



UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Daniel Oliveira de Castro

**CONSTRUÇÃO DE UM LEVITADOR
PNEUMÁTICO UTILIZANDO CONTROLE
PID PARA O LABORATÓRIO DE CONTROLE
DA UNIVERSIDADE DO ESTADO DO
AMAZONAS**

Manaus
2015

Daniel Oliveira de Castro

**CONSTRUÇÃO DE UM LEVITADOR
PNEUMÁTICO UTILIZANDO CONTROLE
PID PARA O LABORATÓRIO DE CONTROLE
DA UNIVERSIDADE DO ESTADO DO
AMAZONAS**

Trabalho de Conclusão de Curso submetido à Coordenação do curso de Engenharia de Controle e Automação da Universidade do Estado do Amazonas como parte dos requisitos necessários para a obtenção do grau de Engenheiro.

Orientador Dr. Israel Mazaira Morales

Manaus
2015

Daniel Oliveira de Castro

**CONSTRUÇÃO DE UM LEVITADOR PNEUMÁTICO
UTILIZANDO CONTROLE PID PARA O
LABORATÓRIO DE CONTROLE DA UNIVERSIDADE
DO ESTADO DO AMAZONAS**

Trabalho de Conclusão de Curso submetido à Coordenação do curso de Engenharia de Controle e Automação da Universidade do Estado do Amazonas como parte dos requisitos necessários para a obtenção do grau de Engenheiro.

Aprovado em 25 de Junho de 2015.

BANCA EXAMINADORA

Dr. Israel Mazaira Morales
Orientador

Dr. Walter Andrés Vermehren
Valenzuela
Presidente da banca

Dr. Daniel Guzman del Rio
Convidado 1

Dr. Israel Francisco Benitez Pina
Convidado 2

Manaus
2015

Agradecimentos

A Deus, por ter cuidado de mim durante todos esse anos e por ter me dado paciência e persistência para não desistir ante as muitas dificuldades. A minha mãe Edenice Castro e meu pai Ismael Castro, por seus apoios e incentivos, me ensinando valores que para sempre levarei. Ao meu irmão Iamael Azevedo, que muito colaborou com ideias para melhorias no projeto. À minha tia Iva Castro, que muito me ajudou a iniciar minha vida acadêmica, desde o Ensino Médio. Também ao meu tio Clóvis Castro, que mesmo não estando mais em vida, será sempre lembrado pelas suas histórias e lições de vida. Aos meus amigos: Dilermando Ferreira, Rodrigo Santa Rita, Rafael Goulart e Antônio Benjamim, que com mútua ajuda levamos a faculdade. Agradeço à Coordenação do Curso de Engenharia de Controle e Automação e todos os professores, especialmente ao meu orientador Prof. Israel Mazaira Morales, e também à Universidade do Estado do Amazonas, que juntos forneceram os subsídios necessários para a realização deste projeto.

*“O único lugar aonde o sucesso
vem antes do trabalho,
é no dicionário“
(Albert Einstein)*

Resumo

Foi desenvolvido um Levitador Pneumático para o Laboratório de Controle da Universidade do Estado do Amazonas, constituído de uma estrutura física onde um corpo é levitado com um fluxo de ar proveniente de uma ventoinha. Um sensor de distância foi responsável pela medição da posição em que o objeto se posicionava. Um aplicativo de controle foi desenvolvido no programa Matlab® onde o usuário determina a altura que o objeto devia estacionar. Neste Aplicativo também foi realizado o processamento das informações advindas do sensor, verificando se a velocidade da ventoinha de ser aumentada ou diminuída, tendo sido possível aplicar um controlador Proporcional Integral e Derivativo nos sinais, alterando os parâmetros de modo a obter a melhor resposta possível, isto é, o melhor tempo de estabilização. Para tanto, foram estudados neste projeto alguns métodos de sintonia os quais serão aplicados no controlador podendo visualizar qual método se torna mais adequado para a estabilização do sistema conforme a resposta desejada . Um microcontrolador Arduíno foi utilizado para intermediar a comunicação entre o aplicativo, o sensor e a ventoinha. Este trabalho demonstrou de forma analítica, em detalhes e com clareza como foi desenvolvido este sistema.

Palavras-chave: Levitador. Arduíno. Controladores PID. Ventoinha.

Abstract

A Pneumatic Levitator was built to the Control and Automation Laboratory of Amazonas State University, composed by a physical structure where an object will be levitated by an air flow from a fan. A distance sensor is responsible to measure the exact position where the object will be. An Application was developed in Matlab® where the user will determine the desired stationary position of the object. This software process the data from the sensor, to verify if the fan velocity must be increased or decreased, applying a proportional-integral-derivative controller (PID controller) in signals, to change the parameters to obtain the better possible response, i.e., a better stabilizing time. For that, it was studied some tuning methods which were applied in the controller to become more suitable for the stabilization of the system and its desired response. An Arduino microcontroller was used to control the interaction between application, sensor and fan. This work demonstrates succinctly how this system will be developed.

Key-words: Levitator. Arduino. PID controllers. Fan.

Lista de ilustrações

Figura 1 – Curva de um sensor analógico	16
Figura 2 – Sinal Contínuo x Sinal Discreto	16
Figura 3 – Curva dos sensores linear e não linear	17
Figura 4 – Sinal Linearizado	17
Figura 5 – Partes de um Motor DC de ímã permanente	19
Figura 6 – Funcionamento do Motor DC de ímã permanente	19
Figura 7 – Diagrama de Bocos de um Motor DC	20
Figura 8 – Sistema eletromecânico de um motor DC	20
Figura 9 – Diagrama em Blocos de um Microcontrolador	23
Figura 10 – Onda quadrada - Clock	24
Figura 11 – Circuito Chaveado	25
Figura 12 – Onda Quadrada	25
Figura 13 – Tensão Média	25
Figura 14 – Controle de tensão por largura de pulso	26
Figura 15 – Diagrama em Blocos do Controlador PID	26
Figura 16 – Resposta ao degrau unitário de uma planta	29
Figura 17 – Curva com S em seu corpo	29
Figura 18 – Curva de Nyquist para $G_C(j\omega)G(j\omega)$	31
Figura 19 – Tela Principal do MATLAB®	32
Figura 20 – IDE do <i>Guide</i> do MATLAB®	33
Figura 21 – Diagrama em Blocos do Módulo	35
Figura 22 – Root Locus - Sem controlador	36
Figura 23 – Root Locus - Com controlador	36
Figura 24 – Sistema de levitação com controle digital e atuador chaveado	37
Figura 25 – Posição relativa do corpo em relação ao núcleo	38
Figura 26 – Diagrama em Blocos da Metodologia	40
Figura 27 – Sensor Ultrasônico HC-SR04	41
Figura 28 – Funcionamento do Sensor Ultrasônico	42
Figura 29 – Ventoinha KENLOWE 2,3”	43
Figura 30 – Testes de Tensão x Altura	45
Figura 31 – Gráfico Velocidade x Altura	46
Figura 32 – Função linear aproximada	47
Figura 33 – Microcontrolador Arduino UNO	48
Figura 34 – Envio de tensão média ao pino D6 do Arduino	49
Figura 35 – Circuito Amplificador do PWM	51
Figura 36 – Linha de comando que aplica o controle PID	51

Figura 37 – Script da Sintonia por Ziegler e Nichols	52
Figura 38 – Resposta ao degrau unitário e aproximação linear	53
Figura 39 – Ganhos por Ziegler e Nichols	53
Figura 40 – Script da Sintonia por Cohen-Coon	54
Figura 41 – Ganhos por Cohen-Coon	54
Figura 42 – Script da sintonia por método AMIGO	55
Figura 43 – Ganhos por método AMIGO	55
Figura 44 – Aplicativo de Controle do Levitador	56
Figura 45 – Script completo na memória do Arduíno	56
Figura 46 – Função de envio e recebimento de dados em MATLAB	57
Figura 47 – Estrutura do Levitador Pneumático	59
Figura 48 – Prototipagem da Placa Amplificadora	60
Figura 49 – Resposta ao Degral de Magnitude 5	61
Figura 50 – Script de envio de tensão ao Arduíno	61
Figura 51 – Altura obtida para 5V no Motor DC da Ventoinha	62
Figura 52 – Parte do código onde implementa-se o PID	62
Figura 53 – Resposta do degrau com Sintonia de Ziegler e Nichols	63
Figura 54 – Resposta do degrau com Sintonia de Cohen e Coon	64
Figura 55 – Resposta do degrau com Sintonia por AMIGO	64
Figura 56 – Aplicativo de Controle e Monitoramento	65
Figura 57 – Altura de Estabilização para o teste	66

Lista de tabelas

Tabela 1 – Parâmetros do Modelo Eletromecânico	21
Tabela 2 – Regra 1 de sintonia de Ziegler e Nichols	30
Tabela 3 – Regra de sintonia de Cohen e Coon	30
Tabela 4 – Regra de sintonia por método AMIGO	32
Tabela 5 – Especificação do Sensor HC-SR04	41
Tabela 6 – Especificação do Motor DC	43
Tabela 7 – Testes de Respostas ao Degrau	45
Tabela 8 – Características do Arduíno UNO	48
Tabela 9 – Componentes do Circuito Amplificador	50

Sumário

1	INTRODUÇÃO	12
1.1	Formulação do problema	12
1.2	Justificativa	13
1.3	Motivação	13
1.4	Objetivos	13
1.5	Metodologia	13
2	REFERENCIAL TEÓRICO	15
2.1	Sensores	15
2.1.1	Características fundamentais	15
2.1.1.1	Analógico ou Digital	15
2.1.1.2	Linearidade	17
2.1.1.3	Alcance	17
2.1.1.4	Velocidade de Resposta	18
2.2	Motores DC	18
2.2.1	Princípio de funcionamento	19
2.2.2	Modelagem de um motor de corrente contínua	20
2.3	Microcontroladores	22
2.3.1	Funcionamento	23
2.4	Modulação por largura de pulso	24
2.4.1	Princípio de funcionamento	24
2.5	Controladores PID	26
2.6	Sintonia	28
2.6.1	Método de Ziegler e Nichols - resposta ao degrau	28
2.6.2	Método de Cohen-Coon	30
2.6.3	Método AMIGO	30
2.7	MATLAB®	32
3	TRABALHOS RELACIONADOS	34
3.1	Descrição do sistema	34
3.1.1	Amplificação	35
3.1.2	Controle	35
3.1.3	Controlador digital	36
3.1.4	Sistema eletromecânico	37
3.1.5	Sensor de posição	37
3.1.6	Sistema de controle	38

3.2	Resultados	38
3.3	Comparativo	38
4	MATERIAIS E MÉTODOS	40
4.1	Sensor ultrasônico HC-SR04	40
4.1.1	Funcionamento	41
4.2	A ventoinha utilizada	43
4.2.1	Função de transferência do motor DC	44
4.2.2	Relação velocidade X altura	45
4.3	O microcontrolador arduíno UNO	47
4.3.1	Comunicação com o computador	48
4.4	O PWM aplicado	49
4.5	O amplificador do PWM	50
4.6	O controlador PID aplicado	51
4.7	Sintonia do controlador PID	52
4.7.1	Ziegler e Nichols	52
4.7.2	Cohen-Coon	54
4.7.3	AMIGO	55
4.7.4	O aplicativo de controle	56
5	RESULTADOS OBTIDOS	59
5.1	Construção da estrutura física do levitador	59
5.2	Prototipagem do amplificador	59
5.3	Modelagem e função de transferência	60
5.4	Implementação e sintonização do PID	62
5.4.1	Ziegler e Nichols:	63
5.4.2	Cohen-Coon	63
5.4.3	AMIGO	64
5.5	Testes do controlador no sistema real	65
5.5.1	Aplicativo de controle	65
6	CONCLUSÕES	67
6.1	Trabalhos futuros	68
	REFERÊNCIAS	69

1 INTRODUÇÃO

O curso de Engenharia de Controle e Automação visa formar profissionais que dominem as tecnologias de sistemas de controle automático e manufatura inteligente, visando a capacitação de profissionais nas áreas de sistemas de controle, acionamentos elétricos, eletrônica analógica e digital e computação aplicada à indústria. A área é importante por integrar as Engenharias Elétrica, Eletrônica e de Computação aplicada diretamente à indústria e com importantes implicações socioeconômicas, bem como no desenvolvimento tecnológico sustentável(JURAN, 1997).

Para formar estes profissionais com mais qualificação é importante uma sólida base acadêmica, e é neste ponto que estará o foco deste projeto.

Atualmente, na Universidade do Estado do Amazonas, existe uma grande dificuldade no que diz respeito à questão prática do estudo da Engenharia de Controle Moderno pelos universitários, uma vez que são poucos os sistemas de controle reais onde é possível que o aluno aplique uma técnica de controle moderno e veja o resultado fisicamente.

Este projeto visa diminuir a distância do aluno entre a teoria e a prática, criando um sistema de controle real, mais especificamente um Levitador Pneumático, onde o aluno poderá visualizar o funcionamento de um controlador PID na prática, podendo aplicar diferentes métodos de sintonia e obtendo assim respostas do sistema correspondentes a cada técnica utilizada.

Um levitador pneumático consiste em um sistema em que, uma fonte geradora de um fluxo de ar levitará um objeto, um sensor de distância perceberá a altura que o objeto levitado estará, um microcontrolador fará o processamento do controle aplicado, e um aplicativo de controle fará a interface entre usuário e sistema, podendo o usuário por meio desta interface aplicar técnicas de controle e sintonia mencionadas.

1.1 Formulação do problema

A precariedade dos sistemas de controle reais nos laboratórios de controle da Universidade do Estado do Amazonas impacta diretamente na aprendizagem dos universitários do curso de Engenharia de Controle e Automação, uma vez que sem o devido conhecimento prático e experiência, é muito difícil para o aluno universitário acompanhar as aulas teóricas relativas especificamente a disciplinas de controle moderno.

Até hoje, as aulas relativas a estas disciplinas são ministradas apenas de forma teórica e com simulações em softwares próprios para este fim, deixando muito a desejar a aplicabilidade do assunto abordado devido à falta de prática. Portanto, existe uma distância do aluno com a prática influenciando diretamente na sua formação acadêmica e profissional.

1.2 Justificativa

O desenvolvimento deste projeto se justifica socialmente pelo fato de trazer para o mercado de trabalho profissionais de Engenharia de Controle e Automação mais qualificados e com mais experiência prática na respectiva área de atuação, tendo como consequência a contratação de profissionais com análise crítica desenvolvida e com capacidade de empreender.

Academicamente, este projeto se justifica por necessitar em seu desenvolvimento conhecimentos de várias disciplinas estudadas durante o curso de graduação tais como: Eletrônica Analógica, Eletrônica Digital I e II, Controle I, II e III, Linguagem de programação e Microcontroladores, dentre outras.

1.3 Motivação

Vislunbrando as condições acadêmicas dos laboratórios de controle em que os universitários do curso de Engenharia de Controle e Automação da Universidade do Estado do Amazonas estudam, verificou-se a necessidade de trazer novas ferramentas de estudo para estes laboratórios, o que chamou atenção para o desenvolvimento deste projeto, uma vez que nele, são aplicados conhecimentos de diversas áreas do curso, fazendo com que os alunos possam obter mais proveito nos seus estudos acadêmicos.

Quando os universitários concluem o curso com experiência prática em seu currículo, possuem mais confiança no desenvolvimento de projetos profissionais no mercado de trabalho, levando consigo uma ótima imagem da Universidade do Estado do Amazonas, dando-lhe o prestígio de confiança no mercado.

1.4 Objetivos

Construir um levitador pneumático, utilizando um sensor ultrassônico para determinar a posição do objeto, uma ventoinha para gerar o fluxo de ar controlado, um microcontrolador Arduino UNO para realizar a comunicação e conversão de dados digitais e analógicos, e um Aplicativo de controle a ser desenvolvido no programa MATLAB® para aplicar e modificar parâmetros do Controlador Proporcional, Integral e Derivativos

1.5 Metodologia

- Levantamento bibliográfico não só com relação direta ao tema proposto mas também com relação a todos os elementos utilizados no projeto para obter uma sólida base literária.
- Realizar os testes individuais de cada dispositivo e equipamento utilizado na montagem do Levitador.

- Montagem e acoplamento de todos os dispositivos numa estrutura física.
- Interligação dos equipamentos e dispositivos em suas devidas conexões.
- Estabelecer a comunicação serial de todos os elementos do sistema(sensor, ventoinha, amplificador e microcontrolador) com o computador.
- Desenvolver o Aplicativo de Controle e supervisão do sistema.
- Projetar o Controlador Utilizado para estabilização do sistema.
- Realizar a sintonia do controlador para fazer a validação do sistema.

2 REFERENCIAL TEÓRICO

2.1 Sensores

Segundo Balbinot e Brusamarello (2011) , sensor, é o termo empregado para designar dispositivos sensíveis a alguma forma de energia do ambiente, que pode ser luminosa, térmica, cinética, relacionando informações sobre uma grandeza física que precisa ser mensurada(medida), como: temperatura, pressão, velocidade, corrente, aceleração, posição, dentre outras.

Um sensor nem sempre tem as características elétricas necessárias para ser utilizado em um sistema de controle. Normalmente o sinal de saída deve ser manipulado antes da sua leitura no sistema de controle. Isso normalmente é realizado com um circuito de interface para a produção de um sinal que possa ser lido pelo controlador (ALVES, 2005).

Resumidamente, sensores são equipamentos pelos quais o controlador monitora o processo, sendo também transdutores, uma vez que convertem grandezas físicas em grandezas elétricas.

Supondo que a saída de um sensor, ao ser sensibilizado por uma energia externa é dada por um nível de tensão muito baixo, torna-se necessária a sua amplificação. Essa interface seria então um amplificador capaz de elevar o nível do sinal para sua efetiva utilização (ALVES, 2005).

Há uma série de características relacionadas aos sensores que devem ser levadas em consideração na hora da seleção do instrumento mais indicado para uma dada aplicação. Será abordado aqui apenas as principais.

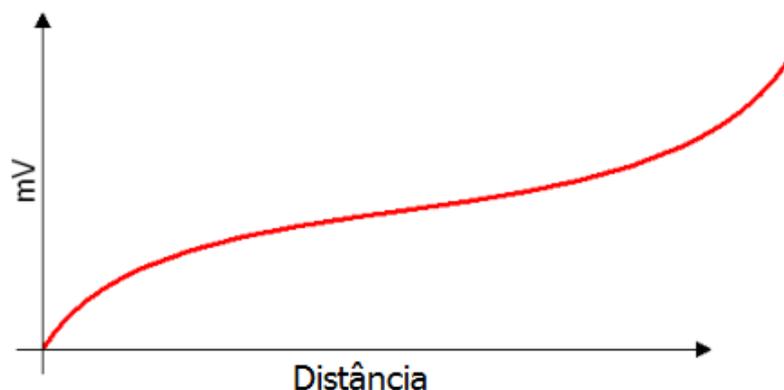
2.1.1 Características fundamentais

2.1.1.1 Analógico ou Digital

Esse modelo de sensor pode assumir ao longo do tempo, qualquer valor no seu sinal de saída contanto que esteja dentro da sua faixa de operação. Essas variáveis são mensuradas por dispositivos sensíveis com circuitos eletrônicos analógicos(THOMAZINI; ALBUQUERQUE, 2005).

A Figura 1 ilustra a variação de uma determinada grandeza física(no caso a distância) de forma analógica.

Figura 1 – Curva de um sensor analógico



Fonte: <http://goo.gl/pqG122>

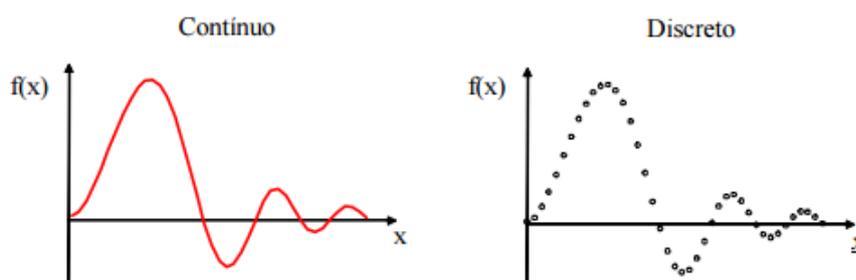
Já o modelo digital de sensor pode assumir apenas dois valores no seu sinal de saída ao longo do tempo, que podem ser interpretados como um ou zero. Obviamente, não existem grandezas físicas que assumam estes valores, entretanto eles são assim mostrados ao sistema de controle depois de serem convertidos por um circuito eletrônico (geralmente um comparador). É utilizado por exemplo, na detecção de passagem de objetos, encoders na determinação de distância ou velocidade, dentre outros (THOMAZINI; ALBUQUERQUE, 2005).

- Discretização dos sinais

Os sinais analógicos provenientes de sensores ou de qualquer outro meio, podem ser discretizados para utilização de controle digital. Para discretizar um sinal contínuo, é necessário uma amostragem de pontos correspondentes a respostas em determinado intervalo de tempo (LATHI, 2007).

Amostrar um sinal contínuo é recolher valores desse sinal (amostras) em instantes discretos. A amostragem é fundamental em processamento digital de sinal porque transforma sinais contínuos em sinais discretos equivalentes, aptos para processamento digital.

Figura 2 – Sinal Contínuo x Sinal Discreto



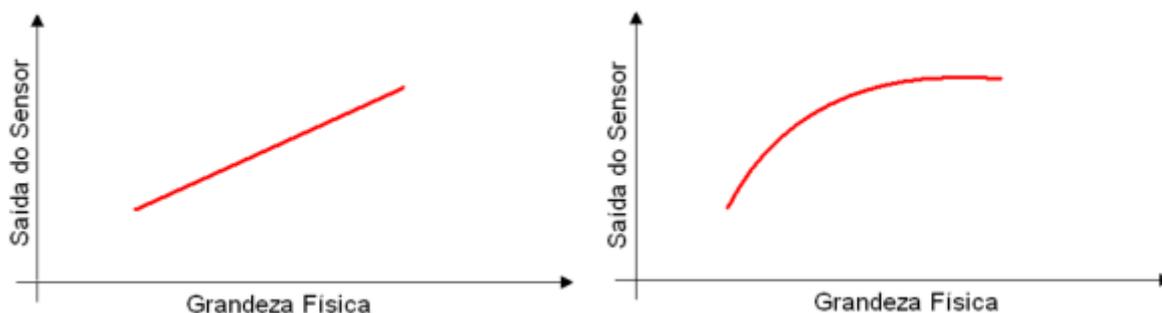
Fonte: (LATHI, 2007)

2.1.1.2 Linearidade

Esse conceito é aplicado a sensores analógicos. Trata-se da curva de saída do sensor, a partir da grandeza medida. Procuram-se respostas proporcionais às entradas, para facilitar a montagem do circuito de interface, porém muitas vezes isso não é possível, pois alguns tipos de sensores não são lineares.

A Figura 3 mostra a diferença entre um sensor linear e um não-linear.

Figura 3 – Curva dos sensores linear e não linear



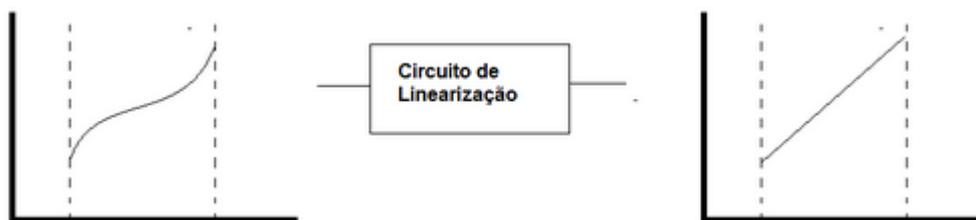
Fonte: <http://goo.gl/pqG122>

- Linearização de sinais

Para estes sinais de sensores não lineares, é possível trabalhá-los de forma a linearizar o sinal, facilitando assim a sua utilização e interpretação.

A Figura 4 ilustra um exemplo de linearização de um sinal após passar por um circuito desenvolvido com este intuito.

Figura 4 – Sinal Linearizado



Fonte: (LATHI, 2007)

2.1.1.3 Alcance

Simplemente é a faixa de valores de entrada que um sensor pode assumir (THOMAZINI; ALBUQUERQUE, 2005).

2.1.1.4 Velocidade de Resposta

Trata-se da velocidade com que o sensor fornece o valor da variável. O ideal é que o sensor possua uma resposta instantânea, pois uma resposta lenta pode prejudicar consideravelmente a eficiência do sistema de controle no que diz respeito ao tempo de resposta, influenciando o tempo de acomodação do sinal (THOMAZINI; ALBUQUERQUE, 2005).

2.2 Motores DC

Pode-se conceituar motor elétrico como sendo uma máquina destinada a transformar energia elétrica em mecânica. É o mais utilizado dentre todos os tipos de motores, uma vez que combinam as vantagens da energia elétrica (baixo custo, facilidade de transporte, limpeza e simplicidade de comando) com sua construção simples, custo reduzido, versatilidade de adaptação as cargas dos mais diversos tipos e melhores rendimentos.

De maneira geral, os motores elétricos trabalham pelo eletromagnetismo, embora hajam motores baseados em outros princípios eletromecânicos, como por exemplo forças eletrostáticas. O princípio fundamental em que os motores eletromagnéticos são baseados é que há uma força mecânica em todo o fio quando está conduzindo a eletricidade contida dentro de um campo magnético. Esta força é descrita pela lei de Lorentz sendo perpendicular ao fio e ao campo magnético. Em um motor giratório, há um elemento girando, o rotor. Os fios e o campo magnético são arrançados de maneira que o torque seja projetado sobre a linha central do rotor fazendo com que o mesmo gire.

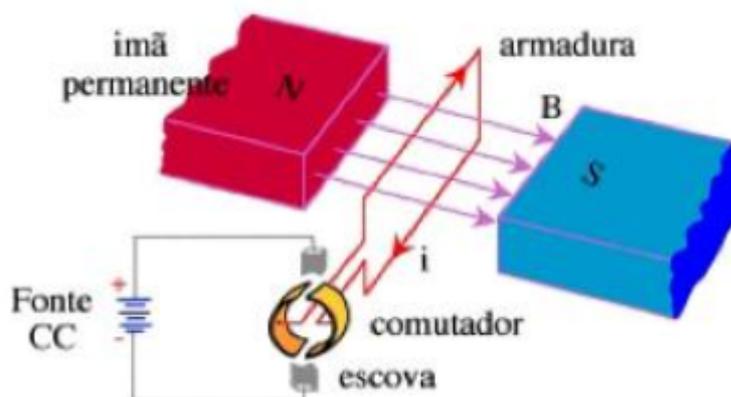
Dentre os motores elétricos, destacam-se os motores de corrente contínua (Motor DC).

Motores de corrente contínua possuem um custo relativamente elevado e, além disso, necessitam de uma fonte de corrente contínua, ou de um dispositivo que converta a corrente alternada comum em contínua. Podem funcionar com velocidade ajustável entre amplos limites e se permitem serem controlados de forma flexível e precisa. Por isso seu uso é restrito a casos especiais em que estas exigências compensam o custo muito mais alto da instalação, ou no caso da alimentação usada ser contínua, como é o caso deste projeto.

Na Figura 5 visualizamos as principais partes de um motor de corrente contínua de ímã permanente.

O núcleo da armadura está conectado ao eixo e é construído de camadas laminadas de aço, provendo uma faixa de baixa relutância magnética entre os pólos. O enrolamento da armadura é constituído de bobinas isoladas entre si e do núcleo da armadura e é posicionado nas ranhuras e eletricamente ligado ao comutador. O comutador providencia o chaveamento necessário para o processo de comutação. Os pólos são constituídos de ferro laminado conectados na carcaça, após a inserção dos rolamentos de campo nos mesmos.

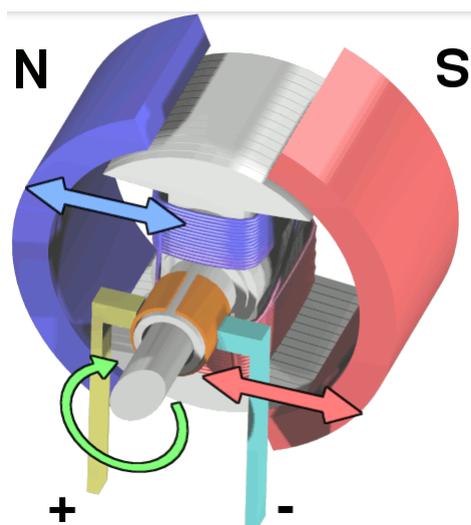
Figura 5 – Partes de um Motor DC de ímã permanente

Fonte: <http://goo.gl/hg8a69>

2.2.1 Princípio de funcionamento

Na carcaça externa, nomeada de estator, há um ímã permanente que determina uma polaridade magnética do sistema. O eixo interno, o rotor, é formado por uma bobina (um enrolamento de fios). A corrente de alimentação é inserida pelos terminais que estão ligadas às escovas com buchas. Esse mecanismo de escovas e buchas transmitem a corrente elétrica para o eixo, fazendo um simples contato elétrico no eixo. Nesse ponto, o eixo é dividido em dois magnetos (material condutor que ao encostar nas buchas recebe a corrente elétrica).

Figura 6 – Funcionamento do Motor DC de ímã permanente

Fonte: <http://goo.gl/PZb3rv>

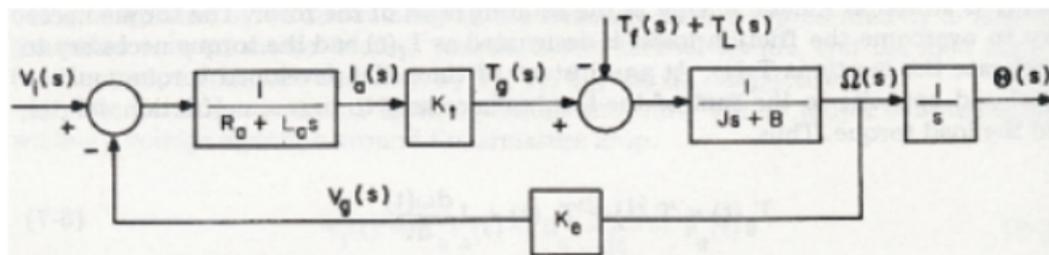
Ao receber a corrente elétrica a bobina cria um campo magnético no sentido oposto, conforme a lei de Lenz. Isso faz com que o eixo gire 180 graus de forma à alinhar com o campo magnético do estator.

No momento em que o eixo gira, os magnetos são invertidos e a bobina passa a gerar um campo magnético no sentido oposto, girando novamente mais 180 graus. E esse processo se repete indefinidamente.

2.2.2 Modelagem de um motor de corrente contínua

Na Figura 7 temos o diagrama em blocos de um Motor DC, o qual será utilizado para desenvolver a função de transferência. Este diagrama representa um sistema de duas entradas com a saída sendo a velocidade angular ω ou a posição angular θ , ou ambas (AWADALLAH; MORCOS, 2004).

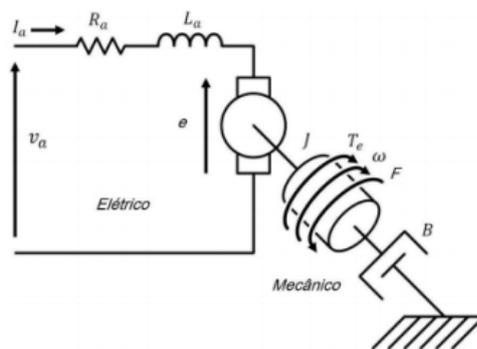
Figura 7 – Diagrama de Blocos de um Motor DC



Fonte: (AWADALLAH; MORCOS, 2004)

Motores de corrente contínua desenvolvem alto torque durante a partida e em baixas rotações, ciclo contínuo mesmo em baixas rotações, grande variação de velocidade, flexibilidade e confiabilidade. Estes motivos fazem com que sejam bastante utilizados em aplicações industriais que necessitam de uso de tecnologia de precisa (AWADALLAH; MORCOS, 2004).

Figura 8 – Sistema eletromecânico de um motor DC



Fonte: (AWADALLAH; MORCOS, 2004)

Os parâmetros elétricos e mecânicos do modelo eletromecânico ilustrado na Figura 8 estão listados na Tabela 1.

Tabela 1 – Parâmetros do Modelo Eletromecânico

Símbolo	Descrição
R_a	Resistência de armadura [Ω]
L_a	Indutância de armadura [H]
v_a	Tensão de armadura [V]
i_a	Corrente de armadura [A]
e	Tensão induzida [V]
J	Momento de Inércia [Nms^2/rad]
B	Coefficiente de atrito viscoso [Nms/rad]
F	Coefficiente de atrito estático [Nm]
ω	Velocidade Angular [rad/s]
T_e	Torque [Nm]

Sendo a velocidade de rotação diretamente proporcional à tensão induzida nas espiras,

$$e = K_e \cdot \omega(t) \quad (2.1)$$

onde K_e é definida como força contra-eletromotriz (fcem), sendo sua unidade [Vs/rad].

O torque desenvolvido no motor de corrente contínua é diretamente proporcional à intensidade da corrente que circula na armadura:

$$T_e = K_t \cdot i_a(t) \quad (2.2)$$

onde K_t é definida como constante de torque, sendo sua unidade [Nm/A].

Aplicando a segunda lei de Kirchhof no circuito elétrico ilustrado na Figura 8, e aplicando a segunda lei de Newton para os sistemas rotacionais obtém-se respectivamente:

$$v_a(t) = R_a \cdot i_a(t) + L_a \frac{d}{dt} i_a(t) + e(t) \quad (2.3)$$

$$T_e(t) = J \cdot \frac{d}{dt} \omega(t) + B \cdot \omega(t) + F \quad (2.4)$$

Utilizando o MATLAB® passamos estas equações para o domínio da frequência e aplicamos a Transformada de Laplace nas 4 equações encontradas acima, obtemos respectivamente:

$$E(s) = K_e \cdot \Omega(s) \quad (2.5)$$

$$T_e(s) = K_t \cdot I_a(s) \quad (2.6)$$

$$V_a(s) = R_a I_a(s) + s \cdot L_a \cdot I_a(s) + E(s) \quad (2.7)$$

$$T_e(s) = s \cdot J \cdot \Omega(s) + B \cdot \Omega(s) + F \quad (2.8)$$

A partir destas equações, com manipulação algébrica matemática, determina-se a função de transferência de um motor de corrente contínua, tendo a tensão de armadura, $V_a(s)$ como entrada, e como saída a velocidade angular $\Omega(s)$, apresentada na equação.

$$\frac{\Omega(s)}{V_a(s)} = \frac{\frac{K_t}{J \cdot L_a}}{s^2 + s \cdot \left(\frac{B}{J} + \frac{R_a}{L_a} \right) + \frac{B \cdot R_a + K_e \cdot K_t}{J \cdot L_a}} \quad (2.9)$$

Simplificando e isolando os termos da equação, chegamos à função de transferência *Tensão x Velocidade* $G(s)$:

$$G(s) = \frac{\Omega(s)}{V_i(s)} = \frac{K_m}{s^2 + \alpha \cdot s + \beta} \quad (2.10)$$

onde,

$$K_m = \frac{K_t}{J \cdot L_a} \quad (2.11)$$

$$\alpha = \frac{B}{J} + \frac{R_a}{L_a} \quad (2.12)$$

$$\beta = \frac{B \cdot R_a + K_e \cdot K_t}{J \cdot L_a} \quad (2.13)$$

2.3 Microcontroladores

O controle de equipamentos industriais é usualmente realizado por meio de um microprocessador embarcado no produto onde é carregado e armazenado um script de um programa composto de algoritmos dedicados à aplicação em questão. O tipo de microprocessador que melhor atende esta categoria de aplicações costuma ser aquele que integra na mesma pastilha de silício além da unidade central de processamento (CPU), também um circuito de memória e uma diversidade de circuitos auxiliares (periféricos) dedicados a funções de entrada e saída (E/S) específicas, tais como conversão analógico-digital (A/D) e saídas digitais moduladas por largura de pulso (PWM). Este tipo de processador é usualmente chamado de microcontrolador (STEPHAN, 2008).

Estes dispositivos possuem uma inteligência programável porque são dotados de uma Unidade Lógica Aritmética, na qual as operações lógicas e matemáticas são executadas. Toda essa lógica é construída na forma de um programa e arquivada dentro do componente, de modo que toda vez que o microcontrolador é alimentado o programa interno é executado.

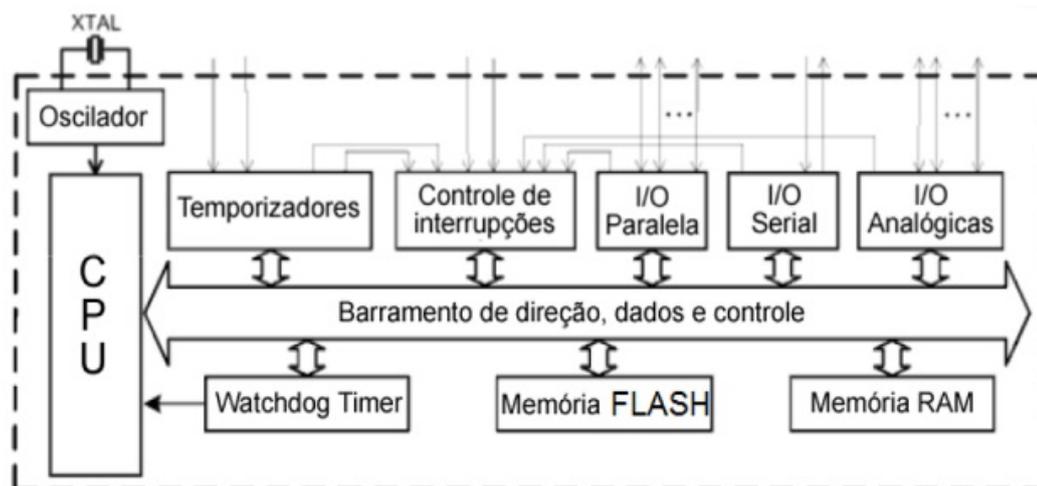
Os microcontroladores são amplamente utilizados no controle de processos, o qual deve ser entendido como o controle de periféricos como led, relés, displays, sensores, etc.

São conhecidos como controles lógicos porque a operação do sistema baseia-se nas ações lógicas que devem ser processadas, dependendo do estado dos periféricos de saída e entrada (FERREIRA, 1998).

2.3.1 Funcionamento

A Figura mostra de maneira geral o diagrama em blocos das partes que compõem um microcontrolador.

Figura 9 – Diagrama em Blocos de um Microcontrolador



Fonte: (FERREIRA, 1998)

- A CPU de um microcontrolador é responsável por processar as instruções contidas no programa e é dividida em Unidade de Controle (UC), Unidade Lógica Aritmética (ULA), Decodificador de Instrução, Registradores. Funciona como o cérebro do controlador.
- O Barramento de direção, dados e controle, permite a transferência de sinais elétricos entre diferentes partes do Microcontrolador.
- Os Registradores são responsáveis por armazenar todos os dados sobre os quais serão realizados algum tipo de operação (ADD, MOVE, SHIFT, dentre outras).
- Os microcontroladores possuem um ou mais registradores de funções especiais chamadas de PORT conectados aos seus pinos de I/O (pinos de entrada/saída).
- As interrupções compõem um sistema capaz de avisar quando uma determinada tarefa acabou, a sem prejuízo para o fluxo do programa principal.
- A memória de programa é construída no formato FLASH, que permite um número bastante elevado de ciclos de gravação utilizando-se apenas de sinais elétricos.

- O Oscilador mostrado no diagrama em blocos é responsável por gerar o clock do sistema, usado para sincronizar coisas dentro do microcontrolador. Na Figura 10 temos um sinal de clock, onde uma onda quadrada passa de 0 a 1 a uma taxa fixa. O início de cada ciclo é quando o sinal de clock passa de 0 a 1. Esse sinal é medido em uma unidade chamada Hertz (Hz), que é o número de ciclos por segundo. Um clock de 100MHz por exemplo, significa que em um segundo existem 100 milhões de ciclos de clock.

Figura 10 – Onda quadrada - Clock

Fonte: <http://goo.gl/X9lw8u>

2.4 Modulação por largura de pulso

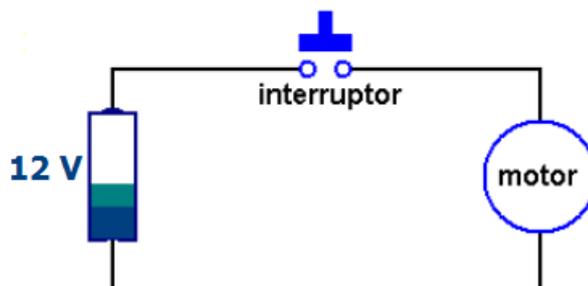
Os controles de potência, fontes chaveadas e muitos outros circuitos, utilizam um recurso tecnológico de Modulação de Largura de Pulso mais conhecida como PWM (Pulse Width Modulation) como base de seu funcionamento. Por meio do chaveamento da tensão na carga, no nosso caso o motor DC da ventoinha, obtém-se sinais com uma largura que podemos manipular e portanto controlar a velocidade de rotação do Motor DC. Logo, de modo geral, um PWM é simplesmente um circuito digital mas com a vantagem de ser utilizado apenas um Bit. Para que um circuito digital possa controlar um circuito analógico, a velocidade de um motor por exemplo, há basicamente duas tecnologias: a conversão D/A (Digital/Analógico) e a modulação por largura de pulso, sendo esta última escolhida para o desenvolvimento deste controle (BRUSSAARD; GLAUDEMANS, 1977).

Com esta técnica pode-se realizar o controle da velocidade de motores, mantendo o torque ainda que em baixas velocidades o que garante partidas suaves mesmo quando há uma carga maior sobre os motores, aspectos estes que fazem do controle PWM como ideal para aplicações na robótica (HOLMES; LIPO, 2003).

2.4.1 Princípio de funcionamento

Analisando o circuito da Figura 11, verifica-se um interruptor que quando acionado faz com que o motor receba 12V e funcione com 100% de potência. E no momento em que o interruptor não está precionado, o motor não recebe energia e portanto não funciona.

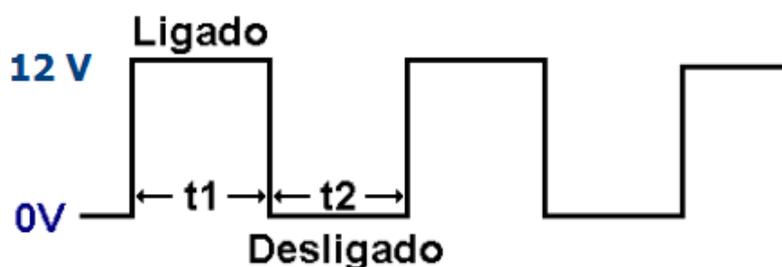
Figura 11 – Circuito Chaveado



Fonte: <http://goo.gl/3jrzHH>

Se fosse possível pressionar e soltar o interruptor um grande número de vezes por segundo, de tal forma que metade do tempo ele fica ligado e metade desligado, o resultado seria uma onda quadrada como o da Figura 12 (PIRES et al., 2012).

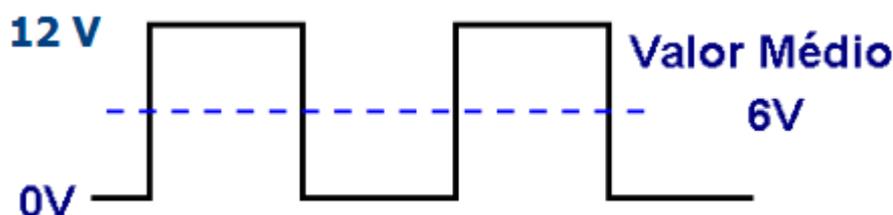
Figura 12 – Onda Quadrada



Fonte: <http://goo.gl/3jrzHH>

O tempo t_1 corresponde ao tempo que o interruptor fica pressionado e t_2 o tempo que ele fica solto. Nesta situação fictícia t_1 é igual a t_2 , durante a metade do tempo o motor recebe a tensão de 12V e na outra metade ele recebe 0V. A tensão média, Figura 13, aplicada ao motor é neste caso de 6V, ou seja, 50% da tensão da bateria (PIRES et al., 2012).

Figura 13 – Tensão Média

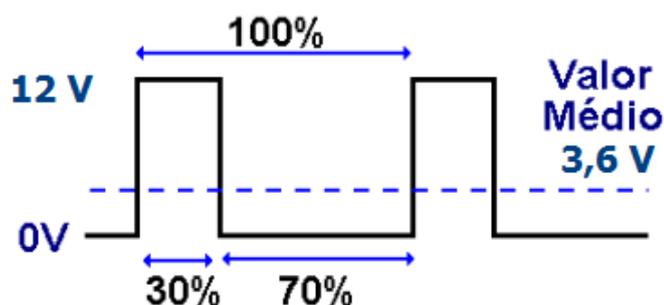


Fonte: <http://goo.gl/3jrzHH>

Obviamente não é possível usar um interruptor em um circuito com PWM, pois não conseguiríamos pressioná-lo na velocidade necessária. Por isso será utilizado para fazer

esta pulsação, um pino com saída PWM do microcontrolador arduino, por meio de um aplicativo de controle que fará a programação dos pulsos. Para diminuir a velocidade do motor, basta reduzir a largura dos pulsos, mantendo o motor menos tempo ligado, conforme Figura 14, neste exemplo o ciclo ativo é de 30% por que o tempo ativo corresponde a 30% do período da onda (PIRES et al., 2012).

Figura 14 – Controle de tensão por largura de pulso



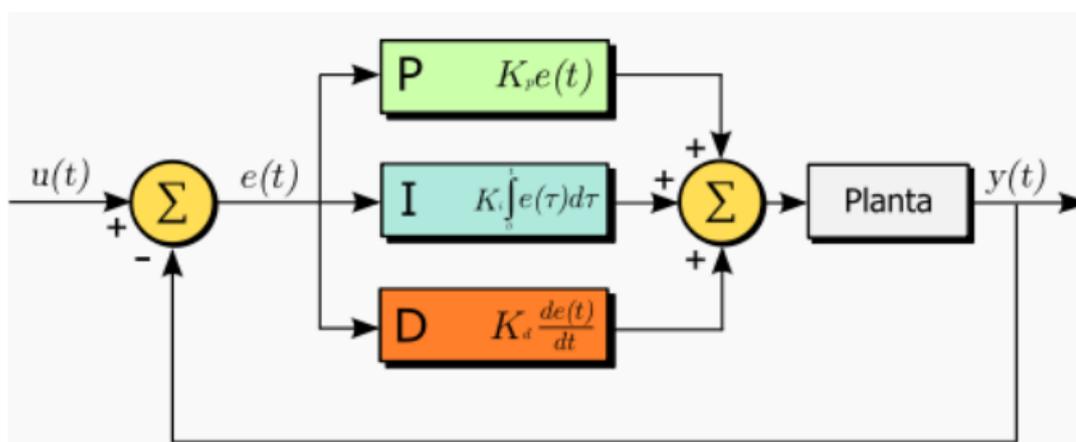
Fonte: <http://goo.gl/3jrzHH>

2.5 Controladores PID

Os controladores PID são controladores com feedback muito utilizados em automação industrial. Esses controladores calculam um erro entre o valor medido na saída e o valor desejado no processo. Assim o controlador tenta diminuir o erro que foi gerado pela saída, ajustando suas entradas (JÚNIOR et al., 2005).

O controlador PID fornece um termo proporcional, um termo integral e um termo derivativo. A partir de um valor de referência escolhido $u(t)$ pelo usuário, o controlador atuará no erro $e(t)$ de modo a compensar este erro, aproximando a resposta $y(t)$ o máximo possível da referência desejada. A Figura 15 ilustra um diagrama de blocos do controlador PID.

Figura 15 – Diagrama em Blocos do Controlador PID



Fonte: <http://goo.gl/o1OLSb>

Resumidamente, este diagrama representa as seguintes operações:

1. Um valor de Referência $u(t)$ é a entrada digitada pelo usuário no Aplicativo de Controle do Sistema.
2. O controlador então faz um cálculo verificando a diferença entre a referência $u(t)$ e a atual saída do sistema $y(t)$, que é o erro $e(t)$.
3. A partir de então, o erro $e(t)$ é multiplicado individualmente por cada um dos termos do controlador, Proporcional, Integral e Derivativo.
4. Após a Multiplicação, é feito a soma dos três termos e enfim aplicado à Planta fornecendo uma resposta compensada.

O Controlador PID atua exatamente de acordo com o erro apresentado $e(t)$. A combinação das ações proporcional, integral e derivativa gera um único sinal de controle, ao que é chamado de controlador proporcional-integral-derivativo ou simplesmente PID. A idéia é aproveitar as características particulares de cada uma destas ações a fim de se obter uma melhora significativa do comportamento transitório e em regime permanente do sistema controlado. O sinal de controle gerado pelo controlador PID é genericamente dado pela Equação 2.14 (NISE; SILVA, 2009) :

$$u(t) = MV(t) = K_p e(t) + T_i \int_0^t e(t) dt + T_d \frac{d}{dt} e(t) \quad (2.14)$$

Desta forma tem-se três parâmetros de sintonia no controlador: o ganho proporcional K_p (ação proporcional), o tempo integral T_i (ação integral) e o tempo derivativo T_d (ação derivativa).

Embora haja a disponibilidade das três ações básicas, de acordo com a aplicação não é necessário a utilização de uma ou mais destas ações. E esta mobilidade proporcionada pelo Controlador PID em modificar a sua ação foi um dos motivos pelo qual foi escolhido para ser utilizado inicialmente neste projeto.

- Basicamente temos 4 configurações possíveis de controladores a partir de uma estrutura PID:
 1. Proporcional (P)
 2. Proporcional-Integral (PI)
 3. Proporcional-Derivativo (PD)
 4. Proporcional-Integral-Derivativo (PID)

Alguns processos simples podem ser controlados de forma satisfatória apenas com a ação proporcional, não sendo este o caso deste projeto. Quanto mais é aumentado o valor de K_p , maior é a banda proporcional e menor será o valor do erro em regime permanente, entretanto este erro nunca será anulado completamente apenas com esta ação proporcional. Quanto maior a banda proporcional, mais dificilmente ocorrerá saturação de controle, ou seja, maior é a região de comportamento linear para o controlador PID (NISE; SILVA, 2009).

A ação integral está ligada diretamente à precisão do sistema sendo responsável por zerar o erro em regime permanente. Entretanto, o controlador de ação Integral causa uma desestabilização no sistema, o qual é contrabalançado pela ação derivativa que tende a aumentar a estabilidade relativa do sistema ao mesmo tempo que torna a resposta do sistema mais rápida devido ao seu efeito antecipatório (OGATA; MAYA; LEONARDI, 1998).

2.6 Sintonia

Sintonizar o Controlador, consiste em encontrar os valores de ganho de cada termo do controlador PID de modo que a resposta do sistema seja satisfatória tanto em questão de tempo como em precisão.

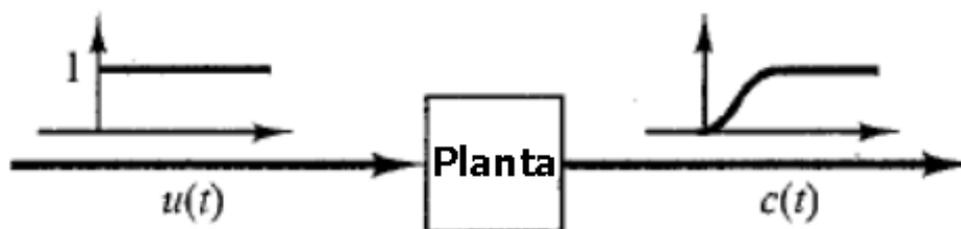
Para isso, existem diversos métodos utilizáveis descritos na literatura. Para este projeto, abordaremos alguns destes métodos.

2.6.1 Método de Ziegler e Nichols - resposta ao degrau

O primeiro método obtém experimentalmente a resposta da planta a uma entrada em degrau unitário, como ilustrado na Figura 14. Se a planta não possui integradores nem pólos complexos conjugados dominantes, então essa curva de resposta ao degrau unitário pode ter o aspecto de um S em seu corpo, como mostrado na Figura 16. Esse método se aplica quando à curva de resposta ao degrau de entrada tiver a forma de um S em seu corpo. Entretanto, segundo Nise e Silva (2009) que realizaram estudos aprofundados sobre este método, pode-se utilizar o método de Ziegler e Nichols para sintonizar controladores PID de funções de transferência de segunda ordem quando o termo de segunda ordem é significativamente pequeno, por isso será utilizado neste projeto.

A curva de resposta ao degrau pode ser gerada experimentalmente ou a partir de uma simulação dinâmica da planta.

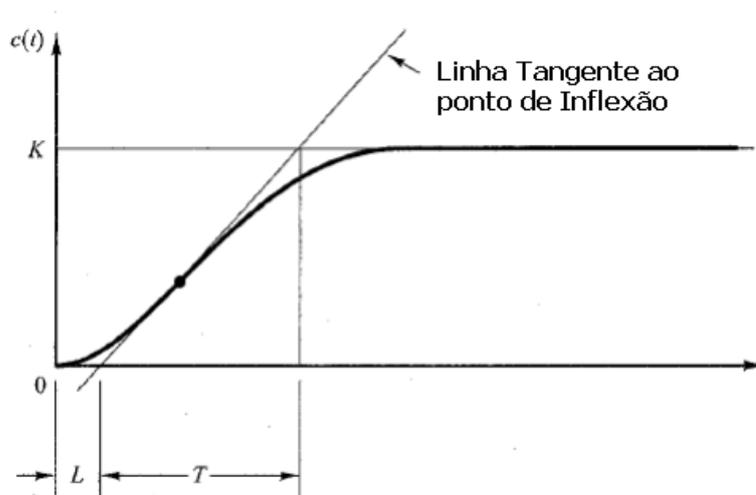
Figura 16 – Resposta ao degrau unitário de uma planta



Fonte: (NISE; SILVA, 2009)

A curva em forma de S pode ser caracterizada por meio de duas constantes, a constante tempo T e o atraso L . A constante de tempo e o atraso são determinados desenhando-se uma linha tangente no ponto de inflexão da curva com formato de S e determinando-se a interseção da linha tangente com o eixo dos tempos e linha $c(t) = K$, como mostrado em Figura 17.

Figura 17 – Curva com S em seu corpo



Fonte: (NISE; SILVA, 2009)

A função de transferência $C(s)/U(s)$ pode ser aproximada por um sistema de primeira ordem com um atraso de transporte, segundo a Equação 2.15.

$$\frac{C(s)}{U(s)} = \frac{K \cdot e^{-L \cdot s}}{Ts + 1} \quad (2.15)$$

Tendo as constantes encontradas, Ziegler e Nichols propuseram, em primeiro critério, fixar os valores de K_p , T_i e T_d de acordo com a fórmula mostrada na Tabela 2.

Tabela 2 – Regra 1 de sintonia de Ziegler e Nichols

Tipo de Controle	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9 \cdot \frac{T}{L}$	$\frac{T}{0,3}$	0
PID	$1,2 \cdot \frac{T}{L}$	$2L$	$0,5 \cdot L$

Fonte: (NISE; SILVA, 2009)

2.6.2 Método de Cohen-Coon

Segundo Bishop (2013) uma das desvantagens do método de Ziegler-Nichols é que em malha fechada tende a produzir respostas mais oscilatórias.

Para apresentar uma melhor resposta em malha fechada, Cohen-Coon sugeriram um grupo de equações similares ao método de Ziegler-Nichols, ilustrado na Tabela 3.

Tabela 3 – Regra de sintonia de Cohen e Coon

Tipo de Controle	K_p	T_i	T_d
P	$\frac{T}{K \cdot L} \cdot \left(1 + \frac{L}{3T}\right)$	-	-
PI	$\frac{T}{K \cdot L} \cdot \left(0,9 + \frac{L}{12T}\right)$	$\frac{\left(30 + \frac{L}{12T}\right)}{\left(0,9 + \frac{L}{12T}\right)}$	-
PID	$\frac{\frac{L}{4T} + \frac{4}{3}}{K \cdot \frac{L}{T}}$	$L \frac{\frac{3L}{4T} + 4}{\frac{L}{T} + \frac{13}{8}}$	$L \frac{2}{\frac{L}{T} + \frac{11}{2}}$

Fonte: (NISE; SILVA, 2009)

Em teoria, este método se aplica a processos que possuem tempo morto elevado, que não é o caso deste projeto. Entretanto, para efeito de estudos, iremos aplicar este método e verificar seu desempenho na planta deste projeto. O método de Cohen-Coon apresenta um desempenho razoável para valores do fator de incontroleabilidade do processo (ϑ/τ) entre 0,4 e 4,5.

2.6.3 Método AMIGO

Em Åström, Panagopoulos e Häggglund (1998) é descrito um método de sintonia de controladores PID tendo como base técnicas matemáticas de otimização tendo o mesmo objetivo que Ziegler e Nichols, ou seja desejava criar uma regra de sintonia geral para vários tipos de processos.

O M-constrained Integral Gain Optimization (MIGO) é uma abordagem computacional que utiliza a maximização do ganho integrativo do controlador como critério de otimização, limitado pela máxima sensibilidade do controlador M_s , definida como o máximo da

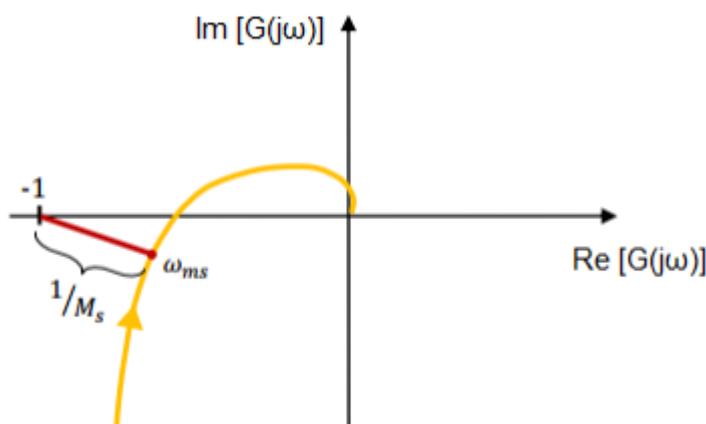
função sensibilidade conforme o aumento da frequência ω definida para um sistema com controlador $G_C(s)$ e planta $G(s)$. Matematicamente, M_s é representado na Equação 2.16.

$$M_s = \max \left| \frac{1}{1+G_C(j\omega)G(j\omega)} \right|_{\omega} \quad (2.16)$$

Maximizar o ganho integrativo nada mais é do que encontrar uma resposta satisfatória no menor intervalo de tempo e também garantir maior estabilidade da resposta independente de distúrbios incidentes no processo.

A máxima sensibilidade é tida como sendo o inverso da menor distância entre a curva de Nyquist da função de transferência $G_C(j\omega)G(j\omega)$ e o ponto crítico de instabilidade -1, ou seja, o ponto da curva de Nyquist mais fácil de ser instabilizado. Esta interpretação gráfica é exibida na Figura 18.

Figura 18 – Curva de Nyquist para $G_C(j\omega)G(j\omega)$



Fonte: (CARDOSO, 2002)

Valores tradicionais de M_s variam entre 2 e 1.2, sendo 2 para um sistema mais sensível, no que diz respeito a incertezas, ou seja a qualidade do modelo usado, e 1.2 para um sistema mais robusto no sentido de que variações paramétricas da planta não causarão mudanças na saída $y(t)$.

Resumidamente, o parâmetro M_s possui relação com duas medidas tradicionais de estabilidade, margem de ganho g_m e margem de fase φ_m , sendo dada por:

$$g_m > \frac{M_s}{M_s - 1} \quad (2.17)$$

$$\varphi_m > 2 \cdot \sin^{-1} \frac{1}{2M_s} \quad (2.18)$$

Tendo este método como princípio, Åström e Hägglund (2004) desenvolveram uma implementação deste método em vários experimentos para diferentes processos industriais, em que os parâmetros foram variados para obter um número considerável de condições diferentes para o teste. A partir dos resultados obtidos, foi realizada uma tentativa de

equacionar essas variações dos parâmetros obtidos em cada condição, o que originou na regra de sintonia aproximada denominada AMIGO (Approximate MIGO).

A sintonia AMIGO obtida para o PID da Equação 2.14 com uma sensibilidade máxima de $M_s = 1,4$ é ilustrada na Tabela 4.

Tabela 4 – Regra de sintonia por método AMIGO

Tipo de Controle	K_p	K_i	K_d
PID	$\frac{1}{K} \cdot \left(0.2 + 0.45 \frac{T}{L}\right)$	$L \cdot \frac{0.4L + 0.8T}{L + 0.1T}$	$\frac{0.5 \cdot L \cdot T}{T + 0.3L}$

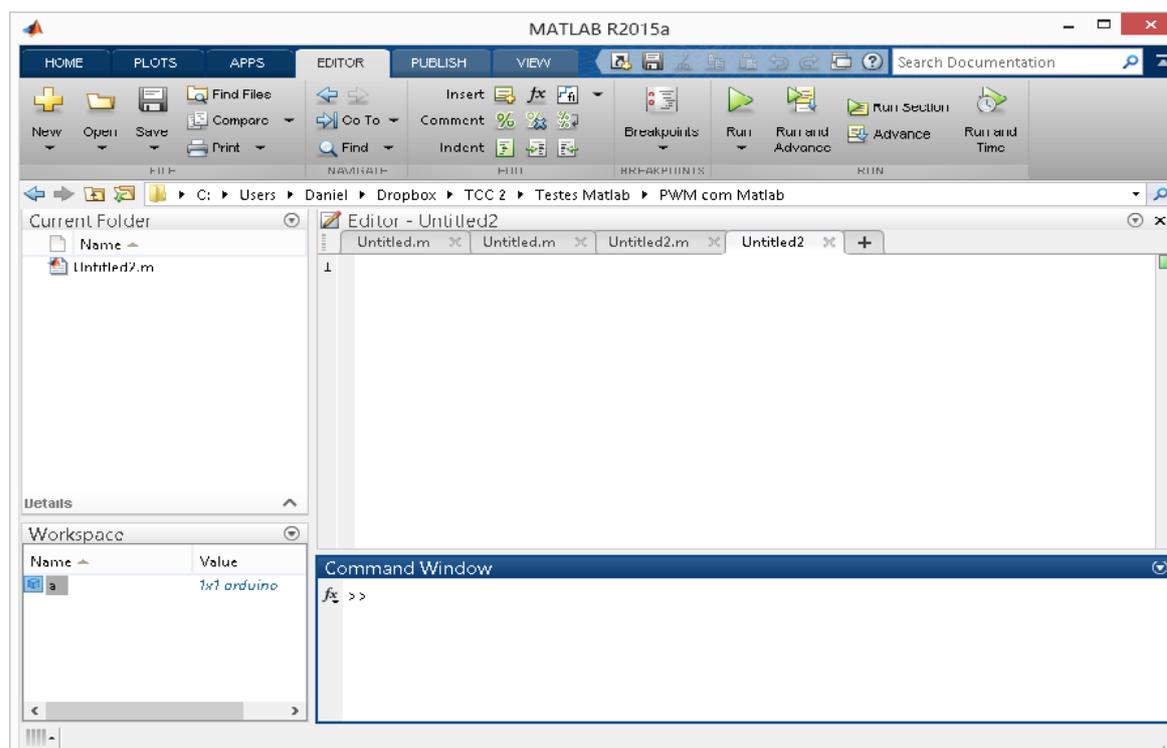
Fonte: (ÅSTRÖM; HÄGGLUND, 2004)

2.7 MATLAB®

MATLAB® é uma linguagem de alto nível e um ambiente interativo de computação numérica, visualização e programação. Com os recursos do MATLAB® pode ser feita análise de dados, desenvolvimento de algoritmos, criação de modelos e aplicações. A linguagem, as ferramentas e funções criadas no próprio *software* permitem a exploração de múltiplos recursos e solução de problemas em tempo menor se comparado com o uso de linguagem de programação tradicionais. É um software proprietário, mas permite a aquisição de uma licença estudantil fornecida no site do fabricante (MATHWORKS, 2015).

A Figura 19 mostra a tela principal do programa.

Figura 19 – Tela Principal do MATLAB®

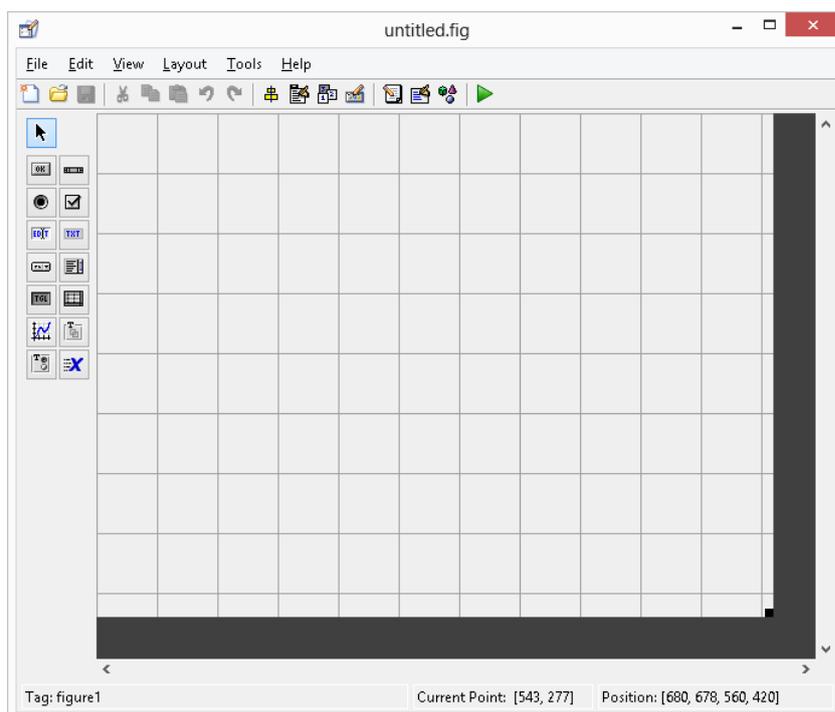


Fonte: (MATHWORKS, 2015)

Dentro de sua plataforma, o *Guide* é um ambiente de desenvolvimento gráfico, no qual pode ser criada uma interface de visualização e comandos de forma a tornar mais amigável a execução de rotinas ou até mesmo desenvolver um aplicativo onde é possível aplicar funções de controle e controladores digitais. Isso, porque o *Guide* é integrado ao MATLAB®, permitindo incorporar algoritmos nas funções dos objetos, e também exportar dados para outros componentes do MATLAB® para uso em outras análises.

A Figura mostra a IDE do componente *Guide*.

Figura 20 – IDE do *Guide* do MATLAB®



Fonte: (MATHWORKS, 2015)

Por ser um ambiente muito utilizado tanto no meio acadêmico como no industrial, sua utilização é fundamental tanto pelos recursos disponíveis quanto pela compatibilidade com outros projetos.

O MATLAB® foi escolhido na utilização deste projeto pela sua grande variedade de aplicações de controle e sintonia de controladores, bem como a possibilidade também de criar um aplicativo no seu componente *Guide*.

Outro fator importante e fundamental para a escolha foi o fato deste software permitir a comunicação com o microcontrolador Arduino UNO por meio tanto da porta serial, como também por uma biblioteca específica de comunicação com o arduino disponibilizada no próprio site do fabricante.

3 TRABALHOS RELACIONADOS

Após pesquisas de campo e literária, observou-se que na área educacional tratando-se de sistemas práticos de controle moderno, ainda não existem muitos trabalhos objetivando este fim. E mais especificamente, a respeito de um levitador pneumático, não foi encontrado nenhum sistema real como tal, para que fosse explanado neste capítulo.

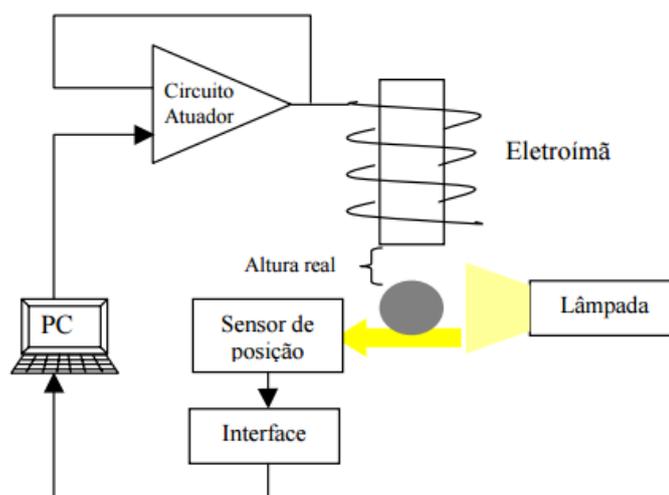
Entretando, neste artigo, Santana, Ferreira e Salazar (2001) propõe a montagem e construção de um levitador magnético pra ser utilizado como módulo educacional, de modo que, um eletroímã é responsável pela atração de um corpo, tentando compensar o efeito da força gravitacional. É feito o controle digital dos níveis de tensão nos terminais da bobina baseando-se em parâmetros medidos por circuitos específicos auxiliares, o que inclui um sensor de posição. O controle é implementado digitalmente por um computador pessoal. Devido a suas características e envolvimento de conhecimentos de diversas áreas, este sistema se apresenta como um módulo a ser utilizado pela especialidades de engenharia, tanto mecânica, como elétrica e também de controle e automação. O trabalho mostra as diferentes partes do sistema mostrando as alternativas de projeto. Mostra a características de estabilidade da planta e sugere uma estratégia de controle para conseguir a estabilidade do mesmo.

3.1 Descrição do sistema

O posicionador eletromagnético ativo proposto, é uma consequência da atuação de campos eletromagnéticos que, em decorrência de informações a respeito da posição relativa do corpo a ser posicionado, agem no sentido de reorganizar o sistema, gerando forças capazes de reposicionar e de mantê-lo em sua devida posição. Um sensor efetua a medida da posição, essa medida é levada ao controlador que analisa e gera um sinal de correção, passando por um amplificador de potência, o qual alimentará a bobina, resultando em um aumento ou diminuição da força necessária para a reorganização do sistema.

O sistema de levitação magnética tem como esquema funcional o diagrama de blocos apresentado na Figura 21. O corpo que irá levitar é uma bola de isopor revestida com pó de ferro, isso faz com que seu peso não exija grandes níveis de força, logo, grandes valores de corrente. Um eletroímã é responsável pela levitação do corpo. Um sensor de posição é utilizado para leitura da distância da bola até o eletroímã. Essa leitura serve como parâmetro de entrada para o controlador o qual fornece um sinal proporcional ao deslocamento necessário ao reposicionamento. Esse sinal é enviado ao circuito atuador que é responsável pela potência adequada no eletroímã. Fontes de alimentação e circuitos auxiliares, serão responsáveis pelos níveis de sinais satisfatórios e pelas normalizações necessárias.

Figura 21 – Diagrama em Blocos do Módulo



Fonte: (SANTANA; FERREIRA; SALAZAR, 2001)

3.1.1 Amplificação

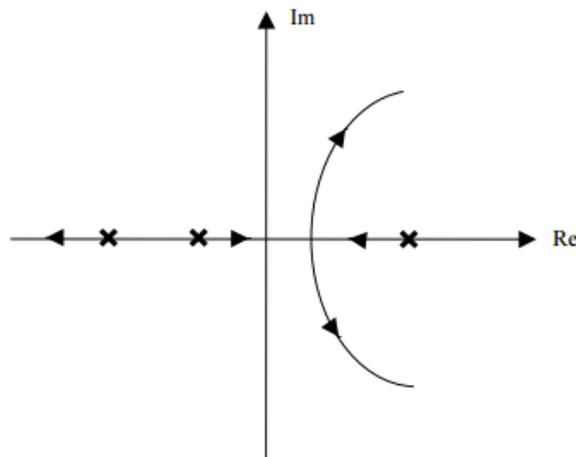
Uma das etapas do processo é a de amplificação de corrente, que na Figura 21 está inclusa no bloco circuito atuador. A amplificação se faz necessária uma vez que o nível do sinal proveniente do PC não apresenta potência suficiente para ativar o eletroímã.

A forma de amplificação utilizada neste projeto, que atua em conjunto com o controle digital, consiste na variação do valor médio de corrente fornecida à bobina. Neste caso, o transistor que age como regulador de corrente funciona como chave, hora permitindo que a bobina receba uma tensão nos seus terminais, hora cortando esse fornecimento, fazendo com que o valor eficaz de corrente no eletroímã varie de acordo com o ciclo de trabalho da chave. Como o atuador atinge frequências de aproximadamente 5 kHz, praticamente não há perdas de potência por dissipação na chave. Assim, é possível a utilização de núcleos de ferrite e transistores menores que produzam os mesmos efeitos do sistema de atuação linear (SANTANA; FERREIRA; SALAZAR, 2001).

3.1.2 Controle

O controle de um sistema de levitação magnética tem como principal finalidade manter a corrente de entrada em torno de um ponto de equilíbrio. No entanto, para se conseguir tal equilíbrio, é necessário que se trabalhe com as equações dinâmicas do sistema, que são não lineares. Porém são utilizados artifícios no intuito de se conseguir aproximações lineares para facilitar o controle. Para o sistema estudado, tem-se o Root Locus como mostrado na Figura 22

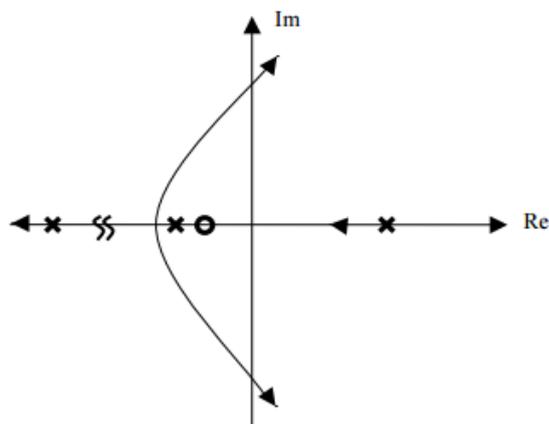
Figura 22 – Root Locus - Sem controlador



Fonte: (SANTANA; FERREIRA; SALAZAR, 2001)

A Figura 22 indica que o sistema pode não chegar a estabilidade apenas com um controlador proporcional, já que o mesmo apresenta um pólo no semiplano direito. Para resolver este problema é utilizado um controlador em atraso de fase o que se adequa perfeitamente ao sistema em questão. Após as devidas considerações, com a inserção do controlador em atraso, o Root Locus do sistema apresenta-se como mostrado na Figura 23.

Figura 23 – Root Locus - Com controlador



Fonte: (SANTANA; FERREIRA; SALAZAR, 2001)

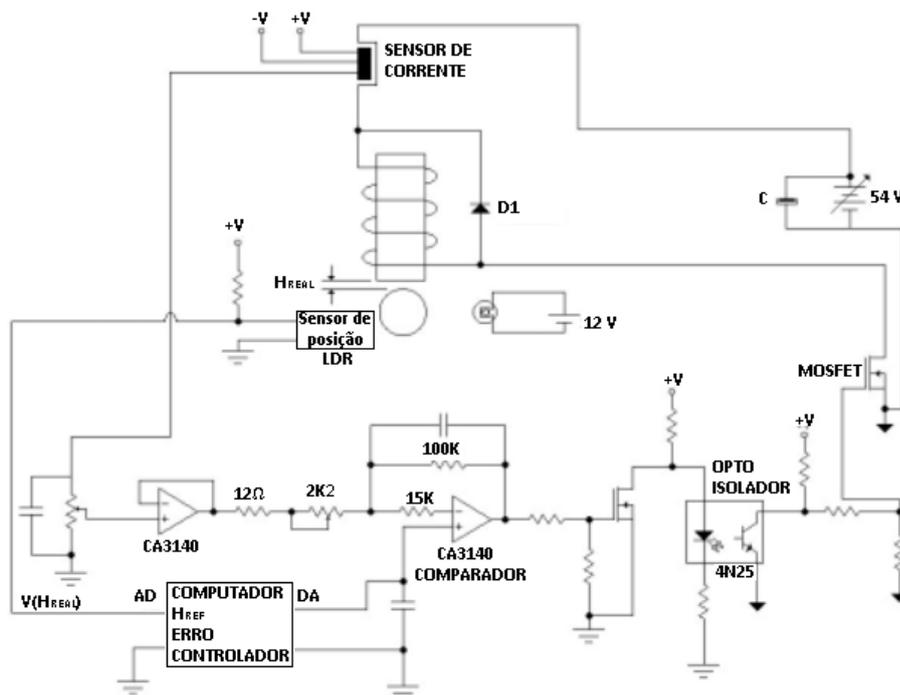
3.1.3 Controlador digital

A próxima etapa foi a implementação de um controlador digital. Um controlador proporcional derivativo (PD) foi suficiente para atender satisfatoriamente às necessidades do levitador magnético. O controle digital apresenta diversas vantagens sobre o controle analógico, dentre elas a possibilidade de testar estratégias de controle variadas em todas as etapas do projeto e do uso de funções complexas, facilidade de calibração de sensores, *offsets*,

testes de calibração e monitoramento *on line* do sistema. De forma geral, possibilidade de mudanças no sistema sem grande prejuízo de tempo e equipamento.

A Figura 24 mostra em detalhes o circuito de sistema de controle do levitador.

Figura 24 – Sistema de levitação com controle digital e atuador chaveado



Fonte: (SANTANA; FERREIRA; SALAZAR, 2001)

3.1.4 Sistema eletromecânico

O eletroímã é constituído por uma bobina com 1000 voltas de fio esmaltado, com uma resistência de 9,5 ohms e capacidade para 1,5 Ampères. O núcleo utilizado na bobina é de material ferromagnético, laminado para o controle analógico e de ferrite para o controle digital. O corpo a ser levantado pesa aproximadamente 0,035 kg e tem 0,054 m de diâmetro. Como atuador, foi utilizado um transistor TJB no caso analógico, e um transistor Mosfet no caso digital.

3.1.5 Sensor de posição

A medida de corrente da bobina foi feita por meio de um sensor de efeito hall. A posição da bola é obtida mediante um sensor óptico o qual fornece uma resistência proporcional à intensidade luminosa em sua superfície. Portanto, alocando de forma retilínea uma lâmpada, a bola e o sensor, tem-se uma variação da incidência de luz no sensor, proporcional à variação da posição da bola. O tipo de sensor mais indicado para esta aplicação é o sensor *Eddy Current*, por apresentar uma precisão maior, porém por motivo de disponibilidade, um sensor mais acessível foi utilizado, conseguindo o mesmo propósito.

3.1.6 Sistema de controle

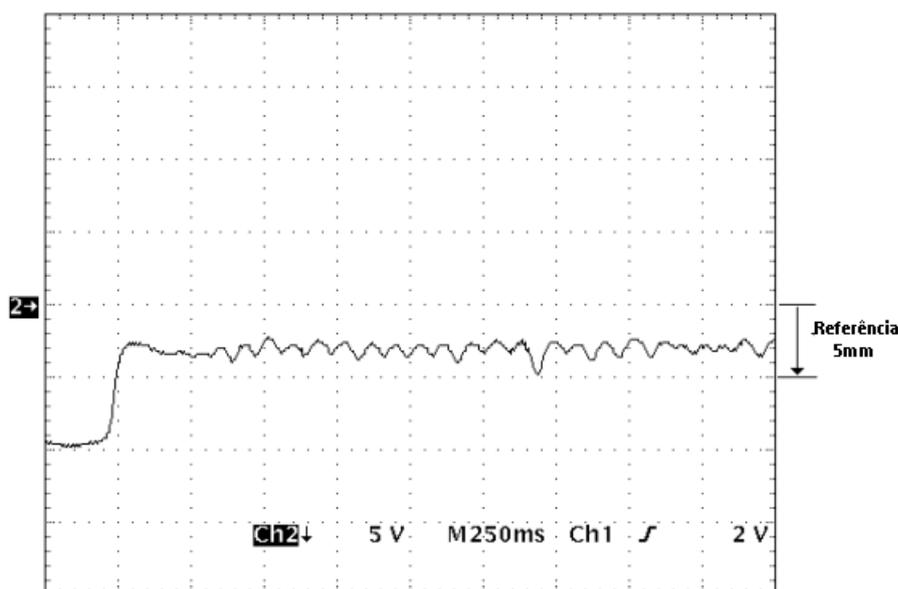
Para o controle via Computador Pessoal foi utilizado um controlador PD programado em linguagem C. Para interface é utilizada uma placa AD/DA modelo PCL 818, responsável pela aquisição e envio de dados ao amplificador de corrente. O controle, na realidade, fornece um sinal responsável pelo acionamento do Mosfet que habilitará ou não o sinal de potência disponibilizado à bobina.

O controle digital proporciona uma maior flexibilidade no controle, uma vez que apresenta uma maior variação de ajuste dos parâmetros, e até mesmo uma variação de controlador, tornando possível a comparação entre diferentes tipos de controladores.

3.2 Resultados

O objetivo inicial foi alcançado com sucesso. O levitador funcionou como proposto. Porém, devido a frequência de chaveamento em torno de 5 KHz e ao fluxo de dispersão do núcleo foi registrado o aparecimento de sinais de interferência eletromagnética, o que podem ser corrigidos com blindagem dos cabos e isolamento do núcleo. O desempenho do sistema pode ser observado na Figura 25. Ela mostra a variação da posição do corpo em relação a uma referência dada pelo núcleo da bobina.

Figura 25 – Posição relativa do corpo em relação ao núcleo



Fonte:(SANTANA; FERREIRA; SALAZAR, 2001)

3.3 Comparativo

Pode-se observar que neste trabalho que o autor procura utilizar um método de controle moderno para alcançar a estabilização do sistema. Neste caso, para a análise do sistema,

foi utilizado o diagrama do lugar geométrico das raízes, onde foi possível concluir que um controlador proporcional derivativo seria necessário e suficiente para satisfazer a estabilização do sistema de forma satisfatória.

Este controlador foi implementado por meio de um circuito digital, como demonstrado na Figura 24, trazendo um bom resultado para o sistema como observado na Figura 25.

Do mesmo modo que este trabalho, nesta dissertação, buscou-se construir um levitador, entretanto pneumático, mas também com intuito de utilização educacional.

4 MATERIAIS E MÉTODOS

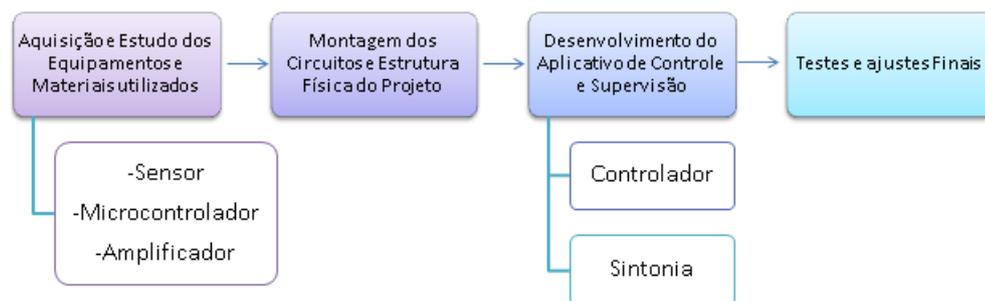
Construir um levitador pneumático é uma tarefa difícil uma vez que trabalhar com vazão de fluxo de ar é algo extremamente sensível. Um valor pequeno de variação na tensão ou na corrente é suficiente para causar um impacto significativo na velocidade de rotação da ventuinha e conseqüentemente na variação da posição de estabilização do objeto levitado (TURNS, 2013).

Por esta razão é fundamental e imprescindível que os equipamentos e dispositivos utilizados na construção do levitador sejam escolhidos considerando todos os parâmetros mínimos de precisão e confiabilidade, para que se possa chegar a um resultado satisfatório ao final da montagem do levitador.

Serão abordados neste capítulo, de forma analítica, a metodologia utilizada para a interligação de todos os elementos e dispositivos utilizados na montagem do projeto. Após a aquisição e a garantia de funcionamento de todos os componentes de forma independente, será feita a montagem do sistema físico com o acoplamento dos dispositivos e o desenvolvimento do aplicativo de controle para realizar a interface entre o usuário e o sistema.

Na Figura 26 tem-se um diagrama em blocos mostrando todas as etapas da execução do projeto.

Figura 26 – Diagrama em Blocos da Metodologia



Fonte: Autor

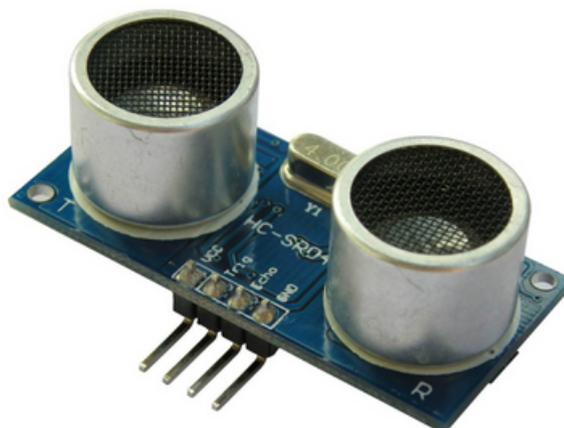
4.1 Sensor ultrasônico HC-SR04

Para a escolha do sensor de distância utilizado para medir a posição em que o objeto levitado estará, foi levado em consideração a capacidade de leitura de distância que deve ser de no mínimo 60 cm que será a altura do levitador, a precisão de medição que deve ser aceitável até 3 milímetros no máximo para uma boa referência com relação à altura total, e o custo.

O sensor de Posição encontrado no mercado que atende a todos os requisitos mencionados anteriormente é o Sensor Ultrassônico HC-SR04, capaz de medir distâncias de 2cm a 4m com ótima precisão. Este módulo possui um circuito pronto com emissor e receptor acoplados e 4 pinos (VCC, Trigger, ECHO, GND) para medição.

A Figura 27 mostra o sensor utilizado.

Figura 27 – Sensor Ultrassônico HC-SR04



Fonte: <http://goo.gl/meCbQo>

A Tabela 5 mostra as especificações do sensor segundo o fabricante, (DATASHEET, 2011):

Tabela 5 – Especificação do Sensor HC-SR04

Variável	Especificação
Distância de Operação	2 a 400 cm
Precisão	2 mm
Tensão de Alimentação	5 V
Corrente de Operação	2mA
Ângulo de percepção	15°
Modo de Conexão	VCC / Trig(T) / Echo(R) / GND

Fonte: (DATASHEET, 2011)

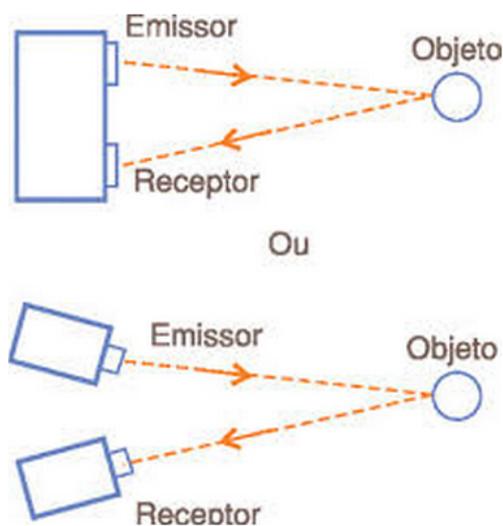
4.1.1 Funcionamento

O princípio de operação desses sensores é exatamente o mesmo do sonar, usado pelo morcego para detectar objetos e presas em seu vôo cego. O pequeno comprimento de onda das vibrações ultra-sônicas faz com que elas se reflitam em pequenos objetos, podendo ser captadas por um sensor colocado em posição apropriada (THOMAZINI; ALBUQUERQUE, 2005).

Na prática um sensor ultrassônico é formado por um emissor e um receptor, tanto fixados num mesmo conjunto como separados, dependendo do posicionamento relativo desejado (THOMAZINI; ALBUQUERQUE, 2005).

Na Figura 28 podemos ver o esquemático de seu funcionamento.

Figura 28 – Funcionamento do Sensor Ultrassônico



Fonte: <http://goo.gl/ID43im>

Para começar a medição é necessário alimentar o módulo e colocar o pino Trigger em nível alto por mais de 10us. Assim o sensor emitirá uma onda sonora que ao encontrar um obstáculo rebaterá de volta em direção ao módulo, sendo que neste tempo de emissão e recebimento do sinal o pino ECHO ficará em nível alto. Logo o cálculo da distância pode ser feito de acordo com o tempo em que o pino ECHO permaneceu em nível alto após o pino Trigger ter sido colocado em nível alto. A Fórmula 4.1 mostra matematicamente como obter a distância em função do tempo que a onda sonora demora pra ir no objeto e voltar.

$$Distância = [TempoECHOemNívelAlto * VelocidadeSom]/2 \quad (4.1)$$

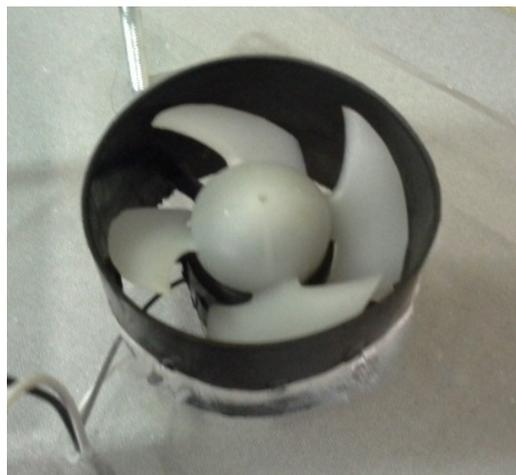
Para efeito deste projeto, podemos considerar a velocidade do som sendo invariavelmente 340 m/s. Com estas unidades, o resultado é obtido em metros se considerado o tempo em segundos, que posteriormente é convertida para milímetros fazendo a multiplicação por 1000.

A divisão por 2 descrita na fórmula deve-se ao fato de que a onda emitida pelo sonar percorre o caminho 2 vezes, isto é, para ir, e para retornar, logo, a distância real entre o sonar e o objeto é a metade do resultado do cálculo.

4.2 A ventoinha utilizada

Será utilizada como fonte geradora de fluxo de ar uma ventoinha elétrica KENLOWE 2,3". Esta Ventoinha é apropriada visto que trabalha com níveis de tensão de 0 a 12V e com uma rotação de até 300 rad/s, suficiente para atender a necessidade de vazão de ar para levantar o objeto utilizado no projeto. Na Figura 29 podemos visualizar o modelo.

Figura 29 – Ventoinha KENLOWE 2,3"



Fonte: Autor

Segundo Such (2009), que elaborou um artigo utilizando esta ventoinha para um sistema de refrigeração, este modelo de ventoinha poderia ser encontrado dentro de um Secador de Cabelo específico. Foi então adquirido este equipamento e retirado a ventoinha para posterior utilização.

A Tabela 6 mostra as especificações dadas segundo o Manual do fabricante (KENLOWE, 2010):

Tabela 6 – Especificação do Motor DC

Variável	Especificação
Tensão Máxima	$V_{i.(MAX)} = 12 \text{ V}$
Corrente Máxima	$I_{a.(MAX)} = 2 \text{ A}$
Resistência da Armadura	$R_a = 3,13 \cdot 10^{-1} \text{ Ohms}$
Indutância da Armadura	$L_a = 3 \text{ milihenries}$
Constante de Torque	$K_t = 6,78 \cdot 10^{-5} \text{ Vs/rad}$
Constante de Força Contraeletromotriz	$K_e = 6,78 \cdot 10^{-5} \text{ Vs/rad}$
Momento de Inércia	$J = 1,918 \times 10^{-4} \text{ Nms}^2/\text{rad}$
Coefficiente de Atrito Viscoso	$B = 1,12387 \times 10^{-6} \text{ Nms/rad}$
Velocidade Angular	$\omega_{NOMINAL} = 300 \text{ radianos/s}$

Fonte: (KENLOWE, 2010)

Utilizaremos estes parâmetros de especificação para desenvolver a modelagem do motor DC utilizado neste projeto.

4.2.1 Função de transferência do motor DC

A modelagem do sistema, para efeito de controle, resumiu-se em modelar o motor DC da ventoinha, uma vez que é o responsável direto pelo deslocamento do corpo e possui uma relação linear conforme será visto na próxima seção.

Tendo a Equação 2.10 desenvolvida, é necessário apenas que seja aplicado os valores de especificação fornecidos no manual do fabricante já citados anteriormente para que seja encontrada a função de transferência do motor DC utilizado neste projeto.

Realizando estas operações chegamos ao seguinte resultado:

$$Km = \frac{6,78 \cdot 10^{-5}}{1,918 \cdot 10^{-4} \cdot 3} = 117,8 \cdot 10^{-3} \quad (4.2)$$

$$\alpha = \frac{1,12387 \cdot 10^{-6}}{1,918 \cdot 10^{-4}} + \frac{0,313}{3} = 1,102 \cdot 10^{-7} \quad (4.3)$$

$$\beta = \frac{1,12387 \cdot 10^{-6} \cdot 0,313 + 6,78 \cdot 10^{-5} \cdot 6,78 \cdot 10^{-5}}{1,928 \cdot 10^{-3} \cdot 3} = 5,986 \cdot 10^{-5} \quad (4.4)$$

$$G(s) = \frac{\Omega(s)}{V_i(s)} = \frac{117,8 \cdot 10^{-3}}{1,102 \cdot 10^{-7} \cdot s^2 + 5,986 \cdot 10^{-5} \cdot s + 4,599 \cdot 10^{-3}} \quad (4.5)$$

Segundo Holmes e Lipo (2003), os distúrbios externos que se aplicam num motor de corrente contínua, tem grande influência quando a questão importante é o torque ou quando o motor trabalha com velocidades baixas, entretanto, quando a sua velocidade angular é acima de 200 rpm, ou quando o torque não é relevante para o projeto, os distúrbios podem ser desprezados, e quanto mais se aumenta a velocidade, mais insignificante o distúrbio se torna, uma vez que as variações são pontuais e em curta duração.

Portanto, para este projeto, os distúrbios referentes a variações de corrente e tensão podem ser desprezados visto que trabalharemos com rotações bastante acima de 200 rpm.

Podemos então considerar a função de transferência final de Tensão x Velocidade como sendo apenas a $G(s)$. desprezando os distúrbios de tensão e corrente.

4.2.2 Relação velocidade X altura

A Função de Transferência obtida na seção anterior nos traz a relação entre a tensão aplicada com a velocidade angular do motor da ventoinha. Entretanto, é preciso estabelecer uma relação entre a velocidade angular e a altura atingida. Para isso, foram aplicados alguns valores de tensão no motor da ventoinha para verificar a altura atingida pelo corpo levitado, dentre os quais ilustramos na Figura 30 o menor e o maior valor de tensão aplicado nos testes.

Figura 30 – Testes de Tensão x Altura



Fonte: Autor

Os testes realizados aplicações de respostas ao degrau de alguns valores mostrados na Tabela 7.

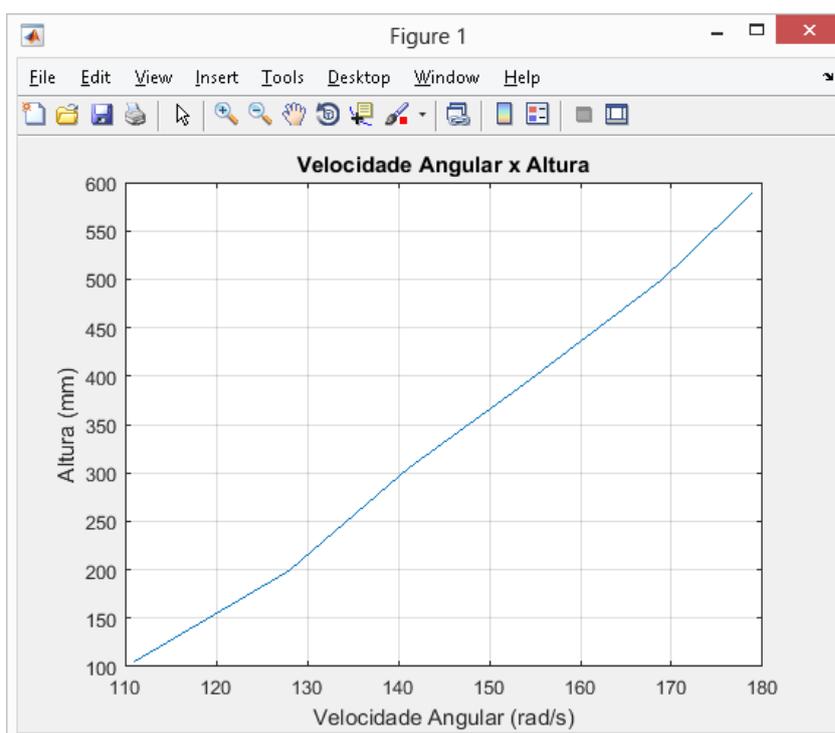
Tabela 7 – Testes de Respostas ao Degrau

Tensão(V)	Velocidade Angular(rad/s)	Altura Obtida(mm)
4,32	111	105
5	128	199
5,5	141	304
6	154	393
6,5	169	500
6,98	179	590

Fonte: Autor

De posse destes valores encontrados nos testes, foi plotado no MATLAB® o gráfico alinhando os pontos da altura obtida em função da velocidade angular encontrada para cada valor de tensão aplicado na função de transferência. E foi verificado que, apesar das pequenas variações de linearidade, a relação entre estas variáveis, segue de fato um correspondência linear, como é possível observar no gráfico ilustrado na Figura 31.

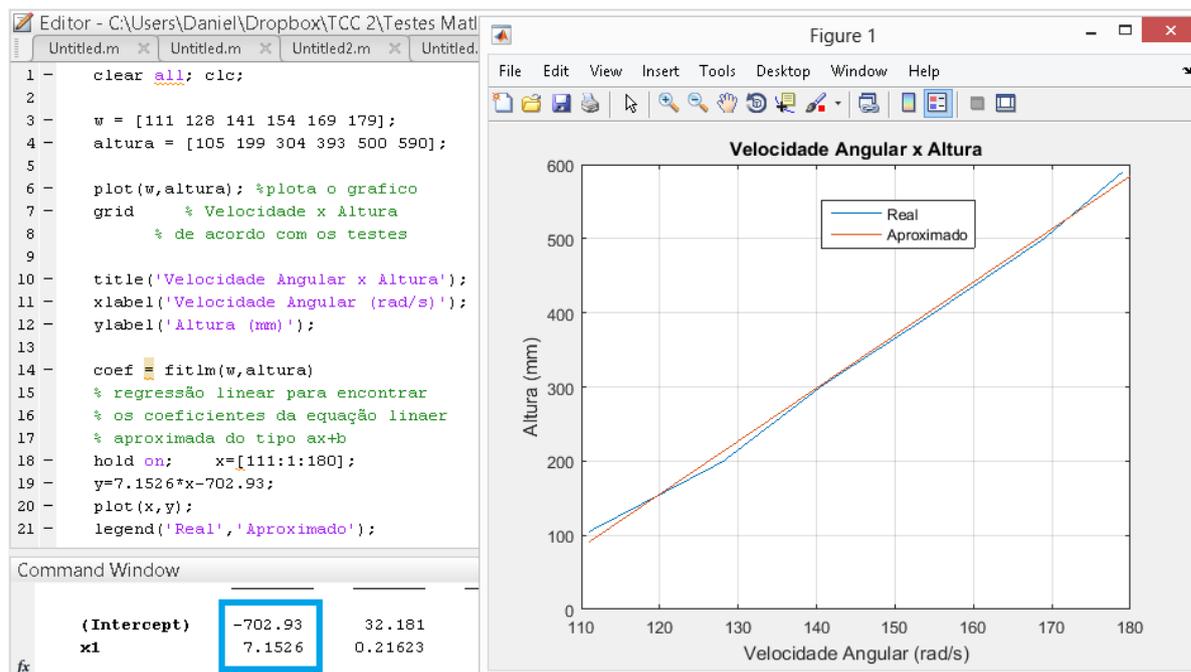
Figura 31 – Gráfico Velocidade x Altura



Fonte: Autor

O Gráfico plota a interligação dos pontos encontrados a partir da Tabela 7. É possível observar que o gráfico não apresenta uma linearidade perfeita como seria desejado e como era previsto após estudos. Por este motivo, foi utilizado a técnica de regressão linear para aproximar os pontos do gráfico a uma função média entre os pontos. Isso foi feito utilizando um comando no MATLAB® exibido na Figura 32, onde é possível ver o script completo da técnica de regressão linear e seu resultado.

Figura 32 – Função linear aproximada



Fonte: Autor

A Figura 32 ilustra a plotagem dos pontos de experimentos reais (linha azul) e a função aproximada obtida após a aplicação da regressão linear (linha vermelha)..

Com a regressão linear obtivemos os coeficientes da Equação 4.6, a qual relaciona a velocidade de rotação com a altura.

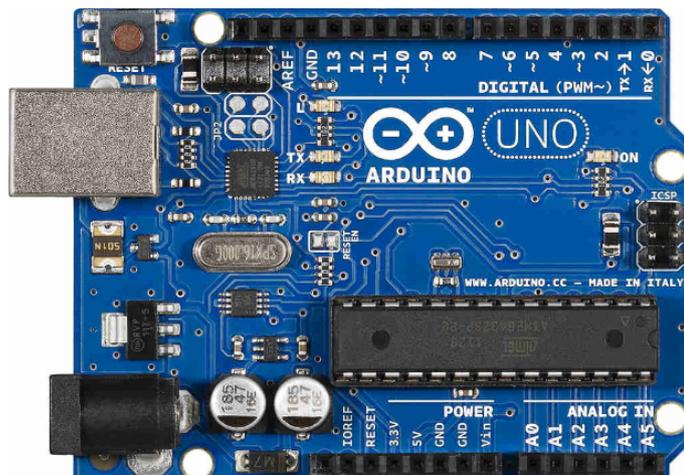
$$altura = 7.1526 \cdot \omega - 702,93 \quad (4.6)$$

4.3 O microcontrolador arduino UNO

O Microcontrolador utilizado para processar a interface e o controle de todo o Sistema, será o Arduino UNO pelo seu simples modo de programação e baixo custo. Basicamente, o Arduino é uma plataforma de prototipagem eletrônica de hardware livre e de placa única, projetada com um microcontrolador Atmel AVR com suporte de entrada/saída embutido, uma linguagem de programação padrão, a qual tem origem em Wiring, e é essencialmente C/C++ (MCROBERTS, 2011).

A Figura 33 ilustra o Microcontrolador Arduino UNO.

Figura 33 – Microcontrolador Arduino UNO



Fonte:(ARDUINO, 2015)

O projeto visa construir o levitador com ferramentas e materiais que são acessíveis, com baixo custo, flexíveis e fáceis de ser utilizado por qualquer usuário com nível médio de conhecimento.

Na Tabela 8 principais especificações fornecidas pelo fabricante.

Tabela 8 – Características do Arduino UNO

Elemento	Especificação
Microprocessador	ATmega328
Tensão de operação	7 - 12V
Limites de tensão	6 - 20V
I/O's Digitais	14(6 podem ser PWM)
Entradas analógicas	6
Corrente DC por I/O	40 mA
Memória <i>FLASH</i>	32 KB
SRAM	2 KB
EEPROM	1 KB
Velocidade do <i>Clock</i>	16 MHz

Fonte: <http://www.arduino.cc/>

4.3.1 Comunicação com o computador

O Arduino Uno possui muitas facilidades para se comunicar com um computador. O Chip ATmega328 fornece comunicação serial UART TTL que está disponível nos pinos digitais 0 (RX) e 1 (TX). Um Chip ATmega8U2 no módulo, canaliza esta comunicação para a USB e aparece como uma porta virtual para o software no computador. O firmware do '8U2 utiliza os drivers padrão USB e COM sem que nenhum driver externo seja

necessário. Entretanto, no Windows, um arquivo .inf é necessário. Ainda faltam as instruções específicas mas em breve estarão disponíveis. Sempre que estiver havendo uma comunicação serial, leds conectados aos pinos RX e TX piscarão conforme recepção e transmissão de dados respectivamente (MCROBERTS, 2011).

O aplicativo desenvolvido em MATLAB® será responsável por fazer a interpretação e processamento desses dados recebidos do Pino TX e também emitirá o comando para enviar dados pelo Pino RX ao microcontrolador.

Para isso, o MATLAB® contará com a biblioteca de comunicação com o Arduíno UNO instalada. Esta biblioteca pode ser encontrada para *download* no próprio *site* da fabricante Mathworks.

4.4 O PWM aplicado

Utilizando esta técnica de controle de tensão média, será feito as alterações na velocidade do motor da ventoinha, conseqüentemente aumentando ou diminuindo a vazão de fluxo de ar no corpo do levitador.

O Microcontrolador Arduino UNO utilizado neste projeto, possui uma resolução de PWM de 256 níveis de tensão variando de 0 a 5V, ou seja, a cada nível, a tensão varia 0,020 V, resolução mais que ideal para desenvolver o controle da velocidade do motor

O PWM será implementado diretamente no aplicativo desenvolvido em MATLAB®, e enviado por meio de valores correspondentes pela porta serial do computador ao arduino.

Não é necessário criar uma lógica para comutar os níveis geradores de PWM, uma vez que existem funções pré-programadas para realizar esta comutação. Apenas é necessário que envie o valor de tensão médio desejado na saída de um determinado pino.

A Figura 34 ilustra parte o script na plataforma do arduíno executa esta ação.

Figura 34 – Envio de tensão média ao pino D6 do Arduino



```
sketch_may15a | Arduino 1.6.4
Arquivo Editar Sketch Ferramentas Ajuda
sketch_may15a $
void setup() {
  Serial.begin(9600); //inicia a porta serial
  pinMode(6,OUTPUT); //define pino 6 como saída }
void loop()
{
char vv = Serial.read(); // Lê o valor recebido pela serial
float V=atoi(vv); // converte o valor recebido para float
analogWrite(6,v); // envia o valor de tensão ao pino 6 arduino
```

Fonte: Autor

Quando o Arduino é conectado no computador, ele é reconhecido como uma determinada porta serial, a qual deve ser identificada no gerenciador de dispositivos do computador.

No caso deste computador, foi reconhecida como 'com3'. E então configurada neste desenvolvedor do próprio arduíno.

O *script* ilustrado na Figura 34 será o código que ficará dentro do microcontrolador responsável pela interpretação dos dados recebidos do aplicativo de controle em MATLAB®.

4.5 O amplificador do PWM

O PWM enviado pelo microcontrolador Arduíno UNO emite uma tensão média de 0 a 5V com uma resolução de 256 níveis e com uma corrente baixa, sendo assim inutilizável para alimentar a ventuinha, visto que ela necessita de tensões de até 12V e corrente a partir de 600 mA, para este projeto. Portanto, foi necessário utilizar um amplificador que aumentasse a tensão proporcionalmente com o aumento dos níveis de PWM.

Matematicamente, a escala de amplificação deveria seguir a Equação 4.9.

$$\frac{V_{PWM}}{5} = \frac{V_{AMP}}{12} \quad (4.7)$$

$$12V_{PWM} = 5V_{AMP} \quad (4.8)$$

$$V_{AMP} = \frac{12 \cdot V_{PWM}}{5} \quad (4.9)$$

Para tanto, foi projetado um circuito amplificador com um Transistor MOSFET IRFZ34N para fazer o chaveamento proporcional da tensão que alimentará o motor DC da ventoinha (BOYLESTAD; NASHELSKY, 2004).

Um diodo 1N4751 que foi colocado em antiparalelo com o motor para que impedisse uma possível passagem de corrente reversa provocada por uma rotação acidental da hélice da ventoinha provocando a queima do MOSFET (BOYLESTAD; NASHELSKY, 2004).

É de grande importância colocar um resistor no GATE do MOSFET para garantir que o circuito não seja acionado acidentalmente no caso de um desvio de corrente, neste caso foi utilizado um resistor de 330Ω (BOYLESTAD; NASHELSKY, 2004).

A Tabela 9 resume os componentes utilizados no circuito.

Tabela 9 – Componentes do Circuito Amplificador

Componente	Especificação
Transistor	IRFZ34N
Diodo	1N4751
Resistor	330Ω
Fonte Externa	12 V

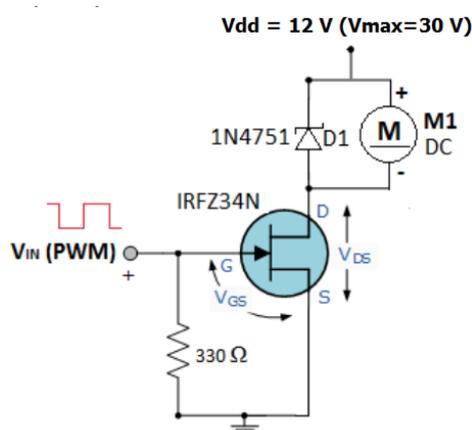
Fonte: (BOYLESTAD; NASHELSKY, 2004)

O pulso PWM enviado pelo microcontrolador é ligado diretamente ao GATE do MOSFET. No momento em que o sinal do PWM está em nível alto, o MOSFET fecha o contato, permitindo a passagem de corrente pelo seu Dreno, e acionando o Motor M1 aplicando em seus polos a tensão de entrada Vdd de 12 V.

No momento em que o nível de tensão do PWM é 0, o MOSFET abre, impedindo a passagem e corrente não sendo aplicado nenhum valor de tensão nos polos do Motor.

Deste modo, o valor de tensão aplicado no Motor, será um valor médio variável de 0 a 12V de acordo com o tempo em que o nível de tensão do PWM permanece em alto. A Figura 35 mostra o esquema elétrico de montagem do circuito amplificador projetado.

Figura 35 – Circuito Amplificador do PWM



Fonte: (BOYLESTAD; NASHELSKY, 2004)

4.6 O controlador PID aplicado

O Controlador PID ilustrado na Figura 15 será implementado virtualmente por meio de um aplicativo desenvolvido no programa MATLAB® e processado no microcontrolador Arduíno UNO.

Na interface deste aplicativo, será possível inserir os parâmetros de sintonia, por meio dos quais serão aplicadas as ações de controle conforme o desejado.

A Figura 36 ilustra a função que aplica o controle PID em MATLAB® na Função de Transferência obtida após a modelagem.

Figura 36 – Linha de comando que aplica o controle PID

```
%% Aplicação dos ganhos no Controlador PID
Kp=1.0415;    % Ganho Proporcional
Ki=0.0056;   % Ganho Integral
Kd=0.0005736; % Ganho Derivativo

Gc=pid(Kp,Ki,Kd); %Cria função de controle
```

Fonte: Autor

4.7 Sintonia do controlador PID

Para determinar os parâmetros de sintonia do controlador, dentre várias técnicas existentes na engenharia de controle moderno, optou-se por utilizar o método de Ziegler-Nichols, Cohen-Coon e a AMIGO.

4.7.1 Ziegler e Nichols

Conforme descrito na teoria no Capítulo 2, inicialmente, precisamos determinar os valores de τ (Tau), $\theta(L)$ e K. Para tanto, é preciso aplicar um degrau unitário na função, e então deve ser traçado uma reta na tangente do ponto de inflexão para que seja calculado estas constantes. O ponto de inflexão da curva, é exatamente onde ela muda de concavidade, e é possível encontrar realizando a derivada de segunda ordem da função

O MATLAB® possui recursos matemáticos que nos permite encontrar estes valores por meio de manipulações algébricas. Na Figura 37 é possível observar cada linha de comando deste cálculo.

Figura 37 – Script da Sintonia por Ziegler e Nichols

```

1 - clear all; close all; clc
2 - %% Sintonia com Método de Ziegler & Nichols
3 - num=[0.1178]; den=[0.0000001102 0.00005986 0.004599];
4 - Gp = tf(num,den);
5
6 - %% Representação linear p/ determinar Teta, Tau e K
7 - dt=0.001;
8 - t=0:dt:0.07; % intervalo da grafico
9 - y=step(Gp,t); % aplica o degrau
10 - dy=diff(y)/dt; % derivada
11 - [m,p]=max(dy); % ponto de inflexão
12 - d2y=diff(dy)/dt; % segunda derivada
13 - yi=y(p);
14 - ti=t(p);
15 - L=ti-yi/m % Retardo
16 - Tau=(y(end)-yi)/m+ti-L % Constante de Tempo
17 - plot(t,y,'b',[0 L L+Tau t(end)],[0 0 y(end) y(end)],'k');
18 - grid
19 - ylabel('Velocidade Angular(rad/s)');
20 - xlabel('Tempo(s)'); legend('Real','Aproximação Linear');
21 - %% Cálculo dos ganhos a partir de Tau e L
22 - Kp=1.2*Tau/L
23 - Ki=2*L
24 - Kd=0.5*L
25

```

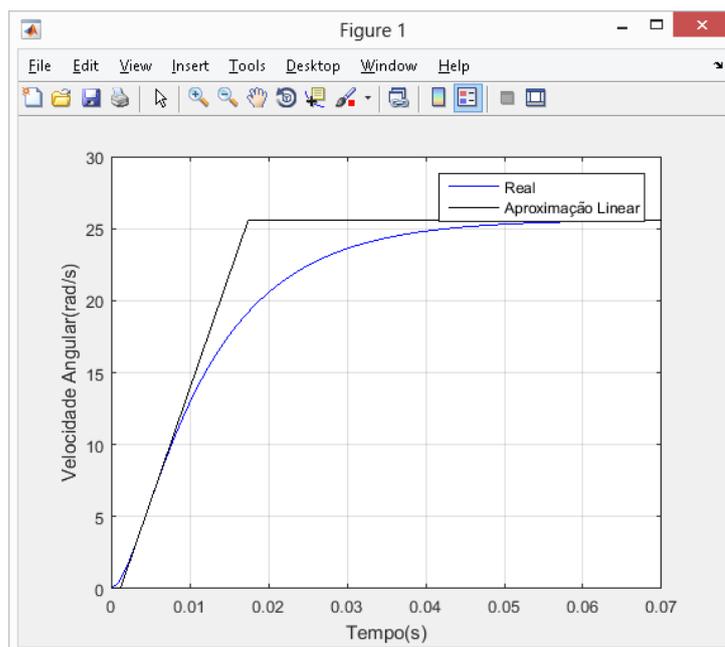
Fonte: Autor

O método para encontrar o ponto de inflexão consiste em colocar a função de transferência no domínio do tempo, e em seguida, derivar duas vezes, e então igualar a zero. Após a resolução da equação, os resultados serão as coordenadas dos pontos de inflexão, sendo assim facilmente calculados as constantes subtraindo dos pontos de cruzamento das retas.

- O primeiro bloco do *script*, atribui a função de transferência na variável Gs.
- O segundo bloco calcula e plota a resposta ao degrau e sua aproximação linear.
- O terceiro bloco calcula os valores de ganho conforme a Tabela 2.

A Figura 38 ilustra a função e sua aproximação linear.

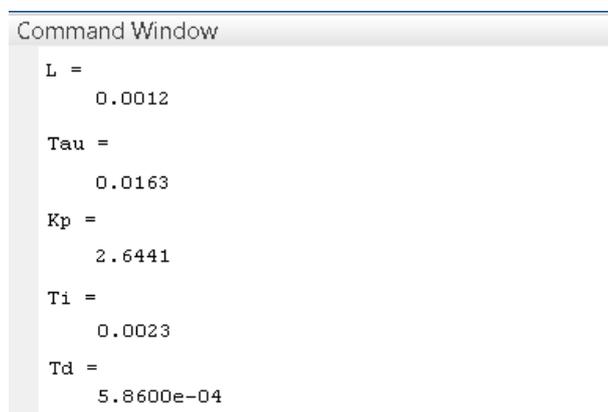
Figura 38 – Resposta ao degrau unitário e aproximação linear



Fonte: Autor

O script desenvolvido nos retorna os valores dos respectivos ganhos do controlador que devem ser utilizados na implementação do controlador PID. A Figura 39 ilustra estes valores.

Figura 39 – Ganhos por Ziegler e Nichols



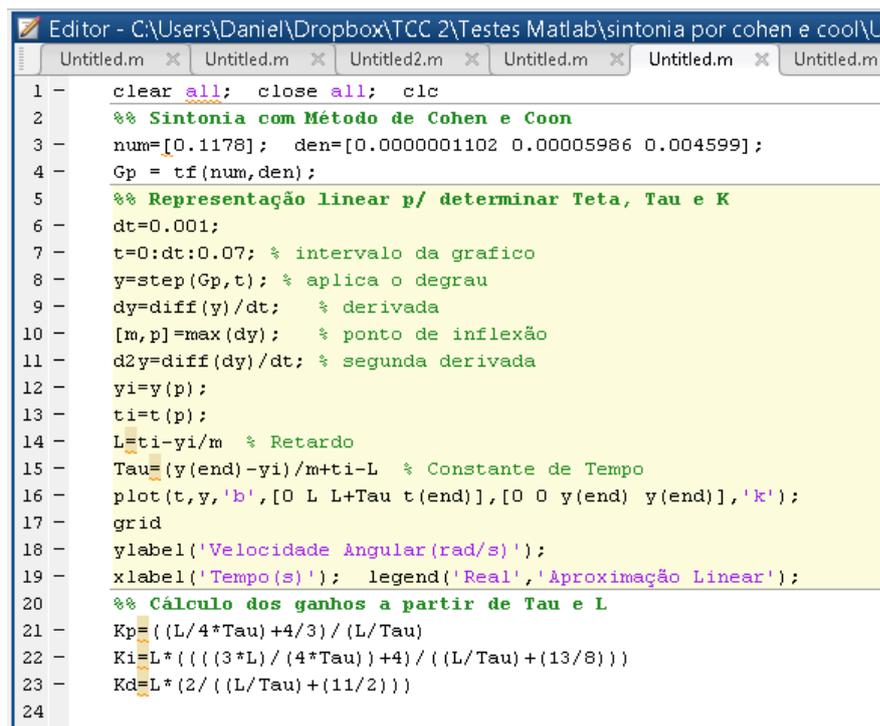
Fonte: Autor

4.7.2 Cohen-Coon

O processo utilizado para determinar os valores do ganho K, do atraso L e da constante de tempo T é exatamente o mesmo do utilizado para a sintonia de Ziegler-Nichols, entretanto devemos aplicar as equações da Tabela 3 no cálculo do ganhos do controlador.

Na Figura 40 pode-se visualizar o *script* do programa que realiza este cálculo.

Figura 40 – Script da Sintonia por Cohen-Coon



```

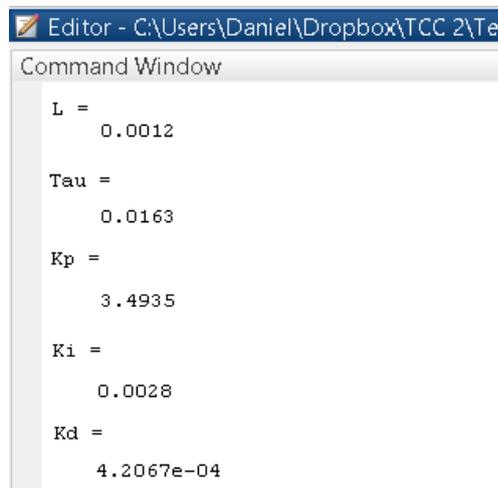
1 - clear all; close all; clc
2 - %% Sintonia com Método de Cohen e Coon
3 - num=[0.1178]; den=[0.0000001102 0.00005986 0.0045999];
4 - Gp = tf(num,den);
5 - %% Representação linear p/ determinar Teta, Tau e K
6 - dt=0.001;
7 - t=0:dt:0.07; % intervalo da grafico
8 - y=step(Gp,t); % aplica o degrau
9 - dy=diff(y)/dt; % derivada
10 - [m,p]=max(dy); % ponto de inflexão
11 - d2y=diff(dy)/dt; % segunda derivada
12 - yi=y(p);
13 - ti=t(p);
14 - L=ti-yi/m % Retardo
15 - Tau=(y(end)-yi)/m+ti-L % Constante de Tempo
16 - plot(t,y,'b',[0 L L+Tau t(end)],[0 0 y(end) y(end)],'k');
17 - grid
18 - ylabel('Velocidade Angular(rad/s)');
19 - xlabel('Tempo(s)'); legend('Real','Aproximação Linear');
20 - %% Cálculo dos ganhos a partir de Tau e L
21 - Kp=((L/4* Tau)+4/3)/(L/Tau)
22 - Ki=L*((3*L)/(4* Tau)+4)/((L/Tau)+(13/8))
23 - Kd=L*(2/((L/Tau)+(11/2)))
24

```

Fonte: Autor

Ao executar este código, é retornado os valores de ganho mostrados na Figura 41.

Figura 41 – Ganhos por Cohen-Coon



```

Command Window

L =
    0.0012

Tau =
    0.0163

Kp =
    3.4935

Ki =
    0.0028

Kd =
    4.2067e-04

```

Fonte: Autor

4.7.3 AMIGO

Utilizaremos também para o cálculo dos parâmetros 'Tau' e 'L' deste método, a curva da resposta ao degrau e aproximação Linear de Zieegler e Nichols.

A Figura 42 ilustra o script do programa que calcula estes parâmetros.

Figura 42 – Script da sintonia por método AMIGO

```

Editor - C:\Users\Daniel\Dropbox\TCC 2\Testes Matlab\Sintonia método AMIGO\Unfile
Untitled.m x Untitled.m x Untitled2.m x Untitled.m x Untitled.m x +
1 - clear all; close all; clc
2 - %% Sintonia com Método AMIGO
3 - num=[0.1178]; den=[0.0000001102 0.00005986 0.004599];
4 - Gp = tf(num,den);
5 - %% Representação linear p/ determinar Teta, Tau
6 - dt=0.001;
7 - t=0:dt:0.07; % intervalo da grafico
8 - y=step(Gp,t); % aplica o degrau
9 - dy=diff(y)/dt; % derivada
10 - [m,p]=max(dy); % ponto de inflexão
11 - d2y=diff(dy)/dt; % segunda derivada
12 - yi=y(p);
13 - ti=t(p);
14 - L=ti-yi/m % Retardo
15 - Tau=(y(end)-yi)/m+ti-L % Constante de Tempo
16 - plot(t,y,'b',[0 L L+Tau t(end)],[0 0 y(end) y(end)],'k');
17 - grid
18 - ylabel('Velocidade Angular(rad/s)');
19 - xlabel('Tempo(s)'); legend('Real','Aproximação Linear');
20 - %% Cálculo dos ganhos a partir de Tau e L
21 - Kp=(0.2+0.45*(Tau/L))
22 - Ki=L*((0.4*L+0.8*Tau)/(L+0.1*Tau))
23 - Kd=(0.5*L*Tau)/(Tau+0.3*L)
24 - t=L/(L+Tau)
25 - if t>0.5
26 -     b=0
27 - else
28 -     b=1
29 - end

```

Fonte: Autor

O valores de sintonia encontrados por meio deste método são apresentados na Figura 43.

Figura 43 – Ganhos por método AMIGO

```

Command Window
L =
    0.0012

Tau =
    0.0163

Kp =
    1.0415

Ki =
    0.0056

Kd =
    5.7360e-04

fx >>

```

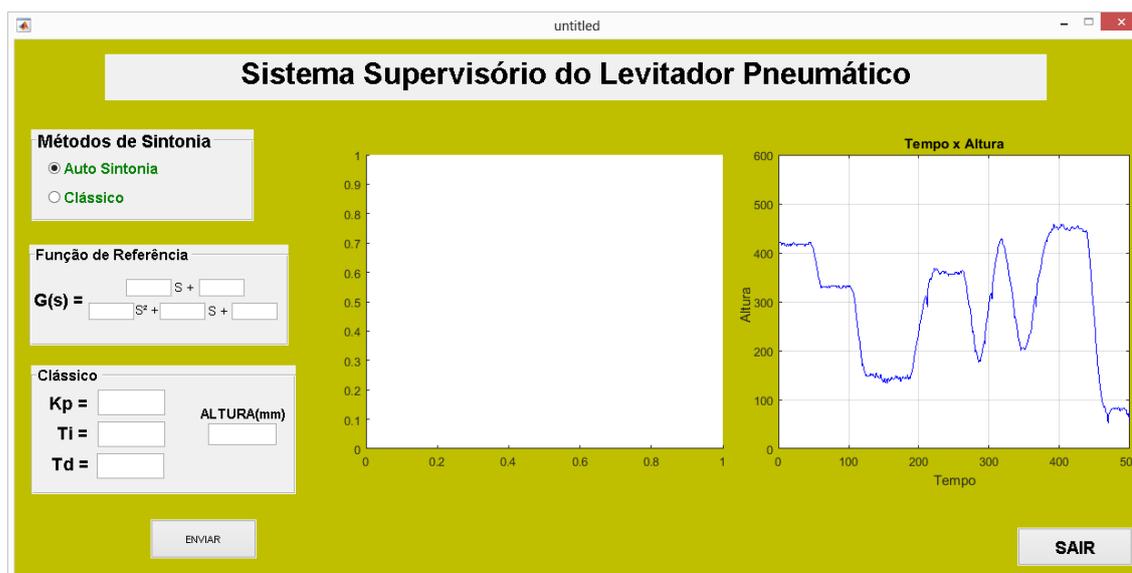
Fonte: Autor

4.7.4 O aplicativo de controle

O Aplicativo de Controle é onde o sistema será monitorado, isto é, o sensor de distância fará a leitura da distância onde o corpo levitado está e também a resposta de controle do sistema para que possa ser feita uma comparação na resposta do sistema.

A Figura 44 ilustra a Interface do Aplicativo desenvolvido.

Figura 44 – Aplicativo de Controle do Levitador



Fonte: Autor

Esta Interface deve apresentar no primeiro gráfico a resposta do sistema sintonizado de acordo com valores determinados pelo usuário, e no segundo gráfico, apresenta o monitoramento da altura instantânea do corpo levitado.

Figura 45 – Script completo na memória do Arduíno

```

sketch_may15a | Arduino 1.6.4
Arquivo Editar Sketch Ferramentas Ajuda
sketch_may15a $
void setup() {
  Serial.begin(9600); //inicia a porta serial
  pinMode(5, INPUT); // define o pino 5 como entrada (recebe)
  pinMode(7, OUTPUT); // define o pino 7 como saída (envia)
  pinMode(6, OUTPUT); //define pino 6 como saída }
void loop(){
  char vv = Serial.read(); // Lê o valor recebido pela serial
  float V=atoi(vv); // converte o valor recebido para float

  analogWrite(6,v); // envia o valor de tensão ao pino 6 arduino

  digitalWrite(7, LOW); // inicio da rotina para calcular a distancia
  delayMicroseconds(2); // zera o pino 7 e mantem por 2 us

  digitalWrite(7, HIGH); // envia 1 para o pino 7 e mantem 10 us
  delayMicroseconds(10);

  digitalWrite(7, LOW); // zera novamente o pino 7

  float distancia = pulseIn(5,HIGH)/29.4/2;
  // calcula a distância tendo o tempo que o sinal enviado levou para retornar
  //Serial.print("Distancia em CM: ");
  Serial.println(distancia);
  delay(50); }

```

Fonte: Autor

Assim como existe um script de envio de tensão média no desenvolvedor do próprio arduíno, também é necessário um script para fazer a leitura da distância pelo sensor ultrassônico. A Figura 45 ilustra o script completo que estará na memória do microcontrolador que será responsável também por receber e enviar os dados pela porta serial do computador.

Com este código em sua memória, o arduíno será capaz de comunicar-se com o aplicativo de controle, isto é, enviará a medição do sensor ultrassônico e receberá o valor de tensão que deverá ser aplicada no pino de saída para o amplificador.

Na Figura 46, pode-se observar a principal função de envio e recebimento de dados.

Figura 46 – Função de envio e recebimento de dados em MATLAB

```

78 function pushbutton1_Callback(hObject, eventdata, handles)
79 % hObject handle to pushbutton1 (see GCBO)
80 % eventdata reserved - to be defined in a future version of MATLAB
81 % handles structure with handles and user data (see GUIDATA)
82 delete(instrfindall);
83 clc;
84 s = serial('COM3'); % atribui em 's' a porta serial 'com3'
85
86 set(s,'InputBufferSize',16); % definições de configurações básicas
87 set(s,'FlowControl','hardware'); % para a comunicação da porta serial
88 set(s,'BaudRate',9600);
89 set(s,'Parity','none');
90 set(s,'Databits',8);
91 set(s,'StopBit',1);
92 set(s,'Timeout',0.1);
93 axes(handles.axes1); % necessário para plotar na interface do guide
94 fopen(s); % abre a porta serial para começar a operar
95 t=1; % variável auxiliar
96 clear x; % limpa variável auxiliar para plotagem
97 clear y; % limpa variável auxiliar para plotagem
98 while(1) % laço onde ocorre o monitoramento da distância
99     pause(0.0001); % pausa para ler a porta serial
100    fprintf(s,TENSAO()); % envia tensão correspondete a altura digitada
101    a=fscanf(s); % faz a leitura da distância via porta serial
102    y(t)=str2double(a); % converte dados string para double
103    y(t)=53.4-y(t); % encontra a altura real do corpo
104    if((y(t)~=NaN) & (y(t)<60)) % filtro passa-baixa impedindo que valores
105        % acima de 60 cm sejam computados
106
107        if t>1
108            if(abs(y(t)-y(t-1))<3) % Filtro para impedir que variações menores
109                y(t)=y(t-1); % que a precisão do sensor influenciem
110            end
111        end
112
113        y(t)=y(t)*10; % converte distância para milímetros
114        x(t)=t; % atribui t ao vetor do eixo 'x' para plotagem
115        plot(x,y,'b-'),grid % plota o gráfico de monitoramento da distância
116        axis([0 500 0 600]); % define o tamanho do objeto de plotagem
117        t=t+1; % incrementa 1 na variável de amostragem
118
119        title('Tempo x Altura');
120        xlabel('Tempo');
121        ylabel('Altura');
122    end
123    fclose(s)

```

Esta função é executada quando o Botão 'ENVIAR' do Aplicativo é pressionado. Cada parte do script está comentada discriminando sua utilidade específica no código.

É válido explicar a respeito da Função 'TENSAO()' descrita na linha 100 do código acima.

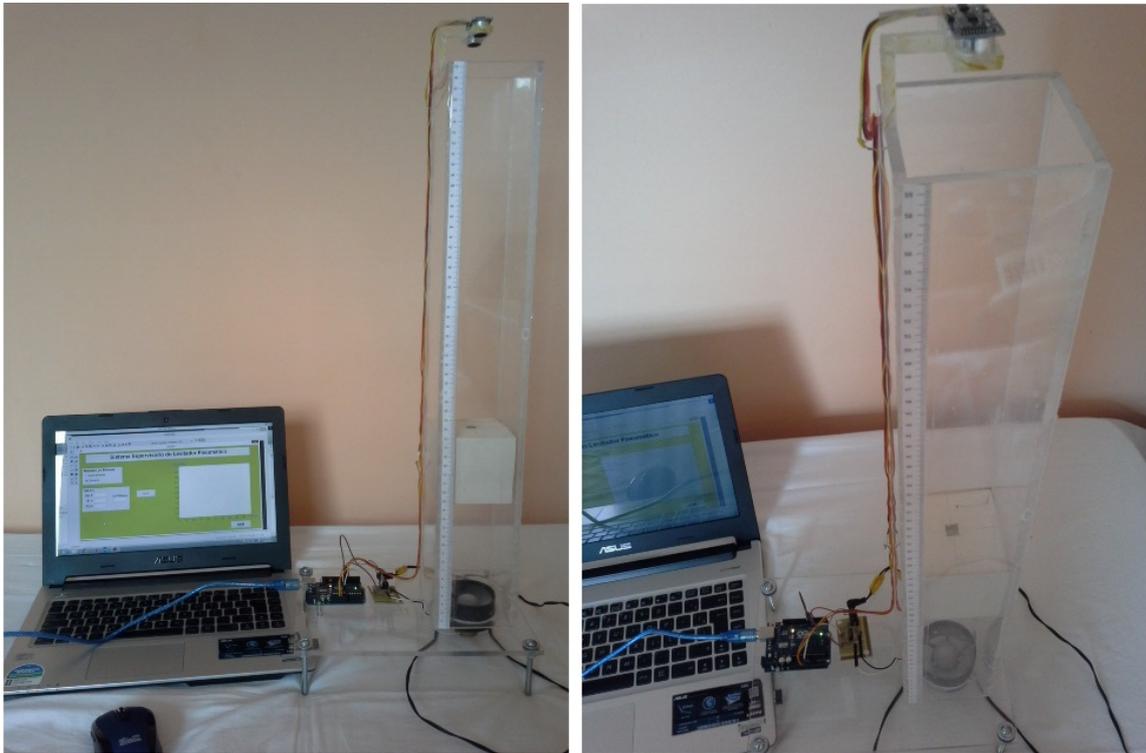
Esta função seria onde o controlador PID seria aplicado. Mas para isso, seria necessário que o sistema físico estivesse em condições bem específicas de especificação. E devido a limitações físicas de construção, não foi possível realizar testes para que essa função pudesse ser concluída. No capítulo 5 será explanado mais a respeito.

5 RESULTADOS OBTIDOS

5.1 Construção da estrutura física do levitador

O primeiro desafio deste projeto foi montar a estrutura física do Levitador, pois é a partir dele que deve ser feito os testes para o desenvolvimento das outras etapas do projeto. Entretanto, a estrutura do levitador, no espaço em que o corpo se movimenta, deveria ser perfeitamente quadrada e praticamente sem atrito para que não influenciasse no deslocamento do corpo, o que se tornou-se uma tarefa extremamente difícil para uma construção manual, apesar de ser cuidadosamente dimensionada e acoplada.

Figura 47 – Estrutura do Levitador Pneumático

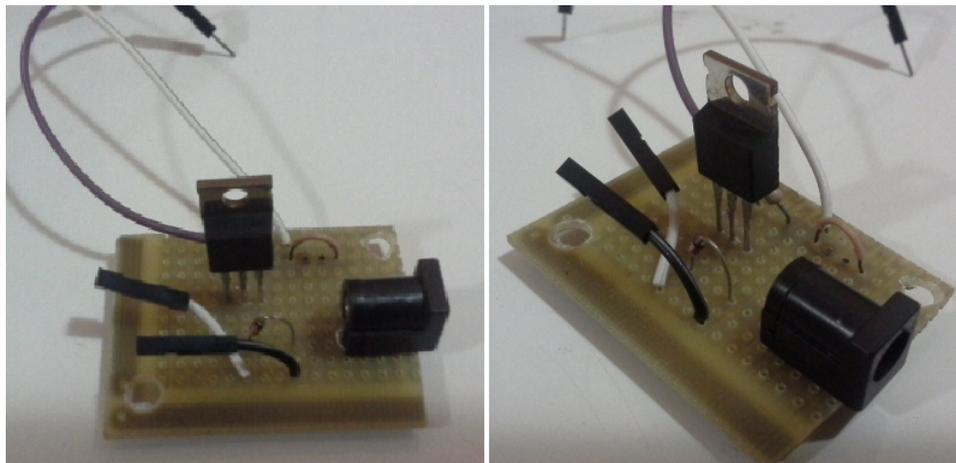


Fonte: Autor

5.2 Prototipagem do amplificador

A montagem da placa amplificadora foi realizada com sucesso em seu funcionamento, obedecendo a proporção perfeita para a qual estava especificada. É possível visualizar na Figura 48 a prototipagem da placa amplificadora montada.

Figura 48 – Prototipagem da Placa Amplificadora



Fonte: Autor

A montagem desta placa é fundamental uma vez que sem a mesma, é impossível somente com o microcontrolador, fornecer a corrente e tensão suficiente para o funcionamento do motor DC.

Foram feitos os testes e ela realmente segue a Equação 4.9, podendo ser utilizada no projeto em sua integralidade.

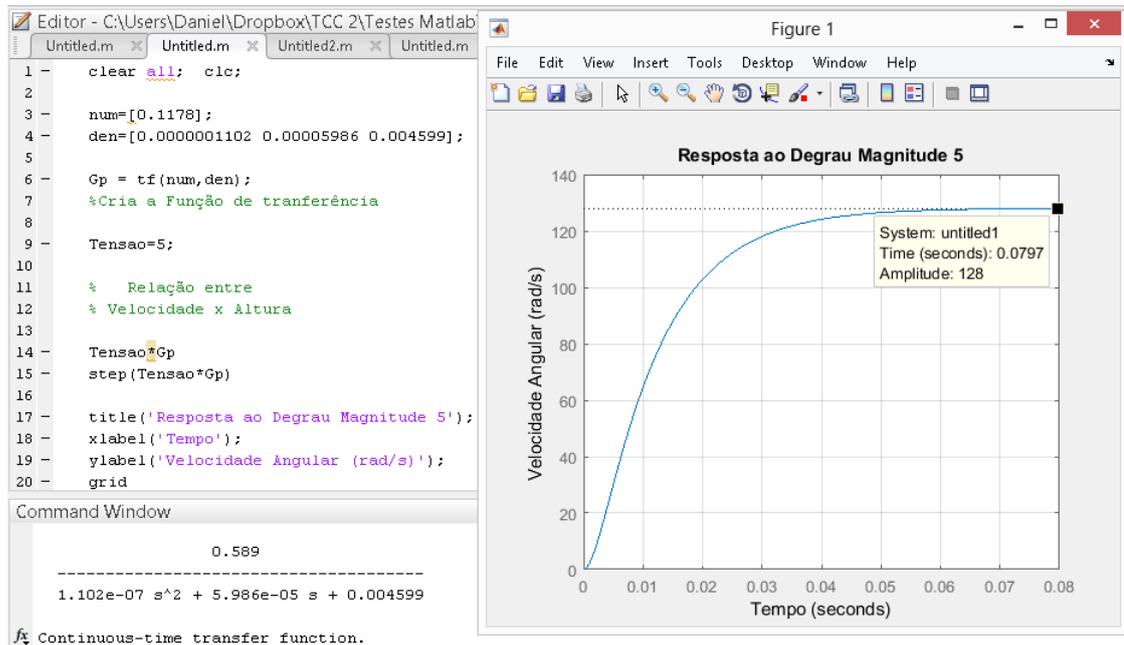
5.3 Modelagem e função de transferência

Primeiramente, é fundamental destacar a elaboração da modelagem matemática do motor DC da ventoinha, pois é nela que será aplicado o controlador PID. Este passo é o primordial para que o processo de estudo e análise das respostas possam ser utilizados para ajustes e melhorias na resposta do sistema.

$$G(s) = \frac{\Omega(s)}{V_i(s)} = \frac{117,8 \cdot 10^{-3}}{1,102 \cdot 10^{-7} \cdot s^2 + 5,986 \cdot 10^{-5} \cdot s + 4,599 \cdot 10^{-3}} \quad (5.1)$$

Após seu desenvolvimento, foi buscado realizar testes para comprovar a eficiência e veracidade deste resultado bem como sua validação que pode realizada por meio do teste com a resposta ao degrau de magnitude 5 observada nas Figuras 49 e 51.

Figura 49 – Resposta ao Degral de Magnitude 5



Fonte: Autor

Aplicando a Equação 4.6 de relação entre a velocidade de rotação da ventoinha e a altura de estabilização do corpo levantado, considerando a velocidade de estabilização para magnitude 5 (5 V), tem-se:

$$altura = 7.1526 \cdot \omega - 702,93 \quad (5.2)$$

$$altura = 7.1526 \cdot 128 - 702,93 \quad (5.3)$$

$$altura = 212 \text{ mm} \quad (5.4)$$

Portanto, aplicando uma tensão de 5 V nos polos do Motor DC, que de acordo com a Equação 4.9 equivale a 2,08 V no PWM do Arduíno, o corpo deveria se estabilizar na altura de 212 mm. Entretanto obtivemos uma variação de 3 milímetros para este teste como visualizamos nas Figuras 50 e 51.

Figura 50 – Script de envio de tensão ao Arduíno

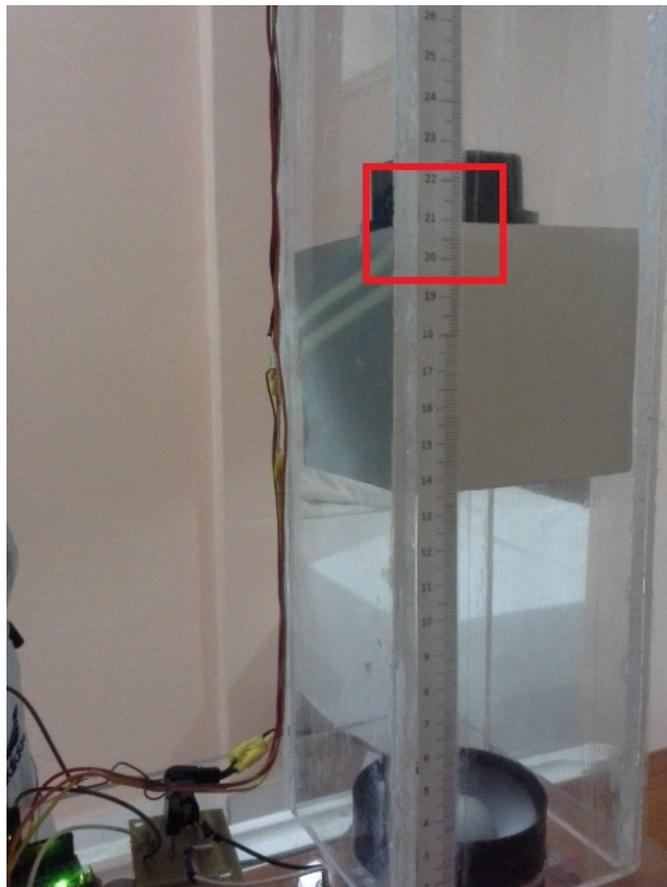
```

1      %% Envio de PWM ao Arduíno UNO
2
3      a = arduino('com3','uno'); % carrega o arduino na porta onde stá conectado
4
5      v=2.08; % valor de tensão setado para 212 mm
6      writePWMPulse(a,'D6',v); % envia valor de tensão diretamente
7      % ao pino do arduino
8

```

Fonte: Autor

Figura 51 – Altura obtida para 5V no Motor DC da Ventoinha



Fonte: Autor

5.4 Implementação e sintonização do PID

Um passo muito relevante foi o desenvolvimento da aplicação do controlador PID por meio do MATLAB®, que na prática consiste apenas em aplicar o comando “pid(Kp,Ki,Kd)” no sinal de controle, dentro do script no MATLAB®, como observado no exemplo da Figura 52.

Figura 52 – Parte do código onde implementa-se o PID

```
13 %% Sintonia do Controlador PID
14 - Kp=5.5434;    % Ganho Proporcional
15 - Ki=0.1319;   % Ganho Integral
16 - Kd=0.0206;  % Ganho Derivativo
17
18 - Gc=pid(Kp, Ki, Kd)
```

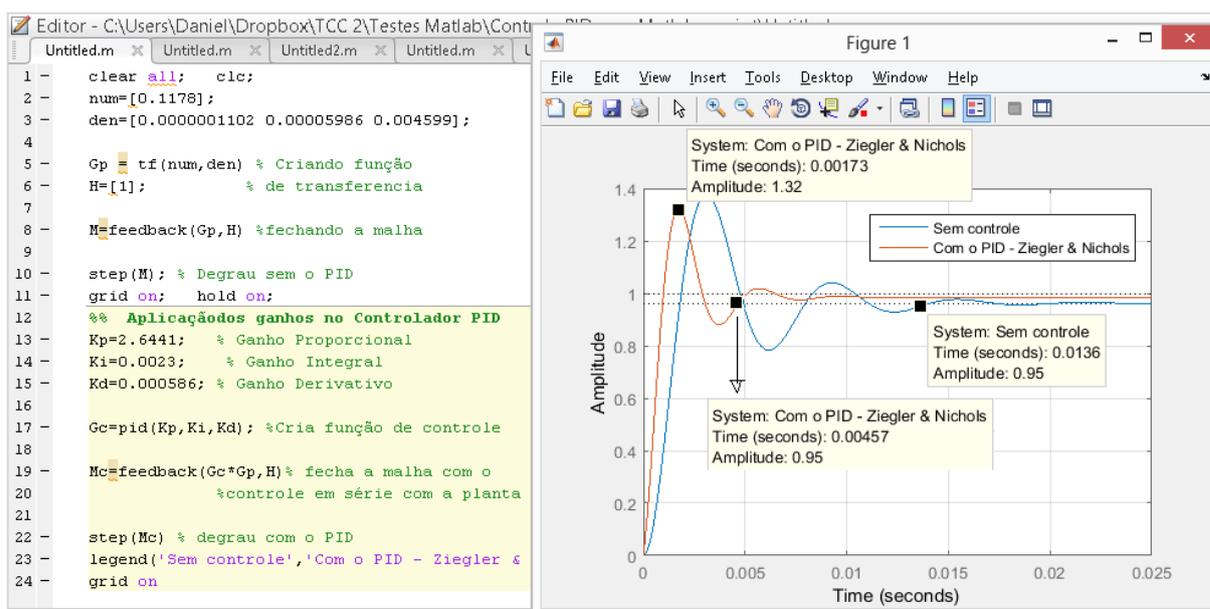
Fonte: Autor

A parte complicada da implementação deste controle, consiste em sua sintonização pelos métodos descritos neste trabalho, os quais forneceram os seguintes resultados:

5.4.1 Ziegler e Nichols:

A Figura 53 exibe o código em MATLAB® da implementação deste método de sintonia utilizando para o cálculo dos ganhos a Tabela 2, juntamente com a resposta ao degrau para esta sintonia.

Figura 53 – Resposta do degrau com Sintonia de Ziegler e Nichols



Fonte: Autor

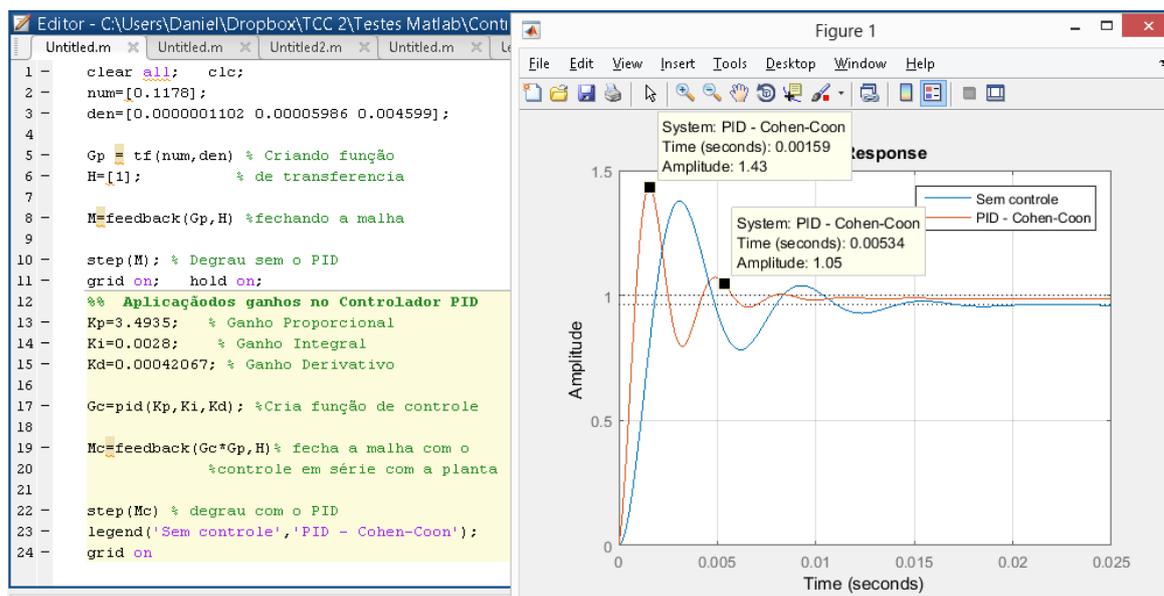
Realizando uma análise da resposta ao degrau de magnitude 5 (5V), é possível observar que foi obtido um tempo de acomodação para 5%, em um terço do tempo para o sistema sem controle, e obteve uma pequena baixa no sobressinal atingindo 32%

5.4.2 Cohen-Coon

Este método baseia-se inicialmente nos trabalhos de Ziegler e Nichols, entretanto propõe a utilização e outras fórmulas matemáticas para o cálculo dos ganhos, sendo estas desenvolvidas por um método próprio já explanado no Capítulo 2. Estas fórmulas estão descritas na Tabela 3 e foram utilizadas na implementação do controlador PID com a sua sintonia, demonstrado na Figura 54.

É possível observar na Figura 54 que o método de sintonia de Cohen-Cool fornece uma resposta não muito satisfatória no que diz respeito ao sobressinal que atinge um valor 43%.

Figura 54 – Resposta do degrau com Sintonia de Cohen e Coon



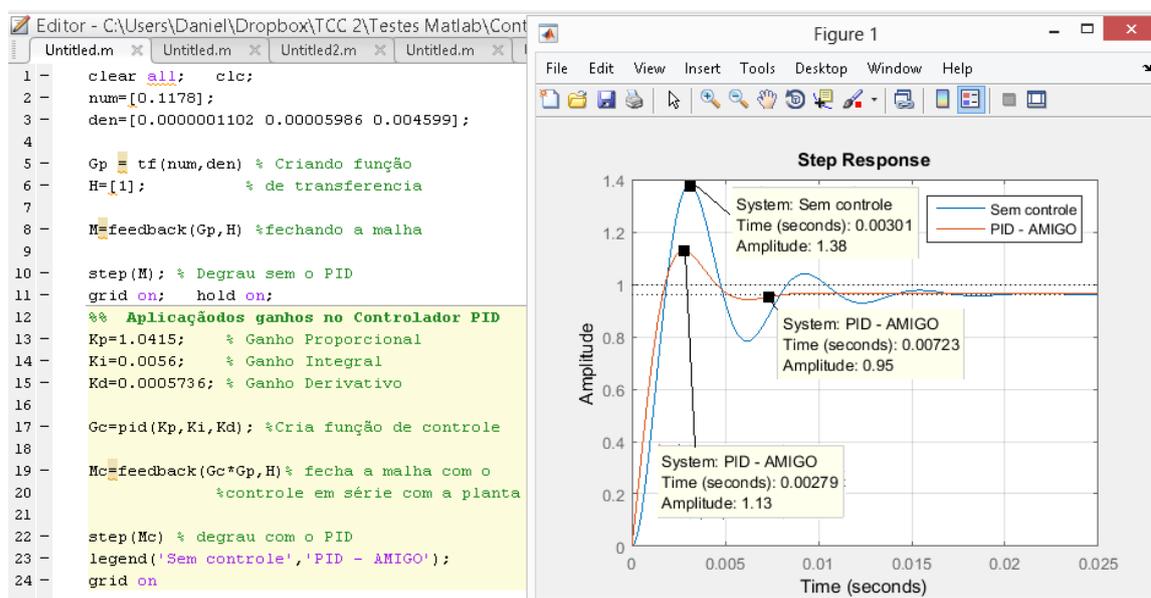
Fonte: Autor

Entretanto o tempo de acomodação do sistema é baixo chegando a 5 milisegundos, um tempo relativamente baixo se comparado ao tempo de acomodação do sistema sem controle que é de aproximadamente 13.6 milisegundos como pode ser observado na Figura 53.

5.4.3 AMIGO

Aplicando as Fórmulas demonstradas na Tabela 4 foi obtido a resposta ao degrau ilustrada na Figura 55.

Figura 55 – Resposta do degrau com Sintonia por AMIGO



Fonte: Autor

5.5 Testes do controlador no sistema real

Infelizmente, devido a limitações físicas da própria estrutura construída para o deslocamento do corpo levitado, não foi possível realizar testes do sistema controlado, pois as medições reais de altura obtidas pelo sensor não corresponderiam às alturas nos seus devidos instantes, trazendo uma resposta real totalmente fora de proporção para a análise de desempenho do sistema, não podendo assim ser utilizada como referência para uma melhora em seu desempenho.

5.5.1 Aplicativo de controle

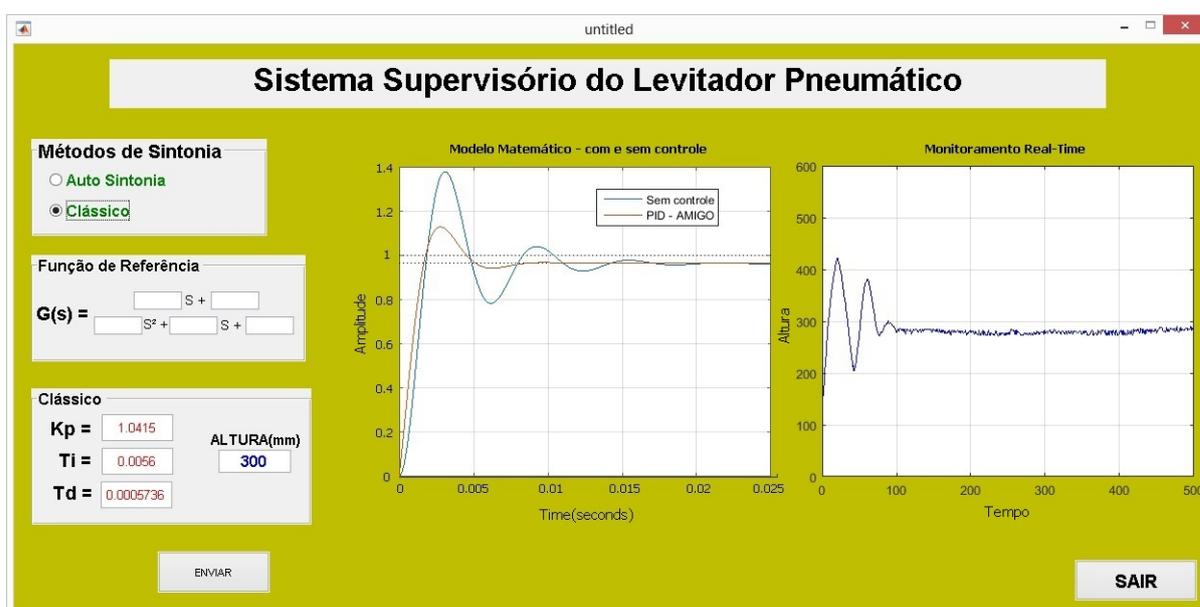
O aplicativo de controle e monitoramento proposto, funcionou corretamente de forma parcial, realizando medição em tempo real da altura do corpo levitado, e mostrando em outro gráfico o desempenho do controlador aplicado na planta do sistema.

Entretanto não é possível fazer qualquer tipo de comparação entre o modelo controlado e a resposta real exibida uma vez que devido a limitações já explanadas, não foi possível aplicar o controle no sinal real do sistema.

Na Figura 56 é possível visualizar um teste realizado de monitoramento real do sistema, onde foi testado para 300 milímetros. É fácil observar que a resposta do sistema real sem controle no gráfico a direita, apresentou bastante erro. Enquanto que no sistema matemático nos trouxe uma resposta bastante satisfatória.

Este erro e variações são justamente devido atritos do sistemas e travas no deslocamento do corpo levitado, o que tornou impossível realizar testes para o ajuste do sistema de controle real.

Figura 56 – Aplicativo de Controle e Monitoramento



Fonte: Autor

O sistema real sem controle em regime permanente para este teste é ilustrado na Figura 57.

Figura 57 – Altura de Estabilização para o teste



Fonte: Autor

Também ainda não foi possível aplicar métodos de auto sintonia propostos neste projeto, podendo ser explorado em trabalhos futuros.

6 CONCLUSÕES

O projeto mostrou-se bastante útil por possibilitar a visualização de princípios básicos de controle moderno os quais serão utilizados em projetos mais complexos. O fato do projeto tratar de uma questão que envolve várias disciplinas de diversos cursos, o torna mais interessante, propiciando a todos que o operam e que com ele trabalham, ampliar o conhecimento em diversas áreas, mais especificamente na de controle moderno.

É possível concluir que a função de transferência exibida na Equação 2.10, encontrada pelo método de modelagem clássico apresentado, corresponde de fato ao motor DC da ventoinha utilizado, uma vez que foi possível estabelecer relação linear entre a resposta da função, dada em radianos por segundo, e a altura de estabilização.

A implementação de um controlador digital embarcado permite ao usuário uma maior liberdade no que se trata de testes com o controlador e também na facilidade de implementação de outros tipos de controladores, sem que para isso seja necessária a construção de um novo circuito como seria no caso do controlador analógico.

Neste trabalho, foram propostas técnicas de sintonia para o controlador PID aplicado. Foram estudados o desempenho de cada um por meio da análise de seus principais índices, isto é, seu tempo de acomodação e o sobresinal de cada resposta. Chegando à conclusão que o melhor método de sintonia para o sistema proposto neste projeto é o AMIGO, os quais nos forneceram os seguintes valores para os ganhos:

$$K_p = 1.0415$$

$$K_i = 0.0056$$

$$K_d = 0.0005736$$

Ou seja, dentro dos métodos de sintonia testados, o corpo levitado atingiria a sua estabilização na altura especificada de forma mais rápida utilizando o método de sintonia AMIGO, o qual nos trouxe uma melhora significativa que passou de 38% para 13% e um tempo de acomodação reduzido em quase a metade do mesmo tempo para o sistema sem controle, isto é, aproximadamente 8 milissegundos.

Pode-se também concluir que é possível construir um levitador pneumático utilizando equipamentos e especificações descritas ao longo deste trabalho, entretanto para que fosse possível o real funcionamento deste sistema, a estrutura deve ser feita de maneira a zerar as limitações físicas e o atrito entre o corpo levitado e as paredes da estrutura, pois torna-se impossível a realização de testes do sistema com o controlador devido a travas mecânicas e atrito na estrutura.

Para que fosse possível um perfeito funcionamento do sistema com controlador, seria necessário sua construção de forma usinada, tanto da estrutura como do corpo levitado, fornecendo perfeitas dimensões, evitando assim travas mecânicas e atritos criados devida a acoplagem das paredes laterais.

6.1 Trabalhos futuros

Neste projeto foi implementado apenas o controlador do tipo PID. Entretanto é possível realizar o estudo para que seja aplicado outros tipos de controle como o controle robusto ou técnicas de controle adaptativo.

É interessante também a possibilidade de modelar este sistema para que se possa utilizar como objeto de levitação qualquer corpo dentro de determinadas especificações de tamanho e peso obviamente.

É possível ainda substituir a fonte de fluxo de ar, colocando por exemplo um reservatório de ar comprimido e verificando se melhora o desempenho do sistema como um todo.

Outro desafio seria construir um levitador semelhante entretanto de forma cilíndrica, usando como corpo levitado uma bolinha que inclusive poderia gerar menos atrito nas laterais do levitador.

É pertinente ainda, o estudo da utilização de um rotâmetro para gerar o fluxo de ar pelas laterais e não pelo centro como está desenvolvido este trabalho, possibilitando um menor atrito e melhor deslizamento na estrutura.

Referências

- ALVES, J. L. L. *Instrumentação, controle e automação de processos*. [S.l.]: Livros Técnicos e Científicos, 2005. 15
- ARDUINO. *Arduino UNO Overview*. 2015. Disponível em: <<http://www.arduino.cc/en/main/arduinoBoardUno>>. 48
- ÅSTRÖM, K.; HÄGGLUND, T. Revisiting the ziegler–nichols step response method for pid control. *Journal of process control*, Elsevier, v. 14, n. 6, p. 635–650, 2004. 31, 32
- ÅSTRÖM, K. J.; PANAGOPOULOS, H.; HÄGGLUND, T. Design of pi controllers based on non-convex optimization. *Automatica*, Elsevier, v. 34, n. 5, p. 585–601, 1998. 30
- AWADALLAH, M.; MORCOS, M. Switch fault diagnosis of pm brushless dc motor drive using adaptive fuzzy techniques. *Energy Conversion, IEEE Transactions on*, IEEE, v. 19, n. 1, p. 226–227, 2004. 20
- BALBINOT, A.; BRUSAMARELLO, V. J. *Instrumentação e fundamentos de medidas*. [S.l.]: Grupo Gen-LTC, 2011. 15
- BISHOP, R. D. R. *Sistema de Controle Moderno*. [S.l.]: LTC, 2013. 30
- BOYLESTAD, R. L.; NASHELSKY, L. *Dispositivos eletrônicos e teoria de circuitos*. [S.l.]: Pearson Prentice Hall, 2004. v. 8. 50, 51
- BRUSSAARD, P. J.; GLAUDEMANS, P. W. M. *Shell-model applications in nuclear spectroscopy*. [S.l.]: North-Holland Pub. Co., 1977. 24
- CARDOSO, P. M. *Estudo, proposta e avaliação de novas metodologias de sintonia automática de controladores PID baseadas no ensaio do relé realimentado*. Tese (Doutorado) — Universidade Federal de Uberlândia, 2002. 31
- DATASHEET, I. P. *Document*. 2011. 41
- FERREIRA, J. M. M. *Introdução ao projecto com sistemas digitais e microcontroladores*. [S.l.]: FEUP Edições, 1998. 23
- HOLMES, D. G.; LIPO, T. A. *Pulse width modulation for power converters: principles and practice*. [S.l.]: John Wiley & Sons, 2003. v. 18. 24, 44
- JÚNIOR, F. G. F. et al. Implementação de controladores pid utilizando lógica fuzzy e instrumentação industrial. *VII Simpósio Brasileiro de Automação Inteligente-SBAI*, 2005. 26
- JURAN, J. M. Qualidade no século xxi. *HSM Management*, n. 3, p. 96–104, 1997. 12
- KENLOWE, P. *MegaDySC, Manual do fabricante*. KENLOWE Technologies. 2010. Manualventuinha. 43
- LATHI, B. P. *Sinais e Sistemas Lineares-2*. [S.l.]: Bookman, 2007. 16, 17

- MATHWORKS. *Matlab, the language of Technical Computing*. 2015. Disponível em: <<http://www.mathworks.com/products/matlab>>. 32, 33
- MCROBERTS, M. *Arduino básico*. [S.l.]: Novatec Editora, 2011. 47, 49
- NISE, N.; SILVA, F. R. da. *Engenharia de sistemas de controle*. [S.l.]: LTC, 2009. 27, 28, 29, 30
- OGATA, K.; MAYA, P. Á.; LEONARDI, F. *Engenharia de controle moderno*. [S.l.]: LTC, 1998. v. 3. 28
- PIRES, L. et al. *CONTROLE ANALÓGICO E DIGITAL DE UM MOTOR*. [S.l.]: Clube de Autores, 2012. 25, 26
- SANTANA, M. S. de; FERREIRA, J. M. de S.; SALAZAR, A. O. Módulo educativo de um levitador magnético. 2001. 34, 35, 36, 37, 38
- STEPHAN, R. M. Acionamento, comando e controle de máquinas elétricas. *Universidade Federal do Rio de Janeiro. Rio de Janeiro*, 2008. 22
- SUCH, R. N. Implementação de um controlador fuzzy em um sistema de refrigeração híbrido através de uma rede de sensores zigbee. *Trabalho de graduação, Faculdade de Tecnologia, Universidade de Brasília*, 2009. 43
- THOMAZINI, D.; ALBUQUERQUE, P. U. B. d. Sensores industriais—fundamentos e aplicações. *São Paulo*, v. 3, 2005. 15, 16, 17, 18, 41, 42
- URNS, S. R. *Fundamentos de Instrumentação Industrial e Controle de Processos*. [S.l.]: Bookman Editora, 2013. 40