

PLATAFORMA ROBÓTICA PARA ANÁLISE BIOMECÂNICA DE NADADORES: CONTRUÇÃO, ACIONAMENTO E CONTROLE EM MALHA ABERTA

Amanda Dylana Dantas Cavalcante¹, Dr. Olympio Cipriano da Silva Filho², Dr. Adalberto Veronese da Costa³

¹Engenharia Elétrica - Universidade Federal Rural do Semi-Árido (UFERSA) – Mossoró – RN – Brasil

²Coordenação de Engenharia Elétrica – Universidade Federal Rural do Semi-Árido (UFERSA) – Mossoró, RN - Brasil

³Departamento de Educação Física – Universidade Estadual do Rio Grande do Norte (UERN) – Mossoró, RN - Brasil

amandadyllana@gmail.com, olympio.cipriano@ufersa.edu.br, Adalberto.ef@gmail.com

Resumo. A análise biomecânica é uma ferramenta muito utilizada em estudos de movimento humano e possui vasta aplicação na área da saúde, educação física e na prática esportiva. A aplicação da tecnologia na análise biomecânica, permite através de softwares e hardwares resultados cada vez mais aprimorados. A Plataforma de Análise Biomecânica será um sistema voltado para a captura de mídia no ambiente da prática de natação, em diferentes ângulos, resultando na coleta de material de qualidade para análise biomecânica. O estudo apresenta a construção do protótipo da Plataforma de Análise Biomecânica - PAB colocando em ênfase as etapas realizadas para a construção de um protótipo, com controle em malha aberta que atende a variáveis de um treino programado, transmitidas via comunicação bluetooth de um aplicativo móvel desenvolvido para sistema operacional *Android*.

Palavras-chave: Natação; Biomecânica; PABApp; PABCar; Robótica.

1. INTRODUÇÃO

Biomecânica é a ciência que estuda o movimento humano e suas características estruturais e funcionais, a análise biomecânica é uma ferramenta que estuda o comportamento motor humano usada em estudos aplicados a saúde ou aperfeiçoamento de performance em práticas de exercício ou esportivas. A partir das variáveis biomecânicas é possível determinar padrões de movimento, identificar características técnicas de modalidades esportivas e identificar limitações em movimentos necessários para vida diária. O conhecimento obtido através dessa ferramenta orienta o ensino pela educação física escolar, com otimização de movimentos, na saúde, na prevenção e reabilitação de atividade de trabalho e do cotidiano das pessoas [1].

A tecnologia associada a prática regular de atividades físicas vem se destacando neste século como alternativa determinante para a saúde e qualidade de vida da população. Observa-se um aumento no mercado móvel relacionando aplicativos de smartphones aos mais diversos tipos de hardwares para o monitoramento da saúde estimulando as pessoas para um estilo de vida saudável. Através da robótica são desenvolvidos dispositivos que impulsionam as pesquisas nesta área. Esses dispositivos robóticos permitem que a prática de tarefas específicas sejam reproduzidas diversas vezes, de forma controlada e confiável, proporcionando consequentemente o aumento da capacidade motora e melhora do desempenho das atividades funcionais [2].

Além disso, é válido o argumento de que para obter resultados precisos, é importante realizar a captura no ambiente de prática do exercício, fazendo necessário o uso de tecnologias portáteis. Com o avanço no desenvolvimento e redução de custo de componentes essenciais na robótica como microcontroladores, sensores e transmissores sem fio, o uso de uma plataforma de medição reduzida torna-se viável e pode oferecer resultados mais adequados a medição da real performance dos movimentos durante a prática de exercícios [3].

Dessa maneira, foi idealizada a Plataforma de Análise Biomecânica – PAB que realizará captura de imagens e vídeos ao lado de uma piscina semiolímpica através de um robô microcontrolado que se movimentará sob trilhos, como suporte para câmeras que realizaram a captura em diferentes ângulos para análises posteriores da mídia armazenada. Para o presente estudo, ainda em fase inicial que tem como objetivo central desenvolver o controle

em malha aberta de uma plataforma robótica que atenda as especificações de um treino de natação configuradas via aplicação móvel para sistema operacional *Android* transmitidas via comunicação bluetooth.

2. DESENVOLVIMENTO

Nesse capítulo é apresentada a Plataforma de Análise Biomecânica composta por PABCar que representa a parte física da plataforma e pelo PABApp que é a interface da plataforma. E como foco deste trabalho também é apresentado o protótipo inicial em fase de controle da plataforma.

2.1. Plataforma de Análise Biomecânica – PABCar

O PABCar é um equipamento microcontrolado que serve como suporte para câmeras com o propósito de realizar capturas de mídia para análise biomecânica em ângulos previamente determinados. O uso da plataforma é sob trilhos próximos a borda de piscinas semiolímpica ou olímpica, movimentando-se em dois sentidos horizontais (ida e volta) na distância(m) e tempo de volta(s) programáveis pelo aplicativo PABApp conforme ilustra a Figura 1.

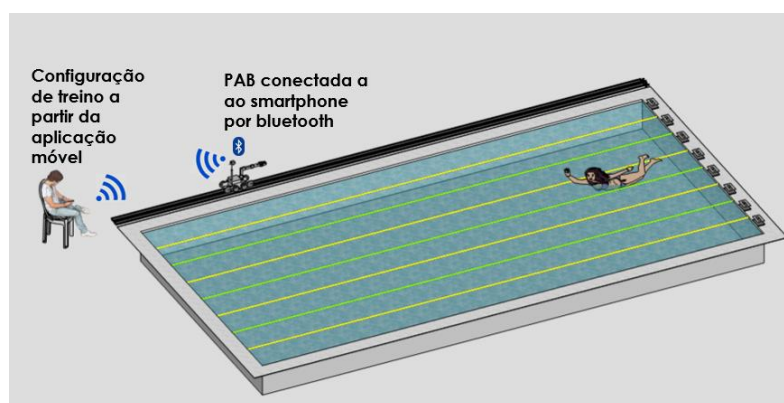


Figura 1. Uso da Plataforma de Análise Biomecânica. (Autoria Própria)

Para utilizar a PABCar é necessário instalar a aplicação móvel desenvolvida para sistemas *Android* no smartphone e parear o dispositivo com o módulo *bluetooth* da plataforma. Feito isso, o usuário pode configurar o treino de natação e enviar as especificações para a plataforma via comunicação *bluetooth*.

A plataforma funcionará com total autonomia com alimentação por bateria recarregável e comunicação sem fio por *bluetooth* com o *smartphone* e comunicação Wi-Fi com as câmeras através de um sistema de transmissão *First Person View – FPV* (Visão de Primeira de Pessoa), que permite o acesso às mídias capturadas em tempo real pelo *smartphone*, tornando opcional o armazenamento.

2.2. Aplicação Móvel: PABApp

A interface do PABCar é uma aplicação móvel desenvolvida externamente para o sistema operacional *Android*. A ideia é transmitir através desta, uma *string* com as variáveis do treino programado para a PABCar via conexão *bluetooth*, uma vez que o usuário estiver com o *smartphone* conectado com a plataforma.

A Figura 2 exibe as telas dispostas no aplicativo. Quando iniciado é mostrada a tela de apresentação do PABApp, em seguida é aberta uma tela que verifica se o *bluetooth* do dispositivo está habilitado e apresenta as especificações de *bluetooth* do PABCar. Após conectar corretamente os dispositivos, a tela com o formulário de treino de natação é liberada.

As especificações de treino são:

- Tamanho da piscina [m]: parâmetro que define se a piscina de treino é semiolímpica (25 metros) ou olímpica (50 metros);
- Distância a ser nadada [m]: distância em metros que o nadador fará sem interrupções, equivalente a uma série;
- Quantidade de séries: Quantidade de vezes que o nadador executará o exercício;
- Tempo de volta [s]: tempo em segundos no qual o nadador realizará uma série;
- Intervalo [s]: intervalo definido em segundos entre cada série.

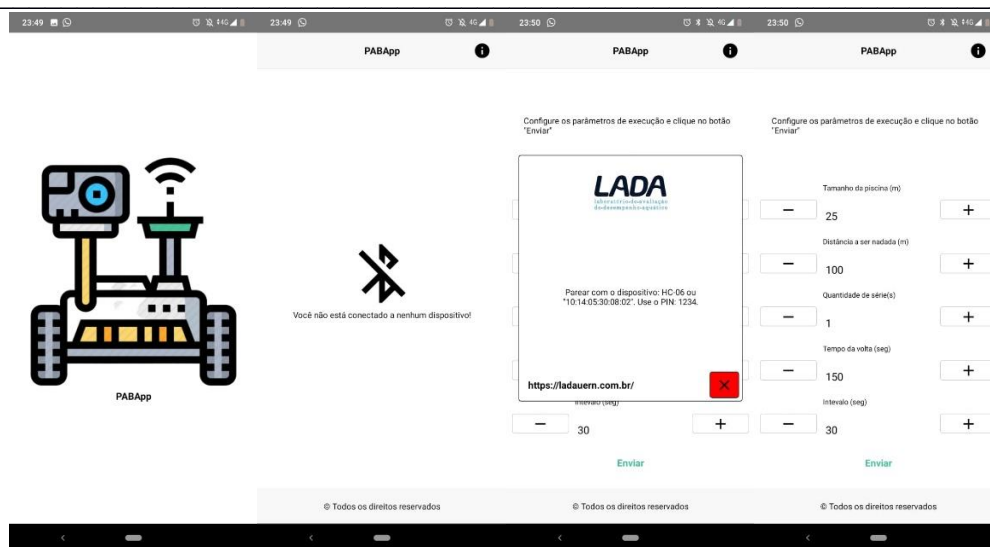


Figura 2. Aplicativo PABApp. (Autoria Própria)

2.3. Protótipo da Plataforma de Análise Biomecânica

Para compreender as necessidades de um projeto e executa-lo, faz necessário realizar estudos prévios. É comum o desenvolvimento de protótipos em fase iniciais do projeto, do qual pode-se explorar melhor todas as ideias envolvidas no processo e obter experiência antes da execução real do projeto estudado. O protótipo da PABCar é representada por um carrinho (Figura 4) construído a partir de componentes utilizados em robótica listados na Quadro 1.

Quadro 1. Principais componentes do protótipo da PABCar. (Autoria Própria)

1. Motor de Engrenagem	2. Driver Ponte H – L298	3. Arduino UNO
		
4. Base de Acrílico, jumpers e alimentação 5V	5. Sensor de Velocidade Encoder – 5mm	6. Módulo Bluetooth HC-06
		

Nota: As imagens são ilustrativas e foram obtidas no Google Imagens.

Seguindo a ordem descrita na Tabela 1, tem-se:

- Motor de Engrenagem: trata-se de um equipamento constituído essencialmente de um motor elétrico de Corrente Contínua - CC com seu eixo ligado a um sistema de redução formado por engrenagens. Com baixa potência, fornecem grande força motriz em baixa velocidade. O sistema de redução para o motor adotado é de 1/48;
- Driver Ponte H L298: é um driver utilizado para alimentar até duas cargas, é normalmente utilizada para variar o sentido da corrente e controlar sua potência, o funcionamento dele é apresentado no diagrama da

Figura 3.

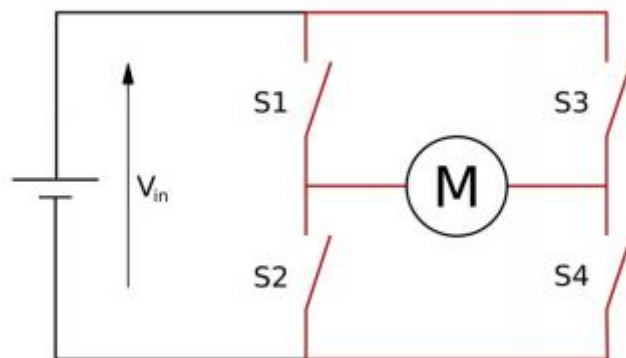


Figura 3. Circuito de uma ponte H. [4]

Quando acionadas as chaves S1 e S2, a corrente flui em um sentido. Quando acionadas as chaves S2 e S3, a corrente flui no sentido oposto. Acionando S1 e S2 ou S3 e S4 nada acontece. Dessa forma, é mantido um par de chaves fixo acionado, acionando outro par quando necessário mudar o sentido do eixo de rotação do motor alimentado pela ponte;

- Arduino Uno: é uma plataforma de prototipagem eletrônica de hardware livre e de placa única, projetada com um microcontrolador Atmel AVR com suporte de entrada/saída embutido, uma linguagem de programação padrão, a qual tem origem em Wiring, e é essencialmente C/C++ para desenvolver *sketch* no ambiente programável IDE. Esse componente é essencial para o funcionamento do protótipo, pelo qual é alimentado todos os outros componentes listados;
- Sensor de Velocidade Encoder: Chave Óptica 5mm, é um módulo eletrônico que pode ser utilizado para fazer a contagem do número de rotações de um disco encoder acoplado a um motor. O sensor é usado para obter a velocidade do eixo do motor em rpm, a partir da qual pode-se obter outras informações. Foi utilizado no protótipo para mensurar a velocidade máxima média da plataforma a partir de uma *sketch* programada no IDE do Arduino compondo um sistema de calibração de velocidade;
- Módulo Bluetooth: O módulo de série HC-06 permite o controle remoto via comunicação Bluetooth qualquer projeto com microcontrolador. Utilizado na protótipo para realizar a comunicação entre o protótipo e qualquer smartphone que possua a aplicação móvel desenvolvida para o projeto em questão.

A Figura 4 apresenta a montagem final do primeiro protótipo da Plataforma, visando a partir deste o desenvolvimento de sistema de controle de locomoção que possa atender ao projeto futuramente.

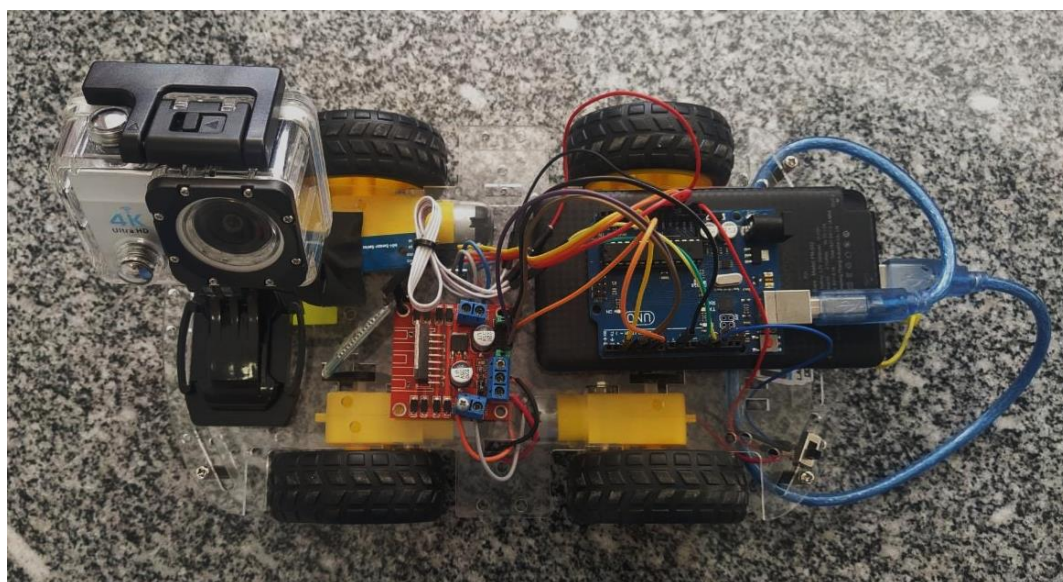


Figura 4. Primeiro Protótipo. (Autoria Própria)

Foram obtidas algumas especificações técnicas para o protótipo, presentes na Quadro 2.

Quadro 2. Especificações do protótipo da P.A.B. (Autoria Própria)

Especificações Técnicas

Tensão de entrada [V]	5
Quantidade de motores	4
Trações	2
Tensão por tração [V]	3.5
Corrente por tração [A]	1.17
Velocidade máxima média [m/s]	0.3202
Outras Especificações	
Diâmetro da Roda [m]	2x0.031
Comprimento da base [cm]	30
Altura da base [cm]	6

Nota: Resultados obtidos em testes de bancada.

3. CONTROLE

As etapas que compõem o controle da plataforma consiste em acionamento e controle de velocidade de 4 motores CC com um micro controlador através da função razão cíclica e saída em Modulação por Largura de Pulso - PWM.

O controle discriminado ao longo desta seção é de malha aberta, sem realimentação do sistema, sendo realizada externamente através de uma *sketch*, a calibragem da velocidade máxima média da PABCar, necessitando de atualização manual da velocidade máxima média da PAB periodicamente ou diante de qualquer alteração realizada no equipamento.

3.1. Acionamento dos Motores de Engrenagem

O acionamento dos motores é realizado através da alimentação fornecida pelo driver ponte H que recebe 5V da placa Arduino. A ponte H possui 4 saídas destinadas a alimentação de cargas, duas para cada tração, 4 entradas para configuração dos sentidos da corrente fornecida, duas para cada tração, ligadas ao microcontrolador por onde é realizado o controle das cargas.

O acionamento foi realizado para plataforma em trilho, que permite a locomoção da plataforma em dois sentidos. Na ponte H, as entradas 1 e 2 controlam a tração frontal e as entradas 3 e 4 controlam a tração traseira da plataforma. De maneira tal que os dois sentidos (ida e volta) são acionados de acordo com a Quadro 3.

Quadro 3. Configuração de sentidos 1 e 2 no driver ponte H. (Autoria Própria)

Entradas	Sentido 1 – Ida	Sentido 2 – Volta
E1	0	1
E2	1	0
E3	0	1
E4	1	0

Nota: Configuração de sentidos adotada para o sentido 1 e 2.

A Figura 5 apresenta o esquema de ligação implementada para o acionamento dos 4 motores de engrenagem, na qual cada tração constam de dois motores ligados em paralelo tornando possível realizar o controle com o driver ponte H L298. As entradas da L298 são ligadas a saídas PWM da placa Arduino por onde pode-se variar os níveis de tensão aplicada as duas trações.

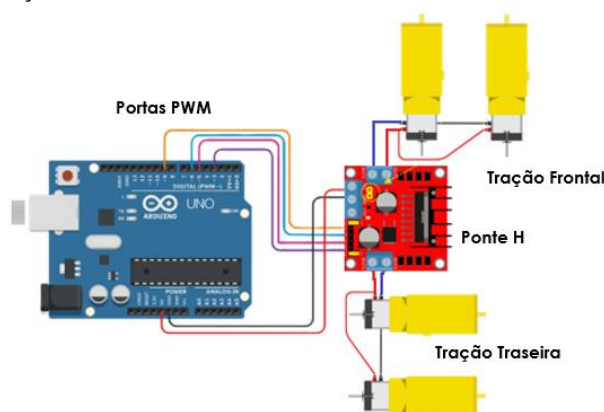


Figura 5. Esquema de ligação para o acionamento dos motores CC. (Autoria Própria)

Além disso, a configuração atual para a alimentação das cargas não contam com alimentação externa exclusiva ao driver, o que apesar de limitar a tensão enviada a ponte H, proporciona economia de energia para realização dos testes na fase de prototipagem.

3.1. Calibragem da Velocidade Máxima Média com Sensor Encoder

O sistema de calibragem de velocidade máxima do protótipo tem como objetivo obter uma média de velocidades medidas a cada 20 segundos de funcionamento, intervalo no qual são obtidas 20 medidas de velocidades em metros por segundo. A mensuração da velocidade é realizada através de um sensor de velocidade encoder que contabiliza pulsos a partir de um disco encoder conectado ao eixo do motor de engrenagem.

As medidas de velocidade são obtidas inicialmente em rotações por minuto a partir da Equação 3.1 dentro de um ciclo que gera um resultado por segundo com a quantidade de pulsos sentidos pelo sensor em um segundo.

$$RPM = \frac{\frac{60 \cdot 1000}{ppv}}{(ta - \text{millis}) \cdot \text{pulsos}} \quad [\text{Eq. 3.1}]$$

Onde,

RPM: Rotações por Minuto;

ppv: Quantidade de aberturas do disco encoder;

millis: Contador de tempo em milissegundos;

ta: tempo contabilizado anteriormente;

pulsos: Quantidade de vezes que o sensor foi sensibilizado.

Após mensurado o valor em rpm, obtêm-se ainda dentro do ciclo o valor corresponde em metros por segundo para a o valor em rpm mensurado, conforme mostra a Equação 3.2.

$$V = \frac{RPM \cdot 2\pi \cdot \text{raio do pneu}}{60} \quad [\text{Eq. 3.2}]$$

Dessa maneira, após contabilizado n valores de velocidade em 20 segundos definidos aleatoriamente é obtido o valor médio de velocidade para o intervalo pela Equação 3.3.

$$V_{\text{média}} = \frac{\sum V \text{ obtidas em } t \text{ definido}}{t \text{ definido (20s)}} \quad [\text{Eq. 3.3}]$$

A implementação do sistema de calibragem é mostrado na Figura 6, na qual é colocado um sensor de velocidade encoder em um dos motores do protótipo para a coleta de dados via Monitor Serial disposto no ambiente programável da placa Arduino IDE com código apresentado no Anexo 2 deste trabalho.

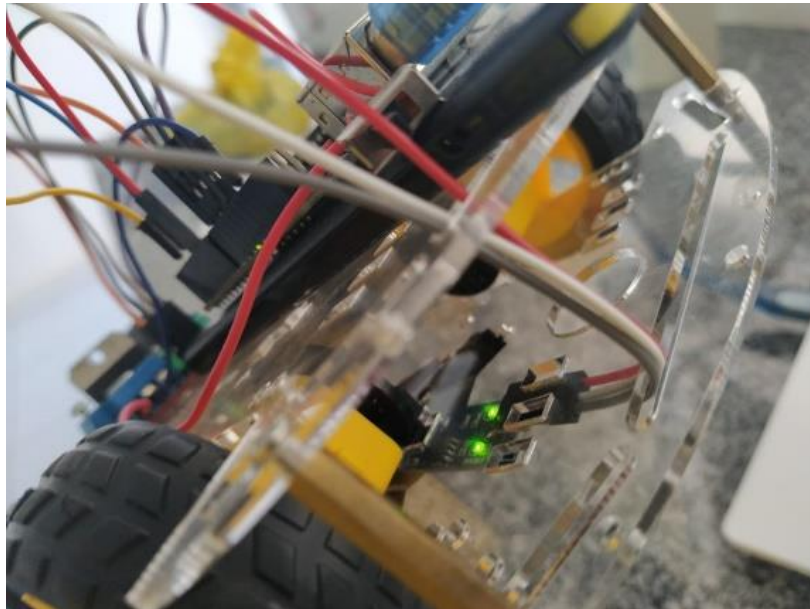


Figura 6. Sensor de velocidade encoder implementado a protótipo. (Autoria Própria)

3.2. Controle de Velocidade por razão cíclica

A variação da velocidade dos motores é realizada a partir da variação da tensão aplicada ao driver ponte H que alimenta os motores. A velocidade do eixo do motor é proporcional a tensão aplicada, dessa forma é possível variar a velocidade aplicando o percentual de tensão equivalente a velocidade pedida. Através do uso de portas de saída PWM no Arduino, com a função razão cíclica obtém-se a variação de tensão aplicada ao driver em uma faixa de 0 a 255, onde 255 é equivalente a 100% da tensão, por onde obtém-se a velocidade máxima no eixo dos motores ligados ao driver ponte H para o maior nível de tensão fornecida.

A interface do equipamento solicita na configuração de treino, o tempo de volta (tempo equivalente para nadar a distância também programada), a partir desse tempo é calculada via sketch a velocidade necessária para a plataforma realizar a distância em metros no treino programado. Uma vez calibrada a velocidade média máxima da plataforma, foi encontrado um fator que varia de 0 a 1 em função da velocidade calculada para o tempo pedido, de maneira tal que para tempos maiores que t_{min} (tempo no qual a plataforma trabalha com velocidade máxima), o fator reduza a velocidade máxima. A Equação 3.4 apresenta o fator de velocidade.

$$FV = \frac{Vc}{Vmax} \quad [Eq. 3.4]$$

Onde,

Vc: Velocidade calculada;

Vmax: Velocidade máxima medida;

FV: Fator de Velocidade.

O fator de velocidade deve ser menor que um 1, em função de tempos maiores que o tempo mínimo para a capacidade da plataforma. No entanto, para que se mantenha o funcionamento, quando o fator de velocidade for maior que 1, a plataforma se movimentará em velocidade máxima, considerando o fator de velocidade igual a 1.

O percentual de tensão que sai para a ponte H é ajustada pelo fator de velocidade conforme é mostrada na Equação 3.5.

$$Vpwm = FV \cdot 255 \quad [Eq. 3.5]$$

Além do controle em módulo da velocidade através do razão cíclica aplicada, também é preciso atentar ao fato de que a cada piscina nadada a plataforma precisa mudar o sentido de rotação dos eixos dos motores acoplados. Então foi implementado ao código uma variável que informa de acordo com a distância a ser nada, o número de piscinas equivalente a essa distância. Para a implementação do código, é completada uma volta cada vez que a plataforma precisa ir e voltar, sendo atribuído dois sentidos por piscina. O número de piscinas pode ser obtido pela Equação 3.6.

$$Np = \frac{d}{S_p} \quad [\text{Eq. 3.6}]$$

Onde,

Np : Número de piscinas;

d : Distância a ser nadada;

S_p : Tamanho da piscina em metros.

Dessa forma pode ser calculado a quantas voltas a plataforma terá que realizar para cumprir a distância a ser nadada programada no treino pela Equação 3.7.

$$v = \frac{Np}{2} \quad [\text{Eq. 3.7}]$$

Assim, é realizada o número de séries com os intervalos definidos entre cada completando o exercício programado.

4. VERIFICAÇÃO DE VELOCIDADE

Afim de verificar os resultados obtidos foram geradas tabelas para análise. A Quadro 4 apresenta as médias de velocidade obtidas pelo sistema de calibragem para as variações de tensão de alimentação das cargas.

Quadro 4. Velocidades alcançadas para as razões cíclicas. (Autoria Própria)

Razão Cíclica	Velocidade média [m/s]
100%	0,3202
75%	0,2423
50%	0,1705
25%	0,0834

Nota: Valores obtidos no sistema de calibragem.

Os valores obtidos pelo sistema de calibragem são muito próximos do valores ideais, porém sabe-se que há fatores que influenciam na velocidade do protótipo, como peso o do próprio equipamento e atritos encontrados de uma medição para a outra. O atrito por exemplo, é um fator de influência significativa, isso se mostra na Quadro 5, que apresenta medições realizadas com a plataforma posicionada a uma altura que permitisse o giro das rodas livre.

Quadro 5. Velocidade média obtida no sensor encoder. (Autoria Própria)

Médias coletadas	Velocidade média máxima [m/s]
Média 1	0,6568
Média 2	0,6602
Média 3	0,5872
Média 4	0,6783

Nota: Valores obtidos para a plataforma sem atrito.

Percebe-se que a retirar o fator atrito elevou a velocidade acima do dobro das velocidades obtidas na Tabela 4.

Para verificar a coerência no dados coletados pelo sistema de calibragem foram realizados alguns testes de bancada, manualmente, para comparar com os dados obtidos na calibragem. Os resultados obtidos são mostrados na Quadro 6.

Quadro 6. Velocidade obtida no teste de bancada x velocidade medida no sensor encoder. (Autoria Própria)

Mensurações	Teste de Bancada [m/s]	Sistema de Calibragem [m/s]
1	0,3397	0,3895
2	0,3382	0,3579
3	0,3275	0,4134

Nota: Valores obtidos para a plataforma com atrito em um espaço de 3 metros.

Embora perceba-se diferença entre os valores coletados e os valores calculados pelo teste de bancada, é vista que essa diferença pode ocorrer devido a erros relacionados com a paralaxe, e devido a possíveis erros no sistemas de calibragem.

Na Quadro 7 foram obtidos os valores de tempo em segundos que a plataforma leva para percorrer 25 metros,

em chão cerâmico sem trilhos. Há uma variação considerável de segundos entre as mensurações, essas variações responderam na distância que a plataforma irá percorrer, uma vez que, ela realizar trabalho no tempo exato que for enviado pelo PABApp no treino programado. Então em velocidades menores, a plataforma poderá não percorrer a distância completa solicitada, de maneira análoga, em velocidades maiores a plataforma poderá percorrer uma distância maior.

Quadro 7. Tempo que a plataforma leva para fazer 25 metros para as velocidades medidas. (Autoria Própria)

Mensurações	Velocidades [m/s]	Distância [m]	Tempo [s]
1	0,3895	25	64,1848
2	0,3579	25	69,8519
3	0,4134	25	60,4741

Dessa forma, é válido atentar a importância de realizar com um periodicidade curta testes de velocidade média, afim de atenuar as variações mostradas na Quadro 7, pois quanto mais preciso for a velocidade média máxima inserida na plataforma, mais preciso será o funcionamento da PABCar. Visando orientar o melhor uso da plataforma, foi desenvolvido o fluxograma de rotina mostrado na Figura 7.

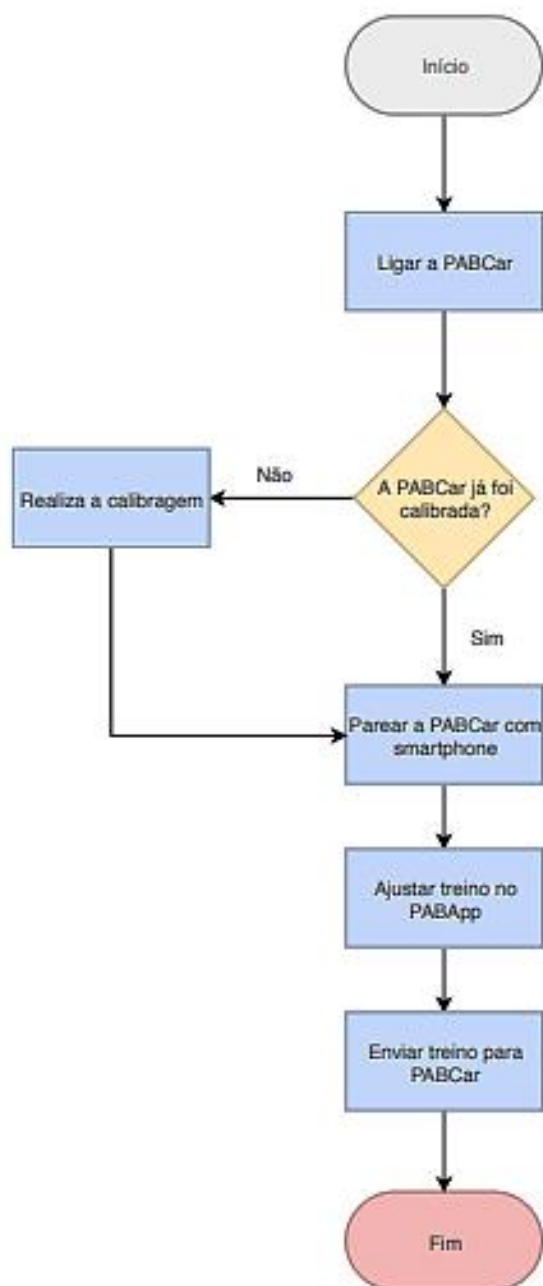


Figura 7. Rotina de Uso da Plataforma de Análise Biomecânica. (Autoria Própria)

O aplicativo PABApp permite parar o treino enviado, interrompendo o funcionamento da plataforma simultaneamente, cabendo ao usuário refazer a programação do treino e enviar novamente.

5. CONSIDERAÇÕES FINAIS

A aplicação de tecnologias na área da saúde possibilitam à ferramentas como a análise biomecânica, avanço em pesquisas relacionadas ao movimento humano, como também torna possível o aprimoramento de atividades motoras, a captura de imagens, vídeos de qualidade possibilitam futuramente o emprego de softwares de análise ou ao olho humano, percepção sobre determinadas situações não observadas pelo treinador.

O trabalho mostrado representa o início da construção de um projeto e o mesmo consta de muitas etapas a serem realizadas. No que diz respeito ao que foi trabalhado, é no mínimo interessante realizar futuramente para a plataforma um sistema de controle de malha fechada, que possa corrigir as variações de velocidade constantemente, de maneira tal, que possa proporcionar maior precisão para a plataforma. No que diz respeito a outras etapas, pode-se citar as seguintes:

- Implementar um hardware robusto que atenda aos tempos reais em treinos de natação;
- Implementar um sistema de transmissão que possibilite a comunicação Wi-Fi das câmeras ao dispositivo receptor;

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BERNARDINA, Gustavo Ramos Dalla. **UTILIZAÇÃO DE CÂMERAS DE AÇÃO PARA ANÁLISE CINEMÁTICA TRIDIMENSIONAL DO MOVIMENTO HUMANO**. 2016. 129 f. Dissertação (Mestrado) - Curso de Educação Física, Universidade Federal de Viçosa, Viçosa Mg, 2016. Cap. 1.
- [2] ARAÚJO, R. C. **Desenvolvimento e avaliação de um sistema de auxílio à reabilitação motora do membro superior após acidente vascular encefálico**. 117 f. Tese de Doutorado em Bioengenharia. Escola de Engenharia da Universidade Federal de Minas Gerais, Belo Horizonte, 2011.
- [3] GAFFNEY, **An automated calibration tool for high performance wireless inertial measurement in professional sports**. sensors, 2011 IEEE, pp. 262-265.
- [4] SEIXAS, Falcondes J. M. et al. **PROTÓTIPO DE UM VEÍCULO COM ESTEIRAS A PARTIR DE MATERIAIS RECICLADOS**. In: XIV SAFETY, HEALTH AND ENVIRONMENT WORLD CONGRESS, 14., 2014, Cubatão. DOI 10.14684. Cubatão: Copec, 2014. v. 1, p. 20 - 23.

7. ANEXO

Anexo A - Código de controle de malha aberta da Plataforma de Análise Biomecânica

```
//Controle de Velocidade com Arduino - Amanda Dylana Dantas Cavalcante
//Projeto de Engenharia Elétrica - UFERSA
//Laboratório de Análise de Desempenho Aquático - LADA
//Orientador: Olympio Cipriano
//Corientador: Adalberto Veronese

#include <Arduino.h>
#include <SoftwareSerial.h>

SoftwareSerial hc06(12,13);

//variaveis correspondentes a ponteH ( Portas PWM) - 3 e 5 Frontal, 6 e 9 Traseira

int IN1 = 3;
int IN2 = 5;
int IN3 = 6;
int IN4 = 9;

// COMANDS POSITION
// 0 STATUS
// 1 TAMANHO_DA_PISCINA
// 2 DISTAN
// 3 QTD_SERIE
// 4 TMP_VOLTA
// 5 INTERVALO
```

```

const size_t bufferSize = 6; // Quantidade de variáveis
int form[bufferSize];
String cmd=""; // A string que vai armazenar os dados lidos no bluetooth

// Configurações
int DP = 0; // distância a ser nadada
float vmedia = 0; // metros por segundo – sistema de calibragem
float tmin = 0;
float t = 90; // tempo de volta
float tintervalo = 30; // form 5
// 0 START, 1 stop, // parar o carrinho
int STATUS = 1; // ligar o carrinho

//Configuração dos Pinos
void setup() {
  // Pinagem da Ponte H
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  Serial.begin(9600); //velocidade do monitor serial
  hc06.begin(9600); //velocidade do monitor serial
}

void configure(){

  //Ler o dado do modulo HC06
  while(hc06.available()>0){
    cmd+=(char)hc06.read();
  }

  if(cmd!=""){ // se a string n estiver vazia fazer

    char *p = strtok((char *)cmd.c_str(), ","); // converte a string para um array de caracteres
    size_t index = 0; // variavel de indice

    while (p != NULL && index < bufferSize) { // enquanto p n for nulo e indice for menor que o array
//transforma para inteiro
      form[index++] = atoi(p); //letra em numero se for válida
      p = strtok(NULL, ","); // limpa o array
    }

    STATUS = form[0];
    DP = form[2]/form[1]; // numero de piscinas
    vmedia = 0.3202; // metros por segundo
    tmin = form[2]/vmedia;
    t = form[4];
    tintervalo = form[5];

    // protecao de tensao para garantir que seja trabalhado com valores aceitaveis
    if(t < tmin){
      t = tmin;
    }

    cmd=""; // zera os comandos do bluetooth
  }
}

//Essa função converte segundos para milissegundo
float seg2mills(float seg){
  return seg*1000;
}

```

```

}

int v2t(float f){ //velocidade para tensão
    return int(f * 255);
}

void custom_delay_with_stop_condition(int tempo){ //checa a cada 1 segundo se n chegou comando
//do bluetooth, durante um comando
    configure();

    if(STATUS==0 && tempo > 0){ se tiver ligado e tempo do comando for maior que zero
        delay(seg2mills(1)); // executa por 1 segundo
        custom_delay_with_stop_condition(tempo-1);
    }
}

void sentido1(float t){
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);

    custom_delay_with_stop_condition(t);
}

void sentido2(float t){
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);

    custom_delay_with_stop_condition(t);
}

void intervalo(float t){
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);

    custom_delay_with_stop_condition(t);
}

void loop() {
    configure();

    if (STATUS == 0 && form[3] > 0){

        float v = form[2]/t;
        float f = v/vmedia
        int tensao = v2t(f);

        analogWrite(IN1, tensao);
        analogWrite(IN2, tensao);
        analogWrite(IN3, tensao);
        analogWrite(IN4, tensao);

        // 2 = quantidade de sentidos
        int repeticoes_sentidos = DP/2;

        while (repeticoes_sentidos > 0) {
            sentido1(t/DP);
            sentido2(t/DP);

```

```
    repeticoes_sentidos = repeticoes_sentidos - 1;
}

intervalo(tintervalo); // descanso/intervalo
form[3] = form[3] - 1;
}
}
```