



PLATAFORMA ROBÓTICA COM IMU UTILIZADA PARA PROTEÇÃO, DETECÇÃO DA INCLINAÇÃO E POSIÇÃO ANGULAR

Carlos Renato Borges dos Santos^{*1} e Ana Flávia Peixoto de Camargos²

¹IFTM campus Paracatu

²(NEPE) - IFMG campus Formiga

Resumo - Este trabalho tem como objetivo apresentar duas versões eletrônicas de uma plataforma robótica móvel, controlada por smartphone e projetada para percorrer terrenos desnivelados. A primeira versão utilizou-se o Arduino Mega 2560, o qual contém o módulo *Bluetooth*, bateria, ponte H, chassi e display LCD. Na segunda versão, utilizou-se com o módulo ESP32. A medição da inclinação foi realizada por uma IMU (*Inertial Measurement Unit*). Como resultados, utilizou-se o aplicativo *Inventor 2* para direcionar o veículo para frente, ré, esquerda, direita e parada. Além disso, foi possível programar previamente que o veículo paralisasse suas atividades, evitando assim o tombamento ao atingir uma certa inclinação crítica.

Palavras-Chave- Arduino, *Bluetooth*, IMU, robótica, veículo.

ROBOTIC PLATFORM WITH IMU USED FOR PROTECTION, SLOPE DETECTION AND ANGULAR POSITION

Abstract - This work aims to present two electronic versions of a mobile robotic platform, controlled by smartphone and designed to ride uneven terrain. The first version used the Arduino Mega 2560, which contains the Bluetooth module, battery, H bridge, chassis and LCD display. In the second version, the ESP32 module was used. The inclination measurement was performed by an IMU (*Inertial Measurement Unit*). As a result, the *Inventor 2* application was used to direct the vehicle forward, reverse, left, right and stopped. In addition, it was possible to pre-program the vehicle to stop its activities, thus avoiding tipping when reaching a certain critical inclination.

Keywords - Arduino, *Bluetooth*, IMU, robotic platform, vehicle.

I. INTRODUÇÃO

Projetos de robótica vêm sendo utilizados para estimular e complementar a formação de engenheiros [1]. No trabalho de [2], desenvolveu-se um robô capaz de seguir uma pessoa,

desde que esta envie suas coordenadas de GPS ao robô. Embora o GPS seja um sistema que informa a localização global, o erro das medições depende do número de satélites rastreados pelo aparelho, assim como da altitude e dos obstáculos físicos [3]. Para reduzir o erro de localização, pode-se utilizar filtro digital, tal como o Filtro de Kalman [4], que também é utilizado em sistemas de navegação com uma Unidade de Medição Inercial (IMU) [5, 6, 7]. Para uma IMU, a filtragem é necessária devido à natureza ruidosa desses sensores, principalmente os de baixo custo [7].

Dentro deste contexto existem inúmeros torneios envolvendo robótica como a competição de robôs, *soccer*, sumô, seguidor de linha, combate e *trekking* os quais utilizam microcontrolador, FPGA (*field-programmable gate array*), ou mesmo computador para o controle e acionamento dos veículos de forma autônoma ou remotamente [8, 9].

Este trabalho teve como objetivo desenvolver duas versões de plataforma robótica móvel, com chassi do tipo tanque com lagartas, que percorrem terrenos acidentados. Para o controle do veículo, foi desenvolvido um aplicativo com o *App Inventor 2*, do MIT, instalado em um smartphone com o sistema operacional Android. A comunicação sem fio adotada foi a *Bluetooth*.

II. MATERIAL E MÉTODOS

Para a construção das duas versões da plataforma robótica foram utilizados os materiais descritos a seguir.

A. Arduino

O Arduino é um microcontrolador desenvolvido com o intuito de simplificar a aplicação da eletrônica em projetos automatizados. Ele é uma plataforma de computação física de fonte aberta, com base em uma placa simples de entrada/saída. Ele contém uma conexão USB, o que permite sua ligação a um computador, além de diversos terminais que permitem a conexão com vários dispositivos externos, como motores, relés, sensores e outros [10].

O *hardware* constituiu-se de um dispositivo *open source* simples desenvolvido para um microcontrolador Atmel AVR de 8 bits. O mesmo pode ser aplicado para o desenvolvimento de

*carlosrenato@iftm.edu.br

objetos interativos independentes ou conectado a um computador, rede ou internet. Seus pinos podem ser programados para atuar como entrada ou saída de dados, conectados diretamente a sensores, displays ou shields.

O Arduino Mega 2560 foi utilizado para controlar os motores do chassi, receber os comandos do smartphone, bem como receber e processar os dados de inclinação. Ele pode ser visto na Figura 1.

Figura 1: Arduino Mega 2560.



B. Chassi

Foi utilizado o chassi ilustrado na Figura 2, o qual contém as seguintes dimensões: 15 x 31 x 7,6 cm (Largura X Comprimento X Altura). Ele possui também dois motores para as lagartas, os quais são utilizados para mover o tanque para frente, para trás, para a direita e para a esquerda, trabalhando de forma independente.

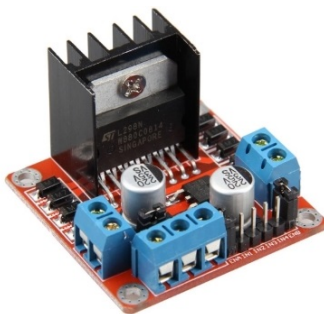
Figura 2: O chassi utilizado.



C. Ponte H

A ponte H é o *drive* de potência que controla os motores do chassi a partir dos sinais oriundos do microcontrolador Arduino. A ponte H é ilustrada na Figura 3.

Figura 3: Ponte H para os dois motores do chassi.



D. MPU 6050

Este módulo é uma IMU que é utilizado para fornecer os dados de aceleração e de giroscópio nos 3 eixos [11]. Sua in-

terface de comunicação é I2C. Para a utilização do MPU 6050, foi necessário o desenvolvimento de uma biblioteca em C++ para converter os valores lidos pelos acelerômetros, dos três eixos de rotação, em ângulos longitudinal e transversal. Este módulo é ilustrado na Figura 4.

Figura 4: Unidade de Medição Inercial.



Para a filtragem dos sinais da IMU foi utilizado o filtro passa-baixas frequências (também conhecido por Filtro Complementar), devido à sua simplicidade de cálculo e ao reduzido tempo de processamento. Ele pode ser implementado pela equação 1 [4, 5, 7].

$$x_{i+1} = \alpha \cdot \bar{x} + (1 - \alpha) \cdot x_i \quad (1)$$

Onde:

- α - Parâmetro de filtragem, entre 0 e 1.
- \bar{x} - Valor acumulado.
- x_i - Valor medido na iteração i .

A detecção das inclinações lateral e longitudinal podem ser realizadas pela equação 2, equação 3 e pela equação 4.

$$D = \sqrt{A_x^2 + A_y^2 + A_z^2} \quad (2)$$

$$\alpha = \frac{180}{\pi} \cdot \text{asin} \left(\frac{A_x}{D} \right) \quad (3)$$

$$\beta = \frac{-180}{\pi} \cdot \text{asin} \left(\frac{A_y}{D} \right) \quad (4)$$

Onde:

- A_x - Aceleração do eixo x .
- A_y - Aceleração do eixo y .
- A_z - Aceleração do eixo z .
- D - Vetor resultante das acelerações.
- α - Inclinação longitudinal, em graus.
- β - Inclinação lateral, em graus.

E. Módulo Bluetooth

Este módulo realiza a comunicação entre a plataforma robótica e o *smartphone*, o qual pode ser ilustrado na Figura 5.

Figura 5: Módulo *bluetooth*.



F. Bateria selada

A bateria utilizada nesta pesquisa foi de 1,3Ah com tensão de 12V. Ela alimenta todos os dispositivos da plataforma robótica e pode ser visualizada na Figura 6.

Figura 6: Bateria selada.



G. LCD gráfico

O LCD selecionado foi o Nokia 5110, ilustrado na Figura 7. Este foi escolhido em razão do seu preço, dimensões físicas e pela praticidade. A biblioteca desse componente por ser encontrada em [12].

Figura 7: LCD gráfico.



O LCD foi inserido no projeto para permitir informar ocorrências, como por exemplo a parada do veículo por inclinação, informando o motivo pelo qual ocorreu a parada. Assim, neste caso, exclui-se, por exemplo, a possibilidade da parada por mau contato, descarga da bateria ou mesmo erros de algoritmo.

H. App Inventor 2

Este é uma plataforma *online*, criada pelo MIT, a qual é capaz de criar aplicativos sem um código de programação que utiliza blocos construtivos. A interface do App Inventor 2 pode ser ilustrada na Figura 8.

Figura 8: O aplicativo utilizado em *smartphone*.



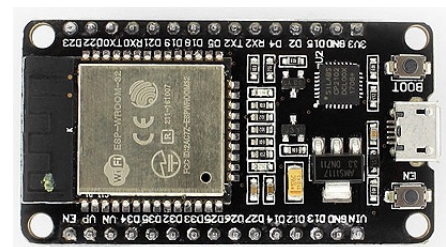
Pela figura acima pode-se visualizar o botão para iniciar a conexão com o módulo *Bluetooth*. Nesta interface foram inseridos cinco botões para direcionar o veículo para frente, ré, esquerda, direita e parada. Outros dois botões foram inseridos para aumentar e reduzir a velocidade dos motores.

I. ESP32

O módulo ESP32 é de alta performance para aplicações que utilizam *wi-fi* e baixo consumo de energia. Ele permite criar variadas aplicações para projetos de Internet das Coisas (IoT), acesso remoto, *webservers* e *dataloggers*, dentre outros [13].

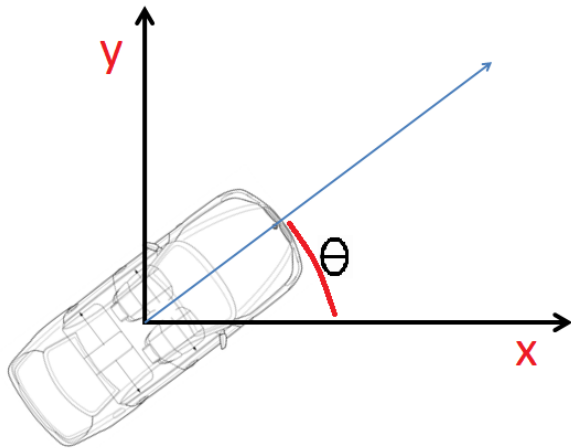
O ESP32 foi utilizado nesta pesquisa em sua versão 2, o qual substituiu o Arduino Mega 2560 e o módulo *Bluetooth*. Isso ocorreu pois o ESP32 contém internamente essas tecnologias e reduz espaço, uma vez que o Arduino Mega 2560 e o módulo *Bluetooth* ocupam mais que o dobro do volume do ESP32.

Figura 9: O ESP32-dev-kit.



Devido ao fato do ESP32 ter dois núcleos, o primeiro ficou reservado para a comunicação *Bluetooth*, ao passo que o segundo ficou responsável pela integração necessária por parte do giroscópio. Dessa forma, foi possível realizar o cálculo de integração do giroscópio, o que permitiu a determinação da posição angular do veículo em um sistema plano *x, y*, como visto na Figura 10.

Figura 10: Coordenadas x, y e θ .

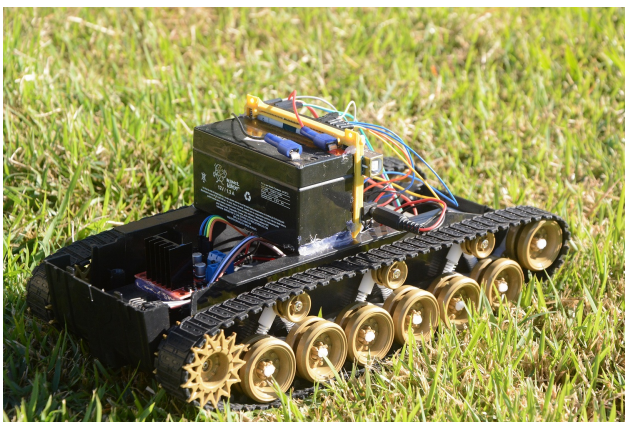


III. RESULTADOS

Foram desenvolvidas duas versões eletrônicas de uma plataforma robótica móvel, controlada por *smartphone* e projetada para percorrer terrenos desnivelados. A primeira versão foi desenvolvida utilizando o Arduino Mega 2560, o qual contém o módulo *Bluetooth*, bateria, ponte H, IMU, chassi e *display* LCD. O *display* foi utilizado para auxiliar na visualização dos dados de inclinação, do valor do PWM e do estado do veículo. Na segunda versão, retirou-se o Arduino Mega 2560, o módulo *bluetooth* e o LCD, trocando todos esses dispositivos pelo módulo ESP32.

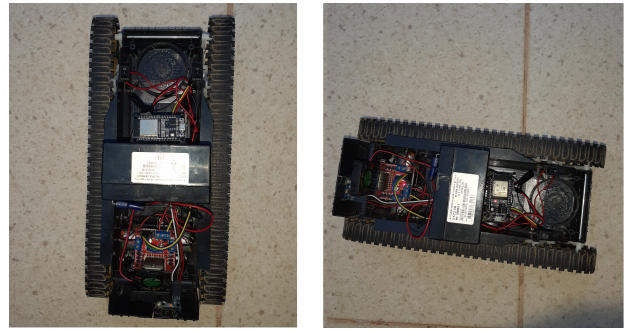
Na primeira versão, o processamento dispendido para escrever no LCD limitou os cálculos de integração do giroscópio, mas permitiu desenvolver outras partes do algoritmo, como por exemplo, a detecção da inclinação máxima, impedindo o tombamento do veículo. A primeira versão pode ser visualizada na Figura 11.

Figura 11: Tanque andando em terreno gramado.



Na segunda versão, foi possível realizar a integração do giroscópio, permitindo um giro automático de 90 graus, como visto na Figura 12. Durante as medições, constatou-se que houve um erro sistemático, fazendo com que o veículo ultrapasse um pouco mais de 90 graus.

Figura 12: Giro de 90 graus medido pelo giroscópio.



A integração pode ser obtida pela equação 5.

$$\begin{cases} gyro_X = gyro_X + \frac{(\theta_X - bias_X)}{t} \\ gyro_Y = gyro_Y + \frac{(\theta_Y - bias_Y)}{t} \\ gyro_Z = gyro_Z + \frac{(\theta_Z - bias_Z)}{t} \end{cases} \quad (5)$$

Onde:

- t - tempo decorrido entre os cálculos.
- $gyro_X$ - posição angular do eixo X.
- $gyro_Y$ - posição angular do eixo Y.
- $gyro_Z$ - posição angular do eixo Z.
- θ - velocidade angular de cada eixo.

IV. CONCLUSÕES

O projeto de uma plataforma robótica com IMU (Unidade de Medição Inercial) apresentou duas versões controlada por *smartphone* e projetada para percorrer terrenos desnivelados, sendo, a segunda, uma versão melhor na forma de realizar tarefas distintas.

A utilização de um *display* LCD forneceu dados relevantes para o desenvolvimento do sistema e contribuiu de forma significativa para a finalização do projeto. O aplicativo Inventor 2 foi utilizado para direcionar o veículo para frente, ré, esquerda, direita e parada. Além disso, foi possível programar previamente a parada do veículo, evitando assim o tombamento, ao atingir uma inclinação crítica.

A ponte H apresentou um aquecimento acima do esperado quando utilizada por períodos prolongados, sugerindo então que a mesma seja substituída em outra pesquisa.

Com o uso da IMU, foi possível verificar a inclinação lateral e longitudinal do veículo, além de permitir conhecer a posição angular do mesmo. Por outro lado, o projeto deste trabalho ainda precisa de mais ajustes e pesquisas para identificar a origem do erro sistemático nos cálculos de integração do giroscópio, pois o veículo ultrapassou um pouco o ângulo de 90 graus ao realizar o giro.

REFERÊNCIAS

- [1] T. S. Rodrigues, I. G. S. Fernandes, F. A. C. Oliveira, S. R. J. Oliveira. "Implementação de uma rede de sensores sem fio no torneio universitário de robótica". *XVI CEEL*, 2018.

- [2] A. F. R. Costa, H. E. Magalhães, J. P. O. Fornari, V. T. Santos, A. M. M. Medeiros, C. H. Fujisawa, B. Q. Oliveira. "Robô seguidor de pessoas controlado por arduino baseado em coordenadas geográficas e comunicação via bluetooth", *XVI CEEL*, 2018.
- [3] L. N. Couto. *Sistema para localização robótica de veículos autônomos baseado em visão computacional por pontos de referência*. 2012. Tese de Doutorado. Universidade de São Paulo.
- [4] P. Kim. *Kalman filter for beginners: with MATLAB examples*. CreateSpace, 2011.
- [5] V. H. R. Machado, K. G. Júnio, G. S. Faria. "Sistema de navegação inercial baseado em IMU". *XVI CEEL*, 2018.
- [6] W. S. Oliveira, E. N. Gonçalves. "Implementação em c: filtro de kalman, fusão de sensores para determinação de ângulos". *ForScience*, v. 5, n. 3, 2017
- [7] W. G. Aguas, K. R. G. Silva. "Aplicação de filtro de Kalman em acelerômetros e giroscópios". *XV CEEL*, 2017.
- [8] Winter Challenge. Acessado em 12 de julho de 2018, em: www.robocore.net/eventos/wc14.
- [9] W. G. Auas, K. R. Silva. "Projeto e implementação de um robô jogador de futebol da categoria IEEE very small size soccer". *XV CEEL*, 2017.
- [10] S. Monk. *Programação com Arduino II: Passos avançados com sketches*. Bookman Editora, 2015. (Tekne). ISBN 9788582602973. Disponível em: books.google.com.br/books?id=yC6UBQAAQBAJ.
- [11] MPU 6050. Datasheet. Acedido em 13 de junho de 2019, em: pdf1.alldatasheet.com/datasheet-pdf/view/1132807/TDK/MPU-6050.html.
- [12] Biblioteca LCD Nokia 5110. Acessado em 12 de junho de 2019, em: github.com/adafruit/Adafruit-GFX-Library.
- [13] ESP32. Módulo ESP32. Acessado em 28 de setembro de 2020, em: www.filipeflop.com/produto/modulo-wifi-esp32-bluetooth/.