

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA AUTOMOTIVA

VINICIUS TEODORO ELZINGA

ALGORITMO PARA TROCA DE MARCHAS EM POWERTRAIN AUTOMATIZADO
BASEADO EM REDE NEURAL

Joinville

2024

VINICIUS TEODORO ELZINGA

ALGORITMO PARA TROCA DE MARCHAS EM POWERTRAIN AUTOMATIZADO
BASEADO EM REDE NEURAL

Trabalho apresentado como requisito para obtenção do título de Engenheiro no Curso de Graduação em Engenharia Automotiva do Centro Tecnológico de Joinville da Universidade Federal de Santa Catarina.

Orientador(a): Dr. Sérgio Junichi Idehara

Joinville

2024

Elzinga, Vinícius Teodoro

Algoritmo para troca de marchas em powertrain
automatizado baseado em rede neural. / Vinícius Teodoro
Elzinga ; orientador, Sérgio Junichi Idehara, 2024.
66 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Joinville,
Graduação em Engenharia Automotiva, Joinville, 2024.

Inclui referências.

1. Engenharia Automotiva. 2. Cambio automatizado. 3.
Powertrain. 4. Inteligencia Artificial. 5. Rede
perceptron. I. Idehara, Sérgio Junichi. II. Universidade
Federal de Santa Catarina. Graduação em Engenharia
Automotiva. III. Título.

Vinícius Teodoro Elzinga

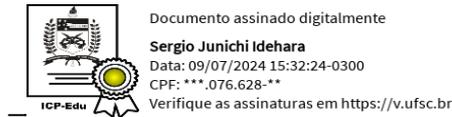
Título: Algoritmo para troca de marchas em powertrain automatizado baseado em rede neural.

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de Bacharel e aprovado em sua forma final pelo Curso Engenharia Automotiva.

Local Joinville, 05 de julho de 2024.

Coordenação do Curso de engenharia Automotiva

Banca examinadora



Prof. Sergio Junichi Idehara, Dr.
Universidade Federal de Santa Catarina
Orientador

Prof.(a) Lucas Weihmann, Dr.(a)
Universidade Federal de Santa Catarina
Membro

Renan Luís Knabben, Eng.
Pixida
Membro

Joinville, 2024.

Dedico este trabalho às famílias Elzinga e Riva, em especial aos meus pais
Léo Osmar Elzinga e Marivânia Riva Elzinga.

AGRADECIMENTOS

Primeiramente agradeço à Deus por me dar força, capacidade para chegar até aqui e por colocar em meu caminho milagres em forma humana.

Gostaria de agradecer o Professor Dr. Sérgio Junichi Idehara pelo conhecimento transmitido ao longo da graduação, por me orientar no desenvolvimento desta obra e pela paciência ao longo deste tempo.

Gratidão à minha mãe, Marivânia, por ter moldado a pessoa que sou hoje, pelo carinho, pelo amor, pelo apoio neste projeto que também é dela, pelos puxões de orelha e por ser a mãe que todo filho gostaria de ter. Agradeço ao meu pai, Léo, por me apoiar, pelos ensinamentos da vida, pelos conselhos, pela cobrança, por ser pai e amigo, por estar torcendo por mim, e espero que mesmo não estando mais nesse plano, ele sinta orgulho e alegria com a conclusão deste projeto, que também era dele.

Agradeço à minha Irmã Patrícia, meu irmão Lucas, e gostaria de agradecer especialmente meu irmão e colega de faculdade Arthur, pelo companheirismo, pelas conversas, desabafos, discussões calorosas, pelo auxílio na faculdade e na vida.

Agradeço à toda minha família pelo apoio, pelas orações e por acreditar no meu potencial. Gratidão em especial à minha madrinha Marcia, meu padrinho Paulo Ademir meu padrinho João Orlei e suas famílias, pelo apoio, carinho, pelos conselhos e pela inspiração.

Gratidão aos amigos e amigas que fiz ao longo da vida, em especial à Bruno e Gabriel, por desde o começo da graduação estarmos unidos, enfrentando as adversidades e comemorando as vitórias, na faculdade e na vida.

Agradeço à minha namorada Fernanda, pelo apoio, por estarmos juntos nos momentos bons e nas adversidades, por ser meu porto seguro e por me dar a visão de um futuro mais colorido.

Gostaria de agradecer aos meus conterrâneos da cidade de Entre Rios do Sul - RS, por atuarem ativamente na minha formação como estudante e como pessoa, desde o início da minha vida, pelo apoio e pela torcida.

Obrigado à comunidade acadêmica, servidores e demais profissionais que trabalham na UFSC do campus Joinville. Em especial agradeço à banca por prestigiar e auxiliar na construção deste trabalho.

Por fim, gostaria de agradecer a todos aqueles que acreditam no caminho da educação, e que de forma direta ou indireta atuaram na minha formação, como estudante, profissional, e principalmente como ser humano.

RESUMO

Este trabalho apresenta o desenvolvimento da modelagem numérica do powertrain automatizado de um veículo a combustão interna, através da ferramenta Matlab®. A partir dos dados obtidos através da modelagem numérica em diferentes condições, busca-se treinar uma rede neural *Perceptron*, assim como executar testes para verificar a capacidade de aprendizado e atuação como ferramenta de controle das trocas de marcha. Por fim, a rede *Perceptron* foi treinada a partir de dados experimentais, para verificar a precisão nas escolhas das marchas em uma situação prática. Para o processo de validação e testes em diferentes condições foram obtidas precisões de 97% e 98,25%. Para o processo utilizando dados numéricos, a precisão obtida foi de 94,71%

Palavras-chave: Powertrain. Caixa de transmissão automatizada. Rede neural.

ABSTRACT

This paper presents the development of numerical modeling of the automated powertrain of an internal combustion vehicle, using the Matlab® tool. Based on the data obtained through numerical modeling under different conditions, the aim is to train a Perceptron neural network, as well as to carry out tests to verify its learning capacity and its performance as a tool for controlling gear changes. Finally, the Perceptron network was trained using experimental data to check the accuracy of gear choices in a practical situation. For the validation process and tests in different conditions, accuracies of 97% and 98.25% were obtained. For the process using numerical data, the accuracy obtained was 94.71%

Keywords: Powertrain; Automated Transmission; Neural Network.

LISTA DE FIGURAS

Figura 1 - Diferencial para tração dianteira e motor transversal	20
Figura 2 - Neurônios de uma rede <i>Perceptron</i>	21
Figura 3 – Ilustração da rede neural <i>Perceptron</i>	23
Figura 4 - Modelo veicular simplificado com 2 rodas	25
Figura 5 - Curvas de Torque e Potência.....	29
Figura 6 - Curva de Torque	30
Figura 7 - Processo de treinamento e testes com dados numéricos	32
Figura 8 - Percurso de aquisição dos dados experimentais	33
Figura 9 - Processo de treinamento e teste com dados experimentais	34
Figura 10 - Curva de Torque	36
Figura 11 - Curvas de torque diversas.....	38
Figura 12 - Deslocamento obtido numericamente	39
Figura 13 - Velocidade obtida numericamente	39
Figura 14 – Valor do erro do teste 1 com torque máximo	41
Figura 15 - Valor do erro do teste 2 com torque máximo.....	41
Figura 16 - Valor do erro do teste 3 com torque máximo.....	41
Figura 17 - Valor do erro do teste 1 com curvas de torque diversas	42
Figura 18 - Valor do erro do teste 2 com curvas de torque diversas	42
Figura 19 - Valor do erro do teste 3 com curvas de torque diversas	43
Figura 20 - Valor do erro do teste 1 com dados com ruído	43
Figura 21 - Valor do erro do teste 2 com dados com ruído	44
Figura 22 - Valor do erro do teste 3 com dados com ruído	44
Figura 23 - Valores obtidos experimentalmente no primeiro trajeto.....	45
Figura 24 - Valores obtidos experimentalmente no segundo trajeto.....	46
Figura 25 - Valor do erro do teste 1 com dados experimentais	48
Figura 26 - Valor do erro do teste 2 com dados experimentais	48
Figura 27 - Valor do erro do teste 3 com dados experimentais	48

LISTA DE TABELAS

Tabela 1 - Dados Volkswagen Polo MPI 2018	28
Tabela 2 - Parâmetros para modelagem das resistências.....	36
Tabela 3 - Velocidades de troca de marcha	37
Tabela 4 - Precisão dos testes com curva de torque máximo	40
Tabela 5 - Precisão dos testes com curvas de torque diversas.....	42
Tabela 6 - Precisão dos testes com ruídos e dados aleatorizados.....	43
Tabela 7 - Precisão dos testes com dados experimentais.....	47

LISTA DE ABREVIATURAS E SIGLAS

ABBU	Associação Brasileira de Bibliotecas Universitárias Brasileiras
CAN	Controller Area Network
OBD2	On Board Diagnostic2
cv	Cavalo-Vapor
Nm	Newton-metro
RPM	Rotações por minuto

SUMÁRIO

1	INTRODUÇÃO	16
1.1	OBJETIVOS.....	17
1.1.1	Objetivo Geral	17
1.1.2	Objetivos Específicos	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	TRANSMISSÃO DE POTÊNCIA EM VEICULOS	18
2.1.1	Transmissão automatizada	18
2.2	DIFERENCIAL E RELAÇÃO FINAL.....	20
2.3	REDE <i>PERCEPTRON</i>	21
3	METODOLOGIA	23
3.1	REDE NEURAL PERCEPTRON BACKPROPAGATION.....	23
3.2	SIMULAÇÃO DA DINÂMICA VEICULAR (LONGITUDINAL).....	24
3.3	MATLAB.....	27
3.3.1	Integradores	27
3.3.1.1	<i>ODE45</i>	27
3.4	DESENVOLVIMENTO DA MODELAGEM NUMÉRICA.....	28
3.4.1	Dados do veículo	28
3.4.2	Curvas de torque	29
3.4.3	Integração numérica	30
3.5	TREINAMENTO COM DADOS NUMÉRICOS	31
3.6	TREINAMENTO COM DADOS EXPERIMENTAIS	32
4	APRESENTAÇÃO DOS RESULTADOS	35
4.1	TREINAMENTO COM DADOS NUMÉRICOS	35
4.1.1	FUNÇÃO MODELO	35
4.1.2	Modelagem e simulação com curva de torque máximo	36
4.1.3	Modelagem e simulação com curvas de torque diversas	37
4.1.4	Preparação dos dados numéricos	39
4.1.5	Treinamento da rede neural com dados numéricos	40
5	TREINAMENTO COM DADOS EXPERIMENTAIS	45
5.1.1	Preparação dos dados experimentais	46
5.1.2	Treinamento da rede neural com dados experimentais	47

6	CONCLUSÃO	49
6.1	ESTUDOS FUTUROS	50
	REFERÊNCIAS	52
	ANEXO A – ALGORITMOS	55

1 INTRODUÇÃO

De acordo com Custódio (2021), até 2030, 90% dos veículos novos comercializados no Brasil contarão com transmissão automática. O aumento nas vendas é resultado da procura por veículos com maior conforto no trânsito congestionado das grandes cidades, a desmistificação da manutenção e a otimização das transmissões automáticas e automatizadas, principalmente quanto ao consumo de combustível, e tem como reflexo o fato de que 63,5% dos automóveis emplacados entre janeiro e maio de 2022 contam com transmissão automática (Miragaya, 2022).

Segundo Pineau (1982), conforto é o fator que está associado a todos os aspectos que contribuem para o bem-estar e conveniência. O conforto é um dos fatores que mais impactam na compra de um automóvel, estando em 4º lugar, seguindo o preço de compra, consumo de combustível e segurança (Santana, 2021). Segundo Gama (2023), entre os anos de 2017 e 2023 o custo médio de um veículo de passeio zero quilômetro passou de R\$ 70.877,00 para R\$ 140.150,00, aumento justificado pelas consequências da pandemia de COVID-19, além da ocorrência de fenômenos de mercado e conflitos.

Sistemas de transmissão manual automatizada têm como objetivo combinar a eficiência das transmissões manuais com a praticidade das transmissões automáticas (Bertsche; Et Al., 2011). Os sistemas automatizados exigem menores esforços mentais e físicos por parte do motorista ao eliminar a necessidade de acionar mecanicamente a embreagem e a alavanca da transmissão, reduzindo a fadiga e permitindo uma maior concentração ao trânsito e a condução do veículo. (Genta; Morello, 2009).

Entretanto, transmissões automatizadas convencionais apresentam a ocorrência de trancos e atrasos, fenômenos que ocorrem devido à má operação, método de calibração do sistema de transmissão e falhas no software. (Kuchenbecker, 2021). O comportamento instável gera vibrações de baixa frequência, que em uma faixa entre 4 e 20 Hz causa náuseas e êmese (Lida, 2005).

O presente trabalho apresenta uma proposta de algoritmo para o controle das trocas de marchas, baseado em informações coletadas experimentalmente para um veículo manual da linha leve, através da comunicação CAN bus, via porta OBDII e

simulação numérica para um veículo com powertrain manual automatizado, em condições semelhantes às dos testes experimentais.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Este trabalho tem como objetivo modelar e implementar o comportamento dinâmico de um veículo com powertrain automatizado convencional, a partir de um algoritmo de inteligência artificial.

1.1.2 Objetivos Específicos

- Desenvolver um modelo numérico que simula a dinâmica de um veículo com powertrain automatizado convencional através da ferramenta Matlab ®;
- Executar testes práticos para obtenção de dados sobre a dinâmica de um veículo com transmissão manual para serem utilizados no modelo numérico;
- Implementar um algoritmo de inteligência artificial para as trocas de marchas;
- Treinar a rede neural baseado nos testes experimentais.

2 FUNDAMENTAÇÃO TEÓRICA

A compreensão sobre os parâmetros e teorias utilizados é fundamental para o desenvolvimento do modelo numérico capaz de simular de forma adequada o powertrain automotivo. Este capítulo apresenta os conceitos sobre os elementos que constituem uma transmissão automatizada convencional, alternativas adotadas atualmente para contornar problemas apresentados pelos modelos e teorias que embasam o desenvolvimento do modelo numérico do powertrain automotivo.

2.1 TRANSMISSÃO DE POTÊNCIA EM VEÍCULOS

De acordo com Genta; Morello (2009), motores alternativos de combustão interna são caracterizados pela geração de trabalho útil em função da rotação do eixo do motor, onde o torque gerado apresenta diferentes valores em função da velocidade angular do eixo virabrequim. A escolha de um motor ocorre em função da potência a qual o motor é capaz de gerar, e que deve estar disponível em qualquer velocidade do veículo (Genta; Morello, 2009).

Motores de combustão interna apresentam um limite de rotação mínimo, definido pelo tempo do ciclo de combustão e pela inércia do sistema mecânico (Genta; Morello, 2009). Em situações em que é necessária a interrupção na transmissão de força, como quando o veículo está parado ou na troca entre relações de transmissão no uso do veículo, é utilizada a embreagem, elemento que promove o desacoplamento entre motor e transmissão quando acionado (Bosch, 2005).

Segundo Bosch (2005), transmissões automotivas atuam fornecendo o torque gerado pelo motor para as rodas de tração, induzindo o movimento do veículo. A transmissão é responsável por adaptar o torque disponibilizado pelo motor à demanda exercida pela força trativa nas rodas (Bosch, 2005).

2.1.1 Transmissão automatizada

No sistema de transmissão manual, a escolha da relação de transmissão acionada em cada situação é feita pelo motorista, que faz a seleção da relação de transmissão através de uma alavanca, que transmite a força e o movimento

realizados pelo motorista ao interior da transmissão para acionar os mecanismos de sincronização e acoplamento das engrenagens da transmissão (Barbosa, 2007). Segundo Crolla (2009), a transmissão manual tem grande eficiência na transmissão de força, custo de produção e manutenção relativamente menor, peso até 50% menor em comparação à transmissão automática e tamanho relativamente menor. Em contrapartida, o controle do consumo de combustível e emissão de poluentes são influenciados pelo modo de direção do motorista, o acionamento da embreagem e da transmissão são mais exaustivos principalmente em situações de trânsito pesado (Crolla, 2009).

Transmissões automáticas convencionais são caracterizadas pela combinação de um conversor de torque, em geral acionado hidraulicamente, e um sistema de engrenagens epicicloidais para a transmissão mecânica da força (Crolla, 2009). Este sistema de transmissão automotiva tem como vantagem trocas de marchas mais suaves, dado que não é necessário o desacoplamento da transmissão (Bosch, 2005). Entretanto, transmissões automáticas convencionais tem custos de produção e manutenção mais altos, maior peso, maiores dimensões e menor eficiência quando comparadas aos sistemas manuais (Crolla, 2009).

Transmissões veiculares automatizadas são sistemas derivados das transmissões manuais, e herdam os componentes mecânicos das mesmas, com exceção dos mecanismos de seleção da velocidade e acionamento da embreagem. As decisões a respeito da troca de velocidades ficam a cargo do sistema (Genta; Morello, 2009). Sistemas automatizados se diferenciam dos totalmente automáticos pois precisam desacoplar da fonte de potência durante as trocas de marchas (Crolla, 2009).

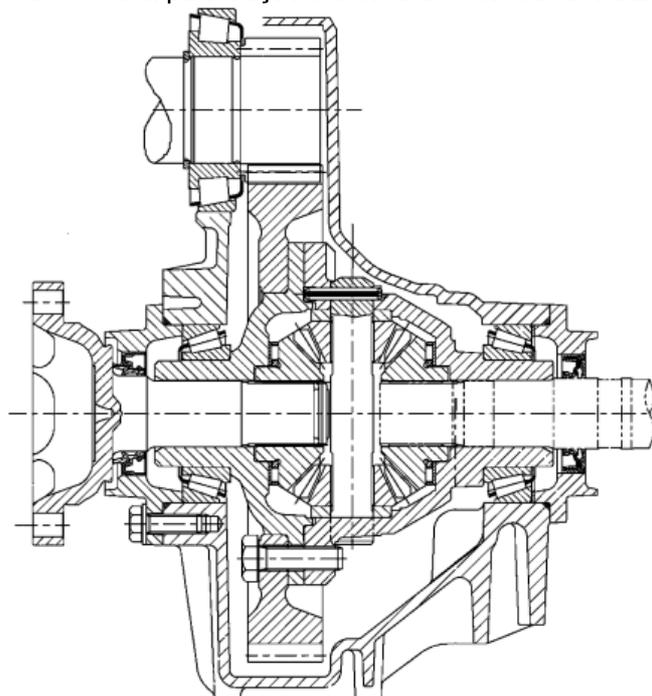
Para ocorrer a interrupção no torque durante as trocas de marcha, é adotada a embreagem, onde o acionamento deste componente é realizado de forma automatizada através de sistemas elétricos e hidráulicos, fazendo com que o processo se torne mais confortável para o motorista (Crolla; Mashasdi, 2012). A transmissão automatizada tem como vantagens a alta eficiência na transmissão de força, baixo custo e a tecnologia empregada no sistema de transmissão de força é similar aos modelos de transmissão manual (Bosch, 2005). Entretanto, a conversão de uma caixa de transmissão manual para automatizada, requer um algoritmo que preveja o momento em que as marchas devem ser trocadas.

Alguns dos modelos comercializados no mercado nacional de automóveis leves são o Dualogic/GSR da Fiat, I-motion da Volkswagen, EasyR da Renault e Easytronic da GM (Fortunatti, 2023).

2.2 DIFERENCIAL E RELAÇÃO FINAL

Diferencial é o sistema responsável por compensar a diferença entre as rotações das rodas interna e externa do eixo de tração durante as curvas, assim como controlar a taxa de torque fluindo para as rodas, em situações em que uma das rodas está em condições que dificultam a movimentação, fazendo com que o torque seja direcionado para a roda que gira com maior facilidade (Bosch, 2005). Segundo Genta e Morello (2009), geralmente, para veículos com tração dianteira, um trem de engrenagens de relação fixa denominada de relação final integra o sistema de diferencial. Essa relação tem grande importância na dinâmica longitudinal e influencia a escolha da marcha, baseado na carga sobre o veículo, pois faz parte da relação total de transmissão do trem de potência veicular. A Figura 1 apresenta a representação em corte de um diferencial para um veículo com tração dianteira e motor transversal.

Figura 1 - Diferencial para tração dianteira e motor transversal



Fonte: Genta; Morello (2009)

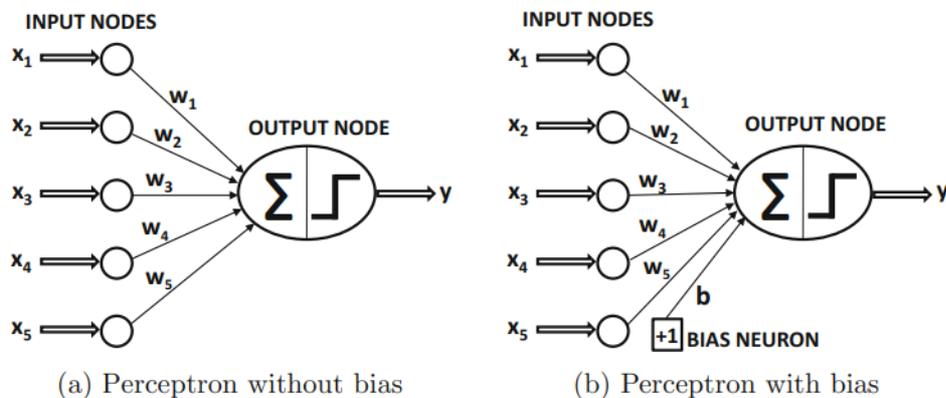
2.3 REDE PERCEPTRON

De acordo com Aggarwal (2018), uma rede neural é um método de aprendizagem de máquina inspirado no processo de aprendizagem do cérebro humano, onde a conexão entre as células básicas, denominada nós ou neurônios, depende de estímulos externos. A conexão entre nós ocorre de forma direta, apresentando diferentes pesos, que são informações utilizadas para resolver um problema (Faussett, 1993).

Quando uma informação rotulada é inserida, a rede neural registra as conexões estabelecidas entre nós, como resposta ao estímulo. Ao inserir consecutivas vezes informações semelhantes, a máquina ajusta as conexões entre nós, de acordo com os padrões presentes nas informações de entrada estabelecendo um padrão que é ativado em uma situação preditiva, como a tomada de decisão em determinada situação. (Aggarwal, 2018).

A arquitetura de uma rede neural apresenta diversos formatos, de acordo com a demanda computacional para a resolução de um problema, tendo seus nós subdivididos em camadas implícitas. A rede neural mais simples é denominada *Perceptron*, que admite uma camada de informações de entrada armazenadas e processadas em um único nó de saída, como representado na Figura 2 (Aggarwal, 2018).

Figura 2 - Neurônios de uma rede *Perceptron*



Fonte: Aggarwal, 2018

Taxa de aprendizagem e o momento são parâmetros que variam entre 0 e 1 utilizados para controlar a precisão e velocidade com a qual as iterações ocorrem na rede. A taxa de aprendizagem afeta o peso de cada neurônio nas iterações, a cada iteração. Uma taxa de aprendizagem mais baixa reduz a variação dos pesos entre iterações, tornando a rede mais estável, ao custo de uma convergência mais lenta (Haykin, 2000).

O aumento no valor da taxa de aprendizagem acelera o desenvolvimento dos cálculos, porém, pode tornar a rede instável (Haykin, 2000) O termo de momentum é adotado para filtrar variações altas entre iterações, possibilitando o uso de uma taxa de aprendizagem mais alta (Rumelhart, 1987). Além disso, a adoção do momento reduz a possibilidade de o processo de aprendizagem convergir para um ponto de mínimo (Haykin, 2000).

A rede neural do tipo *Perceptron* é empregada neste trabalho como controle do momento de troca de marchas.

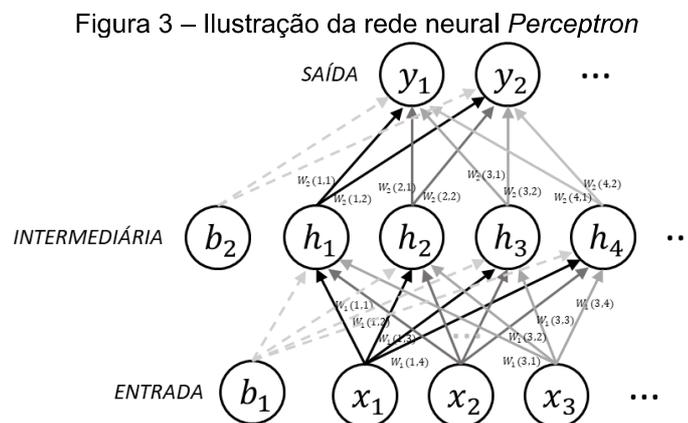
3 METODOLOGIA

Esta seção detalha os métodos utilizados para o desenvolvimento desta obra. Com base na segunda Lei de Newton, foi desenvolvida a modelagem do veículo, em função da força trativa e das resistências atuantes. Utilizando os dados numéricos e experimentais, foram realizados treinamentos e testes de uma rede *Perceptron*, a fim de verificar a capacidade de aprendizagem e precisão na replicação dos padrões aprendidos.

Neste capítulo é apresentado o software Matlab e as ferramentas utilizadas para a integração numérica, utilizados para implementação do modelo e simulação numérica.

3.1 REDE NEURAL PERCEPTRON BACKPROPAGATION

A rede neural do tipo *Perceptron* utilizada neste trabalho tem uma camada de entrada, uma intermediária e uma de saída, como ilustrada na Figura 3. Assim, a camada de entrada e saída têm o mesmo número de neurônios que as variáveis de entrada e saída, respectivamente. Na camada intermediária, o número de neurônios é definido como a soma do número de neurônios da entrada e saída. Os pesos entre a camada de entrada e intermediária (W_1 e b_1) e entre a camada intermediária e a saída (W_2 e b_2) são definidos aleatoriamente por uma distribuição uniforme no intervalo definido por padrão no algoritmo, entre -0,05 e 0,05, como dado inicial do treinamento.



Fonte: Autor, 2024

O cálculo do vetor de saída (y_i), a partir da saída da camada intermediária (h_j), é dado por:

$$y_i = \text{sigmoide} \left\{ \sum_j (W_{2j,i} \cdot h_j) + b_2(i) \right\} \quad (1)$$

Enquanto a camada intermediária é dada pelos dados de entrada, como:

$$h_i = \text{sigmoide} \left\{ \sum_j (W_{1j,i} \cdot h_j) + b_1(i) \right\} \quad (2)$$

Os pesos $W_{1j,i}$ e $W_{2j,i}$ são os coeficientes que multiplicam os valores da camada anterior, somando entre si e somado a um termo independente b_1 e b_2 , para determinar os termos da camada seguinte. Ao valor de saída do neurônio é aplicado a função de ativação do tipo sigmoide (logística), definida matematicamente como:

$$\text{sigmoide}(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

O cálculo dos coeficientes é feito pelo método *Back Propagation*, baseado na determinação do gradiente da função de erro, gerado a cada iteração do ajuste dos parâmetros.

3.2 SIMULAÇÃO DA DINAMICA VEICULAR (LONGITUDINAL)

Segundo Genta e Morello (2009), a resistência imposta a um veículo pode ser obtida através de uma análise simplificada, desprezando os efeitos verticais da suspensão e dos pneus, partindo da equação (4):

$$m \cdot \ddot{x} = \sum_{\forall i} F_{xi} \quad (4)$$

onde,

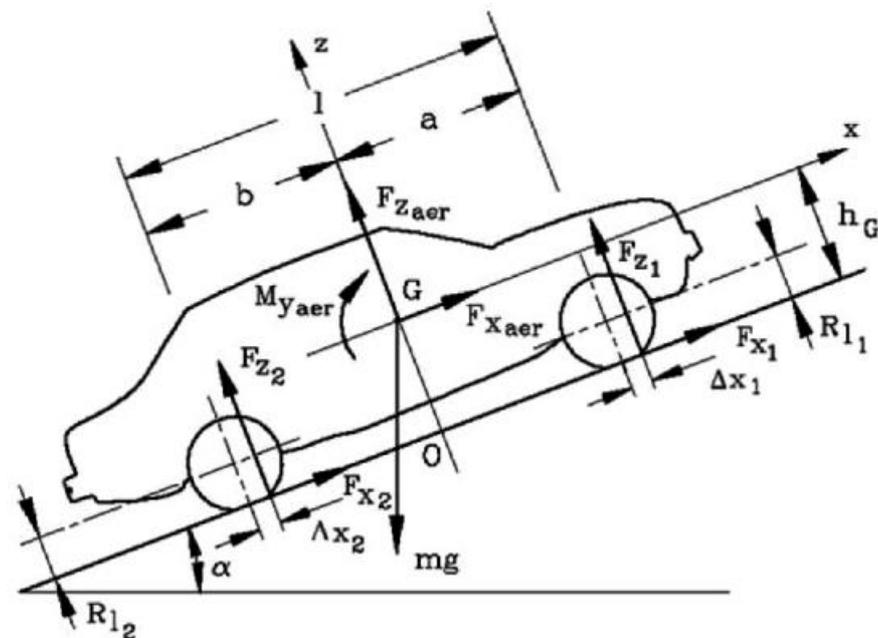
m = massa do veículo;

\ddot{x} = aceleração longitudinal;

F_{xi} = Resistências e tração atuando no veículo.

Veículos com mais de 3 rodas são consideradas estaticamente indeterminados, tendo a distribuição de carga para o solo definida pelas características da suspensão. Se o veículo apresenta simetria em relação ao plano longitudinal, é razoável utilizar a representação com apenas 2 rodas, como uma viga bi apoiada, o que torna o sistema estaticamente determinado (Genta, Morello, 2009). A Figura 4 mostra o modelo de duas rodas, com a representação que atuam no mesmo.

Figura 4 - Modelo veicular simplificado com 2 rodas



Fonte: Genta; Morello (2009)

A equação (5) mostra a relação entre aceleração, massa, forças e resistências atuando em um veículo (Wong, 2008). Segundo Wong (2008), em um terreno plano, sem a resistência gerada por reboque, as principais resistências ao movimento do veículo são a resistência aerodinâmica e resistência ao rolamento.

$$m \cdot \frac{d^2x}{dt^2} = \frac{W}{g} a = F_f + F_r - R_a - R_{rf} - R_{rr} - R_d - R_g \quad (5)$$

sendo,

$$\frac{d^2x}{dt^2} = a = \text{aceleração instantânea};$$

m = Massa do veículo;

W = Força peso do veículo;

F_f = Força trativa nos pneus dianteiros;

F_r = Força trativa nos pneus traseiros;

R_a = Resistencia aerodinâmica;

R_{rf} = Resistencia ao rolamento nos pneus dianteiros;

R_{rr} = Resistência ao rolamento nos pneus traseiros;

R_d = Resistencia gerada por reboque;

R_g = Resistencia de aclave;

g = aceleração da gravidade.

Para o desenvolvimento da análise, algumas das forças que ocorrem nos pneus podem ser consideradas como uma única força. Portanto, considerou-se para o desenvolvimento do modelo a união das forças tratativas no pneu dianteiro e traseiro como uma única força. Para a resistência ao rolamento foi adotada a mesma consideração. A modelagem do veículo considerou um percurso em terreno plano, portanto, a resistência ao aclave tem valor nulo.

A força de arrasto aerodinâmico atuando no veículo pode ser obtida pela equação (6), onde ρ é a densidade do ar, C_d o coeficiente de arrasto aerodinâmico, A_f a área frontal projetada do veículo, $V_x = \dot{x}$ é a velocidade do veículo na direção x e V_{wind} a velocidade do vento (Rajamani, 2005).

$$F_{aero} = \frac{1}{2} \cdot \rho \cdot C_d \cdot A_f \cdot (V_x + V_{wind})^2 \quad (6)$$

A resistência ao rolamento é determinada através da equação (7), relação dada em função do quadrado da velocidade do veículo (Genta; Morello, 2009).

$$R_r = m \cdot g \cdot (f_o + K \cdot V^2) \quad (7)$$

Onde, V é a velocidade, f_o é o coeficiente de resistência ao rolamento em repouso, m a massa, g a aceleração da gravidade e K o coeficiente de resistência ao rolamento.

3.3 MATLAB

MATLAB é a abreviatura para Matrix Laboratory, sendo este um software que inicialmente visava a resolução de problemas envolvendo matrizes, e ao longo dos anos foi aprimorado, tornando-se um programa otimizado para resolução de problemas matemáticos científicos e de engenharia (Chapman, 2016).

O software MATLAB implementa uma linguagem de programação procedimental de mesmo nome, onde os procedimentos a serem realizados para a resolução de um problema são efetivamente instruções (Chapman, 2016). A estrutura da MATLAB, faz com que esta linguagem de programação se assemelhe com outras linguagens populares, como C, Basic, Fortran e Pascal (Chapman, 2016). A MATLAB é uma linguagem de programação de quarta geração, voltada para o trabalho com conjuntos de dados ao invés de bits e bytes, facilitando o desenvolvimento do algoritmo e permitindo maior foco no desenvolvimento das tarefas e projetos (Sizemore; Mueller, 2016).

3.3.1 Integradores

MATLAB tem um vasto conjunto de ferramentas para a resolução de sistemas equações ordinárias de primeira ordem, conhecidas como funções ODE. Estas funções podem ser classificadas em duas categorias. As funções para problemas de valor inicial, são utilizadas quando os valores iniciais são previamente conhecidos, já as funções para problemas de valor limite, devem ser utilizadas em casos em que as condições do problema são conhecidas em pontos específicos (Bober, 2014).

3.3.1.1 ODE45

ODE45 é uma função presente no Matlab utilizada para realizar a integração de Equações diferenciais ordinárias. Se trata de um método iterativo, que resolve a equação ordinária e estima o erro para ajustar o intervalo de tempo para a próxima

iteração, através do uso simultâneo de fórmulas de Runge-Kutta de 4 e 5 ordem (Chapra, 2013).

O algoritmo permite a resolução de equações ordinárias onde a demanda de precisão não é extrema. (Dormand; Price, 1980)

3.4 Desenvolvimento da modelagem numérica

O presente tópico descreve a aquisição dos dados do veículo, curva de torque do motor e parâmetros utilizados para determinar as resistências ao movimento, utilizados para o processo de simulação numérica.

3.4.1 Dados do veículo

O Volkswagen Polo é um modelo Hatch de produção nacional reinserido pela marca no mercado no ano de 2017, ocupando uma posição de mercado um pouco acima do Volkswagen Gol, apresentando um valor de mercado relativamente baixo, e conta com alguns itens e características presentes em modelos mais sofisticados da marca (Motor1, 2018). O Volkswagen Polo utilizado como referência para o desenvolvimento do modelo é a versão MPI, com motor tri cilíndrico com capacidade volumétrica de 1000cm^3 e câmbio manual (Nasser, 2017). Quando abastecido com gasolina, o modelo gera 75 cv ou 55 kW de potência à 6250 RPM e torque de 95 Nm à 3000 RPM. Utilizando etanol, os valores de potência e torque passam a ser 84 cv ou 62 kW à 6250 rpm, e 102 Nm à 3000 RPM, respectivamente. As informações sobre o veículo e valores utilizados para desenvolvimento da modelagem das forças são apresentadas na Tabela 1.

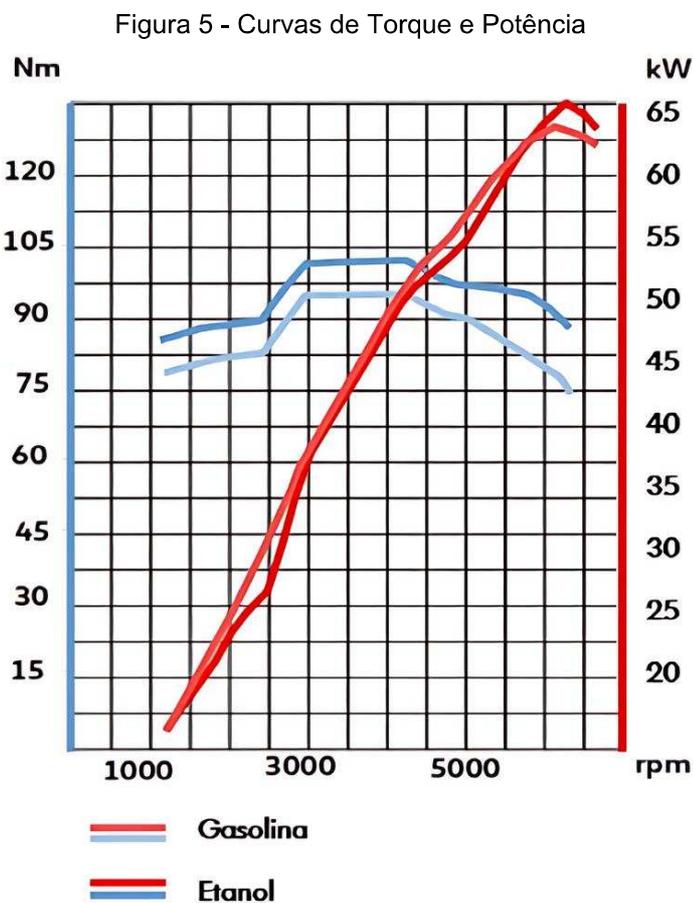
Tabela 1 - Dados Volkswagen Polo MPI 2018

Peso em ordem de marcha [kg]	1058				
Relação do diferencial	4,929:1				
Relações de transmissão	1°: 4,167	2°: 2,300	3°: 1,430	4°: 0,975	5°: 0,776
Area frontal [m ²]	18				
Coefficiente aerodinâmico	62				
Pneus	185/65 R15				

Fonte: Manual de instruções (2017)

3.4.2 Curvas de torque

A curva de torque utilizada como base para o desenvolvimento da modelagem numérica é apresentada na Figura 5 (Volkswagen, 2018). Para o desenvolvimento do modelo, foi utilizado como base a curva de torque para o motor abastecido com gasolina.



Fonte: Volkswagen (2018)

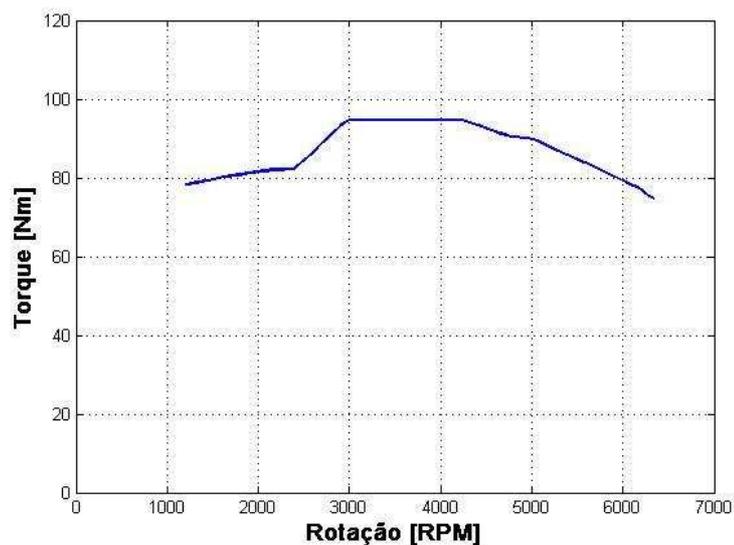
A conversão do diagrama apresentado na imagem em dados numéricos foi feita através da ferramenta Paint. Por padrão, o Paint utiliza a unidade de medida pixel para definir medidas e posições em uma imagem, portanto, alguns pontos como os valores de rotação e torque nos eixos foram marcados e tiveram sua posição definida, assim como foram definidos pontos sobre a curva, tendo sua localização identificada a fim de definir valores posteriormente.

Inicialmente foram marcadas e identificadas as posições dos valores de referência representados eixos de torque e rotação do gráfico. Para os valores no eixo das rotações, foram obtidas as posições horizontais para o desenvolvimento da curva, e para os valores no eixo do torque as posições verticais.

Ao longo da curva de torque foram destacados 37 pontos, onde o posicionamento dos pontos foi tomado para definir os valores de torque e rotação correspondentes. Cada ponto sobre a curva gerou dois valores de posicionamento, correspondentes as coordenadas vertical e horizontal em pixels.

Os valores de torque e rotação foram definidos através de interpolação linear entre as posições dos pontos destacados sobre a curva e os valores e posições nos eixos de torque e potência. A Figura 6 apresenta a curva de torque obtida.

Figura 6 - Curva de Torque



Fonte: Autor, 2024

A marcha lenta do Volkswagen Polo ocorre na rotação de 980 RPM, o que pode ser percebido pela ausência de valores de torque nas rotações iniciais do diagrama apresentado.

3.4.3 Integração numérica

A integração da função modelo foi feita utilizando a função ODE45, implementada nativamente no Matlab. A integração da função modelo retornou

valores de posição e velocidade do veículo para o intervalo de tempo utilizado na simulação, que serão apresentados nos próximos capítulos.

Para a realização das simulações foi definido o intervalo de tempo de 0 a 25 segundos, sendo segmentado em intervalos de 0,01 segundo. O processo de integração numérica foi realizado para obtenção dos dados em cada um dos seis casos, com as diferentes curvas de torque estipuladas.

Determinados os valores de velocidade em cada instante de tempo no intervalo, são feitos os cálculos para determinar os valores de rotação do motor e a marcha a qual o veículo está. Os valores de rotação e a marcha do veículo em cada instante de tempo foram determinados fora da função modelo, pois para determinar os valores de velocidade, o integrador estipula valores diferentes de tempo, gerando uma série de resultados correspondentes as iterações realizadas pelo algoritmo para convergir nos intervalos de tempo de integração definidos.

3.5 TREINAMENTO COM DADOS NUMÉRICOS

Antes do processo de aprendizado, foi necessário o preparo e normalização dos dados numéricos. O conjunto de dados utilizados consiste nos valores de velocidade do veículo, em metros por segundo, rotações do motor em rotações por minuto e o valor de torque do motor em Newton-metro como dados de entrada. A marcha acionada no instante de tempo foi definida e utilizada como informação de saída para o treinamento da rede neural. A informação de marcha é dada na forma de um vetor de cinco posições, onde a posição na qual a marcha está acionada apresenta o valor um, e as demais posições apresentam o valor zero.

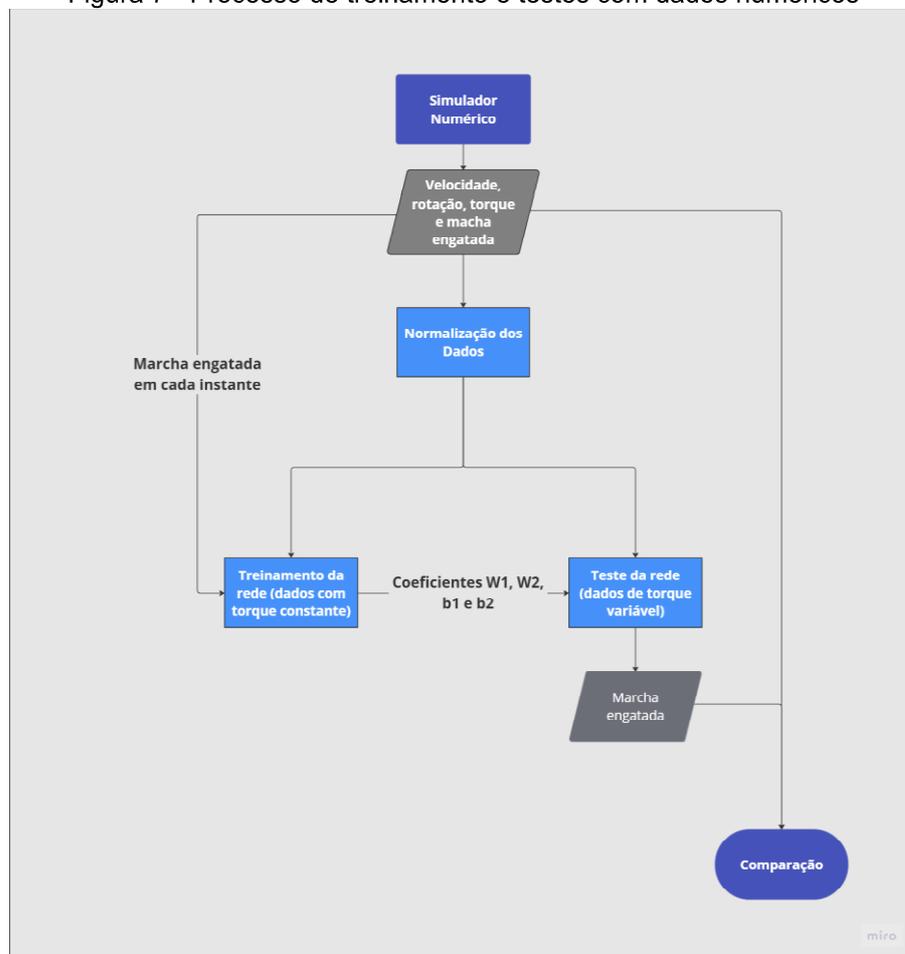
A normalização dos dados de velocidade, rotação e torque do motor é realizada utilizando como base os respectivos valores máximos obtidos na simulação desenvolvida com a curva de torque máximo.

Para o desenvolvimento deste trabalho, foi utilizado um modelo de rede *Perceptron* implementado em Matlab (Stojkovic, 2012). A realização do treinamento e testes da rede a partir dos resultados das simulações numéricas pode ser dividido em duas etapas. O fluxograma da Figura 7 apresenta os processos realizados para preparo dos dados e desenvolvimento dos testes e treinamentos com dados numéricos

Inicialmente, a rede neural foi treinada e testada utilizando os dados obtidos a partir da simulação numérica com base no modelo que utiliza a curva de torque máximo do veículo.

A segunda etapa consistiu em treinar a rede com dados da simulação numérica realizados utilizando curvas de torque constante com valores inferiores aos da curva de torque máximo. Utilizando os valores dos pesos e bias obtidos no treinamento, foi realizado o teste utilizando os dados obtidos nas simulações realizadas com curvas de torque variáveis, com valores de torque oscilando entre os valores de torque constante, utilizado para o treinamento da rede.

Figura 7 - Processo de treinamento e testes com dados numéricos



Fonte: Autor, 2024

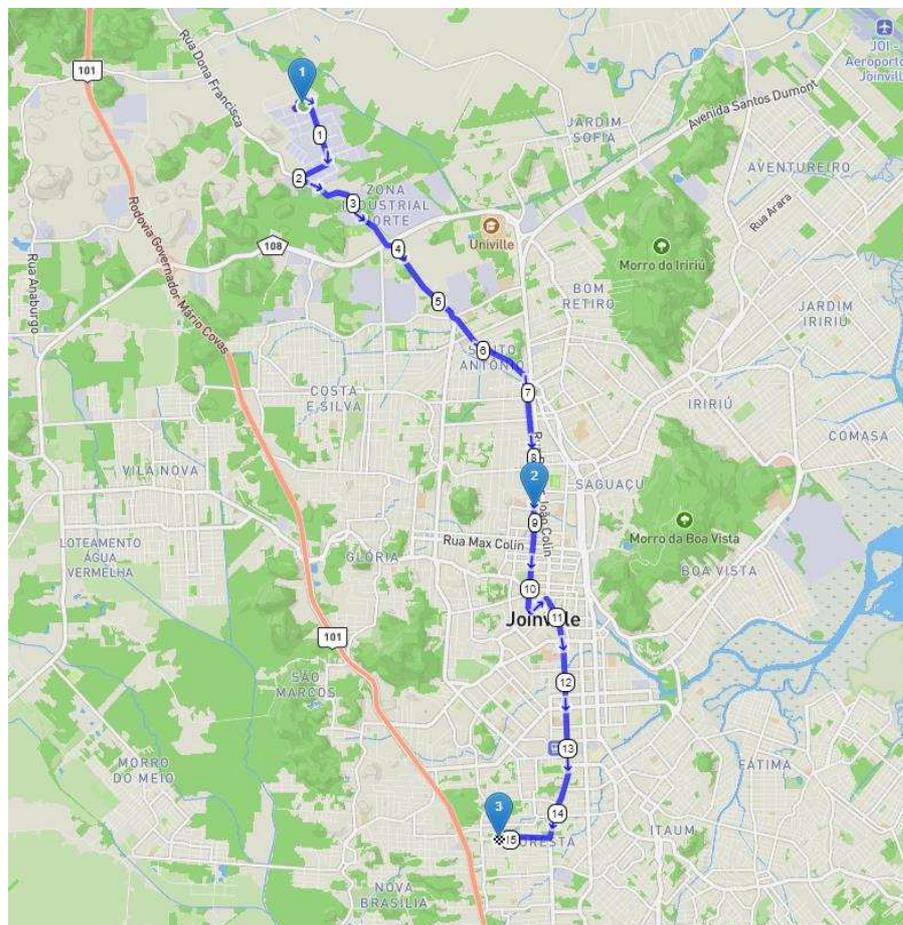
3.6 TREINAMENTO COM DADOS EXPERIMENTAIS

Após verificar a capacidade de aprendizado e replicação dos padrões, a rede neural foi treinada utilizando dados reais, obtidos experimentalmente. Os dados

experimentais utilizados foram coletados para o desenvolvimento da obra de Costa (2019), a partir de um veículo da linha leve na cidade de Joinville.

Para a aquisição dos dados, o veículo de testes percorreu um trajeto urbano, apresentado na Figura 8 (Costa, 2019). Em sua tese Costa (2019) dividiu o circuito percorrido em duas partes, gerando assim dois conjuntos de dados. Para o desenvolvimento da análise apresentada nesta obra, o primeiro conjunto de dados foi utilizado para treinamento da rede neural, e o segundo conjunto de dados foi utilizado para o testar a precisão da rede neural após o treinamento.

Figura 8 - Percurso de aquisição dos dados experimentais



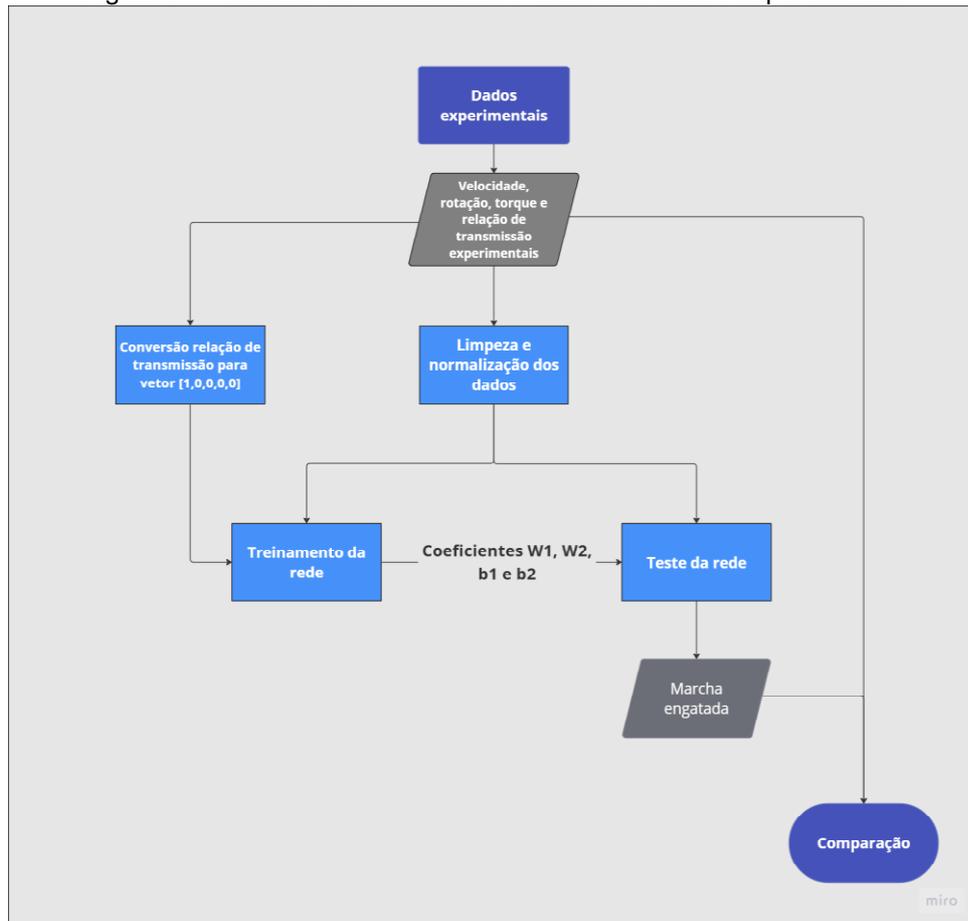
Fonte: Costa, (2019)

A aquisição das informações do veículo durante os testes foi feita utilizando um Arduino uno conectado à porta OBDII (Costa; 2019). O treinamento da rede neural utilizou os dados experimentais de velocidade do veículo, em metros por segundo, rotação do motor, em rotações por minuto, valores das estimativas de

torque do motor realizadas pela central do veículo, em Newton-metro e os valores os valores da relação total de transmissão da marcha acionada em cada instante de tempo.

A Figura 9 mostra o processo de preparo dos dados, treinamento e teste da rede, utilizando os dados experimentais.

Figura 9 - Processo de treinamento e teste com dados experimentais



Fonte: Autor, 2024

4 APRESENTAÇÃO DOS RESULTADOS

Esta seção apresenta os resultados obtidos no desenvolvimento da modelagem numérica de um veículo, treinamento e teste com dados numéricos e experimentais.

Ao longo do capítulo, são apresentados os valores obtidos através da simulação numérica, que foram utilizados como dados de entrada e saída no processo de treinamento da rede neural. Também são apresentados os dados utilizados para avaliar o desempenho da rede *Perceptron*, como a precisão obtida nos testes da rede e a evolução do erro ao longo das iterações durante o processo de aprendizado da rede.

4.1 TREINAMENTO COM DADOS NUMÉRICOS

Inicialmente, os dados utilizados para teste da rede *Perceptron* foram obtidos através de simulação numérica, utilizando como base o modelo Volkswagen Polo 2018 1.0 MPI.

4.1.1 FUNÇÃO MODELO

Os valores de velocidade e deslocamento foram obtidos a partir da integração de uma função onde a partir de dados do modelo base, condição inicial de deslocamento e velocidade e da curva de torque definida para a simulação, obtém-se a velocidade e aceleração do veículo no instante de tempo definido.

A função modelo recebe como entradas o instante de tempo, um vetor com a condição de velocidade e posição iniciais, massa, relações de transmissão das marchas e do diferencial, eficiência mecânica da transmissão, velocidades para as trocas de marcha e o valor para a seleção da curva de torque. Os dados de torque ao longo de toda faixa de rotação do veículo são configurados dentro da própria função modelo.

Para o desenvolvimento da simulação numérica foram definidas cinco curvas de torque além da curva de torque máximo do motor. Para obter os valores de velocidade utilizados no treinamento numérico da rede, foram definidas três curvas

de torque com valor constante enquanto para o teste da rede, foram definidas duas curvas de torque variável.

Dentro da função modelo, foram definidos os cálculos para o valor das resistências, como apresentado no tópico 3.1. Os valores dos coeficientes e parâmetros que caracterizam as resistências são apresentados na Tabela 2 (Genta; Morello, 2009).

Tabela 2 - Parâmetros para modelagem das resistências.

ρ	Densidade do ar [kg/m ²]	1,18	Fox; et. Al, 2010
f_0	Coefficiente de rolamento em repouso	0,0096	Genta; Morello , 2009
K	Coefficiente de rolamento	2,8E-8	Genta; Morello , 2009
η	Eficiência mecânica [%]	95	Genta; Morello, 2009

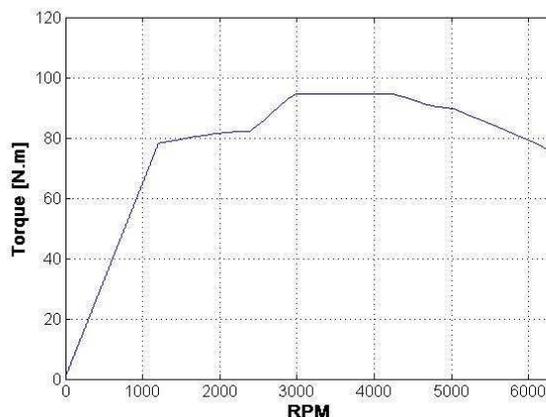
Fonte: Autor, 2024

4.1.2 Modelagem e simulação com curva de torque máximo

Para os testes iniciais, os dados foram obtidos através de simulação numérica, utilizando como base a curva de torque máximo do Volkswagen Polo. A curva de torque utilizada como base para o desenvolvimento desta simulação é apresentada no tópico 3.2.2.

A curva obtida na análise apresentada no tópico 3.2.2 apresenta valores de torque a partir de 1206 RPM. Para contornar a ausência de dados, foi definido uma relação linear crescente até o valor do torque em 1206 RPM. A curva de torque é apresentada na Figura 10.

Figura 10 - Curva de Torque



Fonte : Autor, 2024

De acordo com o manual do proprietário, a primeira relação de transmissão deve ser utilizada apenas para pôr o carro em movimento. De acordo com CFC Celso (2017), a primeira marcha deve ser utilizada na faixa de velocidades entre zero e vinte quilômetros por hora, a segunda marcha na faixa entre vinte e quarenta quilômetros por hora, a terceira marcha na faixa entre quarenta e sessenta quilômetros por hora, a quarta marcha em velocidades na faixa entre sessenta e setenta e cinco quilômetros por hora.

Para a simulação, as velocidades de troca de marcha são apresentadas na Tabela 3 (CFC Celso, 2017). Como pode ser visto no apêndice 1 as velocidades de troca de marcha foram armazenadas em um vetor de quatro posições.

Tabela 3 - Velocidades de troca de marcha

Troca de marcha	Velocidade (m/s)	Velocidade (km/h)
1 > 2	2,8	10
2 > 3	11,1	20
3 > 4	16,7	40
4 > 5	20,8	75

Fonte: Autor , 2024

A partir da integração numérica da função modelo, utilizando os dados do veículo apresentados no tópico 3.4.1 e a curva de torque máximo, são obtidos os valores de deslocamento e velocidade para as condições especificadas. A partir da velocidade do modelo são definidos os valores de rotação, torque e a relação de transmissão acionada em cada instante de tempo.

Os dados obtidos na simulação numérica são armazenados em uma matriz, e salvos em um arquivo de texto, para posteriormente serem utilizados para treinamento da rede neural.

4.1.3 Modelagem e simulação com curvas de torque diversas

Para a segunda etapa dos experimentos, foram geradas cinco curvas de torque diferentes, onde os valores de torque nessas curvas são limitados pela curva de torque máximo do veículo.

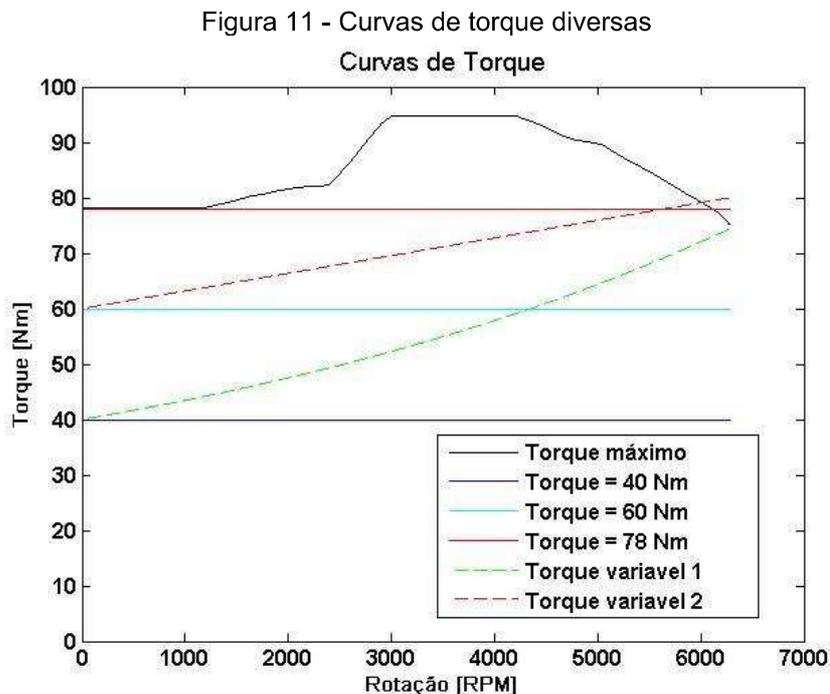
Para o treinamento da rede neural, foram definidas três curvas de torque com valor constante ao longo das rotações do motor. Os dados utilizados para o teste da rede neural foram obtidos com base em curvas de torque variáveis,

definidas pelas funções (8) e (9). Os valores das curvas de torque variáveis são limitados em uma faixa de valores entre 40 Nm e 78 Nm.

$$\tau = 20 * e^{\frac{\omega}{l_{rot}}} + 20 \quad (8)$$

$$\tau = 20 * \frac{\omega}{l_{rot}} + 60 \quad (9)$$

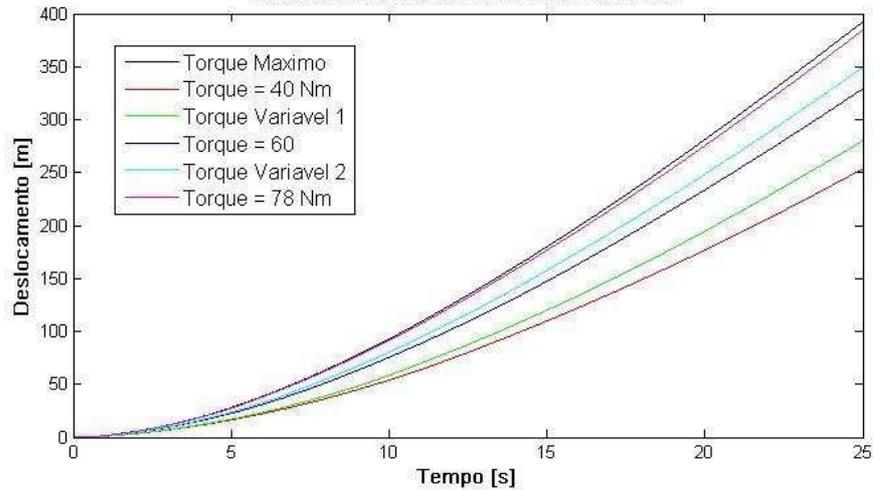
Onde, ω é a rotação do motor e l_{rot} é o comprimento total do vetor de rotações do motor. A Figura 11 mostra todas as curvas de torque definidas para o desenvolvimento do treinamento e dos testes.



Fonte: Autor, 2024

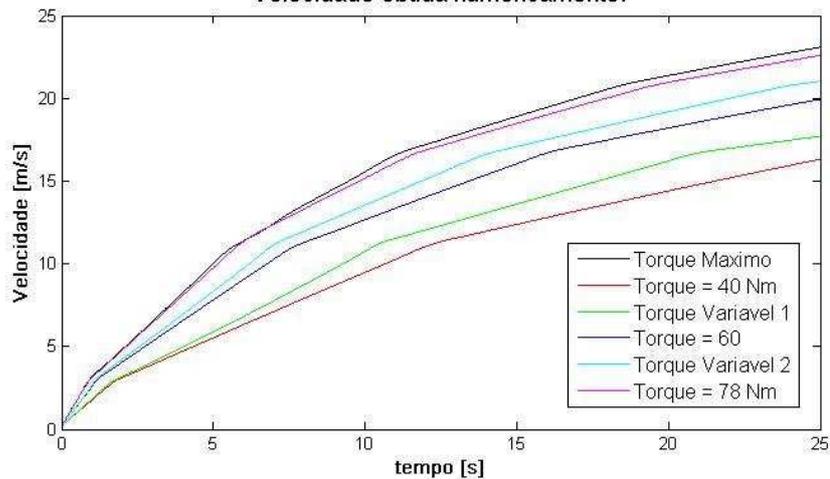
O processo de integração numéricas é o mesmo descrito no tópico 4.1.3. Os gráficos de deslocamento e velocidade obtidos pela simulação nos diferentes casos são apresentados na Figura 12 e Figura 13.

Figura 12 - Deslocamento obtido numericamente
Deslocamento obtido numericamente.



Fonte: Autor, 2024

Figura 13 - Velocidade obtida numericamente
Velocidade obtida numericamente.



Fonte: Autor, 2024

4.1.4 Preparação dos dados numéricos

Segundo Ali (2022) conjuntos de dados em que os valores dos diferentes parâmetros ocorrem em intervalos de valor diferentes, muitas vezes é interessante normalizar as escalas dos parâmetros, equalizando os pesos dos valores, a fim de evitar a ocorrência da aprendizagem com comportamento tendencioso.

A normalização min-máx. é utilizada a fim de dar o mesmo peso às variáveis utilizadas no processo de aprendizagem da rede, fazendo com que tenham o mesmo peso na tomada de decisão, independente da magnitude dos valores (Ali, 2022).

A normalização dos dados foi feita utilizando os maiores valores de velocidade e rotação considerando todos os dados dos conjuntos obtidos. Para a normalização do torque, foi utilizado como parâmetros o maior valor de torque na curva de torque máximo. O valor mínimo para a normalização das variáveis de entrada é sempre zero, visto que dentro do intervalo de valores obtidos neste trabalho, este é o menor valor que pode ocorrer para qualquer uma das variáveis.

4.1.5 Treinamento da rede neural com dados numéricos

O processo de treinamento e teste utilizando a curva de torque máximo do veículo consistiu em treinar a rede com os dados obtidos numericamente e, utilizando os mesmos dados de entrada utilizados no treinamento, verificou-se as saídas dadas pela rede neural, comparando com as saídas obtidas numericamente. Visto que o processo de aprendizado da rede *Perceptron* inicia com a rede definindo valores aleatórios para os neurônios, o processo de treinamento e validação foi realizado três vezes, onde os valores de precisão obtidos são apresentados na Tabela 4. Os valores de precisão apresentados referem-se a taxa de acerto na escolha das marchas obtidas nos testes, utilizando os pesos obtidos no treinamento da rede neural.

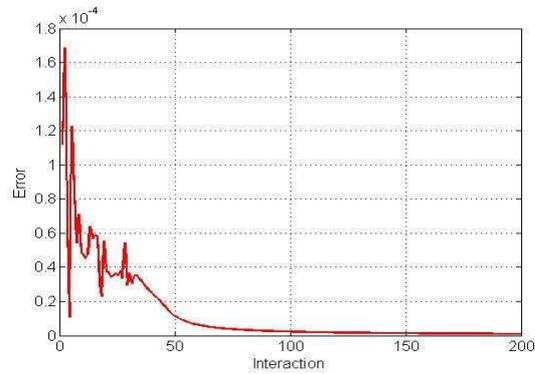
Tabela 4 - Precisão dos testes com curva de torque máximo

Teste com curva de torque máximo	
Teste 1	98,16%
Teste 2	96,20%
Teste 3	96,64%

Fonte: Autor, 2024

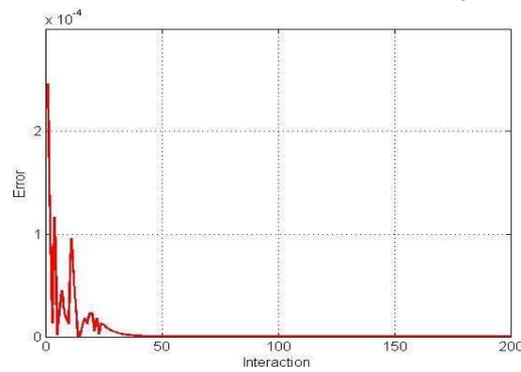
A alta precisão da rede mostra que a mesma tem capacidade de aprendizado dos padrões numéricos, atingindo uma precisão média de 97% na seleção das marchas. A Figura 14, Figura 15 Figura 16 mostram a variação do erro no processo de aprendizado ao longo do primeiro, segundo e terceiro teste, respectivamente.

Figura 14 – Valor do erro do teste 1 com torque máximo



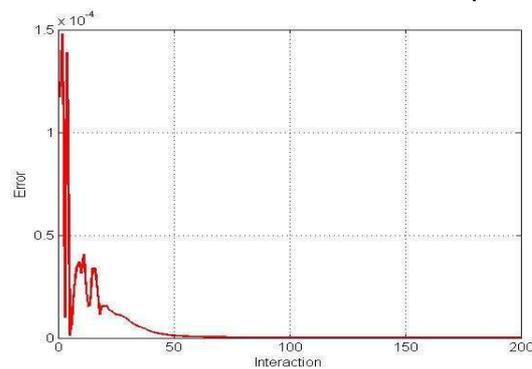
Fonte: Autor, 2024

Figura 15 - Valor do erro do teste 2 com torque máximo



Fonte: Autor, 2024

Figura 16 - Valor do erro do teste 3 com torque máximo



Fonte: Autor, 2024

O processo de treinamento e teste da rede neural com diferentes curvas de torque ocorre de forma semelhante ao processo para a curva de torque máximo. No entanto, para esta etapa, o treinamento da rede foi realizado utilizando os dados

obtidos com valores de torque constante enquanto o teste utilizou as curvas de torque variável, como apresentado no tópico 4.1.3.

Assim como para o teste com a curva de torque máximo, o processo de aprendizagem e testes da rede *Perceptron* foi executado três vezes. A precisão dos testes, no que se refere à taxa de acerto na seleção das marchas, utilizando curvas de torque diferente são apresentados na Tabela 5.

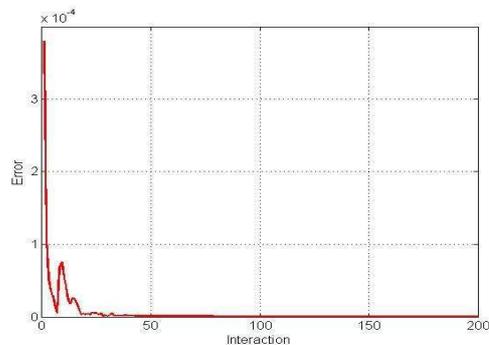
Tabela 5 - Precisão dos testes com curvas de torque diversas

Teste com curvas de torque diversos	
Teste 1	94,74%
Teste 2	100%
Teste 3	100%

Fonte: Autor, 2024

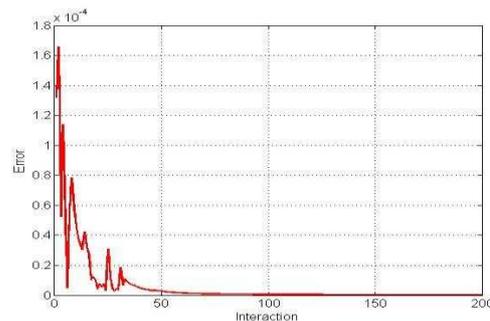
As variações do erro ao longo do processo de aprendizado com curvas de torque diversas para o primeiro, segundo e terceiro teste são apresentadas na Figura 17, Figura 18 e Figura 19. Para esta condição a rede apresentou uma precisão média de 98,25%, resultado acima do obtido nos testes anteriores.

Figura 17 - Valor do erro do teste 1 com curvas de torque diversas



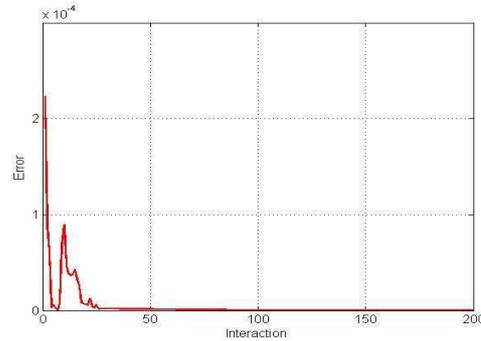
Fonte: Autor, 2024

Figura 18 - Valor do erro do teste 2 com curvas de torque diversas



Fonte: Autor, 2024

Figura 19 - Valor do erro do teste 3 com curvas de torque diversas



Fonte: Autor, 2024

A fim de testar a capacidade, dada a alta precisão obtida nos testes com diferentes curvas de torque, o processo de treinamento da rede neural foi realizado uma segunda vez, adicionando um ruído de 10% nas velocidades de troca das marchas, a fim de tornar o instante exato da troca de marcha incerto, assim como ocorre na prática, na operação de um câmbio manual. Além disso, a sequência dos dados de treinamento foi aleatorizada.

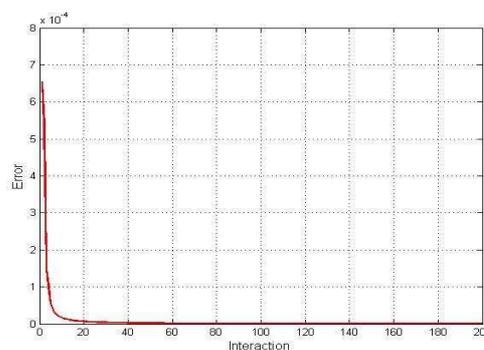
Os valores de precisão obtidos para os três testes realizados são exibidos na Tabela 6. A variação do erro ao longo do processo de aprendizagem com dados aleatorizados é mostrada na Figura 20, Figura 21 e Figura 22.

Tabela 6 - Precisão dos testes com ruídos e dados aleatorizados

Teste com curvas de torque diversos	
Teste 1	100%
Teste 2	100%
Teste 3	100%

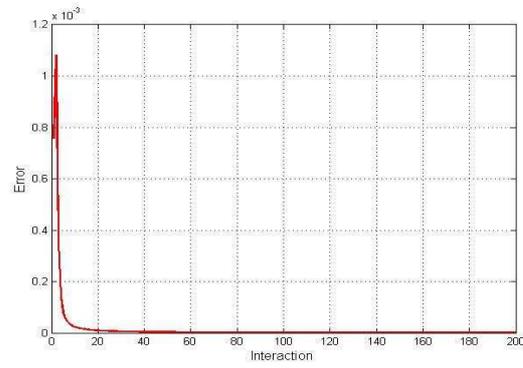
Fonte: Autor, 2024

Figura 20 - Valor do erro do teste 1 com dados com ruído



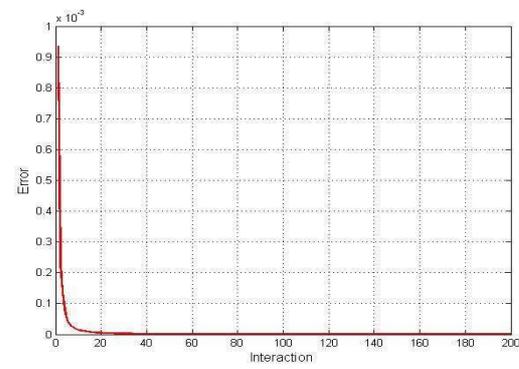
Fonte: Autor, 2024

Figura 21 - Valor do erro do teste 2 com dados com ruído



Fonte: Autor, 2024

Figura 22 - Valor do erro do teste 3 com dados com ruído

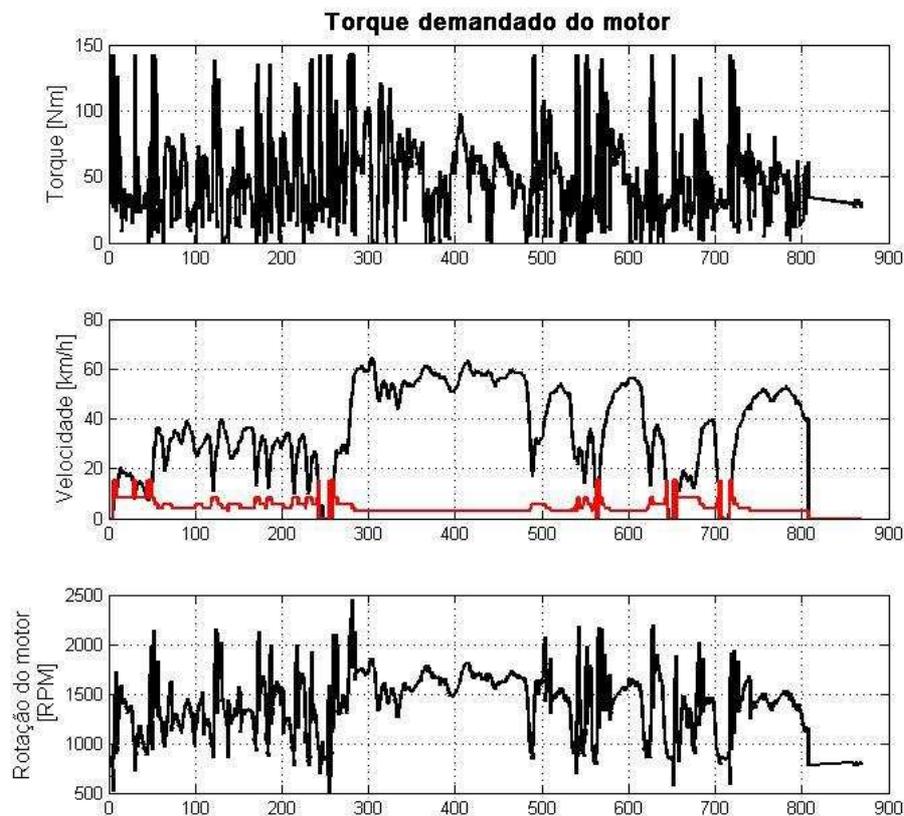


Fonte: Autor, 2024

5 TREINAMENTO COM DADOS EXPERIMENTAIS

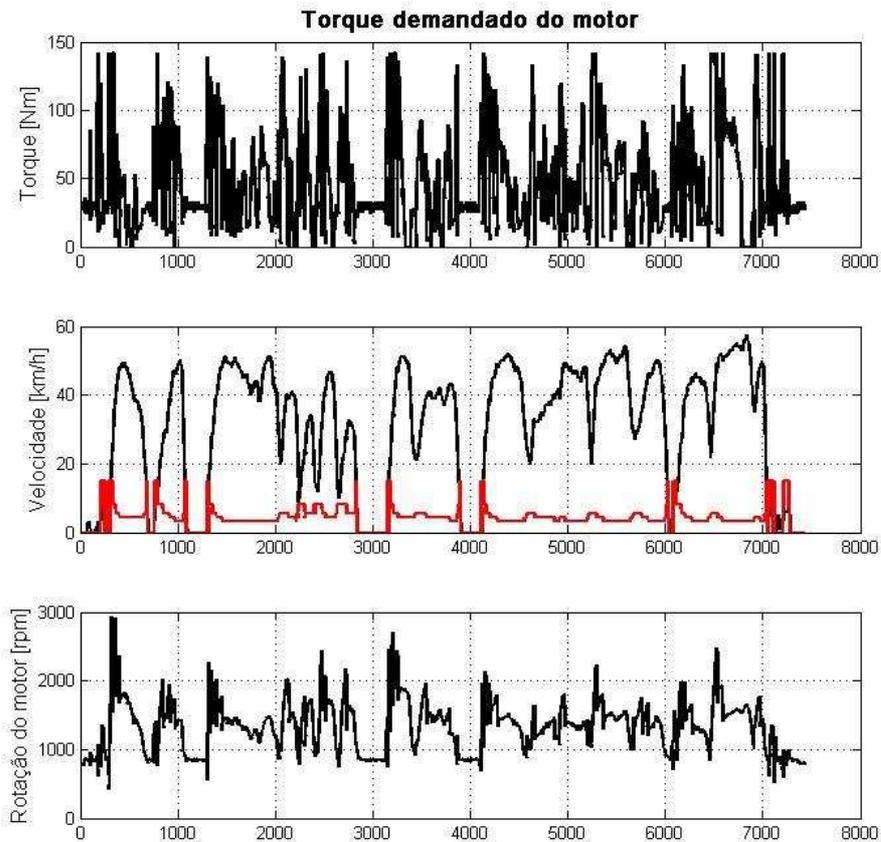
A terceira etapa dos testes da rede neural consistiu em treinar e testar a precisão da rede neural utilizando os dados obtidos experimentalmente. As Figura 23 e Figura 24 apresentam os dados experimentais (Costa, 2019). A curvas em preto representam os valores de torque, velocidade e rotação do motor obtidos ao longo do trajeto de aquisição dos dados. A curva em vermelho, apresentada no gráfico da velocidade representa o valor total da relação de transmissão na marcha engatada e relação do diferencial, onde a primeira marcha é representada pelo valor 15,18, a segunda marcha pelo valor 8,34, a terceira marcha o valor 5,56, a quarta marcha o valor 4,19 e a quinta marcha o valor 3,09.

Figura 23 - Valores obtidos experimentalmente no primeiro trajeto



Fonte: Costa, 2019

Figura 24 - Valores obtidos experimentalmente no segundo trajeto



Fonte: Costa, 2019

5.1.1 Preparação dos dados experimentais

A aquisição dos dados experimentais ocorreu em um circuito urbano, onde a velocidade do veículo é dada em função do fluxo de veículos e das regras de trânsito, o que faz com que haja diversas ocorrências de intervalos de tempo com velocidade zero. A fim de manter apenas dados relevantes para a análise, os instantes de tempo em que o valor da velocidade é nulo foram removidos dos conjuntos de dados.

Inicialmente, a seleção da marcha acionada representada pelo valor da relação de transmissão total foi mantida para o treinamento da rede. Porém, com o intuito de deixar o teste com dados experimentais o mais próximo possível do teste realizado com dados numéricos, o valor que representa a seleção da relação de transmissão foi convertido para o formato utilizado nos testes numéricos, ou seja, no

formato binário. Por exemplo, a primeira marcha é representada por [1,0,0,0,0], a segunda marcha por [0,1,0,0,0] e assim por diante para as outras três marchas.

Assim como o processo com dados numéricos, os outros valores experimentais de velocidade do veículo, rotação e torque do motor foram normalizados, utilizando como parâmetro máximo para normalização os maiores valores de velocidade, rotação e torque, respectivamente, considerando os dois conjuntos de dados utilizados.

5.1.2 Treinamento da rede neural com dados experimentais

Para o treinamento da rede neural, foi utilizado o conjunto de dados obtido na primeira parte do percurso de aquisição de dados.

O teste de precisão foi realizado utilizando o conjunto de dados obtido na segunda parte do percurso de aquisição de dados. Para o treinamento da rede, foram utilizados os dados de entrada e de saída do primeiro conjunto de dados, a fim de treinar a rede de acordo com os comportamentos do motorista no processo de aquisição dos dados. O segundo conjunto de dados experimentais foi tomado com o mesmo motorista, o que faz com que os comportamentos de condução sejam os mesmos. Para realizar o teste da rede neural, foram utilizados os dados de entrada do segundo conjunto de dados de teste, considerando os pesos gerados durante o treinamento da rede.

A Tabela 7 apresenta a precisão das respostas da rede neural treinada com o primeiro conjunto de dados. A fim de avaliar a repetibilidade, o teste da precisão de escolha da rede foi realizado três vezes, utilizando como dados de entrada a velocidade, rotação e torque do segundo conjunto de dados e os pesos obtidos no treinamento da rede com o primeiro conjunto de dados. A precisão obtida é dada comparando a marcha engatada, obtido pela rede neural, e a marcha esperada, conforme os dados de saída experimentais.

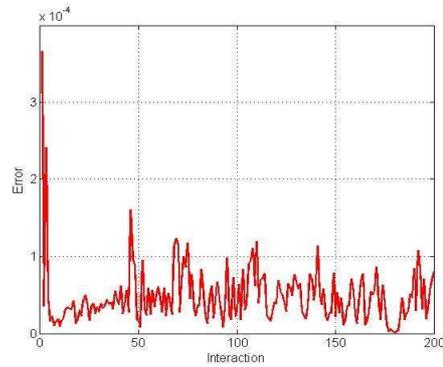
Tabela 7 - Precisão dos testes com dados experimentais

Teste com dados experimentais	
Teste 1	94,4%
Teste 2	95,06%
Teste 3	94,68%

Fonte: Autor, 2024

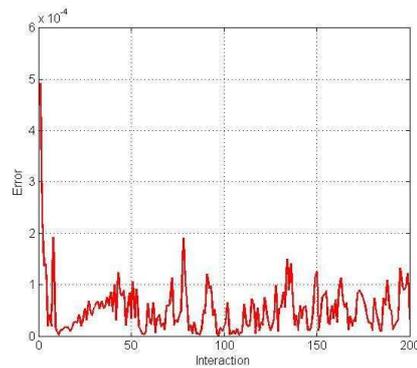
A variação do erro ao longo de cada um dos testes é apresentada na Figura 25, Figura 26 e Figura 27.

Figura 25 - Valor do erro do teste 1 com dados experimentais



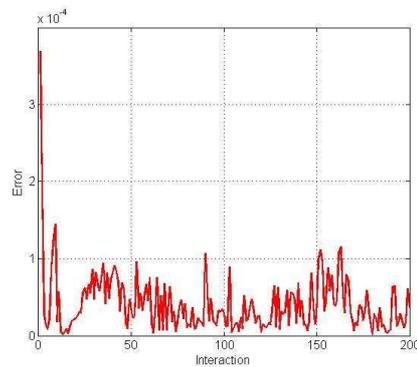
Fonte: Autor, 2024

Figura 26 - Valor do erro do teste 2 com dados experimentais



Fonte: Autor, 2024

Figura 27 - Valor do erro do teste 3 com dados experimentais



Fonte: Autor, 2024

6 CONCLUSÃO

Este trabalho teve como objetivo implementar a dinâmica veicular de um veículo com powertrain automatizado convencional, investigando a possibilidade de utilizar um sistema de inteligência artificial atuando no processo de decisão para a troca de marchas. Utilizando uma rede *Perceptron* para o treinamento e testes, com base em valores numéricos e experimentais, foi possível verificar a eficiência da ferramenta no processo de aprendizado dos padrões de condução e a precisão na replicação dos padrões identificados.

Os resultados obtidos nos testes apontam que o algoritmo de rede *Perceptron* implementado no sistema de transmissão convencional automatizado apresenta uma boa precisão no processo de escolha das marchas. O sistema apresentou uma alta precisão atuando em diferentes situações, o que indica grande robustez da rede neural, mostrando que a mesma é plenamente capaz de aprender os padrões e replicá-los em diversas situações de forma precisa mesmo utilizando dados aleatorizados dentro de um mesmo conjunto.

Durante o desenvolvimento dos testes experimentais, em um primeiro momento, a rede foi treinada e testada usando os valores numéricos da relação de transmissão total para representar a marcha acionada em cada instante de tempo. O uso deste formato apresentou uma queda aparente na precisão da rede neural, o que foi corrigido ao converter o modo de representar a marcha acionada para o vetor de cinco posições.

Analisando o comportamento do erro do processo de aprendizagem utilizando dados numéricos, nota-se que o valor converge desde as primeiras iterações. Para a análise feita com a curva de torque máximo, nota-se que nas primeiras iterações do processo de aprendizado, o valor do erro apresenta instabilidade, e passa a apresentar um comportamento de estabilização. A instabilidade no início ocorre pois os valores das três variáveis utilizadas no aprendizado da rede apresentam variação no seu valor ao longo do intervalo de tempo da simulação.

Para os testes com dados obtidos utilizando diferentes curvas de torque para obtenção dos valores de treinamento e teste, nota-se que o erro apresenta um comportamento estável de queda, apresentando convergência. Para o treinamento,

o valor do torque é sempre constante, independentemente do valor de rotação do motor. Com os dados em sequência, a constância dos valores de torque ao longo de grandes intervalos de dados pode fazer com que a rede dê um peso menor ao fator. Entretanto, ao apresentar dados com este comportamento, a rede acaba apresentando alguma instabilidade nas primeiras iterações, por não compreender em que momento o valor do torque muda. Ao montar o vetor de entrada dos dados de treinamento de forma aleatória, a rede passa a dar um peso maior ao dado de torque, dado que o padrão na ocorrência dos dados de torque deixa de existir. Analisando o erro nos testes aleatorizados, é possível perceber que, dado a quebra na ordem dos dados, o aprendizado da rede é mais rápido e preciso.

Avaliando o erro nos testes utilizando dados experimentais, nota-se que o mesmo apresenta um padrão de convergência, porém, com o valor do erro oscilando em uma faixa cada vez menor. Por se tratar de dados experimentais, há uma ocorrência maior de ruídos, visto que os valores não seguem um padrão conhecido de variação. Além disso, nos testes experimentais, o valor de velocidade apresenta comportamentos crescentes e decrescentes, o que não ocorre no caso dos dados numéricos.

O estudo desenvolvido neste trabalho, limitou-se a uma análise considerando apenas os parâmetros de velocidade, rotação e torque para a escolha das marchas. Além disso, a análise realizada considerou apenas a condução do veículo em peso de marcha, com apenas uma pessoa, nenhum tipo de carga adicional de valor constante, ou variações de carga, como a gerada pelo consumo do combustível ao longo do trajeto. A simulação numérica se limita a um intervalo de vinte segundos, onde os valores obtidos de velocidade, rotação e torque têm um comportamento crescente ao longo de todo intervalo. A presença dos valores de 100% e a queda do valor do erro no início do processo de aprendizagem apontam uma maior capacidade por parte da rede, o que requer uma investigação mais aprofundada.

6.1 ESTUDOS FUTUROS

A partir do estudo desenvolvidos nesta obra, são sugeridos os seguintes temas para estudos futuros:

- Análise do impacto do sistema de acoplamento no processo de tomada de decisão;
- Desenvolvimento da dinâmica veicular e do processo de tomada de decisão, incluindo informações como consumo específico de combustível, aceleração do veículo e inclinação do terreno;
- Precisão da rede *Perceptron* na de trocas de marcha, incluindo diferentes modos de condução;
- Inclusão de um processo de customização nos padrões de pilotagens, a partir de um sistema de troca manual ou semiautomático;
- Simulação de um modelo numérico incluindo reduções nos valores de velocidade ;
- Simulações baseadas em outros veículos.

REFERÊNCIAS

- AGGARWAL, C. C. **Neural networks and deep learning: a textbook**. Berlin: Springer, 2018.
- ALI, P. J. M. . **Investigating the Impact of Min-Max Data Normalization on the Regression Performance of K-Nearest Neighbor with Different Similarity Measurements**. Aro-The Scientific Journal Of Koya University, [S.L.], v. 10, n. 1, p. 85-91, 21 jun. 2022. Koya University. <http://dx.doi.org/10.14500/aro.10955>
- BERTSCHE, B; NAUNHEIMER, H; NOVAK, W; RYBORZ, J. **Automotive transmissions: fundamentals, selection, design and application**. 2. Ed. Heidelberg: Springer Berlin, 2011.
- BOBER, W. **Introduction to Numerical and Analytical Methods with MATLAB® for Engineers and Scientists**. Boca Raton: Taylor & Francis Group, 2014.
- BOSCH, R. **Manual de tecnologia automotiva**. 25 ed. São Paulo: Edgard Blucher, 2005.
- CFC CELSO. **Como mudar de marchas corretamente**. CFC Celso, 12 jan. 2017. Disponível em: <https://cfccelso.com.br/como-mudar-de-marchas-corretamente/>. Acesso em : 27 maio 2024
- CHAPMAN, S. J.. **Programação em MATLAB para Engenheiros**. 5. ed. [S.l]: Cengage Learning, 2016.
- CHAPRA, S. C.. **Métodos Numéricos Aplicados com MATLAB para engenheiros e cientistas**. 3. ed. Porto Alegre: Amgh Editora Ltda, 2013.
- CROLLA, D. A. MASHADI, B. **Vehicle powertrain systems**. Reino Unido: Wiley, 2012.
- CROLLA, D. **Automotive engineering: powertrain, chassis system and vehicle body**. Oxford: Elsevier, 2009.
- CUSTÓDIO, A. Câmbio automático deve equipar mais de 90% dos veículos até o final desta década no Brasil, preveem especialistas. **Gaúcha Zero Hora**, 09 de jun. 2021. Disponível em: <https://gauchazh.clicrbs.com.br/comportamento/carros/sobre-rodas/noticia/2021/06/cambio-automatico-deve-equipar-mais-de-90-dos-veiculos-ate-o-final-desta-decada-no-brasil-preveem-especialistas-ckppz2f9i009h018m473f4fkq.html>. Acesso em: 25 de nov. 2022.
- DORMAND, J. R.; PRINCE, P. J.. A family of embedded Runge-Kutta formulae. **Journal Of Computational And Applied Mathematics**. Cidade. Mar. 1980.
- FAUSETT, L. **Fundamentals of neural networks: architectures, algorithms and applications: architectures, algorithms and applications**. Nova Jersey: Prentice hall, 1993.

FELDMAN, B. Fim do câmbio automatizado: já vai tarde! **Autopapo**. 23 de mar. 2019. Disponível em: <https://autopapo.uol.com.br/noticia/fim-do-cambio-automatizado/>. Acesso em: 18 de dez. de 2022.

FOX, W.R.; MCDONALD, A.T.; PRITCHART, P. J.; LEYLEGIAN, J.C.. **Introdução à mecânica dos fluidos**. 8 ed. Barueri: Grupo Editora Nacional, 2010.

GAMA, J. Por que o preço dos carros subiu na pandemia e não vai mais baixar? **Uol**. 25 dez. 2023. Disponível em: <https://www.uol.com.br/carros/colunas/paula-gama/2023/12/25/por-que-preco-dos-carros-subiu-na-pandemia-e-nao-vai-mais-baixar.html>. Acesso em: 25 jun. 2023.

GENTA, G. MORELLO, L. **The Automotive Chassis**: Volume 2: System Design. Dordrecht: Springer Science + Business Media, 2009.

HAYKIN, Simon. *Redes Neurais - Princípios e Práticas*. Brasil: Bookman, 2000.
IIDA, I. **Ergonomia**: projeto e produção. 2. ed. São Paulo: Edgard Blucher, 2005.

KUCHENBECKER, D. **Ferramenta computacional para cálculo de pontos de troca de marcha em transmissões manuais automatizadas**. 2021. Trabalho de Conclusão de Curso (graduação em Engenharia de Controle e Automação) – Centro de Blumenau, Universidade Federal de Santa Catarina, Blumenau, 2021.

MIRAGAYA, F. Carros automáticos têm cada vez mais procura e fazem sucesso entre usados. **Quatro Rodas**, 27 de jul. 2022. Disponível em : <https://quatorrodas.abril.com.br/auto-servico/carros-automaticos-tem-cada-vez-mais-procura-e-fazem-sucesso-entre-usados/>. Acesso em: 23 de nov. 2022.

MOTOR1 BRASIL. Novo Polo 1.0 MPI - Versão básica vale a pena ? Youtube, 18 de março de 2018. Disponível em: https://youtu.be/RrXdmnjl-mQ?si=sYKG1IUFjwZrk_b. Acesso em: 15 de abril de 2024.

NASSER, R. Novo Polo, o arrasa-quarteirão. **Autoentusiastas**. 19 de set. 2017. Disponível em: <https://autoentusiastas.com.br/2017/09/novo-polo-o-arrasa-quarteirao/>. Acesso em 05 de set. 2023.

NORTON, R., L. **Projeto de máquinas**: uma abordagem integrada. 4 ed. Porto Alegre: Bookman, 2013.

PINEAU, C. The psychological meaning of Comfort. **Applied Psychology**, Paris, v.31, n.2, p.271 – 282, abr. 1982.

RAJAMANI, R. **Vehicle Dynamics and Control**. Minneapolis: Springer, 2005.

RUMELHART, D.E.; HINTON, G.E.; WILLIAMS, R.J.. Learning Internal Representations by Error Propagation. **Readings In Cognitive Science**, [S.L.], p. 399-421, 1988. Elsevier. <http://dx.doi.org/10.1016/b978-1-4832-1446-7.50035-2>.

SANTANA, N. Pesquisa mostra ordem dos fatores decisivos na hora de comprar carros. **Garagem 360**, 8 de nov. 2021. Disponível em: <https://garagem360.com.br/pesquisa-mostra-ordem-dos-fatores-decisivos-na-hora-de-comprar-carros/>. Acesso em: 25 de nov. 2022.

SILVA, C., W. **Vibration**: Fundamentals and practice. Boca Raton: CRC, 2000.

SIZEMORE, J.; MUELLER, J.. **MATLAB Para Leigos**. Rio de Janeiro: Alta Books, 2016.

STOJKOVIC, M.. **Wine Classification using neural networks**. Neuroph, 2012.

Disponível em:

<https://neuroph.sourceforge.net/tutorials/wines1/WineClassificationUsingNeuralNetworks.html>. Acesso em: 25 ago. 2023.

VOLKSWAGEN DO BRASIL. **Manual de instruções**: Polo. São Paulo: Volkswagen, 2017 .

ANEXO A – ALGORITMOS

ALGORITMO PARA DEFINIR A CURVA DE TORQUE DO MOTOR

```

clear; clc; format short eng
% modelo de referência: VW polo 2018 1.0 MSI manual
reldiftsi = 4.929; %relação diferencial
relmarctsi = [4.167 2.300 1.433 0.975 0.776]; %relações de marcha

% GRAFICO DENTE DE SERRA: os dados da matriz de velocidades em função da
% marcha e da rotação, exibidos em conjunto com a rotação geram o grafico
% dente de serra. (pra que serve este grafico ?)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%referência de torque
rtrq = 0:15:120; %vetor de torqr
ptrqs0 = [552 489 432 367 304 244 183 122]; %vetor com pixels de torque sem o pixel torque 0 Nm
medptrq = 0;

for h = 1:(numel(ptrqs0)-1)
    medptrq = medptrq + (ptrqs0(h) - ptrqs0(h+1));
end
medptrq = medptrq/h;
ptrq = [ptrqs0(1)+fix(medptrq) 552 489 432 367 304 244 183 122];

%referência de rotação do eixo do grafico
prot = [69 100 134 168 202 235 269 303 337 370 404 438 471 505 534]; %Vetor com pixels de referencia para
localização das rotações
rrot = 0:500:7000; %vetor com os valores de rotação

%Dados em posição de pixel: Posicionamento de alguns dos pontos das curvas do grafico de torque em função
da rotação para o veiculo escolhido

% Pontos das rotações no grafico
protg = [69 148 156 164 177 188 195 206 214 228 240 250 256 262 269 289 316 352 364 374 381 389 400 408 415
423 430 443 446 462 470 476 479 483 486 490 494]; %rotações no grafico
% Pontos dos valores de torque no grafico
ptrqg = [610 291 289 287 283 281 279 277 276 275 261 247 239 231 225 225 225 225 230 235 239 242 244 246
251 256 260 268 270 281 286 290 292 295 298 303 305]; %torque no grafico

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5

rotacao = (protg);
i = 1;
j=1;

while i <= length(protg) %Calculo para definição da rotação da curva de torque com dados mais apurados
    if protg(j) > protg(i)
        rotacao(i+1) = ((rrot(j)-rrot(j-1))/(protg(j)-protg(j-1))*(protg(i)-protg(j-1)))+rrot(j-1);
        i=i+1;
        j=1;
    else
        j = j+1;
    end
end

torque = ptrqg;
i=1;
j=1;

while i <= length(ptrqg)
    if ptrqg(j) < ptrqg(i) %Calculo para definição do torque no grafico de torque com dados mais apurados
        torque(i) = ((rtrq(j)-rtrq(j-1))/(ptrqg(j)-ptrqg(j-1))*(ptrqg(i)-ptrqg(j-1)))+rtrq(j-1);
        i=i+1;
        j=1;
    else
        j = j+1;
    end
end

rot_s0 = rotacao(3:length(rotacao));
trq_s0 = torque(2:length(torque));
% for i = 3:1:length(rotacao)
%     rot_s0(i-1) = rotacao(i);
%     trq_s0(i-1) = torque(i);
% end
%Exibe curva de torque sem extrapolação inicial

```

```

figure
plot(rot_s0,trq_s0);
axis([0 7000 0 120]);

xlabel('Rotação [RPM]');
ylabel('Torque [Nm]');

torque = [0 torque];

%EXIBE CURVA DE TORQUE INICIAL (IMPRECISA)

% plot(rotacao,torque);
% axis([0 6338 0 120]);

rpmtorque = 1:1:round(rotacao(37));

i=1;
j=2;
t = 1:1:length(rpmtorque); % t é o vetor que vai armazenar os valores de torque obtidos através de
interpolação linear

while i<=length(rpmtorque)
    if rpmtorque(i) < rotacao(j)
        t(1,i) = torque(j) - ((rotacao(j)-rpmtorque(i))/(rotacao(j)-rotacao(j-1)))*(torque(j)-torque(j-1));
        i = i+1;
        j = 2;
    else
        j=j+1;
    end
end

% % % GERA GRAFICO COM A CURVA DE TORQUE (INTERPOLADA)
% % % Esta curva de torque tem maior precisão
figure
plot(rpmtorque,t);
axis([0 6338 0 120]);
grid;
ylabel('Torque [N.m]');
xlabel('RPM');
% save ('vetor_torque.txt', 't', '-ascii');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

trqsai = 1:1:(length(t));

for j = 1:1:length(reymarctsi)
    for i = 1:1:(length(t))
        trqsai(j,i) = t(i)*(reldiftsi*reymarctsi(j)); %matriz com valores de torque de saida, na roda, em
função da marcha, sem considerar perdas no sistema
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
rpmmax = length(t); %>>>>>>>>>
rpmmotor = 1:1:rpmmax; %rotação do motor %>>>>>>>>>

rpmsaida = zeros(5,rpmmax); %rotação de saida %>>>>>>>>>
vangult = 0; %>>>>>>>>>>
a = 1; %>>>>>>>>>>>>

for j = 1:1:length(reymarctsi)
    for i = 1:1:(rpmmax)
        a = ((i*2*pi)/60)/(reldiftsi*reymarctsi(j));%convertido em rad/s
        if a > vangult
            rpmsaida(j,i) = a;
        end
    end
    vangult = rpmsaida(j,i);
end

raio_pol = 15; %raio da roda em polegada, não é o raio dinâmico
% Pneu 185/65 R15
w_pneu = 0.185; %largura do pneu [m]
h_pneu = 0.65*w_pneu; % altura do pneu [m]
raio_total = (raio_pol/39.37) + h_pneu; %raio da roda em polegada, não é o raio dinâmico
coefatrdin = 0.8; %coeficiente de artito aerodinamico
raiom = raio_total/((2-coefatrdin)); %raio da roda em metros (raio estatico da roda considerando o coef de
atrito dinamico para o asfalto = 0.8
% vcar = 0:1:6500; %vetor velocidade do carro

vcar = zeros(5,rpmmax);

for j = 1:1:length(reymarctsi)
    for i = 1:1:(rpmmax)
        if rpmsaida(j,i) ==0
            vcar(j,i) = b;
        else
            vcar(j,i) = rpmsaida(j,i)*raiom*3.6;%velocidade do veiculo em metros por segundo
        end
    end
end

```

```

        b = vcar(j,i);
    end
end
end
end

% save ('vcar.txt', 'vcar', '-ascii')

Acel = zeros(5,length(t));

%Valores para constantes e parametros do veiculo, parametros de rodagem,
%aerodinamica, aderencia,

m = 1138; % peso do veiculo em ordem de marcha + condutor de 80 kg
densar = 1.18;
cd = 0.333;
f0 = 0.015;
alfa = 0;
ef = 0.99;
af = 2.15; %area frontal em m2
g = 9.81;

%curvas de força trativa em função das velocidades, para cada marcha.

for j = 1:length(relmarctsi)
    for i = 2:(length(t)-1)
        if vcar(j,i) == vcar(j,i-1)
            Acel(j,i-1) = 0 ;
        else
            Acel(j,i-1) = (1/m)*((forcatrativa(t(i-1),reldiftsi*relmarctsi(j),0.95,raiom)) -
(resaero(vcar(j,i-1),m,densar,cd) + resrol(vcar(j,i-1),f0,m) + resang(m,alfa)));
        end
    end
end

for j = 1:length(relmarctsi)
    for i = 1:(length(t))
        if Acel(j,i) < 0
            Acel(j,i) = 0;
        end
    end
end

% diagrama da força trativa pela velocidade. A matriz Acel retorna a força
% trativa (nome dado anteriormente pode enganar, mas Acel se refere a força
% trativa)

for i = 1:(length(t))
    v1ft(i) = vcar(1,i);
    v2ft(i) = vcar(2,i);
    v3ft(i) = vcar(3,i);
    v4ft(i) = vcar(4,i);
    v5ft(i) = vcar(5,i);
    F1tr(i) = Acel(1,i);
    F2tr(i) = Acel(2,i);
    F3tr(i) = Acel(3,i);
    F4tr(i) = Acel(4,i);
    F5tr(i) = Acel(5,i);
end

% plot(v1ft, F1tr, v2ft, F2tr , v3ft, F3tr, v4ft, F4tr, v5ft, F5tr);
% xlabel('velocidade');
% ylabel('força');

%%

for i=1:length(t)
    if i <= 1300
        if t(i) <= 78.25
            trq0(i) = 78.25;
        elseif t(i) > 78.25
            trq0(i) = t(i);
        end
    end
    if i > 1300
        trq0(i) = t(i);
    end
end
end

```

ALGORITMO PARA INTEGRAÇÃO DA FUNÇÃO MODELO

```

% Arquivo principal
% Dados de entrada do carro
% Converter km/h> m/s: Vkmh = 3.6*Vms

clear all
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%VARIÁVEL DE ATIVAÇÃO: 10: EXECUTA SIMULAÇÃO COM CURVA DE TORQUE MÁXIMO
%11: EXECUTA SIMULAÇÃO COM CURVAS DE TORQUE CONSTANTES
%13: Executa simulação para teste de torque constante intermediário

ativ_var = 13;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% global v torque vel_tst trq_tst rot_tst mrc
%% Dados para uso em todas as simulações
global trq_tst rot_tst vel_tst mrc tmp_tst
% global trq_tst50 rot_tst50 vel_tst50 mrc50 tmp_tst50
% global trq_tst55 rot_tst55 vel_tst55 mrc55 tmp_tst55
% global trq_tst60 rot_tst60 vel_tst60 mrc60 tmp_tst60
% global trq_tst65 rot_tst65 vel_tst65 mrc65 tmp_tst65
% global trq_tst70 rot_tst70 vel_tst70 mrc70 tmp_tst70

trq_tst = 0; rot_tst = 0; vel_tst = 0; mrc = 0; tmp_tst = 0;
% trq_tst50 = 0; rot_tst50 = 0; vel_tst50 = 0; mrc50 = 0; tmp_tst50 = 0;
% trq_tst55 = 0; rot_tst55 = 0; vel_tst55 = 0; mrc55 = 0; tmp_tst55 = 0;
% trq_tst60 = 0; rot_tst60 = 0; vel_tst60 = 0; mrc60 = 0; tmp_tst60 = 0;
% trq_tst65 = 0; rot_tst65 = 0; vel_tst65 = 0; mrc65 = 0; tmp_tst65 = 0;
% trq_tst70 = 0; rot_tst70 = 0; vel_tst70 = 0; mrc70 = 0; tmp_tst70 = 0;

% Dados do carro
trqrot_imp = (importdata('vetor_torque.txt')); %importa valores para curva de torque (cada posição do vetor equivale a 1
rotação na escala do veículo)
trq = transpose(trqrot_imp);
rot_0 = 1:length(trq);

t=0:0.01:25; % Intervalo de tempo do integra
m=1138; % 1058 kg peso do carro em ordem de marcha + condutor de 80 kg [kg]
%Peso em ordem de marcha: 1070 kg
%Peso em ordem de marcha com condutor de 80kg :
raio_pol = 15; %raio da roda em polegada, não é o raio dinâmico
% Pneu 185/65 R15
w_pneu = 0.185; %largura do pneu [m]
h_pneu = 0.65*w_pneu; % altura do pneu [m]
raio_total = (raio_pol/39.37) + h_pneu;
coefatrdin = 0.8;
% Rd = raio_total/((2-coefatrdin)*39.37);
Rd = raio_total/((2-coefatrdin));
idif = 4.929; %relação diferencial
it = [4.167 2.300 1.433 0.975 0.776]; %relações de marcha
neta=0.95;

% trq0 = 1:1:0.5*(length(trq));
for i=1:length(trq)
    if i <= 1300
        if trq(i) <= 78.25
            trq0(i) = 78.25;
        elseif trq(i) > 78.25
            trq0(i) = trq(i);
        end
    end
    if i > 1300
        trq0(i) = trq(i);
    end
end

load('ycurva0_semretainicial.txt')
v_trqmax_inicial = ycurva0_semretainicial;

x0 = [0 0.2]; %Vetor de entrada [posição_inicial velocidade_inicial]

v_troca = [2.8 11.1 16.7 20.8];

%% Teste com Curva de torque máximo
if ativ_var == 10;
%resultados para treinamento da IA
%% Treinamento inicial com a curva de torque 100%

[t_int,y] = ode45(@funcao_carro_adap,t,x0,[],m,Rd,it,idif,neta,v_troca,0);

a_it10 = zeros(length(y(:,2)),5);
a_it = zeros(1,length(y(:,2)));
rot_mot = 1:length(y(:,2));
t_0 = 1:length(y(:,2));
for i = 1:length(y(:,2))
    if y(i,2) > 0 && y(i,2) <= v_troca(1) %0 à 20
        a_it(i) = it(1);
        a_it10(i,1) = 1;
    elseif y(i,2) > v_troca(1) && y(i,2) <= v_troca(2) %20 à 30
        a_it(i) = it(2);
        a_it10(i,2) = 1;
    elseif y(i,2) > v_troca(2) && y(i,2) <= v_troca(3) %30 à 40

```

```

        a_it(i) = it(3);
        a_it10(i,3) = 1;
    elseif y(i,2) > v_troca(3) && y(i,2) <= v_troca(4) %40 à 60
        a_it(i) = it(4);
        a_it10(i,4) = 1;
    elseif y(i,2) > v_troca(4) %60 e acima
        a_it(i) = it(5);
        a_it10(i,4) = 1;
    end
    rot_roda = y(i,2)/Rd;
    rot_mot(i) = (rot_roda*30*a_it(i)*idif)/pi;
    t_0(i) = interp1(rot_0,trq0,rot_mot(i));
end

% Figuras
figure
subplot(2,1,1)
plot(t,y(:,1),'k')
title('Deslocamento')
xlabel('Tempo [s]')
ylabel('Deslocamento [m]')
subplot(2,1,2)
plot(t,y(:,2), 'k' )

title('Velocidade')
xlabel('Tempo [s]')
ylabel('Velocidade [m/s]')

dados_teste0 = [y(:,2) transpose(rot_mot) transpose(t_0) a_it10];
save('dados_curva0.txt', 'dados_teste0','-ascii');

end % Final da simulação numérica com curva de torque maxima do carro

%% Aquisição ados para treinamento com curvas de torque diferentes
%%Dados para treinamento da IA com diferentes curvas de torque

if ativ_var == 11

% v12 = v_troca(1)*(1+((rand(1,1)-0.5)/5));
for i = 1:length(v_troca)
    v_troca40(i) = v_troca(i)*(1+((rand(1,1)-0.5)/10));
    v_troca60(i) = v_troca(i)*(1+((rand(1,1)-0.5)/10));
    v_troca78(i) = v_troca(i)*(1+((rand(1,1)-0.5)/10));
end

% v_troca40 = v_troca;
% v_troca60 = v_troca;
% v_troca78 = v_troca;

v_trocas_tcte = [40 v_troca40;60 v_troca60;78 v_troca78];
save('vel_troca_marcha.txt', 'v_trocas_tcte','-ascii');
[t_int40,y40] = ode45(@funcao_carro_adap,t,x0,[],m,Rd,it,idif,neta,v_troca40,40);
[t_int60,y60] = ode45(@funcao_carro_adap,t,x0,[],m,Rd,it,idif,neta,v_troca60,60);
[t_int78,y78] = ode45(@funcao_carro_adap,t,x0,[],m,Rd,it,idif,neta,v_troca78,78);

% Velocidades de troca de marcha são definidas dentro da função função_carro_adap

a_it40 = zeros(length(y40(:,2)),5);
rot_mot40 = 1:length(y40(:,2));
t_40 = (ones(length(y40(:,2)),1))*40;

for i = 1:length(y40(:,2))
    if y40(i,2) > 0 && y40(i,2) <= v_troca40(1) %0 à 20
        aux40 = it(1);
        a_it40(i,1) = 1;
    elseif y40(i,2) > v_troca40(1) && y40(i,2) <= v_troca40(2) %20 à 30
        aux40 = it(2);
        a_it40(i,2) = 1;
    elseif y40(i,2) > v_troca40(2) && y40(i,2) <= v_troca40(3) %30 à 40
        aux40 = it(3);
        a_it40(i,3) = 1;
    elseif y40(i,2) > v_troca40(3) && y40(i,2) <= v_troca40(4) %40 à 60
        aux40 = it(4);
        a_it40(i,4) = 1;
    elseif y40(i,2) > v_troca40(4) %60 e acima
        aux40 = it(5);
        a_it40(i,4) = 1;
    end
    rot_roda40 = y40(i,2)/Rd;
    rot_mot40(i) = (rot_roda40*30*aux40*idif)/pi;
end

a_it60 = zeros(length(y60(:,2)),5);
rot_mot60 = 1:length(y60(:,2));
t_60 = (ones(length(y60(:,2)),1))*60;
for i = 1:length(y60(:,2))
    if y60(i,2) > 0 && y60(i,2) <= v_troca60(1) %0 à 20
        aux60 = it(1);
        a_it60(i,1) = 1;
    elseif y60(i,2) > v_troca60(1) && y60(i,2) <= v_troca60(2) %20 à 30
        aux60 = it(2);
        a_it60(i,2) = 1;
    elseif y60(i,2) > v_troca60(2) && y60(i,2) <= v_troca60(3) %30 à 40
        aux60 = it(3);
        a_it60(i,3) = 1;
    elseif y60(i,2) > v_troca60(3) && y60(i,2) <= v_troca60(4) %40 à 60
        aux60 = it(4);
        a_it60(i,4) = 1;
    elseif y60(i,2) > v_troca60(4) %60 e acima
        aux60 = it(5);
    end
end

```

```

        a_it60(i,4) = 1;
    end
    rot_roda60 = y60(i,2)/Rd;
    rot_mot60(i) = (rot_roda60*30*aux60*idif)/pi;
end

a_it78 = zeros(length(y78(:,2)),5);
rot_mot78 = 1:1:length(y78(:,2));
t_78 = (ones(length(y78(:,2)),1))*78;

for i = 1:1:length(y78(:,2))
    if y78(i,2) > 0 && y78(i,2) <= v_troca78(1) %0 à 20
        aux78 = it(1);
        a_it78(i,1) = 1;
    elseif y78(i,2) > v_troca78(1) && y78(i,2) <= v_troca78(2) %20 à 30
        aux78 = it(2);
        a_it78(i,2) = 1;
    elseif y78(i,2) > v_troca78(2) && y78(i,2) <= v_troca78(3) %30 à 40
        aux78 = it(3);
        a_it78(i,3) = 1;
    elseif y78(i,2) > v_troca78(3) && y78(i,2) <= v_troca78(4) %40 à 78
        aux78 = it(4);
        a_it78(i,4) = 1;
    elseif y78(i,2) > v_troca78(4) %78 e acima
        aux78 = it(5);
        a_it78(i,4) = 1;
    end
    rot_roda78 = y78(i,2)/Rd;
    rot_mot78(i) = (rot_roda78*30*aux78*idif)/pi;
end

t_40 = (ones(length(y40(:,2)),1))*40;
t_60 = (ones(length(y60(:,2)),1))*60;
t_78 = (ones(length(y78(:,2)),1))*78;

dados_treino_num40 = [y40(:,2) transpose(rot_mot40) t_40 a_it40];
ind_rand_40 = randperm(length(dados_treino_num40));
dados_treino_num40 = dados_treino_num40(ind_rand_40,:);

dados_treino_num60 = [y60(:,2) transpose(rot_mot60) t_60 a_it60];
ind_rand_60 = randperm(length(dados_treino_num60));
dados_treino_num60 = dados_treino_num60(ind_rand_60,:);

dados_treino_num78 = [y78(:,2) transpose(rot_mot78) t_78 a_it78];
ind_rand_78 = randperm(length(dados_treino_num78));
dados_treino_num78 = dados_treino_num78(ind_rand_78,:);

dados_treino_num = [dados_treino_num40; dados_treino_num60; dados_treino_num78];
ind_rand_treino = randperm(length(dados_treino_num));
dados_treino_num = dados_treino_num(ind_rand_treino,:);
save('dados_treino_num_406078.txt', 'dados_treino_num','-ascii');

figure
subplot(2,1,1)
plot(t,y40(:,1))
title('Deslocamento (T = 40Nm)')
xlabel('Tempo [s]')
ylabel('Deslocamento[m]')

subplot(2,1,2)
plot(t,y40(:,2))
title('Velocidade (T = 40Nm)')
xlabel('Tempo [s]')
ylabel('Velocidade [m/s]')
%%
figure
subplot(2,1,1)
plot(t,y60(:,1))
title('Deslocamento (T = 60 Nm)')
xlabel('Tempo [s]')
ylabel('Deslocamento [m]')

subplot(2,1,2)
plot(t,y60(:,2))
title('Velocidade (T = 60 Nm)')
xlabel('Tempo [s]')
ylabel('Velocidade [m/s]')
%%
figure
subplot(2,1,1)
plot(t,y78(:,1))
title('Deslocamento (T = 78 Nm)')
xlabel('Tempo [s]')
ylabel('Deslocamento [m]')

subplot(2,1,2)
plot(t,y78(:,2))
title('Velocidade (T = 78 Nm)')
xlabel('Tempo [s]')
ylabel('Velocidade [m/s]')

end
%% Aquisição ados para teste com curvas de torque diferentes

if ativ_var == 13

    % resultados para teste da IA
    for i = 1:1:length(v_troca)
        v_troca55(i) = v_troca(i)*(1+((rand(1,1)-0.5)/5));
        v_troca65(i) = v_troca(i)*(1+((rand(1,1)-0.5)/5));
    end

    v_trocas_teste = [55 v_troca55;65 v_troca65];

```

```

save('vel_troca_marcha_teste.txt', 'v_trocas_teste','-ascii');

[t_int55,y55] = ode45(@funcao_carro_adap,t,x0,[],m,Rd,it,idif,neta,v_troca55,55);
[t_int65,y65] = ode45(@funcao_carro_adap,t,x0,[],m,Rd,it,idif,neta,v_troca65,65);

a_it55 = zeros(length(y55(:,2)),5);
rot_mot55 = 1:1:length(y55(:,2));
rotacao = 1:1:length(trq);
l_rot = length(rotacao);
trq55 = exp((rotacao/l_rot))*20 + 20 ;

for i = 1:1:length(y55(:,2))
    if y55(i,2) > 0 && y55(i,2) <= v_troca55(1) %0 à 20
        aux55 = it(1);
        a_it55(i,1) = 1;
    elseif y55(i,2) > v_troca55(1) && y55(i,2) <= v_troca55(2) %20 à 30
        aux55 = it(2);
        a_it55(i,2) = 1;
    elseif y55(i,2) > v_troca55(2) && y55(i,2) <= v_troca55(3) %30 à 40
        aux55 = it(3);
        a_it55(i,3) = 1;
    elseif y55(i,2) > v_troca55(3) && y55(i,2) <= v_troca55(4) %40 à 55
        aux55 = it(4);
        a_it55(i,4) = 1;
    elseif y55(i,2) > v_troca55(4) %55 e acima
        aux55 = it(5);
        a_it55(i,4) = 1;
    end

    rot_roda55 = y55(i,2)/Rd;
    rot_mot55(i) = (rot_roda55*30*aux55*idif)/pi;
    t_55(i) = interp1(rot_0,trq55,rot_mot55(i));
end

a_it65 = zeros(length(y65(:,2)),5);
rot_mot65 = 1:1:length(y65(:,2));
trq65 = (rotacao/l_rot)*30 + 45;

for i = 1:1:length(y65(:,2))
    if y65(i,2) > 0 && y65(i,2) <= v_troca65(1) %0 à 20
        aux65 = it(1);
        a_it65(i,1) = 1;
    elseif y65(i,2) > v_troca65(1) && y65(i,2) <= v_troca65(2) %20 à 30
        aux65 = it(2);
        a_it65(i,2) = 1;
    elseif y65(i,2) > v_troca65(2) && y65(i,2) <= v_troca65(3) %30 à 40
        aux65 = it(3);
        a_it65(i,3) = 1;
    elseif y65(i,2) > v_troca65(3) && y65(i,2) <= v_troca65(4) %40 à 65
        aux65 = it(4);
        a_it65(i,4) = 1;
    elseif y65(i,2) > v_troca65(4) %65 e acima
        aux65 = it(5);
        a_it65(i,4) = 1;
    end

    rot_roda65 = y65(i,2)/Rd;
    rot_mot65(i) = (rot_roda65*30*aux65*idif)/pi;
    t_65(i) = interp1(rot_0,trq65,rot_mot65(i));
end

dados_teste_num55 = [y55(:,2) transpose(rot_mot55) transpose(t_55) a_it55];
dados_teste_num65 = [y65(:,2) transpose(rot_mot65) transpose(t_65) a_it65];
dados_teste_num = [dados_teste_num55; dados_teste_num65];

save('dados_teste_num.txt', 'dados_teste_num','-ascii');

% Plota os graficos das curvas intermediarias
figure
subplot(2,1,1)

plot(t,y55(:,1))
title('Deslocamento (Torque de teste 1)')
xlabel('Tempo [s]')
ylabel('Deslocamento [m]')

subplot(2,1,2)
plot(t,y55(:,2))
title('Velocidade (Torque de teste 1)')
xlabel('Tempo [s]')
ylabel('Velocidade [m/s]')
%%
figure
subplot(2,1,1)
plot(t,y65(:,1))

title('Deslocamento (Torque de teste 2)')
xlabel('Tempo [s]')
ylabel('Deslocamento [m]')

subplot(2,1,2)
plot(t,y65(:,2))
title('Velocidade (Torque de teste 2)')
xlabel('Tempo [s]')
ylabel('Velocidade [m/s]')

end %final da simulação numérica com torque constante

```

```

%%
if ativ_var == 14
v_troca40 = v_troca;
v_troca60 = v_troca;
v_troca78 = v_troca;
v_troca55 = v_troca;
v_troca65 = v_troca;

[t_int,y] = ode45(@funcao_carro_adap,t,x0,[],m,Rd,it,ridif,neta,v_troca,0);
[t_int40,y40] = ode45(@funcao_carro_adap,t,x0,[],m,Rd,it,ridif,neta,v_troca40,40);
[t_int60,y60] = ode45(@funcao_carro_adap,t,x0,[],m,Rd,it,ridif,neta,v_troca60,60);
[t_int78,y78] = ode45(@funcao_carro_adap,t,x0,[],m,Rd,it,ridif,neta,v_troca78,78);
[t_int55,y55] = ode45(@funcao_carro_adap,t,x0,[],m,Rd,it,ridif,neta,v_troca55,55);
[t_int65,y65] = ode45(@funcao_carro_adap,t,x0,[],m,Rd,it,ridif,neta,v_troca65,65);

figure
plot(t ,y(:,1),'k',t ,y40(:,1),'r',t ,y55(:,1),'g',t ,y60(:,1),'b',t ,y65(:,1),'c',t ,y78(:,1),'m')
leg = legend('Torque Maximo', 'Torque = 40 Nm', 'Torque Variavel 1', 'Torque = 60' , 'Torque Variavel 2' , 'Torque = 78 Nm');
set(leg,'FontSize',12 )
title('Deslocamento obtido numericamente.', 'FontSize',14)
xlabel('Tempo [s]','FontSize',12,'FontWeight','Demi')
ylabel('Deslocamento [m]','FontSize',12,'FontWeight','Demi')

figure
plot(t ,y(:,2),'k',t ,y40(:,2),'r',t ,y55(:,2),'g',t ,y60(:,2),'b',t ,y65(:,2),'c',t ,y78(:,2),'m')
leg = legend('Torque Maximo', 'Torque = 40 Nm', 'Torque Variavel 1', 'Torque = 60' , 'Torque Variavel 2' , 'Torque = 78 Nm');
set(leg,'FontSize',12 )
title('Velocidade obtida numericamente.', 'FontSize',14)
xlabel('tempo [s]','FontSize',12,'FontWeight','Demi')
ylabel('Velocidade [m/s]','FontSize',12,'FontWeight','Demi')

end

```

FUNÇÃO MODELO

```

function dy=funcao_carro_adap(t,y,m,Rd,rit,ridif,neta,v_troca,sel_curva_trq)

% y é um vetor com as condições iniciais (posição, velocidade[entrar com dado em m/s])
global trq_tst rot_tst vel_tst mrc tmp_tst
% global trq_tst40 rot_tst40 vel_tst40 mrc40 tmp_tst40
% global trq_tst55 rot_tst55 vel_tst55 mrc55 tmp_tst55
% global trq_tst60 rot_tst60 vel_tst60 mrc60 tmp_tst60
% global trq_tst65 rot_tst65 vel_tst65 mrc65 tmp_tst65
% global trq_tst78 rot_tst78 vel_tst78 mrc78 tmp_tst78

trqrot_imp = (importdata('vetor_torque.txt')); %importa valores para curva de torque (cada posição do vetor
equivale a 1 rotação na escala do veiculo)
trq = transpose(trqrot_imp);
rotacao = 1:1:length(trq);
l_rot = length(rotacao);
trq55 = exp((rotacao/l_rot))*20 + 20 ;

trq65 = (rotacao/l_rot)*20 + 60;

if sel_curva_trq ==0

end

aux = 0;
if y(2) > 0 && y(2) <= v_troca(1)
aux = rit(1);
elseif y(2) > v_troca(1) && y(2) <= v_troca(2)
aux = rit(2);
elseif y(2) > v_troca(2) && y(2) <= v_troca(3)
aux = rit(3);
elseif y(2) > v_troca(3) && y(2) <= v_troca(4)
aux = rit(4);
elseif y(2) > v_troca(4)
aux = rit(5);
end
rot_rod = y(2)/Rd; % Vel linear / Rdinamico = rotação da roda em rad/s
rot = (rot_rod*30*aux*ridif)/pi;

%determina a rotação do motor de acordo com a velocidade do veiculo

%determina o valor de torque em função da rotação do motor
if sel_curva_trq == 0
trq0 = zeros(1,length(trq));
for i=1:1:length(trq)

```

```

        if i <= 1300
            if trq(i) <= 78.25
                trq0(i) = 78.25;
            elseif trq(i) > 78.25
                trq0(i) = trq(i);
            end
        end
    end
    if i > 1300
        trq0(i) = trq(i);
    end
end
torque_var = interp1(rotacao,trq0,rot);
tmp_tst = [tmp_tst t];
trq_tst = [trq_tst torque_var];
vel_tst = [vel_tst y(2)];
rot_tst = [rot_tst rot];
mrc = [mrc aux];
elseif sel_curva_trq == 40
    torque_var = 40;
    % tmp_tst40 = [tmp_tst40 t];
    % trq_tst40 = [trq_tst40 torque_var];
    % vel_tst40 = [vel_tst40 y(2)];
    % rot_tst40 = [rot_tst40 rot];
    % mrc40 = [mrc40 aux];
elseif sel_curva_trq == 55
    torque_var = interp1(rotacao,trq55,rot);
    % tmp_tst55 = [tmp_tst55 t];
    % trq_tst55 = [trq_tst55 torque_var];
    % vel_tst55 = [vel_tst55 y(2)];
    % rot_tst55 = [rot_tst55 rot];
    % mrc55 = [mrc55 aux];
elseif sel_curva_trq == 60
    torque_var = 60;
    % tmp_tst60 = [tmp_tst60 t];
    % trq_tst60 = [trq_tst60 torque_var];
    % vel_tst60 = [vel_tst60 y(2)];
    % rot_tst60 = [rot_tst60 rot];
    % mrc60 = [mrc60 aux];
elseif sel_curva_trq == 65
    torque_var = interp1(rotacao,trq65,rot);
    % tmp_tst65 = [tmp_tst65 t];
    % trq_tst65 = [trq_tst65 torque_var];
    % vel_tst65 = [vel_tst65 y(2)];
    % rot_tst65 = [rot_tst65 rot];
    % mrc65 = [mrc65 aux];
elseif sel_curva_trq == 78
    torque_var = 78;
    % tmp_tst78 = [tmp_tst78 t];
    % trq_tst78 = [trq_tst78 torque_var];
    % vel_tst78 = [vel_tst78 y(2)];
    % rot_tst78 = [rot_tst78 rot];
    % mrc78 = [mrc78 aux];
end
%dados do veiculo para determinar as resistencia

densar = 1.18;
cd = 0.333;
fo = 0.0096; %pg 81 genta morello V2 0.0096 pg 804 small car c
af = 2.15; %area frontal em m2
g = 9.81;
Ka = 2.8E-8;

R = ((0.5*densar*cd*af)*((( y(2))^2 ))) + (m*g*(fo + (Ka*((y(2))^2) ));%pg 160 Morello Genta V2

dyl=y(2);

%%Diferentes valores de torque usadas para teste
dy2 = ((( torque_var*aux*ridif*neta)/Rd) - R )/m);
% dy2 = ((( 50*aux*ridif*neta)/Rd) - R )/m);
% dy2 = ((( 60*aux*ridif*neta)/Rd) - R )/m);
% dy2 = ((( 70*aux*ridif*neta)/Rd) - R )/m);
dy = [ dyl; dy2];

%%
% Arquivo com dados do teste
% Velocidade, rotaçao, torque, marcha

```

ALGORITMO UTILIZADO PARA PREPARAÇÃO DOS DADOS EXPERIMENTAIS

```

% A0ext_teste1 = speed_obd2/max(speed_obd2);
% A0ext_teste2 = rpm_obd2/(max(rpm_obd2)-min(rpm_obd2));
% A0ext_teste3 = load_obd2/(max(load_obd2)-min(load_obd2));
% A0ext_teste4 = it_est / (max(it_est)-min(it_est));
%1 = 15,18; 2 = 8,34; 3 = 5,66; 4 = 4,19; 5 = 3,09;

%Parametros de normalização dos valores dos testes experimentais
load 'Medicao_Perini_Blumenau.txt'
load 'Medicao_Blumenau_AntonioRamos.txt'
val_test1 = Medicao_Perini_Blumenau;
val_test2 = Medicao_Blumenau_AntonioRamos;
val_test = [val_test1; val_test2];

speed_all=val_test(:,6);
rpm_all=val_test(:,7);
load_all=142.20*val_test(:,8)/100;

speed_max_all = max(speed_all);
rpm_min_all = min(rpm_all);
rpm_max_all = max(rpm_all);
load_min_all = min(load_all);
load_max_all = max(load_all);

if aux == 10
b = 1;
for w = 1:length(it_est)
if it_est(w,1) ~= 0
limp_dado_exp_1(b,2) = speed_obd2(w,1); %(max(speed_obd2));
limp_dado_exp_1(b,3) = rpm_obd2(w,1);
limp_dado_exp_1(b,4) = load_obd2(w,1);
limp_dado_exp_1(b,5) = it_est(w,1);
b = b+1;
end
end

limp_dado_exp_1(:,1) =1:length(limp_dado_exp_1(:,2));
figure
subplot(3,1,1)
plot(limp_dado_exp_1(:,1),limp_dado_exp_1(:,4),'k','LineWidth',2)
title('Torque demandado do motor','FontSize',14,'FontWeight','Demi')
ylabel('Torque [Nm]','FontSize',12)
grid on
subplot(3,1,2)
plot(limp_dado_exp_1(:,1),limp_dado_exp_1(:,2),'k',limp_dado_exp_1(:,1),limp_dado_exp_1(:,5),'r','LineWidth',2)
ylabel('Velocidade [km/h]','FontSize',12)
grid on
subplot(3,1,3)
plot(limp_dado_exp_1(:,1),limp_dado_exp_1(:,3),'k','LineWidth',2)
ylabel('Rotação do motor [rpm]','FontSize',12)
grid on

%%%%%%%%%% dados para teste
cont1 = 0;
cont2 =0;
b = 1;

for w = 1:length(it_est)
if it_est(w,1) ~= 0
dado_exp_1(b,1) = speed_obd2(w,1)/speed_max_all; %(max(speed_obd2));
dado_exp_1(b,2) = rpm_obd2(w,1)/(rpm_max_all-rpm_min_all);
dado_exp_1(b,3) = load_obd2(w,1)/(load_max_all-load_min_all);
if it_est(w,1) == 15.18;
dado_exp_1(b,4:8) = [1 0 0 0 0];
b=b+1;
elseif it_est(w,1) == 8.34;
dado_exp_1(b,4:8) = [0 1 0 0 0];
b=b+1;
elseif it_est(w,1) == 5.66;
dado_exp_1(b,4:8) = [0 0 1 0 0];
b=b+1;
elseif it_est(w,1) == 4.19;
dado_exp_1(b,4:8) = [0 0 0 1 0];
b=b+1;
elseif it_est(w,1) == 3.09;
dado_exp_1(b,4:8) = [0 0 0 0 1];
b=b+1;
end
end
end
save('dados_treinamento_exp.txt', 'dado_exp_1', '-ascii');

end

```

```

%-----
if aux == 11
%   in_dado_teste_2 = zeros(length(it_est),5);

b = 1;
for w = 1:length(it_est)
    if it_est(w,1) ~= 0

        limp_dado_exp_2(b,2) = speed_obd2(w,1); %(max(speed_obd2));
        limp_dado_exp_2(b,3) = rpm_obd2(w,1);
        limp_dado_exp_2(b,4) = load_obd2(w,1);
        limp_dado_exp_2(b,5) = it_est(w,1);
        b = b+1;
    end
end
limp_dado_exp_2(:,1) = 1:length(limp_dado_exp_2(:,2));
figure
subplot(3,1,1)
plot(limp_dado_exp_2(:,1),limp_dado_exp_2(:,4),'k','LineWidth',2)
title('Torque demandado do motor','FontSize',14,'FontWeight','Demi')
ylabel('Torque [Nm]','FontSize',12)
grid on
subplot(3,1,2)
plot(limp_dado_exp_2(:,1),limp_dado_exp_2(:,2),'k',limp_dado_exp_2(:,1),limp_dado_exp_2(:,5),'r','LineWidth',2)
ylabel('Velocidade [km/h]','FontSize',12)
grid on
subplot(3,1,3)
plot(limp_dado_exp_2(:,1),limp_dado_exp_2(:,3),'k','LineWidth',2)
ylabel('Rotação do motor [rpm]','FontSize',12)
grid on

cont1 = 0;
cont2 = 0;
b = 1;
%***** Dados para treinamento e teste
for w = 1:length(it_est)
    if it_est(w,1) ~= 0
        dado_exp_2(b,1) = speed_obd2(w,1)/speed_max_all;%(max(speed_obd2));
        dado_exp_2(b,2) = rpm_obd2(w,1)/(rpm_max_all-rpm_min_all);%(max(rpm_obd2)-min(rpm_obd2));
        dado_exp_2(b,3) = load_obd2(w,1)/(load_max_all-load_min_all);%(max(load_obd2)-min(load_obd2));
        if it_est(w,1) == 15.18;
            dado_exp_2(b,4:8) = [1 0 0 0 0];
            b=b+1;
        elseif it_est(w,1) == 8.34;
            dado_exp_2(b,4:8) = [0 1 0 0 0];
            b=b+1;
        elseif it_est(w,1) == 5.66;
            dado_exp_2(b,4:8) = [0 0 1 0 0];
            b=b+1;
        elseif it_est(w,1) == 4.19;
            dado_exp_2(b,4:8) = [0 0 0 1 0];
            b=b+1;
        elseif it_est(w,1) == 3.09;
            dado_exp_2(b,4:8) = [0 0 0 0 1];
            b=b+1;
        end
    end
end
c = 1;

save('dados_teste_exp.txt', 'dado_exp_2', '-ascii');
end

```

ALGORITMO PRINCIPAL DE TREINO E TESTE DA REDE PERCEPTRON MODIFICADO

```

% Example of neural network
% 25/08/2023
% SJI

clear all
close all

```

```

% -----
% Input
% -----
% Example of wine classification
% http://neuroph.sourceforge.net/tutorials/wines1/WineClassificationUsingNeuralNetworks.html
% Example of wine classification
% http://neuroph.sourceforge.net/tutorials/wines1/WineClassificationUsingNeuralNetworks.html
% load wine_classification.txt
% data=wine_classification;

%%% Ativação de cada seção do código
%%
load vetor_torque.txt
% Variavel de ativação (valores diferentes ativam diferentes etapas do algoritmo)
var_ativ = 11;
%%
trqrot_imp = vetor_torque; %importa valores para curva de torque (cada posição do vetor equivale a 1
rotação na escala do veiculo)
trq = transpose(trqrot_imp);

% for i=1:length(trq)
%     if trq(i) <= 78.25
%         trq0(i) = 78.25;
%     elseif trq(i) > 78.25
%         trq0(i) = trq(i);
%     end
% end

trq_max = max(trq);

load dados_curva0.txt
datan0 = dados_curva0;
v_max0 = max(datan0(:,1));
rot_max0 = max(datan0(:,2));

% Valores de ativação:
% 10 : Realiza treinamento e teste com dados numéricos do modelo utilizandoa curva de torque do polo 1.0
msi 2017 em 100% WOT
% 11 : Realiza treinamento com dados numéricos do modelo com curva de torque de valor constante ( 50, 60 e
70 Nm) e teste com curvas de valor constante intermediario (55 e 65 Nm)
% 12 : Realiza treinamento e teste com 2 conjuntos de dados experimentais disponibilizados pelo professor
% 94 : Valor de segurança, para não executar o modelo acidentalmente durante modificações

if var_ativ == 94
'Nada'
break
end

if var_ativ == 10
%%% Dados partir da curva de torque com 100% WOP
load dados_curva0.txt
datan0 = dados_curva0;
% datan0(:,1)=(datan0(:,1)-min(datan0(:,1)))/(max(datan0(:,1))-min(datan0(:,1)));
% datan0(:,2)=(datan0(:,2)-min(datan0(:,2)))/(max(datan0(:,2))-min(datan0(:,2)));
% datan0(:,3)=(datan0(:,3)-min(datan0(:,3)))/(max(datan0(:,3))-min(datan0(:,3)));

%%% Normalização de torque é feita com base nos valores da curva original
datan0(:,1)=(datan0(:,1))/(v_max0);
datan0(:,2)=(datan0(:,2))/(rot_max0);
datan0(:,3)=(datan0(:,3))/(trq_max);
end

if var_ativ == 11
%%% Dados para treinamento da curva com curvas de torque constante
load vel_troca_marcha.txt
% os valores da primeira coluna se referem ao torque e os demais às
% velocidades de troca de marcha. [val_curva_torque v12 v23 v34 v45]
v_troca_marcha = vel_troca_marcha
load vel_troca_marcha_teste.txt
v_troca_teste = vel_troca_marcha_teste

load dados_treino_num_406078.txt
data_trn_num = dados_treino_num_406078;
data_trn_num(:,1)=(data_trn_num(:,1))/(v_max0); % Normalização velocidade
data_trn_num(:,2)=(data_trn_num(:,2))/(rot_max0); % Normalização rotação
data_trn_num(:,3)=(data_trn_num(:,3))/(trq_max); % Normalização Torque

%%% Dados para teste do treinamento com dados numéricos

load dados_teste_num.txt
data_tst_num = dados_teste_num;
data_tst_num(:,1)=(data_tst_num(:,1))/(v_max0);
data_tst_num(:,2)=(data_tst_num(:,2))/(rot_max0);
data_tst_num(:,3)=(data_tst_num(:,3))/(trq_max);

```

```

end

if var_ativ == 12
%Treinamento e teste com dados experimentais
%Não precisa normalizar pois os dados ja foram normalizados
load dados_treinamento_exp.txt
load dados_teste_exp.txt

data_trn_exp = dados_treinamento_exp;
data_tst_exp = dados_teste_exp;

end

%%
if var_ativ == 10
% Input: [Nvar_input,N_samples]
input_data=datan0(:,1:end-5)';
output_data=datan0(:,end-4:end)';
end

if var_ativ == 11
%Treinamento com curvas de torque 50 60 e 70
input_data=data_trn_num(:,1:end-5)';
output_data=data_trn_num(:,end-4:end)';

input_data_teste=data_tst_num(:,1:end-5)';
output_data_teste=data_tst_num(:,end-4:end)';
end

if var_ativ == 12
%Treinamento com dados experimentais
input_data = data_trn_exp(:,1:end-5)';
output_data = data_trn_exp(:,end-4:end)';

input_data_teste = data_tst_exp(:,1:end-5)';
output_data_teste = data_tst_exp(:,end-4:end)';

end

%% Run Neural Network (Backward)
neta=0.3;
mi=0.9;
[W11,W12,b11,b12]=neuralnetwork_perceptron(input_data',output_data',neta,mi);

%% Test data
% Forward calculation - estimation
if var_ativ == 10
%Validação inicial com a curva WOT 100%
input_test=input_data';
output_test=output_data';
output_test_estimated=neuralnetwork_perceptron_forward(input_test,output_test,W11,W12,b11,b12);
end

if var_ativ == 11 | var_ativ == 12
input_test=input_data_teste';
output_test=output_data_teste';
output_test_estimated=neuralnetwork_perceptron_forward(input_test,output_test,W11,W12,b11,b12);
end

```