

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**LUIZ OTAVIO HEOFACKER ALVES DE CARVALHO**

**APLICAÇÃO DE SENSORES INERCIAIS PARA DETECÇÃO E CLASSIFICAÇÃO  
DE MANOBRAS DE SKATEBOARD**

**CURITIBA**

**2022**

**LUIZ OTAVIO HEOFACKER ALVES DE CARVALHO**

**APLICAÇÃO DE SENSORES INERCIAIS PARA DETECÇÃO E CLASSIFICAÇÃO  
DE MANOBRAS DE SKATEBOARD**

**Application of inertial sensors for detection and classification of skateboard  
tricks**

Trabalho de conclusão de curso de graduação  
apresentado como requisito para obtenção do título  
de Bacharel em Engenharia Eletrônica da  
Universidade Tecnológica Federal do Paraná  
(UTFPR).

Orientador: Luiz Fernando Copetti

**CURITIBA**

**2022**



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**LUIZ OTAVIO HEOFACKER ALVES DE CARVALHO**

**APLICAÇÃO DE SENSORES INERCIAIS PARA DETECÇÃO E CLASSIFICAÇÃO  
DE MANOBRAS DE SKATEBOARD**

Trabalho de Conclusão de Curso de Graduação  
apresentado como requisito para obtenção do título  
de Bacharel em Engenharia Eletrônica da  
Universidade Tecnológica Federal do Paraná  
(UTFPR).

Data de aprovação: 06 de Dezembro de 2022

---

Luiz Fernando Copetti  
Mestre em Engenharia Elétrica e Informática Industrial  
Universidade Tecnológica Federal do Paraná

---

Gustavo Benvenuto Borba  
Doutor em Engenharia Elétrica e Informática Industrial  
Universidade Tecnológica Federal do Paraná

---

Juliano Mourão Vieira  
Mestre em Ciências da Computação  
Universidade Tecnológica Federal do Paraná

**CURITIBA**

**2022**

Dedico este trabalho aos excelentes professores com quem tive a oportunidade de aprender, e também à sociedade que viabilizou minha oportunidade de aprender em uma universidade pública gratuita de qualidade como a UTFPR.

## **AGRADECIMENTOS**

Agradeço, primeira e principalmente, à todos os meus amigos e familiares que estiveram presentes ao longo da minha formação acadêmica e que, cientes ou não, me deram a motivação para continuar minha jornada acadêmica e profissional.

Agradeço ao meu falecido pai, a quem não terei mais o prazer de compartilhar minhas conquistas, por sempre ter feito questão de me ensinar a ser feliz acima de tudo.

Agradeço à minha mãe que sempre trabalhou muito mais do que deveria para garantir que eu tivesse condições de cursar um ensino superior.

Agradeço ao falecido professor Hugo Vieira Neto por ter me inspirado a buscar excelência não apenas nas minhas linhas de código, mas na maneira como trato as pessoas ao meu redor.

## RESUMO

A prática do esporte de *skateboard* consiste, em parte, na execução de manobras com o skate, dentre as quais estão manobras que são classificadas por saltos, ajustes e rotações do *skate* no ar, que exigem experiência e acuidade visual para serem corretamente classificadas. Este trabalho descreve o desenvolvimento e discute os resultados experimentais de um projeto de aplicação de sensores inerciais (acelerômetros e giroscópios) em um *skateboard*. A captação e classificação destas manobras baseado nas leituras dos sensores é feita por um programa de computador capaz de receber e analisar os dados dos sensores. O projeto se mostrou capaz de classificar corretamente 5 manobras básicas realizadas no *skate* com o auxílio das mãos, como prova de conceito de funcionamento do projeto durante a prática real do esporte. Como classificadores das manobras, foram usados o formato das curvas inerciais, a quantidade de rotações do *skate* contadas em cada eixo em relação ao chão e a quantidade de rotações do *skate* em seu eixo transversal.

Palavras-chave: acelerômetro; giroscópio; *skateboard*; esporte; quatérnions; sistemas embarcados.

## ABSTRACT

The practice of the skateboard sport consists, in part, in the execution of skateboard tricks, which includes aerial tricks that are classified by jumps, adjustments and rotations of the skateboard in the air, which demand experience and visual acuity in order to be correctly classified. This document describes the development and discusses the experimental results of a project for application of inertial sensors (accelerometers and gyroscopes) on a skateboard. The capture and classification of these tricks based on the sensors' readings are done by a computer program capable of receiving and analyzing the sensors' data. The project was capable of correctly classifying 5 basic tricks performed on the skateboard by hand, as a proof of concept of the project's functionality for classifying a real trick. As classifiers for the tricks, the project uses the overall shape of the inertial curves, the skateboard's number of rotations in each axis in relation to the ground and the skateboard's number of rotations in its transversal axis.

Keywords: accelerometer, gyroscope, *skateboard*, sport, quaternions, embedded systems.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Representação dos ângulos de Euler em um caso clássico de uso em aeronaves.....	11
Figura 2 - Diagrama de blocos simplificado do projeto.....	15
Figura 3 - Captura de Tela do Software: Tela de conexão via <i>Bluetooth</i> .....	20
Figura 4 - Captura de Tela do Software: Tela de registro e classificação de manobras.....	21
Figura 5 - Ilustração das rotações no eixo transversal do <i>skateboard</i> .....	22
Figura 6 - Captura de tela das manobras registradas.....	24
Figura 7 - Sigmoide de pontuação de similaridade de curvas, onde a pontuação (eixo y) é função da pontuação de similaridade.....	26



## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>9</b>
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>10</b>
<b>2.1 Equações Cinemáticas de Euler.....</b>	<b>10</b>
<b>2.2 Quatérnions.....</b>	<b>12</b>
<b>3 DESENVOLVIMENTO.....</b>	<b>15</b>
<b>3.1 Sensor Inercial (IMU).....</b>	<b>16</b>
<b>3.2 Microcontrolador.....</b>	<b>16</b>
<b>3.3 Software.....</b>	<b>18</b>
3.3.1 Comunicação <i>Bluetooth</i> .....	19
3.3.2 Registro de Manobras.....	20
3.3.3 Integração da Orientação do <i>Skateboard</i> .....	21
3.3.4 Registro de manobras.....	22
3.3.5 Classificação de Manobras.....	25
3.3.6 Interface Gráfica.....	26
<b>3.4 Testes e Validação.....</b>	<b>28</b>
<b>4 CONCLUSÃO.....</b>	<b>30</b>
<b>REFERÊNCIAS.....</b>	<b>31</b>

## 1 INTRODUÇÃO

A prática do esporte de *skateboard*, tanto em caráter competitivo quanto recreativo, envolve a execução de manobras aéreas com o *skate*, geralmente partindo de uma posição com o atleta em cima do *skate*. Com um movimento, ou uma série de movimentos, o *skate* é elevado do chão e rotacionado ou não no ar, terminando com o atleta aterrissando em cima do *skate* novamente.

A classificação de manobras de *skateboard* normalmente é feita visualmente por pessoas que já têm experiência com o esporte e têm conhecimento para reconhecer as manobras executadas. Mesmo assim, a classificação não é trivial no sentido de que, com pequenas variações na sequência de movimentos do *skate* ou do atleta, a dificuldade da manobra pode variar drasticamente, como alterar a rotação do *skate* no meio de outra rotação, ou adicionar rotações a mais em manobras já existentes, que são difíceis de acompanhar e contar.

Neste contexto, o propósito deste projeto é fazer com que a tarefa de classificação de manobras seja mais precisa, para que, por exemplo, auxilie juízes em competições, e também torne o esporte mais acessível, ajudando novos atletas a reconhecer e aprender manobras novas e aperfeiçoar manobras aprendidas.

A proposta do projeto é o desenvolvimento de uma prova de conceito de um sistema embarcado que pode ser instalado no *skate*, de forma a não atrapalhar a prática do esporte, enviando dados inerciais do *skate* a um programa de computador ou aplicativo de telefone. O programa teria uma interface capaz de capturar dados de manobras e classificá-los com relação à manobras já cadastradas em um banco de dados.

Potencialmente, um programa deste tipo poderia ser treinado para aprender diversas manobras, assim como manobras mal-executadas, para ajudar atletas inexperientes a saber como identificar problemas na execução das suas manobras.

Por fim, o conceito será validado por manobras executadas no *skate* com o auxílio das mãos, de forma a tentar executar a mesma sequência de movimentos e rotações do *skate* que aconteceriam nas manobras reais.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos utilizados para a solucionar os problemas relacionados à detecção de manobras a partir dos dados de sensores inerciais. Além de conceitos básicos do curso de bacharelado em Engenharia Eletrônica, como programação de sistemas embarcados, processamento de sinais e estruturas de dados, o algoritmo de classificação foi construído a partir do entendimento das equações cinemáticas de Euler e de Quatérnions.

### 2.1 Equações Cinemáticas de Euler

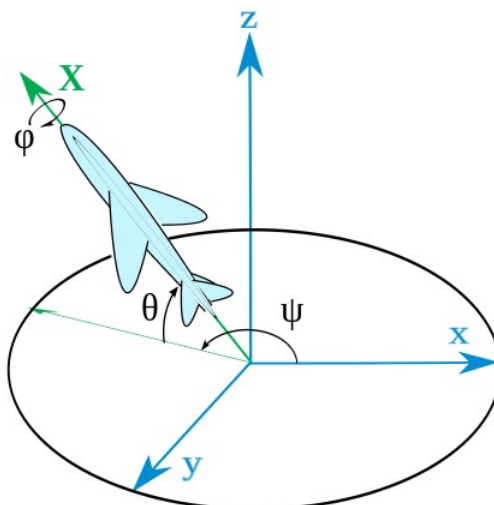
As Equações Cinemáticas de Euler são um conjunto de equações diferenciais de primeira ordem que descrevem a cinemática de um corpo rígido em respeito a uma referência fixa no espaço. Estas equações fazem uso da parametrização de rotações finitas de um corpo rígido no espaço euclidiano, descritas por Euler em 1775, que descrevem a rotação de um corpo como uma série de 3 rotações em eixos diferentes (EULER, 1775, p.189).

Atualmente, o uso de Ângulos de Euler ainda é uma das maneiras mais comuns para representar a orientação de um objeto no espaço (DIEBEL, 2006, p.15).

Conforme observados na Figura 1 em relação à referência fixa  $xyz$ , os ângulos de Euler são:

- Precessão, ou *Yaw* em inglês ( $\psi$ ): Rotações ao redor do eixo Z do corpo rígido;
- Nutação, ou *Pitch* em inglês ( $\theta$ ): Rotações ao redor do eixo Y do corpo rígido;
- *Spin*, ou *Roll* em inglês ( $\phi$ ): Rotações ao redor do eixo X do corpo rígido.

**Figura 1 - Representação dos ângulos de Euler em um caso clássico de uso em aeronaves**



**Fonte: Blog DreamAndDead, 2019**

Tendo conhecimento das velocidades angulares na referência do corpo rígido e sua atual rotação no espaço em relação a uma referência fixa, é possível calcular também a velocidade angular do corpo rígido em relação à referência fixa através das Equações Cinemáticas de Euler.

Apesar de Euler ter parametrizado a orientação de um objeto como uma sequência de três rotações sucessivas, tanto a ordem dessas rotações quanto quais ângulos de Euler são usados variam conforme a convenção usada. Dentre várias combinações de rotações possíveis, uma das mais comuns usadas na literatura e aeronáutica é a de Precessão-Nutação-Spin, ou *Yaw-Pitch-Roll* em inglês, ou seja, primeiro é executada a precessão, depois a nutação e enfim o *spin*.

Considere, nas equações abaixo, a seguinte nomenclatura para descrever as equações cinemáticas de Euler de Precessão-Nutação-Spin:

- $\bar{\omega}_{b/v}$  representa o vetor de velocidades angulares do corpo rígido em relação a si mesmo.
- $H(\psi, \theta, \phi)$  é a matriz que representa as equações cinemáticas, em função apenas dos ângulos de nutação e *spin*.
- $\dot{\Phi}$  é o vetor de velocidades angulares do corpo rígido em relação à referência fixa.

$$\bar{\omega}_{b/v} = H^{-1}(\theta, \phi) \cdot \dot{\Phi}$$

$$\begin{bmatrix} \omega_{\phi b} \\ \omega_{\theta b} \\ \omega_{\psi b} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\text{sen}(\theta) \\ 0 & \cos(\phi) & \cos(\phi) \cdot \cos(\theta) \\ 0 & -\text{sen}(\phi) & \cos(\phi) \cdot \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

Da mesma forma, através da matriz inversa é possível obter as velocidades angulares em relação à referência fixa a partir das velocidades angulares do corpo rígido e sua posição atual:

$$\dot{\Phi} = H(\theta, \phi) * \bar{\omega}_{b/v}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \text{sen}(\phi) \cdot \tan(\theta) & \cos(\phi) \cdot \tan(\theta) \\ 0 & \cos(\phi) & -\text{sen}(\phi) \\ 0 & \text{sen}(\phi) \cdot \text{sec}(\theta) & \cos(\phi) \cdot \text{sec}(\theta) \end{bmatrix} \cdot \begin{bmatrix} \omega_{\phi b} \\ \omega_{\theta b} \\ \omega_{\psi b} \end{bmatrix}$$

Também é possível usar as equações cinemáticas na sua forma diferencial, de forma a usá-las em cálculos integrativos para obter, em tempo real, a orientação de um corpo rígido baseado em suas velocidades angulares.

Observe que, devido à presença das funções trigonométricas não contínuas da tangente e secante de  $\theta$ , as equações apresentam singularidades que inviabilizam o cálculo da cinemática do objeto em orientações cujo ângulo  $\theta$  se aproxima de múltiplos de  $90^\circ$ , onde as funções de tangente e secante são indefinidas.

Devido a essa singularidade, o uso das equações diferenciais para calcular a orientação em tempo real é inviável quando for esperado que o objeto assumira orientações com precessão próxima a  $90^\circ$  ou  $-90^\circ$ .

## 2.2 Quatérnions

Os quatérnions são estruturas quadridimensionais que podem ser usadas para representar a orientação de um objeto no espaço euclidiano tridimensional. Descritos por Hamilton no século XIX, os quatérnions foram propostos como alternativa às equações cinemáticas de Euler (HAMILTON, 2000).

Ainda no século XX, os quatérnions se tornaram uma ferramenta muito utilizada em problemas de cinemática inversa para aplicações em robótica e engenharia aeroespacial (AMORUSO, 1996).

Os quatérnions são uma generalização dos números complexos para quatro dimensões, compostos por uma parte escalar  $q_0$  e uma parte vetorial  $\vec{q} = \{q_1, q_2, q_3\}$  (JAMBERSI, 2016). Os quatérnions podem ser representados como:

$$\begin{aligned}\hat{q} &= q_0 + q_1 \cdot i + q_2 \cdot j + q_3 \cdot k \\ \hat{q} &= \{q_0 \quad q_1 \quad q_2 \quad q_3\}^T \\ \hat{q} &= \begin{Bmatrix} q_0 \\ \vec{q} \end{Bmatrix}\end{aligned}$$

Um quatérnion  $\hat{q}$  também possui conjugado  $\hat{q}^*$  e módulo  $|\hat{q}|$  tal que:

$$\begin{aligned}\hat{q}^* &= q_0 - \vec{q} = q_0 - q_1 \cdot i - q_2 \cdot j - q_3 \cdot k \\ |\hat{q}| &= \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}\end{aligned}$$

Para uso dos quatérnions nas equações cinemáticas, é possível abstrair que a parte escalar  $q_0$  representa uma rotação  $\theta$  ao redor de um vetor  $\vec{q}$ , que por sua vez é composto por um vetor unitário  $\bar{e}$ , de forma que:

$$\hat{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ \bar{e} \cdot \text{sen}(\theta/2) \end{bmatrix}$$

As equações cinemáticas dos quatérnions podem então ser representadas da seguinte forma, onde  $\dot{q}$  representa a velocidade angular representada em quatérnion em relação à referência fixa,  $\Omega$  é uma matriz transformadora composta pelas velocidades angulares do corpo rígido e  $q$  é a orientação atual do corpo rígido representada em quatérnion (LUM, 2020):

$$\dot{q} = \frac{1}{2} \cdot \Omega \cdot q$$

Onde  $\Omega$  é composta da seguinte forma, onde  $\omega_\psi$ ,  $\omega_\theta$  e  $\omega_\phi$  representam as velocidades angulares de precessão, nutação e *spin*, respectivamente:

$$\Omega = \begin{bmatrix} 0 & -\omega_\phi & -\omega_\theta & -\omega_\psi \\ \omega_\phi & 0 & \omega_\psi & -\omega_\theta \\ \omega_\theta & -\omega_\psi & 0 & \omega_\phi \\ \omega_\psi & \omega_\theta & -\omega_\phi & 0 \end{bmatrix}$$

Por fim, é possível determinar também a relação direta entre um quatérnion e um conjunto de ângulos de Euler, como demonstrado por Jose-Luiz (JOSE-LUIZ, 2010):

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan \frac{2 \cdot (q_0 \cdot q_1 + q_2 \cdot q_3)}{1 - 2 \cdot (q_1^2 + q_2^2)} \\ \arcsen(2(q_0 \cdot q_2 - q_3 \cdot q_1)) \\ \arctan \frac{2 \cdot (q_0 \cdot q_3 + q_1 \cdot q_2)}{1 - 2 \cdot (q_2^2 + q_3^2)} \end{bmatrix}$$

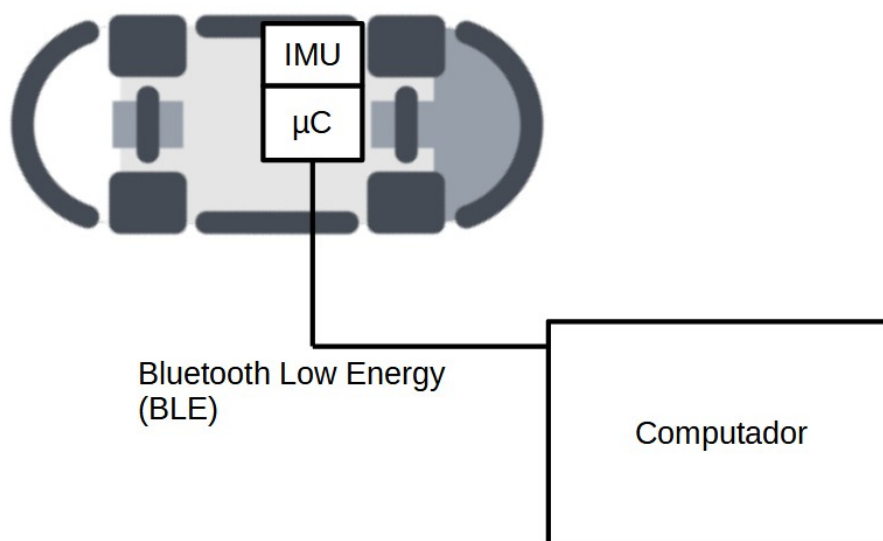
### 3 DESENVOLVIMENTO

De forma geral, o projeto está estruturado da seguinte forma:

- Dois sensores inerciais, ou IMU (*Inertial Measurement Unit*), montados no *skateboard*;
- Um microcontrolador (abreviado como  $\mu\text{C}$ ) modelo Arduino Uno embarcado no *skateboard*, com uma fonte de energia;
- Um módulo de comunicação Wi-Fi e *Bluetooth* HC-05 montado no *skateboard*, de forma a interfacear os dados colhidos pelo microcontrolador com o *software*;
- Um *software* desenvolvido para computador, capaz de usar *drivers* do adaptador *Bluetooth* do sistema operacional para receber os dados do sensor inercial e processá-los.

A Figura 2 abaixo ilustra a arquitetura do projeto de forma simplificada:

**Figura 2 - Diagrama de blocos simplificado do projeto**



Fonte: Autoria Própria (2022).

Este capítulo está estruturado de forma a apresentar cada parte do projeto individualmente, antes de analisar o resultado final.



### 3.1 Sensor Inercial (IMU)

Os sensores inerciais utilizados são do modelo MPU6050, que é um módulo comumente utilizado para projetos embarcados devido à sua facilidade de implementação e precisão suficiente para movimentos na escala humana. O módulo possui um acelerômetro de 3 eixos e um giroscópio de 3 eixos, totalizando 6 graus de liberdade.

A interface de comunicação do IMU é I2C, e os protocolos de comunicação que executam as operações são documentados e disponibilizados através da biblioteca “MPU6050\_6Axis\_MotionApps20.h”.

O sensor foi conectado ao Arduino apenas pela fonte de alimentação e dois pinos da interface I2C. Também foi necessário ligar o pino de ID do segundo IMU à alimentação do Arduino para configurar um ID diferente do primeiro IMU.

Para serem inicializados corretamente, os sensores foram configurados para um fundo de escala de 16 g nos acelerômetros, e um fundo de escala de 2000 °/s. Não foi feito um estudo das forças e rotações envolvidas na prática do *skate*, mas foi estimado que 16 g seria suficiente para detectar todas as forças humanas envolvidas (impulso das pernas, pulo, movimento de rotação com os pés, etc.), e 2000 °/s seria suficiente dado que uma das manobras de *skate* com rotação mais rápida consiste em 3 rotações (1080°) em um pulo só.

Durante o planejamento do projeto, foi previsto o uso de dois IMUs para que houvesse informação suficiente para discernir entre quaisquer manobras executadas no *skate*, mas nos últimos estágios do desenvolvimento do projeto foi notado que isto era desnecessário, fazendo com que a única vantagem de haver dois IMUs fosse a redundância de dados.

### 3.2 Microcontrolador

Para a escolha do microcontrolador, foram avaliadas as necessidades do projeto de velocidade de processamento, interfaces de comunicação, entradas e saídas digitais. O microcontrolador escolhido foi o modelo popular Arduino Uno, dado que tem velocidade de processamento suficiente para amostrar os sensores inerciais e enviar dados via *Bluetooth* numa taxa na ordem de milissegundos.

Porém, a principal razão de escolha do Arduino Uno foi a existência de bibliotecas usadas para interfacear o *firmware* (programa embarcado) com ambos os sensores inerciais e o módulo *Bluetooth*.

O *firmware* foi então programado para:

- Iniciar os IMUs e a interface I2C para uso dos sensores inerciais;
- Iniciar uma interface serial virtual para comunicação com o módulo de *Bluetooth* HC-05;
- Configurar o módulo *Bluetooth* com nome e senha para ser acessado pelo computador;
- Ativar um *timer* de interrupção recorrente de 20ms para fornecer uma base de tempo para a amostragem dos IMUs e envio de dados;
- Um laço principal no qual, quando ocorre o sinal da interrupção que 20ms se passaram, amostra e envia os dados dos IMUs para o módulo *Bluetooth*;
- Um protocolo de envio de dados pelo módulo *Bluetooth* para ser reconhecido pelo *software*.

Assim, foi possível testar o sistema embarcado como um todo, através do uso de um aplicativo de escuta serial no celular.

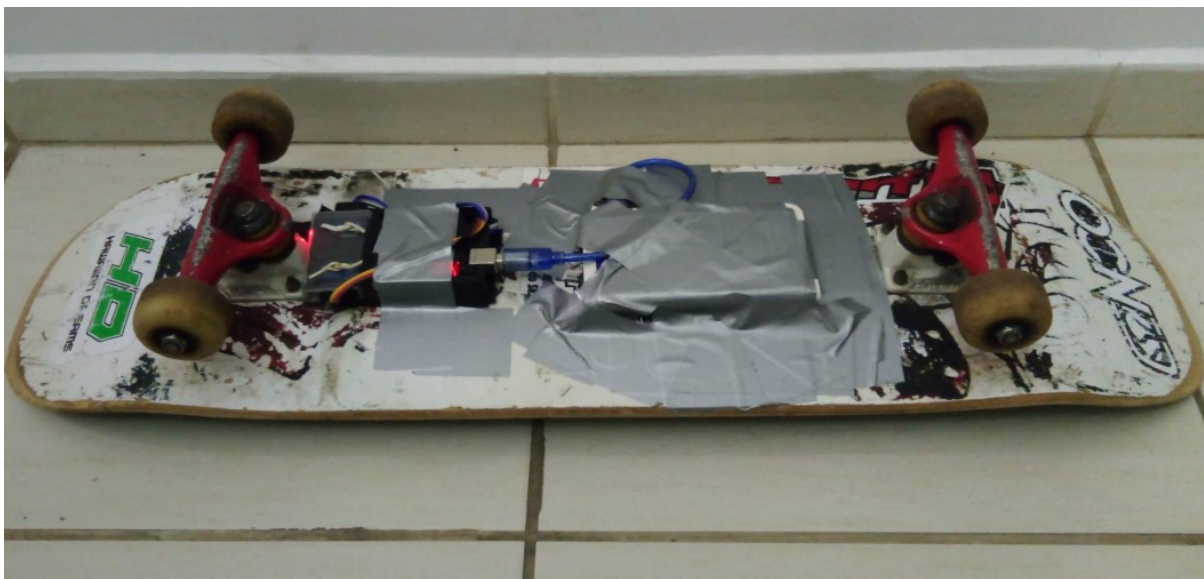
A periodicidade de envio dos dados foi validada posteriormente com o auxílio do *software* desenvolvido para o projeto.

A principal desvantagem da escolha do Arduino Uno, assim como dos módulos de sensores inerciais e do módulo *Bluetooth* é que, somados os volumes de todos os dispositivos, foi impossível montar o sistema embarcado em um local no *skate* que não atrapalhe na prática normal do esporte. Para este projeto, todos os dispositivos foram montados na parte inferior do *skate*.

Idealmente, seria possível fabricar uma placa com apenas os circuitos integrados e conexões estritamente necessárias ao projeto, assim como o uso de uma bateria pequena, para montá-lo em um local melhor.

Na Fotografia 1 é possível observar o projeto montado na parte inferior do *skate*, usando parafusos para fixar o microcontrolador e um invólucro plástico com os módulos de sensores e *Bluetooth* e usando fita adesiva para fixar a bateria e reforçar a montagem do projeto. Na Fotografia 2 é possível ver o local de fixação dos parafusos.

Fotografia 1 - Foto do projeto montado na parte inferior do skate.



Fonte: A autoria própria (2022).

Fotografia 2 - Foto do projeto montado na parte superior do skate.



Fonte: A autoria própria (2022).

### 3.3 Software

O programa de computador (*software*) criado para realizar o processamento de dados e interface com o usuário foi programado na linguagem C#, com o propósito de guiar o usuário através de todas as etapas de uso do projeto.

A interface gráfica foi criada usando o modelo UWP (*Universal Windows Platform*), um modelo novo criado para a plataforma de desenvolvimento da Microsoft, para os sistemas Windows 10 e 11.

O *software* desenvolvido executa 6 tarefas essenciais:

- Estabelecer a comunicação com o microcontrolador embarcado via *Bluetooth*;
- Registrar os dados recebidos dos sensores inerciais ao longo de uma janela de tempo quando uma manobra é detectada;
- Integrar a orientação do *skate* a todos os momentos para inferir, a partir dos dados dos giroscópios, quantas rotações o *skate* deu em cada eixo em respeito à referência fixa do chão;
- Cadastrar manobras novas em um banco de dados local e armazenar os dados inerciais como um padrão, capaz de aproximar uma média de várias amostras diferentes;
- Comparar manobras capturadas pelos sensores com as manobras armazenadas e classificá-las;
- Fornecer uma interface gráfica ao usuário, para disponibilizar os controles de captura de manobras, alteração do banco de dados e exibir os dados recebidos dos sensores inerciais de forma inteligível para permitir análise e validação.

As metodologias utilizadas e dificuldades enfrentadas na execução de cada uma dessas tarefas será discutida em capítulos individuais para cada uma.

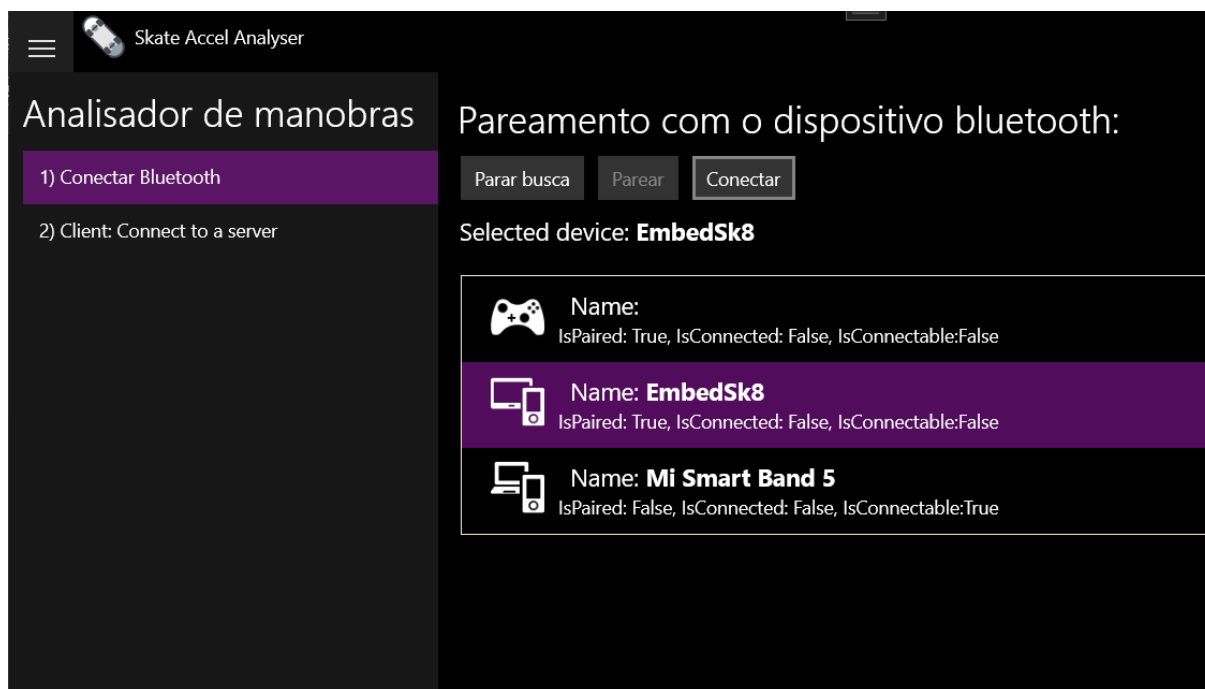
### 3.3.1 Comunicação *Bluetooth*

Para implementar a interface *Bluetooth*, primeiramente foi identificado no manual do módulo *Bluetooth* HC-05 que o protocolo *Bluetooth Low Energy* (BLE) era utilizado. O Protocolo BLE realiza uma série de transações entre o cliente (computador) e o servidor (microcontrolador) para informar ao cliente qual serviço está disponível e qual a forma de conexão.

Para realizar esta função, o *software* foi baseado em um exemplo pronto fornecido pela Microsoft para o uso de desenvolvedores, com as funções de busca e conexão com um servidor BLE. A interface gráfica é mostrada na Figura 3 abaixo.

A recepção dos dados foi implementada em uma função chamada por esse *framework*, lendo o protocolo enviado pelo microcontrolador e populando os dados do sensor em classes na memória RAM.

**Figura 3 - Captura de Tela do Software: Tela de conexão via *Bluetooth***



Fonte: Autoria Própria (2022).

### 3.3.2 Registro de Manobras

Para registrar os dados amostrados pelos sensores inerciais na janela de tempo de execução de uma manobra, foi criada uma condição de gatilho para iniciar uma captura de manobra, que armazena os próximos 2 s de amostras do sensor inercial em memória para ser processado.

A condição de gatilho foi definida como uma variação contínua da aceleração em qualquer eixo (x, y ou z) que dure mais de 60 ms com a mesma derivada e tenha módulo maior que 0,15 g durante todo o tempo. Com esses parâmetros, foi possível detectar quando o usuário exerce impulsos no *skate* para iniciar uma manobra, sem contar ruídos devido à rolagem do *skate* em superfícies não lisas, por exemplo.

Ao ativar o gatilho de captura de uma manobra, as próximas 100 amostras recebidas do microcontrolador são armazenadas em memória RAM para posterior processamento e exibição na tela.

Por fim, antes de comparar e classificar a manobra, seus valores são normalizados, de forma a não discriminar a mesma manobra executada com intensidades diferentes, como fazer a manobra mais alta ou com mais força.

**Figura 4 - Captura de Tela do Software: Tela de registro e classificação de manobras**



**Fonte: Autoria Própria (2022).**

Para exibir os dados dos sensores inerciais na tela na forma de gráficos, foi implementada uma pequena biblioteca para desenhar curvas na tela. Outras soluções prontas para a plataforma UWP foram consideradas, mas apresentavam um excesso de funções desnecessárias ao projeto que diminuam a taxa de exibição dos dados na tela abaixo da taxa de recebimento dos dados dos sensores.

### 3.3.3 Integração da Orientação do *Skateboard*

Para calcular a posição do *skateboard* ao longo de toda a manobra, foram usados Ângulos de Euler para representar sua posição atual, usando a convenção Precessão-Nutação-*Spin* dos ângulos de Tait-Bryan, e quatérnions para integrar as velocidades angulares medidas pelos giroscópios.

Para definir a posição inicial do *skateboard*, foi assumida a premissa de que, no momento de gatilho de uma manobra detectada, o *skateboard* sempre parte de uma posição paralela ao chão, ou seja, Ângulos de Euler =  $\{0^\circ, 0^\circ, 0^\circ\}$ .

Com a informação da orientação do *skateboard* a todos os momentos e suas velocidades angulares medidas pelos giroscópios, é possível usar as equações cinemáticas inversas de Euler para calcular as velocidades angulares do *skateboard* em relação ao chão e, assim, contar de quantas rotações do *skateboard* em cada eixo é composta a manobra.

Além da quantidade de rotações em cada eixo, também é medida a quantidade de rotações no eixo transversal direto da leitura de *spin* dos giroscópios, que é um fator importante de classificação de manobras, conhecida no esporte como *flips*. As rotações no eixo transversal são ilustradas na Figura 5.

Como manobras com quantidades diferentes de rotações podem ter curvas dos sensores inerciais com formatos quase idênticos, a contagem de rotações e *flips* se tornou um classificador importante das manobras.

**Figura 5 - Ilustração das rotações no eixo transversal do *skateboard*.**



**Fonte: Autoria própria (2022).**

### 3.3.4 Registro de manobras

O registro de novas manobras no banco de dados local é uma das funcionalidades mais importantes do *software*, que permite ao usuário o cadastro de qualquer manobra de *skateboard*, de forma generalizada.

Foi previsto que, para ser possível criar um padrão de uma manobra usado no algoritmo de classificação, seria necessário ao usuário executar a manobra

diversas vezes, e então calcular uma média desses dados e cadastrá-la como o padrão.

Além disso, também foi criado um modo de, durante o processo de cadastro de manobras, aplicar pesos a cada execução da manobra, a fim de permitir ao usuário influenciar a criação do padrão com seu próprio conhecimento.

Como as curvas de captura de manobra são normalizadas, o registro de manobras e algoritmo de classificação não têm viés de magnitude das forças aplicadas no *skateboard*, mas sim do formato das curvas dos acelerômetros e giroscópios.

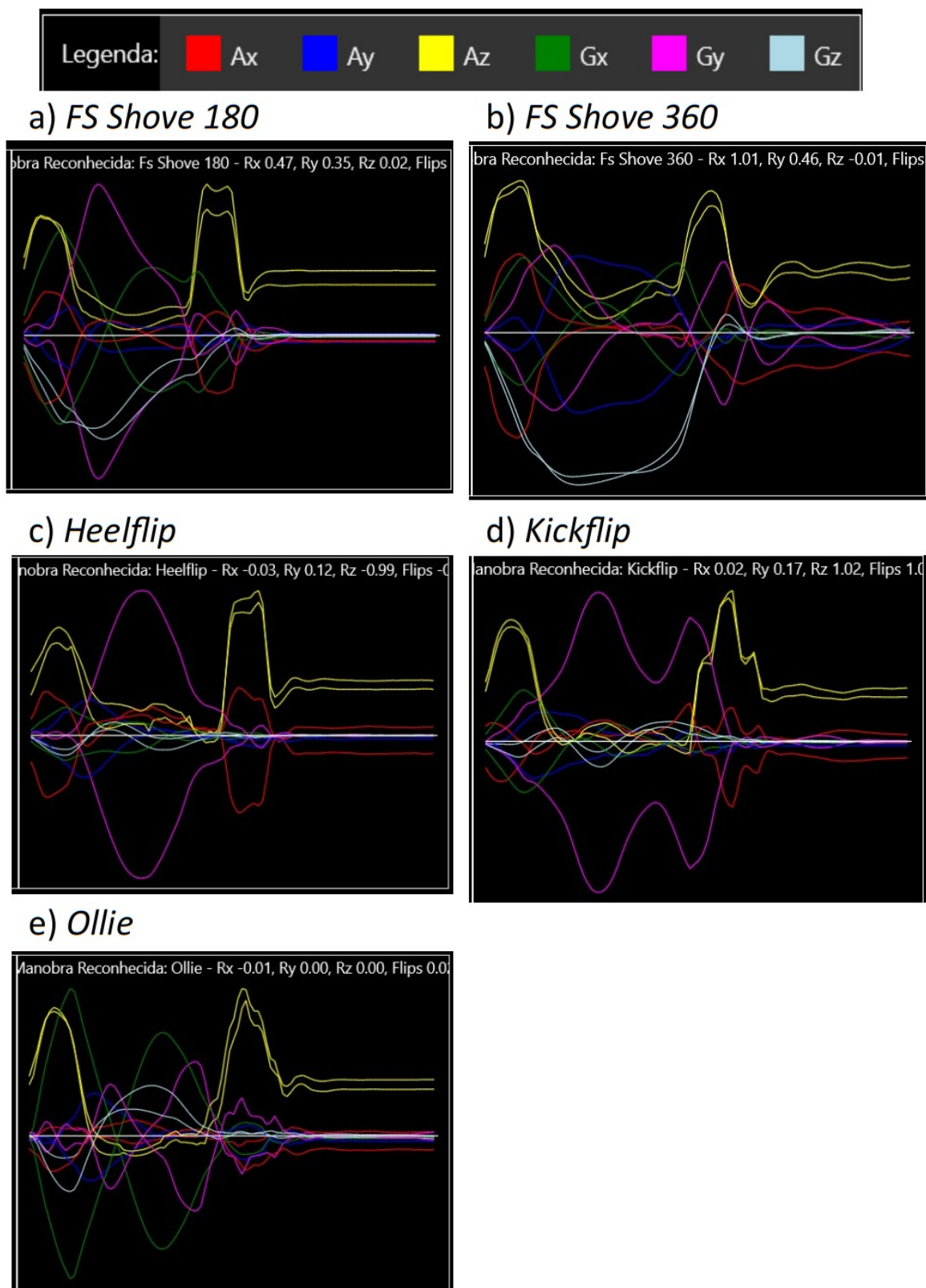
Junto com os formatos das curvas inerciais, também é armazenado o número de rotações contadas em cada eixo e o número de *flips* no eixo transversal.

Na Figura 6 são mostradas todas as 5 manobras registradas para a validação da prova de conceito do projeto. Os dados de ambos os sensores inerciais estão sobrepostos, observando que, devido aos sensores estarem montados em orientações diferentes, os eixos x e y têm leituras invertidas.

É possível observar uma grande similaridade entre as curvas inerciais de cada manobra, o que é ainda mais evidente na curva de aceleração do eixo z (vertical). Isso é dado pelo fato que todas as manobras iniciam com o *skate* parado no chão, que é acelerado para cima, fica suspenso por um momento no ar e finalmente sente aceleração no eixo vertical novamente quando aterrissa no chão virado para cima.



Figura 6 - Captura de tela das manobras registradas.



Fonte: Autoria própria (2022).

### 3.3.5 Classificação de Manobras

Como mencionado anteriormente, o algoritmo de classificação de manobras usa 3 principais parâmetros como classificadores:

- Formato das curvas inerciais normalizadas (aceleração e velocidade angular nos eixos X, Y e Z de ambos os sensores inerciais);
- Número de rotações em cada eixo em relação à referência do chão;
- Número de *flips* em torno do eixo transversal do *skateboard*.

Com estes 3 parâmetros, é passado um primeiro critério de seleção: a manobra capturada só pode ser classificada em manobras com o mesmo número de rotações em cada eixo e *flips*, com um erro máximo de meia rotação.

Após selecionar as manobras registradas com o mesmo número de rotações, são comparadas as curvas inerciais da manobra capturada com a registrada, de forma a atribuir uma pontuação de similaridade entre as duas manobras.

Para calcular a pontuação de similaridade entre duas curvas, é necessário observar duas grandes dificuldades enfrentadas quando há uma comparação de dois sinais de tempo, que são o atraso no tempo e a diferença de magnitude dos sinais.

Foi considerado o uso da convolução entre as duas curvas para determinar o valor máximo da convolução entre duas curvas para superar o problema do atraso no tempo. Outra solução considerada foi a extração de características da curva, para tentar identificar em quais momentos da manobra há uma interação entre o atleta e o *skate*. Porém, dado que as curvas inerciais são normalizadas e que toda janela de captura de manobra começa aproximadamente no momento que o *skate* começa a se movimentar, foi adotada uma solução mais simples de integrar a área entre as duas curvas e, assim, calcular o erro entre elas.

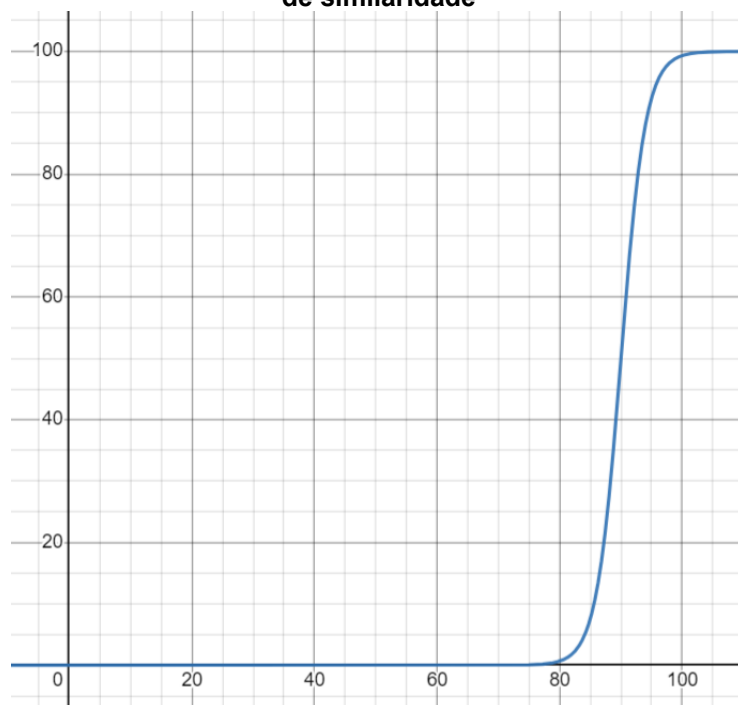
A pontuação de similaridade é atribuída calculando-se o módulo da diferença entre cada amostra das duas curvas, somando assim o erro de todas as amostras. Por fim a soma dos erros é dividida pela soma total de erros possível, ou seja, duas curvas normalizadas opostas ( $y=1$  e  $y=-1$ ), resultando em um erro total de 0 a 100%.

Assim, é calculado o erro total de todas as 12 curvas disponíveis dos sensores inerciais, calculada a média desses valores e, por fim, atribuída uma pontuação de similaridade temporária de 0 (erro máximo) a 100 (sem erro).

Como o conjunto das curvas inerciais que podem ser capturadas em uma manobra são bastante similares entre si, também é aplicada uma função sigmoide para a separação de manobras com pontuação que devem ser classificadas como “iguais” das manobras que devem ser classificadas como “diferentes”.

Após ajustes experimentais, foi aplicada a sigmoide vista na Figura 7 para atribuir uma pontuação final à classificação das manobras, a partir da pontuação de similaridade:

**Figura 7 - Sigmoide de pontuação de similaridade de curvas, onde a pontuação (eixo y) é função da pontuação de similaridade**



Fonte: Autoria Própria (2022).

Nota-se que manobras com pontuação de similaridade abaixo de 80 recebem pontuação final de 0%.

### 3.3.6 Interface Gráfica

A interface gráfica implementada teve como objetivo permitir não só a usabilidade do *software* ao usuário, mas também facilitar a análise dos dados durante o desenvolvimento do projeto e facilitar a depuração de erros.

A interface foi implementada em um modelo UWP e consiste em duas páginas: uma para efetuar a conexão com o microcontrolador via *Bluetooth* e outra para exibir os dados dos sensores, manobras capturadas, classificadas e registradas.

A primeira página tem apenas um conjunto de botões e uma lista de dispositivos encontrados para estabelecer a comunicação *Bluetooth*.

A segunda página é composta por 2 gráficos desenhando as curvas inerciais amostradas ao vivo pelo microcontrolador, uma série de controles de usuário para auxiliar com a captura e registro de manobras e mais 2 gráficos para exibir as manobras capturadas e classificadas.

Visto que houve grande dificuldade em achar uma biblioteca para o desenho dos gráficos que acompanhasse a taxa de atualização de dados das 12 curvas dos sensores inerciais ao vivo, foi desenvolvida uma biblioteca própria para uso no projeto, com uso eficiente das funções de desenho aplicadas ao escopo do projeto, sem funções custosas adicionais.

Os controles disponíveis ao usuário são:

- Gravar manobra: inicia o processo de captura normal;
- Forçar gravação: ignora o gatilho de detecção de manobra e inicia a janela de captura imediatamente;
- Salvar: permite salvar uma manobra recém-capturada no banco de dados para compor o padrão de uma manobra existente, assim como selecionar o peso (de 1 a 5) que a manobra deve ter no cálculo do padrão;
- Criar manobra: cria uma nova manobra com o nome selecionado pelo usuário;
- Exibir manobra: exibe no gráfico de manobra classificada as curvas inerciais de alguma manobra selecionada pelo usuário.

### 3.4 Testes e Validação

Com os equipamentos montados e o *software* funcionando, foi planejado no escopo do projeto o cadastramento e testes de 3 manobras distintas e acessíveis a iniciantes do esporte: *Ollie*, *Kickflip* e *Fs Shove 180°*. Porém, devido a uma lesão na coluna do aluno, o projeto só pode ser testado com manobras executadas através de rotações do *skateboard* com as mãos, tentando executar as manobras com as mesmas rotações e mesma velocidade que seriam feitas normalmente.

Foram então testadas as seguintes manobras manualmente: *Ollie*, *Kickflip*, *Heelflip*, *Fs Shove 180°* e *Fs Shove 360°*. As manobras foram escolhidas de forma a validar diferentes qualidades de manobras do *skate*:

- *Ollie*: Salto, nivelção do *skate* no ar e aterrissagem;
- *Kickflip*: Salto, nivelção do *skate* no ar, *flip* de 180° para frente e aterrissagem;
- *Heelflip*: Salto, nivelção do *skate* no ar, *flip* de 180° para trás e aterrissagem;
- *Fs Shove 180°*: Salto pequeno ou não existente, precessão de 180° e aterrissagem;
- *Fs Shove 360°*: Salto pequeno ou não existente, precessão de 360° e aterrissagem;

As manobras foram cadastradas, através da função de registro de manobras com a média de várias execuções para cada manobra, e então o projeto foi testado executando manobras diferentes e testando suas classificações, avaliando a pontuação recebida e se o *software* foi capaz de classificar a manobra corretamente ou não.

Apesar do sistema de pontuação das manobras não ter sido finamente ajustado para ter um sentido intuitivo do que significa cada pontuação, foi observado que a classificação final da manobra sempre reconheceu a manobra correta como a mais similar. Foram feitas pelo menos 5 tentativas para cada uma das 5 manobras.

Também foi testada a execução de manobras inexistentes, que não poderiam ser classificadas como manobras válidas. Dentre as manobras testadas, o resultado variava entre simplesmente atribuir uma pontuação de 0% (classificou corretamente como manobra desconhecida) ou atribuiu uma pontuação baixa a uma manobra que coincidia de ter o mesmo número de rotações e *flips* (classificou como parecida com uma manobra existente, mas ainda com pontuação baixa).

Não foi possível realizar uma análise estatística de sensibilidade e especificidade do algoritmo de classificação, uma vez que o número de classes e amostras é pequeno, e a execução das manobras é enviesada por serem executadas sempre pela mesma pessoa.

Apesar da prova de conceito ter obtido êxito com as manobras testadas, para demonstrar a viabilidade do projeto na prática do esporte, seria necessário ainda um ajuste fino do sistema de pontuação e do algoritmo de classificação baseado em dados reais das manobras executadas no *skateboard*. Também seria necessária uma análise de performance do algoritmo de classificação usando dados reais de vários praticantes de *skateboard*, assim como registro de mais manobras além das 5 testadas.

## 4 CONCLUSÃO

Para o desenvolvimento do projeto deste trabalho de conclusão de curso, foram usados conceitos de várias áreas de conhecimento de competência da Engenharia Eletrônica: sistemas embarcados, protocolos de comunicação, programação, estruturas de dados, processamento de dados, integração, entre outros.

Como resultado final, apesar de não ter sido viável realizar análises estatísticas para avaliar a eficácia do projeto, foi entendido pelo aluno como um sucesso durante os experimentos e validações.

Apesar de o escopo ter sido limitado para que fosse viável o desenvolvimento do projeto pelo aluno, algumas melhorias do projeto ficaram incompletas:

- Aplicativo móvel: seria preferível a usar um computador, uma vez que é mais razoável que os praticantes do esporte carreguem um *smartphone* a um computador;
- Uso de manobras não simuladas: mesmo que o aluno não tivesse sofrido uma lesão que impossibilitasse a execução de manobras, a quantidade de manobras que seria possível cadastrar ainda seria limitada à habilidade do aluno. Idealmente, o projeto deveria ser expandido com o auxílio de atletas experientes;
- Pontuação intuitiva: a pontuação fornecida pelo algoritmo de classificação poderia ser ajustada para ser intuitiva, ou seja, recompensar com notas boas manobras bem executadas e deixar claro que notas ruins se devem à manobras mal-executadas;
- Análise de erros comuns nas manobras: potencialmente, o sistema de cadastro de manobras poderia ser expandido para incluir manobras executadas com erros comuns de principiantes, de forma a ajudá-los a reconhecer problemas de execução e como resolvê-los.

## REFERÊNCIAS

AMORUSO, M. J. **Euler Angles and Quaternions in Six Degree of Freedom Simulations of Projectiles - Technical Report (DTIC Document, Fort Belvoir).** 1996.

DIEBEL, J. **Matrix.** 2006. Ed. 58, p15.

EULER, L. **Novi Acad. Sci. Petrop.** 1775. Ed. 20, p189.

HAMILTON, W. R. **The Mathematical Papers of Sir William Rowan Hamilton.** 2000.

JAMBERSI, A. B. **A Sutileza dos Quatérnions no Movimento de Rotação de Corpos Rígidos.** Artigos Gerais. Rev. Bras. Ensino Fís. 38 (2). 2016.

JOSE-LUIZ, B. **A tutorial on se (3) transformation parameterizations and on-manifold optimization.** University of Malaga, Tech. Rep. CiteSeerX 10.1.1.468.5407. 2010.

LUM, C. **Computing Euler Angles: Tracking Attitude Using Quaternions.** Media on Youtube, 2020.