

Lucas Bonetti Alves Rezende

**CONSUMO DE ENERGIA EM UM
PROTÓTIPO DE CARRO ELÉTRICO COM
DIFERENTES MÉTODOS DE SINTONIA
DE CONTROLADORES**

Manaus
2015

Lucas Bonetti Alves Rezende

**CONSUMO DE ENERGIA EM UM
PROTÓTIPO DE CARRO ELÉTRICO COM
DIFERENTES MÉTODOS DE SINTONIA
DE CONTROLADORES**

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 requi-
sitos necessários para a obtenção do grau de
Engenheiro.

Orientador Msc. Charles Luiz Silva de Melo

Manaus
2015

Lucas Bonetti Alves Rezende

**CONSUMO DE ENERGIA EM UM PROTÓTIPO DE
CARRO ELÉTRICO COM DIFERENTES MÉTODOS
DE SINTONIA DE CONTROLADORES**

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 requi-
sitos necessários para a obtenção do grau de
Engenheiro.

Aprovado em 19 de junho de 2015.

BANCA EXAMINADORA

Msc. Charles Luiz Silva de Melo
Orientador

Dr. Walter Andrés Vermehren
Valenzuela
Presidente da Banca

Msc. Almir Kimura Junior
Convidado 2

Dr. Luis Delfin Rojas Puron
Convidado 3

Manaus
2015

Agradecimentos

Primeiro gostaria de agradecer a Deus, por ter me abençoado e me dado discernimento nesses anos de faculdade.

Um agradecimento especial aos meus pais, Ana e Geraldo, pela educação que me deram e por sempre me apoiarem. Aos meus irmãos Gustavo e Rodrigo, e a todos os meus familiares que me apoiaram.

Um segundo agradecimento especial à minha namorada, Letícia Oliveira, que me acompanhou por quase todo esse tempo de faculdade, e que sempre me apoiou, me ajudou, e compartilhou comigo momentos incríveis, e tenho certeza que esse é só o começo.

Agradeço também à Universidade do Estado do Amazonas e sua equipe administrativa, à coordenação de Engenharia de Controle e Automação e todos os seus professores, nos quais muitos se tornaram verdadeiros amigos nessa jornada. Agradeço especialmente ao meu orientador, Msc. Charles Melo.

Por fim, e de maneira muito significativa, agradeço aos amigos que criei nessa jornada, e que levarei pra vida inteira: Anderson "Xupapito"Hiago, Alexandre Gomes, Dianny Machado, Eiji Maeda, Antonio Medeiros, Wellington "Biru"di Tommaso, Guilherme Linardi, Isabelle Stoco, Rafael Goulart, Izoneide Leite, Filipe Máximo, Nilteomar Gabay, Rodrigo Santa Rita, Renan Baima, Dilermano Ferreira, Talita Timbó, Kenny Caldas, Luanda Marinho, Érica Kido, Iana Lobato, Daniel Oliveira, Álvaro Belota, Cassio Rodrigues, Lana Libório, Luane Almeida, Cecília Nogueira, Iago Aguiar, Adriell Rudah, João Vítor, Michel Csasznick, Lucas Begnini e Desirée Rondon. E às minhas amigas de infância Nathali Carramanho e Lívia Porto.

Espero que nossa amizade continue além da faculdade!

Resumo

Utilizando-se um protótipo de carro elétrico, cuja velocidade será controlada por controlador PID, com motores de corrente contínua, microcontrolador, módulo de comunicação sem fio, um transdutor de posição angular (*encoder*, em inglês), para medição de velocidade e um sensor de corrente. O protótipo é utilizado para realizar testes, obter os dados necessários e fazer uma análise de como o método de sintonia do controlador PID influencia no seu consumo de energia e um sistema supervisório é implementado no *LabVIEW®*. Este trabalho tem como objetivo demonstrar, como as diferenças entre os métodos de sintonia de controladores PID podem influenciar no comportamento do protótipo e, consequentemente, no seu consumo de energia. Constatou-se que o controlador e método de sintonia mais eficiente, levando em consideração o consumos de energia, foi o PI com método de sintonia Ziegler-Nichols. Nos controladores PID, o método de sintonia mais eficiente foi o ISE, ficando em segundo lugar na comparação geral.

Palavras-chaves: controladores PID. setpoint. regime transitório. regime permanente. consumo de energia.

Abstract

Using an electric car prototype, whose speed is controlled by PID controller, with DC motors, microcontroller, wireless communication module, an encoder, for speed measuring and a current sensor. The prototype is used to perform tests, to obtain the necessary data and to analyse how the PID controller tuning method influences on its energy consumption and a supervisory system is implemented in LabVIEW®. This paper aims to demonstrate how differences between the PID controller tuning methods can influence the prototype's behavior and consequently on their energy consumption. It was found that the most efficient and controller tuning method, taking into account the power consumption, it has the IP Ziegler-Nichols tuning method. In PID controllers, the most efficient tuning method was the ISE, placing second in the overall comparison.

Key-words:PID controller. setpoint. steady state. transitional state. energy consumption.

Lista de ilustrações

Figura 1 – Diagrama de blocos com controlador PID	12
Figura 2 – Sistema de controle no universo	13
Figura 3 – Comparação entre um sistema simulado e um experimento com Controlador PID	14
Figura 4 – Diagrama de blocos do controlador PID	21
Figura 5 – Resposta padrão de um sistema de primeira ordem com tempo morto .	23
Figura 6 – Ações do campo magnético no estator e no rotor de um motor CC . . .	25
Figura 7 – Processo de Identificação de Sistemas	27
Figura 8 – Arduino Mega 2560	28
Figura 9 – Exemplo de funcionamento da técnica PWM	29
Figura 10 – Vista superior do chassi do protótipo de carro elétrico	30
Figura 11 – Vista lateral do chassi do protótipo de carro elétrico	31
Figura 12 – Rodas utilizadas no protótipo	32
Figura 13 – Parte interna da roda	32
Figura 14 – Motor CC utilizados no projeto	32
Figura 15 – Principais dimensões, em milímetros, do motor CC utilizado no projeto	33
Figura 16 – Suportes utilizados para fixação do motor ao chassi do protótipo . . .	33
Figura 17 – Principais dimensões, em milímetros, do suporte utilizado para fixação do motor ao chassi	33
Figura 18 – <i>Encoder</i> ENC01A-0J1216	34
Figura 19 – Encaixe do <i>encoder</i> com o motor, o suporte e a roda	34
Figura 20 – Drive de motor TB6612FNG	35
Figura 21 – Sensor de corrente ajustável ACS-712	36
Figura 22 – Motor CC simulado no Simulink	40
Figura 23 – Resposta do motor real e do motor simulado antes do início da estimativa	41
Figura 24 – Resposta do motor real e do motor simulado após 6 iterações	42
Figura 25 – Resposta do motor real e do motor simulado após 12 iterações	42
Figura 26 – Controlador PID do LabVIEW®	43
Figura 27 – Configuração da comunicação serial no sistema supervisório	45
Figura 28 – Escolha dos parâmetros do controlador PID no sistema supervisório .	45
Figura 29 – Escolha do nome e formato do arquivo externo	46
Figura 30 – Botões para iniciar e parar o teste	46
Figura 31 – Indicador da situação do teste	46
Figura 32 – Botão para ativar e desativar o protótipo	46
Figura 33 – Botão para parar a execução do programa	47
Figura 34 – Seleção da distância e velocidade dos percursos	47

Figura 35 – Informações sobre o percurso atual	47
Figura 36 – Informações gerais do teste	48
Figura 37 – Informações sobre a velocidade	48
Figura 38 – Informações sobre o consumo de energia	49
Figura 39 – Bloco <i>Match Pattern</i>	49
Figura 40 – Programação implementada para identificar na mensagem valores desejados	50
Figura 41 – Programação para dividir um número inteiro em centena, dezena e unidade	50
Figura 42 – <i>XBee® Series 1</i>	51
Figura 43 – <i>Shield</i> para o <i>Xbee Series 1®</i>	52
Figura 44 – <i>Explorer</i> para o <i>Xbee Series 1®</i>	52
Figura 45 – Resposta da aplicação de um degrau de magnitude 12 na função de transferência do motor	55

Lista de tabelas

Tabela 1 – Efeitos dos parâmetros do PID no processo	22
Tabela 2 – Parâmetros do PID de acordo com o primeiro método de Ziegler-Nichols	23
Tabela 3 – Parâmetros do PID de acordo com o método de Cohen-Coon	23
Tabela 4 – Parâmetros do controlador proporcional de acordo com as integrais do erro com relação ao tempo.	24
Tabela 5 – Parâmetros do controlador proporcional-integrativo de acordo com as integrais do erro com relação ao tempo.	24
Tabela 6 – Parâmetros do controlador proporcional-integral-derivativo de acordo com as integrais do erro com relação ao tempo.	25
Tabela 7 – Características principais do Arduino Mega 2560	28
Tabela 8 – Comparação entre o valor de corrente lido pelo sensor e pelo amperímetro	37
Tabela 9 – Constantes de proporcionalidade calculadas na calibração do sensor de corrente	38
Tabela 10 – Valores gerados pela identificação	43
Tabela 11 – Principais característica do <i>XBee® Series 1</i>	51
Tabela 12 – Percursos do teste de consumo de energia	52
Tabela 13 – Testes realizados	53
Tabela 14 – Características obtidas da Equação 5.2	55
Tabela 15 – Valores de K_P obtidos a partir da sintonia para o controlador P	56
Tabela 16 – Valores de K_P e T_i obtidos a partir da sintonia para o controlador PI	56
Tabela 17 – Valores de K_P , T_i e T_d obtidos a partir da sintonia para o controlador PID	56
Tabela 18 – Valores de consumo de energia do teste 6 (Controlador Proporcional-Integrativo e sintonia ISE)	57
Tabela 19 – Valores de consumo de energia do teste 7 (Controlador Proporcional-Integrativo e sintonia IAE)	58
Tabela 20 – Valores de consumo de energia do teste 8 (Controlador Proporcional-Integrativo e sintonia ITAE)	59
Tabela 21 – Valores de consumo de energia do teste 9 (Controlador Proporcional-Integrativo e sintonia Cohen-Coon)	60
Tabela 22 – Valores de consumo de energia do teste 10 (Controlador Proporcional-Integrativo e sintonia Ziegler-Nichols)	61
Tabela 23 – Valores de consumo de energia do teste 11 (Controlador Proporcional-Integral-Derivativo e sintonia ISE)	62
Tabela 24 – Valores de consumo de energia do teste 12 (Controlador Proporcional-Integral-Derivativo e sintonia IAE)	63

Tabela 25 – Valores de consumo de energia do teste 13 (Controlador Proporcional-Integral-Derivativo e sintonia ITAE)	64
Tabela 26 – Valores de consumo de energia do teste 14 (Controlador Proporcional-Integral-Derivativo e sintonia Cohen-Coon)	65
Tabela 27 – Resultados de consumo de energia do controlador Proporcional-Integrativo com método de sintonia ISE	85
Tabela 28 – Resultados de consumo de energia do controlador Proporcional-Integrativo com método de sintonia IAE	87
Tabela 29 – Resultados de consumo de energia do controlador Proporcional-Integrativo com método de sintonia ITAE	88
Tabela 30 – Resultados de consumo de energia do controlador Proporcional-Integrativo com método de sintonia Cohen-Coon	90
Tabela 31 – Resultados de consumo de energia do controlador Proporcional-Integrativo com método de sintonia Ziegler-Nichols	91
Tabela 32 – Resultados de consumo de energia do controlador Proporcional-Integral-Derivativo (PID) com método de sintonia ISE	93
Tabela 33 – Resultados de consumo de energia do controlador Proporcional-Integrativo com método de sintonia IAE	94
Tabela 34 – Resultados de consumo de energia do controlador Proporcional-Integrativo com método de sintonia ITAE	96
Tabela 35 – Resultados de consumo de energia do controlador Proporcional-Integrativo com método de sintonia Cohen-Coon	97

Sumário

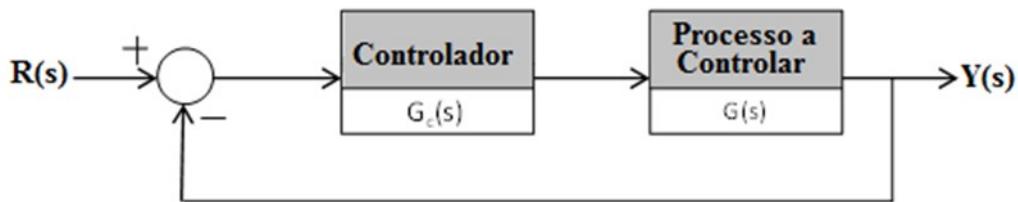
1	INTRODUÇÃO	12
1.1	Formulação do problema	13
1.2	Justificativa	14
1.3	Motivação	15
1.4	Objetivos	15
1.5	Metodologia	16
1.6	Organização do trabalho	17
2	TRABALHOS RELACIONADOS	18
2.1	<i>Comparison of some well-known PID tuning formulas</i>	18
2.2	<i>Maximum Sensitivity based PID Controller Tuning: A survey and comparison</i>	18
2.3	<i>Refine PID Tuning Rule Using ITAE Criteria</i>	19
2.4	<i>PID Controller Tuning for Glycerin Bleaching Process Using Well-Known Tuning Formulas - A Simulation Study</i>	20
3	REFERENCIAL TEÓRICO	21
3.1	Controlador PID	21
3.1.1	Sintonia de controladores PID	22
3.2	Motores de corrente contínua e sua modelagem	25
3.3	Identificação de sistemas	26
3.4	Sistemas microcontrolados	27
3.5	Conhecimentos referentes a circuitos elétricos	28
3.5.1	Pulse Width Modulation (PWM)	28
3.5.2	Potência e consumo de energia	29
4	MATERIAIS E MÉTODOS	30
4.1	Projeto e montagem do protótipo de carro elétrico	30
4.1.1	Projeto do protótipo de carro elétrico	30
4.1.2	Rodas, motores e suportes	31
4.1.3	Encoder ENC01A-0J1216	34
4.1.4	Drive de motor TB6612FNG	35
4.1.5	Sensor de corrente ACS-712	36
4.2	Programação do microcontrolador	38
4.3	Identificação da função de transferência do motor CC	40
4.4	Implementação do controlador PID	43

4.5	Criação do sistema supervisório	44
4.6	Comunicação entre o sistema supervisório e o protótipo	51
4.7	Sintonia do controlador PID	52
4.8	Testes de consumo de energia	52
5	RESULTADOS E DISCUSSÕES	54
5.1	Protótipo de carro elétrico	54
5.2	Função de transferência do motor CC	54
5.3	Sintonia do controlador PID	55
5.4	Testes de consumo de energia	57
5.4.1	Testes 1 a 5	57
5.4.2	Testes 6 a 14	57
5.4.3	Teste 15	65
5.5	Comparação dos resultados de consumo de energia	66
6	CONCLUSÕES	68
6.1	Trabalhos futuros	68
	REFERÊNCIAS	70
	APÊNDICE A – ALGORITMO IMPLEMENTADO NO MICOCONTROLADOR ARDUINO	72
	APÊNDICE B – SISTEMA SUPERVISÓRIO IMPLEMENTADO NO LABVIEW®	77
	APÊNDICE C – ALGORITMO IMPLEMENTADO NO MATLAB® PARA CÁLCULO DO K_p, T_i E T_d	78
	APÊNDICE D – ESQUEMA DE LIGAÇÃO ELÉTRICA DO PROTÓTIPO DE CARRO ELÉTRICO	83
	APÊNDICE E – FOTO DO PROTÓTIPO DE CARRO ELÉTRICO	84
	APÊNDICE F – RESULTADOS DOS TESTES DE CONSUMO DE ENERGIA	85

1 INTRODUÇÃO

Um problema de controle consiste, basicamente, em encontrar uma forma de afetar o funcionamento do sistema a fim de que ele se comporte de uma maneira esperada, conhecido na literatura como *setpoint*. Um sistema com realimentação e controlador é exemplificado na Figura 1. Os sistemas são montados dessa maneira porque, geralmente, não é possível mudar a estrutura funcional do processo a controlar. Assim, os controladores atuam antes do processo, como compensadores, onde recebem o valor do sinal de erro (diferença entre o valor da saída do sistema e o valor que deseja-se que ele opere), realizam operações matemáticas e enviam para a planta um valor que visa compensar o seu funcionamento, para que sua saída chegue cada vez mais próxima do valor desejado (ARAUJO, 2007).

Figura 1 – Diagrama de blocos com controlador PID



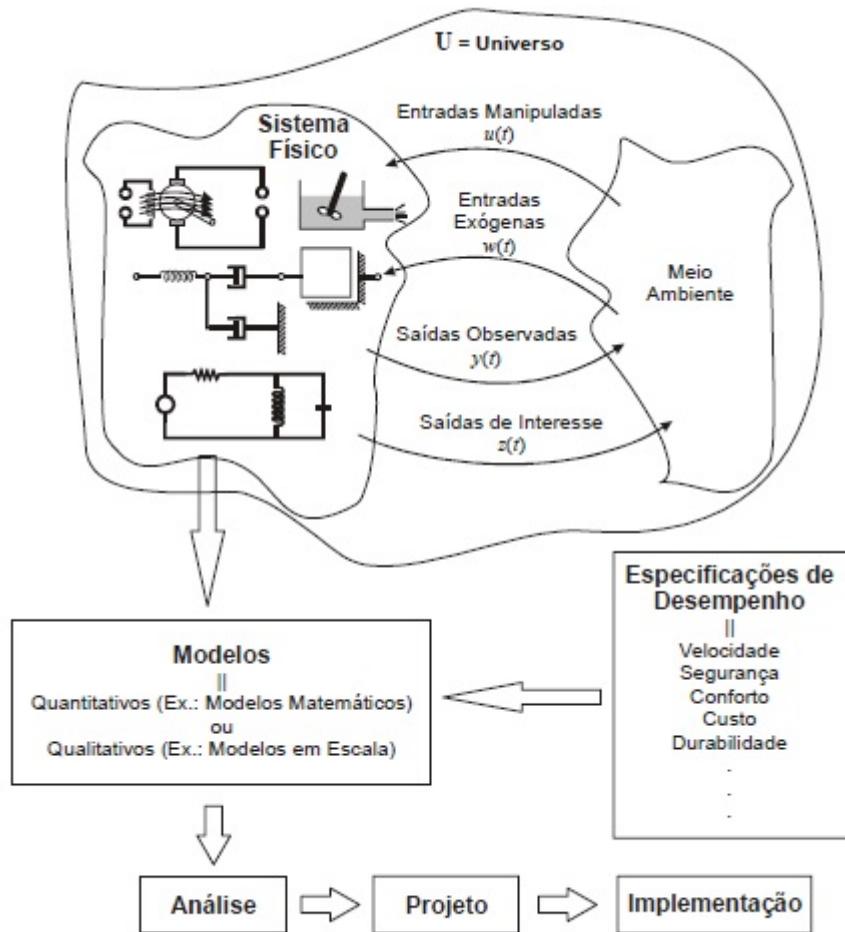
Fonte: (DORF; BISHOP, 2013)

Um sistema possui, portanto, as entradas manipuladas (onde o controlador atua), as entradas exógenas (relativas ao ambiente que o sistema está presente), as saídas observadas (um sistema pode possuir diversos fatores que caracterizam seu comportamento) e as saídas de interesse (aqueles que são de fato monitoradas e utilizadas pelo controlador, para que ele atue nas entradas manipuladas), como exemplificado na Figura 2.

Segundo Dorf e Bishop (2013), um dos controladores utilizados em muitos processos industriais é o do tipo proporcional-integral-derivativo (PID), e atribui sua popularidade, de modo geral, a dois fatores: seu bom desempenho em uma ampla faixa de operações e sua simplicidade funcional, que permite aos engenheiros operá-los de modo simples e direto.

Os três termos do controlador PID satisfazem a maioria das necessidades dos sistemas de controle, e o número de termos é suficientemente pequeno para permitir a sintonização manual por um operador, ou uso de controle adaptativo, métodos determinísticos ou o uso de heurísticas (KNOSPE, 2006).

Figura 2 – Sistema de controle no universo



Fonte: (ARAUJO, 2007)

1.1 Formulação do problema

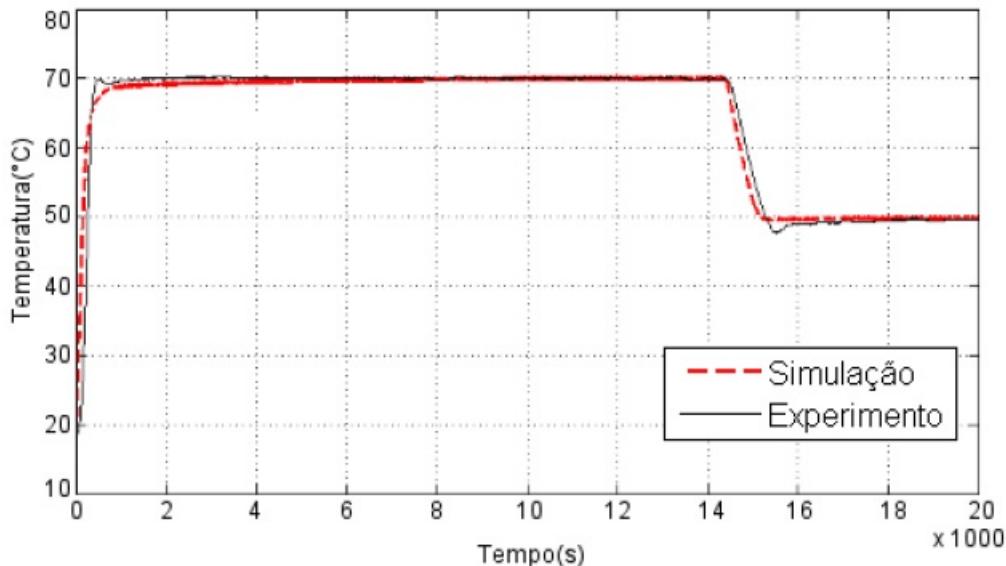
Nas últimas décadas, existem inúmeros estudos sobre os métodos de sintonia (processo de escolha dos ganhos) dos controladores PID. Uma das principais questões nesses estudos é como se comparar os diferentes métodos. De modo geral, aplica-se um degrau na entrada e compara-se fatores como o sobressinal (*overshoot*), tempo de subida (*rise time*) e tempo de acomodação (*settling time*). Em estudos mais avançados, considera-se também a robustez do sistema (TAN et al., 2006).

Porém, na vasta pesquisa bibliográfica realizada, não foi encontrado um estudo que compare a taxa de variação em torno do *setpoint* que diferentes métodos de sintonia de controladores PID podem ter, e como isso pode influenciar em outros fatores do sistema, como o seu consumo de energia.

1.2 Justificativa

Na Figura 2 observa-se que existem diversos fatores que podem influenciar no comportamento de um sistema. Em simulações computacionais, muitas vezes alguns desses fatores não são considerados, principalmente os fatores exógenos ao sistema. Esses fatores fazem com que o comportamento de um sistema simulado computacionalmente seja diferente de um sistema real, onde a estabilidade perfeita dificilmente é alcançada. Essa diferença é exemplificada na Figura 3, onde observa-se a diferença entre um sistema de controle de temperatura de um forno elétrico simulado e real (SOUZA; ROSSETTO; OLIVEIRA, 2009).

Figura 3 – Comparaçāo entre um sistema simulado e um experimento com Controlador PID



Fonte: (SOUZA; ROSSETTO; OLIVEIRA, 2009)

Para Souza, Rossetto e Oliveira (2009) as variações em torno do *setpoint* são consideradas normais e aceitáveis, uma vez que um sistema com controlador PID como o da Figura 3, “apresentou um desempenho compatível com os índices de desempenho como tempo de acomodação, tempo de subida e nível de sobressinal estabelecidos”.

O presente projeto pretende, portanto, verificar se a utilização de diferentes métodos de sintonia de controladores PID, para um mesmo sistema e mesma aplicação, pode influenciar para que as variações em torno do *setpoint* sejam maiores ou menores e verificar como isso pode influenciar em outros fatores do sistema, que não se restrinjam apenas ao tempo de acomodação, tempo de subida e sobressinal.

O consumo de energia será o parâmetro de comparação, procurando encontrar um método de sintonia de controladores PID mais eficiente nesse quesito.

1.3 Motivação

Em 2013, o Balanço Energético do Rio Grande do Sul informou que o consumo de energia mundial, em 2010, atingiu 13.205 bilhões de toneladas equivalentes de petróleo (tep). De acordo com as projeções do *International Energy Outlook* - IEO (2013), que espera um aumento no consumo de energia de 1,5% ao ano, no período de 2010 a 2040, a demanda de energia no mundo deve aumentar 56,44% nesse período, chegando a 20.656 milhões de tep em 2040 (CAPELETTO; MOURA, 2013).

Considerando-se apenas o cenário nacional, ainda de acordo com o Balanço Energético do Rio Grande do Sul, o consumo de energia do Brasil atingiu em 2012 a marca de 253.422 milhões de tep. Também de acordo com as projeções do IEO (2013), o consumo de energia brasileiro deve aumentar cerca de 2,1% ao ano, no período de 2010 a 2040, chegando a 453.490 milhões de tep no final desse período. O cenário se torna preocupante quando o consumo de energia do Brasil em 2012 já correspondia a 89,36% da Oferta Interna de Energia - OIE. Ou seja, é possível que se chegue a um gargalo entre o consumo e geração de energia nos próximos anos, caso a Oferta Interna de Energia não acompanhe o crescimento do consumo (CAPELETTO; MOURA, 2013).

Levando em consideração essas projeções, toda e qualquer alternativa de diminuição do consumo de energia é válida. Levando em consideração também a grande utilização dos controladores PID (cerca de 90% dos problemas de controle, segundo Acharya, Mitra e Halder (2013)), tem-se como motivação determinar um método de sintonia de controladores PID que proporcione um consumo de energia mais eficiente, para um mesmo sistema e mesma aplicação.

1.4 Objetivos

Construir um protótipo de carro elétrico cuja velocidade será controlada por controlador PID, que terá motores de corrente contínua, microcontrolador, módulo de comunicação sem fio, um transdutor de posição angular (*encoder*, em inglês), para medição de velocidade e um sensor de corrente. O protótipo será utilizado para realizar os testes, obter os dados necessários e fazer uma análise de como o método de sintonia do controlador PID influencia no seu consumo de energia. Este objetivo geral será alcançado quando todos seguintes objetivos específicos forem atingidos:

- Construção e montagem de um protótipo de carro elétrico;
- Identificação da função de transferência do sistema;
- Implementação de um controlador PID para que a velocidade do protótipo seja controlada;

- Realizar a sintonia do controlador PID pelos principais métodos de sintonia presentes na literatura;
- Criação do sistema supervisório;
- Comunicação do protótipo de carro elétrico com o sistema supervisório;
- Realizar os testes e medições do consumo de energia com o protótipo em operação;
- Realizar a aquisição de dados das variáveis do protótipo;
- Comparação dos dados obtidos.

1.5 Metodologia

Inicialmente, pesquisas foram realizadas para identificar, no mundo científico, o que existe relacionado com o tema proposto, e em que pontos o presente projeto se assemelha e se diferencia dos trabalhos já existentes.

Um referencial teórico foi criado, abordando os seguintes temas: controladores PID e sua sintonia, motores de corrente contínua, identificação e modelagem de sistemas, sistemas microcontrolados e conhecimentos referentes a circuitos elétricos (PWM, potência e consumo de energia).

Em seguida, foi realizada a construção e montagem do protótipo de carro elétrico, com quatro motores de corrente contínua. O protótipo é microcontrolado com um Arduino Mega 2560 e alimentado por uma bateria de 12 volts. A medição de velocidade do protótipo é realizada com um *encoder* ENC01A-0J1216. O interfaceamento entre o sinal PWM do Arduino, a tensão elétrica fornecida pela bateria e os motores é realizado pelo *drive* TB6612FNG. Foi implementado o sensor de corrente ACS-712 para medir a corrente elétrica consumida pelo sistema. Uma programação foi implementada no Arduino para que, tanto a informação lida dos sensores presente no protótipo como os dados provenientes do sistema supervisório sejam enviados e recebidos de maneira correta.

A identificação da função de transferência do motor de corrente contínua (que é o “processo a controlar” presente na Figura 1) foi realizada simulando um motor de corrente contínua no *Simulink®* e utilizando uma ferramenta de estimativa de parâmetros para que o *software* aproxime a resposta do motor simulado com resposta do motor real. Após a aproximação, as características do motor real são estimadas e sua função de transferência encontrada.

O controlador PID foi implementado por meio do *software* *LabVIEW®* e durante os testes práticos, a variável controlada foi a velocidade do protótipo. A sintonia dos parâmetros do controlador utilizará os métodos Ziegler-Nichols, ISE, IAE, ITAE e Cohen-Coon.

Para que o usuário possa determinar as velocidades, os pontos em que as mudanças ocorrem, assim como monitorar as variáveis do protótipo (como a velocidade, consumo de energia elétrica, etc.) um sistema supervisório foi criado no *LabVIEW®*.

Com o sistema supervisório em funcionamento pode-se, então, controlar e monitorar todas as variáveis necessárias para a realização dos testes, e salvá-las para que uma comparação possa ser feita entre os dados coletados. A comunicação do protótipo com o sistema supervisório é realizada com um módulo de comunicação sem fio *XBee®*, por meio de comunicação serial.

Os testes foram realizados da seguinte maneira: o protótipo de carro elétrico percorre trajetos padronizados com mudanças de velocidade em pontos previamente estabelecidos (por exemplo, 100 metros no total, com sub-trajetos de 10 metros, 30 metros, 20 metros e 40 metros e velocidades em cada sub-trajeto de 10 cm/s, 20 cm/s, 15 cm/s e 40 cm/s, respectivamente), e para um mesmo trajeto, diferentes variações do controlador PID (P, PI e PID) e diferentes métodos de sintonia desses controladores foram utilizados, um de cada vez. Assim, as condições de teste são sempre as mesmas, possibilitando uma comparação mais exata.

1.6 Organização do trabalho

Este trabalho está dividido em seis capítulos para seu melhor entendimento.

O capítulo 1 aborda uma breve introdução sobre sistemas de controle e consumo de energia, e conceitua o projeto por meio da justificativa e motivação. Também traz objetivos a serem alcançados e a metodologia que foi utilizada.

Já no capítulo 2 se dá a apresentação de trabalhos relacionados a essa monografia, encaixando-a no contexto científico, descrevendo as principais semelhanças e diferenças com os trabalhos já publicados.

Um referencial teórico, direto e aplicado, necessário ao desenvolvimento do projeto, é realizado no capítulo 3. Isso ocorre apresentando os tópicos relevantes por meio da visão de autores de obras consultadas e de trabalhos de pesquisa voltados para teorias e tecnologias aplicadas nesta monografia.

Em seguida, o capítulo 4 descreve como foi realizado o processo de desenvolvimento e construção do protótipo de carro elétrico, citando os materiais utilizados. Além disso, todas as etapas da metodologia são detalhadas, descrevendo-se o que foi necessário implementar para a realização dos testes e obtenção de dados de consumo de energia.

No capítulo 5 os resultados obtidos são apresentados e discutidos.

Por fim, no capítulo 6, os resultados são comparados, e uma conclusão é formulada, de acordo com o tema proposto, buscando-se determinar o método de sintonia de controlador PID mais eficiente, levando em consideração o consumo de energia obtidos nos testes.

2 TRABALHOS RELACIONADOS

Neste capítulo serão abordados alguns trabalhos científicos que, de alguma maneira, se assemelham com o tema desta monografia. O objetivo é encaixá-la no contexto científico, descrevendo as semelhanças e diferenças com os trabalhos já publicados.

2.1 Comparison of some well-known PID tuning formulas

“Comparação de algumas fórmulas de sintonia PID conhecidas” (TAN et al., 2006), em língua portuguesa.

Neste trabalho, as fórmulas de sintonia de controladores PID e seu desempenho são comparadas por meio de critérios baseados na rejeição de perturbações e robustez do sistema. Uma medida simples da robustez é definida e os ganhos integrais dos controladores PID mostraram-se uma boa medida para a rejeição de perturbações. A análise entre as fórmulas conhecidas de sintonia de controladores PID revela que a robustez deve situar-se entre 3 e 5 para ter um bom compromisso entre desempenho e performance.

Segundo o autor, um dos questionamentos que surge ao se analisar os trabalhos que abordam a sintonia de controladores PID é como comparar os métodos de sintonia. Uma resposta simples é usar a resposta ao degrau para sistemas de malha fechada e comparar o sobressinal, o tempo de subida, e o tempo de acomodação.

O objetivo final é encontrar um método de medir a robustez do sistema, para então comparar os diferentes métodos de sintonia de controladores PID. Os métodos considerados são Ziegler-Nichols, Cohen-Coon, controle de modelo interno (*internal model control*, IMC, em inglês), margem de ganho de fase (*gain-phase margin*, G-P, em inglês) e modelos baseados em integrais do erro (IAE, ITAE, ISE e ITSE).

A semelhança entre o trabalho de Tan et al. (2006) e o aqui proposto é a comparação entre as fórmulas de sintonia de controladores PID e a diferença é a utilização da robustez como critério de comparação, e não o consumo de energia.

2.2 Maximum Sensitivity based PID Controller Tuning: A survey and comparison

“Sintonia de controladores PID baseados na sensibilidade máxima: uma pesquisa e comparação” (JONES; THAM, 2006), em língua portuguesa.

O artigo considera a sensibilidade máxima, M_S , como uma das especificações de sistemas em malha fechada que satisfazem várias exigências de um método de sintonia de controladores PID útil e prático. Uma variedade de métodos de sintonia de controladores

PID que usam essa especificação foram desenvolvidos, sendo a maioria para fins de sintonia automática. Logo, o trabalho de Jones e Tham (2006) examina a aplicabilidade geral dos métodos de sintonia comparando sua performance em uma planta de referência.

Para o autor, a sensibilidade máxima, M_S , que representa o inverso da distância mínima entre a função de transferência e seu ponto crítico, no diagrama de Nyquist, se mostrou robusta quando utilizada como um parâmetro de sintonia. Logo, são criados novos métodos de sintonia baseados na sensibilidade máxima, dentre eles: métodos baseados na correlação, abordagens analíticas aproximadas, sintonia por iterações e otimização não-convexa.

A semelhança entre o trabalho de Jones e Tham (2006) e o aqui proposto é a utilização de apenas um critério (a sensibilidade máxima) para realizar a comparação entre os métodos sintonia do controlador PID e a utilização de apenas uma planta de referência. A diferença é que neste projeto, o critério de comparação será o consumo de energia.

2.3 Refine PID Tuning Rule Using ITAE Criteria

“Refinando a sintonia de controladores PID usando o critério ITAE” (AWOUDA; MAMAT, 2010), em língua portuguesa.

Este artigo discute uma abordagem para sintonizar os controladores PID utilizando o método de otimização, calculando analiticamente os ganhos proporcional K_P , o tempo integral T_i e tempo derivativo T_d para sistemas de primeira ordem com tempo morto (FOPDT, *first order plus dead time*). Um algoritmo é implementado no MATLAB® para encontrar os valores ideais dos ganhos que fazem o sistema atingir a maioria de seus requisitos, como reduzir o sobressinal, manter o sistema com uma resposta rápida, uma boa rejeição a distúrbios e manter a robustez. Os métodos de sintonia tradicionais são alterados, melhorando seus parâmetros, e tendo como objetivo final minimizar a integral do tempo multiplicado pelo erro absoluto (ITAE). Uma comparação entre os métodos de sintonia tradicionais e os métodos de sintonia propostos por Awouda e Mamat (2010) é realizada no MATLAB®.

Segundo Awouda e Mamat (2010), existem na literatura 219 métodos de sintonia para controladores PI e 381 métodos de sintonia para controladores PID, porém, pesquisas recentes indicam que 30% dos controladores instalados ainda são sintonizados de maneira manual, 30% dos sistemas mudam de comportamento durante o uso, 25% usam as definições de fábrica e 30% dos sistemas possuem algum tipo de problema. Os métodos de sintonia tradicionais utilizados foram Ziegler-Nichols e o AMIGO (*approximate M-constrained integral gain optimization*), proposto por Astrom e Hagglund (2006).

O autor conclui que o método de sintonia proposto tem comportamento melhor ou parecido com o método AMIGO e muito melhor que o método Ziegler-Nichols.

A semelhança entre o trabalho de Awouda e Mamat (2010) e esta monografia é a importância que ele dá aos métodos de sintonia, sua comparação e como a escolha do

melhor método pode influenciar diretamente no comportamento do sistema. A diferença é a criação de um novo método de sintonia baseado nos métodos tradicionais. Nesta monografia os métodos tradicionais serão comparados sem alterações em sua estrutura, tendo como critério de comparação o consumo de energia.

2.4 PID Controller Tuning for Glycerin Bleaching Process Using Well-Known Tuning Formulas - A Simulation Study

“Sintonia de controladores PID para o processo de branqueamento de glicerina usando fórmulas de sintonia conhecidas - um estudo-simulação” (KAMARUDDIN et al., 2009), em língua portuguesa.

O artigo aborda um estudo e simulação sobre os métodos de sintonia de controladores PID em um processo de branqueamento de glicerina. Os métodos utilizados foram Ziegler-Nichols, Cohen-Coon, IAE, ISE, ITAE. Em cada método os parâmetros analisados foram o sobressinal, tempo de subida e tempo de acomodação. Os parâmetros simulados foram testados na planta, que se comunica com o LabVIEW® e os resultados revelaram que o melhor método foi o ITAE.

O autor define o controlador PID como um controlador que considera o presente (com o elemento proporcional), o passado (com o elemento integrativo) e o futuro (com o elemento derivativo) do erro do sistema.

As semelhanças entre o trabalho de Kamaruddin et al. (2009) e o aqui proposto é a utilização e comparação dos métodos de sintonia ISE, ITAE, IAE, Ziegler-Nichols e Cohen-Coon, além da utilização de técnicas de identificação de sistemas para encontrar a planta do processo de branqueamento de glicerina. A diferença é que o autor utiliza parâmetros como o sobressinal, tempo de subida e tempo de acomodação como fatores de comparação, e não utiliza o consumo de energia, utilizado neste trabalho.

3 REFERENCIAL TEÓRICO

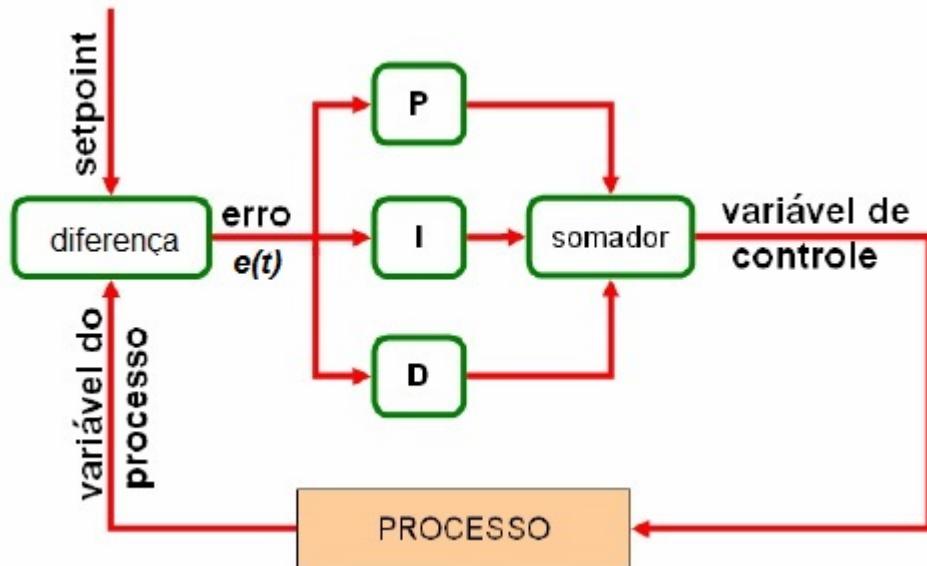
Neste capítulo será realizada uma breve abordagem teórica dos seguintes assuntos, que são necessários para o melhor desenvolvimento do presente trabalho: controlador PID e seus métodos de sintonia, motores de corrente contínua e sua modelagem, identificação de sistemas, sistemas microcontrolados e conhecimentos referentes a circuitos elétricos (PWM, potência e consumo de energia).

3.1 Controlador PID

Com os avanços da tecnologia, diversos modos avançados de esquemas de controle estão disponíveis. Porém, cerca de 90% dos controladores industriais ainda são implementados baseados no controlador PID e em suas variações, principalmente nos níveis mais baixos, como em motores, bombas e atuadores (ANG; CHONG; LI, 2005).

Na Figura 4 é ilustrado que esse controlador atua diretamente sobre o sinal de erro atuante ($e(t)$) proveniente da diferença entre a variável do processo (valor atual da saída do processo) e o *setpoint* (valor desejado da variável do processo).

Figura 4 – Diagrama de blocos do controlador PID



Fonte: (SOUZA; MELO; RODRIGUES, 2013)

Logo, segundo Castrucci, Bittar e Sales (2011), as três parcelas do controlador PID correspondem aos efeitos Proporcional, Integrador e Derivativo do sinal de erro atuante e o modelo matemático do controlador PID é

$$m(t) = K_p \left(e(t) + \frac{1}{T_i} \int_o^t e(t) dt + T_d \frac{d}{dt} e(t) \right) \quad (3.1)$$

sendo:

$m(t)$: sinal de saída do controlador, a variável de controle;

$e(t)$: sinal de entrada do controlador, o erro atuante;

K_P, T_i e T_d : parâmetros de ajuste do controlador.

Esse modelo matemático, após passar pela transformada de Laplace, se transforma na função de transferência do controlador PID, que é, ainda segundo Castrucci, Bittar e Sales (2011):

$$PID(s) = \frac{M(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (3.2)$$

A Tabela 1 contém informações de como cada parâmetro do PID influencia no comportamento geral do processo, em malha fechada. Nela, Ang, Chong e Li (2005) consideram como parâmetros K_P , K_I (que é a divisão entre o K_P e o T_i) e K_D (que é a multiplicação entre o K_P e o T_d):

Tabela 1 – Efeitos dos parâmetros do PID no processo

	Tempo de Subida	Sobressinal	Tempo de Acomodação	Erro Atuante
Aumentando K_P	Diminui	Aumenta	Aumenta Pouco	Diminui
Aumentando K_I	Diminui Pouco	Aumenta	Aumenta	Diminui Muito
Aumentando K_D	Diminui Pouco	Diminui	Diminui	Mínima Mudança

Fonte: (ANG; CHONG; LI, 2005)

3.1.1 Sintonia de controladores PID

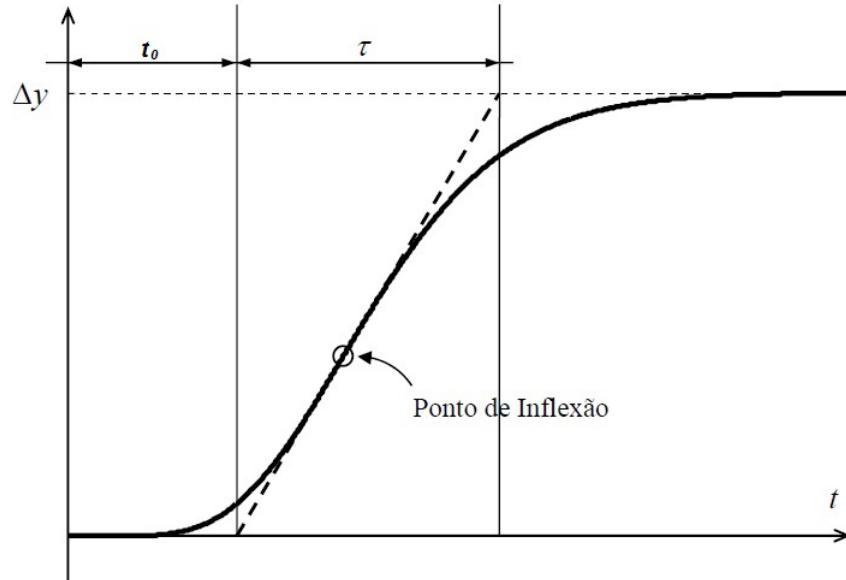
Ogata (2013) define que o processo de selecionar os parâmetros do controlador PID a fim de que o sistema tenha determinado comportamento é chamado de sintonia do controlador.

A maioria dos métodos de sintonia de controladores é baseada em sistemas de primeira ordem com tempo morto (FOPDT: *First-Order-Plus-Dead-Time*, em inglês), cuja forma básica encontra-se na Equação 3.3, uma vez que utilizam os valores do ganho do processo K , da constante de tempo τ e do tempo morto t_0 . O ganho do processo determina a aumento da resposta do sistema para cada uma unidade de variação da entrada, a constante de tempo é o tempo necessário para a saída do sistema em malha aberta, e sem controlador, atingir 63,2% do seu valor final e o tempo morto é o intervalo entre a aplicação de um sinal na entrada e o início da resposta do sistema (SMITH; CORRIPIO, 2005).

A resposta padrão do sistema FOPDT, onde aparecem os valores do tempo morto t_0 e da constante de tempo τ , encontra-se na Figura 5.

$$G_1(s) = \frac{K e^{-t_0 s}}{\tau s + 1} \quad (3.3)$$

Figura 5 – Resposta padrão de um sistema de primeira ordem com tempo morto



Fonte: (OGATA, 2013)

Considerando os sistemas FOPDT, os métodos de sintonia abordados serão: Ziegler-Nichols (1º método), Cohen-Coon, ISE, IAE e ITAE. Eles utilizam em seus cálculos os valores de K , t_0 e τ da Equação 3.3.

O primeiro método de Ziegler-Nichols determina os parâmetros do controlador PID de acordo com a Tabela 2.

Tabela 2 – Parâmetros do PID de acordo com o primeiro método de Ziegler-Nichols

	K_P	T_i	T_d
P	$\frac{\tau}{t_0}$	∞	0
PI	$0,9\frac{\tau}{t_0}$	$\frac{t_0}{0,3}$	0
PID	$1,2\frac{\tau}{t_0}$	$2t_0$	$0,5t_0$

Fonte: (OGATA, 2013)

O método de Cohen-Coon, por sua vez, determina os parâmetros do controlador PID de acordo com a Tabela 3 (KAMARUDDIN et al., 2009).

Tabela 3 – Parâmetros do PID de acordo com o método de Cohen-Coon

	K_P	T_i	T_d
P	$\frac{1}{K} \frac{\tau}{t_0} \left[1 + \frac{t_0}{\tau} \right]$	∞	0
PI	$\frac{1}{K} \frac{\tau}{t_0} \left[0,9 + \frac{t_0}{12\tau} \right]$	$\frac{t_0 \left[30 + 3\left(\frac{\tau}{t_0}\right) \right]}{9 + 20\left(\frac{\tau}{t_0}\right)}$	0
PID	$\frac{1}{K} \frac{\tau}{t_0} \left[\frac{16\tau + 30}{12\tau} \right]$	$\frac{t_0 \left[32 + 6\left(\frac{\tau}{t_0}\right) \right]}{13 + 8\left(\frac{\tau}{t_0}\right)}$	$\frac{4t_0}{11 + 2\left(\frac{\tau}{t_0}\right)}$

Fonte: (KAMARUDDIN et al., 2009)

Os outros três métodos abordados, por sua vez, são baseados em integrais do erro com relação ao tempo. Isso é possível porque o erro é uma função do tempo da duração da resposta, portanto a soma do erro em cada instante de tempo deve ser minimizada. As três integrais são, de acordo com Smith e Corripi (2005):

- ISE (*Integral of the square of the error* - Integral do quadrado do erro):

$$\int_0^{\infty} = |e^2(t)| dt \quad (3.4)$$

- IAE (*Integral of the absolute value of the error* - Integral do valor absoluto do erro):

$$\int_0^{\infty} = |e(t)| dt \quad (3.5)$$

- ITAE (*Integral of the time-weighted absolute value of the error* - Integral do tempo multiplicado pelo valor absoluto do erro):

$$\int_0^{\infty} = t |e(t)| dt \quad (3.6)$$

Lopez, Murrill e Smith (1967), baseado nessas integrais, desenvolveram as fórmulas presentes na Tabelas 4, 5 e 6.

Tabela 4 – Parâmetros do controlador proporcional de acordo com as integrais do erro com relação ao tempo.

Fórmula	ISE	IAE	ITAE
$K_P = \frac{a}{K} \left(\frac{\tau}{t_0} \right)^b$	$a = 1,411$	$a = 0,902$	$a = 0,490$

Fonte: (SMITH; CORRIPIO, 2005)

Tabela 5 – Parâmetros do controlador proporcional-integrativo de acordo com as integrais do erro com relação ao tempo.

Fórmulas	ISE	IAE	ITAE
$K_P = \frac{a}{K} \left(\frac{\tau}{t_0} \right)^b$	$a_1 = 1,305$	$a_1 = 0,984$	$a_1 = 0,859$
	$b_1 = -0,959$	$b_1 = -0,986$	$b_1 = -0,977$
$T_i = \frac{\tau}{a_2} \left(\frac{\tau}{t_0} \right)^{b_2}$	$a_2 = 0,492$	$a_2 = 0,608$	$a_2 = 0,674$
	$b_2 = 0,739$	$b_2 = 0,707$	$b_2 = 0,680$

Fonte: (SMITH; CORRIPIO, 2005)

Tabela 6 – Parâmetros do controlador proporcional-integral-derivativo de acordo com as integrais do erro com relação ao tempo.

Fórmula	ISE	IAE	ITAE
$K_P = \frac{a}{K} \left(\frac{\tau}{t_0} \right)^b$	$a = 1,495$ $b = -0,945$	$a = 1.435$ $b = -0,921$	$a = 1.357$ $b = -0.947$
$T_i = \frac{\tau}{a_2} \left(\frac{\tau}{t_0} \right)^{b_2}$	$a_2 = 1.101$ $b_2 = 0,771$	$a_2 = 0,878$ $b_2 = 0,749$	$a_2 = 0,842$ $b_2 = 0.738$
$T_d = a_3 \tau \left(\frac{\tau}{t_0} \right)^{b_3}$	$a_3 = 0.560$ $b_3 = 1.006$	$a_3 = 0,482$ $b_3 = 1.137$	$a_3 = 0,381$ $b_3 = 0.995$

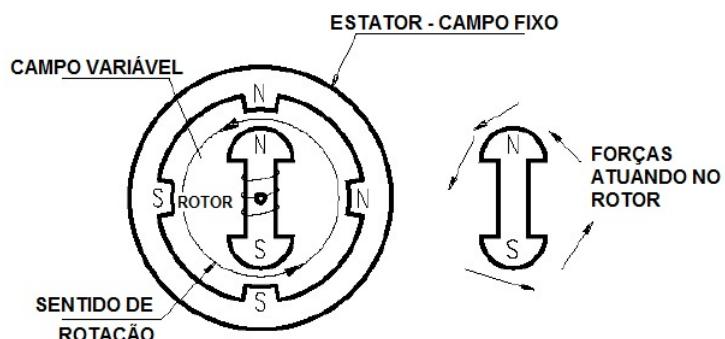
Fonte: (SMITH; CORRIPIO, 2005)

3.2 Motores de corrente contínua e sua modelagem

Dorf e Bishop (2013) define os motores de corrente contínua (motor CC) como um dispositivo que converte energia elétrica de corrente contínua em energia mecânica rotacional. São amplamente utilizados em processos em que é necessário ter velocidades de rotação variáveis e em que não é possível ou viável fazer modificações físicas no sistema, como a troca de relação entre as polias, caixas mecânicas de redução ou sistemas de fricção. Assim, o motor CC destaca-se pela simplicidade em se controlar a velocidade de rotação e o torque quando combinado com circuitos elétricos específicos, como o circuito Chopper comandado por um controle PWM.

O motor CC pode ser dividido em duas partes, sendo uma fixa e outra móvel. A primeira é o estator ou campo, que possui sapatas polares formada por lâminas de aço silício, por onde são enrolados fios condutores, formando bobinas. A segunda é o rotor ou armadura, que é ligado ao eixo de transmissão de movimento, e que fica dentro do estator. Como ilustrado na Figura 6, quando uma corrente elétrica contínua passa pelos condutores, um polo magnético é gerado no rotor. Os polos do rotor são atraídos pelo polo do estator, gerando uma força magnética que causa o deslocamento angular e o movimento rotacional (FUENTES, 2005).

Figura 6 – Ações do campo magnético no estator e no rotor de um motor CC



Fonte: (FUENTES, 2005)

A armadura do motor CC é modelada como tendo uma resistência constante R_a , em *ohms* (Ω), em série com uma indutância constante L_a , em *Henry* (H). Naturalmente, nem toda energia elétrica fornecida ao motor é transformada em energia mecânica, principalmente por causa das perdas por inércia e atrito. Assim, outras duas constantes são definidas: o momento de inércia total J , em $\text{kg} \cdot \text{m}^2$, que representa o momento de inércia total do rotor e da carga com relação ao eixo do motor, e o coeficiente de atrito viscoso B , em $\text{N.m}/\text{rpm}$, que representa todo o atrito viscoso do motor e da carga com relação ao eixo do motor. Por fim, define-se as constante do motor K_t e elétrica K_e , que são constantes de proporcionalidade que dependem de detalhes construtivos do motor e das características do campo magnético. Como, após considerar as perdas por inércia e atrito, a potência mecânica desenvolvida deve ser igual a potência elétrica, as duas constantes podem ser consideradas iguais (RICCI, 2011).

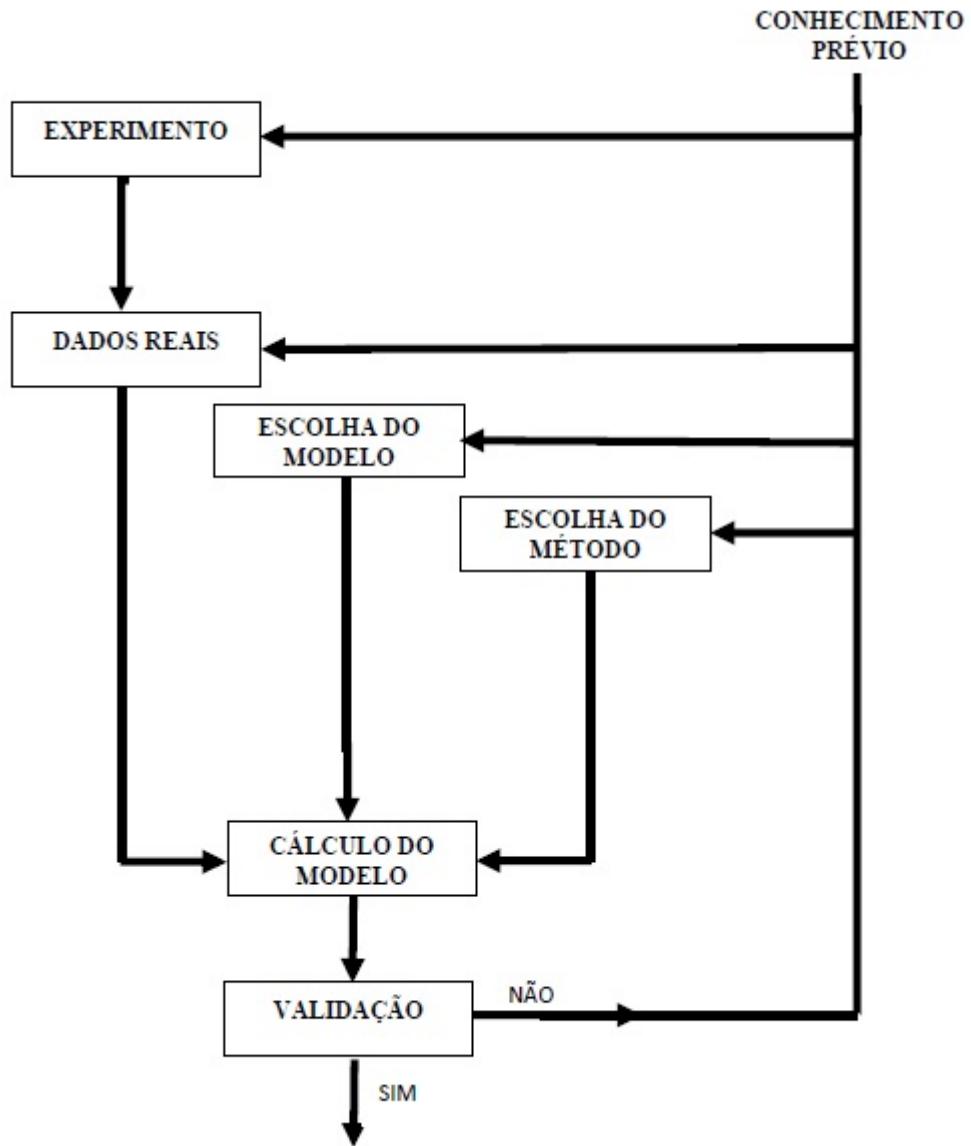
A obtenção da função de transferência tensão-velocidade (sendo a tensão elétrica em volts e a velocidade em rotações por minuto) do motor CC, finalizando sua modelagem, ocorre, segundo Ricci (2011) aplicando-se os valores de R_a , L_a , J , B e K na Equação 3.7.

$$G_1(s) = \frac{a(s)}{V_i(s)} = \frac{K_t}{(L_a s + R_a)(J s + B) + K_t K_e} \quad (3.7)$$

3.3 Identificação de sistemas

A obtenção de características do sistemas a partir de dados obtidos do processo é chamado de identificação de sistemas, cujo objetivo é obter modelos dinâmicos a partir de sinais medidos de entrada e saída. A identificação de sistemas abrange problemas na obtenção de modelos de sistemas *caixa preta* (cujos parâmetros necessários para obtenção do modelo matemático são totalmente desconhecidos) e *caixa cinza* (quando apenas alguns parâmetros são conhecidos). A confiabilidade do método escolhido para identificar o sistema depende, majoritariamente, dos dados reais obtidos, que serão utilizados durante o processo de identificação e validação. Quanto mais confiável os dados forem, mais confiável será a identificação e sua posterior validação (LJUNG, 1999). Ljung (1999) atribui os modelos considerados deficientes e que falharam na validação aos seguintes motivos: o método de identificação falhou na procura do melhor modelo de acordo com os critérios escolhidos, ou o critério não foi bem escolhido, ou o modelo escolhido não contém informações “suficientemente boas” sobre o sistema, e, por fim, os dados reais podem não possuir informações suficientes para que a identificação ocorra de acordo com o método e o modelo escolhido. Por isso, o método de identificação deve ser escolhido de maneira que os dados reais disponíveis possam fornecer informações confiáveis sobre o sistema. O processo completo de identificação de sistemas encontra-se na Figura 7.

Figura 7 – Processo de Identificação de Sistemas



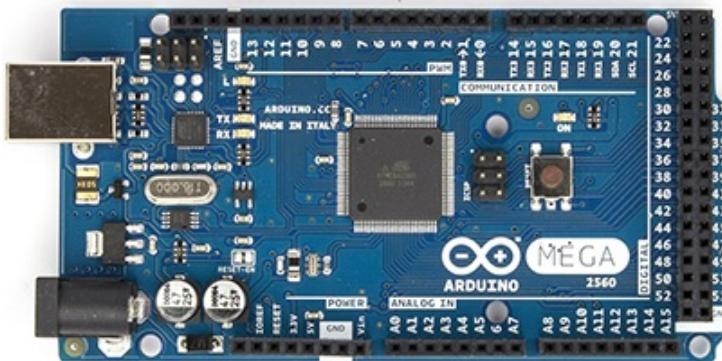
Fonte: (LJUNG, 1999)

3.4 Sistemas microcontrolados

Sistemas microcontrolados são comumente utilizados embarcados no produto que vão controlar ou acionar, e possuem em sua memória um programa compostos de algoritmos dedicados à aplicação em questão. Os microcontroladores atuais possuem, além da unidade de processamento (CPU), memória e circuitos auxiliares de entrada e saída (E/S) com funções específicas, como conversão analógico-digital e saídas digitais moduladas por largura de pulso (PWM) (STEPHAN, 2008).

O microcontrolador utilizado neste projeto foi o Arduino Mega 2560 (Figura 8), cujas principais características encontram-se descritas na Tabela 7.

Figura 8 – Arduino Mega 2560



Fonte: <http://www.arduino.cc/en/Main/ArduinoBoardMega2560>

Tabela 7 – Características principais do Arduino Mega 2560

Descrição	Característica
Microcontrolador	ATmega2560
Tensão Elétrica de Operação	5V
Tensão Elétrica de Entrada	7-12V
Pinos Digitais de E/S	54 (15 com função PWM disponível)
Portas Analógicas	16

Fonte: <http://www.arduino.cc/en/Main/ArduinoBoardMega2560>

A linguagem de programação do Arduino é essencialmente C/C++, com algumas alterações específicas para aplicações do microcontrolador.

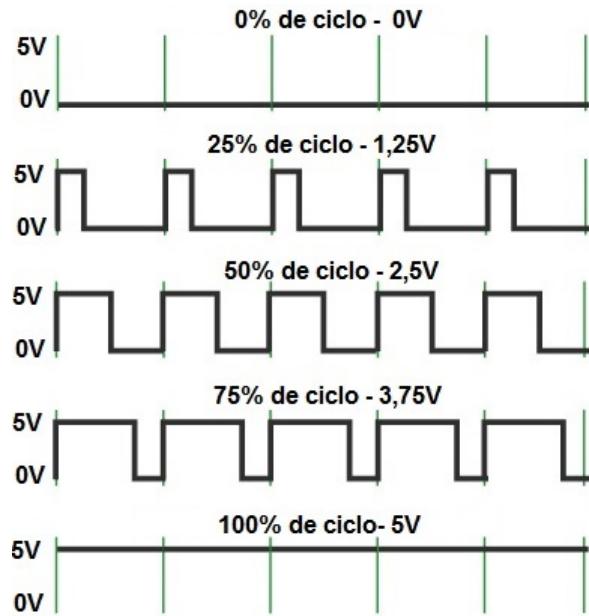
3.5 Conhecimentos referentes a circuitos elétricos

Dois conceitos fundamentais sobre circuitos elétricos serão abordados: PWM (*Pulse Width Modulation*, em inglês e modulação por largura de pulso, em português) e consumo de energia.

3.5.1 Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM) é uma técnica para controlar digitalmente o nível de um sinal analógico (como uma fonte de energia, por exemplo). Com a alteração do tempo de ciclo por um chaveamento de alta resolução (em que a transição entre os estados “ligado” e “desligado” pode ocorrer de maneira muito rápida) o tamanho de uma onda quadrada é modulada para o nível de sinal analógico desejado (BARR, 2001). Um exemplo é ilustrado na Figura 9, onde um sinal analógico de 5V é modulado por uma chave que permanece 0%, 25%, 50%, 75% e 100% do tempo ativada, gerando uma tensão elétrica de saída resultante de 0V, 1,25V, 2,5V, 3,75V e 5V.

Figura 9 – Exemplo de funcionamento da técnica PWM



Fonte: (BARR, 2001)

Nas saídas PWM do microcontrolador Arduino, o valor pode assumir valores de 0 a 5 volts, quando se varia, de 0 a 255 o valor presente na função utilizada na programação que ativa a saída PWM (“*analogWrite(255)*”, por exemplo). Ou seja, 0 na função resulta em uma tensão elétrica de 0V na porta (0% de ciclo), 127 em 2,5V (50% de ciclo) e 255 em 5V (100% de ciclo).

3.5.2 Potência e consumo de energia

Nilsson e Susan (2008) definem potência como a taxa de variação temporal do gasto ou absorção de energia, conforme a Equação 3.8.

$$P = \frac{dw}{dt} \quad (3.8)$$

onde P é a potência em *Watts*, w a energia em *Joules* e t o tempo em segundos. logo, 1W equivale a 1 Joule/segundo.

A potência também é definida como o produto entre a corrente em um elemento qualquer e a tensão elétrica em seus terminais, conforme a Equação 3.9.

$$P = v.i \quad (3.9)$$

O consumo de energia é uma relação entre a potência consumida e o tempo, em horas, que determinado sistema foi utilizado, de acordo com a Equação 3.10.

$$E = P.\Delta t \quad (3.10)$$

A unidade é o Wh (*Watt-hora*) ou kWh (*Kilowatt-hora*).

4 MATERIAIS E MÉTODOS

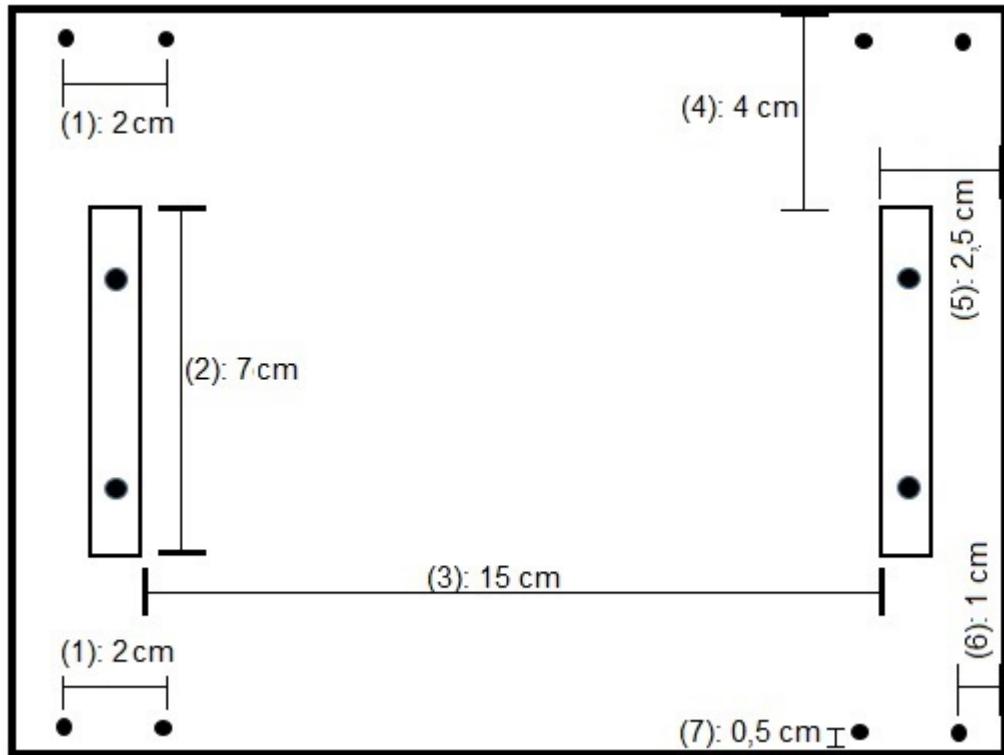
Neste capítulo serão abordados, de maneira detalhada, cada etapa da metodologia utilizada neste projeto, e quais foram os materiais utilizados em cada etapa.

4.1 Projeto e montagem do protótipo de carro elétrico

4.1.1 Projeto do protótipo de carro elétrico

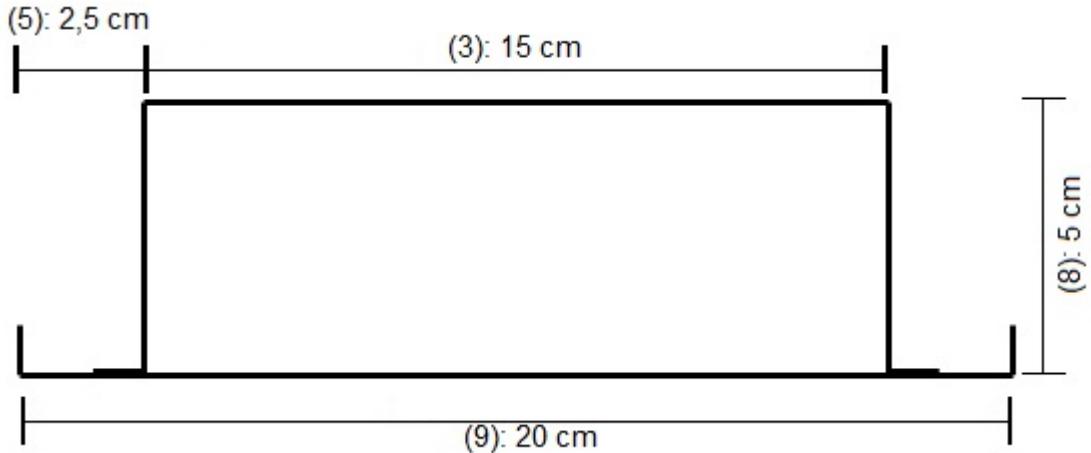
O protótipo de carro elétrico foi construído com chassi metálico feito a partir de uma chapa de eletrocalha. Ele possui 15 centímetros de largura e 20 centímetro de comprimento. Uma segunda peça metálica foi adicionada para servir de suporte para o microcontrolador. A bateria fica coberta por essa segunda peça, na parte de baixo do microcontrolador. Nas Figuras 10 e 11 tem-se dois croquis das vistas superior e lateral do chassi protótipo.

Figura 10 – Vista superior do chassi do protótipo de carro elétrico



Fonte: Autor

Figura 11 – Vista lateral do chassi do protótipo de carro elétrico



Fonte: Autor

Sendo:

- (1): Distância entre os furos de fixação do suporte do motor;
- (2): Comprimento da base do suporte do microcontrolador;
- (3): Distância entre as bases do suporte do microcontrolador;
- (4): Distância entre a base do suporte do microcontrolador e a lateral do protótipo;
- (5): Distância entre a base do suporte do microcontrolador e a frente do protótipo;
- (6): Distância entre o furo de fixação do suporte do motor e a frente do protótipo;
- (7): Distância entre o furo de fixação do suporte do motor e a lateral do protótipo;
- (8): Altura do suporte do microcontrolador;
- (9): Comprimento total do protótipo.

No projeto todas as distâncias foram calculadas para que os componentes, principalmente os mais pesados, como a bateria, fiquem sempre centralizados com relação ao centro de massa do protótipo, evitando assim que um dos motores seja sobrecarregado.

4.1.2 Rodas, motores e suportes

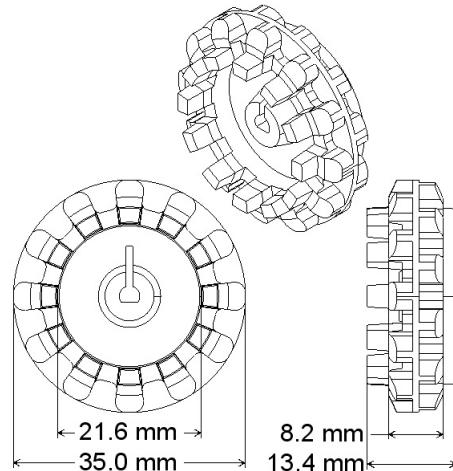
As rodas utilizadas (Figura 12) são fabricadas pela *Pololu Robotics & Electronics®*. Sua parte externa, de borracha, tem 42 milímetros de diâmetro e 19 milímetros de espessura. A parte interna (Figura 13), de plástico, tem 35 milímetros de diâmetro e 13,4 milímetros de espessura. É nessa parte interna que se encontram os 12 “dentes” que são utilizados na leitura do *encoder*.

Figura 12 – Rodas utilizadas no protótipo



Fonte: *Pololu Robotics & Electronics* (<https://www.pololu.com/product/1090>)

Figura 13 – Parte interna da roda



Fonte: *Pololu Robotics & Electronics®* (<https://www.pololu.com/product/1090>)

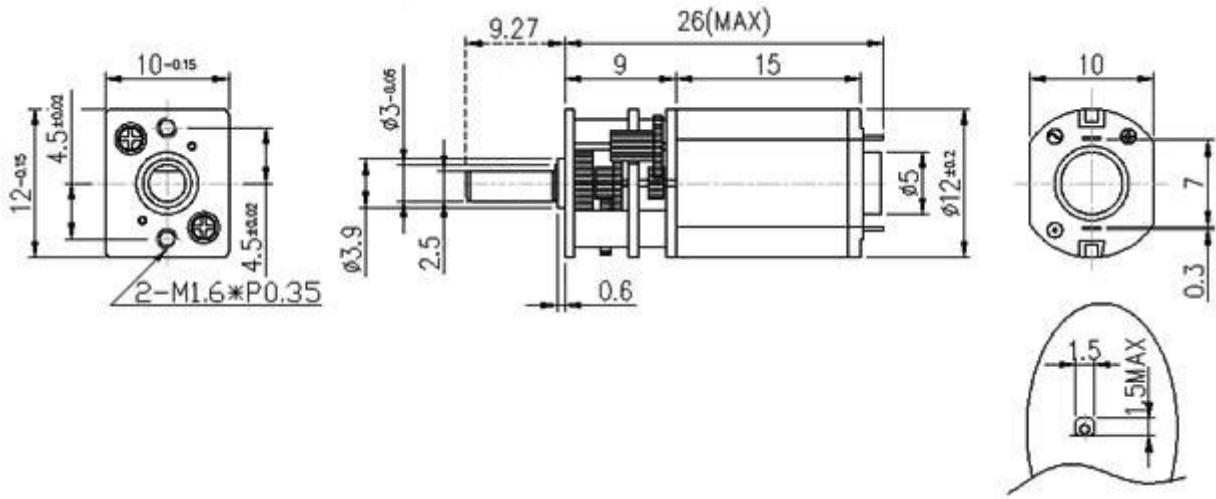
Os motores de corrente contínua utilizados também são fabricados pela *Pololu Robotics & Electronics®* (Figura 14 e principais dimensões, em milímetros, na Figura 15). Possuem uma caixa de redução de 1:100, fixada ao eixo interno do motor. Ou seja, cada 100 rotações do eixo interno gera 1 rotação do eixo externo. Sua rotação máxima é, no eixo externo, 300 rpm, quando aplicado em seus terminais a tensão elétrica máxima de operação de 12 volts.

Figura 14 – Motor CC utilizados no projeto



Fonte: *Pololu Robotics & Electronics®* (<https://www.pololu.com/product/1101>)

Figura 15 – Principais dimensões, em milímetros, do motor CC utilizado no projeto



Fonte: *Pololu Robotics & Electronics®* (<https://www.pololu.com/product/1101>)

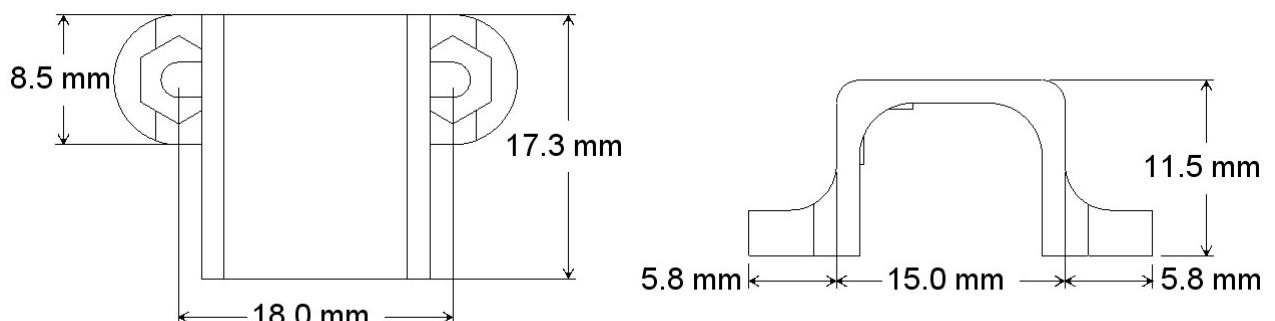
A fixação do motor ao chassis do protótipo é realizada por meio de suportes plásticos (Figura 16 e principais dimensões, em milímetros, na Figura 17) que encaixam-se perfeitamente ao motor, e cuja fixação à parte inferior do chassis é realizada com a utilização de parafusos e porcas.

Figura 16 – Suportes utilizados para fixação do motor ao chassis do protótipo



Fonte: *Pololu Robotics & Electronics®* (<https://www.pololu.com/product/1189>)

Figura 17 – Principais dimensões, em milímetros, do suporte utilizado para fixação do motor ao chassis

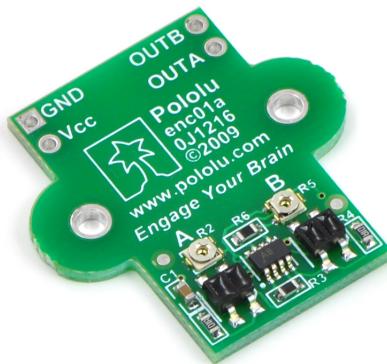


Fonte: *Pololu Robotics & Electronics®* (<https://www.pololu.com/product/1189>)

4.1.3 Encoder ENC01A-0J1216

A medição da velocidade do protótipo é realizada pelo transdutor de posição angular, ou *encoder*, em inglês, ENC01A-0J1216 fabricado pela *Pololu Robotics & Electronics®* (Figura 18). Ele é um *encoder* de quadratura, ou seja, possui dois sensores infravermelhos de reflectância que reconhecem o movimento dos 12 “dentes” presentes na parte plástica da roda. Os sensores são espaçados de forma a fornecer sinais defasados aproximadamente em 90°, permitindo que a direção de rotação seja determinada e fornecendo 48 contagens por volta da roda, com resolução de 4 contagens por dente e resolução linear de menos de 3 milímetros.

Figura 18 – *Encoder* ENC01A-0J1216



Fonte: *Pololu Robotics & Electronics®* (<https://www.pololu.com/product/1217>)

O *encoder* deve ser alimentado eletricamente nos pinos Vcc e GND com uma tensão elétrica entre 4,5 e 5,5 V e os pinos OUTA e OUTB, que contém o sinal de saída, devem ser conectados ao microcontrolador.

A maneira como o *encoder*, o motor, o suporte e a roda se encaixam é ilustrado na Figura 19.

Figura 19 – Encaixe do *encoder* com o motor, o suporte e a roda

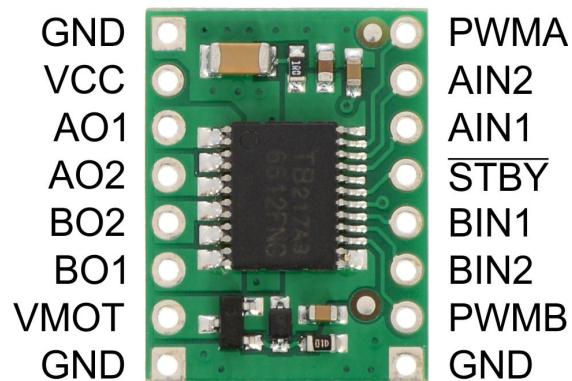


Fonte: *Pololu Robotics & Electronics®* (<https://www.pololu.com/product/1217>)

4.1.4 Drive de motor TB6612FNG

Como o valor de tensão elétrica na saída PWM do Arduino tem valor máximo de 5V, um interfaceamento deve ser realizado para que os motores CC sejam alimentados com os 12V fornecidos pela bateria. O responsável por isso é o *drive* de motor TB6612FNG (Figura 20), fabricado pela *Pololu Robotics & Electronics®*. Esse dispositivo consegue controlar até dois motores ao mesmo tempo e possui uma ponte H baseada em MOSFET, o que permite também controlar o sentido de rotação de cada um dos motores.

Figura 20 – Drive de motor TB6612FNG



Fonte: *Pololu Robotics & Electronics®* (<https://www.pololu.com/product/713>)

Sua ligação física com o microcontrolador, os motores e a bateria se dá utilizando os pinos descritos a seguir, tendo como referência inicial a parte superior esquerda da Figura 20 e percorrendo-a no sentido anti-horário:

- GND: Conectado ao terminal terra do microcontrolador;
- VCC: Conectado ao terminal 5V do microcontrolador;
- AO1: Conectado ao polo negativo do motor 1;
- AO2: Conectado ao polo positivo do motor 1;
- BO2: Conectado ao polo positivo do motor 2;
- BO1: Conectado ao polo negativo do motor 2;
- VMOT: Conectado ao polo positivo da fonte de energia utilizada para alimentar os motores(bateria);
- GND: Conectado ao polo negativo da fonte de energia utilizada para alimentar os motores(bateria);
- PWMA: Conectado a um pino PWM do microcontrolador;
- AIN2: Conectado a um pino digital do microcontrolador;

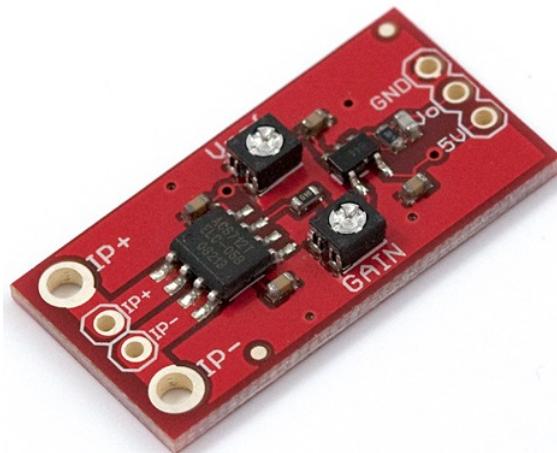
- AIN1: Conectado a um pino digital do microcontrolador;
- STBY: Conectado a um pino digital do microcontrolador;
- BIN1: Conectado a um pino digital do microcontrolador;
- BIN2: Conectado a um pino digital do microcontrolador.

São os pinos AIN2, AIN1, BIN1 e BIN2 que permitem a escolha do sentido de rotação do motor, alterando-se qual pino de cada par (“A” ou “B”) recebe sinal lógico “ligado” do pino digital do microcontrolador. Já o pino STBY permite, também alterando-se o sinal lógico que ele recebe do microcontrolador, ligar ou desligar os dois motores que estão ligados ao *drive*. Como o projeto possui quatro motores são necessários dois *drivers* para que eles funcionem.

4.1.5 Sensor de corrente ACS-712

Para que se tenha valores do consumo de energia do protótipo é necessário medir a corrente consumida pelo sistema. O dispositivo usado para fazer essa medição é o sensor de corrente ajustável ACS-712 (Figura 21), fabricado pela *SparkFun Electronics®*. Sua saída é uma tensão elétrica analógica proporcional a corrente medida, e um amplificador operacional ajustável permite um ganho 4,27 a 47 vezes no valor da saída, permitindo a leitura de correntes pequenas. A corrente que passa pelo sensor é medida por efeito Hall.

Figura 21 – Sensor de corrente ajustável ACS-712



Fonte: *SparkFun Electronics®* (<https://www.sparkfun.com/products/8883>)

O sensor deve ser alimentado eletricamente nos pinos GND (terra) e 5V. Já o pino de saída, *Vo*, deve ser conectado a um dos pinos de entrada analógica do microcontrolador.

A sua calibração foi realizada medindo a corrente consumida pelo sistema com um amperímetro ET-2030 fabricado pela *Minipa®* e, ao mesmo tempo, monitorando a leitura analógica realizada pelo microcontrolador da saída do sensor. Porém, nos testes práticos,

constatou-se que a tensão elétrica na saída do sensor, ao contrário do informado pelo fabricante, não aumentava proporcionalmente conforme a corrente que passa pelo sensor aumenta.

Ou seja, não é possível obter apenas uma constante de proporcionalidade pelo qual o valor de tensão elétrica da saída do sensor deve ser multiplicado para que se obtenha um valor de corrente igual ao lido pelo amperímetro. As constantes utilizadas, para cada faixa de tensão na saída do sensor, encontram-se na Tabela 9. Na Tabela 8 encontra-se um comparativo entre o valor de corrente lido pelo sensor e pelo amperímetro, após a calibração, quando diferentes valores de tensão elétrica são aplicados nos motores CC. A maior variação verificada foi de 0,03A, quando o sensor de corrente mediou 0,48A e o amperímetro 0,51A.

Tabela 8 – Comparaçāo entre o valor de corrente lido pelo sensor e pelo amperímetro

Tensão elétrica aplicada nos motores (em volts)	Sensor de Corrente (em Amperes)	Amperímetro (em Amperes)
0,47	0,18	0,17
0,94	0,21	0,21
1,41	0,27	0,26
1,88	0,31	0,31
2,35	0,35	0,37
2,82	0,41	0,41
3,29	0,44	0,46
3,76	0,47	0,49
4,23	0,5	0,52
4,7	0,53	0,54
5,17	0,53	0,56
5,64	0,56	0,57
6,11	0,56	0,57
6,58	0,56	0,57
7,05	0,54	0,56
7,52	0,53	0,55
8	0,5	0,52
8,47	0,48	0,51
8,94	0,45	0,47
9,41	0,44	0,44
9,88	0,39	0,40
10,35	0,35	0,36
10,82	0,32	0,33
11,29	0,29	0,3
12	0,29	0,28

Fonte: Autor

Tabela 9 – Constantes de proporcionalidade calculadas na calibração do sensor de corrente

Faixa de tensão elétrica (em volts)	Constante de Proporcionalidade	Faixa de Corrente Lida (em Amperes)
Menor que 2,56	0	0
2,56 e 2,61	0,065134	0,16 e 0,17
2,62 a 2,64	0,072243	0,18 a 0,20
2,65 a 2,67	0,082397	0,21 e 0,22
2,68 a 2,69	0,092936	0,23 a 0,25
2,70 a 2,72	0,102564	0,27 e 0,28
2,73 a 2,75	0,109090	0,29 e 0,30
2,76 a 2,79	0,1151079	0,31 a 0,33
2,80 a 2,82	0,1174377	0,34 e 0,35
2,83 a 2,84	0,1245551	0,36 e 0,37
2,85 a 2,86	0,1385435	0,39 e 0,4
2,87 a 2,89	0,145833	0,41 a 0,43
2,90 a 2,95	0,1535836	0,44 e 0,45
2,96 a 2,99	0,161616	0,47 a 0,49
3,00 a 3,02	0,169435	0,5 a 0,52
3,03 a 3,05	0,17763	0,53 a 0,55
3,06 a 3,07	0,185667	0,56 a 0,58

Fonte: Autor

4.2 Programação do microcontrolador

Uma programação foi implementada para que o microcontrolador Arduino possa obter os dados do *encoder* e do sensor de corrente, enviá-los, por meio de comunicação serial, para o sistema supervisório, e receber dele, após a atuação do controlador PID, o valor de PWM que corresponde à tensão elétrica que deve ser aplicado nos motores. A programação completa encontra-se no Apêndice A. O algoritmo foi dividido em 8 partes, cujas principais características são:

- Parte 1: CONFIGURAÇÃO DAS PORTAS DO MICROCONTROLADOR:
 - Determinação das portas utilizadas no Arduino: primeiro, as sete utilizadas por cada *driver* de motor (dois pares para determinar o sentido de rotação de cada motor, duas para determinar o valor de PWM, também de cada motor, e uma porta para determinar se os dois motores conectados àquele *driver* estão ligados ou desligados), e depois as 2 portas utilizadas pelo *encoder*.
- Parte 2: DECLARAÇÃO DAS VARIÁVEIS:
 - As variáveis que vão receber o valor de PWM, da leitura do sensor de corrente e do *encoder* são declaradas.

- Parte 3: CONFIGURAÇÃO DE ENTRADA E SAÍDA:
 - Determina-se que todas as portas utilizadas pelo *driver* serão portas de “saída” de informações.
- Parte 4: CONFIGURAÇÃO DA COMUNICAÇÃO SERIAL:
 - A comunicação serial do Arduino é iniciada com uma taxa de transmissão de 9600 *bits* por segundo.
- Parte 5: CONFIGURAÇÃO DO *ENCODER*:
 - As portas que vão receber as informações do *encoder* ficam prontas para receber dados a qualquer momento.
- Parte 6: RECEBIMENTO DE DADOS DO *LABVIEW®*:
 - Como o recebimento dos dados seriais pelo Arduino se dá um *byte* de cada vez e o valor recebido do sistema supervisório é, no máximo, um número com três dígitos (por exemplo, “123”), quando existem dados para serem recebidos, três leituras seguidas são realizadas, uma para cada dígito do número: o primeiro recebido é a unidade, o segundo é a dezena e o terceiro é a centena (para que os dígitos cheguem dessa maneira no Arduino, eles devem ser enviados dessa maneira pelo sistema supervisório). Para juntá-los, a centena é multiplicada por cem, a dezena multiplicada por dez, e as duas são somadas com a unidade.
- Parte 7: DADOS ENVIADOS PARA AS PORTAS DO MICROCONTROLADOR:
 - É determinado o nível lógico das portas conectadas aos *drivers*. Uma das portas de cada um dos pares que determinam a rotação dos motores recebe nível lógico “alto” e a outra nível lógico “baixo”. O valor de PWM que foi recebido do sistema supervisório é enviado para os quatro motores do protótipo, para que todos tenham a mesma rotação. O valor de PWM, na programação, varia entre 0 e 255, representando um valor de 0 a 5V na saída do microcontrolador. Após o envio dos dados, é realizada uma leitura do *encoder* e seu valor é zerado para a próxima leitura.
- Parte 8: LEITURA DO SENSOR DE CORRENTE:
 - O valor de tensão elétrica do pino de saída do sensor de corrente é lido, e as constantes de proporcionalidade da Tabela 9 são implementadas. Com isso, obtém-se a corrente consumida pelo sistema. Esse valor é multiplicado por cem para ser enviado para o sistema supervisório sem casas decimais.

- Parte 9: ENVIO DE DADOS PARA O LABVIEW®:

- Os valores da contagem do *encoder* e da corrente são enviados pela porta serial. Como eles são enviados por comunicação serial, e em uma única mensagem, eles são delimitados pelas letras “a” e “b”. Elas serão utilizadas pelo sistema supervisório para determinar em que ponto da mensagem encontra-se cada valor. Por fim, um atraso de 90 milissegundos ocorre até a próxima execução do programa, para que o sistema supervisório realize suas operações e envie novos valores para o microcontrolador. Logo, a quantidade de troca de informações entre o sistema supervisório e o microcontrolador depende diretamente desse tempo.

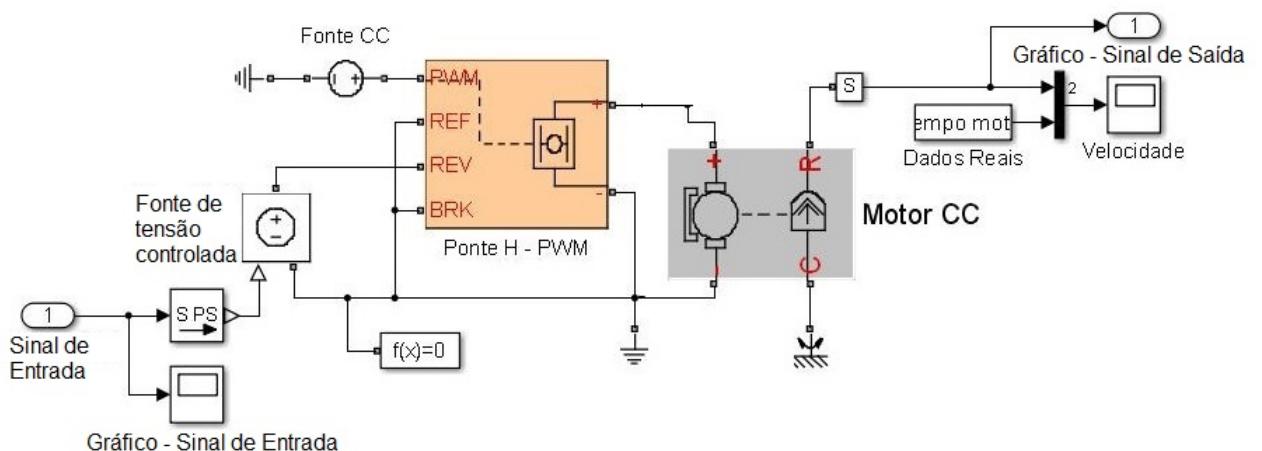
4.3 Identificação da função de transferência do motor CC

De acordo com a teoria abordada na seção 3.3, três informações devem ser previamente conhecidas para que seja possível realizar a identificação de um sistema: o modelo do sistema real, o método de identificação mais apropriado e os dados do sistema real.

O modelo é a função de transferência de um motor CC presente na Equação 3.7. É necessário que se determine, por meio de técnicas de identificação de sistemas, os valores de R_a , L_a , J , B e K , que caracterizam o motor.

O método utilizado consiste em simular, por meio do *Simulink®*, um motor CC (Figura 22), cujos valores característicos R_a , L_a , J , B e K podem ser alterados, e cuja ligação elétrica seja idêntica ao do motor real (com uma fonte de energia que alterne sua saída por meio de uma técnica PWM) e alterar seus valores característicos até que a resposta simulada se aproxime da resposta do motor real. As alterações são feitas por uma ferramenta de estimativa de parâmetros (*Parameter Estimation*), que realiza diversas iterações até aproximar as duas respostas (*TUREVSKIY, 2013*).

Figura 22 – Motor CC simulado no Simulink



Fonte: Autor

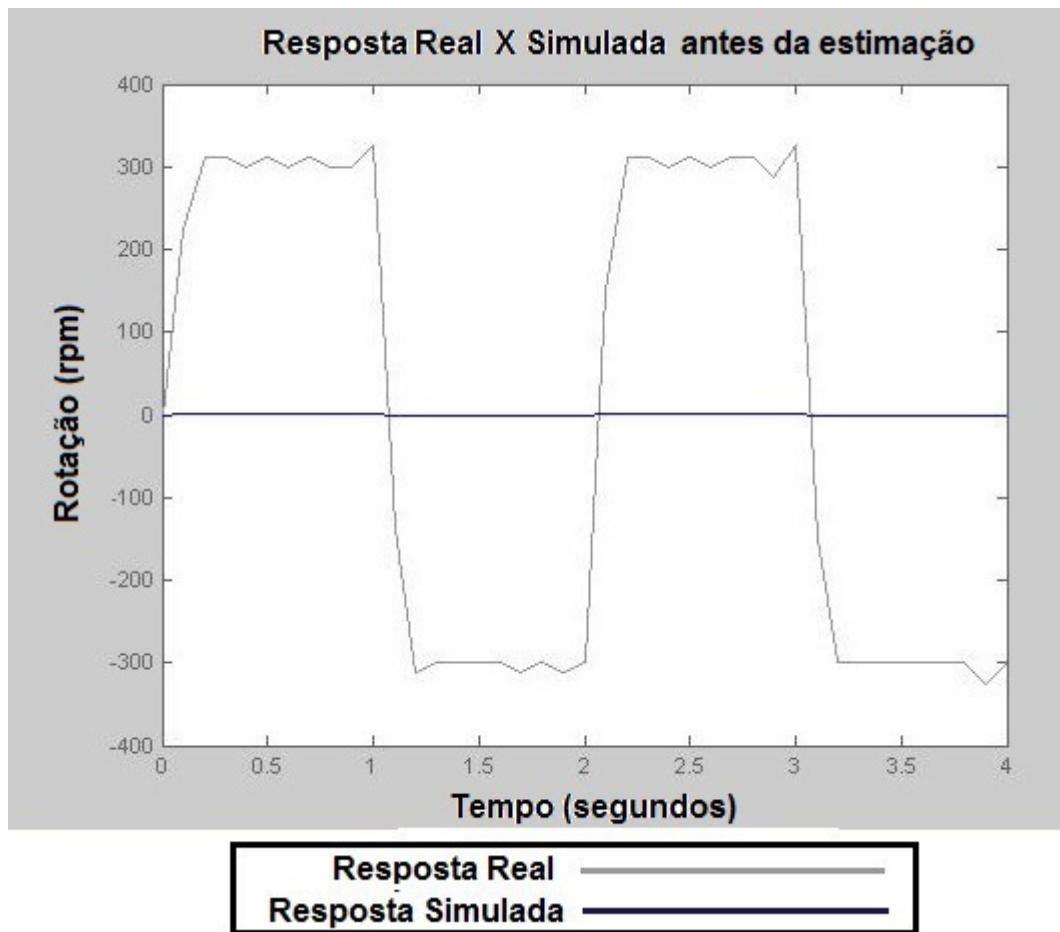
Os dados do motor real foram obtidos da seguinte maneira: a tensão elétrica aplicada em seus terminais foi alternada, a cada segundo, entre +12V e -12V, durante quatro segundos, e sua rotação, em rpm, foi obtida por meio das leituras do *encoder*, obtendo-se valores entre +300 rpm e -300 rpm.

Essa aquisição de dados foi realizada cinco vezes, e uma média dos valores de rpm foi calculada, obtendo-se os dados que foram utilizados na estimação.

O motor simulado teve, aplicado em seus terminais, as mesmas tensões no mesmo intervalo de tempo.

Na Figura 23 pode-se observar o comportamento do motor real e do motor simulado, antes do início da estimação dos parâmetros característicos, e como suas respostas estão diferentes.

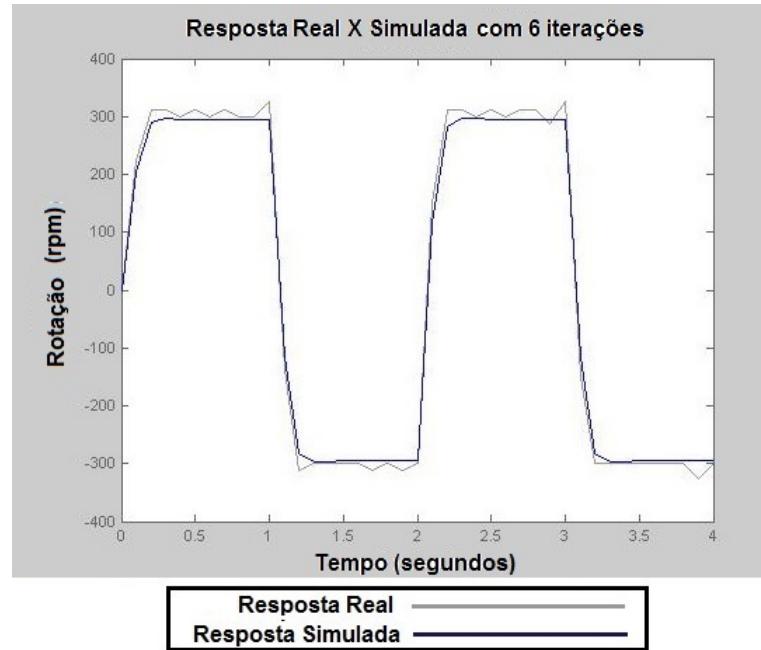
Figura 23 – Resposta do motor real e do motor simulado antes do início da estimação



Fonte: Autor

Com o início da estimação pela função *Parameter Estimation*, os valores de R_a , L_a , J , B e K começam a ser alterados. Após 6 iterações as respostas ficam como ilustrado na Figura 24.

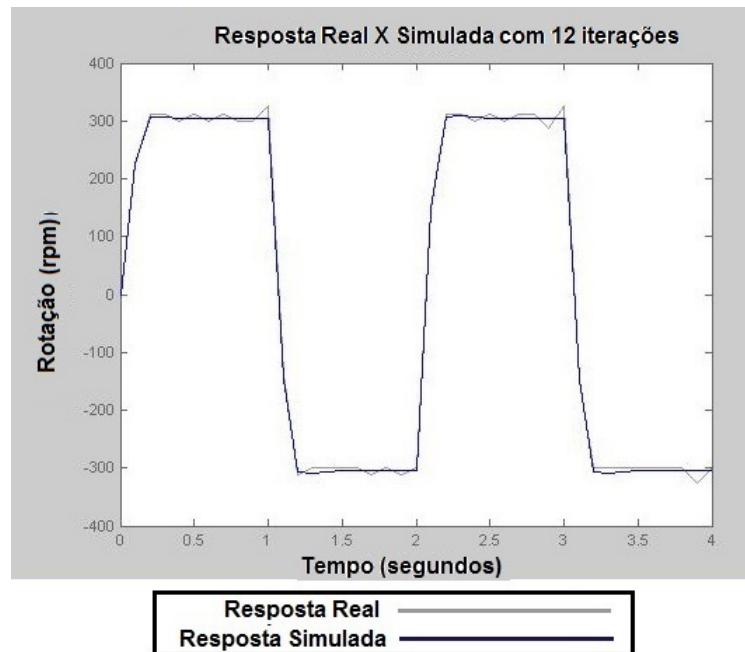
Figura 24 – Resposta do motor real e do motor simulado após 6 iterações



Fonte: Autor

A estimativa terminou após a 12^a iteração, obtendo-se a resposta presente da Figura 25.

Figura 25 – Resposta do motor real e do motor simulado após 12 iterações



Fonte: Autor

Com a aproximação das duas respostas, a estimativa gerou os seguintes valores para R_a , L_a , J , B e K , finalizando a identificação do sistema.

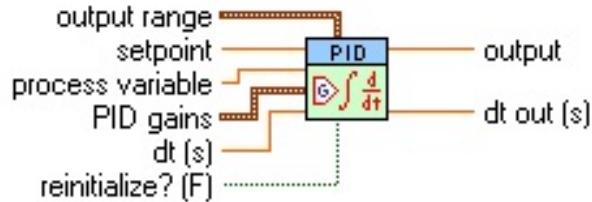
Tabela 10 – Valores gerados pela identificação

Parâmetro	Valor
Resistência da Armadura (R_a)	0,009 Ω
Indutância da Armadura (L_a)	0,348 H
Momento de Inércia Total (J)	0,000070029 $kg \cdot m^2$
Coeficiente de Atrito Viscoso (B)	0,0023 N.m/rpm
Constante do Motor (K)	0,039

Fonte: Autor

4.4 Implementação do controlador PID

Na seção 3.1 definiu-se quais informações fundamentais que o controlador PID necessita para funcionar: o valor da variável do processo, o valor do *setpoint* e o valor de seus ganhos (K_P, T_i e T_d). O controlador PID foi implementado por meio do *LabVIEW®*, junto com o sistema supervisório. As ligações do controlador PID do *LabVIEW®* (Figura 26) e suas funções são:

Figura 26 – Controlador PID do *LabVIEW®*Fonte: *LabVIEW®*

- *Output range*: determinar o valor máximo e mínimo que a saída do controlador pode assumir;
- *Setpoint*: determinar o valor no qual deseja-se que o processo atue;
- *Process variable*: resposta real obtida do processo. Determina o comportamento atual do processo;
- *PID gains*: ganhos K_P, T_i e T_d do controlador PID;
- *dt (s)*: determinar, quando necessário, o intervalo de tempo que o controlador deve atuar;
- *Reinitialize*: reiniciar o controlador, quando necessário;
- *Output*: valor que deve ser enviado de volta ao processo, após a atuação do controlador, para realizar a compensação necessária,;

- *dt out (s)*: caso o *dt (s)* não seja determinado, contém o intervalo de tempo entre as atuações do controlador.

O valor da saída do controlador PID foi limitada entre 0 e 255, por representar o valor PWM que será enviado para o microcontrolador. O valor da velocidade desejada (*setpoint*) é determinado pelo usuário no sistema supervisório. O valor da variável do processo é o valor da velocidade atual do protótipo, em centímetros por segundo (cm/s).

Como o sistema supervisório recebe a quantidade de contagens do *encoder*, esse valor deve ser transformado matematicamente para representar a velocidade. De acordo com os dados do tamanho da roda do protótipo (42 milímetros de diâmetro) e das características do *encoder* (48 contagens por volta), é possível calcular quantos centímetros cada leitura do *encoder* representa, a partir dos seguintes cálculos:

- Comprimento total da roda (C):

$$C = 2 \cdot \pi \cdot R \quad (4.1)$$

onde *R* representa o raio da roda.

Logo,

$$C = 2 \cdot 3,14 \cdot 21 \text{ mm} = 131,88 \text{ mm} \quad (4.2)$$

- Valor, em centímetros, de cada contagem do *encoder*:

$$\frac{131,88 \text{ mm}}{48} = 2,74 \text{ mm/contagem} \quad (4.3)$$

$$\frac{2,74 \text{ mm}}{10} = 0,274 \text{ cm/contagem} \quad (4.4)$$

Para se obter a velocidade, portanto, multiplica-se a quantidade de contagens do *encoder* por 0,274 cm, obtendo-se a distância percorrida, e divide-se esse valor por 0,09 segundos (tempo entre as leituras do *encoder*, que ocorrem a cada troca de informações entre o microcontrolador e o sistema supervisório).

4.5 Criação do sistema supervisório

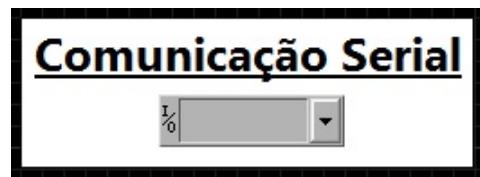
O sistema supervisório foi implementado junto com o controlador PID descrito no item anterior, por meio do software *LabVIEW®*, onde a criação é realizada em duas janelas: a do diagrama de blocos (*Block Diagram*), onde toda a programação é implementada, e a que será utilizada pelo usuário (*Front Panel*), onde se encontram os objetos utilizados para inserir e monitorar as variáveis de teste. Todas as informações que serão inseridas

ou monitoradas pelo usuário, no *Front Panel*, possuem uma programação específica no diagrama de blocos.

O sistema supervisório completo encontra-se no Apêndice B. Cada parte do sistema supervisório e suas principais características são:

- Comunicação serial (Figura 27): o usuário deve selecionar por qual porta serial a comunicação com o protótipo será realizada, para que seja possível realizar a conexão. Esse valor não pode ser alterado quando o sistema supervisório está conectado com o protótipo.

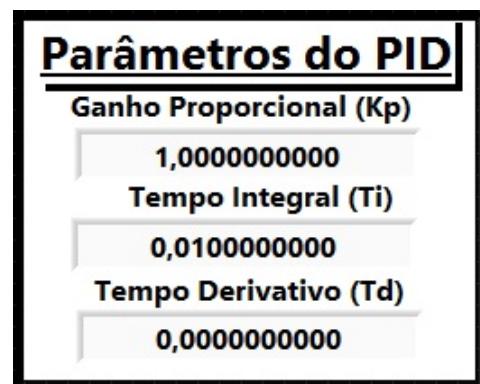
Figura 27 – Configuração da comunicação serial no sistema supervisório



Fonte: Autor

- Parâmetros do PID (Figura 28): o usuário deve inserir os valores de K_P , T_i e T_d . Esses valores podem ser alterados a qualquer momento durante o teste.

Figura 28 – Escolha dos parâmetros do controlador PID no sistema supervisório



Fonte: Autor

- Nome do arquivo externo (Figura 29): o usuário deve digitar, antes de iniciar a conexão com o protótipo, o nome e o formato do arquivo externo no qual os dados do teste serão salvos (por exemplo, “Teste1.xls” para salvar em uma planilha ou “Teste2.txt” para salvar como documento de texto). O nome do arquivo e seu formato não podem ser alterados durante o teste.

Figura 29 – Escolha do nome e formato do arquivo externo



Fonte: Autor

- Botões para iniciar e parar o teste (Figura 30): o usuário deve utilizar esses botões para iniciar, e quando necessário, parar o teste.

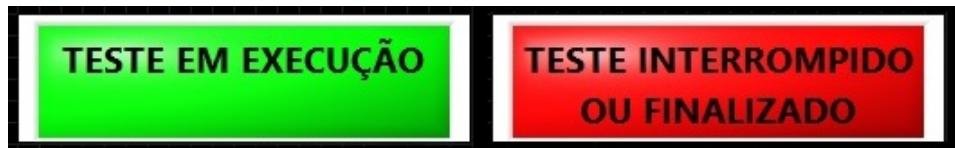
Figura 30 – Botões para iniciar e parar o teste



Fonte: Autor

- Indicador da situação do teste (Figura 31): são exibidas as mensagens “TESTE EM EXECUÇÃO” ou “TESTE INTERROMPIDO OU FINALIZADO”, de acordo com a situação atual do teste.

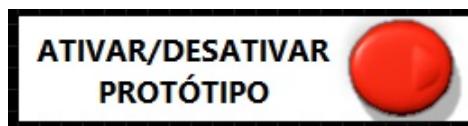
Figura 31 – Indicador da situação do teste



Fonte: Autor

- Botão para ativar e desativar o protótipo (Figura 32): botão de segurança, que deve ser acionado autorizando o protótipo a se movimentar. Pode ser acionado novamente, a qualquer momento, fazendo o protótipo parar.

Figura 32 – Botão para ativar e desativar o protótipo



Fonte: Autor

- Botão para parar a execução do programa (Figura 33): interrompe a execução do sistema supervisório e a conexão com o protótipo.

Figura 33 – Botão para parar a execução do programa



Fonte: Autor

- Escolha dos percursos (Figura 34): o usuário deve escolher, em cada aba, a distância (em centímetros) e a velocidade (em centímetros por segundo) que o protótipo deve percorrer em cada um dos quatro percursos. As distâncias e as velocidades podem ser alteradas a qualquer momento durante o teste.

Figura 34 – Seleção da distância e velocidade dos percursos



Fonte: Autor

- Informações do percurso atual (Figura 35): exibe informações sobre qual percurso o protótipo está naquele momento, a distância percorrida no percurso, a distância total que deve ser percorrida e a velocidade esperada daquele percurso, além do tempo total que o protótipo levou para percorrer cada um dos percursos.

Figura 35 – Informações sobre o percurso atual



Fonte: Autor

- Informações gerais do teste (Figura 36): exibe a distância total percorrida naquele teste, o tempo decorrido e possui um botão que zera todos os contadores, para que um novo teste possa ser realizado.

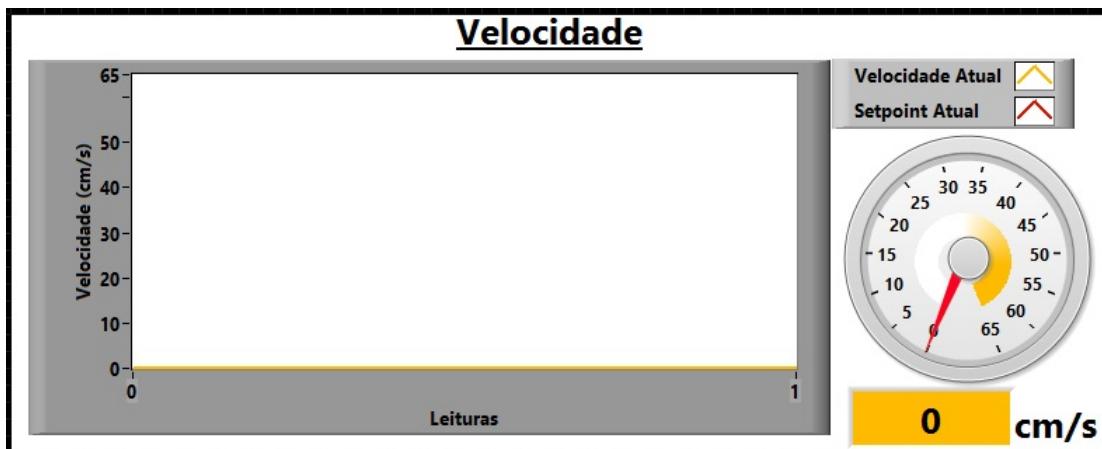
Figura 36 – Informações gerais do teste



Fonte: Autor

- Velocidade (Figura 37): um gráfico e um velocímetro exibem o *setpoint* (velocidade esperada) e a velocidade atual do protótipo, podendo ser possível visualizar como ele se comporta nas mudanças de velocidade.

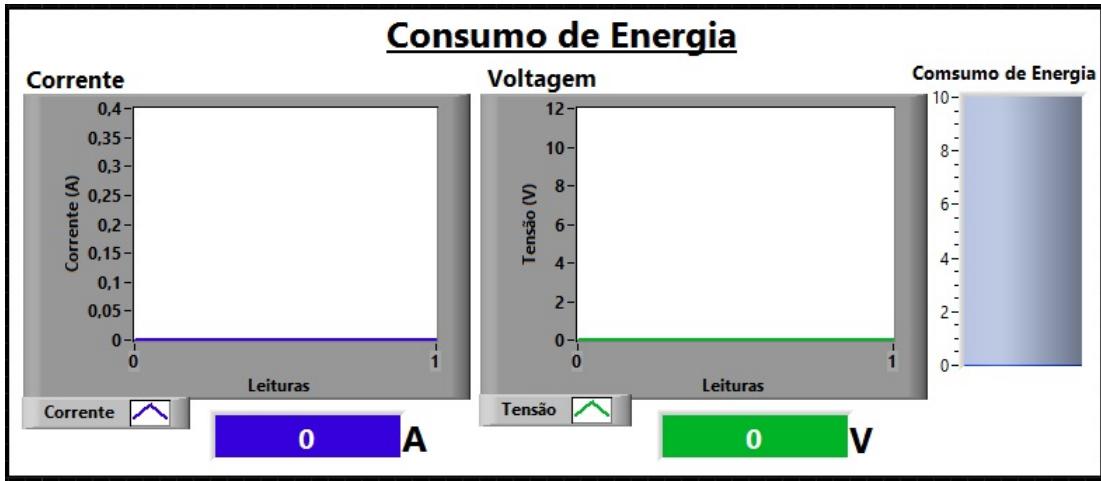
Figura 37 – Informações sobre a velocidade



Fonte: Autor

- Consumo de energia (Figura 38): dois gráficos exibem os valores atuais da corrente consumida pelo protótipo e a tensão elétrica aplicada nos motores. A corrente consumida é obtida a partir do sensor de corrente e a tensão elétrica aplicada aos motores a partir do valor do PWM proveniente do controlador PID, uma vez que esse valor representa o valor final da tensão elétrica aplicada nos motores. É a partir da multiplicação da tensão elétrica, da corrente consumida e do tempo, em horas, que o consumo de energia é obtido.

Figura 38 – Informações sobre o consumo de energia

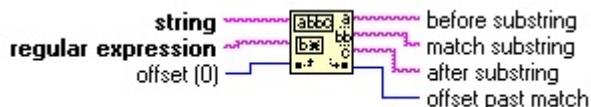


Fonte: Autor

Nas partes 6 e 9 da programação do microcontrolador, presente na Seção 4.2, foram abordadas a maneira pelo qual os dados são recebidos pelo microcontrolador e enviados para o sistema supervisório, respectivamente.

O sistema supervisório, ao receber os dados do microcontrolador, deve identificar em que ponto da mensagem se encontram as letras “a” e “b”, e considerar apenas o valor numérico que encontra-se entre as letras (o valor da leitura do *encoder* entre as letras “a” e da corrente entre as letras “b”).

Para isso, utilizou-se na programação o bloco *Match Pattern* (Figura 39). Sua função é procurar em uma frase (*string*) uma determinada letra ou palavra (*regular expression*) definida pelo usuário. Quando essa letra ou palavra é encontrada, o bloco divide a frase em três partes: o que está antes (*before substring*) da letra ou palavra, a própria letra ou palavra (*match substring*) e o que está depois dela (*after substring*).

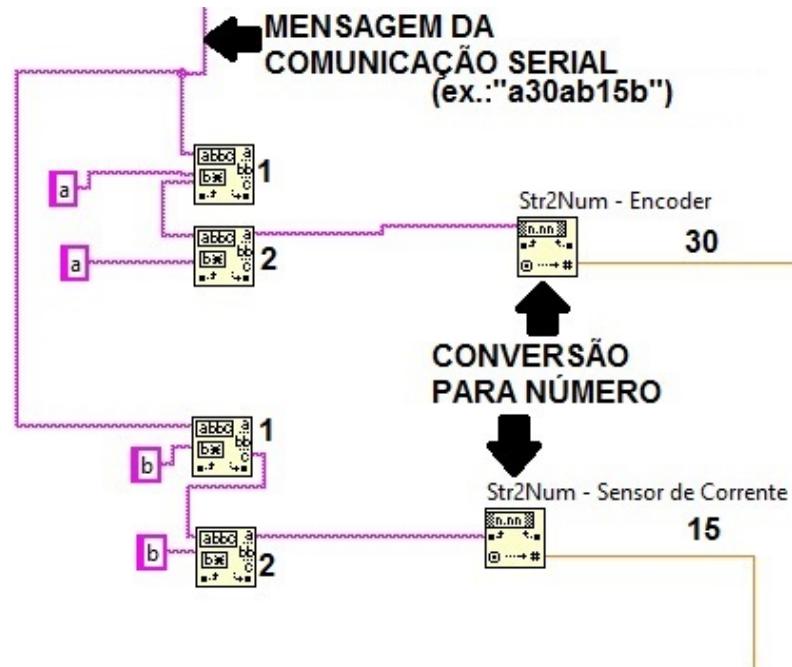
Figura 39 – Bloco *Match Pattern*Fonte: *LabVIEW®*

Na programação implementada (Figura 40), o bloco *Match Pattern* 1 procura o primeiro “a” ou o primeiro “b” e envia para o segundo bloco toda a mensagem que vem depois da letra. O bloco 2, por sua vez, retorna como resultado final o que vem antes da próxima letra “a” ou “b”, que já é o valor numérico desejado, mas ainda como uma frase (*string*). Eles são convertidos para números que podem ser utilizados pelo sistema supervisório.

Já no envio de informações, apenas um dado é enviado do sistema supervisório para o microcontrolador: o valor de PWM, que representa a tensão elétrica que deve ser aplicado nos motores. Para ser enviado, esse valor precisa ser separado em três partes: centena,

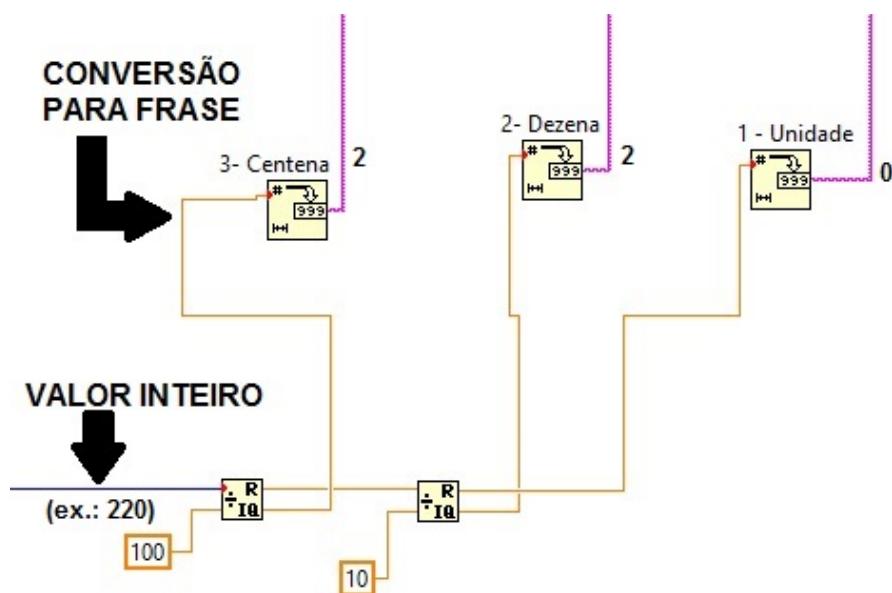
dezena e unidade. Na programação em diagrama de blocos (Figura 41), o número é dividido por 100, obtendo-se no cociente a centena. O resto dessa divisão é dividido por 10, obtendo-se no cociente a dezena e no resto a unidade. Após a separação, cada parte é transformada para frase (*string*), para poder ser enviado pela comunicação serial para o microcontrolador. O envio se dá na seguinte ordem: o primeiro é a unidade, o segundo é a dezena e o terceiro a centena.

Figura 40 – Programação implementada para identificar na mensagem valores desejados



Fonte: Autor

Figura 41 – Programação para dividir um número inteiro em centena, dezena e unidade



Fonte: Autor

4.6 Comunicação entre o sistema supervisório e o protótipo

A comunicação serial entre o microcontrolador presente no protótipo de carro elétrico e o sistema supervisório se dá com a utilização de dois módulos *XBee® Series 1* (Figura 42), fabricados pela *Digi International Inc.*, que utilizam o protocolo de comunicação 802.15.4 (padrão *ZigBee*, que efetua o controle de acesso para redes sem fios pessoais de baixas taxas de transmissão). As suas principais características encontram-se na Tabela 11.

Figura 42 – *XBee® Series 1*



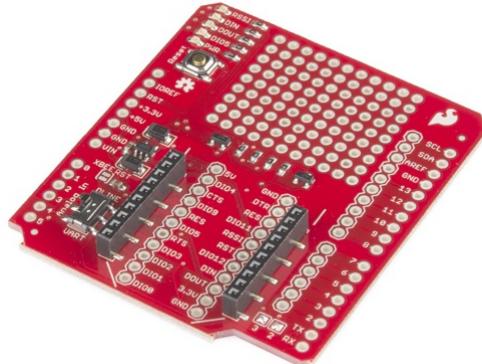
Fonte: *Digi International Inc.®* (<http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module#overview>)

Tabela 11 – Principais características do *XBee® Series 1*

Característica	Valor
Potência de saída	1 mW (+0dBm)
Alcance em uso interno/urbano	30 metros
Alcance em uso externo	90 metros
Taxa máxima de transmissão	250 Kbps
Frequência de operação	2,4 GHz

Fonte: *Digi International Inc.®* (<http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module#overview>)

Um dos módulos é conectado ao microcontrolador Arduino por meio de um *shield* (dispositivo que se encaixa no microcontrolador e que lhe fornece alguma função específica) e o outro a um *explorer* (dispositivo que é conectado ao computador por meio de uma porta USB). Ambos são fabricados pela *SparkFun Electronics®*, e encontram-se ilustrados nas Figuras 43 e 44, respectivamente.

Figura 43 – *Shield* para o *Xbee Series 1®*

Fonte: *SparkFun Electronics®* (<https://www.sparkfun.com/products/12847>)

Figura 44 – *Explorer* para o *Xbee Series 1®*

Fonte: *SparkFun Electronics®* (<https://www.sparkfun.com/products/11812>)

4.7 Sintonia do controlador PID

As fórmulas presentes nas Tabelas 2, 3, 4, 5 e 6 foram implementadas por meio de um algoritmo no *MATLAB®*. O usuário deve inserir os valores de K , t_0 e τ e os valores de K_P , T_i e T_d são calculados para os métodos de sintonia abordados. O algoritmo implementado encontra-se no Apêndice C.

4.8 Testes de consumo de energia

Em cada teste o protótipo percorre quatro percursos seguidos, cada um com uma distância e uma velocidade, conforme Tabela 12.

Tabela 12 – Percursos do teste de consumo de energia

Percorso	Distância	Velocidade
1	1000 cm	10 cm/s
2	3000 cm	30 cm/s
3	2000 cm	20 cm/s
4	4000 cm	40 cm/s

Fonte: Autor

No total, 15 testes foram realizados, sempre com os percursos da Tabela 12, conforme Tabela 13.

Tabela 13 – Testes realizados

Teste	Controlador	Sintonia
1	Proporcional (P)	ISE
2		IAE
3		ITAE
4		Cohen-Coon
5		Ziegler-Nichols
6	Proporcional-Integrativo (PI)	ISE
7		IAE
8		ITAE
9		Cohen-Coon
10		Ziegler-Nichols
11	Proporcional-Integral-Derivativo (PID)	ISE
12		IAE
13		ITAE
14		Cohen-Coon
15		Ziegler-Nichols

Fonte: Autor

Para cada percurso, de cada um dos testes, o consumo de energia foi obtido para o regime transitório (antes de atingir a velocidade desejada) e para o regime permanente (quando a velocidade se mantém em uma variação de 5% do valor desejado) (CASTRUCCI; BITTAR; SALES, 2011).

O cálculo do consumo de energia ocorreu utilizando-se a média da tensão elétrica aplicada aos motores, a média da corrente consumida pelo sistema e o tempo, em horas, de cada regime.

O regime transitório é considerado o intervalo de dados do início do percurso até o último valor antes da velocidade atingir 95% do valor esperado.

No regime permanente, considera-se o intervalo de dados da primeira vez que a velocidade atingiu 95% do valor esperado até o fim do percurso e, para fins de comparação, foi calculado a porcentagem média de desvio da velocidade com relação ao valor esperado.

Assim, pode-se estabelecer uma relação entre o consumo de energia e o quanto a velocidade se manteve próxima do valor esperado.

5 RESULTADOS E DISCUSSÕES

Neste capítulo serão abordados e discutidos todos os resultados obtidos no projeto, a partir da metodologia abordada no capítulo anterior.

5.1 Protótipo de carro elétrico

Na Seção 4.1 foram abordados o projeto e os materiais usados na montagem do protótipo de carro elétrico. No Apêndice D encontra-se um esquema de ligação elétrica de dois motores, que utilizam apenas um *driver*. É possível observar como a ligação do *driver* e do sensor de corrente foram feitas no microcontrolador. A ligação do segundo *driver* e dos outros dois motores ocorre da mesma maneira.

No Apêndice E encontram-se fotos do protótipo montado. O protótipo possui muitos fios de conexão devido a quantidade de portas necessárias para cada *driver* de motor funcionar. Uma *protoboard* foi utilizada para realizar a ligação entre o microcontrolador, os *drivers*, e os motores. O *encoder* e o sensor de corrente são ligados diretamente no microcontrolador. Uma chave liga/desliga foi implementada para interromper a alimentação da bateria quando necessário.

5.2 Função de transferência do motor CC

Aplicando-se os dados presentes na tabela 10, obtidos na identificação do motor CC, na Equação 3.7, obteve-se a função de transferência presente na Equação 5.1.

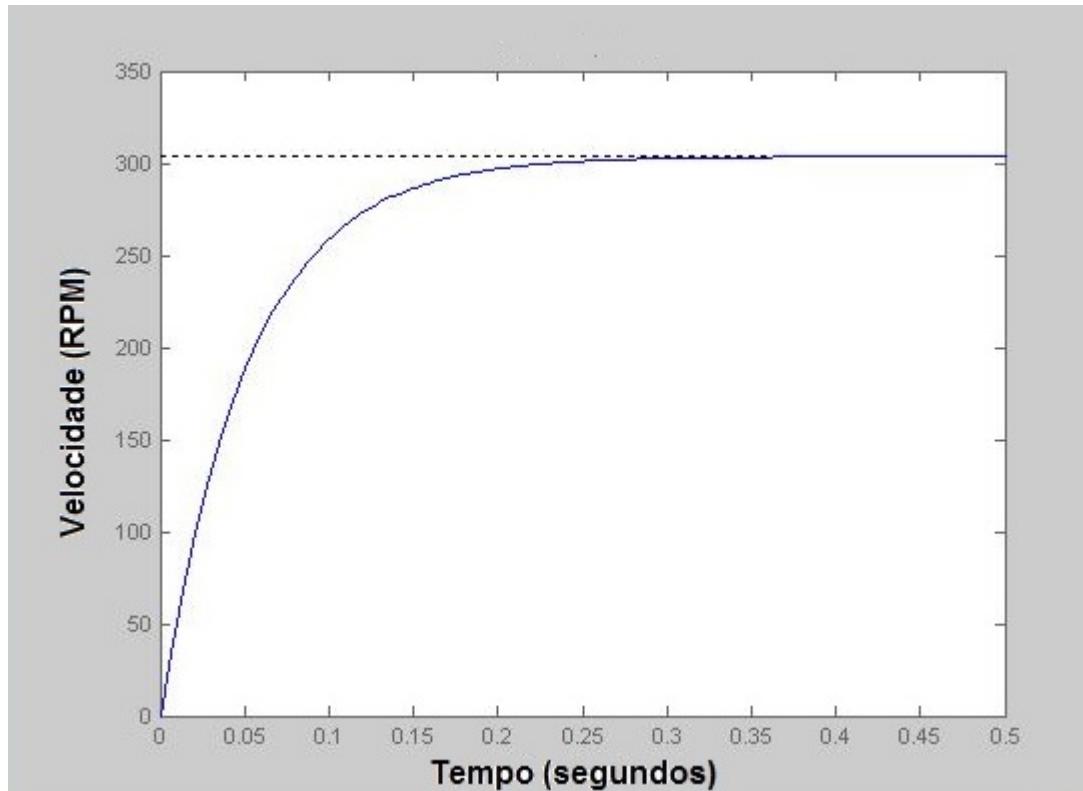
$$G(s) = \frac{25.3}{0.001581s^2 + 0.05233s + 1} \quad (5.1)$$

De acordo com Ricci (2011), quando o termo que acompanha o s^2 é muito pequeno, pode-se anulá-lo sem causar prejuízos ao sistema. Portanto, obtém-se o sistema de primeira ordem presente na Equação 5.2.

$$G(s) = \frac{25.3}{0.05233s + 1} \quad (5.2)$$

A validação dessa função de transferência ocorreu simulando-se, no *MATLAB®*, a aplicação de 12 volts no motor (degrau de magnitude 12), e observando-se se a resposta obtida corresponde à do motor real. A resposta encontra-se na figura 45.

Figura 45 – Resposta da aplicação de um degrau de magnitude 12 na função de transferência do motor



Fonte: Autor

Essa resposta é igual a resposta do motor real quando 12 volts é aplicado em seus terminais, atingindo a velocidade máxima de rotação em um tempo muito curto. Portanto, a função de transferência representa o motor real.

5.3 Sintonia do controlador PID

A função de transferência de primeira ordem (Equação 5.2) do motor CC, quando comparada à Equação característica de um sistema de primeira ordem com tempo morto (Equação 3.3), fornece as informações presentes na tabela 14 e que serão utilizados na sintonia dos controladores PID.

Como o sistema, originalmente, não apresentou tempo morto, considerou-se, para os cálculos da sintonia dos controladores PID, que ele possui um tempo morto muito pequeno, de 0,01 segundos.

Tabela 14 – Características obtidas da Equação 5.2

Característica	Valor
Constante K	25,3
Tempo morto t_0	0,01 s
Constante de tempo τ	0,05233 s

Fonte: Autor

Ao aplicar os valores de K , t_0 e τ no algoritmo presente no Apêndice C, obteve-se os valores de K_P , T_i e T_d presentes nas tabelas 15, 16 e 17.

Tabela 15 – Valores de K_P obtidos a partir da sintonia para o controlador P

Tipo de Controlador	Método de Sintonia	Parâmetro	Valor
Proporcional (P)	ISE	K_P	0,2544
	IAE		0,1820
	ITAE		0,1165
	Cohen-Coon		0,22
	Ziegler-Nichols		5,23

Fonte: Autor

Tabela 16 – Valores de K_P e T_i obtidos a partir da sintonia para o controlador PI

Tipo de Controlador	Método de Sintonia	Parâmetro	Valor
Proporcional-Integrativo (PI)	ISE	K_P	0,2522
		T_i	0,0313
	IAE	K_P	0,1989
		T_i	0,0267
	ITAE	K_P	0,1710
		T_i	0,0252
	Cohen-Coon	K_P	0,1894
		T_i	0,0238
	Ziegler-Nichols	K_P	4,7097
		T_i	0,0333

Fonte: Autor

Tabela 17 – Valores de K_P , T_i e T_d obtidos a partir da sintonia para o controlador PID

Tipo de Controlador	Método de Sintonia	Parâmetro	Valor
Proporcional-Integral-Derivativo (PID)	ISE	K_P	0,2823
		T_i	0,0133
		T_d	0,0055
	IAE	K_P	0,2604
		T_i	0,0173
		T_d	0,0038
	ITAE	K_P	0,2571
		T_i	0,0183
		T_d	0,0038
	Cohen-Coon	K_P	0,2857
		T_i	0,0228
		T_d	0,0035
	Ziegler-Nichols	K_P	6,2796
		T_i	0,02
		T_d	0,005

Fonte: Autor

5.4 Testes de consumo de energia

Levando em consideração os percursos apresentados na tabela 12 e os testes apresentados na tabela 13.

5.4.1 Testes 1 a 5

Todos os testes no qual apenas o controlador Proporcional (P) foi utilizado não levaram o protótipo à estabilização da velocidade, não sendo possível obter dados sobre o consumo de energia.

5.4.2 Testes 6 a 14

Os resultados detalhados do consumo de energia dos testes 6 a 14 encontram-se no Apêndice F.

As tabelas 18 a 26 contém os resultados de maneira resumida, apenas com o consumo de energia e o tempo de cada regime, valores totais dos percursos e valores totais do teste. Os valores na coluna “%” representam o valor médio de desvio dos dados do regime permanente com relação ao valor esperado.

Tabela 18 – Valores de consumo de energia do teste 6 (Controlador Proporcional-Integrativo e sintonia ISE)

Perc.	Regime Transitório		Regime Permanente		
	Tempo (s)	Consumo de Energia (Wh)	Tempo (s)	Consumo de Energia (Wh)	%
1	68,92	0,0055	58,51	0,0087	3,9835
2	64	0,0355	53,33	0,0428	3,4453
3	43,42	0,0278	42,07	0,022	4,6812
4	64,14	0,0573	48,50	0,0501	3,5457

VALORES TOTAIS DOS PERCURSOS		
Perc.	Tempo do Percurso (s)	Consumo de Energia do Percurso (Wh)
1	127,44	0,0142
2	117,34	0,0783
3	85,5	0,0499
4	112,65	0,1074

TOTAL DO TESTE	
Tempo do Teste (s)	Consumo de Energia do Teste (Wh)
442,93	0,2499

Fonte: Autor

Na tabela 18, observa-se que no teste 6 o protótipo levou 68,92 segundos e consumiu 0,0055 Wh para começar a movimentar-se e chegar a 9,5 cm/s (5% abaixo do valor esperado

de velocidade para o percurso 1, que é 10 cm/s), 64 segundos com consumo de 0,0355 Wh para aumentar a sua velocidade e chegar a 28,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 2, que é 30 cm/s), 43,42 segundos com consumo de 0,0278 Wh para reduzir sua velocidade e chegar a 21 cm/s (5% acima do valor esperado de velocidade para o percurso 3, que é de 20 cm/s) e 64,14 segundos com consumo de 0,0573 Wh para aumentar a sua velocidade e chegar a 38 cm/s (5% abaixo do valor esperado de velocidade para o percurso 4, que é de 40 cm/s).

Já nos regimes permanentes dos percursos 1 a 4, o protótipo permaneceu 58,51, 53,33, 42,07 e 48,50 segundos e consumiu 0,0087, 0,0428, 0,022 e 0,0501 Wh, respectivamente, para se manter dentro da faixa dos 5% do valor esperado, com tempo médio de 50,60 segundos e variação média com relação ao valor esperado de 3,9139%.

O protótipo levou 442,93 segundos com consumo de 0,2499 Wh para realizar o teste 6.

Tabela 19 – Valores de consumo de energia do teste 7 (Controlador Proporcional-Integrativo e sintonia IAE)

Perc.	Regime Transitório		Regime Permanente		
	Tempo (s)	Consumo de Energia (Wh)	Tempo (s)	Consumo de Energia (Wh)	%
1	74,51	0,0057	55,51	0,0081	4,5755
2	67,18	0,036	51,84	0,0409	3,9494
3	47,54	0,0304	37,91	0,02	4,6435
4	66,54	0,0589	46,82	0,0499	3,9262

VALORES TOTAIS DOS PERCURSOS

Perc.	Tempo do Percurso (s)	Consumo de Energia do Percurso (Wh)
1	130,02	0,0139
2	119,03	0,0770
3	85,46	0,0504
4	113,37	0,1088

TOTAL DO TESTE

Tempo do Teste (s)	Consumo de Energia do Teste (Wh)
447,89	0,2503

Fonte: Autor

Na tabela 19, observa-se que no teste 7 o protótipo levou 74,51 segundos e consumiu 0,0057 Wh para começar a movimentar-se e chegar a 9,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 1, que é 10 cm/s), 67,18 segundos com consumo de 0,036 Wh para aumentar a sua velocidade e chegar a 28,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 2, que é 30 cm/s), 47,54 segundos com consumo de 0,0304 Wh para reduzir sua velocidade e chegar a 21 cm/s (5% acima do valor esperado de velocidade para o percurso 3, que é de 20 cm/s) e 66,54 segundos com consumo de 0,0589 Wh para

aumentar a sua velocidade e chegar a 38 cm/s (5% abaixo do valor esperado de velocidade para o percurso 4, que é de 40 cm/s).

Já nos regimes permanentes dos percursos 1 a 4, o protótipo permaneceu 55,51, 51,84, 37,91 e 46,82 segundos e consumiu 0,0081, 0,0409, 0,02 e 0,0499 Wh, respectivamente, para se manter dentro da faixa dos 5% do valor esperado, com tempo médio de 48,02 segundos e variação média com relação ao valor esperado de 4,2736%.

O protótipo levou 447,89 segundos com consumo de 0,2503 Wh para realizar o teste 7.

Tabela 20 – Valores de consumo de energia do teste 8 (Controlador Proporcional-Integrativo e sintonia ITAE)

Perc.	Regime Transitório		Regime Permanente		
	Tempo (s)	Consumo de Energia (Wh)	Tempo (s)	Consumo de Energia (Wh)	%
1	81,13	0,0064	52,23	0,0077	5,1761
2	73,55	0,04	47,57	0,0381	4,5317
3	51,01	0,0331	33,67	0,0183	4,6911
4	75,56	0,0679	39,19	0,0418	4,1488

VALORES TOTAIS DOS PERCURSOS

Perc.	Tempo do Percurso (s)	Consumo de Energia do Percurso (Wh)
1	133,36	0,0142
2	121,13	0,0782
3	84,69	0,0515
4	114,76	0,1098

TOTAL DO TESTE

Tempo do Teste (s)	Consumo de Energia do Teste (Wh)
453,95	0,2538

Fonte: Autor

Na tabela 20, observa-se que no teste 8 o protótipo levou 81,13 segundos e consumiu 0,0064 Wh para começar a movimentar-se e chegar a 9,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 1, que é 10 cm/s), 73,55 segundos com consumo de 0,04 Wh para aumentar a sua velocidade e chegar a 28,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 2, que é 30 cm/s), 51,01 segundos com consumo de 0,0331 Wh para reduzir sua velocidade e chegar a 21 cm/s (5% acima do valor esperado de velocidade para o percurso 3, que é de 20 cm/s) e 75,56 segundos com consumo de 0,0679 Wh para aumentar a sua velocidade e chegar a 38 cm/s (5% abaixo do valor esperado de velocidade para o percurso 4, que é de 40 cm/s).

Já nos regimes permanentes dos percursos 1 a 4, o protótipo permaneceu 52,23, 47,57, 33,67 e 39,19 segundos e consumiu 0,0077, 0,0381, 0,0183 e 0,0418 Wh, respectivamente, para se manter dentro da faixa dos 5% do valor esperado, com tempo médio de 43,17 segundos e variação média com relação ao valor esperado de 4,6369%.

O protótipo levou 453,95 segundos com consumo de 0,2538 Wh para realizar o teste 8.

Tabela 21 – Valores de consumo de energia do teste 9 (Controlador Proporcional-Integrativo e sintonia Cohen-Coon)

Perc.	Regime Transitório		Regime Permanente		
	Tempo (s)	Consumo de Energia (Wh)	Tempo (s)	Consumo de Energia (Wh)	%
1	65,92	0,005	61,43	0,0089	4,1406
2	63,01	0,034	54,13	0,0428	3,3816
3	45,07	0,0283	40,4	0,02	4,6935
4	61,04	0,0532	51,24	0,0548	2,9021

VALORES TOTAIS DOS PERCURSOS					
Perc.	Tempo do Percurso (s)	Consumo de Energia do Percurso (Wh)			
1	127,35				0,0140
2	117,15				0,0768
3	85,47				0,0490
4	112,29				0,1080

TOTAL DO TESTE		
Tempo do Teste (s)	Consumo de Energia do Teste (Wh)	
442,28	0,2479	

Fonte: Autor

Na tabela 21, observa-se que no teste 9 o protótipo levou 65,92 segundos e consumiu 0,005 Wh para começar a movimentar-se e chegar a 9,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 1, que é 10 cm/s), 63,01 segundos com consumo de 0,034 Wh para aumentar a sua velocidade e chegar a 28,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 2, que é 30 cm/s), 45,07 segundos com consumo de 0,0283 Wh para reduzir sua velocidade e chegar a 21 cm/s (5% acima do valor esperado de velocidade para o percurso 3, que é de 20 cm/s) e 61,04 segundos com consumo de 0,0532 Wh para aumentar a sua velocidade e chegar a 38 cm/s (5% abaixo do valor esperado de velocidade para o percurso 4, que é de 40 cm/s).

Já nos regimes permanentes dos percursos 1 a 4, o protótipo permaneceu 61,43, 54,13, 40,4 e 51,24 segundos e consumiu 0,0089, 0,0428, 0,02 e 0,0548 Wh, respectivamente, para se manter dentro da faixa dos 5% do valor esperado, com tempo médio de 51,80 segundos e variação média com relação ao valor esperado de 3,7794%.

O protótipo levou 442,28 segundos com consumo de 0,2479 Wh para realizar o teste 9.

Tabela 22 – Valores de consumo de energia do teste 10 (Controlador Proporcional-Integrativo e sintonia Ziegler-Nichols)

Perc.	Regime Transitório		Regime Permanente		
	Tempo (s)	Consumo de Energia (Wh)	Tempo (s)	Consumo de Energia (Wh)	%
1	4,68	0,0004	95,01	0,0141	0,4392
2	1,85	0,001	97,39	0,0772	0,6139
3	3,53	0,0018	93,68	0,0421	0,3006
4	3,53	0,0032	95,51	0,0991	0,2827

VALORES TOTAIS DOS PERCURSOS		
Perc.	Tempo do Percurso (s)	Consumo de Energia do Percurso (Wh)
1	99,70	0,0146
2	99,25	0,0782
3	97,22	0,0439
4	99,05	0,1024

TOTAL DO TESTE		
Tempo do Teste (s)	Consumo de Energia do Teste (Wh)	
395,23	0,2393	

Fonte: Autor

Na tabela 22, observa-se que no teste 10 o protótipo levou 4,68 segundos e consumiu 0,0004 Wh para começar a movimentar-se e chegar a 9,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 1, que é 10 cm/s), 1,86 segundos com consumo de 0,001 Wh para aumentar a sua velocidade e chegar a 28,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 2, que é 30 cm/s), 3,53 segundos com consumo de 0,0018 Wh para reduzir sua velocidade e chegar a 21 cm/s (5% acima do valor esperado de velocidade para o percurso 3, que é de 20 cm/s) e 3,53 segundos com consumo de 0,0032 Wh para aumentar a sua velocidade e chegar a 38 cm/s (5% abaixo do valor esperado de velocidade para o percurso 4, que é de 40 cm/s).

Já nos regimes permanentes dos percursos 1 a 4, o protótipo permaneceu 95,01, 97,39, 93,68 e 95,51 segundos e consumiu 0,0141, 0,0772, 0,0421 e 0,0991 Wh, respectivamente, para se manter dentro da faixa dos 5% do valor esperado, com tempo médio de 95,40 segundos e variação média com relação ao valor esperado de 0,4108%.

O protótipo levou 395,23 segundos com consumo de 0,2393 Wh para realizar o teste 10.

Tabela 23 – Valores de consumo de energia do teste 11 (Controlador Proporcional-Integral-Derivativo e sintonia ISE)

Perc.	Regime Transitório		Regime Permanente		
	Tempo (s)	Consumo de Energia (Wh)	Tempo (s)	Consumo de Energia (Wh)	%
1	26,43	0,0019	82,65	0,0118	1,088
2	24,13	0,0128	81,03	0,0637	0,8758
3	15,83	0,0095	76,98	0,0344	1,7414
4	22,66	0,0188	81,16	0,0837	1,0358

VALORES TOTAIS DOS PERCURSOS		
Perc.	Tempo do Percurso (s)	Consumo de Energia do Percurso (Wh)
1	109,08	0,0137
2	105,17	0,0766
3	92,82	0,044
4	103,82	0,1026

TOTAL DO TESTE	
Tempo do Teste (s)	Consumo de Energia do Teste (Wh)
410,90	0,237

Fonte: Autor

Na tabela 23, observa-se que no teste 11 o protótipo levou 26,43 segundos e consumiu 0,0019 Wh para começar a movimentar-se e chegar a 9,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 1, que é 10 cm/s), 24,13 segundos com consumo de 0,0128 Wh para aumentar a sua velocidade e chegar a 28,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 2, que é 30 cm/s), 15,83 segundos com consumo de 0,0095 Wh para reduzir sua velocidade e chegar a 21 cm/s (5% acima do valor esperado de velocidade para o percurso 3, que é de 20 cm/s) e 22,66 segundos com consumo de 0,0188 Wh para aumentar a sua velocidade e chegar a 38 cm/s (5% abaixo do valor esperado de velocidade para o percurso 4, que é de 40 cm/s).

Já nos regimes permanentes dos percursos 1 a 4, o protótipo permaneceu 82,65, 81,03, 76,98 e 81,16 segundos e consumiu 0,0118, 0,0637, 0,0344 e 0,0837 Wh, respectivamente, para se manter dentro da faixa dos 5% do valor esperado, com tempo médio de 80,45 segundos e variação média com relação ao valor esperado de 1,1852%.

O protótipo levou 410,90 segundos com consumo de 0,237 Wh para realizar o teste 11.

Tabela 24 – Valores de consumo de energia do teste 12 (Controlador Proporcional-Integral-Derivativo e sintonia IAE)

Perc.	Regime Transitório		Regime Permanente		
	Tempo (s)	Consumo de Energia (Wh)	Tempo (s)	Consumo de Energia (Wh)	%
1	37,39	0,0026	76,28	0,0107	1,79
2	32,35	0,0165	75,97	0,0588	1,7761
3	25,63	0,0153	64,70	0,0293	2,5579
4	34,06	0,0283	72,03	0,073	1,3771

VALORES TOTAIS DOS PERCURSOS		
Perc.	Tempo do Percurso (s)	Consumo de Energia do Percurso (Wh)
1	113,67	0,0134
2	108,33	0,0754
3	90,34	0,0447
4	106,09	0,102

TOTAL DO TESTE		
Tempo do Teste (s)	Consumo de Energia do Teste (Wh)	
418,45	0,2357	

Fonte: Autor

Na tabela 24, observa-se que no teste 12 o protótipo levou 37,39 segundos e consumiu 0,0026 Wh para começar a movimentar-se e chegar a 9,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 1, que é 10 cm/s), 32,35 segundos com consumo de 0,0165 Wh para aumentar a sua velocidade e chegar a 28,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 2, que é 30 cm/s), 25,63 segundos com consumo de 0,0153 Wh para reduzir sua velocidade e chegar a 21 cm/s (5% acima do valor esperado de velocidade para o percurso 3, que é de 20 cm/s) e 34,06 segundos com consumo de 0,0283 Wh para aumentar a sua velocidade e chegar a 38 cm/s (5% abaixo do valor esperado de velocidade para o percurso 4, que é de 40 cm/s).

Já nos regimes permanentes dos percursos 1 a 4, o protótipo permaneceu 76,28, 75,97, 64,70 e 72,03 segundos e consumiu 0,0107, 0,0588, 0,0293 e 0,073 Wh, respectivamente, para se manter dentro da faixa dos 5% do valor esperado, com tempo médio de 72,25 segundos e variação média com relação ao valor esperado de 1,8752%.

O protótipo levou 418,45 segundos com consumo de 0,2357 Wh para realizar o teste 12.

Tabela 25 – Valores de consumo de energia do teste 13 (Controlador Proporcional-Integral-Derivativo e sintonia ITAE)

Perc.	Regime Transitório		Regime Permanente		
	Tempo (s)	Consumo de Energia (Wh)	Tempo (s)	Consumo de Energia (Wh)	%
1	41,70	0,003	73,18	0,0104	1,8669
2	37,39	0,0199	71,64	0,0562	1,6072
3	24,48	0,015	65,49	0,0306	3,253
4	36,99	0,0309	69,74	0,0712	1,3748

VALORES TOTAIS DOS PERCURSOS		
Perc.	Tempo do Percurso (s)	Consumo de Energia do Percurso (Wh)
1	114,88	0,0134
2	109,03	0,0761
3	89,97	0,0456
4	106,73	0,1022

TOTAL DO TESTE		
Tempo do Teste (s)	Consumo de Energia do Teste (Wh)	
420,64	0,2375	

Fonte: Autor

Na tabela 25, observa-se que no teste 13 o protótipo levou 41,70 segundos e consumiu 0,003 Wh para começar a movimentar-se e chegar a 9,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 1, que é 10 cm/s), 37,39 segundos com consumo de 0,0199 Wh para aumentar a sua velocidade e chegar a 28,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 2, que é 30 cm/s), 24,48 segundos com consumo de 0,015 Wh para reduzir sua velocidade e chegar a 21 cm/s (5% acima do valor esperado de velocidade para o percurso 3, que é de 20 cm/s) e 36,99 segundos com consumo de 0,0309 Wh para aumentar a sua velocidade e chegar a 38 cm/s (5% abaixo do valor esperado de velocidade para o percurso 4, que é de 40 cm/s).

Já nos regimes permanentes dos percursos 1 a 4, o protótipo permaneceu 73,182, 71,6467, 65,4906 e 69,7440 segundos e consumiu 0,0104, 0,0562, 0,0306 e 0,0712 Wh, respectivamente, para se manter dentro da faixa dos 5% do valor esperado, com tempo médio de 70,01 segundos e variação média com relação ao valor esperado de 2,0254%.

O protótipo levou 420,64 segundos com consumo de 0,2375 Wh para realizar o teste 13.

Tabela 26 – Valores de consumo de energia do teste 14 (Controlador Proporcional-Integral-Derivativo e sintonia Cohen-Coon)

Perc.	Regime Transitório		Regime Permanente		
	Tempo (s)	Consumo de Energia (Wh)	Tempo (s)	Consumo de Energia (Wh)	%
1	47,63	0,0039	69,20	0,0103	2,1365
2	41,47	0,023	69,09	0,0567	2,0277
3	28,64	0,0183	59,83	0,029	4,0789
4	44,51	0,0392	63,36	0,0683	1,796

VALORES TOTAIS DOS PERCURSOS		
Perc.	Tempo do Percurso (s)	Consumo de Energia do Percurso (Wh)
1	116,83	0,0142
2	110,56	0,0797
3	88,48	0,0473
4	107,88	0,1075

TOTAL DO TESTE		
Tempo do Teste (s)	Consumo de Energia do Teste (Wh)	
423,76	0,249	

Fonte: Autor

Na tabela 26, observa-se que no teste 14 o protótipo levou 47,63 segundos e consumiu 0,0039 Wh para começar a movimentar-se e chegar a 9,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 1, que é 10 cm/s), 41,47 segundos com consumo de 0,023 Wh para aumentar a sua velocidade e chegar a 28,5 cm/s (5% abaixo do valor esperado de velocidade para o percurso 2, que é 30 cm/s), 28,64 segundos com consumo de 0,0183 Wh para reduzir sua velocidade e chegar a 21 cm/s (5% acima do valor esperado de velocidade para o percurso 3, que é de 20 cm/s) e 44,51 segundos com consumo de 0,0392 Wh para aumentar a sua velocidade e chegar a 38 cm/s (5% abaixo do valor esperado de velocidade para o percurso 4, que é de 40 cm/s).

Já nos regimes permanentes dos percursos 1 a 4, o protótipo permaneceu 69,20, 69,09, 59,83 e 63,36 segundos e consumiu 0,0103, 0,0567, 0,029 e 0,0683 Wh, respectivamente, para se manter dentro da faixa dos 5% do valor esperado, com tempo médio de 65,37 segundos e variação média com relação ao valor esperado de 2,5097%.

O protótipo levou 423,76 segundos com consumo de 0,249 Wh para realizar o teste 14.

5.4.3 Teste 15

No teste realizado com controlador Proporcional-Integral-Derivativo (PID) e método de sintonia Ziegler-Nichols não levou o protótipo à estabilização da velocidade, não sendo possível obter dados do consumo de energia.

5.5 Comparação dos resultados de consumo de energia

Nos testes de consumo de energia, comparando-se apenas os cinco testes com controladores PI, têm-se que o método de sintonia mais eficiente foi Ziegler-Nichols, utilizado no teste 10. Com este método de sintonia, o regime transitório foi, comparado com os outros testes, extremamente rápido, com tempos entre 1,85 segundos (para passar do percurso 2 para o percurso 3) e 4,68 segundos (para começar a movimentar-se e chegar a 9,5 cm/s, 5% abaixo da velocidade do percurso 1, 10 cm/s). Pelo fato do tempo ser pequeno, o consumo de energia no regime transitório também foi o menor de todos os controladores PI. Já no regime permanente, o consumo foi, numericamente, o maior de todos os testes. Porém, isso ocorreu porque ele ficou mais tempo no regime permanente, mais de 90 segundos, enquanto nos outros testes com controlador PI o regime permanente durava, em média, 50 segundos. Ou seja, ele gastou, numericamente, mais energia mas permaneceu mais tempo dentro do valor de velocidade esperado. Além disso, a taxa de variação em torno do valor esperado de velocidade foi menos de 1% em todos os percursos do teste 10, se mostrando mais eficiente que os outros testes com controlador PI, em que a taxa de variação sempre foi maior que 3%. O tempo total do teste 10 e o seu consumo de energia total foram 395,23 segundos e 0,2393 Wh, também os menores dentre todos os testes com controlador PI.

Comparando-se os outros quatro testes (6 a 9) com controlador PI, o maior consumo de energia total foi 0,2538 Wh, com tempo total de 453,95 segundos, registrado no teste 8, que utiliza o método de sintonia ITAE, seguido pelo teste 7, que utiliza o método de sintonia IAE, com consumo de energia total de 0,2503 Wh e tempo total de 447,89 segundos e pelo teste 6, que utiliza o método de sintonia ISE, que tem consumo de energia total de 0,2499 Wh e tempo total de 442,93 segundos. O teste com menor consumo de energia e menor tempo entre esses quatro testes foi o 9, com método de sintonia Cohen-Coon, que registrou consumo de energia total de 0,2479 Wh e tempo total de teste de 442,28 segundos. Com relação à taxa percentual de variação em torno da velocidade esperada, no regime permanente, considerando-se os valores médios, a ordem foi a mesma do consumo de energia: a maior variação ocorreu no teste 8, 4,6351%, seguido pelo teste 7, 4,2736%, pelo teste 6, 3,9139% e, por fim, o teste 9, com 3,7794%. O consumo de energia no regime transitório foi diretamente proporcional ao tempo que o protótipo permaneceu nesse regime. O teste 8 teve os regimes transitórios mais demorados e o teste 9, os mais rápidos (com exceção do regime transitório do percurso 3, que no teste 6 foi cerca de 1,5 segundos mais rápido que no teste 8).

Ou seja, nos testes com controlador PI, o método mais eficiente foi o de Ziegler-Nichols, que se mostrou muito mais eficiente que os outros. Considerando-se os outros quatro testes realizados, em segundo lugar tem-se o teste 9, que utiliza o método de sintonia Cohen-Coon, uma vez que foi o mais econômico com relação ao consumo de energia, o mais rápido, e cuja velocidade sofreu menores variações em torno da velocidade esperada no regime permanente.

Já nos testes que utilizaram controlador PID, o consumo de energia total foi praticamente igual nos testes 11, 12 e 13, que utilizaram os métodos de sintonia ISE, IAE e ITAE: 0,237, 0,2357 e 0,2375 Wh, respectivamente. Porém, o tempo total teve uma diferença considerável: o mais rápido foi o teste 11, com 410,90 segundos e o mais demorado o teste 13, com 420,64 segundos. O teste 12 ficou entre os dois, com 418,45 segundos. O teste 11 teve o menor tempo total porque os seus regimes transitórios foram os mais rápidos, permanecendo assim mais tempo regime permanente. Isso fez com que seu consumo de energia fosse um dos maiores (praticamente igual a do teste 13). Porém, pode-se considerá-lo o mais eficiente, pois seu gasto de energia maior foi necessário para manter o protótipo mais tempo próximo da velocidade desejada. Isso é comprovado analisando-se a porcentagem média de desvio com relação ao valor de velocidade desejado: a do teste 11 foi a menor, com 1,1852%, seguido pelos testes 12, com 1,8752% e 13, com 2,0254%.

O teste 14, dentre os quatro realizados com controlador PID, teve o maior consumo de energia, 0,249 Wh, e o maior tempo, 423,7687 segundos, com porcentagem média de desvio de 2,5097%

6 CONCLUSÕES

Nos testes com controlador PI, tem-se que o considerado o método de sintonia mais eficiente, tendo como critério o consumo de energia, é o Ziegler-Nichols com controlador PI, uma vez que teve o regime transitório mais rápido (1,85 a 4,68 segundos) e com consumo baixo, o regime permanente mais demorado e com menor taxa de variação percentual com relação ao *setpoint* (menos de 1%) , assim como o menor tempo total (395,23 segundos) e o menor consumo de energia total (0,2393 Wh).

O menos eficiente com o controlador PI é o método de sintonia ITAE, por ter o regime transitório mais demorado (51,01 a 81,13 segundos) e com maior consumo, regime permanente mais rápido e com maior taxa de variação percentual com relação ao *setpoint*, além do maior tempo total (453,95 segundos) e maior consumo de energia total (0,2538 Wh).

Já nos controladores PID, o método de sintonia mais eficiente é o ISE, uma vez que teve o regime transitório mais rápido (15,83 a 26,43 segundos) e com consumo baixo, o regime permanente mais demorado e com menor taxa de variação percentual com relação ao *setpoint* (1,1852% na média) , assim como o menor tempo total (410,9 segundos). O seu consumo de energia foi numericamente parecido com o dos métodos de sintonia IAE e ITAE. Porém, ele foi considerado o mais eficiente por ter o menor tempo total e a menor taxa de variação percentual com relação ao *setpoint*.

O menos eficiente com o controlador PID é o método de sintonia Cohen-Coon, por ter o regime transitório mais demorado (28,64 a 47,63 segundos) e com maior consumo, regime permanente mais rápido e com maior taxa de variação percentual com relação ao *setpoint*, além do maior tempo total (423,76 segundos) e maior consumo de energia total (0,249 Wh).

Na comparação entre os controladores PI e PID, o mais eficiente é o controlador PI com método de sintonia Ziegler Nichols. Com exceção dele, todos os testes com controlador PID mostraram-se mais eficiente que os testes com controlador PI. Por isso, como segundo melhor tem-se o controlador PID com método de sintonia ISE.

6.1 Trabalhos futuros

Tem-se como trabalhos futuros

- Verificação da autonomia da bateria, quando o controlador PID é sintonizado de maneiras diferentes;
- Realização dos testes em ambientes no qual o protótipo esteja sujeito a disturbios previamente estabelecidos, como uma rampa ou um peso extra;

- Testes de consumo de energia com mais métodos de sintonia de controladores PID;
- Testes de consumo de energia com outros tipos de controladores, como os que envolvem controle robusto, inteligência artificial e controle adaptativo;
- Realizar os testes com diferentes tipos de sistemas (térmicos, controle de nível, entre outros), verificando-se se a diferença de consumo de energia entre os diferentes métodos de sintonia de controlador PID também existe nesses sistemas;
- Melhorar a *interface* do sistema supervisório, exibindo as informações em diferentes janelas.

Referências

- ACHARYA, A.; MITRA, D.; HALDER, K. Stability analysis of delayed system using bode integral. In: IEEE. *Computer Communication and Informatics (ICCCI), 2013 International Conference on.* [S.l.], 2013. p. 1–5. 15
- ANG, K. H.; CHONG, G.; LI, Y. Pid control system analysis, design, and technology. *Control Systems Technology, IEEE Transactions on*, IEEE, v. 13, n. 4, p. 559–576, 2005. 21, 22
- ARAUJO, P. F. M. U. d. *Sistema de Controle*. Natal - RN, 2007. 12, 13
- ASTROM, K. J.; HAGGLUND, T. *Advanced PID control*. [S.l.]: ISA-The Instrumentation, Systems, and Automation Society; Research Triangle Park, NC 27709, 2006. 19
- AWOUDA, A. E. A.; MAMAT, R. B. Refine pid tuning rule using itae criteria. In: IEEE. *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*. [S.l.], 2010. v. 5, p. 171–176. 19
- BARR, M. Pulse width modulation. *Embedded Systems Programming*, v. 14, n. 10, p. 103–104, 2001. 28, 29
- CAPELETTO, G. J.; MOURA, G. Bers - balanço energético do rio grande do sul 2013: Ano base 2012. *Porto Alegre: Grupo CEEE*, 2013. 15
- CASTRUCCI, P. d. L.; BITTAR, A.; SALES, R. M. *Controle Automático*. [S.l.]: LTC, 2011. 21, 22, 53
- DORF, R.; BISHOP, R. *Sistema de Controle Moderno*. [S.l.]: LTC, 2013. 12, 25
- FUENTES, R. C. *Apostila de Automação Industrial*. [S.l.], 2005. 25
- JONES, R. W.; THAM, M. T. Maximum sensitivity based pid controller tuning: A survey and comparison. In: IEEE. *SICE-ICASE, 2006. International Joint Conference*. [S.l.], 2006. p. 3258–3263. 18, 19
- KAMARUDDIN, N. et al. Pid controller tuning for glycerin bleaching process using well-known tuning formulas-a simulation study. In: IEEE. *Industrial Electronics, 2009. IECON'09. 35th Annual Conference of IEEE*. [S.l.], 2009. p. 1682–1686. 20, 23
- KNOSPE, C. Pid control. *IEEE Control Systems Magazine*, v. 26, n. 1, p. 30–31, 2006. 12
- LJUNG, L. *System Identification: Theory for the User*, PTR Prentice Hall Information and System Sciences Series. [S.l.]: Prentice Hall, New Jersey, 1999. 26, 27
- LOPEZ, A.; MURRILL, P.; SMITH, C. Controller tuning relationships based on integral performance criteria. *Instrum. Technol*, v. 14, n. 12, p. 57–62, 1967. 24
- NILSSON, J. W.; SUSAN, A. R. *Circuitos Elétricos. 8ª Edição*. [S.l.]: Editora: Pearson, Páginas, 2008. 29
- OGATA, K. *Engenharia de Controle Moderno*. [S.l.]: Pearson Prentice Hall, 2013. 22, 23

- RICCI, M. C. *Modelo Matemático de Motores de Corrente Contínua.* [S.l.], 2011. 26, 54
- SMITH, C. A.; CORRIPIO, A. B. *Principles and Practices of Automatic Process Control.* 3. ed. [S.l.: s.n.], 2005. 22, 24, 25
- SOUZA, E. S. d.; MELO, L. T. C.; RODRIGUES, V. B. Relatório final: Projeto desafio - controle de luminosidade de led. 2013. 21
- SOUZA, W. M.; ROSSETTO, F. R.; OLIVEIRA, V. A. Sistema de controle em tempo real. 2009. 14
- STEPHAN, R. M. Acionamento, comando e controle de máquinas elétricas. *Universidade Federal do Rio de Janeiro. Rio de Janeiro*, 2008. 27
- TAN, W. et al. Comparison of some well-known pid tuning formulas. *Computers & chemical engineering*, Elsevier, v. 30, n. 9, p. 1416–1423, 2006. 13, 18
- TUREVSKIY, A. 2013. Disponível em: <<http://www.mathworks.com/videos/estimating-dc-motor-parameters-97057.html>>. 40

APÊNDICE A – Algoritmo implementado no microcontrolador Arduino

```
//BIBLIOTECAS

#include <stdio.h>
#include <PololuWheelEncoders.h>

PololuWheelEncoders encoders;

// 1. CONFIGURAÇÃO DAS PORTAS

//Driver Motor A
int AIN1A = 42;
int AIN2A = 43;
int BIN1A = 44;
int BIN2A = 45;
int PWMAMotor1 = 5;
int PWMAMotor2 = 4;
int STBYA = 30;

//Driver Motor B
int AIN1B = 46;
int AIN2B = 47;
int BIN1B = 48;
int BIN2B = 49;
int PWMBMotor3 = 3;
int PWMBMotor4 = 2;
int STBYB = 31;

//Portas Encoder
const unsigned char m1a = 50;
const unsigned char m1b = 51;
const unsigned char m2a = 52; //nao usada
const unsigned char m2b = 53; //nao usada
```

```
// 2. DECLARAÇÃO DAS VARIÁVEIS

    int pwm;
    int encoder , sensor ;
    int pwmv[ 3 ];

void setup() {

// 3. CONFIGURAÇÃO DE ENTRADA E SAÍDA

    pinMode(AIN1A,OUTPUT);
    pinMode(AIN2A,OUTPUT);
    pinMode(BIN1A,OUTPUT);
    pinMode(BIN2A,OUTPUT);
    pinMode(AIN1B,OUTPUT);
    pinMode(AIN2B,OUTPUT);
    pinMode(BIN1B,OUTPUT);
    pinMode(BIN2B,OUTPUT);
    pinMode(PWMAMotor1,OUTPUT);
    pinMode(PWMAMotor2,OUTPUT);
    pinMode(PWMBMotor3,OUTPUT);
    pinMode(PWMBMotor4,OUTPUT);
    pinMode(STBYA,OUTPUT);
    pinMode(STBYB,OUTPUT);

// 4. CONFIGURAÇÃO DA COMUNICACAO SERIAL

    Serial.begin(9600);

// 5. CONFIGURACAO DO ENCODER

    encoders.init(m1a,m1b,m2a,m2b);

}

void loop(){

// 6. RECEBIMENTOS DE DADOS DO LABVIEW

    if( Serial.available()>0){
```

```

        for ( int i=0;i<3;){
            if ( Serial.available ()>0){
                pwmv[ i ] = Serial.read () - '0';
                i++;
            }
        }

        pwm=pwmv[ 2 ]*100 + pwmv[ 1 ]*10 + pwmv[ 0 ];
    }
}

```

// 7. DADOS ENVIADOS PARA AS PORTAS

```

digitalWrite (AIN1A,HIGH);
digitalWrite (AIN2A,LOW);
digitalWrite (BIN1A,LOW);
digitalWrite (BIN2A,HIGH);
digitalWrite (STBYA,HIGH);
analogWrite (PWMAMotor1, pwm);
analogWrite (PWMAMotor2, pwm);
digitalWrite (AIN1B,HIGH);
digitalWrite (AIN2B,LOW);
digitalWrite (BIN1B,LOW);
digitalWrite (BIN2B,HIGH);
digitalWrite (STBYB,HIGH);
analogWrite (PWMBMotor3, pwm);
analogWrite (PWMBMotor4, pwm);
encoder = encoders.getCountsAndResetM1 ();

```

// 8. LEITURA DO SENSOR DE CORRENTE

```

float volts = 0;

volts = analogRead (A8);

float amperes = 0;

if ( volts <= 2.55){
    amperes = 0;
}

```

```
        }  
  
        if (volts > 2.55 && volts <= 2.61 ){  
            amperes = volts * 0.065134;  
        }  
  
        if (volts > 2.61 && volts <= 2.64 ){  
            amperes = volts * 0.072243;  
        }  
  
        if (volts > 2.64 && volts <= 2.67 ){  
            amperes = volts * 0.082397;  
        }  
  
        if (volts > 2.67 && volts <= 2.69 ){  
            amperes = volts * 0.092936;  
        }  
  
        if (volts > 2.69 && volts <= 2.72 ){  
            amperes = volts * 0.102564;  
        }  
  
        if (volts > 2.72 && volts <= 2.75 ){  
            amperes = volts * 0.109090;  
        }  
  
        if (volts > 2.75 && volts <= 2.79 ){  
            amperes = volts * 0.1151079;  
        }  
  
        if (volts > 2.79 && volts <= 2.82 ){  
            amperes = volts * 0.1174377;  
        }  
  
        if (volts > 2.82 && volts <= 2.84 ){  
            amperes = volts * 0.1245551;  
        }  
  
        if (volts > 2.84 && volts <= 2.86 ){  
            amperes = volts * 0.1385435;
```

```
        }

    if (volts > 2.86 && volts <= 2.89 ){
        amperes = volts * 0.145833;      }

    if (volts > 2.89 && volts <= 2.95 ){
        amperes = volts * 0.1535836;
    }

    if (volts > 2.95 && volts <= 2.99 ){
        amperes = volts * 0.161616;
    }

    if (volts > 2.99 && volts <= 3.02 ){
        amperes = volts * 0.169435;
    }

    if (volts > 3.02 && volts <= 3.05){
        amperes = volts * 0.17763;
    }

    if (volts > 3.05 && volts <= 3.07 ){
        amperes = volts * 0.185667;
    }

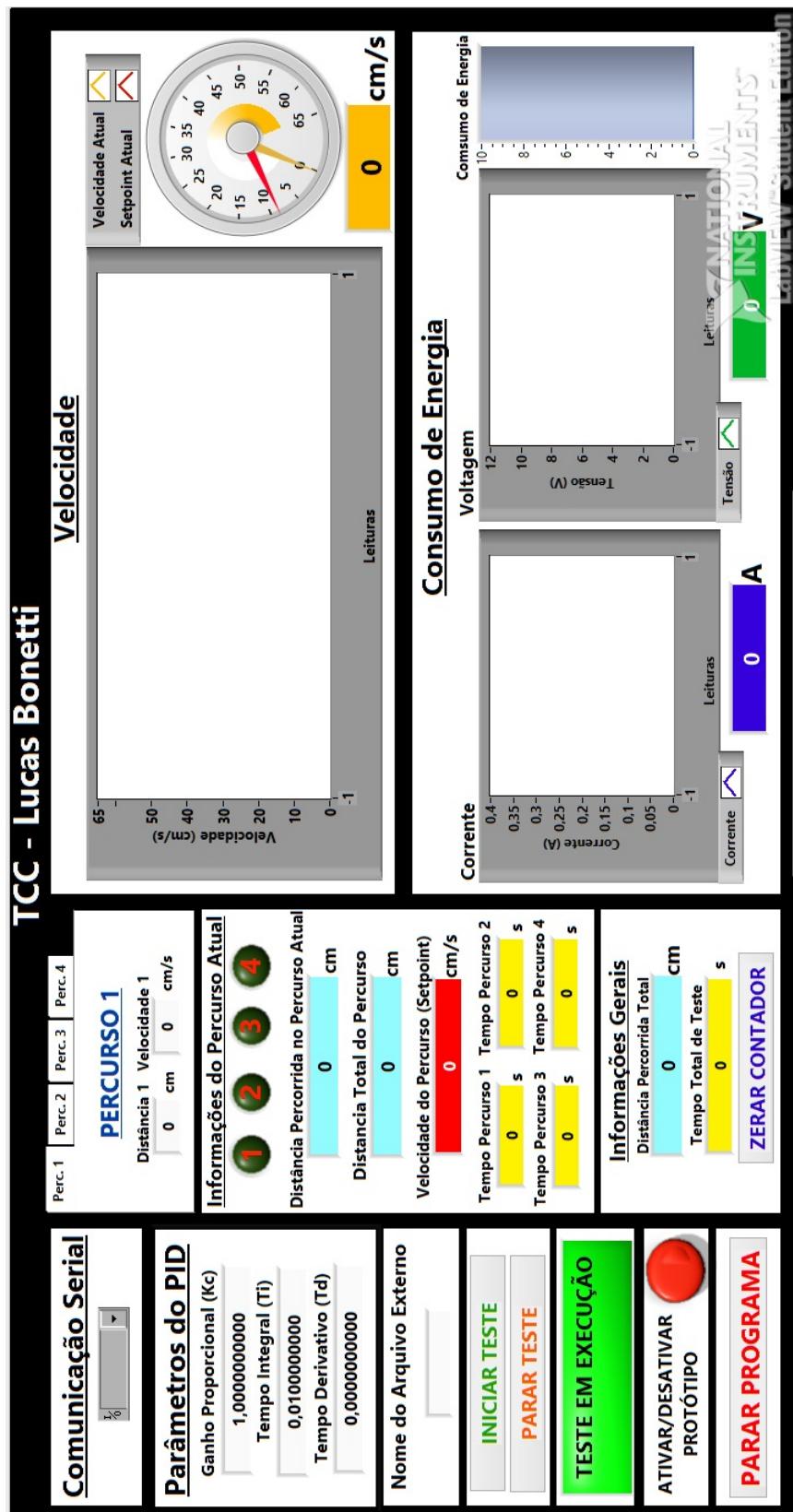
    int amperes2 = amperes*100;

// 9. ENVIO DE DADOS PARA O LABVIEW

    Serial.print("a");
    Serial.print(encoder);
    Serial.print("a");
    Serial.print("\n");
    Serial.print("b");
    Serial.print(amperes2);
    Serial.print("b");

    delay(90);
}
```

APÊNDICE B – Sistema supervisório implementado no LabVIEW®



APÊNDICE C – Algoritmo implementado no MATLAB® para cálculo de K_P , T_i e T_d

%% Sintonia de Controladores PID

```
disp( 'SINTONIA DE CONTROLADORES' )
```

```
teta=input( 'Insira a Constante de Tempo' )
```

```
t0=input( 'Insira o Tempo Morto' )
```

```
K=input( 'Insira o K' )
```

```
disp( 'CONTROLADOR PROPORCIONAL (P)' )
```

%%ISE

```
a=1.411;
```

```
b=-0.917;
```

```
disp( 'ISE' )
```

```
Kp=(a/K)*(( t0 / teta )^b)
```

%%IAE

```
a=0.902;
```

```
b=-0.985;
```

```
disp( 'IAE' )
```

```
Kp=(a/K)*(( t0 / teta )^b)
```

%%ITAE

```
a=0.49;
```

```
b=-1.084;
```

```
disp( 'ITAE' )
```

```
Kp=(a/K)*(( t0 / teta )^b)
```

```
disp('-----')
```

```
disp('CONTROLADOR PROPORCIONAL-INTEGRATIVO (PI)')
```

```
%%ISE
```

```
a1=1.305;
```

```
b1=-0.959;
```

```
a2=0.492;
```

```
b2=0.739;
```

```
disp('ISE')
```

```
Kp=(a1/K)*((t0/teta)^b1)
```

```
ti=(teta/a2)*((t0/teta)^b2)
```

```
%%IAE
```

```
a=0.984;
```

```
b=-0.986;
```

```
a2=0.608;
```

```
b2=0.707;
```

```
disp('IAE')
```

```
Kp=(a/K)*((t0/teta)^b)
```

```
ti=(teta/a2)*((t0/teta)^b2)
```

```
%%ITAE
```

```
a=0.859;
```

```
b=-0.977;
```

```
a2=0.674;
```

```
b2=0.680;
```

```
disp('ITAE')
```

```
Kp=(a/K)*((t0/teta)^b)
ti=(teta/a2)*((t0/teta)^b2)
td=(a3*teta)*((t0/teta)^b3)

disp('CONTROLADOR PROPORCIONAL-INTEGRATIVO-DERIVATIVO (PID)')

%%ISE
a1=1.495;
b1=-0.945;
a2=1.101;
b2=0.771;
a3=0.56;
b3=1.006;

disp('ISE')

Kp=(a1/K)*((t0/teta)^b1)
ti=(teta/a2)*((t0/teta)^b2)
td=(a3*teta)*((t0/teta)^b3)

%%IAE
a=1.435;
b=-0.921;
a2=0.878;
b2=0.749;
a3=0.482;
b3=1.137;

disp('IAE')

Kp=(a/K)*((t0/teta)^b)
ti=(teta/a2)*((t0/teta)^b2)
td=(a3*teta)*((t0/teta)^b3)

%%ITAE
a=1.357;
```

```
b=-0.947;
```

```
a2=0.842;
```

```
b2=0.738;
```

```
a3=0.381;
```

```
b3=0.995;
```

```
disp( 'ITAE' )
```

```
Kp=(a/K)*(( t0 / teta )^b)
```

```
ti=(teta / a2)*(( t0 / teta )^b2)
```

```
td=(a3*teta )*(( t0 / teta )^b3)
```

```
%% Método de COHEN-COON
```

```
disp( 'COHEN-COON' )
```

```
disp( 'Controlador Proporcional' )
```

```
Kp=(1/K)*( teta / t0 )*(1+(1/3*( t0 / teta )))
```

```
disp( 'Controlador Proporcional-Integrativo' )
```

```
Kp=(1/K)*( teta / t0 )*(0.9+(1/12*( t0 / teta )))
```

```
ti=t0*((30+3*( t0 / teta ))/(9+20*( t0 / teta )))
```

```
disp( 'Controlador Proporcional-Integrativo-Derivativo' )
```

```
Kp=(1/K)*( teta / t0 )*((4/3)+(1/4*( t0 / teta )))
```

```
ti=t0*((32+6*( t0 / teta ))/(13+8*( t0 / teta )))
```

```
td=t0*(4)/(11+2*( t0 / teta ))
```

```
%% Ziegler-Nichols
```

```
disp( 'ZIEGLER-NICHOLS' )
```

```
disp( 'Controlador Proporcional' )
```

```
Kp=teta / t0
```

```
disp( 'Controlador Proporcional-Integrativo' )
```

```
Kp=0.9*( teta / t0 )
```

```
ti=t0 / 0.3
```

```
disp( 'Controlador Proporcional-Integral-Derivativo' )
```

```
Kp=1.2*( teta / t0 )
```

```
ti=t0 * 2
```

```
td=t0 * 0.5
```

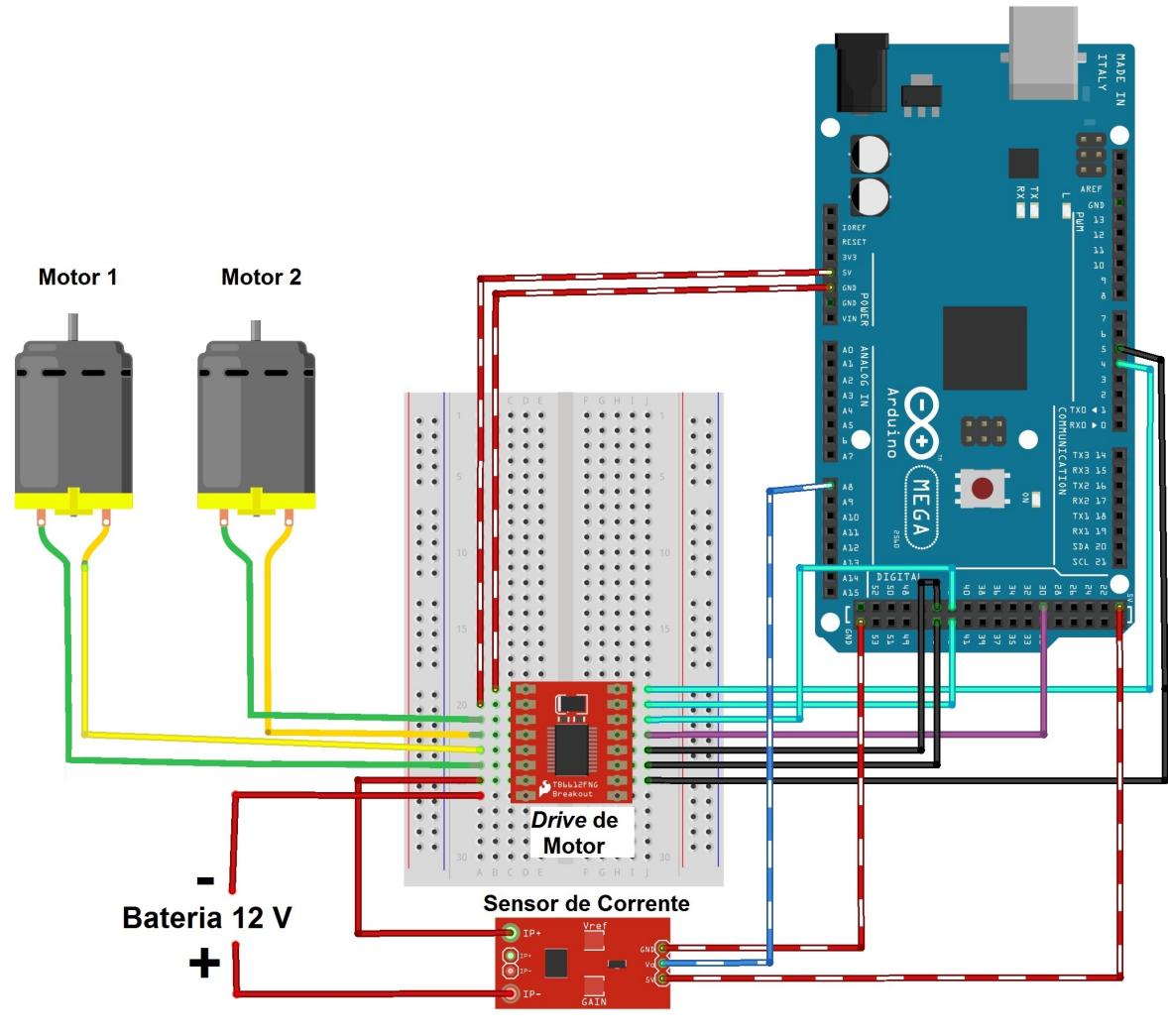
APÊNDICE D – Esquema de ligação elétrica do protótipo de carro elétrico

Os fios vermelhos representam a alimentação elétrica de 12V, provenientes da bateria. Os fios vermelhos com listras brancas representam a alimentação de 5V, proveniente do microcontrolador Arduino, que alimenta o *drive* de motor e o sensor de corrente.

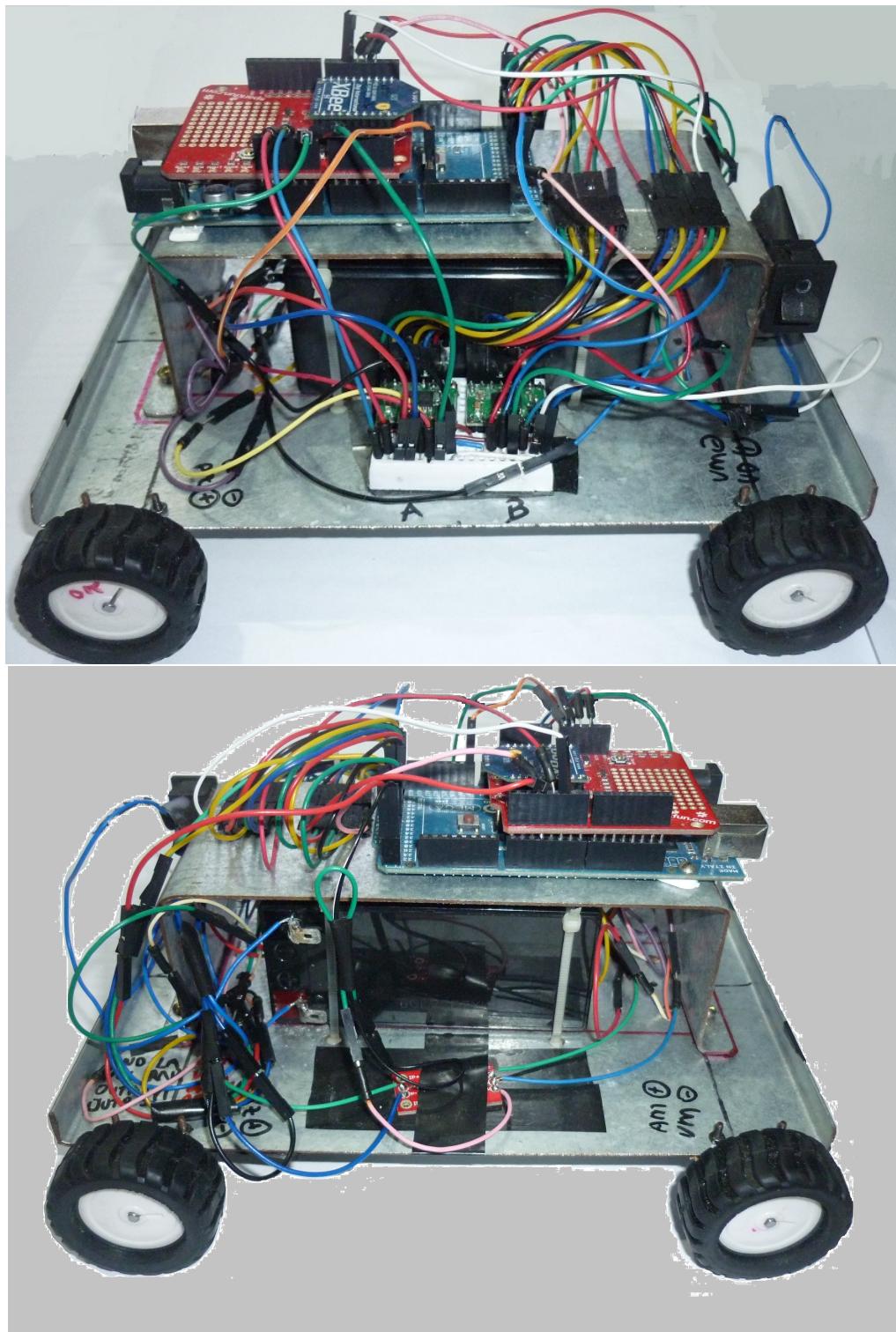
O sinal de saída do sensor de corrente é representado pelo fio azul com listras brancas

Os fios verdes são os polos negativos dos motores, e os amarelos os polos positivos.

Os fios azul-claro são o sinal de PWM e as portas AIN1 e AIN2 do *driver* de motor, que determinam o sentido de rotação e a velocidade do motor 1. Já os fios pretos se ligam as portas BIN1, BIN2 e PWM que determinam o sentido de rotação e a velocidade do motor 2, e o fio roxo é a porta *standby*, que permite ligar ou desligar os dois motores ligados ao *driver*.



APÊNDICE E – Foto do protótipo de carro elétrico



APÊNDICE F – Resultados dos testes de consumo de energia

Teste 6

Neste teste foi utilizado um controlador Proporcional-Integrativo (PI) com método de sintonia ISE. Os resultados obtidos encontram-se na Tabela 27.

Tabela 27 – Resultados de consumo de energia do controlador Proporcional-Integrativo com método de sintonia ISE

PERCURSO 1			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	1,2237	Tensão Média (Volts)	1,8161
Corrente Média (Amperes)	0,2367	Corrente Média (Amperes)	0,2951
Tempo (segundos)	68,92	Tempo (segundos)	58,51
Tempo (horas)	0,0191	Tempo (horas)	0,0162
Consumo de Energia (Wh)	0,0055	Consumo de Energia (Wh)	0,0087
		Porcentagem média de desvio (%)	3,9835
PERCURSO 1 TOTAL			
Tempo (segundos)		127,44	
Consumo de Energia (Wh)		0,0142	
PERCURSO 2			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,1387	Tensão Média (Volts)	5,3605
Corrente Média (Amperes)	0,4826	Corrente Média (Amperes)	0,5392
Tempo (segundos)	64	Tempo (segundos)	53,33
Tempo (horas)	0,0177	Tempo (horas)	0,0148
Consumo de Energia (Wh)	0,0355	Consumo de Energia (Wh)	0,0428
		Porcentagem média de desvio (%)	3,4453
PERCURSO 2 TOTAL			
Tempo (segundos)		117,34	
Consumo de Energia (Wh)		0,0783	

PERCURSO 3

Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,5814	Tensão Média (Volts)	3,9926
Corrente Média (Amperes)	0,5035	Corrente Média (Amperes)	0,4729
Tempo (segundos)	43,42	Tempo (segundos)	42,07
Tempo (horas)	0,012	Tempo (horas)	0,0116
Consumo de Energia (Wh)	0,0278	Consumo de Energia (Wh)	0,022
		Porcentagem média de desvio (%)	4,6812

PERCURSO 3 TOTAL

Tempo (segundos)	85,5
Consumo de Energia (Wh)	0,0499

PERCURSO 4

Regime Transitório		Regime Permanente	
Tensão Média (Volts)	6,0654	Tensão Média (Volts)	7,01
Corrente Média (Amperes)	0,5303	Corrente Média (Amperes)	0,5308
Tempo (segundos)	64,14	Tempo (segundos)	48,50
Tempo (horas)	0,0178	Tempo (horas)	0,0134
Consumo de Energia (Wh)	0,0573	Consumo de Energia (Wh)	0,0501
		Porcentagem média de desvio (%)	3,5457

PERCURSO 4 TOTAL

Tempo (segundos)	112,65
Consumo de Energia (Wh)	0,1074

TOTAL TESTE 6

Tempo (segundos)	442,93
Consumo de Energia (Wh)	0,2499

Fonte: Autor

Teste 7

Neste teste foi utilizado um controlador Proporcional-Integrativo (PI) com método de sintonia IAE. Os resultados obtidos encontram-se na Tabela 28.

Tabela 28 – Resultados de consumo de energia do controlador Proporcional-Integrativo com método de sintonia IAE

PERCURSO 1			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	1,2092	Tensão Média (Volts)	1,8148
Corrente Média (Amperes)	0,2309	Corrente Média (Amperes)	0,2907
Tempo (segundos)	74,51	Tempo (segundos)	55,51
Tempo (horas)	0,0206	Tempo (horas)	0,0154
Consumo de Energia (Wh)	0,0057	Consumo de Energia (Wh)	0,0081
		Porcentagem média de desvio (%)	4,5755
PERCURSO 1 TOTAL			
Tempo (segundos)		130,02	
Consumo de Energia (Wh)		0,0139	
PERCURSO 2			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,0885	Tensão Média (Volts)	5,3392
Corrente Média (Amperes)	0,4724	Corrente Média (Amperes)	0,5327
Tempo (segundos)	67,18	Tempo (segundos)	51,84
Tempo (horas)	0,0186	Tempo (horas)	0,0144
Consumo de Energia (Wh)	0,036	Consumo de Energia (Wh)	0,0409
		Porcentagem média de desvio (%)	3,9494
PERCURSO 2 TOTAL			
Tempo (segundos)		119,0350	
Consumo de Energia (Wh)		0,0770	
PERCURSO 3			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,604	Tensão Média (Volts)	4,0362
Corrente Média (Amperes)	0,4999	Corrente Média (Amperes)	0,4722
Tempo (segundos)	47,54	Tempo (segundos)	37,91
Tempo (horas)	0,0132	Tempo (horas)	0,0105
Consumo de Energia (Wh)	0,0304	Consumo de Energia (Wh)	0,02
		Porcentagem média de desvio (%)	4,6435
PERCURSO 3 TOTAL			
Tempo (segundos)		85,46	
Consumo de Energia (Wh)		0,0504	

PERCURSO 4			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	6,0478	Tensão Média (Volts)	7,2874
Corrente Média (Amperes)	0,5273	Corrente Média (Amperes)	0,5267
Tempo (segundos)	66,54	Tempo (segundos)	46,82
Tempo (horas)	0,0184	Tempo (horas)	0,013
Consumo de Energia (Wh)	0,0589	Consumo de Energia (Wh)	0,0499
		Porcentagem média de desvio (%)	3,9262

PERCURSO 4 TOTAL

Tempo (segundos)	113,37
Consumo de Energia (Wh)	0,1088

TOTAL TESTE 7

Tempo (segundos)	447,89
Consumo de Energia (Wh)	0,2503

Fonte: Autor

Teste 8

Neste teste foi utilizado um controlador Proporcional-Integrativo (PI) com método de sintonia ITAE. Os resultados obtidos encontram-se na Tabela 29.

Tabela 29 – Resultados de consumo de energia do controlador Proporcional-Integrativo com método de sintonia ITAE

PERCURSO 1			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	1,2157	Tensão Média (Volts)	1,8287
Corrente Média (Amperes)	0,2351	Corrente Média (Amperes)	0,2937
Tempo (segundos)	81,13	Tempo (segundos)	52,23
Tempo (horas)	0,0225	Tempo (horas)	0,0145
Consumo de Energia (Wh)	0,0064	Consumo de Energia (Wh)	0,0077
		Porcentagem média de desvio (%)	5,1761

PERCURSO 1 TOTAL		
Tempo (segundos)	133,36	
Consumo de Energia (Wh)	0,0142	

PERCURSO 2

Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,1146	Tensão Média (Volts)	5,3881
Corrente Média (Amperes)	0,4767	Corrente Média (Amperes)	0,5353
Tempo (segundos)	73,55	Tempo (segundos)	47,57
Tempo (horas)	0,0204	Tempo (horas)	0,0132
Consumo de Energia (Wh)	0,04	Consumo de Energia (Wh)	0,0381
		Porcentagem média de desvio (%)	4,5317

PERCURSO 2 TOTAL

Tempo (segundos)	121,13
Consumo de Energia (Wh)	0,0782

PERCURSO 3

Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,6635	Tensão Média (Volts)	4,1098
Corrente Média (Amperes)	0,5023	Corrente Média (Amperes)	0,4771
Tempo (segundos)	51,01	Tempo (segundos)	33,67
Tempo (horas)	0,0141	Tempo (horas)	0,0093
Consumo de Energia (Wh)	0,0331	Consumo de Energia (Wh)	0,0183
		Porcentagem média de desvio (%)	4,6911

PERCURSO 3 TOTAL

Tempo (segundos)	84,69
Consumo de Energia (Wh)	0,0515

PERCURSO 4

Regime Transitório		Regime Permanente	
Tensão Média (Volts)	6,1419	Tensão Média (Volts)	7,3296
Corrente Média (Amperes)	0,5273	Corrente Média (Amperes)	0,5241
Tempo (segundos)	75,56	Tempo (segundos)	39,19
Tempo (horas)	0,0209	Tempo (horas)	0,0108
Consumo de Energia (Wh)	0,0679	Consumo de Energia (Wh)	0,0418
		Porcentagem média de desvio (%)	4,1488

PERCURSO 4 TOTAL

Tempo (segundos)	114,76
Consumo de Energia (Wh)	0,1098

TOTAL TESTE 8

Tempo (segundos)	453,95
Consumo de Energia (Wh)	0,2538

Fonte: Autor

Teste 9

Neste teste foi utilizado um controlador Proporcional-Integrativo (PI) com método de sintonia Cohen-Coon. Os resultados obtidos encontram-se na Tabela 30.

Tabela 30 – Resultados de consumo de energia do controlador Proporcional-Integrativo com método de sintonia Cohen-Coon

PERCURSO 1			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	1,1768	Tensão Média (Volts)	1,7897
Corrente Média (Amperes)	0,2353	Corrente Média (Amperes)	0,2933
Tempo (segundos)	65,92	Tempo (segundos)	61,43
Tempo (horas)	0,0183	Tempo (horas)	0,017
Consumo de Energia (Wh)	0,005	Consumo de Energia (Wh)	0,0089
		Porcentagem média de desvio (%)	4,1406
PERCURSO 1 TOTAL			
Tempo (segundos)		127,35	
Consumo de Energia (Wh)		0,0140	

PERCURSO 2			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,05	Tensão Média (Volts)	5,2865
Corrente Média (Amperes)	0,4804	Corrente Média (Amperes)	0,5387
Tempo (segundos)	63,01	Tempo (segundos)	54,13
Tempo (horas)	0,0175	Tempo (horas)	0,0150
Consumo de Energia (Wh)	0,034	Consumo de Energia (Wh)	0,0428
		Porcentagem média de desvio (%)	3,3816
PERCURSO 2 TOTAL			
Tempo (segundos)		117,15	
Consumo de Energia (Wh)		0,0768	

PERCURSO 3			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,5142	Tensão Média (Volts)	3,9235
Corrente Média (Amperes)	0,5022	Corrente Média (Amperes)	0,4689
Tempo (segundos)	45,07	Tempo (segundos)	40,4
Tempo (horas)	0,0125	Tempo (horas)	0,0112
Consumo de Energia (Wh)	0,0283	Consumo de Energia (Wh)	0,02
		Porcentagem média de desvio (%)	4,6935
PERCURSO 3 TOTAL			
Tempo (segundos)		85,47	
Consumo de Energia (Wh)		0,0490	

PERCURSO 4			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	5,9270	Tensão Média (Volts)	7,1901
Corrente Média (Amperes)	0,5293	Corrente Média (Amperes)	0,5357
Tempo (segundos)	61,04	Tempo (segundos)	51,24
Tempo (horas)	0,0169	Tempo (horas)	0,0142
Consumo de Energia (Wh)	0,0532	Consumo de Energia (Wh)	0,0548
		Porcentagem média de desvio (%)	2,9021
PERCURSO 4 TOTAL			
Tempo (segundos)		112,29	
Consumo de Energia (Wh)		0,1080	

TOTAL TESTE 9	
Tempo (segundos)	442,28
Consumo de Energia (Wh)	0,2479

Fonte: Autor

Teste 10

Neste teste foi utilizado um controlador Proporcional-Integral (PI) com método de sintonia Ziegler-Nichols. Os resultados obtidos encontram-se na Tabela 31.

Tabela 31 – Resultados de consumo de energia do controlador Proporcional-Integrativo com método de sintonia Ziegler-Nichols

PERCURSO 1			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	1,5267	Tensão Média (Volts)	1,8656
Corrente Média (Amperes)	0,2496	Corrente Média (Amperes)	0,2878
Tempo (segundos)	4,68	Tempo (segundos)	95,01
Tempo (horas)	0,0013	Tempo (horas)	0,0263
Consumo de Energia (Wh)	0,0004	Consumo de Energia (Wh)	0,0141
		Porcentagem média de desvio (%)	0,4392
PERCURSO 1 TOTAL			
Tempo (segundos)		99,70	
Consumo de Energia (Wh)		0,0146	

PERCURSO 2

Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,5938	Tensão Média (Volts)	5,4325
Corrente Média (Amperes)	0,4271	Corrente Média (Amperes)	0,5252
Tempo (segundos)	1,85	Tempo (segundos)	97,39
Tempo (horas)	0,0005	Tempo (horas)	0,027
Consumo de Energia (Wh)	0,001	Consumo de Energia (Wh)	0,0772
		Porcentagem média de desvio (%)	0,6139

PERCURSO 2 TOTAL

Tempo (segundos)	99,25
Consumo de Energia (Wh)	0,0782

PERCURSO 3

Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,0235	Tensão Média (Volts)	3,6583
Corrente Média (Amperes)	0,4705	Corrente Média (Amperes)	0,4426
Tempo (segundos)	3,53	Tempo (segundos)	93,68
Tempo (horas)	0,0009	Tempo (horas)	0,026
Consumo de Energia (Wh)	0,0018	Consumo de Energia (Wh)	0,0421
		Porcentagem média de desvio (%)	0,3006

PERCURSO 3 TOTAL

Tempo (segundos)	97,22
Consumo de Energia (Wh)	0,0439

PERCURSO 4

Regime Transitório		Regime Permanente	
Tensão Média (Volts)	6,6094	Tensão Média (Volts)	7,3121
Corrente Média (Amperes)	0,4977	Corrente Média (Amperes)	0,5113
Tempo (segundos)	3,53	Tempo (segundos)	95,51
Tempo (horas)	0,0009	Tempo (horas)	0,0265
Consumo de Energia (Wh)	0,0032	Consumo de Energia (Wh)	0,0991
		Porcentagem média de desvio (%)	0,2827

PERCURSO 4 TOTAL

Tempo (segundos)	99,05
Consumo de Energia (Wh)	0,1024

TOTAL TESTE 10

Tempo (segundos)	395,23
Consumo de Energia (Wh)	0,2393

Fonte: Autor

Teste 11

Neste teste foi utilizado um controlador Proporcional-Integral-Derivativo (PID) com método de sintonia ISE. Os resultados obtidos encontram-se na Tabela 32.

Tabela 32 – Resultados de consumo de energia do controlador Proporcional-Integral-Derivativo (PID) com método de sintonia ISE

PERCURSO 1			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	1,2142	Tensão Média (Volts)	1,8165
Corrente Média (Amperes)	0,2161	Corrente Média (Amperes)	0,2834
Tempo (segundos)	26,43	Tempo (segundos)	82,65
Tempo (horas)	0,0073	Tempo (horas)	0,0229
Consumo de Energia (Wh)	0,0019	Consumo de Energia (Wh)	0,0118
		Porcentagem média de desvio (%)	1,088

PERCURSO 1 TOTAL

Tempo (segundos)	109,08
Consumo de Energia (Wh)	0,0137

PERCURSO 2

Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,0744	Tensão Média (Volts)	5,3242
Corrente Média (Amperes)	0,4717	Corrente Média (Amperes)	0,5316
Tempo (segundos)	24,13	Tempo (segundos)	81,03
Tempo (horas)	0,0067	Tempo (horas)	0,0225
Consumo de Energia (Wh)	0,0128	Consumo de Energia (Wh)	0,0637
		Porcentagem média de desvio (%)	0,8758

PERCURSO 2 TOTAL

Tempo (segundos)	105,17
Consumo de Energia (Wh)	0,0766

PERCURSO 3

Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,4592	Tensão Média (Volts)	3,6427
Corrente Média (Amperes)	0,4884	Corrente Média (Amperes)	0,4425
Tempo (segundos)	15,83	Tempo (segundos)	76,9889
Tempo (horas)	0,0043	Tempo (horas)	0,02
Consumo de Energia (Wh)	0,0095	Consumo de Energia (Wh)	0,0344
		Porcentagem média de desvio (%)	1,7414

PERCURSO 3 TOTAL

Tempo (segundos)	92,82
Consumo de Energia (Wh)	0,044

PERCURSO 4			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	5,7808	Tensão Média (Volts)	7,2332
Corrente Média (Amperes)	0,5171	Corrente Média (Amperes)	0,5138
Tempo (segundos)	22,66	Tempo (segundos)	81,16
Tempo (horas)	0,0062	Tempo (horas)	0,0225
Consumo de Energia (Wh)	0,0188	Consumo de Energia (Wh)	0,0837
		Porcentagem média de desvio (%)	1,0358
PERCURSO 4 TOTAL			
Tempo (segundos)		103,82	
Consumo de Energia (Wh)		0,1026	

TOTAL TESTE 11	
Tempo (segundos)	410,90
Consumo de Energia (Wh)	0,237

Fonte: Autor

Teste 12

Neste teste foi utilizado um controlador Proporcional-Integral-Derivativo (PID) com método de sintonia IAE. Os resultados obtidos encontram-se na Tabela 33.

Tabela 33 – Resultados de consumo de energia do controlador Proporcional-Integrativo com método de sintonia IAE

PERCURSO 1			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	1,2177	Tensão Média (Volts)	1,8271
Corrente Média (Amperes)	0,2077	Corrente Média (Amperes)	0,2787
Tempo (segundos)	37,39	Tempo (segundos)	76,28
Tempo (horas)	0,0103	Tempo (horas)	0,0211
Consumo de Energia (Wh)	0,0026	Consumo de Energia (Wh)	0,0107
		Porcentagem média de desvio (%)	1,79
PERCURSO 1 TOTAL			
Tempo (segundos)		113,67	
Consumo de Energia (Wh)		0,0134	

PERCURSO 2

Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,0268	Tensão Média (Volts)	5,3606
Corrente Média (Amperes)	0,4584	Corrente Média (Amperes)	0,5205
Tempo (segundos)	32,35	Tempo (segundos)	75,97
Tempo (horas)	0,0089	Tempo (horas)	0,0211
Consumo de Energia (Wh)	0,0165	Consumo de Energia (Wh)	0,0588
		Porcentagem média de desvio (%)	1,7761

PERCURSO 2 TOTAL

Tempo (segundos)	108,33
Consumo de Energia (Wh)	0,0754

PERCURSO 3

Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,4722	Tensão Média (Volts)	3,7026
Corrente Média (Amperes)	0,4827	Corrente Média (Amperes)	0,4416
Tempo (segundos)	25,63	Tempo (segundos)	64,70
Tempo (horas)	0,0071	Tempo (horas)	0,0179
Consumo de Energia (Wh)	0,0153	Consumo de Energia (Wh)	0,0293
		Porcentagem média de desvio (%)	2,5579

PERCURSO 3 TOTAL

Tempo (segundos)	90,34
Consumo de Energia (Wh)	0,0447

PERCURSO 4

Regime Transitório		Regime Permanente	
Tensão Média (Volts)	5,8637	Tensão Média (Volts)	7,24
Corrente Média (Amperes)	0,5102	Corrente Média (Amperes)	0,5088
Tempo (segundos)	34,06	Tempo (segundos)	72,03
Tempo (horas)	0,0094	Tempo (horas)	0,02
Consumo de Energia (Wh)	0,0283	Consumo de Energia (Wh)	0,073
		Porcentagem média de desvio (%)	1,3771

PERCURSO 4 TOTAL

Tempo (segundos)	106,09
Consumo de Energia (Wh)	0,102

TOTAL TESTE 12

Tempo (segundos)	418,45
Consumo de Energia (Wh)	0,2357

Fonte: Autor

Teste 13

Neste teste foi utilizado um controlador Proporcional-Integral-Derivativo (PID) com método de sintonia ITAE. Os resultados obtidos encontram-se na Tabela 34.

Tabela 34 – Resultados de consumo de energia do controlador Proporcional-Integrativo com método de sintonia ITAE

PERCURSO 1			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	1,2292	Tensão Média (Volts)	1,8303
Corrente Média (Amperes)	0,2146	Corrente Média (Amperes)	0,2799
Tempo (segundos)	41,70	Tempo (segundos)	73,18
Tempo (horas)	0,0115	Tempo (horas)	0,0203
Consumo de Energia (Wh)	0,003	Consumo de Energia (Wh)	0,0104
		Porcentagem média de desvio (%)	1,8669
PERCURSO 1 TOTAL			
Tempo (segundos)		114,88	
Consumo de Energia (Wh)		0,0134	
PERCURSO 2			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,1227	Tensão Média (Volts)	5,3939
Corrente Média (Amperes)	0,4653	Corrente Média (Amperes)	0,5237
Tempo (segundos)	37,39	Tempo (segundos)	71,64
Tempo (horas)	0,0103	Tempo (horas)	0,0199
Consumo de Energia (Wh)	0,0199	Consumo de Energia (Wh)	0,0562
		Porcentagem média de desvio (%)	1,6072
PERCURSO 2 TOTAL			
Tempo (segundos)		109,03	
Consumo de Energia (Wh)		0,0761	
PERCURSO 3			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,5555	Tensão Média (Volts)	3,7858
Corrente Média (Amperes)	0,4842	Corrente Média (Amperes)	0,4454
Tempo (segundos)	24,48	Tempo (segundos)	65,49
Tempo (horas)	0,0068	Tempo (horas)	0,0181
Consumo de Energia (Wh)	0,015	Consumo de Energia (Wh)	0,0306
		Porcentagem média de desvio (%)	3,253
PERCURSO 3 TOTAL			
Tempo (segundos)		89,97	
Consumo de Energia (Wh)		0,0456	

PERCURSO 4			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	5,9206	Tensão Média (Volts)	7,2462
Corrente Média (Amperes)	0,5086	Corrente Média (Amperes)	0,5077
Tempo (segundos)	36,99	Tempo (segundos)	69,74
Tempo (horas)	0,0102	Tempo (horas)	0,0193
Consumo de Energia (Wh)	0,0309	Consumo de Energia (Wh)	0,0712
		Porcentagem média de desvio (%)	1,3748
PERCURSO 4 TOTAL			
Tempo (segundos)		106,73	
Consumo de Energia (Wh)		0,1022	

TOTAL TESTE 13	
Tempo (segundos)	420,64
Consumo de Energia (Wh)	0,2375

Fonte: Autor

Teste 14

Neste teste foi utilizado um controlador Proporcional-Integral-Derivativo (PID) com método de sintonia Cohen-Coon. Os resultados obtidos encontram-se na Tabela 35.

Tabela 35 – Resultados de consumo de energia do controlador Proporcional-Integrativo com método de sintonia Cohen-Coon

PERCURSO 1			
Regime Transitório		Regime Permanente	
Tensão Média (Volts)	1,2372	Tensão Média (Volts)	1,8215
Corrente Média (Amperes)	0,2394	Corrente Média (Amperes)	0,2960
Tempo (segundos)	47,63	Tempo (segundos)	69,20
Tempo (horas)	0,0132	Tempo (horas)	0,0192
Consumo de Energia (Wh)	0,0039	Consumo de Energia (Wh)	0,0103
		Porcentagem média de desvio (%)	2,1365
PERCURSO 1 TOTAL			
Tempo (segundos)		116,83	
Consumo de Energia (Wh)		0,0142	

PERCURSO 2

Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,1125	Tensão Média (Volts)	5,4109
Corrente Média (Amperes)	0,4861	Corrente Média (Amperes)	0,5461
Tempo (segundos)	41,47	Tempo (segundos)	69,09
Tempo (horas)	0,0115	Tempo (horas)	0,0191
Consumo de Energia (Wh)	0,023	Consumo de Energia (Wh)	0,0567
		Porcentagem média de desvio (%)	2,0277

PERCURSO 2 TOTAL

Tempo (segundos)	110,56
Consumo de Energia (Wh)	0,0797

PERCURSO 3

Regime Transitório		Regime Permanente	
Tensão Média (Volts)	4,5543	Tensão Média (Volts)	3,8028
Corrente Média (Amperes)	0,505	Corrente Média (Amperes)	0,46
Tempo (segundos)	28,64	Tempo (segundos)	59,83
Tempo (horas)	0,0079	Tempo (horas)	0,0166
Consumo de Energia (Wh)	0,0183	Consumo de Energia (Wh)	0,029
		Porcentagem média de desvio (%)	4,0789

PERCURSO 3 TOTAL

Tempo (segundos)	88,48
Consumo de Energia (Wh)	0,0473

PERCURSO 4

Regime Transitório		Regime Permanente	
Tensão Média (Volts)	5,966	Tensão Média (Volts)	7,2725
Corrente Média (Amperes)	0,532	Corrente Média (Amperes)	0,5338
Tempo (segundos)	44,51	Tempo (segundos)	63,36
Tempo (horas)	0,0123	Tempo (horas)	0,0176
Consumo de Energia (Wh)	0,0392	Consumo de Energia (Wh)	0,0683
		Porcentagem média de desvio (%)	1,796

PERCURSO 4 TOTAL

Tempo (segundos)	107,88
Consumo de Energia (Wh)	0,1075

TOTAL TESTE 14

Tempo (segundos)	423,76
Consumo de Energia (Wh)	0,249

Fonte: Autor