Cainã Sagaz Correia

CONTROLE FUZZY DE UM MOTOR DC UTILIZANDO SISTEMA DE AQUISIÇÃO DE DADOS E O MATLAB

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia Elétrica da Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Elétrica.

Orientadora:Profa. Dra. Paloma M. S. Rocha Rizol

Guaratinguetá

2014

C824c

Correia, Cainã Sagaz

Controle fuzzy de um motor dc utilizando sistema de aquisição de dados e o matlab / Cainã Sagaz Correia – Guaratinguetá : [s.n], 2014.

42 f : il.

Bibliografia: f. 41

Trabalho de Graduação em Engenharia Elétrica – Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2014.

Orientador: Profª Drª Paloma M. S. Rocha Rizol

1. Motores elétricos de corrente continua 2. Lógica difusa I. Título

CDU 621.313.291

unesp UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
CAMPUS DE GUARATINGUETÁ

CONTROLE FUZZY DE UM MOTOR DC UTILIZANDO SISTEMA DE AQUISIÇÃO DE DADOS E O MATLAB

CAINÃ SAGAZ CORREIA

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO PARTE DO REQUISITO PARA A OBTENÇÃO DO DIPLOMA DE **GRADUADO EM ENGENHARIA ELÉTRICA**

APROVADO EM SUA FORMA FINAL PELO CONSELHO DE CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Prof. Dr. Leonardo Mesquita
Coordenador

BANCA EXAMINADORA:

Profa. Dra. Paloma Maria Silva Rocha Rizol
Orientadora/UNESP-FEG

Prof. Dr. Leonardo Mesquita
UNESP-FEG

Prof. Dr. Francisco Antonio Lotufo
UNESP/FEG

de modo especial, à minha mãe Soraya, ao meu irmão Kauê e minha namorada Janaína, por todo apoio.

AGRADECIMENTOS

Primeiramente gostaria de agradecer a minha família pelo suporte dado durante todos esses anos,

à minha namorada, que me ajudou muito em todos esses anos de faculdade,

e aos meus amigos, por todas as experiências vividas.

à minha orientadora, *Profª. Dra. Paloma Maria Silva Rocha Rizol* pela oportunidade de poder realizar esse trabalho, e por toda ajuda dada,

aos professores que tive durante o curso.

CORREIA, C. S. **Controle *fuzzy* de um motor DC utilizando sistema de aquisição de dados e o Matlab.** 2014. 42f. Trabalho de Graduação (Graduação em Engenharia Elétrica) – Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2014.

RESUMO

O presente trabalho trata do desenvolvimento de um sistema de inferência *fuzzy* para controle da velocidade de rotação de um motor de corrente contínua disponível no *Kit* da *Degem*. Para tanto, deve-se utilizar o *toolbox fuzzy* do *software Matlab* em conjunto com a placa de aquisição de dados NI – USB – 6009 da *National Instruments*. Primeiramente é apresentada a introdução sobre lógica *fuzzy*, o modelo matemático do motor DC e o funcionamento da placa de aquisição de dados. Após isto, fez-se a implementação do controlador *fuzzy* utilizando o *toolbox fuzzy* e o *simulink* do *Matlab*, em conjunto com a placa de aquisição de dados para controle do motor DC, que é descrito detalhadamente. Por fim, mostra-se a montagem realizada e apresentam-se os resultados obtidos pelo simulador.

PALAVRAS CHAVE: Motor DC, Controlador *Fuzzy*, *Simulink*, Lógica *Fuzzy*, Sistema de Aquisição de Dados.

CORREIA, C. S. **Fuzzy control of a DC motor using a data acquisition system and Matlab.** 2014. 42f. Graduate Work (Graduate in Electrical Engineering) – Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2014.

ABSTRACT

The present work develops a fuzzy inference system to control the rotation speed of a DC motor available in Degem Kit. Therefore, it should use the fuzzy toolbox of Matlab in conjunction with the data acquisition board NI – USB – 6009, a National Instrument's board. An introduction to fuzzy logic, the mathematical model of a DC motor and the operation of data acquisition board is presented first. Followed by the controller fuzzy model implemented using Simulink which is described in detail. Finally, the prototype is shown and the simulator results are presented.

Keywords:DC Motor, Fuzzy Controller, Simulink, Fuzzy Logic, Data Acquisition System.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

1.1 CONSIDERAÇÕES GERAIS

Nos últimos anos, os *softwares* para aquisição e análise de sinais têm evoluído significantemente. Por meio dessas ferramentas, um computador em conjunto com uma placa de aquisição de dados, pode-se adquirir um sinal, fazer sua análise e tratamento, e portanto é possível fazer o controle de sistemas de diversas áreas de aplicação.

A lógica *Fuzzy* também tem evoluído muito em suas aplicações e é cada vez mais utilizada, mostrando ser uma técnica de controle altamente eficiente e de fácil aplicação.

Este trabalho visa mostrar uma aplicação de controle de motor DC, utilizando a lógica *fuzzy* por meio de uma placa de aquisição de dados.

1.2 OBJETIVOS

Este trabalho tem por objetivo realizar o estudo do sistema de inferência *fuzzy* para controle da velocidade de rotação de um motor de corrente contínua. Para tanto, deve-se utilizar o *toolbox fuzzy* do *software* Matlab em conjunto com a placa de aquisição de dados NI – USB – 6009 da National Instruments.

1.3 MOTIVAÇÃO

O principal fator que motiva a execução deste trabalho é a oportunidade de verificação do desempenho de um controlador *Fuzzy* em uma planta real do *Kit* da *degem* para controle de velocidade do motor de corrente contínua e também fazer a integração entre o *toolbox fuzzy* do Matlab com o *Kit* para o controle de um motor de corrente contínua, por meio de um sistema de aquisição de dados.

1.4 PLANO DE TRABALHO

Nessa seção é apresentado uma visão geral do trabalho como um todo, apresentando uma breve descrição dos temas abordados em cada capitulo.

No capitulo dois é apresentada uma breve introdução à lógica *fuzzy*, as equações e o modelo matemático do motor de corrente contínua e o funcionamento da placa de aquisição de dados.

O capitulo três consiste no desenvolvimento do sistema de controle no *simulink* descrevendo em detalhes os seus subsistemas e a montagem realizada no motor.

No capitulo quatro é apresentado os resultados referentes as simulações feitas usando o modelo do controlador desenvolvido no capítulo três.

No capitulo cinco discute-se sobre os resultados do capitulo quatro, concluindo como o motor responde ao modelo implementado, como a placa de aquisição de dados se comporta e também se o controle *fuzzy* é válido para a aplicação.

2 EMBASAMENTO TEÓRICO

2.1 LÓGICA *FUZZY*

A lógica *fuzzy* foi proposta por Lotfi A. Zadeh em 1965. Esta ideia surgiu como um aprimoramento da abordagem binária clássica do tema, onde respostas do tipo "verdadeiro" (1) ou "falso" (0) não eram adequadas para situações reais, admitindo um valor intermediário "talvez" (0,5).

A teoria dos conjuntos *fuzzy* utiliza a descrição linguística, por exemplo, avaliando a temperatura como *quente, morno, frio,* dentre outras. E para isso utiliza o conhecimento de especialistas, conhecedores do sistema estudado para orientarem como o controle deve ser feito.

A aceitação da teoria dos conjuntos *fuzzy* pela comunidade científica foi lenta. Sendo a primeira aplicação da lógica *fuzzy* em um sistema de controle foi realizada por Mamdani em 1974, sendo que Mamdani aplicou a lógica *fuzzy* no controle de uma caldeira (RIZOL, 2008).

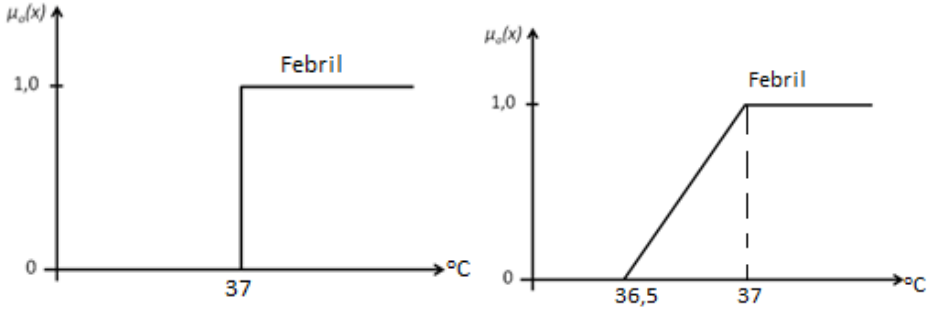
2.1.1 Conjunto *Fuzzy*

Em um conjunto clássico os valores assumidos são bem definidos, sendo que um elemento pertencerá ou não pertencerá a um conjunto. Então, a pertinência é limitada por dois valores (1/0, Verdadeiro/Falso).

Para um conjunto *fuzzy*, um elemento possui variados graus de pertinência. Como por exemplo, na teoria clássica uma pessoa é considerada febril quando sua temperatura medida é maior que 37ºC, então se alguém está com 36,9 ºC ela não está com febre.

Se o mesmo exemplo fosse usado para um conjunto *fuzzy,* as incertezas para o que seria estado febril ficariam representado por um grau de pertinência, como mostrado na Figura 1.

Figura 1 – Funções de pertinência booleana (esquerda) ou clássica *fuzzy* (direita)



Fonte: (SANTIAGO, 2008) modificado pelo autor

Portanto, a teoria de conjuntos *fuzzy* busca traduzir em termos formais a informação imprecisa que ocorre de forma natural nos fenômenos da natureza.

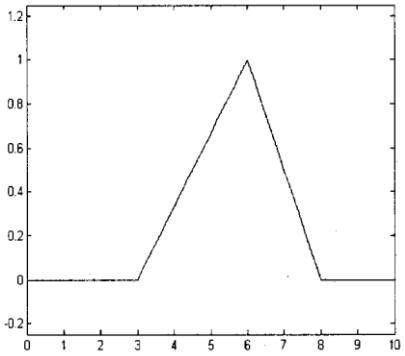
2.1.2 Funções de Pertinência

Funções de pertinência são funções que mapeiam um conjunto clássico de elementos em seus valores correspondentes em um conjunto *fuzzy*. São curvas que definem qual o grau de pertinência de cada um dos elementos de um conjunto de entrada, com valores que variam entre 0 e 1 (OMIDEH, 2003). É importante ressaltar que os graus de pertinência não são definidos de maneira absoluta, mas são dependentes do contexto (por exemplo, 1,75m pode ser considerado uma pessoa alta aqui no Brasil, mas não é na Alemanha).

Existem vários tipos de funções de pertinência, sendo que o tipo a ser utilizado depende da aplicação. As mais comuns são:

a) Triangular: definida pelos parâmetros {a,b,c} como mostrado na Figura 2.

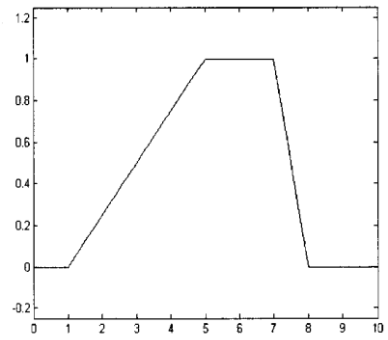
Figura 2 – Função de pertinência do tipo triangular, triângulo (x; 3,6,8).



Fonte: (OMIDEH, 2003).

b) Trapezoidal: definido pelos parâmetros {a,b,c,d} como mostrado na Figura 3. O grau de pertinência é um valor que pode variar de 0 a 1, que define o quanto um elemento pertence ao conjunto *fuzzy*.

Figura 3 – Função de pertinência do tipo trapezoidal, trapézio (x; 1,5,7,8).



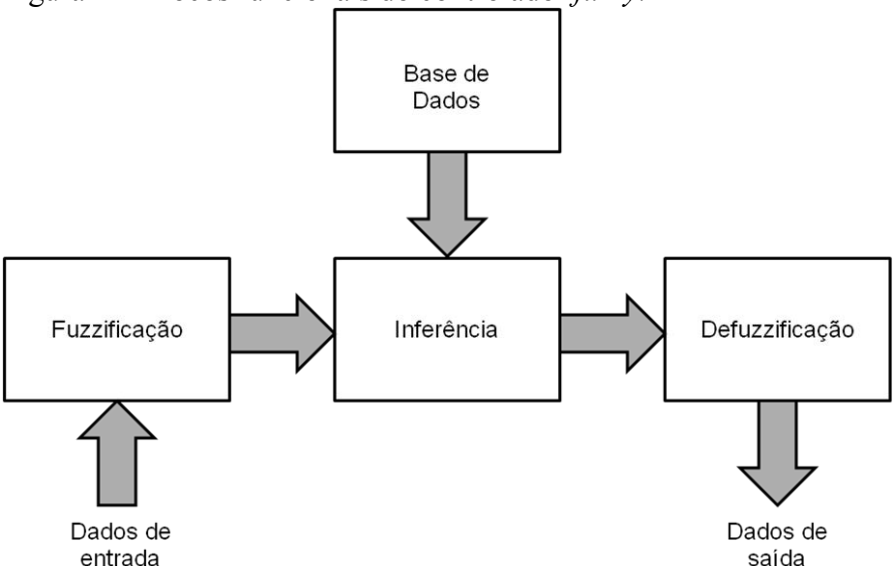
Fonte: (OMIDEH, 2003).

Outras funções de pertinência utilizadas são: Gaussiana, Pi, Função S, Função Z, entre outras.

2.1.3 Controlador *Fuzzy*

Um controlador *fuzzy* é composto por quatro blocos, sendo estes a fuzzificação, inferência, base de dados e defuzzificação.

Figura 4 – Blocos funcionais do controlador *fuzzy*.



Fonte: (LAC, 2008)

A seguir, será apresentado a descrição de cada bloco functional do controlador *fuzzy*.

2.1.3.1 Fuzzificação

É na fuzzificação onde os dados de entrada ("*crisp*") são transformados de domínio real para domínio *fuzzy*, ou seja, em variáveis lingüísticas, com seus respectivos graus de pertinências (JANE, 2004).

2.1.3.2 Base de regras

A base de regras ou de dados é um conjunto de regras do tipo *se-então*. Contém a quantificação, na forma de lógica *fuzzy,* do como conseguir um bom controle (PASSINO; YURKOVICH, 1998).

As regras são feitas utilizando-se descrição linguística do sistema a ser controlado. Essa descrição pode ser geralmente dividida em algumas partes. Haverá variáveis linguísticas para cada variável de entrada e de saída do controlador *fuzzy*. Por exemplo, *erro* pode descrever a variável *e(t)*, *variação do erro* pode descrever $\dot{e}(t)$, *força* pode descrever *u(t)*, etc (PASSINO; YURKOVICH, 1998). Com o passar do tempo, essas variáveis linguísticas irão assumir *valores linguísticos*, do tipo: *negativo de grande intensidade*, *nulo*, *positivo de pequena intensidade*, etc (PASSINO; YURKOVICH, 1998). Assim, as regras poderão ser da seguinte forma: se *erronegativo de grande intensidade* e *variação do erro* nula, então *força* positiva de pequena intensidade (PASSINO; YURKOVICH, 1998).

Cada regra da base é uma quantificação *fuzzy* das variáveis linguísticas realizada pelas funções de pertinência.

2.1.3.3 Máquina de inferência

A máquina de inferência possui basicamente duas tarefas: a primeira, determinar qual o grau de relevância de uma determinada regra no momento atual, com base nos valores das entradas; a segunda, tirar conclusões baseadas nas variáveis de entrada e na base de regras (PASSINO; YURKOVICH, 1998).

A seguir são descritos dois sistemas de inferência *fuzzy*.

- Sistema de inferência proposto por Mamdani

É o método mais utilizado, tanto em aplicações como na literatura, sendo composto por regras do tipo: "*se* x_1 é A_1 e x_2 é A_2 então y é igual a B", onde x_i corresponde a um

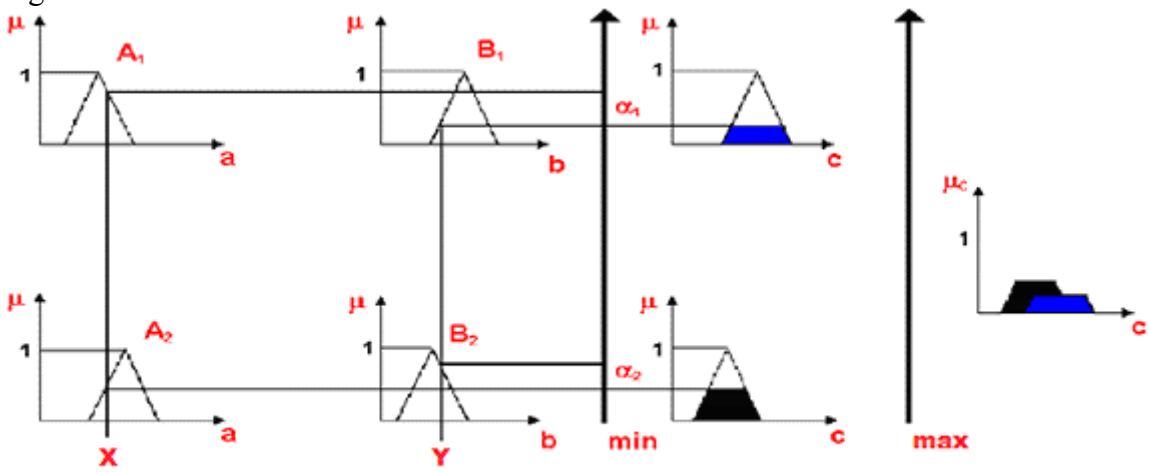
valor de entrada, A_i corresponde a um conjunto *fuzzy* que representa um antecedente da regra, y corresponde à saída do controlador e B corresponde um conjunto *fuzzy* que representa o consequente da regra. (ROSS, 2004).

Para calcular o valor da saída $\mu_B(y)$, é feita a agregação dos graus de pertinência resultante do consequente da regra.

$$\mu_B(y) = \max [\min[\mu_{A1}(x_1), \mu_{A2}(x_2)]] \quad (1)$$

Usando uma representação gráfica, quando duas regras são possíveis, a saída será determinada da seguinte forma:

Figura 5 – Sistema de inferência de Mamdani.



Fonte: (SANTIAGO, 2008)

- Sistema de inferência *fuzzy* proposto por Takagi-Sugeno

Neste modelo, as regras SE/ENTÃO são da seguinte forma (LAC, 2008):

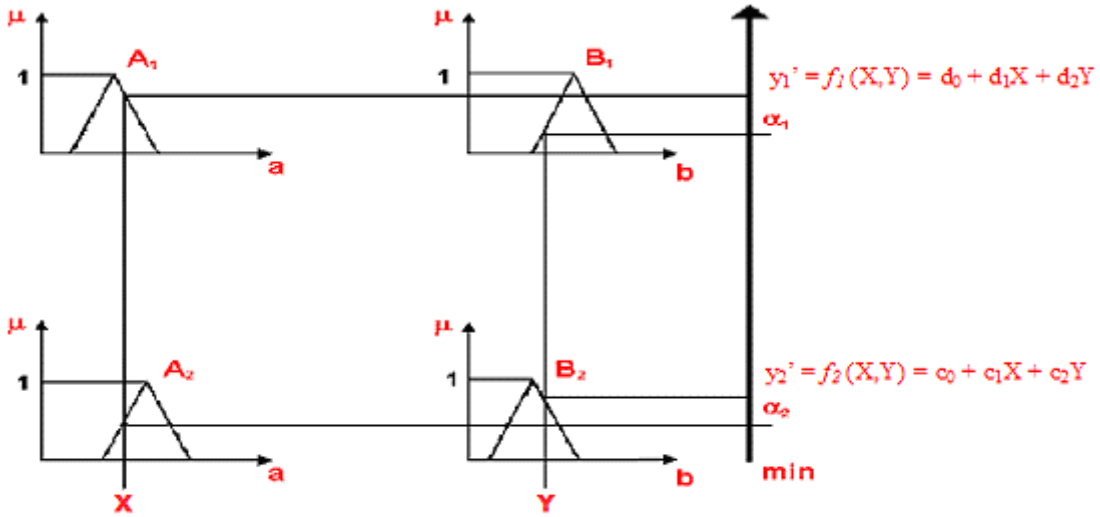
$$\textbf{SE } x_1 \text{ é } F_1' \, e \ldots e \, x_n \text{ é } F_n' \textbf{ ENTÃO } y' = c_0' + c_1' x_1 + \ldots + c_n' x_n$$

Onde F_i' são os conjuntos *fuzzy*, y' é a função de saída e c_n' são os parâmetros estimados.

Uma vez que com o método de Takagi-Sugeno cada regra tem uma saída, dada por uma função, a saída total é obtida por meio da média ponderada, o que torna este método mais rápido em relação ao de Mamdani. (ROSS, 2004).

A Figura 6 ilustra o funcionamento do sistema de inferência Takagi-Sugeno

Figura 6 – Sistema de inferência de Takagi-Sugeno (Modelo de Interpolação).



Fonte: (SANTIAGO, 2008)

2.1.3.4 Defuzzificação

A etapa de defuzzificação é responsável por converter os dados *fuzzy* que vem da máquina de inferência em dados reais (crisp) que podem ser usados numa ação de controle.

Existem vários métodos de defuzzificação. Neste trabalho serão apresentados o Método Centro-da-Área e o Método da Média Ponderada.

- Método Centro-da-Área

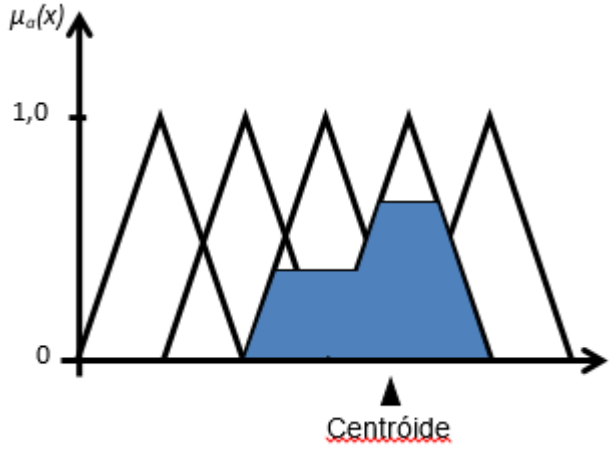
Este é o método de *defuzzificação* mais conhecido que consiste em calcular o centroide das saídas *fuzzy* resultantes.

O cálculo do centro de gravidade é dado por:

$$Centróide = \frac{\sum_{i=1}^{N} \omega_i y_i}{\sum_{i=1}^{N} \omega_i} \qquad (2)$$

Onde N é o número de regras ativada, ω_i é a área de uma função de pertinência e y_i é a posição do centróide de cada função de pertinência.

Figura 7 – Exemplo do método de defuzzificação centro-da-área.



Fonte: (SANTIAGO, 2008)

- Método Média Ponderada

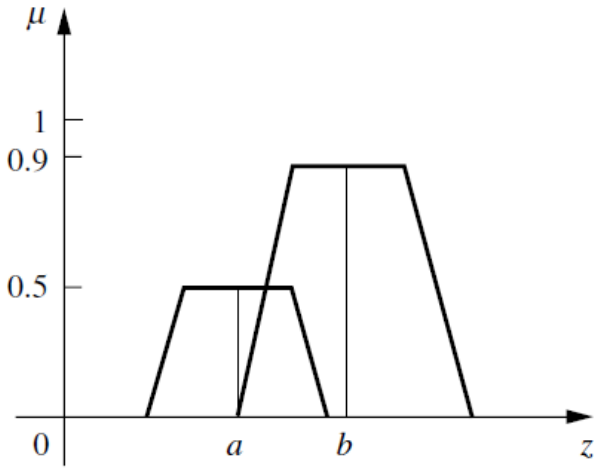
Este é o método de defuzzificação mais utilizado em aplicações fuzzy, uma vez que é o método com maior eficiência computacional (ROSS, 2004). Seja $\bar{z}$ o centroide de cada função de pertinência. A saída z* é calculada da seguinte forma.

$$z^* = \frac{\sum \mu_{B_i}(\bar{z}).\bar{z}_i}{\sum \mu_{B_i}(\bar{z})} \quad (3)$$

Sendo que $\mu_{Bi}(\bar{z})$ é o grau de pertinência de cada função de pertinência da parte consequente da base de regras.

A Figura 8 apresenta como funciona esse método:

Figura 8 – Método de defuzzificação média ponderada.



Fonte: (ROSS, 2004)

2.2 MOTOR DC DE IMÃ PERMANENTE

Para este trabalho foi utilizado o *kit* de laboratório PCT-1 da DEGEM *System*, um *kit* educacional para controle de velocidade de um motor de corrente contínua.

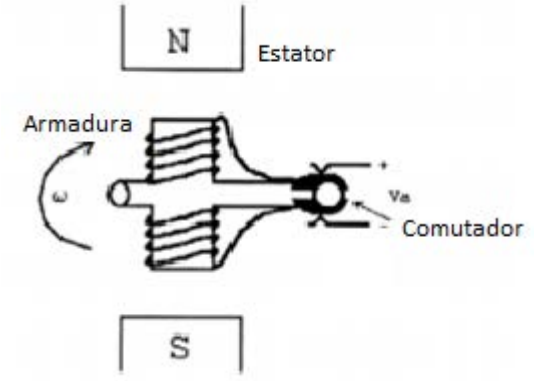
O motor DC presente nesse *kit* é de imã permanente controlado pela armadura, que é um mecanismo que converte energia elétrica em energia mecânica por meio de um acoplamento magnético. A energia elétrica é fornecida por uma fonte de tensão, enquanto a força mecânica é fornecida por um rotor girante.

Um motor DC básico constitui-se de dois componentes principais: o rotor ou armadura e o estator. A armadura gira dentro da estrutura do estator estacionário.

O estator consiste de imãs permanentes que criam um campo magnético, enquanto que a armadura consiste de um eletroímã criado por uma bobina enrolada em torno de um núcleo de ferro. A armadura gira devido ao fenômeno de forças de atração e repulsão de dois campos magnéticos. Um campo magnético é gerado pela armadura ao enviar uma corrente elétrica por meio da bobina e a polaridade é constantemente alterada mudando o sentido da corrente através da bobina, fazendo com que a armadura gire.

A Figura 9 mostra um exemplo de um motor de corrente contínua.

Figura 9 – Configuração de um motor de corrente contínua.



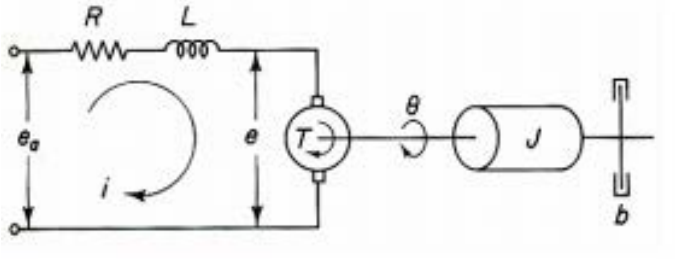
Fonte: (ALTAS, 2007)

2.2.1 Modelo matemático do motor de corrente contínua

Para representar por meio de um esquema o motor, deve-se levar em consideração a tensão de entrada e a bobina. A bobina terá uma resistência (R), uma indutância (L) e uma forca contra-eletromotriz (e) proporcional a velocidade do rotor. O torque (T) é gerado por uma força que depende do fluxo magnético e da corrente (i). Como o fluxo magnético é constante, pode-se relacionar o torque gerado pelo motor como sendo proporcional a corrente.

A Figura 10 mostra a representação esquemática do motor DC com imãs permanentes.

Figura 10 – Esquema de um motor DC de imãs permanentes.



Fonte: (PUCRS, 2014)

Onde:

$e_a(t)$	Tensão de alimentação do motor
i(t)	Corrente de armadura
R	Resistência de armadura
L	Indutância de armadura
e(t)	Força contra-eletromotriz
T(t)	Torque gerado no motor
ɵ(t)	Posição angular do eixo do motor (rotor)
ω(t)	Velocidade do rotor
J	Momento de inércia do rotor
b	Coeficiente de atrito viscoso do rotor

A equação diferencial para o circuito equivalente pode ser obtida pela Lei de Kirchhoff para tensões onde a soma de todas as tensões em torno de um circuito deve ser igual a zero:

$$e_a(t) = Ri(t) + L\frac{di(t)}{dt} + e(t) \quad (4)$$

A equação para a força contra-eletromotriz está mostrada abaixo:

$$e(t) = K_\omega \,.\, \omega(t) \quad (5)$$

A força contra-eletromotriz é proporcional a velocidade do motor, tal proporção é representada por uma constante de velocidade(K_ω) que é uma característica construtiva do motor.

Realizando um balanço de energia no sistema, a soma dos torques do motor deve ser zero. Portanto:

$$T(t) = K_T \,.\, i(t) \tag{6}$$

$$T(t) = J.\frac{d\omega(t)}{dt} + b\omega(t) \tag{7}$$

Nota-se que o torque gerado é proporcional a corrente de armadura, esta proporção é representada por uma constante de torque (K_T) que é uma característica construtiva do motor.

Substituindo a equação (6) em (7), tem-se:

$$J.\frac{d\omega(t)}{dt} + b\omega(t) - K_T.\, i(t) = 0 \tag{8}$$

2.2.2 Diagramas de bloco do motor de corrente contínua

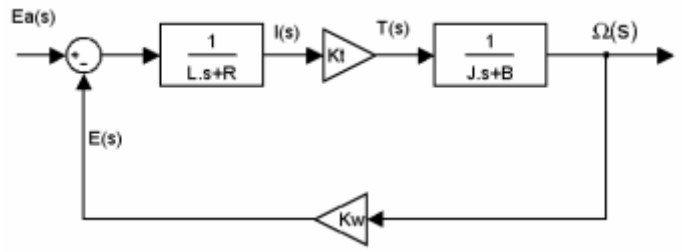
Para fazer o diagrama de blocos, optou-se por passar as equações encontradas no item anterior para o domínio da frequência por meio da Transformada de Laplace. Portanto, tem-se:

$$E_a(s) = RI(s) + L.s.I(s) + K_\omega.\Omega(s) \tag{9}$$

$$J.s.\Omega(s) + b\Omega(s) - K_T.I(s) \tag{10}$$

Com isso, a representação do diagrama de blocos será da seguinte forma:

Figura 11 – Diagrama de blocos de um motor DC.



Fonte: (PUCRS, 2014)

2.3 PLACA DE AQUISIÇÃO DE DADOS

A Placa de Aquisição de dados utilizada para este trabalho foi produzida pela National Instruments denominada NI USB 6009, é um dispositivo de baixo custo para interfaceamento Entrada/Saída de sinais com o computador.

Esta placa pode ser conectada via cabo USB a um PC executando o Matlab, e o Matlab realiza as tarefas de enviar e ler tensões elétricas nos terminais na placa.

A USB 6009 tem como características básicas:

- 8 entradas analógicas (14 bits, 48kS/s);
- 2 saídas analógicas (12 bit, até 150S/s);
-12 E/S digitais, contadores de 32 bits
- As tensões de saída variam entre 0 a 5V e as tensões de entrada entre -10 a 10V

A Figura 12 ilustra a NI USB 6009.

Figura 12- NI USB 6009

Fonte: (NATIONAL INSTRUMENTS, 2014)

Neste trabalho foram utilizadas as entradas e saídas AI0+, AI0- e AO0 analógicas.

3 MONTAGEM E IMPLEMENTAÇÃO DO SISTEMA NO MATLAB

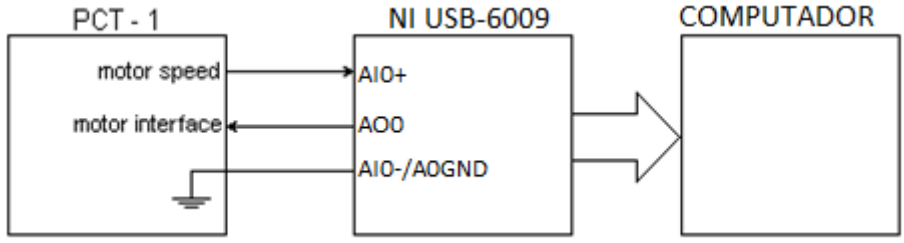
Neste capítulo será mostrado como foi feita a montagem, a implementação e a simulação do sistema no Matlab, utilizando o *Toolbox Fuzzy* e o Simulink. O controlador fuzzy utilizou o método de infêrencia de Mamdani e o método de defuzzificação utilizado foi o centro de área (CoA).

Mas primeiramente é necessário realizar a comunicação entre o Matlab e a placa USB 6009, para isso é recomendado utilizar o Matlab 32 bits. Na primeira vez que utilizar o pacote *Data Acquisition Toolbox*, é necessário executar o Matlab como administrador e executar o comando *daqregister('nidaq')*.

3.1 MONTAGEM REALIZADA

Os equipamentos, anteriormente descritos, foram ligados segundo o esquema mostrado na Figura 13.

Figura 13 – Esquema da montagem realizada



Fonte: (MARIMOTO, 2000, com adaptações do autor)

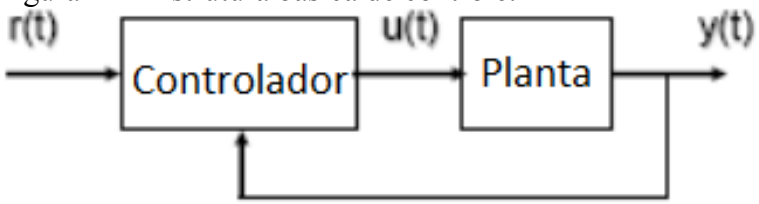
Do motor de corrente contínua, é necessário medir-se o sinal do tacômetro, ou seja, a velocidade de rotação do motor (*motor speed*) e também atuar no mesmo (*motor interface*) de forma a inserir rotação desejada. Na entrada do motor há um amplificador no qual é possível controlar o ganho da tensão de referência em relação a velocidade do motor.

Dessa forma, ligou-se a saída "*motor speed*" à entrada analógica AI0+ da placa de aquisição de dado. A saída analógica AO0 da placa de aquisição de dados foi ligada à entrada "*motor interface*" do PCT –1. Para se completar o circuito, a entrada AI0- e o AOGND foram ligados no terra do PCT-1.

3.2 ESTRUTURA DE CONTROLE

Um sistema básico de controle é mostrado na Figura 14. Neste sistema tem-se que u(t) é a entrada da planta, y(t) é a saída da planta e r(t) é a entrada de referência.

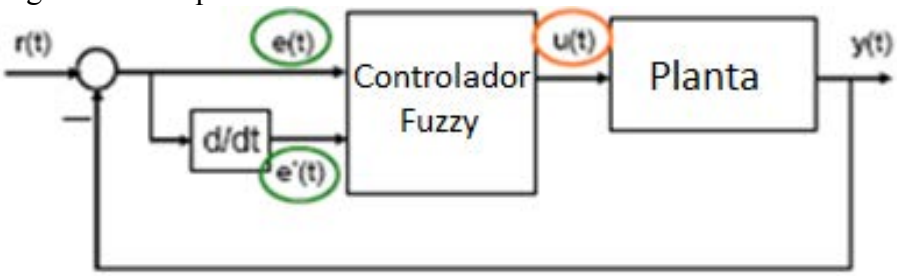
Figura 14 – Estrutura básica de controle.



Fonte: (PASSINO; YURKOVICH; 1998) modificado pelo autor

O controlador *fuzzy* utilizado tem como entrada duas variáveis, que são o erro e a variação (derivada) do erro, e uma saída. O erro utilizado é a variação da tensão lida pela entrada analógica da placa de aquisição de dados em relação a entrada desejada, e a derivada do erro é a taxa com que esse erro varia. A saída do controlador será a tensão necessária para atuar na planta (motor DC), aumentando ou diminuindo a velocidade do motor. Na Figura 15 tem-se o diagrama esquemático utilizado neste trabalho, onde e(t) e e'(t) são entradas e u(t) é saída do controlador.

Figura 15 – Esquema do controle do motor utilizado

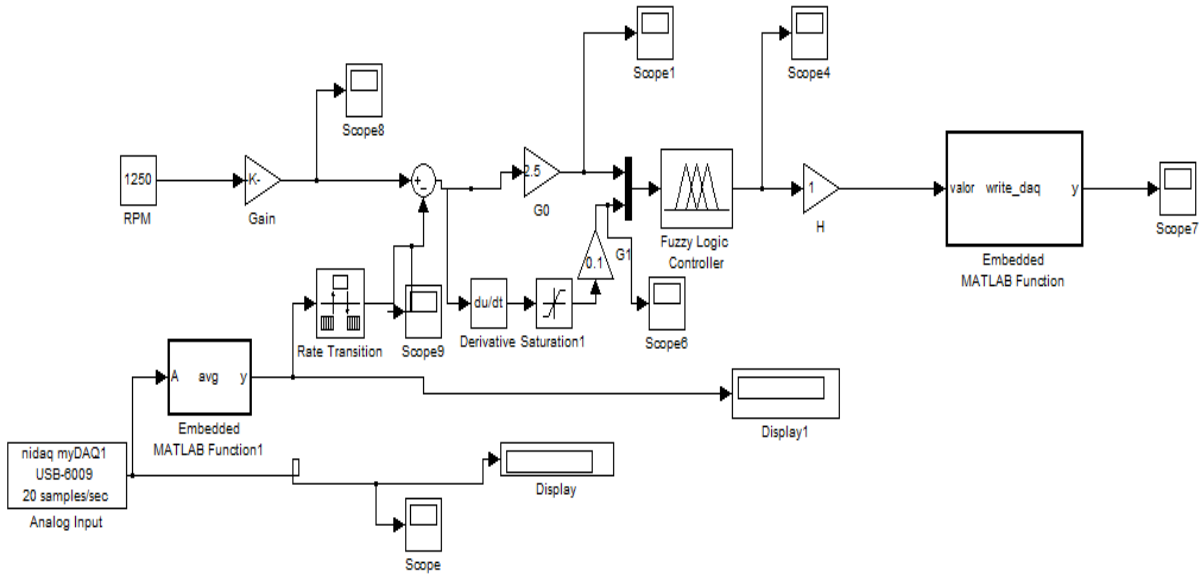


Fonte: (PASSINO; YURKOVICH, 1998) modificado pelo autor.

3.3 DIAGRAMA DE BLOCOS

A Figura 16 mostra o sistema implementado no *Simulink*. Para tornar mais fácil a compreensão, o modelo será explicado detalhadamente nas seguintes partes: entrada, controlador *fuzzy*,*write_daq*, *Analog Input* e ajustes das funçoes de pertinência.

Figura 16- Modelo de controle implementado no *Simulink.*

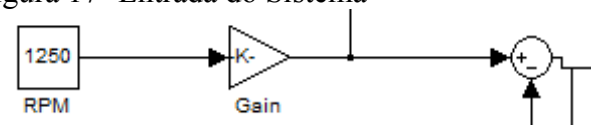


Fonte: (Autoria própria)

3.3.1 Entrada

Na entrada tem-se o valor de referência da velocidade do motor, ou seja, sua velocidade desejada, a velocidade que deve-se obter na saída (Figura 17). O ganho após o valor de entrada é o valor de conversão entre RPM e Volts, que para o kit PCT-1 é 1/500, ou seja, 1000 RPM equivalem a 2V.

Figura 17- Entrada do Sistema

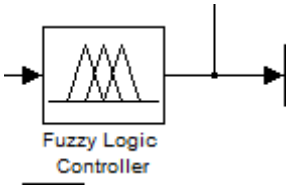


Fonte: (Autoria Própria)

3.3.2 Controlador *fuzzy*

O bloco que faz o controle *fuzzy* por meio do *toolbox fuzzy* do Matlab é mostrado na figura 18.

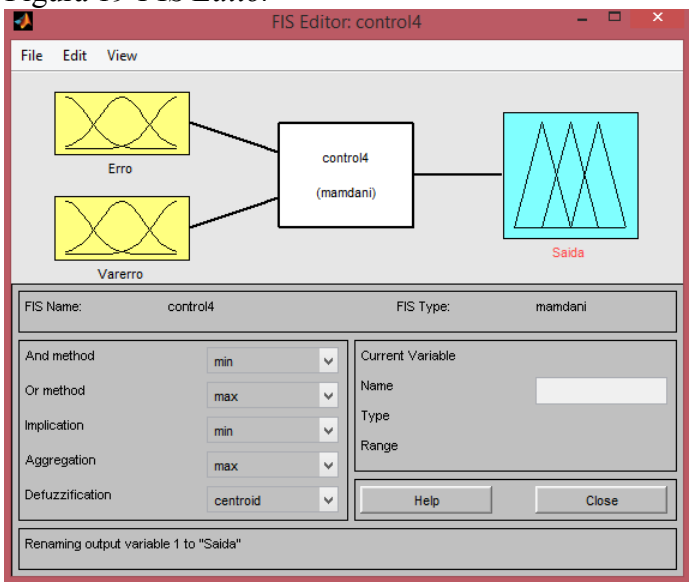
Figura 18- Controlador *fuzzy*



Fonte: (Autoria Própria)

Para criar um sistema de inferências no Matlab é necessário utilizar o *FIS Editor* do *Fuzzy Logic Toolbox*. O *FIS* criado é mostrado na Figura 19.

Figura 19-*FIS Editor*

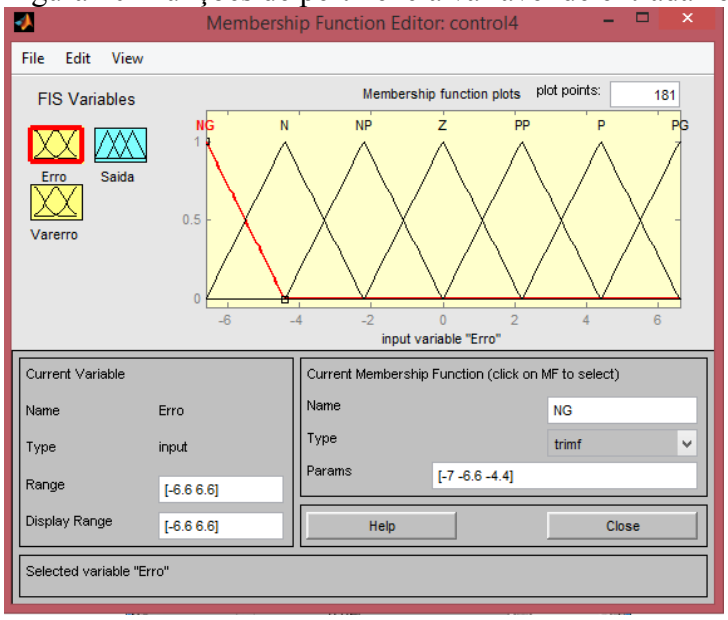


Fonte: (Autoria própria)

Neste editor, como pode-se perceber, é possível escolher o tipo de modelo utilizado (Mamdani ou Sugeno), o método de defuzzificação, número de entradas, entre outras coisas.

As funções de pertinência de entrada e saída são ilustradas a seguir.

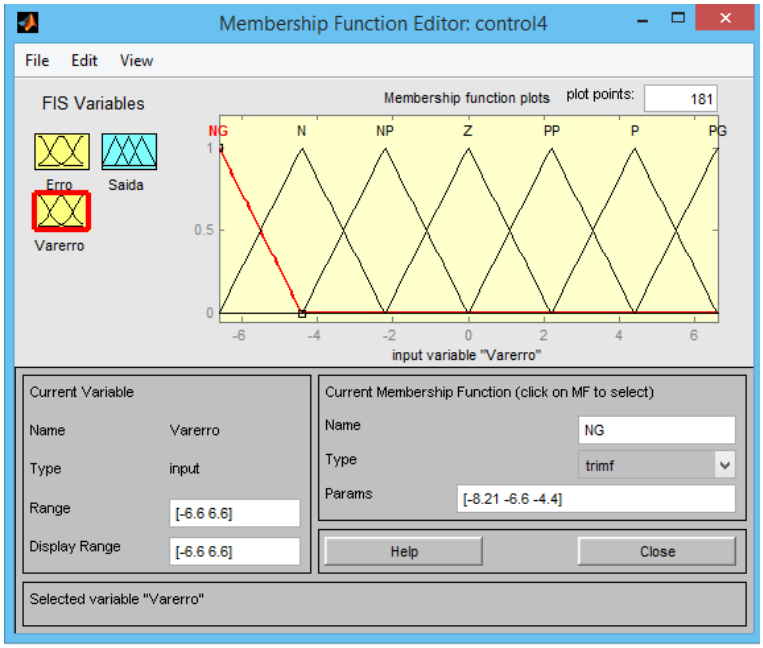
Figura 20- Funções de pertinência variável de entrada "erro"



Fonte: (Autoria própria)

Os termos utilizados para a função de pertinência do erro foram NG (Negativo Grande), N (Negativo), NP (Negativo Pequeno), Z (Zero), PP (Positivo Pequeno), P (Positivo), PG (Positivo Grande), os três primeiros para o caso em que a tensão lida no tacogerador pela entrada analógica da USB 6009 seja maior que a desejada, o zero para quando a tensão lida seja próxima ou igual a desejada e os três últimos para quando a tensão lida seja menor que a tensão desejada.

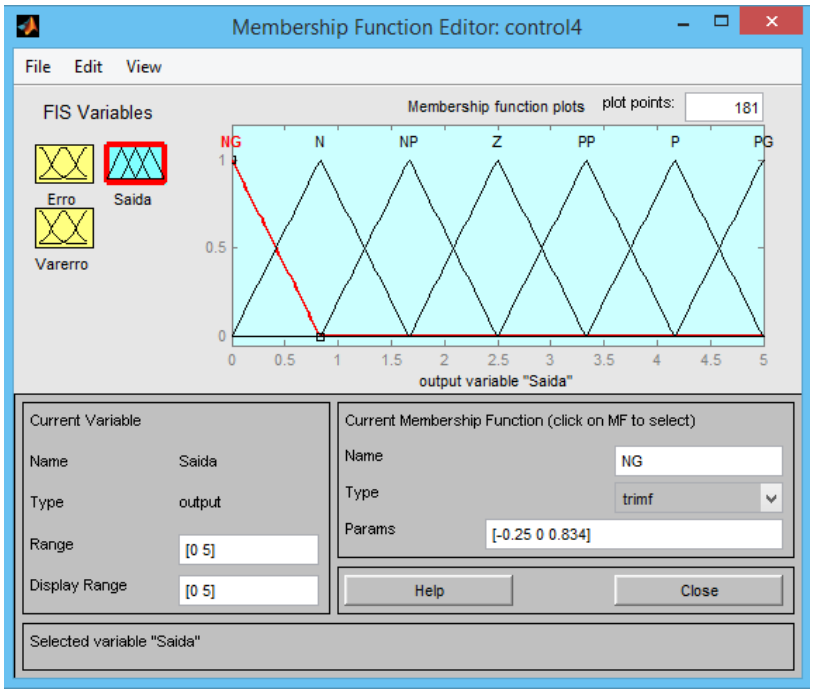
Figura 21- Função de pertinência entrada “Varerro”



Fonte: (Autoria própria)

Para esta entrada, a função de pertinência e os termos utilizados são os mesmos que para entrada “erro”, porém esta é utilizada para verificar a variação do erro obtido.

Figura 22- Função de pertinência da saída



Fonte: (Autoria própria)

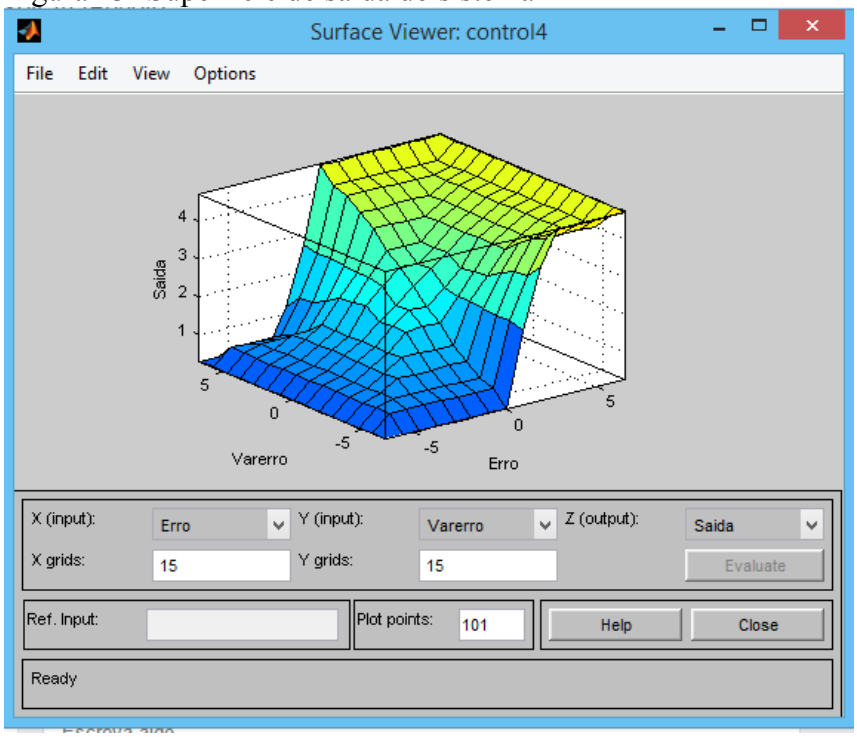
Para a saída, os termos utilizados foram os mesmos da entrada, porém a faixa de 0 a 5 se deve ao fato que a placa USB 6009 tem um tensão de saída que varia de 0 a 5V, portanto o controle foi adaptado seguindo essa condição. A saída é escolhida conforme a base de regras mostrada na Tabela 1.

Tabela 1- Representação da base de regras do sistema de inferência

		Erro						
		NG	**N**	**NP**	**Z**	**PP**	**P**	**PG**
	NG	NG	NG	NG	NG	PG	PG	PG
	N	NG	N	N	N	P	P	PG
Variação	**NP**	NG	N	NP	NP	PP	P	PG
do Erro	**Z**	NG	N	NP	Z	PP	P	PG
	PP	NG	N	NP	PP	PP	P	PG
	P	NG	N	N	P	P	P	PG
	PG	NG	NG	NG	PG	PG	PG	PG

A superfície de saída do sistema é apresentada na Figura 23.

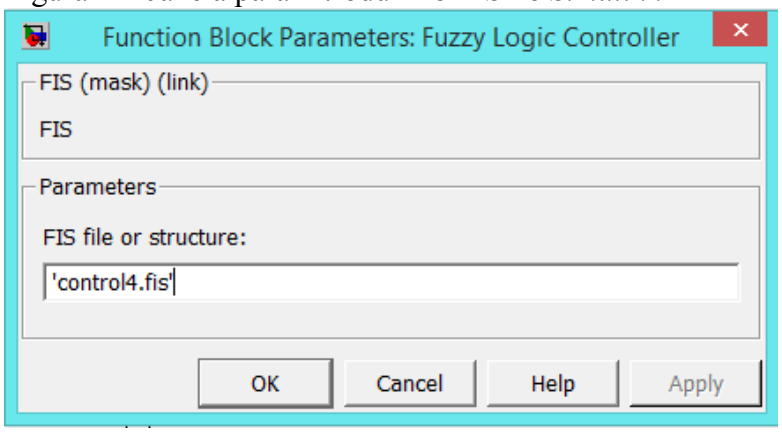
Figura 23- Superfície de saída do sistema



Fonte: (Autoria própria)

Com este sistema de inferência *fuzzy* pronto colocá-lo no *Simulink* conforme mostrado na Figura 24.

Figura 24- Janela para introduzir o *FIS* no *Simulink*

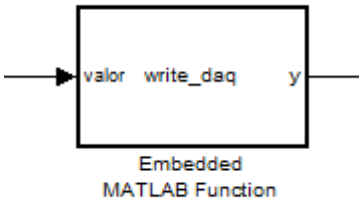


Fonte: (Autoria própria)

3.3.3 Função *write_daq*

Como o bloco *Analog Output* do *Simulink* não funciona para a placa NI USB 6009, pois esta não tem sinal de *clock* interno, foi necessário criar uma função para que fosse possível escrever um valor de tensão na interface do motor.

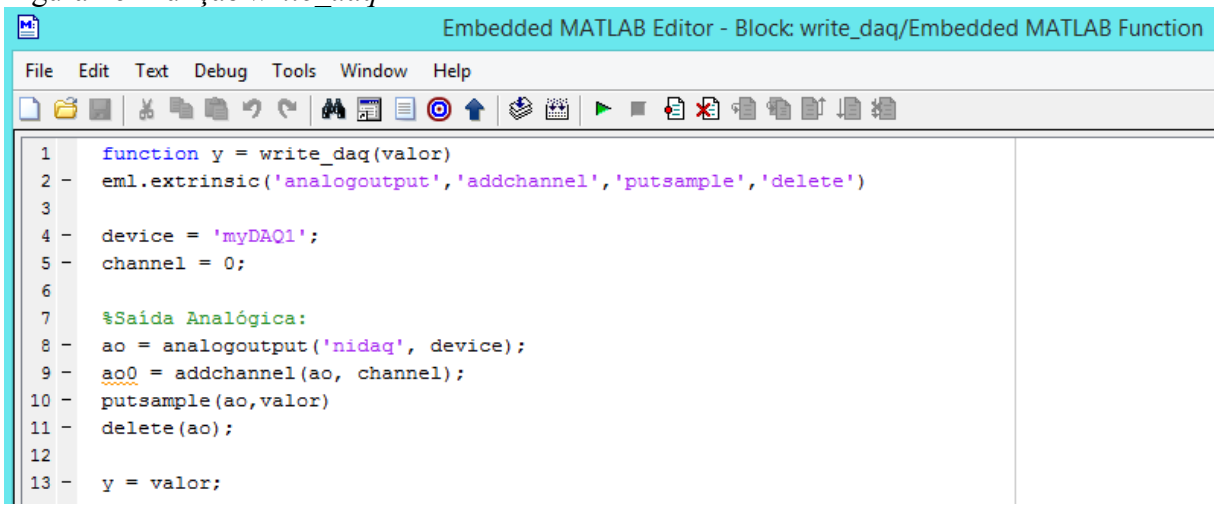
Figura 25- Bloco criado para gerar uma tensão de saída



Fonte: (Autoria própria)

A função criada é mostrada na Figura 26.

Figura 26- Função *write_daq*



Fonte: (Autoria própria)

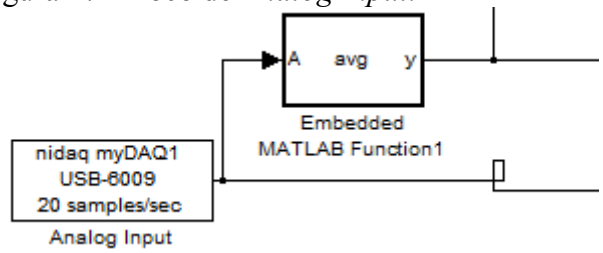
3.3.4 *Analog Input*

Este bloco (Figura 27) é responsável por fazer a leitura da tensão vinda do tacogerador, fazendo a conversão desta tensão é possível saber a velocidade atual do motor.

Com este bloco é possível escolher o número de amostras por segundo, quantos canais serão utilizados e o tipo de sinal.

Como o tipo de sinal utilizado foi o diferencial, a saída desse bloco é uma matriz 2x1 e esse é um dado que o controlador *fuzzy* não entende, portanto não seria possível compilar a simulação. Para isso, foi criado uma função que pega os dados da matriz e faz a média desses valores, tendo como resultado um dado unitário que o controlador *fuzzy* consegue trabalhar.

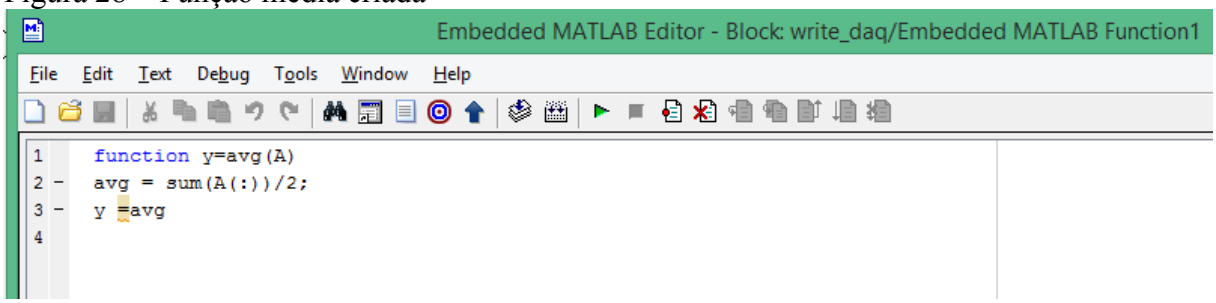
Figura 27- Bloco de *Analog Input*.



Fonte: (Autoria própria)

Na Figura 28, tem-se o código que calcula a função média criada para o bloco de *Analog Input*.

Figura 28 – Função média criada

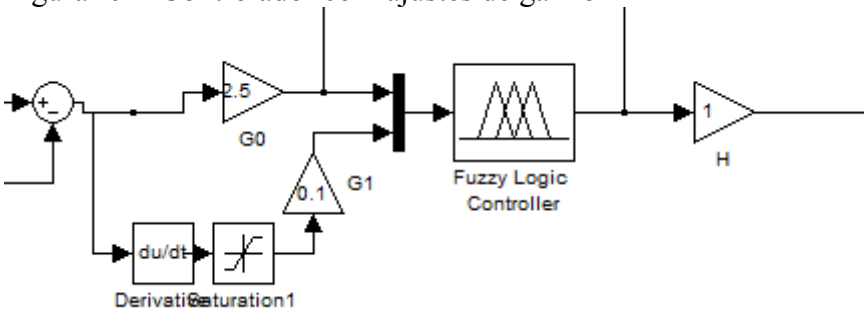


Fonte: (Autoria própria)

3.3.5 Ajustes das funções de pertinência

Para obter melhores resultados do controlador *fuzzy* é necessário introduzir ganhos nas entradas e saída do controlador, como ilustrado na Figura 29.

Figura 29 – Controlador com ajustes de ganho



Fonte: (Autoria própria)

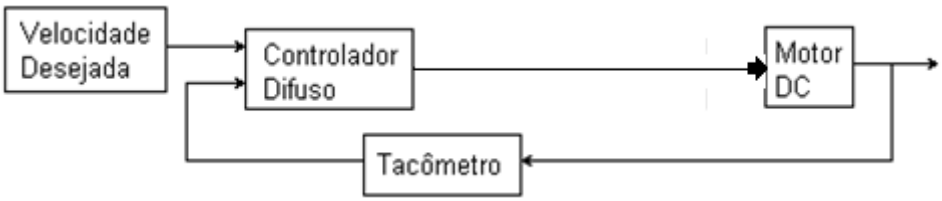
Para a escolha desses ganhos, devem ser testados diferentes valores em cada um deles separadamente, ou seja, enquanto o ajuste de um ganho é feito os outros devem ter o ganho unitário.

Neste trabalho, para os valores testados, os melhores resultados obtidos foram para G0 = 2.5, G1 = 0.1 e H = 1.

4 RESULTADOS

Para realizar o controle da velocidade de rotação do motor de corrente contínua, implementou-se o esquema mostrado na Figura 30.

Figura 30 - Esquema do controle da velocidade de rotação do motor



Fonte: (MARIMOTO, 2000) modificado pelo autor

Pela Figura 30, é possível observar que o controlador *fuzzy* possui como entradas o valor de velocidade desejada e a tensão gerada pelo tacômetro do motor de corrente contínua.

O controlador *fuzzy* compara os dois valores e por meio da lógica implementada no *toolbox fuzzy* do Matlab e define o nível de tensão necessária na saída.

4.1 VARIÁVEIS PARA O CONTROLADOR

Para se determinar essas variáveis, utilizou-se o *kit* PCT – 1 para se obter o valor das grandezas. Como a tensão de saída analógica da plana NI – USB – 6009 vai de 0 a 5V, utilizou-se um ganho no motor tal que a velocidade de referência e a velocidade do motor fossem iguais para um nível de tensão de 2.5V. A Tabela 2 mostra a relação entre a tensão de entrada com a tensão medida no tacômetro e a velocidade do motor.

Tabela 2- Medições para as variáveis do controlador

Vin do Motor (V)	V Tacômetro (V)	Velocidade do motor (RPM)
0,5	0	0
1	0,7	360
1,5	1,3	675
2	1,9	990
2,5	2,5	1240
3,0	3,2	1620
3,5	3,8	1900
4,0	4,3	2250

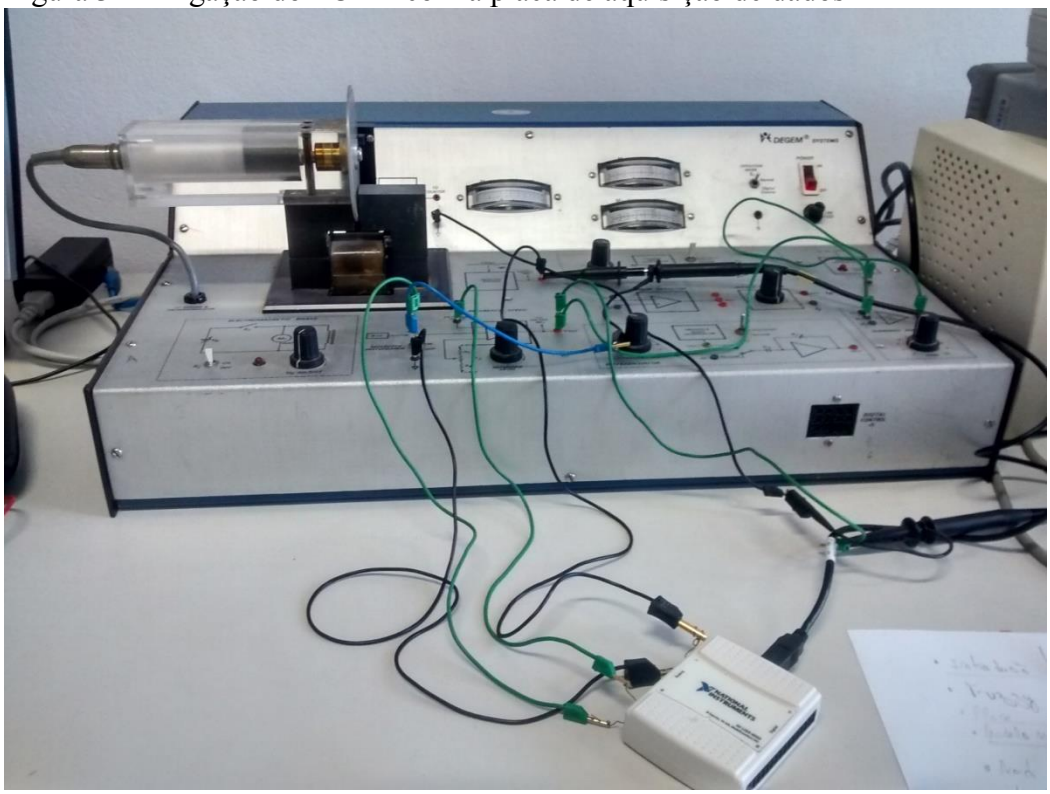
Tabela 2- Medições para as variáveis do controlador

4,5	5,0	2500
5,0	5,6	2650
5,5	6,1	3150

4.2 MONTAGEM REALIZADA E RESULTADOS OBTIDOS

A Figura 31 mostra como o motor foi ligado à placa NI – USB - 6009.

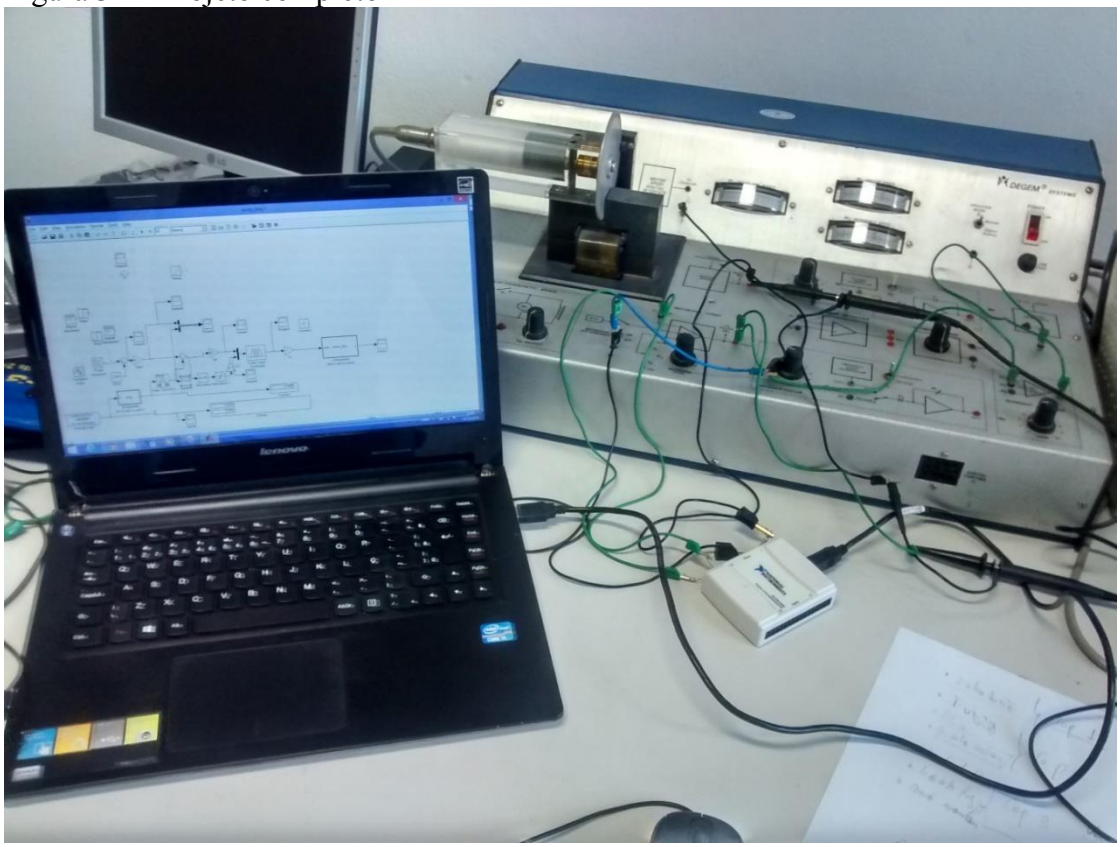
Figura 31 – Ligação do PCT-1 com a placa de aquisição de dados



Fonte: (Autoria própria)

Na Figura 32 tem-se a imagem do sistema completo, com o computador ligado à placa e controlando o motor.

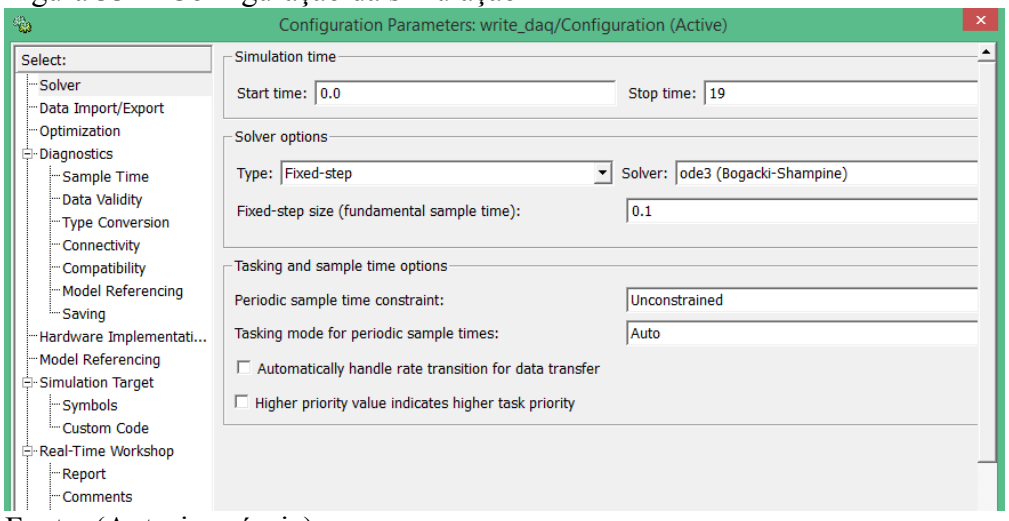
Figura 32 – Projeto completo



Fonte: (Autoria própria)

Para a realização dos testes, o bloco *Analog Input* foi ajustado para 20 amostras por Segundo e a simulação do *Simulink* foi configurada conforme é mostrado na Figura 33.

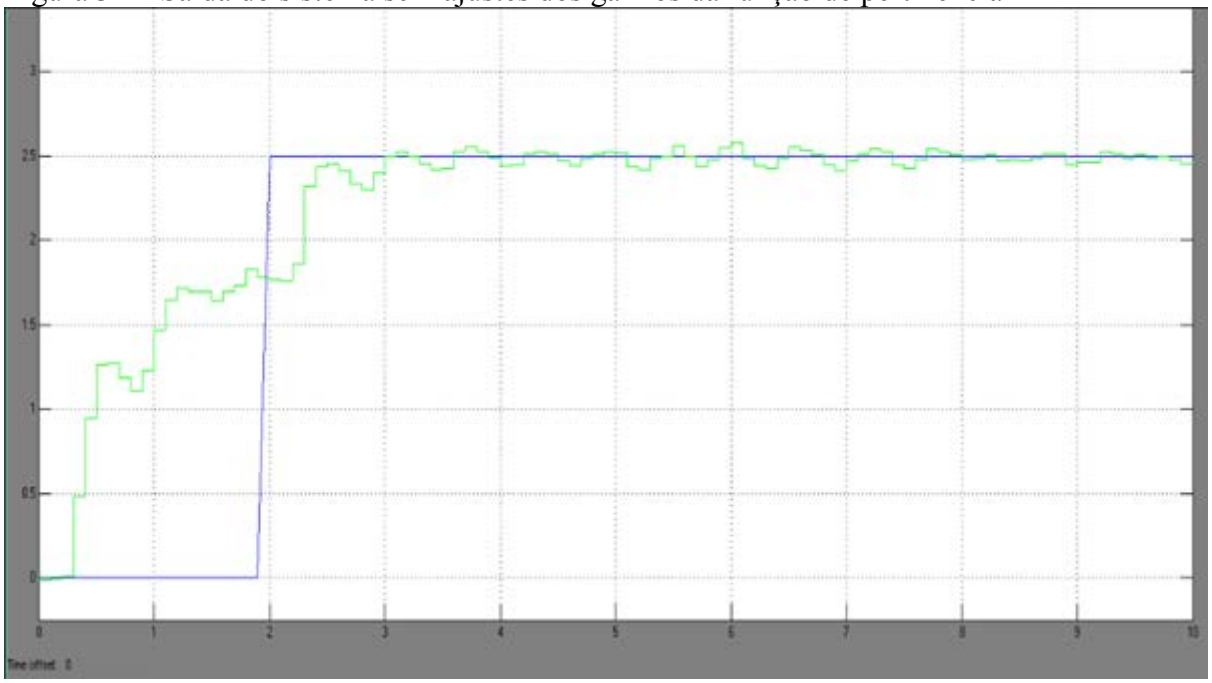
Figura 33 – Configuração da simulação



Fonte: (Autoria própria)

Para uma entrada desejada de 1250 RPM (2.5V) e sem os ganhos para ajustar as funções de pertinência, tem-se o sinal de saída mostrado na Figura 34.

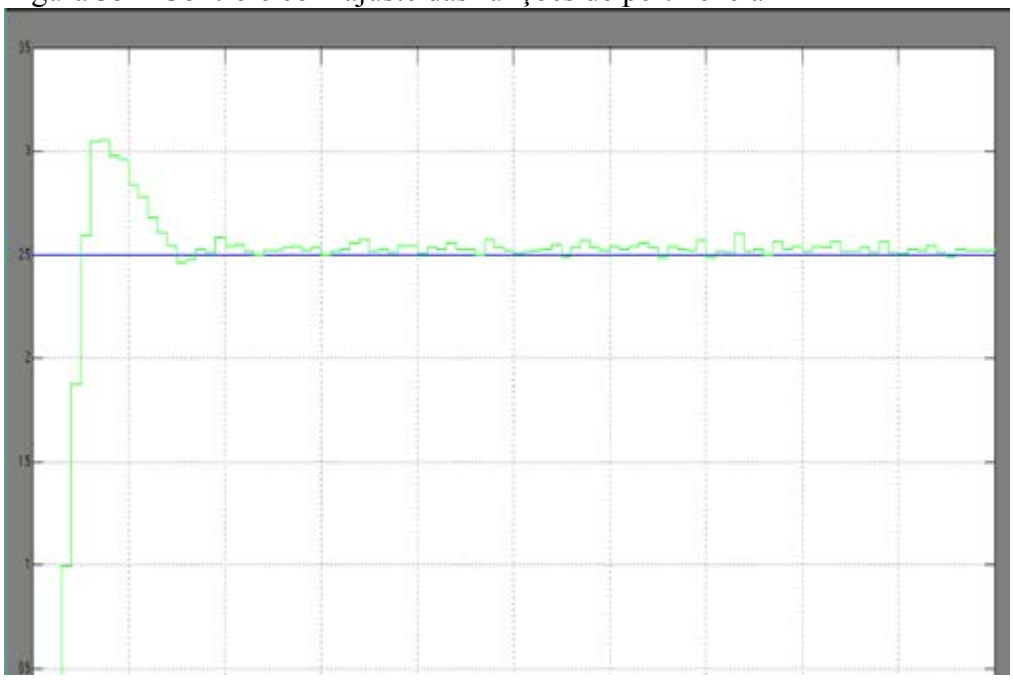
Figura 34 – Saída do sistema sem ajustes dos ganhos da função de pertinência



Fonte: (Autoria própria)

Inserindo os ganhos de ajuste das funções de pertinência e utilizando a mesma entrada de 1250 RPM, tem-se a seguinte saída.

Figura 35 – Controle com ajuste das funções de pertinência

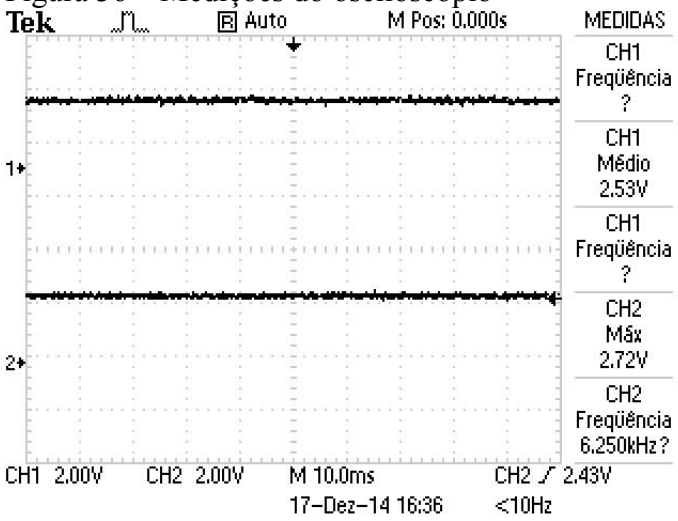


Fonte: (Autoria própria)

Com o oscilóscopio medindo a tensão do tacogerador no Canal 1 e a tensão de saída da placa USB 6009 no Canal 2, tem-se a medida apresentada na Figura 36. Como foi

medido um sinal contínuo, o osciloscópio não consegue ler um valor de frequência correto e por isso aparece um ponto de interrogação em seu visor.

Figura 36 – Medições do osciloscópio



Fonte: (Autoria própria)

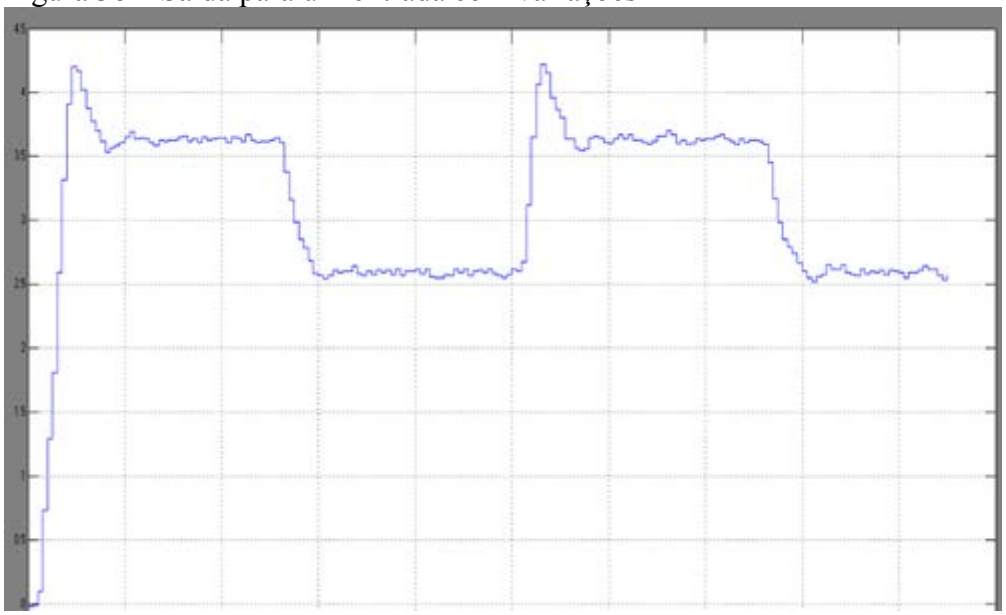
Para verificar se a saída está acompanhando a referência foi colocada na entrada uma função que varia entre 1250 RPM e 2000 RPM (de 2.5V a 4V), como mostrado na Figura 37.

Figura 37 – Entrada desejada com variação

Fonte: (Autoria própria)

Com esta entrada, a saída obtida foi mostrada na Figura 38.

Figura 38 – Saída para um entrada com variações



Fonte: (Autoria própria)

Pode-se perceber que a saída responde às variações, porém quando o valor desejado é de 4V, a saída tem um pequeno erro em relação à entrada. Isto ocorre porque foi desenvolvido um controlador PD *Fuzzy* e, portanto, esse erro é natural. Para não ocorrer esse erro, é necessário desenvolver um controlador PID *Fuzzy*.

5 CONCLUSÃO

Neste trabalho foi apresentado o estudo da lógica *fuzzy,* bem como a utilização de uma placa de aquisição de dados para controlar o *kit* de motor de corrente contínua por meio de um modelo criado no *simulink.*

Para a construção do controlador PD *fuzzy* no *simulink* utilizou-se o *Toolbox Fuzzy* do *Matlab* onde não houveram grandes complicações, visto que este *toolbox* é bem intuitivo e de fácil implementação.

Para fazer a comunicação entre o computador e a placa de aquisição de dados foi necessário utilizar os blocos *DAQ* do *simulink*, porém por uma limitação da placa NI USB 6009 o bloco *Analog Output* não pôde ser utilizado, sendo necessário criar uma função para escrever um valor de tensão na saída analógica, esta função funcionou conforme o esperado e com ela foi possível acionar o motor.

Os resultados encontrados foram satisfatórios, mostrando que o controlador funcionou corretamente e foi possível ajustar a velocidade do motor, somente quando há variação na entrada do sistema que aparece um erro na saída, como foi explicado na seção 4. Esse erro poderia ser contornado ajustando as funções de pertinência do *fuzzy* ou implementando um controlador PID *Fuzzy*.

Por fim, os objetivos deste trabalho foram alcançados, com a utilização de uma placa de aquisição de dados para controlar um motor DC utilizando lógica *fuzzy*

Como trabalhos futuros recomenda-se o desenvolvimento de um controlador PID *Fuzzy* e também utilizar lógica *fuzzy* tipo 2. Também recomenda-se utilizar o freio presente no *kit* para verificar se o sistema está funcionando corretamente.

REFERÊNCIAS

ALTAS, I. H. **Dynamic Model of a permanente magnet DC Motor.** 2007 Disponível em: < http://www.che.rochester.edu/course-sites/CHE%20272/Homeworks/Doug272_2013.pdf> Acessado em setembro de 2014.

JANE, D. de A. **Uma introdução ao estudo da lógica *fuzzy*** Disponível em: <http://www.faeso.edu.br/horus/artigos anteriores/2004/artigo_dario.pdf> Acessado em novembro de 2014.

LAC, Retirado de **Escola de Verão 2008, Laboratório Associado de Computação e Matemática Aplicada – LAC** Disponível em: <http://www.lac.inpe.br/ ~demisio/download/ia_elac07/ia-lac6.pdf> Acessado em novembro de 2014.

MARIMOTO, D.S. **Controle de motor DC utilizando sistema de aquisição de dados e lógica difusa (*fuzzy logic*).** 2000. Trabalho de graduação (Graduação em Engenharia Elétrica) – Faculdade de Engenharia de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2000.

NATIONAL INSTRUMENTS Disponível em: <http://sine.ni.com/nips/cds/view/p/lang/pt/nid/201987> Acessado em dezembro de 2014.

OMIDEH, Z.T. **Fuzzy Logic Control of a magnetic levitation system with hardware implementation**. Arlington, 2003. 111p.

PASSINO, K. M.; YURKOVICH, S. **Fuzzy Control**. Menlo Park: Addison Wesley Longman, Inc., 1998. 502 p.

PUCRS, Retirado de **Faculdade de Engenharia da Pontifícia Universidade do Rio Grande do Sul – PUCRS** Disponível em: <http://www.feng.pucrs.br/~gacs/new/disciplinas/model/apostilas/Aula5.pdf > Acessado em setembro de 2014.

RIZOL, P.M.S.R **Introdução a lógica Difusa**. Universidade Estadual Paulista – Campus Guaratinguetá. Departamento de Engenharia Elétrica. 2008.

RODRIGUEZ, C.F.O. **Estudo de aplicação utilizando controladores fuzzy.** 2010. Trabalho de graduação (Graduação em Engenharia Elétrica) – Faculdade de Engenharia de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2010.

ROSS, T.J. **Fuzzy Logic with engineering applications**. Second Edition. West Sussex: John Wiley & Sons Ltd, 2004. 628p

SANTIAGO, R. Y. **Modelagem e controle de um pêndulo invertido utilizando a lógica fuzzy no matlab.** 2008. Trabalho de graduação (Graduação em Engenharia Elétrica) – Faculdade de Engenharia de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2008.