

IMPLEMENTAÇÃO DE LÓGICA FUZZY EMBARCADA NO RASPBERRY PI, PARA O CONTROLE DE SISTEMAS DE ABASTECIMENTO DE ÁGUA

Walter Luiz Ferreira dos Anjos – walteranjos2005@yahoo.com.br
Saul da Silva Munareto – saul.munareto@ifes.edu.br
Richard Junior Manuel Godinez Tello – richard@ifes.edu.br
Instituto Federal do Espírito Santo – IFES
ES-010, Km-6,5 - Manguinhos
29173-087 – Serra – ES

Resumo: *Esse artigo apresenta o desenvolvimento de um Controlador Fuzzy embarcado no Raspberry PI3, para controle de pressão em Sistemas de Abastecimento de Água, através da Variação da velocidade do Conjunto Moto- Bomba.*

A aquisição de sinais de pressão e de set-point é feita através de um conversor analógico-digital de 15bits, o qual transmite para o Raspberry PI3 via protocolo I2C.

Com base nas regras de Inferência, dos sinais de set-point de pressão e da pressão do sistema, o controlador gera um sinal PWM de controle, o qual é aplicado a um Inversor de frequências para controle da velocidade do Conjunto Moto-Bomba.

Todo o processamento do controle Fuzzy é realizado no Raspberry PI3 e a programação feita na linguagem Python, o que possibilitou a construção de uma plataforma compacta de baixo custo, que pode ser expandida e alterada conforme necessidades.

Palavras-chave: *Água, Raspiberry PI, Controlador Fuzzy.*

1 INTRODUÇÃO

Na indústria, a necessidade de se controlar sistemas e processos revelou-se de fundamental importância na enorme expansão de bens disponíveis.

Os sistemas de abastecimento de água, que compreendem a captação, tratamento e distribuição, são constituídos por Estações de Elevatórias de Água Bruta (EEABs), Estações de Tratamento de Água (ETAs) e Estações Elevatórias de Água Tratada (EEATs). Em todo o processo desde a captação da água bruta até a distribuição ao consumidor final, os conjuntos moto-bombas (motor e bomba) constituem os ativos principais, os corações do sistema de abastecimento de água. Com potências variando de algumas frações até milhares de quilowatt (kW) representam as cargas com maior consumo de energia de um sistema de abastecimento, tornando-se necessário sistemas de controle eficazes, de modo que operem com um consumo mínimo de energia.

Os sistemas de abastecimento de água utilizam atualmente controladores PID's (Proporcional-Integral-Derivativo) [1], pois eles possuem estruturas que são simples e práticas no controle dos processos envolvidos, atuando no controle da velocidade de bombeamento e na diminuição do erro de estado estacionário.

O controle PID clássico, trabalha com um único modelo e com um único objetivo. Embora seja muito eficiente no controle de sistemas lineares simples, dificuldades são observadas quando o modelo não é capaz de atuar sobre variações não previstas [2] ou quando o sistema é não linear [3], como é o caso de um Sistema de Distribuição de Água Tratada, que são sistemas sujeitos à variação de consumo diário, à sazonalidade, e ao crescimento populacional de forma não linear [4], o que torna frequente a necessidade de ajuste do PID [4]. Devido essa característica, torna-se necessário que seja utilizado no processo, controladores capazes responder a essa não linearidade. Nesses processos, o controle Fuzzy, também chamado controle nebuloso, vem se mostrando uma excelente alternativa [5].

A possibilidade de economia de energia utilizando controladores fuzzy em sistema pressurizados de abastecimento de água, também foi abordada por Bezerra et al. [6], através de pesquisas realizadas em laboratório.

A lógica Fuzzy é uma técnica da inteligência artificial que incorpora a forma de pensar do ser humano em um sistema de controle [7]. Ela surgiu em 1965 com a publicação de um artigo sobre a teoria dos conjuntos Fuzzy por L.A. Zadeh, com o objetivo de fornecer um ferramental matemático para tratamento de informações de caráter vago ou impreciso,

O trabalho de Moratori et al [5] que compara os resultados de um controlador PID com um Controlador Fuzzy, no controle de um sistema de abastecimento de água e o trabalho de Bezerra et al [6] que apresenta um sistema Fuzzy para controle de pressão de uma rede de distribuição de água, mostram a eficiência do controlador Fuzzy no controle de processos não lineares.

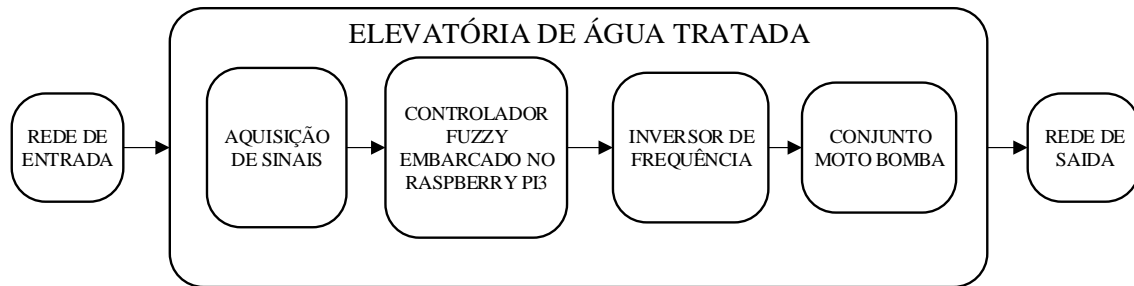
Na Indústria, normalmente, o controle Fuzzy é implementado em um Controlador Lógico Programável (PLC), em Sistemas Supervisórios ou em Controladores dedicados para essa função. Tal implementação, devido os custos envolvidos, muitas vezes torna inviável a relação custo benefício, principalmente para sistemas de pequeno porte ou de menor importância para a planta.

Esse artigo mostra a implementação de um controlador Fuzzy, de baixo custo, no Raspberry PI, para aplicação no controle de pressão de sistemas de abastecimento de água de pequeno porte.

O Raspberry PI3 juntamente com um sistema de aquisição, formam o controlador embarcado em questão. Um conversor analógico/digital de 15 bits recebe o sinal de set-point e a da pressão a ser controlada e transmite para o Raspberry PI3 via protocolo I2C, o qual faz o processamento conforme as regras de inferência fuzzy, gerando um sinal de saída PWM, para o controle do inversor ao qual o conjunto moto-bomba está ligado. A utilização dessa plataforma de processamento na Indústria se mostra uma excelente alternativa de baixo custo para pequenas aplicações, devido ao seu custo reduzido, capacidade de processamento e interfaces de entrada e saída disponíveis, bem como a sua utilização no campo acadêmico para estudos e desenvolvimentos, visto que o Raspberry PI3 é um sistema aberto com uma vasta gama de interfaces de entrada/saídas que podem ser controladas via software.

2 DESCRIÇÃO DO SISTEMA

Figura 1 – Diagrama do sistema de abastecimento



Fonte – Elaborado pelo autor

O Sistema de abastecimento de água tratada em questão, compreende uma Estação Elevatória de Água Tratada (EEAT) formada por um conjunto Moto-Bomba acionado por um Inversor de frequências e abastecida por um Reservatório de Água ou por uma Estação de Tratamento de água. A Elevatória tem como função bombear água para a rede de abastecimento do consumidor final, mantendo a pressão em um valor previamente ajustado.

3 O SISTEMA DE CONTROLE FUZZY PROPOSTO

O sistema de controle em questão, consiste de um controlador Fuzzy embarcado no Raspberry PI3, que recebe os sinais de pressão de água de entrada e saída da elevatória e, conforme o set-point de pressão previamente ajustado, gera os sinais de controle do Inversor, fig.1. O sistema de aquisição é formado por um conversor analógico/digital de 15 bits e transmissores de pressão, ligados aos barriletes de sucção e recalque da bomba. Os sinais de pressão e o set point de operação são enviados para o controlador que, com base nas regras de inferência Fuzzy e no sinal de set-point, gera o sinal de controle do Inversor.

3.1 Etapas do controle Fuzzy

A Lógica Fuzzy (Lógica Nebulosa) é a lógica que suporta os modos de raciocínio que são aproximados ao invés de exatos [8]. Ela é utilizada para tratar problemas onde a imprecisão e a incerteza são variáveis complexas que dificultam a implementação nos moldes convencionais [9]. Derivada do conceito de conjuntos Fuzzy, a lógica Fuzzy pode ser utilizada em sistemas lineares e não lineares, o que possibilita sua aplicação em sistemas onde o controle PID clássico não apresenta bom desempenho devido a não linearidade. No controle Fuzzy a metodologia de projeto está focalizada no comportamento dos operadores [8], ou seja, como eles ajustariam os parâmetros de controle para um determinado conjunto de circunstâncias. [9] As Estratégias de controle Fuzzy nascem da experiência e de experimentos, em vez de modelos matemáticos, tornando a implementação linguística (baseada em regras) muitas vezes mais rápida e efetiva de se implementar. Sua implementação envolve as seguintes fases: Pré-processamento, Fuzzyficação, Inferência, Defuzzyficação e Pós-Processamento.

- **Pré-processamento**

Recebe as medições realizadas por sistemas de aquisição de dados, realiza o condicionamento dos valores medidos antes que entrem ao controlador (Normalização, filtragem de ruído, cálculo da média, diferenciação, integração).

- **Fuzzyficação**

Utiliza variáveis linguísticas como exemplo: baixa, média, alta. Os valores são descritos por intermédio de conjuntos Fuzzy, representados por funções de pertinência como triangular, trapezoidal e gaussiana. Realiza a avaliação das entradas de acordo às regras e retorna como resultado um nível de pertinência.

- **Inferência**

Conjunto de Regras no formato SE-ENTÃO, que relacionam as variáveis do controlador com as variáveis de entrada para obter conclusões caracterizando a capacidade de raciocínio do sistema. As regras podem conter múltiplas entradas e saídas. Para definir as ações do sinal de controle pode-se utilizar o sinal de erro, a variação do sinal de erro, a integral do sinal de erro. [11].

Um ou mais especialistas da área, deverão elaborar em conjunto as regras de inferência do sistema Fuzzy.

- **Pós-processamento**

Realiza o escalonamento do valor de saída (se for preciso). Pode-se incluir um ganho de saída que pode ser ajustado. Por exemplo uma normalização de [-1 1] pode ser escalonada a uma \grandeza de tensão [-10 10] volts.

3.2 Raspberry PI 3

O Raspberry PI3, (fig2) é um microcomputador portátil, single board, com processador ARM quad-core de 1.2Ghz, 64 bits produzido pela Raspberry Pi Foundation. Possui 1Ghz de memória RAM, 4 portas USB, suporte total a HDMI, wireless LAN e Bluetooth. Utiliza um sistema operacional baseado em linux com boot a partir de um cartão SD. Possui uma GPIO com 40 pinos, com comunicação I2C, entradas e saídas digitais, uma USART, dentre outros[12].

Figura 2 – Pertinência de um conjunto Fuzzy

Figura 2 – Raspberry PI3,



Fonte – www.RaspberryPI.org

O Raspberry PI3, (fig2) possui instalado interpretadores Python 2 e 3. Python é uma linguagem interpretada, orientada a objetos, que possui recursos avançados de manipulação de textos, listas e outras estruturas de dados. Apresenta estruturas de loops indentadas em substituição do “Begin” e “End”.

Na área de Controle de Processos, possui bibliotecas específicas para Controle (matplotlib) e manipulação de matrizes (numpy), com funções semelhantes às do Matlab. Possui funções diversificadas para plotagem de dados em várias bibliotecas.

3.3 Biblioteca Fuzzy

Essa biblioteca contém um pacote que implementa a lógica Fuzzy no ambiente Python[13], chamado SkFuzzy[14], a qual possui funções de pertinência triangulares, gaussianas, trapezoidais, dentre outras. Muitas funcionalidades desse bloco Fuzzy, provém de pacotes agregados a ele, que compoem os módulos das funções de inferência, os módulos de operação entre os conjuntos Fuzzy e operações gráficas \\

A biblioteca Python-Fuzzy deve ser importada para o Python através de comandos específicos. [15]. A Python-Fuzzy usa arrays Numpy, o qual é o pacote fundamental para computação científica do Python. Ela é a biblioteca do Python que suporta arrays e matrizes multidimensionais, possuindo uma larga coleção de funções matemáticas para operações rápidas com arrays, incluindo operações matemáticas, lógicas, manipulação de formas, seleções, I/O, transformada de Fourier, álgebra linear básica, operações estatísticas básicas, simulações aleatórias, etc...

Numpy possui algumas diferenças do Matlab[14]:

- Os elementos dos vetores devem ser separados por vírgula. Por exemplo: $a=[1,2,6]$;
- Funções de transferência são normalmente implementadas para sistemas SISO. Para sistemas MIMO deve ser utilizado a representação em espaço de estados;
- Em Numpy o primeiro elemento possui índice zero. O elemento inicial é $a[0]$ ao contrário do Matlab cujo elemento inicial é 1. Numpy também permite o acesso aos elementos do vetor em qualquer ordem, sendo o último elemento $a[-1]$, o penúltimo $a[-2]$. Por exemplo, se um vetor contém 2 elementos eles podem ser referenciados da seguinte forma: o primeiro elemento é $a[0]$ ou $a[-2]$, o segundo $a[1]$, que também é $a[-1]$.

3.4 METODOLOGIA

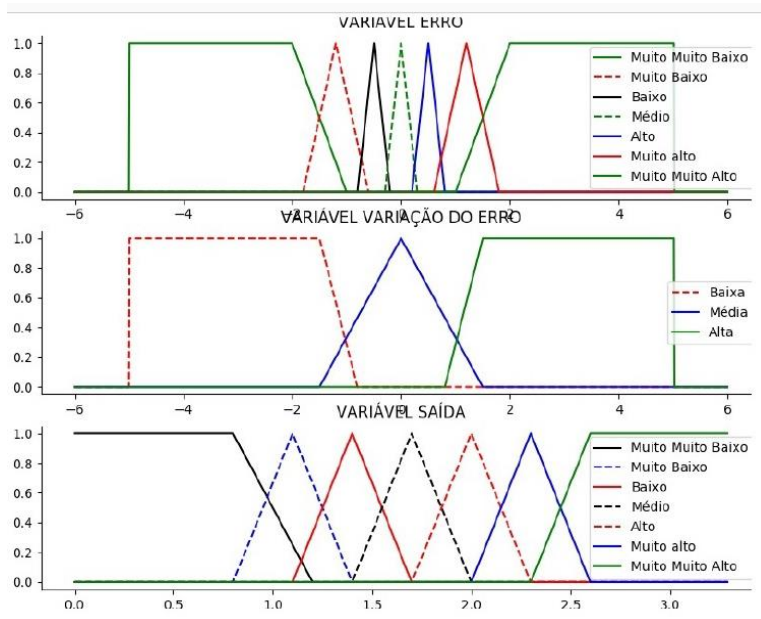
A metodologia utilizada para validação da implementação do controlador Fuzzy no Raspberry PI3 foi a implementação do controlador Fuzzy propriamente dito no Raspberry PI, utilizando a linguagem Python 3, seguida da implementação de hardware para aquisição do set-point e da variável controlada.

Para a execução da lógica Fuzzy no Raspberry PI3 foi necessário a instalação do módulo Scikit Fuzzy, além da utilização de outras bibliotecas, como a de manipulação de matrizes (numpy) e a de plotagem (matplotlib).

Foram escolhidas as variáveis linguísticas erro, variação do erro e tensão de saída para descrever o comportamento do sistema. Cada uma dessas variáveis com suas funções de pertinências específicas, conforme o comportamento desejado.

A variável Erro, que representa o erro entre o set-point de pressão ajustado e a pressão na saída da bomba e a variável saída, possuem as funções de pertinência Muito Muito Alto, Muito Alto, Alto, Médio, Baixo, Muito Baixo, Muito Muito Baixo (fig3). A variável variação do erro representa a variação do erro entre duas leituras consecutivas (Delta erro), e possui as funções de pertinência Baixa, Médio, Alto. Ao todo foram definidas 21 regras relacionando a variável erro, delta erro a saída, fig.3. As regras foram estabelecidas com as condições "SE", conforme: Se o "Erro" é "Muito Muito Baixo" "E" a variável de saída é "Baixa", então a saída do controlador será "Muito Muito baixa".

Figura 3 – Variáveis de entrada e saída com funções de pertinência



Fonte – Fonte própria simulada no Raspberry PI3

Foi utilizado uma grande quantidade de funções de pertinência com o objetivo de buscar uma maior precisão na determinação dos sinais de controle nas proximidades do ponto de operação.

FUNÇÕES DE PERTINÊNCIA

#Variável erro

```
Erro_MMB = fuzz.trapmf(Erro, [-5,-5,-2,-1])
Erro_MB = fuzz.trimf(Erro, [-1.8,-1.2,-0.6])
Erro_B = fuzz.trimf(Erro, [-0.8,-0.5,-0.2])
Erro_M = fuzz.trimf(Erro, [-0.3,0,0.3])
Erro_A = fuzz.trimf(Erro, [0.2,0.5,0.8])
Erro_MA = fuzz.trimf(Erro, [0.6,1.2,1.8])
Erro_MMA = fuzz.trapmf(Erro, [1,2,5,5])
```

#Variável Variação do erro

```
DeltaErro_B = fuzz.trapmf(DeltaErro, [-5,-5,-1.5,-0.8])
DeltaErro_M = fuzz.trimf(DeltaErro, [-1.5,0,1.5])
DeltaErro_A = fuzz.trapmf(DeltaErro, [0.8,1.5,5,5])
```

#Variável Tensão de saída

```
Saida_MMB = fuzz.trapmf(Saida, [0,0,0.8,1.2])
Saida_MB = fuzz.trimf(Saida, [0.8,1.1,1.4])
Saida_B = fuzz.trimf(Saida, [1.1,1.4,1.7])
Saida_M = fuzz.trimf(Saida, [1.4,1.7,2.0])
Saida_A = fuzz.trimf(Saida, [1.7,2.0,2.3])
Saida_MA = fuzz.trimf(Saida, [2.0,2.3,2.6])
Saida_MMA = fuzz.trapmf(Saida, [2.3,2.6,3.3,3.3])
```

Erros negativos indicam que a variável de processo está com um valor acima do set-point, e tem que ser reduzida. Erros positivos indicam que a variável de processo está abaixo do set-point e devem aumentar. Da mesma forma, variações no erro negativas indicam tendência a crescimentos negativos do erro e variações no erro positivas indicam tendência a decaimento da variável erro.

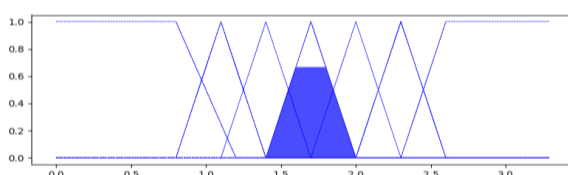
Após a inferência foi feito a agregação e a fuzzificação. Na agregação todas as regras ativas são combinadas para gerar uma saída com a representação dessas regras ativas. Após a agregação, foi feita a defuzzificação pelo método do centróide. A defuzzificação produziu uma saída de controle que varia de 0.507 como um valor mínimo e 2.87 para um valor máximo.

3.5 Experimentos

O sistema de controle Fuzzy foi implementado e testado em bancada. A saída PWM do controlador foi convertida em 0-10Volts e enviada à entrada analógica de um Inversor para controle da frequência do motor e um sinal de referência aplicado ao controlador, apresentando os resultados mostrados abaixo:

Se o erro atual aumenta em relação ao erro anterior, a saída do controlador diminui de forma a reduzir o erro (erro atual =0.1 e erro anterior=0.05) (fig4).

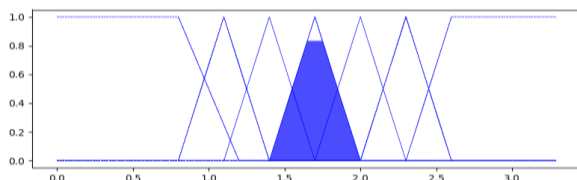
Figura 4 – Variável saída do controlador Fuzzy



Fonte – Fonte própria simulada no Raspberry PI

A medida que o erro e a variação do erro se estabilizam e aproximam de zero, a função de pertinência médio da variável saída se aproxima de 1 (100% - área preenchida da função de pertinência saída) (fig5).

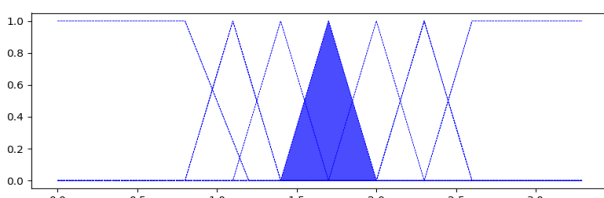
Figura 5 – Variável saída do controlador Fuzzy



Fonte – Fonte própria simulada no Raspberry PI

Se o erro e a variação do erro são iguais a zero, o sinal de saída assume o valor máximo desejado igual a 100% (função de pertinência médio da variável de saída) (fig6).

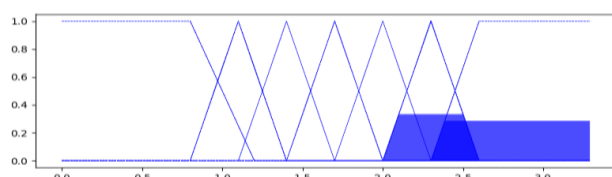
Figura 6 – Variável saída do controlador Fuzzy



Fonte – Fonte própria simulada no Raspberry PI

Se o erro atual é bem superior ao erro anterior, a saída aumenta rapidamente de forma a reduzir o erro (funções de pertinência Alto, Muito Alto e Muito Muito Alto) (fig7).

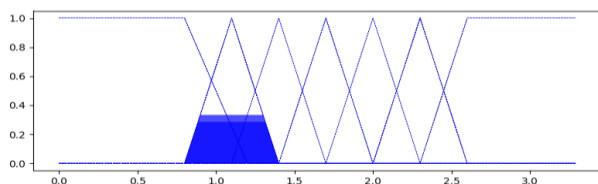
Figura 7 – Variável saída do controlador Fuzzy



Fonte – Fonte própria simulada no Raspberry PI

Se o erro atual é bem inferior ao erro anterior (erro negativo indica que a tensão de saída é superior ao set-point), a saída diminui rapidamente de forma a reduzir o erro (funções de pertinência Baixo, Muito Baixo e Muito Muito Baixo) (fig8).

Figura 8– Variável saída do controlador Fuzzy



Fonte – Fonte própria simulada no Raspberry PI

4 CONCLUSÕES

A utilização de funções de pertinência trapezoidais nos extremos trouxe uma resposta rápida ao controlador, quando o erro era muito grande e a concentração de funções mais estreitas na parte central, permitiu maior precisão de controle quando o erro era pequeno. Através módulo Scikit Fuzzy do Python 3, foi possível implementar o controlador Fuzzy embarcado no Raspberry PI 3. Tal implementação mostra a possibilidade de utilização do controlador em sistemas de abastecimento de água.

Agradecimentos

Agradeço a Deus pela oportunidade de ter obtido esse conhecimento, pela sabedoria e persistência frente aos obstáculos, a meus pais e minha esposa pelo incentivo e compreensão, aos professores do IFES pela dedicação e paciência comigo.

REFERÊNCIAS

- [1] Gomes, H. P. et al. **Sistemas de Bombeamento - Eficiência Energética**. 1ª edição. João Pessoa: Editora Universitária UFPB. 2009.
- [2] Bezerra, Saulo. **Sistema Fuzzy para Controle Piezométrico de Sistemas de Distribuição de Água Visando à Economia de Água e Energia**. 142 f. Tese (Doutorado em Engenharia Mecânica) - Universidade Federal da Paraíba, 2009.
- [3] Dorf, Bishop
DORF, Richard C.; BISHOP, Robert H. **Sistemas de controle modernos**. São Paulo: Editora LTC. 2001.

- [4] Barreto, Victor; Barbosa Douglas; Maziero Thiago; Hoff Victor. **Desenvolvimento de um Controlador Fuzzy para Controle de Pressão em Sistemas de Abastecimento de Água**, Universidade Federal Fluminense. 2016.
- [5] Moratori, Patrick; Barreto, Victor; Copetti Alessandro; Bertini Luciano. **Melhorando a Eficiência de um Sistema de Controle de Abastecimento de Água Utilizando Sistemas Nebulosos**. Universidade Federal Fluminense, Campus de Rio das Ostras, 2016.
- [6] Bezerra, Saulo; Silva, Simplício; Gomes, Heber. **Operational Optimisation of Water Supply Networks Using a Fuzzy System**. Water SA, v. 38 n. 4:565-572, 2012.
- [7] Simões, Marcelo G.; SHAW, Ian S. **Controle e modelagem Fuzzy**. São Paulo, 2007.
- [8] R.Tanscheit, **Sistemas Fuzzy** 1ª edição, São Paulo, 2007.
- [9] L.A. zadeh. **Fuzzy sets. Information and Control**. ISBN 8:338-353, 1965.
- [10] L.A. Prado; M. C. Masselli. **Aplicação de lógica Fuzzy e Arduino em controle de processos**. 2007.
- [11] Norris, Donald J. **Begining Artificial Intelligence with Raspberry PI**. Editora Apress. 2017.
- [12] RaspberryPI3 model B. Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> Acesso em 28/04/2018
- [13] Severance, Charles. **Python for everybody- Exploring data in Python 3**.
- [14] Matthes, Eric. **Python Crash Course**. Editora Ciência Moderna. 1972. ISBN 978-1-59327-603-4.
- [15] Campos, Mario M.; Saito Kaku. **Sistemas Inteligentes em controle e automação de processos**. Editora Ciência Moderna. 2004.

IMPLEMENTATION OF EMBEDDED FUZZY LOGIC IN RASPBERRYPI, FOR THE CONTROL OF WATER SUPPLY SYSTEMS

Abstract: *This paper presents the development of a Fuzzy Controller embedded in the Raspberry PI3, for pressure control in Water Supply Systems, through the Variation of the speed of the Motor Pump.*

The acquisition of pressure and set-point signals is done through a 15-bit analog-to-digital converter, which transmits to the Raspberry PI via the I2C protocol.

Based on the rules of Inference, pressure set-point signals and system pressure, the control controller generates a control PWM signal, which is applied to a Frequency Inverter to control the speed of the Motor-Pump Assembly.

All the processing of the Fuzzy control is realized in the Raspberry PI and the programming made in the Python language, which made possible the construction of a compact platform of low cost, that can be expanded and modified as needed.

Key-words: Water, Raspberry PI, Fuzzy Controller.