

**UNIVERSIDADE FEDERAL DE SÃO CARLOS – UFSCAR
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA – CCET
DEPARTAMENTO DE ENGENHARIA MECÂNICA – DEMEC**

HENRIQUE ARON FERREIRA

**CLASSIFICAÇÃO DE DESCONTINUIDADES EM VIAS
ASFALTADAS USANDO REDES NEURAIIS TREINADAS A
PARTIR DE MODELOS FÍSICOS DA DINÂMICA
VERTICAL DE VEÍCULOS**



**SÃO CARLOS
2023**

HENRIQUE ARON FERREIRA

**CLASSIFICAÇÃO DE DESCONTINUIDADES EM VIAS ASFALTADAS USANDO
REDES NEURAS TREINADAS A PARTIR DE MODELOS FÍSICOS DA DINÂMICA
VERTICAL DE VEÍCULOS**

Trabalho de conclusão de curso apresentada ao Curso de Graduação em Engenharia Mecânica da Universidade Federal de São Carlos, para obtenção do título de Bacharel em Engenharia Mecânica.

Orientador: Prof. Dr. Sidney Bruce Shiki

São Carlos

2023

*

Henrique Aron Ferreira

Classificação de descontinuidades em vias asfaltadas usando redes neurais treinadas a partir de modelos físicos da dinâmica vertical de veículos/ Henrique Aron Ferreira. – São Carlos, 2023-

83p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Sidney Bruce Shiki

Trabalho de conclusão de curso – Universidade Federal de São Carlos – UFSCar
Centro de Ciências Exatas e de Tecnologia – CCET
Departamento de Engenharia Mecânica – DEMec, 2023.

1. Redes neurais artificiais. 2. Dinâmica veicular. 2. Descontinuidades em vias asfaltadas. I. Prof. Dr. Sidney Bruce Shiki . II. Universidade Federal de São Carlos. III. Faculdade de Engenharia Mecânica. IV. Classificação de descontinuidades em vias asfaltadas usando redes neurais treinadas a partir de modelos físicos da dinâmica vertical de veículos



FUNDAÇÃO UNIVERSIDADE FEDERAL DE SÃO CARLOS
COORDENAÇÃO DO CURSO DE ENGENHARIA MECÂNICA - CCEMec/CCET
 Rod. Washington Luís km 235 - SP-310, s/n - Bairro Monjolinho, São Carlos/SP, CEP 13565-905
 Telefone: (16) 33519703 - <http://www.ufscar.br>

DP-TCC-FA nº 37/2023/CCEMec/CCET

Graduação: Defesa Pública de Trabalho de Conclusão de Curso

Folha Aprovação (GDP-TCC-FA)

FOLHA DE APROVAÇÃO

SIDNEY BRUCE SHIKI

**CLASSIFICAÇÃO DE DESCONTINUIDADES EM VIAS ASFALTADAS USANDO REDES NEURAS TREINADAS A PARTIR DE
 MODELOS FÍSICOS DA DINÂMICA VERTICAL DE VEÍCULOS**

Trabalho de Conclusão de Curso

Universidade Federal de São Carlos – Campus São Carlos

São Carlos, 06 de setembro de 2023

ASSINATURAS E CIÊNCIAS

Cargo/Função	Nome Completo
Orientador	Sidney Bruce Shiki
Membro da Banca 1	Alexandre Tácito Malavolta
Membro da Banca 2	Vitor Ramos Franco



Documento assinado eletronicamente por **Sidney Bruce Shiki, Docente**, em 06/09/2023, às 18:14, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Vitor Ramos Franco, Docente**, em 06/09/2023, às 18:14, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Alexandre Tacito Malavolta, Docente**, em 06/09/2023, às 18:14, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site <https://sei.ufscar.br/autenticacao>, informando o código verificador **1171468** e o código CRC **99B25267**.

Referência: Caso resposta a este documento, indicar expressamente o Processo nº 23112.032291/2023-38

SEI nº 1171468

Modelo de Documento: Grad: Defesa TCC: Folha Aprovação, versão de 02/Agosto/2019

Dedico este trabalho aos meus pais. Que sempre me desafiaram a expandir minhas fronteiras, de conhecimento e de tomada de decisão, utilizando todos os aspectos da realidade para melhor fazê-lo.

AGRADECIMENTOS

Agradeço a meus professores e mentores, os quais sempre me trouxeram desafios e reflexões, agregando não só conhecimento técnico, mas à forma de análise da realidade e de solução de problemas.

"Quando aceitamos nossos limites, conseguimos ir além deles."

Albert Einstein

RESUMO

Com o crescimento das cidades, tem sido notável o maior despenho de tempo das pessoas dentro dos meios de transporte em vias asfaltadas, com isso, diversas problemáticas relacionadas à ergonomia emergem. Neste contexto, este trabalho de conclusão de curso propõe o desenvolvimento de uma metodologia de treinamento de redes neurais artificiais para a classificação de discontinuidades em vias asfaltadas, empregando modelos físicos da dinâmica vertical de veículos e sinais de aceleração medidos por dispositivos móveis. Inicialmente, foi elaborado um modelo físico discreto de carro completo com 7 graus de liberdade em ambiente computacional, buscando representar a dinâmica veicular vertical. A partir desse modelo, obtiveram-se estimativas de aceleração vertical do veículo, simulando um *smartphone* posicionado no painel do automóvel. Posteriormente, foi implementado um algoritmo de classificação com base em redes neurais artificiais, visando analisar os sinais de vibração do veículo e identificar diferentes discontinuidades em estradas, como lombadas, buracos e estradas normais. O algoritmo de classificação foi submetido a testes com dados simulados, considerando distintas discontinuidades e condições de velocidade do veículo, além de ser parametrizado para permitir a alteração da velocidade no modelo. Por fim, realizou-se o treinamento e a validação da rede neural, demonstrando a eficácia da metodologia proposta na criação de sistemas capazes de identificar discontinuidades em vias asfaltadas. Tal contribuição auxilia na melhoria da qualidade das estradas e, conseqüentemente, na segurança e conforto dos usuários.

Palavras-chave: Redes neurais artificiais. Dinâmica veicular. Descontinuidades em vias asfaltadas. Modelo de carro completo.

ABSTRACT

With the growth of cities, there has been a noticeable increase in the amount of time people spend within transportation means on paved roads. Consequently, various ergonomic-related issues arise. In this context, this undergraduate thesis proposes the development of a methodology for training artificial neural networks to classify discontinuities in asphalt roads, using physical models of vehicle vertical dynamics and acceleration signals measured by mobile devices. Initially, a discrete full-car physical model with 7 degrees of freedom was developed in a computational environment, aiming to represent the vertical vehicle dynamics. From this model, estimates of the vehicle's vertical acceleration were obtained, simulating a smartphone placed on the car's dashboard. Subsequently, a classification algorithm based on artificial neural networks was implemented, aiming to analyze the vehicle's vibration signals and identify different discontinuities in roads, such as speed bumps, potholes, and normal roads. The classification algorithm was tested with simulated data, considering various discontinuities and vehicle speed conditions, and it was parameterized to allow for changes in the model's speed. Finally, the neural network was trained and validated, demonstrating the effectiveness of the proposed methodology in creating systems capable of identifying discontinuities in asphalt roads. This contribution aids in improving road quality and, consequently, the safety and comfort of users.

Keywords: Artificial neural networks. Vehicle dynamics. Discontinuities in asphalt roads. Modeling. Full-car model.

LISTA DE FIGURAS

Figura 1 – Modelo de carro completo.	21
Figura 2 – Diagrama de corpo livre do modelo discreto do sistema.	21
Figura 3 – Diagrama da arquitetura de Rede Neural do sistema	28
Figura 4 – Visualização do efeito da camada de convolução 1D	29
Figura 5 – Percurso das rodas no obstáculo de lombada	32
Figura 6 – Percurso das rodas no obstáculo de sonorizador	33
Figura 7 – Percurso das rodas no obstáculo de buraco	33
Figura 8 – Percurso das rodas no obstáculo de tachão	34
Figura 9 – Percurso das rodas na estrada normal	35
Figura 10 – Deslocamento das rodas no obstáculo de lombada	36
Figura 11 – Resultado da simulação para Obstáculo - Lombada	37
Figura 12 – Deslocamento das rodas no obstáculo de sonorizador	38
Figura 13 – Resultado da simulação para Obstáculo - Sonorizador	39
Figura 14 – Deslocamento das rodas no obstáculo de buraco	40
Figura 15 – Resultado da simulação para Obstáculo - Buraco	41
Figura 16 – Deslocamento das rodas no obstáculo de tachão	42
Figura 17 – Resultado da simulação para Obstáculo - Tachão	43
Figura 18 – Deslocamento das rodas no obstáculo de Estrada normal	44
Figura 19 – Resultado da simulação para estrada normal	45
Figura 20 – Gráfico de Dispersão da inferência para o obstáculo de Lombada	46
Figura 21 – Matriz de Confusão da inferência para o obstáculo de Lombada	47
Figura 22 – Gráfico de Dispersão da inferência para o obstáculo de Sonorizador	48
Figura 23 – Matriz de Confusão da inferência para o obstáculo de Sonorizador	49
Figura 24 – Gráfico de Dispersão da inferência para o obstáculo de Buraco	50
Figura 25 – Matriz de Confusão da inferência para o obstáculo de Buraco	51
Figura 26 – Gráfico de Dispersão da inferência para o obstáculo de Tachão	52
Figura 27 – Matriz de Confusão da inferência para o obstáculo de Tachão	53
Figura 28 – Gráfico de Dispersão da inferência para o obstáculo de Estrada normal	53
Figura 29 – Matriz de Confusão da inferência para o obstáculo de Estrada normal	54
Figura 30 – Matriz de confusão do Modelo de Rede Neural	54

LISTA DE TABELAS

Tabela 1 – Constantes de massa, rigidez, amortecimento, comprimento e inércia	30
---	----

LISTA DE ABREVIATURAS E SIGLAS

CG	Centro de Gravidade
CNN	Convolutional Neural Network (Rede Neural de Convolução)
GRU	Gated Recurrent Unit (Unidade Recorrente com Portão)
LSTM	Long short-term memory (Memória de Longo Prazo de Curto Prazo)
PINN	Physics-informed Neural Network (Rede Neural Físico-informada)
ReLU	Rectified Linear Unit (Unidade Linear Retificada)
RNFI	Rede Neural Físico-informada
RNN	Recursive Neural Network (Rede Neural Recursiva)
IoT	Internet of Things (Internet das Coisas)
GDL	Grau de Liberdade
SVM	Support Vector Machine (Máquina de Vetor-Suporte)

LISTA DE SÍMBOLOS

\in	Pertence
\vec{r}	Vetor raio
ω_c	Velocidade angular da estrutura do veículo
$\vec{\omega}_c$	Vetor velocidade angular
$\vec{\alpha}$	Vetor aceleração angular

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVOS GERAIS	17
1.2	OBJETIVOS ESPECÍFICOS	17
2	REVISÃO DA LITERATURA	18
2.1	APRENDIZADO DE MÁQUINA E REDES NEURAIS	18
2.2	AQUISIÇÃO DE DADOS DE DESCONTINUIDADES NO PAVIMENTO	19
3	MODELO E METODOLOGIA	21
3.1	MODELO VEICULAR ESCOLHIDO	21
3.2	MODELO DE REDE NEURAL	27
3.2.1	Arquitetura	27
3.2.2	Geração de dados de treinamento para Rede Neural	29
3.3	PARÂMETROS DE SIMULAÇÃO	30
3.3.1	Geração de ruído	31
3.4	PERCURSO TEÓRICO DA SIMULAÇÃO E OBSTÁCULOS	31
3.4.1	Lombada	31
3.4.2	Sonorizador	32
3.4.3	Buraco	33
3.4.4	Tachão	34
3.4.5	Estrada normal	34
4	RESULTADOS	36
4.1	RESPOSTA CINEMÁTICA E CINÉTICA DO MODELO MATEMÁTICO	36
4.1.1	Lombada	36
4.1.2	Sonorizador	37
4.1.3	Buraco	39
4.1.4	Tachão	40
4.1.5	Estrada normal	41
4.2	CLASSIFICAÇÃO DE OBSTÁCULOS POR REDE NEURAL	43
4.2.1	Lombada	44
4.2.2	Sonorizador	46
4.2.3	Buraco	47
4.2.4	Tachão	48
4.2.5	Estrada normal	49
4.2.6	Desempenho Geral do Modelo	50

5	CONCLUSÃO	55
	REFERÊNCIAS	56
	APÊNDICE A – ROTINA COMPUTACIONAL MATLAB PARA SIMULAÇÃO	58
	APÊNDICE B – ROTINA COMPUTACIONAL PYTHON DE PRÉ-PROCESSAMENTO DOS DADOS DE SIMULAÇÃO	74
	APÊNDICE C – ROTINA COMPUTACIONAL PYTHON DE APRENDIZADO DE MAQUINA	77
	APÊNDICE D – ELEMENTOS DA MATRIZ DE AMORTECIMENTO	82
	APÊNDICE E – ELEMENTOS DA MATRIZ DE RIGIDEZ	83

1 INTRODUÇÃO

O estudo da dinâmica veicular ganha mais relevância na medida em que o domínio da paisagem urbana cresce. Conceitos como ergonomia física e cognitiva colocam-se como variáveis importantes na construção do bem-estar do cidadão do século XXI. Tais estudos visam construir uma sociedade mais eficiente e produtiva (DIAS; MUKAI; FEIBER, 2006).

Os aspectos mecânico-cinéticos presentes na dinâmica de veículos sob vias pavimentadas representam parcela significativa da interação do indivíduo com o meio, podendo somar períodos maiores que 1 hora todos os dias (TRANSITO, 2022), acresce-se a isso o aumento no uso de transportes individuais e se obtém tamanha relevância da relação entre os efeitos produzidos por esse dinâmica e métodos eficientes de identificação e melhorias dos mesmos (CARVALHO, 2016).

Dentre o aspecto mais proeminente na composição da dinâmica veicular tem-se a suspensão do veículo, nesse caso podendo-se generalizar à composição do sistema pneu-mola-amortecedor. Desse modo, o sistema composto de um conjunto de molas e amortecedores, pode ser considerados como um filtro mecânico de frequência, possuindo seus modos de vibrar, reposta em frequência característica e assim podendo amplificar ou detrair vibrações de determinada periodicidade e magnitude do conjunto de perturbações vindas do solo (MINOZZO, 2012).

Continuando a discussão sobre a dinâmica veicular, é importante ressaltar que a carga principal nas rodas de um eixo específico pode ser ajustada, independentemente da distribuição de carga determinada pela localização do centro de gravidade. Isso é possível através da utilização de eixos dianteiro e traseiro com diferentes tipos de suspensão e rigidez de molas. A rigidez, por sua vez, pode ser modificada através da seleção de molas específicas e do uso de estabilizadores (LEAL; ROSA; NICOLAZZI, 2001). A ausência de molas e amortecedores comprometeria seriamente o conforto, assim inviabilizando o uso do veículo em terrenos menos preservados ou para viagens de maiores distâncias (DEODATO, 2019).

Se apropriando dos aspectos mecânicos citados e agregando o aumento das cidades, da sua malha rodoviária e também do tempo de permanência média das pessoas em deslocamento por veículos automotores, como mostrado em IBGE (2023) e CBTU e TrensUrb (2022), o assunto de ergonomia se coloca de maneira central para uma população que gasta 35% do dia em média dentro de um veículo. Por conseguinte, métodos que nos permitam viabilizar essa análise, barateando-a e diminuindo barreiras de entrada para fazê-la, podem ser empregados para diminuição das discontinuidades no pavimento e, com ajuda da dinâmica veicular e equivalências matemáticas do meio, permitindo a identificação de cenários críticos para uma melhor ergonomia no trânsito, assim aumentando a agilidade no processo compreendido entre a criação de imperfeições no asfalto e seu reparo.

Nesse contexto, a utilização da tecnologia pode auxiliar na detecção de imperfeições no pavimento, assim permitindo a melhoria passiva dos trechos frequentados por motoristas, sejam eles de veículos de pequeno, médio ou grande porte. Para tal, tem-se o *smartphone* como o instrumento de uso diário adotado em massa, estando presente no dia a dia de 84,4% dos brasileiros segundo o IBGE (2022).

Esse dispositivo possui um conjunto de sensores capazes de não só medir acelerações do movimento nos 3 eixos principais, mas também é equipado com giroscópio, capaz de identificar giros e suas acelerações e compor um mapa de deslocamento completo. Derivado dessas capacidades, pode-se colocá-lo como ferramenta central análise de vibrações, em maior ou menor frequência e amplitude (HARDIYANTI; LAWI; DIARAYA, 2018), tais quais as produzidas pelos veículos anteriormente citados.

Aliado à identificação dos sinais pertinentes para a análise do problema de vibrações causadas pelas imperfeições do solo na dinâmica veicular, coloca-se a utilização de redes neurais que, por sua vez, conseguem identificar padrões não "visíveis" ou programáveis por algoritmos de computador. Possuem também a capacidade de se adaptar a diferentes tipos de fenômenos de alvo para a análise, aprendendo continuamente a partir de um reino de dados selecionados de interesse, representando a gama de variações a se analisar, podendo até extrapolar-lá, permitindo uma expansão dos cenários de análise além do esperado a partir de quanto mais é treinada em uma abrangência de cenários (CERRI; BARROS; CARVALHO, 2014).

A capacidade das redes neurais de se adaptar e resolver problemas com dados ainda a se determinar a torna adequada para o problema de identificação de imperfeições em ruas pavimentadas, visto a alta variabilidade e baixa repetibilidade de micro-imperfeições e outros aspectos do terreno, tal qual material e angulação. Assim, dado um certo conjunto de dados de treinamento, no caso de buscar maiores capacidades de análise basta incrementar os dados de treinamento da rede, levando à maior abrangência e precisão dos resultados alcançados, sem a necessidade de alterações arquiteturais na concepção do modelo ou retreinamento completo.

Assim, aliando o sensoriamento de ponta presente no dispositivos inteligentes de telefonia móvel com o modelos teóricos cinético-mecânicos, tais quais o de carro completo a ser apresentado, e versatilidade e capacidade preditiva de redes neurais físico-baseadas, fecha-se o ciclo necessário para análise preditiva dos pavimentos em veículos automotores de 4 rodas em quase tempo real, permitindo da coleta de dados do sensoriamento participativo proposto o desempenho de melhorias e planejamento urbano e rodoviário eficiente fundamentado e dados factuais e anônimos dos colaboradores, pavimentando um futuro mais ergonômico e ortopedicamente mais longo.

1.1 OBJETIVOS GERAIS

Desenvolver uma metodologia de treinamento de redes neurais artificiais visando a classificação de discontinuidades em vias asfaltadas usando como base um modelo dinâmico de carro completo e geração dos sinais de aceleração por simulando uma captação por dispositivos móveis.

1.2 OBJETIVOS ESPECÍFICOS

- Desenvolver em ambiente computacional um modelo físico discreto de carro completo representativo da dinâmica veicular vertical com 7 graus de liberdade;
- A partir do modelo dinâmico desenvolvido, estimar as medidas de aceleração vertical do veículo que seriam capturadas por um *smartphone* fixado ao painel do veículo;
- Implementar um algoritmo de classificação de sinais de vibração de um veículo baseado em redes neurais artificiais com o intuito de detectar discontinuidades em estradas;
- Testar o algoritmo de classificação com dados simulados sob diferentes discontinuidades e condições de velocidade do veículo.

2 REVISÃO DA LITERATURA

Para a fundamentação teórica dos processos até agora introduzidos, é em sequência descrita a revisão da literatura que embasa, progride e justifica matematicamente o desenvolvimento a ser apresentado. Para isso, tangencia os temas de sensoriamento e captação de informações de dispositivos móveis, modelagem teórica de carro completo, assim trazendo uma visão matemática do sistema e, também, dos conceitos relacionados a um modelo de rede neural a ser arquitetada, treinada e construída para identificação automática de imperfeições do pavimento.

2.1 APRENDIZADO DE MÁQUINA E REDES NEURAIAS

Para o correto processamento dos dados gerados a partir do equacionamento teórico, é necessário a escolha de um modelo que seja capaz de interpretar os dados e fazer as conexões ocultas da rede neural com tendências físicas, assim aumentando a acurácia no momento de fazer as predições seguintes.

Dentro da literatura, avanços e experimentações recentes mostraram que as *PINNs* (Redes Neurais Físico-Informadas, do inglês do inglês *Physics-Informed Neural Networks*, *RFNIs* daqui em diante), tem obtido resultados proeminentes no que tange a problemas de equacionamento não-linear, integração e solução equações diferenciais ordinárias ou parciais (CUOMO et al., 2022). A concepção dessas redes neurais se baseiam em redes neurais "convencionais", todavia imbuídas de uma base de treinamento de dados com viés físico-matemático, que leva à solução mais próxima do teórico esperado (BLECHSCHMIDT; ERNST, 2021).

Uma das variáveis a serem consideradas dentro da gama de capacidades necessárias do modelo neural a ser levantado, é a capacidade de lidar com tempo discreto nos dados. Com o uso de novas abordagens no processamento dos dados e na arquitetura de algoritmos de aprendizado computacional profundo, é possível alcançar melhor capacidade de previsão a partir de *RFNIs* (KOLLMANNNSBERGER et al., 2021).

Como notado por Cuomo et al. (2022), a camada de ativação presente na rede neural tem papel importante no desempenho. Para cenários de não-linearidade, a função de ativação "*swish*", tida como $x \cdot \text{Sigmoid}(\beta x)$ obteve o melhor resultado quando comparado com outras funções mais simples. Todavia, a utilização de abordagens menos complexas e facilmente diferenciáveis, as quais corroboram para a velocidade de treinamento e inferência do modelo, são possíveis, como por exemplo a função de ativação *ReLU* (Unidade Linear Retificada, do inglês *Rectified Linear Unit*), ou até mesmo o *Sigmoid* isolado.

Na busca por otimizar o desempenho do modelo de aprendizado de máquina para detecção de buracos usando dados de acelerômetro, foi possível obter caminhos baseado em vários estudos de pesquisa. Como notado por Wang et al. (2021), foi demonstrado a superioridade

das Unidades Recorrentes com Porta (*GRUs*, do inglês *Gated Recurring Unit*) sobre as unidades de Memória de Longo Prazo de Curto Prazo (*LSTM*, do inglês *Long-Short Time Memory*) em termos de precisão de classificação, apesar da simplicidade relativa das primeiras. Essa descoberta levou a considerar a substituição das unidades *LSTM* no modelo por *GRUs*, potencialmente melhorando o desempenho do modelo sem aumentar sua complexidade.

Além disso, o uso de Redes Neurais Convolucionais (*CNNs*, do inglês *Convolutional Neural Networks*) para classificação de séries temporais produz resultados promissores, já que as *CNNs* podem extrair automaticamente características locais dos dados de entrada. Desse modo, é possível considerar um modelo híbrido que aproveita os pontos fortes das *CNNs* e *RNNs* (Redes Neurais Recorrentes, do inglês *Recurring Neural Networks*), potencialmente aprimorando a capacidade do modelo de aprender com os dados do acelerômetro (UM; BABAKESHIZADEH; KULIĆ, 2016) (GALDO et al., 2022).

Mais a fundo, Um, Babakeshizadeh e Kulić (2016) também propuseram uma abordagem inovadora de formatação de dados de séries temporais como imagens e alimentação dessas no *CNN*. Esta representação baseada em imagens dos dados de séries temporais poderia potencialmente melhorar a capacidade do modelo de extrair características significativas dos dados, corroborando especialmente ao incorporar uma *CNN* no modelo.

Como possível outro ponto de aprimoramento e mudança de abordagem para solução do problema, Sharma, Lee e Chung (2011) enfatizou os possíveis benefícios da engenharia de características. O estudo sugeriu que calcular características adicionais a partir das leituras brutas do acelerômetro, como a magnitude do vetor de aceleração, poderia melhorar o desempenho do modelo. Nesse caso essa abordagem levaria a uma reestruturação dos dados de entrada e significativo aumento da complexidade e especificidade do uso do modelo, já na fase de pré-treinamento e portanto não foi utilizada.

2.2 AQUISIÇÃO DE DADOS DE DESCONTINUIDADES NO PAVIMENTO

De modo a viabilizar as análises seguintes, é importante delimitar um processo de detecção das imperfeições nas vias pavimentadas, visto inclusive avanços no sensoriamento e usos de dispositivos móveis, equipados com acelerômetros e receptores GPS para geolocalização.

Foi notado por Sabah e Şimşek (2017) que o uso de aparelhos móveis que possuem acelerômetros e giroscópios é viável para a detecção de imperfeições nas vias asfálticas. Dentre essas imperfeições é relevante não somente verificar que elas acontecem, mas também de que tipo são, cada uma com suas devidas características e efeitos. Todo esse processo, com o uso de dispositivos móveis é possível, se utilizando do sensoriamento cinético existente e geolocalização, permitindo uma análise de baixo custo e com velocidade, tanto de detecção dos dados quanto de simples extração.

O uso de ferramentas de IoT (Internet das Coisas, do inglês *Internet of Things*) é

outra abordagem disponível para a verificação e análise de cenários correlatos. Para o uso do mesmo, algumas técnicas são relevantes para a produção de melhores resultados relacionados (BHAMARE et al., 2021). Nesse caso, permite-se a construção de uma rede próxima a malha de "tempo-real", para um menor tempo de reação a partir da detecção de sua existência.

A abordagem para a execução da análise de condições de pavimentos não se resume somente ao uso de redes neurais. Métodos baseados em delimitação de limites de movimentação e, portanto, uso de "gatilhos" para a detecção das imperfeições também é possível. Estes, porém, tem "inteligência" pré-definida, visto que são implementações baseadas em algoritmos e não atreladas a modelos que podem ter seu aprendizado expandido, tanto em termos de variação de condições esperadas, quanto em variáveis ocultas do processo, porém ainda relevantes. O uso de métodos baseados em algoritmos pré-implementados possuem o benefício de não requerem uma etapa de pré-treinamento e já embutirem consigo as necessidades de pré-processamento do sinal, porém dado sua inflexibilidade, tem ficado para trás de outras abordagens dinâmicas (CHHABRA; SINGH, 2021).

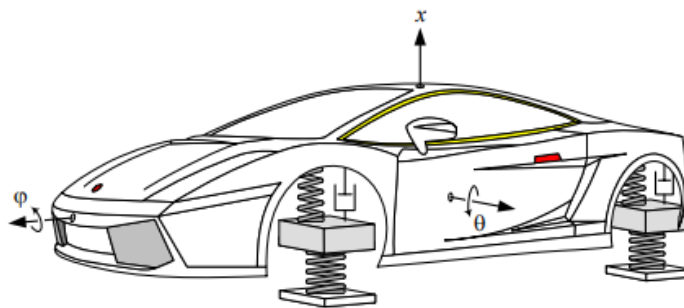
Outros métodos de detecção de discontinuidades e aspereza de vias pavimentadas com o uso de redes neurais também foram explorados. Como constatado por Nitsche et al. (2014), SVM (Máquina de Vetor-Suporte, do inglês *Support Vector Machine*) e uma rede neural multicamada de *Perceptrons* são adequadas para a inferência de cenários de imperfeições e podem produzir resultados satisfatórios. Para tal, porém, é necessário que esteja claro que haja uma extração adequada de pontos importantes (do inglês, *features*) por parte dos modelos, o que implica necessidade de dados balanceados e uma estrutura de camadas de rede neural suficientemente complexa para captar cada uma delas.

3 MODELO E METODOLOGIA

3.1 MODELO VEICULAR ESCOLHIDO

Para a interpretação mais abrangente quanto a resposta veicular, em relação a modelos simplificados, que podem ofuscar alguns aspectos presentes no cenário factual, tais quais a resposta de barras anti-rolagem presentes no carro. Buscou-se tal fidedignidade no modelo de carro completo, como proposto por Jazar (2008). A figura 1 demonstra o esboço isométrico e o referencial de eixos utilizados para o equacionamento posterior.

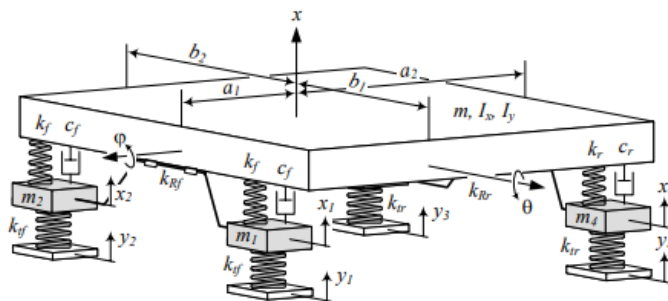
Figura 1 – Modelo de carro completo.



Fonte: Jazar (2008)

A partir da visualização, é possível construir o diagrama de corpo livre do sistema, composto por 7 GDL, assim dispostos na figura 2, constando as massas, molas e amortecedores entendidos no sistema e também todos os elementos no equacionamento.

Figura 2 – Diagrama de corpo livre do modelo discreto do sistema.



Fonte: Jazar (2008)

Na discretização apontada, tem-se os elementos: a é a distância entre o centro de gravidade e os eixos frontal e traseiro. b é a distância do centro de gravidade e as rodas esquerda e direita, x é o deslocamento vertical do Centro de Gravidade do modelo, φ o giro entre lado esquerdo e direito do carro, ou rolagem, θ o giro entre frente e trás, ou arfagem, m é a massa

do veículo I_x e I_y os momentos de inércia longitudinal e lateral, respectivamente, x_i são os deslocamentos verticais das rodas, m_i são as massas das rodas, k_t são as rigidezes dos pneus, k as rigidezes da suspensão, k_R são as rigidezes torcionais das barras anti-rolagem, c os coeficientes de amortecimento da suspensão e y_i são os deslocamentos que excitarão a estrutura e serão o dados de entrada deste estudo. Os índices f e r denotam frontal e traseiro, respectivamente.

Para o desenvolvimento do modelo, é necessário assumir algumas premissas simplificadoras, diminuindo a complexidade matemática e o número de parâmetros de entrada necessárias para a determinação do equacionamento. Dentre elas, assume-se:

1. Simetria entre lado direito e esquerdo quanto à rigidez das molas para ambos os pneus e suspensão.
2. Simetria entre lado direito e esquerdo quanto ao coeficiente de amortecimento
3. Coeficiente de amortecimento dos pneus já considerados na suspensão e portanto desprezado na modelagem do pneu

Desse modo, diferencia-se somente tais parâmetros quando comparado parte frontal e traseira do veículo. Além disso, é dito que o amortecimento (perda de energia) ocasionado pelo efeito dos pneus é muito menor que do efeito provocado pelo sistema de suspensão e, portanto, será desconsiderado.

De tal modo, simplificando as massas dos pneus do veículo resulta:

$$m_1 = m_2 = m_f \quad (1)$$

$$m_3 = m_4 = m_r \quad (2)$$

Também considerado na modelagem, o efeito da barra anti-rolagem, adotado somente na parte dianteira por Jazar (2008), levando ao seguinte equacionamento regente do momento torsor no problema:

$$M_R = -k_R \left(\varphi - \frac{x_1 - x_2}{d_f} \right) \quad (3)$$

$$k_R = k_{Rf} \quad (4)$$

De modo que d_f representa a largura do veículo (distância entre duas rodas do mesmo eixo). Expressando em termos dos parâmetros da simulação posteriormente retratados na tabela 1 e presente no diagrama de corpo livre temos:

$$d_f = b_1 + b_2 \quad (5)$$

Baseado nos elementos apresentados e de modo a calcular as respostas de deslocamento do sistema, utiliza-se da função de transferência do sistema, compreendendo os 7 GDL:

$$[m]\ddot{x} + [c]\dot{x} + [k]x = F \quad (6)$$

Por sua vez, composta pelas matrizes 7×7 de $[m]$ (massa), $[c]$ (amortecimento) e $[k]$ (rigidez), multiplicadas por seus respectivos fatores \ddot{x} (aceleração), \dot{x} (velocidade) e x (deslocamento), no qual x é a saída a se determinar, englobando u (saída de deslocamento vertical do CG) e u_i (saída de deslocamento vertical de cada roda), também considerando F (entrada de deslocamento) no sistema, que posteriormente aliadas à rigidez dos pneus se converte em força. Os elementos não explícitos das matrizes de amortecimento e rigidez encontram-se, respectivamente, nos apêndices D e E. Portanto compreendidas a seguir:

$$[x] = \begin{bmatrix} u \\ \varphi \\ \theta \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (7)$$

$$[m] = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I_x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_y & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_f & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_f & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & m_r & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & m_r \end{bmatrix} \quad (8)$$

$$[c] = \begin{bmatrix} c_{11} & c_{12} & c_{13} & -c_f & -c_f & -c_r & -c_r \\ c_{12} & c_{22} & c_{23} & -b_1 c_f & b_2 c_f & b_1 c_r & -b_2 c_r \\ c_{13} & c_{23} & c_{33} & a_1 c_f & a_1 c_f & -a_2 c_r & -a_2 c_r \\ -c_f & -b_1 c_f & a_1 c_f & c_f & 0 & 0 & 0 \\ -c_f & b_2 c_f & a_1 c_f & 0 & m_f & 0 & 0 \\ -c_r & b_1 c_r & -a_2 c_r & 0 & 0 & c_r & 0 \\ -c_r & -b_2 c_r & -a_2 c_r & 0 & 0 & 0 & c_r \end{bmatrix} \quad (9)$$

$$[k] = \begin{bmatrix} k_{11} & k_{12} & k_{13} & -k_f & -k_f & -k_r & -k_r \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & b_1 k_r & -b_2 k_r \\ k_{31} & k_{32} & k_{33} & a_1 k_f & a_1 k_f & -a_2 k_r & -a_2 k_r \\ -k_f & k_{42} & a_1 k_f & k_{44} & -\frac{k_R}{d_f^2} & 0 & 0 \\ -k_f & k_{52} & a_1 k_f & -\frac{k_R}{d_f^2} & k_{55} & 0 & 0 \\ -k_r & b_1 k_r & -a_2 k_r & 0 & 0 & k_r + k_{t_r} & 0 \\ -k_r & -b_2 k_r & -a_2 k_r & 0 & 0 & 0 & k_r + k_{t_r} \end{bmatrix} \quad (10)$$

Por último, sumariza-se as forças de entrada do sistema, compreendidas na compressão das molas de cada roda. Desse modo, tem-se que não há forças sendo externamente aplicadas ao CG ou outras forças de rolagem/arfagem, tais quais poderiam ser desbalanços no peso do carro ou outros objetos em deslocamento externos à carcaça do veículo.

$$[F] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ y_1 k_{t_f} \\ y_2 k_{t_f} \\ y_3 k_{r_f} \\ y_4 k_{r_f} \end{bmatrix} \quad (11)$$

A solução do sistema mencionado é possível utilizando a Transformada de Laplace, trazendo a para o domínio da Laplace, a qual foi posteriormente simulada no domínio do tempo no Matlab, cuja rotina computacional encontra-se no apêndice A, então:

$$\mathcal{L}(x(t)) = X(s) \quad (12)$$

$$\mathcal{L}(\dot{x}(t)) = sX(s) \quad (13)$$

$$\mathcal{L}(\ddot{x}(t)) = s^2 X(s) \quad (14)$$

$$\mathcal{L}(F(t)) = F(s) \quad (15)$$

Assim, a equação do domínio s , de Laplace, fica:

$$[m]s^2 X(s) + [c]sX(s) + [k]X(s) = F(s) \quad (16)$$

Isolando a matriz $[X]$ de interesse:

$$X(s) = [[m]s^2 + [c]s + [k]]^{-1} F(s) \quad (17)$$

Desse modo, relacionando $H(s)$ com $F(s)$ para obter $X(s)$ nas equações 18 e 19.

$$H(s) = [[m]s^2 + [c]s + [k]]^{-1} \quad (18)$$

$$X(s) = [H]F(s) \quad (19)$$

Portanto, as variáveis de saída são representadas a partir da matriz 7×7 da seguinte forma:

$$\begin{bmatrix} U \\ \Phi \\ \Theta \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = [H] \begin{bmatrix} 0 \\ 0 \\ 0 \\ y_1 k_{tf} \\ y_2 k_{tf} \\ y_3 k_{rf} \\ y_4 k_{rf} \end{bmatrix} \quad (20)$$

Expandindo os termos, para segmentação das entradas e parâmetros do problema de modo a permitir a solução do equacionamento no domínio s e posteriormente a obtenção da resposta $x(t)$ no tempo, resulta na equação matricial 21. Com a aplicação das entradas y_i , é possível a obtenção da resposta no tempo via simulação *lsim* no Matlab.

$$\begin{bmatrix} U \\ \Phi \\ \Theta \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = [H] \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{tf} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{tf} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{rf} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & k_{rf} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \quad (21)$$

Para o dado cenário a ser simulado, busca-se identificar acelerações pertinentes nos eixos X, Y e Z não do CG do veículo nesse caso, mas sim de uma posição possível do dispositivo móvel. Nesse caso, faz-se necessário a utilização da dinâmica de corpo rígido para, a partir da arfagem e rolagem, determinar a posição relativa do ponto de interesse.

Primeiro, baseado nos dados de arfagem e rolagem obtidos da solução da equação 21, é possível calcular a velocidade angular ω_c da estrutura do veículo, dada vetorialmente por:

$$\omega_c = \begin{bmatrix} 0 \\ \frac{d\phi}{dt} \\ \frac{d\theta}{dt} \end{bmatrix} \quad (22)$$

Depois a aceleração angular:

$$\alpha = \begin{bmatrix} 0 \\ \frac{d\omega_{cx}}{dt} \\ \frac{d\omega_{cy}}{dt} \end{bmatrix} \quad (23)$$

Ambos os cálculos das equações 22 e 23 são feitos de maneira discreta, ou seja, computacionalmente são utilizados a diferença entre cada passo obtido de posição e velocidade, levando a aproximação da derivada para aquele ponto, tão precisa quanto os intervalos são pequenos.

Para a correta caracterização do ponto de interesse, define-se o vetor de posição que parte do CG, até o local desejado, nas 3 coordenadas de interesse, resultando que:

$$\vec{r} = \begin{bmatrix} X_p - X_{CG} \\ Y_p - Y_{CG} \\ Z_p - Z_{CG} \end{bmatrix} = \begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix} \quad (24)$$

Para o cálculo da velocidade e aceleração total ficam calculadas nas equações 25 e 26, respectivamente:

$$\vec{V} = V_{CG} + \vec{\omega}_c \times \vec{r} \quad (25)$$

Fazendo $\frac{dV}{dt}$, pela regra do produto vem:

$$\vec{A} = A_{CG} + \vec{\alpha} \times \vec{r} + \vec{\omega}_c \times (\vec{\omega}_c \times \vec{r}) \quad (26)$$

Da mesma forma, é possível obter a posição do ponto integrando a velocidade:

$$P = \int_{t_o}^{t_f} V dt \quad (27)$$

Com a condição de contorno que em $t(0)$, $P = P_{CG} + \vec{r} \Rightarrow C = P_{CG}$, portanto:

$$P = \int_{t_o}^{t_f} V dt + P_{CG} \quad (28)$$

Dada a natureza discreta da simulação. As integrações referidas nas equações 27 e 28 são feitas de maneira numérica, via *Matlab*. O resultado todavia deve ser numericamente congruente ao da análise contínua, compartilhando a justificativa matemática.

Assim, embasado pelo equacionamento proposto pela expressão 26, é possível calcular a aceleração do ponto de interesse para os três eixos de interesse. Adjunto a isso, se desejado a obtenção da posição do ponto em si em cada instante, o mesmo pode ser obtido a partir das equações 27 e 28.

3.2 MODELO DE REDE NEURAL

Para a análise dos dados gerados pela simulação, há de se escolher um modelo capaz de performar as operações matemáticas, através das camadas e funções de ativação adequadas, dadas as entradas e resultado final esperado.

Desse modo, utilizando-se também da revisão bibliográfica no que tange ao assunto, foi possível definir uma arquitetura visando desempenho e precisão. Visando solucionar o problema levantado, foi construído um modelo de rede neural que tem como entradas os dados de aceleração nos eixos X, Y e Z, e saída a classificação da existência do tipo de descontinuidade presente, variando entre **1** (buraco), **2** (lombada), **3** (sonorizador), **4** (tachão) ou **0** (não existência de descontinuidades), essa análise sempre dentro de um conjunto de dados históricos apresentados para inferência do modelo.

3.2.1 Arquitetura

Como pontuado na revisão bibliográfica, quanto as possíveis arquiteturas de rede neural capazes de se adaptar e analisar os dados produzidos, traz-se aqui o modelo utilizado.

O principal constituinte da arquitetura de modelos de Redes Neurais são as camadas que os compõem. De tal forma, a ordem que acontecem, somado funções de ativação e outras transformações entre camadas, permitem caracterizar o modelo, suas entradas necessárias e saídas.

No presente problema, visa-se, a partir dos dados de entrada de aceleração dos três eixos principais e ortonormais X, Y e Z, classificar o tipo de buraco presente no conjunto de dados fornecido, respeitando os intervalos de dados discutidos abaixo. Assim, é possível executar uma contagem e identificação no tempo de quantas descontinuidades foram detectadas, seu respectivo tipo e quando foram detectadas, isso tudo de uma forma automática e versátil, ainda possibilitando, dado um percurso conhecido, a caracterização do local em que foram percebidos, tudo isso utilizando nada mais que um *smartphone* de uso cotidiano.

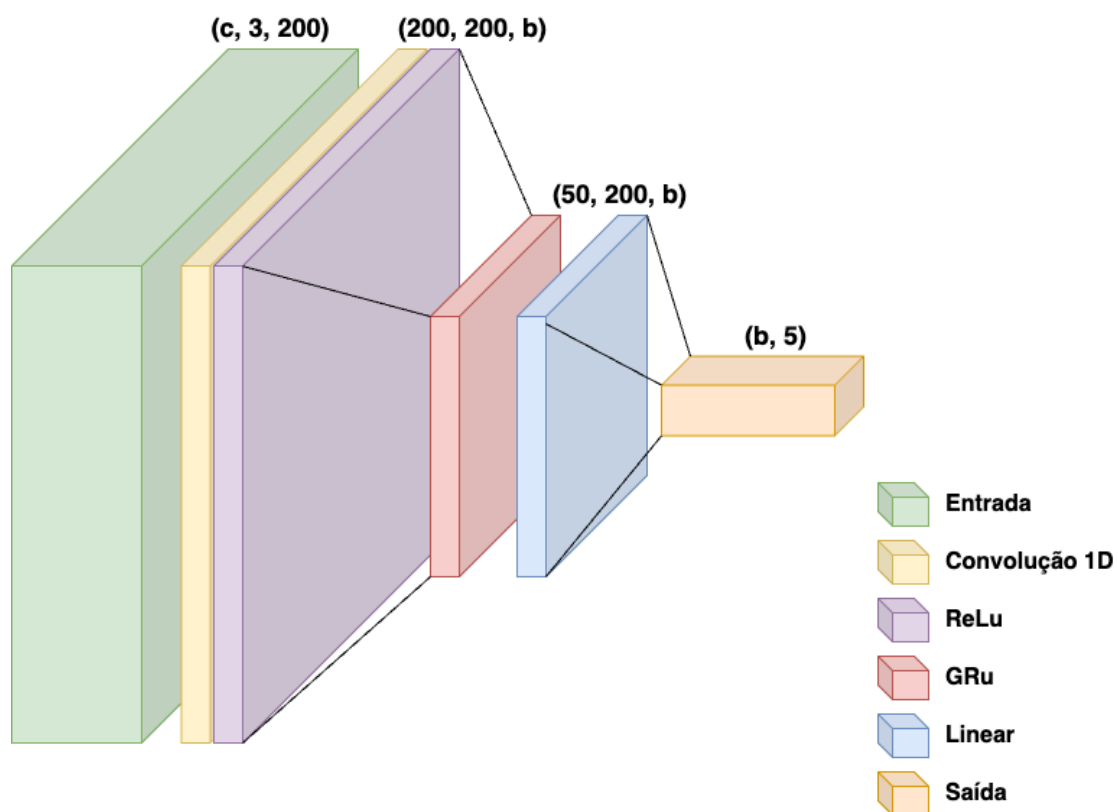
Para a correta inferência do modelo, é necessário que o intervalo de dados analisado seja suficientemente grande para abranger uma janela de descontinuidade inteira, ou seja, ao menos incluir totalmente os diferentes tipos de obstáculos, para que o treinamento possa levar as respostas esperadas.

Dado os parâmetros das descontinuidades simuladas, foi constatado que, como será

mostrado no detalhamento dos percursos abaixo, a maior descontinuidade possui comprimento de 1,7 m, assim, fixou-se como 2 metros o intervalo de análise, valor que se justifica também pela distância entre eixos veículo. Considerando que a resolução espacial dos dados produzidos pela simulação presente na rotina computacional foi parametrizada em 0,01 m. Junto a isso, o vetor de tempo da simulação é construído a partir da velocidade do veículo. Desse modo, resulta que, para obter 2 m de análise, são necessários 200 valores em cada vetor de aceleração.

Portanto, dado a característica dos dados de entrada, faz-se possível o uso de uma Rede Neural de Convolução (CNN) como primeira camada, reestruturando os dados para uma estrutura mais enxuta levando a uma rede menor e assim melhor extração dos padrões de interesse. Deste modo, como é possível notar na figura 3, há uma alteração na estrutura dos dados de entrada, que vinha no formato $(c, 3, 200)$, no qual c é uma variável, de valor arbitrário que pode ser qualquer número inteiro de 1 a $+\infty$, representando a quantidade de medições que se deseja fazer em sequência. Esse valor " c " corresponde diretamente a quantidade de inferências que serão produzidas na saída do modelo. Para os dados de simulação gerados neste trabalho especificamente, foi adotado $c = 791$. Após a camada de convolução, o formato é de $(200, 200, b)$, diminuindo significativamente a dimensão de comprimento do conjunto de dados, mas colocando na dimensão antes de 3, esse cenário é demonstrado visualmente pela figura 4 e está descrito na rotina em código *Python* nos apêndices B e C.

Figura 3 – Diagrama da arquitetura de Rede Neural do sistema

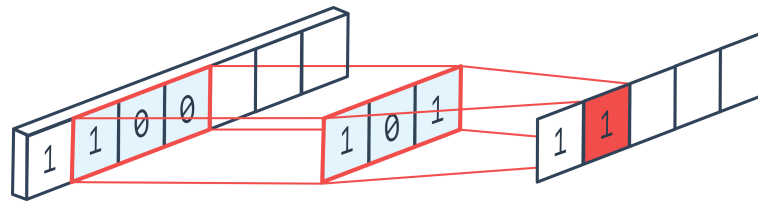


Fonte: Próprio Autor.

A dimensão " b " presente em várias das etapas é consequência do paralelismo presente no processamento das entradas, chamado de "*batch*" ou agrupamento, em português, sendo essa uma dimensão arbitrária, ajustada apenas por razões de desempenho de inferência e treinamento. No processo específico implementado, foi adotado $b = 64$, pelo modelo estar rodando em uma *CPU*, sendo assim, um alto paralelismo, por exemplo 256 ou mais, não surte efeitos positivos na velocidade de execução do modelo.

A camada de convolução deve ser seguida por uma camada de ativação, para que os padrões de interesse sejam notados e "ativados" consequentemente nos cenários esperados após o treinamento. Assim, após a primeira camada, sua saída é diretamente conectada a uma camada "ReLU" de ativação, que não altera estrutura, mas sim tem efeito sobre os dados presentes, somente permitindo a passagem dos não negativos.

Figura 4 – Visualização do efeito da camada de convolução 1D



Fonte: Ganesh (2019)

A partir desse cenário, uma série de opções são possíveis para melhor permitir desempenho do modelo, porém, como dito na bibliografia, foi possível notar que, em conjunto com uma camada de convolução para tratativa de sinais, uma camada "GRu" possui bom desempenho para a solução do problema. Nela, existe uma alteração de estrutura, desta vez saindo dos anteriores $(200, 200, b)$ para $(50, 200, b)$, fazendo nesse caso um *subsampling* dos dados (reduzindo sua complexidade), permitindo que somente os padrões relevantes sejam identificados.

Por fim, com o objetivo de fazer a classificação final dos dados dentro dos grupos pré-determinados, é aplicada uma camada linear e seu resultado tem formato $(b, 5)$, representando a classificação do tipo encontrado dentro de um conjunto de 200 leituras.

3.2.2 Geração de dados de treinamento para Rede Neural

Como parte importante para a execução dos passos relevantes no algoritmo de rede neural, é importante que haja larga existência de dados gerados, estes contendo variações relevantes para o aumento de versatilidade do sistema. Nesse caso, a estratégia abordada para permitir a geração de dados capazes de trazer mais adaptabilidade ao modelo foi:

1. Gerar dados com adição de um ruído aleatório, descrito no detalhamento dos percursos gerados

2. Gerar dados nos quais a posição da descontinuidade varia, assim não trazendo nenhum viés posicional
3. Gerar um grupo de dados balanceado quanto ao tipo de descontinuidade, para diminuir a possibilidade de viés quanto ao tipo detectado

Para o processo de treinamento e avaliação do modelo, todavia, ressalta-se a existência de 4º vetor, sendo 3 deles representando as acelerações já comentados, acrescido de um 4º vetor de "labels" de treinamento, ou seja, das classificações exatas de qual o tipo de obstáculo existente naquele conjunto de dados observado.

3.3 PARÂMETROS DE SIMULAÇÃO

Dentre as constantes necessárias para solução do equacionamento levantado, ficam destacadas na tabela 1. Para tal, foi utilizado como referência valores presentes em Jazar (2008), acrescida de valores experimentais obtidos na análise de suspensão de um VW Fusca, observado por Reginatto (2020) de modo a dar fidedignidade a valores equivalentes presente em veículos bi-eixo.

Tabela 1 – Constantes de massa, rigidez, amortecimento, comprimento e inércia

Grandeza	Magnitude	Unidade	Descrição
m	1.250	kg	Massa do veículo
m_f	20	kg	Massa das rodas dianteira
m_r	20	kg	Massa das rodas traseiras
a_1	1,1	m	Distância do CG ao eixo dianteiro
a_2	1,0	m	Distância do CG ao eixo traseiro
b_1	0,65	m	Distância do CG à roda esquerda
b_2	0,65	m	Distância do CG à roda direita
P_x	0,8	m	Distância em X do CG ao ponto de interesse
P_y	0	m	Distância em Y do CG ao ponto de interesse
P_z	0	m	Distância em Z do CG ao ponto de interesse
k_f	20.000	N/m	Rigidez das suspensões dianteiras
k_r	18.000	N/m	Rigidez das suspensões traseiras
k_{tf}	200.000	N/m	Rigidez dos pneus dianteiros
k_{tr}	200.000	N/m	Rigidez dos pneus traseiros
c_f	3.000	N.s/m	Amortecimento das suspensões dianteiras
c_r	3.000	N.s/m	Amortecimento das suspensões traseiras
k_R	30.000	N/m	Rigidez da barra anti-rolagem
I_x	4.100	kg.m ²	Momento de inércia no comprimento do veículo
I_y	5.500	kg.m ²	Momento de inércia na largura do veículo

Fonte: Adaptado de Jazar (2008), Reginatto (2020) e Próprio autor

Tais valores são tratados como variáveis na rotina computacional referida, aqui adotados para refletir um cenário cotidiano, porém adaptáveis para corresponder a outros veículos de 4 rodas, tais quais veículos de transporte de passageiros e carga.

3.3.1 Geração de ruído

Para dar fidedignidade aos dados gerados, para treinamento e simulação da resposta do veículo e do ponto de interesse, há de se produzir um "ruído" capaz de se assemelhar com o produzido por uma pista asfaltada.

A construção do ruído supra-mencionado se dá por meio de uma Distribuição Gaussiana que, dada uma faixa contínua de valores possíveis, se utiliza da média, do desvio padrão e do atrator mencionado, aqui denominado de γ , para criar as perturbações.

Assim, dado um intervalo de contínuo de valores I , contínuo em $[x, x + \delta] \forall \delta \in \mathbb{R}^+$, então I sendo:

$$I = [x, x + \delta]$$

Calcula-se a média e desvio padrão (σ) de I pelas equações 29 e 30 respectivamente:

$$\bar{I} = \frac{1}{n} \sum_{i=1}^n x_i \quad (29)$$

$$\sigma = \delta \cdot \gamma \quad (30)$$

Adicionalmente, acrescentando o fator randômico no cálculo, resulta no vetor de perturbações, do mesmo comprimento que o vetor de percurso, no qual R é um número randômico $\in [0, 1]$ e cada elemento é calculado pela equação 31:

$$p_i = \bar{I} + \sigma \cdot R \quad (31)$$

3.4 PERCURSO TEÓRICO DA SIMULAÇÃO E OBSTÁCULOS

Em objetivo a dar um contexto real de obstáculo e, portanto sua simulação, assim como de ter um objeto de estudo delimitado o suficiente para que fosse possível gerar dados abrangentes para o treinamento de um modelo de rede neural, foram escolhidos os tipos de obstáculo: lombada, sonorizador, buraco, tachão e asfalto normal. Assim, constroem-se os insumos para treinamento e validação do modelo a ser gerado.

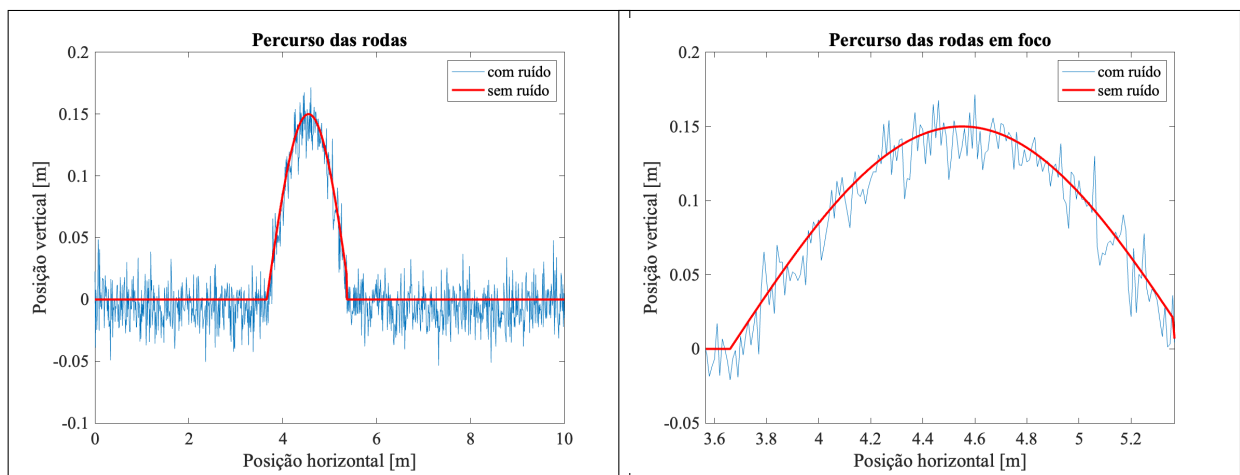
3.4.1 Lombada

Com tal cenário em mente, o primeiro obstáculo destacado é o de lombada, este porém com característica senoidal, ondulação única, altura positiva e magnitude mais expressiva que de outras discontinuidades. Ela se dá desse modo para que simule uma lombada média, como constatado por Santos (2018), com sua devida característica geométrica. A geometria que se

busca assemelhar é a de um formato de cunha, ou seja, tem início gradual e de característica semi-circular, assim como ao seu final e não quadrada como também poderia ser teoricamente compreendida.

Para sua representação matemática em termos de relevo no percurso, foi construída com altura de 15 cm e comprimento de 1,7 m, como mostrado na figura 5. Por simetria do problema, é combinado o percurso das rodas direitas e esquerdas.

Figura 5 – Percurso das rodas no obstáculo de lombada



Fonte: Próprio autor.

De maneira subsequente, para a transdução do percurso (em distância) para o deslocamento (tempo), foi considerada uma velocidade de 5,5 m/s (≈ 20 km/h), resultando no cenário da figura 10.

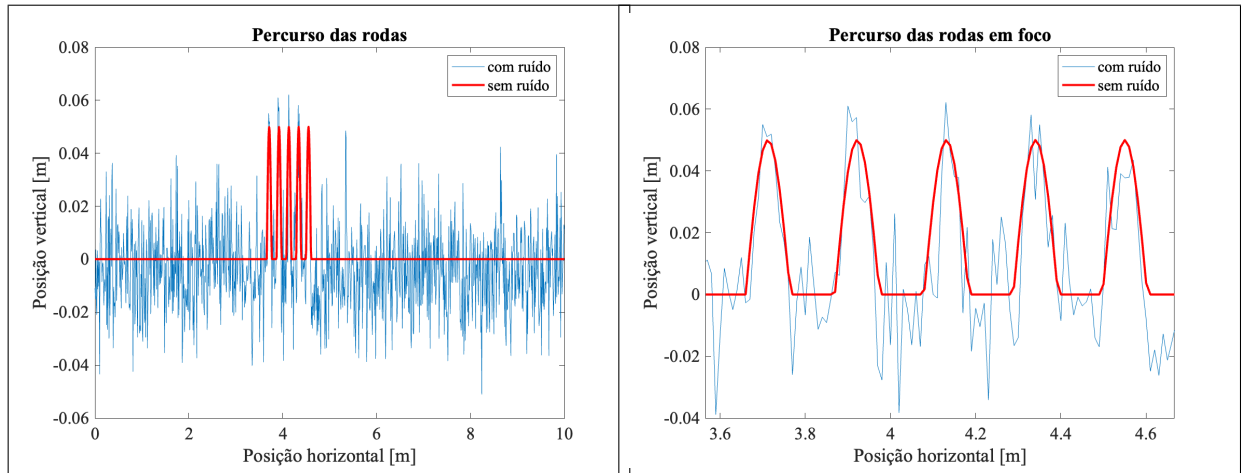
3.4.2 Sonorizador

Por sua vez, o sonorizador foi o segundo obstáculo considerado. Este possui alta frequência e comumente encontrado próximo a curvas em estradas ou na chegada em zonas urbanas, quando se deseja ocasionar aumento da atenção do motorista ou chamar a atenção para uma redução de velocidade eminente.

Para sua representação matemática em termos de relevo no percurso, foi construído com altura de 5 cm e comprimento de 2 m, como mostrado na figura 6. Por simetria do problema, é combinado o percurso das rodas direitas e esquerdas.

De maneira subsequente, para a transdução do percurso (em distância) para o deslocamento (tempo), foi considerada uma velocidade de 5,5 m/s (≈ 20 km/h), resultando no cenário da figura 12.

Figura 6 – Percurso das rodas no obstáculo de sonorizador



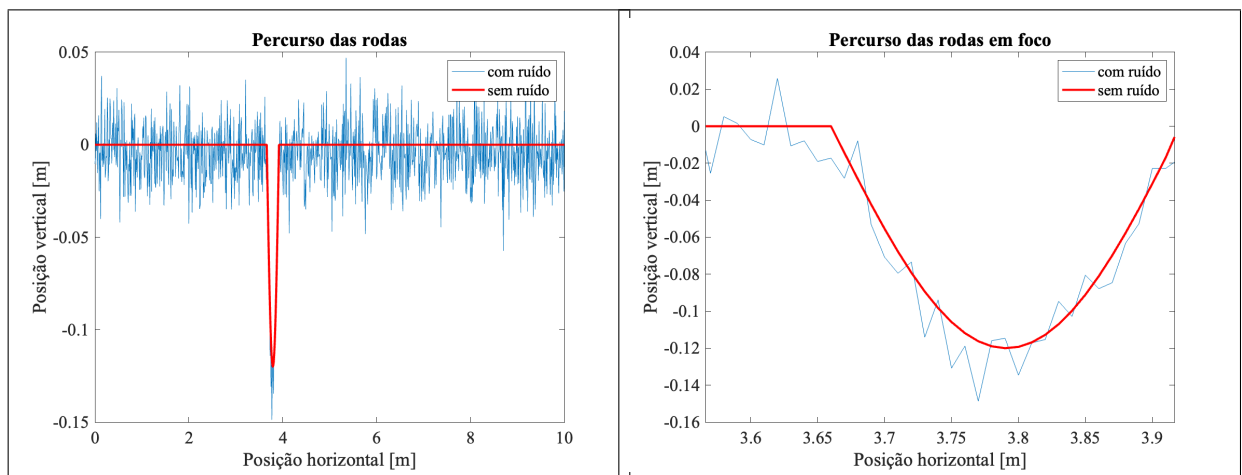
Fonte: Próprio autor.

3.4.3 Buraco

Como parte integral da análise, também resalta-se o buraco em si, generalizado aqui como uma profundidade negativa e com característica senoidal de meia-onda.

Para sua representação matemática em termos de relevo no percurso, foi construído com profundidade de 12 cm e comprimento de 25 cm, como mostrado na figura 7. Por simetria do problema, é combinado o percurso das rodas direitas e esquerdas.

Figura 7 – Percurso das rodas no obstáculo de buraco



Fonte: Próprio autor.

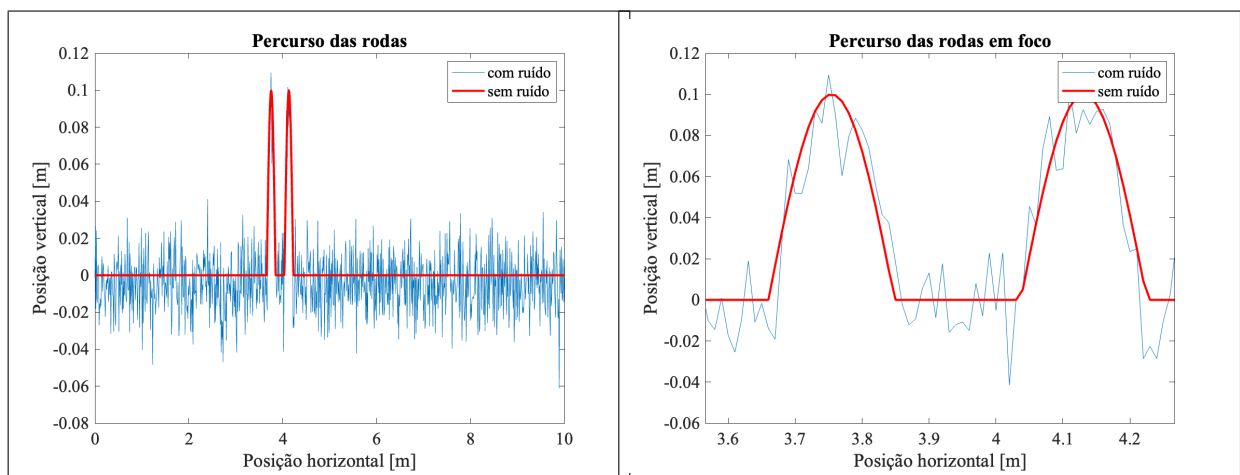
De maneira subsequente, para a transdução do percurso (em distância) para o deslocamento (tempo), foi considerada uma velocidade de 5,5 m/s (≈ 20 km/h), resultando no cenário da figura 14.

3.4.4 Tachão

Como parte integral da análise, também ressalta-se o tachão em si, generalizado aqui como uma profundidade negativa e com característica semi-circular.

Para sua representação matemática em termos de relevo no percurso, foi construído com profundidade de 30 cm e comprimento de 25 cm, como mostrado na figura 8. Por simetria do problema, é combinado o percurso das rodas direitas e esquerdas.

Figura 8 – Percurso das rodas no obstáculo de tachão



Fonte: Próprio autor.

De maneira subsequente, para a transdução do percurso (em distância) para o deslocamento (tempo), foi considerada uma velocidade de 5,5 m/s (≈ 20 km/h), resultando no cenário da figura 16.

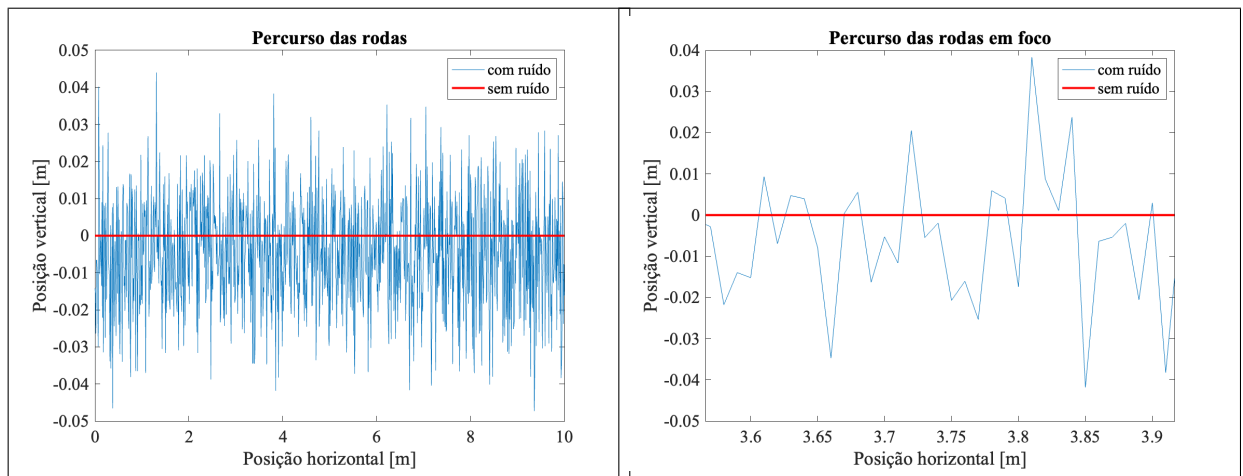
3.4.5 Estrada normal

Como referência base das simulações, para simular discontinuidades de baixa magnitude no terreno e normalmente distribuídas. Todavia, a rotina computacional responsável pela criação dessas discontinuidades possui um atrator parametrizável que traduz a qualidade da estrada para a magnitude da variabilidade do "ruído" gerado.

Assim, as imperfeições criadas pelo método descrito podem assumir valores de -2 cm até 1 cm. Tais valores de ruído estão presentes não só no cenário de asfalto normal, o qual não possui discontinuidades artificiais, mas também em todas as outros percursos, visto que sua base constitutiva é a mesma, o que implica em a característica do ruído ser a mesma, inclusive durante as discontinuidades em si, exceto no cenário da tachão. O resultado pode ser visto na figura 9.

De maneira subsequente, para a transdução do percurso (em distância) para o deslocamento (tempo), foi considerada uma velocidade de 5,5 m/s (≈ 20 km/h), resultando no cenário da figura 18. Por simetria do problema, é combinado o percurso das rodas direitas e esquerdas.

Figura 9 – Percurso das rodas na estrada normal



Fonte: Próprio autor.

É importante citar que a velocidade delimitada nos passos anteriores é parametrizável via rotina computacional e, portanto, pode ser ajustada para avaliar cenários relevantes em maiores ou menores velocidades, tais quais um buraco em uma estrada ou uma rua excessivamente esburacada.

4 RESULTADOS

De posse de todos os processos e percursos delimitados, visa-se analisar a resposta da rolagem, arfagem, deslocamento do CG e acelerações em X, Y e Z do ponto de interesse.

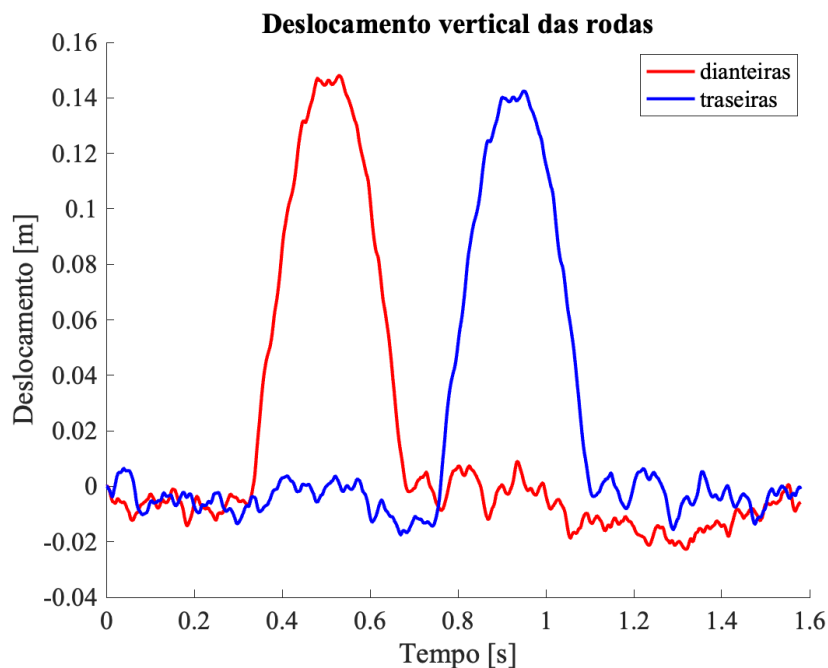
Adjunto aos resultados da simulação, busca-se também avaliar a classificação do tipo de descontinuidade do percurso, obtidos a partir do modelo de rede neural.

4.1 RESPOSTA CINEMÁTICA E CINÉTICA DO MODELO MATEMÁTICO

4.1.1 Lombada

Os resultados de Rolagem, Arfagem, deslocamento do CG e aceleração do ponto de interesse nos eixos X, Y e Z para o obstáculo de lombada estão presentes na figura 11, assim como o deslocamento das rodas demonstrado pela figura 10.

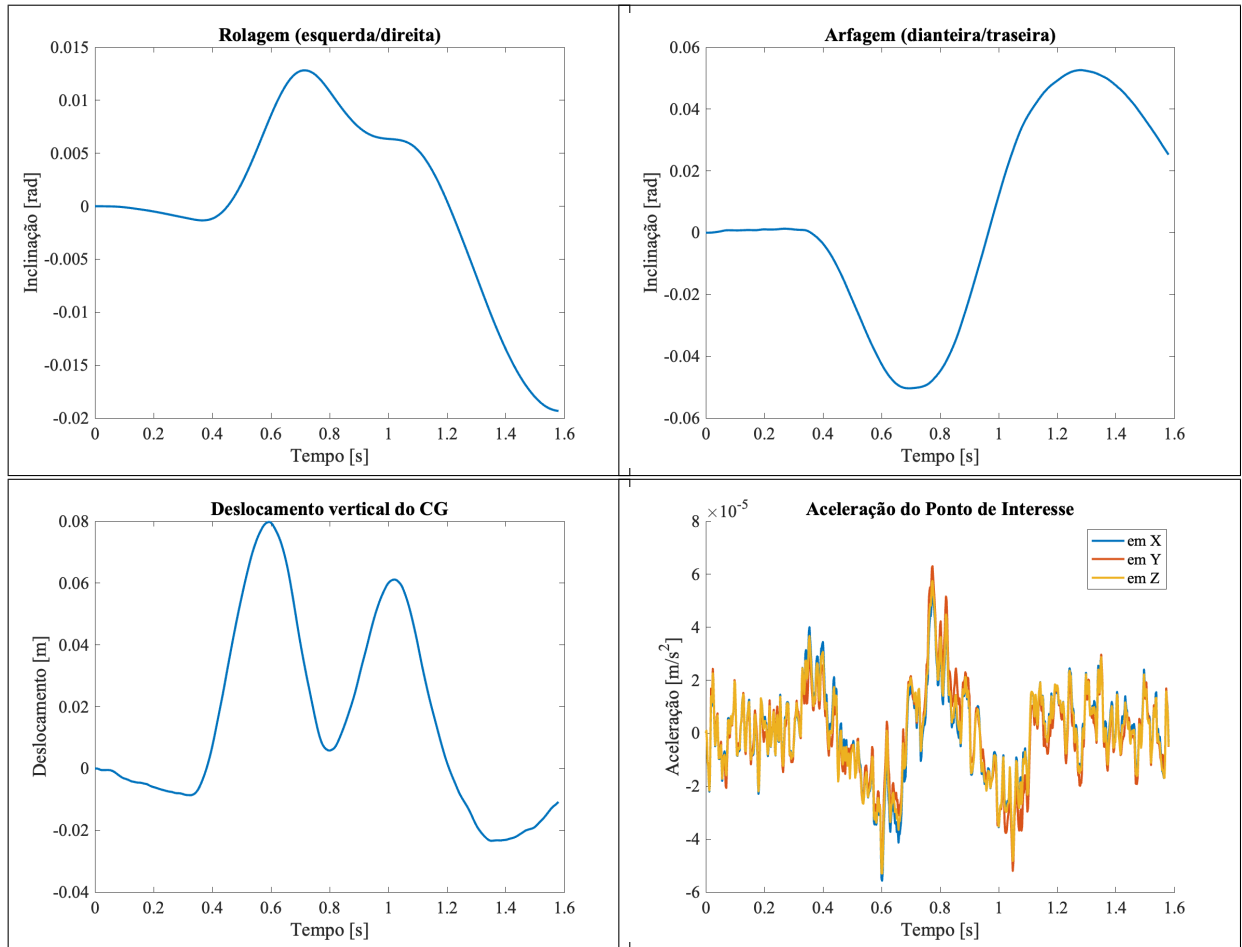
Figura 10 – Deslocamento das rodas no obstáculo de lombada



Fonte: Próprio autor.

No obstáculo de lombada, devido a perturbação espelhada entre o lado direito e esquerdo, a primeira análise a se fazer é a magnitude baixa da rolagem, na ordem de grandeza de 10^{-3} rad. Por outro lado, na arfagem, a inércia da roda, que tende a manter o movimento vertical para cima, leva a uma magnitude uma ordem de grandeza maior, na região de 10^{-2} rad. Neste caso, formam-se 2 "dentes" com característica similar e invertidos, dado a natureza simétrica entre

Figura 11 – Resultado da simulação para Obstáculo - Lombada



Fonte: Próprio autor.

parte dianteira e traseira do conjunto de rodas, porém com pequenas assimetria, como denotado pela tabela 1 de parâmetros.

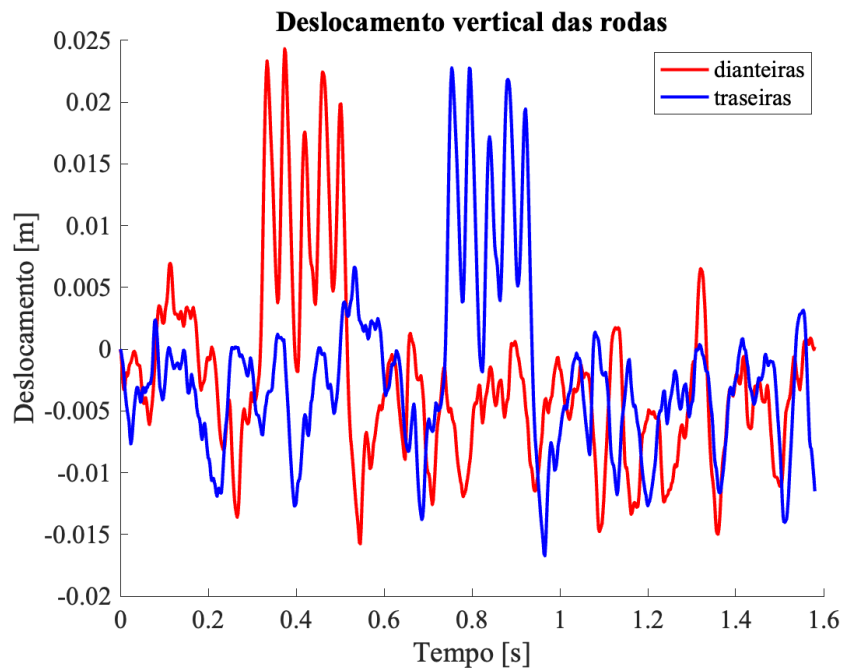
Também é possível notar que o deslocamento do CG forma os dois dentes, como esperado, também demonstrando o efeito "rebote" da lombada na suspensão do veículo, visto que assume valores negativos após ambos os impactos. Diretamente relacionado aos parâmetros anteriormente citados, e ao deslocamento do CG, é possível notar a aceleração do ponto de interesse nos 3 eixos, o qual está defasado do CG em X e Y, tendo efeito portanto não só do deslocamento do CG, mas também da arfagem e da rolagem.

4.1.2 Sonorizador

Os resultados de Rolagem, Arfagem, deslocamento do CG e aceleração do ponto de interesse nos eixos X, Y e Z para o obstáculo de Sonorizador estão presentes na figura 13, assim como o deslocamento das rodas demonstrado pela figura 12.

No obstáculo de Sonorizador, devido a perturbação espelhada entre o lado direito e

Figura 12 – Deslocamento das rodas no obstáculo de sonorizador



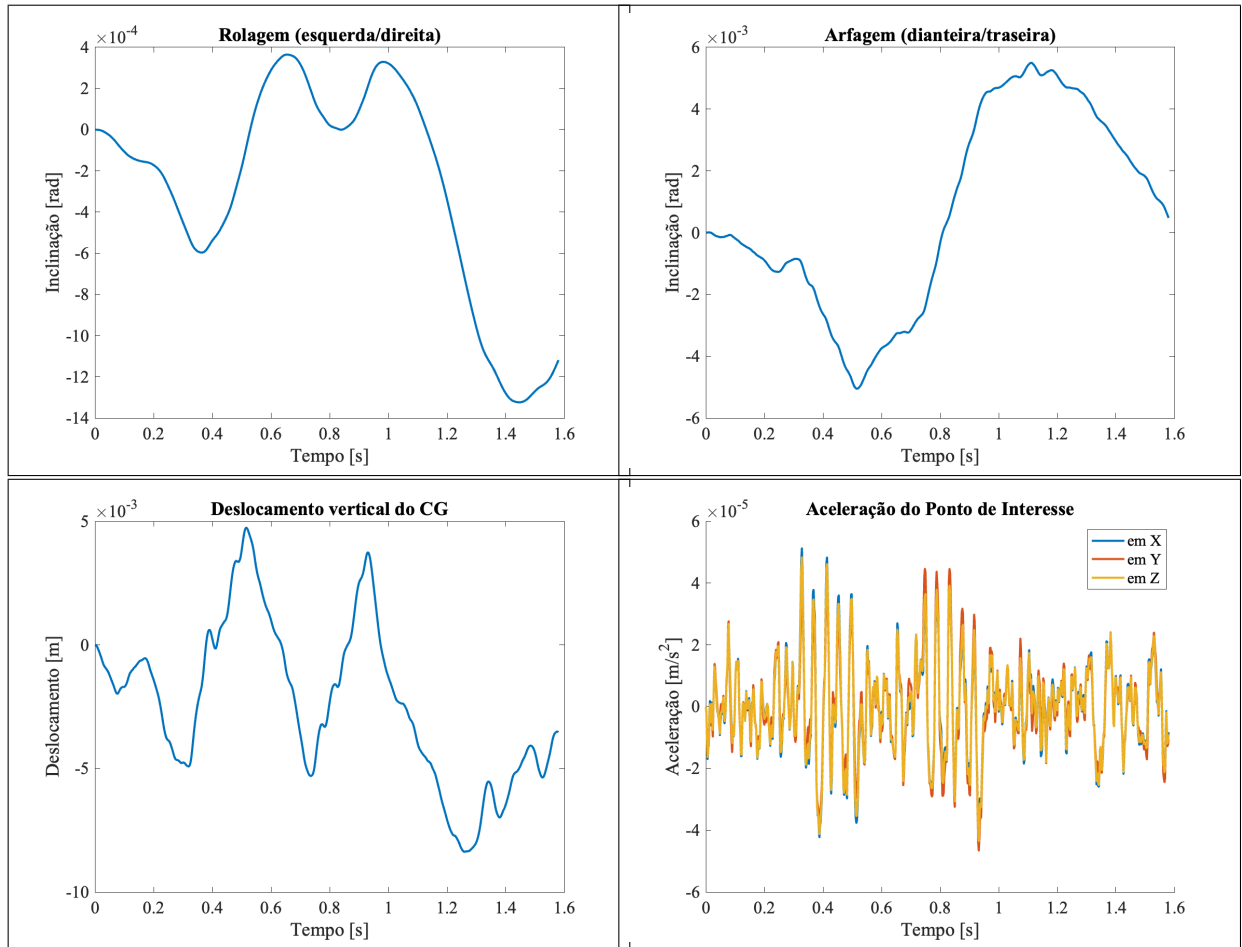
Fonte: Próprio autor.

esquerdo, a primeira análise a se fazer é a magnitude baixa da rolagem, na ordem de grandeza de 10^{-3} rad. Na arfagem, dado a menor magnitude do obstáculo e sim uma característica cíclica e de alta frequência, leva a uma magnitude também na região de 10^{-3} rad.

No caso do Sonorizador, como denotado na figura 12 é possível notar que sua magnitude quase se confunde com as próprias descontinuidades do percurso, especialmente quando considerado um pavimento de estado de conservação mais precário. Assim, a variação do CG tem o efeito de "dente" de baixa magnitude, referenciando a passagem do eixo dianteiro e traseiro, cronologicamente, mas que tem efeito quase tão intenso quanto o próprio pavimento no deslocamento absoluto observado em Z. Diretamente relacionado aos parâmetros anteriormente citados, e ao deslocamento do CG, é possível notar a aceleração do ponto de interesse nos 3 eixos, o qual está defasado do CG em X e Y, tendo efeito portanto não só do deslocamento do CG, mas também da arfagem e da rolagem. Neste, fica mais perceptível a frequência da ocorrência do sonorizador em si, visto que sofre os efeitos dos parâmetros de angulação citados, que tem impacto direto na magnitude da aceleração do mesmo.

Uma característica interessante de ressaltar no caso do sonorizador e proveniente de sua alta frequência é que no deslocamento do CG, essa "frequência" pouco é notada, o que leva a uma análise com relação a característica de filtragem do sistema, possuindo as características de um sistema de filtragem "passa baixa", dado que a característica de menor frequência é a que é possível ser notada após a "filtragem" feita pelas rodas e sistema de suspensão do veículo.

Figura 13 – Resultado da simulação para Obstáculo - Sonorizador



Fonte: Próprio autor.

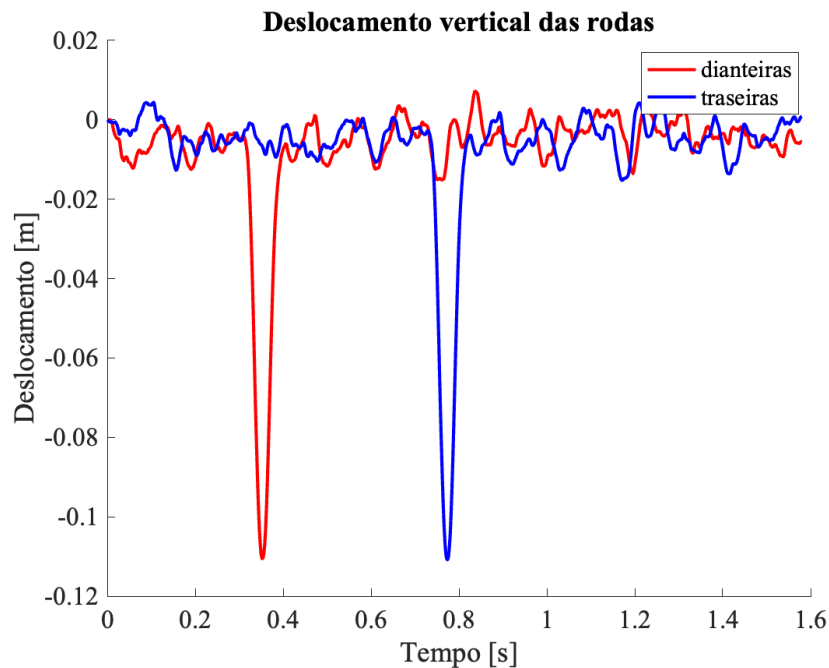
4.1.3 Buraco

Os resultados de Rolagem, Arfagem, deslocamento do CG e aceleração do ponto de interesse nos eixos X, Y e Z para o obstáculo de Tachão estão presentes na figura 15, assim como o deslocamento das rodas demonstrado pela figura 14.

No obstáculo de buraco, devido a perturbação espelhada entre o lado direito e esquerdo, a primeira análise a se fazer é a magnitude baixa da rolagem, na ordem de grandeza de 10^{-3} rad. Por outro lado, na arfagem, há o efeito contrário à inércia da roda e do pneu, o que leva a uma magnitude menor que no caso da lombada, assumindo valores dentro da ordem de grandeza de 10^{-3} rad.

Importante citar neste caso, comparativamente, que formam-se 2 "dentes" com característica similar e invertidos, muito similares ao da lombada, porém de sinais invertidos, visto que o buraco se da com valores de altura negativos. Assim, devido a essa diferença anatômica, é possível notar que a resposta entre ambos se dá de maneira diferente, visto que em um dos casos a força da gravidade acontece contra o movimento (lombada) e no buraco acontece a favor do

Figura 14 – Deslocamento das rodas no obstáculo de buraco



Fonte: Próprio autor.

movimento vertical natural do pavimento. Portanto, nota-se que a velocidade de resposta inicial quando se dá a presença do obstáculo é maior e pode ter efeitos mais intensos na suspensão do veículo, também notando que a inversão do sentido do movimento (troca de sinal da primeira derivada) se dá por efeito da mesma e não da força da gravidade.

Desse modo, a aceleração do ponto de interesse também produz padrões claros, produzindo valores de magnitude elevada (2x maior que o obstáculo de lombada com parâmetros geométricos similares) próximos a $3 \cdot 10^{-3} \text{ m/s}^2$. Esse maior valor pode ser justificado pela inércia da massa do veículo, a qual aumenta a intensidade para que haja alteração no sentido de rotação, que é imposto pela descontinuidade.

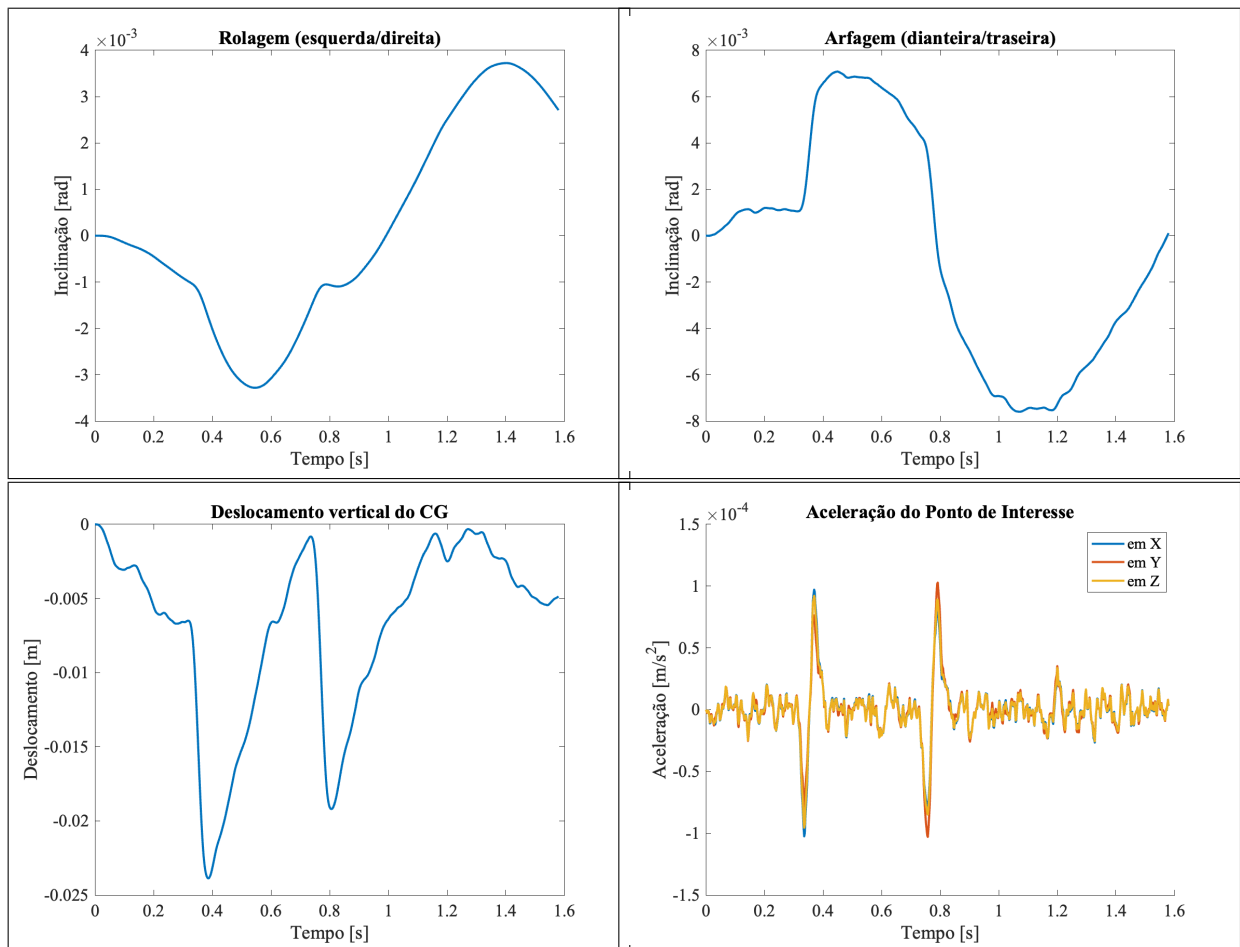
4.1.4 Tachão

Os resultados de Rolagem, Arfagem, deslocamento do CG e aceleração do ponto de interesse nos eixos X, Y e Z para o obstáculo de Buraco estão presentes na figura 17, assim como o deslocamento das rodas demonstrado pela figura 16.

No obstáculo de Tachão, devido a perturbação espelhada entre o lado direito e esquerdo, a primeira análise a se fazer é a magnitude baixa da rolagem, na ordem de grandeza de 10^{-3} rad. Na arfagem, também é possível notar a resposta com característica simétrica dentre as perturbações, de maior magnitude que a rolagem, mas ainda dentro da ordem de grandeza de 10^{-3} rad.

O obstáculo produz duas perturbações positivas no CG, dado característica abrupta de

Figura 15 – Resultado da simulação para Obstáculo - Buraco



Fonte: Próprio autor.

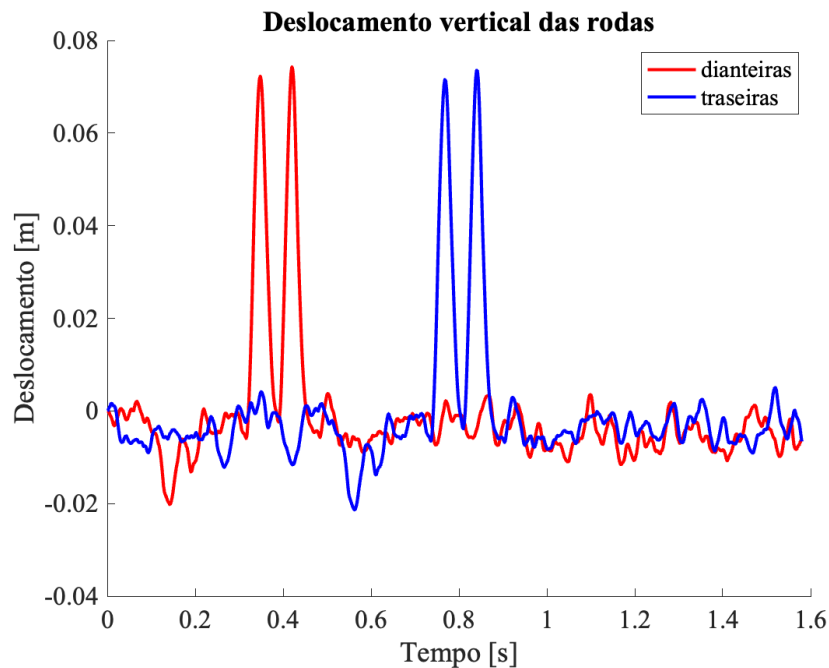
seu início e fim, culminando em duas respostas acentuadas com formato de "montanha", que não só assumem valores positivos, mas apresentam um rebote visto que logo depois assumem valores negativos, sendo consistente tanto na passagem da dianteira quando da traseira. Ambas as perturbações no CG se limitaram a 10 cm como visto na figura 8, acompanhando a magnitude do obstáculo em si e das rodas.

As acelerações do ponto de interesse refletem ambas as respostas observadas no deslocamento do CG, assumindo valores maiores do que das rugosidades do pavimento, mas ainda dentro de ordem de grandeza de 10^{-5} , de duas a 3 vezes menor do que perturbações mais acentuadas e a favor da gravidade, como no caso do buraco.

4.1.5 Estrada normal

Os resultados de Rolagem, Arfagem, deslocamento do CG e aceleração do ponto de interesse nos eixos X, Y e Z para o percurso sem obstáculos artificiais estão presentes na figura 19, assim como o deslocamento das rodas demonstrado pela figura 18.

Figura 16 – Deslocamento das rodas no obstáculo de tachão



Fonte: Próprio autor.

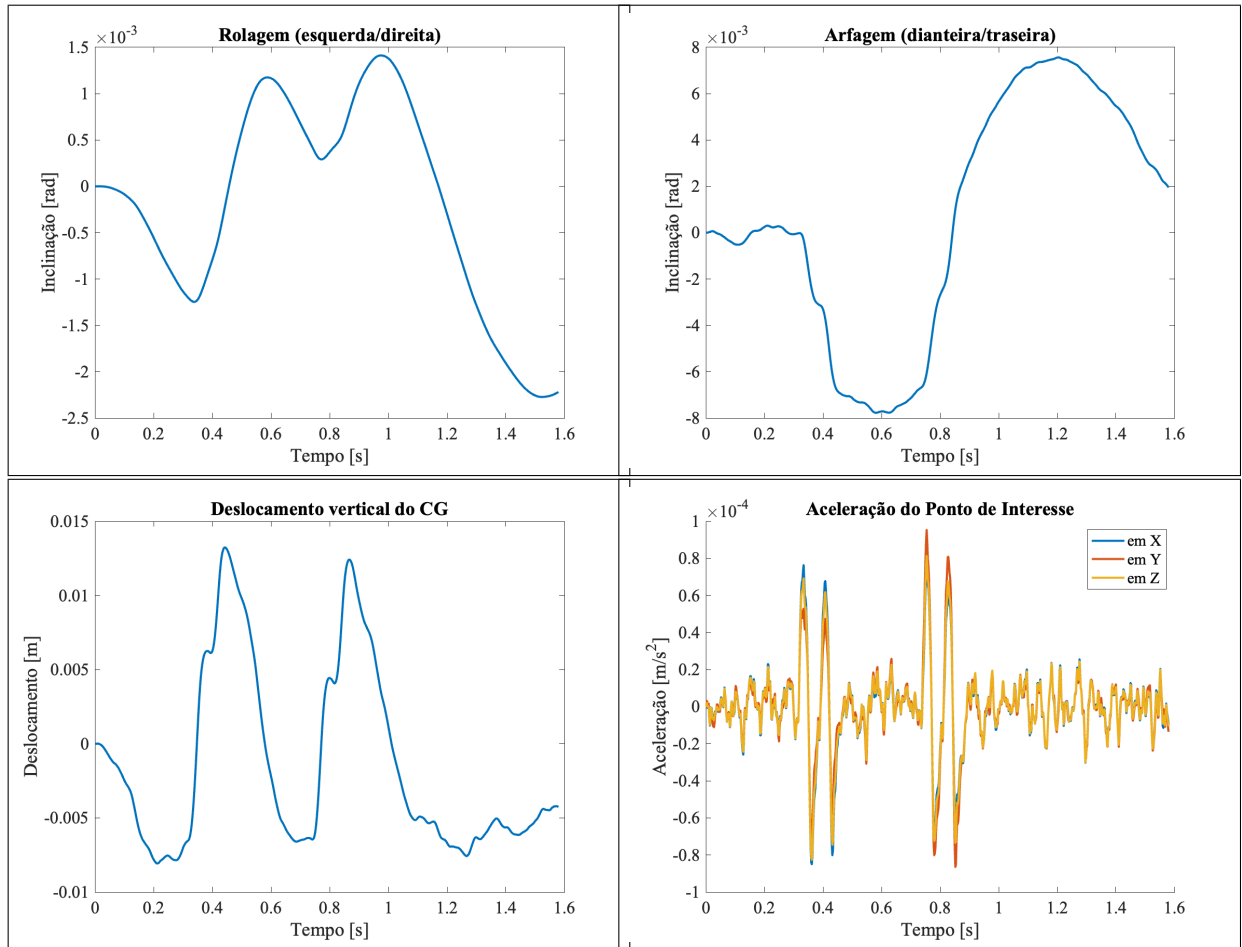
No cenário de estrada sem obstáculos artificiais, é possível estabelecer um cenário de controle, no qual não há padrões claros ou tendências. Nele, é possível notar que a magnitude dos valores de arfagem e rolagem estão na ordem de 10^{-4} rad, próximos inclusive um do outro já que não existem perturbações para elevar o valor da arfagem, como acontece nos outros cenários. Desse modo, o mesmo pode ser notado no caso do deslocamento do CG, isento de padrões, porém com magnitudes menores do que as observadas nas rodas (figura 18).

Essa magnitude mais baixa de movimentação do CG também se justifica de maneira análoga a comentada para o cenário do sonorizador, visto que o ruído causado pelas imperfeições do terreno também assumem característica similar e de alta frequência, assim sendo em sua maioria filtrados pela suspensão do veículo.

Tal qual os parâmetros mencionados nos parágrafos anteriores para este caso, a aceleração do ponto de interesse também assume valores sem tendência e distribuídos ao longo do percurso. Esta informação é especialmente relevante para o treinamento do modelo de rede neural, que deve ser capaz de não somente descartar rugosidades do solo durante sua inferência, mas também de saber quando não há obstáculos impostos ao sistema e trata-se de um ambiente "normal".

Assim, a aceleração do ponto de interesse não possui parâmetros identificáveis e assume magnitudes na ordem de 10^{-5} , com padrão de alta frequência de oscilação, se assemelhando com o cenário de rugosidade dos materiais, os quais vindo em escala microscópica possuem característica caótica e abrupta, mas quando comparados aos desvios de forma são meramente vistos. Nesse caso, buscou-se elucidar qual era esse valor de controle e o que essa analogia à

Figura 17 – Resultado da simulação para Obstáculo - Tachão



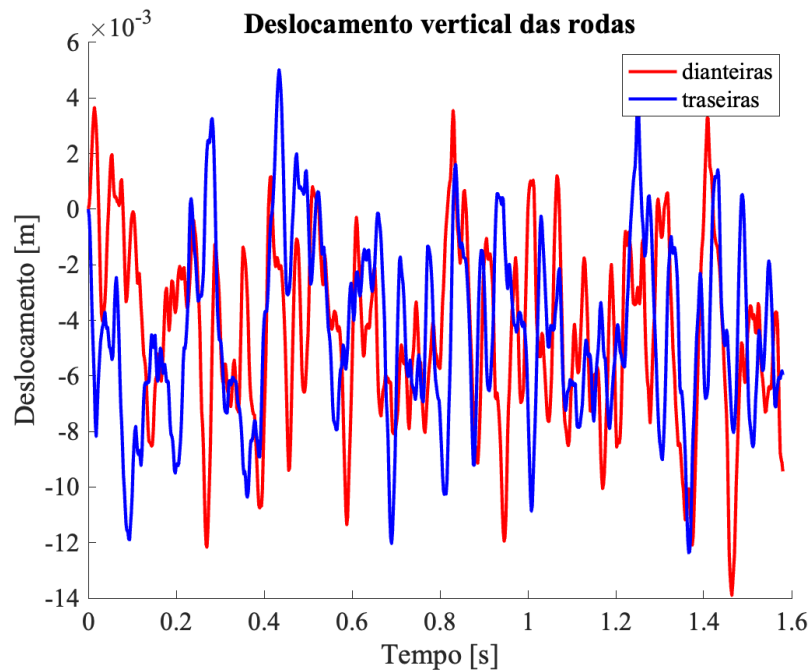
Fonte: Próprio autor.

rugosidade e desvios de forma representa nesse cenário de simulação do pavimento acrescidos dos obstáculos.

4.2 CLASSIFICAÇÃO DE OBSTÁCULOS POR REDE NEURAL

Como objetivo do trabalho e pelos métodos descritos anteriormente, também visa-se obter um modelo capaz de interpretar os dados de entrada, segmentados em "janelas" de 200 elementos a fim de classificá-los quanto ao tipo de discontinuidade e qual discontinuidade está presente. Tais "janelas" representam o intervalo de dados no qual será feita uma inferência única, ou seja, dado 200 passos de medição, que serão inseridos ao modelo, será inferido um tipo de discontinuidade dentro da classificação de 0 a 4 (inclusive) já explicitada. Na prática, isso representa que é necessário como entrada 3 vetores de dados com 200 elementos cada um, para que seja possível a execução de uma inferência. Para o treinamento, o cenário é análogo, mas com a adição de um 4º vetor, representando os resultados esperados, com base no qual será feita a propagação inversa dos valores, de modo a treinar o modelo para produzir os resultados

Figura 18 – Deslocamento das rodas no obstáculo de Estrada normal



Fonte: Próprio autor.

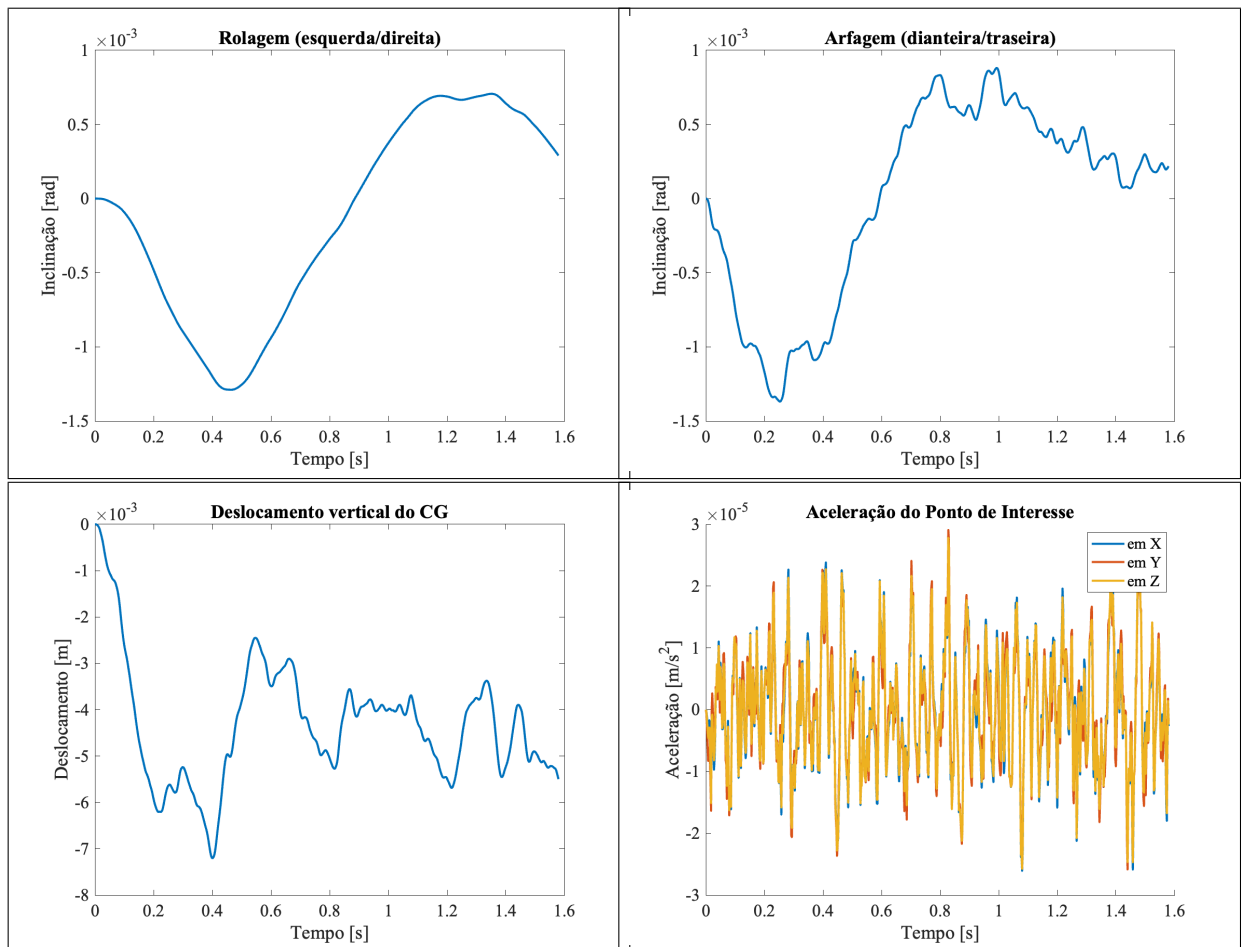
esperados. Assim, segmentando a análise da performance do modelo construído baseado em redes neurais, para cada tipo de descontinuidade são obtidos resultados descritos na seção que segue.

4.2.1 Lombada

O gráfico de dispersão referente ao obstáculo de Lombada na figura 20 demonstram os pontos inferidos pelo modelo, comparado ao valor de fato esperado. Os pontos em laranja demonstram os valores em que houve equivalência e os pontos em vermelho demonstram os pontos nos quais houve diferença entre o valor esperado e o que o modelo inferiu a partir dos dados de entrada. Para o desempenho da inferência, foram utilizados dados equivalente a duas simulações do tipo específico, principalmente para facilitar a visualização e por conter parcela amostral suficiente em termos de variabilidade dos dados.

Como é possível observar, dentro dos dados amostrais foram notados 25 pontos de invalidez, representando 2,81% dos dados amostrais, presentes nos pares "2-0", "0-2" e "0-3", ou seja, o par buraco-estrada normal e buraco-sonorizador (que também produziu desvios em outros cenários). Esse desvio pode ser explicado por janelas em que haja certa ambiguidade quanto a existência ou não de obstáculo dentro da janela observada. Ou seja, inícios de obstáculo ou muito ao final, onde os dados não sejam conclusivos suficientes. É possível observar no gráfico que a existência dos pontos sustenta a tese levantada, se concentrando primordialmente no final do primeiro intervalo e do segundo. Também foram notados alguns pontos no interstício, em que

Figura 19 – Resultado da simulação para estrada normal



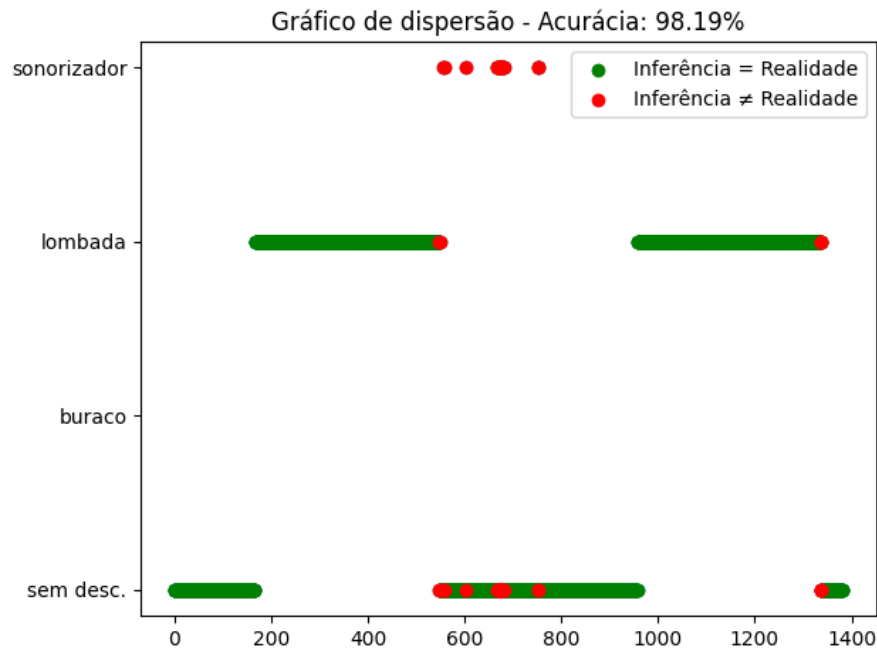
Fonte: Próprio autor.

foi inferido o tipo "3" de obstáculo, sonorizador, não relacionados a capacidade de inferência do modelo com relação ao obstáculo de lombada.

É possível visualizar, adicionalmente, de maneira quantitativa os pontos de concordância e discordância das previsões do modelo e dados reais a partir da Matriz de Dispersão da figura 21, a qual mostra os valores esperados confrontados aos valores de fato medidos, sendo esperado que apenas os índices da diagonal assumam valores diferentes de 0.

Nesse caso, foi notado que 599 dos 621 pontos em que se encontrava um buraco foram inferidos corretamente, sendo o 599 a inferência no par "buraco-buraco", e os 22 pontos de inferência incorreta se encontram nos elementos da primeira linha da matriz, mas em outras colunas que não a primeira, portanto $i = 1, j \neq 1$, ou seja, qualquer elemento na primeira linha que esteja fora da diagonal principal. Também foi notado que 759 dos 762 pontos em que não havia descontinuidades foram inferidos corretamente, tendo o par errôneo "sem descontinuidade-buraco", onde foi inferido "buraco", mas na verdade era "sem descontinuidade", na linha 3 o responsável pelos 3 erros nesse cenário.

Figura 20 – Gráfico de Dispersão da inferência para o obstáculo de Lombada



Fonte: Próprio Autor.

Nas métricas relevantes à performance do modelo, também acresce-se a visualização da acurácia obtida, que neste caso está preocupada em avaliar quantas das inferências estão de acordo com o dado real. Assim, coloca-se a acurácia como "dentre todas as classificações de classe Positivo que o modelo fez, quantas estão corretas".

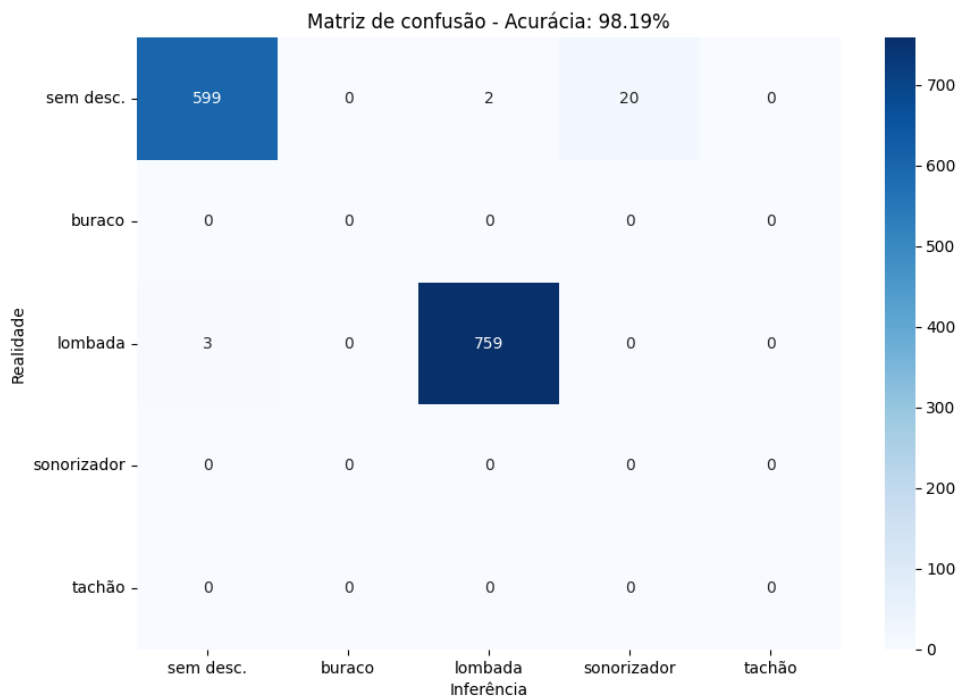
Para este caso específico, foi notada uma acurácia de 98,19%, na qual, $P = \frac{VP+VN}{VP+FP+VN+FN} = 0,9819$.

4.2.2 Sonorizador

O gráfico de dispersão referente ao obstáculo de Sonorizador na figura 22 demonstram os pontos inferidos pelo modelo, comparado ao valor de fato esperado. Os pontos em laranja demonstram os valores em que houve equivalência e os pontos em vermelho demonstram os pontos em que houve diferença entre o valor esperado e o que o modelo inferiu a partir dos dados de entrada. Para o desempenho da inferência, foram utilizados dados equivalente a duas do tipo específico simulações, principalmente para facilitar a visualização e por conter parcela amostral suficiente em termos de variabilidade dos dados.

Como é possível observar, dentro dos dados amostrais foram notados 149 pontos de invalidez, o que representa 11,77% dos dados amostrais, significativamente acima do valor para o caso anterior. Acrescido do cenário de início e fim dos intervalos, também é possível notar outros pontos de desvio das inferências, em geral no meio dos intervalos. No caso do sonorizador especificamente, os períodos intermediários do intervalo podem ser confundidos como estrada

Figura 21 – Matriz de Confusão da inferência para o obstáculo de Lombada



Fonte: Próprio Autor.

normal, tanto devido à magnitude dos valores, quando a forma de arranjo, de frequência mais alta. Ademais, houveram algumas inferências errôneas, que foram taxadas como o cenário "4", tachão, em cenários intersticiais, possivelmente justificáveis pelos valores e tendência assumida entre os obstáculos no caso do tachão.

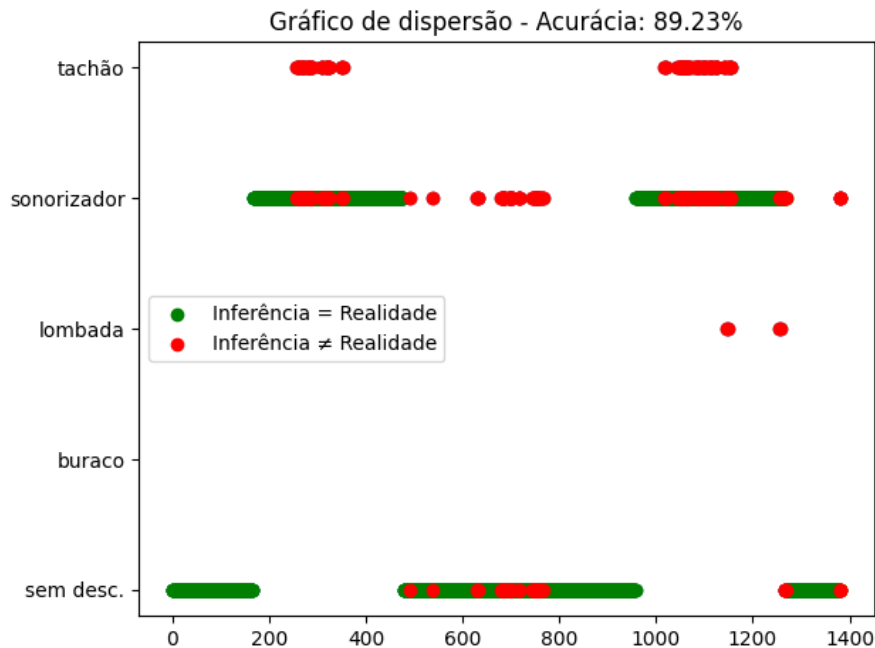
É possível visualizar, adicionalmente, de maneira quantitativa os pontos de concordância e discordância das previsões do modelo e dados reais a partir da Matriz de Dispersão da figura 23, a qual mostra os valores esperados confrontados aos valores de fato medidos, sendo esperado que apenas os índices da diagonal assumam valores diferentes de 0.

4.2.3 Buraco

O gráfico de dispersão referente ao obstáculo de Buraco na figura 24 demonstram os pontos inferidos pelo modelo, comparado ao valor de fato esperado. Os pontos em laranja demonstram os valores em que houve equivalência e os pontos em vermelho demonstram os pontos onde houve diferença entre o valor esperado e o que o modelo inferiu a partir dos dados de entrada. Para o desempenho da inferência, foram utilizados dados equivalente a duas simulações do tipo específico, principalmente para facilitar a visualização e por conter parcela amostral suficiente em termos de variabilidade dos dados.

Como é possível observar, dentro dos dados amostrais foram notados 93 pontos de

Figura 22 – Gráfico de Dispersão da inferência para o obstáculo de Sonorizador



Fonte: Próprio Autor.

invalidez, o que representa 6,72% dos dados amostrais. Esse desvio pode ser explicado por janelas em que haja certa ambiguidade quanto a existência ou não de obstáculo dentro da janela observada. Ou seja, inícios de obstáculo ou muito ao final, onde os dados não sejam conclusivos suficientes. Também é notado a maior concentração de erros de medição quando confrontados com o cenário de tachão, podendo-se justificar pelo perfil mais rápido do obstáculo, fazendo com que alguns intervalos contenham apenas o ruído, podendo dar ambiguidade na leitura.

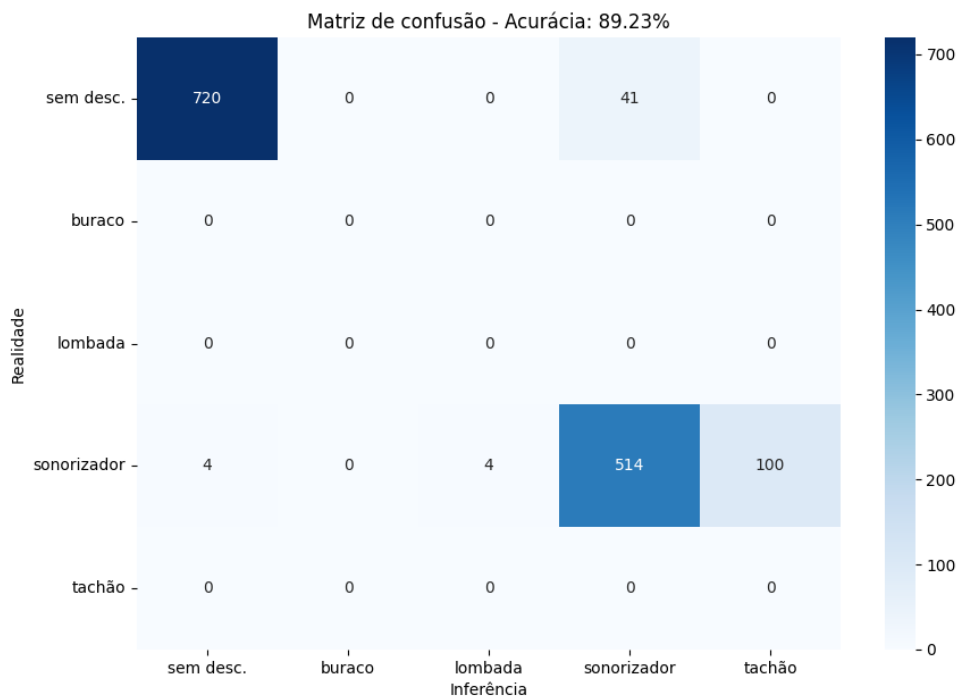
É possível visualizar, adicionalmente, de maneira quantitativa os pontos de concordância e discordância das previsões do modelo e dados reais a partir da Matriz de Dispersão da figura 25, a qual mostra os valores esperados confrontados aos valores de fato medidos, sendo esperado que apenas os índices da diagonal assumam valores diferentes de 0.

4.2.4 Tachão

O gráfico de dispersão referente ao obstáculo de Tachão na figura 26 demonstram os pontos inferidos pelo modelo, comparado ao valor de fato esperado. Os pontos em laranja demonstram os valores em que houve equivalência e os pontos em vermelho demonstram os pontos onde houve diferença entre o valor esperado e o que o modelo inferiu a partir dos dados de entrada. Para o desempenho da inferência, foram utilizados dados equivalente a duas simulações do tipo específico, principalmente para facilitar a visualização e por conter parcela amostral suficiente em termos de variabilidade dos dados.

Como é possível observar, dentro dos dados amostrais foram notados 27 pontos de

Figura 23 – Matriz de Confusão da inferência para o obstáculo de Sonorizador



Fonte: Próprio Autor.

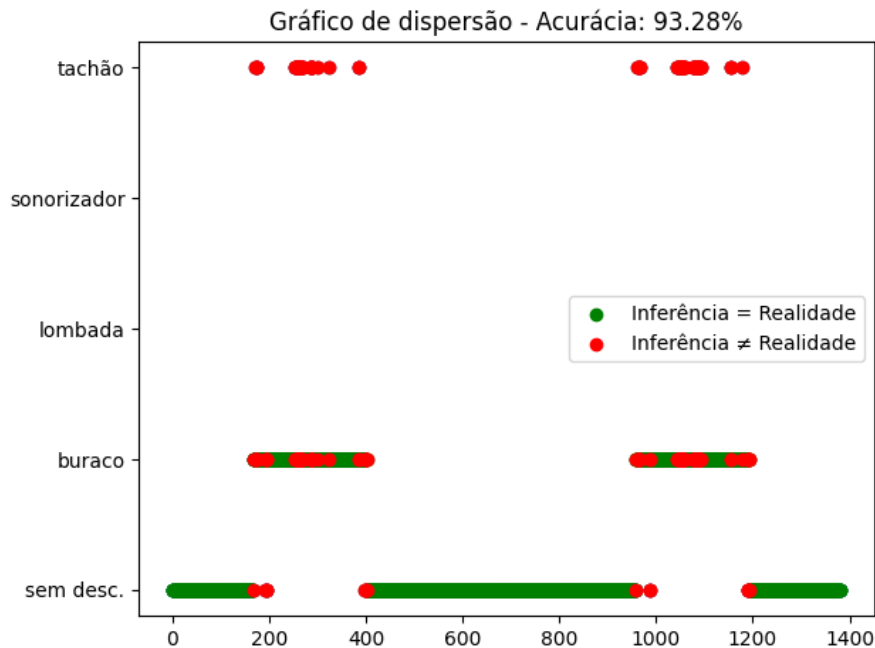
invalidez, o que representa 1,95% dos dados amostrais. Esse desvio pode ser explicado por janelas em que haja certa ambiguidade quanto a existência ou não de obstáculo dentro da janela observada. Ou seja, inícios de obstáculo ou muito ao final, onde os dados não sejam conclusivos suficientes. Não foram observadas discrepâncias entre o cenário esperado e inferências em períodos intersticiais. É possível observar no gráfico de dispersão que a existência dos pontos sustenta a tese levantada.

É possível visualizar, adicionalmente, de maneira quantitativa os pontos de concordância e discordância das previsões do modelo e dados reais a partir da Matriz de Dispersão da figura 27, a qual mostra os valores esperados confrontados aos valores de fato medidos, sendo esperado que apenas os índices da diagonal assumam valores diferentes de 0.

4.2.5 Estrada normal

O gráfico de dispersão referente ao obstáculo de Estrada normal na figura 28 demonstram os pontos inferidos pelo modelo, comparado ao valor de fato esperado. Os pontos em laranja demonstram os valores em que houve equivalência e os pontos em vermelho demonstram os pontos onde houve diferença entre o valor esperado e o que o modelo inferiu a partir dos dados de entrada. No caso de Estrada Normal, a validação é para garantir que cenários de variabilidade esperados do próprio terreno sejam contemplados como tal, ou seja, não tragam

Figura 24 – Gráfico de Dispersão da inferência para o obstáculo de Buraco



Fonte: Próprio Autor.

imprecisões para a medição de outros tipos ou do interstício entre a existência de obstáculos. Para o desempenho da inferência, foram utilizados dados equivalente a duas do tipo específico simulações, principalmente para facilitar a visualização e por conter parcela amostral suficiente em termos de variabilidade dos dados.

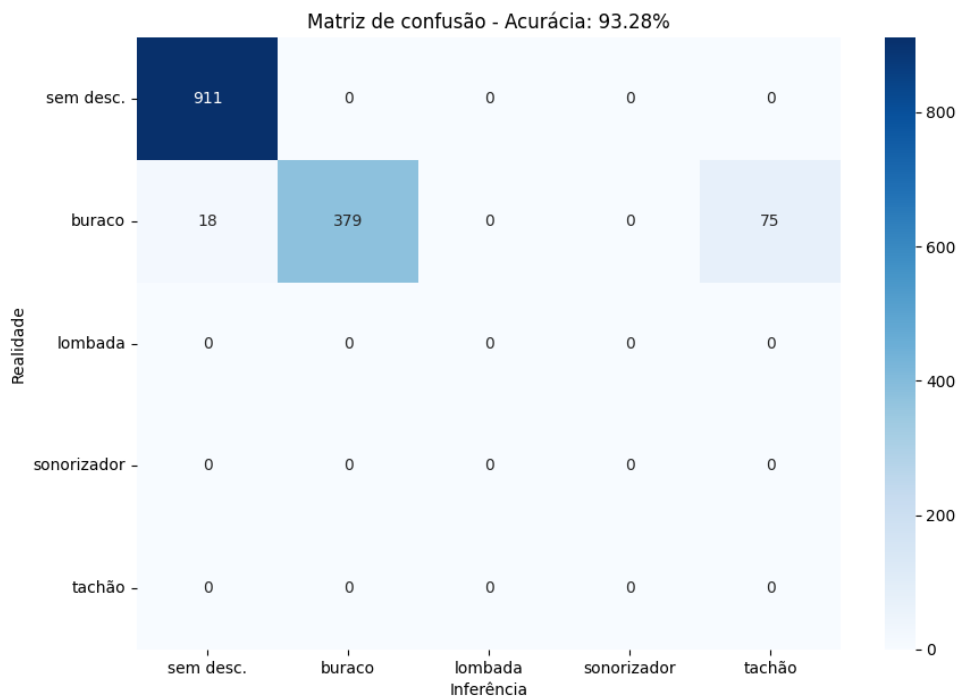
Como é possível observar, dentro dos dados amostrais foram notados 169 pontos de invalidez, o que representa 12,22% dos dados amostrais. Nesse caso, apenas de ruído é constituído as entradas, como possível ver pela figura 18. Foi possível notar um não padrão da distribuição dos dados, somente que a maior concentração de falhas na inferência se dá com o sonorizador, devido aos aspectos de forma, magnitude e frequência dos sinais.

É possível visualizar, adicionalmente, de maneira quantitativa os pontos de concordância e discordância das predições do modelo e dados reais a partir da Matriz de Dispersão da figura 29, a qual mostra os valores esperados confrontados aos valores de fato medidos, sendo esperado que apenas os índices da diagonal assumam valores diferentes de 0.

4.2.6 Desempenho Geral do Modelo

Uma das grandes motivações da solução do problema de classificação de tipos de descontinuidade e detecção utilizando redes neurais artificiais, assim como já fundamentado na revisão da bibliografia, é dar flexibilidade ao processo e capacidade iterativa de melhoria e incremento das capacidades. Esse incremento, tanto relacionado às variáveis que fazem parte dos dados já utilizados para treinamento, mas também e, principalmente, para variáveis ainda

Figura 25 – Matriz de Confusão da inferência para o obstáculo de Buraco



Fonte: Próprio Autor.

não exploradas, sem que seja necessária a construção e treinamento de um novo modelo a partir do início, mas sim que sejam aproveitadas as competências já desenvolvidas e apenas agregado novas bases de conhecimento, muito como acontece com o aprendizado humano.

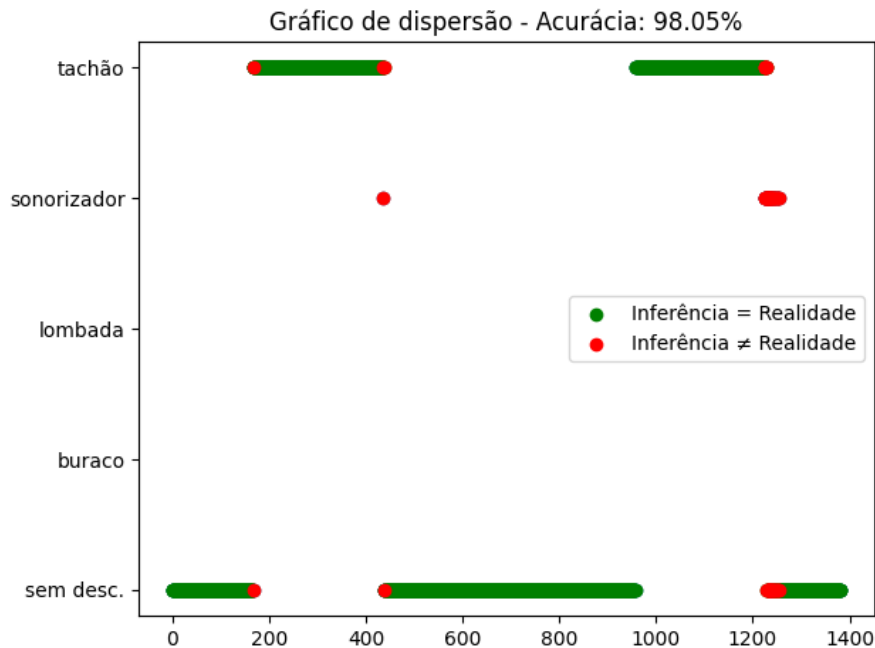
Dito isso, é importante analisar não só as competências individuais de classificação de cada tipo de descontinuidade, mas também o seu desempenho como generalista, variando dentre todos os tipos. Visando esse objetivo e buscando maior robustez na análise, foi levantada uma base de dados de simulação de 40 percursos diferentes para cada tipo de descontinuidade de interesse, resultando em centenas de milhares de leituras individuais, o que, dado a presença do ruído aleatório, pode simular a característica aleatória de medição do mundo real.

A partir desse processo, pôde-se observar que o modelo atingiu acurácia de 98,77%, para a inferência de dados que não estavam presentes no conjunto de teste nem validação, portanto dados novos, ainda não vistos pelo modelo, dados estes que não seguem uma ordem específica e respeitam a mesma divisão de "janelas" mencionadas em outras etapas.

Para exemplificar tal acurácia, utiliza-se de uma matriz de confusão na figura 30, a qual demonstra a distribuição de acertos dentro do universo de dados.

Na matriz de confusão, o resultado 100% correto, ou seja, produzido por um modelo com 100% de acurácia levaria a diagonal principal da matriz a possuir somente valores iguais a 1 e todos os índices a possuir valores iguais a 0. Portanto, em congruência ao resultado explicitado

Figura 26 – Gráfico de Dispersão da inferência para o obstáculo de Tachão



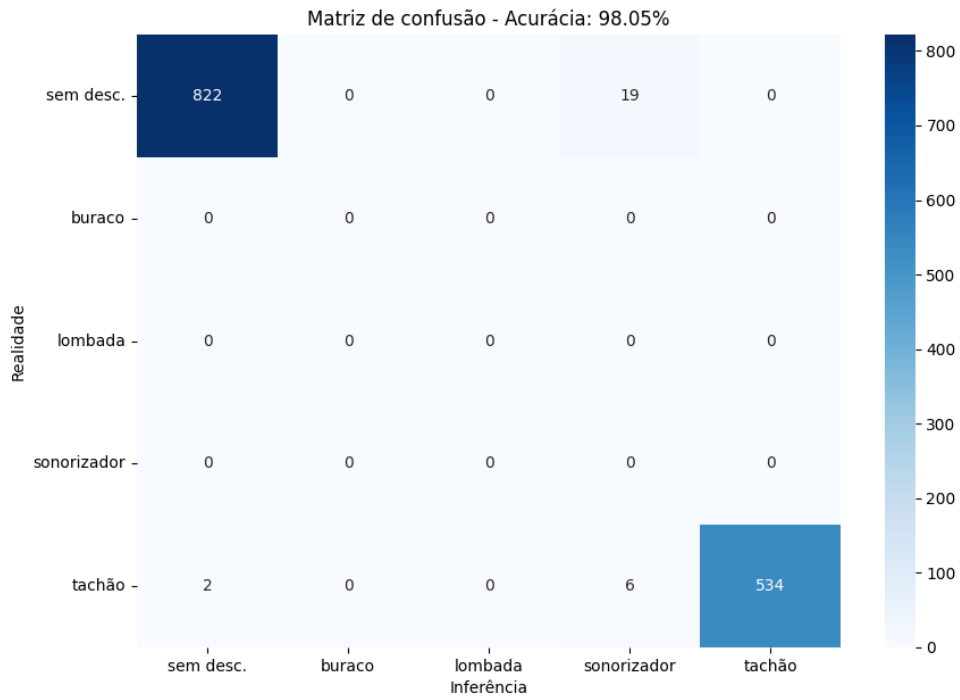
Fonte: Próprio Autor.

anteriormente, houveram pontos de imprecisão do modelo, sob os quais será feito um juízo de suas magnitudes.

Como é possível notar na figura 30, as maiores concentrações de palpites imprecisos foram no cenários em que estavam inclusos pares não envolvendo descontinuidades, mais especificamente os pares simétricos "3-0" e "0-3", ou seja a situação de estrada normal-sonorizador. Isso pode ser explicado pela magnitude e perfil de ambos, nos quais, para o caso do sonorizador é o que apresenta maior similaridade com o dado apenas conteúdo o ruído do pavimento, ou seja, que um desvio de acurácia menor do modelo tem maior chance de levar a palpite errôneo. Assim, espera-se que na inclusão de uma maior massa de testes referentes ao cenário de sonorizador no caso de avanços deste modelo, seria possível aumentar a acurácia geral do modelo e também para o par de valor real/inferência levantado.

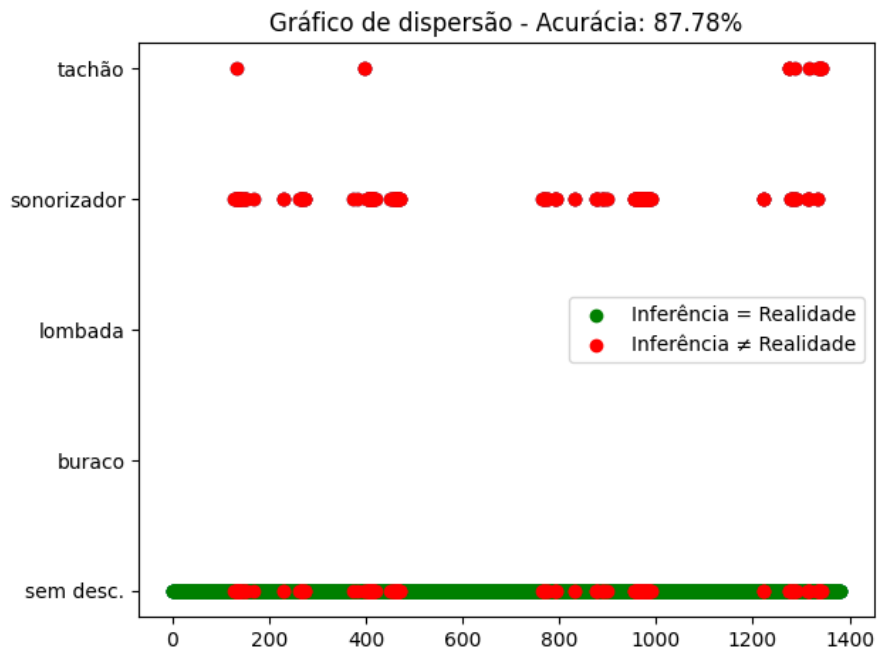
A baixa variabilidade quanto à apresentação de resultados imprecisos dentre os tipos de obstáculos classificados, retirados o par citado no parágrafo anterior, é consequência da boa distribuição de dados de teste para cada um dos tipos e a se inferir. Ou seja, o bom balanceamento dos dados também corroborou para sua boa performance dentre os diferentes tipos, os quais mantiveram valores variando dentro do intervalo de 60-140 imprecisões, em média, o que representa menos de 0,1% da massa de dados total, que engloba 158.000 janelas de predição.

Figura 27 – Matriz de Confusão da inferência para o obstáculo de Tachão



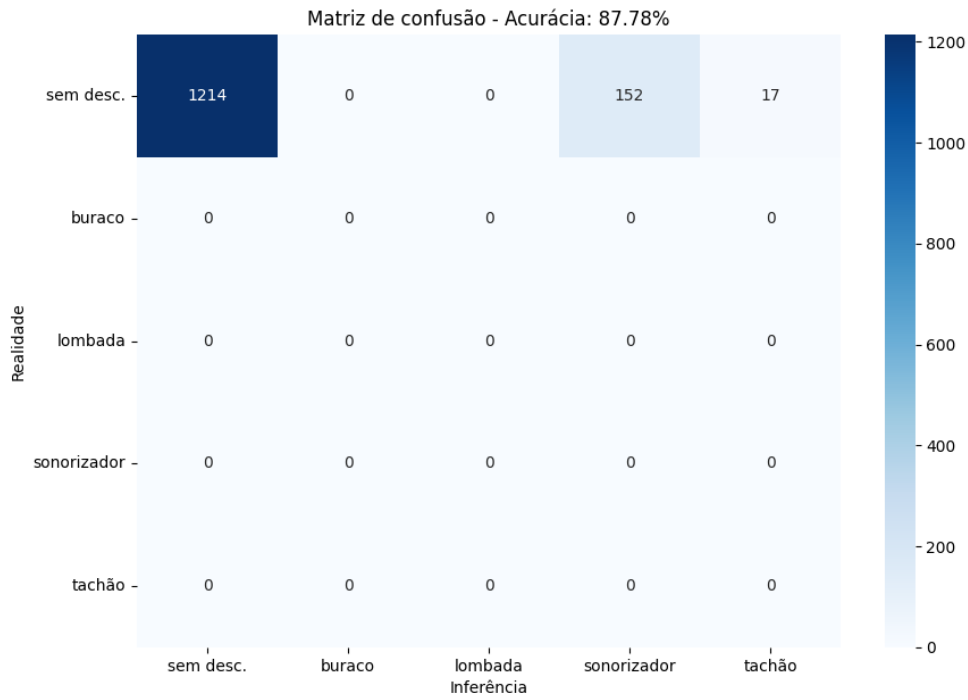
Fonte: Próprio Autor.

Figura 28 – Gráfico de Dispersão da inferência para o obstáculo de Estrada normal



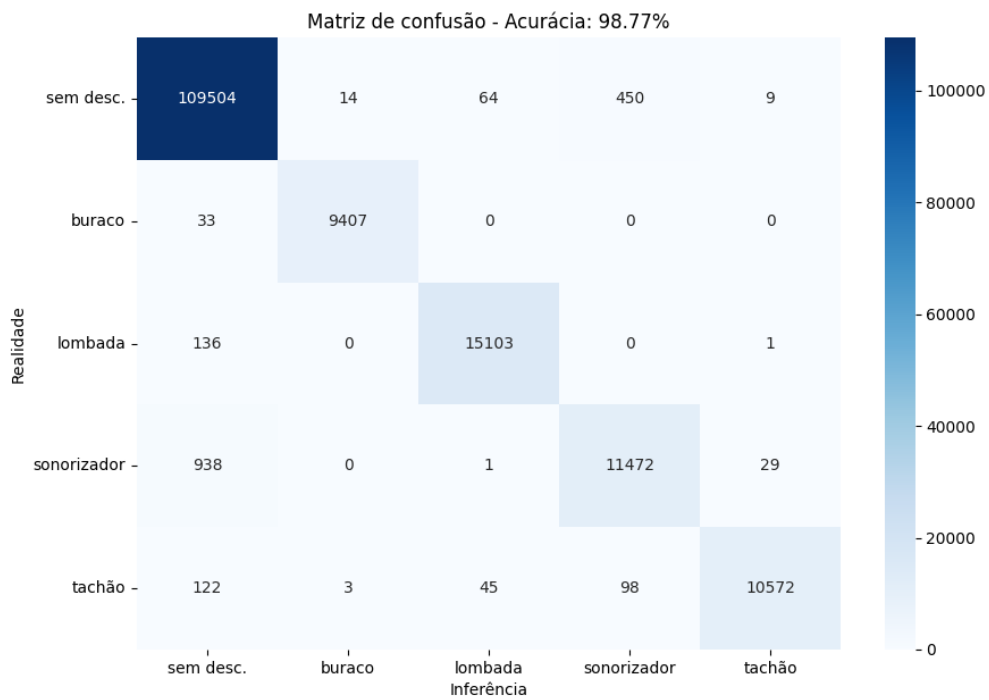
Fonte: Próprio Autor.

Figura 29 – Matriz de Confusão da inferência para o obstáculo de Estrada normal



Fonte: Próprio Autor.

Figura 30 – Matriz de confusão do Modelo de Rede Neural



Fonte: Próprio Autor.

5 CONCLUSÃO

A rotina de geração de dados de simulação elaborada permitiu emular as perturbações resultantes nos GDLs do sistema para o modelo de carro completo, não obstante, também permitindo a análise da resposta de aceleração de um corpo fora de centro ao veículo modelado. Assim, dado os diferentes tipos de imperfeições do pavimento apontados, dados teóricos foram obtidos, acrescidos de ruídos randomicamente distribuídos, permitindo a simulação e treinamento de um modelo de rede neural para a detecção dos tipos e da existência ou não de tais obstáculos nas vias, visando a criação de um cenário análogo a um dispositivo móvel preso a um veículo, considerando variáveis como velocidade e parâmetros de suspensão.

Foi possível validar as análises iniciais quanto ao tipo de modelo de rede neural capaz de executar as inferências desejadas. O modelo com base em convoluções, geralmente utilizado para tratativas de imagem, se mostrou competente para a análise do sinal tridimensional das acelerações, os quais puderam ser observados analogamente a uma imagem de padrão colorido a partir da matiz verde, vermelho e azul. Assim, com uma precisão global acima de 95% na inferência do modelo, utilizando infraestrutura para treinamento presentes em computadores pessoais de uso cotidiano, julga-se o modelo como eficiente e de performance satisfatória.

Dado esse contexto, valida-se os objetivos do presente trabalho como atingidos, dando embasamento teórico para o treinamento e modelagem de redes neurais a partir de dados de simulação de aceleração num modelo de carro completo, produzindo, portanto, uma rede neural capaz de ser usada em cenários reais de detecção de perturbações em vias pavimentadas, sem que haja necessidade de levantamento de uma grande quantidade de dados experimentais para o cenário específico, o que poderia ser temporalmente custoso. Adicionalmente, em evoluções futuras dos tópicos abrangidos, destaca-se o interesse em validar a utilização de outras arquiteturas de modelo de rede neural para a solução do problema em questão, validando-o quanto à eficiência e precisão. Também como excedente ao presente trabalho, é possível um incremento à modelagem teórica dos sinais de aceleração, acrescentando-a outros graus de liberdade, presentes, por exemplo, em um suporte de celular para veículos, no qual a análise de momento presente no "braço" gerado pela fixação do suporte e modelagem da mola circular que exerce força contrária ao movimento do aparelho se demonstram como variáveis relevantes do problema, possivelmente levando a uma maior precisão dos dados resultantes da simulação, assim como o confronto do mesmo com o cenário simplificado abordado por este trabalho.

REFERÊNCIAS

- BHAMARE, L. et al. Study of types of road abnormalities and techniques used for their detection. In: **2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)**. [S.l.: s.n.], 2021. p. 472–477. Citado na página 20.
- BLECHSCHMIDT, J.; ERNST, O. G. Three ways to solve partial differential equations with neural networks — a review. **GAMM-Mitteilungen**, v. 44, n. 2, 2021. Citado na página 18.
- CARVALHO, C. H. R. de. **Desafios da Mobilidade Urbana no Brasil**. 1. ed. Brasília: Instituto de Pesquisa Econômica Aplicada – Ipea, 2016. Citado na página 15.
- CBTU; TRENSURB. **Pemob**. 2022. Disponível em: <<https://www.gov.br/cidades/pt-br/assuntos/mobilidade-urbana/pemob-2022>>. Acesso em: 14 jul 2023. Citado na página 15.
- CERRI, R.; BARROS, R. C.; CARVALHO, A. C. de. Hierarchical multi-label classification using local neural networks. **Journal of Computer and System Sciences**, v. 80, n. 1, p. 39–56, 2014. ISSN 0022-0000. Citado na página 16.
- CHHABRA, R.; SINGH, S. A survey on smart phone-based road condition detection systems. In: **International Conference on Emerging Technologies: AI, IoT, and CPS for Science & Technology Applications**. NITTTTR Chandigarh, India: [s.n.], 2021. Citado na página 20.
- CUOMO, S. D. C. et al. Scientific machine learning through physics–informed neural networks: Where we are and what’s next. **Journal of Scientific Computing**, v. 92, n. 3, p. 88, Jul 2022. ISSN 1573-7691. Citado na página 18.
- DEODATO, J. P. R. **Rotina computacional para projeto de mola helicoidal de compressão aplicado a um protótipo veicular do tipo Baja Sae**. [S.l.], 2019. Citado na página 15.
- DIAS, S. I. S.; MUKAI, H.; FEIBER, F. N. Ergonomia e urbanismo: a visão sociotécnica, e a atuação do ergonomista urbano. FAG, 2006. Citado na página 15.
- GALDO, A. L. et al. **Detecting train driveshaft damages using accelerometer signals and Differential Convolutional Neural Networks**. [S.l.], 2022. Citado na página 19.
- GANESH, P. **Types of Convolution Kernels : Simplified**. 2019. Disponível em: <<https://towardsdatascience.com/types-of-convolution-kernels-simplified-f040cb307c37>>. Acesso em: 19 jul 2023. Citado na página 29.
- HARDIYANTI, N.; LAWI, A.; DIARAYA. Classification of human activity based on sensor accelerometer and gyroscope using ensemble svm method. In: . [S.l.: s.n.], 2018. p. 304–307. Citado na página 16.
- IBGE. **Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal 2021 / IBGE, Coordenação de Pesquisas por Amostra de Domicílios**. 2022. Disponível em: <https://biblioteca.ibge.gov.br/visualizacao/livros/liv101963_informativo.pdf>. Acesso em: 17 jun 2023. Citado na página 16.

IBGE. **De 2010 a 2022, população brasileira cresce 6,5203,1 milhões**. 2023. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/37237-de-2010-a-2022-populacao-brasileira-cresce-6-5-e-chega-a-203-1-milhoes>>. Acesso em: 14 jul 2023. Citado na página 15.

JAZAR, R. N. **Vehicle Dynamics: Theory and Application**. 1. ed. [S.l.]: Springer, 2008. Citado 3 vezes nas páginas 21, 22 e 30.

KOLLMANNBERGER, S. et al. Physics-informed neural networks. In: _____. **Deep Learning in Computational Mechanics: An Introductory Course**. Cham: Springer International Publishing, 2021. p. 55–84. ISBN 978-3-030-76587-3. Citado na página 18.

LEAL, L. d. C. M.; ROSA, E. d.; NICOLAZZI, L. C. **Uma introdução à modelagem quase-estática de automóveis**. [S.l.]: GRANTE - Departamento de Engenharia Mecânica da UFSC, 2001. Citado na página 15.

MINOZZO, R. **Aquisição de forças dinâmicas atuantes na suspensão dianteira de um veículo protótipo Baja SAE**. [S.l.], 2012. Citado na página 15.

NITSCHKE, P. et al. Comparison of machine learning methods for evaluating pavement roughness based on vehicle response. **Journal of Computing in Civil Engineering**, v. 28, n. 4, p. 04014015, 2014. Citado na página 20.

REGINATTO Éverton. **Avaliação de uma suspensão duplo A aplicada na estrutura de um veículo VW Fusca visando aumento da vida útil do amortecedores**. Dissertação (Trabalho de conclusão de curso) — Universidade de Caxias do Sul, Campus Universitário da Região dos Vinhedos, 2020. Citado na página 30.

SABAH, L.; ŞİMŞEK, M. Highways disturbance detection with the use of open source code geographic information systems, mobile and web applications. **ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences**, IV-4/W4, p. 325–329, 11 2017. Citado na página 19.

SANTOS, J. A. dos. **Construção de lombadas em vias urbanas: um estudo de caso na cidade de Patrocínio-MG**. Dissertação (Trabalho de conclusão de curso) — Centro Universitário do Cerrado Patrocínio, 2018. Curso de Engenharia Civil. Citado na página 31.

SHARMA, A.; LEE, Y.-D.; CHUNG, W.-Y. **High Accuracy Human Activity Monitoring using Neural network**. [S.l.], 2011. Citado na página 19.

TRANSITO, P. do. **Tempo gasto no trânsito por moradores das capitais do país**. 2022. Site "Portal do Trânsito". Disponível em: <<https://www.portaldotransito.com.br/noticias/mobilidade-e-tecnologia/pesquisa-mostra-que-moradores-das-capitais-do-pais-gastam-em-media-2h-por-dia-no-transito/>>. Acesso em: 07 mai 2023. Citado na página 15.

UM, T. T.; BABAKESHIZADEH, V.; KULIĆ, D. **Exercise Motion Classification from Large-Scale Wearable Sensor Data Using Convolutional Neural Networks**. [S.l.], 2016. Citado na página 19.

WANG, L. et al. **Animal Behavior Classification via Accelerometry Data and Recurrent Neural Networks**. [S.l.], 2021. Citado na página 18.

APÊNDICE A – ROTINA COMPUTACIONAL MATLAB PARA SIMULAÇÃO

```

1 close all;
2 clear; clc
3 %% Defini o das variaveis do modelo [Meio carro - Jazzar] (Pagina 878)
4 % No modelo do Jazzar, considerada uma barra anti-rolagem na parte
5 % frontal do carro. Como em ve culos baja a barra anti-rolagem na
6 % traseira , os dados incluidos s e r o invertidos em frontal/traseiro ,
   apesar
7 % de manter a nomenclatura da refer ncia.
8 m = 1250; % [kg] massa do carro
9 mf = 20; % [kg] (roda dianteira)
10 mr = 22; % [kg] (roda traseira)
11
12 Ix = 4100; % [kgm2] momento de in rcia longitudinal
13 Iy = 5500; % [kgm2] momento de in rcia lateral
14
15 a1 = 1.1; % [m] distancia do eixo dianteiro ao centro
16 a2 = 1.0; % [m] distancia do eixo traseiro ao centro
17 b1 = 0.65; % [m] distancia da roda esquerda ao centro
18 b2 = 0.65; % [m] distancia da roda direita ao centro
19
20 dpx = 0.8; % [m] distancia do ponto de interesse (dispositivo) ao centro
   do carro
21 dpy = 0.3; % [m] distancia (lateral) do ponto de interesse (dispositivo)
   ao centro do carro
22 dpz = 0; % [m] distancia (altura) do ponto de interesse (dispositivo)
   ao centro do carro
23
24 kf = 20e3; % [N/m] rigidez da suspens o dianteira
25 kr = 20e3; % [N/m] rigidez da suspens o traseira
26 k_R_roll = 30000; % [N/m] rigidez da barra anti rolagem
27
28 ktf = 200e3; % [N/m] rigidez do pneu dianteiro
29 ktr = 200e3; % [N/m] rigidez do pneu traseiro
30
31 cf = 3e3; % [Ns/m] amortecimento da suspens o dianteira
32 cr = 3e3; % [Ns/m] amortecimento da suspens o traseira
33
34 %% Parametros de percurso
35
36 resol_simu = 0.01; % [m] step de distancia por unidade de tempo em x
37 x_pavimento = (0:resol_simu:10); % [m]Posicao horizontal do pavimento p/
   cada dx
38

```

```
39 prof_buraco = 0.3; % [m] Altura do buraco no ponto mais profundo
40 compr_buraco = 0.25; % [m] Distancia de uma extremidade a outra do
    buraco, em x
41 altura_lombada = 0.15; % [m] Altura da lombada
42 compr_lombada= 1.7; % [m] Distancia de uma extreminadade a outra da
    lombada, em x
43 compr_sonorizador = 1; % [m] Distancia de uma extremidade a outra do
    sonorizador, em x
44 compr_tartaruga = 0.6; % [m] Distancia de uma extremidade a outra da
    tartaruga, em x
45
46 dist_entre_eixos = a1 + a2;
47 inicio_descontinuidade = (4 - dist_entre_eixos - 0.1) * rand() + a1 + a2
    + 0.1; % [m] Posicao inicial da descontinuidade no pavimento
48
49 percursos = ["lombada", "sonorizador", "buraco", "sem buraco", "
    tartaruga"];
50
51 for j=1:length(percursos)
52     y_lh_pavimento = zeros(size(x_pavimento)); % [m] inicializa posicao
    vertical do pavimento p/ roda esquerda a cada x
53     y_rh_pavimento = zeros(size(x_pavimento)); % [m] inicializa posicao
    vertical do pavimento p/ roda direita a cada x
54
55     percurso = percursos(j);
56
57     if percurso == "buraco"
58         compr_interesse = compr_buraco;
59     elseif percurso == "lombada"
60         compr_interesse = compr_lombada;
61     elseif percurso == "sonorizador"
62         compr_interesse = compr_sonorizador;
63     elseif percurso == "tartaruga"
64         compr_interesse = compr_tartaruga;
65     end
66
67     cont = 1;
68     for i=round(inicio_descontinuidade/resol_simu):round((
    inicio_descontinuidade+compr_interesse)/resol_simu)
69         if percurso == "buraco"
70             y_lh_pavimento(i) = -prof_buraco*sin(3*(1/compr_buraco)*
    x_pavimento(cont)); %//buraco
71             y_rh_pavimento(i) = -prof_buraco*sin(3*(1/compr_buraco)*
    x_pavimento(cont)); %//buraco
72         elseif percurso == "lombada"
73             y_lh_pavimento(i) = altura_lombada*sin(3*(1/compr_lombada)*
    x_pavimento(cont)); %//lombada
```

```
74         y_rh_pavimento(i) = altura_lombada*sin(3*(1/compr_lombada)*
x_pavimento(cont)); %//lombada
75         elseif percurso == "sonorizador"
76             y_lh_pavimento(i) = max(0.05*sin(30*(1/compr_interesse)*
x_pavimento(cont)), 0); %//sonorizador
77             y_rh_pavimento(i) = max(0.05*sin(30*(1/compr_interesse)*
x_pavimento(cont)), 0); %//sonorizador
78         elseif percurso == "tartaruga"
79             y_lh_pavimento(i) = max(0.1*sin(10*(1/compr_interesse)*
x_pavimento(cont)), 0); %//tartaruga
80             y_rh_pavimento(i) = max(0.1*sin(10*(1/compr_interesse)*
x_pavimento(cont)), 0); %//tartaruga
81         end
82
83         cont = cont + 1;
84     end
85
86 %% Imperfeicoes do asfalto (Distribuicao Gaussiana)
87
88 % Define a faixa de altura das imperfeicoes
89 disturbanceRange = [-0.02, 0.01]; % [m] Valores maximo e minimo para
as imperfeicoes
90
91 % Define a aspereza do asfalto
92 tarmacRoughness = 0.5; % Atrator na distribuicao randomica dos
valores simulando qualidade do asfalto, onde 0 e suave e 1 e aspero
93
94 % Calcula a media e o desvio padrao dos disturbios
95 disturbanceMean = mean(disturbanceRange);
96 disturbanceStdDev = diff(disturbanceRange) * tarmacRoughness;
97
98 % Gera os disturbios
99 disturbances = disturbanceMean + disturbanceStdDev * randn(1, length
(y_lh_pavimento));
100
101 % Adiciona os disturbios ao mapa de altura
102 y_lh_pavimento = y_lh_pavimento + disturbances;
103 y_rh_pavimento = y_rh_pavimento + disturbances;
104
105 %% Cinetica e defasagem no percurso
106
107 a = 5; % [m/s] velocidade do carro
108 v = a*ones(size(x_pavimento)); % [m/s] vetor velocidade do carro
109
110 def_x = round(((dist_entre_eixos) / resol_simu)) + 1; % Defasagem no
array devido as duas rodas precisarem estar dentro da simulacao
111 aux = def_x - 1; % Auxiliar pra diminuir repeticao
```

```

112
113     y1 = zeros(1,length(x_pavimento)-aux);
114     y2 = zeros(1,length(x_pavimento)-aux);
115     y3 = zeros(1,length(x_pavimento)-aux);
116     y4 = zeros(1,length(x_pavimento)-aux);
117
118     t = zeros(1,length(x_pavimento)-aux);
119
120     for i=def_x:1:length(x_pavimento)
121         y1(i-aux) = y_lh_pavimento(i);
122         y2(i-aux) = y_rh_pavimento(i);
123
124         y3(i-aux) = y_rh_pavimento(i-aux);
125         y4(i-aux) = y_lh_pavimento(i-aux);
126     end
127
128     for i=def_x:1:length(x_pavimento)-1
129         t(i-aux+1) = ((x_pavimento(i-aux+1)- x_pavimento(i-aux))/v(i-aux
130 ))+t(i-aux);
131     end
132
133     figure
134     plot(x_pavimento ,y_lh_pavimento);
135     set(gca,"fontname","times new roman");
136     set(gca,"fontsize",12); title("Percurso roda esquerda - " + percurso
137 )
138     xlabel("Posicao horizontal [m]")
139     ylabel("Posicao vertical [m]")
140     saveas(gcf, "percurso_lh_" + percurso + ".png");
141     % [ymax2,imax2] = max(y_lh_pavimento(:,1));
142     % [ymin2,imin2] = min(y_lh_pavimento(:,1));
143     % ylim([ymin2-0.05,ymax2+0.05]);
144
145     figure
146     plot(x_pavimento ,y_rh_pavimento);
147     set(gca,"fontname","times new roman");
148     set(gca,"fontsize",12);
149     title("Percurso roda direita - " + percurso)
150     xlabel("Posicao horizontal [m]")
151     ylabel("Posicao vertical [m]")
152     saveas(gcf, "percurso_rh_" + percurso + ".png");
153     % [ymax2,imax2] = max(y_rh_pavimento(:,1));
154     % [ymin2,imin2] = min(y_rh_pavimento(:,1));
155     % ylim([ymin2-0.05,ymax2+0.05]);
156
157     figure
158     plot(t,y1);

```

```
157     set(gca,"fontname","times new roman");
158     set(gca,"fontsize",12);
159     title("Deslocamento da roda esquerda frontal - " + percurso)
160     xlabel("Tempo [s]")
161     ylabel("Posicao vertical [m]")
162     saveas(gcf, "deslocamento_lh_" + percurso + ".png");
163     % [ymax2,imax2] = max(y1(1,:));
164     % [ymin2,imin2] = min(y1(1,:));
165     % ylim([ymin2-0.05,ymax2+0.05]);
166
167     figure
168     plot(t,y4);
169     set(gca,"fontname","times new roman");
170     set(gca,"fontsize",12);
171     title("Deslocamento da roda esquerda traseira - " + percurso)
172     xlabel("Tempo [s]")
173     ylabel("Posicao vertical [m]")
174     saveas(gcf, "deslocamento_lh_traseira_" + percurso + ".png");
175     % [ymax2,imax2] = max(y4(1,:));
176     % [ymin2,imin2] = min(y4(1,:));
177     % ylim([ymin2-0.05,ymax2+0.05]);
178
179     figure
180     plot(t,y2);
181     set(gca,"fontname","times new roman");
182     set(gca,"fontsize",12);
183     title("Deslocamento da roda direita frontal - " + percurso)
184     xlabel("Tempo [s]")
185     ylabel("Posicao vertical [m]")
186     saveas(gcf, "deslocamento_rh_" + percurso + ".png");
187     % [ymax2,imax2] = max(y2(1,:));
188     % [ymin2,imin2] = min(y2(1,:));
189     % ylim([ymin2-0.05,ymax2+0.05]);
190
191     figure
192     plot(t,y3);
193     set(gca,"fontname","times new roman");
194     set(gca,"fontsize",12);
195     title("Deslocamento da roda direita traseira - " + percurso)
196     xlabel("Tempo [s]")
197     ylabel("Posicao vertical [m]")
198     saveas(gcf, "deslocamento_rh_traseira_" + percurso + ".png");
199     % [ymax2,imax2] = max(y3(1,:));
200     % [ymin2,imin2] = min(y3(1,:));
201     % ylim([ymin2-0.05,ymax2+0.05]);
202
203     %% D e f i n i o das matrizes (massa , amortecimento , rigidez)
```

```

204     matriz_m = zeros(7,7);
205         matriz_m(1,1) = m;
206         matriz_m(2,2) = Ix;
207         matriz_m(3,3) = Iy;
208         matriz_m(4,4) = mf;
209         matriz_m(5,5) = mf;
210         matriz_m(6,6) = mr;
211         matriz_m(7,7) = mr;
212
213     matriz_c = zeros(7,7);
214         c11 = 2*cf + 2*cr;
215         c21 = b1*cf - b2*cf - b1*cr - b2*cr;
216         c31 = 2*a2*cr - 2*a1*cf;
217         c22 = (cf + cr)*(b1^2 + b2^2);
218         c32 = a1*b2*cf - a1*b1*cf - a2*b1*cr + a2*b2*cr;
219         c33 = 2*cf*a1^2 + 2*cr*a2^2;
220
221     matriz_c(1,:) = [c11, c21, c31, -cf, -cf, -cr, -cr];
222     matriz_c(2,:) = [c21, c22, c32, -b1*cf, b2*cf, b1*cr, -b2*cr];
223     matriz_c(3,:) = [c31, c32, c33, a1*cf, a1*cf, -a2*cr, -a2*cr];
224     matriz_c(4,:) = [-cf, -b1*cf, a1*cf, cf, 0, 0, 0];
225     matriz_c(5,:) = [-cf, b2*cf, a1*cf, 0, cf, 0, 0];
226     matriz_c(6,:) = [-cr, b1*cr, -a2*cr, 0, 0, cr, 0];
227     matriz_c(7,:) = [-cr, -b2*cr, -a2*cr, 0, 0, 0, cr];
228
229     matriz_k = zeros(7,7);
230     wf = b1+b2; %Pela barra de rolagem dianteira wf = b1+b2 <> krf = kr
231         k11 = 2*kf + 2*kr;
232         k21 = b1*kf - b2*kf - b1*kr + b2*kr; k31 = 2*a2*kr - 2*a1*kf;
233         k22 = k_R_roll + (kf + kr)*(b1^2 + b2^2);
234         k32 = a1*b2*kf - a1*b1*kf - a2*b1*kr + a2*b2*kr; k42 = -b1*kf -
(k_R_roll/wf);
235         k52 = b2*kf + (k_R_roll/wf);
236         k33 = 2*kf*(a1^2) + 2*kr*(a2^2); k44 = kf + ktf + (k_R_roll/(wf
^2));
237         k55 = kf + ktf + (k_R_roll/(wf^2));
238
239     matriz_k(1,:) = [k11, k21, k31, -kf, -kf, -kr, -kr];
240     matriz_k(2,:) = [k21, k22, k32, k42, k52, b1*kr, -b2*kr];
241     matriz_k(3,:) = [k31, k32, k33, a1*kf, a1*kf, -a2*kr, -a2*kr];
242     matriz_k(4,:) = [-kf, k42, a1*kf, k44, -k_R_roll/(wf^2), 0, 0];
243     matriz_k(5,:) = [-kf, k52, a1*kf, -k_R_roll/(wf^2), k55, 0, 0];
244     matriz_k(6,:) = [-kr, b1*kr, -a2*kr, 0, 0, kr+ktr, 0];
245     matriz_k(7,:) = [-kr, -b2*kr, -a2*kr, 0, 0, 0, kr+ktr];
246
247     %% Sistema [m]x"" + [c]x" + [k]x = F
248     s = tf("s");

```



```
249 H = ((matriz_m*s^2) + (matriz_c*s) + (matriz_k));
250 H = H\eye(size(H)); % Divide por identidade
251
252 F = [0; 0; 0; ktf; ktf; ktr; ktr]; % Vetor forza
253
254 % CG
255 X_y1 = H(1,4)*ktf;
256 X_y2 = H(1,5)*ktf;
257 X_y3 = H(1,6)*ktr;
258 X_y4 = H(1,7)*ktr;
259 % CG no tempo
260 [auxC1,t] = lsim(X_y1,y1,t);
261 [auxC2,t] = lsim(X_y2,y2,t);
262 [auxC3,t] = lsim(X_y3,y3,t);
263 [auxC4,t] = lsim(X_y4,y4,t);
264 saida_CG(:,1) = auxC1(:,1) + auxC2(:,1) + auxC3(:,1) + auxC4(:,1);
265
266 % Rolagem
267 Phi_y1 = H(2,4)*ktf;
268 Phi_y2 = H(2,5)*ktf;
269 Phi_y3 = H(2,6)*ktr;
270 Phi_y4 = H(2,7)*ktr;
271 % Rolagem No tempo
272 [auxP1,t] = lsim(Phi_y1,y1,t);
273 [auxP2,t] = lsim(Phi_y2,y2,t);
274 [auxP3,t] = lsim(Phi_y3,y3,t);
275 [auxP4,t] = lsim(Phi_y4,y4,t);
276 giro_rh_lh(:,1) = auxP1(:,1) + auxP2(:,1) + auxP3(:,1) + auxP4(:,1);
277
278 %Arfagem
279 Theta_y1 = H(3,4)*ktf;
280 Theta_y2 = H(3,5)*ktf;
281 Theta_y3 = H(3,6)*ktr;
282 Theta_y4 = H(3,7)*ktr;
283 % Arfagem No tempo
284 [auxT1,t] = lsim(Theta_y1,y1,t);
285 [auxT2,t] = lsim(Theta_y2,y2,t);
286 [auxT3,t] = lsim(Theta_y3,y3,t);
287 [auxT4,t] = lsim(Theta_y4,y4,t);
288 giro_fr_rh(:,1) = auxT1(:,1) + auxT2(:,1) + auxT3(:,1) + auxT4(:,1);
289
290 % Rodas
291 X1_y1 = H(4,4)*ktf;
292 X1_y2 = H(4,5)*ktf;
293 X1_y3 = H(4,6)*ktr;
294 X1_y4 = H(4,7)*ktr;
295
```

```

296     X2_y1 = H(5,4)*ktf;
297     X2_y2 = H(5,5)*ktf;
298     X2_y3 = H(5,6)*ktr;
299     X2_y4 = H(5,7)*ktr;
300
301     X3_y1 = H(6,4)*ktf;
302     X3_y2 = H(6,5)*ktf;
303     X3_y3 = H(6,6)*ktr;
304     X3_y4 = H(6,7)*ktr;
305
306     X4_y1 = H(7,4)*ktf;
307     X4_y2 = H(7,5)*ktf;
308     X4_y3 = H(7,6)*ktr;
309     X4_y4 = H(7,7)*ktr;
310
311     %% Ponto de interesse
312     % Define a posicao do ponto de interesse em relacao ao CG
313     r = repmat([dpx, dpy, dpz], size(giro_rh_lh, 1), 1); % repete o
vetor linha para cada passo de tempo
314
315     % Calcula a posicao angular, velocidade e aceleracao
316     omega = [zeros(size(giro_rh_lh)), [0; diff(giro_rh_lh)], [0; diff(
giro_fr_rh)]]; % velocidade angular
317     alpha = [zeros(size(giro_rh_lh)), [0; diff(omega(:,2))], [0; diff(
omega(:,3))]]; % aceleracao angular
318
319     % Calcula a velocidade e aceleracao do CG
320     V_CG = [0; diff(saida_CG)];
321     A_CG = [0; diff(V_CG)];
322
323     % Calcula a velocidade e aceleracao do ponto de interesse
324     V = V_CG + cross(omega, r);
325     A = A_CG + cross(alpha, r) + cross(omega, cross(omega, r));
326
327     % Calcula a posicao do ponto de interesse integrando a velocidade
328     P = cumtrapz(V) + r; % Adiciona de volta a constante de integracao,
que e a posicao inicial (quando X = 0)
329
330     %% Resultados das FTs
331     figure
332     plot(t, saida_CG(:,1));
333     set(gca, "fontname", "times new roman");
334     set(gca, "fontsize", 12);
335     title("Deslocamento vertical do CG - " + percurso);
336     xlabel("Tempo [s]");
337     ylabel("Deslocamento [m]");
338     saveas(gcf, "deslocamento_cg_" + percurso + ".png");

```

```
339
340     figure
341     plot(t,giro_rh_lh(:,1));
342     set(gca,"fontname","times new roman");
343     set(gca,"fontsize",12);
344     title("Rolagem (esquerda/direita) - " + percurso);
345     xlabel("Tempo [s]");
346     ylabel("Inclinação [rad]");
347     saveas(gcf, "rolagem_" + percurso + ".png");
348
349     figure
350     plot(t,giro_fr_rh(:,1));
351     set(gca,"fontname","times new roman");
352     set(gca,"fontsize",12);
353     title("Arfagem (dianteira/traseira) - " + percurso);
354     xlabel("Tempo [s]");
355     ylabel("Inclinação [rad]");
356     saveas(gcf, "arfagem_" + percurso + ".png");
357
358     figure
359     [aux1,t] = lsim(X1_y1,y1,t);
360     [aux2,t] = lsim(X1_y2,y2,t);
361     [aux3,t] = lsim(X1_y3,y3,t);
362     [aux4,t] = lsim(X1_y4,y4,t);
363     saida_mola_x1(:,1) = aux1(:,1) + aux2(:,1) + aux3(:,1) + aux4(:,1);
364     plot(t,saida_mola_x1(:,1));
365     title("Deslocamento vertical da roda dianteira esquerda - " +
percurso)
366     set(gca,"fontname","times new roman");
367     set(gca,"fontsize",12); xlabel("Tempo [s]");
368     ylabel("Deslocamento [m]");
369     saveas(gcf, "deslocamento_lh_" + percurso + ".png");
370
371     figure
372     [aux1,t] = lsim(X2_y1,y1,t);
373     [aux2,t] = lsim(X2_y2,y2,t);
374     [aux3,t] = lsim(X2_y3,y3,t);
375     [aux4,t] = lsim(X2_y4,y4,t);
376     saida_mola_x2(:,1) = aux1(:,1) + aux2(:,1) + aux3(:,1) + aux4(:,1);
377     plot(t,saida_mola_x2(:,1));
378     title("Deslocamento vertical da roda dianteira direita - " +
percurso)
379     set(gca,"fontname","times new roman");
380     set(gca,"fontsize",12); xlabel("Tempo [s]");
381     ylabel("Deslocamento [m]");
382     saveas(gcf, "deslocamento_rh_" + percurso + ".png");
383
```

```

384     figure
385     [aux1,t] = lsim(X3_y1,y1,t);
386     [aux2,t] = lsim(X3_y2,y2,t);
387     [aux3,t] = lsim(X3_y3,y3,t);
388     [aux4,t] = lsim(X3_y4,y4,t);
389     saida_mola_x3(:,1) = aux1(:,1) + aux2(:,1) + aux3(:,1) + aux4(:,1);
390     plot(t,saida_mola_x3(:,1));
391     title("Deslocamento vertical da roda traseira direita - " + percurso
)
392     set(gca,"fontname","times new roman");
393     set(gca,"fontsize",12);
394     xlabel("Tempo [s]");
395     ylabel("Deslocamento [m]");
396     saveas(gcf, "deslocamento_rh_traseira_" + percurso + ".png");
397
398     figure
399     [aux1,t] = lsim(X4_y1,y1,t);
400     [aux2,t] = lsim(X4_y2,y2,t);
401     [aux3,t] = lsim(X4_y3,y3,t);
402     [aux4,t] = lsim(X4_y4,y4,t);
403     saida_mola_x4(:,1) = aux1(:,1) + aux2(:,1) + aux3(:,1) + aux4(:,1);
404     plot(t,saida_mola_x4(:,1));
405     title("Deslocamento vertical da roda traseira esquerda - " +
percurso)
406     set(gca,"fontname","times new roman");
407     set(gca,"fontsize",12);
408     xlabel("Tempo [s]");
409     ylabel("Deslocamento [m]");
410     saveas(gcf, "deslocamento_lh_traseira_" + percurso + ".png");
411
412     figure
413     hold on
414     plot(t,A(:,1));
415     plot(t,A(:,2));
416     plot(t,A(:,3));
417     hold off
418     legend('em X', 'em Y', 'em Z')
419     title("Aceleracao do Ponto de Interesse - " + percurso)
420     set(gca,"fontname","times new roman");
421     set(gca,"fontsize",12);
422     xlabel("Tempo [s]");
423     ylabel("Aceleracao [m/s^2]");
424     saveas(gcf, "aceleracao_ponto_interesse_" + percurso + ".png");
425
426 end
427
428 %% Gerar dados para treinamento e validacao do Modelo de Rede neural

```

```
429 % Devem ser gerados 10 conjuntos para cada um dos obstaculos, sendo
430 % persistido somente os dados de aceleracao do ponto de interesse nos 3
431 % eixos.
432
433 samples = 50;
434
435 def_x = round(((dist_entre_eixos) / resol_simu)) + 1; % Defasagem no
      array devido as duas rodas precisarem estar dentro da simulacao
436 aux = def_x - 1; % Auxiliar pra diminuir repeticao
437
438 for k=1:samples
439     for j=1:length(percursos)
440         y_lh_pavimento = zeros(size(x_pavimento)); % [m] inicializa
      posicao vertical do pavimento p/ roda esquerda a cada x
441         y_rh_pavimento = zeros(size(x_pavimento)); % [m] inicializa
      posicao vertical do pavimento p/ roda direita a cada x
442         labels = zeros(size(x_pavimento));
443
444         percurso = percursos(j);
445
446         if percurso == "buraco"
447             compr_interesse = compr_buraco;
448         elseif percurso == "lombada"
449             compr_interesse = compr_lombada;
450         elseif percurso == "sonorizador"
451             compr_interesse = compr_sonorizador;
452         elseif percurso == "tartaruga"
453             compr_interesse = compr_tartaruga;
454         end
455
456         cont = 1;
457         for i=round(inicio_descontinuidade/resol_simu):round((
      inicio_descontinuidade+compr_interesse)/resol_simu)
458             if percurso == "buraco"
459                 y_lh_pavimento(i) = -prof_buraco*sin(3*(1/compr_buraco)*
      x_pavimento(cont)); %//buraco
460                 y_rh_pavimento(i) = -prof_buraco*sin(3*(1/compr_buraco)*
      x_pavimento(cont)); %//buraco
461             elseif percurso == "lombada"
462                 y_lh_pavimento(i) = altura_lombada*sin(3*(1/
      compr_lombada)*x_pavimento(cont)); %//lombada
463                 y_rh_pavimento(i) = altura_lombada*sin(3*(1/
      compr_lombada)*x_pavimento(cont)); %//lombada
464             elseif percurso == "sonorizador"
465                 y_lh_pavimento(i) = max(0.05*sin(30*(1/compr_interesse)*
      x_pavimento(cont)), 0); %//sonorizador
466                 y_rh_pavimento(i) = max(0.05*sin(30*(1/compr_interesse)*
```

```
x_pavimento(cont)), 0); %//sonorizador
467     elseif percurso == "tartaruga"
468         y_lh_pavimento(i) = max(0.1*sin(10*(1/compr_interesse)*
x_pavimento(cont)), 0); %//tartaruga
469         y_rh_pavimento(i) = max(0.1*sin(10*(1/compr_interesse)*
x_pavimento(cont)), 0); %//tartaruga
470     end
471
472     val = 0;
473     if percurso == "buraco"
474         val = 1;
475     elseif percurso == "lombada"
476         val = 2;
477     elseif percurso == "sonorizador"
478         val = 3;
479     elseif percurso == "tartaruga"
480         val = 4;
481     end
482
483     cont = cont + 1;
484     labels(i:i+aux) = val;
485 end
486
487 % Define a faixa de altura das imperfeicoes
488 disturbanceRange = [-0.02, 0.01]; % [m] Valores maximo e minimo
para as imperfeicoes
489
490 % Define a aspereza do asfalto
491 tarmacRoughness = 0.5; % Atrator na distribuicao randomica dos
valores simulando qualidade do asfalto, onde 0 e suave e 1 e aspero
492
493 % Calcula a media e o desvio padrao dos disturbios
494 disturbanceMean = mean(disturbanceRange);
495 disturbanceStdDev = diff(disturbanceRange) * tarmacRoughness;
496
497 % Gera os disturbios
498 disturbances = disturbanceMean + disturbanceStdDev * randn(1,
length(y_lh_pavimento));
499
500 % Adiciona os disturbios ao mapa de altura
501 y_lh_pavimento = y_lh_pavimento + disturbances;
502 y_rh_pavimento = y_rh_pavimento + disturbances;
503
504
505 a = 5; % [m/s] velocidade do carro
506 v = a*ones(size(x_pavimento)); % [m/s] vetor velocidade do carro
507
```

```

508     y1 = zeros(1,length(x_pavimento)-aux);
509     y2 = zeros(1,length(x_pavimento)-aux);
510     y3 = zeros(1,length(x_pavimento)-aux);
511     y4 = zeros(1,length(x_pavimento)-aux);
512     labels = labels(1:length(labels)-aux);
513
514     t = zeros(1,length(x_pavimento)-aux);
515
516     for i=def_x:1:length(x_pavimento)
517         y1(i-aux) = y_lh_pavimento(i);
518         y2(i-aux) = y_rh_pavimento(i);
519
520         y3(i-aux) = y_rh_pavimento(i-aux);
521         y4(i-aux) = y_lh_pavimento(i-aux);
522     end
523
524     for i=def_x:1:length(x_pavimento)-1
525         t(i-aux+1) = ((x_pavimento(i-aux+1)- x_pavimento(i-aux))/v(i
526 -aux))+t(i-aux);
527     end
528
529     matriz_m = zeros(7,7);
530     matriz_m(1,1) = m;
531     matriz_m(2,2) = Ix;
532     matriz_m(3,3) = Iy;
533     matriz_m(4,4) = mf;
534     matriz_m(5,5) = mf;
535     matriz_m(6,6) = mr;
536     matriz_m(7,7) = mr;
537
538     matriz_c = zeros(7,7);
539     c11 = 2*cf + 2*cr;
540     c21 = b1*cf - b2*cf - b1*cr - b2*cr;
541     c31 = 2*a2*cr - 2*a1*cf;
542     c22 = (cf + cr)*(b1^2 + b2^2);
543     c32 = a1*b2*cf - a1*b1*cf - a2*b1*cr + a2*b2*cr;
544     c33 = 2*cf*a1^2 + 2*cr*a2^2;
545
546     matriz_c(1,:) = [c11, c21, c31, -cf, -cf, -cr, -cr];
547     matriz_c(2,:) = [c21, c22, c32, -b1*cf, b2*cf, b1*cr, -b2*cr];
548     matriz_c(3,:) = [c31, c32, c33, a1*cf, a1*cf, -a2*cr, -a2*cr];
549     matriz_c(4,:) = [-cf, -b1*cf, a1*cf, cf, 0, 0, 0];
550     matriz_c(5,:) = [-cf, b2*cf, a1*cf, 0, cf, 0, 0];
551     matriz_c(6,:) = [-cr, b1*cr, -a2*cr, 0, 0, cr, 0];
552     matriz_c(7,:) = [-cr, -b2*cr, -a2*cr, 0, 0, 0, cr];
553
554     matriz_k = zeros(7,7);

```

```

554     wf = b1+b2; %Pela barra de rolagem dianteira wf = b1+b2 <> krf =
      kr
555     k11 = 2*kf + 2*kr;
556     k21 = b1*kf - b2*kf - b1*kr + b2*kr; k31 = 2*a2*kr - 2*a1*kf
      ;
557     k22 = k_R_roll + (kf + kr)*(b1^2 + b2^2);
558     k32 = a1*b2*kf - a1*b1*kf - a2*b1*kr + a2*b2*kr; k42 = -b1*
      kf - (k_R_roll/wf);
559     k52 = b2*kf + (k_R_roll/wf);
560     k33 = 2*kf*(a1^2) + 2*kr*(a2^2); k44 = kf + ktf + (k_R_roll
      /(wf^2));
561     k55 = kf + ktf + (k_R_roll/(wf^2));
562
563     matriz_k(1,:) = [k11, k21, k31, -kf, -kf, -kr, -kr];
564     matriz_k(2,:) = [k21, k22, k32, k42, k52, b1*kr, -b2*kr];
565     matriz_k(3,:) = [k31, k32, k33, a1*kf, a1*kf, -a2*kr, -a2*kr
      ];
566     matriz_k(4,:) = [-kf, k42, a1*kf, k44, -k_R_roll/(wf^2), 0,
      0];
567     matriz_k(5,:) = [-kf, k52, a1*kf, -k_R_roll/(wf^2), k55, 0,
      0];
568     matriz_k(6,:) = [-kr, b1*kr, -a2*kr, 0, 0, kr+ktr, 0];
569     matriz_k(7,:) = [-kr, -b2*kr, -a2*kr, 0, 0, 0, kr+ktr];
570
571     s = tf("s");
572     H = ((matriz_m*s^2) + (matriz_c*s) + (matriz_k));
573     H = H\eye(size(H)); % Divide por identidade
574
575     F = [0; 0; 0; ktf; ktf; ktr; ktr]; % Vetor forca
576
577     % CG
578     X_y1 = H(1,4)*ktf;
579     X_y2 = H(1,5)*ktf;
580     X_y3 = H(1,6)*ktr;
581     X_y4 = H(1,7)*ktr;
582     % CG no tempo
583     [auxC1,t] = lsim(X_y1,y1,t);
584     [auxC2,t] = lsim(X_y2,y2,t);
585     [auxC3,t] = lsim(X_y3,y3,t);
586     [auxC4,t] = lsim(X_y4,y4,t);
587     saida_CG(:,1) = auxC1(:,1) + auxC2(:,1) + auxC3(:,1) + auxC4
      (:,1);
588
589     % Rolagem
590     Phi_y1 = H(2,4)*ktf;
591     Phi_y2 = H(2,5)*ktf;
592     Phi_y3 = H(2,6)*ktr;

```



```
593     Phi_y4 = H(2,7)*ktr;
594     % Rolagem No tempo
595     [auxP1,t] = lsim(Phi_y1,y1,t);
596     [auxP2,t] = lsim(Phi_y2,y2,t);
597     [auxP3,t] = lsim(Phi_y3,y3,t);
598     [auxP4,t] = lsim(Phi_y4,y4,t);
599     giro_rh_lh(:,1) = auxP1(:,1) + auxP2(:,1) + auxP3(:,1) + auxP4
(:,1);
600
601     %Arfagem
602     Theta_y1 = H(3,4)*ktr;
603     Theta_y2 = H(3,5)*ktr;
604     Theta_y3 = H(3,6)*ktr;
605     Theta_y4 = H(3,7)*ktr;
606     % Arfagem No tempo
607     [auxT1,t] = lsim(Theta_y1,y1,t);
608     [auxT2,t] = lsim(Theta_y2,y2,t);
609     [auxT3,t] = lsim(Theta_y3,y3,t);
610     [auxT4,t] = lsim(Theta_y4,y4,t);
611     giro_fr_rh(:,1) = auxT1(:,1) + auxT2(:,1) + auxT3(:,1) + auxT4
(:,1);
612
613     % Rodas
614     X1_y1 = H(4,4)*ktr;
615     X1_y2 = H(4,5)*ktr;
616     X1_y3 = H(4,6)*ktr;
617     X1_y4 = H(4,7)*ktr;
618
619     X2_y1 = H(5,4)*ktr;
620     X2_y2 = H(5,5)*ktr;
621     X2_y3 = H(5,6)*ktr;
622     X2_y4 = H(5,7)*ktr;
623
624     X3_y1 = H(6,4)*ktr;
625     X3_y2 = H(6,5)*ktr;
626     X3_y3 = H(6,6)*ktr;
627     X3_y4 = H(6,7)*ktr;
628
629     X4_y1 = H(7,4)*ktr;
630     X4_y2 = H(7,5)*ktr;
631     X4_y3 = H(7,6)*ktr;
632     X4_y4 = H(7,7)*ktr;
633
634     % Define a posicao do ponto de interesse em relacao ao CG
635     r = repmat([dpx, dpy, dpz], size(giro_rh_lh, 1), 1); % repete o
vetor linha para cada passo de tempo
636
```

```
637     % Calcula a posicao angular, velocidade e aceleracao
638     omega = [zeros(size(giro_rh_lh)), [0; diff(giro_rh_lh)], [0;
diff(giro_fr_rh)]]; % velocidade angular
639     alpha = [zeros(size(giro_rh_lh)), [0; diff(omega(:,2))], [0;
diff(omega(:,3))]]; % aceleracao angular
640
641     % Calcula a velocidade e aceleracao do CG
642     V_CG = [0; diff(saida_CG)];
643     A_CG = [0; diff(V_CG)];
644
645     % Calcula a velocidade e aceleracao do ponto de interesse
646     V = V_CG + cross(omega, r);
647     A = A_CG + cross(alpha, r) + cross(omega, cross(omega, r));
648
649     % Salva os vetores de aceleracao do ponto de interesse nos
respectivos arquivos
650     saida_A(:,1) = A(:,1);
651     saida_A(:,2) = A(:,2);
652     saida_A(:,3) = A(:,3);
653
654     % Salve em um arquivo txt para cada eixo
655     dlmwrite("dataset/" + "A_x_" + percurso + "_" + k + ".txt",
saida_A(:,1), 'delimiter', '\t');
656     dlmwrite("dataset/" + "A_y_" + percurso + "_" + k + ".txt",
saida_A(:,2), 'delimiter', '\t');
657     dlmwrite("dataset/" + "A_z_" + percurso + "_" + k + ".txt",
saida_A(:,3), 'delimiter', '\t');
658     dlmwrite("dataset/" + "labels_" + percurso + "_" + k + ".txt",
labels(:), 'delimiter', '\t');
659     end
660 end
```

APÊNDICE B – ROTINA COMPUTACIONAL PYTHON DE PRÉ-PROCESSAMENTO DOS DADOS DE SIMULAÇÃO

```

1 import torch
2 import pandas as pd
3 from sklearn.model_selection import train_test_split
4 from sklearn.preprocessing import StandardScaler
5 from torch.utils.data import DataLoader, TensorDataset
6
7 def load_data(window_size, file_count = 5, file_type = None,
8             initial_file = 1):
9     # The files to be loaded are of the format (A_(x|y|z)_(buraco|lombada|
10    sem buraco|sonorizador|tartaruga)_(1|2|...|50).txt)
11    # And labels are of the format (labels_(buraco|lombada|sem buraco|
12    sonorizador|tartaruga)_(1|2|...|50).txt)
13    # We must load all of them and concatenate each type of acceleration
14    into a single DataFrame
15
16    # Load data from all txt files (using loops)
17    for i in range(initial_file, file_count + initial_file):
18        for j in ['buraco', 'lombada', 'sem buraco', 'sonorizador', '
19        tartaruga']:
20            if file_type is None:
21                if i == initial_file and j == 'buraco':
22                    x_data = pd.read_csv('data/A_x_buraco_1.txt', header=None,
23                    names=['acceleration_x'])
24                    y_data = pd.read_csv('data/A_y_buraco_1.txt', header=None,
25                    names=['acceleration_y'])
26                    z_data = pd.read_csv('data/A_z_buraco_1.txt', header=None,
27                    names=['acceleration_z'])
28                    labels = pd.read_csv('data/labels_buraco_1.txt', header=None
29                    , names=['labels'])
30                else:
31                    x_data = pd.concat([x_data, pd.read_csv('data/A_x_' + j + '_'
32                    + str(i) + '.txt', header=None, names=['acceleration_x'])], axis=0)
33                    y_data = pd.concat([y_data, pd.read_csv('data/A_y_' + j + '_'
34                    + str(i) + '.txt', header=None, names=['acceleration_y'])], axis=0)
35                    z_data = pd.concat([z_data, pd.read_csv('data/A_z_' + j + '_'
36                    + str(i) + '.txt', header=None, names=['acceleration_z'])], axis=0)
37                    labels = pd.concat([labels, pd.read_csv('data/labels_' + j +
38                    '_' + str(i) + '.txt', header=None, names=['labels'])], axis=0)
39                elif file_type == j:
40                    if i == initial_file:
41                        x_data = pd.read_csv('data/A_x_' + j + '_1.txt', header=None
42                        , names=['acceleration_x'])

```

```
29         y_data = pd.read_csv('data/A_y_' + j + '_1.txt', header=None
30         , names=['acceleration_y'])
31         z_data = pd.read_csv('data/A_z_' + j + '_1.txt', header=None
32         , names=['acceleration_z'])
33         labels = pd.read_csv('data/labels_' + j + '_1.txt', header=
34         None, names=['labels'])
35     else:
36         x_data = pd.concat([x_data, pd.read_csv('data/A_x_' + j + '_'
37         + str(i) + '.txt', header=None, names=['acceleration_x'])], axis=0)
38         y_data = pd.concat([y_data, pd.read_csv('data/A_y_' + j + '_'
39         + str(i) + '.txt', header=None, names=['acceleration_y'])], axis=0)
40         z_data = pd.concat([z_data, pd.read_csv('data/A_z_' + j + '_'
41         + str(i) + '.txt', header=None, names=['acceleration_z'])], axis=0)
42         labels = pd.concat([labels, pd.read_csv('data/labels_' + j +
43         '_' + str(i) + '.txt', header=None, names=['labels'])], axis=0)
44
45     print('Loaded all data from files', x_data.shape, y_data.shape, z_data
46         .shape, labels.shape)
47
48     # Combine the X, Y, and Z data into a single DataFrame
49     data = pd.concat([x_data, y_data, z_data], axis=1)
50
51     print('Combined all data into a single DataFrame', data.shape)
52
53     # Normalize the data
54     scaler = StandardScaler()
55     data = scaler.fit_transform(data)
56
57     # Convert the data and labels to tensors
58     data = torch.tensor(data, dtype=torch.float32)
59     labels = torch.tensor(labels.values, dtype=torch.long) # Changed to
60     long because the labels are now integers
61
62     # Create windows of accelerometer readings
63     data = data.unfold(0, window_size, 1)
64
65     # Adjust the labels to match the windows
66     labels = labels[window_size - 1:]
67
68     print(data.shape, labels.shape)
69
70     if file_type is not None:
71         return DataLoader(TensorDataset(data, labels), shuffle=False)
72
73     # Split the data into a training set and a temporary set using an
74     70/30 split
75     data_train, data_temp, labels_train, labels_temp = train_test_split(
```

```
        data, labels, test_size=0.3)
66
67 # Split the temporary set into a validation set and a test set using a
    50/50 split
68 data_val, data_test, labels_val, labels_test = train_test_split(
    data_temp, labels_temp, test_size=0.5)
69
70 # Create PyTorch DataLoaders
71 train_loader = DataLoader(TensorDataset(data_train, labels_train),
    batch_size=64, shuffle=True)
72 val_loader = DataLoader(TensorDataset(data_val, labels_val),
    batch_size=64, shuffle=False)
73 test_loader = DataLoader(TensorDataset(data_test, labels_test),
    shuffle=False)
74
75 return train_loader, val_loader, test_loader
```

APÊNDICE C – ROTINA COMPUTACIONAL PYTHON DE APRENDIZADO DE MAQUINA

```

1 import torch
2 from torch import nn
3 from load import load_data
4 import matplotlib.pyplot as plt
5 import numpy as np
6 from sklearn.metrics import confusion_matrix
7 import seaborn as sns
8
9 device = torch.device('cpu')
10 torch.autograd.set_detect_anomaly(True)
11
12 # Define the model
13 class PotholeDetector(nn.Module):
14     def __init__(self, input_size, hidden_size, num_layers, num_classes,
15                 readings_window_size):
16         super(PotholeDetector, self).__init__()
17         self.conv1 = nn.Conv1d(input_size, readings_window_size, kernel_size
18                                =5, stride=1, padding=2)
19         self.relu = nn.ReLU()
20         self.gru = nn.GRU(readings_window_size, hidden_size, num_layers,
21                            batch_first=True)
22         self.fc = nn.Linear(hidden_size, num_classes) # Change the output
23                size to num_classes
24
25     def forward(self, x):
26         out = self.conv1(x)
27         # print('shape from conv', out.shape)
28         out = self.relu(out)
29         out, _ = self.gru(out)
30         # print('shape from gru', out.shape)
31         out = self.fc(out[:, -1, :])
32         # print('shape from fc', out.shape)
33         return out # Remove the sigmoid activation function
34
35 readings_window_size = 200
36 # Instantiate the model
37 model = PotholeDetector(input_size=3, hidden_size=50, num_layers=2,
38                          num_classes=5, readings_window_size=readings_window_size)
39 model = model.to(device)
40
41 # Define loss function and optimizer
42 criterion = nn.CrossEntropyLoss()

```

```
38 optimizer = torch.optim.Adam(model.parameters())
39
40 # Load data
41 train_loader, val_loader, test_loader = load_data(readings_window_size)
42
43 # Training loop
44 val_losses = []
45 num_epochs = 8
46 best_val_loss = float('inf')
47
48 # Get current date to calculate time it takes to train for each epoch
   and predict the expected time to train the whole model
49 import datetime
50 start_time = datetime.datetime.now()
51
52 for epoch in range(num_epochs):
53     # Training
54     model.train()
55     for i, (inputs, labels) in enumerate(train_loader):
56         optimizer.zero_grad()
57         outputs = model(inputs.to(device))
58         loss = criterion(outputs, labels.to(device).squeeze()) # Squeeze
   the labels tensor to match the output tensor's shape
59
60         loss.backward()
61         optimizer.step()
62
63     end_time = datetime.datetime.now()
64     time_elapsed = end_time - start_time
65     expected_time = time_elapsed * (num_epochs - epoch)/(epoch+1)
66
67     print(f'Epoch {epoch}/{num_epochs}, Time Elapsed: {time_elapsed},
   Expected Time: {expected_time}')
68
69 # Validation
70 model.eval()
71 val_loss = 0
72 with torch.no_grad():
73     for inputs, labels in val_loader:
74         outputs = model(inputs.to(device))
75         loss = criterion(outputs, labels.to(device).squeeze()) # Squeeze
   the labels tensor to match the output tensor's shape
76         val_loss += loss.item()
77
78 val_loss /= len(val_loader)
79
80 # Print loss every 10 epochs
```

```
81 # if epoch % 10 == 0:
82     print(f'Epoch {epoch}/{num_epochs}, Training Loss: {loss.item()},
83           Validation Loss: {val_loss}')
84
85     val_losses.append(val_loss)
86
87     # Save the model with the best validation loss
88     if val_loss < best_val_loss:
89         torch.save(model.state_dict(), 'pothole_detector_v2.pth')
90         best_val_loss = val_loss
91
92     # Display the graphs of the losses
93     plt.plot(val_losses, label='Validation Loss')
94     plt.legend()
95     plt.show()
96
97     # Evaluation
98     model.load_state_dict(torch.load('pothole_detector_v2.pth'))
99     model.eval()
100
101     with torch.no_grad():
102         for data_type in ['buraco', 'lombada', 'sem buraco', 'sonorizador', '
103             tartaruga']:
104             # If it has file type, load only that type of data and only returns
105             a test_loader
106             test_loader = load_data(readings_window_size, 2, data_type,
107                 initial_file=30)
108
109             correct = 0
110             total = 0
111             predictions, allLabels = [], []
112             for inputs, labels in test_loader:
113                 labels = labels.to(device)
114                 outputs = model(inputs.to(device))
115                 predicted = torch.argmax(outputs, dim=1)
116
117                 predictions.append(predicted.cpu().numpy().tolist())
118                 allLabels.append(labels.squeeze().cpu().numpy().tolist())
119
120                 total += labels.numel()
121                 correct += (predicted == labels.squeeze()).sum().item() # Squeeze
122                 the labels tensor here
123
124             accuracy = 100 * correct / total
125             # Plot the predictions vs the labels in a scatter plot (just 250
126             results)
127             predictLimit = 250
```



```
122
123     predictions = np.array(predictions).flatten()
124     allLabels = np.array(allLabels).flatten()
125
126     # Resize for smaller sample
127     # predictions = predictions[:predictLimit]
128     # allLabels = allLabels[:predictLimit]
129
130     print(predictions.shape, allLabels.shape)
131     x_axis_length = predictions.shape[0]
132
133     plt.figure()
134     plt.title('Grafico de dispersao - Precisao: ' + str(round(accuracy,
135     2)) + '%')
136     plt.scatter(np.arange(x_axis_length), predictions)
137     plt.scatter(np.arange(x_axis_length), allLabels, label='Inferencia =
138     Realidade')
139     # Show where the predictions and labels are different
140     red_flag = False
141     for i in range(x_axis_length):
142         if predictions[i] != allLabels[i]:
143             if not red_flag:
144                 plt.scatter(i, predictions[i], color='red', label="
145                 Inferencia != Realidade")
146                 red_flag = True # Set the flag to True after the first
147                 red point legend entry
148             else:
149                 plt.scatter(i, predictions[i], color='red')
150                 plt.scatter(i, allLabels[i], color='red')
151     plt.legend()
152     plt.savefig('./graphs/scatter_' + data_type + '.png')
153     plt.close() # Close the current figure
154
155     # Compute confusion matrix
156     cm = confusion_matrix(allLabels, predictions, labels=[0,1,2,3,4])
157
158     # Plot the confusion matrix
159     plt.figure(figsize=(10,7))
160     ax = sns.heatmap(cm, annot=True, fmt='g', cmap='Blues')
161     ax.set_xticks(np.arange(5))
162     ax.set_yticks(np.arange(5))
163     ax.set_xticklabels([0,1,2,3,4])
164     ax.set_yticklabels([0,1,2,3,4])
165     plt.xlabel('Inferencia')
166     plt.ylabel('Realidade')
167     plt.title('Matriz de confusao - Precisao: ' + str(round(accuracy, 2)
168     ) + '%')
```

```
164 plt.savefig('./graphs/confusion_matrix_' + data_type + '.png')
165 plt.close() # Close the current figure
166
167 # Print the accuracy
168 print(f'Accuracy for {data_type}: {accuracy}%')
```

APÊNDICE D – ELEMENTOS DA MATRIZ DE AMORTECIMENTO

$$c_{11} = 2(c_f + c_r) \quad (32)$$

$$c_{12} = c_{21} = b_1 c_f - b_2 c_f - b_1 c_r + b_2 c_r \quad (33)$$

$$c_{13} = c_{31} = 2(a_2 c_r - a_1 c_f) \quad (34)$$

$$c_{22} = b_1^2 + b_2^2 c_f + b_1^2 c_r + b_2^2 c_r \quad (35)$$

$$c_{32} = c_{23} = a_1 b_2 c_f - a_1 b_1 c_f - a_2 b_1 c_r + a_2 b_2 c_r \quad (36)$$

$$c_{33} = 2(a_1^2 c_f + a_2^2 c_r) \quad (37)$$

APÊNDICE E – ELEMENTOS DA MATRIZ DE RIGIDEZ

$$k_{11} = 2(k_f + k_r) \quad (38)$$

$$k_{12} = k_{21} = b_1 k_f - b_2 k_f - b_1 k_r + b_2 k_r \quad (39)$$

$$k_{13} = k_{31} = 2(a_2 k_r - a_1 k_f) \quad (40)$$

$$k_{22} = k_R + b_1^2 k_f + b_2^2 k_f + b_1^2 k_r + b_2^2 k_r \quad (41)$$

$$k_{32} = k_{23} = a_1 b_2 k_f - a_1 b_1 k_f - a_2 b_1 k_r + a_2 b_2 k_r \quad (42)$$

$$k_{42} = k_{24} = -b_1 k_f - \frac{1}{d_f} k_R \quad (43)$$

$$k_{52} = k_{25} = b_2 k_f + \frac{1}{d_f} k_R \quad (44)$$

$$k_{33} = 2(a_1^2 k_f + a_2^2 k_r) \quad (45)$$

$$k_{44} = k_f + k_{t_f} + \frac{1}{d_f^2} k_R \quad (46)$$

$$k_{55} = k_f + k_{t_f} + \frac{1}{d_f^2} k_R \quad (47)$$