

**GUSTAVO DINIZ SILVA**

**PROJETO, MODELAGEM, SIMULAÇÃO,  
CONSTRUÇÃO E CONTROLE DE UM VEÍCULO DE  
DUAS RODAS PÊNULO INVERTIDO.**



UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE ENGENHARIA MECÂNICA

**2017**

**GUSTAVO DINIZ SILVA**

**PROJETO, MODELAGEM, SIMULAÇÃO, CONSTRUÇÃO E  
CONTROLE DE UM VEÍCULO DE DUAS RODAS PÊNDULO  
INVERTIDO.**

**Monografia de Conclusão de curso** apresentada no programa de graduação em Engenharia Mecatrônica da Universidade Federal de Uberlândia, como parte dos requisitos para obtenção do título de **Bacharel em Engenharia Mecatrônica**.

Área de Concentração: Controle de Sistemas.

Orientador: Prof. Dr. José Jean-Paul Tavares

**UBERLÂNDIA**

**2017**

**Dedico essa Monografia**

À minha mãe e minha irmã que ofereceram todo o apoio possível em todos os momentos durante a graduação.

## AGRADECIMENTOS

A minha mãe Girlene Firmina Diniz, a minha Irmã Yngrid Martins Diniz e meu Tio Leomar José da Silva por todo o apoio oferecido pelo meu ambiente familiar.

À Universidade Federal de Uberlândia e à Faculdade de Engenharia Mecânica pela oportunidade de realizar esse curso.

Ao Professor Dr. José Jean-Paul Tavares pela possibilidade de realizar este projeto, além de toda a orientação e auxílio oferecidos no decorrer do trabalho.

Aos meus amigos do *Manufacturing Automated Planning Laboratory* (MAPL) por todo o apoio e incentivo fornecido durante a confecção deste projeto.

Aos meus amigos e colegas de curso da Engenharia Mecatrônica.

A todos os amigos da Universidade Federal de Uberlândia.

PROJETO, MODELAGEM, SIMULAÇÃO, CONSTRUÇÃO E CONTROLE DE UM VEÍCULO DE DUAS RODAS PÊNDULO INVERTIDO. Monografia de Conclusão de Curso (Bacharel) Universidade Federal de Uberlândia, Uberlândia, 2017.

### Resumo

Esse trabalho visa obter o equilíbrio estático da posição vertical de um veículo de duas rodas do tipo pêndulo invertido, utilizando controladores PIDs obtidos através de técnicas clássicas como a sintonia experimental e o método do lugar das raízes e utilizando também controladores mais sofisticados como o controlador *Fuzzy*. Para isso, foi realizado o projeto de um novo veículo de duas rodas, e sua construção é detalhada ao longo do trabalho. Com esse veículo foi possível obter bons resultados com as técnicas de controle apresentadas.

O veículo foi modelado utilizando o formalismo de Newton-Euler. Como nem sempre é fácil manter o veículo dentro das condições de linearização o modelo não linear obtido foi utilizado para a simulação do veículo com auxílio do método numérico Runge-Kutta de 4<sup>o</sup> ordem, para obter um resultado mais semelhante possível do comportamento real do veículo. Após a identificação dos parâmetros do veículo as respostas da simulação e em tempo real foram comparadas e utilizando testes estatísticos com 95% de confiança foi observado que as respostas não são diferentes para a maioria dos controladores.

Ao projetar um controlador utilizando o modelo linear, dependendo das condições de operação do veículo esse controlador poderá não cumprir os requisitos de projeto, dessa forma também foi proposto uma forma de adaptar um controlador PID obtido pelo método do lugar das raízes ao modelo não linear utilizando a simulação, e foi observado uma melhora substancial do controlador com parâmetros adaptados.

---

*Palavras Chave: Pêndulo invertido, Modelo não linear, Controlador PID, Controlador Fuzzy*

## ABSTRACT

*This work aims to obtain the static equilibrium of the vertical position of an two-wheeled inverted pendulum vehicle using PID controllers obtained through classical techniques as experimental tuning and the root locus method and also using more sophisticated controllers such as the Fuzzy controller . For this, the design of a new two-wheeled vehicle was realized, and its construction is detailed throughout the work. With this vehicle it was possible to obtain good results with the presented control techniques.*

*The nonlinear dynamic model of the two-wheeled inverted pendulum is obtained through the Newton-Euler formulation. Since it is not always possible to keep the vehicle within the linearization conditions, the nonlinear model obtained was used to simulate the vehicle using the 4th order Runge-Kutta method, to obtain a result similar to the actual behavior of the vehicle. After the identification of the vehicle's parameters, the simulation and real-time responses were compared and using statistical tests with 95% confidence, it was observed that the responses are not different for most the controllers.*

*When designing a controller using the linear model, depending on the vehicle's operating conditions, this controller might not accomplish the project requirements, so it was also proposed a way to adapt a PID controller obtained by the root locus method to the nonlinear model using the simulation, and a substantial improvement of the controller with adapted parameters was observed.*

## Lista de Figuras

|   |    |
|---|----|
| Figura 1.1 - Veículo elétrico de transporte EN-V (EN-V, 2017)   | 2  |
| Figura 1.2 – Outros veículos tipo pêndulo invertido. (a) veículo <i>Segway</i> (Segway, 2017). (b) veículo Balanduino (Balanduino, 2017).   | 2  |
| Figura 2.1 - Esquema do circuito elétrico que descreve o motor DC.  | 5  |
| Figura 2.2 - Figura mostrando (a) o esquema de um pêndulo invertido e (b) o diagrama de corpo livre com as forças que atuam no pêndulo. Fonte: Adaptado, OGATA, 2007.                                       | 7  |
| Figura 2.3 - Gráfico da resposta de um sistema de segundo grau a um degrau  | 9  |
| Figura 2.4 - Diagrama de blocos considerando o segurador de ordem zero.   | 11 |
| Figura 2.5 - Função pertinência utilizada para classificar a temperatura ambiente em Muito frio, Frio, Fresco, Quente e Muito Quente.   | 13 |
| Figura 2.6 - Formalismo dos conjuntos difusos e interações lógicas entre esses conjuntos definidos pelas suas funções pertinências. Adaptado (Silva, G. V. M., 2006).                                       | 14 |
| Figura 2.7 - Esquema simplificado do controlador <i>Fuzzy</i> .   | 15 |
| Figura 2.8 - Funções pertinências de saída e também a função pertinência total obtida juntamente com a posição do centroide e a indicação da chance de chover no dia de 42%.                                | 16 |
| Figura 3.1 - Esboço do desenho do veículo de duas rodas mostrando suas principais dimensões e os principais elementos utilizados. O esboço não está em escala.  | 18 |
| Figura 3.2 - Desenho esquemático mostrando as ligações físicas entre os componentes eletrônicos do Hardware do veículo. Os conversores bidimensionais permitiram a interação entre tecnologias 5.0V e 3.3V. | 19 |
| Figura 3.3 - Imagem do motor DC 6.0 V utilizado no veículo. Adaptado de Motor DC, 2017.   | 20 |
| Figura 3.4 - Imagem do driver ponte H L298N utilizado. Adaptado de Ponte H, 2017.   | 21 |
| Figura 3.5 - Imagem da MPU6050. Adaptado de MPU6050, 2017.  | 21 |
| Figura 3.6 - Imagem do Arduino DUE. Adaptado de Arduino DUE, 2017.  | 22 |
| Figura 3.7 - Regulador de tensão X16009. Adaptado de Regulador de tensão, 2017.   | 22 |
| Figura 3.8 - Representação com dois graus de liberdade do veículo de duas rodas. Primeiro grau se refere ao pêndulo invertido na integral e o segundo o deslocamento em x.                                  | 23 |
| Figura 3.9 - Representação dos dois graus de liberdade da base do veículo de duas rodas.  | 23 |

Figura 3.10 - Relação entre o referencial fixo no ponto O e o referencial com orientação fixa no veículo. Observa-se que o referencial do carrinho acompanha a orientação da trajetória, enquanto o inercial não. 24

Figura 3.11 - Diagrama de corpo livre das forças que atuam no veículo de duas rodas. As forças que atuam na base e também as forças que atuam na parte suspensa do veículo de duas rodas pendulo invertido. 25

Figura 3.12 - Figura esquemática mostrando a simplificação do veículo considerando as massas dos componentes eletrônicos e demais componentes como massas  $M_1$  e  $M_2$  concentradas em cada uma das placas do veículo de duas rodas. 29

Figura 3.13. Teste para encontrar os parâmetros do modelo do veículo de duas rodas. Em vermelho a resposta com os parâmetros que minimiza o erro médio quadrático no algoritmo de otimização *Neuro-Fuzzy* e em azul a resposta real do veículo utilizada para a identificação. 31

Figura 3.14 - Gráfico do ângulo com a vertical do veículo de duas rodas pendula invertido para o controle puramente proporcional com  $K=2.8$ . 32

Figura 3.15 - Gráfico da resposta em frequência do ângulo para encontrar o período crítico. 32

Figura 3.16 - Diagrama de blocos do veículo de duas rodas considerando dois graus de liberdade. 33

Figura 3.17 - Gráfico da função transferência do segurador de ordem zero e a linearização da função transferência. Observa-se que se mantém próximo as funções para um tempo de amostragem de 2ms em quase toda a região entre (-400 e 1000). 35

Figura 3.18 - Root locus da função de transferência do veículo de duas rodas com um pólo no zero proveniente da parcela integrativa do controlador. 36

Figura 3.19 - Root locus da função de transferência do veículo de duas rodas com um pólo no zero e dois zeros complexos próximos ao pólo de projeto. 37

Figura 3.20 - Funções pertinência para ângulo com a vertical, valores de ângulo em graus. 39

Figura 3.21 - Funções pertinência para a velocidade angular com a vertical, valores de ângulo em radianos/s. 39

Figura 3.22 - Funções pertinências triangulares para a ação de controle. 41

Figura 3.23 - Fluxograma do algoritmo utilizado em simulação. O fluxograma acima mostra apenas a função principal, a função que realiza a integração numérica, para o entendimento geral do algoritmo é necessária consultar as figuras () onde se encontra o fluxograma das funções *Fpend* e *Fcontrol*. 42

Figura 3.24 - Fluxograma das funções *Fpend* e *Fcontrol* utilizadas na simulação do veículo de duas rodas. Observa-se que esse fluxograma, assim como o da função principal, é uma



|  |    |
|--|----|
| lógica genérica que pode ser utilizada por qualquer lógica de controle e para o modelo linear e não linear.  | 42 |
| Figura 1. Fluxograma utilizado para adaptar os parâmetros do PID ao modelo não linear.   | 47 |
| Figura 4.1. Imagem do veículo de duas rodas real.  | 49 |
| Figura 4.2 - Resultados reais no tempo da posição angular do pêndulo com a vertical para o controlador PID com parâmetros experimentais obtidos pelo método de Z-N de malha fechada.           | 50 |
| Figura 4.3. Ação de controle no tempo para o PID com parâmetros experimentais.   | 50 |
| Figura 4.4. Detalhes do regime transiente do resultado real do veículo e ajuste de curva para a obtenção do tempo de acomodação e outras propriedades importantes da resposta do sistema.      | 51 |
| Figura 4.5 - Resultados simulados para o PID Z-N com as mesmas condições iniciais que foram realizadas os experimentos no veículo real.  | 51 |
| Figura 4.6 - Detalhes do regime transiente do resultado simulado do veículo e ajuste de curva para a obtenção do tempo de acomodação e outras propriedades importantes da resposta do sistema. | 52 |
| Figura 4.7. Gráfico mostrando o resultado da posição angular para o controlador obtido pelo método de alocação dos pólos.  | 53 |
| Figura 4.8. Ação de controle para o controlador PID PROJ no tempo.   | 54 |
| Figura 4.9 - Detalhes do regime transiente para o controlador PID PROJ juntamente com uma curva de ajuste para a melhor obtenção dos resultados.   | 54 |
| Figura 4.10 - Resultados Simulação para o controlador PID PROJ.  | 55 |
| Figura 4.11 - Detalhes do regime transiente para o controlador PID PROJ na simulação juntamente com uma curva de ajuste.   | 55 |
| Figura 4.12 - Resultados reais da resposta do veículo para o controlador <i>Fuzzy</i> .  | 56 |
| Figura 4.13 - Ação de controle no tempo para a resposta real do veículo para o controlador <i>Fuzzy</i> .  | 57 |
| Figura 4.14 - Detalhes do regime transiente da resposta do veículo real para o controlador <i>Fuzzy</i> .  | 57 |
| Figura 4.15 - Resultados simulados para o controlador <i>Fuzzy</i>   | 58 |
| Figura 4.16 - Detalhes do regime transiente da simulação da resposta para o controlador <i>Fuzzy</i> .   | 58 |

Figura 4.17 - Resultados simulados e reais do veículo de duas rodas para o PID com parâmetros adaptados ao modelo não linear. Na esquerda observam-se os detalhes em regime transiente para a resposta e a simulação, e na figura à direita mostra a resposta durante 170 segundos de experimento. 59

Figura 4.18 - Imagem representando uma leitura errada do ângulo  $\theta$  associado a acelerações do centro de massa que não deveriam ser contabilizadas. Os eixos z e x mostrados são os eixos do acelerômetro MPU6050 não os eixos considerados na secção de modelagem dinâmica. 60

## Lista de Tabelas

|  |    |
|--|----|
| Tabela 2.1 – Parâmetros para encontrar o controlador PID experimentalmente pelo método de Z-N.   | 12 |
| Tabela 3.1 - Dados do fabricante sobre a operação do motor.  | 20 |
| Tabela 3.2 - Parâmetros ótimos do veículo de duas rodas para um ajuste de 95%.   | 30 |
| Tabela 1.3 - Relações entre as variáveis linguísticas de entradas criando a variável linguística de saída $u$ .  | 41 |
| Tabela 4.1. Tabela com as principais informações do resultado do controlador para o experimento no veículo em tempo real e para a simulação do veículo em mesmas condições iniciais.               | 52 |
| Tabela 4.2 - Tabela com as principais informações do resultado do controlador para o experimento no veículo em tempo real e para a simulação do veículo em mesmas condições iniciais.              | 55 |
| Tabela 4.3. Tabela com as principais informações do resultado do controlador <i>Fuzzy</i> para o experimento no veículo em tempo real e para a simulação do veículo em mesmas condições iniciais.  | 58 |
| Tabela 4.4 - Tabela com as principais informações do resultado do controlador <i>Fuzzy</i> para o experimento no veículo em tempo real e para a simulação do veículo em mesmas condições iniciais. | 59 |
| Tabela 4.5. Comparação do desempenho dos controladores aplicados no veículo. A marcação em azul indica o melhor controlador em cada critério avaliado.   | 61 |

## Lista de Símbolos

|                 |   |
|-----------------|---|
| $m$             | Massa da parte suspensa   |
| $M$             | Massa da base do veículo  |
| $g$             | Aceleração da gravidade   |
| $l$             | Distância entre o eixo das rodas e o centro da gravidade              |
| $J_w$           | Momento de inércia centroidal da roda                                 |
| $J$             | Momento de inércia do pêndulo ao eixo da roda                         |
| $J_p$           | Momento de inércia centroidal do pêndulo em torno do eixo de balanço  |
| $J_z$           | Momento de inércia da base do pêndulo em relação ao eixo z do veículo |
| $r$             | Raio da roda do veículo   |
| $d$             | Metade da distância entre as rodas do veículo                         |
| $C_a$           | Atrito dinâmico entre o eixo do veículo e o pêndulo                   |
| $b_m$           | Atrito dinâmico interno ao motor                                      |
| $V_{cc}$        | Máxima tensão de alimentação dos motores                              |
| $K_m$           | Constante de proporcionalidade entre o torque e a corrente            |
| $K_e$           | Constante do motor  |
| $R$             | Resistência elétrica interna do motor                                 |
| $dz$            | <i>Dead zone</i> do motor em percentual da tensão de alimentação      |
| $V_{cc}dz$      | <i>Dead zone</i> do motor em Volts                                    |
| $T_m$           | <i>Torque produzido pelo motor</i>                                    |
| $V_m$           | <i>Força contraeletromotriz do motor</i>                              |
| $\theta$        | Ângulo com a vertical do veículo de duas rodas                        |
| $\dot{\theta}$  | Velocidade angular com a vertical                                     |
| $\ddot{\theta}$ | Aceleração angular com a vertical                                     |
| $x$             | Distância percorrida pelo veículo                                     |
| $\dot{x}$       | Velocidade do veículo na direção da trajetória                        |
| $\ddot{x}$      | Aceleração do veículo na direção da trajetória                        |
| $\phi$          | Ângulo de orientação no plano (x, y) do veículo de duas rodas.        |
| $\dot{\phi}$    | Velocidade angular de orientação do veículo de duas rodas.            |
| $\ddot{\phi}$   | Aceleração do angular do veículo em torno do eixo z                   |
| $x_p$           | Posição x do centro de massa do pêndulo invertido de 2G.              |
| $y_p$           | Posição y do centro de massa do pêndulo invertido 2G.                 |

|                        |   |
|------------------------|---|
| $z_p$                  | Posição z do centro de massa do pêndulo invertido 3G.                               |
| $x_{Veiculo/O}$        | Posição x do centro de massa da base do veículo em relação ao referencial inercial. |
| $y_{Veiculo/O}$        | Posição y do centro de massa da base do veículo em relação ao referencial inercial. |
| $\dot{\varphi}$        | Velocidade angular do eixo do motor.  |
| $\dot{\varphi}_{eixo}$ | Velocidade angular do eixo das rodas.   |
| $\varphi_R$            | Ângulo de rotação da roda direita.  |
| $\varphi_L$            | Ângulo de rotação da roda esquerda.   |
| $i_R$                  | Corrente elétrica no motor direito.   |
| $i_L$                  | Corrente elétrica no motor esquerdo.  |
| $u_R$                  | Ação de controle para o motor direito.  |
| $u_L$                  | Ação de controle para o motor esquerdo.   |
| $F_R$                  | Força aplicada na roda por efeito do motor direito.                                 |
| $F_L$                  | Força aplicada na roda per efeito do motor esquerdo.                                |
| $f$                    | Força total atuando na base do veículo.   |
| $H$                    | Força de contato horizontal entre a base e a parte suspensa do veículo              |
| $V$                    | Força de contato vertical entre a base e a parte suspensa do veículo                |
| $N$                    | Força normal de contato entre a base do veículo e o chão.                           |
| $f^*$                  | Força total em condições ideais na base do veículo.                                 |
| $T_R$                  | Torque aplicado na roda direita do veículo.   |
| $T_L$                  | Torque aplicado na roda esquerda do veículo.  |
| $T_{ac}$               | Tempo de acomodação   |
| $T_a$                  | Tempo de amostragem   |
| $\vec{r}_K$            | Vetor de estado do sistema  |
| $f(t)$                 | Função com dependência temporal genérica  |
| $F(S)$                 | Transformada de Laplace da função temporal  |
| $G_{zOH}(S)$           | Função transferência do segurador de ordem zero                                     |
| $G(S)$                 | Função transferência do sistema a ser controlador                                   |
| $G_c(S)$               | Função transferência do controlador   |
| $\omega_n$             | Frequência natural  |
| $\xi$                  | Amortecimento   |

|              |   |
|--------------|---|
| $OS$         | <i>Overshoot</i>                            |
| $PID(S)$     | Função transferência do PID                 |
| $E(S)$       | Função erro no domínio S                    |
| $U_{PID}(S)$ | Função ação de controle do PID no domínio S |
| $K_d$        | Constante proporcional                      |
| $K_i$        | Constante integrativa                       |
| $K_p$        | Constante derivativa                        |
| $e(t)$       | Sinal de erro no tempo                      |
| $u_{PID}$    | Sinal de controle do PID no tempo           |

## LISTA DE ABREVIATURAS

|                  |                                  |
|------------------|----------------------------------|
| CI               | Circuito Integrado               |
| DC               | <i>Direct Current</i>            |
| FFT              | <i>Fast Fourier Transform</i>    |
| I <sup>2</sup> C | <i>Inter-Integrated Circuit</i>  |
| MPU              | <i>Motion Processing Unit</i>    |
| P                | Proporcional                     |
| PD               | Proporcional Derivativo          |
| PI               | Proporcional Integral            |
| PID              | Proporcional Integral Derivativo |
| PWM              | <i>Pulse-Width Modulation</i>    |
| USB              | <i>Universal Serial Bus</i>      |
| ZNMF             | Ziegler-Nichols em Malha Fechada |
| exp              | Função exponencial               |

## Sumário

|  |           |
|--|-----------|
| <b>CAPÍTULO I: INTRODUÇÃO .....</b>  | <b>1</b>  |
| 1.1 Objetivos .....  | 3         |
| 1.2 Justificativas.....  | 3         |
| <b>CAPÍTULO II: FUNDAMENTAÇÃO TEÓRICA .....</b>                            | <b>4</b>  |
| 2.1 Modelagem de Motores DC .....  | 4         |
| 2.2 Revisão e Modelagem Dinâmica Clássica do Pêndulo Invertido. ....       | 6         |
| 2.3 Revisão da Teoria de Controle para Sistemas Lineares .....             | 9         |
| 2.4 Lógica Fuzzy .....   | 13        |
| <b>CAPÍTULO III : METODOLÓGIA .....</b>                                    | <b>17</b> |
| 3.1 Construção do Veículo de Duas Rodas .....                              | 18        |
| 3.1.1 Estrutura mecânica .....   | 17        |
| 3.1.2 Hardware do veículo de duas rodas.....                               | 19        |
| 3.2 Modelo Dinâmico e Identificação dos Parâmetros do Veículo.....         | 23        |
| 3.2.1 Desenvolvimento do modelo dinâmico do veículo .....                  | 23        |
| 3.2.2 Cálculos e identificação dos parâmetros do veículo .....             | 29        |
| 3.3 Sintonia de PID Experimental - PID Z-N.....                            | 31        |
| 3.4 Projeto de um PID pela Teoria Clássica de Controle – PID PROJ .....    | 33        |
| 3.5 Projeto de um Controlador Fuzzy .....                                  | 38        |
| 3.6 Simulação do Sistema Linear e do Sistema Não Linear.....               | 42        |
| 3.7 Aplicação dos Controladores no Veículo Real.....                       | 46        |
| 3.8 Adaptação do Controlador PID ao Modelo Não Linear - PID ADP .....      | 47        |
| <b>CAPÍTULO IV : RESULTADOS E DISCUSSÕES.....</b>                          | <b>48</b> |
| 4.1 Resultados da construção do veículo .....                              | 49        |
| 4.2 Resultados para o controlador PID Z-N. ....                            | 50        |
| 4.3 Resultados para o controlador PID PROJ.....                            | 53        |
| 4.4 Resultados para o controlador Fuzzy. ....                              | 56        |
| 4.5 Resultados Controlador PID ADP .....                                   | 59        |
| 4.6 Comparação e análises detalhadas das respostas dos controladores ..... | 60        |
| <b>CAPÍTULO V: CONCLUSÃO .....</b>   | <b>62</b> |
| Referência Bibliográfica.....  | 64        |
| APÊNDICE .....   | 66        |



# CAPÍTULO I

## INTRODUÇÃO

A necessidade de um sistema de transporte que atenda necessidades individuais a custo reduzido tem estimulado o desenvolvimento de veículos de duas rodas do tipo pêndulo invertido como é mostrado o veículo EN-V (*Electric Networked-Vehicle*) na Figura 1.1. Além do EN-V, foguetes, o *Segway* (veículo elétrico de transporte), pernas de robôs humanoides e veículos de duas rodas (Figura 1.2) podem ser considerados uma variação de um pêndulo invertido e podem utilizar das técnicas de controle que foram desenvolvidas para ele.

Apesar do estudo do pêndulo invertido não se resumir apenas em pesquisas na área de controle, o pêndulo invertido é amplamente utilizado em estudos das metodologias de controle (Bonafilia, 2014; Chen *et. al.*, 2008; Saifizul, 2006) devido às características instáveis e as não linearidades do sistema. Esse sistema não linear é comumente usado em pesquisas que estudam sobre a eficiência, propostas de melhorias e robustez dos controladores.

Há métodos de controle aplicados ao pêndulo invertido que exploram o controle de sistemas não lineares a partir da linearização do modelo, no entanto, ao realizar o procedimento, partes das características do modelo são negligenciadas e podem reduzir a eficiência do controlador projetado. Deste modo, uma simulação do sistema não linear para auxílio no projeto do controlador pode ajudar principalmente quando o sistema está no limite da região de linearização.

O controle *fuzzy* (Gomide e Gudwin, 1994) manuseia informações qualitativas de acordo com a lógica *fuzzy*, levando em consideração a falta de exatidão e a incerteza inerente ao sistema a ser controlado. A sua utilização em sistemas de controle em tempo real, em computadores ou microcontroladores, é das mais convenientes, dado que, geralmente, não envolvem nenhum problema computacional complexo. A teoria de modelagem e controle *fuzzy* trata do relacionamento entre entradas e saídas, agregando vários parâmetros de controle. Isso permite a consideração de processos complexos, de modo que os sistemas de controle resultantes proporcionam um resultado mais acurado, além de um desempenho estável e robusto, como é o caso da aplicação do controle *fuzzy* no controle de veículos do tipo pêndulo invertido.



Figura 1.1 - Veículo elétrico de transporte EN-V (EN-V, 2017)



Figura 1.2 – Outros veículos tipo pêndulo invertido. (a) veículo Segway (Segway, 2017). (b) veículo Balduino (Balduino, 2017).

## 1.1. Objetivos

O objetivo geral desse projeto visa utilizar técnicas de controle para a estabilidade vertical estática de um veículo de duas rodas do tipo pêndulo invertido.

Os objetivos específicos são:

- Planejamento e construção da estrutura de um veículo de duas rodas do tipo pêndulo invertido.
- Escolha do *Hardware* para o veículo.
- Estudo e implementação de técnicas de sintonia experimental de PID.
- Estudo e implementação de técnicas de controle clássicas.
- Desenvolvimento de um modelo não linear com 3 graus de liberdade para o veículo.
- Desenvolvimento de um simulador para testes de controladores utilizando o modelo linear e não linear.
- Projeto de um controlador *Fuzzy*.
- Melhoria dos controladores projetados utilizando a simulação do modelo não linear.
- Comparação dos métodos de controle implementados.

## 1.2. Justificativa

O problema do pêndulo invertido é um sistema não linear que exige que as técnicas de controle aplicadas “compensem” essas não linearidades. Este problema é extremamente comum e muito utilizado em estudo na área de controle, sendo assim um *benchmark* para a área. Desse modo, o projeto se propõe no controle de um pêndulo invertido utilizando técnicas de controle clássicas e adaptando-as para sistemas não lineares, buscando um controle eficiente e robusto.

Para a solução do controle do pêndulo invertido, supõe-se na maioria dos trabalhos que esteja dentro da região de linearização, onde nessa região o pêndulo tem um comportamento quase linear. Porém, durante os testes realizados com o pêndulo observa-se que nem sempre é fácil manter as condições de linearização durante o controle, seja por limitação de *hardware*, ou outro limitante. Sendo assim esse trabalho também se justifica por fazer uso de técnicas de simulação para pré-adaptação dos controladores.

## CAPÍTULO II

### FUNDAMENTAÇÃO TEÓRICA

Este capítulo trata de alguns assuntos importantes para a compreensão do trabalho, tais como: modelagem de motor DC, revisão dinâmica e modelagem clássica do pêndulo invertido, revisão da teoria clássica de controle para sistemas lineares e lógica *fuzzy*.

#### 2.1. Modelagem de Motores DC

Um motor DC pode ter sua representação simplificada no circuito descrito da Figura 2.1. Utilizando a lei de Kirchhoff a equação da malha do motor DC é descrita pela Equação (2.1).

Considerando a corrente elétrica constante, desconsiderando reações na armadura, efeitos magnéticos secundários e também deformidades na geometria do motor, o fluxo magnético que atravessa o entreferro do motor DC pode ser considerado constante e proporcional da velocidade angular do eixo do motor. Como mostra a Equação (2.2).

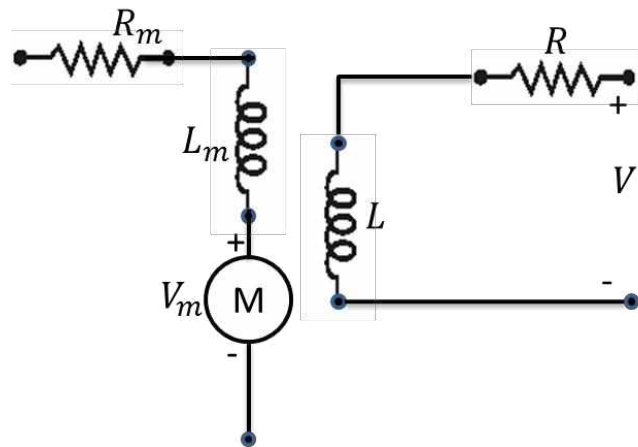


Figura 2.1 - Esquema do circuito elétrico que descreve o motor DC.

$$V = Ri + L \frac{di}{dt} + V_m \quad (2.1)$$

$$V_m = K_e \dot{\phi} \quad (2.2)$$

Uma força é exercida nos condutores imersos em um campo magnético. O campo magnético do motor DC é gerado pela corrente elétrica nas bobinas que são representadas com a resistência  $R$  e a indutância  $L$ .

A força líquida produzida no eixo do motor é então proporcional à corrente, ao número de espiras das bobinas entre outros fatores. O torque líquido que atua sobre o eixo do motor é a força líquida multiplicada pelo raio. Essas influências constantes no torque que dependem da geometria e do material se acumulam em uma constante, e o torque que é aplicado no eixo fica proporcional a corrente no motor DC, como mostra a Equação (2.3).

$$T_m = K_m i \quad (2.3)$$

Considerando o motor acoplado a um disco de inércia com momento de inércia  $J_w$  (no caso é à roda do veículo de duas rodas) e com um atrito dinâmico  $b$  no eixo do motor a equação dinâmica do motor acoplado ao disco encontra-se na Equação (2.4).

$$T_m = J_w \ddot{\phi} + b \dot{\phi} \quad (2.4)$$

As equações acima foram encontradas utilizando várias simplificações, para o motor real existem várias não linearidades devido a folgas mecânicas, a fatores que foram desconsiderados, ou são considerados constantes.

Também existe outra propriedade não linear importante do motor que é a sua zona morta (*dz- dead zone*). Uma tensão abaixo da necessária para vencer a zona morta do motor não causará nenhum movimento do eixo do motor até que essa supere a tensão mínima (explicando o termo zona morta). Necessita-se assim de uma pequena alteração da equação do torque para considerar a *dead zone*, o que será explorado posteriormente.

## 2.2. Revisão e Modelagem Dinâmica Clássica do Pêndulo Invertido.

Realizando uma revisão da dinâmica para sistemas de partículas, ou para sistema composto por múltiplos corpos distinto. Nesse caso as propriedades dinâmicas do conjunto podem ser simplificadas utilizando as grandezas dinâmicas atuando no centro de massa do sistema.

Dado um referencial a coordenada do centro de massa pode ser descrita pela Equação (2.5) onde  $n$  é o número de corpos distintos do sistema. A força resultante que atua no centro de massa do sistema é o somatório das forças atuando em cada corpo, idem para o torque resultante, e essas grandezas dinâmicas se relacionam com o movimento do sistema através respectivamente das Equações (2.6) e (2.7). Observando-se que a Equação (2.7) é uma simplificação para problemas dinâmicos onde os produtos de inércia são nulos para os eixos adotados e  $j$  indica um eixo generalizado que pode se referir aos eixos  $x$ ,  $y$  ou  $z$ .

$$\vec{r}_{CM} = \frac{\sum_{i=1}^n m_i \vec{r}_i}{\sum_{i=1}^n m_i} \quad (2.5)$$

$$\sum_{i=1}^n \vec{F}_i = \left( \sum_{i=1}^n m_i \right) \frac{d^2 \vec{r}_{CM}}{dt^2} = m_T \frac{d^2 \vec{r}_{CM}}{dt^2} \quad (2.6)$$

$$\sum_{i=1}^n T_{ji} = \left( \sum_{i=1}^n (J_{oji} + m_i |r_{ji}|^2) \right) \frac{d^2 \alpha_{jCM}}{dt^2} = J_{CM} \frac{d^2 \alpha_{jCM}}{dt^2} \quad (2.7)$$

Agora, foca-se no problema do pêndulo invertido, exemplificado na Figura 2.2. A posição do centro de massa da parte suspensa (barra acima da parte móvel) pode ser descrito no plano  $(x, y)$  através das Equações (2.8) e (2.9).

$$x_p = x + l \sin \theta \quad (2.8)$$

$$y_p = l \cos \theta \quad (2.9)$$

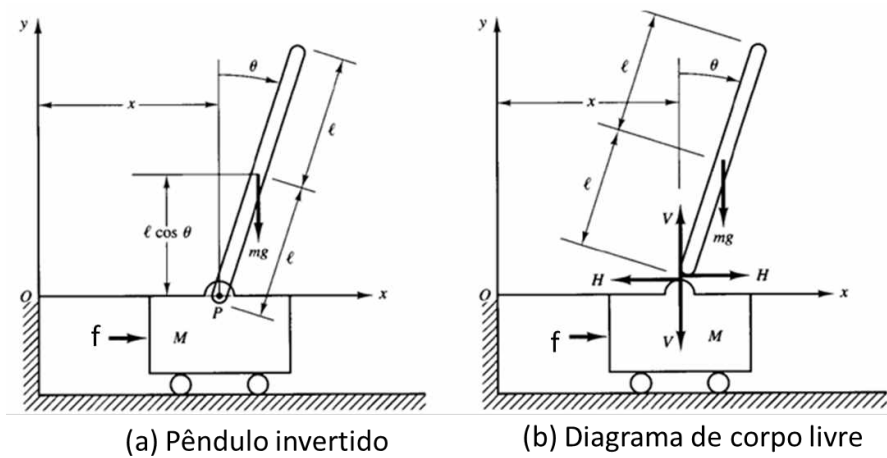


Figura 2.2 - Figura mostrando (a) o esquema de um pêndulo invertido e (b) o diagrama de corpo livre com as forças que atuam no pêndulo. Fonte: Adaptado, OGATA, 2007.

As equações dinâmicas que descrevem o comportamento do pêndulo podem ser representadas pelo conjunto de equações conforme mostrado na Equação (2.10).

$$\begin{cases} (f - H) = M\ddot{x} \\ H = m\ddot{x}_p = m(\ddot{x} - l \sin \theta \dot{\theta}^2 + l \cos \theta \ddot{\theta}) \\ V - mg = m\ddot{y}_p = m(-l \cos \theta \dot{\theta}^2 - l \sin \theta \ddot{\theta}) \\ J_p \ddot{\theta} = Vl \sin \theta - Hl \cos \theta \end{cases} \quad (2.10)$$

Linearizando as equações cima, considerando a condição de ângulos pequenos (menores que  $5^\circ$ ) e velocidade angular pequena e utilizando aproximação por série de Taylor fazendo  $\sin(\theta) \approx \theta$ ,  $\cos(\theta) \approx 1$ ,  $\dot{\theta}^2 = 0$ . O sistema se simplifica no conjunto de equações conforme mostrado na Equação (2.11).

$$\begin{cases} (f - H) = M\ddot{x} \\ H = m\ddot{x}_p = m(\ddot{x} + l\ddot{\theta}) \\ V - mg = m\ddot{y}_p = 0 \\ J_p \ddot{\theta} = Vl\theta - Hl \end{cases} \quad (2.11)$$

Isolando as equações do Sistema acima e reescrevendo no sistema da Equação (2.12), o comportamento dinâmico do pêndulo invertido se resume nas duas equações desse sistema dentro da região de linearização.

$$\begin{cases} (M + m)\ddot{x} + ml\ddot{\theta} = f \\ (J_p + ml^2)\ddot{\theta} + m\ddot{x} = mgl\theta \end{cases} \quad (2.12)$$

Essa simplificação vale para o pêndulo invertido de 2 graus de liberdade, dentro da região de linearização e desconsiderando atritos no eixo que conecta a base e o pêndulo. Também não foi considerada ainda a origem da força que atua na base do veículo de duas rodas.

Isolando as variáveis de derivadas superiores no sistema da Equação (2.12) encontra-se a Equação diferencial de cada grau de liberdade do pêndulo invertido na Equação (2.13).

$$\begin{cases} \left( M + m - \frac{m^2 l^2}{J_p + ml^2} \right) \ddot{x} = f + \frac{m^2 l^2 g \theta}{J_p + ml^2} \\ \left( J_p + ml^2 - \frac{m^2 l^2}{m + M} \right) \ddot{\theta} - mgl\theta = -\frac{mlf}{m + M} \end{cases} \quad (2.13)$$

Observa-se que o sistema de equações diferenciais acima é um sistema acoplado, e raramente é trivial obter a solução de ambas as equações simultaneamente de forma analítica. Porém a segunda equação desse sistema depende apenas da entrada do sistema e conhecendo a mesma, é possível solucionar a equação diferencial homogênea de 2º ordem e posteriormente a particular. Já a solução da equação diferencial para a posição da base vai depender além da função de força externa também a solução da equação diferencial citada anteriormente.



### 2.3. Revisão da Teoria de Controle para Sistemas Lineares

Uma importante ferramenta para a análise de sistemas lineares é transformada de Laplace que muda o domínio da função no tempo para funções no domínio  $s$ . Através de suas propriedades, as equações diferenciais no tempo são representadas apenas como polinômios no domínio  $S$  simplificando consideravelmente as análises. A transformada e sua principal propriedade considerando as condições iniciais nulas (veículo inicialmente em repouso), na qual é utilizada nesse trabalho encontra-se respectivamente nas Equações (2.14) e (2.15).

$$F(s) = \int_0^{\infty} f(t) \exp(-st) dt \quad (2.14)$$

$$\int_0^{\infty} \frac{d^n f(t)}{dt^n} \exp(-St) dt = S^n F(s) \quad (2.15)$$

Alguns sistemas dinâmicos, quando em malha fechada e inserido os controladores adequados, têm um comportamento dinâmico de um sistema estável de segunda ordem. A resposta de um sistema de segunda ordem no tempo é mostrada pelo gráfico da Figura (2.3).

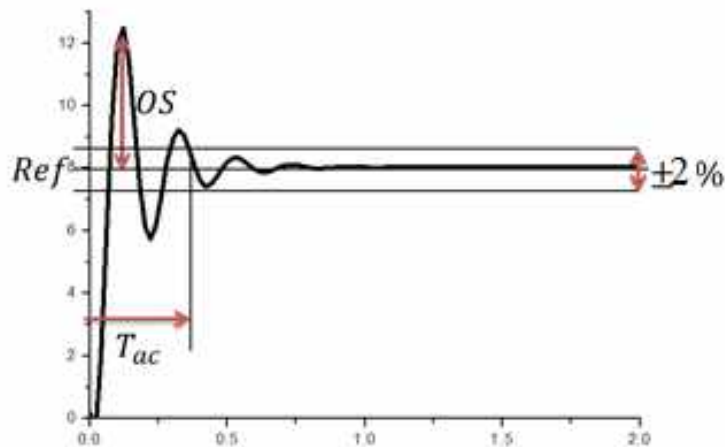


Figura 2.3 - Gráfico da resposta de um sistema de segundo grau a um degrau.

Em sistemas de segunda ordem existem duas principais características muito importantes na área de controle que é o amortecimento e a frequência natural do sistema. O conhecimento dessas variáveis permite a estimativa do *overshoot* ( $OS$ ) que o sistema pode

sofrer e também o tempo de acomodação ( $T_a$ ). Essas propriedades são muito importantes e permitem uma análise qualitativa e quantitativa da resposta do sistema no tempo analisando o sistema pelo domínio  $S$ .

$$OS = \exp(-\xi / \sqrt{1-\xi^2}) \quad (2.16)$$

$$T_a = \frac{4}{\xi \omega_n} \quad (2.17)$$

A partir das Equações (2.16) e (2.17) acima, podem-se definir as posições aproximadas dos pólos (onde a função transferência tem uma raiz no denominador) do sistema analisado pelo domínio  $s$  a partir da Equação (2.18).

$$P = -\xi \omega_n \pm \omega_n \sqrt{1-\xi^2} i \quad (2.18)$$

Para o sistema em malha fechada com compensador, a equação dos pólos do sistema em conjunto é mostrada pela Equação (2.19).

$$1 + KG_C(S)G(S) = 0 \rightarrow \begin{cases} \angle(KG_C(S)G(S)) = -180^\circ \\ K = \frac{1}{|G_C(S)| \cdot |G(S)|} \end{cases} \quad (2.19)$$

O lugar das raízes é o conjunto de pólos, onde a equação acima é válida para a função transferência do sistema  $G(S)$  e a função transferência do controlador  $G_C(S)$ , ou seja, se existe um conjunto de valores de  $K$  que satisfazem a equação, o conjunto de pólos para cada  $K$  encontrado é chamado de *root locus* (lugar das raízes).

No *root locus* encontra-se todas as raízes para todos os valores de  $K$  que satisfaz a Equação (2.19) e é muito utilizada para projeto de controladores.

Porém, nos dias atuais, o controlador projetado para uma malha de controle utiliza-se de um microcontrolador com saídas analógicas (PWM) e entradas analógicas. Porém, essas entradas e saídas não podem ser consideradas totalmente analógicas, necessitando assim de algumas adaptações utilizando a teoria de Controle Digital considerando o tempo de amostragem  $T$ .

Um importante conceito que se utilizado em controle digital é a presença do segurador de ordem zero na planta a ser controlada devido ao microcontrolador, como mostra a Figura 2.4 uma malha de controle simplificada. A diferença entre o sinal de referência e o sinal lido no sensor é interpretada pela lógica do controlador que decide a ação de controle. Essa ação de controle através de uma saída analógica (contendo o segurador de ordem zero) é enviada como entrada na planta e gera uma saída que é novamente lida pelo sensor e assim sucessivamente.

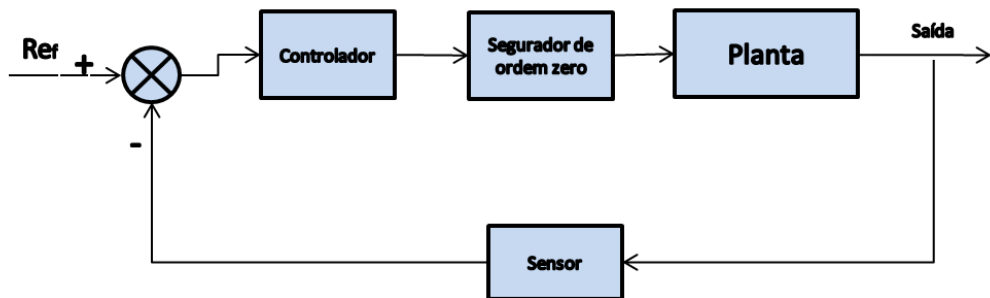


Figura 2.4 - Diagrama de blocos considerando o segurador de ordem zero.

A função transferência do segurador de ordem zero encontra-se na Equação (2.20) onde  $T$  é o tempo de amostragem. Quando se analisa sistemas de controle considerando um sistema contínuo no tempo para então aplicar um microcontrolador que trabalha apenas com variáveis discretas, a função transferência da planta total deve ser alterada e levar em consideração o segurador de ordem zero.

$$G_{ZOH}(S) = \frac{1 - e^{-sT}}{s} \quad (2.20)$$

Como algumas análises utilizando de uma função exponencial são complexas, como é o caso do *root locus* e posteriormente para encontrar o controlador, necessita-se de uma linearização para as análises. A exponencial foi linearizada utilizando a série de Taylor em torno de zero e é mostrada abaixo a função transferência do segurador de ordem zero.

$$\frac{1 - e^{-sT}}{s} = \frac{1 - \left( \sum_{n=0}^{\infty} \left[ \frac{d^n}{dt^n} (e^{-sT}) \right]_{s=0} \frac{s^n}{n!} \right)}{s} = T - \frac{T^2 s}{2!} + \frac{T^3 s^2}{3!} - \frac{T^4 s^3}{4!} + \dots \quad (2.21)$$

Sendo assim, a expansão de primeira ordem para um tempo de amostragem  $T$  considerando o segurador de ordem zero é apenas os dois primeiros termos da série mostrada na Equação (2.21) truncando o restante dos termos da série.

Um controlador PID (proporcional, iterativo e derivativo) tem sua ação descrita como mostra a Equação (2.22), onde o controle em parte é proporcional ao erro, parte proporcional a integral do erro e parte proporcional à taxa de variação temporal do erro.

Já sua função transferência no domínio  $S$ , encontra-se na Equação (2.23). Essa função transferência é usada para encontrar o controlador dado os requisitos de projeto e as equações do *root locus* descritas anteriormente. O controlador PID para um sistema linear é então determinado por quais são os seus parâmetros  $K_p$ ,  $K_i$  e  $K_d$ .

$$u_{PID} = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (2.22)$$

$$\frac{U_{PID}(S)}{E(S)} = PID(S) = K_p + \frac{K_i}{S} + K_d S \quad (2.23)$$

Existem métodos de sintonia para PID que não levam em consideração o modelo, facilitando a obtenção do controlador. Um desses métodos é Método de Sintonização de Ziegler-Nichols (Z-N) de Malha Fechada.

Esse método consiste nas seguintes etapas:

- Inserir um controlador proporcional e anotar o valor onde se inicia uma oscilação periódica no veículo de duas rodas, esse valor de proporcional é o  $K_{CRT}$
- O período de oscilação mínimo é o  $P_{CRT}$

E após encontrar o valor de período crítico e proporcional crítico, os parâmetros do controlador podem ser obtidos através da Tabela 2.1.

Tabela 2.1 – Parâmetros para encontrar o controlador PID experimentalmente pelo método de Z-N.

| Controlador | $K_p$         | $K_i$         | $K_d$          |
|-------------|---------------|---------------|----------------|
| P           | $0.50K_{CRT}$ | 0             | 0              |
| PI          | $0.45K_{CRT}$ | $1.2/P_{CRT}$ | 0              |
| PID         | $0.60K_{CRT}$ | $2/P_{CRT}$   | $0.125P_{CRT}$ |

## 2.4. Lógica Fuzzy

A ideia de Lógica *fuzzy* (difusa) foi criada para tratar melhor certos problemas que são difíceis de classificar de forma booleana. Por exemplo, se te perguntarem “Como está a temperatura ambiente para você nesse dia?”, se você dispuser de um termômetro no momento você poderia responder com a temperatura medida, porém o mais comum que você responda qualitativamente entre quente, fresco ou frio. Supondo a resposta seja quente, mas pode acontecer que alguém ao seu lado discorde de você.

Sendo assim fica difícil classificar o dia em quente, frio ou fresco de forma booleana. Nesse caso, define-se a ideia de conjunto difuso, onde para certa temperatura ambiente a temperatura pertence em certo grau ao conjunto de quente, em certo grau ao conjunto de fresca e em certo grau ao conjunto de frio.

O quanto uma variável pertence a um conjunto é avaliado pela função pertinência que é responsável por avaliar entre 0 e 1 o quanto essa variável pertence ao conjunto, sendo 1 participação total e 0 a não participação. Como por exemplo, a Figura 2.5 mostra um exemplo funções pertinências para a temperatura ambiente.

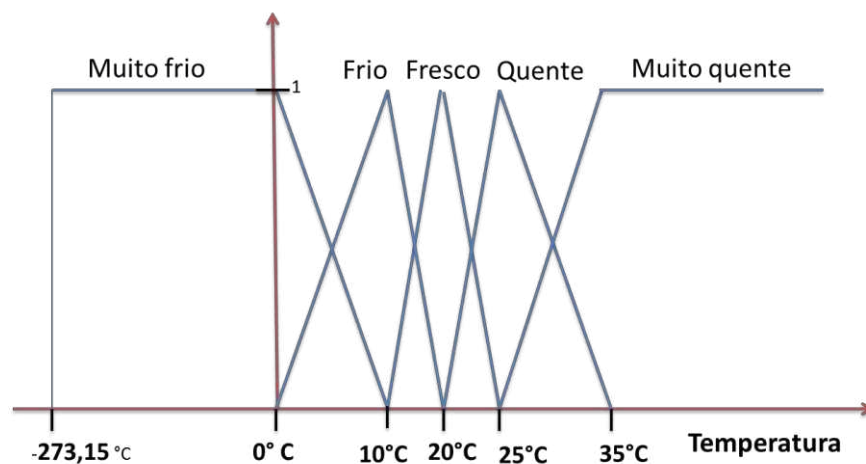


Figura 2.5 - Função pertinência utilizada para classificar a temperatura ambiente em Muito frio, Frio, Fresco, Quente e Muito Quente.

Para a utilização dos conjuntos difusos em uma lógica, estes conjuntos que são definidos através de suas funções pertinências, e não sobre ação da lógica booleana como nos conjuntos convencionais, devem-se definir formalmente as principais interações entre conjuntos como união, intersecção e complemento.

Para a teoria clássica de conjuntos dado dois conjuntos A e B, a intersecção entre eles são todos elementos que pertencem ao mesmo tempo ao conjunto A e ao B, Já a união são todos os elementos que pertencem ao conjunto A ou ao B. O complemento de um conjunto é definido como todos os elementos que não pertencem ao conjunto.

Observa-se então que uma forma natural de se definir as principais operações entre conjuntos *fuzzy* na Figura 2.6 para dois conjuntos difusos A e B, pois são quase as mesmas operações descritas anteriormente para os conjuntos clássicos. Logo, o formalismo matemático para a definição dessas importantes operações com conjuntos é mostrado na Equação (2.22).

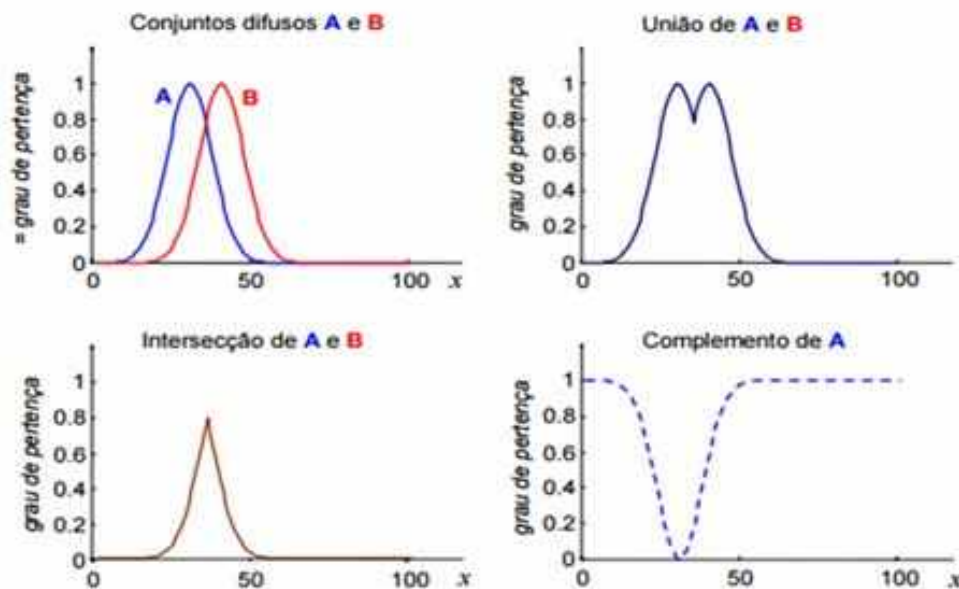


Figura 2.6 - Formalismo dos conjuntos difusos e interações lógicas entre esses conjuntos definidos pelas suas funções pertinências. Adaptado (Silva, G. V. M., 2006).

$$\begin{cases} \mu_{A \cap B} = \min(\mu_A, \mu_B) \\ \mu_{A \cup B} = \max(\mu_A, \mu_B) \\ \mu_{\bar{A}} = 1 - \mu_A \end{cases} \quad (2.22)$$

Para a utilização desse conceito em um controlador, o controlador fuzzy apresenta a estrutura mostrado na Figura 2.7. Primeiro a fuzzificação das variáveis, posteriormente a regra de inferência sobre as variáveis de entrada gerando a(s) variável(s) de saída e em seguida a defuzzificação dessas variáveis.

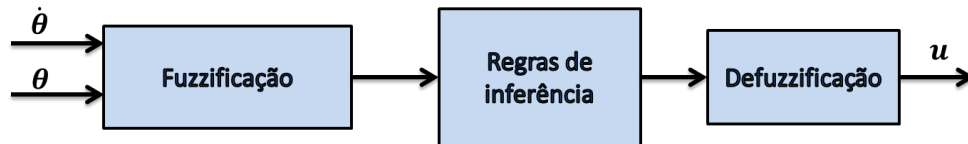


Figura 2.7 - Esquema simplificado do controlador *Fuzzy*.

Para a utilização dessa estrutura de controlador devem-se seguir os seis passos abaixo.

1. Definição das variáveis de entrada e saída.

Nessa etapa, devem-se definir quais são as variáveis importantes que serão utilizadas nas regras para o controle. No exemplo da temperatura, acrescentando outras variáveis como umidade relativa e pressão atmosférica seria possível tentar decidir se vai ou não chover naquele dia.

Neste caso, as variáveis de entrada são a temperatura ambiente, pressão atmosférica e a umidade relativa do ar. Através dessas variáveis utilizando as regras de inferência que um especialista em clima determina, pode-se relacionar com a variável de saída a chance de chover no dia, que pode ser classificada, por exemplo, em altas, medianas ou baixas.

2. Definição do universo de discurso.

O universo de discurso é a definição de quais os valores máximos e mínimos que uma variável pode assumir. Por exemplo, na Figura 2.5 pode-se definir o limite inferior para a temperatura como sendo o zero absoluto (-273.15 °C). Já o limite superior, pode-se utilizar a máxima temperatura que já foi registrada na região onde ocorre a análise.

3. Definição das funções pertinências

A definição das funções pertinências de cada conjunto é a avaliação de quanto uma variável pertence a cada um dos conjuntos chamados de variáveis linguísticas utilizadas. Como é mostrado na Figura 2.6 as funções pertinências de cada conjunto ou variável linguística para as possíveis temperaturas.

4. Escolha das conectivas a se utilizar.

Nesta etapa, devem-se escolher como duas ou mais funções pertinências de conjuntos diferentes devem se relacionar com operações básicas de conjuntos, como união e intersecção. Essa etapa é muito utilizada para avaliar simultaneamente mais de uma função

pertinência e mais de uma variável e também para encontrar a decisão do controlador na última etapa.

#### 5. Escolha das regras de inferência.

Para escolher as regras de inferência que relacionam as variáveis de entrada e a decisão da variável de saída, deve-se ter certo conhecimento, mesmo que qualitativo, sobre o sistema que está sendo analisando.

Desse modo, quem faz o projeto do controlador Fuzzy pode decidir nas regras de inferência de acordo com as variáveis de entrada qual deverá ser a variável de saída. Através dessas regras, encontram-se os pesos das funções pertinências das variáveis de saída.

#### 6. Defuzzificação da saída.

Após a escolha de regras, a definição das variáveis linguísticas de saída, das suas funções pertinências e também das conectivas que serão utilizadas, pode-se encontrar a função pertinência total da variável de saída através da união de todas as funções pertinências com seus devidos pesos.

O método de Defuzzificação utilizado é o método do centroide, se encontra a função pertinência total para a variável de saída e encontra-se o centroide da figura formada. A variável de saída que pode ser uma ação de controle ou a chance de chover no dia é a coordenada do centroide da área formada pela função pertinência total, como mostra a Figura 2.8.

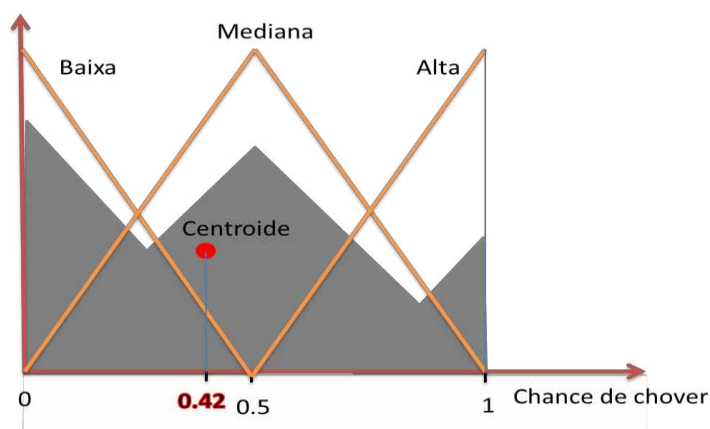


Figura 2.8 - Funções pertinências de saída e também a função pertinência total obtida juntamente com a posição do centroide e a indicação da chance de chover no dia de 42%.



## CAPÍTULO III

### METODOLOGIA

Neste capítulo será apresentado todo o desenvolvimento do projeto, que foi dividido em oito partes. A primeira foi à construção do veículo de duas rodas pêndulo invertido, a segunda foi referente à sintonia experimental de PID, a terceira é referente ao desenvolvimento do modelo dinâmico e a determinação dos parâmetros do modelo, a quarta consiste no projeto de um controlador PID, a quinta o projeto de um controlador *fuzzy*, a sexta parte a aplicação dos controladores na simulação do modelo não linear do veículo, a sétima parte a aplicação dos controladores no veículo de duas rodas real e a oitava parte a proposta de um método de melhoria dos coeficientes de um controlador PID utilizando a simulação.

#### 3.1. Construção do Veículo de Duas Rodas

##### 3.1.1. Estrutura Mecânica

O projeto do veículo de duas rodas acima foi baseado no veículo já existente no laboratório MAPL, e também em um veículo de duas rodas similar comercialmente conhecido como Balanduino (Figura 1.2).

O esboço do projeto da estrutura mecânica encontra-se na Figura (3.1). Nessa imagem encontram-se as principais dimensões planejadas do veículo. Encontram-se também

as disposições dos elementos mecânicos na composição do carrinho e o detalhe da fixação das placas de alumínio utilizando arruelas e porcas.

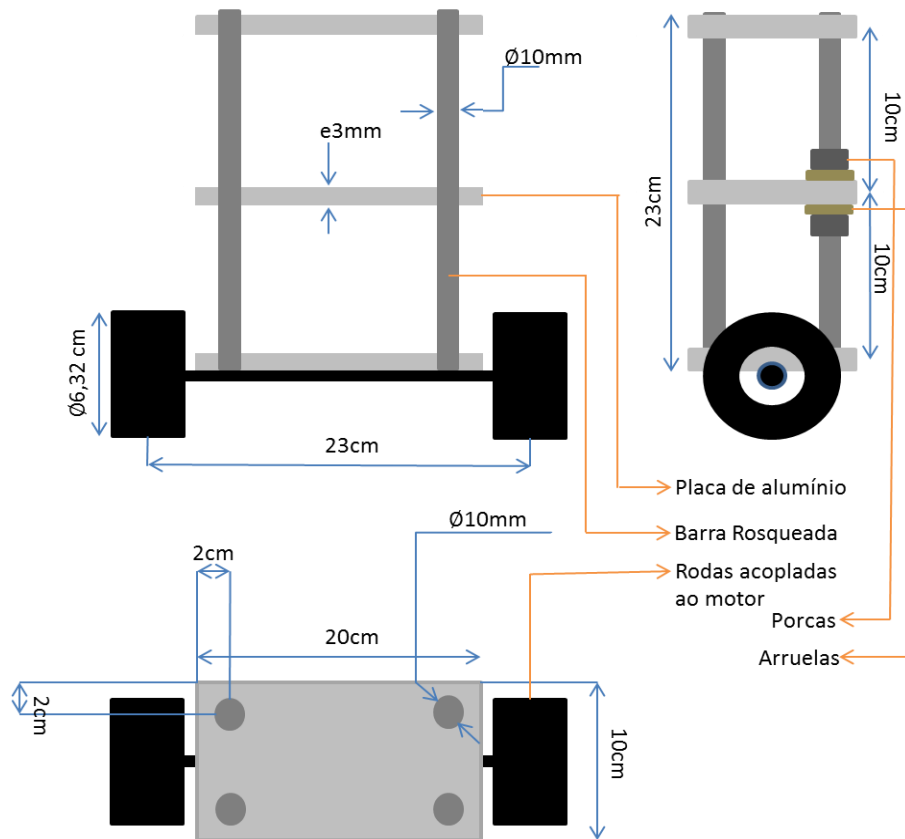


Figura 3.1 - Esboço do desenho do veículo de duas rodas mostrando suas principais dimensões e os principais elementos utilizados. O esboço não está em escala.

Para a confecção do veículo de duas rodas pendulo invertido foi necessária à utilização dos seguintes elementos:

- 3 Placas de Alumínio 10 cm x 20 cm x 3 mm de espessura
- 4 parafusos rosqueados M12 com 23 cm de altura e 10 mm de diâmetro
- 24 arruelas para fixação com diâmetro interno de 15 mm e externo de 30 mm
- 24 porcas M 12 para fixação com diâmetro interno rosqueado 10 mm
- 2 motores com eixos e rodas acoplados para fixação em baixo da estrutura

O veículo tem como objetivo a simplicidade de confecção e montagem e também a facilidade de se variar a geometria e também a massa do veículo (alterando o número de placas de alumínio) para futuros estudos da influência desses parâmetros e no desempenho de controladores.

### 3.1.2. Hardware do veículo de duas rodas

Para o Hardware do veículo de duas rodas foram necessários os seguintes componentes eletrônicos:

- 4 Baterias de Lítio 5800mAh;
- Interruptor simples para ligar e desligar;
- Regulador de tensão *step up* e *step down*;
- Placa Ponte H para 2 motores;
- Acelerômetro e Giroscópio (MPU6050);
- 2 Conversores lógicos bidirecionais 3.3/5 V;
- Arduino DUE;
- 2 motores DC;

A representação esquemática do Hardware do veículo é mostrada na Figura (3.2). Podem-se observar através da figura as ligações entre os componentes eletrônicos. As funções dos componentes eletrônicos mais importantes são especificadas nos subtópicos abaixo.

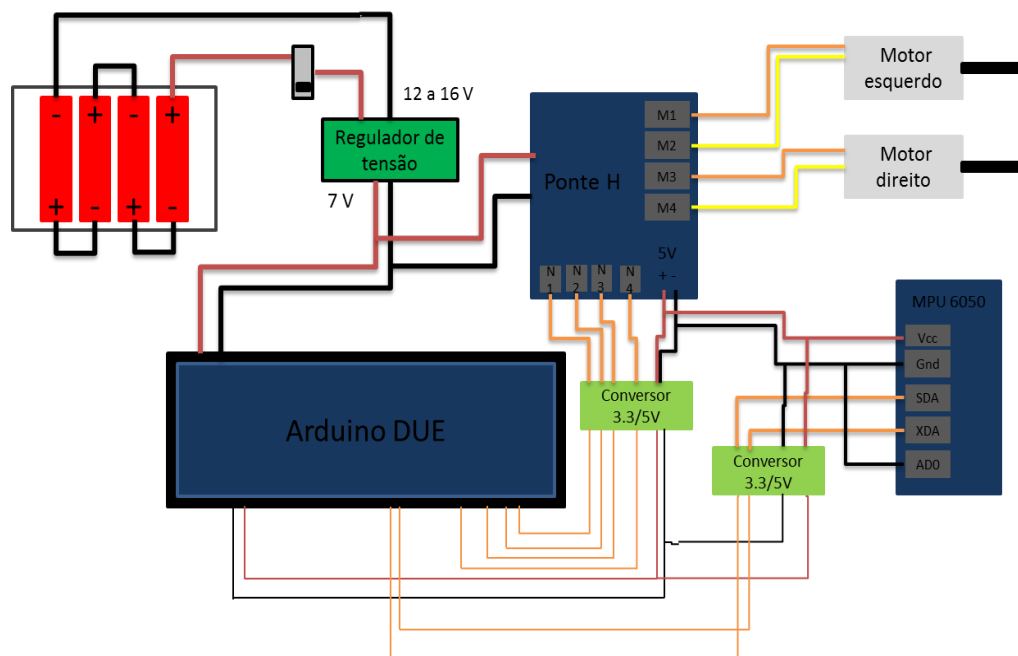


Figura 3.2 - Desenho esquemático mostrando as ligações físicas entre os componentes eletrônicos do Hardware do veículo. Os conversores bidimensionais permitiram a interação entre tecnologias 5.0V e 3.3V.

### 3.1.2.1 Motor DC

O motor DC utilizado no veículo encontra-se na Figura (3.3) juntamente com a roda e dispositivos de suporte na fixação do motor nas placas de alumínio. As especificações do motor fornecidas pelo fabricante encontram-se na Tabela 3.1.



Figura 3.3 - Imagem do motor DC 6.0 V utilizado no veículo. Adaptado de Motor DC, 2017.

Tabela 3.1 - Dados do fabricante sobre a operação do motor (Motor DC, 2017).

|                               |              |
|-------------------------------|--------------|
| Fornecedor                    | FELIPFLOP    |
| Tensão de operação            | 6.0 V        |
| Torque em máxima eficiência   | 2.0 kgf x cm |
| Rotação em máxima eficiência  | 170 RPM      |
| Corrente em máxima eficiência | 0.6 A        |
| Máximo torque                 | 5.2 kgf x cm |
| Rotação em máximo torque      | 110 RPM      |
| Corrente em máximo torque     | 1.1 A        |

Foram utilizados dois desses motores com rodas e suporte para a fixação na placa de alumínio da base do veículo de duas rodas.

### 3.1.2.2 Driver para motor Ponte H L298N

O Driver Ponte H L298N, mostrado na Figura (3.4), permite o acionamento de dois motores DC simultaneamente com tensão entre 0.0 e 6.0 V com a corrente necessária de operação, utilizando a alimentação da ponte e uma saída analógica com modulação em largura de pulso (PWM) no nível de tensão de 5.0 V e baixa corrente gerada no microcontrolador.



Figura 3.4 - Imagem do driver ponte H L298N utilizado. Adaptado de Ponte H, 2017.

### 3.1.2.3 MPU6050

A MPU6050 (visto na Figura 3.5) é um chip de medição inercial contendo um acelerômetro e um giroscópio. A MPU se comunica através de protocolo de comunicação I2C, utilizando nível de tensão de 3.3 a 5.0 V, porém, os testes mostram que a comunicação não é eficaz no nível de tensão 3.3 V, necessitando de um conversor de tensão bidirecional.

A MPU consegue obter as acelerações em todas as direções  $x$ ,  $y$  e  $z$  através do acelerômetro e também consegue obter as velocidades angulares de ambos os eixos através de um giroscópio. Assim, considerando um instante sem forças externas atuando sobre o veículo resulta apenas na aceleração gravitacional, o ângulo entre a aceleração total e o eixo Z é o ângulo  $\theta$ . A velocidade angular em cada eixo é lida diretamente da MPU.

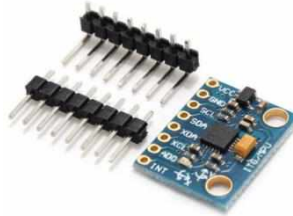


Figura 3.5 - Imagem da MPU6050. Adaptado de MPU6050, 2017

### 3.1.2.4 Arduino DUE

O Arduino DUE, mostrado na Figura 3.6, diferentemente dos outros da família Arduino é baseado em um microcontrolador ARM de 32 bits. O DUE tem um clock de 84 MHz, uma grande quantidade de entradas e saídas e também maior quantidade de pinos de comunicação. Por esses motivos, e também por não ser um microcontrolador caro, foi escolhido para o projeto do veículo.



Figura 3.6 - Imagem do Arduino DUE. Adaptado de Arduino DUE, 2017.

O único inconveniente dessa placa é por utilizar processador ARM, a placa tem nível de tensão 3.3 V. Mas esse problema foi resolvido com os conversores de tensão bidirecionais 3.3/ 5.0 V que possibilitaram a comunicação com a MPU e a ponte H.

#### 3.1.2.4 Conversor lógico bidirecional 3.3/5 V BBS138.

O Conversor de nível lógico bidirecional permite a comunicação entre os dispositivos 3.3 V e 5.0 V, como a comunicação entre o Arduino DUE e a ponte H para enviar o sinal PWM e a comunicação I2C entre o DUE e a MPU6050.

#### 3.1.2.4 Regulador de tensão *step up* e *step down* X16009.

A imagem do regulador de tensão encontra-se na Figura 3.7. O regulador de tensão tipo *step up* e *step down* consegue tanto gerar um nível mais alto de tensão quanto abaixar se necessário o nível da tensão. Dessa forma, utilizando de um potenciômetro no Regulador, é possível ajustar a tensão de saída para 7.0 V com auxílio de um multímetro. Para esse nível de tensão ambos os equipamentos eletrônicos tem suas tensões de operação (Arduino DUE é recomendado uma alimentação acima de 7.0 V).



Figura 3.7 - Regulador de tensão X16009. Adaptado de Regulador de tensão, 2017.

### 3.2. Modelo Dinâmico e Identificação dos Parâmetros do Veículo.

#### 3.2.1. Desenvolvimento do modelo dinâmico do veículo

Para encontrar as equações do modelo dinâmico do veículo, primeiramente é necessário definir uma representação simplificada do veículo. Essa representação com 2 graus de liberdade pode ser mostrada na Figura 3.8. Observa-se na figura a capacidade do veículo se balançar em torno do eixo  $y$  (conferir Figura 3.12 mostrando todos os eixos) e também de deslocar ao longo do eixo  $x$ .

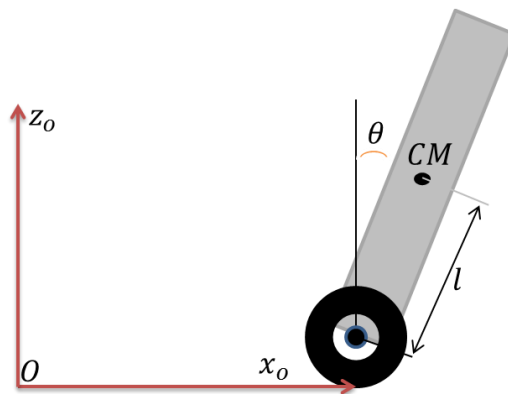


Figura 3.8 - Representação com dois graus de liberdade do veículo de duas rodas. Primeiro grau se refere ao pêndulo invertido na íntegra e o segundo o deslocamento em  $x$ .

Porém o veículo de duas rodas também tem a possibilidade de rotacionar em torno do eixo  $z$ . Dessa forma o veículo tem na verdade três graus de liberdade, um grau de liberdade para cada uma das rodas que são independentes e o terceiro ao seu balanço referente ao a parte suspensa do veículo (como é mostrado na Figura (3.9)).

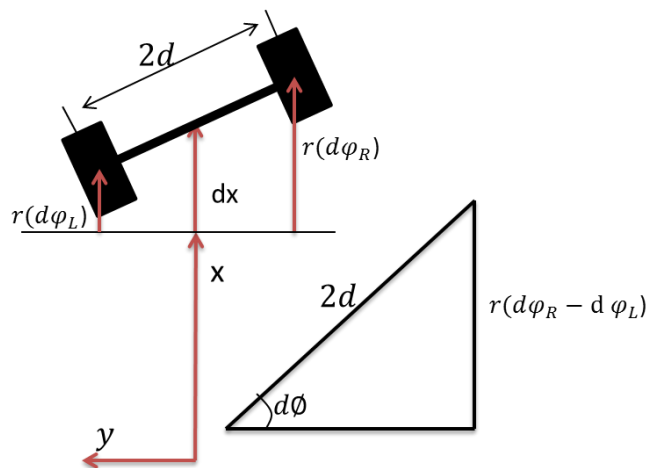


Figura 3.9 - Representação dos dois graus de liberdade da base do veículo de duas rodas

Para o veículo considera-se  $x$  o deslocamento total realizado na direção do eixo  $x$  para o referencial no veículo (referencial normal/tangencial onde o eixo  $x$  representa o eixo tangencial à trajetória e  $y$  o normal a trajetória). Já o ângulo  $\theta$  é o ângulo de inclinação da massa suspensa do carrinho com o vertical. E o ângulo  $\phi$  é o ângulo entre o eixo  $x_o$  fixo no referencial O (mostrado na Figura 3.10) e o referencial  $x$  adotado pelo veículo.

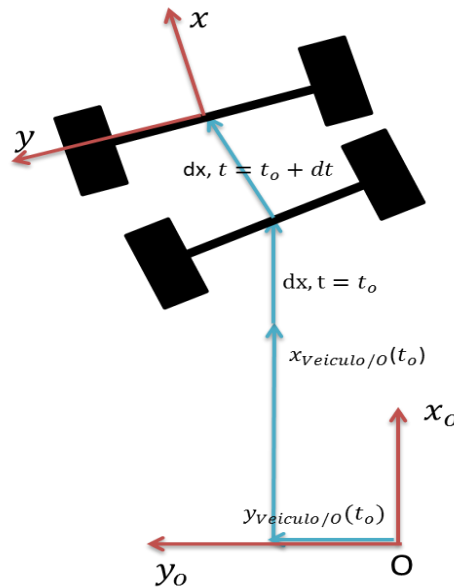


Figura 3.10 - Relação entre o referencial fixo no ponto O e o referencial com orientação fixa no veículo. Observa-se que o referencial do carrinho acompanha a orientação da trajetória, enquanto o inercial não.

A relação entre esses referenciais podem ser descritas pelas Equações (3.1) e (3.2).

$$dx_{\text{veiculo}/O} = dx \cdot \cos(\phi) \quad (3.1)$$

$$dy_{\text{veiculo}/O} = dx \cdot \sin(\phi) \quad (3.2)$$

Dessa forma a posição do veículo em relação ao referencial inercial pode ser obtida pelas Equações (3.1) e (3.2) referentes à posição  $x$  e  $y$  respectivamente do veículo em relação ao referencial adotado. Considerando que o veículo não sofra escorregamento e que o atrito no pneu do veículo também se oponha a qualquer força normal, pode-se, sem perdas de informações dinâmicas, representar o veículo pelos graus de liberdade  $(x, \theta, \phi)$ .



Essas variáveis descritas no parágrafo anterior se relacionam com as rotações independentes de cada uma das rodas pelas Equações (3.3) e (3.4), como se pode notar através da Figura 3.9.

$$x = (r/2).(\varphi_R + \varphi_L) \quad (3.3)$$

$$\phi = (r/2d).(\varphi_R - \varphi_L) \quad (3.4)$$

Pode-se assim, descrever a posição do centro de massa da parte suspensa do veículo, como se pode observar pela Figura 3.8, através das Equações (3.5) e (3.6).

$$x_p = x + l \sin \theta \quad (3.5)$$

$$z_p = 2r + l \cos \theta \quad (3.6)$$

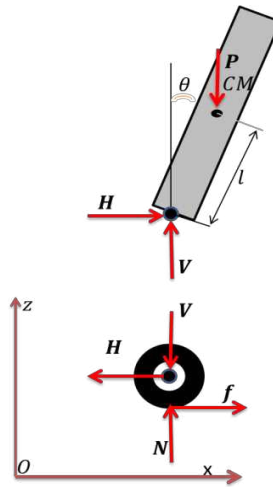


Figura 3.11 - Diagrama de corpo livre das forças que atuam no veículo de duas rodas. As forças que atuam na base e também as forças que atuam na parte suspensa do veículo de duas rodas pendulo invertido.

A partir das Equações (3.5) e (3.6) e a equação de movimento descrita na fundamentação teórica, as Figuras (3.9) e (3.10), encontra-se as equações dinâmicas da base e da parte suspensa do veículo de duas rodas descritas pelas Equações (3.7), (3.8), (3.9), (3.10) e (3.11).

As Equações (3.7) e (3.8) são a segunda lei de Newton para a translação aplicada respectivamente a aceleração  $x$  e a aceleração  $z$  do centro de massa da parte suspensa do

veículo. Já a Equação (3.9) diz respeito à segunda lei de Newton para a rotação da parte suspensa do veículo de duas rodas.

$$m \frac{d^2}{dt^2} (x + l \cdot \sin \theta) = H = \ddot{x} - l \sin \theta \cdot \dot{\theta}^2 + l \cos \theta \cdot \ddot{\theta} \quad (3.7)$$

$$m \frac{d^2}{dt^2} (2r + l \cdot \cos \theta) = V - m \cdot g = -l \cos \theta \cdot \dot{\theta}^2 - l \sin \theta \cdot \ddot{\theta} \quad (3.8)$$

$$J_p \frac{d^2 \theta}{dt^2} = V \cdot l \sin \theta - H \cdot l \cos \theta - C_a (\dot{\theta} - \dot{\phi}_R) - C_a (\dot{\theta} - \dot{\phi}_L) \quad (3.9)$$

A equação dinâmica para a base do veículo pode ser representada com as Equações (3.10) e (3.11), onde a primeira é a aplicação da Segunda lei de Newton para a translação em x e a segunda é a aplicação da Segunda lei para a rotação da base do veículo em relação ao eixo z.

$$M \frac{d^2 x}{dt^2} = f - H \quad (3.10)$$

$$J_z \frac{d^2 \phi}{dt^2} = d \cdot F_R - d \cdot F_L \quad (3.11)$$

A força líquida sobre a base do veículo é mostrada na Equação (3.12), e pode ser compreendida como a soma da força líquida produzida em cada um dos motores, que é a força gerada pelo torque descontando os efeitos do torque de inércia necessário para acelerar a roda, o eixo e as engrenagens. E descontando também o atrito dinâmico inerente aos motores. A Equação (3.12) é obtida através da Equação (2.4).

$$f = F_R + F_L = \frac{l}{r} \cdot (T_R - J_w \ddot{\phi}_R - b \dot{\phi}_R) + \frac{l}{r} \cdot (T_L - J_w \ddot{\phi}_L - b \dot{\phi}_L) \quad (3.12)$$

Utilizando uma “força esperada” definida pela Equação (3.13), que tem como o significado a força que seria transmitida a base do carrinho se não houvesse atrito interno ao motor e também se as inércias de transmissão e da roda fossem desprezíveis.

$$f^* = \frac{(T_R + T_L)}{r} \quad (3.13)$$

As Equações (3.14) e (3.15) são obtidas através das Equações (3.3) e (3.4). Agora as variáveis cinemáticas das rodas no modelo podem ser substituídas pelas variáveis cinemáticas ( $x$ ,  $\theta$  e  $\phi$ ).

$$\begin{cases} \varphi_R = \frac{x}{r} + \frac{d \cdot \phi}{r} \\ \dot{\varphi}_R = \frac{\dot{x}}{r} + \frac{d \cdot \dot{\phi}}{r} \\ \ddot{\varphi}_R = \frac{\ddot{x}}{r} + \frac{d \cdot \ddot{\phi}}{r} \end{cases} \quad (3.14)$$

$$\begin{cases} \varphi_L = \frac{x}{r} - \frac{d \cdot \phi}{r} \\ \dot{\varphi}_L = \frac{\dot{x}}{r} - \frac{d \cdot \dot{\phi}}{r} \\ \ddot{\varphi}_L = \frac{\ddot{x}}{r} - \frac{d \cdot \ddot{\phi}}{r} \end{cases} \quad (3.15)$$

A força total atuando na base do veículo pode então ser escrita pela Equação (3.16).

$$f = f^* - \frac{2J_W}{r^2} \ddot{x} - \frac{2b}{r^2} \dot{x} \quad (3.16)$$

O movimento de rotação da base do veículo pode ser reescrita pela Equação (3.17)

$$J_z \frac{d^2 \phi}{dt^2} = d \cdot \frac{(T_R - T_L)}{r} - 2J_W \left( \frac{d}{r} \right)^2 \ddot{\phi} - 2b \left( \frac{d}{r} \right)^2 \dot{\phi} \quad (3.17)$$

Isolando as forças V e H nas Equações (3.7) e (3.8) e substituindo na Equação (3.9), substituindo também as Equações (3.14) e (3.15) na Equação (3.9). Substituindo a Equação (3.16) na Equação (3.10) e reescrevendo a Equação (3.17) após algumas manipulações matemáticas, encontram-se o sistema de equações diferenciais na Equação (3.18) que

descrevem o comportamento do veículo de duas rodas em relação as suas principais variáveis de estado ( $x$ ,  $\theta$  e  $\phi$ ).

$$\begin{cases} (J_p + ml^2)\ddot{\theta} + (ml \cos \theta)\ddot{x} = m \lg \sin \theta - 2C_a \dot{\theta} + 2(C_a / r)\dot{x} \\ (ml \cos \theta)\ddot{\theta} + (M + m + 2J_w / r^2)\ddot{x} = f^* - (2b / r^2)\dot{x} + ml \sin \theta \dot{\theta}^2 \\ (J_z + 2(d / r)^2 J_w)\ddot{\phi} = (d / r^2)(T_R - T_L) - 2b(d / r)^2 \dot{\phi} \end{cases} \quad (3.18)$$

Para encontrar as posições, velocidades e acelerações ao longo do tempo são necessárias às soluções de ambas as três equações diferenciais acima simultaneamente. Devido a ser um sistema acoplado e também não linear, não existem soluções analíticas para esse sistema, necessitando assim de um método numérico.

Para simplificar as Equações (3.18) acima, e também para a utilização de técnicas de controle clássicas são necessárias equações diferenciais lineares, sendo assim as equações em questão podem ser linearizadas considerando que o ângulo de inclinação seja menor que  $5^\circ$  e que também a velocidade angular seja pequena. Utilizando aproximação de séries de Taylor nessas condições:

$$\sin(\theta) \approx \theta, \quad \cos(\theta) \approx 1, \quad \dot{\theta}^2 = 0.$$

E o modelo linearizado pode ser representado pelo sistema de equações descritas na Equação (3.19).

$$\begin{cases} (J_p + ml^2)\ddot{\theta} + (ml)\ddot{x} = m \lg \theta - 2C_a \dot{\theta} + 2(C_a / r)\dot{x} \\ (ml)\ddot{\theta} + (M + m + 2J_w / r^2)\ddot{x} = f^* - (2b / r^2)\dot{x} \\ (J_z + 2(d / r)^2 J_w)\ddot{\phi} = (d / r^2)(T_R - T_L) - 2b(d / r)^2 \dot{\phi} \end{cases} \quad (3.19)$$

### 3.2.2. Cálculos e identificação dos parâmetros do veículo

O veículo de duas rodas pêndulo invertido pode ser representado de forma mais detalhada pela Figura 3.12. Na figura as massas dos componentes eletrônicos e outros componentes necessários para a funcionalidade do veículo são representados como massas M1 e M2 uniformemente distribuídas sobre as duas placas do veículo, essas massas foram consideradas no cálculo dos momentos de inércias do veículo.

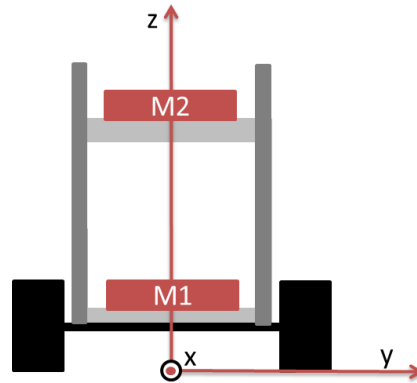


Figura 3.12 - Figura esquemática mostrando a simplificação do veículo considerando as massas dos componentes eletrônicos e demais componentes como massas M1 e M2 concentradas em cada uma das placas do veículo de duas rodas.

A massa total do veículo foi medida utilizando uma balança calibrada com cada uma das partes separadamente já com os componentes eletrônicos fixados à estrutura e posteriormente realizando um soma das massas de todas as partes juntamente com os quatro parafusos rosqueados utilizados na estrutura.

Um experimento realizando equilíbrio do veículo de duas rodas também foi utilizado para a obtenção do valor experimental do centro de massa que foi encontrado de aproximadamente (7.50 cm).

Com os valores das massas das partes individuais e as principais dimensões do veículo, Considerando que cada corpo individual contribui com massa  $M_i$ , com altura  $h_{CMi}$  em relação ao eixo do motor e com momento de inércia em relação ao próprio centro de massa  $J_{CMi}$ . Para os cálculos foram considerados as bases de alumínio, as massas sobre as placas M1 e M2, as barras rosqueadas, parafusos e arruelas. As equações utilizadas são mostradas nas Equações (3.20), (3.21) e (3.22). O programa utilizando em software Matlab para os cálculos se encontra em apêndice I.

$$l = \frac{\sum_{i=1}^N h_{CMi} M_i}{m} = 0.0745m \quad (3.20)$$

$$J = \sum_{i=1}^N (J_{CMi} + h_{CMi}^2 M_i) = 120.10^{-4} \text{ kgm}^2 \quad (3.21)$$

$$J = J_p + ml^2 \rightarrow J_p = 48.10^{-4} \text{ kgm}^2 \quad (3.22)$$

Como foi descrito no projeto do veículo, na Tabela 3.1 encontra-se os dados do fornecedor do motor e a partir das Equações (2.1), (2.2) e (2.3) do motor DC, pode-se estimar os parâmetros do motor DC utilizado a Equação (3.23) e os valores na tabela.

$$T = \frac{(K_m V_{cc})}{R} - \frac{(K_m K_e) \dot{\phi}_{eixo}}{R} = A + B \dot{\phi}_{eixo} \quad (3.23)$$

Utilizando regressão linear utilizando os dados fornecidos pelo fabricante encontra-se:  $A=1.107$  e  $B= -0.052$ .

Considerando que o torque seja aproximadamente nulo na velocidade máxima do eixo do motor, encontra-se um valor da constante  $K_e$  de 0.28.

Reescrevendo os valores das constantes na Equação (3.24), encontra-se:

$$A \rightarrow \frac{K_m}{R} = 0.1845 \quad B \rightarrow \frac{K_m}{R} = 0.1869 \quad (3.24)$$

Considerando a Equação (1.3) que relaciona  $K_m$  e o torque, encontram-se valores de  $K_m$  entre 0.33 e 0.57.

Utilizando o modelo dinâmico linear que foi obtido acima, utilizando os valores que foram obtidos acima dentro de uma margem, e utilizando um algoritmo de otimização baseado na lógica *neuro-fuzzy* (Pereira, B.L, 2017), foi possível obter os melhores valores de cada parâmetro que se ajustasse no modelo de forma a obter o menor erro entre a resposta do modelo e a resposta real do veículo de duas rodas. Os valores encontrados através desse algoritmo são mostrados na Tabela 3.2. O gráfico comparando a resposta real com a resposta simulada para esses parâmetros encontra-se na Figura 3.13.

Tabela 3.2 - Parâmetros ótimos do veículo de duas rodas para um ajuste de 95%.

| Símbolo | Interpretação física                 | Valor (unidade)                           |
|---------|--------------------------------------|---|
| $J_p$   | Momento de inércia do pêndulo        | $48 \times 10^{-4}$ (kg.m <sup>2</sup> )  |
| $J_w$   | Momento de inércia de cada roda      | $3.1 \times 10^{-5}$ (kg.m <sup>2</sup> ) |
| $g$     | Aceleração da Gravidade              | 9.79 (m/s <sup>2</sup> )                  |
| $r$     | Raio da roda                         | 0.0316 (m)                                |
| $M$     | Massa das duas rodas                 | 0.062 (kg)                                |
| $m$     | Massa do pêndulo                     | 1.304 (kg)                                |
| $d$     | Metade da distância entre as rodas   | 0.115 (m)                                 |
| $l$     | Altura do centro de massa do pêndulo | 0.0748 (m)                                |

|          |                                    |                               |
|----------|------------------------------------|-------------------------------|
| $C_a$    | Fricção dinâmica no eixo das rodas | 0.0005 (kg.m <sup>2</sup> /s) |
| $b$      | Fricção dinâmica no eixo do motor  | 0.002 (kg.m <sup>2</sup> /s)  |
| $V_{cc}$ | Máxima tensão no motor             | 6 (V)                         |
| $K_m$    | Constante do motor                 | 0.54 (Nm/A)                   |
| $K_e$    | Constante do motor                 | 0.33 (Vs)                     |
| $R$      | Resistência interna do motor       | 0.92 (ohm)                    |
| $dz$     | Zona morta do motor                | 0.04 (arb)                    |

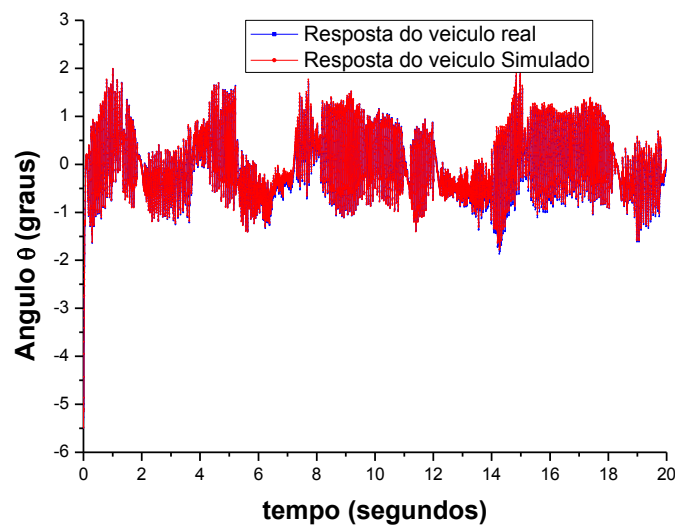


Figura 3.13. Teste para encontrar os parâmetros do modelo do veículo de duas rodas. Em vermelho a resposta com os parâmetros que minimiza o erro médio quadrático no algoritmo de otimização *Neuro-Fuzzy* e em azul a resposta real do veículo utilizada para a identificação.

### 3.3. Sintonia de PID Experimental – PID Z-N.

A sintonia experimental do PID foi realizada utilizando o método de Ziegler-Nichols de Malha Fechada. O pêndulo invertido, como é de amplo conhecimento, é um sistema instável e não linear, sendo assim, em malha aberta, a princípio há apenas uma região de mínimo potencial que é exatamente se  $\theta=0^\circ$  e velocidades nulas (equilíbrio instável). Porém, qualquer estímulo externo faz com que o pêndulo invertido sai dessa região de mínimo local e busque a região de equilíbrio estável, que no caso é  $\theta=180^\circ$ .

Já em malha fechada, com um controlador proporcional, o sistema consegue pelo menos ficar oscilando em torno da posição de equilíbrio instável.

O valor que se iniciou a oscilação foi um valor de constante proporcional de 2.8. Após o início das oscilações os dados foram colhidos utilizando esse controlador e o gráfico desses dados é mostrado na Figura 3.14.

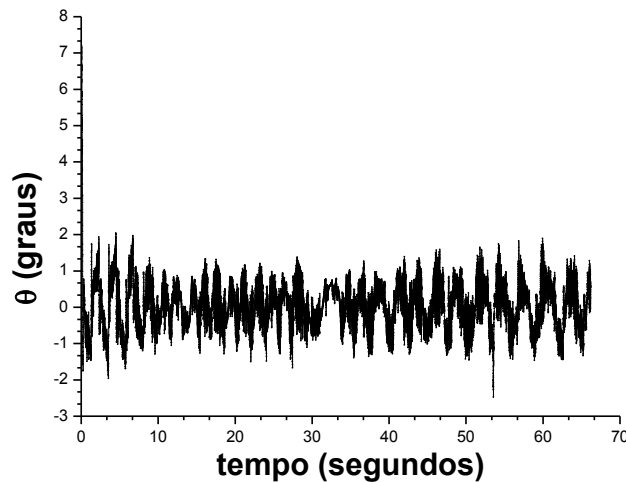


Figura 3.14 - Gráfico do ângulo com a vertical do veículo de duas rodas pendula invertido para o controle puramente proporcional com  $K=2.8$ .

Utilizando o software Matlab, realizando uma transformada discreta de Fourier para os resultados obtidos com o ganho crítico obtém-se o gráfico na figura abaixo. O código utilizado encontra-se no apêndice II.

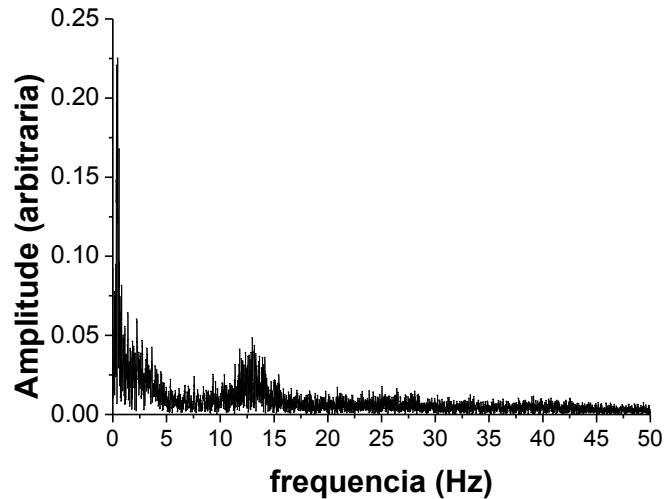


Figura 3.15 - Gráfico da resposta em frequência do ângulo para encontrar o período crítico.

As principais frequências identificadas nos picos do gráfico acima foram de 0.451 Hz e de 12.974 Hz, como o interesse era no menor período, interessa apenas o pico de maior frequência que no caso tem um período de 0.0771 segundos. Logo o  $P_{CRT} = 0.0771$ .

As constantes  $K_p$ ,  $K_i$  e  $K_d$  são calculadas a partir da Tabela 1.1 e seus valores são respectivamente: 1.68, 43.6 e 0.0162.



### 3.4. Projeto de um PID pela Teoria Clássica de Controle – PID PROJ

O diagrama de blocos completo do veículo de duas rodas encontra-se na Figura 3.16. Esse diagrama foi montado com o interesse do controle da estabilidade vertical do pêndulo e no controle da velocidade de translação do veículo. O controle de ambos os graus de liberdade não foi abortado durante o trabalho de conclusão de curso, porém está em processo de implementação para o mesmo veículo.

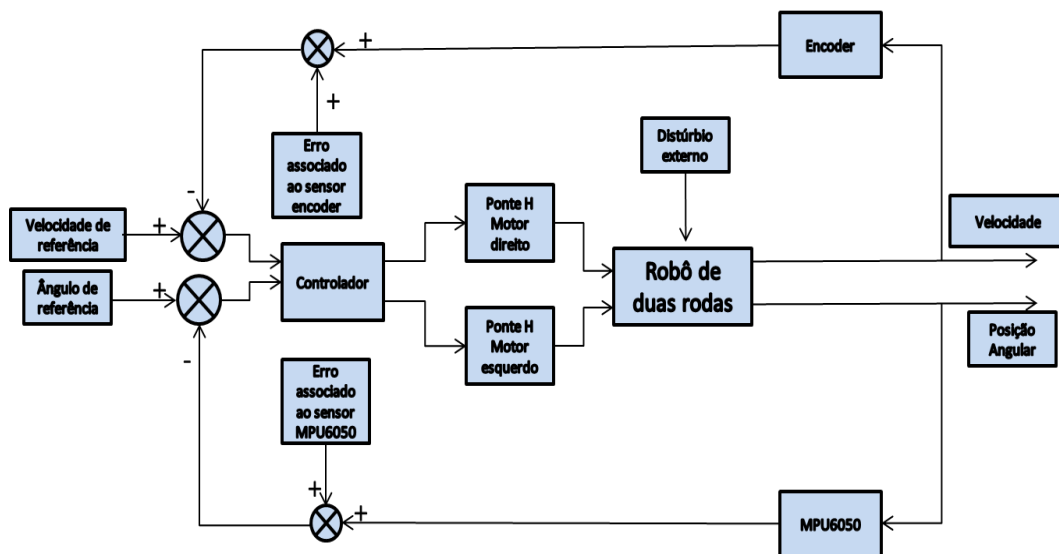


Figura 3.16 - Diagrama de blocos do veículo de duas rodas considerando dois graus de liberdade.

Para o trabalho de conclusão de curso a análise do sistema se compreende apenas na estabilidade estática do veículo que diz respeito apenas ao equilíbrio vertical do pêndulo invertido. O diagrama de blocos que foi utilizado nesse caso é mostrado na Figura 2.4.

Para o projeto do controlador primeiramente é necessário a obtenção da função transferência do veículo de duas rodas. Para isto são necessárias as transformadas de Laplace do sistema de equações na Equação (3.19) que descrevem o comportamento dinâmico do veículo de duas rodas dentro da sua região de linearização dado uma ação de controle. Após a transformada para o domínio  $S$  encontra-se as equações abaixo considerando as condições iniciais nulas. Para simplificação considera-se o conjunto rodas e motores idênticos e também a mesma ação de controle para ambos os motores.

$$\begin{cases} (J_p + ml^2)s^2\Theta(s) + (ml)s^2X(s) = mlg\Theta(s) - 2C_aS\Theta(s) + 2(C_a/r)sX(s) \\ (ml)s^2\Theta(s) + (M + m + 2J_w/r^2)s^2X(s) = F^*(s) - (2b/r^2)sX(s) \\ F^*(s) = 2T(s)/r \\ T(s) = K_m(V_{cc}U(s) - K_eX(s)/r)/R \end{cases} \quad (3.25)$$

Reescrevendo o sistema acima da Equação (3.25) juntando as variáveis encontra-se a Equação (3.26):

$$\begin{cases} [(J_p + ml^2)s^2 + 2C_aS - mgl]\Theta(s) = [-(ml)s^2 + 2(C_a/r)s]X(s) \\ [(M + m + 2J_w/r^2)s^2 + (2b/r^2)s]X(s) = F^*(s) - (ml)s^2\Theta(s) \\ F^*(s) = 2K_m(V_{cc}U(s) - K_e(s)X(s)/r)/(rR) \end{cases} \quad (3.26)$$

Reescrevendo novamente após substituir o termo da força na segunda equação do sistema da Equação (3.26) encontra-se o sistema da Equação (3.27):

$$\begin{cases} [(J_p + ml^2)s^2 + 2C_aS - mgl]\Theta(s) = [-(ml)s^2 + 2(C_a/r)s]X(s) \\ [(M + m + 2J_w/r^2)s^2 + \left(2b/r^2 + \frac{2K_mK_eX(s)}{Rr^2}\right)s]X(s) = \frac{2K_mV_{cc}U(s)}{Rr} - (ml)s^2\Theta(s) \end{cases} \quad (3.27)$$

Por fim, substituindo o X(s) na primeira equação e após algumas manipulações matemáticas no sistema descrito na Equação (3.27), obtém-se o sistema a relação entre o ângulo de saída e a ação de controle no domínio s, descrito pela Equação (3.28).

$$\begin{aligned} & \left[ (J_p + ml^2)s^2 + 2C_aS - mgl + \frac{[-(ml)s + 2(C_a/r)]mls^2}{\left[ (M + m + 2J_w/r^2)s + \left(2b/r^2 + \frac{2K_mK_e}{Rr^2}\right) \right]} \right] \Theta(s) \\ &= \frac{[-(ml)s^2 + 2(C_a/r)s] \left( \frac{2K_mV_{cc}U(s)}{Rr} \right)}{\left[ (M + m + 2J_w/r^2)s + \left(2b/r^2 + \frac{2K_mK_e}{Rr^2}\right) \right]} \end{aligned} \quad (3.28)$$

Considerando os primeiros termos da expansão em série de Taylor da Equação (2.21) para um tempo de amostragem de 2 ms observa-se o gráfico na Figura (3.17) contendo a comparação entre a função total e a linearização para o segurador de ordem zero.

$$ZOH(s) = T + \frac{T^2}{2}s \quad (3.29)$$

A Equação (3.29) é a aproximação de primeira ordem do segurador de ordem zero na Equação (2.21), utilizando-se o software Matlab utilizando o código em apêndice III. Encontra-se a função transferência para o ângulo com a vertical do veículo de duas rodas mostrado na Equação (3.30) com três pólos em -332, -8.72 e 9.04. E dois zeros em 0 e 1000.

$$G(s) = \frac{0.003689s^2 - 3.689s}{s^3 + 416.8s^2 - 180.7s - 32900} \quad (3.30)$$

A técnica escolhida para o projeto do controlador é o método pelo root locus. A função transferência de um controlador PID é mostrada na Equação (3.31).

$$G_C(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} = K_d \frac{(s^2 + bs + c)}{s} \quad (3.31)$$

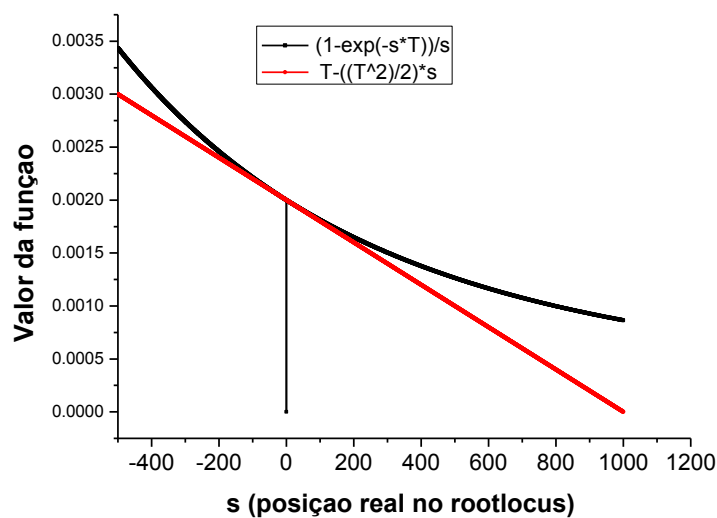


Figura 3.17 - Gráfico da função transferência do segurador de ordem zero e a linearização da função transferência. Observa-se que se mantém próximo as funções para um tempo de amostragem de 2ms em quase toda a região entre (-400 e 1000).

Primeiramente, definindo os requisitos de projeto com *overshoot* máximo de 30% e tempo de acomodação de 1.25 s, encontra-se o pólo de projeto através das equações (2.16), (2.17) e (2.18). Encontra-se o polo de projeto:

$$P_{PRJ} = -3.2000 + 8.3499i \quad (2.32)$$

O *root locus* da função de transferência com o pólo no zero proveniente do fator integrativo é mostrado na Figura (3.18). Adicionando dois zeros complexos conjugados próximos ao pólo de projeto, o root locus muda e se observa a formação de uma curva aberta próxima aos pólos de projetos vindos de uma separação entre os dois pólos do lado esquerdo em direção aos zeros adicionados como mostra à Figura (3.19).

Os dois zeros complexo próximos ao pólo de projeto foram escolhidos, pois facilitam que o root locus passe pelo pólo de projeto. Comparando a Equação (3.33) e a Equação (3.31), observa-se que com essa estrutura é possível obter quaisquer valores das constantes  $b$  e  $c$  positivas. Como as constantes  $K_p$   $K_I$   $K_D$  do PID são positivas, não há perda de informações ao reescrever dessa maneira.

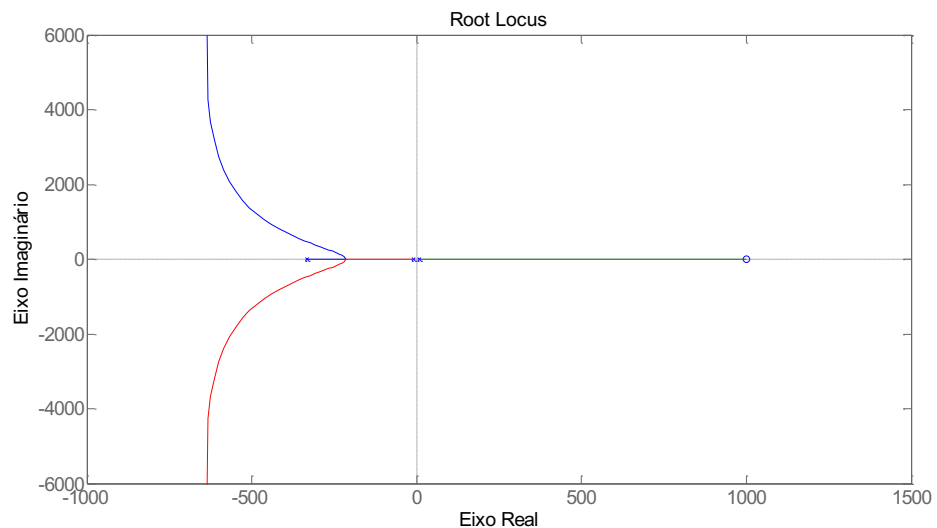


Figura 3.18 - Root locus da função de transferência do veículo de duas rodas com um pólo no zero proveniente da parcela integrativa do controlador.

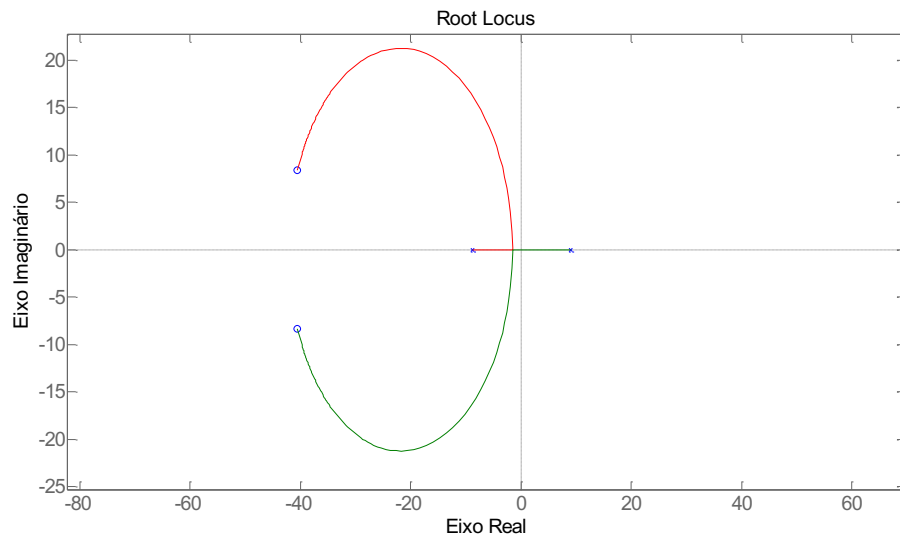


Figura 3.19 - Root locus da função de transferência do veículo de duas rodas com um pólo no zero e dois zeros complexos próximos ao pólo de projeto.

$$G_C(s) = K_C \frac{(s^2 + 2as + (a^2 + b^2))}{s} = K_C \frac{(s + a - bi)(s + a + bi)}{s} \quad (3.33)$$

Considerando  $b=4$  e utilizando e considerando que a fase do controlador e da planta no pólo de projeto deve ser  $-180$  como diz a Equação (2.19) e utilizando a Equação (3.33) acima do controlador obtém-se um controlador PID com constantes:

$$\begin{cases} K_p = 0.6K_{CRT} = 1.675 \\ K_i = 1/(0.5P_{CRT}) = 25.07 \\ K_d = 0.125P_{CRT} = 0.02807 \end{cases} \quad (3.34)$$

Em apêndice IV encontra-se uma demonstração da representação da função transferência do controlador pode-se ser aplicada no caso em estudo e no apêndice V se encontra o programa em Matlab que foi utilizado para realizar os cálculos que acharam as constantes do controlador PID.

### 3.5. Projeto de um controlador *Fuzzy*

O projeto do controlador *Fuzzy* mantém os mesmos passos indicados na fundamentação teórica da lógica *Fuzzy*. Desse modo, são construídos os seguintes passos:

- Escolha das variáveis de entradas e saídas:

Como o foco do trabalho inicialmente é apenas a estabilidade estática vertical do veículo de duas rodas, as variáveis de entrada escolhidas foram: a posição angular e a velocidade angular  $\theta, \dot{\theta}$ .

Utilizou-se apenas uma variável de saída, a ação de controle  $u$ .

- Definição dos Universos de discursos.

Como o veículo vai se movimentar em um plano, o ângulo máximo e mínimo em caso de uma queda é respectivamente  $90^\circ$  e  $-90^\circ$  com a vertical. Logo:

$$-\pi/2 \leq \theta \leq \pi/2 \quad (3.35)$$

O universo da velocidade angular pode ser obtido utilizando a Equação abaixo que é encontrada desconsiderando todas as perdas no modelo para o cálculo da maior variação de velocidade angular durante qualquer situação de operação do veículo.

$$\left( J_p + ml^2 - \frac{(ml)^2}{(M + m + 2J_w / r^2)} \right) \ddot{\theta}_{\max} = -(ml) \frac{2V_{cc} K_m u_{\max}}{Rr} \quad (3.36)$$

$$\begin{cases} \ddot{\theta}_{\max} = -Cont.u_{\max} \\ Const. = (ml) \frac{2V_{cc} K_m}{Rr} / \left( J_p + ml^2 - \frac{(ml)^2}{(M + m + 2J_w / r^2)} \right) = 2267 \end{cases} \quad (3.37)$$

Considerando uma integração discreta em um único intervalo de tempo, obtém-se a Equação (3.38). Considerando o tempo de amostragem de 2 ms e ação de controle máxima como +1 ou -1.

$$\int \ddot{\theta}_{max} dt = -Const. \int u_{max} dt \quad (3.38)$$

$$\dot{\theta}_{max} = -Const. u_{max} T_{amostragem} = \pm 4.534 rad / s \approx 260^\circ / s$$

$$-260 < \dot{\theta} < 260 \text{ } [^\circ / s] \quad (3.39)$$

- Definição das funções pertinências.

As variáveis linguísticas podem ser definidas nas regiões do Ângulo  $\theta$  e da velocidade angular  $\dot{\theta}$  dentro dos seus Universos como mostra as Figuras 3.20 e 3.21. Onde MN é muito negativo, PN é pouco negativo, ZR é centrado, PP é pouco positivo e MP é muito positivo.

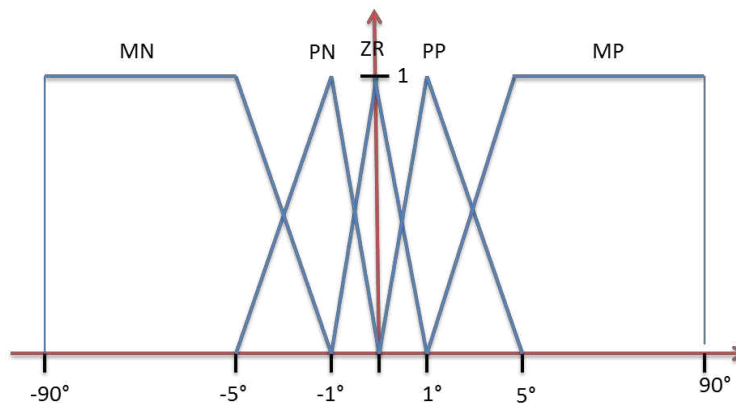


Figura 3.20 - Funções pertinência para ângulo com a vertical, valores de ângulo em graus.

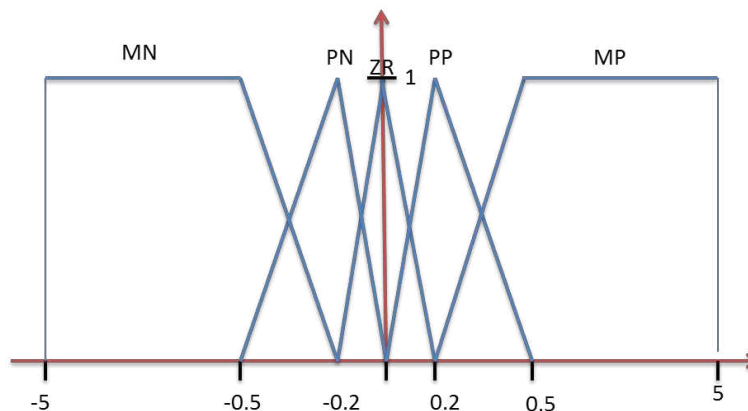


Figura 3.21 - Funções pertinência para a velocidade angular com a vertical, valores de ângulo em radianos/s.

- Escolha das conectivas a utilizar.

Para o controlador Fuzzy projetado as conectivas necessárias serão de união e intersecção definidas como:

$$\begin{aligned}\mu_{A \cup B} &= \max(\mu_A, \mu_B) \\ \mu_{A \cap B} &= \min(\mu_A, \mu_B)\end{aligned}\tag{3.40}$$

- Escolha das regras de inferência.

A variável “Fuzzificada” de saída é obtida através da relação linguística da Tabela 3.3 e dos pesos utilizados nas conectivas das variáveis linguísticas de entrada.

Tabela 2.3 - Relações entre as variáveis linguísticas de entradas criando a variável linguística de saída  $u$ .

|          |    | $\dot{\theta}$ |    |    |    |    |
|----------|----|----------------|----|----|----|----|
|          |    | $u$            | MN | PN | ZR | PP |
| $\theta$ | MN | MN             | MN | MN | PN | ZR |
|          | PN | MN             | MN | PN | ZR | PP |
|          | ZR | MN             | PN | ZR | PP | MP |
|          | PP | PN             | ZR | PP | MP | MP |
|          | MP | ZR             | PP | MP | MP | MP |

A função pertinência total da entrada do ângulo, será representada por  $X_1$  enquanto a segunda entrada a velocidade angular será representada por  $X_2$ . E a função total pode ser representada como:

$$X_1 = \mu_{MN1}X_1^{MN} + \mu_{PN1}X_1^{PN} + \mu_{ZR1}X_1^{ZR} + \mu_{PP1}X_1^{PP} + \mu_{MP1}X_1^{MP}\tag{3.41}$$

$$X_2 = \mu_{MN2}X_2^{MN} + \mu_{PN2}X_2^{PN} + \mu_{ZR2}X_2^{ZR} + \mu_{PP2}X_2^{PP} + \mu_{MP2}X_2^{MP}\tag{3.42}$$

Dessa forma um ângulo de qualquer pode ser representado com suas funções pertinências de cada uma das variáveis linguísticas que representam o ângulo, por exemplo,  $3^\circ$  pode ser representado na lógica *fuzzy* como pertencendo 50% ao conjunto dos ângulos pouco positivos e 50% ao conjunto dos ângulos muito positivos.

Agora, devem-se definir as funções pertinências da ação de controle de modo semelhante com o que foi realizado com o ângulo e a velocidade angular (Figura 3.22).



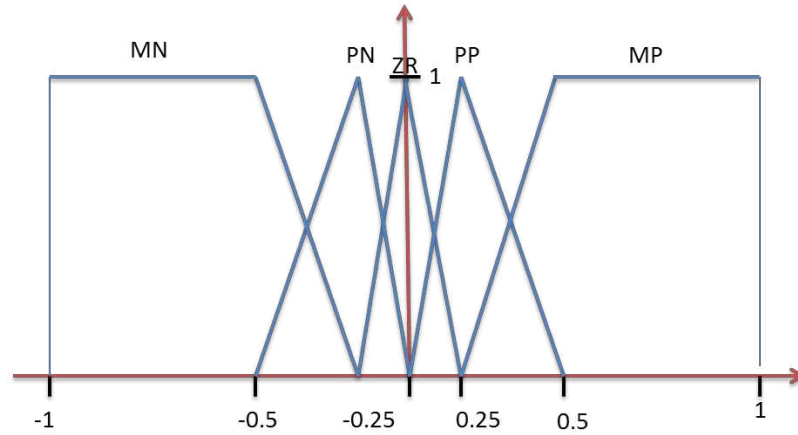


Figura 3.22 - Funções pertinências triangulares para a ação de controle.

- “Defuzzificação” da Saída.

Da Tabela 3.3 acima, e considerando a saída  $Y$ , representada da mesma forma que ambas as entradas, pode-se concluir:

$$\left\{ \begin{array}{l} \mu_Y = \mu_{MNS} Y^{MN} \cup \mu_{PNS} Y^{PN} \cup \mu_{ZRS} Y^{ZR} \cup \mu_{PPS} Y^{PP} \cup \mu_{MPS} Y^{MP} \\ \mu_{MNS} = \mu_{MN1} \cdot (\mu_{MN2} + \mu_{PN2} + \mu_{ZR2}) + \mu_{PN1} \cdot (\mu_{MN2} + \mu_{PN2}) + \mu_{ZR1} \cdot \mu_{MN2} \\ \mu_{PNS} = \mu_{MN1} \mu_{PP2} + \mu_{PN1} \mu_{ZR2} + \mu_{ZR1} \mu_{PN2} + \mu_{PP1} \mu_{MN2} \\ \mu_{ZRS} = \mu_{MN1} \mu_{MP1} + \mu_{PN1} \mu_{PP2} + \mu_{ZR1} \mu_{ZR2} + \mu_{PP1} \mu_{PN2} + \mu_{MP1} \mu_{MN2} \\ \mu_{PPS} = \mu_{MP1} \mu_{PN2} + \mu_{PP1} \mu_{ZR2} + \mu_{ZR1} \mu_{PP2} + \mu_{PN1} \mu_{MP2} \\ \mu_{MPS} = \mu_{MP1} \cdot (\mu_{MP2} + \mu_{PP2} + \mu_{ZR2}) + \mu_{PP1} \cdot (\mu_{MP2} + \mu_{PP2}) + \mu_{ZR1} \cdot \mu_{MN2} \end{array} \right. \quad (3.43)$$

Após encontrar a função pertinência total que é definida com a união das funções pertinências de cada uma das variáveis linguísticas da ação de controle. E também utilizando o centroide da área gerada com essa função pertinência encontra-se qual a ação de controle que deverá atuar no sistema.

$$u = \frac{\int_{-1}^1 u \cdot \mu(u) du}{\int_{-1}^1 \mu(u) du} \approx \frac{\sum_{-1}^1 u_i \cdot \mu(u_i)}{\sum_{-1}^1 \mu(u_i)} \quad (3.44)$$

### 3.6. Simulação do Modelo Linear e do Não Linear.

O fluxograma utilizado na simulação é mostrado na Figura 3.23. A estrutura do programa principal é montada para realizar a integração numérica a partir de dados como tempo de amostragem, tempo do passo de integração, condições iniciais, o tempo total de simulação. O simulador utiliza do método numérico RK4 para obter o vetor de estado  $r$  no tempo como é mostrado na Figura 3.23.

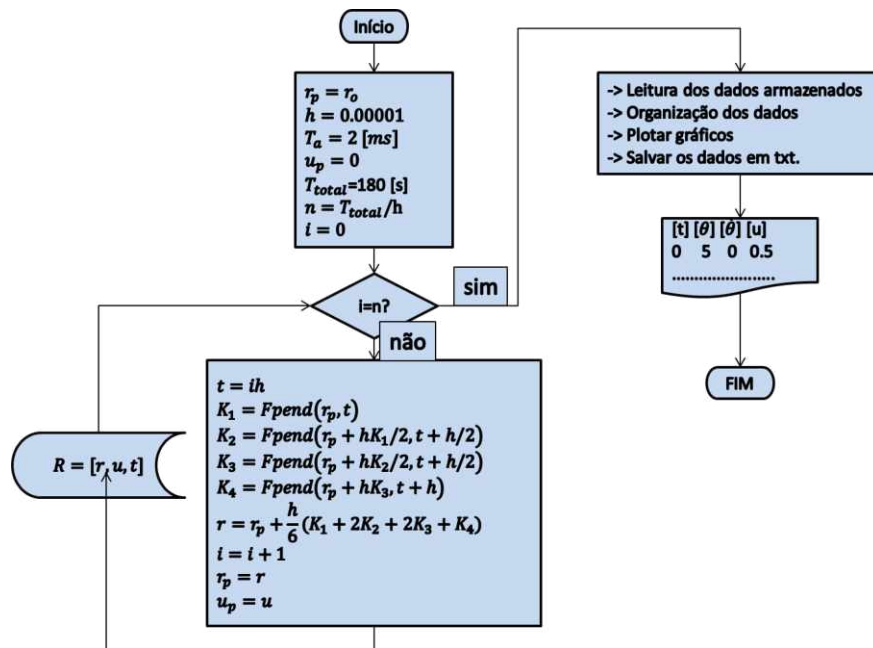


Figura 3.23 - Fluxograma do algoritmo utilizado em simulação. O fluxograma acima mostra apenas a função principal, a função que realiza a integração numérica, para o entendimento geral do algoritmo é necessária consultar as figuras (3.24a e 3.24b) onde se encontra o fluxograma das funções *Fpend* e *Fcontrol*.

A função principal que é mostrada na Figura (3.23) chama uma função auxiliar que é nomeada como *Fpend*, que de modo simplificado é mostrada na Figura (3.24). Como essa função tem as características dinâmicas do veículo de duas rodas, elas são distintas para o modelo linear e o modelo não linear, essas diferenças serão explicadas com inclusão das equações matemáticas que embasam o fluxograma de cada um dos modelos.

Para o controle do modelo que está sendo simulado, é necessária a criação de uma função controlador, nomeada como *Fcontrol*. Ela é solicitada pela *Fpend* para gerar a ação de controle que será aplicada nos motores do modelo simulado que gerarão as forças e influenciam no comportamento do modelo. O fluxograma dessa função é mostrado na Figura (3.24).

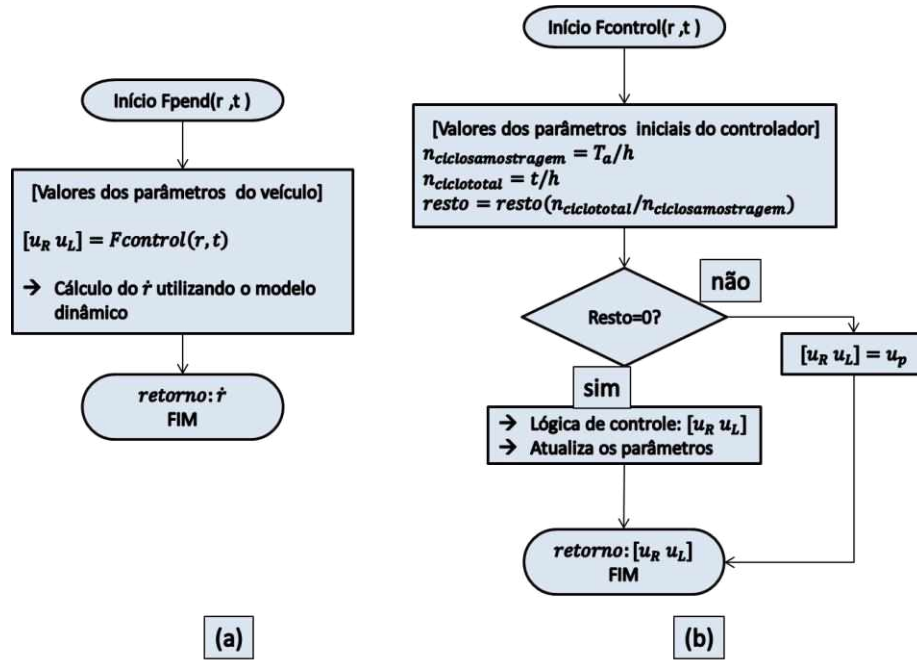


Figura 3.24 - Fluxograma das funções  $Fpend$  e  $Fcontrol$  utilizadas na simulação do veículo de duas rodas. Observa-se que esse fluxograma, assim como o da função principal, é uma lógica genérica que pode ser utilizada por qualquer lógica de controle e para o modelo linear e não linear.

Formada a estrutura da função principal agora é necessária a formulação matemática do vetor de estado  $r$  que é utilizado nos fluxogramas acima. E também a obtenção da derivada temporal desse vetor de estado dado o modelo dinâmico do veículo de duas rodas. Esse último, terá algumas modificações dependendo se a simulação será realizada para o modelo linear ou para o modelo não linear.

Primeiramente é necessária a definição do vetor de estado no instante  $t = K\Delta t$ , o mesmo é definido pela Equação (3.45).

$$\vec{r}_K = [\theta \quad \dot{\theta} \quad x \quad \dot{x} \quad \phi \quad \dot{\phi}]_{t=K\Delta t}^T \quad (3.45)$$

Posteriormente as velocidades angulares em cada uma das rodas podem ser obtidas pelas variáveis do vetor de estado através da equação abaixo que foram obtidas através das Equações (3.3) e (3.4).

$$\begin{cases} \dot{\phi}_R = (\dot{x} + d\dot{\phi})/r \\ \dot{\phi}_L = (\dot{x} - d\dot{\phi})/r \end{cases} \quad (3.46)$$

As correntes nos motores podem ser calculadas com base na Equação (2.1) como:

$$\begin{cases} i_R = (V_{cc}u_R - K_e\dot{\phi}_R) / R \\ i_L = (V_{cc}u_L - K_e\dot{\phi}_L) / R \end{cases} \quad (3.47)$$

Logo em seguida os torques em cada um dos motores podem ser calculados com base na Equação (2.3) como é descrito abaixo considerando a zona morta:

$$\begin{cases} T_R = K_m i_R \mu(|u_R| - dz) \\ T_L = K_m i_L \mu(|u_L| - dz) \end{cases} \quad (3.48)$$

Onde a função  $\mu$  é a função degrau unitário e esta função é definida como:

$$\mu(x) = \begin{cases} 1 \rightarrow x \geq 0 \\ 0 \rightarrow x < 0 \end{cases} \quad (3.49)$$

Após os cálculos dos torques é necessário então o cálculo da “força esperada” calculada através da Equação (3.13). Em seguida utilizando-se as equações (3.50), (3.51) e (3.52) para os cálculos das acelerações do sistema para o modelo não linear.

$$\begin{aligned} \ddot{\theta}_K = & \frac{1}{\left( J_p + ml^2 - \frac{(ml \cos(\theta_K))^2}{m + M + 2J_w / r^2} \right)} \left[ \tau_{ext} - \frac{ml \cos(\theta_K) f^*}{m + M + 2J_w / r^2} + mgl \sin(\theta_K) \right. \\ & \left. - \left( 2C_a + \frac{m^2 l^2 \sin(2\theta_K) \dot{\theta}_K}{2(m + M + 2J_w / r^2)} \right) \dot{\theta}_K + \left( \frac{2C_a}{r} + \frac{(2b/r^2) ml \cos(\theta_K)}{m + M + 2J_w / r^2} \right) \dot{x}_K \right] \end{aligned} \quad (3.50)$$

$$\ddot{x}_K = \frac{1}{m + M + 2J_w / r^2} \left( f^* + ml \sin(\theta_K) (\dot{\theta}_K)^2 - ml \cos(\theta_K) \ddot{\theta}_K - (2b/r^2) \dot{x}_K \right) \quad (3.51)$$

$$\ddot{\phi}_K = \frac{(T_R - T_L)}{r(J_Z + 2(d/r)^2 J_w)} \quad (3.52)$$

Para o modelo linear as equações encontradas são (3.53), (3.54) e (3.55), em vez das equações descritas no parágrafo anterior.

$$\ddot{\theta}_K = \frac{1}{\left( J_p + ml^2 - \frac{(ml)^2}{m + M + 2J_w / r^2} \right)} \cdot \left[ \tau_{ext} - \frac{mlf^*}{m + M + 2J_w / r^2} + mgl\theta_K - (2C_a)\dot{\theta}_K + \left( \frac{2C_a}{r} + \frac{(2b/r^2)ml}{m + M + 2J_w / r^2} \right) \dot{x}_K \right] \quad (3.53)$$

$$\ddot{x}_K = \frac{1}{m + M + 2J_w / r^2} (f^* - ml\ddot{\theta}_K - (2b/r^2)\dot{x}_K) \quad (3.54)$$

$$\ddot{\phi}_K = \frac{(T_R - T_L)}{r(J_z + 2(d/r)^2 J_w)} \quad (3.55)$$

As equações acima foram obtidas com seus respectivos modelos após consecutivas manipulações matemáticas, obtêm-se os resultados das acelerações mostradas acima.

Após encontrar as acelerações a derivada temporal do vetor de estado pode ser compreendida pela Equação (3.56)

$$\dot{\vec{r}}_K = [\dot{\theta} \quad \ddot{\theta} \quad \dot{x} \quad \ddot{x} \quad \dot{\phi} \quad \ddot{\phi}]_{t=k\Delta t}^T \quad (3.56)$$

Aplicando as Equações (3.45) até a (3.56) acima, juntamente com as lógicas de controle explicadas nos tópicos anteriores, pode-se completar o algoritmo e simular o veículo de duas rodas com o modelo linear e o modelo não linear para as lógicas de controle PID e Fuzzy descritas anteriormente.

A função principal em software Matlab encontra-se em apêndice VI, e também as funções Fpend () não linear em apêndice VII, e a linear em apêndice VIII, a Fcontrol () para o controlador PID em apêndice IX e para o controlador Fuzzy encontram-se em apêndice X.

### 3.7. Aplicação dos controladores no veículo real

Para a aplicação dos controladores no veículo de duas rodas foi desenvolvido o software para Arduino, para a aplicação no microcontrolador Arduino DUE que foi utilizando no projeto do veículo.

O ângulo com a vertical do veículo é obtido através do ângulo entre a aceleração total (sem forças externas no veículo, a única aceleração a atuar no veículo é a gravidade) no veículo e os planos x, y e z de referência do acelerômetro da MPU6050. A velocidade angular é obtida através do giroscópio da MPU6050.

Foram utilizadas como base para esse software as bibliotecas prontas fornecidas pelo próprio ambiente de desenvolvimento da plataforma Arduino (IDE) como a I2C e a Kalman.

A biblioteca I2C foi utilizada para adquirir os dados da MPU6050 que usa esse protocolo de comunicação para transmitir dados para o Arduino DUE. Há bibliotecas prontas para a I2C para vários microcontroladores da família Arduino, porém para o DUE não se conseguiu encontrar uma biblioteca que funcionasse satisfatoriamente, necessitando de algumas adaptações.

Como o microcontrolador tem uma velocidade de processamento relativamente alta (quando comparada a outros microcontroladores da família Arduino), o mesmo consegue rodar a função principal void loop () em torno de 5 vezes mais rápido do que a o tempo de amostragem que foi utilizado (2ms). Sendo assim, a função principal não utiliza de interrupções para a leitura de dados, fazendo a leitura uma vez por ciclo da função principal e utiliza de um filtro de Kalman para melhorar a leitura dos ângulos a cada instante reduzindo o erro na leitura dos dados.

A versão final do software encontra-se em apêndice V para o controlador PID, porém, para alterar o controlador basta alterar a lógica que gera a ação de controle na função void control ().

### 3.8. Adaptação do controlador PID ao modelo não linear - PID ADP.

O fluxograma mostrado na Figura 3.25 indica a sequência para a melhoria dos parâmetros do controlador PID na tentativa de adaptar o controlador ao modelo não linear. Os parâmetros iniciais do controlador são salvos no vetor P e calcula-se o erro médio quadrático inicial ( $e_{MSE0}$ ).

Cada parâmetro do controlador é analisado individualmente em ordem no vetor. Criam-se dois valores candidatos para cada parâmetro, um superior ( $P_{m1}$ ) e um inferior ( $P_{m2}$ ) de acordo com uma taxa. Em seguida calcula-se através da simulação por 50 s o erro médio quadrático para o modelo não linear para cada um dos parâmetros candidatos.

Se o novo parâmetro produzir um erro médio quadrático menor para a resposta esse parâmetro substitui o parâmetro anterior e é salvo no vetor P. É repetido o processo para todos os parâmetros e no final os parâmetros do controlador geram um erro médio quadrático menor para a simulação utilizando o modelo não linear.

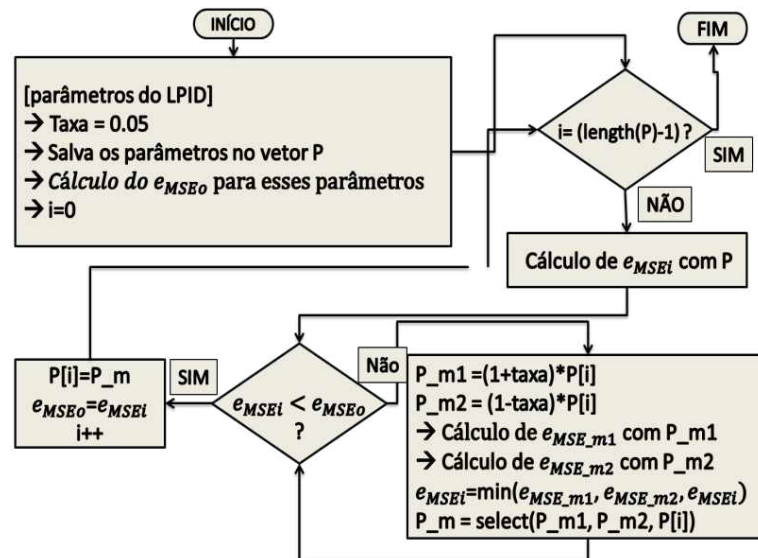


Figura 3.25. Fluxograma utilizado para adaptar os parâmetros do PID ao modelo não linear.

Após a aplicação do algoritmo acima com cada parâmetro do PID obtém-se os parâmetros do PID ADP abaixo:

$$\begin{cases} K_p = 1.68 \\ K_I = 21.98 \\ K_d = 0.018 \end{cases} \quad (3.57)$$

## **CAPÍTULO IV**

### **RESULTADOS E DISCUSSÕES**

Neste capítulo serão apresentados os resultados obtidos com a metodologia adotada no Capítulo 3. Este foi dividido em seis partes, a primeira se refere aos resultados da construção do veículo, a segunda, terceira, quarta e quinta parte se referem aos resultados dos controladores que foram apresentados na metodologia. A última parte consiste em uma discussão aprofundada sobre alguns aspectos das respostas dos controladores.

O resultado de cada controlador proposto na metodologia encontra-se nos tópicos abaixo. É realizado o experimento no veículo de duas rodas e a simulação para o mesmo controlador com as mesmas condições iniciais e ambos os resultados reais e da simulação se encontram nos tópicos de cada controlador. Alguns outros tópicos das discussões das respostas são encontrados posteriormente no tópico 4.6.



#### 4.1. Resultados da construção do veículo

A imagem do veículo de duas rodas em sua versão final é mostrada na Figura 4.1. Nessa etapa, o veículo já está pronto para os testes com os controladores. Observa-se também a disposição dos componentes eletrônicos onde a maior parte foi fixada na placa inferior do veículo.

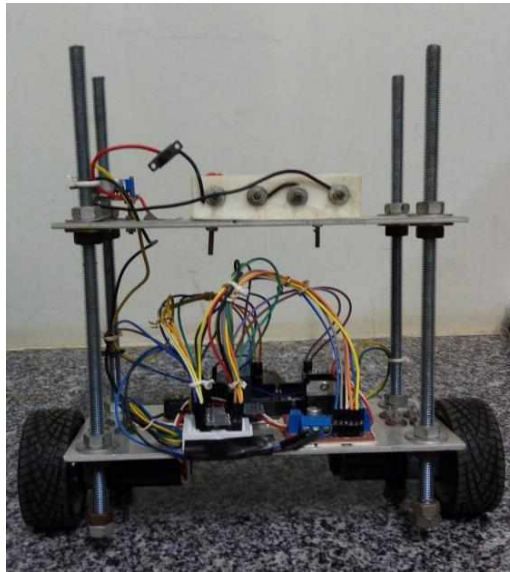


Figura 4.1. Imagem do veículo de duas rodas real.

Após os primeiros testes com o veículo, observou-se que as três placas de alumínio do projeto não seriam necessárias, pois, com apenas duas placas o veículo se encontrava satisfatoriamente rígido. Sendo assim, a terceira placa apenas iria aumentar a massa do veículo, optando-se assim pela sua retirada, restando apenas as duas que continham os componentes eletrônicos.

Nos primeiros testes do veículo, observou-se também um problema de Hardware onde um dos componentes, a MPU6050 interrompia sua comunicação I2C com o Arduino DUE inesperadamente.

Após os primeiros dias de teste observou-se que o erro acontecia sempre quando havia a mudança de sentido da rotação dos motores, levando a possibilidade de uma interferência eletromagnética proveniente dos motores e da estrutura condutora. Com alguma investigação sobre a hipótese, foi confirmada a interferência eletromagnética dos motores, e o problema foi resolvido aterrando a estrutura do veículo ao 0V da bateria.

## 4.2. Resultados para o controlador PID Z-N.

Os resultados experimentais obtidos para o controlador PID Z-N são mostrados nos gráficos na Figura 4.2 e 4.3. O ângulo inicial do veículo é a aproximadamente  $10^\circ$  que foi medido com o programa que é utilizado no controle do veículo.

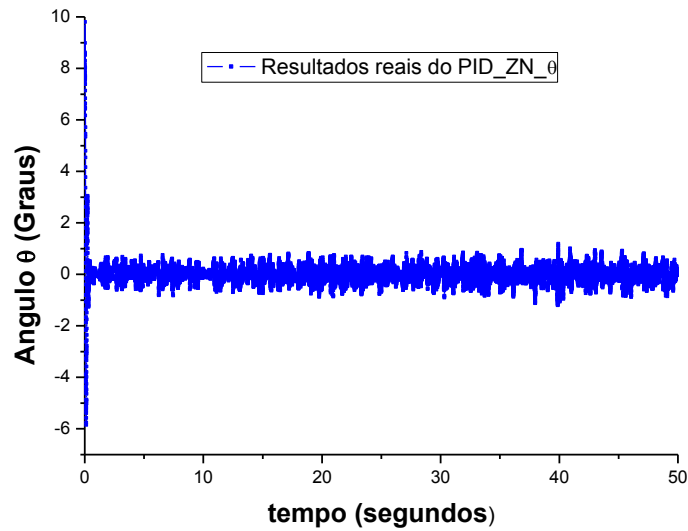


Figura 4.2 - Resultados reais no tempo da posição angular do pêndulo com a vertical para o controlador PID com parâmetros experimentais obtidos pelo método de Z-N de malha fechada.

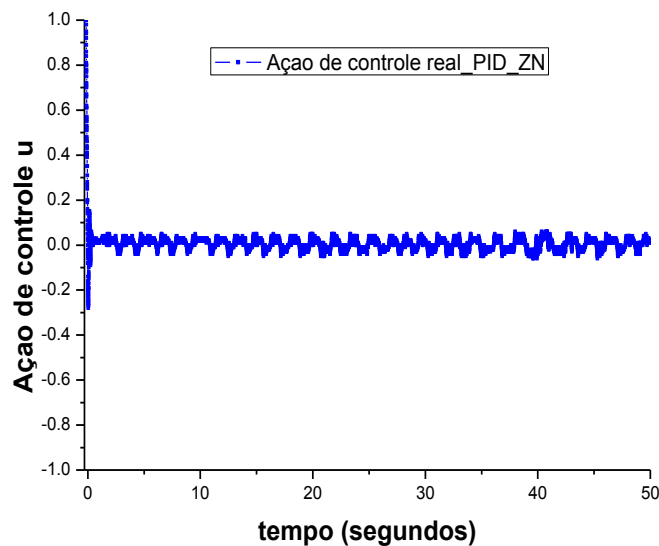


Figura 4.3. Ação de controle no tempo para o PID com parâmetros experimentais.

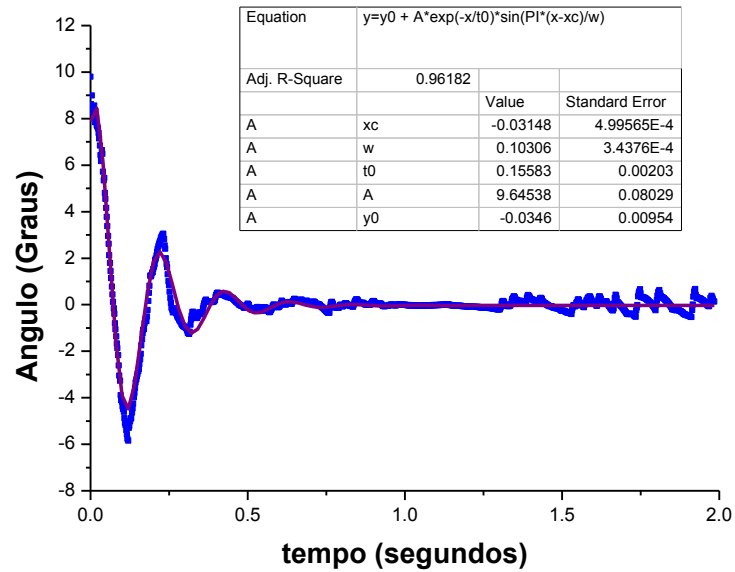


Figura 4.4. Detalhes do regime transiente do resultado real do veículo e ajuste de curva para a obtenção do tempo de acomodação e outras propriedades importantes da resposta do sistema.

Os resultados simulados para as mesmas condições iniciais e mesmo controladores encontram-se abaixo nas Figuras 4.5 e 4.6.

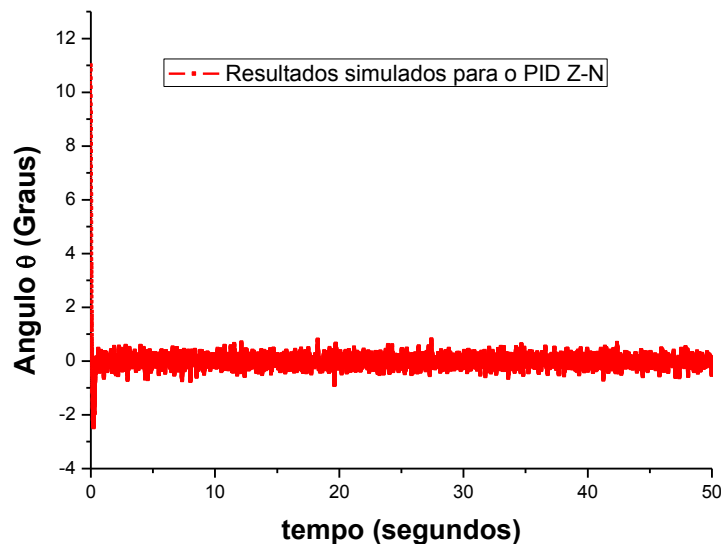


Figura 4.5 - Resultados simulados para o PID Z-N com as mesmas condições iniciais que foram realizadas os experimentos no veículo real.

Da mesma forma que foi realizada para os resultados reais, foi realizado um ajuste de curva para o período transiente da resposta da simulação no tempo. O resultado encontra-se no gráfico da Figura 4.6 e na Tabela 4.1 abaixo.

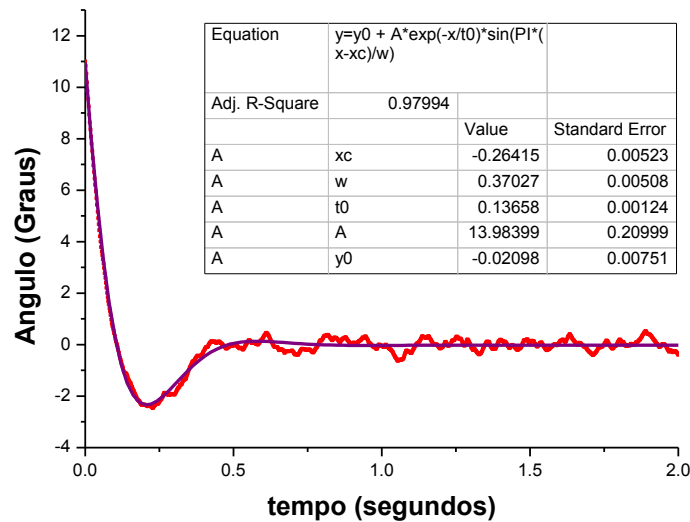


Figura 4.6 - Detalhes do regime transiente do resultado simulado do veículo e ajuste de curva para a obtenção do tempo de acomodação e outras propriedades importantes da resposta do sistema.

Foi realizado um ajuste de curva do tipo senoidal decrescente (equação encontra-se na Tabela 4.1), que é muito utilizada em ajustes de resposta de sistemas de segunda ordem por ser a solução temporal da equação diferencial de um sistema linear de segunda ordem com amortecimento entre 0 e 1 (apêndice XX).

Tabela 4.1. Tabela com as principais informações do resultado do controlador para o experimento no veículo em tempo real e para a simulação do veículo em mesmas condições iniciais.

| Controlador PID Z-N             | Simulação  | Resultado em tempo real                                  |
|---------------------------------|--|--|
| Equação utilizada para o Ajuste | $\theta = \theta_o + Ae^{-t/t_o} \sin(\pi(t - t_c) / w)$ | $\theta = \theta_o + Ae^{-t/t_o} \sin(\pi(t - t_c) / w)$ |
| Tempo de acomodação critério 5% | 0.55 s   | 0.62 s   |
| Overshoot máximo.               | 31%  | 54%  |
| Erro médio                      | $5.4 \times 10^{-4}$                                     | $5.5 \times 10^{-4}$                                     |
| Desvio padrão do erro           | 0.356  | 0.773  |
| Erro médio quadrático em regime | 0.206  | 0.284  |
| Número de dados                 | 25000  | 40595  |

Observam-se algumas diferenças entre os resultados simulados e o resultado em tempo real para esse controlador. Dentre as principais o overshoot e o desvio padrão do erro, porém os outros critérios analisados foram semelhantes.

Aplicando um teste de hipótese para a diferença das médias e para a diferença das variâncias observa-se que não se pode afirmar que as respostas são diferentes para um nível de confiança estatística de 95%. De fato analisando as respostas observa-se uma semelhança grande em regime permanente, porém há pequenas diferenças em regime transiente.

### 4.3. Resultados para o controlador PID PROJ.

Os resultados reais obtidos pelo controlador PID PROJ encontram-se nos gráficos das Figuras 4.7 e 4.8 abaixo. A condição inicial utilizado no experimento foi um ângulo inicial de  $-5.2^\circ$  com a vertical. Observam-se também na Figura 4.9 detalhes do regime transiente da resposta do veículo ao controlador.

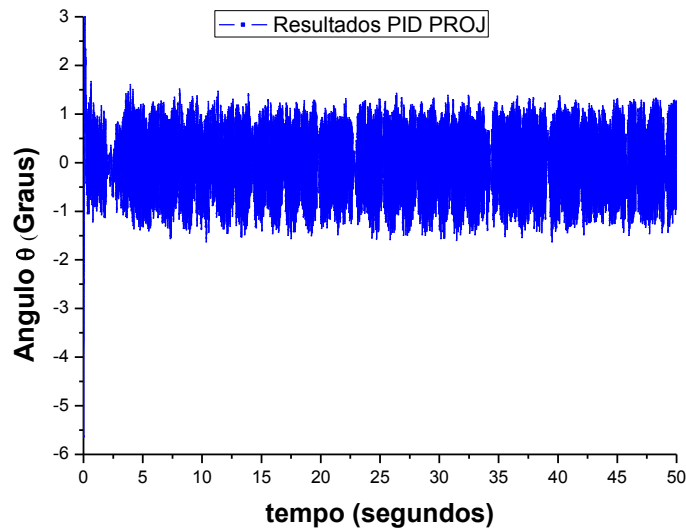


Figura 4.7. Gráfico mostrando o resultado da posição angular para o controlador obtido pelo método de alocação dos pólos.

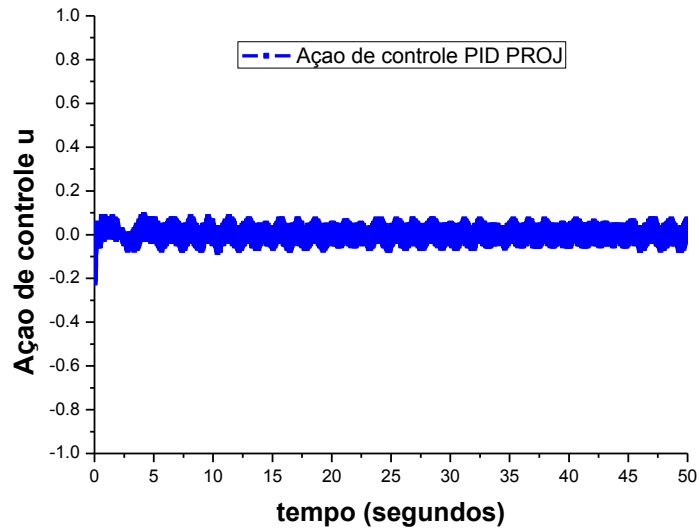


Figura 4.8. Ação de controle para o controlador PID PROJ no tempo

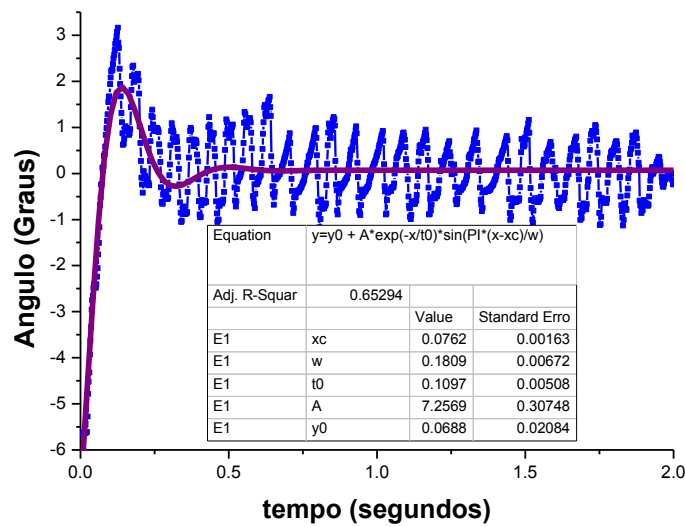


Figura 4.9 - Detalhes do regime transiente para o controlador PID PROJ juntamente com uma curva de ajuste para a melhor obtenção dos resultados.

Os resultados encontrados através da simulação para o mesmo controlador e as mesmas condições iniciais encontra-se nos gráficos das Figuras 4.10 e 4.11. Na Figura 4.11 encontram-se detalhes do regime transiente para a resposta simulada do veículo de duas rodas.

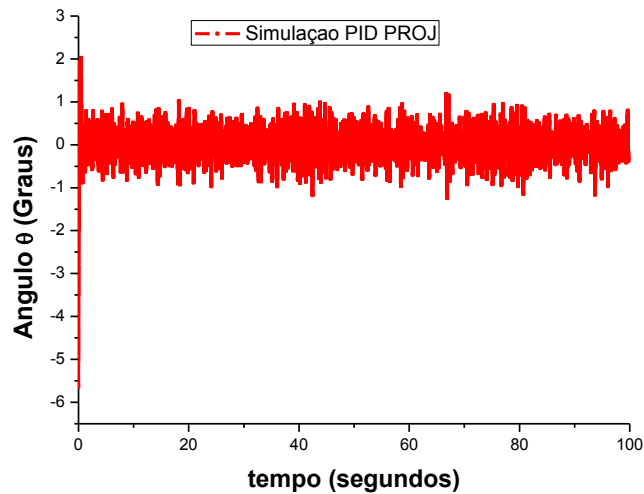


Figura 4.10 - Resultados Simulação para o controlador PID PROJ.

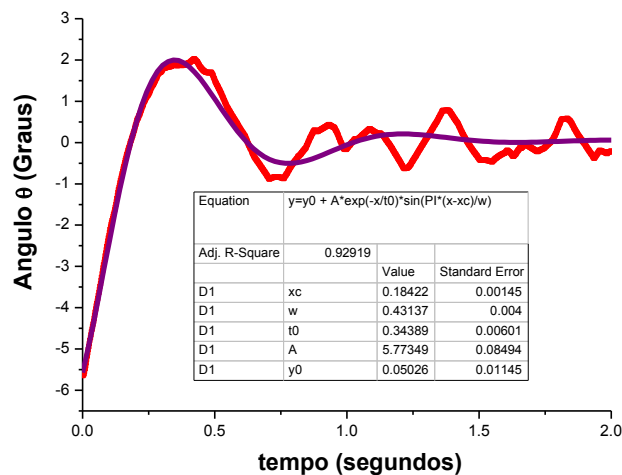


Figura 4.11 - Detalhes do regime transiente para o controlador PID PROJ na simulação juntamente com uma curva de ajuste.

Tabela 4.2 - Tabela com as principais informações do resultado do controlador para o experimento no veículo em tempo real e para a simulação do veículo em mesmas condições iniciais.

| Controlador PID PROJ            | Simulação  | Resultado em tempo real                              |
|---------------------------------|--|--|
| Equação utilizada para o Ajuste | $\theta = \theta_o + Ae^{-t/t_o} \sin(\pi(t-t_c)/w)$ | $\theta = \theta_o + Ae^{-t/t_o} \sin(\pi(t-t_c)/w)$ |
| Tempo de acomodação critério 5% | 1.32 s   | 0.44 s   |
| Overshoot máximo.               | 33%  | 50%  |
| Erro médio                      | 0.0004   | 0.0007   |
| Desvio padrão do erro           | 0.381  | 0.721  |
| Erro médio quadrático em regime | 0.342  | 0.719  |
| Número de dados                 | 50001  | 152273   |

Para esse controlador os resultados encontram-se na Tabela 4.2 e quase todos os critérios analisados foram muito diferentes. Observa-se que a simulação cumpre com os requisitos de projeto enquanto o experimento no veículo não. Observa-se também além de diferenças no regime transiente as oscilações no regime permanente também são diferentes.

Aplicando o teste de hipótese para as respostas verifica-se com 95% de nível de confiança estatística que os resultados são diferentes entre o veículo e a simulação para esse mesmo controlador.

#### 4.4. Resultados para o controlador fuzzy.

Os resultados para o controlador Fuzzy encontram-se abaixo nas Figuras 4.12 e 4.13. As condições iniciais são do veículo parado com uma inclinação inicial de  $-3^\circ$ . Detalhes do regime transiente são mostrados na Figura 4.14 juntamente com uma curva de ajuste.

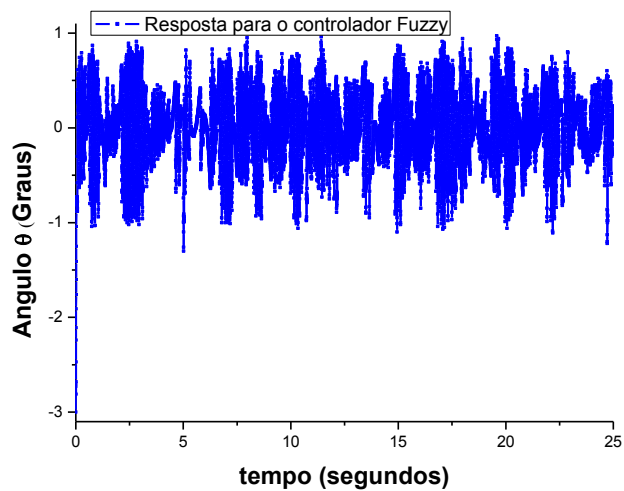


Figura 4.12 - Resultados reais da resposta do veículo para o controlador Fuzzy.



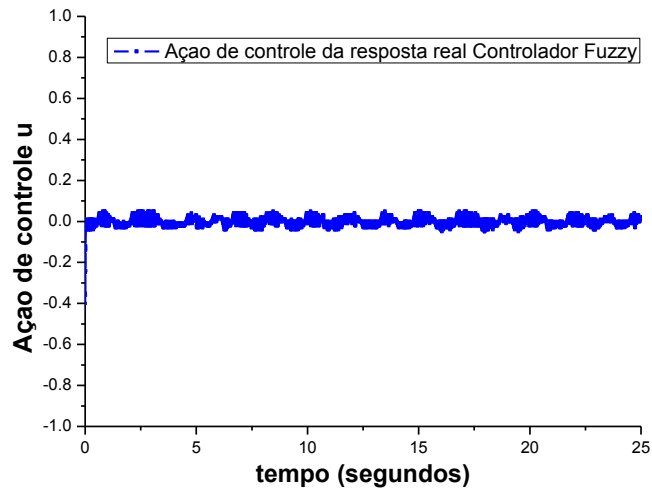


Figura 4.13 - Ação de controle no tempo para a resposta real do veículo para o controlador Fuzzy.

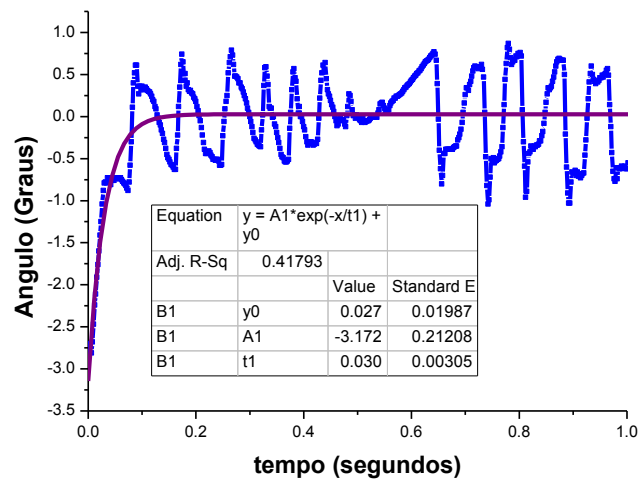


Figura 4.14 - Detalhes do regime transiente da resposta do veículo real para o controlador Fuzzy.

A resposta da simulação do veículo de duas rodas para o controlador Fuzzy com mesmas condições iniciais encontra-se na Figura 4.15. Da mesma forma que a resposta em tempo real do veículo, observam-se os detalhes da resposta em regime transiente na Figura 4.16.

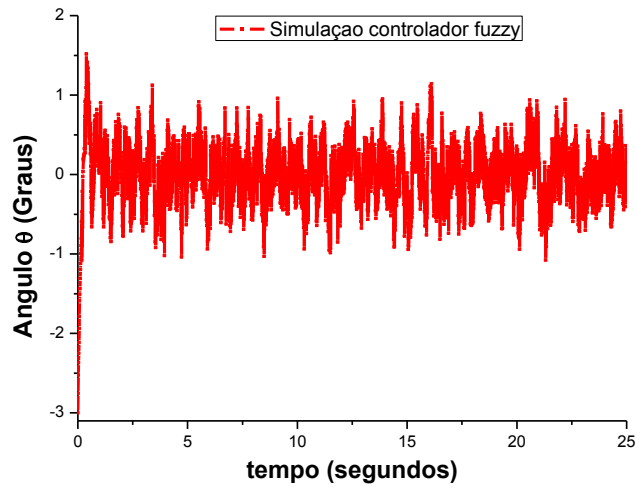


Figura 4.15 - Resultados simulados para o controlador Fuzzy

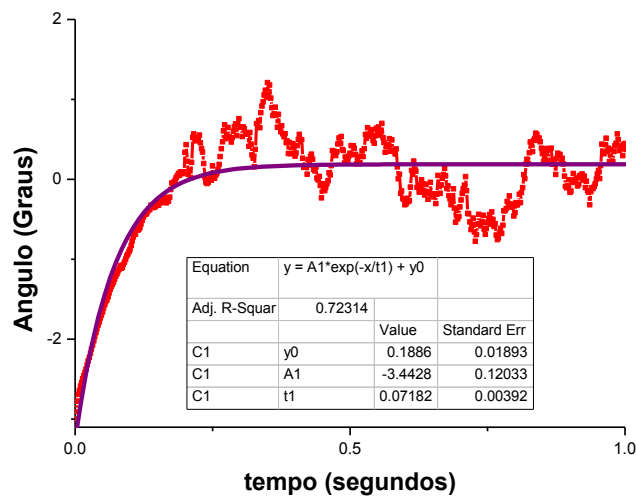


Figura 4.16 - Detalhes do regime transiente da simulação da resposta para o controlador Fuzzy.

Tabela 4.3. Tabela com as principais informações do resultado do controlador Fuzzy para o experimento no veículo em tempo real e para a simulação do veículo em mesmas condições iniciais.

| Controlador Fuzzy               | Simulação                         | Resultado em tempo real           |
|---------------------------------|-----------------------------------|-----------------------------------|
| Equação utilizada para o Ajuste | $\theta = \theta_o + Ae^{-t/t_o}$ | $\theta = \theta_o + Ae^{-t/t_o}$ |
| Tempo de acomodação critério 5% | 0.28 s                            | 0.12 s                            |
| Overshoot máximo.               | 18%                               | 16%                               |
| Erro médio                      | -0.0005                           | -0.003                            |
| Desvio padrão do erro           | 0.406                             | 0.367                             |
| Erro médio quadrático em regime | 0.377                             | 0.359                             |
| Número de dados                 | 12501                             | 13545                             |

A comparação dos resultados ocorre na Tabela 4.3. Como nos outros controladores apresentam uma pequena diferença de tempo de acomodação, mas overshoot próximos e comportamento em regime permanente muito parecido. Observam-se tempos de acomodação muito pequenos quando comparados com os outros controladores tanto na simulação quanto na resposta em tempo real do veículo.

Utilizando um teste de hipótese para a diferença entre as médias e a diferença entre as variâncias verifica-se com um nível de confiança de 95% que as respostas não são diferentes.

#### 4.5. Resultados do Controlador adaptado através da simulação

A resposta do veículo de duas rodas ao controlador PID adaptado ao modelo não linear através do PID projetado pelo método do lugar das raízes (PID PROJ) tem sua resposta nos gráficos da Figura 4.17.

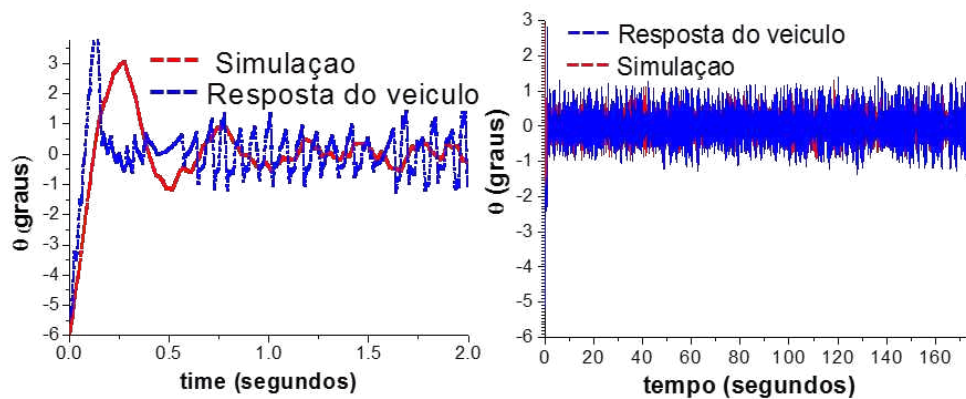


Figura 4.17 - Resultados simulados e reais do veículo de duas rodas para o PID com parâmetros adaptados ao modelo não linear. Na esquerda observam-se os detalhes em regime transiente para a resposta e a simulação, e na figura à direita mostra a resposta durante 170 segundos de experimento.

A comparação dos resultados para esse controlador encontra-se na Tabela 4.4.

Esse controlador foi o que apresentou resultados mais próximos entre a simulação e a resposta em tempo real do veículo. Observam-se poucas diferenças entre as respostas. Utilizando o mesmo teste de hipótese aplicado aos outros controladores verifica-se com 95% de confiança que as respostas não são diferentes.

Tabela 4.4 - Tabela com as principais informações do resultado do controlador Fuzzy para o experimento no veículo em tempo real e para a simulação do veículo em mesmas condições iniciais.

| Controlador PID                 | Simulação  | Resultado em tempo real                              |
|---------------------------------|--|--|
| Equação utilizada para o Ajuste | $\theta = \theta_o + Ae^{-t/t_o} \sin(\pi(t-t_c)/w)$ | $\theta = \theta_o + Ae^{-t/t_o} \sin(\pi(t-t_c)/w)$ |
| Tempo de acomodação critério 5% | 0.82 s   | 0.74 s   |
| Overshoot máximo.               | 50%  | 57%  |
| Erro médio                      | 0.0006   | 0.0004   |
| Desvio padrão do erro           | 0.406  | 0.467  |
| Erro médio quadrático em regime | 0.275  | 0.269  |
| Número de dados                 | 65001  | 65354  |

#### 4.6. Comparação e análises detalhadas das respostas dos controladores

Além dos testes de hipótese aplicados para a resposta da simulação e o resultado em tempo real para cada controlador outras análises também podem ser exploradas. Observa-se que para todos os controladores o resultado em tempo real apresenta uma resposta mais rápida (menor tempo de acomodação) do que a simulação. Parte desse efeito pode ser associada ao modo de leitura do ângulo com a vertical como o ângulo entre a componente z da aceleração e a componente total lida pelo acelerômetro como mostra a Figura 4.18.

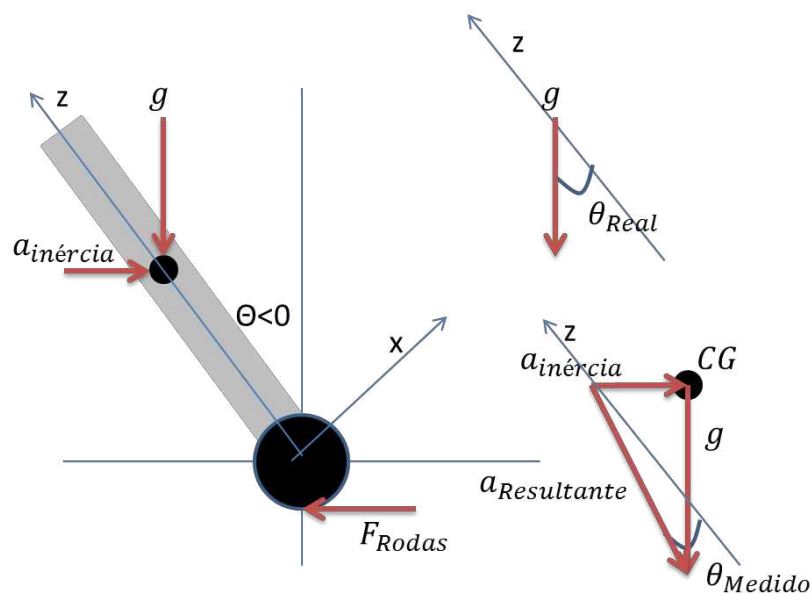


Figura 4.18 - Imagem representando uma leitura errada do ângulo  $\theta$  associada a acelerações do centro de massa que não deveriam ser contabilizadas. Os eixos  $z$  e  $x$  mostrados são os eixos do acelerômetro MPU6050 não os eixos considerados na secção de modelagem dinâmica.

No instante inicial dependendo do ângulo de inclinação o torque que o motor aplica e consequentemente em cadeia aumentando as acelerações no centro de massa podem apresentar uma influência significativa na leitura do ângulo com a vertical. Esse efeito é reduzido com a utilização de um filtro de Kalman como é detalhado anteriormente, mas o efeito não é totalmente removido.

Mas, a diferença entre o tempo de acomodação da resposta em tempo real e da simulação pode também ser proveniente dos parâmetros do motor, pois não foi utilizado nenhum experimento para confirmá-los.

O algoritmo *neuro-fuzzy* (Pereira, B. L.) tem como função custo o erro médio quadrático entre as respostas, para minimizá-lo o algoritmo pode ter selecionado os parâmetros que geram uma oscilação em torno de zero mais próxima entre as respostas, porém o algoritmo pode ter considerado ruídos, o erro de leitura indicado anteriormente, entre outros fatores que poderiam gerar parâmetros elétricos do motor diferentes dos reais, gerando assim uma resposta principalmente em regime transiente diferente.

Um segundo fator que indica diferenças entre os parâmetros adotados e os reais parâmetros do veículo é que o controlador projetado através desses parâmetros obteve a maior diferença entre a resposta em tempo real e a simulação.

Na Tabela 4.5 se encontra uma comparação para o desempenho dos controladores em tempo real.

Tabela 4.5. Comparação do desempenho dos controladores aplicados no veículo. A marcação em azul indica o melhor controlador em cada critério avaliado

| Controlador                     | PID Z-N | LPID   | PID Adp. | Fuzzy  |
|---------------------------------|---------|--------|----------|--------|
| Tempo de acomodação critério 5% | 0.82 s  | 0.44 s | 0.74 s   | 0.12   |
| Overshoot máximo.               | 50%     | 50%    | 57%      | 16%    |
| Erro médio                      | 0.0006  | 0.0007 | 0.0004   | -0.003 |
| Erro médio quadrático em regime | 0.275   | 0.719  | 0.269    | 0.359  |

Observa-se também que o controlador com parâmetros adaptados para o modelo não linear via simulação obteve bons resultados sendo o controlador que tem menor amplitude de oscilação (avaliado pelo Erro médio quadrático) dos controladores testados no veículo.

Outro ponto notável é o aumento do *overshoot* da resposta ao controlador em tempo real em relação à simulação, onde apenas foi exceção para o controlador fuzzy. Também chama a atenção o controlador fuzzy no critério de tempo de acomodação sendo o menor, e também teve o menor *overshoot*.

## CAPÍTULO V

### CONCLUSÃO

Observa-se que foi possível utilizar as técnicas de controle apresentadas para a estabilidade vertical estática do veículo de duas rodas do tipo pêndulo invertido, e também utilizar a simulação do modelo linear para comparar a resposta da simulação e da resposta real do veículo.

Através dos métodos apresentados neste trabalho, os controladores foram capazes de resolver o problema do pêndulo invertido equilibrando o veículo com uma oscilação máxima de  $1^\circ$  em regime permanente. Dessa forma todos os métodos de controle utilizados se mostraram eficientes no controle de um pêndulo invertido.

A simulação proposta para o modelo não linear também funcionou de modo satisfatório, pois, em geral as respostas em tempo real e da simulação foram próximas e foi verificado estatisticamente por um teste de hipótese para a diferença das médias e para a diferença das variâncias com um nível de confiança de 95%.

Apesar do teste de hipótese verificar que as respostas não são diferentes entre a simulação e a resposta em tempo real, as mesmas apresentam uma pequena diferença em regime transiente, parte das explicações podem ser justificadas pela leitura do sensor, porém, deve-se investigar melhor os parâmetros do motor que foram obtidos através do algoritmo de otimização utilizado, pois como já foi discutido anteriormente, podem ser atribuídos outros fatores aos valores dos parâmetros como ruídos e erros de leitura.

O controlador puramente Fuzzy apresenta um bom desempenho, porém tem mais oscilações em regime permanente quando comparado ao melhor resultado de um PID, uma

simulação de um controlador PID-Fuzzy tem resultados significativamente melhores que os apresentados e pode ser uma alternativa para um controlador com menor tempo de acomodação e menor oscilação.

Também se pode concluir desse trabalho que é possível melhorar os parâmetros de um controlador PID, ou mesmo de outros controladores utilizando a simulação com o modelo não linear. O PID com parâmetros melhorados obteve melhores resultados em praticamente todos os quesitos analisados quando comparados ao PID projetado pelo método do lugar das raízes.

Como observação final, a realização deste trabalho salientou a importância de uma abordagem multidisciplinar no aprendizado, pois um projeto como esse exigiu conhecimentos de diversas áreas apresentadas no curso de Engenharia Mecatrônica, como instrumentação, eletrônica, desenvolvimento de software, controle de sistemas lineares, controle digital, sistemas digitais, processamento digital de sinais, dinâmica e cálculo numérico.

## TRABALHOS FUTUROS

A identificação dos parâmetros dos motores do veículo de duas rodas pode ser realizada através da identificação da função transferência de cada motor. A obtenção desses parâmetros é muito importante para a continuidade de trabalhos utilizando o veículo de duas rodas.

Posteriormente utilizando técnicas de controle MIMO (múltiplas entradas e múltiplas saídas) pode-se realizar o controle simultâneo do equilíbrio da posição vertical, a velocidade ou distância percorrida pelo veículo e também a orientação da base do veículo utilizando esse trabalho. Uma das propostas para esse veículo é um controle em cascata utilizando dois controladores PID, um para o ângulo com a vertical e outro para a posição da base do veículo.

Considerando os três graus de liberdade apresentado no trabalho, posteriormente pode ser realizado o controle e simulação de trajetórias no plano utilizando esse veículo e o modelo não linear apresentado.

Através desse trabalho, também pode ser realizado o projeto e a construção de veículos de duas rodas do tipo pêndulo invertido similares ou até veículos com capacidade de transportar uma pessoa como o *Segway*.

## Referências Bibliográficas

Arduino DUE. Disponível em <<https://www.filipeflop.com/blog/arduino-due-atmel-arm-32bits/>>. Acessado em 15/08/2017.

Bonafilia, B., 2014. *Self-balancing two-wheeled robot*. p.1-11.

Balduino. Disponível em: <<http://www.balduino.net/>>. Acessado em: 20/09/2017.

Chen, C.H., Lin, C.J. and Lin, C.T. , 2008. *Nonlinear System Control Using Adaptive Neural Fuzzy Networks Based on a Modified Differential Evolution*, *IEEE Transactions on Systems, Man, and Cybernetics — Part C: Applications and Reviews*, Taiwan,v.39,n.4,p.459-473.

El-Nagar, A. M., El-Bardini, M. and EL-Rabaie, N. M. , 2014. *Intelligent control for nonlinear inverted pendulum based on interval type-2 fuzzy PD controller*, *Alexandria Engineering Journal* 53, 23–32.

EN-V. Disponível em <<http://transitolivre.org.br/carro-de-ruas-rodas-da-gm-ja-e-realidade/>>. Acessado em 20/09/2017.

Feifei, R., Lei, Z., Le, T., Yanhai, H., and Pengpeng, Z., 2013. *Design and Research of Double Closed-Loop Control Strategy for Inverted Pendulum System*, 3th International Conference on Intelligent System Design and Engineering Applications.

Juang, H.S., Lum, K.Y., 2013. *Design and Control of a Two-Wheel Self-Balancing Robot using the Arduino microcontroller Board*, 10th IEEE International Conference on Control and Automation (ICCA), China, p. 634639.

Motor DC. Disponível em <<https://www.filipeflop.com /produto/motor-dc-6v-210rpm-com-encoder-e-adaptador/>>. Acessado em 15/08/2017.

MPU6050. Disponível em <<https://www.filipeflop.com/produto/acelerometro-e-giroscopio-3-eixos-6-dof-mpu-6050/>>. Acessado em 15/08/2017.

OGATA, K. Engenharia de Controle Moderno. Tradução: Paulo Álvaro Maya. 4. ed., 3ª reimpressão 2007. São Paulo: Pearson Prentice Hall, 2003. 808p.

Pereira, B.L. Aplicação de sistemas neuro-fuzzy e evolução diferencial na modelagem e controle de veículo de duas rodas. Dissertação de Mestrado – Universidade Federal de Uberlândia, 2017.

Ponte H. Disponível em <<https://www.filipeflop.com/produto/driver-motor-ponte-h-1298n/>>. Acessado em 15/08/2017.

Regulador de tensão. Disponível em: <[https://produto.mercadolivre.com.br/MLB-874467514-xl6009-conversor-dc-dc-step-up-step-down-regulador-tenso-\\_JM](https://produto.mercadolivre.com.br/MLB-874467514-xl6009-conversor-dc-dc-step-up-step-down-regulador-tenso-_JM)>. Acessado em 16/08/2017.



Saifizul, A.A., 2006. *Intelligent Control for Self-erecting Inverted Pendulum Via Adaptive Neuro-fuzzy Inference System*, American Journal of Applied Sciences, Malaysia, v.3, n.4, p.1795-1802, 2006.

Segway. Disponível em: < <http://urbanwheels.co.nz/> >. Acessado em 29/09/2017.

Silva, G. V. M., 2006. *Controlo não linear*, disponível em: < <http://www.etla.com.pt/images/cnl.pdf> >. Acessado em 10/07/2017.

Zadeh, L.A, 1965. *Fuzzy Sets Information and Control*, Berkeley, v. 8, p. 338-353.

## APÊNDICE I – Cálculo dos parâmetros

```

Clc; clear all; close all
% variação da distância entre placas
h=0:0.1:20;

% medidas
M_2motor = 0.242;
M1 = 0.155;
M2 = 0.245;
dM_fios = (25)*0.003;
M = 0.062;
M_4barras = 0.252;
M_placa = 0.160;
m = M_2motor+ M1 + M2 + 2*M_placa + M_4barras+ dM_fios;
h1 = 2.1;
d_h1 = 1.0;
d_h2 = 1.5;
d_hb = (3.8-1.5);
hp = 23;

e = 0.3;
R_motor = 1.15;
r = 6.3/2;
r_interno = 5.3/2;
d_eixos = 23;

%calculos
Io_2motor = (M_2motor)*(R_motor^2)/2;
Io_placa = M_placa*(e^2)/12;
Io_M1 = M1*(d_h1^2)/12;
Io_M2 = M2*(d_h2^2)/12;
Io_4barras = M_4barras*(hp) .* (hp)/12;
Io_rodas = M*((r+r_interno)/2)^2;
I_2motor = Io_2motor;
I_placa1 = Io_placa + M_placa*(h1^2);
I_M1 = Io_M1 + M1*((h1+d_h1)^2);
I_4barras = Io_4barras + M_4barras*((hp/2)-d_hb) .* ((hp/2)-d_hb);
I_placa2 = Io_placa + M_placa*((h1+h) .* (h1+h));
I_M2 = Io_M2 + M2*((h1+h+d_h2) .* (h1+h+d_h2));
I_eixo = I_2motor+I_placa1+I_M1+I_4barras+I_placa2+I_M2;
m_x_l_cm = M_placa*h1 + M1*(h1+d_h1) + M_4barras*(hp/2 -d_hb) +
M_placa*(h1+h) + M2*(h+h1+d_h2) + dM_fios*(h1+h/2);
l_cm = (m_x_l_cm)/m;

```

## APÊNDICE II – Parâmetros Experimentais do PID

```

clear all clc close all
edit 'teste_ZN_T1_GB.txt';
ensaio = load('teste_ZN_T1_GB.txt');
theta = ensaio(:,3);
time = ensaio(:,1);
time = time - time(1,1);
n = length(time);
t_max = time(length(time));
dt = (t_max - time(1))/n;

tempo = [0:dt:(n-1)*dt]';

fs = 1/dt;
df = fs/n;
f = 0:df:(n/2-1)*df;

figure
%plot(t,xt);
XT= fft(theta,n)*2/n;
plot(f,abs(XT(1:n/2)));

K_crit = 2.8;
P_crit = 0.0771;
Kp = 0.6*K_crit;
Ki = Kp/(0.5*0.0771);
Kd = Kp*0.125*0.0771;

fid = fopen('Z-N_resultados_ensaio_tempo.txt','wt');
A0011 = [tempo';theta'];
fprintf(fid,'%f %f\n',A0011);
fclose(fid);

fid = fopen('Z-N_resultados_ensaio_frequencia.txt','wt');
A0011 = [f;abs(XT(1:n/2))'];
fprintf(fid,'%f %f\n',A0011);
fclose(fid);

```

## APÊNDICE III – G(s) e projeto do controlador PID

```

clc; clear all; close all;
s = tf([1 0],1);
l = 0.0748;           % distância entre o eixo e o centro de massa
g = 9.79;           % valor aproximado da gravidade em Uberlândia
M = 0.062;          % massa não suspensa, 2 rodas;
m = 1.304;          % massa pendulo, estrutura e componentes;
I = 48*10^-4;       % calculado aproximadamente
b = 0.002;          % atrito dinâmico no motor;
Ct = 0.0001;        % atrito dinâmico no pendulo;
R = 0.9246;         % Resistência interna do motor
Km = 0.54;          % coeficiente Km do motor
Ke = 0.34;          % coeficiente Ke do motor
r = 0.0316;         % raio da roda
Ic = (0.3096)*10^-4; % Inércia da roda;
Vcc = 6;            % tensão de alimentação do motor [volts]
d_z = 0.05;         % zona morta do motor

termo1 = (I+m*l^2)*s^2- m*g*l;
num1 = [-m*l*s];
den1 = minreal([(M+ m+ Ic/r^2)*s+(2*Km*Ke/R + 2*b)/r^2]);
termo2 = minreal((num1/den1)*m*l*s^2);
num2 = [num1/den1]*(2*Km*Vcc/(R*r));
den2 = minreal([termo1 + termo2]);
Gs = minreal(num2/den2)
Polo = pole(Gs)
zero = zero(Gs)
% rlocus(Gs);
K = minreal(Gs*(s-Polo(1))*(s-Polo(2))*(s-Polo(3))/(s-zero))
Gs = K*(s-zero)/[(s-Polo(1))*(s-Polo(2))*(s-Polo(3))]

% projeto do controlador
OS= 0.1;
Tac= 1.25;
qsi = (-log(OS))/sqrt(pi^2 + (log(OS))^2);
wn = 4/(qsi*Tac);
P1 = wn*(-qsi + sqrt(-1+qsi^2))
%Gc = (Kd*s^2 + Kp*s + Ki)/s
b=0.4*imag(P1);
% Gc = K(s^2 + 2as + (a^2+b^2))/s
%phase do controlador + phase da planta = -180
phase_sGc = phase_controlador + phase(P1);
tg_t = tan(phase_sGc)

Re= real(P1);
Im=imag(P1);
G_teste = 1/[tg_t*[Re^2+s^2+2*s*Re+b^2-Im^2]-[2*Re*Im+2*s*Im]]
a_t = pole(G_teste)
a=a_t(1)
Gc = (s^2 + 2*a*s + (a^2+b^2))/s
% rlocus(Gs*Gc)
K_total = K*abs(P1)*abs(P1+a-b*i)*abs(P1+a+b*i)/(abs(P1)*abs(P1-
Polo(1))*abs(P1-Polo(2))*abs(P1-Polo(3)))
Gc = minreal((1/-K_total)*(s+a-b*i)*(s+a+b*i)/s)
rlocus(Gs*Gc)

```

## APÊNDICE IV – Demonstração da possibilidade de utilização da representação do PID

$$G_c = \frac{K_d \left( s^2 + \left( \frac{K_p}{K_d} \right) s + \left( \frac{K_i}{K_d} \right) \right)}{s} = \frac{K_d (s^2 + 2as + (a^2 + b^2))}{s}$$

$$K_i > K_p \rightarrow a^2 + b^2 > 2a \rightarrow a^2 - 2a + 1 > 1 - b^2 \rightarrow (a + 1)^2 > 1 - b^2$$

$$\text{se } b > 1 \rightarrow (1 - b^2) < 0 \rightarrow (a - 1)^2 > 0 > (1 - b^2) \text{ para qual quera}$$

$$\text{se } 0 < b < 1 \rightarrow |a - 1| > \sqrt{1 - b^2} \rightarrow a > 1 + \sqrt{1 - b^2} \text{ ou } a < 1 - \sqrt{1 - b^2}$$

$$\log a |a - 1| < \sqrt{1 - b^2} \text{ se } 1 - \sqrt{1 - b^2} < a < 1 + \sqrt{1 - b^2}$$

$$f(a) = a^2 + b^2 - 2a = \frac{K_i - K_p}{K_d}$$

$$f(a) > 0 \text{ se } a > 1 + \sqrt{1 - b^2} \text{ ou } 0 < a < 1 - \sqrt{1 - b^2} \text{ ou } b > 1$$

$$f(a) < 0 \text{ se } 1 - \sqrt{1 - b^2} < a < 1 + \sqrt{1 - b^2} \text{ e } 0 < b < 1$$

$$f(a) \text{ tem míni moem } a = 1 \Rightarrow \frac{K_p}{K_d} = 2 \text{ e } \frac{K_i}{K_d} = 1 + b^2$$

Conclusões:

Para um valor  $K_i$  maior que o valor de  $K_p$  existe quaisquer combinações possíveis dessas constantes para essa estrutura de controlador.

Para um valor de  $K_i$  menor que o valor de  $K_p$  esse controlador restringe os valores possíveis do valor de  $K_i$ , porém, a experiência com a dinâmica do pêndulo invertido mostra que um  $K_i$  melhora principalmente a estabilidade da base do veículo de duas rodas, assim, não se perde nada nessa configuração de controlador.

## APÊNDICE V – software em arduino para aplicação do controle do veículo de duas rodas

```

#include <Wire.h>
#include <Kalman.h>

#define M1_IN1 4
#define M1_IN2 5
#define M2_IN1 4
#define M2_IN2 5
#define M_PWM 3
#define T_amostragem 2e-3
#define PI_T 3.1415
#define PWM_NVL 4096*6/8

Kalman kalmanX; // Create the Kalman instances
float kalmanY;

/* IMU Data */
double accX, accY, accZ;
double gyroX;
double kalAngleX, kalAngleY; // Calculated angle using a Kalman filter
double omega;
double tempo = 0;
float t_estabiliza = 10;
uint32_t N_Ciclo = 0;
uint32_t timer;
uint32_t timer1;
uint8_t i2cData[15]; // Buffer for I2C data
double theta;
double integral = 0;
float uot;

void control()
{
    uot = 1.68*(theta * PI_T / 180) + 43.6*(integral*PI_T/180)+
    0.0162*(omega*PI_T/180);

    int x;
    x = abs(uot) * PWM_NVL;
    if (x > PWM_NVL) x = PWM_NVL;

    if (uot > 0) {
        analogWrite(M1_IN1, 0);
        analogWrite(M1_IN2, 4096);
        analogWrite(M_PWM, x);
    }
    else if (uot < 0) {
        analogWrite(M1_IN1, 4069);
        analogWrite(M1_IN2, 0);
        analogWrite(M_PWM, x);
    }
    else {
        analogWrite(M1_IN1, 4096);
        analogWrite(M1_IN2, 4096);
        analogWrite(M_PWM, 0);
    }
}

void setup() {

```

```

Serial.begin(115200);
pinMode(M_PWM, OUTPUT);
pinMode(M1_IN1, OUTPUT);
pinMode(M1_IN2, OUTPUT);
pinMode(M2_IN1, OUTPUT);
pinMode(M2_IN2, OUTPUT);
pinMode(LED_BUILTIN, OUTPUT);
digitalWrite(LED_BUILTIN, HIGH);
Wire.begin();
#if ARDUINO >= 157
Wire.setClock(100000UL); // Set I2C frequency to 100kHz, Standard-
mode
#else
TWBR = ((F_CPU / 100000UL) - 16) / 2; // Set I2C frequency to 100kHz
, Standard-mode
#endif

i2cData[0] = 7; // Set the sample rate to 1000Hz - 8kHz/(7+1) =
1000Hz
i2cData[1] = 0x00; // Disable FSYNC and set 260 Hz Acc filtering,
256 Hz Gyro filtering, 8 KHz sampling
i2cData[2] = 0x00; // Set Gyro Full Scale Range to  $\pm 250$ deg/s
i2cData[3] = 0x00; // Set Accelerometer Full Scale Range to  $\pm 2g$ 
delay(100);
while (i2cWrite(0x19, i2cData, 4, false)); // Write to all four
registers at once
delay(100);
while (i2cWrite(0x6B, 0x01, true)); // PLL with X axis gyroscope
reference and disable sleep mode
delay(1000);
while (i2cRead(0x3B, i2cData, 6));
accX = (int16_t)((i2cData[0] << 8) | i2cData[1]);
accY = (int16_t)((i2cData[2] << 8) | i2cData[3]);
accZ = (int16_t)((i2cData[4] << 8) | i2cData[5]);

#ifdef RESTRICT_PITCH // Eq. 25 and 26
double roll = atan2(accY, accZ) * RAD_TO_DEG;
double pitch = atan(-accX / sqrt(accY * accY + accZ * accZ)) *
RAD_TO_DEG;
#else // Eq. 28 and 29
double roll = atan(accY / sqrt(accX * accX + accZ * accZ)) *
RAD_TO_DEG;
double pitch = atan2(-accX, accZ) * RAD_TO_DEG;
#endif

kalmanX.setAngle(roll); // Set starting angle
kalmanY = pitch;
timer = micros();
timer1 = micros();
}

void loop() {

/* Update all the values */
while ((i2cRead(0x3B, i2cData, 14))) {
}
accX = (int16_t)((i2cData[0] << 8) | i2cData[1]);
accY = (int16_t)((i2cData[2] << 8) | i2cData[3]);
accZ = (int16_t)((i2cData[4] << 8) | i2cData[5]);
gyroX = (int16_t)((i2cData[8] << 8) | i2cData[9]);

```

```

#ifdef RESTRICT_PITCH // Eq. 25 and 26
    double roll = atan2(accY, accZ) * RAD_TO_DEG;
    double pitch = atan(-accX / sqrt(accY * accY + accZ * accZ)) *
RAD_TO_DEG;
#else // Eq. 28 and 29
    double roll = atan(accY / sqrt(accX * accX + accZ * accZ)) *
RAD_TO_DEG;
    double pitch = atan2(-accX, accZ) * RAD_TO_DEG;
#endif

    omega = (gyroX / 131.0); // Convert to deg/s
    double dt = (double)(micros() - timer) / 1000000; // Calculate delta
time
    timer = micros();

    // This fixes the transition problem when the accelerometer angle
jumps between -180 and 180 degrees
    if ((pitch < -90 && kalAngleY > 90) || (pitch > 90 && kalAngleY < -
90)) {
        kalmanY = pitch;
    } else
        kalAngleY = pitch; // Calculate the angle using a Kalman filter
    if (abs(kalAngleY) > 90) omega = -omega; // Invert rate, so it fits
the restricted accelerometer reading

    //VARIÁVEIS DINÂMICAS//

    omega = omega + 3.8847 + 0.1267;
    kalAngleX = kalmanX.getAngle(roll + 0.1193, omega, dt); // Calculate
the angle using a Kalman filter
    float cft = 0.75;
    theta = (1 - cft) * (0.9 * (theta + omega * dt) + 0.1 * (roll +
0.1193)) + cft * kalAngleX ; // Calculate the angle using a
Complimentary filter
    integral = integral + theta * dt;
    digitalWrite(LED_BUILTIN, HIGH );

    tempo = (double)(micros() - timer1) / 1000000;
    if (tempo > (double) N_Ciclo * T_amostragem) {
        N_Ciclo = N_Ciclo + 1;
        //Serial.print(tempo); Serial.print("\t");
        //Serial.print(omega); Serial.print("\t");
        Serial.print(theta); Serial.print("\t");
        control();
        Serial.print(omega); //Serial.print("\t");
        Serial.print("\r\n");
    }
}

```

```

    const uint8_t IMUAddress = 0x68; // AD0 is logic low on the PCB
    const uint16_t I2C_TIMEOUT = 1000; // Used to check for errors
in I2C communication

```



```

uint8_t i2cWrite(uint8_t registerAddress, uint8_t data, bool
sendStop) {
    return i2cWrite(registerAddress, &data, 1, sendStop); //
Returns 0 on success
}

uint8_t i2cWrite(uint8_t registerAddress, uint8_t *data,
uint8_t length, bool sendStop) {
    Wire.beginTransmission(IMUAddress);
    Wire.write(registerAddress);
    Wire.write(data, length);
    uint8_t rcode = Wire.endTransmission(sendStop); // Returns 0
on success
    if (rcode) {
        Serial.print(F("i2cWrite failed: "));
        Serial.println(rcode);
        digitalWrite(LED_BUILTIN, LOW );
        analogWrite(M1_IN1, 4096);
        analogWrite(M1_IN2, 4096);
        analogWrite(M_PWM, 0);
        delay(50000);
    }
    return rcode; // See:
http://arduino.cc/en/Reference/WireEndTransmission
}

uint8_t i2cRead(uint8_t registerAddress, uint8_t *data, uint8_t
nbytes) {
    uint32_t timeOutTimer;
    Wire.beginTransmission(IMUAddress);
    Wire.write(registerAddress);
    uint8_t rcode = Wire.endTransmission(false); // Don't release
the bus
    if (rcode) {
        Serial.print(F("i2cWrite failed: "));
        Serial.println(rcode);
        digitalWrite(LED_BUILTIN, LOW );
        analogWrite(M1_IN1, 4096);
        analogWrite(M1_IN2, 4096);
        analogWrite(M_PWM, 0);
        delay(50000);
    }
}

```

```

        return          rcode;          //          See:
http://arduino.cc/en/Reference/WireEndTransmission
    }
    Wire.requestFrom(IMUAddress, nbytes, (uint8_t)true); // Send
a repeated start and then release the bus after reading
    for (uint8_t i = 0; i < nbytes; i++) {
        if (Wire.available())
            data[i] = Wire.read();
        else {
            timeOutTimer = micros();
            while ((micros() - timeOutTimer) < I2C_TIMEOUT) &&
!Wire.available());
            if (Wire.available())
                data[i] = Wire.read();
            else {
                Serial.println(F("i2cRead timeout"));
                return 5; // This error value is not already taken by
endTransmission
            }
        }
    }
    return 0; // Success
}

```

## APÊNDICE VI – Software em Matlab para a simulação do veículo de duas rodas: Função PRINCIPAL

```

% Programa que chama a função dinâmica e execura a solução numérica do
% sistema dinâmico
clear all; clc; close all

T_o=0; % Tempo inicial da simulação
T_f=2; % Tempo total de simulação
% definição de varáveis Globais importantes
global Up
global Ta
Ta = 2; % (ms) % Tempo de amostragem
Up =0; % Ação de controle no instante
passado
h=0.0001; % passo na solução numérica do
sistema
n=(T_f-T_o)/h; % número total de pontos
wr=zeros(10,1); % vetor de estado passado com
condições iniciais nulas
w=zeros(10,1); % vetor de estado atual
resposta =zeros(10,n+1); % matriz que salva os valores da
simulação
k1=zeros(10,1); % Coeficientes K1, K2, K3 e K4 do RK
4° ordem
k2=k1;
k3=k1;
k4=k1;
wr(3,1)= -5.2*pi/180; % condições iniciais;

% Solução numérica utilizando um RK de 4° Ordem
for i=1:1:(n+1)
    t= h*i;
    % RK 4° ordem

    Linear = 0; % escolhe entre o modelo linear e o modelo não linear

    if(Linear ==0)
        % Não LINEAR
        [k1 U1 i1]=(Pendulo3G(t,wr,h));
        Up = U1;
        [k2 U2 i2]=(Pendulo3G(t+h/2,wr+h*k1/2,h));
        Up = U2;
        [k3 U3 i3]=(Pendulo3G(t+h/2,wr+h*k2/2,h));
        Up = U3;
        [k4 U4 i4]=(Pendulo3G(t+h,wr+h*k3,h));
        end

    if(Linear ==1)
        %LINEAR
        [k1 U1 i1]=(Pendulo3G_L(t,wr,h));
        Up = U1;
        [k2 U2 i2]=(Pendulo3G_L(t+h/2,wr+h*k1/2,h));
        Up = U2;
        [k3 U3 i3]=(Pendulo3G_L(t+h/2,wr+h*k2/2,h));
        Up = U3;
        [k4 U4 i4]=(Pendulo3G_L(t+h,wr+h*k3,h));
        end

    U = (U1+2*U2+2*U3+ U4)/6; % Tensão normalizada

```

```

    i_m = (i1+2*i2+2*i3+ i4)/6;      % Corrente normalizada
    Up = U;                          % Ação de controle em tensão passada
    w= wr+ (h/6)*(k1+2*k2+2*k3+k4); % Calculando o vetor de estado atual
    w(9,1)= wr(9,1)+h*w(3,1);       % Variável auxiliar utilizada no
cálculo da integral do Ângulo theta
    wr=w;                            % Salva o vetor de estado atual para
o vetor passado do próximo ciclo
    % salvando os resultados e modificando as variáveis auxiliares
    resposta(1,i)=w(1,1);
    resposta(2,i)=w(2,1);
    resposta(3,i)=w(3,1);
    resposta(4,i)=w(4,1);
    resposta(5,i)=w(5,1);
    resposta(6,i)=w(6,1);
    resposta(7,i)=w(7,1);
    resposta(8,i)=i_m;
    resposta(9,i)=0;
    resposta(10,i)=U;
end

% Plotando os resultados

T=T_o:h:T_f;
figure %posição
hold on
plot(T,100*resposta(1,:))
title('Posição da base');
xlabel('tempo t_[s]');
ylabel('Posição x_[cm]');
hold off

% figure %Velocidade
% hold on
% plot(T,resposta(2,:))
% title('Velocidade do carrinho');
% xlabel('tempo_[s]');
% ylabel('velocidade v_[m/s]');
% hold off

figure %Velocidade angular
hold on
plot(T,resposta(4,:))
title('velocidade angular do pendulo');
xlabel('tempo_[s]');
ylabel('Velocidade Angular w_[rad/s]');
hold off

figure %ação de controle
hold on
plot(T,resposta(10,:)/7)
title('tensão');
xlabel('tempo_[s]');
ylabel('V');
hold off

figure %angulo
hold on
plot(T,(180/pi)*resposta(3,:))
title('Posição angular do pendulo');
xlabel('tempo_[s]');
ylabel('Posição Angular_[graus]');

```

```

hold off

% w_roda = (resposta(2,:))/0.0316; %[rad/s]
% w_roda = w_roda/(2*pi);          %[Hz, RPS]
% w_roda = w_roda*60;              % RPM
% figure %Velocidade angular da roda
% hold on
% plot(T,w_roda)
% title('velocidade angular da roda');
% xlabel('tempo_[s]');
% ylabel('Velocidade Angular da roda RPM');
% hold off

% Calculos estatísticos do resultado
M_angulo =
(180/pi)*sum(resposta(3,5/h:length(resposta(3,:))))/length(resposta(3,:))
EQM_angulo = (sum(resposta(3,:).*(resposta(3,:)))/length(resposta(3,:))
EQM_tensao =
(sum(resposta(10,:).*(resposta(10,:)))/length(resposta(10,:))

% salvando em um arquivo txt os resultados da simulação
U_b = resposta(10,+)/6;
theta_b = resposta(3,);
x_b = resposta(1,);
T_p = T_o:0.001*Ta:T_f;
n = 0;

for i = 1:1:length(x_b);
    if( (i-1) >= (n*0.001*Ta/h))
        theta(n+1)=theta_b(i);
        U(n+1) = U_b(i);
        n = n+1;
    end
end

fid = fopen('NOVO_P_Resultados_PID_SS_teorico.txt','wt');
A0011 = [T_p; U ;theta];
fprintf(fid,'%f %f %f\n',A0011);
fclose(fid);

```

## APÊNDICE VII – Software em Matlab para a simulação do veículo de duas rodas: Função Fpend NÃO LINEAR.

```

function [dXdT Q i_m] = Pendulo3G(t,X,h)
%UNTITLED3 Summary of this function goes here
% t - instante de tempo;
% X - Vetor coluna X= [x1,x2,x3,x4]';
Y = zeros(10,1);

% ação de controle
%V = fuzzyc(t,X,h);
V =PID(t,X,h);

%tensões de cada motor
Ve = V;
Vd = V;
Q = V;

% definição das constantes com valores em SI
l = 0.0748;           % distância entre o eixo e o centro de massa
g = 9.79;            % valor aproximado da gravidade em Uberlândia
M = 0.062;           % massa não suspensa, 2 rodas;
m = 1.304;           % massa pendulo, estrutura e componentes;
I = 48*10^-4;        % calculado aproximadamente
Cx = 0;              % coeficiente de atrito estático entre o pneu e o
chão;
b = 0.002;           % atrito dinâmico no motor;
Ct = 0.0001;         % atrito dinâmico no pendulo;
R = 0.9246;          % Resistência interna do motor
Km2 =0.54 ;          % coeficiente Km do motor 0.5373
Km1 =0.34;           % coeficiente Ke do motor 34
r =0.0316;           % raio da roda
Ic =(0.3096)*10^-4; % Inércia da roda;
Iz = 7.16*10^-3;    % distância entre eixos
d = 0.23;            % declaração de um distúrbio externo interpretado
T=0;                 em torque
Vcc = 6;              % tensão de alimentação do motor [volts]
d_z = 0.05;          %zona morta do motor, 3% no motor esquerdo e 3.5%
no motor direito.

%definição das variáveis de estado
x = X(1,1);          % Posição da base
vx = X(2,1);         % Velocidade da base
theta= X(3,1);       % Ângulo com a vertical do pêndulo
w = X(4,1);          % Velocidade angular do pêndulo
fi = X(5,1);         % Rotação em torno da vertical de pêndulo
dfi = X(6,1);        % Velocidade angular em torno da vertical

% dinâmica dos motores

% Velocidade angular do eixo das rodas
wd = (vx+d*dfi)/r;
we = (vx-d*dfi)/r;

% Calculo da corrente nos motores

```

```

if(abs(Vd) < d_z*Vcc)
    id = 0;
else
    id = (Vd - Km1*wd)/R;
end
if(abs(Ve) < d_z*Vcc)
    ie = 0;
else
    ie = (Ve - Km1*we)/R;
end

% Cálculo dos torques, considerando as inércias no modelo do pêndulo
i_m = id;
TRd = Km2*id;
TRe = Km2*ie;

% Dinâmica do pendulo

u = (TRd + TRe)/r;
term3 = (Iz + 2*Ic*(d/r)^2)*(r/d);
Y(5,1) = X(6,1);
Y(6,1) = (TRd - TRe)/term3;

% % Inserindo um distúrbio em forma de delta de Dirac no sistema no
tempo t
% if (t==10)
%     T = 0*1/h;
%     char = 'metade'
% end

T = m*g*1*1*(pi/180)*normrnd(0,1);
term1 = m + M + 2*Ic/(r^2);
term2 = I+m*1^2-((m*1*cos(theta))^2)/term1;
Y(1,1)=X(2,1);
Y(3,1)=X(4,1);
T_u = -(m*1*cos(theta))/term1;
T_w = -(2*Ct +(w*sin(2*theta)*(m*1)^2)/(2*term1));
T_t = m*1*g*sin(theta);
T_vx = m*1*cos(theta)*((2*b/r^2)+Cx)/term1 + (2*Ct/r);
T_ext = T;
Y(4,1) = (T_ext + T_t + T_u*u + T_w*w + T_vx*vx)/term2;
Y(2,1)=(u -((2*b/r^2)+Cx)*vx +sin(theta)*m*1*w^2 -
m*1*cos(theta)*Y(4,1))/term1;
Y(9:10,1)=0;
dXdt = Y;
end

```

## APÊNDICE VII – Software em Matlab para a simulação do veículo de duas rodas: Função Fpend LINEAR.

```

function [dXd_t Q i_m] = Pendulo3G_L(t,X,h)
%UNTITLED3 Summary of this function goes here
% t - instante de tempo;
% X - Vetor coluna X= [x1,x2,x3,x4]';
Y = zeros(10,1);

% ação de controle
%V = fuzzyc(t,X,h);
V =PID(t,X,h);

%tensões de cada motor
Ve = V;
Vd = V;
Q = V;

% definição das constantes com valores em SI
l = 0.0748;           % distância entre o eixo e o centro de massa
g = 9.79;            % valor aproximado da gravidade em Uberlândia
M = 0.062;           % massa não suspensa, 2 rodas;
m = 1.304;           % massa pendulo, estrutura e componentes;
I = 48*10^-4;        % calculado aproximadamente
Cx = 0;              % coeficiente de atrito estático entre o pneu e o
chão;
b = 0.002;           % atrito dinâmico no motor;
Ct = 0.0000001;      % atrito dinâmico no pendulo;
R = 0.9246;          % Resistência interna do motor
Km2 =0.5373;         % coeficiente Km do motor
Km1 =0.34;           % coeficiente Ke do motor
r =0.0316;           % raio da roda
Ic =(0.3096)*10^-4; % Inércia da roda;
Iz = 7.16*10^-3;     % distância entre eixos
d = 0.23;            % declaração de um distúrbio externo interpretado
T=0;                 em torque
Vcc = 6;              % tensão de alimentação do motor [volts]
d_z = 0.05;          %zona morta do motor, 3% no motor esquerdo e 3.5%
no motor direito.

%definição das variáveis de estado
x = X(1,1);          % Posição da base
vx = X(2,1);         % Velocidade da base
theta= X(3,1);       % Ângulo com a vertical do pêndulo
w = X(4,1);          % Velocidade angular do pêndulo
fi = X(5,1);         % Rotação em torno da vertical de pêndulo
dfi = X(6,1);        % Velocidade angular em torno da vertical

% dinâmica dos motores

% Velocidade angular do eixo das rodas
wd = (vx+d*dfi)/r;
we = (vx-d*dfi)/r;

% Calculo da corrente nos motores

```



```

if(abs(Vd) < d_z*Vcc)
    id = 0;
else
    id = (Vd - Km1*wd)/R;
end
if(abs(Ve) < d_z*Vcc)
    ie = 0;
else
    ie = (Ve - Km1*we)/R;
end

% Cálculo dos torques, considerando as inércias no modelo do pêndulo
i_m = id;
TRd = Km2*id;
TRe = Km2*ie;

% Dinâmica do pendulo

u = (TRd + TRe)/r;
term3 = (Iz + 2*Ic*(d/r)^2)*(r/d);
Y(5,1) = X(6,1);
Y(6,1) = (TRd - TRe)/term3;

% % Inserindo um distúrbio em forma de delta de Dirac no sistema no
tempo t
% if (t==10)
%     T = 0*1/h;
%     char = 'metade'
% end

T = m*g*1*1*(pi/180)*normrnd(0,1);
term1 = m + M + 2*Ic/(r^2);
term2 = I+m*1^2-((m*1*1)^2)/term1;
Y(1,1)=X(2,1);
Y(3,1)=X(4,1);
T_u = -(m*1*1)/term1;
T_w = -(2*Ct);
T_t = m*1*g*(theta);
T_vx = m*1*1*((2*b/r^2)+Cx)/term1 + (2*Ct/r);
T_ext = T;
Y(4,1) = (T_ext + T_t + T_u*u + T_w*w + T_vx*vx)/term2;
Y(2,1)=(u - ((2*b/r^2)+Cx)*vx-m*1*1*Y(4,1))/term1;
Y(9:10,1)=0;
dXdt = Y;
end

```

## APÊNDICE VIII – Software em Matlab para a simulação do veículo de duas rodas: Função Fcontrol PID.

```
function [U] = PID(t,X,h)
teta = X(3,1)+ 1*(pi/180)*normrnd(0,1);
omega = X(4,1);
integral = X(9,1);
global Up;
global Ta;
U = Up;
Tt = t/h; % garantindo que o Tt seja um número inteiro
Tam = (Ta/1000);
if mod(Tt,Tam/h)<= 0.05*Tam/h
uot = 1.68*teta + 43.6*integral + 0.0162*omega;
%uot = 1.655*teta + 21.98*integral + 0.0278*omega;
%uot = 1.675*teta + 25.07*integral + 0.023*omega;
U = 6*uot;
end

    if U > 6
    U = 6;
    end
    if U < -6
    U = -6;
    end
end
```

## APÊNDICE IX – Software em Matlab para a simulação do veículo de duas rodas: Função Fcontrol FUZZY.

```

Function [U] = fuzzyc(t,X,h)
%UNTITLED2 Summary of this function goes here
% Detailed explanation goes her
teta = X(3,1)+ 1*(pi/180)*normrnd(0,1);
ttg = abs(teta*180/pi);% angulo em graus
omega = X(4,1);
omg = abs(omega*180/pi);% velocidade em graus/s
veloc = X(2,1);
a = zeros(1,7);
b = zeros(1,5);
integral = X(9,1);
global Up;
global Ta;
U = Up;
Tt = t/h; % garantindo que o Tt seja um número inteiro
Tam = Ta/1000;

if mod(Tt,Tam/h)<=0.05*Tam/h

if ttg<=1
a(4)=100*(1-ttg);
if teta >= 0
a(5)=100-a(4);
else
a(3)=100-a(4);
end
else
if ttg <=5
if(teta>0)
a(6)=25*(ttg-1);
a(5)=100-a(6);
else
a(2)=25*(ttg-1);
a(3)=100-a(2);
end
else
if ttg<= 10
if(teta>0)
a(7)=20*(ttg-5);
a(6)=100-a(7);
else
a(1)=20*(ttg-5);
a(2)=100-a(1);

end
else
if(teta>0)
a(7)=100;
else
a(1)=100;
end

end
end
end
end
end

```

```

if omg<=20
    b(3)=5*(20-omg);
    if omega>=0
        b(4)=100-b(3);
    else
        b(2)=100-b(3);
    end
end
else
    if omg<=100
        if omega>0
            b(5)=1.25*(omg-20);
            b(4)=100-b(5);
        else
            b(1)=1.25*(omg-20);
            b(2)=100-b(1);
        end
    else
        if omg>100
            if omega>0
                b(5)=100;
            else
                b(1)=100;
            end
        end
    end
end
end
c(1)= 1*(a(1)*(b(3)+b(2)+b(1))+a(2)*(b(2)+b(1))+a(3)*b(1))/10000;
c(2)= 1*(a(1)*b(4)+a(2)*b(3)+a(3)*b(2)+a(4)*b(1))/10000;
c(3)= 1*(a(1)*b(5)+a(2)*b(4)+a(3)*b(3)+a(4)*b(2)+a(5)*b(1))/10000;
c(4)= 1*(a(2)*b(5)+a(3)*b(4)+a(4)*b(3)+a(5)*b(2)+a(6)*b(1))/10000;
c(5)= 1*(a(3)*b(5)+a(4)*b(4)+a(5)*b(3)+a(6)*b(2)+a(7)*b(1))/10000;
c(6)= 1*(a(4)*b(5)+a(5)*b(4)+a(6)*b(3)+a(7)*b(2))/10000;
c(7)= 1*(a(5)*b(5)+a(6)*(b(4)+b(5))+a(7)*(b(3)+b(4)+b(5)))/10000;
fator_peso=1;
y=zeros(1,7);
h=0.01;
soma1=0;
soma2=0;
for ui=h:h:1
    if ui<=0.25
        y(5)=(ui)/0.25;
        y(3)=y(5);
        y(4)=1-y(5);
    else
        if ui<=0.5
            y(6)=(ui-0.25)/0.25;
            y(2)=y(6);
            y(5)=1-y(6);
            y(3)=y(5);
        else
            if ui<=0.75
                y(7)=(ui-0.5)/0.25;
                y(1)=y(7);
                y(6)=1-y(7);
                y(2)=y(6);
            else
                y(7)=1;
                y(1)=1;
            end
        end
    end
end
end
end

```

```

YP = max([c(4)*y(4),c(5)*y(5),c(6)*y(6),c(7)*y(7)])/fator_peso;
YN = max([c(4)*y(4),c(3)*y(3),c(2)*y(2),c(1)*y(1)])/fator_peso;
soma1 = soma1 + (ui+h/2)*YP + (-ui-h/2)*YN;
soma2 = soma2 + YP + YN;
end
    soma1 = soma1 + c(4)*h/2;
    soma2 = soma2 + c(4);
    u1 = soma1/(soma2+0.0001);
    fator_fuzzy = 0.5;
    C_PID = 1.68*teta + 43.6*integral + 0.0162*omega;
    u = fator_fuzzy*u1 + (1-fator_fuzzy)*C_PID;
    U = 6*u;
    if U>6
    U = 6;
    end
    if U < -6
    U = -6;
    end
    % 1.5956e-05
end
end

```