

Desenvolvimento de um protótipo de robô móvel de baixo custo para práticas de ensino e pesquisa

Raphaell Maciel de Sousa ^[1], Gustavo Pereira Bruno ^[2], Gerberson Felix da Silva ^[3], Fábio Araújo de Lima^[4]

[1] raphaellmsousa@gmail.com. IFPB\Campus Cajazeiras/UNIND. [2] brunodf@gmail.com. IFPB\Campus Cajazeiras/GPCAR. [3] gerberson.felix@gmail.com. IFPB\Campus Cajazeiras/GPCAR. [4] fabioarsp@gmail.com. IFPB\Campus Cajazeiras/UNIND.

RESUMO

Nos últimos anos, a robótica tem despertado a atenção da comunidade docente e discente no âmbito do IFPB. A Instituição tem promovido diversas práticas abordando a robótica como tema central. Entre estas práticas, destacam-se as competições realizadas na Instituição ou em outros centros educacionais, nas quais as equipes são estimuladas a produzir robôs que resolvam alguns problemas propostos, inspirados em desafios práticos do cotidiano. Dessa forma, a robótica pode ser uma ferramenta prática que permite aplicar e integrar os conhecimentos abordados em sala de aula, desenvolvendo, além das habilidades técnicas, a capacidade de liderança, gerenciamento e trabalho em equipe – que são requisitos essenciais no mundo do trabalho. De modo geral, kits didáticos criados para desenvolvimento e ensino de robótica possuem um custo elevado; além disso, comumente, possuem tecnologia proprietária, de modo que as práticas são limitadas ao que é proposto pelos fabricantes. Nessa direção, este trabalho propõe o desenvolvimento de um robô móvel que possa ser utilizado para fins didáticos em estudos iniciais ou avançados de robótica autônoma. O robô desenvolvido possui sensores infravermelhos para detecção e desvio de obstáculos, medição de velocidade e um sistema de medição de coordenadas globais GPS. O robô é baseado nas plataformas *open-source*: Arduino e Raspberry Pi. Resultados de simulação e experimentais são apresentados para demonstrar o funcionamento do protótipo desenvolvido bem como suas potencialidades, nas quais podem-se destacar a detecção e desvio de obstáculos, navegação remota com *feedback* de imagens e controle de velocidade por meio de reguladores PID (Proporcional, Integrativo e Derivativo).

Palavras-chave: Robótica móvel. Arduino. Raspberry Pi. Controle. Kit didático.

ABSTRACT

In the last years robotics has attracted attention of professors and students at IFPB. The institution has promoted many practical activities related to robotics as a central theme, among these activities, stand-out robotics competitions at the institution or other educational centers, in which teams are encouraged to create robots that can solve some proposed problems inspired in practical daily challenges. In this sense, robotics can be a practical tool that enables people to apply and integrate knowledge learned in class, developing technical skills, leadership, management and work team abilities, for these are basic requirements to the labor market. In general, the didactic kits for the development and teaching of robotics are expensive. They also have proprietary technology, so, practical approaches are limited to proprietary architecture. Thus, this paper proposes the development of a mobile robot to be used for beginning or advanced teaching purpose of autonomous robotics. The robot developed has infrared sensors for detection and obstacles detour, it calculates velocity and has a Global Positioning System (GPS). The robot is based on open-source platforms: Arduino and Raspberry Pi. Experimental and simulation results are presented to demonstrate the working prototype and its potentialities. As for its qualities, we can highlight obstacle detection and detour, remote guidance with image feedback and control of velocity by using Proportional, Integral and Derivative (PID) controls.

Keywords: Mobile Robotics. Arduino. Raspberry Pi. Control.

1 Introdução

O ensino de engenharia baseado em projeto, ou metodologia ativa, é uma prática pedagógica adotada em muitas instituições de ensino técnico (COCOTA; D'ANGELO; MONTEIRO, 2015). Nesse sentido, o IFPB Campus Cajazeiras criou o Grupo de Pesquisa em Controle, Automação e Robótica – GPCAR. Esse grupo desenvolve atividades de ensino e pesquisa baseadas em projetos de robótica e automação. Esse tipo de metodologia difere do ensino tradicional, baseado na exposição de conteúdo teórico, muitas vezes sem conexão com atividades práticas.

Cursos baseados em projetos práticos estão cada vez mais em alta. Diversas plataformas *on-line*, tais como Coursera, Udacity, Edx, entre outras, estão sendo lançadas. Essas plataformas de ensino têm em comum a característica de serem baseadas; nelas, os estudantes possuem objetivos bem definidos e, ao final de cada curso, desenvolvem uma série de atividades aplicadas.

Nessa direção, a robótica tem despertado o interesse de docentes e pesquisadores, visto que pode ser uma ferramenta de auxílio pedagógico multidisciplinar. Embora se percebam os benefícios, estudar robótica no Brasil pode ter um custo relativamente alto, pois grande parte dos componentes eletrônicos utilizados são importados de outros países, de modo que o processo de compra está sujeito a tributações que aumentam o valor final dos dispositivos. Por outro lado, os recentes avanços nos processos de fabricação de dispositivos microeletrônicos têm contribuído para o desenvolvimento e difusão da robótica. Quanto ao *hardware*, este tem se tornado cada vez mais robusto, com melhora significativa no desempenho e preços mais acessíveis, como pode ser observado nas plataformas microcontroladas, como o Arduino, ou nos processadores com *Linux* embarcado, tais como os da *Raspberry* ou *Beaglebone* (RASPERRY PI FOUNDATION, 2015). No que diz respeito ao *software*, o desenvolvimento do *hardware* influencia diretamente no desempenho do robô, pois permite a aplicação de algoritmos complexos para reconhecimento de imagem, seguimento de trajetória e tomada de decisões (HWANG, 2016), que normalmente envolvem uma grande quantidade de processamento de dados.

Quando se trata de robótica, o tema é amplo, com diversas abordagens e diversas linhas de pesquisa relacionadas, tais como controle aplicado (LIU et al.,

2017), redes neurais (XIAO et al., 2017), visão computacional (SHENE; SRIDHARAN; SUDHA, 2016), entre outras áreas. O interesse com relação ao tema ultrapassa as salas de aula, de modo que a robótica, no nosso cotidiano, pode integrar diversas aplicações práticas, que vão desde a utilização de dispositivos teleoperados, utilizados em manutenções e vistorias de locais de difícil acesso, até robôs guia dentro de shoppings, por exemplo.

Com relação à robótica móvel, vários são os desafios relacionados, entre estes, muitas vezes, os robôs precisam seguir trajetórias definidas de modo autônomo (LIAN et al., 2016; LIANG et al., 2016); neste caso, trajetórias geralmente são dinâmicas e algumas soluções estão sendo investigadas para que o robô seja capaz de navegar de modo seguro (SHENE; SRIDHARAN; SUDHA, 2016).

Nessa direção, este trabalho apresenta um protótipo de robô móvel de baixo custo, que pode ser utilizado em sala de aula, auxiliando o aprendizado; além disso, o protótipo pode ser utilizado para validar estudos diversos de robótica móvel. Resultados experimentais que exploram as potencialidades do protótipo desenvolvido são apresentados.

2 Objetivos

O principal objetivo deste trabalho é inserir a robótica dentro do processo de ensino e aprendizagem. A robótica pode ser utilizada como um elemento motivador; além disso, os discentes podem aplicar os conhecimentos obtidos em sala de aula e comparar os resultados teóricos e experimentais. Entre os objetivos deste trabalho, podem ser destacados:

- Desenvolver trabalhos interdisciplinares, nos quais os discentes possam ser estimulados a aplicar conhecimentos de algoritmos, eletrônica, mecânica, entre outros, para resolver problemas reais;
- Realizar a simulação computacional e antecipar o comportamento do modelo real;
- Prover uma plataforma de baixo custo para o estudo de controle de processos, comunicação *wireless*, visão computacional, entre outras aplicações possíveis.

3 Prototipagem

O robô é dotado de um Arduino Due para processamento dos algoritmos de controle; uma *Raspberry Pi*, para processamento de imagens e comunicação *wireless*. Além de sensores infravermelhos para detecção de obstáculos e medição de distâncias, são utilizados *encoders* para medição da velocidade e deslocamento. Também possui uma placa GPS para informações sobre posição global que podem ser utilizadas em projetos de navegação autônoma.

3.1 Modelagem dinâmica

O protótipo desenvolvido neste trabalho é baseado no modelo unicycle (SUN et al., 2016), aqui adotado por conta de sua relativa simplicidade. As Equações 1-3 descrevem a dinâmica do robô unicycle.

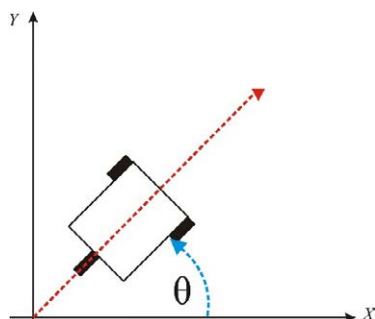
$$\dot{x} = R/2 (v_r + v_l) \cos(\Theta) \tag{1}$$

$$\dot{y} = R/2 (v_r + v_l) \sin(\Theta) \tag{2}$$

$$\dot{\Theta} = R/L (v_r - v_l) \tag{3}$$

em que R e L são o raio das rodas e o comprimento entre elas, respectivamente. Pode ser observado, na Equação 3, que a dinâmica da velocidade angular é função das variáveis R e L, de modo que, fixando-se o valor do comprimento L e aumentando-se o valor do raio R, a velocidade angular aumenta; o contrário ocorre, ou seja, fixando-se o valor de R e aumentando-se o valor do comprimento L, conseqüentemente diminui. As variáveis v_r e v_l correspondem à velocidade das rodas direita e esquerda, respectivamente. O ângulo Θ corresponde à orientação do robô, considerando um plano fixo de duas coordenadas (x,y), como pode ser observado na Figura 1.

Figura 1 – Sistemas de coordenadas



Fonte: Elaboração própria.

O deslocamento do robô é realizado a partir do sistema de coordenadas, apresentado na Figura 1. Sendo assim, é possível calcular o ângulo de deslocamento do robô por meio de operações simples de trigonometria, o que permite ao discente ter uma noção prática dessa área.

3.1.1 Modelo de Controle

O modelo de controle utilizado para o projeto dos reguladores de velocidade é baseado na dinâmica do motor CC com uma carga acoplada no eixo, simulando o deslocamento da massa do robô e o coeficiente de atrito com o robô em contato com a pista. Com base nisto, a seguinte função de transferência (FT) pode ser utilizada:

$$\frac{\Omega(s)}{E(s)} = \frac{k_m}{Ts + 1} \tag{4}$$

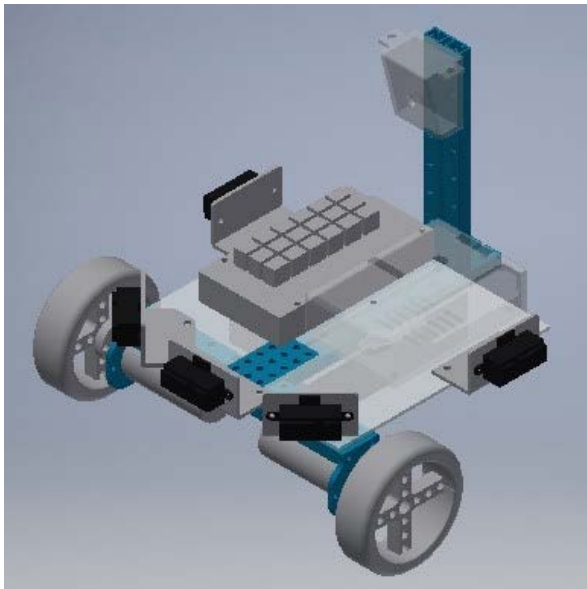
em que $\Omega(s)$ é a transformada de Laplace da velocidade angular $[\omega(t)]$, e T são o ganho e a constante de tempo do motor, respectivamente; E(s) é a transformada de Laplace da tensão de alimentação do motor CC. Para se determinarem as constantes e da FT, apresentada na Equação 4, foram realizados ensaios considerando a resposta ao degrau, que consiste na aplicação de uma tensão CC constante e determinação das constantes de tempo. Desse modo, a função de transferência obtida experimentalmente é determinada por:

$$\frac{\Omega(s)}{E(s)} = \frac{232.9}{0.28s + 1} \tag{5}$$

3.2 Mecânica

Na Figura 2, pode ser observado o modelo de robô unicycle proposto neste trabalho. A estrutura mecânica é baseada nos moldes das peças do kit da *makerbot*. O protótipo foi desenhado no *software inventor*, e as peças podem ser reproduzidas por meio de impressoras 3D. Outros kits de baixo custo podem ser utilizados para confeccionar o robô – as peças utilizadas também podem ser feitas em acrílico ou madeira, por exemplo. Para locomoção, foram utilizadas rodas comerciais de carros de rádio controle, com cerca de 3,5 cm de raio e uma distância de 17 cm de eixo dianteiro.

Figura 2 – Layout proposto para o robô desenvolvido



Fonte: Elaboração própria.

3.3 Hardware e software

Quanto ao *hardware* e *software* utilizados, todo o projeto é baseado em plataformas *open-source*. Desse modo, espera-se que os custos sejam reduzidos. Além disso, existe uma gama de conteúdo de suporte na internet que pode ser utilizado para auxiliar a compreensão de cada elemento de *hardware* e *software* que compõe o robô. A seguir, serão descritos os sensores escolhidos para o protótipo bem como o *Hardware* utilizado para execução dos algoritmos de controle e comunicação via web.

Sensores de proximidade – A detecção de proximidade confere ao robô a capacidade de determinar a distribuição espacial do ambiente ao seu redor. Esta detecção deve ocorrer sem o contato do robô com o obstáculo. Existem diversos tipos de sensores para este fim, tais quais sensores do tipo ultrassônico, indutivos, capacitivos ópticos, entre outros. Na Figura 3, pode ser observado o sensor SHARP 2Y0A21 F 5Y, que foi utilizado neste trabalho.

Figura 3 – Sensor infravermelho para detecção de obstáculos e medição de distâncias



Fonte: (SHARP, 2006)..

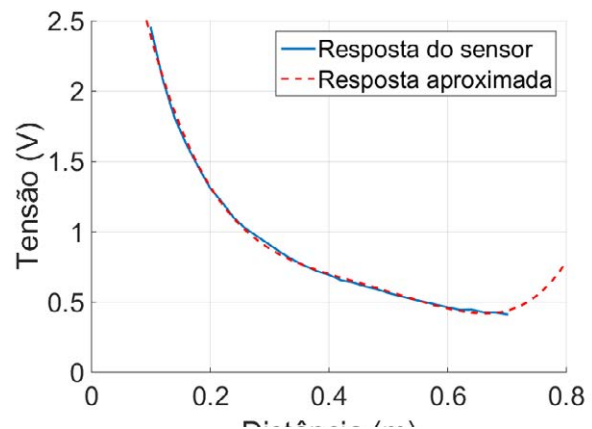
O sensor SHARP (Figura 3) é do tipo infravermelho e, de acordo com o seu *datasheet*, é adequado para medições entre 10 e 80 cm. Isto pode ser observado no gráfico apresentado na Figura 4. Neste gráfico, pode ser observado o comportamento da tensão de saída desse sensor em função da distância entre este e o obstáculo. Pode-se observar, a partir da Figura 4, que o comportamento da tensão de saída do referido sensor possui um comportamento não linear, logo, é necessário obter uma aproximação polinomial para realizar medição de distância de um modo mais aproximado.

$$p(x)=p_1 x^n+p_2 x^{(n-1)}+\dots+p_n x+p_{(n+1)} \quad (6)$$

em que

$$n=4, p_1=65, p_2=-120, p_3=90, p_4=-30 \text{ e } p_5=4,6.$$

Figura 4 – Tensão proporcional à distância medida pelo sensor SHARP 2Y0A21 F 5Y



Fonte: Elaboração própria.

Na Figura 5, apresenta-se a plataforma *Raspberry Pi*, que foi utilizada neste trabalho para processamento de dados e comunicação com a plataforma de controle. A filosofia de *softwares* abertos como o

Linux, por exemplo, tem se expandido, de modo que o conceito está sendo aplicado aos *hardwares* também. A plataforma de desenvolvimento *Raspberry Pi* é baseada na filosofia de *hardware* livre, de modo que sua estrutura é compacta, de baixo custo e é baseada em *Linux*, tornando possível interligar aplicações de *software* com *hardwares* de baixo nível. A plataforma *Raspberry Pi* é ideal para desenvolvimento e prototipagem rápida, de modo que traz consigo todas as vantagens e facilidades do *Linux* embarcado, permitindo a interface com diferentes *hardwares*. Entre as características do *Raspberry Pi*, destacam-se (RASP-BERRY PI FOUNDATION, 2015):

- Um processador quadre-core ARM Cortex-A7 de 900MHz;
- Memória de 1GB RAM;
- 4 portas USB;
- 40 pinos GPIO;
- HDMI;
- Porta Ethernet;
- Interface para câmera;
- Interface para display;
- Slot de cartão Micro SD;
- Processador gráfico.

Encoders – Este tipo de equipamento é utilizado para medição de velocidade e deslocamento. Os motores de corrente contínua, utilizados neste trabalho, possuem *encoders* acoplados ao eixo, de modo que cada volta do eixo do motor fornece uma quantidade de pulsos. Desse modo, o microcontrolador utiliza esta informação de contagem de pulsos para determinar a velocidade e o deslocamento do robô.

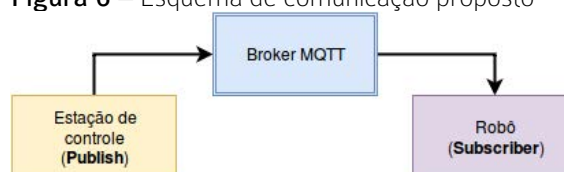
Figura 5 – Plataforma experimental desenvolvida



Fonte: Elaboração própria.

Sistema de comunicação – Na Figura 6, pode ser observado o sistema de controle e supervisão, utilizado neste trabalho para permitir o fluxo de dados entre o robô e a plataforma de supervisão e controle. Para tanto, foi utilizado o protocolo MQTT – que é responsável pelo envio de comandos de movimentação do protótipo. O referido protocolo é do tipo *Publish-Subscriber*. A informação com os comandos de controle é enviada pela central de controle (*Publisher*); em seguida, é recebida pelo *Broker* ou servidor MQTT; este, por sua vez, encaminha a mensagem para os clientes (*Subscriber*), que, neste caso, são os robôs conectados à rede. O pacote de dados do protocolo MQTT é bastante simples, contendo apenas os comandos de controle e o endereço do *Subscriber*, desse modo, a latência do MQTT é relativamente baixa, comparada a outros protocolos (KRAIJAK; TUWANUT, 2015).

Figura 6 – Esquema de comunicação proposto



Fonte: Elaboração própria.

A seguir serão descritos o protocolo de transmissão de imagens da câmera do robô para a estação de controle, bem como o envio de comandos de movimentação, com transmissão de dados baseada no protocolo MQTT (Figura 6):

a) *Envio de Imagens* – O envio de imagens foi realizado por meio do *software* MJPG-STREAMER. Este *software* converte cada frame em imagens com extensão JPEG compatíveis com *Linux-UVC* (classe de vídeo USB), via HTTP, para navegadores web, VLC e outros *softwares*. Após sua aquisição, as imagens ficam disponíveis para serem acessadas via rede, por meio do IP da *Raspberry Pi*. Na Figura 7, pode ser observada a tela com o *streamer* de vídeo da câmera embarcada na *Raspberry Pi*.

Figura 7 – Tela com *feedback* de imagem da câmera embarcada no robô móvel



Fonte: Elaboração própria.

b) *Envio de comandos* – O controle do protótipo é realizado enviando comandos específicos que habilitam a movimentação do robô por meio do acionamento dos motores de corrente contínua ou acionando os servomotores que compõem a garra manipuladora. Para tanto, um algoritmo de controle foi desenvolvido em *python*, utilizando-se a biblioteca *pygame*, que, por meio do teclado do computador cliente, capta a tecla pressionada, de modo que se essa tecla pressionada corresponder a alguma função de controle pré-definida, a função é executada pelo robô.

c) *Pacotes de dados enviados e recebidos*: Os pacotes de dados enviados e recebidos, tanto da estação de controle para o robô quanto do robô para a estação, são enviados para o servidor que, por sua vez, trata esses dados antes de passá-los para o destinatário final. Cada pacote de dados deve ser analisado de modo que, em cada ciclo de comunicação, o pacote contenha apenas um tipo de comando para o robô, caso contrário, o pacote será descartado e o servidor ficará aguardando novos dados. Isto ocorre pelo fato de os comandos serem enviados de modo serial – o objetivo desse protocolo, neste caso, é fazer com que o robô não execute funções indesejadas ou realize a mesma função diversas vezes em um único ciclo. Como exemplo desse inconveniente, podemos prever um usuário pressionar acidentalmente uma tecla de comando de movimentação do robô para a frente mais de uma vez em um mesmo ciclo de leitura de comandos. Isso poderia gerar um deslocamento do protótipo maior do que o desejado, fazendo o robô colidir em um obstáculo.

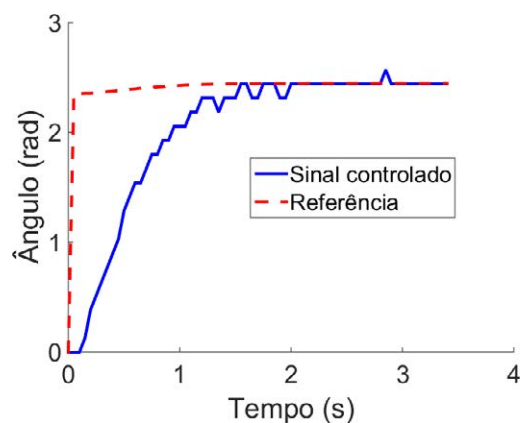
4 Resultados de simulação

A seguir, são apresentados resultados de simulação para diferentes situações dentro da navegação autônoma, tais como controle de velocidade, detecção e desvio de obstáculo, entre outras possibilidades. As simulações foram desenvolvidas na linguagem C e antecipam o comportamento do modelo uniclo (Equações 1-3).

4.1 Algoritmo *go-to-angle*

Neste esquema de controle, o robô deve ser capaz de se deslocar para a frente, com uma angulação determinada pelo usuário (ver Figura 1). Na Figura 8, é possível observar o resultado de simulação para esta estratégia de controle. Para tanto, um controlador PID é responsável por determinar a velocidade adequada das rodas esquerda e direita, de modo que o robô se mantenha na direção correta.

Figura 8 – Resultado de simulação para a estratégia de controle *go-to-angle*



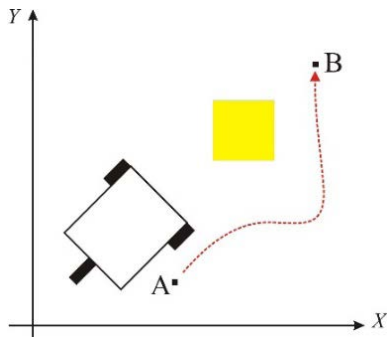
Fonte: Elaboração própria.

4.2 Algoritmo de desvio de obstáculos

Neste experimento, são utilizados os sensores infravermelhos para detecção e medição da distância entre o robô e o obstáculo a ser desviado. Na Figura 9, pode ser observado o processo de desvio de obstáculo. Nesta Figura, o robô precisa navegar de forma autônoma entre os pontos A e B. Desse modo, o ângulo de deslocamento é calculado com relação aos eixos x e y, e o robô inicia o plano de rota; em um dado momento, porém, o obstáculo em amarelo é detectado e o robô precisa recalcular a nova rota

para chegar ao ponto B e evitar a colisão. Isto pode ser observado pela linha pontilhada em vermelho.

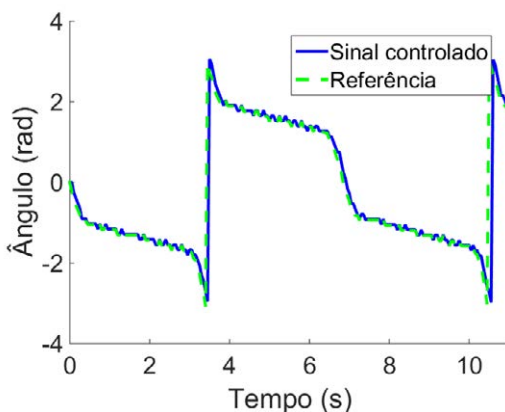
Figura 9 – Detecção e desvio de obstáculos



Fonte: Elaboração própria.

Na Figura 10, pode ser observado o resultado de simulação para o algoritmo *avoid obstacle*. Nesta figura, a linha azul representa o sinal controlado, ou seja, o ângulo em que o robô se movimenta (ver Figura 9). Pode-se observar que, aproximadamente no instante $t = 3,8s$, ocorre uma detecção de obstáculo e um novo ângulo de referência é calculado, de modo que o robô pode evitar o referido obstáculo. Pode-se observar, neste gráfico, que a linha pontilhada em verde, que representa a referência, está sendo seguida com erro nulo em regime permanente.

Figura 10 – Resultado de simulação para a estratégia de controle

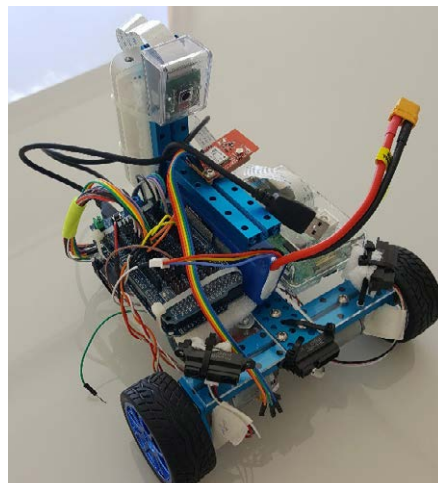


Fonte: Elaboração própria.

5 Resultados experimentais

Os resultados experimentais foram obtidos com o protótipo desenvolvido e apresentado neste trabalho (ver Figura 11). Com relação aos parâmetros de controle, tais como ganhos dos controladores, filtros digitais, entre outros, estes são totalmente acessíveis e flexíveis. Os resultados obtidos a seguir são baseados no desempenho de controladores Proporcional, Integrativo e Derivativo (PID), porém, outras estratégias e desempenhos dinâmicos podem ser obtidos. Para tanto, os ganhos utilizados para o controlador PID foram $K_p = 0.5$, $K_i = 1$ e $K_d = 0.01$. Outros ganhos podem ser utilizados, conseqüentemente, diferentes desempenhos dinâmicos podem ser obtidos e avaliados pelos discentes em sala de aula. Desse modo, após realizar simulações computacionais em sala de aula, será possível ajustar os parâmetros dos controladores e avaliar parâmetros de desempenhos, tais como: tempo de acomodação, máximo sobressinal, entre outras grandezas. O período de integração utilizado é de $T = 0.05s$. Para medição de velocidade e distância percorrida são utilizados encoders com 700 pulsos por volta de resolução. Os motores de corrente contínua funcionam com 12V de tensão de alimentação e velocidade de até 200 RPM (Rotações Por Minuto). A energia fornecida ao protótipo provém de uma bateria de lítio de 2200 mA/h. Além disso, é utilizada uma bateria extra de 5.000 mA/h para alimentar o Arduino Due e a *Raspberry Pi*. Foram realizados testes para o controle de velocidade do robô e para detecção e desvio de obstáculos.

Figura 11 – Plataforma experimental desenvolvida



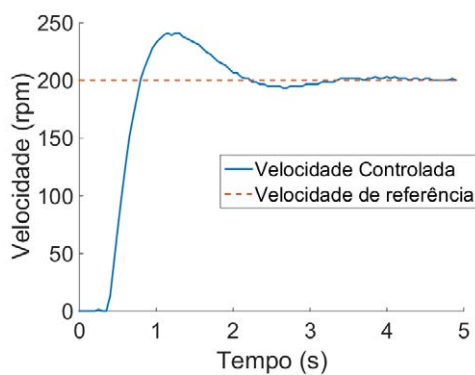
Fonte: Elaboração própria.

5.1 Controle de velocidade

O controle de velocidade do robô proposto é feito por meio do acionamento dos motores de corrente contínua, utilizando-se uma ponte H. Assim, um controlador Proporcional, Integral e Derivativo (PID) é utilizado para gerar as tensões de referência necessárias para regular a velocidade dos motores de corrente contínua.

Nas Figuras 12-13, podem ser observados os resultados experimentais para o controle de velocidade. Na Figura 12, a linha pontilhada corresponde a uma referência tipo degrau, com amplitude $Vel_{ref} = 200$ RPM.

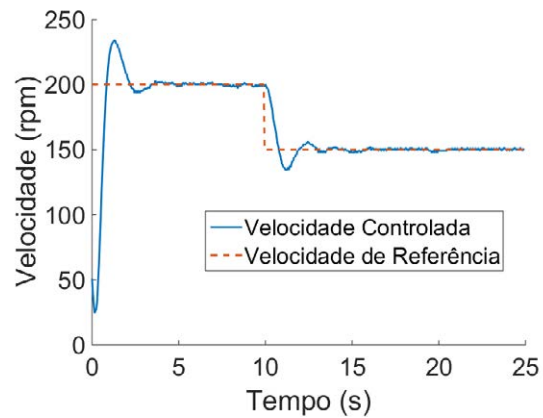
Figura 12 – Resultado experimental para o controle de velocidade



Fonte: Elaboração própria.

Na Figura 13, pode ser observado o resultado experimental para a variação de velocidade. Nesta figura, foi definida uma velocidade inicial de 200 RPM, como se vê na linha vermelha pontilhada. No instante $t = 10$ s, foi realizado um transitório de velocidade; a nova referência foi de 150 RPM. Pode-se observar, pela linha azul – que corresponde à velocidade controlada – que o controlador foi capaz de seguir a referência, com erro nulo em regime permanente. No instante da partida, porém, é possível observar que ocorre um sobre sinal de cerca de 30%. Isto pode ser melhorado a partir da sintonia do controlador PID. Esse pode ser um exercício desenvolvido em sala de aula.

Figura 13 – Resultado experimental para o controle de velocidade com variação de referência



Fonte: Elaboração própria.

5.2 Comunicação

Quanto à velocidade de transmissão de dados, o pacote de informações enviado do *publish* para o *subscriber* é relativamente pequeno para o caso em que o MQTT é utilizado. Desse modo, pelo tamanho da informação, a velocidade de transmissão ocorre em função da qualidade e velocidade da rede que, em funcionamento estável e baixa latência, apresenta resposta rápida na execução dos comandos. No caso da transmissão de imagens, no atraso de resposta, para uma internet de 5 Mb, observou-se um *delay* de transmissão de comandos com aproximadamente 0,65 segundos.

6 Trabalhos futuros

A plataforma proposta neste trabalho pode ser utilizada em experimentos para demonstrar estratégias de controle de velocidade e navegação autônoma, tal qual foi demonstrado nos resultados de simulação e experimentais.

A partir de agora, pretende-se estudar diferentes técnicas de controle que possam melhorar o desempenho dinâmico do robô. Além disso, serão estudados e implementados algoritmos de planejamento de trajetórias. Outra linha de pesquisa possível será a aplicação de técnicas de visão computacional aplicada à navegação autônoma.

7 Discussão

Ao longo deste trabalho, foram demonstrados diversos modos de operação do protótipo proposto. Existem outros modelos comerciais de robôs para fins didáticos, como é o caso do *khepera*, nas versões III e IV. O referido robô é dotado de um conjunto de equipamentos que possibilitam uma gama variada de aplicações. No Quadro 1, pode ser observado um comparativo entre o *khepera IV* e o protótipo desenvolvido. Pode-se observar que, em termos de recursos, o protótipo desenvolvido não possui acelerômetro e giroscópio, embora seja possível adicionar estes recursos, sem grandes alterações. A grande diferença está no quesito preço, o *khepera IV* custa em torno de R\$ 11 mil reais, enquanto que o protótipo proposto neste trabalho custa cerca de R\$ 1,5 mil reais, ou seja, pouco mais de 10% do custo total do *khepera*.

Quadro 1 – Comparação entre o robô proposto um modelo comercial

Característica	Khepera IV	Protótipo Proposto
Sistema Operacional	Linux	Linux
WiFi	Sim	Sim
Bluetooth	Sim	Sim
Acelerômetro	Sim	Não
Giroscópio	Sim	Não
Câmera	Sim	Sim
GPS	Não	Sim
Odometria	Sim	Sim

Fonte: Elaboração própria.

8 Conclusões

A plataforma experimental proposta neste trabalho é totalmente baseada em *software* e *hardware open-source*, desse modo, torna-se uma opção atrativa no que se refere ao custo. O investimento realizado foi de aproximadamente R\$ 1.500,00 para confeccionar o protótipo, enquanto um kit similar, como o *khepera*, custa em média 7 vezes mais. Com relação à aplicabilidade do robô, foi possível demonstrar, por meio de resultados de simulação e experimentais, que diversos estudos relacionados com robótica, controle de processos, comunicação wireless, entre outras áreas, podem ser realizados, de modo que o discente seja capaz de aplicar a teoria abordada

em sala de aula em um experimento prático. Nessa direção, o GPCAR (Grupo de Pesquisa em Controle, Automação e Robótica), do IFPB Campus Cajazeiras, tem desenvolvido um trabalho nesta linha, utilizando a robótica como elemento motivador para o ensino e pesquisa. As ações do grupo têm obtido resultados na comunidade, de modo que seus trabalhos têm sido apresentados em diversos eventos científicos e competições de robótica.

REFERÊNCIAS

- COCOTA. J. A. N.; D'ANGELO. T.; MONTEIRO. P. M. B. A Project-Based Learning Experience in the Teaching of Robotics. **IEEE Revista Iberoamericana de Tecnologias del Aprendizaje**, v. 10, n. 4, p. 302-309, 2015.
- HWANG. L. Comparison of Path Tracking Control of a Car-Like Mobile Robot With and Without Motor Dynamics. **IEEE/ASME Transactions on Mechatronics**, v. 21, n. 4, p. 1801-1811, 2016.
- KRAIJAK, S.; TUWANUT, P. A survey on IoT architectures, protocols, applications, security, privacy, real-world implementation and future trends. In: 11TH INTERNATIONAL CONFERENCE ON WIRELESS COMMUNICATIONS, NETWORKING AND MOBILE COMPUTING (WICOM 2015), p. 1-6, 2015.
- LIAN. C. et al. Near-Optimal Tracking Control of Mobile Robots Via Receding-Horizon Dual Heuristic Programming. **IEEE Transactions on Cybernetics**, v. 46, n. 11, p. 2484-2496, 2016.
- LIANG. X. et al. Leader-Following Formation Tracking Control of Mobile Robots Without Direct Position Measurements. **IEEE Transactions on Automatic Control**, v. 61, n. 12, p. 4131-4137, 2016.
- LIU. A. et al. Moving Horizon Estimation for Mobile Robots With Multirate Sampling. **IEEE Transactions on Industrial Electronics**, v. 64, n. 2, p. 1457-1467, 2017.
- RASPBERRY PI FOUNDATION. **Raspberry Pi 2 Model B**. Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>>. Acesso em: 27 mai. 2015.
- SHARP. Distance Measuring Sensor, GP2Y0A21YK0F datasheet. Dec. 2006.
- SHENE. T. N.; SRIDHARAN. K.; SUDHA. N. Real-Time SURF-Based Video Stabilization System for an FPGA-Driven Mobile Robot. **IEEE Transactions on Industrial Electronics**, v. 63, n. 8, p. 5012-5021, 2016.

SUN. W. et al. Two Time-Scale Tracking Control of Nonholonomic Wheeled Mobile Robots.

IEEE Transactions on Control Systems Technology, v. 24, n. 6, p. 2059-2069, 2016.

XIAO. H. et al. Robust Stabilization of a Wheeled Mobile Robot Using Model Predictive Control Based on Neurodynamics Optimization. **IEEE Transactions on Industrial Electronics**, v. 64, n. 1, p. 505-516, 2017.