



Ministério da
**Ciência, Tecnologia
e Inovação**



sid.inpe.br/mtc-m19/2011/11.07.22.08-TDI

SISTEMA DE NAVEGAÇÃO INERCIAL GPS/INS DE BAIXO CUSTO COM COMPENSAÇÃO DE ERROS POR REDES NEURAIS ARTIFICIAIS

Edmundo Alberto Marques Filho

Tese de Doutorado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle, orientada pelos Drs. Atair Rios Neto, e Hélio Koiti Kuga, aprovada em 06 de dezembro de 2011.

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/3AP6SB5>>

INPE
São José dos Campos
2011

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@sid.inpe.br

CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE (RE/DIR-204):**Presidente:**

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Membros:

Dr^a Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Dr^a Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr^a Regina Célia dos Santos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Dr. Ralf Gielow - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr. Wilson Yamaguti - Coordenação Engenharia e Tecnologia Espacial (ETE)

Dr. Horácio Hideki Yanasse - Centro de Tecnologias Especiais (CTE)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Deicy Farabello - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Vivéca Sant´Ana Lemos - Serviço de Informação e Documentação (SID)



Ministério da
**Ciência, Tecnologia
e Inovação**



sid.inpe.br/mtc-m19/2011/11.07.22.08-TDI

SISTEMA DE NAVEGAÇÃO INERCIAL GPS/INS DE BAIXO CUSTO COM COMPENSAÇÃO DE ERROS POR REDES NEURAIAS ARTIFICIAIS

Edmundo Alberto Marques Filho

Tese de Doutorado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle, orientada pelos Drs. Atair Rios Neto, e Hélio Koiti Kuga, aprovada em 06 de dezembro de 2011.

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/3AP6SB5>>

INPE
São José dos Campos
2011

Dados Internacionais de Catalogação na Publicação (CIP)

Marques Filho, Edmundo Alberto.

M348s Sistema de navegação inercial GPS/INS de baixo custo com compensação de erros por redes neurais artificiais / Edmundo Alberto Marques Filho. – São José dos Campos : INPE, 2011.
xxiv + 123 p. ; (sid.inpe.br/mtc-m19/2011/11.07.22.08-TDI)

Tese (Doutorado em Mecânica Espacial e Controle) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2011.

Orientadores : Drs. Atair Rios Neto, e Hélio Koiti Kuga.

1. Navegação de baixo custo. 2. Redes neurais artificiais (RNA). 3. Navegação inercial. 4. GPS. 5. UMI. 6. SNI. 7. Filtro de Kalman. I.Título.

CDU 629.7.05:004.89

Copyright © 2011 do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecânico, fotográfico, reprográfico, de microfilmagem ou outros, sem a permissão escrita do INPE, com exceção de qualquer material fornecido especificamente com o propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright © 2011 by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, microfilming, or otherwise, without written permission from INPE, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de Doutor(a) em
Engenharia e Tecnologia Espaciais/Mecânica
Espacial e Controle

Dr. Paulo Giacomio Milani

Presidente / INPE / SJCampos - SP

Dr. Hélio Koiti Kuga

Orientador(a) / INPE / SJCampos - SP

Dr. Atair Rios Neto

Orientador(a) / FIBRAFORTE / SJCampos - SP

Dr. Valdemir Carrara

Membro da Banca / INPE / SJCampos - SP

Dr. Elder Moreira Hemerly

Convidado(a) / ITA/DCTA / São José dos Campos - SP

Dr. Jaime Augusto da Silva

Convidado(a) / EMBRAER / São José dos Campos - SP

Este trabalho foi aprovado por:

() maioria simples

unanimidade

Aluno (a): Edmundo Alberto Marques Filho

São José dos Campos, 06 de dezembro de 2011

AGRADECIMENTOS

Meus agradecimentos ao Instituto Nacional de Pesquisas Espaciais, INPE, por disponibilizar a infraestrutura e os recursos humanos necessários para a condução deste trabalho, especialmente ao corpo docente e funcionários da Divisão de Mecânica Espacial e Controle, envolvidos com o curso de Pós-graduação em Engenharia e Tecnologia Espaciais. Aos colegas de curso, pela convivência, troca de experiências, incentivo e ajuda mútua, desde os tempos do curso de mestrado.

Agradeço aos membros da Banca pelo interesse demonstrado, por disponibilizarem seu tempo na leitura dos textos, pela participação nas etapas de apresentação, pelas críticas e sugestões. Ao Dr. Jaime Augusto da Silva e Dr. Elder Moreira Hemerly por aceitarem prontamente nosso convite, e ao Dr. Valdemir Carrara e Dr. Paulo Giácomo Milani que estiveram em todas as minhas apresentações, desde o mestrado.

Meus agradecimentos especiais aos orientadores Dr. Helio Koiti Kuga e Dr. Atair Rios Neto, que sempre me apoiaram nas mais diferentes situações, cada um de uma forma diferente e complementar. Dívida *ad-aeternum*.

Este trabalho é dedicado as minhas meninas Mônica e Ana Luiza.

RESUMO

Este trabalho aborda a aplicação de redes neurais artificiais (RNA) em um sistema de navegação inercial de baixo custo auxiliado por GPS. Busca-se suprir a necessidade que ocorre já que o GPS não trabalha em todos os ambientes, ou pode trabalhar de forma inadequada, durante certo intervalo de tempo. Desenvolve-se solução para prover a falta de informações na porção do tempo onde o GPS não opera, explorando a integração de sensores inerciais com o GPS. Sensores inerciais de baixo custo têm, porém, a desvantagem de apresentar um grande e contínuo acúmulo de erros, acarretando baixo desempenho no sistema. Utiliza-se rede neural, neste contexto, para melhorar a solução de navegação dada pelo sistema inercial nos períodos de ausência das informações provenientes do GPS. É apresentada uma revisão dos principais conceitos e técnicas envolvidas com o tema em questão; uma abordagem para configurar a RNA, quanto aos sinais de entrada e saída, baseada num conjunto simplificado de equações de navegação inercial e seus dois modos de operação, treinamento e predição. É abordada uma metodologia para o processamento de dados de treinamento e respectivo algoritmo de treinamento utilizando um filtro de Kalman adaptativo. Também é proposto um método para o treinamento da RNA com a característica de alternar a apresentação dos padrões de treinamento entre o modo lote, com conjuntos de tamanho fixo, e o modo sequencial, filtrando os padrões de treinamento individualmente, característica que confere ao método certa capacidade de treinamento em tempo aproximadamente-real. Finalmente, são apresentados os resultados obtidos, por simulação numérica, de uma aplicação de posicionamento veicular terrestre com dados adquiridos de uma IMU Crossbow CD400-200 e de um receptor GPS Ashtech Z12. Os métodos propostos foram testados em situações distintas de movimento do veículo e avaliados os erros de posição, ou predição, obtidos durante ausência simulada do GPS. Os erros de predição, obtidos pela RNA, apresentaram menor magnitude quando comparados com um sistema INS/GPS convencional, integrado por um filtro de Kalman, também simulado sem a atualização do GPS. Os resultados indicaram que a RNA foi mais eficiente em representar a cinemática do veículo, num dado intervalo de tempo, que o sistema de navegação convencional.

LOW COST GPS/INS INERTIAL NAVIGATION SYSTEMS WITH ERROR COMPENSATION BY ARTIFICIAL NEURAL NETWORKS

ABSTRACT

This work addresses the use of artificial neural network (ANN) in GPS aided inertial navigation systems. The GPS technology dominates, nowadays, the positioning and navigation (POS/NAV) market, and alternative POS/NAV systems are only needed because GPS does not work in all environments, or can not provide reliable solutions during some time interval. There are different solutions to fulfill information during GPS blockage and integrated inertial sensors systems with GPS are frequently used. However, low cost inertial sensors have the disadvantage of accumulating continuous errors in great extension, leading to poor system performance. In this context, ANN is applied to provide better NAV/POS solutions during the lack of information in GPS outages. This work introduces a review of the main used concepts and techniques, an approach to define input-output ANN signals based on a reduced set of inertial navigation equations, and the ANN prediction and training operation modes. It presents a methodology to preprocessing input data and a training algorithm, based on adaptive Kalman filtering approach. It also proposes a method for ANN training with the characteristic of alternating the training patterns from batch mode, with a constant data set size, to sequential mode, by filtering individual pattern-by-pattern of training data, which gives to the method some real time training capacity. Finally, numerical simulation results are assessed from urban vehicular positioning application, with data acquired from an MEMS IMU Crossbow CD400_200 and an Ashtech Z12 GPS receiver. The proposed methods were tested with different vehicle dynamic situations and the position errors, acquired in prediction mode or simulated GPS outage, were assessed. When compared to a conventional INS/GPS system, integrated by a Kalman filter and operating without GPS updates, the ANN position errors have lower magnitudes. These results indicate that ANN was more capable to learn the vehicle's kinematics, for a certain time interval, than the modeling presented by the conventional navigation system.

LISTA DE FIGURAS

	<u>Pág.</u>
2.1 - Sistemas de coordenadas.....	12
2.2 - Sistema referencial do veículo.....	14
2.3 – Fluxograma simplificado de um sistema de navegação inercial	19
2.4 – Fluxograma do INS no referencial de navegação.....	23
2.5 – Sistema fracamente acoplado.....	29
2.6 – Sistema fortemente acoplado.....	29
3.1 - Modelo de um neurônio.....	32
3.2 - Redes alimentadas adiante (<i>feedforward</i>)	34
3.3 – Rede MLP com uma camada oculta.....	35
5.1 – Conceção básica da RNA: entradas e saídas.....	53
5.2 – Configuração da RNA para simulação.....	55
5.3 – Modo treinamento	56
5.4 – Modo predição.....	57
5.5 – Método de treinamento 01.....	59
5.6 – Método de treinamento 02.....	59
5.7 – Método de treinamento 03.....	60
5.8 – Método de treinamento 04.....	61
6.1 - Deslocamento fornecido pelo GPS.....	65
6.2 - Velocidades estimadas via GPS.....	66
6.3 – Forças específicas fornecidas pela IMU.....	66
6.4 - Velocidades angulares fornecidas pela IMU.....	67
6.5 - Sinais da IMU pré-filtrados.....	68
6.6 - Velocidade v^f pré-filtrada.....	68
6.7 - Comparação dos erros de posição e de velocidade (FKE-A, LM).....	69
6.8 – Erro de treinamento da RNA (FKE-A, LM).....	70
6.9 - Comparação dos erros de posição e de velocidade (FKE-A, LM).....	71
6.10 - Erro de treinamento da RNA (FKE-A, LM).....	72

6.11 - Comparação dos erros de posição e de velocidade (FKE, FKE-A).....	72
6.12 - Erro de treinamento da RNA (FKE, FKE-A).....	73
6.13 - Comparação dos erros de posição e de velocidade (FKE, FKE-A).....	74
6.14 - Erro de treinamento da RNA (FKE, FKE-A).....	75
6.15 – Comparação entre erros de treinamento – LM.....	75
6.16 - Comparação entre erros de treinamento – FKE-A.....	76
6.17 - Comparação entre erros de treinamento – FKE.....	76
6.18 – Índice R para dados de validação (Leste).....	78
6.19 - Índice R para dados de validação (Norte).....	78
6.20 – Índice R e erros de treinamento vs. neurônios ocultos.....	79
6.21 – Índice R e erros de treinamento vs. iterações.....	79
6.22 – Posição Leste e Norte – GPS.....	81
6.23 – Velocidade v^f – GPS.....	82
6.24 – Força específica e velocidade angular – IMU.....	82
6.25 – Erro de predição $t_{pred} = 155s$. a) método 1, b) método 2.....	83
6.26 - Erro de predição $t_{pred} = 155s$. a) método 3, b) método 4.....	83
6.27 – Erros de predição $jan = 20s$, $t_{pred} = 155s$. a) Leste, b) Norte.....	84
6.28 - Erros de predição $jan = 20s$, $t_{pred} = 155s$, plano L-N.....	84
6.29 – Diagrama dos treinamentos para $t_{pred} = 155s$	85
6.30 - Posição Leste e Norte – GPS.....	86
6.31 - Velocidade v^f – GPS.....	86
6.32 – Força específica e velocidade angular – IMU.....	87
6.33 - Erro de predição $t_{pred} = 705s$. a) método 1, b) método 2.....	87
6.34 - Erro de predição $t_{pred} = 705s$. a) método 3, b) método 4.....	88
6.35 - Erros de predição $jan = 60s$, $t_{pred} = 705s$. a) Leste, b) Norte.....	88
6.36 - Erros de predição $jan = 60s$, $t_{pred} = 705s$, plano L-N.....	89
6.37 - Erro de predição $t_{pred} = 755s$. a) método 1, b) método 2.....	89
6.38 - Erro de predição $t_{pred} = 755s$. a) método 3, b) método 4.....	90
6.39 - Erros de predição $jan = 60s$, $t_{pred} = 755s$. a) Leste, b) Norte.....	90
6.40 - Erros de predição $jan = 60s$, $t_{pred} = 705s$, plano L-N.....	91
6.41 – Deslocamento Leste-Norte fornecido pelo GPS.....	91

6.42 – Velocidade v^f utilizada na entrada da RNA.....	92
6.43 – Sinais da IMU utilizados na entrada da RNA.....	92
6.44 – Índice R e erros de treinamento <i>vs.</i> neurônios ocultos.	93
6.45 – Índice R e erros de treinamento <i>vs.</i> iterações.	93
6.46 - Erro de predição $t_{pred} = 65s$. a) método 1, b) método 2.	94
6.47 - Erro de predição $t_{pred} = 65s$. a) método 3, b) método 4.	94
6.48 - Erros de predição $jan = 20s$, $t_{pred} = 65s$. a) Leste, b) Norte.	94
6.49 - Erros de predição $jan = 20s$, $t_{pred} = 65s$, plano L-N.....	95
6.50 – Erro de predição $t_{pred} = 85s$. a) método 1, b) método 2.	96
6.51 – Erros de predição $t_{pred} = 85s$. a) método 3. b) método 4.	96
6.52 - Erro de predição $t_{pred} = 115s$. a) método 1, b) método 2.	96
6.53 - Erros de predição $t_{pred} = 115s$. a) método 3. b) método 4.....	97
6.54 - Erros de predição $jan = 40s$, $t_{pred} = 115s$. a) Leste, b) Norte.	97
6.55 - Erros de predição $jan = 40s$, $t_{pred} = 115s$, plano L-N.....	98
6.56 - Erro de predição $t_{pred} = 140s$. a) método 1, b) método 2.	98
6.57 - Erros de predição $t_{pred} = 140s$. a) método 3. b) método 4.....	99
6.58 - Erros de predição $jan = 60s$, $t_{pred} = 140s$. a) Leste, b) Norte.	99
6.59 - Erros de predição $jan = 60s$, $t_{pred} = 140s$, plano L-N.....	100
6.60 – Erros de treinamento para o subconjunto	101
6.61 – Posição Leste simulada e GPS [50s, 600s].....	102
6.62 - Posição Norte simulada e GPS [50s, 600s].....	102
6.63 – Velocidade v^f simulada e GPS [50s, 600s].....	103
6.64 – Erros de posição e velocidade [50s, 600s].....	103
6.65 – Índice R <i>vs.</i> iterações.	104
6.66 – Índice R <i>vs.</i> neurônios ocultos.....	105
A.1 – Receptor GPS Ashtec Z12 – características técnicas	119

LISTA DE TABELAS

	<u>Pág.</u>
6.1 – Erros de posição e velocidade (mse) – dados pré-filtrados	69
6.2 – Erros de posição e velocidade (mse) – dados originais	71
6.3 - Erros de posição e velocidade (mse) – dados pré-filtrados	73
6.4 - Erros de posição e velocidade (mse) – dados originais.....	74

LISTA DE SIGLAS E ABREVIATURAS

ANN	<i>Artificial Neural Network</i>
CG	Centro de Gravidade
DCM	<i>Direction Cosine Matrix</i>
DGPS	<i>Differential Global Positioning System</i>
DR	<i>Dead-Reckoning</i>
ECEF	<i>Earth-Centered, Earth-Fixed</i>
ECI	<i>Earth-Centered-Inertial</i>
ENU	<i>East-North-Up</i>
FKE	Filtro Estendido de Kalman
FKE-A	Filtro Estendido de Kalman e Adaptivo
FKU	Filtro de Kalman <i>Unscented</i>
FNL	<i>Functional Link Network</i>
GEKF	<i>Global Extended Kalman Filter</i>
GPS	<i>Global Positioning System</i>
IA	Inteligência Artificial
IMU	<i>Inertial Measurement Unit</i>
INS	<i>Inertial Navigation System</i>
LM	<i>Levenberg-Marquardt</i>
LTP	<i>Local Tangent Plane</i>
MEMS	<i>Micro Electrical Mechanical Systems</i>
MLP	<i>Multi Layer Perceptron</i>
MST	<i>Micro System Technology</i>
NAV	Navegação
NAVSTAR	<i>NAVgation Satellite with Time and Ranging</i>
NED	<i>North-East-Down</i>
POS	Posicionamento
PUA	<i>Position Update Architecture</i>
PVUA	<i>Position and Velocity Update Architecture</i>

RNA	Redes Neurais Artificiais
RNC	Redes Neurais Construtivas
SINS	<i>Strapdown Inertial Navigation System</i>
VUA	<i>Velocity Update Architecture</i>
ZUPT	<i>Zero Velocity Update</i>

LISTA DE SÍMBOLOS

Latinos

b_k	<i>Bias</i> aplicado ao neurônio k
$\{b_u, b_v, b_w\}$	Correções, ou <i>bias</i> , estimadas para correção de $\tilde{\mathbf{f}}^b$
$\{b_p, b_q, b_r\}$	Correções, ou derivas, estimadas para correção de $\tilde{\omega}_{ib}^b$
d_k	Saída desejada do neurônio k
a	Semi-eixo maior
e	Excentricidade
\mathbf{e}	Vetor de erros
\mathbf{e}_{POS}	Vetor de erros de posição
\mathbf{e}_{VEL}	Vetor de erros de velocidade
e_k	Erro relativo à saída do neurônio k
\mathbf{f}^b	Vetor de forças específicas computadas via IMU, expresso em (b)
$\tilde{\mathbf{f}}$	Vetor de forças específicas medidos pela IMU
$g(0)$	Valor da gravidade em função da latitude
\mathbf{g}^n	Vetor gravidade, expresso em (n)
h	Altitude
\mathbf{H}_k	Matriz que relaciona estados com observações
$\vec{\mathbf{i}}, \vec{\mathbf{j}}, \vec{\mathbf{k}}$	Vetores unitários, no referencial (b)
\mathbf{I}_n	Matriz identidade, de ordem n

$J(\mathbf{w})$	Função de custo em função dos pesos sinápticos
\mathbf{K}	Vetor dos ganhos de Kalman
n	Instante de tempo discreto
$p = 1 \dots L$	Número de pares de treinamento
$\{p, q, r\}$	Taxas angulares medidas por giroscópios
\mathbf{p}^n	Vetor de coordenadas de posição (lat, lon, alt), no referencial (n)
\mathbf{p}_u	Última informação de posição disponível do GPS
\mathbf{P}_k	Matriz de covariâncias dos erros das estimativas a priori
\mathbf{Q}	Matriz de densidade espectral
\mathbf{Q}_k	Matriz de covariâncias dos ruídos (dinâmico) de processo
\mathbf{R}_k	Matriz de covariâncias dos ruídos (erros) de medida
R_M	Raio de curvatura meridional
R_N	Raio de curvatura transversal
R^0	Raio médio de curvatura da Terra
$\mathbf{R}_Z, \mathbf{R}_Y, \mathbf{R}_X$	Matrizes de rotação
\mathbf{R}_b^n	Matriz de rotação de (b) para (n)
t	Tempo, instante
t_u	Instante da última informação, antes da perda de sinal do GPS
v^f	Velocidade na direção do movimento do veículo (<i>forward</i>)
v_{GPS}^f	Velocidade v^f computada com informações do GPS

v_k	Campo local induzido ou potencial de ativação do neurônio k
$\{V_N, V_E, V_D\}$	Velocidades lineares ao longo dos eixos N, E, D
\mathbf{v}^n	Vetor de velocidades, expresso nas coordenadas NED
\bar{v}	Média de velocidades
w_{kj}	Peso sináptico que liga o sinal de entrada j ao neurônio k
\mathbf{w}	Vetor de parâmetros livres da RNA (pesos e bias)
$\hat{\mathbf{w}}$	Vetor de parâmetros livres estimado durante treinamento da RNA
x_j	Sinal de entrada j, na RNA
\mathbf{x}	Vetor de sinais de entrada, ou vetor de estado
$\{X_i, Y_i, Z_i\}$	Eixos do referencial inercial (i)
$\{X_e, Y_e, Z_e\}$	Eixos do referencial Terrestre (e)
$\{X_n, Y_n, Z_n\}$	Eixos do referencial de navegação (n)
$\{X_b, Y_b, Z_b\}$	Eixos do referencial do veículo, ou do corpo (b)
y_k	Sinal de saída do neurônio k, ou de observações
\mathbf{y}	Vetor dos sinais de saída da RNA
$\{u, v, w\}$	Eixos do referencial do veículo. Notação alternativa

Gregos

α_j	Parâmetro empírico, associado ao estado j
$\delta_{i,l}$	Função delta de Kronecker
δt	Intervalo de tempo

$\Delta \mathbf{p}_{\text{GPS}}$	Vetor de incrementos de posição do GPS
$\Delta \mathbf{p}_{\text{RNA}}$	Vetor de incrementos de posição fornecido pela RNA
$\Delta \mathbf{v}_{\text{GPS}}^f$	Vetor de incrementos de velocidade
Δt	Intervalo de tempo
λ	Longitude geodésica
σ_v	Desvio padrão, com relação a medidas de velocidade
\mathbf{v}	Vetor de ruídos de medida
ϕ, θ, ψ	Ângulos de Euler
φ	Latitude geodésica
$\varphi(\cdot)$	Função de ativação
$\Phi_{k,k-1}$	Matriz de transição de estados
ω_e	Magnitude da rotação da Terra
ω_{ie}^e	Velocidade angular de (e), em relação à (i), expressa em (e)
ω_{en}^n	Velocidade angular de (n), em relação à (e), expressa em (n)
ω_{in}^n	Velocidade angular de (n), em relação à (i), expressa em (n)
ω_{ib}^b	Velocidade angular de (b), em relação à (i), expressa em (b)
$\tilde{\omega}_{ib}^b$	Velocidade angular medida pela IMU
$\omega_N, \omega_E, \omega_D$	Componentes da velocidade ω_{en}^n
ω	Vetor dos ruídos de processo
Ω	Forma, ou matriz, anti-simétrica de um vetor ω

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO.....	1
1.1 Apresentação do tema	1
1.2 Revisão bibliográfica	3
1.3 Objetivo e organização do trabalho	7
2 FUNDAMENTOS DE NAVEGAÇÃO INERCIAL.....	11
2.1 Sistema de posicionamento global – GPS.....	11
2.2 Sistemas de coordenadas	11
2.2.1 Sistema Inercial (i)	12
2.2.2 Sistema Terrestre (e).....	12
2.2.3 Sistema de Navegação (n)	13
2.2.4 Sistema do Veículo (b).....	13
2.2.5 Transformação entre os sistemas do Veículo e de Navegação.....	14
2.3 Equações de navegação inercial	18
2.4 Equações simplificadas de navegação inercial para um veículo terrestre.....	23
2.5 Integração dos sistemas GPS/INS.....	28
3 FUNDAMENTOS DE REDE NEURAS ARTIFICIAIS.....	31
3.1 Introdução	31
3.2 Redes perceptron de múltiplas camadas (MLP).....	34
3.3 Treinamento de redes MLP	36
3.4 Algoritmo de retropropagação	37
4 ABORDAGEM COM O FILTRO DE KALMAN.....	41
4.1 Introdução	41
4.2 Solução com o Filtro estendido de Kalman.....	41
4.3 Solução adaptativa: estimação do ruído de estado	45
4.4 Pré-filtragem de dados de treinamento	48

5	NAVEGAÇÃO INERCIAL AUXILIADA POR REDES NEURAIS.....	53
5.1	Definição da RNA para o problema de posicionamento	53
5.2	Modos de operação da RNA	55
5.3	Métodos de treinamento para o sistema de navegação proposto	58
5.3.1	Método 1	58
5.3.2	Método 2	59
5.3.3	Método 3	59
5.3.4	Método 4	60
6	SIMULAÇÃO E RESULTADOS	63
6.1	Introdução	63
6.2	Verificação <i>off-line</i> do algoritmo de treinamento proposto	63
6.2.1	Descrição dos dados utilizados na verificação <i>off-line</i>	65
6.2.2	Resultados da verificação <i>off-line</i>	68
6.2.3	Considerações sobre o número de neurônios ocultos e de iterações	77
6.3	Simulação dos métodos propostos para o sistema de navegação	80
6.3.1	Simulação no intervalo [50s, 500s].....	81
6.3.2	Simulação no intervalo [630s, 1000s]	85
6.3.3	Simulação com um segundo conjunto de dados	91
6.3.4	Simulações complementares	100
7	COMENTÁRIOS FINAIS E CONCLUSÕES.....	107
7.1	Introdução	107
7.2	Análise dos resultados e conclusões	108
7.3	Sugestões para extensões e trabalhos futuros	111
	REFERÊNCIAS BIBLIOGRÁFICAS.....	113
	APÊNDICE A – CARACTERÍSTICAS DA IMU e RECEPTOR GPS.....	119
	APÊNDICE B – Matriz de derivadas parciais H.....	121

1 INTRODUÇÃO

1.1 Apresentação do tema

Atualmente, com a disponibilidade de receptores NAVSTAR-GPS (*NAVgation Satellite with Time and Ranging*), ou simplesmente GPS (*Global Positioning System*), que atendem a uma ampla gama de aplicações e preços, a tecnologia GPS domina o mercado de posicionamento e navegação (POS/NAV). O sistema, além de ser globalmente disponível, é altamente portátil, possui baixo consumo de potência e é adequado para integração com outros sensores. Estas características fazem com que novas aplicações de baixo custo surjam constantemente, ou aquelas já consideradas tradicionais se tornem mais acessíveis, sendo possível citar dentre outras: navegação veicular terrestre, aérea, marítima e fluvial; cartografia; meio ambiente; robótica; agricultura; aplicações militares; lazer; e esportes.

A necessidade de sistemas de posicionamento e navegação alternativos só ocorre porque o GPS não trabalha em todos os ambientes, ou pode trabalhar de forma inadequada. Ou seja, um sistema de navegação baseado somente na utilização do GPS pode não operar satisfatoriamente em ambientes com obstáculos artificiais (por exemplo, construções civis) ou naturais (florestas, montanhas, vales) ou em ambientes sujeitos à intensa interferência radiométrica, proposital ou não, que podem degradar ou bloquear parcialmente ou totalmente o sinal proveniente dos satélites; ou em situações dinâmicas onde o posicionamento da antena pode ser desfavorecido por estar sujeito a deslocamentos, curvas, etc.

Diferentes soluções para a integração de sensores ou base de dados (*map matching*) são constantemente propostas para suprir a falta de informações na porção do tempo onde o GPS não opera (EL-SHEIMY; NIU, 2007). A integração das informações do GPS com um sistema de navegação inercial, ou INS (*Inertial Navigation System*), é uma alternativa de solução dos problemas citados anteriormente. Durante a ausência de sinal do GPS, um sistema de navegação inercial pode prover informações para navegação. Um sistema de navegação inercial é composto de uma unidade de medida inercial ou

IMU (*Inertial Measurement Unit*), que é composta por um arranjo de acelerômetros e giroscópios, e de um processador de informações. Porém a deriva dos giroscópios e o erro sistemático dos acelerômetros causam degradação na solução de navegação (TITTERTON; WESTON, 2004).

Portanto GPS e INS são sistemas complementares. As informações do GPS (posicionamento, velocidade e tempo) permanecem estáveis por longos períodos, e quando o sinal de recepção é interrompido e/ou a geometria dos satélites não está dentro de limites aceitáveis, pode ser integrado ao sistema de navegação inercial. As posições e velocidades derivadas do sistema GPS são medidas externas que podem atualizar o INS melhorando sua estabilidade e precisão de longo termo.

IMUs de baixo custo são possíveis graças à tecnologia *Micro Electrical Mechanical Systems* ou MEMS, também conhecida como *Micro System Technology* (MST), que é empregada na fabricação de diversos tipos de sensores comerciais. Os sensores inerciais MEMS, giroscópios e acelerômetros, apresentam vantagens em relação ao peso, dimensão, resistência ao impacto, consumo de energia e custo reduzido, pois podem ser fabricados em grande escala (YAZDI et al., 1998). Porém, quando comparados com sensores convencionais, os MEMS exibem erros bem maiores, em especial as derivas e nível de ruído dos giroscópios.

Técnicas estocásticas são tradicionalmente utilizadas para estimar, compensar e limitar os erros dos sensores inerciais e para integrar o sistema GPS/INS, sendo o filtro de Kalman o método mais utilizado na integração entre GPS, sensores inerciais da IMU e o INS (FARREL; BARTH, 1998). O filtro de Kalman possui a capacidade de incorporar todas as medições disponíveis, contanto que a dinâmica (modelo) do sistema e dos sensores seja bem estabelecida; que exista uma descrição estatística dos ruídos do sistema, dos erros de medições e incertezas dos modelos; e que sejam conhecidas as condições iniciais das variáveis de interesse. Nos períodos quando o sinal GPS não está disponível, o filtro opera somente na fase de propagação com aumento da incerteza nas estimativas.

Uma abordagem proposta para reduzir o impacto de tais limitações é a incorporação de técnicas de inteligência artificial (IA) nos algoritmos de integração, gerando assim soluções híbridas (CHIANG; EL-SHEIMY, 2005). Este é o tema principal proposto para este trabalho, que aborda o emprego de redes neurais artificiais no auxílio aos sistemas inerciais de baixo custo, nos períodos de tempo em que não é possível utilizar as informações do GPS.

1.2 Revisão bibliográfica

Redes neurais artificiais se constituem em uma técnica de inteligência artificial com capacidade de aprender, por meio de exemplos, e de generalizar as informações aprendidas. São capazes de atuar, dentre outras características importantes, como mapeadores universais de funções multivariáveis, possuem capacidade de auto-organização e de processamento temporal. A forma mais comum de utilização das RNA é o aprendizado por meio de um conjunto de dados, e suas principais aplicações são (BRAGA et al., 2007; HAYKIN, 2001; NORGAARD et al., 2000): **classificação** (reconhecimento de caracteres e imagens, diagnóstico e detecção de falhas, por exemplo); **categorização** (agrupamento ou *clustering*, mineração de dados e agrupamento de seqüência de DNA, por exemplo); **aproximação** (mapeamento das variáveis de entrada e saída de funções contínuas); **previsão**; **otimização** e **controle** de sistemas dinâmicos.

Uma etapa fundamental no uso das RNA é o aprendizado, que consiste em um processo iterativo de ajuste dos parâmetros da rede que armazenam, ao final da etapa, o conhecimento adquirido a partir de um ambiente externo, ou seja, a partir dos dados de treinamento. Neste trabalho é explorado o filtro de Kalman no treinamento das RNA. O filtro de Kalman estendido é base para métodos de treinamento alternativos aos métodos tradicionais como, por exemplo, o Levenberg-Marquardt e Gradiente Conjugado (HAYKIN, 2001), com a possibilidade de assumir uma arquitetura que aceita a apresentação de dados de treinamento de uma forma seqüencial, ao contrário dos outros métodos citados que trabalham com dados em lote (*batch mode*). O método foi, inicialmente, proposto por Singhal e Wu (1989) para redes neurais multicamadas

estáticas, e visto como um problema de identificação para um sistema não linear. Embora tenha apresentado bons resultados em relação à convergência em poucas iterações, o método era, na época, computacionalmente complexo para redes neurais com muitos parâmetros, uma vez que a dimensão da matriz de covariância cresce com o quadrado do número de parâmetros. O método se tornou a base de uma família de técnicas de treinamentos com inúmeras derivações propostas para seu melhoramento, tanto em relação ao desempenho como sua extensão às redes recorrentes (IIGUNI et al., 1992; PUSKORIUS; FELDKAMP, 1991; RIOS NETO, 1995, 1997; SHAH; PALMIERI, 1990).

A abordagem do uso de RNA em navegação inercial, como em outras aplicações, implica ser necessária, na fase de projeto, a definição da classe de RNA, da sua arquitetura, do algoritmo de treinamento *on-line* e *off-line*, do critério de parada para o treinamento, e a seleção de sinais de entrada e saída. O ponto em comum entre as diversas concepções é a existência de dois modos de operação básicos: o modo treinamento, enquanto o sinal GPS está disponível, e o modo predição, quando o sinal GPS está indisponível ou degradado.

Ao longo da última década várias classes de RNA foram investigadas em aplicações voltadas à navegação inercial. As mais comuns, *perceptrons* de múltiplas camadas, ou MLPs (*multilayer perceptrons*), serão revisadas com mais detalhes no presente trabalho. Porém, devem ser citadas a rede mapa auto-organizável ou SOM, de *self-organizing map*, investigada por Forrest et al. (2000) para a fusão de sensores inerciais; a rede de Hopfield, estudada por Shi et al. (2005), na estimação de estado de um sistema INS/GPS; e por último a rede de função de base radial, ou RBF, proposta por Noureldin et al. (2004), treinada para predizer o erro de posicionamento na ausência de sinal GPS.

A primeira referência encontrada sobre o assunto, Dumville e Tsakiri (1994), investiga a integração, via RNA, entre as informações de um GPS diferencial (DGPS) e um sistema DR (*dead-reckoning*) composto por odômetro e bússola, para um veículo terrestre em uso urbano. A arquitetura proposta foi uma MLP com uma camada oculta e retropropagação de erros com base na regra Delta generalizada.

Ibrahim et al. (2000) apresentaram um esquema de integração INS/DGPS para veículo de uso terrestre utilizando redes MLP com duas camadas ocultas. Duas redes foram treinadas para direções nas coordenadas norte e leste, de forma desacoplada. O método de treinamento utilizado foi o filtro de Kalman.

Chiang e El-Sheimy (2002), e de maneira similar Chiang et al. (2003), propuseram uma rede MLP para a fusão de dados entre um sistema de navegação inercial, INS, e DGPS. A arquitetura proposta foi uma MLP, com uma camada oculta e retropropagação de erros com o algoritmo do gradiente conjugado e regra Delta generalizada. O sistema foi testado num veículo terrestre e duas classes de IMU foram utilizadas, uma classe táctica (Honeywell HG1700) e outra classe navegação (Litton LTN 90-100).

Wang e Gao (2003) investigaram um sistema de navegação inercial de baixo custo, composto por um conjunto de dois acelerômetros MEMs (ADLX202E), uma bússola digital (Honeywell HMR3300) calibrada, e auxiliado por GPS (JAVAD-Legacy). O sistema foi proposto para veículo terrestre urbano utilizando um algoritmo de navegação restrito, ou seja, com equações de navegação reduzidas objetivando a redução de erros. Foi proposta uma RNA, do tipo MLP, com uma camada oculta.

El-Sheimy et al. (2006) propuseram duas arquiteturas de RNA, do tipo MLP, com algoritmo do gradiente conjugado, para integrar um sistema de navegação inercial com um DGPS. Testes em campo foram realizados com uma IMU Honeywell 1700 e um NovAtel GPS 600 montados num veículo terrestre de uso urbano. A primeira arquitetura proposta foi denominada PUA de *Position Update Architecture*. A segunda arquitetura proposta contém duas RNA operando em paralelo, a PUA, mencionada anteriormente, e a VUA, de *Velocity Update Architecture*, e é denominada PVUA. A conclusão apresentada foi que ambas as soluções PUA e PVUA apresentaram um resultado melhor que a solução por filtro de Kalman, durante o período de ausência de sinal do DGPS.

Chiang et al., 2004 propuseram um método para ajuste on-line dos pesos sinápticos de uma RNA, com arquitetura MLP, utilizada para integrar um sistema de navegação inercial com um DGPS. O método é chamado de *window-based weight updating*

method. O método considera que os pesos acumulados anteriormente são aplicados como pesos iniciais para uma nova partida (missão) e ajustados em certos intervalos de tempo, as “janelas”, nos quais o sinal DGPS está disponível. Nestes intervalos, certa quantidade de informações é utilizada no modo de treinamento por lote, e um novo conjunto de pesos é acumulado e utilizado para inicializar o treinamento na próxima “janela”. Se o sinal DGPS fica indisponível o último conjunto de pesos é utilizado na RNA no modo predição. Se o procedimento é aplicado pela primeira vez, os pesos podem ser inicializados aleatoriamente.

Chiang et al. (2006) e Huang e Chiang (2007), sugerem a integração de um sistema de navegação inercial e um DGPS, utilizando as Redes Neurais Construtivas (RNC). Dentre os vários algoritmos construtivos existentes, foi utilizado o *Cascade-Correlation*, ou CasCor, proposto por Fahlman e Lebiere (1990). O princípio comum das RNC é começar como uma pequena rede e adicionar neurônios e camadas ocultas, conforme necessário, durante a fase de treinamento, utilizando algoritmos especiais. Palma Neto e Nicoletti (2005) fornecem mais detalhes sobre os algoritmos das RNC.

Kaygiziz (2004) e posteriormente Kaygiziz et al. (2007), propuseram um esquema no qual uma rede MLP é treinada, para que na ausência do sinal GPS a RNA, previamente treinada, estime os dados de posição para a INS utilizando a estrutura do filtro de Kalman empregada na integração GPS/INS. O esquema proposto foi testado num veículo terrestre de uso urbano. O esquema proposto para a RNA é composto de três MLP paralelas, uma para cada direção, e que recebe os mesmos sinais de entrada. O algoritmo de treinamento utilizado foi o Levenberg-Marquardt.

Chiang e Huang (2008) modificaram seus trabalhos anteriores, (CHIANG et al., 2004; e EL-SHEIMY et al., 2006), visando melhorias na capacidade de generalização. Foi proposta uma variação na arquitetura da RNA, em relação à utilizada anteriormente, denominada PVAUA (*position, velocity and azimuth update*). A arquitetura PVAUA consiste em três MLP operando em paralelo. Para melhorar a capacidade de generalização da rede, foi proposto uma base de dados chamada NAVi (*navigation information database*) para guardar o conhecimento adquirido durante o treinamento e

prover memória de longo prazo ao sistema. O NAVi é constituído pelos pares de treinamento (vetores de entrada e correspondentes de saída) e pelos pesos de cada rede.

Noureldin et al. (2010) trabalham com o conceito de rede neural dinâmica com entradas atrasadas (*Input-Delayed Neural Network, IDNN*), aplicadas na integração de sistemas INS/GPS. O trabalho aborda o uso de elementos de entradas atrasadas, na camada de entrada da RNA, de forma que sua modelagem é baseada nos padrões de treinamento atuais e passados, ou atrasados, com dependência do tempo.

De Boer et al. (2009) implementa uma estratégia para o uso de RNA na compensação dos erros de navegação, de uma IMU baseada em sensores MEMS, nos cenários críticos em que o GPS não funciona. É utilizada uma RNA do tipo MLP, treinada pelo algoritmo *backpropagation*, num esquema *online* baseado no movimento atrasado no tempo do veículo, enquanto há informação do GPS. São apresentados resultados para o veículo parado, com velocidade constante e numa trajetória real.

Aggarwal et al. (2010) revisa, no Capítulo 7 do livro, o papel das redes neurais artificiais no campo da navegação integrada baseada em sensores MEMS. São apresentados os princípios básicos e diferentes tipos de RNA; bem como algumas abordagens, técnicas e análise na utilização das RNA na navegação inercial.

1.3 Objetivo e organização do trabalho

Como mencionado anteriormente, técnicas de IA podem ser utilizadas para a compensação de erros em sensores inerciais de baixo custo, quando informações provenientes do GPS não forem disponíveis. Neste contexto, o objetivo geral proposto para este trabalho é utilizar uma rede neural artificial (RNA), para que seja treinada enquanto existe sinal de GPS, e utilizada para prever, com compensação dos erros da IMU, a posição de um veículo quando da ausência do sinal GPS, ou seja, assumindo a função de navegador do sistema. De uma forma mais detalhada, propõem-se então os seguintes objetivos específicos:

- a) Propor, implementar e testar algoritmos, baseados em RNA, que possibilitem a compensação de erros de posicionamento de um sistema de navegação inercial auxiliado por GPS, baseado em IMUs de baixo custo, garantindo desempenho adequado nos períodos de ausência do sinal GPS.
- b) Modelar uma RNA dedicada à navegação inercial, definindo: os sinais de entrada e saída, seus modos de operação, e formas de apresentação dos padrões de treinamento.
- c) Implementar algoritmos baseados em estimação linear ótima, em especial o filtro de Kalman, para o treinamento *off-line* e *on-line* da RNA .
- d) Implementar procedimento adaptativo com o objetivo de melhorar o resultado da aprendizagem.

Este trabalho está organizado de acordo com a seqüência de capítulos descrita a seguir.

O Capítulo 2 contém os fundamentos de navegação inercial, com o objetivo de deduzir as equações diferenciais simplificadas, de navegação para um veículo terrestre de uso urbano, e que servirão para a modelagem da RNA a ser utilizada ao longo deste trabalho.

No Capítulo 3 estão apresentados os elementos de redes neurais artificiais, em especial as redes alimentadas adiante, do tipo *perceptron* de múltiplas camadas (MLP).

Os conceitos de treinamento de RNA e do algoritmo de retropropagação são apresentados, preparando e antecedendo a discussão da abordagem de treinamento com o filtro de Kalman adaptativo, a ser feita no Capítulo 4.

No Capítulo 4 é descrita, inicialmente, a abordagem de treinamento de RNA com o filtro de Kalman estendido. Na seqüência, é apresentada a solução adaptativa, com a estimação do ruído de estado. Finalmente, é proposto um algoritmo para pré-filtragem dos sinais utilizados para padrões de treinamento, também na forma de um filtro de Kalman.

No Capítulo 5 são definidos os sinais de entrada e saída da RNA dedicada para navegação inercial, tendo como justificativa as equações de navegação simplificadas, apresentadas no Capítulo 2. Na sequência são descritos os modos de operação da RNA: predição e treinamento. Por fim são propostos métodos de treinamento, para a RNA dedicada à navegação, e formas de apresentar os padrões de treinamento.

No Capítulo 6 os algoritmos e métodos propostos são analisados por simulação numérica, utilizando dados reais obtidos de uma IMU e receptor GPS montados num veículo urbano. Os resultados preliminares são apresentados na forma de erros de posicionamento do veículo, com a RNA no modo predição, ou seja, durante o intervalo de tempo sem sinal GPS.

Finalmente, a discussão dos resultados obtidos e sugestões para extensão deste trabalho, são apresentadas no Capítulo 7.

2 FUNDAMENTOS DE NAVEGAÇÃO INERCIAL

Este capítulo apresenta os principais conceitos envolvidos neste trabalho, com relação aos sistemas de coordenadas e navegação inercial, com o objetivo de auxiliar a modelagem da rede neural a ser utilizada nos próximos capítulos.

2.1 Sistema de posicionamento global – GPS

Como mencionado anteriormente, o GPS ou NAVSTAR-GPS é um sistema de radio navegação, de abrangência global, desenvolvido pelo Departamento de Defesa dos Estados Unidos da América. Sua concepção permite que um usuário, em qualquer local na superfície terrestre, tenha à sua disposição um número mínimo de satélites que possibilite a estimativa da sua posição em tempo real e sob diversas condições climáticas. O GPS consiste de três segmentos principais: o segmento espacial, que é a constelação de satélites distribuídos em seis planos orbitais e que transmitem sinais de navegação gerados a bordo; o segmento de controle, composto por estações terrestres monitoras cuja tarefa é a monitoração, controle e manutenção dos satélites; e o segmento de usuários, constituído pelos receptores GPS da comunidade de usuários civis e militares. A descrição, fundamentos e aplicações da tecnologia GPS podem ser encontrados em Monico (2000), Greenspan (1996) e Kaplan (2006), entre outras.

2.2 Sistemas de coordenadas

Os estados de um sistema de navegação, posição, velocidade e atitude, são definidos com referência a diferentes sistemas de coordenadas, os quais permitem que medidas de sensores inerciais ou estimativas da posição da antena de um receptor GPS sejam relacionadas com as direções cardinais da Terra, ou seja, com coordenadas que tenham significado físico em se tratando de navegação nas proximidades da superfície Terrestre. A Figura 2.1 ilustra os sistemas utilizados neste trabalho:

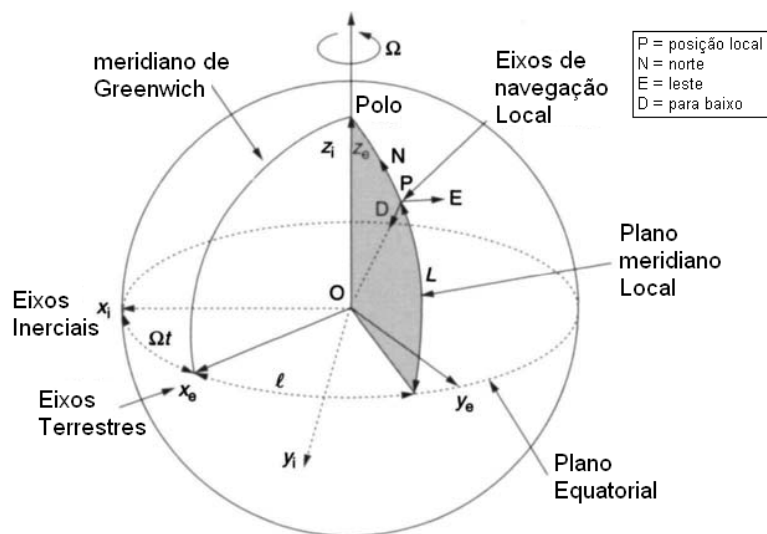


Figura 2.1 - Sistemas de coordenadas.

Fonte: adaptado de Titterton e Weston (2004).

2.2.1 Sistema Inercial (i)

Num sistema referencial inercial são válidas as Leis de Newton, a partir das quais as equações de navegação são derivadas. Também chamado de ECI (Earth-Centered Inertial), o referencial inercial possui as seguintes características:

- Origem no centro de massa da Terra;
- Eixo Z_i é paralelo ao eixo de rotação da Terra e aponta para o norte geográfico;
- Eixo X_i aponta para o equinócio vernal;
- Eixo Y_i completa um sistema ortogonal dextrogiro.

2.2.2 Sistema Terrestre (e)

Também chamado de ECEF (Earth-Centered, Earth-Fixed), o referencial Terrestre possui as seguintes características:

- Origem no centro de massa da Terra;

- b) Eixo Z_e é paralelo ao eixo de rotação da Terra e aponta para o norte geográfico;
- c) Eixo X_e aponta para a intersecção entre o Equador da Terra e o meridiano de Greenwich;
- d) Eixo Y_e completa um sistema dextrogiro.

Ou seja, possui a mesma origem que o sistema ECI, mas gira com a velocidade de rotação da Terra (Ω).

2.2.3 Sistema de Navegação (n)

Também chamado de nível local ou LTP (*Local Tangent Plane*), este referencial de navegação possui as seguintes características:

- a) Origem no C.G. do veículo;
- b) Eixo Z_n aponta para baixo, perpendicularmente ao plano local tangente ao elipsóide de referência;
- c) Eixo X_n aponta para o norte geográfico;
- d) Eixo Y_n aponta para o leste geográfico.

Com essas características, o sistema LTP também é chamado de coordenadas NED (*North-East-Down*), de acordo com Grewall et al. (2001). Caso o eixo Z_n apontasse para cima seria chamado de coordenadas ENU (*East-North-Up*). A mecanização no sistema LTP é frequentemente utilizada para propósitos de navegação, pois sua saída pode ser transformada nas coordenadas geográficas de latitude, longitude e altitude.

2.2.4 Sistema do Veículo (b)

O sistema referencial de coordenadas do veículo, também conhecido como do corpo, possui as seguintes características:

- a) É rigidamente preso ao C.G. do veículo;
- b) Eixo X_b é longitudinal, contido no plano de simetria do veículo e aponta na direção preferencial do seu movimento.
- c) Eixo Z_b é ortogonal ao eixo X_b , também contido no plano de simetria do veículo e aponta para baixo na direção do seu fundo;
- d) Seu eixo Y_b completa um sistema de coordenadas ortogonal dextrogiro.

A notação $[u, v, w]^T$, ao invés de $[X_b, Y_b, Z_b]^T$, também é utilizada em aplicações aeronáuticas para representar o vetor velocidade da aeronave relativo ao sistema do corpo. Nos eixos do sistema de coordenadas do veículo são definidos os movimentos angulares que definem sua atitude, conhecidos como ângulos de rolamento, arfagem e guinada (*roll, pitch, yaw* ou *heading*).

As velocidades angulares, medidas pelos giroscópios, são denotadas por $[p, q, r]$ e são positivas de acordo com a regra da mão direita, conforme mostrado na Figura 2.2.

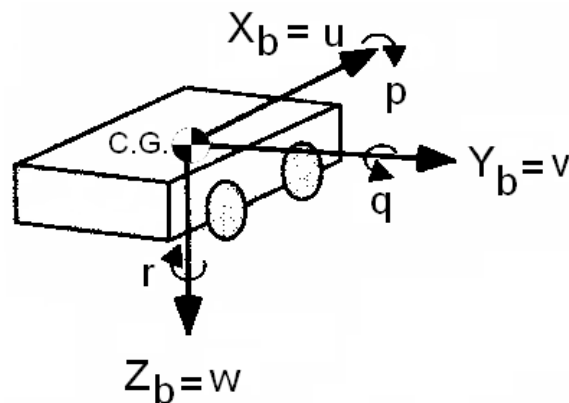


Figura 2.2 - Sistema referencial do veículo.

Fonte: adaptado de Farrel e Barth (1998).

2.2.5 Transformação entre os sistemas do Veículo e de Navegação

Existem métodos que representam e implementam transformações de coordenadas através de rotações planares, tais como os Ângulos de Euler, Matrizes de Cossenos

Diretores (DCM) e Quatérnios. Os Ângulos de Euler relacionam medidas em dois sistemas de coordenadas através de uma série de três rotações ortogonais entre os eixos dos sistemas. A forma analítica final da matriz de transformação depende da ordem com as quais as rotações são tomadas, sendo comum a seguinte ordem (FARREL; BARTH, 1998 e TITTERTON; WESTON, 2004):

- a) Rotacionar, positivamente, o eixo de referência Z, de uma rotação finita $\mathbf{R}_Z(\psi)$, com um ângulo ψ ;
- b) Rotacionar o eixo resultante Y' de uma rotação finita $\mathbf{R}_{Y'}(\theta)$, de um ângulo θ ;
- c) Rotacionar o eixo resultante X'' de uma rotação finita $\mathbf{R}_{X''}(\phi)$, de um ângulo ϕ .

Ou seja,

$$\mathbf{R}_n^b = \mathbf{R}_{X''}(\phi) \mathbf{R}_{Y'}(\theta) \mathbf{R}_Z(\psi) \quad (2.1)$$

$$\mathbf{R}_n^b = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - s\psi c\phi & s\psi s\theta s\phi + c\psi c\phi & s\phi c\theta \\ c\psi c\phi s\theta + s\psi s\phi & s\psi c\phi s\theta - c\psi s\phi & c\phi c\theta \end{bmatrix} \quad (2.2)$$

Como a matriz \mathbf{R}_n^b é ortogonal a relação $\mathbf{R}_b^n = \left(\mathbf{R}_n^b\right)^{-1} = \left(\mathbf{R}_n^b\right)^T$, então

$$\mathbf{R}_b^n = \begin{bmatrix} c\theta c\psi & -c\phi s\psi + s\phi s\theta c\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -s\phi c\psi + c\phi s\theta s\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad (2.3)$$

onde c denota cosseno, s denota seno e \mathbf{R}_b^n denota a matriz de rotação que transforma do Sistema de Coordenadas do Veículo (b), ou Corpo, para o Sistema de Navegação (n).

Os ângulos ϕ, θ, ψ são conhecidos como Ângulos de Euler e correspondem aos ângulos calculados ou medidos a partir de sensores, usualmente giroscópios (*roll, pitch, yaw*).

A transformação entre os Sistemas Terrestre e de Navegação é dada por:

$$\mathbf{R}_e^n = \mathbf{R}_Y \left(-\phi - \frac{\pi}{2} \right) \mathbf{R}_Z (\lambda) \quad (2.4)$$

$$\mathbf{R}_e^n = \begin{bmatrix} -s\phi c\lambda & -s\phi s\lambda & c\phi \\ -s\phi & c\lambda & 0 \\ -c\phi c\lambda & -c\phi s\lambda & -s\phi \end{bmatrix} \quad (2.5)$$

onde ϕ denota a latitude geodésica e λ a longitude geodésica, e novamente s denota seno e c denota cosseno.

O efeito da velocidade de rotação da Terra é observado por $\boldsymbol{\omega}_{ie}^e$, ou seja, a velocidade do sistema referencial Terrestre em relação ao inercial, expressa no sistema Terrestre.

$$\boldsymbol{\omega}_{ie}^e = \begin{bmatrix} 0 & 0 & \omega_e \end{bmatrix}^T \quad (2.6)$$

Expressa no sistema de navegação, toma a seguinte forma:

$$\boldsymbol{\omega}_{ie}^n = \mathbf{R}_e^n \boldsymbol{\omega}_{ie}^e = \begin{bmatrix} \omega_e \cos\phi \\ 0 \\ -\omega_e \sin\phi \end{bmatrix} \quad (2.7)$$

onde $\omega_e = 7,2921158 \times 10^{-5}$ rad/s é a magnitude da rotação da Terra.

O vetor $\boldsymbol{\omega}_{en}^n$ denota as velocidades angulares do Sistema de Navegação em relação ao Sistema Terrestre, expressas no sistema de navegação, e que são obtidas em função da taxa de variação da latitude e longitude (FARREL; BARTH, 1998 e TITTERTON; WESTON, 2004):

$$\boldsymbol{\omega}_{en}^n = \begin{bmatrix} \omega_N \\ \omega_E \\ \omega_D \end{bmatrix} = \begin{bmatrix} \dot{\lambda} \cos \varphi \\ -\dot{\varphi} \\ -\dot{\lambda} \sin \varphi \end{bmatrix} \quad (2.8)$$

onde,

$$\dot{\lambda} = \frac{v_E}{\left(R_N + h \right) \cos \varphi} \quad (2.9)$$

$$\dot{\varphi} = \frac{v_N}{\left(R_M + h \right)} \quad (2.10)$$

Resultando em:

$$\boldsymbol{\omega}_{en}^e = \begin{bmatrix} \omega_N \\ \omega_E \\ \omega_D \end{bmatrix} = \begin{bmatrix} \frac{v_E}{\left(R_N + h \right)} \\ -\frac{v_N}{\left(R_M + h \right)} \\ -\frac{v_E \tan \varphi}{\left(R_N + h \right)} \end{bmatrix}, \quad (2.11)$$

onde o vetor $\boldsymbol{\omega}_{en}^e$ denota as velocidades angulares do Sistema de Navegação em relação ao Sistema Terrestre e expressas também no Sistema Terrestre, h é a altitude do C.G. do veículo em relação à superfície do elipsóide, R_M é o raio de curvatura meridional e R_N o raio de curvatura transversal, definidos por (FARREL; BARTH, 1998 e TITTERTON; WESTON, 2004):

$$R_N = \frac{a}{\left(1 - e^2 \sin^2 \varphi \right)^{1/2}} \quad (2.12)$$

$$R_M = \frac{a(1 - e^2)}{(1 - e^2 \sin^2 \varphi)^{3/2}} \quad (2.13)$$

Com relação às Equações 2.12 e 2.13, e ao elipsóide de referência, e denota sua excentricidade e a o seu semi-eixo maior.

Finalmente o vetor $\boldsymbol{\omega}_{in}^n$, que descreve as velocidades angulares do Sistema de Navegação em relação ao inercial, expressas no referencial de navegação, é dado por:

$$\boldsymbol{\omega}_{in}^n = \boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n \quad (2.14)$$

$$\boldsymbol{\omega}_{in}^n = \begin{bmatrix} \omega_e \cos \varphi + \frac{v_E}{(R_N + h)} \\ \frac{-v_N}{(R_M + h)} \\ -\omega_e \sin \varphi - \frac{v_E \tan \varphi}{(R_N + h)} \end{bmatrix} \quad (2.15)$$

2.3 Equações de navegação inercial

Navegação inercial pode ser definida como o processo de estimar, ao longo do tempo, a posição e velocidade, linear e angular, de um veículo, com relação a um sistema de referência (coordenadas) adequado, através de medições feitas por uma unidade de medida inercial, IMU, composta por acelerômetros, para medir forças específicas lineares, e giroscópios, ou girômetros, para medir as velocidades angulares do veículo em questão.

Sistemas de navegação inercial tipo *strapdown*, ou SINS (*strapdown inertial navigation system*), utilizam arranjos de acelerômetros e giroscópios rigidamente montados nos

eixos dos veículos. As informações de atitude e posicionamento são integradas a partir das medidas da velocidade angular e da força específica, descritas num sistema de coordenadas conveniente. Essa integração, também chamada de mecanização das medidas inerciais, pode ser feita em diferentes sistemas de coordenadas. Os conceitos e tecnologia de SINS são amplamente abordados em Titterton e Weston (2004), Farrel e Barth (1998) e Grewal et al. (2001). A Figura 2.3 mostra um esquema simplificado da mecanização de um INS.

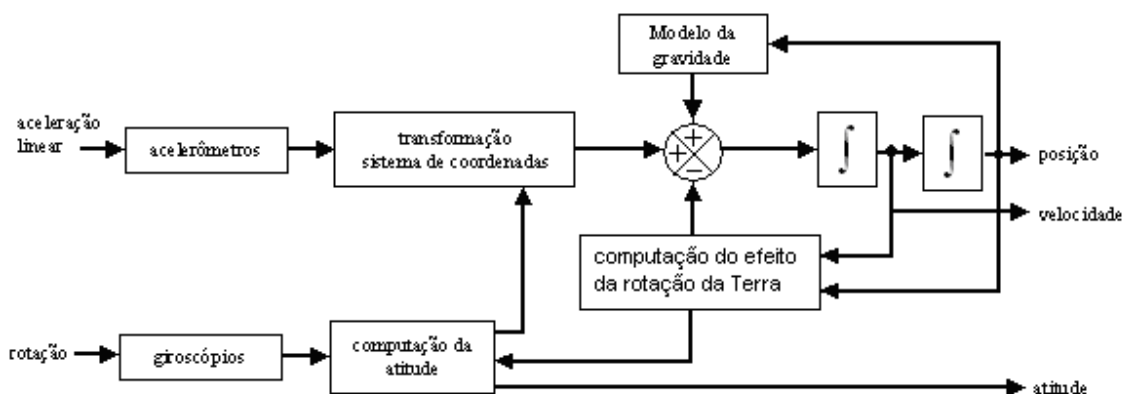


Figura 2.3 – Fluxograma simplificado de um sistema de navegação inercial

No Sistema de Navegação (LTP), as coordenadas de posição (latitude, longitude e altitude) são dadas, respectivamente, por:

$$\mathbf{p}^n = [\varphi \quad \lambda \quad h]^T \quad (2.16)$$

Cujas derivadas são denotadas por:

$$\dot{\mathbf{p}}^n = [\dot{\varphi} \quad \dot{\lambda} \quad \dot{h}]^T \quad (2.17)$$

As velocidades do veículo, expressas no sistema de navegação nas coordenadas NED, são representadas por:

$$\mathbf{v}^n = \begin{bmatrix} v_N & v_E & v_D \end{bmatrix}^T \quad (2.18)$$

A relação entre \mathbf{v}^n e $\dot{\mathbf{p}}^n$ é dada pelas seguintes equações (FARREL; BARTH, 1998 e TITTERTON; WESTON, 2004):

$$\mathbf{v}^n = \begin{bmatrix} v_N \\ v_E \\ v_D \end{bmatrix} = \begin{bmatrix} (R_M + h) & 0 & 0 \\ 0 & (R_N + h) \cos(\varphi) & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} \quad (2.19)$$

$$\dot{\mathbf{p}}^n = \begin{bmatrix} \dot{\phi} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \mathbf{D}^{-1} \begin{bmatrix} v_N \\ v_E \\ v_D \end{bmatrix} \quad (2.20)$$

onde,

$$\mathbf{D}^{-1} = \begin{bmatrix} \frac{1}{(R_M + h)} & 0 & 0 \\ 0 & \frac{1}{(R_N + h) \cos(\varphi)} & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (2.21)$$

As equações de mecanização, descritas no sistema de navegação, são dadas por (FARREL; BARTH, 1998 e TITTERTON; WESTON, 2004):

$$\dot{\mathbf{v}}^n = \mathbf{R}_b^n \mathbf{f}^b - \left(2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n \right) \times \mathbf{v}^n + \mathbf{g}^n, \quad (2.22)$$

onde \mathbf{g}^n é o vetor gravidade e \mathbf{f}^b denota o vetor força específica, computado a partir das medições acelerométricas $\tilde{\mathbf{f}}$ da IMU:

$$\mathbf{f}^b = \begin{bmatrix} \tilde{f}_u \\ \tilde{f}_v \\ \tilde{f}_w \end{bmatrix} - \begin{bmatrix} b_u \\ b_v \\ b_w \end{bmatrix}, \quad (2.23)$$

O vetor $\begin{bmatrix} b_u, b_v, b_w \end{bmatrix}^T$ representa as correções, ou *bias*, estimadas durante o processo de navegação.

A matriz de transformação \mathbf{R}_b^n é obtida pela seguinte equação diferencial:

$$\dot{\mathbf{R}}_b^n = \mathbf{R}_b^n \mathbf{\Omega}_{nb}^b, \quad (2.24)$$

onde, $\mathbf{\Omega}$ denota a forma, ou matriz, anti-simétrica de um vetor, e $\mathbf{\Omega}_{nb}^b$ é dada por:

$$\mathbf{\Omega}_{nb}^b = \mathbf{\Omega}_{ib}^b - \mathbf{\Omega}_{in}^b. \quad (2.25)$$

Então,

$$\dot{\mathbf{R}}_b^n = \mathbf{R}_b^n \left(\mathbf{\Omega}_{ib}^b - \mathbf{\Omega}_{in}^b \right). \quad (2.26)$$

A computação da Equação 2.26 pode ser crítica em sistemas do tipo *strapdown*, especialmente em aplicações que exigem atualização da atitude em alta frequência. Neste caso são utilizados métodos mais elaborados, do que a simples integração direta, com o objetivo de preservar as propriedades de ortogonalidade da matriz de rotação, bem como tornar os cálculos mais robustos e eficientes do ponto de vista computacional (TITTERTON; WESTON, 2004).

A matriz $\mathbf{\Omega}_{ib}^b$ é obtida a partir das medições giroscópicas $\tilde{\mathbf{\omega}}_{ib}^b$ da IMU:

$$\tilde{\mathbf{\omega}}_{ib}^b = \begin{bmatrix} \tilde{p} \\ \tilde{q} \\ \tilde{r} \end{bmatrix} - \begin{bmatrix} b_p \\ b_q \\ b_r \end{bmatrix} \quad (2.27)$$

O vetor $\begin{bmatrix} b_p, b_q, b_r \end{bmatrix}^T$ representa as correções, ou derivas, estimadas durante o processo de navegação.

Os Ângulos de Euler também podem ser obtidos por integração direta das seguintes equações diferenciais (FARREL; BARTH, 1998):

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.28)$$

Resumindo, as equações diferenciais para um sistema de navegação inercial, descrito no sistema LTP, são dadas por:

$$\begin{bmatrix} \dot{\mathbf{p}}^n \\ \dot{\mathbf{v}}^n \\ \dot{\mathbf{R}}_b^n \end{bmatrix} = \begin{bmatrix} \mathbf{D}^{-1} \mathbf{v}^n \\ \mathbf{R}_b^n \mathbf{f}^b - \left(2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n \right) \times \mathbf{v}^n + \mathbf{g}^n \\ \mathbf{R}_b^n \left(\boldsymbol{\Omega}_{ib}^b - \boldsymbol{\Omega}_{in}^b \right) \end{bmatrix}, \quad (2.29)$$

onde \mathbf{D}^{-1} é descrito pela Equação 2.21. Para aplicações onde o conhecimento preciso da gravidade não é necessário, é suficiente admitir sua variação em função da altitude (TITTERTON; WESTON, 2004):

$$\mathbf{g}^n = \begin{bmatrix} 0 \\ 0 \\ -g(h) \end{bmatrix} \quad (2.30)$$

$$g(h) = \frac{g(0)}{\left(1 + \frac{h}{R_0} \right)^2}, \quad (2.31)$$

onde h denota altitude, R_0 é o raio médio de curvatura da Terra, e $g(0)$, em $[m/s^2]$, é calculado em função da latitude φ , por (TITTERTON; WESTON, 2004):

$$g(0) = 9,780318 \left(1 + 5,3024 \cdot 10^{-3} \sin^2 \varphi - 5,9 \cdot 10^{-6} \sin^2 2\varphi \right) \quad (2.32)$$

As equações de navegação, dadas em 2.29, definem o posicionamento, velocidade e atitude para um veículo genérico no Sistema de Navegação (LTP). O fluxo de informações contido nessas equações pode ser observado na Figura 2.4.

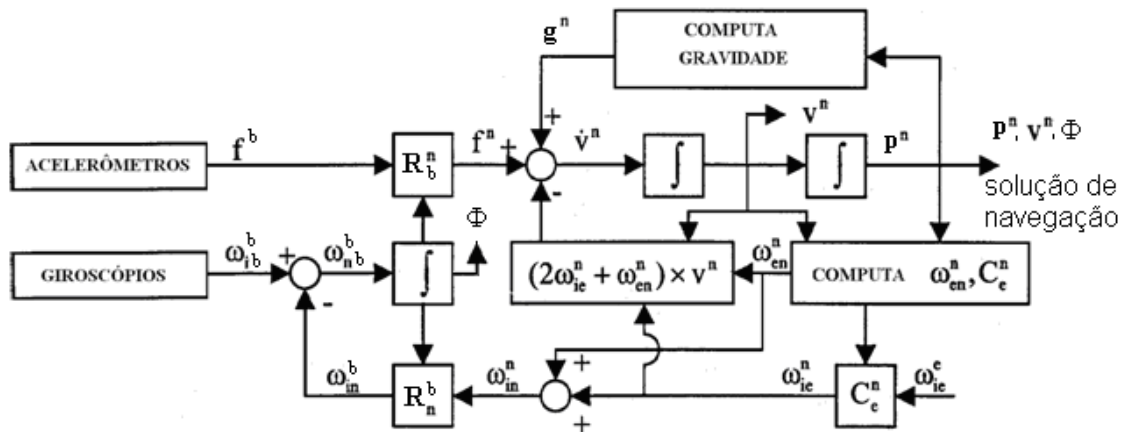


Figura 2.4 – Fluxograma do INS no referencial de navegação.

2.4 Equações simplificadas de navegação inercial para um veículo terrestre

Neste trabalho os conceitos de RNA serão testados no posicionamento de um veículo terrestre de uso urbano. Portanto sua modelagem, relativa aos sinais de entrada e saída e ao modo de operação, deverá refletir adequadamente a dinâmica do veículo em questão.

Um modelo cinemático simplificado pode ser obtido a partir das equações completas de navegação, apresentadas anteriormente, quando se incorporam algumas simplificações, ou restrições que, de acordo com Brandt e Gardner (1998) e Dissanayake et al. (2001), podem ser:

- A direção da velocidade do veículo coincide com a direção do seu eixo longitudinal (X_b ou u);

- b) Os ângulos de rolamento (*roll*) e arfagem (*pitch*), relativos à superfície Terrestre, são considerados pequenos;
- c) O veículo sempre permanece em contato com a superfície Terrestre;
- d) Os efeitos da rotação da Terra são pequenos e o vetor gravidade varia pouco, considerando que ambos os efeitos não são adequadamente mensuráveis em sensores inerciais MEMS de baixo custo.

O sistema de equações inerciais pode então ser reescrito, somente para as coordenadas NED e considerando as restrições citadas anteriormente. Da restrição descrita pelo item d:

$$\mathbf{r}^n = \begin{bmatrix} p_N & p_E & p_D \end{bmatrix}^T \quad (2.33)$$

$$\dot{\mathbf{v}}^n = \mathbf{R}_b^n \mathbf{f}^b + \mathbf{g}^n \quad (2.34)$$

$$\dot{\Phi} = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T \quad (2.35)$$

$$\boldsymbol{\omega}_{ib}^b = \begin{bmatrix} p & q & r \end{bmatrix}^T \quad (2.36)$$

$$\mathbf{C}_b^n = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \quad (2.37)$$

As equações acima podem ser reagrupadas na seguinte forma:

$$\begin{bmatrix} \dot{\mathbf{r}}^n \\ \dot{\mathbf{v}}^n \\ \dot{\Phi}_b^n \end{bmatrix} = \begin{bmatrix} \mathbf{v}^n \\ \mathbf{R}_b^n \mathbf{f}^b + \mathbf{g}^n \\ \mathbf{C}_b^n \boldsymbol{\omega}_{ib}^b \end{bmatrix} \quad (2.38)$$

Considerando as restrições descritas pelos itens a, c:

$$\dot{v}_{Y_b}^b = \dot{v}_{Z_b}^b = 0 \quad (2.39)$$

$$\dot{v}_{X_b}^b = \dot{\mathbf{v}}^f = v^f \vec{\mathbf{i}} \quad (2.40)$$

onde $\vec{\mathbf{i}}$, $\vec{\mathbf{j}}$, $\vec{\mathbf{k}}$ denotam os vetores unitários do referencial do veículo, e $\dot{\mathbf{v}}^f$ é o vetor velocidade do veículo na direção do movimento $\vec{\mathbf{i}}$. Derivando a Equação 2.40 é obtida a aceleração na direção do movimento:

$$\dot{\dot{\mathbf{v}}}^f = \dot{v}^f \vec{\mathbf{i}} + v^f \dot{\vec{\mathbf{i}}} = \dot{v}^f \vec{\mathbf{i}} + v^f \left(\boldsymbol{\omega}_{ib}^b \times \vec{\mathbf{i}} \right) \quad (2.41)$$

$$\dot{\dot{\mathbf{v}}}^f = \dot{v}^f \vec{\mathbf{i}} + v^f r \vec{\mathbf{j}} - v^f w \vec{\mathbf{k}} \quad (2.42)$$

Pré-multiplicando a Equação 2.34 por \mathbf{R}_n^b , as acelerações são reescritas no referencial do veículo:

$$\mathbf{R}_n^b \dot{\mathbf{v}}^n = \dot{\dot{\mathbf{v}}}^f = \mathbf{f}^b + \mathbf{R}_n^b \mathbf{g}^n \quad (2.43)$$

Comparando as componentes de aceleração da Equação 2.42 com 2.43, obtém-se o seguinte conjunto de equações:

$$\begin{bmatrix} \dot{\dot{v}}^f \\ v^f r \\ -v^f w \end{bmatrix} = \begin{bmatrix} f_u^b \\ f_v^b \\ f_w^b \end{bmatrix} + \mathbf{R}_n^b \mathbf{g}^n \quad (2.44)$$

Utilizando a Equação 2.2 para \mathbf{R}_n^b :

$$\dot{\dot{v}}^f = f_u^b - g \sin \theta \quad (2.45)$$

$$v^f r = f_v^b + g \sin \phi \cos \theta \quad (2.46)$$

$$\mathbf{v}^f_w = -f^b_w - g \cos\phi \cos\theta \quad (2.47)$$

A atitude $\Phi = [\phi \quad \theta \quad \psi]^T$ do veículo pode ser obtida pela integração das Equações 2.28, ou seja,

$$\dot{\phi} = p + (\sin\phi \tan\theta)q + (\cos\phi \tan\theta)r \quad (2.48)$$

$$\dot{\theta} = \cos\phi q - \sin\phi r \quad (2.49)$$

$$\dot{\psi} = \frac{\sin\phi}{\cos\theta} q + \frac{\cos\phi}{\cos\theta} r \quad (2.50)$$

O conjunto de Equações 2.45 até 2.50 descreve o modelo simplificado de navegação inercial para um veículo terrestre, considerando as restrições impostas anteriormente, e serão utilizadas na modelagem da RNA dedicada ao problema de navegação.

Com relação às equações de atitude do veículo, observa-se que em torno de $\theta = \pm 90^\circ$ existe indeterminação no rolamento do veículo (*roll*). Porém, tal situação não se aplica a um veículo urbano sob condições normais de utilização.

Considere as informações de posição e velocidade do veículo, no referencial de

navegação, $\mathbf{p}^n = [\phi \quad \lambda \quad h]^T$ e $\mathbf{v}^n = \begin{bmatrix} v_N & v_E & v_D \end{bmatrix}^T$:

$$\mathbf{v}^n = \mathbf{R}_b^n \mathbf{v}^f = \mathbf{R}_b^n \begin{bmatrix} v^f \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} v_N \\ v_E \\ v_D \end{bmatrix} \quad (2.51)$$

Utilizando a Equação 2.3 para \mathbf{R}_b^n :

$$v_N = \cos\theta \cos\psi v^f \quad (2.52)$$

$$v_E = \cos\theta \sin\psi v^f \quad (2.53)$$

$$v_D = -\text{sen}\theta v^f \quad (2.54)$$

Considerando que o veículo está em movimento, são possíveis as seguintes observações com as informações do GPS:

$$v_{\text{GPS}}^f = \sqrt{\left(v_{\text{N}}^{\text{GPS}}\right)^2 + \left(v_{\text{E}}^{\text{GPS}}\right)^2 + \left(v_{\text{D}}^{\text{GPS}}\right)^2} \quad (2.55)$$

$$\psi_{\text{GPS}} = \tan^{-1} \left(\frac{v_{\text{E}}^{\text{GPS}}}{v_{\text{N}}^{\text{GPS}}} \right) \quad (2.56)$$

$$\theta_{\text{GPS}} = \text{sen}^{-1} \left(-\frac{v_{\text{D}}^{\text{GPS}}}{v^f} \right) \quad (2.57)$$

Uma questão que surge, principalmente na estimativa de ψ_{GPS} , é a determinação de velocidade zero, ou muito pequena, com base nas informações disponíveis. Ou seja, deve-se adotar um critério que defina se o veículo está em movimento ou não.

Havendo a possibilidade de pós-processamento de dados, a detecção utilizada no chamado evento ZUPT (*Zero Velocity Update*) pode ser uma alternativa (GREJNER-BRZEZINSKA et al., 2001):

$$\left| v_{\text{N}} \right| - \bar{v}_{\text{N}}^0 \leq k \sigma_{v_{\text{N}}} \quad (2.58)$$

$$\left| v_{\text{E}} \right| - \bar{v}_{\text{E}}^0 \leq k \sigma_{v_{\text{E}}} \quad (2.59)$$

onde $\bar{v}_{\text{N,E}}^0$ denota uma média empírica tomada das estimativas de velocidade do GPS para o veículo parado, e $\sigma_{v_{\text{N,E}}}$ o correspondente desvio padrão. Em Grejner-Brzezinska et al. (2001), por exemplo, $k = 3$.

2.5 Integração dos sistemas GPS/INS

O problema da integração de dados envolve um processo de estimação, ou seja, a obtenção de valores para um parâmetro ou estado de interesse, com base em regras de decisão cujo argumento é uma seqüência de informações. O filtro de Kalman é um estimador linear recursivo que computa, de forma ótima sob diversos critérios e hipóteses, uma estimativa de variância mínima de um estado que evolui no tempo, a partir de observações linearmente relacionadas com esse estado. É um procedimento que combina saídas ruidosas de sensores para estimar o estado de um sistema com incertezas dinâmicas. Um maior detalhamento da teoria sobre o filtro de Kalman pode ser encontrado em Rios Neto e Hemerly (2007) e Gelb (1974).

No caso de um sistema de navegação os principais sensores são os giroscópios, acelerômetros, hodômetros, magnetômetros e os receptores GPS. Os estados do sistema nesse caso podem incluir posição, velocidade, aceleração e atitude de um veículo e as incertezas dinâmicas podem incluir distúrbios imprevisíveis no comportamento do veículo ou dos parâmetros dos sensores. A diferença entre as estratégias para a integração de dados entre GPS e INS num único sistema, reside no tipo de informação que é dividida entre as unidades individuais. Os tipos mais comuns de estratégia para acoplamento, encontradas na literatura, são chamados de fracamente acoplada e fortemente acoplada (Grewal et alli, 2001; Farrel e Barth, 1998).

Num sistema fracamente acoplado (*loosely coupled*), o receptor GPS possui seu próprio filtro de Kalman para processar as próprias medidas que são utilizadas para calcular posições e velocidades, que são então combinadas com as posições obtidas pelo INS para formar um vetor de resíduos. Os resíduos obtidos são filtrados por um outro filtro de Kalman, que corrige o INS e IMU de um modo realimentado, de forma que os erros de navegação causados por bias e fator de escala dos acelerômetros e giroscópios são diminuídos.

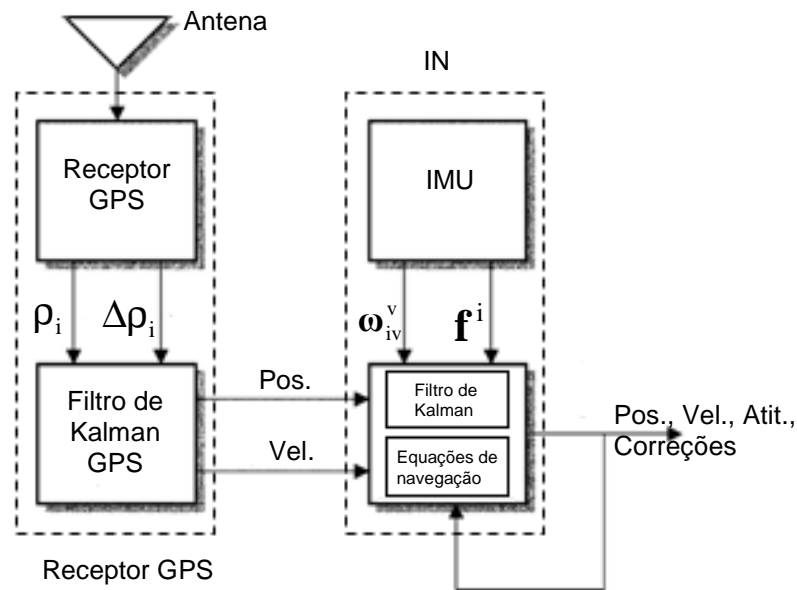


Figura 2.5 – Sistema fracamente acoplado.

Adaptado de: Grewal et alli, 2001.

Na estratégia fortemente acoplada (*tightly coupled*), os filtros de Kalman individuais do receptor GPS e INS são combinados num único filtro integrado que compartilha as observações das pseudomedidas e medidas Doppler do GPS, bem como velocidades angulares e forças específicas. Neste caso, o vetor de estado inclui os erros do INS (posição, velocidade, atitude e erros dos acelerômetros), além dos novos estados representando erros do relógio do receptor GPS.

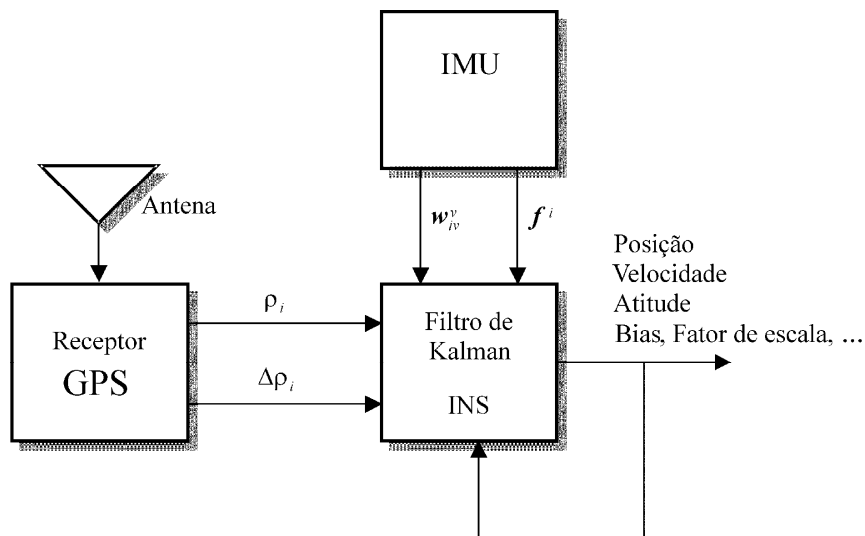


Figura 2.6 – Sistema fortemente acoplado.

Adaptado de: Grewal et alli, 2001.

3 FUNDAMENTOS DE REDE NEURAIS ARTIFICIAIS

3.1 Introdução

Redes neurais artificiais, ou simplesmente redes neurais, são sistemas paralelos e distribuídos compostos por unidades de processamento simples, chamados de neurônios artificiais, que computam determinadas funções matemáticas e possuem a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Os neurônios são dispostos em uma ou mais camadas e interligados por um grande número de conexões.

O conhecimento é adquirido por uma rede neural através de um processo de aprendizagem, ou treinamento, e é, na maioria dos modelos de RNA, armazenado em unidades chamadas de pesos sinápticos, os quais ponderam as entradas recebidas pelos neurônios da rede. A função do treinamento é modificar os pesos sinápticos, de uma forma ordenada, até que um objetivo pré-determinado seja alcançado.

Um dos benefícios das RNA é a possibilidade de modelar sistemas complexos não-lineares que envolvem o mapeamento de dados de entrada para dados de saída, sem o conhecimento anterior dos processos matemáticos que poderiam descrever o modelo em questão. Esta capacidade de aprender por meio experimental, ou por exemplos, e de generalizar seu conhecimento é a principal característica das RNA na solução de problemas. O conceito de generalização está associado à capacidade da RNA dar respostas adequadas para dados que não foram utilizados no seu treinamento.

Uma RNA pode ser caracterizada então por (PALMA NETO; NICOLETTI, 2005):

- a) Seus processadores básicos, os neurônios;
- b) Uma função de ativação, que representa o estado de um neurônio;
- c) Um padrão de conexão existente entre os neurônios;

d) Um algoritmo de treinamento, ou aprendizado.

O modelo de um neurônio básico é composto de três elementos, conforme ilustrado na Figura 3.1:

- Um conjunto de pesos sinápticos w_j , onde j se refere ao sinal de entrada;
- Um somador para os sinais de entrada ponderados pelos pesos w_j , e $w_0 = b$, que é um bias aplicado externamente e que possui o efeito de aumentar ou diminuir a entrada líquida da função de ativação;
- Uma função de ativação $f(\cdot)$ que restringe a amplitude da saída de um neurônio.

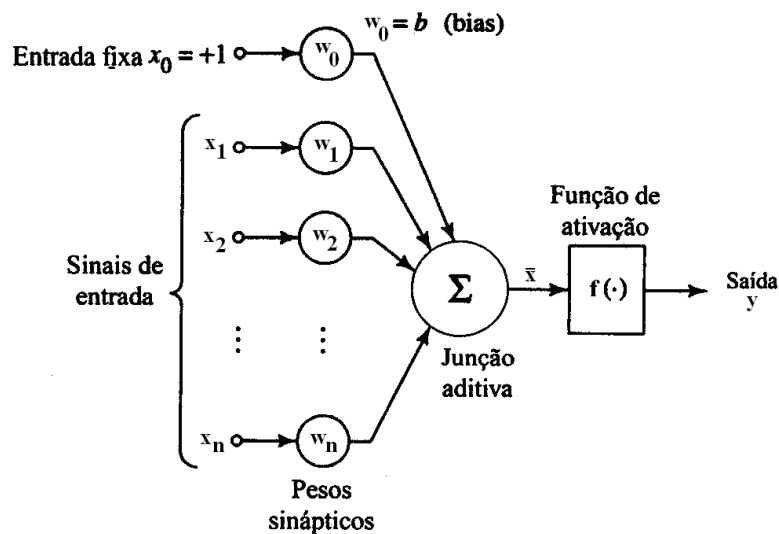


Figura 3.1 - Modelo de um neurônio.

Fonte: Adaptado de Haykin (2001).

A saída do neurônio é dada por:

$$y = f(\bar{x}) \quad (3.1)$$

onde \bar{x} , chamado de campo local induzido ou potencial de ativação, é o resultado do somador. Ou seja,

$$\bar{x} = \sum_{j=0}^n w_j x_j \quad (3.2)$$

Os três tipos básicos de função de ativação são: função de limiar (Equação 3.3); função linear por partes (Equação 3.4) e função sigmóide (Equação 3.5) e suas variações, como a função tangente hiperbólica (tanh).

$$f(\bar{x}) = \begin{cases} 1 & \text{se } \bar{x} \geq 0 \\ 0 & \text{se } \bar{x} < 0 \end{cases} \quad (3.3)$$

$$f(\bar{x}) = \begin{cases} 1 & \text{se } \bar{x} \geq 1 \\ \bar{x} & \text{se } 0 < \bar{x} < 1 \\ 0 & \text{se } \bar{x} \leq 0 \end{cases} \quad (3.4)$$

$$f(\bar{x}) = \frac{1}{1 + e^{-a\bar{x}}} \quad (3.5)$$

As funções de ativações, mostradas anteriormente e definidas de 0 a +1, também podem ser reescritas para o intervalo de -1 a +1 (caso da tanh).

O modo como o neurônio está disposto numa rede neural é chamado de arquitetura, ou estrutura, de rede. Em geral, as RNA podem ser classificadas em três classes de arquiteturas (HAYKIN, 2001):

- a) Redes alimentadas adiante com camada única: Os neurônios são dispostos na forma de camadas e na sua forma mais simples é constituída de uma camada de entrada externa de nós de fonte, que faz interface com o ambiente, e que se projeta sobre uma camada de saída de neurônios, as unidades de processamento, num caminho de direção única.
- b) Redes alimentadas adiante com múltiplas camadas: diferencia-se da classe anterior pela presença de uma ou mais camadas ocultas, cujos neurônios são chamados de ocultos. A função dos neurônios ocultos é estabelecer uma relação útil entre a camada de entrada e a de saída. A Figura 3.2

mostra as redes alimentadas adiante, ou *feedforward*, com e sem camadas ocultas.

- c) Redes recorrentes: são distintas das classes anteriores por ter pelo menos um laço de realimentação.

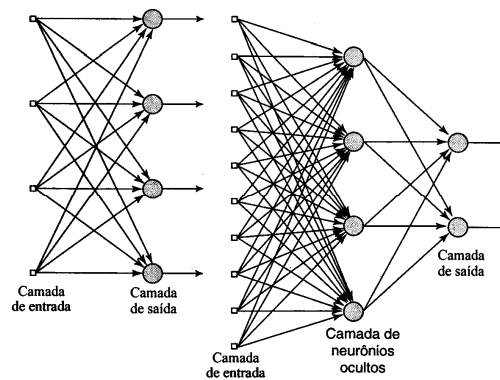


Figura 3.2 - Redes alimentadas adiante (*feedforward*)

Fonte: Adaptado de Haykin (2001).

As redes alimentadas adiante, chamadas de perceptrons de múltiplas camadas (ou *multilayer perceptron*, MLP) têm sido utilizadas com sucesso para resolver diversos problemas difíceis, através do seu treinamento de forma supervisionada, com um algoritmo de retropropagação de erros (*error backpropagation*), e serão aplicadas no presente trabalho com um algoritmo de treinamento baseado no filtro de Kalman.

3.2 Redes perceptron de múltiplas camadas (MLP)

As redes MLP apresentam a configuração mínima de três camadas, entrada, oculta e saída, com, ao menos, um neurônio por camada. Na camada de entrada, os neurônios representam as variáveis de entradas independentes, e que se propagam para a próxima camada (oculta). Na camada oculta os neurônios realizam o processamento dos sinais, através das funções de ativação e das chamadas regras de propagação. Os sinais processados são distribuídos para a camada de saída, cujos neurônios representam as respostas da rede.

Normalmente a camada oculta contém funções de ativação não lineares (sigmoidais ou *tanh*) e suas saídas são combinadas linearmente na camada de saída. Uma propriedade importante das redes tipo MLP é que uma rede contendo apenas uma camada escondida, com funções de ativação sigmoidais, é capaz de aproximar qualquer função contínua arbitrária. Entretanto, o chamado Teorema da Aproximação Universal, não diz se a camada oculta é ótima no sentido de tempo de aprendizagem, facilidade de implementação, ou generalização (HAYKIN, 2001). A Figura 3.3 ilustra uma rede MLP de três camadas.

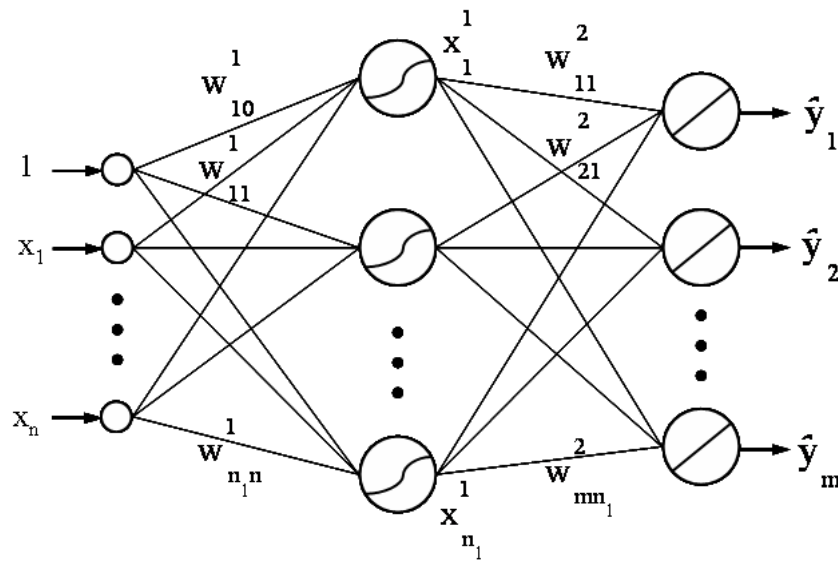


Figura 3.3 – Rede MLP com uma camada oculta.

Na Figura 3.3, x_j denota um sinal de entrada, w_{ij}^k é o peso, da camada k , que conecta o sinal j ao neurônio i , e x_i^k representa as saídas da mesma camada k . A saída $\hat{y}_i(t)$, da rede, onde t representa um instante discreto, é dada por:

$$\hat{y}_i(t) = \sum_{j=1}^{n_1} w_{ij}^2 \left[f \left(\sum_{j=1}^n w_{ij}^1 x_j(t) + w_{i0}^1 \right) \right] \quad (3.6)$$

As saídas x_i^k de uma camada k qualquer, podem ser representadas por (HAYKIN, 2001; CARRARA, 1997):

$$\mathbf{x}_i^k = f^k \left(\sum_{j=1}^{n_{k-1}} w_{ij}^k x_j^{k-1} + b_i^k \right), \quad (3.7)$$

que na forma matricial podem ser reescritas da seguinte forma:

$$\mathbf{x}^k = f^k \left(\mathbf{W}^k \mathbf{x}^{k-1} \right) \quad (3.8)$$

$$\mathbf{x}^k = \left[f^k \left(\frac{\mathbf{x}_1^k}{1} \right) \quad \dots \quad f^k \left(\frac{\mathbf{x}_{n_k}^k}{1} \right) \quad 1 \right]^T \quad (3.9)$$

$$\mathbf{W}^k = \begin{bmatrix} w_{11}^k & \dots & w_{1n_{k-1}}^k & b_1^k \\ w_{21}^k & \dots & w_{2n_{k-1}}^k & b_2^k \\ \vdots & & \vdots & \vdots \\ w_{n_k 1}^k & \dots & w_{n_k n_{k-1}}^k & b_{n_k}^k \end{bmatrix} \quad (3.10)$$

Desta forma uma rede MLP de l camadas, com n_k neurônios em cada camada, possui a seguinte saída:

$$\mathbf{x}^l = f^l \left(\mathbf{W}^l f^{l-1} \left(\mathbf{W}^{l-1} \dots f^1 \left(\mathbf{W}^1 \mathbf{x}^0 \right) \dots \right) \right) \quad (3.11)$$

3.3 Treinamento de redes MLP

Não há regras de aprendizagem únicas no projeto de redes neurais. Uma regra básica é a aprendizagem por correção de erro, que é caracterizada como sendo supervisionada, pois um sinal de saída-alvo ou desejado é conhecido e utilizado para computar um erro. Por definição (HAYKIN, 2001):

$$e_i(t) = d_i(t) - y_i(t) \quad (3.12)$$

O sinal de saída do neurônio i , no instante de tempo discreto t , é dado por $y_i(t)$; $d_i(t)$ é a saída desejada correspondente e $e_i(t)$ o erro gerado. O erro é utilizado para ajustes corretivos nos pesos sinápticos w_{ij} , de forma iterativa, fazendo com que $y_i(t)$ aproxime-se de $d_i(t)$.

O processo usual de ajustes é conseguido minimizando-se uma função de custo, definida em função do erro na saída da rede, através do uso de diferentes algoritmos de treinamento. Uma possibilidade é a estimação estocástica (RIOS NETO, 1997) na qual uma função de custo é definida pela ponderação entre uma informação a priori, dos parâmetros da rede a serem estimados, e o erro na saída da rede:

$$J(\mathbf{w}) = \frac{1}{2} \left[(\mathbf{w} - \bar{\mathbf{w}})^T \bar{\mathbf{P}}^{-1} (\mathbf{w} - \bar{\mathbf{w}}) + \sum_{p=1}^L [\mathbf{d}(p) - \mathbf{y}(p)]^T \mathbf{R}^{-1}(p) [\mathbf{d}(p) - \mathbf{y}(p)] \right] \quad (3.13)$$

onde são conhecidos os $p=1, 2, \dots, L$ pares de treinamento apresentados em lote, uma estimativa *a priori* do vetor de pesos $\bar{\mathbf{w}}$, e as matrizes de ponderação $\bar{\mathbf{P}}^{-1}$ e $\mathbf{R}^{-1}(p)$.

3.4 Algoritmo de retropropagação

O termo retropropagação está, normalmente, associado com a propagação recursiva dos erros, ou seja, com a avaliação dos erros nas camadas intermediárias da RNA. O algoritmo consiste na obtenção da variação apresentada na saída de um determinado neurônio, em relação a alguma alteração sofrida pelos pesos da rede, através da aplicação da regra da cadeia da derivada parcial da saída deste neurônio, em relação a um determinado peso de uma camada anterior (CARRARA, 1997; HAYKIN, 2001).

Consideram-se as derivadas das saídas x_i^k , em relação aos pesos w_{ij}^k , na camada k .

Quando não existe conexão entre os pesos e as saídas, o resultado é nulo:

$$\frac{\partial x_i^k}{\partial w_{nj}^k} = 0, \text{ se } n \neq m \quad (3.14)$$

Quando $n = m$, o peso w_{nj}^k está conectado com a saída x_m^k , então

$$\frac{\partial x_i^k}{\partial w_{ij}^k} = f^{k'} \left(\bar{x}_i^k \right) x_j^{k-1}, \quad (3.15)$$

que na forma vetorial pode ser reescrita como:

$$\frac{\partial \mathbf{x}_i^k}{\partial \mathbf{w}_{ij}^k} = \mathbf{F}^k \begin{bmatrix} 0 \\ \vdots \\ x_j^{k-1} \\ \vdots \\ 0 \end{bmatrix}, \quad (3.16)$$

onde \mathbf{F}^k denota a matriz diagonal das derivadas $f^{k'} \left(\bar{\mathbf{x}}^k \right)$.

$$\mathbf{F}^k = \frac{d\mathbf{f}^k \left(\bar{\mathbf{x}}^k \right)}{d\bar{\mathbf{x}}^k} = \begin{bmatrix} f^{k'} \left(\bar{x}_1^k \right) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & f^{k'} \left(\bar{x}_{n_k}^k \right) \end{bmatrix} \quad (3.17)$$

A derivada parcial da saída x_i^k , $i = m$, em relação aos pesos w_{ij}^{k-1} , da camada anterior,

$k-1$, é dada por:

$$\frac{\partial x_m^k}{\partial w_{ij}^{k-1}} = f^{k'} \left(\bar{x}_m^k \right) \sum_{s=1}^{n_{k-1}} w_{ms}^k \frac{\partial x_s^{k-1}}{\partial w_{ij}^{k-1}}, \quad (3.18)$$

onde a última parcela da Equação 3.18, a derivada parcial de x_s^{k-1} , pode ser obtida da mesma forma descrita anteriormente pela Equação 3.15. Desta forma,

$$\frac{\partial x_m^k}{\partial w_{ij}^{k-1}} = f^{k'} \left(\bar{x}_m^k \right) w_{mi}^k f^{k-1'} \left(\bar{x}_i^{k-1} \right) x_j^{k-2} = \mathbf{F}^k \mathbf{W}^k \mathbf{F}^{k-1} \begin{bmatrix} 0 \\ \vdots \\ x_j^{k-2} \\ \vdots \\ 0 \end{bmatrix} \quad (3.19)$$

São determinadas, então, as derivadas em relação aos pesos da camada k e, em seguida, as derivadas em relação à camada $k-1$. Ou seja, em relação à rede neural, a computação das derivadas tem início na saída da rede e, de forma recursiva, a retro-propagação é feita na direção das camadas iniciais. Então, de forma recorrente, se a rede neural possui ℓ camadas:

$$\frac{\partial x_m^\ell}{\partial w_{ij}^k} = \delta_{m,i}^k x_j^{k-1}, \quad (3.20)$$

onde os escalares $\delta_{m,i}^k$ são computados, a partir de:

$$\delta_{m,i}^k = f^{k'} \left(\bar{x}_i^k \right) \sum_{s=1}^{n_{k+1}} \delta_{m,s}^{k+1} w_{si}^{k+1}, \quad (3.21)$$

com a seguinte condição inicial:

$$\delta_{m,i}^\ell = \begin{cases} f^{\ell'} \left(\bar{x}_i^\ell \right), & \text{se } i = m. \\ 0, & \text{se } i \neq m. \end{cases} \quad (3.22)$$

As equações obtidas podem ser reescritas na forma vetorial, e computadas de forma recorrente. Então

$$\frac{\partial x^{\ell}}{\partial w_{ij}^k} = \Delta^k \begin{bmatrix} 0 \\ \vdots \\ x_{k-1}^j \\ \vdots \\ 0 \end{bmatrix} \quad (3.23)$$

$$\Delta^k = \mathbf{F}^{\ell} \mathbf{W}^{\ell} \mathbf{F}^{\ell-1} \mathbf{W}^{\ell-1} \dots \mathbf{F}^{k+1} \mathbf{W}^{k+1} \mathbf{F}^k = \Delta^{k+1} \mathbf{W}^{k+1} \mathbf{F}^k \quad (3.24)$$

O valor inicial de Δ^k , para a camada de saída ℓ , é dado por:

$$\Delta^{\ell} = \mathbf{F}^{\ell} \quad (3.25)$$

O algoritmo da retro-propagação será utilizado na computação de derivadas parciais, da saída de rede neural em relação aos seus pesos, requeridas durante seu treinamento via filtro de Kalman, a ser empregado nos próximos capítulos.

4 ABORDAGEM COM O FILTRO DE KALMAN

4.1 Introdução

O treinamento de uma RNA, como já mencionado anteriormente, consiste na estimação dos seus parâmetros, pesos sinápticos e biases, através de algoritmos numéricos que normalmente minimizam uma função de custo baseada no erro apresentado na saída da rede. Os muitos algoritmos disponíveis para treinamento fazem uso da retropropagação de erros para a estimação dos parâmetros, destacando-se os algoritmos do gradiente e suas derivações (RUMELHART et al., 1986), do gradiente conjugado, de Levenberg-Marquardt, entre outros. Cada algoritmo possui características próprias e são mais ou menos adequados a diferentes classes de problemas (HAYKIN, 2001).

A técnica do filtro de Kalman para o ajuste dos parâmetros de uma RNA também é uma alternativa de treinamento. A técnica considera os parâmetros ótimos da RNA como sendo os estados do sistema a serem estimados, e a saída da rede a medida associada.

4.2 Solução com o Filtro estendido de Kalman

Chamado de GEKF (*Global Extended Kalman Filter*) na sua forma original, o método considera que uma RNA ideal pode ser descrita pelo seguinte sistema discreto não-linear, caracterizado como sendo um processo estacionário corrompido por ruído:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \boldsymbol{\omega}(t) \quad (4.1)$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{w}(t), \mathbf{x}(t)) + \mathbf{v}(t) \quad (4.2)$$

Aqui $\mathbf{w}(t)$ denota o vetor de estado formado pelos parâmetros da RNA, no instante discreto t ou índice denotando a ordem de apresentação do padrão de treinamento, e $\mathbf{y}(t)$ é a equação de observação que representa a saída desejada da RNA como função do vetor de entrada $\mathbf{x}(t)$ e do vetor de parâmetros $\mathbf{w}(t)$. Os termos $\boldsymbol{\omega}(t)$ e $\mathbf{v}(t)$ são, respectivamente, os ruídos de processo e os ruídos de medida, e são caracterizados como sendo ruído branco com média zero e covariâncias dadas por:

$$E[\mathbf{v}(t)\mathbf{v}(t)^T] = \delta_{p,q} \mathbf{R}(t), \text{ e}$$

$$E[\boldsymbol{\omega}(t)\boldsymbol{\omega}(t)^T] = \delta_{p,q} \mathbf{Q}(t), \quad (4.3)$$

onde $\delta_{p,q}$ representa a função delta de Kronecker (vale 1 para $p = q$ e 0 para $p \neq q$).

A RNA pode ser representada pelo seguinte mapeamento não linear:

$$\hat{\mathbf{y}}(t) = \hat{\mathbf{f}}(\mathbf{x}(t), \mathbf{w}(t)), \quad (4.4)$$

onde $\hat{\mathbf{y}}(t)$ denota o vetor de saídas da RNA, segundo a aproximação dada pela RNA, $\mathbf{x}(t)$ seu vetor de entradas, e $\mathbf{w}(t)$ o seu vetor de parâmetros livres.

$$\hat{\mathbf{y}}(t) = [\hat{y}_1(t) \quad \hat{y}_2(t) \quad \cdots \quad \hat{y}_m(t)]^T \quad (4.5)$$

$$\mathbf{x}(t) = [x_1(t) \quad x_2(t) \quad \cdots \quad x_n(t)]^T \quad (4.6)$$

A dimensão de \mathbf{w} é dada por n_w e depende do número de neurônios n_k , em cada uma das ℓ camadas da RNA:

$$n_w = \sum_{k=1}^{\ell} n_k (n_k - 1) \quad (4.7)$$

O treinamento da RNA é feito, geralmente por aprendizado supervisionado, a partir de um conjunto com L pares de treinamento, podendo ser apresentados sequencialmente a cada instante discreto t (RIOS NETO, 1997), ou seja:

$$\{(\mathbf{x}(t), \mathbf{y}(t)) : \mathbf{y}(t) = \mathbf{f}(\mathbf{x}(t)), \quad t = 1, 2, \dots, L\} \quad (4.8)$$

A cada iteração i , também chamada de época, todos L pares são apresentados, e o processo de treinamento busca minimizar o seguinte funcional:

$$\mathbf{J}(\mathbf{w}) = \frac{1}{2} \left[(\mathbf{w} - \bar{\mathbf{w}})^T \bar{\mathbf{P}}^{-1} (\mathbf{w} - \bar{\mathbf{w}}) + \sum_{t=1}^L \left((\mathbf{y}(t) - \hat{\mathbf{f}}(\mathbf{x}(t), \mathbf{w}))^T \mathbf{R}^{-1} (\mathbf{y}(t) - \hat{\mathbf{f}}(\mathbf{x}(t), \mathbf{w})) \right) \right] \quad (4.9)$$

A Equação 4.4 pode ser expandida em primeira ordem, em torno da estimativa *a priori* do vetor de pesos $\bar{\mathbf{w}}$, resultado em:

$$\mathbf{y}(t) = \bar{\mathbf{y}}(t, \bar{\mathbf{w}}) + \hat{f}_{\mathbf{w}}(\mathbf{x}(t), \bar{\mathbf{w}})(\mathbf{w}(t) - \bar{\mathbf{w}}) \quad (4.10)$$

$$\bar{\mathbf{w}}(1) = \bar{\mathbf{w}} \quad (4.11)$$

$$\hat{f}_{\mathbf{w}}(\mathbf{x}(t), \bar{\mathbf{w}}) = \left. \frac{\partial \hat{f}(\mathbf{x}(t), \mathbf{w})}{\partial \mathbf{w}} \right|_{\mathbf{w}=\bar{\mathbf{w}}} = \mathbf{H}(t, \bar{\mathbf{w}}) \quad (4.12)$$

A matriz \mathbf{H} possui dimensão (m, n_w) . A linearidade da Equação 4.10 é garantida adotando-se uma perturbação linear, através do parâmetro ajustável α , do seguinte modo (RIOS NETO, 1997):

$$\alpha(t)[\mathbf{y}(t) - \bar{\mathbf{y}}(t, \bar{\mathbf{w}})] + \mathbf{H}(t, \bar{\mathbf{w}})\bar{\mathbf{w}} = \mathbf{H}(t, \bar{\mathbf{w}})\mathbf{w}(t) \quad (4.13)$$

$$0 < \alpha(t) \leq 1 \quad (4.14)$$

$$\mathbf{z}(t) = \mathbf{H}(t, \bar{\mathbf{w}})\mathbf{w}(t) \quad (4.15)$$

$$\mathbf{z}(t) = \alpha(t)[\mathbf{y}(t) - \bar{\mathbf{y}}(t, \bar{\mathbf{w}})] + \mathbf{H}(t, \bar{\mathbf{w}})\bar{\mathbf{w}} \quad (4.16)$$

A solução para minimizar o funcional da Equação 4.9 se dá, na forma de um filtro de Kalman, considerando o seguinte problema de estimação linear estocástica (RIOS NETO, 1997; RIOS NETO e HEMERLY, 2007):

$$\bar{\mathbf{w}}(t) = \mathbf{w}(t) + \bar{\mathbf{e}} \quad (4.17)$$

$$\mathbf{z}(t) = \mathbf{H}(t)\mathbf{w}(t) + \mathbf{v}(t) \quad (4.18)$$

São, também, consideradas válidas as seguintes hipóteses:

$$E[\mathbf{v}(t)] = 0, \quad E[\mathbf{v}(t)\mathbf{v}(1)^T] = \delta_{t,1} \mathbf{R}(t) \quad (4.19)$$

$$E[\bar{\mathbf{e}}] = 0, \quad E[\bar{\mathbf{e}}\bar{\mathbf{e}}^T] = \bar{\mathbf{P}}(t), \quad (4.20)$$

onde \mathbf{R} é a matriz de covariâncias dos ruídos da medida, diagonal com dimensão (m, m) e $\bar{\mathbf{P}}$ é a matriz de covariâncias dos erros de estimação a priori, com dimensão (n_w, n_w) .

Novamente, $\delta_{p,q}$ representa a função delta de Kronecker (vale 1 para $p=q$ e 0 para $p \neq q$).

A solução, via filtro de Kalman, é dada a seguir, juntamente com as condições iniciais $\bar{\mathbf{w}}(1) = \bar{\mathbf{w}}$, $\bar{\mathbf{P}}(1) = \bar{\mathbf{P}}$, (RIOS NETO, 1997; RIOS NETO e HEMERLY, 2007).

Para $t = 1, \dots, L$:

a) Linearizar todas as medições:

$$\mathbf{z}(t) = \alpha(t) [\mathbf{y}(t) - \bar{\mathbf{y}}(t, \bar{\mathbf{w}})] + \mathbf{H}(t, \bar{\mathbf{w}}) \bar{\mathbf{w}} \quad (4.21)$$

b) Fase de atualização :

$$\mathbf{K}(t) = \bar{\mathbf{P}}(t) \mathbf{H}^T(t, \bar{\mathbf{w}}) [\mathbf{R}(t) + \mathbf{H}(t, \bar{\mathbf{w}}) \bar{\mathbf{P}}(t) \mathbf{H}^T(t, \bar{\mathbf{w}})]^{-1} \quad (4.22)$$

$$\hat{\mathbf{w}}(t) = \bar{\mathbf{w}}(t) + \mathbf{K}(t) [\mathbf{z}(t) - \mathbf{H}(t, \bar{\mathbf{w}}) \bar{\mathbf{w}}(t)] \quad (4.23)$$

$$\mathbf{P}(t) = \bar{\mathbf{P}}(t) - \mathbf{K}(t) \mathbf{H}(t, \bar{\mathbf{w}}) \bar{\mathbf{P}}(t) \quad (4.24)$$

c) Fase de previsão (propagação):

$$\bar{\mathbf{P}}(t+1) = \mathbf{P}(t) + \mathbf{Q}(t) \quad (4.25)$$

$$\bar{\mathbf{w}}(t+1) = \hat{\mathbf{w}}(t) \quad (4.26)$$

Ao final de cada iteração, $t = L$, checar um critério de parada de treinamento. Se o critério for satisfeito, encerrar o treinamento e considerar:

$$\bar{\mathbf{w}} = \hat{\mathbf{w}}(L) \quad (4.27)$$

$$\bar{\mathbf{P}} = \hat{\mathbf{P}}(L) \quad (4.28)$$

Caso contrário, se o critério não for satisfeito, continuar o treinamento numa nova iteração, e relinearizar as medições em torno de $\bar{\mathbf{w}}_{\text{nov}} = \hat{\mathbf{w}}(L)$ considerando, na nova iteração:

$$\bar{\mathbf{w}}(1) = \hat{\mathbf{w}}(L) \quad (4.29)$$

$$\bar{\mathbf{P}}(1) = \bar{\mathbf{P}} \quad (4.30)$$

Nas equações anteriores, do filtro estendido de Kalman, \mathbf{K} é o chamado ganho de Kalman, com dimensão (n_w, m) ; e \mathbf{Q} é a matriz de covariâncias dos ruídos do processo, com dimensão (n_w, n_w) .

4.3 Solução adaptativa: estimação do ruído de estado

Quando, numa situação de aplicação prática, o filtro de Kalman processa um conjunto de treinamento com muitos pares, seu desempenho pode ser afetado se ocorre o fenômeno da divergência das estimativas, em consequência de erros cometidos na modelagem do processo, e/ou de erros devidos ao mau condicionamento numérico dos procedimentos envolvidos.

Nessa situação, a matriz de covariâncias calculada, \mathbf{P} , não representa adequadamente a dispersão dos erros presentes na estimativa do estado, podendo ocorrer que os valores das variâncias se tornem irrealisticamente pequenos, fazendo com que o ganho \mathbf{K} pondere mais as estimativas anteriores do que as novas informações (RIOS NETO e KUGA, 1985). Ou seja, o conjunto de parâmetros \mathbf{w} , que representa o conhecimento adquirido pela RNA, não se modifica na presença de um novo par de treinamento não redundante. A margem de incerteza propiciada pelo ruído, através da matriz de covariâncias dos erros da estimativa propagada, é que permite sensibilidade para continuar tirando informação das novas observações (RIOS NETO e HEMERLY, 2007).

Um possível modo de evitar essa situação é dosar a quantidade de ruído da matriz \mathbf{Q} , através de um procedimento adaptativo baseado num critério de consistência estatística, de forma que novas informações, trazidas por novos pares de treinamento, possam ser ponderadas com as informações já obtidas e armazenadas pelo conjunto de parâmetros \mathbf{w} . O critério utilizado é dado por Rios Neto (1997), e considera a modelagem dada pelas Equações 4.1 e 4.2:

$$\beta E[v_j^2(t)] = \mathbf{H}_j(t, \bar{\mathbf{w}}) [\mathbf{P}(t) + \mathbf{Q}(t)] \mathbf{H}_j^T(t, \bar{\mathbf{w}}) \quad (4.31)$$

onde $j=1, \dots, m$ e β é um parâmetro ajustável empiricamente com valor próximo de 1. A função de β é manter uma consistência estatística entre a ordem de grandeza dos ruídos de medida e dos erros de estimativa dos estados e propiciar que o aprendizado seja distribuído entre os L pares do conjunto de treinamento. Com $\beta = 1$, as informações trazidas por novos pares de treinamento possuem a mesma importância daquelas já armazenadas. Valores de β ligeiramente menores que 1 indicam que as informações já armazenadas possuem mais importância que as novas informações processadas.

A Equação anterior pode ser expandida na seguinte forma:

$$\beta R_j(t) - \mathbf{H}_j(t, \bar{\mathbf{w}}) \mathbf{P}(t) \mathbf{H}_j^T(t, \bar{\mathbf{w}}) = \mathbf{H}_j(t, \bar{\mathbf{w}}) \mathbf{Q}(t) \mathbf{H}_j^T(t, \bar{\mathbf{w}}) \quad (4.32)$$

Define-se, então, a seguinte observação associada:

$$z_j^q(t+1, \beta) = \beta R_j(t) - \mathbf{H}_j(t, \bar{\mathbf{w}}) \mathbf{P}(t) \mathbf{H}_j^T(t, \bar{\mathbf{w}}) \quad (4.33)$$

$$z_j^q(t+1, \beta) = \mathbf{H}^q(t+1) \mathbf{q}(t) \quad (4.34)$$

onde:

$$\mathbf{q}(t) = \begin{bmatrix} q_1(t) & q_2(t) & \dots & q_{n_w}(t) \end{bmatrix}^T \quad (4.35)$$

$$\mathbf{H}^q(t+1) \mathbf{q}(t) = \mathbf{H}_j(t, \bar{\mathbf{w}}) \mathbf{Q}(t) \mathbf{H}_j^T(t, \bar{\mathbf{w}}) = \sum_{k=1}^{n_w} H_{jk}^2(t, \bar{\mathbf{w}}) q_k(t) \quad (4.36)$$

$$\mathbf{Q}(t) = \text{diag}[\mathbf{q}(t)] \quad (4.37)$$

O problema da estimação associada de $\mathbf{q}(t)$ pode então ser definido como (RIOS NETO, 1997):

$$0 = \mathbf{q}(t) + \bar{\mathbf{e}}^q(t) \quad (4.38)$$

$$z_j^q(t+1, \beta) = \mathbf{H}^q(t+1) \mathbf{q}(t) + \mathbf{v}^q(t+1) \quad (4.39)$$

$$E[\bar{\mathbf{e}}^q] = 0, E[\bar{\mathbf{e}}^q \bar{\mathbf{e}}^{qT}] = \mathbf{I}_{n_w} \quad (4.40)$$

$$E[\mathbf{v}^q(t+1)] = 0, E[\mathbf{v}^q(t+1) \mathbf{v}^{qT}(t+1)] = \mathbf{R}^q(t+1) = 0 \quad (4.41)$$

onde \mathbf{I}_{n_w} denota uma matriz identidade de dimensão n_w .

Trata-se de um problema de estimação com observações exatas, as quais podem ser processadas por um filtro de Kalman particularizado à condição limite de $\mathbf{R}^q = 0$ (FREITAS PINTO e RIOS NETO, 1990). O algoritmo seqüencial é apresentado a seguir:

a) Condições iniciais:

$$\bar{\mathbf{q}} = 0, \bar{\mathbf{P}}^q = \mathbf{I}_{n_w} \quad (4.42)$$

b) Para $j = 1$ até m , faça:

$$z_j^q = \beta R_j(t) - \mathbf{H}_j(t, \bar{\mathbf{w}}) \mathbf{P}(t) \mathbf{H}_j^T(t, \bar{\mathbf{w}}) \quad (4.43)$$

$$\mathbf{H}_j^q = \sum_{k=1}^{n_w} \mathbf{H}_{jk}^2(t, \bar{\mathbf{w}}) \quad (4.44)$$

$$\mathbf{K}_j^q = \bar{\mathbf{P}}^q \mathbf{H}_j^{qT} [\mathbf{H}_j^q \bar{\mathbf{P}}^q \mathbf{H}_j^{qT}]^{-1} \quad (4.45)$$

$$\hat{\mathbf{q}} = \bar{\mathbf{q}} + \mathbf{K}_j^q [z_j^q - \mathbf{H}_j^q \bar{\mathbf{q}}] \quad (4.46)$$

$$\mathbf{P}^q = \bar{\mathbf{P}}^q - \mathbf{K}_j^q \mathbf{H}_j^q \bar{\mathbf{P}}^q \quad (4.47)$$

c) Para $j = m$:

$$\bar{\mathbf{q}} = \hat{\mathbf{q}}, \mathbf{e}$$

$$\bar{\mathbf{P}}^q = \mathbf{P}^q \quad (4.48)$$

d) Ao final do processamento das m medidas, e para $k = 1, \dots, n_w$:

$$q_k(t) = \begin{cases} 0 & \text{se } \hat{q}_k < 0 \\ \hat{q}_k & \text{se } \hat{q}_k \geq 0 \end{cases} \quad (4.49)$$

$$Q(t) = \text{diag}[q(t)] \quad (4.50)$$

4.4 Pré-filtragem de dados de treinamento

Um dos fatores que podem determinar o desempenho adequado da RNA é a caracterização adequada da cinemática do veículo pelos sensores inerciais e pelas informações do GPS. Por esta razão, os sinais de entrada e as saídas desejadas devem formar um conjunto de treinamento adequado para o treinamento da RNA. Uma das características dos sensores inerciais MEMS é seu nível de ruído, devendo-se, então, implementar um esquema que formate os sinais da saída da IMU, juntamente com as informações correspondentes de um receptor GPS, que também possuem um determinado nível de ruído. Técnicas de pré-filtragem são utilizadas para ajustar tanto a frequência com que os sinais da IMU são disponibilizados, como uma tentativa de diminuição do nível de ruído (GELB, 1974).

Considera-se a situação onde medições, ou observações, estão disponíveis numa frequência maior do que a necessária. Se \mathbf{x} denota o vetor de observações, de dimensão $n \times 1$, a ser medido pela IMU, então:

$$\mathbf{x} = \{\boldsymbol{\omega}_{ib}^b, \mathbf{f}^b\}^T \quad (4.51)$$

onde $\boldsymbol{\omega}_{ib}^b$ representa o vetor de velocidades angulares e \mathbf{f}^b o vetor de forças específicas.

O modelo discreto do estado e correspondentes observações podem ser caracterizados como:

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \boldsymbol{\omega}_k \quad (4.52)$$

$$\mathbf{y}_k = \mathbf{x}_k + \mathbf{v}_k \quad (4.53)$$

Ou seja, assumindo que a dinâmica do processo possa ser definida como estacionária, a matriz de transição de estados, e a matriz que relaciona estados com observações, são dadas por:

$$\Phi_{k,k-1} = \mathbf{I}_n \quad (4.54)$$

$$\mathbf{H}_k = \mathbf{I}_n \quad (4.55)$$

Aqui, \mathbf{I}_n denota a matriz identidade de ordem n , ω_k modela o ruído do processo, \mathbf{v}_k modela o ruído da medida. Os erros são assumidos independentes e caracterizados por:

$$\omega_k = N(\mathbf{0}, \mathbf{Q}_k) \quad (4.56)$$

$$\mathbf{v}_k = N(\mathbf{0}, \mathbf{R}_k) \quad (4.57)$$

onde \mathbf{Q}_k é a matriz de covariância do ruído dinâmico e \mathbf{R}_k é a matriz de covariância dos erros de observação. A matriz \mathbf{R}_k poder ser obtida a partir dos dados caracterizados estatisticamente em processos de calibração:

$$\mathbf{R}_k = \text{diag}\left(\sigma_{\omega_\phi}^2 \quad \sigma_{\omega_\theta}^2 \quad \sigma_{\omega_\psi}^2 \quad \sigma_{a_x}^2 \quad \sigma_{a_y}^2 \quad \sigma_{a_z}^2\right) \quad (4.58)$$

Já os elementos da matriz \mathbf{Q}_k podem ter a dosagem da incerteza propiciada pelo ruído, feita de forma heurística, envolvendo algum esforço de calibração que é sempre dependente do problema (RIOS NETO e HEMERLY, 2007).

Assumindo que o intervalo de tempo δt_k , entre duas observações, é suficientemente pequeno, de forma que:

$$\delta t_k = t_k - t_{k-1} \rightarrow 0, \quad (4.59)$$

então, considerando o modelo do sistema descrito pelas Equações 4.41 e 4.42, e de acordo com Gelb (1974), a matriz \mathbf{Q}_k pode ser aproximada para:

$$\mathbf{Q}_k \cong \mathbf{Q} \cdot \delta t_k \quad (4.60)$$

onde \mathbf{Q} é a matriz de densidade espectral e considerada constate durante o intervalo δt_k . Seus termos da diagonal principal podem ser aproximados pela Equação 4.61, onde α_j é um parâmetro empírico, cujo ajuste depende do tipo da IMU e das aplicações típicas do veículo em questão, que definem sua envoltória de desempenho.

$$q_{j,j} = \alpha_j \cdot (\bar{x}_{k,j})^2 / \delta t_k, \text{ e } j=1, \dots, n \quad (4.61)$$

Conhecidas as condições iniciais,

$$\hat{\mathbf{x}}_0 = \mathbf{x}_0 \quad (4.62)$$

$$\hat{\mathbf{P}}_0 = \mathbf{P}_0, \quad (4.63)$$

pode-se, então, processar as medidas em intervalos δt_k utilizando as equações do filtro de Kalman, para o modelo descrito anteriormente, e que são descritas a seguir:

$$\bar{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} \quad (4.64)$$

$$\bar{\mathbf{P}}_k = \hat{\mathbf{P}}_{k-1} + \mathbf{Q}_k \quad (4.65)$$

$$\mathbf{K}_k = \bar{\mathbf{P}}_k (\bar{\mathbf{P}}_k + \mathbf{R}_k)^{-1} \quad (4.66)$$

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k) \bar{\mathbf{P}}_k \quad (4.67)$$

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{y}_k - \bar{\mathbf{x}}_k) \quad (4.68)$$

Se as medições provenientes da IMU são necessárias somente a cada intervalo Δt , então podem ser computadas a cada r processamentos do filtro de Kalman, descrito anteriormente, onde r é definido por:

$$r \approx \frac{\Delta t}{\delta t}, \text{ para } r > 1 \text{ e inteiro} \quad (4.69)$$

As informações requeridas podem, então, ser extraídas, ou seja:

$$\bar{\mathbf{x}}_m = \hat{\mathbf{x}}_k \quad (4.70)$$

$$t_m = t_k \quad (4.71)$$

$$\bar{\mathbf{P}}_{e,m} = \hat{\mathbf{P}}_k \quad (4.72)$$

Aqui, m denota um novo intervalo discreto, de acordo com Δt .

De forma geral, o objetivo da pré-filtragem é a preparação dos dados de treinamento para que, o modelo obtido pela RNA treinada, reproduza de forma adequada as características cinemáticas do veículo, num determinado intervalo de tempo. Porém, o desempenho do pré-filtro pode influir no resultado final desta modelagem, quando modifica a resposta original do veículo, por exemplo, quando remove excessivamente as características de alta frequência dos sinais, resultado num filtro de característica passa - baixa muito acentuada.

A modelagem simplificada do pré-filtro, dada pelas Equações 4.52 e 4.53, e a sintonia manual da matriz \mathbf{Q} , conforme mostrado na Equação 4.61, sugerem que este filtro é adequado em aplicações de baixo desempenho, ou seja, para um veículo submetido às baixas acelerações angulares e lineares. Veículos com alto desempenho talvez necessitem de modelos mais elaborados dos sensores inerciais.

5 NAVEGAÇÃO INERCIAL AUXILIADA POR REDES NEURAIAS

A metodologia proposta neste trabalho é a de desenvolver e implementar algoritmos numéricos baseados em redes neurais artificiais, para compensar erros de posicionamento em um sistema inercial auxiliado por GPS. Os algoritmos serão testados via simulação por computador utilizando dados reais de uma IMU-MEMS de baixo custo (CROSSBOW CD400-200) e de um receptor GPS (Ashtech Z12), obtidos a partir de um veículo terrestre de uso urbano. Neste Capítulo será definida a arquitetura, modos de operação e métodos de treinamento para uma RNA dedicada ao problema de posicionamento.

5.1 Definição da RNA para o problema de posicionamento

Como mencionado anteriormente, uma rede neural do tipo MLP pode ser utilizada com a finalidade de suprir informações de navegação para um sistema integrado GPS/INS, durante o período de ausência do sinal GPS. Para tanto, a RNA deve ser adequadamente modelada, e posteriormente, devidamente treinada durante os períodos nos quais o GPS está operando.

Inicialmente, devem ser analisadas as entradas e saídas da RNA que definam a cinemática do veículo, conforme discutido no Capítulo 2, de acordo com as informações que possam ser observadas via GPS, e juntamente com informações adicionais disponíveis, como por exemplo, a qualidade dos sensores inerciais.

É proposta então, com base na discussão prévia, a seguinte modelagem básica de RNA:

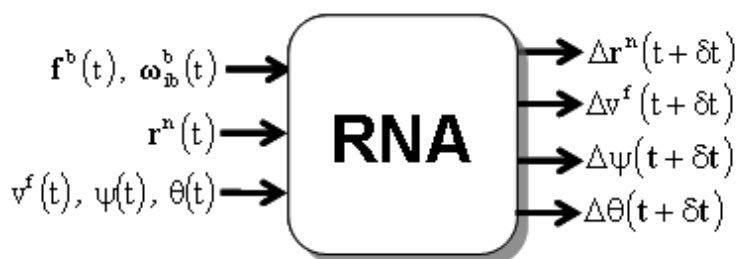


Figura 5.1 – Concepção básica da RNA: entradas e saídas.

Os sinais de saídas estão definidos na forma de incrementos e dessa forma, a RNA deverá cumprir dois objetivos: atuar como um integrador das equações de navegação, podendo estimar incrementos de posição, velocidade e dos ângulos ψ e θ ; e também atuar como compensador dos erros da IMU. Os incrementos são calculados na mesma frequência, δt , na qual os sinais de entrada são apresentados, da seguinte forma:

$$\Delta f(t + \delta t) = f(t + \delta t) - f(t) \quad (5.1)$$

Para validar os conceitos desenvolvidos no Capítulo 2, um modelo de RNA é derivado a partir da concepção básica apresentada na Figura 5.1, e considerando as seguintes hipóteses:

- a) O veículo se desloca no plano e em velocidade baixa o suficiente para que os movimentos de rolamento (*roll*) e arfagem (*pitch*) possam ser desconsiderados; portanto, com relação à sua atitude, somente mudanças de direção (*yaw*) são consideradas.
- b) As medidas inerciais são fornecidas por uma IMU-MEMS de baixo custo, com sinais ruidosos.
- c) O objetivo do sistema é definido para estimar a posição do veículo nas coordenadas Norte, p_N , Leste, p_E , e sua velocidade v^f , definida na direção preferencial do veículo.

Desta forma, a solução de navegação proposta é descrita por:

$$\mathbf{p}^n = [p_E \quad p_N \quad v^f]^T \quad (5.2)$$

A RNA fica configurada, com relação às entradas/saídas da seguinte forma:

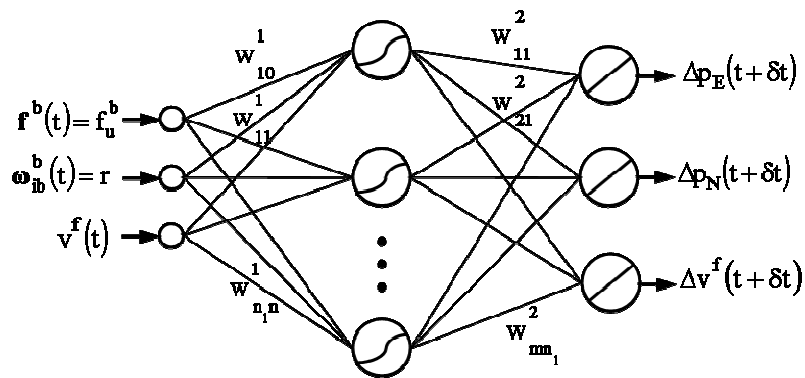


Figura 5.2 – Configuração da RNA para simulação.

5.2 Modos de operação da RNA

Redes neurais com técnicas de aprendizado supervisionado são aplicadas na solução de problemas em que se deseja obter um mapeamento entre padrões de entrada e saída, e são implementadas em duas formas básicas: *off-line* e *on-line*. No treinamento *off-line*, os padrões do conjunto de treinamento não se alteram, e, uma vez treinada, os pesos sinápticos e biases permanecem fixos, e a RNA está pronta para sua aplicação. Se novos padrões de treinamento são disponibilizados a RNA precisa ser re-treinada. No caso *on-line* o conjunto de treinamento muda continuamente, e a rede deve estar em contínuo processo de adaptação (BRAGA et al., 2007).

Logo, para uma RNA aplicada na solução de um problema de navegação, o treinamento só é possível se as informações do GPS e IMU estiverem disponíveis simultaneamente, para a formação de um conjunto de treinamento. Uma vez treinada, a RNA está pronta para sua utilização, e nesta situação somente os sinais da IMU estão disponíveis. Neste trabalho, sua aplicação ocorre na ausência das informações do GPS. Em geral, denominam-se dois modos de operação para este tipo de RNA: modo treinamento (*update mode*) e modo predição (*prediction mode*) (AGGARWAL et al., 2010).

No modo treinamento, enquanto o sinal GPS está disponível, os sinais de entrada são as forças específicas e velocidades angulares, no instante t , medidas pela IMU em intervalos δt ; e os demais sinais são estimados conforme a solução de navegação do GPS, $\mathbf{p}_{\text{GPS}}^n(t)$, e as Equações 2.52 até 2.57. A Figura 5.3 ilustra o modo treinamento.

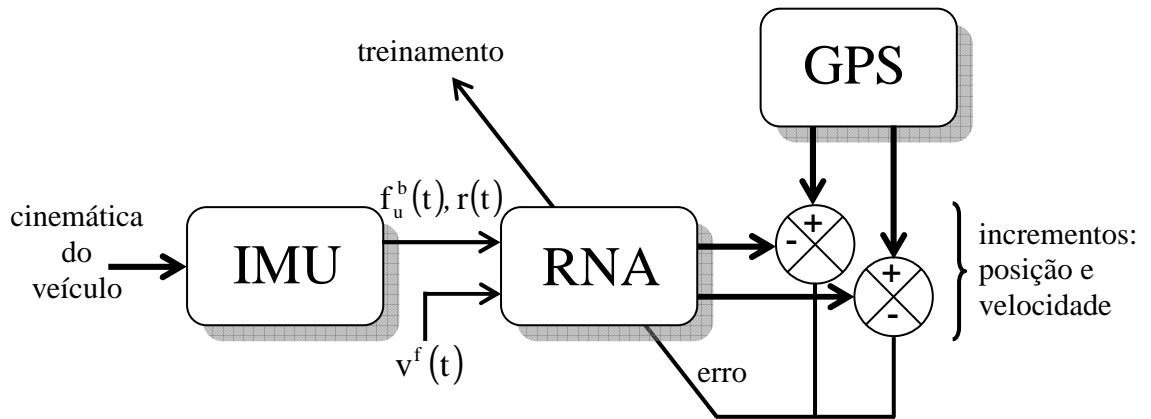


Figura 5.3 – Modo treinamento

Os incrementos, utilizados como saídas desejadas no cálculo dos erros de treinamento da RNA, podem ser obtidos por interpolação a partir das informações fornecidas pelo GPS, cujo intervalo de leitura é Δt , $\Delta t > \delta t$. Considere-se, por exemplo, o incremento de posição e velocidade obtido após duas leituras consecutivas do GPS:

$$\Delta \mathbf{p}_{\text{GPS}}^n(t + \Delta t) = \mathbf{p}_{\text{GPS}}^n(t + \Delta t) - \mathbf{p}_{\text{GPS}}^n(t) \quad (5.3)$$

$$\Delta \mathbf{v}_{\text{GPS}}^f(t + \Delta t) = \mathbf{v}_{\text{GPS}}^f(t + \Delta t) - \mathbf{v}_{\text{GPS}}^f(t) \quad (5.4)$$

Ou seja, a cada intervalo Δt são interpolados valores de incrementos de posição e velocidade para cada intervalo δt . De uma forma aproximada seriam interpolados n_i incrementos. No caso de interpolação linear, por exemplo, os valores interpolados, no intervalo $[t, t + \Delta t]$, seriam:

$$n_i \cong \frac{\Delta t}{\delta t} \quad (5.5)$$

$$\Delta \mathbf{p}_{\text{GPS}}^n(t + n\delta t) = [\mathbf{p}_{\text{GPS}}^n(t + \Delta t) - \mathbf{p}_{\text{GPS}}^n(t)] n \frac{\delta t}{\Delta t} \quad (5.6)$$

$$\Delta \mathbf{v}_{\text{GPS}}^f(t + n\delta t) = [\mathbf{v}_{\text{GPS}}^f(t + \Delta t) - \mathbf{v}_{\text{GPS}}^f(t)] n \frac{\delta t}{\Delta t} \quad (5.7)$$

onde $n = 0, 1, \dots, n_i$.

O erro $\mathbf{e} = [\mathbf{e}_{\text{POS}} \quad \mathbf{e}_{\text{VEL}}]$, nesse caso, é gerado para cada conjunto de treinamento, da seguinte forma.

$$\mathbf{e}_{\text{POS}}(t + \delta t) = \Delta \mathbf{p}_{\text{GPS}}^n(t + \delta t) - \Delta \mathbf{p}_{\text{RNA}}^n(t + \delta t) \quad (5.8)$$

$$\mathbf{e}_{\text{VEL}}(t + \delta t) = \Delta \mathbf{v}_{\text{GPS}}^f(t + \delta t) - \Delta \mathbf{v}_{\text{RNA}}^f(t + \delta t) \quad (5.9)$$

A entrada da rede $\mathbf{v}^f(t)$, no intervalo $[t, t + \Delta t]$, é obtida da seguinte forma:

$$\mathbf{v}^f(t) = \mathbf{v}_{\text{GPS}}^f(t) = \mathbf{v}_0^f \quad (5.10)$$

$$\mathbf{v}^f(t + n\delta t) = \mathbf{v}_0^f + \sum_{n=1}^{n_1} \Delta \mathbf{v}_{\text{GPS}}^f(t + n\delta t) \quad (5.11)$$

O mesmo procedimento é repetido para todos os sinais envolvidos que necessitem de interpolação.

No modo predição, conforme Figura 5.4, enquanto o sinal GPS não está disponível, os sinais de entrada provenientes da IMU continuam sendo apresentados na frequência δt , enquanto que os demais são computados da seguinte forma.

Considere $\mathbf{p}^n(t_u) = \mathbf{p}_{\text{GPS}}^n(t_u) = \mathbf{p}_u^n$, onde t_u é o instante da última informação disponível do GPS, antes da perda do sinal. No próximo instante,

$$\mathbf{p}_{\text{NAV}}^n(t_u + i\delta t) = \mathbf{p}_u^n + \sum_i \Delta \mathbf{p}_{\text{RNA}}^n(t_u + i\delta t) \quad (5.12)$$

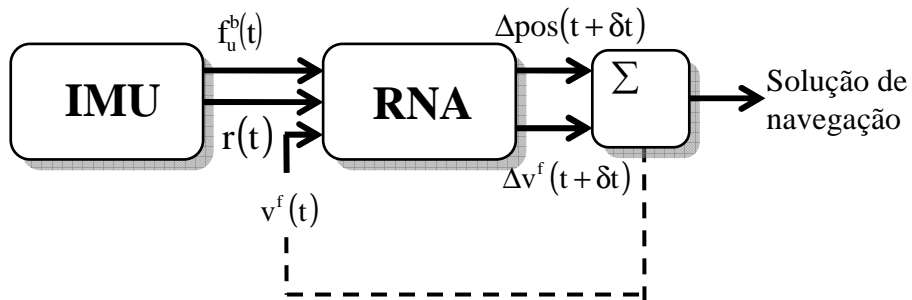


Figura 5.4 – Modo predição.

Ou seja, os incrementos na saída da RNA são somados à última informação proveniente do GPS e, no caso da velocidade \mathbf{v}^f , realimentados na camada de entrada. No modo

predição, os sinais de saída da RNA representam uma estimativa da solução de navegação, a partir do instante t_u .

5.3 Métodos de treinamento para o sistema de navegação proposto

Durante o período de treinamento, enquanto as informações do GPS estão disponíveis, existem métodos diferentes de combinar o treinamento da RNA pela abordagem do filtro de Kalman com a forma de apresentação dos dados de treinamento, e que serão discutidos a seguir.

5.3.1 Método 1

A forma mais simples de treinar a RNA, para o sistema de navegação e treinamento propostos, é processar e armazenar sequencialmente as informações do GPS e IMU, em conjuntos, ou janelas, de dados para treinamento, de tamanho pré-definido; e ao se completar um conjunto, o procedimento de treinamento começa. Ao final do treinamento, o resultado (os pesos sinápticos da RNA) é armazenado e disponibilizado para uso. Assim, enquanto se forma uma nova janela e havendo necessidade, a RNA utiliza os parâmetros obtidos do treinamento da janela imediatamente anterior.

Numa situação de aplicação, é improvável que a interrupção, ou degradação severa, do sinal GPS ocorra exatamente após a formação de uma janela de dados. Uma possível desvantagem em se trabalhar com janelas de dimensão fixa, é que, se a cinemática do veículo se alterar de maneira significativa, no intervalo da formação de uma nova janela, a modelagem anterior pode não predizer adequadamente a trajetória do veículo. Janelas de maior extensão e veículos com uma cinemática mais rápida estariam mais propensos a essa situação. Uma possível solução, porém com custo computacional elevado, seria adotar uma janela móvel de dados, com o descarte e inserção de dados na mesma proporção e posterior treinamento.

Considerando a escala temporal na Figura 5.5, ao se completar um conjunto de dados J_i , de dimensão constante, o respectivo treinamento T_i é realizado, e os pesos da RNA são

sempre reinicializados aleatoriamente. Ao final do treinamento o resultado W_i é armazenado.

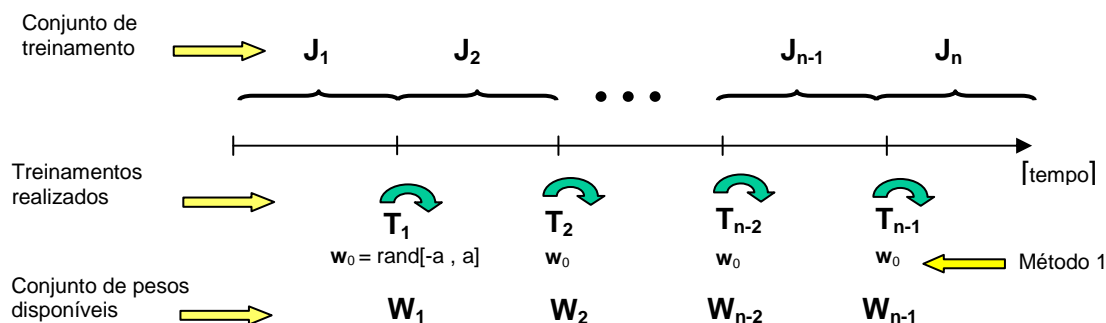


Figura 5.5 – Método de treinamento 01.

5.3.2 Método 2

Uma variação, em relação ao método 1, é inicializar os pesos aleatoriamente somente no primeiro treinamento, enquanto que os demais são inicializados com o resultado do treinamento anterior. Trata-se, segundo Chiang et al. (2004), de uma tentativa de manter alguma memória do treinamento anterior, modificando os pesos armazenados com informações novas. A figura 5.6 ilustra o método 2.

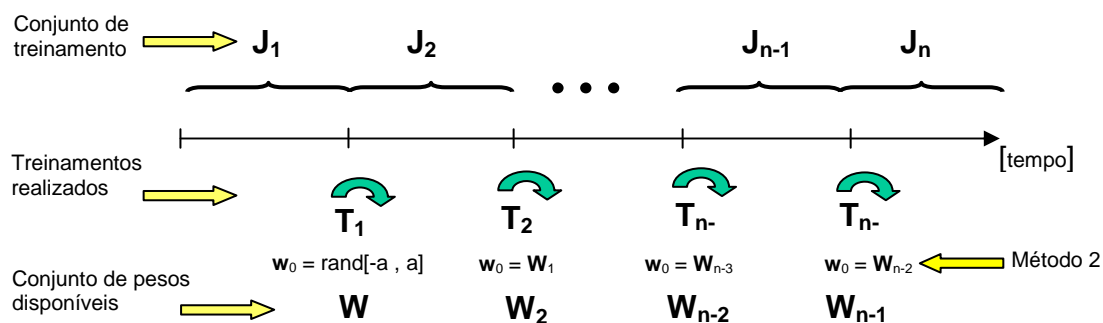


Figura 5.6 – Método de treinamento 02.

5.3.3 Método 3

Trata-se de uma modificação proposta, em relação ao método 2, pela atualização dos pesos armazenados, via “filtragem” de novos padrões de treinamento, usando a abordagem do filtro de Kalman adaptativo, enquanto se aguarda um novo conjunto de treinamento ser formado. O objetivo é evitar que alterações na cinemática do veículo,

no intervalo de tempo de formação de uma nova janela de dados, não deteriore a modelagem representada pelos pesos armazenados.

De acordo com a Figura 5.7, a seguir:

- No primeiro treinamento, T_1 , os pesos são inicializados aleatoriamente. O resultado, W_1 , é armazenado;
- No segundo treinamento T_2 , enquanto o conjunto de dados, J_2 , é formado, o resultado prévio W_1 é atualizado pela filtragem de novos pares de treinamento, padrão-por-padrão. Ao se completar J_2 , o conjunto de dados é treinado com a condição inicial $w_0 = W_{1\text{upd}}$, ou seja, com o resultado do treinamento anterior atualizado.
- Repete-se o item anterior, durante a navegação, enquanto houver sinal GPS.

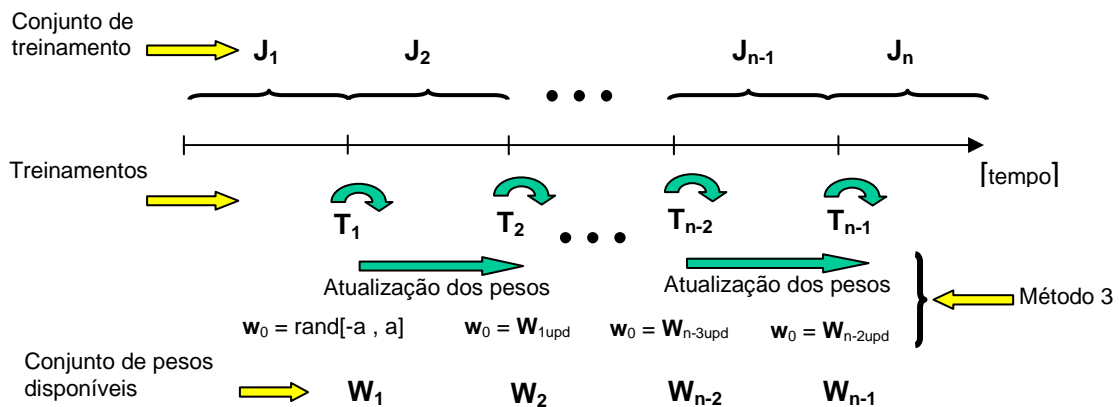


Figura 5.7 – Método de treinamento 03.

5.3.4 Método 4

É proposto que ao término do primeiro treinamento, após a formação do primeiro conjunto de padrões, J_1 , os pesos resultantes, W_1 , sejam somente atualizados, via “filtragem” de novos padrões de treinamento, até que um determinado critério seja atingido, e o processo então se repete. A Figura 5.8 mostra o método 4.

O processo de treinamento pode ser reinicializado logo após o período de interrupção do GPS, quando uma nova janela de dados pode ser formada; ou também ser reinicializado após um intervalo de tempo pré-determinado.

O método explora o conceito de adaptação, discutido no Capítulo 4, e aqui repetido apenas para facilidade de leitura:

$$\beta E \left[v_j^2(t) \right] = \mathbf{H}_j(t, \bar{\mathbf{w}}) [\mathbf{P}(t) + \mathbf{Q}(t)] \mathbf{H}_j^T(t, \bar{\mathbf{w}}),$$

lembrando que valores do parâmetro empírico β , ligeiramente menores que 1, indicam que as informações armazenadas possuem mais valor que as novas informações processadas, denotando o caráter de preservação de memória quando se atualiza o resultado do treinamento de um conjunto, ou janela, de dados anterior, com novos padrões de treinamento.

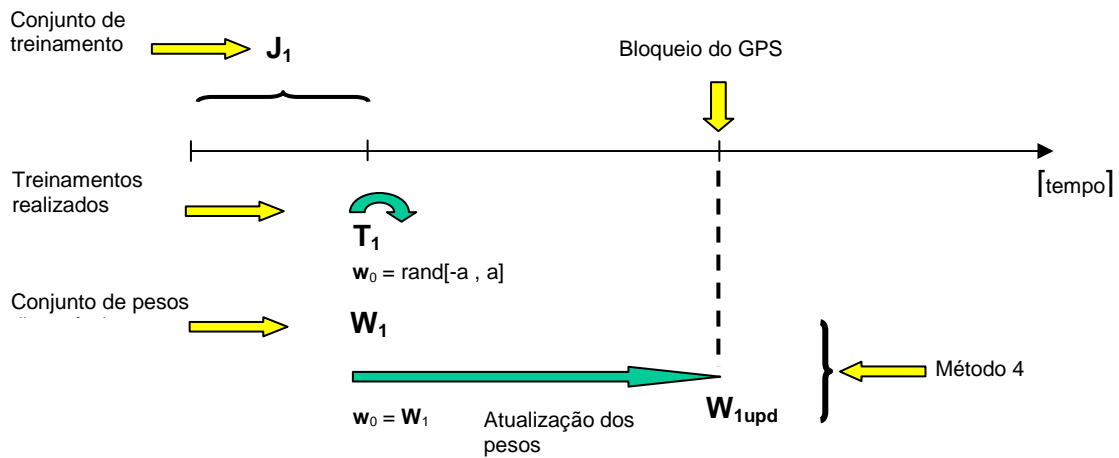


Figura 5.8 – Método de treinamento 04

6 SIMULAÇÃO E RESULTADOS

6.1 Introdução

Os dados para simulação foram obtidos a partir de uma unidade de medida inercial Crossbow CD400-200, baseada em sensores com tecnologia MEMS e considerada de baixo custo, e de um receptor GPS Ashtech Z-12. Maiores informações sobre a IMU e o receptor GPS podem ser encontrados no Apêndice A.

As campanhas para aquisição de dados foram realizadas dentro das dependências do INPE, em São José dos Campos, S.P., em ruas e avenidas relativamente planas, com entorno caracterizado por construções de médio porte e árvores de grande estatura. Os instrumentos foram montados num veículo de pequeno porte de uso urbano e foi respeitada a sinalização interna de trânsito, com velocidade máxima permitida de 40 km/h. Em todos os ensaios, procurou-se nivelar e alinhar a IMU às condições iniciais de posição e atitude já conhecidas.

6.2 Verificação *off-line* do algoritmo de treinamento proposto

O algoritmo de treinamento de redes neurais utilizando a abordagem do filtro de Kalman com procedimento adaptativo (RIOS NETO, 1997), apresentado no Capítulo 4, já havia sido preliminarmente testado por Souza Filho e Rios Neto (1998) e por Silva e Rios Neto (1999) em problemas clássicos de *benchmark* do tipo PROBEN 1 (diagnóstico de câncer) e XOR (ou exclusivo). Em ambos os casos os resultados foram classificados como positivos em relação à convergência e precisão e o método avaliado como promissor quando seu desempenho foi comparado com o algoritmo do gradiente. Uma aplicação do mesmo algoritmo também pode ser encontrada em Rios Neto, W. (2008) no treinamento de uma rede neural do tipo *Functional Link Network* (FLN) utilizada na estimação de coeficientes aerodinâmicos e de controle de uma aeronave de alto desempenho. Considerando os resultados obtidos nas aplicações citadas anteriormente, foram adotados os seguintes valores empíricos para os parâmetros de ajuste do procedimento adaptativo: $\alpha = 0,2$, para o parâmetro de perturbação linear da

Equação 4.16, que garante a linearidade das medidas; e $\beta = 0,9$, para o parâmetro da Equação 4.33, que trata da observação associada do processo de estimação da quantidade de ruído da matriz \mathbf{Q} .

Neste trabalho o algoritmo de treinamento e a modelagem da RNA propostos anteriormente foram verificados, inicialmente, por simulação off-line de um problema de aproximação de função e utilizando os dados de treinamento adquiridos nas campanhas de aquisição com o GPS e IMU. Os resultados foram comparados com aqueles obtidos pela mesma RNA treinada pelo algoritmo Levenberg-Marquardt (LM), conhecido pela sua eficiência (HAGAN e MENHAJ, 1994). Foi utilizado o software MatLab® na implementação das simulações, com soluções próprias para os métodos baseados no filtro de Kalman e utilizadas funções do *Neural Network Toolbox* para o treinamento com o método LM. Deve ser recordado que o método LM requer a apresentação dos padrões de treinamento em lote (*batch mode*), enquanto que o Filtro de Kalman também admite a forma seqüencial de apresentação, além da forma em lote. O objetivo da simulação é avaliar a capacidade de aprendizagem da RNA com sinais reais.

Os dados disponíveis para treinamento são divididos em dois conjuntos distintos por sorteio aleatório. São apresentados 70% dos dados para o treinamento da RNA propriamente dito e os restantes 30% para teste de validação. Os pares de treinamento foram submetidos aos métodos de treinamento em 100 partidas independentes, ou seja, inicializando-se aleatoriamente, a cada partida, os parâmetros da RNA, pesos e biases, no intervalo $[-0,5 \ 0,5]$. O critério de parada utilizado foi finalizar o treinamento após 40 iterações (épocas). As RNA simuladas possuem 20 neurônios na camada oculta, com função de ativação sigmóide, e função linear na camada de saída.

Concluído o treinamento, os dados disponíveis são, na sua totalidade, reapresentados à RNA no modo predição, e as saídas correspondentes são comparadas com a trajetória do GPS, aqui tomada como referência. Os resultados são, então, comparados entre o método de treinamento proposto (FKE-A) e o algoritmo de Levenberg-Marquardt.

6.2.1 Descrição dos dados utilizados na verificação *off-line*

As Figuras 6.1 até 6.4 descrevem a trajetória do veículo, no intervalo de tempo utilizado na simulação [632s, 715s], e respectivas medidas da IMU, forças específicas (acelerações lineares) e velocidades angulares. Deve ser observado que nesta campanha de aquisição, os dados da IMU foram adquiridos numa frequência de 20Hz, e os dados do receptor GPS em 2Hz.

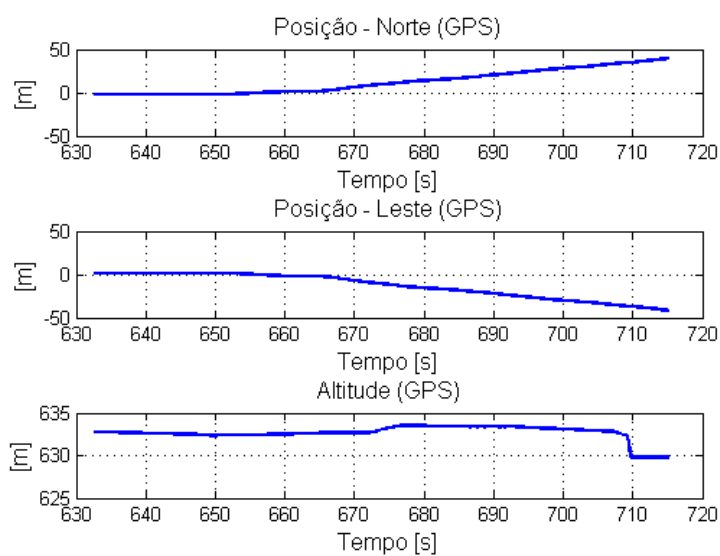


Figura 6.1 - Deslocamento fornecido pelo GPS.

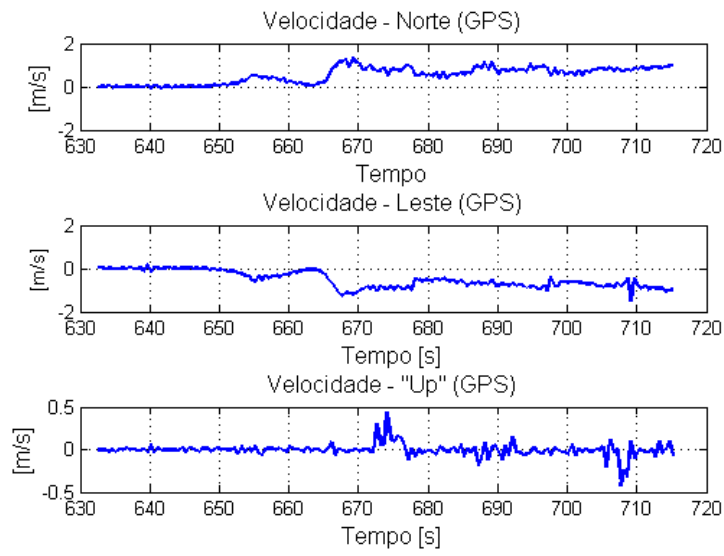


Figura 6.2 - Velocidades estimadas via GPS.

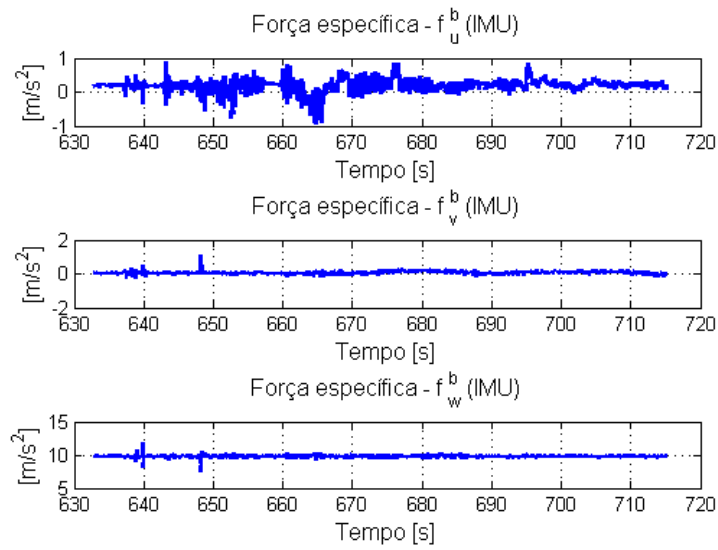


Figura 6.3 – Forças específicas fornecidas pela IMU.

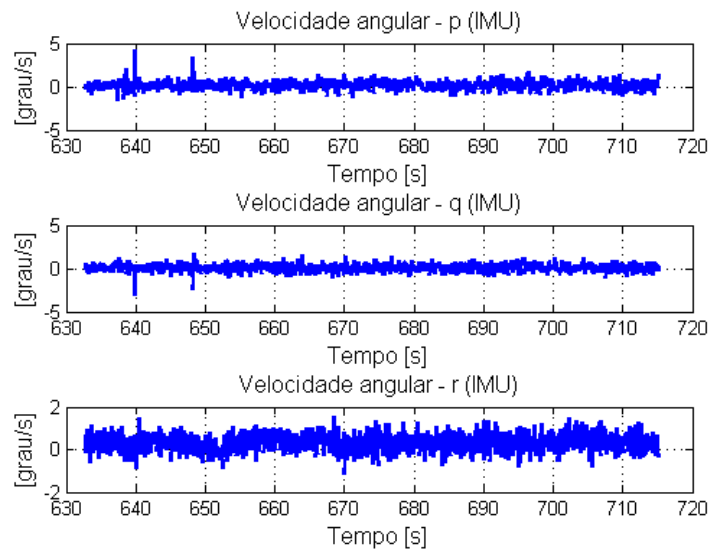


Figura 6.4 - Velocidades angulares fornecidas pela IMU.

Um dos fatores que pode determinar um desempenho adequado da RNA é a caracterização adequada da cinemática do veículo pelos sensores inerciais e pelas informações do GPS, e contidas no conjunto de pares de treinamento. O vetor de entradas da RNA, com medidas de força específica e velocidade angular, fornecidos por sensores MEMS, e a velocidade v^f , estimada por dados fornecidos por GPS podem apresentar nível de ruído excessivo, devendo, nesse caso ser pré-processado, numa tentativa de suavização dos sinais. O objetivo é fazer com que a RNA aprenda mais com a variação temporal dos sinais e menos com os respectivos ruídos.

O filtro de Kalman, apresentado na seção 4.4, é utilizado para a pré-filtragem. As Figuras 6.5 e 6.6 mostram os resultados obtidos para os sinais de entrada da RNA proposta para navegação.

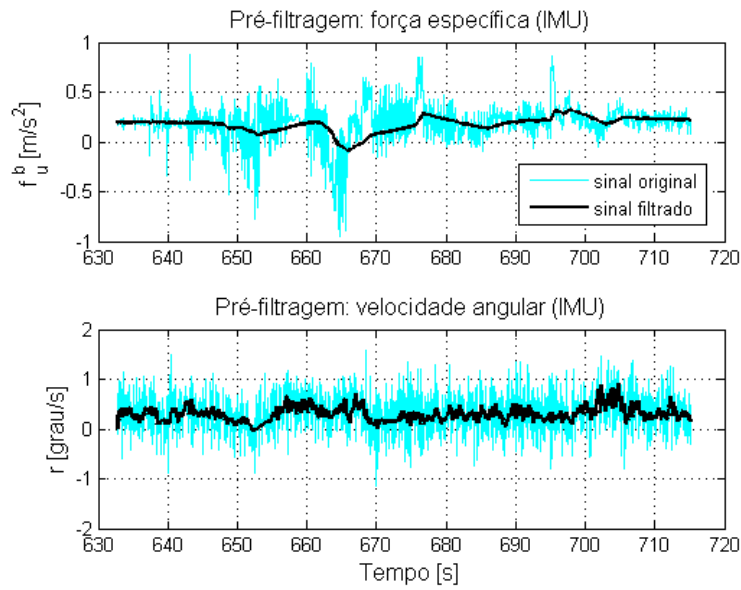


Figura 6.5 - Sinais da IMU pré-filtrados.

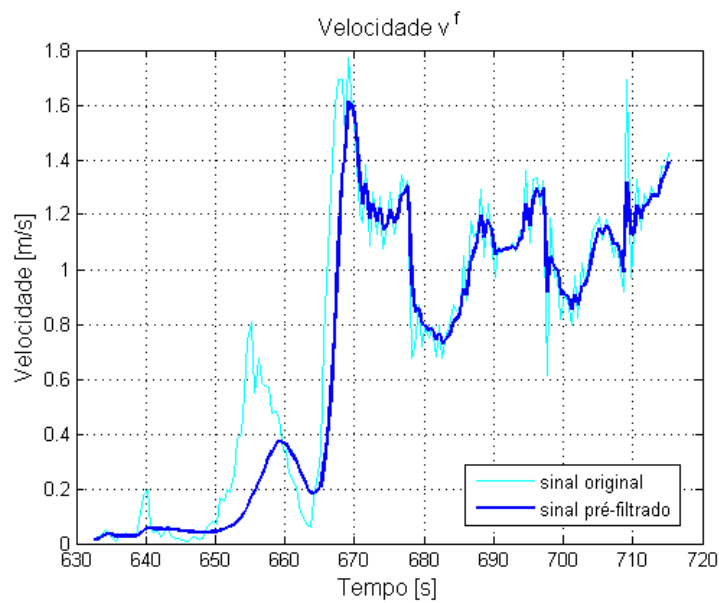


Figura 6.6 - Velocidade v^f pré-filtrada.

6.2.2 Resultados da verificação *off-line*

A primeira simulação foi conduzida com dados pré-filtrados. A Figura 6.7 mostra os erros de posicionamento e velocidade, tomados em relação às informações do GPS, e obtidos pela média das 100 partidas, enquanto que a Tabela 6.1 compara o erro

quadrático médio das posições e velocidade, que foi tomado ao longo do intervalo simulado.

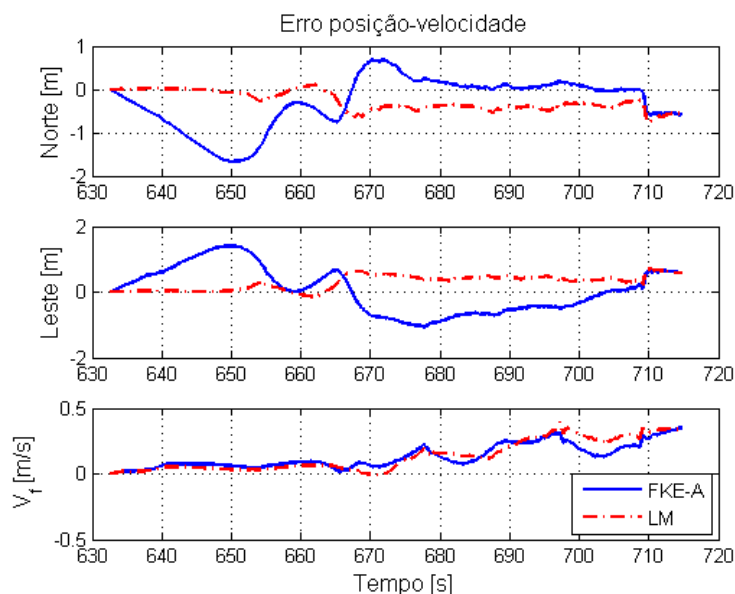


Figura 6.7 - Comparação dos erros de posição e de velocidade (FKE-A, LM).

Tabela 6.1 – Erros de posição e velocidade (mse) – dados pré-filtrados

	FKE-A	LM
Leste	0,4889	0,1489
Norte	0,4510	0,1395
v^f	0,0249	0,0313

A Figura 6.8 compara as curvas do erro de treinamento dos dois métodos, em função do número de iterações. Após a conclusão das 100 partidas foi tomada a média do erro final de treinamento, ou seja, a média dos erros de treinamento da última iteração e respectivo desvio padrão, numa tentativa de se avaliar a robustez dos métodos:

a) FKE-A: $1,0710 \cdot 10^{-6} \pm 7,5732 \cdot 10^{-6}$

b) LM: $2,1520 \cdot 10^{-5} \pm 2,2124 \cdot 10^{-6}$

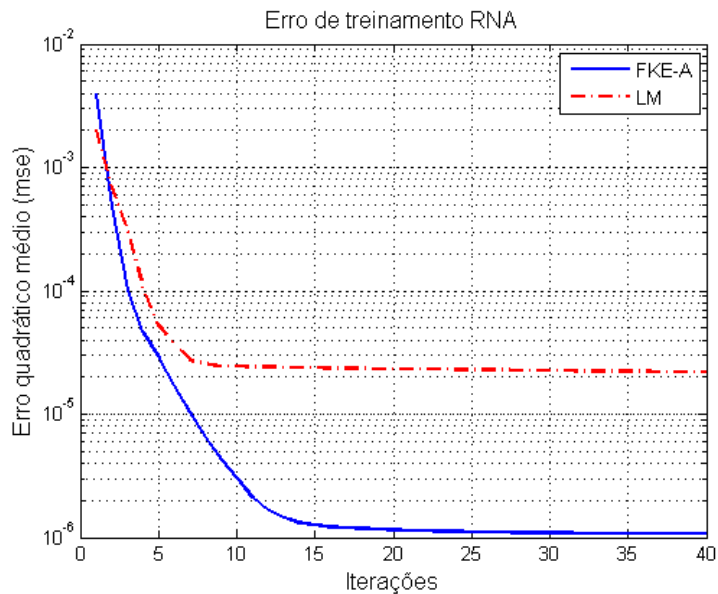


Figura 6.8 – Erro de treinamento da RNA (FKE-A, LM).

Pode ser observado que para o intervalo simulado os métodos FKE-A e LM possuem desempenho semelhante. O método LM apresenta erros médios de posição menores que o FKE_A, conforme tabela 6.1, indicando neste problema de aproximação de função, uma melhor característica de generalização, uma vez que os erros também consideram os dados de teste (30 % do total de dados), mesmo com o método FKA-E apresentando um erro médio de treinamento menor. O método LM também parece estabilizar em torno de um valor final em menos iterações, como pode ser visto na Figura 6.8.

A mesma simulação foi repetida com dados sem pré-filtragem, ou seja, sem suavização. A Figura 6.9 mostra os erros de posicionamento e velocidade; a Tabela 6.2 compara o erro quadrático médio das posições e velocidade, e finalmente a Figura 6.10 compara as curvas do erro de treinamento dos métodos, em função do número de iterações. Também foi calculada a média dos erros de treinamento da última iteração com respectivo desvio padrão:

a) FKE-A: $6,0240 \cdot 10^{-6} \pm 4,2596 \cdot 10^{-5}$

b) LM: $8,6259 \cdot 10^{-5} \pm 3,6031 \cdot 10^{-6}$

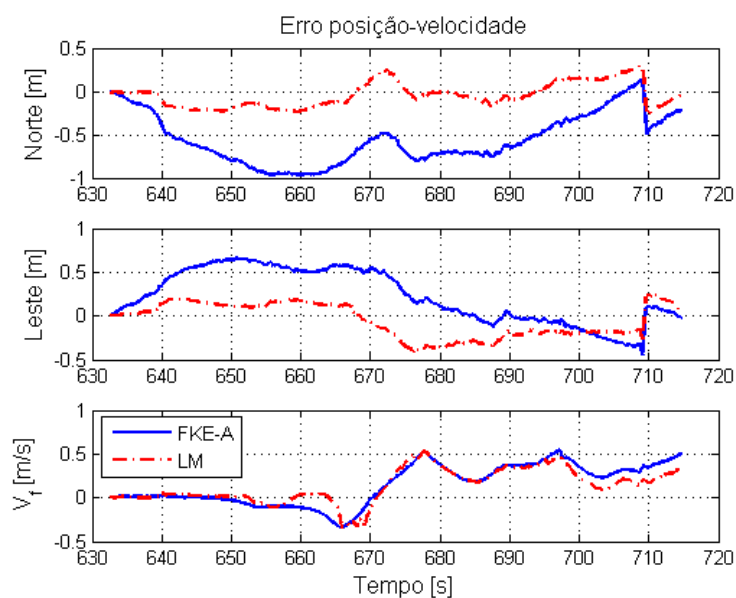


Figura 6.9 - Comparação dos erros de posição e de velocidade (FKE-A, LM).

Tabela 6.2 – Erros de posição e velocidade (mse) – dados originais

	FKE-A	LM
Leste	0,1437	0,0411
Norte	0,3928	0,0226
v^f	0,0726	0,0335

Pode ser observada nesta segunda simulação, com os padrões de treinamento mais ruidosos, uma ligeira melhora do método LM, em relação ao FKE-A, quanto aos erros de posicionamento, bem como o aumento nos erros de treinamento para ambos os métodos.

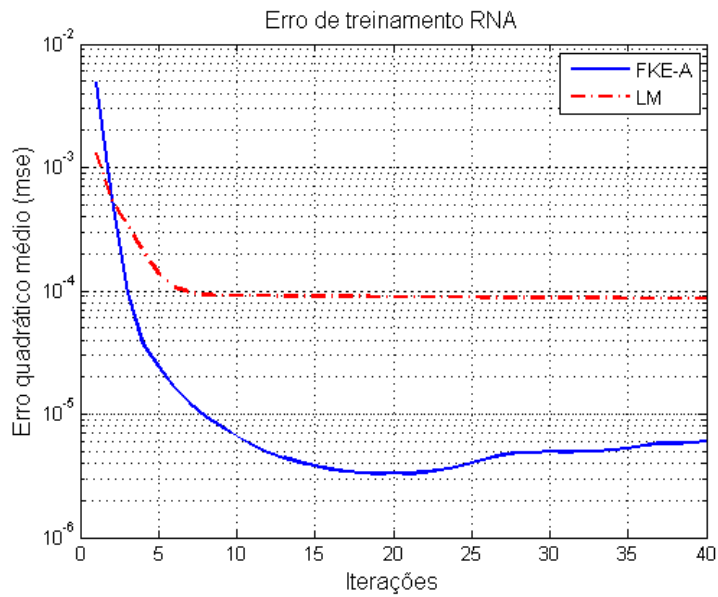


Figura 6.10 - Erro de treinamento da RNA (FKE-A, LM).

O efeito da solução adaptativa, com a estimação do ruído no estado, pode ser observado comparando-se os resultados obtidos anteriormente com aqueles obtidos pela mesma RNA treinada por um FKE padrão. As Figuras 6.11 e 6.12 mostram, respectivamente, os erros de posição e velocidade e erros de treinamento, para a simulação com padrões de treinamento pré-filtrados.

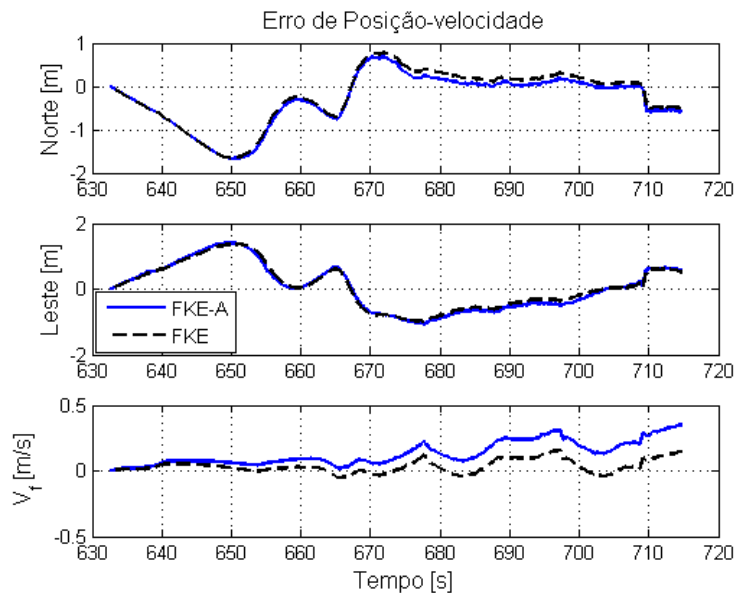


Figura 6.11 - Comparação dos erros de posição e de velocidade (FKE, FKE-A).

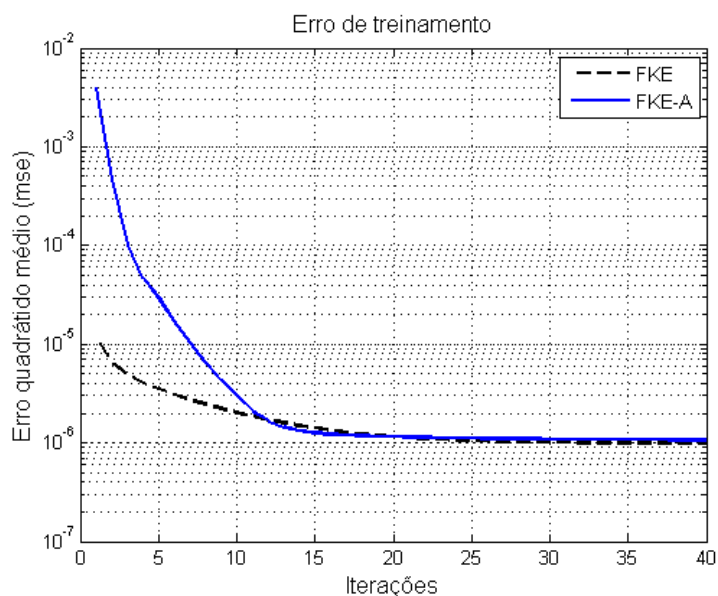


Figura 6.12 - Erro de treinamento da RNA (FKE, FKE-A).

A Tabela 6.3 compara o erro quadrático médio das posições e velocidade. Também foi calculada a média dos erros de treinamento da última iteração com respectivo desvio padrão:

a) FKE-A: $1,071 \cdot 10^{-6} \pm 7,5732 \cdot 10^{-5}$

b) FKE: $9,8406 \cdot 10^{-7} \pm 6,9583 \cdot 10^{-6}$

Tabela 6.3 - Erros de posição e velocidade (mse) – dados pré-filtrados

	FKE-A	FKE
Leste	0,4889	0,4475
Norte	0,4510	0,4546
v^f	0,029	0,035

As Figuras 6.13 e 6.14 mostram, respectivamente, os erros de posição e velocidade e erros de treinamento, para a simulação com padrões de treinamento sem pré-filtragem. A Tabela 6.4 compara o erro quadrático médio das posições e velocidade. Também foi calculada a média dos erros de treinamento da última iteração com respectivo desvio padrão:

a) FKE-A: $6,0240 \cdot 10^{-6} \pm 4,2596 \cdot 10^{-5}$

b) FKE: $8,6043 \cdot 10^{-6} \pm 6,0841 \cdot 10^{-5}$

Tabela 6.4 - Erros de posição e velocidade (mse) – dados originais

	FKE-A	FKE
Leste	0,1437	17,9671
Norte	0,3928	13,2266
v^f	0,0726	0,3265

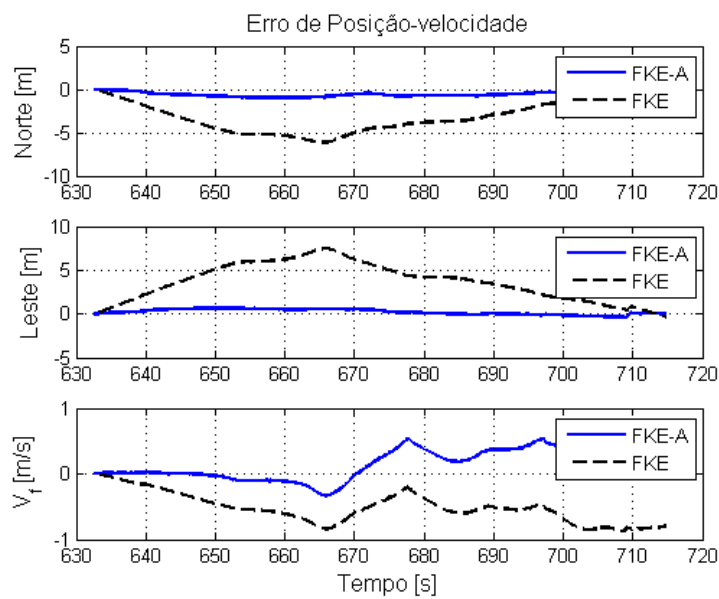


Figura 6.13 - Comparação dos erros de posição e de velocidade (FKE, FKE-A).

Fica evidente o efeito da solução adaptativa no treinamento das RNA, principalmente nos erros de posicionamento e de treinamento, conforme pode ser observado na Tabela 6.4 e Figura 6.14.

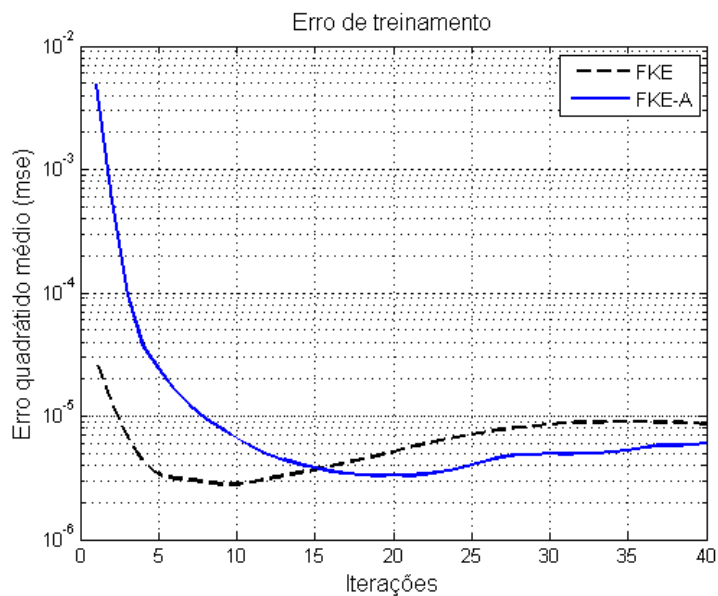


Figura 6.14 - Erro de treinamento da RNA (FKE, FKE-A).

Para finalizar a simulação *off-line* é comparado o efeito da pré-filtragem para cada método de treinamento testado, ou seja, LM, FKE-A e FKE, conforme mostrado nas Figuras 6.15 até 6.17, onde estão plotadas as curvas de erro de treinamento para dados originais vs. pré-filtrados.

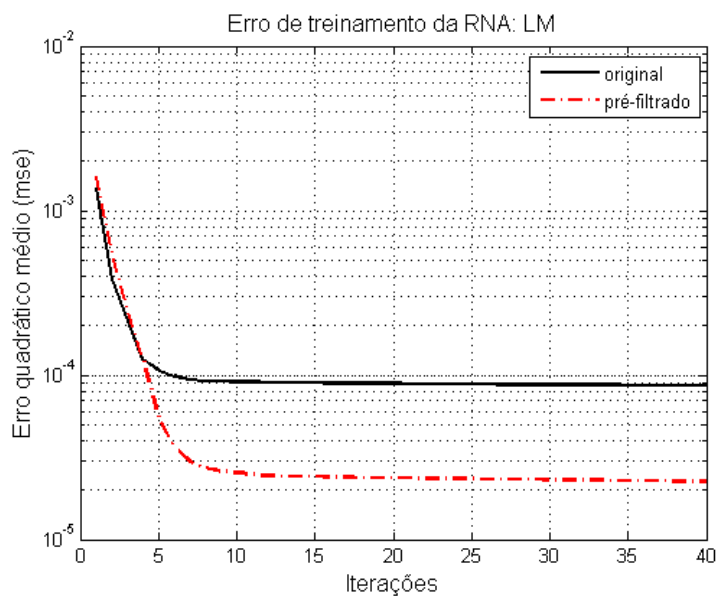


Figura 6.15 – Comparação entre erros de treinamento – LM

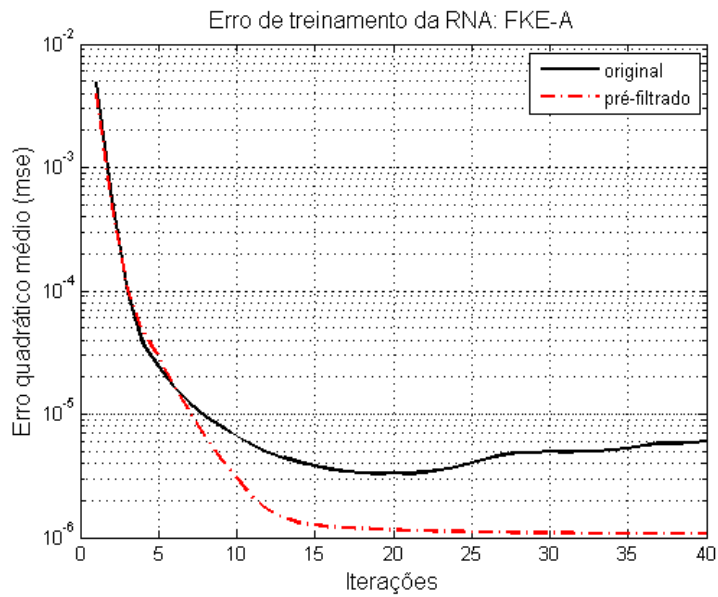


Figura 6.16 - Comparação entre erros de treinamento – FKE-A

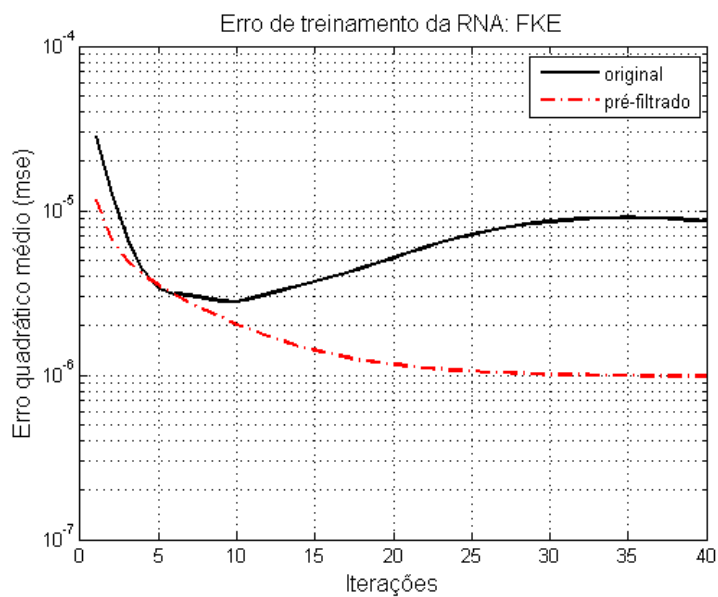


Figura 6.17 - Comparação entre erros de treinamento – FKE

Ao final desta simulação, para o problema de aproximação de função no intervalo de dados proposto, pode-se afirmar que o método de treinamento proposto baseado no filtro de Kalman com solução adaptativa, FKE-A, tem desempenho similar ao tradicional método LM, o qual parece apresentar mais habilidade de generalização, uma vez que apresentou menores erros médios de posição; enquanto que o FKE-A aparenta

ter seus erros de treinamentos estacionados em valores menores e possui a característica de admitir a apresentação de dados de forma sequencial, o que é vantajoso, tendo em vista aplicações em tempo real.

6.2.3 Considerações sobre o número de neurônios ocultos e de iterações

A arquitetura da RNA dedicada ao problema de navegação, com relação às entradas e saídas, foi definida no Capítulo 5. Porém a determinação do número de neurônios ocultos, na sua camada intermediária, é fundamental na questão do seu aprendizado, pois determina seu desempenho e capacidade de generalização. Como o número de neurônios define a complexidade da rede, define também a sua abrangência em termos de soluções possíveis. Então, deve-se ajustar adequadamente a complexidade do modelo neural à complexidade do problema em questão (BRAGA et al., 2007).

Não há na literatura uma regra geral que determine de forma precisa o número de neurônios em função de um determinado problema, mas existem abordagens que tentam contornar e apresentar opções de solução como, por exemplo, Regularização Bayesiana, técnicas de validação, e abordagem multi-objetivo para treinamento; ou mesmo heurísticas dedicadas à um problema específico como determinar o número de neurônios em função do tamanho do conjunto de treinamento e erros de treinamento associados (CARRARA; RIOS NETO, 1999).

Considere um treinamento típico, para a RNA do problema de aproximação de função apresentado anteriormente na seção 6.2. Concluído o treinamento, os dados de validação, que correspondem a 30% do total de dados disponíveis para treinamento, são apresentados à RNA treinada e as correspondentes saídas são confrontadas com os valores desejados num problema de regressão linear. As Figuras 6.18 e 6.19 mostram o resultado da regressão, para uma situação em que a RNA tem 20 neurônios na camada oculta. O índice R mede a correlação do problema, podendo fornecer alguma indicação da generalização da RNA para o problema dado. Índices próximos à unidade indicam uma boa correspondência entre a resposta da RNA e as saídas desejadas. Maiores detalhes sobre análise de regressão linear e a determinação do coeficiente de correlação, linear podem ser encontrados em Bingham e Fry (2010).

Para verificar a influência do número de neurônios ocultos no índice R, repete-se esta avaliação para diferentes quantidades. A Figura 6.20 apresenta os valores médios, para cinco partidas independentes de treinamento, do índice R e dos erros de treinamento e de validação. Observa-se que a partir de 20-25 neurônios o índice R não melhora.

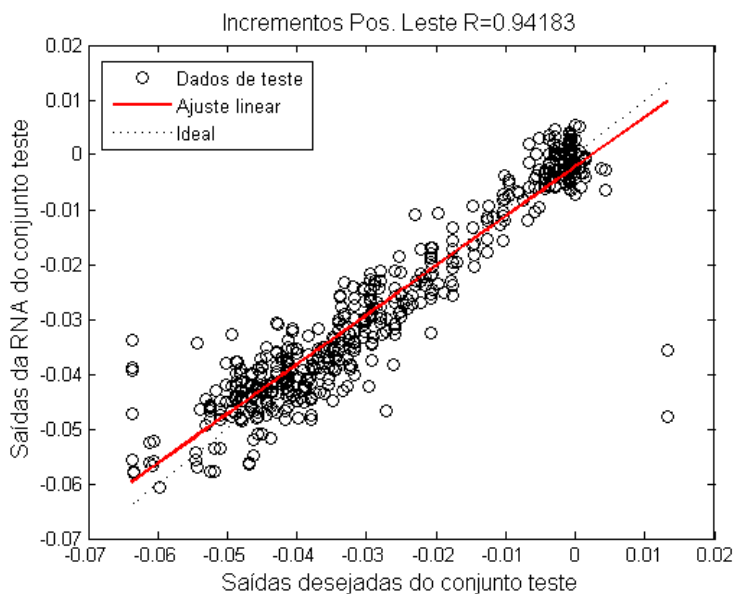


Figura 6.18 – Índice R para dados de validação (Leste).

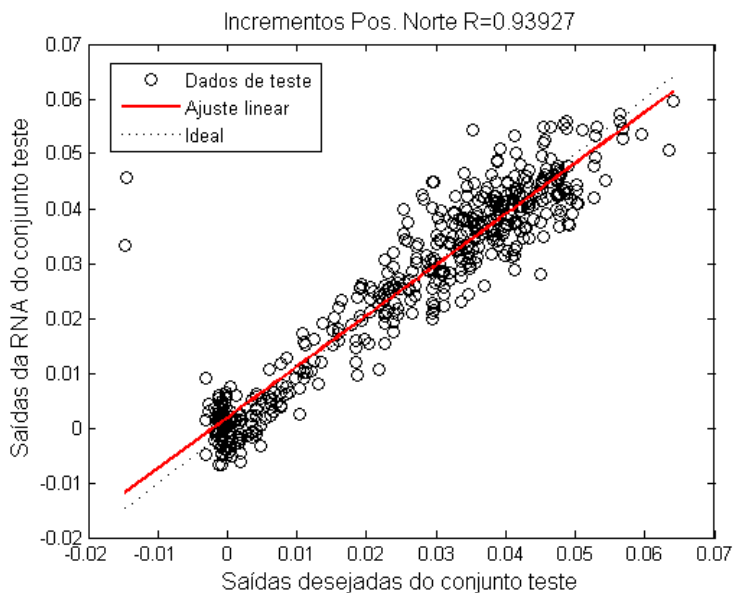


Figura 6.19 - Índice R para dados de validação (Norte).

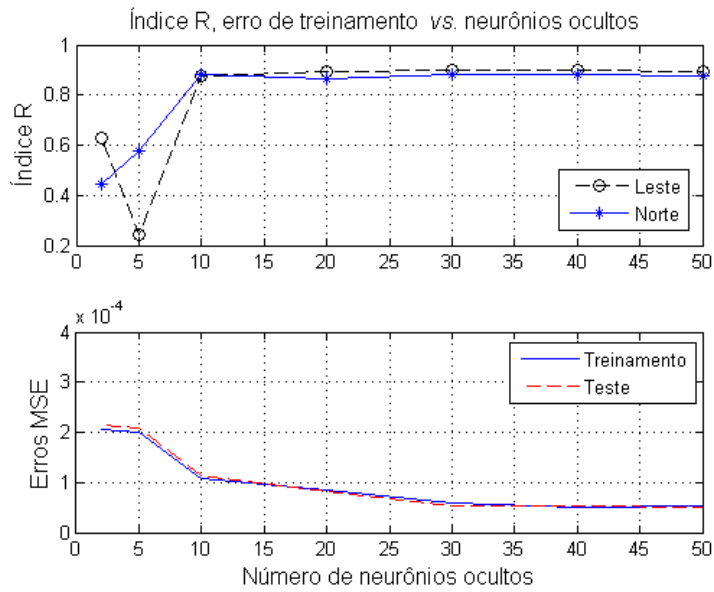


Figura 6.20 – Índice R e erros de treinamento vs. neurônios ocultos.

Repete-se o teste para a RNA com 20 neurônios ocultos, para verificar a influência do número de iterações, ou épocas de treinamento, no índice R. A Figura 6.21 apresenta os valores médios, para cinco partidas independentes de treinamento, do índice R e dos erros de treinamento e de validação. Observa-se que a partir de 40 iterações o índice R não melhora.

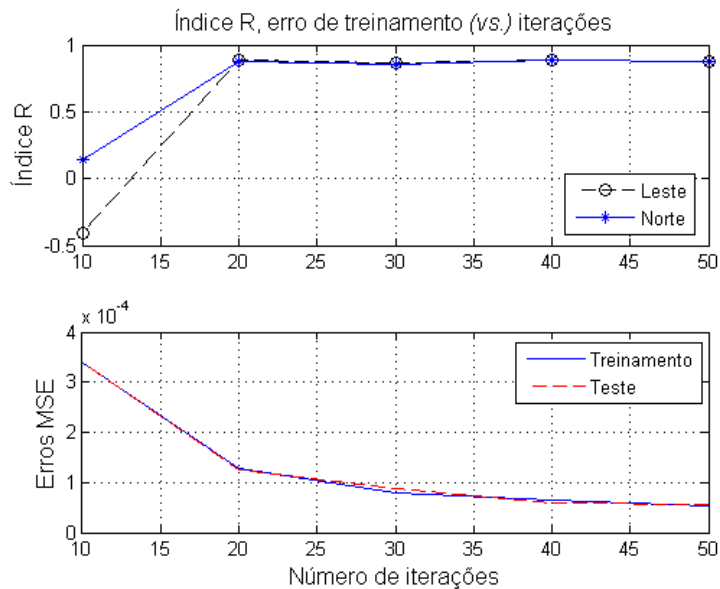


Figura 6.21 – Índice R e erros de treinamento vs. iterações.

A generalização de uma RNA representa sua capacidade de apresentar respostas adequadas quando novas entradas, diversas do conjunto de treinamento, são apresentadas. De acordo com Haykin (2001), são três os fatores que influenciam na generalização: O tamanho do conjunto de treinamento e sua representatividade do ambiente de interesse; a arquitetura da RNA e a complexidade do problema em questão.

A definição do tamanho do conjunto de dados neste trabalho, se confunde com a definição das chamadas janelas de dados dos métodos de treinamento, de 1 até 4, apresentados no Capítulo 5. Trata-se de um problema de solução empírica, para a qual se deve considerar o tipo de navegação requerida, ou seja, se o veículo se desloca com mais ou menos aceleração; se faz curvas com mais ou menos velocidade ou se mantém velocidade constante por longos períodos de tempo.

Janelas de dados menores podem ser treinadas com maior rapidez, porém podem não possuir dados em número ou qualidade suficiente para caracterizar a cinemática do veículo, no intervalo de tempo em questão. Uma janela maior pode demandar um número de neurônios maior, levando mais tempo para ser treinada, mas pode aprender mais as características do comportamento do veículo.

6.3 Simulação dos métodos propostos para o sistema de navegação

Considere os métodos de treinamento propostos para a RNA dedicada à navegação, detalhados no Capítulo 5, que serão resumidos a seguir, apenas para uma melhor continuidade da leitura:

Método 1: o treinamento da RNA é realizado a cada formação de um novo conjunto, ou janela de dados, e os pesos são sempre reinicializados aleatoriamente.

Método 2: o treinamento é realizado a cada novo conjunto de dados, ou padrões. No primeiro treinamento os pesos são inicializados aleatoriamente, enquanto que os demais são inicializados com o resultado do treinamento anterior.

Método 3: difere do método anterior pela atualização dos pesos, via “filtragem”, enquanto aguarda um novo conjunto de treinamento ser formado.

Método 4: O primeiro treinamento é realizado após a formação do primeiro conjunto de padrões e então os pesos são atualizados, via “filtragem”, até o início da predição.

A simulação dos métodos consiste em apresentar os dados pré-filtrados à RNA, que opera no modo treinamento enquanto as informações provenientes do GPS forem disponibilizadas. Será simulada, em diferentes instantes, a interrupção do sinal do GPS e então a RNA passa a operar no modo predição. O tempo de predição será de 45 segundos. Considerando as observações da seção 6.2, o número de neurônios ocultos utilizado será 20 e o critério de parada de 40 iterações.

As saídas da RNA correspondentes à posição Leste e Norte serão comparadas com as informações do GPS, aqui tomadas como referência, para gerar um erro de posicionamento durante o período predição. Os erros de predição da RNA serão comparados com os equivalentes de um sistema de navegação convencional, integrado por um filtro de Kalman tipo *Unscented* (FKU) de 15 estados (KUGA et al., 2007), que foi submetido aos mesmos dados e interrupções.

6.3.1 Simulação no intervalo [50s, 500s]

As Figuras 6.22, 6.23 e 6.24 detalham o intervalo [50s, 500s] com relação ao posicionamento e sinais de entrada da RNA. O veículo está estático.

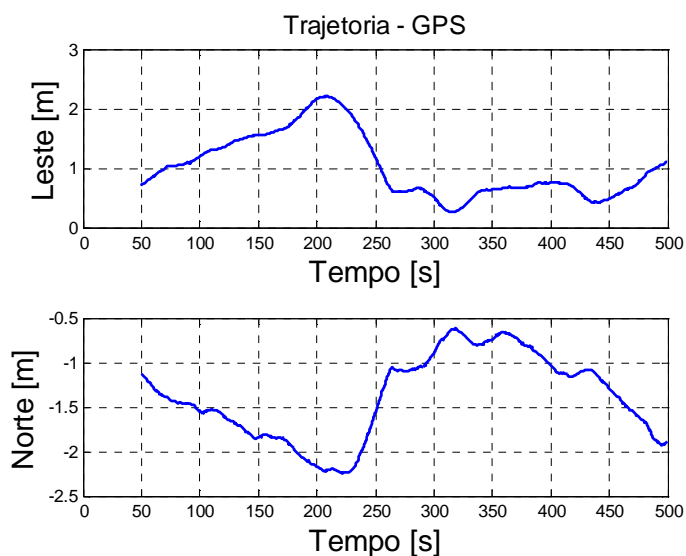


Figura 6.22 – Posição Leste e Norte – GPS.

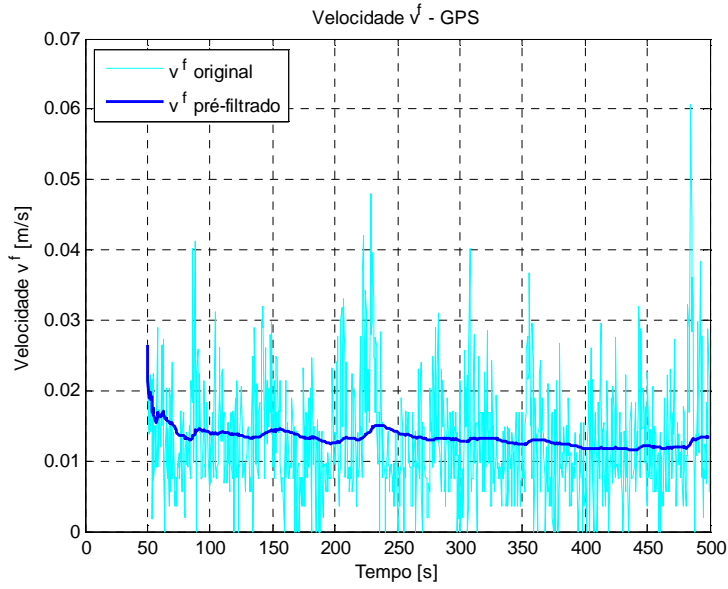


Figura 6.23 – Velocidade v^f – GPS.

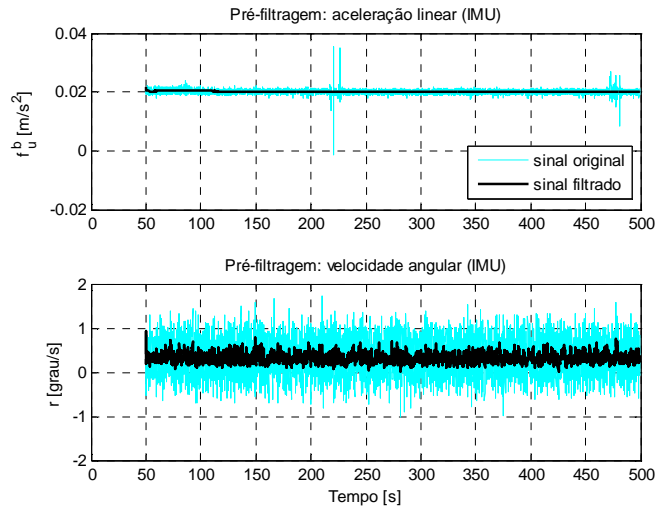


Figura 6.24 – Força específica e velocidade angular – IMU

A simulação começa em torno de $t = 50s$ e o instante do início da predição é $t_{pred} = 155s$. Inicialmente será comparado o efeito da dimensão da janela (20s, 40s, 60s) por método. Para uma comparação mais direta, considere o erro de posição, no período de predição, do veículo no plano Leste-Norte, na seguinte forma:

$$e_{pred} = \sqrt{e_{pred_L}^2 + e_{pred_N}^2} \quad (6.1)$$

onde e_{pred_L} denota erro de posição Leste, e e_{pred_N} erro de posição Norte.

As Figuras 6.25 e 6.26 mostram os resultados médios dos quatro métodos, obtidos de 5 partidas independentes, para o início da predição em $t_{\text{pred}} = 155\text{s}$.

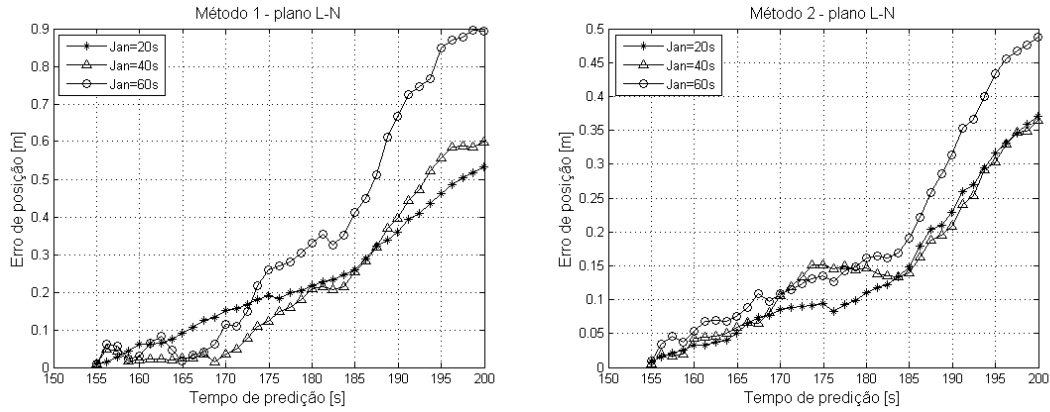


Figura 6.25 – Erro de predição $t_{\text{pred}} = 155\text{s}$. a) método 1, b) método 2.

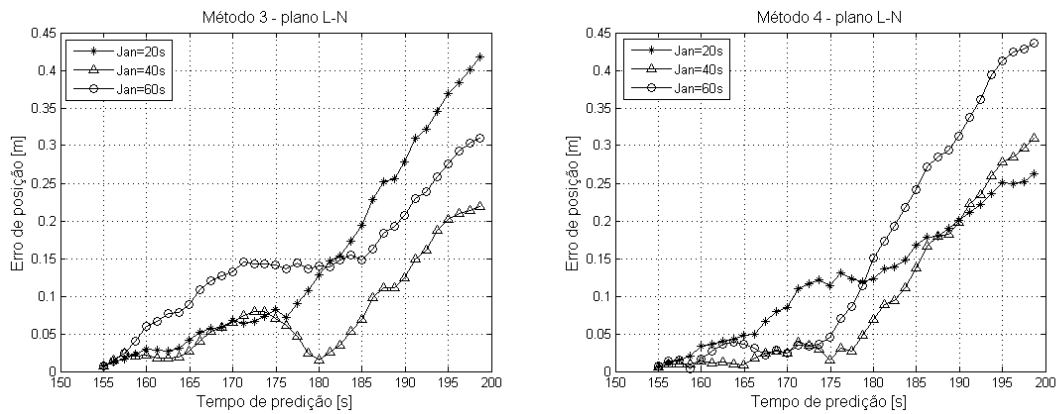


Figura 6.26 - Erro de predição $t_{\text{pred}} = 155\text{s}$. a) método 3, b) método 4.

A seguir os métodos são comparados entre si, juntamente com o erro de predição do FKSP, para uma janela de 20s. As Figuras 6.27 a) e b) mostram, respectivamente, os erros de predição para a posição Leste e Norte, enquanto que a Figura 6.28 mostra o erro de predição no plano Leste-Norte.

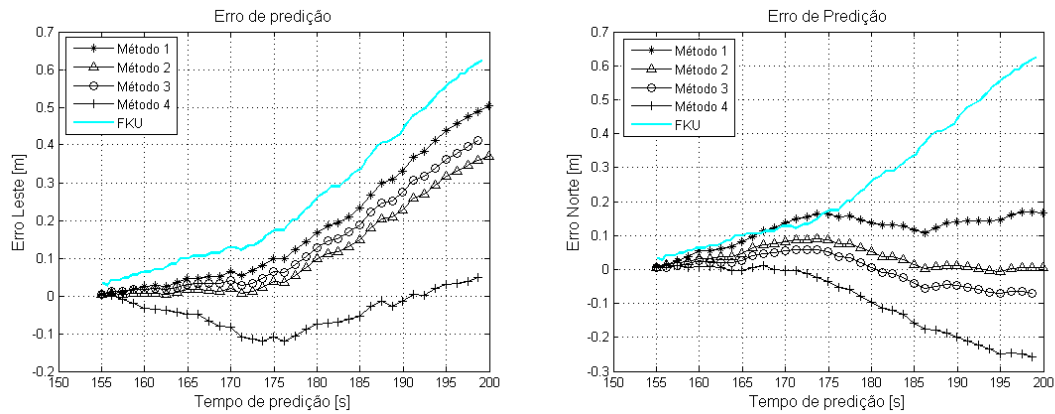


Figura 6.27 – Erros de predição $jan = 20s$, $t_{pred} = 155s$. a) Leste, b) Norte.

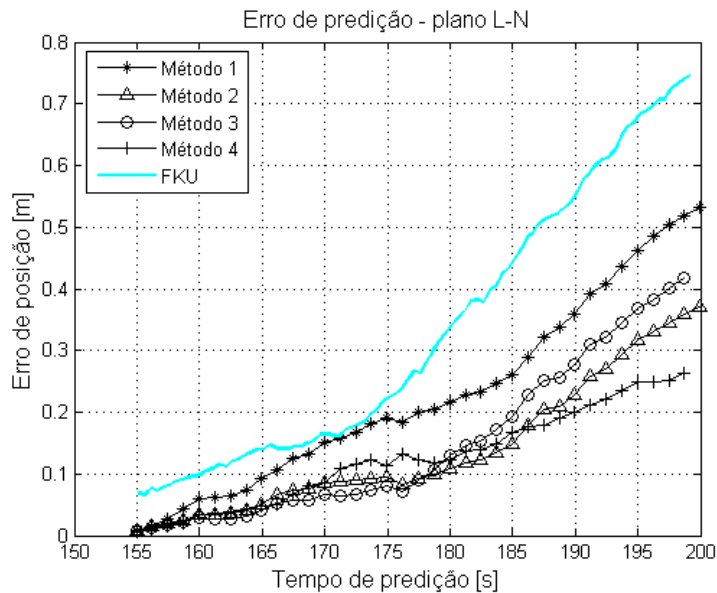


Figura 6.28 - Erros de predição $jan = 20s$, $t_{pred} = 155s$, plano L-N.

Deve ser observado que a simulação começa em torno de $t = 50s$. Como o tamanho do conjunto de treinamento é de $20s$, o método 4 começa a filtrar individualmente os pares de treinamento em $t = 70s$ aproximadamente, logo após o primeiro conjunto ser completado. Assim, o tempo total de “filtragem” até o bloqueio simulado do GPS, é próximo de $85s$. Para os métodos 1 até 3, o número de “janelas” treinadas a mais, quando comparado com o método 4, foi de quatro, até o bloqueio simulado em $t_{pred} = 155s$. A Figura 6.29, compara, na escala de tempo, os quatro métodos.

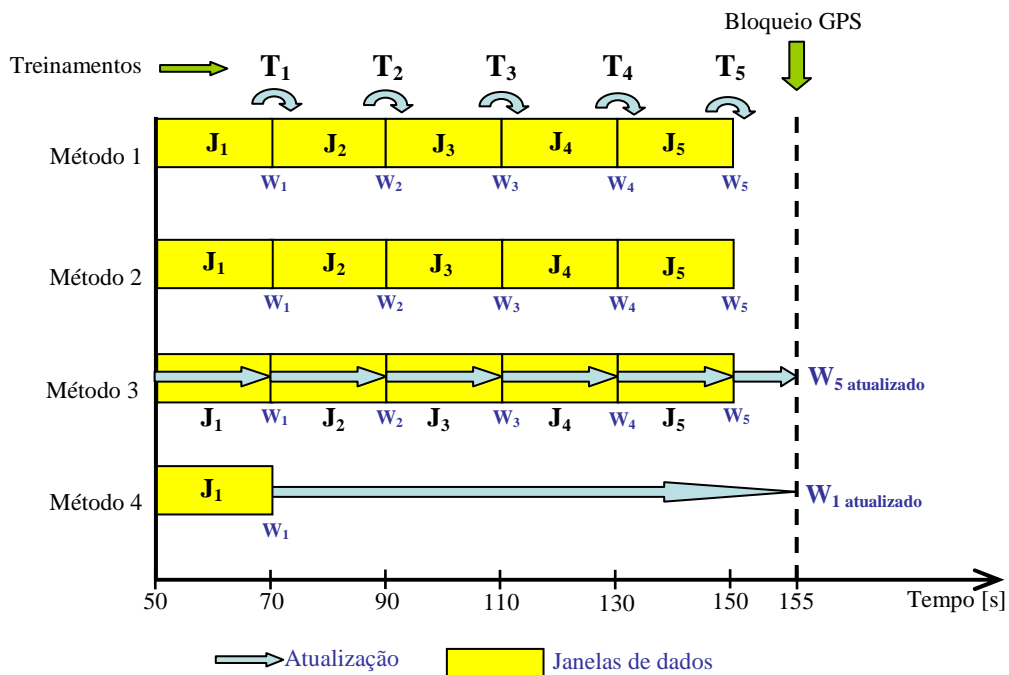


Figura 6.29 – Diagrama dos treinamentos para $t_{pred} = 155s$.

Para o veículo estático os erros de predição não variaram muito entre os métodos de 1 até 4, ficando abaixo de 0,5m; enquanto que para o FKU em torno de 0,8m, após 45s de predição. O tamanho da janela parece não ter influenciado no resultado, sugerindo que as diferentes janelas foram capazes de fornecer as informações necessárias para um aprendizado similar da RNA. Talvez, o fato dos pares de treinamento (entradas/saídas) não variarem muito nesta situação estática, tenha contribuído para este desempenho.

6.3.2 Simulação no intervalo [630s, 1000s]

As Figuras 6.30, 6.31 e 6.32 detalham o intervalo [630s, 1000s] com relação ao posicionamento e sinais de entrada da RNA. O veículo está em movimento.

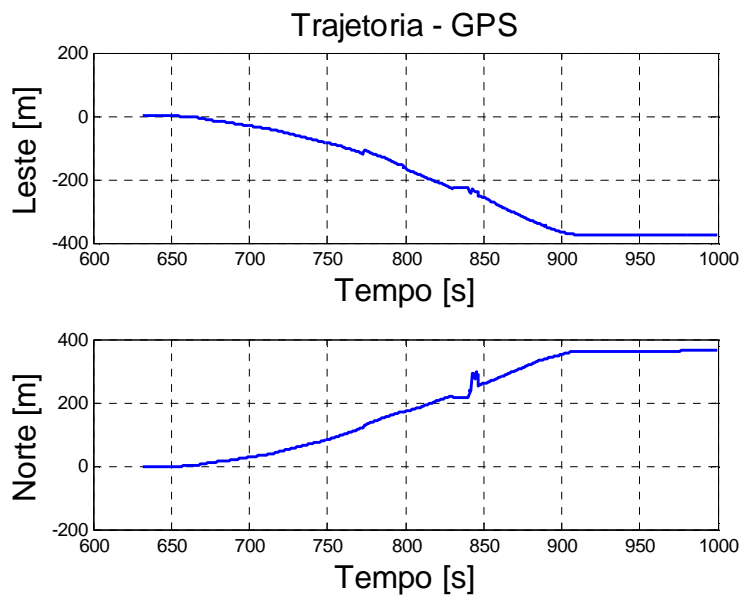


Figura 6.30 - Posição Leste e Norte – GPS.

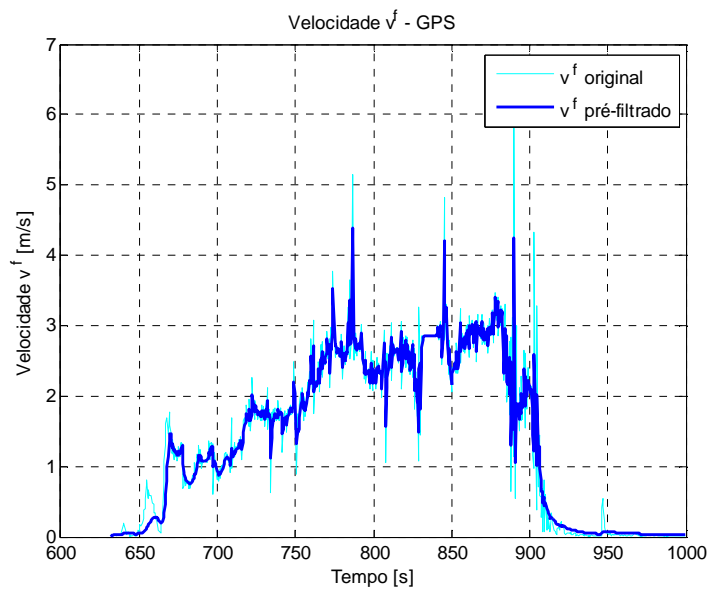


Figura 6.31 - Velocidade v^f – GPS.

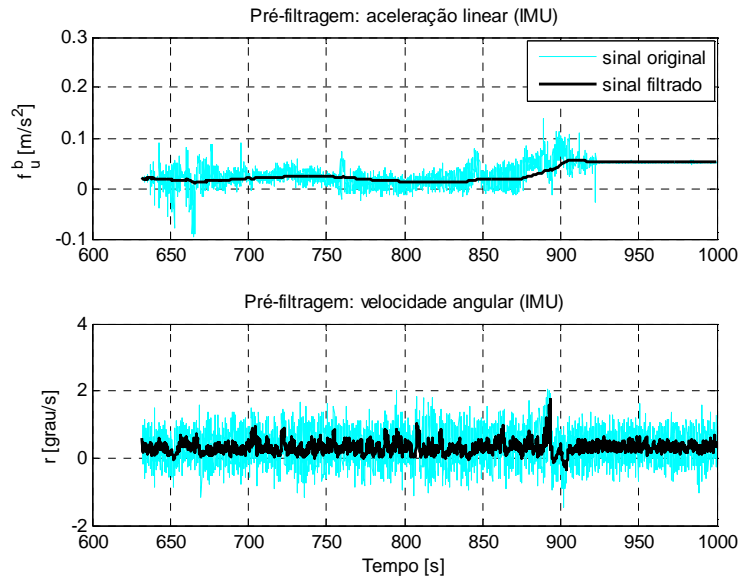


Figura 6.32 – Força específica e velocidade angular – IMU.

A simulação começa em torno de $t = 630s$, com o início da predição em $t_{pred} = 705s$ e $t_{pred} = 755s$.

O procedimento utilizado é o mesmo da seção 6.3.1. As Figuras 6.33 e 6.34 mostram os resultados médios dos quatro métodos, obtidos de 5 partidas independentes, para o início da predição em $t_{pred} = 705s$.

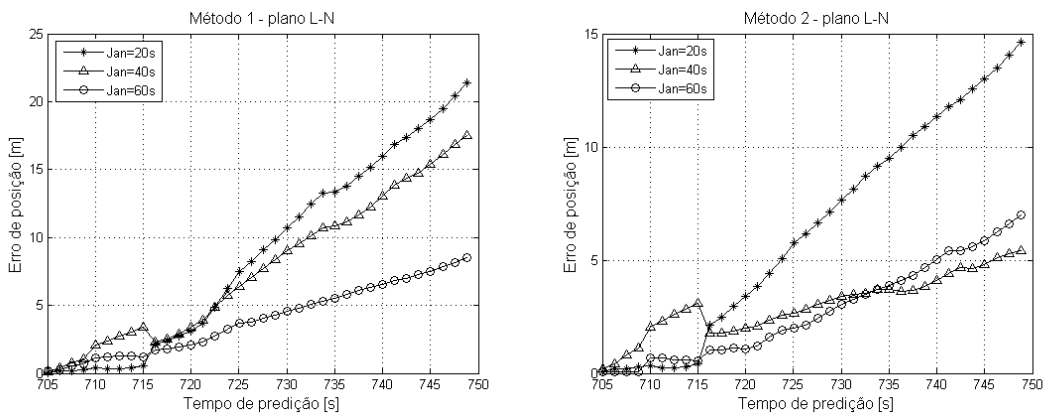


Figura 6.33 - Erro de predição $t_{pred} = 705s$. a) método 1, b) método 2.

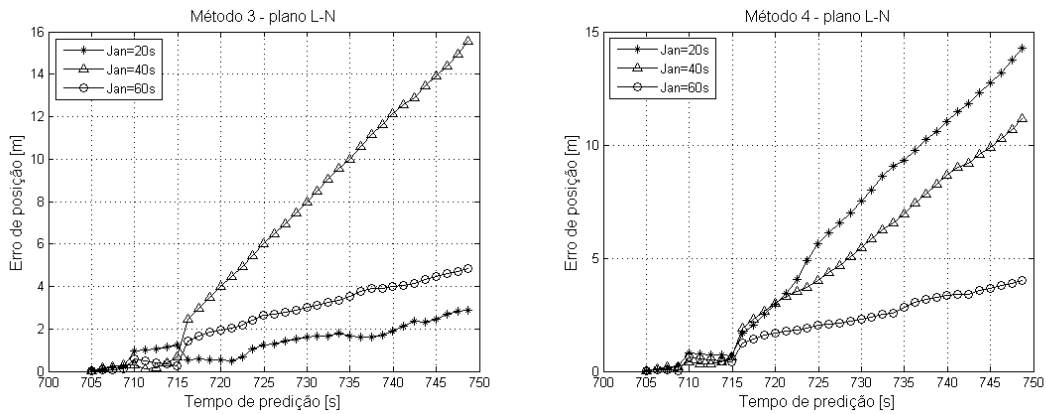


Figura 6.34 - Erro de predição $t_{pred} = 705s$. a) método 3, b) método 4.

Com o veículo em movimento, o erro de predição da RNA com as janelas de 60s tem, no geral, um melhor desempenho entre os métodos, ficando sempre abaixo de 10m, enquanto que para o FKU o erro de predição fica em torno de 70 m no plano Leste-Norte.

Novamente os métodos são comparados entre si, juntamente com o erro de predição do FKU, para uma janela de 60s. As Figuras 6.35 a) e b) mostram, respectivamente, os erros de predição para a posição Leste e Norte, enquanto que a Figura 6.36 mostra o erro de predição no plano Leste-Norte.

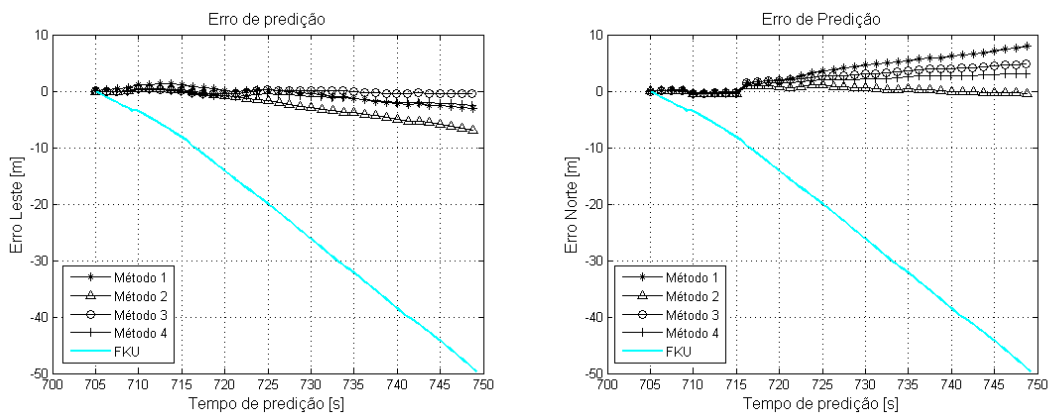


Figura 6.35 - Erros de predição jan = 60s, $t_{pred} = 705s$. a) Leste, b) Norte.

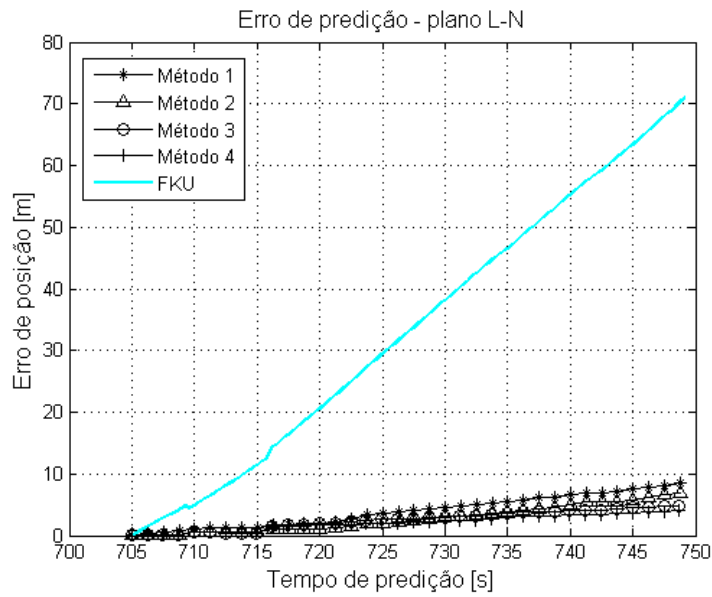


Figura 6.36 - Erros de previsão $t_{pred} = 705s$, plano L-N.

A simulação é repetida para o início da previsão em $t_{pred} = 755s$. As Figuras 6.37 e 6.38 mostram os resultados médios dos quatro métodos, obtidos de 5 partidas independentes.

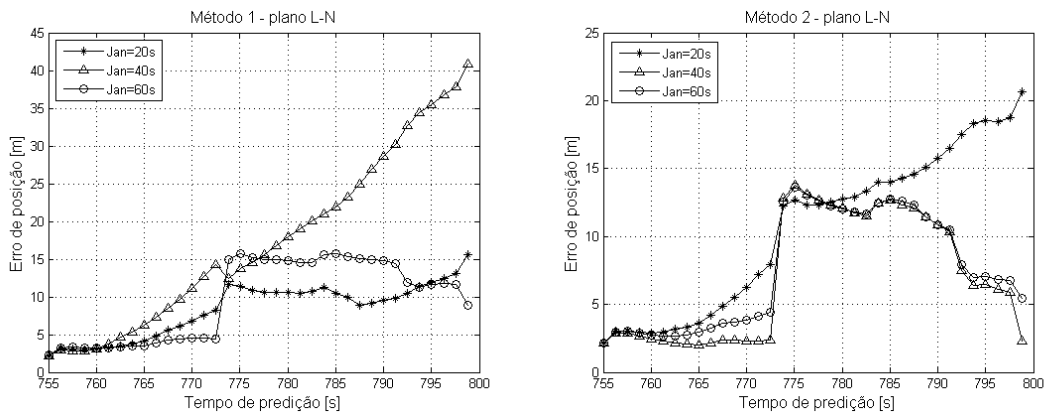


Figura 6.37 - Erro de previsão $t_{pred} = 755s$. a) método 1, b) método 2.

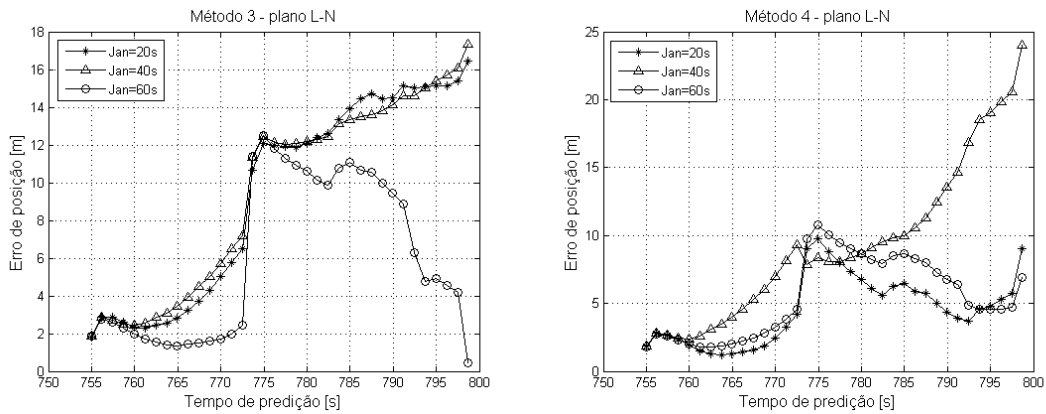


Figura 6.38 - Erro de predição $t_{pred} = 755s$. a) método 3, b) método 4.

Repetindo a característica da simulação anterior, o erro de predição da RNA com as janelas de 60s tem, no geral, um melhor desempenho entre os métodos, ficando sempre abaixo de 20m, enquanto que para o FKU o erro de predição fica em torno de 110 m no plano Leste-Norte.

Novamente os métodos são comparados entre si, juntamente com o erro de predição do FKU, para uma janela de 60s. As Figuras 6.39 a) e b) mostram, respectivamente, os erros de predição para a posição Leste e Norte, enquanto que a Figura 6.40 mostra o erro de predição no plano Leste-Norte.

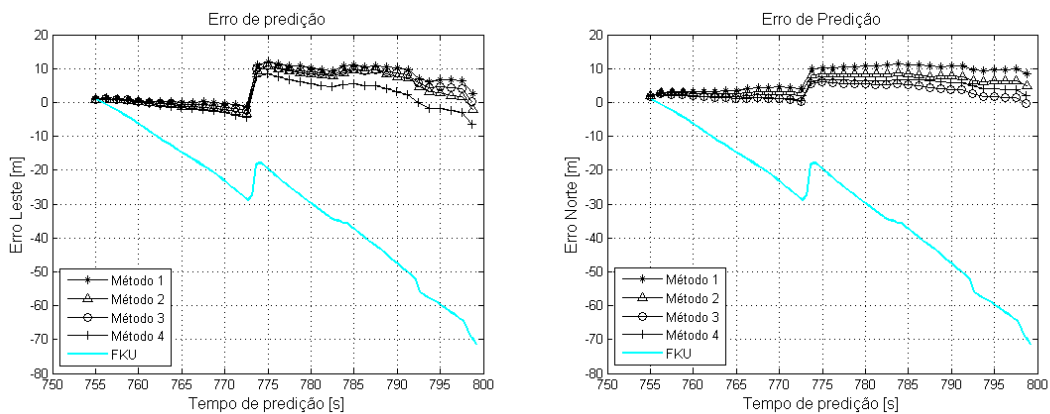


Figura 6.39 - Erros de predição jan = 60s, $t_{pred} = 755s$. a) Leste, b) Norte.

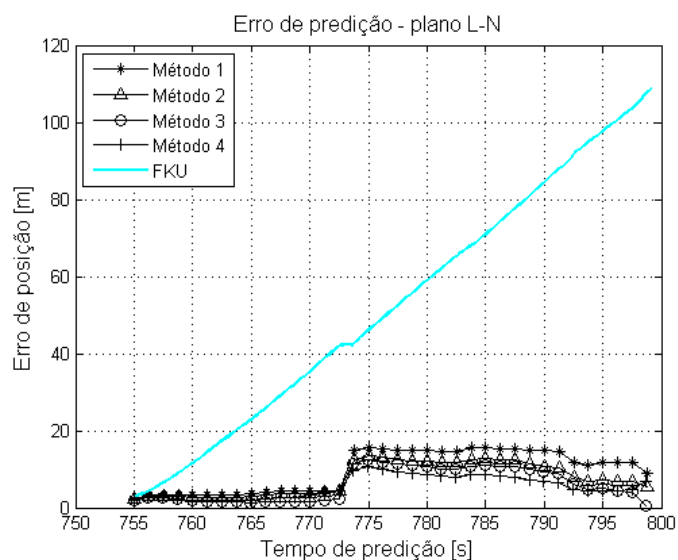


Figura 6.40 - Erros de predição $\text{jan} = 60\text{s}$, $t_{\text{pred}} = 705\text{s}$, plano L-N.

6.3.3 Simulação com um segundo conjunto de dados

Dados de uma segunda campanha de aquisição foram utilizados nas mesmas simulações das seções anteriores. A segunda campanha foi realizada em data diferente, num outro veículo, de pequeno porte e uso urbano, e foi utilizada uma trajetória similar à primeira. A unidade de medida inercial e receptor GPS utilizados foram os mesmos, porém os dados do receptor GPS foram adquiridos numa frequência de 1Hz, lembrando que na primeira campanha o receptor estava ajustado para 2Hz. As Figuras 6.41, 6.42 e 6.43 detalham as informações utilizadas.

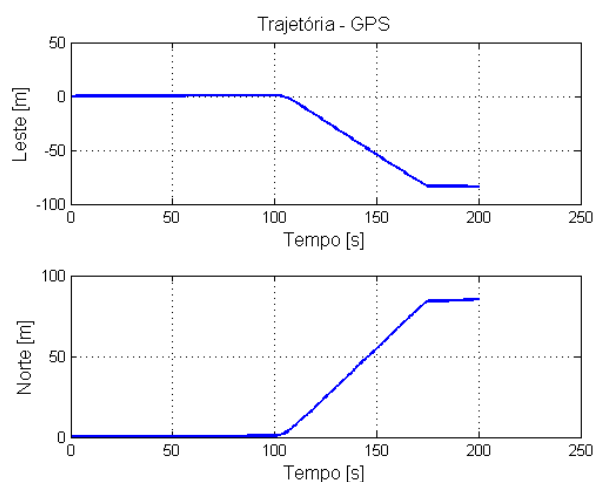


Figura 6.41 – Deslocamento Leste-Norte fornecido pelo GPS.

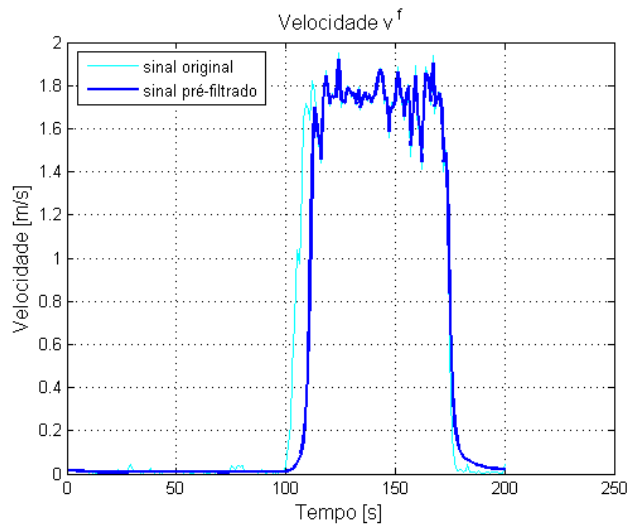


Figura 6.42 – Velocidade v^f utilizada na entrada da RNA.

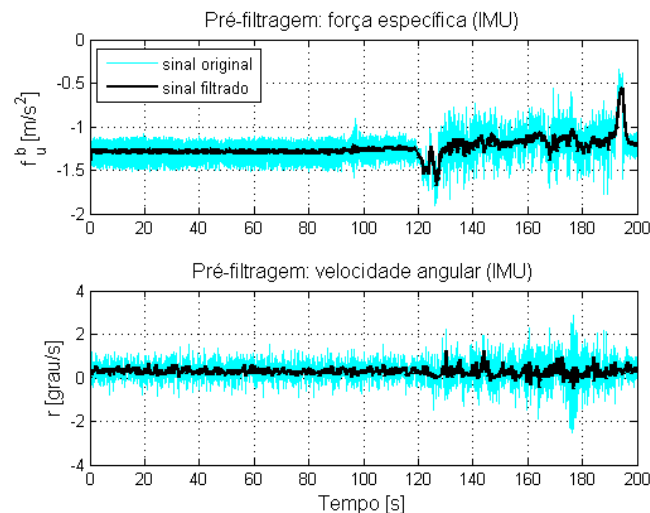


Figura 6.43 – Sinais da IMU utilizados na entrada da RNA.

Um conjunto de dados, no intervalo [90s, 150s], foi utilizado para se obter informações iniciais sobre a influência do número de neurônios ocultos e iterações de treinamento na RNA. O procedimento usado foi o mesmo da seção 6.2.3. As Figuras 6.44 e 6.45 mostram os resultados médios obtidos, para cinco partidas independentes. Pode-se observar que a partir de 20 neurônios o índice R não melhora, e que não há melhora significativa do índice R e erros de treinamento a partir de 30 iterações, resultado similar aos obtidos na simulação com os dados da primeira campanha. Para efeito de comparação com as simulações anteriores serão adotados 20 neurônios ocultos, 40 iterações como critério de parada, e as mesmas janelas de 20s, 40s e 60s de duração.

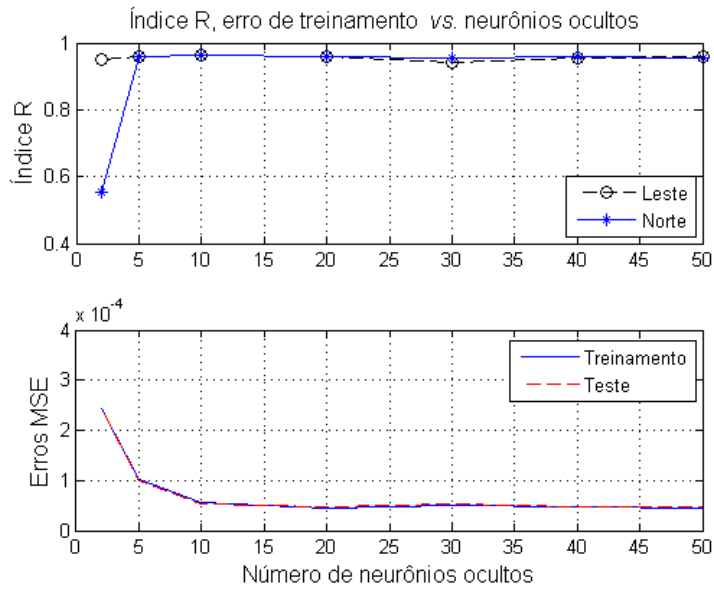


Figura 6.44 – Índice R e erros de treinamento vs. neurônios ocultos.

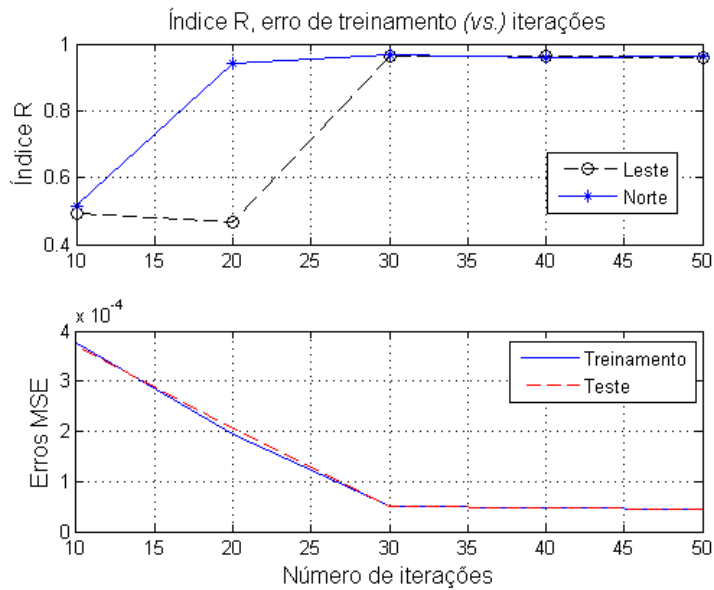


Figura 6.45 – Índice R e erros de treinamento vs. iterações.

As simulações de predição tiveram início em $t = 0s$. A primeira simulação teve início com o veículo estático, e o bloqueio do GPS foi simulado no instante $t_{pred} = 65s$. As Figuras 6.46 e 6.47 mostram os resultados médios, obtidos de 5 partidas independentes.

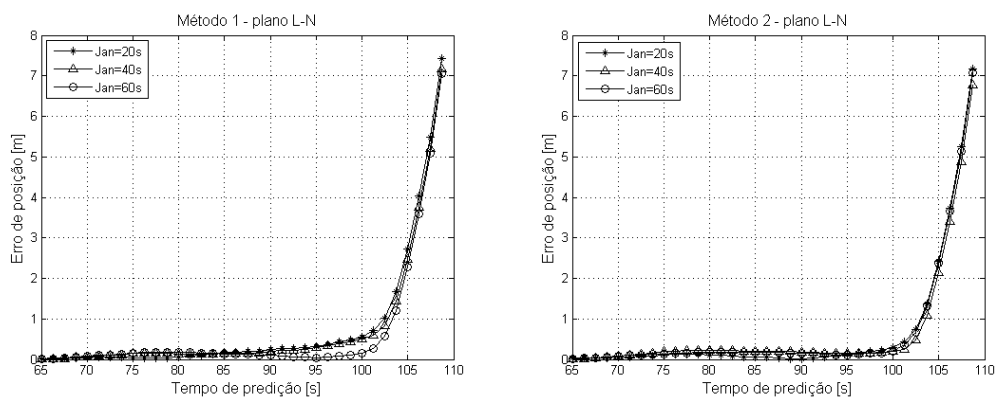


Figura 6.46 - Erro de predição $t_{pred} = 65s$. a) método 1, b) método 2.

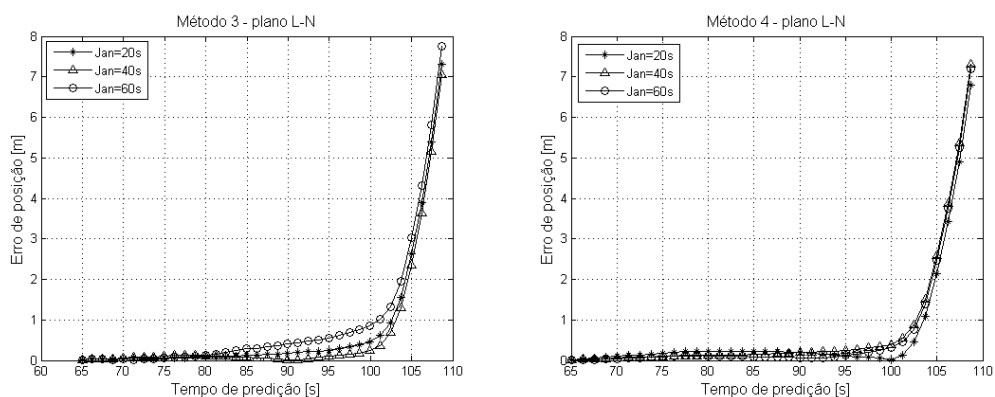


Figura 6.47 - Erro de predição $t_{pred} = 65s$. a) método 3, b) método 4.

A seguir os métodos, de 1 até 4, foram comparados entre si, juntamente com o erro de predição do FKU, para uma janela de 20s. As Figuras 6.48 a) e b) mostram, respectivamente, os erros de predição para a posição Leste e Norte, enquanto que a Figura 6.49 mostra o erro de predição no plano Leste-Norte.

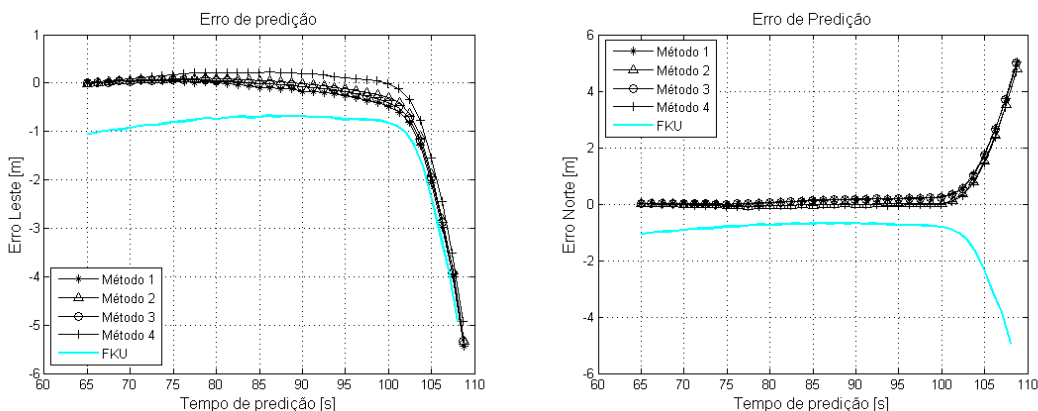


Figura 6.48 - Erros de predição $jan = 20s$, $t_{pred} = 65s$. a) Leste, b) Norte.

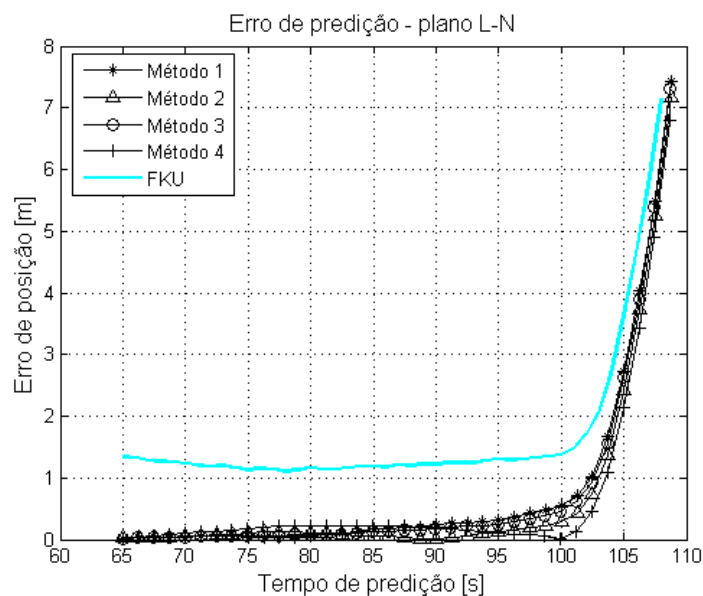


Figura 6.49 - Erros de predição $\text{jan} = 20\text{s}$, $t_{\text{pred}} = 65\text{s}$, plano L-N.

Para o veículo estático, os erros de predição não variaram muito entre os métodos 1 até 4, bem como não houve muita influência por parte das diferentes janelas. O comportamento da simulação foi similar à anterior, com dados da primeira campanha e veículo estático, até aproximadamente o instante $t = 100\text{s}$, a partir do qual os erros crescem de forma mais acentuada.

Diferente da primeira campanha, na qual o veículo permanecia estático durante todo intervalo de predição, nesta segunda campanha o veículo se movimenta durante a predição, como pode ser observado nas Figuras 6.41 e 6.42, a partir de $t = 100\text{s}$ aproximadamente. Esta alteração, de estático para em movimento, modificou as entradas da RNA e do FKU de uma forma diferente da primeira campanha. Repetindo a simulação para o início da predição em $t_{\text{pred}} = 85\text{s}$, foram obtidos resultados similares, conforme pode ser observado nas Figuras 6.50 e 6.51.

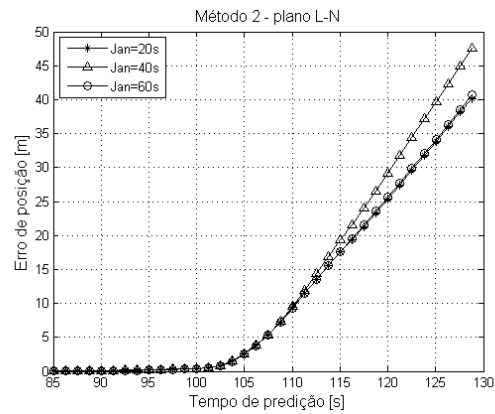
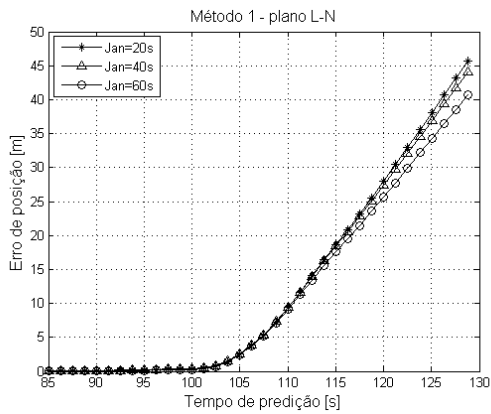


Figura 6.50 – Erro de predição $t_{pred} = 85s$. a) método 1, b) método 2.

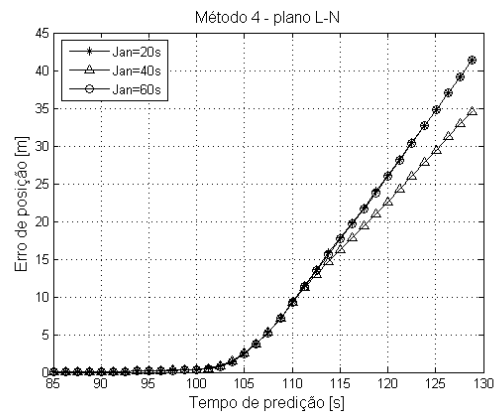
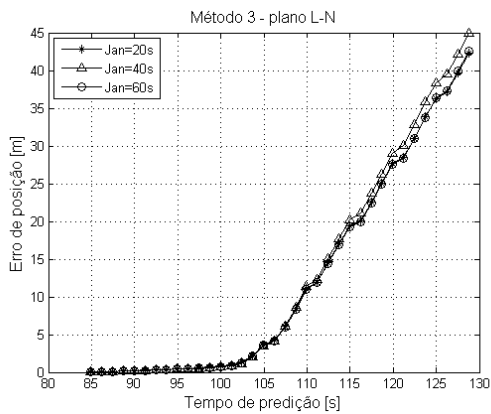


Figura 6.51 – Erros de predição $t_{pred} = 85s$. a) método 3, b) método 4.

A simulação foi repetida com o veículo em movimento, com início da predição em $t_{pred} = 115s$. As Figuras 6.52 e 6.53 mostram os resultados médios, para os quatro métodos, obtidos a partir de 5 partidas independentes.

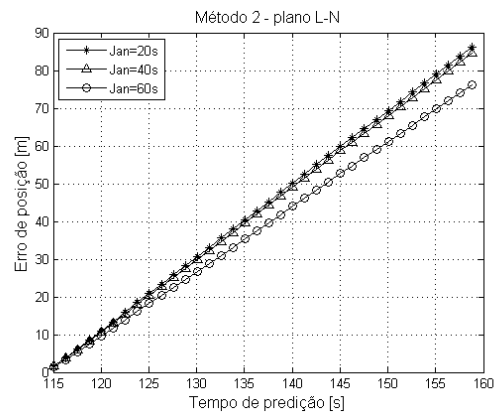
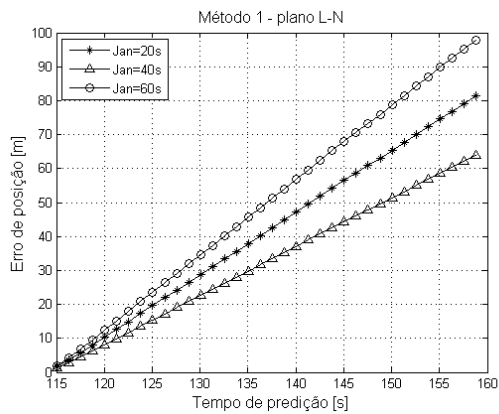


Figura 6.52 - Erro de predição $t_{pred} = 115s$. a) método 1, b) método 2.

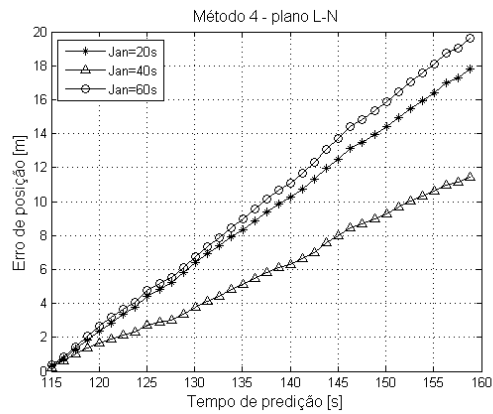
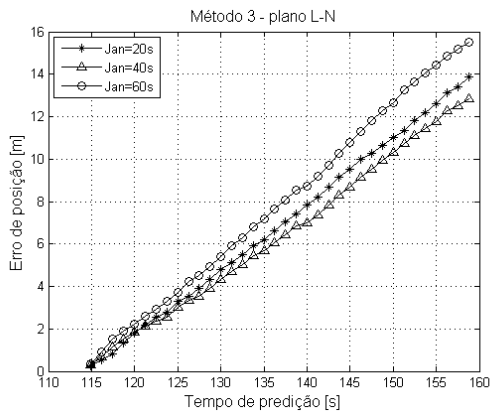


Figura 6.53 - Erros de predição $t_{pred} = 115s$. a) método 3. b) método 4.

Para este instante de predição em particular, a RNA com janela de 40s parece apresentar os melhores resultados junto com os métodos 3 e 4, com erros de predição abaixo de 20m, no plano Leste-Norte. Os métodos foram comparados entre si, juntamente com o erro de predição do FKU, para uma janela de 40s. As Figuras 6.54 a) e b) mostram, respectivamente, os erros de predição para a posição Leste e Norte, enquanto que a Figura 6.55 mostra o erro de predição no plano Leste-Norte.

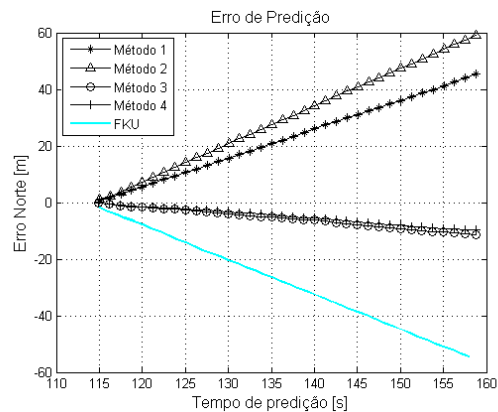
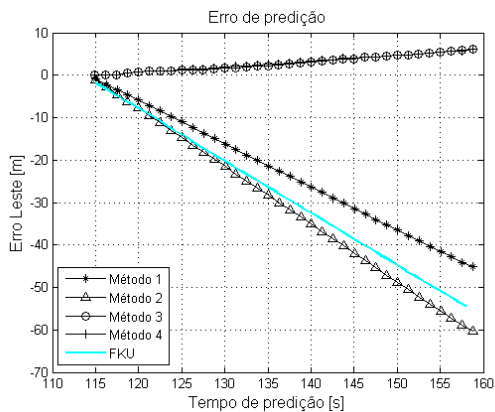


Figura 6.54 - Erros de predição $jan = 40s$, $t_{pred} = 115s$. a) Leste, b) Norte.

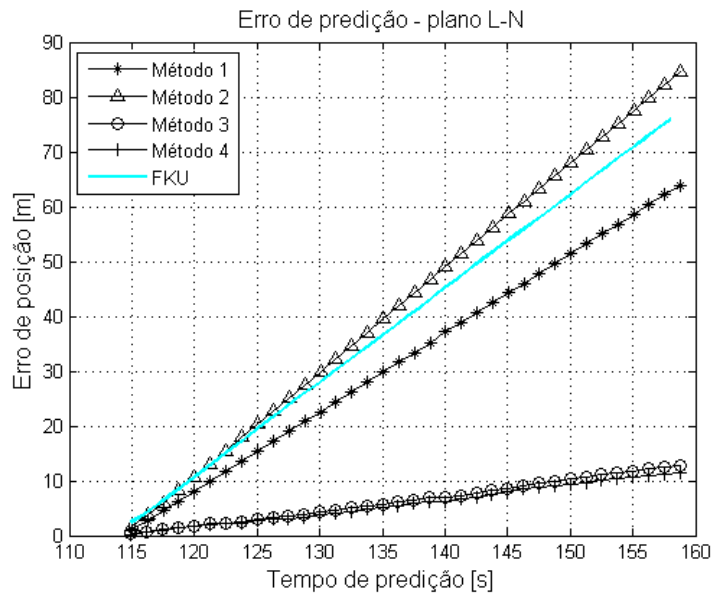


Figura 6.55 - Erros de predição $jan = 40s$, $t_{pred} = 115s$, plano L-N.

A simulação é repetida para o início da predição em $t_{pred} = 140s$. As Figuras 6.56 e 6.57 mostram os resultados médios dos quatro métodos, obtidos de 5 partidas independentes.

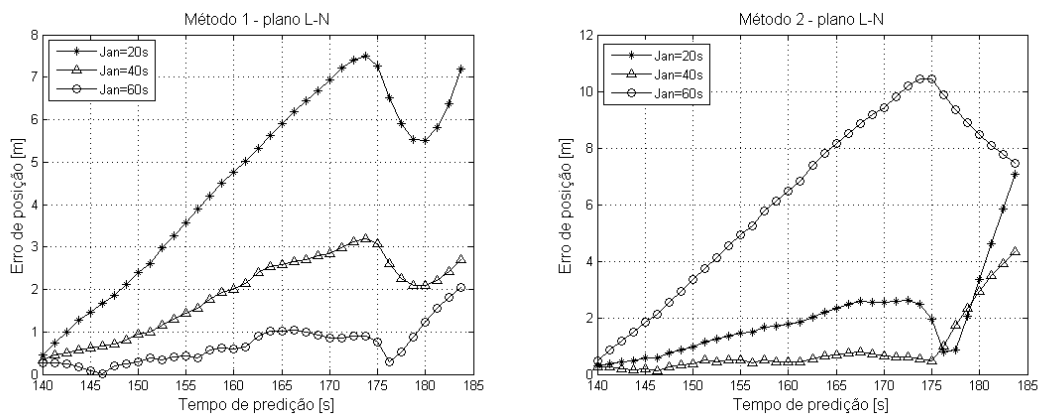


Figura 6.56 - Erro de predição $t_{pred} = 140s$. a) método 1, b) método 2.

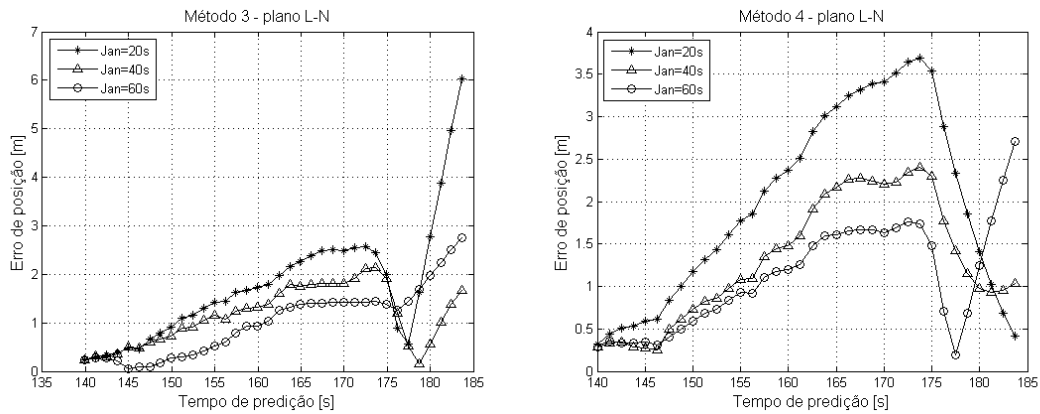


Figura 6.57 - Erros de predição $t_{pred} = 140s$. a) método 3. b) método 4.

Para este instante de predição em particular, a RNA com as janelas de 40s e 60s parece apresentar os melhores resultados, com erros menores que 10m, no plano Leste_Norte. Os métodos foram comparados entre si, juntamente com o erro de predição do FKU, para uma janela de 60s. As Figuras 6.56 a) e b) mostram, respectivamente, os erros de predição para a posição Leste e Norte, enquanto que a Figura 6.57 mostra o erro de predição no plano Leste-Norte.

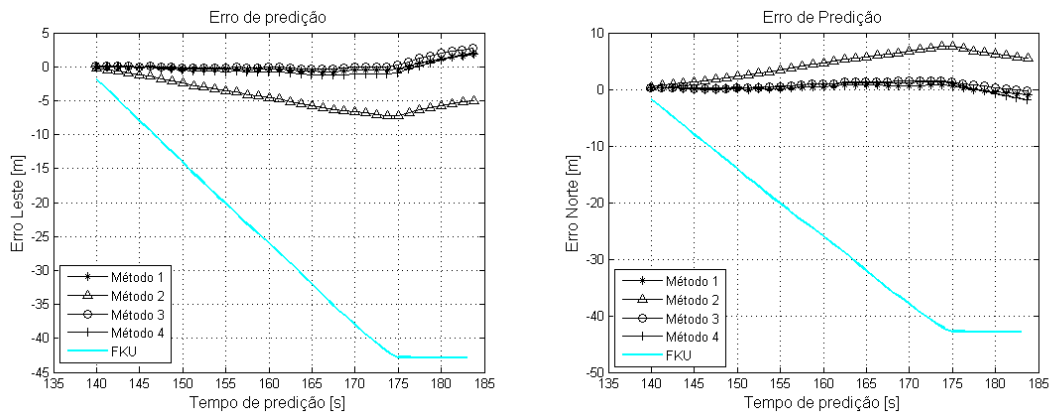


Figura 6.58 - Erros de predição jan = 60s, $t_{pred} = 140s$. a) Leste, b) Norte.

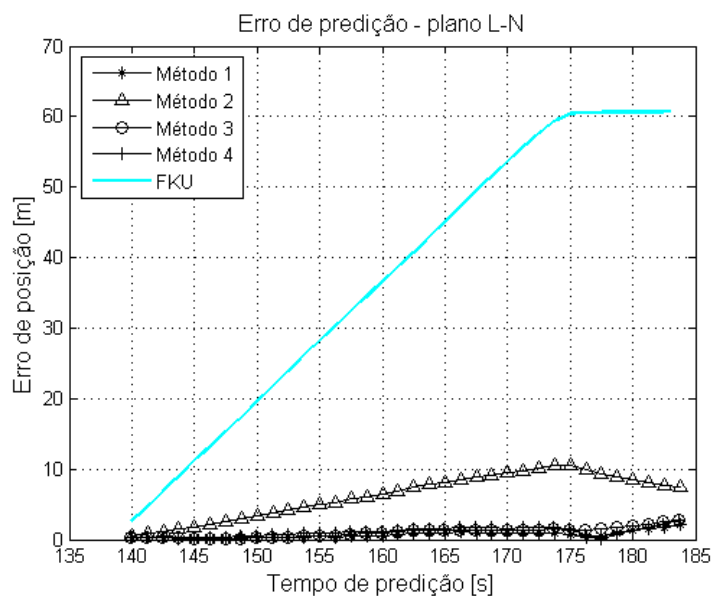


Figura 6.59 - Erros de predição $jan = 60s$, $t_{pred} = 140s$, plano L-N.

Ao final das simulações pode-se observar que, de uma forma geral, os menores erros de predição foram obtidos pela RNA treinada pelos métodos 3 e 4 e com janelas de 40 e 60s de dados de treinamento. Quando comparados com o FKU, os erros de predição apresentados pela RNA foram menores, sugerindo uma maior capacidade de modelar a cinemática do veículo nas condições de simulação apresentadas.

6.3.4 Simulações complementares

O modelo da RNA operando no modo predição, conforme apresentado no Capítulo 5, mostra a realimentação da velocidade v^f obtida a partir dos incrementos Δv^f , como mostram as Equações 5.10 e 5.11.

Nas redes alimentadas adiante (*feedforward*), as saídas são funções somente das entradas para um determinado instante, portanto, para entradas constantes, as saídas são também constantes. Porém, a realimentação de v^f faz com que as saídas da RNA tenham dependência temporal. Para uma dada entrada e a partir de uma saída inicial da RNA conhecida, sua resposta pode convergir ou adquirir características instáveis (HAGAN et al., 1995).

Com o objetivo de observar a saída da RNA proposta, e operando no modo predição, foi efetuada a seguinte simulação. Para o veículo estático no intervalo [50s, 600s], dos dados da primeira campanha, foi escolhido aleatoriamente um subconjunto para treinamento e teste com aproximadamente 9% dos dados disponíveis. Dos dados do subconjunto (1208 pares de treinamento), 70% foram utilizados para treinamento e 30% para verificação. A Figura 6.60 mostra os erros de treinamento e teste para o subconjunto.

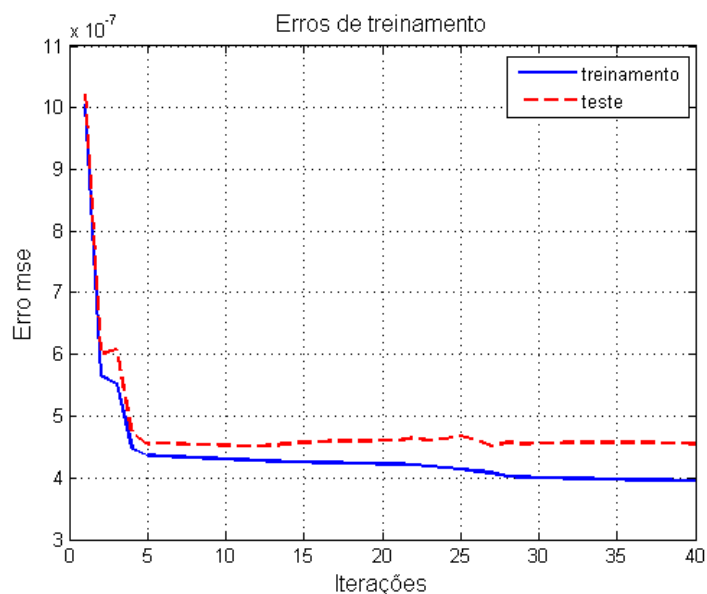


Figura 6.60 – Erros de treinamento para o subconjunto.

Em seguida foram apresentados seqüencialmente à RNA, treinada e no modo predição, todos os dados disponíveis no intervalo [50s, 600s], num total de 10986 entradas. A solução de navegação simulada está mostrada nas Figuras 6.61 até 6.64, na qual os erros, em relação ao GPS, estão mostrados.

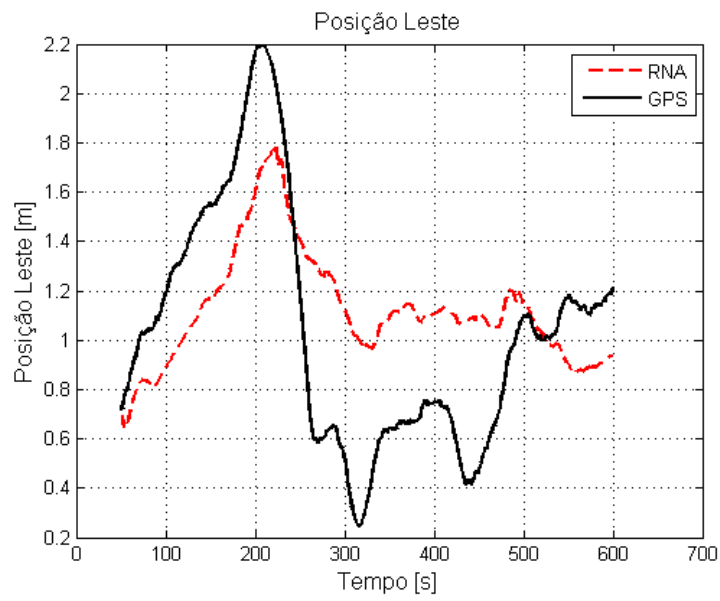


Figura 6.61 – Posição Leste simulada e GPS [50s, 600s].

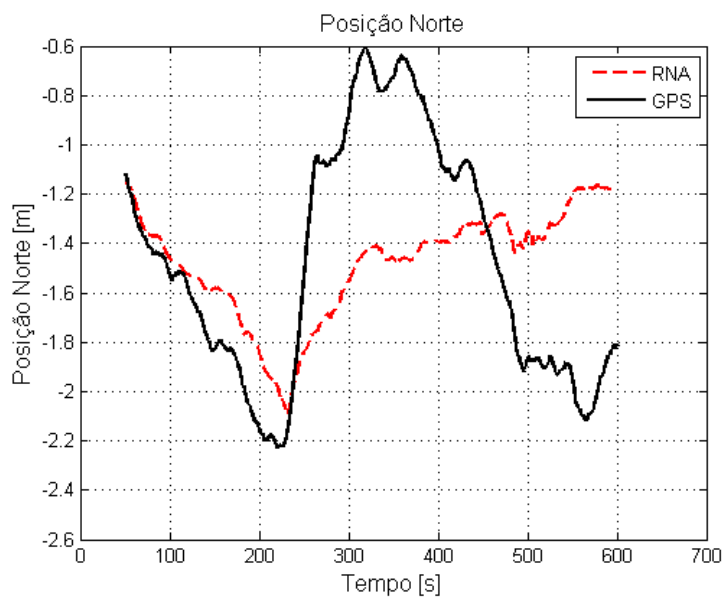


Figura 6.62 - Posição Norte simulada e GPS [50s, 600s].

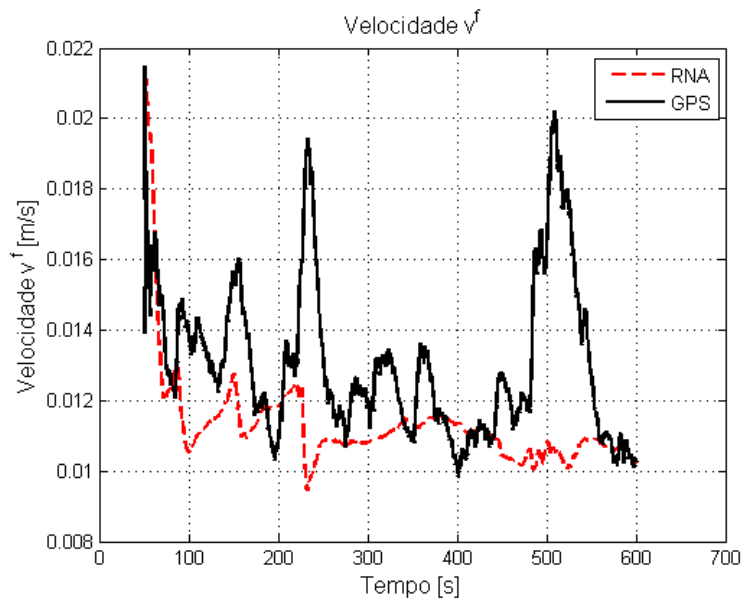


Figura 6.63 – Velocidade v^f simulada e GPS [50s, 600s].

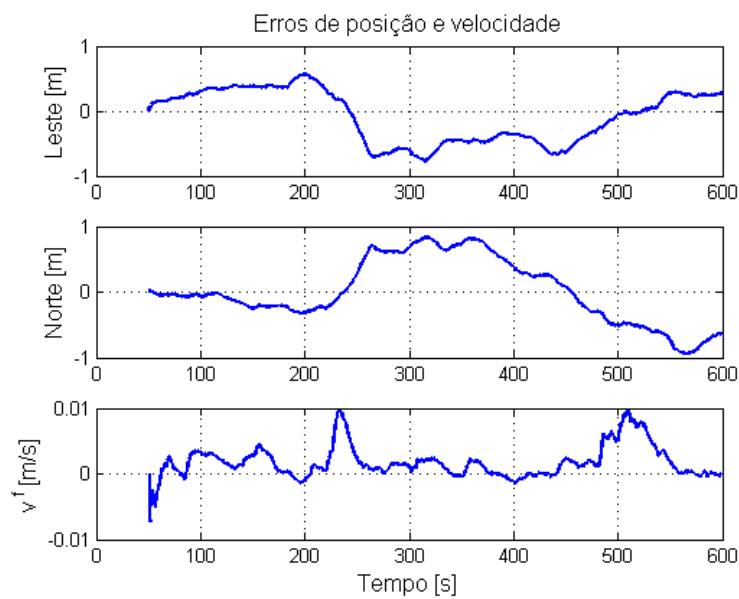


Figura 6.64 – Erros de posição e velocidade [50s, 600s].

Pode ser observado na Figura 6.64 que os erros de posição e velocidades, com relação ao GPS, são limitados entre valores máximos e mínimos, não apresentando características divergentes. Lembrando que, apesar do veículo estar estático, as entradas da RNA são ruidosas e também limitadas, portanto os resultados sugerem um comportamento estável da RNA no modo predição. Esta simulação não esgota o

assunto. Uma possibilidade de análise do comportamento de RNA recorrentes, com relação à sua estabilidade, pode ser realizada através da técnica de Lyapunov (HAGAN et al., 1995), porém não será abordada neste trabalho.

A velocidade v^f faz parte da modelagem da RNA, conforme apresentado nos Capítulos 2 e 5. A importância de v^f pode ser observada pela sua remoção, como sinal de entrada e seu respectivo incremento na saída da RNA e, repetindo-se a simulação da seção 6.2, com o método do filtro de Kalman adaptativo. Neste caso as entradas da RNA são compostas pelo sinal do acelerômetro e giroscópio e a saída pelos incrementos de posição Leste e Norte.

A Figura 6.65 mostra os valores médios, para 5 partidas independentes, dos valores do índice R em função do número de iterações, para 20 neurônios ocultos; enquanto que a Figura 6.66 mostra o índice R em função dos neurônios ocultos, para 30 iterações. Podem ser observados baixos valores do índice R, indicando baixa generalização; e valores mais elevados dos erros de treinamento quando comparados com aqueles da seção 6.2 na Figura 6.8, bem como um comportamento oscilatório.

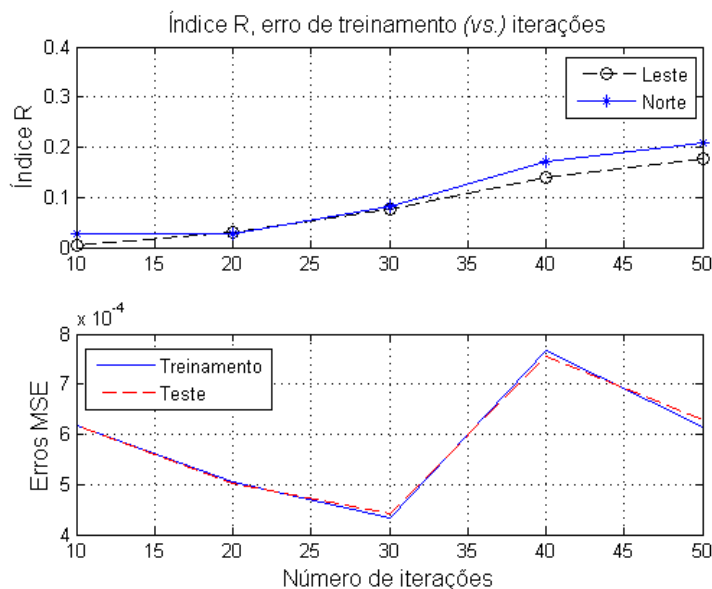


Figura 6.65 – Índice R vs. iterações.

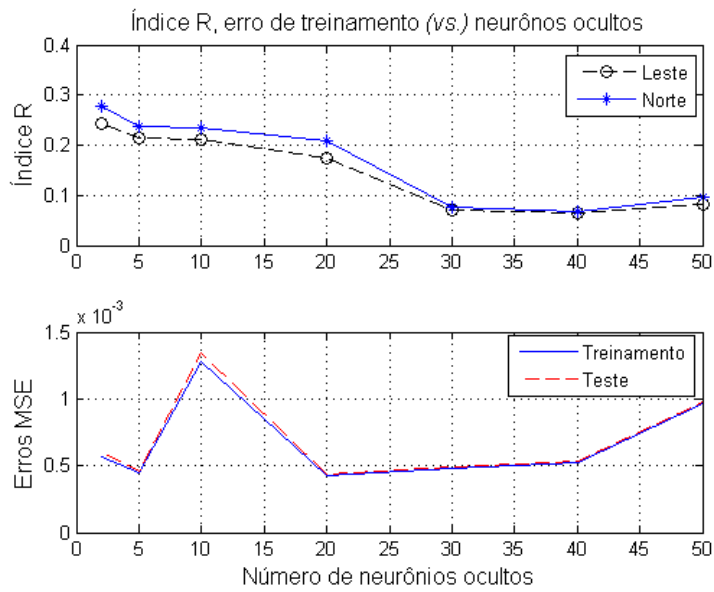


Figura 6.66 – Índice R vs. neurônios ocultos.

A opção pela entrada realimentada de v^f tem como objetivo a simplificação do modelo e trabalhar somente com uma RNA. Uma alternativa poderia ser a substituição da realimentação por uma RNA dedicada a aprender somente v^f (CHIANG; HUANG, 2008), com o custo de se treinar uma RNA a mais.

7 COMENTÁRIOS FINAIS E CONCLUSÕES

7.1 Introdução

Este trabalho investigou a possibilidade da aplicação de redes neurais artificiais, do tipo perceptron de múltiplas camadas, para compensação de erros de unidades inerciais, numa aplicação de posicionamento veicular no sistema de navegação LTP (*Local Tangent Plane*), com coordenadas no plano Norte-Leste. A rede neural deve ser treinada enquanto houver informações via GPS, e operar predizendo as coordenadas de posição, na ausência do GPS, com informações provenientes de uma unidade inercial de baixo custo, com sensores MEMS. Tradicionalmente, a integração de sistemas de navegação inercial com GPS é feita usando técnicas de estimação estocástica.

Optou-se por redes neurais do tipo MLP pelo fato destas possuírem uma estrutura simples, e também aceitarem vários métodos de treinamento. São também capazes de resolver problemas de previsão de maneira semelhante a um problema de aproximação de funções, no qual a informação temporal é emulada por atrasos de dados, as chamadas janelas de dados.

Inicialmente, foi apresentada a modelagem da rede neural, ou seja, a escolha das variáveis de entrada e saída, baseada num conjunto simplificado de equações de navegação inercial. O modelo resultou na utilização da leitura de um acelerômetro (força específica), um giroscópio (velocidade angular) e da velocidade na direção preferencial do veículo. A escolha de um modelo simplificado com uma modelagem mais simples, com menos variáveis, permitiu colocar o foco no treinamento e análise da rede neural. Esta situação é a de navegação de veículos terrestres em local plano e com velocidade (baixa) controlada, justificando as hipóteses simplificadoras para as equações inerciais.

Foi proposta uma metodologia para o treinamento da RNA, usando a abordagem de um filtro de Kalman adaptativo, que dentre outras características, aceita a apresentação de

dados tanto na forma sequencial como em lotes. Os algoritmos e métodos propostos foram testados, por simulação numérica com o software MatLab[™].

7.2 Análise dos resultados e conclusões

Inicialmente, o algoritmo de treinamento usando a abordagem do filtro de Kalman adaptativo (FKA), foi comparado com o tradicional método Levenberg-Marquardt (LM), num problema de aproximação de função com uma janela de dados de aproximadamente 83s. Os resultados mostraram erros médios de posicionamento ligeiramente maiores para o FKA, porém com a mesma ordem de grandeza e a vantagem da apresentação sequencial de dados, uma vez que o método LM só aceita a apresentação em lote. O efeito da solução adaptativa no treinamento e da pré-filtragem de dados foi analisado com a comparação direta com um filtro de Kalman sem o módulo de adaptação e a apresentação dos dados com e sem pré-filtragem.

Foram propostos e avaliados quatro métodos de treinamento para a rede neural dedicada ao problema de posicionamento. A avaliação foi feita com informações obtidas em duas campanhas independentes de coleta de dados. Foram simuladas ausências de GPS em vários instantes, com o veículo estático e em movimento. Os resultados, na forma de erros de posicionamento, foram comparados com aqueles gerados por um sistema IMU/GPS integrado, de forma convencional, por um filtro de Kalman *Unscented*. Com relação às simulações, as principais conclusões que podem ser feitas são:

- Na maioria das situações simuladas, os erros de posicionamento apresentados pela rede neural sugerem que, depois de treinada, foi capaz de aprender a cinemática desenvolvida pelo veículo, no intervalo de uma determinada janela, de uma forma mais eficiente que a modelagem do FKU.
- Com relação ao aprendizado da cinemática do veículo, existe uma relação de dependência entre o tamanho das janelas e o tipo de movimento que o veículo faz. Na situação estática, ou de velocidade constante, a dimensão da janela parece não ser um fator preponderante. Mas, nas situações de alteração de

velocidade ou curvas, as janelas de maior duração possuem maior chance de captar essas variações.

- Para as situações simuladas, a rede neural proposta mostrou capacidade de plasticidade ao ser treinada com janelas de diferentes tamanhos (20s, 40s e 60s), sempre com 20 neurônios ocultos.
- O método 4 mostrou a possibilidade de continuamente extrair informações, através da atualização contínua dos pesos sinápticos, sem danificar excessivamente, durante certo intervalo de tempo, o conhecimento acumulado do treinamento anterior. Essa característica confere ao método certa capacidade de treinamento em tempo aproximadamente-real, uma vez que, mesmo considerando que as computações sejam realizadas em tempo real, o sistema deve aguardar duas medições consecutivas do GPS, para que possa ser feita a interpolação linear com as informações da IMU, formando os padrões de treinamento. Além disso, o método 4 processa um número menor de janelas de dados durante o seu treinamento.

A possibilidade de processamento sequencial de informações é uma questão importante, quando se usam sensores inerciais de baixo custo no treinamento de RNA. Como suas características de ruído podem variar muito, entre duas partidas, seus erros de inicialização (*start-up errors*) também podem ser de grande magnitude, inviabilizando o treinamento *off-line*, com base em trajetórias efetuadas previamente.

O estabelecimento de um limite de erro de posicionamento depende da aplicação. Na ausência de uma aplicação, ou critério, pré-estabelecido, foi assumido que um erro de 20m, em 30s de predição, satisfaz aplicações gerais, de um veículo de uso urbano de baixa velocidade, conforme citado em El-Sheimy e Niu (2007). Os resultados mostram que os erros de predição atingem a meta apresentada. Porém, deve ser observado que os erros de predição apresentados pela RNA são relativos ao GPS, que por sua vez possui seus próprios erros nas coordenadas de posição. Portanto, como a RNA aprende com o GPS, os erros de predição podem conter mais uma componente de erro, por exemplo, ± 10 m nas coordenadas Leste e Norte.

É difícil a comparação direta de resultados com outros trabalhos correlatos, principalmente porque um grande número de publicações faz uso de IMUs de outras categorias, mais caras; e o modo de treinamento é baseado em ensaios *off-line* de trajetórias anteriores, como por exemplo, em Chiang e El-Sheimy (2002 e 2005), Chiang et al. (2003, 2004 e 2006). Considerando somente os termos MEMS de baixo custo e treinamento *on-line*, embora a IMU e o receptor GPS não estejam especificados, o trabalho de De Boer et al. (2009) possui alguma semelhança e apresenta resultados que podem ser comparados: 125m de erro médio absoluto (MAE) de posição, em 30s de predição para um veículo estático; 100m de erro de posição, em 30s de predição, para um veículo com velocidade constante e 200m de erro de posição, em 30s de predição, para um veículo numa trajetória real. Os resultados obtidos nesta Tese, no Capítulo 6, mostram, de uma forma geral, erros menores que 1m em 30s de predição, para o veículo estático; e menores que 20m nas outras situações. Deve ser observado que a estratégia proposta por De Boer et al. (2009) prevê a utilização de todos acelerômetros e giroscópios, dentre outras informações, no vetor de entrada da RNA, permitindo que o esquema seja usado em qualquer topografia.

Havendo necessidade, o sistema de navegação proposto pode fornecer a solução de navegação na frequência da IMU, utilizando a RNA no modo predição com o conjunto de pesos sinápticos do treinamento mais recente, ou o mais atualizado, e as leituras do GPS para atualizar a solução:

$$\mathbf{p}_{NAV}^n(t_u + i\delta t) = \mathbf{p}_u^n + \sum_i \Delta \mathbf{p}_{RNA}^n(t_u + i\delta t), \quad i = 1, 2, 3, \dots$$

Aqui, t_u denota o instante da última informação proveniente do GPS, e o índice i é zerado a cada atualização. Em outras palavras, trata-se de uma solução de navegação IMU/GPS desacoplada, com compensação dos erros de posição.

Na forma como foi proposto o sistema de navegação neste trabalho, sua aplicação mais óbvia é aquela voltada para veículos terrestres, incluindo veículos robóticos. Uma extensão das entradas e saídas, contemplando outros ângulos de atitude e a altitude, poderia estender sua utilização para pequenos barcos e aeronaves civis, sempre com

aplicações de baixo custo. A comparação entre o nível de ruído dos sinais dos acelerômetros e giroscópios MEMS, sugere que uma aplicação interessante seria a utilização de IMUs acelerométricas (*gyro-free*) auxiliadas por RNA e GPS.

7.3 Sugestões para extensões e trabalhos futuros

Os resultados obtidos neste trabalho, dentro do escopo de RNA aplicadas à navegação inercial, indicam algumas possibilidades para sua continuidade. São sugeridas as seguintes recomendações para extensões ou trabalhos futuros:

- a) Quanto ao método de treinamento da RNA utilizada neste trabalho, baseado em técnicas estocásticas como o filtro de Kalman, é sugerida a avaliação de outras variações, como por exemplo, a forma fatorizada U-D (BILSKI, 2005 e ZHANG; LI, 1999), e a abordagem *por pontos-sigma* (*Sigma Point KF*) e sua variação chamada *unscented Kalman filter* (WAN; VAN DER MERWE, 2000). O objetivo da sugestão é analisar possíveis melhorias na generalização da rede treinada. Soluções simplificadas do filtro de Kalman com processamento paralelo (RIOS NETO, 1997) ou formas desacopladas (PUSKORIUS; FELDKAMP, 1991), podem ser testadas buscando a redução do tempo de treinamento.
- b) Parece ser coerente a investigação das formas recorrentes de RNA, uma vez que esta estrutura pode ser utilizada na solução de problemas que envolvam processamento temporal, como em atividades de previsão (BRAGA ET AL., 2007). Dentre os algoritmos básicos de aprendizagem para o treinamento de redes recorrentes, o filtro de Kalman estendido, na sua forma desacoplada, pode ser uma opção ao lado da retropropagação através do tempo (BPTT) e da aprendizagem recorrente em tempo real (ARTR), (HAYKIN, 2001).
- c) Diferentes arquiteturas de RNA com diversas configurações de entradas-saídas podem ser investigadas. Pode-se explorar a utilização

de outros sensores como fonte de informação, por exemplo, um magnetômetro com medidas de direção (*heading*) (EL-SHEIMY; CHIANG; NOURELDIN, 2006). Redes com arquiteturas especializadas, com várias entradas e saída única, também podem ser testadas (GOODALL et al., 2006).

- d) Os métodos de treinamento, de 1 até 4, propostos no Capítulo 5 podem ser alterados da seguinte forma: onde houver treinamento baseado numa janela de dados, utiliza-se o método LM, com processamento em lote (*batch*), ao invés do FKE, que seria utilizado apenas para a atualização sequencial do conjunto de pesos sinápticos obtidos no último treinamento. Assim é possível explorar os benefícios de ambos os métodos, ou seja, a boa generalização e capacidade de estabilizar o erro de treinamento em menos iterações do método LM (HAGAN; MENHAJ, 1994), e a possibilidade de processamento sequencial do FKE.
- e) Fica sugerida a investigação da forma híbrida conhecida como filtro Kalman-neural (Stubberud et al., 2007), como uma alternativa de ampliar a utilização de RNA aplicadas à navegação inercial, por exemplo, na identificação on-line de biases de sensores, para sua posterior compensação.

REFERÊNCIAS BIBLIOGRÁFICAS

AGGARWAL, P.; SYED, Z.; NOURELDIN, A.; EL-SHEIMY, N. **MEMS-based integrated navigation**. Norwood, MA, USA: Artech House, 2010. 196 p.

BILSKI, J. The UD RLS algorithm for training feedforward neural networks. **International Journal of Applied Mathematics and Computer Science**. AMCS, n. 1, vol. 15, 2005, p. 115-123.

BINGHAM, N. H.; FRY, J. M. **Regression: linear models in statistics**. London: Springer Verlag, 2010. 284 p.

BRAGA, A. P.; CARVALHO, A. C. P. L.; LUDERMIR, T. B. **Redes neurais artificiais: teoria e aplicações**. Segunda Edição. Rio de Janeiro: LTC Editora, 2007. 226 p.

BRANDT, A.; GARDNER, J. F. Constrained navigation algorithms for strapdown inertial navigation systems with reduced set of sensors. In: AMERICAN CONTROL CONFERENCE, 24-26 june, 1998. Philadelphia, PA, USA. **Proceedings ...** Philadelphia. AACC-IEEE, 1998. p. 1848-1852

CARRARA, V. **Redes neurais aplicadas ao controle de atitude de satélites com geometria variável**. 1997. 202 p. (INPE-6384-TDI/603). Tese (Doutorado em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle) – Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 1997.

CARRARA, V.; RIOS NETO, A. Redes neurais para controle de satélites. CONGRESSO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 4., 08-10 de setembro, 1999, São Paulo, S.P. **Anais ...** São Paulo: [s.n], 1999.

CHIANG, K.-W.; EL-SHEIMY, N. INS/GPS integration using neural networks for land vehicle navigation applications. In: ION GPS 2002 -Institute of Navigation. Portland, OR. **Proceedings ...** Portland: ION, 2002. p. 535-544.

CHIANG, K.-W.; EL-SHEIMY, N. An alternative low cost MEMS IMU/GPS integration scheme. **Coordinates**, Sept., p.10-13, 2005. Disponível em: <www.coordinates.org>. Acesso em: 27 mar 2008.

CHIANG, K.-W.; NOURELDIN, A.; EL-SHEIMY, N. Multisensor integration using neuron computing for land-vehicle navigation. **GPS Solutions**, n. 6, p. 209-218, 2003.

CHIANG, K.-W.; NOURELDIN, A.; EL-SHEIMY, N. A new weight updating method for INS/GPS integration architectures based on neural networks. **Measurement Science and Technology**. Institute of Physics Publishing Ltd, n. 15, p. 2053-2061, 2004.

CHIANG, K.-W.; NASSAR, S.; EL-SHEIMY, N. A constructive autonomous integration scheme of low-cost GPS/MEMS IMU for land vehicular navigation applications. In: POSITION, LOCATION, AND NAVIGATION SYMPOSIUM ,2006. **Proceedings...** [S.l.]: IEEE/ION, 2006, p. 235-243.

CHIANG, K.-W.; HUANG Y.-W. An intelligent navigator for seamless INS/GPS integrated land vehicle navigation applications. **Applied Soft Computing**. Elsevier B.V.: n. 8, p. 722-733, 2008.

DISSANAYAKE et al. The aiding of low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications. **IEEE Transactions on Robotics and Automation**, v. 17, n. 5, 731-747 p., 2001.

DE BOER, J.-R.; CALMETTES, V.; TOURNERET J.-Y; LESOT B. Outage mitigation for GNSS/MEMS navigation using neural networks. In: European Signal Processing Conference, 17., 2009. Glasgow, Scotland, August 24-28. **Proceedings.... EURASP 2009**, p. 2156-2160.

DUMVILLE, M.; TSAKIRI, M. An adaptive filter for land navigation neural computing. In: ION-GPS 1994. **Proceedings...** Institute of Navigation, 1994, p. 1349-1356.

EL-SHEIMY, N.; CHIANG, K.-W.; NOURELDIN, A. The utilization of artificial neural networks for multisensor system integration in navigation and positioning instruments. **IEEE Transactions on Instrumentation and Measurement**, v. 55, n. 5, 1606-1615 p., 2006.

EL-SHEIMY,N.; NIU X. The promise of MEMS to the navigation community. **Inside GNSS**, march/april, 46-56 p., 2007. Disponível em: <<http://www.insidegnss.com>>. Acesso em: 12 ago 2008.

FAHLMAN, S. E.; LEBIERE, C. The cascade learning architecture. In: TOURETZKY D. (Ed.). **Advances in neural information processing system 2**. Denver: Morgan Kaufman, 1990.

FARREL, J. A.; BARTH M. **The global positioning system and inertial navigation**. New York, NY: McGraw-Hill, 1998. 340 p.

FAUSSET, L. **Fundamentals of neural networks**: architectures, algorithms, and applications. Englewood Cliffs, NJ: Prentice-Hall Inc., 1994. 461 p.

FORREST, M.; SPRACKLEN, T.; RYAN, N. An inertial navigation data fusion system employing an artificial neural network as the data integrator. In: ION - NTM 2000. Anaheim, CA. **Proceedings ...** Institute of Navigation, 2000, p. 153-158.

FREITAS PINTO, R. L. U.; RIOS NETO, A. An optimal linear estimation approach to solve systems of linear algebraic equations. **Journal of Computational and Applied Mathematics**, Elsevier Science Publishers, B.V., North-Holland, 1990, n. 33, 261-268

- GELB, E. (Ed.). **Applied optimal estimation**. Massachusetts: MIT Press, 1974. 374 p.
- GOODALL, C.; SYED, Z.; EL-SHEIMY, N. Improving INS/GPS navigation accuracy through compensation of Kalman filter errors. In: VTC Fall -2006. Montreal, Quebec, Canada. **Proceedings ...IEEE**, 2006, p. 1-5.
- GREENSPAN, R. L. GPS and inertial navigation. In: PARKINSON, B. W.; SPILKER J. J. (Ed.). **Global positioning system: theory and applications**. Washington, DC: AIAA, 1996. vol. II, p. 187-220.
- GREJNER-BRZEZINSKA, D. A., et al. Bridging GPS gaps in urban canyons: can ZUPT really help? In: ION- GPS 2001, Salt Lake City, UT. **Proceedings...** The Institute of Navigation, 2001, p. 3139-3148.
- GREWAL, M. S.; WEILL, L. R.; ANDREWS A. P. **Global positioning systems, inertial navigation and integration**. New York, NY: John Wiley and Sons Ltd, 2001. 392 p.
- HAGAN, M. T., MENHAJ, M. B. Training feedforward with the Marquardt algorithm. **IEEE Transactions on Neural Networks**, vol 5, n. 6, 989-993 p., november 1994.
- HAGAN, M. T.; DEMUTH, H. B.; BEALE, M. **Neural networks design**. Boston, MA: PWS Publishing Co. 1995.
- HAYKIN, S. **Redes neurais: princípios e práticas**. Porto Alegre: Bookman, 2001. 900 p.
- HUANG, Y-W.; CHIANG, K-W. An intelligent and autonomous MEMS IMU/GPS integration scheme for low cost land navigation applications. **GPS Solutions**, 2007. Disponível em: <http://www.springerlink.com/content/hm07427_61784757r/fulltext.pdf>. Acesso em: 27 mar 2008.
- IBRAHIM, F.; TASCILLO, A.; AL-HOLOU, N. DGPS/INS integration using neural network methodology. In: INTERNATIONAL CONFERENCE ON TOOLS WITH ARTIFICIAL INTELLIGENCE (ICTA 2000), 12., 2000, Vancouver, BC, Canada. **Proceedings...** Washington, DC, USA: IEEE Computer Society, 2000. p. 114-121.
- IIGUNI, Y.; SAKAI, H. A real time learning algorithm for a multilayered neural network based on the extended Kalman filter. **IEEE Transactions on Signal Processing**, v. 40, n 4. April, p. 959-966, April 1992.
- KAPLAN, E. D.; HEGARTY, C. J. **Understanding GPS: principles and applications**. Second edition. Norwood, MA: Artech House, 2006. 703 p.
- KAYGISIZ, B. H. GPS/INS Enhancement for Land Navigation using Neural Network. **The Journal of Navigation**, n. 57, p. 297-310, 2004.

KAYGIZIZ, B. H.; ERKMEN, A. M., ERKMEN, I. Enhancing positioning accuracy of GPS/INS system during GPS outages utilizing artificial neural network. **Neural Processing Letters**, Springer: n. 25, p. 171-186, 2007.

KUGA, H. K.; LOPES, R. V. F. ; EINWOEGERER, W. Experimental static calibration of an IMU (Inertial Measurement Unit) based on MEMS. In: INTERNATIONAL CONGRESS OF MECHANICAL ENGINEERING, 19., 5-9 november, 2007, Brasília, D.F., Brazil. **Proceedings...** Brasília: ABCM COBEM, 2007.

MARQUES FILHO, E. A., RIOS NETO, A. KUGA, H. K. Artificial neural networks applied to low cost INS/GPS integrated navigation systems. In: WORLD CONGRESS: INTERNATIONAL CONFERENCE ON MATHEMATICAL PROBLEMS IN ENGINEERING, AEROSPACE AND SCIENCE, 8., 30 june-03 july, 2010, São José dos Campos, S.P., Brazil. **Proceedings ...** São José dos Campos: 8th ICNPAA: [s.n] 2010a.

MARQUES FILHO, E. A., RIOS NETO, A. KUGA, H. K. A low cost INS/GPS navigation system integrated with a multilayer feedforward neural network. SIMPÓSIO BRASILEIRO DE NAVEGAÇÃO INERCIAL, 6., 20-22 de outubro, 2010, Rio de Janeiro, R.J. **Anais ...** Rio de Janeiro: VI SBEIN: [s.n], 2010b.

MARQUES FILHO, E. A., RIOS NETO, A. KUGA, H. K. A low cost INS/GPS navigation system integrated by an adaptive neural network training Kalman filter methodology. CONGRESSO BRASILEIRO DE INTELIGÊNCIA COMPUTACIONAL, 10., 8-11 de novembro, 2011, Fortaleza. CE. **Anais ...** Fortaleza: X CBIC: [s.n], 2011.

MONICO, J. F. G. **Posicionamento pelo NAVSTAR-GPS**. São Paulo: Editora UNESP, maio 2000. 287 p.

NORGAARD, M.; RAVN, O.; POULSEN, N. K.; HANSEN L. K. **Neural networks for modeling and control of dynamic systems**. London Berlin Heidelberg: Springer-Verlag, 2000. 246p.

NOURELDIN, A.; SHARAF R.; OSMAN A.; EL-SHEIMY, N. INS/GPS data fusion technique utilizing radial basis functions neural networks. In: IEEE-ION POSITION LOCATION AND NAVIGATION SYMPOSIUM, 2004, Monterey, CA, EUA. **Proceedings...** Monterey: IEEE-ION-PLANS, 2004, p. 280-284.

NOURELDIN, A.; EL-SHAFIE, A.; BAYOUMI, M. GPS/INS integration utilizing dynamic neural networks for vehicular navigation. Article in press. **Information Fusion**, Elsevier B.V., 2010. doi: 10.1016/j.inffus.2010.01.003.

PALMA NETO, L. G.; NICOLETTI, M. C. **Introdução às redes neurais construtivas**. São Carlos: EdUFSCar, 2005. 192 p.

PUSKORIUS, G. V.; FELDKAMP, L. A. Decoupled extended Kalman filter training of feedforward layered networks. In : IEEE International Joint Conference on Neural Networks, 1991, Seattle, EUA. **Proceedings...** Seattle: IEEE,1991. v. I, p. 771-777.

RIOS NETO, A.; KUGA, H. K. Kalman filtering state noise adaptive estimation. In: IASTED INTERNATIONAL CONFERENCE IN TELECOM AND CONTROL – TELECON'85, 2., 1985, Rio de Janeiro. **Proceedings...** Rio de Janeiro, RJ, Brazil, 1985, p. 210-213.

RIOS NETO, A. Kalman filtering stochastic optimization algorithm and usual backpropagation in neural nets training. In: BRAZILIAN CONGRESS IN NEURAL NETWORKS, 2., 1995, Curitiba PR. **Proceedings ...**Curitiba, 1995, p. 139-144.

RIOS NETO, A. Stochastic optimal linear parameter estimation and neural nets training is system modeling. **RBCM – Journal of the Braz. Soc. Mechanical Sciences**, v. XIX, n. 2, p. 138-146, 1997.

RIOS NETO, A.; HEMERLY, E. M. Observadores de estado e filtro de Kalman. In: AGUIRRE, L. A., et al. (Ed.). **Enciclopédia de automática: controle e automação**. São Paulo, S.P.: Editora Blucher, 2007. vol. III3, p. 158-181.

RIOS NETO, W. **Estimação dos coeficientes aerodinâmicos utilizando Adaptive Functional Link Network**. 2008. 154 p. Tese (Doutorado em Mecânica e Controle de Vôo) – Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, 2008.

RUMELHART, D. E.; HINTON G. E.; WILLIAMS R. J. Learning internal representations by error propagation. In: RUMELHART, D. E.; MCCLELLAND, J. I. **Parallel distributed processing: exploration and microstructure of cognition**. Cambridge, MA: MIT Press, p. 318-362, 1986.

SHAH, S.; PALMIERI, F. MEKA – A fast local algorithm for training feedforward neural networks. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, 1990, San Diego, CA. **Proceedings ...** San Diego, 1990, v. 3, p. 41-46.

SHI, H.; ZHU, J.; SUN, Z. INS/GPS integrated system state estimation based on hopfield neural network. In: INTERNATIONAL CONFERENCE ON NEURAL NETWORKS AND BRAIN, 2005, Beijing. **Proceedings...** Beijing: IEEE, 2005 v. 2, p. 975-979.

SILVA, J. A.; RIOS NETO, A. Preliminary testing and analysis of an adaptive neural network training Kalman filtering algorithm. In: BRAZILIAN CONFERENCE ON NEURAL NETWORKS, 4., July 1999, São José dos Campos, S.P., Brazil. **Proceedings ...** São José dos Campos: ITA,1999, p. 247-251.

SINGHAL, S.; WU, L. Training multilayer perceptrons with the extended Kalman algorithm. In: TSOURTZKY D. S. (ed.). **Advances in neural information 1**. San Mateo, CA. Morgan Kaufman: p. 133-140, 1989.

SOUZA FILHO, D. F.; RIOS NETO, A. Test of a local processing neural network training Kalman filtering algorithm on a breast cancer diagnosis benchmark problem. BRAZILIAN AUTOMATIC CONTROL CONFERENCE, 12., Sept.1998, Uberlândia, M.G., Brazil. **Proceedings ...** Uberlândia: DEENE - CETEC - UFU, 1998, v. 2, p. 417-422.

STUBBERUD, S. C.; KRAMER K. A.; GEREMIA, J. A. Online sensor modeling using a neural Kalman filter. **IEEE Transaction on Instrumentation and Measurements**, v. 56, n. 4, august 2007, p. 1451-1458.

TITTERTON, D. H.; WESTON, J. L. **Strapdown inertial navigation technology**. Second edition. Reston, VA, USA: AIAA, 2004. 558 p.

WAN, E. A., VAN DER MERWE, R. The unscented Kalman filter for nonlinear estimation. In: ADAPTIVE SYSTEMS FOR SIGNAL PROCESSING, COMMUNICATIONS, AND CONTROL. SYMPOSIUM,(AS-SPCC,2000). oct. 2000, Lake Louise, Alta., Canada. **Proceedings ...** Lake Louise: IEEE 2000. p. 153-158.

WANG, J.-H.; GAO, Y. Performance improvement of a low-cost gyro-free INS for land vehicle navigation by using constrained navigation algorithm and neural network. In: ION- GNSS 2003, 2003, Portland, OR. **Proceedings...** Portland:Institute of Navigation, 2003, p. 762-768.

YAZDI, N; AYAZI, F.; NAJAFI, K. Micromachined inertial sensors. **Proceedings of the IEEE**, v. 86, n. 8, p. 1640-1659, ago. 1998.

ZHANG, Y., LI X. R. A fast U-D factorization-based learning algorithm with applications to nonlinear system modeling and identification. **IEEE Transactions on Neural Networks**. Vol. 10, n. 4. July, 1999, p. 930-938.

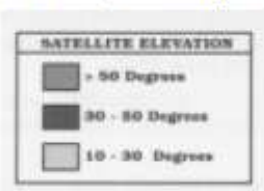
APÊNDICE A – CARACTERÍSTICAS DA IMU e RECEPTOR GPS

Z-12 TECHNICAL SPECIFICATIONS

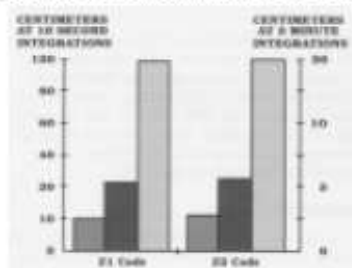
Measurement Precision

C/A (>25°)		
Carrier phase (1 sec)	(25 Hz)	0.15 cm 0.02 cm
Pseudo-range (5 sec)	(10 sec)	20.00 cm 3.60 cm
P-Code A/S Off (>25°)		
L1 Carrier phase (5 min)	(10 sec)	0.10 cm 0.02 cm
L1 Pseudo-range (5 min)	(10 sec)	5.00 cm 0.90 cm
L2 Carrier phase (5 min)	(10 sec)	0.10 cm 0.02 cm
L2 Pseudo-range (5 min)	(10 sec)	7.00 cm 1.30 cm
Real-Time Differential Position (PDOP<4)		<1 m
Static, Rapid Static or Pseudo-Kinematic Survey		5 mm + 1 ppm

P-Code A/S On (Z-Tracking)



OBSERVABLE RMS IN CENTIMETERS FOR THE Z-12



Systematic Errors (Between Satellites)
 Pseudo-Range (all bands) <1.00 cm
 Carrier Phase (all bands) <0.01 cm

Ashtech P-Code GPS receivers have been FGCC tested and are capable of performing first order survey (report available upon request).

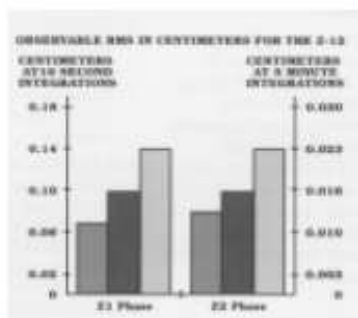
Z- 12, Z-Tracking, PNAV and PRISM II are trademarks of Ashtech Inc.

Specifications are subject to change without notice.

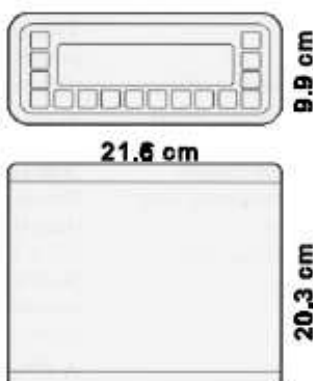
Environmental

Waterproof to	5 psi
Temperature Ranges	
Receiver/Data Logger	
Operating	-20° to +55°C
Storage	-30° to +75°C
Antenna	
Operating	-40° to +65°C
Storage	-55° to +75°C
Humidity	100%
Weight	
Receiver	4.0 kg
Antenna	1.7 kg
Speed (Max)	Does not exceed 1,000 nautical miles-per-hour
Altitude (Max)	Does not exceed 60,000 Ft.

OBSERVABLE RMS IN CENTIMETERS FOR THE Z-12



Dimensions



Standard Features

- 12 Channel "All-In-View" operation
- Automatic Switching to Z-Tracking when A/S is activated.
- Full wavelength carrier on L 1 and L2.
- 21 Watt power consumption (typical)
- 10 - 32 VDC input
- 2 Power inputs
- Audible alarm for low power
- Internal RAM data recorder
- 8-Line by 40-character display
- 4 RS-232 ports (115,200 baud max)
- Static, rapid static, kinematic, pseudo-kinematic surveys
- Waypoint navigation
- Real-time data outputs
- 1 PPS timing signal
- Cold start - 2 Minutes to first data
- Warm start - <30 Seconds to first data

Standard Accessories

- Precision geodetic antenna
- 10-metre antenna cable
- External power cable
- RS-232 data cable (Z-format)
- Battery and charger
- Rotatable Tribrach adapter
- High-impact shipping case
- Receiver operating manual (Shipping weight of standard Z-12 package is 48 pounds)

Optional Features

- External frequency standard input 1 to 20 MHz in 10 KHz steps
- Real-time differential GPS RTCM format
- Expanded internal memory

Optional Accessories

- Survey Tribrach
- Kinematic bipod and pole
- 10, 30 and 60-meter antenna cable, expandable to 150 meters w/line amps
- External battery
- Battery charger 110/120 VAC
- PRISM II™ Software Package
- PNAV Software Package

*Display off/with LNA.

Figura A.1 – Receptor GPS Ashtec Z12 – características técnicas

Tabela A.1 - Crossbow IMU400CD – especificações

Giroscópios	Valor	Acelerômetros	Valor
Range Roll, Pitch, Yaw (°/sec)	+/- 200	Range X,Y,Z (g)	+/- 4
Bias Roll, Pitch, Yaw (°/sec)	<+/- 1.0	Bias X,Y,Z (mg)	< +/- 12
Scale Factor Accuracy (%)	< 1	Factor Accuracy (%)	< 1
Non-Linearity (% FS)	< 0.3	Non-Linearity (% FS)	< 1
Resolution (°/sec)	< 0.05	Resolution (mg)	< 0.6
Bandwidth (Hz)	> 25	Bandwidth (Hz)	> 75
Randon Walk (°/hr ^{1/2})	< 4.5	Random Walk (m/s/hr ^{1/2})	< 1

APÊNDICE B – Matriz de derivadas parciais \mathbf{H}

A avaliação da matriz de derivadas parciais, \mathbf{H} , do vetor de saída da RNA em relação aos seus pesos, é necessária para os algoritmos propostos anteriormente. Será apresentada uma forma recursiva para sua obtenção, com a mesma notação usada no Capítulo 3 (CARRARA, 1997; HAYKIN, 2001).

A matriz \mathbf{H} é definida por:

$$\hat{f}_{\mathbf{w}}(\mathbf{x}(t), \bar{\mathbf{w}}) = \left. \frac{\partial \hat{f}(\mathbf{x}(t), \mathbf{w})}{\partial \mathbf{w}} \right|_{\mathbf{w}=\bar{\mathbf{w}}} = \mathbf{H}(\mathbf{x}(t), \bar{\mathbf{w}}) \quad (\text{B.1})$$

Numa rede neural constituída por ℓ camadas, a saída da camada k , com n_k neurônios é dada por:

$$\mathbf{x}^k = \left[f^k(\bar{x}_1^k) \quad \dots \quad f^k(\bar{x}_{n_k}^k) \quad 1 \right]^T \quad (\text{B.2})$$

Onde $f^k(\cdot)$ denota a função de ativação do campo local induzido, ou potencial de ativação \bar{x}^k :

$$\mathbf{x}^k = f^k(\bar{\mathbf{x}}^k) = f^k(\mathbf{W}^k \mathbf{x}^{k-1}) \quad (\text{B.3})$$

$$\mathbf{W}^k = \begin{bmatrix} w_{11}^k & \dots & w_{1n_{k-1}}^k & b_1^k \\ w_{21}^k & \dots & w_{2n_{k-1}}^k & b_2^k \\ \vdots & & \vdots & \vdots \\ w_{n_k 1}^k & \dots & w_{n_k n_{k-1}}^k & b_{n_k}^k \end{bmatrix} \quad (\text{B.4})$$

Na ℓ -ésima camada, o vetor de saída da rede neural é dado por:

$$\mathbf{y} = \mathbf{x}^\ell \quad (\text{B.5})$$

De forma análoga ao algoritmo de retropropagação, descrito no Capítulo 3, a derivada parcial do vetor saída, em relação às diversas camadas da rede neural, é dada por:

$$\mathbf{H}_i^k = \frac{\partial \mathbf{x}^\ell}{\partial w_i^k} = \mathbf{F}^\ell \mathbf{W}^\ell \frac{\partial \mathbf{x}^{\ell-1}}{\partial w_i^k} \quad (\text{B.6})$$

Onde \mathbf{F}^k denota uma matriz diagonal de derivadas, $f^{k'}(\bar{\mathbf{x}}^k)$.

$$\mathbf{F}^k = \frac{d\mathbf{f}^k(\bar{\mathbf{x}}^k)}{d\bar{\mathbf{x}}^k} = \begin{bmatrix} f^{k'}(\bar{x}_1^k) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & f^{k'}(\bar{x}_{n_k}^k) \end{bmatrix} \quad (\text{B.7})$$

Tomando novamente a derivada parcial para as camadas subseqüentes, em direção à entrada da rede, a matriz \mathbf{H} toma a seguinte forma:

$$\mathbf{H}_i^k = \mathbf{F}^\ell \mathbf{W}^\ell \mathbf{F}^{\ell-1} \mathbf{W}^{\ell-1} \dots \mathbf{F}^{k+1} \mathbf{W}^{k+1} \mathbf{F}^k \frac{\partial \bar{\mathbf{x}}}{\partial w_i^k} \quad (\text{B.8})$$

Pode-se, então, estabelecer uma relação de recorrência na seguinte forma:

$$\Delta^k = \mathbf{F}^\ell \mathbf{W}^\ell \mathbf{F}^{\ell-1} \mathbf{W}^{\ell-1} \dots \mathbf{F}^{k+1} \mathbf{W}^{k+1} \mathbf{F}^k = \Delta^{k+1} \mathbf{W}^{k+1} \mathbf{F}^k \quad (\text{B.9})$$

Onde o valor inicial de Δ^k , para a camada de saída ℓ , é dado por:

$$\Delta^\ell = \mathbf{F}^\ell \quad (\text{B.10})$$

$$\mathbf{H}_i^k = \Delta^k \frac{\partial \bar{\mathbf{x}}}{\partial w_i^k} \quad (\text{B.11})$$

A derivada do campo local induzido, ou potencial de ativação, da camada k em relação aos pesos w_i é dada por:

$$\frac{\partial \bar{\mathbf{x}}}{\partial w_i^k} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{k-1} & x_2^{k-1} & \cdots & x_{n_{k-1}}^{k-1} \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad (\text{B.12})$$

Observa-se na matriz, de n_k linhas, descrita na Equação (B.12), que somente a i -ésima linha possui elementos não nulos, uma vez que não existe conexão entre os demais neurônios e os pesos w_i , resultando em derivadas nulas.

Considerando todas as ℓ camadas da rede, a matriz \mathbf{H} é descrita na seguinte forma:

$$\mathbf{H} = [\mathbf{H}^1 \quad \mathbf{H}^2 \quad \dots \quad \mathbf{H}^\ell] = [\mathbf{H}_1^1 \dots \mathbf{H}_{n_1}^1 \quad \mathbf{H}_1^2 \dots \mathbf{H}_{n_2}^2 \quad \dots \quad \mathbf{H}_1^\ell \dots \mathbf{H}_{n_\ell}^\ell] \quad (\text{B.13})$$

A matriz \mathbf{H} possui dimensão $(n_\ell \times n_w)$, onde n_ℓ é a dimensão da camada de saída da rede, e n_w corresponde ao número de parâmetros da rede neural, os pesos sinápticos e biases:

$$n_w = \sum_{m=1}^{\ell} n_m (n_{m-1} + 1) \quad (\text{B.14})$$

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Manuais Técnicos (MAN)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Notas Técnico-Científicas (NTC)

Relatórios de Pesquisa (RPQ)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programa de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Propostas e Relatórios de Projetos (PRP)

Publicações Didáticas (PUD)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Incluem apostilas, notas de aula e manuais didáticos.

Publicações Seriadas

Programas de Computador (PDC)

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Consta destas publicações o International Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. São aceitos tanto programas fonte quanto executáveis.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.