

LUÍS GABRIEL PEREIRA CONDADOS

**CONTROLE EMBARCADO EM ROBÔS  
COM ACIONAMENTO DIFERENCIAL E  
*ENCODERS* DE BAIXA RESOLUÇÃO.**

Natal – RN

Dezembro de 2020

LUÍS GABRIEL PEREIRA CONDADOS

**CONTROLE EMBARCADO EM ROBÔS COM  
ACIONAMENTO DIFERENCIAL E *ENCODERS*  
DE BAIXA RESOLUÇÃO.**

Trabalho de Conclusão de Curso de Engenharia de Computação da Universidade Federal do Rio Grande do Norte, apresentado como requisito parcial para a obtenção do grau de Bacharel em Engenharia de Computação

Orientador: Adelardo Adelino Dantas de Medeiros

Universidade Federal do Rio Grande do Norte – UFRN

Departamento de Engenharia de Computação e Automação – DCA

Curso de Engenharia de Computação

Natal – RN

Dezembro de 2020

Universidade Federal do Rio Grande do Norte - UFRN  
Sistema de Bibliotecas - SISBI  
Catalogação de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede

Condados, Luís Gabriel Pereira.

Controle embarcado em robôs com acionamento diferencial e encoders de baixa resolução / Luís Gabriel Pereira Condados. - 2020.

63 f.: il.

Monografia (Graduação) - Universidade Federal do Rio Grande do Norte, Centro de Tecnologia, Curso de Engenharia de Computação, Natal, RN, 2020.

Orientador: Prof. Dr. Adelardo Adelino Dantas de Medeiros.

1. Engenharia de Computação - Monografia. 2. Controle Embarcado - Monografia. 3. Robôs com acionamento diferencial - Monografia. 4. Filtro de Kalman - Monografia. 5. ESP32 - Monografia. 6. Motores CC - Monografia. I. Medeiros, Adelardo Adelino Dantas de. II. Título.

RN/UF/BCZM

CDU 004

LUÍS GABRIEL PEREIRA CONDADOS

**CONTROLE EMBARCADO EM ROBÔS COM  
ACIONAMENTO DIFERENCIAL E *ENCODERS*  
DE BAIXA RESOLUÇÃO.**

Trabalho de Conclusão de Curso de Engenharia de Computação da Universidade Federal do Rio Grande do Norte, apresentado como requisito parcial para a obtenção do grau de Bacharel em Engenharia de Computação

Orientador: Adelardo Adelino Dantas de Medeiros

Trabalho de Conclusão de Curso apresentado à banca examinadora composta pelos seguintes membros:

---

**Prof. Dr. Adelardo Adelino Dantas de Medeiros - Orientador**  
DCA/UFRN

---

**Prof. Dr. Carlos Eduardo Trabuco Dórea**  
DCA/UFRN

---

**Prof. Dr. Pablo Javier Alsina**  
DCA/UFRN

Natal – RN  
Dezembro de 2020

# RESUMO

Este trabalho apresenta uma proposta e testa na prática um sistema de observação e controle de velocidade para motores de corrente contínua equipados com *encoders* de baixa resolução. O trabalho tem como aplicação alvo mini robôs de acionamento diferencial. O sistema proposto foi implementado no microcontrolador (*ESP32*) e mostrou, pelos resultados experimentais, ser capaz de controlar a velocidade do par motor-rodá de robôs com acionamento diferencial (com *encoders* rotativos de baixa precisão) de forma a conseguir reduzir as assimetrias dos motores. Foi feito o uso de um esquema de controle com duas estratégias, do tipo *Feedforward/Backward* e usou-se o filtro de *Kalman* para realizar a estimativa da velocidade, com o objetivo de reduzir o erro de quantização dos sensores.

**Palavras-chaves:** Controle Embarcado. Motores CC. Filtro de *Kalman*. *ESP32*. Robôs com acionamento diferencial.

# ABSTRACT

This work presents a proposal and tests in practice a speed observer and controller for DC motors equipped with low-resolution encoders. The target application of this work are mini robots with differential drive. The system was implemented in the microcontroller (ESP32). It demonstrated, by experimental results, to be able to control the speed of the motor-wheel pairs of differential drive robots (with low-precision rotary encoders) to reduce their asymmetries. A control scheme with two control strategies, of the type Feedforward / Backward, was used, with the Kalman filter as the speed estimator to reduce the sensors quantization error.

**Keywords:** Embedded Control. DC Motors. Kalman Filter. ESP32. Robots with differential Drive.

# LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama esquemático de um motor CC. . . . .	16
Figura 2 – Equivalente elétrico de um motor CC . . . . .	17
Figura 3 – Diagrama de exemplificação de sistema de controle em malha aberta. . . . .	18
Figura 4 – Diagrama de exemplificação de sistema de controle em malha fechada. . . . .	18
Figura 5 – Diagrama de um sistema controlado por PID. . . . .	20
Figura 6 – Resposta no tempo ao degrau unitário de um sistema de primeira ordem com ganho unitário e contante de tempo $T$ . . . . .	21
Figura 7 – Ilustração do erro de quantização na medição de velocidade de rotação. . . . .	25
Figura 8 – Ilustração de leitura de <i>encoder</i> periódica. . . . .	25
Figura 9 – Sinal em quadratura para rotação no sentido horário. . . . .	27
Figura 10 – Sinal em quadratura para rotação no sentido anti-horário. . . . .	27
Figura 11 – Protótipo do circuito eletrônico dos robôs. . . . .	30
Figura 12 – Ilustração do sistema embarcado como um todo. . . . .	31
Figura 13 – Fluxograma simplificado da rotina de comunicação. . . . .	32
Figura 14 – Estrutura básica usada na comunicação. . . . .	32
Figura 15 – Telecomando para coleta de dados. . . . .	33
Figura 16 – Telecomando de velocidades de referências. . . . .	34
Figura 17 – Telecomando de sinal de controle. . . . .	34
Figura 18 – Comportamento da curva $\omega_{ss}(u)$ . . . . .	36
Figura 19 – Visão geral da rotina de calibração. . . . .	37
Figura 20 – Ilustração do sinal em quadratura em uma revolução completa do eixo do motor no sentido horário. . . . .	38
Figura 21 – Diagrama simplificado do ciclo de controle. . . . .	43
Figura 22 – Diagrama de um sistema de controle <i>Feedforward + Backward</i> . . . . .	44
Figura 23 – Vista explodida do robô completo. . . . .	45
Figura 24 – Frota de mini robôs da Equipe Poti de Futebol de robôs. . . . .	45
Figura 25 – Par de <i>encoders</i> Magnéticos de até 12 pulsos por revolução. . . . .	46
Figura 26 – Micro Motor de 6 V com caixa de redução de 30:1 e <i>encoder</i> magnético. .	46
Figura 27 – <i>driver</i> Motor TB6612FNG. . . . .	47
Figura 28 – Placa de desenvolvimento ESP32 Dev1. . . . .	47
Figura 29 – Bateria Lipo 1S 3,7 V 750 mah Ne-ba931. . . . .	48
Figura 30 – Diagrama esquemático do projeto eletrônico dos robôs. . . . .	49
Figura 31 – Experimento 1. <i>Sinal de controle e Referência igual a 1.</i> . . . . .	52
Figura 32 – Experimento 2. <i>Sinal de controle e Referência igual a 1.</i> . . . . .	54
Figura 33 – Experimento 3. <i>Sinal de controle e Referência igual a 1.</i> . . . . .	56

Figura 34 – Experimento 4. *Sinal de controle e Referência igual a 1.* . . . . . 57

# LISTA DE TABELAS

Tabela 1 – Código de 2 bits para identificar o sentido de rotação.	27
Tabela 2 – Definições utilizadas.	32
Tabela 3 – Codificação de 4 <i>bits</i> para a rotação no sentido horário.	39
Tabela 4 – Codificação de 4 <i>bits</i> para a rotação no sentido anti-horário.	39
Tabela 5 – <i>Lookup table</i> para o motor esquerdo.	39
Tabela 6 – <i>Lookup table</i> para o motor direito.	39
Tabela 7 – Resultado da calibração para os diferentes experimentos.	51

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
1.1	Contextualizando	10
1.2	Trabalhos Relacionados	11
1.3	Motivações e Objetivos	12
1.4	Equipe Poti de Futebol de Robôs	13
1.5	Estrutura do Trabalho	14
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>15</b>
2.1	Motor de Corrente Contínua	15
2.2	Sistemas de Controle	17
2.2.1	Controlador PID	19
2.2.2	Sistemas de Primeira Ordem	20
2.3	Mínimos Quadrados	21
2.4	Filtro de Kalman	22
2.5	<i>Encoders</i> Incrementais	24
2.6	Bluetooth	27
2.7	Modulação por Largura de Pulso	28
<b>3</b>	<b>METODOLOGIA</b>	<b>29</b>
3.1	Sistema Embocado - <i>Firmware</i>	30
3.1.1	Rotina de Comunicação	31
3.1.2	Rotina de Calibração	35
3.1.3	Rotina de Leitura dos Sensores	37
3.1.4	Rotina de Controle	42
3.2	<b>Os Robôs</b>	<b>44</b>
3.2.1	Componentes	46
3.2.2	Projeto Eletrônico	48
<b>4</b>	<b>RESULTADOS</b>	<b>51</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>59</b>
	<b>REFERÊNCIAS</b>	<b>61</b>

# 1 INTRODUÇÃO

Em robótica móvel (mas não limitando-se) é comum o uso de controladores que possuem como saída as velocidades de giro que devem ser atingidas pelos conjuntos motor-rodas do robô. Como por exemplo: controladores estabilizantes, seguidores de caminho e seguidores de trajetória. Portanto é essencial garantir que os motores tentem/consigam atingir essas velocidades, preferivelmente por meio do uso de controladores embarcados no robô. Com isso vem a necessidade de medição e do processamento extra por parte do sistema embarcado do robô, o que pode ser um problema em aplicações com baixo poder computacional e/ou restrições com relação ao uso de sensores de alta precisão, sejam essas limitações econômicas ou estruturais.

## 1.1 Contextualizando

Uma forma muito comum de acionamento de motores de corrente continua (CC) é por meio da modulação por largura de pulso, ou do inglês *Pulse Width Modulation* (PWM), que consiste basicamente em manipular a tensão média de alimentação do motor por meio de sinais digitais. É comum em certas aplicações o uso do sinal *PWM* para manipular a velocidade do motor, ou seja, um controle de malha aberta, esse tipo de sistema apenas regula o quanto rápido o motor irá girar. Porém na maioria das aplicações o controle em malha aberta não é suficiente, como por exemplo, não há garantias que se aplicadas tensões iguais (*PWM*) em ambos os motores de um robô com acionamento diferencial (robôs com duas rodas tracionadas e independentes) o movimento será estritamente linear (Sharan et al., 2007), isso é devido às diferenças entre os dois conjuntos de motores/rodas que fazem com que esses não girem a uma mesma velocidade, mesmo sobre uma mesma tensão.

Em contraste com o sistema de malha aberta, um sistema de malha fechada utiliza o sinal da saída da planta como realimentação para realizar o controle. Esse tipo tende a manter a variável que se está controlando o mais próximo possível da referência desejada. Isso é feito usando a diferença entre a saída e a entrada (erro) como meio de controle.

Portanto, ter-se uma boa realimentação em sistemas de controle em malha fechada é de extrema importância (Petrella et al., 2007), (Sharan et al., 2007), tanto para se trabalhar em alta performance, quanto para garantir a estabilidade do sistema. No contexto de controle de rotação de motores, os sensores mais comuns são os codificadores de eixo (*encoders*).

Os dois principais tipos são: *encoders* absolutos e os incrementais. Os primeiros fornecem a informação da posição angular absoluta e os segundos uma informação da variação de giro.

A resolução desses sensores influencia fortemente no desempenho e até mesmo na viabilidade do controlador, porém, sensores com resoluções maiores são, por consequência, mais caros, fazendo-se que seja necessário um compromisso (Petrella et al., 2007), (Taeg-Joon Kweon; Dong-Seok Hyun, 1999) entre custo e desempenho. Existem diversas técnicas que tentam melhorar a estimativa da velocidade sem que seja necessário o uso de sensores mais precisos, um tipo de técnica que geralmente é usada para esse fim são os estimadores/observadores de estado (Petrella et al., 2007), (Phan Quoc Dzung; Le Minh Phuong, 2005), (Shi; Wang; Xia, 2015), que calculam uma estimativa da velocidade a partir dos dados do sensor e/ou de algum conhecimento sobre o comportamento do sistema. Porém, também há um compromisso com relação a qual estimador usar. Bons estimadores exigem mais poder computacional e/ou exigem que tenha-se mais conhecimento sobre a planta, o que pode ser um problema para certos tipos de aplicações, como é o caso de uma aplicação que deve rodar em um microcontrolador de um robô.

## 1.2 Trabalhos Relacionados

O controle (embarcado ou não) de velocidade de motores e o estudo sobre observadores de estados voltados para estimação de velocidade, usando-se na maioria dos casos *encoders* incrementais como fonte de medição, são assuntos bastante estudados, tanto individualmente quanto em conjunto. No trabalho (Taeg-Joon Kweon; Dong-Seok Hyun, 1999) por exemplo, os autores demonstram em seus resultados uma melhora no desempenho do controle de velocidade do motor usando-se uma técnica de mínimos quadrados recursiva (*recursive least squares*) para fazer o ajuste automático dos parâmetros do controlador e os parâmetros do observador de velocidade por filtro de Kalman. Essa ideia de se empregar estimadores de velocidade/posição e o uso de técnicas para identificação automática de certos parâmetros do sistema também são explorados em (Gaolin Wang et al., 2006), (Hori, 1993a) e (Hori, 1993b).

Em (Xia et al., 2016) os autores propõem um esquema de controle com dois graus de liberdade (equivalente ao *Feedforward + Backward*) de velocidade para servo motores equipados com *encoders* absolutos, com o objetivo de atingir uma boa relação entre a rejeição de perturbações e o rastreio da referência. Os autores também propõem o uso do filtro de Kalman como estimador de estado.

Em (Petrella et al., 2007) os autores apresentam uma análise comparativa entre diferentes técnicas para estimar a velocidade e posição de rotores equipados com *encoders*. O trabalho de (Briz; Cancelas; Diez, 1994) se assemelha a esse, porém as técnicas analisadas por este são apenas as abordagens mais clássicas.

Em (Ravi Kumar et al., 2015) os autores apresentam um trabalho prático que implementa um sistema de controle de velocidade para motores CC usando um microcontrolador e sinal PWM como sinal de controle. Similarmente em (Wu; Chen; Hu, 2008) os autores apresentam uma solução prática de um controle de velocidade de motores CC embarcado em um microcontrolador (ARM).

O trabalho (Okoro, 2019) propõe e mostra bons resultados para o uso de um esquema de controle do tipo *Feedforward feedback* para controle de velocidade de motores CC, com o objetivo de contornar os problemas do controlador PID (controlador *feedback*) com relação a oscilações e problemas com mudanças nos parâmetros desse controlador.

Diante do exposto, é notável que diferentes estratégias estão sendo implementadas com o mesmo intuito: de utilizar técnicas para aprimorar a informação proveniente de sensores de baixa precisão, com o objetivo de melhorar o controle de velocidade/posição de motores; unir diferentes esquemas de controle com a intenção de somar os pontos fortes e compensar os fracos de cada um deles; e de aplicar esses métodos de forma a serem eficientes computacionalmente, para serem factíveis suas implementações em arquiteturas micro-controladas.

### 1.3 Motivações e Objetivos

Existem diversos desafios no projeto de robôs pequenos, além da questão mecânica, o tamanho reduzido também restringe a possibilidade de se usar hardwares com mais capacidade computacional, além dos tipos e a quantidade de sensores que podem ser usados. Robôs de acionamento diferencial como os da equipe Poti de Futebol de robôs (do Departamento de Engenharia de Computação e Automação (DCA) da UFRN), que são voltados para a participação na *Latin American Robotic Competition* (LARC) na categoria *IEEE Very Small Size Soccer* (VSSS), têm dificuldades para conseguirem comportar um hardware capaz de executar um controle digital de velocidade para os dois motores presentes no robô, além de possuírem sensores com uma baixa acurácia, dificultando assim o bom desempenho do sistema de controle.

O objetivo geral deste trabalho é projetar um sistema que realize o controle de velocidade dos motores de um robô com acionamento diferencial utilizando um estimador de velocidade sobre medições feitas com *encoders* de baixa resolução. Para a validação do trabalho foi feita a implementação do sistema proposto, testado em robôs reais e feitas análises da eficiência dos controladores com relação à diminuição das assimetrias dos conjuntos motor-rodas, bem como da qualidade da filtragem/estimativa da velocidade.

## 1.4 Equipe Poti de Futebol de Robôs

A Equipe Poti de Futebol de Robôs surgiu no Departamento de Engenharia de Computação e Automação da Universidade Federal do Rio Grande do Norte (DCA-UFRN), sendo participante na categoria *IEEE Very Small Size Soccer*.

Nessa categoria duas equipes de 3 robôs de até 7,5 x 7,5 x 7,5 cm jogam uma partida de futebol. Os robôs são controlados remotamente por um computador, mas sem intervenção humana (IEEE League, 2020). O computador processa a imagem de uma câmera de vídeo colocada acima do campo e comanda os robôs. Na equipe Poti cada robô possui uma função no jogo, como: o goleiro, o atacante e o defensor, de forma que dependendo da estratégia de jogo eles podem se alternar para melhor desempenho do time. A arquitetura do sistema do Futebol de Robôs do Time Poti é formada por vários módulos, sendo eles: visão, localização, estratégia, controle e transmissão.

O módulo de visão consiste de uma câmera que fotografa o campo com os jogadores e a bola. A taxa de captura da câmera é o fator que limita o tempo de amostragem de todo o sistema, atualmente a equipe utiliza uma taxa de amostragem de 100 quadros por segundo. O módulo de localização fornece a posição da bola e dos jogadores através das cores que foram predefinidas. O módulo de estratégia é responsável por definir a próxima ação que os robôs deverão executar, a partir das imagens atuais. O módulo de controle é responsável por converter as referências de posição e orientação, definidas pela estratégia, em sinais que correspondem às velocidades que devem ser aplicadas nas rodas direita e esquerda de cada robô e o módulo de transmissão envia esse sinal para os robôs (Equipe POTI, 2019).

## 1.5 Estrutura do Trabalho

Este trabalho apresenta uma introdução sobre o tema, mostrando os fatores que motivam a implantação da ideia, além da justificativa e dos objetivos. Em sequência, o Capítulo 2 aborda os principais conceitos teóricos e técnicos utilizados no trabalho. O Capítulo 3, por sua vez, descreve os detalhes de organização e implementação de todas as partes do sistema proposto, bem como os procedimentos adotados para os testes e validação, enquanto o Capítulo 4 trata de mostrar os resultados obtidos experimentalmente e apresenta discussões a respeito destes resultados. Por fim, o Capítulo 5 traz as principais conclusões e contribuições deste trabalho, bem como sugestões de trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

Nesta seção serão apresentados os principais conceitos utilizados neste trabalho, sendo eles: motores de corrente contínua; Filtro de Kalman; Método dos mínimos quadrados; São introduzidos alguns conceitos-chaves sobre a teoria de controle; Breve explicação sobre a comunicação *Bluetooth*; Detalhes sobre o funcionamento dos *encoders* rotativos incrementais e por fim fala-se sobre o funcionamento básico dos sinais PWM.

### 2.1 Motor de Corrente Contínua

Como o próprio nome indica, os motores CC são acionados por uma fonte de corrente contínua. Esse tipo de motor é amplamente utilizado em diversas aplicações.

Ele é constituído basicamente pelo enrolamento de armadura, enrolamento de campo ou ímãs permanentes, comutador e as escovas, onde:

- Enrolamento de armadura: é localizado na parte giratória do motor de CC (rotor) que é responsável por produzir o torque que o movimenta, bem como a tensão de saída quando em modo de gerador.
- Enrolamento de campo: parte fixa responsável pelo fluxo magnético constante que irá atravessar a armadura. Em motores CC de pequenas dimensões, como os utilizados neste trabalho, o enrolamento de campo muitas vezes é substituído pela colocação de ímãs permanentes ao redor da armadura, responsáveis por gerar um campo magnético constante.
- Comutador: Possui a função de manter a corrente da armadura circulando no mesmo sentido, fazendo com que o torque mantenha seu sentido para uma tensão de entrada constante.
- Escovas: é por onde acontece o contato do enrolamento de armadura com a fonte de alimentação.

As máquinas de corrente contínua são bastante utilizadas em sistemas de controle em razão do seu comportamento essencialmente linear. O diagrama esquemático de uma máquina (motor ou gerador) CC é mostrado na figura 1. O enrolamento de campo tem resistência  $R_f$  e indutância  $L_f$  e o enrolamento de armadura tem resistência  $R_a$  e indutância  $L_a$ . As correntes e tensões nos enrolamentos de campo e de armadura  $i_f$ ,  $v_f$ ,

$i_a$  e  $v_a$ , respectivamente. A tensão induzida na armadura é  $v_g$ . O torque e a velocidade angular no eixo do rotor são  $\tau$  e  $\omega$ , respectivamente.

A tensão induzida no enrolamento de armadura é dada por:

$$v_g = K_1 \phi \omega \quad (2.1)$$

E o torque é dado por:

$$\tau = K_1 \phi i_a \quad (2.2)$$

onde  $K_1$  é um parâmetro determinado pela estrutura física da máquina e  $\phi$  é o fluxo magnético. Supondo que a máquina esteja operando na zona linear (ou seja, que o núcleo não esteja saturado), o fluxo é dado pela equação:

$$\phi = K_2 i_f \quad (2.3)$$

onde  $K_2$  é uma constante e depende das características magnéticas do núcleo e do enrolamento de campo. No caso de motores com ímãs permanentes, o fluxo  $\phi = \phi_{\text{const}}$  é constante, determinado pelos materiais e características de construção dos ímãs.

Motores geram potência mecânica, de forma que a velocidade de rotação  $\omega$  é o sinal de saída e a tensão  $v_a$  aplicada é o sinal de entrada.

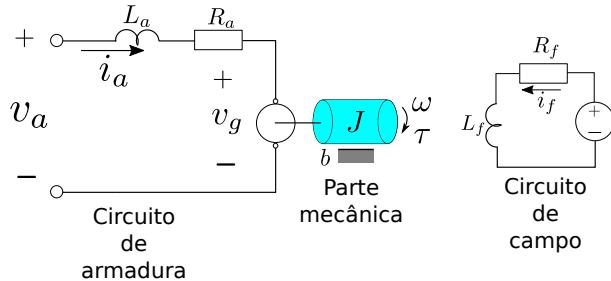


Figura 1 – Diagrama esquemático de um motor CC.

Fonte – Própria.

A Figura 1 apresenta o diagrama esquemático para um motor de corrente contínua (CC) controlado pela armadura, ou seja, o sinal de entrada é a tensão aplicada na armadura ( $v_a$ ). Nesse diagrama a carga está sendo modelada por um momento de inércia  $J$  e um atrito viscoso de coeficiente  $b$ .

$$v_g = K_1 \phi \omega = K_1 K_2 i_f \omega = K_m \omega$$

$$\tau = K_1 \phi i_a = K_1 K_2 i_f i_a = K_m i_a$$

A constante  $K_m$  é conhecida como a constante do motor. Devido à relação apresentada anteriormente é possível modelar o circuito equivalente do motor CC como na imagem 2.

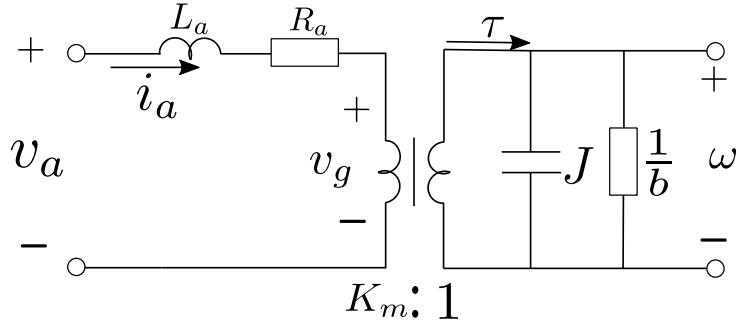


Figura 2 – Equivalente elétrico de um motor CC

Fonte – Própria.

Do circuito da figura 2 extrai-se a seguinte função de transferência:

$$\frac{\Omega(s)}{V_a(s)} = \frac{K_m}{JL_a s^2 + (JR_a + BL_a) s + BR_a + K_m^2} \left[ \frac{\text{rad.s}^{-1}}{V} \right]$$

Caso a impedância da armadura seja desprezada ( $L_a \rightarrow 0$ ), o que quase sempre é possível pois a constante de tempo relacionada ao comportamento mecânico do motor é muito maior do que a relacionada ao funcionamento elétrico:

$$\frac{\Omega(s)}{V_a(s)} = \frac{K_m}{JR_a s + BR_a + K_m^2} = \frac{K}{T_m s + 1} \left[ \frac{\text{rad.s}^{-1}}{V} \right] \quad (2.4)$$

Portanto, caso a impedância da armadura seja desprezada, a função de transferência do motor que relaciona a velocidade angular com a tensão de entrada se comporta como a de um sistema de primeira ordem. A maior dificuldade encontrada ao se controlar motores CC é a amplitude elevada da corrente de armadura, o que requer a utilização de sinal  $v_a$  de entrada fornecido por uma fonte de alta potência.

## 2.2 Sistemas de Controle

O controle automático é um componente importante e intrínseco em sistemas de veículos espaciais, sistemas robóticos, modernos sistemas de manufatura e quaisquer operações industriais que envolvam o controle de temperatura, pressão, umidade, viscosidade, vazão etc. (Ogata, 2011). Um problema de controle consiste em determinar uma forma de afetar um dado sistema físico de modo que seu comportamento atenda às especificações de desempenho previamente estabelecidas. Como, normalmente, não é possível alterar

a estrutura funcional do sistema físico em questão, a satisfação das especificações de desempenho é atingida mediante o projeto e implementação de controladores.

**Sistema de controle em malha aberta (*Feedforward*).** Sistema de controle que não possui a entrada influenciada pela saída do sistema. Isso quer dizer que o sinal de saída não é medido nem realimentado para comparação com a entrada (Ver Figura 3). Devido à natureza desse tipo de controle, a entrada de referência é mapeada para uma saída correspondente do controlador. Dessa maneira, a precisão do sistema depende de uma calibração para melhor ajustar/mapear a referência em um determinado sinal de controle.

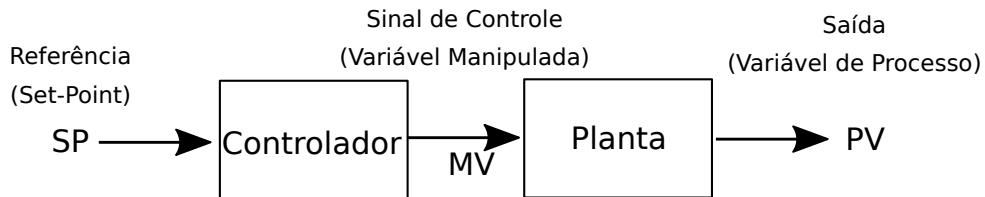


Figura 3 – Diagrama de exemplificação de sistema de controle em malha aberta.

Fonte – Própria.

**Sistema de controle em malha fechada.** Difere essencialmente do em malha aberta, pois neste tipo a entrada do controlador é influenciada pela saída do sistema de controle (variável de processo), (Ver Figura 4). Esse tipo de sistema de controle também é denominado de controle com realimentação. A relação entre a saída e a entrada de referência se dá por meio da comparação entre elas, utilizando a diferença como meio de controle.

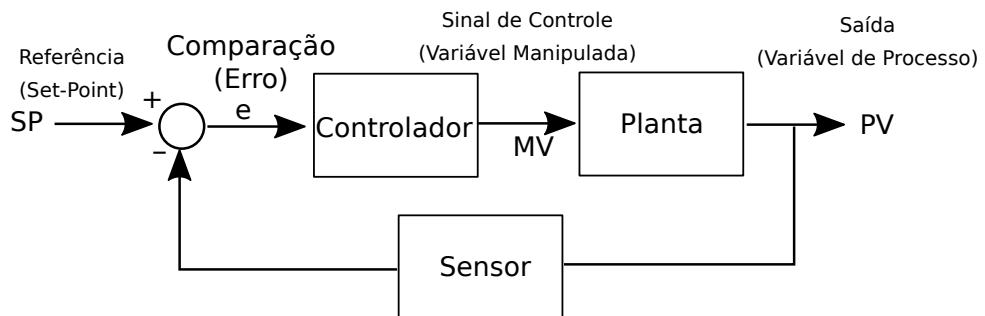


Figura 4 – Diagrama de exemplificação de sistema de controle em malha fechada.

Fonte – Própria.

Segundo (Ogata, 2011) as principais vantagens do sistema de controle em malha aberta são:

1. São de fácil construção/implementação e fácil manutenção.

2. Não apresentam problemas de estabilidade.
3. São mais adequados em situações em que a medição precisa da saída é um problema.

E algumas desvantagens:

1. Distúrbios e mudanças na calibração causam erros, e a saída pode apresentar diferenças em relação ao padrão desejado.
2. Para que a saída mantenha a qualidade requerida, é necessária uma regulagem periódica.

### 2.2.1 Controlador PID

Um controlador clássico do tipo realimentado e que ainda é muito utilizado é o controlador proporcional Integral derivativo (PID). Como o próprio nome indica, esse tipo de controlador une as ações derivativa, integral e proporcional. De maneira simplificada, a ação proporcional faz com que o erro diminua e influencia no tempo de resposta do sistema, a ação integral elimina o erro de regime e a ação derivativa influencia no tempo de resposta do sistema, por meio da antecipação do erro.

Definindo-se os seguintes sinais no tempo:  $u(t)$  como sinal de saída (sinal de controle) e  $e(t)$  como o erro entre a referência e o valor atual da saída da planta, o controlador pode ser definido como:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2.5)$$

Aplicando a transformada Laplace temos,

$$\frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (2.6)$$

Sendo,

- $K_p$ : O ganho proporcional.
- $K_i$ : Ganho integral.
- $K_d$ : Ganho derivativo.
- $s$ : Frequência complexa.

Esse tipo de controlador pode ser utilizado em mais três formas básicas além do padrão PID: Controle Proporcional (P). Apenas a ação proporcional; Controle Proporcional Integral (PI) e Controle Proporcional Derivativo (PD).

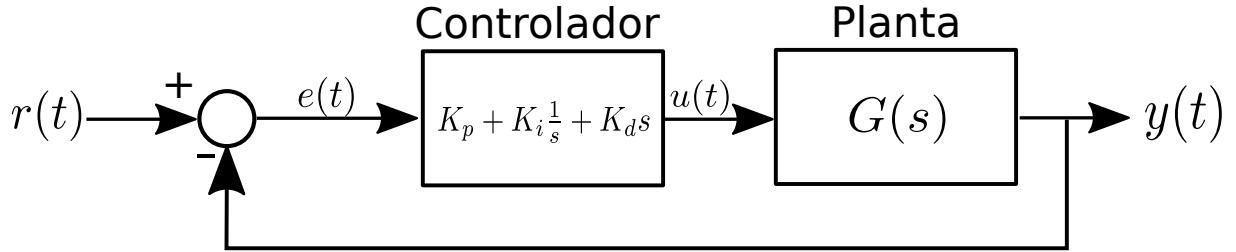


Figura 5 – Diagrama de um sistema controlado por PID.

Fonte – Própria.

A Figura 5 ilustra em diagrama de blocos o uso do controlador PID. Onde  $r(t)$  é o sinal de referência (*set-point*),  $y(t)$  é a saída da planta e  $G(s) = Y(s)/U(s)$  é a função de transferência da planta.

## 2.2.2 Sistemas de Primeira Ordem

Um sistema de primeira ordem apresenta a seguinte forma genérica:

$$Ty + y = Ku \quad (2.7)$$

e em Laplace com a seguinte função de transferência,

$$G(s) = \frac{K}{Ts + 1} \quad (2.8)$$

e sua resposta no tempo para uma entrada do tipo degrau unitário e condições iniciais nulas,

$$y(t) = K \left(1 - e^{-t/T}\right) \quad (2.9)$$

Sendo  $K$  o ganho do sistema e a constante  $T$  é conhecida como constante de tempo do sistema e  $T > 0$ . A Figura 6 ilustra o comportamento no tempo de um sistema de primeira ordem com ganho  $K = 1$  e a influência da constante de tempo  $T$  na resposta do sistema.

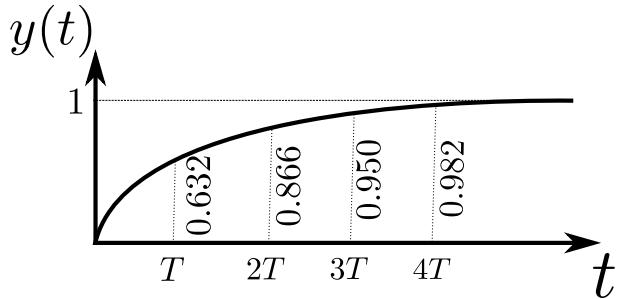


Figura 6 – Resposta no tempo ao degrau unitário de um sistema de primeira ordem com ganho unitário e contante de tempo  $T$ .

Fonte – Própria.

A parte linear da planta a ser controlada (motor CC) é de primeira ordem. Portanto, como não possui nenhum polo na origem, apresentará erro de regime caso seja controlada apenas por um controlador realimentado proporcional.

## 2.3 Mínimos Quadrados

O Método dos Mínimos Quadrados (MMQ) é uma técnica de otimização matemática que procura encontrar o melhor ajuste para um conjunto de dados por meio da minimização da soma dos quadrados das diferenças entre os valores estimados e os observados (tais diferenças são chamadas resíduos).

Um requisito para o método dos mínimos quadrados é que o erro possua uma distribuição normal. Outro requisito é que as variáveis devem apresentar uma relação linear entre si (Brandt, 2014).

Ao usar uma base de dados com  $k$  variáveis explicativas ( $x$ ) e  $n$  observações ( $y$ ), o modelo pode ser escrito na forma matricial:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\alpha} + \mathbf{e} \quad (2.10)$$

onde as matrizes possuem a seguinte forma:

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{21} & \dots & x_{k1} \\ 1 & x_{12} & x_{22} & \dots & x_{k2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1n} & x_{2n} & \dots & x_{kn} \end{bmatrix} \times \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_k \end{bmatrix}, \mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_n \end{bmatrix}$$

, onde  $x_{ji}$  representa o valor da  $j$ -ésima variável da  $i$ -ésima observação.

A solução de mínimos quadrados se dá através da minimização da soma do quadrado dos resíduos conforme apresentado em (Wikipédia, 2020) e o resultado é apresentado na Equação 2.11.

$$\hat{\boldsymbol{\alpha}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \quad (2.11)$$

sendo  $\hat{\boldsymbol{\alpha}}$  a melhor estimativa para os parâmetros que relacionam  $\mathbf{X}$  com  $\mathbf{Y}$ .

Por exemplo, para se realizar uma regressão de um conjunto de observações com o seguinte comportamento linear:

$$y = b + ax \quad (2.12)$$

podemos arranjar os dados na forma matricial:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\alpha}$$

sendo,

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}, \boldsymbol{\alpha} = \begin{bmatrix} b \\ a \end{bmatrix}$$

com isso podemos calcular a melhor estimativa  $\hat{\boldsymbol{\alpha}} = [\hat{b} \ \hat{a}]^T$  como em 2.11.

## 2.4 Filtro de Kalman

"A filtragem de Kalman é um processo de estimativa de estado ótimo aplicado a um sistema dinâmico que envolve perturbações aleatórias. Mais precisamente, o filtro de Kalman fornece um procedimento recursivo linear que visa minimizar a variância do erro para estimar de forma otimizada o estado desconhecido de um sistema dinâmico de dados ruidosos obtidos em tempo real discreto. Tem sido amplamente utilizado em muitas aplicações, como por exemplo, sistemas de rastreamento, navegação por satélite, estimar trajetória de mísseis balísticos e radar."(Chui; Chen, 2009).

O filtro de Kalman usa um modelo dinâmico do sistema (por exemplo, leis físicas do movimento), entradas de controle conhecidas para esse sistema e várias medições sequenciais para formar uma estimativa das quantidades variáveis do sistema (seu estado  $\mathbf{x}_k$ ) que é

melhor do que a estimativa obtida usando apenas as medições. Como tal, é um algoritmo comum de fusão de sensores e fusão de dados (Wikipedia contributors, 2020), (Kalman, 1960).

O filtro produz uma estimativa do estado do sistema como uma média do estado previsto do sistema e da nova medição usando uma média ponderada. O objetivo dos pesos é que os valores com menor incerteza sejam mais "confiáveis". O ganho de Kalman é o peso relativo dado às medições e estimativa do estado atual.

Para o uso do filtro de Kalman deve-se modelar o processo de acordo com a seguinte estrutura. Isso significa especificar as seguintes matrizes (Chui; Chen, 2009):

- $\mathbf{F}_k$ : Modelo de transição de estado.
- $\mathbf{H}_k$ : Modelo de Observação.
- $\mathbf{Q}_k$ : Covariância do ruído do processo.
- $\mathbf{R}_k$ : Covariância do ruído da observação(medição).
- $\mathbf{B}_k$ : Modelo de entrada de controle (caso o sistema dinâmico tenha entrada).

Modelo do sistema para o filtro de Kalman:

$$\mathbf{x}_k = \mathbf{F}_x \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k; \mathbf{w}_k \sim N(0, \mathbf{Q}_k) \quad (2.13)$$

Sendo  $\mathbf{w}_k$  o ruído de processo, que assume-se ter uma distribuição normal de média zero e covariância  $\mathbf{Q}_k$ . ( $\mathbf{w}_k \sim N(\mathbf{0}, \mathbf{Q}_k)$ ).

Modelo da observação/medição será:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k; \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k) \quad (2.14)$$

Sendo  $\mathbf{v}_k$  o ruído de observação, que assume-se ser um ruído gaussiano branco de média zero e covariância  $\mathbf{R}_k$ . ( $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k)$ ).

A filtragem é frequentemente dividida em duas fases: Predição e atualização (Alguns autores também contam a etapa de medição como uma fase do algoritmo). A fase de predição usa a estimativa anterior para produzir uma estimativa do estado atual. Na fase seguinte (de atualização), a predição atual é combinada com as informações da medição atual para aprimorar a estimativa do estado. As etapas do filtro são apresentadas a seguir.

**Predição:**

$$\check{\mathbf{x}}_k = \mathbf{F}_k \hat{\mathbf{x}}_{k-1} + \mathbf{B}_k \mathbf{u}_k$$

$$\check{\mathbf{P}}_k = \mathbf{F}_k \hat{\mathbf{P}}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k$$

Sendo  $\check{\mathbf{P}}$  uma predição da covariância do estado seguinte (predito,  $\mathbf{x}_k$ ).

**Atualização:**

$$\begin{aligned}\mathbf{K}_k &= \check{\mathbf{P}}_k \mathbf{H}^T \left( \mathbf{H}_k \check{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1} \\ \hat{\mathbf{x}}_k &= \check{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \check{\mathbf{x}}_k) \\ \hat{\mathbf{P}}_k &= (\mathbf{I} - \mathbf{K} \mathbf{H}_k) \check{\mathbf{P}}_k\end{aligned}$$

Uma observação importante sobre a etapa de atualização é que o valor atualizado irá tender para o valor predito caso o ganho do filtro  $\mathbf{K}_k$  seja baixo (resultado de um alto erro na medição). O contrário também ocorre, ou seja, para medições precisas ( $\mathbf{R}_k$  baixo) o ganho do filtro tende a ser alto e a etapa de atualização vai tender a usar mais a informação da medição na estimação do estado ótimo ( $\hat{\mathbf{x}}$ ).

## 2.5 Encoders Incrementais

As saídas de um *encoder* incremental são, normalmente, duas ondas quadradas defasadas em  $90^\circ$  uma da outra (em quadratura). Essa diferença de fase nos permite medir o sentido de rotação. Abordagens mais eficientes de leitura desses sensores são por meio da detecção das bordas dessas ondas, pois isso permite quadruplicar o número de pulsos por revolução (NPR) (Briz; Cancelas; Diez, 1994). Existem dois métodos básicos para se realizar a medição da velocidade por meio desses sensores, são eles: **frequencímetro** e **periodímetro**.

Na medição por frequência (**frequencímetro**), conta-se o número de pulsos que ocorreram em um determinado período de tempo fixo. Com isso a velocidade pode ser obtida pela seguinte aproximação:

$$\omega = \frac{d\theta}{dt} \cong \frac{\Delta\theta}{T} \cong \frac{2\pi\Delta N}{N_{PR}T} [\text{rad.s}^{-1}] \rightarrow \frac{60\Delta N}{N_{PR}T} [\text{RPM}] \quad (2.15)$$

Sendo  $N_{PR}$  o número de pulsos por revolução e  $\Delta N$  é o número de pulsos que aconteceram dentro da janela de tempo  $T$ . Existe um erro de quantização nesse método de leitura devido à variação de ângulo medida ser sempre um múltiplo inteiro de  $2\pi/N_{PR}$ . O erro de quantização  $\Delta\omega$  pode ser modelado pela seguinte equação:

$$\Delta\omega = \frac{2\pi}{N_{PR}T} [\text{rad.s}^{-1}] \rightarrow \frac{60}{N_{PR}T} [\text{RPM}] \quad (2.16)$$

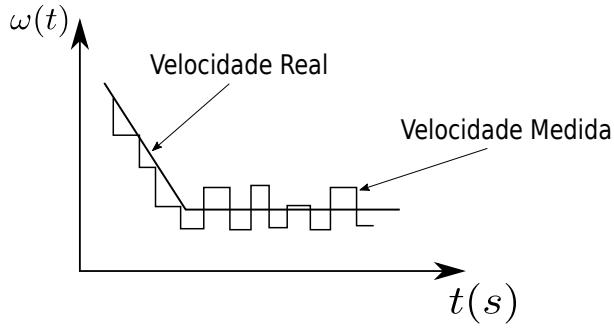


Figura 7 – Ilustração do erro de quantização na medição de velocidade de rotação.

Fonte – Própria.

A Figura 7 ilustra um exemplo de medição de velocidade com erro de quantização.

Nota-se pela Equação 2.16 que o erro de quantização para a medição por frequência decresce com o aumento do número de pulsos por revolução e/ou com o aumento da janela de tempo ( $T$ ), porém o aumento do período de observação acrescenta um atraso na medição da velocidade. O erro relativo à velocidade pode ser descrito como a seguir:

$$e_\omega = \frac{2\pi}{\omega N_{PR} T} \quad (2.17)$$

Pela Equação 2.17 observa-se que o erro relativo decresce conforme a velocidade  $\omega$  aumenta, ou seja, o erro de quantização será mais significante para baixas velocidades.

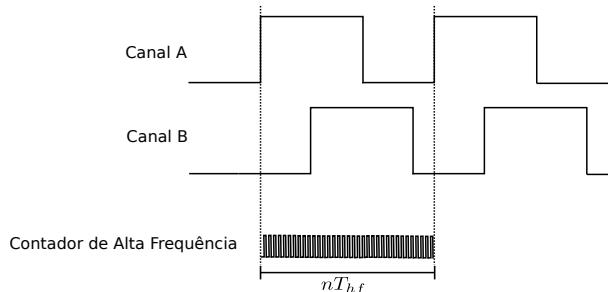


Figura 8 – Ilustração de leitura de *encoder* periódica.

Fonte – Própria.

O outro método de leitura dos *encoders* incrementais é por medição de intervalos de tempo de um mesmo pulso (**periodímetro**) (ver Figura 8). A seguinte formulação pode ser obtida considerando a velocidade do motor constante e sem levar em conta o sentido

de giro:

$$\omega = \frac{d\theta}{dt} \cong \frac{\Delta\theta}{nT_{hf}} \cong \frac{2\pi}{N_{PR}nT_{hf}} [\text{rad.s}^{-1}] \rightarrow \frac{60}{N_{PR}nT_{hf}} [\text{RPM}] \quad (2.18)$$

onde  $T_{hf}$  é o período do sinal do contador de alta frequência do microcontrolador e  $n$  é a quantidade de pulsos, desse contador, que ocorreram entre os pulsos dos *encoder*.

O período entre pulsos será:

$$T_\omega(\omega) = \frac{2\pi}{N_p\omega} [s] \quad (2.19)$$

Uma aproximação para o pior caso do erro de quantização relativo a velocidade é apresentado a seguir (Petrella et al., 2007):

$$e_\omega = \frac{T_{hf}}{\frac{2\pi}{N_{PR}\omega} - T_{hf}} \cong \frac{\omega N_{PR}T_{hf}}{2\pi} \quad (2.20)$$

A Equação 2.20 mostra que o erro é diretamente proporcional à velocidade e a quantidade de pulsos por revolução ( $N_{PR}$ ) e decresce com o aumento da frequência do contador de alta frequência (diminuição de  $T_{hf}$ ).

Nesta versão do trabalho, foi adotada a medição da velocidade pelo método do periodímetro, devido ao fato do método do frequencímetro apresentar erros maiores em baixas velocidades, que normalmente são as situações onde se precisa de um controle mais preciso, e consequentemente de uma medição mais confiável. O método do periodímetro pode apresentar erro maior em altas velocidades, mas em compensação disponibiliza muitas medidas dentro de um mesmo ciclo de controle, que podem ser fundidas pelo filtro de Kalman e gerar um resultado estimado com baixo erro.

Já para se obter o **sentido da velocidade**  $\omega$  é necessário fazer uso do padrão binário gerado pelas bordas dos sinais em quadratura. A Figura 9 ilustrado os sinais em quadratura para uma rotação no sentido horário e destaca o padrão binário gerado nas bordas. A Figura 10 ilustra os sinais no sentido anti-horário. O padrão de dois bits é apresentado na Tabela 1.

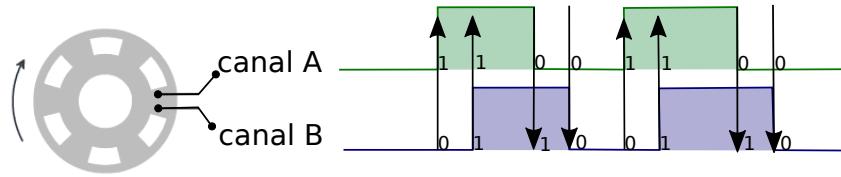


Figura 9 – Sinal em quadratura para rotação no sentido horário.

Fonte – Própria.

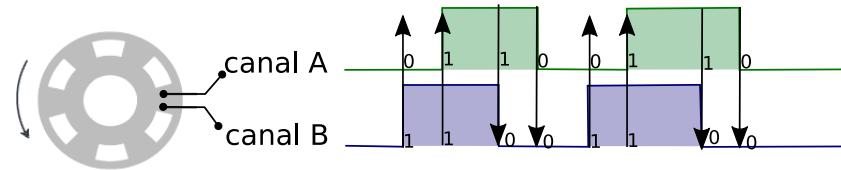


Figura 10 – Sinal em quadratura para rotação no sentido anti-horário.

Fonte – Própria.

Sentido Horário		Sentido Anti-Horário	
A	B	A	B
1	0	0	1
1	1	1	1
0	1	1	0
0	0	0	0

Tabela 1 – Código de 2 bits para identificar o sentido de rotação.

## 2.6 Bluetooth

A tecnologia *Bluetooth* suporta várias opções de topologia, incluindo conexões simples ponto a ponto. Operando na faixa de frequência industrial, científica e médica de 2,4 GHz, a tecnologia *Bluetooth* suporta várias opções de rádio.

O rádio *Bluetooth* BR/EDR opera com baixo consumo de energia e também utiliza uma abordagem robusta de *Adaptive Frequency Hopping*, transmitindo dados por 79 canais. O *Bluetooth* BR/EDR inclui várias opções de configuração para a camada física (*PHY*) que suportam taxas de dados de 1 Mb/s a 3 Mb/s e suporta vários níveis de transmissão, de 1 mW a 100 mW, além de várias opções de segurança.

## 2.7 Modulação por Largura de Pulso

*Pulse Width Modulation* (PWM) refere-se a um sinal digital pulsante. Esse sinal é utilizado, por exemplo, para simular uma saída analógica em microcontroladores e por isso geralmente é muito usado para o controle de atuadores elétricos como motores, aquecedores, dentre outras coisas, como o controle do brilho de LEDs.

A modulação *PWM* pode ser vista como uma maneira de codificar digitalmente níveis de sinal analógico. Nesta técnica, através do uso de contadores de alta resolução, o ciclo de trabalho de uma onda quadrada é modulado para codificar um nível de sinal analógico específico para que então ele atenda os requisitos de uma aplicação desejada. O tempo de ativação é o tempo durante o qual a alimentação CC é aplicada (sinal em nível alto) à carga e o tempo de desativação é o período durante o qual a alimentação é desligada (sinal em nível baixo). Dada uma largura de banda suficiente, qualquer valor analógico pode ser codificado com *PWM*.

### 3 METODOLOGIA

Nessa seção será apresentada de forma técnica e descritiva a metodologia utilizada para as implementações do sistema de controle, do observador/estimador de velocidade (filtro de Kalman), dentre outras implementações que foram relevantes para o trabalho. Além disso serão apresentados os componentes utilizados, bem como uma descrição sobre os robôs que serviram como aplicação alvo para o presente trabalho.

Para a validação do trabalho, o sistema proposto (controlador e filtro de Kalman) foi implementado no microcontrolador (ESP32) dos robôs de acionamento diferencial da equipe Poti. Foram feitas análises em quatro experimentos diferentes, com o intuito de: analisar o comportamento do sistema proposto para diferentes conjuntos de motores e sensores; e em situações diferentes. Os experimentos são:

- **Experimento 1.** Testes em um sistema protótipo equivalente a um robô (Figura 11), sem a parte estrutural e com alimentação por fonte de bancada (fornecendo a mesma tensão de alimentação das baterias nos robôs, 7,4 V). Esse experimento deve fornecer dados menos ruidosos que os demais, pois além de ser feito com motores e sensores mais novos que os utilizados nos demais experimentos, os motores rodam livremente, sem atrito com o chão e sem ter as rodas equipadas nos motores, além de usar uma fonte de alimentação mais estável que as baterias, por se tratar de uma fonte de bancada.
- **Experimento 2.** Testes no robô 1 (robô com a *Tag* verde na Figura 24) sem contato com o chão. Esse experimento, assim como o experimento 3, serve para fornecer dados do comportamento do sistema em uma situação mais realista que o experimento 1, pois é realizado com os robôs, porém suspensos, o que deve fazer com que o ruído nas leituras dos sensores seja menor e as estimativas para os ganhos da planta e dos controladores, calculados na rotina de calibração, sejam diferentes do que se fosse feito o teste no chão.
- **Experimento 3.** Testes no robô 2 (robô com a *Tag* rosa na Figura 24). Esse experimento tem o objetivo de fornecer dados para um conjunto de motor e sensores diferentes do experimento 2, mas são esperados os mesmos resultados.
- **Experimento 4.** Robô 1 no chão. Esse último experimento é o mais realista, ele repete os testes feitos no experimento 3, porém com a diferença de ser realizado no chão.

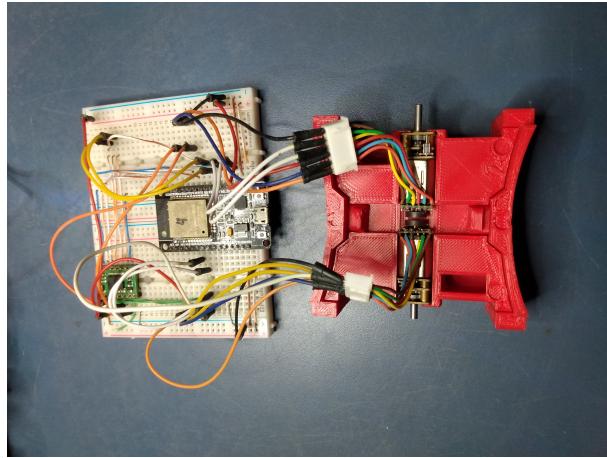


Figura 11 – Protótipo do circuito eletrônico dos robôs.

Fonte – Própria.

Para todos os experimentos foram obtidos os resultados da rotina de calibração (parâmetros do controlador *Feedforward* e ganho do controlador P) e alguns gráficos comparativos.

Os gráficos visam comparar os pares de motores (motor esquerdo e direito) com relação à: velocidades medidas versus velocidades filtradas; sistema com controle versus sistema em malha aberta; comportamento observado pelas medições versus comportamento teórico do sistema. Esses resultados foram obtidos para diferentes velocidades de referência, porém no trabalho serão mostrados apenas os resultados para as máximas velocidades, pois devido à leitura dos *encoders* ser pelo método do periodímetro, as altas velocidades apresentam o maior erro relativo à  $\omega$  (como apresentado na seção 2.5), podendo assim, ser a referência mais problemática para o estimador de velocidade, consequentemente, para o sistema de controle também.

### 3.1 Sistema Embarcado - *Firmware*

A implementação em software embarcado do sistema proposto é dividida em quatro rotinas principais, sendo elas: rotina de comunicação, rotina de calibração, rotina de leitura dos sensores e a rotina de controle. É importante notar que nesse trabalho a rotina denominada de rotina de calibração possui mais de uma função e apesar da nomenclatura coincidir com um conceito muito popular que é o de calibração de sensores, essa rotina não possui essa função, as funções da rotina de calibração foram: identificação de parâmetros (ganho em malha aberta, ganho do controlador *Feedforward* e estimativa da zona morta) e sintonia do controlador proporcional. Essa rotina será explicada em detalhes mais adiante.

Este trabalho fez uso dos múltiplos *cores* presentes no ESP32. O núcleo 1 (também denominado por *APP CPU*) é responsável apenas pela execução do ciclo de controle que atua em ambos os motores (direito e esquerdo). As demais rotinas são executadas de forma concorrente no núcleo 0 (*PRO CPU*), sendo que as rotinas de leitura dos sensores possuem máxima prioridade e a de comunicação possui uma prioridade inferior. Foram realizados testes de tal forma a comprovar que o microcontrolador consegue lidar com essa divisão de tarefas sem que ocorra monopolização de recursos por parte de alguma das rotinas, ou seja, o ESP32 (mesmo configurado para a menor frequência de operação, 80 MHz) consegue lidar com as altas taxas de acionamento das rotinas/interrupções.

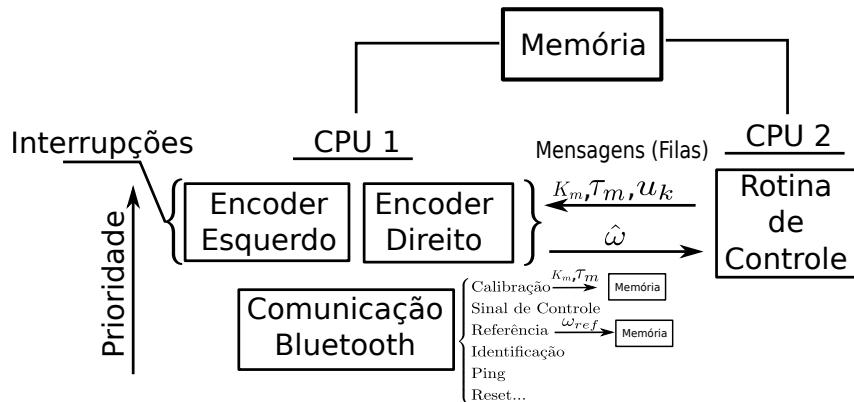


Figura 12 – Ilustração do sistema embarcado como um todo.

A Figura 12 apresenta uma visão do sistema como um todo. A seguir são apresentadas com mais detalhes cada uma das rotinas presentes no sistema embarcado.

### 3.1.1 Rotina de Comunicação

A rotina de comunicação opera em um *loop* infinito no núcleo principal do microcontrolador e é responsável por tratar os telecomandos recebidos pelo *Bluetooth*. Ao ser identificado um recebimento de mensagem pelo sinal de interrupção da comunicação *Bluetooth* é acionada a função de tratamento de interrupção correspondente que possui como única função encaminhar as mensagens válidas (verificar cabeçalho) para a rotina principal de comunicação. Isso é feito para evitar sobrecarregar a interrupção (podendo atrapalhar as interrupções dos *encoders* devido à alta prioridade da interrupção). A Figura 13 ilustra as etapas da rotina de comunicação.

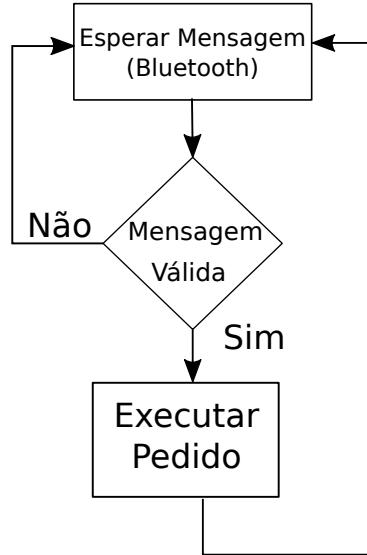


Figura 13 – Fluxograma simplificado da rotina de comunicação.

Na rotina principal a mensagem é interpretada e caso ela seja identificada como um telecomando válido, será executada a devida resposta, conforme apresentado mais adiante.

O protocolo implementado foi pensado para conter até três campos básicos, o **Header** com 4 bits (servindo de preâmbulo), para ajudar a sincronizar os pacotes, o identificador de comando **CMD** também com 4 bits, possibilitando assim até 16 comandos distintos e por fim, o campo de argumentos com tamanho variável, (ver Figura 14). Foram implementados 7 comandos.

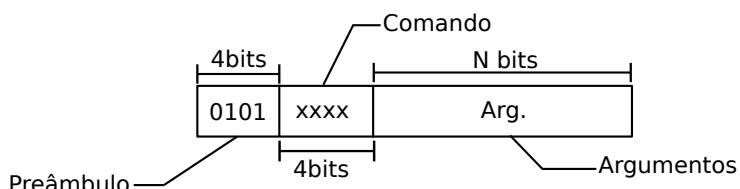


Figura 14 – Estrutura básica usada na comunicação.

DEFINIÇÕES	VALOR(HEX)
HEAD	A0
CMD_REQ_CAL	00
CMD_REQ_OMEGA	03
CMD_CALIBRATION	04
CMD_IDENTIFY	05
CMD_SET_POINT	0A
CMD_CONTROL_SIGNAL	0B
CMD_PING	0F

Tabela 2 – Definições utilizadas.

A Tabela 2 apresenta as definições/nomenclaturas utilizadas nas descrições a seguir.

## Comandos

- **CMD\_REQ\_CAL:**  
 $host$  envia, para solicitar os dados provenientes da calibração do controlador *Feed-forward*. O escravo (robô) envia 4 *FLOATs*, referente aos coeficientes do controlador.
- **CMD\_REQ\_OMEGA:**  
 $host$  envia, para solicitar as velocidades atuais de ambos os motores, em *rad/s*. O escravo responde com dois *FLOATs*, referentes aos valores medidos das velocidades correspondentes.
- **CMD\_CALIBRATION:**  
 $host$  envia, para fazer com que o robô inicie sua rotina de calibração do controlador.
- **CMD\_IDENTIFY:**  
 $host$  envia, fazendo com que o robô inicie sua rotina de identificação (rotina que armazena as velocidades durante um certo período de tempo e envia para o  $host$ , útil para testes). O  $host$  deve enviar uma mensagem conforme ilustrado na Figura 15.

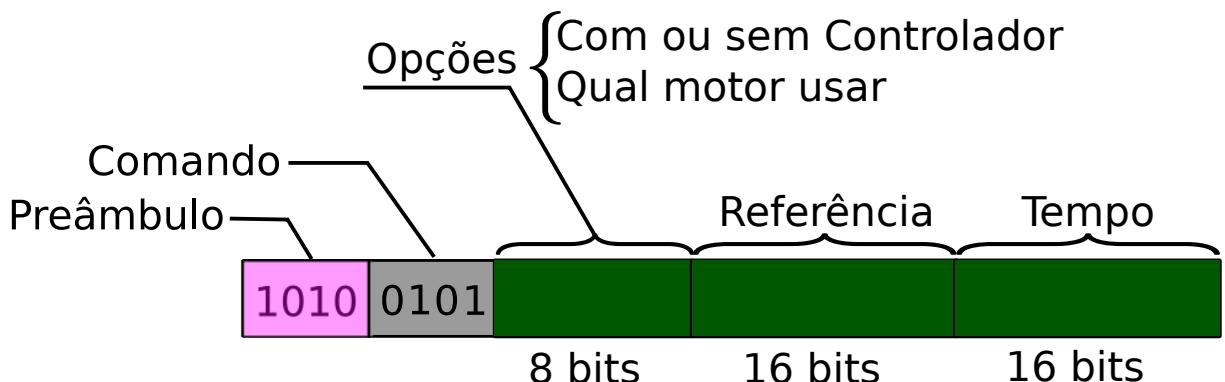


Figura 15 – Telecomando para coleta de dados.

Sendo o campo de opções de 1 byte, contendo a informação de qual motor será feita a identificação e se deve ser usado o controlador.

Ao concluir a rotina de identificação, o robô responde enviando o vetor de velocidades medidas, durante a rotina, para o  $host$ , que deve estar aguardando recebê-las.

- **CMD\_SET\_POINT:**

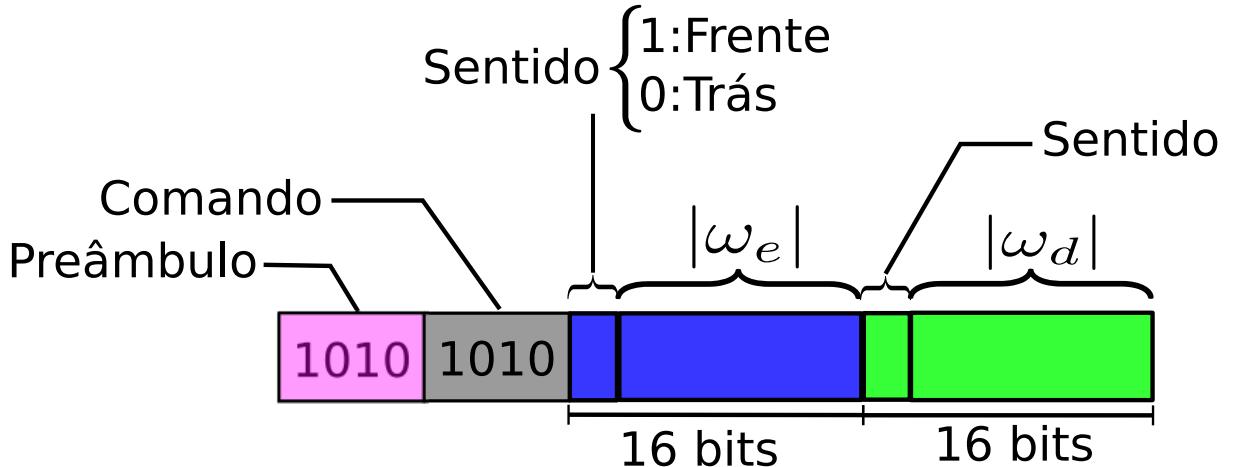


Figura 16 – Telecomando de velocidades de referências.

Neste os campos de sentido (conforme Figura 16), como o próprio nome sugere, indicam o sentido de rotação do motor, 0 para trás e 1 para rodar para frente (convertidos em sinal das velocidades de referência) portanto só ocupam 1 bit, já os campos referentes às velocidades desejadas ocupam 15 bits, sendo assim é possível enviar referências com uma precisão de  $1/2^{15}$ , já que as referências serão enviadas inteiras ( $0 - 2^{15}$ ) e mapeadas de  $-1$  a  $1$ , indicando uma porcentagem da referência da velocidade máxima do robô. Ou seja os campos referentes às referências contêm valores proporcionais à velocidade máxima do robô.

- **CMD\_CONTROL\_SIGNAL:**

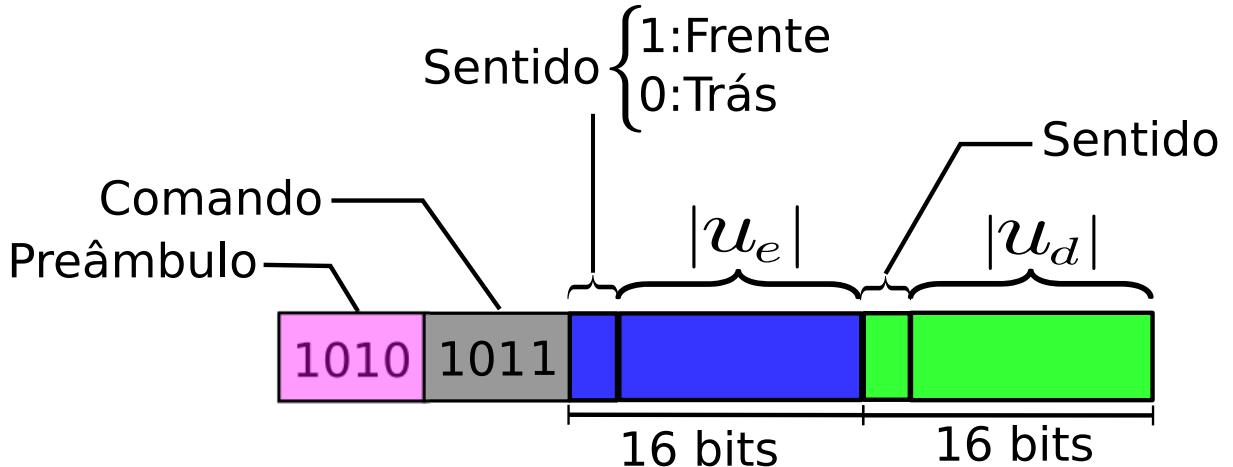


Figura 17 – Telecomando de sinal de controle.

O comando difere apenas no campo de comando do telecomando anterior. O restante da estrutura é exatamente igual, pois a principal diferença ocorre no microcontrolador. Em vez dos campos referentes às velocidades de referência, neste comando o robô irá

interpretar esses campos como contendo sinais de controle (após convertê-los para *float* de -1 a 1).

- **CMD\_PING:**

Neste comando o *host* pode enviar qualquer mensagem no campo de argumentos, pois o robô irá apenas responder com a mesma mensagem. Este comando é útil para testar conexão e testar a latência da conexão.

### 3.1.2 Rotina de Calibração

Rotina responsável por realizar a identificação dos parâmetros do modelo dos motores direito e esquerdo do robô: constante de tempo; ganho de malha aberta; parâmetros do controlador *FeedForward*; velocidade máxima de cada motor, bem como calcular os ganhos para o controlador *P* de forma a se obter uma resposta pré-definida em malha fechada.

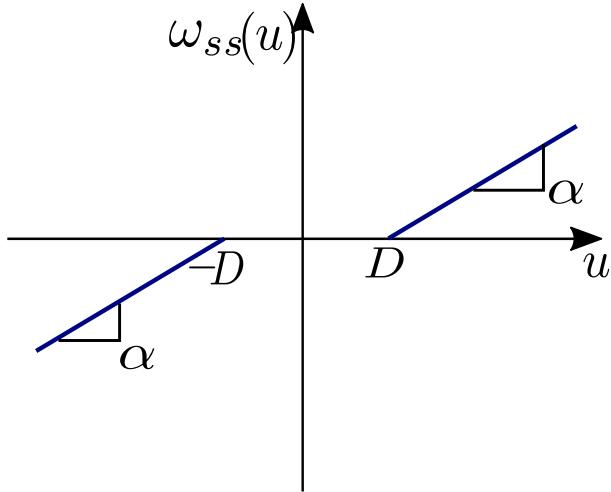
A rotina de calibração consiste em três etapas que são repetidas para todas as configurações: motor direito para frente; motor direito para trás; motor esquerdo para frente e motor esquerdo para trás. Cada etapa é descrita a seguir:

**ETAPA 1.** Estimar a zona morta e o ganho da planta em malha aberta. Para isso é realizada a coleta de  $N$  pontos  $(\omega, u)$ , o primeiro ponto é coletado para  $u = \pm 1$  (valor máximo no sentido de giro atual) e são realizados sucessivos decréscimos neste valor até a parada do motor ( $\omega = 0$ ). A aquisição destes pontos ocorre da maior velocidade para a menor, pois assim o valor em módulo da zona morta será menor, devido à inércia do motor.

Ao se encerrar a coleta destes pontos ( $\omega_{ss} = 0$  detectado) é realizada uma regressão linear por mínimos quadrados, tendo  $u$  no eixo das ordenadas e  $\omega$  no eixo das abscissas, o coeficiente angular dessa reta relaciona a velocidade angular em regime ( $\omega_{ss}$ ) com o sinal de controle ( $u$ ), já o coeficiente linear representa a zona morta, ou seja, o menor  $u$  que pode iniciar o giro do motor (*dead zone (D)*). A regressão linear é possível, apesar do comportamento não linear (como também explorado em (Mendes; Medeiros, 2010)) entre a velocidade e o sinal de controle e ilustrado na Figura 18, se analisado um sentido de rotação por vez.

$$\omega_{ss}(u) = K(u - D) \quad (3.1)$$

A Figura 18 apresenta uma ilustração da relação entre o sinal *PWM* (sinal de controle) e a velocidade de regime  $\omega_{ss}$  do motor.

Figura 18 – Comportamento da curva  $\omega_{ss}(u)$ .

Fonte – Própria.

**ETAPA 2.** Estimar a constante de tempo. Para obter a constante de tempo da planta na configuração atual, faz-se uso mais uma vez de interpolação por *MMQ* e tira-se proveito do conhecimento do ganho da planta para simplificar e tornar possível essa interpolação de forma simples. Para estimar a constante de tempo obtém-se  $M$  pontos  $(t, \omega)$  e aplica-se o *MMQ* para uma interpolação linear, para isso é necessário fazer a seguinte alteração:

$$\begin{aligned}\omega(t) &= K \left(1 - e^{-t/T_m}\right) \\ \ln \omega(t) &= \ln \left[K(1 - e^{-t/T_m})\right] \\ \ln \left(1 - \frac{\omega(t)}{K}\right) &= -\frac{t}{T_m} \\ y_{aux}(t) &= -\frac{t}{T_m}\end{aligned}$$

Convertendo  $\omega(t)$  para  $y_{aux}(t)$  o coeficiente angular resultante da interpolação linear será:  $coef.angular = -\frac{1}{T_m}$ , dessa forma obtemos a constante de tempo.

**ETAPA 3.** Calcular os parâmetros do controlador realimentado. O ganho do controlador proporcional se relaciona com o polo (inverso do negativo da constante de tempo desejada) desejado para o sistema em malha fechada, por meio da relação 3.2. Esse cálculo só é possível devido à identificação dos parâmetros da planta resultante das etapas anteriores. O polo desejado para todas as configurações da planta é uma constante pré-definida, que para os experimentos apresentados neste trabalho foi de  $-20$  (equivalente

a um  $\tau_m$  desejado de 0,08 s).

$$K_p = -\frac{\tau_m S_d + 1}{K_m} \quad (3.2)$$

Sendo  $S_d$  o polo desejado para o sistema em malha fechada, que é equivalente à  $S_d = -1/\tau_{m_d}$ .

Ao se passar por todas as configurações de motor/sentido a rotina seleciona a menor das velocidades máximas apresentada por alguma dessas configurações e armazena esta velocidade como sendo a velocidade máxima atingida pelos motores deste robô (na prática são armazenados 90% desse valor, para dar uma margem de factibilidade), isso é importante, para assegurar que a referência  $\omega_{ref}$  seja factível para todas as configurações.

Por fim os resultados são armazenados na memória permanente do microcontrolador, sendo atualizados/sobrescritos apenas ao final da próxima chamada da rotina de calibração. A Figura 19 apresenta uma visão geral da rotina de calibração.

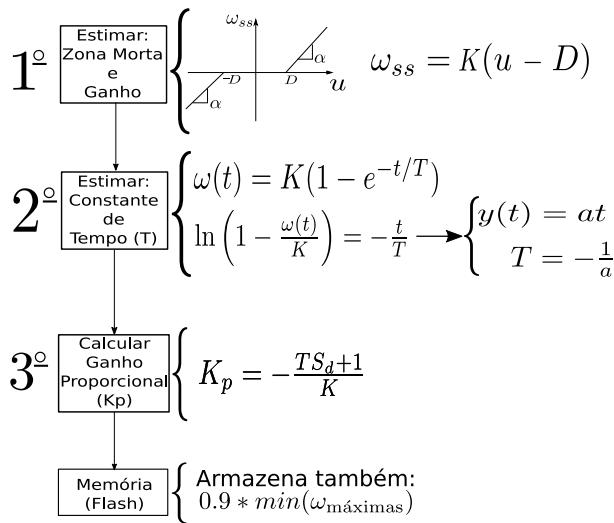


Figura 19 – Visão geral da rotina de calibração.

### 3.1.3 Rotina de Leitura dos Sensores

Há duas interrupções associadas aos sinais provenientes dos *encoders* rotativos, uma para cada motor, cujo objetivo é medir a velocidade de rotação do eixo do motor ( $\omega_{medido}$ ) bem como aplicar o filtro de Kalman para uma melhor estimativa da mesma ( $\hat{\omega}$ ). As interrupções são provocadas pelas bordas dos pulsos de ambos os canais, fazendo com que a resolução do sensor seja utilizada ao máximo, pois dessa maneira os sensores que possuem uma resolução de três pulsos por revolução (em cada canal, ver Figura 20) consigam acionar doze (12) vezes a interrupção que irá computar o  $\omega_{motor}$ , passando assim

a ser ter uma resolução equivalente a doze pulsos por revolução.

Para o cálculo do módulo da velocidade de rotação do eixo do motor faz-se uso da medição pelo período do sinal (conforme a Equação 2.18) com os seguintes parâmetros:

- $N_{PR} = 12$ . Pois são 12 interrupções por canal (monitorando ambas as bordas de subida e descida de um dos canais);
- $T_{hf} = 1\mu s$ . O contador de alta precisão do *ESP32* possui, idealmente, uma resolução de  $1\mu s$  (Systems, a).

A escolha do método de leitura por periodímetro em vez do frequencímetro se deu devido assim ser possível a leitura imediata da velocidade, pois basta apenas uma interrupção para se ter uma medição e também devido à boa precisão da leitura para baixas velocidades, mesmo que isso piore para altas velocidades (como mostrado na Equação 2.20).

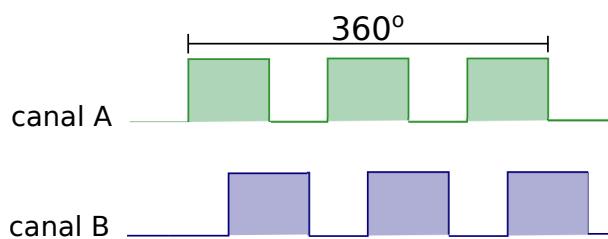


Figura 20 – Ilustração do sinal em quadratura em uma revolução completa do eixo do motor no sentido horário.

Fonte – Própria.

Já para se obter o sentido de rotação do motor, fez-se uso do padrão *Gray Code* gerado pela diferença de fase entre os diferentes canais de um mesmo sensor. Uma maneira de fazer isso é ler os *GPIO's* associados aos canais do *Encoder* e verificar o padrão em binário e inferir o sentido de rotação (conforme apresentado na seção 2.5). Porém esse procedimento apresentou uma alta taxa de erro na inferência do sentido para médias e altas velocidades.

A abordagem adotada aqui foi armazenar os dois *bits* e concatenar/somar com os últimos dois *bits* (da leitura anterior) deslocados em dois (equivalente à multiplicar por  $2^2$  ou operar bit-a-bit:  $bits_{anteriores} \ll 2$ ), criando assim um padrão com 4 *bits*, sendo os dois mais significativos o padrão da leitura anterior e os dois menos significativos a leitura atual. Esse procedimento é ilustrado para uma rotação no sentido horário e no sentido anti-horário respectivamente nas Tabelas 3 e 4, dessa forma geram-se quatro(4)

padrões/códigos que caracterizam um tipo de rotação. Esse padrão de 4 bits é armazenado de forma estática em um vetor(uma tabela de cola/ *Lookup Table*) nas rotinas de ambos os motores. Esse vetor mapeia o código binário em 1, -1 ou zero para as combinações que não caracterizam um sentido de giro. O sinal do valor corresponde ao sentido horário ou anti-horário e depende do motor. Os vetores *lookup table* para ambos os motores são apresentados nas Tabela 5 e 6.

<b><math>A_{ant}</math></b>	<b><math>B_{ant}</math></b>	<b><math>A_{atual}</math></b>	<b><math>B_{atual}</math></b>	<b>DEC</b>
0	0	1	0	2
1	0	1	1	11
1	1	0	1	13
0	1	0	0	4

Tabela 3 – Codificação de 4 bits para a rotação no sentido horário.

<b><math>A_{ant}</math></b>	<b><math>B_{ant}</math></b>	<b><math>A_{atual}</math></b>	<b><math>B_{atual}</math></b>	<b>DEC</b>
0	0	0	1	1
0	1	1	1	7
1	1	1	0	14
1	0	0	0	8

Tabela 4 – Codificação de 4 bits para a rotação no sentido anti-horário.

<b>Índice</b>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<b>Valor</b>	0	-1	1	0	1	0	0	-1	-1	0	0	1	0	1	-1	0

Tabela 5 – *Lookup table* para o motor esquerdo.

<b>Índice</b>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<b>Valor</b>	0	1	-1	0	-1	0	0	1	1	0	0	-1	0	-1	1	0

Tabela 6 – *Lookup table* para o motor direito.

É importante notar que é possível usar apenas uma *lookup table*, porém no trabalho foi feito o uso de duas diferentes, pois cada uma delas é alocada estaticamente em suas respectivas interrupções (uma na interrupção do *encoder* do motor direito e outra na do motor esquerdo) e já apresentam os sinais correspondentes ao sentido, sendo +1, indicando um sentido de giro que favorece o movimento para frente do robô e -1 o movimento para trás, de acordo com o motor.

Com a *lookup table* e o módulo da velocidade é possível calcular a velocidade de rotação da seguinte forma:

$$\omega_{\text{medido}} = \frac{2\pi}{N_{PR}\Delta t} \text{table[code]} \quad (3.3)$$

Sendo,

$$\Delta t = nT_{hf}$$

Uma vez calculado o  $\omega_{\text{medido}}$ , calcula-se em seguida a melhor estimativa para  $\omega(\hat{\omega})$  utilizando-se o filtro de Kalman (conforme apresentado na seção 2.4). Para isso é considerado que o sistema  $\omega(t)$  possui um comportamento de primeira ordem (conforme Equação 2.4), fazendo com que as variáveis do filtro sejam:

$$\begin{cases} \mathbf{x}_k = [\omega_k] \\ \mathbf{z}_k = [\omega_k] \\ \mathbf{H}_k = [1] \\ \mathbf{F}_k = [e^{-\Delta t/\tau_m}] \\ \mathbf{B}_k = [K_m(1 - e^{-\Delta t/\tau_m})] \end{cases}$$

Sobre as equações apresentadas anterior:

- As matrizes  $\mathbf{F}_k$  e  $\mathbf{B}_k$  foram encontradas pela discretização do sistema de primeira ordem ( $G(z) = \Omega(z)/U(z)$ );
- As etapas seguintes desta seção não usarão a notação de vetor (negrito) para se referir às matrizes do filtro, pois essas matrizes serão tratadas como escalares.

No trabalho é feita a consideração de que os erros de quantização nas observações possuem um comportamento gaussiano (essa consideração é possível de acordo com (Sripad; Snyder, 1977)).

Modelo da **medição**:

$$z_k = \omega_{\text{medido}}$$

Com isso a etapa de **predição** do filtro torna-se:

$$\begin{aligned} \check{\omega}_{k|k-1} &= F_k \hat{\omega}_{k-1} + B_k u_k \\ \check{P}_{k|k-1} &= F_k \hat{P}_{k-1} F_k + Q_k \end{aligned}$$

Sendo  $u_k$  o sinal de controle no instante  $k$ .

E a etapa de atualização **Atualização** é:

$$\begin{aligned} K_k &= \check{P}_k (\check{P}_k + R_k)^{-1} = \frac{\check{P}_k}{\check{P}_k + R_k} \\ \hat{\omega}_k &= \check{\omega}_k + K_k (\omega_{k_{medido}} - \check{\omega}_k) \\ \hat{P}_k &= (1 - K_k) \check{P}_k \end{aligned}$$

Os parâmetros do filtro foram sintonizados experimentalmente, sendo eles:

- $Q = 10$ ;
- $R = 1200$ , notou-se pouca variação nesse valor da variância da medição (o valor foi obtido calculando-se a variância da resposta da planta/motor em regime permanente).
- $P_0 = 60$ , valor inicial da incerteza da melhor estimativa.

O algoritmo do filtro de Kalman é executado cada vez que chega uma medida (pulso) de *encoder*, gerando uma medida da melhor estimativa disponível no momento sobre a velocidade angular.

Essa filtragem é assíncrona em relação ao algoritmo de controle: ao ser atingido o próximo instante de geração do sinal de controle, que obedece a uma temporização fixa e pré-definida (período de amostragem), o controlador utiliza o último valor disponível para a velocidade angular estimada.

Foram utilizados estruturas do tipo fila (de tamanho 1, para não causar atrasos extras) para enviar e receber estruturas de dados entre as interrupções dos sensores e as demais rotinas em execução no ESP32. Dessa maneira foi possível evitar o uso de variáveis globais e permitiu-se a troca de informação com a rotina de controle que fica em execução em um núcleo diferente do microcontrolador. A fila de saída contém a estimativa da velocidade e a fila de entrada contém os parâmetros do filtro, bem como o último sinal de controle. Um pseudo código ilustrando a rotina de leitura dos sensores é apresentado a seguir:

---

**Algorithm 1** Rotina de Leitura dos Sensores

---

```

1: lookup_table  $\leftarrow \{0, 1, -1, 0, 0, 0, 0, 1, 0, 0, 0, -1, 0, 1, -1, 0\}$ 
2: code  $\leftarrow (\text{code} \ll 2)$ 
3: code  $\leftarrow \text{code} + ((\text{READ\_GPIO(CANAL\_A)} \ll 1) + \text{READ\_GPIO(CANAL\_B)})$ 
   ▷ Obtém o tempo atual em segundos.
4: t1  $\leftarrow \text{get\_time}()$ 
5:  $\Delta t \leftarrow t_1 - t_0$ 
   ▷ Atualiza a referência de tempo anterior t0
6: t0  $\leftarrow t_1$ 
7:  $\omega_{medido} \leftarrow \frac{2\pi}{N_{PR} * \Delta t} * \text{lookup\_table}[\text{code}]$ 
   ▷ Verifica se tem os dados para usar o filtro.
8: if queue_input.empty() E nunca recebeu then
9:   ▷ Coloca o  $\omega_{medido}$  na fila de saída e encerra a rotina.
10:  queue_output  $\leftarrow \omega_{medido}$ 
11:  return;
12: end if
   ▷ Caso tenha novos dados (fila de entrada não vazia) atualizar os dados presentes
       na rotina. Se não usar os dados anteriores.
13:  $\omega_{ss,\tau} \leftarrow \text{queue\_input}$ 
   ▷ Etapa de predição.
14:  $\check{\omega} \leftarrow \omega_{medido} + (\omega_{ss} - \omega_{medido}) * (1 - \exp(-\Delta t/\tau))$ 
15:  $\check{P} \leftarrow \hat{P} + Q$ 
   ▷ Etapa de atualização.
16:  $K_{gain} \leftarrow \check{P}/(\check{P} + R)$ 
17:  $\hat{\omega} \leftarrow \check{\omega} + K_{gain} * (\omega_{medido} - \check{\omega})$ 
18:  $\hat{P} \leftarrow (1 - K_{gain}) * \check{P}$ 
19: queue_output  $\leftarrow \hat{\omega}$ 

```

---

### 3.1.4 Rotina de Controle

Para garantir que não ocorram interrupções, o controle é executado sozinho no núcleo secundário (APP CPU) do ESP32. A rotina opera em *loop* infinito com uma taxa de atualização de 5 ms e de forma independente das interrupções dos *encoders* (a rotina de leitura dos sensores e a do controle são assíncronas).

Primeiramente a rotina verifica (para cada motor) se existem novas leituras de velocidades, providas pelas interrupções dos sensores. Caso existam, esses valores são atualizados e é passado para a etapa seguinte. Caso contrário é verificado a quanto tempo esses valores não são atualizados. Caso esse tempo seja superior a um limiar pré-definido (no trabalho foi usado esse limiar igual a 500 ms) será considerado que o motor-roda em questão está parado ( $\hat{\omega} = 0$ ) e avança-se para a etapa seguinte.

Com a informação da velocidade atualizada (ou mantida, caso não tenha havido

nova leitura), lê-se as referências para os motores direito e esquerdo. Essas referências são valores no intervalo  $[-1, 1]$ , pois indicam a porcentagem da velocidade que deve ser o *set point* do controlador.

A etapa seguinte é realizar os cálculos de controle, porém antes disso é necessário converter a referência para *rad/s*, faz-se isso multiplicando as referências relativas (provenientes da comunicação) pela velocidade máxima do robô ( $\omega_{max}$ ), obtida na etapa de calibração. A seguir, são computadas, separadamente, a contribuição do controlador *Feedforward* (conforme 3.4) e o proporcional (conforme 3.5) para ao final somá-las e assim obter o sinal de controle ( $PWM/u(t) \in [-1, 1]$ ). Após o sinal de controle passar por um saturador (para garantir que esteja no intervalo admissível) o sinal PWM para ambos os motores é enviado para o *Driver motor* que tratará de excitar as entradas dos motores com as correspondentes tensões. A Figura 21 apresenta uma visão geral simplificada do sistema de controle .

$$u_{ff} = \frac{\omega_{ref}}{K} + D \quad (3.4)$$

$$u_p = (\omega_{ref} - \hat{\omega})K_p \quad (3.5)$$

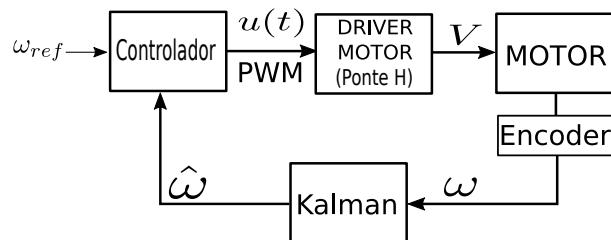


Figura 21 – Diagrama simplificado do ciclo de controle.

Fonte – Própria.

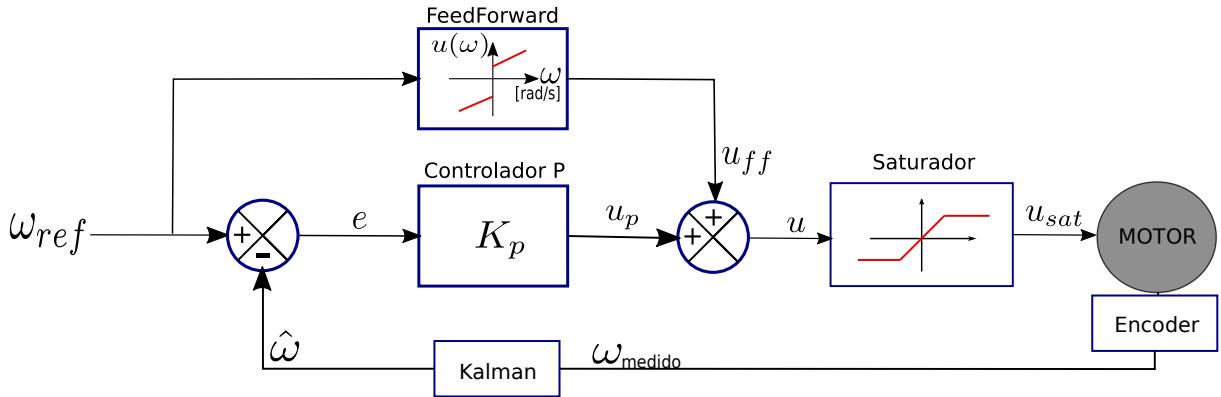


Figura 22 – Diagrama de um sistema de controle *Feedforward + Backward*

Fonte – Própria.

A Figura 22 ilustra o ciclo de controle *Feedforward + Backward*. Podemos ver nessa Figura que a função do *Feedforward*, ou seja, a função que mapeia  $\omega$  num sinal PWM, está sendo feita pela inversão da função ( $\omega_{ss} \times u$ ) estimada na rotina de identificação e sintonia, como apresentado na subseção 3.1.2. A ideia por trás de se somar a ação do *feedforward* com o controlador P é de tentar anular/zear a zona morta do motor, as assimetrias entre motores e o erro de regime, conforme subseção 2.2.2.

### 3.2 Os Robôs

A estrutura dos robôs utilizados foi projetada em software CAD e impressa com o uso de impressão 3D de forma a possuírem uma base quadrada de 7,5 cm de lado e 6,5 cm de altura contando com o raio das rodas que é de 3 cm. A Figura 23 apresenta uma imagem do modelo 3D do robô.

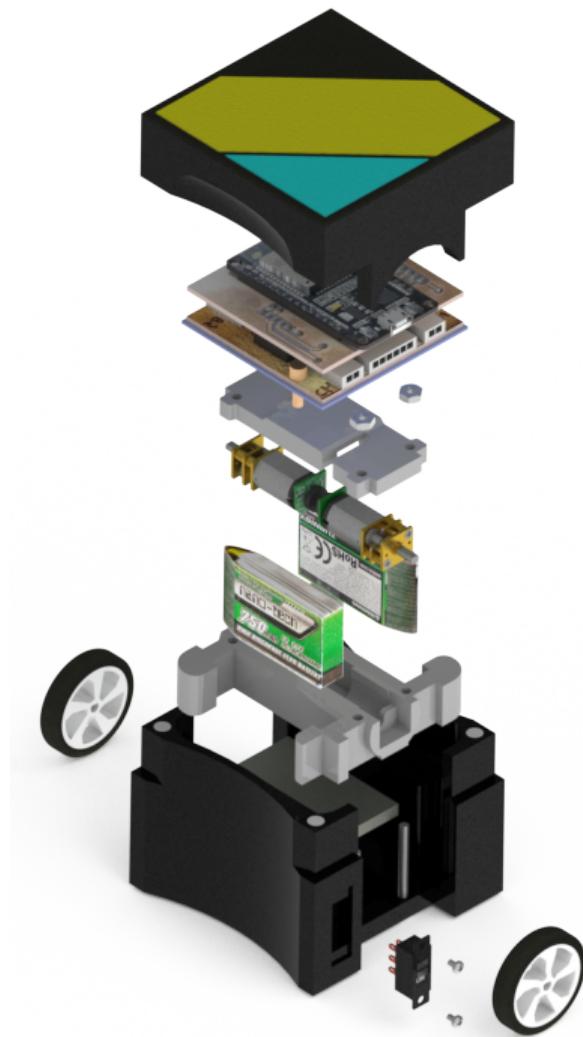


Figura 23 – Vista explodida do robô completo.

Fonte – Victor Duarte de Queiroz (não publicada).

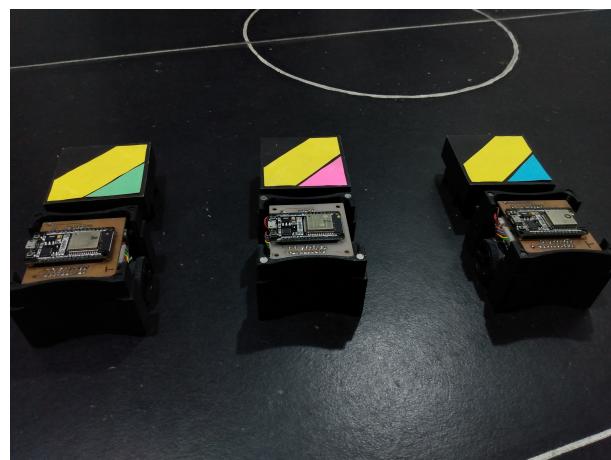


Figura 24 – Frota de mini robôs da Equipe Poti de Futebol de robôs.

Fonte – Própria.

Já a Figura 24 mostra três dos robôs reais (um time) projetados, desenvolvido pela equipe Poti. As subseções seguintes apresentam mais detalhes a respeito do projeto desses robôs.

### 3.2.1 Componentes

São cinco o número de componentes eletro-eletrônicos básicos que compõem os robôs presentes neste trabalho, sendo eles: um par de atuadores (motor direito e esquerdo), um par de sensores de rotação (*encoders* magnéticos), um *driver* motor multicanal, um microcontrolador e baterias recarregáveis. Devido às características dos componentes escolhidos para o projeto, apenas estes cinco tipos foram suficientes para compor a eletrônica do robô de forma a respeitar as restrições dimensionais, realizar o controle dos motores de forma eficiência e com um bom período de amostragem e baixo gasto energético, além de um baixo custo financeiro. A seguir serão apresentados mais detalhes dos componentes supracitados.

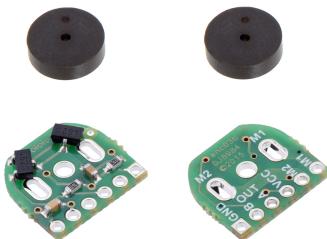


Figura 25 – Par de *encoders* Magnéticos de até 12 pulsos por revolução.

Fonte – (Pololu, b).

Os sensores utilizados são *encoders* rotativos magnéticos do tipo incremental (ver Figura 25). Eles podem ser trabalhados com uma resolução de até 12 pulsos por revolução (se utilizado todas as bordas dos pulsos gerados). Esses sensores operam com uma faixa de tensão de alimentação de 2,7 V até 18 V.



Figura 26 – Micro Motor de 6 V com caixa de redução de 30:1 e *encoder* magnético.

Fonte – (Pololu, c).

A Figura 26 mostra o motor utilizado, equipado com o módulo que contém o *encoder* magnético em seu eixo estendido. Esse é um micro motor de 6 V com uma caixa de redução de  $\approx 30 : 1$ . A caixa de redução faz com que a resolução do sensor seja 30 vezes maior na roda (eixo da caixa de redução) do que no eixo do motor, ou seja, de uma resolução (máxima) de 12 PPR passa a gerar até o equivalente a 360 pulsos por revolução da roda.



Figura 27 – *driver* Motor TB6612FNG.

Fonte – (Pololu, a).

A Figura 27 mostra o *driver motor* utilizado. Ele é responsável por fazer o interfuncionamento entre o microcontrolador e os motores para garantir a separação dos circuitos de alta potência (motores) do circuito de baixa potência. O *driver* possui dois canais de controle de motores, ou seja, possui dois conjuntos de saídas de alta potência (até 3 A e 13,5 V) e dois de entrada para sinal PWM (frequência máxima de 100 kHz, tensão lógica máxima de 5,5 V) para controlar dois motores separadamente. Além disso ele possui circuitos de proteção contra correntes inversas que possam ser geradas pelos motores.



Figura 28 – Placa de desenvolvimento ESP32 Dev1.

Fonte – Própria.

O microcontrolador utilizado foi o ESP32 da empresa Espressif Systems em um *kit* de desenvolvimento rápido (ver Figura 28). Ele possui um microprocessador Tensilica Xtensa LX6 de 32 bits com as seguintes especificações (Systems, b):

- CPUs: 2 núcleos principais e um terceiro núcleo de baixo consumo enérgico.
- Frequência do *clock* dos núcleos: Até 240 MHz
- Desempenho: 600 DMPIS
- Interface *Wireless*: Wi-Fi: 802.11 b/g/n/e/i (802.11n @ 2,4 GHz até 150 Mbit/s), Bluetooth: v4.2 BR/EDR e Bluetooth Low Energy (BLE).
- Memória: 448 Kb de ROM e 520 Kb de SRAM.



Figura 29 – Bateria Lipo 1S 3,7 V 750 mah Ne-ba931.

Fonte – (Systems, c)

A Figura 29 apresenta a bateria utilizada. É uma bateria de 1 célula (1S), portanto, de 3,7 V e com uma capacidade de carga de 750 mAh.

### 3.2.2 Projeto Eletrônico

O principal desafio por trás de se fazer uma placa de circuito eletrônico para esses robôs é acomodar os componentes, de forma a respeitar os limites das dimensões estabelecidos pela competição na qual os robôs serão utilizados (caber dentro de um cubo de 7,5 cm de aresta). A placa deve conter o microcontrolador, no seu kit de desenvolvimento, *driver motor* para acionamento dos motores CC, os *encoders* e ser alimentada por duas baterias de 1 célula do tipo *LiPo* (fornecendo assim uma alimentação de 7,4 V).

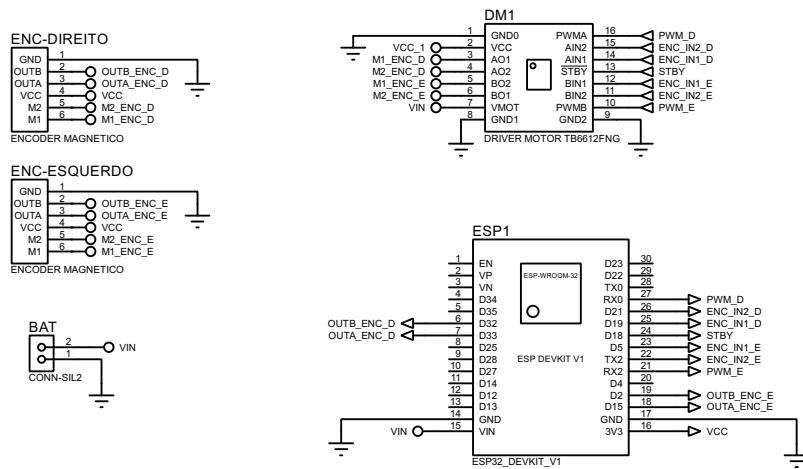


Figura 30 – Diagrama esquemático do projeto eletrônico dos robôs.

Fonte – Própria.

A Figura 30 mostra o diagrama esquemático do sistema eletrônico utilizada nos robôs da equipe Poti. Uma descrição mais detalhada sobre os pinos e as nomenclatura utilizadas no diagrama é apresentado a seguir.

- **OUTA\_ENC\_(D ou E)**: Canal A do sinal de saída do *encoder* do motor direito (D) ou esquerdo (E).
  - **OUTB\_ENC\_(D ou E)**: Canal B do sinal de saída do *encoder* do motor direito (D) ou esquerdo (E).
  - **VCC**: Alimentação de 3,3 V proveniente da saída retificada pelo kit de desenvolvimento do ESP32.
  - **VIN**: Alimentação de 7,4 V resultante da ligação em série das duas baterias Lipo de 1 célula.
  - **M(1 ou 2)\_ENC\_(D ou E)**: Saída de alta potência (7,4 V) vinda de um dos canais de saída *driver motor* (canal A ou B).
  - **STBY**: Pino digital para habilitar ou desabilitar o *driver motor*.

- **ENC\_IN(1 ou 2)\_(D ou E)**: Pinos digitais que configuram o modo de operação da ponte H. Ou seja, controlam o sentido de rotação dos motores.
- **PWM\_(D ou E)**: Saída do microcontrolador para o sinal PWM que regula a alimentação nos motores direito (D) e esquerdo (E).

## 4 RESULTADOS E DISCUSSÕES

Nesse capítulo serão apresentados os resultados e serão feitas algumas observações sobre os dados obtidos por meio da rotina de calibração para os quatro experimentos realizados nesse trabalho. Além de apresentar, graficamente, outras informações (que foram obtidas por meio do telecomando de coleta de dados) sobre o comportamento do sistema implementado.

Experimento	$K_m$ [rad/s]		$\tau_m$ [s]		$K_p$ [1/(rad/s)]	
	Motor Esquerdo	Motor Direito	Motor Esquerdo	Motor Direito	Motor Esquerdo	Motor Direito
	1	3236.80	3047.72	7.62e-02	6.57e-02	1.37e-04
2	4446.69	4514.56	5.71e-02	5.40e-02	3.01e-05	3.23e-05
3	4335.81	4394.23	5.48e-02	5.30e-02	1.54e-05	2.11e-05
4	3345.83	3644.55	4.43e-02	5.90e-02	5.55e-05	3.80e-05

Tabela 7 – Resultado da calibração para os diferentes experimentos.

A Tabela 7 contém, para os quatro experimentos e para cada motor, os resultados da calibração: Ganho da planta  $K_m$ ; Constante de tempo  $\tau_m$  e o ganho do controlador proporcional que foi calculado para se atingir o polo desejado  $S_d = -20$  ( $\tau_d = 0,08$  s) no experimento 1 e  $S_d = -10$  ( $\tau_d = 0,05$  s) nos experimentos 2, 3 e 4.

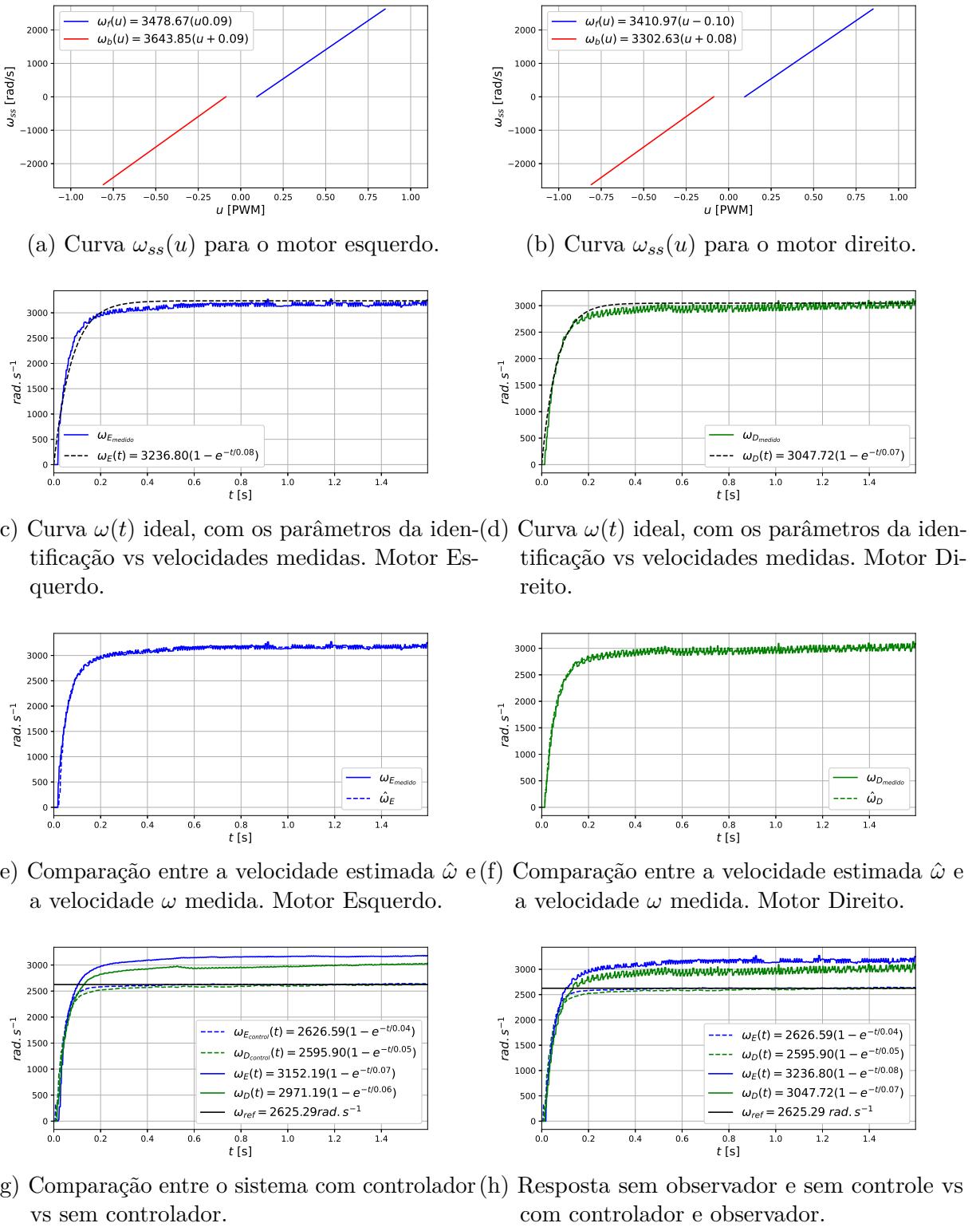


Figura 31 – Experimento 1. Sinal de controle e Referência igual a 1.

Os gráficos comparativos para o experimento 1 estão exibidos na Figura 31. A partir dessas Figuras e da Tabela 7 extraí-se que para o **Experimento 1**:

- Analisando as Figuras 31a e 31b:

As zonas mortas ( $|D|$ ) para o motor esquerdo e direito são respectivamente: 0,09 (em ambos os sentidos de giro), 0,10 e 0,08 (na rotação que favorece o movimento para frente do robô e para trás, respectivamente).

- Pelas Figuras 31c e 31d:

Observa-se que a curva ideal (considerando o sistema de primeira ordem e com os parâmetros da calibração) apresenta um ganho levemente maior do que o que as medições aparentam para ambos os motores. A constante de tempo aparenta corresponder bem ao observado pelas medições.

- Pelos gráficos das Figuras 31e e 31f observa-se que:

para ambos os motores a curva correspondente à velocidade  $\hat{\omega}$  estimada pelo filtro, sobrepõe-se à curva das velocidades medidas, porém, sem apresentar as mesmas oscilações (ruídos de quantização).

- Já pela Figura 31g nota-se para ambos os motores: que a resposta do sistema controlado está bem próxima da referência  $\omega_{ref}$  e que a constante de tempo  $\tau$  está como o desejado (0,08 s).

- Por fim a Figura 31h compara a resposta do sistema com o controlador+observador, com os mesmos e sem estes.

Nota-se que esse experimento apresenta naturalmente uma boa simetria na resposta de ambos os motores, apesar do motor direito apresentar um elevado erro de quantização. Mesmo assim, é possível observar que tanto a simetria é melhorada/mantida quanto o erro de quantização diminui significativamente para ambos os motores.

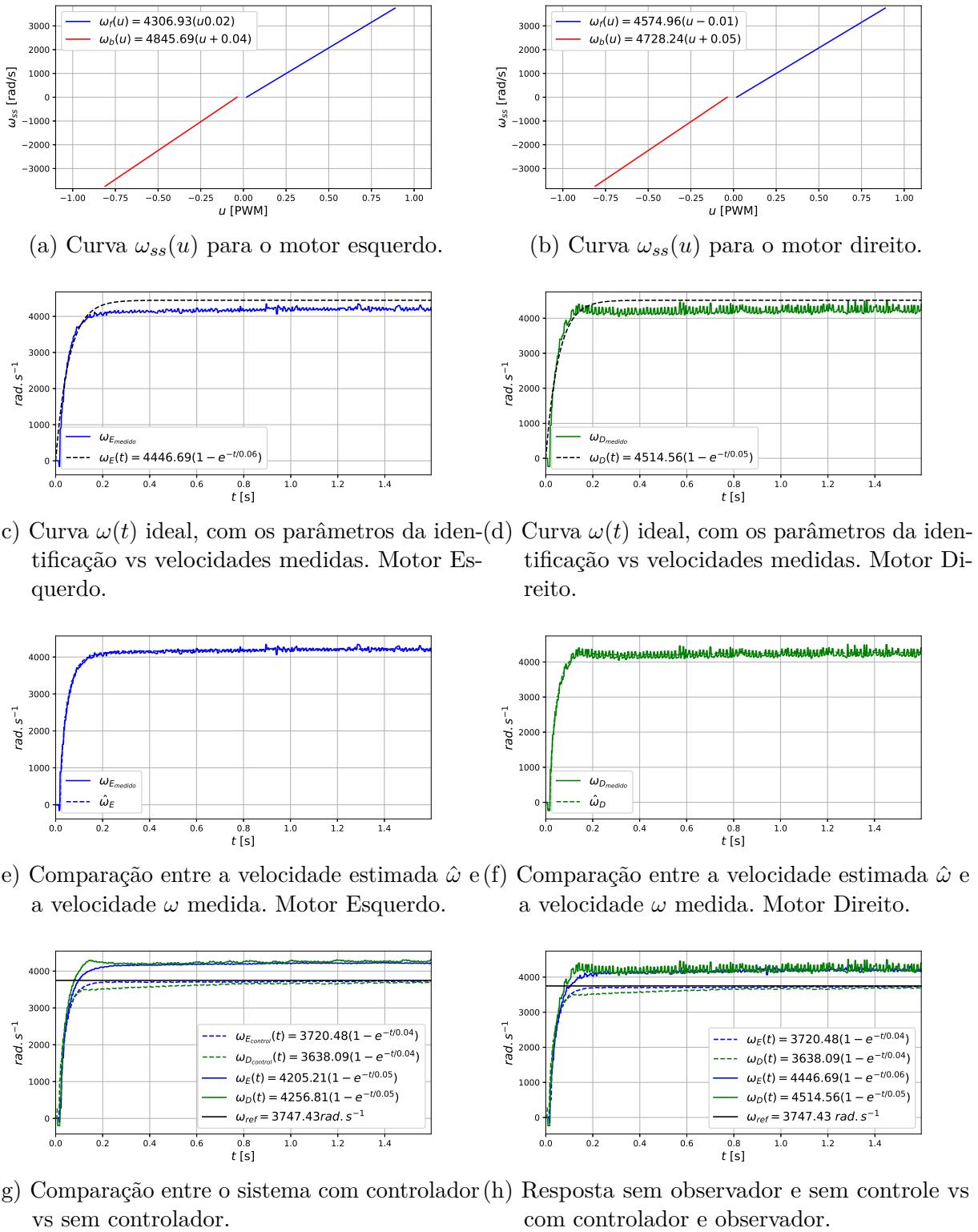


Figura 32 – Experimento 2. Sinal de controle e Referência igual a 1.

Os gráficos comparativos para o experimento 2 estão exibidos na Figura 32. A partir dessas Figuras e da Tabela 7 extraí-se que para o **Experimento 2**:

- Analisando as Figuras 32a e 32b:

As zonas mortas ( $|D|$ ) para o motor esquerdo e direito são respectivamente: 0,02, 0,04 (para frente e para trás, respectivamente), 0,01 e 0,05 (na rotação que favorece o movimento para frente do robô e para trás, respectivamente).

- Pelas Figuras 32c e 32d:

Observa-se que, assim como para o experimento 1, a curva ideal (considerando o sistema de primeira ordem e com os parâmetros da calibração) apresenta um ganho levemente maior do que o que as medições aparentam para ambos os motores. E a constante de tempo aparenta corresponder bem ao observado pelas medições.

- Pelos gráficos das Figuras 32e e 32f observa-se que:

para ambos os motores a curva correspondente à velocidade  $\hat{\omega}$  estimada pelo filtro, sobrepõe-se à curva das velocidades medidas, porém, sem apresentar o ruído de quantização.

- Já pela Figura 32g nota-se para ambos os motores: que a resposta do sistema controlado está bem próxima da referência  $\omega_{ref}$  e que a constante de tempo  $\tau$  está como o desejado (0,05 s).

- Por fim a Figura 32h compara a resposta do sistema com o controlador+observador, com os mesmos sem estes.

As mesmas observações feitas no experimento 1, são válidas para esses gráficos neste experimento.

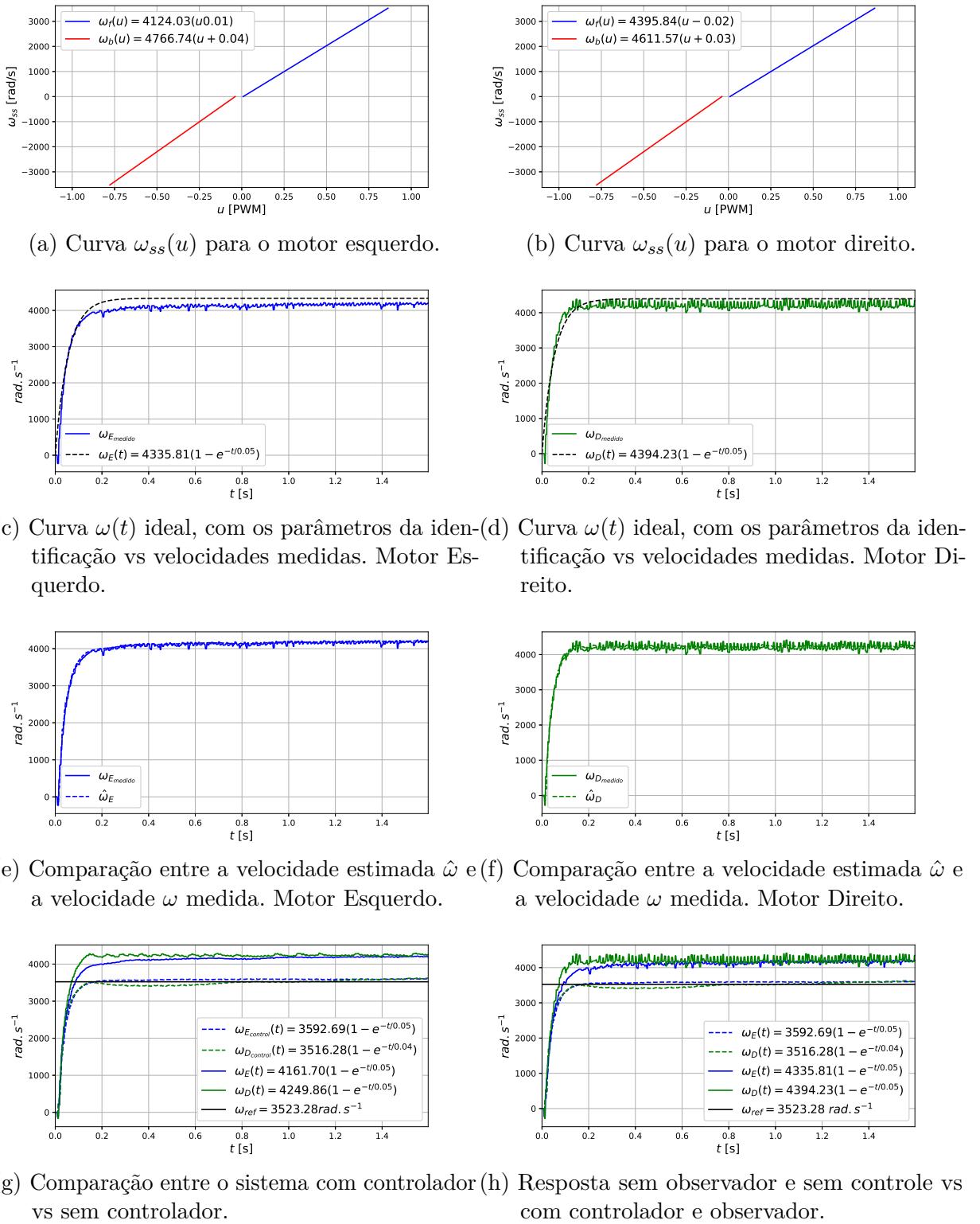


Figura 33 – Experimento 3. Sinal de controle e Referência igual a 1.

Os gráficos comparativos para o experimento 3 estão exibidos na Figura 33. A partir dessas Figuras e da Tabela 7 é possível fazer as mesmas observações que no experimento 1 e 2, com a diferença nas zonas mortas (que podem ser observadas nas Figuras 33a e 33b).

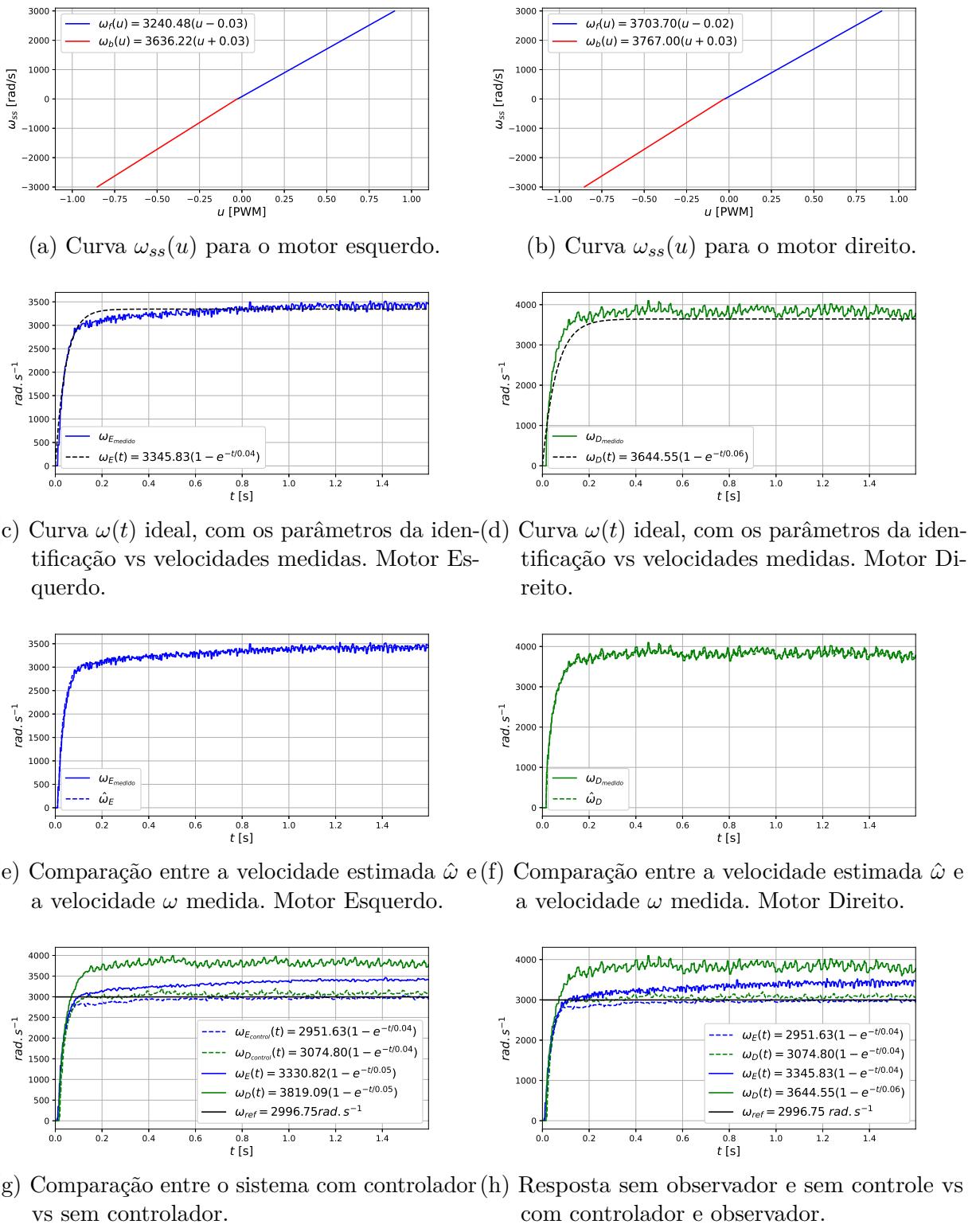


Figura 34 – Experimento 4. Sinal de controle e Referência igual a 1.

Os gráficos comparativos para o experimento 4 estão exibidos na Figura 34. A partir dessas Figuras e da Tabela 7 extraí-se que para o **Experimento 4**:

- Analisando as Figuras 34a e 34b:

As zonas mortas ( $|D|$ ) para o motor esquerdo e direito são respectivamente: 0,03 (para ambos os sentidos de giro), 0,02 e 0,03 (na rotação que favorece o movimento para frente do robô e para trás, respectivamente).

- Pelas Figuras 34c e 34d:

Observa-se que, as curvas ideais/teóricas (considerando o sistema de primeira ordem e com os parâmetros da calibração) conseguem representar bem o comportamento apresentado pela resposta dos motores esquerdo e direito.

- Pelos gráficos das Figuras 34e e 34f observa-se que:  
para ambos os motores a curva correspondente à velocidade  $\hat{\omega}$  estimada pelo filtro, sobrepoê-se à curva das velocidades medidas, porém, com um ruído bem menor.
- Já pela Figura 34g nota-se para ambos os motores: que a resposta do sistema controlado está bem próxima da referência  $\omega_{ref}$  e que a constante de tempo  $\tau$  está como o desejado (0,05 s).
- Por fim a Figura 34h compara a resposta do sistema com o controlador+observador, com os mesmos sem estes.

Nota-se que os conjuntos motor-rola esquerdo e direito apresentam assimetrias significativas, principalmente com relação ao ganho (o conjunto direito apresenta-se com um ganho maior). Porém a resposta controlada mostra que se conseguiu diminuir significativamente essas assimetrias nesse experimento.

# 5 CONCLUSÃO

Pelo exposto no Capítulo 4 podemos notar que:

- A aproximação do comportamento dos motores para um sistema de primeira ordem, mostrou-se ser suficientemente para todos os experimentos realizados (como esperado, conforme apresentado na seção 2.1);
- O uso do filtro de Kalman para estimar a velocidade de rotação dos motores, mostrou-se eficiente, pois conseguiu reduzir quase que completamente os erros de quantização, mesmo com medições provenientes de sensores de baixa resolução e operando em altas velocidades (pior condição para a leitura de *encoders* pelo método da medição de períodos);
- O controlador *Feedforward* em conjunto com o controlador proporcional (*Backward*) também mostrou bons resultados, conseguindo rastrear a referência com uma margem de erro relativamente baixa, além de atingir os tempos de respostas desejados (constante de tempo desejada) e contornar as pertubações externas (como mostrado principalmente no experimento 4 nas Figuras 34h).

Conclui-se portanto que o sistema proposto é capaz de oferecer boas estimativas para as velocidades dos motores, por meio do filtro de Kalman, mesmo para sensores com baixa resolução. E que o sistema de controle (fazendo uso dessas estimativas) é capaz de reduzir as assimetrias do par motor-rodas de (mas não limitando-se) robôs com acionamento diferencial. Além disso, o sistema proposto apresentou-se como uma solução simples o suficiente para conseguir ser implementado em um *hardware* com recursos limitados, como um microcontrolador.

Fazem-se necessárias também algumas observações importantes a respeito dos experimentos e de suas análises apresentados no capítulo anterior. São elas:

- A escolha de se trabalhar com as velocidades no eixo do motor antes da caixa de redução fez com que os valores em módulo dessas velocidades fossem altos, dificultando um pouco a análise visual dos gráficos, fazendo (por exemplo) com que erros absolutos no rastreio da referência sejam mascarados devido à escala dos gráficos.

- Outro ponto é com relação à variedade dos experimentos: os quatro experimentos mostram pouca diferença entre si. Provavelmente isso é devido ao fato de que a maioria deles foram realizados com os robôs suspensos (sem contato com o chão), apenas o experimento 4 foi feito com um robô em contato com o chão (situação de operação normal do robô). O motivo disso foram problemas técnicos encontrados durante a fase de experimentação e testes, os problemas técnicos fizeram com que dois dos três robôs disponíveis não conseguissem operar normalmente, limitando assim as possibilidades de testes.

O presente trabalho possui como objetivos futuros:

- A adição da ação integral (controlador PI) no sistema de controle, para eliminar o erro residual que o controlador *Feedforward* não está sendo capaz de eliminar;
- A calibração automática (dinâmica), principalmente do ganho do sistema, por meio da técnica de mínimos quadrados recursivos, como apresentado em alguns dos trabalhos descritos na seção 1.2.

# REFERÊNCIAS

- Brandt, S. The method of least squares. In: \_\_\_\_\_. *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*. Cham: Springer International Publishing, 2014. p. 209–265. ISBN 978-3-319-03762-2. Disponível em: <[https://doi.org/10.1007/978-3-319-03762-2\\_9](https://doi.org/10.1007/978-3-319-03762-2_9)>.
- Briz, F.; Cancelas, J. A.; Diez, A. Speed measurement using rotary encoders for high performance ac drives. In: *Proceedings of IECON'94 - 20th Annual Conference of IEEE Industrial Electronics*. [S.l.: s.n.], 1994. v. 1, p. 538–542 vol.1.
- Chui, C. K.; Chen, G. Kalman filter: An elementary approach. In: \_\_\_\_\_. *Kalman Filtering: with Real-Time Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 20–32. ISBN 978-3-540-87849-0. Disponível em: <[https://doi.org/10.1007/978-3-540-87849-0\\_2](https://doi.org/10.1007/978-3-540-87849-0_2)>.
- Equipe POTI. *TDP Equipe Poti de Futebol de Robôs da UFRN 2019*. 2019. <http://sistemaolimpo.org/mídias/uploads/a17801f6c003c72c7c9e0f8efd5c2d4e.pdf>. Acesso em: 3 dez. 2020.
- Gaolin Wang et al. Low speed control of permanent magnet synchronous motor based on instantaneous speed estimation. In: *2006 6th World Congress on Intelligent Control and Automation*. [S.l.: s.n.], 2006. v. 2, p. 8033–8036.
- Hori, Y. High performance control of servomotors with low precision shaft encoder using instantaneous speed observer and adaptive identification of inertia moment. In: *Proceedings 1993 Asia-Pacific Workshop on Advances in Motion Control*. [S.l.: s.n.], 1993. p. 7–12.
- Hori, Y. Robust and adaptive control of a servomotor using low precision shaft encoder. In: *Proceedings of IECON '93 - 19th Annual Conference of IEEE Industrial Electronics*. [S.l.: s.n.], 1993. p. 73–78 vol.1.
- IEEE League. *Regras IEEE Very Small Size Soccer (VSSS)*. 2020. [http://200.145.27.208/cbr/wp-content/uploads/2020/07/vssRegras\\_Portugues.pdf](http://200.145.27.208/cbr/wp-content/uploads/2020/07/vssRegras_Portugues.pdf). Acesso em: 3 dez. 2020.
- Kalman, R. E. A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, v. 82, n. Series D, p. 35–45, 1960.
- Mendes, E. P.; Medeiros, A. A. D. Identification of quasi-linear dynamic model with dead zone for mobile robot with differential drive. In: *2010 Latin American Robotics Symposium and Intelligent Robotics Meeting*. [S.l.: s.n.], 2010. p. 132–137.
- Ogata, K. *Engenharia de controle moderno*. PRENTICE HALL BRASIL, 2011. ISBN 9788576058106. Disponível em: <<https://books.google.com.br/books?id=iL3FYgEACAAJ>>.
- Okoro, I. Feedback-feedforward compensation of a dc motor. In: *2019 IEEE PES/IAS PowerAfrica*. [S.l.: s.n.], 2019. p. 385–389.

Petrella, R. et al. Speed measurement algorithms for low-resolution incremental encoder equipped drives: a comparative analysis. In: *2007 International Aegean Conference on Electrical Machines and Power Electronics*. [S.l.: s.n.], 2007. p. 780–787.

Phan Quoc Dzung; Le Minh Phuong. Control system dc motor with speed estimator by neural networks. In: *2005 International Conference on Power Electronics and Drives Systems*. [S.l.: s.n.], 2005. v. 2, p. 1030–1035.

Pololu. *Driver Motor*: Tb6612fng dual motor driver carrier. [S.l.]: Pololu Robotics and Eletronics. <https://www.pololu.com/product/713/resources>. Acesso em: 3 dez. 2020.

Pololu. *Magnetic Encoder*: Magnetic encoder pair kit for micro metal gearmotors, 12 cpr, 2.7-18v. [S.l.]: Pololu Robotics and Eletronics. <https://www.pololu.com/product/3081/specs>. Acesso em: 3 dez. 2020.

Pololu. *Pair, Magnetic Encoder*: Magnetic encoder pair kit for micro metal gearmotors. <https://www.pololu.com/product/3081>. Acesso em: 3 dez. 2020.

Ravi Kumar, K. S. et al. Microprocessor based closed loop speed control of dc motor using pwm. In: *2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*. [S.l.: s.n.], 2015. p. 255–257.

Sharan, R. V. et al. A simple speed feedback system for low speed dc motor control in robotic applications. In: SOBH, T. et al. (Ed.). *Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications*. Dordrecht: Springer Netherlands, 2007. p. 397–399. ISBN 978-1-4020-6266-7.

Shi, T.; Wang, Z.; Xia, C. Speed measurement error suppression for pmsm control system using self-adaption kalman observer. *IEEE Transactions on Industrial Electronics*, v. 62, n. 5, p. 2753–2763, 2015.

Sripad, A.; Snyder, D. A necessary and sufficient condition for quantization errors to be uniform and white. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 25, n. 5, p. 442–448, 1977.

Systems, E. *ESP-IDF Programming Guide*. Disponível em: <<https://docs.espressif.com/projects/esp-idf/en/stable/api-reference/index.html>>.

Systems, E. *ESP32-Datasheet*. [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf). Acesso em: 3 dez. 2020.

Systems, T. P. *Ne-ba931*. <http://www.futuramix.com.br/bateria-lipo-1s-37v-750mah-ne-ba931-para-solo-pro-180-helicoptero-pode-retirar/prod-4516975/>. Acesso em: 3 dez. 2020.

Taeg-Joon Kweon; Dong-Seok Hyun. High-performance speed control of electric machine using low-precision shaft encoder. *IEEE Transactions on Power Electronics*, v. 14, n. 5, p. 838–849, 1999.

Wikipedia contributors. *Kalman filter — Wikipedia, The Free Encyclopedia*. 2020. [Online; accessed 4-December-2020]. Disponível em: <[https://en.wikipedia.org/w/index.php?title=Kalman\\_filter&oldid=991999986](https://en.wikipedia.org/w/index.php?title=Kalman_filter&oldid=991999986)>.

- Wikipédia. *Método dos mínimos quadrados — Wikipédia, a encyclopédia livre*. 2020. [Online; accessed 19-novembro-2020]. Disponível em: <[https://pt.wikipedia.org/w/index.php?title=M%C3%A9todo\\_dos\\_m%C3%A3nimos\\_quadrados&oldid=59834895](https://pt.wikipedia.org/w/index.php?title=M%C3%A9todo_dos_m%C3%A3nimos_quadrados&oldid=59834895)>.
- Wu, H.; Chen, X.; Hu, L. Embedded system of dc motor speed control based on arm. In: *2008 ISECS International Colloquium on Computing, Communication, Control, and Management*. [S.l.: s.n.], 2008. v. 2, p. 123–126.
- Xia, C. et al. Two-degree-of-freedom proportional integral speed control of electrical drives with kalman-filter-based speed estimation. *IET Electric Power Applications*, v. 10, n. 1, p. 18–24, 2016.