

CONTROLE DE TEMPERATURA DE UMA TOCHA DE PLASMA TÉRMICO COM ACOPLAMENTO INDUTIVO UTILIZANDO A PLACA ARDUINO

GUSTAVO FERNANDES DE LIMA*, GLAUCO GEORGE C. MANIÇOBA*, ANDRÉS ORTIZ SALAZAR*

*Laboratório de Avaliação de Medição em Petróleo
Centro de Tecnologia - Universidade Federal do Rio Grande do Norte
59072-970, Lagoa Nova, Natal, RN.

Emails: gustavoflima@msn.com, glaucogeorge@yahoo.com.br, andres@dca.ufrn.br

Abstract— This work presents the use of the Arduino board to control the temperature of a thermal plasma torch through fuzzy control. The paper describes the components of the cooling system water torch and results in real experiments. The results showed that it is possible to control the temperature of a plasma torch using the Arduino board.

Keywords— Arduino, torch, fuzzy logic, temperature, thermal plasma.

Resumo— Este trabalho apresenta a utilização da placa Arduino no controle de temperatura de uma tocha de plasma térmico por meio do controle *fuzzy*. O trabalho descreve os componentes do sistema de refrigeração a água da tocha e os resultados obtidos em experimentos reais. Os resultados mostram que é possível controlar a temperatura de uma tocha de plasma utilizando a placa Arduino.

Keywords— Arduino, tocha, lógica *fuzzy*, temperatura, plasma térmico.

1 Introdução

Arduino é uma plataforma de prototipagem eletrônica *open source* baseada em *hardware* e *software* flexíveis e fáceis de usar. É destinada aos artistas, designers, hobbistas, e qualquer pessoa interessada em criar objetos ou ambientes interativos (Banzi et al., 2006).

A placa Arduino pode interagir com o ambiente recebendo em suas entradas sinais dos mais variados tipos de sensores e pode atuar nesse mesmo ambiente por meio do acionamento de luzes, motores ou outros atuadores.

Neste trabalho o ambiente de interação do Arduino foi um sistema de refrigeração a água de uma tocha de plasma térmico. Ele recebeu em uma entrada o sinal elétrico de um sensor de temperatura e atuou na planta de refrigeração por meio de um inversor de frequência que acionou uma bomba d'água e, também, atuou em um ventilador acoplado a um radiador.

A forma como o sistema de refrigeração foi projetado permitiu a implementação de um controlador utilizando a lógica *fuzzy* embarcado na placa Arduino para atuar simultaneamente ou não na velocidade de giro da bomba d'água e no acionamento do ventilador do radiador.

2 Revisão Bibliográfica

2.1 Lógica Fuzzy

Em 1965 o professor Loft A. Zadeh da Universidade de Berkeley (USA) propôs uma nova teoria de conjuntos, onde a passagem da pertinência para a não pertinência fosse feita de uma forma lenta e gradual e não abrupta como na teoria usual de

conjuntos (Bilobrovec et al., 2004). Daí surgiram os Conjuntos *Fuzzy*, do inglês, nebuloso ou difuso.

Segundo Filho et al. (2002) a Lógica *Fuzzy* é uma técnica de inteligência artificial, disciplina que procura maneiras de máquinas simularem o raciocínio humano na solução de problemas diversos. O diferencial do controle *Fuzzy* é que ele permite modelar as ações a partir de informações fornecidas pelo operador, ou seja, conhecimento especialista, em vez de modelar o processo em si (Carvalho et al., 2010).

Um sistema baseado em Lógica *Fuzzy* pode ter sua ação esquematizada pelos seguintes elementos constituintes (Filho et al., 2002): Fuzzificador; Regras (base de conhecimento); Inferência (lógica de tomada de decisões); e Defuzzificador. Esta estrutura pode ser visualizada na figura 1.

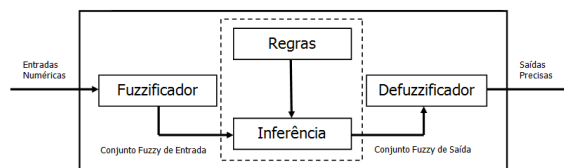


Figura 1: Blocos do Controlador *Fuzzy*.

O *fuzzificador* tem por função transformar os valores numéricos das entradas em suas respectivas variáveis lingüísticas.

A *base de conhecimento* representa o modelo do sistema a ser controlado, consistindo numa base de dados e uma base de *regras fuzzy lingüísticas*. A base de dados fornece definições numéricas e a base de regras caracteriza os objetivos do controlador e sua estratégia usada, geralmente fornecida por pessoas especialistas no sistema (Filho et al., 2002).

A *inferência* tem por característica transformar os valores lingüísticos de entrada, junto com as regras, em valores lingüísticos de saída.

O *defuzzificador* tem por objetivo traduzir os valores lingüísticos de saída, inferidos pelas regras, em valores numéricos de saída, utilizáveis numa ação de controle concreta no mundo real.

2.2 A placa Arduino

O Arduino é uma pequena placa de circuito impresso (6,8 x 5,5 x 1,0 cm), sendo indicado para criação de protótipos de eletrônica, baseado nas filosofias de *software* e *hardware* livres.

Para o presente projeto foi escolhido o modelo Arduino Uno pelas seguintes razões: possui conexão USB, conector tipo jack para alimentação externa, programação simples, preço baixo e fácil manuseio.

Uma placa Arduino Uno conta com 01 microcontrolador ATmega328P (fabricado pela empresa Atmel), 14 entradas/saídas digitais, 06 entradas analógicas, 01 oscilador de cristal de 16 MHz e pinos de alimentação com 3,3 V, 5 V e Terra (GND). Toda essa estrutura pode ser visualizada na figura 2, onde os principais blocos da placa foram identificados.

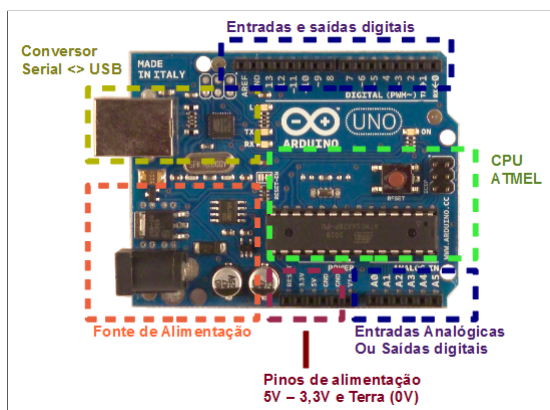


Figura 2: Blocos Identificados do Arduino Uno.
Fonte: www.robotizando.com.br

Ainda foram utilizados dois módulos conectados ao Arduino, para que este pudesse expandir suas funcionalidades. O primeiro módulo foi o de cartão SD (figura 3(a)) e o segundo foi o *display* de cristal líquido (LCD) 16 x 2 (figura 3(b)).

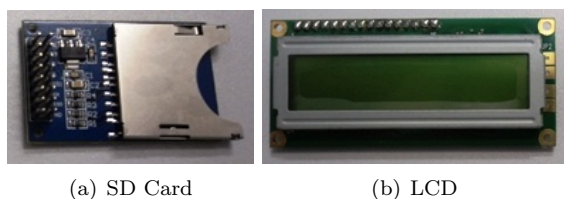


Figura 3: Módulos para Arduino.

O módulo de cartão tem por objetivo adicionar ao Arduino a função de armazenamento de dados, enquanto que, o módulo de LCD acrescenta a funcionalidade de exibição de dados.

Neste projeto, o cartão SD foi usado para o armazenamento das medições de temperatura e tempo em arquivo txt, enquanto que o *display* mostrou em tempo real os valores dessas mesmas grandezas para monitoramento.

2.3 Sensor de Temperatura NTC

No projeto foi utilizado um termistor NTC (Coeficiente de Temperatura Negativo), semiconductor de baixo custo e que diminui sua resistência elétrica com o aumento da temperatura. Tem como vantagem a medição de variação rápida de temperatura.

O sensor proposto, de acordo com o fabricante, trabalha a uma resistência de 520 Ω em 298,15 K (25 °C) e 75,41 Ω em 373,15 K (100 °C), com uma tolerância de 1 a 5 %, conferindo ao sensor uma curva característica não linear. Para obtenção de um gráfico dessa curva foi utilizada a expressão 1.

$$R = R_0 \cdot \exp \left[\beta \cdot \left(\frac{1}{T} - \frac{1}{T_0} \right) \right] \quad (1)$$

Onde R é a resistência do termistor na temperatura T , R_0 é a resistência do termistor na temperatura T_0 e β é a constante do material que vale 2.864,30 (Maniçoba, 2013).

A figura 4 apresenta a curva característica do sensor utilizado no projeto, com base nos dados do fabricante citados anteriormente.

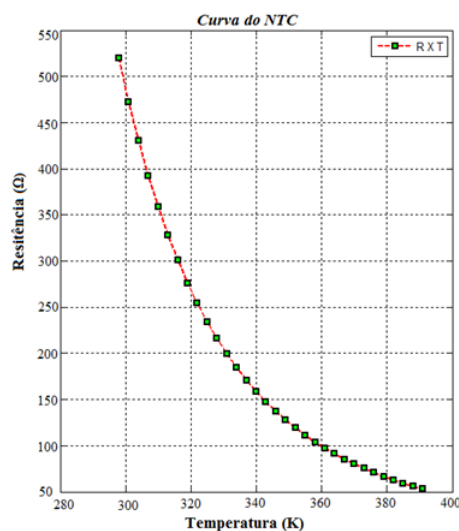


Figura 4: Curva Característica do Sensor NTC.

A entrada analógica do Arduino foi desenvolvida para receber leituras de tensão de até 5 V_{DC} . Por isso, surgiu a necessidade de conversão da resistência do NTC em um sinal de tensão

correspondente. Assim, foi desenhado um circuito eletrônico que convertesse a variação de resistência, decorrente da variação de temperatura lida, em um sinal analógico compatível com o Arduino.

Um circuito divisor de tensão foi escolhido para converter a resistência do sensor NTC em uma tensão mensurável pela placa Arduino. Na configuração escolhida, o divisor de tensão permitiu que um aumento de temperatura implicasse no aumento de tensão V_0 , dada pela expressão 2.

$$V_0 = V_{cc} \cdot \left(\frac{R_1}{R_1 + R_{NTC}} \right) \quad (2)$$

A saída do divisor de tensão V_0 foi conectada à entrada analógica A0 do Arduino e este ao receber essa tensão precisou calcular a resistência R_{NTC} , pela expressão 3.

$$R_{NTC} = \left(R_1 \cdot \frac{V_{cc}}{V_0} \right) - R_1 \quad (3)$$

De posse do valor de resistência R_{NTC} foi possível determinar a temperatura atual do sistema em Kelvin. A relação entre resistência e temperatura no sensor NTC é dada pela equação de Steinhart-Hart, conforme expressão 4.

$$T(K) = \frac{1}{a + b \cdot \ln(R_{NTC}) + c \cdot (\ln(R_{NTC}))^3} \quad (4)$$

Os coeficientes de Steinhart-Hart (a, b, c) são normalmente fornecidos pelos fabricantes de termistor ou podem ser determinados utilizando a equação 5.

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & \ln(R_1) & (\ln(R_1))^3 \\ 1 & \ln(R_2) & (\ln(R_2))^3 \\ 1 & \ln(R_3) & (\ln(R_3))^3 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 1/T_1 \\ 1/T_2 \\ 1/T_3 \end{bmatrix} \quad (5)$$

Para o projeto, tais coeficientes foram extraídos de Maniçoba (2013), cujos valores foram:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0,0012 \\ 0,00033151 \\ 0,0000019625 \end{bmatrix} \quad (6)$$

De posse desses coeficientes foi possível implementar o trecho de código fonte para Arduino capaz de converter a tensão lida pelo sensor NTC, calcular a resistência R_{NTC} , depois determinar a temperatura em Kelvin e por fim, converter de Kelvin para uma temperatura em graus Celsius, conforme código 1.

```
//Conversao da leitura sensor em temper graus C
TensaoSensor=analogRead(A0); // le porta A0
V0=TensaoSensor*5/1023; // converte p/ volts
Rntc=(5*150/V0)-150; // RNTC=(Vcc*R1/Vin)-R1
b1=log(Rntc); // LN(RNTC)
b1=0.00033151*b1; // b*LN(RNTC)
c1=log(Rntc); // LN(RNTC)
c1=0.0000019625*c1*c1*c1; // c*(LN(RNTC)^3)
tempK=0.0012+b1+c1; // a+b*LN(RNTC)+c*(LN(RNTC)^3)
tempK=1/tempK; // 1/(a+b*LN(RNTC)+c*(LN(RNTC)^3))
tempC=tempK-273.15; // converte p/ graus C
```

Código 1: Trecho de Código para Arduino.

2.4 Tocha de Plasma

O plasma pode ser definido como um gás parcialmente ionizado onde co-existem elétrons livres e íons positivos em movimento, em um processo onde ocorre transferência de energia por radiação, condução e aquecimento Joule entre outros (Giné-Rosias, 1998).

Já a tocha de plasma para Nishio and Takabatake (2002) é um dispositivo que transforma energia elétrica em calor transportado por um gás. Com estes dispositivos, virtualmente qualquer gás pode ser levado ao estado de plasma.

Para o projeto foi escolhida uma tocha de plasma ICP (Inductive Coupled Plasma). Segundo Maniçoba et al. (2012) uma tocha de plasma acoplado indutivamente tem por vantagem não possuir eletrodos metálicos internos, a serem erodidos pelo jato de plasma, evitando possíveis contaminações. A figura 5 apresenta uma imagem da tocha de plasma acoplado indutivamente utilizada no projeto.

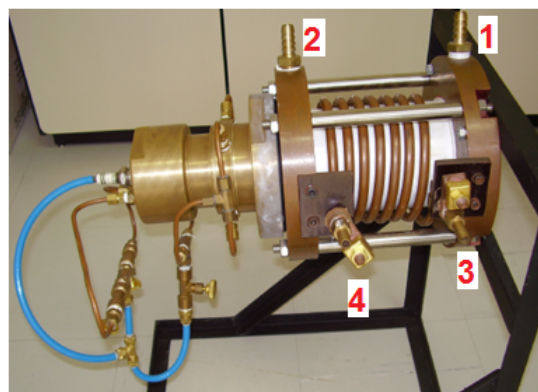


Figura 5: Tocha ICP do Projeto.

Esta tocha foi construída de maneira a permitir o fluxo de água, para resfriamento da mesma, por dois caminhos distintos. Ainda na figura 5 é possível ver uma numeração indicativa desses caminhos. No primeiro, a água entra pelo bocal de exaustão (nº 1), passa pelo tubo de confinamento (corpo de cor branca) e sai pela base da tocha (nº 2). O segundo caminho é por dentro da bobina, entrando no nº 3 e saindo no nº 4.

3 Metodologia

3.1 Sistema de Refrigeração a Água

Foi montado um sistema de refrigeração a água composto por um radiador (trocador de calor) tendo acoplado um ventilador acionado por um motor trifásico, um vaso de expansão, uma bomba d'água acionada por um inversor de frequência e um sensor NTC, conforme figura 6.

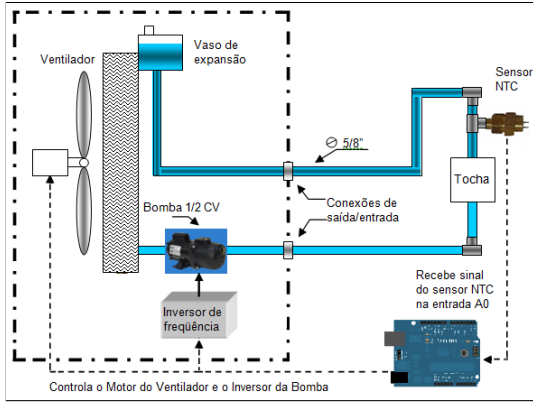


Figura 6: Sistema de Refrigeração da Tocha.

Tais componentes possuem as seguintes funcionalidades no sistema de refrigeração:

- **Radiador** - Constituído por aletas cujo objeto é trocar o calor da água aquecida pela tocha de plasma com o ar ambiente.
- **Vaso de Expansão** - Reservatório de plástico semitransparente que permite a visualização do nível da água.
- **Bomba d'água** - Responsável pela circulação de água dentro do sistema de resfriamento da tocha de plasma.
- **Sensor de Temperatura** - Responsável pelas medições de temperatura da água do sistema de resfriamento da tocha.

3.2 Funcionamento do Controle

O controle de temperatura proposto pode ser visualizado no diagrama de blocos da figura 7.

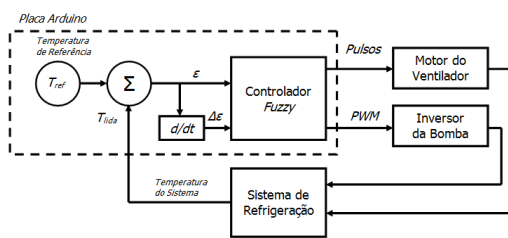


Figura 7: Diagrama de Blocos do Sistema.

Nesse diagrama, a temperatura da água do sistema foi lida pelo Arduino e este a subtraiu da temperatura de referência para determinar o erro (ϵ), conforme expressão 7. Esta é a primeira variável do controlador *fuzzy* e medida em $^{\circ}\text{C}$.

$$\epsilon (^{\circ}\text{C}) = T_{lida} - T_{ref} \quad (7)$$

Depois este valor de erro (atual) foi subtraído do valor de erro anterior para produzir a variável 'taxa de variação do erro' ($\Delta\epsilon$), conforme

expressão 8. Esta é a segunda variável do controlador *fuzzy* e medida em $^{\circ}\text{C}/\text{min}$.

$$\Delta\epsilon (^{\circ}\text{C}/\text{min}) = \frac{\epsilon_{atual} - \epsilon_{anterior}}{dt} \quad (8)$$

De posse desses dois valores, o controlador *fuzzy* combinou as variáveis ϵ e $\Delta\epsilon$ em cada uma das regras de inferência, do tipo **se ... então**, e produziu as saídas apropriadas. As saídas individuais de cada regra foram *defuzzificadas* gerando dois sinais de saída. O primeiro para acionamento do inversor de frequência e o segundo para acionamento do ventilador do radiador.

3.3 Controlador Fuzzy

Foi utilizada a ferramenta *Fuzzy Logical Toolbox* do programa MATLAB[®] para criar as variáveis de entrada do sistema *fuzzy* e assim determinar os valores das variáveis de saída.

A variável erro (ϵ) foi definida com valores de -55 a 55 $^{\circ}\text{C}$ e figura 8 ilustra o conjunto *fuzzy* gerado, onde os valores linguísticos possíveis para o erro foram: baixa (B), média (M), no *setpoint* (SP), alta (A) e muito alta (MA).

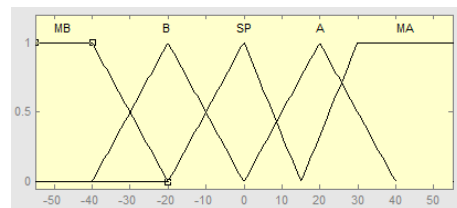


Figura 8: Pertinências da Variável 'Erro'.

A variável 'taxa de erro' foi definida na faixa de $-5,5$ a $5,5$ $^{\circ}\text{C}/\text{s}$. Valores negativos indicam diminuição da temperatura e valores positivos indicam elevação da temperatura. A figura 9 ilustra o conjunto *fuzzy* gerado, onde os valores linguísticos definidos foram: caindo rápido (CR), caindo devagar (CD), estável (ES), subindo devagar (SD) e subindo rápido (SR).

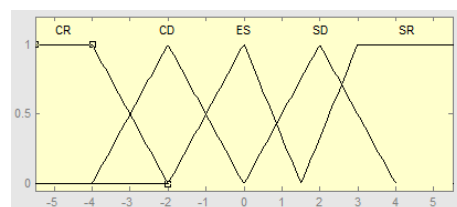


Figura 9: Pertinências da Variável 'Taxa de Erro'.

A primeira variável de saída foi chamada de 'PWM' e representou o sinal de acionamento do inversor de frequência da bomba d'água. Esta variável controlou o estado de giro da bomba variando de 0 a 100%, aumentando ou diminuindo o fluxo d'água no sistema de refrigeração. A figura

10 ilustra o conjunto *fuzzy* gerado, onde os valores linguísticos foram: pequena (P), média (M), grande (G) e muito grande (MG).

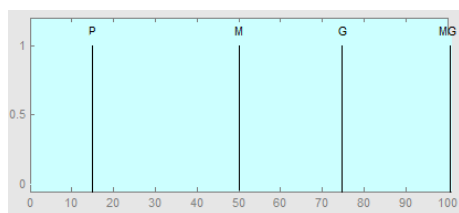


Figura 10: Pertinências da Variável 'PWM'.

A segunda variável de saída foi chamada 'Pulso' e representou o sinal de desligado e ligado do ventilador do radiador, conforme figura 11.

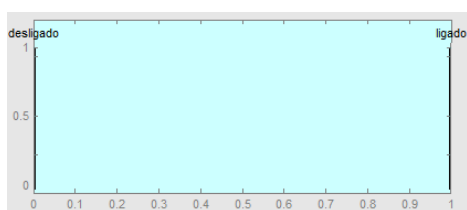


Figura 11: Pertinências da Variável 'Pulso'.

O Bloco de Regras produzido pela ferramenta *Fuzzy Logical Toolbox* possuiu 25 regras, de maneira a cobrir todas as combinações das duas variáveis de entrada. Essas regras foram organizadas na tabela 1.

Tabela 1: Regras *Fuzzy*

PWM/Pulso	ϵ					
	B	M	SP	A	MA	
$\Delta\epsilon$	CR	DP/D	P/D	P/-	M/L	M/L
	CD	P/D	P/D	P/-	M/L	G/L
	ES	P/D	P/D	P/-	G/L	MG/L
	SD	P/D	P/D	M/-	G/L	MG/L
	SR	P/D	M/D	G/-	MG/L	MG/L

Para a máquina de inferência foi usada a *Mamdani*, que utiliza como implicação a t-norma mínimo [min] e como agregação a sua co-norma máximo [max]. Inicialmente esta base de regras foi simulada várias vezes com o programa MATLAB® a fim de testar sua coerência e consistência. A *defuzzificação* foi calculada utilizando a média da pertinência máxima.

A figura 12 apresenta um exemplo da utilização da base de regras para o cálculo das variáveis de saída. Quando o erro calculado foi - 25 °C (1ª coluna) e a taxa de erro foi 0,6 °C/min (2ª coluna) produziram uma velocidade da bomba igual a 15 % (3ª coluna) e ventilador desligado (4ª coluna). Com essas condições foi permitido uma a elevação de temperatura até chegar na referência.

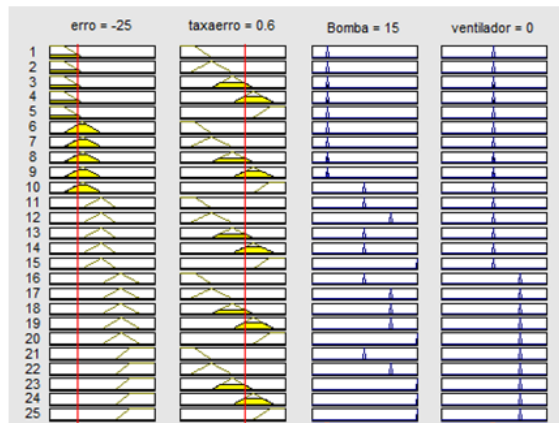


Figura 12: Exemplo de Cálculo de Variáveis.

4 Resultados e Discussões

Para demonstrar o funcionamento do controle foram realizados cinco experimentos reais. Em todas, a temperatura de referência foi configurada em 45 °C, por motivos de segurança. Em função dessa referência, a temperatura de acionamento do ventilador do radiador foi configurada em 50 °C e depois do ventilador ligado, só desligou quando a temperatura chegou em 40 °C.

A figura 13 apresenta a resposta da temperatura no tempo, com o controlador *fuzzy* configurado para manter a variável 'PWM' em um valor contante. No primeiro ensaio esse valor foi 15%, no segundo foi 50%, assim por diante. As leituras de temperatura foram realizadas a cada um minuto e por um tempo total de 40 minutos.

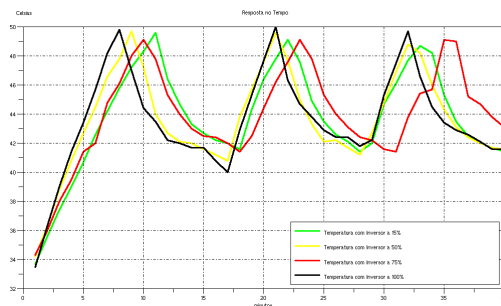


Figura 13: Temperaturas a Velocidade Constante.

Analisando a figura 13 foi possível visualizar que a temperatura do sistema com a variável 'PWM' em 75% (vermelha) apresentou melhor resultado, pois variou entre 41,5 °C e 49,1 °C.

A figura 14 apresenta a resposta da temperatura no tempo, com o controlador *fuzzy* configurado para qualquer valor da variável 'PWM'. Isso permitiu que a velocidade da bomba d'água pudesse variar livremente ao longo da realização do quinto ensaio.

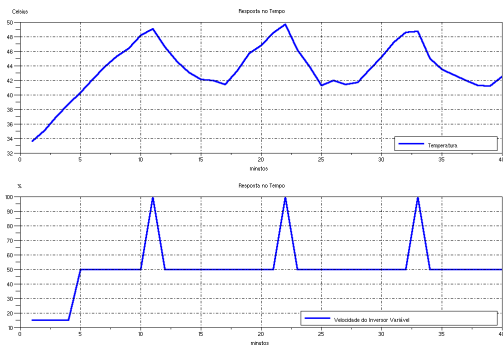


Figura 14: Temperaturas a Velocidade Variável.

Ainda na figura 14, foi possível ver os picos de velocidade coincidirem com os picos de temperatura, representando o início do resfriamento da água do sistema de refrigeração. A variação de temperatura ficou entre 41,2 °C e 49,7 °C.

E a figura 15 apresenta uma superposição dos cinco experimentos realizados. Isto permitiu visualizar o comportamento da temperatura quando a variável 'PWM' foi mantida constante (quatro ensaios) e quando ela variou livremente (quinta ensaio). Em todas o controlador *fuzzy* operou como esperado e permitiu que a variação de temperatura fosse menor do que a faixa predefinida de 40 a 50 °C.

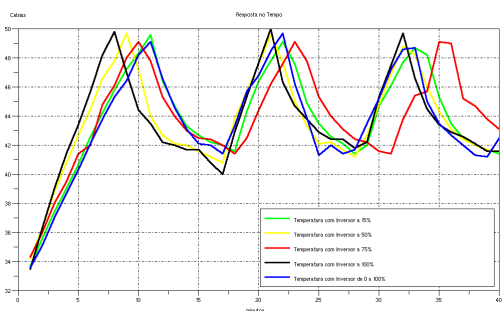


Figura 15: Superposição de Respostas.

Este resultado será melhor quando novos experimentos forem realizados a partir de novos ajustes das pertinências do controlador *fuzzy*. E quando o acionamento do ventilador do radiador for por meio de um inversor de frequência.

5 Conclusões

Neste trabalho foi apresentada a utilização da placa Arduino no controle de temperatura da água que resfria uma tocha de plasma térmico por meio do controle *fuzzy*. O objetivo foi alcançado, uma vez que, a temperatura da água variou entre 41,2 °C e 49,7 °C dentro da faixa predefinida de 40 a 50 °C. Tal resultado foi possível porque a temperatura é uma grandeza que varia lentamente, o que permitiu ao Arduino trabalhar sem grandes esforços computacionais.

Agradecimentos

Ao LAMP, FINEP e PPGEEC/UFRN.

Referências

- Banzi, M., Cuartielles, D., Igoe, T., Martino, G. and Mellis, D. (2006). **Site Arduino**, www.arduino.cc. Acesso em: 26 abr. 2013.
- Bilobrovec, M., Marçal, R. F. M. and Kovaleski, J. L. (2004). **Implementação de um Sistema de Controle Inteligente Utilizando a Lógica Fuzzy, XI SIMPEP**, Bauru, SP.
- Carvalho, A. S., de Sousa, A. L., dos Santos Martins, L. and Bringhamti, P. M. (2010). **Modelo Autorregressivo e Controlador Fuzzy para um Sistema de Controle Digital de Velocidade do Kit Digital Servo, VII Simpósio de Excelência em Gestão e Tecnologia**.
- Filho, F. M. L., Gosmann, H. L. and Bauchspiess, A. (2002). **Controle Fuzzy para Sistema de Nível de Líquidos, XIV Congresso Brasileiro de Automática**, Natal, RN, pp. 3017–3022.
- Giné-Rosias, M. F. (1998). **Espectrometria de Emissão Atômica com Plasma Acoplado Indutivamente . (ICP-AES)**, 1 edn, CENA, Piracicaba, SP. (Série Didática, v.3).
- Maniçoba, G. G. C. (2013). **Controle de Temperatura do Sistema de Refrigeração a Água de uma Tocha de Plasmática Indutiva Utilizando Lógica Fuzzy, Dissertação de mestrado**, Universidade Federal do Rio Grande do Norte, Natal, RN.
- Maniçoba, G. G. C., Salazar, A. O., Amado, J. A. D., de Oliveira, J. A. B., Dubut, J. P., Lock, A. S. and de Oliveira Quintaes, F. (2012). **Estudo e Simulação de uma Tocha de Plasma Térmico com Acoplamento Indutivo, VII Congresso Nacional de Engenharia Mecânica**, São Luiz, MA.
- Nishio, N. M. and Takabatake, A. (2002). **Produção de Energia a partir dos Resíduos do Reuso da Água, Projeto de graduação**, Departamento de Engenharia de Energia e Automação Elétricas, São Paulo, SP.