



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**LARISSA PACHECO MONTENEGRO**

**CONTROLE DIGITAL DE VELOCIDADE DE UM MOTOR CC DE BAIXA  
POTÊNCIA UTILIZANDO ARDUINO**

**FORTALEZA**

**2021**

LARISSA PACHECO MONTENEGRO

CONTROLE DIGITAL DE VELOCIDADE DE UM MOTOR CC DE BAIXA POTÊNCIA  
UTILIZANDO ARDUINO

Trabalho de Conclusão de Curso apresentado ao curso de Graduação em Engenharia Elétrica da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Wilkley Bezerra Correia

FORTALEZA

2021

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

M783c Montenegro, Larissa Pacheco.  
Controle digital de velocidade de um motor CC de baixa potência utilizando arduino / Larissa Pacheco  
Montenegro. – 2021.  
59 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia,  
Curso de Engenharia Elétrica, Fortaleza, 2021.  
Orientação: Prof. Dr. Wilkley Bezerra Correia.

1. Motor CC. 2. Controle de velocidade. 3. Controle PI. 4. Arduino. I. Título.

CDD 621.3

---

LARISSA PACHECO MONTENEGRO

CONTROLE DIGITAL DE VELOCIDADE DE UM MOTOR CC DE BAIXA POTÊNCIA  
UTILIZANDO ARDUINO

Trabalho de Conclusão de Curso apresentado ao curso de Graduação em Engenharia Elétrica da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em: \_\_\_/\_\_\_/\_\_\_\_\_.

BANCA EXAMINADORA

---

Prof. Dr. Wilkley Bezerra Correia (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Diego de Sousa Madeira  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Victor de Paula Brandão Aguiar  
Universidade Federal do Ceará (UFC)

## **AGRADECIMENTOS**

Aos meus pais, Adalberto e Rita, por todo amor, por todo esforço, por toda dedicação para que eu pudesse prosseguir com meus objetivos e por todos os ensinamentos que hoje compõem meu caráter.

À minha irmã e cunhado, por acreditaram em mim, por todo suporte emocional e por me concederem a oportunidade de tornar meu sonho uma realidade.

A todos meus familiares, pelas conversas e momentos agradáveis.

Aos meus amigos e amigas, que estiveram juntos nesta trajetória, estudando, fazendo trabalhos, apoiando, compartilhando desde momentos felizes de descontração a períodos de intensa dedicação.

Ao Professor Dr. Wilkley Bezerra Correia pela orientação, por toda disponibilidade, pela paciência e competência durante a realização deste trabalho.

“Não importa o que fizeram com você. O que importa é o que você faz com aquilo que fizeram com você” (Jean Paul Sartre).

## RESUMO

Este trabalho apresenta umas das técnicas de implementação de estratégia de controle digital do tipo PI (proporcional – integral), para regular a velocidade de um motor de corrente contínua de baixa potência, fazendo uma análise da modulação de pulsos (PWM), utilizando Arduino e ponte H. Primeiramente, obteve-se o modelo matemático do motor CC. Em seguida foram realizados alguns testes práticos para adquirir a resposta degrau da planta, com intuito de obter a função de transferência. Posteriormente foi realizada a sintonização do controlador proporcional e integral, que foi implementado em código via Arduino. Ao final testou-se a planta com controle implementado, confirmando que o compensador desenvolvido funciona de forma eficiente no controle do motor escolhido para a realização deste trabalho.

**Palavras-chave:** Motor CC; Controle de velocidade; Controle PI; Arduino.

## ABSTRACT

This paper presents a methodology to develop a proportional and integral (PI) controller to regulate the speed of a low potency DC- motor, making an analysis of pulse width modulation (PWM), using Arduino and H-bridge. Work began with the mathematical model of the DC engine. Afterwards a few practical tests were made to acquire the step response to the plant, with the intention of obtaining the transference function. Subsequently the syntonization of the proportional and integral controller was performed, implemented in code by Arduino. Finally, the plant was tested with the implemented control, confirming that the developed compensator works efficiently in the control of the engine selected for this work.

**Keywords:** DC engine; Velocity control; PI control; Arduino.



## LISTA DE FIGURAS

Figura 1 – Sistema do Controlador.....	17
Figura 2 – <i>Layout</i> da placa e pinos do Arduino UNO.....	18
Figura 3 – Modulação de Pulso.....	19
Figura 4 – Disco <i>Encoder</i> .....	20
Figura 5 – Sensor de Velocidade <i>encoder</i> .....	21
Figura 6 – <i>Motor Shield</i> L293D <i>driver</i> ponte H.....	22
Figura 7 – Partes Integrantes do Motor.....	23
Figura 8 – Representação esquemática do motor CC.....	24
Figura 9 – Motor CC com caixa de redução.....	25
Figura 10 – Esquemático de armadura de uma máquina CC com comutador e escova.....	26
Figura 11 – Representação do funcionamento do motor CC.....	27
Figura 12 – Representação de um sistema de controle.....	30
Figura 13 – Representação de um sistema de controle em malha aberta.....	31
Figura 14 – Representação de um sinal de controle em malha aberta.....	31
Figura 15 – Representação gráfica de (a) sinais de tempo contínuo e (b) sinais de tempo discreto.....	32
Figura 16 – Diagrama de blocos para sistema de controle com computador digital.....	33
Figura 17 – Diagrama de blocos controlador PI.....	34
Figura 18 – Resposta degrau.....	35
Figura 19 – Gráfico de modelo matemático obtido.....	36
Figura 20 – Modelo matemático associado a entrada degrau.....	38
Figura 21 – Informações da resposta degrau.....	40
Figura 22 – <i>Layout</i> do GUI SISO <i>Disign Tool</i> .....	41
Figura 23 – <i>Layout</i> do <i>compensator editor</i> .....	42
Figura 24 – Gráfico <i>Root locus</i> e compensador projetado.....	43
Figura 25 – Resposta degrau do controlador C.....	43
Figura 26 – Características da resposta degrau do controlador C.....	44
Figura 27 – Diagrama de blocos controlador digital.....	46
Figura 28 – Resposta degrau do controlador.....	48
Figura 29 – Informações da resposta degrau.....	49
Figura 30 – Esquemático do circuito implementado.....	49
Figura 31 – Resposta do controlador PI para 150 rpm.....	50

Figura 32 – Resposta do controlador PI .....	51
Figura 33 – Resposta do controlador PI para dois valores de referência .....	52
Figura 34 – Comportamento do controlador PI para dois valores de referência.....	53
Figura 35 – Circuito utilizado.....	57
Figura 36 – Conexões do Motor <i>Shield</i> .....	57
Figura 37 – Conexões do Motor <i>Shield</i> no Arduino.....	58
Figura 38 - Bancada.....	58

## LISTA DE TABELAS

Tabela 1 – Especificações técnicas Arduino Uno .....	14
Tabela 2 – Especificações técnicas Sensor de Velocidade <i>Encoder</i> .....	17
Tabela 3 – Especificações técnicas <i>Motor Shield</i> L293D .....	18
Tabela 4 – Especificações técnicas motor CC com caixa de redução .....	21
Tabela 5 – Dados de tempo e velocidade .....	31

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>14</b>
<b>1.1 Justificativa</b> .....	<b>15</b>
<b>1.2 Objetivos</b> .....	<b>15</b>
<i>1.2.1 Objetivo geral</i> .....	<i>15</i>
<i>1.2.2 Objetivos específicos</i> .....	<i>15</i>
<b>1.3 Organização do Trabalho</b> .....	<b>15</b>
<b>2 DESCRIÇÃO</b> .....	<b>17</b>
<b>2.1 Arduino</b> .....	<b>17</b>
<b>2.2 Modulação Por Largura de Pulso (PWM)</b> .....	<b>18</b>
<b>2.3 Verificação da velocidade</b> .....	<b>19</b>
<i>2.3.1 Encoder</i> .....	<i>20</i>
<i>2.3.2 Sensor de medição de velocidade</i> .....	<i>20</i>
<b>2.4 Ponte H</b> .....	<b>22</b>
<b>2.5 Motores elétricos</b> .....	<b>23</b>
<i>2.5.1 Motor de corrente contínua</i> .....	<i>24</i>
<i>2.5.2 Princípio de funcionamento</i> .....	<i>26</i>
<b>2.6 Modelagem da planta</b> .....	<b>27</b>
<b>3 PROJETO DO CONTROLADOR</b> .....	<b>30</b>
<b>3.1 Sistemas de Controle</b> .....	<b>30</b>
<i>3.1.1 Sistemas de Controle em Malha Aberta</i> .....	<i>30</i>
<i>3.1.2 Sistemas de Controle em Malha Fechada</i> .....	<i>31</i>
<i>3.1.3 Sistemas de Controle de Tempo Contínuo e Discreto</i> .....	<i>32</i>
<i>3.1.4 Sistema de Controle Digital</i> .....	<i>33</i>
<i>3.1.5 Controle PID</i> .....	<i>33</i>
<b>3.2 Sintonização do Compensador</b> .....	<b>34</b>
<i>3.2.1 Aquisição de dados</i> .....	<i>34</i>
<i>3.2.2 Obtenção da Função de Transferência</i> .....	<i>36</i>
<i>3.2.3 Projeto do Controlador PI Digital</i> .....	<i>40</i>
<b>4 RESULTADOS</b> .....	<b>47</b>
<b>4.1 Simulações</b> .....	<b>47</b>
<b>4.2 Resultados Práticos</b> .....	<b>49</b>
<b>5 CONCLUSÕES</b> .....	<b>54</b>

<b>REFERÊNCIAS .....</b>	<b>55</b>
<b>ANEXO A – IMAGENS DA BANCADA .....</b>	<b>57</b>
<b>ANEXO B – PROGRAMA COMENTADO .....</b>	<b>59</b>

## 1 INTRODUÇÃO

Sistema de controle para engenharia consiste na teoria da realimentação no qual uma variável controlada, como temperatura ou velocidade, é medida por um sensor e as informações são retransmitidas ao controlador, com o intuito de manipular a variável controlada (Franklin, 2013). Existem inúmeros sistemas físicos nos quais é possível implementar a engenharia de controle. Por exemplo, o projeto de um compensador para controle de posição azimutal de uma antena de radar meteorológico a partir da atuação de um motor de corrente contínua, (Castrucci, 2011) ou o projeto para controle de temperatura do forno elétrico (Ogata, 2013).

Dentre os diversos tipos de sistemas de controle, no presente trabalho será implementado o sistema de controle digital. Segundo Dorf (2001), “Um sistema de controle digital usa sinais digitais e um computador digital para controlar processos.”, ou seja, o controle por computador funciona da seguinte forma: ele recebe o sinal de realimentação, realiza a correção do erro e envia o sinal corrigido para a planta a ser controlada. Sistemas de controle automático são cada vez mais comuns no dia a dia da sociedade moderna. O desenvolvimento de tecnologias, principalmente dos microcontroladores, associado a placas de desenvolvimento, permitiu a realização de projetos de sistemas mais simples de forma rápida e com baixo custo. Está presente em diversas aplicações, por exemplo, o Robô *Rover* (*Robot Rover*) desenvolvido por alunos do curso de Técnico Eletrônica, Automação e Comando do Instituto de Educação Técnica (INETE)<sup>1</sup> que utilizou a plataforma de prototipagem eletrônica Arduíno Mega.

Tendo em mente as considerações acima, propõe-se neste trabalho o desenvolvimento de um projeto de engenharia de controle digital visando influenciar a velocidade do motor de corrente contínua de baixa potência, utilizando o controlador do tipo proporcional-integral (PI) e a placa de desenvolvimento do Arduíno.

---

<sup>1</sup> Disponível em: Robot Rover para Exploração do Ambiente Envolve. Disponível em: [https://www.researchgate.net/publication/304657545\\_Robot\\_Rover\\_para\\_Exploracao\\_do\\_Ambiente\\_Envolve](https://www.researchgate.net/publication/304657545_Robot_Rover_para_Exploracao_do_Ambiente_Envolve). Acessa em 28 ago. 2021.

## **1.1 Justificativa**

Levando em consideração o que foi exposto anteriormente, o estudo dos sistemas de controle para engenharia, em especial, o sistema de controle digital, é essencial para o entendimento dos processos de automação. Sendo assim, o projeto desenvolvido neste trabalho uma excelente aplicação do controle automático, que representa inúmeras aplicações reais. Além disso, com a implementação deste projeto, é possível analisar o comportamento de uma planta automatizada utilizando uma plataforma didática de baixo custo e de fácil implementação.

## **1.2 Objetivos**

Nesta seção, serão apresentados os objetivos deste trabalho.

### ***1.2.1 Objetivo geral***

O objetivo geral deste trabalho é controlar o motor de corrente contínua de baixa potência através da implementação de estratégia de controle digital do tipo PI (proporcional-integral) e visualizar na prática todos os desenvolvimentos teóricos discutidos neste projeto.

### ***1.2.2 Objetivos específicos***

Podem ser destacados os objetivos específicos a seguir:

1. Realizar a modelagem do motor CC para obter a função de transferência;
2. Desenvolver códigos no Arduino para a obtenção de dados e controle da planta;
3. Projetar um controlador proporcional integral;
4. Implementar de forma didática a planta analisada;
5. Analisar os resultados obtidos com a implementação do controlador.

## **1.3 Organização do Trabalho**

Este trabalho está dividido em cinco capítulos. O primeiro apresenta uma introdução ao tema, sua importância, bem como sua justificativa, sua motivação e seus objetivos.

O segundo capítulo apresenta a descrição de cada elemento utilizado neste projeto, assim como a modelagem da planta empregada.

O terceiro capítulo apresenta a fundamentação teórica sobre sistemas de controle essenciais para o entendimento do estudo realizado neste trabalho. Além disso, este capítulo abrange os procedimentos utilizados para o desenvolvimento do controlador digital.

O quarto capítulo é destinado a analisar os resultados de simulações, bem como os resultados práticos obtidos durante os testes realizado no sistema.

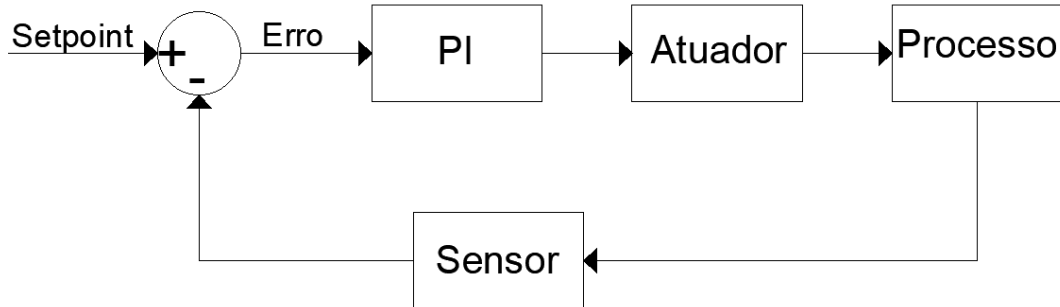
O quinto capítulo apresenta as conclusões deste trabalho e sugestões para trabalhos futuros.



## 2 DESCRIÇÃO

Neste capítulo serão apresentados os elementos presentes na estrutura de controle aplicado neste estudo. A figura 1 abaixo mostra o sistema de controle em malha fechada.

Figura 1 – Sistema controlado



Fonte: Próprio Autor.

A partir da análise da figura 1, nota-se que, para este projeto, o sistema de controle será composto principalmente por o compensador PI, que será implementado utilizando o Arduino, por um atuador que consiste na ponte H acionada por PWM, por um sensor que será executado pelo conjunto: *encoder* + sensor de velocidade e, por fim, o processo representado pelo motor corrente contínua (CC).

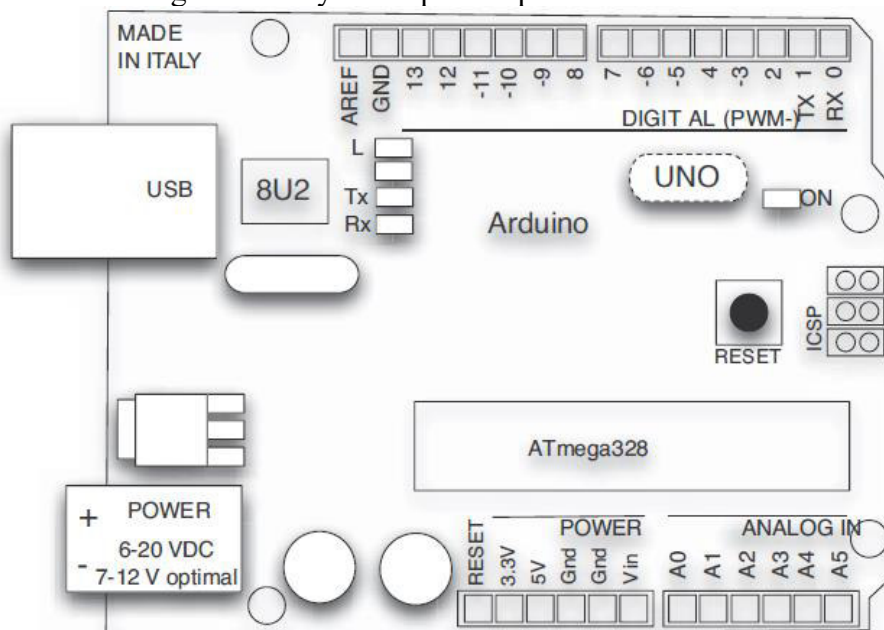
### 2.1 Arduino

Arduino é uma plataforma *open-source* (código aberto) de baixo custo e fácil manipulação, com software baseado na linguagem C/C++. Esta plataforma foi planejada para auxiliar estudantes italianos de design e arte em seus projetos acadêmicos (HOCHENBAUM, 2013), porém rapidamente ganhou destaque na comunidade acadêmica. Atualmente, o Arduino é utilizado no mundo inteiro, tanto por universitários, quanto por desenvolvedores, profissionais ou entusiastas, interessados no levantamento e na prototipagem de seus próprios projetos.

Hoje em dia existem diversas versões de hardware Arduino, possuindo variações que vão do tipo de controlador implementado até o número de portas disponíveis, de modo a atender diversos tipos de projetos.

Na figura 2 abaixo, mostra-se o *layout* do hardware e, na tabela 1, apresentam-se as especificações do Arduino utilizado.

Figura 2 – Layout da placa e pinos do Arduino Uno



Fonte: HOCHENBAUM (2013, p.27).

Tabela 1 – Especificações técnicas Arduino UNO

Especificações	
Microcontrolador	ATmega328P
Tensão de operação	5V
Tensão de entrada	7-12V
Tensão de entrada máxima	6-20V
Corrente de operação para pino 3.3V	50mA
Corrente de operação máxima por pino I/O	20mA
Pinos digitais	14
Pinos Analógicos	6
Pinos digitais para PWM	6
Memória flash	32 KB

Fonte: Próprio Autor.

## 2.2 Modulação Por Largura de Pulso (PWM)

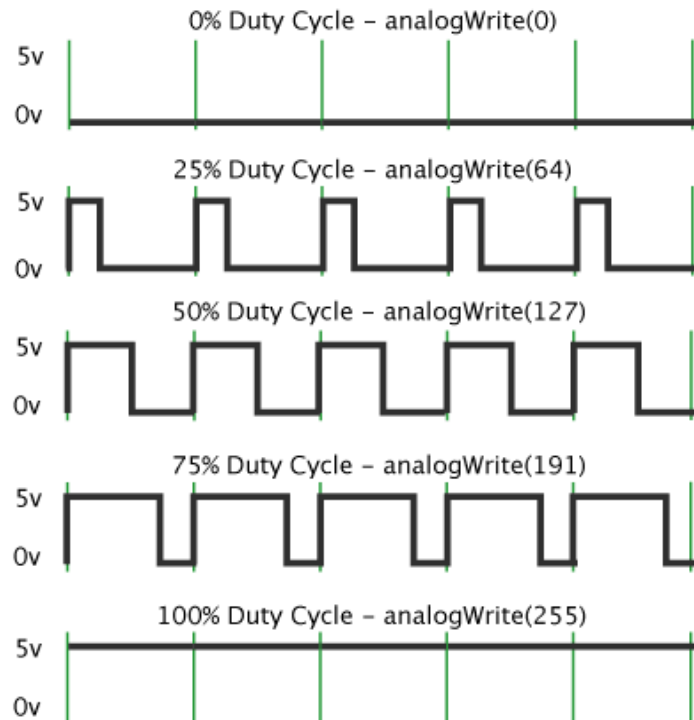
De acordo com Hart (2012) a técnica de modulação por largura de pulso (em inglês *pulse width modulation*, PWM), consiste em um sistema de pulsos em que a frequência de uma onda quadrada permanece constante, enquanto o tempo em que o sinal fica em nível lógico alto (*duty cycle*) pode ser modificado, tal que o valor médio da onda quadrada é variável entre zero

e a tensão máxima da onda quadrada. Assim, o valor da potência entregue à carga dependerá da largura do pulso.

Sabendo que no Arduino a modulação por largura de pulso é a responsável por realizar o ajuste de velocidade do motor, é imprescindível o entendimento do seu funcionamento para a realização deste trabalho. Diante disso, foi verificado que, para o Arduino UNO, o PWM varia entre 0 e 255, sendo 0 para um *duty cycle* de 0%(0V) e 255 para *duty cycle* de 100% (5V).

A figura abaixo representa a modulação de pulso para diferentes valores *duty cycle*.

Figura 3 – Modulação de pulso



Fonte: Arduino<sup>2</sup>

### 2.3 Verificação da velocidade

Para realizar o controle da planta estudada, é preciso conhecer o valor do erro atuante. Este valor pode ser obtido encontrando-se a diferença entre o sinal de referência de velocidade desejada e o valor de velocidade efetivamente desenvolvido pelo motor, medido pelo sensor de velocidade.

<sup>2</sup> ARDUINO CC. **PWM**, 2018. Disponível em: <https://www.arduino.cc/en/Tutorial/Foundations/PWM>. Acesso em: 28 jul. 2021.

Para este trabalho, a variável de saída é a velocidade do motor, sendo assim necessário realizar sua medição para prosseguir com o projeto de controle do processo. Diante disso, a medição do sinal de saída foi realizada por dois componentes: disco *encoder* e sensor de velocidade.

### 2.3.1 *Encoder*

O *encoder* é um dispositivo eletromecânico utilizado, principalmente, para definir o número de rotações de um objeto e o posicionamento de determinados equipamentos. Esse equipamento é constituído por um disco com perfurações. Ao ser rotacionado e submetido a um feixe de luz infravermelha constante, as perfurações permitem a passagem do feixe de luz, ao passo que as partes íntegras do disco impossibilitam a transmissão desta luz. Esse processo resulta na transformação de uma posição em um sinal digital.

Neste projeto foi empregado um disco *encoder* (figura 4) simples, com apenas 20 perfurações, conectado à caixa de redução do motor.

Figura 4 – Disco *encoder*



Fonte: Autocore Robótica<sup>3</sup>

### 2.3.2 *Sensor de medição de velocidade*

O sensor de velocidade *encoder* (figura 5) é um módulo eletrônico. O sensor é dotado de quatro pinos, a saber, entrada de 5V, terra, saída digital (D0) e saída analógica (A0).

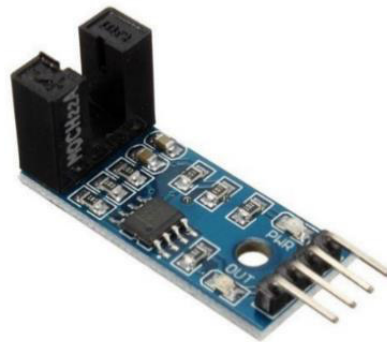
---

<sup>3</sup> AUTOCORE ROBOTICA. **Disco Encoder ideal para sensor de velocidade**. 2021. Disponível em: <https://www.autocorerobotica.com.br/disco-encoder-ideal-para-sensor-de-velocidade>. Acesso em: 23 jul. 2021.

Os pinos de alimentação do sensor foram conectados à saída do servo motor da ponte H, recebendo uma tensão de 5V. O pino de saída digital (D0) foi conectado à entrada digital do pino 2 do Arduino. O pino de saída analógico não foi utilizado neste estudo.

O sensor de medição de velocidade é responsável por fazer a leitura das perfurações do disco *encoder*, interpretando a luz e sua ausência como uma sequência de sinais, ora nível lógico alto, ora nível lógico baixo. O sensor entrega esse conjunto de rotações por unidade de tempo como onda quadrada. A onda é enviada pela saída digital (D0) para o microcontrolador do Arduino.

Figura 5 – Sensor de velocidade *encoder*



Fonte: Eletrogate<sup>4</sup>

Tabela 2 – Especificações técnicas Sensor de velocidade *encoder*

Especificações	
Controlador	LM393
Tensão de operação	3,3 - 5VDC
Saída Digital	1
Saída Analógica	1
LED indicador para presença	Tensão
LED indicador para	Saída digital

Fonte: Próprio Autor

<sup>4</sup> ELETROGATE. **Sensor de velocidade Encoder**. 2021. Disponível em: [https://www.eletrogate.com/sensor-de-velocidadeencoder?gclid=CjwKCAjwr56IBhAvEiwA1fuqGujdA4lRqh9PjgLohYyXGXWhFk4Dbm2sfCdWAUxAEkHtF11h0BuQxoC9J4QAvD\\_BwE](https://www.eletrogate.com/sensor-de-velocidadeencoder?gclid=CjwKCAjwr56IBhAvEiwA1fuqGujdA4lRqh9PjgLohYyXGXWhFk4Dbm2sfCdWAUxAEkHtF11h0BuQxoC9J4QAvD_BwE). Acesso em: 23 jul. 2021.

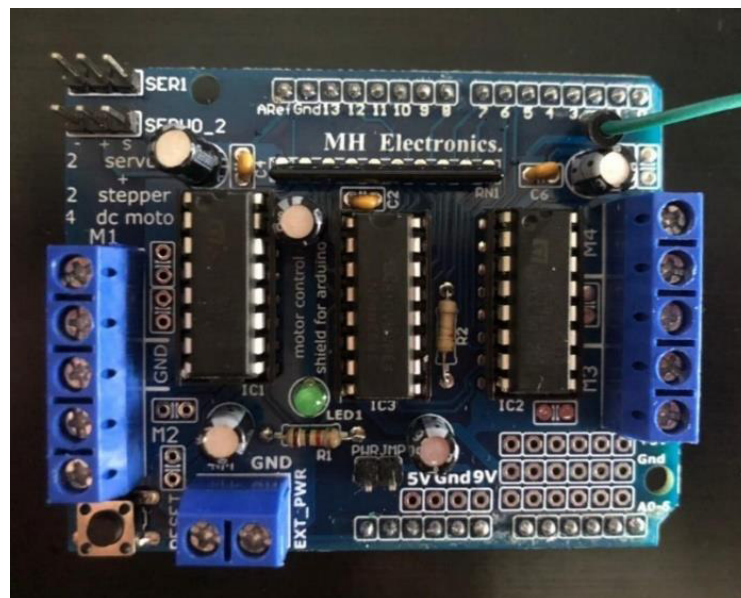
## 2.4 Ponte H

Neste trabalho, utilizou-se circuito de ponte H para garantir o funcionamento do motor CC sem danificar o *hardware* do Arduino. O circuito de ponte H se faz necessário, já que a corrente necessária para o acionamento do motor é maior que aquela que pode ser obtida diretamente nas portas de saída do Arduino, tornando inviável a conexão direta entre o motor e o Arduino. Dessa forma, o acionamento da máquina rotativa de corrente contínua foi dividido em dois circuitos, um com uma fonte externa para alimentar o motor com a tensão necessária para seu funcionamento e outro para controlar seu comportamento.

Para o presente projeto foi utilizado o *Motor Shield L293D*, *hardware* composto por dois circuitos integrados L293D, podendo controlar até quatro motores de corrente contínua de forma independente, e um registrador de oito bits (74H595), utilizado como expensor de portas para o Arduino. O *motor shield* também pode controlar até dois servos motores ou dois motores de passo.

A figura 6 mostra com mais detalhes o *hardware* do *driver* ponte H e a tabela 3 apresenta suas especificações.

Figura 6 – Motor shield L293D *driver* ponte H



Fonte: Próprio Autor

Tabela 3 – Especificações técnicas Motor shield L293D

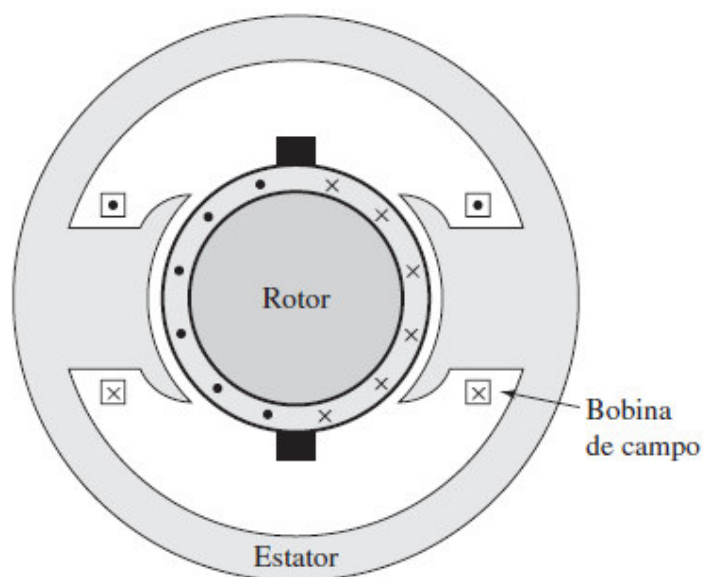
Especificações técnicas	
Chip	293D
Corrente de Saída	600mA por canal
Tensão de Saída	4,5 - 36 V
Controle	Até 4 Motores DC, 2 Motores de Passo ou 2 Servos
4 Pontes H	0,6A por ponte (1,2 de pico)
Terminais	Em bloco de 2 pinos e jumper para alimentação externa

Fonte: Próprio Autor

## 2.5 Motores elétricos

Motores elétricos são máquinas rotativas que transformam energia elétrica em energia mecânica partindo do princípio da conservação de energia. Os motores elétricos são constituídos, basicamente, por dois elementos principais: o estator, parte fixa, e o rotor, parte rotativa. A figura 7 ilustra os elementos principais do motor citados anteriormente.

Figura 7– Vista



Fonte: UMANS (2014, p. 215).

O princípio de funcionamento das máquinas elétricas rotativas é explicado pela Lei de Lenz-Faraday. O motor elétrico funciona pelo fornecimento de energia elétrica a parte estática (estator) do conjunto, que gera um campo magnético variante no tempo e faz o estator funcionar como um eletroímã. Esse comportamento gera uma movimentação na parte móvel do motor (rotor), que acompanha a variação do campo magnético.

Os motores elétricos podem ser classificados em:

- Motores de Corrente Contínua;
- Motores de Corrente Alternada;
- Motores Especiais.

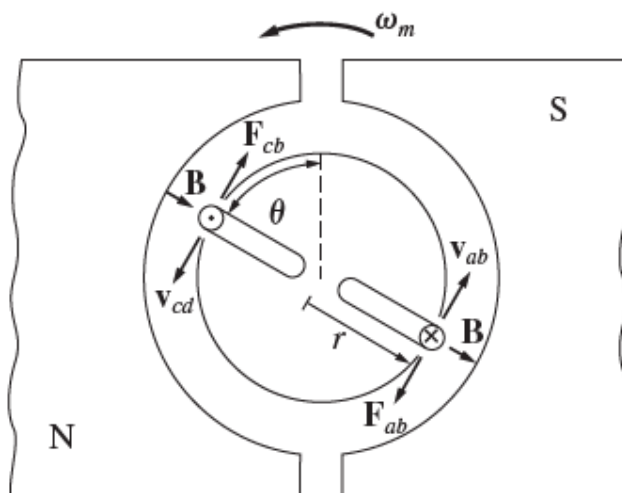
Para a realização deste projeto foi escolhido um motor CC, assim esse tipo de máquina elétrica rotativa será melhor detalhada a seguir.

### 2.5.1 Motor de corrente contínua

A máquinas rotativas de corrente contínua são acionadas por corrente contínua, caracterizando um sistema com campo magnético contínuo, porém com fluxo variável. Chapman (2013) afirma:

“O campo magnético é constante e perpendicular à superfície do rotor em todos os pontos debaixo das faces polares e rapidamente cai a zero além das bordas dos polos. (...) Assim, a tensão gerada na máquina é igual ao produto do fluxo presente no interior da máquina vezes a velocidade de rotação da máquina, multiplicado por uma constante que representa os aspectos construtivos da máquina.”

Figura 8 – Representação esquemática do motor cc



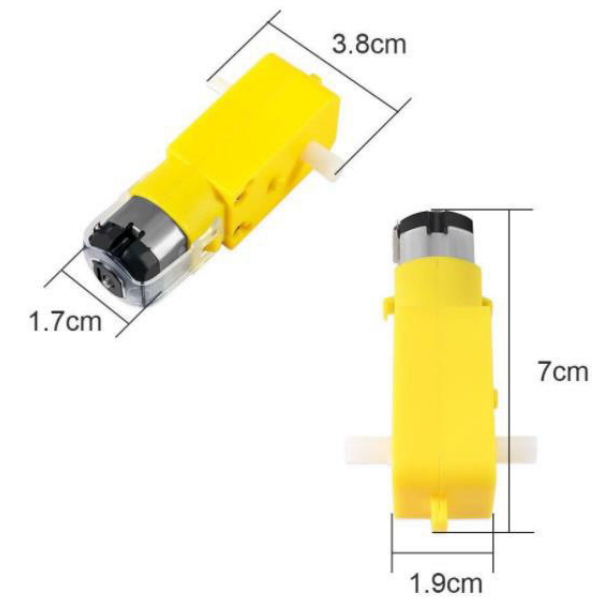
Fonte: Chapman (2013, p. 406).



Os motores de correntes contínua, além de possuírem estator e rotor, também são constituídos por comutador e escovas. Essa máquina rotativa é comumente empregada em sistemas de malha fechada quando é pretendido realizar controle de velocidade, posição e reversão de sentido de giro de forma rápida e simples. Além disso, esses motores também são aplicados em projetos em que se deseja obter alto conjugado de partida (torque), por exemplo, veículos elétricos de golfe, que precisam de uma força de arranque para superar a inércia

Neste trabalho, foi utilizado um mini motor CC com caixa de redução, eixo duplo e alimentação de 6 V conforme ilustrado na figura 9:

Figura 9 – Motor CC com caixa de redução



Fonte: Fábrica de Bolso<sup>5</sup>

<sup>5</sup>FÁBRICA DE BOLSO. **Motor Dc 3-6v com caixa de redução p/ robótica educativa**. 2021. Disponível em: [https://loja.fabricadebolso.com.br/motor-dc-3-6v-com-caixa-de-reducao-p-roboticaeducativa?utm\\_source=Site&utm\\_medium=GoogleMerchant&utm\\_campaign=GoogleMerchant&gclid=CjwKCAjw0qOIBhBhEiwAyvVcf1H8fyTXgXN50nG2t-ui-n7uPnB9WkGaSdWnp584-c7IXeV\\_UbVJchoCQ\\_QQAvD\\_BwE](https://loja.fabricadebolso.com.br/motor-dc-3-6v-com-caixa-de-reducao-p-roboticaeducativa?utm_source=Site&utm_medium=GoogleMerchant&utm_campaign=GoogleMerchant&gclid=CjwKCAjw0qOIBhBhEiwAyvVcf1H8fyTXgXN50nG2t-ui-n7uPnB9WkGaSdWnp584-c7IXeV_UbVJchoCQ_QQAvD_BwE). Acesso em: 08 de ago. 2021.

Tabela 4 – Especificações técnicas Motor CC com caixa de redução

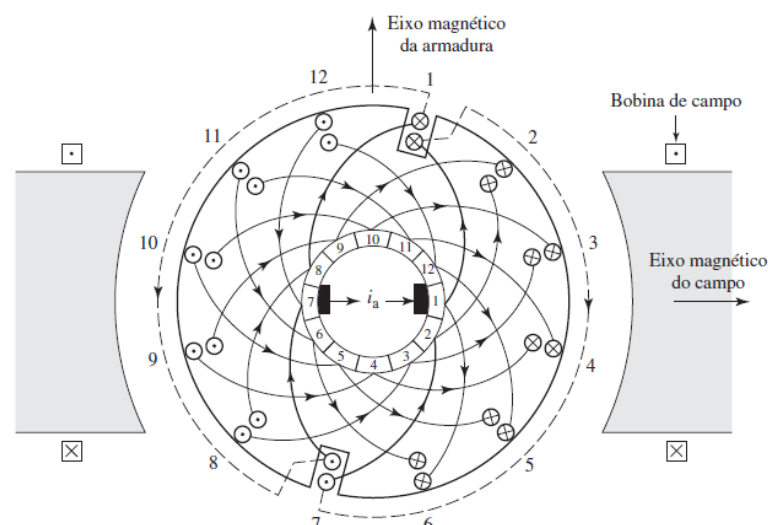
Especificações técnicas	
Tensão Nominal	6V
Tensão de Operação	3V - 6V
Torque	343 N/m (3V) e 785 N/m (6V)
Corrente sem Carga	150mA(3V) e 150mA(6V)
Velocidade sem Carga	90 RPM (3V) e 200 RPM (6V)
Redução	1,48

Fonte: Próprio Autor.

### 2.5.2 Princípio de funcionamento

O funcionamento dos motores de corrente contínua, de acordo com Umans (2014) pode ser explicado a partir da análise da armadura e do anel de comutação com as escovas. Ele afirmar que: “A combinação de comutador e escovas forma um retificador mecânico, resultando em uma tensão CC de armadura e uma onda de força magnetomotriz de armadura que está fixa no espaço.” Assim, a tensão de corrente alternada gerada é convertida em tensão de corrente contínua. O estator funciona como um eletroímã de polaridades fixas, então quando as escovas entram em contato com o comutador a corrente passa pelos enrolamentos de bobina. Devido à ação das forças magnéticas entre a parte estática e a parte móvel do motor, o rotor começa a girar buscando nova condição de equilíbrio, por meio do deslocamento angular.

Figura 10 – Esquemático de armadura de uma máquina CC com comutador e escova



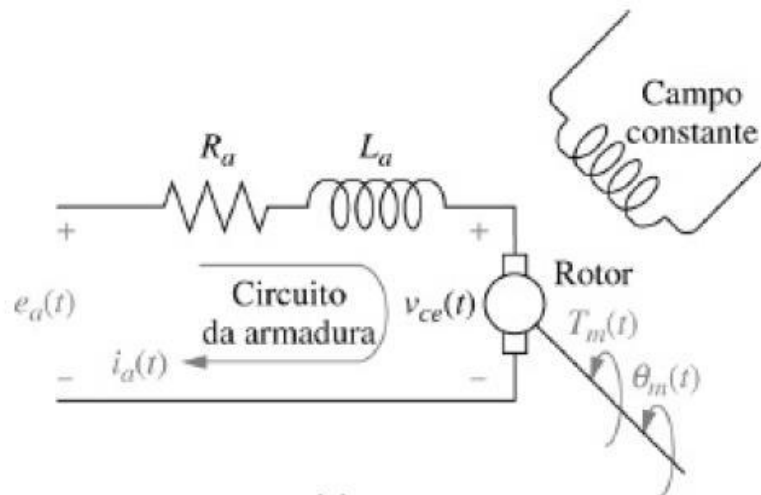
Fonte: UMANS (2014, p. 412).

## 2.6 Modelagem da planta

Para realizar o projeto do compensador de forma adequada é necessário obter a função de transferência da planta. Assim, foi desenvolvido um modelo matemático em malha fechada representando o sistema físico do motor de corrente contínua.

Partindo dos princípios de funcionamento das máquinas elétricas rotativas de corrente contínua e analisando a figura 11, é possível encontrar a função de transferência para a planta estudada neste trabalho.

Figura 11 – Representação do funcionamento do motor cc



Fonte: NISE (2013, p. 129).

Nota-se que, o enrolamento da armadura do motor é representado por uma resistência ( $R_a$ ) em série com uma indutância ( $L_a$ ), enquanto a tensão gerada na armadura é caracterizada por  $V_{ce}(t)$ , o deslocamento angular por  $\Theta_m(t)$ , o torque desenvolvido pelo motor por  $T_m(t)$  e a força eletromotriz por  $e_a(t)$ .

Segundo Nise (2013, p. 128),  $e_a(t)$  é a tensão resultante do movimento ortogonal do condutor a um campo magnético. Diante disso, pode-se afirmar que a corrente de armadura  $i_a(t)$  é proporcional ao torque. Assim,

$$V_{ce}(t) = K_{ce} \frac{d\Theta_m(t)}{dt}, \quad (2.1)$$

$$T_m(t) = K_m i_a(t) \quad (2.2)$$

em que  $K_{ce}$  e  $K_m$  constantes de proporcionalidade chamadas, respectivamente, de constante de força eletromotriz e constante de torque do motor; e a derivada de  $\Theta_m(t)$  é a velocidade angular ( $\omega(t)$ ).

Analisando o circuito de armadura usando a lei de Kirchhoff, tem-se a seguinte equação:

$$e_a(t) = Rai_a(t) + La \frac{di_a(t)}{dt} + V_{ce}(t), \quad (2.3)$$

Aplicando a Lei de Newton, tem-se:

$$T_m(t) - f \frac{d\Theta_m(t)}{dt} = J \frac{d^2\Theta_m(t)}{dt^2} \quad (2.4)$$

em que  $f$  é o coeficiente de atrito viscoso nos mancais e  $J$  é o momento de inércia da carga.

Ao aplicar a transformada de Laplace nos dois lados das equações (2.1), (2.2), (2.3), (2.4), obtém-se as seguintes equações:

$$\bullet V_{ce}(s) = K_{ce}s\Theta(s), \quad (2.5)$$

$$\bullet T_m(s) = K_m i_a(s), \quad (2.6)$$

$$\bullet e_a(s) = Rai_a(s) + sLa i_a(s) + V_{ce}(s), \quad (2.7)$$

$$\bullet T_m(s) - sf\Theta(s) = s^2J \Theta(s). \quad (2.8)$$

Fazendo algumas substituições é possível encontrar o modelo matemático do motor CC a partir das equações de transferência desenvolvidas acima.

Isolando  $\Theta(s)$  da equação (2.5) e  $V_{ce}(s)$  da (2.7), tem-se:

$$\Theta(s) = \frac{V_{ce}}{K_{ce}s} \quad (2.9)$$

$$V_{ce}(s) = e_a(s) - i_a(s)(Ra + sLa) \quad (2.10)$$

Substituindo (2.10) em (2.9), tem-se

$$\Theta(s) = \frac{e_a(s) - i_a(s)(Ra + sLa)}{K_{ce}s} \quad (2.11)$$

Sendo  $i_a(s) = \frac{T_m(s)}{K_m}$ , então:

$$\Theta(s) = \frac{e_a(s) - \frac{T_m(s)}{K_m}(Ra + sLa)}{K_{ce}s} \quad (2.12)$$

Substituindo  $T_m(s)$  por (2.8), é obtido:

$$\Theta(s) = \frac{e_a(s) - \frac{s\Theta(s)(sJ + f)}{K_m}(Ra + sLa)}{K_{ce}s} = \frac{K_m e_a(s)}{s[K_{ce}K_m + (Ra + sLa)(f + sJ)]} \quad (2.13)$$

Sabendo que,  $\frac{La}{Ra}$  é a constante de tempo elétrica ( $\tau_e$ ) e  $\frac{J}{f}$  é a constante de tempo mecânica ( $\tau_m$ ) é possível fazer as seguintes substituições em (2.13):

$$\Theta(s) = \frac{K_m e_a(s)}{s[K_{ce}K_m + fRa(\tau_m s + 1)(\tau_e s + 1)]} \quad (2.14)$$

Nota-se que, a constante de tempo elétrica ( $\tau_e s + 1$ ), neste caso, pode ser considerada nula, pois a indutância de armadura é muito pequena, assim (2.13) ficará igual a:

$$\theta(s) = \frac{K_m e_a(s)}{s[K_{ce}K_m + fRa(\tau_m s + 1)]} = \frac{K_m e_a(s)}{s[K_{ce}K_m + fRa(f + s)]} \quad (2.15)$$

Substituindo,

$$\tau = \frac{Ra}{K_{ce}K_m + fRa},$$

$$K = \frac{K_m}{K_{ce}K_m + fRa},$$

na equação (2.14), encontra-se:

$$\theta(s) = \frac{K e_a(s)}{s(\tau s + 1)} \Rightarrow \frac{\theta(s)}{e_a(s)} = \frac{K}{s(\tau s + 1)} \quad (2.16)$$

Para a planta analisada neste trabalho, a variável de saída estudada é a velocidade angular ( $\omega(t)$ ), então para prosseguir é necessário derivar o deslocamento angular para assim obter  $W(t)$ . Com isso, tem-se:

$$\frac{W(s)}{e_a(s)} = s \frac{\theta(s)}{e_a(s)}, \quad (2.17)$$

Dessa forma, o modelo matemático do motor de corrente contínua encontrado foi o seguinte sistema de primeira ordem:

$$\frac{W(s)}{e_a(s)} = G(s) = \frac{K}{s\tau + 1}, \quad (2.18)$$

em que  $K$  é o ganho do motor CC e  $\tau$  é a constante de tempo.

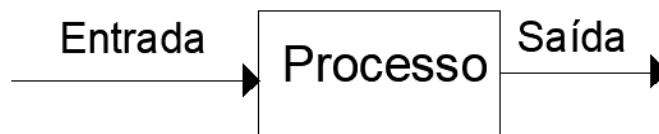
### 3 PROJETO DO CONTROLADOR

Nesse capítulo será apresentada, primeiramente, uma breve introdução aos sistemas de controle dinâmicos. Posteriormente serão mostrados os procedimentos desenvolvidos para o projeto do compensador PI da planta analisada no item 2.6.

#### 3.1 Sistemas de Controle

Sistemas de controle são representados por um conjunto de regras comumente descritas por equações que gerenciam o comportamento de sistemas físicos. Visam garantir uma resposta desejada diante de um desempenho desejado. Esse conceito pode ser simplificado a partir da análise da figura 12 a seguir:

Figura 12 – Representação de um sistema de controle



Fonte: Próprio autor.

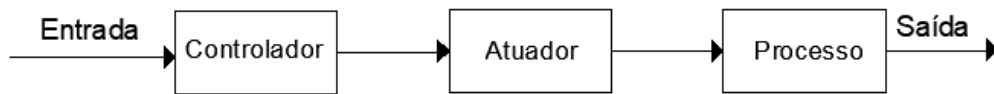
O sistema de controle é representado pelo conjunto Entrada+Processo+Saída, em que bloco “Processo” representa o sistema físico a ser controlado.

Os sistemas de controle dinâmicos podem ser definidos em sistemas de malha aberta e de malha fechada. Adiante, será feito um breve estudo sobre esses dois tipos de sistemas.

##### 3.1.1 Sistemas de Controle em Malha Aberta

Segundo Nise (2013), sistemas de controle em malha aberta são aqueles que não utilizam uma realimentação. Neles controla-se o processo diretamente por meio de atuadores, desta forma, o sinal de saída não desempenha qualquer ação de controle sobre o sinal de entrada. Com isso, na presença de quaisquer distúrbios, o controle em malha aberta não leva necessariamente à saída desejada. A figura abaixo representa o esquema do sistema de malha aberta:

Figura 13 – Representação de um sistema de controle em malha aberta



Fonte: Próprio autor.

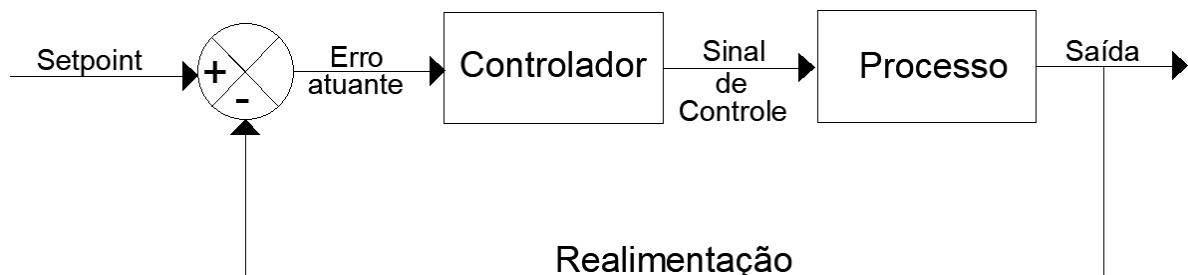
### 3.1.2 Sistemas de Controle em Malha Fechada

O sistema de controle em malha fechada resume-se em manter o sinal de saída em torno da resposta desejada, mesmo que haja algum distúrbio que possa vir a interferir no funcionamento do processo.

Para realizar o processo de compensação das perturbações externas, é introduzido ao sistema um sinal de realimentação. De posse desse sinal, é possível calcular o erro atuante, que é a diferença entre o valor de *setpoint* (sinal de referência) e o sinal de realimentação. A resultante desse cálculo é a responsável por realimentar o controlador e, por consequência, minimizar o erro e corrigir a resposta do processo para o valor projetado.

O diagrama básico do sistema em malha fechada é mostrado na figura a seguir:

Figura 14 – Representação de um sistema de controle em malha fechada



Fonte: Próprio autor.

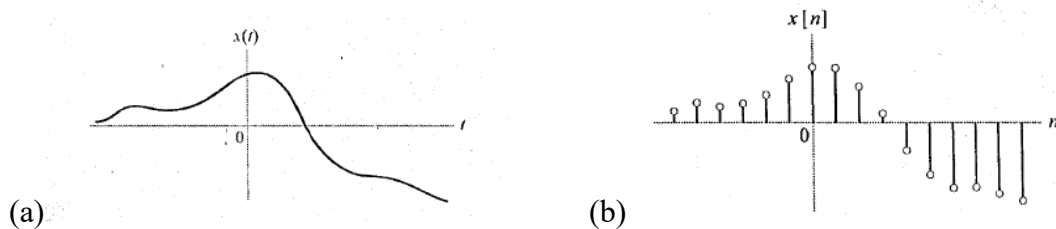
Foi utilizado um sistema de controle em malha fechada durante a realização deste trabalho na medida que necessitou-se introduzir um sinal de realimentação para estabilizar a velocidade do motor em um valor desejado.

### 3.1.3 Sistemas de Controle de Tempo Contínuo e Discreto

De acordo com Oppenheim (2010) os sistemas de controle contínuo são aqueles em que as variáveis de entrada de tempo contínuo são aplicados e resultam em sinais definidos para um intervalo de o tempo contínuo. Por outro lado, sistemas de controle de tempo discreto são aqueles em que as grandezas apresentam seus valores apenas em determinados instantes de tempo. Este sistema pode ser representado por sinais do tipo: amostrado, que possua amplitude continua em um intervalo de tempo, e digital, que possui amplitude quantizada.

Estes sistemas podem ser representados, respectivamente, por: sinais em tempo contínuo (figura 15- a), que são assim denominados porque existem em uma faixa contínua de valores, análogas ao comportamento observado fisicamente. Daí vem o nome sinal analógico. E sinais digitais (figura 15 - b), que por outro lado, existem em uma faixa discreta de valores igualmente espaçados, dando origem ao termo "período de amostragem".

Figura 15 – Representação gráfica de (a) sinais de tempo contínuo e (b) sinais de tempo discreto



Fonte: HAYKIN (2001, p. 35).

Para a planta desenvolvida neste trabalho, o compensador é implementado em computador, ou seja, os parâmetros de ajuste de controle podem ser modificados de forma simples alterando o programa, sendo assim necessário o projeto do controlador digital.

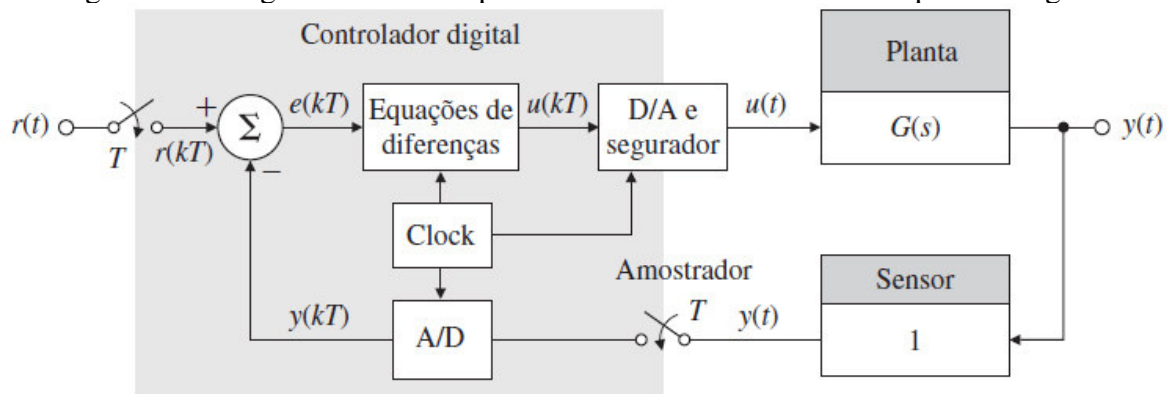
Para desenvolver o compensador digital, foi necessário adquirir a função de transferência de tempo discreto  $C(z)$ . Desta forma, foi utilizado o método de discretização para encontrar a função de transferência no plano  $z$ .



### 3.1.4 Sistema de Controle Digital

O sistema de controle digital consiste na utilização de computadores para executar ações sobre um sistema físico. Para Franklin (2013) “um controlador digital fornece ao projetista muito mais flexibilidade para fazer alterações na lei de controle após o projeto, pois a fórmula para o cálculo do sinal de controle é um programa, em vez de um circuito analógico.” Isso resulta na automatização de processos e conseqüentemente na redução de tempo empregado na realização do controle da planta, pois os processos que antes eram realizados de forma manual, podem ser realizados apenas fazendo ajustes no programa. A figura 16 mostra o diagrama de bloco para um sistema de controle com digital.

Figura 16 – Diagrama de blocos para sistema de controle com computador digital



Fonte: Franklin (2013, p. 501).

Para desenvolver o compensador digital, foi necessário adquirir a função de transferência de tempo discreto  $C(z)$ . Desta forma, foi utilizado o método de discretização para encontrar a função de transferência no plano  $z$ .

### 3.1.5 Controle PID

De acordo com Dorf o controlador PID ou controlador de três termos possui a seguinte função de transferência:

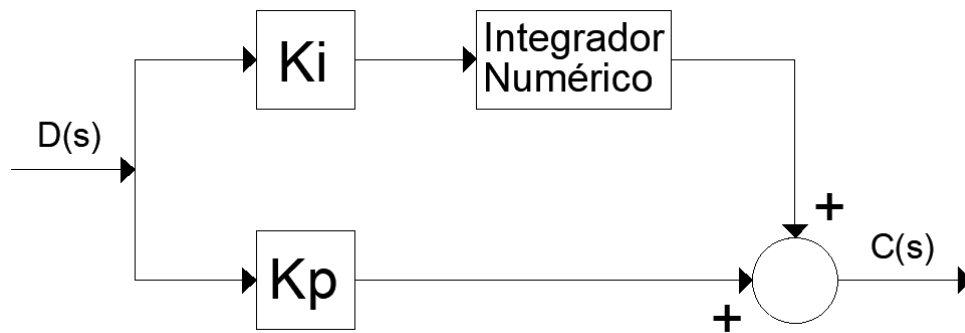
$$G(S) = K_p + \frac{K_i}{s} + K_d s \quad (3.1)$$

Sendo  $K_p$  o termo proporcional,  $K_i$  o termo integral e  $K_d$  o termo derivativo. Contudo para a realização deste trabalho o  $K_d$  é igual 0 devido as características de modelagem da planta estudada no capítulo 2. Com  $K_d=0$  tem-se o controlador proporcional e integral com a seguinte função de transferência:

$$G(s) = K_p + \frac{K_i}{s} \quad (3.2)$$

A figura 17 mostra o diagrama de blocos referente a função de transferência desenvolvida em (3.2).

Figura 17 – Diagrama de blocos controlador PI



Fonte: Próprio autor.

O controlador PI (proporcional-integral) é uma técnica de controle de processos que atua através de ações proporcionais (P) e integrais (I), permitindo realizar ajustes finos no ganho (K) e no tempo (t) em sistemas de malha fechada. A ação proporcional é responsável por minimizar o erro, assim, seu ganho é inversamente proporcional ao erro ( $K_p * e(t)$ ), enquanto a ação integral é encarregada de zerar o erro, pois seu funcionamento é baseado na proporcionalidade entre a velocidade e o erro, sendo assim seu ganho igual a ( $K_i / e(t)$ ).

## 3.2 Sintonização do Compensador

Para o desenvolvimento do controlador PI foi necessário coletar dados da planta estudada, para assim realizar algumas simulações computacionais com o intuito de obter as funções de transferência tanto para o sistema de controle em tempo contínuo, como para o sistema de controle em tempo discreto.

### 3.2.1 Aquisição de dados

Como citado anteriormente à aquisição de dados é realizada a partir do sensor de velocidade *encoder* e enviada ao *software* do Arduino. Essas informações coletadas possibilitaram a análise das respostas degrau da planta. As primeiras informações coletadas foram o tempo em que motor levou para estabilizar sua a velocidade, em repetições por minuto

(rpm), referente a esse período de tempo. Na tabela 5 é disponibilizada uma amostra dos dados coletados de velocidade do motor em rpm, para um intervalo de cinco segundos obtidos a partir do monitor serial do Arduíno.

Tabela 5 – Dados de tempo e velocidade

TEMPO (s)	VELOCIDADE EM RPM
0	0
0,1	330
0,2	480
0,3	540
0,5	570
0,8	450
0,9	450
1	480
1,1	480
1,2	510
1,5	600
1,6	450
1,7	600
2	480
2,1	480
2,2	510
2,5	450
2,6	600
2,7	510
3	570
3,1	540
3,2	450
3,5	540
3,6	510
3,7	570
4	510
4,1	600
4,2	510
4,5	510
4,6	600
4,7	570
5	480

Fonte: Próprio autor.

Nota-se que os valores de velocidade obtidos estão acima dos valores da especificação técnica do motor. Isso pode ser ocasionado por imprecisão de medição do sensor de velocidade ou por a baixa resolução do disco *encoder*, que possui apenas 20 pulsos por revolução. Contudo, esse erro de medição não interferiu, de forma significativa, no desenvolvimento do

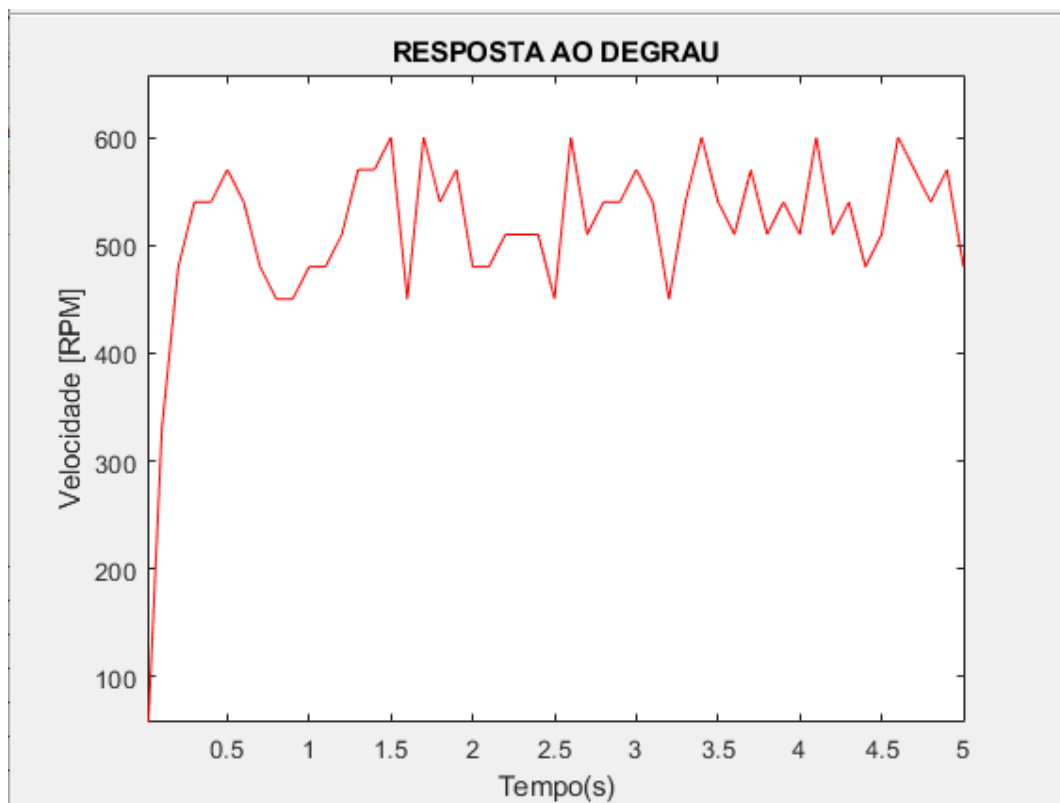
compensador, pois mesmo com valores alterados esses dados continuam descrevendo o comportamento da planta.

### 3.2.2 Obtenção da Função de Transferência

De acordo com Ogata (2011) “na teoria do controle, as funções de transferência são comumente utilizadas para caracterizar as relações de entrada e de saída de componentes ou de sistemas que podem ser descritos por equações diferenciais lineares invariantes no tempo.” Para adquirir a função de transferência, é preciso analisar a resposta degrau da planta. Desta análise, estimam-se os valores de ganho estático ( $K$ ), a constante de tempo ( $\tau$ ) e os dados necessários para chegar aos valores da constante proporcional ( $K_p$ ) e a constante integral ( $K_i$ ).

A partir dos dados importados do Arduíno e com o auxílio do *software* Matlab foi possível plotar o gráfico da resposta ao degrau do motor CC. A figura 18, mostra o resultado gráfico da resposta ao degrau da planta para um sinal de controle de 250 PWM, durante um tempo de observação de cinco segundos e com intervalo de medição de 0.1 segundos.

Figura 18 – Resposta degrau.



Fonte: Próprio autor.

Observando a figura 18 nota-se que a resposta é bastante ruidosa e que os valores de velocidade em rpm estão acima do esperado para o motor escolhido para compor a bancada. Isso se deve, respectivamente, ao número reduzido de perfurações do disco *encoder* e aos erros de leitura do sensor utilizado. A seguir será realizada uma análise matemática da função de transferência de primeira ordem.

Sabe-se que função de primeira ordem que descreve a sistema estudado é dada por:

$$G(s) = \frac{K}{\tau s + 1} \quad (3.3)$$

Sendo G(s) igual a saída Y(s) dividido pela entrada X(s) e que a transformada de Laplace da função degrau unitário é de 1/s, tem-se a seguintes equações:

$$\frac{Y(s)}{X(s)} = \frac{K}{\tau s + 1} \quad (3.4)$$

$$Y(s) = \frac{K}{\tau s + 1} \frac{1}{s} \quad (3.5)$$

Expandindo a equação (3.5) em frações parciais:

$$Y(s) = \frac{1}{s} - \frac{1}{s + (\frac{1}{\tau})} \quad (3.6)$$

Obtendo a transformada inversa de Laplace da equação (3.6), será obtido:

$$Y(t) = 1 - e^{-\frac{t}{\tau}}, \text{ para } t \geq 0 \quad (3.7)$$

A partir da equação (3.7) pode-se concluir que:

- Para t=0, Y(t) será zero;
- Para um t=∞, Y(t) será unitário;
- Quando t=τ, a resposta do processo será, aproximadamente, 63% da variação total.
- O ganho K será igual ao valor em estado de equilíbrio.

Diante das informações obtidas acima pode-se realizar efetivamente os cálculos de K e τ.

### 1) Cálculo de K:

Para obter o valor de K aplica-se a seguinte equação:

Sabendo que:

Vel\_rpm = Velocidade de regime permanente em rpm;

Cont\_pwm = Sinal de controle em pwm escolhida para implementação.

$$K = \frac{Cont\_rpm}{Vel\_pwm} \quad (3.8)$$

Como houve muito ruído na aferição da velocidade da resposta degrau, optou-se por realizar uma média móvel dos dados disponíveis na tabela 5, para alcançar o valor de final em estado de equilíbrio.

A média móvel resultante foi de 540 rpm, assim o ganho estático será de:

$$K = \frac{540}{250} = 2,16 \quad (3.9)$$

## 2) Cálculo de $\tau$ :

Visto que a velocidade em rpm referente a constante de tempo é aproximadamente 63% da variação total em estado de equilíbrio, então a resposta ao processo será de:

$$Vel_{\tau} = \frac{63}{100} 540 = 340 \text{ rpm} \quad (3.10)$$

Em seguida, analisou-se o gráfico da figura 18 para localizar o tempo correspondente ao valor de rpm calculado em (3.9) e assim obter  $\tau$ .

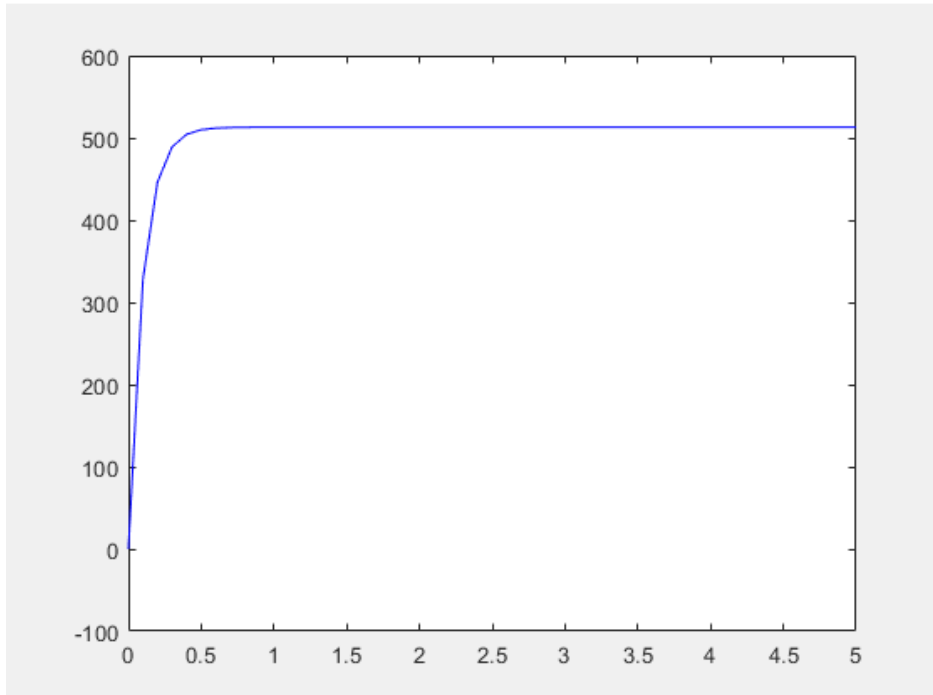
$$\tau = 0,098 \text{ s} \quad (3.11)$$

De posse dos valores do ganho estático e da constante de tempo calculados obtém-se a função de transferência de um sistema de tempo contínuo, substituindo os valores de  $K$  e  $\tau$  em 3.1.

$$G(s) = \frac{2,16}{0,098s + 1} \quad (3.12)$$

A figura 19 mostra o gráfico correspondente ao modelo matemático da função de transferência ( $G(s)$ ) conseguida em (3.12).

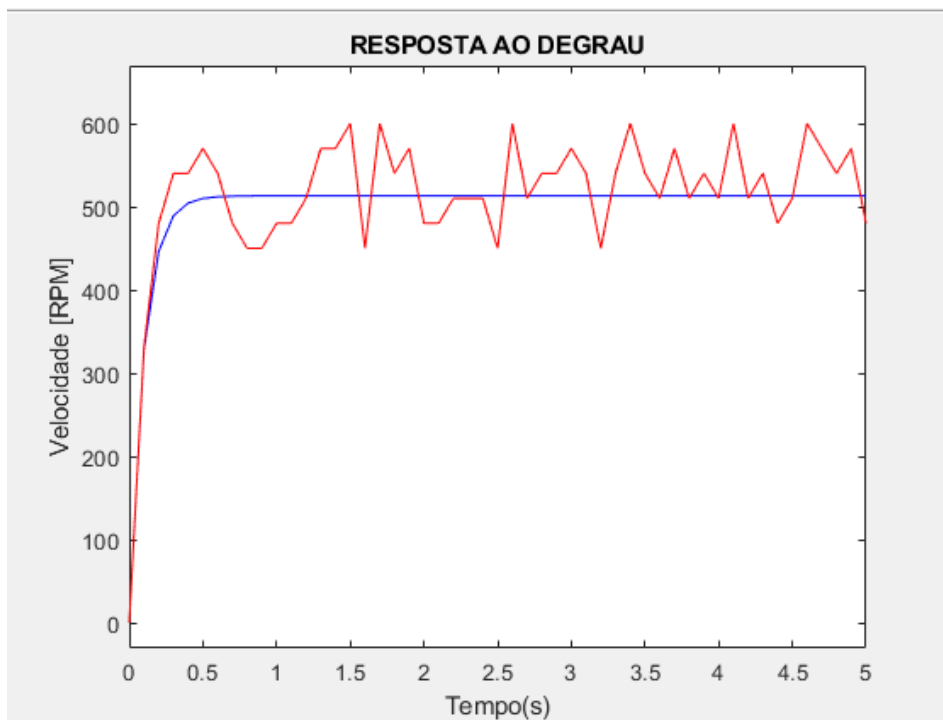
Figura 19 – Gráfico do modelo matemático obtido



Fonte: Próprio autor.

A seguir, a partir da figura 20, pode-se observar o gráfico referente ao modelo matemático, curva em azul, associada à resposta degrau do motor utilizado, curva em vermelho.

Figura 20 – Modelo matemático associado a entrada degrau



Fonte: Próprio autor.

Realizando uma análise entre as duas curvas apresentada na figura 20 é possível afirmar que o modelo matemático desenvolvido atende as características da entrada degrau obtida do sistema físico.

### 3.2.3 Projeto do Controlador PI Digital

O projeto digital é realizado a partir das seguintes etapas:

1. Discretização da função de transferência (3.12) com período de amostragem de  $T_s=0.0098$  segundos;
2. Projeto do compensador PI na forma  $C(z) = A * \frac{(z-c)}{(z-1)}$ , sendo  $A$  o ganho do controlador e  $0 < c < 1$  o zero do controlador.
3. Projeto do controlador diretamente no plano- $z$ , através da técnica de posicionamento de polos. Para isso se faz uso de ferramentas computacionais disponíveis em MATLAB/ Octave como rltool ou sisotool.

### Discretização da Função de Transferência

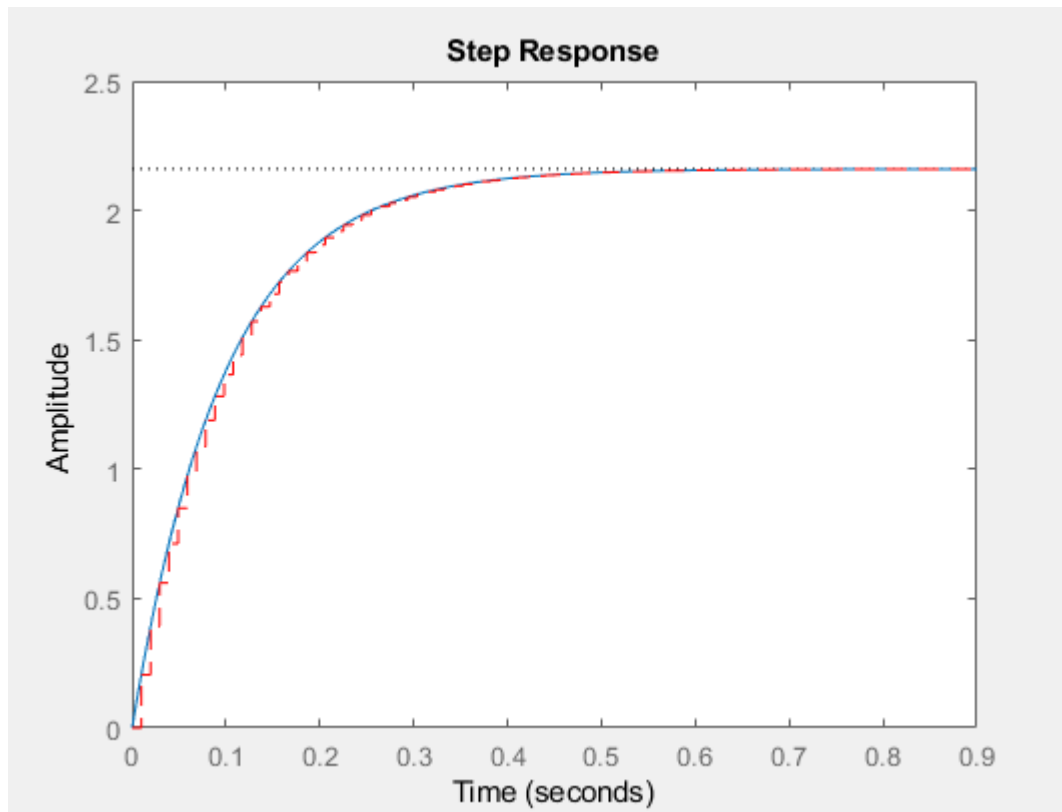
Para realizar o processo de discretização da função da transferência foi necessário realizar algumas análises no Matlab. Foi utilizada a função interna do Matlab: *Continuous - to - Discrete* (c2d), responsável por converter a função de tempo contínuo analisada no item (3.2.2) em uma função de transferência de tempo discretos. Como resultado deste procedimento obteve-se a seguinte função de transferência de tempo discreto:

$$Gsd(z) = \frac{1,365}{(z - 0.03679)} \quad (3.13)$$

Para realizar uma análise do comportamento da resposta degrau da função discretizada foi plotada o gráfico da figura 21. Nota-se que, a resposta a entrada degrau da função de tempo contínuo discretizada descreve de forma eficiente o comportamento da função de transferência de tempo contínuo, confirmando as simulações realizadas anteriormente.



Figura 21 – Informações da resposta degrau



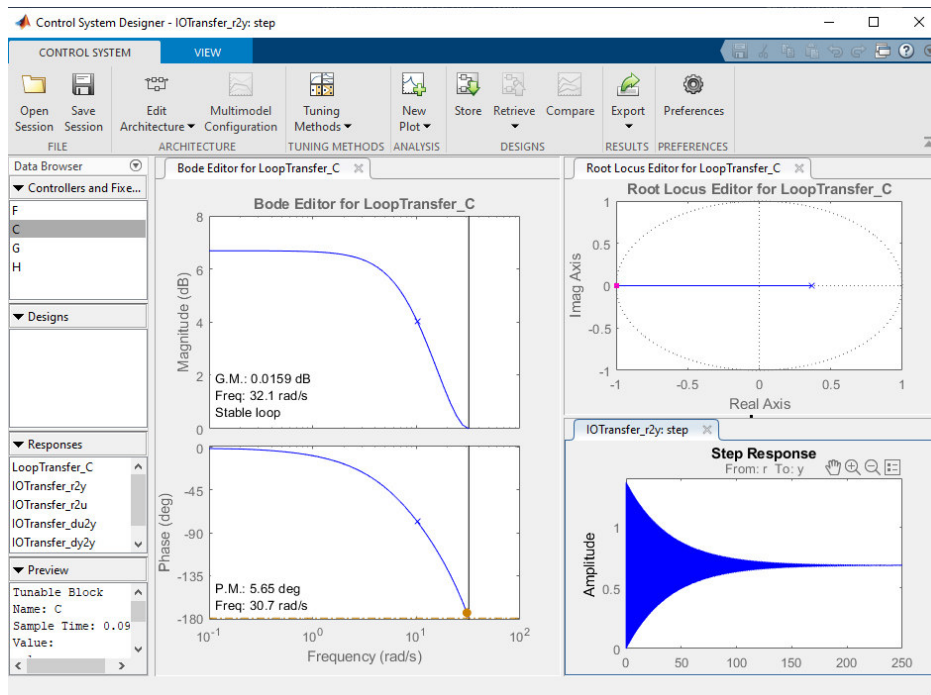
Fonte: Próprio autor.

### ***Obtenção do Controlador $C(z)$***

Com finalidade de obter o compensador de tempo de discreto foi utilizado a função interna do MATLAB *sisotool*.

O *sisotool* é uma função utilizada para inicializar a ferramenta de projeto de controladores que possuem um único sinal de entrada e um único sinal de saída - *SISO Design Tool (Design single-input, single-output)*. Ao executar o programa a função *sisotool* (Gsd) é inicializada a interface gráfica do usuário, em inglês: *graphical-user interface* (GUI), ambiente que permite a interação entre o usuário do computador e o programa de forma rápida e fácil. A figura 22 mostra o *layout* do GUI executado e os elementos gráficos que podem ser usados para projetar o compensador.

Figura 22 – Layout do GUI SISO Design Tool

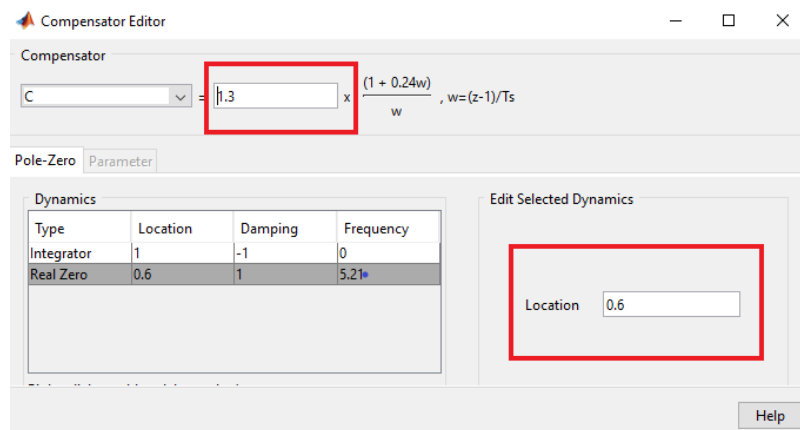


Fonte: Próprio autor.

Para a realização deste de trabalho foi usado o método do lugar geométrico das raízes com a finalidade de definir adequadamente o controlador. Com isso foi utilizado apenas o *Root Locus Editor* (Editor do Lugar das raízes) do GUI. Para obter a estrutura mais adequada do compensador foi feito algumas alterações no gráfico do lugar das raízes, sendo adicionado: um polo na origem, chamado de integrador, e um zero real. Além disso, foram realizados ajustes finos no valor do ganho (K) e na localização do zero por meio do *Compensator Editor* (editor compensador).

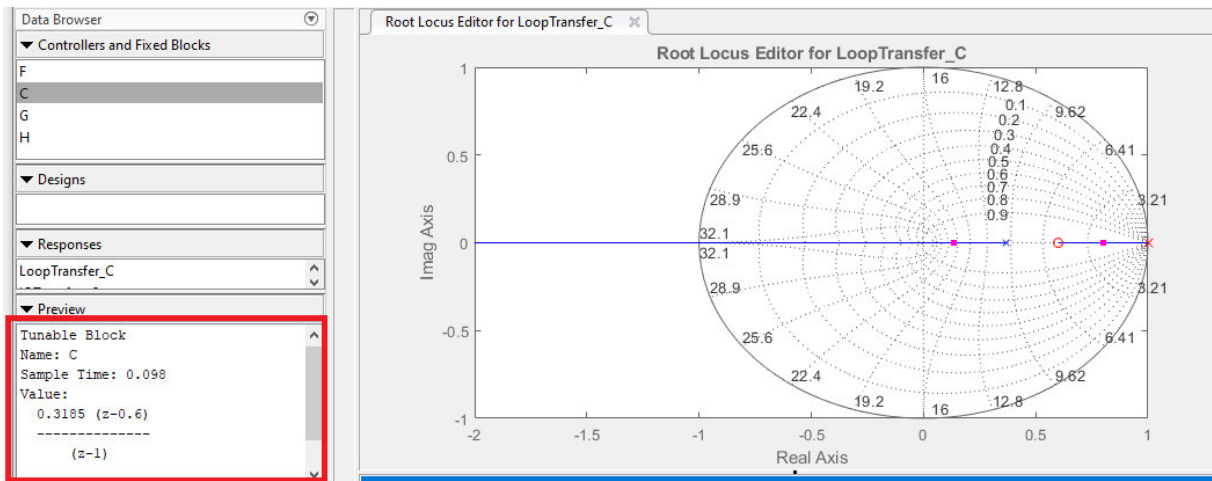
A figura 23 mostra a interface do *compensator editor* e o valor ajustado de K com a nova posição do zero real, enquanto a figura 24 exhibe o gráfico do *Root Locus Editor* e do controlador dimensionado, após finalizado todos os ajustes.

Figura 23 – Layout do compensador editor



Fonte: Próprio autor.

Figura 24 – Gráfico Root locus e compensador projetado



Fonte: Próprio autor.

Deste modo, foi determinado que  $C(z)$  é:

$$C(z) = \frac{0.3185 (z - 0.6)}{z - 1} \quad (3.14)$$

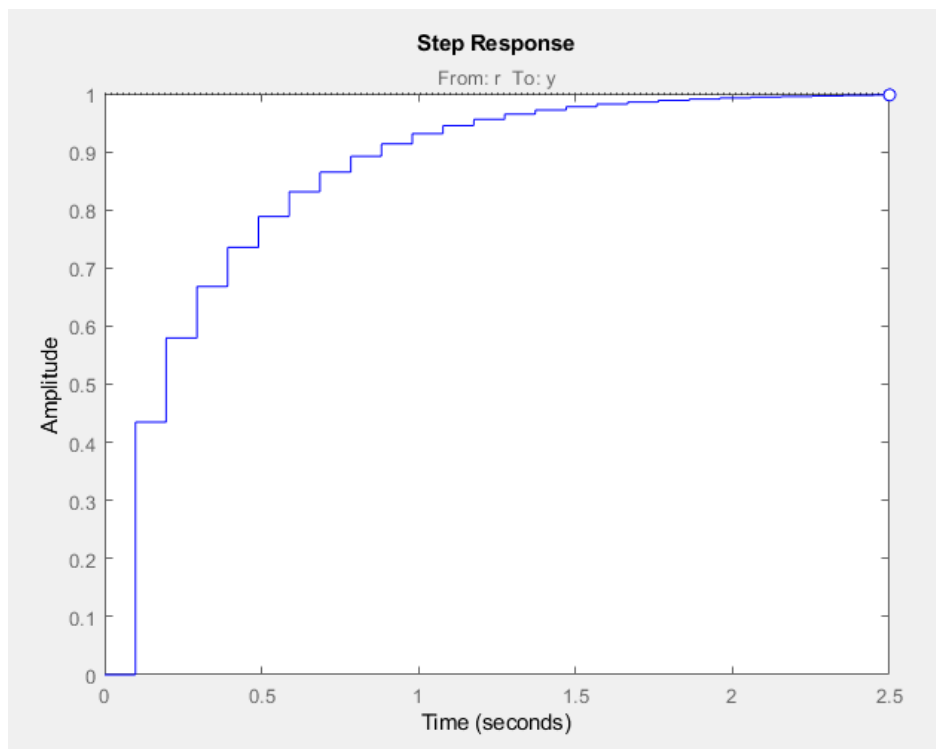
Para finalizar o processo de simulações computacionais foi realizada uma análise do gráfico da resposta degrau do compensador projetado. Na figura 25 é mostrado a onda de resposta ao degrau com seus pontos característicos destacados.

Nota-se que:

1. O tempo gasto para o sinal acomodar - tempo de acomodação (ou *settling time*) é de 1.52 s;
2. Que a diferença entre o valor máximo atingido e o valor final em porcentagem, chamado de sobressinal (ou *overshoot*) é de 0%

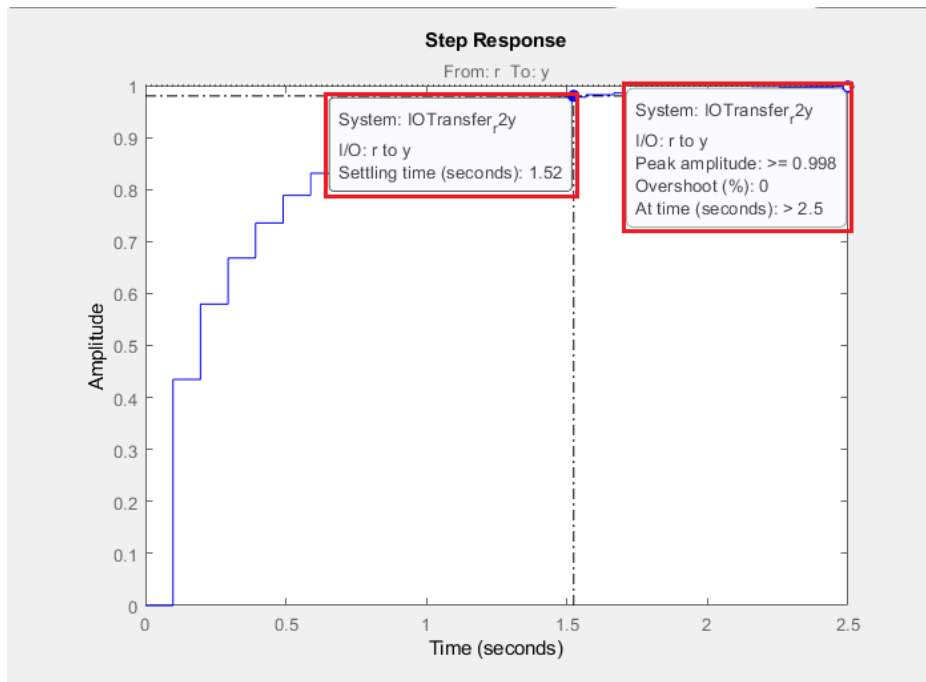
Com isso, pode-se inferir que as especificações para a resposta transitória do controlador projetado são satisfatórias e conseguem atender ao modelo de processo implementado neste trabalho.

Figura 25 – Resposta degrau do controlador C



Fonte: Próprio autor.

Figura 26 – Características da resposta degrau do controlador C



Fonte: Próprio autor.

### *Aproximação de Tustin*

Segundo Phillips (1995) os valores de  $K_i$  e  $K_p$  podem ser encontrados utilizando o método de integração numérica de Tustin, que consiste no mapeamento dos polos e zeros de uma função de tempo contínuo. De acordo com a aproximação de Tustin, o compensador digital pode ser obtido a partir das seguintes equações:

$$C(z) = Kp + \frac{Ki}{w} \quad (3.15)$$

$$C(z) = C(s) \Big|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} \quad (3.16)$$

Sabendo que  $w$ , de (3.15), é o integrador numérico, então o controlador  $C(z)$  é obtido substituído  $s$  em (3.15), conforme é mostrado a seguir:

$$C(z) = Kp + \frac{Ki}{\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} = Kp + Ki \frac{T}{2} \left(\frac{Z+1}{Z-1}\right) \quad (3.17)$$

### ***Cálculo de $K_i$ e $K_p$***

Para obter as constantes proporcional e integral foi desenvolvida as seguintes etapas de cálculo:

Primeiramente, a função (3.14) foi simplificada:

$$C(z) = 0,3185 \left( \frac{Z}{Z-1} \right) - \frac{0,3185 * 0,6}{Z-1} \quad (3.18)$$

$$C(z) = \frac{0,3185Z - 0,1911}{Z-1} \quad (3.19)$$

Em seguida foi analisada a função (3.14), na qual foi substituído  $\frac{T K_i}{2}$  por  $K_i'$  para facilitar os as manipulações algébricas:

$$C(Z) = K_p + \frac{K_i' Z + K_i'}{Z-1} = \frac{K_p Z - K_p + K_i' Z + K_i'}{Z-1} \quad (3.20)$$

Colocando Z em evidencia é encontrada a seguinte equação:

$$C(Z) = \frac{(K_i' + K_p)Z - (K_p - K_i')}{Z-1} \quad (3.21)$$

Substituindo (3.17) em (3.21)

$$\frac{0,3185Z - 0,1911}{Z-1} = \frac{(K_i' + K_p)Z + (K_p - K_i')}{Z-1} \quad (3.22)$$

A partir de (3.22) foi extraído o sistema de equações a seguir:

$$K_p + K_i' = 0,3185 \quad (3.23)$$

$$K_p - K_i' = 0,1911 \quad (3.24)$$

Para encontrar o valor de  $K_p$  soma-se (3.23) com (3.24)

$$K_p = \frac{0,5096}{2} = 0,2548 \quad (3.25)$$

Substituindo  $K_p$  em (3.23) tem-se o valor de  $K_i'$

$$K_i' = 0,3185 - 0,2548 = 0,0637 \quad (3.26)$$

Sendo  $K_i'$  igual a  $\frac{T K_i}{2}$  e T igual  $\tau$ , então  $K_i$  será:

$$K_i = 2 \frac{K_i'}{T} = 1,30 \quad (3.27)$$

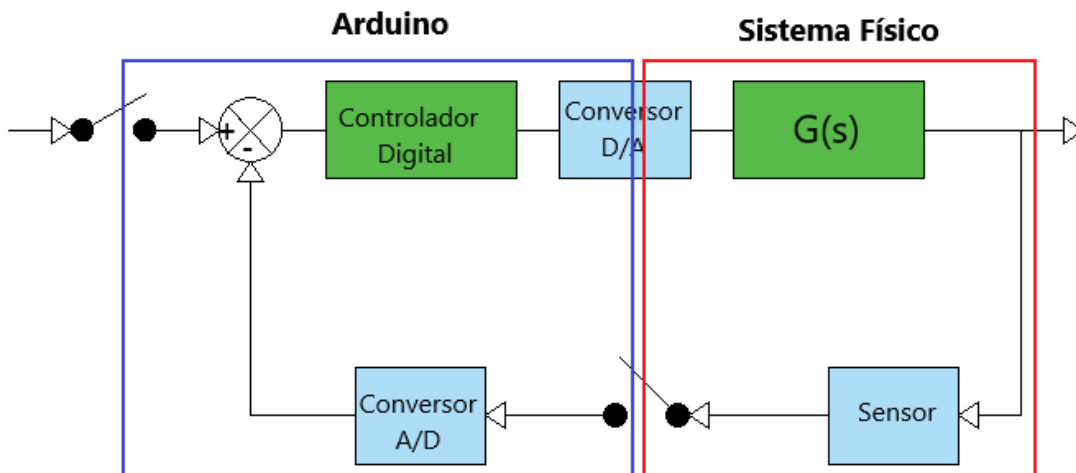
Assim, foi encontrado que  $K_i = 1,30$  e  $K_p = 0,2548$ .

## 4 RESULTADOS

Neste capítulo são apresentados os resultados simulados e reais do projeto de controle de velocidade do motor de corrente contínua. Observa-se que para obtenção dos resultados, primeiramente, é preciso entender o funcionamento do sistema de controle digital utilizado.

A partir da análise da figura 27, é possível inferir que o sistema físico é representado por uma planta analógica (motor CC) e um sensor (*encoder* + sensor de velocidade) responsável por coletar os dados analógicos formando assim o sistema físico, enquanto o Arduino é encarregado por transformar os dados coletados em sinais digitais (A/D) para realizar o cálculo do erro atuante e com isso efetuar as devidas correções. Após executar o controle digital o Arduino faz novamente uma conversão, agora o sinal digital é convertido em sinal analógico e enviando para o processo, fechando a malha.

Figura 27 – Diagrama de blocos controlador digital



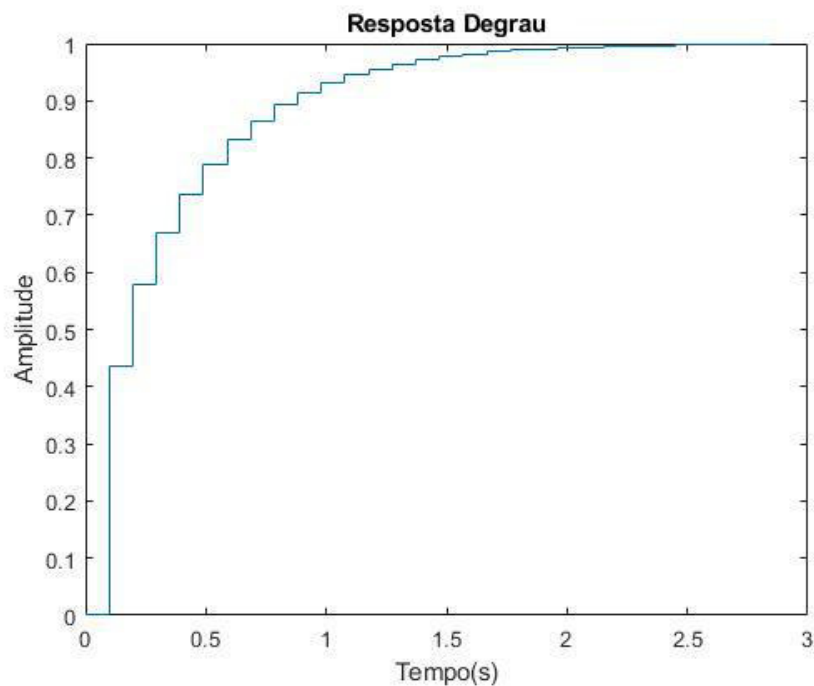
Fonte: Próprio autor.

### 4.1 Simulações

Para o desenvolvimento do controlador PI foi necessário coletar dados da planta estudada, para assim realizar algumas simulações computacionais com o intuito de obter as funções de transferência tanto para o sistema de controle em tempo contínuo, como para o sistema de controle em tempo discreto.

Primeiramente a função de transferência de tempo contínuo foi discretizada com auxílio da função interna do MATLAB – `c2d`; em seguida é obtido a função de transferência do controlador utilizando a função *transfer function model*<sup>6</sup>(`tf`); e por último a resposta ao degrau é conseguida fechando-se a malha do sistema e implementando uma realimentação unitária, por meio do comando *feedback*<sup>7</sup>. A figura 28 exibe a resposta degrau obtida.

Figura 28 –Resposta degrau do controlador



Fonte: Próprio autor.

Após adquirir a figura 28 foi obtido, a partir do comando `stepinfo`<sup>8</sup>, os parâmetros de desempenho do controlador  $C(z)$  em malha fechada com realimentação unitária mostrados na figura 29.

Analisando a figura 29, observa-se que o tempo de estabilização apresenta uma pequena diferença de valor para aquele encontrado na figura 26 (*settling time* = 1,52 s), isso era esperado pois no procedimento de desenvolvimento do controlador PI, a partir da ferramenta interna do MATLAB *sisotool*, foi modificada a localização do zero real resultando em uma diminuição do tempo de estabilização.

<sup>6</sup> Função Modelo de Função de Transferência

<sup>7</sup> Função de Realimentação.

<sup>8</sup> Função que mostra as informações as características do gráfico da resposta degrau.



Figura 29 – Informações da resposta degrau

```

RiseTime: 0.7840
SettlingTime: 1.5680
SettlingMin: 0.9138
SettlingMax: 1.0000
Overshoot: 0
Undershoot: 0
Peak: 1.0000
PeakTime: 6.7620

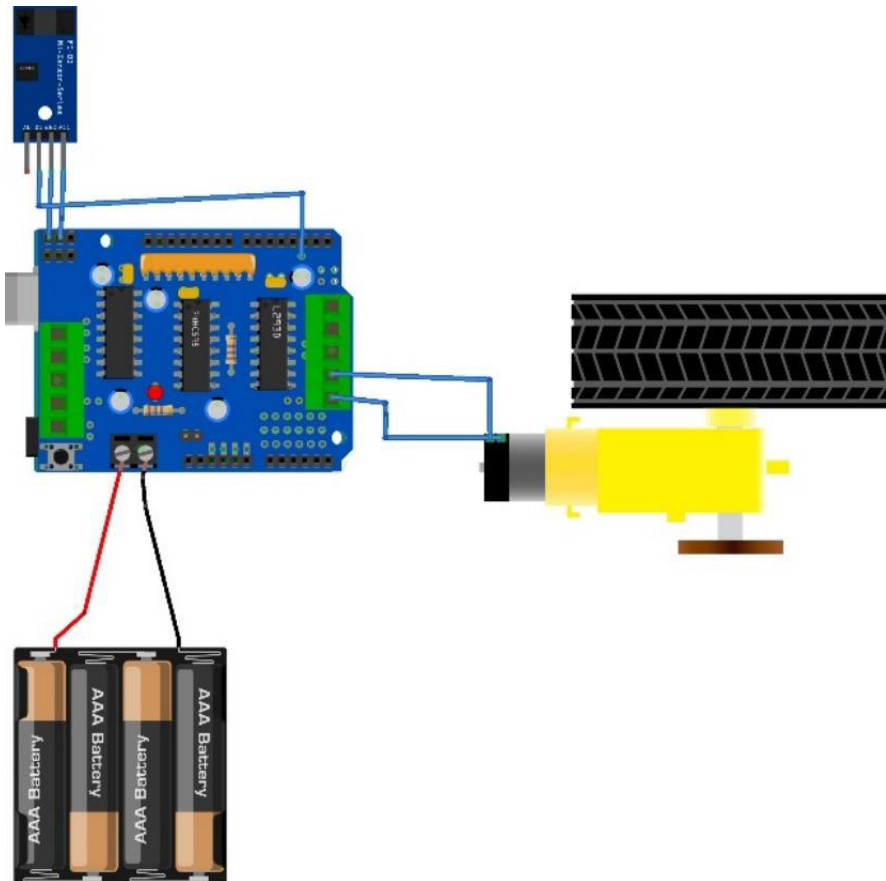
```

Fonte: Próprio autor.

## 4.2 Resultados Práticos

Com o controlador devidamente projetado, realizou-se a montagem do sistema de controle. A figura 30 mostra o esquemático do circuito implementado, nota-se que o Motor *Shield* L293D é conectado diretamente no Arduino, estando sua placa sobreposta à placa do Arduino.

Figura 30 – Esquemático do circuito implementado

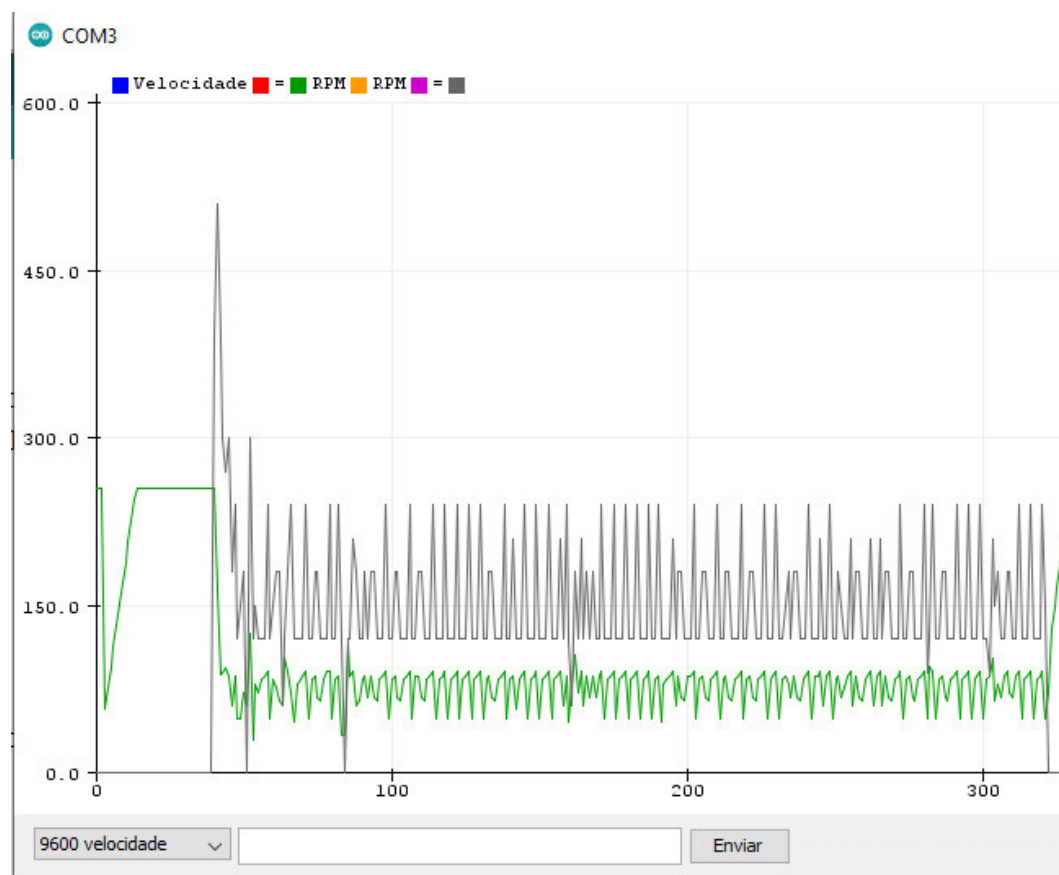


Fonte: Próprio autor.

A partir do *plotter serial*, recurso disponível no Arduino que permite mostrar em forma de gráficos as informações obtidas na programação, foi realizado dois testes de verificação para aferir se o controlador PI projetado estava funcionando corretamente.

**Teste 1:** Foi analisado o comportamento do compensador para um único valor de referência. A figura 31, mostra a resposta do controlador para um *setpoint* de 150 rpm (curva azul) e o sinal de controle em PWM (curva verde), tendo a velocidade em rpm representada pelo eixo y e o tempo em milissegundos representado pelo eixo x.

Figura 31 – Resposta do controlador PI para 150 rpm



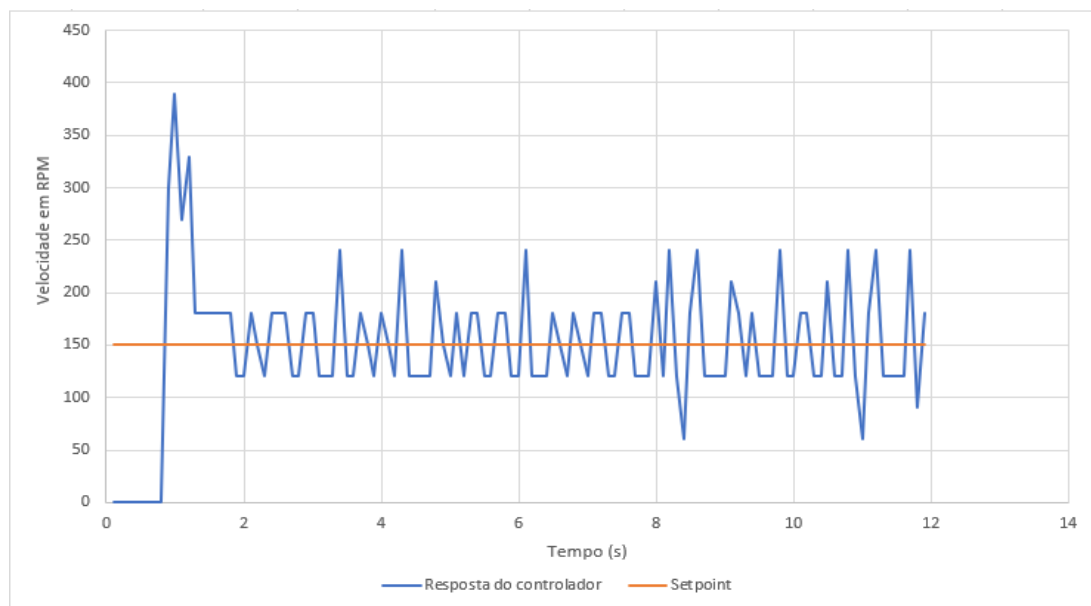
Fonte: Próprio autor.

Nota-se que, o conversor D/A fica ajustado com sinal de controle, representado pela curva verde, no valor máximo (PWM = 255) antes da rotina de controle ter sido iniciada. Então, após 40 milissegundos, o sensor de velocidade é posicionado junto ao *encoder* e neste momento o motor inicializa a operação em valor máximo ocasionando um pico de velocidade no sinal de medido. Depois, a cada iteração seguinte, o controlador atua.

Observa-se que, apesar do sinal coletado ser bastante ruidoso, devido as imprecisões na medição, o sinal de resposta do controlador estava oscilando em torno de 150 rpm, mostrando que o controlador estava funcionando corretamente.

Na figura 32 é apresentado o gráfico, a partir de dados coletados pela comunicação serial do Arduino, no qual é possível visualizar de forma mais clara, o comportamento de controle do compensador projetado.

Figura 32 – Resposta do controlador PI

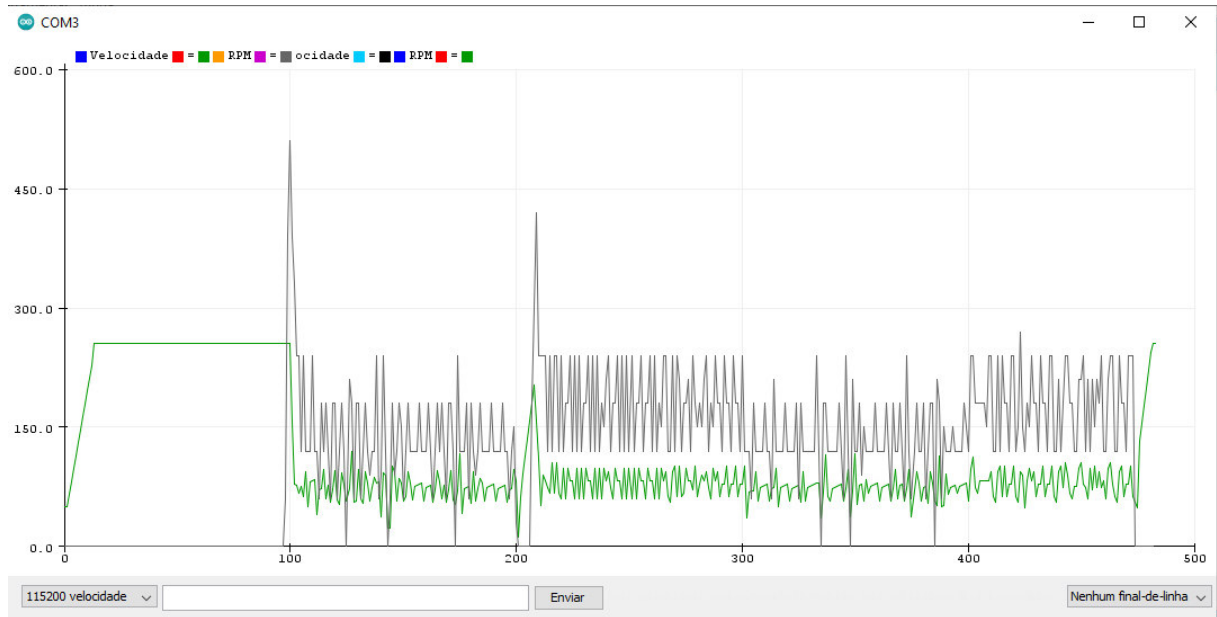


Fonte: Próprio autor.

**Teste 2:** Para confirmar o funcionamento do controlador PI projetado, foi inserido um segundo valor de referência e observado a comportamento do controlador diante de uma variação de entrada.

A figura 33 apresenta o comportamento do compensador PI, para dois valores de referência: *setpoint 1* de 130 rpm e *setpoint 2* de 180 rpm.

Figura 33 – Resposta do controlador PI para dois valores de referência



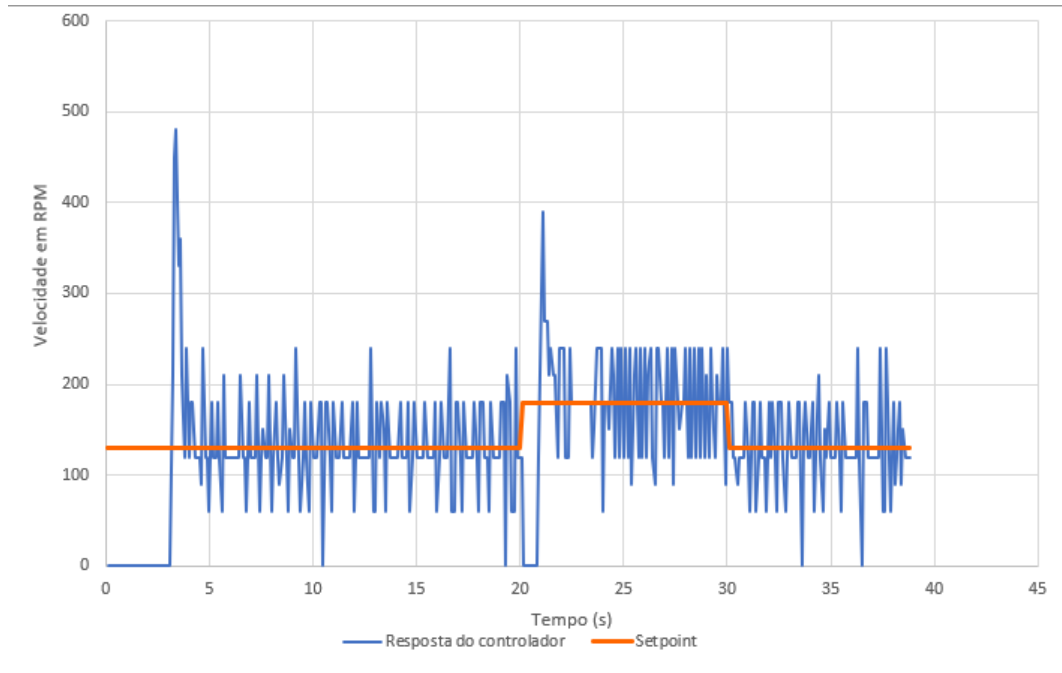
Fonte: Próprio autor.

Nota-se que o sinal continua bastante ruidoso, confirmado a análise realizada no primeiro teste.

Analisando a figura 33 é possível inferir que, para os primeiros 10 segundos, o motor estava funcionando em sua velocidade máxima, em seguida o sensor é introduzido no circuito fazendo assim o compensador inicializar o processo de correção do erro. Observa-se que o controle se iniciou com o valor de *setpoint* de 130 rpm, permanecendo neste valor durante 10 segundos. Nos segundos seguintes o valor de referência é modificado, passando a ser de 180 rpm e permanecendo neste valor por 10 segundos. Em seguida o valor de *setpoint* retorna para 130 rpm. O ciclo é fechado quando o valor de referência retorna para 180 rpm.

Para obter uma melhor visualização do processo de controle foi coletado os dados de velocidade em rpm para os primeiros 40 segundos e implementado graficamente. A figura 34 mostra o gráfico resultante dessa análise.

Figura 34 – Comportamento do controlador PI para dois valores de referência



Fonte: Próprio autor.

Observando a figura 34, nota-se que o controlador conseguiu acompanhar as mudanças dos valores de referência realizando o controle corretamente e corrigindo de forma adequada o erro nas diferentes situações apresentadas.

## 5 CONCLUSÕES

Este trabalho possibilitou entender e desenvolver conhecimentos sobre sistemas de controle discreto, discretização de sistemas em tempo contínuo, conceitos de automação e programação em Arduino para assim viabilizar o desenvolvimento do controlador PI digital estudado. O sistema didático desenvolvido permite compreender, de forma prática, o funcionamento de um sistema de controle discreto, além de possibilitar a visualização tanto por interface gráfica, como em tempo real do funcionamento do compensador PI desenvolvido para a planta em análise.

Nota-se que, apesar das interferências causadas por ruídos na medição necessária para obtenção de dados e conseqüentemente para o projeto do compensador PI, o controle de velocidade do motor corrente contínua de baixa potência via modulação por largura de pulso funcionou de forma satisfatória, mostrando que o controlador foi projetado corretamente, cumprindo com o objetivo central deste estudo.

Como trabalho futuro, recomenda-se o estudo de um processo de filtragem *online* para melhoria do ruído de medição. Outra sugestão seria usar a planta do motor de corrente contínua usado neste projeto para descrever o funcionamento de um motor CC utilizado para esguicho de água e, assim, montar um pequeno sistema didático de controle de nível de líquido.

## REFERÊNCIAS

- ARDUINO CC. **PWM**, 2018. Disponível em:  
<https://www.arduino.cc/en/Tutorial/Foundations/PWM>. Acesso em: 28 jul. 2021.
- AUTOCORE ROBOTICA. **Disco Encoder ideal para sensor de velocidade**. 2021. Disponível em: <https://www.autocorerobotica.com.br/disco-encoder-ideal-para-sensor-de-velocidade>. Acesso em: 23 jul. 2021.
- BOJORGE, Ninoska. **Controle de processos**. Departamento de Engenharia Química e de Petróleo: UFF, 2016. Disponível em:  
<https://docplayer.com.br/15194714-Sumario-disciplina-teq102-controle-de-processos-prof-a-ninoska-bojorge-departamento-de-engenharia-quimica-e-de-petroleo-uff.html>. Acesso em: 30 jul. 2021.
- CASTRUCCI, Plínio de Lauro. **Controle automático**, Rio de Janeiro: LTC, 2011
- CHAPMAN, Stephen J. **Fundamentos de máquinas elétricas** [recurso eletrônico] / Stephen J. Chapman; tradução: Anatólio Laschuk. – 5ª Ed – Dados eletrônicos. – Porto Alegre: AMGH, 2013.
- Dorf, Richard C.; Bishop, Robert H. **Sistemas de Controle Modernos**, Rio de Janeiro: LTC, 2009. 8ª edição
- ELETROGATE. **Sensor de velocidade Encoder**. 2021. Disponível em:  
[https://www.eletrogate.com/sensor-de-velocidade-encoder?gclid=CjwKCAjwr56IBhAvEiwA1fuqGujdA4IRqh9Pj-gLohYyXGXWhFk4DbM2sfCdWAUxAekHtF11h0BuQxoC9J4QAvD\\_BwE](https://www.eletrogate.com/sensor-de-velocidade-encoder?gclid=CjwKCAjwr56IBhAvEiwA1fuqGujdA4IRqh9Pj-gLohYyXGXWhFk4DbM2sfCdWAUxAekHtF11h0BuQxoC9J4QAvD_BwE). Acesso em: 23 jul. 2021.
- EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. **Arduino EMAÇÃO** – 1ª. Ed – São Paulo: Novatec Editora Ltda, 2015.
- FÁBRICA DE BOLSO. **Motor Dc 3-6v com caixa de redução p/ robótica educativa**. 2021. Disponível em: [https://loja.fabricadebolso.com.br/motor-dc-3-6v-com-caixa-de-reducao-p-roboticaeducativa?utm\\_source=Site&utm\\_medium=GoogleMerchant&utm\\_campaign=GoogleMerchant&gclid=CjwKCAjw0qOIBhBhEiwAyyVcf1H8fyTXgXN50nG2t-uirn7uPnB9WkGaSdWnp584-c7lXeV\\_UbVJchoCQ\\_QQAvD\\_BwE](https://loja.fabricadebolso.com.br/motor-dc-3-6v-com-caixa-de-reducao-p-roboticaeducativa?utm_source=Site&utm_medium=GoogleMerchant&utm_campaign=GoogleMerchant&gclid=CjwKCAjw0qOIBhBhEiwAyyVcf1H8fyTXgXN50nG2t-uirn7uPnB9WkGaSdWnp584-c7lXeV_UbVJchoCQ_QQAvD_BwE). Acesso em: 08 de ago. 2021.
- FRANKLIN, Gene F. **Sistema de controle para engenharia** [recurso eletrônico] / Gene F. Franklin, J. David Powell, Abbas Emami - Naeini; tradução: Fernando de Oliveira Souza; revisão técnica: Antonio Pertence Júnior. – 6ª. Ed – Dados eletrônicos. – Porto Alegre: Bookman, 2013.
- HART, Daniel W. **Eletrônica de potência: análise e projetos de circuitos** / Daniel W.Hart; Tradução: Romeu Abdo; revisão técnica: Antonio Pertence Júnior. – Porto Alegre: AMGH, 2012.

NISE, Norman S. **Engenharia de sistemas de controle** / Norman S. Nise; tradução e revisão técnica Jackson Paul Matsuura. – 6ª – [Reimpr.]. – Rio de Janeiro: LTC, 2013.

OGATA, Katsuhiko. **Engenharia de controle moderno** / Katsuhiko Ogata; tradutora Heloísa Coimbra de Souza; revisor técnico Eduardo Aoun Tannuri. – 5ª – Ed – São Paulo: Pearson Prentice Hall, 2010.

OPPENHEIM, Alan V. **Sinais e sistemas** / Alan V. Oppenheim, Alan S. Hamid Nawab; tradução Daniel Vieira e Rogéria Bettoni; revisão técnica Marcio Elisencraft e Maria D.Miranda – São Paulo: Pearson Pretice Hall, 2010.

PHILLIPS, Charles L. **Digital control system analysis and design** / Charles L. Phillips, H. Troy Nagle. – 3ª Ed. – Englewood Cliffs, New Jersey: Prentice-Hall, Inc, 1995.

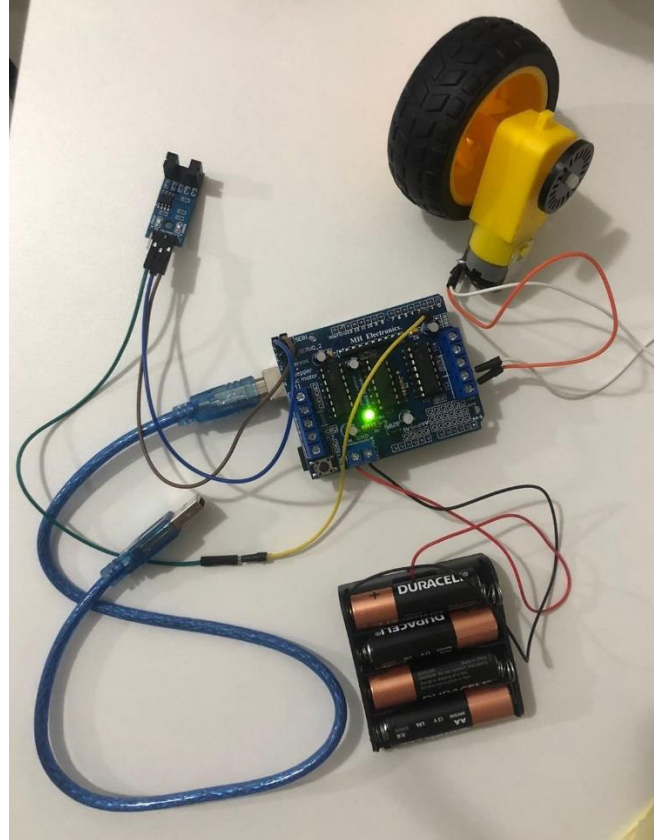
Robot Rover para Exploração do Ambiente Envolvente. Disponível em:  
[https://www.researchgate.net/publication/304657545\\_Robot\\_Rover\\_para\\_Exploracao\\_do\\_Ambiente\\_Envolvente](https://www.researchgate.net/publication/304657545_Robot_Rover_para_Exploracao_do_Ambiente_Envolvente). Acessa em 28 ago. 2021.

UMANS, Stephen D. **Máquinas elétricas de Fitzgerald e Kingsley** [recurso eletrônico] / Stephen D. Umans; tradução: Anatólio Laschuk. – 7ª – Ed – Dados eletrônicos. – Porto alegre: AMGH, 2014.



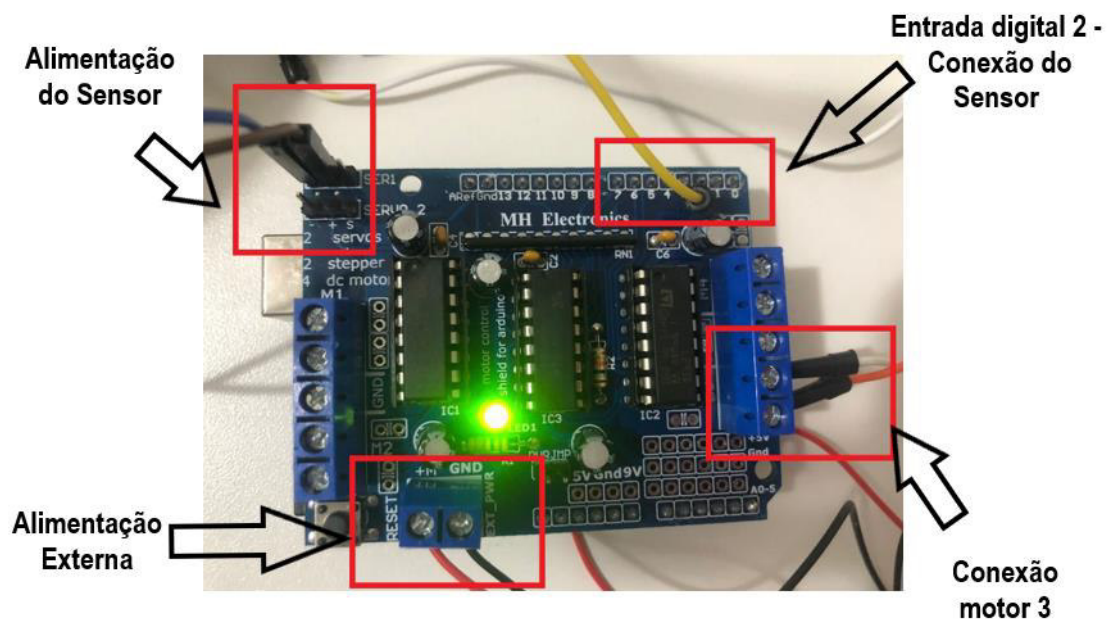
## ANEXO A – IMAGENS DA BANCADA

Figura 35 – Circuito utilizado



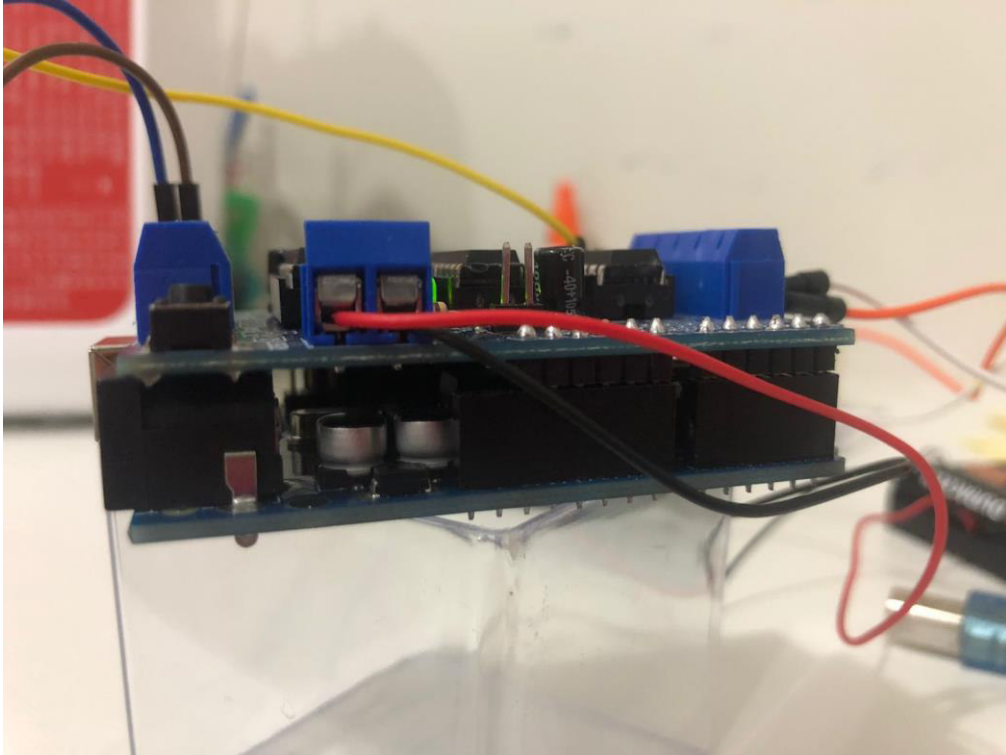
Fonte: Próprio autor.

Figura 36 – Conexões do Motor *Shield*



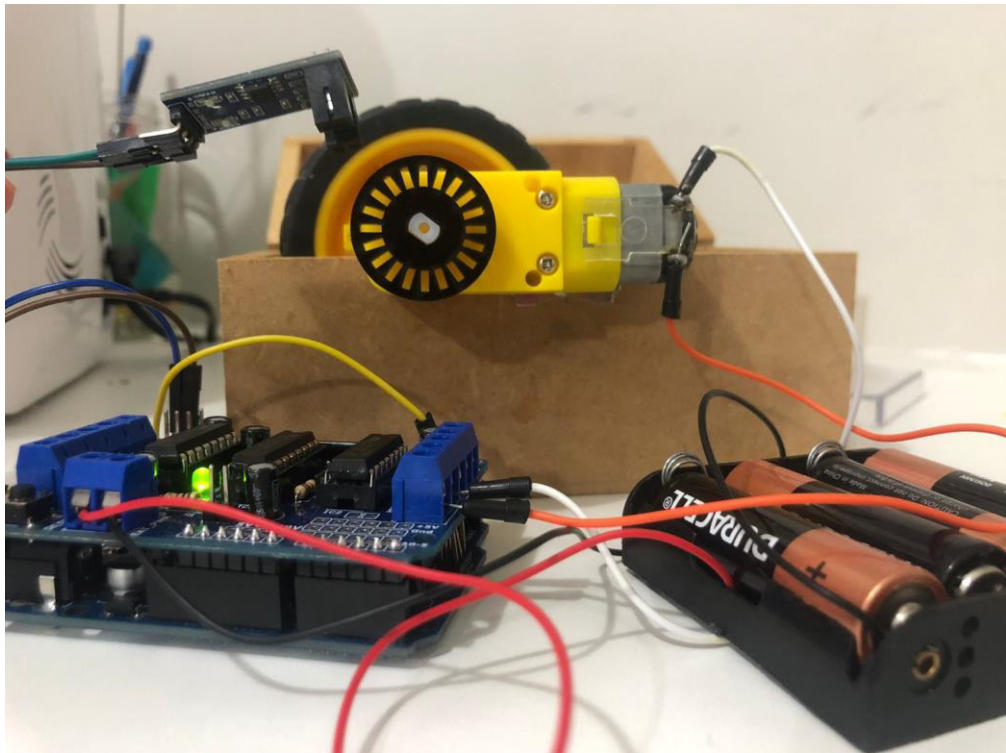
Fonte: Próprio autor.

Figura 37 – Conexões do Motor *Shield* no Arduino



Fonte: Próprio autor.

Figura 38 – Bancada



Fonte: Próprio autor.

## ANEXO B – PROGRAMA COMENTADO

```

/* PROGRAMA ARDUINO UNO
* CONTROLE PI DA VELOCIDADE DE UM MOTOR CC
* LARISSA P. MONTENEGRO
* 2021
*/
// Inclui biblioteca do motor shield e do controle PID
#include <AFMotor.h>
#include<PID_v1.h>
AF_DCMotor motor(3); // Seleciona o motor 3
// VARIÁVEIS PID
#define pwm_Max 255 // Valor máximo de pwm do Arduino
#define pwm_Min 0 // Valor mínimo de pwm do Arduino
#define kp 0.2548 // Define variável kp
#define ki 1.247 // Define variável ki
#define kd 0 // Define variável kd
int pino_D0 = 2; // Atribui pino 2 ao sensor de velocidade
double rpm; // Define varável rpm
volatile byte pulsos; // Define variável pulsos
unsigned long tempo_passado; // Define variável tempo_passado
unsigned int pulsos_por_volta = 20; // Define quantas perfurações possui o disco encoder
double velocidade = 0; // output
double Setpoint1=130; // Primeiro valor de referência
double Setpoint2=180; // Segundo valor de referência
unsigned long X=20000; // Variável de controle
// Configuração inicial do PID
PID motorPID(&rpm, &velocidade, &Setpoint1, kp, ki, kd, DIRECT);
PID motorPID2(&rpm, &velocidade, &Setpoint2, kp, ki, kd, DIRECT);
// Interrupção
void contador()
{
// incrementa o contador
pulsos++;

```

```

}
void setup() {
  Serial.begin (115200); // Inicia a comunicação serial com velocidade de 115200 bps
  pinMode(pino_D0, INPUT);
  // Aciona o contador a cada pulso
  attachInterrupt (0,contador, FALLING);
  pulsos=0;
  rpm=0;
  tempo_passado=0;
  // Configuração de controle do motor
  motorPID.SetOutputLimits(pwm_Min,pwm_Max); // Limites de output do PID
  motorPID.SetMode(AUTOMATIC); // Configura PID como automático
  motorPID2.SetOutputLimits(pwm_Min,pwm_Max); // Limites de output do PID2
  motorPID2.SetMode(AUTOMATIC); // Configura PID como automático
}

void loop() {
  if (millis()- tempo_passado>= 100){
  // Desabilita a interrupção durante o cálculo
  detachInterrupt(0);
  // Realiza o cálculo de rpm
  rpm = (60 * 1000/ pulsos_por_volta) / (millis() - tempo_passado)* pulsos;
  tempo_passado= millis();
  pulsos=0;
  // Imprime os valores de velocidade em rpm e pwm
  Serial.print("Velocidade = ");
  Serial.print (velocidade, 2);
  Serial.print (" ");
  Serial.print ("RPM = ");
  Serial.println(rpm, 0);
  //Habilita interrupcao
  attachInterrupt(0, contador, FALLING);
}
  motorPID.Compute(); // Realiza o cálculo de controle
  motor.setSpeed(velocidade); // Define a velocidade do motor

```

```

motor.run(FORWARD); // Aciona o motor
// Inicia o controle para o segundo valor de setpoint
if (tempo_passado >= X){
    while (tempo_passado<=X+10000){
        if (millis()- tempo_passado>= 100){

// Desabilita a interrupção durante o cálculo
detachInterrupt(0);
// Realiza o cálculo do rpm
rpm = (60 * 1000/ pulsos_por_volta) / (millis() - tempo_passado)* pulsos;
tempo_passado= millis();
pulsos=0;
// Imprime os valores de velocidade em rpm e pwm
Serial.print("Velocidade = ");
Serial.print (velocidade, 2);
Serial.print (" ");
Serial.print ("RPM = ");
Serial.println(rpm, 0);
//Habilita interrupcao
attachInterrupt(0, contador, FALLING);
}
motorPID2.Compute(); // Realiza o cálculo de controle
motor.setSpeed(velocidade); // Define a velocidade do motor
motor.run(FORWARD); // Aciona o motor
}
}
// Incrementa a variável de controle;
if (tempo_passado>=X+10000) {
}
}

```