



UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

Previsão de Trajetórias de Veículos Aéreos pela Abordagem da Máquina de Suporte Vetorial

Simão Mendes Lopes

Dissertação para obtenção do Grau de Mestre em
Engenharia Aeronáutica
(Ciclo de estudos integrado)

Orientador: Prof. Doutor Kouamana Bousson

Covilhã, janeiro de 2016

Agradecimentos

Em primeiro lugar, gostaria de deixar uma grande nota de agradecimento ao meu professor e orientador, Prof. Doutor Kouamana Bousson. Desde o início que sempre se destacou por querer envolver os alunos num ambiente mais prático nas suas aulas, apresentando histórias de contexto real e destacando-se pela metodologia de ensino em que coloca os alunos a trabalhar e a programar desde a primeira aula. Mas foi neste período de orientação e cooperação no desenvolvimento da dissertação que consegui aperceber-me com maior detalhe a enorme extensão dos seus conhecimentos. Não só consegue transmitir ao aluno a maneira como vários elementos diferentes do dia-a-dia estão interligados, como o faz de uma maneira motivadora e brilhante. Por me ter mostrado as enormes possibilidades do mundo Aeronáutico e da Engenharia em si, por ter acreditado em mim e nunca ter duvidado das minhas capacidades, e me ter acompanhado nesta jornada, um muito obrigado.

Uma palavra de apreço também por todo o restante corpo docente e não-docente do Departamento de Ciências Aeroespaciais da Universidade da Beira Interior, que todos os dias contribuem para que os alunos se sintam cada vez mais em casa, dentro e fora das salas de aula.

A toda a minha família, e em especial, à minha mãe e ao meu pai, por todo o carinho, amor e paciência deste Mundo, e por todo o apoio emocional que prestaram ao longo de toda esta jornada e de toda a minha vida, sem nunca terem duvidado das minhas capacidades e das minhas motivações. A eles, sem dúvida, o maior e mais sentido agradecimento.

A todos os meus colegas de curso, em particular, António Moreira, Carlos Vieira, Diogo Martins e Henrique Mendes pelos trabalhos, aventuras e momentos inesquecíveis ao longo desta jornada académica que certamente durarão para o resto dos dias.

Por fim, um carinho muito especial aos meus amigos de longa data, em especial, António Marques, Carolina Lobo, Carolina Teodózio, Gonçalo Pinto, João Devezas, João Diogo Matos, João Duarte, Laurie-Anne Oliveira, Leonor Gama, Luís Ferreira, Maria Arrifano, Marta Jacob, Rita Arcão e Rita Santos. Todos eles desempenharam um papel importantíssimo ao longo do meu crescimento, não só durante a minha vida académica, mas desde sempre. A todos eles, muito obrigado.

Resumo

O presente trabalho visa conciliar a temática das redes neuronais artificiais à previsão de trajetórias de aeronaves. Especificamente, centra-se à volta da aplicação de métodos de regressão através de algoritmos de Máquina de Suporte Vetorial com base nos Mínimos Quadrados (ou LS-SVM *Least-Square Support Vector Machines*) na previsão de trajetórias de quatro dimensões de uma aeronave comercial.

Inicialmente, é feito um enquadramento da temática de aprendizagem automática e redes neuronais através de uma contextualização teórica, progredindo para um desenvolvimento de algumas funções aplicadas no tema das redes neuronais artificiais. Dentro destas funções, destacam-se o Método Kernel e as Funções de Bases Radiais (ou RBF - *Radial Basis Function*), aplicadas também na obtenção de resultados práticos. Na segunda parte deste capítulo, são desenvolvidos os temas de LS-SVM, projeto SESAR (responsável pela implementação de trajetórias 4D no céu europeu), e um dos métodos de aquisição de dados de trajetória em aeronaves comerciais, o *Automatic Dependent Surveillance-Broadcast* (ADS-B). Toda esta explicação teórica, dá lugar à apresentação dos conteúdos formais e matemáticos por detrás dos métodos implementados, tais como: Condições de Otimalidade, Geração de Erro Gaussiano, Funções Kernel aplicadas, bem como uma explicação de como a Localização Aeroespacial decorre numa aeronave comercial.

Por fim, os dados foram simulados e validados num contexto de Previsão de Trajetória, permitindo assim obter, no final, uma função que preveja a trajetória da aeronave em todo o seu percurso. Para isso, foram utilizadas três trajetórias distintas, criadas através da implementação do Algoritmo de Simulação Numérica RK-Butcher, bem como um ambiente de navegação de quatro dimensões, em que a aeronave define a sua trajetória inicialmente, através de um conjunto de *waypoints* definidos anteriormente. No final, é possível observar não só o sucesso da implementação desta metodologia, bem como fazer um estudo comparativo entre os diferentes métodos utilizados.

Após uma reflexão sobre os resultados obtidos, e analisando o conteúdo de todo o trabalho através de vários prismas, é possível fazer uma apreciação bastante positiva do projeto desenvolvido. Não só os resultados foram de acordo com o pretendido, tendo sido obtidas todas as funções de previsão com sucesso, mas também foi possível obter uma visão geral acerca da vasta aplicação que estes métodos detêm, não só na área da Engenharia Aeronáutica, mas como em qualquer setor de investigação científica.

Palavras-chave

Inteligência artificial; Previsão de trajetórias; Trajetórias 4D; Navegação por Waypoints; Aprendizagem automática; Aprendizagem semi-supervisionada; Redes neuronais; Funções de Bases Radiais; Algoritmos de Máquina de Suporte Vetorial com base nos Mínimos Quadrados.

Abstract

The present work aims to reconcile the neural networks subject, with the aircraft trajectory prediction theme. Specifically, it revolves around the application of Least-Squares Support Vector Machines methods in the prediction of 4-dimensional trajectories of commercial aircrafts.

Initially, this work explains the guidelines regarding the machine learning and neural networks subjects through a theoretical explanation, gradually progressing to a more detailed elucidation of some of the functions applied in artificial neural networks. Within these functions, we highlight the Kernel Method and Radial Basis Functions, which were essential in order to obtain the practical results. In the last part of this chapter, the LS-SVM topic is approached theoretically, as well as the SESAR Project, which is responsible for the implementation of 4D trajectories in the European sky and the main trajectory data acquisition method in commercial aircraft, the Automatic Dependent Surveillance-Broadcast (ADS-B). All this theoretical explanation gives rise to the enunciation of the formal and mathematical content of this project, such as: Optimality Conditions, generation of Gaussian Error, Kernel Functions applied, as well as an explanation of how the Aerospace Localization methods work in a commercial aircraft.

Lastly, all the data was simulated and validated in a context of Trajectory Prediction, allowing the algorithm to build a function that could predict the trajectory of the aircraft in its complete course. In order to achieve that, three different trajectories were used, created through the implementation of the Numeric Simulation Algorithm of RK-Butcher, as well as an environment of 4D trajectory navigation, in which the aircraft defines its trajectory through the definition of previously mentioned waypoints. In the end, it is possible to observe not only the success of the implementation of this method, but to make a comparative study between the different parameters used as well.

After some reflection on the results obtained, and analyzing the content of this work through different states of mind, it is possible to make a very positive appreciation of the project that was developed. Not only the results matched the purposes defined in the beginning, but it also made possible to obtain a wider overview about the vast applications that these methods possess, not only in the field of Aeronautical Engineering, but in all the subjects of scientific investigation.

Keywords

Artificial intelligence; Trajectory prediction; 4D Trajectories; Waypoint navigation; Machine learning; Semi-supervised learning; Neural networks; Radial Basis Functions; Least-Square Support Vector Regression.

Índice

Capítulo 1	1
1. Introdução	1
1.1. Reconhecimento de Padrões e Filtragem de Dados	2
1.2. Aprendizagem Automática	2
1.2.1. Problemas Principais Abordados pela Aprendizagem Automática	3
1.3. Aprendizagem Supervisionada	4
1.4. Redes Neurais Clássicas.....	4
1.4.1. Topologia das Redes Neurais	6
1.5. Método Kernel.....	7
1.6. Funções de Bases Radiais.....	8
1.6.1. Algoritmo de Aprendizagem	10
1.6.2. Efeito de Sigma	10
1.7. Relação com Redes Neurais.....	11
1.7.1. Comparação com Redes Neurais Clássicas	12
1.8. Máquina de Suporte Vetorial com Base nos Mínimos Quadrados	13
1.9. SESAR e a Navegação por Waypoints 4D	14
1.10. Automatic Dependent Surveillance-Broadcast	15
1.11. Algumas Aplicações do Método LS-SVM	16
1.12. Objetivos do Trabalho Proposto	17
1.13. Estrutura da Dissertação	18
Capítulo 2	20
2. Princípios da Aprendizagem Automática com Base na Máquina de Suporte Vetorial	20
2.1. Condições de Otimalidade	20
2.1.1. Otimização Não Linear	20
2.2. Método da Regressão através da Máquina de Suporte Vetorial com base nos Mínimos Quadrados.....	23
2.2.1. Lista de Funções Kernel Aplicadas.....	27
2.2.2. Implementação de Erro Gaussiano (Distribuição Normal)	28
2.3. Localização Aeroespacial.....	29

2.3.1.	Sistema de Navegação por Satélite - GNSS	30
2.3.2.	Funcionamento do Sistema ADS-B	31
2.3.3.	Sistemas de Navegação Inercial (INS)	32
2.3.4.	Rumo de uma Aeronave (ψ)	34
2.3.5.	Ângulo de Trajetória de uma Aeronave (γ)	34
2.3.6.	Velocidade da Aeronave.....	35
3.	Simulação e Validação em Previsão de Trajetórias.....	37
3.1.	Exemplo de Previsão de Funções	37
3.2.	Estrutura do Programa Implementado	39
3.3.	Geração de Dados da Trajetória	40
3.3.1.	Implementação do Algoritmo de Simulação Numérica RK-Butcher	41
3.3.2.	Cálculo da Distância $dP_{k+1}(X)$	42
3.3.3.	Cálculo dos Tempos de Chegada aos <i>Waypoints</i>	43
3.3.4.	Loxodromia	43
3.3.5.	Comprimento de um Arco de Loxodromia	44
3.4.	Injeção de Erro Gaussiano	50
3.5.	Reestruturação dos Dados	54
3.6.	Implementação do Algoritmo LS-SVM	57
3.7.	Cálculo do Erro Médio	61
3.8.	Comparação de Resultados	62
3.9.	Trajetórias Previstas pelo Algoritmo LS-SVM	67
3.10.	Criação da Função da Trajetória	71
4.	Conclusões	75
4.1.	Trabalhos Futuros	76
	Bibliografia.....	79
	Anexos	83

Lista de Figuras

Figura 1 - Modelo de um algoritmo em aprendizagem automática	3
Figura 2 - Rede Neuronal Biológica vs. Rede Neuronal Artificial [8].....	5
Figura 3 - Ave (Biologia) vs. Aeronave (Engenharia).....	5
Figura 4 - Processos existentes num nó de uma rede neuronal.....	6
Figura 5 - Estrutura de uma rede neuronal feedforward	7
Figura 6 - Influência de σ no conjunto dos dados	9
Figura 7 - Resultado da interpolação com alto sigma vs. Resultado da interpolação com baixo sigma.....	11
Figura 8 - Interpolação gaussiana com diferentes valores de w para sigma alto e sigma baixo respectivamente	11
Figura 9 - Rede Neuronal de uma Função de Base Radial	12
Figura 10 - Rede Neuronal de uma Função de Base Radial vs. Rede Neuronal Clássica	12
Figura 11 - Janela de Análise de um Modelo RBF	13
Figura 12 - Comparação dos valores normalizados das medições e previsões da radiação solar [14].	17
Figura 13 - Obtenção de Mínimos Locais para Redes Neuronal e Mínimo Global para LS-SVM	23
Figura 14 - Princípio da Minimização do Erro de Estimção.....	24
Figura 15 - Relação entre coordenadas Cartesianas e coordenadas Polares	29
Figura 16 - Constelação GPS [30]	30
Figura 17 - Conceito básico do posicionamento GPS [30].....	31
Figura 18 - Funcionamento do ADS-B [31]	32
Figura 19 - Estrutura básica de um sistema INS [32]	33
Figura 20 - Indicador de Rumo e Rumo em relação a um Waypoint	34
Figura 21 - Ângulo de Trajetória de uma Aeronave.....	34
Figura 22 - Sistema Tubo de Pitot -> Velocímetro	35
Figura 23 - Comparação da Função Seno Original com a Função Seno Prevista	39
Figura 24 - Estrutura hierárquica dos processos efetuados pelo programa.....	40
Figura 25 - Distâncias loxodrómicas (rhumb line), em direção ao Pólo Norte	44
Figura 26 - Sistema de coordenadas ECEF (Earth-Centered, Earth-Fixed).....	45
Figura 27 - Trajetória 1 (Coordenadas Geocêntricas) com Listagem de Waypoints.....	46
Figura 28 - Trajetória 1 representada na superfície terrestre.....	47
Figura 29 - Trajetória 2 (Coordenadas Geocêntricas) com Listagem de Waypoints.....	48
Figura 30 - Trajetória 2 representada na superfície terrestre.....	48
Figura 31 - Trajetória 3 (Coordenadas Geocêntricas) com Listagem de Waypoints.....	49
Figura 32 - Trajetória 3 representada na superfície terrestre.....	50
Figura 33 - Diferentes amplitudes do Erro Gaussiano	51
Figura 34 - Erro Gaussiano aplicado à Coordenada X para a Trajetória 1	51

Figura 35 - Erro Gaussiano aplicado à Coordenada Y para a Trajetória 1	51
Figura 36 - Erro Gaussiano aplicado à Coordenada Z para a Trajetória 1	52
Figura 37 - Trajetória 1: Original vs. com Erro Gaussiano.....	53
Figura 38 - Trajetória 2: Original vs. com Erro Gaussiano.....	53
Figura 39 - Trajetória 3: Original vs. com Erro Gaussiano.....	53
Figura 40 - Variação do Parâmetro Regularizador e Valores Utilizados	58
Figura 41 - Exemplo de uma Função Kernel para a Trajetória 1	60
Figura 42 - Exemplo de uma das soluções alpha para a coordenada X - Trajetória 1	60
Figura 43 - Exemplo de uma das soluções alpha para a coordenada Y - Trajetória 1	60
Figura 44 - Exemplo de uma das soluções alpha para a coordenada Z - Trajetória 1	61
Figura 45 - Erros obtidos por Coordenada para a função Multiquádrica - Trajetória 1.....	67
Figura 46 - Erros Global obtido para a função Multiquádrica - Trajetória 1	67
Figura 47 - Trajetória 1: Original vs. Previsão do Algoritmo LS-SVM	68
Figura 48 - Trajetória 1: Original vs. Previsão do Algoritmo LS-SVM na superfície terrestre ..	68
Figura 49 - Trajetória 2: Original vs. Previsão do Algoritmo LS-SVM	69
Figura 50 - Trajetória 2: Original vs. Previsão do Algoritmo LS-SVM na superfície terrestre ..	69
Figura 51 - Trajetória 3: Original vs. Previsão do Algoritmo LS-SVM	70
Figura 52 - Trajetória 3: Original vs. Previsão do Algoritmo LS-SVM na superfície terrestre ..	70

Lista de Tabelas

Tabela 1 - Sistemas de Radar convencionais vs. Sistema ADS-B.....	32
Tabela 2 - Erros obtidos na previsão da função seno.....	38
Tabela 3 - Coordenadas ECEF da aeronave ao longo da trajetória.....	45
Tabela 4 - Coordenadas dos Aeroportos e Waypoints com Tempo de Chegada da Trajetória 1	45
Tabela 5 - Parâmetros operacionais da aeronave 1.....	46
Tabela 6 - Coordenadas dos Aeroportos e Waypoints com Tempo de Chegada da Trajetória 2	47
Tabela 7 - Coordenadas dos Aeroportos e Waypoints com Tempo de Chegada da Trajetória 3	49
Tabela 8 - Coordenadas da aeronave com erro gaussiano.....	52
Tabela 9 - Reestruturação das Coordenadas X, Y e Z para uma dimensão = 3.....	54
Tabela 10 - Reestruturação das Coordenadas X, Y e Z para uma dimensão = 4.....	54
Tabela 11 - Reestruturação das Coordenadas X, Y e Z para uma dimensão = 5.....	54
Tabela 12 - Reestruturação das Coordenadas X, Y e Z para uma dimensão = 6.....	55
Tabela 13 - Reestruturação das Coordenadas X, Y e Z para uma dimensão = 7.....	55
Tabela 14 - Reestruturação das Coordenadas X, Y e Z para uma dimensão = 8.....	55
Tabela 15 - Dados do computador utilizado na simulação.....	62
Tabela 16 - Tempos de Execução do Programa.....	63
Tabela 17 - Erros Mínimos e Máximos obtidos - Trajetória 1.....	63
Tabela 18 - Erros obtidos vs. Dimensão dos Resultados.....	66
Tabela 19 - Erros obtidos vs. Parâmetro Regularizador.....	66
Tabela 20 - Alguns resultados de exemplo (Trajetória 1 - Dimensão 6).....	71
Tabela 21 - Alguns resultados de exemplo (Trajetória 1 - Dimensão 3).....	71
Tabela 22 - Alguns resultados de exemplo (Trajetória 2 - Dimensão 7).....	71
Tabela 23 - Alguns resultados de exemplo (Trajetória 1 - Dimensão 4).....	71
Tabela 24 - Exemplo 1: Previsão de Soluções para Função Multiquádrica.....	72
Tabela 25 - Exemplo 2: Previsão de Soluções para Função Multiquádrica.....	72
Tabela 26 - Exemplo 3: Previsão de Soluções para Função Multiquádrica.....	73
Tabela 27 - Exemplo 4: Previsão de Soluções para Função Exponencial.....	73
Tabela 28 - Exemplo 5: Previsão de Soluções para Função Gaussiana.....	73
Tabela 29 - Exemplo 6: Previsão de Soluções para Função Laplaciana.....	74

Lista de Acrónimos

ASR	<i>Airport Surveillance Radar</i>
ECF	<i>Earth-Centered, Earth-Fixed</i>
ERM	<i>Empirical Risk Minimization</i>
GPS	<i>Global Positioning System</i>
IA	<i>Inteligência Artificial</i>
IRM	<i>International Reference Meridian</i>
IRP	<i>International Reference Pole</i>
KKT	<i>Karush-Kuhn-Tucker</i>
LS-SVM	<i>Least Squares Support Vector Machines</i>
RBF	<i>Radial Basis Function</i>
RVM	<i>Relevance Vector Machines</i>
SRM	<i>Structural Risk Minimization</i>
SSR	<i>Secondary Surveillance Radar</i>
SVM	<i>Support Vector Machines</i>
S³VM	<i>Semi-supervised Support Vector Machines</i>
TSM	<i>Transductive Support Vector Machines</i>
UAV	<i>Unmanned Aerial Vehicle</i>

Lista de Símbolos

$h(x)$	Função de previsão de hipótese para um valor de entrada x
$\ x - x_n\ $	Distância entre o ponto x_n e o ponto x
(x_n, y_n)	Dado de entrada x_n cuja solução correspondente é y_n
λ ou α	Vetor dos multiplicadores de Lagrange (<i>support vectors</i>)
$c(x)$	Restrições (<i>constraints</i>) de desigualdade
∇f	Gradiente da função f
$L(x)$	Função Lagrangeana em ordem a x
$K(x)$	Função Kernel em ordem a x
γ	Parâmetro regularizador do algoritmo LS-SVM
σ	Parâmetro de afinação de funções RBF
U_1 e U_2	Variáveis aleatórias com distribuição normal no intervalo $\in [0, 1]$
V	Velocidade de voo
ψ	Ângulo de rumo de voo
r	Distância do centro da Terra ao C.G. da aeronave
R_E	Raio da Terra
h	Altitude de voo acima do nível do mar
(X, Y, Z)	Posição da aeronave na referência geocêntrica
a, b	Semieixo maior e menor da Terra, respetivamente
e	Excentricidade orbital da Terra
$(AB)_{NM}^{loxo}$	Distância loxodrómica entre A e B em milhas náuticas (<i>nautical miles</i>)

Capítulo 1

1. Introdução

Os problemas de estimação através de processos de regressão datam de um passado bastante distante. Corria o ano de 1632 quando Galileo Galilei utilizou um método que pode ser considerado como a atribuição de uma relação linear entre dados observados pelo mesmo. Essa “colocação” de uma linha no meio de vários pontos medidos revela um problema de regressão linear clássico. Aproximadamente 125 anos depois, Roger Joseph Boscovich (1757) abordou o problema matemático fundamental, determinando quais os parâmetros que se enquadrariam melhor em equações relacionadas com os dados. Desde então, foram desenvolvidos inúmeros métodos para solucionar problemas de regressão linear.

Provavelmente, o método mais conhecido será o método dos mínimos quadrados. Este mesmo método foi publicado primeiramente por Legendre em 1805 e por Gauss em 1809. Ambos os autores aplicaram o método na determinação de órbitas de corpos estelares em torno do sol, através de observações astronómicas [1].

Devido ao aumento do interesse em torno desta área, tornou-se claro que nem sempre é fácil encontrar um modelo paramétrico (linear ou não-linear) que explique alguns fenómenos. A atenção centrava-se na procura de um método mais flexível, onde “os dados pudessem falar por eles próprios”. Posto isto, foram criados os métodos de alisamento não-paramétricos. Esta abordagem foi posta de parte inicialmente devido à simplicidade computacional e conveniência matemática que a abordagem paramétrica demonstrava no ramo estatístico.

Os métodos de regressão continuam a representar uma área ativa de pesquisa. Nas décadas recentes foram criados novos métodos para regressão robusta, regressão em que as variáveis representam curvas, imagens, gráficos ou outros padrões complexos, regressão não-paramétrica, etc.. Além disso, a importância da estimação através de regressão tem aumentado devido ao crescimento dos métodos tecnológicos nas passadas décadas. A necessidade de processos capazes de extrair informação rapidamente de grandes conjuntos de dados tem crescido exponencialmente.

O presente trabalho visa conciliar métodos de regressão avançados, recorrendo a algoritmos de aprendizagem automática, com uma temática recente na indústria aeronáutica: o conceito de navegação 4D, introduzido pelo programa SESAR (*Single European Sky ATM Research*).

O programa SESAR representa um programa de pesquisa cofundado pela SJU (*Sesar Joint Undertaking*), a Comunidade Europeia e a *Eurocontrol*. O objetivo do programa SESAR é implementar uma reforma nas regras e procedimentos do Controlo de Tráfego Aéreo europeu com a finalidade de triplicar o aumento da capacidade de tráfego aéreo, diminuindo os atrasos tanto em terra, como no ar, aumentar a segurança num fator de 10%, reduzir o impacto

ambiental provocado pelas aeronaves em 10% e, por fim, proporcionar serviços de CTA aos utilizadores do espaço aéreo com um custo reduzido em 50% [2].

O conceito I4D (*Initial 4D*) é o primeiro passo da implementação da gestão de trajetórias 4D do programa SESAR. Neste conceito, a aeronave fornece a sua trajetória 4D escolhida, composta pelos *waypoints* definidos pela sua latitude, longitude e altitude e pela previsão do tempo de chegada a cada um.

Numa fase introdutória, será feito um pequeno enquadramento teórico na temática da aprendizagem automática, explicitando conceitos base necessários para uma compreensão completa da metodologia de regressão por algoritmos semi-supervisionados.

1.1. Reconhecimento de Padrões e Filtragem de Dados

A expressão “reconhecimento de padrões” (em inglês, *pattern recognition*) constitui uma disciplina cujo propósito se enquadra na classificação de determinados objetos e na sua integração num determinado número de categorias ou classes. Simbioticamente, a filtragem de dados reflete a necessidade de apurar os resultados com maior precisão e interesse para o utilizador dentro de todo o universo de objetos a classificar. Dependendo da sua aplicação, estes objetos podem ser imagens, ondas de sinais ou quaisquer outras medidas que necessitam de ser classificadas [3].

Tipicamente, a origem dos dados ou padrões a filtrar contem uma grande quantidade de informação que pode não vir a ser totalmente necessária ao utilizador final. Na realidade, apenas um subconjunto de dados costuma ir de acordo com as necessidades de quem as procura. Assim, a filtragem de dados soluciona este problema, apresentando uma vasta gama de estratégias e soluções para tratar os padrões de dados e de pesquisa, excluindo todo o tipo de dados que se possam tornar repetitivos ou irrelevantes. Este tratamento de dados é obtido através da estimação do estado de um sistema a analisar, a partir do momento em que um conjunto de objetos se torna disponível em tempo real (*online*). Este problema é de especial importância para vários ramos da ciência, engenharia, economia, etc. Para resolvê-lo, é necessário modelar a evolução do sistema e do ruído contido nas medições. Geralmente, os modelos resultantes exibem não-linearidades complexas e distribuições não-Gaussianas, impedindo assim a obtenção de soluções analíticas [4].

1.2. Aprendizagem Automática

“Área de estudo que permite aos computadores (máquinas) possuir a habilidade de conseguirem aprender sem terem sido explicitamente programados para tal”. *Arthur Samuel, 1959.*

Generalizando, a aprendizagem automática pode ser definida como um conjunto de métodos computacionais, que recorrem à experiência para efetuar previsões corretas com o objetivo de melhorar a precisão e exatidão com que o algoritmo efetua as mesmas. Neste contexto, experiência refere-se à informação previamente recolhida pelo aprendiz [5] [6]. Por exemplo, uma máquina que tenha sido treinada com mensagens de correio eletrónico (*emails*), para

conseguir distinguir entre mensagens de *spam* e mensagens de *não-spam*. Após essa aprendizagem, essa máquina (ou algoritmo) pode ser utilizada para classificar *emails* futuros e colocá-los em diferentes diretorias.

Como a prestação de um algoritmo de aprendizagem depende dos dados utilizados, todo o ramo da aprendizagem automática está fortemente ligado à análise de dados e estatística. De outro modo, as técnicas de aprendizagem são métodos conduzidos por dados que combinam conceitos de ciência de computadores com princípios estatísticos, probabilísticos e de otimização [5].

Alguns sistemas de aprendizagem automática tentam eliminar a necessidade da interação humana na análise de dados, enquanto outros adotam uma abordagem colaborativa entre Homem e máquina. No entanto, o papel do ser humano não pode ser inteiramente descartado visto que o arquiteto do sistema tem de especificar como é que os dados serão representados e que mecanismos irão ser utilizados para procurar uma caracterização dos dados. Por outro lado, este conceito poderá gerar alguma controvérsia no ser humano. Apesar do desempenho e movimento da máquina estarem limitados pelo seu construtor, o mesmo critério não se pode aplicar à sua capacidade de aprendizagem, visto esta mesma capacidade estar em constante evolução devido à sua aplicação a longo prazo.

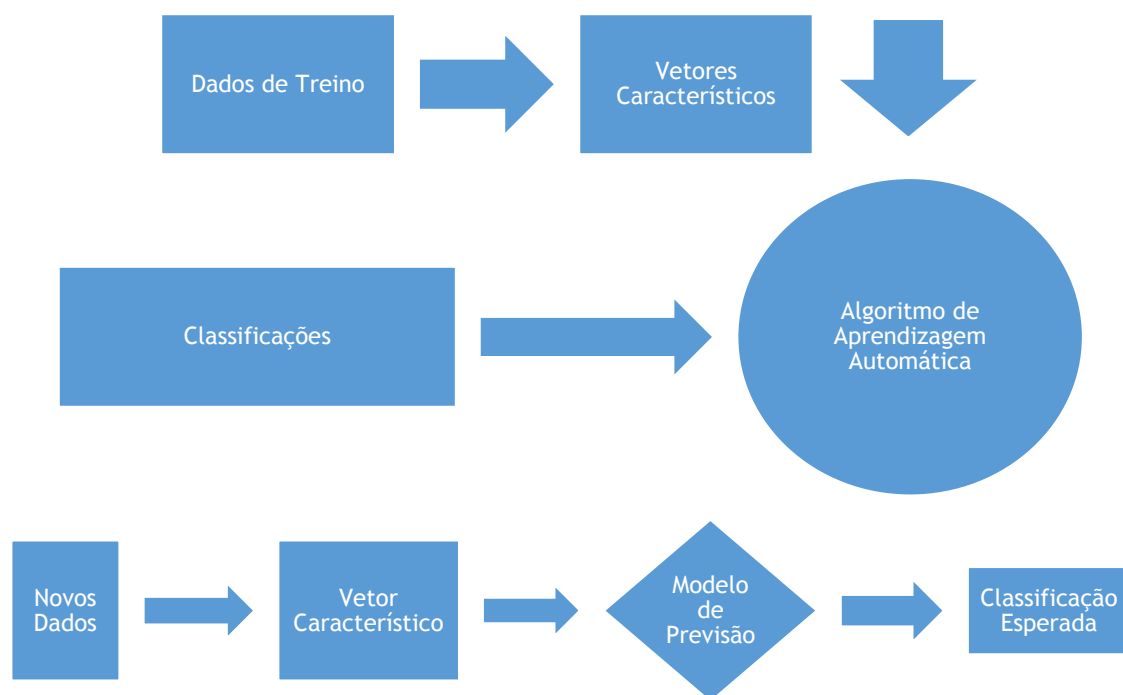


Figura 1 - Modelo de um algoritmo em aprendizagem automática

1.2.1. Problemas Principais Abordados pela Aprendizagem Automática

Todos os dias surgem novas aplicações para os algoritmos de aprendizagem. Além disso, essas mesmas aplicações correspondem a uma vasta variedade de problemas de aprendizagem. As principais classes em que se encaixam esses problemas são:

- Classificação: atribuição de uma categoria para cada objeto;

- Regressão: previsão de um valor real para cada objeto;
- *Ranking*: ordenação de objetos de acordo com um critério predefinido;
- Aglomeração: partição de objetos em regiões homogêneas;
- Redução de dimensionalidade ou aprendizagem múltipla: transformação de uma representação inicial dos objetos numa representação de dimensão mais pequena, preservando algumas propriedades da representação inicial;

A vasta variedade de aplicações acima mencionadas revela a interdisciplinaridade de toda a temática de aprendizagem automática. No entanto, apenas o tópico da regressão será abordado na obtenção dos resultados desta dissertação. Tal como mencionado no ponto anterior, partir-se-á de um conjunto de dados (objetos) e será feita uma previsão da sua solução através de algoritmos de aprendizagem automática.

1.3. Aprendizagem Supervisionada

Nesta categoria de aprendizagem automática, os algoritmos são treinados em exemplos classificados, isto é, onde os dados de saída (*output*) desejados são conhecidos. Neste tipo de aprendizagem, cada exemplo de treino é representado por um par, que consiste num dado de entrada (que normalmente é apresentado em forma de vetor) e num dado de saída (que também pode ser chamado de sinal supervisionado). O algoritmo de aprendizagem supervisionada tenta generalizar uma função de entradas (*inputs*) para saídas (*outputs*) que poderão ser depois usadas para gerar um *output* para *inputs* não registados previamente. Matematicamente, a tarefa da aprendizagem supervisionada pode ser expressa da seguinte forma:

- Dado um conjunto de dados de treino contendo N pares de entradas-saídas: $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$, onde cada y_j foi gerado por uma função desconhecida $y = f(x)$, descubra uma função h que aproxime a função verdadeira f [7].

Neste exemplo, x e y podem representar qualquer valor, não tendo necessariamente de representar um número. A função h representa uma função. Se o resultado da função for um valor contínuo, encontra-se uma situação de regressão, por outro lado, se o algoritmo prever a classe em que a entrada se enquadra, o algoritmo apresenta-se como de classificação [6]. A aprendizagem reflete-se na procura de uma solução no espaço de possíveis hipóteses, mesmo em exemplos novos que se estendam para além do conjunto de treino.

1.4. Redes Neurais Clássicas

No desenvolvimento de muitas áreas da ciência, o Homem inspira-se nos fatores biológicos que o rodeiam para assim conseguir levar a cabo novas experiências e a implementação de novos métodos. Tal como o Homem tem a capacidade de aprender, este quis que as suas máquinas possuíssem a mesma capacidade, replicando a função biológica presente no mundo que o rodeia

e criando uma estrutura biológica artificial. Por outras palavras, ao imitar a estrutura biológica em estudo, o Homem espera que esta se comporte de igual modo.

No caso das redes neuronais, a inspiração biológica surgiu das ligações neurológicas no cérebro, onde se encontram neurónios ligados por sinapses, em que cada neurónio desempenha uma tarefa simples que depende do estímulo de sinapses anteriores.

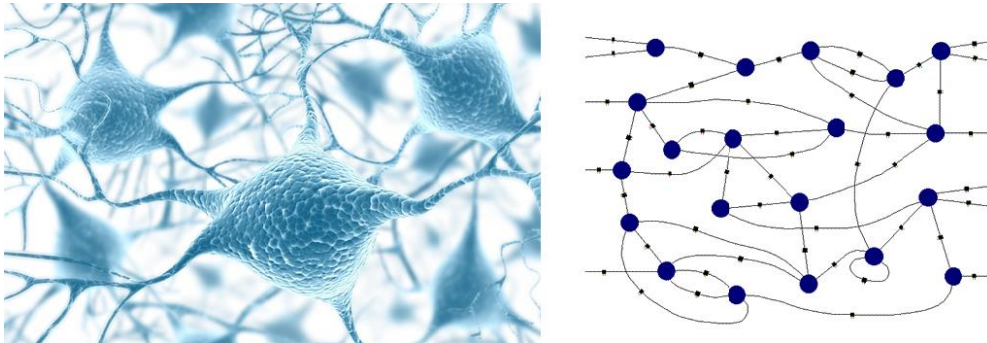


Figura 2 - Rede Neuronal Biológica vs. Rede Neuronal Artificial [8]

Outro exemplo em que o Homem se baseou na natureza para progredir a nível científico e escrever uma das páginas mais importantes da História, foi a observação do comportamento das aves enquanto voavam e a criação de um veículo que lhe permitisse viajar pelos céus de modo semelhante.



Figura 3 - Ave (Biologia) vs. Aeronave (Engenharia)

Após uma longa cadeia de eventos, um dos resultados obtidos pode ser observado na figura anterior. A nível estrutural, é possível retirar bastantes semelhanças entre a estrutura artificial e a estrutura biológica real. No entanto, a engenharia faz uma separação da biologia, na medida em que, após a estrutura inicial estar completa, o seu objetivo é apenas a realização da tarefa em mente, e não replicar por completo a sua inspiração biológica. Isto é, apesar da aeronave apresentar uma estrutura semelhante à de um pássaro, não necessita de bater as asas ou de estar coberto de penas para voar.

Uma característica fundamental das redes neuronais é a sua capacidade de aproximar funções não lineares arbitrárias. As redes neuronais têm o objetivo de serem bastante eficientes nesta

temática, visto que a aprendizagem automática abrange uma grande área de tomada de decisões não lineares. Todo o processo de aproximação de uma função não linear através da interação de um sistema com a aplicação de dados no seu comportamento pode ser interpretado como “aprendizagem”. Através do uso de redes neurais, um sistema inteligente pode ser capaz de aprender e executar tarefas altamente cognitivas.

Uma rede neuronal é constituída por um conjunto de nós, normalmente organizados em fases e interligados através de elementos com o nome de sinapses. Em cada um dos nós, os dados de entrada são somados, limitados e sujeitos a uma função de ativação com o objetivo de gerar um dado de saída (correspondência) desse nó. A estrutura mencionada anteriormente pode ser visualizada na Figura 4, onde $x_1, x_2, x_3, \dots, x_n$ representam os dados de entrada, $w_1, w_2, w_3, \dots, w_n$ representam os pesos sinápticos, Σ representa uma função de transferência e φ representa a função de ativação.

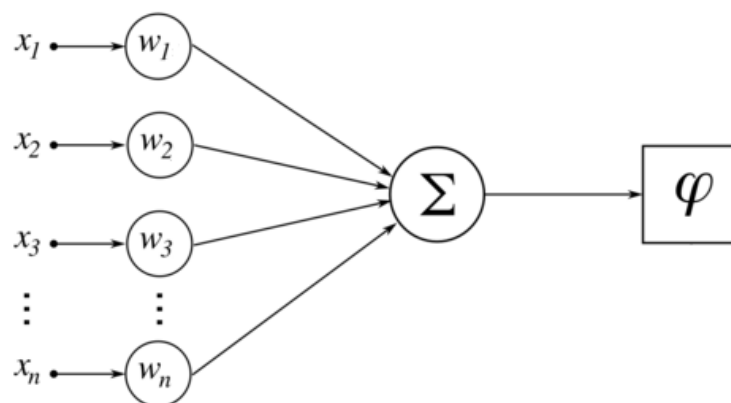


Figura 4 - Processos existentes num nó de uma rede neuronal

Ao contrário dos computadores, que são programados para resolver problemas através de algoritmos sequenciais, o cérebro utiliza uma enorme rede de elementos de computação paralela e distribuída chamados neurónios. Um elevado número de conexões que interliga esses elementos permite ao ser humano a elevada capacidade de executar processos de aprendizagem. As redes neurais artificiais surgiram com a necessidade de possuir sistemas que conseguissem processar a informação de modo semelhante. Estas são compostas por um grande número de elementos de processamento análogos altamente interligados à semelhança de neurónios biológicos e são aglomerados por conexões que correspondem às sinapses cerebrais.

1.4.1. Topologia das Redes Neurais

Por norma, uma rede neuronal artificial é composta por um conjunto de unidades de processamento paralelo e distribuído, chamados nós ou neurónios, que depois são organizados por fases e interligados por meio de canais de sinal ponderado. A Figura 5 representa uma típica rede neuronal *feedforward* onde as conexões representam os pesos (ponderações) e os elementos internos os neurónios (ou nós).

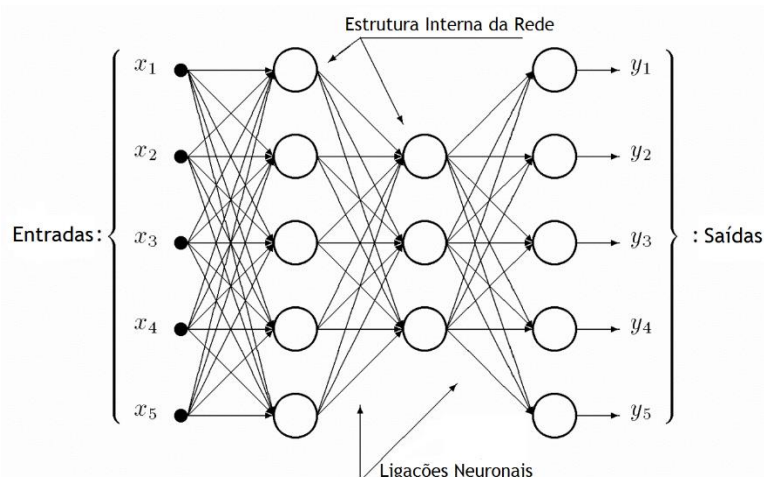


Figura 5 - Estrutura de uma rede neuronal feedforward

As redes neurais adquirem o seu conhecimento através da detecção de padrões e relações encontrados nos dados que lhes são fornecidos. As três características mais importantes que definem uma rede neuronal artificial são as seguintes: topologia da rede, funções de transferência da rede e algoritmo de aprendizagem da rede. As redes *feedforward* relacionam diretamente os dados de entrada com os dados de saída, enquanto que outras redes recorrentes relacionam os dados de saída com os dados de entrada e com o estado do nó anterior. A topologia escolhida foi a de *feedforward* dada a sua velocidade e falta de necessidade de utilizar o estado anterior para prever os dados de saída.

1.5. Método Kernel

Os métodos Kernel enquadram-se numa classe de algoritmos que visam o reconhecimento ou análise de padrões, cujos elementos mais conhecidos são as Máquinas de Suporte Vetorial (ou *Support Vector Machines* - SVM). A característica geral na análise de padrões centra-se na procura e estudo de relações que possam existir entre um dado conjunto de informação (sequências, documentos, conjunto de pontos, vetores, etc.). No entanto, a principal característica dos métodos Kernel é a sua abordagem distinta a este problema. Os métodos Kernel mapeiam os dados num espaço dimensional superior, na esperança de que a informação seja separada com maior facilidade, ou estruturada de maneira mais eficiente. Além disso, não existem quaisquer restrições nesta forma de mapeamento, o que pode levar a uma construção de espaços com dimensões infinitas. No entanto, a função de mapeamento quase nunca necessita de ser calculada, devido à existência de uma ferramenta denominada “truque kernel”.

Este “truque kernel” representa uma ferramenta matemática que pode ser aplicada a qualquer algoritmo que dependa unicamente do produto interno entre dois vetores, na medida em que sempre que surge o cálculo de um produto interno, este é substituído por uma função kernel. Quando devidamente aplicado, os algoritmos com um comportamento linear transformam-se

em algoritmos não-lineares, mantendo a sua equivalência em relação aos originais e operando num espaço característico φ .

O processo de seleção do kernel mais apropriado depende sempre do problema que se está a tentar solucionar e pode revelar-se pesado e entediante, na medida em que os seus parâmetros internos têm de ser ajustados com precisão.

Tal como mencionado anteriormente, um dos problemas fundamentais dos teoremas de aprendizagem automática é a atribuição correta de uma classe a objetos sobre os quais ainda não foi apresentada informação. Formalizando o problema:

$$(x_1, y_1), \dots, (x_m, y_m) \in D \quad (1)$$

onde D representa um conjunto (também tratado por domínio) não vazio de onde os dados de treino x_i (que podem ser chamados *inputs*, observações, casos, etc.) são retirados. Os y_i representam os *outputs* (ou soluções) [9]. É de notar que os dados de treino poderão representar qualquer tipo de dados.

Num tipo de classificação binária, a caracterização das saídas só pode resultar em dois casos: dois *outputs* poderão ser idênticos, ou diferentes. A questão maior levanta-se aquando a escolha da medição de semelhanças dos *inputs*.

A nível matemático, uma demonstração simples desta medição de semelhanças é o produto escalar (ou produto interno). Por exemplo, dados dois vetores $x, x' \in \mathbb{R}^N$, o seu produto interno é definido como:

$$\langle x, x' \rangle = \sum_{i=1}^N [x]_i [x']_i \quad (2)$$

onde $[x]_i$ refere-se ao dado i do vetor x .

Através da interpretação geométrica do produto interno conclui-se que calcula o cosseno do ângulo entre os vetores x e x' , caso o comprimento de ambos seja unitário. Além disso, é possível calcular a norma do vetor através de: $\|x\| = \sqrt{\langle x, x \rangle}$.

Do mesmo modo, a distância entre dois vetores é definida como o comprimento do vetor resultante entre a sua diferença. Assim, ao conseguir calcular o produto interno entre vetores, é possível realizar todas as construções geométricas que poderão ser feitas em termos de ângulos, comprimentos ou distâncias. No Anexo II pode ser consultada uma explicação matemática mais desenvolvida acerca do produto interno.

Para ser possível utilizar o produto interno é necessário representar os *inputs* como vetores, num determinado espaço de soluções \mathcal{H} . Com este objetivo, é representado um mapa:

$$\Phi: \mathcal{X} \rightarrow \mathcal{H} \text{ e } x \mapsto x = \Phi(x) \quad (3)$$

Através desta representação é possível lidar com os dados geometricamente, utilizando algoritmos de aprendizagem que recorram a métodos algébricos e de geometria analítica.

1.6. Funções de Bases Radiais

O modelo básico de uma função de base radial (em Inglês: *Radial Basis Function* - RBF), é dado pela seguinte expressão:

$$\text{Cada } (x_n, y_n) \in D \text{ influencia } h(x) \quad (4)$$

Em termos teóricos, a expressão acima enuncia que cada ponto (x_n, y_n) que pertence a um conjunto de dados D vai influenciar o valor da hipótese h em cada ponto x , ou seja $h(x)$. Tal como em qualquer método de aprendizagem automática, o processo de aprendizagem é feito através dos dados e pela escolha de uma hipótese (ou teorema) que irá ser afetado por esses mesmos dados.

No caso particular em estudo das funções de bases radiais, a hipótese que se coloca é afetada da seguinte forma:

$$\text{Cada } (x_n, y_n) \in D \text{ influencia } h(x) \text{ baseado em } \|x - x_n\| \quad (5)$$

Por outras palavras, um ponto no conjunto de dados vai afetar mais os pontos que se encontram na sua vizinhança do que os pontos que se encontram mais afastados, devido à hipótese ser afetada por um parâmetro de distância $\|x - x_n\|$. Na figura seguinte, pode ser observada a evolução da influência que cada ponto x_n tem sobre os pontos que o rodeiam no conjunto de dados D .

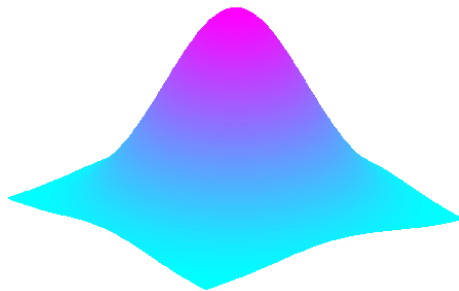


Figura 6 - Influência de x_n no conjunto dos dados

Da análise da figura, retira-se que a influência é simétrica a toda a sua volta, ou seja, tem mais influência na proximidade até que começa a diminuir para os pontos ligeiramente mais afastados, perdendo qualquer poder sobre os pontos que se encontram mais afastados no conjunto de dados.

Assim, numa forma mais desenvolvida, um exemplo de um modelo de uma função de base radial pode apresentar a seguinte forma:

$$h(x) = \sum_{n=1}^N w_n \exp\left(-\frac{\|x - x_n\|^2}{2\sigma^2}\right) \quad (6)$$

onde o somatório representa a soma das influências de todos os pontos, w_n um peso a ser determinado e $\|x - x_n\|^2$ a distância entre os pontos do conjunto de dados. Todo este modelo traduz a contribuição para o ponto x , no qual estamos a avaliar a função, de acordo com o ponto x_n do conjunto de dados.

A partir da expressão acima apresentada, é possível esclarecer o porquê da atribuição do nome “função de base radial”. Na equação anterior, a função de base é representada por $\exp\left(-\frac{\|x - x_n\|^2}{2\sigma^2}\right)$. Esta função é caracterizada por ser o bloco de construção de todo o problema.

Neste caso é utilizada uma função exponencial (gaussiana), mas podia ser aplicada toda uma variedade de funções de base (multiquádrica, laplaciana, etc.). O termo radial, $\|x - x_n\|$, demonstra a importância que as distâncias entre os pontos do conjunto de dados reflete na resolução do problema.

1.6.1. Algoritmo de Aprendizagem

A solução do algoritmo de aprendizagem, sob forma matricial (para um modelo gaussiano), é dada pela seguinte fórmula:

$$\begin{bmatrix} \exp\left(-\frac{\|x_1 - x_1\|^2}{2\sigma^2}\right) & \dots & \exp\left(-\frac{\|x_1 - x_N\|^2}{2\sigma^2}\right) \\ \exp\left(-\frac{\|x_2 - x_1\|^2}{2\sigma^2}\right) & \dots & \exp\left(-\frac{\|x_2 - x_N\|^2}{2\sigma^2}\right) \\ \dots & \dots & \dots \\ \exp\left(-\frac{\|x_N - x_1\|^2}{2\sigma^2}\right) & \dots & \exp\left(-\frac{\|x_N - x_N\|^2}{2\sigma^2}\right) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix} \quad (7)$$

em que a primeira matriz do termo à esquerda é representada pela letra ϕ e traduz a transformação não-linear do modelo RBF. Para garantir que existe solução para o problema acima enunciado, tem de ser garantido que ϕ é invertível, sob estas condições, então:

$$w = \phi^{-1} \cdot y \quad (8)$$

Esta solução tem o nome de interpolação exata, ou seja, nos pontos em que a solução é conhecida (dados de treino), obtém-se a solução exata (y_n).

O Kernel (gaussiano neste caso) efetua uma interpolação entre o conjunto de dados, para obter as soluções para todos os valores de x . A interpolação é exacta porque a solução que se procura é coincidente com a solução correta.

1.6.2. Efeito de Sigma

De seguida, será analisado o efeito que a escolha do valor do parâmetro σ pode apresentar na solução final do modelo RBF:

$$h(x) = \sum_{n=1}^N w_n \exp\left(-\frac{\|x - x_n\|^2}{2\sigma^2}\right) \quad (9)$$

Se for atribuído um valor alto ao parâmetro sigma, resultará numa influência demasiado plana na previsão das soluções, obtendo uma curva gaussiana bastante larga. Por outro lado, se for atribuído um valor demasiado baixo a sigma, a curva gaussiana resultante será bastante côncava. Esta diferença entre a escolha do parâmetro sigma pode ser analisada na figura seguinte:



Figura 7 - Resultado da interpolação com alto sigma vs. Resultado da interpolação com baixo sigma

Dependendo de onde os pontos se encontram, ou quão afastados estão entre si, existe uma grande diferença a nível de resultados entre interpolar com um valor baixo de sigma, ou um valor alto.

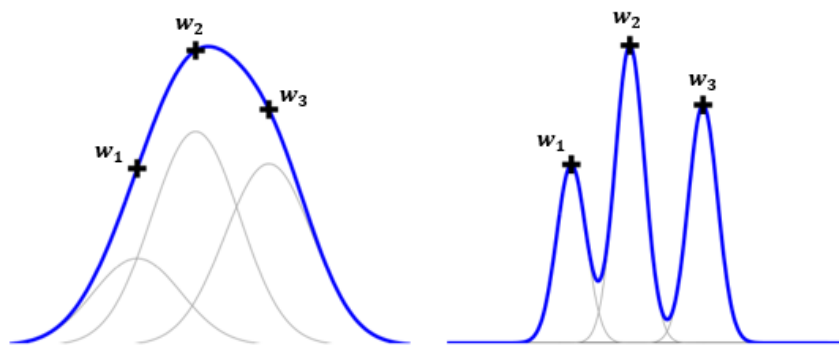


Figura 8 - Interpolação gaussiana com diferentes valores de w para sigma alto e sigma baixo respetivamente

Na figura anterior, as linhas mais claras representam a contribuição individual de cada um dos pontos (w_1, w_2 e w_3) e, conseqüentemente, (x_1, x_2 e x_3), enquanto que a linha a cheio representa a contribuição total das três interpolações. Neste último caso, a interpolação passa exactamente nos três pontos estudados e resulta da soma entre os gaussianos de w_1, w_2 e w_3 . Como se pode observar no segundo caso, os gaussianos individuais desaparecem rapidamente. Apesar de ainda se satisfazerem as equações, a interpolação é muito pobre, visto que a influência de cada ponto desaparece rapidamente.

1.7. Relação com Redes Neurais

Uma rede neuronal de um modelo de uma função de base radial pode ser ilustrada como na Figura 9.

Partindo de um valor inicial x , a sua hipótese $h(x)$ obtém-se através do cálculo do “componente radial”, que por sua vez depende da distância a x_1, x_n e x_N , sucessivamente. De seguida, o resultado é entregue ao componente não-linear da rede neuronal ϕ . Os resultados dessas funções de base, são depois ligados às variáveis w com a finalidade de criar a solução $h(x)$. De acordo com a função explícita anteriormente, as características são: $\exp(-\frac{\|x-x_N\|^2}{2\sigma^2})$. Esta transformação não-linear depende do conjunto de dados inicial D .

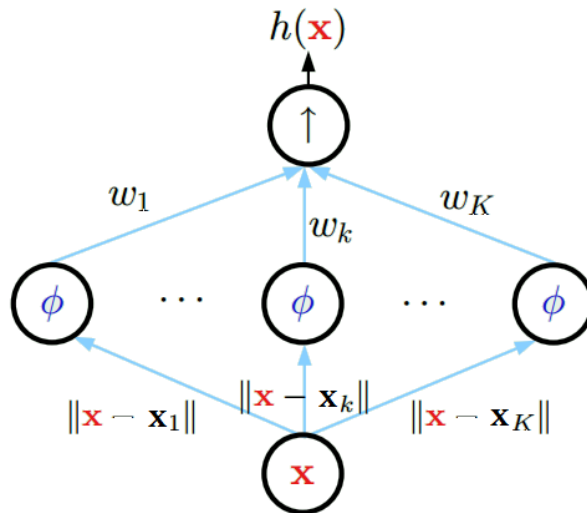


Figura 9 - Rede Neuronal de uma Função de Base Radial

Nos problemas com RBF, costuma ser adicionado um termo de viés (*bias term* - \mathbf{b} ou w_0), que entra na última camada do processo.

1.7.1. Comparação com Redes Neurais Clássicas

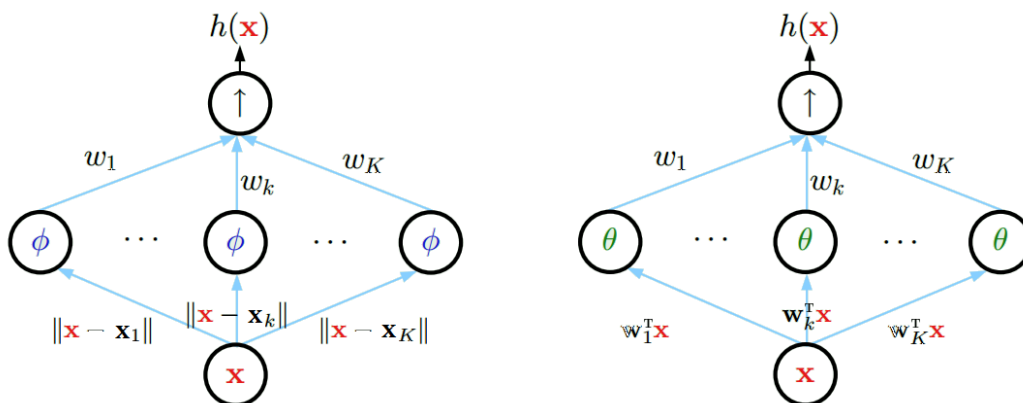


Figura 10 - Rede Neurais de uma Função de Base Radial vs. Rede Neurais Clássica

Em ambas as redes, o processo inicia-se com um dado de entrada (*input*) x e depois são calculadas as “características”. Nas redes RBF, as “características” dependem da distância entre os pontos, na medida em que quanto maior for a distância entre eles, menor será a sua influência. Ou seja, se o valor de $\|x - x_1\|$ for elevado, a sua característica correspondente ϕ não terá contribuição para a solução (*output*) $h(x)$. Contrariamente, nas redes neurais clássicas, o valor $w_1^T x$, independentemente da sua magnitude, passará pelo sigmóide θ , tendo sempre alguma contribuição para a solução final.

A conclusão que pode ser retirada, é o facto das redes RBF observarem apenas regiões locais no espaço de dados, menosprezando a influência dos pontos mais afastados.

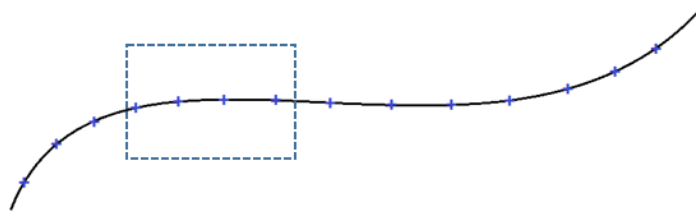


Figura 11 - Janela de Análise de um Modelo RBF

À medida que a análise prossegue, todo o processo efetuado nos pontos anteriores não irá interferir na janela que está a ser analisada.

1.8. Máquina de Suporte Vetorial com Base nos Mínimos Quadrados

Teoricamente, os SVM revelam-se como um dos métodos de classificação mais eficazes da era moderna da aprendizagem automática. Este método representa uma generalização não-linear do algoritmo de descrição generalizada desenvolvido na década de 60 por Vapnik, Lerner e Chervonenkis [10]. A sua popularidade deve-se à aplicação na resolução de problemas de classificação, regressão e deteção e graças ao seu contexto industrial. Uma das características importantes dos SVM é que a determinação dos modelos paramétricos corresponde a um problema de otimização convexo, fazendo com que qualquer solução local seja também um valor ótimo global. A técnica de classificação dos algoritmos SVM pode ser subdividida em duas partes distintas:

1. Aprendizagem: consiste em treinar o algoritmo SVM com exemplos previamente confirmados;
2. Previsão: onde são inseridos novos dados sem que se saiba o resultado dos mesmos. Os resultados produzidos terão sempre em conta os exemplos utilizados na fase de aprendizagem.

As abordagens clássicas de técnicas de reconhecimento de dados apoiam-se no princípio ERM (*Empirical Risk Minimization*), enquanto que os SVM baseiam-se no conceito SRM (*Structural Risk Minimization*) [11]. Com base em teorias de aprendizagem estatística, o conceito SRM apresenta uma melhor generalização (ex.: em dados nunca antes reconhecidos) através de uma minimização do intervalo superior (soma dos dados adquiridos com erros e com os termos dependentes da dimensão VC).

Nota: Dimensão VC (Vapnik-Chervonenkis - criadores do método SVM) representa a capacidade da classificação estatística de um algoritmo, definida como o maior padrão de pontos que o algoritmo consegue quebrar).

No entanto, o método tradicional de treino de algoritmos SVM apresenta algumas limitações no que toca a complexidade computacional, especialmente quando o conjunto de dados de treino é bastante grande tornando o tempo de treino bastante elevado, não sendo compatível com a necessidade de obter os requisitos práticos com a maior rapidez [12]. Por outro lado, o método de Máquina de Suporte Vetorial com Base nos Mínimos Quadrados (*Least-Square Support Vector Machines* - doravante denominado LS-SVM) obtêm a solução através da resolução de um conjunto de equações lineares em conjunto com um método de gradiente em vez de um problema de programação quadrática convexa apresentado pelos algoritmos clássicos SVM. Os métodos de aprendizagem LS-SVM foram introduzidos por Suykens et. al [13] e representam uma versão modificada e simplificada dos algoritmos tradicionais SVM. Este método permite lidar com variáveis lineares e não lineares de um modo muito mais rápido [14] e tem como objetivo a minimização do erro do quadrado mínimo nos dados de treino, enquanto ao mesmo tempo maximiza a margem entre duas classes [15]. Através dos estudos extensivos realizados no passado [13], é possível concluir que os LS-SVM demonstram uma boa *performance* em vários problemas de classificação e regressão. No entanto, surgem duas limitações como principais problemas deste método. Em primeiro lugar, o procedimento de treino dos algoritmos LS-SVM tem como base a resolução de um conjunto de equações lineares. Embora, teoricamente, seja possível o problema de treino, na prática torna-se impossível para um grande número de dados de início, porque a sua complexidade computacional tende a escalar de forma cúbica com a dimensão dos exemplos de entrada. Em segundo lugar, a solução do algoritmo LS-SVM é muito ambígua, fazendo com que a velocidade dos testes seja muito inferior em comparação com outros algoritmos de treino, como os SVM.

1.9. SESAR e a Navegação por Waypoints 4D

Em 2004 a Comissão Europeia iniciou o projeto do Céu Único Europeu (*Single European Sky ATM Research - SESAR*). Este programa é a resposta operacional e tecnológica aos maiores desafios do CTA Europeu, assegurando a modernização do mesmo através da coordenação e da concentração de investigações relevantes e de esforços desenvolvidos na União Europeia com o objetivo de harmonizar a implementação [2].

As trajetórias a quatro dimensões (4D) representam um passo chave no conceito futurístico de gestão de tráfego aéreo que está atualmente a ser desenvolvido pelo programa SESAR.

Antes e ao longo de todo o voo, as informações da trajetória da aeronave que contêm as posições atual e prevista serão trocadas com todos os ANSPs (*Air Navigation Service Providers*) envolvidos e todas as limitações temporais que poderão existir serão definidas em waypoints específicos na vizinhança de áreas congestionadas. As operações de Controlo de Tráfego Aéreo serão automatizadas a um nível superior do que aquele em que se encontram hoje, com trocas de dados diretas entre sistemas aéreos e terrestres. Irá registar-se um aumento na segurança e na previsão de voo devido ao facto de serem fornecidas informações precisas acerca das posições atuais e futuras de cada aeronave, dadas em qualquer instante.

As quatro dimensões em que se dividem as trajetórias 4D são as seguintes:

- Dimensões 1 & 2: Latitude e Longitude - informam a posição da aeronave no mapa através das suas coordenadas respetivas;
- Dimensão 3: Altitude - a altitude da aeronave em relação ao solo, ou nível do mar, dependendo da referência utilizada para uma determinada fase de voo;
- Dimensão 4: Tempo - a posição e altitude da aeronave são definidas para momentos específicos no tempo durante todo o voo.

O desafio principal na implementação destas trajetórias assenta na definição das regras básicas, procedimentos e metodologias, também como nos padrões mundiais para trocas de informações acerca das trajetórias 4D [16]. Terão de ser formalizados acordos internacionais para cobrir a interoperabilidade de viabilizações técnicas, tais como os equipamentos de comunicação internacionais.

1.10. Automatic Dependent Surveillance-Broadcast

Atualmente, os sistemas de vigilância aérea no que toca à gestão do tráfego aéreo, atingiram os seus limites operacionais, e não conseguem acompanhar o aumento do tráfego aéreo que se tem registado nos últimos tempos [17]. Entre outros problemas, os sistemas de vigilância atuais sofrem de uma falha na operação quando operam a baixas altitudes ou em áreas remotas (como por exemplo acima de largos territórios oceânicos) e durante condições climatéricas extremas. Além disso, o desempenho destes sistemas desatualizados é insuficiente para satisfazer os requerimentos de funcionamento das operações com trajetórias 4D, tal como mencionado no tópico anterior. Com isto em mente, a Organização Internacional de Aviação Civil (ICAO) propôs um novo sistema de vigilância: o *Automatic Dependent Surveillance Broadcast* (ADS-B). O ADS-B representa uma tecnologia de vigilância que está a revolucionar o paradigma da comunicação e navegação, na qual a aeronave determina a sua posição através da navegação de satélites, que depois transmite periodicamente, permitindo assim que esteja seja localizada. Esta informação pode ser recebida pelas estações terrestres de controlo de tráfego aéreo, como um substituto de um radar secundário. Também poderá ser recebida por outra aeronave, permitindo assim uma auto-preparação em caso de possível contacto. Já provado e certificado como sendo um ótimo substituto de baixo custo do sistema de radar convencional, o ADS-B permite ao piloto e aos controladores aéreos “ver” e controlar a aeronave com maior precisão, e ao longo de uma percentagem maior da superfície terrestre. Este sistema é automático, na medida em que não necessita de quaisquer ações do piloto ou de um *input* exterior. A sua única dependência encontra-se na necessidade de utilizar dados do sistema de navegação da aeronave.

O sistema de vigilância cooperativa ADS-B faz parte de uma das reformas do plano SESAR. Atualmente é obrigatório em algumas partes do espaço aéreo Australiano e já faz parte do controlo de tráfego aéreo Canadiano. Em espaço aéreo Europeu, será obrigatório para algumas aeronaves a partir do ano 2017 [18].

1.11. Algumas Aplicações do Método LS-SVM

Seguidamente, são mostradas algumas aplicações em áreas distintas, com a finalidade de ilustrar a relevância prática da estimação e previsão de dados através de métodos de regressão. É necessário ter em atenção que nem todos os métodos ou técnicas descritas nesta dissertação poderão ser aplicados diretamente neste tipo de conjunto de dados.

Exemplo 1.1 - Ciências Biomédicas:

Numa análise efetuada em pacientes com doenças terminais, existe um forte interesse em prever o tempo médio de esperança de vida, dada uma determinada descrição de dados relevantes para o caso, tais como: tipo de doença, medições sanguíneas, idade, terapia, etc.. O resultado pode ser utilizado para determinar a terapia adequada para cada paciente, maximizando assim o tempo médio de vida do mesmo [19].

Exemplo 1.2 - Economia e Finanças:

Em casos de empréstimos bancários, um banco tem necessidade em prever o retorno do investimento feito num cliente. Supõe-se que o banco tem acesso ao perfil do cliente, incluindo um histórico do cartão de crédito, bens, profissão, salário, idade, etc.. O retorno previsto poderá afetar na atribuição ou não atribuição do empréstimo por parte do banco, bem como as condições do mesmo [20].

Exemplo 1.3 - Identificação de Sistemas:

Após terem sido recolhidos dados de uma coluna de destilação, o objetivo seria determinar a temperatura no fundo e no topo da coluna. As previsões obtidas teriam de apresentar medições precisas da temperatura, visto que estes fatores controlam a qualidade final do produto. Neste exemplo real, é utilizado um modelo linear para descrever as temperaturas com elevada precisão [21].

Exemplo 1.4 - Engenharia:

Previsão de energia solar: A quantidade de radiação solar que atinge um determinado local representa um dado essencial quando se trata de painéis fotovoltaicos. A eficiência dos sistemas diminui facilmente devido às alterações climáticas. Neste estudo [14], a aplicação de algoritmos LS-SVM tem como objetivo a previsão da radiação solar em determinados dias, recorrendo a medições da mesma em dias anteriores. O modelo de previsão contém cinco *inputs* e um *output*. O dia, temperatura média diária, temperatura máxima diária, duração diária de sol e radiação solar do dia anterior formam os parâmetros de entrada no algoritmo LS-SVM. A radiação solar diária representa o *output*. Os dados foram separados em dois conjuntos de 1096 dias para o treino e de 365 dias para o teste. A comparação entre os valores normalizados dos

dados das medições e dos dados previstos pelo algoritmo foi efetuada para medir o desempenho do algoritmo LS-SVM e podem ser observados na figura 4.

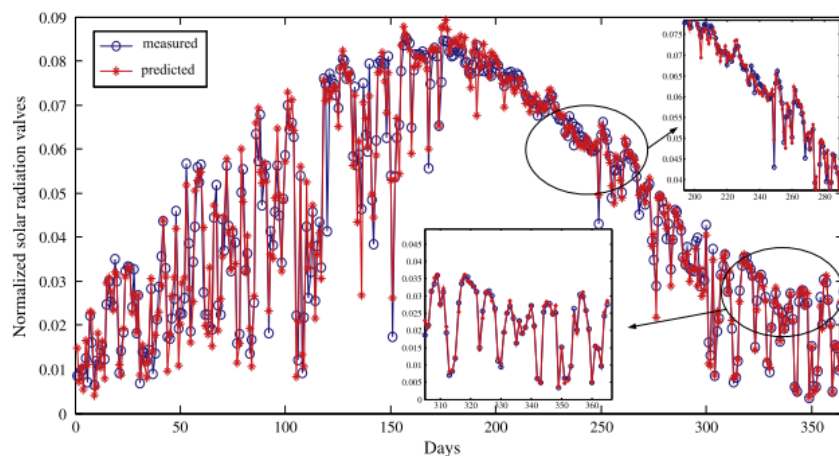


Figura 12 - Comparação dos valores normalizados das medições e previsões da radiação solar [14].

O desempenho do método proposto foi comparado com outras técnicas de IA e com trabalhos realizados previamente. Através da avaliação dos resultados, o sucesso deste trabalho é evidente quando comparado com trabalhos realizados anteriormente.

Os resultados demonstraram que o método proposto baseado em regressão com algoritmos LS-SVM revela uma grande eficácia e torna-se fiável na estimação de valores de radiação solar a partir de dados meteorológicos.

1.12. Objetivos do Trabalho Proposto

O trabalho desenvolvido ao longo desta dissertação tem como pilar a aplicação de métodos de aprendizagem automática na previsão de trajetórias de aeronaves. Especificamente, são aplicados métodos de regressão que se inserem na máquina de suporte vetorial, neste caso, os *Least-Squares Support Vector Machines* a um conjunto de dados de trajetórias reais com quatro dimensões (4D), com o objetivo de criar uma função que preveja estas mesmas trajetórias ao longo de todo percurso, permitindo assim definir um perfil completo de toda a trajetória da aeronave, desde o seu ponto de descolagem até ao ponto de aterragem. Este processo foi efetuado recorrendo a várias simulações computacionais, após a criação e aplicação dos algoritmos necessários. Para além disso, são enunciados ao longo da dissertação, todos os teoremas e formulações matemáticas necessários para a compreensão total dos métodos aplicados.

O objetivo principal passa pela demonstração da interdisciplinaridade dos métodos de suporte vetorial e também pelo interesse que pode suscitar na aplicação à previsão de trajetórias de aeronaves comerciais, quando estas operam num ambiente de quatro dimensões. Além disso, permite fazer um estudo comparativo no que toca à metodologia das redes neuronais clássicas

e dos processos utilizados pelas redes neurais aqui implementadas, tais como as funções de base radial descritas anteriormente.

1.13. Estrutura da Dissertação

Inicialmente, no primeiro capítulo, é feita uma contextualização teórica acerca do método de regressão LS-SVM e a temática do projeto SESAR, no âmbito da aplicação de trajetórias 4D no céu europeu. Inicia-se com uma introdução ao tema mais geral, com o objetivo de integrar o leitor com alguns fundamentos da inteligência artificial e as suas aplicações na vida real, dando lugar a temas mais objetivos, tais como o desenvolvimento de redes neurais artificiais, ferramentas utilizadas no desenvolvimento e aplicação do método LS-SVM (método kernel, funções radiais de base, etc.) e uma pequena abordagem ao tema dos *support vector machines* e *least-squares support vector machines*, especificando assim a temática, para que se fique com uma imagem clara tanto do conceito geral, como dos temas mais objetivos. Para além da abordagem teórica ao projeto SESAR e ao céu único europeu, é também feita uma breve análise acerca de uma das ferramentas de vigilância de tráfego aéreo atual, o ADS-B, que irá ser desenvolvida em maior detalhe no capítulo seguinte. Ainda no final do primeiro capítulo, são apresentadas outras aplicações do método LS-SVM para que o leitor compreenda as possibilidades deste método de aprendizagem automática, fora do âmbito da engenharia aeronáutica.

O segundo capítulo tem como principal objetivo desenvolver os princípios matemáticos pelos quais o método LS-SVM se rege, para assim ser possível obter uma compreensão total dos procedimentos utilizados por trás do processo de modelação dos resultados. É de referir que algumas formulações matemáticas foram remetidas para anexo devido ao seu conteúdo excessivamente extenso. Para além do desenvolvimento do método LS-SVM, são também ilustradas todas as funções RBF que foram aplicadas a nível prático, bem como o método de injeção de erro Gaussiano nos dados, para assim permitir uma maior fiabilidade dos mesmos. Na última parte do segundo capítulo, é também feita uma pequena abordagem a alguns dos princípios de localização aeroespacial, para que o leitor fique inteirado sobre os temas de sistemas de navegação por satélite (ADS-B) e sistemas de navegação inercial, bem como rumo, ângulo de trajetória e velocidade de uma aeronave.

No terceiro capítulo são apresentados os resultados que foram obtidos através da simulação e validação dos métodos apresentados no capítulo anterior. Inicialmente, é feita uma abordagem mais académica ao tema, aplicando o método à previsão de uma função simples (função seno), e só depois é demonstrada a aplicação do LS-SVM em previsão de trajetórias. Antes da apresentação dos resultados finais, são demonstrados também alguns métodos adicionais que foram implementados na obtenção dos resultados, bem como a geração dos dados de trajetória, a injeção de erro gaussiano nos dados e a reestruturação dos dados iniciais. No final, são analisadas três trajetórias distintas e efetuadas previsões para cada uma delas, através da função classificadora gerada.

Por último, o quarto capítulo tem como objetivo apresentar ao leitor, as conclusões obtidas com este projeto, não só a nível da previsão de trajetórias, mas também no que toca a desafios e contrapartidas encontrados ao longo do desenvolvimento do trabalho. É também feita uma breve referência a trabalhos que podem vir a ser realizados no futuro, dentro e fora da área da engenharia aeronáutica.

De referir ainda que após a secção reservada à bibliografia, se encontram todos os anexos referentes a esta dissertação, onde podem ser encontrado um artigo escrito com base nos resultados obtidos neste projeto, alguns dos princípios matemáticos consultados e utilizados para realizar alguns dos métodos propostos bem como uma listagem dos tempos de execução dos programas construídos para esta dissertação.

Capítulo 2

2. Princípios da Aprendizagem Automática com Base na Máquina de Suporte Vetorial

2.1. Condições de Otimalidade

Antes de ser desenvolvido em detalhe o método de regressão através de LS-SVM, têm de ser cimentados alguns conceitos matemáticos para assim obter uma melhor compreensão das regras a seguir na otimização de problemas não lineares. Posto isto, são apresentadas as condições de otimalidade de 1ª e 2ª ordem no que toca à otimização não linear.

2.1.1. Otimização Não Linear

Em problemas de otimização não linear, a formulação geral do problema é dada pela seguinte expressão [22]:

$$\min_{x \in R^n} f(x) \quad (10)$$

Sujeito a: $c(x) \geq 0$

em que $f(x)$ e $c(x)$ representam funções não lineares em x .

$$x = (x_1, x_2, \dots, x_n)^T \quad (11)$$

$$f(x) : R^n \rightarrow R, c(x) : R^n \rightarrow R^m$$

onde n é o número de variáveis do problema e m o número de restrições de desigualdade.

$$c(x) = (c_1(x), c_2(x), \dots, c_m(x))^T \quad (12)$$

Supondo que $f(x)$ e $c(x)$ são funções duas vezes diferenciáveis.

Seja λ um vector de m elementos (vector dos multiplicadores de Lagrange):

$$\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)^T \quad (13)$$

A função Lagrangeana associada ao problema 10 é:

$$L(x, \lambda) = f(x) - \lambda^T c(x) = f(x) - \sum_{i=1}^m \lambda_i c_i(x) \quad (14)$$

Notas:

Restrições Ativas: Seja \bar{x} um ponto dado. As restrições $c_i(\bar{x}) \geq 0, i \in \mathcal{A}$, dizem-se ativas em \bar{x} se $c_i(\bar{x}) = 0$. O conjunto \mathcal{A} contém os índices das restrições ativas.

Condição de Regularidade: Seja x^* uma solução do problema. Se os vetores $\nabla c_i(x^*), i \in \mathcal{A}$ (gradientes das restrições ativas, calculados na solução) forem linearmente independentes, então x^* é ponto regular.

Condições necessárias e suficientes de 1ª Ordem

Seja x^* uma solução do problema. Se x^* é ponto regular, então existe um λ^* tal que:

$$\nabla_x L(x^*, \lambda^*) = \nabla f(x^*) - \nabla c(x^*)\lambda^* = 0 \text{ - ponto estacionário da função Lagrangeana;}$$

$$c(x^*) \geq 0 \text{ - admissibilidade;}$$

$$\lambda^* \geq 0 \text{ - positividade;}$$

$$(\lambda^*)^T c(x^*) = 0 \text{ - complementaridade.}$$

Interpretação das Condições Karush-Kuhn-Tucker

A complementaridade $((\lambda^*)^T c(x^*) = 0)$ afirma que as restrições não ativas têm multiplicadores iguais a zero $(c_i(x^*) > 0 \Rightarrow \lambda_i^* = 0, i \notin \mathcal{A})$.

Para as restrições ativas, os multiplicadores de Lagrange correspondentes podem ou não ser zero. Se forem zero, trata-se de um problema degenerado. No caso de não existirem multiplicadores iguais a zero para as restrições ativas, trata-se de um caso de complementaridade estrita:

$$c(x^*) \geq 0 \tag{15}$$

o que significa que o ponto verifica as restrições, ou seja, x^* é um ponto admissível.

$$\begin{aligned} \nabla_x L(x^*, \lambda^*) &= \nabla f(x^*) - \nabla c(x^*)\lambda^* = 0 \\ \Leftrightarrow \nabla f(x^*) &= \nabla c(x^*)\lambda^* \\ \Leftrightarrow \nabla f(x^*) &= \sum_{i=1}^m \lambda_i^* \nabla c_i(x^*) \end{aligned} \tag{16}$$

A expressão acima significa que o gradiente de f (∇f) é uma combinação linear dos gradientes das restrições (das colunas de $\nabla c(x^*)$).

Exemplo - Nash & Sofer

Supondo que a formalização do problema é dada por [22]:

$$\min_{x \in \mathbb{R}^2} f(x) \equiv x_1 \tag{17}$$

Sujeito a:

$$\begin{aligned} (x_1 + 1)^2 + x_2^2 &\geq 1 \\ x_1^2 + x_2^2 &\leq 2 \end{aligned} \tag{18}$$

Considerando: $x^1 = (0,0)^T$, $x^2 = (-1,-1)^T$ e $x^3 = (0, \sqrt{2})^T$.

Tem-se que:

$$\begin{aligned} L(x, \lambda) &= x_1 - (\lambda_1, \lambda_2) \begin{pmatrix} (x_1 + 1)^2 + x_2^2 - 1 \\ 2 - x_1^2 - x_2^2 \end{pmatrix} \\ &= x_1 - \lambda_1((x_1 + 1)^2 + x_2^2 - 1) + \lambda_2(x_1^2 + x_2^2 - 2) \end{aligned} \tag{19}$$

Condições Karush-Kuhn-Tucker

A partir da aplicação das condições KKT mencionadas acima, vem:

$$\nabla_x L(x, \lambda) = \begin{pmatrix} 1 - 2\lambda_1(x_1 + 1) + 2\lambda_2 x_1 \\ -2\lambda_1 x_2 + 2\lambda_2 x_2 \end{pmatrix} \tag{20}$$

Para $x^1 = (0,0)^T$ apenas a primeira restrição está ativa, logo $\lambda_2 = 0$. Resolvendo $L(x^1, \lambda)$ em ordem a λ_1 vem:

$$\begin{cases} 1 - 2\lambda_1 = 0 \\ 0 = 0 \end{cases} \Rightarrow \lambda_1 = \frac{1}{2} \quad (21)$$

Para $x^2 = (-1, -1)^T$ ambas as restrições estão ativas, e resolvendo $L(x^2, \lambda)$ em ordem aos multiplicadores de Lagrange, obtém-se:

$$\begin{cases} 1 - 2\lambda_2 = 0 \\ 2\lambda_1 - 2\lambda_2 = 0 \end{cases} \Rightarrow \lambda_1 = \lambda_2 = \frac{1}{2} \quad (22)$$

Para $x^3 = (0, \sqrt{2})^T$ apenas a segunda restrição está ativa e resolvendo $L(x^3, \lambda) = 0$ em ordem a λ_2 ($\lambda_1 = 0$) obtém-se:

$$\begin{cases} 1 + 2\lambda_2(0) = 0 \\ 2\lambda_2\sqrt{2} = 0 \end{cases} \quad (23)$$

que é um sistema inconsistente e conseqüentemente x^3 não satisfaz as condições de optimalidade de primeira ordem.

Condição Necessária de 2ª Ordem - Mínimo

Seja x^* uma solução do problema. Se x^* é ponto regular, então todos os vetores $s \in R^n$ que verificam $\nabla \mathcal{C}(x^*)^T s = 0$, em que $\mathcal{C}(x^*)$ é uma matriz formada pelas restrições ativas em x^* , tem-se:

$$s^T \nabla_{xx}^2 L(x^*, \lambda^*) s \geq 0 \quad (24)$$

Condição Suficiente de 2ª Ordem - Mínimo

Se (x^*, λ^*) é um par KKT (Karush-Kuhn-Tucker), isto é, verifica as condições de Karush-Kuhn-Tucker:

$$\begin{cases} \nabla_x L(x^*, \lambda^*) = 0 \\ \nabla_\lambda L(x^*, \lambda^*) = 0 \end{cases} \quad (25)$$

e se:

$$s^T \nabla_{xx}^2 L(x^*, \lambda^*) s > 0 \quad (26)$$

Para todo o s ($s \neq 0$) tal que:

$$\nabla \mathcal{C}_+(x^*)^T s = 0 \quad (27)$$

então x^* é um minimizante local forte.

\mathcal{C}_+ é uma matriz formada pelas restrições ativas não degeneradas (multiplicadores diferentes de zero).

O exemplo em x^1

$$\nabla_{xx}^2 L(x, \lambda) = \begin{pmatrix} 2(\lambda_2 - \lambda_1) & 0 \\ 0 & 2(\lambda_2 - \lambda_1) \end{pmatrix} \quad (28)$$

Logo para $x^1 = (0,0)^T$ e $\lambda^1 = (\frac{1}{2}, 0)^T$ tem-se:

$$\nabla_{xx}^2 L(x, \lambda) = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \quad (29)$$

Como $\nabla \mathcal{C}(x^1) = (2,0)^T$ tem-se $s = (0, s_2)^T$ e conseqüentemente:

$$(0, s_2) \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ s_2 \end{pmatrix} = -s_2^2 \leq 0 \quad (s_2 \neq 0) \quad (30)$$

Logo x^1 não é mínimo local. Não é máximo local porque $\lambda_1 \geq 0$.

O exemplo em x^2

Como $\nabla \mathcal{C}(x^2) = \begin{pmatrix} 2 & 0 \\ 2\sqrt{2} & -2\sqrt{2} \end{pmatrix}$ tem-se que:

$\nexists s \neq 0 : \nabla \mathcal{C}(x^2)^T s = 0$ e a condição suficiente é verificada trivialmente.

2.2. Método da Regressão através da Máquina de Suporte Vetorial com base nos Mínimos Quadrados

A popularidade que rodeia métodos de regressão lineares deve-se à sua facilidade de utilização. Deste modo, mesmo em casos em que se saiba que as classes não são separáveis linearmente, poderá ser escolhido um classificador linear, independentemente do facto do seu desempenho inferior no ponto de vista do erro de classificação [3].

Apesar das redes neuronais clássicas poderem apresentar o mesmo tipo de aplicações que o método LS-SVM, na prática tornam-se evidentes as suas desvantagens perante o método proposto. Uma vantagem significativa dos SVM é que enquanto as redes neuronais artificiais podem apresentar vários mínimos locais, a solução para um método SVM é única e global, tal como apresentado na Figura 13.

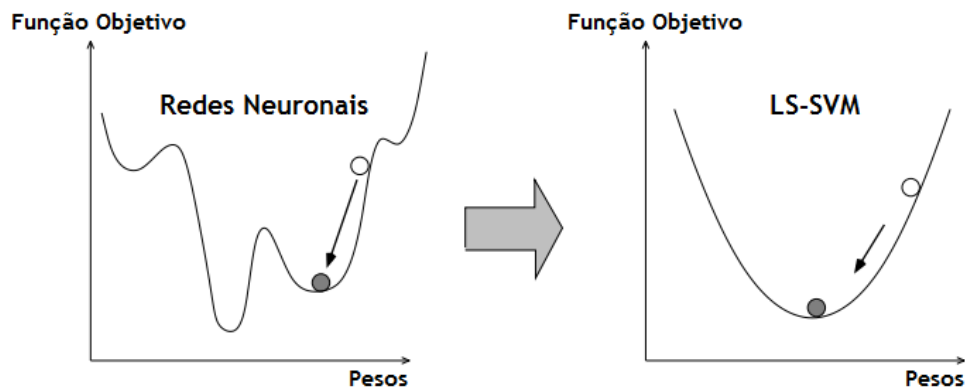


Figura 13 - Obtenção de Mínimos Locais para Redes Neuronais e Mínimo Global para LS-SVM

De acordo com Suykens e Vandewalle [13], a convexão é uma propriedade importantíssima dos LS-SVM quando comparados com as redes neuronais, devido à existência de apenas uma solução mínima. Este fato ocorre porque, neste caso, a estimação é feita através da resolução de um problema de otimização convexo. Além disso, os LS-SVM tendem a ultrapassar o desempenho das redes neuronais devido ao fato de nunca causarem um sobre ajustamento (*overfitting*) dos dados.

A formalização matemática do problema mencionado de seguida baseia-se nos trabalhos feitos por Xu et. al [23] e por Suykens e Vandewalle [13], em que o método de LS-SVM é aplicado num

conjunto de dados experimentais, com a finalidade de testar a precisão deste algoritmo em detrimento de outros métodos de aprendizagem semi-supervisionada.

Supondo um conjunto de dados de treino $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ com $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$ ($i = 1, 2, \dots, m$), onde cada $x_i \in \mathbb{R}^d$ representa um dado de entrada cujo dado de saída (correspondência) é $y_i \in \mathbb{R}$. Sendo $y = (y_1, y_2, \dots, y_m)^T$ [24]. O método LS-SVM inicia com a aproximação de uma função desconhecida, descrita pela seguinte equação:

$$f(x) = w^T \varphi(x) + b \quad (31)$$

onde $\varphi: \mathbb{R}^d \rightarrow F$ representa uma função não-linear que “mapeia” os dados de entrada para um espaço característico de dimensão maior, w representa um valor de coeficientes e b o termo de viés (bias term). O princípio do LS-SVM é estimar os parâmetros w e b que minimizem a função anterior de acordo com o princípio da minimização do risco estrutural proposto por Vapnik [25].

Princípio da Minimização do Risco Estrutural [25]: Este princípio parte da necessidade de selecionar um modelo generalizado a partir de um conjunto de dados finito que pode resultar num problema de sobre ajustamento (overfitting), tornando o modelo pouco preciso para obtenção de novos dados. Logo, o princípio de minimização do risco estrutural resolve este problema, balançando a complexidade do modelo com a eficácia de adaptação aos dados de treino.

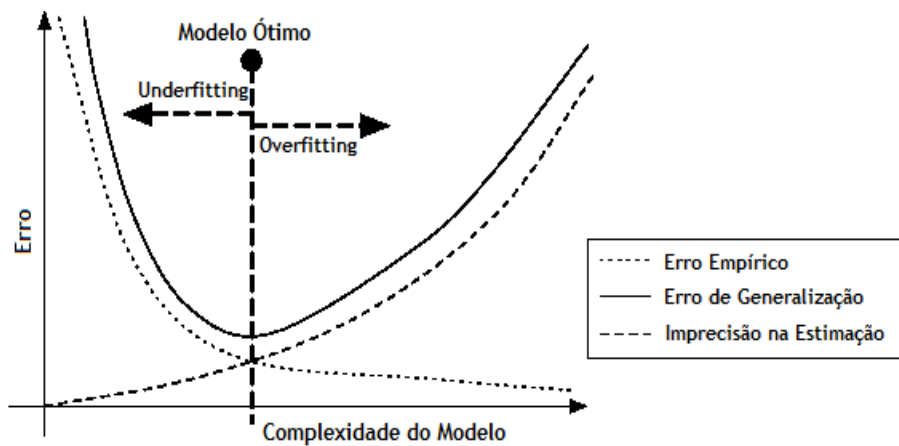


Figura 14 - Princípio da Minimização do Erro de Estimação

Através da Figura 14, é possível observar que a solução ótima do processo de estimação será obtida num compromisso entre a complexidade do modelo e uma minimização do erro empírico e do erro de generalização.

O problema de otimização do LS-SVM pode ser formalmente definido como:

$$\min_{w \in \mathbb{R}^n, b \in \mathbb{R}, \xi \in \mathbb{R}^m} J(w, b, \xi) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \xi^T \xi \quad (32)$$

$$\text{s.t. } y_i = w^T \varphi(x_i) + b + \xi_i, i = 1, 2, \dots, m \quad (33)$$

onde $\varphi: \mathbb{R}^d \rightarrow \mathbb{R}^{n_h}$ é um “mapeamento” para um espaço característico maior \mathcal{H} , cada componente de $\xi = (\xi_1, \xi_2, \dots, \xi_m)^T$ é uma variável desprezável e $\gamma > 0$ uma constante real que é utilizada para controlar o grau de punição para uma má classificação [26]. Em situações em que os dados contenham muito ruído, é indicado assumir o valor γ como sendo um valor baixo.

O objetivo de encontrar um mínimo na Equação 32 advém do fato de o vetor w ser considerado o vetor de pesos na função objetivo. Esta minimização consiste em encontrar o valor mínimo que surge do resultado do seu produto interno (produto escalar) $w^T w$. Esta minimização fará com que as dimensões dos pesos sejam regularizadas, penalizando valores de pesos elevados. A finalidade deste processo tem explicação no fato de os pesos de valores elevados perturbarem a generalização do método LS-SVM, podendo causar uma variância excessiva.

A solução do problema de otimização expresso pelas Equações 32-33 vai de acordo com as condições de *Karush-Kuhn-Tucker* (KKT). Estas condições representam uma generalização do método dos multiplicadores de Lagrange e são necessárias para que uma solução em problemas de programação não-linear seja ótima [27]:

Teorema Karush-Kuhn-Tucker (KKT): em problemas de otimização matemática, as condições KKT representam um conjunto de regras necessárias para a obtenção de uma solução ótima num problema de programação, desde que algumas condições sejam satisfeitas.

Partindo da função objetivo definida pela Equação 32, o problema formaliza-se da seguinte forma:

$$\bullet \min_{w \in \mathbb{R}^{n_h}, b \in \mathbb{R}, \xi \in \mathbb{R}^m} J(w, b, \xi) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \xi^T \xi \quad (34)$$

Sujeito a:

$$y_i = w^T \varphi(x_i) + b + \xi_i, \quad (35)$$

A solução ótima pode ser encontrada através do multiplicador de Lagrange:

$$L = (w, b, \xi, \alpha) = J(w, b, \xi) - \sum_{i=1}^m \alpha_i \{w^T \varphi(x_i) + b + \xi_i - y_i\} \quad (36)$$

onde α_i representa o multiplicador de Lagrange de ordem i . Neste problema de otimização, que contém apenas restrições de igualdade, as condições KKT a serem respeitadas são as seguintes:

Assumindo que $w^* = (w_1^*, w_2^*, \dots, w_n^*)$ minimiza $J(w, b, \xi)$, sujeito às restrições y_i , para $i = 1, 2, \dots, m$, então verifica-se uma das condições:

1. Os vetores $\nabla y_1(x^*), \nabla y_2(x^*), \dots, \nabla y_m(x^*)$ são linearmente dependentes;
2. Existe um vetor $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*)$ tal que $\nabla L(w^*, b^*, \xi^*, \alpha^*) = 0$, ou seja:

$$\begin{aligned} \frac{\delta L}{\delta w} &= w - \sum_{i=1}^m \alpha_i \varphi(x_i) = 0 \Rightarrow w = \sum_{i=1}^m \alpha_i \varphi(x_i) \\ \frac{\delta L}{\delta b} &= - \sum_{i=1}^m \alpha_i = 0 \Rightarrow \sum_{i=1}^m \alpha_i = 0 \end{aligned} \quad (37)$$

$$\frac{\delta L}{\delta \xi_i} = \gamma \xi_i - \alpha_i = 0 \Rightarrow \alpha_i = \gamma \xi_i$$

$$\frac{\delta L}{\delta \alpha_i} = w^T \varphi(x_i) + b + \xi_i - y_i = 0 \Rightarrow y_i = w^T \varphi(x_i) + b + \xi_i$$

Recorrendo às expressões na Equação 37, e eliminando os parâmetros w e ξ_i da equação, obtém-se a seguinte matriz:

$$\begin{bmatrix} 0 & e^T \\ e & K + \gamma^{-1}I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \quad (38)$$

onde $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)^T$ representa um vector que contém os multiplicadores de Lagrange (também chamados *support vectors*), $e = (1, 1, \dots, 1)^T$, I representa a matriz identidade e $K_{i,j} = K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ ($i, j = 1, 2, \dots, m$) onde $K(\cdot, \cdot)$ representa uma função Kernel que vai de acordo com o teorema de Mercer [9].

Teorema de Mercer: *Sejam $K \in L^2(C)$, $g \in L^2(C)$ onde C é um sub-conjunto compacto de \mathbb{R}^d , $K(t, z)$ o produto interno entre t e z e $t, z \in \mathbb{R}^d$. Para garantir que uma função contínua e simétrica K tem uma expansão:*

$$K(t, z) = \sum_{i=1}^{\infty} \alpha_i \phi_i(t) \phi_i(z) \quad (39)$$

com os coeficientes $\alpha_i > 0$. Então é necessário (e suficiente) que a condição:

$$\iint_C K(t, z) g(t) g(z) dt dz \geq 0 \quad (40)$$

seja válida para todo $g \in L^2(C)$.

Utilizando a condição de Mercer, é possível escrever $K(t, z) = \sum_{i=1}^{n_f} \sqrt{\alpha_i} \phi_i(t) \sqrt{\alpha_i} \phi_i(z)$ e definir $\varphi_i(t) = \sqrt{\alpha_i} \phi_i(t)$ e $\varphi_i(z) = \sqrt{\alpha_i} \phi_i(z)$ de tal modo que a função kernel possa ser expressa como o produto interno:

$$K(t, z) = \varphi(t)^T \varphi(z) \quad (41)$$

Assim, definindo um kernel positivo é uma condição que garante que a equação (41) é válida.

Nota: Para a resolução do sistema representado pela Equação 38 é necessário recorrer ao cálculo de uma matriz pseudo-inversa. Isto deve-se ao fato do sistema linear não possuir apenas uma solução. A matriz pseudo-inversa procura a solução mínima no espaço Euclidiano. Mais informações acerca da matriz pseudo-inversa podem ser encontradas no Anexo III.

Para finalizar a resolução do sistema descrito pela Equação 38, supondo que $H = K + \gamma^{-1}I$. São obtidas as seguintes equações:

$$\begin{cases} e^T \alpha = 0 \\ eb + H\alpha = y \end{cases} \quad (42)$$

Assim, as variáveis α e b são obtidas apenas através de:

$$\begin{cases} \alpha = H^{-1}(y - eb) \\ b = e^T H^{-1} y (e^T H^{-1} e)^{-1} \end{cases} \quad (43)$$

Se a solução do sistema linear da Equação 38 for $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_m^*)^T$ e b^* , então a função objetivo (modelo de regressão) que permitirá estimar um dado x não classificado será:

$$\hat{y}(x) = f(x) = w^{*T} \varphi(x) + b^* = \sum_{i=1}^m \alpha_i^* \varphi(x_i)^T \varphi(x) + b^* = \sum_{i=1}^m \alpha_i^* K(x_i, x) + b^* \quad (44)$$

2.2.1. Lista de Funções Kernel Aplicadas

Kernel Gaussiano

Tal como mencionado anteriormente, o kernel Gaussiano é um exemplo de uma função de base radial, que se traduz da seguinte forma:

$$h(x) = \exp\left(-\frac{\|x - x_n\|^2}{2\sigma^2}\right) \quad (45)$$

O parâmetro sigma σ tem um papel bastante importante no desempenho do kernel, e deverá ser cuidadosamente escolhido tendo em conta o problema em que o kernel se insere. Este parâmetro será também fundamental para algumas das funções kernel que se seguem.

Kernel Exponencial

Este kernel está bastante relacionado com o kernel Gaussiano descrito anteriormente, na medida em que também faz parte do grupo das funções de base radial. A única diferença reside no fato de a distância entre o ponto x e x_n não estar elevada ao quadrado. Assim, o kernel Exponencial pode ser escrito do seguinte modo:

$$h(x) = \exp\left(-\frac{\|x - x_n\|}{2\sigma^2}\right) \quad (46)$$

Apesar do parâmetro sigma ter de ser ajustado da mesma forma que para o kernel Gaussiano, não poderá ser utilizado o mesmo valor, visto que as funções diferem no resultado.

Kernel de Laplace

O kernel de Laplace é idêntico ao kernel Exponencial, com a diferença de ser menos sensível a alterações do parâmetro sigma. Posto isto, é possível concluir que o kernel de Laplace também representa uma função de base radial. É definido da seguinte maneira:

$$h(x) = \exp\left(-\frac{\|x - x_n\|}{\sigma}\right) \quad (47)$$

Para este kernel, e para os dois mencionados anteriormente, tem de ser feito um estudo iterativo para determinar o valor ideal para o parâmetro sigma, antes de proceder à simulação completa.

Kernel Rational Quadratic

O kernel *Rational Quadratic* é utilizado como alternativa ao kernel Gaussiano, visto que a sua carga computacional é inferior. Traduz-se pela expressão:

$$h(x) = 1 - \frac{\|x - x_n\|^2}{\|x - x_n\|^2 + \eta} \quad (48)$$

O parâmetro η , também presente nas funções kernel seguintes, depende da dimensão do conjunto de dados D que irá ser utilizado na previsão de soluções.

Kernel Multiquádrico

O kernel Multiquádrico pode ser utilizado nas mesmas circunstâncias que o kernel *Rational Quadratic*. A sua expressão é a seguinte:

$$h(x) = \sqrt{\|x - x_n\|^2 + \eta} \quad (49)$$

Foi desenvolvido e estudado por Michelli, em 1996, e é tido em conta como uma das funções kernel com melhor desempenho na previsão de resultados.

Kernel Multiquádrico Inverso

Este kernel é em tudo semelhante ao kernel Multiquádrico, mas toma a sua forma inversa, ou seja:

$$h(x) = \frac{1}{\sqrt{\|x - x_n\|^2 + \eta}} \quad (50)$$

2.2.2. Implementação de Erro Gaussiano (Distribuição Normal)

Com o intuito de aproximar as trajetórias criadas pelo algoritmo a dados reais, após a geração das coordenadas geocêntricas para a posição da aeronave, foi implementado um erro gaussiano a cada uma delas (X, Y, Z) . A criação deste erro gaussiano aleatório pode ser justificada pelo teorema de Box-Muller, publicado em 1958. Para uma melhor compreensão deste teorema é feita uma pequena abordagem teórica ao mesmo nos parágrafos que se seguem.

Coordenadas Cartesianas e Polares

Antes de ser explicada a transformada de Box-Muller, é necessário estabelecer alguns conceitos sobre a relação entre coordenadas cartesianas e coordenadas polares. Se X e Y representarem dois pontos no plano cartesiano, estes podem ser representados com coordenadas polares com um raio r e um ângulo θ utilizando as seguintes relações [28]:

$$\begin{aligned} R^2 &= X^2 + Y^2 \\ \tan(\theta) &= \frac{Y}{X} \end{aligned} \quad (51)$$

logo,

$$\begin{aligned} X &= R \cos(\theta) \\ Y &= R \sin(\theta) \end{aligned} \quad (52)$$

O desenvolvimento das expressões anteriores pode ser consultado em maior detalhe no Anexo IV.

Se se verificar a particularidade de $R \leq 1$ e $\theta \in [0, 2\pi]$, então é possível representar os valores no círculo trigonométrico, tal como observado na figura abaixo.

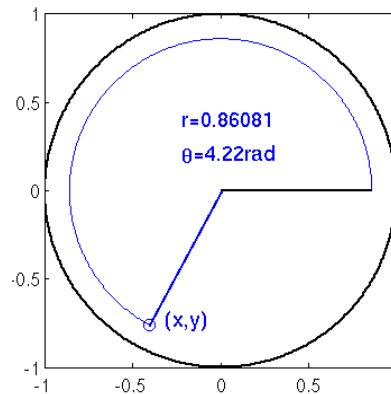


Figura 15 - Relação entre coordenadas Cartesianas e coordenadas Polares

Transformada Box-Muller

A abordagem seguinte pode ser utilizada para gerar um par de números aleatórios, que pertencem à mesma distribuição normal, partindo de dois valores aleatórios. Sejam U_1 e U_2 duas variáveis independentes e aleatórias, que seguem uma distribuição uniforme no intervalo $(0,1)$ (Nota: uma distribuição uniforme, significa que a distribuição da sua probabilidade é contínua e uniforme). Partindo destas variáveis, é efetuado o seguinte processo:

$$\begin{aligned} X &= \sqrt{-2 \ln(U_2)} \sin(2\pi U_1) \\ Y &= \sqrt{-2 \ln(U_2)} \cos(2\pi U_1) \end{aligned} \quad (53)$$

Graças a esta transformação, estudada por Box e Muller em 1958 [29], é possível assumir que as variáveis X e Y representam um conjunto de valores aleatórios que se encontram na mesma distribuição normal, com média $\mu = 0$ e variância $\sigma^2 = 1$. A dedução das expressões anteriores pode ser consultada no Anexo IV.

2.3. Localização Aeroespacial

Numa situação de voo, uma aeronave depende de vários instrumentos para determinar a sua posição e os seus parâmetros de voo, tais como, altitude, velocidade, rumo, ângulo de trajetória, etc. Apesar de os dados utilizados terem sido criados a partir de um algoritmo de geração de trajetórias, que irá ser desenvolvido no tópico seguinte, são explicitados alguns métodos de aquisição de dados de voo nos parágrafos seguintes.

2.3.1. Sistema de Navegação por Satélite - GNSS

A navegação através de satélites, ou *satnav* representa um conjunto de satélites que fornecem um posicionamento geo-espacial e que apresentam uma cobertura global. Dada a sua cobertura global, podem adotar o nome de sistema de navegação global por satélite (GNSS - *Global Navigation Satellite System*). Os únicos dois sistemas GNSS operacionais atualmente são detidos pelos Estados Unidos - GPS e pela Rússia - GLONASS.

O Sistema de Posicionamento Global (GPS - *Global Positioning System*) representa um sistema de navegação baseado em satélites que foi desenvolvido pelo Departamento de Defesa dos Estados Unidos da América. Inicialmente, o GPS foi desenvolvido com a intenção de complementar as necessidades militares dos Estados Unidos. No entanto, expandiu-se ao mundo civil, podendo agora ser utilizado por utilizadores militares ou civis. Essencialmente, o GPS consiste numa constelação de 24 satélites operacionais que iniciou as suas funções em Julho de 1993. Para assegurar uma cobertura global contínua, os satélites GPS estão dispostos de tal forma que quatro satélites estejam colocados em cada um dos seis planos orbitais. Através desta configuração, entre quatro a dez satélites GPS são sempre visíveis em qualquer local do globo.



Figura 16 - Constelação GPS [30]

Cada satélite transmite, continuamente, um sinal de rádio micro-ondas composto por dois portadores, dois códigos e uma mensagem de navegação. Quando um recetor de GPS é ligado, este irá captar o sinal GPS através da sua antena. Assim que o recetor adquire o sinal de GPS, irá processá-lo utilizando o seu *software*. O resultado do processamento de sinal consiste nas distâncias até aos satélites GPS através dos códigos digitais (também conhecidas como pseudo-distâncias) e as coordenadas do satélite através da mensagem de navegação.

Teoricamente, seriam apenas necessárias três distâncias a três satélites localizados simultaneamente. Neste caso, o recetor seria localizado na interseção de três esferas, cada uma com um raio igual a uma das distâncias do recetor ao satélite. No entanto, a nível prático é necessária a utilização de um quarto satélite para ter em conta o *offset* do relógio do recetor.

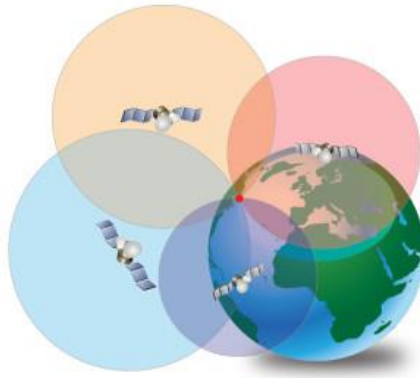


Figura 17 - Conceito básico do posicionamento GPS [30]

Em março de 2002, a União Europeia e a Agência Espacial Europeia introduziram uma alternativa para o sistema GPS, com o nome de *Galileo*. Um dos objetivos principais do sistema Galileo é a criação de um sistema de posicionamento de alta precisão que esteja ao dispor das nações europeias. Espera-se que as medições deste sistema estejam dentro de um intervalo de precisões de 1 metro, fornecendo um melhor serviço de posicionamento que qualquer outro.

EGNOS - European Geostationary Navigation Overlay Service

O sistema EGNOS, desenvolvido pela Agência Espacial Europeia (ESA, pela Comissão Europeia e pela EUROCONTROL, representa um suplemento aos sistemas GPS, GLONASS e Galileo. Este sistema é constituído por um conjunto de quatro satélites geoestacionários e uma rede de estações terrestres. A nível prático, o sistema demonstra precisões de valores abaixo de um metro. Também existem serviços semelhantes nos outros continentes, como por exemplo o sistema WAAS - *Wide Area Augmentation System* e o MSAS - *Multi-functional Satellite Augmentation System* na América do Norte e na Ásia, respetivamente.

2.3.2. Funcionamento do Sistema ADS-B

Atualmente, a vigilância do espaço aéreo conta com os métodos tradicionais de controlo de tráfego aéreo, ou seja, com a utilização de radares primários e secundários. Os procedimentos de vigilância do controlo de tráfego aéreo representam uma técnica de vigilância dependente, que necessita que os pilotos reportem as suas posições através de canais de rádio, o que pode está sujeito a conter erros humanos [31].

O sistema ADS-B foi concebido para ultrapassar as técnicas convencionais de controlo de tráfego aéreo, bem como fornecer novas capacidades que irão fortalecer a gestão do espaço aéreo. Além disso, o ADS-B acarreta um aumento na precisão quando comparado com os radares convencionais, que não se deteriora com o aumento da distância ao recetor. A tabela seguinte revela uma comparação geral entre os sistemas de radar atuais e os atributos do ADS-B.

Tabela 1 - Sistemas de Radar convencionais vs. Sistema ADS-B

Sistema de Radar Atual	Sistema ADS-B
Sistema terrestre, dependente da participação do Homem	Sistema na aeronave e fornece um fluxo constante de dados da posição mais precisos
Falhas de cobertura em algumas áreas	Estações terrestres de ADS-B podem ser colocadas em qualquer lado
Posições são atualizadas a cada 12 segundos	Posições são atualizadas a cada segundo
Manutenção e instalação cara	Manutenção e instalação mais barata

Este sistema utiliza uma combinação de satélites de navegação global (GNSS - *Global Navigation Satellite System*), transmissores e recetores, para fornecer informações específicas acerca da localização e velocidade das aeronaves a sobrevoar a zona, à tripulação e ao controlo terrestre.

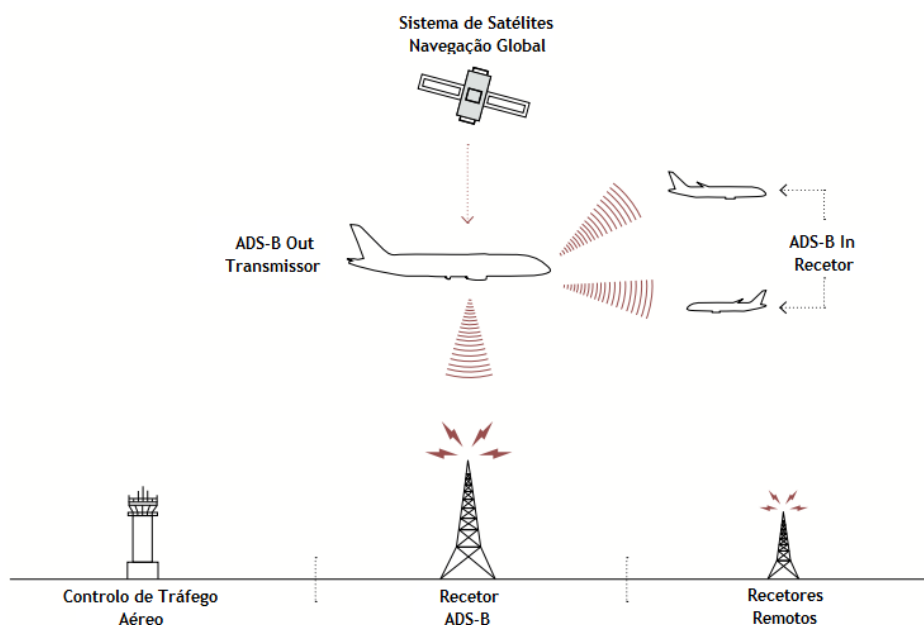


Figura 18 - Funcionamento do ADS-B [31]

A sua arquitetura pode ser dividida em dois subsistemas, o “ADS-B Out” e o “ADS-B In”. O primeiro subsistema é definido como a transmissão da informação do sistema ADS-B da aeronave que depois é recebida por controladores de tráfego aéreo no solo, permitindo uma vigilância terrestre da aeronave. O subsistema complementar “ADS-B In” permite que outras aeronaves recolham a informação transmitida pelo “ADS-B Out”, possibilitando um maior alerta para proximidades entre aeronaves no ar [17].

2.3.3. Sistemas de Navegação Inercial (INS)

O princípio geral da navegação inercial baseia-se no conceito simples de estimação de navegação, ou *dead reckoning*. Considerando um acelerómetro, como um instrumento que mede a aceleração da aeronave ao longo de um determinado eixo. Se se integrar o resultado

da leitura do acelerómetro uma vez, obtém-se a velocidade da aeronave. Integrando novamente, obtém-se uma variação da posição da aeronave ao longo do eixo do acelerómetro. Partindo da direção da trajetória, é possível obter a posição atual [32]. A navegação inercial difere da navegação radio na medida em que não utiliza qualquer referência externa. Recorre apenas à posição inicial da aeronave, velocidade e altitude, e depois na medição das taxas de atitude e acelerações do veículo, dependendo apenas das leis de mecânica clássica de Newton. O funcionamento de um sistema INS utiliza um conjunto de três acelerómetros, em que os seus eixos estão orientados segundo: norte-sul, este-oeste e verticalmente. Para que os acelerómetros mantenham a sua orientação quando a aeronave manobra, estes encontram-se suspensos num conjunto de três giroscópios que estão sempre orientados com os eixos dos acelerómetros. Esta estrutura pode ser analisada na figura seguinte.

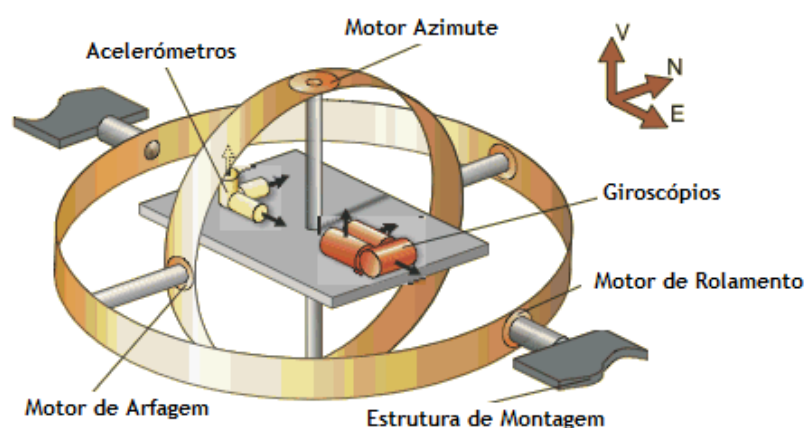


Figura 19 - Estrutura básica de um sistema INS [32]

Os giroscópios fornecem um dado de saída proporcional ao ângulo sob o qual foram rodados, que depois é utilizado para movimentar o motor respetivo através de um servo, mantendo o sistema com uma orientação constante.

GPS/INS

A utilização de sinais provenientes de satélites GPS é bastante comum na calibração das soluções dos sistemas INS. Os sistemas de navegação inercial fornecem uma solução precisa apenas para um curto espaço de tempo, visto que os acelerómetros tendem a produzir um pequeno erro de leitura, que ao ser integrado duplamente, resulta num erro de posição. O GPS fornece um valor para a posição sem oscilações, que pode ser utilizado para restabelecer a solução do sistema INS, ou que pode ser misturada através da aplicação de algoritmos matemáticos, tais como os filtros de Kalman. A vantagem da utilização destes sistemas em simultâneo, é dada pelo fato de as soluções do sistema INS poderem ser calibradas pelos sinais GPS, ao mesmo tempo que o sistema INS fornece uma atualização da posição e do ângulo de trajetória da aeronave mais rápida que o GPS [33].

2.3.4. Rumor de uma Aeronave (ψ)

Essencialmente, o rumor de uma aeronave (em inglês - *heading*) ψ indica a direção na qual o nariz da aeronave aponta. O seu valor é calculado através da utilização de uma bússola, cuja referência está baseada no norte magnético, e é apresentado no painel de instrumentos através do indicador de rumor.

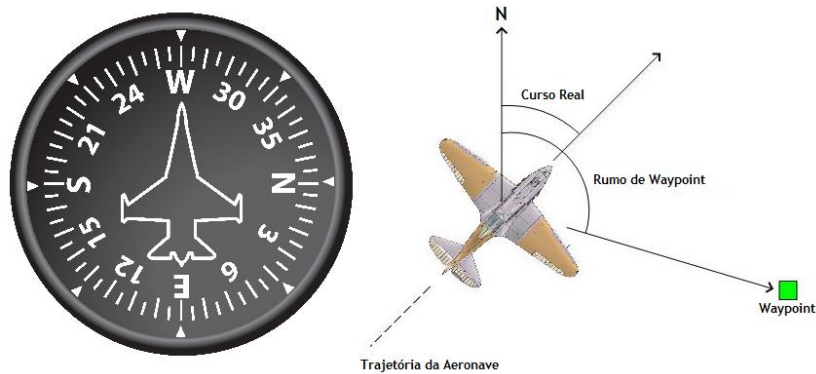


Figura 20 - Indicador de Rumor e Rumor em relação a um Waypoint

No entanto, para evitar a considerável quantidade de erros presentes nos resultados dados por uma bússola, o indicador de rumor opera utilizando um mecanismo de giroscópios. Sempre que a aeronave efetua uma manobra, o giroscópio presente no indicador de rumor irá resistir à mudança de direção, movendo a energia gerada através dessa resistência ao movimento para a bússola contida no indicador. Este processo faz com que o resultado não seja afetado por perturbações magnéticas.

2.3.5. Ângulo de Trajetória de uma Aeronave (γ)

O ângulo de trajetória de uma aeronave (em inglês - *flight path angle*) γ , representa a amplitude entre o eixo do horizonte e o eixo da trajetória da aeronave, ou o vetor da velocidade. A diferença entre o ângulo de trajetória e os ângulos de ataque e de arfagem, pode ser analisada visualmente na figura seguinte.

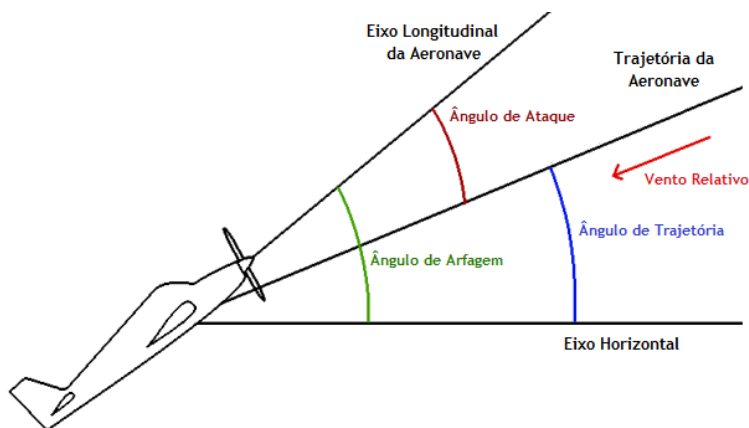


Figura 21 - Ângulo de Trajetória de uma Aeronave

É através da medição da amplitude deste ângulo que se pode determinar se a aeronave se encontra numa fase de subida ou numa fase de descida. Em aeronaves comerciais mais recentes, este ângulo pode ser consultado no painel de voo principal, e é calculado com referência ao solo.

2.3.6. Velocidade da Aeronave

A velocidade de voo é uma medida da velocidade da aeronave relativamente à velocidade do ar que a rodeia. Um sistema de tubo de pitot estático é utilizado para calcular a velocidade relativa da aeronave. Este engenho funciona como um medidor de pressão relativo, criado por Henry Pitot em 1732.

Tal como o altímetro e o indicador de velocidade vertical, o velocímetro faz parte do sistema estático de pitot. A abertura do tubo de pitot, normalmente montado na asa da aeronave, está direccionada de acordo com o escoamento do ar que a rodeia. O indicador da velocidade mede a diferença de pressão entre um sensor estático que está colocado fora da corrente do escoamento, com um sensor colocado no escoamento do ar (tubo de pitot). A esquemática de um tubo de pitot utilizado na mediação da velocidade da aeronave, pode ser analisado na figura seguinte.

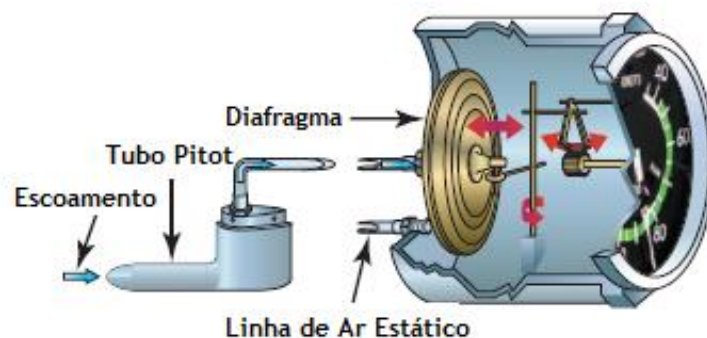


Figura 22 - Sistema Tubo de Pitot -> Velocímetro

Quando a aeronave está parada, a pressão nos tubos é idêntica, fazendo com que a velocidade da aeronave seja nula. Um aumento na velocidade cria um aumento de pressão no tubo de pitot. Em contrapartida, a pressão do ar move uma membrana flexível que se encontra ligada a um ponteiro mecânico, na face do indicador. O indicador encontra-se calibrado para compensar o efeito de correntes de ar no escoamento medido. A nível electrónico, também existem compensações para diferenciais de altitude e temperaturas do ar, para manter o nível de precisão ao máximo.

O valor da velocidade obtido diretamente do tubo de pitot tem o nome de: velocidade indicada (IAS - *Indicated Airspeed*), que não é necessariamente igual à velocidade própria da aeronave (TAS - *True Airspeed*). A diferença entre estes valores da velocidade deve-se ao facto de existir uma compressibilidade do ar para velocidades acima dos 200 nós, das imprecisões dos

instrumentos, das condições de medição (sonda, tubo de pitot, antena, etc.) e da densidade relativa do ar na altitude de voo.

Inicialmente, o valor de velocidade indicado (IAS) é o valor apresentado no indicador do tubo de pitot, calibrado apenas para refletir efeitos normais da compressibilidade atmosférica do ar, não tendo em conta os erros instrumentais ou de posição da aeronave. Este valor é bastante útil, na medida em que muitas das propriedades aerodinâmicas (sustentação, arrasto, velocidade de perda, etc.) dependem da pressão dinâmica gerada pelo escoamento do ar, e não pela velocidade verdadeira da aeronave.

A partir do valor apresentado acima, é calculado o valor da velocidade convencional (CAS - *Conventional Airspeed* ou *Calibrated Airspeed*). Este valor da velocidade tem em conta os erros dos instrumentos, os erros posicionais e os erros de instalação, e traduz-se pela seguinte expressão: $CAS = IAS + \varepsilon_i$, onde ε_i é o erro instrumental. Para algumas aeronaves, é necessário ter em conta a precisão da antena, representada pelo coeficiente k_a : $CAS = k_a \cdot (IAS + \varepsilon_i)$.

De seguida, o objetivo é calcular a velocidade equivalente (EAS - *Equivalent Airspeed*). Este valor é definido como sendo a velocidade ao nível do mar que produziria a mesma pressão dinâmica incompressível que a velocidade verdadeira da aeronave à altitude de voo em que se encontra. A uma altitude elevada, a compressibilidade do ar varia, fazendo com que o valor de CAS seja cada vez menos preciso. O coeficiente de compressibilidade k_c apresenta o valor de $k_c = 1$ se $CAS < 200$ nós, caso contrário: $k_c = \frac{1}{100} \left(102.5 - \frac{Z \times CAS}{12} \cdot 10^{-5} \right)$, onde Z é a altitude em pés e CAS a velocidade convencional em nós.

Por fim, a velocidade verdadeira da aeronave (TAS - *True Airspeed*) representa o valor calculado anteriormente, mas corrigido para a pressão e temperatura à altitude a que se encontra, ou seja, tendo em conta a densidade relativa do ar à altitude atual: $TAS = \frac{EAS}{\sigma}$, onde σ é a densidade relativa do ar.

No capítulo seguinte, irão ser demonstrados todos os resultados obtidos após a aplicação e validação dos métodos acima descritos. Devido à enorme quantidade de dados obtidos, muitos não puderam ser colocados nesta dissertação. No entanto, é feita uma análise bastante detalhada de todos os pormenores a retirar.

Capítulo 3

3. Simulação e Validação em Previsão de Trajetórias

3.1. Exemplo de Previsão de Funções

A título de exemplo, e para uma melhor compreensão do método a implementar, foi feita uma abordagem mais simples ao problema a ser analisado nesta dissertação. Assim, foi utilizada uma função matemática menos complexa para demonstrar os resultados que podem ser obtidos com o método LS-SVM. A função escolhida foi a função seno, e o formato utilizado foi o seguinte:

$$y(x) = Am \times \sin(f \cdot x) \quad (54)$$

onde Am representa a amplitude do sinal e f representa a frequência do sinal.

Nesta simulação, a amplitude utilizada teve um valor de $Am = 12$ e a frequência $f = 0.4 \text{ s}^{-1}$. Assim, a equação acima toma a seguinte forma:

$$y(x) = 12 \times \sin(0.4x) \quad (55)$$

Onde x representa um conjunto de pontos aleatórios criado pelo programa. Para esta simulação foram utilizados um total de 400, como poderá ser observado nos resultados obtidos. Por outras palavras, o sinal x pode ser escrito como: $x = [x_1, x_2, x_3, \dots, x_{400}]$.

Adicionalmente, e para ir de acordo com o tema abordado no Capítulo 2.4, foi injetado um erro de tipo gaussiano na função representada anteriormente. Deste modo, é possível fazer uma previsão do comportamento do algoritmo quando confrontado com um sinal que contém alguns erros nas suas medições. Tal como mencionado anteriormente, o erro gaussiano segue uma distribuição normal com média $\mu = 0$ e uma variância $\sigma^2 = 1$. Logo, o sinal de erro gaussiano toma a seguinte forma, $a \sim N(\mu, \sigma) \Leftrightarrow a \sim N(0,1): a = [a_1, a_2, a_3, \dots, a_{400}]$. . Posto isto, a função final analisada foi:

$$y(x) = 12 \times \sin(0.4x) + a \quad (56)$$

onde a representa um vetor composto por variáveis aleatórias, com uma distribuição normal e com a mesma dimensão que o sinal original x (neste caso, 400 pontos).

Os dados foram analisados por todos os modelos das funções RBF descritas no Capítulo 2.2.1. Os erros obtidos para cada uma das funções, tendo em conta o parâmetro regularizador utilizado, podem ser observados na Tabela 2. Os erros máximos e mínimos obtidos estão representados a vermelho e a verde, respetivamente. Nas funções RBF: Gaussiana, Exponencial e Laplaciana, é também assinalado o valor do parâmetro σ para o qual os resultados foram obtidos.

Tabela 2 - Erros obtidos na previsão da função seno

Função Multiquádrica

Parâmetro Regularizador (Gamma)	Erro Médio (RMSE)
1200	0,16800441
1440	0,142530039
1728	0,12060812
2073,6	0,101828698
2488,32	0,085805501
2985,984	0,072181563

Função Multiquádrica Inversa

Parâmetro Regularizador (Gamma)	Erro Médio (RMSE)
1200	2,388292502
1440	2,374837435
1728	2,360424202
2073,6	2,344996668
2488,32	2,328511473
2985,984	2,310940492

Função Rational Quadratic

Parâmetro Regularizador (Gamma)	Erro Médio (RMSE)
1200	1,100821992
1440	1,056211793
1728	1,01106507
2073,6	0,965575386
2488,32	0,919975292
2985,984	0,874525557

Função Gaussiana ($\sigma = 2$)

Parâmetro Regularizador (Gamma)	Erro Médio (RMSE)
1200	0,00749673
1440	0,00625577
1728	0,0052191
2073,6	0,004353422
2488,32	0,00363077
2985,984	0,00302768

Função Exponencial ($\sigma = 0,8$)

Parâmetro Regularizador (Gamma)	Erro Médio (RMSE)
1200	0,005790789
1440	0,00482624
1728	0,004022272
2073,6	0,003352174
2488,32	0,002793674
2985,984	0,002328197

Função Laplaciana ($\sigma = 0,4$)

Parâmetro Regularizador (Gamma)	Erro Médio (RMSE)
1200	0,007397705
1440	0,006165609
1728	0,005138601
2073,6	0,00428258
2488,32	0,003569103
2985,984	0,002974452

Dos resultados acima apresentados, é possível concluir que a função Exponencial demonstrou um erro inferior quando comparada com as outras funções. No entanto, é preciso reforçar que esta função requer uma maior afinação dos parâmetros a serem utilizados, visto que, para além da escolha ideal do parâmetro regularizador γ , é também necessário conciliar essa escolha com uma procura de um parâmetro σ ideal.

Graficamente, pode ser feita uma comparação entre a função original, descrita pela Equação 65, e a função prevista pelo algoritmo:

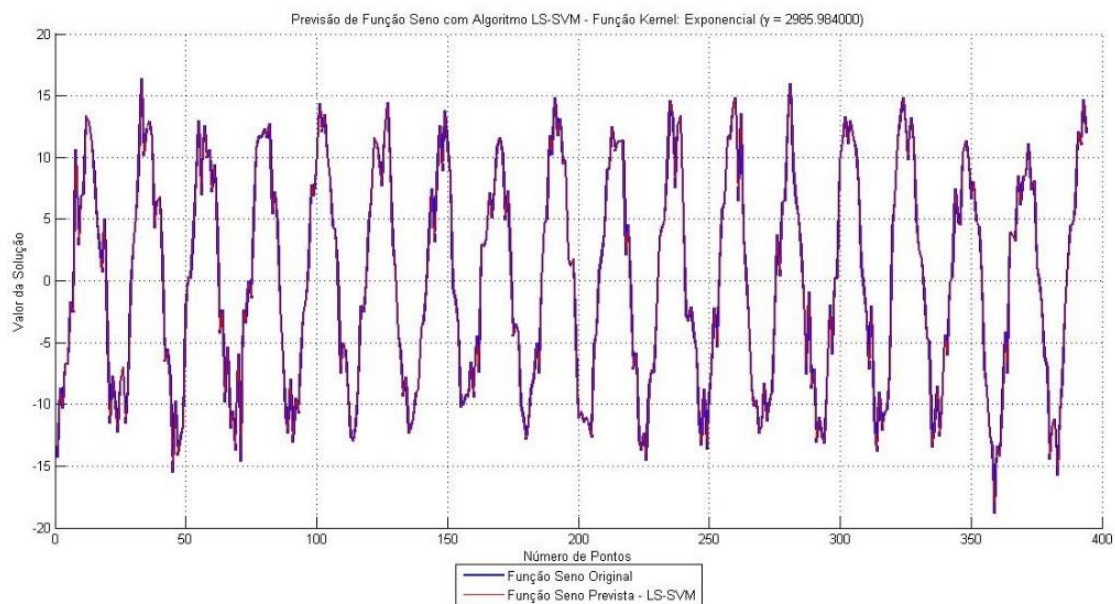


Figura 23 - Comparação da Função Seno Original com a Função Seno Prevista

A figura anterior demonstra a previsão feita com um modelo RBF de uma função Exponencial, para um parâmetro regularizador $\gamma = 2985,984$ e para um $\sigma = 0.4$, devido ao fato de ter sido a que obteve um erro inferior quando comparada com os outros modelos. Tal como mencionado na legenda da Figura 24, a função original é descrita pela linha azul, enquanto que a função prevista pelo método LS-SVM está representada a vermelho. É possível concluir que a função prevista pelo algoritmo LS-SVM é quase coincidente com a função original. O que facilmente pode ser comprovado pelo valor dos erros obtidos na Tabela 2. A vantagem da aplicação deste método é a capacidade de prever resultados para dados de entrada para os quais não havia dados de saída (classificações) anteriormente.

3.2. Estrutura do Programa Implementado

Para uma melhor compreensão do método que foi desenvolvido ao longo deste trabalho, foi criado um organograma hierárquico com a sequência de tarefas a serem executadas pelo programa criado, que pode ser consultado na Figura 24.

Inicialmente, é criada a trajetória que se pretende estudar tendo em conta a aeronave que irá percorrer essa rota. Em seguida, é escolhido o modelo RBF a ser utilizado, dentro das seis funções definidas no Capítulo 2.2.1. (Multiquádrica, Multiquádrica Inversa, Rational Quadratic, Gaussiana, Exponencial e Laplaciana). Após ter sido definida a função RBF, é definida a dimensão da janela de dados a analisar, sabendo que foram estudadas seis dimensões diferentes (de dimensão 3 a dimensão 8).

Posteriormente, será feito um desenvolvimento detalhado acerca da temática da dimensionalização da janela de dados. Por fim, o programa prevê as soluções da trajetória escolhida para um conjunto de dez amplitudes de erro gaussiano diferentes. Tal como para a

reestruturação dos dados, a questão da implementação do erro gaussiano será desenvolvida nos próximos tópicos.

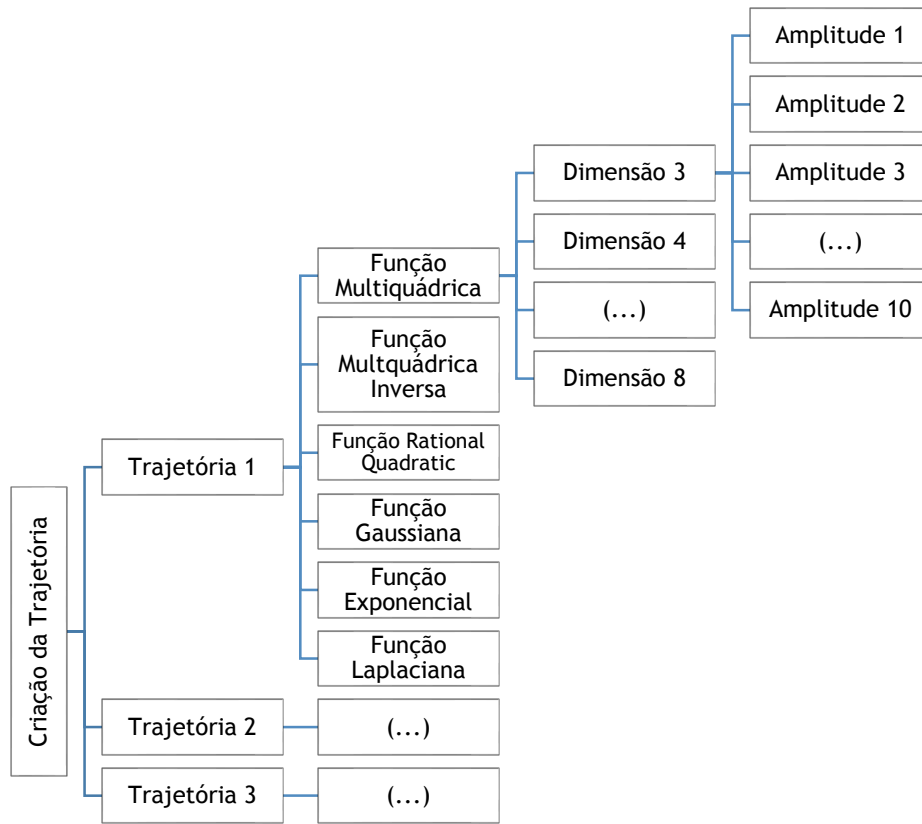


Figura 24 - Estrutura hierárquica dos processos efetuados pelo programa

3.3. Geração de Dados da Trajetória

A orientação da navegação geodética consiste no fornecimento de uma velocidade de referência V_{ref} , num rumo de referência ψ_{ref} e num ângulo de trajetória de referência γ_{ref} com a finalidade de gerar uma trajetória de voo que passe por uma sequência predefinida de $N + 1$ waypoints $(P_0, P_1, \dots, P_k, \dots, P_N)$. O modelo de navegação por waypoints 4D proposto nesta dissertação pode ser consultado em maior detalhe na referência [34].

Os parâmetros para a orientação numa navegação 4D são \dot{r} , $\dot{\lambda}$ e $\dot{\varphi}$, que podem ser obtidos através das seguintes equações de navegação geodética:

$$\begin{aligned} \dot{r} &= V \sin \gamma \\ \dot{\lambda} &= \frac{V \cos \gamma \sin \psi}{r \cos \varphi} \\ \dot{\varphi} &= \frac{V \cos \gamma \sin \psi}{r} \end{aligned} \quad (57)$$

Onde V é a velocidade de voo, γ é o ângulo de trajetória de voo, ψ é o ângulo de rumo, $r = R_E + h$ é a distância do centro da Terra ao CG da aeronave, R_E é o raio da Terra e h é a altitude de voo acima do nível do mar.

O objetivo da simulação passa por obter uma trajetória válida e constante, para uma determinada aeronave, e que passe pelos *waypoints* definidos inicialmente pelo utilizador. No final, a solução será apresentada para um total de N pontos, dependendo do valor do passo temporal da simulação. O método de geração desta trajetória será descrito nos próximos tópicos.

3.3.1. Implementação do Algoritmo de Simulação Numérica RK-Butcher

De forma a realizar a navegação, foi necessário implementar um algoritmo de criação da trajetória. Na implementação deste algoritmo, foi utilizado o Algoritmo de RK-Butcher (Runge-Kutta Butcher), para obter soluções das equações diferenciais resultados dos vetores de estado e de controlo da aeronave [35].

O modelo do sistema implementado é dado por:

$$\dot{x} = equadif(x_k, u_k) \quad (58)$$

Onde,

$$x \in \mathfrak{R}^n \text{ é o vetor de estado} \Rightarrow x = \begin{pmatrix} r \\ \lambda \\ \varphi \end{pmatrix}$$

$$u \in \mathfrak{R}^m \text{ é o vetor de controlo} \Rightarrow u = \begin{pmatrix} V_{ref} \\ \psi_{ref} \\ \gamma_{ref} \end{pmatrix}$$

e h (ou dt) representa o passo temporal da simulação, $x_k = x(t_k)$ e $u_k = u(t_k)$ onde $k = 0, 1, 2, \dots$ e $t_k = t_{k-1} + h$ no k^o passo.

Assumindo as condições iniciais x_0 como um dado presente, os estados do sistema acima descrito em $t_1, t_2, t_3, \dots, t_k$ são determinados por:

$$x_{k+1} = x_k + \frac{1}{90} (7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6) \quad (59)$$

Onde:

$$\begin{aligned} k_1 &= h \cdot f(x_k, u_k); \\ k_2 &= h \cdot f\left(x_k + \frac{k_1}{4}, u_k\right); \\ k_3 &= h \cdot f\left(x_k + \frac{k_1}{8} + \frac{k_2}{8}, u_k\right); \\ k_4 &= h \cdot f\left(x_k + \frac{k_2}{2} + k_3, u_k\right); \\ k_5 &= h \cdot f\left(x_k + \frac{3k_1}{16} + \frac{9k_4}{16}, u_k\right); \\ k_6 &= h \cdot f\left(x_k + \frac{3k_1}{7} + \frac{2k_2}{7} + \frac{12k_3}{7} - \frac{12k_4}{7} + \frac{8k_5}{7}, u_k\right); \end{aligned} \quad (60)$$

Cada waypoint P_k é descrito como um vetor com quatro dimensões, ou seja:

$$P_k = (\lambda_k \varphi_k h_k \tau_k)^T \quad (61)$$

em que λ_k representa a longitude do waypoint em radianos, φ_k a latitude, h_k a altitude (em relação ao nível do mar) em metros e τ_k o tempo previsto de chegada ao waypoint em segundos.

Por fim, o algoritmo criador da trajetória pode ser aplicado, seguindo os passos descritos de seguida:

1. Seja X a posição atual da aeronave na referência geodética. Primeiro, é necessário estabelecer um parâmetro $\sigma > 2b$, onde b representa a envergadura da asa da aeronave;
2. Calcular a distância $d_{\bar{P}_{k+1}}(X)$ da posição atual da aeronave $X = (\lambda, \varphi, h)$ até à posição do próximo waypoint de destino \bar{P}_k , na referência geodética;
3. Se $d_{\bar{P}_{k+1}}(X) \leq \sigma$, então define-se o tempo atual como o tempo de chegada ao waypoint P_k , e altera-se a navegação para P_{k+2} , definindo $k = k + 1$;
4. Calcular a velocidade de referência V_{ref} , rumo ψ_{ref} e ângulo de trajetória γ_{ref} da posição atual da aeronave para o waypoint de destino P_{k+1} ;
5. Enviar os dados da velocidade, rumo e ângulo de trajetória de referência para o algoritmo RK-Butcher mencionado anteriormente.

O cálculo da distância entre waypoints é definido nos parágrafos abaixo. Para o cálculo da velocidade, rumo e ângulo de trajetória de referência poderiam ter sido utilizado vários métodos distintos, no entanto, os cálculos utilizados na implementação do algoritmo desta dissertação podem ser consultados no anexo V.

3.3.2. Cálculo da Distância $d_{P_{k+1}}(X)$

Inicialmente, as coordenadas do waypoint P_{k+1} têm de ser transformadas da referência geodética, para a referência geocêntrica.

Foi proposto por H. Vermeile, em 2002 [36], um modelo transformação direta de coordenadas na referência geodética, para a referência geocêntrica. O modelo proposto pode ser resumido da seguinte forma:

$$\begin{aligned} X &= (h + n) \cos \varphi \cos \lambda \\ Y &= (h + n) \cos \varphi \sin \lambda \\ Z &= (h + n - e^2 n) \sin \varphi \end{aligned} \tag{62}$$

onde a representa o semieixo maior da Terra, e a excentricidade do elipsóide de referência, X, Y e Z as coordenadas na referência geocêntrica, λ, φ e h as coordenadas na referência geodética e o parâmetro n , que pode ser calculado através da fórmula:

$$n = \frac{a}{\sqrt{1 - e^2 \sin^2 \varphi}} \tag{63}$$

Para o cálculo do semieixo maior da Terra e da excentricidade do elipsoidal, recorre-se ao modelo WGS-84 (WGS: World Geodetic System) [37], que representa o modelo geográfico terrestre para a navegação aeronáutica. Apesar de o modelo esférico da Terra ser o mais utilizado, quando se trata de posicionamentos GPS ou de outros meios afastados da Terra, é prudente recorrer ao modelo elipsoidal.

Neste modelo, o raio equatorial, ou a tem o valor de $a = 6378.137 \text{ Km}$, ou semieixo maior da Terra. O raio polar, ou semieixo menor b tem o valor de $b = 6356.7523142 \text{ Km}$. Para se obter a excentricidade, é necessário calcular primeiro a taxa de achatamento f do elipsoide terrestre:

$$f = \frac{a - b}{a} = \frac{1}{298.257223563} \approx 0.0034 \quad (64)$$

Por fim, a excentricidade e é obtida do seguinte modo:

$$e = \sqrt{f(2 - f)} \approx 0.081819191 \quad (65)$$

Após a transformação em coordenadas geocêntricas, as coordenadas do waypoint passam a ser definidas por $P_{k+1}^c = (x_{k+1} \ y_{k+1} \ z_{k+1})^T$. A mesma transformação terá de ser aplicada as coordenadas da posição atual da aeronave X . Assim, a nomenclatura da posição da aeronave é dada por $X^c = (x \ y \ z)^T$.

A distância $d_{P_{k+1}}(X)$ é dada por:

$$d_{P_{k+1}}(X) = \sqrt{(x_{k+1} - x)^2 + (y_{k+1} - y)^2 + (z_{k+1} - z)^2} \quad (66)$$

3.3.3. Cálculo dos Tempos de Chegada aos Waypoints

Os tempos de chegada a cada *waypoint* foram obtidos através da seguinte fórmula:

$$t_{k+1} = t_k + \frac{D}{V_{med}} \times 1.2 \quad (67)$$

onde t_k representa o tempo de chegada ao *waypoint* anterior, D representa a distância entre o *waypoint* atual e o próximo, V_{med} representa um valor médio da velocidade da aeronave, que se encontra entre $V \in [V_{min}; V_{max}]$ e 1.2 um valor de margem de segurança.

As distâncias entre cada *waypoint* D , foram obtidas através do cálculo das distâncias loxodrómicas, explicadas nos pontos seguintes.

3.3.4. Loxodromia

Uma loxodromia é uma curva na superfície terrestre que forma o mesmo ângulo com qualquer meridiano, isto é, loxodromia representa um percurso com um rumo constante quando medido com o norte magnético.

Exemplo: os paralelos (e portanto o equador) são loxodromias (particulares) com um ângulo de corte dos meridianos igual a 90° .

Mesmo sabendo que para obter o percurso mais curto, é necessário correr ao cálculo da distância ortodrômica, a loxodromia representa o tipo de trajeto mais simples e normalmente aplica-se em mapas rodoviários ou marítimos e sistemas de navegação.

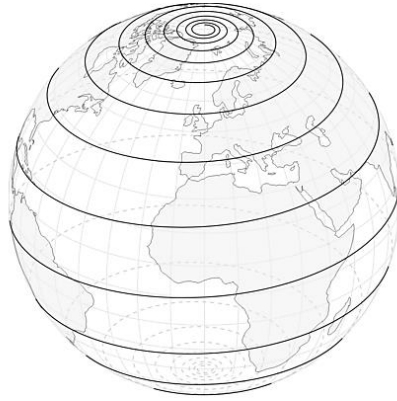


Figura 25 - Distâncias loxodrômicas (rhumb line), em direção ao Pólo Norte

3.3.5. Comprimento de um Arco de Loxodromia

Considerem-se dois pontos na superfície terrestre: A e B , em que a longitude e a latitude de A se representam como λ_A e φ_A respectivamente, e B com longitude λ_B e latitude φ_B . De seguida, é definido o ângulo loxodrômico V , segundo o qual a loxodromia corta os meridianos. Por fim, denota-se por $(AB)_{NM}^{lox}$ a distância loxodrômica, em milhas náuticas, entre A e B . Pode mostrar-se que:

$$(AB)_{NM}^{lox} = \frac{(\Delta\varphi_{AB})_{minutos}}{\cos V} \quad (68)$$

com $\Delta\varphi_{AB} = |\varphi_B - \varphi_A|$, e o ângulo V definido por:

$$V = \arctan_2((\Delta\lambda_{AB})_{minutos}, (\Delta\varphi_{AB}^*)_{minutos}) \quad (69)$$

com $\Delta\lambda_{AB} = |\lambda_B - \lambda_A|$ e $\Delta\varphi_{AB}^* = |\varphi_B^* - \varphi_A^*|$, onde

$$\varphi^* = \frac{10800}{\pi} \ln \left[\tan \left(\frac{\varphi}{2} + 45^\circ \right) \right] \text{ (em minutos angulares)} \quad (70)$$

onde a latitude φ está em graus [38].

Qualquer bússola permite seguir uma distância loxodrômica, uma vez que para seguir uma loxodromia entre dois pontos, basta ir de um para outro mantendo um rumo constante, igual ao ângulo loxodrômico entre estes mesmos (quando o vento for desprezível).

Uma navegação loxodrômica de um ponto A para um ponto B consiste primeiro em escolher uma série de waypoints entre A e B (de modo a ter waypoints consecutivos, relativamente próximos um do outro), e a seguir navegar de um waypoint para o outro (a partir de A) segundo uma rota loxodrômica, até chegar a B .

O sistema de coordenadas ECEF (*Earth-Centered, Earth-Fixed*) foi utilizado durante toda a navegação. Neste sistema, a posição da aeronave é representada pelas coordenadas X , Y e Z , como pode ser observado na Figura 26. O ponto $(0,0,0)$ representa o centro de massa da Terra, daí o seu nome de *Earth-Centered* (centrado na Terra). Os seus eixos são alinhados com o polo de referência internacional (IRP - *International Reference Pole*) e pelo meridiano de referência internacional (IRM - *International Reference Meridian*) que se encontram fixos em relação à superfície terrestre, daí o nome *Earth-Fixed* (fixado na Terra) [39].

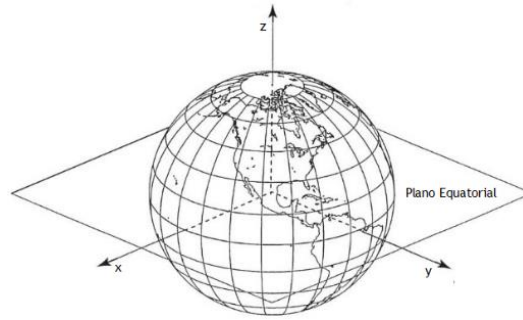


Figura 26 - Sistema de coordenadas ECEF (Earth-Centered, Earth-Fixed)

A trajetória gerada pela aeronave entre os dois aeroportos, de acordo com o sistema de coordenadas ECEF, para um determinado intervalo de tempo foi definida da seguinte forma:

Tabela 3 - Coordenadas ECEF da aeronave ao longo da trajetória

Posição da Aeronave (Coordenadas Geocêntricas)			
Coordenada X [m]	Coordenada Y [m]	Coordenada Z [m]	Tempo [s]
x_1	y_1	z_1	t_1
x_2	y_2	z_2	t_2
x_3	y_3	z_3	t_3
...
x_N	y_N	z_N	t_N

As listagens dos *waypoints* utilizados podem ser observadas nas tabelas seguintes. Com o objetivo de aproximar a simulação a uma situação real, as trajetórias escolhidas representam rotas de aeronaves comerciais reais entre aeroportos internacionais.

Para a primeira trajetória, a navegação escolhida ocorre entre os aeroportos de:

- Aeroporto de Partida: *London Heathrow Airport*;
- Aeroporto de Chegada: *Paris Charles de Gaulle Airport*.

As coordenadas dos aeroportos, bem como a lista dos *waypoints* utilizados entre estes, podem ser consultadas na Tabela 4.

Tabela 4 - Coordenadas dos Aeroportos e Waypoints com Tempo de Chegada da Trajetória 1

	Longitude [rad]	Latitude [rad]	Altitude [m]	Tempo [s]
A: Heathrow Airport (Londres)	-0,008052755	0,898451866	25	0
B: Waypoint 1	0,008215168	0,88878822	12527	715,4104504
C: Waypoint 2	0,017690657	0,88421258	12527	7445,688815
D: Waypoint 3	0,027965168	0,876243133	12527	15261,88598
E: Waypoint 4	0,03459548	0,869651364	12527	15661,65053
F: Waypoint 5	0,040237015	0,862009827	12527	16093,1082
G: Charles de Gaulle (Paris)	0,044467111	0,855381018	119	16458,91326

Mais uma vez, os tempos de chegada a cada *waypoint* foram calculados através da Equação 64 descrita no Capítulo 3.3.3.

Os parâmetros da aeronave utilizada na criação desta trajetória podem ser consultados na tabela seguinte:

Tabela 5 - Parâmetros operacionais da aeronave 1

Dados da Aeronave:

Aeronave 1	Velocidade mínima [m/s]:	64
	Velocidade máxima [m/s]:	253,61
	Ângulo de trajetória mínimo [rad]:	-0,489
	Ângulo de trajetória máximo [rad]:	0,611
	Taxa de ângulo de traj. mín [rad/s]:	-1
	Taxa de ângulo de traj. máx [rad/s]:	1
	Taxa de rumo mínima [rad/s]:	-1
	Taxa de rumo máxima [rad/s]:	1

O nome verdadeiro do fabricante e do modelo da aeronave não podem ser revelados neste trabalho, devido à burocracia exigida por parte dos fabricantes para que os nomes corretos pudessem ser utilizados.

A trajetória obtida, em coordenadas geocêntricas, pode então ser observada na Figura 27.

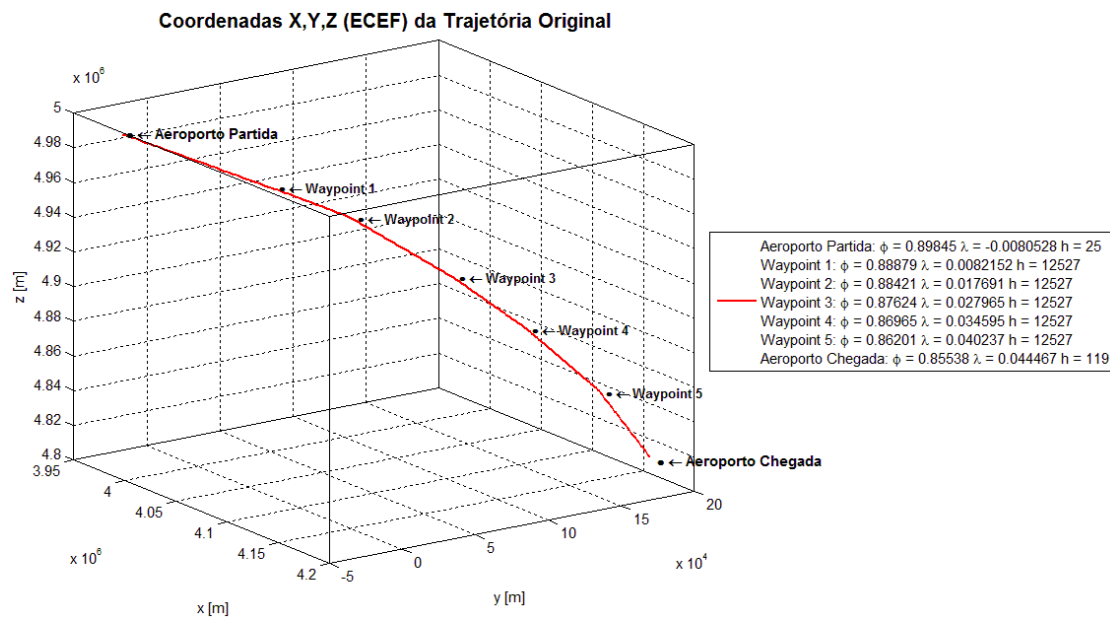


Figura 27 - Trajetória 1 (Coordenadas Geocêntricas) com Listagem de Waypoints

Para a criação da Trajetória 1, foi utilizado um passo temporal da simulação de $dt = 20s$. Este valor foi encontrado tendo em conta um compromisso entre o máximo número de pontos possível, e o tempo total de execução do programa. Para uma observação um pouco mais intuitiva, a Figura 28 mostra a Trajetória 1 na superfície terrestre.

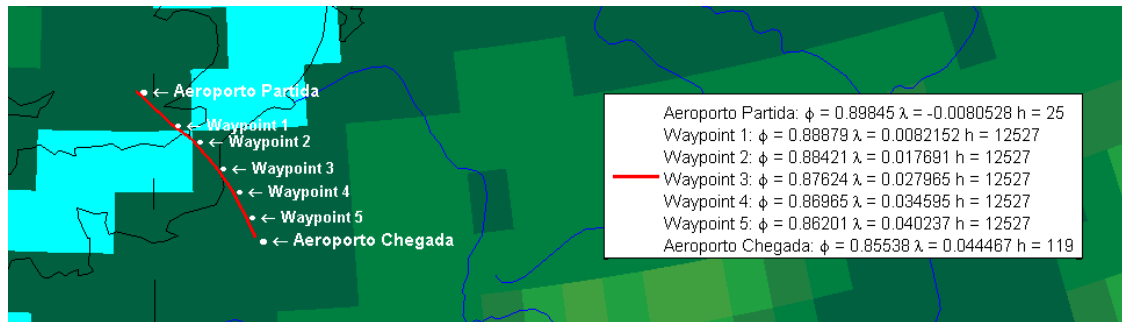


Figura 28 - Trajetória 1 representada na superfície terrestre

Como é possível observar, a Trajetória 1 representa um percurso relativamente curto, entre dois aeroportos bastante próximos. Isto deve-se ao fato de que quanto maior for a distância entre o aeroporto de partida, e o aeroporto de chegada, maior e mais complexa será a trajetória criada, que irá aumentar o tempo de execução do algoritmo de forma exponencial. Assim, a Trajetória 1 foi utilizada como modelo base para comparação entre todas as funções RBF explícitas no Capítulo 2.2.1., para que depois seja feita uma análise em detalhe dos erros obtidos para cada uma delas, permitindo assim a aplicação a trajetórias mais complexas utilizando apenas um modelo RBF.

Foram então criadas mais duas trajetórias para serem aplicadas ao algoritmo LS-SVM, após o apuramento das funções e parâmetros ideais obtidos da experimentação com a Trajetória 1. A segunda trajetória teve como origem e destino os seguintes aeroportos:

- Aeroporto de Partida: *Incheon International Airport (Seul)*;
- Aeroporto de Chegada: *Paris Charles de Gaulle Airport*.

As coordenadas de ambos os aeroportos, bem como a listagem dos waypoints utilizados entre eles pode ser consultada na Tabela 6:

Tabela 6 - Coordenadas dos Aeroportos e Waypoints com Tempo de Chegada da Trajetória 2

	Longitude [rad]	Latitude [rad]	Altitude [m]	Tempo [s]
A: Incheon Airport (Seul)	2,194582237	0,652836916	5113	0
B: Waypoint 1	2,147552595	0,662047019	9167	1953,244937
C: Waypoint 2	2,080572095	0,668845076	9174	4659,154002
D: Waypoint 3	2,034010201	0,719766802	9174	7828,834047
E: Waypoint 4	1,930960726	0,804713722	9174	13582,75326
F: Waypoint 5	1,717644839	0,87830553	9784	21730,18056
G: Waypoint 6	1,418977116	0,999171326	9921	32602,50014
H: Waypoint 7	1,108490024	1,066394428	10356	41390,53895
I: Waypoint 8	0,811743908	1,076115912	10363	48647,35684
J: Waypoint 9	0,59836519	1,047827615	10965	54133,07782
K: Waypoint 10	0,296746115	0,983999179	10973	62847,35005
L: Waypoint 11	0,065743062	0,863840241	5029	72199,8191
M: Charles de Gaulle (Paris)	0,054426347	0,855878049	1593	72752,93279

Após a aplicação do método de navegação por waypoints 4D apresentado no Capítulo 3.3., a trajetória obtida foi a seguinte:

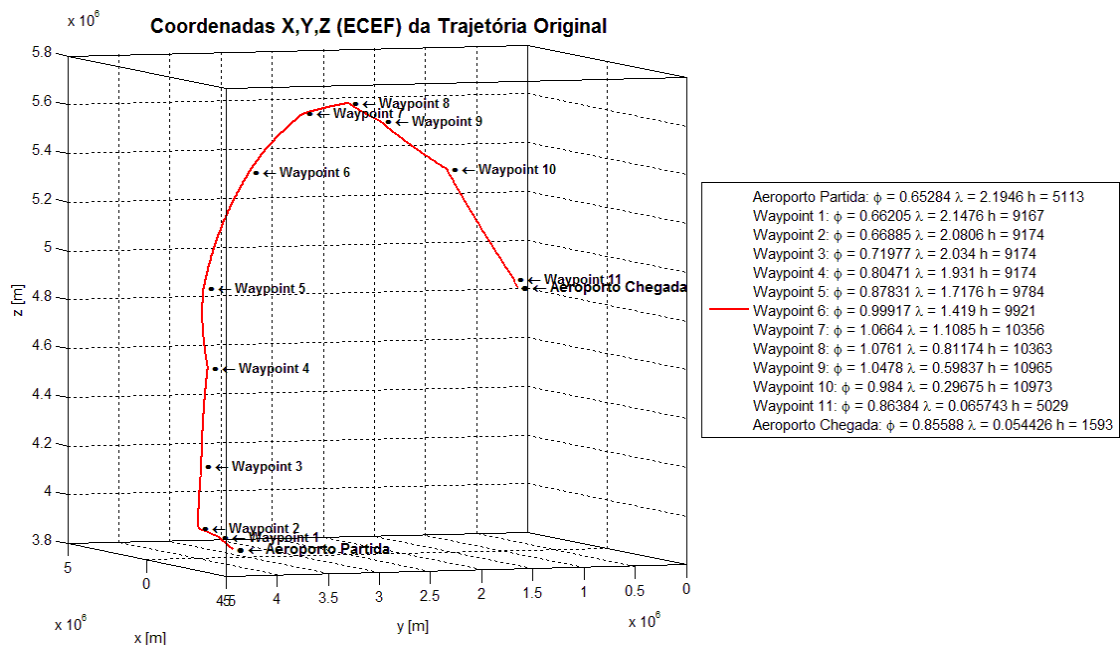


Figura 29 - Trajetória 2 (Coordenadas Geocêntricas) com Listagem de Waypoints

Mais uma vez, para uma melhor compreensão da trajetória a nível visual, a Figura 30 demonstra a rota percorrida pela aeronave entre os aeroportos mencionados na Tabela 6.

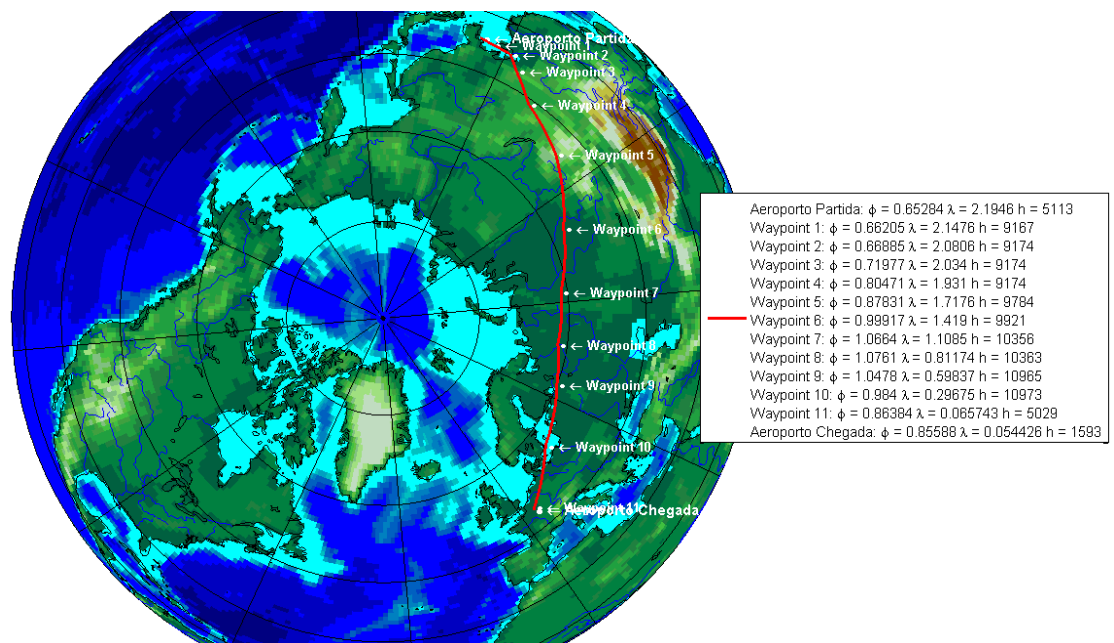


Figura 30 - Trajetória 2 representada na superfície terrestre

Por fim, a última trajetória a ser criada (Trajetória 3), demonstra uma navegação entre os seguintes aeroportos internacionais:

- Aeroporto de Partida: *Vancouver International Airport*;
- Aeroporto de Chegada: *London Heathrow Airport*.

As coordenadas dos pontos de partida e chegada e dos waypoints utilizados, podem ser consultados na Tabela 7.

Tabela 7 - Coordenadas dos Aeroportos e Waypoints com Tempo de Chegada da Trajetória 3

	Longitude [rad]	Latitude [rad]	Altitude [m]	Tempo [s]
A: Vancouver Airport	-2,150041435	0,858660104	0	0
B: Waypoint 1	-2,14320498	0,861243191	3124	262,5398409
C: Waypoint 2	-2,081348766	0,907040631	10058	3332,986413
D: Waypoint 3	-2,038843018	0,93438645	10058	5245,186108
E: Waypoint 4	-1,845106235	1,033997626	10058	12690,9594
F: Waypoint 5	-1,620574862	1,105440934	10058	19272,95646
F: Waypoint 6	-1,314880444	1,147777385	10058	26298,0346
F: Waypoint 7	-0,9459702	1,143368684	10668	34052,52707
F: Waypoint 8	-0,503588576	1,078953817	10668	44557,8548
F: Waypoint 9	-0,324770867	1,041216308	10668	49405,17498
F: Waypoint 10	-0,182039586	0,997577841	10668	53813,18455
F: Waypoint 11	-0,051851987	0,929925388	10668	58927,20015
F: Waypoint 12	-0,039407789	0,921275536	9114	59509,76034
G: Heathrow Airport (Londres)	-0,007433357	0,8982337	0	62504,5111

De modo idêntico às trajetórias anteriores, a Trajetória 3 criada pelo algoritmo tem o seguinte formato:

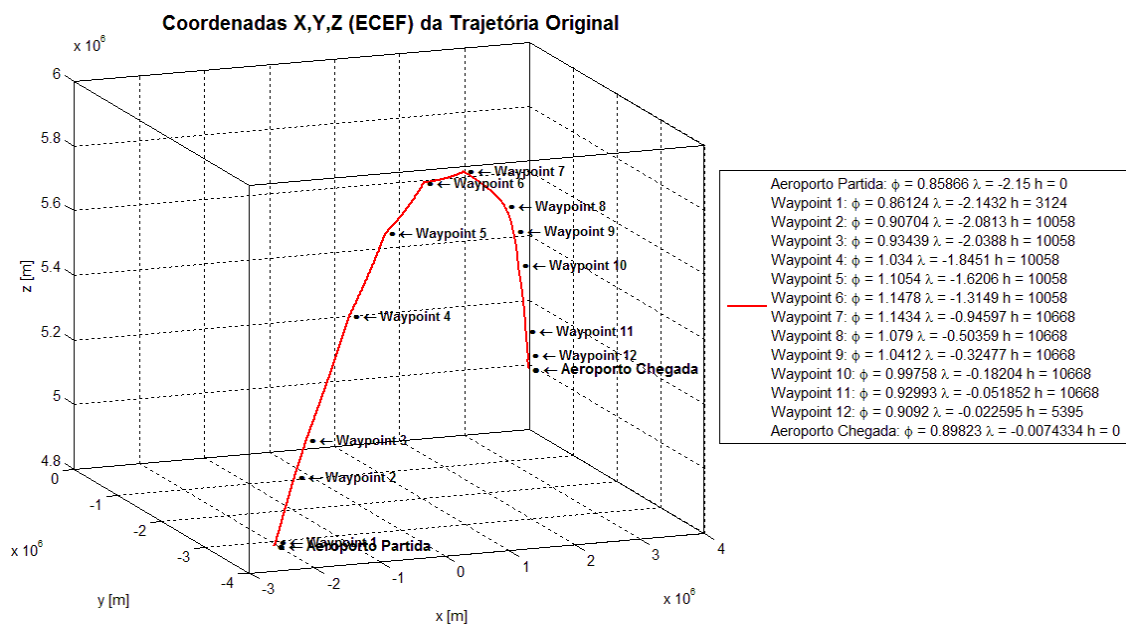


Figura 31 - Trajetória 3 (Coordenadas Geocêntricas) com Listagem de Waypoints

Representando a Trajetória 3 na superfície terrestre, é obtida a seguinte visualização:

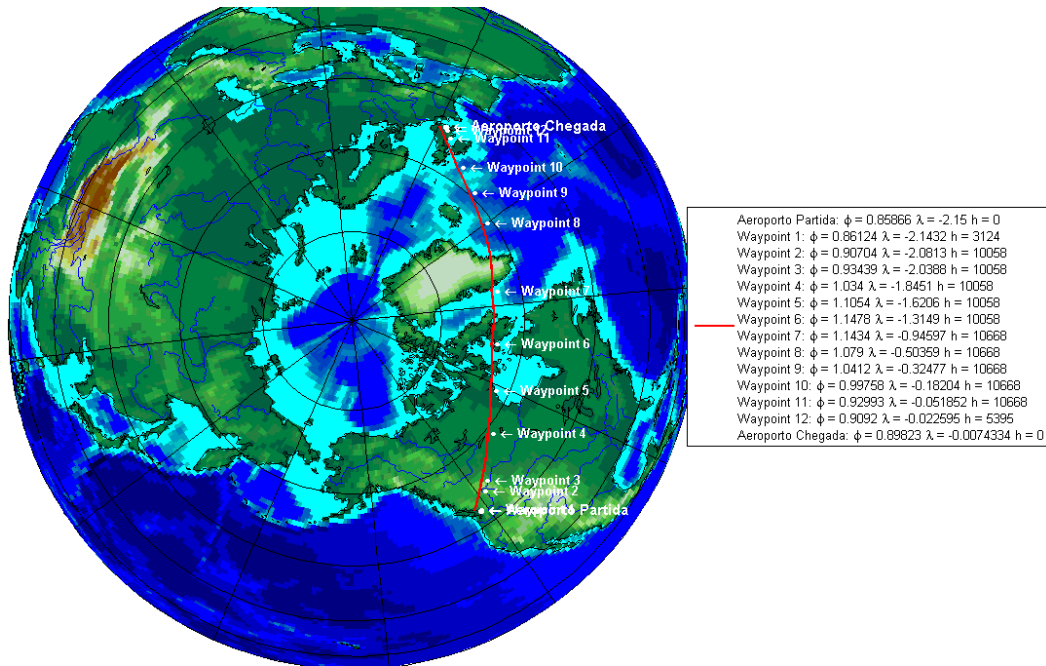


Figura 32 - Trajetória 3 representada na superfície terrestre

No próximo tópico, é feita uma descrição detalhada sobre como as trajetórias criadas foram analisadas pelo algoritmo LS-SVM. Mais uma vez, de realçar que apenas a Trajetória 1 foi utilizada para fazer um estudo comparativo entre todos os modelos RBF, devido à sua “leveza computacional”. As outras duas trajetórias foram aplicadas posteriormente, após se terem obtido os parâmetros ideais para a sua regressão.

3.4. Injeção de Erro Gaussiano

Tal como descrito no Capítulo 2.2.2., e implementado anteriormente no exemplo da função seno no Capítulo 3.1., foi também criado um erro de tipo gaussiano (com uma distribuição normal), com o intuito de aproximar os dados da trajetória gerados no capítulo anterior de uma situação real, como por exemplo, da leitura de dados GPS. Como explicado pelo teorema de Box-Muller, quaisquer números aleatórios com uma distribuição uniforme podem ser transformados em variáveis aleatórias com uma distribuição normal.

Para as coordenadas X e Y foi utilizado o mesmo desvio padrão com $\sigma_x = \sigma_y = 16 m$. Para a coordenada Z o desvio padrão tomou o valor de $\sigma_z = 30 m$. Para além disso, foi feito um estudo para diferentes ruídos, resultantes da aplicação de uma amplitude variável para cada erro diferente. Por outras palavras, o algoritmo LS-SVM foi implementado para as três trajetórias criadas anteriormente, às quais lhes foi adicionado um erro gaussiano com várias amplitudes diferentes. Na simulação computacional foram utilizados dez valores de amplitudes diferentes, e são eles:

$$Amp_{erro} = [5, 10, 20, 40, 80, 160, 320, 640, 1280, 2560] \quad (71)$$

Graficamente, a evolução das amplitudes do erro gaussiano pode ser traçada da seguinte forma:

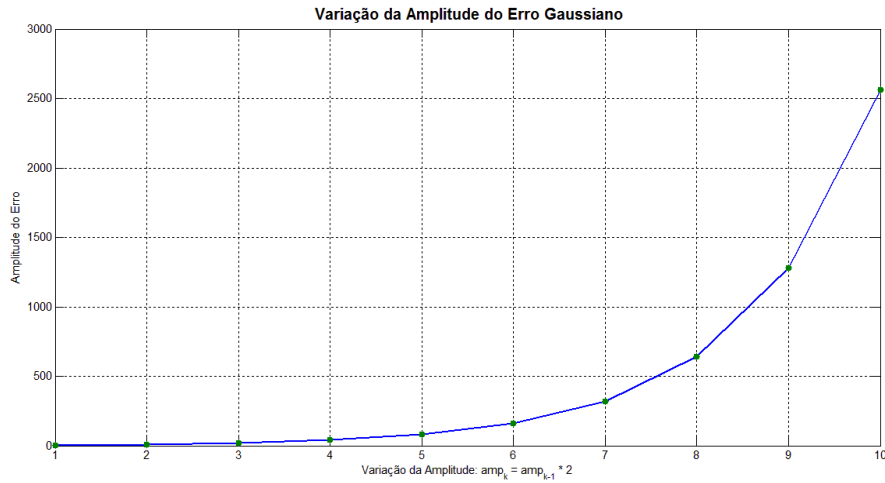


Figura 33 - Diferentes amplitudes do Erro Gaussiano

Seguidamente, são apresentados alguns exemplos de erros criados e aplicados às diferentes coordenadas X, Y e Z:

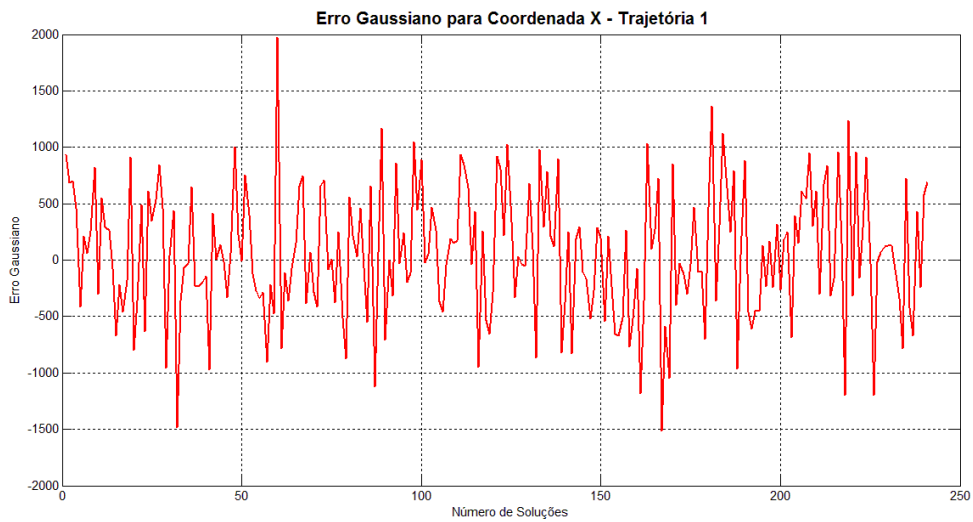


Figura 34 - Erro Gaussiano aplicado à Coordenada X para a Trajetória 1

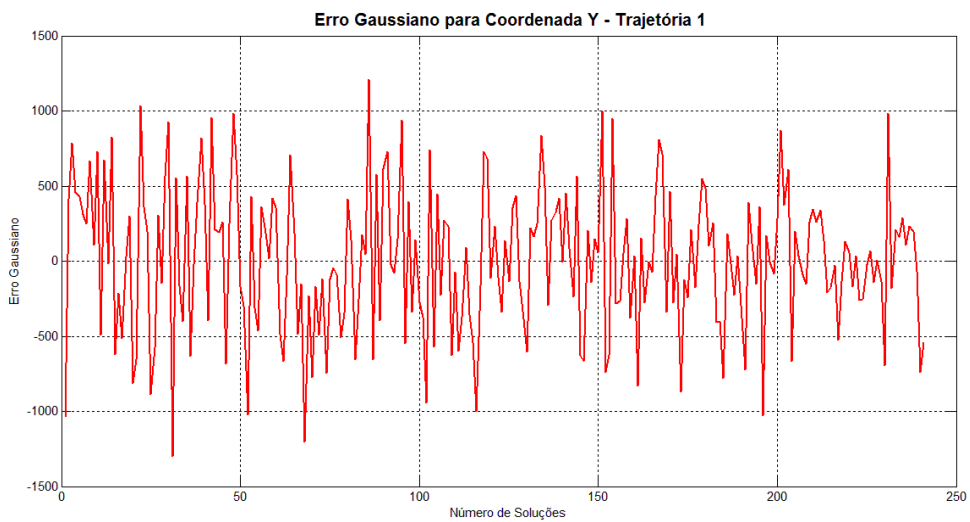


Figura 35 - Erro Gaussiano aplicado à Coordenada Y para a Trajetória 1

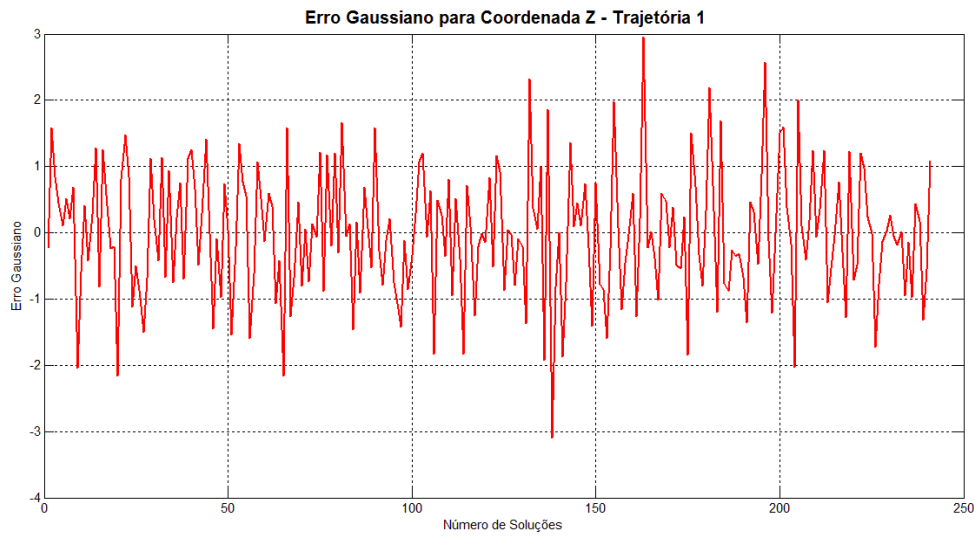


Figura 36 - Erro Gaussiano aplicado à Coordenada Z para a Trajetória 1

Assim, obtém-se um erro gaussiano com um número de variáveis igual ao número de pontos para cada coordenada das trajetórias geradas anteriormente. Por outras palavras, é criado um vetor $Erro_x = [Erro_{x_1}, Erro_{x_2}, \dots, Erro_{x_N}]$ para a coordenada X, $Erro_y = [Erro_{y_1}, Erro_{y_2}, \dots, Erro_{y_N}]$ para a coordenada Y e $Erro_z = [Erro_{z_1}, Erro_{z_2}, \dots, Erro_{z_N}]$ para a coordenada Z.

Estes vetores de erro gaussiano são depois adicionados aos respetivos dados de entrada, como pode ser observado na tabela seguinte:

Tabela 8 - Coordenadas da aeronave com erro gaussiano

Posição da Aeronave com Erro Gaussiano			
Coordenada X [m]	Coordenada Y [m]	Coordenada Z [m]	Tempo [s]
$x_1 + Erro_{x_1}$	$y_1 + Erro_{y_1}$	$z_1 + Erro_{z_1}$	t_1
$x_2 + Erro_{x_2}$	$y_2 + Erro_{y_2}$	$z_2 + Erro_{z_2}$	t_2
$x_3 + Erro_{x_3}$	$y_3 + Erro_{y_3}$	$z_3 + Erro_{z_3}$	t_3
...
$x_N + Erro_{x_N}$	$y_N + Erro_{y_N}$	$z_N + Erro_{z_N}$	t_N

Onde N representa o número de pontos de cada coordenada das trajetórias criadas anteriormente.

Por fim, é possível comparar agora as trajetórias geradas inicialmente, e as trajetórias depois da implementação do erro gaussiano:

Coordenadas X,Y,Z (ECEF) da Trajetória Original e da Trajetória com Erro Gaussiano

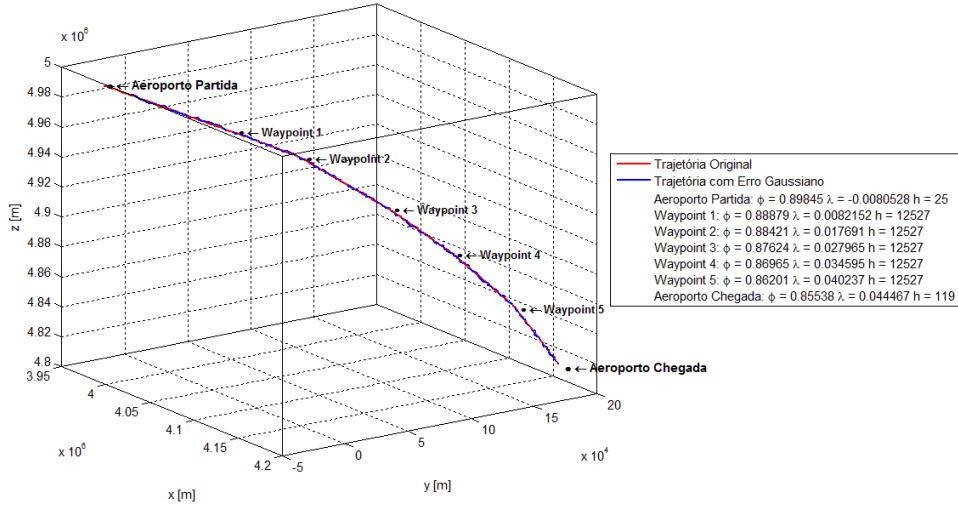


Figura 37 - Trajetória 1: Original vs. com Erro Gaussiano

Coordenadas X,Y,Z (ECEF) da Trajetória Original e da Trajetória com Erro Gaussiano

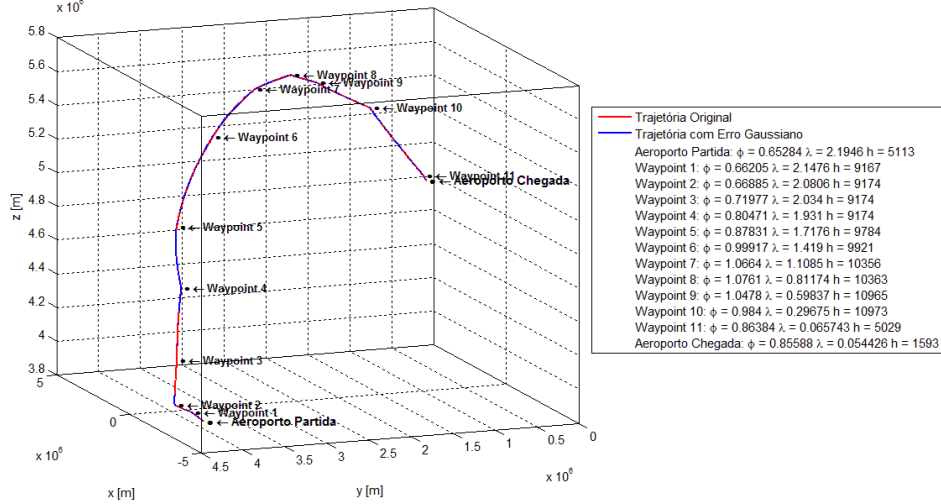


Figura 38 - Trajetória 2: Original vs. com Erro Gaussiano

Coordenadas X,Y,Z (ECEF) da Trajetória Original e da Trajetória com Erro Gaussiano

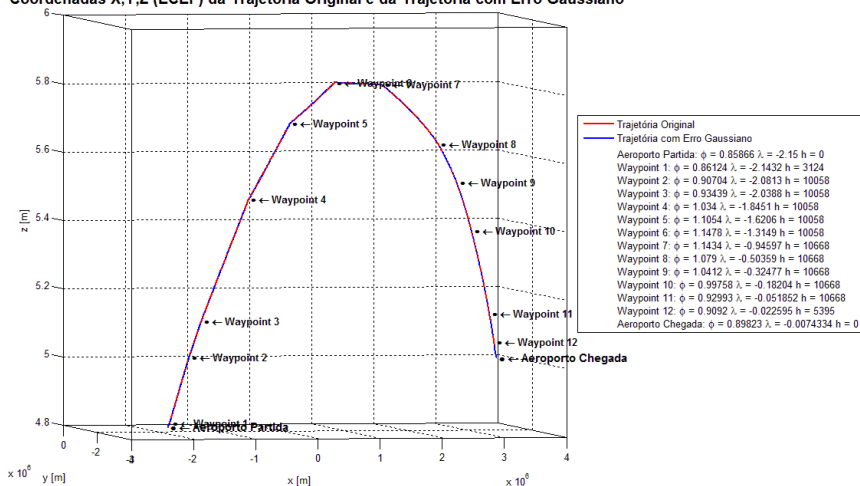


Figura 39 - Trajetória 3: Original vs. com Erro Gaussiano

Nas Figuras 37, 38 e 39 é possível fazer uma comparação entre as trajetórias originais criadas pelo algoritmo de navegação 4D (a vermelho), e as trajetórias após implementação do erro gaussiano (a azul). De notar que estas figuras representam apenas a implementação de um dos erros gaussianos possíveis. Para cada uma das diferentes dimensões, são aplicadas dez amplitudes de erro diferentes, como observado no esquema da Figura 33.

3.5. Reestruturação dos Dados

Após a criação da trajetória e da injeção de erro gaussiano nas coordenadas da mesma, o resultado obtido é igual ao que se encontra na Tabela 8. No entanto, para que o algoritmo LS-SVM possa ser executado, é necessário criar um conjunto de soluções que servirão para treinar o programa em questão. Esse processo toma o nome de reestruturação dos dados de entrada, e necessita de uma dimensão a definir pelo utilizador, tal como apresentado na Figura 24.

Na Figura 24 é possível observar um total de seis dimensões diferentes que foram aplicadas no algoritmo (dimensão 3, 4, 5, 6, 7 e 8). O valor da dimensão define o comprimento da janela de dados que, por sua vez, criará um conjunto de soluções a partir dos dados de entrada. Por exemplo, para qualquer uma das trajetórias, que contém coordenadas X, Y e Z, se for atribuída uma dimensão de $dim = 3$, a reestruturação dos dados apresenta a seguinte forma:

Tabela 9 - Reestruturação das Coordenadas X, Y e Z para uma dimensão = 3

Tempo [s]	Coordenada X [m]				y_x	Coordenada Y [m]				y_y	Coordenada Z [m]				y_z
t_1	x_1	x_2	x_3	x_4	x_4	y_1	y_2	y_3	y_4	y_4	z_1	z_2	z_3	z_4	z_4
t_2	x_2	x_3	x_4	x_5	x_5	y_2	y_3	y_4	y_5	y_5	z_2	z_3	z_4	z_5	z_5
t_3	x_3	x_4	x_5	x_6	x_6	y_3	y_4	y_5	y_6	y_6	z_3	z_4	z_5	z_6	z_6
...
t_N	x_{N-3}	x_{N-2}	x_{N-1}	x_N	x_N	y_{N-3}	y_{N-2}	y_{N-1}	y_N	y_N	z_{N-3}	z_{N-2}	z_{N-1}	z_N	z_N

Da tabela anterior, é possível concluir que os vetores das soluções são dados por y_x, y_y, y_z , e que estes são compostos por valores das coordenadas respetivas, que tinham sido determinados na criação da trajetória. Analogamente, a reestruturação dos dados de entrada para as dimensões 4, 5, 6, 7 e 8 pode ser observada nas tabelas seguintes:

Tabela 10 - Reestruturação das Coordenadas X, Y e Z para uma dimensão = 4

t	Coordenada X [m]					y_x	Coordenada Y [m]					y_y	Coordenada Z [m]					y_z
t_1	x_1	x_2	x_3	x_4	x_5	x_5	y_1	y_2	y_3	y_4	y_5	y_5	z_1	z_2	z_3	z_4	z_5	z_5
t_2	x_2	x_3	x_4	x_5	x_6	x_6	y_2	y_3	y_4	y_5	y_6	y_6	z_2	z_3	z_4	z_5	z_6	z_6
t_3	x_3	x_4	x_5	x_6	x_7	x_7	y_3	y_4	y_5	y_6	y_7	y_7	z_3	z_4	z_5	z_6	z_7	z_7
...
t_N	x_{N-4}	x_{N-3}	x_{N-2}	x_{N-1}	x_N	x_N	y_{N-4}	y_{N-3}	y_{N-2}	y_{N-1}	y_N	y_N	z_{N-4}	z_{N-3}	z_{N-2}	z_{N-1}	z_N	z_N

Tabela 11 - Reestruturação das Coordenadas X, Y e Z para uma dimensão = 5

t	Coordenada X [m]						y_x	Coordenada Y [m]						y_y	Coordenada Z [m]						y_z
t_1	x_1	x_2	x_3	x_4	x_5	x_6	x_6	y_1	y_2	y_3	y_4	y_5	y_6	y_6	z_1	z_2	z_3	z_4	z_5	z_6	z_6
t_2	x_2	x_3	x_4	x_5	x_6	x_7	x_7	y_2	y_3	y_4	y_5	y_6	y_7	y_7	z_2	z_3	z_4	z_5	z_6	z_7	z_7
t_3	x_3	x_4	x_5	x_6	x_7	x_8	x_8	y_3	y_4	y_5	y_6	y_7	y_8	y_8	z_3	z_4	z_5	z_6	z_7	z_8	z_8
...
t_N	x_{N-5}	x_{N-4}	x_{N-3}	x_{N-2}	x_{N-1}	x_N	x_N	y_{N-5}	y_{N-4}	y_{N-3}	y_{N-2}	y_{N-1}	y_N	y_N	z_{N-5}	z_{N-4}	z_{N-3}	z_{N-2}	z_{N-1}	z_N	z_N

Tabela 12 - Reestruturação das Coordenadas X, Y e Z para uma dimensão = 6

Coordenada X [m]							y_x	Coordenada Y [m]							y_y	Coordenada Z [m]						y_z	
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8
x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	z_2	z_3	z_4	z_5	z_6	z_7	z_8	z_9
x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	z_3	z_4	z_5	z_6	z_7	z_8	z_9	z_{10}
...
x_{N-6}	x_{N-5}	x_{N-4}	x_{N-3}	x_{N-2}	x_{N-1}	x_N		y_{N-6}	y_{N-5}	y_{N-4}	y_{N-3}	y_{N-2}	y_{N-1}	y_N		z_{N-6}	z_{N-5}	z_{N-4}	z_{N-3}	z_{N-2}	z_{N-1}	z_N	

Na última tabela, a coluna relativa aos dados do tempo foi retirada, apenas por uma questão de ocupação de espaço.

Tabela 13 - Reestruturação das Coordenadas X, Y e Z para uma dimensão = 7

t	Coordenada X [m]								y_x	Coordenada Y [m]							y_y	
t_1	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9
t_2	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}
t_3	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}
...
t_N	x_{N-7}	x_{N-6}	x_{N-5}	x_{N-4}	x_{N-3}	x_{N-2}	x_{N-1}	x_N		y_{N-7}	y_{N-6}	y_{N-5}	y_{N-4}	y_{N-3}	y_{N-2}	y_{N-1}	y_N	

t	Coordenada Z [m]							y_z
t_1	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8
t_2	z_2	z_3	z_4	z_5	z_6	z_7	z_8	z_9
t_3	z_3	z_4	z_5	z_6	z_7	z_8	z_9	z_{10}
...
t_N	z_{N-7}	z_{N-6}	z_{N-5}	z_{N-4}	z_{N-3}	z_{N-2}	z_{N-1}	z_N

Tabela 14 - Reestruturação das Coordenadas X, Y e Z para uma dimensão = 8

t	Coordenada X [m]									y_x	Coordenada Y [m]								y_y		
t_1	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	
t_2	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	
t_3	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	
...
t_N	x_{N-8}	x_{N-7}	x_{N-6}	x_{N-5}	x_{N-4}	x_{N-3}	x_{N-2}	x_{N-1}	x_N		y_{N-8}	y_{N-7}	y_{N-6}	y_{N-5}	y_{N-4}	y_{N-3}	y_{N-2}	y_{N-1}	y_N		

t	Coordenada Z [m]									y_z
t_1	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8	z_9	z_{10}
t_2	z_2	z_3	z_4	z_5	z_6	z_7	z_8	z_9	z_{10}	z_{11}
t_3	z_3	z_4	z_5	z_6	z_7	z_8	z_9	z_{10}	z_{11}	z_{12}
...
t_N	z_{N-8}	z_{N-7}	z_{N-6}	z_{N-5}	z_{N-4}	z_{N-3}	z_{N-2}	z_{N-1}	z_N	

De outro modo, os dados de entrada correspondentes às coordenadas X, Y e Z passam a ser apresentados sob forma vetorial, onde cada vetor é composto por valores escalares de cada coordenada da trajetória. A título de exemplo, para uma reestruturação de dimensão 6, os vetores das coordenadas X, Y e Z apresentam a seguinte forma:

$$\begin{aligned}
 X_1 &= (x_1, x_2, x_3, x_4, x_5, x_6) & Y_1 &= (y_1, y_2, y_3, y_4, y_5, y_6) & Z_1 &= (z_1, z_2, z_3, z_4, z_5, z_6) \\
 X_2 &= (x_2, x_3, x_4, x_5, x_6, x_7) & Y_2 &= (y_2, y_3, y_4, y_5, y_6, y_7) & Z_2 &= (z_2, z_3, z_4, z_5, z_6, z_7) \\
 &\dots & &\dots & &\dots \\
 X_{N-6} &= (x_{N-6}, x_{N-5}, x_{N-4}, x_{N-3}, x_{N-2}, x_{N-1}) & Y_{N-6} &= (y_{N-6}, y_{N-5}, y_{N-4}, y_{N-3}, y_{N-2}, y_{N-1}) & Z_{N-6} &= (z_{N-6}, z_{N-5}, z_{N-4}, z_{N-3}, z_{N-2}, z_{N-1})
 \end{aligned} \tag{72}$$

De seguida, tal como observado nas Tabelas 9, 10, 11, 12, 13 e 14, o vector y presente no sistema linear representado pela Equação 38, toma a forma:

Para a Coordenada X:

$$\begin{aligned}
y_x &= (x_4, x_5, \dots, x_N)^T - \text{para uma reestruturação de dimensão 3;} \\
y_x &= (x_5, x_6, \dots, x_N)^T - \text{para uma reestruturação de dimensão 4;} \\
y_x &= (x_6, x_7, \dots, x_N)^T - \text{para uma reestruturação de dimensão 5;} \\
y_x &= (x_7, x_8, \dots, x_N)^T - \text{para uma reestruturação de dimensão 6;} \\
y_x &= (x_8, x_9, \dots, x_N)^T - \text{para uma reestruturação de dimensão 7;} \\
y_x &= (x_9, x_{10}, \dots, x_N)^T - \text{para uma reestruturação de dimensão 8;}
\end{aligned} \tag{73}$$

Para a Coordenada Y:

$$\begin{aligned}
y_y &= (y_4, y_5, \dots, y_N)^T - \text{para uma reestruturação de dimensão 3;} \\
y_y &= (y_5, y_6, \dots, y_N)^T - \text{para uma reestruturação de dimensão 4;} \\
y_y &= (y_6, y_7, \dots, y_N)^T - \text{para uma reestruturação de dimensão 5;} \\
y_y &= (y_7, y_8, \dots, y_N)^T - \text{para uma reestruturação de dimensão 6;} \\
y_y &= (y_8, y_9, \dots, y_N)^T - \text{para uma reestruturação de dimensão 7;} \\
y_y &= (y_9, y_{10}, \dots, y_N)^T - \text{para uma reestruturação de dimensão 8;}
\end{aligned} \tag{74}$$

Para a Coordenada Z:

$$\begin{aligned}
y_z &= (z_4, z_5, \dots, z_N)^T - \text{para uma reestruturação de dimensão 3;} \\
y_z &= (z_5, z_6, \dots, z_N)^T - \text{para uma reestruturação de dimensão 4;} \\
y_z &= (z_6, z_7, \dots, z_N)^T - \text{para uma reestruturação de dimensão 5;} \\
y_z &= (z_7, z_8, \dots, z_N)^T - \text{para uma reestruturação de dimensão 6;} \\
y_z &= (z_8, z_9, \dots, z_N)^T - \text{para uma reestruturação de dimensão 7;} \\
y_z &= (z_9, z_{10}, \dots, z_N)^T - \text{para uma reestruturação de dimensão 8;}
\end{aligned} \tag{75}$$

Ainda na resolução da Equação 38, assume-se que: o vetor $e = (1, 1, \dots, 1)^T$ apresenta um número de colunas igual ao número de soluções N , a função kernel: $K_{i,j} = K(X_i, X_j) = \varphi(X_i)^T \varphi(X_j)$, $K(Y_i, Y_j) = \varphi(Y_i)^T \varphi(Y_j)$ e $K(Z_i, Z_j) = \varphi(Z_i)^T \varphi(Z_j)$, onde os dados de entrada estão representados pelas expressões na Equação 67, o parâmetro real e positivo γ encontra-se num intervalo $\gamma = [10, 100]$, I representa uma matriz identidade com dimensões iguais às da matriz $K_{i,j}$ e $\begin{bmatrix} b \\ \alpha \end{bmatrix}$ representa a matriz das soluções do sistema linear.

Assim, a metodologia descrita no parágrafo anterior, pode ser descrita pelo seguinte sistema:

Para uma reestruturação de dimensão 3:

$$X_k = \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \end{pmatrix} \rightarrow \boxed{f} \rightarrow f_x \text{ e } Y_k = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \end{pmatrix} \rightarrow \boxed{f} \rightarrow f_y \text{ e } Z_k = \begin{pmatrix} z^{(1)} \\ z^{(2)} \\ z^{(3)} \end{pmatrix} \rightarrow \boxed{f} \rightarrow f_z \tag{76}$$

Para uma reestruturação de dimensão 4:

$$X_k = \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \\ x^{(4)} \end{pmatrix} \rightarrow f \rightarrow f_x \text{ e } Y_k = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ y^{(4)} \end{pmatrix} \rightarrow f \rightarrow f_y \text{ e } Z_k = \begin{pmatrix} z^{(1)} \\ z^{(2)} \\ z^{(3)} \\ z^{(4)} \end{pmatrix} \rightarrow f \rightarrow f_z \quad (77)$$

Para uma reestruturação de dimensão 5:

$$X_k = \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \\ x^{(4)} \\ x^{(5)} \end{pmatrix} \rightarrow f \rightarrow f_x \text{ e } Y_k = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ y^{(4)} \\ y^{(5)} \end{pmatrix} \rightarrow f \rightarrow f_y \text{ e } Z_k = \begin{pmatrix} z^{(1)} \\ z^{(2)} \\ z^{(3)} \\ z^{(4)} \\ z^{(5)} \end{pmatrix} \rightarrow f \rightarrow f_z \quad (78)$$

Para uma reestruturação de dimensão 6:

$$X_k = \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \\ x^{(4)} \\ x^{(5)} \\ x^{(6)} \end{pmatrix} \rightarrow f \rightarrow f_x \text{ e } Y_k = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ y^{(4)} \\ y^{(5)} \\ y^{(6)} \end{pmatrix} \rightarrow f \rightarrow f_y \text{ e } Z_k = \begin{pmatrix} z^{(1)} \\ z^{(2)} \\ z^{(3)} \\ z^{(4)} \\ z^{(5)} \\ z^{(6)} \end{pmatrix} \rightarrow f \rightarrow f_z \quad (79)$$

Para uma reestruturação de dimensão 7:

$$X_k = \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \\ x^{(4)} \\ x^{(5)} \\ x^{(6)} \\ x^{(7)} \end{pmatrix} \rightarrow f \rightarrow f_x \text{ e } Y_k = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ y^{(4)} \\ y^{(5)} \\ y^{(6)} \\ y^{(7)} \end{pmatrix} \rightarrow f \rightarrow f_y \text{ e } Z_k = \begin{pmatrix} z^{(1)} \\ z^{(2)} \\ z^{(3)} \\ z^{(4)} \\ z^{(5)} \\ z^{(6)} \\ z^{(7)} \end{pmatrix} \rightarrow f \rightarrow f_z \quad (80)$$

Para uma reestruturação de dimensão 8:

$$X_k = \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \\ x^{(4)} \\ x^{(5)} \\ x^{(6)} \\ x^{(7)} \\ x^{(8)} \end{pmatrix} \rightarrow f \rightarrow f_x \text{ e } Y_k = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ y^{(4)} \\ y^{(5)} \\ y^{(6)} \\ y^{(7)} \\ y^{(8)} \end{pmatrix} \rightarrow f \rightarrow f_y \text{ e } Z_k = \begin{pmatrix} z^{(1)} \\ z^{(2)} \\ z^{(3)} \\ z^{(4)} \\ z^{(5)} \\ z^{(6)} \\ z^{(7)} \\ z^{(8)} \end{pmatrix} \rightarrow f \rightarrow f_z \quad (81)$$

Nos esquemas anteriores, é possível observar que cada vetor correspondente a um dado de entrada X_k, Y_k, Z_k será introduzido numa função f e cujo resultado f_x, f_y, f_z corresponde a um valor da trajetória previsto.

3.6. Implementação do Algoritmo LS-SVM

Os resultados obtidos para a previsão de trajetórias tiveram origem em vários parâmetros regularizadores γ . A função de progressão utilizada para este parâmetro foi a mesma $\gamma_{k+1} = \gamma_k \times 1.2$, e pode ser analisada graficamente na Figura 40.

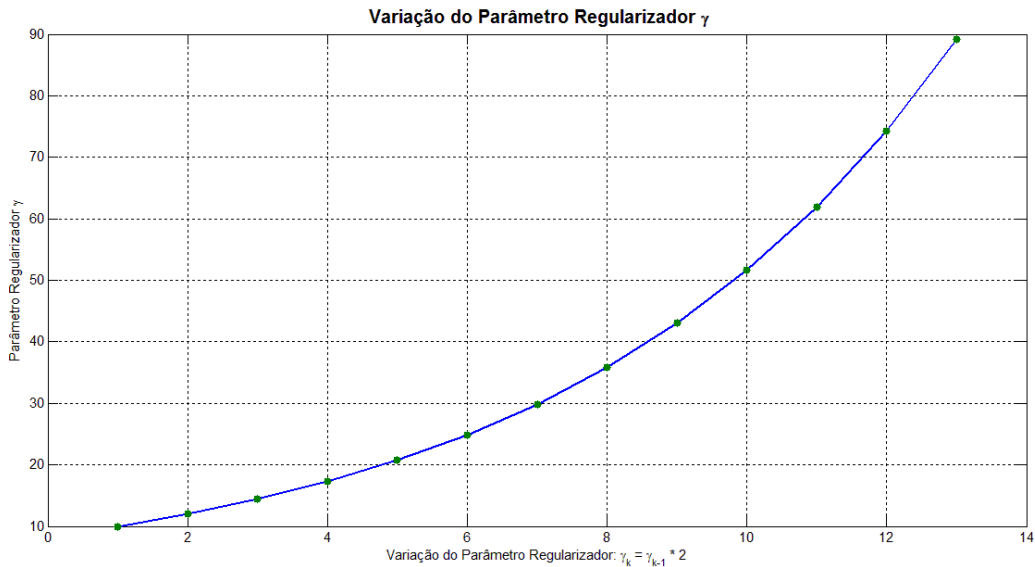


Figura 40 - Variação do Parâmetro Regularizador e Valores Utilizados

As funções kernel utilizadas encontram-se na família dos modelos RBF, e são elas: Gaussiana, Exponencial, Laplaciana, Rational Quadratic, Multiquádrica e Multiquádrica Inversa. Reorganizando as expressões dessas mesmas funções, a função kernel K ou $\varphi(X)_k$, $\varphi(Y)_k$ e $\varphi(Z)_k$ são definidas como:

Para uma Função Gaussiana:

$$\begin{aligned} \varphi(X_k) &= \exp\left(-\frac{\|X - X_k\|^2}{2\sigma^2}\right) \\ \varphi(Y_k) &= \exp\left(-\frac{\|Y - Y_k\|^2}{2\sigma^2}\right) \\ \varphi(Z_k) &= \exp\left(-\frac{\|Z - Z_k\|^2}{2\sigma^2}\right) \end{aligned} \tag{82}$$

Para uma Função Exponencial:

$$\begin{aligned} \varphi(X_k) &= \exp\left(-\frac{\|X - X_k\|}{2\sigma^2}\right) \\ \varphi(Y_k) &= \exp\left(-\frac{\|Y - Y_k\|}{2\sigma^2}\right) \\ \varphi(Z_k) &= \exp\left(-\frac{\|Z - Z_k\|}{2\sigma^2}\right) \end{aligned} \tag{83}$$

Para uma Função de Laplace:

$$\begin{aligned} \varphi(X_k) &= \exp\left(-\frac{\|X - X_k\|}{\sigma}\right) \\ \varphi(Y_k) &= \exp\left(-\frac{\|Y - Y_k\|}{\sigma}\right) \\ \varphi(Z_k) &= \exp\left(-\frac{\|Z - Z_k\|}{\sigma}\right) \end{aligned} \tag{84}$$

Para uma Função Rational Quadratic:

$$\begin{aligned}\varphi(X_k) &= 1 - \frac{\|X - X_k\|^2}{\|X - X_k\|^2 + \eta_x^2} \\ \varphi(Y_k) &= 1 - \frac{\|Y - Y_k\|^2}{\|Y - Y_k\|^2 + \eta_y^2} \\ \varphi(Z_k) &= 1 - \frac{\|Z - Z_k\|^2}{\|Z - Z_k\|^2 + \eta_z^2}\end{aligned}\tag{85}$$

Para uma Função Multiquádrica:

$$\begin{aligned}\varphi(X_k) &= \sqrt{\eta_x^2 + \|X - X_k\|^2} \\ \varphi(Y_k) &= \sqrt{\eta_y^2 + \|Y - Y_k\|^2} \\ \varphi(Z_k) &= \sqrt{\eta_z^2 + \|Z - Z_k\|^2}\end{aligned}\tag{86}$$

Para uma Função Multiquádrica Inversa:

$$\begin{aligned}\varphi(X_k) &= \frac{1}{\sqrt{\eta_x^2 + \|X - X_k\|^2}} \\ \varphi(Y_k) &= \frac{1}{\sqrt{\eta_y^2 + \|Y - Y_k\|^2}} \\ \varphi(Z_k) &= \frac{1}{\sqrt{\eta_z^2 + \|Z - Z_k\|^2}}\end{aligned}\tag{87}$$

Tal como mencionado na listagem das funções RBF, o parâmetro σ desempenha um papel importante na afinação das três primeiras funções (Gaussiana, Exponencial e Laplaciana). Mais à frente, será feita uma comparação entre os vários valores utilizados para este parâmetro e os resultados obtidos para cada um deles. No caso das últimas três funções RBF, o cálculo do parâmetro η (eta) é efetuado separadamente para cada uma das coordenadas, e assume a seguinte expressão:

$$\begin{aligned}\eta_x &= \max_k\{\|X_{k+1} - X_k\|\} \\ \eta_y &= \max_k\{\|Y_{k+1} - Y_k\|\} \\ \eta_z &= \max_k\{\|Z_{k+1} - Z_k\|\}\end{aligned}\tag{88}$$

Assim, as funções $\varphi(X)$, $\varphi(Y)$, $\varphi(Z)$ tomam a forma: $\varphi(X) = (\varphi(X_1), \varphi(X_2), \dots, \varphi(X_N))^T$, $\varphi(Y) = (\varphi(Y_1), \varphi(Y_2), \dots, \varphi(Y_N))^T$ e $\varphi(Z) = (\varphi(Z_1), \varphi(Z_2), \dots, \varphi(Z_N))^T$.

Como se esperava, as funções RBF têm em conta as distâncias entre os dados do conjunto inicial e os parâmetros a serem calculados. Esta afirmação pode ser comprovada visualmente na Figura 34.

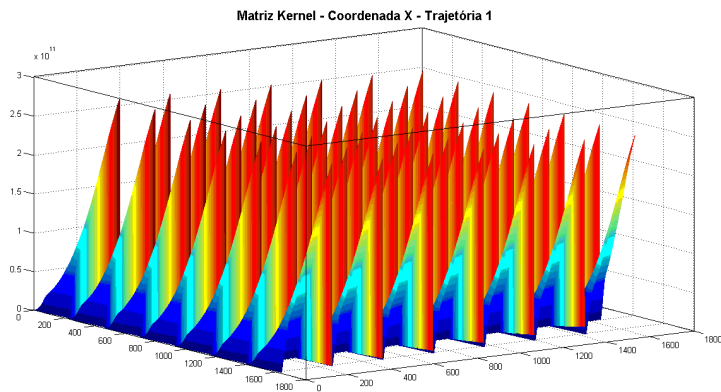


Figura 41 - Exemplo de uma Função Kernel para a Trajetória 1

A título de exemplo, foi utilizada uma matriz kernel de uma função Multiquádrica de dimensão 7 para a Trajetória 1. É possível comparar a figura anterior com a Figura 6 e registrar as semelhanças entre elas. Mais uma vez, esta figura demonstra a distinção entre as influências que os dados têm ao longo da simulação.

Por fim, são obtidos os resultados do sistema de equações presente na Equação 38. Devido à dimensão dos resultados, estes não podem ser apresentados sob forma numérica, no entanto, é possível observar a variação de algumas soluções nas figuras seguintes:

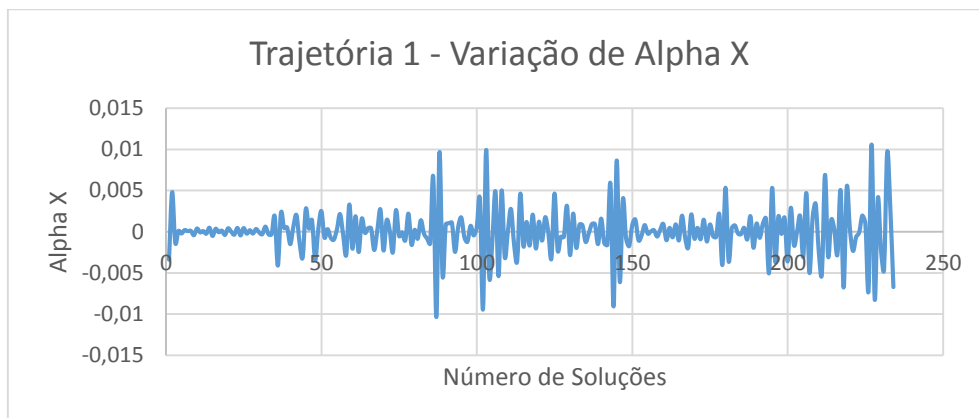


Figura 42 - Exemplo de uma das soluções alpha para a coordenada X - Trajetória 1

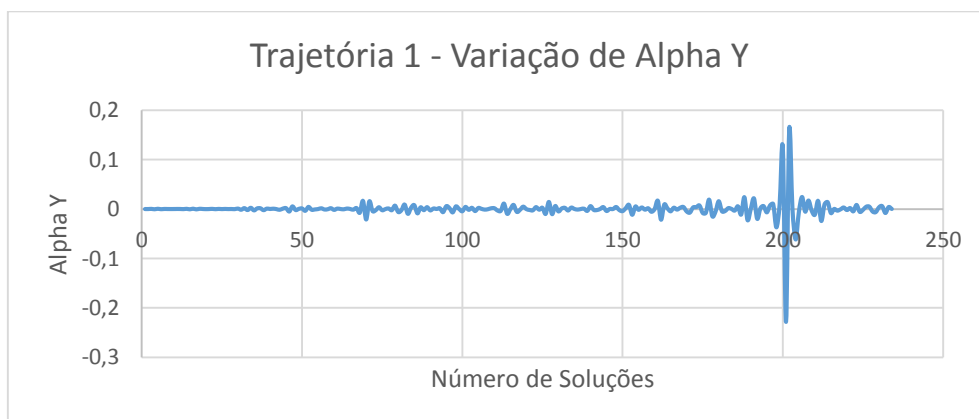


Figura 43 - Exemplo de uma das soluções alpha para a coordenada Y - Trajetória 1

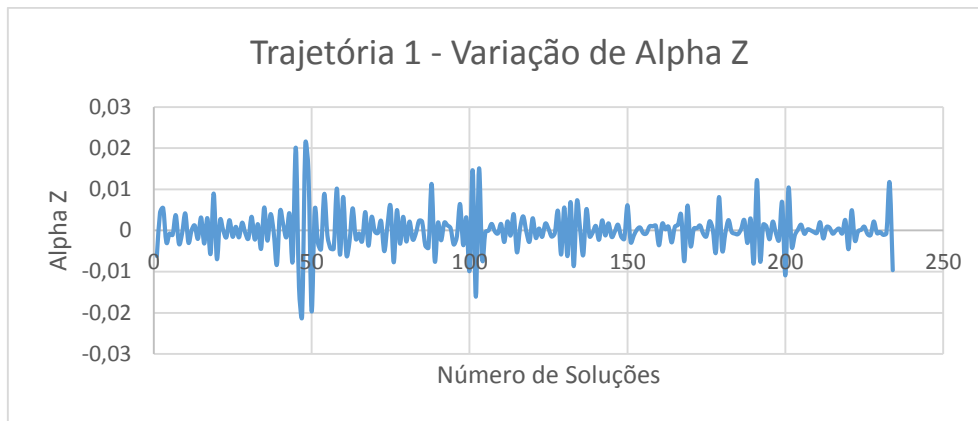


Figura 44 - Exemplo de uma das soluções alpha para a coordenada Z - Trajetória 1

Dos gráficos anteriores, constata-se que o número de soluções coincide com o número total de pontos da trajetória (por coordenada) que tinha sido gerada anteriormente. Antes da apresentação da previsão final da trajetória, são apresentados alguns resultados que visam comparar o desempenho das diferentes funções RBF, tendo em conta a dimensão da reestruturação dos dados e as amplitudes dos erros gaussianos implementados.

3.7. Cálculo do Erro Médio

Após a implementação do algoritmo e obtenção dos resultados, é efetuado um cálculo do erro médio de cada medição para saber qual o modelo ideal a utilizar para previsão de outros resultados. O método utilizado para o cálculo do erro foi o RMSE (*Root Mean Square Error*) visto ser bastante utilizado para medir a diferença entre valores previstos por um determinado modelo e valores observados diretamente do seu ambiente.

Em problemas de regressão, o erro médio (ou RMSE) de um resultado \hat{y}_t em comparação com a variável dependente y é calculado para n resultados. Neste problema, os resultados \hat{y}_t são as classificações previstas para um dado de entrada x_t e y_t o valor da correspondência real do dado de entrada x_t . Os valores de \hat{y}_t são obtidos através da Equação 44 para os dados de entrada com que o algoritmo será treinado. O valor de n é dado consoante o número de previsões feito pelo algoritmo. Assim, n será igual ao número total de dados de saída (classificações) obtidos. O cálculo do erro médio é efetuado através da seguinte fórmula:

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n}} \quad (89)$$

Os valores de erros médios obtidos servem como medida de precisão do algoritmo, mas apenas quando analisando erros entre modelos diferentes para uma determinada variável e não entre variáveis diferentes, isto porque, o erro médio depende da escala em que a variável se encontra quando esta a ser calculado [40].

Após finalizado o cálculo do erro médio por coordenada, pode ser calculado o erro médio global das soluções através da seguinte expressão:

$$RMSE_{global} = \sqrt{RMSE_x^2 + RMSE_y^2 + RMSE_z^2} \quad (90)$$

onde $RMSE_x$, $RMSE_y$ e $RMSE_z$ representam os erros de cada uma das coordenadas da trajetória da aeronave prevista pelo algoritmo. De seguida, é feita uma extensa comparação entre os vários resultados obtidos.

3.8. Comparação de Resultados

Todas as execuções do programa foram efetuadas utilizando o mesmo computador, com a mesma carga computacional. Em cada simulação, o computador efetuava apenas as tarefas do programa, sem qualquer tipo de programas ou aplicações adicionais. Podem ser consultados alguns dados do computador utilizado na tabela seguinte:

Tabela 15 - Dados do computador utilizado na simulação

Windows Edition:	Sistema:
Windows 7 Home Premium Copyright © 2009 Microsoft Corporation	Fabricante: Toshiba Processador: Intel® Core™ i7-3610QM CPU @ 2.30GHz Memória instalada (RAM): 8.00 GB Tipo de sistema: Sistema Operativo de 64 bits

Após terem sido aplicadas todas as funções RBF, com todas as reestruturações dimensionais possíveis e de ter sido feito um pequeno estudo para tentar obter uma melhor afinação de alguns parâmetros definidos pelo utilizador, foram registados todos os tempos de execução do programa para todas as combinações possíveis. Pode ser consultada uma listagem com maior detalhe dos tempos totais de execução no Anexo VI.

Primeiro, na tabela correspondente ao estudo dos diferentes parâmetros σ , foram assinalados aqueles que obtiveram um erro inferior para as respetivas funções RBF. Como referido no Capítulo 3.3, o estudo comparativo entre as várias funções RBF e os seus parâmetros, foi apenas realizado para a Trajetória 1, dada a sua carga computacional inferior. Para encontrar um parâmetro σ ideal, fixou-se uma reestruturação de dimensão 6 para as três funções RBF. Assim, para as funções Gaussiana, Exponencial e Laplaciana, obteve-se:

- Gaussiana: $\sigma = 1250$;
- Exponencial: $\sigma = 70$;
- Laplaciana: $\sigma = 4100$.

Mais uma vez, pode ser consultada uma tabela com maior detalhe no Anexo VI, no entanto, o número de total de parâmetros σ testado foi o seguinte:

- Função Gaussiana: $\sigma = 1250$
 $\sigma_{Gauss} = [0.1; 7.5; 250; 1000; \mathbf{1250}; 1500; 1700; 1900; 2350; 3500; 4100; 5000; 6000]$
- Função Exponencial: $\sigma = 70$
 $\sigma_{Expon} = [30; \mathbf{70}; 100; 250; 400; 700; 1400]$
- Função Laplaciana: $\sigma = 4100$
 $\sigma_{Lap} = [2; 20; 1000; 1500; 2000; 3200; 3700; \mathbf{4100}; 5000]$

Quanto aos tempos de execução das diferentes funções RBF, de acordo com cada uma das reestruturações diferentes, obtiveram-se os seguintes resultados:

Tabela 16 - Tempos de Execução do Programa

Função RBF	Multiquádrica	Multiquádrica Inversa	Rational Quadratic	Gaussiana	Exponencial	Laplaciana
Tempo Total [s]	9243.689743	9546.82821	9120.985559	21603.976241	8888.34909	9913.042009
Tempo Médio [s]	1540.614957	1591.1543035	1520.164259	3600.6627068	1481.391515	1652.173668

Da tabela anterior, é possível concluir que, a nível de tempo de execução, a função Exponencial demonstra o tempo de execução inferior, tendo assim uma carga computacional inferior em relação a todas as outras, enquanto que a função Gaussiana se encontra no extremo oposto, necessitando quase do triplo do tempo de processamento para levar a cabo as mesmas tarefas que as outras funções. De registar também, que da tabela podem ser retirados os tempos de execução individuais mais rápidos:

- Mínimo: Função Multiquádrica Inversa: $t = 1451.476922$ s
- Máximo: Função Gaussiana: $t = 3706.126090$ s

Os valores obtidos para os tempos de execução estão sujeitos a alguma subjetividade, visto que, apesar de o programa ter sido executado sob condições idênticas para todos os casos, existem sempre alguns picos de processamento que podem influenciar o tempo de cálculo das funções em cada programa.

Por fim, foram comparados os erros obtidos para cada uma das funções, tendo em conta as diferentes reestruturações e as diferentes amplitudes de erro gaussiano:

Tabela 17 - Erros Mínimos e Máximos obtidos - Trajetória 1

Função Multiquádrica					
Dimensão 3	Erro Mín.: 10,9519406369818	Amp =	20	$\gamma =$	61,917364
	Erro Máx.: 290,406581098354	Amp =	320	$\gamma =$	61,91736
Dimensão 4	Erro Mín.: 10,971692406303	Amp =	1280	$\gamma =$	89,161004
	Erro Máx.: 127,238541980781	Amp =	5	$\gamma =$	17,28
Dimensão 5	Erro Mín.: 2,17509005355091	Amp =	5	$\gamma =$	61,917364
	Erro Máx.: 66,0798818373487	Amp =	10	$\gamma =$	29,85984
Dimensão 6	Erro Mín.: 0,760815862750078	Amp =	2560	$\gamma =$	61,917364
	Erro Máx.: 31,9873413197809	Amp =	80	$\gamma =$	29,85984
Dimensão 7	Erro Mín.: 0,481770558303577	Amp =	2560	$\gamma =$	51,5978
	Erro Máx.: 20,3991014541713	Amp =	160	$\gamma =$	10
Dimensão 8	Erro Mín.: 0,63776955532814	Amp =	10	$\gamma =$	61,917364
	Erro Máx.: 14,0685056989452	Amp =	160	$\gamma =$	14,4

Função Multiquádrica Inversa

Dimensão 3	Erro Mín.: 91374,65217	Amp =	80	$\gamma =$	89,161004
	Erro Máx.: 91599,50188	Amp =	1280	$\gamma =$	10
Dimensão 4	Erro Mín.: 90789,77598	Amp =	10	$\gamma =$	89,161004
	Erro Máx.: 90971,66658	Amp =	2560	$\gamma =$	10
Dimensão 5	Erro Mín.: 90185,20531	Amp =	640	$\gamma =$	89,161004
	Erro Máx.: 90357,06203	Amp =	1280	$\gamma =$	10
Dimensão 6	Erro Mín.: 89597,69793	Amp =	5	$\gamma =$	89,161004
	Erro Máx.: 89762,97869	Amp =	2560	$\gamma =$	10
Dimensão 7	Erro Mín.: 88999,87059	Amp =	80	$\gamma =$	89,161004
	Erro Máx.: 89128,78658	Amp =	1280	$\gamma =$	10
Dimensão 8	Erro Mín.: 88408,72946	Amp =	5	$\gamma =$	89,161
	Erro Máx.: 88520,67517	Amp =	2560	$\gamma =$	10

Função Rational Quadratic

Dimensão 3	Erro Mín.: 287,5399912	Amp =	20	$\gamma =$	89,161004
	Erro Máx.: 1563,569366	Amp =	2560	$\gamma =$	10
Dimensão 4	Erro Mín.: 257,4504629	Amp =	40	$\gamma =$	89,161004
	Erro Máx.: 1721,401916	Amp =	5	$\gamma =$	10
Dimensão 5	Erro Mín.: 220,6443663	Amp =	1280	$\gamma =$	89,161004
	Erro Máx.: 1925,276769	Amp =	5	$\gamma =$	10
Dimensão 6	Erro Mín.: 232,6035132	Amp =	640	$\gamma =$	89,161004
	Erro Máx.: 2121,326027	Amp =	10	$\gamma =$	10
Dimensão 7	Erro Mín.: 215,4460804	Amp =	1280	$\gamma =$	89,161004
	Erro Máx.: 2297,336863	Amp =	5	$\gamma =$	10
Dimensão 8	Erro Mín.: 199,2091925	Amp =	2560	$\gamma =$	89,161
	Erro Máx.: 2459,150564	Amp =	5	$\gamma =$	10

Função Gaussiana

Dimensão 3	Erro Mín.: 535,1186788	Amp =	640	$\gamma =$	89,161004
	Erro Máx.: 4461,81366	Amp =	2560	$\gamma =$	10
Dimensão 4	Erro Mín.: 528,4485491	Amp =	1280	$\gamma =$	89,161004
	Erro Máx.: 4829,908247	Amp =	2560	$\gamma =$	10
Dimensão 5	Erro Mín.: 515,6656255	Amp =	640	$\gamma =$	89,161
	Erro Máx.: 5408,572652	Amp =	2560	$\gamma =$	10
Dimensão 6	Erro Mín.: 555,2920904	Amp =	640	$\gamma =$	89,161004
	Erro Máx.: 5780,631671	Amp =	2560	$\gamma =$	10
Dimensão 7	Erro Mín.: 577,537979	Amp =	80	$\gamma =$	89,161004
	Erro Máx.: 6286,955892	Amp =	2560	$\gamma =$	10
Dimensão 8	Erro Mín.: 599,8367775	Amp =	80	$\gamma =$	89,161004
	Erro Máx.: 6539,030774	Amp =	2560	$\gamma =$	10

Função Exponencial

Dimensão 3	Erro Mín.: 265,2209791	Amp =	160	$\gamma =$	89,161004
	Erro Máx.: 1166,743414	Amp =	2560	$\gamma =$	10
Dimensão 4	Erro Mín.: 252,5394134	Amp =	40	$\gamma =$	89,161004
	Erro Máx.: 1077,063041	Amp =	2560	$\gamma =$	10
Dimensão 5	Erro Mín.: 239,9647511	Amp =	640	$\gamma =$	89,161004
	Erro Máx.: 1105,124954	Amp =	2560	$\gamma =$	10
Dimensão 6	Erro Mín.: 231,1619897	Amp =	80	$\gamma =$	89,161004
	Erro Máx.: 1028,1608	Amp =	2560	$\gamma =$	10
Dimensão 7	Erro Mín.: 222,6117739	Amp =	40	$\gamma =$	89,161004
	Erro Máx.: 1005,633901	Amp =	2560	$\gamma =$	10
Dimensão 8	Erro Mín.: 211,5632481	Amp =	1280	$\gamma =$	89,161
	Erro Máx.: 981,4236313	Amp =	2560	$\gamma =$	10

Função Laplaciana

Dimensão 3	Erro Mín.: 176,7522973	Amp =	640	$\gamma =$	89,161004
	Erro Máx.: 1272,516117	Amp =	2560	$\gamma =$	10
Dimensão 4	Erro Mín.: 170,0855177	Amp =	640	$\gamma =$	89,161004
	Erro Máx.: 1189,587027	Amp =	2560	$\gamma =$	10
Dimensão 5	Erro Mín.: 163,4942971	Amp =	640	$\gamma =$	89,161
	Erro Máx.: 1223,63315	Amp =	2560	$\gamma =$	10
Dimensão 6	Erro Mín.: 165,4559466	Amp =	2560	$\gamma =$	89,161004
	Erro Máx.: 1249,807938	Amp =	2560	$\gamma =$	10
Dimensão 7	Erro Mín.: 165,5581417	Amp =	1280	$\gamma =$	89,161004
	Erro Máx.: 1332,189111	Amp =	2560	$\gamma =$	10
Dimensão 8	Erro Mín.: 165,2484019	Amp =	1280	$\gamma =$	89,161004
	Erro Máx.: 1423,023064	Amp =	2560	$\gamma =$	10

Nas tabelas anteriores, é possível observar os erros mínimos e máximos para cada dimensão (de 3 a 8) e para cada uma das funções RBF utilizadas. Os valores da amplitude do erro gaussiano e do parâmetro regularizador γ para os quais foram obtidos, também estão assinalados. Os erros mínimos e máximos globais, para cada uma das funções RBF estão assinalados a verde e a vermelho respetivamente. Após o cruzamento de resultados entre todos os erros mínimos e os erros máximos, pode concluir-se que os erros mínimos e máximos globais foram:

- Erro Mín. = 0,481770558303577 para uma Função Multiquádrica - Dimensão 7 - com $Amp_{erro} = 2560$ e $\gamma = 51,5978$;
- Erro Máx. = 91599,50188 para uma Função Multiquádrica Inversa - Dimensão 3 - com $Amp_{erro} = 1280$ e $\gamma = 10$.

As tabelas de erro completas, para cada uma das coordenadas e para a previsão global, para todas as funções RBF e todas as reestruturações possuem mais de uma centena de páginas, o que impossibilita a consulta como anexo no final deste trabalho.

Podem ser retiradas mais algumas conclusões a partir da Tabela 17, tais como, a comparação entre erros mínimos e máximos para cada uma das dimensões, respectivas a cada função RBF, e a comparação entre os parâmetros regularizadores γ e os erros obtidos também para cada uma das funções RBF. As tabelas seguintes ilustram essas comparações:

Tabela 18 - Erros obtidos vs. Dimensão dos Resultados

Função Multiquádrica	Função Multiquádrica Inversa	Função Rational Quadratic
Dimensão 3 -> Erro Máximo	Dimensão 3 -> Erro Máximo	Dimensão 8 -> Erro Máximo
Dimensão 7 -> Erro Mínimo	Dimensão 7 -> Erro Mínimo	Dimensão 8 -> Erro Mínimo
Função Gaussiana	Função Exponencial	Função Laplaciana
Dimensão 8 -> Erro Máximo	Dimensão 3 -> Erro Máximo	Dimensão 8 -> Erro Máximo
Dimensão 5 -> Erro Mínimo	Dimensão 8 -> Erro Mínimo	Dimensão 5 -> Erro Mínimo

Tabela 19 - Erros obtidos vs. Parâmetro Regularizador

Parâmetro Regularizador	Erros Obtidos	Funções RBF
$\gamma = 10$	Erro Máximo: 5 vezes	Multiquádrica Inversa; Rational Quadratic; Gaussiana; Exponencial; Laplaciana
$\gamma = 61,917364224$	Erro Máximo: 1 vez	Multiquádrica
$\gamma = 51,59780352$	Erro Mínimo: 1 vez	Multiquádrica
$\gamma = 89,161004482$	Erro Mínimo: 5 vezes	Multiquádrica Inversa; Rational Quadratic; Gaussiana; Exponencial; Laplaciana

Analisando o valor obtido para o erro mínimo, com a função Multiquádrica, é possível afirmar que a previsão efetuada pelo programa é quase perfeita. Para se analisar em detalhe a evolução dos erros por coordenada (X, Y e Z) e do erro global, podem ser consultados os gráficos seguintes, que tiveram como base a função Multiquádrica, numa reestruturação de dimensão 7 e uma amplitude de erro gaussiano $Amp_{erro} = 2560$, devido ao fato de ter sido a combinação que obteve o menor valor de erro.

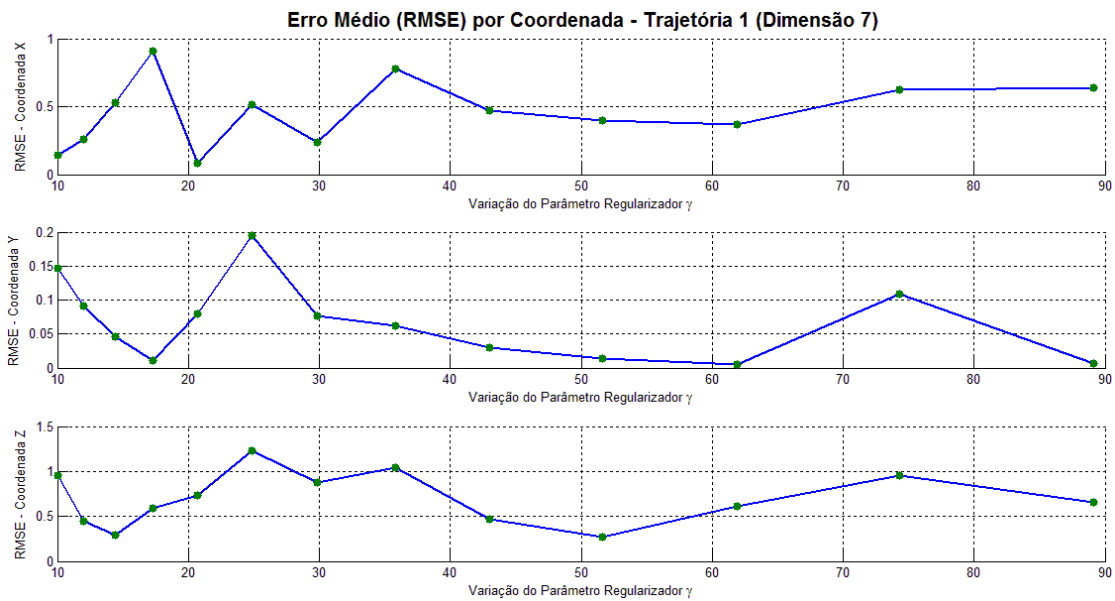


Figura 45 - Erros obtidos por Coordenada para a função Multiquádrica - Trajetória 1

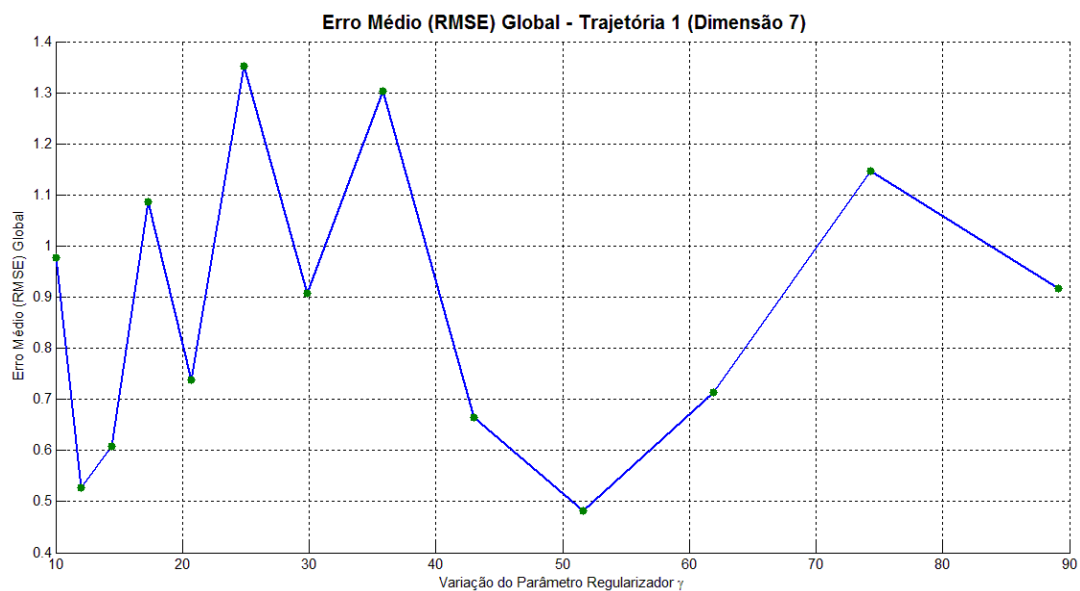


Figura 46 - Erros Global obtido para a função Multiquádrica - Trajetória 1

Mais uma vez, da figura anterior pode ser retirado que o valor do erro mínimo se registra para um parâmetro regularizador $\gamma = 51,5978$. Seguidamente, serão apresentadas as trajetórias previstas pelo algoritmo e será feita uma comparação entre as trajetórias geradas e as soluções do programa.

3.9. Trajetórias Previstas pelo Algoritmo LS-SVM

Numa fase final do programa, após apuramento da combinação ideal que registra o valor mínimo de erro, exposta no capítulo anterior, é possível traçar um gráfico com a trajetória prevista:

Trajétória 1 - Original vs. Trajetória 1 - Algoritmo LS-SVM (Dimensão 7)

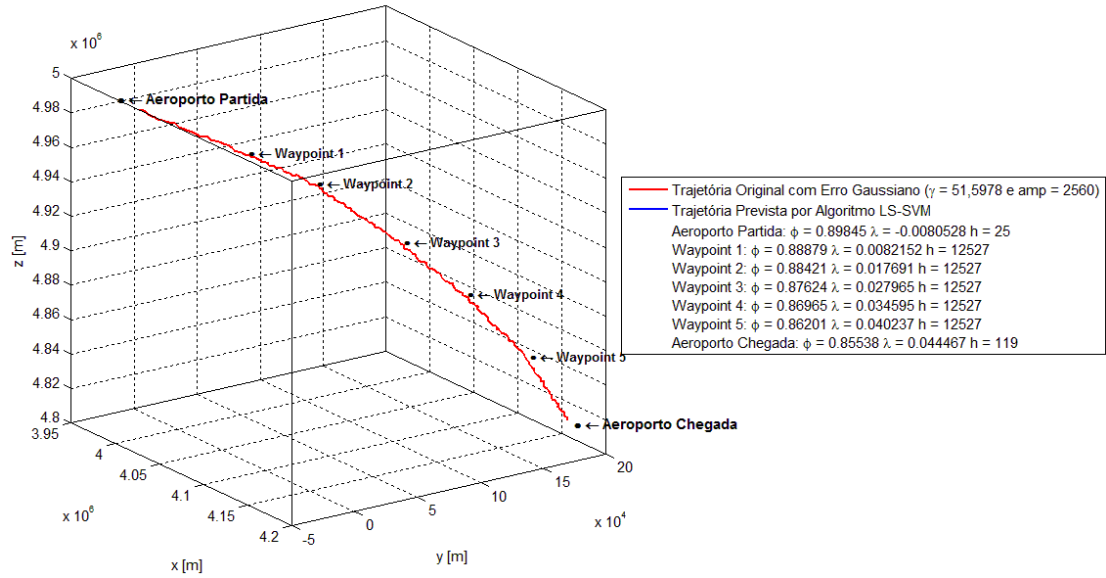


Figura 47 - Trajetória 1: Original vs. Previsão do Algoritmo LS-SVM

Aplicando os dados da figura anterior numa representação na superfície terrestre, obtém-se:



Figura 48 - Trajetória 1: Original vs. Previsão do Algoritmo LS-SVM na superfície terrestre

Depois de ter sido determinada a função RBF e a dimensão da reestruturação que resultaria num valor de erro mínimo, o algoritmo pode ser aplicado às outras duas trajetórias, mencionadas no Capítulo 3.3. Após a aplicação dos dados da Trajetória 2, obtiveram-se os seguintes resultados:

Trajatória 2 - Original vs. Trajetória 2 - Algoritmo LS-SVM (Dimensão 7)

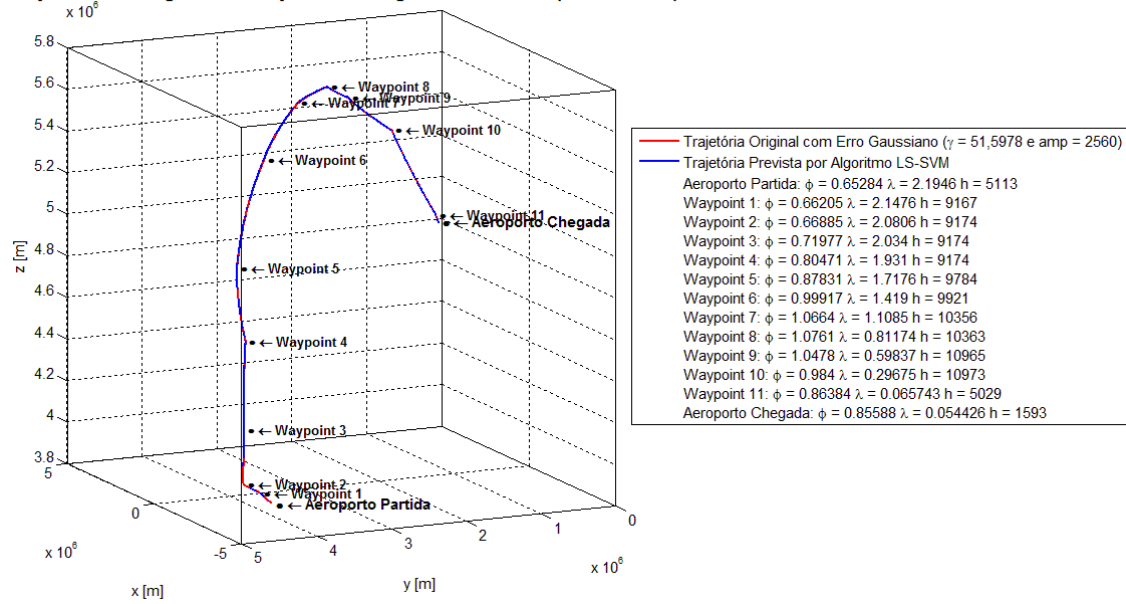


Figura 49 - Trajetória 2: Original vs. Previsão do Algoritmo LS-SVM

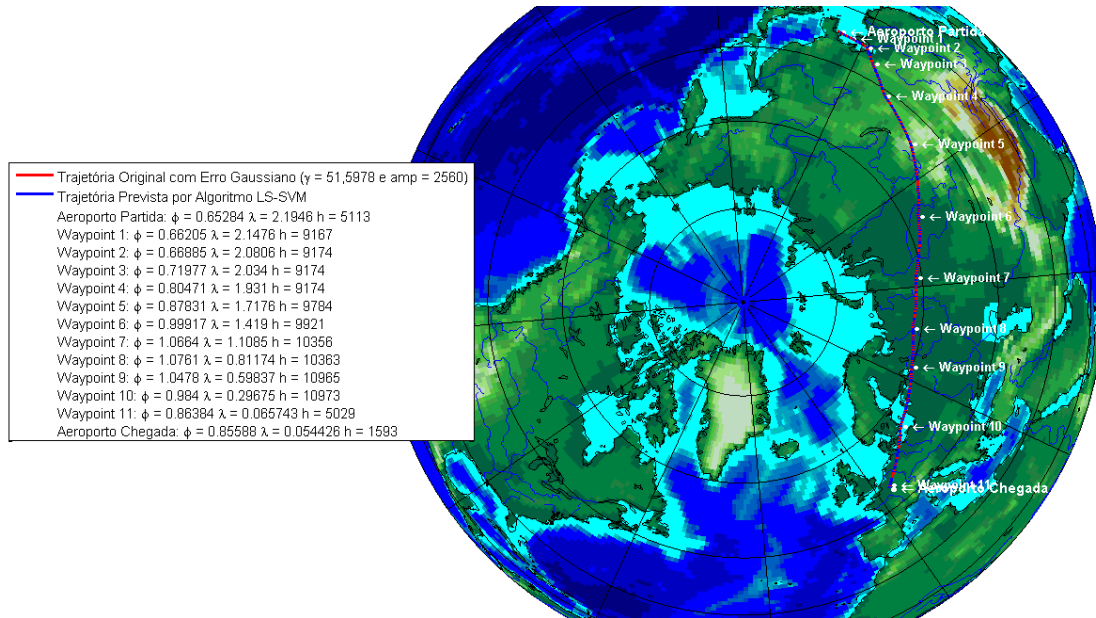


Figura 50 - Trajetória 2: Original vs. Previsão do Algoritmo LS-SVM na superfície terrestre

E por fim, com os dados da Trajetória 3, foi possível obter a seguinte previsão:

Trajatória 3 - Original vs. Trajetória 3 - Algoritmo LS-SVM (Dimensão 7)

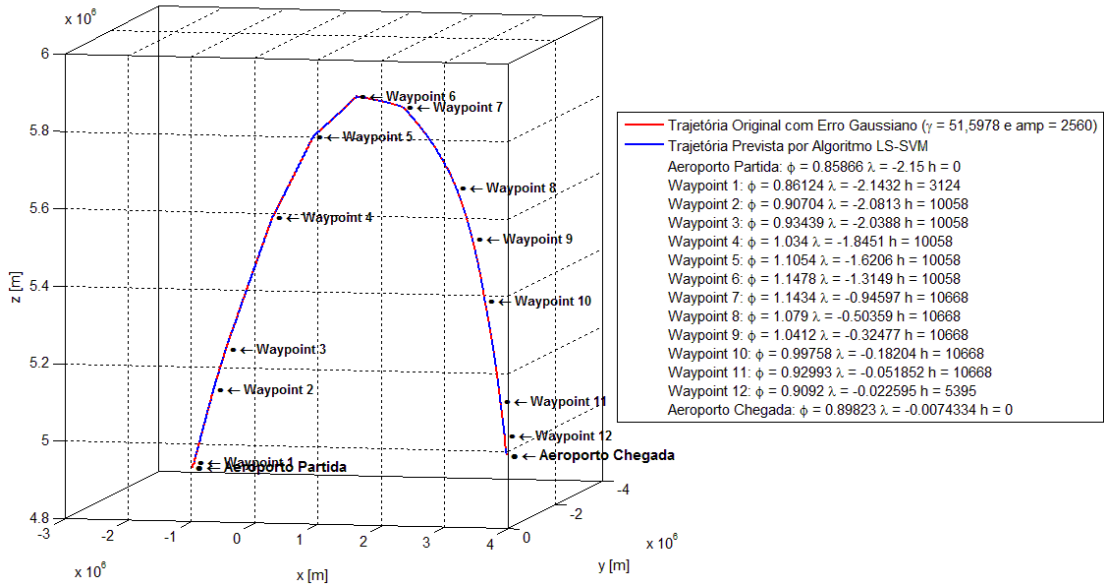


Figura 51 - Trajetória 3: Original vs. Previsão do Algoritmo LS-SVM

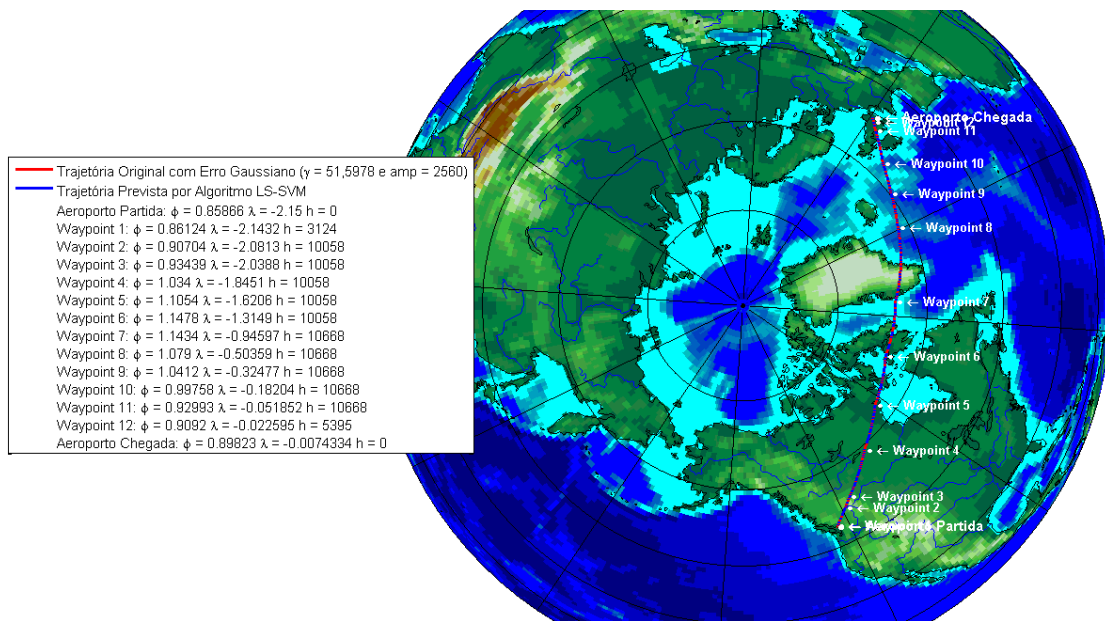


Figura 52 - Trajetória 3: Original vs. Previsão do Algoritmo LS-SVM na superfície terrestre

A partir da análise das figuras acima, é possível constatar que as previsões efetuadas pelo algoritmo LS-SVM são quase coincidentes com as trajetórias criadas originalmente. Esta afirmação é suportada pelo fato de os valores do erro global serem próximos de zero, para uma função RBF Multiquádrica e para uma reestruturação de dimensão 7. No entanto, a vantagem particular da implementação desta metodologia ainda não foi explorada, visto que após a previsão da trajetória efetuada pelo algoritmo, o utilizador tem a possibilidade de construir uma função idêntica à presente na Equação 44, o que lhe permite obter dados da trajetória para os quais não possuía qualquer tipo de informação anteriormente. Assim, desde que os dados que procure estejam dentro da trajetória real percorrida pela aeronave (entre o

aeroporto de partida e o aeroporto de chegada), o utilizador tem a possibilidade de saber a posição em qualquer ponto da rota, mesmo que este não tenha sido definido na listagem inicial dos *waypoints*. Esta particularidade irá ser desenvolvida no tópico seguinte.

3.10. Criação da Função da Trajetória

Como exposto nas Tabelas 9, 10, 11, 12, 13 e 14 no Capítulo 3.5, os dados originais das trajetórias são reestruturados de acordo com um valor de dimensão definido pelo utilizador, para depois ser possível criar um vetor de soluções para efetuar o treino do algoritmo. Após a execução completa do programa, podem agora ser observados alguns exemplos desta reestruturação dos dados entrada:

Tabela 20 - Alguns resultados de exemplo (Trajetória 1 - Dimensão 6)

Posição da Aeronave - Trajetória 1 - Dimensão 6 - Coordenada X						Solução Real [m]	Solução Prevista [m]
3972348,39	3973903,24	3975524,69	3977141,04	3978744,09	3980379,70	3981993,72	3981988,49
3973903,24	3975524,69	3977141,04	3978744,09	3980379,70	3981993,72	3983617,66	3983612,43
3975524,69	3977141,04	3978744,0	3980379,70	3981993,72	3983617,66	3985228,56	3985223,33

Tabela 21 - Alguns resultados de exemplo (Trajetória 1 - Dimensão 3)

Posição da Aeronave - Trajetória 1 - Dimensão 3 - Coordenada Y			Solução Real [m]	Solução Prevista [m]
-31987,96127	-30181,57368	-28321,21301	-26457,92455	-26459,54543
-30181,57368	-28321,21301	-26457,92455	-24595,78106	-24597,36861
-28321,21301	-26457,92455	-24595,78106	-22729,64254	-22731,19572

Tabela 22 - Alguns resultados de exemplo (Trajetória 2 - Dimensão 7)

Posição da Aeronave - Trajetória 2 - Dimensão 7 - Coordenada Z							Solução Real [m]	Solução Prevista [m]
3877460,7	3879335,7	3881301,7	3883247,8	3885218,7	3887191,1	3889154,3	3891118,8	3891501,4
3879335,7	3881301,7	3883247,8	3885218,7	3887191,1	3889154,3	3891118,8	3893070,4	3892265,5
3881301,7	3883247,8	3885218,7	3887191,1	3889154,3	3891118,8	3893070,4	3895028,9	3895187,0

Tabela 23 - Alguns resultados de exemplo (Trajetória 1 - Dimensão 4)

Posição da Aeronave - Trajetória 1 - Dimensão 4 - Coordenada X				Solução Real [m]	Solução Prevista [m]
3986857,83	3988495,566	3990063,47	3991696,003	3993317,61	3993338,092
3988495,566	3990063,47	3991696,003	3993317,61	3994919,255	3994940,695
3990063,47	3991696,003	3993317,61	3994919,255	3996532,423	3996548,479

Poderiam ser apresentados milhares de exemplos relativamente à reestruturação dos dados. No entanto, as tabelas acima permitem visualizar o funcionamento do processo de reestruturação dos dados, bem como a previsão bastante acertada do algoritmo LS-SVM para a maioria dos casos.

Pela análise dos padrões coloridos que estão destacados, é visível que a solução real criada pelo programa através da reestruturação vai fazer parte do vetor de entrada seguinte, ocupando uma posição anterior à que ocupava inicialmente. Este padrão repete-se para qualquer coordenada, em qualquer trajetória, sob qualquer dimensão de reestruturação.

Na coluna à direita das soluções reais (utilizadas pelo programa para treinar o algoritmo), encontra-se as soluções previstas pelo algoritmo LS-SVM. A partir destas soluções, puderam ser traçadas as trajetórias previstas no capítulo anterior. É também com estas soluções, que é efetuado o cálculo do erro para cada uma das coordenadas, que por sua vez, permitirá o cálculo do erro global da trajetória prevista.

Nesta fase final, após a obtenção de todos os resultados, pode finalmente ser implementado o modelo descrito pela Equação 44, e aqui simplificado da seguinte forma:

$$f(x) = \sum_{i=1}^m \alpha_i^* \varphi(x_i)^T \varphi(x) + b^* \quad (91)$$

onde α_i^* e b^* representam as soluções do sistema descrito pela Equação 38, e $\varphi(x_i)^T$ a matriz gerada pela resolução da função RBF escolhida na implementação do algoritmo. O parâmetro $\varphi(x)$ será o resultado dessa mesma função RBF, aplicado às coordenadas que o utilizador desejar determinar.

Tal como mencionado anteriormente, as soluções α_i^* e as matrizes $\varphi(x_i)^T$ que resultam da execução do programa, são de dimensões demasiado elevadas, o que impossibilita a sua transcrição numérica neste trabalho. No entanto, ficam alguns exemplos das infinitas possibilidades de soluções que o algoritmo oferece:

Tabela 24 - Exemplo 1: Previsão de Soluções para Função Multiquádrica

Posição da Aeronave Função Multiquádrica			Solução Prevista [m]		
Coordenada X Original [m]	Coordenada Y Original [m]	Coordenada Z Original [m]	Coordenada X Prevista [m]	Coordenada Y Prevista [m]	Coordenada Z Prevista [m]
4100000	70000	4900000	4104401,423	74394,484	4896945,862

onde as soluções previstas são soluções das equações:

- Coordenada X - $f(4100000) = \sum_{i=1}^m \alpha_{x_i}^* \varphi(x_i)^T \varphi(4100000) + b_x^*$;
- Coordenada Y - $f(70000) = \sum_{i=1}^m \alpha_{y_i}^* \varphi(y_i)^T \varphi(70000) + b_y^*$;
- Coordenada Z - $f(4900000) = \sum_{i=1}^m \alpha_{z_i}^* \varphi(z_i)^T \varphi(4900000) + b_z^*$;

Tabela 25 - Exemplo 2: Previsão de Soluções para Função Multiquádrica

Posição da Aeronave Função Multiquádrica			Solução Prevista [m]		
Coordenada X Original [m]	Coordenada Y Original [m]	Coordenada Z Original [m]	Coordenada X Prevista [m]	Coordenada Y Prevista [m]	Coordenada Z Prevista [m]
4023721	-18135	4830017	4031108,94	-10065,30	4831580,42

onde as soluções previstas são soluções das equações:

- Coordenada X - $f(4023721) = \sum_{i=1}^m \alpha_{x_i}^* \varphi(x_i)^T \varphi(4023721) + b_x^*$;
- Coordenada Y - $f(-18135) = \sum_{i=1}^m \alpha_{y_i}^* \varphi(y_i)^T \varphi(-18135) + b_y^*$;
- Coordenada Z - $f(4830017) = \sum_{i=1}^m \alpha_{z_i}^* \varphi(z_i)^T \varphi(4830017) + b_z^*$;

Tabela 26 - Exemplo 3: Previsão de Soluções para Função Multiquádrica

Posição da Aeronave Função Multiquádrica			Solução Prevista [m]		
Coordenada X Original [m]	Coordenada Y Original [m]	Coordenada Z Original [m]	Coordenada X Prevista [m]	Coordenada Y Prevista [m]	Coordenada Z Prevista [m]
3999641	103478	4930143	4011183,85	107418,04	4927047,82

onde as soluções previstas são soluções das equações:

- Coordenada X - $f(3999641) = \sum_{i=1}^m \alpha_{x_i}^* \varphi(x_i)^T \varphi(3999641) + b_x^*$;
- Coordenada Y - $f(103478) = \sum_{i=1}^m \alpha_{y_i}^* \varphi(y_i)^T \varphi(103478) + b_y^*$;
- Coordenada Z - $f(4930143) = \sum_{i=1}^m \alpha_{z_i}^* \varphi(z_i)^T \varphi(4930143) + b_z^*$;

Tabela 27 - Exemplo 4: Previsão de Soluções para Função Exponencial

Posição da Aeronave Função Exponencial			Solução Prevista [m]		
Coordenada X Original [m]	Coordenada Y Original [m]	Coordenada Z Original [m]	Coordenada X Prevista [m]	Coordenada Y Prevista [m]	Coordenada Z Prevista [m]
4100000	70000	4900000	4104390,99	72471,67	4896591,06

onde as soluções previstas são soluções das equações:

- Coordenada X - $f(4100000) = \sum_{i=1}^m \alpha_{x_i}^* \varphi(x_i)^T \varphi(4100000) + b_x^*$;
- Coordenada Y - $f(70000) = \sum_{i=1}^m \alpha_{y_i}^* \varphi(y_i)^T \varphi(70000) + b_y^*$;
- Coordenada Z - $f(4900000) = \sum_{i=1}^m \alpha_{z_i}^* \varphi(z_i)^T \varphi(4900000) + b_z^*$;

Tabela 28 - Exemplo 5: Previsão de Soluções para Função Gaussiana

Posição da Aeronave Função Gaussiana			Solução Prevista [m]		
Coordenada X Original [m]	Coordenada Y Original [m]	Coordenada Z Original [m]	Coordenada X Prevista [m]	Coordenada Y Prevista [m]	Coordenada Z Prevista [m]
4100000	70000	4900000	4100454,07	72434,42	4897436,08

onde as soluções previstas são soluções das equações:

- Coordenada X - $f(4100000) = \sum_{i=1}^m \alpha_{x_i}^* \varphi(x_i)^T \varphi(4100000) + b_x^*$;
- Coordenada Y - $f(70000) = \sum_{i=1}^m \alpha_{y_i}^* \varphi(y_i)^T \varphi(70000) + b_y^*$;
- Coordenada Z - $f(4900000) = \sum_{i=1}^m \alpha_{z_i}^* \varphi(z_i)^T \varphi(4900000) + b_z^*$;

Tabela 29 - Exemplo 6: Previsão de Soluções para Função Laplaciana

Posição da Aeronave Função Laplaciana			Solução Prevista [m]		
Coordenada X Original [m]	Coordenada Y Original [m]	Coordenada Z Original [m]	Coordenada X Prevista [m]	Coordenada Y Prevista [m]	Coordenada Z Prevista [m]
4100000	70000	4900000	4101534,98	72761,32	4897200,51

onde as soluções previstas são soluções das equações:

- Coordenada X - $f(4100000) = \sum_{i=1}^m \alpha_{x_i}^* \varphi(x_i)^T \varphi(4100000) + b_x^*$;
- Coordenada Y - $f(70000) = \sum_{i=1}^m \alpha_{y_i}^* \varphi(y_i)^T \varphi(70000) + b_y^*$;
- Coordenada Z - $f(4900000) = \sum_{i=1}^m \alpha_{z_i}^* \varphi(z_i)^T \varphi(4900000) + b_z^*$;

As coordenadas foram selecionadas aleatoriamente, tendo apenas de obedecer à seguinte regra: $x_{partida} < x_{previsão} < x_{chegada}$, $y_{partida} < y_{previsão} < y_{chegada}$ e $z_{partida} < z_{previsão} < z_{chegada}$. Ou seja, qualquer previsão terá de estar dentro dos valores da trajetória percorrida pela aeronave.

Este capítulo encerra a demonstração prática da implementação do algoritmo LS-SVM, bem como a demonstração para a previsão de trajetórias 4D. No capítulo seguinte, será feita uma revisão final, bem como tiradas todas as conclusões acerca do trabalho exposto anteriormente.

Capítulo 4

4. Conclusões

O principal objetivo definido inicialmente, de efetuar uma previsão de trajetórias num ambiente de quatro dimensões, recorrendo a métodos de regressão de aprendizagem automática, mais especificamente, algoritmos de Máquina de Suporte Vetorial com base nos Mínimos Quadrados (LS-SVM), foi atingido com sucesso, na medida em que foi possível construir funções que possibilitaram a classificação das trajetórias analisadas em qualquer ponto do seu percurso (desde a descolagem até a aterragem). Apesar de neste trabalho os algoritmos criados terem sido aplicados apenas num tema relacionado com a Engenharia Aeronáutica, através de uma visão mais abrangente, pode ser também observada a interdisciplinaridade desta disciplina, na medida em que os métodos de regressão poderiam ter sido aplicados a qualquer conjunto de dados iniciais (previsão de mercado bolseiro, comportamento de um eletrocardiograma, consumo de combustível, etc.), ou até, aplicando métodos de classificação recorrendo também à teoria LS-SVM.

Num nível mais específico, no primeiro caso ilustrativo, referente à previsão da função seno, é possível observar a exatidão que o método demonstra quando aplicado a uma função relativamente simples. A função Exponencial apresentou o valor mínimo de erro médio nos resultados, enquanto que a função Multiquádrica Inversa apresentou o erro máximo. No entanto, é necessário referir que a função Exponencial, necessita de uma experimentação paramétrica bastante maior que as três primeiras funções em estudo (Multiquádrica, Multiquádrica Inversa e Rational Quadratic), na medida em que precisa da afinação do parâmetro σ , para além da correcção do parâmetro γ , como todas as outras.

Na segunda aplicação do método, ilustrada no capítulo anterior, é possível observar a enorme facilidade com que é possível gerar trajetórias através dos métodos de navegação com *waypoints* 4D. As três trajetórias foram geradas com bastante rapidez, e rapidamente poderiam ter sido alteradas através da mudança dos *waypoints* escolhidos. No entanto, a temática da navegação 4D ainda está numa fase embrionária.

Dentro do âmbito da previsão LS-SVM, a própria estrutura do programa permitiu que as soluções obtidas se aproximassem o mais possível de valores reais, na medida em que foram implementados erros gaussianos nas trajetórias geradas, que vão de encontro com valores de erro produzidos por leitores GPS. A reestruturação dos dados de diferentes dimensões possibilitou uma nova visualização no tratamento de dados. Este facto, aliado à experimentação das diferentes funções RBF, concluiu que a função Multiquádrica desempenhou o melhor papel na aplicação à previsão de trajetórias. Tal como observado anteriormente, este erro mínimo foi obtido para uma reestruturação dos dados de entrada de dimensão 7. Esta reestruturação permitiu a possibilidade dos dados se treinarem a eles próprios. Especificamente, num cenário

de dimensão 7, os dados de entrada foram reestruturados em vetores lineares com 7 elementos, cuja solução atribuída correspondia ao valor de entrada imediatamente a seguir. Para esta dimensão, o valor do erro obtido foi inferior a uma unidade $Erro\ Mín. \approx 0.48177$. Este resultado vem suportar o facto de os algoritmos LS-SVM serem apropriados para a aplicação de previsão de trajetórias.

A nível de desempenho computacional, também se regista que a função Multiquádrica é das que mostra menores tempos de execução, quando comparada com as outras. O menor tempo obtido foi na aplicação da função Exponencial $Tempo\ Mín. = 8888.34909\ s$. No entanto, nesta parte, podem ser tecidas algumas críticas negativas a este método. Como pode ser observado no anexo VI, a média de tempos de execução do programa ronda os 1500 s, o que se aproxima de 30 min. Sendo a ser bastante superior no caso da função Gaussiana. Este facto poderia ser agravado, caso o conjunto de dados de entrada (neste caso, as coordenadas das trajetórias) fosse muito maior. Por outras palavras, quanto mais pormenorizada fosse a geração da trajetória, mais tempo levaria ao algoritmo para executar as suas funções completas. Num ambiente de aplicação em tempo real, seria necessário efetuar algumas correções para que os resultados fossem obtidos com uma maior rapidez.

No penúltimo ponto do capítulo anterior, é possível observar a aplicação do algoritmo LS-SVM a trajetórias um pouco mais elaboradas que a Trajetória 1. Para a execução completa do algoritmo aplicado às Trajetórias 2 e 3, o computador necessitou de um período de tempo bastante superior, o que não permitiu que fosse efetuado o mesmo estudo comparativo de modelos RBF como para a Trajetória 1. No entanto, é visível que a previsão obtida é bastante correta, sendo quase coincidente com a trajetória criada inicialmente.

Por fim, foi possível efetuar a criação de uma nova função que define a trajetória prevista, permitindo a obtenção da posição seguinte da aeronave, recorrendo a soluções obtidas pelo programa. Apesar de se revelar um pouco limitador, na medida em que o algoritmo apenas pode prever uma solução para valores conhecidos da trajetória ($x_{previsão} < x_{min}$ ou $x_{previsão} > x_{max}$), a criação da função permite efetuar um perfil completo da trajetória, podendo o utilizador saber a posição da aeronave em qualquer instante, e não só nos *waypoints* que definiu inicialmente.

4.1. Trabalhos Futuros

Tal como mencionado nos parágrafos anteriores, uma das principais limitações deste projeto foi, sem dúvida, o elevado tempo de execução registado pelos processos computacionais. Nesta aplicação do método LS-SVM, foram feitos todos os processos possíveis para a redução do tempo de execução. Assim, sugere-se que num trabalho futuro possa ser feito um estudo comparativo entre o método LS-SVM e outros métodos de aprendizagem automática dentro e fora da área da Máquina de Suporte Vetorial, que possam apresentar teoremas matemáticos menos complexos.

Pode também registar-se uma limitação, no que toca à comparação de dados que poderiam ter sido utilizados, nomeadamente, a variedade de trajetórias que poderiam ter sido utilizadas (não só num ambiente de quatro dimensões, mas também tridimensional), bem como outro tipo de informação relacionada com a Engenharia Aeronáutica (previsão de consumo de combustível, classificação de imagens de radar, etc.) Fica também como sugestão, uma possível reaplicação destes métodos noutras áreas do interesse aeronáutico.

Numa nota mais abrangente, como foi possível observar no Capítulo 1, é de notar uma enorme interdisciplinaridade na aplicação dos métodos de regressão LS-SVM. Isto deve-se ao fato dos seus princípios matemáticos terem a facilidade de serem aplicados em qualquer tipo de contexto. No mundo atual, é possível ver um enorme leque de aplicações possíveis para este tipo de metodologia, mesmo fora da área da engenharia. Tanto na aplicação de previsão de variações nas cotações da bolsa (área económica), como na previsão do comportamento de um órgão vital de um ser humano (biologia).

De um modo geral, seja qual for o tipo de dados a analisar, o método LS-SVM apresenta uma forte ferramenta de previsão com um menor esforço computacional quando comparado com outros métodos, permitindo assim fazer uma classificação mais pormenorizada dos dados a estudar.

Bibliografia

- [1] S. M. Stigler, "Gauss and the Invention of Least Squares," *Ann. Stat.*, vol. 9, no. 3, pp. 465-474, 1981.
- [2] G. A. Di Meo, A. Cavallo, and S. Chiesa, "Sesar and military: Towards ATM integration," *AIAA/IEEE Digit. Avion. Syst. Conf. - Proc.*, pp. 1-13, 2013.
- [3] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Fourth Edi. Elsevier, Amsterdam, 2003.
- [4] H. Zhang and Y. Zhao, "The performance comparison and analysis of extended Kalman filters for GPS/DR navigation," *Opt. - Int. J. Light Electron Opt.*, vol. 122, no. 9, pp. 777-781, May 2011.
- [5] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. The MIT Press, Cambridge, 2012.
- [6] J. Hua, "Study on the Application of Rough Sets Theory in Machine Learning," *2008 Second Int. Symp. Intell. Inf. Technol. Appl.*, pp. 192-196, Dec. 2008.
- [7] S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, Third Edit. Pearson Education, 1994.
- [8] D. Pleniz, "Theoretical neuroscience: A leak-proof model," *Nat. Phys.*, vol. 6, pp. 717-718, 2010.
- [9] B. Schölkopf and A. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. The MIT Press, Cambridge, 2002.
- [10] A. Smola and B. Schölkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, no. 3, pp. 199-222, 2004.
- [11] H. Byun and S. Lee, "Applications of support vector machines for pattern recognition: A survey," *Pattern Recognit. with Support vector Mach.*, pp. 213-236, 2002.
- [12] C. Wang and S. Chen, "An improved LS-SVM based on SSOR-PCG," *2013 Ninth Int. Conf. Nat. Comput.*, pp. 28-33, Jul. 2013.
- [13] J. a. K. Suykens and J. Vandewalle, "Least Squares Support Vector Machine Classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293-300, 1999.
- [14] B. B. Ekici, "A least squares support vector machine model for prediction of the next day solar insolation for effective use of PV systems," *Measurement*, vol. 50, pp. 255-262, Apr. 2014.
- [15] L. Jiao, L. Bo, and L. Wang, "Fast sparse approximation for least squares support vector machine," *Neural Networks, IEEE Trans.*, vol. 18, no. 3, pp. 685-697, 2007.
- [16] P. Hagelauer and F. Mora-Camino, "A soft dynamic programming approach for on-line aircraft 4d-trajectory optimization," *Eur. J. Oper. Res.*, vol. 107, pp. 87-95, 1998.
- [17] B. S. Ali, W. Y. Ochieng, W. Schuster, A. Majumdar, and T. K. Chiew, "A safety assessment framework for the Automatic Dependent Surveillance Broadcast (ADS-B) system," *Saf. Sci.*, vol. 78, pp. 91-100, 2015.

- [18] Y. Fang and M. Zan, "Study on airworthiness requirement for the position quality of ADS-B system," *Procedia Eng.*, vol. 17, pp. 415-421, 2011.
- [19] J. Dippon, P. Fritz, and M. Kohler, "A statistical approach to case based reasoning, with application to breast cancer data," *Comput. Stat. Data Anal.*, vol. 40, pp. 579-602, 2002.
- [20] L. Györfi, A. Urbán, and I. Vajda, "Kernel-based semi-log-optimal empirical portfolio selection strategies," *Int. J. Theor. Appl. Financ.*, vol. 10, no. 3, pp. 505-516, 2007.
- [21] B. Huyck, K. De Brabanter, F. Logist, J. De Brabanter, J. Van Impe, and B. De Moor, "Identification of a Pilot Scale Distillation Column : A Kernel Based Approach," *World Congr.*, vol. 18, no. 1, pp. 471-476, 2011.
- [22] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, Third Edit. Springer, 2008.
- [23] S. Xu, X. An, X. Qiao, L. Zhu, and L. Li, "Semi-supervised least-squares support vector regression machines," *J Inf Comput Sci*, vol. 6, no. June, pp. 885-892, 2011.
- [24] H. Xing and T. Jin, "Detection of Weak Signal in Chaotic Clutter Using Advanced LS-SVM Regression," *2009 2nd Int. Congr. Image Signal Process.*, pp. 1-5, Oct. 2009.
- [25] V. Vapnik, "Principles of risk minimization for learning theory," *Adv. Neural Inf. Process. Syst.*, vol. 4, pp. 831-838, 1992.
- [26] J. Xie, "Kernel optimization of LS-SVM based on damage detection for smart structures," *2009 2nd IEEE Int. Conf. Comput. Sci. Inf. Technol.*, pp. 406-409, 2009.
- [27] R. Fletcher, *Practical methods of optimization*, Second Edi. Wiley, 2013.
- [28] P. I. Corke, F. Spindler, and F. Chaumette, "Combining Cartesian and polar coordinates in IBVS," *2009 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS 2009*, no. 4, pp. 5962-5967, 2009.
- [29] G. E. P. Box and M. E. Muller, "A Note on the Generation of Random Normal Deviates," *Ann. Math. Stat.*, vol. 29, no. 2, pp. 610-611, 1958.
- [30] E. Ayrapetian, "Designing with GPS," *Pulse*, vol. 120, pp. 36-39, 2014.
- [31] D. McCallie, J. Butts, and R. Mills, "Security analysis of the ADS-B implementation in the next generation air transportation system," *Int. J. Crit. Infrastruct. Prot.*, vol. 4, no. 2, pp. 78-87, 2011.
- [32] A.D. King, "Inertial Navigation - Forty Years of Evolution," *GEC Rev.*, vol. 13, no. 3, pp. 140-149, 1998.
- [33] M. S. Grewal, L. R. Weill, and a. P. Andrews, *Global Positioning Systems, Inertial Navigation and Integration*, vol. 2. 2001.
- [34] K. Bousson, "4D Geodetic Waypoint Navigation Guidance," *Trajectory Optim. Control courses notes Aeronautical Engineering University of Beira Interior*, pp. 2-5, 2013.
- [35] V. Murugesan and K. Murugesan, "RK-Butcher algorithms for singular system-based electronic circuit," *Int. J. Comput. Math.*, vol. 86, no. 3, pp. 523-536, 2009.
- [36] H. Vermeille, "Direct transformation from geocentric coordinates to geodetic coordinates," *J. Geod.*, vol. 76, no. 8, pp. 451-454, 2002.

- [37] S. K. Balaji and a. Tatnall, "Modeling of periodic relative orbits using orbital element method," *2006 IEEE Aerosp. Conf.*, pp. 0-5, 2006.
- [38] W. Tseng, "The Shortest Overall Distance of Two Piecewise," pp. 1153-1157, 2014.
- [39] G. Cai, B. M. Chen, and T. H. Lee, *Unmanned Rotorcraft Systems*. London: Springer London, 2011.
- [40] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *Int. J. Forecast.*, vol. 22, no. 4, pp. 679-688, Oct. 2006.
- [41] R. Penrose and J. a. Todd, "A generalized inverse for matrices," *Math. Proc. Cambridge Philos. Soc.*, vol. 51, no. 03, p. 406, 1955.
- [42] B. Schölkopf and A. Smola, "A short introduction to learning with kernels," *Adv. Lect. Mach. Learn.*, pp. 41-64, 2003.

Anexos

Anexo I

Artigo Científico

Trajectory Prediction of Aerial Vehicles Through the Support Vector Machine Approach

Simão M. Lopes

(Artigo submetido para uma revista com referee e listada na ISI Journal Citation Reports)

TRAJECTORY PREDICTION OF AERIAL VEHICLES THROUGH THE SUPPORT VECTOR MACHINE APPROACH

Simão M. LOPES¹

¹ University of Beira Interior, Department of Aerospace Sciences, 6201-001 Covilhã, Portugal

The interest in linear regression methods has been growing exponentially due to their easy implementation. The present paper proposes a symbiotic relation between machine learning and its computational regression theorems, and aeronautical engineering with an implementation of Least-Squares Support Vector Machine methods in predicting the trajectory of an aerial vehicle, while working in a 4-dimensional waypoint navigation environment. This will allow to create a function that can fully define the information about the aircraft's coordinates in every location of its trajectory.

Key words: trajectory prediction; commercial aircraft; 4D waypoint navigation; machine learning; semi-supervised learning; support vector machines; kernel methods; radial basis functions.

1. INTRODUCTION

The whole subject of applying regression methods as the solution of estimation problems dates a few centuries ago. Due to the increasing interest in this matter, it soon became clear that it's not always easy to find a parametric model (linear or non-linear) that best explains some problems. All the attention was centered in the search for a more flexible method, where "the data could speak for itself". This constantly increasing development of regression methods is explained by the need of being able to extract detailed information from large amounts of data.

One discipline that takes in account the problems mentioned above, while simultaneously explores the technological aspect of the subject is artificial intelligence, or more specifically, machine learning. The latter can be defined as the area that studies and develops the computational methods that resort to experience in order to perform correct predictions, for the purpose of enhancing the accuracy and precision of the algorithm that is performing the task [1]. Machine learning can be divided into different categories, according to their different goals. The category that's centered in working with algorithms that are trained with classified examples (where the desired output is known) is called semi-supervised learning [2]. This subsector of machine learning, which is a compromise between supervised learning and non-supervised learning, meets the demands of wanting better and faster results at the same time.

The Least-Squares Support Vector Machine (LS-SVM) method falls within the semi-supervised category of machine learning. This method was created due to the fact that traditional Support Vector Machines (SVM) algorithms presented some complex computational limitations, especially when the data set being analyzed was too big [3]. In this paper, the LS-SVM method is implemented in the trajectory prediction of a commercial aircraft that flies between two airports. In order to create the trajectories used, the 4D waypoint navigation method was implemented. This method was chosen due to the increasing attention that has been receiving in the past years by the European air traffic control organization (Eurocontrol), and the new Single European Sky research program that will be implemented in the near future.

The purpose of this work is implementing an LS-SVM algorithm in a known trajectory, in order to find a function that can predict the coordinates of the aircraft in locations that were not known previously by the user.

2. ALGORITHM MODULATION AND IMPLEMENTATION

2.1 Optimality Conditions

In nonlinear programming problems, the general formula of the problem is given by the following expression [4]:

$$\min_{x \in \mathbb{R}^n} f(x) \quad (1)$$

subjected to: $c(x) \geq 0$, where $f(x)$ and $c(x)$ represent nonlinear functions in x .

Assuming that λ is a vector of m elements (Lagrange multipliers), then:

$$\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)^T \quad (2)$$

The Lagrangean function associated with problem (1) is:

$$L(x, \lambda) = f(x) - \lambda^T c(x) = f(x) - \sum_{i=1}^m \lambda_i c_i(x) \quad (3)$$

Notes:

Active Restrictions: Assume \bar{x} as a given point. The restrictions $c_i(\bar{x}) \geq 0, i \in \mathcal{A}$ are active in \bar{x} if $c_i(\bar{x}) = 0$.

Regularity Condition: Assume x^* as a solution for the problem. If the vectors $\nabla c_i(x^*), i \in \mathcal{A}$ are linearly independent, then x^* is a regular point.

2.1.1 Necessary and sufficient conditions – 1st Order

Assuming x^* as a solution for the problem. If x^* is a regular point, then there is a λ^* which verifies:

Stationary point of the Lagrangian Function	Admissibility	Positivity	Complementarity
$\nabla_x L(x^*, \lambda^*) = \nabla f(x^*) - \nabla c(x^*) \lambda^* = 0$	$c(x^*) \geq 0$	$\lambda^* \geq 0$	$(\lambda^*)^T c(x^*) = 0$

(4)

2.1.2 Karush-Kuhn-Tucker Conditions

Complementarity ($(\lambda^*)^T c(x^*) = 0$) affirms that the non-active restrictions have multipliers equal to zero ($c_i(x^*) > 0 \Rightarrow \lambda_i^* = 0, i \notin \mathcal{A}$).

For the active restrictions, the corresponding Lagrange multipliers may or may not be zero. If they are zero, then it's called a degenerated problem. Otherwise, it is called strict complementarity:

$$c(x^*) \geq 0 \quad (5)$$

which means that the point verifies the active restrictions, in other words, x^* is an admissible point.

$$\begin{aligned} \nabla_x L(x^*, \lambda^*) = \nabla f(x^*) - \nabla c(x^*) \lambda^* = 0 &\Leftrightarrow \nabla f(x^*) = \nabla c(x^*) \lambda^* \Leftrightarrow \nabla f(x^*) \\ &= \sum_{i=1}^m \lambda_i^* \nabla c_i(x^*) \end{aligned} \quad (6)$$

The expression above means that the gradient of $f(\nabla f)$ is a linear combination of the gradients of the restrictions.

2.2 Least-Squares Support Vector Machines

After generating all the trajectories, the LS-SVM algorithm was implemented. This method can be described by:

Given a data training set $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ with $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R} (i = 1, 2, \dots, m)$. Let $y = (y_1, y_2, \dots, y_m)^T$ [5]. Then, the optimization problem of the LS-SVM method can be defined as:

$$\min_{w \in \mathbb{R}^{n_h}, b \in \mathbb{R}, \xi \in \mathbb{R}^m} J(w, \xi) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \xi^T \xi \quad (7)$$

s.t. $y_i = w^T \varphi(x_i) + b + \xi_i, i = 1, 2, \dots, m$

where $\varphi: \mathbb{R}^d \rightarrow \mathbb{R}^{n_h}$ is a mapping to a higher dimensional feature space \mathcal{H} with n_h dimensions, each component of $\xi = (\xi_1, \xi_2, \dots, \xi_m)^T$ is a slack variable and γ is a positive and real parameter [6]. In cases where the data being analyzed contains too much noise, it is indicated to use a lower γ value [7]. Through the Karush-Kuhn-Tucker (KKT) conditions of the Lagrangian, the solution of the problem (7) is transcribed by the following linear system:

$$\begin{bmatrix} 0 & e^T \\ e & K + \gamma^{-1}I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \quad (8)$$

where $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)^T$ represents a vector containing the Lagrange multipliers (also called support vectors), $e = (1, 1, \dots, 1)^T$, I represents an identity matrix and $K_{i,j} = K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) (i, j = 1, 2, \dots, m)$ represents a kernel function meeting the Mercer's theorem [8].

Note: to resolve the linear system (8) it is necessary to calculate the pseudo-inverse matrix, due to the fact that the system doesn't have an unique solution [9].

Kernel functions can be represented by linear or polynomial functions, radial basis functions and many others. In this particular case, radial basis functions (RBFs) are applied. The list of kernel functions applied is the following:

Table 1

List of Kernel Functions Applied

Gaussian Kernel	Exponential Kernel	Laplacian Kernel
$h(x) = \exp\left(-\frac{\ x - x_n\ ^2}{2\sigma^2}\right)$	$h(x) = \exp\left(-\frac{\ x - x_n\ }{2\sigma^2}\right)$	$h(x) = \exp\left(-\frac{\ x - x_n\ }{\sigma}\right)$
Rational Quadratic Kernel	Multiquadric Kernel	Inverse Multiquadric Kernel
$h(x) = 1 - \frac{\ x - x_n\ ^2}{\ x - x_n\ ^2 + \eta}$	$h(x) = \sqrt{\ x - x_n\ ^2 + \eta}$	$h(x) = \frac{1}{\sqrt{\ x - x_n\ ^2 + \eta}}$

If the solutions of the linear system (8) are $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_m^*)^T$ and b^* , then the corresponding decision function (regression model) will be:

$$f(x) = w^{*T} \varphi(x) + b^* = \sum_{i=1}^m \alpha_i^* \varphi(x_i)^T \varphi(x) + b^* = \sum_{i=1}^m \alpha_i^* K(x_i, x) + b^* \quad (9)$$

To obtain an optimal solution in the regression model defined in (9), the first partial derivatives should be equal to zero, which means:

$$\begin{cases} \frac{\delta L}{\delta w} = 0 \Rightarrow w = \sum_{i=1}^m \alpha_i \varphi(x_i) \\ \frac{\delta L}{\delta b} = 0 \Rightarrow \sum_{i=1}^m \alpha_i \end{cases} \quad (10)$$

2.3 Gaussian Error

To better approximate the data generated to a real life scenario, a Gaussian error was applied to the trajectories generated by the algorithm, similar to what happens in GPS data. The Gaussian error was applied individually to each coordinate (X, Y, Z) , and the Box-Muller method was used to generate the Gaussian variables [10]. This method can be easily explained by analyzing the following process:

Given two independent and random variables U_1 and U_2 with a uniform distribution, in the interval $(0,1)$, then one can generate random Gaussian values by applying the following transform:

$$X = \sqrt{-2 \ln(U_2)} \sin(2\pi U_1) \quad Y = \sqrt{-2 \ln(U_2)} \cos(2\pi U_1) \quad (11)$$

where X and Y represent random variables normally distributed, with mean $\mu = 0$ and variance $\sigma^2 = 1$.

2.4 Spatial Localization

In a real life scenario, aircrafts depend on several instruments in order to determine its position and its flight parameters, such as: altitude, velocity, heading, angle of trajectory (flight path angle), etc. Here are some examples of how those instruments play a role in the aircraft's performance.

2.4.1 Automatic Dependent Surveillance-Broadcast

The ADS-B system was developed to overcome the conventional techniques of air traffic control, as well as giving new capabilities which will reinforce air traffic management. The following table shows a comparison between present radar systems and the attributes of the ADS-B system:

Table 2

Radar systems vs. ADS-B systems

Actual Radar Systems	ADS-B System
Land system, that depends on Human participation	Aircraft system, that gives a constant flow of more precise position data
Bad coverage in some areas	ADS-B land stations can be placed virtually anywhere
Position is updated every 12 seconds	Position is updated every second
Expensive installation and maintenance	Cheap installation and maintenance

This system uses a combination of navigation satellites (GNSS – Global Navigation Satellite System), transmitters and receivers to give specific information about the localization and speed of the aircraft that are flying over that area to the land control and crew.

2.4.2 Inertial Navigation System

The basic principle of inertial navigation is based on the simple concept of navigation estimation, or dead reckoning. The accelerometer gives the acceleration of the aircraft in a certain axis. If that acceleration is integrated, the velocity of the aircraft is obtained. With a second integration, the position of the aircraft, regarding that axis, is given. By knowing the orientation of the aircraft, it is possible to obtain the actual position [11].

2.4.3 Flight Path Angle

The flight path angle γ represents the amplitude between the horizontal axis and the trajectory axis of the aircraft, or the velocity vector. It is by measuring this angle that the pilot can determine if the aircraft is on a climb or descending phase.

2.4.4. Aircraft Velocity

Flight velocity is a measure of the aircraft's velocity relatively to the velocity of the air that surrounds the airplane. To calculate the relative speed of the aircraft, it is used a static pitot system. The value that is obtained directly from the pitot tube IAS – Indicated Air Speed, is not necessary equal to the aircraft's velocity: TAS – True Air Speed. This difference is explained by the compressibility of the air for velocities over 200 knots, the errors of the instruments, measure conditions (pitot tube, antenna, etc.) and relative density of the air.

3. SIMULATION AND VALIDATION IN TRAJECTORY PREDICTION

To better understand the structure of the program applied, the next image shows a hierarchy tree with all the processes that were implemented:

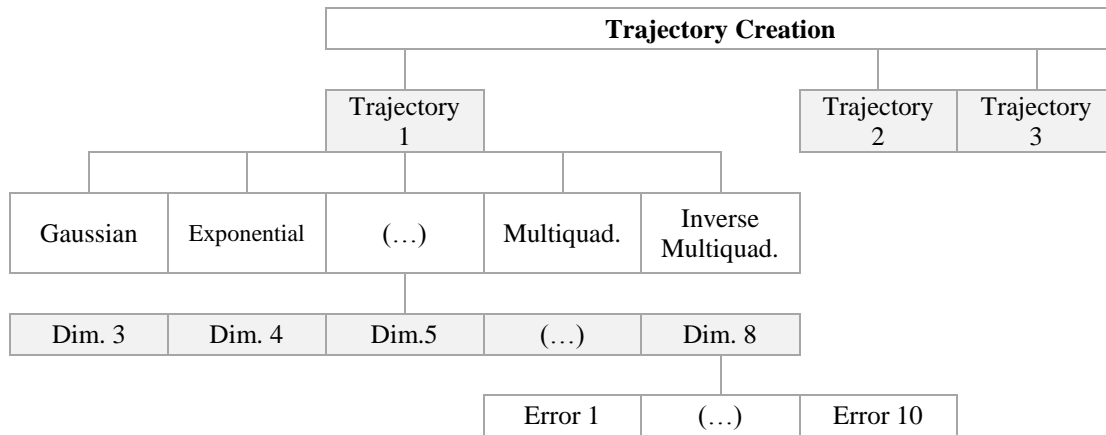


Fig. 1 – Structure of the Algorithm

From the analysis of Fig. 1, it is possible to get a general idea of how the program works, since the creation of the trajectory, to the implementation of algorithm.

3.1 Trajectory Creation

The four dimensional waypoint navigation method was used, in order to generate the trajectories to be implemented in the algorithm. This method is described in further detail in [12], and it consists in giving three parameters of the aircraft to a program: V_{ref} – reference velocity, ψ_{ref} – reference heading and γ_{ref} – reference flight path angle, so it can generate a trajectory that will fly through a predefined sequence of $N + 1$ waypoints $(P_0, P_1, \dots, P_k, \dots, P_N)$ given by the operator.

The three equations used in geodetic navigation are the following:

$$\dot{r} = V \sin \gamma \quad \dot{\lambda} = \frac{V \cos \gamma \sin \psi}{r \cos \varphi} \quad \dot{\varphi} = \frac{V \cos \gamma \sin \psi}{r} \quad (12)$$

where V is the flight velocity, γ is the flight path angle, ψ is the heading angle, r is the distance from the aircraft's center of gravity, to the center of the earth ($r = R_E + h$), where R_E is the earth's radius and h the altitude of the aircraft above sea level.

Each waypoint P_k is described as a four-dimensional vector, defined as:

$$P_k = (\lambda_k \ \varphi_k \ h_k \ \tau_k)^T \quad (13)$$

where λ_k represents the longitude of the waypoint, in radians, φ_k the latitude in radians as well, h_k the altitude above sea level, in meters and τ_k the scheduled time of arrival at the specific waypoint, in seconds.

From the equations stated above, one can conclude that it is necessary to update the status of the aircraft in each point, in order to create the trajectory. To do so, the RK-Butcher algorithm was implemented. This method can be analyzed in further detail in [13].

The model of the system that was used, can be described by:

$$\dot{x} = \text{equadif}(x_k, u_k) \quad (14)$$

where *equadif* represents a differential equation, and:

$$x \in \mathfrak{R}^n \text{ represents the state vector} \quad \Rightarrow x = \begin{pmatrix} r \\ \lambda \\ \varphi \end{pmatrix};$$

$$u \in \mathfrak{R}^m \text{ represents the control vector} \quad \Rightarrow u = \begin{pmatrix} V_{ref} \\ \psi_{ref} \\ \gamma_{ref} \end{pmatrix}$$

The constant h (or dt) represents the time step of the simulation, which means that for $x_k = x(t_k)$ and $u_k = u(t_k)$, where $k = 0, 1, 2, \dots$, the time difference between two consecutive points can be defined by $t_k = t_{k-1} + h$.

Assuming the initial conditions x_0 as given data, the updates of the system mentioned above, for $t_1, t_2, t_3, \dots, t_k$ are determined by:

$$x_{k+1} = x_k + \frac{1}{90} (7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6) \quad (15)$$

where:

$$\begin{aligned} k_1 &= h \cdot f(x_k, u_k); & k_4 &= h \cdot f\left(x_k + \frac{k_2}{2} + k_3, u_k\right); \\ k_2 &= h \cdot f\left(x_k + \frac{k_1}{4}, u_k\right); & k_5 &= h \cdot f\left(x_k + \frac{3k_1}{16} + \frac{9k_4}{16}, u_k\right); \\ k_3 &= h \cdot f\left(x_k + \frac{k_1}{8} + \frac{k_2}{8}, u_k\right); & k_6 &= h \cdot f\left(x_k + \frac{3k_1}{7} + \frac{2k_2}{7} + \frac{12k_3}{7} - \frac{12k_4}{7} \right. \\ & & & \left. + \frac{8k_5}{7}, u_k\right); \end{aligned} \quad (16)$$

Finally, the trajectory creator algorithm can be applied, by following the steps below:

1. Let X be the current position of the aircraft in the geodetic reference. First, it is necessary to establish a parameter $\sigma > 2b$, where b is the wing span of the aircraft;
2. Compute the distance $d_{\bar{P}_{k+1}}(X)$ from the current aircraft position $X = (\lambda, \varphi, h)$ to the next destination waypoint position \bar{P}_k , in geodetic reference;
3. If $d_{\bar{P}_{k+1}}(X) \leq \sigma$, then display the current time as the actual time of arrival at waypoint P_k , and switch navigation to waypoint P_{k+2} , and set $k = k + 1$;
4. Compute the reference velocity V_{ref} , heading ψ_{ref} and flight path angle γ_{ref} from the current aircraft position to the destination waypoint P_{k+1} ;
5. Send the reference velocity, heading and flight path angle to the RK-Butcher algorithm stated before.

Once again, the computations involved in the above steps, are explained in reference [12].

3.2 Waypoint Time of Arrival

To obtain the scheduled time of arrival of the aircraft at each waypoint, the following equation was applied:

$$t_{k+1} = t_k + \frac{D}{V_{med}} \times 1.2 \quad (17)$$

where t_k represents the time of arrival at the previous waypoint, D the loxodromic distance between the present and the next waypoint, V_{med} the mean velocity of the aircraft and 1.2 a safety parameter.

Given λ_A, φ_A and λ_B, φ_B as the longitude and latitude of waypoints A and B respectively, the loxodromic distance between them $(AB)_{NM}^{loxo}$, in nautical miles, is given by:

$$(AB)_{NM}^{loxo} = \frac{(\Delta\varphi_{AB})_{minutes}}{\cos V} \quad (18)$$

with $\Delta\varphi_{AB} = |\varphi_B - \varphi_A|$, and the angle V defined by:

$$V = \arctan_2((\Delta\lambda_{AB})_{minutes}, (\Delta\varphi_{AB}^*)_{minutes}) \quad (19)$$

where $\Delta\lambda_{AB} = |\lambda_B - \lambda_A|$ and $\Delta\varphi_{AB}^* = |\varphi_B^* - \varphi_A^*|$, in which:

$$\varphi^* = \frac{10800}{\pi} \ln \left[\tan \left(\frac{\varphi}{2} + 45^\circ \right) \right] \text{ (in arcminutes)} \quad (20)$$

where φ is in degrees [14].

3.3 Implementation

Firstly, it starts by creating the trajectories chosen by the user. In the present work, three different trajectories were created in order to test the algorithm. However, only the first trajectory was applied to all the different kernel functions, due to the heavy calculation process required to obtain all the solutions. This topic will be developed in more detail in the next chapters.

After the trajectories were created, the user chooses which kernel function will be applied in the LS-SVM method, between the six different functions available (Multiquadric, Inverse Multiquadric, Rational Quadratic, Gaussian, Exponential and Laplacian).

Then, the program needs to train the algorithm with solutions from the original data set (the coordinates X,Y,Z from the trajectories created), according to a reorganization of the data chosen by the user. This reorganization was tested for six different dimensions, which were (dimension 3, 4, 5, 6, 7 and 8). This process can be visually interpreted as:

Table 3

Coordinate Reorganization with Dimension = 3

Time [s]	X coordinate [m]				y_x	Y coordinate [m]				y_y	Z coordinate [m]				y_z
	x_1	x_2	x_3	x_4		y_1	y_2	y_3	y_4		z_1	z_2	z_3	z_4	
t_1	x_1	x_2	x_3	x_4	x_4	y_1	y_2	y_3	y_4	y_4	z_1	z_2	z_3	z_4	z_4
t_2	x_2	x_3	x_4	x_5	x_5	y_2	y_3	y_4	y_5	y_5	z_2	z_3	z_4	z_5	z_5
t_3	x_3	x_4	x_5	x_6	x_6	y_3	y_4	y_5	y_6	y_6	z_3	z_4	z_5	z_6	z_6
...
t_N	x_{N-3}	x_{N-2}	x_{N-1}	x_N	x_N	y_{N-3}	y_{N-2}	y_{N-1}	y_N	y_N	z_{N-3}	z_{N-2}	z_{N-1}	z_N	z_N

The same process occurred for the other five dimensions, with the only difference in the size of the vectors created that will match the size of the dimension defined by the user. From Table 2, it is possible to see that the vectors y_x, y_y and y_z correspond to the solutions of the initial data set that will initially train the algorithm.

Just like mentioned in Chapter 2.3, all the data points of the trajectory created in the beginning, suffered an addition of a Gaussian error. In order to test different error values, different amplitudes were used when the Gaussian vector was created. The different amplitude values were: $Amp_{gauss.error} = [5, 10, 20, 40, 80, 160, 320, 640, 1280, 2560]$. Some examples of the error values added to the coordinates can be observed in Fig. 2.

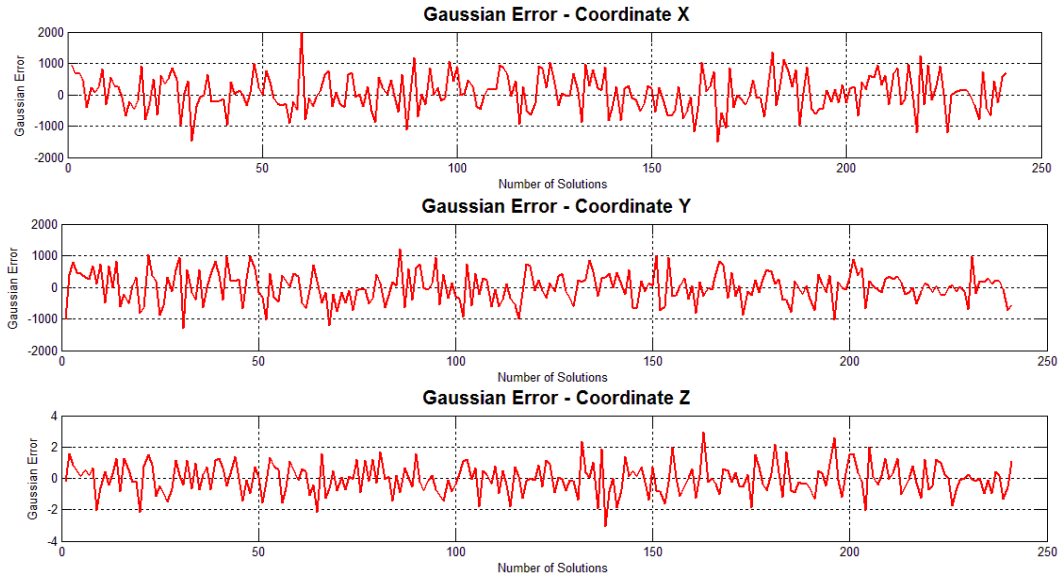


Fig. 2 – Some examples of the Gaussian error added to the coordinates

3.4 Trajectories Created

To the implementation of the LS-SVM method, three different commercial aircraft trajectories were chosen in order to predict the position of the vehicle in different points of its flight. The first trajectory was defined between two European airports:

1. Departure: London Heathrow Airport;
2. Arrival: Paris Charles de Gaulle Airport.

In order to obtain coordinates from different points of the trajectory, the waypoint navigation system was selected. The altitude and time values were obtained accordingly to the service ceiling of the aircraft and through the loxodromic distances between each point, and can be analyzed in the following table:

Table 4

Airport and Waypoints coordinates with Time of Arrival of Trajectory 1

	Long [rad]	Lat [rad]	Alt [m]	Time [s]
Heathrow	-0,0081	0,8985	25	0
Waypoint 1	0,0082	0,8888	12527	715,4
Waypoint 2	0,0177	0,8842	12527	7445,7
Waypoint 3	0,0280	0,8762	12527	15261,9
Waypoint 4	0,0346	0,8697	12527	15661,7
Waypoint 5	0,0402	0,8620	12527	16093,1
Charles de Gaulle	0,0445	0,8554	119	16458,9

The specification values of the selected aircraft can be consulted in Table 5:

Table 5

Aircraft Specifications

Minimum Velocity [m/s]	64	Minimum Trajectory Angle Rate [rad/s]:	-1
Maximum Velocity [m/s]	253,61	Maximum Trajectory Angle Rate [rad/s]:	1
Minimum Trajectory Angle [rad]:	-0,489	Minimum Heading Rate [rad/s]:	-1
Maximum Trajectory Angle [rad]:	0,611	Maximum Heading Rate [rad/s]:	1

This trajectory was used in order to compare the results between different reorganization dimensions and different kernel functions, so it was possible to obtain the lowest error value for the predicted trajectory. This happened because trajectory 1, as it will be established, contains fewer points than the other two trajectories, giving it a much faster computing time when compared to the others.

The second trajectory created had the following airports as origin and destination:

1. Departure: Incheon International Airport (Seoul);
2. Arrival: Paris Charles de Gaulle Airport.

The coordinates of both airports, as well as the complete list of the waypoints used between them can be seen in Table 6:

Table 6

Airport and Waypoints coordinates with Time of Arrival of Trajectory 2

	Long [rad]	Lat [rad]	Alt [m]	Time [s]
Incheon	2,1946	0,6528	5113	0
Waypoint 1	2,1476	0,6620	9167	1953
Waypoint 2	2,0806	0,6688	9174	4659
Waypoint 3	2,0340	0,7198	9174	7828
Waypoint 4	1,9310	0,8047	9174	13582
Waypoint 5	1,7176	0,8783	9784	21730
Waypoint 6	1,4190	0,9992	9921	32602

	Long [rad]	Lat [rad]	Alt [m]	Time [s]
Waypoint 7	1,1085	1,0664	10356	41390
Waypoint 8	0,8117	1,0761	10363	48647
Waypoint 9	0,5984	1,0478	10965	54133
Waypoint 10	0,2967	0,9840	10973	62847
Waypoint 11	0,0657	0,8638	5029	72199
Charles de Gaulle	0,0544	0,8559	1593	72752

Lastly, trajectory 3 demonstrates a navigation between the following international airports:

1. Departure: Vancouver International Airport;
2. Arrival: London Heathrow Airport.

Table 7

Airport and Waypoints coordinates with Time of Arrival of Trajectory 3

	Long [rad]	Lat [rad]	Alt [m]	Time [s]
Vancouver	-2,1500	0,8587	0	0
Waypoint 1	-2,1432	0,8612	3124	262
Waypoint 2	-2,0813	0,9070	10058	3333
Waypoint 3	-2,0388	0,9344	10058	5245
Waypoint 4	-1,8451	1,0340	10058	12691

	Long [rad]	Lat [rad]	Alt [m]	Time [s]
Waypoint 7	-0,9460	1,1434	10668	34052
Waypoint 8	-0,5036	1,0790	10668	44557
Waypoint 9	-0,3248	1,0412	10668	49405
Waypoint 10	-0,1820	0,9976	10668	53813
Waypoint 11	-0,0519	0,9299	10668	58927

Waypoint 5	- 1,6206	1,1054	10058	19273	Waypoint 12	-0,0394	0,9213	9114	59509
Waypoint 6	- 1,3149	1,1478	10058	26298	Heathrow	-0,0074	0,8982	0	62504

3.5 Results

3.5.1 Root Mean Square Error

In regression problems, the mean error (or root mean square error) of a result \hat{y}_t compared to the dependent variable y is estimated to all n results, depending on the number of predictions made by the algorithm. This results in the following equation:

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n}} \quad (21)$$

The output of the mean errors obtained allows to measure the precision of the algorithm [15].

The results were obtained for many different values of γ . The equation $\gamma_{k+1} = \gamma_k \times 1.2$ was used for the variation of this parameter, which lead to the following results:

$$\gamma_V = [10.00, 12.00, 14.40, 17.28, 20.73, 24.788, 29.85, 35.83, 42.99, 51.59, 61.91, 74.30, 89.16] \quad (22)$$

On Chapter 2.2., more specifically, in the list of the RBF functions used, the parameter σ can be found on three different functions (Gaussian, Exponential and Laplacian). On the final part of this topic, the results shown for this three functions will have different values of σ for each other, due to the fact that some tests were made in order to find the ideal σ parameter for each one of them.

The parameter η found in the Rational Quadratic, Multiquadric and Inverse Multiquadric functions, is obtained separately for each coordinate, given by the following expression:

$$\eta_x = \max_k \{(x_{k+1} - x_k)\} \quad \eta_y = \max_k \{(y_{k+1} - y_k)\} \quad \eta_z = \max_k \{(z_{k+1} - z_k)\} \quad (23)$$

Finally, the solutions $\begin{bmatrix} b \\ \alpha \end{bmatrix}$ of the linear system (8) are applied in equation (9) in order to create a new trajectory, made by the results obtained.

Since many values of γ were tested, the results contain a lot of predicted trajectories. In order to simplify the visualization of the results, the values presented next correspond to the ones that presented the lowest mean error value (RMSE). In order to calculate the global error of the solutions, the following expression was used:

$$\text{RMSE}_{global} = \sqrt{\text{RMSE}_x^2 + \text{RMSE}_y^2 + \text{RMSE}_z^2} \quad (24)$$

The values for the global errors obtained for the different functions, reorganization Dimensions and amplitudes of the Gaussian error were compared, and the best and worst results determined were:

- Min. Error = 0,481770558303577 for a Multiquadric Function – Dimension 7 – with $\text{Amp}_{gauss.error} = 2560$ and $\gamma = 51,5978$;
- Max. Error = 91599,50188 for an Inverse Multiquadric Function – Dimension 3 – with $\text{Amp}_{gauss.error} = 1280$ and $\gamma = 10$.

Looking at the value for the minimum error obtained for the Multiquadric function, it is possible to assume that the prediction made was almost perfect, as seen in the following figure:

Trajectory 1 - Original vs. Trajectory 1 - LS-SVM Algorithm (Dimension 7)

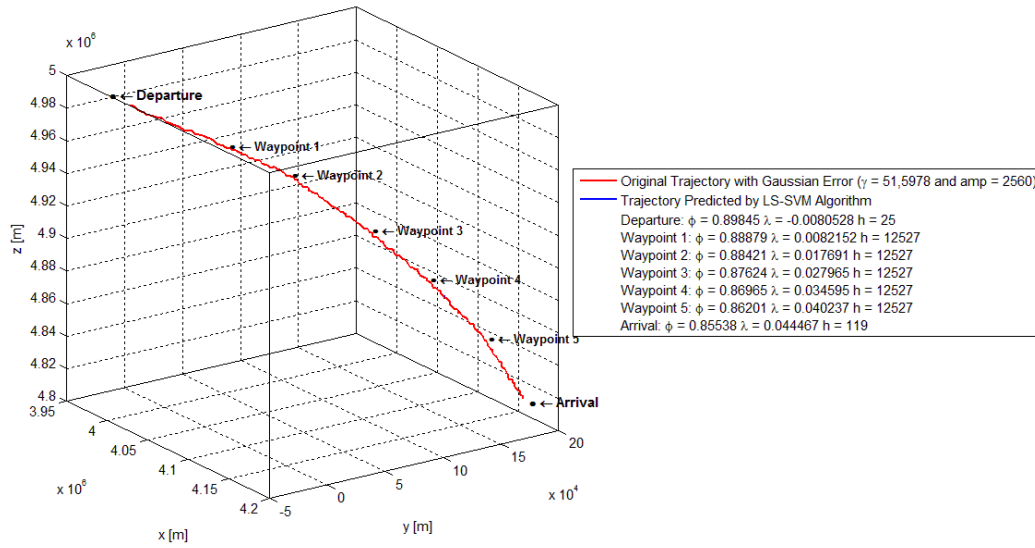


Fig. 3 – Trajectory 1: Original vs. Predicted

After determining the ideal RBF function and reorganization dimension, the algorithm was applied to the other two trajectories mentioned in the previous chapter. The data for the trajectory 2 resulted in the following prediction:

Trajectory 2 - Original vs. Trajectory 2 - LS-SVM Algorithm (Dimension 7)

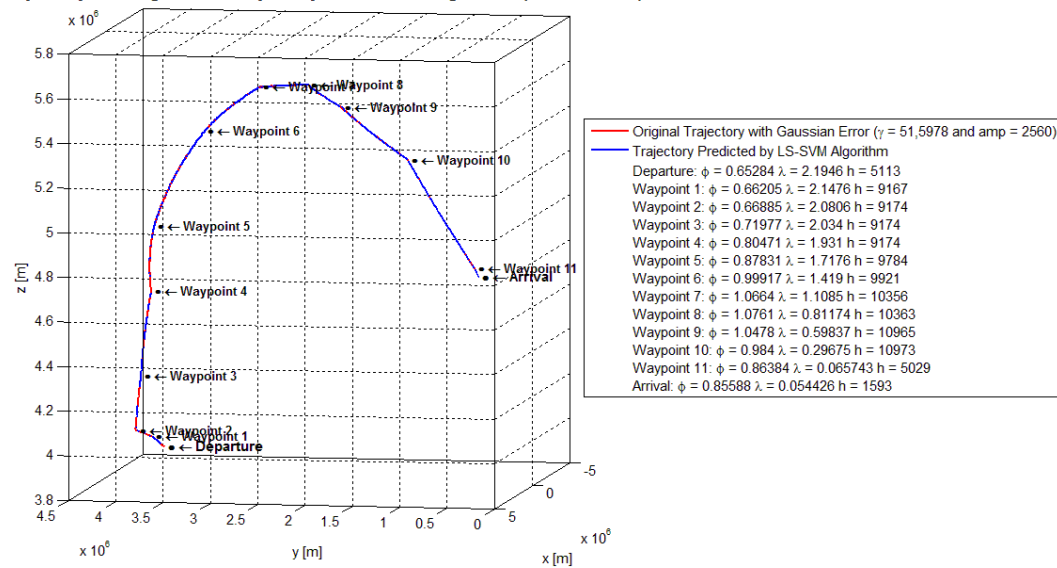


Fig. 4 – Trajectory 2: Original vs. Predicted

As before, after applying the Multiquadric function with a reorganization with dimension 7, the trajectory 3 was generated. The results can be seen in the following figure:

Trajectory 3 - Original vs. Trajectory 3 - LS-SVM Algorithm (Dimension 7)

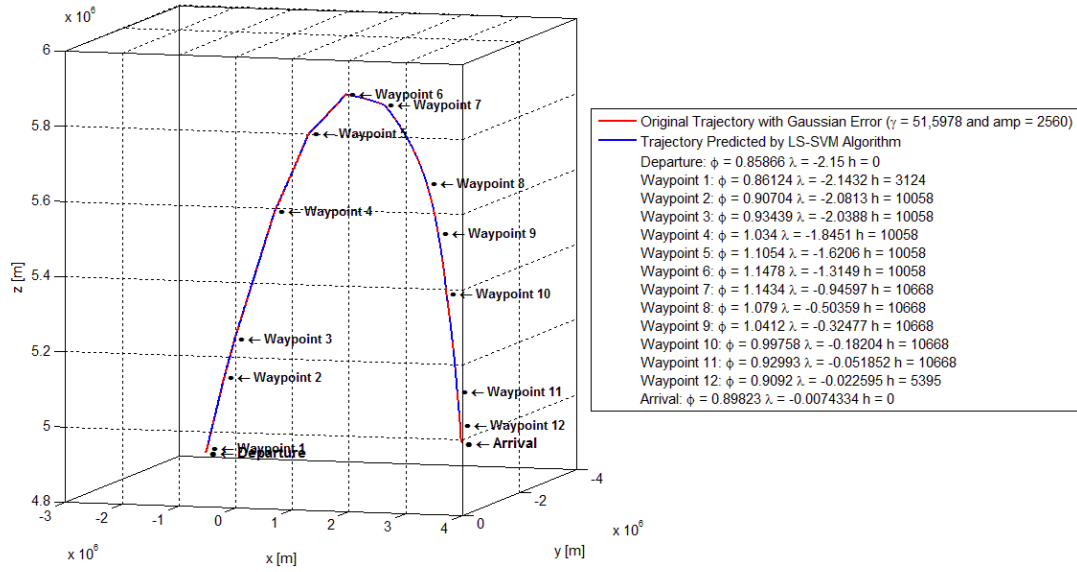


Fig. 5 – Trajectory 3: Original vs. Predicted

From the analysis of the previous figures, it is possible to conclude that the predictions made by the LS-SVM algorithm are almost coincident to the original trajectories created. However, the true advantage of this method reveals itself when a function that can classify the entire trajectory is created, like mentioned in equation (9).

The next table shows the results obtained for the prediction of the original trajectories, for each coordinate separately. It is clear that the solutions predicted by the algorithm are very close to the original solutions that correspond to the entry data.

Table 8

Example results for the X coordinate (Trajectory 1 – Dimension 6)

Aircraft Position – Trajectory 1 – Dimension 6 – X coordinate [m]						Real Solution	Predicted Solution
3972348	3973903	3975524	3977141	3978744	3980379	3981993,72	3981988,49
3973903	3975524	3977141	3978744	3980379	3981993	3983617,66	3983612,43
3975524	3977141	3978744	3980379	3981993	3983617	3985228,56	3985223,33

Table 9

Example results for the Y coordinate (Trajectory 1 – Dimension 3)

Aircraft Position – Trajectory 1 – Dimension 3 – Y coordinate [m]			Real Solution	Predicted Solution
-31987,96127	-30181,57368	-28321,21301	-26457,92455	-26459,54543
-30181,57368	-28321,21301	-26457,92455	-24595,78106	-24597,36861
-28321,21301	-26457,92455	-24595,78106	-22729,64254	-22731,19572

Table 10

Example results for the Z coordinate (Trajectory 2 – Dimension 7)

Aircraft Position – Trajectory 2 – Dimension 7 – Z coordinate [m]							Real Solution	Predicted Solution
387746	387933	388130	388324	388521	388719	388915	3891118,8	3891501,4
387933	388130	388324	388521	388719	388915	389111	3893070,4	3892265,5
388130	388324	388521	388719	388915	389111	389307	3895028,9	3895187,0

From the observation of the colored patterns on the previous tables, one can conclude that the real solution created by the program will take place as an element of the next entry vector. This pattern repeats itself for each coordinate, for all the trajectories under every dimension of reorganization.

On the last column are the solutions predicted by the algorithm. These will be used to train the algorithm and will allow the user to create a function that will be able to classify the trajectory in points that weren't classified before.

After obtaining the results, the user can implement the function transcribed in equation (9): $f(x) = \sum_{i=1}^m \alpha_i^* \varphi(x_i)^T \varphi(x) + b^*$, where α_i^* and b^* represent the solutions of the equation (8) and $\varphi(x_i)^T$ the kernel matrix created by the resolution of the RBF function chosen by the user. The term $\varphi(x)$ will be the result of the application of that same RBF function, but to the coordinates that the user wishes to determine. Here are some examples of the infinite solutions that the algorithm can generate:

Note: The solutions predicted are transcribed by the following equations:

- X coordinate – $f(4100000) = \sum_{i=1}^m \alpha_{x_i}^* \varphi(x_i)^T \varphi(4100000) + b_x^*$;
- Y coordinate – $f(70000) = \sum_{i=1}^m \alpha_{y_i}^* \varphi(y_i)^T \varphi(70000) + b_y^*$;
- Z coordinate – $f(4900000) = \sum_{i=1}^m \alpha_{z_i}^* \varphi(z_i)^T \varphi(4900000) + b_z^*$.

Table 11

Example 1: Predicted Solutions for Multiquadric Function

Aircraft Position Multiquadric Function (Coordinates) [m]			Predicted Solution (Coordinates) [m]		
Original - X	Original - Y	Original - Z	Predicted - X	Predicted - Y	Predicted - Z
4100000	70000	4900000	4104401,423	74394,484	4896945,862

Table 12

Example 2: Predicted Solutions for Exponential Function

Aircraft Position Exponential Function (Coordinates) [m]			Predicted Solution (Coordinates) [m]		
Original - X	Original - Y	Original - Z	Predicted - X	Predicted - Y	Predicted - Z
4100000	70000	4900000	4104390,99	72471,67	4896591,06

Table 13

Example 3: Predicted Solutions for Gaussian Function

Aircraft Position Exponential Function (Coordinates) [m]			Predicted Solution (Coordinates) [m]		
Original - X	Original - Y	Original - Z	Predicted - X	Predicted - Y	Predicted - Z
4100000	70000	4900000	4100454,07	72434,42	4897436,08

Table 14

Example 4: Predicted Solutions for Laplacian Function

Aircraft Position Exponential Function (Coordinates) [m]			Predicted Solution (Coordinates) [m]		
Original - X	Original - Y	Original - Z	Predicted - X	Predicted - Y	Predicted - Z

4100000	70000	4900000	4101534,98	72761,32	4897200,51
---------	-------	---------	------------	----------	------------

The coordinates were selected randomly, as long as they were inside the boundaries of the training set: $x_{departure} < x_{predicted} < x_{arrival}$, $y_{departure} < y_{predicted} < y_{arrival}$ and $z_{departure} < z_{predicted} < z_{arrival}$.

Once again, the solutions predicted by the algorithm are in accordance with expected values, since they follow a logical progression when compared to the values that precede them.

4. CONCLUSIONS

It is possible to say that the application of LS-SVM methods in trajectory prediction was successful. As shown in Tables 11, 12, 13, and 14, the algorithm is capable of creating a function that will allow the user to obtain data from any point of the trajectory, as long as it stays inside the boundaries of the original data set.

Furthermore, the mean errors (RMSE) for each coordinate, as well as the global error registered were very low for every $\gamma \in [10, 100]$ tested. This boundary was created in order to save time in the computational process.

However, this regression method can be limiting, since the data in which it can predict solutions must be inside the boundaries defined by the training set. Outside these values, the LS-SVM method becomes inaccurate and can lead to faulty results.

Another offset of this method is the whole computational process. When applying this method, it can become very heavy for the computer to calculate everything while maintaining a fast and efficient time of execution.

REFERENCES

- HUA, J., "Study on the Application of Rough Sets Theory in Machine Learning," *2008 Second Int. Symp. Intell. Inf. Technol. Appl.*, pp. 192–196, Dec. 2008.
- RUSSEL, S., NORVIG, P., *Artificial Intelligence: A Modern Approach*, Third Edit. Pearson Education, 1994.
- WANG, C., CHEN, S., "An improved LS-SVM based on SSOR-PCG," *2013 Ninth Int. Conf. Nat. Comput.*, pp. 28–33, Jul. 2013.
- LUENBERGER, D.G., YE, Y., *Linear and Nonlinear Programming*, Third Edit. Springer, 2008.
- XING, H., JIN, T., "Detection of Weak Signal in Chaotic Clutter Using Advanced LS-SVM Regression," *2009 2nd Int. Congr. Image Signal Process.*, pp. 1–5, Oct. 2009.
- XIE, J., "Kernel optimization of LS-SVM based on damage detection for smart structures," *2009 2nd IEEE Int. Conf. Comput. Sci. Inf. Technol.*, pp. 406–409, 2009.
- SCHÖLKOPF, B., SMOLA, A., "A short introduction to learning with kernels," *Adv. Lect. Mach. Learn.*, pp. 41–64, 2003.
- SCHÖLKOPF, B., SMOLA, A., *Learning with kernels: support vector machines, regularization, optimization, and beyond*. The MIT Press, Cambridge, 2002.
- PENROSE, R., TODD, J.A., "A generalized inverse for matrices," *Math. Proc. Cambridge Philos. Soc.*, vol. 51, no. 03, p. 406, 1955.
- BOX, G.E.P., MULLER, M.E., "A Note on the Generation of Random Normal Deviates," *Ann. Math. Stat.*, vol. 29, no. 2, pp. 610–611, 1958.
- KING, A.D., "Inertial Navigation – Forty Years of Evolution," *GEC Rev.*, vol. 13, no. 3, pp. 140–149, 1998.
- BOUSSON, K., "4D Geodetic Waypoint Navigation Guidance," *Trajectory Optimization Control course notes Aeronautical Engineering University of Beira Interior*, pp. 2–5, 2013.
- MURUGESH, V., MURUGESAN, K., "RK–Butcher algorithms for singular system-based electronic circuit," *Int. J. Comput. Math.*, vol. 86, no. 3, pp. 523–536, 2009.
- TSENG, W., "The Shortest Overall Distance of Two Piecewise," pp. 1153–1157, 2014.
- HYNDMAN, R.J., KOEHLER, A.B., "Another look at measures of forecast accuracy," *Int. J. Forecast.*, vol. 22, no. 4, pp. 679–688, Oct. 2006.

Anexo II

Produto Interno

Em matemática, o produto interno (ou produto escalar) representa uma operação algébrica que através de uma sequência de números com dimensões iguais (normalmente vetores) tem como solução um único valor.

Seja \mathcal{V} um espaço vetorial real. Toma o nome de produto interno em \mathcal{V} uma aplicação:

$$\begin{aligned}\langle \cdot, \cdot \rangle : \mathcal{V} \times \mathcal{V} &\rightarrow \mathbb{R} \\ (u, v) &\rightarrow \langle u, v \rangle\end{aligned}$$

que verifique as seguintes propriedades:

- Simetria hermitiana: $\forall u, v \in \mathcal{V}, \langle u, v \rangle = \langle v, u \rangle$;
- Distributividade: $\forall u, v, w \in \mathcal{V}, \langle u + v, w \rangle = \langle u, w \rangle + \langle v, w \rangle$;
- Associatividade: $\forall u, v \in \mathcal{V}, \forall \alpha \in \mathbb{R}, \langle \alpha u, v \rangle = \alpha \langle u, v \rangle$;
- Positividade: $\forall u \in \mathcal{V}, \langle u, u \rangle \geq 0$ e $\langle u, u \rangle = 0 \Leftrightarrow u = 0_{\mathcal{V}}$.

Se \mathcal{V} for um espaço vetorial real, de dimensão finita e que possua um produto interno, então \mathcal{V} diz-se um espaço Euclidiano.

Exemplos: Para $x = (x_1, \dots, x_n), y = (y_1, \dots, y_n) \in \mathbb{R}^n$, $\langle x, y \rangle = \sum_{k=1}^n x_k y_k$, define um produto interno em \mathbb{R}^n .

Seja $C[a, b]$ o conjunto das aplicações $f : [a, b] \rightarrow \mathbb{R}$ que são contínuas. $C[a, b]$ representa um espaço vetorial real para a adição usual de funções e multiplicação de uma função por um escalar. Para $f, g \in C[a, b]$,

$$\langle f, g \rangle = \int_a^b f(x)g(x)dx$$

define um produto interno em $C[a, b]$. Neste caso $C[a, b]$ também é um espaço Euclidiano porque, para quaisquer funções f e g , $f(x)g(x)$ é integrável.

Proposição: Seja \mathcal{V} um Euclidiano real.

- $\forall u \in \mathcal{V}, \langle 0_{\mathcal{V}}, u \rangle = 0$;
- Se $\langle u, v \rangle = 0$ para todo $v \in \mathcal{V}$, então $u = 0_{\mathcal{V}}$;
- Se $\langle u, v \rangle = \langle u', v \rangle$ para todo $v \in \mathcal{V}$, então $u = u'$.

Definições: Seja \mathcal{V} um Euclidiano real e sejam $u, v \in \mathcal{V}$

- A norma de u é o escalar não negativo $\|u\| = \sqrt{\langle u, u \rangle}$, Se $\|u\| = 1$ então diz-se que u é unitário;
- A distância de u a v é representada por $\|u - v\|$;
- Os vetores u e v dizem-se ortogonais se $\langle u, v \rangle = 0$;
- A projecção ortogonal de u sobre $v \neq 0_{\mathcal{V}}$ é o vetor:

$$\text{proj}_v u = \frac{\langle u, v \rangle}{\|v\|^2} v$$

Anexo III

Matriz Pseudo-inversa de Moore-Penrose

A matriz pseudo-inversa de Moore-Penrose é única pode ser definida para qualquer matriz. Além disso, tal como irá ser provado de seguida, esta matriz traz uma clareza tanto a nível de notação como conceptual no estudo das soluções de sistemas lineares de equações e de problemas lineares de mínimos quadrados.

Considerando uma matriz $A \in \mathbb{R}_r^{m \times n}$, então todas as matrizes $A \in \mathbb{R}_r^{m \times n}$ têm uma pseudo-inversa, com a notação $A^+ \in \mathbb{R}_r^{m \times n}$.

Teorema: Se $A \in \mathbb{R}_r^{m \times n}$, então $G = A^+$ se e só se:

- $AGA = A$;
- $GAG = G$;
- $(AG)^T = AG$;
- $(GA)^T = GA$.

O teorema acima permite confirmar se uma matriz G é realmente a pseudo-inversa de A se as quatro condições mencionadas forem cumpridas simultaneamente [41].

Aplicações à solução de sistemas lineares

Sejam $A \in \mathbb{R}_r^{m \times n}$, $B \in \mathbb{R}^m$ e supondo que $AA^+b = b$. Então, qualquer vetor da forma:

$$x = A^+b + (I - A^+A)y$$

onde $y \in \mathbb{R}^n$ é arbitrário, é uma solução do sistema:

$$Ax = b$$

Para verificar que é de facto uma solução do sistema, multiplica-se por A :

$$\begin{aligned} Ax &= AA^+b + A(I - A^+A)y \\ &= b + (A - AA^+A)y \\ &= b \end{aligned}$$

Quando A representa uma matriz quadrada e não singular, $A^+ = A^{-1}$ e então $(I - A^+A) = 0$. Logo, não existe componente arbitrário, deixando apenas uma solução única $x = A^{-1}b$.

Anexo IV

Desenvolvimento do Método Box-Muller

Para que seja possível explicitar o método de Box-Muller, é necessário definir algumas formas introduzidas no desenvolvimento da dissertação, que ilustram a relação entre coordenadas cartesianas e coordenadas polares:

$$R^2 = X^2 + Y^2 \text{ e } \tan \theta = \left(\frac{Y}{X}\right)$$

Para além disso, é também obrigatório esclarecer algumas relações entre expressões trigonométricas:

$$\sin \theta = \frac{\tan \theta}{\sqrt{1+\tan^2 \theta}} \text{ e } \cos \theta = \frac{1}{\sqrt{1+\tan^2 \theta}}$$

As relações entre as coordenadas cartesianas e as coordenadas polares podem ser justificadas da seguinte forma:

1) Provar que: $X = R \cos \theta$

$$R^2 = X^2 + Y^2 \Leftrightarrow X^2 = R^2 - Y^2 \text{ e } Y = X \cdot \tan \theta$$

logo:

$$X^2 = R^2 - (X \cdot \tan \theta)^2 \Leftrightarrow X^2 = R^2 - X^2 \cdot \tan^2 \theta \Leftrightarrow X^2 + X^2 \cdot \tan^2 \theta = R^2 \Leftrightarrow 1 + \tan^2 \theta = \frac{R^2}{X^2} \Leftrightarrow$$

$$\Leftrightarrow X^2 = \frac{R^2}{1 + \tan^2 \theta} \Leftrightarrow X = \sqrt{\frac{R^2}{1 + \tan^2 \theta}} \Leftrightarrow X = \frac{1}{\sqrt{1 + \tan^2 \theta}} \times R \Leftrightarrow X = R \cos \theta$$

2) Provar que $Y = R \sin \theta$

$$R^2 = X^2 + Y^2 \Leftrightarrow Y^2 = R^2 - X^2 \text{ e } X = \frac{Y}{\tan \theta}$$

logo:

$$Y^2 = R^2 - \left(\frac{Y}{\tan \theta}\right)^2 \Leftrightarrow Y^2 = R^2 - \frac{Y^2}{\tan^2 \theta} \Leftrightarrow Y^2 + \frac{Y^2}{\tan^2 \theta} = R^2 \Leftrightarrow 1 + \frac{1}{\tan^2 \theta} = \frac{R^2}{Y^2} \Leftrightarrow$$
$$\Leftrightarrow \frac{\tan^2 \theta + 1}{\tan^2 \theta} = \frac{R^2}{Y^2} \Leftrightarrow Y^2 = \frac{\tan^2 \theta \cdot R^2}{\tan^2 \theta + 1} \Leftrightarrow Y = \sqrt{\frac{\tan^2 \theta \cdot R^2}{\tan^2 \theta + 1}} \Leftrightarrow Y = \frac{\tan \theta}{\sqrt{1 + \tan^2 \theta}} \times R \Leftrightarrow Y = R \sin \theta$$

Por fim, o método de Box-Muller é justificado pelos seguintes princípios:

3) Provar que os números aleatórios seguem uma distribuição normal:

A fórmula para uma curva de distribuição normal (*probability density function*) é dada por:

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

onde μ é a média e σ^2 a variância da distribuição. Numa distribuição normal com média $\mu = 0$ e variância $\sigma^2 = 1$, a expressão acima pode ser simplificada, tomando a seguinte forma:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

Para obter uma variável aleatória com distribuição normal, seria necessário aplicar uma transformada inversa à expressão anterior. Este processo seria bastante complicado, devido ao termo exponencial presente na fórmula. Assim, a transformada de Box-Muller pretende facilitar este processo, efetuando o processo para duas variáveis aleatórias.

Seja (X, Y) um par de variáveis aleatórias independentes que apresentam uma distribuição normal. Então, $f(X, Y)$ pode ser definida como:

$$f(X, Y) = \int_{-\infty}^Y \int_{-\infty}^X \left\{ \frac{1}{\sqrt{2\pi}} e^{-\frac{x'^2}{2}} \right\} \left\{ \frac{1}{\sqrt{2\pi}} e^{-\frac{y'^2}{2}} \right\} dx' dy'$$

É possível concluir que as expressões: $\left\{ \frac{1}{\sqrt{2\pi}} e^{-\frac{x'^2}{2}} \right\}$ e $\left\{ \frac{1}{\sqrt{2\pi}} e^{-\frac{y'^2}{2}} \right\}$ representam as fórmulas para uma distribuição normal para as variáveis x' e y' , respectivamente. Continuando a simplificação da expressão:

$$f(X, Y) = \int_{-\infty}^Y \int_{-\infty}^X \left\{ \frac{1}{2\pi} e^{-\frac{x'^2+y'^2}{2}} \right\} dx' dy'$$

Devido às relações entre X e Y definidas no início deste capítulo (R e θ), é possível definir $f(X, Y)$ através de R e θ ($drd\theta$):

$$\begin{aligned} f(X, Y) = f(R, \theta) &= \int_{-\infty}^{\theta} \int_{-\infty}^R \left\{ \frac{1}{2\pi} e^{-\frac{R'^2}{2}} \right\} dR' d\theta' = \int_0^{\theta} \int_0^R \left\{ \frac{1}{2\pi} e^{-\frac{R'^2}{2}} \right\} dR' d\theta' \\ &= \frac{1}{2\pi} \int_0^{\theta} \left[-e^{-\frac{R'^2}{2}} \right]_0^R d\theta' = \frac{\theta}{2\pi} \left\{ 1 - e^{-\frac{R^2}{2}} \right\} = f(R)f(\theta) \end{aligned}$$

Se U_1 e U_2 forem duas variáveis com distribuição uniforme, e $U_1, U_2 \in [0,1]$, então:

$$f(\theta) \leq U_1, f(R) \leq U_2$$

Como $U_2 \in [0,1]$, então $1 - U_2 \in [0,1]$, logo:

$$f(\theta) \leq U_1, f(R) \leq 1 - U_2$$

Substituindo na expressão anterior:

$$\begin{aligned} U_1 &= \frac{\theta}{2\pi} \Leftrightarrow \theta = 2\pi U_1 \\ 1 - U_2 &= 1 - e^{-\frac{R^2}{2}} \Leftrightarrow U_2 = e^{-\frac{R^2}{2}} \Leftrightarrow R = \sqrt{-2 \ln(U_2)} \end{aligned}$$

Por fim, é possível resolver em ordem a X e Y da seguinte forma:

$$X = \sqrt{-2 \ln(U_2)} \sin(2\pi U_1)$$

$$Y = \sqrt{-2 \ln(U_2)} \cos(2\pi U_1)$$

Estas duas expressões permitem gerar números aleatórios e independentes, com uma distribuição normal a partir de dois números aleatórios U_1 e U_2 que se encontrem num intervalo $[0,1]$, e apresentem uma distribuição uniforme.

Anexo V

Cálculo da Velocidade de Referência V_{ref}

Seja $\delta_k = d_{p_{k+1}}(X)$ a distância entre a posição atual da aeronave e o waypoint de destino, como calculada anteriormente, $RV = [V_{min}, V_{max}]$ o alcance de velocidades da aeronave e a_{max} a aceleração máxima da mesma, onde V_{min} e V_{max} são as velocidades mínimas e máximas da aeronave, respectivamente.

A velocidade de referência é obtida, obedecendo aos seguintes cálculos:

1. Se $(\tau_k > t)$

$$V^* = \frac{\delta_k}{(\tau_k - t)};$$

onde t representa o tempo atual e τ_k o tempo de chegada ao waypoint.

se a condição inicial não for cumprida,

$$V^* = V_{max};$$

esta última condição verifica-se no caso de o tempo atual ser superior ao tempo de chegada ao waypoint seguinte, o que se traduz por um atraso no cumprimento do horário por parte da aeronave.

2. $T_V = 1/5a_{max};$

3.
$$V^* = \begin{cases} V_{min} & \text{se } V^* < V_{min} \\ V^* & \text{se } V_{min} < V^* < V_{max}; \\ V_{max} & \text{se } V^* > V_{max} \end{cases}$$

4. $V_{ref} = V + (1 - e^{-dt/T_V})(V^* - V).$

Cálculo do Rumo de Referência ψ_{ref}

Partindo do pressuposto que as longitudes (λ_k e λ) e as latitudes (φ_k e φ) estão expressas em radianos, os passos para calcular o rumo de referência são os seguintes:

1. $\Delta\lambda = \lambda_k - \lambda;$ onde λ_k e λ representam as longitudes do waypoint de destino e da posição atual da aeronave, respectivamente;
2. $\varphi^{(1)} = \ln(|\tan(\varphi/2 + \pi/4)|);$ onde φ representa a latitude da posição atual da aeronave;
3. $\varphi^{(2)} = \ln(|\tan(\varphi_k/2 + \pi/4)|);$ em que φ_k representa a latitude do waypoint de destino;
4. $\Delta\varphi = \varphi^{(2)} - \varphi^{(1)};$
5. $\varphi_{LOS} = \arctan_2(\Delta\lambda, \Delta\varphi);$ em radianos;
6. $T_H = 1/(5 \cdot d\psi_{max});$
7. $\psi_{ref} = \psi + (1 - e^{-dt/T_H})(\varphi_{LOS} - \psi);$

Cálculo do Ângulo de Trajetória de Referência γ_{ref}

Seja $RG = [\gamma_{min}, \gamma_{max}]$, o intervalo operacional dos ângulos de trajetória para uma aeronave considerada. O cálculo do ângulo de trajetória de referência desenvolve-se do seguinte modo:

1. $\xi_{LOS} = \arcsin\left(\frac{h_k - h}{d_{P_{k+1}}(X)}\right)$; onde h_k e h são as altitudes do waypoint de destino e da posição atual, respetivamente, e $d_{P_{k+1}}(X)$ a distância entre os dois;
2. $T_\gamma = 1/(5 \cdot d\gamma_{max})$;
3. $\xi_{LOS} = \begin{cases} \gamma_{min} & \text{se } \xi_{LOS} < \gamma_{min} \\ \xi_{LOS} & \text{se } \gamma_{min} < \xi_{LOS} < \gamma_{max}; \\ \gamma_{max} & \text{se } \xi_{LOS} > \gamma_{max} \end{cases}$
4. $\gamma_{ref} = \gamma + (1 - e^{-dt/T_\gamma})(\xi_{LOS} - \gamma)$;

Anexo VI

Tempo de Execução - Trajetória 1

Multiquádrica	T1 A1 K1 D3 - Elapsed time is 1474.793983 seconds.
	T1 A1 K1 D4 - Elapsed time is 1595.308727 seconds.
	T1 A1 K1 D5 - Elapsed time is 1630.371867 seconds.
	T1 A1 K1 D6 - Elapsed time is 1496.548735 seconds.
	T1 A1 K1 D7 - Elapsed time is 1537.836890 seconds.
	T1 A1 K1 D8 - Elapsed time is 1508.829541 seconds.

Multiquádrica Inversa	T1 A1 K2 D3 - Elapsed time is 1484.149693 seconds.
	T1 A1 K2 D4 - Elapsed time is 1467.422492 seconds.
	T1 A1 K2 D5 - Elapsed time is 1753.027798 seconds.
	T1 A1 K2 D6 - Elapsed time is 1534.583342 seconds.
	T1 A1 K2 D7 - Elapsed time is 1451.476922 seconds.
	T1 A1 K2 D8 - Elapsed time is 1856.265574 seconds.

Rational Quadratic	T1 A1 K3 D3 - Elapsed time is 1489.798993 seconds.
	T1 A1 K3 D4 - Elapsed time is 1452.451298 seconds.
	T1 A1 K3 D5 - Elapsed time is 1471.010998 seconds.
	T1 A1 K3 D6 - Elapsed time is 1589.836711 seconds.
	T1 A1 K3 D7 - Elapsed time is 1606.397307 seconds.
	T1 A1 K3 D8 - Elapsed time is 1511.490252 seconds.

Gaussiana	T1 A1 K4 S1250 D3 - Elapsed time is 3609.946873 seconds.
	T1 A1 K4 S1250 D4 - Elapsed time is 3579.805732 seconds.
	T1 A1 K4 S1250 D5 - Elapsed time is 3618.242713 seconds.
	T1 A1 K4 S1250 D6 - Elapsed time is 3602.083499 seconds.
	T1 A1 K4 S1250 D7 - Elapsed time is 3706.126090 seconds.
T1 A1 K4 S1250 D8 - Elapsed time is 3487.771334 seconds.	

Exponencial	T1 A1 K5 S70 D3 - Elapsed time is 1480.661161 seconds.
	T1 A1 K5 S70 D4 - Elapsed time is 1465.590966 seconds.
	T1 A1 K5 S70 D5 - Elapsed time is 1470.791735 seconds.
	T1 A1 K5 S70 D6 - Elapsed time is 1572.208968 seconds.
	T1 A1 K5 S70 D7 - Elapsed time is 1438.545592 seconds.
	T1 A1 K5 S70 D8 - Elapsed time is 1460.550668 seconds.

Laplaciana	T1 A1 K6 S4100 D3 - Elapsed time is 1521.415101 seconds.
	T1 A1 K6 S4100 D4 - Elapsed time is 1743.258887 seconds.
	T1 A1 K6 S4100 D5 - Elapsed time is 1474.044848 seconds.
	T1 A1 K6 S4100 D6 - Elapsed time is 1792.440865 seconds.
	T1 A1 K6 S4100 D7 - Elapsed time is 1711.590429 seconds.
	T1 A1 K6 S4100 D8 - Elapsed time is 1670.291879 seconds.

Diferentes Sigmas	
Gaussiana	T1 A1 K4 D6 S0,1 - Elapsed time is 3415.833597 seconds.
	T1 A1 K4 D6 S7,5 - Elapsed time is 3456.908235 seconds.
	T1 A1 K4 D6 S250 - Elapsed time is 3501.289605 seconds.
	T1 A1 K4 D6 S1000 - Elapsed time is 3435.419775 seconds.
	T1 A1 K4 D6 S1250 - Elapsed time is 3602.083499 seconds.
	T1 A1 K4 D6 S1500 - Elapsed time is 3520.195571 seconds.
	T1 A1 K4 D6 S1700 - Elapsed time is 3626.045015 seconds.
	T1 A1 K4 D6 S1900 - Elapsed time is 3496.880363 seconds.
	T1 A1 K4 D6 S2350 - Elapsed time is 3499.019992 seconds.
	T1 A1 K4 D6 S3500 - Elapsed time is 3510.515663 seconds.
	T1 A1 K4 D6 S4100 - Elapsed time is 3502.887249 seconds.
T1 A1 K4 D6 S5000 - Elapsed time is 3432.413027 seconds.	
T1 A1 K4 D6 S6000 - Elapsed time is 3579.271451 seconds.	

Exponencial	T1 A1 K5 D6 S30 - Elapsed time is 1578.349832 seconds.
	T1 A1 K5 D6 S70 - Elapsed time is 1572.208968 seconds.
	T1 A1 K5 D6 S100 - Elapsed time is 1465.509330 seconds.
	T1 A1 K5 D6 S250 - Elapsed time is 2501.416193 seconds.
	T1 A1 K5 D6 S400 - Elapsed time is 1488.905152 seconds.
	T1 A1 K5 D6 S700 - Elapsed time is 1458.681519 seconds.
	T1 A1 K5 D6 S1400 - Elapsed time is 1494.924785 seconds.

Laplaciana	T1 A1 K6 D6 S2 - Elapsed time is 1450.456214 seconds.
	T1 A1 K6 D6 S20 - Elapsed time is 1446.056457 seconds.
	T1 A1 K6 D6 S1000 - Elapsed time is 1549.425138 seconds.
	T1 A1 K6 D6 S1500 - Elapsed time is 1568.642937 seconds.
	T1 A1 K6 D6 S2000 - Elapsed time is 1698.670764 seconds.
	T1 A1 K6 D6 S3200 - Elapsed time is 1463.882102 seconds.
	T1 A1 K6 D6 S3700 - Elapsed time is 1476.658033 seconds.
	T1 A1 K6 D6 S4100 - Elapsed time is 1792.440865 seconds.
T1 A1 K6 D6 S5000 - Elapsed time is 1613.412831 seconds.	

Legenda:

- T1 - Trajetória 1
- A1 - Aeronave 1
- KX - Função Kernel X
- 1 - Função Multiquádrica
- 2 - Função Multiquádrica Inversa
- 3 - Função Rational Quadratic
- 4 - Função Gaussiana
- 5 - Função Exponencial
- 6 - Função Laplaciana
- SX - Sigma
- DX - Dimensão da Reestruturação

