



UNIVERSIDADE FEDERAL DE OURO PRETO
ESCOLA DE MINAS
COLEGIADO DO CURSO DE ENGENHARIA DE
CONTROLE E AUTOMAÇÃO - CEAU



ARTHUR REIS LARA MIRANDA

COMPARAÇÃO ENTRE CONTROLADORES CLÁSSICOS E FUZZY
COMANDANDO UM PENDULO INVERTIDO TIPO CARRO IMPLEMENTADO
COM LEGO MINDSTORM EV3

MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA
DE CONTROLE E AUTOMAÇÃO

Ouro Preto, 2015

ARTHUR REIS LARA MIRANDA

COMPARAÇÃO ENTRE CONTROLADORES CLÁSSICOS E FUZZY COMANDANDO
UM PENDULO INVERTIDO TIPO CARRO IMPLEMENTADO COM LEGO
MINDSTORM EV3

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para obtenção do Grau de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Paulo Marcos de Barros Monteiro

Co-Orientador: D. Sc. Gustavo Medeiros Freitas

Ouro Preto

Escola de Minas – UFOP

Julho/2015

Monografia defendida e aprovada, em 03 de julho de 2015, pela comissão avaliadora constituída pelos professores:



Prof. Dr. Paulo Marcos de Barros Monteiro - Orientador



Prof. Dr. Gustavo Medeiros Freitas (Co-orientador)



Prof. MSc. José Alberto Navés Cocota Junior – Professor Convidado

AGRADECIMENTOS

Agradeço a todos as pessoas que foram importantes para o desenvolvimento deste trabalho: Aos meus familiares que sempre se incentivam, à República Skulaxu; aos amigos da automação que compartilham os laboratórios de Máquinas Elétricas e LAP; Aos meus Orientadores Paulo Marcos De Barros Monteiro, Gustavo Medeiros Freitas e José Alberto Naves Cocota Junior por me ensinarem tanto ao longo dos últimos semestres; Ao professor João Carlos Vilela De Castro pelas dicas; Aos funcionários da Escola de Minas; à Flávia que de forma especial e carinhosa me deu força e coragem, me apoiando nos momentos de dificuldades.

RESUMO

Com o avanço da tecnologia e com o surgimento de novas ferramentas educacionais é possível estudar várias técnicas de controle de forma prática em um laboratório de universidade. Técnicas de controle são comuns no dia a dia, aplicadas em eletroeletrônicos, passando pela indústria, indo até aplicações espaciais. Dentre estas técnicas os controladores clássicos e os de lógica nebulosa se destacam por possuir diferentes abordagens, benefícios e custos computacionais. Os controladores clássicos são muito utilizados pois apresentam provas matemáticas de estabilidade e podem ser facilmente ajustados pelo método do Lugar das Raízes. Já o *Fuzzy* é utilizado com frequência no controle de sistemas para os quais o modelo matemático é difícil de ser determinado. Neste trabalho apresenta-se o uso de controladores por avanço de fase e de lógica nebulosa empregados no controle de atitude de um pêndulo invertido. O exemplo de pêndulo invertido abordado neste trabalho é o de um carro que serve como base para uma longa haste, que deve ser equilibrada na posição vertical. Cada controlador analisado é programado por meio do toolbox SIMULINK do *Software* MATLAB e embarcado no controlador do kit de robótica LEGO *Mindstorms* EV3. Simulações e experimentos são realizados para analisar o desempenho de cada controlador analisando a aplicação de cada um deles para o comando de um pêndulo invertido do tipo carro.

Palavras Chave: Teoria de controle, Avanço de Fase, Controlador Nebuloso, LEGO.

ABSTRACT

The advance of technologies and the appearance of new educational tools allow the practical study of new controls techniques inside a university's laboratory. Control techniques are commonly used in the daily basics, applied from the Household Appliances, through industry, to the Space applications. Among them the Classical controller and the Fuzzy controllers stands out for different approaches, benefits and computational costs. The Classical one is the most common controller; it has a mathematical background to prove the stability and is easily adjustable by the root locus. The Fuzzy is a rising technique, mostly applied to control systems where the mathematical model is unknown. In this paper, we demonstrate the use of a classical controller and Fuzzy controller to control the attitude of an inverted pendulum that consist of a cart serving as a base for a long rod that should be balanced on the vertical position. Each controller is programmed by the toolbox SIMULINK of the Software MATLAB and embedded in the CPU of a LEGO's Mindstorms EV3 robotic kit. All the simulations and practical experiments are performed on SIMULINK to compare the difference between each control approach performance to analyse the utility of them applied to the pendulum.

Keywords: Control Theory, Phase Lead, Fuzzy Controller, LEGO

SUMÁRIO

1	INTRODUÇÃO	7
1.1	Objetivo	9
1.2	Justificativa	9
1.3	Metodologia.....	9
1.4	Estrutura do trabalho.....	10
2	REFERENCIAL TEÓRICO	11
2.1	Modelagem matemática.....	11
2.1.1	Espaço de estados e função de transferência	13
2.2	A plataforma LEGO <i>Mindstorms</i>	15
2.2.1	LEGO <i>Mindstorms</i> EV3 motor CC	16
2.2.2	Modelo Matemático de um Motor CC.....	17
2.3	Controlador por Avanço de Fase	18
2.3.1	Efeitos da adição de pólos	19
2.3.2	Efeitos da adição de zeros.....	19
2.4	O projeto de um controlador por avanço de fase por meio do lugar das raízes.....	19
2.5	A lógica <i>Fuzzy</i>	20
2.5.1	Variáveis Linguísticas.	21
2.5.2	Função de Pertinência.....	21
2.5.3	Fuzificação e Inferência.....	23
2.5.4	Defuzificação	24
2.6	Controlador Nebuloso.....	25
2.6.1	A simulação em tempo real	26
2.6.2	Controladores Fuzzy para um pêndulo invertido.....	26
3	MATERIAIS E MÉTODOS	31
3.1	Desenvolvimento da estrutura física.....	31
3.2	Modelagem do Pêndulo Invertido.	33
3.2.1	Redução de ordem da função de transferência	34
3.2.2	Análise da estabilidade	35
3.2.3	Projeto do controlador por avanço de fase	36
3.2.4	Projeto do Controlador <i>Fuzzy</i>	37

3.3	Simulações e Experimentos.....	39
3.3.1	Sistema de aquisição do ângulo da Haste.....	39
3.3.2	Simulação do controlador por avanço de fase.....	40
3.3.3	Experimento do controlador por avanço de fase.....	41
3.3.4	Simulação do controlador nebuloso.....	42
3.3.5	Experimento do controlador nebuloso.....	42
3.3.5.1	Controlador nebuloso de 17 regras.....	43
3.3.5.2	Controlador nebuloso de 9 regras.....	45
4	ANÁLISE DE RESULTADOS.....	48
4.1	Resultados do controlador por avanço do fase.....	48
4.2	Análise dos resultados do controlador por avanço de fase.....	49
4.3	Resultados das simulações com controlador <i>Fuzzy</i> (25 regras).....	50
4.4	Experimentos dos controladores nebulosos simplificados.....	51
4.4.1	Controlador nebuloso de 17 regras.....	51
4.4.2	Controlador nebuloso de 9 regras.....	52
4.5	Análise do desempenho do controlador <i>Fuzzy</i> de 9 regras.....	55
4.6	Comparação entre os controladores <i>Fuzzy</i> e por avanço de fase.....	55
5	CONCLUSÃO.....	56
	REFERÊNCIAS BIBLIOGRÁFICAS.....	58
	APÊNDICE.....	62

1 INTRODUÇÃO

A Engenharia de Controle e Automação estuda em sua área de conhecimento o controle de atividades ou processos buscando a maior eficiência energética, rapidez e redução de custos. Dentre as várias opções de controladores, o controlador clássico é projetado com base no modelo do sistema dinâmico a ser comandado. Este modelo aproxima a dinâmica do sistema por um conjunto de equações, que pode não ser único. Já a lógica nebulosa não emprega diretamente um modelo matemático do sistema. Ao invés disso, essa estratégia utiliza definições genéricas e informações fornecidas por operadores especialistas do sistema a ser controlador.

A escolha do pêndulo é feita por se tratar de um tradicional problema de controle. De que de acordo com Dorf e Bishop (2011, p. 186) o sistema de pêndulo invertido está relacionado, por exemplo, ao controle de foguetes, mísseis, plataformas e naves espaciais em que o centro de gravidade está localizado acima do ponto de ação da força. Goodwin, Graebe e Salgado (2000, p. 55) comparam o problema do pêndulo invertido com o de um leme de um navio, onde o movimento deste pêndulo atua na direção do navio.

Este sistema é instável, ou seja, não retorna à posição inicial após receber um pequeno distúrbio. Pretende-se utilizar o sistema de um pêndulo invertido para comparar o desempenho de um controlador convencional por avanço de fase e um controlador inteligente baseado em Lógica Nebulosa.

Uma melhor definição de um pêndulo é dada por Castro (2012, p. 1), onde são exemplificados os pêndulos não-invertidos como as cargas de guindastes e o braço oscilatório de grandes relógios em que tendem a oscilar com amplitude controlada em torno de um ponto de equilíbrio o qual é atingido quando a força que origina o movimento é cessada. Os pêndulos invertidos não podem naturalmente se estabilizar em um ponto de equilíbrio ou oscilar controladamente, após o distúrbio o pêndulo tende a cair. Devido à esta dinâmica instável, pêndulos invertidos raramente são empregados em produtos do dia-a-dia. Outros exemplos de atuação de um pêndulo invertido são listados por Akhtaruzzaman e Shafie (2010, p. 1342) como o sistema de pouso automático de aviões e o sistema que os mantém em voo estável durante uma tempestade e/ou momentos de turbulência.

Outro exemplo muito estudado atualmente é o *Segway*, pêndulo invertido de duas rodas, os artigos de Kanada, Watanabe e Chen (2011) e Kushida et al. (2012) estudam a aplicação de diferentes controladores para o comando deste tipo de pêndulo e utilizam o kit LEGO *Mindstorms* para validação de seus modelos.

A resposta de um sistema dinâmico nem sempre satisfaz as necessidades de seu funcionamento, por isso modelos matemáticos podem necessitar de uma compensação para atender determinadas especificações. Ogata (2003, p. 342) exemplifica essas especificações como requisitos da resposta transitória e do regime permanente. Existem vários métodos para se projetar um controlador clássicos. Vários *Software* possuem ferramentas para sintonia de controladores PID, como o *PID tuner* no MATLAB que automatiza os cálculos para obtenção dos parâmetros do controlador.

A outra abordagem para controlar um sistema dinâmico utilizada neste trabalho é a lógica *Fuzzy* ou lógica nebulosa. Com base no conhecimento dos especialistas, o controlador nebuloso atua no processo de tomada de decisões por meio de regras. Essas regras são declarações no formato: SE (Premissa) ENTÃO (Conclusão). Segundo Sandri e Correa (1999, p. c73), as informações dos especialistas possuem relação com a Teorias dos Conjuntos Nebulosos (Fuzzy sets) e com a Teoria de Possibilidade desenvolvida por Zadeh em 1965.

Sandri e Correa (1999, p. c73) descrevem as utilidades da lógica nebulosa como sendo: “[...]controle de eletrodomésticos ao controle de satélites, do mercado financeiro à medicina, e tendem a crescer cada vez mais, sobretudo em sistemas híbridos, que incorporam abordagens conexionistas e evolutivas, no que é chamado hoje em dia, de *soft computing*”. A lógica nebulosa é uma das tecnologias atuais mais bem sucedidas para o desenvolvimento de sistemas para controlar processos sofisticados. Com sua utilização, requerimentos complexos podem ser implementados em controladores simples, de fácil manutenção e baixo custo.

De acordo com Sivanandam et al. (2007), a percepção de mundo pelos seres humanos é cercada de conceitos que não possuem forma ou valores definidos. Estes valores são denominados variáveis linguísticas, representando valores que as grandezas observadas podem assumir, e.g. muito, pouco, muito maior que, menor que, quente, frio, seco, úmido, dentre outros. Essas variáveis linguísticas correspondem à maneira com que o cérebro humano processa informações no dia a dia, enquanto computadores empregam números binários para a realização

de cálculos exatos. Desta forma o uso da lógica *Fuzzy* se torna mais fácil por utilizar conceitos intuitivos aos projetistas.

Outros tipos de controladores para um pêndulo invertido podem ser encontrados no trabalho de Akhtaruzzaman e Shafie (2010) onde são implementados um controlador PID de 2 graus de liberdade, um controlador do tipo regulador quadrático linear, LQR (*Linear Quadratic Regulator*) e um FSF (*Full State Feedback*) em um pêndulo invertido com base rotacional.

Com o uso de duas técnicas tão distintas de controle de sistemas dinâmicos, e um aprofundado estudo de modelagem matemática, este trabalho pretende oferecer uma grande base de conhecimento acadêmico e servir como ponto de partida para desenvolvimento de novos modelos utilizando o LEGO *Mindstorms* nos laboratórios da Escola de Minas da Universidade Federal de Ouro Preto.

1.1 Objetivo

O objetivo deste trabalho consiste em implementar os controladores clássico e nebuloso para o sistema de um pêndulo invertido. O problema de controle consiste em manter a haste do pêndulo na posição vertical.

1.2 Justificativa

O estudo do controle de atitude de um pêndulo invertido faz-se necessário em vários sistemas onde a força é aplicada em baixo do centro de massa. Este sistema se torna instável, ou seja, não retorna à posição inicial após receber um pequeno distúrbio.

Diversos controladores podem ser aplicados para efetuar o controle de um pêndulo, neste trabalho busca-se o estudo de dois destes controladores, o controlador por avanço de fase e o *Fuzzy*.

1.3 Metodologia

Este trabalho realiza uma abordagem do problema de controle de atitude de um pêndulo invertido a partir da revisão bibliográfica referente ao estudo de controladores por avanço de fase, e das possíveis modelagens de um sistema dinâmico de pêndulo invertido. Após a modelagem, é realizada a montagem da estrutura do pêndulo invertido tipo carro com o kit

LEGO *Mindstorms*. O modelo é validado e suas grandezas físicas são mensuradas para aplicar às simulações.

Posteriormente faz-se o teste da integração entre o LEGO e as plataformas de *Software* utilizadas. Após estes testes, o MATLAB/SIMULINK é escolhido como o ambiente de desenvolvimento dos controladores. A análise dos resultados fez-se por meio das leituras dos sensores do LEGO. Estes foram lidos pelo MATLAB que produz gráficos referentes às inclinações da haste do pêndulo invertido.

Por fim é projetado um controlador por lógica nebulosa utilizando o *toolbox* de lógica *Fuzzy* do MATLAB, a interação com o LEGO será feito por meio do SIMULINK e os dados dos experimentos serão apresentados de forma análoga aos do controlador clássico.

1.4 Estrutura do trabalho

No capítulo 1, é abordado o uma introdução ao problema de controle de atitude de um pêndulo invertido e aos controladores utilizados para o comando do carro.

O capítulo 2, aborda um estudo sobre a modelagem de um pêndulo invertido, estudo de teoria de controle, estudo de um motor de corrente contínua, análise das características do LEGO *Mindstorms* e um estudo sobre lógica *Fuzzy*.

No capítulo 3, é apresentado a estrutura física baseada no LEGO para este estudo, o modelo matemático do pêndulo montado no laboratório, o desenvolvimento dos controladores e a metodologia adotada para simulações e experimentos.

O capítulo 4, aborda uma análise dos resultados obtidos nas simulações comparando-os entre si, e explicando as possíveis variações existentes.

No capítulo 5, é apresentada as conclusões sobre o trabalho apresentando opiniões sobre o desenvolvimento do assunto, uso do LEGO e sugestões de trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo aborda alguns conceitos referentes ao entendimento do trabalho, faz-se uma revisão de modelagem matemática, da dinâmica de um pêndulo invertido, teoria de controle, apresenta informações referente ao LEGO *Mindstorms* e do estudo de lógica nebulosa (*Fuzzy*).

2.1 Modelagem matemática

Existem várias abordagens de modelagem matemática para um pêndulo invertido, para o caso do controle de atitude, este modelo é definido quando o centro de massa está localizado acima do ponto de aplicação da força. Ogata (2003, p. 71), Goodwin, Graebe e Salgado (2000, p. 55) e Dorf e Bishop (2011, p 186) abordam a dinâmica de um pêndulo invertido utilizando a modelagem no espaço de estados.

A Figura 2.1 ilustra o sistema de um pêndulo invertido, onde um carro serve como base pra uma haste que se equilibra sobre um ponto de apoio, na Tabela 2.1 são retratados os nomes das variáveis.

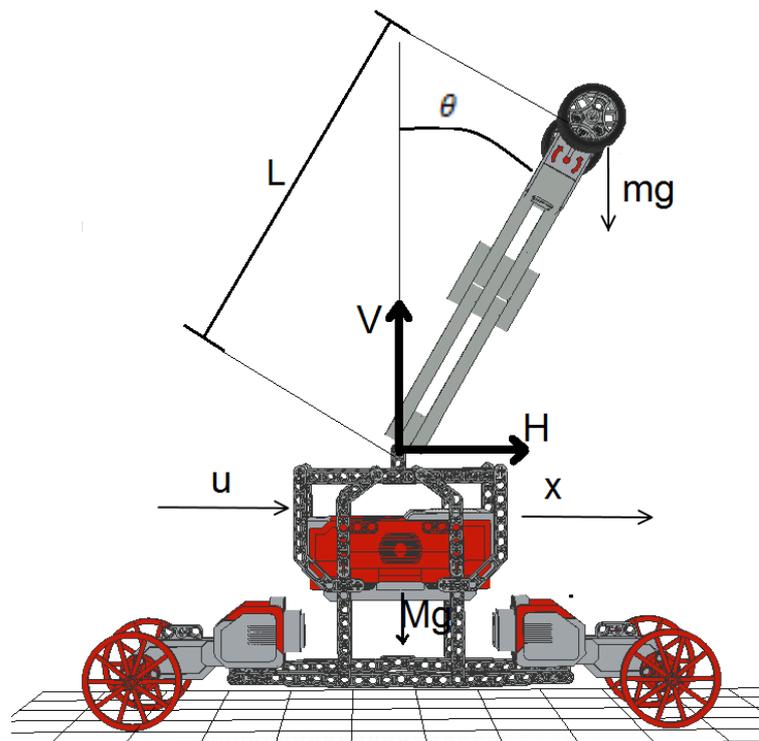


Figura 2.1- Sistema pêndulo invertido

Tabela 2.1-Símbolos do sistema de pêndulo invertido

Símbolo	Nome
θ	Ângulo de Inclinação do Pêndulo
L	Distância da base ao CM da haste
M	Massa do Carrinho
m	Massa da haste
g	Aceleração da Gravidade

O movimento rotacional da haste do pêndulo em torno do seu centro de gravidade é:

$$I\ddot{\theta} = VL \sin \theta - Hl \cos \theta \quad (2.1)$$

Onde I é o momento de inércia da haste. As equações seguintes representam o movimento horizontal expresso por H na Equação (2.2) e vertical V na Equação (2.3) do centro de gravidade da haste, e o movimento horizontal do carro (2.4).

$$m \frac{d^2}{dt^2} (x + L \sin \theta) = H \quad (2.2)$$

$$m \frac{d^2}{dt^2} (L \cos \theta) = V - mg \quad (2.3)$$

$$M \frac{d^2 x}{dt^2} = u - H \quad (2.4)$$

O objetivo do controle é manter o pêndulo na posição vertical. Para linearizar o sistema nesta configuração, aproxima-se os valores de $\sin \theta$ por θ e $\cos \theta$ por 1 para pequenos ângulos. Utilizando as Equações (2.2, 2.4) com valores linearizados deduz-se a Equação (2.5) isolando o termo H em cada uma delas. A partir das Equações (2.1-2.3) linearizadas é obtida a Equação (2.6) isolando-se os termos V e H em cada uma delas. Para o modelo em que o centro de massa da haste se encontra no topo da mesma, o valor de I é aproximado para 0.

$$(M + m)\ddot{x} + mL\ddot{\theta} = u \quad (2.5)$$

$$(I + mL^2)\ddot{\theta} + mL\ddot{x} = mgL\theta \quad (2.6)$$

Goodwin, Graebe e Salgado (2000, p. 55) modelam o pêndulo invertido usando física newtoniana, e também aplica a linearização pelo valor de θ . As equações (2.7) e (2.8) representam as acelerações no eixo x, chamado de $y(t)$ e angular respectivamente.

$$\ddot{y} = \frac{1}{\lambda_m} \left[\frac{f(t)}{m} - g\theta(t) \right] \quad (2.7)$$

$$\ddot{\theta} = \frac{1}{l\lambda_m} \left[-\frac{f(t)}{m} + (1 + \lambda_m)g\theta(t) \right] \quad (2.8)$$

Onde as grandezas físicas utilizadas são discriminadas na Tabela 2.2:

Tabela 2.2 - Símbolos utilizados por Goodwin, Graebe e Salgado

Símbolo	Significado
$\theta(t)$	Ângulo da haste do pêndulo
$f(t)$	Força aplicada no carro
m	Massa da haste, com centro de massa na ponta
λ_m	(M/m)
M	Massa do Carro

2.1.1 Espaço de estados e função de transferência

Modelagem no espaço de estados consiste em determinar o valor da saída do sistema quando se conhece os valores das variáveis, chamadas variáveis de estado, em um determinado $t = t_0$ e os valores das entradas pra $t \geq t_0$, ou seja no presente e no futuro. De acordo com Dorf e Bishop (2011, p 163) as variáveis de estado “[...] descrevem a configuração atual de um sistema e pode ser usada para determinar a resposta no futuro, dadas as entradas e as equações que descrevem a dinâmica do modelo”.

Ogata (2003) deduz a função de transferência do sistema eliminando \ddot{x} e $\ddot{\theta}$ das Equações (2.5, 2.6) realizando algumas manipulações matemáticas, obtendo as Equações (2.10, 2.11). Aplicando Laplace na Equação (2.10) encontra-se a função de transferência expressa pela Equação (2.12).

$$Ml\ddot{\theta} = (M + m)g\theta - u \quad (2.10)$$

$$M\ddot{x} = u - mg\theta \quad (2.11)$$

$$\frac{\Theta(s)}{-F(s)} = \frac{1}{Mls^2 - (M+m)g} \quad (2.12)$$

Na modelagem de espaço de estados Dorf e Bishop (2011, p 186) e Ogata seguem caminhos opostos quanto a seleção de suas variáveis de estado, invertendo as equações vetoriais-matriciais. A Tabela 2.3 explica as diferenças nas escolhas das variáveis de estado dentre os 3 livros. As equações (2.5) e (2.6) do modelo de Dorf e Bishop relacionam a aceleração angular

com o eixo y, e não x como o Ogata. Goodwin, Graebe e Salgado também relacionam a aceleração angular com o eixo x, porém chamando-o de $y(t)$.

Tabela 2.3-Diferença de escolha das variáveis de estado

Ogata		Dorf e Bishop		Goodwin, Graebe e Salgado	
X_1	θ	X_1	y	X_1	y
X_2	$\dot{\theta}$	X_2	\dot{y}	X_2	\dot{y}
X_3	x	X_3	θ	X_3	θ
X_4	\dot{x}	X_4	$\dot{\theta}$	X_4	$\dot{\theta}$

Ogata (2003, p. 74) apresenta as equações vetoriais-matriciais da modelagem no espaço de estados do sistema dinâmico de um pêndulo invertido, equações (2.13) e (2.14).

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{M+m}{Ml}g & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{m}{M}g & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{Ml} \\ 0 \\ \frac{1}{M} \end{bmatrix} u \quad (2.13)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (2.14)$$

Onde $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \theta \\ x \end{bmatrix}$.

Para a modelagem desenvolvida por Dorf e Bishop, as equações vetoriais-matriciais estão representadas nas equações (2.15) e (2.16).

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g}{l} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{Ml} \end{bmatrix} u \quad (2.15)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (2.16)$$

A modelagem proposta por Goodwin, Graebe e Salgado (2000, p. 55), utiliza as mesmas variáveis de estado. As equações vetoriais-matriciais são diferentes do modelo de Dorf e Bishop

devido à imensa possibilidades que a modelagem por Espaço de Estados permite. As equações (2.17) e (2.18) mostram o modelo proposto por Goodwin, Graebe e Salgado que apresenta uma única saída, o deslocamento em $y(t)$.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{m}{M}g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{M+m}{Ml}g & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{Ml} \end{bmatrix} u \quad (2.17)$$

$$[y(t)] = [1 \quad 0 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (2.18)$$

2.2 A plataforma LEGO *Mindstorms*

O kit de robótica LEGO *Mindstorms* EV3 proporciona uma experiência no estudo de várias áreas da engenharia a partir de uma estrutura física de fácil montagem e de uma central de processamento chamada de BRICK. Esta CPU possui diversas funcionalidades, como interação com diversos *Software* de programação, comunicação com sensores e atuadores além de *display* gráfico e de baterias.

Diversas pesquisas abordam modelos matemáticos utilizando-se a plataforma do LEGO. Dentre eles Kanada, Watanabe e Chen (2011, p. 136) justificam a importância da utilização do LEGO como uma alternativa de baixo custo e elevado grau de liberdade para a validação de modelos para teoria de controle. Outra forma de uso acadêmico/educacional do LEGO é defendido por Kushida et al. (2012, p. 5640) que consideram um sistema adequado para uso quando este é produzido em larga escala, possui um sistema atualizável e apresenta tecnologia de sistema embutido. Um sistema educacional também deve ser barato e seguro para facilitar o acesso por alunos baixa renda. Outros Autores como Kim (2010) e Papadimitriou (2007) também realizam seus estudos de teoria de controle utilizando a facilidade dos kits da LEGO.

O LEGO *Mindstorms* pode ser integrado com diversos *Software*, inclusive alguns de livre uso, Canale e Brunet (2013, p. 2549) descrevem uma grande quantidade de possibilidade de programação do BRICK, desde a opção do *Software* oferecido pelo LEGO, passando por *Software* complexos de engenharia como LabView e MATLAB, até possibilidades de

customizar o *Firmware* fabricando novos sensores e ferramentas de sistemas em tempo real como nxtOSEK e TOPPERS/JSP.

O processamento do Lego é feito por um processador ARM9 de 300MHz, 64 MB de memória RAM e 16MB de memória Flash interna com a possibilidade de expansão via cartão de memória SD.

2.2.1 LEGO Mindstorms EV3 motor CC

A obtenção dos parâmetros do motor CC é bastante trabalhosa, por isso este trabalho utiliza valores encontrados na literatura. Autores como Canale e Brunet (2013, p. 2551) e Kanada, Watanabe e Chen (2011, p. 136) oferecem os parâmetros referentes ao motor da versão NXT do LEGO *Mindstorms* de acordo com a Tabela 2.4.

Tabela 2.4 – Parametros do Motor DC do LEGO EV3

Variável	Valor	Unidade	Descrição
J	10^{-5}	Kgm ²	Momento de inércia
R _a	6,69	Ω	Resistência do Motor
K _b	0,468	Vs/rad	FCEM
K _m	0,317	Nm/A	Torque
b	0,0022	Nmrad/s	Coefficiente de atrito
V _a	8	V	Tensão de Alimentação
n	1		Fator de redução

Dalsager et al. (2006, p. 107) propôs um estudo bem detalhado do motor do LEGO NXT determinando valores para algumas constantes, onde podemos salientar o valores de 0,0021Kg.m² para o momento de inercia e o coeficiente de atrito como 0,0012N.m.s

A LegoEngineering oferece em seu site uma secção de suporte onde vários usuários podem oferecer seus projetos, estudos e trabalhos. Um destes colaboradores, Hurbain, realizou diversos testes com todos os motores do LEGO, obtendo os seguintes dados de acordo com a Tabela 2.5 e Tabela 2.6:

Tabela 2.5 – Características do Motor CC do LEGO EV3

Variável	Valor	Unidade
Peso	82	G
Teste em vazio (9V)		
Velocidade	175	rpm
Corrente	60	mA
Carga Máxima (rotor paralisado)		
Torque	43	N cm

Corrente	1.8	A
----------	-----	---

A Tabela 2.6 oferece dados referente a testes com variação da tensão aplicada ao motor, de acordo com a variação do sinal PWM que atua no motor.

Tabela 2.6 – Teste de variação de tensão do Motor DC do LEGO EV3

Tensão (V)	Torque (Nm)	Velocidade (RPM)	Corrente (A)	Potência Mecânica (W)	Potência Elétrica (W)	Eficiência (%)
4.5	17.3	24	0.69	0.43	3.10	14
6	17.3	51	0.69	0.92	4.14	22
7.5	17.3	78	0.69	1.41	5.17	27
9	17.3	105	0.69	1.90	6.21	31
10.5	17.3	132	0.69	2.39	7.24	33
12	17.3	153	0.69	2.77	8.28	33

2.2.2 Modelo Matemático de um Motor CC

O motor de CC é um dispositivo atuador que entrega energia para a carga, este motor converte corrente contínua (CC) em energia mecânica rotacional. Os motores CC são usados em inúmeras aplicações de controle, devido à características como alto torque, controle de velocidade, portabilidade e etc. Dorf e Bishop (2011, p. 72).

A modelagem matemática apresentada a seguir é sugerida por Dorf e Bishop (2011, p. 73) e Franklin, Powell e Emani-Naeini (2002, p. 55). A Figura 2.2 representa o diagrama de blocos de um motor de Corrente Contínua para um sistema de controle pela tensão de armadura.

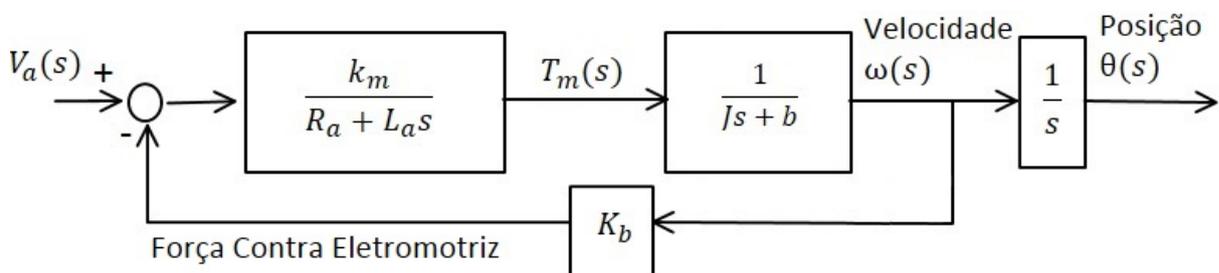


Figura 2.2 -Diagrama de Blocos de um Motor CC

Adaptado de FRANKLIN; POWELL; EMANI-NAEINI (2002, p. 55)

A Equação 2.19 representa a função de transferência do sistema entre a entrada $V_a(s)$ e a saída, considerando o distúrbio $T_d(s)$ desprezível.

$$G(s) = \frac{\Theta(s)}{V_a(s)} = \frac{K_m}{s[(R_a + L_a s)(Js + b) + K_b K_m]} \quad (2.19)$$

Entretanto, para a maioria dos motores CC a constante de tempo da armadura $\tau_a = L_a/R_a$ é suficientemente pequena para ser desconsiderada, resultando na Equação (2.20).

$$G(s) = \frac{\Theta(s)}{V_a(s)} = \frac{K_m}{s[(R_a)(Js + b) + K_b K_m]} \quad (2.20)$$

2.3 Controlador por Avanço de Fase

O pêndulo invertido é um sistema de fase não-mínima, naturalmente instável. O controlador por avanço de fase pode ser utilizado para controlar este tipo de sistema. Goodwin, Graebe e Salgado (2000, p. 168) aproximam a ação do compensador de avanço com uma ação de derivação, e o compensador de atraso com uma integração. O projeto de um controlador por avanço de fase pode ser feito por duas metodologias, utilizando o lugar das raízes ou por meio do diagrama de bode, este trabalho faz uso de parte das duas técnicas para encontrar um controlador eficiente.

De acordo com Dorf e Bishop (2011, p. 751) a resposta no domínio da frequência do compensador é adicionado à resposta frequência do sistema não-compensado. A análise inicial é feita pela plotagem do diagrama de Bode de $G(j\omega)H(j\omega)$. O diagrama pode ser analisado para determinar um local para os polos e zeros do compensador para satisfazer as exigências da resposta em frequência.

Dorf e Bishop oferecem uma metodologia para se escolher um controlador por avanço de fase $G_c(j\omega)$ de forma completa por diagrama de Bode, porém este trabalho prioriza o método do lugar das raízes de tal forma que o uso do diagrama de Bode será reduzido. Dorf e Bishop (2011, p. 752) sugerem a plotagem do sistema não compensado com um ganho escolhido para permitir um erro de sistema permanente aceitável, então a margem de ganho é analisada para saber se o sistema satisfaz ou não as especificações. Na situação em que a margem de ganho não é suficiente então um controlador por avanço de fase pode ser adicionado em uma localização específica. Este trabalho se atem ao uso desta técnica apenas para aferição da margem de ganho.

Para projetar um controlador por avanço de fase utilizando a metodologia do diagrama do lugar geométrico das raízes, é necessário conhecer a influência que polos e zeros tem sobre o sistema.

2.3.1 Efeitos da adição de pólos

De acordo com Ogata (2003, p. 345), o lugar das raízes é deslocado para a direita quando se tem a adição de um polo, tendendo a diminuir a estabilidade do sistema e retardando a acomodação da resposta. O controlador integral adiciona um polo na origem tornando o sistema menos estável.

2.3.2 Efeitos da adição de zeros

A adição de um zero no sistema, de acordo com Ogata (2003, p. 345), significa a introdução de um controle derivativo no sistema, o efeito deste controle é o aumento na rapidez da resposta transitória, pois o lugar das raízes é puxado para a esquerda, aumentando a estabilidade do sistema.

2.4 O projeto de um controlador por avanço de fase por meio do lugar das raízes

O controlador por avanço de fase pode ser implementado seguindo várias abordagens, tais como: digital ou contínuo, por meio de circuitos com amplificadores operacionais, circuitos RC ou por sistemas mecânicos mola-amortecedor. Ogata (2003, p. 347) sugere a função de transferência de um compensador por avanço de fase utilizando-se um circuito RC de acordo com a Equação (2.21). Dorf e Bishop (2011, p 757) propõe o modelo de um controlador semelhante representado pela Equação (2.22).

$$G_c(k) = K_c \frac{s + \frac{1}{T}}{s + \frac{1}{\alpha T}} \quad (2.21)$$

$$G_c(k) = \frac{s+z}{s+p} \quad (2.22)$$

O controlador por avanço se caracteriza pela adição de um polo e um zero à esquerda dos polos e zeros originais do sistema. O ganho K_c é projetado para atender requisitos do sistema (requisito de ganho de malha aberta), como velocidade e tempo de acomodação. T e α são determinados com base na deficiência angular. Em situações que o polo e o zero do controlador fiquem sobre

o eixo real, teremos uma equivalência de $1/T = z$ e $1/\alpha T = p$ igualando assim os dois modelos.

Ogata (2003, p. 347) e Dorf e Bishop (2011, p. 758) listam alguns passos para se obter o controlador por avanço de fase baseado em lugar das raízes para se controlar sistemas instáveis (exemplo o pêndulo invertido) ou estáveis com características de resposta transitória indesejáveis. Para um sistema onde não é necessário calcular a deficiência angular ϕ (polos e zeros do sistema sobre o eixo real) tem-se os seguintes passos para cálculo de controlador por avanço de fase:

- A- A partir das especificações de desempenho determina-se a posição dos polos a malha fechada dominantes.
- B- Traçar o diagrama do lugar geométrico das raízes, selecionar o lugar do zero do controlador no lugar desejado ou à esquerda dos dois primeiros polos. A localização do polo deve garantir um ângulo de 180° .
- C- Admite-se que o compensador por avanço de fase $G_c(s)$ seja representado como nas equações (2.21) e (2.22).
- D- Conferir o ganho total do sistema, e analisar a resposta do sistema, caso não alcance a dinâmica desejada, o processo deve ser refeito com novas posições para o polo e o zero do controlador.

2.5 A lógica *Fuzzy*

A lógica *Fuzzy* ou difusa pode ser utilizada em diversos ramos da engenharia e do dia a dia, como por exemplo controle de ar condicionado, sistemas bancários para classificar um cliente como apto ou não para um empréstimo, controle de guindastes em portos e etc. Diversas pesquisas sobre técnicas de controle abordam a estabilização de um pêndulo invertido. Os trabalhos de Miranda et al. (2003) e de Filho et al. (2013) estudam o funcionamento de um controlador *Fuzzy* para estabilizar pêndulos invertidos. Outros estudos abordam a comparação entre os controladores clássicos e a lógica nebulosa (Morais et al., 2005). O artigo de Silva et al. (2013) realiza a comparação dos controladores PID e *Fuzzy* com base em experimentos realizado com um kit de robótica LEGO *Mindstorms* NXT, onde um potenciômetro é adaptado a uma plataforma experimental para medir o ângulo do pêndulo.

A lógica difusa surge para suprir a necessidade de se expressar algumas situações em que uma lógica binária não consegue representar. Os conjuntos *Fuzzy* foram inicialmente propostos por Zadeh (1965) permitindo responder a perguntas como “o copo está pela metade”.

2.5.1 Variáveis Linguísticas.

As variáveis Linguísticas de acordo com Miranda, Junior e Kronbauer (2003 p. 12), são as palavras utilizadas pra representar de modo impreciso e linguístico as características mensuráveis de um problema. São permitidas como valores somente expressões linguísticas como por exemplo: alto, grande, pouco, quente, longe, muito longe, rápido e etc. Variáveis linguísticas podem assumir valores distintos ao longo de um universo de valores. Para isso utilizamos partições *Fuzzy* para dar nome a setores deste universo. Por exemplo a renda de uma pessoa pode ser considera “baixa” para valores inferiores a R\$800,00. Para valores entre R\$800,00 e R\$2500,00 a partição *Fuzzy* utilizada é “média”. Assim, classificamos como “alta” rendas entre R\$2500,00 até R\$7000,00 e como “muito alta” para rendas superiores a estes valores.

2.5.2 Função de Pertinência.

De acordo com Passino e Yurkovich (1998 p. 32) as funções de pertinência são utilizadas para quantificar os significados dos valores linguísticos (partição *Fuzzy*), ou seja um conjunto nebuloso tem a sua função de pertinência que é a representação do grau de pertinência (valor entre 0 e 1) de cada entrada.

Essa curva pode assumir diversas formas, que podem ser escolhidas para melhor representar a variável de entrada de um processo a ser controlado. As mais comuns são: triangular (Figura 2.3), trapezoidal (Figura 2.4) e Gaussiana (Figura 2.5).

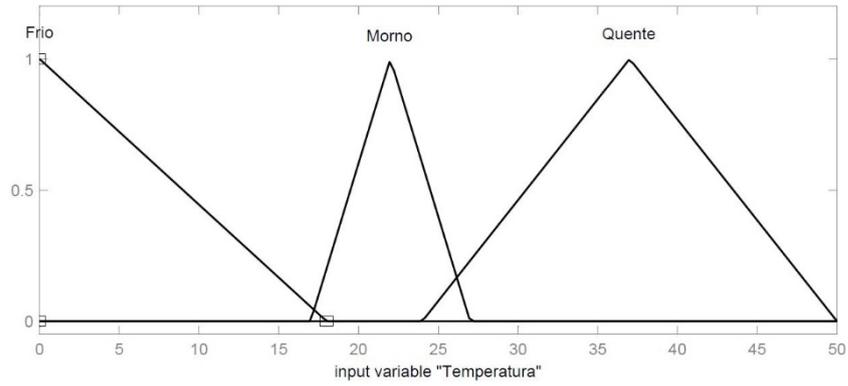


Figura 2.3-Função de Pertinência triangular

Uma função de pertinência triangular pode assumir várias formas, pois não é necessário utilizar um tipo fixo de triângulo como isósceles ou retângulo. Um mesmo valor como por exemplo 25°C pode pertencer aos valores Morno e Quente.

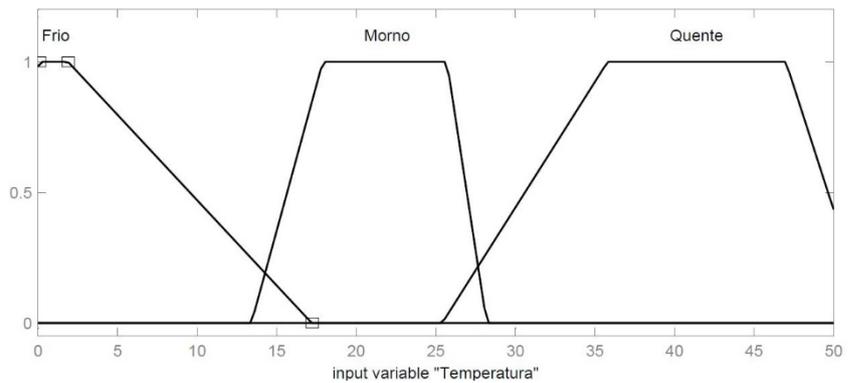


Figura 2.4- Função de Pertinência trapezoidal

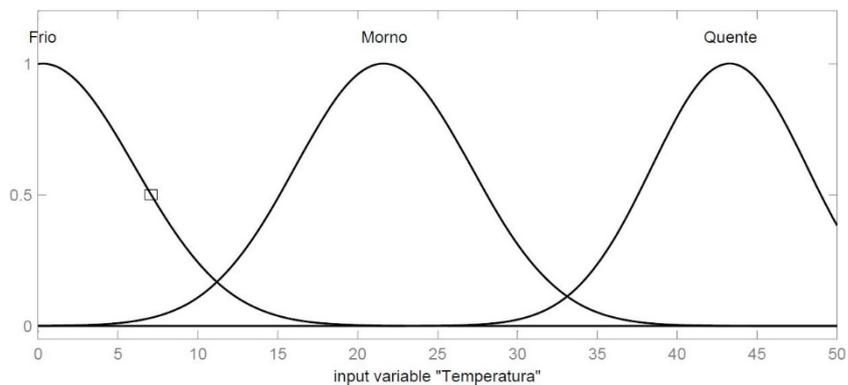


Figura 2.5- Função de Pertinência Gaussiana

2.5.3 Fuzificação e Inferência.

Um processo importante na construção de um controlador nebuloso é a Fuzificação, descrita por Sandri e Correa (1999, p. c077) como “A interface de Fuzificação faz a identificação dos valores das variáveis de entrada, as quais caracterizam o estado do sistema (variáveis de estado), e as normaliza em um universo de discurso padronizado.” Desta forma um o valor de uma variável de entrada já fuzificado pode ser inferido.

A inferência é um processo de grande importante na lógica *Fuzzy*. As FISs (*Fuzzy inference systems*) são tratadas por Sivanandam, Sumathi e Deepa (2007, p.118) como as responsáveis por criar regras e o embasamento teórico para que as decisões sejam tomadas, ou seja as FISs utilizam as regras *Fuzzy IF-THEN* e os conectores *and* e *or* para tomar decisões. O método de tomada de decisões utilizado nesse trabalho será o Mamdani considerado o método mais empregado atualmente.

De acordo com Sivanandam, Sumathi e Deepa (2007, p.120) o método de Mamdani foi proposto por Mamdani em 1975 como uma tentativa de controlar um motor à vapor e um boiler sintetizando um conjunto de variáveis linguísticas disponibilizada por um operador experiente, formando uma base de conhecimento.

A base de conhecimento armazena todas as funções de pertinência das variáveis do sistema, e os conjuntos de regras que associam as variáveis linguísticas de entrada com as de saída. Filho et al. (2013, p. 5) descreve a base de conhecimento como sendo a soma da base de dados com a base de regras. A base de dados contém a definição dos conjuntos *Fuzzy* e a base de regras é responsável por compilar as regras *Fuzzy*. Estas regras, juntamente com os dados de entrada, são processados pelo FIS, o qual infere as ações de controle de acordo com o estado do sistema, aplicando o operador de implicação. O FIS pode ser dividido em quatro etapas de acordo com Sandri e Correa (1999, p. c078).

- Verificação do grau de compatibilidade entre os fatos e as cláusulas das regras.
- Determinação do grau de compatibilidade global da premissa de cada regra;
- Determinação do valor da conclusão, em função do grau de compatibilidade da regra com os dados e a ação de controle constante na conclusão (precisa ou não)
- Agregação dos valores obtidos como conclusão nas várias regras, obtendo-se uma ação de controle global.

Um exemplo básico da aplicação do método Mamdani pode ser analisado na Figura 2.6

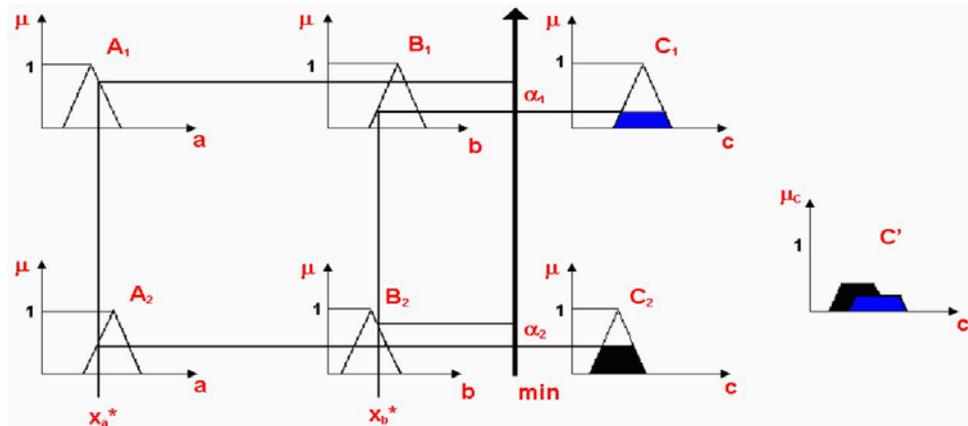


Figura 2.6- Método de Mamdani.

Fonte: SANDRI; CORREA (1999, p. c080)

2.5.4 Defuzificação

A Defuzificação é o processo que consiste na transformação do resultado da inferência em um valor numérico, pois o FIS produz uma quantidade de conjuntos *Fuzzy* de saída igual ao número de regras contidas na base de conhecimento. Dentre os métodos de defuzificação abordados na literatura, tem-se: Centro de área ou Centro de gravidade, Centro de somas, Centro de área máxima, Meio do máximo, Bisector, entre outros. (Filho et al. 2013, p. 6).

Além dos métodos citados, Sandri e Correa (1999, p. c079) evidenciam o uso de três:

- Primeiro Máximo (SOM): Encontra o valor de saída por meio do ponto em que o grau de pertinência da distribuição da ação de controle atinge o primeiro valor máximo.
- Método da Média dos Máximos (MOM): Encontra o ponto médio entre os valores que têm o maior grau de pertinência inferido pelas regras;
- Método do Centro da Área (COA): O valor de saída é o centro de gravidade da função de distribuição de possibilidade da ação de controle.

Uma representação matemática do Centro da Área (COA) e do Método da Média dos Máximos (MOM) é proposta por Sivanandam, Sumathi e Deepa (2007, p.122) respectivamente de acordo com as equações 2.23 e 2.24

$$Z = \frac{\sum_{j=1}^q Z_j u_c(Z_j)}{\sum_{j=1}^q u_c(Z_j)} \quad (2.23)$$

Onde: z é o centro da Área e u_c é a pertinência na classe c no valor z_j .

$$z = \sum_{j=1}^l \frac{z_j}{l} \quad (2.24)$$

Onde: z é a média dos máximos, z_j é o ponto no qual a função pertinência é máxima, e l é a quantidade de vezes que a distribuição alcança o nível máximo. A Figura 2.7 representa como várias opções de métodos de Defuzificação respondem ao mesmo projeto de controlador.

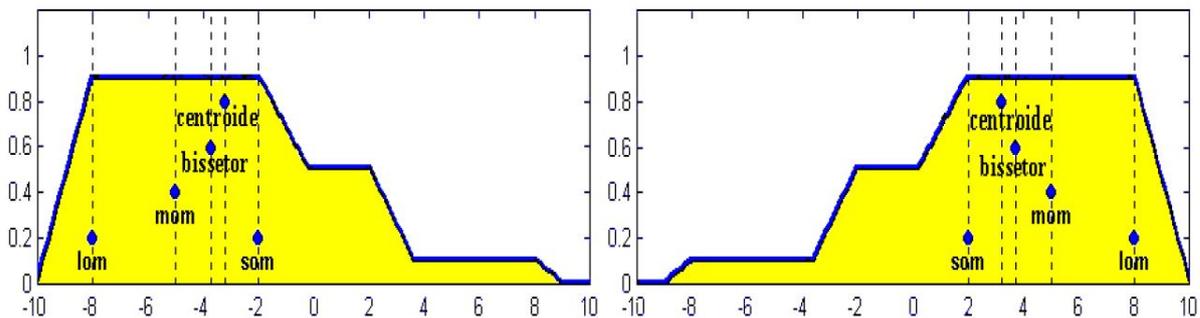


Figura 2.7- Exemplos de respostas para diferentes métodos de Defuzificação

Fonte: SANDRI; CORREA (1999, p. c082)

2.6 Controlador Nebuloso.

De acordo com Sivanandam, Sumathi e Deepa (2007, p.121) o controlador nebuloso pode ser dividido em seis etapas. A primeira cria o conjunto de regras *Fuzzy*, utilizando as regras *IF-THEN* e os conectores *and* e *or*; A segunda é a Fuzificação para mapear as entradas e transformá-las em variáveis linguísticas; A terceira é a combinação das entradas já fuzificadas; A quarta e quinta etapas consistem em analisar a força de cada regra e fazer a inferência obtendo uma distribuição de saída; A última etapa é a Defuzificação. A estrutura deste controlador pode ser analisado na Figura 2.8.

De acordo com Filho et al. (2013, p. 6) um controlador *Fuzzy* é robusto em relação a variação de parâmetros, não havendo a necessidade de escrever uma regra para cada situação possível. Isto decorre do fato da variável linguística ser definida por uma função de pertinência que cobre uma faixa de valores das variáveis de entrada.

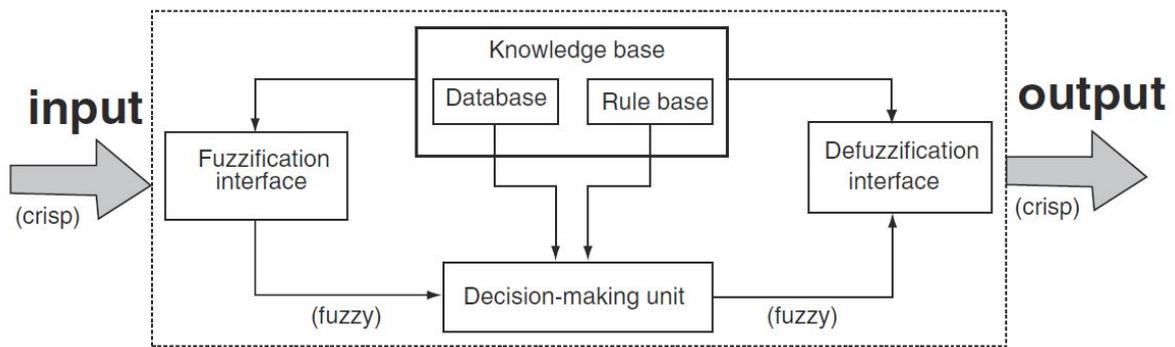


Figura 2.8- Diagrama de blocos de um controlador nebuloso.

Fonte: SIVANANDAM; SUMATHI; DEEPA (2007, p.119)

2.6.1 A simulação em tempo real

Um dos empecilhos para que os controladores *Fuzzy* não sejam utilizados em 100% das malhas de controle, é uma demanda maior por processamento comparado com os controladores clássicos. Passino e Yurkovich (1998, p. 97) exemplificam a quantidade de regras que um controlador *Fuzzy* possui como sendo uma função exponencial. Para um controlador de 2 entradas e 11 funções de pertinência, o número de regras é $11^2=121$. A maneira proposta para se reduzir o tempo de processamento é evitar que mais de duas regras sejam acionadas ao mesmo tempo.

Para uma simulação em um Hardware de baixa capacidade de processamento como o caso do LEGO é essencial minimizar a necessidade de processamento e armazenamento de variáveis. Porém o uso de métodos adaptativos (*toolbox Fuzzy*) para sintonia das funções de pertinência acaba reduzindo a margem de trabalho para que se possa economizar na demanda por processamento e memória.

2.6.2 Controladores Fuzzy para um pêndulo invertido

Passino e Yurkovich (1998) abordam vários aspectos referentes ao projeto de um controlador *Fuzzy* para o controle de um pêndulo invertido, este trabalho seguirá alguns conceitos abordados no livro apresentados nesta secção.

De acordo com Passino e Yurkovich (1998, p. 26) o objetivo do projeto do controlador nebuloso é automatizar o processo de um sistema controlado por humanos experientes. Como a única ação disponível (variável de saída) é movimentar o carro do pêndulo para um lado ou para o

outro e o único sensor disponível mede posição, a malha de controle proposta se encontra na Figura 2.9. Onde r é a referência, normalmente 0° , e o erro do ângulo, u o sinal de controle, e y o sinal de saída posição angular.

As variáveis linguísticas deste modelo foram escolhidas para facilitar o entendimento da posição da haste (erro da posição), da velocidade do pêndulo (variação do erro) e da força que atua no sistema. Sendo assim, são proposto cinco valores: MN, PN, Z, PP, MP (muito negativo, pouco negativo, zero, pouco positivo, muito positivo).

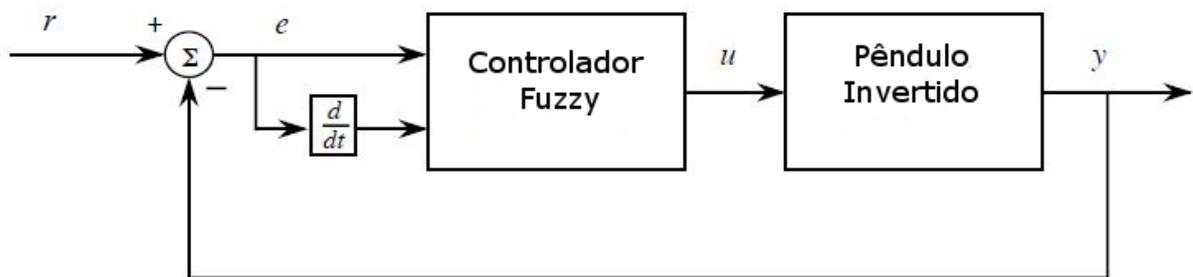


Figura 2.9- Malha de controle para um pêndulo invertido utilizando um controlador Fuzzy

Adaptado de PASSINO; YURKOVICH (1998, p. 27)

Passino e Yurkovich (1998, p. 29) descrevem algumas configurações possíveis de comportamento para o pêndulo invertido:

- A frase “erro é MP” pode representar a situação onde o pêndulo está em ângulo significativamente para a *esquerda* da vertical.
- A frase “erro é PN” pode representar a situação onde o pêndulo está um pouco para a *direita* da vertical, mas não muito perto para justificar sua posição como “zero” e nem muito longe para justificar o uso de “MN”.
- A frase “erro é Z” pode representar a situação onde o pêndulo está muito perto da posição vertical. (Uma quantificação linguística não é muito precisa uma vez que estamos esperando qualquer valor de erro perto de $e(t) = 0$ como sendo “Z”).
- A frase “erro é MP e velocidade é PP” pode representar a situação onde o pêndulo está à *esquerda* da vertical e desde que $\frac{dy}{dx} < 0$, o pêndulo está se afastando da posição vertical.

- A frase “erro é PN e velocidade é PP” pode representar a situação onde o pêndulo está um pouco a *direita* da vertical e como $\frac{dy}{dx} < 0$, o pêndulo está se movendo em direção a posição vertical.

A base de regras do modelo é muito importante para garantir o funcionamento do pêndulo utilizando uma quantidade de regras que o torna estável e com um custo computacional adequado com as limitações de *Hardware* a ser utilizado. É possível reduzir o número de regras reduzindo o número de valores linguísticos para velocidade, por exemplo, ou para a variável de saída.

Para ilustrar a base de regras Passino e Yurkovich (1998, p. 30) exemplificam 3 possíveis regras de controle do pêndulo invertido utilizando um controlador difuso, as representações das regras estão disponíveis na Figura 2.10.

A- *Se erro é MN and velocidade é MN Then força é MP*

Representado na Figura 2.10 (a), o pêndulo tem um ângulo positivo grande e está se movendo no sentido horário, logo para controlar devemos aplicar uma força grande no sentido contrário.

B- *Se erro é Z and velocidade é PP Then força é PN*

Representado na Figura 2.10 (b), pêndulo tem um ângulo de quase 0° com a vertical e está se movendo no sentido anti-horário, então tem que ser aplicada uma pequena força pra a esquerda para agir contra o movimento, se aplicarmos uma força para a direita pode causar um *overshoot*.

C- *Se erro é MP and velocidade é PN Then força é PN*

Representado na Figura 2.10 (c), o pêndulo está muito para a esquerda e se movendo no sentido anti-horário, deve-se aplicar uma força pequena no sentido negativo, esquerda, mas não uma muito forte desde que o pêndulo já está se movendo no sentido correto.

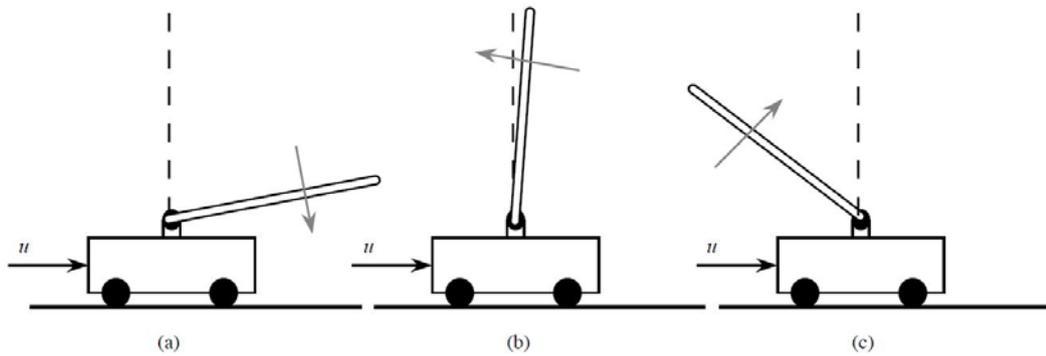


Figura 2.10- Exemplo de funcionamento da base de regras

Fonte: PASSINO; YURKOVICH (1998)

Para completar o controlador *Fuzzy* a Figura 2.11 representa as funções de pertinência utilizada por Passino e Yurkovich (1998, p. 36), onde *neglarge* = MN, *negsmall* = PN, *possmall* = PP e *poslarge* = MP. Na Tabela 2.7 podemos observar todo o conjunto de regras proposto para se controlar o pêndulo invertido.

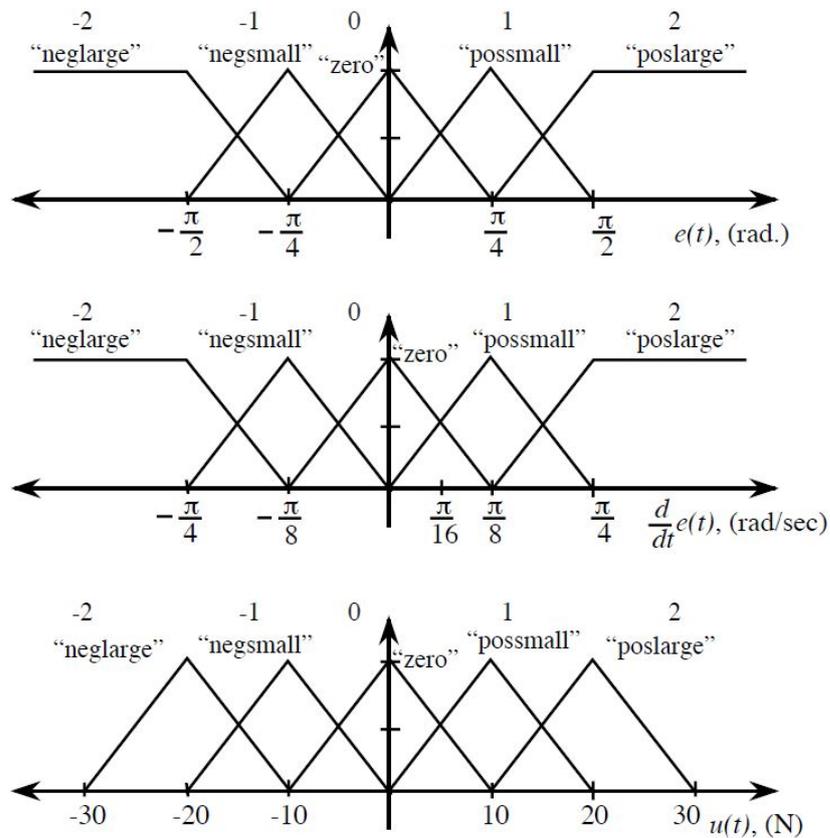


Figura 2.11-Funções de pertinência de Passino e Yurkovich

Fonte: PASSINO; YURKOVICH (1998)

Tabela 2.7- Base de regras do controlador proposto por Passino e Yurkovich

Força u		Velocidade \dot{e}				
		MN	PN	Z	PP	MP
Erro e	MN	MP	MP	MP	PP	Z
	PN	MP	MP	PP	Z	PN
	Z	MP	PP	Z	PN	MN
	PP	PP	Z	PN	MN	MN
	MP	Z	PN	MN	MN	MN

Uma alternativa ao simples controlador *Fuzzy*, é a opção de mesclar o seu funcionamento com a de um controlador PID. Yadav e Gaur (2011) propõe um sistema de controle para um pêndulo invertido misto, com variações de *Fuzzy* com os controladores: PD, PD+I e PD+PID .

3 MATERIAIS E MÉTODOS

Baseado nas informações contidas no Capítulo 2, apresentam-se duas propostas de sistemas de controle para o pêndulo invertido, o compensador por avanço de fase e o controlador por lógica difusa (*Fuzzy*). O desenvolvimento deste projeto foi formulado em cinco etapas: a primeira etapa envolve a montagem da estrutura física do pêndulo por meio das peças do kit LEGO *Mindstorms* EV3, pesagem do corpo do carro e da haste, e medição do comprimento da haste e do diâmetro das rodas; a segunda etapa o estudo da dinâmica do pêndulo invertido, juntamente com técnicas de controle; a terceira etapa simulação e experimentos do sistema de controle por avanço de fase; a quarta etapa estudo da lógica nebulosa; a quinta etapa a realização de simulações e experimentos do controlador nebuloso e comparação de resultados.

No projeto, utiliza-se o kit LEGO *Mindstorms* pela facilidade em se montar a estrutura física, pela disponibilidade de um sistema pronto e completo com unidade de processamento, sensores, atuadores, protocolos de comunicação, *firmware*, e pelo *toolbox* para MATLAB. As rodas utilizadas nos ensaios são os modelos 81.6x15 *Motorcycle* disponível em versões antigos do kit LEGO e que atualmente podem ser compradas separadamente em sites especializados na venda de peças e sensores para o LEGO *Mindstorms*.

3.1 Desenvolvimento da estrutura física

Com o auxílio da peças de fácil montagem do kit LEGO desenvolve-se uma estrutura rígida buscando uma solidez da parte inferior, evitando-se a dobra da mesma, e uma haste firme e inflexível que não se desloque lateralmente para permitir apenas o movimento desejado de rotação da haste em torno de um ponto fixo.

Para utilizar as equações 2.2, 2.3 e 2.4, o centro de massa da haste foi deslocado para a extremidade superior da haste. Assim, posiciona-se duas rodas de um Kit LEGO e o *Gyro Sensor* (sensor que detecta variações angulares por meio da força da gravidade) no topo da haste.

A Figura 3.1 representa a montagem da base do LEGO com os motores e os eixos onde as rodas são posicionadas, no centro da imagem o sistema de junção entre as bases onde cada motor foi montada proporciona uma rigidez à estrutura evitando-se a sua flambagem. Na Figura 3.2

demonstra-se a base da haste, na cor branca e os suportes laterais pretos que evitam seu deslocamento lateral. O pêndulo invertido completo pode ser visto na Figura 3.3



Figura 3.1 -Base do LEGO com os motores

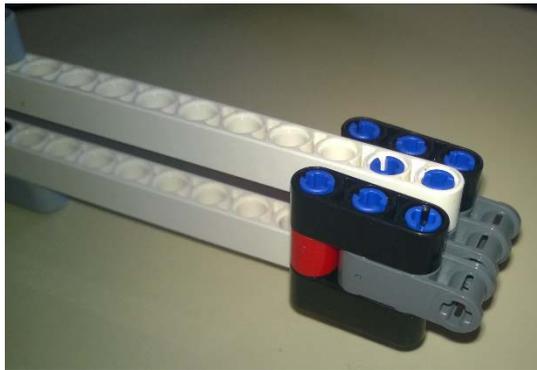


Figura 3.2 -Base da Haste



Figura 3.3 -Montagem completa do LEGO

Na Tabela 3.1 representa-se as grandezas físicas do pêndulo invertido necessárias na modelagem da função de transferência do pêndulo e dos atuadores. As medidas de massa foram realizadas com uma balança com precisão de ($\pm 0,5g$), o diâmetro das rodas mensurados com um paquímetro e o comprimento a haste com uma régua comum de 30 cm.

Tabela 3.1-Grandezas Físicas do LEGO

Medida	Valor	Unidade
Peso do Carrinho	0,715	Kg
Peso da Haste	0,155	Kg
Comprimento da Haste	0,28	m
Diâmetro das rodas	0,085	m

3.2 Modelagem do Pêndulo Invertido.

Os dados adquiridos na secção 3.1 foram aplicados na função de transferência (2.12), resultando na função da planta do pêndulo invertido, equação 3.1.

$$\frac{\Theta(s)}{F(s)} = \frac{-1}{0.2002s^2 - 8.526} \quad (3.1)$$

Aplicando-se os valores das constantes do motor conforme a secção 2.5.1 no diagrama de blocos da Figura 2.2 podemos formar o diagrama de blocos da Figura 3.4, onde temos a entrada de tensão que vem do controlador. As funções de transferência $t1$ e $t2$ representam o modelo matemático da secção 2.6 e os dados do motor contidos na secção 2.5.1, Tabela 2.4 utilizando J_m e B obtidos por Dalsager et al. (2006, p. 107). O torque do motor foi dividido pelo raio da roda procedendo na força que cada motor entrega, este valor é multiplicado pela quantidade de motores resultando na força aplicada na função de transferência da planta.

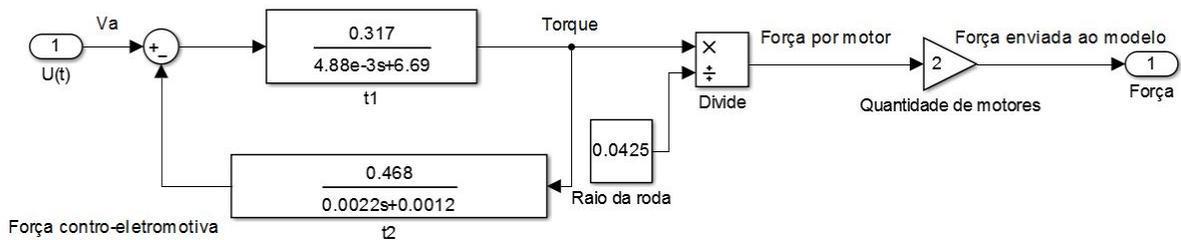


Figura 3.4- Modelo dos atuadores no Simulink.

A função de transferência referente ao diagrama de blocos da Figura 3.4 é representada na equação 3.2.

$$\frac{F(s)}{V_a(s)} = \frac{0.03133s+0.0179}{1.025*10^{-5}s^2+0.01405s + 0.1564} \quad (3.2)$$

A função de transferência completa em malha aberta do sistema do pêndulo invertido pode ser encontrado multiplicando-se as equações 3.1 e 3.2, resultando na equação 3.3

$$\frac{\Theta(s)}{V_a(s)} = \frac{-0.03133s-0.0179}{2.052*10^{-6}s^4+0.002814s^3+0.03122s^2-0.1198s-1.33} \quad (3.3)$$

3.2.1 Redução de ordem da função de transferência

A equação 3.3 demonstra a presença de um polo muito a esquerda no lugar das raízes (fase não mínima), além de ser um sistema de quarta ordem o que aumenta a complexidade do projeto de um controlador. Para isso as técnicas de redução de ordem podem ser muito eficiente quando se deseja trabalhar com entradas em baixas frequências.

A análise da redução de ordem de uma FT é mais simples quando a função de transferência está disposta na forma ZPK, ou seja pelos os zeros, os polos e o ganho. A equação 3.4 representa a equação 3.3 utilizando as funções *zpkdata* e *zpk* do Matlab.

$$\frac{\Theta(s)}{V_a(s)} = \frac{-15269(s+0.5417)}{(s+1360)(s+11.22)(s+6.526)(s-6.526)} \quad (3.4)$$

Para reduzir a ordem de uma função de transferência de forma mais simples, é necessário eliminar o polo menos significativo e dividir o ganho do sistema por $\sqrt{1^2 + 1360^2} \cong 1360$. Resultando na equação 3.5.

$$\frac{\Theta(s)}{V_a(s)} = \frac{-11.227 (s+0.5417)}{(s+11.22)(s+6.526)(s-6.526)} \quad (3.5)$$

Um método de avaliação do processo de redução de ordem é a plotagem do diagrama de bode dos dois sistemas, analisando o ganho e a fase de cada um de acordo com a frequência. Na Figura 3.5 representa-se o comparativo entre o desempenho do sistema original e o de ordem reduzida, devido à natureza do sinal de entrada do pêndulo invertido ser lento o uso do sistema de ordem reduzida é aceitável.

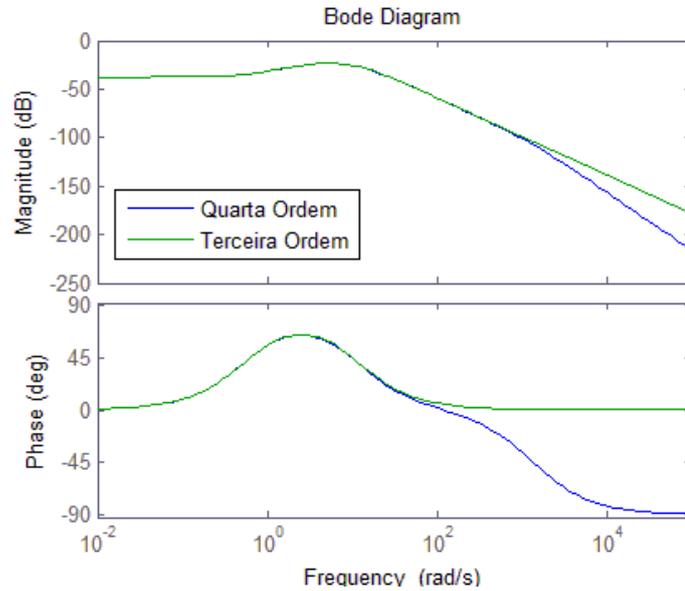


Figura 3.5-Diagrama de Bode comparativo entre o sistema completo e o reduzido

3.2.2 Análise da estabilidade

De acordo com Ogata (2003, p. 180). “A característica mais importante do comportamento dinâmico de um sistema de controle é a estabilidade absoluta, isto é, se o sistema é estável ou instável”. Um pêndulo invertido é instável pois a tendência é a divergência do ponto de operação, ou seja, cair. O pêndulo invertido é um sistema de fase não-mínima, naturalmente instável, por apresentar um polo no semiplano direito (+6.526). A Figura 3.6 representa o lugar geométrico das raízes do sistema reduzido comprovando a instabilidade do mesmo.

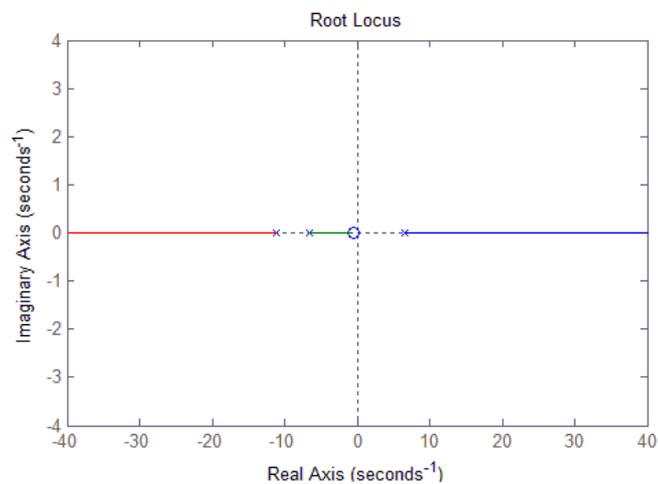


Figura 3.6-Lugar das Raízes do pêndulo invertido.

3.2.3 Projeto do controlador por avanço de fase

O controlador por avanço se caracteriza pela adição de um polo e um zero à esquerda dos polos e zeros originais do sistema. O ganho K_c é projetado para atender requisitos do sistema, como velocidade e tempo de acomodação. Os valores utilizados neste trabalho são destacados na Equação (3.6): polo em -25; zero em -22; ganho de -85. A combinação do controlador com o modelo resulta no Lugar das Raízes apresentado na Figura 3.7. Como pode ser observado, o sistema é estável a partir de um valor de ganho crítico ($K_c \leq -5$). O diagrama de Bode da Figura 3.8 mostra que o ganho de -85 do controlador oferece um ganho positivo em baixas frequências e negativo em altas frequências.

$$G_c(k) = -85 \frac{s+22}{s+25} \quad (3.6)$$

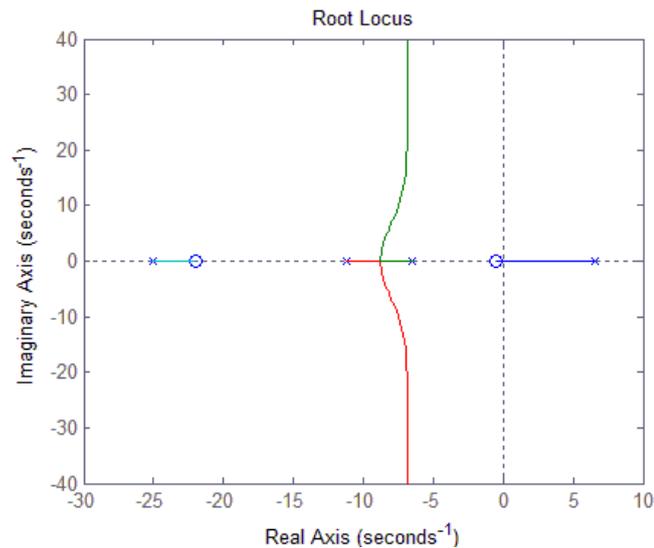


Figura 3.7-Lugar das raízes do sistema com o controlador por avanço de fase.

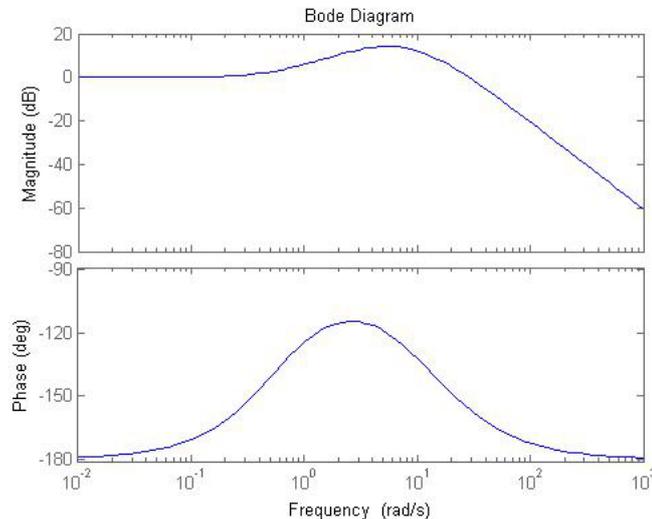


Figura 3.8- Diagrama de Bode do Sistema com o controlador por avanço de fase

3.2.4 Projeto do Controlador *Fuzzy*

O controlador nebuloso desta simulação possui as entradas conforme descrito na secção 2.8, para a saída iremos utilizar o valor de tensão de acionamento do motor de 8V. Desta forma a saída do controlador nebuloso e do avanço de fase utilizam o mesmo sinal V_a . A função de pertinência do erro da posição angular, da variação do erro de posição e da saída se encontram nas (Figura 3.9, Figura 3.10 e Figura 3.11) respectivamente.

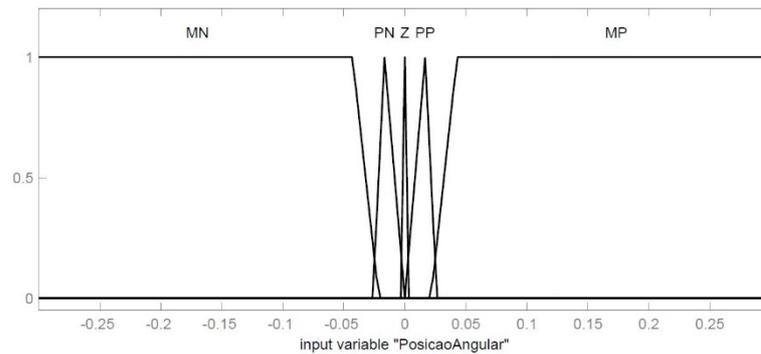


Figura 3.9- Função pertinência da posição para a simulação

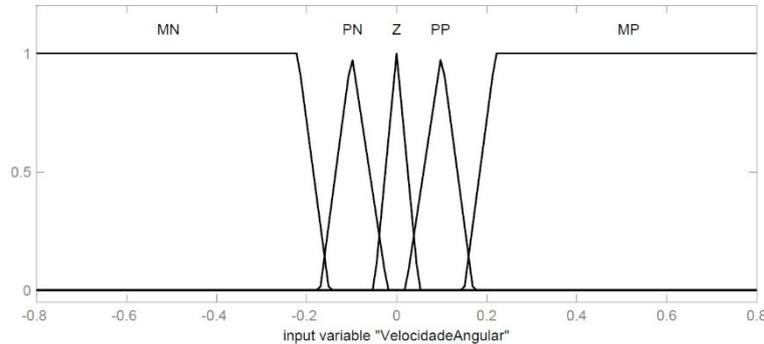


Figura 3.10- Função pertinência da velocidade para a simulação

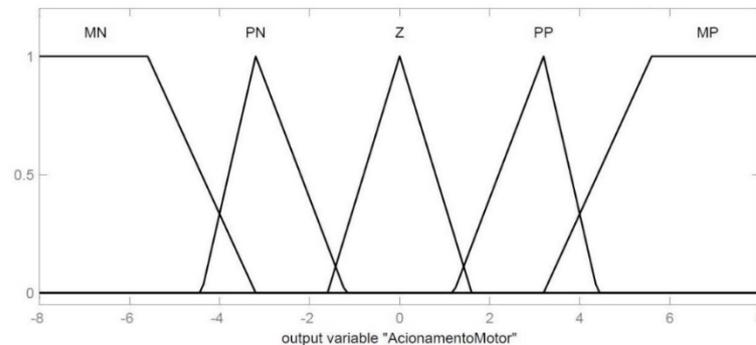


Figura 3.11- Função pertinência da saída para a simulação

A Tabela 3.2 mostra os limites de todas as funções de pertinência, suas variáveis linguísticas e o vetor com seus dados. Para uma variável linguística do tipo triangular, o primeiro e o último

valor representam as posições dos vértices das bases do triângulo, e o valor intermediário representa a projeção da posição do pico do triângulo sobre a sua base. Analogamente para os trapézios, o primeiro e o último valor representam os vértices das bases e os valores intermediários representam as projeções dos dois vértices superiores sobre a base do trapézio.

Tabela 3.2-Parâmetros das funções de pertinência

Função de Pertinência	Variáveis Linguísticas	Tipo	Vetor de dados
Erro da Posição	MN	Trapezoidal	[-0.4286 -0.3 -0.04286 -0.02143]
	PN	Triangular	[-0.02627 -0.0167 0]
	Z	Triangular	[-0.000432 0 0.000432]
	PP	Triangular	[0 0.01672 0.02627]
	MP	Trapezoidal	[0.02143 0.04286 0.3 0.4286]
Velocidade	MN	Trapezoidal	[-1 -0.8 -0.2199 -0.15]
	PN	Triangular	[-0.17 -0.1001 -0.02]
	Z	Triangular	[-0.05001 0 0.05]
	PP	Triangular	[0.02 0.1 0.17]
	MP	Trapezoidal	[0.15 0.2199 0.8 1]
Saída	MN	Trapezoidal	[-9.6 -8 -5.6 -3.2]
	PN	Triangular	[-4.4 -3.2 -1.2]
	Z	Triangular	[-1.6 0 1.6]
	PP	Triangular	[1.2 3.2 4.4]
	MP	Trapezoidal	[3.2 5.6 8 9.6]

Para esta aplicação o método de Defuzificação utilizado apresentou uma influência muito grande no tempo gasto para a inicialização e compilação do projeto no SIMULINK. Utilizando o método da bissetriz é necessário 30 segundos para terminar o processo, para o método do *centroid* (centro de massa) o mesmo conjunto gasta 105 segundos. A superfície de tomada de decisão para esta simulação utilizando o método *centroid* pode ser analisado na Figura 3.12.

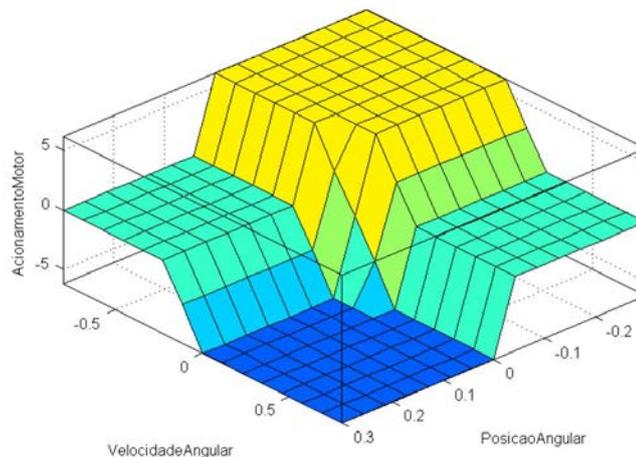


Figura 3.12-Superfície de tomada de decisão controlador Fuzzy para simulação

3.3 Simulações e Experimentos

Com o intuito de aferir a qualidade dos controladores projetado na secções 3.2.3 e 3.2.4, esta secção descreve todo o procedimento utilizado para verificar o desempenho de cada controlador.

Para manter um padrão entre simulações e experimentos, todos eles foram realizados durante quatro segundos. Um distúrbio (impulso aproximado por um degrau de curta duração) é aplicado via SIMULINK por meio de dois blocos de *step* no instante 0,5s com amplitude de 3° ($0,052\text{rad}$) com duração de 20ms, para que se possa analisar o regime transitório do sistema. São realizadas quatro verificações, duas para cada controlador, sendo uma simulação utilizando apenas o SIMULINK e um experimento prático utilizando o LEGO. As simulações são executadas em um computador com processador Intel i7 e 8GB de RAM utilizando a versão 2014a do MATLAB. Os experimentos são realizados com o mesmo computador, comunicando com o controlador do LEGO.

Para a versão 2014a do MATLAB, a única interface de comunicação disponível é por wireless e apenas o adaptador *wifi* da NETGEAR é compatível com a CPU do LEGO. A comunicação entre LEGO e MATLAB consome tempo de processamento do controlador embarcado, reduzindo o tempo disponível para processamento do controlador. (O apêndice A exemplifica um passo a passo e de como conectar o LEGO em uma rede *wifi* residencial).

Durante os experimentos, os controladores implementados no MATLAB enviam o sinal de controle para os motores por meio de blocos funcionais do *toolbox* do LEGO no SIMULINK. Com o objetivo de reduzir a velocidade de rotação dos motores, uma limitação de 60% é implementada para agilizar a comutação no sentido de rotação.

O software MATLAB é utilizado por proporcionar suporte educacional ao projeto, permitindo o estudo do modelo matemático, das técnicas de controle, métodos de ajuste, e a realização de simulações e experimentos.

3.3.1 Sistema de aquisição do ângulo da Haste.

O *Gyro Sensor* oferecido com o kit LEGO *Mindstorms EV3* tem como característica a capacidade de medir velocidades angulares a uma frequência de 1kHz. A Figura 3.13 representa o sistema de aquisição de dados, onde o *Gyro Sensor* informa um valor de velocidade angular,

em 100Hz. Este valor é convertido para um formato de dados do tipo *double* para que o MATLAB possa utiliza-lo em qualquer tipo de blocos. Um filtro passa baixa foi projeto com frequência de corte de 50Hz para inibir a ação de ruídos na leitura do sistema. Um bloco de ganho é responsável por converter as unidades de graus para radianos. Para se ter a posição angular do pêndulo um integrador é utilizado.

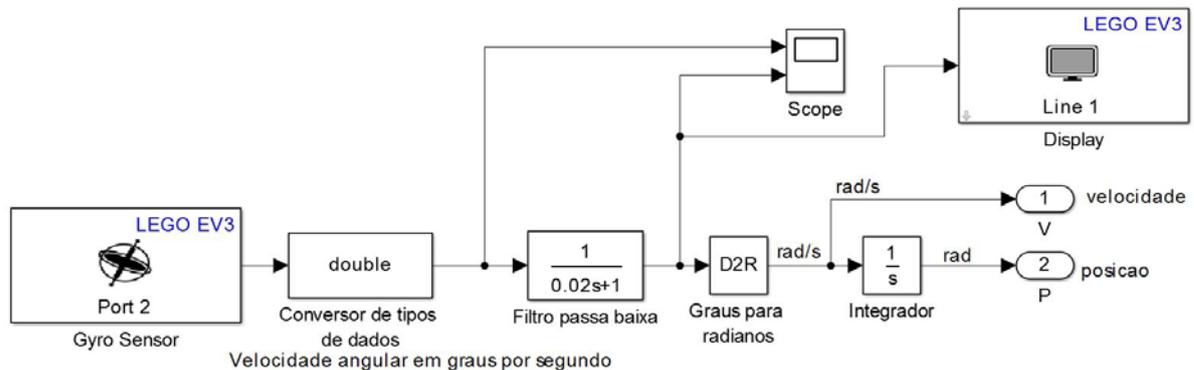


Figura 3.13-Sistema de aquisição do Ângulo da haste e tratamento de dados

Um *scope* e um *display* são utilizados para mostrar os valores da velocidade angular antes e depois do filtro, ajudando na sintonia de demais parâmetros do sistema.

3.3.2 Simulação do controlador por avanço de fase.

A simulação do pêndulo invertido, é o resultado da montagem do controlador por avanço de fase, do atuador, e da função de transferência da planta realimentada e com ponto de operação selecionado como zero rad (posição vertical para cima). Devido à ausência de um bloco de impulso no SIMULINK, foram utilizados dois blocos de *step* onde um aplica o degrau positivo e posteriormente o outro bloco insere um degrau negativo no sistema.

A Figura 3.14 demonstra o modelo completo da simulação teórica, onde o bloco “Modelagem do motor” internamente é representa o mesmo diagrama de blocos da Figura 3.4. A saída da planta é um valor de uma posição angular em radianos, para se obter a velocidade angular a título de comparação com a simulação no LEGO, deriva-se a posição e por meio de um multiplexador os dados são exibidos em um *scope*.

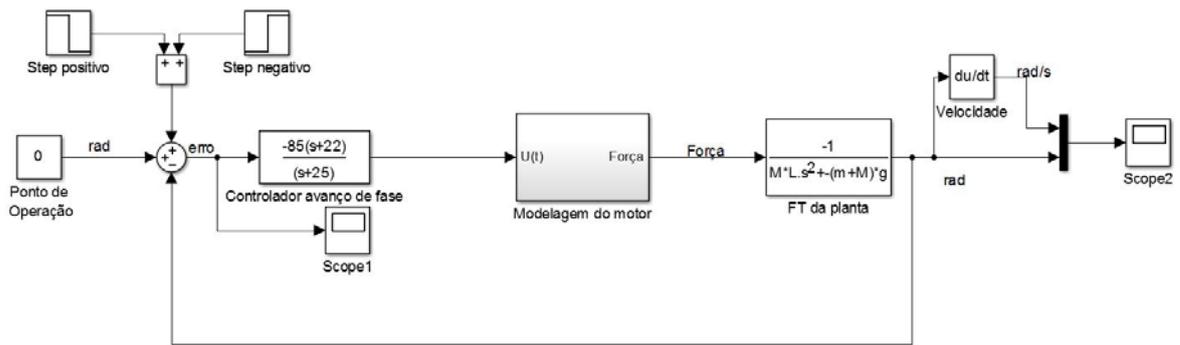


Figura 3.14-Diagrama de blocos da simulação do controlador por avanço de fase.

3.3.3 Experimento do controlador por avanço de fase.

Para compreender a ação do controle em um ambiente real, o ensaio da resposta do LEGO ao um impulso no MATLAB permite compreender a ação do controlador sobre os atuadores para manter o equilíbrio da haste. A Figura 3.15 representa o diagrama de blocos utilizado para a simulação, onde o valor da posição angular é obtida por meio do sistema descrito na Figura 3.13. O erro gerado no bloco de soma é aplicado na função do controlador por avanço de fase que gera o sinal para os motores. Um saturador limita esse sinal ao intervalo $[-100, 100]$ de acordo com as especificações do motor e do *toolbox* do LEGO para Matlab.

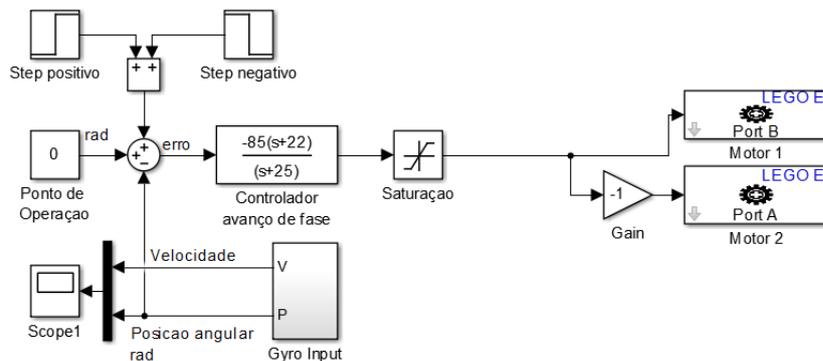


Figura 3.15-Diagrama de blocos do experimento do controlador por avanço de fase.

Os motores estão dispostos de costas um para o outro na montagem física do LEGO, por isso é necessário inverter o sentido de rotação de um dos motores para que ambos façam força na mesma direção. Os *scopes* são utilizados para visualizar posição e velocidade angular, além do sinal do erro e o sinal de controle antes e depois da saturação. Assim como no teste da secção 3.3.2 foram utilizados uma combinação de dois blocos *step* e o ponto de operação mantido em zero rad.

3.3.4 Simulação do controlador nebuloso.

Inicialmente é necessário montar a estrutura onde será inserido o bloco do SIMULINK para controladores por lógica *Fuzzy* de acordo com a Figura 2.9. Na Figura 3.16 ilustra-se o diagrama de blocos para a simulação do controlador *Fuzzy*, onde temos uma constante como referência, dois *steps* para simular um impulso, um bloco de derivada para gerar o erro de posição, o controlador *Fuzzy*, a planta e os atuadores modelados de acordo com a secção 3.2.

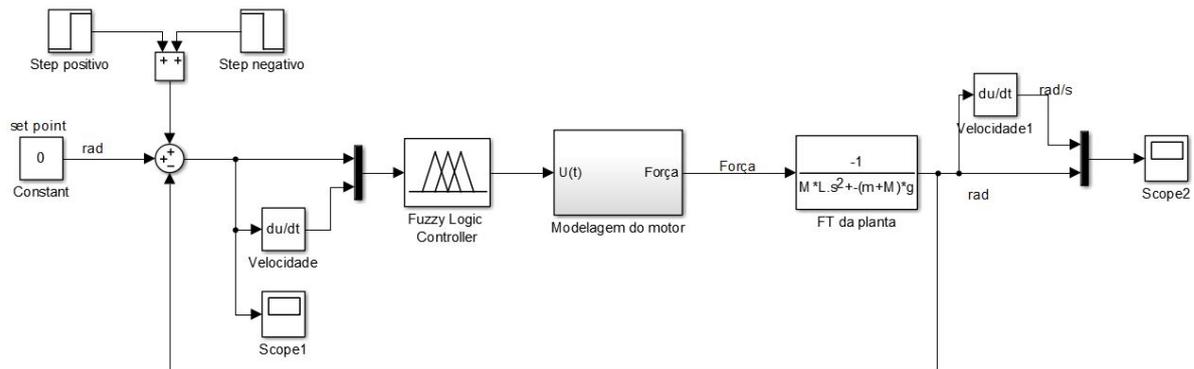


Figura 3.16-Diagrama de blocos da simulação do controlador *Fuzzy*

3.3.5 Experimento do controlador nebuloso.

O diagrama de blocos para o experimento é similar ao utilizado nos testes da secção 3.3.3, a diferenciação é feita pela utilização de um bloco derivador pra obtenção da velocidade de variação do erro e do bloco do controlador por lógica *Fuzzy*, como pode ser observado na Figura 3.17. Este controlador nebuloso se difere do utilizado na simulação em vários aspectos, inicialmente a função de pertinência da saída terá um universo de trabalho diferente, pois na simulação a modelagem dos atuadores é feita para o intervalo $[-8 \ 8]$ Volts, e como já citado anteriormente o bloco de acionamento dos motores do LEGO por meio do SIMULINK trabalha recebendo valores no intervalo $[-100 \ 100]$, ou seja porcentagem de velocidade em cada sentido. Na prática, o conjunto do LEGO tem uma grande dificuldade de inverter o sentido da rotação de um motor quando este alcança rotações medianas, por isso o intervalo utilizado será $[-60 \ 60]$.

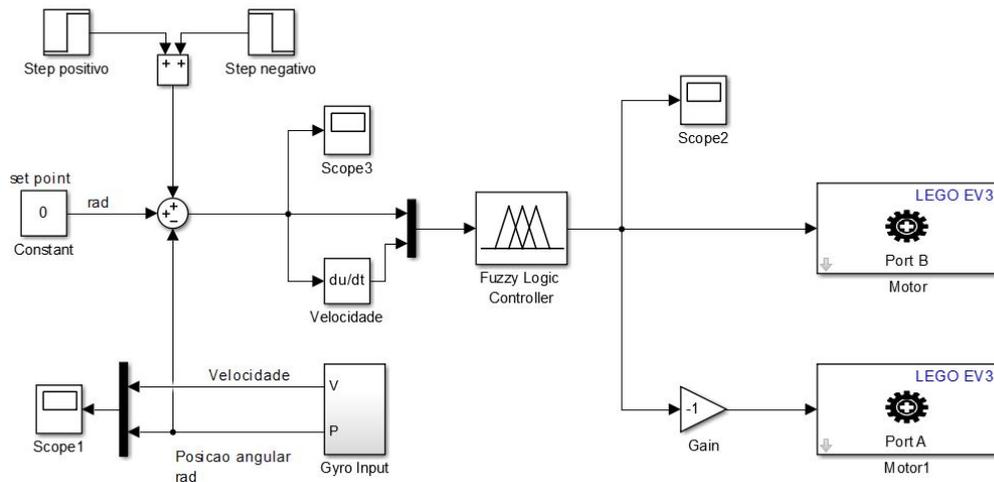


Figura 3.17- Diagrama de blocos do controlador nebuloso para o experimento

A maior dificuldade de utilizar este tipo de controlador em um *Hardware* embarcado é a sua capacidade de processar e armazenar todos os cálculos que o controlador nebuloso exige. Para reduzir a quantidade de regras do controlador, a quantidade de variáveis linguísticas para a função de pertinência da velocidade foi reduzida de cinco para três. Resultando em um controlador de 17 regras. Um segundo controlador será proposto de acordo com o artigo de Silva et al. (2013) onde um controlador *Fuzzy* para um pêndulo invertido foi implementado utilizando a plataforma LEGO NXT.

3.3.5.1 Controlador nebuloso de 17 regras

Para o controlador de 17 regras a função de pertinência da posição angular foi mantida a mesma, e a da velocidade (do erro) pode ser analisada na Figura 3.18

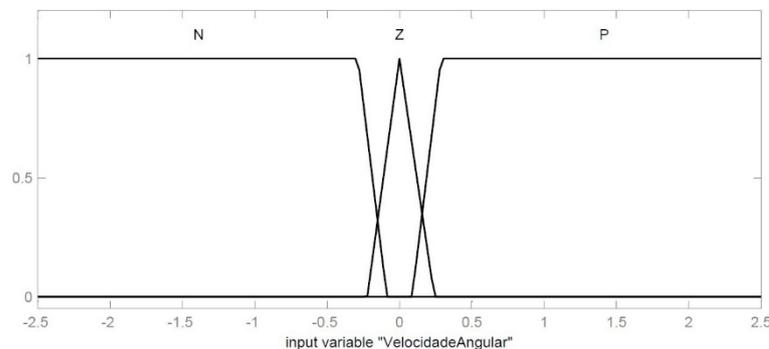


Figura 3.18- Função de pertinência para a variação do erro, experimento

A função de pertinência para a saída do controlador nebuloso pode ser analisado Figura 3.19, pra intensificar o acionamento do motor para as variáveis linguísticas PN e PP as suas formas foram alteradas para trapezoidal garantindo uma probabilidade maior do acionamento do motor

receber os valores +20 ou -20 que foram valores semelhantes aos utilizados pelo controlador clássico.

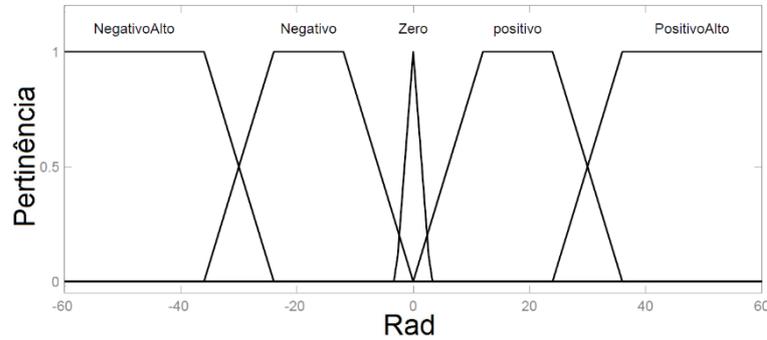


Figura 3.19-Função de pertinência do acionamento do motor, experimento

O conjunto de regras deste controlador é de 17, somando as 15 regras presentes na Tabela 3.3 com duas outras regras que foram adicionadas utilizando apenas a posição angular, são elas: IF posiçãoangular MN then AcionamentoMotor MP. IF posiçãoangular MP then AcionamentoMotor MN. A Tabela 3.4 representa os parâmetros das variáveis linguísticas para este controlador no mesmo padrão da Tabela 3.2.

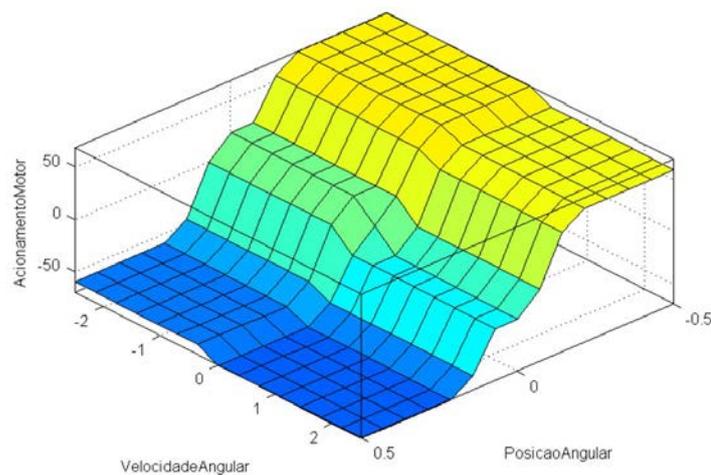
Tabela 3.3- Base de Regras do controlador nebuloso prático

Força u		Velocidade \dot{e}		
		N	Z	P
Erro e	MN	MP	MP	PP
	PN	PP	PP	Z
	Z	PP	Z	PN
	PP	Z	PN	PN
	MP	PN	MN	MN

A superfície de tomada de decisão para o controlador de 17 regras do experimento utilizando o método *centroid* pode ser analisado na Figura 3.20.

Tabela 3.4- Parâmetros das variáveis Linguísticas

Função de Pertinência	Variáveis Linguísticas	Tipo	Vetor de dados
Erro da Posição	MN	Trapezoidal	[-0.61 -0.55 -0.197 -0.1]
	PN	Triangular	[-0.2 -0.1 0]
	Z	Triangular	[-0.0132 0 0.0132]
	PP	Triangular	[0 0.1 0.2]
	MP	Trapezoidal	[0.1 0.197 0.571 1]
Vari. Posição	N	Trapezoidal	[-4.188 -3.209 -0.2876 -0.08745]
	Z	Triangular	[-0.2 0 0.2]
	P	Trapezoidal	[0.0875 0.2876 2.553 2.553]
Saída	MN	Trapezoidal	[-120 -100 -50 -25]
	PN	Trapezoidal	[-28 -25 -12.5 -10]
	Z	Triangular	[-12.1 0 12.5]
	PP	Trapezoidal	[10 12.49 25 28]
	MP	Trapezoidal	[25 50 100 120]

**Figura 3.20- Superfície de tomada de decisão do experimento do controlador nebuloso**

3.3.5.2 Controlador nebuloso de 9 regras

Uma outra proposta de controlador *Fuzzy* é apresentada a seguir, este segundo modelo visa economizar processamento ao máximo, desta forma serão usadas apenas nove regras. As duas variáveis linguísticas de entrada utilizadas empregam apenas três valores linguísticos e a de saída cinco. A Figura 3.21 mostra os três valores linguísticos do erro da posição, a Figura 3.22 representa a variação do erro da posição (velocidade) e finalmente a Figura 3.23 exibe a saída do controlador com suas cinco variáveis.

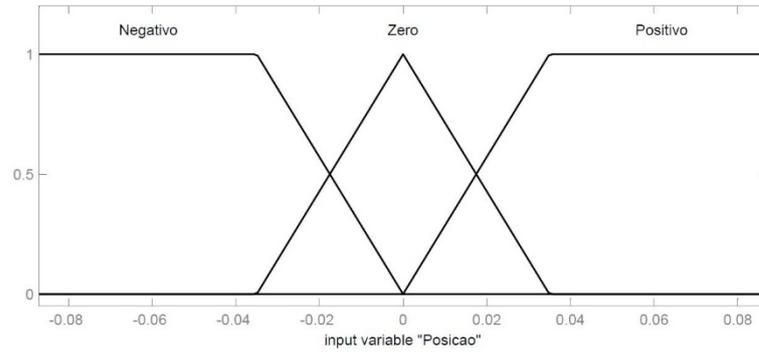


Figura 3.21- Função de pertinência da posição, para o controlador *Fuzzy* de 9 regras

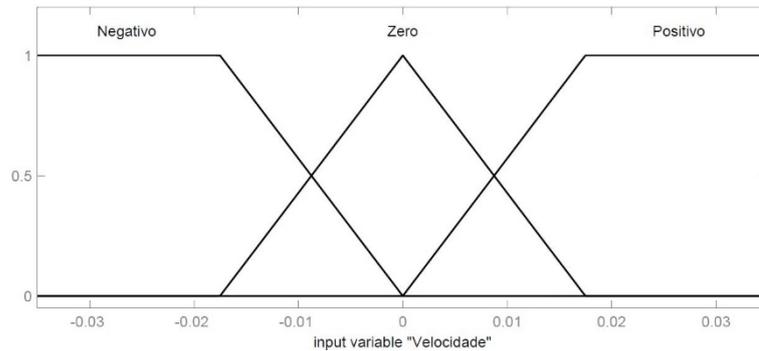


Figura 3.22- Função de pertinência da variação da posição, para o controlador *Fuzzy* de 9 regras

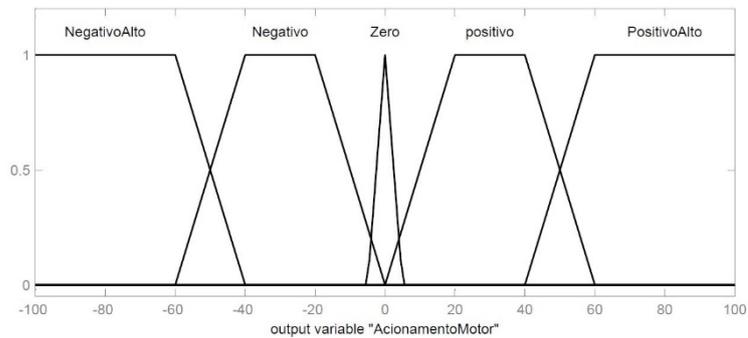


Figura 3.23- Função de pertinência da saída, para o controlador *Fuzzy* de 9 regras

A Tabela 3.5 representa a base de regras escolhida por Silva et al. (2013, p. 6), devido a uma diferença de convenção este autor utiliza uma força negativa quando o ângulo do pêndulo é negativo no intuito de fazer força no mesmo sentido da inclinação do pêndulo. Os parâmetros com os limites de cada variável linguística estão disponíveis na Tabela 3.6.

Tabela 3.5- Base de regras para o controlador *Fuzzy* de 9 regras

Força u		Velocidade \dot{e}		
		N	Z	P
Erro e	N	MN	PN	Z
	Z	PN	Z	PP
	P	Z	PP	MP

Tabela 3.6- Parâmetros das variáveis Linguísticas para o controlador *Fuzzy* de 9 regras

Função de Pertinência	Variáveis Linguísticas	Tipo	Vetor de dados
Erro da Posição	N	Trapezoidal	[-0.1 -0.09 -0.035 0]
	Z	Triangular	[-0.035 0 0.035]
	P	Trapezoidal	[0 0.035 0.09 0.1]
Vari. Posição	N	Trapezoidal	[-0.06 -0.05 -0.0175 0]
	Z	Triangular	[-0.0175 0 0.0175]
	P	Trapezoidal	[0 0.0175 0.05 0.06]
Saída	MN	Trapezoidal	[-150 -100 -60 -40]
	PN	Trapezoidal	[-60 -40 -20 0]
	Z	Triangular	[-5.00 0 5]
	PP	Trapezoidal	[0 20 40 60]
	MP	Trapezoidal	[40 60 100 150]

A superfície de tomada de decisão para este controlador nebuloso pode ser visualizada na Figura 3.24.

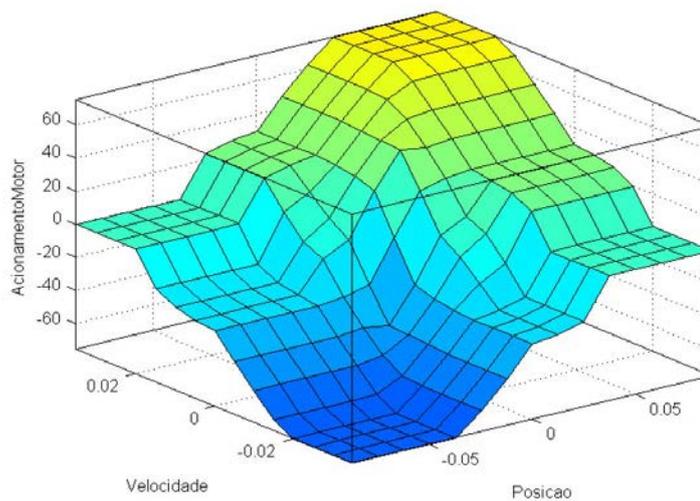


Figura 3.24- Superfície de tomada de decisão do segundo controlador prático

4 ANÁLISE DE RESULTADOS

Este capítulo apresenta os resultados das simulações e experimentos realizados na secção 3.3 por meio de gráficos das posições e velocidade angulares, sinal de controle e erro.

4.1 Resultados do controlador por avanço do fase.

Na Figura 4.1 são apresentados os resultados da simulação e do experimento do controlador por avanço de fase. Para a simulação é possível notar que a posição angular teve um pico de 0,025 rad ($1,43^\circ$) e o tempo de acomodação foi de aproximadamente 0,8 segundos. Já o experimento apresentou um tempo de acomodação maior, e um erro de regime permanente gerado pelo cabo que conecta o *Gyro Sensor* no LEGO. Este erro é de aproximadamente 0,0025 rad ou ($0,14^\circ$).

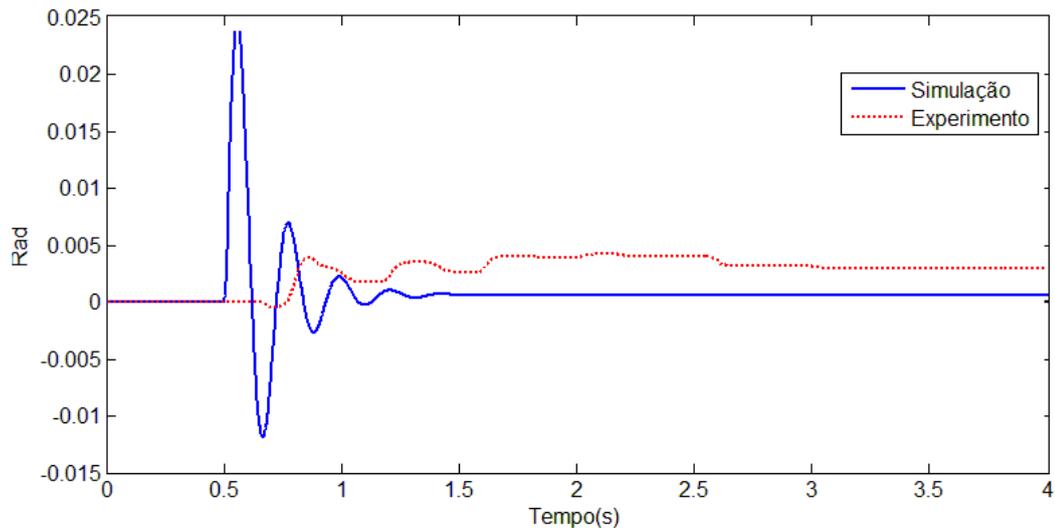


Figura 4.1-Comparação das posições entre simulação e experimento do controlador por avanço de fase, ao receber um impulso em 0,5s

Na Figura 4.2(a) observa-se o sinal de erro e em (b) o sinal de controle gerado pelo controlador por avanço de fase. No experimento é possível notar o erro em regime permanente e que o controlador produz um sinal de comando para eliminá-lo, porém este sinal é insuficiente para acionar os motores.

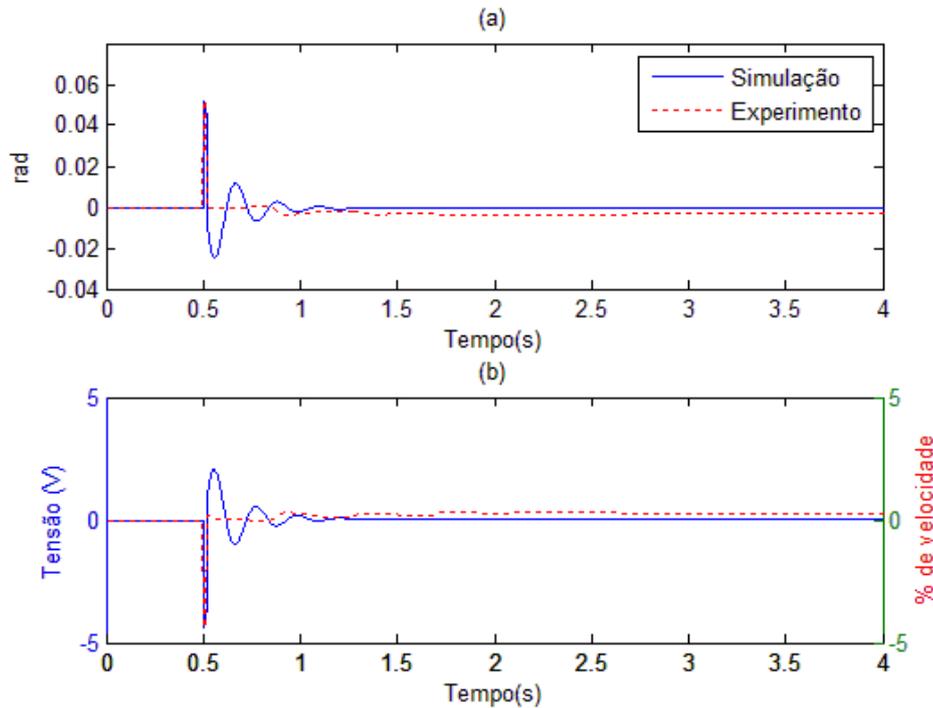


Figura 4.2-Erro da posição e sinal de controle para o experimento ao receber um impulso em 0,5s.

4.2 Análise dos resultados do controlador por avanço de fase.

Analisando as amplitudes das posições angulares da simulação e do experimento, percebe-se que houve uma discrepância entre os resultados. Algumas diferenças entre os modelos simulados e ensaiados explicam esses resultados:

Em primeiro lugar, a obtenção dos valores da posição e velocidade angular, no caso do experimento o sensor oferece uma velocidade angular e que por meio de um integrador se obtém a posição. Já na simulação todos os cálculos são feitos utilizando a posição angular.

Em segundo lugar, o modelo não considera nenhum tipo de atrito entre as rodas e a superfície de contato causando um retardo na inclinação do pêndulo quando o degrau é aplicado.

Em terceiro lugar o atrito do eixo da haste com o carrinho; em quarto lugar, o tempo de processamento do BRICK (este realiza diversas tarefas ao mesmo tempo) pode causar um atraso no acionamento do motor.

Em quinto lugar, o valor medido pelo acelerômetro é filtrado gerando uma pequena atenuação no seu valor, lembrando que este não possui resolução para medidas inferiores a um grau por segundo.

Em sexto lugar, os motores do LEGO pelo MATLAB são acionados por um sinal de amplitude $[-100, 100]$ referente à porcentagem de velocidade em cada sentido de rotação que não está relacionado a nenhuma variável física, como corrente e tensão. Logo em simulações, a saída do controlador é uma tensão de armadura que produz uma força aplicada na FT. Já no experimento a saída do controlador é ligada diretamente ao motor representando um sinal de porcentagem de velocidade.

4.3 Resultados das simulações com controlador *Fuzzy* (25 regras).

Os resultados dos testes teóricos para o controlador nebuloso podem ser encontrados na Figura 4.3. A Figura 4.4 apresenta o sinal do erro e o sinal de controle produzido pelo controlador *Fuzzy* como pode ser observado, este controlador consegue rapidamente reduzir o sinal de erro da posição porém gerando um sinal de controle que alterna entre sinais positivos e negativos em alta frequência, e isto é difícil de se alcançar em um experimento no LEGO.

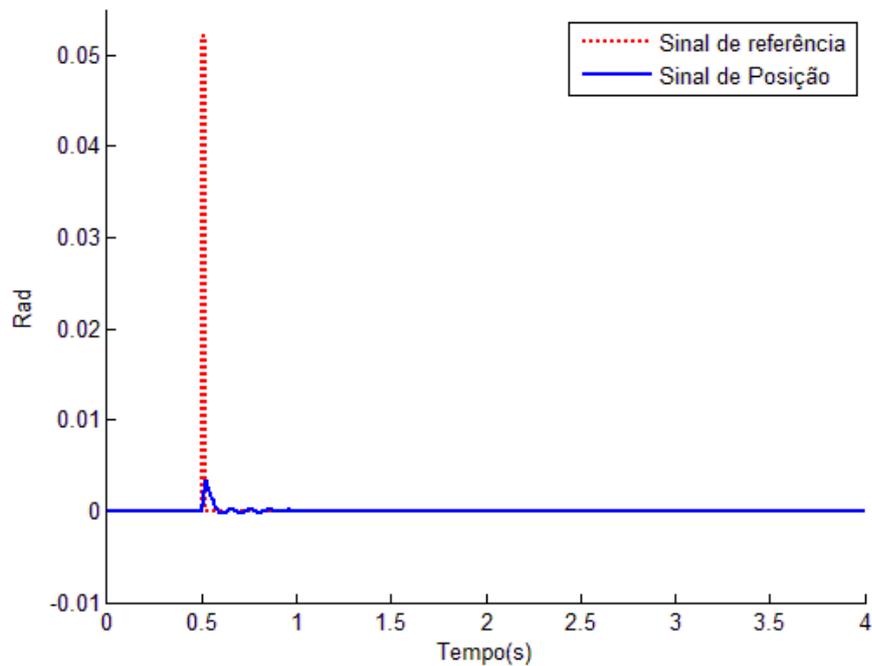


Figura 4.3- Simulação do controlador *Fuzzy*, sinal de controle em verde, posição angular em vermelho e velocidade angular em azul.

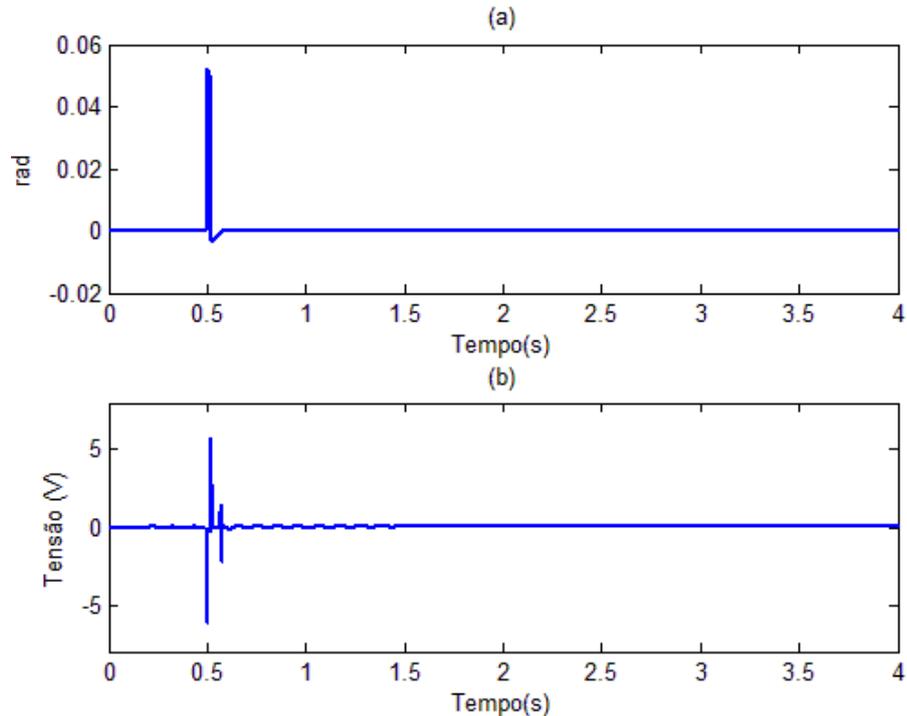


Figura 4.4- Simulação do controlador *Fuzzy*, ao impulso em 0,5s (a) Sinal de erro da posição e (b) sinal de controle

4.4 Experimentos dos controladores nebulosos simplificados.

Nesta secção serão apresentados os resultados dos ensaios dos controladores, as tentativas de realização de um controlador simples o suficiente para que o LEGO consiga processar e a simulação dos controladores.

4.4.1 Controlador nebuloso de 17 regras.

O experimento deste controlador foi programada para 10 segundos, porém o seu tempo de execução real foi de 25 segundos. Isso explica a dificuldade do LEGO em processar e comunicar com o MATLAB simultaneamente, tornando impossível conseguir controlar a posição do pêndulo com este controlador. A Figura 4.5 representa a saída do controlador nebuloso focalizado no intervalo de tempo entre o impulso (1,5s para este experimento) e o momento em que equilibrar a haste se torna impossível (2,3s). Na Figura 4.6 observa-se que em 2,15 segundos a velocidade angular do pêndulo começa a aumentar. Neste momento a posição angular já é suficientemente grande para fazer com que o controlador *Fuzzy* assume uma amplitude menor que -20, porém o atraso de processamento faz com que esse acionamento

ocorra apenas em 2,25 quando a haste do pêndulo já possui uma velocidade muito grande para ser controlada.

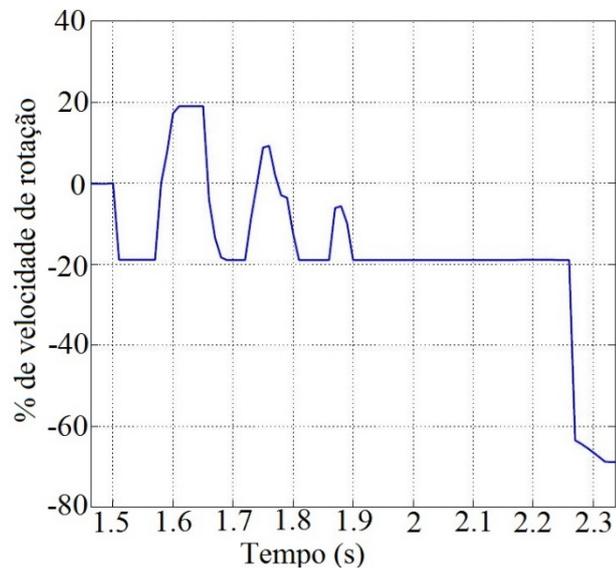


Figura 4.5- Acionamento dos motores pelo controlador de 17 regras

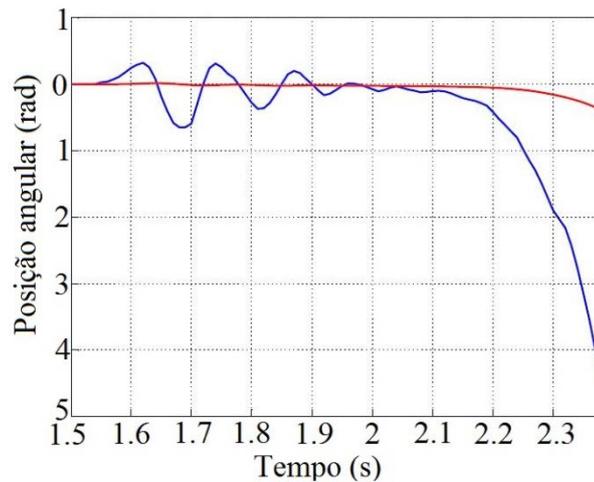


Figura 4.6- Posição angular em vermelho e velocidade angular em azul da haste do pêndulo com o controlador nebuloso de 17 regras.

O atraso de um décimo de segundo é o suficiente para fazer com que a haste do pêndulo se desloque aproximadamente 6° , valor superior aos utilizados no universo de discurso da função de pertinência da posição.

4.4.2 Controlador nebuloso de 9 regras.

O controlador de nove regras como apresentado por Silva et al. (2013, p. 6) apresentou um custo computacional reduzido em relação ao controlador de 17 regras, porém não foi possível

realizar a simulação e nem o experimento com este controlador. Quando uma simulação é iniciada no SIMULINK, mesmo estando o sistema parado sem distúrbio, o controlador *Fuzzy* gera um sinal de comando oscilatório em alta frequência. Esta oscilação ativa uma proteção do SIMULINK que cancela a simulação quando algum sinal do modelo alterna entre positivo e negativo mil vezes em um curto espaço de tempo.

A principal causa desse comportamento oscilatório está no compartilhamento do uso do valor 0 pelos valores linguísticos negativo, zero e positivo das funções de pertinência, como pode ser observado na secção 3.3.5.2. Para tentar corrigir o excesso de oscilação foi proposta uma alteração na função de pertinência da variação do erro de posição conforme a Figura 4.7

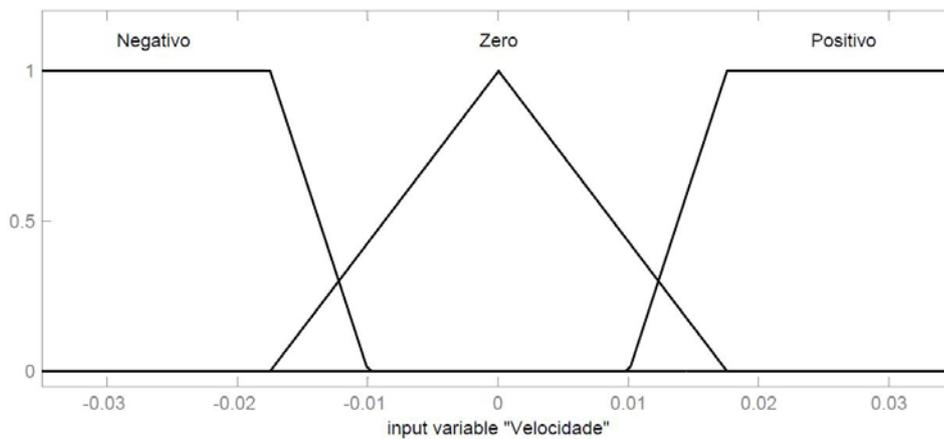


Figura 4.7- Nova função de pertinência para velocidade do controlador de 9 regras

O modelo Silva et al. (2013, p. 6) foi projetado para manter o pêndulo estável sem a presença de um impulso, o universo de discurso da função de pertinência da posição utilizado era de aproximadamente 0,035rad em cada sentido, ou seja inferior ao padrão de impulsos utilizado neste trabalho. Um novo universo de discurso para a posição angular é proposto com amplitude de 0,08rad pra cada lado.

Desta forma extinguimos a oscilação inicial no sistema causada por ruídos aleatórios. Os valores linguísticos Negativo e Positivo só serão acionados quando a variação do erro alcançar 0,01 rad/s ($0,57^\circ/s$) em algum dos sentidos.

Na Figura 4.8 ilustra-se a posição angular da haste do pêndulo durante a simulação e o experimento. A amplitude máxima da posição angular durante a simulação foi inferior a 0,001 rad e no experimento de 0,008 rad ($0,45^\circ$).

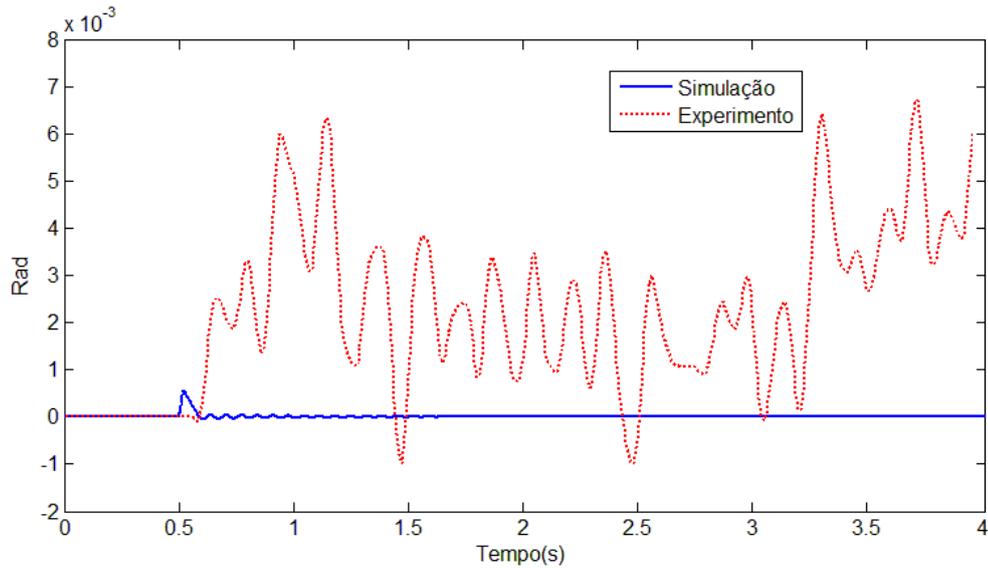


Figura 4.8- Comparação entre os controladores nebulosos, ao impulso em 0,5s.

Na Figura 4.9 observa-se o erro e o sinal de controle para a simulação e para o experimento. A simulação apresenta um erro em regime permanente ínfimo, que gera um oscilação no sinal de controle. Já no experimento, o erro em regime permanente foi maior obrigando o controlador a manter um sinal de controle oscilatório durante todo o período de verificação.

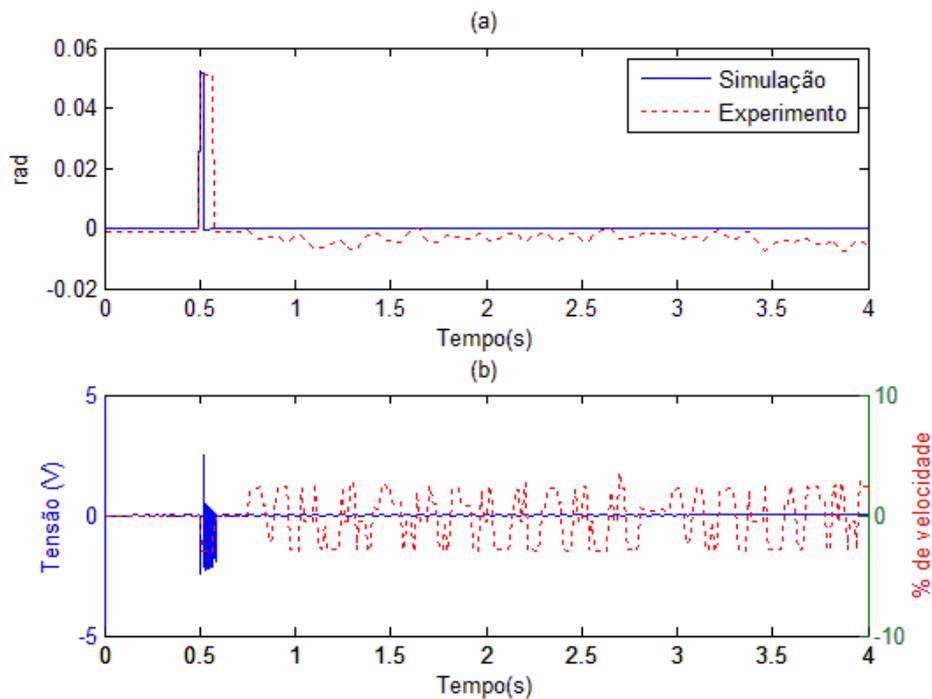


Figura 4.9 – Comparação do controlador *Fuzzy*, ao impulso em 0,5s : (a) Sinal de Erro e (B) Sinal de controle

4.5 Análise do desempenho do controlador *Fuzzy* de 9 regras.

Este controlador apresentou respostas rápidas que geraram uma oscilação que controla a posição da haste na simulação. O modelo utilizado para o controlador *Fuzzy* também apresenta as mesmas simplificações do modelo utilizado no controlador por avanço de fase.

Na simulação o controlador rejeitou o distúrbio quase por completo. Já no experimento, o controlador *Fuzzy* não consegue estabilizar a haste na posição vertical; ainda assim, ele evitou a sua queda ao longo do experimento.

Assim como observado no trabalho de Silva et al. (2013), o controlador *Fuzzy* não elimina o erro em regime permanente, mantendo uma oscilação durante o experimento.

4.6 Comparação entre os controladores *Fuzzy* e por avanço de fase.

Uma comparação entre os resultados obtidos por meio dos experimentos é apresentada na Figura 4.9. Para definir qual é o controlador mais adequado para uma aplicação, é preciso estabelecer os requisitos de operação do sistema. No caso do pêndulo invertido, se o objetivo for reduzir ao máximo o erro em regime permanente, o controlador clássico apresenta uma resposta mais adequada. O controlador *Fuzzy* é mais indicado para suprir a necessidade de uma resposta rápida, quando é aceitável variações do sinal controlado dentro de uma faixa de valores.

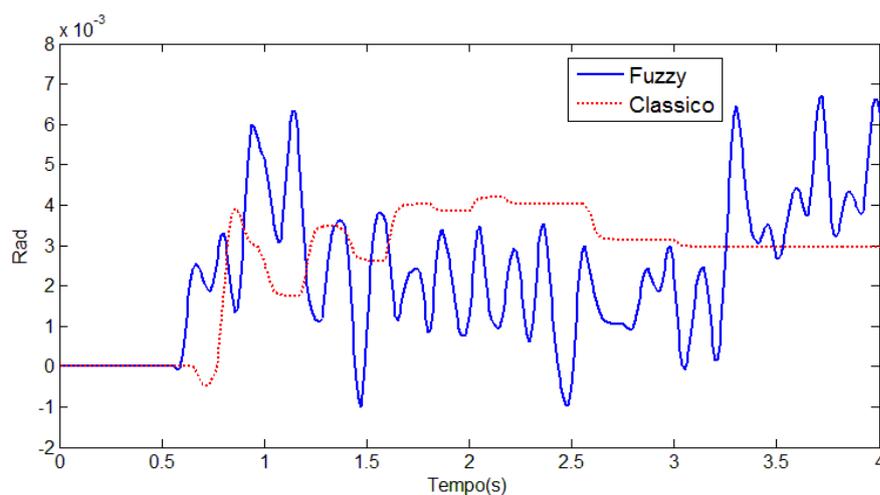


Figura 4.10- Comparação entre os controladores *Fuzzy* e Clássico, ao receber um impulso em 0,5s.

5 CONCLUSÃO

Este trabalho abordou o estudo de dois controladores aplicados num pêndulo invertido tipo carro. O trabalho aplica na prática conceitos de teoria de controle para adaptar e ajustar controladores por avanço de fase e *Fuzzy*, a fim de estabilizar o pêndulo. Foi utilizada a plataforma de baixo custo LEGO *Mindstorms* EV3 para realização dos experimentos com os controladores. O estudo de técnicas de controle por meio de uma atividade prática provou-se ser bastante importante, permitindo o aprofundamento de várias áreas de teoria de controle e o entendimento de conceitos que muitas vezes são abordados dentro de uma sala de aula de forma mais sucinta.

Os resultados alcançados com o compensador por avanço de fase neste trabalho são bons considerando que as simulações e experimentos apresentaram funcionamento semelhante e dentro do esperado no início do projeto, ou seja trazendo o pêndulo de volta ao ponto de equilíbrio após receber um impulso controlado.

O modelo matemático apresentado contém diversas simplificações, e não considera o atrito entre a haste e o carro, o atrito entre as rodas e o solo, as forças de tração e compressão existentes no interior do cabo de conexão do *Gyro Sensor* e as saturações do sistema real, o que explica as diferenças observadas entre simulações e experimentos.

Devido às simplificações utilizadas neste trabalho, o resultado do experimento com o controlador nebuloso foi inferior ao obtido com o controlador por avanço de fase. Com este resultado conclui-se que é fundamental ter o *Hardware* adequado para escolher o controlador. A escolha de um controlador tem que ser feita a partir do modelo de processador disponível.

Como esperado, os controladores apresentaram dinâmicas distintas. O controlador por avanço de fase representa uma economia de custo computacional e um uso reduzido de sinal de controle, enquanto o controlador *Fuzzy* apresentou respostas rápidas, porém oscilatórias, podendo exigir maior custo computacional.

O kit de robótica LEGO *Mindstorms* se mostrou muito eficiente e prático nas fases iniciais do projeto, acelerando a montagem da estrutura e programação de um controlador embarcado. Já em fases mais avançadas os vários *bugs* da plataforma, como dificuldade de leitura confiável de sensores e inúmeras falhas em tentativas de conexão com a rede sem fio, gerou um grande atraso na realização das simulações e experimentos. Como esperado o controlador nebuloso

demanda uma quantidade de processamento muito grande, o que acabou criando a necessidade de realizar um controlador nebuloso muito simples, ou seja, muito inferior a total capacidade deste controlador.

Para o controle de um pêndulo invertido o uso de um controlador por avanço de fase é recomendado quanto se tem um capacidade computacional reduzida. Para outros modelos de pêndulo invertido motores mais potentes devem ser utilizados para cargas maiores e um controlador mais rápido pode ser utilizado para expandir o ângulo máximo de controle. O modelo matemático utilizado neste trabalho sempre apresentará um fator limitante no ângulo máximo da haste do pêndulo, devido às saturações do sistema real.

Em trabalhos futuros, deseja-se utilizar o ROS (Robot Operating System) na plataforma do LEGO para o desenvolvimento de um pêndulo invertido, e investigar a aplicação de outros sensores, por exemplo, um acelerômetro com um módulo sem fio. Por meio desta montagem será possível ter um maior conhecimento dos componentes. Utilizando um sistema operacional dedicado à robótica, é possível realizar um controle mais preciso utilizando a mesma base de conhecimento descrita neste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

AKHTARUZZAMAN, M; SHAFIE, A. **Modeling and Control of a Rotary Inverted Pendulum Using Various Methods, Comparative Assessment and Result Analysis**, International Conference on Mechatronics and Automation, 1342 p, Aug 2010.

CANALE, M; BRUNET, S. **A Lego Mindstorms NXT experiment for Model Predictive Control Education**, European Control Conference, 2549 p, Jul 2013.

CASTRO, A. **Modeling And Dynamics Analysis Of A Two-Wheeled Inverted Pendulum**.2012. 178p. Tese de mestrado (Master of Mechanical Engineering)- Georgia Institute of Technology, Atlanta, USA, 2012

DALSAGER, H. ESBENSEN, T. JENSEN, B. JENSEN, M. NISS, M. SLOTH, C. **Camera Controlled Robot**. 2006. 124 p. Faculty of Engineering, Science and Medicine Department of Electronic Systems, Aalborg University, Aalborg, Dinamarca. 2006

DORF, R; BISHOP, R. **Modern control systems**. 12. ed. Indianapolis: Addison-Wesley, 2011

FILHO, M; JESUS, H; CORTES, J; CARVALHO, A; JÚNIOR, G. **Controle Fuzzy para Posicionamento de um Pêndulo Invertido**. Universidade Estadual do Norte Fluminense, 2013

FRANKLIN, G. F; POWELL, J. D; EMANI-NAEINI, A; **Feedback Control of Dynamics Systems**. 4. ed. New Jersey: Pearson Prentice Hall, 2002

GOODWIN, G; GRAEBE, S; SALGADO, M. **Control System Design**. 1. ed. Valparaíso: Pearson Prentice Hall, 2000

HURBAIN, P. **LEGO 9V Technic Motors compared characteristics**. Disponível em: <<http://www.philohome.com/motors/motorcomp.htm>>. Acesso em: 19 dez. 2014.

KANADA, T; WATANABE, Y; CHEN, G. **Robust H2 Control For Two-Wheeled Inverted Pendulum Using LEGO Mindstorms**, Australian Control Conference, 2011

KIM, Y. **Control System Lab Using A LEGO Mindstorms NXT Motor System**. 18th Mediterranean Conference on Control and Automation. Marrakeche, 2010

KUSHIDA, N; YAMASHITA, S; KIMURA, J; DAI, F. **The embedded educational system using NXT wheeled inverted pendulum robot in OCMT**, 31st Chinese Control Conference, 5640p, Jul 2012.

LEGO, **Lego Mindstorms ev3 User Guide**, 2013. Disponível em: < <http://www.lego.com/en-us/mindstorms/?domainredir=mindstorms.lego.com> >. Acesso em: 19 dez. 2014

LEGOENGINEERING, **EV3 support**, 2014. Disponível em: < <http://www.legoengineering.com/category/support/ev3/> >. Acesso em: 19 dez. 2014

MATHWORKS MatLab 2014a. Versão 8.3.0.532: The Mathworks, INC, 2014. 1DVD

MIRANDA, P.; VILELA, M.; KRONBAUER, D. **Sistema de Controle Difuso de Mandani**. Aplicações: Pêndulo Invertido e outras. Monografia do curso de Análise de Sistemas. 2003. 53p. Monografia (Trabalho de Final de Curso Análise de sistemas) Universidade Federal de Mato Grosso do Sul, Campo Grande, Mato Grosso do Sul, 2003

MORAIS, M. MURALIKRISHNA, A. GUIMARÃES, L. **Sistema de Controle PID e Nebuloso Aplicado ao Problema do Pêndulo Invertido**. Instituto Nacional de Pesquisas Espaciais, Programa de Pós-Graduação em Computação Aplicada. São José dos Campos, 2005

OGATA, K. **Engenharia de Controle Moderno**.4. ed. Sao Paulo:Pearson Pretice Hall, 2003

PASSINO, K; YURKOVICH, S. **Fuzzy Control**. 1. ed. Menlo Park: Addison-Wesley,1998

PAPADIMITRIOU, V; PAPADOPOULOS, E. **Putting Low-Cost Commercial Robotics Components To The Test: Development Of An Educational Mechatronics/Robotics Platform Using Lego Components**, IEEE Robotics And Automation Magazine, 99 p, Set 2007.

SANDRI, S; CORREA, C. **Lógica nebulosa**. Conselho Nacional de Redes Neurais - ITA, São José dos Campos, SP, 1999. p. c073-c090.

SILVA, M. **Estudo De Técnicas De Controle Aplicada Ao Sistema Pêndulo Invertido**. 2009. 51 p. Monografia (Trabalho de Final de Curso em Engenharia de Controle e Automação) – Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto, Minas Gerais, 2009.

SILVA, H. GONGORA, W. SUMAR, R. VALLIM, M. GOEDTEL, A. **Estudo Comparativo Entre Os Controladores Fuzzy E PID Aplicados Ao Controle De Um Pêndulo Invertido Utilizando Plataforma Lego Mindstorms Nxt.** Simpósio Brasileiro de Automação Inteligente. Fortaleza, Brasil, 2013.

SIVANANDAM, S; SUMATHI, S; DEEPA, S. **Introduction to Fuzzy Logic using MATLAB.** 1. ed. Berlim: Springer-Verlag, 2007

YADAV, A; GAUR, P. **Comparative Analysis of Various Control Techniques for Inverted Pendulum,** Nova Deli, India. 2011

ZADEH, L. A. **Fuzzy Sets: Information and Control,** 1. ed. Berkeley. Elsevier, 1965

APÊNDICE

APÊNDICE - Configuração de uma rede sem fio para o LEGO MINDSTORMS

Existem poucos modelos de adaptadores *wifi* que são compatíveis com o LEGO *Mindstorms*. Atualmente a NETGEAR oferece um que de acordo com a LEGO pode ser utilizado, o WNA1100 (WNA1100M não é compatível). Futuramente novas versões de *Firmware* para o LEGO podem possibilitar o uso de outros modelos.

Não é possível conectar o LEGO a um computador em um rede direta, por exemplo a AD-HOC. É necessário utilizar um roteador com o sistema de proteção WPA2 (ou sem nenhuma senha). O padrão IEEE 802.11 disponível depende de qual adaptador *wifi* está conectado ao LEGO, lembrando que dentre as 4 opções de versões disponíveis (a, b, g e n) se um roteador é configurado para o padrão “n” e se o adaptador só suportar o padrão “b”, a comunicação não irá funcionar.

A comunicação utilizada neste trabalho foi feita em IEEE 802.11b, com um adaptador NETGEAR WNA1100 com utilizado o sistema de proteção WPA2. É aconselhável conectar o conector *wifi* antes de ligar o LEGO.