

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE**  
**CENTRO DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECATRÔNICA**  
**MESTRADO EM ENGENHARIA MECATRÔNICA**

**RELATÓRIO DO PRIMEIRO PROJETO  
CONTROLES EM MALHAS ABERTA E FECHADA  
DE UM MOTOR DC COM ENCODER**

**TURMA: 01  
GRUPO 01**

FRANCISCO PAIVA DA SILVA NETO: 20231022815

JORGE ALISSON NASCIMENTO DOS SANTOS: 20231022780

LUIZ VITOR CLEMENTINO: 20241005792

SAMUEL VICTOR MACIEL DA SILVA: 20241005827

Natal-RN  
2024

FRANCISCO PAIVA DA SILVA NETO: 20231022815

JORGE ALISSON NASCIMENTO DOS SANTOS: 20231022780

LUIZ VITOR CLEMENTINO: 20241005792

SAMUEL VICTOR MACIEL DA SILVA: 20241005827

**RELATÓRIO DO PRIMEIRO PROJETO  
CONTROLES EM MALHAS ABERTA E FECHADA DE UM MOTOR DC  
COM ENCODER**

Primeiro relatório apresentado à disciplina de Controle Inteligente de Sistemas Mecatrônicos, correspondente à 1º avaliação do semestre 2024.1 do mestrado em Engenharia Mecatrônica da Universidade Federal do Rio Grande do Norte, sob orientação do **Prof. Fábio Meneghetti Ugulino de Araújo**.

Professor: Fábio Meneghetti Ugulino de Araújo.

Natal-RN  
2024

## RESUMO

O presente relatório visa apresentar a discussão e os resultados obtidos com base em um modelo real do controle de um motor DC utilizando Arduino. No experimento foi utilizado um esquema montado em protoboard, em que posteriormente foi analisado o seu comportamento com e sem um controlador PID. Inicialmente, o projeto foi dividido em duas partes, sendo a primeira para analisar a saída do sistema em malha aberta enviando apenas um sinal de controle para o motor, depois, foi implementado um controlador PID para verificar novamente os dados obtidos agora com o sistema realimentado. Com isso, foram observados os tempos necessários para os cálculos de velocidade e controle, bem como a estabilidade e precisão das medições nos dois casos, permitindo validar os dados teóricos por meio dos resultados coletados.

**Palavras-chave:** Arduino; Controle Malha Fechada; Controle Malha Aberta; Motor DC.

# Lista de Figuras

1	Vista de cima da placa arduino . . . . .	9
2	Condições de operação . . . . .	9
3	Pinout . . . . .	10
4	Construção do motor DC . . . . .	11
5	Repulsão do rotor com os imãs . . . . .	11
6	Modelos diferentes de motores . . . . .	12
7	Disco encoder absoluto . . . . .	13
8	Disco encoder incremental . . . . .	13
9	Circuito ponte H . . . . .	14
10	Potenciômetro rotativo . . . . .	15
11	Potenciômetro internamente . . . . .	15
12	Protoboard 830 pontos . . . . .	18
13	Arduino Mega 2560 R3 . . . . .	18
14	Módulo ponte H L298N . . . . .	19
15	Motor DC com encoder acoplado . . . . .	19
16	Circuito malha aberta . . . . .	20
17	Circuito malha fechada . . . . .	21
18	Saída do sistema de controle . . . . .	25
19	Atuação das ações de controle . . . . .	26
20	Saída do sistema do controlador P (Proporcional) . . . . .	27
21	Sinais de controle para o controlador P (Proporcional) . . . . .	28
22	Saída do sistema do controlador PD (Proporcional-Derivativo) . . . . .	29
23	Sinal de controle do controlador PD (Proporcional-Derivativo) . . . . .	30
24	Saída do sistema de controle PID - dados 1 . . . . .	31
25	Saída do sistema de controle PID - dados 2 . . . . .	32
26	Controlador PID - conjunto de dados 1 . . . . .	33
27	Controlador PID - conjunto de dados 2 . . . . .	34
28	Simulação períodos de amostragem diferentes . . . . .	35
29	Sinais de controle para períodos diferentes . . . . .	35
30	Controlador PID - Filtros . . . . .	36

# Sumário

<b>1 INTRODUÇÃO</b>	<b>7</b>
<b>2 REFERENCIAL TEÓRICO</b>	<b>8</b>
2.1 Arduino . . . . .	8
2.1.1 Principais características do Arduino Mega 2560 . . . . .	8
2.1.2 Condições de operação . . . . .	9
2.1.3 Pinout . . . . .	10
2.2 Motor DC . . . . .	10
2.2.1 Funcionamento . . . . .	10
2.2.2 Especificações principais . . . . .	12
2.3 Encoder . . . . .	12
2.3.1 Tipo absoluto . . . . .	12
2.3.2 Tipo incremental . . . . .	13
2.3.3 Tecnologia óptica vs magnética . . . . .	14
2.4 Ponte H . . . . .	14
2.5 Potenciômetro . . . . .	15
2.6 Filtros . . . . .	16
<b>3 METODOLOGIA</b>	<b>17</b>
3.1 Ferramentas Utilizadas . . . . .	17
3.1.1 Protoboard 830 furos . . . . .	17
3.1.2 Arduino Mega 2560 R3 . . . . .	17
3.1.3 Ponte H Dupla L298N . . . . .	18
3.1.4 Motor DC 6V com Encoder . . . . .	18
3.2 Métodos e Procedimentos . . . . .	20
3.2.1 Análise do Problema . . . . .	20
3.2.2 Montagem do circuito . . . . .	20
3.2.3 Coleta dos dados . . . . .	22
3.2.4 Tratamento dos dados . . . . .	22
3.3 Desenvolvimento dos roteiros . . . . .	22
3.4 Filtros . . . . .	22
<b>4 RESULTADOS</b>	<b>22</b>
4.1 Controle em Malha Aberta . . . . .	23
4.1.1 Limites Operacionais do Motor (RPM mínimo e máximo) . . . . .	23
4.1.2 Zona Morta (Valor de PWM necessário para mover o motor) . . . . .	23
4.1.3 Tempos Necessários para a Realização dos Cálculos e Exibição dos Dados .	23
4.1.4 Validação da Resolução do Encoder . . . . .	24
4.1.5 "Estabilidade" das medições de posição e velocidade . . . . .	24
4.2 Controle em Malha Fechada . . . . .	25

4.2.1	Teste de Diferentes Valores de Ganho . . . . .	25
4.2.2	Estratégias de Controle P, PD, PID com e sem Filtro Derivativo e Anti-Windup	27
4.2.3	Diferentes Períodos de Amostragem . . . . .	30
4.2.4	Filtro Passa baixa e Anti-WindUp . . . . .	31
<b>5</b>	<b>CONCLUSÃO</b>	<b>32</b>
<b>6</b>	<b>REFERÊNCIAS</b>	<b>33</b>

# 1 INTRODUÇÃO

Um sistema de controle é o conjunto de dispositivos ou programas (algoritmos) utilizados para regular ou coordenar o comportamento de sistemas dinâmicos, assegurando que suas saídas se mantenham dentro de limites desejados, mesmo quando enfrentam interferências externas, incertezas nos modelos do sistema e flutuações nas condições de operação.

Existem dois tipos principais de sistemas de controle: em malha aberta e malha fechada. Para o primeiro citado, não há mecanismo de realimentação para ajustar o comportamento do sistema com base em suas saídas, ou seja, as ações de controle são determinadas apenas por meio da leitura de suas entradas. Por outro lado, em malha fechada, há um processo contínuo de comparação entre as saídas do sistema e um conjunto de referência, permitindo ajustes em tempo real através de um mecanismo de realimentação.

Partindo disso, nota-se que para alcançar um resultado específico, cada problema tem que ser classificado conforme a sua criticidade, isto é, as ações de controle e projeto de controladores são tomadas somente a partir da compreensão da influência de cada variável do sistema. Com base nisso, surge a aplicação dos controladores responsáveis pela leitura e ajuste dos dados desejados, os quais podem ser classificados como: Tipo P (Proporcional), Tipo PI (Proporcional-Integral), Tipo PD (Proporcional-Derivativo) e Tipo PID (Proporcional-Integral-Derivativo).

O controlador proporcional é o mais simples e age proporcionalmente ao erro atual do sistema, calculando o sinal de controle multiplicando o erro pelo ganho proporcional ( $K_p$ ). Ele pode levar a um erro de estado estacionário permanente em sistemas com erro de offset. Já o controlador PI considera a integral do erro ao longo do tempo para corrigir o erro acumulado. Útil para eliminar o erro de estado estacionário, mas pode causar oscilações e instabilidade se manipulado incorretamente. O controlador PD, por sua vez, considera a taxa de variação do erro com o tempo, antecipando mudanças na saída do sistema para reduzir sobressinal e estabilizar o sistema. Pode aumentar a sensibilidade a ruídos no sinal de entrada. Por fim, o controlador PID combina os três tipos de controle - proporcional, integral e derivativo - para lidar com uma variedade de situações e fornecer um desempenho melhor. No entanto, ajustar seus parâmetros pode ser complexo devido à variedade de cálculos envolvidos.

Logo, tendo em vista esse contexto e sua importância, o presente relatório irá descrever a implementação de um controle PID de um motor DC com encoder, especificando cada um dos materiais, programas e elementos utilizados, assim como a análise dos resultados obtidos, como: limites operacionais do motor (RPM mínimo e máximo), zona morta (Valor de PWM necessário para mover o motor), tempos necessários para realização de cálculos e exibição de dados, estabilidade das medições de posição e velocidade e a validação da resolução do encoder.

## 2 REFERENCIAL TEÓRICO

### 2.1 Arduino

Dado que um dos objetivos do experimento é modificar o sistema de acordo com as especificações desejadas, é necessário a implementação de um controlador para a definição da variável manipulada, de forma que ela modifique a velocidade do motor DC. O controlador neste experimento é a plataforma Arduíno Mega 2560, uma placa microcontroladora fabricada pela Arduíno.

A Arduino Mega 2560 é uma placa sucessora do Arduino Mega, é dedicada a aplicações e projetos que requerem grande número de pinos de entrada e saída e casos de uso que precisam de alto poder de processamento. O Arduino Mega 2560 vem com um conjunto muito maior de IOs quando comparamos com a placa Uno tradicional, considerando o formato de ambas as placas [1].

Ao mesmo tempo, as operações e o processamento não ficam lentos devido à sua RAM significativamente maior do que a do outros processadores.

#### 2.1.1 Principais características do Arduino Mega 2560

- Processador ATmega2560;
  - Taxa de transferência de até 16 MIPS (milhões de instruções por segundo) a uma frequência de 16 MHz;
  - Contador em tempo real com oscilador separado
  - Quatro USARTs seriais programáveis
  - 8k bytes de SRAM
- ATmega16U2;
  - É um processador serial USB que atua como interface entre os sinais de entrada USB e o processador principal. Isso aumenta a flexibilidade de interface e conexão de periféricos da placa.[1]
- Conexão USB-B;
- Adaptador AC/DC externo;
- 54 pinos digitais;
- 16 entradas analógicas;
- 15 saídas PWM
- 4 USARTs
- Corrente máxima de 20 mA por pino
- Suporta bateria de 9V

A figura 1 ilustra a vista de cima da placa com seus componentes eletrônicos. Os principais componentes mencionados nesta seção estão marcados na cor azul, como o conector USB (código USB), o chip do processador ATmega2560 (código IC3), e o chip do processador serial USB ATmega16U2 (código IC4). Também há um regulador de tensão linear (IC1) do modelo NCP1117, que estabiliza a tensão de alimentação do Arduino Mega em 5V. O led verde (ON) indica que a placa está ligada, e todos os conectores dos pinos digitais e analógicos estão representados pelo código XIO.

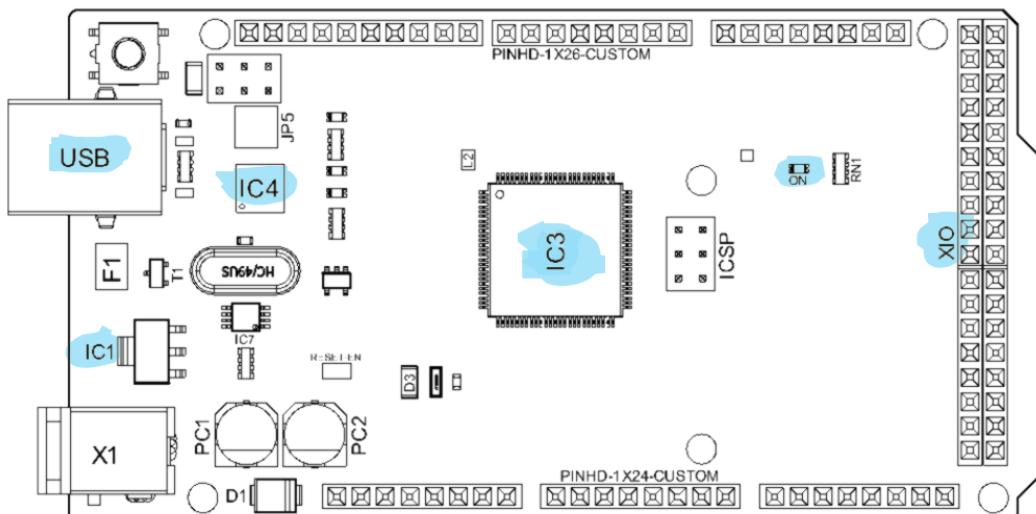


Figura 1: Vista de cima da placa arduino

### 2.1.2 Condições de operação

As condições de operação do Arduino são mostradas na figura 2.

Símbolo	Descrição	Min	Tip	Máx	Unidade
$V_{IN}$	Tensão de entrada do conector jack	7	7.0	12	V
$V_{USB}$	Tensão de entrada do USB	4.8	5.0	5.5	V
$T_{OP}$	Temperatura de operação	-40	25	85	°C

Figura 2: Condições de operação

### 2.1.3 Pinout

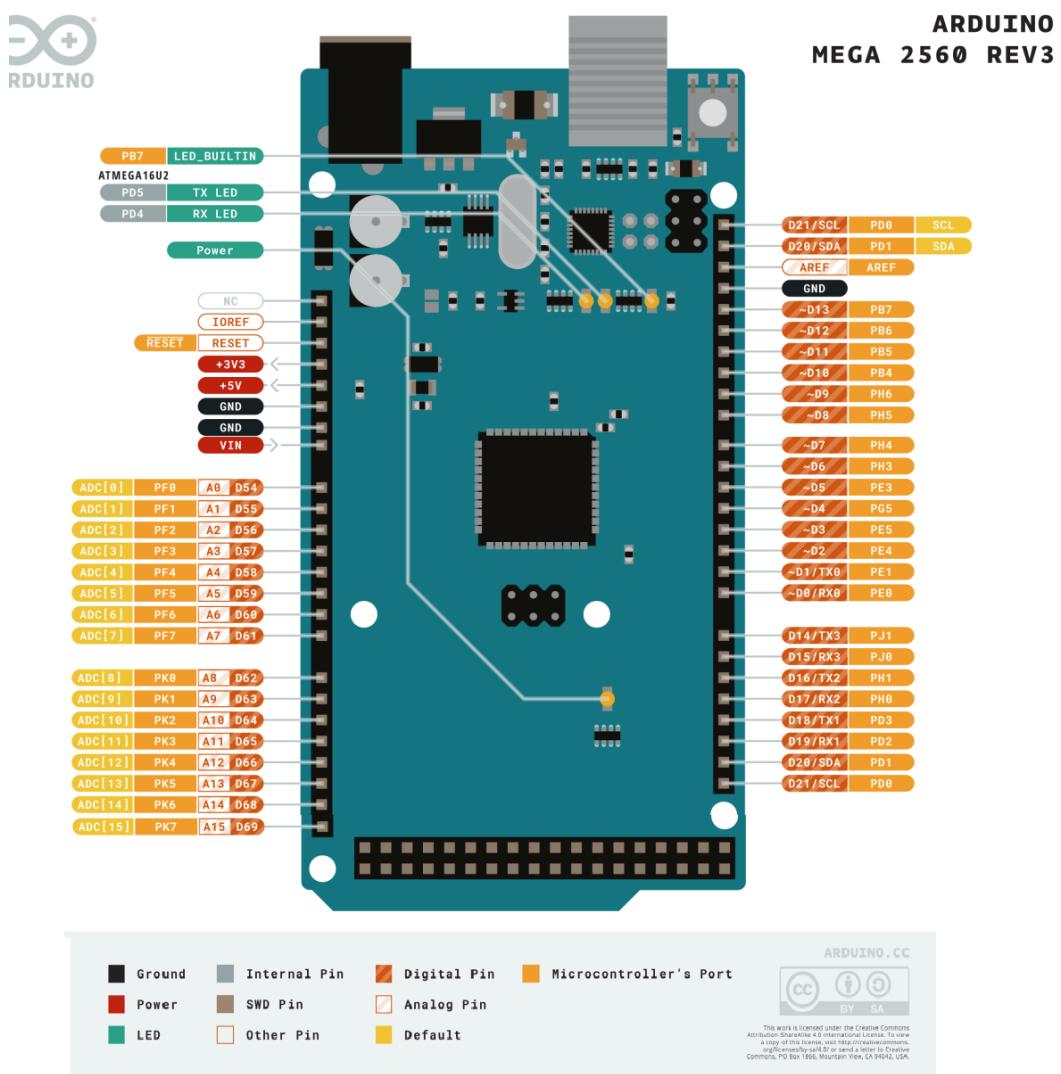


Figura 3: Pinout

## 2.2 Motor DC

### 2.2.1 Funcionamento

Motores de corrente continua (CC) ou motores DC (*Direct Current*) são dispositivos que operam com forças de atração e repulsão geradas por eletroimãs ou imãs permanentes, muito utilizados em aparelhos eletrônicos que possuem recursos mecânicos [2]. Sabe-se, segundo a teoria do eletromagnetismo, que ao percorrer uma corrente elétrica em um fio condutor será criado um campo magnético em volta desse fio, criando um eletroimã com polos norte (N) e sul (S), então se existirem dois fios com um campo magnético ao seu redor, ao aproximar-los podemos ter atração ou repulsão, devido aos polos.

Essa ideia é aplicada ao motor de corrente continua, sendo que entre dois imãs (ou eletroimãs), de modo que seus polos se atraiam (polos diferentes se atraem), tenha uma bobina (rotor),

conforme figura 4, de forma que a bobina do rotor ao ser percorrida por uma corrente elétrica gera dois polos que são atraídos ou repelidos pelos polos dos imãs. Quando temos polos iguais alinhados ocorre uma repulsão, fazendo com que o rotor gire para o outro lado, como mostra na figura 5.

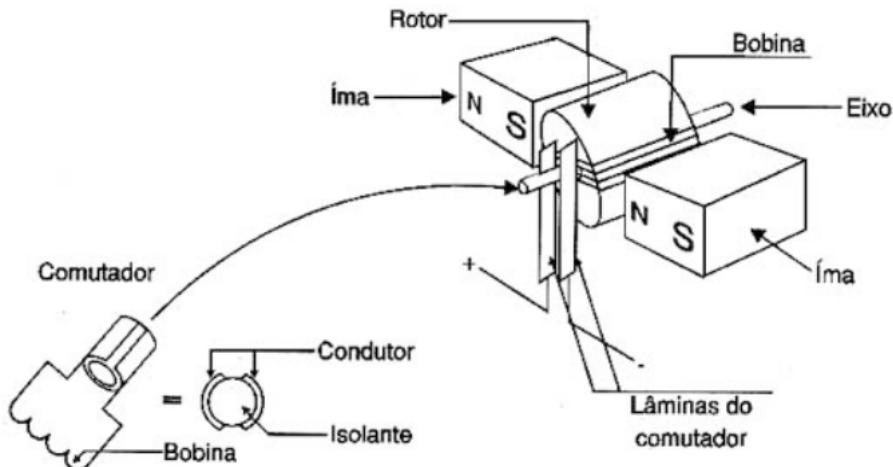


Figura 4: Construção do motor DC

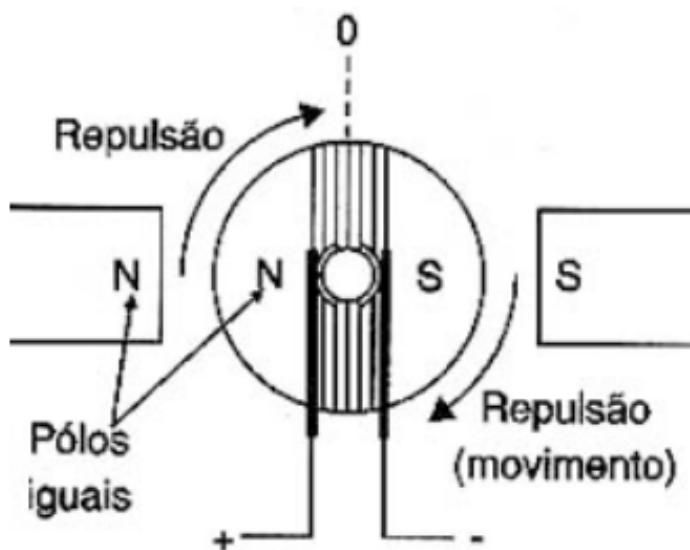


Figura 5: Repulsão do rotor com os imãs

Quando ocorre a repulsão do rotor com os imãs, o movimento também acontece, fazendo com que o polo norte do rotor chegue ao polo sul do imã e acabem de atraindo. Mas para continuar o movimento por conta da repulsão, há um comutador que inverte o sentido da corrente que percorre a bobina do rotor, fazendo seus polos se inverterem e consequentemente gerar a repulsão entre rotor e imãs novamente. Essa troca de sentido faz o movimento de rotação continuar e o motor não para enquanto o comutador fizer sua ação aproximadamente a cada meia volta do rotor. Pode-se interpretar o movimento do rotor devido a repulsão também com a lógica de atração dos polos.

## 2.2.2 Especificações principais

Para saber se um motor dc pode atender as necessidades de uma aplicação é fundamental avaliar três características principais:

**Velocidade do eixo** - O valor de tensão aplicado no motor gera uma velocidade proporcional, tipicamente dada em RPM (rotações por minuto). A velocidade de um motor informada por um fabricante se refere ao motor trabalhando de maneira ideal, quando não há torque.

**Torque de saída** - É a força de rotação gerada pelo eixo do motor, dado em unidades de força-distância (geralmente em N·m).

**Tensão disponível** - A variável responsável pela velocidade do motor, alguns motores funcionam em um valor de tensão fixa.

A figura 6 mostra alguns tipos diferentes de motores DC.



Figura 6: Modelos diferentes de motores

## 2.3 Encoder

Um encoder é um sensor eletrônico que tem como função principal converter movimento linear ou rotativo mecânico em pulsos elétricos, de forma que possibilita o controle de posição, velocidade e direção de elementos rotativos através da leitura fornecida. O encoder trabalha gerando sinais de saída digital para cada posição em tempo real, podendo fornecer essa leitura para vários dispositivos, como CLPs, inversores de frequência, etc.

### 2.3.1 Tipo absoluto

Determina a posição com base em um ponto de referência estático, possuindo dois discos alinhados ao eixo central, onde um é fixo ao eixo e o outro tem movimentação livre. Os dois contêm anéis

concêntricos com marcadores de compensação. Conforme o disco vai girando, esses marcadores mudam de posição no disco fixo, fornecendo sinais de saída digitais – words ou bits exclusivos para cada posição.

A figura 7 ilustra como são os discos de um encoder absoluto, no qual cada anel concentrico de diametros diferentes tem marcadores de comprimentos diferentes, esses aneis também são chamados de trilha, e cada trilha representa um bit. Quanto mais bits, mais preciso é o sensor com relação a posição.

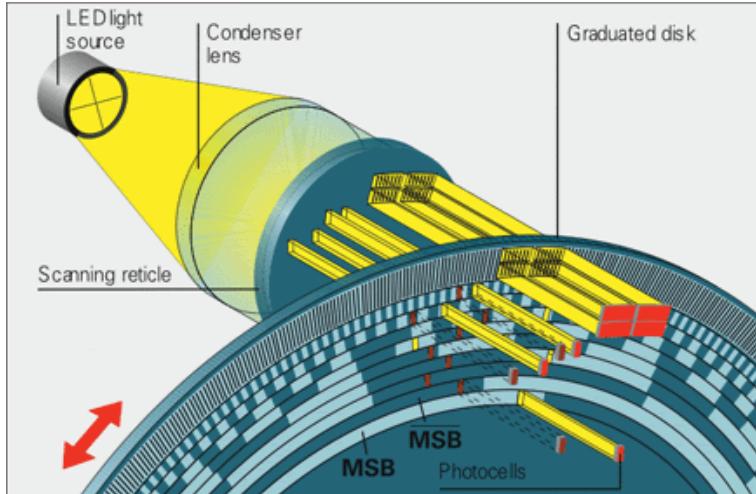


Figura 7: Disco encoder absoluto

### 2.3.2 Tipo incremental

O encoder incremental dá o número de pulsos a partir de cada rotação inteira do eixo. Para encoder com tecnologia óptica, cada pulso é gerado pela recepção de um sinal luminoso que passa por orifícios igualmente espaçados de um disco para um receptor desse sinal do outro lado do disco. A construção do encoder incremental se dá pelo disco representado na figura 8, sendos os sinais emitidos por A e B que identificam o sentido de rotação.

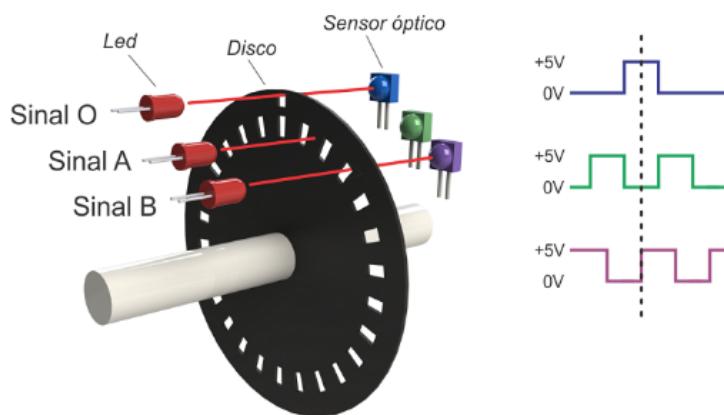


Figura 8: Disco encoder incremental

### 2.3.3 Tecnologia óptica vs magnética

A tecnologia óptica utilizada nos encoders contém leds que emitem a luz através dos orifícios dos discos e também um fotoreceptor do outro lado do disco e alinhado com os leds, fazendo a detecção da luz. Para detectar o sentido de rotação há dois canais emissores, o A e o B, como já mostrado na figura 8. Esses canais podem identificar nível lógico alto (detectando luz) ou nível lógico baixo (sem detecção da luz). Quando o canal A muda de estado e o canal B está com o mesmo estado, significa que o giro é no sentido horário, mas se o estado do canal A muda e o estado do canal B é o oposto significa que o giro é no sentido anti-horário.

A tecnologia magnética em vez de utilizar leds e receptores como nos encoders ópticos, utilizam imãs e sensores de efeito Hall. Os imãs são dispostos pelo disco em posições fixas, os sensores de efeito Hall ficam posicionados fora do disco, em posições fixas, enquanto o disco com imãs gira junto com o eixo de um motor por exemplo. Assim, a leitura da posição se dá pela detecção do imãs pelo sensor magnético.

## 2.4 Ponte H

O circuito conhecido pelo nome ponte H é um driver para motor DC que permite mudar seu sentido de rotação. O nome ponte H é devido a carga (motor) ficar entre 4 chaves ,que lembra a letra H, como está na figura 14. Este circuito também é usado para controlar também a tensão do motor.

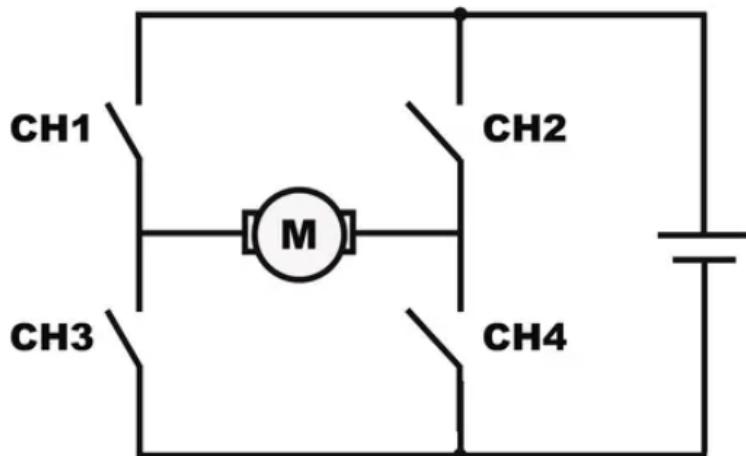


Figura 9: Circuito ponte H

A logica de mudança de sentido de rotação do motor DC se dá pelo fato de que ao inverter a polaridade de alimentação do motor, seu sentido de rotação é invertido. Logo, dado o circuito acima, quando somente as chaves CH1 e CH4 estiverem fechadas a corrente flui da esquerda para a direita, de acordo com a fonte de alimentação representada, isso faz o motor girar no sentido horário. Quando somente as chaves CH3 e CH2 estiverem fechadas acontece o oposto e o motor gira no sentido anti-horário. Essas chaves são na verdade transistores que trabalham em regiões de corte e saturação, funcionando como chave aberta ou fechada.

## 2.5 Potenciômetro

O potenciômetro é um tipo especial de resistor, podendo variar sua resistência para ter flexibilidade em várias aplicações como mixers de áudio, eletrodomésticos, televisores, equipamentos industriais, joysticks, etc. É chamado também de resistor variável e divisor de tensão variável. Existem vários tipos de potenciômetros, como do tipo slider, duplo, trimmers, multivoltas e o tipico rotativo linear que será abordado neste texto. A figura 10 mostra o potenciômetro rotativo e linear.

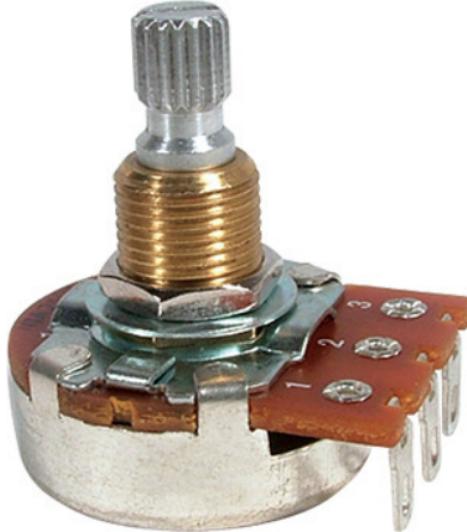


Figura 10: Potenciômetro rotativo

De acordo com a figura 11, o potenciômetro é composto por três terminais. Um par externo, composto pelos terminais "início" e "fim", está conectado aos lados opostos de um elemento resistivo interno, conhecido como trilha ou pista. O terceiro terminal, central, está internamente ligado a um contato chamado cursor. Esse cursor está em contato com a pista e pode ser movido de um extremo ao outro da pista. Isso pode ser feito através da rotação de um eixo. Quando aplicamos uma tensão elétrica entre os terminais de início e fim, a tensão elétrica obtida no terminal central irá variar conforme o cursor se move ao longo da pista

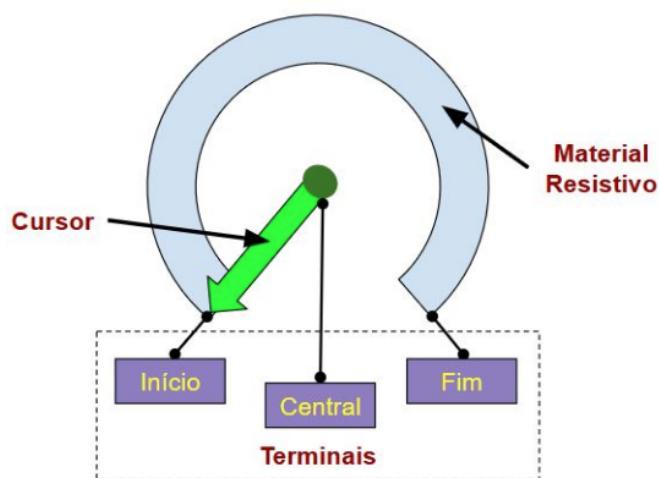


Figura 11: Potenciômetro internamente

## 2.6 Filtros

O fundamento do controlador PID é fazer uso de ação proporcionais ao erro entre o valor real do sistema e o valor desejado, assim como sua derivada e integral. Cada uma dessas parcelas (proporcional, integrativa e derivativa) contribui para modificar o sinal de controle de forma diferente. A parte proporcional é responsável pela rapidez que o sistema chega no valor desejado (setpoint). Quanto maior esse valor, maior a velocidade para no valor de setpoint, porem, um valor muito alto pode causar instabilidades. A parte integrativa visa fazer o sinal de controle atingir efetivamente o valor desejado, pois a parte proporcional sozinha não é capaz de atingir pois o valor diminui com a aproximação ao setpoint. Por fazer a integral do erro, o valor aumenta com o tempo até alcançar a permanência de regime. (GOMES, 2019). Porém, enquanto o sistema não alcançar o valor de regime, o valor da parcela integrativa irá continuar crescendo e provocando atrasos na estabilidade e podendo causar oscilações indesejadas. Isso ocorre devido ao valor máximo que um atuador consegue alcançar ou seja, se mantem num valor de saturação do sinal de controle, essa ação é chamada no inglês de Winds Up ou auto carregamento. (DA SILVA, 2000). A porção derivativa é responsável por deixar o sistema mais robusto. Tem o objetivo de atenuar a variação rápida transação do processo sendo capaz de tirar o sistema da inercia em caso de mudança brusca de setpoint. Porem, a parcela derivativa é muito sensível a ruídos no sistema, podendo adicionar picos ao sinal de controle. Esses picos podem causar desgastes aos atuadores e comprometer os mesmos. Para diminuir esses picos e atenuar essa oscilação, um filtro passa-baixa pode ser usado na parcela derivativa (GOMES, 2019).

### 3 METODOLOGIA

A partir de um modelo de simulação virtual de controle de motor DC desenvolvido no *Tinkercad*, as etapas para a análise do problema e desenvolvimento da solução foram da preparação da simulação e dos componentes necessários, a configuração do sistema de coleta de dados no que diz respeito as conexões em protoboard entre os componentes. Abaixo podem ser vistos os componentes utilizados na Tab.?? e a descrição de conexão ao longo da seção.

**Tabela 1: Lista de componentes utilizados**

Componente	Quantidade (UN)
Protoboard 830 furos	1
Arduino Mega 2560 R3	1
Ponte H Dupla L298N	1
Motor DC 6V com Encoder	1
Fonte Externa 12V 1A	1
Regulador de tensão L7806	1
Capacitor Eletrolítico 10uF/25v	1
Capacitor Cerâmico 100nF	1
Potenciômetro Rotativo Linear 10KΩ	1
Jumpers macho-macho	12
Jumpers macho-fêmea	3

#### 3.1 Ferramentas Utilizadas

##### 3.1.1 Protoboard 830 furos

A Figura.12 mostra a protoboard de 830 furos utilizada para realizar a conexão dos módulos da ponte H com o motor DC, o Arduino Mega e o potenciômetro de acordo com esquemáticos feitos no Tinkercad, além de alguns jumpers para conectar os componentes encaixados na placa.

##### 3.1.2 Arduino Mega 2560 R3

A Figura.13 Arduino Mega foi utilizado com a função de controlador do problema, realizando os cálculos das ações de controle, recebimento, tratamento e envio dos sinais de entrada e de saída para o motor, de modo que fosse possível utilizar o monitor serial para visualizar os dados que eram enviados a variável de processo representada pela velocidade do motor.

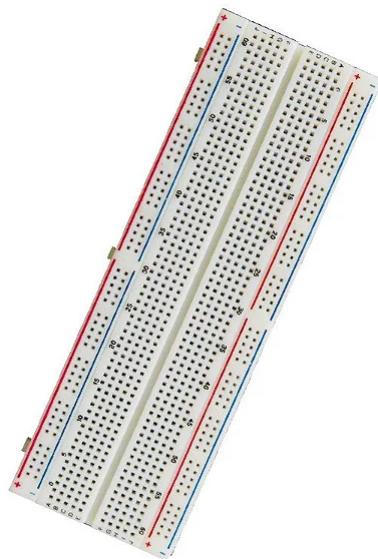


Figura 12: Protoboard 830 pontos

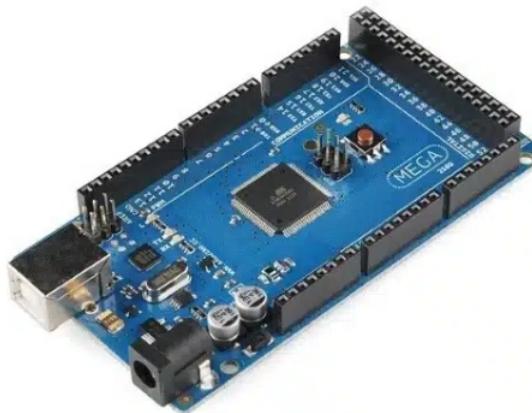


Figura 13: Arduino Mega 2560 R3

### 3.1.3 Ponte H Dupla L298N

A ponte H que pode ser vista na Fig.14 foi a responsável por fazer o acionamento do motor, dado que, o Arduino não possui potência suficiente nas portas digitais para acionar o motor, sendo assim, o acionamento foi realizado com uso da ponte H que vinha com um driver acoplado que era alimentado por uma fonte externa de 12 V que passava por um regulador de tensão, chegando no módulo em 6V, usados na alimentação do motor, e pelos 5V diretamente da porta do Arduino Mega que enviava apenas sinais digitais de baixa corrente para indicar a ação que deveria ser executada.

Cada canal da ponte suporta...

### 3.1.4 Motor DC 6V com Encoder

O motor DC com encoder da Fig.15 foi escolhido devido a sua estrutura já conter um encoder rotativo, dispensando circuitarias externas adicionais, aém de permitir uma leitura mais facilitada por meio de canais acoplados a sua armadura, fazendo com que o experimento pudesse ser realizado de forma facilitada, sendo alimentado por uma tensão de 6V que vinha da ponte H, enquanto que com

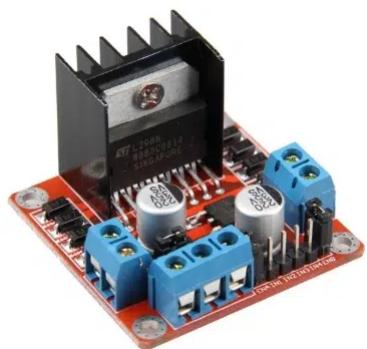


Figura 14: Módulo ponte H L298N

relação ao encoder, sua alimentação vinha direto da porta 3.3V do Arduino Mega, e os seus canais A e B iam direto para as portas 3 e 4 do Arduino, respectivamente, devido a necessidade de usar interrupção.



Figura 15: Motor DC com encoder acoplado

## 3.2 Métodos e Procedimentos

Para o correto desenvolvimento das atividades dispostas foi necessário compartmentalizar o problema em partes diferentes para uma melhor análise da situação.

### 3.2.1 Análise do Problema

Para que fosse possível realizar os experimentos, foram analisadas as atividades propostas a fim de que fosse possível determinar uma forma de começar a abordar o problema, organizando os modos como iriam ser feitas as simulações, como também o modo em que iriam ser inseridos e coletados os dados, bem como o modo em que iriam ser organizados posteriormente para tratamento e apresentação em relatório.

### 3.2.2 Montagem do circuito

A montagem do circuito foi feita com base no datasheet dos componentes, que inicialmente forneceu as ideias de funcionamento dos componentes, seguido de esquemáticos disponibilizados previamente nos materiais do professor, bem como em esquemáticos montados no software *Tinkercad*, que fornecia os componentes em um ambiente de simulação virtual, permitindo realizar testes iniciais antes da implementação real.

Abaixo podem ser vistos nas Fig.16 e Fig.17 os circuitos em malha aberta e malha fechada, respectivamente, montados no *Tinkercad*.

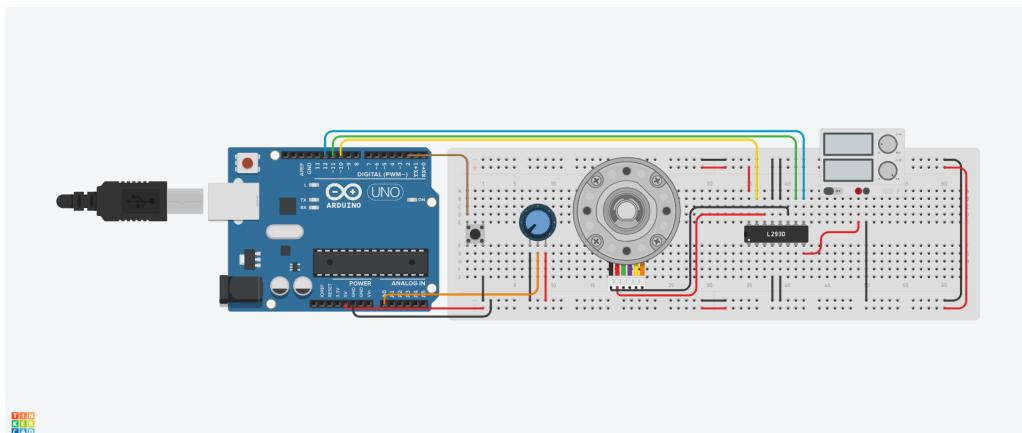


Figura 16: Circuito malha aberta

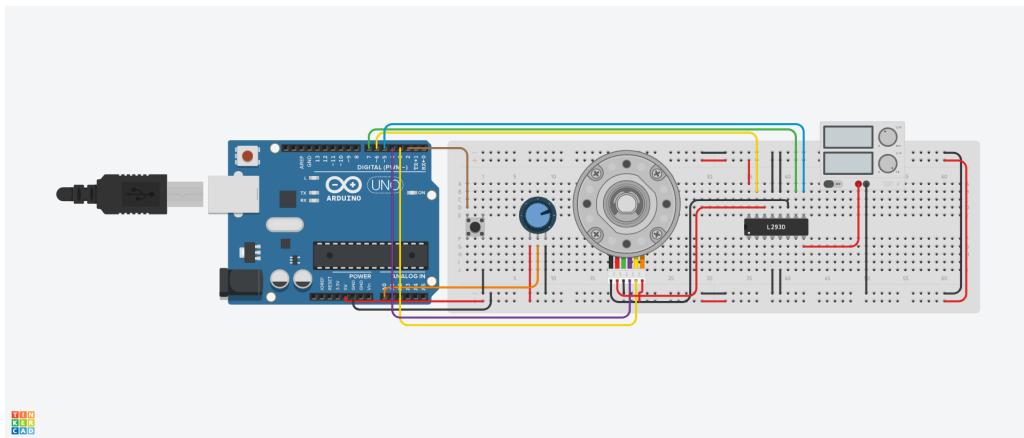


Figura 17: Circuito malha fechada

É importante ressaltar que devido a limitações do software, o Arduino utilizado foi o modelo UNO R3, contudo, na prática foi utilizado o modelo descrito anteriormente, de modo que a pinagem se manteve a mesma, como pode ser visto na Tab.2.

Tabela 2: **Potência no MPP dos painéis e do conversor boost**

Pino do componente	Pinagem no Arduino Mega
Canal A Encoder	3
Canal B Encoder	4
Enable Ponte H	5
Entrada 1 Ponte H	7
Entrada 2 Ponte H	6
Potenciômetro	A0

### 3.2.3 Coleta dos dados

A coleta dos dados foi realizada por meio do monitor serial da IDE do Arduino, em que os dados eram formatados em uma função responsável por sua exibição na tela, e posteriormente coletados em um arquivo csv que será usado para o tratamento dos resultados.

A função de exibição formatava os dados de acordo com a opção inserida como parâmetro inicial, que era repassada para um comando switch case no código que selecionava quais seriam os dados de entrada, e como seria o formato de saída.

VERIFRICAR SE COLOCAMOS EXEMPLO DE COMO OS DADOS SAO FORMATADOS

### 3.2.4 Tratamento dos dados

Com a finalização da coleta de dados era necessário desenvolver os mesmos. Para este fim foram utilizado algoritmos em python a fim de se desenvolver médias, bancos de dados, gráficos, e qualquer outro tratamento de dados que por ventura foi necessário. O uso do script em python foi escolhido devido a facilidade de trabalhar com grande quantidade de dados em bibliotecas já prontas e disponíveis, além da facilidade de trabalho e geração dos scripts.

## 3.3 Desenvolvimento dos roteiros

Para a confecção dos relatórios, após a realização dos experimentos os resultados foram divididos em subcategorias das atividades sugeridas para os experimentos em malhas aberta e fechada, respectivamente.

## 3.4 Filtros

Para resolver o crescimento demasiado da ação integrativa e consequentemente um atraso na resposta correta de controle, foi colocado um condicional anti-windUp. Essa condicional força um valor máximo e mínimo para parcela integrativa fazendo a mesma saturar. No Experimento, foram utilizados os limites -160 e 160 que correspondem aos valores máximos e mínimos que o sinal de controle deve expressar. O filtro utilizado na ação derivada foi um filtro digital de primeira ordem. Esse filtro compara o valor de entrada com o valor anterior da saída do filtro (inicialmente zero). Se o valor de entrada for maior que o da saída anterior, ele irá fazer uma soma, se for menor, irá fazer uma subtração. O valor a ser somado ou subtraído da saída anterior do filtro (dependendo da comparação anterior) é determinado pela equação: Modulo ((entrada – saída) / PesoFiltro) (ELETROGATE, 2021).

# 4 RESULTADOS

Com o método definido, o desenvolvimento dos dois cenários de controle se seguiu de forma simples. Onde o primeiro experimento em malha aberta pretendia demonstrar o ganho necessário para que se traduzisse a tensão do equipamento em altura. Enquanto o segundo experimento das relações de vazões em um tanque pretendia modelar o funcionamento de um tanque.

## 4.1 Controle em Malha Aberta

### 4.1.1 Limites Operacionais do Motor (RPM mínimo e máximo)

A fim de testar os limites operacionais do motor, foram testados os valores máximos em rotações por minuto do eixo do motor, verificados nos dois sentidos de rotação, visto que dada as condições de montagem do circuito e os componentes, poderiam haver perdas dos sinais transmitidos e recebidos pelo Arduino. Dessa forma, começando com o sentido anti-horário, foi feito o teste alterando o valor do potenciômetro, girando-o no sentido horário, verificando inicialmente o valor registrado de RPM máximo com a resistência mínima e, o valor mínimo mostrado pelo monitor serial em que ainda era possível notar movimento no eixo do motor, capturando seus valores de rotação e PWM. O procedimento foi novamente realizado no sentido horário, como pode ser visto nos resultados concatenados na Tab.3.

Tabela 3: **Limites operacionais**

Sentido	RPM mínimo	RPM máximo	PWM mínimo	PWM máximo
Sentido anti-horário	9	172	37	255
Sentido horário	-9	-179	37	255

### 4.1.2 Zona Morta (Valor de PWM necessário para mover o motor)

Tabela 4: **Valores de Zona Morta**

Sentido	Velocidade angular (RPM)	PWM
Anti-horário	127	84
Horário	-88	68

### 4.1.3 Tempos Necessários para a Realização dos Cálculos e Exibição dos Dados

Como o programa realizava cálculos para estabelecer a velocidade do motor, foi necessário verificar o tempo de cálculo, bem como o tempo que o algoritmo levava para exibir os dados, sendo o tempo de cálculo limitado ao período.

Dessa forma, pelo uso das funções micros() e millis(), nativas do próprio Arduino, foram capturados os tempos de cálculo e de exibição dos dados, respectivamente, registrados na Tab.??

A justificativa do uso da função micros() se deu devido os testes com a função millis() retornarem valores tão pequenos que mostravam um tempo zerado, significando que os valores eram calculados de forma rápida, logo, por necessidade foi buscado uma função com uma precisão melhor.

Com isso, é possível notar que o tempo de exibição do algoritmo pode ser considerado aproximadamente dentro do período estabelecido de 0,1 segundos, embora o período seja estabelecido apenas para a realização dos cálculos de velocidade e da ação de controle

**Tabela 5: Tempo de cálculo e exibição dos dados**

Tempo de cálculo ( $\mu s$ )	9,50
Tempo de exibição dos dados (s)	0,1020

#### 4.1.4 Validação da Resolução do Encoder

Para validar a resolução do encoder é necessário comparar o a resolução já conhecida do encoder utilizado com as resoluções calculadas através da sua leitura de posição. A resolução do encoder é de 341,2 pulsos/volta. Para calcular a resolução experimentalmente foi realizado o cálculo da razão entre a quantidade de pulsos e a quantidade de voltas dadas em um minuto pelo motor. A quantidade de pulsos registrada era desde o inicio do programa.

$$Resolucao_{1min} = \frac{Qtde.depulsos}{Qtde.devoltas}.$$

**Tabela 6: Resoluções calculadas**

Quantidade de pulsos	Quantidade de voltas	Resolução calculada
44.659	121	369,08 pulsos/volta
38.984	107	364,33 pulsos/volta
56.274	155	363,06 pulsos/volta
50.854	137	371,20 pulsos/volta
31.709	85	364,47 pulsos/volta

A resolução média foi de 366,43 pulsos/volta, tendo um erro relativo de 7,4%.

#### 4.1.5 "Estabilidade" das medições de posição e velocidade

Para verificar os valores de estabilidade das medições, foram pegos os dados de velocidade e posição e realizados médias e desvios padrões para verificar a dispersão dos dados em relação ao setpoint. Para a velocidade bastou fazer a média e desvio padrão populacional comum, mas, para a posição, visto que é um valor incremental, essas medições foram feitas com base no erro entre duas amostras consecutivas, de modo que quanto menor a média e o seu desvio padrão, mais precisas seriam as medições. Vale salientar que as medições foram feitas para o sentido horário e anti-horário, a fim de verificar se os valores permaneceram iguais, caso contrário, qual seria sua diferença.

Com isso, verificou-se uma mudança nas médias de velocidade, embora a média do erro tenha permanecido próxima, assim como ocorreu para os valores de posição.

- Sentido horário
  - Estabilidade de medição da velocidade:  $-177,41 \pm 7,94$ ;
  - Estabilidade de medição do erro da velocidade:  $1,23 \pm 8,12$ ;
  - Estabilidade de medição de posição:  $3,94 \pm 22,18$ .

- Sentido anti-horário
  - Estabilidade de medição de velocidade:  $170,00 \pm 7,00$ ;
  - Estabilidade de medição do erro da velocidade:  $1,23 \pm 8,12$ .

## 4.2 Controle em Malha Fechada

### 4.2.1 Teste de Diferentes Valores de Ganho

Para verificar o funcionamento do controlador, foram testados diferentes valores de ganho proporcional e integral, a fim de encontrar valores ótimos para essa execução, sendo escolhidos 4 valores para os resultados variando do pior resultado até o melhor, como podem ser vistos na Fig.18 e na Fig.19

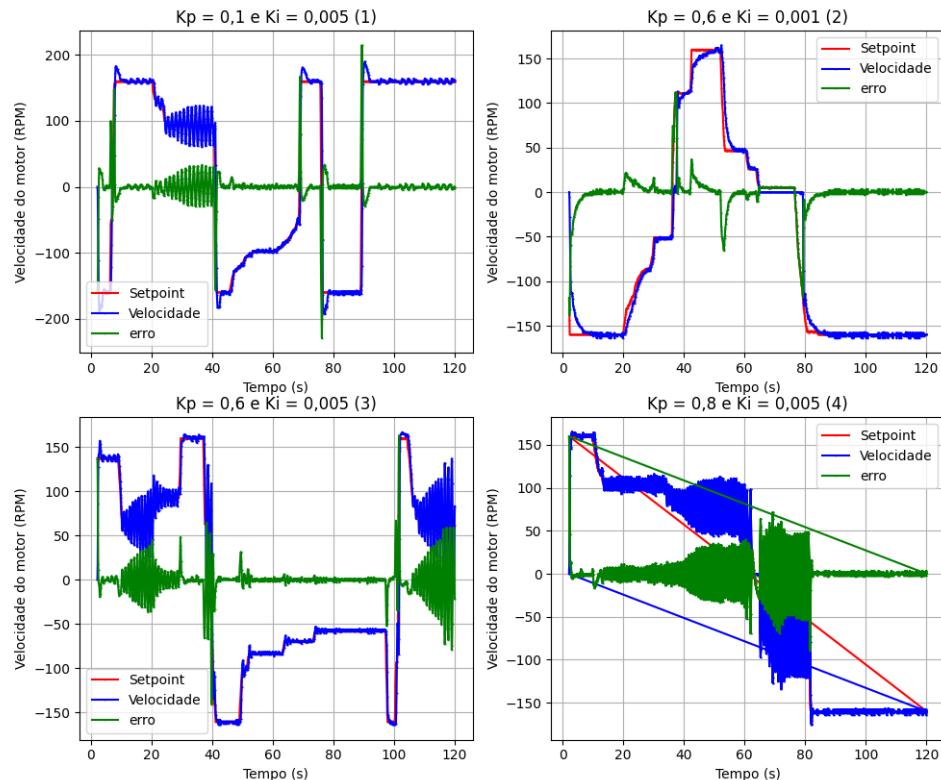


Figura 18: Saída do sistema de controle

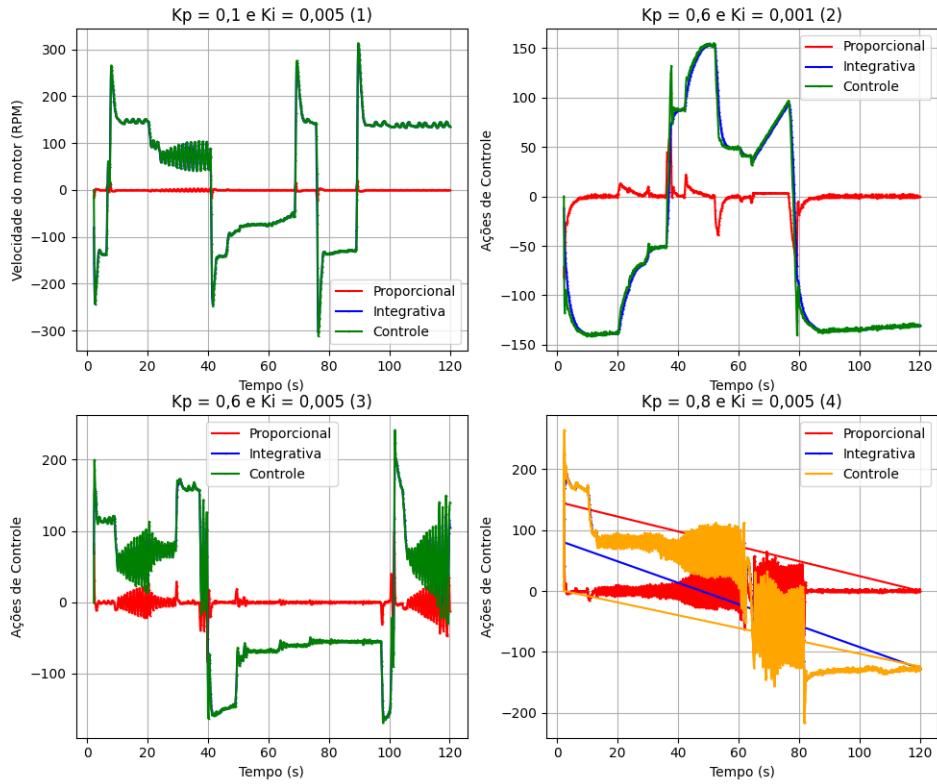


Figura 19: Atuação das ações de controle

Com relação a Fig.18, é possível notar que para valores menores de  $K_p$ , como no gráfico 1 da figura, o sistema apresenta algumas oscilações em valores de setpoint menores, visto que o integrador começa a ter dificuldades em seguir o sinal de controle e seguir a referência, levando em consideração que para tais valores de ganhos, a ação proporcional atua de forma mínimo.

Nesse contexto, ao aumentar o valor do ganho proporcional, mantendo o integrativo, o sistema começa a oscilar nas transições de setpoint de forma brusca, já que a ação integral começa a oscilar muito rapidamente para manter o sinal próximo ao controle e consequentemente próximo ao setpoint, enquanto que a ação proporcional começa a atuar de forma a dar um ganho no sinal de controle, o adequando a saída sem apresentar tantas oscilações, indicando um possível valor ótimo de funcionamento.

A afirmação anterior pode ser vista se comparados os gráficos (1), (3) e (4), sendo o último simulado nas piores condições, em que o motor não funciona fora dos limites operacionais máximos, oscilando em setpoints mais baixos e forçando o eixo do motor em movimento abruptos tentando chegar na estabilidade e reduzir o erro de regime.

A Figura.19 evidencia os argumentos citados anteriormente, bem como, corrobora a afirmação de que o gráfico (2) foi o melhor, visto que a ação proporcional atuou apenas no sentido de adequar o sinal em caso de mudanças de setpoint, mesmo nas mudanças abruptas, sem introduzir um overshoot muito grande, e a ação integral se manteve idêntica ao sinal de controle, fazendo com que o erro tendesse a quase zero e o motor permanecesse no setpoint e conseguisse seguir variações na variável

sem perturbações.

#### 4.2.2 Estratégias de Controle P, PD, PID com e sem Filtro Derivativo e Anti-Windup

As outras atividades sugeridas implementavam outras variações do PID, sendo elas o P, PD e PID, respectivamente, assim como apresentado abaixo nos resultados da Fig.20 até a Fig.27.

Para o controlador proporcional, as medições de saída para a velocidade são identicas ao set-point com uma amplitude reduzida, dado que o valor do ganho é menor do que 1, o sinal sofre uma atenuação, embora não consiga chegar na referência de acordo com o ganho dado.

Sendo assim, foram testados quatro valores de ganhos, como pode ser visto na Fig.20, mostrando que o melhor para apenas o controlador proporcional, o ganho que obteve um menor erro foi  $K_p = 1,5$ , embora para um ganho de  $K_p = 0,7$  o sistema apresentou menores oscilações ao longo da simulação, sendo escolhido como valor próximo ao ótimo para as próximas simulações, uma vez que as outras ações de controle compensam o erro.

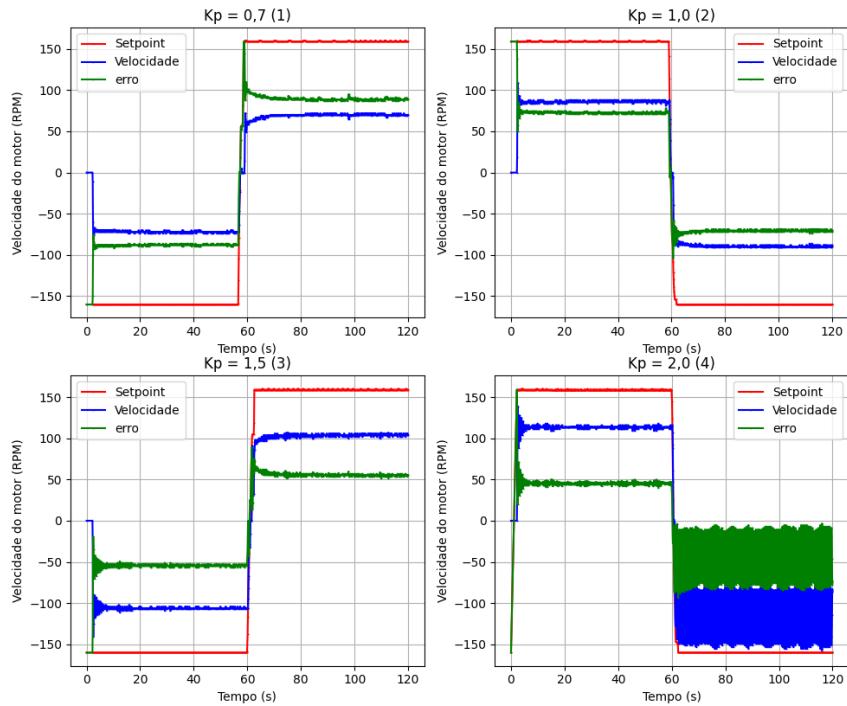


Figura 20: Saída do sistema do controlador P (Proporcional)

De acordo com as ações de controle, considerando o sinal de controle igual ao proporcional, é notório que embora o ganho de  $K_p = 1,5$  tenha tido o menor erro, fica em um valor muito próximo a uma faixa de valores em que o sinal começa a obter um comportamento muito oscilatório em um dos sentidos de rotação em que  $K_p = 2,0$ , explicado pela subita mudança no setpoint que faz o valor do  $K_p$  variar abruptamente em um comportamento oscilatório, como visto na Fig.21.

Com isso, implementado o controlador proporcional e avaliados os valores ótimos de ganho, o próximo passo foi a implementação de um controlador PD, a fim de avaliar a ação derivativa no

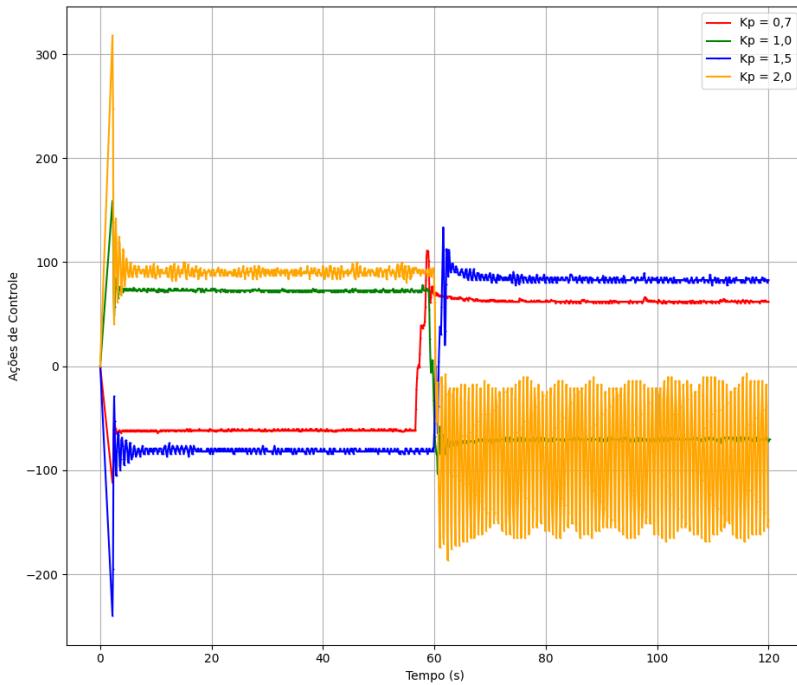


Figura 21: Sinais de controle para o controlador P (Proporcional)

sistema.

Durante os testes com o controlador PD, foi possível perceber um comportamento associado similar ao proporcional, com alguns testes apresentando um menos de ruído devido a ação do derivativa suavizar e evitar variações bruscas no sinal de controle, embora não possua uma ação influente na velocidade do motor.

Na Figura.22 o comportamento da velocidade do motor, com os valores de erro aproximados do experimento anterior, indicando que o derivativo não apresenta ações significativas no sistema, uma vez que, o sinal do motor não demonstra valores ruidosos, podendo ser notado tal comportamento em todos os quatro gráficos.

O sinal de controle na Fig.23 evidencia que a ação derivativa não interfere significativamente no sistema com valores de atuação próximos de 0 e com ganhos  $K_d$  bastante altos para conseguir o mínimo de interação com o sistema, mostrando que para o problema basta apenas um controlador PI, de modo que o proporcional dê um ganho para o integrativo reduzir o erro de regime e manter o sinal de controle igual ao setpoint.

Para o controlador PID, mais dados foram simulados, uma vez que, ao integrar as 3 ações de controle, a saída do sistema poderia ser afetada de forma a melhorar a estabilidade na velocidade do motor.

Foi possível notar que embora as três ações de controle tenham sido implementadas, independentemente dos valores dos ganhos da ação derivativa, sua ação ainda foi considerada nula, como mostram as Fig.26 e Fig.27.

Nesse contexto, foi possível notar que valores de  $K_i$  próximos a  $K_i = 0,005$  introduziram uma

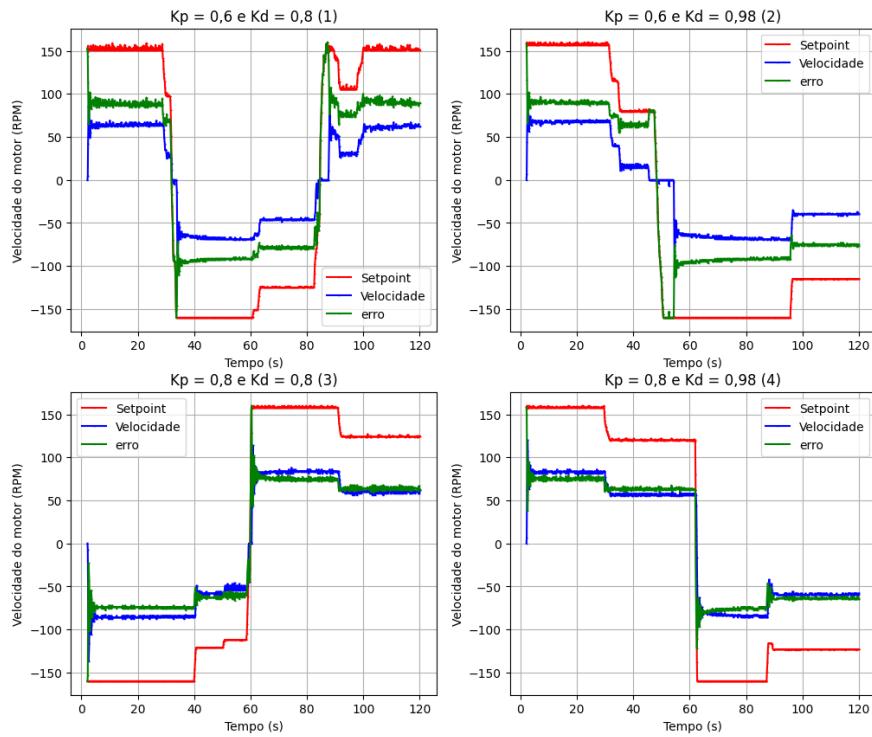


Figura 22: Saída do sistema do controlador PD (Proporcional-Derivativo)

oscilação bastante considerável no sistema, visto que afetou sua performance em valores de setpoint abaixo de 100, fazendo com que o eixo do motor ficasse girando de maneira irregular podendo levar a danos no motor e até mesmo a sua quebra, sendo tais valores representados na Fig.?? que mostra as simulações com valores mais altos de ganho integral para os mesmos ganhos proporcionais e derivativos da Fig.?? que mostra o funcionamento ideal conseguido para o sistema.

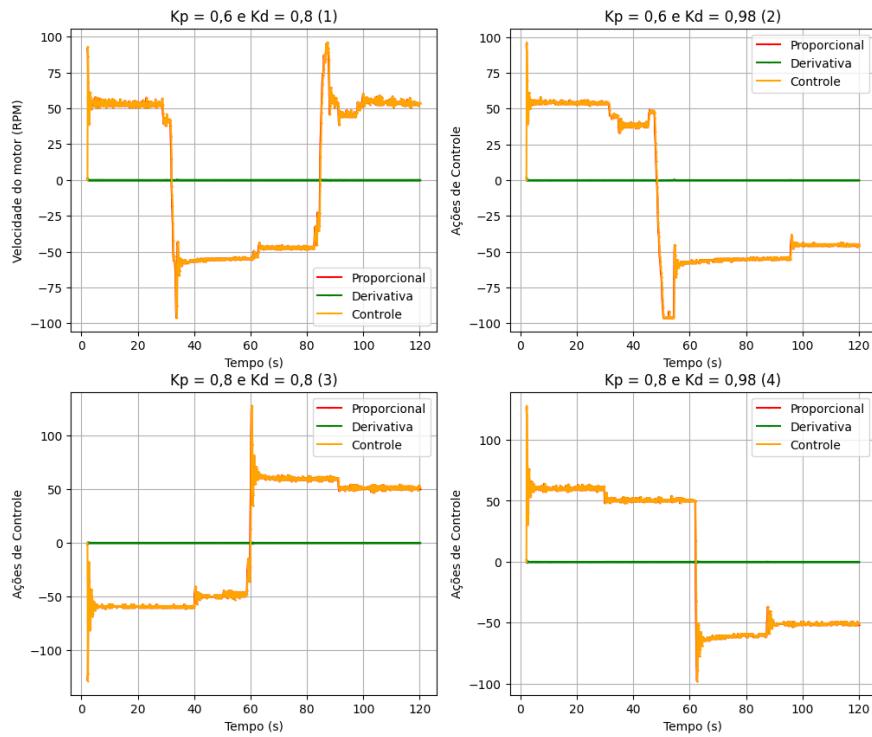


Figura 23: Sinal de controle do controlador PD (Proporcional-Derivativo)

#### 4.2.3 Diferentes Períodos de Amostragem

Nas simulações dos itens anteriores, foi considerado um período de 0,1 segundos para os casos simulados, logo, ao se obterem os resultados, foi simulado novamente o modelo com valores diferentes de período, a fim de avaliar as ações dessa mudança na convergência do sistema, bem como, sua influência no sinal de controle.

Foi escolhido realizar as ações diretamente na implementação do PID, visto que integra todas as ações de controle e permite uma análise geral dos casos.

Nesse contexto, os resultados dos testes são apresentados na Fig.28, mostrando o efeito da mudança no comportamento da velocidade do motor, e na Fig.29, mostrando as ações de controle para os casos simulados.

É possível notar que embora os valores do período de amostragem tenham se alterado, o comportamento do sistema permaneceu o mesmo, visto que a velocidade do motor ainda conseguiu se manter próxima ao setpoint com um erro mínimo, embora o tempo de convergência tenha aumentado, fato que pode ser notado pelo alargamento na escala de tempo.

Na simulação prática, o motor atingia a convergência dando saltos de velocidade, partindo de velocidades maiores para velocidades menores, e vice versa, o que pode ocasionar uma força desgastante se aplicado em condições reais com uma carga em seu eixo.

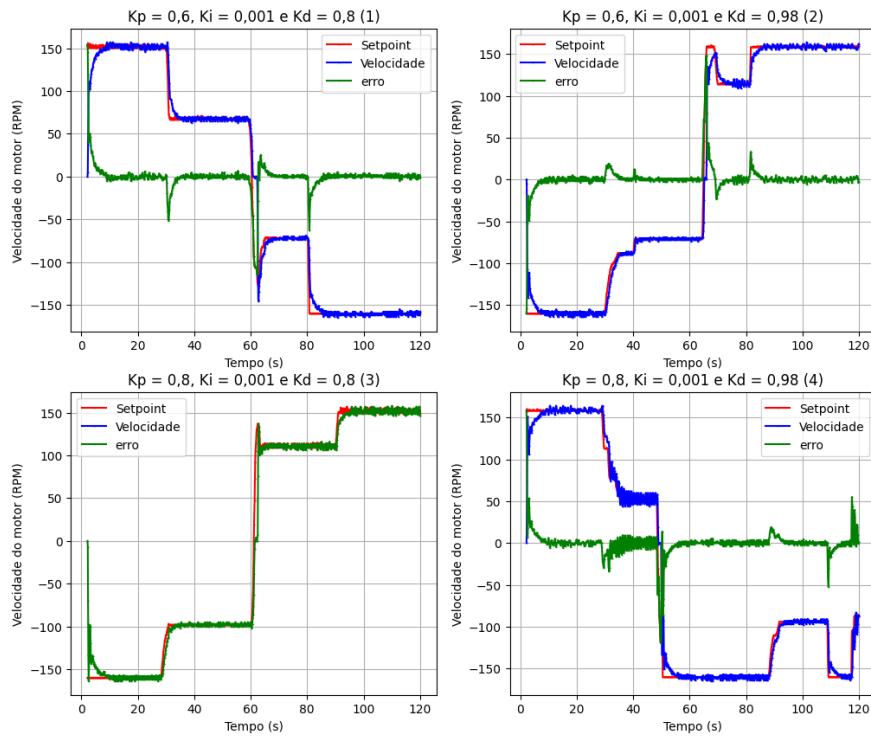


Figura 24: Saída do sistema de controle PID - dados 1

#### 4.2.4 Filtro Passa baixa e Anti-WindUp

A Figura Fig.30 mostra o funcionamento do sistema com o uso dos dois filtros aplicados, anti-WindUp e passa-baixa. Como observado na figura, o valor de controle sobe ate um certo valor e satura. Isso ocorre pela limitação condicional do filtro Anti-WindUp. Entorno do meio do periodo amostral tem 3 variações bruscas (momentos em que foram modificadas o SetPoint) e no final do periodo amostral o valor de controle satura novamente.

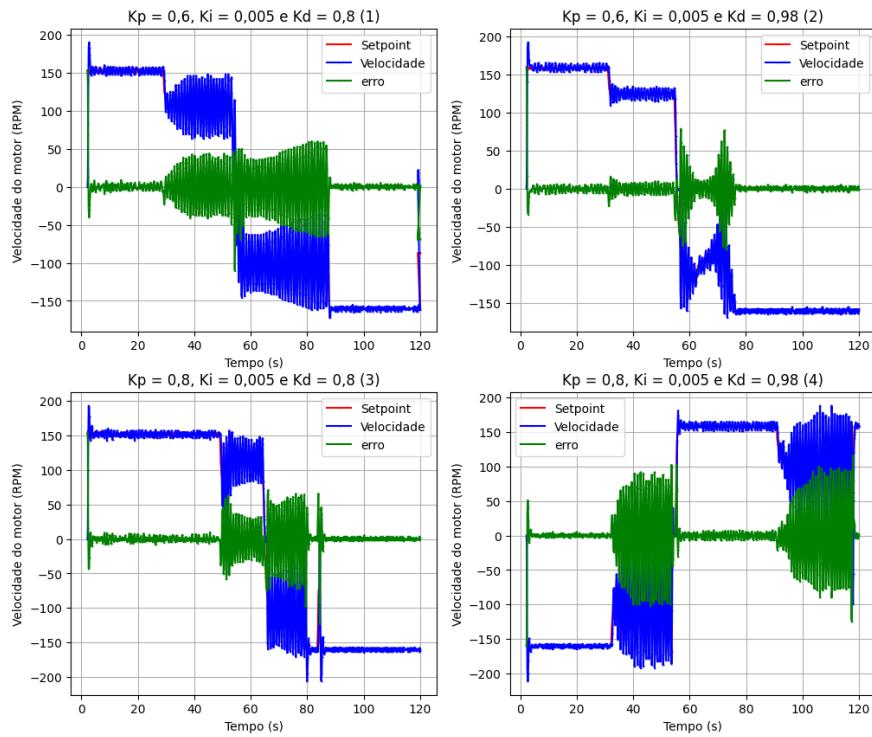


Figura 25: Saída do sistema de controle PID - dados 2

## 5 CONCLUSÃO

Para ganhos da parcela proporcional do controlador foi possível observar que ao iniciar com valores pequenos, um leve aumento melhorou o erro, ou seja, a ação final de controle seguiu bem o setpoint, mas com valores um pouco mais altos o sistema apresentou um comportamento muito distante do esperado, oscilando muito mais.

Vendo os resultados dos filtros, é perceptivo que a parcela derivativa não é muito influente no valor do sinal de controle. Dessa forma, o filtro lógico passa-baixa usado também não foi muito útil, uma vez que o valor do filtro se aproxima bastante do valor da parcela derivativa. Em contra partida, o filtro condicional anti-WinUp na parcela integrativa apresenta melhorias significativas quando comparadas com os testes sem o filtro.

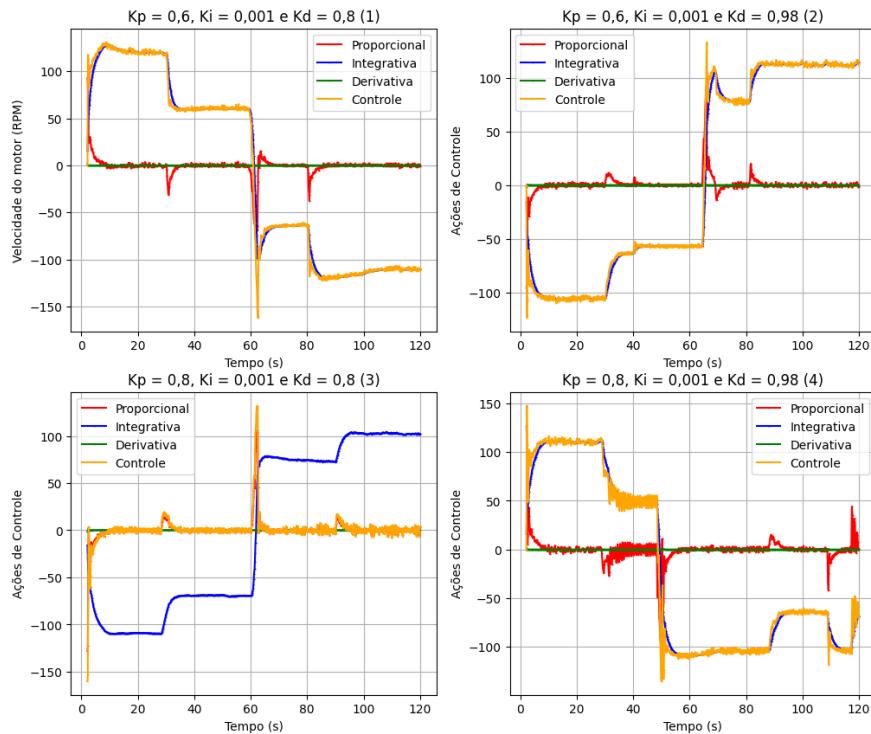


Figura 26: Controlador PID - conjunto de dados 1

## 6 REFERÊNCIAS

- [1]Arduino Mega 2560 datasheet - Arduino. Disponível em: <<https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf>>.
- [2]Como funciona o motor de corrente contínua (ART476) - Instituto Newton C. Braga. Disponível em: <<https://www.newtoncbraga.com.br/como-funciona/3414-art476a.html>>.
- [3]DORF, R. C.; BISHOP, R. H. **Sistemas de controle modernos**. Rio De Janeiro (Rj): Ltc, 2009.
- [4]MATHWORKS. MATLAB - MathWorks. Disponível em: <<https://www.mathworks.com/products/matlab.html>>.
- [5]MENEGETTI, Fábio. Roteiro de laboratório. Universidade Federal do Rio grande do Norte, 2022.
- [6]QUANSER. **Coupled Tanks** . Disponível em: <<https://www.quanser.com/products/coupled-tanks/>>. Acesso em: 1 de maio de 2023.
- [7]Simulink - Simulation and Model-Based Design. Disponível em: <<https://www.mathworks.com/products/simulink.html>>.
- [8]DA SILVA, Joao Manoel. Wind-up da ação integral. [S. l.], 2000. Disponível em: <http://www.ece.ufrgs.br/jmgomes/pid/Apostila/apostila/node31.html>. Acesso em: 18 abr. 2024.
- [9]ELETROGATE. Filtro Digital no Arduino. [S. l.], 2021. Disponível em: <https://blog.eletrogate.com/filtro-digital-no-arduino/>. Acesso em: 18 abr. 2024.
- [10]GOMES, LUCAS MOURA. VALIDAÇÃO EXPERIMENTAL DE FILTRO ESTATÍSTICO PARA CONTROLE PID. [S. l.: s. n.], 2019. Disponível em: [https://bdm.unb.br/bitstream/10483/28125/1/2019\\_LucasMouraGomes\\_tcc.pdf](https://bdm.unb.br/bitstream/10483/28125/1/2019_LucasMouraGomes_tcc.pdf). Acesso em: 18 abr. 2024.

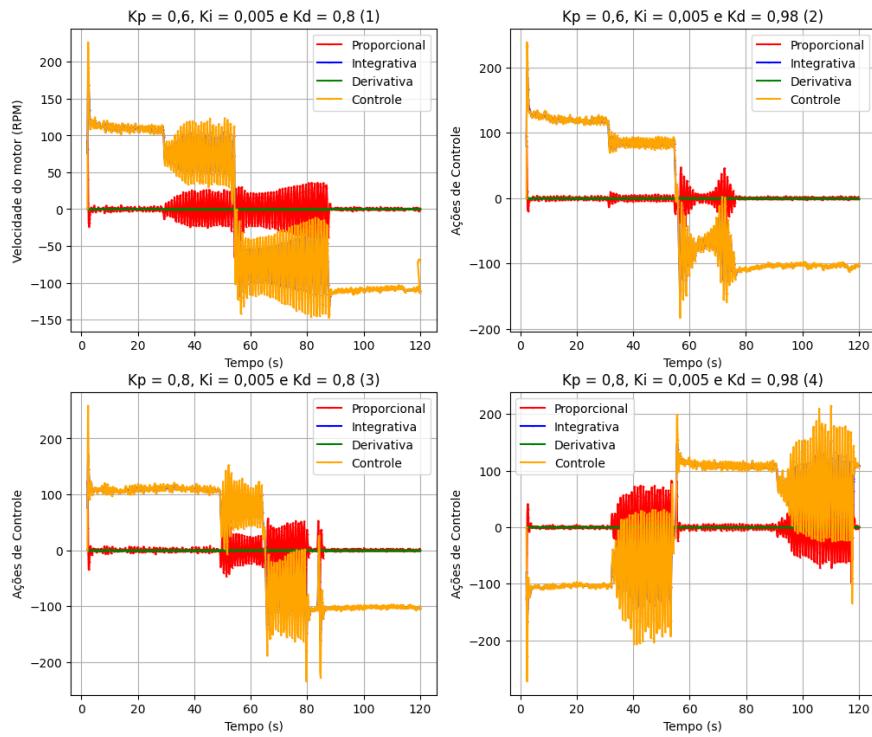


Figura 27: Controlador PID - conjunto de dados 2

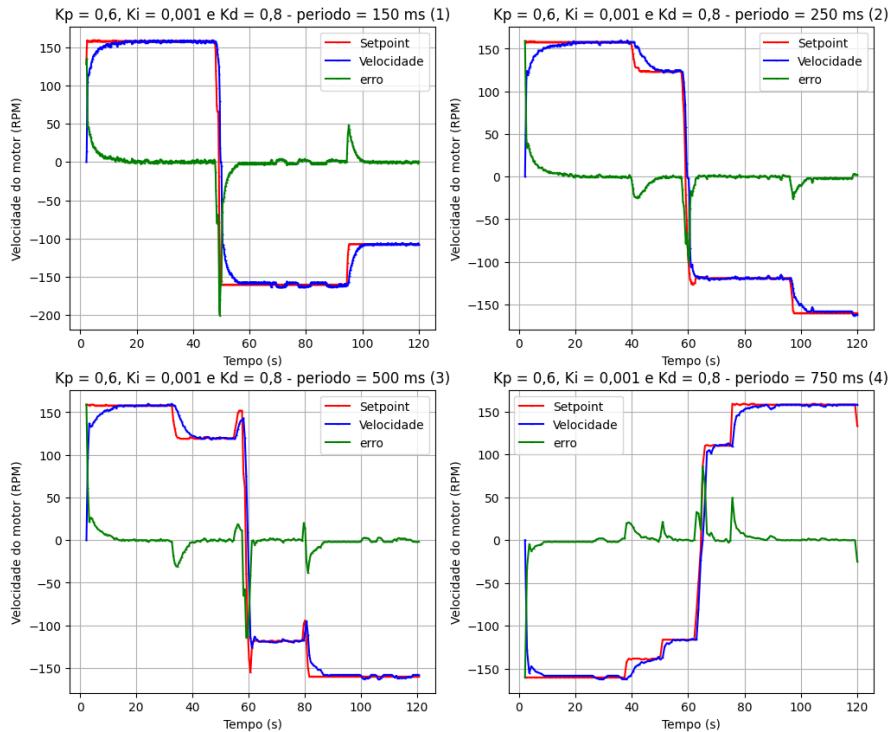


Figura 28: Simulação períodos de amostragem diferentes

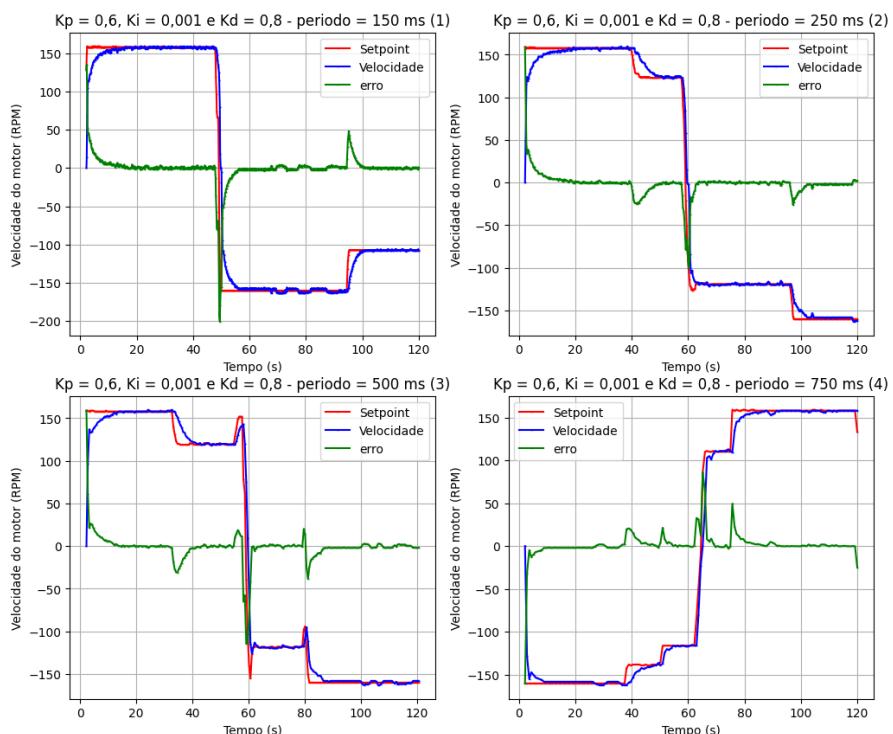


Figura 29: Sinais de controle para períodos diferentes

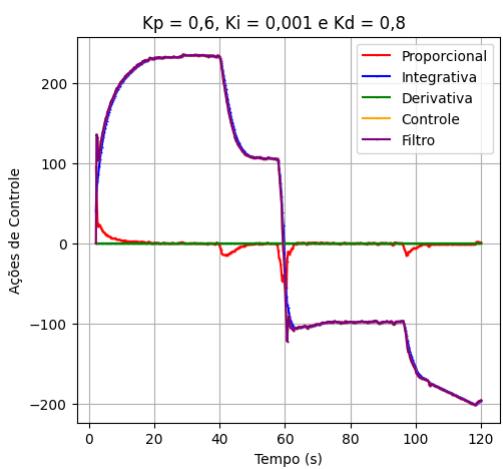


Figura 30: Controlador PID - Filtros