

UNIVERSIDADE FEDERAL DE ALFENAS

LUCAS EDUARDO RIBEIRO
VINICIUS NORONHA ALMEIDA MARTINIANO DE AZEVEDO

**DESENVOLVIMENTO DE UM CONTROLADOR PID DE VELOCIDADE PARA
MOTORES DE CORRENTE CONTÍNUA**

POÇOS DE CALDAS/MG

2025

LUCAS EDUARDO RIBEIRO
VINICIUS NORONHA ALMEIDA MARTINIANO DE AZEVEDO

**DESENVOLVIMENTO DE UM CONTROLADOR PID DE VELOCIDADE PARA
MOTORES DE CORRENTE CONTÍNUA**

Produto de Conclusão PIEPEX apresentada como parte dos requisitos para obtenção do título de Bacharel Interdisciplinar em Ciência e Tecnologia pela Universidade Federal de Alfenas.
Orientador: Prof. Dr. Laos Alexandre Hirano

POÇOS DE CALDAS/MG
2025

Sistema de Bibliotecas da Universidade Federal de Alfenas
Biblioteca Campus Poços de Caldas

Ribeiro, Lucas Eduardo.

Desenvolvimento de um controlador PID de velocidade para motores de corrente contínua / Lucas Eduardo Ribeiro, Vinicius Noronha Almeida Martiniano de Azevedo. - Poços de Caldas, MG, 2025.

54 f. -

Orientador(a): Laos Alexandre Hirano.

Trabalho de Conclusão de Curso (Graduação em Bacharelado Interdisciplinar em Ciência e Tecnologia) - Universidade Federal de Alfenas, Poços de Caldas, MG, 2025.

Bibliografia.

1. Controle. 2. Robótica Móvel. 3. Odometria. I. Azevedo, Vinicius Noronha Almeida Martiniano de. II. Hirano, Laos Alexandre, orient. III. Título.

Ficha gerada automaticamente com dados fornecidos pelo autor.

LUCAS EDUARDO RIBEIRO
VINICIUS NORONHA ALMEIDA MARTINIANO DE AZEVEDO

**DESENVOLVIMENTO DE UM CONTROLADOR PID DE VELOCIDADE PARA
MOTORES DE CORRENTE CONTÍNUA**

O(A) Presidente da banca examinadora abaixo assina a aprovação da apresentada como parte dos requisitos para obtenção do título de Bacharel em Interdisciplinar em Ciência e tecnologia pela Universidade Federal de Alfnas

Aprovada em: 05 de Dezembro de 2025

Prof. Dr^a. Carolina Del Roveri
Universidade Federal de Alfnas

Assinatura:

Prof^a. Dr. Flávio Augusto Dias
Universidade Federal de Alfnas

Assinatura:

Prof. Dr. Laos Alexandre Hirano (presidente)
Universidade Federal de Alfnas

Assinatura:

AGRADECIMENTOS

Agradecemos primeiramente ao professor Laos Alexandre Hirano, que confiou o projeto a nós e nos auxiliou tanto no desenvolvimento deste, como no nosso próprio desenvolvimento ao decorrer dos últimos anos, como orientador e como amigo. Agradecemos aos materiais cedidos pela LAPSE para a construção do protótipo desenvolvido, assim como pelo espaço para que pudéssemos desenvolvê-lo.

Agradecemos também, ao importante apoio cedido de nossos familiares, pais, avós, irmãos, servindo como pilares para nossa formação como indivíduos, e como estrutura para que nos fosse possível desenvolver o projeto. Lembramos também dos amigos e companheiros que estiveram presentes em toda essa etapa tão significativa de nossa formação, nos ajudando a lidar com os problemas enfrentados, e ajudando a lidar com toda a dificuldade enfrentada ao decorrer desse processo.

RESUMO

O trabalho apresenta o desenvolvimento de um controlador PID de velocidade aplicado a um robô móvel diferencial, este, foi criado para uso em atividades de robótica educacional no laboratório LAPSE e na rede RECARE, da Universidade Federal de Alfenas. A motivação central residiu na necessidade de garantir deslocamentos mais precisos e controlados, de modo a viabilizar aplicações didáticas em contextos de ensino STEAM. Inicialmente, são abordados os fundamentos teóricos relacionados à robótica móvel diferencial, ao comportamento de motores de corrente contínua, aos sistemas de controle em malha fechada, ao método de sintonia de Ziegler Nichols e à odometria baseada em encoder óptico. Na etapa experimental, foi desenvolvido um sistema de medição utilizando disco encoder com 20 furos e sensor infravermelho, bem como realizada a caracterização de dois motores, por meio da análise da variação da velocidade e da corrente em função da tensão aplicada, com ajuste de modelos via método dos mínimos quadrados e avaliação pelo teste de qui quadrado. O controlador PID é implementado em Arduino, utilizando a uma ponte H L298N e um sensor INA219 para monitoramento da corrente e tensão. Os resultados evidenciaram diferenças importantes entre os motores quanto à resposta em rotações por minuto e consumo de corrente, justificando a adoção de controladores independentes. Verifica-se que a sintonia clássica por Ziegler-Nichols não produz resultados plenamente satisfatórios nas condições e ferramentas disponíveis, ao passo que a sintonização empírica por tentativa e erro apresenta melhor desempenho, com overshoot e erro em regime considerados aceitáveis em faixas específicas de velocidade. Conclui-se que o controlador desenvolvido representa um avanço relevante na etapa de controle de velocidade e constitui base para a implementação futura de um controle cinemático completo, com potencial de aplicação em atividades de ensino técnico e de engenharia.

Palavras-Chave: Controle; Robótica Móvel; Odometria.

ABSTRACT

This work presents the development of a PID speed controller applied to a differential mobile robot, designed for use in educational robotics activities at the LAPSE laboratory and within the RECARE network at the Federal University of Alfenas. The central motivation lies in the need to ensure more precise and controlled motion, thereby enabling didactic applications in STEAM education contexts. Initially, the theoretical foundations related to differential mobile robotics, the behavior of DC motors, closed-loop control systems, the Ziegler–Nichols tuning method, and odometry based on an optical encoder are addressed. In the experimental stage, a measurement system is developed using an encoder disk with 20 slots and an infrared sensor, and the characterization of two motors is carried out by analyzing the variation of speed and current as a function of the applied voltage, with model fitting via the least squares method and evaluation using the chi-square test. The PID controller is implemented on an Arduino platform, using an L298N H-bridge and an INA219 sensor for current and voltage monitoring. The results show relevant differences between the motors in terms of revolutions-per-minute response and current consumption, which justifies the adoption of independent controllers. It is verified that classical Ziegler–Nichols tuning does not yield fully satisfactory results given the available conditions and tools, whereas empirical tuning by trial and error exhibits better performance, with overshoot and steady-state error considered acceptable within specific speed ranges. It is concluded that the developed controller represents a significant advance in the speed control stage and provides a basis for the future implementation of a complete kinematic controller, with potential application in technical and engineering education activities.

Keywords: Control; Mobile Robotics; Odometry.

LISTA DE ILUSTRAÇÕES

Figura 1 - Robô aspirador automático, analisando o espaço ao redor	11
Figura 2 - Ponte H L298N	13
Figura 3 - Movimento de robô com dois graus de liberdade	15
Figura 4 - Construção básica de robôs móveis com dois graus de liberdade	15
Figura 5 - Processo genérico de controlador em malha fechada	17
Figura 6 - Esquema genérico de um controlador PID	18
Figura 7 - Gráfico típico do método Ziegler e Nichols	20
Figura 8 - Modelo esquemático funcionamento de um encoder	21
Figura 9 - Arduino UNO	22
Figura 10 - Encoder	23
Figura 11 - INA 219	24
Figura 12 - Fontes de Tomada e de Bancada respectivamente	25
Figura 13 - Suporte IR	25
Figura 14 - Modelo usado para medição de deslocamento angular	27
Figura 15 - Fluxograma Hodômetro	28
Figura 16 - Montagem para avaliar curva característica do motor	28
Figura 17 - Fluxograma do programa para curva característica do motor	29
Figura 18 - Montagem para implementação dos controles PID	30
Tabela 1 - Tabela de cálculo de ganhos pelo método Ziegler e Nichols	20
Tabela 2 - Tabela de preços	30
Gráfico 1 - Curva Característica dos Motores	33
Gráfico 2 - Curva Característica dos Motores	34
Gráfico 3 - Teste de Ziegler Nichols	35
Gráfico 4 - Verificação do controlador Motor 1 velocidades (50, 75, 100)	36
Gráfico 5 - Verificação do controlador Motor 1 velocidades (125, 150, 175)	37
Gráfico 6 - Verificação do controlador Motor 1 velocidades (200, 225, 250)	38
Gráfico 7 - Verificação do controlador Motor 2 velocidades (50, 75, 100)	39
Gráfico 8 - Verificação do controlador Motor 2 velocidades (125, 150, 175)	39
Gráfico 9 - Verificação do controlador Motor 2 velocidades (200, 225, 250)	40
Gráfico 10 - Gráfico dos erros absolutos referentes ao desvio em regime	41
Gráfico 11- Comparação dos nas velocidades 125 e 150	42

SUMÁRIO

1. INTRODUÇÃO	10
2. OBJETIVOS	12
2.1. OBJETIVO PRINCIPAL	12
2.2. OBJETIVOS ESPECÍFICOS	12
3. REVISÃO BIBLIOGRÁFICAS	12
3.1. COMPOSIÇÃO BÁSICA DE UM ROBÔ MÓVEL	12
3.1.1. Funcionamento de uma ponte H	13
3.2. ROBÔS DIFERENCIAIS: O QUE SÃO, PRINCIPAIS CARACTERÍSTICAS, MODELO MATEMÁTICO	14
3.3. COMPORTAMENTO TÍPICO DE MOTORES CC	16
3.4. CONTROLADORES	16
3.5. CONTROLADORES DE VELOCIDADE APLICADOS A MOTORES CC	18
3.6. MÉTODO DE SINTONIZAÇÃO DE CONTROLADORES DE ZIEGLER-NICHOLS	19
3.7. ODÔMETRO BASEADO POR ENCODER ÓPTICO	20
4. METODOLOGIA	21
4.1. MATERIAIS UTILIZADOS	21
4.1.1. Arduino	22
4.1.2. Ponte H L298N	22
4.1.3. Motor DC 3V - 6V	22
4.1.4. Disco encoder	23
4.1.5. IR speed sensor	23
4.1.6. Multímetro	24
4.1.7. INA 219	24
4.1.8. Fontes	24
4.1.9. Chassi	25
4.1.10. Suporte IR Speed sensor	25
4.1.11. Custo dos materiais	26
4.2. DESENVOLVIMENTO DO HODÔMETRO BASEADO EM ENCODER ÓPTICO	27
4.3. CARACTERIZAÇÃO DOS MOTORES CC	28
4.4. IMPLEMENTAÇÃO DOS CONTROLADORES PID DE VELOCIDADE	30
5. RESULTADOS E DISCUSSÃO	32
5.1. DESEMPENHO DO ODÔMETRO BASEADO EM ENCODER ÓPTICO	32
5.2. CARACTERIZAÇÃO DOS MOTORES CC	32
5.2.1. DISCUSSÕES SOBRE OS TRABALHOS AVALIADOS	34
5.3. SINTONIZAÇÃO DOS GANHOS PID DOS CONTROLADORES DE VELOCIDADE	35
6. CONCLUSÃO	42
7. PERSPECTIVAS FUTURAS	43
REFERÊNCIAS	44

ANEXO A - CÓDIGO PARA CARACTERIZAÇÃO DOS MOTORES	47
ANEXO B - CÓDIGO DO CONTROLADOR PID	50

1. INTRODUÇÃO

Dentro da Universidade Federal de Alfenas, Campus Poços de Caldas, desenvolveu-se o projeto de pesquisa e extensão Laboratório de Prototipagem em Sistemas Embarcados - LAPSE, voltado ao desenvolvimento de dispositivos microcontrolados aplicados a diversos problemas de engenharia e educação STEAM (sigla de Ciências, Tecnologias, Engenharias, Artes e Matemática). Entre as atividades desenvolvidas dentro do LAPSE, a Rede de Capacitação e Apoio em Robótica Educacional – RECARE é um projeto de extensão em colaboração com a Superintendência Regional de Educação de Poços de Caldas (SRE-Poços de Caldas) e tem como objetivo viabilizar atividades de robótica em todas as escolas da circunscrição (UNIVERSIDADE FEDERAL DE ALFENAS, s.d.).

A RECARE desenvolve protótipos robóticos para divulgação do projeto e fins didáticos, elabora e oferece cursos e oficinas de robótica para professores do ensino fundamental II e médio e presta apoio, mentoria e consultoria às escolas parceiras para a implementação de um laboratório básico de robótica.

O termo robótica educacional caracteriza as atividades de apoio ao ensino curricular e desenvolvimento de habilidades cognitivas e sociais dos estudantes por meio de trabalhos individuais e/ou em grupo para planejamento, construção e programação de robôs e pequenos projetos de automação voltados a problemas com complexidade adequada aos limites dos estudantes. Especialmente no ensino STEAM que busca o desenvolvimento de habilidades e competências a partir de abordagens em Ciência, Tecnologia, Engenharias, Arte e Matemática, as atividades com robôs são convenientes por despertar a curiosidade e criatividade dos estudantes. (BENITTI, 2012).

Nesse contexto, atividades relacionadas à robótica móvel trazem atividades interessantes no contexto curricular por trabalhar com eletrônica básica, instrumentação, lógica de programação e controle de processos, desenvolvendo e/ou reforçando habilidades como pensamento computacional e lógica. Além disso, a robótica móvel envolve diversos problemas relacionados à rotação e translação de robôs dentro de um plano inercial que reforça conhecimento de geometria e permite o estudante desenvolver seu pensamento analítico.

Dentro da robótica móvel, um dos problemas mais clássicos é o deslocamento planejado em um plano com um referencial inercial inicial. Neste problema, o robô deve realizar manobras precisas permitindo que a plataforma mude sua posição atual para uma posição desejada. A título de exemplo, tem-se atualmente a utilização de aspiradores de pó

automáticos, equipamentos que necessitam de ferramentas para a verificação da localização deste, assim como do trajeto já percorrido, a fim de garantir que esse percorra de maneira eficiente o local planejado, funcionamento ilustrado na figura 1.

Figura 1 - Robô aspirador automático, analisando o espaço ao redor



Fonte: <https://techtudo>

No problema de deslocamento planejado, é essencial que o robô seja capaz de reconhecer a sua posição relativa à um referencial inercial. Uma abordagem é a odometria diferencial em que o robô registra a velocidade de cada roda em função do tempo para determinar a sua posição relativa. Nesta abordagem, além de um sistema para registrar o histórico de velocidade em função do tempo, um controle preciso da velocidade para cada roda é essencial para que o robô apresente um desempenho satisfatório no problema de deslocamento planejado (SIEGWART et al, 2011).

Um controlador é um dispositivo ou algoritmo responsável por regular o comportamento de um sistema dinâmico, ajustando suas entradas de forma a manter a saída próxima de um valor desejado, denominado referência ou *setpoint*. O controlador atua comparando a variável medida com a referência estabelecida e, a partir disso, gera um sinal de controle que corrige desvios entre o valor real e o valor desejado. Segundo Ogata (2010), o objetivo principal de um sistema de controle é obter desempenho estável, preciso e rápido, mesmo diante de perturbações externas ou variações nos parâmetros do sistema.

Este trabalho apresenta o desenvolvimento de controladores PID (proporcional, integral e derivativo) de velocidade aplicados a motores de corrente contínua (motores CC) com caixa de redução. Este trabalho é uma etapa que antecede ao desenvolvimento de um robô móvel com dois graus de liberdade equipados com controladores PID em seus motores,

buscando um controle sensível da posição e orientação do robô em relação a um referencial inercial.

2. OBJETIVOS

2.1. OBJETIVO PRINCIPAL

- Desenvolvimento de um controlador PID de velocidade para motores CC com caixa de redução.

2.2. OBJETIVOS ESPECÍFICOS

- a) Desenvolvimento de um odômetro com encoder ótico;
- b) Caracterização dos motores:
 - Velocidade em função da tensão;
 - Corrente aplicada sobre o motor em função da tensão;
- c) Desenvolvimento do controlador PID de velocidade:
 - Através do método de Ziegler-Nichols;
 - Por tentativa e erro.

3. REVISÃO BIBLIOGRÁFICAS

3.1. COMPOSIÇÃO BÁSICA DE UM ROBÔ MÓVEL

Um robô móvel é constituído por diferentes subsistemas interligados, cuja finalidade é permitir sua movimentação autônoma ou semi autônoma, bem como a interação com o ambiente ao seu redor. Esses subsistemas englobam elementos mecânicos, elétricos, eletrônicos e computacionais, que operam de forma integrada para garantir o funcionamento eficiente da plataforma robótica. De acordo com Siegwart, Nourbakhsh e Scaramuzza (2011), a arquitetura fundamental de um robô móvel é composta pelo sistema de locomoção, sistema de atuação, sistema de controle, sistema sensorial e sistema de energia, sendo esse conjunto considerado a base estrutural e funcional da maioria dos projetos na área de robótica móvel.

Craig (2012) destaca que o controlador é o centro de processamento da informação em sistemas robóticos, executando algoritmos de controle e gerenciando as informações provenientes dos sensores. Exemplos de sensores, pode-se considerar, sensores ultrassônicos, infravermelho e *bluetooth*. Por fim, o sistema de alimentação fornece energia adequada a

todos os componentes, sendo essencial para a autonomia do robô. A eficiência de um robô móvel depende da correta integração entre esses subsistemas, o que exige um projeto multidisciplinar.

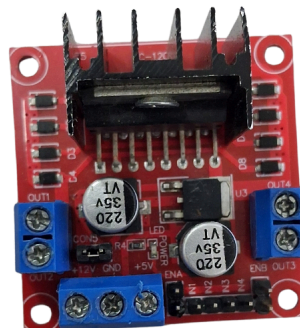
3.1.1. Funcionamento de uma ponte H

Pontes H são utilizadas para controlar motores de corrente contínua. Seu funcionamento baseia-se no princípio de que um motor DC gira em sentidos diferentes conforme a direção da corrente que passa por ele. Assim, ela usa conjuntos de transistores internos para inverter essa corrente e permitir que o motor gire tanto para frente quanto para trás.

O modelo utilizado no trabalho trata-se do L298N, que para controlar o movimento, possui duas entradas de controle por motor, como IN1 e IN2, e IN3 e IN4. Quando uma está em nível alto e a outra em nível baixo, o motor gira em um sentido; quando os níveis se invertem, o sentido também muda. Se ambas estiverem em nível baixo, o motor permanece parado. A velocidade é controlada enviando-se um sinal PWM aos pinos ENA ou ENB, que modulam a energia entregue ao motor. Assim, quanto maior o ciclo ativo do PWM, maior a velocidade de rotação.

O módulo possui ainda duas alimentações: uma para a lógica interna e outra para os motores, permitindo trabalhar com tensões diferentes. Como ele usa transistores bipolares, há uma queda de tensão interna relativamente alta, o que gera aquecimento, mas ainda assim o L298N é amplamente usado pela simplicidade e capacidade de controlar dois motores simultaneamente. O L298N pode ser visto na Figura 2.

Figura 2 - Ponte H L298N



Fonte: Autoral.

3.2. ROBÔS DIFERENCIAIS: O QUE SÃO, PRINCIPAIS CARACTERÍSTICAS, MODELO MATEMÁTICO

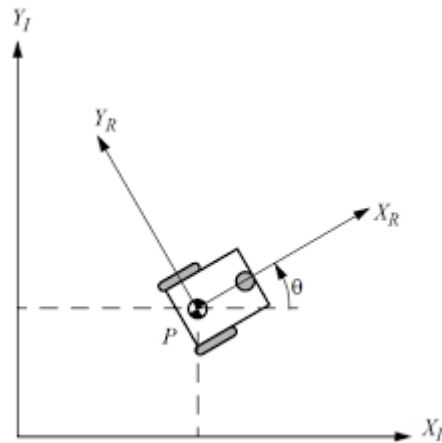
Robôs diferenciais são robôs móveis terrestres cujo movimento é obtido pela variação independente da velocidade de duas rodas motrizes paralelas, normalmente posicionadas em lados opostos do corpo do robô. A partir da diferença de velocidade entre essas rodas, o robô pode deslocar-se em linha reta, realizar curvas suaves ou girar sobre o próprio eixo.

Segundo Siegwart, Nourbakhsh e Scaramuzza (2011), o robô diferencial pertence à classe dos robôs de tração diferencial e caracteriza-se por possuir duas rodas acionadas de forma independente e, em geral, uma ou mais rodas passivas (rodas livres ou castores) destinadas apenas ao equilíbrio mecânico. A principal vantagem dessa configuração está na simplicidade construtiva e no controle relativamente direto da direção e da velocidade, sem a necessidade de mecanismos complexos de esterçamento.

Seja um robô diferencial, conforme representado na equação(1), Figura 3 e Figura 4, cujo ponto P é a posição do seu centro de massa, X_R e Y_R é a posição cartesiana em relação ao referencial inercial e θ é o ângulo da frente do robô em relação ao eixo x.

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \quad (1)$$

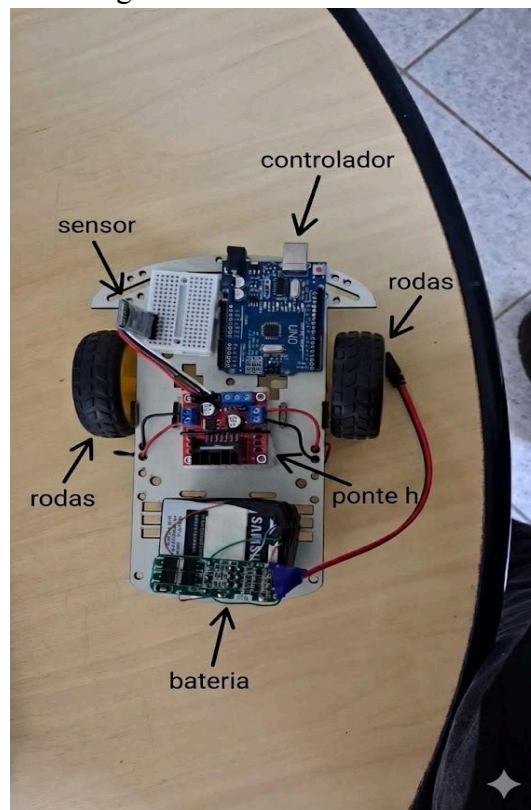
Figura 3 - Movimento de robô com dois graus de liberdade



Fonte: RODRIGUES (2019).

Para que seja possível estimar o ponto P e a orientação do protótipo, é essencial a utilização de um controlador de velocidade, visto que é não só necessário estimar o deslocamento das rodas, mas também garantir que as mesmas se comportem da maneira desejada.

Figura 4 - Construção básica de robôs móveis com dois graus de liberdade



Fonte: Autoral

3.3. COMPORTAMENTO TÍPICO DE MOTORES CC

Segundo Fitzgerald, Kingsley e Umans (2014), o motor CC converte energia elétrica em energia mecânica através da interação entre campos magnéticos gerados no estator e correntes que percorrem os enrolamentos da armadura. Quando uma tensão é aplicada aos terminais do motor, uma corrente elétrica passa a circular pela armadura, gerando um campo magnético que interage com o campo do estator, produzindo torque eletromagnético responsável pela rotação do eixo.

A modelagem matemática do motor CC é essencial para o desenvolvimento de sistemas de controle eficientes. De acordo com Franklin, Powell e Emami-Naeini (2014), o comportamento dinâmico do motor pode ser descrito por equações elétricas e mecânicas que, quando combinadas, resultam em um modelo capaz de representar o desempenho real do sistema. As equações descrevem a relação entre tensão aplicada, corrente, velocidade e torque, permitindo a obtenção de funções de transferência ou modelos em espaço de estados. Segundo Ogata (2010), esses modelos permitem prever a resposta do motor e fundamentar o projeto dos controladores. Equações elétricas citadas que podem em parte ser representadas pela equação (2).

$$v_i(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + v_g(t). \quad (2)$$

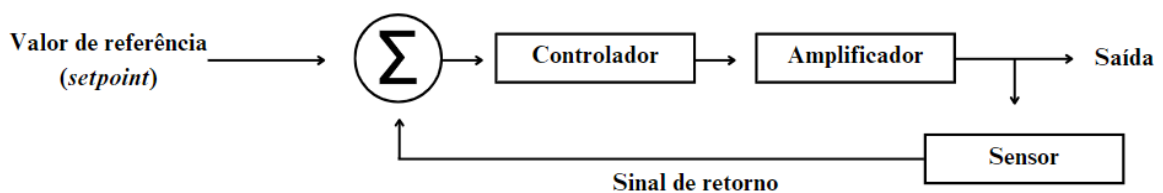
3.4. CONTROLADORES

Um controlador é um dispositivo ou algoritmo responsável por regular o comportamento de um sistema dinâmico, ajustando suas entradas de forma a manter a saída próxima de um valor desejado, denominado referência ou *setpoint*. O controlador atua comparando a variável medida com a referência estabelecida e, a partir disso, gera um sinal de controle que corrige desvios entre o valor real e o valor desejado. Segundo Ogata (2010), o objetivo principal de um sistema de controle é obter desempenho estável, preciso e rápido, mesmo diante de perturbações externas ou variações nos parâmetros do sistema.

Os sistemas de controle podem ser classificados em malha aberta e malha fechada. No controle em malha aberta, a ação de controle é independente da saída do sistema, ou seja, não

há realimentação. Isso significa que o sistema não corrige automaticamente eventuais erros causados por perturbações ou variações de carga. Em contrapartida, no controle em malha fechada ocorre a realimentação, na qual a saída é medida por sensores e comparada com o valor de referência, como observado na Figura 5. Dessa forma, o controlador ajusta constantemente o sinal de controle, promovendo maior precisão e robustez. De acordo com Nise (2011), sistemas em malha fechada apresentam melhor desempenho e menor sensibilidade a distúrbios, tornando-se essenciais em aplicações que exigem controle preciso.

Figura 5 - Processo genérico de controlador em malha fechada



Fonte: Citysystem, (2016).

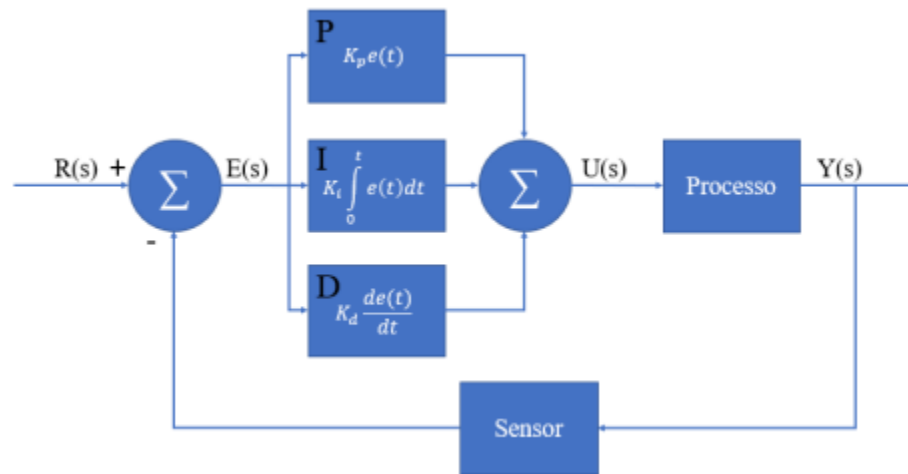
Entre os controladores mais utilizados em sistemas em malha fechada, destacam-se os controladores do tipo liga-desliga (*on-off*), proporcional (P), proporcional-integral (PI) e proporcional-integral-derivativo (PID).

O controlador *on-off* é o mais simples, operando de forma binária, acionando totalmente o atuador quando o erro é positivo e desligando-o quando o erro é negativo. Embora seja de aplicação simples e baixo custo, apresenta oscilações ao redor do ponto de equilíbrio, conforme descrito por Ogata (2010).

O controlador proporcional (P) gera um sinal de controle proporcional à magnitude do erro, proporcionando resposta mais rápida em relação ao controle *on-off*, porém pode resultar em erro em regime permanente. O controlador proporcional-integral (PI) adicionou uma ação integrativa ao controle proporcional, eliminando o erro estacionário e proporcionando maior precisão em regime permanente. Já o controlador proporcional-integral-derivativo (PID) incorpora a ação derivativa, que antecipa o comportamento do erro, aumentando a estabilidade e melhorando a resposta transitória do sistema.

Conforme Dorf e Bishop (2011), o controlador PID é amplamente empregado em aplicações industriais devido à sua eficiência e adaptabilidade em diferentes tipos de sistemas. O procedimento pode ser verificado na Figura 6.

Figura 6 - Esquema genérico de um controlador PID



Fonte: ALVES (2023).

A Figura 6 mostra um exemplo de controle em malha fechada com um controlador PID. Nela, a variável de saída do processo é medida e retorna ao sistema, onde é comparada com o valor de referência. A partir desse erro, atuam as ações proporcional, integral e derivativa, que são somadas para gerar novamente o sinal de controle aplicado ao processo.

Portanto, a escolha do tipo de controlador depende das características do sistema, dos requisitos de desempenho e das condições de operação. A correta seleção e sintonia dos parâmetros do controlador são fundamentais para garantir estabilidade, rapidez e precisão ao sistema controlado.

3.5. CONTROLADORES DE VELOCIDADE APLICADOS A MOTORES CC

O princípio fundamental do controle de velocidade é a utilização da realimentação, que permite comparar o valor de saída com a referência desejada, corrigindo continuamente o erro. Para Nise (2011), sistemas em malha fechada são indispensáveis quando se busca precisão e confiabilidade, pois conseguem compensar distúrbios e alterações nos parâmetros do sistema de forma automática. Isso é especialmente importante em motores CC, onde fatores como aquecimento e desgaste mecânico influenciam diretamente o desempenho.

No que se refere à implementação prática, a medição da velocidade pode ser realizada por meio de encoders, sensores *Hall* ou tacogeradores. Bose (2001) destaca que o uso da modulação por largura de pulso (PWM) é essencial para controlar a tensão aplicada ao motor, garantindo eficiência energética e bom desempenho. Em sistemas digitais, deve-se considerar os efeitos de atrasos de processamento e ruído de medição, sendo necessária a aplicação de filtros digitais para evitar erros de leitura, conforme discutido por Ogata (2010).

Os autores Pereira e Lopes (2018) apresentam a implementação prática de um controlador PID para o controle da velocidade angular de um motor de corrente contínua utilizando Arduino, ponte H e encoder magnético. O sistema foi caracterizado experimentalmente para relacionar o sinal PWM à velocidade do motor, permitindo a aplicação do controle em malha fechada.

Os ganhos do controlador (K_p , K_i e K_d) foram ajustados de forma experimental, sem métodos complexos de otimização. Os resultados mostraram bom desempenho para velocidades acima de 2,5 rad/s, indicando que o PID é eficiente mesmo com uma implementação simples.

3.6. MÉTODO DE SINTONIZAÇÃO DE CONTROLADORES DE ZIEGLER-NICHOLS

A sintonia de controladores é uma etapa fundamental em sistemas de controle automático, pois impacta diretamente na estabilidade, desempenho e robustez do sistema. Entre os métodos clássicos de sintonia de controladores PID, o método de Ziegler-Nichols destaca-se por sua simplicidade, ampla difusão e relevância histórica, sendo amplamente adotado em ambientes industriais e acadêmicos desde sua proposição.

Desenvolvido por John G. Ziegler e Nathaniel B. Nichols em 1942, esse método fornece regras empíricas baseadas em ensaios experimentais para ajuste dos parâmetros Proporcional (P), Integral (I) e Derivativo (D) de controladores PID (OGATA, 2010). Esse método baseia-se na resposta ao degrau da planta em malha aberta. Mede-se a curva resultante para identificar dois parâmetros:

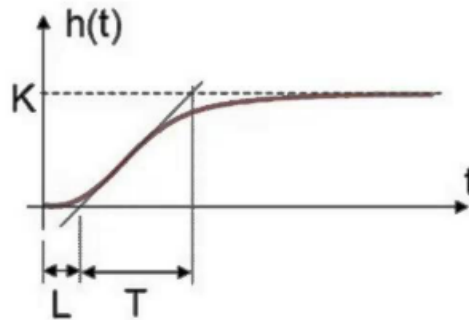
- Tempo de atraso (L): tempo até o início da resposta;
- Constante de tempo (T): taxa de crescimento da resposta.

A curva é aproximada por uma reta tangente no ponto de maior inclinação, conforme o modelo de primeira ordem com atraso. Com base nesses parâmetros, utilizam-se fórmulas tabeladas para calcular os ganhos do controlador P, PI ou PID (OGATA, 2010).

No segundo método, a planta é colocada em malha fechada apenas com controle proporcional. O ganho é aumentado progressivamente até que ocorram oscilações sustentadas na saída. O ganho correspondente é chamado de ganho crítico e o período dessas oscilações é conhecido como período crítico. A partir desses valores, os parâmetros do controlador PID são definidos conforme regras empíricas estabelecidas por Ziegler e Nichols, onde K , T_i e T_d são determinados da maneira apresentada na Figura 7 e é calculado da maneira apresentada na

Tabela 1. É importante relevar que durante o documento será referido a K , T_i e T_d como K_P , K_I e K_D respectivamente.S

Figura 7 - Gráfico típico do método Ziegler e Nichols



Fonte: Scoeerg (2019).

Tabela 1 - Tabela de cálculo de ganhos pelo método Ziegler e Nichols

Tipo de controlador	K_p	T_i	T_d
P	T/L	∞	0
PI	$0,9 T/L$	$L/0,3$	0
PID	$1,2 T/L$	$2L$	$0,5L$

Fonte: STROSKI (2019).

3.7. ODÔMETRO BASEADO POR ENCODER ÓPTICO

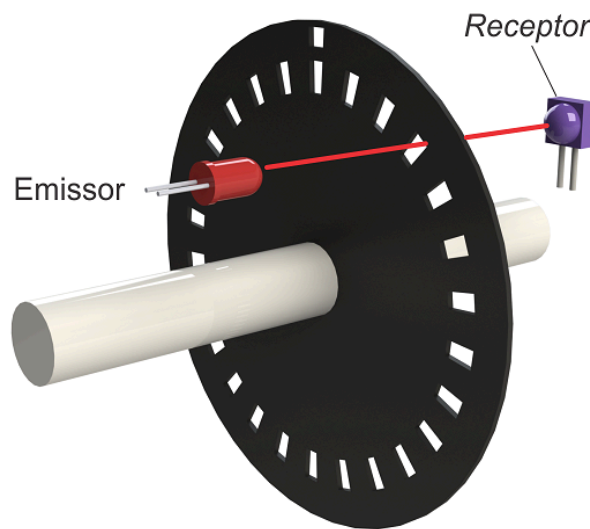
A odometria consiste na estimativa da posição e orientação de um sistema móvel a partir de sensores internos, normalmente baseando-se na medição do deslocamento de elementos mecânicos como rodas ou eixos. No contexto da robótica móvel e de veículos autônomos, essa técnica é amplamente considerada devido ao seu baixo custo, simplicidade de implementação e elevada taxa de resposta, sendo frequentemente empregada como fonte primária de posicionamento em trajetórias curtas (SICILIANO et al., 2016).

Nesse contexto, o método mais utilizado para realização da odometria é o emprego de encoders, dispositivos capazes de converter movimento mecânico em sinais elétricos proporcionais à rotação do eixo em que são acoplados. Quando o encoder empregado é do tipo óptico, a leitura é feita através da interrupção de um feixe de luz por um disco perfurado

ou padronizado, permitindo alta precisão e confiabilidade na contagem de pulsos (BOLTON, 2016).

O encoder óptico é composto basicamente por uma fonte emissora de luz (LED infravermelho), um sensor receptor e um disco codificado inserido entre ambos. À medida que o eixo gira, o disco interrompe o feixe luminoso, gerando pulsos digitais que podem ser contados por sistemas embarcados, com microcontroladores e controladores lógicos programáveis (CLPs) (BOLTON, 2016), como indicado na figura 8.

Figura 8 - Modelo esquemático funcionamento de um encoder



Fonte: Adaptado de Ogata (2010).

4. METODOLOGIA

4.1. MATERIAIS UTILIZADOS

Um ponto importante a ser destacado sobre os materiais utilizados, é o critério de escolha apresentado para esses. Os materiais utilizados são os mesmos que são utilizados nos kits de robótica educacional que são disponibilizados pela recare, portanto são materiais escolhidos principalmente devido ao baixo preço, o que muitas vezes acaba por limitar a qualidade dos materiais comprados, porém, torna os resultados obtidos replicáveis para aquelas escolas atingidas pelo projeto.

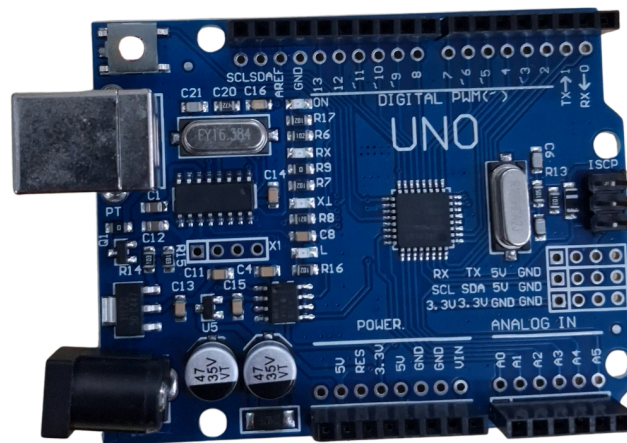
O controlador desenvolvido tem como objetivo equiparar a atuação dos dois motores, um controlador eficiente seria capaz de realizar esses ajustes mesmo que com motores com comportamentos distintos. Portanto, a utilização de motores que tivessem diferenças em seus

comportamentos era de grande interesse, certificaria a capacidade do controlador. A escolha desses no entanto foi feita de forma aleatória, escolhendo ao acaso o motor dentre os disponíveis para sua utilização pelas escolas, logo, alguma possível diferença em seu funcionamento não seria induzida, mas sim de ocorrência natural tornando a prática replicável nas atividades desenvolvidas pelas escolas.

4.1.1. Arduino

A plataforma de computação física embarcada Arduino UNO se faz um componente imprescindível devido a seu baixo custo e fácil aplicação em protótipos como já citado anteriormente. Pode-se observar um Arduino na Figura 9.

Figura 9 - Arduino UNO



Fonte: Autoral.

4.1.2. Ponte H L298N

A ponte H L298N explicitada em tópicos anteriores foi utilizada.

4.1.3. Motor DC 3V - 6V

Os motores DC são indispensáveis para a construção do protótipo visto seu baixo valor econômico e relativamente baixa tensão para superar a inércia. O motor DC funciona transformando energia elétrica em movimento. Quando a corrente passa pelas bobinas do rotor, forma-se um campo magnético que interage com os ímãs fixos do estator. Essa interação cria forças que fazem o rotor girar.

O comutador e as escovas têm a função de inverter continuamente o sentido da corrente nas bobinas, garantindo que o motor continue girando sem parar. Caso a corrente elétrica for aplicada no sentido oposto o motor simplesmente roda no sentido contrário, se demonstrando um motor ideal para robôs móveis de baixo custo.

4.1.4. Disco *encoder*

A fim de compor um odômetro foi acoplado a cada um dos motores DC um disco *encoder* que se constitui de um disco fracionado em 20 furos passantes equidistantes entre si como exemplificado na Figura 10.

Figura 10 - *Encoder*



Fonte: Autoral.

4.1.5. IR speed sensor

Para mensurar a velocidade foi utilizado como odômetro pois um sensor de velocidade IR com LM393 funciona utilizando um par emissor–receptor de infravermelho e um comparador eletrônico para detectar quando um objeto interrompe a luz. O sensor possui um LED infravermelho que emite luz e um fotodiodo que recebe essa luz. Quando nada está entre eles, o receptor recebe a luz normalmente.

O LM393, que é um comparador de tensão, interpreta essa variação de luz como uma mudança elétrica. Ele compara o nível de sinal vindo do foto transistor com um valor de referência ajustado pelo potenciômetro do próprio módulo. Quando a luz chega ao receptor, o sinal elétrico fica acima ou abaixo do limite, e quando a luz é bloqueada, o sinal muda de estado. O LM393 transforma essa diferença em uma saída digital limpa, gerando pulsos 0 e 1 bem definidos. Cada interrupção da luz produz um pulso elétrico.

Os pulsos elétricos por sua vez são facilmente contados por um microcontrolador como Arduino. Quando se acopla o disco *encoder* aos motores de modo a ficar entre o emissor e o receptor de luz torna-se possível calcular o RPM dos motores, devido ao formato do disco, basta contar a quantidade de pulsos em um intervalo de tempo.

4.1.6. Multímetro

Com intuito de averiguar a validade dos testes contemplados pelo trabalho e a qualidade das medições do sensor INA 219, foram usados dois multímetros, um como voltímetro e outro como amperímetro. O multímetro utilizado para os testes foi da marca Hikari modelo hm 1100.

4.1.7. INA 219

O INA 219 funciona medindo a corrente, a tensão e a potência de um circuito DC por meio de um resistor *shunt* e de um conversor analógico-digital interno. O processo começa quando a corrente da carga passa pelo resistor *shunt* (normalmente $0,1 \Omega$). Essa passagem gera uma queda de tensão muito pequena, chamada *Vshunt*, que o INA 219 mede com alta precisão. A corrente é então calculada pela relação básica da lei de Ohm:

Assim, conhecendo o valor do resistor shunt, o sensor determina exatamente a corrente que circula no circuito.

Além disso, o INA 219 mede a tensão do barramento, ou *Vbus*, que é a tensão aplicada à carga. Esse valor é obtido diretamente pelo conversor interno do sensor e representa a tensão real no ponto de alimentação.

Com as duas grandezas medidas corrente (*I*) e tensão (*Vbus*) sendo enviadas ao monitor *serial*, foi possível avaliar esses dados quase simultaneamente ao RPM, além de automatizar a coleta das informações. O INA 219 usado pode ser exemplificado pela figura 11.

Figura 11 - INA 219



Fonte: Autoral

4.1.8. Fontes

Na fase experimental do estudo se demonstrou necessário a utilização de duas fontes, garantindo tensões constantes, de forma a facilitar a interpretação dos fenômenos que ocorreram na prática. Na etapa de curvas características dos motores foi utilizado uma fonte de bancada modelo Sunshine P-3005A, que também foi doada pela Receita Federal; e uma

fonte de tomada de 9V e 0,6A de modelo tp-link model T090060 foi usada para os testes de controle de velocidade. As fontes, convencional de tomada e de bancada, azul Sunshine P-3005A foram respectivamente representadas na Figura 12.

Figura 12 - Fontes de Tomada e de Bancada respectivamente



Fonte: Autoral.

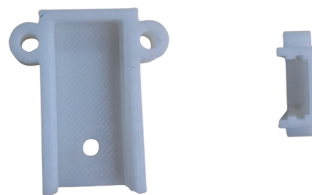
4.1.9. Chassi

Para suportar os testes propostos foi necessário utilizar um chassi de MDF cortado por uma cortadora a *laser* na própria Lapse.

4.1.10. Suporte IR Speed sensor

Foi impresso pela impressora 3D do laboratório LAPSE suportes para fixar o odômetro escolhido no chassi, garantindo melhor estabilidade e diminuindo os ruídos durante as medições. O suporte foi impresso em PLA branco e o modelo foi obtido no *site* Thingiverse e se encontra disponível no *site* da LAPSE e Exemplificado na Figura 13

Figura 13 - Suporte IR



Fonte: Autoral.

4.1.11. Custo dos materiais

A Tabela 2 representa os custos dos materiais estritamente necessários para que seja possível a realização do procedimento, visto que uma parte dos materiais são improvisáveis e portanto sua precificação depende da preferência do usuário, por exemplo, fontes, multímetros e o chassi ou suporte . A tabela já leva em consideração o valor dos materiais que necessitam de componentes repetidos.

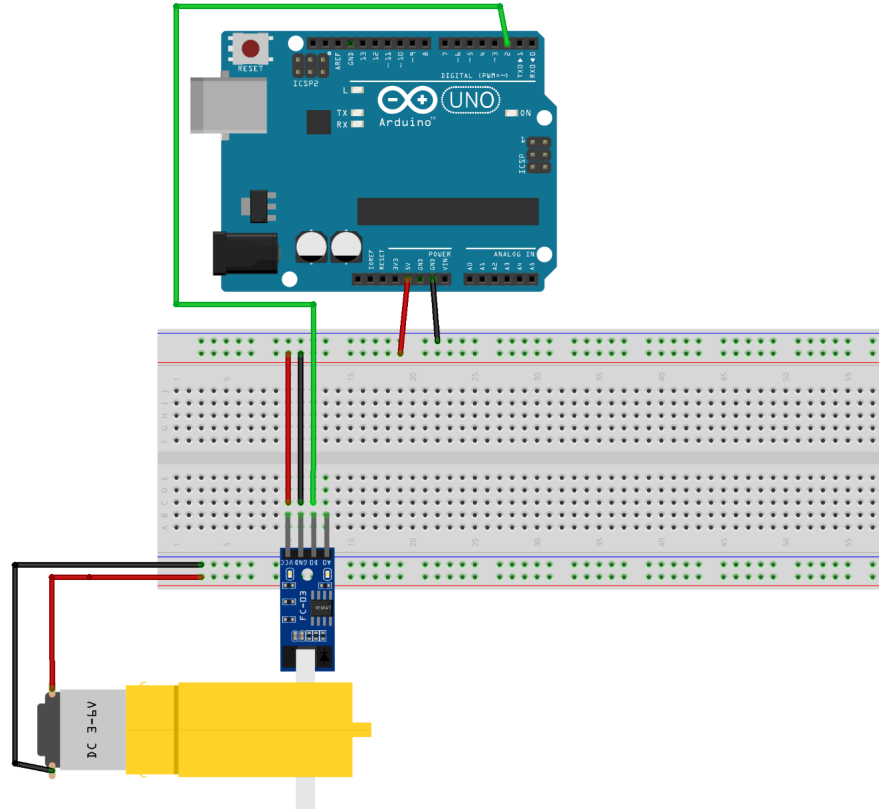
Tabela 2 - Tabela de preços

Materiais	Custo
Arduino UNO	R\$ 10,00
Disco Encoder	R\$ 4,00
Ina 219 - model HW831-B	R\$ 7,00
Motores DC 3 - 6V	R\$ 20,00
IR speed Sensor - model L393	R\$ 11,20
Ponte H L298N - model FC-120	R\$ 12,00
Preço total	R\$ 64,20

Fonte: Autoral.

4.2. DESENVOLVIMENTO DO HODÔMETRO BASEADO EM ENCODER ÓPTICO

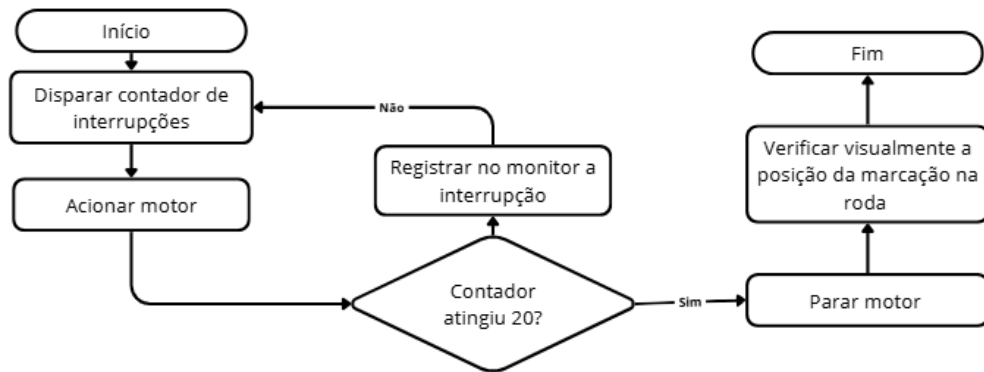
Figura 14 - Modelo usado para medição de deslocamento angular



Fonte: Autoral.

Com a finalidade de medir as rotações por minuto de um motor foi realizada a montagem verificada na Figura 14, nela pode-se observar, conectado à porta número 2 do Arduino UNO o IR *speed* sensor que foi alimentado pelo 5V da própria placa de prototipagem. Paralelo a primeira montagem, o motor teve seu disco *encoder* acoplado a um dos lados do eixo e do outro uma roda de maneira a tornar evidente deslocamento angular do motor.

Figura 15 - Fluxograma Hodômetro

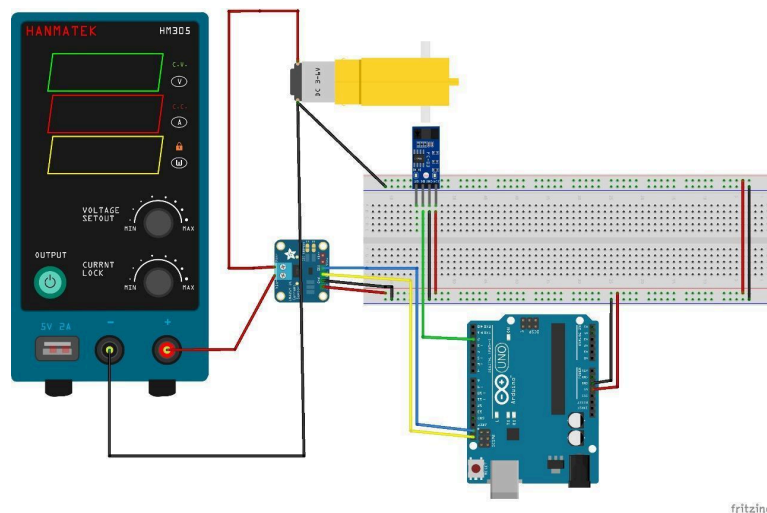


Fonte: Autoral.

Na Figura 15 demonstra-se o fluxo desenvolvido com intuito de validar o deslocamento através do número de pulsos captados pelo sensor óptico. Foi realizada uma marcação na roda do motor a ser medido, e no momento que o contador atingia 20 interrupções era verificada a posição da marcação, caso, diferente da posição inicial eram feitos ajustes, até atingir uma medida ideal.

4.3. CARACTERIZAÇÃO DOS MOTORES CC

Figura 16 - Montagem para avaliar curva característica do motor



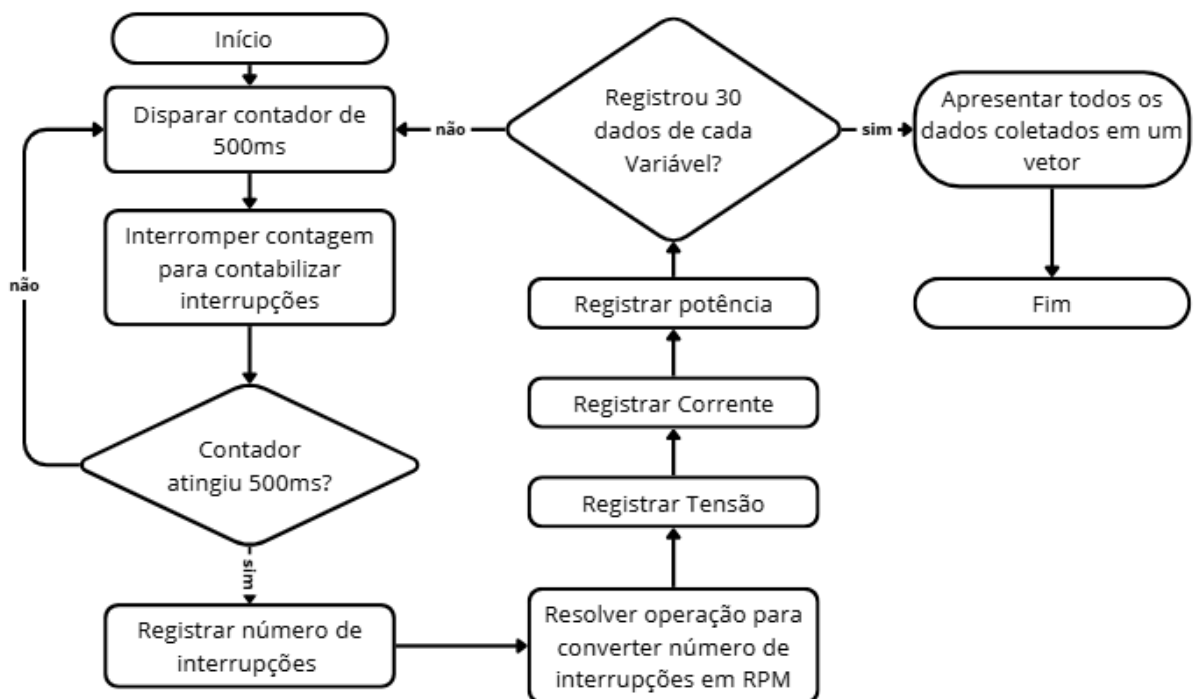
Fonte: Autoral.

A figura 16 representa a montagem realizada para que fosse possível analisar o comportamento individual dos dois motores, vale ressaltar que a montagem foi realizada para ambos. O esquema contém o Arduino UNO conectado ao hodômetro previamente definido, e ao INA 219, dessa forma o Arduino era capaz de registrar a tensão, a corrente e a velocidade.

O IR *speed* sensor, estava conectado na porta número 2 do Arduino e era alimentado pelo 5V que a placa de prototipagem fornecia ao ser conectado via USB. O INA 219 estava conectado às portas de comunicação IDC do Arduino e também era alimentado pelos 5V.

O motor por sua vez era alimentado por uma fonte de bancada de maneira a ser possível definir diferentes tensões e avaliar diferentes velocidades por consequência. O INA219 tinha seus terminais de medição, V1 e V2, respectivamente, conectados à fonte e ao motor de modo a ser capaz de acompanhar as variações de voltagem e amperagem. Todos os sensores e equipamentos por fim convergiam a um mesmo *ground*.

Figura 17 - Fluxograma do programa para curva característica do motor



Fonte: Autoral.

A Figura 17 descreve a atividade realizada pela montagem. Os testes para avaliar o comportamento dos motores em diferentes tensões consistiu em ajustar a fonte de bancada em 9V para não sobrecarregar o motor, e 0,5A para garantir a segurança dos equipamentos usados. Em seguida iniciar o motor a ser testado, e o Arduino simultaneamente. Inicialmente era disparado um contador de 500 milissegundos, e em frações de instantes interrompia-se o contador para contar a interrupções e registrar os dados de corrente, tensão potência, a velocidade era encontrada através de um cálculo que dividia o número de interrupções por 20, que era o total do *encoder*, e 0,5 segundos, que era o tempo de amostragem e multiplicar por 60 segundos a fim de extrapolar a contagem para um minuto, dessa forma identificando RPM,

enquanto essa atividade não se repetia 30 vezes, o programa não era encerrado. O código completo pode ser visto no Anexo A.

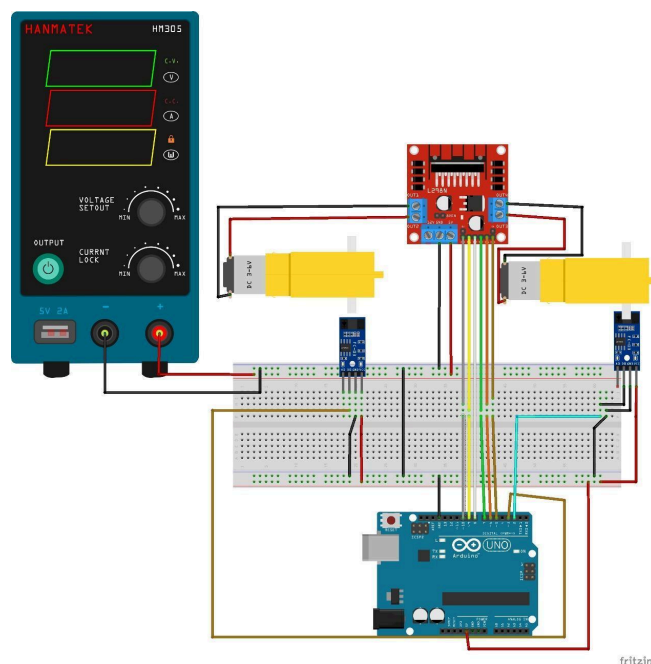
Após o fim do processo de registrar 30 intervalos, realizado pelo microcontrolador, foi diminuído 0,5V da tensão inicial ajustada, e reiniciou-se o Arduino. Esse procedimento foi repetido até que a tensão fosse zerada, e foram feitas triplicatas, para ambos os motores. Esse protocolo resultou, portanto, em, 90 dados por *setpoint* de tensão

Os dados foram tratados e transformados em um gráfico considerando todas as incertezas diagnosticadas na atividade e ajustadas a uma reta pelo método do qui quadrado representado na Equação 3. Vale ressaltar que o regime foi aplicado a cada um dos motores individualmente.

$$\chi^2 = \sum_{i=1}^{m \times n} \frac{(O_i - E_i)^2}{E_i} \quad (3)$$

4.4. IMPLEMENTAÇÃO DOS CONTROLADORES PID DE VELOCIDADE

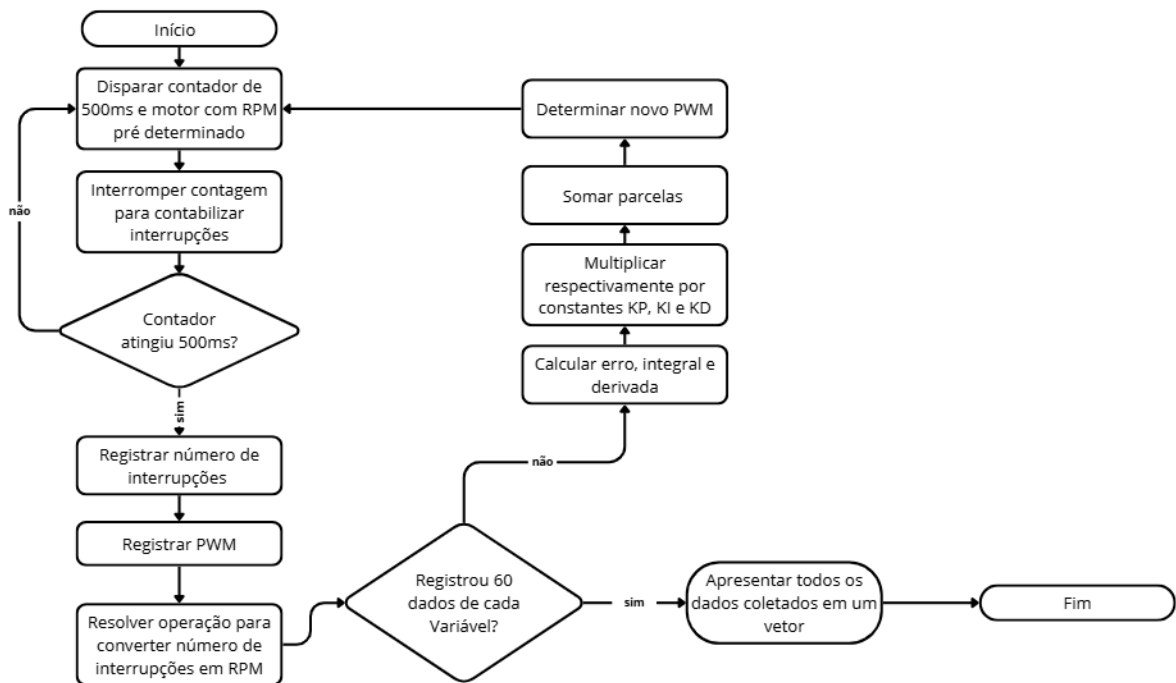
Figura 18 - Montagem para implementação dos controles PID



Fonte: Autoral.

A montagem segue a Figura 18 e consiste em conectar a ponte H ao Arduino e a fonte de bancada, cuidando para ligar as entradas que controlam o PWM nas portas PWM do Arduino. Os sensores foram alimentados pela porta 5V, e o sinal dos dois sensores infravermelhos nas portas 2 e 3 do Arduino que por sua vez era alimentado pela porta USB. Deve-se ressaltar que tudo foi aterrado junto garantindo o funcionamento do sistema. O algoritmo para os dois métodos seguiu o fluxograma representado pela Figura 19. Da mesma maneira que o código anterior define-se espaços temporais de 500 milissegundos onde são contadas interrupções, porém dessa vez é determinado um *setpoint* como desejado, e o controlador tenta suprir o erro através do cálculo encontrado no Anexo B.

Figura 19 – Algoritmo executado para realização dos métodos Ziegler-Nichols e Tentativa e erro



Fonte: Autoral.

4.4.1. Descrição do protocolo de testes realizados pelo método Ziegler-Nichols

A Figura 18 representa a montagem utilizada no protocolo para verificação do Ziegler Nichols foram testes onde se avalia a velocidade dos motores durante 30 segundos com o PWM fixo, coletando pontos no processo. Esse teste foi feito com alguns determinados valores de PWM com a intenção de realizar a método para sistemas de primeira ordem, portanto a intenção é coletar aspecto da curva do motor até que essa se estabilizasse, para que fosse possível estabelecer os determinados os ganhos.

4.4.2. Descrição da sintonização por tentativas e erros

Para a determinação dos ganhos, foi necessário recorrer ao método de estimar as constantes até que essas gerassem um comportamento desejado no sistema, como descrito por Tavares, (2019) o método de tentativa e erro, ou descida ordenada, pode ser sim uma opção interessante para a determinação dos ganhos para um sistema de motor CC, já que nesse trabalho os ganhos atingidos por esse método geraram uma resposta melhor do que com outros tipos de métodos matemáticos como o próprio ziegler-nichols, ou Cohen-Coon.

Foi utilizada a montagem referenciada na figura 18. Os valores dos ganhos K_P , K_I , K_D partiram de 0,6 e foram ajustados gradativamente, variando de 0,1, para cima ou para baixo cada um deles, e posteriormente somando ou subtraindo todos os ganhos ao mesmo tempo e verificando seu desempenho resultante. Isso foi repetido até alcançar um resultado satisfatório com flutuação de até 5% em relação ao valor de setup os controladores foram realizados 5 vezes para cada motor, cada teste contempla 60 pontos por medição, os pontos continham o RPM avaliados a cada 500 milissegundos. Essas avaliações foram realizadas partindo de 50 RPM até 250 RPM, considerando um intervalo de 25 RPM entre cada *setpoint*.

5. RESULTADOS E DISCUSSÃO

5.1. DESEMPENHO DO ODÔMETRO BASEADO EM *ENCODER* ÓPTICO

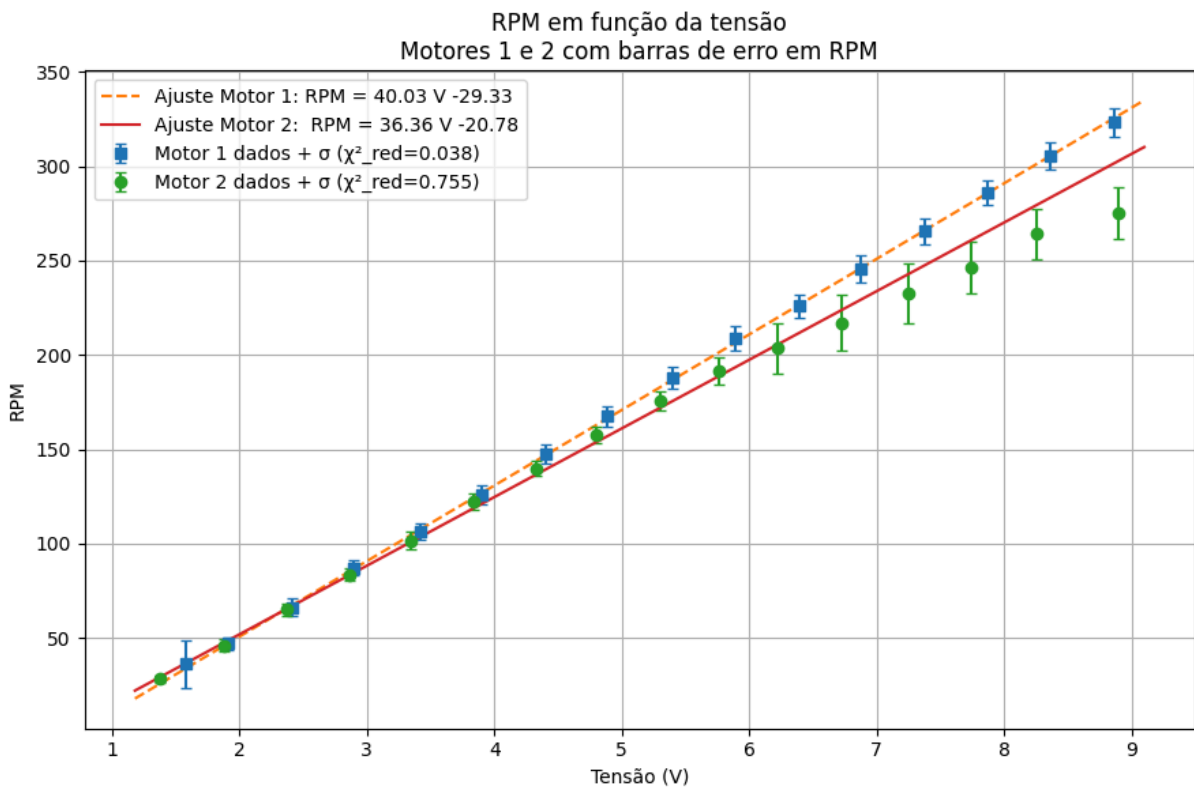
O desempenho do odômetro é de fundamental importância para a análise dos dados que se seguem, isso porque é a partir dos dados obtidos com relação à rotação que serão obtidos todos os outros dados dos testes e principalmente no que se refere ao funcionamento do próprio controlador em questão.

Para a definição do intervalo de contagem da velocidade, viu-se a necessidade de que esse fosse o menor possível, porém que esse tamanho do intervalo não interferisse de forma significativa na velocidade real do motor, para isso foi então contabilizada a velocidade deste motor medida dentro de 1 minuto, e esse tempo foi sendo diminuído até que se chegasse em um tempo mínimo satisfatório, sendo esse de 500 milissegundos.

5.2. CARACTERIZAÇÃO DOS MOTORES CC

Para um melhor entendimento da necessidade de um controlador de velocidade para os motores, foi feita a curva característica do comportamento desses que relaciona a tensão aplicada no motor pela sua velocidade em RPM's o que possibilita entender se realmente há alguma diferença quanto a rotação desses dois motores quanto a tensão aplicada, através do ajuste feito, é possível entender a forma como o motor se comporta, coletando assim seu coeficiente linear, necessário para entender a variação da rotação entre um motor e outro.

Gráfico 1 - Curva Característica dos Motores



Fonte: Autoral.

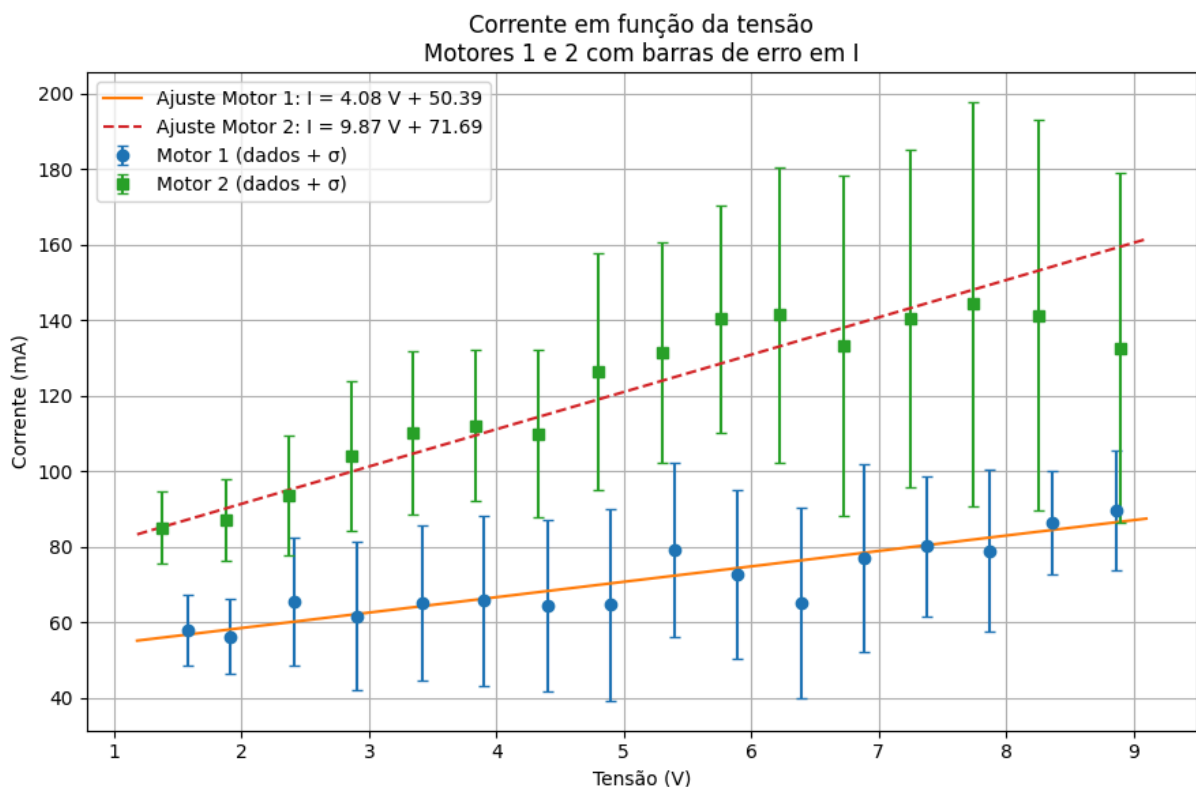
O Gráfico 1 apresenta a curva característica de tensão por RPM dos motores utilizados, o ajuste utilizando o MMQ permitiu, através do ajuste, obter o coeficiente linear e angular das retas de tendência dos resultados obtidos. Outra informação importante a ser observada é o ponto mínimo de tensão que permite o funcionamento dos motores, esse ponto se mostrou nos dois gráficos entre 1.5V e 2V, não sendo observado nenhum movimento nos motores abaixo desta tensão, indicando que essa é a tensão mínima de funcionamento dos motores.

O coeficiente angular da reta traçada para o primeiro motor foi de 36,36V, enquanto que o coeficiente referente ao segundo motor foi de 40,03V, uma diferença de aproximadamente 10%, suficiente para criar uma diferença, próxima a 50 RPM quando testada na maior tensão (9V). Os testes foram suficientes para determinar a necessidade de um controlador para as velocidades, porque apesar dessa se dar em função da tensão aplicada, não se distribuí de forma igualitária aos dois motores, que dependem de seus coeficientes lineares para a determinação dessa. Um importante fator a ser considerado é que essa curva característica obtida pode ser influenciada por fatores envolvidos na construção do motor, como sua transmissão e mesmo fatores como a manutenção desse, fatores que podem ser abordados com mais detalhes em trabalhos futuros.

O ajuste das retas foi avaliado por meio do teste do χ^2 . Para o primeiro motor, obtiveram-se χ^2 de 10,57 e χ^2 reduzido de 0,755, enquanto para o segundo motor os valores foram de 0,536 para o χ^2 e 0,038 para o χ^2 reduzido, o que indicam que esse ajuste realizado é adequado para a dispersão dos dados apresentados, atribuindo maior confiabilidade ao processo.

5.2.1. Discussões sobre os trabalhos avaliados

Gráfico 2 - Curva Característica dos Motores



Fonte: Autoral.

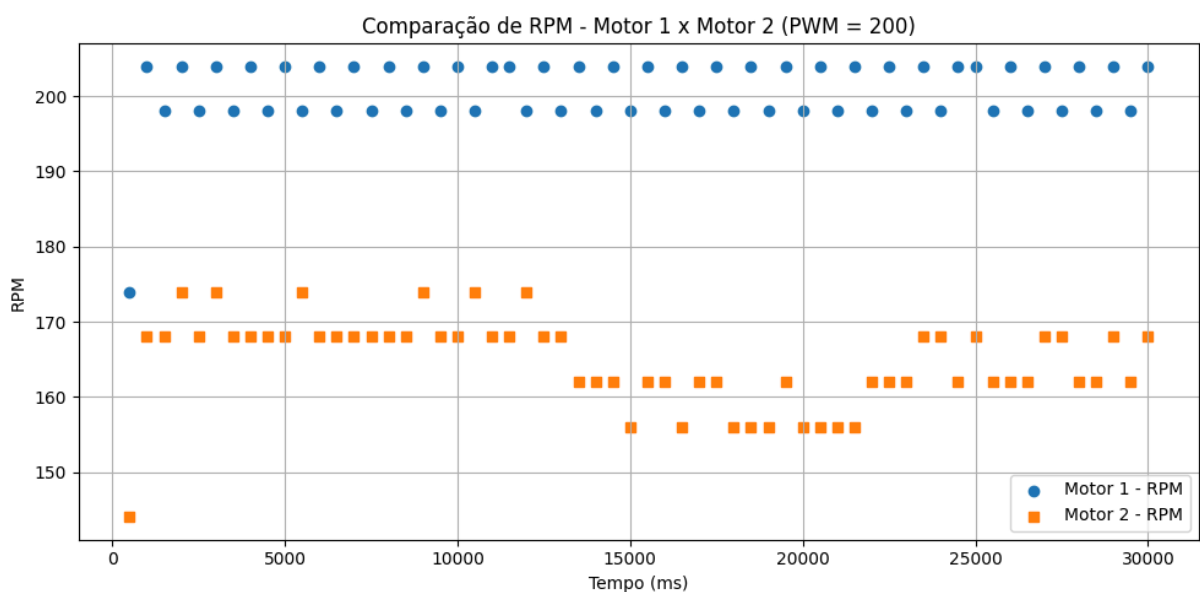
No Gráfico 2 é apresentado o teste de corrente consumida a partir da tensão, onde novamente pôde-se averiguar um comportamento linear ajustado a partir de MMQ. O coeficiente angular do motor 1 foi de 4,08, enquanto o do motor 2 foi de 9,87, mais que o dobro, além do motor 2 apresentar um coeficiente linear de 71,69, valor maior que o do motor 1 com 50,39. Essa diferença identificada indica que o Motor 2 aumenta o consumo de corrente duas vezes mais rápido que o motor 1, demandando mais potência elétrica. Uma maior variação dos dados pode ser observada no motor 2, podendo essa ter origem em detalhes construtivos do próprio motor.

O ajuste realizado no Gráfico 2 foi verificado assim como no primeiro gráfico pelo método do χ^2 , obtendo um valor de 0,618 para o Motor 1, com o χ^2 reduzido de 0,0441. O ajuste feito no Motor 2 apresentou os valores de 0,976 para o χ^2 e 0,0697 para o χ^2 reduzido.

5.3. SINTONIZAÇÃO DOS GANHOS PID DOS CONTROLADORES DE VELOCIDADE

Quando aplicado o teste de Ziegler Nichols, como é possível ver no Gráfico 3, a estabilização do sistema aconteceu de forma muito rápida, tornando difícil a coleta dos dados necessários para a composição da curva gerada para verificação do teste em sua primeira ordem. Essas limitações instrumentais e com relação a coleta de dados tornam o teste inconclusivo, impossibilitando a sua utilização de forma confiável para o trabalho.

Gráfico 3 - Teste de Ziegler Nichols



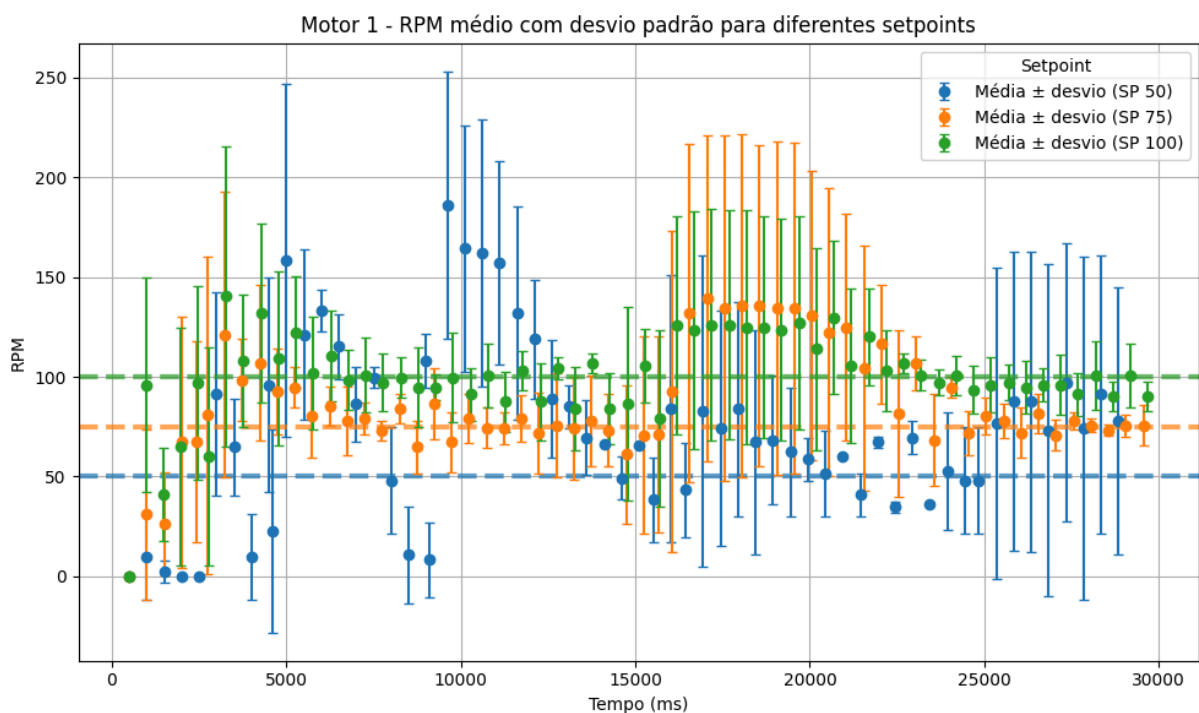
Fonte: Autoral.

O método de tentativa e erro foi aplicado para o segundo motor, onde foram obtidos os valores de valor de 0,54 para o KP, 0,60 para o KI e 0,11 para o Kd. medidas que geraram testes com bons resultados para o motor 2, motor com menor coeficiente angular, porém, o motor 1 não se adequava aos parâmetros estabelecidos, devido a variação quanto a resposta a voltagem que pôde ser vista na curva característica do motor, foi portanto gerados ganhos com um KP aproximadamente 10% menor que o do motor 2, no valor de 0,49, o que representou um resultado que pode compensar em parte a diferença existente entre os motores.

Para a visualização adequada dos resultados, foi feita a separação de 3 *setpoints* de velocidade definidos para o controlador em cada gráfico de forma crescente, essa divisão facilita a análise visual do funcionamento dos motores, além de prevenir que *setpoints* muito baixos ou muito altos, que podem ter maiores variações, interferem na visualização de *setpoints* onde o controlador se encontra bem dimensionado.

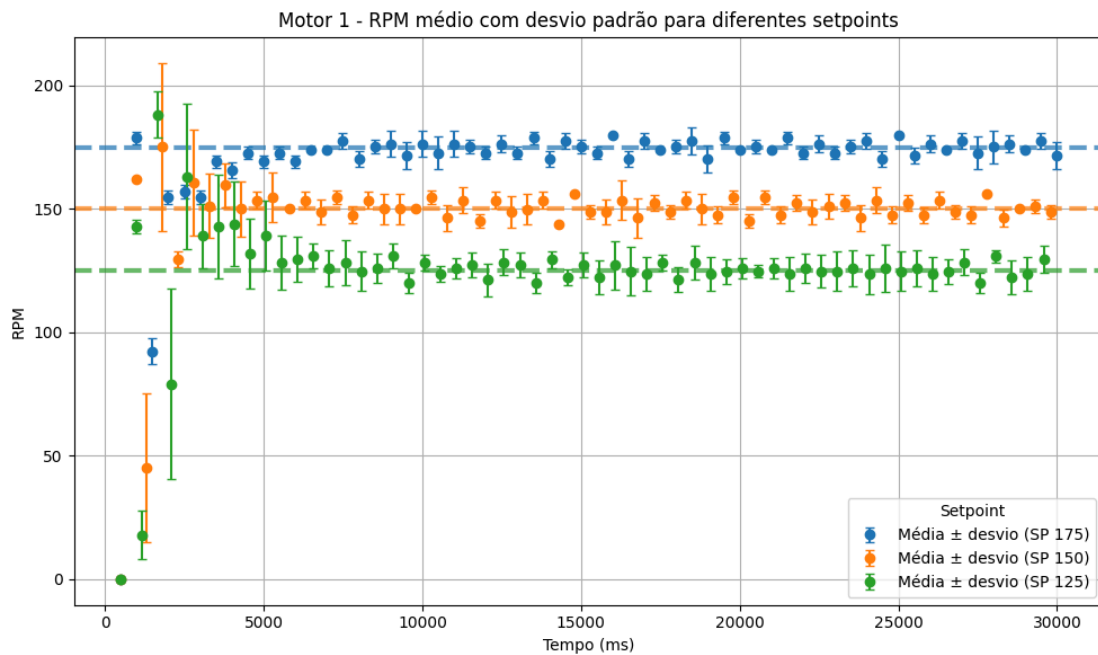
No Gráfico 4, pode-se observar a variação das velocidades obtidas ao decorrer do tempo para o Motor 1 nos *setpoints* de 50, 75 e 100 RPM, onde pode-se observar de forma clara a má adequação dos ganhos projetados, onde o erro no regime atingiu 113,2%, 59,6% e 32,9% do *setpoint* respectivamente, enquanto o *overshoot* máximo atingido para os pontos foram de 356%, 204% e 128%. Os resultados indicam que o controlador projetado não é capaz de se estabilizar em baixas faixas de operação para os ganhos ajustados.

Gráfico 4 - Verificação do controlador Motor 1 velocidades (50, 75, 100)



Fonte: Autoral.

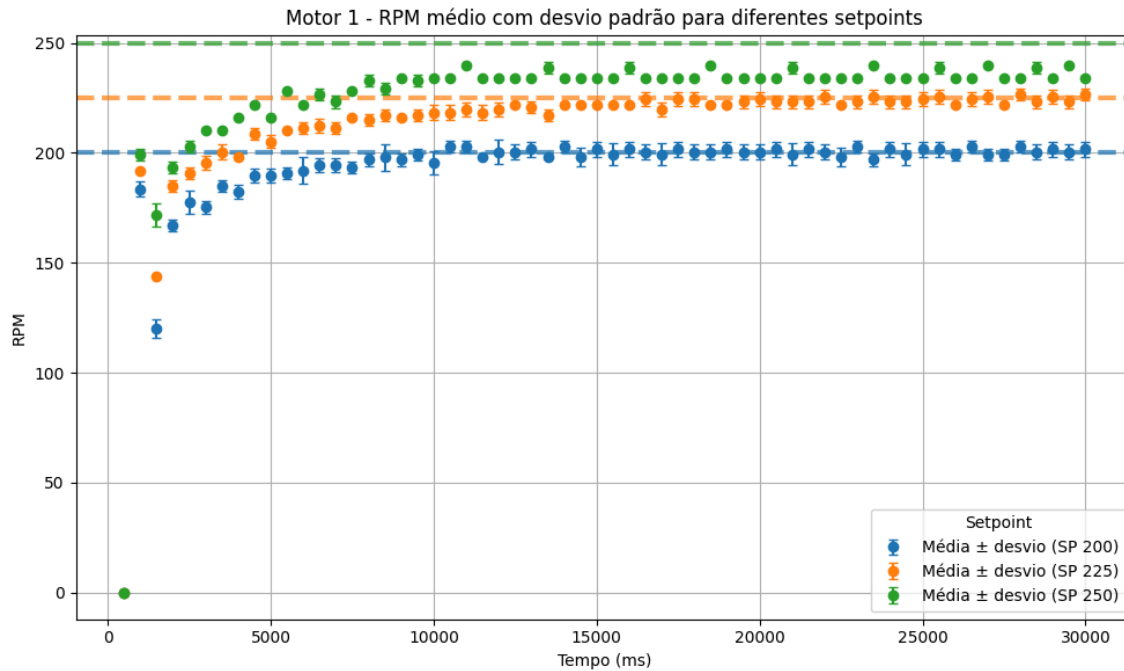
Gráfico 5 - Verificação do controlador Motor 1 velocidades (125, 150, 175)



Fonte: Autoral.

No Gráfico 5, pode-se observar que o Motor 1 apresentou um comportamento mais satisfatório no regime estável, apresentando 8% de variação, atingindo o objetivo de 5% de variação nos *setpoints* 150 e 175 RPM, onde esse desvio foi de 4,1% e 2,89% do *setpoint* respectivamente, apresentando uma diminuição gradual do *overshoot* que vai de 72% em 125 RPM para 2,86% aos 175, uma melhora significativa que indica a estabilização do sistema de acordo com as metas estipuladas.

Gráfico 6 - Verificação do controlador Motor 1 velocidades (200, 225, 250)

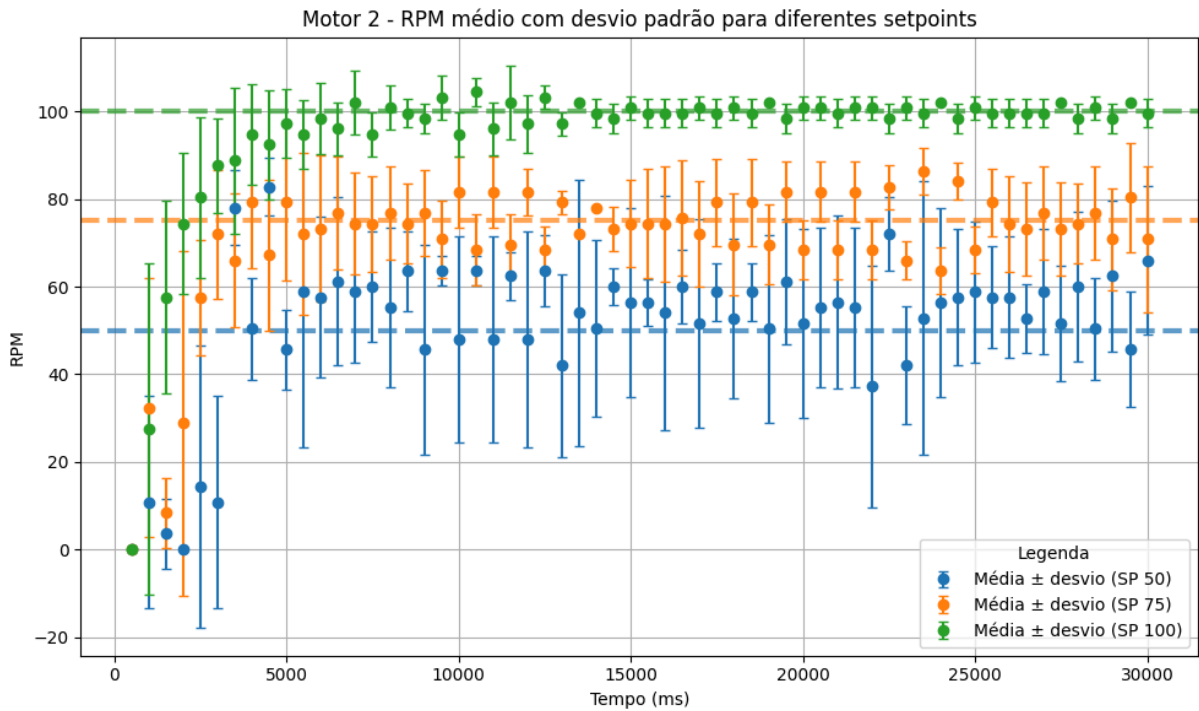


Fonte: Autoral.

O Gráfico 6, representa um resultado igualmente satisfatório para o comportamento do controlador, abordando os *setpoints* de velocidade mais elevados, 200, 225 e 250 RPM, apresenta desvios no regime estável 2% 1.38% e 2.73% do *setpoint*, com um *overshoot* de 4%, 3% e -4%, respectivamente. Ambos os critérios adotados foram atendidos nas medidas analisadas, com a diferença de que para o motor não conseguiu atingir a velocidade do último *setpoint* em 250 RPM, demonstrando uma estabilidade, porém numa região abaixo da determinada.

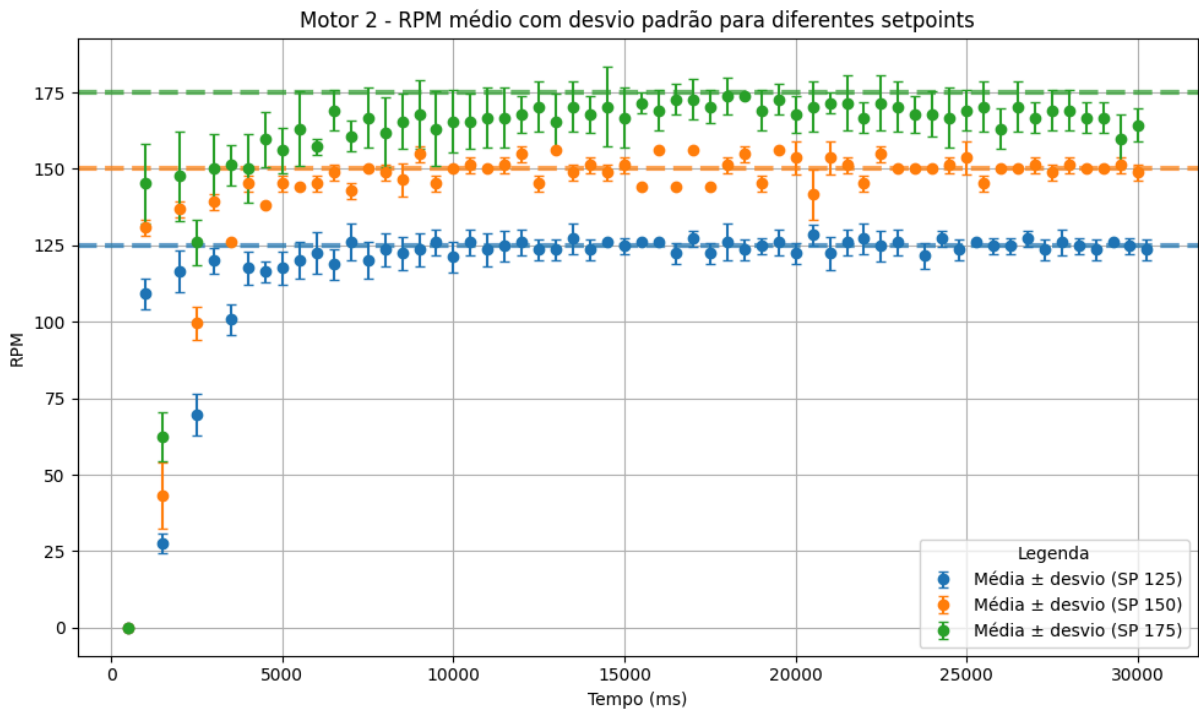
O gráfico 7 exibe os testes do controlador sendo realizados nos *setpoints* de 50, 75 e 100 RPM do Motor 2, apresentando valores de 18,8%, 10,65% e 5,66% com relação ao desvios em regime estável, enquanto o *overshoot* de 92% 36% e 14%, apresentando também uma diminuição gradual. É possível notar um certo grau de instabilidade na faixa selecionada, porém, é evidente que essa instabilidade, mesmo que fora dos critérios propostos, apresenta valores mais brandos do que os encontrados no Motor 1.

Gráfico 7 - Verificação do controlador Motor 2 velocidades (50, 75, 100)



Fonte: Autoral.

Gráfico 8 - Verificação do controlador Motor 2 velocidades (125, 150, 175)

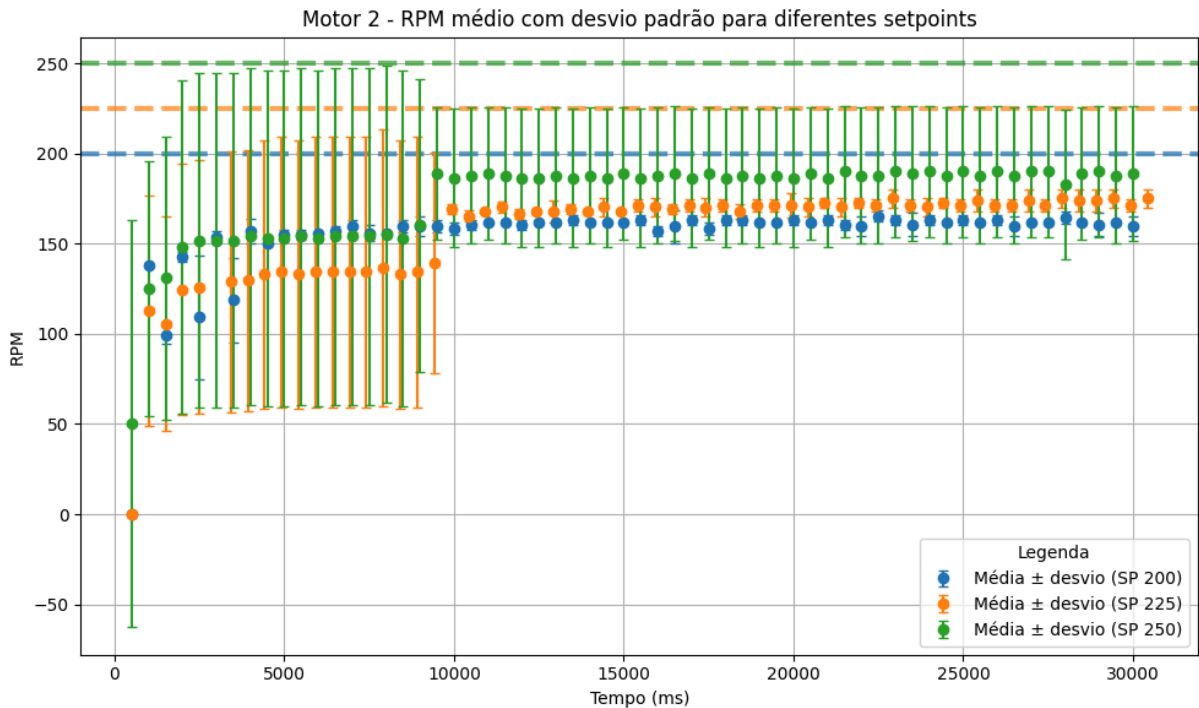


Fonte: Autoral.

A Gráfico 8, apresenta os *setpoints* de 125, 150 e 175 RPM para o motor 2, gerando um desvio regime estável de 4,49%, 4,39% e 3,85% respectivamente, com *overshoot* de 5,6%,

4% e 2,86, diminuindo a variabilidade na região de testes em questão, porém, atendendo ao critério de 5% de variabilidade apenas nos *setpoints* de 125 e 150 RPM, começando a apresentar maior desvio no *setpoint* de 175 RPM.

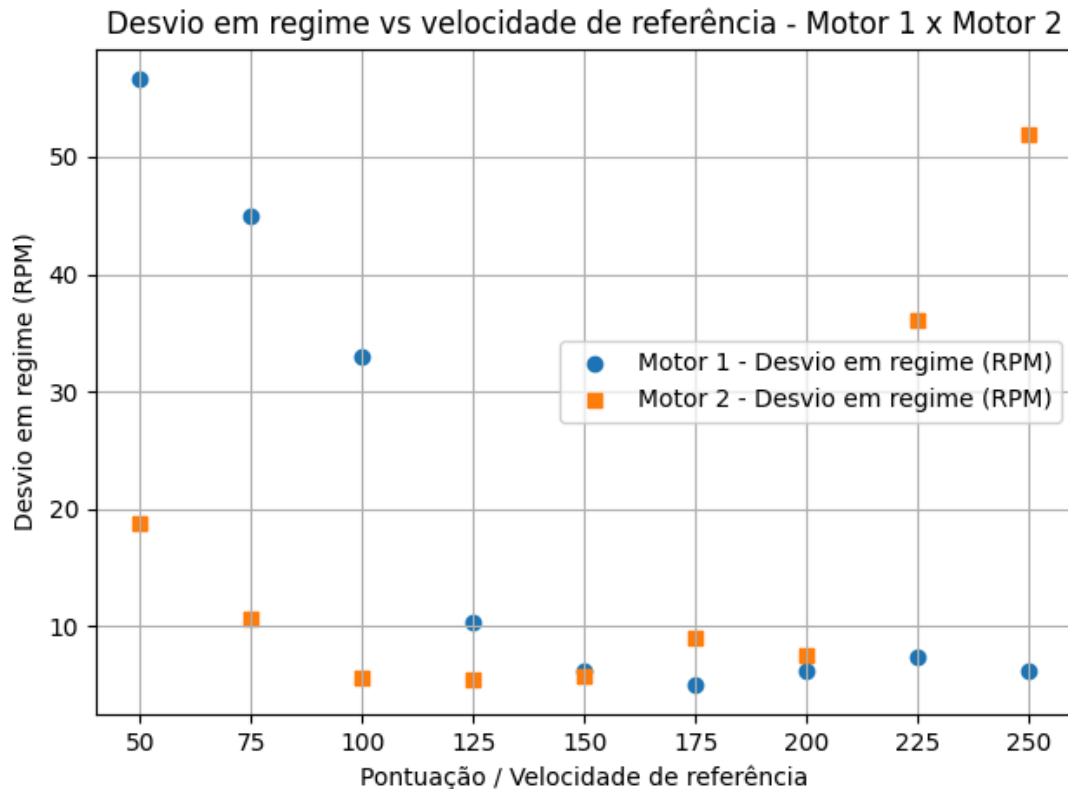
Gráfico 9 - Verificação do controlador Motor 2 velocidades (200, 225, 250)



Fonte: Autoral.

No Gráfico 9, é possível observar que nenhum dos gráficos apresentou uma distribuição das medidas próximas ao *setpoint* determinado, gerando um aumento nos desvios de acordo com o aumento dos *setpoints*, esse, se mantendo em 20,78% para as velocidades de 225 e 250 RPM, não sendo capaz de atingir a tensão determinada e não atendendo aos critérios de avaliação definidos.

Gráfico 10 - Gráfico dos erros absolutos referentes ao desvio em regime

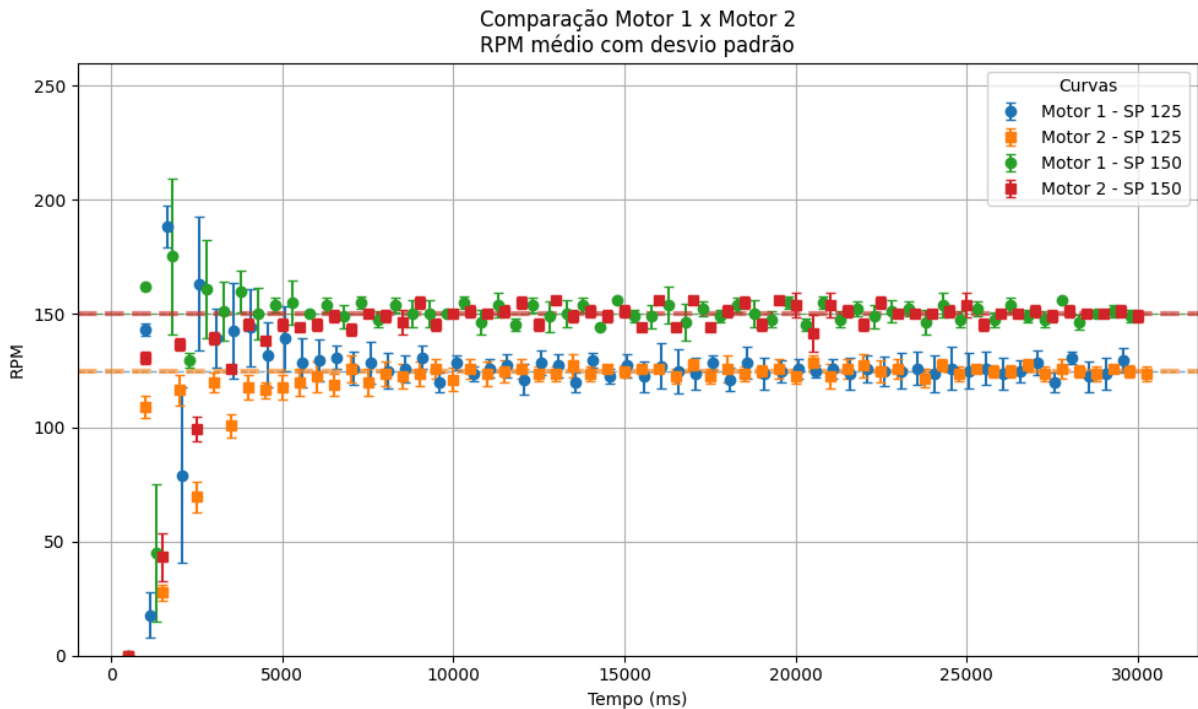


Fonte: Autoral.

No gráfico 10, pode-se observar a comparação dos erros absolutos referentes ao desvio em regime dos diferentes motores, nos respectivos setpoints de velocidade. Esse gráfico evidencia de uma forma mais simples a diferença no comportamento do desvio nos dois motores, pode-se observar que o primeiro motor tem um desvio muito maior quando observado em pequenas velocidades, quando comparado ao motor 2, esse entretanto, começa a apresentar um desvio maior quando analisado em maiores velocidades.

Os testes dos *setpoints* que obtiveram os resultados mais aproximados e que satisfizeram o critério de desvio adotado, assim como em sua maioria os testes referentes a variação do *overshoot* foram as velocidades de 150 e 175 RPM. Devido a isso, essas duas velocidades foram utilizadas para a comparação visual no gráfico 11, onde há a exibição do resultado referente aos dois motores.

Gráfico 11- Comparação dos nas velocidades 125 e 150



Fonte: Autoral.

Pode-se verificar, visualmente no gráfico 11, que os dados possuem uma variação muito semelhante, além de que no *setpoint* 150 possuem valores de variação no regime estável compatíveis com os critérios adotados, assim como o *overshoot*. Observou-se com os testes, os parâmetros gerados pelos gráficos característicos dos motores foram determinantes para a determinação de sua dinâmica com relação a diferentes *setpoints*, já que o motor que apresentou maior consumo de corrente se adaptou melhor à *setpoints* mais baixos, sendo que o contrário aconteceu com aquele que apresentou menor consumo.

6. CONCLUSÃO

O desenvolvimento do odômetro foi bem sucedido, possibilitando que demais testes fossem feitos para a verificação dos parâmetros relacionados aos motores utilizados e à verificação do próprio controlador dimensionado.

Os testes de velocidade gerada e tensão consumida pelos motores geraram resultados de extrema importância para o entendimento da dinâmica entre esses dois motores, justificando a utilização e elaboração de um controlador para tal. Tais testes, possibilitaram o levantamento de propriedades que modificam o comportamento do motor em determinadas velocidades e de que forma elas podem interferir no desempenho desses.

O método de Ziegler-Nichols apresentou resultados não satisfatórios, já que o regime transiente não pôde ser averiguado de forma apropriada, tanto pela velocidade em que o sistema se estabiliza, quanto pelas limitações na Instrumentação utilizada.

O método por tentativa e erro, se mostrou eficaz para a determinação dos ganhos do PID, através da estimação individual desses ganhos, ou mesmo considerando a variação entre esses observada pelos testes de curva característica.

O desenvolvimento do controlador PID de velocidade para os motores CC se mostrou eficaz, podendo ser avaliado através de critérios previamente definidos quanto a variabilidade desse em diferentes *setpoints* de velocidade. Entretanto observou-se que a eficácia deste depende do valor do *setpoint*, sendo mais satisfatória em certas faixas de operação devido a particularidade de cada motor, apresentando maior variabilidade nos extremos das faixas de tensão aplicadas.

7. PERSPECTIVAS FUTURAS

O controlador desenvolvido no trabalho é de extrema importância para a continuação dos trabalhos referentes à controladores desenvolvidos em ambiente acadêmico e para demais projetos desenvolvidos no mesmo laboratório, já que os testes realizados podem ser usados como bases para outros tipos de sistemas.

O trabalho em questão servirá como base para a implementação posterior de um controlador cinemático completo, controlando tanto a velocidade como posição do robô diferencial, a implementação deste depende de um controle preciso da velocidade dos motores de forma individual, e necessita de um bom desenvolvimento desses testes iniciais para uma eficaz estruturação inicial.

Para trabalhos futuros, tem-se como foco, a implementação dos controladores desenvolvidos sob o efeito de peso, com o robô em contato direto com o chão, para que possam ser feitos testes mais alinhados com a realidade e verifique-se a teoria obtida a partir das experimentações em bancada e então desenvolvimento de um controlador cinemático.

Um controlador cinemático seria de grande utilidade quando integrado ao ensino de física no ambiente universitário e mesmo quanto a práticas realizadas no ensino médio. Podendo ser formulada como um experimento mais básico até algo devidamente complexo, a determinação de velocidades e predição de posições futuras a partir dessas é um exercício que pode atingir vários níveis de complexidade e necessita de conhecimentos específicos a

respeito da dinâmica cinemática do robô, o que pode tornar as aulas que envolvam cinemática muito mais interessantes e práticas para o aluno, conectando o aluno com o ensino superior.

REFERÊNCIAS

- ALIEXPRESS. *Item 1005006053120735*. Disponível em: <https://pt.aliexpress.com/item/1005006053120735.html>. Acesso em: 3 dez. 2025.
- ALIEXPRESS. *Item 1005006209160046*. Disponível em: <https://pt.aliexpress.com/item/1005006209160046.html>. Acesso em: 3 dez. 2025.
- ALIEXPRESS. *Item 1005006563330148*. Disponível em: <https://pt.aliexpress.com/item/1005006563330148.html>. Acesso em: 3 dez. 2025.
- ALIEXPRESS. *Item 1005006585004406*. Disponível em: <https://pt.aliexpress.com/item/1005006585004406.html>. Acesso em: 3 dez. 2025.
- ALIEXPRESS. *Item 1005007425745438*. Disponível em: <https://pt.aliexpress.com/item/1005007425745438.html>. Acesso em: 3 dez. 2025.
- ALIEXPRESS. *Item 1005008318848935*. Disponível em: <https://pt.aliexpress.com/item/1005008318848935.html>. Acesso em: 3 dez. 2025.
- ALVES, Max Matias. Implementação de sistema pi para controle de temperatura utilizando o método da frequência de ziegler-nichols. Trabalho de Conclusão de Curso. Brasil. 2023. Disponível em: <https://repository.ufrpe.br/handle/123456789/4613>. Acesso em: 03 dez 2025.
- da Silva, R. O., Araujo, W. M., & Cavalcante, M. M. (2019). Visão Geral Sobre Microcontroladores e Prototipagem com Arduino. *TECNOLOGIAS EM PROJEÇÃO*, 10(1), 36–46. Recuperado de <https://projecaociencia.com.br/index.php/Projecao4/article/view/1357>
- BENITTI, Fabiane Barreto Vavassori. **Exploring the educational potential of robotics in schools: A systematic review**. *Computers & Education*, v. 58, n. 3, p. 978-988, 2012. DOI: 10.1016/j.compedu.2011.10.006.
- BOLDEA, I.; NASAR, S. A. *Electric drives*. 3. ed. Boca Raton: CRC Press, 2017.
- BOLTON, W. **Mecatrônica: sistemas de controle em engenharia**. 6. ed. São Paulo: Pearson, 2016.
- BORÉNSTEIN, J.; EVERETT, H. R.; FENG, L. **Where am I? Sensors and methods for mobile robot positioning**. Ann Arbor: University of Michigan, 1996.
- CAMPION, G.; BASTIN, G.; DANDREA-NOVEL, B. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, v. 12, n. 1, p. 47–62, 1996.

CITISYSTEMS. **Como selecionar um controlador de temperatura?** Disponível em: <https://www.citisystems.com.br/controlador-de-temperatura/>. Acesso em: 03 dez. 2025.

COSTA, Cesar. **Modelagem de um motor de corrente contínua.** [Apostila]. [S.l.: s.n.], [s.d.];2010. Disponível em: https://professorcesarcosta.com.br/upload/imagens_upload/Modelagem%20de%20um%20motor%20de%20corrente%20continua.pdf. Acesso em: 03 dez. 2025.

ELETROGATE. *Disco encoder para sensor de velocidade.* Disponível em: <https://www.eletrogate.com/disco-encoder-para-sensor-de-velocidade>. Acesso em: 3 dez. 2025.

FITZGERALD, A. E.; KINGSLEY, C.; UMANS, S. D. *Máquinas elétricas: com introdução à eletromecânica.* 7. ed. Porto Alegre: AMGH, 2014.

GROOVER, M. P. **Automação industrial e sistemas de manufatura.** 3. ed. São Paulo: Pearson, 2015.

GEDDES, Mark. **Manual de Projetos do Arduino: 25 projetos práticos para começar.**São Paulo. Novatec Editora, 2017.

HANSEN, C.; HANSEN, V. *Petzi en voyage.* [s.l: s.n.], 2015.

PRADO, T. P. do. Tinkercad: ferramenta online e gratuita de simulação de circuitos elétricos. **Embarcados**, 2018. Disponível em: <<https://embarcados.com.br/tinkercad/>>. Acesso em: 10 jun. 2024.

PEREIRA, Alysso Aurélio de Azevedo; LOPES, Adriano Nogueira Drumond. **Implementação de um controlador PID para o rastreamento da velocidade angular de um motor CC.** Anais da Mostra Nacional de Robótica (MNR 2018). [S. l.: s. n.], 2018.

OGATA, Katsuhiko. *Modern Control Engineering.* 5. ed. Upper Saddle River: Prentice Hall, 2010.

OLIVEIRA, K. L. R. de; SILVA, M. A. de F. da; OLIVEIRA, M. G. de; SCARPATI, R.; BATTESTIN, V. **FORMAÇÃO ONLINE DE PROFESSORES EM ROBÓTICA EDUCACIONAL COM PRÁTICAS NO SIMULADOR TINKERCAD.** Revista Brasileira de Ensino de Ciências e Matemática, [S. l.], v. 5, n. especial, 2022.

O que é Encoder? Para que serve? Como escolher? Como interfacear? | HI Tecnologia. Disponível em: <<https://materiais.hitecnologia.com.br/blog/o-que-%C3%A9-encoder-para-que-serve-como-escolher-como-interfacear/>>.

RODRIGUES, Ronaldo Mohr. **Construção, modelagem e controle de um robô móvel de acionamento diferencial.** 2019. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná,2019.

SANTOS, M. R. dos; MENEGHETTI, R. C. G. **Aspectos da interdisciplinaridade em dissertações e teses que versam sobre a Robótica Educacional com alunos de escolas públicas de Educação Básica**. *Ciência & Educação* (Bauru), v. 30, 2024.

Scoerger. **What is the mathematical basis for the Ziegler-Nichols method for tuning a PID loop?** *Engineering Stack Exchange*, 27 nov. 2019. Disponível em: <https://engineering.stackexchange.com/questions/31969/what-is-the-mathematical-basis-for-the-ziegler-nichols-method-for-tuning-a-pid-l>. Acesso em: 03 dez. 2025.

SICILIANO, B. et al. **Robotics: modelling, planning and control**. London: Springer, 2016.

SIEGWART, R.; NOURBAKHSI, I. R.; SCARAMUZZA, D. **Introduction to autonomous mobile robots**. 2. ed. Cambridge: MIT Press, 2011.

SOUZA, Naélio Freires Roberto de. **Uso da robótica educacional para ensino de programação: robótica educacional**. 2020. 39 f. Trabalho de Conclusão de Curso (Graduação em Sistema de Informação)- Universidade Federal do Ceará, Campus de Quixadá, Quixadá, 2020.

STROSKI, P. N. **PID controller tuning: Ziegler-Nichols methods - Electrical e-Library**. Disponível em: <https://electricalibrary.com/en/2019/11/06/pid-controller-tuning-ziegler-nichols-methods/>. Acesso em: 3 dez. 2025.

TECHTUDO. **WAP: fabricante**. TechTudo, [s. l.], [s. d.]. Disponível em: <https://www.techtudo.com.br/tudo-sobre/wap/>(<https://www.techtudo.com.br/tudo-sobre/wap/>). Acesso em: 16 dez. 2025.

THINGIVERSE.COM. **IR speed sensor holder MH-sensor**. Disponível em: <https://www.thingiverse.com/thing:4114000>>. Acesso em: 3 dez. 2025.

TZAFESTAS, S.G. **Mobile Robot Control**. National Technical University of Athens Athens, Greece, 2013.

UNIVERSIDADE FEDERAL DE ALFENAS. Rede de capacitação e apoio em robótica educacional - RECARE. Controle de Ações de Extensão, Alfenas, [s. d.]. Disponível em: https://sistemas.unifal-mg.edu.br/app/caex/comum/paginas/acoesVigentes.php?proposta_id=8291. Acesso em: 10 dez. 2025.

ANEXO A - CÓDIGO PARA CARACTERIZAÇÃO DOS MOTORES

```

#define m11 7
#define m12 6
#define vel1 5
#define m21 8
#define m22 9
#define vel2 10

const int pinoSensor = 2;
const int pinoSensor2 = 3;
const unsigned int pulsosDisco = 20;

const uint8_t TAMANHO_BUFFER = 30;

double rpmLog1[TAMANHO_BUFFER];
double rpmLog2[TAMANHO_BUFFER];
uint8_t idx = 0;

volatile unsigned long pulsos1 = 0;
volatile unsigned long pulsos2 = 0;

unsigned long tAnterior = 0;
double rpm1 = 0.0, rpm2 = 0.0;

const int PWM_FIXO = 0; // ajuste conforme necessário

void contador1() { pulsos1++; }
void contador2() { pulsos2++; }

void setup() {
  Serial.begin(9600);

  pinMode(m11, OUTPUT);
  pinMode(m12, OUTPUT);
  pinMode(vel1, OUTPUT);
  pinMode(m21, OUTPUT);
  pinMode(m22, OUTPUT);
  pinMode(vel2, OUTPUT);

  digitalWrite(m11, HIGH);
  digitalWrite(m12, LOW);
  digitalWrite(m21, HIGH);
  digitalWrite(m22, LOW);

  analogWrite(vel1, PWM_FIXO);
  analogWrite(vel2, PWM_FIXO);

  pinMode(pinoSensor, INPUT);
  pinMode(pinoSensor2, INPUT);

```

```
attachInterrupt(digitalPinToInterrupt(pinoSensor), contador1, FALLING);
attachInterrupt(digitalPinToInterrupt(pinoSensor2), contador2, FALLING);
```

```
for (uint8_t i = 0; i < TAMANHO_BUFFER; i++) {
  rpmLog1[i] = 0.0;
  rpmLog2[i] = 0.0;
}
```

```
tAnterior = millis();
}
```

```
void loop() {
  unsigned long agora = millis();
  if (agora - tAnterior >= 2000) { // agora 2 segundos
    noInterrupts();
    unsigned long capt1 = pulsos1;
    unsigned long capt2 = pulsos2;
    pulsos1 = 0;
    pulsos2 = 0;
    unsigned long dt = agora - tAnterior;
    tAnterior = agora;
    interrupts();

    if (dt > 0 && pulsosDisco > 0) {
      rpm1 = (capt1 * 60000.0) / (pulsosDisco * (double)dt);
      rpm2 = (capt2 * 60000.0) / (pulsosDisco * (double)dt);
    } else {
      rpm1 = 0.0;
      rpm2 = 0.0;
    }

    if (idx < TAMANHO_BUFFER) {
      rpmLog1[idx] = rpm1;
      rpmLog2[idx] = rpm2;
      idx++;
    }

    // Quando o buffer estiver cheio (10 amostras)
    if (idx == TAMANHO_BUFFER) {
      Serial.println(F("\n==== Buffer preenchido: 10 amostras ===="));
      Serial.print(F("RPM1 ["));
      for (uint8_t i = 0; i < TAMANHO_BUFFER; i++) {
        Serial.print(rpmLog1[i], 0);
        if (i < TAMANHO_BUFFER - 1) Serial.print(F(", "));
      }
      Serial.println(F("]"));

      Serial.print(F("RPM2 ["));
      for (uint8_t i = 0; i < TAMANHO_BUFFER; i++) {
```

```

    Serial.print(rpmLog2[i], 0);
    if (i < TAMANHO_BUFFER - 1) Serial.print(F(" "));
  }
  Serial.println(F("]"));
  Serial.println(F("=====\n"));

  // Zera o buffer após impressão
  for (uint8_t i = 0; i < TAMANHO_BUFFER; i++) {
    rpmLog1[i] = 0.0;
    rpmLog2[i] = 0.0;
  }
  idx = 0; // Reseta o índice do buffer para começar novamente
}

// Imprime as amostras individualmente enquanto o buffer não estiver cheio
Serial.print(F("Amostra ")); Serial.print(idx);
Serial.print(F(" | RPM1: ")); Serial.print(rpm1, 0);
Serial.print(F(" RPM2: ")); Serial.println(rpm2, 0);
}
}

```

ANEXO B - CÓDIGO DO CONTROLADOR PID

```

#define MIN_PWM 0
#define MAX_PWM 255

// Ganhos PID
#define KP 0.49
#define KI 0.60
#define KD 0.11

// Intervalo entre amostras em ms
#define SAMPLE_INTERVAL_MS 500 // 500 ms

// Quantidade de amostras
#define NUM_SAMPLES 60

// PINAGEM L298
const int IN1 = 4; // direção motor
const int IN2 = 5; // direção motor
const int PWM1 = 10; // PWM motor

// Sensor infravermelho
const int pinoSensor2 = 3; // encoder motor 1 (interrupt 0)
volatile unsigned long pulsos1 = 0;

// Encoder
unsigned int pulsosDisco = 20; // numero de dentes no disco

// Setpoint de RPM
double setpoint1 = 120.0;

// Variáveis de RPM e PID
double rpm1 = 0.0;

double erro1 = 0.0;
double erroAnterior1 = 0.0;
double integral1 = 0.0;
double pwmOut1 = 0.0;

// Tempo
unsigned long timeold = 0;
unsigned long startTime = 0;

// Vetores de coleta
unsigned long timeSamples[NUM_SAMPLES];
float rpmSamples[NUM_SAMPLES];
float pwm1Samples[NUM_SAMPLES];

int sampleIndex = 0;
bool coletaFinalizada = false;

```

```

// Interrupção do encoder
void contador1() {
  pulsos1++;
}

// PID motor 1
void atualizaPIDMotor1(double dt) {
  erro1 = setpoint1 - rpm1;
  integral1 += erro1 * dt;

  // Anti windup simples
  if (integral1 > 1000) integral1 = 1000;
  if (integral1 < -1000) integral1 = -1000;

  double derivada = (erro1 - erroAnterior1) / dt;
  double saida = KP * erro1 + KI * integral1 + KD * derivada;

  pwmOut1 = saida;
  if (pwmOut1 > MAX_PWM) pwmOut1 = MAX_PWM;
  if (pwmOut1 < MIN_PWM) pwmOut1 = MIN_PWM;

  erroAnterior1 = erro1;
}

// Impressão formato Python
void imprimeVetoresPython() {
  Serial.println();
  Serial.print("Testes finalizados. Amostras coletadas: ");
  Serial.println(NUM_SAMPLES);
  Serial.println();

  // Tempo
  Serial.println("timeSamples_ms = np.array(");
  Serial.print("[");
  for (int i = 0; i < NUM_SAMPLES; i++) {
    Serial.print(timeSamples[i]);
    if (i < NUM_SAMPLES - 1) Serial.print(", ");
  }
  Serial.println("]");
  Serial.println();

  // RPM
  Serial.println("rpm1Samples = np.array(");
  Serial.print("[");
  for (int i = 0; i < NUM_SAMPLES; i++) {
    Serial.print(rpmSamples[i], 2);
    if (i < NUM_SAMPLES - 1) Serial.print(", ");
  }
  Serial.println("]");
}

```

```

Serial.println();

// PWM
Serial.println("pwm1Samples = np.array(");
Serial.print("[");
for (int i = 0; i < NUM_SAMPLES; i++) {
  Serial.print(pwm1Samples[i], 2);
  if (i < NUM_SAMPLES - 1) Serial.print(", ");
}
Serial.println("]");
Serial.println();
}

void setup() {
  Serial.begin(9600);

  // Ponte H
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(PWM1, OUTPUT);

  // Sensor
  pinMode(pinoSensor2, INPUT);
  attachInterrupt(digitalPinToInterrupt(pinoSensor2), contador1, FALLING);

  pulsos1 = 0;
  rpm1 = 0.0;

  timeold = millis();
  startTime = timeold;
  sampleIndex = 0;
  coletaFinalizada = false;

  // Sentido do motor
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);

  Serial.println("Sistema PID de velocidade - 1 motor - coleta para Python");
  Serial.print("Setpoint M1 (RPM): ");
  Serial.println(setpoint1);
  Serial.print("NUM_SAMPLES: ");
  Serial.println(NUM_SAMPLES);
}

void loop() {
  if (coletaFinalizada) {
    analogWrite(PWM1, 0);
    return;
  }
}

```

```

unsigned long agora = millis();

if (agora - timeold >= SAMPLE_INTERVAL_MS) {
    unsigned long tempoDecorrido = agora - timeold;
    timeold = agora;
    double dt = tempoDecorrido / 1000.0;

    // Captura pulsos
    noInterrupts();
    unsigned long pulsosCapt = pulsos1;
    pulsos1 = 0;
    interrupts();

    // Cálculo de RPM
    if (tempoDecorrido > 0) {
        rpm1 = (60.0 * 1000.0 / pulsosDisco) / (double)tempoDecorrido * (double)pulsosCapt;
    } else {
        rpm1 = 0.0;
    }

    // Atualiza PID
    atualizaPIDMotor1(dt);

    // Aplica PWM
    analogWrite(PWM1, (int)pwmOut1);

    // Armazena amostras
    if (sampleIndex < NUM_SAMPLES) {
        timeSamples[sampleIndex] = agora - startTime;
        rpmSamples[sampleIndex] = (float)rpm1;
        pwm1Samples[sampleIndex] = (float)pwmOut1;

        sampleIndex++;

        if (sampleIndex >= NUM_SAMPLES) {
            coletaFinalizada = true;
            analogWrite(PWM1, 0);
            imprimeVetoresPython();
        }
    }
}
}
}
}

```