



Universidade Federal
do Rio de Janeiro

Escola Politécnica

UM SISTEMA DE POSICIONAMENTO DE VEÍCULOS POR SATÉLITE
TOLERANTE A AUSÊNCIAS DE COBERTURA

Lucas de Carvalho Gomes

Projeto de Graduação apresentado ao Curso de Computação e Informação da Escola Politécnica da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Computação e Informação.

Orientadores: Luís Henrique Maciel Kosmalski
Costa
João Batista Pinto Neto

Rio de Janeiro
Setembro de 2018

UM SISTEMA DE POSICIONAMENTO DE VEÍCULOS POR SATÉLITE
TOLERANTE A AUSÊNCIAS DE COBERTURA

Lucas de Carvalho Gomes

PROJETO SUBMETIDO AO CORPO DOCENTE DO CURSO DE COMPUTAÇÃO E INFORMAÇÃO DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO DE COMPUTAÇÃO E INFORMAÇÃO.

Examinadores:

Prof. Luís Henrique Maciel Kosmalski Costa, Dr.

Prof. João Batista Pinto Neto, M.Sc.

Prof. Pedro Braconnot Velloso, Dr.

Prof. Rodrigo de Souza Couto, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2018

Gomes, Lucas de Carvalho

Um Sistema de Posicionamento de Veículos por Satélite Tolerante a Ausências de Cobertura/Lucas de Carvalho Gomes. – Rio de Janeiro: UFRJ/POLI – COPPE, 2018.

XIII, 101 p.: il.; 29,7cm.

Orientadores: Luís Henrique Maciel Kosmalski Costa

João Batista Pinto Neto

Projeto (graduação) – UFRJ/ Escola Politécnica/ Curso de Computação e Informação, 2018.

Referências Bibliográficas: p. 99 – 101.

1. Navegação Inercial. 2. GNSS. 3. Sensores Inerciais. 4. Aprendizado de Máquina. 5. Análise de Séries Temporais. I. Costa, Luís Henrique Maciel Kosmalski *et al.* II. Universidade Federal do Rio de Janeiro, Escola Politécnica/ Curso de Computação e Informação. III. Título.

Agradecimentos

Em primeiro lugar, agradeço à minha família, que sempre me deu suporte e incentivo para que eu batalhasse pelos meus objetivos.

Em especial, agradeço à minha mãe por sempre acreditar em mim, por acompanhar sempre tão de perto toda a minha caminhada até aqui, por ser um exemplo de força e determinação e por todo o carinho, todo o suporte e todos os ensinamentos. Muito do que sou hoje eu devo especialmente a você.

Também agradeço especialmente ao meu pai, que, para mim, é um exemplo de perseverança e dedicação. Obrigado pelo apoio e pelo carinho.

Um muito obrigado ao meu padrasto, que acabou por me inserir no mundo da informática. Muitas das minhas primeiras experiências com computadores foram por conta dele, e isto despertou minha curiosidade, tanto que escolhi seguir a computação como carreira.

Agradeço muito ao professor Luís Henrique Costa pela paciência, pelo suporte que sempre me foi dado como meu orientador nos meus quase 5 anos no Grupo de Teleinformática e Automação (GTA) e por me orientar neste trabalho. Também lhe agradeço por ter permitido utilizar o seu carro para que fossem feitas as coletas dos dados usadas aqui!

Agradeço muito também ao João Batista Pinto Neto, doutorando, professor, co-orientador e grande amigo, com quem trabalhei junto por bastante tempo no GTA. Sua atitude sempre positiva e sua persistência em resolver desafios, por mais complexos que pareçam, são uma fonte de inspiração.

Agradeço aos professores Pedro Braconnot Velloso e Rodrigo de Souza Couto por aceitarem participar da banca de avaliação deste projeto de fim de curso.

Por fim, agradeço à UFRJ como um todo pelas oportunidades, pelo aprendizado e por ser um ambiente onde me desenvolvi muito, não só profissionalmente, mas também pessoalmente.

Resumo do Projeto de Graduação apresentado à Escola Politécnica da UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Computação e Informação.

UM SISTEMA DE POSICIONAMENTO DE VEÍCULOS POR SATÉLITE TOLERANTE A AUSÊNCIAS DE COBERTURA

Lucas de Carvalho Gomes

Setembro/2018

Orientadores: Luís Henrique Maciel Kosmowski Costa
João Batista Pinto Neto

Curso: Engenharia de Computação e Informação

Sistemas de posicionamento por satélite são utilizados em diversos contextos para a localização de pessoas e objetos. Estes sistemas funcionam através da propagação de ondas eletromagnéticas. Além dos fatores ambientais que afetam esta propagação e, conseqüentemente, a acurácia da localização, uma desvantagem de tais sistemas é o fato de que a perda da cobertura dos satélites torna a informação da posição indisponível. Isto ocorre em alguns ambientes cobertos, como cânions geográficos e urbanos, túneis e florestas. Com base neste contexto, este trabalho propõe um sistema experimental de posicionamento que calcula estimativas da posição quando não são recebidas posições através dos satélites. Informações anteriores obtidas através de um sistema de posicionamento e leituras extraídas de sensores foram aplicadas em algoritmos de Aprendizado de Máquina com técnicas de pré-processamento, e a melhor combinação de técnicas e parâmetros foi selecionada, a partir da qual foi projetado um sistema experimental.

Palavras-Chave: Navegação Inercial, GNSS, Sensores Inerciais, Aprendizado de Máquina, Análise de Séries Temporais.

Abstract of the Undergraduate Project presented to Poli/UFRJ as a partial fulfillment of the requirements for the degree of Computer and Information Engineer.

A RESISTANT TO LACKS OF COVERAGE SATELLITE VEHICLE POSITIONING SYSTEM

Lucas de Carvalho Gomes

September/2018

Advisors: Luís Henrique Maciel Kosmalski Costa
João Batista Pinto Neto

Course: Computer and Information Engineering

Satellite positioning systems are used in various contexts for obtaining the locations of people and objects. These systems operate through the propagation of electromagnetic waves. Aside from the environmental factors that affect this propagation and, by consequence, affect the location accuracy, one disadvantage of such systems is the fact that the loss of satellite coverage causes an unavailability of the location. That occurs in ambients such as urban and geographic canyons, tunnels and forests. Considering this context, this work proposes an experimental positioning system that calculates position estimates when no position is obtained through the satellites. Past information obtained from a satellite positioning system and readings extracted from sensores were applied in Machine Learning algorithms with pre-processing techniques, and the best combination was selected. From this combination, an experimental system was designed.

Keywords: Dead Reckoning, GNSS, Inertial Sensors, Machine Learning, Time Series Analysis.

Sumário

Lista de Figuras	x
Lista de Tabelas	xiii
1 Introdução	1
1.1 Trabalhos Relacionados	6
1.2 Organização do Trabalho	7
2 Modelos Estáticos e Dinâmicos	8
2.1 Modelos Estáticos	8
2.2 Modelos Dinâmicos	8
2.2.1 Definição de uma Série Temporal	9
2.2.2 Índice de Correlação de Pearson	9
2.2.3 Autocorrelação	10
2.2.4 Estacionariedade Estrita	11
2.2.5 Estacionariedade Fraca	12
2.2.6 Autocorrelação Parcial	12
2.2.7 Correlação Cruzada	14
2.2.8 Ruído Branco Gaussiano	14
2.2.9 Modelo de Média Móvel (MA)	15
2.2.10 Modelo Não-linear de Média Móvel (NMA)	15
2.2.11 Modelo Autorregressivo (AR)	16
2.2.12 Modelo Não-linear Autorregressivo (NAR)	16
2.2.13 Modelo ARMA	16
2.2.14 Modelo NARMA	17
2.2.15 Modelos com Entradas Exógenas	17
2.2.16 Regressão Dinâmica	18
2.2.17 Modelos Integrados	19
2.2.18 Modelos com Erros Dinâmicos	19
2.2.19 Classificação dos Modelos de Séries Temporais	20
2.3 Seleção de Modelos	21

2.3.1	Tratamento de Tendências	21
2.3.2	Análise da Autocorrelação	23
2.3.3	Análise da Autocorrelação Parcial	24
2.3.4	Comportamentos Simulados dos Modelos Teóricos Não- Exógenos	26
2.3.5	Análise da Correlação Cruzada	31
3	Análise dos Dados Coletados	32
3.1	Equipamentos Utilizados	32
3.1.1	Formato das Informações Disponíveis	34
3.2	Coleta dos Dados	35
3.3	Análise do Conjunto de Dados	36
3.3.1	Posição Geográfica	36
3.3.2	Sensores	43
3.3.3	Análise da Correlação Cruzada	47
4	Técnicas de Aprendizado de Máquina	52
4.1	Algoritmos de Regressão	52
4.1.1	Regressão Linear	52
4.1.2	Rede Neural Artificial	54
4.2	Técnicas de Pré-Processamento	63
4.2.1	Normalização	64
4.2.2	Particionamento dos Dados por Análise de Agrupamentos	64
4.3	Divisões dos Dados para Validação dos Modelos	66
4.3.1	Validação Simples	67
4.3.2	Validação Cruzada	67
5	Resultados	69
6	Conclusão e Trabalhos Futuros	83
A	Conceitos Básicos de Estatística e Modelos Probabilísticos	85
A.1	Probabilidade	85
A.2	Espaço Amostral	85
A.3	Variável Aleatória Discreta	85
A.3.1	Densidade de Probabilidade	86
A.3.2	Distribuição de Probabilidade	87
A.4	Variável Aleatória Contínua	87
A.4.1	Distribuição de Probabilidade	87
A.4.2	Densidade de Probabilidade	87
A.5	Distribuição Normal ou Gaussiana	88

A.6	Valor Esperado	88
A.6.1	Propriedades	88
A.7	Momentos	89
A.7.1	Média	89
A.7.2	Variância e Desvio Padrão	89
A.8	Independência Estatística	89
A.9	Intervalo de Confiança	90
B	Provas das Propriedades dos Modelos MA e AR	91
B.1	Autocorrelação para modelos MA	91
B.2	Autocorrelação para modelos AR	94
B.3	Valores dos Coeficientes de um Modelo AR	97
	Referências Bibliográficas	99

Lista de Figuras

1.1	Divisão dos Algoritmos de Aprendizado de Máquina.	3
2.1	Autocorrelação do modelo MA(2) simulado.	27
2.2	Autocorrelação do modelo AR(2) simulado.	27
2.3	Autocorrelação do modelo ARMA(2,2) simulado.	28
2.4	Autocorrelação parcial do modelo MA(2) simulado.	29
2.5	Autocorrelação parcial do modelo AR(2) simulado.	29
2.6	Autocorrelação parcial do modelo ARMA(2, 2) simulado.	30
3.1	Exemplo de diagrama de conexões em um barramento CAN.	33
3.2	Trajeto realizado na coleta de dados pela Ilha do Fundão.	37
3.3	Comportamento da latitude em relação ao tempo.	37
3.4	Comportamento da longitude em relação ao tempo.	38
3.5	Comportamento da primeira diferença da latitude em relação ao tempo.	38
3.6	Comportamento da primeira diferença da longitude em relação ao tempo.	39
3.7	Comportamento da segunda diferença da latitude em relação ao tempo.	40
3.8	Comportamento da segunda diferença da longitude em relação ao tempo.	40
3.9	Autocorrelações das primeiras diferenças das coordenadas.	41
3.10	Autocorrelações parciais das primeiras diferenças das coordenadas.	41
3.11	Autocorrelações das segundas diferenças das coordenadas.	42
3.12	Autocorrelações parciais das segundas diferenças das coordenadas.	42
3.13	Comportamentos das frequências de rotação das rodas com o tempo.	44
3.14	Comportamento da velocidade de translação do carro com o tempo.	44
3.15	Diferenças entre frequências de rotação contra velocidades.	45
3.16	Autocorrelações das frequências de rotação das rodas.	46
3.17	Autocorrelação da velocidade de translação do veículo.	46
3.18	Autocorrelações parciais das frequências de rotação das rodas.	46
3.19	Autocorrelação parcial da velocidade de translação do veículo.	47

3.20	Correlação Cruzada entre a primeira diferença da latitude e a frequência de rotação da roda esquerda.	48
3.21	Correlação Cruzada entre a primeira diferença da latitude e a frequência de rotação da roda direita.	48
3.22	Correlação Cruzada entre a primeira diferença da latitude e a frequência de rotação da roda esquerda.	49
3.23	Correlação Cruzada entre a primeira diferença da longitude e a frequência de rotação da roda esquerda.	49
3.24	Correlação Cruzada entre a primeira diferença da longitude e a frequência de rotação da roda direita.	50
3.25	Correlação Cruzada entre a primeira diferença da longitude e a frequência de rotação da roda esquerda.	50
4.1	Processo realizado por uma Regressão Linear unidimensional.	53
4.2	Diagrama da arquitetura de um Perceptron de Múltiplas Camadas.	55
4.3	Diagrama de um neurônio, unidade básica de processamento de uma RNA.	56
5.1	Exibição dos trajetos de treino e teste para o primeiro arranjo da validação cruzada no teste de acurácia.	72
5.2	Exibição dos trajetos de treino e teste para o segundo arranjo da validação cruzada no teste de acurácia.	72
5.3	Exibição dos trajetos de treino e teste para o terceiro arranjo da validação cruzada no teste de acurácia.	73
5.4	Exibição dos trajetos de treino e teste para o primeiro arranjo da validação cruzada no teste de acurácia.	73
5.5	Comportamento temporal do erro para o primeiro arranjo da validação cruzada no teste de acurácia.	74
5.6	Comportamento temporal do erro para o segundo arranjo da validação cruzada no teste de acurácia.	74
5.7	Comportamento temporal do erro para o terceiro arranjo da validação cruzada no teste de acurácia.	75
5.8	Comportamento temporal do erro para o quarto arranjo da validação cruzada no teste de acurácia.	75
5.9	Exibição dos trajetos de treino e teste para o primeiro arranjo da validação cruzada na previsão de um trecho.	76
5.10	Exibição dos trajetos de treino e teste para o segundo arranjo da validação cruzada na previsão de um trecho.	76
5.11	Exibição dos trajetos de treino e teste para o terceiro arranjo da validação cruzada na previsão de um trecho.	77

5.12	Exibição dos trajetos de treino e teste para o quarto arranjo da validação cruzada na previsão de um trecho.	77
5.13	Comportamento temporal do erro para o primeiro arranjo da validação cruzada na previsão de um trecho.	77
5.14	Comportamento temporal do erro para o segundo arranjo da validação cruzada na previsão de um trecho.	78
5.15	Comportamento temporal do erro para o terceiro arranjo da validação cruzada na previsão de um trecho.	78
5.16	Comportamento temporal do erro para o quarto arranjo da validação cruzada na previsão de um trecho.	78
5.17	Comportamento temporal do erro da latitude para o primeiro arranjo da validação cruzada na previsão de um trecho.	79
5.18	Comportamento temporal do erro da latitude para o segundo arranjo da validação cruzada na previsão de um trecho.	79
5.19	Comportamento temporal do erro da latitude para o terceiro arranjo da validação cruzada na previsão de um trecho.	79
5.20	Comportamento temporal do erro da latitude para o quarto arranjo da validação cruzada na previsão de um trecho.	80
5.21	Comportamento temporal do erro da longitude para o primeiro arranjo da validação cruzada na previsão de um trecho.	80
5.22	Comportamento temporal do erro da longitude para o segundo arranjo da validação cruzada na previsão de um trecho.	80
5.23	Comportamento temporal do erro da longitude para o terceiro arranjo da validação cruzada na previsão de um trecho.	81
5.24	Comportamento temporal do erro da longitude para o quarto arranjo da validação cruzada na previsão de um trecho.	81

Lista de Tabelas

2.1	Classificação dos modelos lineares de Séries Temporais.	20
2.2	Classificação dos modelos não-lineares de Séries Temporais.	20
3.1	Especificações dos equipamentos utilizados.	34
3.2	Especificações gerais do conjunto de dados coletado.	36

Capítulo 1

Introdução

Atualmente, sistemas de posicionamento por satélite são utilizados em diversos contextos. Por exemplo, são usados para localizar objetos, pessoas ou veículos no mundo, para encontrar objetos perdidos, para aplicações de navegação e aplicações que visam aumentar a segurança no trânsito, tais como sistemas de prevenção de colisões [1].

Tais sistemas calculam a posição com base no tempo de propagação de ondas eletromagnéticas [2], emitidas a partir de um ou mais satélites a um receptor. As coordenadas que indicam a posição do receptor na Terra são a latitude (distância em graus do receptor ao Equador, que varia de 0 a 90° para norte e para sul) e a longitude (distância em graus do receptor ao Meridiano de Greenwich, que varia de 0 a 180° para leste e para oeste).

Atualmente, existem quatro sistemas de posicionamento por satélite disponíveis no mundo: o americano GPS (*Global Positioning System*), o russo GLONASS (sigla para *Globalnaya Navigatsionnaya Sputnikovaya Sistema*, russo para Sistema de Navegação Global por Satélite), o europeu GALILEO e o chinês BeiDou. A diferença principal entre estes sistemas são as constelações de satélites às quais cada um destes possui acesso.

Existem vários fatores ambientais e relativos aos equipamentos que afetam o desempenho destes sistemas [2]. Os fatores abaixo afetam a precisão e a acurácia da posição obtida:

- Camadas da atmosfera terrestre: a troposfera e a ionosfera causam desvios no sinal propagado;
- Condições do tempo atmosférico: nuvens podem causar desvios adicionais nos sinais eletromagnéticos, prejudicando a transmissão dos sinais;
- Efeito de múltiplos caminhos (ou multicaminho): reflexões de ondas eletromagnéticas por objetos, prédios, árvores e os desvios causados pela atmosfera

fazem com que o sinal emitido pelo satélite possa chegar à antena do receptor por mais de um caminho, causando a recepção de dois ou mais pulsos do mesmo sinal; esses pulsos podem ter diferenças de fase entre si, causando interferências que geram ruídos e contribuem para a atenuação do sinal;

- Atrasos nos relógios dos satélites: os relógios dos satélites são usados para medir o tempo de propagação, e a acurácia dos mesmos afeta a acurácia da posição calculada;
- Erros nas posições dos satélites (ou *ephemeris*): os satélites também possuem sensores que indicam as suas posições, usados para determinar a posição do receptor em relação a eles com base no tempo de propagação dos sinais enviados; a precisão e a acurácia dos mesmos afetam a precisão e a acurácia finais obtidas.

Sistemas baseados em satélites também são sujeitos à perda de cobertura em alguns ambientes, como florestas muito densas, túneis e cânions (geográficos e urbanos), o que impede a comunicação entre satélite e receptor. Este é um problema especialmente grave, porque prejudica ou impede o funcionamento de aplicações que dependem dos dados de posição, dependendo da duração da indisponibilidade, e porque, no caso de aplicações que envolvam segurança no trânsito ou condução autônoma de veículos, aumenta o risco de acidentes.

A possibilidade da perda de cobertura é uma limitação de qualquer aplicação que utilize tais sistemas como o único método de posicionamento e, por isso, boa parte das aplicações que demandam um alto grau de disponibilidade (principalmente voltadas à segurança no trânsito) empregam outros sensores para obter a localização relativa, tais como sensores de proximidade com ultrassom ou lasers, LiDARs (*Light Detection And Ranging*) e câmeras. Por outro lado, reduzir a quantidade de sensores usados em uma aplicação reduz os custos, a complexidade da instalação dos equipamentos e diminui o consumo de energia.

Com base nesse contexto, este trabalho visa projetar um sistema que forneça a localização em tempo real, seja tolerante às possíveis perdas de sinal e mantenha a precisão em um nível aceitável para aplicações dependentes da posição geográfica. Além disto, um aspecto importante deste sistema é utilizar sensores já disponíveis, de modo a evitar custos adicionais. O funcionamento desejado é que o sistema consiga construir uma trajetória aproximada para o objeto de interesse durante um período de ausência de cobertura. Ou seja: enquanto há posições disponíveis através de um sistema de posicionamento por satélite, o sistema deve “aprender” o comportamento das leituras dos sensores. Quando perde-se a cobertura dos satélites, o sistema deve, a partir apenas das leituras atuais dos sensores e do aprendizado anterior, prever a trajetória do veículo sem auxílio do sistema de navegação.

Um dos procedimentos que se pode utilizar para construir um modelo que realize estas predições é através de algoritmos de Aprendizado de Máquina. Aprendizado de Máquina é uma subárea da Inteligência Artificial, cujo objetivo geral é fazer com que um sistema computacional aprenda a realizar uma tarefa, por meio de simulações da tarefa em questão e do emprego de uma métrica de desempenho que possa orientar um algoritmo de aprendizado. Para isto, esta área emprega uma família de algoritmos, que buscam resolver três classes de problemas diferentes:

- Aprendizado Supervisionado: a um computador se apresenta exemplos de entradas e saídas (resultados) desejados, com o objetivo de que ele aprenda uma regra geral que relacione entradas e saídas;
- Aprendizado Não-Supervisionado: nenhum tipo de resultado esperado é fornecido; a tarefa realizada pelo computador é encontrar estruturas e padrões nos dados fornecidos;
- Aprendizado por Reforço: um sistema é posto para interagir com um ambiente dinâmico, no qual ele deve realizar uma tarefa; à medida que ele realiza ações baseadas no seu conhecimento até o momento, o ambiente envia respostas (positivas ou negativas), que causam ajustes nos parâmetros do sistema.

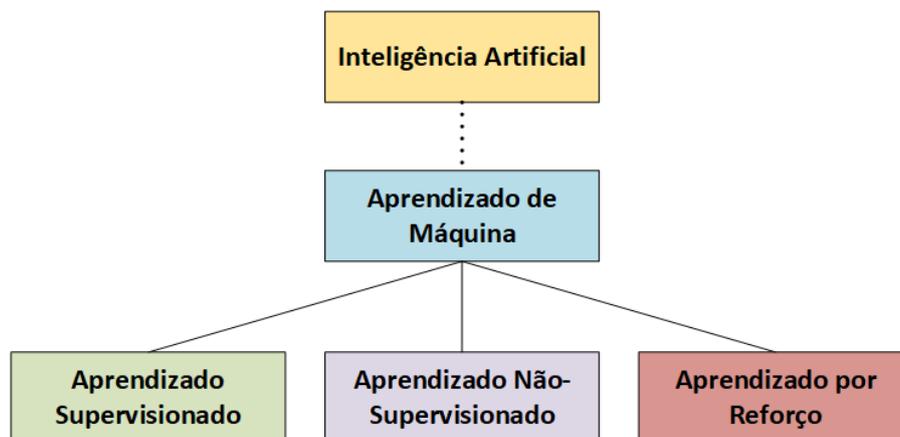


Figura 1.1: Divisão dos Algoritmos de Aprendizado de Máquina.

A Figura 1.1 apresenta a divisão dos tipos de algoritmos de Aprendizado de Máquina. Dado que há disponíveis entradas (sensores) e saídas (posições) desejados, e que o objetivo do sistema é utilizar uma regra que o permita realizar predições das saídas, o sub-ramo da Inteligência Computacional de interesse é o Aprendizado Supervisionado. Para este ramo, diz-se que um algoritmo gera um modelo após ser treinado com os dados fornecidos.

O Aprendizado Supervisionado divide-se ainda em dois sub-ramos:

- Regressão: com um conjunto de entradas, deseja-se estimar o valor numérico das saídas desejadas; as métricas de desempenho são medidas para quantificar o erro de predição;
- Classificação: deseja-se determinar dentro de qual categoria (ou classe) se enquadra um conjunto de valores de entradas - esta é a saída desejada; as métricas de desempenho são funções das taxas de acertos e taxas de erros para cada classe.

Uma vez que o objetivo no cenário em questão (predição da trajetória) é estimar as coordenadas, que possuem valores numéricos, o problema que se deseja resolver é o de Regressão.

Como mencionado nos parágrafos anteriores, as técnicas de Aprendizado de Máquina dependem de dados fornecidos ao sistema computacional. Uma consequência disto é que a qualidade das regras construídas depende diretamente da qualidade e da quantidade dos dados fornecidos. Dados muito *imprecisos* (por exemplo, com muito ruído) podem afetar a eficiência do sistema de aprendizado. Se os tipos de informações fornecidos forem insuficientes para se construir uma regra adequada, o algoritmo também não conseguirá gerar um modelo muito eficiente. Por outro lado, adicionar muitos tipos de informações *não-correlacionadas* com a saída desejada pode interferir negativamente na eficiência. Além disto, a qualidade da regra é diretamente proporcional à quantidade de leituras fornecidas. A *proporção* entre os diferentes tipos de leituras também afeta a precisão do modelo - por exemplo, se o sistema proposto aqui for alimentado com muitas leituras extraídas de veículos em alta velocidade, suas predições para veículos em baixa velocidade ou parados podem ficar prejudicadas. Além disso, alguns modelos, como será visto nos próximos capítulos, são sensíveis à *escala* dos dados (o intervalo entre o maior e o menor valor assumido).

Para suavizar estes efeitos do conjunto de dados no desempenho, existem diversos métodos de pré-processamento. Alguns métodos serão utilizados e serão mencionados mais adiante. Além disto, para selecionar as variáveis de entrada, pode-se utilizar métricas de correlação [3].

Cada tipo de modelo possui uma lei de formação de regras. Por exemplo, alguns algoritmos estimam relações lineares entre entradas e saídas, outros estimam relações polinomiais, e outros modelos calculam aproximações por meio de outros tipos de relações não-lineares. A relação entre os dados depende do problema, portanto, o melhor algoritmo depende do caso. Gerar modelos que não estão de acordo com o comportamento real geram previsões com menos acurácia, gerando dois fenômenos:

- *Underfitting* ou subajuste: ocorre quando o algoritmo gera um modelo menos complexo que a relação real – um exemplo é aproximar uma relação polinomial

(como ax^2) por uma equação linear (como $ax + b$), e se manifesta por meio de um desempenho não-satisfatório;

- *Overfitting* ou sobreajuste: ocorre quando o algoritmo gera um modelo mais complexo que a relação real; este problema se manifesta quando as previsões possuem bons desempenhos em dados conhecidos pelo modelo (os mesmos dados usados para o treino) e desempenhos insatisfatórios em dados desconhecidos (o que indica que o modelo “memorizou” os dados usados no treinamento e apresenta baixa capacidade de generalização).

Para identificar o melhor tipo de modelo, uma das maneiras mais simples é testar diversos candidatos no mesmo conjunto de dados e verificar qual apresenta o melhor desempenho. Dois algoritmos foram escolhidos neste projeto, um que gera regras lineares (Regressão Linear) e um aproximador de funções não-lineares (Rede Neural Artificial). O desempenho deles será comparado neste trabalho.

Um outro efeito interessante a se investigar é a dependência do valor das coordenadas em relação a valores anteriores das mesmas. Dado que é possível, durante o funcionamento do sistema, armazenar valores obtidos quando há cobertura dos satélites e devido à existência de informações auxiliares (os sensores), cujos valores anteriores também podem ser armazenados se necessário, uma possível modelagem interessante para o problema é a modelagem como um sistema dinâmico. Sistemas dinâmicos são modelados como modelos de séries temporais, um ramo da estatística que explora a relação entre o valor atual de uma variável com informações do passado, muito utilizado em sistemas dinâmicos e na econometria. Existem diversas maneiras de modelar sistemas dinâmicos, e estas serão descritas e comparadas nos próximos capítulos. As modelagens serão implementadas pelos algoritmos de regressão mencionados no parágrafo anterior.

O resultado final da análise é o projeto de um sistema de navegação cega que:

- Monitora periodicamente o receptor de um sistema de posicionamento por satélite, aguardando uma posição nova;
- Monitora periodicamente os sensores, armazenando uma quantidade a ser determinada de leituras mais recentes;
- Armazena, caso necessário, uma eventual nova posição recebida do receptor;
- Estima, no caso de não haver uma posição nova, a posição atual do veículo com base nas leituras dos sensores e nas posições anteriores.

1.1 Trabalhos Relacionados

Existem diversos trabalhos com o objetivo de prover a localização geográfica no caso de uma falha de cobertura. O popular aplicativo de navegação *Waze* possui um projeto chamado *Waze Beacons*, que instala microcontroladores (ou *beacons*) de baixo custo da fabricante *Bluvision* com comunicações *Bluetooth* em túneis para prover a localização geográfica dentro dos mesmos e possibilitar aos usuários a publicação de alertas sobre as condições de trânsito no interior dos túneis. O projeto já está implementado em algumas cidades em parceria com seus governos, como Haifa (Israel), Chicago (Estados Unidos) e Rio de Janeiro (Brasil). Por meio das comunicações entre os *beacons* e os *smartphones* dos usuários, o aplicativo estima a posição do usuário. Cada beacon custa US\$ 28,50, e o *website* do *Waze* informa que são necessários aproximadamente 42 *beacons* por milha de um túnel. Os *beacons* possuem uma bateria com duração entre 5 e 6 anos [4][5][6]. A desvantagem deste tipo de abordagem é a necessidade de instalar dispositivos auxiliares, o que implica em custos adicionais. Utilizando os valores fornecidos pelo *Waze*, pode-se concluir que o custo de instalar os dispositivos para cobrir um túnel é aproximadamente 8033 reais por quilômetro.

O trabalho de Mascharka [7] segue uma abordagem parecida com o *Waze Beacons* para localizar pessoas em ambientes fechados: utiliza pontos de acesso *Wi-Fi* localizados no interior de um prédio, que se comunicam com o *smartphone* de um usuário. Por meio destas comunicações, o sistema extrai a potência do sinal recebido por cada ponto de acesso. O sistema fornece estes dados e os valores fornecidos por sensores inerciais do *smartphone* (acelerômetro, magnetômetro, sensores de rotação e sensores de orientação) a um algoritmo de Aprendizado de Máquina para que a posição seja estimada. A abordagem oferece erros de no máximo três metros para a previsão da posição de um usuário parado e de no máximo 10.15 metros na estimativa da localização de um usuário em movimento. A abordagem tem a mesma limitação do *Waze Beacons*: a dependência de uma infraestrutura externa. Além disto, o trabalho não indica qual é o desempenho do sistema no caso de uma falha de cobertura que dure mais que um instante de atualização do sistema de posicionamento por satélite utilizado; cada estimativa foi feita possuindo-se as posições verdadeiras anteriores. Também é importante notar que este trabalho opera em um ambiente diferente do veicular: os pedestres apresentam velocidades muito mais baixas que as dos veículos, e já foi possível se observar um aumento significativo no erro quando as estimativas foram feitas com um pedestre em movimento, o que indica que o ambiente veicular poderia oferecer erros maiores com a mesma abordagem.

O trabalho de Pinto Neto [8] é uma proposta recente de navegação cega. Utiliza um conjunto de modelos treinados pelo algoritmo de regressão linear para estimar as

posições. A partir de sensores de frequências de rotação das rodas de um veículo, é determinado o tipo de trajetória realizado pelo carro, e as estimativas são calculadas pelo modelo que foi treinado com posições e leituras de sensores correspondentes a este tipo de trajetória. Para prever a trajetória de um veículo, é necessário utilizar dados de posições anteriores, o que causa um acúmulo dos erros das consecutivas predições. Para contornar este problema, o trabalho utiliza um fator de correção dinâmico periodicamente ajustado.

Belhajem [9] possui um outro trabalho recente na área. Utiliza um modelo não-linear de predição, o Filtro de Kalman Estendido (ou *Extended Kalman Filter*) para prever a localização a partir de um acelerômetro e um giroscópio. As estimativas a partir deste algoritmo apresentam um erro rapidamente cumulativo. Para compensar o crescimento do erro, o sistema proposto no seu trabalho calcula fatores de correção dinâmicos através de uma Rede Neural Artificial. Os resultados apresentam erros abaixo de 1 metro.

1.2 Organização do Trabalho

O restante do trabalho está organizado da seguinte forma. O Capítulo 2 descreve as modelagens estáticas (sem dependências com o passado) e dinâmicas (modelos de séries temporais), bem como os critérios utilizados para selecionar os melhores candidatos a modelos e os conceitos nos quais eles se baseiam. A partir dos conceitos trabalhados, o Capítulo 3 discorre sobre os dados coletados para a simulação de um protótipo do sistema - o processo de coleta, os equipamentos utilizados e a análise dos mesmos de acordo com os critérios descritos no Capítulo 2 -, além de indicar quais são os possíveis candidatos para modelagens do sistema. O Capítulo 4 descreve as técnicas de Aprendizado de Máquina que serão utilizadas para calcular as estimativas de acordo com as modelagens selecionadas - os algoritmos de regressão e os métodos de pré-processamento. Com as conclusões do Capítulo 3 e os algoritmos descritos no Capítulo 4, o Capítulo 5 descreve as simulações realizadas com os candidatos a modelos para o sistema e discute os resultados obtidos. Por fim, o Capítulo 6 apresenta as conclusões deste projeto de fim de curso e aponta possíveis trabalhos futuros.

Capítulo 2

Modelos Estáticos e Dinâmicos

Este capítulo descreve os candidatos a modelos matemáticos e algoritmos para calcular as estimativas das posições e os conceitos necessários para analisá-los e selecioná-los. Como mencionado na introdução deste trabalho, há dois tipos possíveis de abordagem: modelos estáticos, que não consideram dependências com o passado, e modelos dinâmicos, que consideram informações obtidas no passado. Neste capítulo, descreve-se com mais detalhes os dois tipos de abordagens. Por fim, na última seção deste capítulo são apresentadas as diferentes métricas que podem ser utilizadas para a avaliação dos modelos.

2.1 Modelos Estáticos

Esta classe de modelos é chamada *estática* por não considerar dependências dos valores anteriores da posição ou dos sensores, utilizando apenas os valores mais recentes. Um modelo estático pode ser representado matematicamente como:

$$y(t) = F(x_1(t), x_2(t), \dots, x_n(t), \epsilon(t)), \quad (2.1)$$

onde y é a variável que se deseja estimar, as variáveis x_i são as grandezas a partir das quais se estima o valor de y , $\epsilon(t)$ é um termo aleatório que representa o erro de predição e F é alguma função que indica a relação entre as variáveis. Esta abordagem dá origem a dois tipos gerais de modelos: um modelo estático linear e um modelo estático não-linear. Este tipo de modelo também é conhecido na literatura como *regressão de séries temporais*.

2.2 Modelos Dinâmicos

Modelos dinâmicos expressam o comportamento de uma variável em função do seu valor atual e de informações do passado. Por isto, podem ser representados como

modelos de *séries temporais*, que preveem o valor da variável de interesse em função dos valores de três tipos de grandezas diferentes em uma quantidade arbitrária de instantes:

- Valores anteriores da variável objetivo;
- Termos de ruído branco;
- Entradas exógenas (outras grandezas - elas mesmas séries temporais - que influenciam no comportamento da variável objetivo).

Por isto, são tipos de modelos interessantes para a aplicação em sistemas de comportamento dinâmico como o proposto neste trabalho. Esta seção apresenta, em primeiro lugar, a definição formal de uma série temporal e características de interesse usadas na definição dos modelos e de suas análises. A seguir, descreve-se os diferentes tipos de modelos dinâmicos. Uma resumida descrição dos conceitos de probabilidade e estatística utilizados nesta seção encontra-se no Apêndice A. A discussão feita nesta seção é baseada no exposto por Shumway [10] e Hyndman [11].

2.2.1 Definição de uma Série Temporal

Uma série temporal é uma sequência de valores associada a um processo, representado por uma variável de interesse. O valor desta variável depende do instante (tempo) de execução deste processo. Matematicamente, isto pode ser definido como o conjunto $\mathbf{S} : [x(1), x(2), \dots, x(t)]$, no qual x é a variável de interesse e o seu argumento é o instante no qual um valor para x foi obtido. O comportamento deste processo pode ser descrito em função de diferentes termos do passado, como mencionado anteriormente, e as dependências destes termos pode ser verificada através das métricas de correlação, apresentadas a seguir.

2.2.2 Índice de Correlação de Pearson

O índice de correlação de Pearson é uma métrica que determina a dependência entre duas variáveis x e y . Ele é calculado a partir de um conjunto de observações das duas variáveis obtidas em diversos instantes. Seu valor pode ser calculado como:

$$r(x, y) = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y)}{[\frac{1}{n} \sum_{i=1}^n (x_i - \mu_x)^2]^{1/2} [\frac{1}{n} \sum_{i=1}^n (y_i - \mu_y)^2]^{1/2}}, \quad (2.2)$$

$$r(x, y) = \frac{C_{x,y}}{\sigma_x \sigma_y}, \quad (2.3)$$

onde μ_x e μ_y são as médias de x e y , σ_x e σ_y são os desvios padrão de x e y e $C_{x,y}$ é a covariância entre x e y .

O valor de r varia no intervalo $[-1, +1]$. Valores grandes do módulo de r indicam uma dependência significativa entre as duas variáveis, e valores de r próximos de zero indicam que a dependência não é significativa ou é inexistente. No entanto, necessita-se determinar os limiares a partir dos quais o valor da correlação indica uma dependência significativa.

Para isso, suponha a existência de duas variáveis a e b . Caso elas sejam independentes e randômicas, o valor esperado para a correlação é 0. No entanto, se as duas variáveis forem simuladas e um número finito (N) de pares (x, y) for gerado, pode-se obter valores para a correlação diferentes de 0, dado que a amostra obtida pode não ser representativa o suficiente. Além disso, de acordo com Shumway [10], se este processo for realizado diversas vezes e extrair-se o valor da correlação, será possível concluir que o valor da correlação segue uma distribuição normal, com média nula e desvio padrão de $\sigma_r = 1/\sqrt{N}$.

Com esta informação, gera-se um método para determinar se os valores de correlação são significativos ou não. Um valor considerado significativo está fora do intervalo $[-2/\sqrt{N}, +2/\sqrt{N}]$ (ou mais ou menos dois desvios padrão). Este intervalo é uma aproximação do intervalo de confiança de 95% da correlação. Observe que 95% é um valor usualmente adotado como nível de confiança.

A medida de correlação, para qualquer modelo de Aprendizado de Máquina, é utilizado para dois propósitos, de acordo com Hastie [3]:

- Determinar se há uma dependência relevante entre variáveis de saída e variáveis de entrada; apenas as variáveis de entrada das quais a variável de saída dependa significativamente devem ser utilizadas;
- Determinar se há correlações entre as variáveis de entrada. Valores muito altos de correlação entre duas variáveis de entrada são um indicativo de que uma das duas pode ser eliminada.

No caso de uma série temporal, como se deseja prever o valor atual de uma variável usando os seus valores anteriores, um dos procedimentos usados na análise da série é verificar a correlação do valor atual com seus valores anteriores; esta métrica se chama autocorrelação e é descrita a seguir.

2.2.3 Autocorrelação

A autocorrelação de uma série temporal é a correlação entre o valor atual da variável objetivo ($x(t)$) e o valor da mesma variável atrasado de k unidades de tempo ($x(t - k)$). Se houver N valores disponíveis para uma variável, podemos montar $N - k$ pares de valores do tipo $(x(t), x(t - k))$ - dado que a série foi atrasada de k unidades

-, onde $t = 1, \dots, N - k$, e, com estes pares, pode-se calcular a autocorrelação da seguinte forma, em função do atraso k :

$$r(x, k) = r(x(t), x(t - k)) \quad (2.4)$$

$$r(x, k) = \frac{1}{N - k} \frac{1}{\sigma_x^2} \sum_{t=1}^{N-k} (x(t) - \mu_x)(x(t + k) - \mu_x), \quad (2.5)$$

ou, resumidamente:

$$r(x, k) = \frac{AC_x(k)}{\sigma_x^2}, \quad (2.6)$$

sendo μ_x e σ_x , respectivamente, a média e a variância da variável x , e $AC_x(k)$ a função de autocovariância de x com um atraso k . Um gráfico que exibe valores de autocorrelação por valores de atraso é chamado de *autocorrelograma*.

Os intervalos de significância de um autocorrelograma são calculados da mesma forma que na medida de correlação. Um detalhe que se deve observar é o fato de que, quanto maior o atraso, menor é a quantidade de leituras disponíveis para formar pares $x(t), x(t - k)$ para o cálculo da autocorrelação. Por isso, o tamanho do intervalo de significância é diretamente proporcional ao atraso.

A análise de um autocorrelograma permite verificar a quantidade de instantes futuros para os quais é possível prever valores da variável a partir de uma leitura já obtida. Também indica quais valores atrasados são bons candidatos para se utilizar na previsão de um valor futuro.

2.2.4 Estacionariedade Estrita

Em muitas séries temporais, existe um comportamento regular e inalterado com o tempo. Uma série temporal com esta regularidade é interessante por poder ser analisada e modelada com diversos conceitos que dependem de um comportamento não alterado. Este comportamento regular é formalmente definido como a propriedade de estacionariedade.

Uma série temporal estritamente estacionária, de acordo com [10], é uma sequência de valores de uma variável qualquer x na qual o *comportamento probabilístico* de uma coleção de valores $[x(t_1), x(t_2), \dots, x(t_k)]$ é idêntico ao *comportamento probabilístico* do conjunto deslocado no tempo $[x(t_1 + h), x(t_2 + h), \dots, x(t_k + h)]$. Isto pode ser descrito com distribuições de probabilidade cumulativas multivariadas, da seguinte forma:

$$P(x(t_1) \leq c_1, \dots, x(t_k) \leq c_k) = P(x(t_1 + h) \leq c_1, \dots, x(t_k + h) \leq c_k) \quad (2.7)$$

Destas distribuições, pode-se concluir que os momentos e a autocovariância de x devem ser constantes, independentemente do deslocamento no tempo.

No entanto, esta propriedade é muito difícil de ser verificada e muito forte para muitas aplicações práticas, dado que frequentemente há dados insuficientes para concluir que ela é válida. Portanto, no contexto das séries temporais, utiliza-se uma versão mais branda desta propriedade, descrita a seguir.

2.2.5 Estacionariedade Fraca

A estacionariedade fraca é uma versão menos rígida da autocorrelação estrita, e apenas considera os valores de dois momentos (média e variância). Uma série temporal fracamente estacionária satisfaz as duas propriedades abaixo:

- A média (ou primeiro momento) é constante e independente do tempo;
- A autocovariância $r(t, k)$ depende apenas do módulo do atraso k , sendo também independente em relação ao tempo.

A partir daqui, o termo *estacionariedade* será usado como sinônimo de *estacionariedade fraca*.

2.2.6 Autocorrelação Parcial

A autocorrelação parcial é uma medida adicional que visa eliminar dependências entre termos subsequentes na sequência de uma série temporal. Como será visto com detalhes no próximo capítulo, se o valor de uma série temporal depende de uma quantidade de valores anteriores, ela possuirá uma dependência indireta de valores anteriores a estes.

Para exemplificar este efeito, seja $x(t)$ uma variável cujo valor depende do instante. Suponha que ela possa ser calculada da seguinte forma:

$$x(t) = 0.9x(t - 1) + 2. \quad (2.8)$$

Veja que $x(t - 1)$ pode ser escrita utilizando a mesma recorrência, o que altera a maneira de calcular $x(t)$:

$$x(t) = 0.9(0.9x(t - 2) + 2) + 2. \quad (2.9)$$

$$x(t) = 0.81x(t - 2) + 3.8. \quad (2.10)$$

Esta relação poderia ser desenvolvida mais vezes, elaborando esta dependência indireta em relação a termos anteriores. Esta influência faz com que, em séries que

apresentam dependências em relação a valores anteriores, também chamadas de séries *autorregressivas*, os valores da autocorrelação sejam influenciados pela autocorrelação de termos anteriores a estes. Como será visto adiante, isto dificulta a análise de modelos deste tipo, pois torna difícil identificar a quantidade de instantes anteriores que de fato explicam adequadamente o seu comportamento.

A métrica de autocorrelação parcial é definida de modo a eliminar estas dependências intermediárias. Ela será definida para uma série temporal estacionária de média nula (caso a média da mesma não seja nula, basta subtrair o valor da média da série de todos os seus elementos e realizar o mesmo tratamento).

Para isto, será necessário apresentar resumidamente a técnica de regressão linear.

Descrição Geral de uma Regressão Linear

Este algoritmo estima a relação linear entre uma variável de saída e um conjunto de variáveis de entrada. Seja um vetor de entradas $\mathbf{x}^T = [x_1, x_2, \dots, x_p]$. A partir dele, estima-se o valor da variável de saída y como:

$$\hat{y} = a_0 + \sum_{i=1}^p a_i x_i, \quad (2.11)$$

onde os coeficientes a_i correspondem aos pesos das variáveis x_i , e a_0 é um valor arbitrário, chamado de coeficiente linear ou *bias*.

A estimativa da relação é feita ajustando-se os coeficientes a_i . O critério de desempenho que guia o algoritmo é o erro quadrático (EQ); esta técnica calcula os parâmetros que o minimizam. Considerando a existência de uma variável de saída y , que haja n pares entrada-saída (y_i, x_i) , correspondentes aos dados disponíveis para o treinamento do modelo, e que as estimativas calculadas para as saídas y_i a partir das entradas x_i sejam \hat{y}_i , o EQ pode ser escrito como:

$$EQ(\mathbf{a}) = \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (2.12)$$

ou, em termos da operação de valor esperado:

$$EQ(\mathbf{a}) = E[(y_i - \hat{y}_i)^2]. \quad (2.13)$$

O algoritmo será descrito em mais detalhes no Capítulo 4.

Definição da Autocorrelação Parcial

Seja $\hat{x}(t+k)$, para $k \geq 2$, um estimador de $x(t+k)$ por meio de uma regressão linear em função de $[x(t+k-1), x(t+k-2), \dots, x(t+1)]$, de modo que:

$$\hat{x}(t+k) = \beta_1 x(t+k-1) + \beta_2 x(t+k-2) + \dots + \beta_{k-1} x(t+1).$$

Considerando a regressão linear de $x(t)$ em função dos mesmos valores:

$$\hat{x}(t) = \beta_1 x(t+k-1) + \beta_2 x(t+k-2) + \dots + \beta_{k-1} x(t+1).$$

Dado que x é estacionário, os coeficientes são os mesmos nas duas equações. Pode-se definir, então, a autocorrelação parcial para um atraso $k = 1, 2, \dots$ como:

$$\phi_k = r(x(t+1), x(t)) = r(x, 1), k = 1 \quad (2.14)$$

$$\phi_k = r(x(t+k) - \hat{x}(t+k), x(t) - \hat{x}(t)), k \geq 2. \quad (2.15)$$

Este procedimento faz com que os valores de autocorrelação calculados aqui não tenham influência de valores de instantes entre os valores comparados.

2.2.7 Correlação Cruzada

A correlação cruzada é uma métrica para analisar a dependência entre duas séries temporais. Ela é também baseada na métrica da correlação de Pearson. Para defini-la formalmente, sejam $s_1(t)$ e $s_2(t)$ duas séries temporais sincronizadas (ou seja, que se atualizam no mesmo instante) e de mesma frequência. A correlação cruzada, pode, então, ser definida como a correlação entre $s_1(t)$ e $s_2(t-k)$, sendo k um atraso arbitrário:

$$r(s_1(t), s_2(t-k)) = r(s_1, s_2, k) = \frac{C_{s_1(t), s_2(t-k)}}{\sigma_{s_1(t)} \sigma_{s_2(t-k)}}, \quad (2.16)$$

onde $C_{s_1(t), s_2(t-k)}$ é a covariância entre $s_1(t)$ e $s_2(t-k)$. Os limiares de significância são calculados exatamente da mesma maneira que na correlação convencional.

2.2.8 Ruído Branco Gaussiano

O ruído branco é um modelo matemático que gera um sinal aleatório. O fato de o sinal gerado ser aleatório significa que o seu valor em um instante qualquer é aleatório e *independente* dos valores nos instantes anteriores. Isto gera uma série temporal

simples. Pode-se representar este comportamento com qualquer distribuição de probabilidade; uma das mais utilizadas é a distribuição normal, sendo o ruído chamado, neste caso, de ruído branco gaussiano. Matematicamente, o ruído branco gaussiano é definido como:

$$w(t) \sim N(0, \sigma_w^2). \quad (2.17)$$

Utiliza-se como média o valor 0, e como variância um valor ajustável, a depender da intensidade desejada.

2.2.9 Modelo de Média Móvel (MA)

O modelo MA (do inglês *Moving Average*) indica que a saída a ser prevista depende linearmente do valor atual e de uma quantidade de valores anteriores de termos de erro. A quantidade de termos considerados no modelo, n , é chamada de ordem.

Os termos de erro ($\epsilon(t)$) são independentes e identicamente distribuídos (possuem uma mesma distribuição de probabilidade). Todos são modelados como ruídos brancos gaussianos e apresentam o mesmo valor de variância.

Considerando que a variável de saída seja $y(t)$, que o erro atual seja $\epsilon(t)$ e que os coeficientes a_i sejam os pesos atribuídos a cada valor passado do erro ($\epsilon(t - i)$, para algum i inteiro), pode-se expressar um modelo MA de ordem n como:

$$\hat{y}(t) = \omega + \epsilon(t) + \sum_{i=1}^n a_i \epsilon(t - i), \quad (2.18)$$

onde o termo ω é um valor fixo que é ajustado pelo modelo, e também é chamado de viés (*bias*) ou coeficiente linear. Seu valor é igual à média dos valores da série temporal.

2.2.10 Modelo Não-linear de Média Móvel (NMA)

Assim como o modelo MA, o NMA (*Non-linear Moving Average*) é um modelo de série temporal de uma variável, que determina que a variável desejada é uma função de termos de ruído branco gaussiano. No entanto, a função é não-linear. Um modelo NMA de ordem n pode ser expresso como:

$$\hat{y}(t) = \phi[\epsilon(t), \epsilon(t - 1), \dots, \epsilon(t - n)], \quad (2.19)$$

onde ϕ é uma função não-linear que deve ser determinada.

2.2.11 Modelo Autorregressivo (AR)

O modelo AR (*AutoRegressive*) também é um modelo de série temporal univariável. Diferentemente do MA, este modelo relaciona o valor atual de uma variável a uma quantidade n de valores anteriores da mesma variável e a um termo de ruído (considerado o erro atual). Um modelo AR de ordem n é descrito formalmente como:

$$\hat{y}(t) = \omega + \sum_{i=1}^n a_i y(t-i) + \epsilon(t). \quad (2.20)$$

O modelo AR possui três propriedades:

- Os erros $\epsilon(t)$ são independentes e identicamente distribuídos, como no modelo MA;
- Os erros $\epsilon(t)$ são independentes de $x(t)$;
- Os valores de a_i são menores que 1;
- A série temporal $[y(1), y(2), \dots]$ é fracamente estacionária.

Para predições de termos futuros, que são o interesse deste trabalho, o termo de erro ϵ é considerado a média da sua distribuição de probabilidade, ou seja, 0.

2.2.12 Modelo Não-linear Autorregressivo (NAR)

O NAR (*Non-linear AutoRegressive*), como o modelo AR, expressa o valor de uma variável em um instante t em função dos valores da mesma em n instantes anteriores e de um termo de ruído. A função, neste caso, é uma função não-linear ϕ . Um modelo NAR de ordem n pode ser representado como:

$$\hat{y}(t) = \phi[y(t-1), \dots, y(t-n), \epsilon(t)]. \quad (2.21)$$

2.2.13 Modelo ARMA

Os modelos ARMA (*AutoRegressive Moving Average*) unem as características dos modelos MA e AR, representando a variável objetivo como função dos seus valores passados e dos valores atual e passados dos erros. Desta maneira, o modelo ARMA com a termos autorregressivos e b termos de média móvel pode ser explicado pela equação:

$$\hat{y}(t) = \omega + \epsilon(t) + \sum_{i=1}^a a_i y(t-i) + \sum_{j=1}^b b_j \epsilon(t-j), \quad (2.22)$$

na qual os coeficientes a_x correspondem aos pesos das saídas anteriores do sistema e os coeficientes b_x correspondem aos pesos atribuídos aos erros de predição.

2.2.14 Modelo NARMA

O modelo NARMA (*Non-linear AutoRegressive Moving Average*) é a versão não-linear do ARMA. Um modelo $NARMA(a, b)$ é explicado pela equação:

$$\hat{y}(t) = \phi[y(t-1), \dots, y(t-a), \epsilon(t), \epsilon(t-1), \dots, \epsilon(t-b)], \quad (2.23)$$

onde ϕ é alguma função não-linear que pode ser aplicada.

2.2.15 Modelos com Entradas Exógenas

Uma possibilidade adicional é a dependência das variáveis objetivo em relação a uma outra grandeza, chamada de variável exógena. Esta variável é também uma série temporal, e a dependência em relação a ela pode se manifestar em relação ao valor atual e/ou em relação a valores anteriores.

Um fato interessante sobre inserir variáveis exógenas é que isto permite inserir duas séries não estacionárias na previsão de uma das duas. Desde que alguma combinação linear de termos das duas séries forme uma série estacionária, as propriedades exigidas dos modelos discutidos anteriormente não são violadas. Quando isto ocorre, as duas séries são consideradas *cointegradas*.

A dependência de uma série temporal auxiliar, adicionada aos modelos anteriores, gera uma nova família de modelos. A quantidade de termos da entrada exógena é chamada de *ordem* dos termos exógenos. Considerando que a variável exógena seja $x(t)$ e que as ordens sejam respectivamente a , b e c para termos autorregressivos, de média móvel e exógenos, os modelos desta família são:

Modelo MAX (*Moving Average with eXogenous inputs*)

Obtido através da inserção de termos exógenos a um modelo MA convencional, é descrito pela equação:

$$\hat{y}(t) = \omega + \epsilon(t) + \sum_{i=1}^b a_i \epsilon(t-i) + \sum_{i=0}^c c_i x(t-i). \quad (2.24)$$

Modelo NMAX (*Non-linear Moving Average with eXogenous inputs*)

Obtido através da inclusão de termos da variável exógena a um modelo NMA, o modelo NMAX estima a série temporal objetivo por uma função não-linear de termos de ruído e uma série temporal exógena, como descrito na equação:

$$\hat{y}(t) = \phi[\epsilon(t), \epsilon(t-1), \dots, \epsilon(t-b), x(t), x(t-1), \dots, x(t-c)]. \quad (2.25)$$

Modelo ARX (*AutoRegressive with eXogenous inputs*)

O modelo ARX é gerado através da inclusão de termos exógenos a um modelo AR. É descrito pela equação abaixo:

$$\hat{y}(t) = \omega + \epsilon(t) \sum_{i=1}^a a_i y(t-i) + \sum_{i=1}^c c_i x(t-i) + \epsilon(t). \quad (2.26)$$

Modelo NARX (*Non-linear AutoRegressive with eXogenous inputs*)

Gerado através da inclusão de uma série temporal auxiliar a um modelo NAR. Descrito pela equação:

$$\hat{y}(t) = \phi[y(t-1), \dots, y(t-a), x(t), x(t-1), \dots, x(t-c), \epsilon(t)]. \quad (2.27)$$

Modelo ARMAX (*AutoRegressive Moving Average with eXogenous inputs*)

O ARMAX é um modelo ARMA com a inclusão de termos exógenos:

$$\hat{y}(t) = \omega + \epsilon(t) + \sum_{i=1}^a a_i y(t-i) + \sum_{i=1}^b b_i x(i) + \sum_{i=1}^c c_i \epsilon(t-i). \quad (2.28)$$

Modelo NARMAX (*Non-linear AutoRegressive Moving Average with eXogenous inputs*)

Por fim, o modelo NARMAX é um modelo NARMA com a inclusão de termos exógenos, descrevendo os valores de uma série temporal como funções de valores anteriores, termos de ruído branco gaussiano e termos de uma série temporal exógena. Descrito pela equação abaixo:

$$\hat{y}(t) = \phi[y(t-1), \dots, y(t-a), \epsilon(t), \epsilon(t-1), \dots, \epsilon(t-b), x(t), x(t-1), \dots, x(t-c)]. \quad (2.29)$$

2.2.16 Regressão Dinâmica

Uma alternativa aos modelos (N)ARX pode ser empregada quando não há dependência entre a variável que se deseja prever e seus valores do passado. Logo, a previsão depende apenas de alguma quantidade de termos da série exógena. Este modelo é representado pela equação:

$$\hat{y}(t) = \omega + \sum_{i=1}^n a_i x(t-i) + \epsilon(t). \quad (2.30)$$

Como todos os outros modelos, este supõe a presença de um termo de erro de predição que corresponde a um ruído branco gaussiano.

2.2.17 Modelos Integrados

Para manter a condição de estacionariedade, em alguns casos é necessário realizar transformações na série temporal que se deseja prever. Por exemplo, em alguns casos, o valor da média varia com o tempo. Para se remover este *comportamento de média móvel*, pode-se aplicar a operação de *diferenciação*. Esta realiza uma transformação no conjunto de valores da série temporal, criando uma série de diferenças entre o valor atual e o valor anterior. A diferenciação pode ser aplicada mais de uma vez, sendo a quantidade aplicada chamada de *ordem* da diferenciação. Matematicamente, a diferenciação de ordem 1 ou *primeira diferença* é definida como:

$$\Delta_1 y(t) = y(t) - y(t - 1). \quad (2.31)$$

Uma diferenciação de ordem 2 ou *segunda diferença* pode ser escrita como:

$$\Delta_2 y(t) = \Delta_1 y(t) - \Delta_1 y(t - 1) = y(t) - 2y(t - 1) + y(t - 2), \quad (2.32)$$

Pode-se generalizar esta operação para uma ordem arbitrária. Uma diferenciação de ordem d ou *d-ésima diferença* pode ser escrita como:

$$\Delta_d y(t) = \Delta_{d-1} y(t) - \Delta_{d-1} y(t - 1) \quad (2.33)$$

Um modelo de séries temporais aplicado sobre uma série diferenciada também é chamado de modelo *integrado*.

2.2.18 Modelos com Erros Dinâmicos

Em alguns cenários, é interessante modelar o próprio termo de erro de predição como uma série temporal com termos AR e/ou MA. Isto é determinado após o ajuste de um modelo, verificando o comportamento do erro de predição obtido e executando sua análise com as métricas de autocorrelação. Caso o próprio erro apresente de fato um comportamento autorregressivo ou de média móvel, ele mesmo pode ser modelado com algum dos modelos lineares ou não-lineares não-exógenos discutidos anteriormente.

Os casos em que isto pode ser necessário são os casos em que há informações não representadas pelas variáveis utilizadas, que se refletem num comportamento do erro que depende do passado;

2.2.19 Classificação dos Modelos de Séries Temporais

As Tabelas 2.1 e 2.2 apresentam uma classificação resumida de todos os modelos apresentados nesta seção, sendo as ordens representadas pelos valores a para termos autorregressivos, b para termos de média móvel, c para variáveis exógenas e d para diferenciações.

Tabela 2.1: Classificação dos modelos lineares de Séries Temporais.

Nome	Termos AR	Integrado	Média Móvel	Exógeno
MA(b)	Não	Não	Sim	Não
IMA(b,c)	Não	Sim	Sim	Não
MAX(b,c)	Não	Não	Sim	Sim
IMAX(b,c)	Não	Sim	Sim	Sim
AR(a)	Sim	Não	Não	Não
ARI(a, d)	Sim	Sim	Não	Não
ARX(a, c)	Sim	Não	Não	Sim
ARIX(a, d, c)	Sim	Sim	Não	Sim
ARMA(a, b)	Sim	Não	Sim	Não
ARIMA(a, d, b)	Sim	Sim	Sim	Não
ARMAX(a, b, c)	Sim	Não	Sim	Sim
ARIMAX(a, d, b, c)	Sim	Sim	Sim	Sim

Tabela 2.2: Classificação dos modelos não-lineares de Séries Temporais.

Nome	Termos AR	Integrado	Média Móvel	Exógeno
NMA(b)	Não	Não	Sim	Não
NIMA(b,c)	Não	Sim	Sim	Não
NMAX(b,c)	Não	Não	Sim	Sim
NIMAX(b,c)	Não	Sim	Sim	Sim
NAR(a)	Sim	Não	Não	Não
NARI(a, d)	Sim	Sim	Não	Não
NARX(a, c)	Sim	Não	Não	Sim
NARIX(a, d, c)	Sim	Sim	Não	Sim
NARMA(a, b)	Sim	Não	Sim	Não
NARIMA(a, d, b)	Sim	Sim	Sim	Não
NARMAX(a, b, c)	Sim	Não	Sim	Sim
NARIMAX(a, d, b, c)	Sim	Sim	Sim	Sim

Dadas as quantidades de tipos de modelos e de parâmetros que podem ser variados, uma análise exaustiva é computacionalmente muito pesada. Por isto, antes de testar os modelos, analisa-se o comportamento de uma série temporal a ser modelada com relação às métricas de correlação, para eliminar candidatos a tipos de modelo que não representem os dados.

2.3 Seleção de Modelos

Para escolher o modelo estático e dinâmico adequado, é necessário determinar a dependência das variáveis objetivo em relação aos seus valores anteriores e às variáveis que podem ser usadas como entrada. Além disso, pode ser necessário tratar tendências, comportamentos que violam a estacionariedade de uma série, dado que prejudicam a análise destas dependências. Por isto, esta seção analisa:

- Maneiras de tratar tendências observáveis através do comportamento temporal de uma série;
- As métricas de autocorrelação e autocorrelação parcial para os tipos de modelos descritos anteriormente, apresentando simulações dos modelos anteriores acompanhadas de autocorrelogramas e autocorrelogramas parciais, para que as propriedades de cada modelo possam ser compreendidas de maneira gráfica com dados experimentais;
- Como verificar a dependência de uma série em relação a outras séries temporais e escolher os termos com os atrasos necessários a partir da correlação cruzada;

2.3.1 Tratamento de Tendências

A partir de um gráfico que exhibe os valores de uma série temporal em cada instante, em alguns casos pode-se observar comportamentos que deslocam a média dos valores da série ou afetam as autocorrelações, fazendo com que estas não sejam constantes com o tempo. Média e autocorrelações variáveis violam a propriedade de estacionariedade, que é importante para a definição dos modelos de séries temporais definidos anteriormente. Caso esta propriedade seja violada, os modelos poderão apresentar estimativas imprecisas, e a análise das métricas de autocorrelação é prejudicada. A seguir, são descritos os comportamentos mais comuns, como observá-los e formas de tratá-los.

Comportamento de Média Móvel

Gráficos de séries temporais normalmente apresentam oscilações em torno do seu valor médio. No entanto, em algumas séries, o valor em torno do qual as oscilações ocorrem apresenta um comportamento variável com o tempo, o que gera um *comportamento de média móvel*, tornando a média da série dependente do tempo. Este comportamento não deve ser confundido com o modelo MA, que é uma série estacionária gerada a partir de uma soma ponderada de ruídos brancos. O comportamento de média móvel pode ser linear ou não-linear. Há duas maneiras de se resolver este comportamento:

- Diferenciação: remove tendências lineares. Caso a série resultante exiba outros comportamentos de média móvel, pode-se aplicar outras diferenciações. Diferenciações excessivas podem causar perdas de informações no modelo, adicionando dependências do passado que não existiam anteriormente e prejudicando a acurácia das estimativas;
- Uso de uma função do tempo de execução como variável: caso diferenciações não funcionem ou caso o comportamento de média móvel seja não-linear, pode-se aplicar uma função do tempo de execução como variável. Para ajustar o peso do tempo de execução, utiliza-se algoritmos de Aprendizado de Máquina de regressão, como os que serão vistos no Capítulo 3.

Amplitudes Variáveis

O comportamento de amplitude variável afeta o valor da autocorrelação, tornando-a dependente do tempo, o que viola a estacionariedade. Para tratar este comportamento, pode-se aplicar uma função não-linear sobre a série temporal. Funções muito utilizadas são a raiz quadrada e o logaritmo.

Múltiplas Tendências

Uma possibilidade em uma série temporal é a existência de diversas tendências dos dois tipos mencionados anteriormente, com intensidades diferentes. Neste caso, pode-se:

- Separar o conjunto de dados, caso se disponha de variáveis relacionadas a estes comportamentos, e utilizar um modelo independente por parte do conjunto de dados, tratando-se as tendências separadamente;
- Acrescentar como variáveis do modelo uma função do tempo de execução por tendência observada. Cada função deve apresentar valores não-nulos apenas no intervalo de ocorrência de cada tendência.

Uso de Variáveis Exógenas

Conforme mencionado na Seção 2.2.15 (Modelos com Entradas Exógenas), caso alguma combinação linear entre a série que se deseja prever e uma outra série temporal utilizada como termo exógeno seja estacionária, um modelo válido pode ser utilizado com as duas séries não-estacionárias sem prejuízos às estimativas.

2.3.2 Análise da Autocorrelação

Esta subseção analisa a métrica de autocorrelação para os modelos MA, AR e ARMA, de modo que seja possível identificar qual tipo de modelagem escolher para representar o comportamento de uma série temporal a partir de um autocorrelograma.

Modelos MA

Seja um modelo MA de ordem arbitrária q , ou $MA(q)$:

$$\hat{x}(t) = \omega + \epsilon(t) + \sum_{i=1}^q a_i \epsilon(t - i). \quad (2.34)$$

É demonstrado no Apêndice B.1 que a autocorrelação de x considerando-se um atraso k pode ser escrita como

$$r(x, k) = \frac{\sigma_\epsilon^2 \sum_{i=k+1}^q a_{i-k} a_i}{\sigma_x^2}. \quad (2.35)$$

Vale ressaltar que, neste somatório, podem aparecer um termo a_0 , um termo a_{q+1} e termos a com índices negativos. $a_0 = 1$, dado que este é o peso dado para o valor de erro com atraso nulo, como pode ser visto na definição do modelo MA; e $a_i = 0$ para qualquer valor de i superior a q ou inferior a 0, dado que o modelo apenas define pesos para atrasos que variem apenas entre 0 e q .

Quando $k > q$, ou seja, quando o atraso supera a ordem do modelo, o valor do somatório é 0, e, por conseguinte, a autocorrelação é nula. Isto torna possível identificar a existência de um modelo MA e a sua ordem através de um autocorrelograma.

Modelos AR

Seja um modelo AR de ordem arbitrária q , ou $AR(q)$:

$$x(t) = \omega + \sum_{i=1}^q a_i x(t - i) + \epsilon(t). \quad (2.36)$$

É demonstrado no Apêndice B.2 que pode-se calcular a autocorrelação por:

$$r(x, k) = \sum_{i=1}^q a_i r(x, k - i). \quad (2.37)$$

Estas equações para a autocorrelação geram um método recursivo de calculá-la. Também são chamadas de Equações de Yule-Walker [10].

Uma análise dos valores dos coeficientes a_i ajuda a descrever o comportamento

do valor da autocorrelação. É demonstrado no Apêndice B.3, através do cálculo da variância para o modelo AR, que $|a_i| < 1$. Com este resultado, é possível obter diversas conclusões sobre a autocorrelação do modelo AR (Equação 2.37):

- Seu valor não diminui abruptamente para 0 quando o número de atrasos supera a ordem do modelo AR ($k > q$), diferentemente do modelo MA.
- Existe uma dependência linear entre um valor de autocorrelação e q valores de autocorrelações com atrasos menores.
- Como $r(x, 0) = 1$ e os coeficientes a_i são menores que 1 em módulo, o módulo de r apresenta um comportamento decrescente em relação a k .
- Se todos os coeficientes a_i forem negativos, a autocorrelação alterna de sinal a cada unidade de atraso.

Modelos ARMA

De acordo com Shumway [10], para modelos com termos autorregressivos e de média móvel, o comportamento da autocorrelação é uma composição dos comportamentos do modelo AR e do modelo MA. Assim, sendo um modelo ARMA(a, b), com ordem autorregressiva a e ordem de média móvel b , definido por:

$$x(t) = \omega + \sum_{i=1}^a a_i x(t-i) + \sum_{i=1}^b b_i \epsilon(t-i) + \epsilon(t), \quad (2.38)$$

a autocorrelação não apresenta quedas abruptas para 0, mesmo quando o atraso supera as ordens de autorregressão e média móvel. O seu comportamento apresenta uma tendência decrescente. Este padrão é o mesmo da autocorrelação de um modelo AR.

2.3.3 Análise da Autocorrelação Parcial

Esta subseção analisa a autocorrelação parcial, de modo que, combinando as informações obtidas com esta métrica com o comportamento da autocorrelação, possa-se determinar o tipo de modelo que os dados experimentais seguem.

Modelos AR

Um modelo AR de ordem arbitrária q pode ser estimado por meio de um algoritmo de regressão linear. Além disso, em [10] é demonstrado que pode-se representá-lo em função dos valores de autocorrelação parcial da seguinte forma (considerando

que a média seja nula):

$$\hat{x}(t) = \sum_{i=1}^q \phi_i x(t-i). \quad (2.39)$$

Desta forma, os valores de autocorrelação parcial também podem ser obtidos através dos coeficientes da regressão linear. Pode-se concluir que, em um modelo AR(q), o valor de ϕ_i é não-nulo para valores de i menores ou iguais a q , e torna-se 0 quando $i > q$. Portanto, a autocorrelação parcial assume o comportamento contrário da autocorrelação para modelos AR.

Modelos MA

É demonstrado em [10] que um modelo MA pode ser representado por uma soma infinita de termos autorregressivos:

$$x(t) = - \sum_{i=1}^{\infty} a_i x(t-i) + \epsilon_t. \quad (2.40)$$

Esta representação permite que um modelo MA também seja exibido em função da autocorrelação parcial:

$$x(t) = - \sum_{i=1}^{\infty} \phi_i x(t-i) + \epsilon_t, \quad (2.41)$$

o que permite que, independentemente do modelo, seja possível extrair a autocorrelação parcial através de um algoritmo de regressão linear. Deste resultado, vê-se que o valor de ϕ_i não se torna 0 quando $i > q$, dado que uma quantidade infinita de termos passados de x influencia no valor de $x(t)$. Novamente, conclui-se que o comportamento da autocorrelação parcial é o contrário do comportamento da autocorrelação.

Modelos ARMA

De acordo com Shumway [10], para modelos com termos autorregressivos e de média móvel, o comportamento da autocorrelação parcial também é uma composição dos comportamentos do modelo AR e do modelo MA. Assim, sendo um modelo ARMA(a, b), com ordem autorregressiva a e ordem de média móvel b , definido por:

$$x(t) = \omega + \sum_{i=1}^a a_i x(t-i) + \sum_{i=1}^b b_i \epsilon(t-i) + \epsilon(t), \quad (2.42)$$

o valor da autocorrelação parcial não apresenta quedas abruptas para 0, mesmo quando o atraso supera as ordens de autorregressão e média móvel. A autocorrelação

parcial apresenta uma tendência decrescente. Este comportamento é o mesmo da autocorrelação parcial de um modelo MA.

2.3.4 Comportamentos Simulados dos Modelos Teóricos Não-Exógenos

Esta subseção realiza simulações com exemplos de modelos MA, AR e ARMA para verificar as propriedades dos seus autocorrelogramas e autocorrelogramas parciais com dados experimentais. Há duas finalidades para simular os modelos teóricos:

- Obter comportamentos de autocorrelações e autocorrelações parciais com uma quantidade de valores da série temporal que seja finita, conhecida e próxima da quantidade de valores do conjunto de dados a ser analisado, a fim de utilizá-los como referências para a análise dos dados de posição geográfica e sensores realizada no Capítulo 4;
- Verificar se há divergências entre o comportamento teórico apresentado na literatura e um resultado simulado, dado que a quantidade de valores gerados é finita e que o ruído branco é uma variável aleatória.

Os modelos simulados são:

- Modelo MA(2): descrito pela equação $x(t) = 0.52\epsilon(t-1) - 0.34\epsilon(t-2) + \epsilon(t)$, sendo ϵ um ruído branco gaussiano de média nula e variância $\sigma_\epsilon^2 = 9$;
- Modelo AR(2): descrito pela equação $x(t) = 0.81x(t-1) - 0.74x(t-2) + \epsilon(t)$, sendo ϵ um ruído branco gaussiano de média nula e variância $\sigma_\epsilon^2 = 9$. Os dois valores iniciais da série são 1.3 e -2.7;
- Modelo ARMA(2, 2): descrito pela equação $x(t) = 0.81x(t-1) - 0.74x(t-2) + 0.52\epsilon(t-1) - 0.34\epsilon(t-2) + \epsilon(t)$. ϵ aqui também é um ruído branco gaussiano de média nula e variância $\sigma_\epsilon^2 = 9$. Os dois valores iniciais da série são 1.3 e -2.7.

Os coeficientes e ordens empregados aqui não apresentam relações com os dados do problema a ser resolvido neste trabalho; são arbitrários. Para as simulações dos três modelos, foram gerados 3000 valores de x . As simulações foram elaboradas na linguagem Python, com o auxílio da biblioteca *random*, já presente na biblioteca padrão da linguagem. Os valores de autocorrelação foram calculados com módulos da biblioteca Statsmodels [12].

As Figuras 2.1, 2.2 e 2.3 exibem, respectivamente, os autocorrelogramas dos modelos MA, AR e ARMA simulados. O eixo horizontal representa o atraso, e o

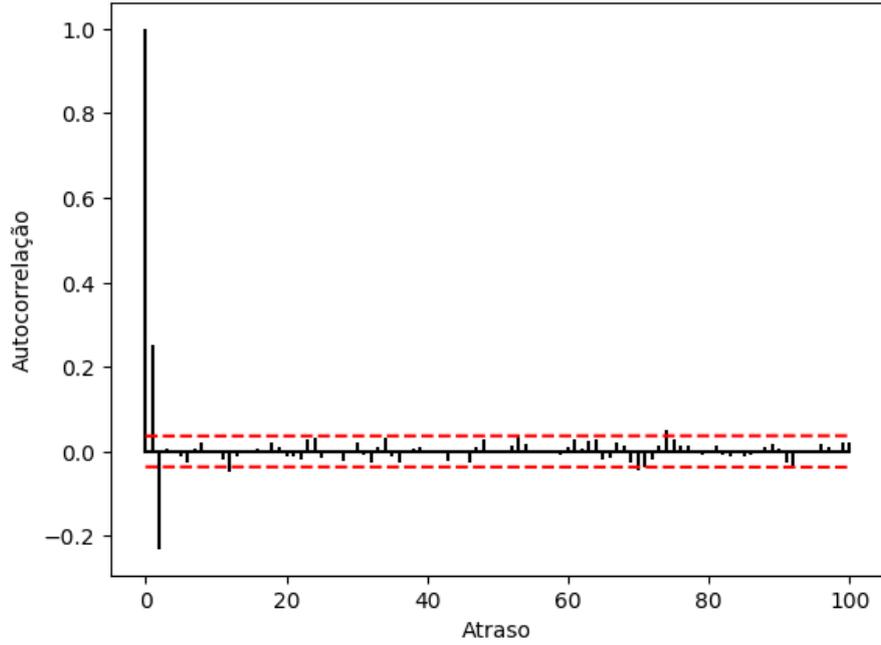


Figura 2.1: Autocorrelação do modelo MA(2) simulado.

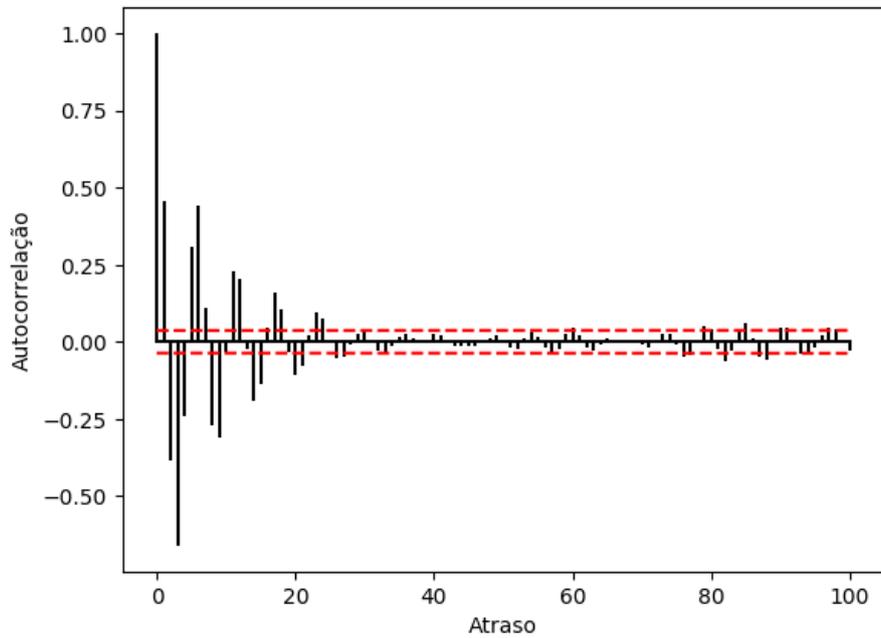


Figura 2.2: Autocorrelação do modelo AR(2) simulado.

eixo vertical apresenta o valor da autocorrelação. Os gráficos incluem o valor para o atraso 0, que é sempre 1.

Os dois gráficos indicam comportamentos diferentes dependendo do tipo de modelo, como discutido anteriormente. No caso do modelo MA simulado, pode-se ver que o valor da autocorrelação apenas está fora dos limiares de significância no caso dos atrasos 1 e 2, o que está de acordo com a sua equação geradora e com a dis-

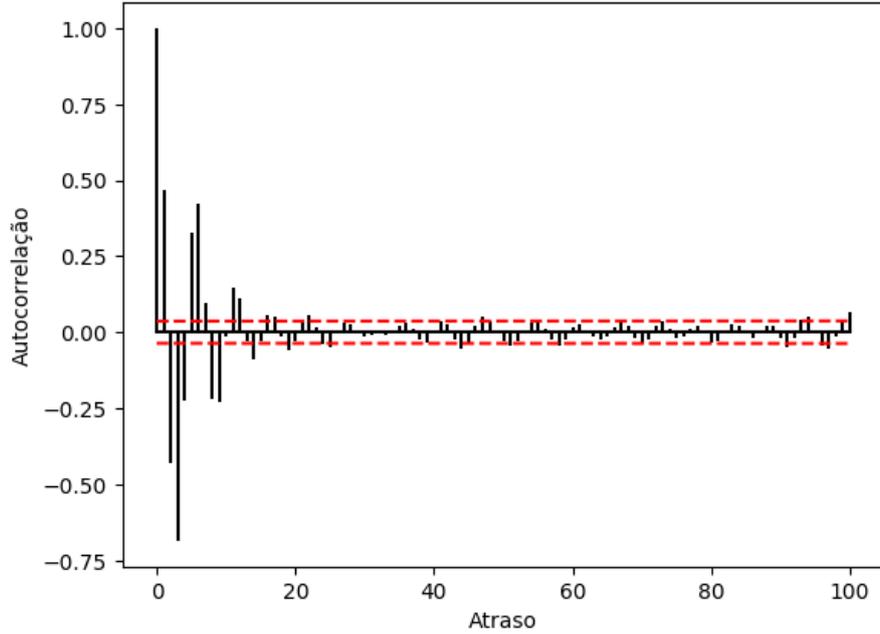


Figura 2.3: Autocorrelação do modelo ARMA(2,2) simulado.

cussão teórica. Há um pequeno pico de significância em um valor muito mais alto de atraso; esta é uma possibilidade ao se trabalhar com dados experimentais. Este pico deve ser ignorado, não só por ele estar em um atraso muito maior que 1 e 2, mas também porque picos de significância que não são adjacentes não refletem o comportamento de um modelo MA. A partir deste resultado, um MA(2) parece ser um bom aproximador para os dados gerados. Para confirmar isto, deve-se utilizar um autocorrelograma parcial.

No caso da simulação do AR(2), pode-se perceber a presença de valores significativos em diversos valores de atraso maiores que 2 e que o módulo da autocorrelação apresenta um comportamento decrescente, sem quedas abruptas de valor. Estas características indicam que o modelo adequado não é um modelo MA. De acordo com as discussões anteriores, o modelo provavelmente apresenta termos autorregressivos. Isto deve ser confirmado com o uso de um autocorrelograma parcial. Além disto, a presença de um coeficiente negativo causa o comportamento oscilatório apresentado. É possível verificar que o decaimento faz com que os valores de autocorrelação sejam não significativos a partir de um atraso finito. A intensidade do decaimento da autocorrelação depende dos coeficientes dos termos autorregressivos.

Para a simulação do ARMA(2, 2), o comportamento da autocorrelação é muito similar ao do AR(2), apresentando uma tendência decrescente do seu módulo e picos de significância para atrasos maiores que 2, o que está de acordo com o referencial teórico. Isto torna difícil diferenciar os dois tipos de modelos a partir de somente um autocorrelograma.

As Figuras 2.4, 2.5 e 2.6 exibem os autocorrelogramas parciais, respectivamente, dos modelos MA, AR e ARMA simulados. Novamente, o eixo horizontal representa o atraso e o eixo vertical apresenta o valor da autocorrelação parcial.

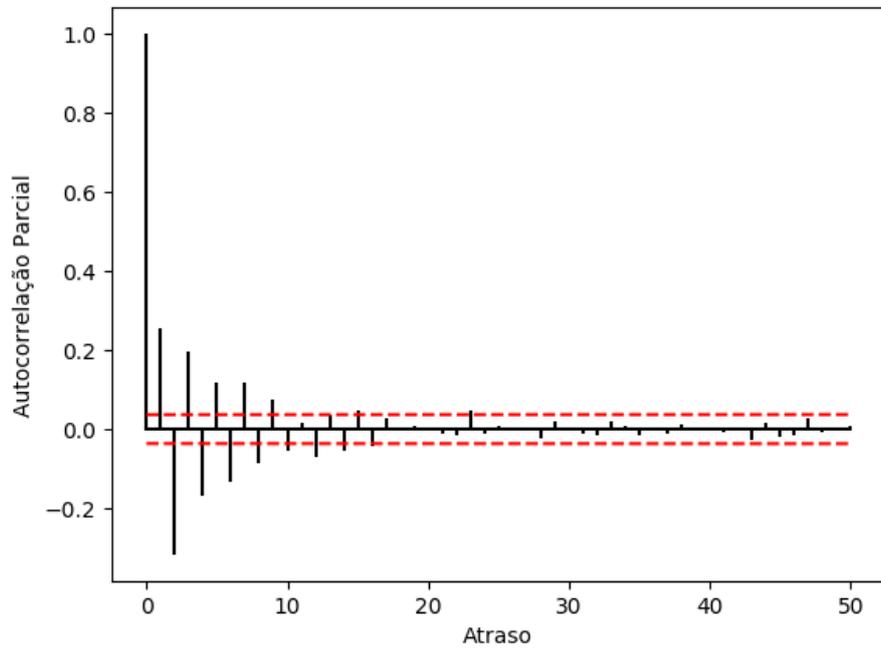


Figura 2.4: Autocorrelação parcial do modelo MA(2) simulado.

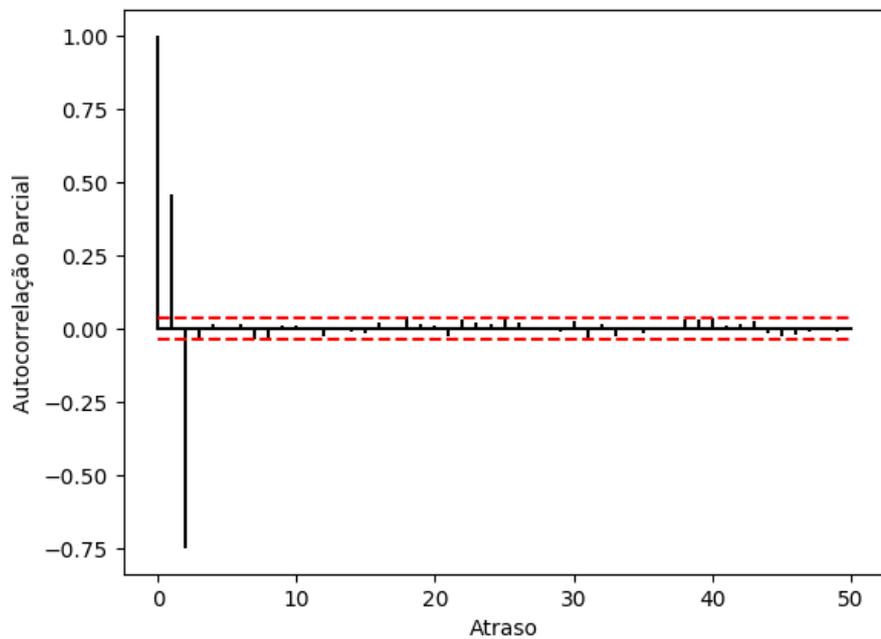


Figura 2.5: Autocorrelação parcial do modelo AR(2) simulado.

Pode-se perceber, novamente, uma diferença nos comportamentos. No modelo MA simulado, o comportamento do módulo da autocorrelação parcial é decrescente e sem quedas abruptas, e há valores significativos além do atraso 2. A quantidade

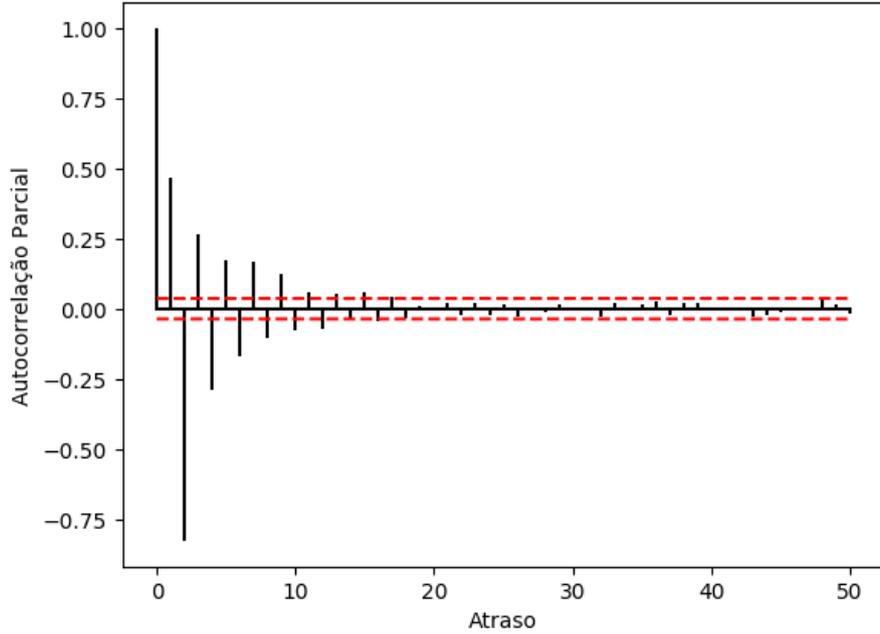


Figura 2.6: Autocorrelação parcial do modelo ARMA(2, 2) simulado.

de valores não-nulos é finita, de modo que se poderia considerar utilizar um modelo AR. No entanto, o comportamento decrescente indica que o melhor aproximador é um modelo MA, e o perfil da autocorrelação do MA simulado, apresentado anteriormente, é um segundo indicativo de que o melhor modelo seja um MA. Portanto, o estimador mais adequado, a partir dos gráficos apresentados, é um modelo MA(2).

Na simulação do modelo AR, existe uma queda abrupta do atraso 2 para o atraso 3. Valores de atraso acima de 2 apresentam uma autocorrelação parcial não-significativa. Este é um indicativo de que um aproximador adequado é um modelo AR(2), e não um modelo de tipo ARMA, o que está de acordo com a equação que o gerou.

O comportamento da autocorrelação parcial do ARMA(2, 2) simulado é idêntico ao do MA simulado, o que também está de acordo com o referencial teórico. Ou seja, o uso dos gráficos das duas métricas de autocorrelação permite distinguir um AR de um ARMA.

O que se pode concluir a partir da análise destas simulações é que, para os três modelos, os padrões da autocorrelação e da autocorrelação com dados simulados estão de acordo com o referencial teórico apresentado. Mais ainda, os autocorrelogramas e autocorrelogramas parciais exibidos aqui servem de referência para comparar com a análise dos dados de posição geográfica e sensores realizada adiante, no Capítulo 4.

2.3.5 Análise da Correlação Cruzada

Terminada a análise dos valores de autocorrelação, deve-se verificar a dependência da série que se deseja prever em relação a outras séries temporais cuja influência na série objetivo possa existir (séries auxiliares). Para isto, utiliza-se a correlação cruzada. A análise para esta métrica segue os passos abaixo:

- Analisa-se a série auxiliar através de seu autocorrelograma;
- Calcula-se o valor da correlação cruzada entre as duas séries através da equação 2.16 para diversos atrasos;
- Com os valores, elabora-se um correlograma entre as duas séries com os limiares de significância;
- Caso haja atrasos com valores significativos de correlação cruzada, existe uma dependência. Logo, os termos atrasados significativos são candidatos que podem ser incluídos no modelo;
- Verifica-se o autocorrelograma da série auxiliar. Caso ela exiba o comportamento de um modelo AR, vários termos atrasados estarão correlacionados entre si. Neste caso, escolhe-se o termo atrasado da série auxiliar com o maior atraso. Este processo é similar ao que se faz para selecionar variáveis em problemas estáticos de Aprendizado de Máquina.

Capítulo 3

Análise dos Dados Coletados

Dado que o objetivo deste projeto é elaborar um sistema de navegação cega que realize predições da posição de um objeto caso haja uma interrupção da comunicação com um sistema de posicionamento por satélites, discutiu-se, no Capítulo anterior, o referencial teórico dado pela área de Séries Temporais, um ramo da estatística que estuda predições de valores futuros a partir de informações do passado.

Após esta discussão teórica, este Capítulo trata da análise do problema de acordo com a teoria discutida anteriormente. O objetivo é que, a partir de dados de posições geográficas e de sensores inerciais coletados experimentalmente, seja realizada uma análise de acordo com a teoria de Séries Temporais para identificar quais são os melhores tipos de modelos para o problema.

Em primeiro lugar, são caracterizados os equipamentos que se utilizou para coletar os dados, os formatos das informações obtidas e o procedimento de coleta. Por fim, os dados são analisados de acordo com o referencial teórico para verificar as dependências das posições em relação às posições anteriores e às leituras dos sensores.

3.1 Equipamentos Utilizados

Como sistema de posicionamento por satélite, foi escolhido o GPS (*Global Positioning System*). É importante ressaltar, no entanto, que a abordagem descrita aqui poderia ser aplicada a qualquer outro sistema de posicionamento por satélite, dado que todos informam posições no mesmo formato.

A versão experimental do sistema desenvolvida neste trabalho considera o ambiente veicular: ela foi desenvolvida para ser executada em um dispositivo embarcado para carros, uma unidade de bordo (ou OBU – *On-Board Unit*). O dispositivo OBU utilizado neste projeto conta com recursos para comunicação veicular (que não foram utilizados aqui, mas são úteis em diversas aplicações para veículos), conexão com o

barramento CAN (*Controller Area Network* – permite a conexão com os sensores do veículo) e um receptor GPS integrado.

O barramento CAN é um tipo de rede cabeada que permite a equipamentos compatíveis a conexão a diferentes sensores e atuadores de um carro, possibilitando analisar suas condições e identificar defeitos, bem como efetuar configurações em mecanismos do carro. A Figura 3.1 abaixo apresenta um diagrama que ilustra a conexão por este barramento.

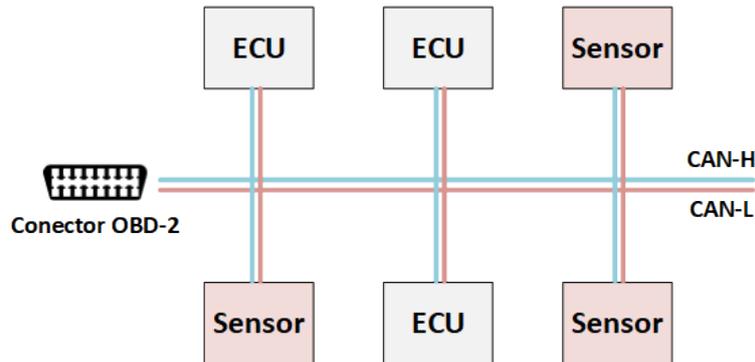


Figura 3.1: Exemplo de diagrama de conexões em um barramento CAN.

Pode-se observar na imagem o conector OBD-2 (*On-Board Diagnostics 2*), usado como interface entre o sistema e o barramento. No barramento, estão conectados sensores do veículo e as ECUs (*Electronic Control Units*), dispositivos que lêem dados de vários sensores do veículo e controlam atuadores de diferentes sistemas do veículo, como por exemplo o sistema anti-bloqueio das rodas (ABS – *Anti-Blocking System*). Há ainda sensores que permitem melhorar o desempenho do motor, ou ainda minimizar o consumo de combustível. Os fios CAN-H e CAN-L são usados para transmissão e recepção dos dados em um arranjo chamado sinalização diferencial: o valor lógico transmitido pelo barramento é calculado a partir da diferença entre as tensões dos dois fios, gerando dois estados. A vantagem da sinalização diferencial é a maior tolerância a ruídos [13].

Qualquer dos nós da rede CAN pode enviar e receber dados. Uma rede em barramento é caracterizada pelo compartilhamento de um meio cabeado entre vários dispositivos. Isto permite que qualquer dispositivo conectado possa se comunicar com qualquer outro dispositivo, porém duas ou mais transmissões não podem ser realizadas simultaneamente, dado que ocorrerá uma colisão, corrompendo os dados transmitidos. Por conta disto, o CAN segue um protocolo de controle de acesso ao meio, que garante a não-ocorrência de transmissões simultâneas.

Através do CAN, pode-se coletar informações de sensores de um veículo. No veículo utilizado, estes sensores medem as frequências de rotação das rodas do veículo e a velocidade de translação do veículo. Devido ao fato de o veículo usado possuir juntas homocinéticas, as velocidades de rotação das rodas esquerdas são diferentes

das velocidades das rodas direitas quando é realizada uma curva, pois a junta homocinética ajusta estas velocidades para que as velocidades angulares de todas as rodas sejam iguais - isto é utilizado para facilitar curvas. Por outro lado, isto pode permitir inferir a existência de curvas no trajeto do veículo a partir das leituras dos sensores. Além disto, a velocidade do carro indica quanto o veículo se movimentou, independentemente da direção. Por isso, os sensores utilizados são os que indicam as velocidades de rotação das rodas traseiras esquerda e direita e o sensor de velocidade. Estes sensores são escolhas adequadas para o sistema, dado que eles não dependem de nenhum tipo de comunicação com o exterior do veículo, garantindo o funcionamento dos mesmos nos ambientes nos quais ocorrem perdas de cobertura do GPS. Estes sensores estão disponíveis de fábrica para qualquer veículo que possua um sistema anti-bloqueio das rodas (ABS), como indicado por Daiss [14].

A Tabela 3.1 fornece as especificações dos equipamentos mencionados.

Tabela 3.1: Especificações dos equipamentos utilizados.

Unidade de Bordo (OBU)	Cohda Wireless modelo MK5
Receptor GPS	U-blox M8N, com acurácia teórica de 2.5m
Veículo	Peugeot 408 modelo 2016
Tempo de Atualização do GPS (t_{GPS})	250 ms
Tempo de Atualização dos Sensores (t_{CAN}) (determinado experimentalmente)	(49.9 +/- 0.2) ms

Vale ressaltar, nesta tabela, que o tempo de atualização dos sensores do CAN é o valor médio, com a incerteza atribuída ao desvio-padrão, obtido a partir da análise das leituras coletadas como descrito na próxima seção.

3.1.1 Formato das Informações Disponíveis

Esta seção visa explicar como estão organizados os dados obtidos de cada equipamento.

Posições

As posições, como mencionado anteriormente, são dadas por um sistema de coordenadas bidimensional. A latitude varia de 0 a 90° para norte ou para sul. No receptor GPS, a noção de direção é fornecida pelo sinal do valor da latitude. Valores positivos correspondem a pontos localizados no hemisfério Norte da Terra, e valores negativos indicam a localização no hemisfério Sul. A longitude varia de 0 a 180° para leste ou para oeste. Analogamente à latitude, a direção é dada pelo sinal. Valores positivos indicam o posicionamento no hemisfério Leste, e valores negativos correspondem ao hemisfério Oeste.

As leituras vêm acompanhadas de uma marca de tempo (ou *timestamp*), que indica quando uma determinada posição foi obtida. Por padrão, o valor é fornecido no formato *Unix time*, que indica quanto tempo passou desde a meia-noite do dia primeiro de janeiro de 1970. No caso do receptor GPS, ele indica esse valor com a precisão de centésimos de segundos, assumindo uma unidade como um segundo. Um exemplo de leitura obtida através do receptor é: 1483559537.15, -22.861166833, -43.227934, onde o primeiro campo é a marca de tempo, o segundo campo é a latitude e o último valor é a longitude.

Sensores

Os sensores instalados no veículo utilizado medem as frequências de rotação das rodas traseira esquerda e traseira direita e a velocidade de translação do carro. As frequências de rotação das rodas são fornecidas como frequências, ou seja, quantas vezes em um segundo a roda gira em torno do seu eixo. A unidade para isto é Hz. A velocidade do carro é dada em metros por segundo (m/s). A leitura destes sensores também é fornecida com uma marca de tempo, esta com uma precisão 100 vezes maior que a temporização do GPS. A medida de tempo também assume segundos como unidades. Um exemplo de leitura obtida através do CAN é: 1483559091.5779, 2.41015625, 2.3619791667, 5.1080729167, na qual o primeiro campo é a marca de tempo (pode-se ver que há duas casas decimais a mais que a marca de tempo do GPS, dado que a precisão da sua marca de tempo é 100 vezes maior), o segundo campo é a frequência de rotação da roda traseira esquerda, o terceiro valor é a frequência de rotação da roda traseira direita e o último valor é a velocidade de translação do veículo.

3.2 Coleta dos Dados

Foram coletados dados reais de posicionamento e das velocidades de rotação das rodas de um veículo. Para realizar isto, a unidade de bordo mencionada na seção anterior foi conectada ao veículo através de seu barramento CAN e programada para coletar as leituras dos sensores ligados ao CAN e as leituras do receptor GPS da própria OBU. Com esta configuração, o veículo percorreu um trajeto no campus da Ilha do Fundão da Universidade Federal do Rio de Janeiro. Os dados coletados foram extraídos da OBU para um *notebook* através de uma conexão cabeada *Ethernet*.

No momento da coleta dos dados, o tempo atmosférico era ensolarado, com céu sem nuvens. A região em que os dados foram coletadas não tinha a presença de obstáculos causadores de perdas de sinal. Dado que as condições atmosféricas têm efeitos na acurácia do receptor GPS, estas condições são as mais adequadas para

coletar dados para o treino dos modelos. A tabela abaixo lista características gerais do conjunto de dados coletado.

Tabela 3.2: Especificações gerais do conjunto de dados coletado.

Número de Leituras do GPS	2107
Número de Leituras do CAN	13614
Tamanho dos Dados	796,8 <i>kilobytes</i>

3.3 Análise do Conjunto de Dados

Esta seção descreve a análise realizada sobre os dados coletados conforme descrito na seção anterior. São considerados os comportamentos dos dados com o tempo e as métricas de autocorrelação para escolher possíveis candidatos de estimadores adequados. Toda a análise feita foi programada na linguagem Python e usou as bibliotecas Statsmodels [12] e SciPy [15] para os cálculos das métricas de correlação e a biblioteca PyPlot [16] para a geração dos gráficos.

3.3.1 Posição Geográfica

A primeira análise será feita com as posições obtidas. O comportamento de cada coordenada será verificado separadamente.

Comportamento no Tempo

Para iniciar a análise do conjunto de dados, os dados serão apresentados em diversos gráficos, nos quais se pode observar o comportamento das grandezas de interesse. Um gráfico que exhibe o perfil do trajeto pode ser vista na Figura 3.2 abaixo. Além disso, as Figuras 3.3 e 3.4 exibem o comportamento dos dois ângulos determinantes da posição, a latitude e a longitude.

Evidentemente, o comportamento destas variáveis depende do local em que o objeto de interesse está no momento. Neste conjunto de dados, como se pode ver pelos gráficos, isto faz com que a média dependa do instante, gerando um comportamento acentuado de *média móvel*. Isto significa que os sinais da latitude e da longitude não são estacionários. A análise dos dados com as ferramentas de séries temporais, portanto, se torna pouco acurada, pois afetará as métricas de correlação, que assumem a estacionariedade. Por este motivo, foi extraída a primeira diferença das coordenadas, conforme visto no Capítulo 2.

Outro argumento para fazer isto é o fato de que treinar um modelo de Aprendizado de Máquina com o valor das coordenadas associa as estimativas das posições ao intervalo assumido por elas no conjunto de dados. Caso o modelo precisasse

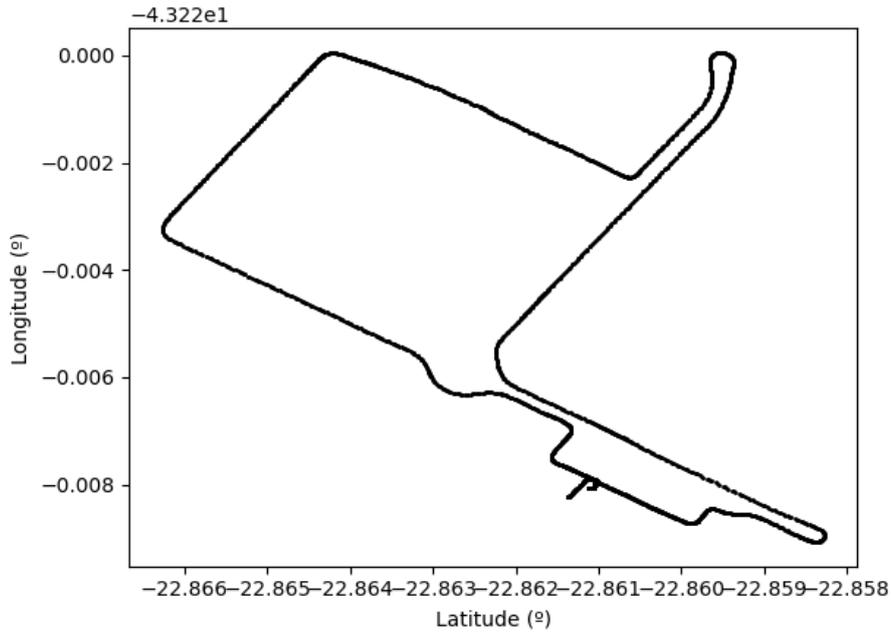


Figura 3.2: Trajeto realizado na coleta de dados pela Ilha do Fundão.

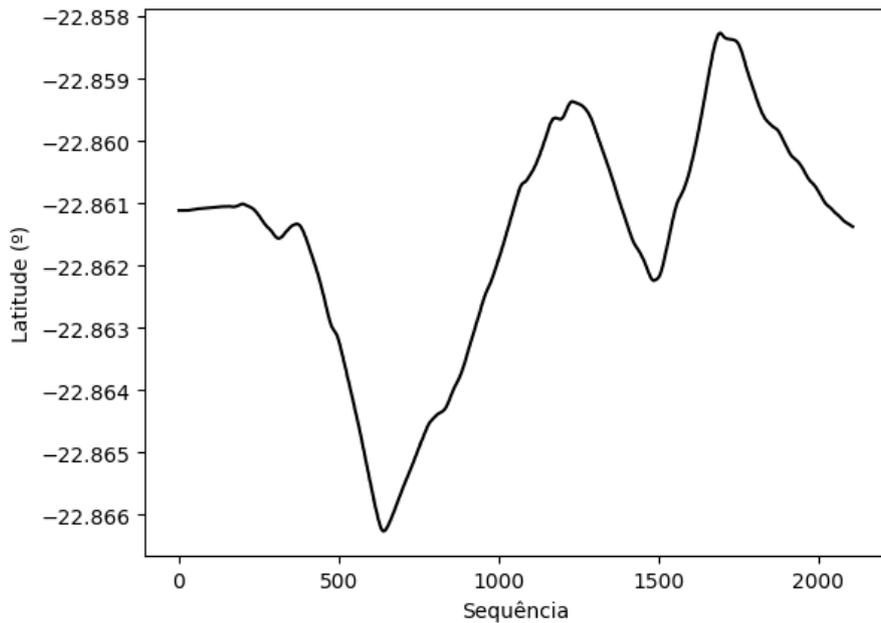


Figura 3.3: Comportamento da latitude em relação ao tempo.

ser utilizado em outro local, fora do campus da Ilha do Fundão, seria necessário retreiná-lo com novos dados. Utilizar as diferenças torna o modelo independente da localidade. A primeira diferença das coordenadas fornece uma noção da *velocidade* da série.

Os comportamentos da primeira diferença de latitude e primeira diferença de longitude são exibidos respectivamente nas Figuras 3.5 e 3.6, nas quais as médias das séries são indicadas com uma linha horizontal vermelha. Uma inspeção visual

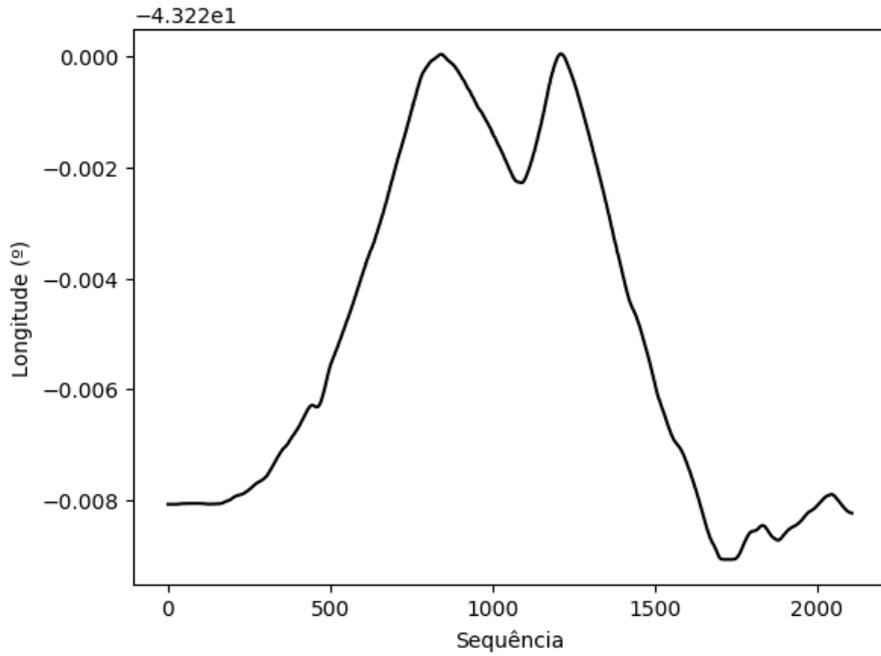


Figura 3.4: Comportamento da longitude em relação ao tempo.

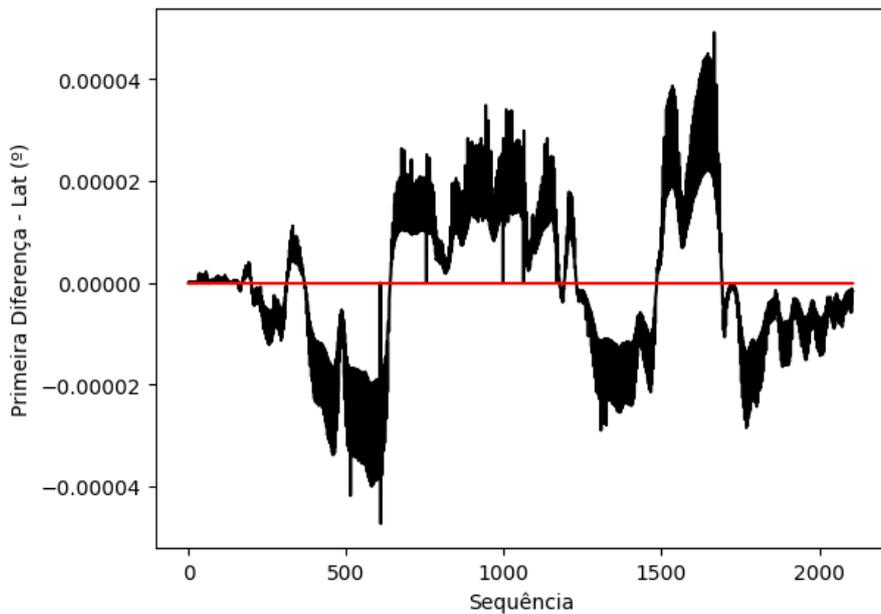


Figura 3.5: Comportamento da primeira diferença da latitude em relação ao tempo.

nos gráficos indica que o comportamento de média móvel foi suavizado. No entanto, ainda está presente. Este é um comportamento esperado, dado que as variações na posição dependem das velocidades linear e angular do veículo, que dependem do comportamento do motorista e das características da pista. As variações também dependem dos fatores que afetam a precisão do receptor GPS, discutidos na Introdução deste trabalho, que podem gerar ruídos nas leituras extraídas.

A série resultante da primeira diferenciação apresenta, portanto, diversas

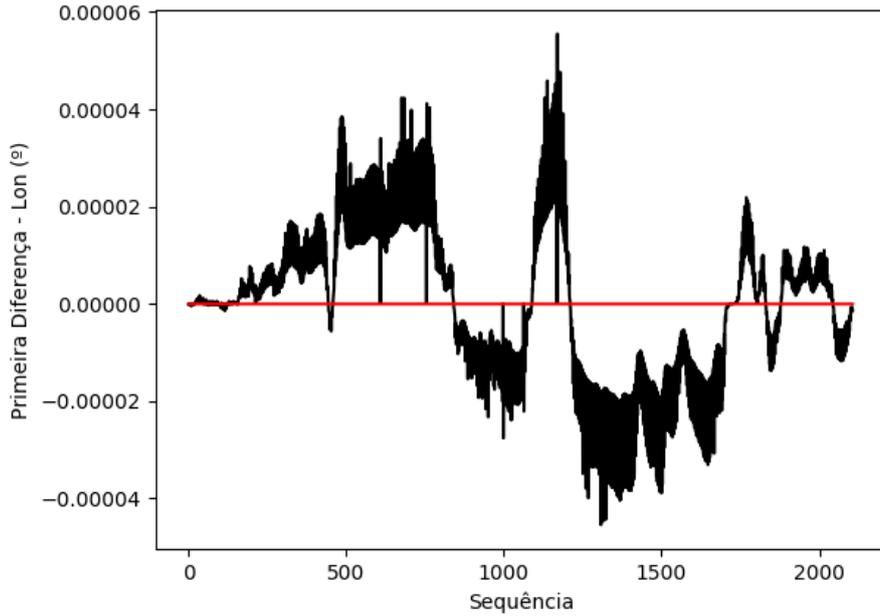


Figura 3.6: Comportamento da primeira diferença da longitude em relação ao tempo.

tendências: há comportamentos de média móvel com diferentes intensidades e a amplitude das oscilações é variável durante todo o trajeto. A amplitude aparentemente depende da velocidade do veículo, dado que, ao início da coleta dos dados, o veículo saía de um estacionamento e ao fim da mesma o veículo retornava ao estacionamento, desenvolvendo baixas velocidades nos dois momentos. Em uma tentativa de suprimir estas tendências para aproximar o comportamento da série de um comportamento estacionário, remover dependências dos estados anteriores e suprimir fatores de imprecisão do GPS que apresentem comportamentos periódicos com frequências menores que as do receptor, foi gerada uma terceira série temporal por meio do cálculo das segundas diferenças. Este valor fornece uma noção da *aceleração* da série. O comportamento da segunda diferença é exibido nas Figuras 3.7 e 3.8, nas quais a linha vermelha novamente indica a média da série.

Após a transformação, uma inspeção visual indica que o comportamento de *média móvel* foi suprimido. Os valores tendem a oscilar em torno da média, cujo valor é aproximadamente 0. No entanto, ainda existe um comportamento dependente do tempo que afeta a amplitude destas oscilações. Este efeito ocorre por causa da variação na aceleração do veículo (também chamada de arrancada ou *jerk*), uma medida que talvez possa ser capturada a partir dos sensores. Novas diferenciações não afetam o comportamento da série temporal, de modo que a maior ordem de diferenciação que pode ser considerada a partir destes dados é 2. Dado que os valores da série assumem valores com uma ordem de grandeza de 10^{-5} , transformações das séries temporais das coordenadas através do cálculo do logaritmo e da raiz quadrada também não provocaram qualquer alteração neste comportamento.

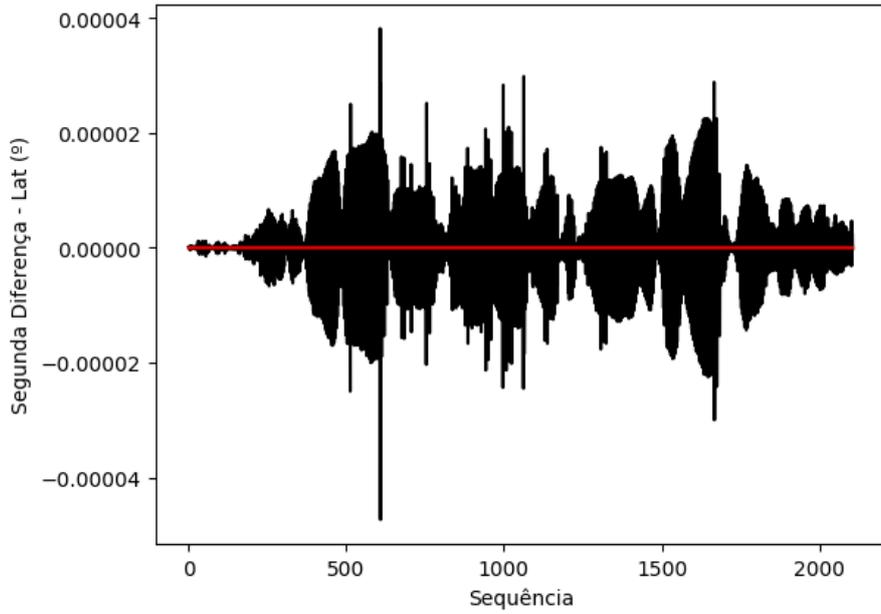


Figura 3.7: Comportamento da segunda diferença da latitude em relação ao tempo.

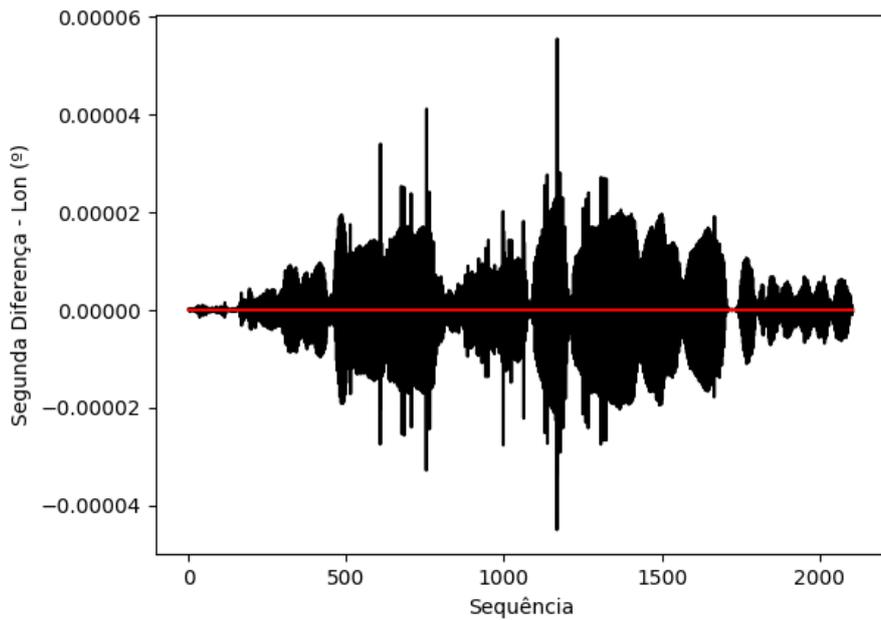


Figura 3.8: Comportamento da segunda diferença da longitude em relação ao tempo.

A seguir, são apresentados os comportamentos das métricas de autocorrelação para a primeira e a segunda diferenças, para determinar a dependência dos seus valores em relação ao passado.

Medidas de Autocorrelação

Primeiramente, será considerada a primeira diferença das coordenadas. As Figuras 3.9 e 3.10 apresentam os autocorrelogramas e autocorrelogramas parciais para

as primeiras diferenças da latitude e da longitude. Os gráficos incluem o valor para o atraso 0 (que é 1). Neste contexto, atraso é o espaçamento em tempo entre duas leituras da mesma variável. Desta maneira, as medidas de autocorrelação podem fornecer candidatos a termos autorregressivos e/ou de média móvel.

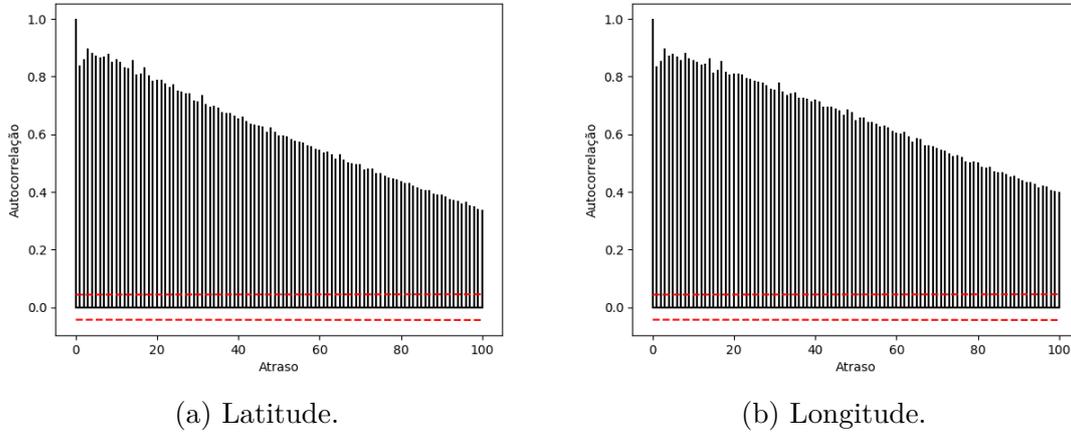


Figura 3.9: Autocorrelações das primeiras diferenças das coordenadas.

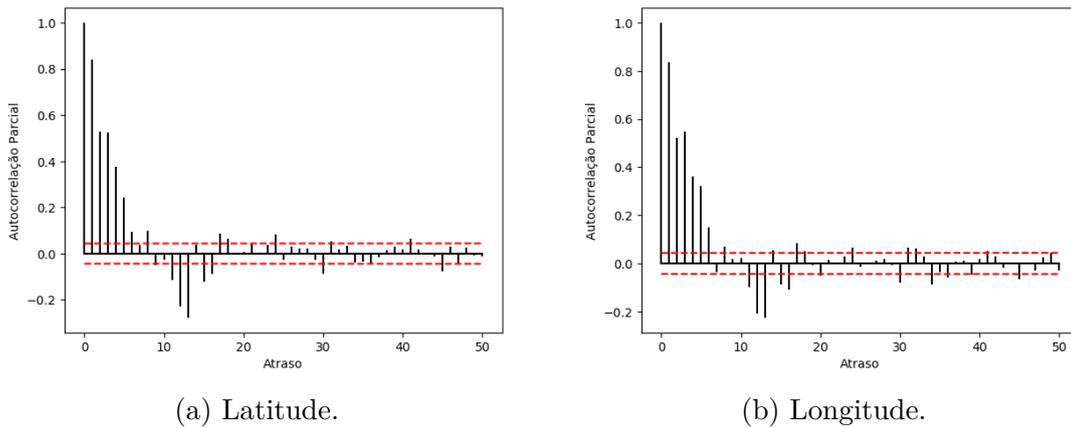


Figura 3.10: Autocorrelações parciais das primeiras diferenças das coordenadas.

Os comportamentos das métricas para as primeiras diferenças das duas coordenadas são, visualmente, muito próximos. Os valores da autocorrelação, por apresentarem uma tendência decrescente, sem quedas abruptas de valor, indicam que um bom estimador para as primeiras diferenças não é um modelo MA, conforme as discussões anteriores. O valor da autocorrelação parcial também não apresenta quedas abruptas de valor. O comportamento indicado aqui indica que o modelo se trata de um ARMA, ou seja, as primeiras diferenças das coordenadas apresentam dependências de termos anteriores e de termos de ruído. No entanto, as diversas tendências presentes na série violam a condição de estacionariedade que os modelos de séries temporais utilizam, e isto pode ser a causa do comportamento apresentado nos gráficos.

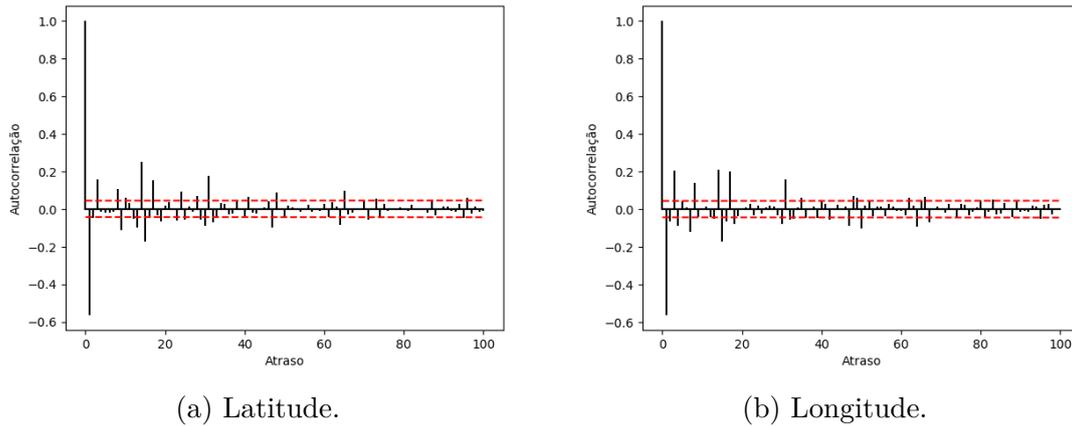


Figura 3.11: Autocorrelações das segundas diferenças das coordenadas.

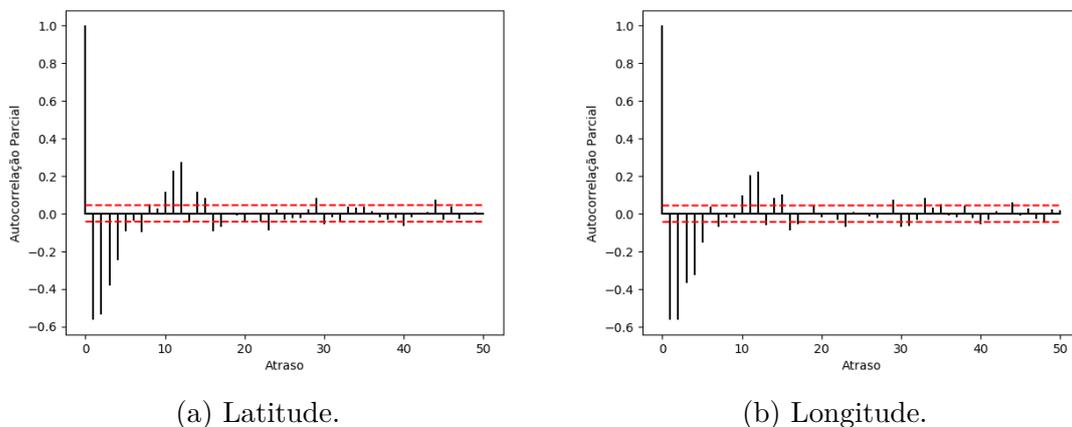


Figura 3.12: Autocorrelações parciais das segundas diferenças das coordenadas.

As Figuras 3.11 e 3.12 apresentam os autocorrelogramas e autocorrelogramas parciais para as segundas diferenças da latitude e da longitude. O comportamento exibido nestes últimos gráficos é diferente quanto à autocorrelação: o gráfico exibe alguns picos de significância afastados entre si. A construção teórica de um modelo MA indica que os picos de autocorrelação são adjacentes, logo, picos espaçados são comportamentos errôneos e, portanto, a aplicação de um termo MA com atraso 1 poderia ser interessante.

Aplicar uma segunda diferenciação parece ter removido os comportamentos de média móvel exibidos no gráfico da primeira diferença. No entanto, o comportamento do motorista, não periódico e imprevisível, gera uma tendência remanescente: um comportamento de amplitude variável. Esta tendência tem efeitos nos valores da autocorrelação, e isto se reflete nestes resultados.

Dado que as séries temporais da primeira e da segunda diferenças das coordenadas apresentam tendências que não puderam ser tratadas, tendo a estacionariedade comprometida, pode-se concluir que modelos de séries temporais não-exógenos não formarão estimadores adequados. Há uma clara dependência do estado do veículo,

e as estimativas das posições devem utilizar medidas que o representem. Portanto, não se pode precisar a quantidade de termos autorregressivos, e o uso de variáveis auxiliares – dados dos sensores – faz-se necessário.

Por isto, os três tipos de modelos que podem ser utilizados aqui são os modelos ARIX, NARIX, estático e regressão dinâmica. Neste caso, para evitar perdas de informações importantes para as estimativas, um fenômeno comum no uso de altas ordens de diferenciações [10], optou-se por utilizar apenas a primeira diferença das coordenadas.

Pode-se utilizar os dados dos sensores como informações auxiliares na determinação do comportamento do motorista para particionar o conjunto de dados e aplicar cada parte em um modelo independente, a fim de suavizar o efeito das tendências remanescentes. Além disto, como mencionado no parágrafo anterior, as informações dadas por eles são necessárias para representar o estado do veículo. Para verificar a capacidade destes em realizar estas duas tarefas, a seguir as séries temporais dos três sensores são analisadas com as métricas já descritas e, por fim, a correlação cruzada entre coordenadas geográficas e valores dos sensores é analisada.

3.3.2 Sensores

Como discutido anteriormente, são três os valores fornecidos pelos sensores:

- Frequência de rotação (em Hz) da roda traseira esquerda do carro;
- Frequência de rotação (em Hz) da roda traseira direita do carro;
- Velocidade de translação do carro, em m/s.

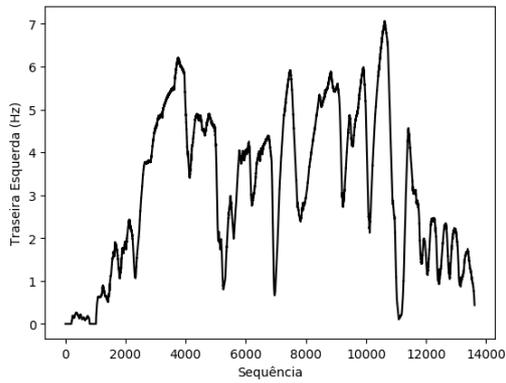
Os três valores variam com o tempo durante o percurso. Logo, serão analisados como séries temporais.

Comportamento no Tempo

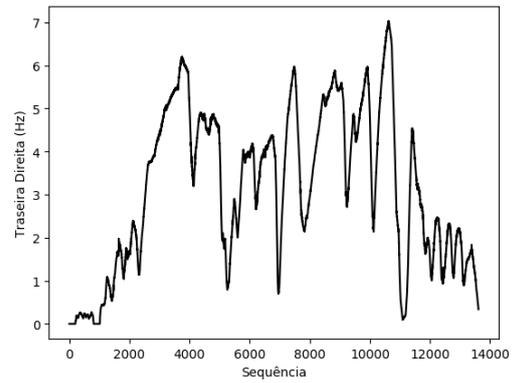
Uma primeira análise é verificar o comportamento dos valores assumidos pelas séries com o tempo. As Figuras 3.13 e 3.14 são gráficos que apresentam estes comportamentos.

Uma inspeção visual permite concluir que os três comportamentos são muito parecidos. Existem comportamentos evidentes de média móvel e amplitude variável nas séries.

Duas informações interessantes para se obter a partir destes sensores são os momentos em que o veículo está fazendo uma curva, andando em linha reta e parado. Determinar se o veículo está parado ou não é determinável apenas pela velocidade, mas, para as outras informações, as duas frequências de rotação são necessárias.



(a) Roda esquerda.



(b) Roda direita.

Figura 3.13: Comportamentos das frequências de rotação das rodas com o tempo.

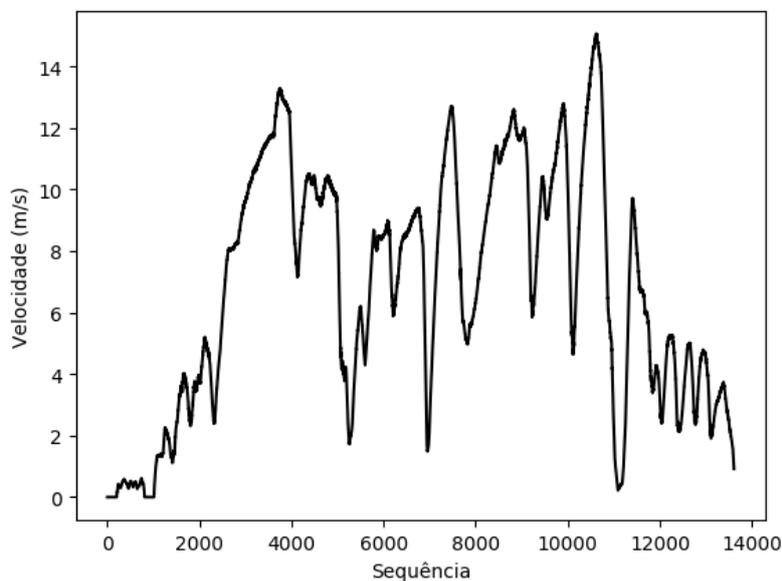


Figura 3.14: Comportamento da velocidade de translação do carro com o tempo.

Para analisar estes estados, foi gerado um gráfico que representa as diferenças entre as duas frequências de rotação em um instante contra as velocidades de translação no mesmo instante. Esta é a Figura 3.15. Valores negativos da diferença entre as frequências indicam um valor maior da frequência da roda direita, o que indica uma curva para a esquerda, e valores positivos da diferença indicam um valor maior da frequência da roda esquerda, o que indica uma curva para a direita.

Neste gráfico, é possível notar que estão presentes diversos regimes de movimento. Quando o veículo está parado, a velocidade informada deve ser, teoricamente, nula. Caso o sensor apresente ruídos, o valor informado deve ser próximo de 0, possivelmente não-nulo. É possível notar que, mesmo com a velocidade assumindo estes valores, há pequenas diferenças entre as rodas visíveis no gráfico. Isto indica a existência de ruídos nos sensores. Há curvas para a esquerda e para a direita, indi-

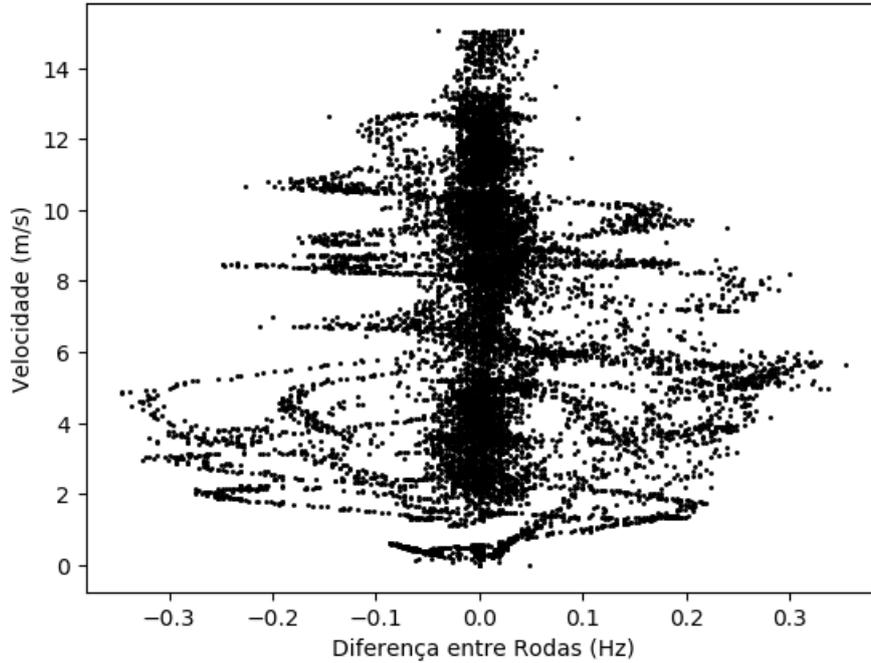
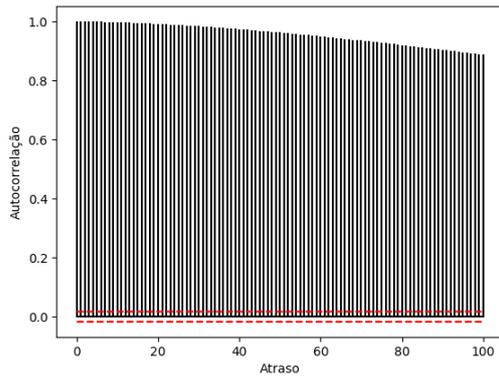


Figura 3.15: Diferenças entre frequências de rotação contra velocidades.

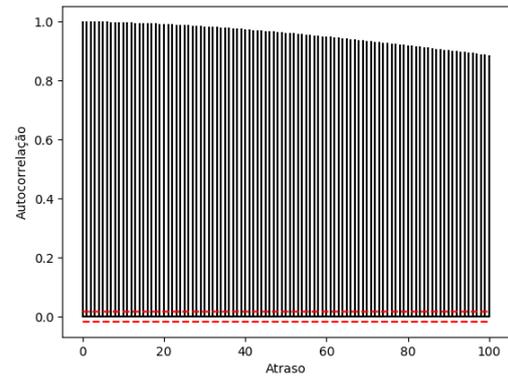
cadadas pelas metades esquerda e direita da imagem. Os valores para a diferença entre as frequências de rotação das rodas tendem a diminuir com o aumento da velocidade, o que reflete o comportamento feito na coleta dos dados: curvas mais fechadas foram feitas a menores velocidades. Mesmo nas maiores velocidades alcançadas, há pontos em que a diferença entre as frequências não é nula; isto pode ser um indicativo do efeito do ruído ou da mudança de faixa do carro dentro de uma via. Estes comportamentos reforçam a ideia de que se pode separar os dados em diferentes partes com diferentes comportamentos. Como não se sabe quais são os comportamentos exatos dos ruídos dos sensores, uma alternativa apropriada para separar os diferentes comportamentos assumidos pelo veículo é utilizar um método de análise de agrupamentos como o K-Médias, descrito no capítulo anterior, que aprende os padrões existentes nos dados. Os comportamentos das três leituras dos sensores indicam um comportamento de média móvel. Isto depende, mais uma vez, do comportamento do motorista e das características da via. Como este comportamento de média móvel se manifesta também no comportamento das posições geográficas, não serão aplicadas diferenciações aos valores dos sensores.

Métricas de Autocorrelação

Por fim, para verificar se termos de diferentes instantes das leituras estão correlacionados, devem ser analisadas a autocorrelação e a autocorrelação parcial para os três sensores. As Figuras 3.16 a 3.19 a seguir apresentam os autocorrelogramas e autocorrelogramas parciais para as três leituras.



(a) Roda esquerda.



(b) Roda direita.

Figura 3.16: Autocorrelações das frequências de rotação das rodas.

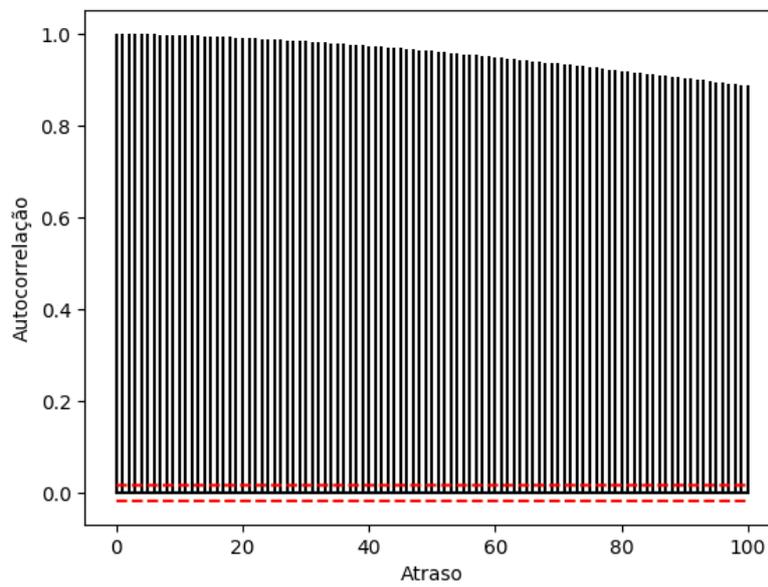
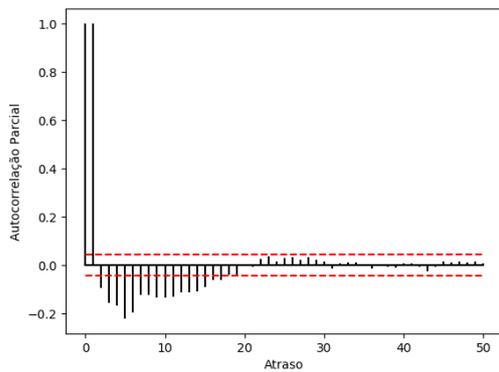
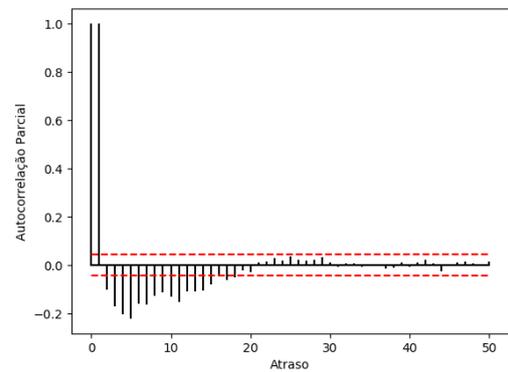


Figura 3.17: Autocorrelação da velocidade de translação do veículo.



(a) Roda esquerda.



(b) Roda direita.

Figura 3.18: Autocorrelações parciais das frequências de rotação das rodas.

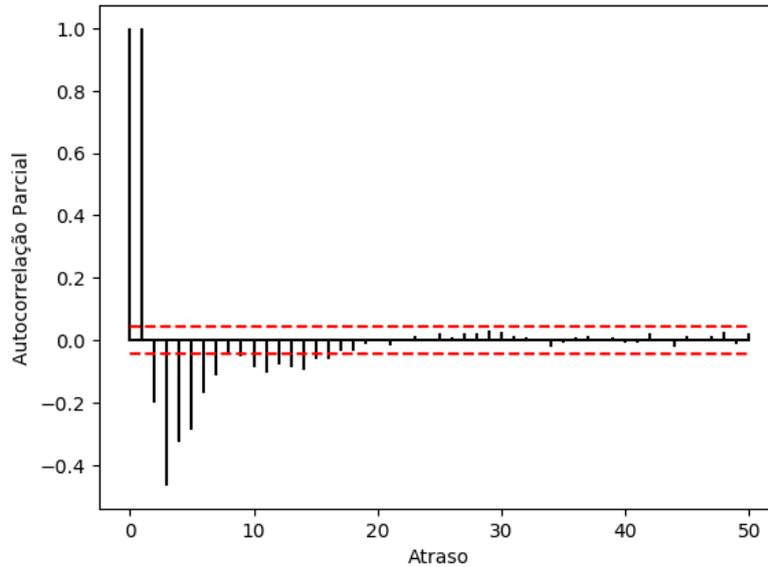


Figura 3.19: Autocorrelação parcial da velocidade de translação do veículo.

O que se pode observar destas imagens é que a autocorrelação em todos os casos apresenta um comportamento decrescente sem quedas abruptas para valores abaixo do limiar de significância. Isto indica que as séries temporais dos sensores apresentam um comportamento autorregressivo, ou seja, indica que os valores apresentados pelos sensores dependem dos valores que eles exibiram anteriormente. Logo, para a construção do estimador, deve-se escolher através da correlação cruzada o termo atrasado dos sensores com o maior valor para a correlação cruzada, de modo a evitar incluir dependências entre as variáveis de entrada nos algoritmos de Aprendizado de Máquina.

3.3.3 Análise da Correlação Cruzada

A medida de correlação cruzada foi extraída entre as duas coordenadas e as três leituras fornecidas pelos sensores. As Figuras 3.20 a 3.22 apresentam as correlações cruzadas para a primeira diferença da latitude, e as Figuras 3.23 a 3.25 mostram as correlações cruzadas para a primeira diferença da longitude, indicando as dependências entre as variações das coordenadas e os valores dos sensores.

Os gráficos indicam um comportamento periódico da correlação cruzada: há dois picos seguidos por dois valores mais baixos. Todos os valores de atraso no intervalo de 0 a 30 apresentam valores significativos. Isto indica que o melhor atraso a se utilizar no estimador é o atraso 0 (ou seja, o valor mais recente dos sensores), dado que este é o valor mais recente e com o maior valor para a correlação cruzada. Dado o comportamento autorregressivo das medidas dos sensores, apenas 1 valor dos sensores deve ser utilizado.

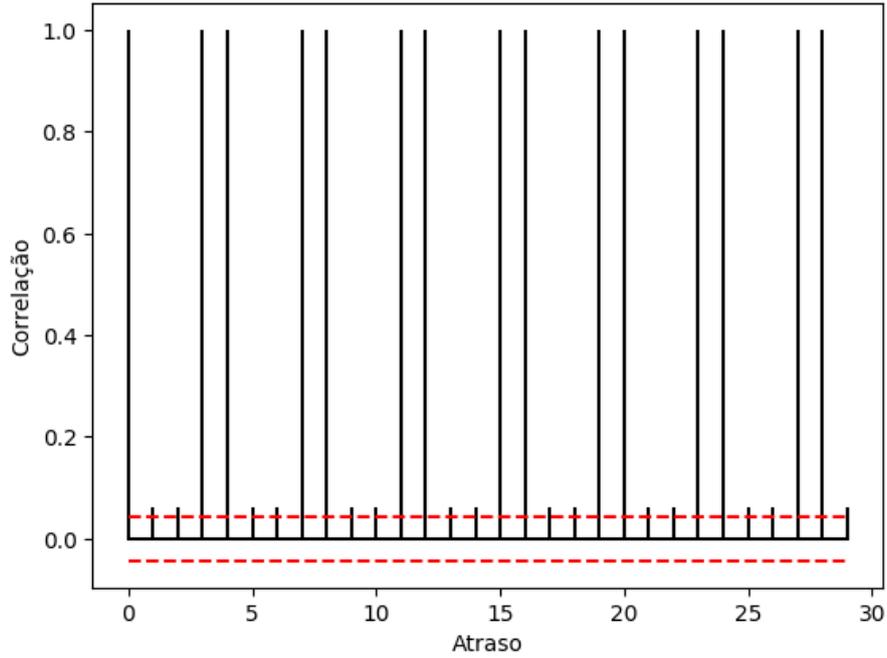


Figura 3.20: Correlação Cruzada entre a primeira diferença da latitude e a frequência de rotação da roda esquerda.

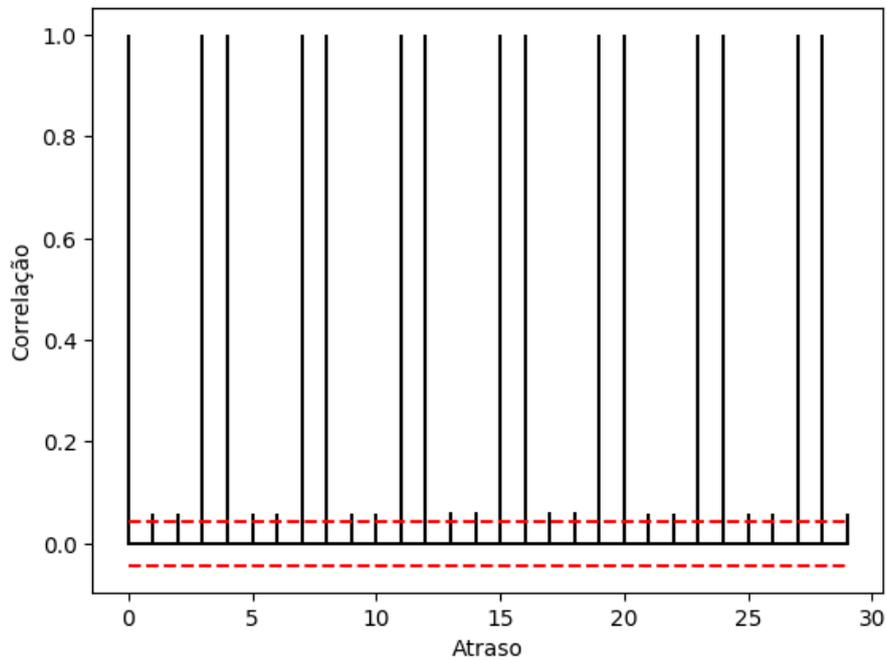


Figura 3.21: Correlação Cruzada entre a primeira diferença da latitude e a frequência de rotação da roda direita.

A análise realizada aqui resulta em algumas conclusões:

- Os modelos de séries temporais não-exógenos não são estimadores adequados para as posições ou para as suas diferenças: existe um comportamento de amplitude variável dependente do comportamento do motorista e das caracte-

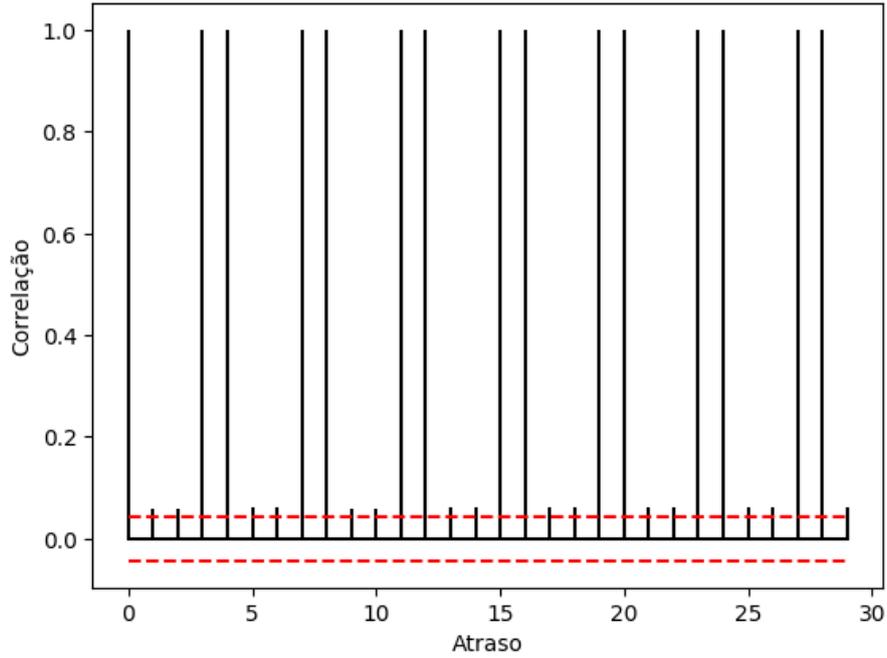


Figura 3.22: Correlação Cruzada entre a primeira diferença da latitude e a frequência de rotação da roda esquerda.

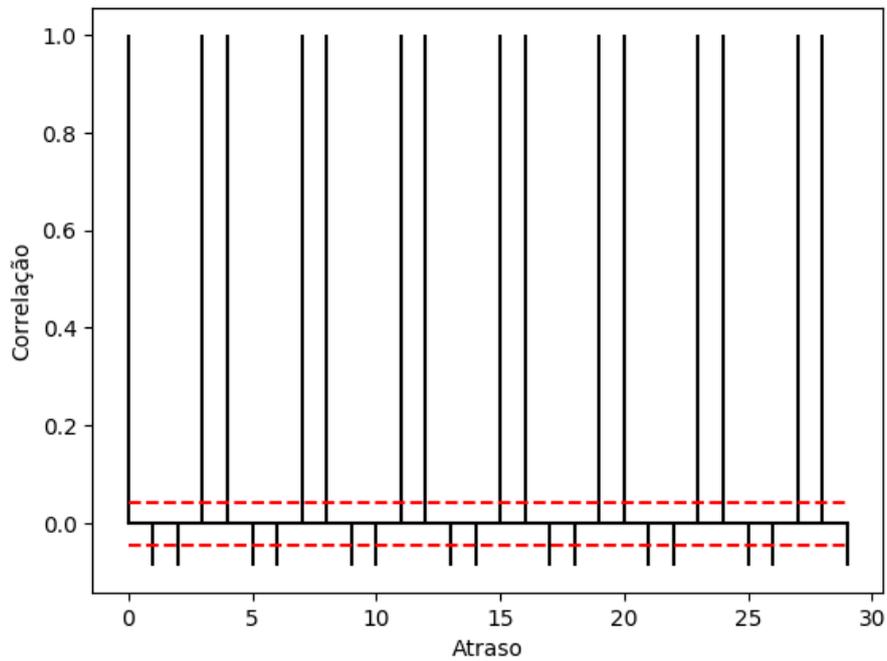


Figura 3.23: Correlação Cruzada entre a primeira diferença da longitude e a frequência de rotação da roda esquerda.

terísticas da pista que não pôde ser eliminado pelas técnicas de tratamento de tendências, violando o princípio de estacionariedade das séries temporais, que garante a validade destes modelos;

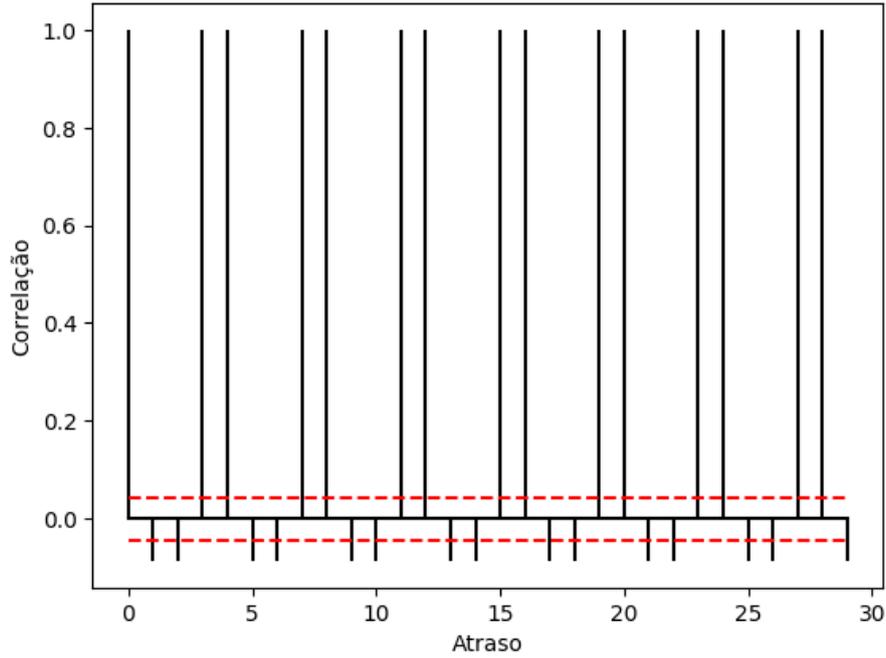


Figura 3.24: Correlação Cruzada entre a primeira diferença da longitude e a frequência de rotação da roda direita.

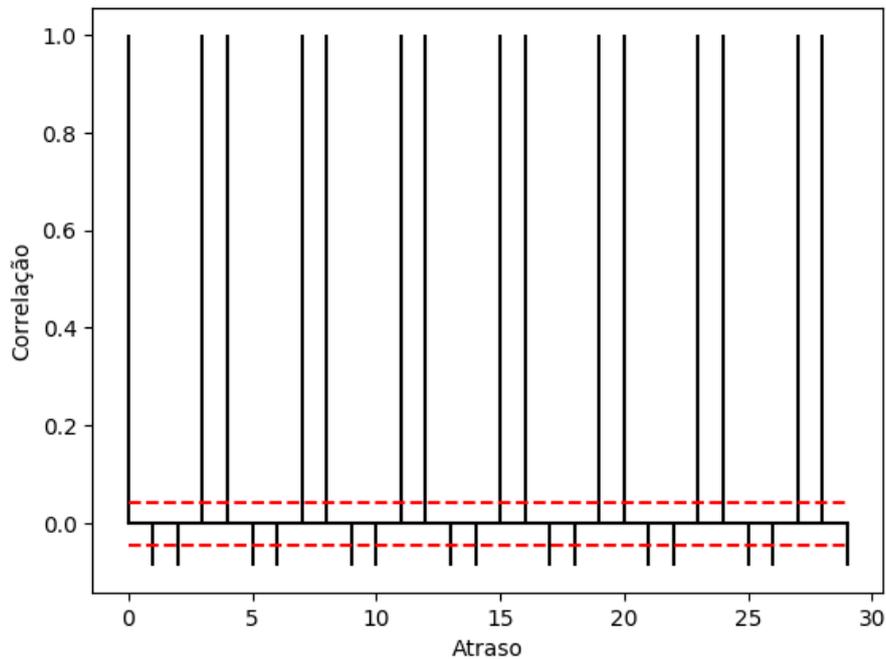


Figura 3.25: Correlação Cruzada entre a primeira diferença da longitude e a frequência de rotação da roda esquerda.

- As leituras dos sensores exibem um comportamento autorregressivo. Devido a isto, inserir mais de um valor atrasado adiciona dependências entre variáveis de entrada de um modelo de Aprendizado de Máquina, o que prejudica as estimativas;

- Deve-se utilizar o valor mais recente das leituras dos sensores no estimador das posições.
- Devido às conclusões anteriores, possíveis candidatos de modelos são: ARIX, NARIX e Modelo Estático.

Os modelos descritos no Capítulo 2 necessitam de um procedimento para determinar seus parâmetros. Por isto, o Capítulo 4 apresenta uma descrição das técnicas de Aprendizado de Máquina utilizadas neste projeto para estimá-los: algoritmos de regressão e técnicas de pré-processamento que os auxiliam.

Capítulo 4

Técnicas de Aprendizado de Máquina

Este capítulo se destina a descrever técnicas da área de Aprendizado de Máquina que foram utilizadas para implementar os modelos que foram selecionados na análise dos dados. Inicia-se pela descrição dos modelos de Aprendizado de Máquina considerados para o cálculo das estimativas. A seguir, discute-se técnicas de processamento utilizadas para auxiliá-los. Por fim, discutem-se as técnicas para dividir um conjunto de dados entre dados de treinamento e dados de teste dos algoritmos de interesse.

4.1 Algoritmos de Regressão

Para representar o comportamento dos modelos descritos anteriormente, no Capítulo 2, empregam-se dois tipos de algoritmos de Aprendizado de Máquina para que se estime o comportamento das posições em relação aos seus valores anteriores e às leituras dos sensores: regressão linear e redes neurais artificiais. A regressão linear foi escolhida por ser um preditor de baixo custo computacional de relações lineares utilizado frequentemente, e a escolha pela rede neural artificial foi feita porque, como será visto mais adiante, ela é considerada um aproximador universal de funções (incluindo funções não-lineares).

4.1.1 Regressão Linear

Esta subseção descreve o algoritmo de regressão linear, um preditor de relações lineares de baixo custo computacional, de acordo com o exposto por Hastie [3].

Descrição

Como mencionado no capítulo anterior, um modelo de regressão linear expressa uma relação linear entre uma variável e um conjunto de outras variáveis. Dado um vetor

de entradas $\mathbf{x}^T = [x_1, x_2, \dots, x_p]$, estima-se o valor da variável de saída y como:

$$\hat{y} = a_0 + \sum_{i=1}^p a_i x_i, \quad (4.1)$$

onde os coeficientes a_i correspondem aos pesos das variáveis x_i , e a_0 é um valor arbitrário, chamado de coeficiente linear ou *bias*. Para apenas uma variável de entrada e uma variável de saída, o procedimento realizado é um ajuste de reta, exibido na Figura 4.1, na qual os dados experimentais são representados em vermelho e a relação estimada pelo algoritmo é a reta em azul.

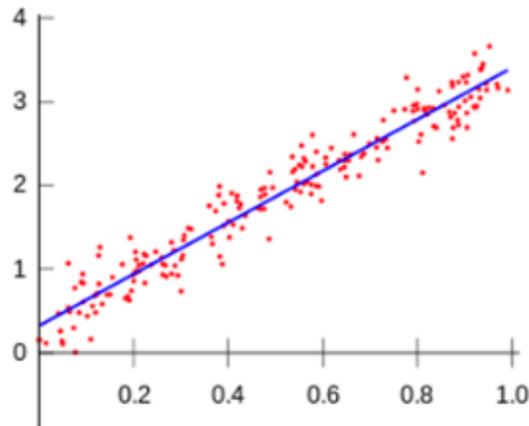


Figura 4.1: Processo realizado por uma Regressão Linear unidimensional.

Uma outra maneira de representar o modelo é adicionar o valor constante 1 ao início do vetor \mathbf{x} , juntar a_0 e os outros coeficientes a_i ordenadamente no vetor \mathbf{a} e escrever a equação em um formato vetorial, como um produto interno:

$$\hat{y} = \mathbf{x}^T \mathbf{a} \quad (4.2)$$

onde \mathbf{x}^T é uma operação de transposição vetorial.

No caso de um modelo que preveja múltiplas saídas, como o sistema proposto, a equação pode ser expandida para um vetor de saídas y . Neste caso, o vetor \mathbf{a} se torna uma matriz de coeficientes \mathbf{A} , na qual cada coluna corresponde a uma variável de saída:

$$\hat{\mathbf{y}} = \mathbf{x}^T \mathbf{A}. \quad (4.3)$$

Ajuste dos Coeficientes

Existem vários métodos para ajustar os coeficientes a_i . O mais utilizado é o método de mínimos quadrados, que visa minimizar o erro quadrático (EQ) trabalhando com a sua derivada. Considerando a existência de uma variável de saída y , que haja n pares entrada-saída (y_i, x_i) , correspondentes aos dados disponíveis para o

treinamento do modelo, e que as estimativas calculadas para as saídas y_i a partir das entradas x_i sejam \hat{y}_i , o EQ pode ser escrito como:

$$EQ(\mathbf{a}) = \sum_{i=1}^n (y_i - \hat{y}_i)^2. \quad (4.4)$$

Usando a Equação 4.2, obtemos a seguinte expressão:

$$EQ(\mathbf{a}) = \sum_{i=1}^n (y_i - x_i^T \mathbf{a})^2. \quad (4.5)$$

O EQ é uma função quadrática dos parâmetros \mathbf{a} , e, portanto, sempre existe um valor mínimo para ele, mas é possível que ele não seja único. Reescrevendo o EQ em uma notação matricial, obtemos:

$$EQ(\mathbf{a}) = (\mathbf{y} - \mathbf{X}\mathbf{a})^T (\mathbf{y} - \mathbf{X}\mathbf{a}), \quad (4.6)$$

onde \mathbf{X} é uma matriz $n \times p$, sendo cada linha um vetor de entrada, de um par entrada-saída, e \mathbf{y} um vetor de dimensão n , correspondendo a todas as saídas do conjunto de treinamento. Derivando o EQ em relação a \mathbf{a} e igualando esta derivada a 0 (a condição que é válida para um ponto de extremo), obtemos as equações normais:

$$\mathbf{X}^T (\mathbf{y} - \mathbf{X}\mathbf{a}) = 0. \quad (4.7)$$

Se $\mathbf{X}^T \mathbf{X}$ não for uma matriz singular (isto é, se ela possuir uma matriz inversa), então a solução única é dada por:

$$\hat{\mathbf{a}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \quad (4.8)$$

As estimativas para as saídas podem ser, então, calculadas como:

$$\hat{y}_i = \hat{y}(x_i) = (x_i)^T \hat{\mathbf{a}}. \quad (4.9)$$

O modelo de Regressão Linear será utilizado neste projeto como estimador dos modelos lineares.

4.1.2 Rede Neural Artificial

Esta subseção faz uma descrição do modelo de uma Rede Neural Artificial de acordo com o exposto por Haykin [17] e Silva [18].

Definição

As Redes Neurais Artificiais (RNAs) surgiram como modelos matemáticos que buscavam emular o processamento neural humano. A unidade básica das mesmas é o neurônio artificial que, a partir de um conjunto de valores de entrada, realiza uma computação sobre a soma ponderada dos mesmos.

O tipo de RNA mais frequentemente utilizado é o Perceptron de Múltiplas Camadas (ou MLP - *Multi-Layer Perceptron*). Esta é uma arquitetura que utiliza múltiplos neurônios agrupados em camadas com diferentes funções. Os neurônios de duas camadas são interligados, mas os neurônios de uma mesma camada não possuem ligações. Com isto, os valores resultantes das computações dos neurônios de uma camada são o conjunto dos valores de entrada da próxima camada. A Figura 4.2 apresenta um diagrama da arquitetura MLP, que gera um vetor de saídas y a partir do vetor de entradas x .

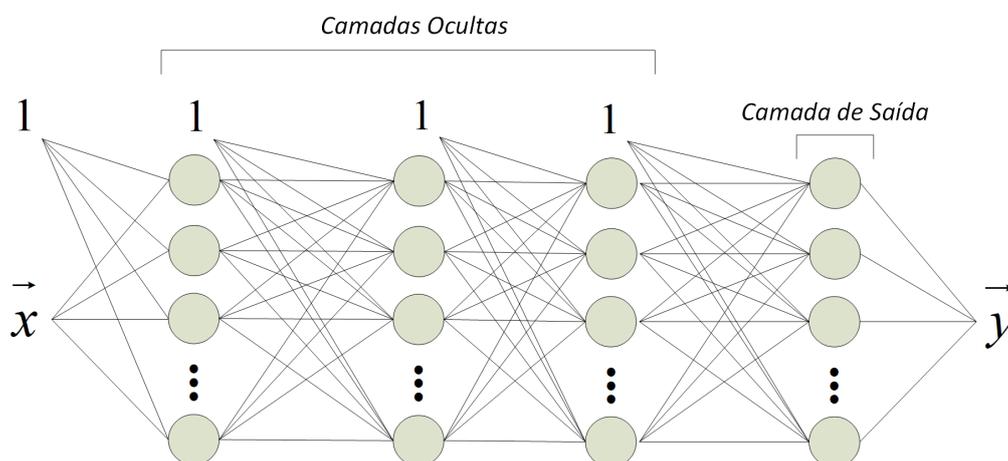


Figura 4.2: Diagrama da arquitetura de um Perceptron de Múltiplas Camadas.

Há dois tipos de camadas nesta arquitetura:

- **Camadas Ocultas:** cada neurônio da camada oculta tem como entradas os valores das variáveis de entrada da rede ou as saídas de todos os neurônios de uma camada oculta anterior. Os resultados dos processamentos realizados nos neurônios de uma camada oculta podem ser fornecidos para uma outra camada oculta ou para uma camada de saída;
- **Camada de Saída:** cada neurônio desta camada corresponde a uma variável de saída. Os neurônios desta camada recebem como entradas os resultados de todos os neurônios da última camada oculta. Os resultados das computações destes neurônios correspondem às variáveis de saída do modelo.

A computação realizada por um neurônio pode ser descrita da seguinte forma: realiza-se uma soma com pesos de todas as suas entradas. Esta soma é aplicada

a uma função de ativação, que fornece a saída do neurônio. Este processamento é exibido na equação abaixo:

$$v_i = f(w_0 + \sum_{i=1}^n w_n x_n) = f(u_i), \quad (4.10)$$

na qual v_i é o resultado da computação do neurônio, f é a função de ativação atribuída ao neurônio, w_0 é um valor de ajuste, também chamado de *viés* ou *bias*, w_i é o peso da entrada x_i e u_i é uma abreviação para a soma ponderada das entradas e do viés. A Figura 4.3 apresenta um diagrama do neurônio artificial.

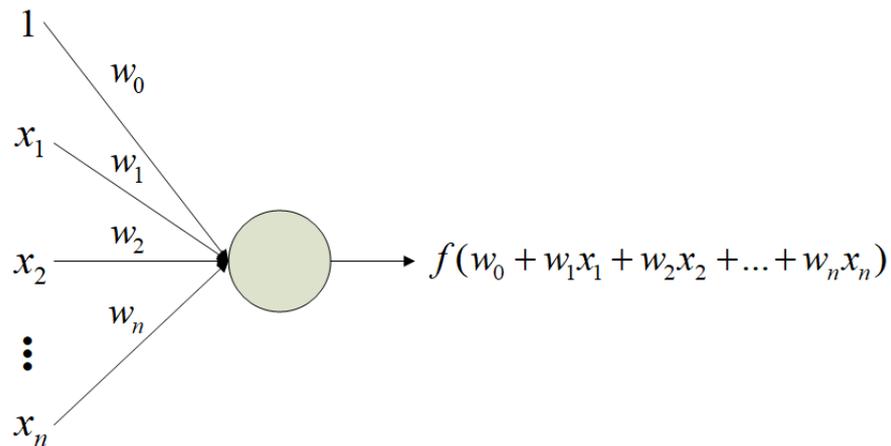


Figura 4.3: Diagrama de um neurônio, unidade básica de processamento de uma RNA.

Uma função de ativação é qualquer função que, a partir de uma entrada numérica, forneça uma saída também numérica. As funções de ativação mais frequentemente utilizadas para os neurônios são:

- Função linear ou identidade ($f(x) = x$): não modifica o valor da entrada; a saída do neurônio será, portanto, a soma ponderada das entradas;
- Função tangente hiperbólica:

$$f(x) = \tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}; f(x) \in [-1, +1]; \quad (4.11)$$

- Função sigmoide:

$$f(x) = \frac{1}{1 + e^{-x}}; f(x) \in [0, +1]; \quad (4.12)$$

- Função retificadora ($f(x) = \max(0, x)$).

Pode-se atribuir funções de ativação diferentes para cada neurônio ou para cada camada, e o número de camadas ocultas pode ser ajustado. Estas funções podem

ser utilizadas para o contexto de classificação ou em problemas de regressão. Como este trabalho objetiva resolver um problema de regressão, as redes descritas aqui são de interesse para este problema.

Dado que já há um modelo de regressão para funções lineares, o interesse na RNA é o de elaborar um aproximador de funções não-lineares. Logo, deseja-se que o conjunto de neurônios realize um processamento não-linear sobre as entradas.

Dimensionamento de uma RNA para Aproximação de Funções

Cybenko [19] e Hornik [20] demonstram, em seus trabalhos, um teorema interessante para o dimensionamento de uma rede neural aproximadora de funções a partir de funções sigmoidais. Sendo uma função sigmoideal uma função σ cujos valores assumidos são, como mencionado anteriormente:

$$\sigma(t) \rightarrow 1, t \rightarrow +\infty; \quad (4.13)$$

$$\sigma(t) \rightarrow 0, t \rightarrow -\infty, \quad (4.14)$$

o teorema é o seguinte:

Teorema da Aproximação Universal: seja I_m o hipercubo m -dimensional $[0, 1]^m$. O espaço de funções contínuas com domínio I_m é $C(I_m)$. Então, dados um valor $\epsilon > 0$, uma função qualquer $f \in C(I_m)$ e uma função sigmoideal σ , existe um número inteiro N finito, constantes reais $a_i, b_i \in \mathbb{R}$ e vetores reais $w_i \in \mathbb{R}^m$, em que $i = 1, 2, 3, \dots, N$ de modo que o somatório finito:

$$F(x) = \sum_{i=0}^N a_i \sigma(w_i^T x + b_i), \quad (4.15)$$

seja um aproximador de $f(x)$ que satisfaça a condição:

$$|F(x) - f(x)| < \epsilon, \forall x \in I_m. \quad (4.16)$$

Ou seja, atribuindo uma quantidade suficiente de funções sigmoidais ao somatório, é possível gerar um aproximador para a função desejada com algum nível de precisão. Para que a aproximação apresente um erro menor, é necessário aumentar a quantidade de funções sigmoidais no somatório. Dado que é possível converter a função tangente hiperbólica em uma função sigmoideal através da transformação:

$$\tanh(x) = 2\sigma(2x) - 1, \quad (4.17)$$

este teorema também vale para a função tangente hiperbólica.

A conclusão que se pode obter através deste teorema é o fato de que, para a aproximação de qualquer função com uma rede neural, apenas é necessário utilizar 1 camada oculta, na qual todos os neurônios de camada oculta apresentem a função sigmoideal ou a função tangente hiperbólica como função de ativação e na qual os neurônios de saída possuam a função de ativação linear. Para aprimorar o nível de precisão, deve-se ajustar a quantidade de neurônios da camada oculta. A seguir, explica-se o procedimento utilizado para ajustar os pesos de uma rede neural.

Ajuste dos Pesos

Como mencionado anteriormente, cada neurônio realiza uma soma ponderada de suas entradas. Este mecanismo permite ajustar as importâncias de cada entrada e o valor do viés. É por meio dos pesos e dos vieses que a rede neural registra o seu “conhecimento” do problema. Para que se ajuste os pesos de maneira apropriada, é necessário um critério de desempenho. Para problemas de regressão, a métrica utilizada é o erro médio quadrático, definido nas equações abaixo:

$$EMQ(W) = \frac{1}{m} \sum_{j=1}^m \sum_{i=1}^n (y_{i,m} - \hat{y}_{i,m})^2, \quad (4.18)$$

ou

$$EMQ(W) = E\left[\sum_{i=1}^n (y_i - \hat{y}_i)^2\right], \quad (4.19)$$

nas quais $y_{i,m}$ é o valor do conjunto de dados da variável de saída y_i na leitura m , $\hat{y}_{i,m}$ é a sua estimativa calculada através da Rede Neural e W é o conjunto de todos os pesos. Deste modo, esta função calcula a média entre todos os pares entrada-saída da soma dos erros em cada variável de saída prevista.

Dado que é difícil obter uma solução analítica para o mínimo desta função (porque se considera funções de ativação não-lineares), utiliza-se métodos iterativos para o ajuste dos pesos visando minimizar esta função.

O método mais comumente utilizado é o método do Gradiente Descendente.

Este busca o mínimo de uma função realizando pequenos passos na direção oposta à derivada da função, dado que o gradiente de uma função indica a direção de maior crescimento. A variação aplicada a um parâmetro qualquer p do qual uma função F depende pode ser representada pela equação:

$$\Delta p = -\alpha \frac{\partial F}{\partial p}. \quad (4.20)$$

O parâmetro α indica o tamanho do passo dado e, no vocabulário de redes neurais, também é chamado de taxa de aprendizado. Não há um critério fixo para escolhê-lo, mas existem valores típicos que são adotados dependendo da natureza do problema.

Para problemas de regressão, costuma-se adotar os valores 0.05 e 0.1. Taxas de aprendizado muito baixas tornam o resultado obtido mais preciso e aumentam a probabilidade de se encontrar um mínimo global. Porém, tornam o treinamento mais lento. Por outro lado, taxas de aprendizado mais altas aceleram o treinamento, mas reduzem a probabilidade de se encontrar um mínimo global.

Dado que a estimativa do modelo depende dos pesos, o seu erro de predição também depende dos mesmos, de modo que usar este método é válido. No entanto, no caso de uma rede neural, existem dependências indiretas, fazendo com que as derivadas sejam resolvidas através da regra da cadeia. No treinamento, os valores de *bias* (viés) e todos os pesos da rede neural são ajustados conforme este método. Para analisar o que envolve os ajustes dos pesos por gradiente descendente, a seguir serão desenvolvidas as expressões de ajuste de pesos por este método.

Para isto, suponha uma rede MLP com k saídas. Dado que se deseja minimizar o erro médio quadrático, a Equação 4.20 pode ser transformada em:

$$\Delta p = -\alpha \frac{\partial EMQ}{\partial p}. \quad (4.21)$$

Como quaisquer parâmetros da rede neural afetam sua estimativa das saídas, pode-se usar a regra da cadeia:

$$\Delta p = -\alpha E \left[\sum_{i=1}^k \frac{\partial (y_i - \hat{y}_i)^2}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial p} \right]. \quad (4.22)$$

Aplicando-a novamente, obtém-se:

$$\Delta p = -\alpha E \left[\sum_{i=1}^k \frac{\partial (y_i - \hat{y}_i)^2}{\partial (y_i - \hat{y}_i)} \frac{\partial (y_i - \hat{y}_i)}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial p} \right], \quad (4.23)$$

ou

$$\Delta p = -\alpha E \left[\sum_{i=1}^k 2(y_i - \hat{y}_i)(-1) \frac{\partial \hat{y}_i}{\partial p} \right], \quad (4.24)$$

ou

$$\Delta p = 2\alpha E \left[\sum_{i=1}^k (y_i - \hat{y}_i) \frac{\partial \hat{y}_i}{\partial p} \right]. \quad (4.25)$$

O resultado da última derivada depende do tipo de parâmetro da rede. Caso haja uma dependência indireta, a regra da cadeia deve ser aplicada mais vezes.

Para desenvolver esta expressão para parâmetros específicos, suponha que a rede neural em questão possua apenas 1 entrada, 1 saída e 1 neurônio de camada oculta, para fins de simplicidade. Neste caso, a equação se torna:

$$\Delta p = 2\alpha E \left[(y_s - \hat{y}_s) \frac{\partial \hat{y}_s}{\partial p} \right], \quad (4.26)$$

onde s é a saída da rede. Caso o parâmetro seja um peso utilizado no neurônio de saída, a equação acima se torna:

$$\Delta w_{is} = 2\alpha E[(y_s - \hat{y}_s) \frac{\partial \hat{y}_s}{\partial w_{is}}], \quad (4.27)$$

onde w_{is} é o peso de uma entrada do neurônio de saída s . Caso este peso seja do *bias* (chamado de w_{bs}):

$$\Delta w_{bs} = 2\alpha E[(y_s - \hat{y}_s)], \quad (4.28)$$

porque o valor do viés é sempre 1. Caso seja a saída do neurônio da camada oculta, tem-se:

$$\Delta w_{hs} = 2\alpha E[(y_s - \hat{y}_s) \frac{\partial \hat{y}_s}{\partial w_{hs}}], \quad (4.29)$$

sendo w_{hs} o peso dado à saída do neurônio da camada oculta no neurônio de saída. A derivada restante também apresenta uma dependência indireta. Aplicando novamente a regra da cadeia, tem-se:

$$\Delta w_{hs} = 2\alpha E[(y_s - \hat{y}_s) \frac{\partial \hat{y}_s}{\partial u_s} \frac{\partial u_s}{\partial w_{hs}}], \quad (4.30)$$

onde u_s é a soma ponderada realizada no neurônio de saída. Calculando a última derivada, chega-se à equação final para o ajuste do peso da saída do neurônio oculto:

$$\Delta w_{hs} = 2\alpha E[(y_s - \hat{y}_s) \frac{\partial \hat{y}_s}{\partial u_s} v_h], \quad (4.31)$$

sendo v_h o resultado da função de ativação do neurônio da camada oculta. $\frac{\partial \hat{y}_s}{\partial u_s}$ é a derivada da função de ativação do neurônio de saída, logo depende da função aplicada.

Se os parâmetros a serem ajustados forem os pesos no neurônio da camada oculta, tem-se:

$$\Delta w_{ih} = 2\alpha E[(y_s - \hat{y}_s) \frac{\partial \hat{y}_s}{\partial u_s} \frac{\partial u_s}{\partial v_h} \frac{\partial v_h}{\partial u_h} \frac{\partial u_h}{\partial w_{ih}}], \quad (4.32)$$

onde $\frac{\partial u_s}{\partial v_h}$ é a derivada da combinação linear do *bias* com v_h , logo:

$$\Delta w_{ih} = 2\alpha E[(y_s - \hat{y}_s) \frac{\partial \hat{y}_s}{\partial u_s} (w_{hs}) \frac{\partial v_h}{\partial u_h} \frac{\partial u_h}{\partial w_{ih}}], \quad (4.33)$$

sendo w_{ih} o peso do valor i no neurônio oculto.

Se o peso for de um viés (chamado w_{bh} para a camada oculta), tem-se:

$$\Delta w_{bh} = 2\alpha E[(y_s - \hat{y}_s) \frac{\partial \hat{y}_s}{\partial u_s} (w_{hs}) \frac{\partial v_h}{\partial u_h}]. \quad (4.34)$$

Se o peso for da entrada x :

$$\Delta w_x = 2\alpha E[(y_s - \hat{y}_s) \frac{\partial \hat{y}_s}{\partial u_s} w_{hs} \frac{\partial v_h}{\partial u_h} x]. \quad (4.35)$$

Estes resultados podem ser generalizados para outras quantidades de entradas e saídas:

- Para um número maior de saídas, os ajustes dos parâmetros das camadas ocultas serão computados para cada saída e somados;
- Para um número maior de camadas ocultas, nas expressões dos pesos das camadas ocultas haverá outros termos de derivadas de funções de ativação dos neurônios ocultos e pesos, sendo que as expressões para os pesos de uma camada oculta terão estes termos das camadas posteriores;
- Para um número maior de neurônios em uma camada oculta, os ajustes nas camadas anteriores deverão ser calculados por neurônio desta camada e somados;
- Por fim, para um número maior de entradas, não há alteração no cálculo dos ajustes.

Com base nestes resultados, pode-se concluir alguns fatos:

- O ajuste dos pesos requer que, em primeiro lugar, seja calculado o erro médio obtido pela rede neural em todos os pares entrada-saída considerados no treino, dado que ele está presente na fórmula, o que exige que se realize uma estimativa inicial.
- O ajuste dos parâmetros de uma camada depende dos pesos atribuídos às saídas dos neurônios de camadas posteriores, o que torna interessante que estes pesos sejam ajustados previamente.
- Como as derivadas das funções de ativação dos neurônios estão presentes nas fórmulas, as suas expressões analíticas devem ser obtidas para que elas sejam aplicadas computacionalmente.
- Porque o erro médio quadrático é uma função quadrática, ele possui um ponto de mínimo. Mas não há garantias de que haja apenas um mínimo global. Podem existir mínimos locais e, por ser utilizado um algoritmo iterativo, a depender dos pesos iniciais, ele pode convergir em um mínimo local.

Devido às duas primeiras conclusões, os ajustes são realizados de trás para frente, iniciando na camada de saída e terminando nos pesos das entradas.

Por conta da quarta conclusão, uma preocupação no projeto de uma rede neural é escolher os valores iniciais dos pesos. Geralmente, utiliza-se pesos aleatórios, a não ser que se disponha de alguma informação prévia do problema.

Com base nesta discussão, criou-se um algoritmo baseado no método do gradiente descendente chamado retropropagação do erro ou *backpropagation*, que é exibido como o Algoritmo 1:

Algoritmo 1: RETROPROPAGAÇÃO

Entrada: *tol* (limiar de variação do EMQ abaixo do qual se considera que houve convergência) e *lim_{it}* (limite de iterações)

```

1 início
2   Inicializa pesos;
3    $n_{it} \leftarrow 0$  // número de iterações atual;
4    $\Delta EMQ \leftarrow +\infty$  // número de iterações atual;
5   enquanto  $\Delta EMQ > tol$  e  $n_{it} < lim_{it}$  faça
6     Apresentar os exemplos do conjunto de treinamento para a rede e
       obter as estimativas  $\hat{y}_{ij}$  de cada saída  $i$  para cada exemplo  $j$ ;
7     Calcular os erros  $y_{ij} - \hat{y}_{ij}$ ;
8     Calcular o EMQ a partir dos erros calculados;
9     Calcular os ajustes dos pesos dos neurônios da camada de saída;
10    Calcular os ajustes dos pesos dos neurônios da última camada oculta;
11    Calcular os ajustes dos pesos dos neurônios da penúltima camada
       oculta;
12    (...);
13    Calcular os ajustes dos pesos dos neurônios da primeira camada oculta;
14    Calcular o novo valor para EMQ;
15     $\Delta EMQ \leftarrow$  diferença entre os dois últimos valores do EMQ;
16     $n_{it} \leftarrow n_{it} + 1$ 
17 fim
18 fim

```

O valor de tolerância típico adotado neste algoritmo é 10^{-4} . Dependendo do tempo de treinamento e da complexidade da rede construída, pode-se atribuir um limite para o número de iterações ou não. Vê-se que o algoritmo executa várias vezes sobre o mesmo conjunto de dados. Uma iteração, no vocabulário de redes neurais, também é chamada de *época de treinamento*.

Uma última conclusão que se pode obter através desta descrição de uma RNA é que o ajuste nos pesos das entradas é diretamente proporcional ao valor assumido pelas entradas. Isto, somado ao fato de que todos os ajustes são proporcionais ao valor do erro, este sujeito à escala das variáveis de saída, é o que torna este algoritmo muito sensível à escala das variáveis empregadas. Entradas que assumem valores maiores recebem ajustes maiores e, por isto, escalas diferentes entre as variáveis de entrada podem prejudicar o treinamento. Além disto, usar variáveis que assumem

valores muito altos afeta o tamanho dos ajustes realizados e, conforme discutido anteriormente para o método do gradiente, isto reduz a chance de se encontrar o mínimo global da função do EMQ.

Devido a esta alta sensibilidade à escala, costuma-se utilizar junto a redes neurais métodos de pré-processamento para reduzir a escala das variáveis e igualar as escalas de todas as variáveis.

Formas de Treinamento

Um resultado prático é que o treinamento, para um conjunto de dados grande, pode se tornar muito lento, como citado por Haykin [17] e Silva [18]. Para acelerar o procedimento, pode-se treinar a rede, em vez de com o conjunto de dados completos, com partes dos dados, realizando o treino uma vez para cada parte. Além disto, pode ser interessante para sistemas adaptativos e variantes com o tempo, que a rede “esqueça” dados muito antigos. Isto motivou a criação de três formas de treinamento com retropropagação:

- Treinamento em *batch* ou batelada: utiliza-se o conjunto de dados inteiro. A retropropagação ajusta em uma execução o conjunto de dados pelo EMQ do conjunto completo, tendo todos os pares entrada-saída a mesma influência nos ajustes; este treinamento é mais lento;
- Treinamento por Regra Delta: o algoritmo de *backpropagation* é utilizado 1 vez por par entradas-saídas. Este método tende a ser mais rápido, mas cada ajuste desfaz parte dos ajustes mais antigos;
- Treinamento em lotes: utiliza-se subconjuntos do conjunto de dados (quantidades maiores que 1 de pares entradas-saídas). Este é o meio termo entre o treinamento em batelada e o treinamento regra delta, sendo mais rápido que o primeiro e tendo um menor efeito de “esquecimento” que o segundo. Um valor típico usado para o tamanho do lote é 200, mas pode-se utilizar valores mais altos caso o treinamento possa ser mais lento.

4.2 Técnicas de Pré-Processamento

Técnicas de pré-processamento são utilizadas para selecionar os dados aplicados aos modelos de aprendizado e realizar transformações convenientes aos dados selecionados, de modo a reduzir o erro de predição [3]. Nesta seção, são descritas as técnicas utilizadas na modelagem deste sistema.

4.2.1 Normalização

A normalização é uma técnica de baixo custo computacional que transforma um conjunto de dados por meio do ajuste da sua escala. A partir de N leituras de uma variável x , calcula-se a média amostral da mesma e, com este valor, calcula-se o desvio padrão amostral de x :

$$\hat{\sigma}_x = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \hat{\mu}_x)^2}, \quad (4.36)$$

onde

$$\hat{\mu}_x = \frac{1}{N} \sum_{i=1}^N x_i. \quad (4.37)$$

Cada valor x_i corresponde a uma leitura observada de x . Com estes valores, realiza-se a seguinte transformação em x , gerando uma nova variável x' :

$$x' = \frac{x - \hat{\mu}_x}{\hat{\sigma}_x}. \quad (4.38)$$

A nova variável x' tem como características média nula e desvio padrão unitário. Esta é uma técnica utilizada em diversos problemas devido à sensibilidade de muitos modelos de Aprendizado de Máquina à escala das variáveis, como os previamente apresentados.

4.2.2 Particionamento dos Dados por Análise de Agrupamentos

Em vários casos, é conveniente particionar um conjunto de dados em partes com diferentes características e treinar mais de um modelo de regressão, cada modelo atuando em uma parte dos dados. Isto torna cada modelo especializado em um tipo de comportamento. Duas maneiras de isto ser realizado são:

- Separar manualmente, caso se saiba determinar quais critérios são interessantes;
- Separar utilizando um algoritmo de análise de agrupamentos ou *clustering* (um algoritmo de Aprendizado de Máquina de Aprendizado Não-Supervisionado), que busca conjuntos de valores que sigam comportamentos parecidos, sem que se especifique em qual conjunto cada valor está.

Dado que há diversos fatores que não se pôde estimar, a abordagem escolhida foi o uso de um algoritmo de análise de agrupamentos, o K-Médias. A escolha por este algoritmo foi feita por conta de seu baixo custo computacional (tanto em

memória quanto em processamento) – característica importante para conjuntos de dados grandes. O K-Médias é descrito a seguir.

K-Médias

Como descrito anteriormente, o K-Médias é um algoritmo que encontra subconjuntos (ou agrupamentos) de valores em um conjunto de dados sem indicações de quais são os agrupamentos de cada valor.

O processo executado pelo algoritmo é iterativo e encontra agrupamentos que minimizem a variância dentro de cada agrupamento (intra-grupos) [3]. A variância intra-grupos é definida como:

$$V(\Omega) = \sum_{j=1}^N \sum_{i=1}^K u_i(x_j) \|x_j - \omega_i\|^2, \quad (4.39)$$

onde x_j é um valor das variáveis consideradas, Ω é um vetor com os pontos centrais de todos os grupos, ω_i é o centro do agrupamento i e u_i é uma função de pertencimento de um registro x_j a um agrupamento G_i , que assume o valor 1 se $x_j \in G_i$ e o valor 0 se $x_j \notin G_i$. Em resumo, a minimização deste critério (a variância) cria grupos nos quais os registros apresentam as menores distâncias possíveis dos centros dos seus grupos. Os centros e pontos podem ser representados em uma ou mais dimensões.

A quantidade de agrupamentos é fornecida pelo usuário. Dado um conjunto inicial de centros fornecido, o algoritmo opera alternando os seguintes passos:

- Para cada centro, identifica-se os pontos que sejam mais próximos dele do que são de outros centros. Estes pontos tornam-se membros do grupo deste centro;
- Para cada grupo, calcula-se a média dos valores, e o valor médio torna-se o novo centro.

Para calcular a distância, pode-se utilizar diversas métricas. A mais comum é a distância euclidiana. Dados dois registros $x = (x_1, x_2, \dots, x_n)$ e $y = (y_1, y_2, \dots, y_n)$, a distância euclidiana pode ser calculada como:

$$D(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}. \quad (4.40)$$

Para x e y unidimensionais, ela é simplesmente:

$$D(x, y) = \sqrt{(x - y)^2} = |x - y|. \quad (4.41)$$

Este procedimento é executado até que um número máximo de iterações seja atingido ou até que a alteração no valor da variância intra-grupos após um passo

seja menor que um limiar de tolerância. O algoritmo é classificado como guloso, pois, em vez de calcular uma solução analítica para a minimização da variância dentro dos grupos, executa um procedimento iterativo que, a cada passo, reduz o valor da métrica que se deseja minimizar. No entanto, não há uma garantia de que o mínimo da função seja encontrado. É possível que o algoritmo convirja para um mínimo local.

Por conta disto, uma preocupação são os valores iniciais para os centros. Um procedimento comumente realizado é o uso de centros iniciais aleatórios. No entanto, pode ser necessário repetir a execução do algoritmo para encontrar um resultado melhor. Para evitar a necessidade de repetição, neste trabalho é utilizado um critério alternativo, o algoritmo K-Médias++, proposto por Arthur e Vassilvitskii [21]. A ideia geral do algoritmo é encontrar centros iniciais com a maior distância entre si, através de outro processo iterativo. Para isto, define-se uma função de distância $D(x)$. $D(x)$ corresponde à distância mais curta de um registro x do centro mais próximo dentre os já escolhidos. O algoritmo executa os seguintes passos:

- Do conjunto de registros X , escolher aleatoriamente um centro ω_1 ;
- Escolher o próximo centro c_i , selecionando $c_i = x' \in X$ com a probabilidade
$$\frac{D(x')^2}{\sum_{x \in X} D(x)^2}.$$
- Repetir o passo anterior até que se gere a quantidade desejada de centros.

Esta modificação torna mais provável que os centros iniciais sejam mais afastados. De acordo com o trabalho que propôs este algoritmo [21], a inicialização do K-Médias++ apresenta ganhos de desempenho com relação ao K-Médias tradicional com centros iniciais aleatórios.

4.3 Divisões dos Dados para Validação dos Modelos

Na avaliação do desempenho (ou validação) das previsões dadas por modelos de Aprendizado de Máquina, separa-se o conjunto de dados em duas partes principais: o conjunto de treinamento e o conjunto de testes. Como os nomes já indicam, o conjunto de treinamento corresponde aos dados que serão usados para ajustar o candidato a modelo, e o conjunto de testes contém os dados que serão usados para calcular as estimativas e verificar o desempenho.

Como apresentado na definição dos modelos de regressão discutidos aqui, o critério para o ajuste dos seus parâmetros é a minimização do erro quadrático ou erro médio quadrático. Como o ajuste realizado depende dos dados do conjunto de

treinamento, isto significa que o desempenho varia de acordo com os dados fornecidos. Por isto, há diferentes métodos para escolher o conjunto de dados e o conjunto de teste. Alguns dos principais são descritos a seguir.

4.3.1 Validação Simples

Este é o método mais simples de dividir os dados. Escolhe-se uma porcentagem dos dados como conjunto de treinamento, e o restante será usado como conjunto de testes. A seleção dos dados pode ser aleatória ou não. No caso de a seleção ser aleatória, o procedimento também é chamado de Método *Holdout*. Uma fração tipicamente utilizada é atribuir 70% dos dados para o conjunto de treinamento e 30% para o conjunto de testes.

4.3.2 Validação Cruzada

A validação cruzada é um método iterativo que gera diversas divisões dos dados. O objetivo é verificar o desempenho dos algoritmos em conjuntos de dados com diferentes características. O desempenho final do algoritmo é a média dos desempenhos de todas as divisões. Há diversos tipos de procedimentos de validação cruzada; cita-se abaixo os dois principais:

- *Leave-p-out*: este método separa uma quantidade arbitrária p de leituras como o conjunto de teste e utiliza o restante como o conjunto de treinamento. Todas as combinações possíveis são testadas. Ou seja, sendo n a quantidade de leituras do conjunto de dados e p a quantidade de leituras separadas para testes, $C_n^p = \frac{n!}{(n-p)! p!}$, as combinações de n elementos tomados p a p , é a quantidade de rodadas realizadas. Por isto, para muitos casos este é um método muito longo, em alguns casos sendo impraticável. Um exemplo disto: se $n = 100$ e $p = 30$, aproximadamente 3×10^{25} rodadas são necessárias. Por testar todas as combinações, este método é considerado um método *exaustivo* de validação cruzada;
- *K-fold*: divide o conjunto de dados em uma quantidade arbitrária k de partes de mesmo tamanho. A cada rodada, uma das partes é escolhida como conjunto de teste e as restantes são utilizadas como conjunto de treinamento. Este é um procedimento muito mais rápido que o *leave-p-out*, pois apenas executa k rodadas, e avalia melhor a capacidade de generalização dos modelos que uma validação simples. Valores típicos para k são 4 e 10.

Dado que os algoritmos de Aprendizado de Máquina podem funcionar diferentemente dependendo das características do conjunto de dados, a Validação Cruzada

torna-se interessante por executar testes independentes em mais de uma partição do conjunto de dados, efetivamente gerando mais de um conjunto de treinamento e mais de um conjunto de testes com diferentes características. Por isto, ela será aplicada neste projeto. Dado que o conjunto de dados apresenta uma grande quantidade de leituras, optou-se por utilizar a versão *K-fold*.

Este capítulo encerra a discussão teórica dos métodos que serão utilizados. No próximo Capítulo, os modelos selecionados pelo Capítulo 3 serão estimados com as técnicas de Aprendizado de Máquina descritas aqui, e os resultados obtidos serão analisados.

Capítulo 5

Resultados

Com base nas informações obtidas através da análise dos dados realizada no Capítulo 3, foram estabelecidos três possíveis candidatos de modelos: ARIX, NARIX e Modelo Estático, todos utilizando apenas as leituras mais recentes dos três sensores. Para selecionar o melhor destes modelos, cada um foi utilizado em uma simulação com os dados coletados mencionados no Capítulo 3. Este capítulo tem como objetivos explicar a realização destas simulações e apresentar os resultados encontrados.

Dois cenários foram programados para os testes dos modelos:

- Uma perda de cobertura ocorre no momento da recepção de uma posição do GPS, de modo que o sistema deva estimar esta posição perdida. As posições anteriores verdadeiras obtidas através do GPS são conhecidas pelo sistema. Este é um teste de acurácia, ou seja, mede quão próximas são as estimativas individuais das posições reais. Repete-se o teste nos diversos pontos do conjunto de testes;
- Uma perda de cobertura ocorre em um trecho do conjunto de dados – o conjunto de testes –, e todas as posições deste trecho devem ser estimadas. Ou seja, deve-se estimar a trajetória tomada pelo veículo, e não só posições individuais. Os únicos valores do passado conhecidos são os valores anteriores a este trecho.

Os dois tipos de testes sinalizam características importantes do sistema. O primeiro é um indicador da acurácia da estimativa para uma leitura. Já o segundo, dado que se utilizam posições estimadas anteriormente para construir o caminho percorrido pelo carro, é um indicador do efeito do erro que é acumulado entre as predições. Além disto, os dois testes respondem às seguintes questões:

- Qual é o melhor tipo de modelo para o cenário – linear ou não-linear;

- Quantas leituras anteriores do GPS são necessárias considerar para obter a melhor precisão, dado que já se considerou qual é o melhor valor de atraso para os sensores;
- Qual o modelo mais adequado para o cenário.

Para verificar a acurácia, é necessário utilizar uma métrica de erro para indicar o desempenho dos modelos. Foi escolhido calcular a distância entre a posição real e a posição estimada, por ser possível estabelecer uma comparação com a precisão de receptores GPS e por ser de mais fácil compreensão que os erros em graus, obtidos comparando as coordenadas.

Para calcular a distância entre dois pontos na Terra, há diversas métricas que fazem aproximações baseadas na curvatura ou em formas geométricas com as quais o planeta pode ser modelado. A expressão usada para a distância entre dois pontos A e B quaisquer é a fórmula de Haversine [22], mostrada nas equações abaixo:

$$D = 2R \times \arcsen(\sqrt{h}), \quad (5.1)$$

onde

$$h = \text{sen}^2\left(C \times \frac{\Delta Lat}{2}\right) + \cos(C \times LatA) \times \cos(C \times LatB) \times \text{sen}^2\left(C \times \frac{\Delta Lon}{2}\right), \quad (5.2)$$

onde $\Delta Lat = LatA - LatB$, $\Delta Lon = LonA - LonB$, $C = \pi/180$ (constante para converter graus para radianos) e R é o raio da Terra (aproximadamente 6.371 km [22]).

Em cada cenário de testes, foi considerada a melhor configuração do sistema a que apresentar o menor erro médio. A vantagem de utilizar a fórmula de Haversine é o fato de ela considerar a curvatura da Terra, o que evita erros de aproximação para pontos muito próximos que ocorrem quando se emprega a geometria esférica. Isto acontece, por exemplo, quando é utilizada a Lei dos Cossenos, um método utilizado para calcular distâncias longas entre dois pontos [22].

As simulações foram programadas com a linguagem Python, com o auxílio das bibliotecas `scikit-learn` [23] - que possui implementações de diversos modelos de Aprendizado de Máquina - e `matplotlib` [16] - para a elaboração dos gráficos.

Os parâmetros das simulações foram os seguintes:

- Quantidade de posições anteriores: variada de 0 a 10, dado o comportamento de média móvel das séries temporais das coordenadas e a presença de poucos valores significativos de autocorrelação e autocorrelação parcial;
- Estimadores: Regressão Linear e Rede Neural Artificial com 1 camada oculta apresentando entre 1 e 50 neurônios;

- Atraso das leituras dos sensores: 0 (ou seja, utilizar a leitura mais atual), dada a discussão anterior sobre a correlação cruzada;
- Utilização do K-Médias como método de particionar o conjunto de dados pelo comportamento do veículo: foram comparados os desempenhos com e sem a utilização desta técnica.

Para particionar o conjunto de dados pelo comportamento, em uma tentativa de suavizar os efeitos de média móvel e de auxiliar o modelo a capturar o comportamento dinâmico do veículo, o K-Médias com a inicialização pelo critério do K-Médias++ foi utilizado. As variáveis utilizadas para representar o comportamento do veículo foram as duas frequências de rotação das rodas e a velocidade de translação do veículo. Para cada subconjunto de dados, um modelo independente foi utilizado.

O conjunto de dados foi dividido através de uma Validação Cruzada *K-fold* que particionou o conjunto em 4 partes, possibilitando a realização de 4 testes com distribuições diferentes de dados, mantendo a proporção de 75% dos dados para o treinamento e 25% dos dados para o teste.

Conforme a análise realizada no Capítulo 3, os modelos utilizados aqui empregam a primeira diferença, sendo modelos integrados de primeira ordem. Para o teste de acurácia, os melhores modelos foram o modelo ARIX (implementado através da Regressão Linear) e o modelo NARIX (implementado através de uma rede neural de 15 camadas – quantidades maiores de neurônios não causaram diferenças), com 3 termos autorregressivos e 1 termo exógeno (sensores), sem o uso do algoritmo K-Médias. Os dois modelos obtiveram o erro médio de 2.25 metros. As Figuras 5.1 a 5.4 exibem os trajetos usados para treino (em preto) e os trajetos usados para teste (azul para os valores reais e vermelho para os valores estimados) para cada arranjo da validação cruzada, e as Figuras 5.5 a 5.8 exibem o comportamento temporal do erro para este teste. Os gráficos apresentaram um comportamento idêntico para os dois modelos, de modo que apenas 1 gráfico foi incluído para os dois. Há uma grande sobreposição entre as linhas azul (pontos reais) e vermelha (pontos estimados) nas Figuras 5.1 a 5.4, devido aos valores baixos assumidos pelo erro em comparação com a extensão do trajeto total realizado pelo veículo.

O uso da segunda diferenciação aumenta o erro médio das predições para 6.668 metros, um indicativo de que de fato há uma perda de informações associada à diferenciação. Duas conclusões que se pode extrair destes resultados são:

- Para a predição de um ponto individual, as estimativas calculadas pela Regressão Linear e pela Rede Neural são equivalentes;
- Os dois modelos conseguem realizar a predição de uma posição individual de

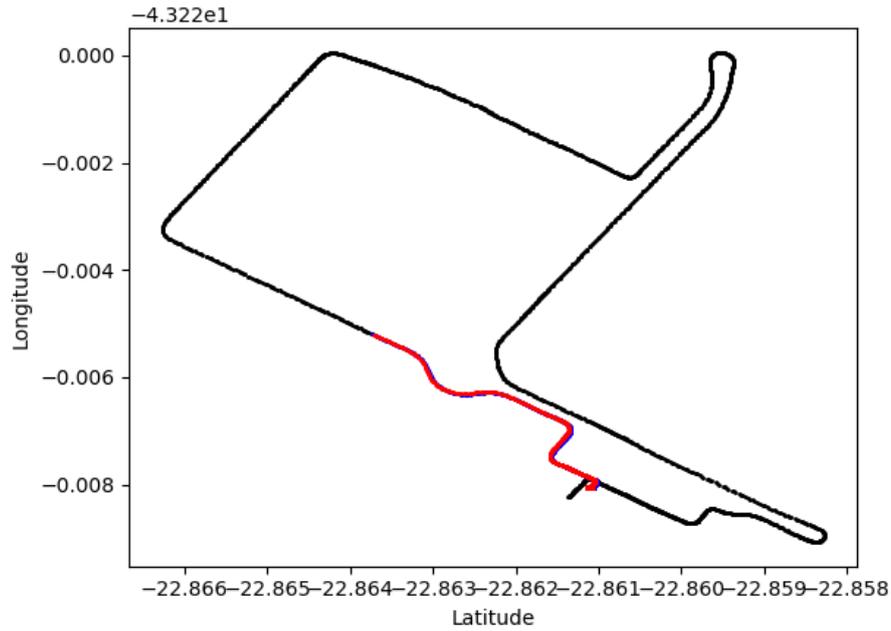


Figura 5.1: Exibição dos trajetos de treino e teste para o primeiro arranjo da validação cruzada no teste de acurácia.

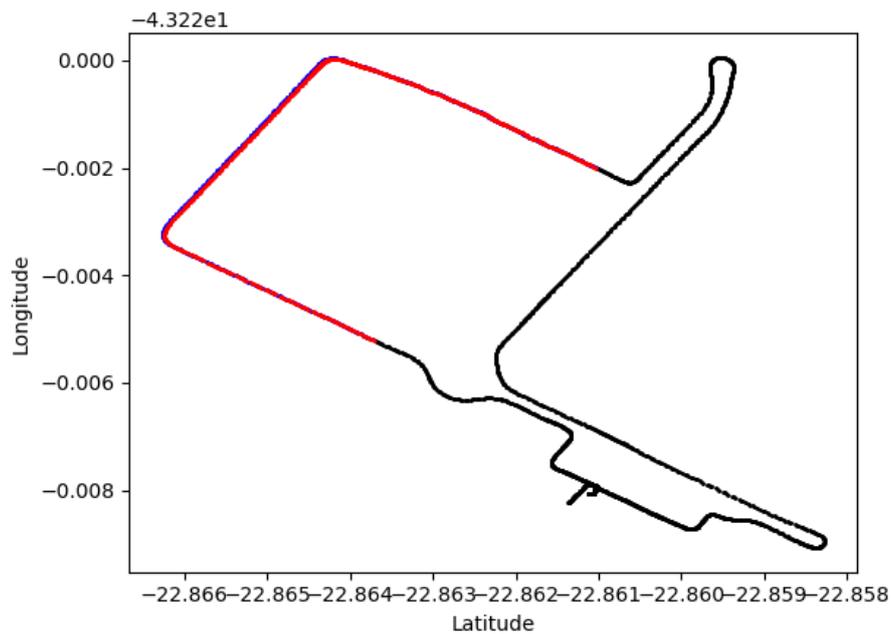


Figura 5.2: Exibição dos trajetos de treino e teste para o segundo arranjo da validação cruzada no teste de acurácia.

maneira eficiente, mantendo a média do erro no valor de 2.2 metros, tolerável para diversos tipos de aplicações, conseguindo suprir eventuais falhas pontuais de um sistema de posicionamento por satélite. Além disto, a abordagem utilizada é útil para a predição de posições futuras. Deste modo, os modelos apresentados são preditores eficientes no contexto de Séries Temporais.

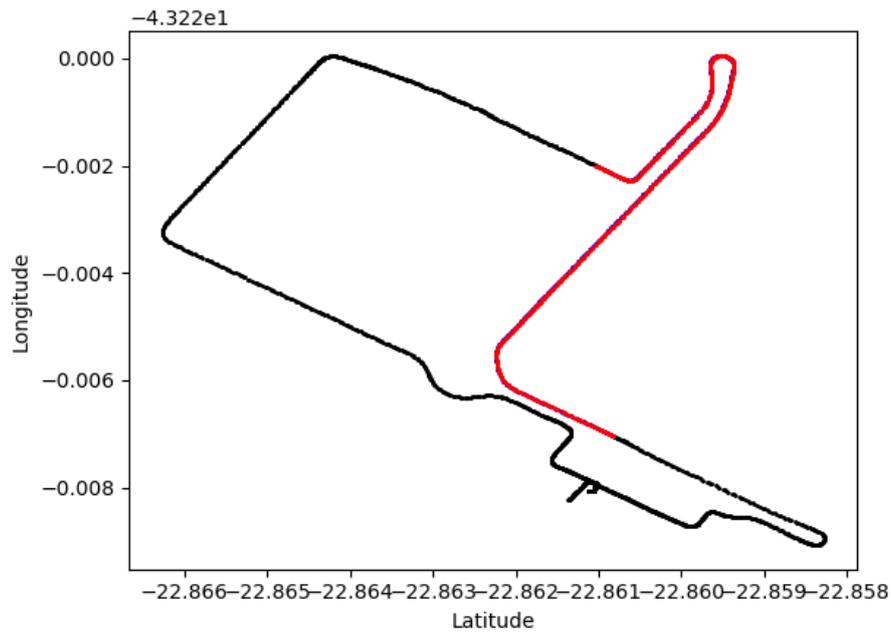


Figura 5.3: Exibição dos trajetos de treino e teste para o terceiro arranjo da validação cruzada no teste de acurácia.

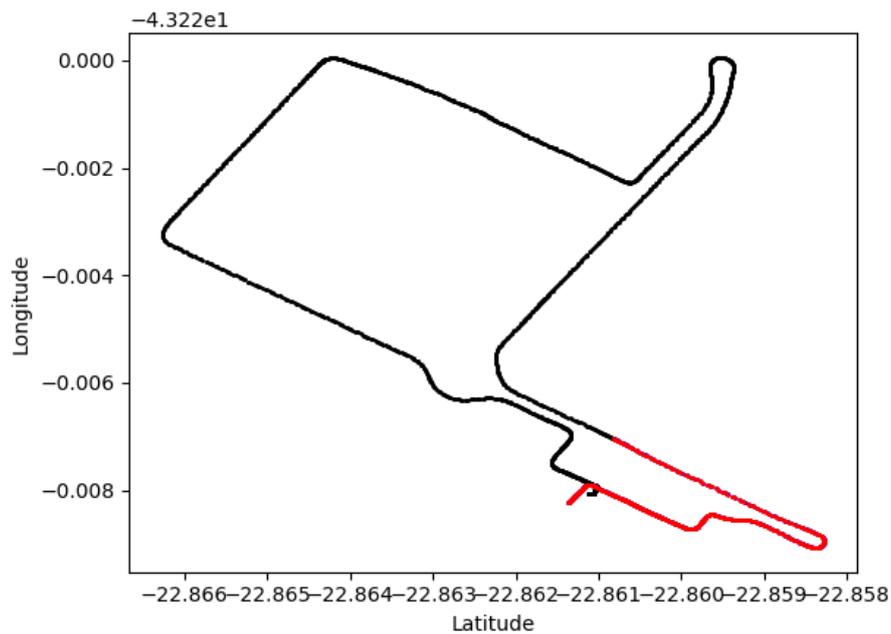


Figura 5.4: Exibição dos trajetos de treino e teste para o primeiro arranjo da validação cruzada no teste de acurácia.

Já o teste da previsão de um trecho inteiro não apresentou resultados adequados. O mesmo modelo ARIX do teste de acurácia foi aplicado na previsão de um trecho, e ele obteve um erro médio de 237,138 metros. O modelo NARIX mencionado anteriormente obteve um erro médio de 647.5 metros. No entanto, ao se aumentar o número de neurônios da camada oculta, obteve-se uma diminuição do erro

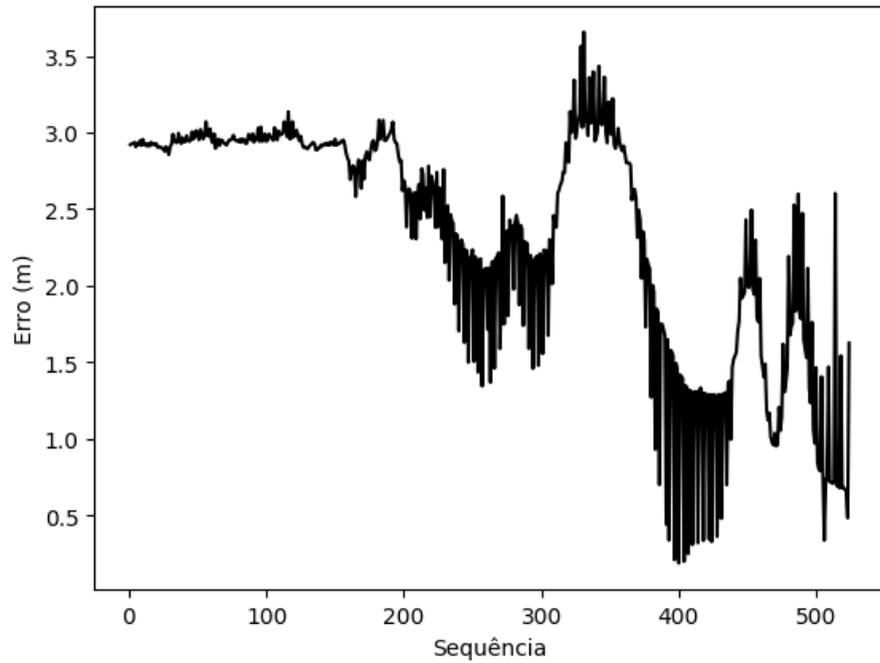


Figura 5.5: Comportamento temporal do erro para o primeiro arranjo da validação cruzada no teste de acurácia.

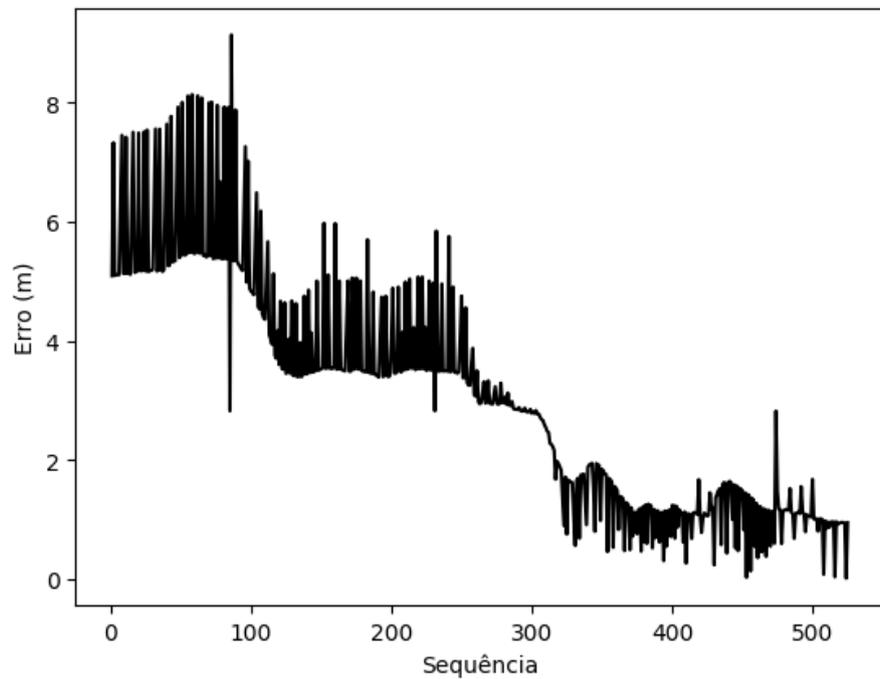


Figura 5.6: Comportamento temporal do erro para o segundo arranjo da validação cruzada no teste de acurácia.

médio. Com 50 neurônios na camada oculta, o valor diminuiu para 378.47 metros, sem aumentos para o erro médio da estimativa de uma única posição. Isto indica que aumentar a quantidade de neurônios torna a rede neural mais resistente à acumulação de erros, ainda que o erro médio apresente um valor maior de erro médio

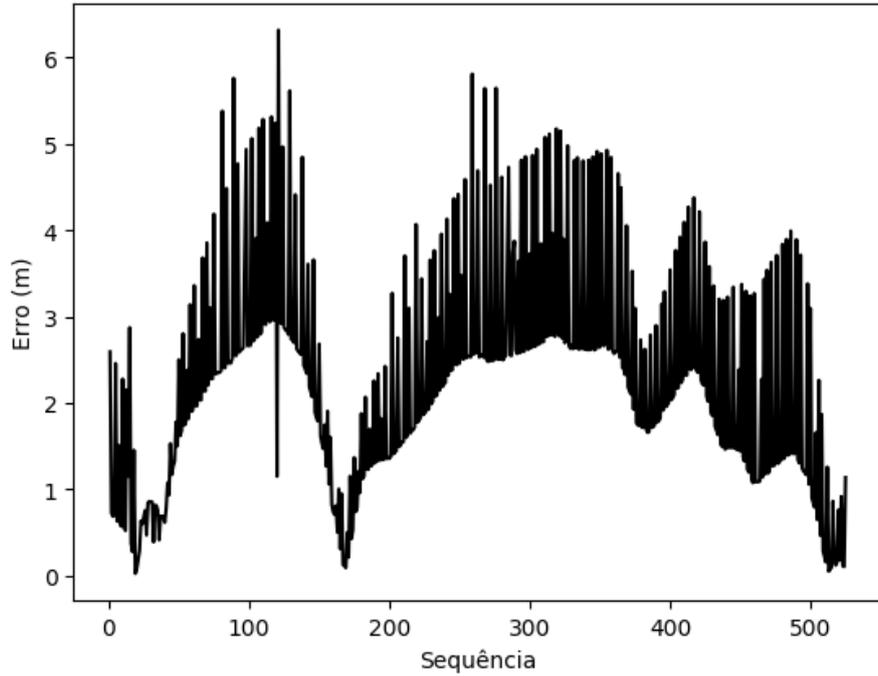


Figura 5.7: Comportamento temporal do erro para o terceiro arranjo da validação cruzada no teste de acurácia.

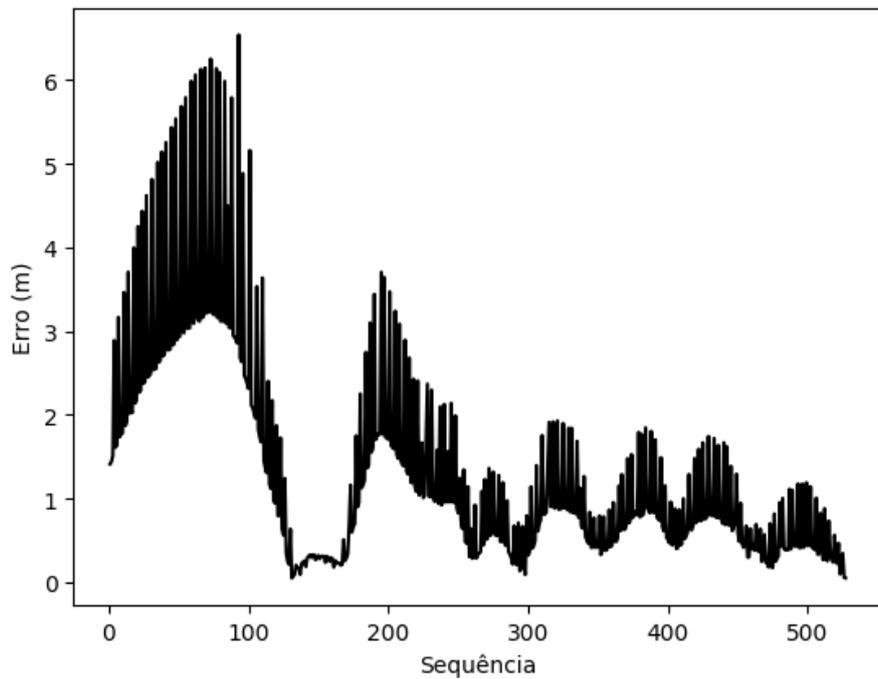


Figura 5.8: Comportamento temporal do erro para o quarto arranjo da validação cruzada no teste de acurácia.

que o do modelo linear. Para este teste, as Figuras 5.9 a 5.12 exibem os trajetos usados para treino (em preto) e os trajetos usados para teste (azul para os valores reais e vermelho para os valores estimados) para cada arranjo da validação cruzada.

As Figuras 5.13 a 5.16 exibem o comportamento temporal do erro, as Figuras de 5.17 a 5.20 exibem o comportamento temporal do erro da latitude e as Figuras de 5.21 a 5.24 exibem o comportamento temporal do erro da longitude. Nestas Figuras, os gráficos da coluna são relacionados ao modelo linear e os gráficos da coluna direita são do modelo não-linear.

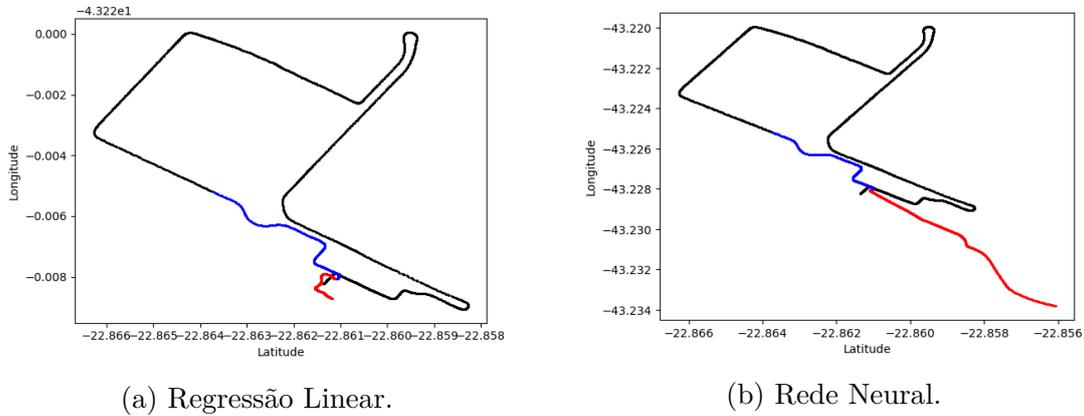


Figura 5.9: Exibição dos trajetos de treino e teste para o primeiro arranjo da validação cruzada na previsão de um trecho.

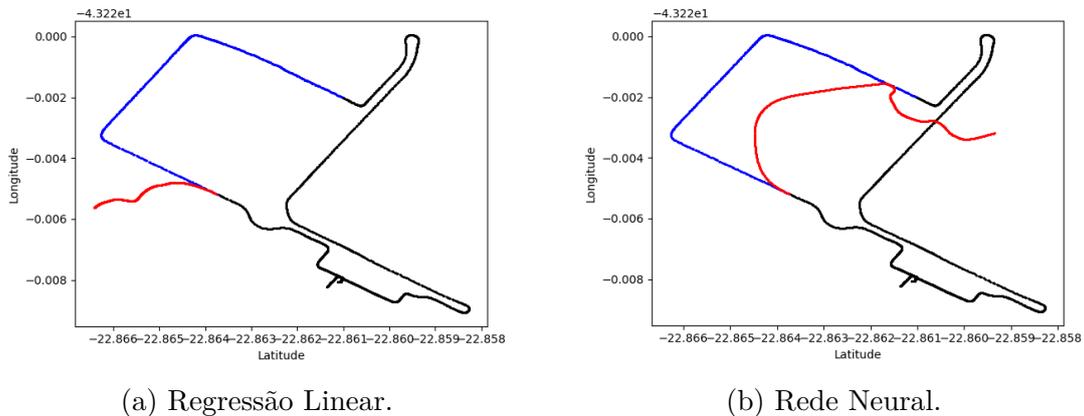
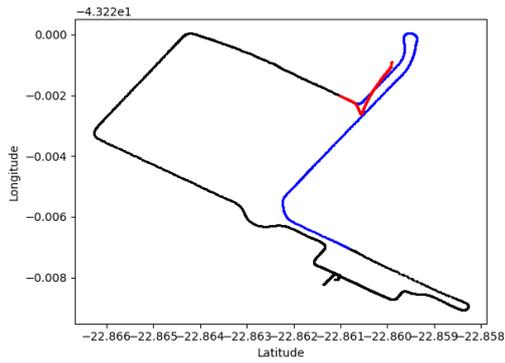
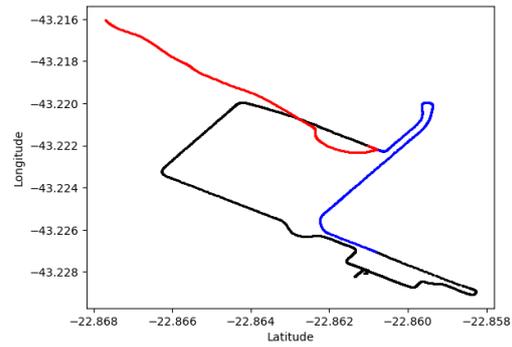


Figura 5.10: Exibição dos trajetos de treino e teste para o segundo arranjo da validação cruzada na previsão de um trecho.

Como é possível observar pelo comportamento dos erros nas Figuras 5.13 a 5.24, apesar de as primeiras previsões apresentarem boa acurácia, o erro de cada previsão é acumulado rapidamente, gerando uma tendência crescente. Uma tendência crescente também ocorre nos módulos dos erros das coordenadas. Este fenômeno ocorre nos dois tipos de modelos. O acúmulo de erros é um problema comum em estimadores que utilizam estimativas anteriores para gerar novas previsões, processo também chamado de operação em série-paralelo, como é descrito por Belhajem [9] e Calôba [24]. A imprecisão crescente das leituras anteriores fornecidas ao modelo não é considerada por ele, gerando valores para estas variáveis de entrada que estão

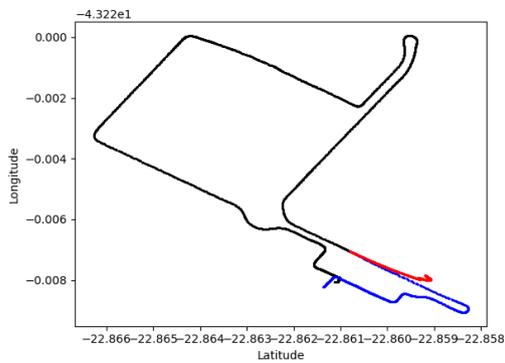


(a) Regressão Linear.

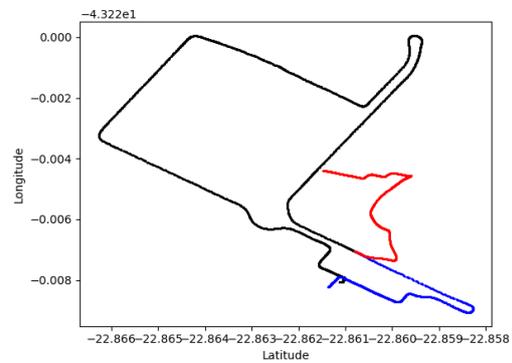


(b) Rede Neural.

Figura 5.11: Exibição dos trajetos de treino e teste para o terceiro arranjo da validação cruzada na previsão de um trecho.

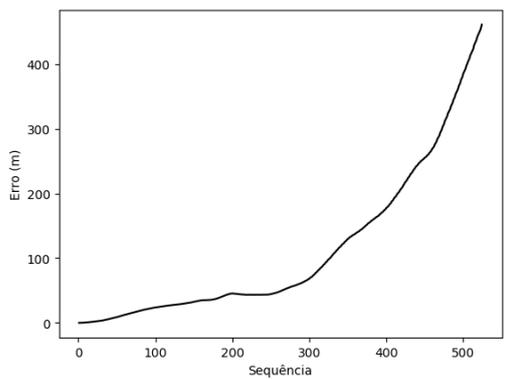


(a) Regressão Linear.

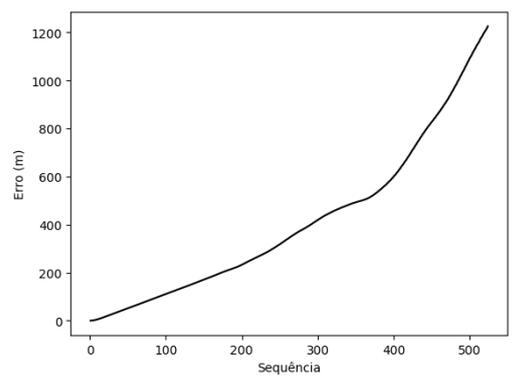


(b) Rede Neural.

Figura 5.12: Exibição dos trajetos de treino e teste para o quarto arranjo da validação cruzada na previsão de um trecho.



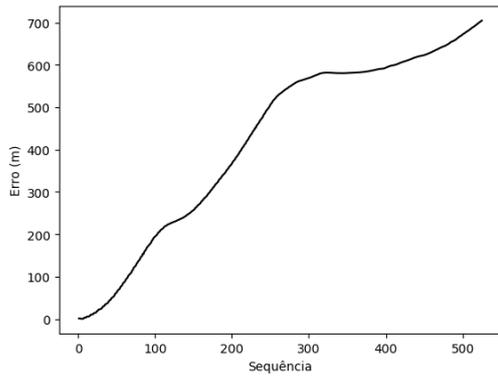
(a) Regressão Linear.



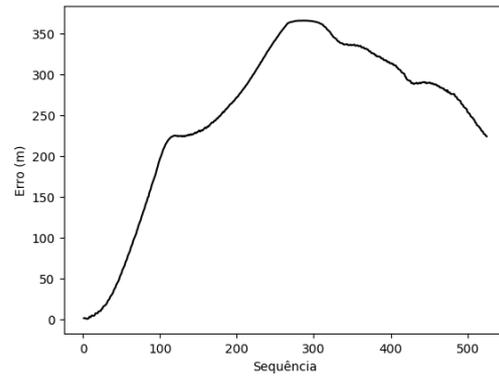
(b) Rede Neural.

Figura 5.13: Comportamento temporal do erro para o primeiro arranjo da validação cruzada na previsão de um trecho.

fora do intervalo que foi usado no treino, de modo que o erro é transmitido às novas estimativas. Além disto, no cenário avaliado por este sistema, o erro também indica que o modelo não consegue capturar parte das características dinâmicas do

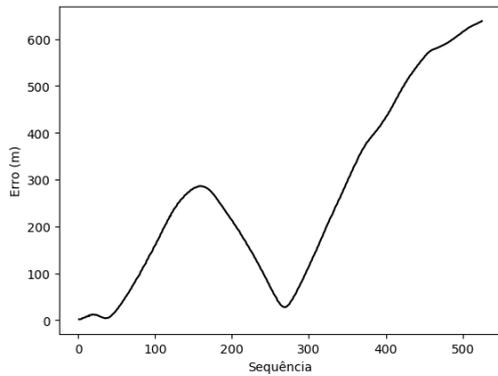


(a) Regressão Linear.

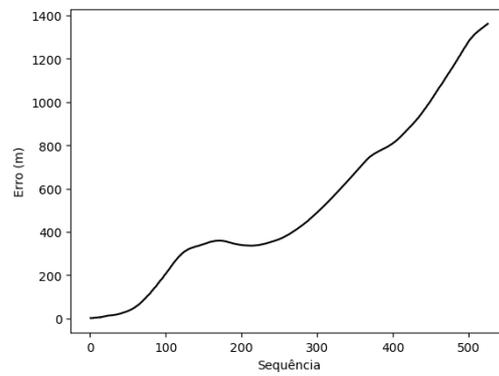


(b) Rede Neural.

Figura 5.14: Comportamento temporal do erro para o segundo arranjo da validação cruzada na previsão de um trecho.

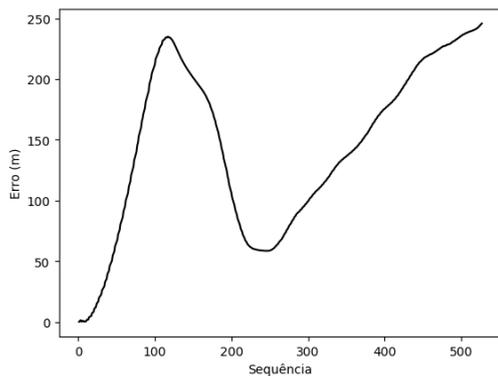


(a) Regressão Linear.

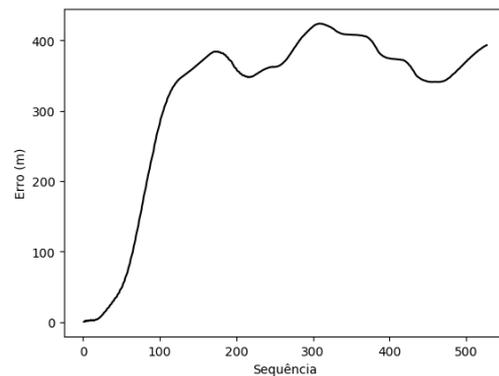


(b) Rede Neural.

Figura 5.15: Comportamento temporal do erro para o terceiro arranjo da validação cruzada na previsão de um trecho.



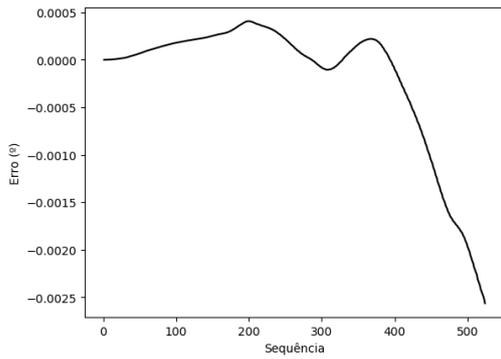
(a) Regressão Linear.



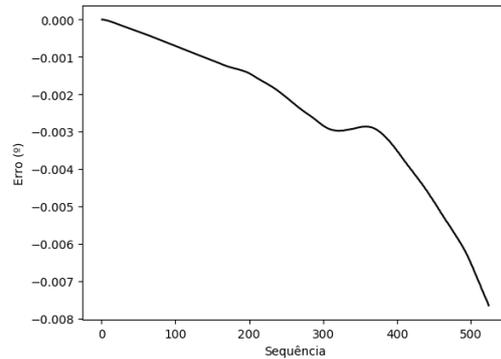
(b) Rede Neural.

Figura 5.16: Comportamento temporal do erro para o quarto arranjo da validação cruzada na previsão de um trecho.

sistema. Uma comparação entre os trechos previstos pela Regressão Linear e pela Rede Neural indica que, na Rede Neural, as previsões divergem mais rapidamente.

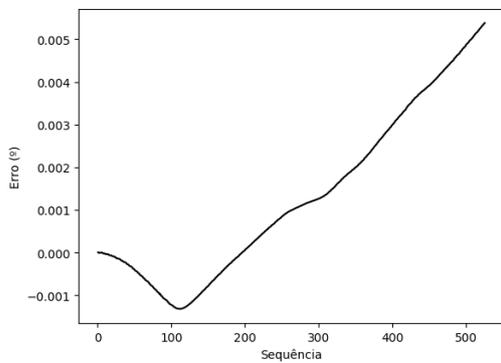


(a) Regressão Linear.

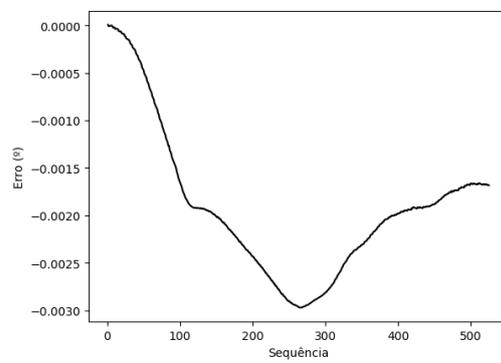


(b) Rede Neural.

Figura 5.17: Comportamento temporal do erro da latitude para o primeiro arranjo da validação cruzada na previsão de um trecho.

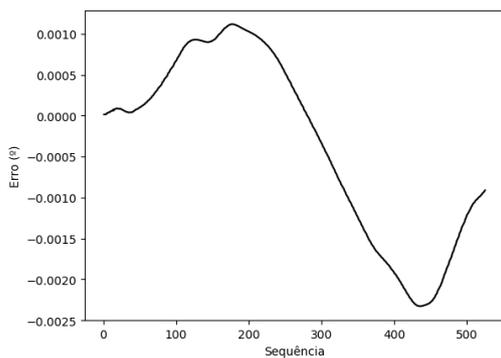


(a) Regressão Linear.

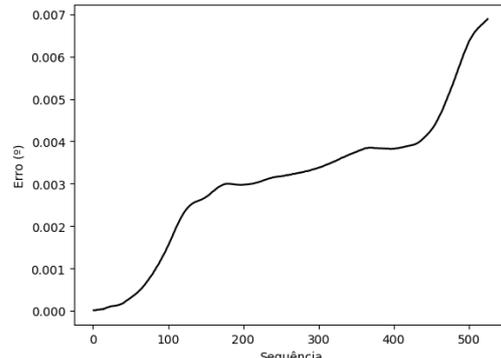


(b) Rede Neural.

Figura 5.18: Comportamento temporal do erro da latitude para o segundo arranjo da validação cruzada na previsão de um trecho.



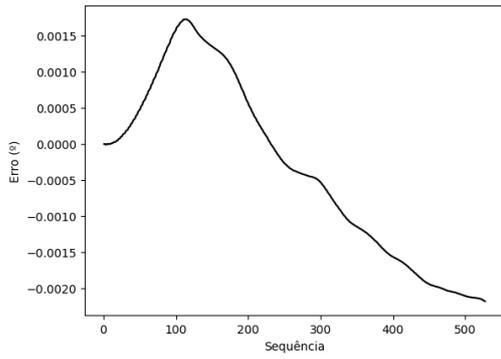
(a) Regressão Linear.



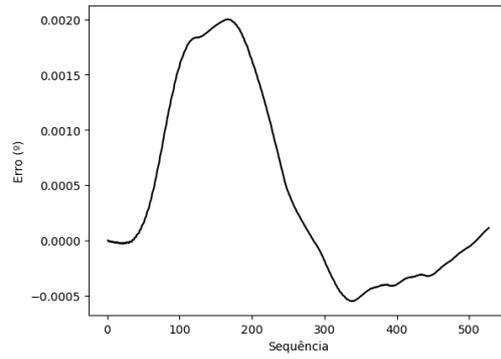
(b) Rede Neural.

Figura 5.19: Comportamento temporal do erro da latitude para o terceiro arranjo da validação cruzada na previsão de um trecho.

Comparando-se as extensões dos trechos previstos e dos trechos reais nos dois modelos, também pode-se ver que o efeito das velocidades também não foi considerado adequadamente, gerando trechos mais curtos que os reais.

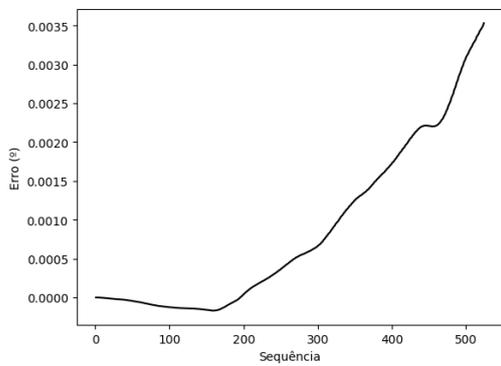


(a) Regressão Linear.

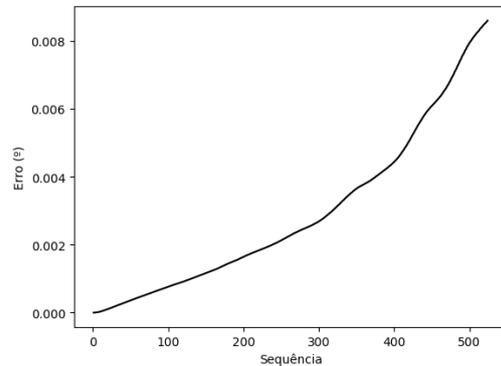


(b) Rede Neural.

Figura 5.20: Comportamento temporal do erro da latitude para o quarto arranjo da validação cruzada na previsão de um trecho.

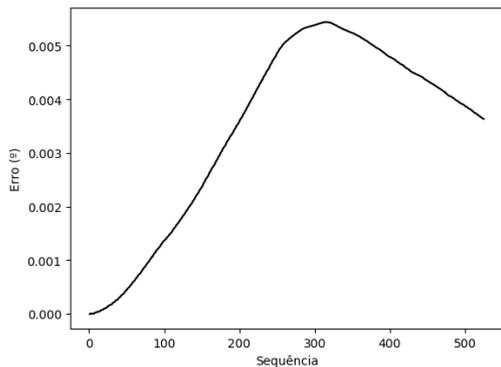


(a) Regressão Linear.

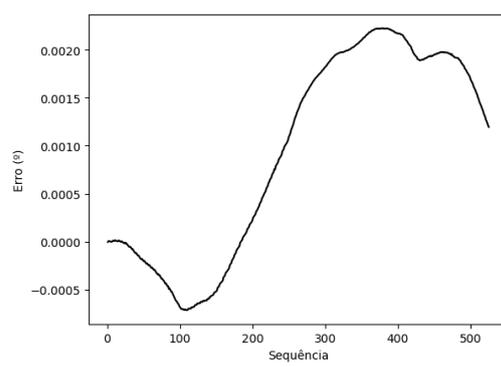


(b) Rede Neural.

Figura 5.21: Comportamento temporal do erro da longitude para o primeiro arranjo da validação cruzada na previsão de um trecho.



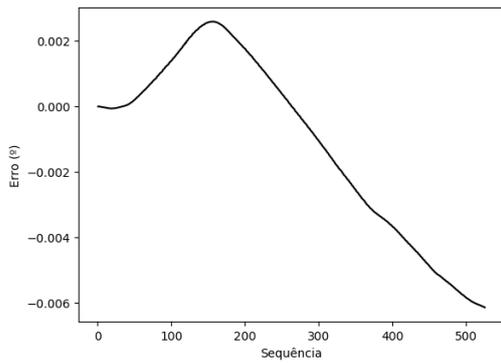
(a) Regressão Linear.



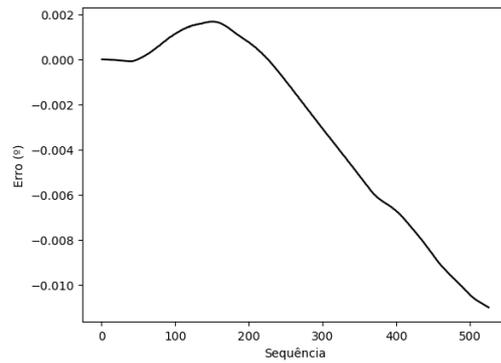
(b) Rede Neural.

Figura 5.22: Comportamento temporal do erro da longitude para o segundo arranjo da validação cruzada na previsão de um trecho.

Como visto anteriormente, no Capítulo 2, um método que se utiliza nos casos em que o erro possui um comportamento dinâmico é modelá-lo como uma série temporal. Com isto, tentou-se utilizar aqui a abordagem descrita por Belhajem [9]

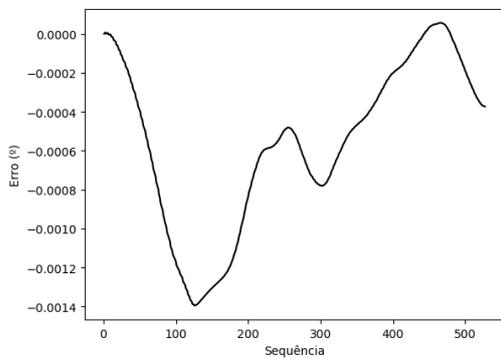


(a) Regressão Linear.

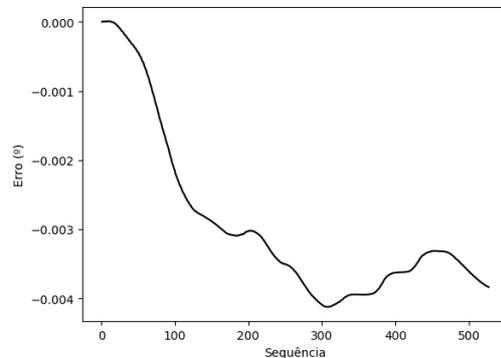


(b) Rede Neural.

Figura 5.23: Comportamento temporal do erro da longitude para o terceiro arranjo da validação cruzada na previsão de um trecho.



(a) Regressão Linear.



(b) Rede Neural.

Figura 5.24: Comportamento temporal do erro da longitude para o quarto arranjo da validação cruzada na previsão de um trecho.

e que obteve êxito no seu trabalho: elaborar um modelo corretor das posições que considera novamente a dinâmica do comportamento do carro. Tentou-se ajustar modelos lineares e Redes Neurais, usando como entradas a velocidade do veículo, as variações instantâneas das duas coordenadas e o tempo desde a última recepção de uma posição pelo GPS (um método de remover tendências crescentes, discutido no Capítulo 2), como modelos corretores. No entanto, os resultados não se reproduziram aqui. Os modelos de correção de erros, na verdade, aumentaram os erros das estimativas.

Com isto, para a predição de trechos com falta de cobertura, será necessário estudar e elaborar métodos alternativos de correção dinâmica do erro que consigam prever as informações do comportamento dinâmico do movimento de um carro que o modelo de predição de posições não pôde reproduzir. Para iniciar esta análise, o erro deve ser analisado também como uma série temporal, e as influências (séries) exógenas de interesse devem ser consideradas na adição de um segundo modelo estimador.

Dado que os desempenhos da Regressão Linear e da Rede Neural na predição de uma posição individual são equivalentes, o problema de predizer uma posição a partir de dados passados pode ser considerado um problema linear. Já o problema de predizer e corrigir os erros de predição na previsão de um trecho, dado os resultados das simulações, é possivelmente um problema não-linear.

Dado que existem trabalhos como o de Belhajem [9] e Pinto Neto [8], que obtiveram valores de erro adequados e aplicáveis em diversos contextos, uma possível modificação que pode tornar esta tarefa mais simples é a troca do sistema de coordenadas adotado. Nos dois trabalhos citados, utiliza-se coordenadas cartesianas, geradas a partir de uma conversão da latitude e da longitude, como indicadores da posição do veículo, e o ângulo de rumo ou *heading*, que corresponde ao ângulo entre a direção assumida por um objeto em movimento em relação à direção norte, para indicar a direção que ele assume em um instante. Neste trabalho, foram utilizadas como indicadores de posição as coordenadas angulares (latitude e longitude) e como indicador de direção as variações das mesmas. Uma possibilidade é que a troca do sistema de coordenadas para o adotado nestes trabalhos torne mais fácil de capturar o comportamento dinâmico. Aqui, foram utilizadas diretamente as coordenadas angulares para evitar o trabalho computacional de convertê-las para o sistema cartesiano e converter as predições de volta para as coordenadas angulares após a predição, dado que estas conversões envolvem o cálculo de várias funções trigonométricas [2].

Capítulo 6

Conclusão e Trabalhos Futuros

Este trabalho propôs um sistema experimental de localização geográfica que visa estimar a posição de um veículo na ausência de cobertura de um sistema de posicionamento por satélite, a partir de posições obtidas anteriormente pelos satélites e de sensores inerciais que não dependam de comunicações externas. Para o seu desenvolvimento, realizou-se uma análise das variáveis envolvidas segundo o referencial teórico de Séries Temporais, uma área da estatística que estuda a previsão de valores futuros e desconhecidos com informações do passado. Uma análise de acordo com os critérios de autocorrelação e autocorrelação parcial indica que a previsão de coordenadas geográficas não é possível apenas através de seus valores anteriores: a série temporal das mesmas exibe comportamentos de média móvel que refletem o comportamento dinâmico do objeto, que não é capturado através apenas destas medidas. Este comportamento deve ser capturado através dos sensores inerciais.

Com base nesta análise, candidatos a modelos lineares (aproximados pelo algoritmo de Regressão Linear) e não-lineares (aproximados por Redes Neurais Artificiais) foram escolhidos. Com estes, foram realizados dois testes: um teste de acurácia, que tinha como objetivo analisar o erro da predição de uma posição individual, e o teste da previsão de um trecho, que visava observar se o modelo capturava eficientemente a dinâmica do comportamento de um veículo. Como foi observado no Capítulo 5, os modelos lineares e não-lineares apresentam desempenhos equivalentes no teste de acurácia, o que indica que o problema de prever a posição com base em dados do passado é linear. O desempenho satisfatório, apresentando um erro médio de 2.25 metros, torna o sistema desenvolvido aqui adequado para previsões individuais de posições futuras, dado que a acurácia de um receptor GPS típico é em torno de 10 metros [2].

No caso da predição de uma trajetória, gerando um trecho percorrido por um veículo, o reuso das estimativas anteriores faz com que o erro das estimativas seja acumulado em grande escala. Este é um efeito comum da chamada operação série-paralelo, na qual se utiliza estimativas anteriores para realizar novas predições.

Tentou-se reproduzir uma estratégia de correção deste efeito, elaborada no trabalho de Belhajem [9]. No entanto, esta estratégia não obteve êxito. Isto é um indicativo de que a combinação entre o modelo de predição e o modelo de estimativas do erro não consegue capturar o comportamento dinâmico do veículo adequadamente. Por conta disto, é necessário estudar outros métodos de correção do erro. Uma possibilidade é analisar o comportamento dos erros das coordenadas como séries temporais, analisando possíveis comportamentos autorregressivos e de média móvel e dependências com variáveis exógenas que descrevam o comportamento dinâmico do veículo. Um fator que pode facilitar esta correção é a adoção de um sistema de coordenadas alternativo.

Como trabalhos futuros, serão desenvolvidos:

- Um estudo de métodos alternativos de correção do erro de predição, baseado na sua análise como uma série temporal;
- Uma análise com novos sistemas de coordenadas, para verificar se é mais fácil capturar o comportamento dinâmico utilizando outras variáveis que indiquem a localização;
- A análise de outras ferramentas de pré-processamento;
- Realizar a divisão realizada pelo K-Médias com base em outros parâmetros;
- A análise de outros algoritmos de Aprendizado de Máquina.

Apêndice A

Conceitos Básicos de Estatística e Modelos Probabilísticos

Os modelos de séries temporais aplicam os conceitos de métricas de autocorrelação e ruído branco. Para que estes conceitos sejam compreendidos, é necessário que se tenha conhecimento de outros conceitos nos quais eles se baseiam. Por isto, este apêndice faz uma resumida revisão do conteúdo necessário, conforme descrito em [25]. Uma descrição mais detalhada pode ser encontrada nesta referência.

A.1 Probabilidade

A probabilidade é uma medida da possibilidade do resultado de um evento ocorrer. Ela é uma função definida pela letra P . Seus valores estão no intervalo $[0, 1]$, no qual o valor 0 indica que tal resultado nunca ocorrerá e o valor 1 indica que aquele resultado ocorrerá sempre.

A.2 Espaço Amostral

O conjunto com todos os resultados possíveis de um experimento aleatório é chamado de espaço amostral. O espaço amostral é usualmente denotado pela letra S .

A.3 Variável Aleatória Discreta

Uma variável aleatória é uma regra de associação. Ela atribui um valor numérico para cada possível resultado de um experimento. Assim, uma variável aleatória X é uma função cujo domínio é o espaço amostral S e cuja imagem é um subconjunto dos números reais.

Ou seja, $X : S \rightarrow \mathbb{R}$, e atribui um valor de \mathbb{R} para qualquer ponto s do espaço amostral S .

A.3.1 Densidade de Probabilidade

Seja A um evento aleatório. Seja um evento A_x , definido a partir de A , que corresponde ao subconjunto do espaço amostral S com pontos $s|X(s) = x$, onde x é um valor arbitrário pertencente a \mathbb{R} . Se X for uma variável discreta ($x \in \mathbb{Z}$), a probabilidade de A_x pode ser expressa como:

$$P(A_x) = P(|X = x|) \rightarrow P(A_x) = P(s|X(s) = x) \rightarrow P(A_x) = \sum_{X(s)=x} P(s). \quad (\text{A.1})$$

Assim, define-se a função de densidade de probabilidade da variável aleatória X , também denotada por $p_x(x)$. Portanto,

$$p_x(x) = P(|X = x|). \quad (\text{A.2})$$

Esta função tem as seguintes propriedades:

- $0 \leq p_x(x) \leq 1$, para qualquer $x \in \mathbb{R}$, dado que $p_x(x)$ é uma probabilidade.
- Dado que a variável aleatória X atribui um valor $x \in \mathbb{R}$ para cada ponto do espaço amostral $s \in S$, deve-se ter que

$$\sum_{x \in \mathbb{R}} p_x(x) = 1. \quad (\text{A.3})$$

- Para uma variável discreta e aleatória X , o conjunto $x|p_x(x) \neq 0$ é um conjunto finito ou infinito e contável dos números reais, denotado como x_1, x_2, \dots , o que permite a reescrita da propriedade anterior como:

$$\sum_i p_x(x_i) = 1. \quad (\text{A.4})$$

A.3.2 Distribuição de Probabilidade

A distribuição de probabilidade ou função de distribuição cumulativa pode ser definida como:

$$F_X(t) = P(-\infty < X \leq t) = P(x \leq t) = \sum_{x \leq t} p_x(x). \quad (\text{A.5})$$

Esta função permite analisar probabilidades de X assumir um valor dentro de um intervalo.

A.4 Variável Aleatória Contínua

Assim como nas variáveis aleatórias discretas, pode-se definir uma variável aleatória X como uma função cujo domínio é o espaço amostral S e cuja imagem é um subconjunto dos números reais. No entanto, neste caso, o espaço amostral é um conjunto não-enumerável, de modo que nem todo subconjunto do mesmo é um evento que pode ser associado a uma probabilidade.

Para qualquer valor real de x , um evento é representado pelo conjunto de pontos do espaço amostral s tais que $X(s) \leq x$, de modo que pode-se atribuir uma probabilidade a este conjunto.

Assim, conseguimos definir a função de distribuição de probabilidade.

A.4.1 Distribuição de Probabilidade

A função de distribuição (cumulativa) ou CDF (*Cumulative Distribution Function*) F_X de uma variável aleatória X é definida como:

$$F_X(x) = P(X \leq x), -\infty < x < \infty. \quad (\text{A.6})$$

Quando X é a única variável aleatória em consideração, podemos simplesmente indicar o símbolo F para a CDF.

Esta função tem como propriedades:

- $0 \leq F(x) \leq 1, -\infty < x < \infty.$
- $F(x)$ é monótona crescente.
- $\lim_{x \rightarrow -\infty} F(x) = 0$ e $\lim_{x \rightarrow \infty} F(x) = 1.$

A.4.2 Densidade de Probabilidade

Para uma variável aleatória X , $f(x) = dF(x)/dx$ é chamada de densidade de probabilidade (ou pdf - do inglês *Probability Density Function* de X).

Esta função permite obter a distribuição de probabilidade por meio da integração:

$$F_X(x) = P(X \leq x) = \int_{-\infty}^x f_X(t) dt, \quad -\infty < x < \infty. \quad (\text{A.7})$$

A densidade de probabilidade possui, na prática, um valor muito baixo para um valor individual x , de modo que se considera que o valor de $f_X(t)$ seja aproximadamente 0.

A.5 Distribuição Normal ou Gaussiana

A densidade de probabilidade normal é dada por:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \quad -\infty < x < \infty. \quad (\text{A.8})$$

Os parâmetros μ e σ correspondem, respectivamente, à média e ao desvio-padrão dos valores de uma variável com esta distribuição.

Se uma variável aleatória X segue esta distribuição, podemos indicar desta forma: $X \sim N(\mu, \sigma^2)$.

A.6 Valor Esperado

O valor esperado ou esperança de uma variável aleatória X é definido por:

$$E[X] = \sum_i x_i p(x_i), \quad (\text{A.9})$$

caso X seja discreta, ou:

$$E[X] = \int_{-\infty}^{+\infty} x f(x) dx, \quad (\text{A.10})$$

caso X seja contínua.

A.6.1 Propriedades

Abaixo seguem algumas propriedades de interesse:

- A operação de valor esperado é linear, de modo que $E[X + Y] = E[X] + E[Y]$ e $E[cX] = cE[X]$, sendo X e Y duas variáveis aleatórias quaisquer e c uma constante escalar.
- Dado que uma constante tem um valor fixo, $E[c] = c$, sendo c novamente uma constante escalar.

A.7 Momentos

Também pode ser interessante calcular o valor esperado de uma função de X . Para realizar este cálculo, consideremos $Y = \phi(X)$ uma outra variável aleatória, cujo valor é uma função do valor de X . Pode-se, assim, dar o valor esperado de Y como:

$$E[Y] = \sum_i \phi(x_i)p_X(x_i), \quad (\text{A.11})$$

caso X seja discreta, ou:

$$E[Y] = \int_{-\infty}^{+\infty} \phi(x)f_X(x)dx, \quad (\text{A.12})$$

caso X seja contínua.

Um caso especial é a função potência de X . Para potências $k = 1, 2, 3, \dots$, $E[X^k]$ é chamado de k -ésimo momento de X .

Outro caso de interesse são valores esperados de potências $X - E[x]$, que são medidas de dispersão de X em relação à sua média. Para potências $k = 1, 2, 3, \dots$, $E[(X - E[x])^k]$ é chamado de k -ésimo momento central de X .

Abaixo seguem os momentos de interesse para este trabalho.

A.7.1 Média

A média de X é simplesmente o valor esperado de X , ou o seu primeiro momento. Assim, podemos denotar a média de X como:

$$\mu_x = E[X]. \quad (\text{A.13})$$

A.7.2 Variância e Desvio Padrão

A variância é o segundo momento central, sendo definida por:

$$\sigma_x^2 = E[(X - E[X])^2]. \quad (\text{A.14})$$

O desvio padrão é a raiz quadrada deste valor:

$$\sigma_x = \sqrt{E[(X - E[X])^2]}. \quad (\text{A.15})$$

A.8 Independência Estatística

Dois eventos quaisquer A e B são independentes se a execução de um não afeta a execução de outro. Formalmente, isto ocorre se $P(A \cap B) = P(A)P(B)$, sendo

$P(A \cap B)$ a probabilidade conjunta, ou seja, a probabilidade de os dois eventos ocorrerem.

Esta propriedade é de interesse neste trabalho devido à sua consequência para duas variáveis aleatórias e independentes X e Y :

$$E[XY] = E[X]E[Y]. \quad (\text{A.16})$$

A.9 Intervalo de Confiança

O intervalo de confiança é uma estimativa do intervalo dentro do qual o verdadeiro valor de um parâmetro relacionado a uma população se encontra, calculado a partir de uma amostra de dados desta população. Este intervalo pode ser calculado com um nível de confiança variável, tornando a confiança da estimativa ajustável. Um nível de confiança usualmente adotado é 95%.

Seja um parâmetro populacional θ cujo valor se deseja estimar através de um estimador $\hat{\theta}$. Se $\hat{\theta}$ satisfizer a condição:

$$P(\hat{\theta} - \epsilon_1 < \theta < \hat{\theta} + \epsilon_2) = \gamma, \quad (\text{A.17})$$

pode-se afirmar que o intervalo aleatório $A(\theta) = (\hat{\theta} - \epsilon_1, \hat{\theta} + \epsilon_2)$ é um intervalo de confiança de $100 \times \gamma\%$, sendo γ chamado de coeficiente de confiança. Costuma-se utilizar intervalos de confiança simétricos, de modo que $\epsilon_1 = \epsilon_2 = \epsilon$.

Uma das maneiras de se obter o intervalo de confiança é através da inequação de Chebyshev, dado que o estimador tenha uma variância conhecida ou que possa ser estimada:

$$\gamma = P(\hat{\theta} - \epsilon_1 < \theta < \hat{\theta} + \epsilon_2) \geq 1 - \frac{Var[\hat{\theta}]}{\epsilon^2}. \quad (\text{A.18})$$

Outra maneira é possível tendo-se conhecimento da distribuição de probabilidade que as estimativas do parâmetro desejado seguem: realizar um processo iterativo, variando-se em pequenos intervalos os valores dos limites do intervalo, e calculando-se a probabilidade cumulativa (CDF) para o valor do estimador do parâmetro dentro do intervalo, até que seja encontrado o nível de confiança desejado.

Apêndice B

Provas das Propriedades dos Modelos MA e AR

B.1 Autocorrelação para modelos MA

Seja um modelo MA de ordem arbitrária q , ou $MA(q)$:

$$\hat{x}(t) = \omega + \epsilon(t) + \sum_{i=1}^q a_i \epsilon(t-i). \quad (\text{B.1})$$

Como ω corresponde à média da série temporal, o modelo pode ser reescrito da seguinte forma:

$$\hat{x}(t) = \mu_x + \epsilon(t) + \sum_{i=1}^q a_i \epsilon(t-i). \quad (\text{B.2})$$

Como mencionado anteriormente, a autocorrelação pode ser expressa como:

$$r(x, k) = \frac{AC_x(k)}{\sigma_x^2}. \quad (\text{B.3})$$

Primeiramente, será desenvolvida a expressão da autocovariância. De acordo com a expressão 2.5 e com a definição de valor esperado, pode-se expressar a autocovariância como:

$$AC_x(k) = E[(x(t) - \mu_x)(x(t+k) - \mu_x)].$$

Desenvolvendo o produto, tem-se que:

$$AC_x(k) = E[x(t)x(t+k) - x(t)\mu_x - x(t+k)\mu_x + \mu_x^2].$$

Usando as propriedades do valor esperado, obtém-se:

$$AC_x(k) = E[x(t)x(t+k)] - E[x(t)\mu_x] - E[x(t+k)\mu_x] + E[\mu_x^2].$$

A média de x , μ_x , é uma constante, logo:

$$AC_x(k) = E[x(t)x(t+k)] - \mu_x E[x(t)] - \mu_x E[x(t+k)] + \mu_x^2.$$

Por fim, o valor esperado de x , em qualquer instante de tempo, é igual à sua média, μ_x . Assim, é encontrada a expressão final para a autocovariância:

$$AC_x(k) = E[x(t)x(t+k)] - \mu_x^2. \quad (\text{B.4})$$

Será calculado, agora, o valor esperado:

$$E[x(t)x(t+k)] = E[(\mu_x + \epsilon(t) + \sum_{i=1}^q a_i \epsilon(t-i))(\mu_x + \epsilon(t+k) + \sum_{i=1}^q a_i \epsilon(t+k-i))].$$

Desenvolvendo este produto, resulta-se em:

$$\begin{aligned} E[x(t)x(t+k)] = & E[\mu_x^2 + \mu_x \epsilon(t+k) + \mu_x \sum_{i=1}^q a_i \epsilon(t+k-i) + \mu_x \epsilon(t) + \epsilon(t)\epsilon(t+k) \\ & + \epsilon(t) \sum_{i=1}^q a_i \epsilon(t+k-i) + \mu_x \sum_{i=1}^q a_i \epsilon(t-i) \\ & + \epsilon(t+k) \sum_{i=1}^q a_i \epsilon(t-i) + \sum_{i=1}^q a_i \epsilon(t-i) \sum_{i=1}^q a_i \epsilon(t+k-i)]. \end{aligned} \quad (\text{B.5})$$

Usando a propriedade de valor esperado da soma, isto se torna:

$$\begin{aligned}
E[x(t)x(t+k)] &= E[\mu_x^2] + E[\mu_x\epsilon(t+k)] + E[\mu_x \sum_{i=1}^q a_i\epsilon(t+k-i)] + E[\mu_x\epsilon(t)] \\
&\quad + E[\epsilon(t)\epsilon(t+k)] + E[\epsilon(t) \sum_{i=1}^q a_i\epsilon(t+k-i)] \\
&\quad + E[\mu_x \sum_{i=1}^q a_i\epsilon(t-i)] + E[\epsilon(t+k) \sum_{i=1}^q a_i\epsilon(t-i)] \\
&\quad + E[\sum_{i=1}^q a_i\epsilon(t-i) \sum_{i=1}^q a_i\epsilon(t+k-i)]. \tag{B.6}
\end{aligned}$$

Dado que μ_x é uma constante, isto se reduz a:

$$\begin{aligned}
E[x(t)x(t+k)] &= \mu_x^2 + \mu_x E[\epsilon(t+k)] + \mu_x E[\sum_{i=1}^q a_i\epsilon(t+k-i)] + \mu_x E[\epsilon(t)] \\
&\quad + E[\epsilon(t)\epsilon(t+k)] + E[\epsilon(t) \sum_{i=1}^q a_i\epsilon(t+k-i)] \\
&\quad + \mu_x E[\sum_{i=1}^q a_i\epsilon(t-i)] + E[\epsilon(t+k) \sum_{i=1}^q a_i\epsilon(t-i)] \\
&\quad + E[\sum_{i=1}^q a_i\epsilon(t-i) \sum_{i=1}^q a_i\epsilon(t+k-i)]. \tag{B.7}
\end{aligned}$$

Dado que o modelo MA considera que os valores do ruído ϵ possuem média nula e são independentes entre si, $E[\epsilon(t)] = 0$ e $E[\epsilon(t)\epsilon(t-k)] = E[\epsilon(t)]E[\epsilon(t-k)] = 0$ para quaisquer valores inteiros de t e k , sendo $k \neq t$, o que reduz a equação a:

$$E[x(t)x(t+k)] = \mu_x^2 + E[\sum_{i=1}^q a_i\epsilon(t-i) \sum_{i=1}^q a_i\epsilon(t+k-i)].$$

Pode-se rearranjar o primeiro somatório como a seguir:

$$E[x(t)x(t+k)] = \mu_x^2 + E[\sum_{i=k+1}^{q+k} a_{i-k}\epsilon(t+k-i) \sum_{i=1}^q a_i\epsilon(t+k-i)].$$

Deste produto de somatórios, restam apenas os produtos de valores de ϵ de mesmo atraso, devido à independência entre instantes e ao fato de a média ser zero. A expressão pode ser, então, representada por:

$$E[x(t)x(t+k)] = \mu_x^2 + E[\sum_{i=k+1}^q a_{i-k}a_i\epsilon^2(t+k-i)].$$

Porque $E[\epsilon(t)] = 0$, $E[\epsilon^2(i)] = \sigma_\epsilon^2$, a variância do ruído, o que leva à expressão final:

$$E[x(t)x(t+k)] = \mu_x^2 + \sigma_\epsilon^2 \sum_{i=k+1}^q a_{i-k}a_i.$$

Com isto, a autocovariância com k atrasos do modelo MA pode ser expressa como:

$$AC_x(k) = \sigma_\epsilon^2 \sum_{i=k+1}^q a_{i-k}a_i.$$

E, por fim, encontra-se que a autocorrelação é igual a:

$$r(x, k) = \frac{\sigma_\epsilon^2 \sum_{i=k+1}^q a_{i-k}a_i}{\sigma_x^2}. \quad (\text{B.8})$$

B.2 Autocorrelação para modelos AR

Seja um modelo AR de ordem arbitrária q , ou $AR(q)$:

$$x(t) = \omega + \sum_{i=1}^q a_i x(t-i) + \epsilon(t). \quad (\text{B.9})$$

Reutilizando a equação encontrada anteriormente para a autocorrelação:

$$r(x, k) = \frac{E[x(t)x(t+k)] - \mu_x^2}{\sigma_x^2}.$$

Para determinar a autocorrelação para qualquer atraso, será calculada a autocorrelação para um atraso arbitrário k para modelos AR de diferentes ordens.

Começando pelo modelo AR(1):

$$x(t) = \omega + a_1 x(t-1) + \epsilon(t). \quad (\text{B.10})$$

Multiplica-se esta equação por $x(t-k)$:

$$x(t)x(t-k) = \omega x(t-k) + a_1 x(t-1)x(t-k) + \epsilon(t)x(t-k). \quad (\text{B.11})$$

Extraindo o valor esperado dos dois lados da equação:

$$E[x(t)x(t-k)] = \omega E[x(t-k)] + a_1 E[x(t-1)x(t-k)] + E[\epsilon(t)x(t-k)]. \quad (\text{B.12})$$

Como $\epsilon(t)$ somente influencia em valores em instantes posteriores a k , segue que:

$$E[x(t)x(t-k)] = \omega E[x(t-k)] + a_1 E[x(t-1)x(t-k)]. \quad (\text{B.13})$$

Devido à estacionaridade de x :

$$E[x(t)x(t-k)] = \omega \mu_x + a_1 E[x(t-1)x(t-k)]. =, \quad (\text{B.14})$$

onde μ_x é a média de x .

Os valores esperados remanescentes podem ser expressos em termos de autocorrelações:

$$r(x, k)\sigma_x^2 + \mu_x^2 = \omega \mu_x + a_1 r(x, k-1)\sigma_x^2 + a_1 \mu_x^2.$$

Simplificando, obtem-se:

$$r(x, k) = \frac{\omega \mu_x + \mu_x^2(a_1 - 1)}{\sigma_x^2} + a_1 r(x, k-1). \quad (\text{B.15})$$

Realizando o mesmo processo para o AR(2), obtem-se que:

$$x(t) = \omega + a_1 x(t-1) + a_2 x(t-2) + \epsilon(t).$$

Multiplicando a equação por $x(t-k)$:

$$x(t)x(t-k) = \omega x(t-k) + a_1 x(t-1)x(t-k) + a_2 x(t-2)x(t-k) + \epsilon(t)x(t-k).$$

Extraindo novamente o valor esperado da equação:

$$E[x(t)x(t-k)] = \omega E[x(t-k)] + a_1 E[x(t-1)x(t-k)] + a_2 E[x(t-2)x(t-k)] + E[\epsilon(t)x(t-k)]. \quad (\text{B.16})$$

Com passos similares aos anteriores, chega-se ao resultado:

$$r(x, k)\sigma_x^2 + \mu_x^2 = \omega\mu_x + a_1r(x, k-1)\sigma_x^2 + a_1\mu_x^2 + a_2r(x, k-2)\sigma_x^2 + a_2\mu_x^2.$$

$$r(x, k) = \frac{\omega\mu_x + \mu_x^2(a_1 + a_2 - 1)}{\sigma_x^2} + a_1r(x, k-1) + a_2r(x, k-2). \quad (\text{B.17})$$

Pode-se ver uma recorrência entre as equações B.15 e B.17, o que permite elaborar a equação do termo geral:

$$r(x, k) = \frac{\omega\mu_x + \mu_x^2[(\sum_{i=1}^q a_i) - 1]}{\sigma_x^2} + \sum_{i=1}^q a_i r(x, k-i). \quad (\text{B.18})$$

Pode-se expressar este resultado de outra maneira calculando-se a média da série temporal. Partindo de um modelo arbitrário:

$$x(t) = \omega + \sum_{i=1}^q a_i x(t-i) + \epsilon(t).$$

Dado que x é estacionário, a média pode ser calculada através de $E[x(t)]$. Além disto, os erros de predição ϵ têm média nula, permitindo o seguinte desenvolvimento:

$$E[x(t)] = E[\omega] + E[\sum_{i=1}^q a_i x(t-i)] + E[\epsilon(t)];$$

$$\mu_x = \omega + E[\sum_{i=1}^q a_i x(t-i)];$$

$$\mu_x = \omega + \sum_{i=1}^q a_i E[x(t-i)];$$

$$\mu_x = \omega + \sum_{i=1}^q a_i \mu_x;$$

$$\mu_x \left(1 - \sum_{i=1}^q a_i\right) = \omega;$$

$$\mu_x = \frac{\omega}{1 - \sum_{i=1}^q a_i}.$$

Retomando da equação B.18:

$$r(x, k) = \frac{\frac{\omega^2}{1 - \sum_{i=1}^q a_i} - \frac{\omega^2}{1 - \sum_{i=1}^q a_i}}{\sigma_x^2} + \sum_{i=1}^q a_i r(x, k - i).$$

E encontra-se o resultado final de:

$$r(x, k) = \sum_{i=1}^q a_i r(x, k - i). \quad (\text{B.19})$$

B.3 Valores dos Coeficientes de um Modelo AR

Partindo novamente de um modelo de ordem arbitrária e aplicando a operação de variância:

$$Var[x(t)] = Var[\omega] + Var\left[\sum_{i=1}^q a_i x(t - i)\right] + Var[\epsilon(t)].$$

A variância de uma constante é nula e os erros têm uma variancia fixa σ_ϵ^2 :

$$Var[x(t)] = Var\left[\sum_{i=1}^q a_i x(t - i)\right] + \sigma_\epsilon^2.$$

Usando as propriedades da variância, tem-se que:

$$Var[x(t)] = \sum_{i=1}^q a_i^2 Var[x(t - i)] + \sigma_\epsilon^2.$$

Seguindo a suposição de que x seja estacionário, $Var[x(t)] = Var[x(t - i)]$ para qualquer valor de i . Vem disto que:

$$Var[x(t)] = \sum_{i=1}^q a_i^2 Var[x(t)] + \sigma_\epsilon^2;$$

$$Var[x(t)](1 - \sum_{i=1}^q a_i^2) = \sigma_\epsilon^2;$$

$$Var[x(t)] = \frac{\sigma_\epsilon^2}{(1 - \sum_{i=1}^q a_i^2)}. \quad (\text{B.20})$$

Dado que a variância é um valor positivo:

$$1 - \sum_{i=1}^q a_i^2 > 0 \longrightarrow |a_i| < 1. \quad (\text{B.21})$$

Referências Bibliográficas

- [1] NETO, J. B. P., GOMES, L. C., CASTANHO, E. M., et al. “An Error Correction Algorithm for Forward Collision Warning Applications”. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1926–1931, Nov 2016. doi: 10.1109/ITSC.2016.7795867.
- [2] ZOGG, J. *GPS: Essentials of Satellite Navigation : Compendium*. U-blox AG, 2009. ISBN: 9783033021396.
- [3] HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J. *The Elements of Statistical Learning*. Springer, 2009. ISBN: 978-0387848570.
- [4] “Página do Projeto Waze Beacons”. . <https://www.waze.com/pt-BR/beacons>, 2018. Acessado em setembro de 2018.
- [5] “Página da Ajuda aos Parceiros do Waze no Suporte da Google (em inglês)”. . <https://support.google.com/waze/partners/answer/7298984?hl=en>, 2018. Acessado em setembro de 2018.
- [6] “Bluvision Selected as Waze Beacons Partner to Power Tunnel Navigation”. . <http://bluvision.com/blog/2016/09/21/bluvision-beacons-power-tunnel-navigation-in-googles-waze-app/>, 2016. Acessado em setembro de 2018.
- [7] MASCHARKA, D., MANLEY, E. “Machine Learning for Indoor Localization Using Mobile Phone-Based Sensors”. 2015.
- [8] NETO, J. B. P., MITTON, N., CAMPISTA, M. E. M., et al. “Dead Reckoning Using Time Series Regression Models”. In: *Nineteenth International Symposium on Mobile Ad Hoc Networking and Computing - MobiHoc’18*, June 2018.
- [9] BELHAJEM, I., MAISSA, Y. B., TAMTAOUI, A. “A robust low cost approach for real time car positioning in a smart city using Extended Kalman Filter and Evolutionary Machine Learning”. In: *4th IEEE International Colloquium on Information Science and Technology (CiSt)*, 2016.

- [10] SHUMWAY, R., STOFFER, D. *Time Series Analysis and its Applications*. Springer, 2011. ISBN: 978-1-4419-7864-6.
- [11] HYNDMAN, R., ATHANASOPOULOS, G. *Forecasting: Principles and Practice*. OTexts, 2018. ISBN: 978-0987507112.
- [12] SEABOLD, S., PERKTOLD, J. “Statsmodels: Econometric and statistical modeling with python”. In: *9th Python in Science Conference*, 2010.
- [13] WATTERSON, C. “AN-1123 - Controller Area Network (CAN) Implementation Guide”. . Analog Systems (<http://www.analog.com/>), 2017. Acessado em setembro de 2018.
- [14] DAISS, A., KIENCKE, U. “Estimation of vehicle speed fuzzy-estimation in comparison with Kalman-filtering”. In: *Proceedings of International Conference on Control Applications*, 1995.
- [15] JONES, E., OLIPHANT, T., PETERSON, P., et al. “SciPy: Open source scientific tools for Python”. . <http://www.scipy.org/>, 2001. Acessado em setembro de 2018.
- [16] HUNTER, J. D. “Matplotlib: A 2D graphics environment”, *Computing In Science & Engineering*, v. 9, n. 3, pp. 90–95, 2007. doi: 10.1109/MCSE.2007.55.
- [17] HAYKIN, S. *Neural Networks and Learning Machines*. Pearson, 2009. ISBN: 978-0-13-147139-9.
- [18] SILVA, I., SPATTI, D., FLAUZINO, R. *Redes Neurais Artificiais para Engenharia e Ciências Aplicadas*. ArtLiber, 2010. ISBN: 978-8588098534.
- [19] CYBENKO, G. “Approximation by Superpositions of a Sigmoidal Function”. In: *Mathematics of Control, Signals, and Systems*, pp. 303–314, 1989.
- [20] HORNIK, K., STINCHCOMBE, M., WHITE, H. “Multilayer feedforward networks are universal approximators”. In: *Neural Networks, Volume 2, Issue 5*, pp. 359–366, 1989.
- [21] ARTHUR, D., VASSILVITSKII, S. “k-means++: The Advantages of Careful Seeding”. In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics*, 2007.

- [22] IVIS, F. “Calculating Geographic Distance: Concepts and Methods”. . <https://www.lexjansen.com/nesug/nesug06/dm/da15.pdf>, 2006. Acessado em setembro de 2018.
- [23] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., et al. “Scikit-learn: Machine Learning in Python”, *Journal of Machine Learning Research*, v. 12, pp. 2825–2830, 2011.
- [24] CALÔBA, L. P. “Introdução ao Uso de Redes Neurais na Modelagem de Sistemas Dinâmicos e Séries Temporais”. In: *Livro de Minicursos do XIV Congresso Brasileiro de Automática, 2002*, 2002.
- [25] TRIVEDI, K. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. Wiley, 2002. ISBN: 0-471-33341-7.