CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

ANÁLISE DE REGRESSÃO BASEADO EM APRENDIZADO DE MÁQUINA PARA A CARACTERIZAÇÃO DA SEVERIDADE DE FALHAS EM MOTORES DIESEL

Denys Pestana Viana

Tese apresentada ao Programa de Pós-Graduação em Instrumentação e Óptica Aplicada, do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos necessários à obtenção do título de doutor.

Orientador: Amaro Azevedo de Lima
Coorientador: Diego Barreto Haddad

Rio de Janeiro,
Junho 2024

ANÁLISE DE REGRESSÃO BASEADO EM APRENDIZADO DE MÁQUINA PARA A CARACTERIZAÇÃO DA SEVERIDADE DE FALHAS EM MOTORES DIESEL

Tese apresentada ao Programa de Pós-Graduação em Instrumentação e Óptica Aplicada, do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos necessários à obtenção do título de doutor.

Denys Pestana Viana

Banca Examinadora:

Presidente, Prof. Amaro Azevedo de Lima, Ph.D. (CEFET/RJ) (Orientador)

Prof. Diego Barreto Haddad, D.Sc. (CEFET/RJ) (Coorientador)

Ana Lucia Ferreira e Barros

Assinado de forma digital por Ana Lucia Ferreira e Barros
Dados: 2024.08.06 10:36:18 -03'00'

Profa. Ana Lucia Ferreira de Barros, D.Sc. (CEFET/RJ)

FELIPE DA ROCHA HENRIQUES:08668257706

Assinado de forma digital por FELIPE DA ROCHA HENRIQUES:08668257706
Dados: 2024.08.06 17:38:04 -03'00'

Prof. Felipe da Rocha Henriques, D.Sc. (CEFET/RJ)

Prof. Ulisses Admar Barbosa Vicente Monteiro, D.Sc. (UFRJ)

Prof. Ricardo Homero Ramírez Gutiérrez, D.Sc. (UEA)

Rio de Janeiro,

Junho 2024

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

V614 Viana, Denys Pestana
Análise de regressão baseado em aprendizado de máquina para a caracterização da severidade de falhas em motores diesel / Denys Pestana Viana – 2024.
123f : il. (algumas color) + apêndice , enc.

Tese (Doutorado). Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2024.
Bibliografia: 115-123.
Orientador: Amaro Azevedo de Lima
Co-orientador: Diego Barreto Haddad

1. Motor diesel. 2. Identificação de falhas. 3. Aprendizado de máquina. 4. Análise de vibração. I. Lima, Amaro Azevedo de (Orient.). II. Haddad, Diego Barreto (Co-orient.). III. Título.

CDD 621.436

Elaborada pelo bibliotecário Leandro Mota de Menezes – CRB/7 nº 5281

DEDICATÓRIA

Dedico este trabalho à Deus e aos meus filhos, Alice Peixoto Pestana e Pedro Peixoto Pestana, que foram fatores motivadores para a realização e finalização deste trabalho.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de pessoal de Nível Superior – Brasil (CAPES) – código de financiamento 001. A minha dedicação exclusiva ao projeto de doutorado foi essencial para a obtenção dos resultados aqui apresentados.

Agradeço a Deus por me dar força, sabedoria e saúde ao longo desta jornada acadêmica. Sem a Sua graça, nada disso seria possível.

Aos meus filhos, pelo amor, paciência e compreensão nos momentos de ausência e dedicação aos estudos. Vocês são minha maior inspiração.

À minha esposa Andressa, pelo apoio incondicional, carinho e incentivo constante. Sua presença foi fundamental para a realização deste trabalho.

À minha mãe Rosangela de Araújo Pestana, por todo incentivo e suporte.

Aos meus orientadores, Amaro Azevedo de Lima e Diego Barreto Haddad, pelo apoio, conhecimento compartilhado, pela orientação precisa e pelo incentivo ao longo de todo o processo. Em especial, agradeço às professoras Ana Lucia Ferreira de Barros e Luciana Faletti Almeida, pelo suporte, sabedoria e dedicação que foram essenciais para o desenvolvimento desta pesquisa.

Agradeço também ao Professores Luís Domingues Tomé Jardim Tarrataca e Milena Faria Pinto, pelo apoio, pelas contribuições, discussões enriquecedoras e pelo suporte técnico, durante a elaboração dos artigos.

Ao professor Fabio Augusto de Alcantara Andrade, pelo suporte dado durante a submissão dos trabalhos à revista Machines. Seu Suporte foi fundamental.

Ao meu colega Dionísio Henrique Carvalho de Sá Só Martins por todo suporte e apoio mútuo durante o desenvolvimento deste trabalho.

À equipe do LEDAV, pela colaboração e pelo ambiente de trabalho inspirador, que foi fundamental para o desenvolvimento deste projeto.

A todos, meu sincero agradecimento.

RESUMO

Análise de Regressão Baseado em Aprendizado de Máquina para a Caracterização da Severidade de Falhas em Motores Diesel

Os motores Diesel utilizados em ambientes industriais e navios estão expostos a condições agressivas, o que os torna ainda mais propensos a falhas. Como esses equipamentos são a principal fonte de energia de um navio, o custo de manutenção dos motores diesel pode representar entre 10% a 20% do valor total da embarcação. Portanto, é vital ter um sistema de identificação de falhas que permita realizar a manutenção de maneira eficiente e a tempo hábil, evitando dano à produção, aos trabalhadores e ao navio em si. A complexidade e a sensibilidade das máquinas diesel exigem uma abordagem meticulosa para a detecção precoce de falhas, a fim de evitar interrupções não planejadas e reduzir custos de manutenção. A simulação oferece um ambiente controlado para gerar conjuntos de dados diversificados, abrangendo várias condições operacionais e cenários de falhas, o que seria difícil de reproduzir em ambientes reais. Além disso, permite capturar detalhadamente o comportamento do sistema, fornecendo uma base sólida para a análise das características e padrões associados às falhas. Este estudo visa identificar as seguintes falhas, bem como a sua severidade em motores diesel: operação normal, redução de pressão no coletor de admissão, redução da taxa de compressão nos cilindros e redução da quantidade de combustível injetado nos cilindros, para auxiliar a manutenção preditiva. Para isso, foi desenvolvido um modelo de simulação de falhas baseado em uma análise das variações de pressão no interior dos cilindros e da resposta de vibração torcional da árvore de manivelas. Uma base de dados foi construída através da criação e validação de um gêmeo digital de um motor diesel de 6 cilindros, com a emulação de 3500 cenários distintos de falha, nomeada como 3500-DEFault e disponibilizada publicamente para a comunidade científica. Foi utilizada a transformada discreta de Fourier a partir dos parâmetros operacionais de pressão, temperatura e vibração, para estimar amplitude, meias ordens e fase dos sinais de vibração torcional, bem como os parâmetros estatísticos média amostral e valor máximo dos sinais de pressão, visando extrair características que serviram de entrada para as técnicas de aprendizado de máquina: Perceptron Multicamada (MLP), *Random Forest* (RF), Máquina de Vetores de Suporte (SVM) e Modelos de Regressão de Processo Gaussiano (GPRM). Esses métodos ajudaram a discriminar a condição operativa da máquina e a severidade da falha associada. O melhor desempenho de ajuste do grau de severidade associado à falha foi obtido com o regressor GPRM, com o valor da raiz do erro quadrático médio (RMSE) de $0{,}015 \pm 0{,}001\%$.

Palavras-chave: Aprendizado de máquina; Identificação de falhas; Análise de vibração; Parâmetros operacionais

ABSTRACT

Regression Analysis Based on Machine Learning for the Characterization of Severity of Failures in Diesel Engines

Diesel engines used in industrial environments and ships are exposed to harsh conditions, making them even more prone to failures. As these equipment are the main source of energy for a ship, the maintenance cost of diesel engines can represent between 10% to 20% of the total value of the vessel. Therefore, it is vital to have a fault identification system that allows maintenance to be carried out efficiently and in a timely manner, avoiding damage to production, workers, and the ship itself. The complexity and sensitivity of diesel engines require a meticulous approach for early fault detection to prevent unplanned downtimes and reduce maintenance costs. Simulation provides a controlled environment to generate diverse datasets, covering various operational conditions and fault scenarios, which would be difficult to reproduce in real environments. Moreover, it allows for detailed capture of the system behavior, providing a solid basis for analyzing the characteristics and patterns associated with faults. This study aims to identify the following faults, as well as their severity, in diesel engines: normal operation, reduced pressure in the intake manifold, reduced compression rate in the cylinders, and reduced amount of fuel injected into the cylinders, to improve predictive maintenance. To achieve this, a fault simulation model was developed based on an analysis of pressure variations inside the cylinders and the torsional vibration response of the crankshaft. A database was built through the creation and validation of a digital twin of a 6-cylinder diesel engine, with the emulation of 3500 distinct fault scenarios, named 3500-DEFault. This fault signal database was made publicly available to the scientific community. The discrete Fourier transform was used based on the operational parameters of pressure, temperature, and vibration to estimate amplitude, half orders, and phase of the torsional vibration signals, as well as the statistical parameters of sample mean and maximum value of the pressure signals, aiming to extract features that served as input for machine learning techniques: Multilayer Perceptron (MLP), Random Forests (RF), Support Vector Machine (SVM), and Gaussian Process Regression Models (GPRM). These methods helped discriminate the operational condition of the machine and the severity of the associated fault. The best performance in adjusting the degree of severity associated with the fault was obtained with the GPRM regressor, with a root mean square error (RMSE) value of $0.015 \pm 0.001\%$.

Keywords: Machine learning; Fault identification; Vibration analysis; Operational parameters

LISTA DE ILUSTRAÇÕES

Figura 1 – Motor Diesel MWM Acteon 6.12 TCE. 31

Figura 2 – Volume de controle da análise termodinâmica. 34

Figura 3 – Validação do modelo termodinâmico desenvolvido para diferentes rotações: (a) 1000, (b) 1200, (c) 2100 e (d) 2500 RPM. Nos gráficos $P_1(\theta)^{exp}$ e $P_1(\theta)^{sim}$ referem-se aos dados experimentais e simulados, respectivamente. 36

Figura 4 – Curvas de pressão de acordo com a ordem de ignição para 2500 RPM. 37

Figura 5 – Árvore de Manivelas do Motor MWM Acteon 6.12TCE (Gutiérrez, 2016). 38

Figura 6 – Modelo equivalente de Massa concentrada da árvore de Manivelas do Motor MWM Acteon 6.12TCE (Gutiérrez, 2016). 38

Figura 7 – Esquema do sistema pistão-biela-manivela. 39

Figura 8 – Curvas de torque de acordo com a ordem de ignição para 2500 RPM. 40

Figura 9 – Vibração torcional para o sinal normal da árvore de manivelas a 2500 RPM. 41

Figura 10 – Diagrama de blocos das falhas do motor Diesel e suas consequências. 45

Figura 11 – Variação dos parâmetros operativos do motor Diesel, para falha da queda porcentual da pressão no coletor de admissão, para diferentes níveis de severidade e para a velocidade de rotação fixa em 1000 RPM: (a) pressão e (b) vibração torcional. Os valores percentuais mencionados correspondem aos índices de falha utilizados nos experimentos. 47

Figura 12 – Variação dos parâmetros operativos do motor Diesel, para falha da queda porcentual da compressão no cilindro 1, para diferentes níveis de severidade e para a velocidade de rotação fixa em 1000 RPM: (a) pressão e (b) vibração torcional. Os valores percentuais mencionados correspondem aos índices de falha utilizados nos experimentos. 47

Figura 13 – Variação dos parâmetros operativos do motor Diesel, para falha da queda porcentual da pressão no common rail, para diferentes níveis de severidade e para a velocidade de rotação fixa em 1900 RPM: (a) pressão e (b) vibração torcional. Os valores percentuais mencionados correspondem aos índices de falha utilizados nos experimentos. 48

Figura 14 – Variação dos parâmetros operativos do motor Diesel, para falha da queda porcentual da massa de ar admitido no cilindro 1, para diferentes níveis de severidade e para a velocidade de rotação fixa em 1900 RPM: (a) pressão e (b) vibração torcional. Os valores percentuais mencionados correspondem aos índices de falha utilizados nos experimentos. 48

Figura 15 – Variação dos parâmetros operativos do motor Diesel, para falha da queda porcentual da massa de combustível injetado no cilindro 1, para diferentes níveis de severidade e para a velocidade de rotação fixa em 2500 RPM: (a) pressão e (b) vibração torcional. Os valores percentuais mencionados correspondem aos índices de falha utilizados nos experimentos. 49

Figura 16 – Variação dos parâmetros operativos do motor Diesel, para falha do atraso porcentual do início da injeção no cilindro 1, para diferentes níveis de severidade e para a velocidade de rotação fixa em 2500 RPM: (a) pressão e (b) vibração torcional. Os valores percentuais mencionados correspondem aos índices de falha utilizados nos experimentos. 49

Figura 17 – Diagrama de bloco do processo de adição de ruído: (a) $M(t)$, (b) $P(\theta)$ e (c) $T(\theta)$. $\nu(t)$ e $\nu(\theta)$ representam o sinal de ruído branco com relação ao tempo e ângulo da árvore de manivelas, considerando diferentes níveis L. 55

Figura 18 – Função densidade de probabilidade da variável $\nu(\cdot)$ para L diferentes níveis de ruído. 56

Figura 19 – Processo de aplicação de ruído aditivo: (a) $M(t)$ com 15 dB, (b) $P(\theta)$ com 15 dB, (c) $T(\theta)$ com 15 dB, (d) $M(t)$ com 0 dB, (e) $P(\theta)$ com 0 dB e (f) $T(\theta)$ com 0 dB. No gráfico acima, a linha cinza é o sinal com adição de ruído e a linha preta é o sinal original sem ruído. A nova variável com AWGN $\check{v}_{s_i}(\cdot)$, é $\check{v}_{s_i}(\cdot) = v_{s_i}(\cdot) + \ nu(\cdot)$, onde $\nu(\cdot)$ é o ruído gaussiano branco com $L = [15, 0]$ dB SNR. 57

Figura 20 – Técnica do k-fold, onde os blocos cinza correspondem aos conjuntos de teste, enquanto os blocos brancos correspondem aos conjuntos de treinamento. 59

Figura 21 – Esquema de único neurônio artificial. 62

Figura 22 – Rede perceptron multicamada (MLP) (a) treino utilizando pares rotulados $\mathbf{x_i}$ e y_i para aprender pesos e (b) teste utilizando dados não rotulados $\mathbf{x_i}$ para fazer previsões. 63

Figura 23 – Diagrama do algoritmo *Random Forest*. 65

Figura 24 – SVM treinado com amostras de duas classes. Os círculos vermelhos são chamados de vetores de suporte, visando formar o hiperplano que separa as duas classes do espaço de características. 68

Figura 25 – ε-função de perda insensível. 69

Figura 26 – Box plot do banco de dados 3500-DEFault para subconjuntos de características $F(k)$, $A(k)$, $P(k)$, μ_{p_i} e M_{p_i}, respectivamente. Para diferentes níveis de AWGN: (a) 60, (b) 30, (c) 15 e (d) 0 dB. O gráfico mostra a mediana, o quartil 25%, o quartil 75% e a faixa inferior e superior (bigodes), os quais são o máximo e o mínimo para cada distribuição, respectivamente. 81

Figura 27 – Diagrama de blocos do sistema proposto composto pelo bloco do modelo dinâmico, o bloco de geração do banco de dados (dividido em falha e criação de sinais normais), o bloco do processo AWGN, o bloco de extração de características e o bloco de regressão. 83

Figura 28 – Diagrama de blocos do programa DEFault. 84

Figura 29 – Gráfico de erro de várias curvas de sintonia com nível AWGN $L = 60$ dB SNR.: (a), (d), (g) e (j) para ΔP_r e regressores MLP, RF, SVM e GPR, respectivamente; (b), (e), (h) e (k) para Δr_1 e regressores MLP, RF, SVM e GPR, respectivamente; (c), (f), (i) e (l) para Δm_{c_1} e regressores MLP, RF, SVM e GPR, respectivamente. Nos gráficos acima, a linha pontilhada representa μ_{RMSE} e as barras representam σ_{RMSE} após a validação cruzada de 5 vezes. 98

Figura 30 – Processo de aplicação de 15 dB SNR de ruído aditivo na variável de torque. No gráfico acima, a linha tracejada representa o sinal com adição de ruído e a linha reta representa o sinal original sem ruído. A nova variável de torque com AWGN ($\{\breve{M}(t)\}$) is $\{\breve{M}(t)\} = \{M(t)\}+\nu(t)$, onde $\nu(t)$ é ruído gaussiano branco com SNR de 15dB. O processo é o mesmo para as demais variáveis (pressão e vibração torcional). 100

Figura 31 – Testes de regressão com nível AWGN $L = 60$ dB SNR: (a), (d), (g) e (j) para ΔP_r e regressores MLP, RF, SVM e GPRM , respectivamente; (b), (e), (h) e (k) para Δr_1 e regressores MLP, RF, SVM e GPR, respectivamente; e (c), (f), (i) e (l) para Δm_{c_1} e regressores MLP, RF, SVM e GPR, respectivamente. No gráfico acima, a linha preta representa a previsão perfeita e os pontos pretos representam os elementos verdadeiros (eixo x) versus elementos previstos (eixo y). Todos os gráficos possuem um intervalo entre zero e cinquenta (eixos x e y), ou seja, com amplitudes iguais às severidades da base de dados. 110

Figura 32 – Testes de regressão com nível AWGN $L = 0$ dB SNR.: (a), (d), (g) e (j) para ΔP_r e regressores MLP, RF, SVM e GPR , respectivamente; (b), (e), (h) e (k) para Δr_1 e regressores MLP, RF, SVM e GPR, respectivamente; e (c), (f), (i) e (l) para Δm_{c_1} e regressores MLP, RF, SVM e GPR, respectivamente. No gráfico acima, a linha preta representa a previsão perfeita e os pontos pretos representam os elementos verdadeiros (eixo x) versus elementos previstos (eixo y). Todos os gráficos possuem um intervalo entre zero a cinquenta (eixos x e y), ou seja, com amplitudes iguais às severidades da base de dados. 111

LISTA DE TABELAS

Tabela 1 – Principais contribuições na detecção de falhas em motores Diesel. 25

Tabela 2 – Especificações técnicas do motor Diesel MWM Acteon 6.12 TCE. 32

Tabela 3 – Pressões experimentais e simuladas. 35

Tabela 4 – Comparação de torque. 40

Tabela 5 – Valores selecionados dos hiperparâmetros de cada regressor. 97

Tabela 6 – Matriz de confusão com reconhecimento de padrões de 4 classes usando o vetor de características V_f com ruído de 60 dB e RF. 101

Tabela 7 – Matriz de confusão com reconhecimento de padrões de 4 classes usando o vetor de características V_f com ruído de 60 dB e RNA. 101

Tabela 8 – Matriz de confusão com reconhecimento de padrões de 8 classes usando o vetor de características V_f com ruído de 60 dB e RF. 102

Tabela 9 – Matriz de confusão com reconhecimento de padrões de 8 classes usando o vetor de característica V_f com ruído de 60 dB e RNA. 102

Tabela 10 – Matriz de confusão com reconhecimento de padrões de severidade de 4 classes usando o vetor de característica V_f com ruído de 60 dB e RF. 103

Tabela 11 – Matriz de confusão com reconhecimento de padrões de severidade de 4 classes usando o vetor de característica V_f com ruído de 60 dB e RNA. 103

Tabela 12 – Resultados da acurácia de predição para os classificadores RF e ANN com SNR de 60, 30, 15 e 0 dB usando o vetor de características V_f. 104

Tabela 13 – Tabela de resumo com falhas e níveis de severidade abordados para cada cenário/caso de entrada. 105

Tabela 14 – Tabela de erro máximo na identificação da severidade de cada variável usando a técnica LMLS e os resultados da classificação usando RF para os classificadores 4-c, 8-c, 14-c e S4, para diferentes níveis de ruído e 2500 RPM. 105
Tabela 15 – RMSE dos regressores para cada parâmetro (FP), considerando 60 dB AWGN. 107
Tabela 16 – RMSE dos regressores para cada parâmetro (FP), considerando 30 dB AWGN. 108
Tabela 17 – RMSE dos regressores para cada parâmetro (FP), considerando 15 dB AWGN. 108
Tabela 18 – RMSE dos regressores para cada parâmetro (FP), considerando 0 dB AWGN. 109
Tabela 19 – Tabela de erro máximo na identificação da severidade de cada variável usando a técnica LMLS e GPRM 112

LISTA DE ALGORITMOS

Algoritmo 1 – PROGRAMA DEFAULT 86

Algoritmo 2 – Função **pressao.cal** 131

Algoritmo 3 – Subfunção **cilindropt009** 133

Algoritmo 4 – Função **dif.volume** 135

Algoritmo 5 – Função **parametros.modais** 137

Algoritmo 6 – Função **delay** 138

Algoritmo 7 – Torque 140

Algoritmo 8 – **replica.voltas** 142

Algoritmo 9 – parametros.modais 144

Algoritmo 10 – Técnica do algoritmo Runge-Kutta. 145

Algoritmo 11 – vibração 146

Algoritmo 12 – vibracao.torcional 147

Algoritmo 13 – ordem.espectro 148

Algoritmo 14 – tensoes_global 150

Algoritmo 15 – pressao.media1 152

Algoritmo 16 – espectro.exemplo 153

Algoritmo 17 – barra.progresso 154

SUMÁRIO

Introdução **19**

1 Revisão bibliográfica **23**

2 Desenvolvimento do modelo termodinâmico e dinâmico do motor Diesel **29**

2.1 Especificações Técnicas do Motor Diesel 31

2.2 Modelo termodinâmico 32

2.2.1 Validação do modelo 34

2.3 Modelo dinâmico torcional da árvore de manivelas 35

2.3.1 Verificação do modelo 40

2.4 Modelo de simulação de falha 41

2.4.1 Severidade das falhas 43

2.4.2 Características das falhas 43

2.4.3 Metodologia do modelo de simulação de falhas 44

2.5 Considerações do capítulo 47

3 Materiais e métodos **51**

3.1 Conjunto de dados 52

3.2 Classes de falhas 53

3.2.1 Normal (sem falhas) 53

3.2.2 Redução de pressão no coletor de admissão 53

3.2.3 Redução de razão de compressão nos cilindros 54

3.2.4 Redução da quantidade de combustível injetado nos cilindros. 54

3.3 Processo de Ruído Branco Aditivo Gaussiano 54

3.4 Normalização do conjunto de dados 57

3.4.1 Normalização mínimo-máximo 58

3.5 Particionamento o conjunto de dados 58

3.6 Regressão 59

3.6.1 Redes neurais artificiais - Perceptron multicamadas 61
3.6.2 *Random Forest* 64
3.6.3 Máquinas de vetores suporte 67
3.6.4 Processo Gaussiano 71
3.6.5 Métricas de regressão 73
3.6.6 Coeficiente de correlação de Pearson 74
3.7 Extração de características 75
3.8 Subconjuntos de características 77
3.8.1 Estimativa da pressão máxima nos cilindros 77
3.8.2 Estimativa da pressão média dentro dos cilindros 77
3.8.3 Análise espectral 77
3.9 Vetor de características 78
3.10 Conclusão 80

4 Programa DEFault 82
4.1 Programa DEFault - Gerador de Sinais de Falha de Motor Diesel 84
4.2 Algoritmos do modelo Termodinâmico 88
4.2.1 Pressao.cal 88
4.2.2 Delay 89
4.2.3 Torque 90
4.2.4 Dif.volume 90
4.2.5 Parametros.modais 91
4.3 Algoritmos do modelo dinâmico 91
4.3.1 Replica.voltas 91
4.3.2 Runge.kutta 92
4.3.3 Vibracao.torcional 93
4.3.4 Ordem.espectro 93
4.3.5 Tensoes.global 94
4.4 Conclusão 94

5 Resultados 95
5.1 Ajuste de hiperparâmetros dos regressores 96
5.2 Testes de classificação 98
5.2.1 Resultados de classificação usando 60 dB SNR AWGN 100

5.2.2 Resultados da classificação abordando diferentes níveis de ruído branco gaussiano 103
5.2.3 Análise Comparativa entre o Método de Levenberg-Marquardt e classificador *Random Forest* 104
5.3 Testes de regressão 107
5.3.1 Análise Comparativa entre o Método de Levenberg-Marquardt e regressor por processo Gaussiano 111

Considerações finais **113**
Trabalhos futuros 116
Produção científica 116

Referências **119**

Apêndices **129**

A Algoritmos do programa DEFault **130**

Introdução

Nos últimos cinquenta anos, o uso de motores Diesel tem se expandido para aplicações em áreas como indústria, agricultura, construção e transporte rodoviário (Mathur et al., 2022). Motores Diesel são amplamente utilizados em ambientes industriais devido à sua robustez e capacidade de atender a altos valores de torque. Por isso, é fundamental adotar uma programação de manutenção efetiva para evitar paradas não planejadas (Xin, 2011a). Ao escolher um motor Diesel para uma aplicação, três atributos precisam ser considerados: custo, desempenho e vida útil (Bui et al., 2022), além de um bom plano de manutenção, é essencial implementar práticas adequadas de operação e monitoramento contínuo para garantir uma vida útil prolongada.

A tecnologia de automação e instrumentação também avançou, resultando em dispositivos de monitoramento eficientes e inteligentes. Isso permitiu a aplicação de técnicas de manutenção preditiva, permitindo que as paradas de produção sejam planejadas. O planejamento adequado do estoque de peças de reposição é crucial para realizar a manutenção planejadamente (Chow, 2000).

Um sistema de prognóstico de equipamentos é fundamental para monitorar sua condição e determinar a vida útil restante, resultando em uma redução nos custos de produção na indústria do petróleo (Zhang and Hu, 2013).

Na análise de diagnóstico de motores Diesel, é estimada a presença de defeitos na máquina, os métodos intrusivos e não intrusivos são amplamente utilizados. Os métodos intrusivos envolvem a intervenção direta no motor, como a análise de óleo, inspeção visual e medição de compressão, sendo mais precisos, porém mais demorados e custosos, além de requererem a parada do motor (Bartoń, 2018).

Já os métodos não intrusivos utilizam sensores e técnicas de monitoramento remoto, como análise de vibração, termografia, monitoramento acústico e análise de gases de exaustão, permitindo o monitoramento contínuo durante a operação do motor. Embora menos detalhados, os métodos não intrusivos são menos invasivos e permitem a detecção precoce de falhas sem interromper o funcionamento do motor. A escolha do método depende das necessidades específicas do diagnóstico e das condições operacionais do motor Diesel (Akpinar and Yumusak, 2016).

Neste contexto, a análise de vibração é uma técnica amplamente utilizada, pois os sensores são fáceis de instalar e produzem resultados confiáveis na identificação de falhas (Xi et al., 2018).

Os motores Diesel são conhecidos por sua alta durabilidade operacional, potência e eficiência, mas a complexidade em identificar falhas com precisão é uma desvantagem significativa. Este trabalho visa aprimorar o método descrito por (Gutiérrez, 2016), por meio do uso de técnicas de aprendizado de máquina, utilizando classificadores e regressores baseados em Perceptron multicamada (MLP), *Random Forest* (RF), Máquinas de Vetores de Suporte (SVM) e Modelos de Regressão de Processo Gaussiano (GPRM).

Este estudo apresenta uma análise baseada em aprendizado de máquina para caracterizar a severidade e as classes de falhas em motores Diesel, incluindo: operação normal, redução de pressão no coletor de admissão, redução da razão de compressão nos cilindros e redução da quantidade de combustível injetado nos cilindros.

As técnicas adotadas irão prever a severidade das falhas com base em dados de entrada, como temperatura, pressão e vibração torcional. O objetivo é fornecer uma ferramenta precisa e confiável para a detecção precoce e diagnóstico de falhas, visando maximizar a disponibilidade e a eficiência dos motores Diesel.

Principais contribuições

É fundamental ter um sistema que realize o prognóstico do equipamento para permitir o monitoramento das condições e informar sobre o estado operativo da máquina, detectando a iminência de uma falha e avaliando se esta poderá causar uma parada de funcionamento.

Dessa forma, esta pesquisa propõe o uso de inteligência artificial nos métodos de regressão, baseados nos regressores MLP, RF, SVM e GPRM, e nos métodos de classificação, baseados nos classificadores MLP e RF, para estimar falhas em motores Diesel, utilizando sinais de vibração, pressão e temperatura como entrada. As contribuições deste trabalho podem ser resumidas da seguinte maneira:

- Uma base de dados de falhas, que emula falhas de um motor Diesel através de um gêmeo digital de seis cilindros, foi construída e disponibilizada publicamente para a

comunidade científica e disponível em (Pestana-Viana, 2021);

- Uma aplicação real de um algoritmo inteligente para fornecer o diagnóstico em motores Diesel, permitindo o monitoramento da condição operacional, bem como a identificação de falhas e sua severidade associada.

A utilização de procedimentos de simulação para a criação de bases de dados destinadas ao treinamento de algoritmos de inteligência artificial voltados à identificação de falhas em máquinas diesel é um aspecto crucial e inovador no campo da manutenção preditiva (Li et al., 2020).

A utilização de simulações proporciona a criação de conjuntos de dados balanceados, com exemplos tanto de condições normais quanto de falhas, essenciais para o treinamento de algoritmos capazes de identificar padrões sutis e complexos. Portanto, a integração de procedimentos de simulação com algoritmos de inteligência artificial representa uma abordagem promissora para melhorar a eficiência e a confiabilidade dos sistemas de manutenção condicional em máquinas diesel, contribuindo significativamente para a otimização dos processos industriais e a redução de custos operacionais.

As simulações baseadas em gêmeos digitais desempenham um papel crucial no contexto da simulação de motor Diesel (Li et al., 2020). Os gêmeos digitais são réplicas virtuais de sistemas físicos, como motores, que permitem simular seu comportamento em condições variadas (Lee et al., 2020).

No caso específico de motores Diesel, os gêmeos digitais podem ser usados para prever o desempenho do motor, identificar possíveis falhas e otimizar estratégias de manutenção (Feng et al., 2020). Ao integrar gêmeos digitais em sistemas de manutenção condicional, é possível monitorar continuamente o estado do motor, detectar anomalias e antecipar falhas iminentes, contribuindo para a redução de custos operacionais e o aumento da eficiência do sistema (Lee et al., 2020).

A combinação de arquiteturas modulares com simulações baseadas em gêmeos digitais representa uma abordagem poderosa para o desenvolvimento de sistemas de manutenção preditiva em uma variedade de aplicações industriais.

Organização do trabalho

Esta tese está estruturada da seguinte maneira:

- O Capítulo 1 apresenta a revisão bibliográfica;
- O Capítulo 2 apresenta os conceitos fundamentais do modelo termodinâmico e também descreve a base matemática do modelo dinâmico torcional da árvore de manivelas;
- No Capítulo 3 é realizada a validação do modelo e a simulação de falha;
- No Capítulo 4 é descrita a organização do programa DEFault;
- Os resultados e as análises são apresentados no Capítulo 5.

1- Revisão bibliográfica

O estado da arte nas técnicas preditivas de diagnóstico de severidade de falhas em motores Diesel têm avançado consideravelmente nos últimos anos, impulsionado pelo desenvolvimento de métodos cada vez mais sofisticados e pela integração de tecnologias emergentes. Dentre as quatro principais abordagens utilizadas, destacam-se:

1. **Modelagem termodinâmica avançada**: Pesquisadores têm desenvolvido modelos termodinâmicos complexos que consideram parâmetros como temperatura, pressão, transferência de calor e características do combustível. Esses modelos são calibrados e validados com medições reais, seja em escala reduzida ou protótipos, utilizando simulações computacionais detalhadas. A escolha do método depende dos recursos e objetivos do estudo, com modelos em escala reduzida sendo mais econômicos e manejáveis, enquanto protótipos em tamanho real oferecem validação mais precisa. Esses processos permitem prever com precisão o comportamento do motor, identificar possíveis falhas e avaliar seu impacto na eficiência e desempenho.

2. **Análise de vibração**: A análise de vibração tem se mostrado uma ferramenta poderosa para a detecção precoce de falhas em motores Diesel. Utilizando acelerômetros e outros sensores de vibração, é possível monitorar as oscilações do motor em tempo real e identificar padrões anômalos que possam indicar a presença de defeitos, como desgaste de componentes, desalinhamento ou folgas excessivas.

3. **Aprendizado de máquina**: O uso de técnicas de aprendizado de máquina, como redes neurais artificiais e algoritmos de classificação, tem se tornado cada vez mais comum para o diagnóstico de severidade de falhas em motores Diesel. Esses métodos são capazes de processar grandes volumes de dados e identificar padrões complexos, permitindo uma análise mais precisa e rápida das condições do motor e a previsão da gravidade das falhas.

4. **Simulação de falhas e testes em bancada**: O desenvolvimento de simuladores de falhas e a realização de testes em bancada têm sido fundamentais para o estudo do comportamento de motores Diesel sob diferentes condições de operação e a

avaliação da severidade de falhas. Essas técnicas permitem reproduzir cenários específicos de defeitos e analisar seu impacto no desempenho do motor, orientando o desenvolvimento de estratégias de manutenção preditiva mais eficazes.

Os estudos abordados demonstram uma variedade de técnicas para a detecção e simulação de falhas em motores Diesel, destacando a importância crescente da manutenção preditiva para garantir a operação confiável desses equipamentos em diferentes cenários. Desde a proposição de modelos termodinâmicos complexos até o uso de técnicas avançadas de análise de sinais, como transformada rápida de Fourier e redes neurais artificiais, os pesquisadores têm explorado uma gama diversificada de métodos para identificar e prevenir falhas nos motores.

A análise de vibração surge como uma ferramenta poderosa para detectar sinais subtis que indicam possíveis problemas nos componentes do motor. Paralelamente, o desenvolvimento de simuladores de falhas tem permitido aos pesquisadores reproduzir cenários específicos de defeitos. Isso facilita a análise do impacto dessas falhas no desempenho do motor e orienta a implementação de estratégias de manutenção mais eficazes. Esses avanços representam uma contribuição significativa para a engenharia de manutenção, proporcionando novas perspectivas para o diagnóstico precoce e a mitigação de falhas em motores Diesel. Tais melhorias prometem benefícios substanciais em termos de segurança operacional, eficiência e custos de manutenção.

A Tabela 1 apresenta um resumo das principais contribuições de estudos relacionados à detecção de falhas em motores Diesel, com foco em motores marítimos e de alta potência. Os estudos abordam diferentes aspectos, desde o desenvolvimento de metodologias de diagnóstico baseadas em modelos termodinâmicos até a utilização de sinais de vibração para identificar falhas específicas nos cilindros. Destaca-se a aplicação de técnicas avançadas de análise de sinais, como a transformada rápida de Fourier e a análise de séries temporais, para prever falhas de combustão com precisão. Além disso, alguns estudos exploram a utilização de algoritmos de aprendizado de máquina para a classificação automática de falhas. Essas pesquisas fornecem uma base sólida para o desenvolvimento de sistemas de manutenção preditiva mais eficazes, visando aumentar a confiabilidade e a segurança dos motores Diesel em diferentes aplicações industriais e navais. A tabela oferece uma visão geral desses estudos, facilitando a identificação das principais contribuições e áreas de pesquisa em andamento nesta área.

Tabela 1 – Principais contribuições na detecção de falhas em motores Diesel.

Referência	**Introdução**	**Objetivo**	**Metodologia**	**Conclusão**
(Charles et al., 2009)	Análise de vibração torcional em motores de alta potência	Prever falhas de combustão nos cilindros	Transformada Rápida de Fourier, Análise de Séries Temporais	Identificação eficaz de falhas
(Hountalas, 2000)	Estudo de defeitos em motor Diesel marítimo	Diagnosticar e determinar severidade de falhas	Modelo termodinâmico de duas zonas, Métodos dos Mínimos Quadrados	Método confiável para diagnóstico
(Hountalas and Kouremenos, 1999)	Investigação de agentes causadores de redução de pressão	Monitorar pressão nos cilindros	Testes em motor de compressão variável	Monitoramento crucial para diagnóstico
(Klinchaeam et al., 2009)	Detecção de falhas em motor de ignição por faísca	Analisar sinais de vibração	Análise temporal, Comparação com referência	Metodologia não intrusiva eficaz
(Pestana-Viana et al., 2018)	Diagnóstico de falhas em motores Diesel	Utilização de resposta de vibração e variações de pressão	Simulações, Algoritmos de Aprendizado de Máquina	Precisão de 99,3% com Random Forest
(Rubio et al., 2018)	Simulador de falhas em motores Diesel marítimos	Reprodução de falhas termodinâmicas	Modelo zero-dimensional ajustado	Testes controlados sem interrupção
(Xi et al., 2018)	Estudo de falhas em motor Diesel de quatro cilindros	Medição de sinais de vibração	Análise de componentes independentes, Rede Neural Artificial	Precisão de 93,33% na classificação de falhas

No artigo de (Charles et al., 2009), o sinal instantâneo de onda de velocidade angular foi usado para analisar a vibração torcional de motores de combustão interna de alta potência, a fim de prever falhas de combustão nos cilindros, uma vez que para esses tipos de motores a velocidade da árvore de manivelas é instável devido à ordem de ignição dos cilindros. O sensor usado nesses testes era um tipo de captador magnético instalado no volante do motor com 16 cilindros. Para prever as falhas, foram utilizadas as técnicas de Transformada Rápida de Fourier e Análise de Séries Temporais, ambas alcançando bons resultados na identificação de falhas de combustão nos cilindros do motor.

No trabalho de (Hountalas, 2000), foram estudados defeitos que ocorrem durante o uso de um motor Diesel marítimo de dois tempos com seis cilindros dispostos em linha. O objetivo era desenvolver uma metodologia capaz de diagnosticar a fonte de uma falha do motor e determinar a severidade dela. Foi proposto um modelo termodinâmico de duas zonas, este modelo incorpora aspectos como admissão e exaustão, transferência de calor através das paredes dos cilindros, e o comportamento do jato de combustível. Após calibração do modelo, diversas simulações foram realizadas considerando defeitos nos componentes do sistema: falha do bico, atraso no ângulo de injeção e defeito de compressão, com a restrição de que os defeitos ocorressem simultaneamente em todos os cilindros. O método dos mínimos quadrados foi utilizado para o diagnóstico de falhas e resultado indicou que o modelo de previsão de falhas é confiável para operações em motores Diesel marítimos, visando evitar falhas súbitas que possam interromper o funcionamento do motor.

No artigo escrito por (Hountalas and Kouremenos, 1999), foram realizados experimentos para investigar agentes causadores de redução de pressão nos cilindros de um motor Diesel. Três casos foram considerados: mudança na razão decompressão, variação de pressão no coletor de admissão e mudança no fluxo de gás. Os testes foram conduzidos em um motor de compressão variável com um único cilindro, monitorando a pressão ao longo de diferentes ângulos da árvore de manivelas. Os resultados mostraram que alterações na pressão no coletor de admissão influenciam significativamente o processo de compressão, enquanto variações no fluxo de gás modificam o estágio final do processo e mudanças na razão decompressão afetam principalmente o processo de expansão. Concluiu-se que a monitoração da curva de pressão dentro dos cilindros é crucial para um diagnóstico eficaz de falhas em motores Diesel.

O trabalho de (Klinchaeam et al., 2009) utilizou sinais de vibração para detectar falhas em um motor de ignição por faísca de quatro tempos. Os testes foram conduzidos sob diferentes cenários: operação normal, separação dos eletrodos da vela, abertura e fechamento das válvulas de admissão. Durante a operação, eventos como abertura e fechamento das válvulas, compressão, combustão e exaustão provocaram mudanças na amplitude das vibrações do motor. Os sinais de vibração foram capturados por um acelerômetro posicionado na cabeçote do cilindro, e um tacômetro foi utilizado para medir as rotações do motor. A análise temporal das vibrações, considerando o ângulo da árvore de manivelas e comparando com sinais de referência de operação sem falhas, demonstrou a eficácia de uma metodologia não intrusiva para o diagnóstico de motores de médio e grande porte.

O trabalho de (Pestana-Viana et al., 2018) desenvolveu um sistema de manutenção preditiva baseado no diagnóstico de falhas de motores Diesel, usando a resposta de vibração da árvore de manivelas e variações nas curvas de pressão nos cilindros. Simulações baseadas em um modelo termodinâmico de dimensão zero foram realizadas, abrangendo 701 cenários de falhas divididos em quatro condições operacionais: falha na razão de compressão, falha na quantidade de combustível injetado nos cilindros, falha na redução de pressão no coletor de admissão e operação normal. Dois algoritmos de aprendizado de máquina foram aplicados: Redes Neurais Artificiais, através do perceptron multicamadas, e classificadores baseados em Random Forest. O algoritmo Random Forest alcançou uma precisão de 99,3%. Os resultados indicaram que a relação sinal-ruído deve ser superior a 15 dB para garantir um desempenho adequado dos classificadores.

Os autores de (Rubio et al., 2018) desenvolveram um simulador de falhas para motores Diesel marítimos de alta velocidade de quatro tempos, baseado em um modelo termodinâmico zero-dimensional ajustado e validado com dados experimentais de motores reais. O simulador foi capaz de reproduzir com confiabilidade várias falhas termodinâmicas típicas de motores Diesel e o comportamento normal (sem falhas) da máquina. Isso permitiu testar o equipamento controladamente sem interromper a produção industrial.

O trabalho de (Xi et al., 2018) investigou falhas em um motor Diesel comercial de quatro cilindros. Para medir o sinal de vibração, quatro acelerômetros piezoelétricos foram montados em cada cabeçote. Cinco tipos de falhas foram estudados: falha no pino do pistão; falha no anel do pistão; falha na válvula de admissão; falha na válvula de exaustão; e falha no eixo de comando de válvulas. A velocidade de rotação do motor

era de 1500 RPM e cada cenário de falha foi testado dez vezes. Um método baseado na análise de componentes independentes de Kernel e na transformada de Stockwell (Stockwell et al., 1996) foi utilizado para fins de pré-processamento. Doze características extraídas deste procedimento inicial foram usadas como entrada para uma rede neural artificial que realizou uma classificação automática de falha. O classificador avançado obteve uma precisão de 93,33% nos cenários de falha.

Recentemente, avanços significativos foram feitos na detecção de falhas em motores Diesel, incluindo parâmetros termodinâmicos (Lamaris and Hountalas, 2010), análise de óleo (Yan et al., 2005), análise de vibração (Yang et al., 2001), entre outros. O desenvolvimento de tecnologias de automação e instrumentação também contribuiu para o desenvolvimento de dispositivos de monitoramento eficazes, promovendo técnicas avançadas de manutenção preditiva. Métodos inteligentes de diagnóstico de falhas têm sido amplamente adotados para estimar as condições operacionais das máquinas (Li et al., 2020; Li et al., 2019; Wang et al., 2020).

O texto descreveu pesquisas realizadas para diagnosticar falhas em motores Diesel marítimos e de combustão interna de alta potência. No geral, o estado da arte nas técnicas preditivas de diagnóstico de severidade de falhas em motores Diesel reflete uma abordagem multidisciplinar e integrada, que combina modelos teóricos avançados, técnicas de sensoriamento e monitoramento em tempo real, e métodos de análise de dados e aprendizado de máquina. Essa abordagem tem potencial para proporcionar benefícios significativos em termos de redução de custos de manutenção, aumento da disponibilidade operacional e prolongamento da vida útil dos motores. Os próximos capítulos têm como objetivo apresentar, de maneira teórica e como método de diagnóstico, as quatro principais técnicas de manutenção preditiva voltadas para o diagnóstico de motores Diesel.

2- Desenvolvimento do modelo termodinâmico e dinâmico do motor Diesel

Esta capítulo apresenta as principais características do modelo de motor Diesel que foi utilizado para criar o banco de dados nas condições normal e com falhas. Estes foram posteriormente empregados para treinar os modelos de regressão que foram desenvolvidos para diagnóstico ao avaliar a falha e sua correspondente severidade. Para simular o comportamento e operação de um motor Diesel em condições normais e com falhas, os seguintes modelos foram desenvolvidos:

1. Um modelo termodinâmico zero-dimensional (0-D);
2. Um modelo de massa concentrada para a vibração torcional da árvore de manivelas;
3. Um modelo de simulação de falhas, através da variação dos parâmetros operacionais.

Após o desenvolvimento, os modelos termodinâmico zero-dimensional e de massa concentrada para a vibração torcional da árvore de manivelas foram validados com base nos dados e informações do fabricante. A validação desses modelos, aplicados a motores Diesel de quatro tempos com sistema *common-rail*[1], é crucial para garantir a precisão e confiabilidade das previsões e diagnósticos gerados.

Através dessa validação, verifica-se a capacidade dos modelos em reproduzir com precisão o comportamento real dos motores sob diferentes condições de operação, incluindo cargas variáveis, velocidades de rotação e condições de falha. Isso proporciona maior confiança aos engenheiros nas previsões e diagnósticos, essenciais para decisões relacionadas à manutenção e operação eficientes dos motores Diesel.

A validação do modelo de simulação de falhas em motores Diesel é crucial para assegurar sua precisão e confiabilidade. No entanto, frequentemente enfrenta-se a dificuldade de validar esse tipo de modelo devido à impraticabilidade de submeter motores Diesel reais a condições controladas de falha em um ambiente de laboratório.

[1]O sistema *common-rail* é uma tecnologia avançada de injeção de combustível para motores diesel, caracterizada pela centralização do combustível em alta pressão em um *common-rail* ou tubo comum.

Diferentemente de outros tipos de equipamentos, como componentes eletrônicos ou máquinas pequenas, a quebra de motores Diesel é um processo complexo e caro, que pode envolver riscos de segurança e impactos financeiros significativos. Portanto, embora a validação do modelo de simulação de falhas seja desejável, muitas vezes não é possível realizá-la devido a essas limitações práticas e de segurança.

Neste capítulo, os modelos termodinâmico zero-dimensional e dinâmico de parâmetros concentrados são baseados nos estudos do motor Diesel MWM Acteon 6.12 TCE, conforme descrito nos trabalhos de (Gutiérrez, 2016). Nesta tese, o objetivo destes modelos, é desenvolver um gêmeo digital, criando uma base de dados de cenários de falhas, com a função de treinar os algoritmos de inteligência artificial, melhorando o tempo de diagnóstico das possíveis falhas nestes motores.

Um modelo dinâmico torcional do sistema pistão-biela-manivela é crucial devido à sua habilidade em simular com precisão as vibrações e os esforços críticos nessa parte fundamental do motor. Este modelo foi empregado especificamente para gerar vibrações torcionais, as quais são influenciadas por diversas condições operacionais. O objetivo principal foi estudar e compreender o comportamento dinâmico do conjunto pistão-biela-manivela sob diferentes cargas e regimes de operação do motor.

Essas vibrações desempenham um papel essencial na análise de desempenho, na detecção de potenciais falhas e na otimização do funcionamento do motor Diesel. Assim, contribui-se significativamente para o desenvolvimento de estratégias avançadas de manutenção preditiva e para a extensão da vida útil do equipamento. O estudo foi realizado em um motor Diesel de quatro tempos equipado com sistema de injeção *common-rail*, conforme detalhado nas especificações apresentadas na Tabela 2 em (Gutiérrez, 2016).

Foi adotado no presente trabalho um motor Diesel de quatro tempos e sistema *common-rail* por várias razões. Primeiramente, os motores de quatro tempos são conhecidos por sua eficiência e durabilidade, o que os torna ideais para uma variedade de aplicações industriais e comerciais. Além disso, esses motores são capazes de fornecer uma potência consistente e uma operação suave.

O sistema *common-rail*, por sua vez, oferece uma injeção de combustível mais precisa e controlada, resultando em uma combustão mais eficiente e uma melhor economia de combustível. Isso também ajuda a reduzir as emissões de poluentes, atendendo aos padrões ambientais mais rigorosos. Além disso, o sistema *common-rail* permite uma

maior flexibilidade no controle da injeção de combustível, o que é essencial para otimizar o desempenho do motor em diferentes condições de operação.

Em resumo, a escolha de um motor Diesel a quatro tempos e com sistema *common-rail* oferece uma combinação de eficiência, durabilidade, desempenho e controle que atende às necessidades de muitas aplicações industriais e comerciais modernas (Hountalas and Kouremenos, 1999).

2.1- Especificações Técnicas do Motor Diesel

O motor escolhido como estudo de caso é o motor Diesel de quatro tempo MWM Acteon 6.12 TCE (Gutiérrez, 2016). A Figura 1 mostra o motor do estudo desta tese, com as especificações técnicas exibidas na Tabela 2.



Figura 1 – Motor Diesel MWM Acteon 6.12 TCE.

Tabela 2 – Especificações técnicas do motor Diesel MWM Acteon 6.12 TCE.

Cilindros	6 em linha
Diâmetro do cilindro	105 mm
Curso do pistão	137 mm
Comprimento da biela	207 mm
Razão de compressão	16,8:1
Ângulo de fechamento da válvula de admissão	203º
Ângulo de abertura da válvula de exaustão	507º
Torque e potência máximos	900 N.m/191 kW
Rotação (no torque máximo)	1600 RPM
Pressão da linha	350 a 1400 bar
Temperatura da água de resfriamento	80 a 100ºC

2.2- Modelo termodinâmico

O modelo desenvolvido baseia-se em um modelo termodinâmico zero dimensional (0-D), com hipóteses simplificadoras, conforme (Heywood, 1988; Gutiérrez, 2016). Foram produzidos perfis de pressão dentro de cilindros, cuja a análise é crucial para o diagnóstico eficaz de motores a diesel. Esses perfis fornecem informações detalhadas sobre o desempenho do motor durante o ciclo de combustão, permitindo a detecção precoce de falhas, a avaliação da eficiência do processo de combustão e a previsão de falhas iminentes.

Além disso, os perfis de pressão são essenciais para validar modelos de simulação e garantir a correspondência entre os resultados teóricos e as condições reais de operação do motor. Em suma, a análise dos perfis de pressão é uma ferramenta fundamental para manter o desempenho, eficiência e confiabilidade dos motores a diesel.

O modelo 0-D em análise de motores Diesel é uma representação simplificada do comportamento termodinâmico do motor. Ele é assim chamado porque não leva em conta as variações de temperatura, pressão e outras propriedades dentro do motor ao longo do espaço tridimensional do cilindro. Em vez disso, ele considera o motor como um único ponto, onde as propriedades médias dos gases dentro do cilindro são calculadas em função do tempo.

As Equações (1) e (2) resumem o comportamento termodinâmico do motor Diesel. No entanto, vale ressaltar que, dependendo da aplicação, o desempenho do motor depende de vários parâmetros operacionais para caracterizar a potência e o torque. Mais

detalhes do modelo 0-D apresentado podem ser encontrados em (Gutiérrez et al., 2018).

A mistura de gases no cilindro passa por vários processos, como admissão, compressão, combustão, expansão e exaustão. Como o foco de interesse é apenas no desempenho, a pressão nos processos de admissão e exaustão foi considerada constante, conforme o ciclo padrão diesel para ar (Gutiérrez, 2016). Enquanto isso, os processos de compressão, combustão e expansão foram representados pela equação universal para gases ideais (Merker et al., 2011):

$$\frac{dP}{d\theta} = \left(\frac{mRd\Psi}{d\theta} - \frac{PdV}{d\theta} \right) \frac{1}{V}, \tag{1}$$

onde $\frac{dP}{d\theta}$ é a taxa de variação de pressão P (Pa) dentro do cilindro em função do ângulo de rotação da manivela θ (graus); m é a massa da mistura de gás dentro do cilindro (kg); R é a constante da mistura de gás dentro do cilindro (J/kg-K); Ψ é a temperatura instantânea dos gases dentro do cilindro (K) e V é o volume instantâneo dentro do cilindro (m^3).

O ciclo termodinâmico do motor Diesel real leva em conta a vazão de massa através das válvulas durante os processos de admissão e escapamento. No entanto, em relação ao desempenho, tal vazão pode ser desconsiderada, uma vez que a massa do volume de controle é considerada constante, como em (Merker et al., 2011).

O modelo zero-dimensional (0-D), baseado nos dados de projeto, é adequado para descrever o funcionamento do motor Diesel (Heywood, 1988). Este modelo é obtido aplicando a primeira lei da termodinâmica em um volume de controle com massa constante, conforme mostrado na Figura 2.

Este comportamento leva à expressão para calcular a temperatura dos gases dentro do cilindro (Heywood, 1988; Gutiérrez et al., 2018), mostrada na Equação (2). A equação resultante para a taxa de variação da temperatura Ψ (K) dentro do cilindro é:

$$\frac{d\Psi}{d\theta} = \left(Q_{tot} \frac{dx}{d\theta} - \frac{\delta Q_w}{d\theta} - P\frac{dV}{d\theta} \right) \frac{1}{mc_v}, \tag{2}$$

onde Q_{tot} é a variação total de calor fornecida ao sistema, $x(\theta)$ representa a fração de massa queimada modelada pela função de Wiebe dupla (Heywood, 1988), $\frac{\delta Q_w(\theta)}{d\theta}$ é a taxa de transferência de calor através das paredes do cilindro, e mc_v é o calor específico a volume constante da mistura (J/kg-K).

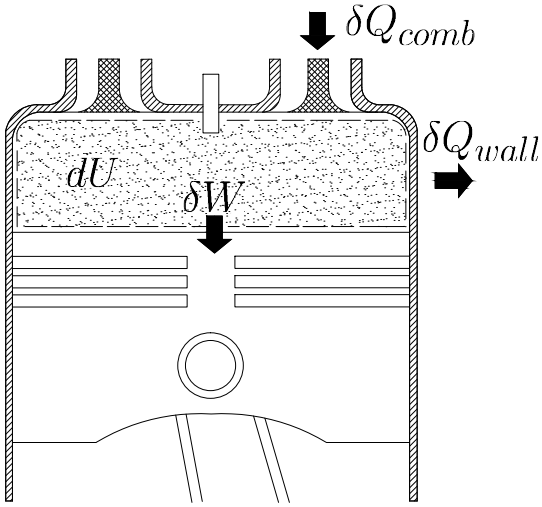


Figura 2 – Volume de controle da análise termodinâmica.

2.2.1 Validação do modelo

Para validar o modelo zero-dimensional de um motor Diesel, é necessário conduzir uma análise abrangente. Primeiramente, foram coletados os dados de pressão experimentais fornecidos pelo fabricante do motor ou obtidos através de instrumentação durante o funcionamento real do motor.

Em seguida, o modelo zero-dimensional foi utilizado para simular o comportamento do motor Diesel sob condições operacionais semelhantes às do experimento real, considerando parâmetros como carga do motor, velocidade de rotação e temperatura ambiente.

Após a simulação, foi realizada uma comparação entre os perfis das curvas de pressão geradas pelo modelo e os dados experimentais. Esta comparação não se limitou apenas à análise visual, mas também incluiu uma análise quantitativa das curvas de pressão, que abrangeu métricas específicas como os erros entre os picos e os valores mínimos de pressão simulados e do fabricante.

Além disso, foi determinado a rotação que melhor se ajustou aos dados do fabricante e com menor erro global, somando-se os erros de validação das pressões médias indicadas P_{PMI} e máximas P_{max}.

Em vista disto, vários perfis de curva foram utilizados para validar o modelo termodinâmico e emular diferentes pressões no cilindro. Esses perfis de pressão foram obtidos através de experimentos com o motor adotado. A Tabela 3 apresenta pressões experimentais e simuladas para verificar as diferenças entre elas.

Tabela 3 – Pressões experimentais e simuladas.

Rotação (RPM)	P_{PMI} (bar)			P_{max} (bar)		
	Exper.	Simul.	Erro (%)	Exper.	Simul.	Erro (%)
1000	11,92	13,13	10	139,4	141,9	2
1200	13,84	14,79	7	136,7	144,5	6
1300	13,98	14,89	7	131,4	143,6	9
1600	14,44	15,34	6	141,1	153,3	9
1900	16,01	15,64	2	149,1	163,7	10
2100	15,96	14,90	7	156,6	152,1	3
2300	15,04	13,97	7	154,0	147,3	4
2500	12,78	13,46	5	145,7	145,0	0

A Figura 3 apresenta alguns dos perfis de pressão simulados e experimentais. De acordo com os resultados (Tabela 3), o modelo termodinâmico desenvolvido representa satisfatoriamente o desempenho do motor Diesel, indicando erros inferiores a 10% para ambas as pressões (ou seja, P_{PMI} e P_{max}).

Apesar do modelo obtido se referir a um único cilindro, de acordo com (Mendes et al., 2008), é possível obter o perfil de pressão para os seis cilindros se forem respeitados o intervalo e a ordem de ignição. O perfil de pressão para 2500 RPM pode ser visto na Figura 4.

2.3- Modelo dinâmico torcional da árvore de manivelas

Para esta pesquisa, um sistema de árvore de manivelas, mostrado na Figura 5, com 11 graus de liberdade foi considerado para compor o modelo de torção, conforme exibido na Figura 6. O modelo de parâmetros concentrados foi desenvolvido por (Mendes et al., 2008). A Equação (3) provê a equação diferencial do movimento relacionado aos parâmetros torcionais da máquina, com o torque externo desenvolvido na árvore de manivelas,

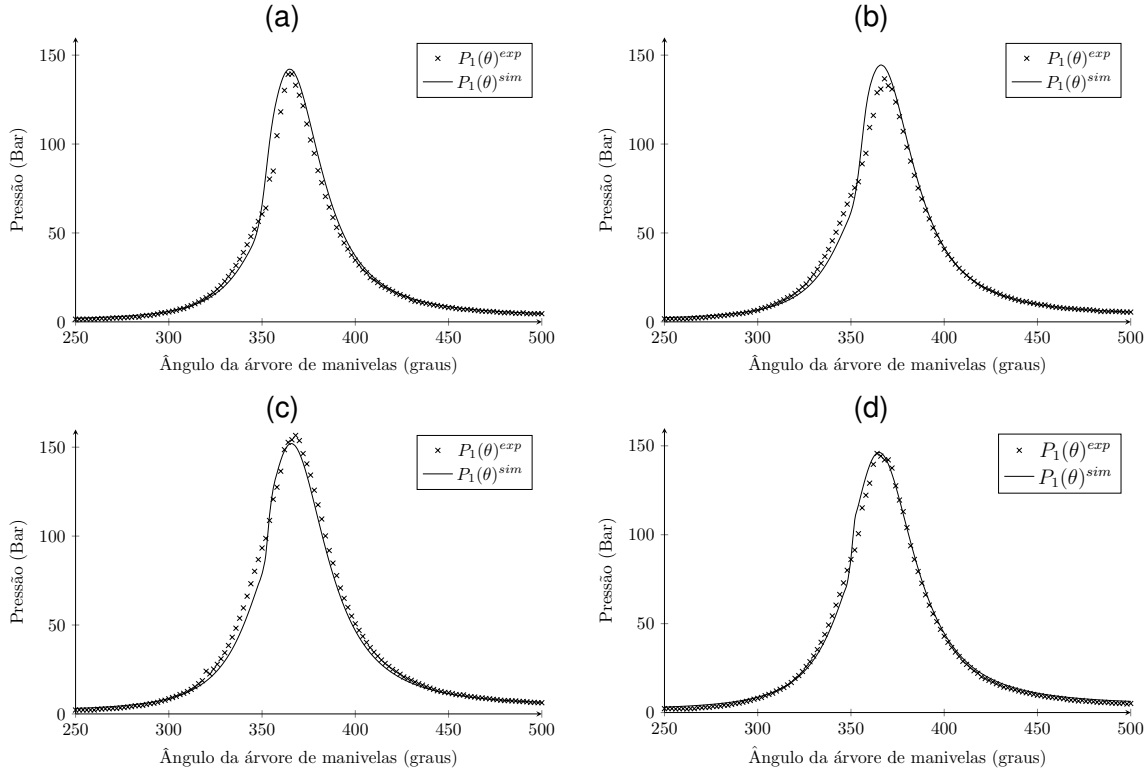


Figura 3 – Validação do modelo termodinâmico desenvolvido para diferentes rotações: (a) 1000, (b) 1200, (c) 2100 e (d) 2500 RPM. Nos gráficos $P_1(\theta)^{exp}$ e $P_1(\theta)^{sim}$ referem-se aos dados experimentais e simulados, respectivamente.

$$[\mathbf{J}]\left\{\ddot{\theta}(t)\right\} + [\mathbf{C}]\left\{\dot{\theta}(t)\right\} + [\mathbf{K}]\left\{\theta(t)\right\} = \left\{M(t)\right\}, \tag{3}$$

onde [**J**], [**C**] e [**K**] são a matriz de inércia, a matriz de amortecimento e a matriz de rigidez torcional, respectivamente. O vetor $\{\theta(t)\}$ é a posição angular, $\left\{\dot{\theta}(t)\right\}$ é a velocidade angular e $\left\{\ddot{\theta}(t)\right\}$ é a aceleração angular, e $\{M(t)\}$ é o vetor de torques externos relacionados a cargas dinâmicas originadas pelas cargas de combustão e inércia.

Os valores de amortecimento absoluto foram obtidos a partir de experimentos e os amortecimentos relativos foram calculados através da Equação (4):

$$c_{ri} = \frac{k_{ti}}{N} d, \tag{4}$$

onde d representa um fator de perda que depende do material e do processo de fabricação. Para os amortecedores torcionais, adotou-se 0,25, enquanto para os outros componentes da árvore de manivelas, este fator foi definido como 0,035 (Mendes et al., 2008).

As cargas de inércia são devidas ao movimento dos componentes e são subdividi-

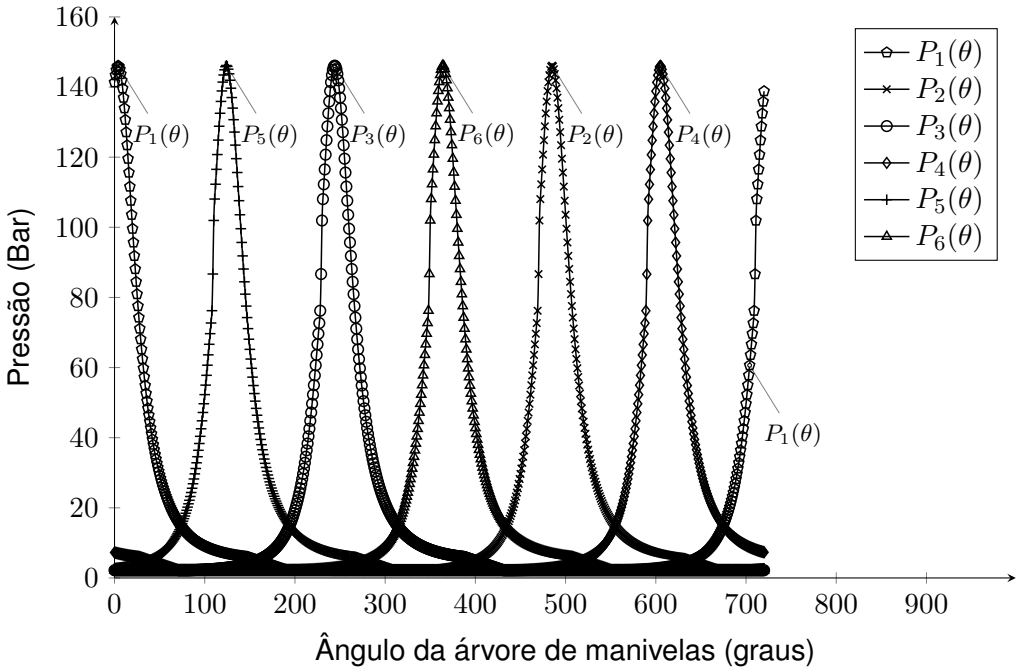


Figura 4 – Curvas de pressão de acordo com a ordem de ignição para 2500 RPM.

das em cargas puramente alternativas e rotativas. No entanto, como o interesse é analisar a torção da árvore de manivelas, somente as forças alternativas serão consideradas. Além disso, a cinemática do sistema deve ser desenvolvida para descrever a força instantânea alternada como uma função da posição angular da árvore de manivelas, observada na Equação (5),

$$F_a(\theta) = m_a r \Omega^2 \left(\cos\theta + l\cos 2\theta - \frac{l^3}{4}\cos 4\theta + \frac{9l^5}{128}\cos 6\theta \right). \tag{5}$$

Na Equação (5), Ω é a velocidade angular da árvore de manivelas, dada em rad/s, e θ é sua posição angular, dada em rad. No caso do mancal, o termo m_a é dado pela Equação (6),

$$m_a = \frac{m_b L_1}{L}, \tag{6}$$

onde m_b é a massa do mancal e L_1 e L são comprimentos do componente, como mostrado na Figura 7.

A queima da mistura ar-combustível dentro do cilindro gera as cargas de combustão. Devido à combustão, a mistura de gases se expande, resultando em uma variação de pressão sobre a superfície do pistão. Isso resulta em uma força transmitida pelo mancal

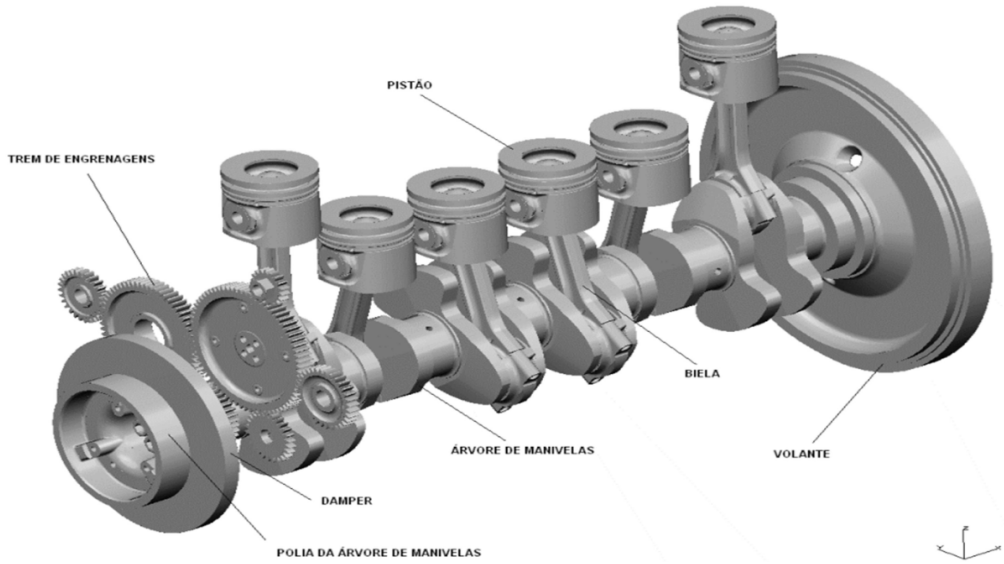


Figura 5 – Árvore de Manivelas do Motor MWM Acteon 6.12TCE (Gutiérrez, 2016).

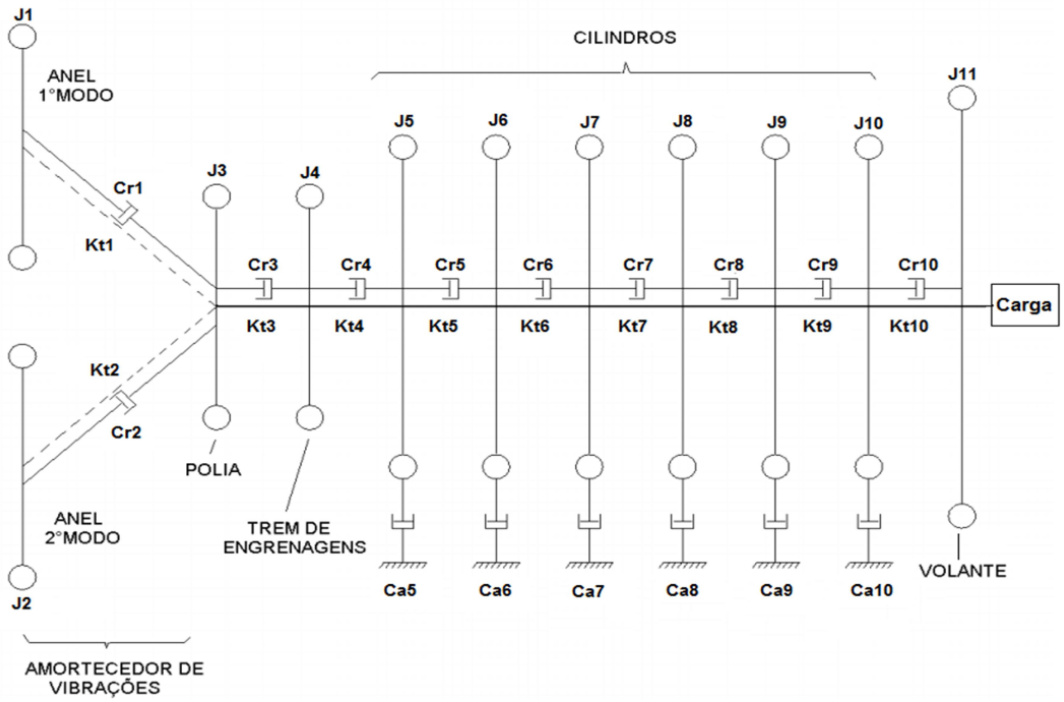


Figura 6 – Modelo equivalente de Massa concentrada da árvore de Manivelas do Motor MWM Acteon 6.12TCE (Gutiérrez, 2016).

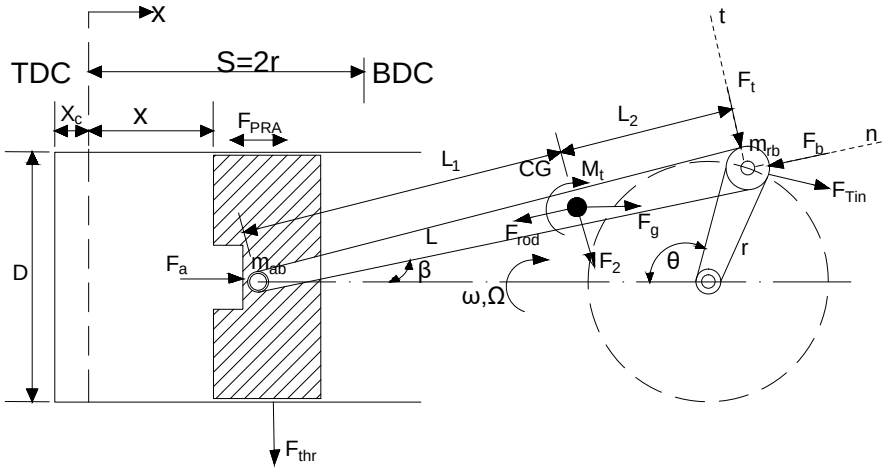


Figura 7 – Esquema do sistema pistão-biela-manivela.

para a árvore de manivelas, como apresentado na Equação (7):

$$F_g(\theta) = P(\theta)\frac{\pi D^2}{4}. \tag{7}$$

Estas forças são calculadas com base no perfil de pressão dentro do cilindro como uma função da posição angular da árvore de manivelas, apresentado na Seção 2.2. Na dinâmica torcional, somente as cargas tangenciais são efetivas para os torques externos, como apresentado na Equação (8),

$$F_t(\theta) = F_{at} + F_{gt}, \tag{8}$$

onde F_{at} e F_{gt} são as componentes tangenciais das cargas alternativas.

A Equação (9) fornece o torque externo para um sistema de manivela deslizante único, considerando r a distância do centro da árvore de manivelas até o ponto de aplicação da força efetiva,

$$M_i(\theta) = F_{ti}(\theta).r, \tag{9}$$

onde $F_{ti}(\theta)$ é a força efetiva para o cálculo do torque relacionado a um sistema de manivela deslizante único.

Como já mencionado na Seção 2.2, apesar do modelo obtido se referir a um único cilindro, é possível obter o torque para os seis cilindros se o intervalo e a ordem de ignição forem respeitados (Solmaz and Karabulut, 2015). A Figura 8 mostra o perfil de torque

para 2500RPM.

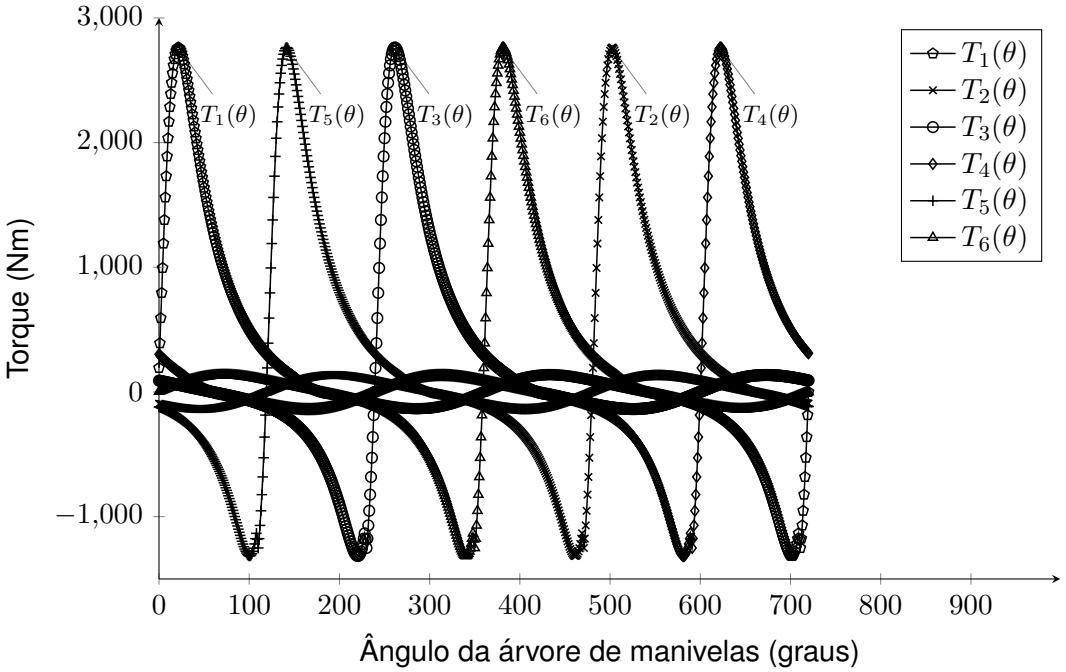


Figura 8 – Curvas de torque de acordo com a ordem de ignição para 2500 RPM.

2.3.1 Verificação do modelo

A Tabela 4 apresenta uma comparação entre os torques informados pelo fabricante e os valores estimados pelo modelo dinâmico. Os momentos de torção calculados através do modelo desenvolvido são apresentados de acordo com diferentes rotações. Observe que a diferença máxima ocorreu a 2500 RPM. A Figura 9 apresenta a resposta à vibração torcional como a variação do torque para 2500 RPM.

Tabela 4 – Comparação de torque.

Rotação	Torque (N.m)		Erro
(RPM)	fabricante	calculado	(%)
1000	800	752	6
1200	885	848	4
2100	850	851	0,1
2500	730	783	7

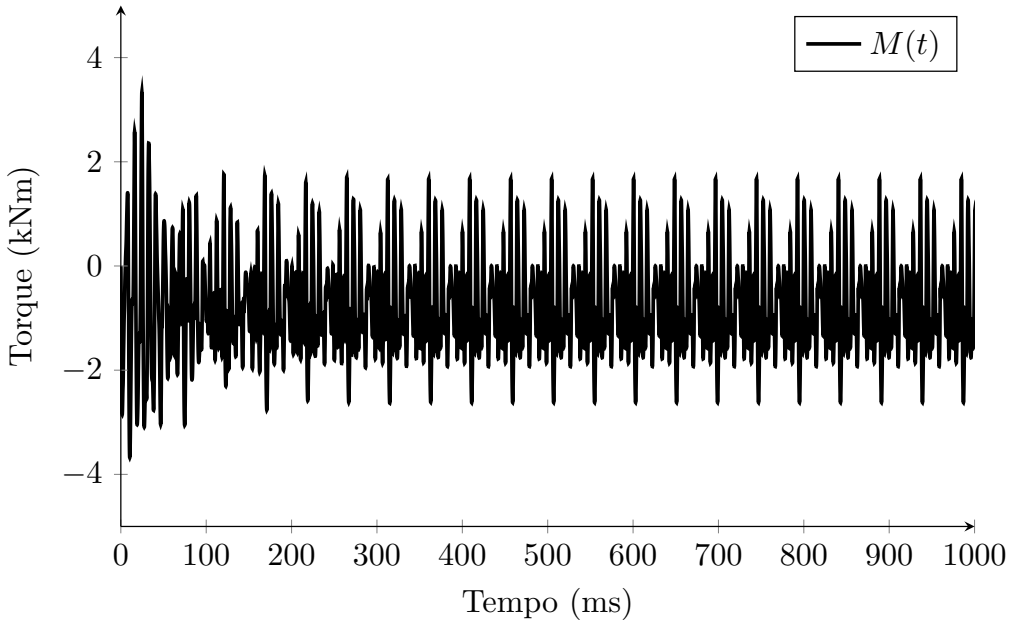


Figura 9 – Vibração torcional para o sinal normal da árvore de manivelas a 2500 RPM.

2.4- Modelo de simulação de falha

Durante a operação de um motor Diesel, qualquer falha de um componente ou sub-sistema afetará o desempenho do motor com diferentes intensidades. Dessa forma, algumas falhas afetam todos os cilindros simultaneamente e outras afetam apenas um cilindro. Essas não conformidades podem causar variações de pressão na admissão e na injeção, variações de temperatura nas paredes dos cilindros, entre outras.

As falhas em um motor Diesel podem ser diversas e abrangem uma ampla gama de componentes. Algumas das falhas mais comuns que podem ocorrer, conforme abordado por (Xin, 2011b), são:

- Problemas de Injeção de Combustível: Isso pode incluir vazamentos nos injetores de combustível, obstruções nos bicos injetores, desgaste dos componentes do sistema de injeção, pressão de combustível inadequada, entre outros;
- Falhas no Sistema de Alimentação de Combustível: Isso abrange problemas como a contaminação do combustível, entupimento ou falha dos filtros de combustível, problemas na bomba de combustível, e falhas no sistema de alimentação de ar;
- Problemas Elétricos e Eletrônicos: Incluem falhas nos sensores de pressão, tem-

peratura e velocidade, mau contato elétrico nos conectores, falhas na unidade de controle do motor (ECU), problemas na fiação elétrica, entre outros;

- Desgaste Mecânico: Isso pode envolver desgaste nos pistões, anéis de pistão, cilindros, bronzinas, bielas, árvore de manivelas, válvulas, e outras peças móveis do motor;
- Sobreaquecimento: Problemas relacionados ao sistema de arrefecimento, como vazamentos no radiador, mangueiras do sistema de arrefecimento danificadas, bomba de água com defeito, termostato com falha, entre outros, podem levar ao sobreaquecimento do motor;
- Problemas de Lubrificação: Isso inclui a falta de lubrificação adequada devido a vazamentos no sistema de lubrificação, falha da bomba de óleo, filtro de óleo obstruído, entre outros;
- Falhas Eletrônicas na Unidade de Controle: Problemas relacionados à programação inadequada da ECU, falhas de software, e mau funcionamento dos sistemas de diagnóstico a bordo (OBD);
- Desgaste ou Danos do Turbocompressor: Isso pode incluir problemas como avarias nas pás do compressor ou da turbina, vazamentos nos vedantes, falhas no atuador de geometria variável, entre outros.

Essas são apenas algumas das falhas comuns que podem ocorrer em um motor Diesel *common-rail*. É importante realizar manutenções preventivas regulares e diagnosticar prontamente qualquer problema para evitar danos mais graves e custosos.

Nesta tese, na Seção 3.2, são apresentados os cenários de falhas analisados. São avaliados quatro tipos diferentes de cenários, a saber:

- operação normal;
- redução de pressão no coletor de admissão;
- redução da razão de compressão nos cilindros;
- redução da quantidade de combustível injetado nos cilindros.

2.4.1 Severidade das falhas

As falhas em máquinas diesel podem ser classificadas em diferentes graus de severidade, dependendo do impacto que têm no desempenho e na segurança da máquina. Esses graus podem ser categorizados da seguinte forma:

- **Falhas de baixa severidade**: Falhas de baixa severidade são aquelas que têm um impacto mínimo no desempenho da máquina e não representam uma ameaça imediata à sua operação. Exemplos incluem pequenas vazamentos de óleo, ruídos anormais e oscilações de temperatura dentro de limites aceitáveis.
- **Falhas de média severidade**: Falhas de média severidade são aquelas que afetam moderadamente o desempenho da máquina e podem levar a uma redução na eficiência ou na vida útil dos componentes. Isso pode incluir problemas como desgaste excessivo de peças, falhas intermitentes e perda gradual de potência.
- **Falhas de alta severidade**: Falhas de alta severidade são aquelas que têm um impacto significativo no funcionamento da máquina e representam uma séria ameaça à sua operação segura e eficiente. Isso pode incluir problemas como falhas de componentes críticos, superaquecimento severo e vazamentos graves de fluidos.

2.4.2 Características das falhas

Diversos fatores influenciam o desempenho de um motor Diesel e podem alterar sua condição normal de operação. Pesquisadores como (Hountalas, 2000), (Hountalas and Kouremenos, 1999), (Abdulaziz and Khan, 2014), (Zaharia et al., 2016) e (Smola and Schökopf, 2000) destacaram alguns desses fatores, incluindo falhas no turbocompressor, sistema de admissão e exaustão, sistema de arrefecimento e sistema de injeção. Além disso, a corrosão dos componentes (como pistão e bico injetor) e falhas estruturais devido a trincas também são relevantes.

As falhas podem ser classificadas em dois tipos: globais e locais. Falhas globais afetam todos os cilindros do motor simultaneamente, enquanto falhas locais ocorrem em

um cilindro específico, deixando os demais em funcionamento normal.

A Figura 10 detalha como falhas em componentes do motor Diesel, como turbocompressor, sistemas de admissão, exaustão e injeção, corrosão dos componentes, sistema de arrefecimento e falhas estruturais, impactam o desempenho do motor. Essas falhas podem resultar em queda de pressão no coletor de admissão, variação na pressão de injeção, alteração da massa de ar e combustível admitido no cilindro, perda de compressão nos cilindros, variação da temperatura das paredes dos cilindros, quebra da árvore de manivelas e perda total ou parcial da combustão, levando a aquecimento do motor e perda de torque e potência.

A Figura 10 ilustra que o mau funcionamento de componentes específicos pode causar variações no desempenho do motor, frequentemente com efeitos semelhantes. Portanto, uma metodologia de diagnóstico robusta é essencial para identificar corretamente essas falhas e mitigar seus impactos no desempenho do motor Diesel.

2.4.3 Metodologia do modelo de simulação de falhas

A avaliação dos diferentes graus de severidade das falhas em máquinas diesel é essencial para a implementação eficaz de estratégias de manutenção preditiva. Ao identificar e classificar adequadamente as falhas, é possível priorizar ações de manutenção e minimizar o tempo de inatividade não planejado, garantindo assim a confiabilidade e a disponibilidade da máquina.

A pressão dentro de cada cilindro depende da variável independente $\theta(t)$ e dos parâmetros operacionais, como a pressão e temperatura no coletor de admissão. A Equação (10) representa a pressão referida,

$$P_j = f\left(\theta, P_i, T_i, r, m_i, m_f, ...\right) \text{ para } j = 1 \; ... \; 6, \tag{10}$$

onde j indica o cilindro que está sendo analisado.

As falhas no sistema de admissão e a corrosão na válvula de admissão são classificadas como falhas globais, pois causam variações na pressão do coletor de admissão (ΔP_i) e na quantidade de ar admitido em cada cilindro ($\Delta m_{i,j}$). Em contrapartida, as falhas locais incluem problemas no sistema de injeção, como corrosão ou obstrução

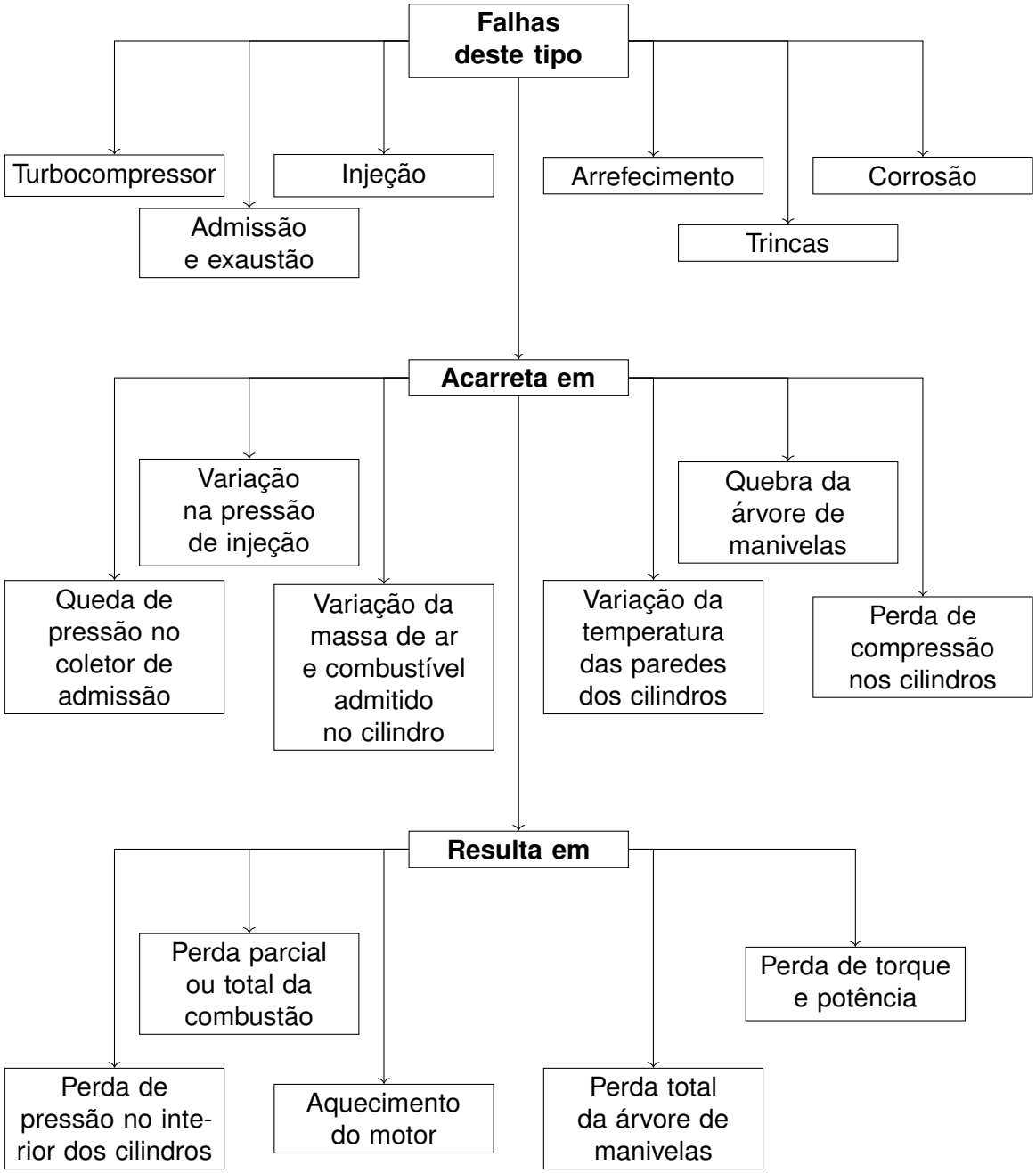


Figura 10 – Diagrama de blocos das falhas do motor Diesel e suas consequências.

do bico injetor de combustível, que resultam em variações na massa de combustível injetada em cada cilindro ($\Delta m_{f,j}$). Além disso, a corrosão do pistão é outra falha local que provoca perda de compressão no cilindro afetado (Δr_j). Portanto, o vetor de parâmetros de falha, Δf, é definido como uma função das variações percentuais desses parâmetros operacionais, conforme apresentado na Equação (11),

$$\Delta f = \{\Delta P_i, \Delta r_j, \Delta m_{i,j}, \Delta m_{i,j}, ...\} \text{ para } j = 1 \; ... \; 6, \tag{11}$$

onde cada elemento desse vetor, Δf_m, é um parâmetro de falha.

Para um único cilindro ($j = 1$), haverá quatro parâmetros de falha, enquanto para todos os cilindros (isto é, 6 cilindros), haverá dezenove parâmetros de falha a serem identificados. Como as falhas causam variação de pressão dentro do cilindro, a Equação (10) pode ser reescrita como a Equação (12),

$$P_j = f\left(\theta, \Delta f_m\right). \tag{12}$$

O modelo de simulação de falhas desenvolvido requer a solução das Equações (1) e (2), para as quais a pressão dentro do cilindro é dada pela Equação (12).

As Figuras 11–16 apresentam a variação dos parâmetros operativos do motor Diesel sob diferentes condições de falha e níveis de severidade, analisando a influência dessas falhas nos sinais de pressão e vibração torcional.

Na Figura 11, observa-se a queda porcentual da pressão no coletor de admissão com rotação fixa de 1000 RPM, mostrando variações de pressão e vibração torcional. A Figura 12 exibe a queda porcentual da compressão no cilindro 1, também a 1000 RPM, com gráficos similares.

Já na Figura 13, a queda da pressão no common rail é analisada a 1900 RPM, destacando novamente as variações de pressão e vibração. A Figura 14 mostra a variação dos parâmetros operativos em caso de queda porcentual da massa de ar admitido no cilindro 1 a 1900 RPM.

A Figura 15 ilustra a queda da massa de combustível injetado no cilindro 1 a 2500 RPM. Por fim, a Figura 16 apresenta o atraso porcentual do início da injeção no cilindro 1 a 2500 RPM. Essas figuras são essenciais para entender como diferentes falhas impactam os sinais de pressão e vibração do motor Diesel, permitindo uma análise detalhada das variações dos parâmetros operativos em condições de falha.

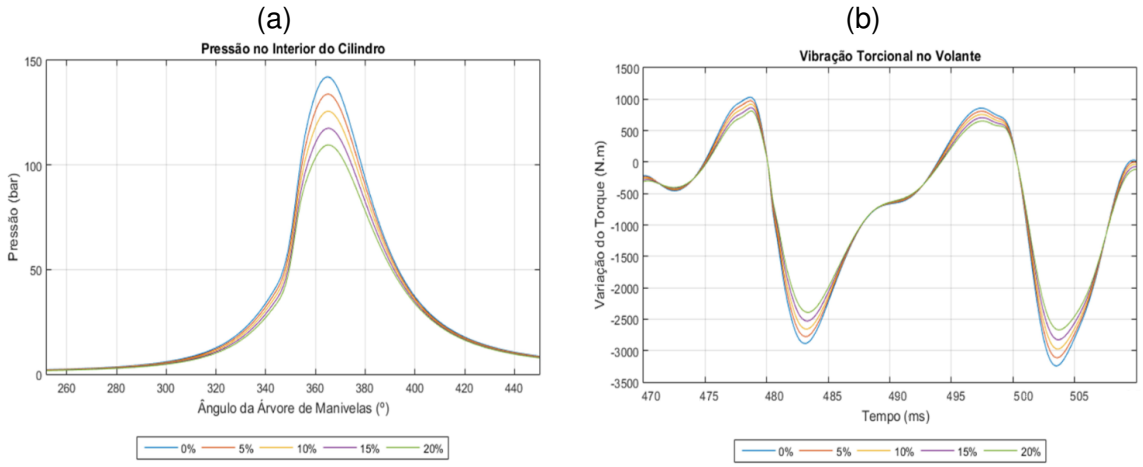


Figura 11 – Variação dos parâmetros operativos do motor Diesel, para falha da queda porcentual da pressão no coletor de admissão, para diferentes níveis de severidade e para a velocidade de rotação fixa em 1000 RPM: (a) pressão e (b) vibração torcional. Os valores percentuais mencionados correspondem aos índices de falha utilizados nos experimentos.

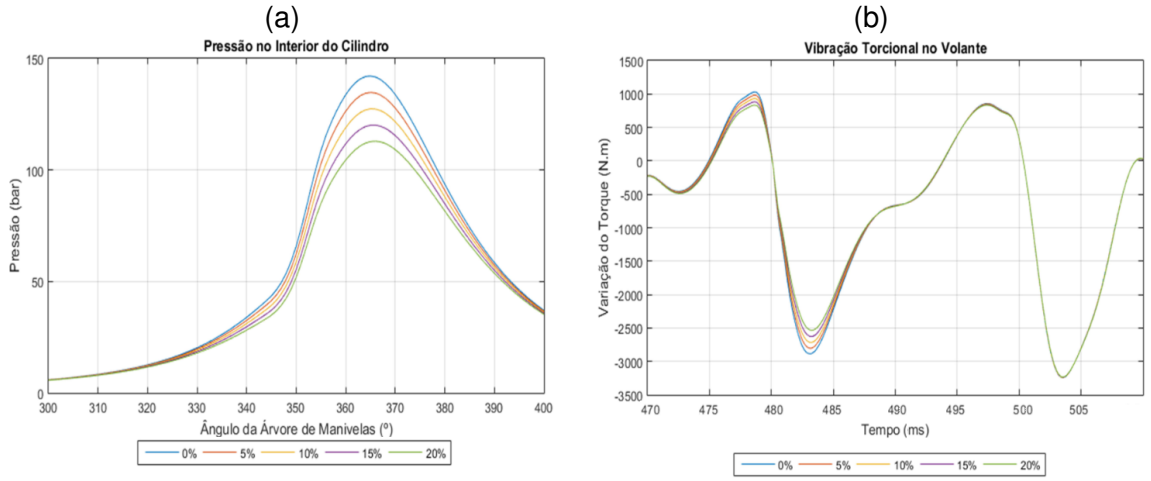


Figura 12 – Variação dos parâmetros operativos do motor Diesel, para falha da queda porcentual da compressão no cilindro 1, para diferentes níveis de severidade e para a velocidade de rotação fixa em 1000 RPM: (a) pressão e (b) vibração torcional. Os valores percentuais mencionados correspondem aos índices de falha utilizados nos experimentos.

2.5- Considerações do capítulo

Este capítulo apresentou os modelos utilizados para simular o comportamento de um motor Diesel e desenvolver os sinais de falha. O modelo do gêmeo digital incluiu um modelo termodinâmico zero dimensional, um modelo de massa concentrada para

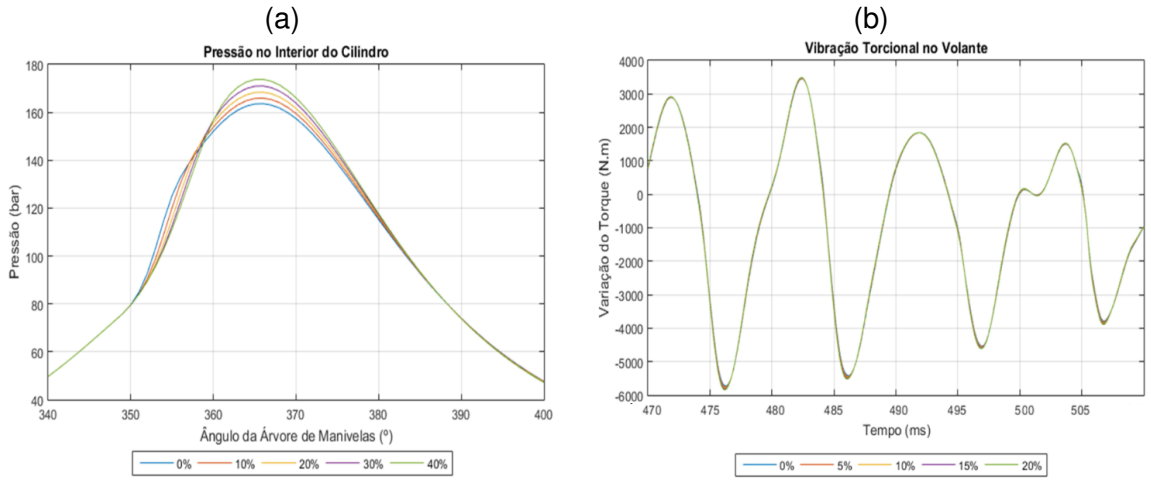


Figura 13 – Variação dos parâmetros operativos do motor Diesel, para falha da queda porcentual da pressão no common rail, para diferentes níveis de severidade e para a velocidade de rotação fixa em 1900 RPM: (a) pressão e (b) vibração torcional. Os valores percentuais mencionados correspondem aos índices de falha utilizados nos experimentos.

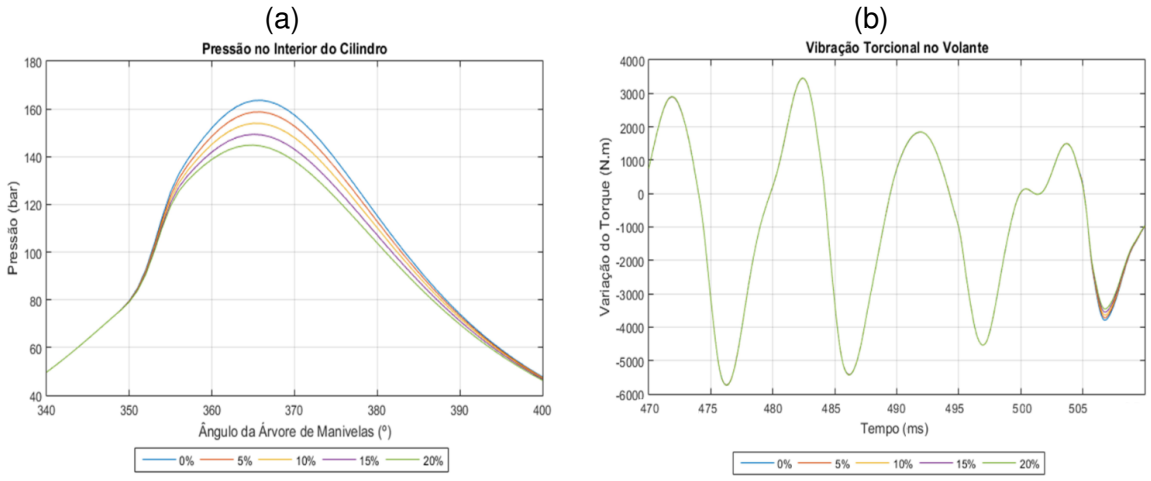


Figura 14 – Variação dos parâmetros operativos do motor Diesel, para falha da queda porcentual da massa de ar admitido no cilindro 1, para diferentes níveis de severidade e para a velocidade de rotação fixa em 1900 RPM: (a) pressão e (b) vibração torcional. Os valores percentuais mencionados correspondem aos índices de falha utilizados nos experimentos.

vibração torcional da árvore de manivelas, com onze graus de liberdade e um modelo de simulação de falhas. A validação desses modelos foi crucial para garantir a precisão das previsões e diagnósticos. Embora a validação do modelo de simulação de falhas fosse desejável, não foi possível devido à falta de disponibilidade de motores reais para testes de falha.

O modelo termodinâmico de dimensão zero gerou perfis de pressão nos cilindros, fundamentais para o diagnóstico eficaz de motores a diesel. A análise quantitativa das

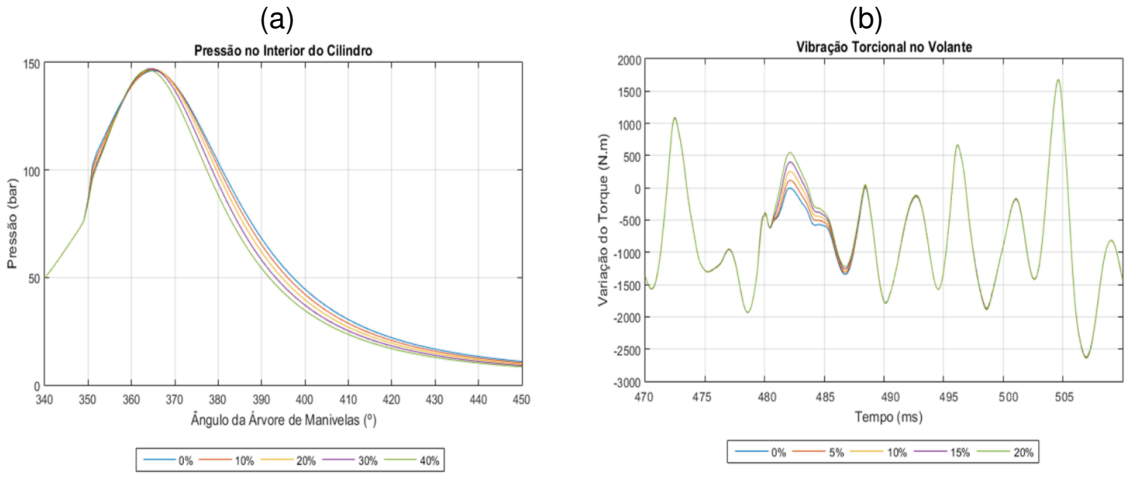


Figura 15 – Variação dos parâmetros operativos do motor Diesel, para falha da queda porcentual da massa de combustível injetado no cilindro 1, para diferentes níveis de severidade e para a velocidade de rotação fixa em 2500 RPM: (a) pressão e (b) vibração torcional. Os valores percentuais mencionados correspondem aos índices de falha utilizados nos experimentos.

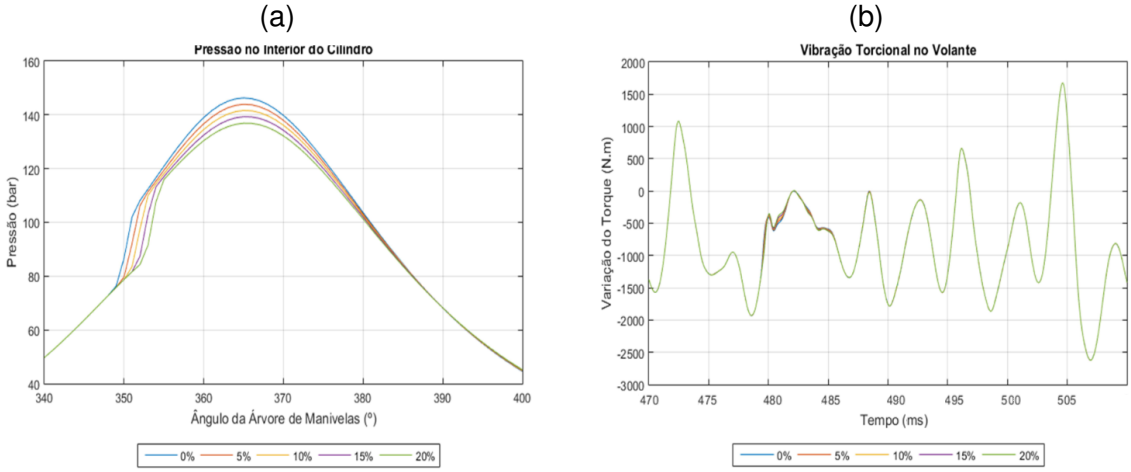


Figura 16 – Variação dos parâmetros operativos do motor Diesel, para falha do atraso porcentual do início da injeção no cilindro 1, para diferentes níveis de severidade e para a velocidade de rotação fixa em 2500 RPM: (a) pressão e (b) vibração torcional. Os valores percentuais mencionados correspondem aos índices de falha utilizados nos experimentos.

curvas de pressão, comparadas aos dados do fabricante, validou o modelo e permitiu ajustes para maior precisão. O modelo de vibração torcional da árvore de manivelas foi essencial para compreender o comportamento do sistema em diferentes condições operacionais.

A escolha de um motor Diesel de quatro tempos com sistema *common-rail* se deveu à sua eficiência e controle preciso da injeção de combustível. A avaliação da severidade das falhas, classificadas como baixa, média e alta, foi essencial para priorizar

ações de manutenção. O modelo de simulação de falhas considerou variações na pressão do coletor de admissão, massa de ar admitida, massa de combustível injetada e perda de compressão devido a falhas no sistema. Esse modelo exigiu a solução de equações para a pressão dentro do cilindro, permitindo simular falhas e diagnosticar o motor Diesel com precisão.

3- Materiais e métodos

Ao considerar as falhas em um motor Diesel, é essencial analisar as diferentes condições de operação e os possíveis efeitos adversos em seu desempenho. Inicialmente, a condição normal, sem falhas, serve como referência para compreender as variações que podem surgir em situações anormais.

Uma das falhas que pode ocorrer em um motor Diesel é a diminuição da pressão no coletor de admissão. Essa falha pode ser causada por diversos fatores, como obstruções no sistema de entrada de ar ou mau funcionamento de componentes, resultando em uma redução do fluxo de ar para os cilindros. Isso pode impactar negativamente a eficiência da combustão e, consequentemente, a potência gerada pelo motor.

Outra falha relevante é a redução da razão de compressão nos cilindros. Essa condição pode surgir devido ao desgaste ou danos nos anéis de vedação do pistão e problemas de vedação nas válvulas. A diminuição da razão de compressão pode levar a uma combustão incompleta do combustível, reduzindo a eficiência do motor e aumentando o consumo de combustível.

Além disso, a redução da quantidade de combustível injetado nos cilindros é outra falha crítica que pode ocorrer em um motor Diesel. Isso pode ser causado por obstruções no sistema de injeção de combustível, falhas nos injetores ou problemas elétricos nos circuitos de controle. Uma redução na quantidade de combustível injetado pode levar a uma combustão deficiente, diminuindo a potência do motor e aumentando as emissões de poluentes.

Essas falhas representam desafios significativos para a operação eficiente e confiável de um motor Diesel, destacando a importância de estratégias de manutenção preditiva e diagnóstico precoce. A compreensão detalhada dessas condições anormais e de seus efeitos sobre o desempenho do motor é fundamental para o desenvolvimento de soluções eficazes de manutenção e para garantir a operação segura e econômica dos motores Diesel em diversas aplicações industriais e comerciais.

3.1- Conjunto de dados

A base de dados emula diversos cenários operacionais, fornecendo informações suficientes para caracterizar e discriminar todas as falhas possíveis da máquina Diesel e as variações das condições do sistema que correspondem aos níveis de severidade. Nesta tese, a base de dados desenvolvida cobriu quatro condições operacionais, consoante às Seções 3.2.1 a 3.2.4.

Em todos os cenários de falhas deste estudo, a frequência de rotação do motor foi definida em 2500 RPM. De acordo com (Gutiérrez, 2016), essa rotação foi escolhida por apresentar a menor taxa de erro conjunto na estimativa da pressão média indicada e da pressão máxima do ciclo de combustão, quando comparados os dados experimentais fornecidos pelo fabricante com os dados simulados durante a etapa de validação dos modelos termodinâmicos e dinâmicos.

A base de dados compreende um total de 3500 cenários de falha diferentes para quatro condições operacionais distintas. Foi idealizado um conjunto de 250 cenários de falhas para cada um dos seis cilindros e para a classe normal. Desse total, 250 cenários pertencem à classe normal, 250 à classe “redução de pressão no coletor de admissão”, 1500 à classe “redução na razão de compressão nos cilindros” (250 cenários de falhas para cada cilindro) e 1500 à classe “redução da quantidade de combustível injetado nos cilindros” (250 cenários de falhas para cada cilindro). Esta base de dados, denominada “Conjunto de Características de Falhas de Motores Diesel (3500-DEFault)”, está disponível no Mendeley *datasets*[1] (Pestana-Viana, 2021).

De posse do conjunto de dados, adotar técnicas de regressão e aprendizado de máquinas na identificação de severidades em motores Diesel é fundamental por várias razões. Essas técnicas oferecem uma abordagem sistemática e eficiente para analisar grandes conjuntos de dados. Permitem identificar padrões complexos e relações não lineares entre variáveis, o que é crucial para diagnosticar a severidade de falhas.

Além disso, as técnicas de regressão e aprendizado de máquinas têm a capacidade de lidar com dados heterogêneos e incompletos, comuns em ambientes industriais. Ao utilizar essas abordagens, é possível construir modelos preditivos precisos que podem detectar sinais sutis de falha, prever seu desenvolvimento e avaliar sua severidade. Isso

[1] Cabe ressaltar que esta base de dados foi construída pelo autor desta tese.

contribui para uma manutenção mais proativa e eficaz, reduzindo o tempo de inatividade não planejado e os custos associados à manutenção corretiva.

Em suma, a aplicação de técnicas de regressão e aprendizado de máquinas oferece uma maneira robusta e automatizada de identificar e avaliar a severidade de falhas em motores Diesel, promovendo uma operação mais confiável e eficiente.

3.2- Classes de falhas

3.2.1 Normal (sem falhas)

Para a classe **normal** não há falha implementada. 250 diferentes instâncias (realizações) são criadas na simulação através da variável $\Delta f_m^{(n)}$, adotada a partir da inserção de 0,1% de severidade máxima com distribuição de probabilidade normal Gaussiana, cobrindo uma faixa entre 0 e 0,1%, isto é, até dez vezes menor que o valor de severidade mínima das classes com falha, nas 27 variáveis de severidade de entrada adotadas dos modelos dinâmicos e termodinâmicos ΔP_a, ΔP_r, ΔT_p, Δr_i, Δm_{a_i}, Δm_{c_i}, $\Delta \theta_{inj_i}$.

O objetivo deste passo é emular o motor real em operação normal, onde as variáveis da máquina divergem dentro de uma pequena faixa, devido a variabilidade ciclo a ciclo da mistura ar-combustível e podendo ainda apresentar pequena dispersão nos ângulos de abertura e fechamento das válvulas da câmara de combustão (Abdulaziz and Khan, 2014; Arcoumanis and Kamimoto, 2009; Heywood, 1988; Zannis and Hountalas, 2005).

3.2.2 Redução de pressão no coletor de admissão

Na classe **redução de pressão no coletor de admissão**, são considerados vários cenários com severidades de $[1,0;\ 1,2;\ 1,4;\ \cdots;\ 50,0]$% para a variável ΔP_a,

totalizando 250 cenários.

3.2.3 Redução de razão de compressão nos cilindros

Esta classe envolve todos os cilindros para criar cenários com severidades de $[1,0;\ 1,2;\ 1,4;\ \cdots;\ 50,0]$% relacionados às variáveis Δr_i e aos cilindros $i \in \{1;\ 2;\ 3;\ 4;\ 5;\ 6\}$. 250 cenários diferentes de **redução de razão de compressão** para cada cilindro são considerados, resultando em um total de 1500 cenários de redução de razão de compressão.

3.2.4 Redução da quantidade de combustível injetado nos cilindros.

Da mesma forma que a classe anterior, e todos os cilindros são utilizados para criar os cenários com severidades de $[1,0;\ 1,2;\ 1,4;\ \cdots;\ 50,0]$% relacionados às variáveis Δm_{c_i} e aos cilindros $i = [1;\ 2;\ 3;\ 4;\ 5;\ 6]$. Também são avaliados 250 cenários diferentes de **redução da quantidade de combustível injetado** para cada cilindro, totalizando 1500 cenários abordados de redução da quantidade de combustível injetado nos cilindros.

3.3- Processo de Ruído Branco Aditivo Gaussiano

O conjunto de dados 3500-DEFault foi submetido a diferentes níveis $L = [60, 30, 15, 0]$ dB de SNR, utilizando ruído branco gaussiano e aditivo (AWGN), nos sinais de originais, a fim de avaliar sua influência no desempenho da regressão e emular um sinal medido.

Também são considerados os diferentes níveis de L de AWGN rotulados como $\check{\mathbf{v}}_s = [\check{M}(t, L), \check{P}(\theta, L), \check{T}(\theta, L)]$, tal que $\check{\mathbf{v}}_s$ é um vetor com os sinais corrompidos originais. A Figura 17 mostra o processo de adição de ruído branco, para cada elemento deste vetor.

O vetor de sinais do motor $\mathbf{v}_s$ é representado pelos sinais $M(t)$, $P(\theta)$ e $T(\theta)$,

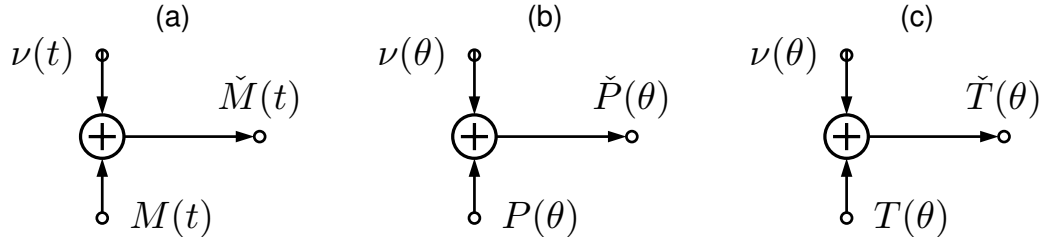


Figura 17 – Diagrama de bloco do processo de adição de ruído: (a) $M(t)$, (b) $P(\theta)$ e (c) $T(\theta)$. $\nu(t)$ e $\nu(\theta)$ representam o sinal de ruído branco com relação ao tempo e ângulo da árvore de manivelas, considerando diferentes níveis L.

conforme Eq. (13):

$$\mathbf{v}_s = \begin{bmatrix} M(t) & P(\theta) & T(\theta) \end{bmatrix}^T. \tag{13}$$

A adição de ruído branco para cada elemento i do vetor $\check{\mathbf{v}}_s$, ilustrada na Figura 17, pode ser apresentado conforme o conjunto de Equações (14):

$$\begin{aligned} \check{M}(t) &= M(t) + \nu(t), \\ \check{P}(\theta) &= P(\theta) + \nu(\theta), \\ \check{T}(\theta) &= T(\theta) + \nu(\theta). \end{aligned} \tag{14}$$

As operações realizadas na Eq.(14) e na Figura 17 podem ser resumidas conforme a Eq. (15):

$$\check{v}_{s_i} = v_{s_i} + \nu(\cdot). \tag{15}$$

O ruído branco $\nu(\cdot)$ pode ser caracterizado como um processo gaussiano aleatório de média zero e variância equivalente ao nível L de SNR, sendo mostrado na Eq. (16),

$$\nu(\cdot) \sim \mathcal{N}(0, \sigma_N^2). \tag{16}$$

O SNR pode ser definido como:

$$SNR = \frac{\max[{v_{s_i}}^2]}{E[\nu(\cdot)^2]} = \frac{{A_V}^2}{{\sigma_N}^2}, \tag{17}$$

onde $E[\cdot]$ é o valor esperado e A_V é o valor máximo para cada elemento i do vetor $\mathbf{v}_s$.

A SNR é calculado em dB a partir da fórmula:

$$\text{SNR}_{\text{dB}} \triangleq 10\ \log_{10}(\text{SNR}), \tag{18}$$

substituindo a Eq. (17) na Eq. (18), temos:

$$\text{SNR}_{\text{dB}} = L = 10\ \log_{10}\left(\frac{{A_V}^2}{{\sigma_N}^2}\right). \tag{19}$$

Assim, podemos obter a variância de ruído ${\sigma_N}^2$, a partir do nível desejado de ruído L em dB, como mostrado na Eq. (20),

$${\sigma_N}^2 = 10^{-\frac{L}{10}}{A_V}^2. \tag{20}$$

A função de densidade de probabilidade (PDF) da variável $\nu(\cdot)$ para diferentes níveis L, é mostrada na Figura 18.

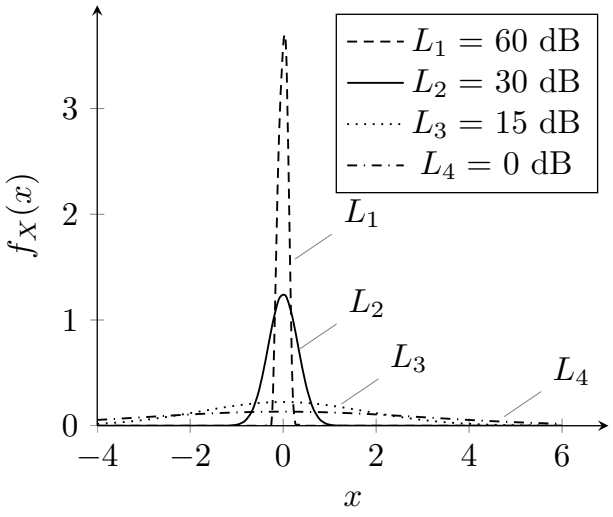


Figura 18 – Função densidade de probabilidade da variável $\nu(\cdot)$ para L diferentes níveis de ruído.

A Figura 19 mostra as variáveis Pressão $P(\theta)$, torque externo $M(t)$ e Torque $T(\theta)$, conforme Equações (1), (3) e (9), respectivamente. Para um único sistema de biela-manivela nas condições com a adição de ruído, conforme descrito nas Equações (13), (14) e (15).

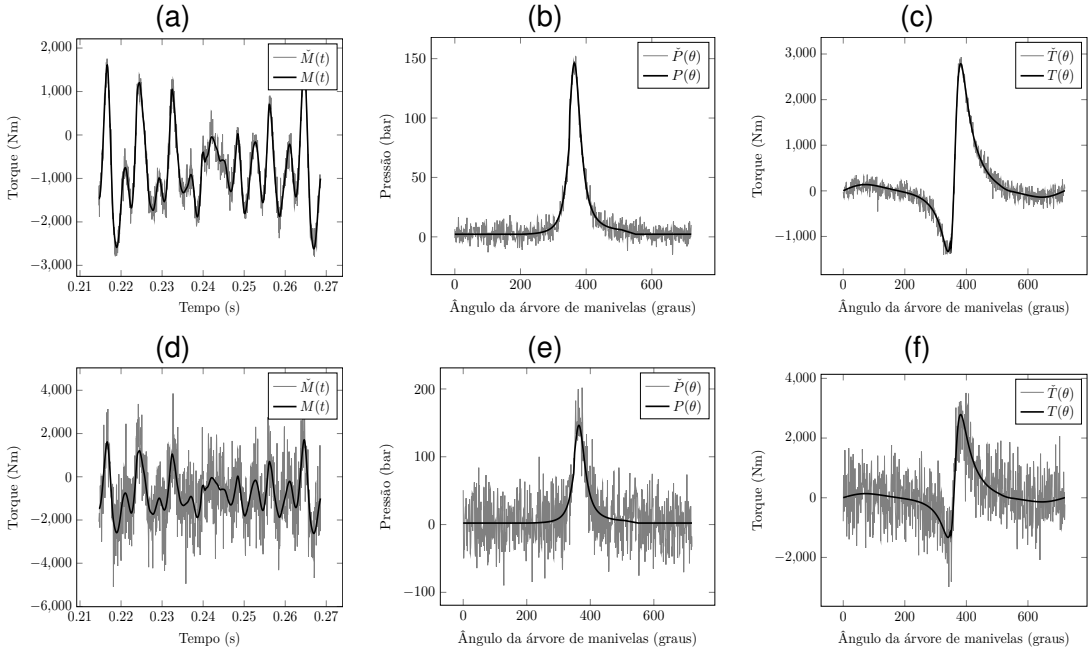


Figura 19 – Processo de aplicação de ruído aditivo: (a) $M(t)$ com 15 dB, (b) $P(\theta)$ com 15 dB, (c) $T(\theta)$ com 15 dB, (d) $M(t)$ com 0 dB, (e) $P(\theta)$ com 0 dB e (f) $T(\theta)$ com 0 dB. No gráfico acima, a linha cinza é o sinal com adição de ruído e a linha preta é o sinal original sem ruído. A nova variável com AWGN $\check{v}_{s_i}(\cdot)$, é $\check{v}_{s_i}(\cdot) = v_{s_i}(\cdot) + \ nu(\cdot)$, onde $\nu(\cdot)$ é o ruído gaussiano branco com $L = [15, 0]$ dB SNR.

3.4- Normalização do conjunto de dados

A normalização é amplamente utilizada em estudos estatísticos para padronizar dados e otimizar o processamento de informações. Em aprendizado de máquina, a normalização desempenha um papel muito importante quando existem atributos que podem prejudicar o processamento de dados, como valores redundantes ou extremos.

Dessa forma, normalizar dados é uma maneira de padronizar e minimizar os problemas decorrentes dessas dispersões ou redundâncias, tornando o processamento de dados eficiente e evitando a análise inconsistente dos dados (Jadhav et al., 2020). Neste trabalho, foi utilizado um tipo de normalização Mínimo-Máximo.

3.4.1 Normalização mínimo-máximo

Esta técnica realiza a normalização dos valores através de seus valores mínimos e máximos, limitando-os ao intervalo entre 0 e 1, segundo:

$$x_{norm}(i) = \frac{\mathbf{x}(i) - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}, \tag{21}$$

onde $x_{norm}(i)$ é o elemento normalizado de $\mathbf{x}(i)$, $\min(\mathbf{x})$ é o menor valor do vetor $\mathbf{x}$ e $\max(\mathbf{x})$ é o maior valor do vetor $\mathbf{x}$.

No entanto, o desvio padrão diminui ao transformar os dados em uma escala entre 0 e 1 e, dessa forma, pode-se excluir informações dos dados $outliers$, que podem de trazer informações importantes para a análise do banco de dados (Patel et al., 2020).

3.5- Particionamento o conjunto de dados

Uma etapa crucial no aprendizado de máquina envolve a divisão das amostras em duas partes distintas: o conjunto de treinamento e o conjunto de teste. As amostras de treinamento são utilizadas para instruir o modelo matemático, neste caso, um regressor, sobre os padrões de dados identificados, enquanto as amostras de teste são reservadas para avaliar a eficácia do modelo selecionado. Adicionalmente, o conjunto de treinamento pode ser subdividido em duas partes: treinamento e validação. Este último é empregado para ajustar os parâmetros do modelo, refinando ainda mais sua precisão e generalização.

Quando a base de dados usada é suficientemente grande, a técnica *holdout* (Yadav and Shukla, 2016) é comumente utilizada para separar os conjuntos, sendo usualmente 70% das amostras utilizadas para treinamento, 10% para validação e 20% para o teste. No entanto, na maioria dos problemas de aprendizado de máquina o uso desta técnica pode gerar um viés no resultado, visto que cada um destes conjuntos pode não ter um quantitativo suficiente de amostras para avaliar o modelo em cada etapa de desenvolvimento. Assim, as técnicas de validação cruzada podem ser utilizadas para mitigar este problema, dentre as quais o k-fold se destaca, produzindo resultados

geralmente mais robustos do que o método *Hold out*.

O método k-fold consiste em particionar aleatoriamente o conjunto de dados em k blocos de tamanho aproximadamente igual. Subsequentemente, usa $k-1$ conjuntos para realizar o treinamento do modelo e o conjunto restante para realizar o teste. Este processo é realizado até que cada bloco seja usado como teste. O desempenho deste método é dada pela acurácia média associada ao desvio padrão. Os valores de k escolhidos com frequência são $k = 5$ ou $k = 10$. Nesta tese, foram utilizados cinco blocos para o k-fold, como visto na Figura 20, onde os blocos cinza correspondem ao conjunto de teste e o bloco azul corresponde ao conjunto de treinamento.

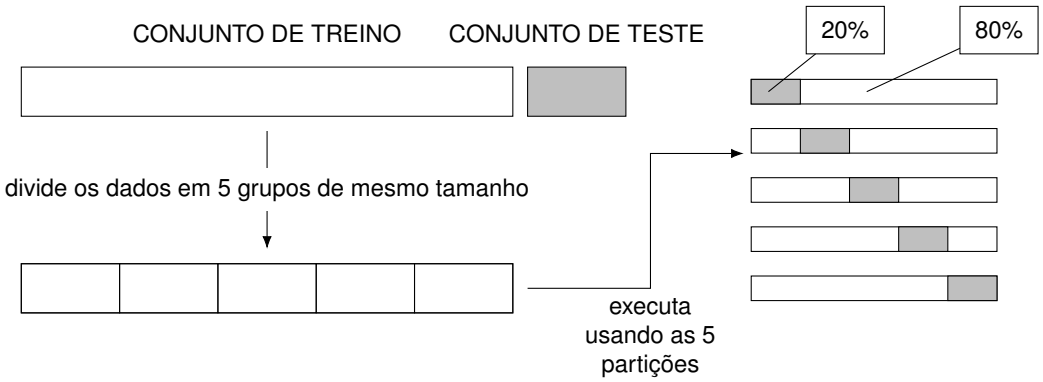


Figura 20 – Técnica do k-fold, onde os blocos cinza correspondem aos conjuntos de teste, enquanto os blocos brancos correspondem aos conjuntos de treinamento.

3.6- Regressão

Dentro dos métodos de aprendizado de máquina – *Machine Learning* (ML), a regressão é uma técnica utilizada para modelar as relações entre variáveis dependentes e independentes por meio de métodos estatísticos. Esses métodos podem ser lineares, quadráticos, polinomiais, entre outros, dependendo do algoritmo utilizado e da natureza do problema a ser resolvido. A regressão busca encontrar uma função que melhor se ajuste aos dados observados, permitindo prever ou estimar valores da variável dependente com base nas variáveis independentes disponíveis. Nesta tese, o método de regressão utilizado é do tipo linear.

A regressão linear pode ser uma ferramenta adequada para identificar a severidade

de falhas em motores Diesel, dependendo do contexto e das características dos dados disponíveis. No entanto, sua adequação depende de vários fatores, incluindo a natureza das variáveis envolvidas, a relação entre elas e a precisão desejada na identificação da severidade da falha (Hountalas, 2000).

A regressão linear é útil quando há uma relação linear ou aproximadamente linear entre as variáveis independentes e dependentes. Se a relação entre a variável de entrada (que descreve a condição da máquina) e a variável de saída (que indica a severidade da falha) for linear ou puder ser aproximada por uma linha reta, então a regressão linear pode ser uma escolha adequada (Haykin, 2008).

Entretanto, em muitos casos, a relação entre a condição da máquina e a severidade da falha pode não ser estritamente linear. Nesses casos, técnicas de modelagem mais avançadas, como regressão por aprendizado de máquina, podem ser mais apropriadas para capturar a complexidade da relação entre as variáveis (Heywood, 1988).

Considerando o modelo linear com múltiplas variáveis, dado um conjunto de dados $y_i, , x_{i1}, \ldots, x_{ip}{}_{i=1}^{n}$ com n variáveis, um modelo de regressão linear assume que a relação entre a variável dependente y e o vetor x é linear. Essa relação é modelada mediante uma equação cujos coeficientes β são definidos pela regressão, podendo-se também adotar variável aleatória de erro ε que quantifica a falha do modelo em ajustar-se aos dados. Existem diversas abordagens para calcular os coeficientes β da regressão linear, as técnicas baseadas em mínimos quadrados ordinários e gradiente descendente são as mais comuns (Soto, 2021). Por fim, o modelo tem a forma:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\mathsf{T}} \beta + \varepsilon_i, \qquad i = 1, \ldots, n. \tag{22}$$

Frequentemente, essas equações são agrupadas em notação matricial como:

$$\mathbf{y} = \mathbf{X}\beta, \tag{23}$$

onde $\mathbf{y}$ é um vetor de tamanho n dos valores de saída, $\mathbf{X}$ é uma matriz de tamanho $n \times p$ dos valores de entrada e β é um vetor de tamanho p dos coeficientes. A notação de transposta, $[\cdot]^T$, ainda se aplica de maneira que ${x_i}^T\beta$ é o produto interno entre os vetores x_i e β,

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon, \tag{24}$$

$$\text{onde, } \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} \mathbf{x}_1^{\mathsf{T}} \\ \mathbf{x}_2^{\mathsf{T}} \\ \vdots \\ \mathbf{x}_n^{\mathsf{T}} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix} \text{ e } \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

3.6.1 Redes neurais artificiais - Perceptron multicamadas

Uma rede neural é uma rede ou circuito de neurônios interconectados da mesma forma que uma rede neural artificial (RNA) é composta por neurônios artificiais ou nós de processamento (Samarasinghe, 2016). Portanto, uma rede neural pode ser biológica quando formada por neurônios reais, ou uma rede artificial, quando compostas por unidade de processamento e empregada para resolver problemas de inteligência artificial (IA) (Hung et al., 2019; Aggarwal, 2018; Samarasinghe, 2016).

O neurônio artificial ou unidade de processamento recebe um conjunto de sinais de entrada, cada um multiplicado por um peso sináptico. Os sinais de entrada ponderados pelas sinapses do neurônio são então somados e aplicados a uma função de ativação ϕ, que define a saída do neurônio em termos do nível de atividade na sua entrada e passa o resultado aos neurônios seguintes (Hung et al., 2019; Samarasinghe, 2016), conforme ilustrado na Figura 21, e na Equação abaixo:

$$\phi\left(\sum_{i=1}^{N} w_i \, x_i\right) = \phi(\mathbf{w}^T\mathbf{x}), \tag{25}$$

dessa forma, a função de ativação para a camada de saída pode ser dada por:

$$\phi(\mathbf{w}^T\mathbf{x}) = \mathbf{w}^T\mathbf{x}. \tag{26}$$

Neste trabalho, a função sigmoidal foi usada como função de ativação, caracterizando uma saída não linear entre dois estados, muito utilizada para o reconhecimento de padrões de falha não lineares, conforme descrito na Eq. (27):

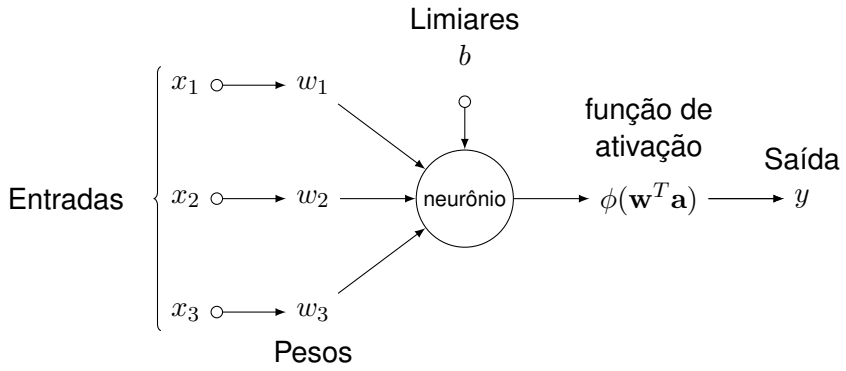


Figura 21 – Esquema de único neurônio artificial.

$$\phi(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})}. \tag{27}$$

A estrutura dos neurônios, ou seja, sua topologia, está diretamente relacionada às habilidades de aprendizado que podem ser exploradas durante o treinamento da rede neural. Neste trabalho, foi utilizada a rede *feedforward* de camadas múltiplas conhecida como *Multi-Layer Perceptron* (MLP) (Haykin, 2009). Esta escolha se deve à capacidade comprovada do MLP de lidar eficientemente com problemas que não são linearmente separáveis, em função dos neurônios presentes em suas camadas ocultas.

Sendo assim, nosso objetivo é treinar uma rede usando dados rotulados utilizando o processo de aprendizado supervisionado, cujo objetivo é fazer com que os parâmetros da rede sejam ajustados para minimizar o erro de saída correspondente a um determinado padrão de entrada $\mathbf{x}_i$ e saída desejada y_i. Após o treinamento, rede e seus pesos treinados formam uma função, denominada $h(\cdot)$, capaz de previsões para qualquer entrada não rotulada (Aggarwal, 2018; Hung et al., 2019).

Observe que uma rede perceptron multicamadas (MLP) é ilustrada na Figura 22, de forma o número de neurônios e quantidade de camadas ocultas inserem complexidade ao modelo e, consequentemente, modificam a capacidade da rede em fazer previsões para diversos problemas.

A saída da rede pode ser representada como uma combinação linear dos pesos e dados de entrada, ou seja:

$$y_i = h(\mathbf{x}_i, \mathbf{w}) = \mathbf{w}^T\mathbf{x}_i. \tag{28}$$

Nossa tarefa então é encontrar os pesos que forneçam o melhor ajuste aos nossos

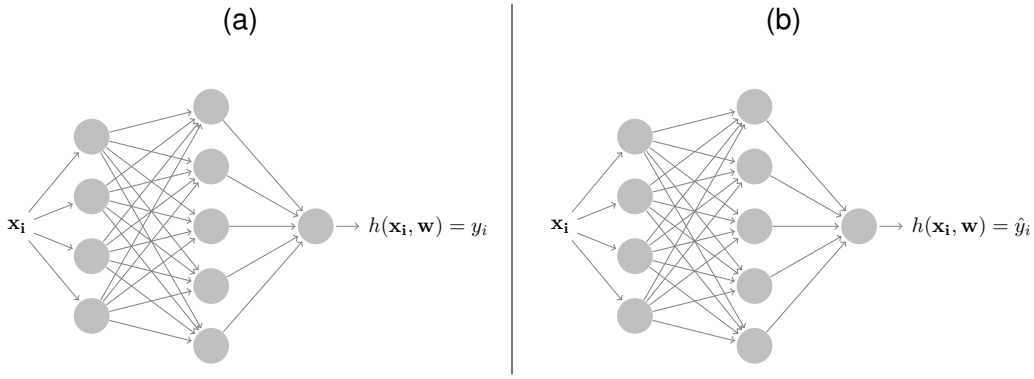


Figura 22 – Rede perceptron multicamada (MLP) (a) treino utilizando pares rotulados $\mathbf{x_i}$ e y_i para aprender pesos e (b) teste utilizando dados não rotulados $\mathbf{x_i}$ para fazer previsões.

dados de treinamento. Uma forma de medir o nosso ajuste é calcular o erro de acordo com uma função de custo, que pode ser definida como:

$$L(\mathbf{w}) = \sum_{i=1}^{N} \left(h(\mathbf{x}_i, \mathbf{w}) - y_i\right)^2. \tag{29}$$

Para encontrar este melhor ajuste, devemos minimizar a função $L(\mathbf{w})$. Esta minimização possui uma solução fechada via mínimos quadrados ordinários (num contexto de regressão linear), mas a solução generalizada emprega a minimização utilizando o método de gradiente descendente (Haykin, 2009).

Utilizando o gradiente descendente da função de custo $\nabla_{\mathbf{w}} L(\mathbf{w})$, primeiro derivamos o gradiente em relação a um determinado peso $w_{j\to k}$, que conecta o neurônio j ao k. Entradas também são tratadas como neurônios, visto que existe um peso $w_{j\to k}$ de cada conexão da entrada para um neurônio da primeira camada oculta (Aggarwal, 2018). De forma geral, temos:

$$\frac{\partial}{\partial w_{j\to k}} L(\mathbf{w}) = \sum_{i=1}^{N} 2\left(h(\mathbf{x}_i, \mathbf{w}) - y_i\right) \frac{\partial}{\partial w_{j\to k}} h(\mathbf{x}_i, \mathbf{w}). \tag{30}$$

Neste ponto, devemos calcular o gradiente de nossa função de rede em relação ao peso em questão $\frac{\partial}{\partial w_{j\to k}} h(\mathbf{x}_i, \mathbf{w})$. Considerando a rede de 3 entradas acima, a função de rede é representada por $h(\mathbf{x}_i, \mathbf{w}) = w_1 \mathbf{x}_i^{(1)} + w_2 \mathbf{x}_i^{(2)} + w_3 \mathbf{x}_i^{(3)}$, tal que gradiente completo pode ser descrito como:

$$\nabla_{\mathbf{w}} L(\mathbf{w}) = \left(\frac{\partial L(\mathbf{w})}{\partial w_1}, \frac{\partial L(\mathbf{w})}{\partial w_2}, \frac{\partial L(\mathbf{w})}{\partial w_3} \right) = \left(\sum_{i=1}^{N} 2\mathbf{x}_i^{(1)} h(\mathbf{x}_i, \mathbf{w}), \sum_{i=1}^{N} 2\mathbf{x}_i^{(2)} h(\mathbf{x}_i, \mathbf{w}), \sum_{i=1}^{N} 2\mathbf{x}_i^{(3)} h(\mathbf{x}_i, \mathbf{w}) \right). \tag{31}$$

Dessa forma, podemos atualizar os pesos utilizando o gradiente descendente:

$$\mathbf{w_{k+1}} = \mathbf{w_k} - \eta \nabla_{\mathbf{w}} L(\mathbf{w}). \tag{32}$$

Deve-se atentar para a seleção o tamanho do passo η, de forma que valores muito pequenos tornam o aprendizado muito lento e valores muito elevados podem comprometer a convergência da minimização. Após uma quantidade definida de k ciclos de treinamento (épocas), os pesos finais correspondem ao melhor ajuste obtido (Samarasinghe, 2016).

Com a rede treinada, o teste consiste em obter uma previsão para amostra de teste $\mathbf{x}_i$ usando $h(\mathbf{x}_i, \mathbf{w})$ de forma que o erro do teste é calculado exatamente como no treinamento:

$$L(\mathbf{w}) = \sum_{i=1}^{N} \left(h(\mathbf{x}_i, \mathbf{w}) - y_i \right)^2 = \sum_{i=1}^{N} \left(\hat{y}_i - y_i \right)^2. \tag{33}$$

3.6.2 *Random Forest*

É um algoritmo de aprendizado de máquina supervisionado que pode ser usado para tarefas de classificação e regressão (Breiman, 2001). Este algoritmo é formado por elementos preditores simples, ou seja, árvores de decisão, conforme ilustrado na Figura 23.

A associação destes preditores simples em comitês (*ensemble*) (Rokach, 2019), ou seja, a inclusão destas “árvores” em “florestas”, geralmente se mostram poderosos do que cada preditor individual (Yang et al., 2008; Rokach, 2019). A técnica de *Random Forest* fornece excelentes resultados quando utilizada na Detecção de previsão de defeitos (de SS Martins et al., 2018; Pestana-Viana et al., 2019), sendo aqui utilizada para a tarefa de regressão.

As árvores de decisão que formam o algoritmo RF são preditores instáveis. Em ou-

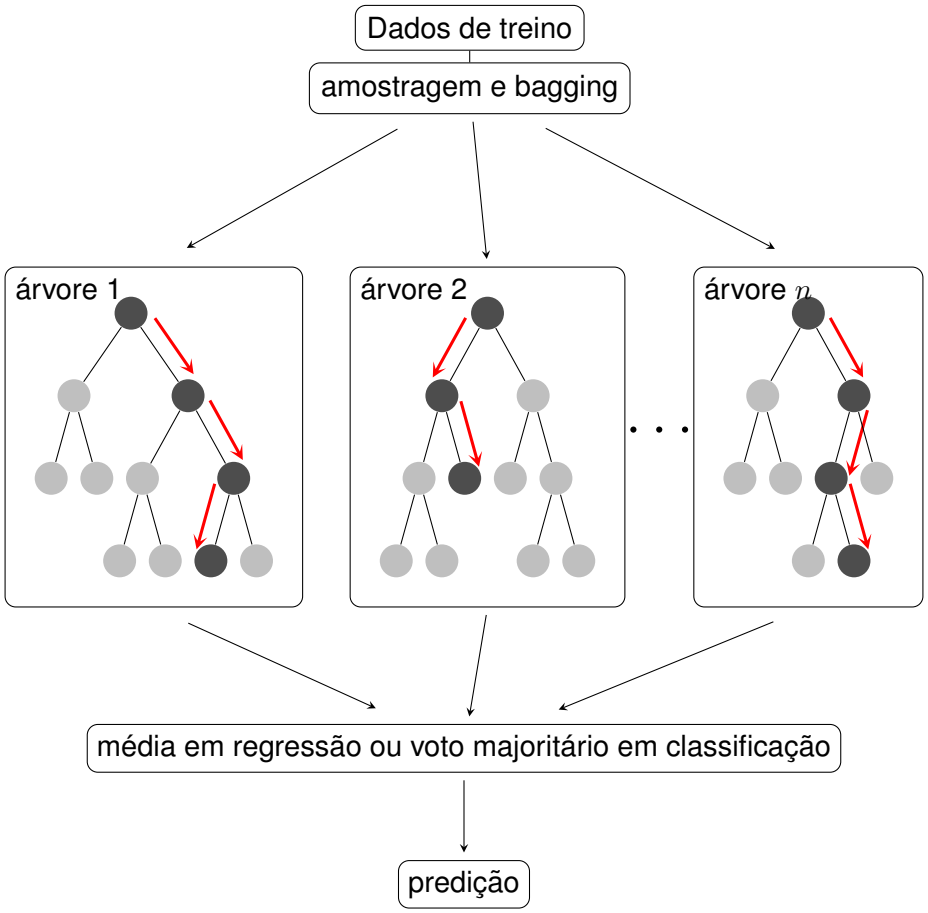


Figura 23 – Diagrama do algoritmo *Random Forest*.

tras palavras: pequenas mudanças no conjunto de treinamento são capazes de provocar diferenças significativas nos resultados do classificador ou da regressão. Por essa razão, o método *bagging* é utilizado para mitigar esse problema, aumentando desempenho da previsão em relação a uma única árvore e reduzindo a variância da previsão (Yang et al., 2008).

As árvores agrupadas segundo o método *bagging* não são totalmente independentes umas das outras, pois os preditores originais são usados nas divisões das árvores (Breiman, 1996). Cada modelo do conjunto é usado para criar uma predição para um novo exemplo, de forma que precisão final do modelo obtida pela média dos resultados de cada árvore.

Dado um conjunto de treinamento $X = x_1, \cdots, x_n$ com saídas $Y = y_1, ..., y_n$, realiza-se o *bagging* repetidamente (B vezes) e treinam-se as árvores amostras, seguindo para $b = 1, ..., B$:

- Amostragem com substituição de n exemplos de treinamento de X, Y, chamados

de X_b, Y_b;

- Treinamento de uma árvore de classificação ou regressão f_b em X_b, Y_b.

Após o treinamento, as previsões para amostras inéditas x' podem ser feitas calculando a média das previsões de todas as árvores de regressão individuais $\hat{f}$ em x', ou por maioria de votos no caso de árvores de classificação, conforme equação a seguir:

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x'). \tag{34}$$

O número de amostras/árvores, B, é um parâmetro livre. Normalmente, algumas centenas a vários milhares de árvores são usadas, dependendo do tamanho e da natureza do conjunto de treinamento. Um número ideal de árvores B pode ser encontrado usando validação cruzada ou observando o erro *out-of-bag*: o erro médio de previsão em cada amostra de treinamento x_i, usando apenas as árvores que não têm x_i em seu exemplo de Bootstrap (James et al., 2014). O erro de treinamento e teste tende a se estabilizar após o ajuste de um certo número de árvores.

O algoritmo *Random Forest* adiciona a seleção um subconjunto aleatório de características neste procedimento citado, sendo este processo é algumas vezes chamado *feature bagging* (Altmann et al., 2010). Dessa forma, diminui-se a correlação das árvores sob uma amostra de *Bootstrap* comum (Larose, 2015).

Dessa forma, a baixa variância do algoritmo *Random Forest* se deve ao fato desta seleção de preditores fortes e complexos que apresentam baixa correlação, proporcionando uma diminuição nas taxas de erro. Devido à independência de escolha de cada preditor em relação aos demais, o *Random Forest* é eficaz com relação às respostas contendo ruído.

Adicionalmente, o algoritmo *Random Forest* é computacionalmente mais eficiente que o *bagging*, visto que no processo de construção do algoritmo apenas parte dos preditores originais são analisados em cada divisão. Além disso, o algoritmo pode ser paralelizado, o que torna este método bastante eficiente no aspecto computacional (Genuer et al., 2017).

3.6.3 Máquinas de vetores suporte

As máquinas de vetores de suporte (SVM) consistem numa técnica estatística de aprendizado supervisionado, formulada a partir do princípio da minimização do risco estrutural (*structural risk minimization*) (Cortes and Vapnik, 1995; Kavitha S et al., 2016; Mustafa et al., 2018).

Vetores de suporte são pontos do conjunto amostral que possuem um papel relevante, pois a remoção dos demais pontos não altera o modelo do classificador ou regressor. Ao representar o modelo por meio de poucos pontos, é possível obter uma certa esparsidade, além de reduzir o custo computacional (Kavitha S et al., 2016). Nesta tese, será avaliada a máquina de vetores de suporte voltada para regressão (SVR) (Mustafa et al., 2018).

A Figura 24 ilustra o SVM treinado com duas classes, destacando os vetores de suporte a partir de duas amostras (em vermelho) para cada classe. Esta figura evidencia a esparsidade do modelo, uma vez que nem todos os pontos do conjunto amostral são utilizados para definir a região de separação entre as duas classes.

O algoritmo de regressão da máquina vetor de suporte utiliza uma função de perda insensível a ε que ignora os erros que estão além de uma certa distância dos valores considerados válidos, ou seja, erros só são permitidos se forem menores que ε (Schlkopf et al., 2018).

Seja $\mathbf{X}$ um conjunto de treinamento com n amostras $x_i \in X$ e $y_i \in Y$ seus respectivos rótulos:

$$f(x) = \mathbf{w}.\mathbf{x} + b, \tag{35}$$

onde o vetor de pesos normal ao hiperplano é dado por $\mathbf{w}$, a constante b representa o limiar, e a $\frac{b}{\|\mathbf{w}\|}$ é o valor do deslocamento do hiperplano com relação à origem (Mustafa et al., 2018).

Considerando a função linear da Equação (35), o algoritmo SVR visa minimizar a norma $\mathbf{w}$ solucionando um problema de otimização quadrática apresentado na Equação (36) e com as restrições mostradas na Eq. (37), conforme (Schlkopf et al., 2018), é mostrada abaixo:

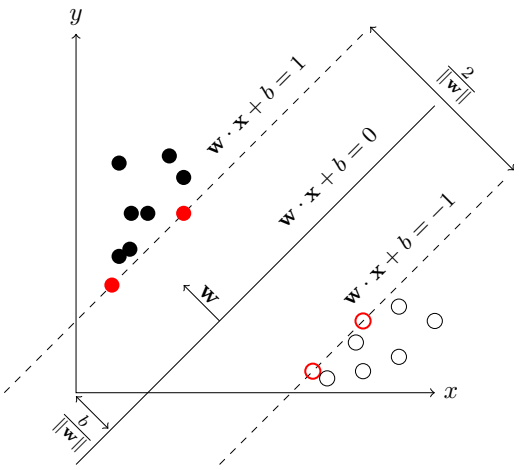


Figura 24 – SVM treinado com amostras de duas classes. Os círculos vermelhos são chamados de vetores de suporte, visando formar o hiperplano que separa as duas classes do espaço de características.

$$\underset{\mathbf{w},b}{\textit{Minimizar}} \ \frac{1}{2}\|\mathbf{w}\|^2 , \tag{36}$$

$$\textit{sujeito a} \ \begin{cases} y_i - \mathbf{w}.\mathbf{x}_i - b \leq \varepsilon_i \\ \mathbf{w}.\mathbf{x}_i + b - y_i \leq \varepsilon_i \end{cases} . \tag{37}$$

Busca-se então a função linear que aproxima os pares de treinamento $(\mathbf{x}_i,\ y_i)$ com uma precisão de ε. A minimização da Equação (36) corresponde a obter o tubo-ε o mais estreito possível. Esse tubo é uma região ao redor da função de regressão onde os erros são ignorados se forem menores que ε. Essas restrições garantem que a função de regressão $f(x) = \mathbf{w}\cdot\mathbf{x} + b$ esteja dentro de uma faixa de ε em relação aos valores reais y_i.

Além disso, problema de minimização da Equação (36) pode ser relaxado introduzindo variáveis de folga ξ_i, permitindo assim lidar com ruídos e $outliers$ de forma que alguns exemplos fiquem fora da região entre $-\varepsilon$ e $+\varepsilon$ com uma certa penalidade. A função ε-insensível é apresentada na Equação (38) e a Figura 25 ilustra graficamente o problema (Schlkopf et al., 2018):

$$|\xi|_\varepsilon = \begin{cases} 0\ , & \textit{se}\ |\xi| \leq \varepsilon \\ |\xi| - \varepsilon\ , & c.c. \end{cases} . \tag{38}$$

Sendo assim, ε-SVR procura-se então encontrar uma função que produza saídas

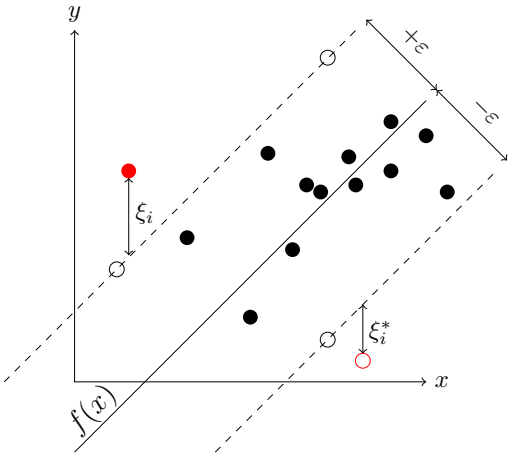


Figura 25 – ε-função de perda insensível.

contínuas para todos os dados de treinamento que se desviem de um máximo de ε de seu rótulo desejado e, ao mesmo tempo, permita que tubo-ε para ser o mais estreito possível (Mustafa et al., 2018).

Essas restrições definem um tubo com um raio ε ao redor da função de regressão hipotética, de modo que se um ponto for posicionado no tubo, a função perda será igual a zero, enquanto se um ponto for posicionado fora do tubo, a perda será proporcional à distância euclidiana entre o ponto e o raio ε do tubo (Mirjalili et al., 2020).

A Equação (39) apresenta a introdução das variáveis de folga e as restrições são apresentadas na Equação (40). A variável de folga ξ_i está associada aos dados localizados abaixo da margem inferior, enquanto ξ_i^* aos localizados acima da margem superior (Schlkopf et al., 2018):

$$\underset{\mathbf{w},b,\xi_i,\xi_i^*}{\textit{Minimizar}} \ \frac{1}{2}\left\|\mathbf{w}\right\|^2 + C\left(\sum_{i=1}^{n}(\xi_i + \xi_i^*)\right), \tag{39}$$

$$\textit{sujeito a} \begin{cases} y_i - \mathbf{w}.\mathbf{x}_i - b \leq \varepsilon_i + \xi_i \\ \mathbf{w}.\mathbf{x}_i + b - y_i \leq \varepsilon_i + \xi_i^* \\ \xi_i,\ \xi_i^* \geq 0 \end{cases}, \tag{40}$$

onde C é a constante de regularização e determina a quantidade de erros de treinamento que serão permitidos e o custo para se obter uma região estreita para tubo-ε.

Os pontos fora do tubo-ε da Figura 25 são chamados de vetores de suporte, por estabelecerem os fundamentos para a função de regressão estimada. Isso significa que todos os outros pontos não estão incluídos no modelo e podem ser removidos após a

construção do modelo SVR. Assim, normalmente, muito menos amostras de treinamento constituem o modelo de regressão, e por este motivo a solução pode ser definida como esparsa (Mirjalili et al., 2020).

Quando uma modelagem não linear é necessário para se adequar aos dados, precisa-se extrapolar o ε-SVR linear para regressão não linear (Abuella and Chowdhury, 2017). A utilização de uma função *kernel* possibilita que os dados de entrada originais sejam implicitamente mapeados no espaço de características de alta dimensão, onde pode ser utilizada uma regressão linear (Arora et al., 2018).

Através da técnica de multiplicadores de Lagrange (Schlkopf et al., 2018), pode-se obter o problema dual apresentado na Equação (41), descrito em termos de produtos internos entre objetos, com as restrições apresentadas na Equação (42). A função objetivo é apresentada na Equação (43):

$$\underset{\alpha,\alpha^*}{\textit{Minimizar}} \ -\frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_i^*)K(x_i,x_j)-\varepsilon\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)+\sum_{i=1}^{n}y_i(\alpha_i-\alpha_i^*), \quad (41)$$

$$\textit{sujeito a} \begin{cases} \sum_{i=1}^{n}(\alpha_i-\alpha_i^*)=0 \\ \alpha_i,\ \alpha_i^* \in [0,C] \end{cases}, \quad (42)$$

$$f(x)=\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)K(x_i,x_j)+b, \quad (43)$$

onde termos α_i e α_i^* representam as variáveis de Lagrange e K é a função *kernel*.

Um *kernel* K é uma função que recebe dois pontos (x_i e x_j) no espaço de entrada original e calcula o produto escalar desses objetos em um espaço de alta dimensionalidade. Desta forma, uma função de *kernel* pode conseguir transformar um conjunto de dados inicialmente não linearmente separável em outro, de forma que estes dados se toram linearmente separáveis (Schlkopf et al., 2018).

As funções de *kernel* mais usadas são lineares, polinomial, função de base radial (RBF) e sigmoidal (Schlkopf et al., 2018). Cada uma dessas funções possui parâmetros que devem ser determinados durante a etapa de ajuste dos hiperparâmetros. Nesta tese, utilizou-se o *kernel* polinomial de ordem 1, cujos parâmetros estão descritos na Equação (44):

$$K(x_i, x_j) = [\delta(x_i, x_j) + \kappa]^d, \tag{44}$$

onde δ, κ e d são os hiperparâmetros da função *kernel* K.

A generalização do modelo SVR depende da seleção correta dos parâmetros C e ε e do tipo de função *kernel* e seus parâmetros. O parâmetro ε é responsável por definir o raio do tubo ε ao redor da função de regressão e, portanto, o número de vetores de suporte selecionados para construir a função de regressão. O valor de ε está relacionado à amplitude do ruído presente no conjunto de treinamento e, como esta informação é normalmente desconhecida, ε pode variar em um intervalo proposto pelo usuário (Schlkopf et al., 2018).

Já o parâmetro C determina a penalidade para os casos em que a função de regressão aceita objetos com desvios maiores que o valor de ε (Xu et al., 2019). Assim, a robustez do modelo de regressão também depende da escolha do valor de C, que pode assumir valores entre 0 e ∞. Quando C é muito alto, o algoritmo é mais flexível, mas há uma alta probabilidade de um ajuste excessivo no conjunto de treinamento, também chamado de *overfitting* (Mathew et al., 2017). Isso significa que o algoritmo terá um desempenho ruim na generalização. No entanto, um valor baixo de C pode fazer com que o algoritmo faça previsões ruins devido à falta de flexibilidade, conhecido como de *underfitting* (Deiss et al., 2020).

3.6.4 Processo Gaussiano

Em teoria da probabilidade e estatística, um processo gaussiano é um modelo estatístico em que as observações ocorrem em um domínio contínuo, onde cada ponto em algum espaço de entrada contínua está associada com uma variável aleatória com distribuição normal. A regressão por processo gaussiano (GPR) é uma técnica que consiste em estimar uma função para entradas arbitrárias, dado um conjunto a priori de treinamento, para a qual os valores interpolados são modelados por um processo gaussiano governado por covariâncias prévias (Rasmussen and Williams, 2005).

Assume-se que para um processo gaussiano f_g observado nas coordenadas x, o vetor de valores $f_g(x)$ é apenas uma amostra de uma distribuição gaussiana multivariada

de dimensão igual ao número de coordenadas observadas n. Assumindo uma distribuição de média zero $f_g(x) \sim N(0, K(\theta, x, x'))$, onde $K(\theta, x, x')$ é a matriz de covariância entre todos os pares possíveis (x, x') para um determinado conjunto de hiperparâmetros θ (Rasmussen and Williams, 2005), a verossimilhança marginal logarítmica é:

$$\log p(f_g(x) \mid \theta, x) = -\frac{1}{2} f_g(x)^T K(\theta, x, x')^{-1} f_g(x') - \frac{1}{2} \log \det(K(\theta, x, x')) - \frac{n}{2} \log 2\, pi, \tag{45}$$

onde $K(\theta, x^*, x)$ é a covariância entre a nova coordenada de estimativa x^* e todas as outras coordenadas observadas x para um determinado vetor de hiperparâmetros θ, $K(\theta, x^*, x^*)$ é a variação no ponto x^* conforme ditado por θ. É importante notar que praticamente a estimativa da média posterior $f_g(x^*)$ (estimativa pontual) é uma combinação linear das observações $f_g(x)$, assim como a variância de $f_g(x^*)$ é independente das observações $f_g(x)$.

Dentre as diferentes funções de covariância que podem ser utilizadas no processo gaussiano destaca-se a covariância de Matérn, comumente usada para definir a covariância estatística entre medições feitas em dois pontos distantes. Desta forma, a covariância de Matérn é estacionária (Rasmussen and Williams, 2005) e, caso a distância for uma distância euclidiana, esta covariância também é isotrópica. A distância de covariância de Matérn pode ser escrita como:

$$C_\nu(d) = \sigma_G^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{d}{\rho} \right)^\nu K_\nu \left(\sqrt{2\nu} \frac{d}{\rho} \right), \tag{46}$$

onde Γ é a função gama, K_ν é a função de Bessel modificada de segundo tipo de ordem ν. O parâmetro ρ determina a taxa com que a autocorrelação espacial diminui à medida que aumenta a distância de separação d.

Considerando a ordem $\nu = p + 1/2,\ p \in \mathbb{N}^+$, a covariância de Matérn pode ser escrita como um produto entre uma exponencial e um polinômio de ordem p (Abramowitz and Stegun, 2012):

$$C_{p+1/2}(d) = \sigma_G^2 \exp\left(-\frac{\sqrt{2p+1}d}{\rho} \right) \frac{p!}{(2p)!} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{2\sqrt{2p+1}d}{\rho} \right)^{p-i}, \tag{47}$$

tal que os parâmetros ρ (escala do kernel) e σ_G (desvio padrão do sinal) são ajustados

durante a etapa de treinamento.

3.6.5 Métricas de regressão

Seja P uma variável discreta, tal que correspondência ponto a ponto entre a solução numérica e as medidas observadas da mesma variável fornece um teste quantitativo para medir a capacidade (ou destreza) do modelo em reproduzir ou estimar os dados observados (Wang and Lu, 2018). Sejam P_{is} e P_{io} respectivamente, a severidade da falha simulada e observada no mesmo ponto no tempo $i = 1, 2, 3, \ldots, N$. A diferença d_i entre os valores simulados e observados no ponto i para as variáveis de entrada x e saída y no tempo t, é dada por:

$$d_i = e_i = P_{is} - P_{io}, \tag{48}$$

onde $P_{is} = P(x, y, t)$ é a severidade simulada e $P_{io} = O(x, y, t)$ a severidade observada para mesma variável no mesmo ponto x e y e no tempo.

Sendo assim, $d_i = 0$ indica uma simulação exata no ponto i, enquanto $d_i >> 0$ ou $d_i << 0$ indica uma simulação não exata. Quanto mais longe de zero estiver o valor de d_i, mais imprecisa é a simulação, ou seja, d_i equivale ao erro e_i de uma determinada análise. A partir do valor de d_i, é possível derivar erros que refletem diferentes componentes do erro total (Willmott and Matsuura, 2006).

A métrica de análise dos resultados adotada neste trabalho e derivada de d_i é a Raiz do Erro Quadrático Médio (RMSE), que aponta para a tendência ou, em termos mais formais, a variância. Em termos sucintos, a variância aponta a tendência do modelo em superestimar ou subestimar a severidade da falha em relação ao que foi observado. Tal métrica é comumente utilizada para expressar a precisão dos resultados numéricos apresentando valores de erro nas mesmas dimensões da variável analisada (Wang and Lu, 2018). O RMSE é definido pela equação abaixo:

$$\text{RMSE} = [\frac{1}{N}\sum_{i=1}^{N}(P_{is} - P_{io})^2]^{\frac{1}{2}}, \tag{49}$$

sendo assim, seja $h \in \mathcal{H}$ uma hipótese candidata para aproximar a função desconhecida

$f : \mathcal{X} \rightarrow \mathcal{Y}$, e $\mathcal{D} = [(x(1), y(1)), \ldots, (x(N), y(N))]$ a amostra rotulada. Considerando $\mathcal{D} \subset \{\mathcal{X}, \mathcal{Y}\} \mid \mathcal{D} \in \mathbb{R}$, o erro quadrático médio RMSE é:

$$\text{RMSE}(h, \mathcal{D}) = \left[\frac{1}{N} \sum_{i=1}^{N} (h[x(n)] - y(n))^2 \right]^{\frac{1}{2}}. \tag{50}$$

Visto que a hipótese h pode ser vista como uma determinada configuração do regressor, visto o uso da partição k-fold dos dados para treinamento, $h[x(n)]$ é o ajuste da variável de regressão para cada um dos sucessivos k da etapa de validação cruzada. Dessa forma, o valor RMSE após as k implementações será:

$$\mu_{\text{RMSE}} = \frac{1}{k} \sum_{j=1}^{k} \text{RMSE}(j), \tag{51}$$

$$\sigma_{\text{RMSE}} = \left[\frac{1}{k-1} \sum_{j=1}^{k} (\text{RMSE}(j) - \mu_{\text{RMSE}})^2 \right]^{\frac{1}{2}}, \tag{52}$$

onde μ_{RMSE} é o valor médio do RMSE e σ_{RMSE} os desvio padrão do RMSE, indicando a precisão do modelo após a etapa de validação cruzada.

A medida $\mu_{RMSE} \pm \sigma_{RMSE}$ será adotada como métrica para avaliação dos regressores, visando quantificar o desempenho do ajuste das curvas de regressão da variável de severidade, utilizando estatísticas de primeira e segunda ordem.

3.6.6 Coeficiente de correlação de Pearson

Para quantificar o desempenho das curvas de regressão, foi utilizado o coeficiente de correlação de Pearson ρ. Esta métrica mede o grau de correlação linear entre duas variáveis quantitativas. É um índice adimensional com valores entre -1 e 1 inclusive, que reflete a intensidade de uma relação linear entre dois conjuntos de dados. $\rho = 1$ descreve uma correlação positiva perfeita entre as duas variáveis. $\rho = -1$ caracteriza uma correlação negativa perfeita entre as duas variáveis, ou seja, se uma aumenta, a outra sempre diminui. $\rho = 0$ significa que as duas variáveis não dependem linearmente uma da outra. No entanto, pode haver outra dependência que seja “não linear”. Assim, o resultado $\rho = 0$ deve ser investigado por outros meios.

Seja $\mathbf{X_p} = x_1, x_2, \ldots, x_n$ os valores de um conjunto de pontos para uma regressão, com $i = 1, \ldots, n$, e seja $\mathbf{Y_p} = y_1, y_2, \ldots, y_n$ um conjunto de pontos representando uma regressão perfeita. O coeficiente de correlação de Pearson é definido como mostrado na Equação (53),

$$\rho = \frac{cov(\mathbf{X_p}, \mathbf{Y_p})}{\sqrt{var(\mathbf{X_p}) \cdot var(\mathbf{Y_p})}}. \tag{53}$$

3.7- Extração de características

A técnica de extração de características para o aprendizado de máquina é um processo fundamental que visa identificar e selecionar os atributos mais relevantes e informativos de um conjunto de dados (Haykin, 1998). Esse processo é crucial para o desenvolvimento de modelos de aprendizado de máquina eficazes, pois ajuda a reduzir a dimensionalidade dos dados, eliminando características irrelevantes ou redundantes e destacando aquelas que contêm informações importantes para o problema em questão (Haykin, 2009).

Existem várias abordagens para a extração de características, que variam conforme a natureza dos dados e o problema específico a ser resolvido. Uma abordagem comum é a utilização de técnicas estatísticas, como análise de variância ou análise de componentes principais, que ajudam a identificar as características mais relevantes com base em critérios estatísticos (Maier and Dandy, 2001). Outra abordagem amplamente utilizada é a aplicação de algoritmos de seleção de características, que avaliam a importância de cada característica utilizando métricas como ganho de informação, coeficiente de correlação ou importância relativa (Dudek, 2016).

Além disso, técnicas mais avançadas, como redes neurais convolucionais, podem ser empregadas para extrair automaticamente características diretamente dos dados brutos, aprendendo padrões complexos e hierárquicos que podem não ser facilmente identificados por métodos tradicionais (Davydenko and Fildes, 2013). Estas técnicas são especialmente úteis em problemas complexos, onde a identificação manual de características relevantes seria impraticável ou ineficiente, permitindo uma análise mais profunda e abrangente dos dados.

Independentemente da abordagem escolhida, a extração de características desempenha um papel crucial no sucesso de modelos de aprendizado de máquina, pois influencia diretamente a capacidade do modelo de generalizar a partir dos dados de treinamento e fazer previsões precisas em dados não vistos (Gil-Alana, 2005).

Neste trabalho, a técnica para extração de características utilizada, consiste em estimar os valores de pressão média e máxima dos 6 sinais de cilindro de pressão $s_{p_1}(n)$, $s_{p_2}(n)$, $s_{p_3}(n)$, $s_{p_4}(n)$, $s_{p_5}(n)$ e $s_{p_6}(n)$, bem como a obtenção de informações espectrais do sinal de vibração torcional $s_v(n)$, através da análise de meias ordens $\mathbf{f_{1/2}}$, utilizando as 24 meias ordens, com $\mathbf{f_{1/2}} = [20,83 \;\; 41,67 \;\; 62,50 \;\; \cdots \;\; 500]$, relativas a velocidade de rotação adotada (2500 RPM, equivalente a 41,67 Hz). As métricas adotadas associadas às falhas, para discriminá-las, estão conforme a Seção 3.8.

A utilização de 24 meias ordens para a extração de características de um motor Diesel é essencial devido à correlação das frequências harmônicas características dos componentes do motor, permitindo uma resolução mais detalhada das vibrações e dos sinais de ruído. A análise em meias ordens ajuda na detecção de falhas e anomalias, capturando sinais sutis que podem diagnosticar problemas específicos como desalinhamento, desequilíbrio e falhas em componentes. Além disso, oferece uma melhor resolução de detalhes das vibrações e ruídos, permitindo a distinção precisa entre diferentes fontes de ruído e vibração, fundamental para a implementação de estratégias de manutenção preditiva (Abdulaziz and Khan, 2014).

Essa abordagem também está relacionada aos atributos dos ciclos de combustão em motores de 4 tempos, onde cada meia ordem de rotação corresponde a um ciclo de queima em cada cilindro, capturando variações com maior precisão. A análise de meias ordens melhora a capacidade diagnóstica dos algoritmos de aprendizado de máquina e de inteligência artificial utilizados para monitorar o desempenho do motor, contribuindo para uma detecção precoce e precisa de problemas, otimizando a identificação de possíveis falhas no motor Diesel (Zannis and Hountalas, 2005).

3.8- Subconjuntos de características

3.8.1 Estimativa da pressão máxima nos cilindros

É o valor máximo de cada curva de pressão discreta associada a cada cilindro $s_{p_1}(n)$, $s_{p_2}(n)$, $s_{p_3}(n)$, $s_{p_4}(n)$, $s_{p_5}(n)$ e $s_{p_6}(n)$, gerando M_{p_1}, M_{p_2}, M_{p_3}, M_{p_4}, M_{p_5} e M_{p_6}, respectivamente. A Eq. (54) resume esse subconjunto de características.

$$M_{p_i} = \max[s_{p_i}(n)], \tag{54}$$

onde M_{p_i} é o valor máximo das curvas de pressão $s_{p_i}(n)$ para cada cilindro i.

3.8.2 Estimativa da pressão média dentro dos cilindros

É o valor esperado de primeira ordem (média) de cada curva de pressão discreta associada a cada cilindro $s_{p_1}(n)$, $s_{p_2}(n)$, $s_{p_3}(n)$, $s_{p_4}(n)$, $s_{p_5}(n)$ e $s_{p_6}(n)$, gerando μ_{p_1}, μ_{p_2}, μ_{p_3}, μ_{p_4}, μ_{p_5} e μ_{p_6}, respectivamente. A Eq. (55) resume esse subconjunto de características:

$$E[s_{p_i}(n)] = \mu_{p_i} = \frac{1}{N}\sum_{n=1}^{N} s_{p_i}(n), \tag{55}$$

onde, $E[\cdot]$ é o valor esperado, μ, associado a cada curva de pressão do cilindro; N é o número de amostras de $s_{p_i}(n)$.

3.8.3 Análise espectral

A técnica de análise espectral é utilizada para estimar o espectro de frequência de torção semelhante ao descrito em (Pestana-Viana et al., 2016). Consiste em calcular uma

DFT N-ponto de $s_v(n)$ conforme a Eq (56), gerando $S_v(k)$, que ser usado para calcular $F(k)$, $A(k)$ e $P(k)$ representando valores de frequência (Hz), amplitude (N.m) e fase (graus), respectivamente. As Eqs. (57)-(59) resumem este subconjunto de características.

$$S_v(k) = \frac{1}{N_{\mathrm{DFT}}} \sum_{k=0}^{N_{\mathrm{DFT}}-1} s_v(n) W_N^{kn}, \tag{56}$$

onde $S_v(k)$ é o N_{DFT}-ponto DFT de $s_v(n)$ com $W_N^{kn} = e^{-\frac{j2\pi}{N_{\mathrm{DFT}}}}$ e j representando o número complexo.

$$F(k) = \frac{kF_s}{N_{\mathrm{DFT}}}, \tag{57}$$

onde $F(k)$ é a frequência harmônica do espectro de torção $S_v(k)$; k é o *bin* de frequência associado a uma frequência do vetor de meias ordens $\mathbf{f_{1/2}}$ e F_s é a frequência de amostragem de aquisição de dados.

$$A(k) = |S_v(k)|\,, \tag{58}$$

onde $A(k)$ é a amplitude (N.m) do espectro de torção $S_v(k)$.

$$P(k) = \frac{360}{2\pi} \arg[S_v(k)], \tag{59}$$

onde $P(k)$ é a fase (grau) do espectro de torção $S_v(k)$ e $arg[\cdot]$ representa o argumento complexo do espectro, ou seja, o qual é a fase de deslocamento entre a parte real e imaginária da variável complexa.

3.9- Vetor de características

Diferente da técnica apresentada em (de Lima et al., 2013) e (Pestana-Viana et al., 2016), esta proposta opta aplicar a extração de características na vibração torcional e abordar valores médios e máximos em vetores de características, modificando a técnica de extração de características, pois a máquina trata-se de um motor Diesel, sendo este novo vetor de características, mais representativo para o problema em questão.

Para gerar o vetor de características, o primeiro passo é estimar os valores

máximos das curvas de pressão para cada cilindro segundo a Eq. (54) fornecendo o subconjunto de características:

$$M_{p_i} = [M_{p_1}, M_{p_2}, M_{p_3}, \cdots, M_{p_6}], \tag{60}$$

onde p_j é a curva de pressão associada a cada cilindro, com $j = 1, 2, \cdots, 6$.

O segundo passo é obter os valores médios das curvas de pressões dos cilindros, conforme Eq. (55), onde o subconjunto de características é:

$$\mu_{p_i} = [\mu_{p_1}, \mu_{p_2}, \mu_{p_3}, \cdots, \mu_{p_6}]. \tag{61}$$

A terceira etapa é calcular os valores espectrais das curvas de vibração torcional $S_v(k)$ segundo as Eqs. (56)-(59), e seus primeiros 24 harmônicos, ou seja, as primeiras 24 meia-ordens do motor, isto é 6 cilindros multiplicados cada um dos 4 tempos da combustão, totalizando 24 meia-ordens. Considerando a rotação fixada em 2500 RPM, a primeira metade da ordem é dada por $\frac{1}{2}\frac{2500}{60}$ Hz. Consequentemente, o vetor de frequência de 24 meia-ordens é $f = [21,083, 41,667, \cdots, 500, 0]$, onde seus elementos são as frequências das primeiras 24 meia-ordens. A relação entre a raia espectral e a frequência é obtida diretamente de $k = [f]\frac{N_{\mathrm{DFT}}}{F_s}$. O subconjunto relacionado aos valores espectrais é então dado por:

$$F(k) = [F(k_1), F(k_2), F(k_3), \cdots, F(k_{24})], \tag{62}$$

$$A(k) = [A(k_1), A(k_2), A(k_3), \cdots, A(k_{24})], \tag{63}$$

$$P(k) = [P(k_1), P(k_2), P(k_3), \cdots, P(k_{24})], \tag{64}$$

onde k_j é a *bin* de frequência associada a cada elemento do vetor de meia-ordem 24, com $j = 1, 2, \cdots, 24$.

A etapa final consiste em combinar todas as medidas em um vetor de características, V_f, usando as etapas descritas nas Eqs. (60)-(64), concatenando os 3 espectrais variáveis, valores máximos e médios, que podem ser expressos por:

$$
\begin{aligned}
V_f \;=\; & \{M_{p_1},\, M_{p_2},\, M_{p_3},\, \cdots,\, M_{p_6}, \mu_{p_1},\; \mu_{p_2},\; \mu_{p_3},\; \cdots,\; \mu_{p_6}, \\
& F(k_1),\, F(k_2),\, F(k_3),\, \cdots,\, F(k_{24}), A(k_1),\, A(k_2),\, A(k_3),\, \cdots,\, A(k_{24}), \\
& P(k_1),\, P(k_2),\, P(k_3),\, \cdots,\, P(k_{24})\}. \qquad (65)
\end{aligned}
$$

O vetor de características V_f possui dimensão 84: 6 valores máximos (em cada cilindro); 6 valores médios (por cilindro) e valores de frequência, amplitude e fase de cada um dos 24 harmônicos - totalizando 72 elementos, sendo este usado para o treinamento e as etapas de teste dos algoritmos de aprendizado de máquina discutidos na próxima seção. Os respectivos valores da distribuição do banco de dados 3500-DEFault dos diversos subconjuntos do vetor V_f, composto por M_{p_i}, μ_{p_i}, $F(k)$, $A(k)$ e $P(k)$, estão organizados em boxplot e mostrados na Figura 26.

3.10- Conclusão

O conjunto de dados utilizado para análise das falhas em motores Diesel abrange diversas condições operacionais, incluindo cenários normais e anormais. Entre as falhas consideradas, destacam-se a redução de pressão no coletor de admissão, a redução da razão de compressão nos cilindros e a redução da quantidade de combustível injetado nos cilindros. Essas falhas podem resultar de obstruções no sistema de admissão de ar, desgaste nos anéis de vedação do pistão, vazamentos nas válvulas ou problemas no sistema de injeção de combustível. O entendimento detalhado dessas condições anormais é essencial para o desenvolvimento de estratégias de manutenção preditiva e diagnóstico precoce.

A base de dados utilizada compreende diversas condições operacionais e severidades de falhas, permitindo a construção de modelos preditivos precisos por meio de técnicas de regressão e aprendizado de máquinas. Essas técnicas oferecem uma abordagem sistemática e eficiente para analisar grandes conjuntos de dados, identificar padrões complexos e relações não lineares entre variáveis, e construir modelos preditivos capazes de detectar sinais sutis de falha e prever seu desenvolvimento. A aplicação

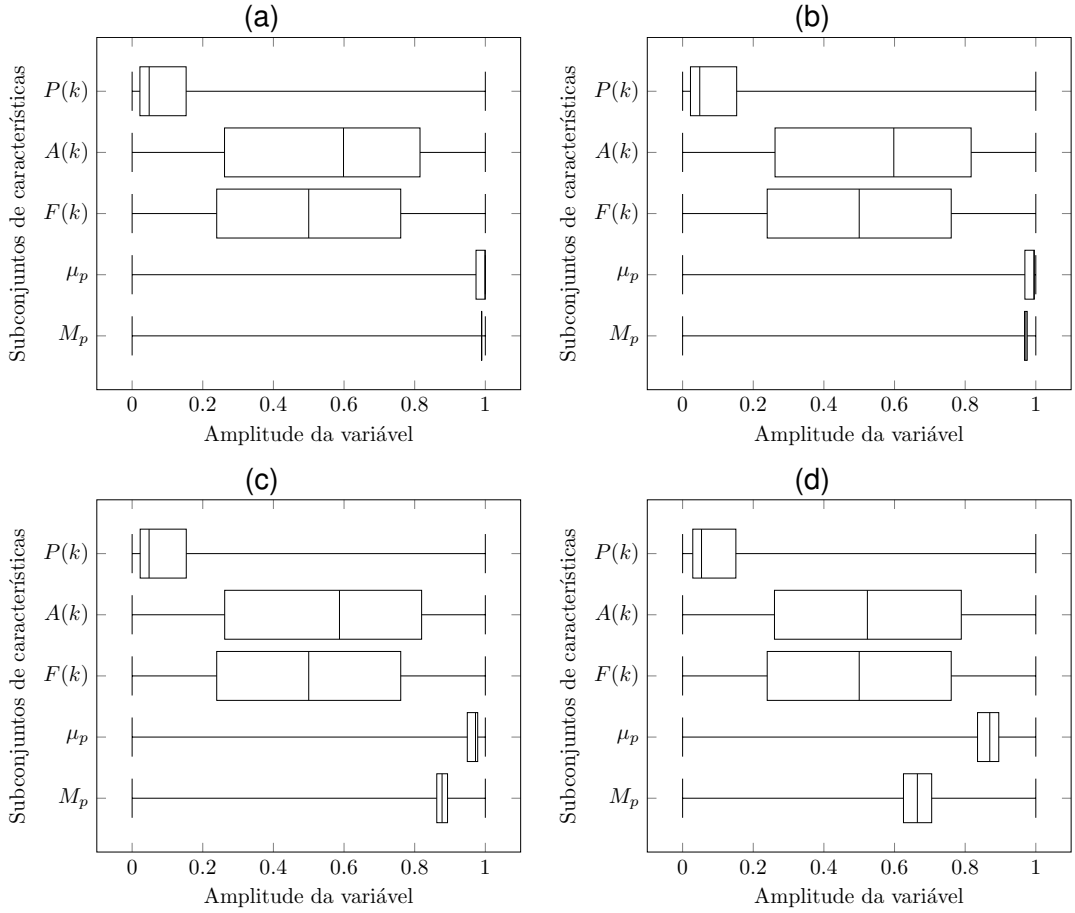


Figura 26 – Box plot do banco de dados 3500-DEFault para subconjuntos de características $F(k)$, $A(k)$, $P(k)$, μ_{p_i} e M_{p_i}, respectivamente. Para diferentes níveis de AWGN: (a) 60, (b) 30, (c) 15 e (d) 0 dB. O gráfico mostra a mediana, o quartil 25%, o quartil 75% e a faixa inferior e superior (bigodes), os quais são o máximo e o mínimo para cada distribuição, respectivamente.

dessas técnicas promove uma operação mais confiável e eficiente dos motores Diesel em diversas aplicações industriais e comerciais.

As seções seguintes implementam o programa de simulação de falhas e exibem os resultados de regressão para os cenários de falhas estudados.

4- Programa DEFault

Este capítulo tem como objetivo principal a exposição detalhada do código subjacente ao programa de simulação de falhas em máquinas a diesel (DEFault). Serão apresentadas todas as rotinas e subfunções que compõem o programa.

Após os capítulos anteriores, que estabeleceram a fundamentação teórica dos modelos termodinâmico e dinâmico: o primeiro objetivando a resolução das equações termodinâmicas aplicadas ao modelo 0-D, para obter as curvas de torque e pressão de um sistema pistão-biela-manivela; o segundo, resolvendo o modelo massa-mola-amortecedor do motor, com 11 graus de liberdade, para obter a função de vibração torcional no volante do motor. Portanto, este capítulo implementa a base teórica desenvolvida, emulando as falhas e sua severidade associada na máquina Diesel. A Figura 28 ilustra o arranjo em diagrama de blocos das etapas do programa DEFault.

O sistema proposto apresenta uma arquitetura modular, semelhante às descritas em (de Lima et al., 2013) e (Pestana-Viana et al., 2016), para um sistema de manutenção condicional composto por doze blocos, conforme ilustrado na Figura 27.

O sistema consiste no bloco **início**, onde é feita a entrada da severidade; a simulação do funcionamento do motor Diesel, presente nos blocos de **modelo dinâmico** e **termodinâmico**, que resolve as equações deste modelo, com base nas severidades aplicadas no bloco de partida e nos sinais de falha retornados.

O bloco de geração do conjunto de dados, que inclui tanto sinais normais quanto de falha, e trata da relação entre as instâncias normais e exemplos de falha; o bloco de processo *Additive white Gaussian noise* (AWGN) - **Processo AWGN**, que adiciona ruído branco aos sinais; o bloco de sinais do conjunto de dados, que armazena o conjunto de dados segundo os níveis de ruído.

O bloco de **extração de características**, que extrai atributos do conjunto de dados para uso na regressão; o bloco de conjunto de dados de características; o bloco de validação cruzada, responsável por avaliar a capacidade de generalização do modelo; o bloco de regressão, composto por regressores baseados em Aprendizado de Máquina, que realiza os testes de aprendizado; e o bloco Fim, que apresenta os resultados. Nesta tese, o acrônimo 3500-DEFault (Diesel Engine Fault) se refere ao conjunto de 3500 sinais

de uma simulação de motor Diesel.

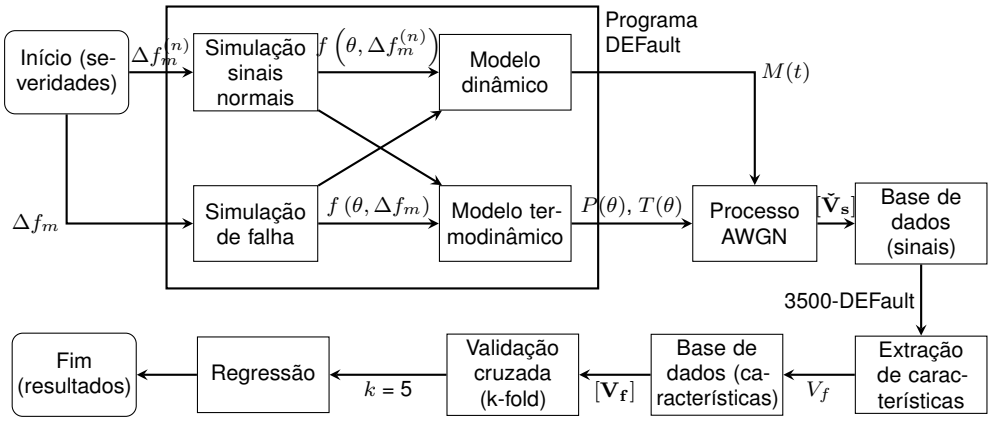


Figura 27 – Diagrama de blocos do sistema proposto composto pelo bloco do modelo dinâmico, o bloco de geração do banco de dados (dividido em falha e criação de sinais normais), o bloco do processo AWGN, o bloco de extração de características e o bloco de regressão.

O diagrama de blocos apresentado esquematiza de forma clara e organizada o fluxo de informações e processos envolvidos no programa DEFault, utilizado para simulação de falhas em máquinas a Diesel. Dividido em duas fases principais, o modelo termodinâmico e o modelo dinâmico.

O programa inicia com a entrada de variáveis relevantes para a simulação. No modelo termodinâmico, a pressão no interior dos cilindros é calculada a partir das variáveis de entrada, seguida pelo cálculo do torque e pela replicação das voltas do motor. Por sua vez, o modelo dinâmico inclui etapas como o cálculo da diferença de volume do motor, a definição dos parâmetros modais e a determinação da vibração torcional no volante do motor, utilizando técnicas como o algoritmo de Runge-Kutta para resolver equações diferenciais.

A estrutura do diagrama é complementada por elementos visuais que destacam a organização do processo. Caixas delimitadoras agrupam os blocos relacionados sob os títulos “Modelo Termodinâmico” e “Modelo Dinâmico”, evidenciando as diferentes fases do programa. As setas direcionais indicam o fluxo de informações entre os blocos, demonstrando como os resultados de uma etapa são utilizados como entrada para a próxima. Com isso, o diagrama não apenas fornece uma representação visual clara do funcionamento do programa DEFault, mas também facilita a compreensão do fluxo de informações e processos envolvidos na simulação de falhas em máquinas a Diesel, aspecto crucial para o entendimento do contexto de pesquisa abordado nesta tese.

4.1- Programa DEFault - Gerador de Sinais de Falha de Motor Diesel

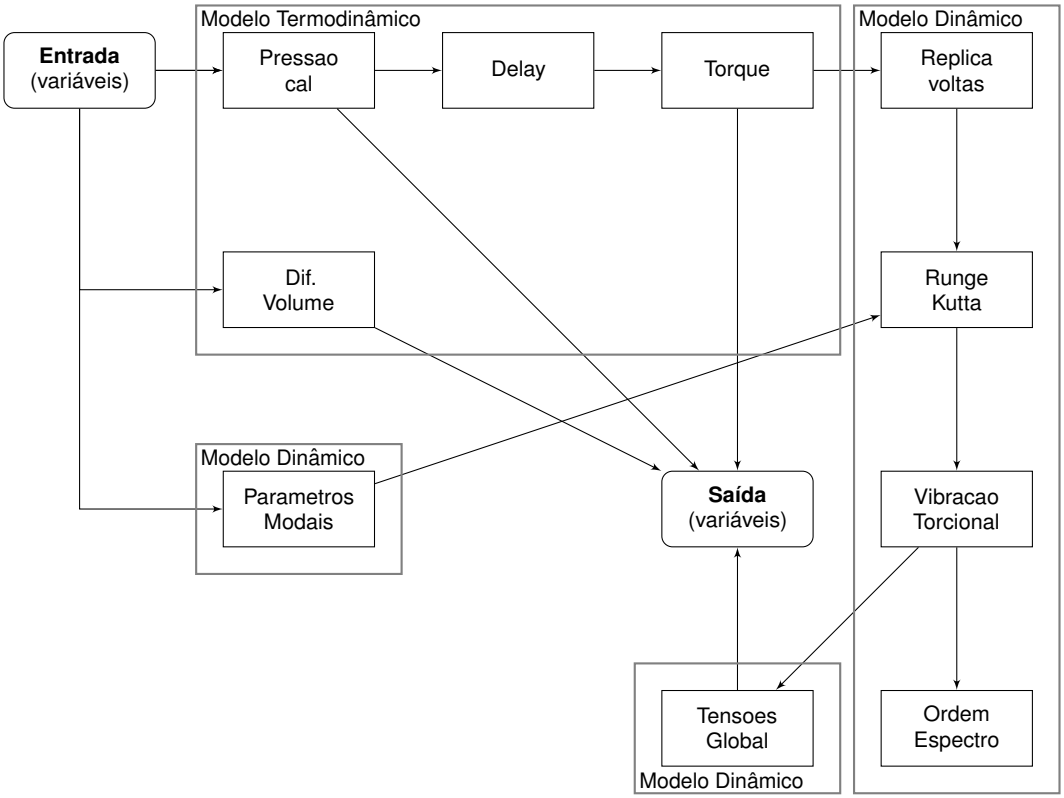


Figura 28 – Diagrama de blocos do programa DEFault.

O Algoritmo 1 é um trecho de código de um modelo de simulação de um motor de combustão interna, escrito em MATLAB®. O código cria diretórios para salvar os resultados da simulação e carrega informações sobre as condições de operação do motor, como a queda de pressão no coletor de admissão, a variação da pressão no *common rail* e a variação da temperatura da água de arrefecimento.

Os detalhes completos dos algoritmos, subfunções e rotinas do programa DEFault, abordados na Figura 28 e no Algoritmo 1, encontram-se na seção de apêndices. Isso permite fornecer uma visão clara e detalhada de sua implementação sem interromper o fluxo principal do texto.

Além disso, facilita a análise mais aprofundada, concentrando todas as informações técnicas relevantes em um só lugar, o que simplifica a compreensão e a replicação dos métodos apresentados. Os algoritmos não estão em nenhuma linguagem de programação específica, o que facilitará a implementação em qualquer linguagem, permitindo uma

maior flexibilidade e adaptação conforme a necessidade de trabalhos futuros.

Em seguida, o código calcula as falhas em cada cilindro do motor, como a falha de compressão, a falha na massa de ar admitida, a falha na massa de combustível injetado, o atraso no início da injeção e a falha estrutural devido a trincas.

O modelo termodinâmico 0-D do motor é então resolvido para cada cilindro, utilizando a função $\mathbf{pressao.cal}$, que considera as condições de operação do motor e as falhas em cada cilindro.

Este trecho de código é apenas uma parte de um modelo de simulação mais complexo e pode ser utilizado para testar diferentes condições de operação do motor e diferentes cenários de falha em cada cilindro. O objetivo final é fornecer informações sobre o desempenho e a confiabilidade do motor em diferentes condições de operação, permitindo a otimização do projeto e a redução dos custos de manutenção e reparação.

Algoritmo 1 – PROGRAMA DEFAULT

Entrada: $kIndices$, $SeveridadesTable$, Rot, $dirBase$
Saída: V, Vd, PT_j, Tg_j, Tv

início
  $perc = 0$;
  Inicialize **barra.progresso**$(perc)$;
  $SeveridadesTable = [DP, DPr, DTp, fcomp_j, far_j, fcomb_j, finj_j]$;
  para cada i **em** $kIndices_i$ **faça**
    $tamBase = max(kIndices_i)$;
    Crie o diretório i em $dirBase$;
    $dirSave = dirBase + i$;
    $S = SeveridadesTable(i, :)$;
    $taf = [202, 508, 552]$;
    $PT_j =$ **pressao.cal**$(Rot, taf, S(i, j)$;
    $[V, Vd] =$ **dif.volume**(n);
    $ord = [360, 600, 120, 480, 240, 0]$;
    $PTF_j =$ **delay**(PT_j, ord_j);
    $Tg_j =$ **torque**(Rot, PT_j, ord_j);
    $Td =$ **replica.voltas**(Rot, T);
    $[P, J, A] =$ **parametros.modais**$(Rot, ftrinc_j)$;
    $rk =$ **runge.kutta**(n, Td, A, J);
    $[Tq, stv] =$ **vibracao.torcional**(rk);
    $F_{ord} =$ **orden.espectro**(Rot, stv);
    $[f, P3, Tv] =$ **tensoes.global**(Rot, stv);
    $Pmi_j =$ **pressao.media**(n, PT_j, V, Vd, taf);
    $Pmax_j = max(PT_j) \times 10^{-5}$;
    $Parametros = [V, Vd, PT_j, Tg_j, Tv]$;
    Salvar $Parametros$ em $dirSave$;
    $perc = 100 \times i \times tamBase^{-1}$;
    Atualizar **barra.progresso**$(perc)$;
  fim
fim

Este algoritmo recebe quatro parâmetros de entrada: índice do cenário de falha a ser analizado $kIndices$, Tabela com matriz de severidades $SeveridadesTable$, rotação nominal da máquina Rot e diretório a ser salvo os sinais de saída $dirBase$. A saída do algoritmo é composta por cinco variáveis: volume V, Vd, PTj, Torque de saída Tgj e sinal de vibração torcional Tv.

A primeira linha do algoritmo inicializa a variável percentual de progresso $perc$ com o valor zero. Em seguida, a função de barra de progresso $\mathbf{barra.progresso}$ é inicializada com esse valor. Em seguida, na linha 4, variável $SeveridadesTable$ é definida como uma matriz que contém sete valores por linha.

A partir da linha 5, o algoritmo entra em *loop For* para executar sequencialmente os cálculos em cada elemento do vetor $kIndices$, cujo tamanho máximo é definido na linha 6 por $tamBase$. Em seguida, um diretório é criado para cada elemento i em $kIndices$ e salvo na variável $dirSave$.

Na linha 9, define-se a matriz S para a linha i da matriz $SeveridadesTable$. Em seguida, o vetor taf é definido com três valores. Na linha 11, o valor da pressão é calculado com base em Rot, taf e o valor correspondente de S. As variáveis V e Vd são calculadas na linha 12 usando a função $\mathbf{dif.volume}$. Na linha 13, a matriz ord é definida com seis valores.

A função $\mathbf{delay}$ é usada na linha 14 para calcular $PTFj$ com base em PTj e $ordj$. Logo após, função $\mathbf{torque}$ calcula Tgj com base em Rot, PTj e $ordj$. Na linha 16, a função $\mathbf{replica.voltas}$ é usada para calcular Td.

Em seguida, a função $\mathbf{parametros.modais}$ é usada para calcular as variáveis P, J e A. A função $\mathbf{runge.kutta}$ é usada na linha 18 para calcular rk com base em n, Td, A e J e, na linha 19, a função $\mathbf{vibracao.torcional}$ calcula Tq e stv com base em rk.

A função $\mathbf{orden.espectro}$ é usada na linha 20 para calcular $Ford$ com base em Rot e stv. Na linha 21, a função $\mathbf{tensoes.global}$ é usada para calcular f, $P3$ e Tv com base em Rot e stv e, na linha 22, A função $\mathbf{pressao.media}$ calcula $Pmij$ com base em n, PTj, V, Vd e taf.

O valor máximo de PTj é calculado na linha 23 e armazenado em $Pmaxj$. Na linha 24, a matriz $Parametros$ é definida como um vetor contendo V, Vd, PTj, Tgj e Tv, sendo estes parâmetros então salvos a seguir no diretório $dirSave$.

Por fim, na linha 26, a variável $perc$ é atualizada com o progresso atual da execução.

4.2- Algoritmos do modelo Termodinâmico

4.2.1 Pressao.cal

Tem como o objetivo o cálculo da pressão ao longo do tempo no funcionamento de um motor Diesel. As entradas incluem o número de cilindros N, a temperatura ambiente taf, a pressão de admissão DP, a pressão residual DPr, a temperatura do processo DTp, e uma série de fatores que influenciam a composição e a combustão $fcomp$, far, $fcomb$ e $finj$. Esses valores são essenciais para determinar o comportamento inicial e as condições de operação do motor.

Em seguida, o algoritmo inicializa uma matriz de parâmetros com valores predefinidos que correspondem a diferentes condições de operação do motor Diesel. Esta matriz contém informações como a velocidade do motor, a pressão máxima, a temperatura, a duração da combustão, e a eficiência da combustão para várias configurações possíveis do motor. Além disso, o algoritmo define alguns parâmetros iniciais, como o tempo inicial $t0$, o tempo final tf, o intervalo de tempo dt, e o número de passos de tempo n. Esses parâmetros determinam o período de simulação e a resolução do cálculo.

O algoritmo então calcula as condições iniciais do motor com base nos parâmetros fornecidos e na matriz de parâmetros predefinidos. Ele verifica o número de cilindros N contra a matriz de parâmetros para encontrar a linha correspondente e extrair os valores apropriados. Com esses valores, ele calcula a condição inicial da pressão, ajustando-a com base na pressão de admissão DP fornecida.

Depois de estabelecer as condições iniciais, o algoritmo procede para a integração numérica das equações diferenciais que descrevem o comportamento da pressão no motor Diesel ao longo do tempo. Ele utiliza o método de Runge-Kutta de quarta ordem para realizar esta integração. Este método é escolhido por sua precisão e estabilidade em resolver equações diferenciais.

O algoritmo então executa um loop ao longo dos passos de tempo definidos. A cada passo, ele calcula uma estimativa da taxa de variação da pressão e de outras variáveis relevantes usando o método de Runge-Kutta. Este método envolve calcular várias aproximações intermediárias $k1$, $k2$, $k3$, $k4$ e combinar essas aproximações para

obter uma estimativa precisa do novo estado do sistema.

Durante o processo de integração, o algoritmo armazena os resultados em uma matriz que registra o tempo e os valores da pressão a cada passo de tempo. Esta matriz, chamada sw, contém uma linha para cada passo de tempo, permitindo que se observe como a pressão e outras variáveis evoluem ao longo do período de simulação.

Finalmente, o algoritmo retorna a matriz que contém o tempo e os valores da pressão ao longo do tempo. Esta matriz permite analisar o comportamento do motor Diesel durante a simulação, oferecendo uma visão detalhada de como a pressão varia com o tempo em resposta às condições iniciais e aos parâmetros de operação.

Este algoritmo, portanto, simula o comportamento da pressão ao longo do tempo no funcionamento de um motor Diesel, utilizando métodos numéricos para integrar as equações diferenciais que governam o sistema. Ao final, fornece uma visão detalhada da dinâmica da pressão no motor, essencial para entender e otimizar seu desempenho.

4.2.2 Delay

Realiza uma operação de atraso em um sinal de pressão com base em um valor específico de tempo. A função recebe como entrada uma matriz com duas colunas, uma representando o tempo e a outra representando a pressão.

A função começa separando as duas colunas da matriz: uma coluna para os tempos e outra para as pressões. Em seguida, a função procura a posição da primeira ocorrência do valor de tempo fornecido na coluna de tempos.

Depois disso, a coluna de pressões é dividida em duas partes: a primeira parte contém as pressões desde o início até a posição encontrada anteriormente, e a segunda parte contém as pressões a partir da posição seguinte até o final. Essas duas partes são então concatenadas, colocando a segunda parte antes da primeira, criando uma nova sequência de pressões.

Por fim, a função combina a coluna original de tempos com a nova sequência de pressões reorganizada. O resultado é uma nova matriz que mantém a coluna de tempos original, mas com a coluna de pressões ajustada para refletir o atraso baseado no valor de tempo fornecido. A função retorna essa nova matriz como resultado.

4.2.3 Torque

Este algoritmo calcula o torque de um cilindro em um motor de combustão interna ao longo de um ciclo de combustão. Ele recebe como entrada o número de rotações por minuto (RPM), a pressão no cilindro P e a ordem do ciclo. O torque é calculado para cada ponto no ciclo de combustão e armazenado em um vetor. O atraso é então aplicado ao vetor de torque de acordo com a ordem especificada.

O algoritmo descreve uma função que calcula o torque em relação ao tempo para um conjunto específico de parâmetros. Esta função recebe como entrada o número de rotações, um vetor representando as pressões e a ordem de atraso. O resultado é um vetor contendo os valores de torque ao longo do tempo. Inicialmente, são inicializadas algumas constantes necessárias para os cálculos. Um vetor vazio é criado para armazenar os valores de tempo e torque. A aceleração angular é calculada com base no ângulo da manivela. O torque resultante é calculado com base na massa, na aceleração angular e em um fator de raio. O torque é então calculado para cada ponto no tempo.

Os valores de tempo e torque calculados são atrasados de acordo com a ordem de combustão especificada. O vetor final é atribuído com o resultado do atraso. Em resumo, a função realiza cálculos iterativos para determinar o torque em relação ao tempo com base nos parâmetros fornecidos. O resultado é um vetor de torque que pode ser utilizado para análises e simulações relacionadas ao torque de um motor.

4.2.4 Dif.volume

Este algoritmo implementa uma função destinada a calcular o volume de um cilindro de combustão interna de maneira incremental, com base na posição angular do pistão. Inicialmente, são definidas as constantes relacionadas às dimensões do cilindro e à cinemática da biela. Em seguida, itera-se sobre cada posição do pistão, calculando o volume incremental do cilindro usando equações que envolvem o raio da manivela, o comprimento da biela, o diâmetro do cilindro e a razão de compressão. Os volumes incrementais são armazenados e somados para obter o volume total do cilindro. Além

disso, o algoritmo determina o volume de deslocamento do cilindro e retorna ambos os valores calculados. Em resumo, essa função fornece uma abordagem eficiente para calcular o volume de um cilindro de combustão interna em diferentes posições do pistão, considerando suas dimensões e cinemática.

4.2.5 Parametros.modais

Este algoritmo é um procedimento computacional para calcular os parâmetros modais de um sistema mecânico, fundamentado em informações sobre rigidez, inércia e amortecimento. Primeiramente, são definidas as curvas de variação da rigidez do sistema, seguidas pela geração das matrizes de inércia, rigidez e amortecimento. Com base nessas matrizes, é construída a matriz de estado do sistema. Em seguida, são calculados os autovalores dessa matriz, fornecendo as frequências naturais e os fatores de amortecimento do sistema. Finalmente, os parâmetros modais, como período e razão de amortecimento, são derivados a partir dos autovalores calculados. Esses parâmetros são cruciais para entender e projetar estruturas sujeitas a vibrações, fornecendo *insights* sobre seu comportamento dinâmico.

4.3- Algoritmos do modelo dinâmico

4.3.1 Replica.voltas

O pseudocódigo descreve a função que replica voltas em uma matriz de torques. Essa função recebe um número inteiro e uma matriz de torques como entrada, retornando uma matriz com as voltas replicadas, em função da ordem de queima do motor. Em seguida, extrai os tempos e os torques da matriz de entrada, preenchendo uma matriz com os torques originais. Um *loop* replica as voltas adicionando as linhas dos torques em posições específicas na matriz, obedecendo a ordem de ignição dos cilindros. Após isso,

calcula o tamanho de uma matriz e gera os valores de tempo com base no primeiro valor e numa diferença calculada do segundo valor dos tempos originais, armazenando-os em uma matriz. Por fim, cria uma matriz combinando os tempos replicados e os torques replicados. Em resumo, a função replica voltas na matriz de torques, mapeando os torques para cada instante de tempo correspondente, considerando o número de voltas desejado e a ordem de ignição. O resultado é uma matriz com os tempos replicados seguidos pelos torques replicados para os seis cilindros do motor.

4.3.2 Runge.kutta

O algoritmo apresentado é uma implementação do método Runge-Kutta para resolver equações diferenciais. Recebendo como entrada o número de iterações N, a função $f(t, x)$, e a condição inicial x_0, o algoritmo retorna o vetor de solução x obtido por esse método. Para um sistema com N graus de liberdade, dado um vetor de forças externas F e condições iniciais x_0, podemos empregar o método de Runge-Kutta de ordem 4 para resolver a equação de movimento. O algoritmo começa definindo o passo de tempo dt e o tempo final t_f, inicializa os vetores de tempo t e solução x, e calcula o número de iterações n. Em um loop de $i = 1$ até N, o algoritmo calcula as derivadas em diferentes pontos, utilizando uma média ponderada dessas derivadas para atualizar o valor da solução. Ao final do loop, retorna o vetor de solução x, fornecendo assim uma solução numérica para a equação diferencial.

A rotina $\mathbf{interp1q}$ teve um papel fundamental neste estudo, ao modificar a subfunção $\mathbf{vibracao}$ de $\mathbf{runge.kutta}$, contribuindo significativamente para a otimização do tempo de processamento dos algoritmos empregados na resolução do modelo diesel. Essa melhoria foi crucial para a solução dos sinais de vibração torcional, com tempo significativamente menor, dos sinais necessários à construção da base de dados, permitindo a solução das equações com um ganho de tempo de 60 vezes, em relação ao trabalho de (Gutiérrez, 2016). Ao oferecer uma interpolação eficiente e rápida dos dados, a função $\mathbf{interp1q}$ facilitou não apenas a convergência das soluções, mas também a possibilidade da geração de 3500 cenários de falha. Essa eficiência computacional foi essencial para a obtenção da base de dados necessária para o treinamento dos algoritmos de regressão

e classificação baseados em *machine learning*.

4.3.3 Vibracao.torcional

O algoritmo retorna o cálculo do torque e os valores de tempo de vibração torcional. O torque é calculado a partir da média da diferença entre os deslocamentos volumétrico e cilíndrico da resposta. Os valores de tempo são extraídos da resposta, e os deslocamentos torsionais são calculados e combinados com os tempos correspondentes em uma matriz de saída.

4.3.4 Ordem.espectro

O algoritmo apresentado realiza a análise espectral de um sinal de vibração torcional para identificar frequências específicas e suas amplitudes associadas. Inicialmente, a matriz de entrada, que contém valores de tempo e vibração torcional, é separada em dois vetores distintos. Em seguida, utiliza-se a função **espectro.exemplo1** para calcular o espectro das frequências e suas respectivas amplitudes. A frequência de rotação é determinada dividindo-se o número de rotações por minuto (RPM) por 60, obtendo-se assim a frequência em Hertz. Variáveis auxiliares são inicializadas e a diferença entre as frequências adjacentes no vetor de frequências é calculada.

O algoritmo então executa iteração de 1 a 24, calculando, para cada iteração, as meias ordens, isto é, os múltiplos da metade da frequência de rotação, divididas em 24 intervalos. Em cada iteração, o algoritmo busca a frequência no espectro que mais se aproxima da frequência alvo (cada uma das meias ordens), extrai as amplitudes associadas a essa frequência, e armazena esses valores em um vetor. Após todas as iterações, o vetor resultante, contendo as frequências específicas e suas amplitudes, é retornada como saída. Dessa forma, o algoritmo fornece uma análise detalhada das componentes de frequência significativas no sinal de vibração torcional.

4.3.5 Tensoes.global

Este algoritmo calcula as tensões em um sistema vibratório utilizando dados de resposta de tempo e vibração torcional. Inicialmente, os dados de tempo e vibração são extraídos e processados para obter vetores de frequência, amplitude e fase através da subfunção **espectro.exemplo**. A frequência de rotação é calculada, e as variáveis de entrada do algoritmo são inicializadas. Em seguida, o algoritmo calcula as componentes específicas no domínio da frequência (fase, módulo e harmônicos) da vibração torcional.

A amplitude de vibração torcional resultante é calculada a partir das componentes ao longo dos eixos horizontal e vertical. Em seguida, o momento de inércia e o módulo de seção são determinados com base nas dimensões especificadas. Utilizando a amplitude calculada, o algoritmo determina a tensão resultante no sistema. Por fim, os vetores de frequência e amplitude, juntamente com a tensão calculada, são retornados como saída. Em resumo, o algoritmo processa dados vibratórios para determinar frequências, amplitudes e a tensão resultante, considerando a geometria e as propriedades físicas do sistema.

4.4- Conclusão

O capítulo conclui apresentando uma síntese abrangente dos algoritmos desenvolvidos para implementar o programa DEFault. A exposição detalhada dos códigos subjacentes aos modelos termodinâmico e dinâmico oferece uma compreensão mais detalhada das técnicas utilizadas para simular falhas em máquinas a diesel. Os algoritmos descritos abrangem desde o cálculo da pressão no cilindro e do torque gerado até a análise espectral e o cálculo das tensões resultantes. Através de métodos numéricos como o algoritmo de Runge-Kutta e técnicas de processamento de sinais, os modelos são capazes de capturar o comportamento complexo dos motores Diesel e as consequências das falhas em seu desempenho. Esses algoritmos fornecem uma base para a simulação de falhas em máquinas a diesel, contribuindo para o entendimento e otimização desses sistemas essenciais em uma variedade de aplicações industriais.

5- Resultados

O experimento realizado nesta tese consiste em avaliar a capacidade do sistema inteligente preditivo de avaliar a severidade de uma falha considerando a pressão máxima, média e medidas espectrais para diversos tipos de falhas. O experimento consiste em realizar a regressão de falhas adotando um procedimento semelhante ao apresentado em (Park et al., 2019), (Guerrero and Jimanez-Espadafor, 2019) e (Pestana-Viana et al., 2019).

Essas tarefas foram realizadas através dos seguintes modelos para regressão: redes neurais artificiais - *Multilayer Perceptron* (MLP), *Random Forest*, Máquinas de Vetores de Suporte e Regressão de Processo Gaussiano.

Para o RF o número de árvores foi obtido empiricamente conforme feito em (Pestana-Viana et al., 2019). Já para o MLP, a camada de entrada foi definida com o mesmo tamanho que vetor de características de entrada adotando-se uma camada intermediária com aproximadamente a mesma quantidade de neurônios que a entrada.

O banco de dados 3500-DEFault original foi dividido em 2 conjuntos disjuntos com aproximadamente 80% e 20% dos sinais para treinamento e teste, respectivamente. Para o modelo MLP utiliza-se 3 conjuntos disjuntos com aproximadamente com aproximadamente 70%, 10% e 20% de os sinais para treinamento, validação e teste.

Cada um dos conjuntos é escolhido para representar os dados com variabilidade máxima do aspecto da intensidade da severidade da falha.

O processo de aprendizado (treinamento) do RF foi realizado aplicando-se o algoritmo *bootstrap aggregating* ($bagging$) e o processo de aprendizado (treinamento) do MLP foi realizado aplicando-se o algoritmo Levenberg-Marquardt $backpropagation$, empregando-se o conjunto de validação para evitar que o modelo se torne excessivamente especializada nos sinais de treinamento, perdendo assim sua capacidade de generalização (de Lima et al., 2013).

Para evitar viés de dados na regressão, a técnica de validação cruzada k-fold foi aplicada em todos os testes de regressão, dividindo a base 3500-DEFault em k = 5 vezes com a finalidade de alterar circularmente o subconjunto de teste mantendo a proporção de 80% e 20% entre treino e teste, bem como reduzir o tempo de execução de cada teste,

conforme os estudos de (James et al., 2014).

Todas as tabelas e gráficos relacionados ao experimento de regressão de falhas utilizam o RMSE (*Root Mean Square Error*) com métrica de desempenho para a regressão.

Para os dados de teste da base 3500-DEFault, X_i representa os elementos observados e $\check{X}_i$ são os elementos previstos. O valor RMSE é representado por $W \pm \sigma$, onde W é a precisão total e σ seu desvio padrão durante a validação k-fold.

Monitorando-se a variância do RMSE associado ao passo k-fold de cada regressor (para o nível de ruído de 60 dB), a variância máxima obtida em cada regressor foi $\sigma^2 = [4,9 \cdot 10^{-4}\ 4,4 \cdot 10^{-2}\ 2,5 \cdot 10^{-3}\ 1,0 \cdot 10^{-6}]$ %. Dessa forma, não houve necessidade de alterar o número de folds, visto que os modelos não se mostraram enviesados.

Por fim, com a redução de folds $k = 10$ (mais usual) para $k = 5$, o tempo total de execução (ToE) de cada regressão (dependendo de cada método e dos níveis de ruído evoluídos) foi significativamente reduzido, visto o menor número de execuções para definir os parâmetros de cada regressor.

O Tempo de Execução (ToE) computacional, é medido em segundos e foi estimado utilizando um computador pessoal, com as seguintes características: CPU core i5, 8 GB RAM, sem GPU e utilizando os 2 núcleos físicos para processamento paralelo.

5.1- Ajuste de hiperparâmetros dos regressores

O ajuste dos hiperparâmetros da MLP, RF, SVM e GPR é uma etapa dedicada para projetar cada regressor de forma a maximizar o desempenho durante treinamento. Tais parâmetros são avaliados para cada tipo de regressor, segundo um conjunto de valores pré-definidos, monitorando-se o valor mínimo do RMSE (média e desvio padrão) na etapa de treino.

Para o MLP, o número de neurônios na camada oculta H simulada foi $H = 1, 2, \ldots, 1000$; para o RF o número de árvores simuladas B foi $B = 1, 2, \ldots, 1000$; para SVM o parâmetro κ da função kernel polinomial de ordem $d = 1$, foi $\kappa = 0, 1, \ldots, 10$ e $\delta = \frac{\kappa}{10}$ e para GPR o parâmetro de desvio padrão σ_G da distância de covariância $C_{5/2}(d)$ simulada foi $\sigma_G = 0, 1, \ldots, 10$ com $\rho = \frac{\sigma_G}{100}$.

Durante as simulações do SVM e do GPRM, e de acordo com os estudos de (Keerthi, 2002), foram definidos $\delta = \frac{\kappa}{10}$ para o SVM e, para o GPR, o desvio padrão da função kernel foi fixado em $\rho = \frac{\sigma_G}{100}$. Esses ajustes foram realizados com o objetivo de reduzir o número excessivo de simulações em ambos os regressores.

A Figura 29 mostra as curvas de ajuste onde são apresentados o valor médio do RMSE μ_{RMSE} e avariabilidade do RMSE σ_{RMSE} do conjunto de treinamento para cada hiperparâmetro avaliado, tal que o eixo x mostra o intervalo de valores atribuídos para cada hiperparâmetro. Por fim, os valores definidos para cada regressor são apresentados na Tabela 5.

Tabela 5 – Valores selecionados dos hiperparâmetros de cada regressor.

Regressor	Hiperparâmetros
MLP	$H = 120$
RF	$B = 100$
SVM	$\kappa = 0,75$, $\delta = 0,08$
GPR	$\sigma_G = 5,45$, $\rho = 0,06$

Observe que quando a curva de sintonia tende a um mínimo transiente ou um mínimo estável, há uma diminuição intrínseca no desvio padrão da validação do conjunto de treinamento. Dessa forma, vemos a importância de avaliar a escolha correta desses parâmetros que implicam em uma menor variabilidade do erro de treinamento, levando a um resultado mais estável.

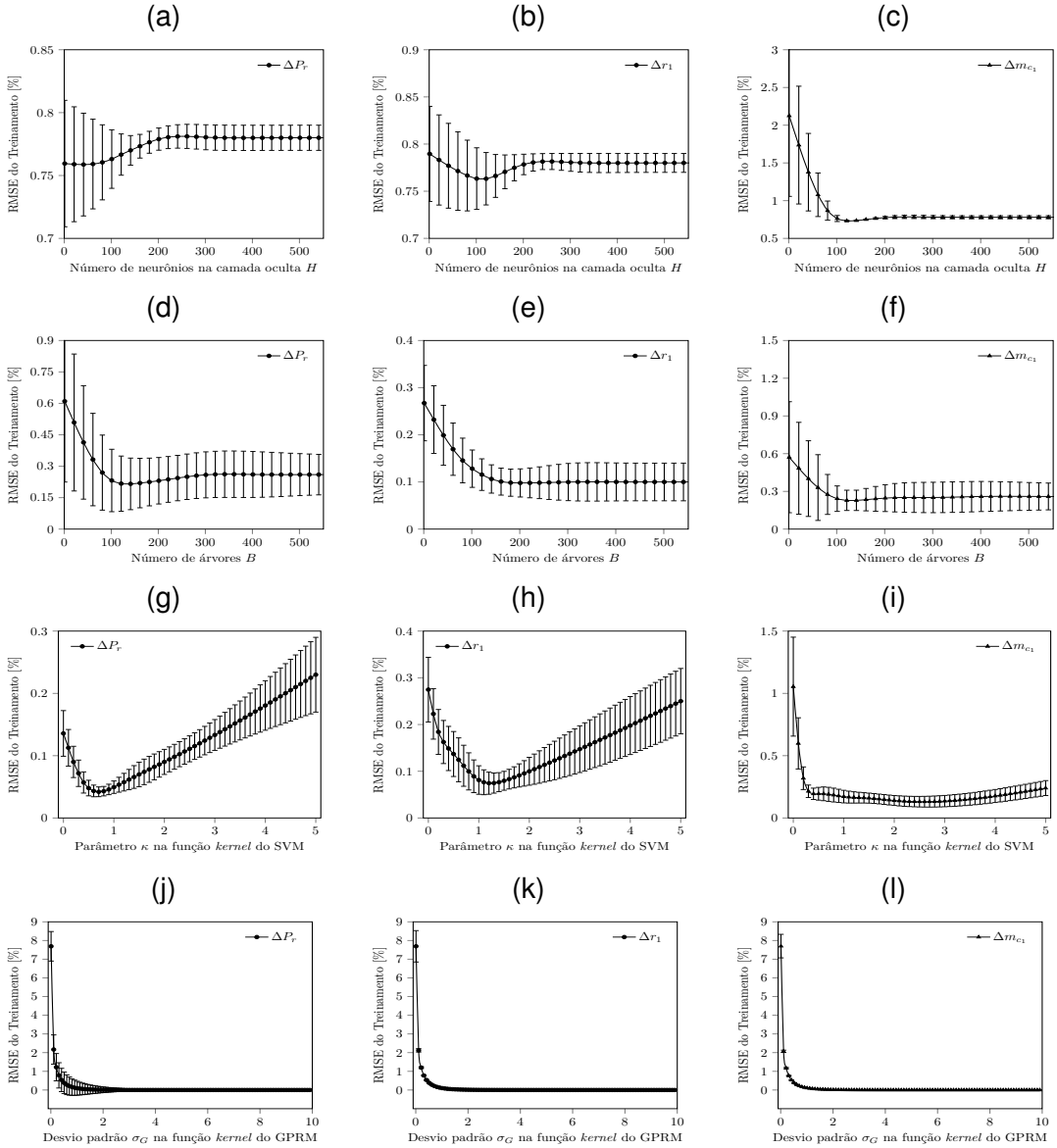


Figura 29 – Gráfico de erro de várias curvas de sintonia com nível AWGN $L = 60$ dB SNR.: (a), (d), (g) e (j) para ΔP_r e regressores MLP, RF, SVM e GPR, respectivamente; (b), (e), (h) e (k) para Δr_1 e regressores MLP, RF, SVM e GPR, respectivamente; (c), (f), (i) e (l) para Δm_{c_1} e regressores MLP, RF, SVM e GPR, respectivamente. Nos gráficos acima, a linha pontilhada representa μ_{RMSE} e as barras representam σ_{RMSE} após a validação cruzada de 5 vezes.

5.2- Testes de classificação

O experimento de classificação de falhas consiste em adotar procedimentos semelhantes apresentados em (Pestana-Viana et al., 2016) e (de Lima et al., 2013), para avaliar a capacidade do sistema de discriminação de falhas, adicionando a pressão

máxima, média e as medidas espectrais, não apenas para o normal, mas também para outras falhas. Essas tarefas foram realizadas com os classificadores: *Random Forest*, com o vetor de características como entrada, com o número de árvores igual a 84, obtidas empiricamente, como feitas em (de Lima et al., 2013) e a saída com o tamanho do número de classes de falha e o *perceptron* multicamada, com camada de entrada com o mesmo tamanho do vetor de características, uma camada oculta com a quantidade de neurônios igual ao tamanho da camada de entrada, também obtida empiricamente e a camada de saída com o número de neurônios igual ao número de classes a serem discriminadas.

O banco de dados original **3500-DEFault** foi dividido em 2 conjuntos disjuntos (mutuamente exclusivos) com aproximadamente 80% e 20% dos sinais para treinamento e teste, respectivamente, e em 3 conjuntos disjuntos com aproximadamente 70%, 10% e 20% de os sinais para treinamento, validação e teste, respectivamente, para o classificador da RNA (de Lima et al., 2013) e (Pestana-Viana et al., 2016). Cada um dos conjuntos é escolhido para representar os dados com variabilidade máxima do aspecto da intensidade da severidade da falha (Pestana-Viana et al., 2018). O processo de aprendizado (treinamento) de RF foi realizado aplicando o algoritmo *bootstrap aggregating (bagging)* e o processo de aprendizado (treinamento) da RNA foi realizado aplicando o algoritmo de retropropagação de erros de Levenberg-Marquardt (Pestana-Viana et al., 2018).

O conjunto de validação é empregado para evitar que a RNA se torne excessivamente especializada nos sinais de treinamento, perdendo sua capacidade de generalização (de Lima et al., 2013) e (Haykin, 2009).

Para evitar o desempenho tendencioso dos dados na classificação, a técnica de validação cruzada k-*fold* foi aplicada em todos os testes de classificação, dividindo o sinal da base **3500-DEFault** em k = 5 *folds*, com o objetivo de alterar circularmente o subconjunto de testes, mantendo a proporção de 80 e 20% para os conjuntos de treinamento e teste, respectivamente.

Todas as tabelas relacionadas ao experimento de classificação de falhas apresentam o desempenho de classificação X_c/Y_c, também chamado de matriz de confusão, para os dados de teste da base de dados **3500-DEFault**, onde X_c representa os sinais reconhecidos e Y_c é o número total de sinais para a classe-alvo sob análise. O valor total do desempenho da classificação de precisão da matriz de confusão é representado por uma pontuação com a variável $W \pm \sigma$, em que W representa a proporção percentual total

de elementos verdadeiros classificados, por um total de elementos de teste e σ é o desvio padrão durante a etapa de validação cruzada k-*fold*.

Na base **3500-DEFault**, diferentes níveis de ruído foram aplicados com 60, 30, 15 e 0 dB de relação sinal-ruído (SNR), usando ruído branco gaussiano aditivo (AWGN) nos sinais originais, para avaliar sua influência sobre o desempenho de classificação e de sorte a emular a influência de uma medição real. A Figura 30 mostra as variáveis torque $\{M(t)\}$ sem adição de ruído como na Eq. (3) e torque com 15 dB SNR AWGN ($\{\check{M}(t)\}$).

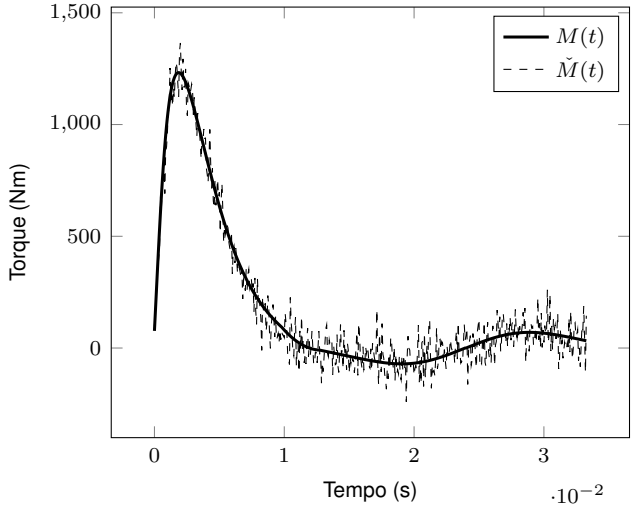


Figura 30 – Processo de aplicação de 15 dB SNR de ruído aditivo na variável de torque. No gráfico acima, a linha tracejada representa o sinal com adição de ruído e a linha reta representa o sinal original sem ruído. A nova variável de torque com AWGN ($\{\check{M}(t)\}$) is $\{\check{M}(t)\} = \{M(t)\} + \nu(t)$, onde $\nu(t)$ é ruído gaussiano branco com SNR de 15dB. O processo é o mesmo para as demais variáveis (pressão e vibração torcional).

O desempenho do aprendizado de máquina foi comparado com a abordagem da técnica de mínimos quadrados de Levenberg-Marquardt (LMLS) (Gutiérrez, 2016), para avaliar a eficiência do sistema proposto.

5.2.1 Resultados de classificação usando 60 dB SNR AWGN

Para os testes abaixo, todas as falhas foram incluídas testando o desempenho de todos os cenários de falha do banco de dados de **3500-DEFault**. Diferentes divisões de classes foram usadas para investigar o desempenho do classificador para diferentes configurações na camada de saída, além de obter os resultados expandindo as classes

de falhas. Os testes foram definidos pelo número de classes a serem previstas pelo classificador e em cada teste foi adotada a seguinte nomenclatura de classe:

- Teste com 4 classes: normal (C1), redução de pressão no coletor de admissão (C2), redução da razão de compressão nos cilindros (C3) e redução da quantidade de combustível injetado nos cilindros (C4);
- Teste com 8 classes: normal (C1), falha do cilindro: cilindro 1 (C2), cilindro 2 (C3), cilindro 3 (C4), cilindro 4 (C5), cilindro 5 (C6) e cilindro 6 (C7) e falha em todos os cilindros (C8);
- Teste de severidade com 4 classes (S4): nível de severidade percentual (S): $1 \leq S \leq 10$% (S1), $10 < S \leq 20$% (S2), $20 < S \leq 30$% (S3) and $30 < S \leq 50$% (S4).

Tabela 6 – Matriz de confusão com reconhecimento de padrões de 4 classes usando o vetor de características V_f com ruído de 60 dB e RF.

Classe	**Alvo**			
	C1	C2	C3	C4
C1	**50/50**	0/45	0/305	0/300
C2	0/50	**45/45**	5/305	0/300
C3	0/50	0/45	**300/305**	0/300
C4	0/50	0/45	0/305	**300/300**
Total	**99,3**	$\pm$ **0,23**	**(%)**	

Tabela 7 – Matriz de confusão com reconhecimento de padrões de 4 classes usando o vetor de características V_f com ruído de 60 dB e RNA.

Classe	**Alvo**			
	C1	C2	C3	C4
C1	**50/50**	0/45	20/305	10/300
C2	0/50	**45/45**	0/305	0/300
C3	0/50	0/45	**285/305**	0/300
C4	0/50	0/45	0/305	**290/300**
Total	**95,7**	$\pm$ **0,97**	**(%)**	

As Tabelas 6 a 11 apresentam uma estrutura semelhante às descritas em (de Lima et al., 2013) e (Pestana-Viana et al., 2016), ou seja, realiza uma tarefa de classificação usando V_f com uma RNA de 84-84-4 e 84-84-8 neurônios nas camadas de entrada, oculta e saída, respectivamente, para testes com 4 e 8 classes, respectivamente, e 84 árvores com 500 divisões para o RF classificador.

Tabela 8 – Matriz de confusão com reconhecimento de padrões de 8 classes usando o vetor de características V_f com ruído de 60 dB e RF.

Classe	**Alvo**							
	C1	C2	C3	C4	C5	C6	C7	C8
C1	**25/25**	0/90	0/135	0/70	5/100	0/80	0/140	0/60
C2	0/25	**90/90**	0/135	0/70	0/100	0/80	0/140	0/60
C3	0/25	0/90	**135/135**	0/70	0/100	0/80	0/140	0/60
C4	0/25	0/90	0/135	**70/70**	0/100	0/80	0/140	0/60
C5	0/25	0/90	0/135	0/70	**95/100**	0/80	0/140	0/60
C6	0/25	0/90	0/135	0/70	0/100	**80/80**	0/140	0/60
C7	0/25	0/90	0/135	0/70	0/100	0/80	**140/140**	0/60
C8	0/25	0/90	0/135	0/70	0/100	0/80	0/140	**60/60**
Total (%)	**99,3 $\pm$ 0,27**							

Tabela 9 – Matriz de confusão com reconhecimento de padrões de 8 classes usando o vetor de característica V_f com ruído de 60 dB e RNA.

Classe	**Alvo**							
	C1	C2	C3	C4	C5	C6	C7	C8
C1	**25/25**	0/90	0/135	0/70	0/100	5/80	0/140	0/60
C2	0/25	**90/90**	0/135	0/70	0/100	0/80	0/140	0/60
C3	0/25	0/90	**135/135**	0/70	0/100	0/80	0/140	0/60
C4	0/25	0/90	0/135	**70/70**	0/100	0/80	0/140	0/60
C5	0/25	0/90	0/135	0/70	**100/100**	0/80	0/140	0/60
C6	0/25	0/90	0/135	0/70	0/100	**75/80**	0/140	0/60
C7	0/25	0/90	0/135	0/70	0/100	0/80	**140/140**	0/60
C8	0/25	0/90	0/135	0/70	0/100	0/80	0/140	**60/60**
Total (%)	**99,3 $\pm$ 0,85**							

Comparando as Tabelas 6 com 11, percebe-se claramente que o uso do RF melhora o desempenho de cada classe individual e consequentemente, a acurácia geral do sistema para mais de 99% nos testes de 4 e 8 classes. Comparando o desempenho com os classificadores da RNA para a mesma tarefa, houve uma diminuição na precisão de 3,6% para os testes de 4 e 8 classes. No entanto, para o teste de classificação de severidade, houve uma diferença de 11,5% no desempenho entre os classificadores RF e RNA.

Percebe-se claramente que o uso de RF melhora o desempenho de cada classe individual e, consequentemente, a precisão geral do sistema para 99% e reduzindo a dispersão entre as realizações de validação cruzada, nos testes de 4 e 8 classes.

Tabela 10 – Matriz de confusão com reconhecimento de padrões de severidade de 4 classes usando o vetor de característica V_f com ruído de 60 dB e RF.

Classe	**Alvo**			
	S1	S2	S3	S4
S1	**65/65**	0/100	0/245	0/240
S2	0/65	**95/100**	5/245	0/240
S3	0/65	5/100	**240/245**	5/240
S4	0/65	0/100	0/245	**235/240**
Total	**97,9**	$\pm$ **0,85**	**(%)**	

Tabela 11 – Matriz de confusão com reconhecimento de padrões de severidade de 4 classes usando o vetor de característica V_f com ruído de 60 dB e RNA.

Classe	**Alvo**			
	S1	S2	S3	S4
S1	**0/65**	0/100	0/245	0/240
S2	65/65	**95/100**	10/245	0/240
S3	0/65	5/100	**230/245**	10/240
S4	0/65	0/100	5/245	**230/240**
Total	**86,4**	$\pm$ **2,7**	**(%)**	

5.2.2 Resultados da classificação abordando diferentes níveis de ruído branco gaussiano

Para os testes abaixo, todas as falhas do banco de dados foram incluídas para avaliar o desempenho de todos os cenários de falha do banco de dados **3500-DEFault**. Nesta seção, uma configuração semelhante à adotada na Seção 5.2.1 foi aplicada, incluindo ainda, uma classe de teste extra de 14 elementos. Os testes foram estabelecidos com as mesmas nomenclaturas adotadas para as configurações de 4 classes, 6 classes e 4 classes de severidade, conforme apresentado na Seção 5.2.1 e uma nomenclatura adicional de 14 classes definida da seguinte forma:

- teste de 14 classes: normal (C1), redução de pressão no coletor de admissão (C2), redução da razão de compressão no: cilindro 1 (C3), cilindro 2 (C4), cilindro 3 (C5), cilindro 4 (C6) , cilindro 5 (C7) e cilindro 6 (C8) e redução da quantidade de combustível injetado em: cilindro 1 (C9), cilindro 2 (C10), cilindro 3 (C11), cilindro 4 (C12), cilindro 5 (C13) e cilindro 6 (C14).

Os resultados mostrados na Tabela 12 indicam um desempenho geral de predição

Tabela 12 – Resultados da acurácia de predição para os classificadores RF e ANN com SNR de 60, 30, 15 e 0 dB usando o vetor de características V_f.

SNR (dB)	N, Classes	Acurácia (%)	
		RF	ANN
60	4	99,3 $\pm$ 0,2	95,7 $\pm$ 1,0
	8	99,3 $\pm$ 0,3	99,3 $\pm$ 0,9
	14	99,3 $\pm$ 0,3	96,4 $\pm$ 0,9
	S4	97,9 $\pm$ 0,9	86,4 $\pm$ 2,7
30	4	98,6 $\pm$ 0,3	92,9 $\pm$ 1,0
	8	98,6 $\pm$ 0,3	97,1 $\pm$ 0,8
	14	97,9 $\pm$ 0,4	95,8 $\pm$ 1,0
	S4	97,9 $\pm$ 0,9	86,1 $\pm$ 2,3
15	4	95,1 $\pm$ 0,3	88,6 $\pm$ 1,0
	8	95,8 $\pm$ 0,3	90,7 $\pm$ 0,8
	14	94,4 $\pm$ 0,4	92,1 $\pm$ 0,8
	S4	92,3 $\pm$ 0,9	76,4 $\pm$ 3,2
0	4	83,5 $\pm$ 0,9	75,7 $\pm$ 1,0
	8	81,4 $\pm$ 0,8	79,3 $\pm$ 1,3
	14	80,7 $\pm$ 0,9	72,9 $\pm$ 1,2
	S4	76,4 $\pm$ 1,3	60,7 $\pm$ 3,7

para o classificador RF de 99,3%, adequado para o problema de classificação de falhas de diesel, tornando o método proposto eficaz para o reconhecimento de padrões de falha em análise neste trabalho. O classificador RNA, em geral, obteve desempenho inferior ao RF, tornando o RF mais adequado para as tarefas de classificação abordadas neste trabalho.

5.2.3 Análise Comparativa entre o Método de Levenberg-Marquardt e classificador *Random Forest*

A análise comparativa entre a técnica de Levenberg-Marquardt para quadrados mínimos (LMLS) (Gutiérrez, 2016) e as abordagens propostas usando 2 classificadores diferentes visando detectar o tipo de falha, bem como a sua severidade associada (normal, ΔP_a, Δr e Δm_c). A localização da falha (normal, cilindros 1 a 6 e todos os cilindros), o tipo e a localização da falha combinados e a faixa de severidade da falha são abordados para observar os aspectos qualitativos de ambas as técnicas.

Deve-se notar que a técnica LMLS e as abordagens propostas apresentam diferenças estruturais significativas. O algoritmo LMLS exige que o tipo de falha seja conhecido anteriormente pelo aplicativo do sistema e retorne o nível de severidade em termos numéricos, enquanto as abordagens propostas usam técnicas de classificação para detectar o tipo e o local da falha, bem como a sua faixa de severidade.

A Tabela 13 apresenta os cenários em que as técnicas foram analisadas. Os casos 1 a 6 foram abordados apenas um tipo de falha, cada um com um determinado nível de severidade e ruído de medição. Somente a falha ΔP_a com 25% de severidade no cilindro 1 foi abordada no caso 1. No caso 2, apenas a falha Δr no cilindro 1 com 25% de severidade foi simulada e o caso 3 somente com a falha Δm_c com 25% de severidade e o cilindro 1 foram considerados. Os casos 7, 8 e 9 são equivalentes a 1, 2 e 3, mas com ruído agregado, compreendido em 15 dB de SNR em cada sinal de falha. Nos 6 casos, foi utilizado o modelo dinâmico acoplado ao termodinâmico.

Tabela 13 – Tabela de resumo com falhas e níveis de severidade abordados para cada cenário/caso de entrada.

Caso	Cyl.	SNR (dB)	Falha simulada & severidade (%)		
			ΔP_a	Δr	Δm_c
1	1	-	25	-	-
2	1	-	-	25	-
3	1	-	-	-	25
4	1	15	25	-	-
5	1	15	-	25	-
6	1	15	-	-	25

Tabela 14 – Tabela de erro máximo na identificação da severidade de cada variável usando a técnica LMLS e os resultados da classificação usando RF para os classificadores 4-c, 8-c, 14-c e S4, para diferentes níveis de ruído e 2500 RPM.

Caso	Metodologia							Tempo	
	LMLS			RF				LMLS (%)	RF
	ΔP_a	Δr	Δm_c	4-c	8-c	14-c	S4	(horas)	(ms)
1	$< 10^{-5}$	0	0	C2	C2	C2	S3	38	21
2	0	$< 10^{-5}$	0	C3	C2	C3	S3	38	21
3	0	0	$< 10^{-5}$	C4	C2	C9	S3	38	21
4	$< 10^{-5}$	0	0	C2	C2	C2	S3	38	21
5	0	$< 10^{-5}$	0	C3	C2	C3	S3	38	21
6	0	0	$< 10^{-5}$	C4	C2	C9	S3	38	21

Na Tabela 14, o erro absoluto de estimativa de severidade para os casos 1 a 6 foi menor que 10^{-5}%, indicando uma alta eficiência na estimativa dos níveis de severidade para a técnica LMLS. No entanto, deve-se mencionar que o algoritmo requer que o conhecimento prévio do tipo de falha seja analisado e leva cerca de 38 horas para avaliar um único cenário.

A abordagem proposta (RF), avalia quatro classificadores diferentes. O primeiro, um experimento de 4 classes (4-c) reconhece os 3 tipos de falhas em análise como sendo das classes C2, C3 e C4, associadas a ΔP_a, Δr e Δm_c, respectivamente. Os classificadores de 8 classes (8-c) detectam adequadamente em qual cilindro a falha ocorreu como sendo da classe C2, ou seja, cilindro 1. O classificador combinado de falhas de tipo e localização, que é um classificador de 14 classes (14-c), foi capaz de detectar com precisão as classes C2, C3 e C9 relacionadas a ΔP_a e cilindro 1, Δr e cilindro 1 e Δm_c e cilindro 1, respectivamente. E, finalmente, a avaliação do intervalo de severidade utilizou um classificador de severidade de 4 classes (S4) e pôde apontar assertivamente a classe S3 correspondente ao intervalo de severidade de 20% a 30%. O custo computacional de uma avaliação da técnica proposta, com o classificador já treinado, levou cerca de 21 ms.

A principal contribuição desta análise reside nas vantagens das técnicas de *machine learning* em comparação com o método tradicional LMLS. Enquanto o LMLS demonstrou excelente precisão na estimativa numérica da severidade de falha, ele requer informações prévias sobre o tipo de falha e implica em um custo computacional elevado. Em contraste, as abordagens baseadas em *machine learning* propostas nesta Seção são baseadas em classificação, capazes de reconhecer com precisão as classes de falhas e suas faixas de severidade nos cenários de teste. Estas técnicas dispensam a necessidade de informações prévias sobre a falha, além de serem computacionalmente mais eficientes. Para superar as limitações da abordagem de classificação, será desenvolvida a técnica baseada na regressão via algoritmos de *machine learning* na Seção 5.3.1.

5.3- Testes de regressão

Após o ajuste dos hiperparâmetros dos regressores, descrito na Seção 5.1, são implementados os testes de regressão empregando os mesmos hiperparâmetros apresentados na Tabela 5, conforme estudos realizados por (Li et al., 2019; Li et al., 2020).

Todas as falhas básicas foram incluídas para avaliar o desempenho para os cenários de falha do banco de dados 3500-DEFault. Diferentes níveis de severidade $S = [1.0, 1.1, 1.2, \ldots, 50]$ % foram utilizados para investigar o desempenho de cada regressor, bem como obter os resultados por meio da análise do grau de severidade de cada variável (falha).

Os valores apresentados nas Tabelas 15–18 representam os erros quadráticos médios (RMSE) dos regressores para cada parâmetro de falha (FP), considerando diferentes níveis de ruído branco gaussiano aditivo (AWGN) nos sinais originais, isto é, AWGN de 60, 30, 15 e 0 dB de SNR, respectivamente. O RMSE é uma medida da diferença entre os valores previstos pelo modelo e os valores reais observados, sendo calculado como a raiz quadrada da média das diferenças ao quadrado. As Figuras 31 e 32 mostram as curvas de regressão para os níveis AWGN L de 60 e 0 dB, respectivamente.

Tabela 15 – RMSE dos regressores para cada parâmetro (FP), considerando 60 dB AWGN.

FP	Regressor			
	MLP	RF	SVM	GPR
ΔP_r	0,84 $\pm$ 0,06	0,20 $\pm$ 0,11	0,05 $\pm$ 0,01	0,015 $\pm$ 0,001
Δr_1	0,85 $\pm$ 0,07	0,10 $\pm$ 0,03	0,11 $\pm$ 0,03	0,018 $\pm$ 0,001
Δr_2	0,85 $\pm$ 0,07	0,16 $\pm$ 0,17	0,14 $\pm$ 0,03	0,018 $\pm$ 0,001
Δr_3	0,84 $\pm$ 0,05	0,17 $\pm$ 0,12	0,09 $\pm$ 0,03	0,018 $\pm$ 0,001
Δr_4	0,84 $\pm$ 0,06	0,16 $\pm$ 0,07	0,13 $\pm$ 0,04	0,018 $\pm$ 0,001
Δr_5	0,86 $\pm$ 0,07	0,27 $\pm$ 0,12	0,11 $\pm$ 0,04	0,017 $\pm$ 0,001
Δr_6	0,84 $\pm$ 0,05	0,29 $\pm$ 0,19	0,13 $\pm$ 0,03	0,018 $\pm$ 0,001
Δm_{c_1}	0,81 $\pm$ 0,05	0,26 $\pm$ 0,21	0,17 $\pm$ 0,05	0,022 $\pm$ 0,001
Δm_{c_2}	0,83 $\pm$ 0,05	0,23 $\pm$ 0,12	0,18 $\pm$ 0,04	0,020 $\pm$ 0,001
Δm_{c_3}	0,84 $\pm$ 0,05	0,15 $\pm$ 0,06	0,20 $\pm$ 0,04	0,019 $\pm$ 0,001
Δm_{c_4}	0,84 $\pm$ 0,04	0,22 $\pm$ 0,11	0,19 $\pm$ 0,05	0,018 $\pm$ 0,001
Δm_{c_5}	0,84 $\pm$ 0,05	0,13 $\pm$ 0,05	0,16 $\pm$ 0,04	0,017 $\pm$ 0,001
Δm_{c_6}	0,83 $\pm$ 0,05	0,24 $\pm$ 0,13	0,12 $\pm$ 0,03	0,020 $\pm$ 0,001
ToE	15,36 $\pm$ 3,57	17,89 $\pm$ 1,08	124,43 $\pm$ 26,83	176,94 $\pm$ 12,53

Tabela 16 – RMSE dos regressores para cada parâmetro (FP), considerando 30 dB AWGN.

FP	Regressor			
	MLP	RF	SVM	GPR
ΔP_r	1,03 $\pm$ 0,11	0,35 $\pm$ 0,14	0,12 $\pm$ 0,03	0,053 $\pm$ 0,002
Δr_1	1,04 $\pm$ 0,13	0,21 $\pm$ 0,11	0,25 $\pm$ 0,06	0,074 $\pm$ 0,015
Δr_2	1,02 $\pm$ 0,14	0,27 $\pm$ 0,25	0,29 $\pm$ 0,08	0,071 $\pm$ 0,004
Δr_3	1,06 $\pm$ 0,11	0,24 $\pm$ 0,19	0,22 $\pm$ 0,06	0,064 $\pm$ 0,003
Δr_4	1,07 $\pm$ 0,15	0,20 $\pm$ 0,14	0,28 $\pm$ 0,07	0,062 $\pm$ 0,004
Δr_5	1,04 $\pm$ 0,16	0,28 $\pm$ 0,18	0,25 $\pm$ 0,06	0,066 $\pm$ 0,002
Δr_6	1,04 $\pm$ 0,16	0,26 $\pm$ 0,17	0,22 $\pm$ 0,06	0,060 $\pm$ 0,001
Δm_{c_1}	1,05 $\pm$ 0,12	0,23 $\pm$ 0,06	0,49 $\pm$ 0,11	0,126 $\pm$ 0,003
Δm_{c_2}	1,07 $\pm$ 0,14	0,24 $\pm$ 0,06	0,53 $\pm$ 0,12	0,143 $\pm$ 0,003
Δm_{c_3}	1,09 $\pm$ 0,14	0,19 $\pm$ 0,04	0,49 $\pm$ 0,13	0,114 $\pm$ 0,005
Δm_{c_4}	1,12 $\pm$ 0,11	0,30 $\pm$ 0,11	0,54 $\pm$ 0,14	0,105 $\pm$ 0,002
Δm_{c_5}	1,04 $\pm$ 0,15	0,18 $\pm$ 0,04	0,50 $\pm$ 0,13	0,116 $\pm$ 0,003
Δm_{c_6}	1,09 $\pm$ 0,13	0,25 $\pm$ 0,05	0,29 $\pm$ 0,08	0,100 $\pm$ 0,003
ToE	32,67 $\pm$ 5,17	18,94 $\pm$ 0,81	20,34 $\pm$ 6,24	174,22 $\pm$ 39,22

Tabela 17 – RMSE dos regressores para cada parâmetro (FP), considerando 15 dB AWGN.

FP	Regressor			
	MLP	RF	SVM	GPR
ΔP_r	1,62 $\pm$ 0,22	0,37 $\pm$ 0,14	0,60 $\pm$ 0,17	0,184 $\pm$ 0,008
Δr_1	1,58 $\pm$ 0,22	0,36 $\pm$ 0,06	0,63 $\pm$ 0,16	0,328 $\pm$ 0,024
Δr_2	1,56 $\pm$ 0,24	0,33 $\pm$ 0,11	0,67 $\pm$ 0,18	0,323 $\pm$ 0,025
Δr_3	1,64 $\pm$ 0,30	0,31 $\pm$ 0,08	0,67 $\pm$ 0,17	0,289 $\pm$ 0,015
Δr_4	1,63 $\pm$ 0,28	0,38 $\pm$ 0,30	0,68 $\pm$ 0,17	0,271 $\pm$ 0,013
Δr_5	1,69 $\pm$ 0,32	0,35 $\pm$ 0,08	0,58 $\pm$ 0,17	0,298 $\pm$ 0,007
Δr_6	1,62 $\pm$ 0,36	0,37 $\pm$ 0,13	0,48 $\pm$ 0,13	0,260 $\pm$ 0,004
Δm_{c_1}	1,79 $\pm$ 0,33	0,61 $\pm$ 0,08	4,67 $\pm$ 1,12	0,728 $\pm$ 0,039
Δm_{c_2}	1,83 $\pm$ 0,28	0,69 $\pm$ 0,14	3,68 $\pm$ 0,81	0,765 $\pm$ 0,028
Δm_{c_3}	1,67 $\pm$ 0,23	0,49 $\pm$ 0,10	1,61 $\pm$ 0,51	0,589 $\pm$ 0,047
Δm_{c_4}	1,69 $\pm$ 0,37	0,52 $\pm$ 0,08	2,05 $\pm$ 0,64	0,559 $\pm$ 0,015
Δm_{c_5}	1,71 $\pm$ 0,35	0,42 $\pm$ 0,02	2,00 $\pm$ 0,59	0,607 $\pm$ 0,022
Δm_{c_6}	1,65 $\pm$ 0,31	0,48 $\pm$ 0,06	1,32 $\pm$ 0,39	0,519 $\pm$ 0,020
ToE	34,11 $\pm$ 1,77	22,67 $\pm$ 1,01	5,37 $\pm$ 0,91	151,23 $\pm$ 18,88

Após a análise da Figura 31 e Tabela 15, nota-se um bom desempenho dos regressores para uma alta relação sinal/ruído, com RMSE menor que 1%. Este comportamento é motivado pela baixa dispersão dos elementos do vetor característico, devido ao baixo nível de ruído presente nos sinais provenientes da máquina.

Podemos também notar o aumento do RMSE na presença de uma baixa relação sinal-ruído conforme mostrado na Figura 32. Neste caso, nota-se a dispersão das previsões em relação à curva de previsão ideal, em particular para o Δm_{c_j}, de acordo com os resultados da Tabela 18.

Em resumo, a maior dispersão dos resultados de regressão em torno da curva de predição perfeita ocorre devido ao elevado nível de ruído presente nos sinais da base de

Tabela 18 – RMSE dos regressores para cada parâmetro (FP), considerando 0 dB AWGN.

FP	Regressor			
	MLP	RF	SVM	GPR
ΔP_r	1,58 $\pm$ 0,34	0,78 $\pm$ 0,25	1,24 $\pm$ 0,31	0,686 $\pm$ 0,035
Δr_1	2,19 $\pm$ 0,31	1,37 $\pm$ 0,13	5,94 $\pm$ 1,35	1,514 $\pm$ 0,065
Δr_2	2,22 $\pm$ 0,33	1,18 $\pm$ 0,14	5,43 $\pm$ 1,47	1,448 $\pm$ 0,145
Δr_3	2,08 $\pm$ 0,29	1,24 $\pm$ 0,09	5,49 $\pm$ 1,48	1,250 $\pm$ 0,051
Δr_4	1,90 $\pm$ 0,33	0,95 $\pm$ 0,16	5,29 $\pm$ 1,08	1,163 $\pm$ 0,032
Δr_5	2,16 $\pm$ 0,24	1,12 $\pm$ 0,14	5,59 $\pm$ 1,47	1,337 $\pm$ 0,063
Δr_6	1,86 $\pm$ 0,18	0,92 $\pm$ 0,10	4,38 $\pm$ 1,55	1,160 $\pm$ 0,045
Δm_{c_1}	3,76 $\pm$ 0,21	3,16 $\pm$ 0,41	6,01 $\pm$ 1,41	3,223 $\pm$ 0,078
Δm_{c_2}	3,16 $\pm$ 0,22	2,83 $\pm$ 0,08	6,01 $\pm$ 1,51	3,033 $\pm$ 0,137
Δm_{c_3}	3,21 $\pm$ 0,18	2,63 $\pm$ 0,36	5,99 $\pm$ 1,48	2,779 $\pm$ 0,111
Δm_{c_4}	3,06 $\pm$ 0,26	2,62 $\pm$ 0,19	5,94 $\pm$ 1,44	2,686 $\pm$ 0,072
Δm_{c_5}	2,82 $\pm$ 0,33	2,25 $\pm$ 0,09	5,95 $\pm$ 1,42	2,332 $\pm$ 0,167
Δm_{c_6}	2,62 $\pm$ 0,32	2,28 $\pm$ 0,23	6,07 $\pm$ 1,38	2,206 $\pm$ 0,116
ToE	32,71 $\pm$ 1,50	31,93 $\pm$ 1,74	2,80 $\pm$ 1,10	257,56 $\pm$ 92,01

dados, antes da etapa de extração de características. Isso resulta no espalhamento dos resultados de regressão, evidenciando a influência significativa do ruído na precisão das predições.

De acordo com os resultados das Tabelas 15 - 18, as 3 classes de falhas estudadas ΔP_r, Δr_j e Δm_{c_j} possuem desempenhos bem específicos de acordo com seu tipo: a falha ΔP_r, que afeta toda a dinâmica de combustão do motor, possui previsão mais precisa dado seu forte acoplamento com os parâmetros operacionais utilizados neste trabalho; as falhas Δr_j e Δm_{c_j}, que afetam a dinâmica de combustão de apenas um cilindro, possuem maior nível de dificuldade de previsão por estarem acopladas apenas com as variáveis cilindro em questão.

O desempenho da regressão para as falhas referentes às variáveis Δm_{c_j} foi inferior às demais falhas, indicando que a previsão desta esta classe de falha possui maior grau de dificuldade, devendo-se avaliar funcionalidades mais discriminantes e ajustes mais específico dos regressores.

Com relação aos regressores, o modelo MLP e RF apresentaram resultados estáveis, com baixo ToE, devido ao nível de ruído e ao ajuste definido, com resultados esperados, consistentes com trabalhos anteriores (Li et al., 2019).

O regressor SVM apresentou resultado semelhante ao RF, porém seu desempenho cai com o aumento do nível de ruído, devendo ser feitos ajustes de acordo com o ruído. O GPRM apresentou resultados promissores para este tipo de cenário e com o menor RMSE até 15 dB. Com 0 dB o desempenho dos regressores é semelhante, com exceção do SVM.

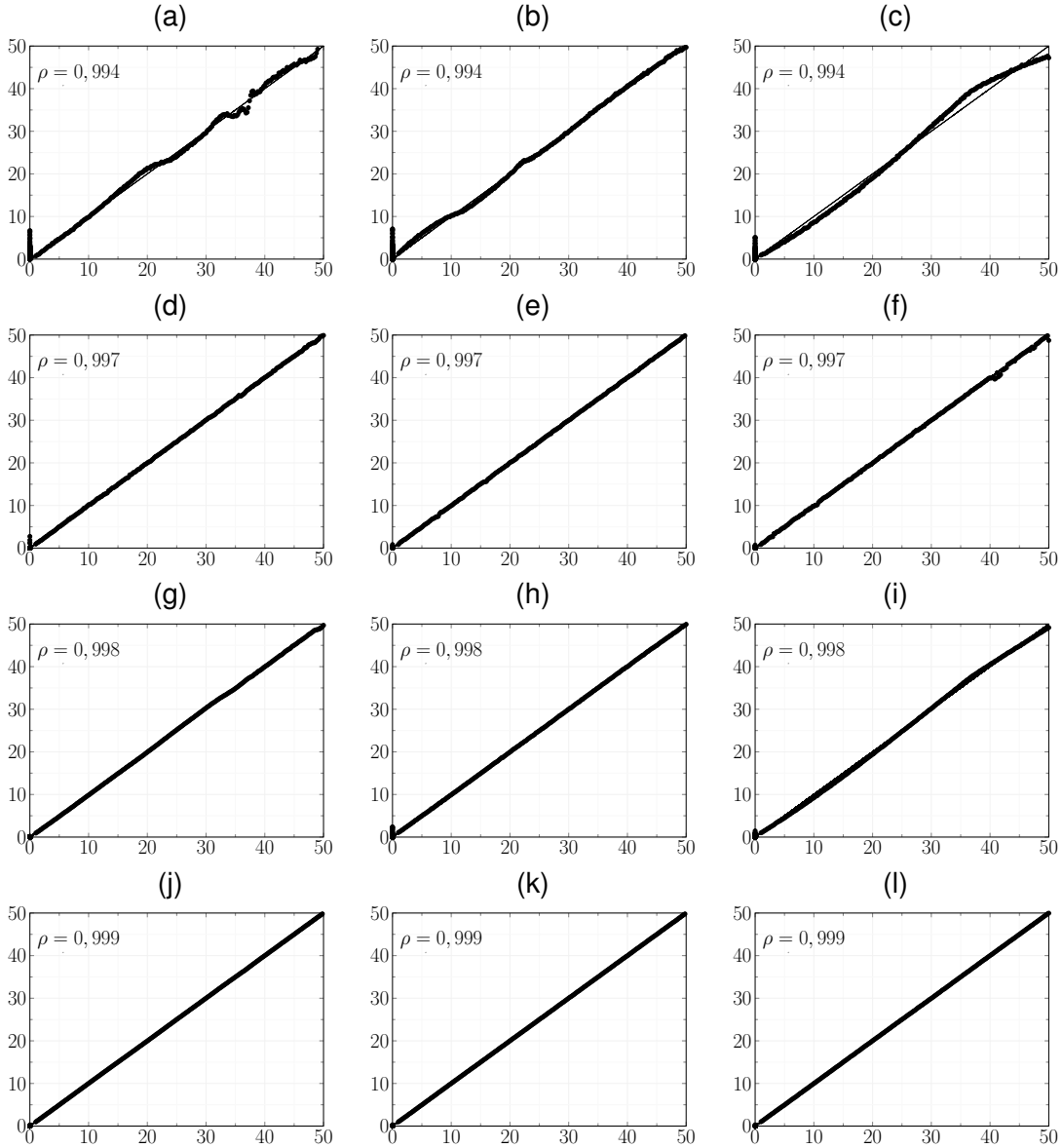


Figura 31 – Testes de regressão com nível AWGN $L = 60$ dB SNR: (a), (d), (g) e (j) para ΔP_r e regressores MLP, RF, SVM e GPRM , respectivamente; (b), (e), (h) e (k) para Δr_1 e regressores MLP, RF, SVM e GPR, respectivamente; e (c), (f), (i) e (l) para Δm_{c_1} e regressores MLP, RF, SVM e GPR, respectivamente. No gráfico acima, a linha preta representa a previsão perfeita e os pontos pretos representam os elementos verdadeiros (eixo x) versus elementos previstos (eixo y). Todos os gráficos possuem um intervalo entre zero e cinquenta (eixos x e y), ou seja, com amplitudes iguais às severidades da base de dados.

Através da análise da dispersão do RMSE após as etapas de validação cruzada, o GPRM obteve os resultados mais estáveis, mesmo com a influência de alto nível de ruído, mas com alto ToE para treinamento da base.

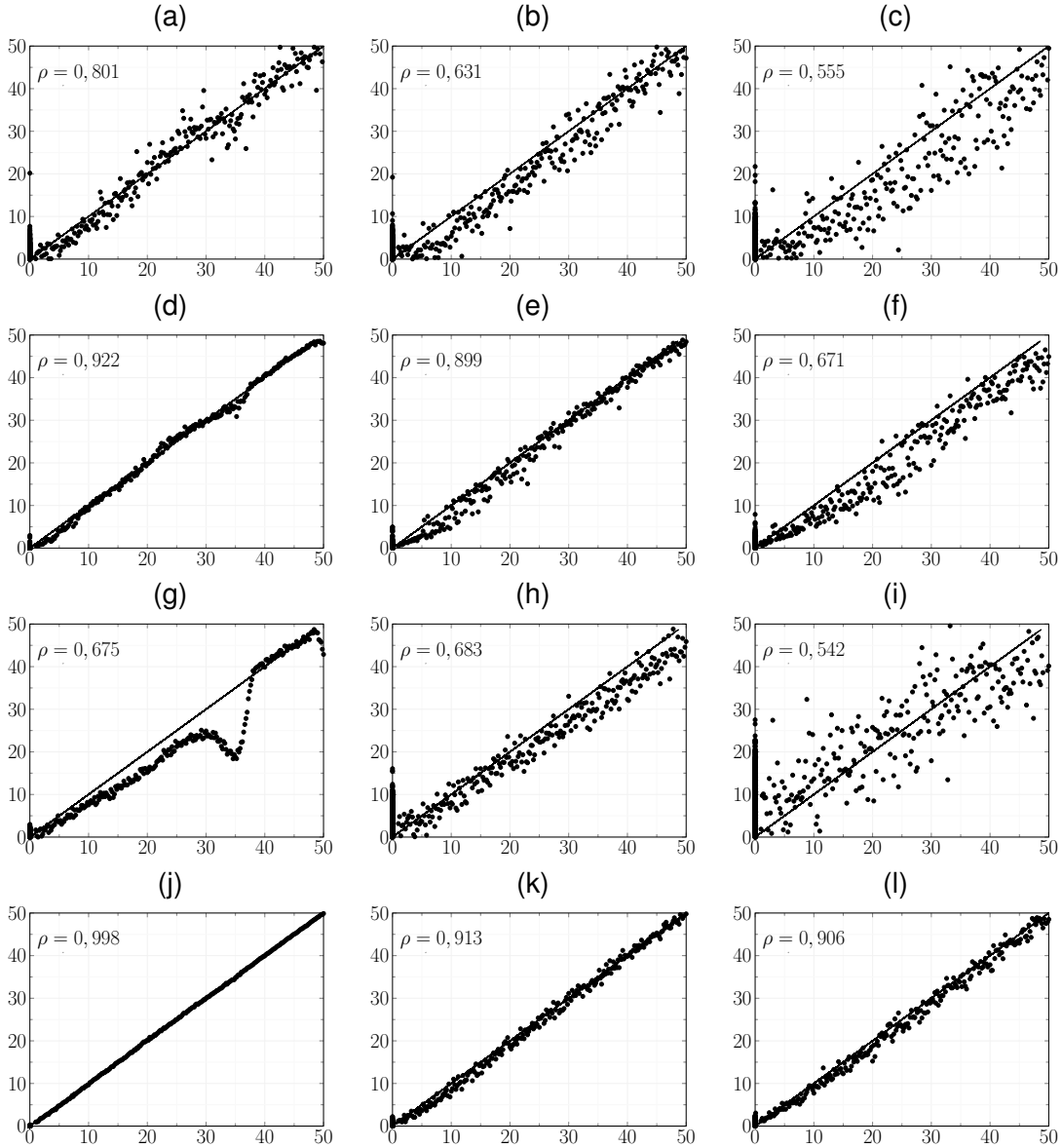


Figura 32 – Testes de regressão com nível AWGN $L = 0$ dB SNR.: (a), (d), (g) e (j) para ΔP_r e regressores MLP, RF, SVM e GPR , respectivamente; (b), (e), (h) e (k) para Δr_1 e regressores MLP, RF, SVM e GPR, respectivamente; e (c), (f), (i) e (l) para Δm_{c_1} e regressores MLP, RF, SVM e GPR, respectivamente. No gráfico acima, a linha preta representa a previsão perfeita e os pontos pretos representam os elementos verdadeiros (eixo x) versus elementos previstos (eixo y). Todos os gráficos possuem um intervalo entre zero a cinquenta (eixos x e y), ou seja, com amplitudes iguais às severidades da base de dados.

5.3.1 Análise Comparativa entre o Método de Levenberg-Marquardt e regressor por processo Gaussiano

A comparação entre o método de regressão Levenberg-Marquardt para mínimos quadrados (LMLS) (Gutiérrez, 2016) e o regressor GPRM foca na detecção e quantificação

da severidade de diferentes tipos de falha (normal, ΔP_a, Δr e Δm_c). Ambas as abordagens são aplicadas para identificar a localização da falha (em cilindros individuais ou todos os cilindros) e para combinar tipo, localização e severidade da falha, explorando aspectos qualitativos.

É fundamental destacar que o método LMLS pressupõe que o tipo de falha seja conhecido previamente pelo sistema, proporcionando uma medida numérica da severidade. Em contrapartida, o GPRM utiliza técnicas de regressão para estimar o tipo, a localização e a faixa de severidade da falha com base nos dados disponíveis.

A Tabela 13 descreve os cenários analisados, onde os casos de 1 a 6 abordam diferentes tipos de falha (normal, ΔP_a, Δr e Δm_c) com níveis específicos de severidade e condições de ruído. Por exemplo, o caso 1 simula exclusivamente a falha ΔP_a com 25% de severidade no cilindro 1, enquanto o caso 2 se concentra na falha Δr no mesmo cilindro com a mesma severidade. Os casos 7, 8 e 9 replicam os casos 1, 2 e 3, respectivamente, mas com adição de ruído, especificamente 15 dB de SNR em cada sinal de falha. Todos os casos utilizam um modelo dinâmico acoplado ao modelo termodinâmico para análise.

A Tabela 19 mostra os resultados comparativos entre as duas técnicas e o tempo de execução em cada análise.

Tabela 19 – Tabela de erro máximo na identificação da severidade de cada variável usando a técnica LMLS e GPRM

Caso	Metodologia						Tempo	
	LMLS			GPRM			LMLS (%) (horas)	GPRM (ms)
	ΔP_a	Δr	Δm_c	ΔP_a	Δr	Δm_c		
1	$< 10^{-5}$	0	0	$< 10^{-4}$	0	0	38	< 1
2	0	$< 10^{-5}$	0	0	$< 10^{-4}$	0	38	< 1
3	0	0	$< 10^{-5}$	0	0	$< 10^{-4}$	38	< 1
4	$< 10^{-4}$	0	0	$< 10^{-3}$	0	0	38	< 1
5	0	$< 10^{-4}$	0	0	$< 10^{-3}$	0	38	< 1
6	0	0	$< 10^{-4}$	0	0	$< 10^{-3}$	38	< 1

A Tabela 19 compara os resultados entre os métodos LMLS e GPRM para a identificação da severidade de diferentes variáveis. Em relação ao parâmetro ΔP_a, o método LMLS demonstra erros máximos inferiores a 10^{-5}, enquanto o GPRM apresenta erros na ordem de 10^{-4}. Para o parâmetro Δr, ambos os métodos alcançam erros máximos muito baixos, com LMLS e GPRM abaixo de 10^{-5}. Já para Δm_c, o LMLS reporta erros máximos inferiores a 10^{-5}, enquanto o GPRM tem erros da ordem de 10^{-4}.

Além das diferenças nos níveis de erro máximo, há uma disparidade significativa no tempo de execução entre os métodos. O LMLS demanda aproximadamente 38 horas para completar a análise, enquanto o GPRM executa em menos de 1 milissegundo. Essa análise sugere que o método LMLS oferece resultados mais precisos em termos de erro máximo para os parâmetros avaliados, porém requer um tempo de processamento substancialmente maior em comparação com o GPRM, que, apesar de menos preciso, é extremamente rápido.

Considerações finais

Este estudo propôs um *framework* quantitativo para análise de severidade composto por processamento de sinal e técnicas de regressão baseadas em aprendizado de máquina. Comparado com os métodos de classificação tradicionais relatados em muitos outros estudos, não visava identificar diferentes padrões de falha do motor Diesel, mas avaliava suas condições de saúde em processo de deterioração, avaliando o valor absoluto da severidade da falha.

O esquema proposto é mais significativo, na prática, pois a vida útil de um motor Diesel sofre um processo de deterioração antes que ocorra sua falha funcional. Conforme os resultados mencionados acima, o esquema proposto envolveu a extração de parâmetros estatísticos representativos dos sinais vindos do interior dos cilindros e a resposta em frequência do sinal de vibração torcional no volante do motor.

A função de tomada de decisão proposta para a predição do nível de severidade com base em técnicas de aprendizado de máquina, atrelada ao tipo de variável operacional da máquina, apresenta pontos positivos em relação aos métodos numéricos tradicionais, por haver uma vantagem na velocidade de a predição da severidade, uma vez que o regressor já está treinado, pois nos métodos numéricos tradicionais há sempre uma necessidade prévia de convergência para estimar o nível de severidade.

Além disso, o SVM obteve bons resultados com baixo nível de ruído, mas com o crescimento desse nível, seu desempenho piorou, sendo necessário um ajuste específico para cada nível de ruído.

O processo de ajuste dos hiperparâmetros dos regressores, combinado com estatísticas de segunda ordem, foi fundamental para a obtenção dos melhores parâmetros dos regressores, para o tipo de análise abordada neste trabalho.

A análise de regressão com quatro níveis diferentes de ruído branco foi extremamente importante, ao ser possível avaliar a robustez de cada regressor, dado o aumento intrínseco do nível de ruído da medição e assim definir qual é o melhor regressor para medições com baixo sinal relação ruído/ruído.

O GPRM foi o regressor mais estável em relação aos quatro diferentes níveis de ruído abordados e apresentou o menor RMSE para as análises realizadas, além de ter

a menor dispersão entre os resultados da validação cruzada. Isso o torna a escolha mais indicada para processos de avaliação de severidade em máquinas Diesel com características semelhantes às deste trabalho e com níveis de ruído entre 0 e 60 dB. No entanto, nas análises onde o ruído do sinal é maior que 15 dB e menor que 0 dB, a RF apresentou resultados próximos ao GPRM, com ToE 10 vezes menor, mas com dispersão 10 vezes maior entre as realizações da validação cruzada.

Este trabalho propôs uma modificação na técnica de identificação de falhas em comparação com a adotada em (Gutiérrez, 2016), diferindo pela utilização da técnica de aprendizado de máquina. Para avaliar essa abordagem, um novo banco de dados com 3500 cenários de falhas foi desenvolvido. O vetor de características proposto aplicado ao problema de reconhecimento de 4 classes (normal, redução de pressão no coletor de admissão, redução da razão de compressão e redução da quantidade de combustível injetado) descrito em (Gutiérrez, 2016) e (Gutiérrez et al., 2018) atingiu a desempenho máximo do sistema de 99,3%.

É apresentada uma revisão dos métodos de análise de motores a Diesel, através do modelo termodinâmico, acoplado ao modelo dinâmico com 11 graus de liberdade. Uma categorização de tais métodos de análise foi descrita juntamente com uma linha do tempo obtida através do estudo da dinâmica do cilindro e do mapeamento do modelo de torção da árvore de manivelas. Além disso, foi desenvolvida uma estrutura sistemática para comparar os métodos numéricos e de aprendizado de máquina para previsão de falhas e severidade de motores a Diesel. Essa estrutura foi implementada, validada, simulada e testada na plataforma do Matlab.

Mesmo em uma tarefa de discriminação mais complexa, como no problema de 14 classes, onde a razão de compressão e a quantidade de redução de combustível injetado são divididas em falhas individuais no cilindro e novas categorias de falhas são adicionadas, o vetor de características proposto aplicado ao sistema de classificação alcançou 99,3% de precisão geral na maior razão sinal-ruído. O decréscimo da relação sinal-ruído (30, 15 e 0 dB) diminui o desempenho geral da classificação, o que em sistemas reais deve ser levado em consideração, ou seja, o uso de meios que mantenham a relação de ruído a maior possível, pois o ruído prejudica o desempenho da classificação.

O tempo total de execução para reproduzir todos os 3500 cenários de falha da base de dados **3500-DEFault**, após as modificações nos algoritmos do trabalho de (Gutiérrez, 2016) e detalhadas na seção Apêndices desta tese, foi de 36 horas. Essas modificações

foram implementadas para melhorar a eficiência do código e a precisão na análise e simulação dos cenários de falha, refletindo um avanço significativo no processamento e na interpretação dos resultados obtidos e a viabilidade da implementação de 3500 cenários de falha.

Trabalhos futuros

Em trabalhos futuros, pretende-se melhorar as vantagens das abordagens propostas, possivelmente incluindo a etapa de regressão de aprendizado de máquina para estimar numericamente o nível de severidade e avaliar sinais corrompidos por várias falhas.

Uma expansão da base **3500-DEFault**, será implementada em trabalhos futuros, para novos cenários de falhas e com diferentes faixas de rotação, visando emular uma gama maior de eventos de falha, abordando, inclusive, falhas combinadas.

Futuros trabalhos poderão abordar a comparação dos sinais da base com sinais de uma máquina real, na mesma classe de potência e número de cilindros, de sorte que o modelo desenvolvido represente o mais próximo possível as características de um sinal real, ou pequenos ajustes deverão ser realizados no modelo, para ajustar possíveis divergências entre os sinais.

Outros recursos serão incorporados para melhorar os resultados da regressão para a falha de combustível na condição de ruído alto.

Produção científica

O Artigo (Pestana-Viana et al., 2018) foi publicado nos anais da 10ª Conferência Internacional sobre Dinâmica dos Rotores (IFToMM), realizada no Rio de Janeiro, Brasil. Apresenta um diagnóstico de falhas em motores a diesel para auxiliar na manutenção preditiva, através da análise da variação das curvas de pressão no interior dos cilindros e da resposta de vibração torcional da árvore de manivelas. Portanto, um modelo

de simulação de falhas baseado em um modelo termodinâmico de dimensão zero foi desenvolvido. Os vetores de características adotados foram escolhidos a partir do modelo termodinâmico e obtidos a partir de sinais de processamento como pressão e temperatura no interior do cilindro, bem como vibração torcional do volante do motor. Esses vetores foram usados como entrada da técnica de aprendizado de máquina, a fim de discriminar as condições operativas da máquina. Os resultados mostraram que o classificador mais adequado foi o *Random Forest*, para este tipo de problema.

O Artigo (Viana et al., 2023) foi publicado na revista *Machines* em 2023, no volume 11, edição 5, páginas 530-548. O objetivo do artigo é propor um método para realizar prognósticos de falhas em motores a diesel usando métodos de regressão de inteligência artificial. O método proposto utiliza redes neurais do tipo *perceptron* de múltiplas camadas e *Random Forest* para avaliar a severidade das falhas em motores a diesel.

Os autores implementaram um banco de dados com 3500 cenários de falha para superar o desafio de induzir falhas destrutivas em motores a diesel. Os sinais de falha do motor Diesel foram desenvolvidos usando um modelo termodinâmico de zero dimensões em um cilindro acoplado a um modelo de vibração torcional da árvore de manivelas. Redes neurais artificiais e modelos de regressão *Random Forest* foram empregados para classificar e quantificar as falhas.

A metodologia proposta foi aplicada juntamente com um simulador de motor para avaliar sua eficácia e precisão. Os resultados mostraram que o melhor desempenho foi obtido com o regressor *Random Forest*, com um valor de RMSE (erro médio quadrático) de 0,10 ± 0,03%.

Em resumo, o artigo descreve um método para prever falhas em motores a diesel usando técnicas de regressão de inteligência artificial. O método proposto demonstrou ser eficaz na classificação e quantificação das falhas, contribuindo para a manutenção preditiva e redução de custos de manutenção em motores a diesel.

O Artigo (Martins et al., 2023) foi publicado na revista *Engineering Applications of Artificial Intelligence*, no Volume 124, em setembro de 2023. O artigo aborda o reconhecimento de falhas combinadas de desequilíbrio e desalinhamento em máquinas rotativas, bem como a quantificação da severidade dessas falhas. É proposta uma técnica chamada *Improved Variable Mode Decomposition* (IVMD) para decompor falhas combinadas de desequilíbrio e desalinhamento, que são as mais frequentes na indústria.

O sistema de detecção de falhas proposto combina um classificador capaz de

distinguir falhas isoladas de desequilíbrio e desalinhamento de falhas combinadas, independentemente da velocidade de rotação ou nível de severidade da falha, com o IVMD. O classificador utiliza o método de *Support Vector Machine* (SVM) aplicado às características obtidas a partir da assinatura de vibração. O IVMD é aplicado para decompor as falhas quando necessário.

Além disso, é estimado um nível contínuo de severidade com base na velocidade do sinal de vibração, permitindo observar a contribuição de cada falha individualmente para o nível geral de vibração. Os resultados obtidos demonstram que o algoritmo baseado em SVM apresenta uma precisão de 95,43% na identificação de falhas isoladas de desequilíbrio e desalinhamento, bem como de falhas combinadas, e a abordagem IVMD separa com precisão as falhas.

O artigo (Martins et al., 2021a) foi publicado no periódico *The International Journal of Advanced Manufacturing Technology* pela editora *Springer* em abril de 2021. O objetivo do artigo é propor um método automático de diagnóstico e medição de falhas combinadas em máquinas rotativas. Foi desenvolvido um sistema de Internet das Coisas (IoT) capaz de identificar falhas combinadas em uma máquina rotativa e prever falhas de maneira não invasiva. Foi criada uma solução embarcada que classifica quatro tipos de condições de operação: normal, desbalanceada, desbalanceada associada ao desalinhamento horizontal e desbalanceada associada ao desalinhamento vertical. O artigo utiliza uma metodologia que combina um banco de simulação e medição de severidade em um ambiente controlado.

Três técnicas distintas de aprendizado de máquina foram comparadas para fins de classificação: máquinas de vetores de suporte, k-vizinhos mais próximos e *Random Forest*. Os resultados obtidos mostram a possibilidade de diferenciar os tipos de falhas combinadas, alcançando uma precisão de 81,41% com um classificador *Random Forest*. Foi desenvolvido um sistema de supervisão responsável por monitorar as máquinas e enviar mensagens de alerta sem fio. Essas mensagens são enviadas para um aplicativo de controle, permitindo a interação do usuário por meio de dispositivos móveis.

Os resultados revelam a possibilidade de diferenciar os tipos de falhas combinadas e também o perfil de severidade de falha do motor para diferentes cenários. Por meio da construção de perfis de severidade, foram registrados valores de vibração elevados em altas velocidades quando ocorriam falhas. A metodologia proposta pode ser utilizada em qualquer máquina rotativa que esteja em conformidade com as condições impostas pela

norma ISO 10816.

O artigo (Martins et al., 2021b) foi publicado na conferência *26th ABCM International Congress of Mechanical Engineering*, 2021, Florianópolis. O objetivo principal do artigo é apresentar um método inovador de extração de características para o diagnóstico de falhas em máquinas rotativas. O método proposto utiliza análise sinusoidal, que envolve várias técnicas tradicionalmente empregadas no processamento de sinais de áudio. Este artigo é pioneiro no uso dessa família de métodos para abordar problemas de reconhecimento de padrões encontrados em máquinas rotativas industriais.

O artigo tem como objetivo distinguir entre falhas isoladas, caracterizadas apenas por desequilíbrio, desalinhamento vertical ou desalinhamento horizontal, e falhas combinadas, que envolvem desequilíbrio associado tanto ao desalinhamento horizontal quanto ao vertical.

No estudo, os sinais de vibração são analisados utilizando uma abordagem sinusoidal, que visa representar esses sinais por meio de um conjunto finito de componentes definidos por amplitude, fase e frequência, os quais variam ao longo do tempo. Após calcular essa representação tempo-frequência, o método extrai as faixas principais do sinal, que encapsulam as informações mais relevantes. Diversas características são derivadas dessas faixas, como frequência de início, comprimento da maior faixa, energia associada à maior faixa e variância das frequências iniciais.

Os experimentos conduzidos demonstraram que essas características sinusoidais são úteis para a classificação de falhas. Utilizando um conjunto de dados específico, foi alcançada uma precisão global de 86,35% ao combinar essas características com o método de seleção de características conhecido como *Out Of Bag Feature Importance* (OOBFI) e o algoritmo *Random Forest* para o reconhecimento de padrões.

Além disso, o artigo investiga a seleção do número ótimo de características e o melhor algoritmo de classificação capaz de distinguir padrões de falhas por meio do reconhecimento de características sinusoidais.

Referências Bibliográficas

Abdulaziz, A. A. and Khan, A. (2014). Cycle-by-cycle variations of si engine and analysis of the effect of various parameters. International Journal of Automotive Technology, 15(3):373–382.

Abramowitz, M. and Stegun, I. (2012). Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables. Dover Books on Mathematics. Dover Publications.

Abuella, M. and Chowdhury, B. (2017). Random forest ensemble of support vector regression models for solar power forecasting. In 2017 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT), pages 1–5.

Aggarwal, C. (2018). Neural Networks and Deep Learning: A Textbook. Springer.

Akpinar, M. and Yumusak, N. (2016). Year ahead demand forecast of city natural gas using seasonal time series methods. Energies, 9(9).

Altmann, A., Toloşi, L., Sander, O., and Lengauer, T. (2010). Permutation importance: a corrected feature importance measure. Bioinformatics, 26(10):1340–1347.

Arcoumanis, C. and Kamimoto, T. (2009). Flow and Combustion in Reciprocating Engines. Springer.

Arora, A., Lin, J.-J., Gasperian, A., Maldjian, J., Stein, J., Kahana, M., and Lega, B. (2018). Comparison of logistic regression, support vector machines, and deep learning classifiers for predicting memory encoding success using human intracranial EEG recordings. Journal of Neural Engineering, 15(6):066028.

Bartoń, K. (2018). MuMIn: Multi-Model Inference.

Breiman, L. (1996). Bagging predictors. Machine learning, 24(2):123–140.

Breiman, L. (2001). Random forests, machine learning 45. J. Clin. Microbiol, 2(30):199–228.

Bui, K. Q., Perera, L. P., and Emblemsvåg, J. (2022). Life-cycle cost analysis of an innovative marine dual-fuel engine under uncertainties. Journal of Cleaner Production, 380:134847.

Charles, P., Sinha, J. K., Gu, F., Lidstone, L., and Ball, A. D. (2009). Detecting the crankshaft torsional vibration of diesel engines for combustion related diagnosis. Journal of sound and vibration, 321(3-5):1171–1185.

Chow, M.-Y. (2000). Guest editorial special section on motor fault detection and diagnosis. IEEE Transactions on Industrial Electronics, 47(5):982–983.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. Machine learning, 20:273–297.

Davydenko, A. and Fildes, R. (2013). Measuring Forecasting Accuracy: The Case Of Judgmental Adjustments To Sku-Level Demand Forecasts. International Journal of Forecasting, 29(3):510–522.

de Lima, A. A., d. M. Prego, T., Netto, S. L., da Silva, E. A. B., Gutierrez, R. H. R., Monteiro, U. A., Troyman, A. C. R., d. C. Silveira, F. J., and Vaz, L. (2013). On fault classification in rotating machines using fourier domain features and neural networks. In 2013 IEEE 4th Latin American Symposium on Circuits and Systems (LASCAS), pages 1–4.

de SS Martins, D. H., Hemerly, D. O., Marins, M., Lima, A. A., Silva, F. L., Prego, T. d. M., Ribeiro, F. M., Netto, S. L., and da Silva, E. A. (2018). Application of machine learning to evaluate unbalance severity in rotating machines. In International Conference on Rotor Dynamics, pages 144–160. Springer.

Deiss, L., Margenot, A. J., Culman, S. W., and Demyan, M. S. (2020). Tuning support vector machines regression models improves prediction accuracy of soil properties in mir spectroscopy. Geoderma, 365:114227.

Dudek, G. (2016). Neural networks for pattern-based short-term load forecasting: A comparative study. Neurocomputing, 205:64–74.

Feng, Q., Jiang, P., Lee, J., Zhou, Q., and Zhang, D. (2020). Digital twin-enabled prognostics and health management: A comprehensive review. IEEE Transactions on Industrial Informatics, 16(1):486–497.

Genuer, R., Poggi, J.-M., Tuleau-Malot, C., and Villa-Vialaneix, N. (2017). Random forests for big data. Big Data Research, 9:28–46.

Gil-Alana, L. A. (2005). Measuring the memory parameter on several transformations of asset returns. International Journal of Theoretical and Applied Finance, 8(6):675–691.

Guerrero, D. P. and Jimanez-Espadafor, F. J. (2019). Torsional system dynamics of low speed diesel engines based on instantaneous torque: Application to engine diagnosis. Mechanical Systems and Signal Processing, 116:858 – 878.

Gutiérrez, R. H. R., Belchior, C. R. P., Vaz, L. A., and Monteiro, U. A. (2018). Diagnostic methodology in four-stroke marine diesel engine by identifying operational parameters. Journal of Brazilian Society of Mechanical Science and Engineering, 40(500):1–10.

Gutiérrez, R. H. R. (2016). Simulação e Identificação de Falhas de Motores Diesel. Tese de doutorado, Universidade Federal do Rio de Janeiro, Rio de Janeiro - Brasil.

Haykin, S. (1998). Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition.

Haykin, S. O. (2008). Neural Networks and Learning Machines. Prentice Hall, New York, 3 edition edition.

Haykin, S. S. (2009). Neural networks and learning machines. Pearson Education, Upper Saddle River, NJ, third edition.

Heywood, J. (1988). Internal Combustion Engine Fundamentals. McGraw-Hill, New York, USA, 1st edition.

Hountalas, D. and Kouremenos, A. (1999). Development and application of a fully automatic troubleshooting method for large marine diesel engines. Applied Thermal Engineering, 19(3):299 – 324.

Hountalas, D. T. (2000). Prediction of marine diesel engine performance under fault conditions. Applied thermal engineering, 20(18):1753–1783.

Hung, C.-C., Song, E., and Lan, Y. (2019). Foundation of deep machine learning in neural networks. Springer, pages 201–232.

Jadhav, R., Dhabe, P., Gandewar, S., Mirani, P., and Chugwani, R. (2020). A new data structure for representation of relational databases for application in the normalization process. In Swain, D., Pattnaik, P. K., and Gupta, P. K., editors, Machine Learning and Information Processing, pages 305–316, Singapore. Springer Singapore.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2014). An Introduction to Statistical Learning: With Applications in R. Springer Publishing Company, Incorporated.

Kavitha S, Varuna S, and Ramya R (2016). A comparative analysis on linear regression and support vector regression. In 2016 Online International Conference on Green Engineering and Technologies (IC-GET), pages 1–5.

Keerthi, S. S. (2002). Efficient tuning of svm hyperparameters using radius/margin bound and iterative algorithms. IEEE Transactions on Neural Networks, 13(5):1225–1229.

Klinchaeam, S., Nivesrangsan, P., and Lokitsangthong, M. (2009). Condition monitoring of a small four-stroke petrol engine using vibration signals. Current Applied Science and Technology, 9(1):9–17.

Lamaris, V. and Hountalas, D. (2010). A general purpose diagnostic technique for marine diesel engines–application on the main propulsion and auxiliary diesel units of a marine vessel. Energy conversion and management, 51(4):740–753.

Larose, D. T. (2015). Data mining and predictive analytics. John Wiley & Sons.

Lee, J., Bagheri, B., and Kao, H.-A. (2020). Digital twin-driven prognostics and health management: A review. Robotics and Computer-Integrated Manufacturing, 63:101896.

Li, X., Tao, F., and Cheng, W. (2020). Digital twin: The convergence of data and intelligence. IEEE Intelligent Systems, 35(6):87–92.

Li, X., Yang, X., Yang, Y., Bennett, I., and Mba, D. (2019). A novel diagnostic and prognostic framework for incipient fault detection and remaining service life prediction with application to industrial rotating machines. Applied Soft Computing, 82:105564.

Li, X., Zhang, W., Ding, Q., and Li, X. (2020). Diagnosing rotating machines with weakly supervised data using deep transfer learning. IEEE Transactions on Industrial Informatics, 16(3):1688–1697.

Maier, H. R. and Dandy, G. C. (2001). Neural network based modelling of environmental variables: A systematic approach. Mathematical and Computer Modelling, 33(6–7):669–682.

Martins, D. H., de Lima, A. A., Gutierrez, R. H., Pestana-Viana, D., Netto, S. L., Vaz, L. A., da Silva, E. A., and Haddad, D. B. (2023). Improved variational mode decomposition for combined imbalance-and-misalignment fault recognition and severity quantification. Engineering Applications of Artificial Intelligence, 124:106516.

Martins, D. H. C. D. S., Viana, D., Lima, A., Faria Pinto, M., Tarrataca, L., Silva, F., Gutierrez, R., Prego, T., Monteiro, U., and Haddad, D. (2021a). Diagnostic and severity analysis of combined failures composed by imbalance and misalignment in rotating machines. The International Journal of Advanced Manufacturing Technology, 114.

Martins, D. H. C. S. S., Lima, A. A., Pinto, M. F., Pestana-Viana, D., Haddad, D. B., and Monteiro, U. A. B. V. (2021b). Application of sinusoidal analysis to feature extraction in rotating machine vibration signals. In 26th ABCM International Congress of Mechanical Engineering, Florianópolis. Artificial Intelligent Applications.

Mathew, J., Luo, M., and Pang, C. K. (2017). Regression kernel for prognostics with support vector machines. In 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pages 1–5.

Mathur, S., Waswani, H., Singh, D., and Ranjan, R. (2022). Alternative fuels for agriculture sustainability: Carbon footprint and economic feasibility. AgriEngineering, 4(4):993–1015.

Mendes, A., Meirelles, P., and Zampieri, D. (2008). Analysis of torsional vibration in internal combustion engines: modelling and experimental validation. Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics, 222(2):22–25.

Merker, G. P., Schwarz, C., and Teichmann, R. (2011). Combustion engines development: mixture formation, combustion, emissions and simulation. Springer Science & Business Media.

Mirjalili, S., Faris, H., and Aljarah, I. (2020). Evolutionary Machine Learning Techniques Algorithms and Applications: Algorithms and Applications. Springer.

Mustafa, A., Rienow, A., Saadi, I., Cools, M., and Teller, J. (2018). Comparing support vector machines with logistic regression for calibrating cellular automata land use change models. European Journal of Remote Sensing, 51(1):391–401.

Park, J., Hamadache, M., Ha, J. M., Kim, Y., Na, K., and Youn, B. D. (2019). A positive energy residual (per) based planetary gear fault detection method under variable speed conditions. Mechanical Systems and Signal Processing, 117:347 – 360.

Patel, M. B., Parikh, S. M., and Patel, A. R. (2020). Global normalization for fingerprint image enhancement. In Smys, S., Tavares, J. M. R. S., Balas, V. E., and Iliyasu, A. M., editors, Computational Vision and Bio-Inspired Computing, pages 1059–1066, Cham. Springer International Publishing.

Pestana-Viana, D. (2021). Diesel engine faults features dataset (3500-default). https://dx.doi.org/10.21227/3q67-j268.

Pestana-Viana, D., Gutiérrez, R. H., de Lima, A. A., e Silva, F. L., Vaz, L., Prego, T. d. M., and Monteiro, U. A. (2018). Application of machine learning in diesel engines fault identification. In International Conference on Rotor Dynamics, pages 74–89. Springer.

Pestana-Viana, D., Gutiérrez, R. H. R., de Lima, A. A., e Silva, F. L., Vaz, L., de M. Prego, T., and Monteiro, U. A. (2019). Application of machine learning in diesel engines fault identification. In Cavalca, K. L. and Weber, H. I., editors, Proceedings of the 10th International Conference on Rotor Dynamics – IFToMM, pages 74–89, Cham. Springer International Publishing.

Pestana-Viana, D., Zambrano-López, R., de Lima, A. A., d. M. Prego, T., Netto, S. L., and da Silva, E. A. B. (2016). The influence of feature vector on the classification of mechanical faults using neural networks. In 2016 IEEE 7th Latin American Symposium on Circuits Systems (LASCAS), pages 115–118.

Rasmussen, C. E. and Williams, C. K. I. (2005). Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press.

Rokach, L. (2019). Ensemble learning: pattern classification using ensemble methods. World Scientific.

Rubio, J. A. P., Vera-García, F., Grau, J. H., Cámara, J. M., and Hernandez, D. A. (2018). Marine diesel engine failure simulator based on thermodynamic model. Applied Thermal Engineering, 144:982–995.

Samarasinghe, S. (2016). Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition. Auerbach publications.

Schlkopf, B., Smola, A. J., and Bach, F. (2018). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. The MIT Press.

Smola, A. J. and Schökopf, B. (2000). Sparse greedy matrix approximation for machine learning. In Proceedings of the Seventeenth International Conference on Machine Learning, ICML '00, page 911–918, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Solmaz, H. and Karabulut, H. (2015). A mathematical model to investigate the effects of misfire and cyclic variations on crankshaft speed fluctuations in internal combustion engines. Journal of Mechanical Science and Technology, 29(4):1493–1500.

Soto, T. (2021). Regression analysis. In Encyclopedia of Autism Spectrum Disorders, pages 3906–3906. Springer.

Stockwell, R., Mansinha, L., and Lowe, R. (1996). Localization of the complex spectrum: The s transform. IEEE Transactions on Signal Processing, 44(4):998 – 1001. Cited by: 2854.

Viana, D. P., de Sá Só Martins, D. H. C., de Lima, A. A., Silva, F., Pinto, M. F., Gutierrez, R. H. R., Monteiro, U. A., Vaz, L. A., Prego, T., Andrade, F. A. A., Tarrataca, L., and Haddad, D. B. (2023). Diesel engine fault prediction using artificial intelligence regression methods. Machines, 11(5).

Wang, J., Du, G., Zhu, Z., Shen, C., and He, Q. (2020). Fault diagnosis of rotating machines based on the emd manifold. Mechanical Systems and Signal Processing, 135:106443.

Wang, W. and Lu, Y. (2018). Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model. IOP Conference Series: Materials Science and Engineering, 324:012049.

Willmott, C. J. and Matsuura, K. (2006). On the use of dimensioned measures of error to evaluate the performance of spatial interpolators. International Journal of Geographical Information Science, 20(1):89–102.

Xi, W., Li, Z., Tian, Z., and Duan, Z. (2018). A feature extraction and visualization method for fault detection of marine diesel engines. Measurement, 116:429–437.

Xin, Q. (2011a). Diesel engine system design. Elsevier.

Xin, Q. H. (2011b). Overview of diesel engine applications for engine system design-part 1: Systems engineering and rational considerations of product r&d organization design. Technical report, SAE Technical Paper.

Xu, J., Xu, C., Zou, B., Tang, Y. Y., Peng, J., and You, X. (2019). New incremental learning algorithm with support vector machines. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49(11):2230–2241.

Yadav, S. and Shukla, S. (2016). Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In 2016 IEEE 6th International conference on advanced computing (IACC), pages 78–83. IEEE.

Yan, X., Zhao, C., Lu, Z., Zhou, X., and Xiao, H. (2005). A study of information technology used in oil monitoring. Tribology International, 38(10):879–886.

Yang, B.-S., Di, X., and Han, T. (2008). Random forests classifier for machine fault diagnosis. Journal of mechanical science and technology, 22(9):1716–1725.

Yang, J., Pu, L., Wang, Z., Zhou, Y., and Yan, X. (2001). Fault detection in a diesel engine by analysing the instantaneous angular speed. Mechanical systems and signal processing, 15(3):549–564.

Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M. J., Ghodsi, A., Gonzalez, J., Shenker, S., and Stoica, I. (2016). Apache spark: A unified engine for big data processing. Commun. ACM, 59(11):56–65.

Zannis, T. C. and Hountalas, D. T. (2005). Experimental investigation of the effect of intake valve timing on the performance and emissions of a di diesel engine. Energy Conversion and Management, 46(3):395–409.

Zhang, L. and Hu, J. (2013). Safety prognostic technology in complex petroleum engineering systems: progress, challenges and emerging trends. Petroleum Science, 10(4):486–493.

Apêndices

A- Algoritmos do programa DEFault

A.1- Função $\mathbf{pressao.cal}$

A função $\mathbf{pressao.cal}$ recebe um conjunto de parâmetros de entrada: N, taf, DP, DPr, DTp, $fcomp$, far, $fcomb$, e $finj$. Esses parâmetros são utilizados para calcular a pressão em um determinado sistema.

A função começa definindo uma matriz chamada parâmetros, que contém uma lista de valores de parâmetros. Em seguida, há um $loop$ que percorre os valores de N na matriz parâmetros e, quando encontra uma correspondência, atribui os valores relacionados a p.

Em seguida, são definidos alguns outros parâmetros, como o tempo inicial $t0$, tempo final tf, intervalo de tempo dt e o número de iterações n. Após, calcula-se um vetor inicial $Y0$ com base em DP e nos valores correspondentes em p.

O algoritmo continua com um $loop$ por n iterações. Em cada iteração, são calculados os valores de $k1$, $k2$, $k3$ e $k4$ usando uma função chamada $\mathbf{cilindropt009}$. Esses valores são usados para atualizar o tempo t e o vetor Y. Além disso, uma matriz sw é atualizada com os valores de t e Y.

No final do algoritmo, o resultado PT é retornado com os valores da matriz sw.

A.2- Subfunção $\mathbf{cilindropt009}$ de $\mathbf{pressao.cal}$

O pseudocódigo apresentado descreve um algoritmo chamado $\mathbf{cilindropt009}$. Esse algoritmo recebe uma série de parâmetros de entrada, representados por t, y, p, taf, DP, DPr, DTp, $fcomp$, far, $fcomb$, e $finj$. O resultado do algoritmo é um vetor chamado DY.

Primeiramente, o valor da constante f é definida como o resultado da divisão de π por 180, sendo usada para converter ângulos de graus para radianos.

Algoritmo 2 – Função **pressao.cal**

Entrada: $N, taf, DP, DPr, DTp, fcomp, far, fcomb, finj$
Saída: $PT = \mathbf{pressao.cal}(Entrada)$

início
  $parametros = \begin{bmatrix} 1000 & 150000 & 316.4 & 20 & 0.88 \\ 1200 & 170000 & 317.9 & 16 & 0.85 \\ 1300 & 170000 & 317.8 & 16 & 0.85 \\ 1600 & 190000 & 320.3 & 16 & 0.80 \\ 1900 & 220000 & 320.9 & 16 & 0.70 \\ 2100 & 220000 & 321.1 & 15 & 0.65 \\ 2300 & 220000 & 322.8 & 16 & 0.6 \\ 2500 & 220000 & 323.1 & 16 & 0.6 \end{bmatrix}$;
  para $i = 1$ **até** 8 **faça**
    se $N == parametros(i, 1)$ **então**
      $p = parametros(i, :)$
    fim
  fim
  $t0 = 0$;
  $tf = 720$;
  $dt = 1$;
  $n = 720/dt$;
  $Y0 = [\frac{(100-DP)}{100} \times p(2), p(3)]^T$;
  $t = t0$;
  $Y = Y0$;
  $sw = [0, Y']$;
  para $k = 1$ **até** n **faça**
    $k1 = dt \times$ **cilindropt009**$(t, Y, p, taf, DP, DPr, DTp, fcomp, far, fcomb, finj)$;
    $k2 = dt \times$ **cilindropt009**$(t + dt/2, Y + k1/2, p, taf, DP, DPr, DTp, fcomp, far, fcomb, finj)$;
    $k3 = dt \times$ **cilindropt009**$(t + dt/2, Y + k2/2, p, taf, DP, DPr, DTp, fcomp, far, fcomb, finj)$;
    $k4 = dt \times$ **cilindropt009**$(t + dt, Y + k3, p, taf, DP, DPr, DTp, fcomp, far, fcomb, finj)$;
    $t = t + dt$;
    $Y = Y + (k1 + 2 \times k2 + 2 \times k3 + k4)/6$;
    $sw = [sw; t, Y']$;
  fim
fim
retorna $PT = sw$

Em seguida, uma série de cálculos é realizada para obter os valores das variáveis $S, A, V,$ e dV. É importante observar que esses cálculos dependem de valores das variáveis de entrada, como $R, L, D,$ e r.

O algoritmo continua com cálculos adicionais envolvendo as variáveis $Sa, Va, Pot,$ e Ce. A variável Pot é calculada usando uma função externa chamada $\mathbf{potencia}$ e a variável Ce usando uma função externa chamada $\mathbf{consumo}$. Em seguida, são realizados cálculos para obter os valores das variáveis $Mc, Ra,$ e Ma. A função externa $\mathbf{fracoes}$ é utilizada para calcular o valor da variável l.

São realizados mais cálculos para determinar os valores das variáveis $Pr, dt, ti, Dtp,$ e Dt. O algoritmo utiliza uma estrutura condicional para atribuir valores às variáveis $xTr, xOr, xNr, Mm, x,$ e dx com base no valor da variável t.

Por fim, o vetor DY é retornado como resultado do algoritmo.

A.3- Função $\mathbf{dif.volume}$

O pseudocódigo apresentado descreve uma função chamada $\mathbf{dif.volume}$ utilizado para calcular o volume e a derivada do volume de um cilindro em função da posição do pistão. A função recebe como entrada um vetor T contendo valores em graus que representam a posição do pistão ao longo do tempo. O resultado da função é um vetor $[V, Vd]$ contendo o volume e a derivada do volume, respectivamente.

Inicializam-se as variáveis D, R, L e r, que representam, respectivamente, o diâmetro do cilindro, o raio da manivela, o comprimento da biela e a razão de compressão.

Calcula-se o número de elementos do vetor T e armazena-se esse valor na variável n.

Inicializam-se os vetores sw e sp como vetores vazios. Esses vetores serão utilizados para armazenar os resultados intermediários dos cálculos.

Inicia-se um $loop$ que percorre cada elemento do vetor T, tal que cada elemento t de T, realiza-se os seguintes cálculos: a constante f como o valor de π dividido por 180; S como a soma de R multiplicado pelo cosseno de $f \cdot t$ e a raiz quadrada de $L^2 - R^2 \cdot (\sin(f \cdot t))^2$; A como o produto de π, D e a soma de $D/2 + L + R + 2 \cdot R/(r-1) - S$; v como o produto de π, D^2 e a soma de $(L + R + 2 \cdot R/(r-1) - S)/4$, acrescentando-se

Algoritmo 3 – Subfunção **cilindropt009**

Entrada: $t, y, p, taf, DP, DPr, DTp, fcomp, far, fcomb, finj$
Saída: $DY = \mathbf{cilindropt009}(Entrada)$

Inicio
- $f = \pi/180$
- $S = R \cdot \cos(f \cdot t) + (L^2 - R^2 \cdot (\sin(f \cdot t))^2)^{0.5}$
- $A = \pi \cdot D \cdot (D/2 + L + R + 2 \cdot R/(r-1) - S)$
- $V = \pi \cdot D^2 \cdot (L + R + 2 \cdot R/(r-1) - S)/4$
- $dV = \pi \cdot D^2 \cdot f \cdot (R \cdot \sin(f \cdot t) + 0.5 \cdot R^2 \cdot \sin(2 \cdot f \cdot t)/(S - R \cdot \cos(f \cdot t)))/4$
- $Sa = R \cdot \cos(f \cdot taf) + (L^2 - R^2 \cdot (\sin(f \cdot taf))^2)^{0.5}$
- $Va = \pi \cdot D^2 \cdot (L + R + 2 \cdot R/(r-1) - Sa)/4$
- $Pot = \mathbf{potencia}(N)$
- $Ce = \mathbf{consumo}(N)$
- $Vc = Pot \cdot Ce$
- $Mc = ((100 - fcomb)/100) \cdot 0.001 \cdot Vc/(30 \cdot Nc \cdot N)$
- $Ra = 1000 \cdot Rg/(0.21 \cdot mO + 0.76 \cdot mN)$
- $Ma = ((100 - far)/100) \cdot c \cdot Pa \cdot Va/(Ra \cdot Ta)$
- $rAC = Ma/Mc$
- $Q = Mc \cdot nc \cdot PCI$
- $xF = \mathbf{fracoes}(nT, rAC, mT, mO, mN)$
- $l = xF(1, 11)$
- $Pr = ((100 - DPr)/100) \cdot (0.7 \cdot N - 350)$
- $dt = 4.66 - 4 \cdot 10^{-2} \cdot (tj) + 1.16 \cdot 10^{-4} \cdot (tj)^2 - 1.12 \cdot 10^{-6} \cdot Pr - 8.95 \cdot 10^{-5} \cdot Pa \cdot 10^{-5} - 9.25 \cdot 10^{-6} \cdot Ta$
- $ti = 360 - (tj) + dt$
- $Dtp = -13.2 - 1.64 \cdot 10^{-2} \cdot Pr + 3.82 \cdot Pa \cdot 10^{-5} + 7.3 \cdot 10^{-2} \cdot Ta + 0.22/l + 3 \cdot 10^{-3} \cdot N$
- $Dt = 30 + 50/(rAC \cdot 0.0691 \cdot 0.7) - 31$
- $Dtd = Dt - Dtp$
- **se** $\underline{t < ti}$ **então**
  - $xTr = xF_{1,1}$
  - $xOr = xF_{1,2}$
  - $xNr = xF_{1,3}$
  - $Mm = Ma$
  - $x = 0$
  - $dx = 0$
- **senão**
  - $xTr = xF_{1,4}$
  - $xOr = xF_{1,5}$
  - $xNr = xF_{1,6}$
  - $Mm = Ma + Mc$
  - $x = 1 - (xpexp(-a((t-ti)/Dtp)^{(mp+1)}) + xdexp(-a((t-ti)/Dtd)^{(md+1)}))$
  - $dx = (xpa(mp+1)/Dtp)((t-ti)/Dtp)^{mp} exp(-a((t-ti)/Dtp)^{(mp+1)}) + (xda * (md+1)/Dtd)((t-ti)/Dtd)^{md} exp(-a((t-ti)/Dtd)^{(md+1)})$
- **fim**

40
41 $xDp = xF(1,7)$
42 $xHp = xF(1,8)$
43 $xOp = xF(1,9)$
44 $xNp = xF(1,10)$
45 $mMo = xTr \times mT + xOr \times mO + xNr \times mN$
46 $a0r = xTr \times a0T + xOr \times a0O + xNr \times a0N$
47 $a1r = xTr \times a1T + xOr \times a1O + xNr \times a1N$
48 $a2r = xTr \times a2T + xOr \times a2O + xNr \times a2N$
49 $a3r = xTr \times a3T + xOr \times a3O + xNr \times a3N$
50 $a4r = xTr \times a4T + xOr \times a4O + xNr \times a4N$
51 $a5r = xTr \times a5T + xOr \times a5O + xNr \times a5N$
52 $a0p = xDp \times a0D + xHp \times a0H + xOp \times a0O + xNp \times a0N$
53 $a1p = xDp \times a1D + xHp \times a1H + xOp \times a1O + xNp \times a1N$
54 $a2p = xDp \times a2D + xHp \times a2H + xOp \times a2O + xNp \times a2N$
55 $a3p = xDp \times a3D + xHp \times a3H + xOp \times a3O + xNp \times a3N$
56 $a4p = xDpa4D + xHpa4H + xOpa4O + xNpa4N$
57 $a5p = xDpa5D + xHpa5H + xOpa5O + xNpa5N$
58 $Cvr = 1000 * (a0r + a1r * \log(y(2)) + a2r * (\log(y(2)))^2 + a3r * (\log(y(2)))^3 + a4r * (\log(y(2)))^4 + a5r * (\log(y(2)))^5 - Rg)/mMo$
59 $dCvr = (1000/(y(2)mMo))(a1r + 2a2r\log(y(2)) + 3a3r(\log(y(2)))^2 + 4a4r(\log(y(2)))^3 + 5a5r(\log(y(2)))^4)$
60 $Cvp = 1000 * (a0p + a1p * \log(y(2)) + a2p * (\log(y(2)))^2 + a3p * (\log(y(2)))^3 + a4p * (\log(y(2)))^4 + a5p * (\log(y(2)))^5 - Rg)/mMo$
61 $dCvp = (1000/(y(2)mMo))(a1p + 2a2p\log(y(2)) + 3a3p(\log(y(2)))^2 + 4a4p(\log(y(2)))^3 + 5a5p(\log(y(2)))^4)$
62 $Rm = 1000Rg/mMo$
63 $Cvm = (1-x)Cvr + xCvp$
64 $Tp = ((100 + DTp)/100)((N + 5750)/75 + 293)$
65 $Vp = (\pi/30)4RN$
66 $h = 7.810^{-3}(Vp)^{1/3}(y(1))^{0.5}(y(2))^{0.5}$
67 $dQt = Qdx$
68 $dQp = hA(y(2) - Tp)/(6N) + bdA((y(2))^4 - (Tp)^4)/(6N)$
69 $dW = y(1)dV$
70 $DY = [0\ 0]$
71 $DY(2) = (dQt - dQp - dW - Mmy(2)(Cvp - Cvr)dx)/(MmCvm + Mmy(2)((1-x)dCvr + xdCvp))$
72 **se** $\underline{t < ta}$ **então**
73 $DY(1) = 0$
74 **se** $\underline{ta \leq t < ti}$ **então**
75 $DY(1) = -\frac{n}{V}dW$
76 **se** $\underline{ti \leq t \leq te}$ **então**
77 $DY(1) = \frac{Mm \times Rm \times DY(2) - dW}{V}$
78 **se** $\underline{te < t < tf}$ **então**
79 $DY(1) = \frac{y(1) - Pa}{t - tf}$
80 **senão**
81 $DY(1) = 0$
82 **fim**
83 **fim**
84 **retorna** DY

o valor de v ao vetor sp.

Ao final do $loop$, o vetor sp contém os valores de volume calculados para cada posição do pistão. Atribui-se o vetor sp à variável V. Calcula-se Vd como o produto de π, D^2, e $R/2$, representando a derivada do volume.

A função retorna os vetores V e Vd.

Portanto, a função $\mathbf{dif.volume}$ recebe um vetor contendo a posição do pistão em graus e retorna o volume do cilindro e a derivada do volume em relação ao tempo. Essa função utiliza cálculos trigonométricos e fórmulas geométricas para determinar esses valores.

Algoritmo 4 – Função $\mathbf{dif.volume}$

Entrada: Um vetor T de valores em graus representando a posição do pistão
Saída: $[V, Vd] = \mathbf{dif.volume}(T)$

início
  $D = 0.105$; // Diâmetro do Cilindro (m)
  $R = 0.0685$; // Raio de Manivela (m)
  $L = 0.207$; // Comprimento da Biela (m)
  $r = 16.8$; // Razão de Compressão
  $n =$ comprimento de T ; // Número de elementos em T
  $sw = [\,]$; // Inicializa o vetor sw como vazio
  $sp = [\,]$; // Inicializa o vetor sp como vazio
  para $i = 1$ **até** n **faça**
    $f = \pi/180$
    $t = T(i)$
    $S = R \cdot \cos(f \cdot t) + \sqrt{L^2 - R^2 \cdot (\sin(f \cdot t))^2}$
    $A = \pi \cdot D \cdot (D/2 + L + R + 2 \cdot R/(r-1) - S)$
    $v = \pi \cdot D^2 \cdot (L + R + 2 \cdot R/(r-1) - S)/4$
    $sp = [sp; v]$
  fim
  $V = sp$
  $Vd = \pi \cdot D^2 \cdot R/2$
fim
retorna V, Vd

A.4- Função $\mathbf{parametros.modais}$

O algoritmo recebe como entrada os valores de N, $ftr1$, $ftr2$, $ftr3$, $ftr4$, e $ftr5$, que representam o número de amostras e os fatores relacionados com a rigidez.

Ele retorna os parâmetros modais P, a matriz de inércia J, e a matriz de estado A.

Para iniciar, o algoritmo define as curvas de variação da rigidez através dos vetores Porcentagem e Rigidez, com valores pré-definidos.

Em seguida, gera a matriz de inércia do sistema J, inicializando um vetor Jt com valores de inércia e construindo a matriz diagonal J a partir dele.

A matriz de rigidez do sistema K é gerada, definindo valores constantes $K1$, $K2$, $K3$, $K4$, interpolando spline para calcular $K5$, $K6$, $K7$, $K8$, e $K9$, e definindo um valor constante $K10$.

A matriz de amortecimento C é construída, primeiro inicializando uma matriz diagonal Ca com valores de amortecimento constantes, e então calculando os valores de amortecimento $Cr1$ a $Cr10$ a partir dos valores de rigidez e das constantes $D1$ e $D2$, e finalmente somando Ca e Cr.

Por fim, a matriz de estado A é gerada, preenchendo a matriz a com as matrizes JK, JC, e I, e os autovalores de A são calculados. As frequências naturais e os fatores de amortecimento são extraídos desses autovalores e armazenados em P.

A.5- Função $\mathbf{delay}$

O pseudocódigo apresentado descreve uma função chamada $\mathbf{delay}$ que realiza uma operação de atraso em um sinal de pressão P com base em um valor de tempo n. A função recebe como entrada uma matriz PT contendo colunas de tempo T e pressão P, e o valor de tempo n. O resultado da função é uma matriz TR contendo o tempo T original e o sinal de pressão atrasado Pn.

As variáveis T e P são inicializadas com as colunas correspondentes da matriz PT, representando o tempo e a pressão, respectivamente.

Algoritmo 5 – Função $\mathbf{parametros.modais}$

Entrada: N, $ftr1$, $ftr2$, $ftr3$, $ftr4$, $ftr5$
Saída: P, J, A

início
  Definir curvas de variação da rigidez;
  $Porcentagem = \{0, 1.9652, 3.3543, 4.0658, 5.2517, 6.0987, 6.5053, 7.4540, 8.7415, 11.0794, 14.4337, 18.3640, 20.6002, 23.8190, 29.5112, 35.0678, 36.6942, 42.3186\}$;
  $Rigidez = \{1254000, 1253569.92, 1252494.723, 1251634.571, 1250989.446, 1250129.294, 1249269.133, 1247978.9, 1245398.427, 1241097.64, 1233571.264, 1224754.654, 1218518.51, 1209056.776, 1189058.12, 1167339.143, 1162608.284, 1138953.958\}$;
  Gerar a matriz de inércia do sistema;
  $Jt = \{0.123, 0.0273, 0.035, 0.014, 0.0468, 0.0328, 0.0468, 0.0468, 0.0328, 0.0488, 1.699\}$;
  $J = \{\mathsf{diag}(Jt)\}$;
  Gerar a matriz de rigidez do sistema;
  $K1 = 105000$, $K2 = 150000$, $K3 = 1064000$, $K4 = 1510000$;
  $K5 = \mathsf{spline}(Porcentagem, Rigidez, ftr1)$;
  $K6 = \mathsf{spline}(Porcentagem, Rigidez, ftr2)$;
  $K7 = \mathsf{spline}(Porcentagem, Rigidez, ftr3)$;
  $K8 = \mathsf{spline}(Porcentagem, Rigidez, ftr4)$;
  $K9 = \mathsf{spline}(Porcentagem, Rigidez, ftr5)$;
  $K10 = 2089000$;
  $K = \mathsf{matriz1}(K1, K2, K3, K4, K5, K6, K7, K8, K9, K10)$;
  Gerar a matriz de amortecimento do sistema;
  $Ca = \mathsf{diag}(0, 0, 0, 0, 6, 6, 6, 6, 6, 6, 0)$;
  $D1 = 0.035$;
  $D2 = 0.25$;
  $Cr1 = D2 \times K1/N$, $Cr2 = D2 \times K2/N$;
  $Cr3 = D1 \times K3/N$, $Cr4 = D1 \times K4/N$, $Cr5 = D1 \times K5/N$, $Cr6 = D1 \times K6/N$, $Cr7 = D1 \times K7/N$, $Cr8 = D1 \times K8/N$, $Cr9 = D1 \times K9/N$, $Cr10 = D1 \times K10/N$;
  $Cr = \mathsf{matriz1}(Cr1, Cr2, Cr3, Cr4, Cr5, Cr6, Cr7, Cr8, Cr9, Cr10)$;
  $C = Ca + Cr$;
  Gerar a matriz de estado;
  $a = \mathsf{zeros}(22, 22)$;
  $JK = -\mathsf{inv}(J) \times K$;
  $JC = -\mathsf{inv}(J) \times C$;
  $I = \mathsf{eye}(11)$;
  $a(12:22, 1:11) = JK$;
  $a(12:22, 12:22) = JC$;
  $a(1:11, 12:22) = I$;
  $A = a$;
  $Q = \mathsf{eig}(A)$;
  $sw = []$;
  para $i = 1$ **passo** 2 **até** 21 **faça**
    $p = Q(i)$ $re = \mathsf{real}(p)$ $im = \mathsf{imag}(p)$ $mod = \sqrt{re^2 + im^2}$ $w = \frac{mod}{2\pi}$ $z = \frac{re}{mod} \times 100$ $sw = \mathsf{append}(sw, [w, z])$
  $P = sw$;

A variável m é atribuída com a posição em que o valor de tempo n é encontrado no vetor T.

A variável j é atribuída com o comprimento do vetor P, representando o número de elementos.

Os valores de pressão anteriores ao valor de tempo n são armazenados no vetor $P1$, através da seleção dos elementos de P do índice 1 até m.

Os valores de pressão a partir do valor de tempo n são armazenados no vetor $P2$, através da seleção dos elementos de P do índice $m+1$ até j.

O vetor Pn é formado pela concatenação dos vetores $P2$ e $P1$, invertendo a ordem dos elementos para criar o sinal de pressão atrasado.

A matriz TR é formada pela combinação das colunas de tempo T e pressão atrasada Pn.

A função retorna a matriz TR contendo o tempo original e o sinal de pressão atrasado.

Dessa forma, a função $\mathbf{delay}$ realiza um atraso no sinal de pressão P, deslocando os valores anteriores ao tempo n para a parte final do vetor, criando um sinal atrasado Pn. Isso é útil para análise de sinais e processamento de dados temporais.

Algoritmo 6 – Função $\mathbf{delay}$

Entrada: PT, n
Saída: $TR = \mathbf{delay}(PT, n)$

início
  $T = PT(:, 1)$;
  $P = PT(:, 2)$;
  $m \rightarrow$ buscar $T == n$;
  $j = length(P)$;
  $P1 = P(1 : m)$;
  $P2 = P(m + 1 : j)$;
  $Pn = [P2\ P1]^T$;
  $TR = [T\ Pn]$;
fim
retorna TR

A.6- Função $\mathbf{torque}$

O pseudocódigo apresenta uma função chamada $\mathbf{torque}$ que calcula o torque em função do tempo para um determinado conjunto de parâmetros. A função recebe como entrada o número de rotações N, um vetor de pressões P e a ordem de atraso $ordem$. O resultado da função é um vetor Tq contendo os valores de torque em função do tempo.

As variáveis D, R, L, r, e Ma são inicializadas com valores constantes correspondentes a diâmetro do cilindro, raio de manivela, comprimento da biela, razão de compressão e um valor de massa Ma.

A variável sw é inicializada como um vetor vazio, que será usado para armazenar os valores de tempo e torque.

Inicia-se um $loop$ de t de 0 a 720, representando o tempo em graus.

A variável f é calculada como $\pi/180$, convertendo o ângulo de graus para radianos.

As variáveis $cos(b)$ e $sin(b)$ são calculadas para determinar os valores de $\cos(b)$ e $\sin(b)$, onde b é o ângulo de manivela.

O valor do torque q é calculado utilizando as variáveis $cos(b)$ e $sin(b)$.

A variável a é calculada como a aceleração angular, que depende do ângulo de manivela b.

O valor do torque Fa é calculado multiplicando Ma, um fator de aceleração angular e um fator de raio.

O torque resultante Ta é calculado multiplicando Fa por q.

O valor da pressão Pc é obtido a partir do vetor P na posição $t+1$, que representa o próximo valor de pressão.

A área A é calculada como a área da seção transversal do cilindro.

A força Fc é calculada multiplicando a pressão Pc pela área A.

O torque Tc é calculado multiplicando a força Fc por q.

O torque total Tt é calculado somando os torques Ta e Tc.

O par de valores de tempo e torque é adicionado ao vetor sw.

Após o término do $loop$, o vetor Tj recebe os valores de tempo e torque calculados.

A função $\mathbf{delay}$ é chamada para atrasar os valores de tempo e torque conforme a ordem especificada.

O vetor T é atribuído com o resultado do atraso.

A função retorna o vetor T contendo os valores de torque em função do tempo, considerando o atraso aplicado.

Dessa forma, a função $\mathbf{torque}$ realiza cálculos iterativos para determinar o torque em função do tempo com base nos parâmetros fornecidos. O resultado é um vetor de torque que pode ser usado para as análises e simulações relacionadas ao torque de um motor Diesel.

Algoritmo 7 – Torque

Entrada: $N, P, ordem$
Saída: $Tq = \mathbf{torque}(N, P, ordem)$
início
  $D = 0.105$; // Diâmetro do Cilindro (m)
  $R = 0.0685$; // Raio de Manivela (m)
  $L = 0.207$; // Comprimento da Biela (m)
  $r = 16.8$; // Razão de Compressão
  $Ma = 2.378$;
  $sw = []$;
  para $t = 0$ **até** 720 **faça**
    $f = \pi/180$; // Conversão de graus para radianos
    $\cos(b) = \sqrt{L^2 - R^2 \cdot (\sin(f \cdot t))^2}/L$; // Cálculo de cos(b)
    $\sin(b) = R \cdot \sin(f \cdot t)/L$; // Cálculo de sin(b)
    $q = R \cdot (\sin(f \cdot t) \cdot \cos(b) + \cos(f \cdot t) \cdot \sin(b)/\cos(b)$; // Cálculo de q
    $a = R \cdot f^2 \cdot \cos(f \cdot t) + (f \cdot R^2)/2 \cdot ((f \cdot R^2)/2 \cdot (L^2 - R^2 \cdot \sin(f \cdot t)^2)^{-1.5} \cdot \sin(2 \cdot f \cdot t)^2 + (L^2 - R^2 \cdot \sin(f \cdot t)^2)^{-0.5} \cdot 2 \cdot f \cdot \cos(2 \cdot f \cdot t))$; // Cálculo de a
    $Fa = Ma \cdot R \cdot (2 \cdot \pi \cdot N/60)^2 \cdot a$; // Cálculo de Fa
    $Ta = Fa \cdot q$; // Cálculo de Ta
    $Pc = P(t + 1)$; // Obtenção do valor de Pc
    $A = \pi \cdot D^2/4$; // Cálculo de A
    $Fc = Pc \cdot A$; // Cálculo de Fc
    $Tc = Fc \cdot q$; // Cálculo de Tc
    $Tt = Ta + Tc$; // Cálculo de Tt
    $sw = [sw; t, Tt]$; // Armazenamento de t e Tt
  fim
  $Tj = sw$; // Atribuição de sw a Tj
  $Tq = \mathbf{delay}(Tj, ordem)$; // Aplicação do atraso
fim

A.7- Função $\mathbf{replica.voltas}$

O pseudocódigo apresenta a função $\mathbf{replica.voltas}$ que replica voltas em uma matriz de torques. Essa função recebe um número inteiro N e uma matriz de torques como entrada, e retorna uma matriz Y que contém a replicação das voltas.

A função começa calculando o valor de z através da divisão de N por 60, e arredonda esse valor para o número inteiro mais próximo para obter x. Dependendo do valor de x, atribui-se x ou $x + 1$ para n.

Em seguida, são extraídos os tempos e os torques da matriz de entrada. Inicialmente, a matriz sw é preenchida com os torques originais. Em seguida, é realizado um loop que replica as voltas adicionando as linhas dos torques nas posições adequadas na matriz sw.

Após a replicação das voltas, é calculado o tamanho da matriz Tj, e os valores dos tempos são gerados a partir do primeiro valor e de uma diferença calculada com base no segundo valor dos tempos originais. Esses valores de tempo são armazenados em uma matriz tj.

Por fim, a matriz final Y é criada combinando as colunas dos tempos replicados e dos torques replicados.

Em resumo, a função $\mathbf{replica.voltas}$ replica voltas na matriz de torques, permitindo que os torques sejam mapeados para cada instante de tempo correspondente. Isso é feito considerando o número inteiro fornecido como parâmetro e replicando os valores de torques conforme o número de voltas desejado. O resultado é uma matriz que contém os tempos replicados seguidos pelos torques replicados.

replica.voltas

O algoritmo $\mathbf{parametros.modais}$ calcula os parâmetros modais com base nas entradas: elementos da matriz de inércia $\mathbf{J}$; elementos da matriz de rigidez torcional $\mathbf{K}$ e elementos da matriz de amortecimento torcional $\mathbf{C}$. A função recebe um número inteiro N e uma matriz $\mathbf{ftr}$ como entrada e retorna uma matriz P como saída.

Inicialmente, são definidas algumas variáveis, como $\mathbf{Porcentagem}$, $\mathbf{Rigidez}$, Jt, J, $\mathbf{K14}$, $\mathbf{K29}$, $K10$, $\mathbf{Ka}$, $\mathbf{K}$, Cat, Ca, $D1$, $D2$, $\mathbf{Da}$, $\mathbf{D}$, $\mathbf{Cra}$, Cr, $\mathbf{C}$, $\mathbf{a}$, $\mathbf{JK}$, $\mathbf{JC}$, $\mathbf{I}$, A, Q e sw.

Em seguida, são definidos os valores das variáveis $\mathbf{Porcentagem}$, $\mathbf{Rigidez}$ e Jt.

Algoritmo 8 – $\mathbf{replica.voltas}$

Entrada: N: um número inteiro, $\mathbf{torques}$: uma matriz de torques
Saída: Y: uma matriz resultante
Variáveis: z, x, n, t, T, sw, p, W, Tj, m, Z, $dtheta$, dt, sz, tj

1 $z = \frac{N}{60}$;
2 $x = \mathbf{round}(z)$;
3 **se** $\underline{\mathbf{mod}(x, 2) = 0}$ **então**
4 $\quad n = x$;
5 **senão**
6 $\quad n = x + 1$;
7 **fim**
8 $t = \mathbf{torques}(:, \mathbf{1})$;
9 $T = \mathbf{torques}(:, \mathbf{2} : \mathbf{7})$;
10 $sw = [T]$;
11 **para** $\underline{i = 1}$ **até** $\underline{n/2 - 1}$ **faça**
12 $\quad p = \mathbf{length}(t)$;
13 $\quad W = T(2 : p, :)$;
14 $\quad sw = [sw; W]$;
15 **fim**
16 $Tj = sw$;
17 $m = \mathbf{length}(Tj)$;
18 $Z = t(1)$;
19 $dtheta = t(2) - t(1)$;
20 $dt = \frac{dtheta}{6 \cdot N}$;
21 $sz = [Z]$;
22 **para** $\underline{i = 1}$ **até** $\underline{m - 1}$ **faça**
23 $\quad Z = Z + dt$;
24 $\quad sz = [sz; Z]$;
25 **fim**
26 $tj = sz$;
27 $Y = [tj \quad Tj]$;

A matriz diagonal J é criada a partir dos valores de Jt.

Posteriormente, são definidas as variáveis $\mathbf{K14}$, $\mathbf{K29}$ e $K10$, e a matriz $\mathbf{Ka}$ é formada concatenando esses valores. A matriz diagonal $\mathbf{K}$ é criada a partir dos valores de $\mathbf{Ka}$.

As variáveis Cat, Ca, $D1$, $D2$, $\mathbf{Da}$, $\mathbf{D}$ e $\mathbf{Cra}$ são definidas e utilizadas para calcular a matriz $\mathbf{Cr}$. A matriz $\mathbf{C}$ é formada somando as matrizes $\mathbf{Ca}$ e $\mathbf{Cr}$.

A matriz $\mathbf{a}$ é inicializada com zeros. Em seguida, as matrizes $\mathbf{JK}$ e $\mathbf{JC}$ são calculadas usando a inversa de J multiplicada por K e C, respectivamente. A matriz identidade $\mathbf{I}$ é criada, e as matrizes $\mathbf{JK}$, $\mathbf{JC}$ e $\mathbf{I}$ são atribuídas às posições adequadas na matriz $\mathbf{a}$.

A matriz A recebe o valor de $\mathbf{a}$. Em seguida, é calculado o vetor Q contendo os autovalores de A.

A matriz sw é inicializada como uma matriz vazia. Em um loop, são iterados os valores de i de 1 a 21, incrementando a cada 2. Dentro do $loop$, cada autovalor é processado para obter sua parte real e imaginária, e então é calculado o módulo desse autovalor. O valor da frequência é obtido dividindo o módulo pelo valor 2π, e o valor da porcentagem é calculado dividindo a parte real pelo módulo e multiplicando por 100. Finalmente, os valores de frequência e porcentagem são concatenados na matriz sw.

No final, a matriz P recebe o valor de sw sendo retornada como resultado.

Em resumo, o algoritmo $\mathbf{parametros.modais}$ realiza uma série de cálculos para obter os parâmetros modais com base nas entradas fornecidas. A função cria matrizes, calcula autovalores e processa informações para finalmente retornar a matriz P contendo os valores de frequência e porcentagem.

A.8- Função $\mathbf{runge.kutta}$

O algoritmo apresentado é a implementação da técnica do método Runge-Kutta para resolver equações diferenciais. Ele recebe três parâmetros como entrada: o número de iterações N, a função F e a condição inicial x_0. O algoritmo retorna o vetor de solução x obtido pelo método Runge-Kutta da função F. A subfunção $\mathbf{vibracao}(t, x)$ é utilizada para calcular as soluções do sinal da função F, para cada uma das quatro ordens do

Algoritmo 9 – parametros.modais

Entrada: N, **ftr**
Saída: P, J, A
Variáveis: **Porcentagem**, **Rigidez**, $J1$, $J2$, $J3$, $J4$, $J5$, $J6$, $J7$, $J8$, $J9$, $J10$, $J11$, Jt, J, $K1$, $K2$, $K3$, $K4$, $K5$, $K6$, $K7$, $K8$, $K9$, $K10$, K, $Ca1$, $Ca2$, $Ca3$, $Ca4$, $Ca5$, $Ca6$, Cat, Ca, $D1$, $D2$, $Cr1$, $Cr2$, $Cr3$, $Cr4$, $Cr5$, $Cr6$, $Cr7$, $Cr8$, $Cr9$, $Cr10$, Cr, C, a, JK, JC, I, A, Q, sw

1 **início**
2 $\quad$ **Porcentagem** $= [0 \ 1.9652 \ 3.3543 \ 4.0658 \ 5.2517 \ 6.0987 \ 6.5053 \ \ldots]$;
3 $\quad$ **Rigidez** $= [1254000 \quad 1253569.92 \quad 1252494.723 \quad 1251634.571 \quad \ldots]$;
4 $\quad$ $Jt = [0.123 \ 0.0273 \ 0.035 \ 0.014 \ 0.0468 \ 0.0328 \ 0.0468 \ 0.0468 \ 0.0328 \ \ldots]$;
5 $\quad$ $J = \mathbf{diag}(Jt)$;
6 $\quad$ **K14** $= [105000 \quad 150000 \quad 1064000 \quad 1510000]$;
7 $\quad$ **K29** $= \mathbf{spline}(\mathbf{Porcentagem}, \mathbf{Rigidez}, \mathbf{ftr})$;
8 $\quad$ $K10 = 2089000$;
9 $\quad$ **Ka** $= [K14 \quad K29 \quad K10]$;
10 $\quad$ **K** $= \mathbf{matriz1}(\mathbf{Ka})$;
11 $\quad$ $Cat = [0 \quad 0 \quad 0 \quad 0 \quad 6 \quad 6 \quad 6 \quad 6 \quad 6 \quad 6 \quad 0]$;
12 $\quad$ $Ca = \mathbf{diag}(Cat)$;
13 $\quad$ $D1 = 0.035$;
14 $\quad$ $D2 = 0.25$;
15 $\quad$ **Da** $= [D2 \quad D2 \quad D1 \quad D1 \quad D1 \quad D1 \quad D1 \quad D1 \quad D1 \quad D1]$;
16 $\quad$ **D** $= \mathbf{diag}(\mathbf{Da})$;
17 $\quad$ **Cra** $= \frac{1}{N} \cdot \mathbf{D} \cdot \mathbf{Ka}$;
18 $\quad$ $Cr = \mathbf{matriz1}(\mathbf{Cra})$;
19 $\quad$ **C** $= \mathbf{Ca} + \mathbf{Cr}$;
20 $\quad$ **a** $= \mathbf{zeros}(22, 22)$;
21 $\quad$ **JK** $= -\mathbf{inv}(J) \cdot K$;
22 $\quad$ **JC** $= -\mathbf{inv}(J) \cdot C$;
23 $\quad$ **I** $= \mathbf{eye}(11)$;
24 $\quad$ $a(12:22, 1:11) = JK$;
25 $\quad$ $a(12:22, 12:22) = JC$;
26 $\quad$ $a(1:11, 12:22) = I$;
27 $\quad$ $A = a$;
28 $\quad$ $Q = \mathbf{eig}(A)$;
29 $\quad$ $sw = []$;
30 $\quad$ **para** $i = 1:2:21$ **faça**
31 $\quad\quad$ $p = Q(i)$;
32 $\quad\quad$ $re = \mathbf{real}(p)$;
33 $\quad\quad$ $im = \mathbf{imag}(p)$;
34 $\quad\quad$ $mod = \sqrt{re^2 + im^2}$;
35 $\quad\quad$ $w = \frac{mod}{2\pi}$;
36 $\quad\quad$ $z = \frac{re}{mod} \times 100$;
37 $\quad\quad$ $sw = \mathbf{concatenar}(sw, \{w, z\})$;
38 $\quad$ **fim**
39 **fim**
40 **retorna** $P = sw$

método de Runge-Kutta.

Para um sistema com N graus de liberdade, dado um vetor de forças externas F e condições iniciais x_0, podemos resolver a equação de movimento utilizando o método de Runge-Kutta de ordem 4.

O algoritmo começa definindo o passo de tempo dt e o tempo final t_f. Em seguida, inicializa os vetores de tempo t e solução x, com t_0 e x_0, respectivamente. Calcula o número de iterações n com base no tempo final e no passo de tempo.

Em um $loop$, de $i = 1$ até N, o algoritmo realiza os seguintes passos: calcula a primeira derivada k_1 usando a função $\mathbf{vibracao}(t, x)$; calcula a segunda derivada k_2 usando $\mathbf{vibracao}(t + \frac{dt}{2}, x + dt\ \frac{k_1}{2})$; calcula a terceira derivada k_3 usando $\mathbf{vibracao}(t + \frac{dt}{2}, x + dt\ \frac{k_2}{2})$; calcula a quarta derivada k_4 usando $\mathbf{vibracao}(t + dt, x + dt\ k_3)$; atualiza o vetor de tempo t para $t_i = t_{(i-1)} + dt$ e Atualiza o vetor de solução x para $x_i = x_{i-1} + \frac{dt}{6(k_1+2k_2+2k_3+k_4)}$.

Ao final do $loop$, o algoritmo retorna o vetor de solução x.

Em resumo, o algoritmo implementa o método Runge-Kutta para obter a solução numérica de uma equação diferencial. Ele avança no tempo, calculando as derivadas em diferentes pontos e usando uma média ponderada dessas derivadas para atualizar o valor da solução. O processo é repetido até atingir o número desejado de iterações.

Algoritmo 10 – Técnica do algoritmo Runge-Kutta.

Entrada: N, F, x_0
Saída: $k = \mathbf{Runge.Kutta}(N, F, x_0)$
início
  Definir o passo de tempo dt e o tempo final t_f;
  Inicializar o vetor de tempo t com $t_0 = 0$;
  Inicializar o vetor de solução x com x_0;
  Calcular o número de iterações $n = \frac{t_f}{dt}$;
  para $i = 1$ **até** N **faça**
    Calcular a primeira derivada: $k_1 = \mathbf{vibracao}(t, x)$;
    Calcular a segunda derivada: $k_2 = \mathbf{vibracao}(t + \frac{dt}{2}, x + dt\ \frac{k_1}{2})$;
    Calcular a terceira derivada: $k_3 = \mathbf{vibracao}(t + \frac{dt}{2}, x + dt\ \frac{k_2}{2})$;
    Calcular a quarta derivada: $k_4 = \mathbf{vibracao}(t + dt, x + dt\ k_3)$;
    Atualizar o vetor de tempo t com $t_i = t_{(i-1)} + dt$;
    Atualizar o vetor de solução x com $x_i = x_{i-1} + \frac{dt}{6(k_1+2k_2+2k_3+k_4)}$;
  fim
  retorna o vetor de solução x;
fim

A.8.1 Subfunção $\mathbf{vibracao}$ de $\mathbf{runge.kutta}$

O algoritmo apresentado na função $\mathbf{vibracao}$ recebe como entrada os parâmetros t, r, T, F, A e M, e retorna o valor de Fxt como saída.

O algoritmo calcula $fi2$ usando a função $\mathbf{interp1q}$, que realiza uma interpolação linear entre os vetores T e F, para obter o valor correspondente a t. Em seguida, remove as colunas adicionais de $fi2$ além da primeira coluna.

Na linha 3, calcula f multiplicando a matriz inversa de M por $fi2$, seguido pela obtenção do número de linhas de M e armazenamento em m.

Inicializa o vetor z com zeros de tamanho m, cria o vetor Fi concatenando z com f e calcula Fxt multiplicando A por r e somando com Fi.

O resultado é o valor de Fxt.

A função $\mathbf{interp1q}$ desempenhou um papel crucial neste trabalho, proporcionando uma significativa melhoria no tempo de processamento dos algoritmos utilizados para resolver o modelo diesel. Tal melhoria possibilitou a geração eficiente de diversos sinais necessários para a construção da base de dados, auxiliando a solucionar as equações em um prazo adequado para a criação da base de dados.

Em resumo, a função $\mathbf{vibracao}$ calcula o vetor Fxt que representa uma resposta de vibração. Ela interpola os valores de F para o tempo t, realiza operações matriciais envolvendo M e A, e retorna o resultado.

Algoritmo 11 – vibração

Entrada: t, r, T, F, A, M
Saída: Fxt
Variáveis: $fi2$, f, m, z, Fi, Fxt

1 $fi2 = (\mathbf{interp1q}(T', F', t \times \mathbf{ones}(\mathbf{size}(F, 1), 1)))'$;
2 $fi2(:, 2 : \mathbf{end}) = []$;
3 $f = \mathbf{inv}(M) \times fi2$;
4 $[m, \sim] = \mathbf{size}(M)$;
5 $z = \mathbf{zeros}(m, 1)$;
6 $Fi = [z; f]$;
7 $Fxt = A \times r + Fi$;

Resultado: Fxt

A.9- Função $\mathbf{vibracao.torcional}$

O algoritmo $\mathbf{vibracao.torcional}$ é uma função que recebe um vetor chamado $resposta$ como entrada e retorna os valores de $torq$ e $sertemp_vib$ como saída.

Esta função calcula o valor de $torq$ arredondando a média absoluta da diferença entre as colunas 12 e 11 do vetor $resposta$, multiplicadas por um valor fixo. Em seguida, cria uma matriz chamada $sertemp_vib$ que contém as colunas $tempo$ e $torcional$ extraídas do vetor $resposta$.

O algoritmo retorna os valores de $torq$ e $sertemp_vib$ como resultado.

Algoritmo 12 – vibracao.torcional

Entrada: $resposta$
Saída: $torq$, $sertemp_vib$
Variáveis: K_6_vol, des_vol, des_cil6, $torq$, $tempo$, $torcional$, $sertemp_vib$

1 $K_6_vol = 2089000$;
2 $des_vol = resposta(:, 12)$;
3 $des_cil6 = resposta(:, 11)$;
4 $torq = \mathbf{round}(\mathbf{abs}(\mathbf{mean}((des_vol - des_cil6) \times K_6_vol)))$;
5 $tempo = resposta(:, 1)$;
6 $torcional = (des_vol - des_cil6) \times K_6_vol$;
7 $sertemp_vib = [\mathbf{tempo} \quad \mathbf{torcional}]$;
8 **retorna** $torq$, $sertemp_vib$;

A.10- Função $\mathbf{ordem.espectro}$

O algoritmo $orden.espectro$ é uma função que recebe dois parâmetros como entrada: um valor inteiro N e um vetor chamado $resposta$. Ele retorna o valor de ord como saída e também modifica a variável sw.

Primeiro, o algoritmo extrai as colunas de tempo e vibração torcional do vetor $resposta$ e as armazena nas variáveis $tempo$ e vib_tor, respectivamente. Em seguida, chama a função $espectro_exemplo1$ para calcular os valores de f, $P3$ e $P6$ com base nos valores de $tempo$ e vib_tor.

Após isso, o algoritmo realiza um conjunto de iterações em um $loop$ for que varia

de 1 a 24, onde realizadas as seguintes operações: calcula-se o valor de fi multiplicando j por $frot$, onde $frot$ é calculado como a divisão de N por 60. Em seguida, é inicializada a variável i com o valor 1.

Em um $loop\ while$, o algoritmo procura o índice i para o qual a diferença absoluta entre $f(i)$ e fi é menor que um terço do valor de df, que representa a diferença entre os valores de frequência adjacentes em f. Ao encontrar o índice i, o algoritmo armazena o valor de n como o índice onde f é igual a $f(i)$, e também armazena os valores de $f(i)$, $P3(i)$ e $P6(i)$ nas variáveis f_i, a_i e b_i, respectivamente.

A cada iteração, o algoritmo concatena os valores de f_i, a_i e b_i ao vetor sw. Por fim, ele atribui os valores de sw à variável ord e retorna os valores de ord e sw como resultado.

Algoritmo 13 – ordem.espectro

Entrada: N, $resposta$
Saída: ord
Variáveis: $tempo$, vib_tor, f, $P3$, $P6$, $frot$, x, y, sw, df, j, fi, i, n, f_i, a_i, b_i, ord

1 $tempo = resposta(:, 1)$;
2 $vib_tor = resposta(:, 2)$;
3 $[f, P3, P6] = \mathbf{espectro_exemplo1}(tempo, vib_tor)$;
4 $frot = \frac{N}{60}$;
5 $x = 0$;
6 $y = 0$;
7 $sw = []$;
8 $df = f(2) - f(1)$;
9 **para** $j = 1$ **até** $j \leq 24$ **faça**
10 $\quad fi = \frac{j}{2} \times frot$;
11 $\quad i = 1$;
12 $\quad$ **enquanto** $|f(i) - fi| \geq df/3$ **faça**
13 $\quad\quad i = i + 1$;
14 $\quad$ **fim**
15 $\quad n = \mathbf{find}(f == f(i))$;
16 $\quad f_i = f(i)$;
17 $\quad a_i = P3(i)$;
18 $\quad b_i = P6(i)$;
19 $\quad sw = \mathbf{concatenar}(sw, \{f_i, a_i, b_i\})$;
20 $\quad ord = sw$
21 **fim**
22 **retorna** ord;

A.11- Função $\mathbf{tensoes.global}$

O algoritmo $tensoes_global$ é uma função que recebe dois parâmetros como entrada: um valor inteiro N e um vetor chamado $resposta$. Ele retorna os valores de f, $P3$ e $Tensao$ como saída.

Primeiro, o algoritmo atribui o valor 85.9 à variável D_vol. Em seguida, ele extrai as colunas de tempo e vibração torcional do vetor $resposta$ e as armazena nas variáveis $tempo$ e vib_tor, respectivamente. Em seguida, a função $espectro_exemplo$ calcula os valores de f, $P3$ e $P6$ com base nos valores de $tempo$ e vib_tor.

Na linha 5, valor de $frot$ dividindo N por 60 é calculado, seguido pela inicialização das variáveis x, y e sw com zero. Calcula a diferença entre os valores de frequência adjacentes em f e atribui à variável df, bem como inicializa a variável ft como um vetor vazio.

Em um $loop\ for$ que itera de 1 a 24, o algoritmo realiza as seguintes operações: calcula o valor de fi multiplicando j por $frot$ e inicializa a variável i com 1. Internamente em $loop\ while$, o algoritmo encontra o índice i na qual a diferença absoluta entre $f(i)$ e fi é menor que um terço do valor de df.

Em seguida, encontra o índice n no qual o valor de f é igual a $f(i)$, e armazena os valores de $P3(n)$, $P6(n)$ (convertido para radianos) nas variáveis ai e bi, respectivamente. Calcula xi e yi com base em ai e bi, e atualiza os valores de x e y somando xi e yi, respectivamente. Concatena o valor de ai ao vetor sw.

Após o $loop$, o algoritmo atribui os valores de x e y às variáveis X e Y. Calcula o valor da amplitude Amp com base nos valores de X e Y. Calcula os valores de Im e Zm usando a fórmula fornecida.

Finalmente, calcula o valor de $Tensao$ dividindo Amp por Zm e multiplicando por 10^{-6}, retornando os valores de f, $P3$ e $Tensao$ como resultado.

Algoritmo 14 – tensoes_global

Entrada: N, $resposta$
Saída: f, $P3$, $Tensao$
Variáveis: D_vol, $tempo$, vib_tor, f, $P3$, $P6$, $frot$, x, y, sw, df, ft, j, fi, i, n, ai, bi, xi, yi, X, Y, Amp, Im, Zm, $Tensao$

1 $D_vol = 85.9$;
2 $tempo = resposta(:, 1)$;
3 $vib_tor = resposta(:, 2)$;
4 $[f, P3, P6] =$ **espectro_exemplo**$(tempo, vib_tor)$;
5 $frot = \frac{N}{60}$;
6 $x = 0$;
7 $y = 0$;
8 $sw = []$;
9 $df = f(2) - f(1)$;
10 $ft = []$;
11 **para** $j = 1$ **até** $j \leq 24$ **faça**
12 $fi = \frac{j}{2} \times frot$;
13 $i = 1$;
14 **enquanto** $|f(i) - fi| \geq df/3$ **faça**
15 $i = i + 1$;
16 **fim**
17 $n =$ **find**$(f == f(i))$;
18 $ai = P3(n)$;
19 $bi = P6(n) \times \frac{\pi}{180}$;
20 $xi = ai \times \cos(bi)$;
21 $yi = ai \times \sin(bi)$;
22 $x = x + xi$;
23 $y = y + yi$;
24 $sw =$ **concatenar**$(sw, \{ai\})$;
25 **fim**
26 $X = x$;
27 $Y = y$;
28 $Amp = \sqrt{X^2 + Y^2}$;
29 $Im = \pi \times \left(\frac{D_vol \times 10^{-3}}{2}\right)^4 / 32$;
30 $Zm = \frac{Im}{D_vol \times 10^{-3}/2}$;
31 $Tensao = \frac{Amp}{Zm} \times 10^{-6}$;
32 **retorna** f, $P3$, $Tensao$

A.12- Função $\mathbf{pressao.media1}$

O algoritmo $pressao_media_1$ é uma função que recebe cinco parâmetros como entrada: os vetores T, P, V, Vd e taf, retornando o valor de pmi como saída.

Neste algoritmo, são realizadas as seguintes operações: inicializa-se a variável i com o tamanho do vetor T e divide-se o vetor T em duas partes, $T1$ e $T2$, com base no índice central $(i+1)/2$.

Em seguida, são encontrados os índices n e m no qual os valores de T são iguais a $taf(1)$ e $taf(2)$, respectivamente. Os vetores P e V são divididos em partes de acordo com esses índices, resultando nos vetores Pa, $P1$, $P2$, Va e $V2$. O vetor $P1$ é reorganizado em ordem reversa.

Posteriormente, são definidas as funções $f1$ e $f2$ como interpolações *spline* dos dados de $V1$ e $P1$, e $V2$ e $P2$, respectivamente. São calculados os comprimentos dos vetores $V1$ e $V2$ e armazenados nas variáveis p e q, respectivamente. Além disso, são determinados os valores iniciais e finais dos vetores $V1$ e $V2$.

São calculadas as integrais das funções $f1$ e $f2$ com limites definidos pelos vetores $Vn1$, $Vn2$, $Vm1$ e $Vm2$. A diferença entre as integrais é armazenada na variável Wi. Em seguida, é calculado o valor de pmi dividindo-se Wi por Vd e multiplicando por 10^{-5}.

O valor de pmi é retornado como resultado da função.

A.13- Função $\mathbf{espectro.exemplo}$

O algoritmo $espectro_exemplo$ é uma função que recebe dois parâmetros como entrada: os vetores T e R. Ele retorna os vetores f, $P3$ e $P6$ como saída.

Neste algoritmo, são realizadas as seguintes operações: calcula-se o comprimento do vetor T e armazena o último elemento na variável Tf. Em seguida, calcula-se a frequência de amostragem Fs dividindo o comprimento pelo valor de Tf, e o período de amostragem Ts como o inverso de Fs.

É aplicada a transformada rápida de Fourier (FFT) no vetor R utilizando a função

Algoritmo 15 – pressao.media1

Entrada: T, P, V, Vd, taf
Saída: pmi
Variáveis: i, $T1$, $T2$, n, m, Pa, $P1$, $P2$, Va, $V1$, $V2$, $f1$, $f2$, p, q, $Vn1$, $Vn2$, $Vm1$, $Vm2$, $Wi1$, $Wi2$, Wi, pmi

1 $i = \mathbf{length}(T)$;
2 $T1 = T(1 : (i + 1)/2)$;
3 $T2 = T((i + 1)/2 : i)$;
4 $n = \mathbf{find}(T == taf(1))$;
5 $m = \mathbf{find}(T == taf(2))$;
6 $Pa = P(n : (i + 1)/2)$;
7 $P1 = \mathbf{reorder_array}(Pa)$;
8 $P2 = P((i + 1)/2 : m)$;
9 $Va = V(n : (i + 1)/2)$;
10 $V1 = \mathbf{reorder_array}(Va)$;
11 $V2 = V((i + 1)/2 : m)$;
12 $f1 = @(w)\mathbf{spline}(V1, P1, w)$;
13 $f2 = @(z)\mathbf{spline}(V2, P2, z)$;
14 $p = \mathbf{length}(V1)$;
15 $q = \mathbf{length}(V2)$;
16 $Vn1 = V1(1)$;
17 $Vn2 = V1(p)$;
18 $Vm1 = V2(1)$;
19 $Vm2 = V2(q)$;
20 $Wi1 = \mathbf{quad}(f1, Vn1, Vn2)$;
21 $Wi2 = \mathbf{quad}(f2, Vm1, Vm2)$;
22 $Wi = Wi2 - Wi1$;
23 $pmi = Wi/Vd/10^5$;

Resultado: pmi

fft, e o resultado é armazenado no vetor Y. O vetor $P2$ é obtido calculando o valor absoluto de Y dividido pelo comprimento L.

Os vetores $P1$, $P3$ e $P6$ são derivados do vetor $P2$, sendo o vetor $P1$ obtido através da seleção de elementos do início até a metade do vetor $P2$, incluindo o primeiro e o último elementos. Os elementos de índice 2 até o penúltimo são multiplicados por 2.

O vetor $P3$ é igual a $P1$. O vetor $P4$ é obtido calculando a fase complexa de Y e desembrulhando-a usando a função $unwrap$. O vetor $P5$ é obtido selecionando os elementos de $P4$ da mesma forma que o vetor $P1$, e os elementos de índice 2 até o penúltimo são multiplicados por 2.

Finalmente, o vetor $P6$ é obtido multiplicando os elementos de $P5$ por $180/\pi$. O vetor f é calculado como a frequência de amostragem Fs multiplicada pelo vetor de índices de 0 até a metade do comprimento L dividido por L.

Os vetores f, $P3$ e $P6$ são retornados como resultado da função.

Algoritmo 16 – espectro.exemplo

Entrada: T, R
Saída: f, $P3$, $P6$
Variáveis: L, Tf, Fs, Ts, Y, $P2$, $P1$, $P3$, $P4$, $P5$, $P6$, f

1 $L = \mathbf{length}(T)$;
2 $Tf = T(L)$;
3 $Fs = L/Tf$;
4 $Ts = 1/Fs$;
5 $Y = \mathbf{fft}(R)$;
6 $P2 = \mathbf{abs}(Y/L)$;
7 $P1 = P2(1 : \mathbf{round}(L/2) + 1)$;
8 $P1(2 : \mathbf{end} - 1) = 2 \times P1(2 : \mathbf{end} - 1)$;
9 $P3 = P1$;
10 $P4 = \mathbf{unwrap}(\mathbf{angle}(Y))$;
11 $P5 = P4(1 : \mathbf{round}(L/2) + 1)$;
12 $P5(2 : \mathbf{end} - 1) = 2 \times P5(2 : \mathbf{end} - 1)$;
13 $P6 = P5 \times 180/\pi$;
14 $f = Fs \times (0 : \mathbf{round}(L/2))/L$;

Resultado: f, $P3$, $P6$

A.14- Função $\mathbf{barra.progresso}$

Essa função auxiliar de visualização recebe dois parâmetros: '$total$', que representa o valor total do progresso do processamento, e '$atual$', que representa seu valor atual.

A variável $porcentagem$ é calculada como a divisão entre '$atual$' e $total$, multiplicada por 100, para obter a percentual atual da barra de progresso.

A variável $barra$ é inicializada como uma $string$ contendo o caractere '—'. Em seguida, é utilizado o comando $\mathbf{repmat}$ para replicar o caractere '—' um número de vezes igual à porcentagem completada, arredondada para o inteiro mais próximo.

Em um $loop$, a barra de progresso é preenchida com o caractere '-' até atingir o comprimento total de 100 caracteres.

Por fim, a barra de progresso é formatada com colchetes '[' e ']' para melhor visualização, sendo retornada como saída do algoritmo.

Algoritmo 17 – barra.progresso

Entrada: $total$, $atual$
Saída: $barra$
Variáveis: $porcentagem$, $barra$, i
início
  $porcentagem = \frac{atual}{total} \times 100$;
  $barra = \mathbf{repmat}('|', 1, \mathbf{round}(porcentagem))$;
  para $\underline{i = \mathbf{length}(barra) + 1}$ **até** $i \leq 100$ **faça**
    $barra = [barra, '-']$;
  fim
  $barra = ['[', barra, ']']$;
fim