

**UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS**

**UNIDADE ACADÊMICA DE GRADUAÇÃO**

**CURSO DE ENGENHARIA DA COMPUTAÇÃO**

**LUCAS CARVALHO GOULART**

**MODELO DE SOFTWARE DE REPRODUÇÃO DE VOLTAS  
EM SIMULADORES DE AUTOMOBILISMO**

**São Leopoldo, RS**

**2022**

**LUCAS CARVALHO GOULART**

**MODELO DE SOFTWARE DE REPRODUÇÃO DE VOLTAS  
EM SIMULADORES DE AUTOMOBILISMO**

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Bacharel em Engenharia da Computação pelo curso de Engenharia da Computação da Universidade do Vale do Rio dos Sinos – UNISINOS.

Orientador:

Prof. Dr. Rodrigo Marques de Figueiredo

São Leopoldo, RS

2022

## RESUMO

Ao longo dos anos os simuladores de automobilismo foram se tornando mais acessíveis e mais fiéis às físicas da pista. Com esta fidelidade, a necessidade de realizar ajustes na aerodinâmica, nos freios e na suspensão dos carros se tornou indispensável para melhor rendimento também nos simuladores. A dificuldade em encontrar o melhor acerto ocorre também fora dos simuladores, pois a cada ajuste é necessário realizar uma volta de teste que verifique se o carro apresentou ou não um melhor resultado. O trabalho proposto apresenta um modelo de *software* capaz de repetir voltas previamente configuradas, diminuindo assim os fatores que podem interferir na análise dos dados durante as voltas de teste de ajustes do carro. Utilizando a telemetria disponibilizada pelos simuladores, o trabalho apresenta um sistema de *software in the loop* que pilota automaticamente o carro de fórmula 1 no circuito de Interlagos no simulador F1 2021. Os dados de telemetria resultantes das voltas são apresentados e comparados com a telemetria de um voluntário para visualização das interferências da pilotagem do ser humano.

**Palavras-chaves:** simuladores; automobilismo; aerodinâmica; telemetria; *software in the loop*.

## LISTA DE FIGURAS

Figura 1 – PXG F1 Telemetry . . . . .	11
Figura 2 – Principais pontos de uma curva . . . . .	13
Figura 3 – Tipos de aproximação . . . . .	13
Figura 4 – Circuito Road America . . . . .	14
Figura 5 – Subesterço e sobresterço . . . . .	14
Figura 6 – Ângulo de deslizamento . . . . .	15
Figura 7 – Simplificação para ângulo neutro . . . . .	20
Figura 8 – Divisão do software em 3 etapas . . . . .	21
Figura 9 – Etapas do desenvolvimento do sistema de controle . . . . .	23
Figura 10 – Mapa de Interlagos . . . . .	24
Figura 11 – Fluxos de teste . . . . .	25
Figura 12 – Interface de calibração . . . . .	28
Figura 13 – Interface principal . . . . .	28
Figura 14 – Controle reconhecido no Windows . . . . .	33
Figura 15 – Traçado da volta de referência . . . . .	34
Figura 16 – Volta automática com limitação de 70 km/h . . . . .	35
Figura 17 – Volta automática com limitação de 70 km/h - Curvas 1 e 2 . . . . .	35
Figura 18 – Repetibilidade com limitação de 70 km/h . . . . .	36
Figura 19 – Repetibilidade com limitação de 70 km/h - Curva 2 . . . . .	36
Figura 20 – Repetibilidade - Curvas 1 e 2 . . . . .	38
Figura 21 – Repetibilidade manual - Curvas 1 e 2 . . . . .	38
Figura 22 – Volta do voluntário - Curvas 1 e 2 . . . . .	40
Figura 23 – Volta do <i>software</i> - Curvas 1 e 2 . . . . .	41
Figura 24 – Diferentes configurações sobrepostas - Curvas 1 e 2 . . . . .	41
Figura 25 – Diferença de velocidade . . . . .	42

## LISTA DE QUADROS

Quadro 1 – Dados de telemetria . . . . .	18
Quadro 2 – Critérios x simuladores . . . . .	19
Quadro 3 – Interface: Aquisição - Sistema de controle . . . . .	22
Quadro 4 – Marcha X Velocidade . . . . .	32
Quadro 5 – Teste de frequência de aquisição . . . . .	34
Quadro 6 – Tempos de volta com limitação de 70 km/h . . . . .	37
Quadro 7 – Tempos de volta . . . . .	37
Quadro 8 – Tempos de volta manual . . . . .	38
Quadro 9 – Voltas automáticas com diferentes configurações . . . . .	39
Quadro 10 – Voltas manuais do voluntário . . . . .	40
Quadro 11 – Voltas manuais do voluntário . . . . .	41
Quadro 12 – Voltas manuais do voluntário . . . . .	42
Quadro 13 – Tempos de volta em Hungaroring . . . . .	43

## LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i> (Interface de Programação de Aplicativos)
DRS	<i>Drag Reduction System</i> (Sistema de Redução de Arrasto)
F1	Fórmula 1
IA	Inteligência artificial
KERS	<i>Kinetic Energy Recovery Systems</i> (Sistema de Recuperação de Energia Cinética)
UDP	<i>User Datagram Protocol</i> (Protocolo de datagrama do usuário)
UNISINOS	Universidade do Vale do Rio dos Sinos

## LISTA DE SÍMBOLOS

$a_{cp}$	Aceleração centrípeta
$wb$	<i>Wheelbase</i> (Distância entre eixos)
$v$	Velocidade
$\Theta_1$	Ângulo da trajetória na curva
$\Theta_2$	Ângulo das rodas dianteiras

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>9</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>10</b>
<b>2.1</b>	<b>Jogos e simuladores</b>	<b>10</b>
<b>2.2</b>	<b>Telemetria</b>	<b>11</b>
<b>2.3</b>	<b>Pilotagem</b>	<b>12</b>
2.3.1	Curvas	12
2.3.2	Subesterço e sobresterço	14
2.3.3	Deslizamento na frenagem	15
<b>2.4</b>	<b><i>Software in the loop</i></b>	<b>15</b>
<b>2.5</b>	<b>Trabalhos relacionados</b>	<b>16</b>
2.5.1	<i>Telemetry-based optimisation for user training in racing simulators</i>	16
2.5.2	<i>Humans and autonomous driving in a professional racing simulator</i>	16
2.5.2.1	<i>Adaptive AI in a racing simulator</i>	17
<b>3</b>	<b>METODOLOGIA</b>	<b>18</b>
<b>3.1</b>	<b>Dados disponíveis via telemetria</b>	<b>18</b>
<b>3.2</b>	<b>Escolha do simulador</b>	<b>19</b>
<b>3.3</b>	<b>Software</b>	<b>21</b>
3.3.1	Aquisição dos dados	21
3.3.2	Controle	23
3.3.3	Sistema de controle	23
<b>3.4</b>	<b>Testes</b>	<b>24</b>
3.4.1	Validação do sistema de controle	25
3.4.2	Validação da repetibilidade	26
3.4.3	Validação dos objetivos do projeto	26
<b>4</b>	<b>IMPLEMENTAÇÃO</b>	<b>27</b>
<b>4.1</b>	<b>Controlador</b>	<b>27</b>
<b>4.2</b>	<b>Interfaces da aplicação</b>	<b>27</b>
<b>4.3</b>	<b>Objeto de aquisição</b>	<b>29</b>
<b>4.4</b>	<b>Sistema de armazenamento da volta</b>	<b>29</b>
<b>4.5</b>	<b>Seguidor de linha</b>	<b>30</b>
<b>4.6</b>	<b>Piloto automático</b>	<b>30</b>
<b>5</b>	<b>ANÁLISE DOS RESULTADOS</b>	<b>33</b>

<b>5.1</b>	<b>Validação do controle virtual</b> . . . . .	<b>33</b>
<b>5.2</b>	<b>Validação da repetibilidade</b> . . . . .	<b>36</b>
<b>5.3</b>	<b>Validação dos objetivos do projeto</b> . . . . .	<b>39</b>
5.3.1	Comparação de tempos . . . . .	40
5.3.2	Comparação de deslizamento . . . . .	40
5.3.3	Comparação de velocidade . . . . .	42
5.3.4	Expansão da aplicação . . . . .	43
<b>6</b>	<b>CONCLUSÃO</b> . . . . .	<b>44</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> . . . . .	<b>45</b>

# 1 INTRODUÇÃO

Corridas automotivas existem desde 1894, ano em que foi organizada a primeira competição. Inicialmente os carros andavam a velocidades de aproximadamente 16 km/h, atualmente, em competições como a Fórmula 1, os carros passam de 330 km/h nas retas. O aumento significativo na velocidade final dos carros vem de evoluções de engenharia que melhoraram a performance das unidades de potência e de transmissão. Entretanto uma corrida não consiste apenas de retas, sendo necessários diversos estudos de aerodinâmica e de soluções para os sistemas de freio e de suspensão dos carros, afim de que possa ser carregada a maior velocidade de curvas possível.

Diversos componentes de um carro de corrida atual são ajustáveis, como por exemplo as asas dianteiras e traseiras, que possuem configurações de maior e menor pressão aerodinâmica. Encontrar o melhor acerto destes componentes exige horas de simulações e testes. Esta dificuldade em configurar o carro ocorre também em simuladores de corrida, que estão cada vez mais fiéis as realidades das pistas.

Em simuladores, o processo de ajuste consiste em alterar um elemento do carro e realizar uma volta de teste para analisar o impacto no tempo de volta. É imprescindível que cada elemento seja testado individualmente para que não sejam camufladas melhorias em um ponto devido a um outro elemento que gerou um impacto negativo durante o mesmo teste.

Durante os testes de um ajuste diferente, existem diversos fatores que podem gerar uma interferência, fazendo com que não se possa apenas olhar para o tempo de volta. O fator humano, da pilotagem, é o que mais pode afetar um teste, visto o critério de repetibilidade e de chegar aos limites da nova configuração.

O objetivo principal do presente trabalho é, através de um sistema de controle capaz de realizar repetidas voltas sobre o mesmo traçado em um simulador, minimizar os fatores de interferência durante as voltas de teste de diferentes ajustes. O fator humano é removido, passando a responsabilidade de replicar voltas para o algoritmo proposto. Abaixo do objetivo principal estão os objetivos específicos de avaliar a diferença de tempo de volta utilizando diferentes configurações de aerodinâmica ou freio, assim como segmentar o *software* para que seja possível alterar o modelo controlador e adaptar para diferentes simuladores.

Para atingir os objetivos propostos é necessário realizar estudos nas áreas de pilotagem, desenvolvimento e teste de softwares. As pesquisas geram a fundamentação teórica necessária para implementação.

## 2 FUNDAMENTAÇÃO TEÓRICA

Ao longo deste capítulo, são apresentados conceitos que embasam o desenvolvimento do trabalho proposto. O conhecimento da participação de simuladores no cenário atual e dos conceitos de *software in the loop* e pilotagem são pilares fundamentais compreensão do projeto. Desse modo, na sequência, são apresentados os tópicos sobre jogos e simuladores, telemetria, pilotagem e *software in the loop*, os quais demonstram os fatores que tornam um jogo em um simulador e apresentam técnicas de pilotagem e de teste. Por fim, estão descritos os trabalhos relacionados, os quais auxiliaram na solução.

### 2.1 Jogos e simuladores

Atualmente, jogos e simuladores de corridas são muito comuns para diferentes plataformas, alguns com propósitos mais lúdicos. Para definir o que é um simulador, Narayanasamy *et al.* (2005) publicaram um estudo que visa categorizar e distinguir jogos de simuladores, elencando suas similaridades e diferenças. O estudo ressalta que devido à popularização e avanço do processamento gráfico de computadores pessoais e com a entrada de indústrias de jogos no mercado de simuladores, cada vez mais jogos se aproximam da realidade. Para os autores, dentre os pontos que caracterizam um jogo com simulador estão a reprodução fiel ao mundo real, utilizando ambientes existentes, e a capacidade de exercitar e desenvolver habilidades.

A capacidade de desenvolver habilidades através de simuladores é apresentada na matéria publicada no New York Times em 2007. Spurgeon (2007) fala sobre a existência de um grande simulador na fábrica da McLaren e elogia o atual hepta campeão de Fórmula 1, Lewis Hamilton, por ter imposto um excelente ritmo de corrida no grande prêmio do Canadá. Lewis, que nunca havia pilotado no Canadá, ao ser questionado sobre a utilização deste simulador diz ter feito algumas simulações, porém foram poucas voltas.

Diversos jogos simuladores de corrida existem atualmente, dentre eles estão iRacing, Asseto Corsa e Fórmula 1. O iRacing é utilizado por pilotos profissionais, como Max Verstappen, Lando Norris e Rubens Barrichello, em seus simuladores pessoais. O título Asseto Corsa Competizione é um produto licenciado para corridas de categoria GT3, reproduzindo com extrema fidelidade os aspectos destes carros para competições online. Por fim, o Fórmula 1 é uma franquia de jogos com lançamento anual, seguindo as temporadas da Fórmula 1. A franquia de jogos publica pacotes de atualização que modificam o desempenho dos carros no jogo para que se aproximem do desempenho real do carro durante a temporada.

As competições *online* estão presentes em todos simuladores citados anteriormente. Além da utilização do simulador para competições eletrônicas, a equipe Red Bull, antes de cada GP, apresenta em suas redes sociais uma volta rápida no circuito, feita por um de seus

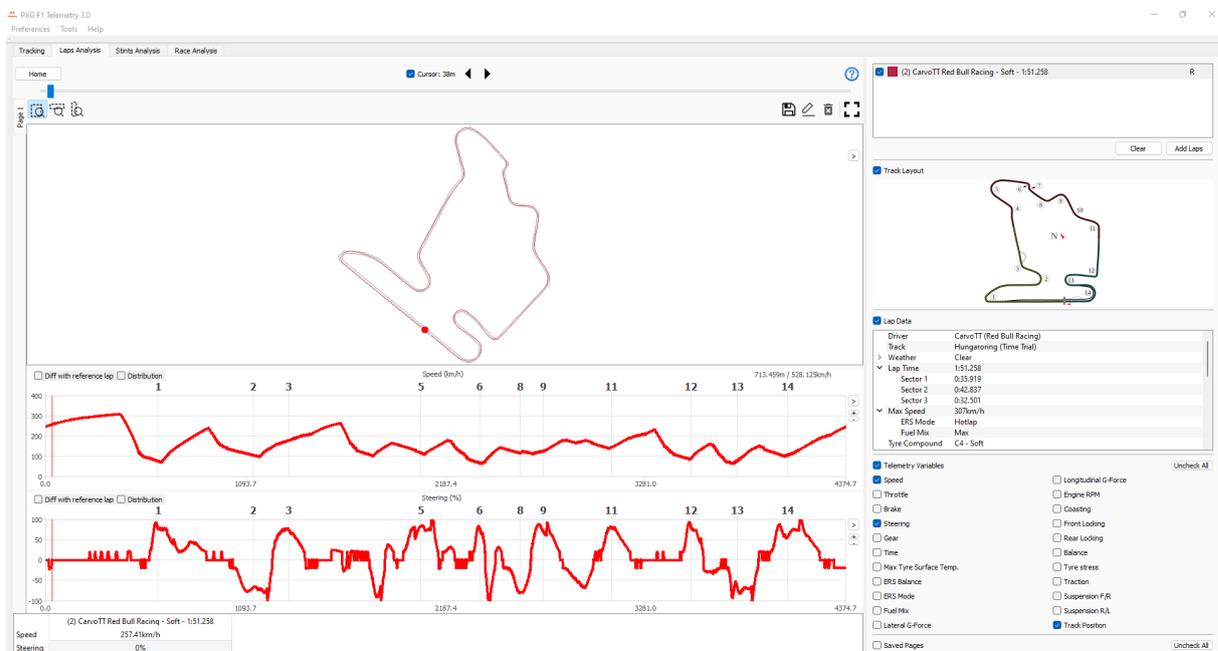
pilotos. As voltas apresentadas possuem tempos semelhantes aos tempos obtidos durante o treino classificatório do final de semana. Cada uma dessas voltas pode ser analisada através da telemetria presente nos simuladores, assim como é feito nas pistas.

## 2.2 Telemetria

A definição de telemetria nos dicionários se resume à aquisição de um conjunto de dados remotamente (MICHAELIS... , 2022). Nos simuladores, é chamado pelo mesmo nome a aquisição de dados dos carros, mesmo que essa aquisição seja feita no mesmo local onde os dados estão sendo gerados. Cada simulador possui seu próprio protocolo para extração da telemetria, podendo ser através de arquivos binários, memória compartilhada ou via API UDP. O conjunto de dados enviados por cada simulador varia, entretanto dados básicos como de posição do carro na pista, velocidade e marcha são comuns a todos.

*Softwares* de análise de telemetria são comuns no mundo do automobilismo. Em simuladores existem plataformas proprietárias de telemetria e plataformas *open source*. O programa PXG F1 Telemetry é um exemplo de ferramenta de código aberto com desenvolvimento colaborativo, com o qual dele é possível realizar a visualização dos principais parâmetros de telemetria. Na Figura 1, é apresentada a interface do *software*.

Figura 1 – PXG F1 Telemetry



Fonte: elaborado pelo autor.

Na proporção direita do software, apresentado na Figura 1, está o menu de configuração, onde é possível visualizar quais voltas estão sendo comparadas, o traçado com as curvas enume-

radar, verificar as configurações utilizadas no carro e selecionar quais variáveis são apresentadas graficamente na aba localizada na esquerda da interface.

O trabalho desenvolvido por Bugeja, Spina e Buhagiar (2017) , consiste em um sistema baseado na telemetria de simuladores que visa auxiliar no treinamento dos pilotos do simulador. A coleta de dados de velocidade e ângulo de cada roda permitiu aos autores criarem um sistema que avalia erros comuns de subesterço e sobresterço e avisa aos pilotos através de um retorno sonoro que um erro foi cometido, auxiliando no desenvolvimento das suas técnicas básicas de pilotagem.

## 2.3 Pilotagem

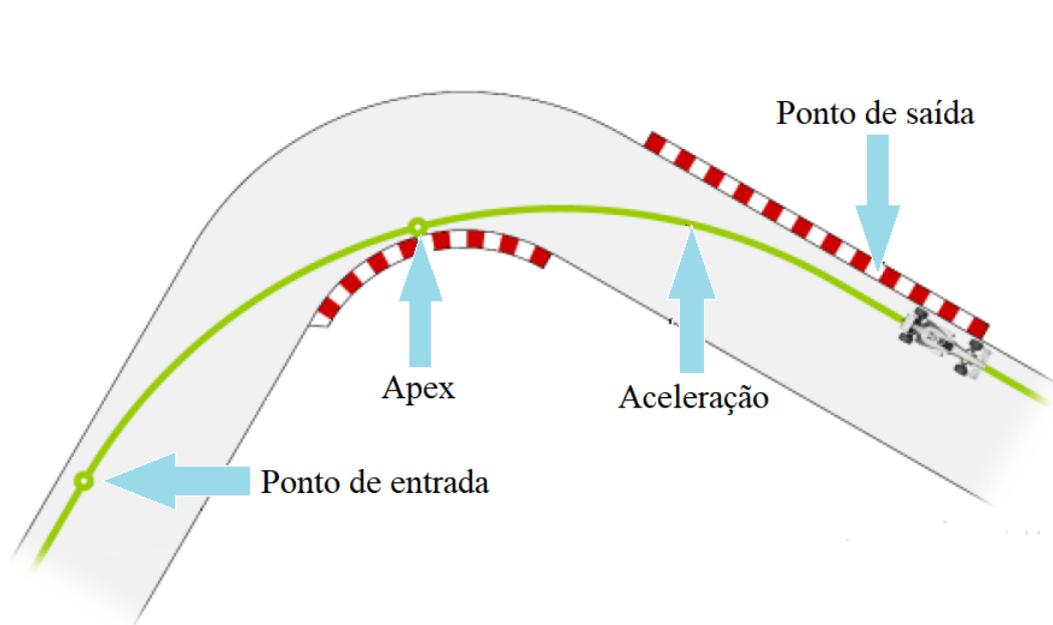
Para Beckman (1991), em sua publicação "*The Physics of Racing Series*", o melhor piloto é aquele capaz de impor a maior velocidade média e terminar no menor tempo. Existem diferentes categorias do automobilismo, como o *Drift*, arrancada, Fórmula e *Gran Turismo*, cada um com suas técnicas de pilotagem. As categorias de circuito visam a maior efetividade da pilotagem em diferentes setores, que apresentam retas, curvas de alta velocidade e curvas de baixa velocidade.

Parte da performance do piloto nas retas é resultado de uma boa saída de curva que antecede a reta. Uma curva percorrida em um traçado não ideal pode comprometer a aceleração do carro no início da reta e conseqüentemente impactar na velocidade final ao término da reta. Devido a isso, o conhecimento de técnicas de aproximação de curvas são fundamentais para um bom resultado.

### 2.3.1 Curvas

Em cada curva de um circuito podem ser elencados quatro pontos principais: o ponto de entrada, o *Apex*, o ponto de aceleração e o ponto de saída, apresentados na Figura 2. Além desses pontos, a curva possui três zonas, a zona de frenagem, a zona intermediária e a zona de saída. A frenagem ocorre antes do ponto de entrada e a retomada de aceleração se dá de forma gradual, entre o ponto de aceleração e o ponto de saída.

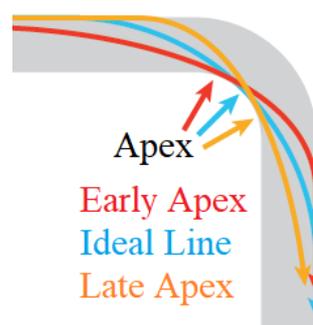
Figura 2 – Principais pontos de uma curva



Fonte: adaptado de Bugeja, Spina e Buhagiar (2017).

O ponto *Apex* geralmente não é localizado no centro geométrico da curva. *Late Apex* é o nome dado à técnica de aproximação de curvas onde o ponto de *Apex* é mais próximo do ponto de saída da curva, já a técnica onde o *Apex* é mais próximo da entrada da curva é chamado de *Early Apex*. Na Figura 3 são apresentadas sobrepostas as diferentes técnicas.

Figura 3 – Tipos de aproximação



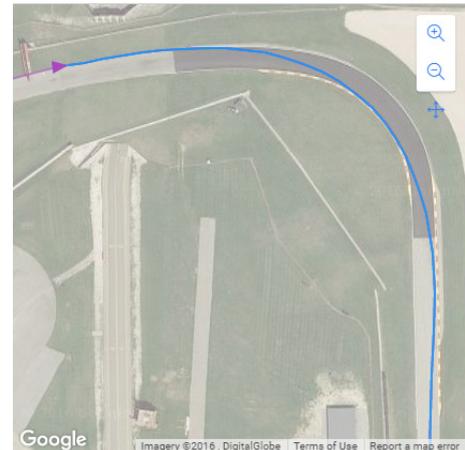
Fonte: adaptado de Niggel (2017).

A escolha da melhor técnica deve levar em consideração a sequência de curvas. A Figura 4 apresenta as imagens a e b utilizadas por Williams (2016) como exemplos reais das utilizações de *Early Apex* e *Late Apex*, respectivamente.

Figura 4 – Circuito Road America



(a) *Early Apex*



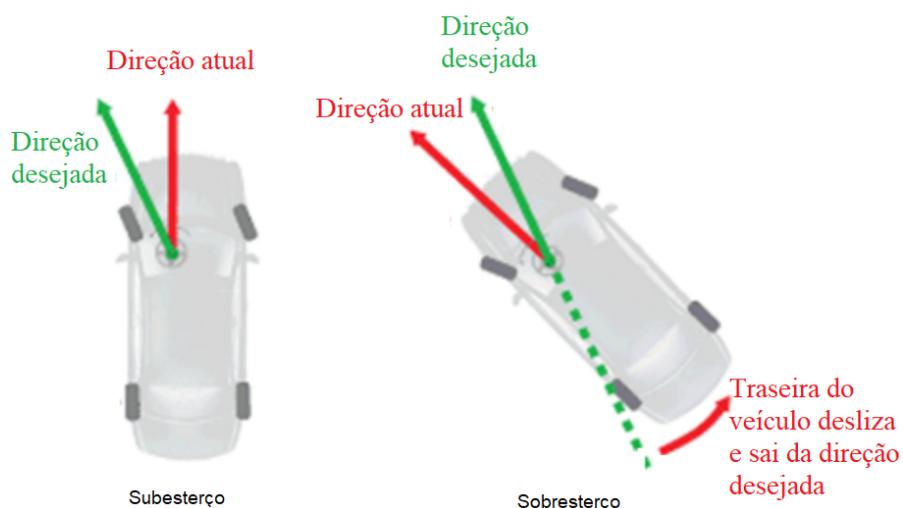
(b) *Late Apex*

Fonte: Williams (2016).

### 2.3.2 Subesterço e sobresterço

Para melhor aproveitamento de uma curva, menor deve ser o deslizamento lateral dos pneus. Na Figura 5 são apresentados os casos de deslizamento. É considerado sobresterço quando o carro perde a aderência lateral dos pneus traseiros, já o subesterço ocorre quando há deslizamento lateral dos pneus frontais (OVERSTEER..., 2015). Bugeja, Spina e Buhagiar (2017) utilizam, para compostos *slick*, valores entre 8° e 10° como o ângulo máximo entre a trajetória ideal e a trajetória real dos pneus dianteiros.

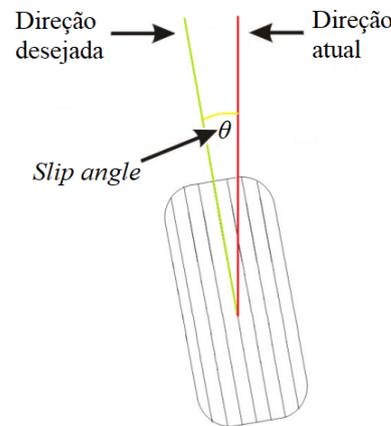
Figura 5 – Subesterço e sobresterço



Fonte: adaptado de Bugeja, Spina e Buhagiar (2017).

Na Figura 6 é apresentado o ângulo de deslizamento.

Figura 6 – Ângulo de deslizamento



Fonte: adaptado de Bugeja, Spina e Buhagiar (2017).

### 2.3.3 Deslizamento na frenagem

Desde 1994, o regulamento da Fórmula 1 não permite auxílios eletrônicos na condução dos veículos (ORICCHIO, 2014). Dentre os sistemas eletrônicos que auxiliam na frenagem, o mais conhecido é o ABS, que impede o travamento das rodas. Com a proibição desta tecnologia, atualmente a modulação da frenagem deve ser feita pelo piloto. Caso o piloto exerça mais força de frenagem que o necessário, ocorre o travamento das rodas, que gera desgaste excessivo do composto e aumenta o espaço e tempo necessário para redução da velocidade (LIMA, 2005). Devido a isso, a análise do deslizamento durante o rolamento dos pneus faz-se necessária em competições com regulamento similar ao da F1. Durante a frenagem Bugeja, Spina e Buhagiar (2017) utilizam como regra a faixa de valores entre 10% e 15% para valores aceitáveis de diferença entre a velocidade dos pneus e a velocidade em relação ao solo. Já para aceleração, os autores descrevem que não deve haver deslizamento, garantindo assim toda a capacidade de tração do veículo.

## 2.4 *Software in the loop*

*Hardware in the loop* e *Software in the loop* fazem parte de um conjunto de técnicas de simulação para testes de produtos e situações. Quando o sistema de interesse é constituído por *hardware*, a técnica de HIL é utilizada, já para sistemas baseados em *software*, a técnica SIL. As técnicas consistem na utilização de um modelo de *hardware* ou *software* que, conectado ao sistema a ser testado, exerce controle sobre este, simulando situações reais de maneira repetitiva (DIGITALE... , 2011 apud RAHMAN; MOHAMAD, 2016). A criação de um modelo de simulação de controle de pilotagem conectado ao simulador configura a utilização de SIL.

A utilização de *Software in the loop* apresenta os benefícios destacados por Meister *et al.* (2016), facilidade de validação de diferentes cenários e a comparação de diferentes estratégias para resolução de tarefas. A reprodução de cenários sobre as mesmas condições garante a repetibilidade em testes de comparação, a qual permite que alterações de parâmetros no software de interesse sejam realizadas e validadas sobre a mesma condição de utilização.

## 2.5 Trabalhos relacionados

Nesta seção são apresentados trabalhos relacionados que forneceram técnicas e guiaram o desenvolvimento do modelo proposto. Inicialmente é apresentado o trabalho de Bugeja, Spina e Buhagiar (2017), o qual utiliza do sistema de telemetria de simuladores, na sequência são apresentados trabalhos relacionados a comparação e utilização de sistema inteligentes para controle automático de carros de corrida.

### 2.5.1 *Telemetry-based optimisation for user training in racing simulators*

A viabilidade de processamento e análise, via software, da telemetria é verificada no trabalho desenvolvido por Bugeja, Spina e Buhagiar (2017). O projeto responde se é possível treinar pilotos inexperientes através de um *feedback* ativo que utiliza a telemetria como base. Alicerçados na afirmativa de que os pilotos inexperientes aprendem com base em dicas de pilotos mais experientes, eles propuseram a automação do processo de *feedback*.

Utilizando o simulador Asseto Corsa, os autores extraem os dados de telemetria fornecido através do protocolos UDP e aplicam lógicas para sinalizar uma possibilidade de melhoria na pilotagem dos voluntários do teste. Durante a coleta de resultados eles percebem melhoras na pilotagem com o sistema ativo, entretanto alguns voluntários, dentre os 27, possuíram dificuldades em aplicar as dicas que eram sugeridas pelo software.

### 2.5.2 *Humans and autonomous driving in a professional racing simulator*

O trabalho categoriza diversas voltas de participantes e realiza a comparação com as voltas obtidas pelo modelo de IA proposto por Remonda, Veas e Luzhnica (2021). Os autores ressaltam que muitas validações de sistemas de IA são através de comparações com seres humanos e que para isto existe a necessidade de um estudo mais aprofundado, visto que a comparação dos resultados entre humanos e inteligência artificial depende de muitos fatores.

A comparação da melhor volta realizada por um humano e a melhor volta do sistema de controle autônomo apresentou a diferença de 180 milésimos de segundo. Através da análise da telemetria os autores verificaram que a modulação de frenagem realizada pelo ser humano é menos eficiente se comparada ao piloto automático, que maximiza o sistema de freios, aproveitando 100% da capacidade de frenagem do veículo. Contribuindo com a demonstração da capacidade de um piloto automático extrair a capacidade máxima de novas configurações.

### 2.5.2.1 Adaptive AI in a racing simulator

Utilizando um simulador *open source*, TORCs, Mackinnon (2012) propôs o próprio sistema de controle autônomo para apresentar uma dificuldade maior aos pilotos que utilizam sua simulação. A inspiração do autor parte da afirmativa de que diversos pilotos consideravam os adversários, controlados automaticamente, muito fáceis ou muito difíceis, sem um nível intermediário.

Assim como no trabalho de Bugeja, Spina e Buhagiar (2017), o artigo apresenta modelos de lógicas para controle de velocidade e cálculo de frenagem. Neste caso aplicados a utilização ao controle automático de um carro dentro do simulador.

Mackinnon (2012) apresenta a técnica de cliente servidor, descartada devido a limitação de hardware. A técnica consiste em utilizar o simulador como servidor, sem alterações do código, e um cliente que realiza o controle do piloto automático. Para o autor, o cliente fica sobrecarregado com o controle e para de realizar as leituras da telemetria do simulador. Esse fato gera a necessidade da verificação da frequência de aquisição obtida, a qual é adicionada à metodologia do projeto.

### 3 METODOLOGIA

Neste capítulo é apresentada a metodologia, dividida em quatro etapas que descrevem o modelo de *software* proposto. A primeira etapa consiste na pesquisa dos dados disponíveis por cada simulador. Na sequência são descritos os critérios que levam a escolha do simulador para aplicação do modelo. A terceira etapa é segmentada em três partes e apresenta os processos executados por cada divisão do software e os requisitos de desenvolvimento. Por último, são elencados os critérios de validação pelos quais o modelo é validado.

#### 3.1 Dados disponíveis via telemetria

Dentre os simuladores pessoais mais conhecidos estão o Asseto Corsa, o F1 2021, o iRacing e o Project Cars 2. Cada simulador possui seu próprio protocolo de disponibilização de telemetria. Entre 100 e 300 variáveis são compartilhadas fornecendo dados de sensores de carros de corrida, simulando com precisão a coleta de dados que é feita nas pistas. No Quadro 1 estão categorizados conjuntos de dados que são comuns aos simuladores, exceto pela categoria *Setup*, que está presente apenas no simulador Fórmula 1.

Quadro 1 – Dados de telemetria

<b>Categoria</b>	<b>Descrição</b>
Painel	Informações sobre velocidade, marcha, rotações do motor.
Pedais	Atuação do acelerador, freio e embreagem após tratamento do jogo.
Volante	Ângulo do volante após tratamento do simulador.
Assistências	Lista de assistências ativas no simulador
Controle	Atuação do acelerador, freio, embreagem e volante no controle.
<i>Motion</i>	Acelerações laterais, <i>roll</i> , <i>pitch</i> e <i>yaw</i> .
Mapa	Posição e velocidade nos eixos X, Y e Z.
Pista	Nome da pista, distancia total e número de setores.
Clima	Condições climáticas atuais e previsões.
Tempos	Volta atual, anterior e melhor volta. Tempos de setores individualmente.
Rodas	Velocidade, deslizamento, pressão e ângulo das rodas dianteiras.
Suspensão	Curso e aceleração da suspensão.
<i>Boost</i>	Dados sobre tecnologias como KERS e DRS.
Integridade	Peças danificadas, nível de combustível e desgaste de pneus.
Temperaturas	Temperaturas do motor, discos de freio e da pista.
<i>Setup</i>	Ajustes atuais do carro.

Fonte: elaborado pelo autor.

Para categorização dos dados presentes no Quadro 1, todos os parâmetros disponíveis por cada simulador são documentados em tabelas com o nome da variável, tipo e descrição

fornecida pelo fabricante. Após a documentação individual, são separadas as variáveis em suas categorias e comparadas entre os simuladores.

### 3.2 Escolha do simulador

Após a verificação dos dados disponibilizados pelos simuladores, para auxiliar na escolha do simulador a ser utilizado para desenvolvimento do trabalho, são elencados cinco critérios principais. Com os critérios definidos foi criada a Tabela 2. As células em cinza indicam que o simulador atende ao requisito. Os critérios utilizados são listados e ordenados por relevância.

1. Protocolo de telemetria com código aberto.
2. Protocolo documentado pela desenvolvedora do simulador.
3. Disponibilização dos dados de posição X e Y, de velocidade e deslizamentos.
4. Existência de um modo de treino.
5. Possibilidade de não utilizar desgaste de pneu e consumo de combustível.
6. Disponibilização dos ajustes atuais do carro via telemetria.

Quadro 2 – Critérios x simuladores

<b>Critério</b>	<b>Asseto Corsa</b>	<b>F1 2021</b>	<b>iRacing</b>	<b>Project Cars 2</b>
1				
2				
3				
4				
5				
6				

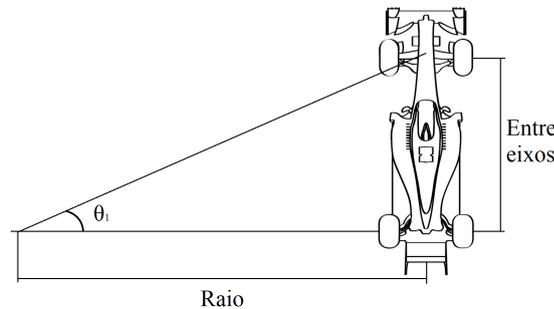
Fonte: elaborado pelo autor.

Analisando o Quadro 2 é possível verificar uma vantagem do Fórmula 1 sobre os demais simuladores, visto que disponibiliza por telemetria os valores de cada ajuste possível do carro. Já o simulador que menos atende aos critérios é o iRacing. Os dados de telemetria disponibilizados pelo simulador iRacing podem ser importados pelo sistema proprietário de análise de telemetria da empresa MoTec.

Considerando os critérios definidos, opta-se pela escolha do simulador Fórmula 1 2021 e utilização apenas o autódromo de Interlagos. O circuito é escolhido por possuir curvas em sequência e curvas de alta velocidade e baixa velocidade. Já o simulador foi escolhido para desenvolvimento, entretanto, um dos objetivos específicos é a possibilidade de adaptação do *software* para diferentes simuladores, que atendam aos critérios 1, 2 e 3.

Devido ao fato de que o simulador F1 2021 não disponibiliza a informação do ângulo de deslizamento, é utilizada a aproximação simples proposta por Santos (2015), Figura 7.

Figura 7 – Simplificação para ângulo neutro



Fonte: elaborado pelo Autor.

Na simplificação é considerado que o raio de curvatura depende apenas da distância entre eixos do carro ( $wb$ ) e do ângulo de esterço da roda ( $\theta_1$ ), em radianos, gerando a equação (1).

$$r = \frac{wb}{\theta_1}. \quad (1)$$

A aceleração centrípeta de um objeto em movimento circular uniforme é apresentada na equação (2), na qual é aplicada a equivalência do raio proposta, gerando a equação (3) que possibilita isolar o objeto de interesse e obter a equação (4) que refere-se a direção atual do carro.

$$a_{cp} = \frac{v^2}{r}, \quad (2)$$

$$a_{cp} = \frac{v^2}{\frac{wb}{\theta_1}}, \quad (3)$$

$$\theta_1 = \frac{wb * a_{cp}}{v^2}. \quad (4)$$

Onde  $a_{cp}$  equivale a aceleração lateral disponível na telemetria e  $v$  à velocidade. O  $\theta_1$  equacionado representa o ângulo da direção atual do carro em radianos.

Para obter o ângulo de deslizamento é realizada a subtração de  $\theta_2$ , ângulo atual das rodas dianteiras, do ângulo de direção atual do carro,  $\theta_1$ , representado na equação (5).

$$SlipAngle(rad) = |\theta_1| - |\theta_2|. \quad (5)$$

A conversão de radianos para graus, necessária para preenchimento da estrutura padrão de aquisição, é determinada pela equação (6).

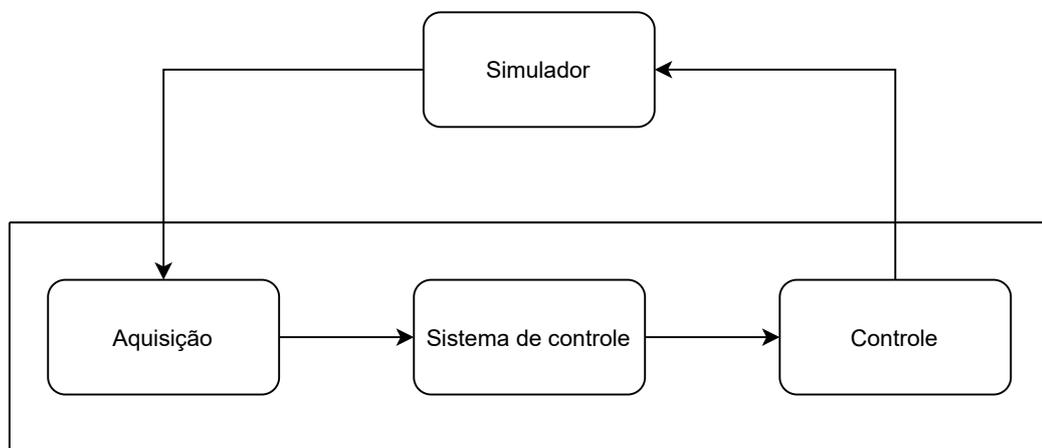
$$SlipAngle(^{\circ}) = \frac{SlipAngle(rad) * 180}{\pi}. \quad (6)$$

A partir dessas definições é possível especificar a arquitetura do software.

### 3.3 Software

O *software* consiste em uma aplicação dividida em três processos menores e independentes, construídos sobre padrões de orientação a objeto, exibidos na Figura 8.

Figura 8 – Divisão do software em 3 etapas



Fonte: elaborado pelo Autor.

A primeira etapa, de aquisição de dados, é responsável pela comunicação de leitura da telemetria do simulador, realizando a tradução do protocolo de comunicação específico para um conjunto de dados que será utilizado pelo sistema de controle. O sistema de controle, módulo principal, compara o conjunto de dados que estão sendo recebidos com um modelo pré definido de traçado da pista e define qual ação deve ser tomada para cada uma das possíveis variáveis de saída. Por último, na etapa que simula um controle virtual, são convertidas as variáveis do sistema de controle para botões reconhecidos pelo simulador como comandos.

Cada bloco da Figura 8 representa uma etapa do software. O equacionamento, fluxo de desenvolvimento e requisitos de cada etapa são apresentados nas subseções 3.3.1, 3.3.2 e 3.3.3.

#### 3.3.1 Aquisição dos dados

Devido à escolha do Fórmula 1, a aquisição de dados é feita via *User Datagram Protocol* (UDP), respeitando o modelo disponível no site da desenvolvedora do jogo. A cada pacote UDP, um tipo de pacote de dados é enviado. Dentre esses tipos estão os pacotes *Motion*, *Car Status* e

*Car Telemetry* enviados com a frequência definida no menu de configurações do jogo, variando entre 10 Hz e 60 Hz (F1..., 2020).

Do pacote *motion*, são utilizadas as variáveis "m\_worldPosition" e "m\_worldVelocity", em suas variações X, Y e Z, além das variáveis "m\_gForceLateral", "m\_frontWheelsAngle", "m\_wheelSlip" e "m\_yaw". Os dados disponíveis no conjunto *motion* definem as posições globais do carro na pista, a velocidades nos eixos, a força de aceleração lateral, o ângulo das rodas dianteiras, o deslizamento durante rolamento e o ângulo que o carro está apontado. As propriedades "m\_maxRPM" e "m\_drsAllowed", que possuem o valor máximo de rotação do motor e a indicação de autorização do uso do *Drag Reduction System* (DRS), serão extraídas do pacote *Car Status*. Por fim, as informações de velocidade, marcha e rotações do motor, referem-se aos atributos "m\_speed", "m\_gear" e "m\_engineRPM" que estão contidos no pacote *Car Telemetry*.

O processo realiza a leitura da telemetria em UDP, aplica o cálculo de deslizamento e converte a estrutura proprietária do simulador para a definida no Quadro 3.

Quadro 3 – Interface: Aquisição - Sistema de controle

Variável	Tipo	Observação
id	long	Identificador único do pacote, incrementado a cada atualização.
speed	float	Em km/h.
gForceLateral	float	
gear	int	Valores de -1 a 8.
engineRPM	int	
maxEngineRPM	int	
slipRatio	float[4]	Vetor com percentual de deslizamento de cada pneu.
slipAngle	float	Ângulo de deslizamento do carro em graus.
steerAngle	float	Ângulo das rodas frontais em graus.
positionX	float	
positionY	float	
positionZ	float	
lapNo	int	
speedX	float	Em m/s.
speedY	float	Em m/s.
speedZ	float	Em m/s.
yaw	float	Em rad.
drs	float	Apenas Fórmula. Valores 0 e 1, onde 1 pode ativar o DRS.

Fonte: elaborado pelo autor.

Com a estrutura apresentada no Quadro 3 é criado um padrão de comunicação de telemetria entre a etapa de aquisição e a etapa de controle. A definição do modelo visa atender ao requisito de modularidade e possibilidade de expansão para diferentes simuladores, através da substituição do processo de aquisição por um processo que realize a leitura de dados de outro simulador e converta para a mesma estrutura.

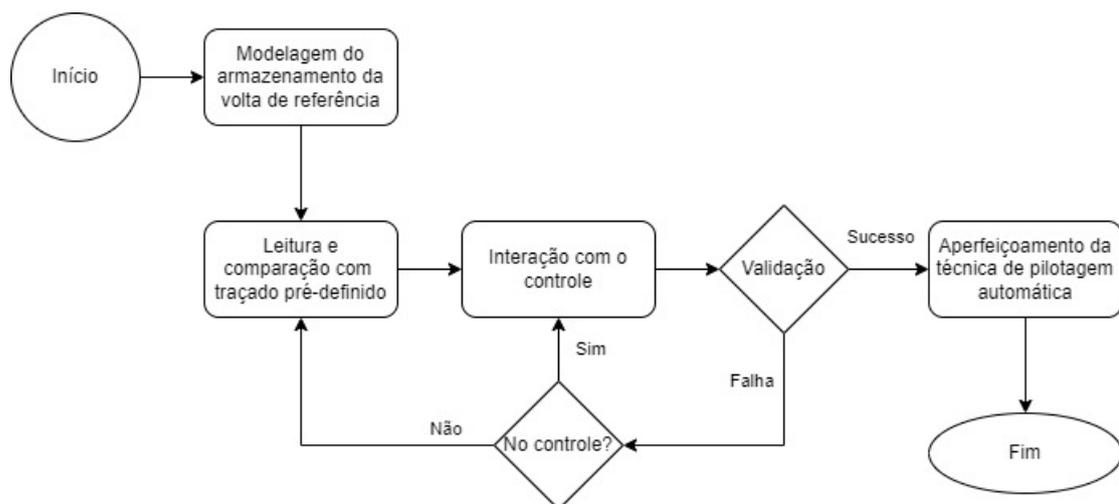
### 3.3.2 Controle

Este objeto é responsável por virtualizar um controlador de jogo, atendendo aos requisitos de simular 3 eixos analógicos e 4 botões digitais que sejam reconhecidos nas plataformas Windows 10 e 11. O primeiro eixo está vinculado à direção do carro e os demais eixos aos pedais de acelerador e freio. Já os botões são atribuídos às ações de abrir o menu, ativar DRS, aumentar e diminuir a marcha.

### 3.3.3 Sistema de controle

O desenvolvimento do sistema de controle é particionado em quatro etapas, sendo elas: modelagem do armazenamento da volta de referência, leitura e comparação com traçado pré-definido, interação com o controle e por último a aplicação da lógica de controle. Na Figura 9 é apresentado o fluxograma com as etapas de desenvolvimento do processo.

Figura 9 – Etapas do desenvolvimento do sistema de controle



Fonte: elaborado pelo Autor.

Na etapa de modelagem do armazenamento da volta de referência, é desenvolvida a funcionalidade de escrita e leitura da volta padrão. A estrutura salva no arquivo se assemelha aos atributos do Quadro 3, com um separador "|" entre eles, de maneira a ser possível realizar a leitura e edição de parâmetros diretamente por um editor de texto. A classe responsável pela leitura do arquivo durante a repetição automática do traçado realiza a leitura do arquivo com nome "REF\_LAP.txt" localizado na pasta da aplicação e armazena os dados em memória durante toda a execução da simulação de volta.

A etapa de leitura e comparação com traçado pré-definido prevê a estruturação das funções para comparação e reconhecimento da necessidade de atuação do acelerador, do freio e do esterçamento do volante. Através das funções, o sistema desempenha o papel semelhante ao

de um robô seguidor de linha, mantendo o carro a 20 km/h e comparando as coordenadas atuais com as coordenadas da telemetria anterior salva.

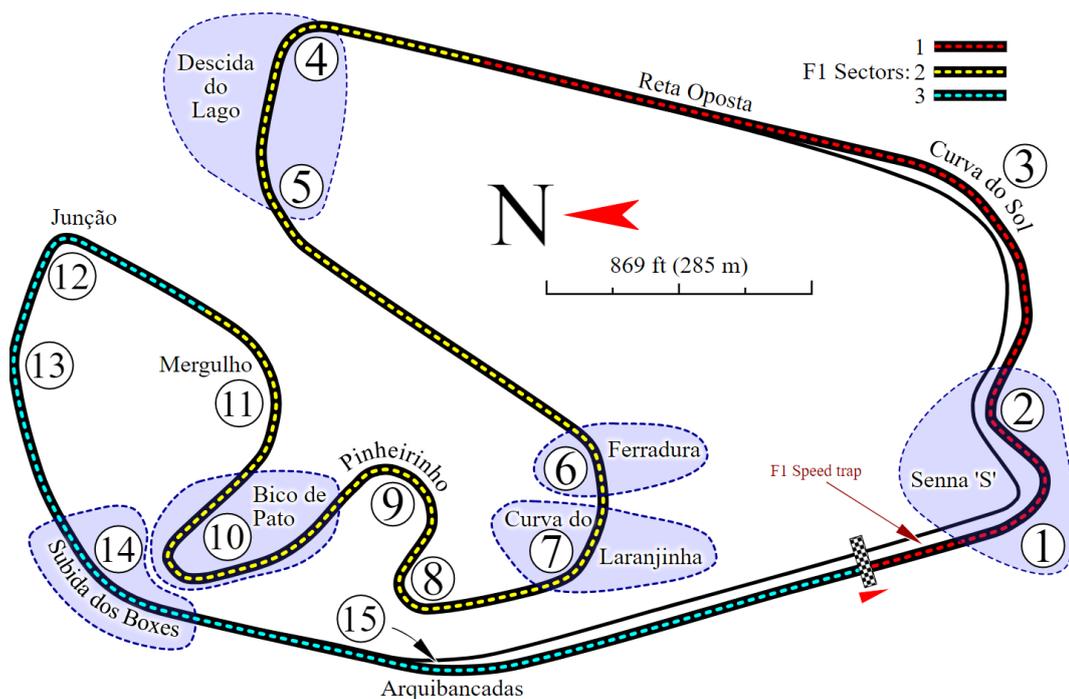
A interação com controle se dá através do resultado das funções desenvolvidas na segunda etapa. Estes resultados são convertidos para variáveis e enviados para o processo de controle, atuando diretamente no controlador virtual, gerando um sinal para o simulador. A conclusão desta etapa intermediária do desenvolvimento permite a validação do sistema como um todo, interagindo desde a aquisição do dado até a resposta via controle.

Após validação intermediária, a lógica de controle, também baseada na comparação com telemetria anterior, utiliza de dados pré-definidos para ponto de frenagem e velocidade máxima das curvas e será aplicada sobre as funções criadas ao longo da segunda etapa de desenvolvimento do sistema de controle, removendo o limitador de velocidade e limitando-se, durante as curvas, à velocidade da volta de referência. Nesta etapa são aplicados os conceitos de deslizamento, garantindo que caso apresente sinais de instabilidade, a aceleração seja automaticamente corrigida.

### 3.4 Testes

Na Figura 10 é apresentado o mapa do autódromo de Interlagos. O circuito é utilizado para aplicação das validações, com foco nas curvas 1 e 2, que apresentam maior diferença entre velocidade de entrada e de contorno.

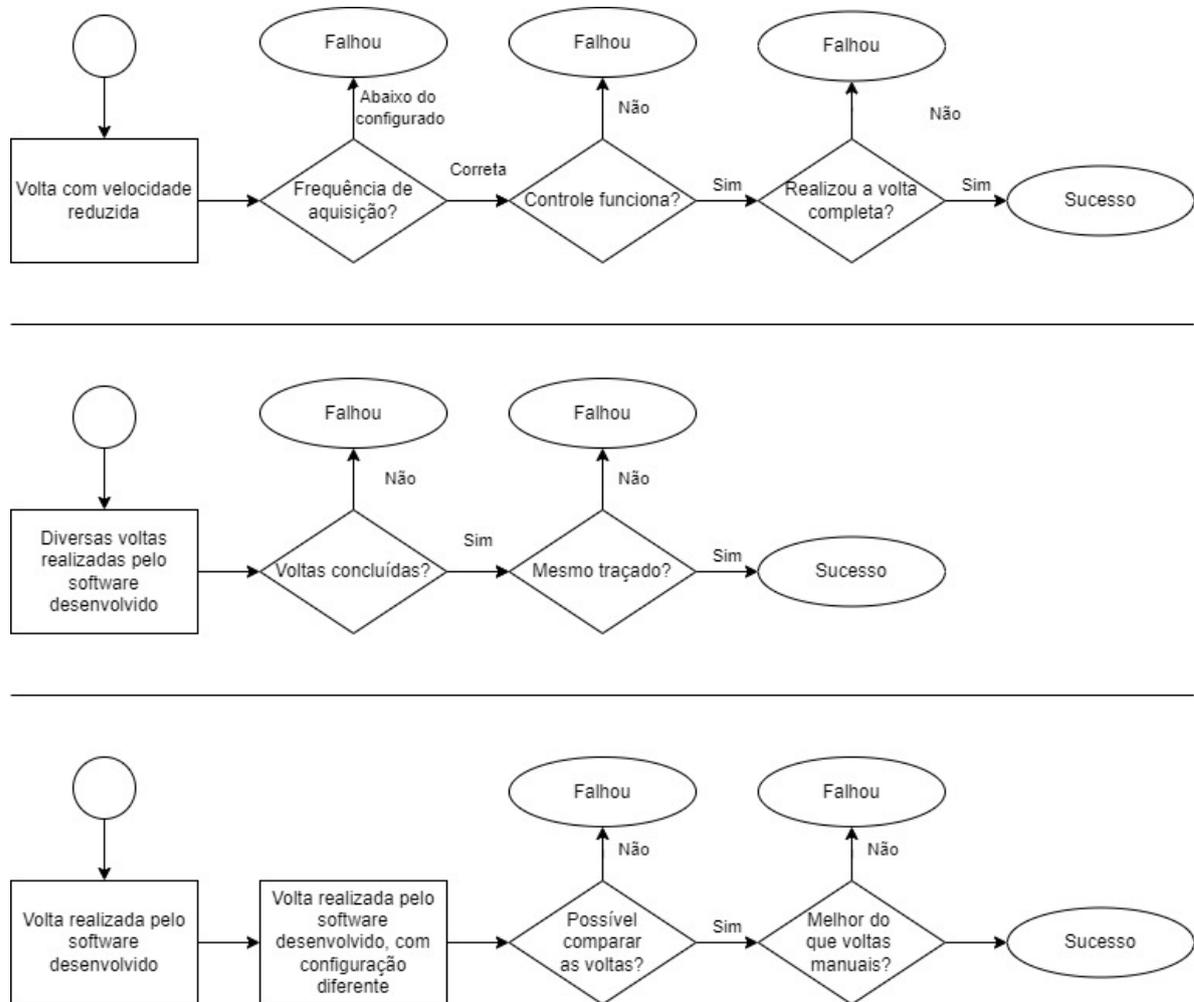
Figura 10 – Mapa de Interlagos



Fonte: Pittenger (2009).

Os três fluxos de validação que visam a verificação do funcionamento do software e de sua utilização para resolução dos objetivos propostos do trabalho são apresentados na Figura 11.

Figura 11 – Fluxos de teste



Fonte: elaborado pelo Autor.

O primeiro fluxo de validação, diferente dos demais, é realizado ao longo do desenvolvimento. Nas subseções 3.4.1, 3.4.2 e 3.4.3 são descritas as validações propostas.

### 3.4.1 Validação do sistema de controle

Este teste tem o objetivo de validar os processos do sistema, de maneira simplificada. Durante a execução de uma volta com velocidade reduzida, são analisados os dados de aquisição, a frequência de atualização e disponibilidade para o sistema de controle. Além disto, é validada a compatibilidade do simulador com o controlador virtual criado.

### 3.4.2 Validação da repetibilidade

Após a conclusão do desenvolvimento, através da utilização do software de telemetria de código aberto, serão comparadas as voltas sobrepostas. Verificando se o traçado e demais parâmetros apresentam repetibilidade entre as execuções utilizando as mesmas configurações de carro.

### 3.4.3 Validação dos objetivos do projeto

Utilizando diferentes configurações de aerodinâmica, é verificado na telemetria o impacto nos resultados. Os resultados utilizando software são comparados com os resultados feitos por uma pessoa pilotando no simulador.

Os dados de comparação são extraídos através do aplicativo de telemetria *open source*, seguindo o critério de que o voluntário realiza três voltas para adaptação aos controles, três voltas para adaptação ao circuito, três voltas com *setup* padrão e três voltas com *setup* de menor pressão aerodinâmica. Apenas as últimas seis voltas são utilizadas para validação, sendo descartas as voltas iniciais de adaptação. Definido o plano de testes é possível iniciar a implementação do *software*.

## 4 IMPLEMENTAÇÃO

Este capítulo aborda as etapas de desenvolvimento do *software*, dividida em seis seções: controlador, interfaces da aplicação, objeto de aquisição, sistema de armazenamento da volta, seguidor de linha e piloto automático. O desenvolvimento tem como objetivo validar o modelo de repetição de voltas para comparação de diferentes configurações de carro sem interferência humana.

### 4.1 Controlador

O modelo de controlador escolhido para desenvolvimento foi a interface de um controle de Xbox, video game desenvolvido pela Microsoft. Os controles de Xbox possuem suporte nativo no sistema operacional Windows e são encontradas bibliotecas python para criação de controladores virtuais. Para desenvolvimento da solução, optou-se pela utilização da biblioteca Vgamepad desenvolvida por Yann Bouteiller em 2021 e disponibilizada de maneira *open source*. A biblioteca disponibiliza funções para definir o estado de pressionamento de cada botão e realizar a conexão e desconexão do controle, porém não possui funcionalidades de pressionar um botão por determinado intervalo de tempo sem afetar o fluxo do programa principal.

Com a biblioteca configurada, é criada uma classe do controle virtual, desenvolvida sobre o modelo *Singleton*, garantindo que independentemente do local onde o objeto seja instanciado, o endereço de memória acessado seja comum. Utilizando esse *design pattern* garante-se que não sejam instanciados diversos controles e que não seja necessária a transferência de objetos do controle entre *threads*.

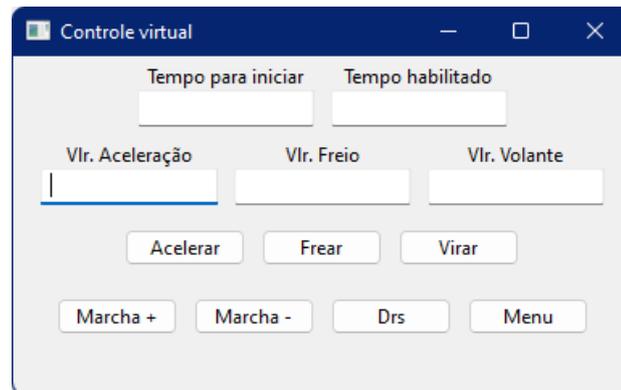
Quando um botão é pressionado, é necessário um determinado intervalo de tempo para que seja reconhecido pelo simulador antes de retornar para o estado de não pressionado. Cenários em que o carro está aumentando ou diminuindo a marcha durante uma curva, a inclusão de um *delay* dentro da função de pressionamento dos botões gera um atraso indesejável, visto que durante esse tempo pararia de analisar outros dados e até mesmo realizar o controle do próprio volante. Para solucionar essa adversidade, os pressionamentos de botões são implementados em *threads* disparadas pela própria classe do controle. A finalização do desenvolvimento do objeto de controle virtual apresenta a necessidade de uma interface para calibração, onde seja possível interagir com os botões do controle virtual para fazer as atribuições no simulador.

### 4.2 Interfaces da aplicação

O desenvolvimento da tela de calibração do controlador utiliza a biblioteca wxPython, que é disponibilizada de maneira *open source* e possui mais de 100 contribuidores no GitHub. Na

interface, existem dois campos para definir o tempo, em segundos, para que o botão seja ativado e o tempo que ele ficará ativo. Além destes dois campos, existem três campos digitáveis de valores, para definir o percentual de aceleração, percentual de freio e o valor de giro do volante, que varia entre 90 e -90. A interface de calibração é apresentada na Figura 12.

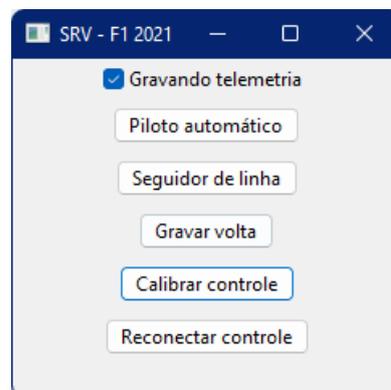
Figura 12 – Interface de calibração



Fonte: elaborado pelo Autor.

Para acessar a tela de calibração, há uma interface principal do programa, Figura 13, aberta quando o programa é executado. Na interface inicial são apresentados cinco botões: piloto automático, seguidor de linha, gravar volta, calibrar controle e reconectar controle. Além dos botões existe a *checkbox* que inicia marcada para determinar qual porta UDP é utilizada para telemetria da volta de reprodução.

Figura 13 – Interface principal



Fonte: elaborado pelo Autor.

Os botões de piloto automático e de seguidor de linha iniciam os processos de controle automático, sendo o segundo limitado a velocidade máxima de 70 km/h. A funcionalidade de gravação de volta também é disponibilizada através da interface inicial, assim como a

possibilidade de reconexão do controle, caso seja necessário reconectá-lo para identificação do simulador. Ao estar utilizando a captura de telemetria realizada por o *software* terceiro, a porta 20777 é ocupada e a telemetria é retransmitida através da porta 20778. Devido a esse fato, a opção de "gravando telemetria" é disponibilizada para que seja corretamente configurada a porta de leitura para as funcionalidades do objeto de aquisição.

### 4.3 Objeto de aquisição

Utilizando recursos *open source*, a classe de aquisição de telemetria é construída sobre a biblioteca Telemetry-F1-2021, desenvolvida por Chris Hannam em setembro de 2021. Através do código de Hannam, todos os dados extraídos via UDP são transformados em objetos com nomes semelhantes aos pacotes da desenvolvedora do simulador. Assim como o controle, inicialmente as operações são executadas sequencialmente. Por este motivo, é necessário que o processo execute dentro de uma *thread* responsável pela atualização dos dados de telemetria.

Utilizando as funções da biblioteca de captura da telemetria, é desenvolvida uma classe que realiza a transformação dos objetos recebidos por UDP na estrutura disponível no Quadro 3. Por existirem três diferentes pacotes que compõem a interface de aquisição, é desenvolvida a função "fromPacket", que recebe por parâmetro o pacote recebido da telemetria, verifica o tipo e atualiza os dados disponíveis no objeto. Em conjunto com a *thread* criada para leitura da telemetria, para cada pacote adquirido da telemetria a função é chamada e o objeto instanciado sobre o padrão Singleton é atualizado e disponibilizado para os demais processos.

### 4.4 Sistema de armazenamento da volta

Com o acesso aos dados estabelecido, foi construída a classe "LapData", responsável pelo armazenamento da volta de referência. O principal atributo desta classe é a lista de objetos de aquisição que combinado com o padrão Singleton disponibiliza globalmente informações de voltas completas em uma lista.

A primeira função que popula a lista de objetos é a função "newFromPacket", processo que recebe por parâmetro o objeto do pacote recebido por UDP e cria uma nova instância na lista sempre que for recebido um pacote do tipo *motion*, copiando os parâmetros da instancia anterior. Os demais tipos de pacote geram apenas atualizações do último objeto adicionado na lista. Ao final da função, caso identifique que uma nova volta foi iniciada, armazena os dados "speed", "gForceLateral", "gear", "engineRPM", "maxEngineRPM", "slipRatio", "slipAngle", "steerAngle", "positionX", "positionY", "positionZ" e "lapNo" da estrutura principal em um arquivo, com o nome gerado através do padrão "MM\_DD\_YYYY\_Lap\_NoLAP.txt", que concatena a data e o número da volta (NoLAP). Esta função é utilizada no fluxo de gravação da volta de referência, e é acionada através do botão "Gravar volta", Figura 13, na interface principal da aplicação.

A segunda função que popula a lista de objetos de aquisição é a função "LoadLap", que preenche a lista com os dados contidos no arquivo de referência "REF\_LAP.txt". A leitura da volta é utilizada internamente nas funções de reprodução de volta, que populam o objeto para realizar comparações.

## 4.5 Seguidor de linha

A implementação do seguidor de linha visa a validação do controlador virtual e do sistema de aquisição de dados. Os pilares fundamentais do desenvolvimento desta etapa do software são a leitura dos dados da volta de referência, a *thread* de telemetria, o controle de velocidade e o controle de angulação do volante. A leitura da volta de referência e o início do sistema de captura de telemetria são feitos ao iniciar o processo, dentro de um *loop* é realizado o controle do acelerador e do volante, ocorrendo em paralelo com a leitura da telemetria.

O controle do acelerador nesta etapa se limita a manter a velocidade do carro em determinado valor. Quando a velocidade está abaixo deste valor, o sistema acelera com 60% da intensidade, ao ultrapassar a velocidade definida, o acelerador passa a 0%. No planejamento do software, a velocidade definida para o seguidor de linha é de 20 km/h, porém estabeleceu-se o valor de 70 km/h, visto que esta é a menor velocidade que o carro atingia durante uma volta.

O ângulo do volante é definido analisando a posição atual do carro e localizando essa coordenada na volta gravada. Com a posição localizada na volta de referência, o sistema localiza para onde o carro deve ir, nos eixos X e Z, cinco coletas de telemetria a frente da posição atual. Com os valores de posição futuros e atuais, o sistema calcula o ângulo do vetor que deve ser realizado e compara com o *yaw*, fazendo a subtração ou soma, a depender do quadrante atual, e aplica sobre o controlador.

Ao longo do desenvolvimento do seguidor de linha verifica-se que quando o carro está em deslocamento com *yaw* de 90° somente valor do eixo X é incrementado, assim como quando o *yaw* é de 0° somente o eixo Z é incrementado. Com isso definiram-se os eixos e quadrantes e se conclui que o eixo Y refere-se a altura do carro em relação ao nível do mar.

## 4.6 Piloto automático

Após a validação do sistema através do seguidor de linha, é implementado o piloto automático. Utilizando como base uma cópia da estrutura do processo anterior, o piloto automático é desenvolvido atualizando o sistema de controle de velocidade para manter a velocidade com base na velocidade da volta de referência na posição atual. Ainda no módulo de controle de velocidade, são adicionadas novas lógicas, para que em velocidades de referência acima de 230km/h o sistema não limite a velocidade atual do carro à velocidade de referência, permitindo que em retas de alta velocidade, caso o piloto automático consiga, ultrapasse a velocidade de referência.

No sistema de piloto automático, são alterados os percentuais de aceleração. Quando a velocidade atual está abaixo de 160 km/h o acelerador é pressionado com 60% da capacidade. Ao ultrapassar os 160 km/h o percentual de aceleração sobe para 70% até os 180 km/h, onde o percentual de aceleração passa a ser de 80%. Para velocidades acima de 200 km/h o acelerador é pressionado com máxima intensidade. Esse sistema previne o deslizamento dos pneus traseiros em baixa velocidade. Caso o deslizamento ultrapasse o fator de 0.2 a aceleração é cortada para 20% até que o fator fique abaixo de 0.2. Além dos controles de aceleração, o controle de frenagem é adicionado ao módulo de controle de velocidade. Caso a velocidade do carro esteja 10 km/h acima da velocidade de referência, o carro aplica uma força de frenagem de 75% até que a velocidade definida seja atingida. O range de 10 km/h entre referência de aceleração e frenagem permite que o carro realize o rolamento livre durante as curvas sem estar sempre realizando a aceleração ou frenagem.

No controle de volante, foram adicionados multiplicadores de intensidade do ângulo do volante. A necessidade de alteração da intensidade de curvatura do volante se verifica através de simulações, que, em altas velocidades é necessário aplicar maior percentual de acionamento do eixo do volante para obter o mesmo ângulo de esterçamento. Com isso se adiciona um multiplicador de 1.5 para todas velocidades e de 2.55 para velocidades entre 130 km/h e 240 km/h. Acima de 240 km/h não é necessária uma correção maior devido ao fato do autódromo de Interlagos não possuir curvas acentuadas de alta velocidade, como seria o exemplo de Silverstone na curva Copse, que, segundo dados de telemetria das voltas de classificação do GP de 2021, os pilotos contornam a curva a 300km/h.

Ainda no sistema de controle do volante, o número de telemetrias a frente verificado para calcular o ângulo do vetor a ser realizado teve que ser adaptado. Para o seguidor de linha é utilizado um valor fixo de 5 interações de telemetria, para o sistema do piloto automático passou a variar em um range entre 5 interações e 30 interações. Quando a velocidade do carro está abaixo de 70km/h, o controlador utiliza a mesma regra do seguidor de linha. Porém entre 70 km/h e 100km/h o sistema utiliza 6 iterações, entre 100 km/h e 150 km/h são utilizadas 8 iterações, de 150 km/h a 180 km/h são 10 iterações, 12 iterações entre 180 km/h e 200 km/h, 15 iterações entre 200 km/h e 250 km/h e 30 iterações para velocidades superiores. Isso garante que quanto mais rápido o carro estiver, mais para frente será o ponto de análise, antecipando e suavizando os movimentos de volante assim como de frenagem.

Ao *loop* são adicionados dois novos procedimentos observadores, de marcha e de DRS, essas funções realizam a verificação se a marcha pode ser incrementada ou diminuída, assim como se o DRS pode ser acionado. O simulador, através da telemetria envia um sinal "1" quando o DRS pode ser acionado, caso esse valor seja verificado na telemetria atual, é disparado o acionamento do DRS do controlador virtual. Para controle de marchas foram utilizados os valores do Quadro 4, apresentado a seguir.

Quadro 4 – Marcha X Velocidade

<b>Marcha</b>	<b>Velocidade limite (km/h)</b>
1	30
2	70
3	120
4	170
5	215
6	260
7	290
8	-

Fonte: elaborado pelo autor.

Caso a velocidade ultrapasse a velocidade limite da marcha atual, o controle virtual é acionado para incremento da marcha, caso a velocidade atual seja inferior ao limite de uma marcha inferior, é acionada a redução de marcha. A conclusão do desenvolvimento destas funcionalidades significa a conclusão da implementação do sistema, permitindo a análise dos resultados.

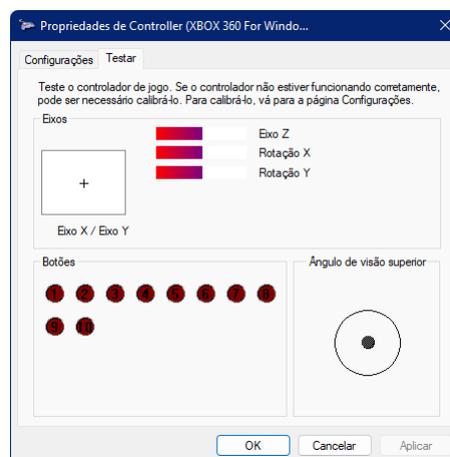
## 5 ANÁLISE DOS RESULTADOS

Nesse capítulo são apresentadas as aplicações dos testes predefinidos na metodologia. A cada etapa de validação são elencados e analisados os resultados obtidos e demonstrados os pontos de melhoria existentes.

### 5.1 Validação do controle virtual

A validação do controle virtual, primeiro cenário de teste, foi realizado ao longo do desenvolvimento. O primeiro teste realizado nesta etapa foi a validação do reconhecimento do controle virtual pelo sistema operacional. Na Figura 14 é apresentado o controlador reconhecido pelo Windows 11. A biblioteca utilizada emula um controle de XBOX 360, possuindo 5 eixos analógicos, 10 botões digitais e um eixo digital, tal capacidade é superior aos requisitos para o controle virtual.

Figura 14 – Controle reconhecido no Windows



Fonte: elaborado pelo Autor.

Após o reconhecimento do controlador pelo sistema operacional, foi validado o reconhecimento dos acionamentos do controle através da interface de calibração desenvolvida. A partir do funcionamento do controlador virtual, foi realizado o teste de frequência de leitura da telemetria através da funcionalidade de gravação da volta.

As configurações utilizadas para verificação da frequência de aquisição da telemetria ao longo do teste foram de 20 Hz e 60 Hz. A primeira volta de validação, utilizando configuração de 20 Hz, possui o tempo total de 86,387 segundos, na qual o valor esperado de pacotes recebidos é de 1.727. Verifica-se que o sistema de gravação capturou 1.606 pacotes de movimento durante

o traçado, cerca de 93% dos pacotes foram recebidos com sucesso, correspondendo a uma frequência de amostragem de 18,5 Hz. A segunda volta de validação visa validar a frequência de aquisição de 60 Hz, neste teste o tempo total da volta é de 70,949 segundos, com valor de pacotes esperado de 4.256, entretanto o número de pacotes salvos foi de 3.500, correspondendo a 82% de sucesso ou 49 Hz. Os resultados deste teste são elencados no Quadro 5.

Quadro 5 – Teste de frequência de aquisição

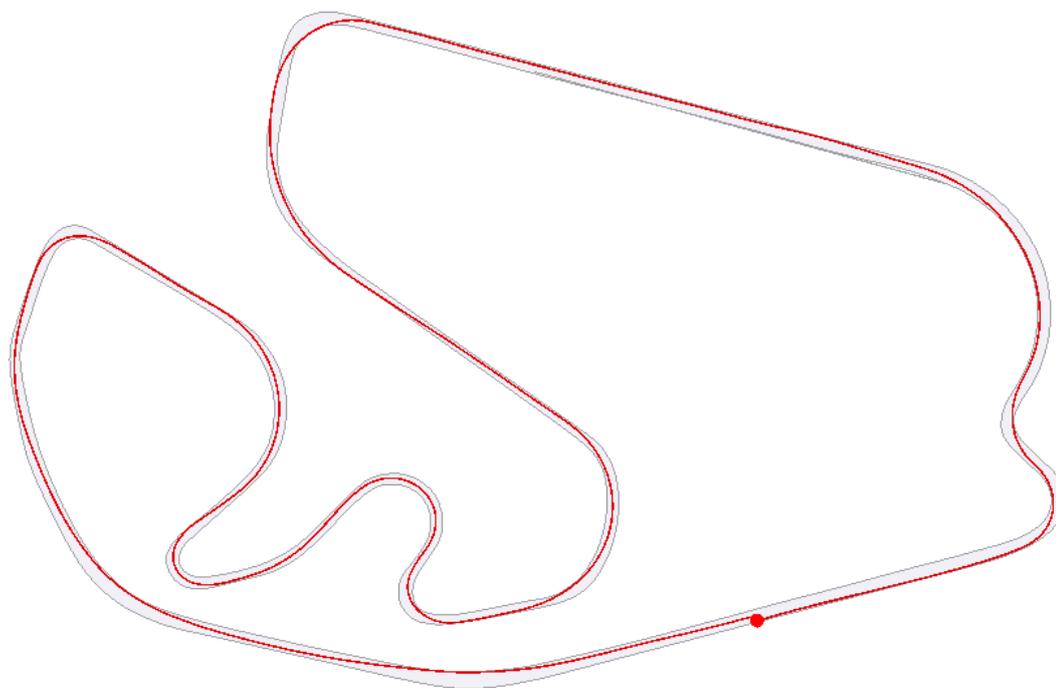
Configuração	Tempo de volta	Pacotes esperados	Pacotes salvos	Frequência real
20 Hz	86,387 s	1.727	1.606	18,5 Hz
60 Hz	70,949 s	4.256	3.500	49 Hz

Fonte: elaborado pelo autor.

Apesar do valor ser abaixo do configurado, considera-se que a taxa de amostragem apresenta precisão suficiente para controle, sendo aproximadamente 1 pacote a cada 20 milésimos de segundo. Com o teste do controle e da frequência realizados, a etapa final da validação corresponde à aplicação sistema de controle com velocidade reduzida.

Para reprodução da volta em baixa velocidade, é utilizada a volta de referência gravada no autódromo de Interlagos sobre o traçado apresentado na Figura 15. O traçado de referência foi realizado com tempo de 1:10.949 utilizando o carro da equipe Red Bull com *setup* padrão.

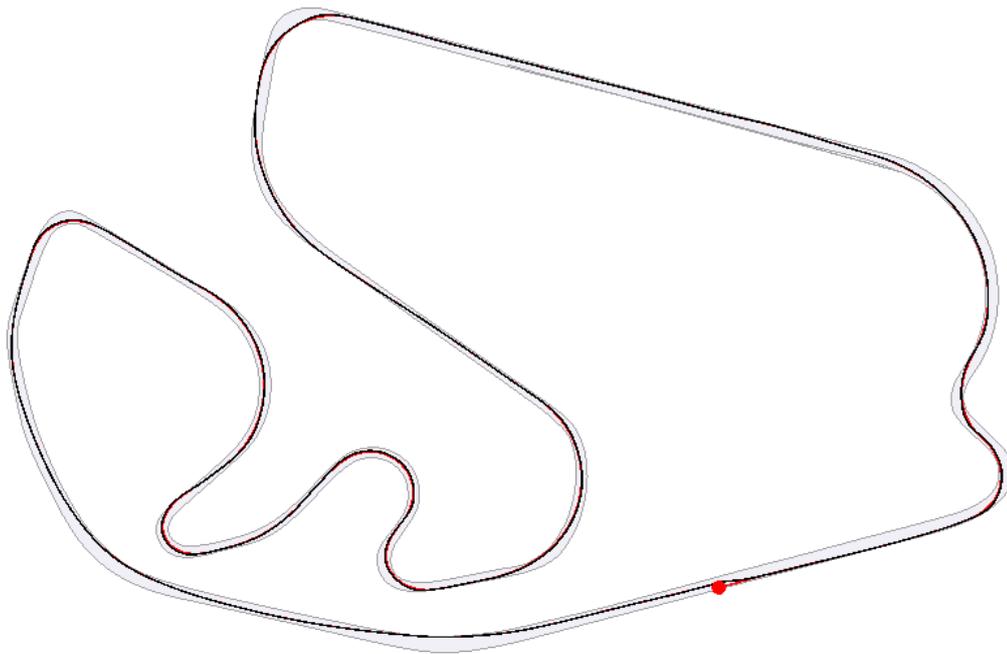
Figura 15 – Traçado da volta de referência



Fonte: elaborado pelo Autor.

O resultado de tempo obtido pelo controle automático com velocidade reduzida foi de 3:39.595, devido a limitação de velocidade em 70 km/h. A comparação do traçado realizado pelo sistema de controle básico é demonstrada na Figura 16, que exhibe o traçado automático na cor preta e o traçado de referência em cor vermelha, parcialmente sobreposto.

Figura 16 – Volta automática com limitação de 70 km/h



Fonte: elaborado pelo Autor.

Analisando a sequência inicial de curvas, é possível verificar uma pequena divergência entre a volta realizada de maneira automática e a volta padrão. Na Figura 17 é apresentada a aproximação para visualização da diferença de traçado existente.

Figura 17 – Volta automática com limitação de 70 km/h - Curvas 1 e 2



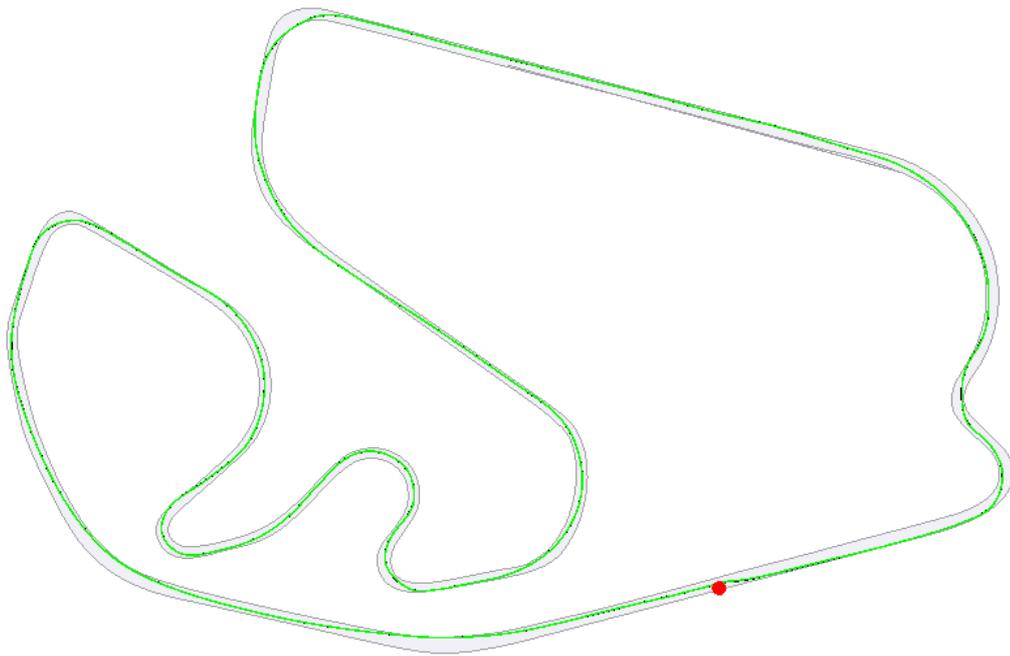
Fonte: elaborado pelo Autor.

A validação da etapa de controle é finalizada com a comprovação da capacidade do sistema realizar a pilotagem autônoma através da comparação com a volta de referência.

## 5.2 Validação da repetibilidade

O início da validação da repetibilidade da-se a partir da execução e comparação de duas voltas com velocidade reduzida. Na Figura 18, a seguir, é possível verificar a precisão do sistema de controle automático, onde basicamente a segunda volta, em verde, sobrepõe a primeira volta, em preto.

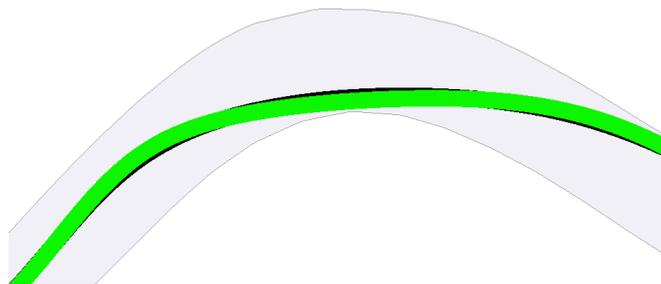
Figura 18 – Repetibilidade com limitação de 70 km/h



Fonte: elaborado pelo Autor.

Aplicando a aproximação na segunda curva é possível verificar que existe uma mínima diferença entre o traçado em verde do traçado em preto, verificada na Figura 19.

Figura 19 – Repetibilidade com limitação de 70 km/h - Curva 2



Fonte: elaborado pelo Autor.

A repetibilidade de traçado apresentada pela execução do sistema reflete também no tempo de conclusão da segunda volta, 3:39.535, diferença de apenas 60 milésimos de segundo. O Quadro 6, com tempos de voltas realizados utilizando o sistema de controle automático.

Quadro 6 – Tempos de volta com limitação de 70 km/h

<b>Volta</b>	<b>Tempo de volta</b>
1	3:39.595
2	3:39.535
3	3:39.483
4	3:39.563
5	3:39.503
6	3:39.539
7	3:39.526

Fonte: elaborado pelo autor.

Com a realização de 7 voltas em sequência que apresentaram uma diferença de 112 milésimos de segundo entre os resultados extremos. Este resultado apresenta consistência na repetição de voltas, porém não é suficiente para realizar uma validação visto que, a partir de voltas com velocidade reduzida, não se pode validar o objetivo de comparação de configurações diferentes de aerodinâmica. A partir dessa afirmação, são realizados os testes com sistema de controle completo, sem limitação de velocidade. Os resultados de tempo são apresentados no Quadro 7.

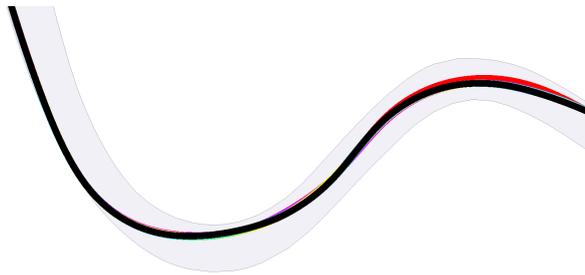
Quadro 7 – Tempos de volta

<b>Volta</b>	<b>Tempos de volta</b>
1	1:25.095
2	1:25.105
3	1:25.171
4	1:25.050
5	1:25.151
6	1:24.992
7	1:25.002

Fonte: elaborado pelo autor.

Assim como no teste com velocidade limitada, o sistema apresenta precisão na repetibilidade de voltas, sendo a diferença entre extremos de 179 milésimos de segundo, com diferença de apenas 15 segundos entre a volta de referência e as voltas replicadas. O traçado realizado pode ser visualizado na Figura 20.

Figura 20 – Repetibilidade - Curvas 1 e 2



Fonte: elaborado pelo autor.

A trajetória em vermelho refere-se a volta 3, e as demais estão sobrepostas pela volta 5, em preto. Para realizar validação de repetibilidade são observadas as telemetrias de 7 voltas manuais em sequência, que apresentam os tempos do Quadro 8.

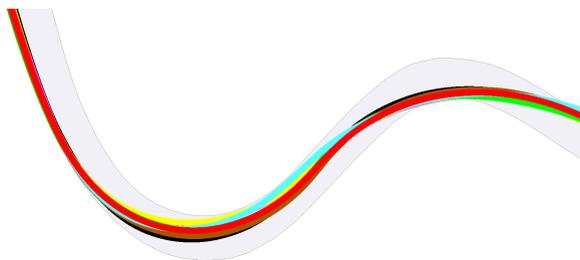
Quadro 8 – Tempos de volta manual

<b>Volta</b>	<b>Tempos de volta</b>
1	1:11.382
2*	1:11.226
3	1:19.877
4	1:11.038
5*	1:12.888
6*	1:11.732
7	1:10.622

Fonte: elaborado pelo autor.

As voltas 2, 5 e 6 apresentam tempos inválidos devido ao traçado ultrapassar os limites de pista. Aproximação das curvas 1 e 2 pode ser observada na Figura 21.

Figura 21 – Repetibilidade manual - Curvas 1 e 2



Fonte: elaborado pelo Autor.

Através destes dados valida-se a repetibilidade do sistema de controle automático, que apresenta maior precisão na reprodução de voltas em sequência, se comparado as voltas manuais.

### 5.3 Validação dos objetivos do projeto

Com a repetibilidade verificada, é realizado o processo de validação dos objetivos do projeto. Através do sistema de piloto automático, foram percorridas 24 voltas, sendo três voltas para cada configuração de carro. O Quadro 9 apresenta os tempos e configurações utilizadas em cada volta.

Quadro 9 – Voltas automáticas com diferentes configurações

<b>Volta</b>	<b>Pressão aerodinâmica</b>	<b>Pressão de freio</b>	<b>Balanco de freio</b>	<b>Tempos de volta</b>
1	F: 11 T: 11	95%	58%	1:25.515
2	F: 11 T: 11	95%	58%	1:25.442
3	F: 11 T: 11	95%	58%	1:25.494
4	F: 8 T: 8	95%	58%	1:25.041
5	F: 8 T: 8	95%	58%	1:25.103
6	F: 8 T: 8	95%	58%	1:25.085
7	F: 6 T: 6	95%	58%	1:24.888
8	F: 6 T: 6	95%	58%	1:24.825
9	F: 6 T: 6	95%	58%	1:24.942
10	F: 4 T: 4	95%	58%	1:24.945
11	F: 4 T: 4	95%	58%	1:24.952
12	F: 4 T: 4	95%	58%	1:25.014
13	F: 2 T: 2	95%	58%	1:25.030
14	F: 2 T: 2	95%	58%	1:24.956
15	F: 2 T: 2	95%	58%	1:25.046
16	F: 1 T: 1	95%	58%	1:24.902
17	F: 1 T: 1	95%	58%	1:24.964
18	F: 1 T: 1	95%	58%	1:24.859
19	F: 6 T: 6	100%	60%	1:24.989
20	F: 6 T: 6	100%	60%	1:24.935
21	F: 6 T: 6	100%	60%	1:25.196
22	F: 1 T: 1	100%	60%	1:25.209
23	F: 1 T: 1	100%	60%	1:25.054
24	F: 1 T: 1	100%	60%	1:24.982

Fonte: elaborado pelo autor.

Entre as voltas 19 e 24 foram repetidas as configurações de aerodinâmica que apresentaram melhores resultados de tempo total, porém com configurações de pressão e balanço de freio alteradas. Ao aumentar a pressão de freio e utilizar um balanço que apresenta maior força de frenagem nas rodas dianteiras do carro, ambas configurações apresentaram piores resultados de tempo. Destaca-se que, entre os conjuntos de mesmo *setup*, a repetibilidade do sistema apresenta aproximadamente 1 décimo de segundo de diferença nas 18 voltas iniciais.

### 5.3.1 Comparação de tempos

A comparação dos resultados utiliza as voltas manuais realizadas por um voluntário, com tempos apresentados no Quadro 11.

Quadro 10 – Voltas manuais do voluntário

<b>Volta</b>	<b>Pressão aerodinâmica</b>	<b>Pressão de freio</b>	<b>Balanço de freio</b>	<b>Tempos de volta</b>
1	F: 8 T: 8	95%	58%	1:36.763
2	F: 8 T: 8	95%	58%	1:48.060
3	F: 8 T: 8	95%	58%	1:41.419
4	F: 1 T: 1	95%	58%	1:38.847
5	F: 1 T: 1	95%	58%	1:38.028
6	F: 1 T: 1	95%	58%	2:20.531

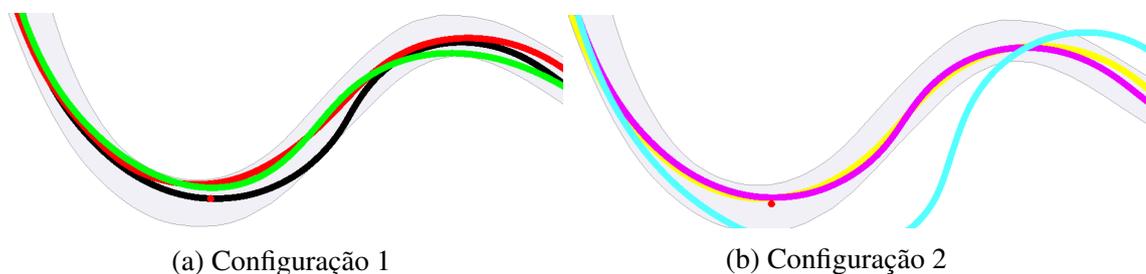
Fonte: elaborado pelo autor.

A comparação direta de tempos demonstra que voltas realizadas por um ser humano, que utiliza simuladores de maneira amadora, não apresentam uma repetibilidade. A mesma variação de tempo das voltas do voluntário pode ser visualizada nas voltas manuais apresentadas no Quadro 8. Considera-se que o modelo cumpre o objetivo de apresentar uma repetibilidade superior se comparado a coletas de telemetria de voltas realizadas manualmente.

### 5.3.2 Comparação de deslizamento

Dado que o parâmetro de tempo não possui uma significância relevante devido a grande variação nas voltas manuais, são analisados os dados de traçado, velocidade e esterçamento do volante para verificação da alteração da configuração do carro. A partir desses dados foi realizada a comparação com 3 voltas manuais realizadas em sequência por um jogador amador, para cada configuração de aerodinâmica. As sobreposições dos traçados são apresentadas na Figura 22.

Figura 22 – Volta do voluntário - Curvas 1 e 2



Fonte: elaborado pelo autor.

Observando apenas os traçados, não é verificada repetibilidade entre voltas. A falta de repetibilidade no traçado implica na adição de fatores externos à comparação. No Quadro 11, são apresentadas as telemetrias de velocidade e esterçamento no ponto intermediário da primeira curva, distante 340 metros da linha de largada, indicado em vermelho na Figura 22. A análise

destes fatores permite a verificação de deslizamento quando o carro está sob mesma velocidade e esterçamento, porém não realiza o mesmo traçado.

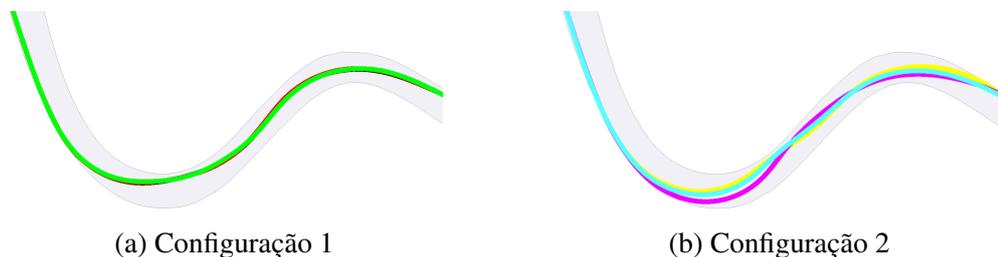
Quadro 11 – Voltas manuais do voluntário

<b>Volta</b>	<b>Cor</b>	<b>Configuração</b>	<b>Esterçamento</b>	<b>Velocidade</b>
1	Preto	1	-58,09%	125
2	Vermelho	1	-35,46%	87,43
3	Verde	1	-47,56%	96
4	Amarelo	2	-44,31%	112,54
5	Rosa	2	-44,38%	120,90
6	Azul	2	-100%	129,68

Fonte: elaborado pelo autor.

Nas voltas do voluntário, não existe repetibilidade de velocidade e de esterçamento para verificação. Os dados de deslizamento poderiam ser diretamente analisados, porém seriam inconclusivos, visto que o evento poderia ocorrer devido a diferença de velocidade carregada na curva e não ao diferente *setup* aerodinâmico utilizado. Extraindo os mesmos dados, porém das voltas simuladas, é possível observar os traçados na Figura 23.

Figura 23 – Volta do *software* - Curvas 1 e 2



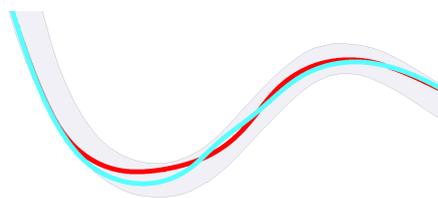
(a) Configuração 1

(b) Configuração 2

Fonte: elaborado pelo autor.

Neste caso verifica-se a repetibilidade do traçado realizado com configuração de maior pressão aerodinâmica. Analisando o traçado de menor pressão aerodinâmica verificam-se desvios no traçado, porém mais adequados a trajetória da pista, possibilitando a comparação entre os diferentes *setups*. Através da sobreposição de uma volta de cada *setup* é possível verificar o comportamento da diferença aerodinâmica no traçado, apresentada a seguir na Figura 24.

Figura 24 – Diferentes configurações sobrepostas - Curvas 1 e 2



Fonte: elaborado pelo autor.

É possível verificar um aumento no deslizamento ao longo da curva 1. Entretanto, para realizar a constatação deste fato é necessário observar as telemetrias de velocidade e esterçamento, apresentadas no Quadro 12.

Quadro 12 – Voltas manuais do voluntário

<b>Volta</b>	<b>Cor</b>	<b>Configuração</b>	<b>Esterçamento</b>	<b>Velocidade</b>
1	Preto	1	-54%	103
2	Vermelho	1	-51,27%	102
3	Verde	1	-70,14%	100
4	Amarelo	2	-99,99%	100
5	Rosa	2	-99,99%	97,66
6	Azul	2	-99,99%	99

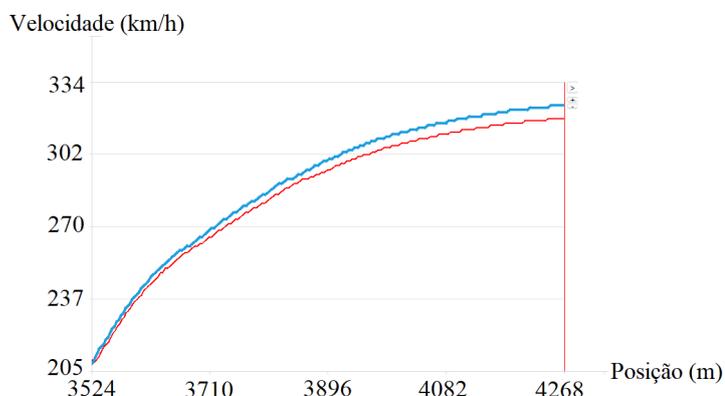
Fonte: elaborado pelo autor.

Diferentemente das voltas manuais, é possível verificar a semelhança nos parâmetros gerando diferentes resultados de traçado. Esta verificação possibilita a conclusão de que para a configuração de menor aerodinâmica é necessária uma menor velocidade ao longo da curva 1, afim de evitar o deslizamento e corrigir a trajetória.

### 5.3.3 Comparação de velocidade

Após a comprovação da eficácia do software para comparação de trajetórias de curva, é realizada a verificação do impacto da diferença aerodinâmica na aceleração e velocidade final. Através da visualização o gráfico de velocidade, disponibilizado no software de telemetria, é verificada a variação da velocidade ao longo da reta principal do autódromo de Interlagos, Figura 25.

Figura 25 – Diferença de velocidade



Fonte: elaborado pelo autor.

Em azul é apresentada a velocidade, ao longo da extensão do autódromo, utilizando a configuração de menor pressão aerodinâmica. A diferença de velocidade no ponto imediatamente anterior a linha de chegada é de 6 km/h. É atingida a velocidade de 325 km/h utilizando a configuração de menor pressão, contra 319 km/h para a configuração de maior pressão aerodinâmica. Todas as 6 voltas replicadas pelo software apresentaram os mesmos valores de velocidade, comprovando a eficácia do modelo para verificações de velocidade em retas.

#### 5.3.4 Expansão da aplicação

Com as eficácias de funcionalidade e repetibilidade comprovadas, o modelo foi submetido a um teste em um circuito diferente do proposto para desenvolvimento do projeto. O circuito Hungaroring foi escolhido devido a semelhanças com circuito de Interlagos, possuindo curvas de baixa e média velocidade e uma reta principal média. A volta de referência utilizada possui o tempo de 1:26.387, as voltas reproduzidas possuem os tempos apresentados no Quadro 13.

Quadro 13 – Tempos de volta em Hungaroring

<b>Volta</b>	<b>Tempo de volta</b>
1	1:51.258
2*	1:51.452
3	1:51.432

Fonte: elaborado pelo autor.

A capacidade de concluir voltas e apresentar tempos semelhantes em diferentes circuitos é comprovada a partir do teste realizado, assim como a expansibilidade de simuladores é comprovada através do desenvolvimento utilizando técnicas de orientação a objeto que permitem a substituição da classe de aquisição por um classe que opere para outros simuladores. Realizando a análise de resultados permite-se discorrer a conclusão do projeto.

## 6 CONCLUSÃO

O desenvolvimento do presente trabalho possuía o objetivo principal de minimizar os fatores de interferência durante as voltas de teste de diferentes ajustes. Além de possuir os objetivos específicos de avaliar a diferença de tempo de volta utilizando diferentes configurações de aerodinâmica ou freio, assim como segmentar o software para que seja possível alterar o modelo controlador e adaptar para diferentes simuladores.

Através dos resultados obtidos, é possível afirmar que os objetivos foram alcançados. O sistema possibilitou minimizar as interferências externas e isolar fatores de comparação para definir os impactos de cada configuração. Além disso, o desenvolvimento modular e a técnica de pilotagem desenvolvida expandem a funcionalidade para diferentes autódromos e possibilitam, através de programação, a adição de novos simuladores.

O papel de auxiliar no ajuste dos veículos, através de análises de telemetria, foi comprovado no modelo. Contudo, existem possibilidades de melhorias para trabalhos futuros, explorando técnicas de aperfeiçoamento que visem a aproximação do tempo da volta replicada através da utilização de controles de deslizamento e verificação do ângulo de esterçamento. Demais trabalhos, além da área automobilística, podem se beneficiar da técnica utilizada na estrutura base do *software*, que pode ser estendida para diferentes sistemas simuladores. Por exemplo o desenvolvimento de um sistema de pouso automático em simuladores de vôo para verificação do impacto das condições climáticas nos comandos da aeronave.

Visando facilitar o desenvolvimento de melhorias e implementações futuras, todos os códigos gerados ao longo do trabalho estão disponíveis no repositório SRV do autor, Goulart (2022), no GitHub. As bibliotecas utilizadas, assim como a forma de instalação e ferramentas utilizadas estão indicadas na documentação do projeto. Concluí-se assim que o trabalho obteve sucesso na apresentação de um modelo de sistema capaz de realizar a reprodução de voltas em simuladores de automobilismo.

## REFERÊNCIAS BIBLIOGRÁFICAS

- BECKMAN, B. The physics of racing series. 1991. Disponível em: <<http://ceb.ac.in/knowledge-center/E-BOOKS/Physics%20Of%20Racing%20Series%20-%20Brian%20Beckman.pdf>>. Citado na página 12.
- BUGEJA, K.; SPINA, S.; BUHAGIAR, F. Telemetry-based optimisation for user training in racing simulators. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. p. 31–38. ISBN 9781509058129. Citado 6 vezes nas páginas 12, 13, 14, 15, 16 e 17.
- DIGITALE Fabrik - Digitaler Fabrikbetrieb. 2011. Citado na página 15.
- F1 2020 UDP Specification - Technical Assistance - Codemasters Community. 2020. Disponível em: <<https://forums.codemasters.com/topic/50942-f1-2020-udp-specification/>>. Citado na página 22.
- GOULART, L. C. *Software de reprodução de voltas*. 2022. Disponível em: <<https://github.com/CarvoT/srv>>. Citado na página 44.
- LIMA, J. de Jesus Botelho de. Sistema antibloqueio (abs) para freios eletromecânicos utilizando controle por modos deslizantes. 8 2005. Disponível em: <<http://pee.ufrj.br/teses/textocompleto/2005080101.pdf>>. Citado na página 15.
- MACKINNON, B. *Adaptive AI in a racing simulator: a challenging and fun environment for any user*. 2012. 2012 p. Disponível em: <[www.computergamesjournal.com](http://www.computergamesjournal.com)>. Citado na página 17.
- MEISTER, O. *et al.* Software-in-the-loop simulation for small autonomous vtol uav with teaming capability. *ARPN Journal of Engineering and Applied Sciences*, v. 11, p. 9789–9795, 2016. Citado na página 16.
- MICHAELIS On-line | Telemetria. 2022. Disponível em: <<https://michaelis.uol.com.br/palavra/V42ye/telemetria/>>. Citado na página 11.
- NARAYANASAMY, V. *et al.* *Distinguishing simulation games from simulators by considering design characteristics Sugeno-Yasukawa qualitative modeling View project A Structural Equation Modelling Approach for Adoption of e-book Amongst Mathematics and Statistics (MAS) Students at Higher Education Institutions in Libya View project Distinguishing Simulation Games from Simulators by considering Design Characteristics*. 2005. Disponível em: <<https://www.researchgate.net/publication/43980435>>. Citado na página 10.
- NIGGEL, M. *Advanced Racing: Hitting the Apex | Propwashed*. 2017. Disponível em: <<https://www.propwashed.com/advanced-racing-hitting-apex/>>. Citado na página 13.
- ORICCHIO, L. *Capítulo 1: A F1 mudou em 94, e a Williams de Senna não era mais a mesma*. 2014. Disponível em: <<https://www.uol.com.br/esporte/f1/ultimas-noticias/2014/04/28/capitulo-1-a-f1-mudou-em-94-e-a-williams-de-senna-nao-era-mais-a-mesma.htm>>. Citado na página 15.
- OVERSTEER / Understeer Objective measurement of tyre oversteer using VBOX equipment. 2015. Disponível em: <<https://www.vboxautomotive.co.uk/downloads/Oversteer-Understeer%20Analysis%20in%20Tyre%20Testing.pdf>>. Citado na página 14.

PITTENGER, W. *Autódromo José Carlos Pace*. 2009. Disponível em: <[https://commons.wikimedia.org/wiki/File:Aut%C3%B3dromo\\_Jos%C3%A9\\_Carlos\\_Pace\\_\(AKA\\_Interlagos\)\\_track\\_map.svg](https://commons.wikimedia.org/wiki/File:Aut%C3%B3dromo_Jos%C3%A9_Carlos_Pace_(AKA_Interlagos)_track_map.svg)>. Citado na página 24.

RAHMAN, A. A. A.; MOHAMAD, N. R. Software-in-the-loop technique: An approach to support reconfiguration of manufacturing system. v. 11, 2016. ISSN 1819-6608. Disponível em: <[www.arpnjournals.com](http://www.arpnjournals.com)>. Citado na página 15.

REMONDA, A.; VEAS, E.; LUZHNICA, G. Comparing driving behavior of humans and autonomous driving in a professional racing simulator. *PLoS ONE*, Public Library of Science, v. 16, 2 2021. ISSN 19326203. Citado na página 16.

SANTOS, R. *3 ways in which suspension affects understeer and oversteer*. 2015. Disponível em: <<http://racingcardynamics.com/understeer-and-oversteer/>>. Citado na página 20.

SPURGEON, B. *Lewis Hamilton's great performance, and his great preparation - The New York Times*. 2007. Disponível em: <<https://rendezvous.blogs.nytimes.com/2007/06/09/lewis-hamiltons-great-performance-and-his-great-preparation/>>. Citado na página 10.

WILLIAMS, D. 3.2: *Fundamentals: The optimal racing line - Virtual Racing School (VRS)*. 2016. Disponível em: <<https://virtualracingschool.com/academy/racing-career-guide/second-season/the-optimal-racing-line/>>. Citado 2 vezes nas páginas 13 e 14.