indiretamente de toda a jornada até aqui.

RESUMO

A agricultura é indispensável na forma como a sociedade humana vive atualmente no mundo, pois é dela que vem a maior parte da alimentação consumida pelos humanos em todo o planeta. Neste sentido, a irrigação é um processo bastante importante, pois possibilita o controle e melhoria da produção desde o início do processo de sedentarismo humano, haja vista que as sociedades que melhor se desenvolveram desde a antiguidade são povos que utilizavam sistemas de irrigação, como egípcios, romanos e povos mesopotâmicos. Dada esta importância, é necessário levar em conta qual a melhor forma de realizar o processo de irrigação. Este trabalho visou comparar diferentes formas de controle de irrigação, dentre eles controladores tradicionais e sistemas de irrigação manual, e verificar qual a forma mais eficiente e viável entre elas por meio da comparação do nível de umidade do solo, economia de água e desempenho dos controladores com base em critérios clássicos. Buscou-se comparar essas formas de controle tradicionais com um controlador que utiliza lógica fuzzy e a plataforma Arduino através da implementação prática de alguns controladores e revisão bibliográfica para reunir comparativos já existentes. Ao término, foi possível concluir que o controle com lógica fuzzy apresentou melhores resultados, tendo maior economia de água e menor sobressinal, embora tenha tido uma resposta mais lenta em comparação com outros tipos de controladores, mesmo com seu início mais rápido.

Palavras-Chave: irrigação; fuzzy; Arduino; sistema microcontrolado; controle inteligente.

ABSTRACT

Agriculture is indispensable to the way human society currently lives in the world, as it is the source of most of the food consumed by humans across the planet. In this sense, irrigation is a very important process, as it allows for the control and improvement of production since the beginning of human sedentism, given that the societies that developed the best since antiquity were those that used irrigation systems, such as the Egyptians, Romans, and Mesopotamian peoples. Given this importance, it is necessary to consider the best way to carry out the irrigation process. This work aimed to compare different forms of irrigation control, among them traditional controllers and manual irrigation systems, and to verify which is the most efficient and viable form among them by comparing soil moisture levels, water savings, and controller performance based on classical criteria. The aim was to compare these traditional control methods with a controller that uses fuzzy logic and the Arduino platform through the practical implementation of some controllers and a literature review to gather existing comparisons. In the end, it was possible to conclude that the control with fuzzy logic presented better results, with greater water savings and less overshoot, although it had a slower response compared to other types of controllers, even with its faster start.

Keywords: irrigation; fuzzy; Arduino; microcontrol system; smart control.

LISTA DE ILUSTRAÇÕES

Figura 1 - Irrigação por aspersão	15
Figura 2 - Irrigação por microaspersão	16
Figura 3 - Irrigação por gotejamento	16
Figura 4 - Exemplo de funções de pertinência	21
Figura 5 - Esboço de função de pertinência triangular	22
Figura 6 - Esboço de função de pertinência trapezoidal	23
Figura 7 - Esboço de função de pertinência gaussiana	23
Figura 8 - Diagrama da estrutura básica de um sistema fuzzy	24
Figura 9 - Formação do conjunto fuzzy de entrada	25
Figura 10 - Composição da saída final	26
Figura 11 - Microcontrolador ATmega328	28
Figura 12 - Arduino Uno	29
Figura 13 - Sensor de umidade do solo	30
Figura 14 - Relé para acionamento	30
Figura 15 - Solenóide	31
Figura 16 - Esquema simplificado de um sistema controlado em malha fechada contínuo	31
Figura 17 - Esquema simplificado de um sistema controlado digital em malha fechada	32
Figura 18 - Diagrama de blocos do sistema controlado implementado	39
Figura 19 - Diagrama de funcionamento de um controlador on/off	42
Figura 20 - Diagrama de funcionamento de um acionador com timer em malha aberta	43
Figura 21 - Funções de pertinência para a variável de entrada	44
Figura 22 - Funções de saída	45
Figura 23 - Superfície de resposta fuzzy	47
Figura 24 - Esquemático da montagem dos circuitos	49
Figura 25 - Montagem real	50
Figura 26 - Nível de umidade do solo de cada sistema no tempo	53
Figura 25 - Acumulado do consumo de água por cada controlador	57

LISTA DE TABELAS

Tabela 1: Resumo de informações dos tipos de irrigação considerados para implementação.	17
Tabela 2 - Parâmetros gerais dos controladores	48
Tabela 3 - Relação de umidade em cada sistema em relação aos ciclos de 30 minutos	52
Tabela 4 - Consumo de água por cada controlador	56

LISTA DE SIGLAS

A/D - Analog to Digital (Conversor Analógico para Digital).

CDA - Centro de Área

CPU - Central Processing Unit (Unidade Central de Processamento).

D/A - Digital to Analog (Conversor Digital para Analógico).

I.A - Inteligência Artificial

IDE - Integrated Development Environment (Ambiente de Desenvolvimento Integrado)

LOM - Largest of Maximum (Maior dos Máximos)

MDM - Média do Máximos

MPM - Método do Primeiro Máximo

PID - Proporcional Integral Derivativo

RAM - Random Access Memory (Memória de Acesso Aleatório).

ROM - Read Only Memory (Memória Somente de Leitura).

SUMÁRIO

1	INTRODUÇÃO	10
1.1	CONTEXTUALIZAÇÃO	10
1.2	JUSTIFICATIVA	11
1.3	OBJETIVOS	12
1.4	ESTRUTURA DO TRABALHO	13
2	FUNDAMENTAÇÃO TEÓRICA E REVISÃO DA LITERATURA	14
2.1	IRRIGAÇÃO NA AGRICULTURA	14
2.2	TIPOS DE IRRIGAÇÃO	14
2.3	AUTOMATIZAÇÃO DA IRRIGAÇÃO	17
2.4	CONTROLADORES FUZZY	19
2.4.1	Funções de pertinência e variáveis linguísticas	20
2.4.2	Fuzzificação e Inferência fuzzy	24
2.4.3	Defuzzificação	26
2.4.4	Características da implementação dos sistemas fuzzy	26
2.5	EQUIPAMENTOS UTILIZADOS	27
2.5.1	Equipamentos, dispositivos e softwares	27
2.5.2	Microcontrolador	28
2.5.3	Componentes	29
2.6	FUNCIONAMENTO DE UM CONTROLADOR DIGITAL	31
2.7	COMPARAÇÃO ENTRE SISTEMA FUZZY E PID	33
3	DESENVOLVIMENTO DO TRABALHO	36
3.1	ESCOLHA DOS CONTROLADORES COMPARADOS E IMPLEMENTADOS	36
3.2	ANÁLISES QUALITATIVA X QUANTITATIVA	38
3.3	DESENVOLVIMENTO DOS CONTROLADORES	38
3.3.1	Testes e montagem dos controladores	40
3.3.2	Controlador liga-desliga com realimentação	41
3.3.3	Acionador com timer sem realimentação	42
3.3.4	Controlador fuzzy	43
3.3.5	Parâmetros gerais de testes e implementação	47
3.4	PESQUISA E COMPARAÇÃO BIBLIOGRÁFICA	48
3.5	IMPLEMENTAÇÃO PRÁTICA	48
4	RESULTADOS	51
4.1	COMPARAÇÃO ENTRE FUZZY E IRRIGAÇÃO MANUAL	51
4.2	COMPARAÇÃO DOS RESULTADOS EXPERIMENTAIS	52
5	CONSIDERAÇÕES FINAIS	58
5.1	CONCLUSÕES	58
5.2	TRABALHOS FUTUROS	58
	REFERÊNCIAS	60
	APÊNDICE A - Código do controlador liga/desliga com realimentação	68
	APÊNDICE B - Código do acionador timer sem realimentação	70
	APÊNDICE C - Código do sistema fuzzy implementado	71

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

A agricultura é uma das principais atividades do ser humano, sendo apontada historicamente como um dos principais motivos do desenvolvimento das sociedades que, desde a antiguidade e com o processo de sedentarização, passaram a se desenvolver melhor tendo uma fonte de alimento garantida, dispensando a necessidade de caçar ou coletar, além de ter se tornado a principal fonte de alimentação das civilizações humanas de modo geral desde então (TAUGER, 2020). Neste processo, a irrigação se tornou um procedimento indispensável pois as plantações e cultivos dependem de água, sendo este, inclusive, um dos principais motivos pelo qual muitos historiadores apontam o desenvolvimento de grandes civilizações: a proximidade de rios e a facilitação da irrigação, um processo que já ocorre há cerca de 9 mil anos (FELDENS, 2018).

O processo de irrigação consiste na aplicação de água no solo de modo artificial, isto é, por meio de mecanismos desenvolvidos pelos seres humanos, o que é capaz de resolver problemas como a falta de chuva ou chuva distribuída de forma não uniforme em uma região (CARVALHO, 2018). Essa aplicação de água no solo precisa ser controlada pois, ao mesmo tempo que a ausência de irrigação pode causar prejuízos irreversíveis devido à quantidade de água na planta ser insuficiente (CASTAN, 2021), o excesso de água também pode ser muito prejudicial, podendo causar sufocamento pela falta de oxigênio, escoamento dos nutrientes e fertilizantes, além do gasto de água desnecessário e prejudicial neste processo chamado de over-watering (LIMA, 2007).

Como não existe uma forma única definida para os processos de irrigação tal que esse processo possa ter valores padronizados, pois cada tipo de cultura tem seus próprios parâmetros e necessidades (FRIZZONE, 2017), o controle do processo de irrigação é muito importante, podendo ser realizado por um ser humano especialista naquela determinada cultura. No entanto, o avanço tecnológico possibilita a implementação de técnicas de automação neste processo, tais como a implementação de controladores eletrônicos, microcontroladores e tecnologias de informação, podendo ser implementado um controle automatizado do sistema de

irrigação (SOUZA, 2002). Neste processo automatizado, o ser humano não precisa mais agir diretamente na irrigação, podendo dedicar-se apenas a definir parâmetros de controle, atuação e supervisão, além de gerenciar o sistema (REHAGRO, 2017).

No processo automatizado é possível ter uma gestão melhor do início e do fim da irrigação, isto é, o tempo total de acionamento, além de ser possível controlar de forma mais precisa o volume de água gasto, utilizando apenas o necessário para a planta e evitando excessos que, além de custosos ambiental e, muitas vezes, financeiramente, podem prejudicar o solo e a planta (CARVALHO, 2018). Apesar disso, a implementação de um sistema automático de irrigação pode ter custos financeiros relativamente altos e também ser difícil principalmente devido à modelagem matemática, isto é, se todas as variáveis envolvidas no processo forem levadas em conta de modo a obter um modelo matemático o mais preciso possível, conforme os modelos tradicionais de controle, o processo pode tornar-se bastante complexo (GOMIDE, 1994), o que deve acarretar aumento dos custos de desenvolvimento ao preço final do controlador. Sendo assim, como alternativa aos métodos de controle tradicionais, existem os controladores que funcionam com lógica fuzzy, baseada em variáveis linguísticas, o que pode facilitar e baratear o processo de modelagem e desenvolvimento, dispensando o uso de técnicas mais complexas de modelagem matemática (SANTOS, 2019), embora os componentes usados para implementar um controle analógico, por exemplo, sejam, em geral, mais baratos do que os equipamentos usados para implementar um sistema fuzzy.

No caso do controle com lógica geralmente são consultados especialistas para a determinação da operação do controlador, determinando em conjunto quais as melhores funções de pertinência com base nas regras de classificação linguística utilizadas por esses especialistas.

O uso de sistemas baseados em lógica fuzzy já vem sendo empregado na agricultura de modo geral há algum tempo, sendo usado em processos de controle dos tratos com as plantas de modo geral e também na irrigação (GODINHO; GASPAROTO; CANEPPELE, 2022), sendo, portanto, válido comparar este método com outros tradicionalmente usados na mesma área de aplicação de modo a tentar descobrir qual apresenta maior vantagem em sua aplicação.

1.2 JUSTIFICATIVA

O consumo excessivo e o desperdício de água no mundo todo vem se tornando um problema devido à escassez de água potável. Apenas no Brasil, mais de 2 milhões de litros de água são gastos por segundo, com esse volume podendo chegar a 2,5 milhões de litros por segundo até 2030, sendo que cerca de 52% desse total, isto é, mais da metade, é destinado à agricultura e irrigação (ECODEBATE, 2019).

Devido a esse consumo de grande volume de água, a busca por formas mais eficientes, capazes de reduzir esse consumo, podem ser benéficas, considerando que água potável é um recurso cada vez mais escasso (LEITÃO, 2018) e precisa ser gerido com cuidado pois é um recurso essencial para a vida.

Com isso, busca-se encontrar uma forma de promover a economia de água consumida no processo de irrigação, automatizar este processo e diminuir o tempo necessário para sua realização. Para tal, busca-se comparar o sistema automatizado com controlador fuzzy e outros sistemas geridos de outras formas, de modo a verificar qual apresenta menor consumo de água e melhor desempenho segundo os parâmetros tradicionais de controle, definindo qual método é capaz de ter menores custos, consumo de água e melhor produtividade em comparação ao outro.

1.3 OBJETIVOS

Comparar o desempenho de um controlador utilizando lógica fuzzy com outros tipos de controladores. Desenvolver o protótipo de um sistema de controle capaz de automatizar o processo de irrigação de plantas utilizando a plataforma Arduino, que conta com o microcontrolador ATmega328, e implementar um controlador baseado em lógica fuzzy, que pode simplificar o processo de abstração e implementação do controlador. Após a implementação do sistema com o controlador, o objetivo foi comparar os resultados obtidos para verificar se houve redução no consumo de água e comparar o desempenho dos diferentes tipos de controle e acionamento de modo a verificar qual apresentou a melhor resposta de controle de umidade do solo. Além da comparação obtida pela implementação prática, também foram consideradas, por meio de revisões bibliográficas, comparações entre o controlador Fuzzy e outros tipos de supervisão de irrigação,

como a irrigação manual e o controle PID. Ao término, foi analisado se havia viabilidade técnica, financeira e ambiental na implementação de um controlador, levando-se em conta a comparação com outros sistemas analisados.

1.4 ESTRUTURA DO TRABALHO

O trabalho está dividido em cinco capítulos. Este capítulo inicial contextualiza e justifica o trabalho. No segundo capítulo é apresentada a fundamentação teórica sobre o assunto obtida através da revisão bibliográfica. O terceiro capítulo mostra o desenvolvimento e a implementação do trabalho. No quarto capítulo são relatados e comentados os resultados obtidos. Por fim, as conclusões finais são apresentadas no quinto capítulo.

2 FUNDAMENTAÇÃO TEÓRICA E REVISÃO DE LITERATURA

Inicialmente foram realizados levantamentos e pesquisas para compreender o processo de irrigação, quais as formas de irrigação utilizadas amplamente, as principais tecnologias e quais os tipos de controladores usados na área. Além disso, é fundamental conhecer e introduzir os principais elementos a serem utilizados no desenvolvimento do trabalho, como os componentes implementados, os softwares utilizados e as tecnologias como a plataforma Arduino, que é base da implementação do controlador deste projeto.

2.1 IRRIGAÇÃO NA AGRICULTURA

Conforme visto, a agricultura é parte fundamental no desenvolvimento de diversas civilizações por possibilitar o acesso amplo a alimentos (TAUGER, 2020). Entretanto, sem o processo de irrigação, a agricultura tem pouco desenvolvimento por ficar dependente de condições climáticas que, por vezes, é desfavorável e pode levar à morte das culturas em períodos de seca, por exemplo. Por isso a irrigação é utilizada na agricultura já há milênios (FELDENS, 2018).

A irrigação também se torna um fator relevante de controle de custos, de recursos, de materiais, de produtividade do solo e, principalmente, de água. Assim, controlar a irrigação é controlar os recursos, econômicos ou naturais, e controlar a produtividade, de modo que a irrigação é “uma estratégia para aumento da produção, produtividade e rentabilidade da propriedade agrícola de forma sustentável, preservando o meio” (MANTOVANI et al, 2013, p. 13).

2.2 TIPOS DE IRRIGAÇÃO

Existem diversos tipos de irrigação diferentes, utilizando técnicas diferentes. Segundo Coelho et al (2007), a escolha do tipo de irrigação dependerá de uma série de fatores como a cultura, o local e o clima, entre outras variáveis. Aqui serão apresentados apenas alguns dos principais tipos de irrigação mais comumente utilizados.

A irrigação por aspersão é um tipo de irrigação que se baseia em simular a chuva, de modo que a água é bombeada para cima e cai em gotículas de modo uniforme sobre a área irrigada (STONE, 2011). Esta técnica é particularmente útil em casos onde a extensão territorial plantada é muito grande e exigiria muitos

recursos materiais para ser irrigada de outra forma ou é de difícil acesso, como encostas. No entanto, este tipo de técnica gera um maior desperdício de água, já que boa parte é evaporada no próprio processo de aspersão, antes mesmo de ser precipitada no solo, e outra grande parte fica em contato com as folhas das plantas, e não com o solo, e também evapora sem ser totalmente aproveitada. Além disso, essa técnica aumenta a umidade relativa do ar, o que pode acarretar no aumento de pragas. A Figura 1 apresenta um exemplo de irrigação por aspersão.

Figura 1: Irrigação por aspersão



Fonte: MaxMaq (2019)

A irrigação por microaspersão é um técnica que também se baseia em aspersão, porém mais próxima do solo, aumentando o rendimento e reduzindo a quantidade de água utilizada para irrigar uma mesma área com microaspersores (SOUSA, 2016). Geralmente esta técnica é utilizada no cultivo de hortaliças, plantas de médio porte e extensões territoriais médias. A Figura 2 apresenta um exemplo de irrigação por microaspersão.

Figura 2: Irrigação por microaspersão



Fonte: BONISSONI (2022)

A irrigação por gotejamento, ao contrário das outras que são específicas para determinados casos de maior extensão de cultivo, pode ser mais interessante para sistemas mais simples, com menor área de cultivo e plantas mais sensíveis à quantidade de água depositada, embora possa ser usada em grandes áreas cultivadas também, tendo baixo consumo de água e baixo custo de montagem (YOHANNES & TADESSE, 1998), embora o custo inicial dos equipamentos seja um pouco mais alto do que outras formas (YARABRASIL, 2022). A Figura 3 apresenta um exemplo de irrigação por gotejamento.

Figura 3: Irrigação por gotejamento.



Fonte: UNIBAVE (2018)

Segundo Medeiros (2018), existem ainda várias outras formas de irrigação, como a irrigação por sulco e a subirrigação, mas sistemas como estes não são vantajosos para este trabalho por serem mais complicados e custosos de implementar, levando em consideração que o objeto de análise é o controlador e não o tipo de irrigação utilizado.

Dentre as três formas consideradas, a Tabela 1 mostra as principais características que cada tipo apresenta, levando em consideração o consumo de água, o custo de implementação, aplicação e a eficiência de rega - EF (PEREZ-ORTOLA; KNOX, 2014).

Tabela 1: Resumo de informações dos tipos de irrigação considerados para implementação.

	Aspersão	Microaspersão	Gotejamento
Vazão de água	Alta	Relativamente baixa	Baixa
Aplicação	Grandes áreas	Hortaliças e porte médio	Plantas sensíveis
Custo	Relativamente alto	Barato	Relativamente barato
EF (%)	65%	90%	90%

Fonte: Autoria própria.

Assim, dentre as opções, as que se mostram melhores para a implementação de um teste são a irrigação por microaspersão e por gotejamento. Dentre elas, optou-se pela de mais simples implementação e menor custo neste caso, que é a por microaspersão.

2.3 AUTOMATIZAÇÃO DA IRRIGAÇÃO

Atualmente, a utilização de processos de automatização na irrigação vem sendo cada vez mais utilizado por uma série de fatores, desde a diminuição da necessidade de trabalho humano, que exige certo esforço físico em algumas condições adversas, até o controle e redução de recursos gastos e consumidos, fazendo com que o uso da tecnologia de automação seja necessária para gerar cada vez mais eficiência no processo e na redução de consumos (DURSON &

OZDEN, 2011).

Dentre as formas de automatizar o processo de irrigação, destacam-se duas. A primeira consiste em apenas automatizar o processo, mas sem nenhum tipo de controlador operando com coletas de dados, ou seja, sem *feedback* de informação, isto é, em malha aberta. Nestes casos, geralmente são utilizados temporizadores que acionam atuadores, abrindo ou fechando a válvula de irrigação por períodos de tempo previamente estabelecidos. Este tipo de processo geralmente é mais simples e exige menos equipamentos, no entanto o gasto de recursos tende a ser maior, como a quantidade de água que é utilizada (mesmo que o solo ainda esteja úmido, ou que tenha chovido, por exemplo, ele aciona após o tempo programado, a não ser que seja desligado manualmente), a quantidade de fertilizantes que pode ser perdida devido ao possível excesso de água no solo em certos casos, ou o inverso: em períodos de seca, onde o solo necessitaria de mais água, o sistema fornece apenas o programado, ainda que o solo continue com nível de umidade baixo. Tudo isso pode danificar a fertilidade do solo devido ao acúmulo (ou falta) de nutrientes e minerais, além de reduzir a produtividade por excesso ou falta de água (MEDEIROS, 2018). Claro que ajustes manuais podem ser feitos nos temporizadores de acordo com a necessidade, mas além de exigir mais conhecimento por parte do especialista, a necessidade de alterações manuais tende a não seguir a melhor curva de aproveitamento e não ter a mesma velocidade que um sistema projetado para realizar a mesma ação automaticamente (NISE, 2011).

A segunda forma atua com controladores baseado em *feedbacks* de dados recolhidos, isto é, malha fechada, como níveis de umidade do solo, temperatura do ambiente ou outras variáveis. A implementação deste tipo de irrigação, com realimentação de dados, é interessante para automatizar com maior proveito o sistema, tendendo a reduzir o consumo desnecessário de recursos, melhorando a velocidade de resposta e ajustando de maneira mais próxima à curva ideal de resposta o sistema, além de levar a um melhor aproveitamento do solo e da cultura plantada, já que responde de acordo com a necessidade destes (MEDEIROS, 2018).

Assim, o sistema com unidade temporizadora em malha fechada, utilizando controle realimentado com *feedback* de dados, é mais interessante para poupar recursos e apresentar uma melhor produtividade, por isso um dos sistemas a ser implementado neste projeto baseia-se nisto para implementar sensores de umidade

do solo, para promover a realimentação do controlador do sistema.

O controlador PID, uma forma de controle tradicional baseada em ações de controle proporcional, integral e derivativa (OGATA, 2011), bastante usada em setores industriais e outras aplicações, não é amplamente utilizado na irrigação de culturas extensivas, tendo seu uso mais restrito a pequenos cultivos, culturas *indoor*, hortas verticais e até mesmo jardinagem (RESENDE, 2014). Outras formas de controle, como redes neurais e algoritmos genéticos, ainda estão em fase de desenvolvimento na área de aplicação e muitas vezes os estudos se baseiam mais em simulações do que implementações físicas (WOLFRANN et al., 2018; MENEZES, 2014), sendo, portanto, pouco usado na prática na área de irrigação, embora sejam amplamente utilizados em outras áreas.

2.4 CONTROLADORES FUZZY

A lógica fuzzy parte do pressuposto que determinados sistemas são complexos demais para serem plenamente controlados, de forma viável, com as lógicas convencionais, como a booleana, por exemplo, ou pelo menos poderiam ser mais facilmente controlados com outro tipo de lógica. Por este motivo, em 1965, Lotfali Askar-Zadeh iniciou as pesquisas para desenvolver novas formas de lógicas para controladores, chegando ao fuzzy, que baseia-se no método linguístico de expressar quantidades intermediárias, e não apenas níveis lógicos alto e baixo, como faz a lógica booleana. Com isto são utilizados também valores intermediários (ROISENBERG, 2018). Assim, é possível determinar não apenas se uma condição está presente, mas também o seu nível de presença, ou pertinência, isto é, se está muito presente, pouco presente, entre outros níveis intermediários, e com base neste nível de pertinência, determinar a ação de controle. Por exemplo, é possível dizer que o solo está úmido ou não na lógica booleana, mas na lógica fuzzy é possível dizer que está muito pouco úmido, pouco úmido, úmido, muito úmido, excessivamente úmido, etc, e com base nisto determinar uma ação de controle que vai depositar maior ou menor volume de água no processo de irrigação. É possível, portanto, determinar vários níveis intermediários entre o “sim” e o “não” e atuar também com valores intermediários.

2.4.1 Funções de pertinência e variáveis linguísticas

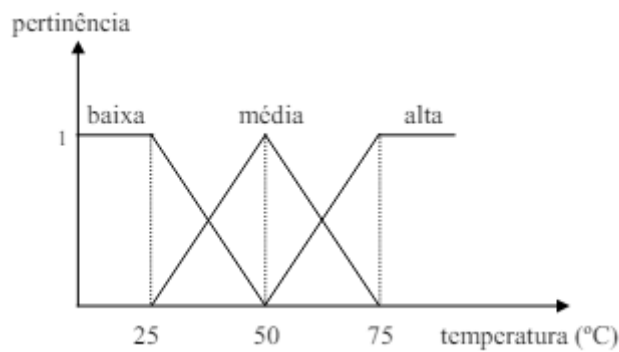
Dada uma variável de entrada que está sob análise, é necessário determinar o nível de pertinência daquela variável a um certo conjunto fuzzy. Este nível de pertinência, também chamado de grau de pertinência, é determinado por uma função de pertinência, da forma mostrada em (1), que determina o quanto um certo valor de uma variável pertence, ou está associado, a um conjunto fuzzy.

$$\mu_A(x): x \rightarrow [0, 1], \text{ onde } x \in X \quad (1)$$

Conforme (1), $\mu_A(x)$ é uma função de pertinência que determina o grau de pertinência de x , que está no universo de discurso X , em relação a um conjunto A .

As chamadas variáveis linguísticas são variáveis cujos valores são nomes de conjuntos fuzzy, sendo que estes podem assumir valores cuja relação com o valor unitário pode ser descrito pelas funções de pertinência (TANSCHKEIT, 2004). Os valores assumidos variam de 0 a 1, onde o nível um determina total pertinência àquele conjunto e o nível zero determina nenhuma pertinência àquele conjunto, com os valores intermediários indicando pertinência proporcional ao valor. Na Figura 4 tem-se como exemplo a variável linguística temperatura. No eixo horizontal estão os valores em graus celsius, enquanto no eixo vertical é apresentado o valor de pertinência. Nota-se, neste exemplo, que abaixo de 25 °C a temperatura é considerada baixa, com nível de pertinência 1 e, quanto mais ela se aproxima de 50 °C, menos “baixa” ela é, e mais “média” ela passa a ser considerada, até atingir o nível de pertinência unitário. A partir dos 50 °C, no entanto, cada vez menos ela é considerada “média” e cada vez mais ela é “alta”, até que a partir de 75°C ela é considerada totalmente alta, chegando ao nível de pertinência unitário.

Figura 4: Exemplo de funções de pertinência.



Fonte: Tanscheit (2004)

Pela Figura 4 também é possível notar que diferentes valores medidos podem ter valores não nulos para mais de uma função de pertinência. Em ambos os casos, onde apenas uma função de pertinência está ativa ou mais de uma está ativa, o sistema vai fornecer uma saída a depender das regras, do método de inferência e do método de defuzzificação utilizados.

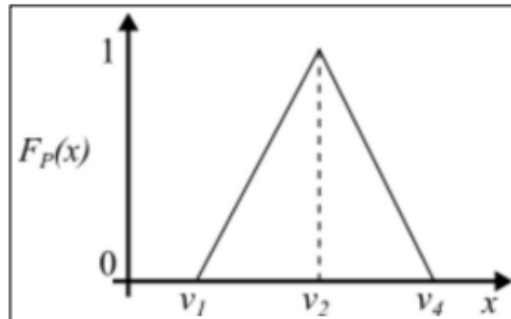
Existem diferentes tipos de funções de pertinência, sendo as principais a triangular, a trapezoidal e a gaussiana.

A função triangular é descrita conforme apresentado na equação (2), onde o valor de pertinência cresce linearmente a partir de a até chegar ao valor m , decrescendo linearmente a partir de m até o valor b . Para valores maiores que b e menores que a , o grau de pertinência é 0. É notável que se $a = m$ a função já se inicia decrescente e se $m = b$, a função termina crescente.

$$\mu(x) = \begin{cases} 0, & \text{se } x \leq a \\ \frac{x-a}{m-a}, & \text{se } x \in [a, m] \\ \frac{b-x}{b-m}, & \text{se } x \in [m, b] \\ 0, & \text{se } x \geq b \end{cases} \quad (2)$$

A Figura 5 mostra um esboço de gráfico de uma função triangular, onde, comparando com a equação (2), $a = V_1$, $m = V_2$ e $b = V_4$.

Figura 5: Esboço de função de pertinência triangular.



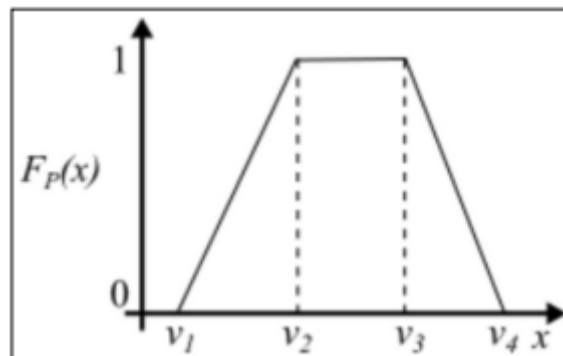
Fonte: Garcia (2020).

A função trapezoidal é parecida com a função triangular, porém apresentando um patamar constante em determinado intervalo. Na equação (3) é possível ver a descrição de uma função de pertinência trapezoidal, que é linearmente crescente do valor a até o valor m , é constante de m até n e de n até b é linearmente decrescente.

$$\mu(x) = \begin{cases} 0, & \text{se } x \leq a \\ \frac{x-a}{m-a}, & \text{se } x \in [a, m] \\ 1, & \text{se } x \in [m, n] \\ \frac{b-x}{b-n}, & \text{se } x \in [n, b] \\ 0, & \text{se } x \geq b \end{cases} \quad (3)$$

É notável que se $x = m$, a função se inicia constante e passa a decrescer após n e se $n = b$, a função se inicia crescente e termina constante. A Figura 6 mostra um esboço gráfico de uma função trapezoidal onde, comparando com a equação (3), $a = V_1$, $m = V_2$, $n = V_3$ e $b = V_4$.

Figura 6: Esboço de função de pertinência trapezoidal

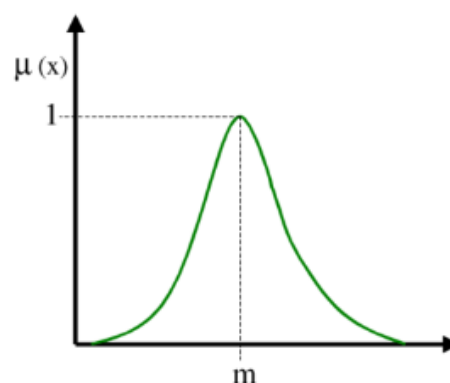


Fonte: Garcia (2020).

Já a função gaussiana é descrita pela equação (4). A Figura 7 apresenta um esboço de uma função de pertinência gaussiana.

$$\mu(x) = e^{-\frac{(x-m)^2}{2\sigma^2}} \quad (4)$$

Figura 7: Esboço de função de pertinência gaussiana.



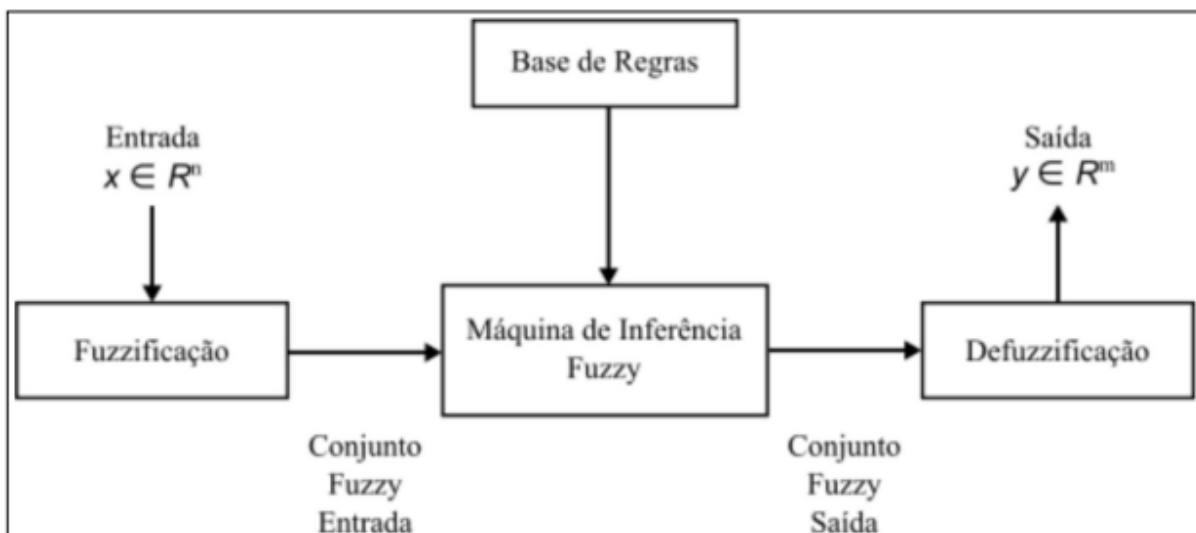
Fonte: Zuffo (2010).

2.4.2 Fuzzificação e Inferência fuzzy

A lógica fuzzy foi desenvolvida com base na linguagem e na estruturação de pensamento humanos. Um sistema de inferência fuzzy busca justamente se aproximar da forma de pensar de uma pessoa ao receber ou notar uma informação do ambiente externo. Assim como uma pessoa tem certos conhecimentos em sua memória e toma decisões com base neles, também os sistemas fuzzy possuem uma base de conhecimento, baseado em regras, por meio do qual uma informação de entrada (variável linguística de entrada) é tratada e é determinada uma saída, que é o sinal de controle (ZIMMERMAN, 1991). Assim, dentro da máquina de inferência, “cada proposição fuzzy é ‘traduzida’ matematicamente por meio das técnicas da lógica fuzzy” (GARCIA, 2020).

Conforme mostrado na Figura 8, a primeira etapa é a etapa de fuzzificação, onde ocorre a entrada da informação e ela é transformada no conjunto fuzzy de entrada. A etapa de fuzzificação “converte cada valor em uma ou mais partições fuzzy” (LIMA JUNIOR; CARPINETTI, 2016). Depois, este conjunto vai para a máquina de inferência que, baseado nas informações da base de dados, determina o conjunto de saída fuzzy, que passa pela etapa de defuzzificação, gerando o sinal de controle na saída.

Figura 8: Diagrama da estrutura básica de um sistema fuzzy.

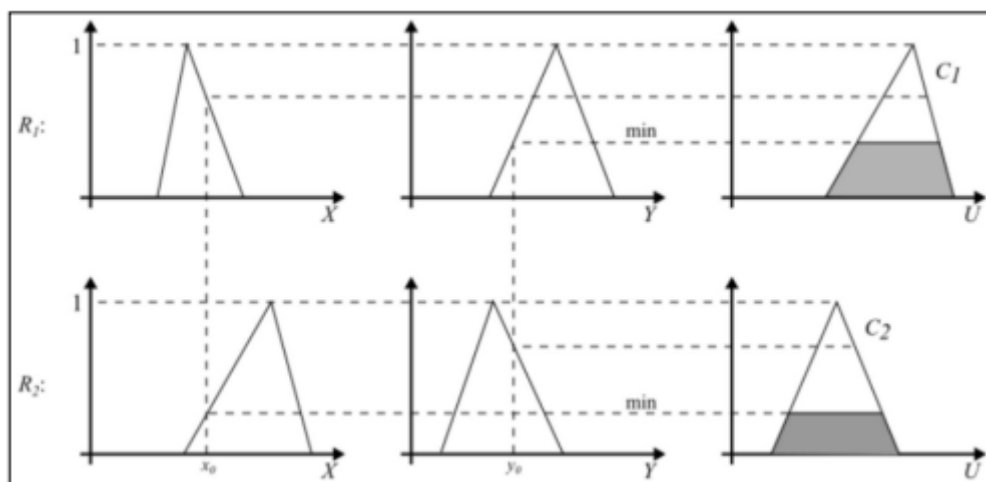


Fonte: GARCIA (2020).

Existem diferentes métodos de inferência fuzzy, sendo o mais consolidado o método Mamdani. Segundo Garcia (2020), uma regra da base de regras fuzzy é “definida pelo produto cartesiano fuzzy que compõe o antecedente (ex: se x é A) com o conseqüente (ex: então y é B)”. As operações lógicas realizadas podem variar de acordo com o método, mas neste caso são usados o operador de máximo para a operação OU, o operador de mínimo para a operação E e o operador complemento para a operação NÃO.

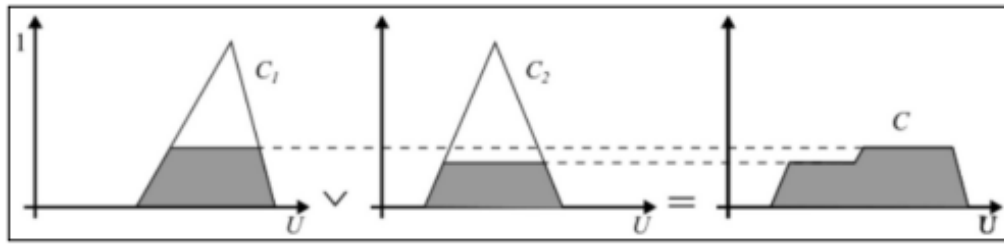
O método de Mamdani se baseia em observar, no método geométrico, onde a variável corta o eixo vertical, sendo este o grau de pertinência na função, podendo ter níveis de pertinência em mais de uma função de pertinência. Depois são aplicadas as t-normas e t-conormas. Se houver mais de uma função de pertinência ativa, é usado o menor valor. Com a matriz de associação fuzzy é feita a agregação das áreas das variáveis (VIEIRA, 2017). A Figura 9 mostra um exemplo de inferência com duas regras ativas, onde nota-se que é tomado para cada regra o menor valor dentre as funções ativas, o valor de intersecção entre elas, gerando o conjunto fuzzy de entrada. Já na Figura 10 é mostrado que das duas saídas obtidas por essas regras é tomada a maior, isto é, para os valores nos quais c_1 é maior, esta é tomada e nos valores nos quais c_2 é maior, esta é tomada, gerando a agregação que vai ser usada para produzir o sinal de saída final do controlador pelo método de defuzzificação.

Figura 9: Formação do conjunto fuzzy de entrada.



Fonte: Garcia (2020)

Figura 10: Composição da saída final.



Fonte: Garcia (2020)

2.4.3 Defuzzificação

A partir da região fuzzy obtida pela agregação, é utilizado o processo chamado de defuzzificação para transformar essa região fuzzy em um valor real que pode ser usado como sinal de controle. Para tal, podem ser realizadas diferentes operações para obter esse valor, a depender do método escolhido (GARCIA, 2022).

Alguns dos principais métodos utilizados são o método do centro de área (CDA), que é determinado pelo cálculo do centro de área da região formada pela agregação, o método da média dos máximos (MDM), que se baseia em utilizar a região onde os graus de pertinência são máximos e calcular a média, o método do primeiro máximo (MPM), que se baseia em escolher o valor onde o valor máximo aparece pela primeira vez (ROVEDA; ROVEDA; LOURENÇO, 2011), o método LOM (Largest of Maximum - Maior dos máximos), que foi o utilizado e se baseia em escolher o valor onde o máximo aparece pela última vez (BALBINOT; BARRETO, 2009; Sabounchi et al. 2011), entre outros métodos.

2.4.4 Características da implementação dos sistemas fuzzy

A utilização da lógica fuzzy para o desenvolvimento de controladores acarreta algumas características, que podem ser vantajosas em alguns casos ou desvantajosas em outros. Por exemplo, em sistemas muito complexos (sistemas não lineares ou imersos em variáveis incertas), pode haver um ganho de tempo considerável ao utilizar lógica fuzzy em relação a outros métodos tradicionais, como PID ou mesmo controladores booleanos (ON/OFF). Além disso, pode trazer uma necessidade de menos chips e sensores para sua implementação e facilitar a

modelagem em alguns casos de sistemas mais complexos (SANTOS, 2019), embora também possam ser sistemas incompletos, complexos demais ou mesmo inadequados para certos tipos de implementação de controladores em casos de sistemas mais simples e ambientes com pouca variação (VIEIRA, 2020).

Sabendo então que os controladores implementando este tipo de lógica podem ser mais interessantes para alguns casos, por serem mais simples de implementar e exigirem menos recursos, mas podem ser inadequados a outros, exigindo recursos específicos que poderiam ser substituídos por outros mais simples ou mesmo por não ser o que apresenta a melhor eficiência, buscou-se, neste projeto, descobrir se a utilização de controladores fuzzy na agricultura, especialmente no que diz respeito ao controle dos níveis de umidade do solo, é vantajoso em relação a outros métodos, como o método sem controlador (irrigação manual), irrigação baseada em timer em malha aberta, sem realimentação numérica do nível de umidade do solo, e controladores tradicionais, do tipo ON/OFF, este sim com realimentação de dados de umidade do solo, isto é, malha fechada.

2.5 EQUIPAMENTOS UTILIZADOS

2.5.1 Equipamentos, dispositivos e softwares

Para a realização do procedimento prático, são necessárias ferramentas virtuais, isto é, softwares, e também ferramentas físicas. Um dos softwares a ser utilizado no processo de desenvolvimento é o Matlab, desenvolvido pela MathWorks. Esse software possibilita a modelagem, o desenvolvimento matemático e a análise de dados com uma linguagem de programação, um conjunto de toolboxes e pode ser usado em diversas áreas dentro das engenharias em geral (OPENCADD, 2022). O próprio Matlab possui uma ferramenta específica para lidar com lógica Fuzzy, chamada Fuzzy Logic Toolbox, onde é possível desenvolver um sistema de controle com as regras da lógica fuzzy e depois exportar para a área de trabalho do próprio Matlab, onde fica disponível para ser utilizada (MATLAB, 2020). Também foram utilizados os softwares Arduino IDE, que é uma plataforma tanto para a escrita de código para implementação no Arduino quanto para a conexão do Arduino com o computador, além do MakeProto, uma plataforma online utilizada para converter arquivos do formato .fis, gerados pela toolbox fuzzy do Matlab, para

código C++ utilizado no Arduino.

Dentre os dispositivos físicos a serem utilizados estão a plataforma Arduino Uno, que possui o microcontrolador ATmega328 (ARDUINO, 2022), além de sensores de umidade do solo, válvulas solenóides para ativar e desativar a irrigação, e relés de atuação controlados pelo microcontrolador, tendo em vista que o arduino não tem condições de fornecer a corrente necessárias para ligar a solenóide, necessitando do relé conectado a uma fonte.

2.5.2 Microcontrolador

Os microcontroladores são dispositivos que possibilitam a implementação de um software de controle em um sistema embarcado ou com uso específico de modo barato e simples. Particularmente, é possível implementar diferentes formas de controladores que atuam de forma semelhante aos métodos convencionais de controle, ou também controladores que atuam de formas diferentes, via software. Isto é possível porque o microcontrolador é formado por partes menores que o compõem, podendo ser considerado um computador em um chip (SCHERZ, 2013).

Dentre os componentes que geralmente fazem parte de um microcontrolador estão a CPU (Central Processing Unit), que é a unidade onde ocorre o processamento, ROM (Read Only Memory), que é uma memória apenas de leitura, e RAM (Random Access Memory), que é uma memória usada durante a execução do programa.

O Arduino Uno não é apenas um microcontrolador, sendo na verdade uma placa de desenvolvimento, possuindo muitos outros componentes além dele, mas ele possui um microcontrolador dentre seus componentes, que é o ATmega328, mostrado na Figura 11, da fabricante Atmel (ATMEL, 2015).

Figura 11: Microcontrolador ATmega328



Fonte: Microchip Technology (2024)

Foi escolhido o Arduino Uno (ARDUINO, 2022) devido à sua facilidade de uso em comparação com outros microcontroladores, como os da família PIC, por exemplo, além da facilidade de acesso e integração com periféricos e ter grande quantidade de materiais e bibliotecas disponíveis já otimizadas, o que pode melhorar o desempenho. O Arduino Uno pode ser visto na Figura 12.

Figura 12: Arduino Uno



Fonte: Casa da Robótica (2024)

2.5.3 Componentes

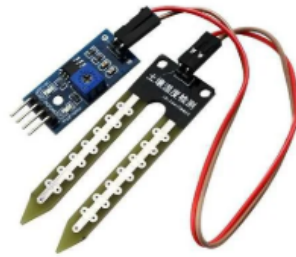
Foram utilizados componentes de sensoriamento e de atuação no sistema. Para o sensoriamento do solo, foi utilizado um sensor de umidade capacitivo FC-28, que pode ser visto na Figura 13. Este tipo de sensor é capaz de enviar um sinal digital, isto é, ele envia sinal lógico de nível alto ou baixo a depender da comparação do valor lido com o valor configurado, mas também realiza medição da umidade, enviando um valor de tensão elétrica proporcional ao nível de umidade, indo de 0V a 5V (FULLER, 2022), que pode ser lido por uma das entradas analógicas do Arduino.

A caracterização de um sensor deste tipo pode ser feita preparando um solo com umidade específica utilizando a equação (5) (NAGAHAGE, NAGAHAGE E FUJINO, 2019).

$$Umidade\ do\ solo\ (\%) = \frac{Massa\ de\ água}{Massa\ do\ solo} * 100 \quad (5)$$

Após a preparação de algumas amostras, é possível caracterizar o sensor já conhecendo previamente a umidade do solo pela equação (5).

Figura 13: Sensor de umidade do solo



Fonte: Eletrogate (2024)

Foi utilizado também um relé, mostrado na Figura 14, controlado pelo Arduino. De modo geral, microcontroladores não são capazes de fornecer correntes muito altas. Com o Arduino também é assim: ele consegue fornecer até 20 mA no máximo sem riscos de danos (BRAGA, 2018). Portanto, para acionar um dispositivo que precisa de corrente maior, é utilizado um relé, conectado a uma fonte, acionado pela porta de saída do Arduino, capaz de conduzir essa corrente.

O relé utilizado é da marca Songle com tensão de bobina de 5V segundo o fabricante, podendo suportar até 10 A de corrente em seus terminais com tensão de 250 V em corrente alternada e 30 V em corrente contínua.

Figura 14: Relé para acionamento

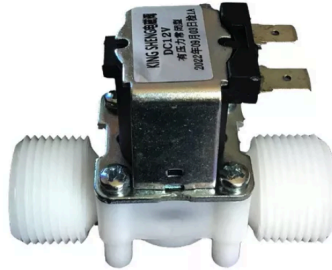


Fonte: Eletrogate (2024)

Para atuar na irrigação propriamente dita, foi utilizada uma solenóide que recebe o sinal vindo do relé, abrindo e fechando para a passagem de água de acordo com a instrução. A corrente de operação é 1 A e a tensão de operação é

12V em corrente contínua. A solenóide é mostrada na Figura 15.

Figura 15: Solenóide

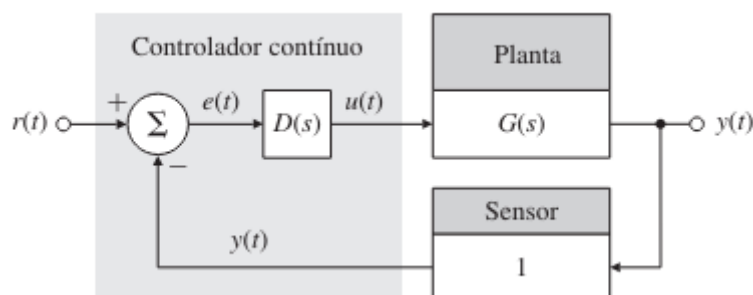


Fonte: Mercado Livre (2024)

2.6 FUNCIONAMENTO DE UM CONTROLADOR DIGITAL

A proposta do projeto implementado se baseia no diagrama convencional de um sistema de controle em malha fechada, isto é, um bloco de controle que recebe um sinal de um sensor, calcula a diferença para uma determinada referência e, com base no erro, atua sobre a planta enviando um sinal de controle que passa por um atuador. Um esquema simplificado de um sistema de controle analógico em malha fechada, ignorando ruídos e perturbações, é mostrado na Figura 16.

Figura 16: Esquema simplificado de um sistema controlado em malha fechada contínuo.

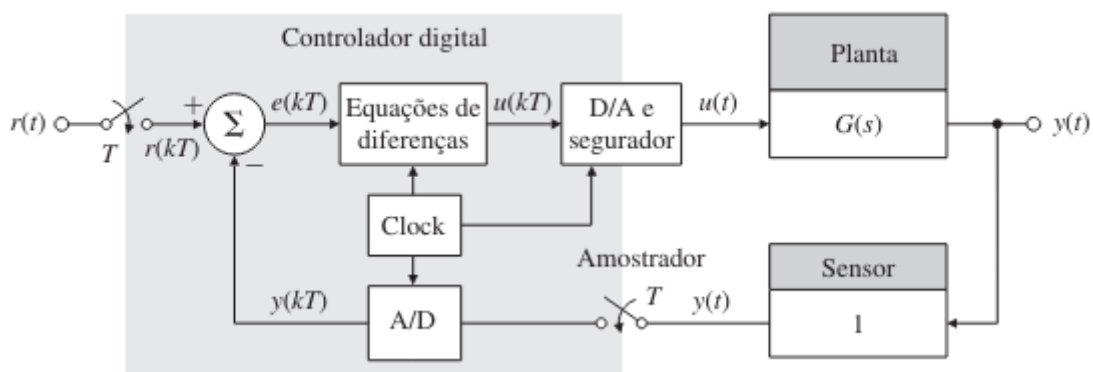


Fonte: Franklin (2013).

Na prática, o sistema implementado é um controlador digital, e não analógico, por usar um microcontrolador, de modo que existam algumas características próprias de um controlador digital que diferem de um controlador analógico

convencional, como conversor A/D para o sinal recebido do sensor, conversor D/A na saída (segurador) e gerador de clock (FRANKLIN, 2013). Apesar de a ideia do projeto ser justamente comparar diferentes formas de controle com um sistema fuzzy que não segue exatamente o processo tradicional de projeto de controlador, em particular neste caso um controlador digital, de modo a encontrar equações de diferença a partir de uma modelagem discreta e utilização de transformadas de laplace e z, mas sim projetar o controlador a partir da lógica fuzzy, ainda assim o diagrama mostrado na Figura 17 representa com bom grau de fidelidade, de forma simplificada e excluindo ruídos e perturbações do diagrama, o funcionamento de um controlador digital, que é o que os acionadores e controladores implementados na prática são.

Figura 17: Esquema simplificado de um sistema controlado digital em malha fechada.



Fonte: Franklin (2013).

Apesar de serem, portanto, controladores digitais, é possível realizar a implementação sem a necessidade de preocupação com parâmetros específicos do controle digital, como frequência de amostragem, pois o recebimento dos dados dos sensores ocorre em um tempo relativamente grande para um sistema microprocessado, na ordem de minutos, enquanto microcontroladores são capazes de operar com tempos de amostragem na ordem de milissegundos, ou seja, o tempo de amostragem é suficientemente grande para não causar sobreamostragem. Também não é necessário a preocupação com conversão A/D e seguradores na saída (D/A), pois o próprio microcontrolador da plataforma Arduino

já realiza este processo.

2.7 COMPARAÇÃO ENTRE SISTEMA FUZZY E PID

O uso de controladores PID na irrigação é, tradicionalmente, diferente do uso de outros tipos de acionamento. Enquanto formas como o timer sem realimentação, o controle liga-desliga e até mesmo o fuzzy são utilizadas em grandes áreas de cultivo, ao ar livre e de diferentes culturas (GARCIA, 2020; BORDIN, 2016; GODINHO; GASPAROTO; CANEPPELE, 2022), o uso do PID se mostra mais restrito a casos onde há pouca interferência externa, o ambiente e suas variáveis são mais estáveis e a área de atuação é menor, como os casos de hortas indoor e até mesmo jardinagem (RESENDE, 2014). A principal explicação para isso é que sistemas abertos, com muitas mudanças nas variáveis de ambiente e longas extensões de aplicação, onde podem ocorrer diferenças de valores de uma mesma variável dentro do mesmo sistema, a depender do local onde a medição ocorre, são bem mais difíceis de serem modelados matematicamente (GOODCHIELD, 2015), o que justifica o uso do fuzzy, que tem como premissa ter uma modelagem baseada em normas linguísticas, de fácil compreensão humana (TANSCHKEIT, 2004). Desta forma, tanto a implementação quanto os estudos do uso do PID na irrigação são voltados para sistemas simples, em pequena escala e usando simplificações em seus parâmetros (GOODCHIELD, 2015; SERVULO; LEITE; DURÇO, 2019).

Como a comparação entre fuzzy e PID em um sistema de irrigação faz pouco sentido, já que são implementados geralmente com usos diferentes, buscou-se a comparação em outras áreas para se ter uma ideia de nível de desempenho. De modo geral, as duas formas costumam apresentar desempenhos semelhantes. Segundo o experimento conduzido por Masselli e Silva (2014), o controle baseado em fuzzy apresentou um desempenho melhor, segundo os autores, do que um PID implementado nas mesmas condições. Apesar de ter apresentado um tempo de subida mais lento, o tempo de acomodação e o percentual de sobressinal foram menores no sistema fuzzy. Outra grande vantagem apresentada por Masselli e Silva é que, apesar de ambos os métodos, PID e fuzzy, terem precisado de reajustes em seus parâmetros e configurações, o fuzzy apresentou uma facilidade de parametrização e desenvolvimento notável, o que poupa muito tempo de

desenvolvimento, ou seja, segundo este estudo, além de ter apresentado uma resposta melhor, nas palavras dos próprios autores, do que o PID, o sistema fuzzy também foi notavelmente mais fácil de ser desenvolvido. Isso não significa que essa avaliação se estende a todas as aplicações, mas é sim possível afirmar, com base neste estudo, que o fuzzy tende a ser mais simples de ser desenvolvido e apresentar desempenho semelhante, podendo, inclusive, ser melhor do que o PID em algumas aplicações, conforme visto.

Outra comparação importante é a do PID com os controladores liga-desliga e com timer sem realimentação especificamente na área de irrigação. Neste trabalho foram implementados na prática controladores destes dois último tipos e, ao comparar os três na área de irrigação especificamente, apesar de todas as ressalvas já pontuadas em relação aos diferentes usos do PID e do fuzzy, é possível fazer uma análise de desempenho entre os dois nesta área. Servulo, Leite e Durço (2019) desenvolveram um controlador PID para irrigação aplicado em um sistema para teste, e neste caso, conforme os autores, o desempenho foi muito parecido com o de um controlador *on-off*. Já em relação ao controle com timer, este apresentou períodos de encharcamento, assim como foi notado no presente trabalho, o que leva a umidade a sempre aumentar, enquanto o PID não apresentou este problema. Ainda segundo os autores, ocorreram problemas na resposta dada de acordo com a leitura dos sensores, o que corrobora com os demais estudos, que apontam a dificuldade de modelar um sistema e obter uma boa resposta com o PID nesta área. Assim, o PID mostrou um desempenho melhor que o acionamento com timer, porém muito próximo do controle liga-desliga realimentado, que foi, de modo geral, inferior ao fuzzy na implementação prática realizada neste trabalho.

Por fim, é válido ressaltar que, apesar do fuzzy ter apresentado melhor desempenho e facilidade de desenvolvimento em outras áreas e as indicações mostrarem a probabilidade de o fuzzy ter melhor desempenho na área de irrigação também, haja vista que Servulo, Leite e Durço (2019) mostraram que o PID se mostrou próximo ao *on-off*, que foi inferior ao fuzzy na prática deste trabalho, ainda assim é importante levar em conta o principal aspecto observado desde o início da pesquisa: a aplicação destes métodos são diferentes na área de irrigação. Assim, é possível combinar as duas formas de controle, de modo a criar um sistema fuzzy-PID, conforme realizado por SIJIA et al (2021), capaz de apresentar

resultados mais rápidos, com tempo de assentamento até 2,5 segundos menor e sobressinal até 23,3% menor do que um sistema fuzzy isolado que, conforme já visto, é mais utilizado do que o PID em sistemas deste tipo por motivos de desempenho e facilidade de desenvolvimento já mencionados anteriormente, além do sistema misto fuzzy-PID apresentar uma resposta a interferência melhor que qualquer uma das demais formas de controle isoladamente, tanto fuzzy quanto PID, conforme verificado em trabalhos desenvolvidos para a diminuição do consumo de água na irrigação e plantio do arroz.

Importante ressaltar que o projeto desenvolvido por Sijia et al não é igual à forma de controle fuzzy que utiliza como entradas os valores e suas derivadas. Esta é uma técnica já utilizada em vários casos onde, além do valor lido, é usada como entrada também a derivada ou a diferenciação no tempo, comparando a entrada atual com as entradas anteriores. Entretanto, o trabalho citado foca principalmente na atuação conjunta de controle fuzzy e PID, sendo uma abordagem diferente dessa.

3 DESENVOLVIMENTO DO TRABALHO

O trabalho foi desenvolvido seguindo uma sequência de etapas com o objetivo de comparar diferentes formas de acionamento e controle de irrigação com um sistema usando um controlador fuzzy. A primeira etapa foi a pesquisa bibliográfica para a compreensão do estado atual dos sistemas de irrigação, o que vem sendo implementado na área e quais as principais opções disponíveis. Em seguida, com base na pesquisa, foram escolhidos quais controladores seriam comparados com o controlador fuzzy, seguindo critérios apresentados adiante neste trabalho. Após essa escolha, foram determinados quais controladores seriam implementados na prática e quais seriam comparados por meio de revisão bibliográfica.

Com todas as decisões prévias tomadas, foram então desenvolvidos e implementados os sistemas a serem comparados e, simultaneamente, foram realizadas as pesquisas para a comparação feita por revisão bibliográfica.

Ao término da implementação e da pesquisa, os resultados foram comparados de acordo com a melhor forma de comparação, seja ela qualitativa, quantitativa ou ambas as formas.

3.1 ESCOLHA DOS CONTROLADORES COMPARADOS E IMPLEMENTADOS

Na etapa de pesquisa bibliográfica inicial foram identificados quais os principais métodos utilizados para o acionamento em sistemas de irrigação. Conforme já visto anteriormente no Item 2.3 deste trabalho, encontram-se entre eles os métodos manuais que, apesar de serem cada vez menos implementados (DURSON & OZDEN, 2011), ainda são utilizados principalmente em pequenas propriedades de cultivos de hortaliças, os métodos automatizados utilizando timer sem realimentação e o controlador liga-desliga em malha fechada (MEDEIROS, 2018).

O controlador PID também é utilizado na área de irrigação, mas geralmente com uma abordagem diferente, sendo mais usado em sistemas onde há maior controle das variáveis do ambiente de modo geral, sendo geralmente usado em sistemas *indoor*, jardinagem ou pequenas culturas (RESENDE, 2014). Embora ele

possa ser desenvolvido para culturas extensivas, este não é seu principal uso.

Já os métodos de controle baseados em sistemas de I.A mais recentes e que exigem nível de processamento computacional maior ou bases de treinamento com muitas informações, como algoritmos genéticos e redes neurais, ainda seguem em fase de desenvolvimento no que diz respeito à irrigação e, embora já sejam amplamente utilizados em outros setores, como classificação de padrões de imagens aéreas na agricultura, nesta área de acionamento de irrigação ainda é pouco implementado em comparação com os demais métodos e os principais artigos e estudos deste tipo de controle são baseados mais em simulações do que implementações práticas, conforme visto no Item 2.3.

A proposta do trabalho é comparar diferentes tipos de controladores com o controlador fuzzy, sendo alguns implementados na prática para a obtenção de resultados e outros, cujos resultados e comparações são apresentados em estudos e trabalhos já realizados, feitos pela pesquisa bibliográfica. Assim, com as informações dispostas anteriormente, foram escolhidos os métodos de acionamento com timer, controlador liga-desliga (on-off) e o fuzzy para serem implementados na prática e serem comparados quantitativamente.

Para o controlador PID foi escolhida uma comparação qualitativa, realizada e apresentada na fundamentação teórica, no item 2.7, baseado em revisão bibliográfica, tendo em vista que ele e o fuzzy não são usados, de modo geral, nos mesmos tipos de irrigação, sendo o fuzzy mais adequado para sistemas mais complexos, extensivos e abertos devido à sua facilidade de modelagem, enquanto o PID geralmente é usado em sistemas menores ou fechados, tendo um melhor desempenho nessas condições. Portanto, a comparação quantitativa por si só não faria sentido, pois deixaria de considerar muitos aspectos próprios da implementação de cada um.

Já em relação à comparação com a irrigação manual, esta também não foi implementada pois já existem estudos publicados na área. Neste caso, a ideia foi comparar tanto quantitativamente, observando os níveis de desempenho, quanto qualitativamente, levando em conta que a irrigação manual é um trabalho que pode ser árduo muitas vezes e pode levar a prejuízos físicos em alguns casos.

Por fim, sistemas como redes neurais, conforme visto, são pouco utilizados na prática na área de irrigação e, portanto, não foram considerados para a comparação.

3.2 ANÁLISES QUALITATIVA X QUANTITATIVA

A análise quantitativa busca levar em consideração valores medidos e desempenho, comparando numericamente, sob as mesmas condições de implementação, os resultados obtidos. Assim, na análise quantitativa, buscou-se comparar dados, como o consumo de água, aliados a critérios de desempenho dos controladores, como níveis de sobressinal, tempo de assentamento e erro de regime permanente, que são critérios de desempenho clássicos amplamente utilizados nos sistemas de controle (OGATA, 2011).

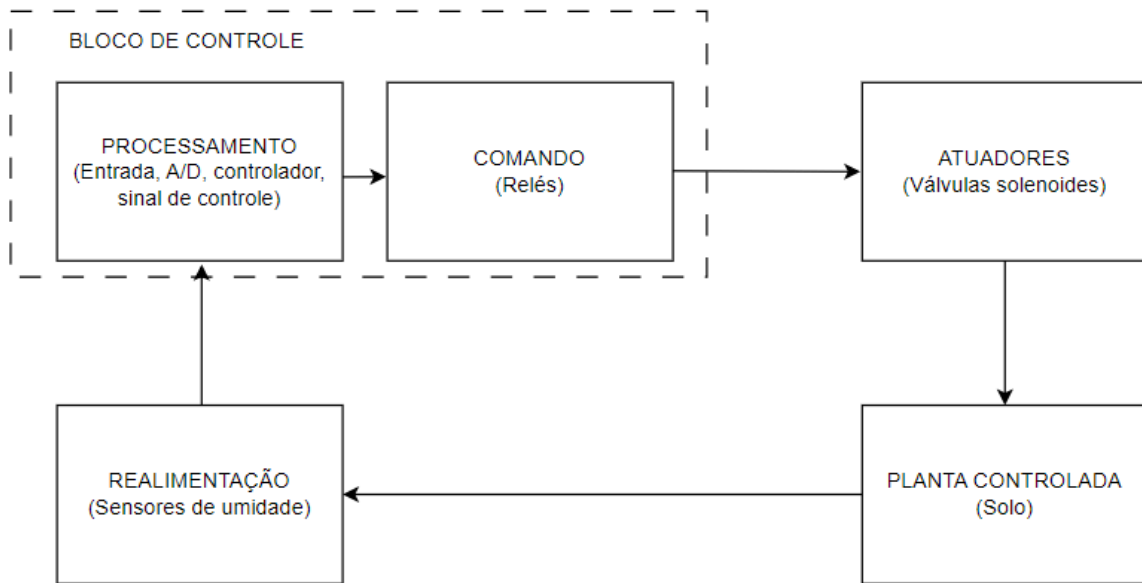
Já a análise qualitativa envolve critérios claros, porém não numéricos, e é utilizada em casos onde a análise estatística e dados numéricos não são úteis, ou suficientes, para compreender a questão em sua totalidade, tendo aspectos mais profundos (MATHIAS, 2022). Nestes casos são levados em consideração aspectos como os locais e principais formas de uso de cada sistema, o nível de ampla utilização, o nível de impacto sobre as pessoas envolvidas com aquele sistema, entre outros possíveis critérios que diferem um sistema do outro sem necessariamente utilizar critérios numéricos de desempenho.

3.3 DESENVOLVIMENTO DOS CONTROLADORES

O diagrama geral dos sistemas de controle montados é mostrado na Figura 18. Todos seguem a mesma estrutura de blocos, composta pelo bloco de processamento, que na implementação física é feita pelo Arduino Uno, e pelo bloco de comandos, exercidos pelo relé, haja vista que o Arduino Uno não consegue entregar corrente o bastante para acionar as válvulas. É no bloco de processamento que serão implementados os comandos e códigos para o controle propriamente dito.

O bloco dos atuadores é onde está o sistema de acionamento da irrigação, neste caso uma válvula solenóide. A planta controlada são as amostras de solo e o bloco de realimentação é composto principalmente pelos sensores de umidade do solo e seus drivers.

Figura 18: Diagrama de blocos do sistema controlado implementado.



Fonte: Autoria Própria (2024).

Para a implementação real, foi considerada a implementação de diferentes variáveis que poderiam ser analisadas além da própria umidade do solo e que comumente aparecem na literatura a respeito de irrigação com sistemas fuzzy, como a temperatura (AZMI, 2022), sensor de chuva, textura e tipo de solo, fase de desenvolvimento da cultura (OLIVEIRA e IMMICH, 2023), entre outras variáveis. Porém, como o objetivo deste trabalho é comparar unicamente os resultados obtidos por diferentes métodos de controle sob as mesmas condições e com as mesmas variáveis e não desenvolver um controlador completo para implementação, optou-se por levar em conta como única variável de entrada a umidade do solo, que é de fácil verificação e torna os sistemas a serem comparados menos suscetíveis a variações de algoritmo, haja vista que sistemas que tem muitas variáveis de entrada tendem a depender, muitas vezes, principalmente do algoritmo utilizado para criar as regras e lógicas internas de tratamento destas variáveis, dificultando a avaliação do tipo de sistema de controle utilizado propriamente. Com várias variáveis de entrada, um mesmo controlador pode ser configurado de diversas formas, sendo que não necessariamente a melhor dentre elas seria a escolhida para ser comparada com outras formas de controle. Desta forma, optou-se por ter uma única

entrada, o que tende a fazer com que a comparação seja menos enviesada em relação à construção do algoritmo de controle, sendo esta uma abordagem que já tem respaldo na literatura e pesquisa de comparação de controladores fuzzy para irrigação com outras formas de controle de irrigação, sendo às vezes utilizada apenas em conjunto com um controle de tempo ou horário (GARCIA, 2020), que foi utilizado também neste trabalho.

3.3.1 Testes e montagem dos controladores

Cada método de acionamento implementado na prática foi desenvolvido segundo critérios encontrados na literatura e baseados em outros trabalhos de desenvolvimento parecidos. Antes da implementação de cada tipo de acionamento foram realizados testes e simulações dos controladores e sistemas de acionamento, além de testes de vazão de água e tempo de absorção de água pelo solo. Esses testes foram conduzidos de modo que a água fosse depositada por 15 segundos, que foi o tempo pré-estipulado, e foi realizado um acompanhamento para ver como era a absorção de água pelo solo, de modo que o ideal seria uma absorção total em até 15 minutos. Em alguns testes, a vazão muito baixa se mostrou ineficiente, sendo absorvida muito rápido, havendo folga para um depósito maior de água, o que melhoraria o desempenho do processo, enquanto em outros casos onde a vazão foi maior, houveram encharcamentos e absorção muito lenta. Após as etapas de testes, foi concluído que, para uma amostra de 10 kg de terra, que foi o padrão utilizado em cada teste, a vazão de 1,2 litro de água em 15 segundos e um tempo de absorção de 10 a 15 minutos era o suficiente para o solo não encharcar e poder absorver totalmente a água depositada, mesmo com níveis de umidade mais altos, acima de 60%, ou até mesmo próximos da saturação. Assim, para haver tempo o bastante, foi determinado o acionamento, ou possibilidade de acionamento, em intervalos de 30 minutos, que é pelo menos o dobro do tempo necessário para a absorção da água pelo solo. Sendo o ideal em solo arenoso, que foi o utilizado, para plantio, uma umidade de 60% a 80% (CRUZ et al, 2011), foi determinado um *setpoint* de 70%, dentro da faixa indicada, para todos os controladores. Todos atuaram sobre amostras coletadas do mesmo local, no mesmo instante e com o mesmo percentual de umidade inicial, que foi 10%.

3.3.2 Controlador liga-desliga com realimentação

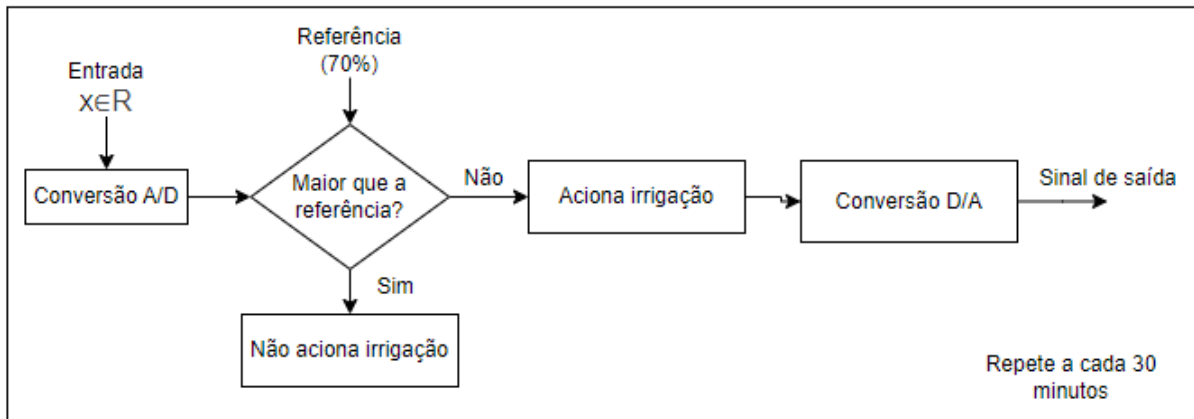
Este é um controlador com realimentação de informação de umidade, que é capaz de identificar se deve acionar a irrigação ou não a depender do nível lido pelo sensor, isto é, se a umidade estiver abaixo do valor de *setpoint*, ele aciona a irrigação e se estiver igual ou acima do valor de referência, ele não aciona. Ele é chamado de liga-desliga, ou *on-off*, justamente por ter apenas dois níveis, sem nenhum tipo de valor intermediário (OGATA, 2011).

Antes da implementação, foi definido que o tempo de abertura da válvula seria de 15 segundos e durante as fases de testes com este controlador, foi determinado que a melhor vazão era de 1,2 l de água nos 15 segundos de abertura, de modo que vazões menores que essa não apresentavam o melhor rendimento possível do controlador, tendo uma resposta mais lenta e podendo causar uma análise de desempenho viciada e vazões maiores apresentavam uma demora maior na absorção da água pelo solo, o que poderia levar a problemas na medição de umidade, principalmente depois que o solo já estivesse com maior nível de saturação de água e a absorção fosse ainda mais lenta - particularmente, acima dos 60% de umidade, conforme verificado nos testes, além de aumentar consideravelmente o nível de sobressinal, o que prejudicaria o desempenho do controlador de forma proposital. Portanto, tempos maiores que 15 segundos tornaram a resposta mais rápida, porém com desempenho prejudicado em outros aspectos.

Para a implementação deste controlador foi feito um código em C++ para ser implementado no Arduino, que pode ser encontrado no Apêndice A. Este código realiza ciclos de 30 minutos, tempo este explicado no item 3.3.1, de modo que 5 minutos antes do término do ciclo é realizada a medição da umidade e esta medição é enviada via serial para ficar disponível para o usuário e. Depois, ao final do ciclo, é feita uma nova medição e se esse valor lido for menor que a referência, isto é, 70% de umidade, então o controlador aciona a válvula (liga) por 15 segundos, depositando 1,2 l de água, e fecha a válvula logo em seguida (desliga). Após isso, é iniciado um novo ciclo que opera com a mesma lógica.

A Figura 19 apresenta um diagrama de funcionamento do acionamento com liga-desliga.

Figura 19: Diagrama de funcionamento de um controlador on/off.



Fonte: Autoria própria.

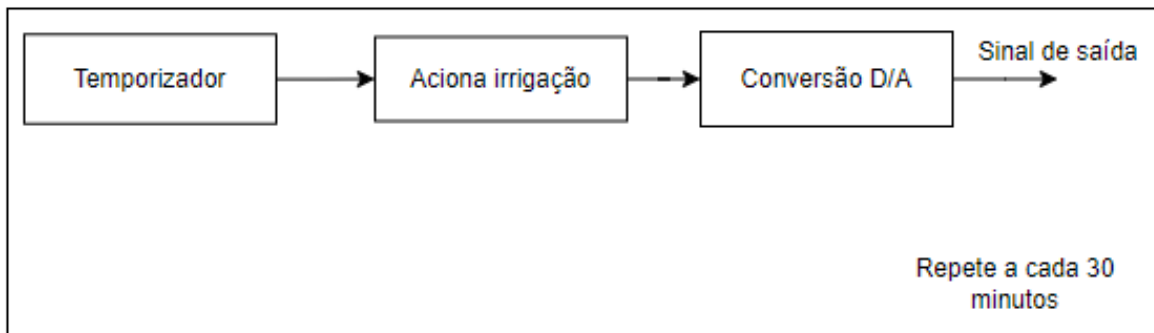
3.3.3 Acionador com timer sem realimentação

Possui parâmetros muito parecidos com o controlador do item 3.3.2, porém o acionamento é determinado unicamente pelo tempo de ciclo determinado (timer), não levando em conta o valor lido de umidade (sem realimentação). O código implementado é apresentado no Apêndice B. Neste caso, o valor lido de umidade não interfere no acionamento da válvula, sendo esta acionada sempre ao término de cada ciclo de 30 minutos. A leitura do valor de umidade é feita apenas para monitoramento via serial.

Ao fim de cada ciclo, a válvula é acionada por 15 segundos, depositando 1,2 l de água no solo, independentemente do valor de umidade que o solo já tenha. Este sistema de acionamento foi desenvolvido com base no controlador apresentado anteriormente, porém desabilitando o comando condicional que decide pela atuação ou não.

A Figura 20 apresenta o diagrama de funcionamento de um acionador com timer em malha aberta.

Figura 20: Diagrama de funcionamento de um acionador com timer em malha aberta.



Fonte: Autoria própria.

3.3.4 Controlador fuzzy

O controlador usando lógica fuzzy foi desenvolvido com a utilização da toolbox Fuzzy Logic do Matlab, que possibilita a implementação de um sistema fuzzy de forma simples, sem a necessidade de realizar via código diretamente esta implementação, o que agiliza o desenvolvimento e facilita nos processos de simulação, já que é fácil alterar os parâmetros para testar diferentes controladores desenvolvidos. É perfeitamente possível implementar um controlador fuzzy diretamente em código, sem a necessidade do Matlab, porém a utilização deste software facilita o desenvolvimento.

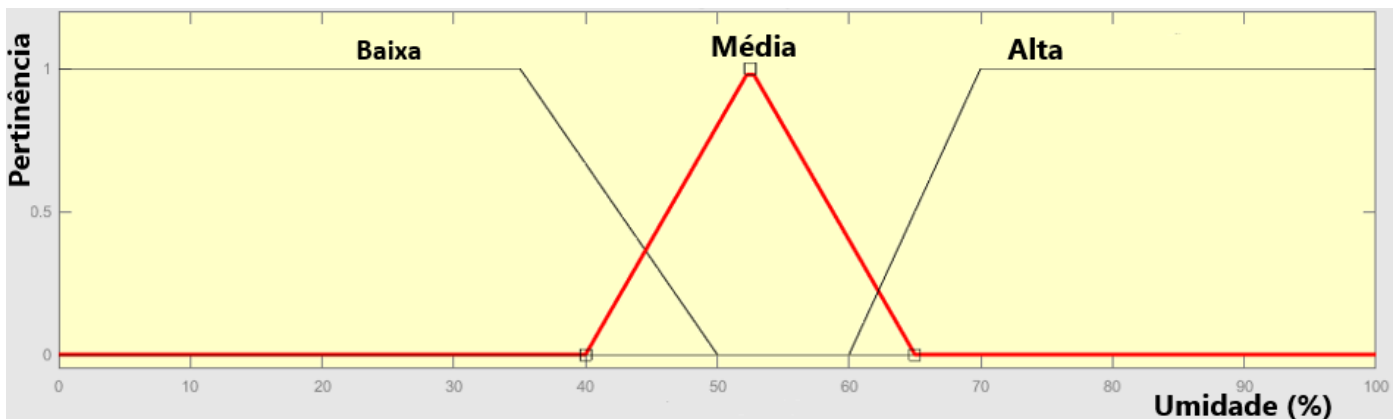
Conforme observado na literatura comparativa entre sistemas fuzzy e outras formas de irrigação, foram utilizadas três funções de pertinência para o nível de umidade do solo, sendo duas trapezoidais e uma triangular, e um controle horário para irrigação, isto é, um controle de tempo para determinar em qual momento poderia ocorrer a irrigação (GARCIA, 2020). Tendo em vista o valor de referência como 70% de umidade, conforme utilizado nos outros sistemas, foram simuladas as três funções de pertinência, alterando seus valores, até encontrar o que apresentou melhores níveis de desempenho de velocidade e sobressinal. Assim, as funções de pertinência que apresentaram o melhor desempenho, combinando velocidade e sobressinal, apresentaram os seguintes parâmetros:

- baixa: função trapezoidal iniciada já em nível constante (nível unitário) a partir de zero, decrescente a partir de 35% e nula a partir de 50% de umidade.

- média: função triangular crescente a partir de 40%, máximo em 52,5% (nível unitário) e decrescente até 65%. Para os demais valores, é nula.
- alta: função trapezoidal nula até 60%, crescente até 70% e constante (nível unitário) a partir de 70%.

As funções de pertinência são mostradas graficamente na Figura 21.

Figura 21: Funções de pertinência para a variável de entrada



Fonte: Autoria própria.

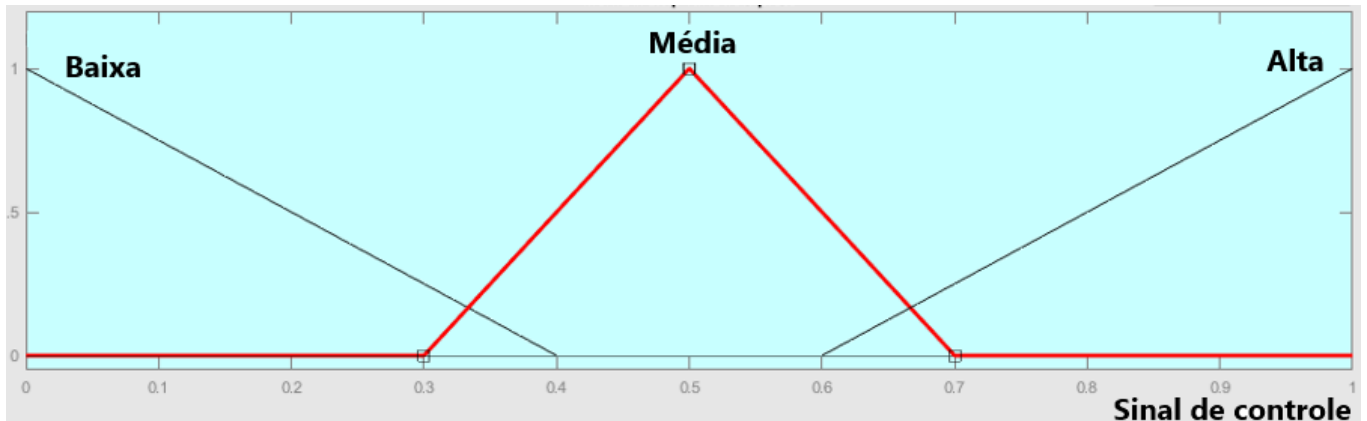
O controle de tempo é o mesmo utilizado para os demais, ou seja, ciclos de 30 minutos.

Para a saída, também seguindo o referencial bibliográfico, foram implementadas três funções, sendo as três triangulares. Para a determinação da saída, também foram realizadas simulações, de modo a obter o melhor desempenho com as seguintes funções:

- baixa: função triangular que inicia decrescente a partir do zero, indo até 0,4, e sendo nula a partir deste valor.
- média: função triangular crescente a partir de 0,3, atinge o máximo (nível unitário) em 0,5 e decresce até 0,7. É nula nos demais valores.
- alta: função triangular nula até 0,6, crescente a partir deste valor até nível máximo (nível unitário) em 1.

As funções de saída são mostradas graficamente na Figura 22.

Figura 22: Funções de saída.



Fonte: Autoria própria.

O método de inferência utilizado foi o Mamdani e as regras utilizadas foram:

- Se o nível de umidade for baixo, a saída é alta;
- Se o nível de umidade for médio, a saída é média;
- Se o nível de umidade for alto, a saída é baixa.

Também foram realizados testes de simulação para determinar o melhor método de defuzzificação, e o que apresentou melhores resultados foi o LOM (Largest of Maximum), que encontra o valor de saída escolhendo o último valor, isto é, o maior valor que apresenta um máximo de saída. Este método de defuzzificação foi escolhido após verificar que para outros métodos, mesmo com valores de umidade muito acima de 70%, continuava havendo a ativação de regiões de saída não nulas, o que resultava em acionamento mesmo com valores altos de umidade. Assim, seria necessário alterar muito as funções de pertinência para resolver este problema, gerando problemas na atuação do controlador. Assim, o método LOM conseguiu resolver este problema mantendo as funções de pertinência adequadas.

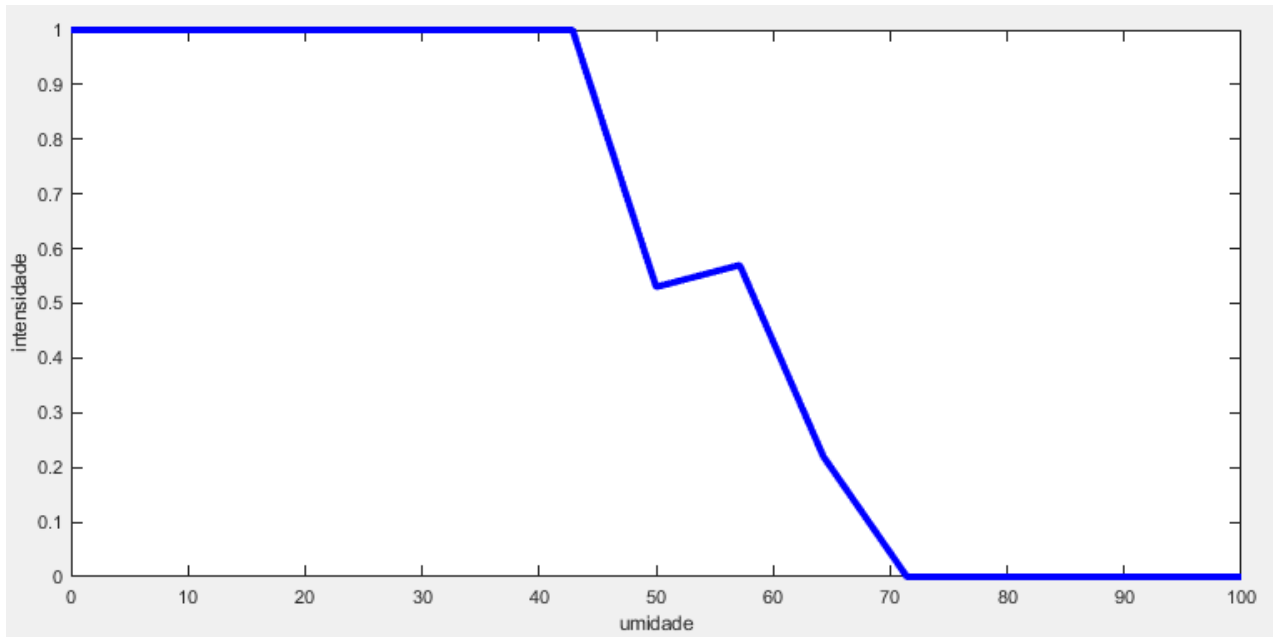
Para a saída, foi utilizada a mesma vazão dos demais sistemas, sendo esta 1,2 l em 15 segundos, porém com a possibilidade de aumento do tempo de abertura da válvula. Como o tipo de acionamento utilizado é apenas aberto ou fechado, sem controle de nível de abertura, então, ao invés de permitir a abertura por um tempo fixo de 15 segundos com controle de nível de abertura, foi utilizada uma vazão

constante com controle de tempo de abertura. Assim, a depender da decisão do controlador, a abertura da válvula pode variar de 0 s a 25 s. Isso ocorre multiplicando o valor de saída, que varia de 0 a 1, pelo tempo máximo de abertura.

Conforme visto nos demais acionadores implementados, o tempo de abertura era fixo de 15 segundos, enquanto no caso do fuzzy esse tempo é variável e pode chegar até a 25 segundos. Este valor foi determinado levando em conta o desempenho do fuzzy, que é capaz de alterar este tempo, diferentemente dos demais. Assim, se fosse usado um tempo de 25 segundos no controlador liga-desliga realimentado, por exemplo, ele apresentaria um sobressinal muito alto, além de apresentar encharcamento quando o nível de umidade estivesse muito alto, mais próximo da saturação, acima de 60%, por exemplo, conforme detectado em testes, o que poderia viciar a comparação, o que não acontece no fuzzy, já que ele diminui o tempo de irrigação conforme o nível de umidade do solo é maior, diminuindo, ou evitando por conta própria os possíveis sobressinais.

A Figura 23 apresenta a superfície de resposta do sistema desenvolvido para a implementação, que traz a resposta que o controlador fuzzy dará a depender do nível de umidade lido, onde o tempo em segundos será o produto do tempo máximo de irrigação, isto é, 25 segundos, com a intensidade de saída, que varia de 0 a 1. Pela figura, nota-se que até o valor de 43% de umidade, o acionamento é total, com intensidade 1. Depois disso a intensidade diminui, apresentando menor tempo de irrigação conforme a umidade é maior, exceto por um intervalo entre 50% e 58% de umidade decorrente da forma de operação do controlador, até que a partir de 71% de umidade o acionamento é totalmente nulo, isto é, não há mais irrigação.

Figura 23: Superfície de resposta fuzzy.



Fonte: autoria própria.

Depois de desenvolvido o controlador com a toolbox Fuzzy Logic do Matlab, o arquivo gerado foi exportado no formato .fis e com o auxílio do software MakeProto, o arquivo gerado foi convertido em código C++ para a implementação no Arduino. Esse software converte apenas o controlador, então foi necessário editar o código para configurar as portas de entrada, saída e tempo de atuação de cada ciclo. O código final implementado está disponível no Apêndice C.

3.3.5 Parâmetros gerais de testes e implementação

Os parâmetros de operação dos controladores obtidos nas etapas de teste e simulação são apresentados na Tabela 2, onde o tempo se refere ao tempo de abertura da válvula para irrigação, a vazão é a vazão de água e a referência é o nível de umidade usado como referência nos casos de realimentação.

Tabela 2: Parâmetros gerais dos controladores.

	Fuzzy	On/Off	Timer
Tempo	0s a 25s	15s	15s
Vazão	80ml/s	80ml/s	80ml/s
Referência	70%	70%	Sem referência

Fonte: Autoria própria.

3.4 PESQUISA E COMPARAÇÃO BIBLIOGRÁFICA

A etapa de pesquisa e comparação bibliográfica foi feita por meio da busca de artigos e publicações voltados principalmente para a área de irrigação. O objetivo era encontrar diferentes formas de controle comparados com o fuzzy, de modo a ser possível a comparação de desempenho entre diferentes formas de acionamento na área.

Durante a etapa de pesquisa, foram encontrados estudos e publicações comparando o fuzzy com o sistema de irrigação manual, realizado por um especialista humano, sem auxílio de tecnologia, trazendo informações sobre o desempenho de cada um na cultura da alface crespa (GARCIA, 2020).

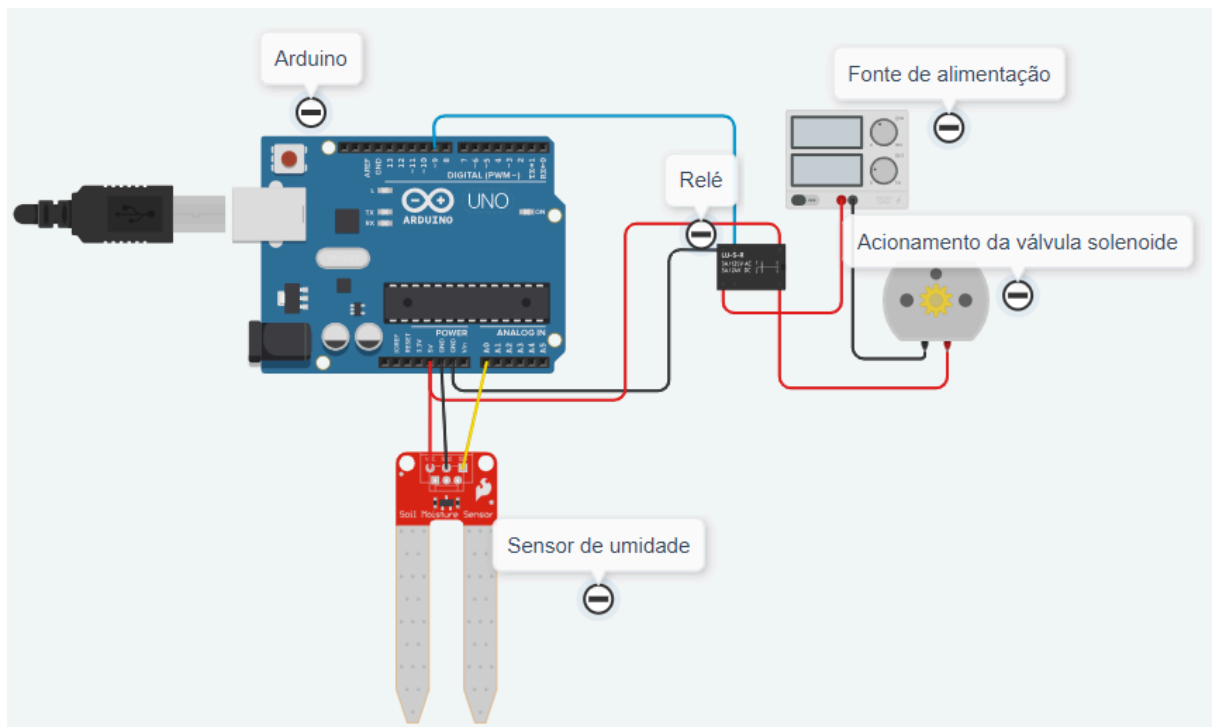
3.5 IMPLEMENTAÇÃO PRÁTICA

Todos os circuitos de controle e acionamento tiveram a mesma montagem, que é mostrada na Figura 24, diferenciando-se um do outro pelo código implementado. Em todos eles, o Arduino fica conectado com um computador que alimenta a placa e recebe informações de medições via serial. O Arduino alimenta o sensor de umidade, já caracterizado na fase de testes de acordo com a equação (5) do item 2.5.3, e este realiza as medições e envia um valor de tensão de acordo com a umidade do solo. Esse valor é lido pela porta analógica A0 do Arduino.

Como o Arduino não é capaz de fornecer a tensão e a corrente necessárias para acionar a válvula solenóide, isto é, 12 V e 1 A, foi utilizado um relé para acionar a válvula quando necessário. Assim, o Arduino alimenta o relé e envia o sinal pela porta 9 para fechar o contato no relé e realizar o acionamento. O relé, por sua vez, está conectado a uma fonte capaz de acionar a válvula e, ao receber o sinal do

Arduino, aciona a solenóide, permitindo a passagem de água.

Figura 24: Esquemático da montagem dos circuitos.

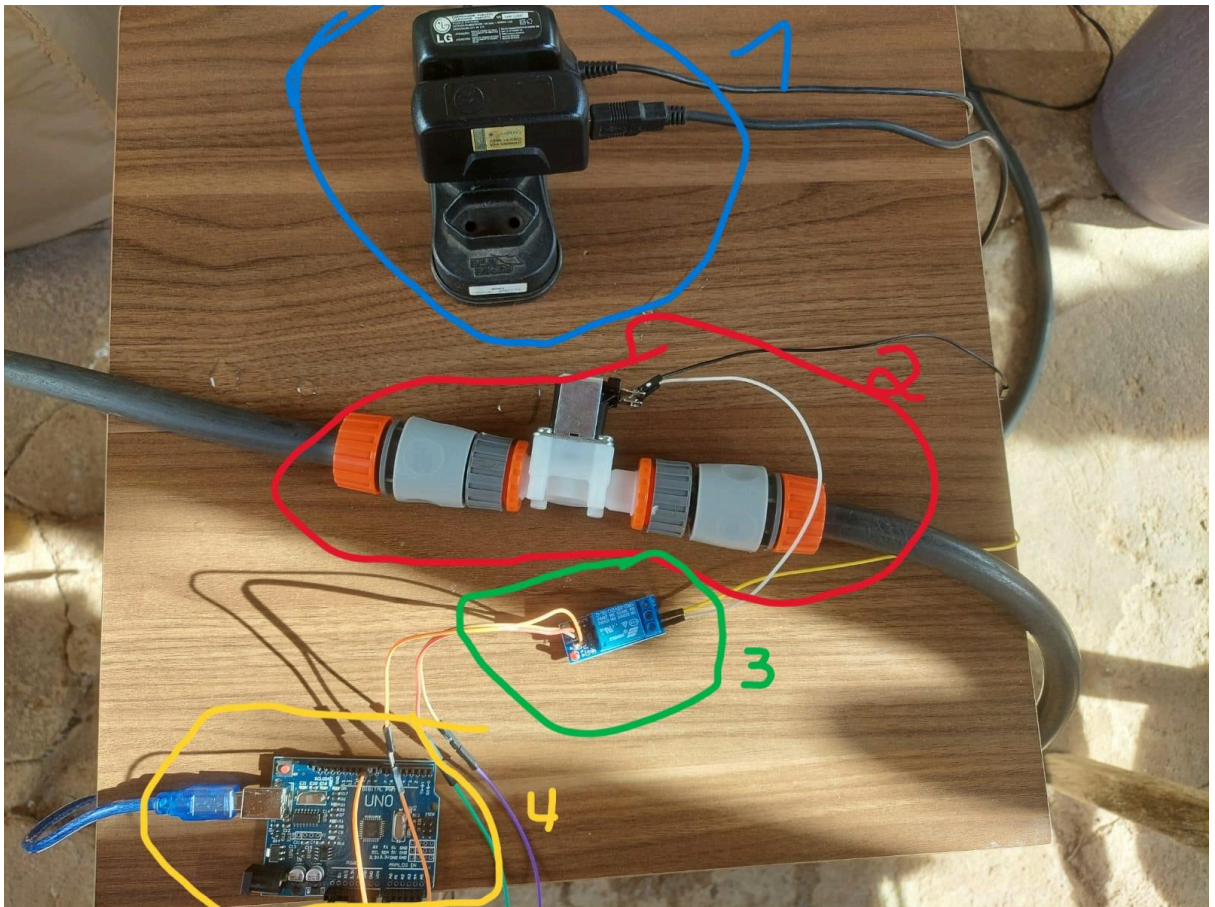


Fonte: autoria própria.

Quando a válvula é aberta, ocorre a passagem de água, que é depositada na amostra.

A Figura 25 mostra a montagem real do circuito, com as indicações numéricas apontando para a montagem da fonte (1), válvula (2), relé de acionamento (3) e Arduino (4).

Figura 25: Montagem real.



Fonte: Autoria própria.

Na bibliografia utilizada como referência, são encontradas amostras coletadas em recipientes de 16 cm (GARCIA, 2020). No entanto, no experimento implementado neste trabalho, optou-se por utilizar recipientes com uma abertura maior, de 25 cm, o que permite uma maior área de absorção da água depositada, diminuindo poças e encharcamentos, melhorando o rendimento e se aproximando de um sistema mais aberto.

Todos os sistemas implementados experimentalmente foram mantidos por 8 horas, que foi tempo suficiente para que os sistemas apresentassem completamente a resposta transitória e chegassem ao regime permanente.

4 RESULTADOS

4.1 COMPARAÇÃO ENTRE FUZZY E IRRIGAÇÃO MANUAL

É possível concluir que os métodos de irrigação automatizados, como o fuzzy, apresentam vantagens de forma geral em relação aos métodos manuais por diferentes aspectos. Dentre eles se encontram a exigência de esforço físico em algumas condições adversas que são exigidas pela irrigação manual, que pode ser contornada com a automatização da irrigação, o controle de recursos consumidos e o desempenho de cada caso (DURSON & OZDEN, 2011).

O trabalho rural manual envolvido na agricultura em geral, inclusive o que envolve a irrigação, exige muito fisicamente das pessoas que trabalham com esta área, podendo levar a doenças e problemas articulares, musculares e ortopédicos (ROSSONI, 2023). Desta forma, o primeiro ganho notável da implementação de um sistema automatizado de irrigação é a diminuição do trabalho com intenso esforço físico em condições que podem levar à degradação da saúde humana.

Outro aspecto importante é o nível de aproveitamento. A resposta humana tende a não ser rápida o bastante em casos onde geralmente ocorre a irrigação manual, pois não lida com valores numericamente lidos por sensores e precisa de alterações maiores para que sejam perceptíveis aos sentidos humanos, fazendo com que a curva de ajuste não seja a mais eficiente (NISE, 2011). Com isto, equipamentos capazes de fazer uma leitura mais precisa e adequada do ambiente tem a capacidade de responder mais rápido a possíveis oscilações das variáveis que influenciam o sistema, se adequando a elas mais rapidamente, o que fornece uma resposta mais adequada ao sistema controlado e traz uma melhoria de desempenho.

Mais um aspecto importante é sobre o rendimento entre um sistema automático e um sistema gerido por um especialista. Atualmente os sistemas de automatização já se mostram bastante robustos para suportar diferentes tipos de ambiente e, particularmente, o fuzzy apresenta resultados superiores a um sistema gerido por um humano. Segundo Garcia (2020), em experimento realizado com controlador fuzzy no cultivo de alface crespa, mesmo um protótipo feito com materiais de baixo custo foi capaz de se manter em funcionamento sem problemas por mais de 2 meses, gerando ao término do experimento alfaces que tiveram o mesmo desempenho produtivo que o sistema gerido manualmente, isto é, ambas as

culturas apresentaram mesmos pesos, mesmos tamanhos de folhas e mesmas quantidades de folhas, porém com o sistema fuzzy tendo consumido 11% de água a menos.

É notável então que é mais interessante que o profissional especialista se dedique ao monitoramento e gerenciamento de um sistema automatizado do que à operação em si, haja vista que esta é mais desgastante, mais suscetível à erros, e menos otimizada que o sistema automático, particularmente no caso do fuzzy.

4.2 COMPARAÇÃO DOS RESULTADOS EXPERIMENTAIS

Após a implementação prática, foram coletados os dados obtidos após 8 horas de experimento. A Tabela 3 mostra os valores de umidade medidos no decorrer do tempo em cada sistema.

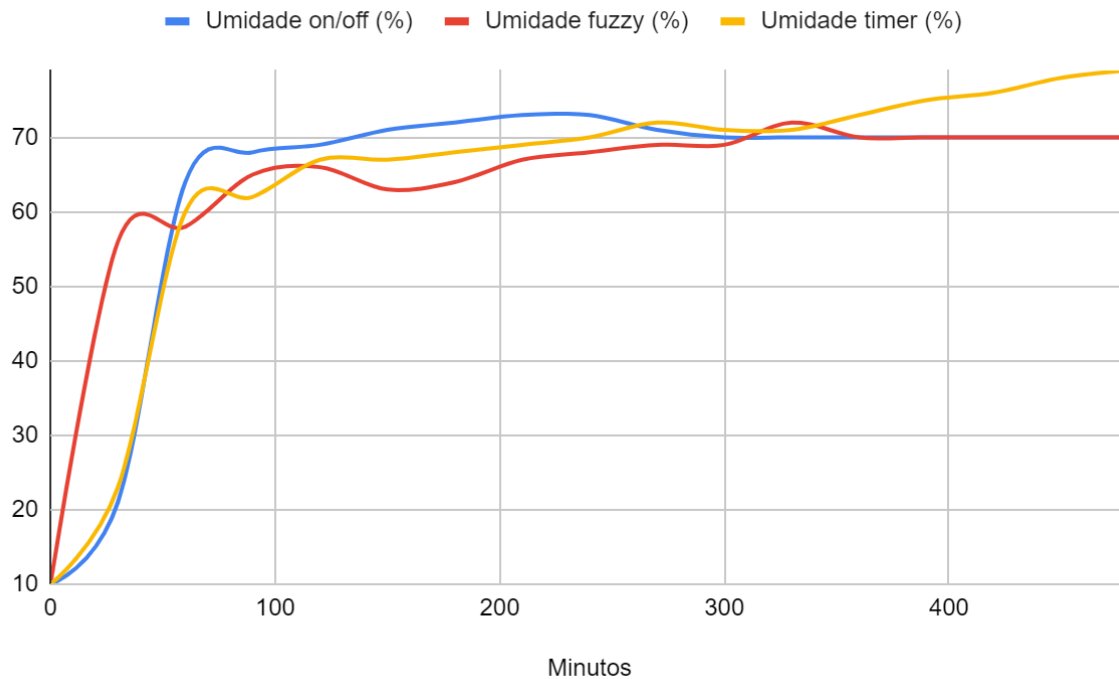
Tabela 3: Relação de umidade em cada sistema em relação aos ciclos de 30 minutos.

Minutos	Umidade on/off (%)	Umidade fuzzy (%)	Umidade timer (%)
0	10	10	10
30	21	56	23
60	64	58	60
90	68	65	62
120	69	66	67
150	71	63	67
180	72	64	68
210	73	67	69
240	73	68	70
270	71	69	72
300	70	69	71
330	70	72	71
360	70	70	73
390	70	70	75
420	70	70	76
450	70	70	78
480	70	70	79

Fonte: Autoria própria.

Graficamente, é possível ver a evolução do nível de umidade do solo no tempo na Figura 26.

Figura 26: Nível de umidade do solo de cada sistema no tempo.



Fonte: Autoria própria.

É notável, pelos dados obtidos, que alguns dos sistemas apresentam aumento e decréscimo em sequência do nível de umidade. Uma hipótese para tentar explicar este efeito diz respeito ao nível de absorção das camadas mais fundas de terra, que ficam mais compactadas pelo peso das camadas superiores e têm maior dificuldade de absorver a água depositada. Assim, é esperado que nos primeiros ciclos, a água permaneça em uma camada mais superior da amostra de solo, o que a torna mais volátil e suscetível à evaporação. O sensor, presente na camada mais acima da amostra, detecta essas variações. Depois que a água é absorvida pelas camadas mais baixas, o nível de umidade tende a permanecer mais estável, oscilando menos.

Comparando os sistemas, nota-se que as respostas dos sistemas com timer sem realimentação e on/off com realimentação são bem parecidas no início, pois é depositado o mesmo volume de água em ambos. No entanto, a partir do quarto ciclo de acionamento, isto é, o quarto ciclo de 30 minutos onde pode ocorrer o

acionamento da irrigação, é possível notar que surgem algumas diferenças maiores, chegando até a 6% de umidade de diferença. Apesar de ambos se manterem crescentes, a velocidade de crescimento entre eles é diferente. Sendo ambos os experimentos mantidos nas mesmas condições, com mesma vazão, amostras de solo retiradas do mesmo lugar ao mesmo tempo e com os sensores caracterizados, algumas hipóteses que podem explicar essa diferença é o nível de compactação do solo em cada amostra, o que pode interferir na capacidade de absorção de água, ou até mesmo algum fator externo, como uma leve diferença de temperatura de uma amostra para a outra, o que pode interferir na quantidade de água evaporada. Entretanto, o principal aspecto esperado que diferencia ambos os sistemas foi compatível com o que ocorreu na prática, que é a estabilização. O sistema controlado com realimentação chegou a apresentar um sobressinal de 4,3%, mas se estabilizou no valor de 70%, sem erro de regime, o que foi um resultado melhor que o esperado, já que era previsto níveis de oscilação de 1% a 2% a cada ciclo em torno do *setpoint* devido à própria natureza do controlador, que só reage quando há essa variação e reage de uma forma única, sempre depositando o mesmo volume. Já o timer sem realimentação seguiu conforme o esperado, sem conseguir se estabilizar, mostrando que neste tipo de sistema de acionamento é necessário o acompanhamento de um especialista, pois há a possibilidade de ocorrerem problemas deste tipo.

É importante levar em conta que, num ambiente de aplicação real do tipo de acionamento com timer sem realimentação, provavelmente haveria um especialista humano capaz de identificar se há a necessidade ou não de acionamento, desligando o timer se necessário, porém seria uma ação baseada na experiência deste especialista, e não necessariamente baseado no valor de umidade lido, já que este tipo de acionamento não costuma implementar sensores de umidade. Assim, se a umidade estivesse de 5% a 10% acima ou abaixo do valor de referência estabelecido, pode ser que, sem a leitura correta de um sensor, o especialista humano não agisse e fosse mantido um valor de umidade inadequado no solo. A implementação do sensor neste trabalho foi unicamente para monitoramento, o que tende a não ocorrer em aplicações reais. Assim, é claro que em situações extremas provavelmente um especialista atuaria, mas ainda assim deve ser levado em conta o caso onde não há atuação, que foi o implementado neste trabalho, que mostra os possíveis problemas que podem ocorrer neste tipo de acionamento sem controle

realimentado, principalmente a instabilidade e encharcamento, que foi um fenômeno também observado neste caso, o que já o torna, dentre os modos de acionamento implementados, o menos confiável e com pior desempenho, já que não se estabiliza no valor que deveria, independentemente do tempo gasto ou outros critérios de desempenho que poderiam ser avaliados.

Já na comparação com o sistema fuzzy, verifica-se que este tem uma subida mais rápida no início, o que é esperado já que o volume de água depositado é variado de forma inversamente proporcional ao nível de umidade. Portanto, com o solo bem seco, ocorre a deposição de uma quantidade bem grande de água. No entanto, depois do segundo ciclo, o nível de umidade já é alto o bastante para que o volume de água depositado seja menor do que 1,2 l, que é o depositado pelos demais sistemas, o que faz com que a resposta fique mais lenta a partir de então, e ela tende a ficar ainda mais lenta conforme o nível de umidade aumenta. Portanto, o sistema fuzzy implementado apresentou uma resposta mais rápida nos primeiros ciclos, porém o tempo de assentamento pelo critério de 2% foi mais lento que o on/off realimentado, tendo este se estabilizado após 270 minutos, isto é, 4,5 horas, enquanto o fuzzy só se estabilizou depois de 6 horas. Entretanto, o fuzzy teve um sobressinal menor e mais rápido. Enquanto o on/off chegou a 4,3% de sobressinal e ficou 2,5 horas neste estado, o fuzzy atingiu 2,8% de sobressinal e ficou apenas 30 minutos. Assim, vale a pena considerar a cultura plantada, se ela precisa de um nível de umidade alto mais rapidamente ou de um nível de umidade mais bem controlado, ainda que com resposta mais lenta.

Por fim, um aspecto importante a ser ressaltado é o consumo de água. O acionamento com timer sem realimentação foi o que mais consumiu água, tendo utilizado 19,2 l ao todo, pois ele nunca cessa de depositar água, independentemente do nível de umidade do solo. Já o controlador simples realimentado consumiu 13,2 l durante todo o período do experimento. Por fim, o fuzzy foi o que consumiu a menor quantidade de água, sendo 9,96 l durante todo o experimento, sendo 24,5% a menos de água do que o sistema on/off realimentado. A Tabela 4 mostra o consumo de água de cada sistema controlado com a passagem do tempo, mostrando o volume depositado a cada acionamento (ação) e o volume total acumulado até aquele instante (acumulado).

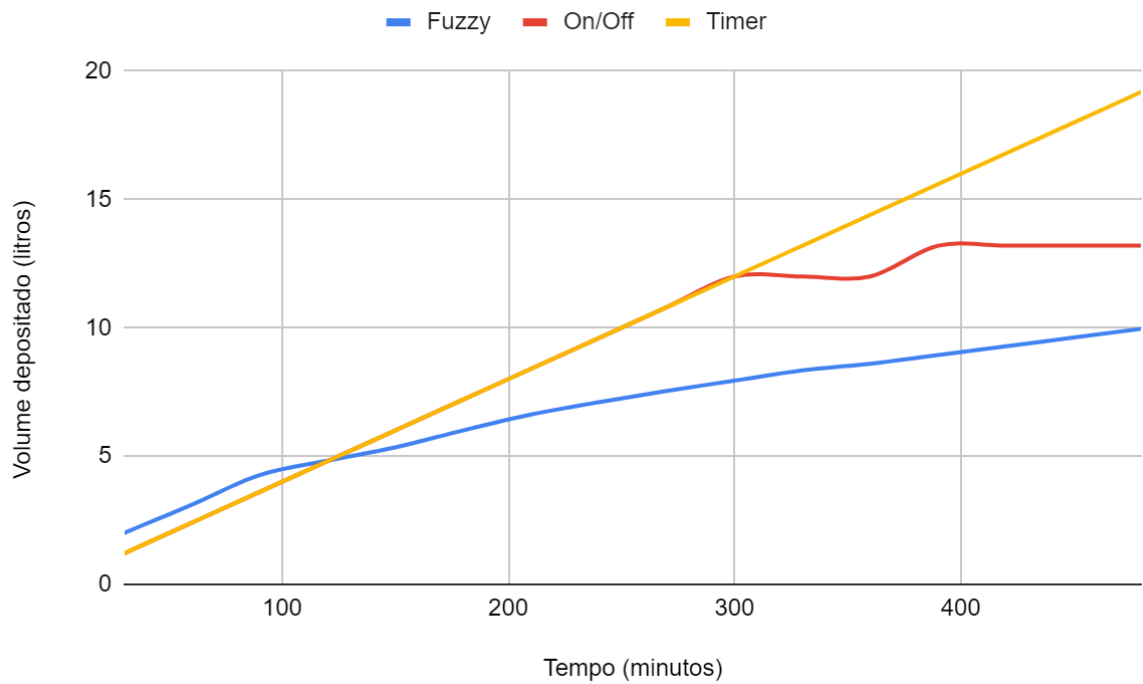
Tabela 4: consumo de água por cada controlador

Tempo	Fuzzy		On/off		Timer	
	Ação	Acumulado	Ação	Acumulado	Ação	Acumulado
30	2	2	1,2	1,2	1,2	1,2
60	1,1	3,1	1,2	2,4	1,2	2,4
90	1,16	4,26	1,2	3,6	1,2	3,6
120	0,56	4,82	1,2	4,8	1,2	4,8
150	0,52	5,34	1,2	6	1,2	6
180	0,66	6	1,2	7,2	1,2	7,2
210	0,62	6,62	1,2	8,4	1,2	8,4
240	0,48	7,1	1,2	9,6	1,2	9,6
270	0,44	7,54	1,2	10,8	1,2	10,8
300	0,4	7,94	1,2	12	1,2	12
330	0,4	8,34	0	12	1,2	13,2
360	0,26	8,6	0	12	1,2	14,4
390	0,34	8,94	1,2	13,2	1,2	15,6
420	0,34	9,28	0	13,2	1,2	16,8
450	0,34	9,62	0	13,2	1,2	18
480	0,34	9,96	0	13,2	1,2	19,2

Fonte: Autoria própria.

Graficamente, é possível observar os dados dispostos na Tabela 3 na Figura 27, que apresenta o acumulado de água em cada sistema utilizado com o passar do tempo.

Figura 27: Acumulado do consumo de água por cada controlador.



Fonte: Autoria Própria.

Assim, é notável que dentre os três sistemas implementados, o sistema sem realimentação apresentou o pior desempenho por não estabilizar, encharcar em certos momentos, e consumir mais água que os demais. Já entre o controlador on/off realimentado e o fuzzy, este apresentou sobressinal menor e mais rápido, e consumiu menos água, enquanto aquele teve uma resposta mais rápida.

5 CONSIDERAÇÕES FINAIS

5.1 CONCLUSÕES

Após o término do trabalho e análise dos resultados obtidos, é notável que a automatização de um sistema de irrigação apresenta diversas vantagens em relação à irrigação manual, principalmente em relação à diminuição do esforço físico humano necessário e no desempenho observado, tendo uma diminuição no consumo de recursos utilizados no caso da irrigação automatizada para obter os mesmos resultados.

Já dentre as formas de irrigação automatizadas, destacou-se o fuzzy em relação aos métodos tradicionais, como o controlador liga-desliga realimentado e o acionamento com timer, tendo menor nível de sobressinal e maior economia de água, embora a resposta tenha sido mais lenta.

Por fim, na comparação entre fuzzy e PID, nota-se que são utilizados em casos diferentes e, embora de alguma forma seja possível comparar os dois sistemas e até verificar uma certa vantagem da utilização do fuzzy em relação ao PID por motivos de facilidade de desenvolvimento e até mesmo desempenho, é mais interessante tratar essas duas formas como complementares, de modo que é possível até mesmo implementar um sistema baseado em fuzzy-PID com melhores desempenhos do que o fuzzy ou o PID isolados.

Portanto, de forma geral, conclui-se que dentre todas as formas estudadas, o controle com lógica fuzzy apresenta melhor resultado levando-se em conta os critérios adotados, tendo menores níveis de sobressinal e maior economia de recursos hídricos em relação a outras formas comparadas.

5.2 TRABALHOS FUTUROS

Existem pontos que podem ser melhorados em trabalhos futuros. O primeiro deles é em relação à projeção de custos. Neste trabalho, e em outros utilizados como fonte de pesquisa, foram utilizados principalmente protótipos, o que podem gerar uma noção inicial de preços e valores, porém é relativamente complicado projetar os valores de um protótipo em fases de testes tão iniciais para uma aplicação em escala. Por isso, é importante a realização de uma pesquisa para a

implementação comercial de grande escala desses controladores para, aliado aos dados obtidos neste trabalho, ter uma análise mais completa da relação entre custo e benefício de cada tipo de controlador ou sistema de acionamento.

Outro aspecto que pode ser melhorado é sobre o uso de variáveis do ambiente. Conforme constatado pela literatura, mostrou-se mais interessante analisar nesta oportunidade apenas uma variável do ambiente, que foi a umidade do solo. Entretanto, em casos futuros, seria interessante levar mais variáveis em consideração, tendo mais tempo e disponibilidade para determinar quantas e quais variáveis seriam essas e, principalmente, como determinar a melhor resposta para cada sistema de controle com essas variáveis de modo que o algoritmo utilizado seja o melhor em cada caso, que foi justamente o motivo da não utilização de mais variáveis neste trabalho.

Outra possibilidade de melhoria é comparar com sistemas de acionamento e controle que não foram trazidos para este trabalho. Controles com redes neurais, por exemplo, não foram implementados devido à sua pouca utilização na prática, mas seria interessante realizar essas comparações em aplicações futuras.

Mais uma melhoria que poderia ser realizada é a utilização de um controlador fuzzy utilizando como entradas também as derivadas do sinal lido, sua diferenciação no tempo, e comparar os resultados com o sistema fuzzy com apenas a entrada do sinal lido.

Por fim, uma melhoria que poderia trazer resultados ainda mais expressivos seria a aplicação em uma grande extensão territorial, se possível com culturas diferentes, ao invés de apenas amostras de solo. Isso levaria mais tempo, pelo menos de 2 a 3 meses, que é o tempo médio de algumas culturas mais simples, e seria mais custoso, pois precisaria de espaço e disponibilidade maiores, mas poderia trazer análises ainda mais precisas e interessantes sobre melhorias na capacidade produtiva.

REFERÊNCIAS

ARDUINO. 2022. Disponível em: <
<https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf> >. Acesso em:
14 out. 2023.

ATMEL. 2015. Disponível em: <
https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf >. Acesso em: 14 out. 2023.

AZMI, Nur A. B. IMPLEMENTATION OF FUZZY LOGIC CONTROLLER ON
SPRINKLER SYSTEM FOR VEGETABLE GARDEN. 2022.

BALBINOT, A; BARRETO, M. PROCESSO DE DEFUZZIFICAÇÃO NA
IMPLEMENTAÇÃO COMPUTACIONAL DE RACIOCÍNIO MÉDICO. 2009.

Disponível em: <

<https://lume.ufrgs.br/bitstream/handle/10183/30412/000730999.pdf?sequence=1> >.

Acesso em 04 de agosto de 2024.

BRAGA, N. O básico sobre microcontroladores. Newton Braga, Mar. Disponível em:
<
<https://newtonbraga.com.br/index.php/electronica/52-artigos-diversos/13264-o-basico-o-sobre-os-microcontroladores-parte-2-mic140> >. Acesso em: 14 out. 2023.

BONISSONI, Kassiana. Irrigação por microaspersão pode gerar economia de até
50% em energia elétrica. 26/01/2022. Disponível em: <

<https://revistacultivar.com.br/noticias/irrigacao-por-microaspersao-pode-gerar-economia-de-ate-50percent-em-energia-eletrica> >. Acesso em 04/06/2024.

BORDIN, Deyver. PLATAFORMA COMPUTACIONAL FUZZY PARA AVALIAÇÃO
NOS ESTÁGIOS DO TOMATEIRO DOS EFEITOS DA IRRIGAÇÃO E SALINIDADE
DA ÁGUA. 2016. Disponível em: <

<https://repositorio.unesp.br/server/api/core/bitstreams/f81f1d28-7ed2-4f02-9aac-f5fb771730e7/content> >. Acesso em 24 de agosto de 2024.

CARVALHO, P. Sistemas de irrigação. p. 67, 2018.

CASA DA ROBÓTICA. Placa Uno SMD R3 Atmega328 Sem Cabo Compatível com Arduino. Disponível em: <

<https://www.casadarobotica.com/placas-embarcadas/arduino/placas/placa-uno-r3-smd-atmega328-sem-cabo> >. Acesso em 04/06/2024.

CASTAN, D. O que significa excesso e déficit hídrico na agricultura? 2021. Disponível em:< <https://agrosmart.com.br/blog/excesso-deficit-hidrico/> >. Acesso em: 14 out. 2023.

COELHO, E. F., TEIXEIRA, A. H. C., COSTA, E. L., LIMA, M. B., & BORGES, A. L. 2007. Irrigação por aspersão.

CRUZ, J. C et al. O PRODUTOR PERGUNTA, A EMBRAPA RESPONDE. 2011.

EMBRAPA. Disponível em: <

<https://ainfo.cnptia.embrapa.br/digital/bitstream/item/73399/1/Manejo-solo-1.pdf> >.

Acesso em 15 de agosto de 2024.

DURSON, M., & OZDEN, S. 2011. A wireless application of drip irrigation automation supported by soil moisture sensors. Technical Education Faculty, Gazi University. Turkey. Ankara

ECODEBATE. 2019. Disponível em: <

<https://www.ecodebate.com.br/2019/04/03/agricultura-irrigada-consome-mais-da-metade-da-agua-usada-no-pais/#:~:text=A%20agricultura%20irrigada%20gasta%20mais,%C3%A1gua%20retirada%20do%20meio%20ambiente.> > Acesso em: 14 out. 2023.

ELETROGATE. Módulo sensor de umidade do solo. Disponível em: <

<https://www.eletrogate.com/modulo-sensor-de-umidade-de-solo> >. Acesso em

04/06/2024.

FELDENS, L. O homem, a agricultura e a história. [s.n.], 2018. Disponível em: <

https://www.univates.br/editora-univates/media/publicacoes/246/pdf_246.pdf >.

Acesso em: 14 out. 2023.

FRANKLIN, G. F; POWELL, J. D; EMAMI-NAEINE, A. Sistemas de Controle Para Engenharia. 6ª Ed. 2013.

FRIZZONE, A. J. Os métodos de irrigação: Notas de aula da disciplina leb 1571 - irrigação. 2017.

GARCIA, Alisson Felipe C. SISTEMA DE IRRIGAÇÃO PARA CULTURAS DE ALFACE CRESPA BASEADO EM LÓGICA FUZZY. 2020. Disponível em: < <https://tede2.uepg.br/jspui/handle/prefix/3318> >.

GODINHO, Emmanuel Z.; GASPAROTTO, Hélio V.; CANEPPELE, Fernando L. LÓGICA FUZZY NA AGRICULTURA. 2022. Disponível em: < https://www.researchgate.net/publication/361596021_Logica_Fuzzy_na_Agricultura >. Acesso em 24 de agosto de 2024.

GOMIDE, F. A. Modelagem, controle, sistemas e lógica fuzzy. p. 19, 1994. Disponível em: < <https://www.sba.org.br/revista/volumes/v4n3/v4n3a02.pdf> >. Acesso em 18 de setembro de 2024.

GOODCHIELD, Martion. A Method for Precision Closed-loop Irrigation Using a Modified PID Control Algorithm. 2015. Disponível em: < https://www.researchgate.net/publication/314890442_A_Method_for_Precision_Closed-loop_Irrigation_Using_a_Modified_PID_Control_Algorithm >. Acesso em 24 de agosto de 2024.

LEITÃO, A. A água, um recurso cada vez mais escasso. Ecodebate, Mar. 2018. Disponível em: < <https://www.ecodebate.com.br/2018/03/22/a-agua-um-recurso-cada-vez-mais-escasso-artigo-de-alexandra-leitao/#:~:text=A%20escassez%20de%20%C3%A1gua%20%C3%A9,do%20regime%20alimentar%20das%20pessoas%20> >. Acesso em: 14 out. 2023.

LIMA, F. Desenvolvimento de um sistema de controle fuzzy do potencial matricial da Água do solo, visando à otimização de processos de irrigação: Tese (doutorado em engenharia mecânica). 2007.

MANTOVANI, Everardo Chartuni; BERNARDO, Salassier; PALARETTI, Luiz Fabiano. Irrigação: Princípios e Métodos. 3ª Edição. Brasil: Editora UFV, 2009. 355 páginas.

LIMA-JUNIOR, Francisco Rodrigues; CARPINETTI, Luiz Cesar R. A multicriteria approach based on fuzzy QFD for choosing criteria for supplier selection. 2016. Disponível em: < <https://www.sciencedirect.com/science/article/abs/pii/S0360835216303540> >. Acesso em 03 de agosto de 2024.

MATLAB, Version 9.8.0.1323502 (R2020a). [S.l.]: The MathWorks Inc., 2020.

MATHIAS, Lucas. PESQUISA QUANTITATIVA E QUALITATIVA: QUAL É A MELHOR OPÇÃO? 14 de junho de 2022. Disponível em: < <https://mindminers.com/blog/pesquisa-qualitativa-quantitativa/> >. Acesso em 14 de agosto de 2024.

MAXMAQ. Irrigação por Aspersão. 01/10/2019. Disponível em: < <https://maxmaq.com.br/blog/irrigacao-por-aspersao/> >. Acesso em 04/06/2024.

MEDEIROS, Pedro H. S. SISTEMA DE IRRIGAÇÃO AUTOMATIZADO PARA PLANTAS CASEIRAS. 2018. Disponível em: < https://www.monografias.ufop.br/bitstream/35400000/1199/1/MONOGRAFIA_SistemaIrrigacaoAutomatizado.pdf >. Acesso em 07/05/2024.

MENEZES, Paulo L. DESENVOLVIMENTO DE MODELO EM REDES NEURAIS PARA SIMULAÇÃO DA PRECIPITAÇÃO NA IRRIGAÇÃO POR ASPERSÃO. 2014.

MERCADO LIVRE. Valvula Solenoide 12v 1/2 Água - Automação Irrigação. Disponível em: < https://produto.mercadolivre.com.br/MLB-3220989681-valvula-solenoide-12v-12-agua-automaco-irrigaco-_JM#position=1&search_layout=stack&type=item&tracking_id=312b347a-1e09-499e-8dc5-7922d55ceb3e >. Acesso em 04/06/2024.

MICROCHIP TECHNOLOGY. ATmega328. Disponível em: < <https://www.microchip.com/en-us/product/atmega328> >. Acesso em 04/06/2024.

NAGAHAGE, E. A. A. D; NAGAHAGE, I. S. P; FUJINO, T. Calibration and Validation of a low-cost capacitive moisture sensor to integrate the automated soil moisture monitoring system. Agriculture, MDPI AG, v. 9, n. 7, p 141.. 2019.

NISE, N.S. 2011. Control Systems Engineering. 6th edition.

OGATA, K. ENGENHARIA DE CONTROLE MODERNO. 5º Ed. São Paulo. Pearson. 2011.

OLIVEIRA, Francisco F. de; IMMICH, Roger. Desenvolvimento de um controlador de lógica fuzzy para irrigação por gotejamento. 2023. Disponível em: < <https://sol.sbc.org.br/index.php/sbiagro/article/view/26545/26368> >.

OPENCADD. MATLAB: o que é e para que serve. 05/04/2022. Disponível em: < <https://www.opencadd.com.br/blog/o-que-e-matlab#:~:text=Ele%20%C3%A9%20um%20linguagem%20de.de%20sinais%2C%20finan%C3%A7as%20e%20outros.> >. Acesso em 04/06/2024.

PEREZ-ORTOLA, M; KNOX, J. W. Water relations and Irrigation requirements of onion (*allium cepa* 1): a review of yield and quality impacts. *Experimental Agriculture*. Cambridge University Press, v. 51, n. 2, p. 210-231, nov 2014. Disponível em: < <https://core.ac.uk/download/pdf/266980187.pdf> >. Acesso em 18 de setembro de 2014.

REHAGRO. 2017. Disponível em: < <http://rehagro.com.br/automacao-em-sistemas-de-irrigacao/> >. Acesso em: 14 out. 2023.

RESENDE, Jhonathan C. CONTROLE PID PARA SISTEMAS DE IRRIGAÇÃO. 2014.

ROISENBERG, M. Sistemas Nebulosos. Departamento de Informática e Estatística - UFSC, 2018.

ROSSONI, Guilherme. AGRO: trabalhador rural e os problemas na lombar e coluna. 2023. Disponível em: < <https://drguilhermerossoni.com.br/agro-trabalhador-rural-e-os-problemas-na-lombar-e-coluna/#:~:text=O%20trabalhador%20rural%20possui%20uma.s%C3%A3o%20cada%20vez%20mais%20frequentes.> >. Acesso em 22 de agosto de 2024.

ROVEDA, S. R. M. M; ROVEDA, J. A .F; LOURENÇO, R. W. APLICAÇÃO DA LÓGICA FUZZY PARA ESTUDO DE PERMEABILIDADE DE SOLOS DE REGIÃO IMPACTADA DA BAIXADA SANTISTA. 2011. Disponível em: < <https://www.cea-unesp.org.br/holos/article/download/5634/4351/28824#:~:text=Entre>

[%20os%20m%C3%A9todos%20de%20defuzzifica%C3%A7%C3%A3o.e%20o%20c
entro%20de%20gravidade](#). >. Acesso em 08 de agosto de 2024.

SABOUNCHI, Nasim M. et al. Fuzzy Modeling of Linguistic Variables in a System Dynamics Context. 2011. Disponível em: <
https://www.researchgate.net/publication/272943976_Fuzzy_Modeling_of_Linguistic_Variables_in_a_System_Dynamics_Context >. Acesso em 24 de agosto de 2024.

SANTOS, R. dos. Desenvolvimento de um sistema de irrigação automático baseado em lógica fuzzy. 2019. Disponível em: <
https://www.academia.edu/40263431/Desenvolvimento_de_um_sistema_de_irrigacao_autom%C3%A1tico_baseado_em_logica_Fuzzy >. Acesso em 18 de setembro de 2024.

SCHERZ, S. M. P. Practical Electronics for Inventors. 3. ed. [S.l.]: The McGraw-Hill Companies, 2013.

SERVULO, M. O; LEITE, M. S; DURÇO, M. A. SISTEMA AUTOMÁTICO DE IRRIGAÇÃO COM CONTROLE PID. 2019. Disponível em: <
<http://www1.sp.senac.br/hotsites/blogs/revistainiciacao/wp-content/uploads/2019/09/02-1.pdf> >. Acesso em 20 de agosto de 2024.

SIJIA, L. et al. Research and Design of Intelligent Irrigation Control System Based on Fuzzy. 2021. Disponível em: <
<https://ieeexplore.ieee.org/document/9544215/authors#authors> >. Acesso em 20 de agosto de 2024.

SILVA, A. E; MASSELLI, Y. M. C. Análise Comparativa entre Controladores PID e Fuzzy Utilizando LabVIEW e Arduino Aplicados em um Sistema de Controle de Nível. 2014. Disponível em: <
<https://www2.inatel.br/biblioteca/todo-docman/pos-seminarios/seminario-de-automacao-industrial-e-sistemas-eletronicos/iv-saisee/9479-analise-comparativa-entre-controladores-pid-e-fuzzy-utilizando-labview/file> >. Acesso em 20 de agosto de 2024.

SOUSA, R. S. 2016. Tudo sobre irrigação localizada. Disponível em: < <https://www.irrigacao.net/irrigacao-localizada/tudo-sobre-irrigacao-localizada/> > Acessado em: 21 de Junho de 2018.

SOUZA, C. R. V. (controller area network) uma abordagem para automação e controle na área agrícola. 94, 2002.

STONE, L. F. 2011a. Irrigação por aspersão. Disponível em: < http://www.agencia.cnptia.embrapa.br/gestor/feijao/arvore/CONTAG01_37_1311200215102.html > Acessado em: 21 de Junho de 2018.

TANSCHKEIT, R. SISTEMAS FUZZY. 2004. Disponível em: < <https://www.inf.ufsc.br/~mauro.roisenberg/ine5377/Cursos-ICA/LN-Sistemas%20Fuzzy.pdf> >. Acesso em 12 de maio de 2024.

TAUGER, M. B. Agriculture in World History. 1. ed. [S.I.]: Routledge, 2020.

VIEIRA, Felipe B. ESTUDO COMPARATIVO ENTRE CONTROLADORES FUZZY E PI PARA UM SISTEMA DE TANQUE. 2017.

VIEIRA, V. PID, MPC ou Fuzzy, qual a melhor estratégia de controle? 2020. Disponível em: < [PID, MPC ou Fuzzy, qual melhor estratégia de controle? | by Victor Vieira | I.Systems—#beyondtogether | Medium](#) >. Acesso em 12 de maio de 2024.

UNIBAVE. Curso de Irrigação por Gotejamento será realizado neste fim de semana no Unibave. 09/08/2018. Disponível em: < <https://unibave.net/noticia/curso-de-irrigacao-por-gotejamento-sera-realizado-neste-fim-de-semana-no-unibave/> >. Acesso em 04/06/2024.

ELETROGATE. Módulo Relé 5V 10A 1 Canal. Disponível em: < [https://www.eletrogate.com/modulo-rele-1-canal-5v?utm_source=Site&utm_medium=GoogleMerchant&utm_campaign=GoogleMerchant&utm_source=google&utm_medium=cpc&utm_campaign=\[MC4\]_\[G\]_\[PMax\]_ArduinoRoboticaSensoresModuloss&utm_content=&utm_term=&gad_source=1&gclid=Cj0KCQjw9Km3BhDjARIsAGUb4nx](https://www.eletrogate.com/modulo-rele-1-canal-5v?utm_source=Site&utm_medium=GoogleMerchant&utm_campaign=GoogleMerchant&utm_source=google&utm_medium=cpc&utm_campaign=[MC4]_[G]_[PMax]_ArduinoRoboticaSensoresModuloss&utm_content=&utm_term=&gad_source=1&gclid=Cj0KCQjw9Km3BhDjARIsAGUb4nx) >

[pSisPEa-eYimZwRkMSmTMOzbs2Oso5tVBW_3NTPuFZrmeP3nknwwaAtaFEALw_wcB](#) >. Acesso em 04/06/2024.

WOLFRANN, J; MENEZES, P. L; VINCENZI, S. L; SCHUTZ, F. C. A. SIMULAÇÃO DE IRRIGAÇÃO POR ASPERSÃO UTILIZANDO REDES NEURAIAS ARTIFICIAIS TREINADAS COM E SEM TERMO MOMENTUM. 2018.

YARABRASIL. Irrigação por gotejamento: como funciona e vantagens. 2022. Disponível em: < [Irrigação por gotejamento: como funciona e vantagens | Yara Brasil](#) >. Acesso em 18 de setembro de 2024.

YOHANNES, F., & TADESSE, T. 1998. Effect of drip and furrow irrigation and plant spacing on yield of tomato at Dire Dawa, Ethiopia. Alemaya University of Agricultur, 35(3).

Zimmermann, H.-J. (1991) Fuzzy Set Theory and Its Applications. Kluwer Academic Publishers, Boston. Disponível em: < <https://doi.org/10.1007/978-94-015-7949-0> >. Acesso em 03 de agosto de 2024.

ZUFFO, Antonio Carlos. ANÁLISE MULTICRITERIAL AO PLANEJAMENTO DE RECURSOS HÍDRICOS: UMA METODOLOGIA FUZZY PARA O ENFOQUE AMBIENTAL. 2010. Disponível em: < https://www.researchgate.net/figure/Figura-44-Formas-de-funcoes-de-pertinencia-A-para-numeros-fuzzy-normais_fig2_260244758 >. Acesso em 08 de agosto de 2024.

APÊNDICE A - Código do controlador liga/desliga com realimentação

```
const int pinoSensor = A0;
const int pinoValvula = 9;
const int limiarSeco = 70;
const int tempoRega = 15;
int umidadeSolo = 0;

void setup() {
  pinMode(pinoValvula, OUTPUT);
  digitalWrite(pinoValvula, HIGH);

  Serial.begin(9600);
}

void loop() {
  /* O loop for abaixo serve para a passagem de tempo. Ele determina um
  atraso de 1800 segundos, isto é, 30 minutos, que é o tempo estipulado
  para cada ciclo */
  for(int i=0; i < 1800; i++) {
    if(i == 1500){
      // Leitura da tensão do sensor
      umidadeSolo = analogRead(pinoSensor);
      // Conversão do valor de tensão em valor de umidade
      umidadeSolo = map(umidadeSolo, 1023, 0, 0, 100);

      // Mostra o valor através da conexão serial
      Serial.println(umidadeSolo);
      Serial.println(" %    ");
    }
    delay(1000);
  }

  // Leitura da tensão do sensor
  umidadeSolo = analogRead(pinoSensor);
  // Conversão do valor de tensão em valor de umidade
```

```
umidadeSolo = map(umidadeSolo, 1023, 0, 0, 100);

if(umidadeSolo < limiarSeco) {
  digitalWrite(pinoValvula, 0);
  // Espera o tempo estipulado
  delay(tempoRega*1000);
  digitalWrite(pinoValvula, 1);
}
else {
  // Espera o tempo estipulado
  delay(3000);
}
}
```

APÊNDICE B - Código do acionador timer sem realimentação

```
const int pinoSensor = A0;
const int pinoValvula = 9;
const int tempoRega = 15;
int umidadeSolo = 0;

void setup() {
  pinMode(pinoValvula, OUTPUT);
  digitalWrite(pinoValvula, HIGH);

  Serial.begin(9600);
}

void loop() {
  /* O loop for abaixo serve para a passagem de tempo de 1800 segundos,
isto é, 30 minutos */
  for(int i=0; i < 1800; i++)
    {if(i == 1500){
      /* Neste código o valor da umidade do solo não determina o
acionamento ou não da válvula, apenas serve para a leitura via serial
do valor de umidade */
      // Leitura da tensão do sensor
      umidadeSolo = analogRead(pinoSensor);
      // Conversão do valor de tensão em valor de umidade
      umidadeSolo = map(umidadeSolo, 1023, 0, 0, 100);

      // Mostra o valor através da conexão serial
      Serial.println(umidadeSolo);
      Serial.println(" %    ");
    }
    delay(1000);
  }

  digitalWrite(pinoValvula, 0);
  // Espera o tempo estipulado
  delay(tempoRega*1000);
  digitalWrite(pinoValvula, 1);}
}
```

APÊNDICE C - Código do sistema fuzzy implementado

```
//*****
**
// Matlab .fis to arduino C converter v2.0.1.25122016
// - Karthik Nadig, USA
// Please report bugs to: karthiknadig@gmail.com
//*****
**

#define FIS_TYPE float
#define FIS_RESOLUTION 101
#define FIS_MIN -3.4028235E+38
#define FIS_MAX 3.4028235E+38
#define pinoSensor A0
#define pinoValvula 9
typedef FIS_TYPE(*_FIS_MF) (FIS_TYPE, FIS_TYPE*);
typedef FIS_TYPE(*_FIS_ARR_OP) (FIS_TYPE, FIS_TYPE);
typedef FIS_TYPE(*_FIS_ARR) (FIS_TYPE*, int, _FIS_ARR_OP);

int umidadeSolo = 0;

//*****
**
// Matlab .fis to arduino C converter v2.0.1.25122016
// - Karthik Nadig, USA
// Please report bugs to:
// https://github.com/karthiknadig/ArduinoFIS/issues
// If you don't have a GitHub account mail to karthiknadig@gmail.com
//*****
**

// Number of inputs to the fuzzy inference system
const int fis_gcI = 1;
// Number of outputs to the fuzzy inference system
const int fis_gcO = 1;
// Number of rules to the fuzzy inference system
```



```

const int fis_gcR = 3;

FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];

// Setup routine runs once when you press reset:
void setup()
{
    // initialize the Analog pins for input.
    // Pin mode for Input: umidade
    pinMode(pinoSensor , INPUT);

    // initialize the Analog pins for output.
    // Pin mode for Output: intensidade
    pinMode(pinoValvula , OUTPUT);

    Serial.begin(9600);

}

// Loop routine runs over and over again forever:
void loop(){
    digitalWrite(pinoValvula , 1);
    // Delay para medição de umidade e possível acionamento da válvula a
    // cada meio hora (1800 segundos).
    for(int i=0; i < 1800; i++) {
        // Faz a leitura do sensor de umidade do solo
        umidadeSolo = analogRead(pinoSensor);
        // Converte a variação do sensor de 0 a 1023 para 0 a 100
        umidadeSolo = map(umidadeSolo, 1023, 0, 0, 100);
        if(i == 900){
            // Exibe a mensagem via serial:
            Serial.println(umidadeSolo);
            Serial.println(" %    ");
        }
    }
}

```

```

    // Espera um segundo
    delay(1000);
}

// Read Input: umidade
g_fisInput[0] = umidadeSolo;

g_fisOutput[0] = 0;

fis_evaluate();

// Set output vlaue: intensidade
int outTime = g_fisOutput[0]*25;
digitalWrite(pinoValvula , 0);

if(g_fisInput[0] < 70 && g_fisInput[0] > 62){
    outTime = (70 - g_fisInput[0]);
}

Serial.println(g_fisInput[0]);
Serial.println(outTime);

delay(outTime*1000);

digitalWrite(pinoValvula , 1);

}

//*****
**
// Support functions for Fuzzy Inference System
//*****
**
// Trapezoidal Member Function
FIS_TYPE fis_trapmf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1], c = p[2], d = p[3];

```

```

    FIS_TYPE t1 = ((x <= c) ? 1 : ((d < x) ? 0 : ((c != d) ? ((d - x) /
(d - c)) : 0)));
    FIS_TYPE t2 = ((b <= x) ? 1 : ((x < a) ? 0 : ((a != b) ? ((x - a) /
(b - a)) : 0)));
    return (FIS_TYPE) min(t1, t2);
}

// Triangular Member Function
FIS_TYPE fis_trimf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1], c = p[2];
    FIS_TYPE t1 = (x - a) / (b - a);
    FIS_TYPE t2 = (c - x) / (c - b);
    if ((a == b) && (b == c)) return (FIS_TYPE) (x == a);
    if (a == b) return (FIS_TYPE) (t2*(b <= x)*(x <= c));
    if (b == c) return (FIS_TYPE) (t1*(a <= x)*(x <= b));
    t1 = min(t1, t2);
    return (FIS_TYPE) max(t1, 0);
}

FIS_TYPE fis_min(FIS_TYPE a, FIS_TYPE b)
{
    return min(a, b);
}

FIS_TYPE fis_max(FIS_TYPE a, FIS_TYPE b)
{
    return max(a, b);
}

FIS_TYPE fis_array_operation(FIS_TYPE *array, int size, _FIS_ARR_OP
pfnOp)
{
    int i;
    FIS_TYPE ret = 0;

    if (size == 0) return ret;

```

```

    if (size == 1) return array[0];

    ret = array[0];
    for (i = 1; i < size; i++)
    {
        ret = (*pfnOp)(ret, array[i]);
    }

    return ret;
}

//*****
**
// Data for Fuzzy Inference System
//*****
**
// Pointers to the implementations of member functions
_FIS_MF fis_gMF[] =
{
    fis_trapmf, fis_trimf
};

// Count of member function for each Input
int fis_gIMFCount[] = { 3 };

// Count of member function for each Output
int fis_gOMFCount[] = { 3 };

// Coefficients for the Input Member Functions
FIS_TYPE fis_gMFI0Coeff1[] = { -10, 0, 35, 50 };
FIS_TYPE fis_gMFI0Coeff2[] = { 40, 52.5, 65 };
FIS_TYPE fis_gMFI0Coeff3[] = { 60, 78, 100, 110 };
FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1, fis_gMFI0Coeff2,
fis_gMFI0Coeff3 };
FIS_TYPE** fis_gMFI0Coeff[] = { fis_gMFI0Coeff };

```

```

// Coefficients for the Output Member Functions
FIS_TYPE fis_gMFO0Coeff1[] = { -0.4167, 0, 0.4 };
FIS_TYPE fis_gMFO0Coeff2[] = { 0.3, 0.5, 0.7 };
FIS_TYPE fis_gMFO0Coeff3[] = { 0.6, 1, 1.416 };
FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1, fis_gMFO0Coeff2,
fis_gMFO0Coeff3 };
FIS_TYPE** fis_gMFOCoeff[] = { fis_gMFO0Coeff };

// Input membership function set
int fis_gMFI0[] = { 0, 1, 0 };
int* fis_gMFI[] = { fis_gMFI0};

// Output membership function set
int fis_gMFO0[] = { 1, 1, 1 };
int* fis_gMFO[] = { fis_gMFO0};

// Rule Weights
FIS_TYPE fis_gRWeight[] = { 1, 1, 1 };

// Rule Type
int fis_gRType[] = { 1, 1, 1 };

// Rule Inputs
int fis_gRI0[] = { 1 };
int fis_gRI1[] = { 2 };
int fis_gRI2[] = { 3 };
int* fis_gRI[] = { fis_gRI0, fis_gRI1, fis_gRI2 };

// Rule Outputs
int fis_gRO0[] = { 3 };
int fis_gRO1[] = { 2 };
int fis_gRO2[] = { 1 };
int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2 };

// Input range Min
FIS_TYPE fis_gIMin[] = { 0 };

```

```

// Input range Max
FIS_TYPE fis_gIMax[] = { 100 };

// Output range Min
FIS_TYPE fis_gOMin[] = { 0 };

// Output range Max
FIS_TYPE fis_gOMax[] = { 1 };

//*****
**
// Data dependent support functions for Fuzzy Inference System
//*****
**
FIS_TYPE fis_MF_out(FIS_TYPE** fuzzyRuleSet, FIS_TYPE x, int o)
{
    FIS_TYPE mfOut;
    int r;

    for (r = 0; r < fis_gcR; ++r)
    {
        int index = fis_gRO[r][o];
        if (index > 0)
        {
            index = index - 1;
            mfOut = (fis_gMF[fis_gMFO[o][index]])(x,
fis_gMFOCoeff[o][index]);
        }
        else if (index < 0)
        {
            index = -index - 1;
            mfOut = 1 - (fis_gMF[fis_gMFO[o][index]])(x,
fis_gMFOCoeff[o][index]);
        }
        else
        {

```

```

        mfOut = 0;
    }

    fuzzyRuleSet[0][r] = fis_min(mfOut, fuzzyRuleSet[1][r]);
}
return fis_array_operation(fuzzyRuleSet[0], fis_gcR, fis_max);
}

FIS_TYPE fis_defuzz_lom(FIS_TYPE** fuzzyRuleSet, int o)
{
    FIS_TYPE step = (fis_gOMax[o] - fis_gOMin[o]) / (FIS_RESOLUTION -
1);
    FIS_TYPE max = 0, dist, value;
    FIS_TYPE max2 = fis_gOMin[o];
    int i, midx = 0;

    for (i = 0; i < FIS_RESOLUTION; ++i)
    {
        dist = fis_gOMin[o] + (step * i);
        value = fis_MF_out(fuzzyRuleSet, dist, o);
        max = max(max, value);
    }

    for (i = 0; i < FIS_RESOLUTION; ++i)
    {
        dist = fis_gOMin[o] + (step * i);
        value = fis_MF_out(fuzzyRuleSet, dist, o);
        if (max == value)
        {
            dist = abs(dist);
            if (max2 < dist)
            {
                max = dist;
                midx = i;
            }
        }
    }
}

```

```

    }

    return (fis_gOMin[o] + (step * midx));
}

//*****
**
// Fuzzy Inference System
//*****
**
void fis_evaluate()
{
    FIS_TYPE fuzzyInput0[] = { 0, 0, 0 };
    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0, };
    FIS_TYPE fuzzyOutput0[] = { 0, 0, 0 };
    FIS_TYPE* fuzzyOutput[fis_gcO] = { fuzzyOutput0, };
    FIS_TYPE fuzzyRules[fis_gcR] = { 0 };
    FIS_TYPE fuzzyFires[fis_gcR] = { 0 };
    FIS_TYPE* fuzzyRuleSet[] = { fuzzyRules, fuzzyFires };
    FIS_TYPE sW = 0;

    // Transforming input to fuzzy Input
    int i, j, r, o;
    for (i = 0; i < fis_gcI; ++i)
    {
        for (j = 0; j < fis_gIMFCount[i]; ++j)
        {
            fuzzyInput[i][j] =
                (fis_gMF[fis_gMFI[i][j]]) (g_fisInput[i],
fis_gMFICoeff[i][j]);
        }
    }

    int index = 0;
    for (r = 0; r < fis_gcR; ++r)
    {
        if (fis_gRType[r] == 1)

```



```

    {
        fuzzyFires[r] = FIS_MAX;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r],
fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1 -
fuzzyInput[i][-index - 1]);
            else
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1);
        }
    }
    else
    {
        fuzzyFires[r] = FIS_MIN;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_max(fuzzyFires[r],
fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_max(fuzzyFires[r], 1 -
fuzzyInput[i][-index - 1]);
            else
                fuzzyFires[r] = fis_max(fuzzyFires[r], 0);
        }
    }

    fuzzyFires[r] = fis_gRWeight[r] * fuzzyFires[r];
    sW += fuzzyFires[r];
}

if (sW == 0)

```

```
{
    for (o = 0; o < fis_gc0; ++o)
    {
        g_fisOutput[o] = ((fis_gOMax[o] + fis_gOMin[o]) / 2);
    }
}
else
{
    for (o = 0; o < fis_gc0; ++o)
    {
        g_fisOutput[o] = fis_defuzz_lom(fuzzyRuleSet, o);
    }
}
}
```