

ESTUDO DE DESEMPENHO DE UM CONTROLADOR FUZZY DESCRITO EM VHDL PARA DOIS TIPOS DE SISTEMA DE TANQUES

ALLYSON A. L. FILGUEIRA, MARCELO R. G. B. VALE, LEONARDO A. CASILLO

Laboratório de Sistemas e Infraestrutura de Comunicação, Universidade Federal Rural do Semi-Árido. Rua Francisco Mota, nº 572 – Bairro Pres. Costa e Silva, Mossoró - RN, CEP 59625-900

E-mails: allyson_alf@hotmail.com, marceloguerra@ufersa.edu.br,
casillo@ufersa.edu.br

Abstract— Along with the need to develop controllers for industrial processes, it arises the need to study advanced control techniques, such as fuzzy controllers, that can supply nonlinearities inherent in the systems. The implementation of these fuzzy controllers based on Hardware Description Language (HDL) has the advantage of high speed and fast reprogramming for use in another project, by changing only desired variables. The objective of this study is to develop a fuzzy controller implemented in VHDL (Very High Speed Integrated Circuit Hardware Description Language) for a tank level system and to evaluate its effectiveness, comparing it with a fuzzy controller developed in Matlab®. Its methodology is to develop a fuzzy system described in VHDL and divided into coding, inference and decoding for a Quanser cylindrical tank system and another system with a trapezoidal format tank. The results of this proposal were satisfactory given the limitations presented by the system.

Keywords— Level controlling, Fuzzy logic, VHDL, Matlab®, Altera Quartus®

Resumo — Com a necessidade de desenvolver controladores para processos industriais, surge a necessidade de estudar técnicas de controle avançado, como por exemplo, controladores *fuzzy*, que possam suprir não linearidades inerentes nos sistemas. A implementação destes controladores *fuzzy* baseados em Linguagem de Descrição de *Hardware (HDL)* tem como vantagem a alta velocidade e a reprogramação rápida para utilização em outro projeto, alterando apenas variáveis desejadas. O objetivo do trabalho é desenvolver um controlador *fuzzy* implementado em VHDL (*Very High Speed Integrated Circuit Hardware Description Language*) para um sistema de nível de tanques e avaliar sua eficácia, comparando-o com um controlador *fuzzy* desenvolvido em Matlab®. Sua metodologia consiste em desenvolver um sistema *fuzzy* descrito em VHDL e divididos em codificação, inferência e decodificação para um sistema de tanque cilíndrico da Quanser e outro sistema com tanque de formato trapezoidal. Os resultados desta proposta se mostraram satisfatórios diante de limitações apresentadas pelo sistema.

Palavras-chave— Controle de nível, lógica fuzzy, VHDL, Matlab®, Altera Quartus®

1 Introdução

No contexto industrial, há uma série de problemas de variados aspectos, e um dos mais recorrentes é o de controle de nível em reservatórios ou tanques. Esses sistemas contêm alguns fatores que dificultam o seu controle. Um dos principais fatores é a não linearidade do sistema, que pode prejudicar o desempenho de controladores convencionais.

O controlador baseado na lógica fuzzy é um dos tipos de controle utilizados para a solução deste problema. Ele é baseado num conjunto de regras, que refletem o conhecimento do especialista do projeto. Este controlador é baseado em três etapas: fuzificação (codificação), mecanismo de inferência, e defuzificação (decodificação). Na etapa de codificação, é convertido os valores de entrada em valores fuzzy; a etapa de inferência avalia as entradas usando o conjunto de regras; por fim, a decodificação converte os valores fuzzy em valores reais (Zadeh, 1965).

A implementação em VHDL (*Very High Speed Integrated Circuits Hardware Description Language*) do sistema fuzzy é adequado para ser implementado em uma aplicação de um circuito integrado específico e possa ser embarcado em uma plataforma de prototipagem em conjunto com um microprocessador,

como, por exemplo, um *Field Programmable Gate Array (FPGA)*. A principal vantagem do uso da linguagem de descrição de *Hardware* é a rápida prototipagem, e permite o uso de importantes ferramentas de síntese (Vuong et al,2006), como por exemplo, o Altera Quartus II 13.

Neste trabalho, serão utilizados dois sistemas de tanque distintos, um de formato cilíndrico, mais convencional, e outro com formato de trapézio, desenvolvido em escala laboratorial para averiguar a capacidade de adaptação dos controladores (tanto o fuzzy quanto o fuzzy descrito em VHDL) com relação a alteração da área da seção transversal do reservatório.

O trabalho está dividido em descrição do sistema de nível, onde será abordado todo o modelo matemático dos dois tipos de tanques para sua modelagem no Matlab®; Projeto do controle fuzzy para estes sistemas, de forma a abordar aspectos relacionado a lógica fuzzy; a implementação do controlador fuzzy descrito em VHDL, com uma abordagem nas etapas da programação; nos resultados encontra-se as principais discussões sobre a resposta do sistema; por último, segue a conclusão onde foi feita uma síntese geral sobre todo o trabalho.

2 Descrição e modelagem do sistema de nível

Esta seção aborda todo o processo de desenvolvimento do controlador fuzzy, e como implementá-lo em linguagem de descrição de *hardware*, começando pela descrição e modelagem do sistema, quais os sensores e atuadores envolvidos no processo, e explicando como a planta trabalha nos ambientes de simulação Matlab®/Simulink e no Modelsim/Altera Quartus II. O processo de obtenção das variáveis foi baseado no trabalho de Vieira (2017), assim como o controlador fuzzy mostrado nas subseções que seguem.

2.1 Descrição dos sistemas de tanques

Em cada um dos dois tanques, o líquido é retirado do fundo através de um orifício de saída, sendo a pressão de saída, a atmosférica. Ambas as inserções de saída são configuráveis e podem ser ajustadas alterando os orifícios roscados na parte inferior de cada tanque, aumentando ou diminuindo a vazão destes. O nível da água em cada tanque é medido usando um sensor sensível à pressão, localizado na parte inferior do tanque.

O primeiro tanque, de formato cilíndrico e linear foi obtido de uma bancada didática. Este contém um orifício responsável pela vazão de saída do sistema, e dentro deste está inserido o sensor de pressão. O modelo de tanque cilíndrico pode ser visto na Figura 1, e seu respectivo modelo em Matlab® pode ser visto na Figura 2.

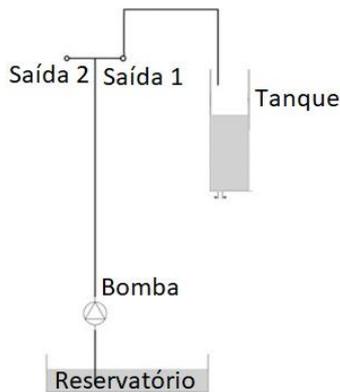


Figura 1. Configuração do modelo de tanque cilíndrico

Sua modelagem pode ser desenvolvida tomando por base a vazão de entrada (F_{in}), que é dado pela relação entre a tensão da bomba (V_p) e a constante da bomba (K_m). A Equação 1 relaciona a vazão de entrada com as variáveis mencionadas anteriormente:

$$F_{in} = K_m * V_p \left[\frac{cm^3}{s} \right] \quad (1)$$

$$V_{out} = \sqrt{2gL} \left[\frac{cm}{s} \right] \quad (2)$$

Têm-se também a equação de saída de água do sistema, regida pela equação de Bernoulli para escoamento da água (Equação 2), onde g é a gravidade (cm/s^2) e L é a altura do volume de água, neste caso, em centímetros.

Tomando por base as duas equações anteriormente citadas, e relacionando vazão de saída do sistema com a área do orifício de saída, a equação que rege a velocidade de escoamento de saída do sistema é determinado pela equação 3.

$$F_{out} = aV_{out} = a\sqrt{2gL} \left[\frac{cm^3}{s} \right] \quad (3)$$

Com o intuito de se obter a variação do nível dos tanques (L') uma das variáveis de controle do sistema, Vieira (2017) comenta que é necessário definir a variação volumétrica (V'), dada pela diferença entre as vazões de entrada e saída do tanque, e por fim, resulta-se a equação 4 para a variação do nível:

$$L' = \frac{K_m}{A} * V_p - \frac{a}{A} \sqrt{2g} \sqrt{L} \left[\frac{cm^3}{s} \right] \quad (4)$$

Onde L' é a variação do nível do tanque, em centímetros; " A " é área da seção transversal do tanque, em centímetros.

Para o tanque de modelo proposto, com geometria trapezoidal, a equação que rege seu comportamento é baseada na altura da coluna d'água e pode ser vista na equação 5, onde:

$$A = 50 + 0.833 * Altura \quad (5)$$

Esta equação foi obtida de acordo com as dimensões do tanque, que pode ser visto na Figura 3.

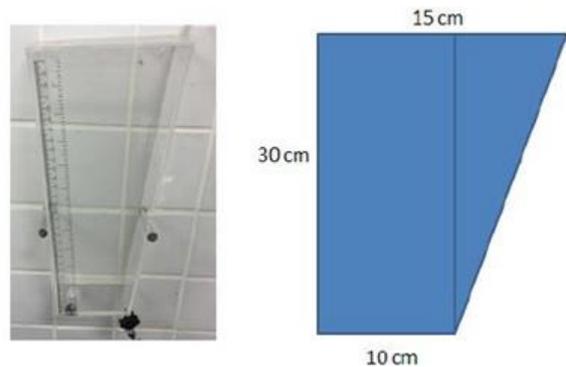


Figura 3. Formato do tanque trapezoidal e suas dimensões

Dividindo a área total do tanque em duas partes, forma-se um retângulo e um triângulo. A inclinação (coeficiente angular) deste triângulo foi obtida com base no ângulo da tangente dos catetos, e assim, com o cálculo desta razão, há um aumento de $0,833 \text{ cm}^2$ à medida que a altura do tanque se modificava, sendo a área da seção transversal a única variável adicional com relação a modelagem do tanque de formato cilíndrico.

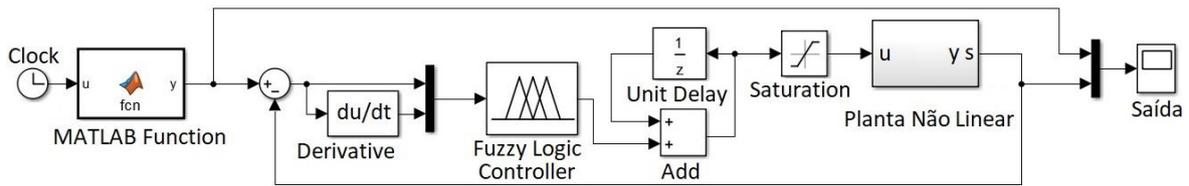


Figura 2. Modelo de tanque cilíndrico no Matlab®

3 Projeto do controle fuzzy para nível de água

O controlador convencional fuzzy é usado no sistema de controle de nível de água do sistema de tanques acoplados. As variáveis de entrada denotam o erro (desvio do valor desejado) e o valor da variação do erro.

$$e(k) = y_r(k) - y(k) \quad (6)$$

$$de(k) = e(k) - e(k-1) \quad (7)$$

A equação 6 expressa o desvio do nível de água, onde $y_r(k)$ significa o nível de água dado. Quando $y(k)$ é maior que $y_r(k)$, o nível de água atual é maior do que o nível de água dado, então o desvio é negativo. Quanto maior o atual nível de água, maior o desvio negativo; Em vez disso, quando $y(k)$ é menor que $y_r(k)$, o nível de água atual é menor que o nível de água dado, o desvio é positivo. Quanto menor o nível de água atual, maior é o desvio positivo.

A equação 7 expressa a variação no desvio do nível de água. Se o nível no ponto k é maior que o valor no ponto $k-1$, significa que o fluxo de água aumentou. Em outras palavras, o nível de água é maior que antes e o desvio é negativo. A saída do controlador fuzzy é igual à variação da tensão na bomba $u(k)$ no qual afeta diretamente a mudança do nível de água $y(k)$. A variável de saída depende da mudança do valor de tensão na bomba: quanto maior o valor da grandeza no atuador, maior a injeção do fluxo de água e vice-versa.

3.1 Codificação das variáveis de entrada/saída:

Durante o controle fuzzy, sobre os parâmetros de entrada/saída, devem ser expressos como um subconjunto fuzzy dos parâmetros de linguagem. Suponha o conjunto fuzzy *Erro* que expressa o desvio de entrada, ou seja:

$$Erro = \{ENG, ENP, EZ, EPP, EPG\}$$

Onde este conjunto, representa, respectivamente: Erro Negativo Grande, Negativo Pequeno, Zero, Positivo Pequeno e Positivo Grande.

Supondo o conjunto fuzzy da linguagem de parâmetros, *Derro* expressa a mudança do desvio de entrada, ou seja:

$$Derro = \{DENG, DENP, DEZ, DEPP, DEPG\}$$

Onde este conjunto, representa, respectivamente, as Derivadas do Erro: Negativo Grande, Negativo Pequeno, Zero, Positivo Pequeno e Positivo Grande. Já o conjunto de saída é dado pela representação “Fo” é tem o seguinte conjunto:

$$Fo = \{NG, NP, Z, PP, PG\}$$

Onde as siglas do conjunto, representam, respectivamente, Negativo Grande, Negativo Pequeno, Zero, Positivo Pequeno e Positivo Grande.

3.2 Determinando a linguística das funções de pertinência e linguagem das regras:

A escolha das funções de pertinência das variáveis de entrada/saída como funções trigonométricas e sua relação com o resultado final é feita através de um conjunto de regras. Nas Figuras 4, 5 e 6, podem ser vistas as funções de entradas e saída do sistema, que respectivamente, representam o erro, a derivada do erro, e a saída Fo. No conjunto de funções da entrada Erro, os valores variam aproximadamente de 7 a -7 centímetros. Para valores além deste limite, foi designado um valor de “saturação, onde será considerado Erro Negativo Grande todo erro menor que o -7cm, e todo erro maior que sete centímetros será considerado Erro Positivo Grande. De maneira análoga, esta regra se aplica as outras duas funções, sendo a entrada Derro e a saída Fo.

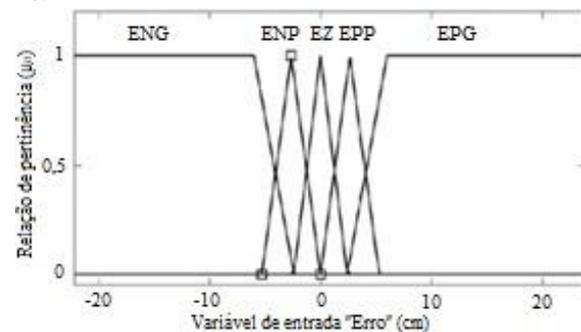


Figura 4. Funções de pertinência para variável “Erro”

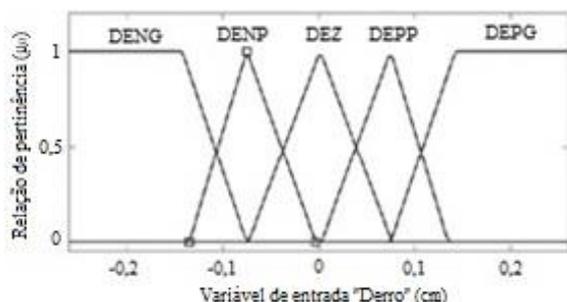


Figura 5. Funções de pertinência para variável "Derro"

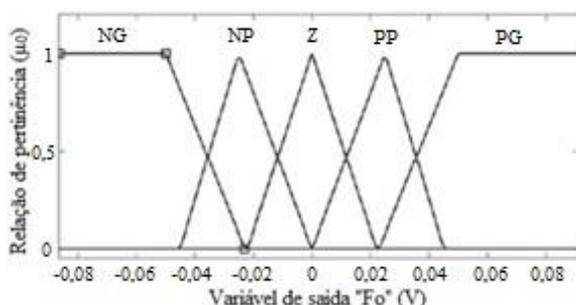


Figura 6. Funções de pertinência para a variável de saída Fo

As regras para a tomada de decisão do controlador dependem do estado das variáveis de entrada (erro e derivada do erro), então a saída do sistema será de acordo com as uniões e interseções dos valores provenientes destas variáveis de entrada. A Tabela 1 mostra o conjunto de regras para a tomada de decisão do controle, onde a primeira coluna desta tabela representa a derivada do erro, e a primeira linha representa o erro. O cruzamento da respectiva linha com a coluna das variáveis de entradas resulta na variável de saída correspondente.

Tabela 1. Conjunto de regras fuzzy

Erro(cm)					
Derro(cm)	ENG	ENP	EZ	EPP	EPG
Fo(V)					
DENG	NG	NG	NG	PP	PP
DENP	NG	NP	NP	PP	PP
DEZ	NG	NG	Z	PP	PP
DEPP	PP	Z	PP	Z	PP
DEPG	Z	PG	PP	Z	PP

3.3 Implementação fuzzy em VHDL

Na etapa de codificação, as variáveis Erro e Derivada do erro são convertidas em valores fuzzy. A declaração desta etapa em entidade em VHDL deve ser declarada – além de um sinal de *clock* – para fins de simplificação com valores inteiros, pois o fator conversão para um sinal digital é mais fácil, e uma possível implementação em um FPGA de ponto flutuante (*floating point*) pode ser inviável por causa do seu valor aquisitivo de mercado e sua complexidade de obtenção. A Figura 7 mostra a

declaração das variáveis de entrada como entidade de maior nível (*Top-Level Entity*).

```

2 entity cosimprojeto is
3 port (erro, derro : in integer;
4       clk: in bit;
5       fo : out integer);
6 end cosimprojeto;
```

Figura 7. Declaração das variáveis em VHDL

As funções de pertinência são escritas como equações de reta, obedecendo aos intervalos de cada função, ou aproximando ao valor inteiro superior mais próximo, caso o valor seja decimal. Para este trabalho, foi adotado um ganho no valor da derivada do erro e da saída de 100 e de 1000, respectivamente, para fins de se obter um valor inteiro desejado, além de a saída ser de um ganho percentual, variando de 0 a 100%. Uma parte do código em VHDL que representa a equação da função de pertinência erro zero (EZ) pode ser encontrada na Figura 8.

```

20 --EZ erro zero
21 ez <= ((42*erro) + 100) when
22 (erro >-2 and erro < 0) else
23 ((-42*erro) + 100) when
24 (erro >0 and erro <2)
25 else 0;
```

Figura 8. Equação da reta da função de pertinência Erro Zero (EZ)

O mecanismo de inferência usado é o método de min-max de Mamdani para obter a saída fuzzy. Cada conjunto de regra, corresponde a um operador de mínimo entre as condições, por exemplo, na regra:

“Se erro for **EZ** e derro for **DEZ** Então Fo é **Z**”

Esta regra pode ser representada pelo conjunto $C = \min[EZ, DEZ]$. Todas as regras estão agrupadas pelo operador de máximo (D_n) para se obter as classes do sistema fuzzy, então os conjuntos D1, D2, D3, D4, D5 estão representadas pelas saídas NG, NP, Z, PP, PG, respectivamente, onde cada conjunto D está associado ao máximo valor entre os mínimos que envolvem as regras fuzzy daquela saída.

No estágio de decodificação, é feita a conversão dos valores fuzzy em valores reais, através de métodos matemáticos. No caso deste trabalho, foi usado o método do centro de gravidade, e foi comparado com o trabalho de Vieira (2017) que também utilizou este método. O código foi gerado para que a cada evento de transição de borda de subida de *clock*, fosse calculado um novo valor de tensão para a bomba. Parte do processo de decodificação discretizado em VHDL pode ser visto a seguir:

```

process (clk, d1, d2, d3, d4, d5)
variable prod,som,saida: integer;
begin if (clk'event and clk = '1') then
prod:=(-50*d1)+(-25*d2)+ (0*d3) + (25*D4) +
(50*D5);
```

```

som:=(d1+d2+d3+d4+d5); if som=0 then sa-
ida<=0;
else saida:=(prod/som);
end if;
end if;
fo<= saida;
end process;

```

Após o uso do comando *port map*, o Altera Quartus é usado para síntese e implementação do projeto em nível RTL (*Register Transfer level*), empregando primitivas disponíveis na ferramenta de síntese como comparadores, somadores, registradores e portas lógicas. O circuito gerado nessa etapa não está associado a nenhuma tecnologia de fabricação em particular, e não está, necessariamente, otimizado (D'amore, 2005). O esquema do diagrama de blocos gerado pelo RTL viewer é mostrado na Figura 9.

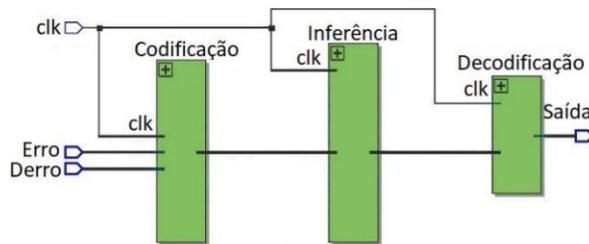


Figura 9. Diagrama lógico fuzzy

Com o FPGA ALTERA Cyclone III EP3C16F484C6 usado como parâmetro de memória, o Software ALTERA Quartus II gera uma tabela de recursos utilizados pelo FPGA escolhido. A tabela com os recursos utilizados para o FPGA DE0 Cyclone III pode ser vista na Tabela 2.

Tabela 2. Recursos utilizados pelo FPGA cyclone III

Recurso	Elementos lógicos
Elementos lógicos totais	7.707/15.408(50%)
Registradores dedicados	32/15.408 (<1%)
Pinos totais	97/347 (28%)
Multiplicadores embutidos	24/112 (21%)

4 Resultados

A análise dos resultados se dá por meio da utilização de um conjunto de valores de referência (primeiro de valor fixo, depois variável) previamente

ajustados e verificar a resposta de como o sistema de controle atua para os controles fuzzy Matlab® e o fuzzy em VHDL.

Os parâmetros serão analisados para os sistemas de tanque de formato cilíndrico e trapezoidal, tanto para um *set-point* fixo quanto para uma série de valores variáveis.

4.1 Resultado do tanque de formato cilíndrico

Iniciando com *Set-point* de 25 centímetros, o gráfico de saída do controle fuzzy Matlab® mostra uma rápida estabilização, precedido de um sobressinal de aproximadamente 3 centímetros, com um erro de regime permanente nulo.

Com relação a resposta do controlador fuzzy em VHDL, ocorreu um pequeno sobressinal, porém de valor menor em relação ao controle fuzzy Matlab®, e erro de regime permanente de aproximadamente 0,4 centímetros. O amortecimento da curva que vai do sobressinal até a estabilização da resposta foi um pouco mais lenta, devido aos valores de saída serem discretos, então os valores de passo são fixos, diferente do controle fuzzy, onde há uma gama maior de valores que podem ser tomados como saída. Na Figura 10, pode-se ver as respostas dos controladores fuzzy VHDL e fuzzy Matlab® em relação a um valor de referência fixo.

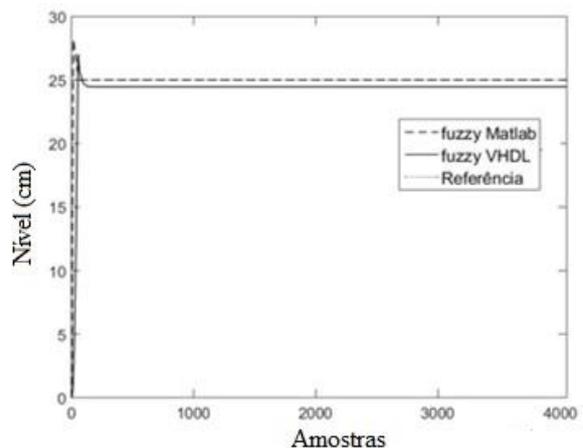


Figura 10. Resposta dos controladores a um *set-point* fixo – tanque cilíndrico

Com a análise dos resultados *set-point* fixo concluída, serão atribuídos valores de *Set-points* variáveis (faixas de 5 a 27 cm) com o intuito de observar a capacidade de adaptação do controlador. As respostas de saída dos sistemas fuzzy Matlab® e fuzzy VHDL podem ser vistas na Figura 11.

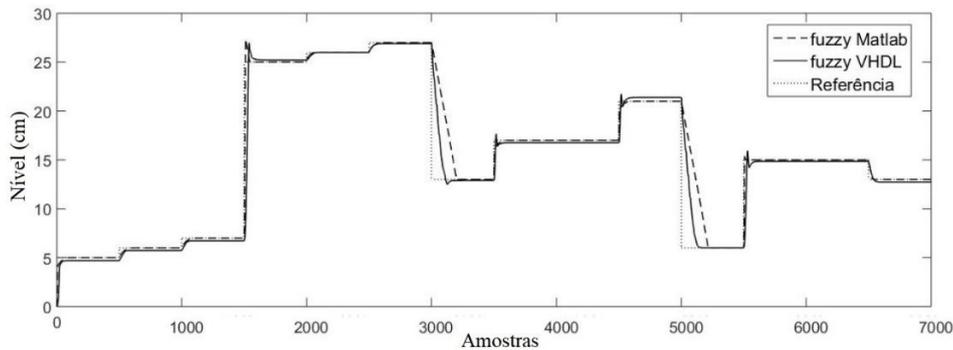


Figura 11. Resposta dos controladores a um *set-point* variável – tanque cilíndrico

Para este caso, o controlador fuzzy Matlab® obteve uma boa resposta, gerando um erro nas variações bruscas de sinal de referência (de 27 para 13 cm, por exemplo), isso se deve às características de sintonização do controlador. Ocorreu um pequeno sobressinal, porém houve uma resposta rápida ao tempo de estabilização.

Em relação ao fuzzy VHDL, houve um sobressinal na maior variação do set-point e um leve atraso para conseguir atingir um sinal de nível menor que o atual entre as amostras entre 3000 e 4000. Houve também mais três pontos de sobressinais nas mudanças de um valor de referência menor para um maior. Como os valores medidos de ambos os controladores foram próximos ao sinal de referência, o erro foi muito próximo de zero e houve sobressinal apenas quando o valor de referência mudou de 7 para 25 centímetros.

4.2 Resultados do tanque de formato trapezoidal

De maneira análoga ao tópico 4.1, será observado a resposta dos controladores estudados (fuzzy Matlab® e o fuzzy VHDL) para o tanque proposto de formato não linear e geometria trapezoidal, onde primeiramente será analisado as respostas do controlador aos *set-point* fixo posteriormente, com valores variáveis.

Em relação a saída do sistema, o comportamento do controlador fuzzy Matlab® foi semelhante ao modelo com tanque linear cilíndrico, com rápida resposta, pequeno erro de regime transitório, erro de regime permanente nulo e com presença de sobressinal de aproximadamente 3 centímetros antes de atingir a estabilidade. Na saída do controlador fuzzy VHDL, sua resposta também se mostrou rápida, com a presença de um sobressinal de aproximadamente 5

centímetros e um erro de regime permanente de aproximadamente 0,8 centímetros. As curvas de resposta podem ser vistas na Figura 12.

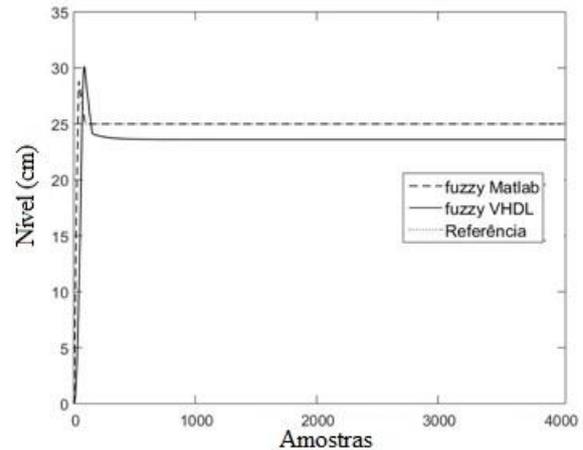


Figura 12. Resposta dos controladores a um *set-point* fixo – tanque trapezoidal

Com os resultados devidamente apresentados para valor de referência fixo, serão mostrados os resultados gerados através do teste de adaptabilidade dos controladores em relação ao sistema de nível, que além do set-point variável, a área do tanque também varia de acordo com o nível da coluna d'água. As saídas dos controladores fuzzy podem ser vistas na Figura 13.

O controlador implementado em VHDL tentou seguir as mudanças impostas pelo sistema, porém com duas variáveis mudando ao longo do tempo (*set-point* e nível baseado na geometria do tanque), e além do emprego de valores inteiros no controlador previamente configurado, não foi possível atingir determinados valores de referência.

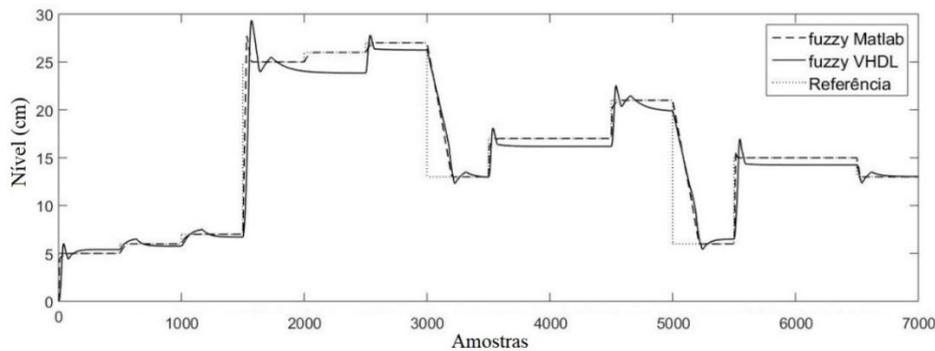


Figura 13. Resposta dos controladores a um *set-point* variável – tanque trapezoidal

O controlador não atingiu o valor de referência em certos pontos do sistema, principalmente quando haviam variações da referência para um valor menor. A resposta do controlador fuzzy Matlab® se mostrou consistente, tentando acompanhar o *set-point*, mas com pequeno atraso, porém de rápida estabilidade. Neste tipo de tanque proposto, o controle não foi eficaz, porém é considerado válido a tentativa, levando em consideração a ação de controle feita como um todo.

4.3 Eficácia do controlador

Para testar a eficácia da implementação do controlador *fuzzy* descrito em VHDL, os valores de saída deste devem ser comparados com valores de saída do controlador *fuzzy* Matlab®. Primeiramente, valores de *erro*, *derivada do erro* e saída (*fo*) foram adquiridos através do *Software* Modelsim/Altera Quartus, onde é feito o processamento de sinal da descrição em VHDL.

Com os valores obtidos no Modelsim/Altera Quartus, deve-se inserir os mesmos valores correspondentes às entradas do sistema (*erro* e *derivada do erro*) no controlador *fuzzy* Matlab® e obter valores de saída. Assim, foi feita a Tabela 3, que mostra os valores de saída de ambos os controladores (*fuzzy* Matlab® e *fuzzy* VHDL) para os mesmos valores de entrada.

Tabela 3. Teste de eficácia dos controladores

Erro	Derro	Quartus	Matlab	Erro(%)
-1	9	-3	-3,26	7,97
4	-42	25	23,1	8,22
1	9	14	13,8	1,45
8	-27	25	23,4	6,83
1	12	15	13,8	8,69
24	-15	25	23,4	6,83
-1	1	-3	-2,5	14,28
-13	26	0	0	0
-10	3	-50	-44,6	12,10
3	-28	25	23,4	6,83

Com estes resultados, pode-se calcular o erro quadrático médio percentual, que foi de aproximadamente 8,3%, valor este que foi abaixo do valor de 10,46% encontrado em Aguilar et al (2014) e acima dos outros valores encontrados na literatura

(Oliveira et al (2010) com 0,84% e Uppalapati (2009) com 0,8%). Este valor percentual calculado teve contribuição principalmente de valores próximos a zero, tanto do erro quanto da derivada do erro, pois exigia de um grau de precisão maior para representá-los no processo de codificação.

5 Conclusão

Com os resultados discutidos na seção anterior, conclui-se que para o tanque cilíndrico utilizando *Set-point* fixo, o controlador descrito em VHDL atingiu a estabilidade com um pequeno erro estacionário. Mesmo com pequenos erros de aproximação e de conversão de sinal, o controlador reagiu satisfatoriamente seu objetivo.

Com valores de referência variando ao longo das amostragens para o tanque cilíndrico, a implementação do sistema *fuzzy* em VHDL foi desenvolvida com sucesso, onde a resposta deste controlador assume características semelhantes às do controlador *fuzzy* Matlab®. Mesmo com erros de aproximação e de conversão de sinal, o controlador reagiu. Em contrapartida, deve-se aprofundar o estudo de estratégias para eliminar o erro proveniente da conversão do sinal para processamento digital (problema do tamanho finito da palavra).

No geral, houve uma perda de sensibilidade da parte do controlador, ocasionando um erro de regime permanente em sua saída. Em relação a eficácia do controlador, que obteve 8,3% de erro quadrático médio, pode-se afirmar que este valor percentual é gerado pelo uso de variáveis inteiras na entrada do controlador *fuzzy* VHDL.

Por fim, os resultados comprovam a eficiência prática do modelo que pode ser utilizado em um sistema real.

Agradecimentos

A CAPES/CNPq pelo apoio financeiro a pesquisa, e à UFRSA, pelo suporte no espaço físico.

Referências Bibliográficas

- Aguilar, A. et al. Efficient design and implementation of a multivariate Takagi-Sugeno *fuzzy* controller on an FPGA. In: Ieee Internacional Conference On Mechatronics, Eletronics And Automotive Engineering (ICMEAE), 2014, Cuernavaca, México, p. 152 – 157.
- D'amore, R. VHDL: descrição e síntese de circuitos digitais. Rio de Janeiro: LTC, 2005.
- Oliveira, D. et al. Design and implementation of a mamdani fuzzy inference system on an FPGA using VHDL. In: Ieee Annual Meeting Of The North American, 2010, Toronto, CA. Fuzzy Information Processing Society (NAFIPS).p. 1-6.
- Uppalapati, S.; Kaur, D. Design and implementation of a mamdani *fuzzy* inference system on an FPGA. In: North American *Fuzzy* Information Processing Society Annual Conference (NAFIPS), 2009. Ohio, Estados Unidos. P.1 – 6.
- Vieira, F. B.; Estudo comparativo entre controladores fuzzy e PI para um sistema de tanque. 2017. Dissertação (mestrado em automação e controle). Universidade Federal Rural do Semi-Árido. Mossoró (RN). 2017.83f.
- Vuong, P. T.; Madni, A. M.; Vuong, J. B.; VHDL implementation for a fuzzy logic controller.(2006). IEEE World Automation Congress (WAC), 24-26 de Julho. Budapest, Hungria.
- Zadeh, L. A., "Fuzzy sets". Information and control,vol.8, pp.338-353, 1965.