

Desenvolvimento de um Software para Bancada de Teste do Sistema de Freio Antitravamento

Development of a Brake System Test Bench Software anti-lock

Ronnis Tarles Dantas Costa^{1*}, Dorgival Albertino da Silva Junior¹

RESUMO

Este trabalho descreve o desenvolvimento de um software para bancada de teste destinada a sensores de velocidade da roda do sistema de freio antitravamento. O gerenciamento da mesma deve ser feito por um Raspberry Pi e os valores de controle são calculados por um aplicativo desenvolvido na plataforma MIT App Inventor 2. A interface gráfica do software foi construída na plataforma QT Designer e posteriormente integrada a um código Python.

Palavras-chave: Bancada de teste; Sensores de velocidade; Software.

ABSTRACT

This work describes the development of a software for a test bench intended for anti-lock brake system wheel speed sensors. Its management must be done by a Raspberry Pi and the control values are calculated by an application developed on the MIT App Inventor 2 platform. The software's graphical interface was built on the QT Designer platform and later integrated into a Python code.

Keywords: Test bench; Speed sensors; Software.

¹ Universidade Federal Rural do Semi-Árido.

*E-mail: ronnis.costa@alunos.ufersa.edu.br

INTRODUÇÃO

Um veículo equipado com sistema de freio antitravamento (Anti-Lock Brake System (ABS)), ao está em movimento, faz com que o sensor de velocidade localizado na roda produza uma onda, em que a sua frequência é proporcional a sua velocidade (FERNÁNDEZ et al., 2021; PECOLT et al., 2021). Com o desenvolvimento de um software, é possível controlar uma bancada que contém um motor de passo capaz de girar a roda fônica da qual o sensor faz leitura, e assim, simular a velocidade da roda com o veículo parado.

O gerenciamento da bancada deve ser feito através de um computador de placa única, chamado Raspberry Pi. Esse computador possui um Sistema Operacional (SO) denominado Raspbian, contendo recursos semelhantes a um computador convencional, como processador, memória e drive gráfico (MUDALIAR; SIVAKUMAR, 2020). Esse minicomputador tem sido utilizado por diversos autores como plataforma de desenvolvimento, por ser uma ferramenta versátil e de baixo custo, além de ser compatível com diferentes linguagens de programação (URREA; KERN, 2021).

Uma dessas linguagens é a Python, que possui código aberto e compatibilidade com diversos sistemas operacionais, como Linux, macOS e Windows (MCKELVEY; GIBANICA, 2021). O Python vem crescendo em popularidade por ser uma linguagem bem documentada, suportar programação orientada a objetos e apresentar diversos pacotes disponíveis, além de ser mais fácil de ler e usar do que linguagens como C, C++ e Fortran (CHING et al., 2022; DOMÍNGUEZ et al., 2021). Diante desse conjunto de fatores, a linguagem de programação Python foi escolhida para desenvolver o software.

Para facilitar a utilização do software pelo usuário foi desenvolvida uma interface gráfica na plataforma QT Designer. Essa plataforma permite a criação de interfaces através de elementos prontos como botões e caixas de texto onde se pode configurar tamanhos, formas e cores desses elementos. Com a área de interação já pronta, é possível salvá-la na forma de um arquivo e chamá-lo dentro do código escrito em Python.

Dessa forma, o usuário pode utilizar facilmente a bancada de teste para sensor de velocidade da roda do ABS. Permitindo assim, usar a bancada em conjunto com outros equipamentos de diagnóstico automotivo como scanner e osciloscópio, de forma a coletar

um maior conjunto de dados para que se possa chegar a um diagnóstico mais preciso e em menos tempo. Dessa forma, tem-se que o tempo de serviço é maximizado e, com isso, o cliente passa a ter seu veículo à disposição em menos tempo.

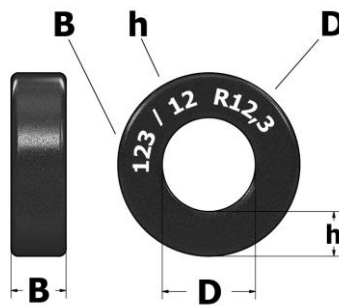
MATERIAIS E MÉTODOS

Cálculo da frequência de comando do motor

Para calcular as frequências de comando equivalentes a um valor de velocidade simulada em km/h, foram montadas algumas equações. Onde as mesmas usam parâmetros encontrados no próprio pneu do veículo, a velocidade a ser simulada, o número de passos por volta do motor e a relação entre o número de dentes das polias do sistema mecânico.

Os parâmetros do pneu podem ser obtidos de acordo com a Figura 1, onde B é o valor da banda de rodagem do pneu em milímetros, h é o percentual da banda de rodagem que corresponde a altura e D o valor do diâmetro interno em polegadas.

Figura 1 - Parâmetros do pneu



Fonte: Autoria própria

Esses parâmetros são utilizados para determinar o valor do raio (R_p) do pneu em metros e posteriormente usar esse raio para obter o perímetro (C_p), de acordo com as Equações 1 e 2.

$$R_p = \left(\frac{Bh}{100} \frac{25,4D}{2} \right) 10^{-3} \quad (1)$$

$$C_p = 2\pi R_p \quad (2)$$

Para determinar o número de Rotações Por Minuto (RPM) do motor de passo, é utilizado o próprio valor de velocidade que se deseja simular, o perímetro do pneu e a relação entre as polias do sistema mecânico. Onde V_s é a velocidade a ser simulada em km/h, P_b é o número de dentes da polia da bandeja giratória da roda fônica e P_m o número de dentes da polia acoplada ao motor. Dessa forma, o valor de RPM (ω_m) para uma dada velocidade pode ser obtido de acordo com a Equação 3.

$$\omega_m = \left(\frac{V_s \frac{10^3}{60}}{C_p} \right) \frac{P_b}{P_m} \quad (3)$$

Com o valor de RPM e o número de passos do motor, pode-se calcular a frequência de comando F_c em Hertz do sinal PWM (Pulse Width Modulation) correspondente a velocidade que se deseja simular, de acordo com a Equação 4. Onde n é o número de passos do motor.

$$F_c = \frac{\omega_m}{60} n \quad (4)$$

Para um eventual ajuste na precisão da frequência de controle do motor foram desenvolvidas duas novas equações. Essas usam o valor de velocidade que se deseja simular (V_s), o valor da frequência de controle encontrado inicialmente (F_c), a velocidade encontrada no scanner no momento de um teste prático (V_{scan}). A primeira delas, Equação 5, é usada pelo aplicativo quando o valor de velocidade simulada é menor que o desejado, já a segunda, Equação 6, é usada quando o valor de velocidade simulada é maior que o desejado.

$$F_{Ajus} = \frac{V_s \cdot F_c}{V_{scan}} + \left| \left(\frac{V_s \cdot F_c}{V_{scan}} - F_c \right) 0,05 \right| \quad (5)$$

$$F_{Ajus} = \frac{V_s \cdot F_c}{V_{scan}} - \left| \left(\frac{V_s \cdot F_c}{V_{scan}} - F_c \right) 0,05 \right| \quad (6)$$

Aplicativo para cálculo das frequências de comando do motor de passo

A plataforma online MIT App Inventor é um ambiente destinado ao desenvolvimento de aplicativos, onde originalmente era fornecido pelo Google e atualmente mantido pelo Massachusetts Institute of Technology (MIT) (SANGEETHALAKSHMI et al., 2021). O MIT App permite que até mesmo iniciantes

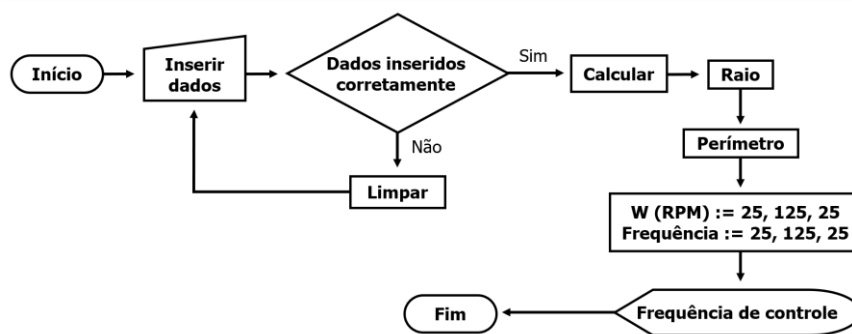
na programação criem aplicativos para os sistemas operacionais Android e IOS (SANGEETHALAKSHMI et al., 2021).

Com o MIT App Inventor 2 foi desenvolvido um aplicativo para calcular as frequências de comando do motor de passo. Essa plataforma permite o desenvolvimento de aplicativos através de blocos que podem ser arrastados e soltos no local desejado (AMORAN et al., 2021).

As Equações 1, 2, 3, 4, 5 e 6 foram utilizadas na programação do aplicativo. Essas foram inseridas na forma de blocos, de modo que ao clicar no botão (botão calcular) da tela inicial calculam-se as frequências correspondentes aos valores de velocidades desejados e as mostra na tela. Ao pressionar o mesmo botão, os valores inseridos pelo usuário são limpos para um novo preenchimento, caso seja necessário.

O veículo Accelo 815 da Mercedes-Benz foi usado como exemplo para o cálculo no aplicativo. O pneu do mesmo apresenta parâmetros $B = 215$, $h = 75$ e $R = 17,5$, além disso, as velocidades de simulação escolhidas foram de 25 a 125 km/h com passos de 25 km/h. A Figura 2 ilustra o fluxograma de execução do aplicativo.

Figura 2 - Fluxograma de execução do aplicativo para o cálculo da frequência



Fonte: Autoria própria

A programação em blocos da primeira tela do aplicativo apresenta quatro trechos. O primeiro deles ilustrado na Figura 3 contém a inicialização das variáveis R, C e W que representam respectivamente o raio do pneu, o perímetro do pneu e o valor de RPM do motor de passo.

Figura 3 - Trecho A da programação em blocos (tela 1)

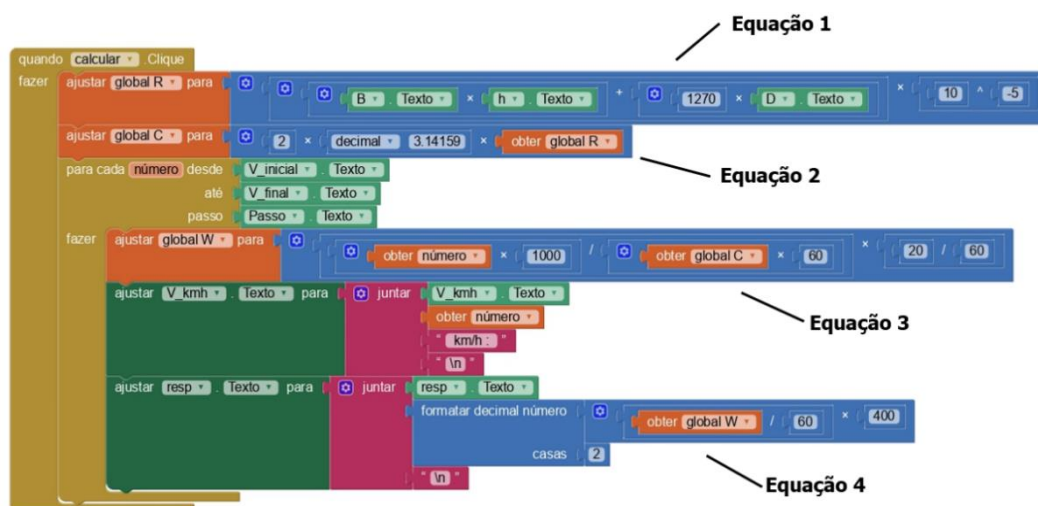


Fonte: Autoria própria

No trecho B ilustrado pela Figura 4, encontram-se as equações utilizadas no cálculo das frequências de comando do motor. De forma que ao clicar no botão para calcular as frequências, as Equações 1 e 2 são utilizadas uma única vez para obter os valores do raio e do perímetro do pneu. Já as Equações 3 e 4 são utilizadas para encontrar o valor de RPM do motor e a frequência de comando para as velocidades determinadas pelo laço de repetição.

O laço de repetição calcula os valores de RPM e posteriormente os valores de frequências de comando de 25 a 125 km/h com intervalos de 25 km/h. Onde os resultados exibidos na tela do aplicativo serão os valores das velocidades a serem simuladas e as frequências de comando do motor correspondentes a essas velocidades.

Figura 4 - Trecho B da programação em blocos (tela 1)



Fonte: Autoria própria

No trecho C ilustrado pela Figura 5 do código é responsável por limpar os valores inseridos pelo usuário quando o mesmo deseja efetuar um novo cálculo ou errou no

preenchimento dos dados de entrada, onde basta pressionar o mesmo botão usado para calcular por alguns instantes.

Figura 5 - Trecho C da programação em blocos (tela 1)



Fonte: Autoria própria

O trecho D ilustrado na Figura 6 é responsável por abrir a segunda tela do aplicativo. Esta, é responsável por fazer o reajuste da frequência de controle caso o usuário julgue necessário.

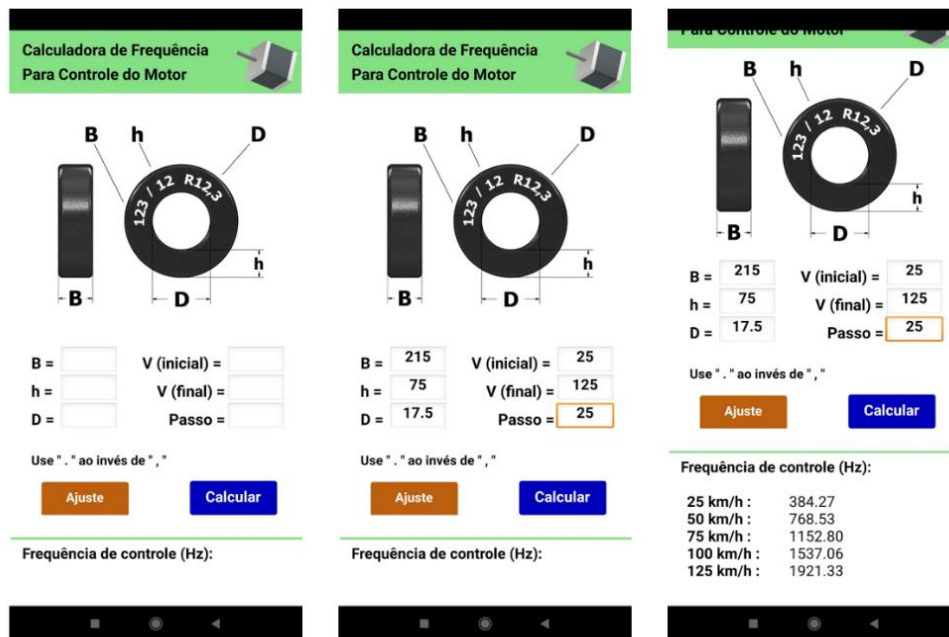
Figura 6 - Trecho D da programação em blocos (tela 1)



Fonte: Autoria própria

A interface de usuário da primeira tela do aplicativo é constituída por quatro partes principais. Sendo elas uma imagem ilustrativa que mostra de onde se deve retirar os dados do pneu, seis caixas de entradas para receber os dados do pneu e das velocidades, dois botões, onde um deles é para calcular as frequências e para limpar os valores preenchidos e o outro direciona a segunda tela responsável pelos ajustes de frequências, além de um espaço destinado à exibição dos valores na forma de lista, conforme ilustrado na Figura 7.

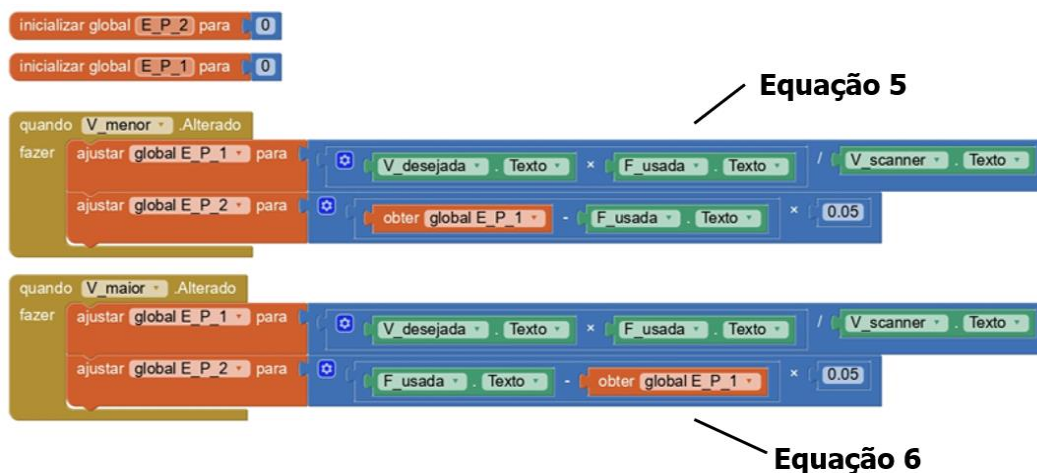
Figura 7 - Tela inicial do aplicativo desenvolvido



Fonte: Autoria própria

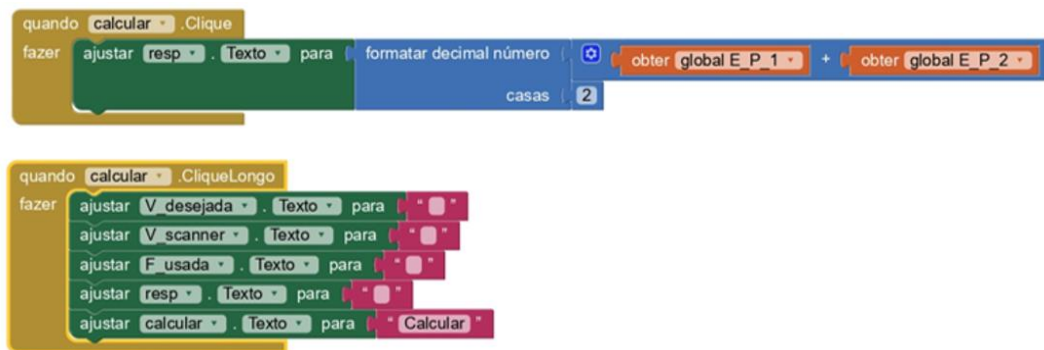
De forma análoga ao que foi feito na construção da tela inicial do aplicativo foram inseridas as Equações 5 e 6 na programação em blocos. A Figura 8 e 9 ilustram a programação em blocos da tela de ajuste e a Figura 10 o fluxograma de execução.

Figura 8 - Trecho E da programação em blocos (tela 2)



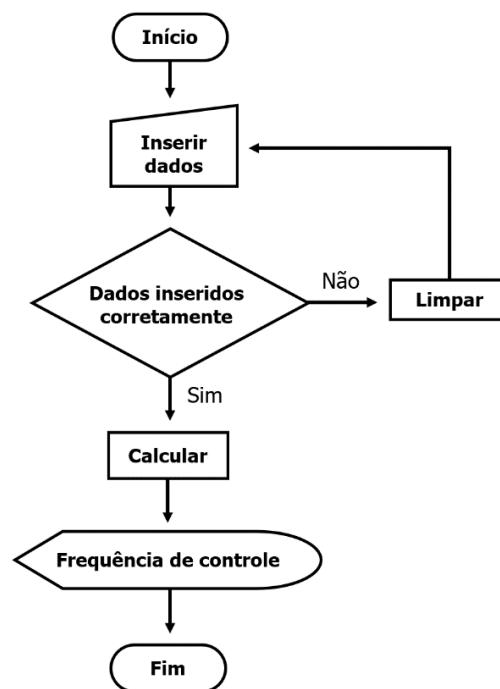
Fonte: Autoria própria

Figura 9 - Trecho F da programação em blocos (tela 2)



Fonte: Autoria própria

Figura 10 - Fluxograma de execução do aplicativo para ajuste da frequência



Fonte: Autoria própria

A interface de usuário da tela de ajuste é também constituída por quatro partes principais. Sendo elas três caixas de entradas para receber os dados, duas caixas de seleção, um botão e um espaço destinado à exibição do valor de frequência ajustado. A tela de ajuste é ilustrada na Figura 11.

Figura 11 - Tela de ajuste do aplicativo desenvolvido (tela 2)

The figure displays three sequential screenshots of a mobile application interface titled "Reajuste da Frequência Para Controle do Motor". Each screen features a green header with a laptop icon and the title. The interface includes several input fields and a "Calcular" button.

- Screen 1 (Left):** The "Velocidade de Simulação Desejada:" field is empty. The "Velocidade Encontrada no Scanner:" field is empty. The "Frequência Usada" field is empty. Below these fields is a question: "O valor encontrado no scanner é menor ou maior que o desejado?" with two radio buttons: "Menor" (selected) and "Maior". A blue "Calcular" button is at the bottom.
- Screen 2 (Middle):** The "Velocidade de Simulação Desejada:" field contains the value "25". The "Velocidade Encontrada no Scanner:" field contains the value "24". The "Frequência Usada" field contains the value "384.27". The "Menor" radio button is selected. A blue "Calcular" button is at the bottom.
- Screen 3 (Right):** The "Velocidade de Simulação Desejada:" field contains the value "25". The "Velocidade Encontrada no Scanner:" field contains the value "24". The "Frequência Usada" field contains the value "384.27". The "Menor" radio button is selected. A blue "Calcular" button is at the bottom.

At the bottom of each screen, there is a black bar with three white icons (a square, a circle, and a triangle). Below the screenshots, the text "Nova Frequência:" is visible, followed by the value "401.08" in the third screenshot.

Fonte: Autoria própria

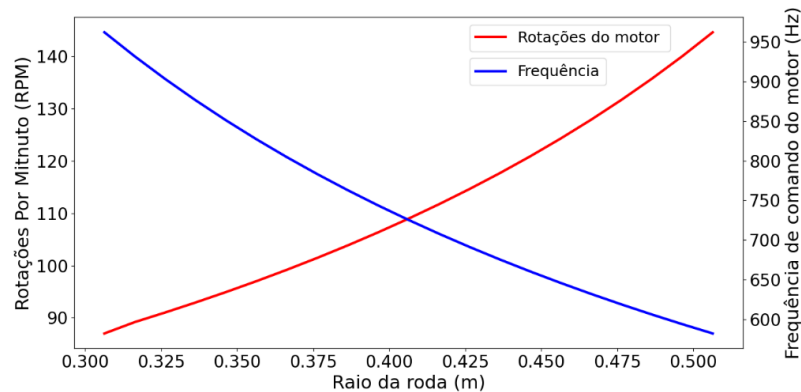
Código python

O software desenvolvido em linguagem de programação Python responsável pelo controle da bancada utiliza as bibliotecas PyQt5, RPi.GPIO e time. A primeira delas permite usar uma interface gráfica criada na plataforma QT Designer, a segunda permite o controle da GPIO (General Purpose Input/Output) do Raspberry Pi e a última permite o controle de tempo na execução de um comando.

A frequência do sinal PWM utilizado para controlar o RPM do motor será o valor de uma variável nomeada como hf , onde a mesma será o valor de uma lista com os valores das frequências de comando com índice igual a uma variável de incremento. Já os valores das velocidades simuladas, mostrados na tela da interface, serão de uma outra lista com esses valores, com o índice igual a mesma variável de incremento.

As relações entre as frequências de comando, raio da roda e as rotações por minuto do motor são ilustradas na Figura 12.

Figura 12 - Relação entre raio da roda, frequência de comando do motor e RPM

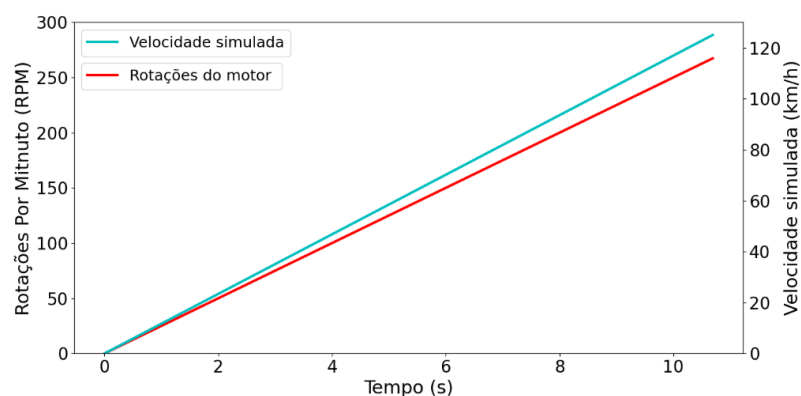


Fonte: Autoria própria

Para administrar os valores das frequências de comando do motor e os valores de velocidades a serem exibidos na tela são usadas duas funções conectadas a dois botões. Esse controle é feito com a alteração do valor da variável de incremento, com passo igual a um, de acordo com o que é solicitado pelo usuário através dos botões de aumentar ou diminuir a velocidade a ser simulada.

A aceleração ou desaceleração do motor é feita através de um laço de repetição. Dessa forma, quando se deseja aumentar ou diminuir o valor de velocidade a ser simulada, é incrementado ou decrementado 1 Hz e posteriormente tem-se um atraso de 6 ms repetindo esse processo até atingir o valor da frequência hf correspondente a um valor de RPM. Onde por sua vez, é correspondente a um valor de velocidade simulada. A Figura 13 ilustra o comportamento descrito.

Figura 13 - Aceleração do motor



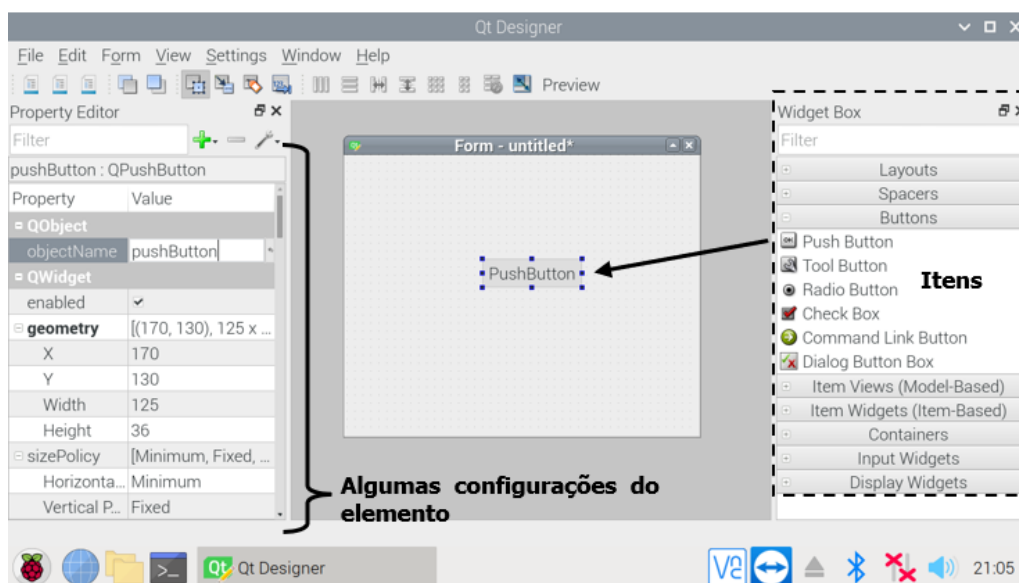
Fonte: Autoria própria

Existe uma função que permite interromper ou parar a simulação de velocidade. Onde ao clicar no botão correspondente à mesma o valor da frequência de comando (variável hf) irá para zero e a variável de incremento voltará a ser igual a sua condição inicial. A situação descrita, ocorre de forma gradual seguindo os mesmos parâmetros da aceleração e desaceleração do motor.

Interface de usuário

A plataforma QT Designer foi utilizada na construção da interface de usuário do software. Essa plataforma permite a criação de interfaces através de objetos como botões, legendas e frames, onde são arrastados e soltos no local desejado. Nesses objetos podem ser alteradas cores, formas e tamanhos, tornando assim, a confecção de interfaces através dessa plataforma bem fluida, como ilustrado na Figura 14.

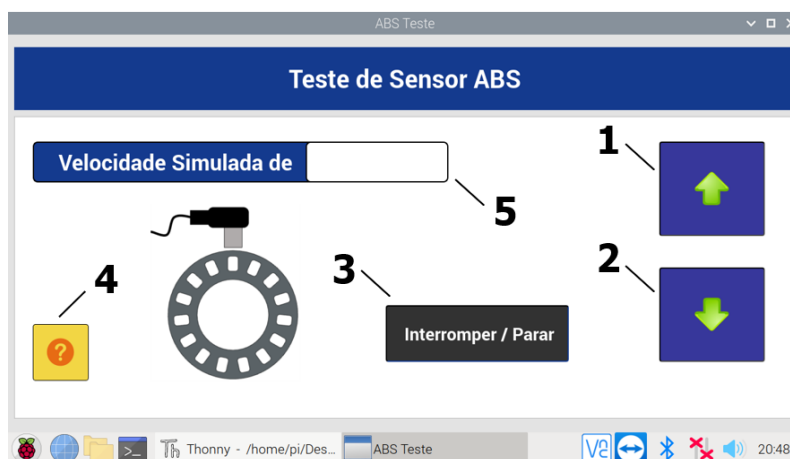
Figura 14 - Tela principal QT Designer



Fonte: Autoria própria

A Figura 15 ilustra a tela principal da interface onde apresenta uma imagem ilustrativa, alguns textos, um campo de exibição 5 para os valores das velocidades simuladas e quatro botões. Os botões 1 e 2 são para aumentar ou diminuir a velocidade simulada, o 3 para interromper ou parar a simulação e o 4 para exibir um campo de ajuda.

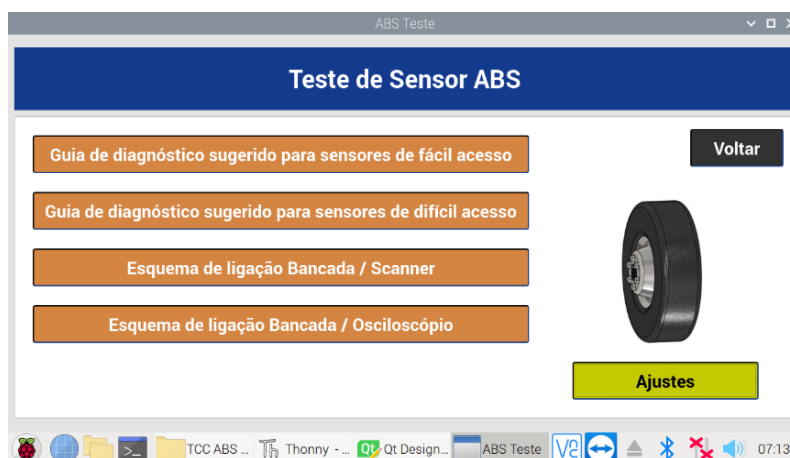
Figura 15 - Tela principal da interface da bancada



Fonte: Autoria própria

O campo de ajuda que pode ser aberto ao clicar no botão 4 é ilustrado na Figura 16. Nesta tela o usuário pode acessar possíveis guias de diagnósticos e esquemas de ligação da bancada.

Figura 16 - Tela contendo o campo de ajuda da interface



Fonte: Autoria própria

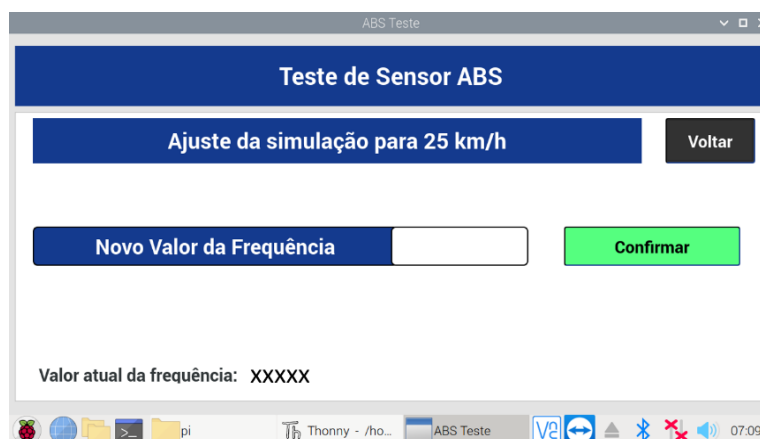
Ao clicar no botão de ajustes contidos na tela de ajuda, o usuário pode ajustar a precisão das simulações da bancada. Primeiramente ele será direcionado a uma tela contendo as simulações, depois deve escolher qual deseja alterar de acordo com a Figura 17. Posteriormente deve inserir o novo valor de frequência calculado com o aplicativo como ilustrado na Figura 18.

Figura 17 - Exemplo das velocidades que a bancada pode simular



Fonte: Autoria própria

Figura 18 - Exemplo de uma tela de ajuste da velocidade simulada



Fonte: Autoria própria

RESULTADOS E DISCUSSÕES

Com o aplicativo desenvolvido foi possível calcular frequências de comando para motores de passo correspondentes a valores de velocidades simuladas. O aplicativo permite o cálculo de maneira simples e rápida, tendo em vista que basta inserir os valores que estão contidos no pneu do veículo e podem ser facilmente coletados e o restante dos valores que são as velocidades de simulação desejadas que são a critério do usuário.

Dessa forma, o aplicativo permite efetuar o cálculo das frequências de comando para qualquer veículo equipado com sistema de freios ABS. Além de efetuar o cálculo da frequência, o aplicativo permite fazer um reajuste da mesma. O que permite aumentar a precisão da simulação. Esse ajuste pode ser feito quantas vezes o usuário julgar necessário ou até mesmo não fazê-lo.

O software desenvolvido permite a simulação das velocidades com base nas frequências calculadas no aplicativo. Sua interface gráfica permite ao usuário controlá-la de forma simples e intuitiva, além de conter outras funcionalidades que podem ajudar o usuário como guias de diagnóstico e esquemas de ligação.

CONSIDERAÇÕES FINAIS

Os procedimentos descritos neste trabalho permitem a confecção de um software para uma bancada de teste de sensores de velocidades do sistema de freios ABS gerenciada por um Raspberry Pi. Onde o aplicativo desenvolvido possibilita encontrar os valores de frequências necessários para as simulações de forma rápida e prática com base nos parâmetros do pneu do veículo. Com isso, o software pode ser aplicado a diversos modelos, categorias e marcas de veículos.

Dessa forma, com o software sendo capaz de controlar a bancada e proporcionar uma interação usuário/máquina simples e intuitiva, torna evidente as contribuições que seriam geradas para o diagnóstico no sistema de freio ABS com a implementação deste software em uma bancada de teste para sensores de velocidades da roda.

REFERÊNCIAS

AMORAN, A. E. et al. Home automated system using bluetooth and an android application. Scientific African, v. 11, p. e00711, 2021. ISSN 2468-2276.

CHING, E. J. et al. Quail: A lightweight open-source discontinuous galerkin code in python for teaching and prototyping. SoftwareX, v. 17, p. 100982, 2022. ISSN 2352-7110.

DOMÍNGUEZ, J. et al. Teaching chemical engineering using jupyter notebook: Problem generators and lecturing tools. Education for Chemical Engineers, v. 37, p. 1–10, 2021. ISSN 1749-7728.

FERNÁNDEZ, J. P. et al. Influence of tire dynamics on a braking process with abs. Transportation Research Procedia, v. 58, p. 189–192, 2021. ISSN 2352-1465. XIV Conference on Transport Engineering, CIT2021.

MCKELVEY, T.; GIBANICA, M. Fsid - a frequency weighted mimo frequency domain identification and rational matrix approximation method for python, julia and matlab.

MUDALIAR, M. D.; SIVAKUMAR, N. Iot based real time energy monitoring system using raspberry pi. Internet of Things, v. 12, p. 100292, 2020. ISSN 2542-6605.

SANGEETHALAKSHMI, K. et al. Hand gesture vocalizer for deaf and dumb people. Materials Today: Proceedings, 2021. ISSN 2214-7853.

URREA, C.; KERN, J. Design and implementation of a wireless control system applied to a 3-dof redundant robot using raspberry pi interface and user datagram protocol. Computers Electrical Engineering, v. 95, p. 107424, 2021. ISSN 0045-7906.

Recebido em: 15/09/2022

Aprovado em: 21/10/ 2022

Publicado em: 26/10/2022