



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-8481-TDI/779

**NOVAS ABORDAGENS EM ASSIMILAÇÃO DE DADOS
METEOROLÓGICOS**

Alexandre Guirland Nowosad

Tese de Doutorado em Computação Aplicada, orientada pelos Drs. Haroldo de Fraga Campos Velho e Atair Rios Neto, aprovada em 14 de março de 2001.

519.6 : 551.5

NOWOSAD, A. G.

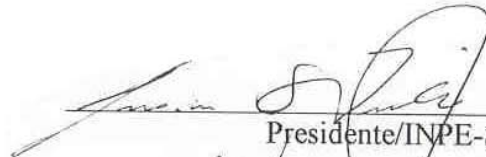
Novas abordagens em assimilação de dados meteorológicos / A.G. Nowosad. – São José dos Campos: INPE, 2001.

120p. – (INPE-8481-TDI/779).

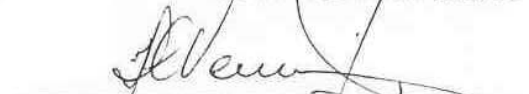
1.Processamento de dados. 2.Meteorologia. 3.Redes neurais. 4.Filtros de Kalman. 5.Sistemas não lineares. I.Título.

Aprovado pela Banca Examinadora em
cumprimento a requisito exigido para a
obtenção do Título de **Doutor** em
Computação Aplicada.


Dr. Jerônimo dos Santos Travelho


Presidente/INPE-SJC/SP

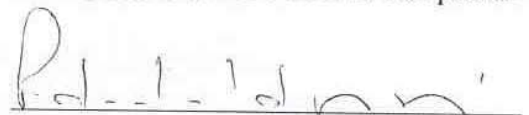
Dr. Haroldo Fraga de Campos Velho


Orientador/INPE-CPaulista/SP


Dr. Atair Rios Neto


Orientador/CDT-EEI-SJCampos/SP


Dr. Pedro Leite da Silva Dias


Membro da Banca
Convidado IAG/USP-São Paulo/SP

Dr. André Ponce de Leon F. de Carvalho


Membro da Banca
Convidado ICMSC/USP-São Paulo/SP

Dr. Ricardo Todling


Membro da Banca
Convidado NASA - EUA

Candidato (a): Alexandre Guirland Nowosad

São José dos Campos, 14 de março de 2001.

RESUMO

Esta tese examina duas novas abordagens para o problema de assimilação de dados atmosféricos, o uso de um Filtro de Kalman Estendido Adaptativo e Redes Neurais de Perceptrons Multicamadas. Primeiro, o Filtro de Kalman Estendido Adaptativo é usado para assimilação de dados em três sistemas dinâmicos não-lineares: sistema de tempo discreto caótico de Hénon, equação diferencial caótica de Lorenz e modelo computacional DYNAMO para a atmosfera. Esta abordagem não requereu que o erro de modelagem fosse estacionário e usa um filtro de Kalman linear para estimar este erro. Este método foi comparado aos métodos usando transformada de Laplace, Filtro de Kalman Estendido e Linear. A conclusão foi que somente o Filtro de Kalman Estendido Adaptativo pôde ser usado com confiabilidade em todos os sistemas. Segundo, Redes Neurais de Perceptrons Multicamadas foram usadas para assimilação de dados nos mesmos sistemas dinâmicos. Nesta abordagem a rede é treinada para "emular" assimilação de dados Filtro de Kalman, com o objetivo de evitar recálculo da matriz de ganho a cada instante de assimilação. Um novo procedimento para treinar as redes também foi proposto, baseado numa modificação do algoritmo de retropropagação. Resultados de testes encorajadores são mostrados. Mostrou-se também que para sistemas de alta dimensão as redes treinadas podem ser computacionalmente mais rápidas que os filtros de Kalman.

NEW APPROACHES TO METEOROLOGIC DATA ASSIMILATION

ABSTRACT

This thesis examines two new approaches to the problem of atmospheric data assimilation, the use of an Adaptive Extended Kalman Filter and Multilayer Perceptron Neural Networks. Firstly, the Adaptive Extended Kalman Filter is used for data assimilation in three nonlinear dynamical systems: Hénon discrete-time chaotic dynamic system, Lorenz chaotic differential equation and computational model DYNAMO for the atmosphere. This approach does not require the modelling error to be stationary and uses a linear Kalman filter to estimate this error. For the last two systems this method has been compared to the methods using Laplace Transform, Linear and Extended Kalman Filter. The conclusion has been that only the Adaptive Extended Kalman Filter could be used reliably in all systems. Secondly, Multilayer Perceptron Neural Networks have been used for data assimilation in the same dynamic systems. In this approach the network is trained to "emulate" Kalman Filter data assimilation method with the objective of avoiding recalculation of the gain matrix at each instant of assimilation. A new procedure for training the networks has also been proposed, based on a modification in the backpropagation algorithm. Encouraging test results are shown. It has also been shown that for high-dimensional systems the trained networks can be computationally faster than the Kalman filters.

SUMÁRIO

Pág.

LISTA DE FIGURAS

CAPÍTULO 1	INTRODUÇÃO	15
CAPÍTULO 2	ALGUNS MÉTODOS PARA ASSIMILAÇÃO DE DADOS METEOROLÓGICOS	23
2.1	Assimilação de Dados	23
2.1.1	Interpolação Ótima	26
2.1.2	Relaxação Dinâmica	27
2.1.3	Análise Variacional em 4 Dimensões	27
2.2	Transformada de Laplace	28
2.3	Filtro de Kalman	29
2.3.1	Um Filtro de Kalman Adaptativo: Algoritmo de Kuga e Rios Neto	33
CAPÍTULO 3	REDES NEURAIIS MULTICAMADAS DE PERCEPTRONS	37
3.1	Introdução: Que é uma Rede Neural	37
3.2	Modelo de um Neurônio	38
3.3	Redes de Perceptrons com Múltiplas Camadas (ou Multicamadas)	39

3.4	Processo de Aprendizado	42
3.5	Algoritmos de Retropropagação	46
3.5.1	Uma Versão Adaptativa	46
3.5.1.1	Modificação no Algoritmo	47
3.6	Approximação de Funções	48

CAPÍTULO 4 SISTEMAS DE EVOLUÇÃO NÃO-LINEARES .. 51

4.1	Sistema de Hénon Caótico	51
4.2	Sistema de Lorenz Caótico	52
4.3	O Modelo Meteorológico DYNAMO	54
4.3.1	Equações do Modelo	55
4.3.2	Adimensionalização das Equações	57
4.3.3	Discretização Espacial	59

CAPÍTULO 5 RESULTADOS NUMÉRICOS DO FILTRO DE KALMAN ESTENDIDO ADAPTATIVO 63

5.1	Resultados Obtidos com o Sistema de Hénon Caótico	63
5.2	Resultados Obtidos com o Sistema de Lorenz	65
5.3	Resultados Obtidos com o Modelo DYNAMO	73

CAPÍTULO 6 RESULTADOS NUMÉRICOS DAS REDES NEURAIS 89

6.1	Resultados com o Sistema de Hénon Caótico	89
-----	---	----

6.2	Resultados com o Sistema de Lorenz Caótico	91
6.3	Resultados com o modelo DYNAMO	94
6.4	Custo Computacional do Novo Método	104
CAPÍTULO 7 CONCLUSÕES GERAIS		107
7.1	Contribuições	107
7.2	Conclusões	107
7.3	Trabalhos futuros	109
REFERÊNCIAS BIBLIOGRÁFICAS		111
APÊNDICE A TEORIA DA ESTIMAÇÃO		119

LISTA DE FIGURAS

	Pág.
2.1	Diagrama esquemático do ciclo de assimilação de dados intermitente. 25
2.2	Diagrama esquemático do ciclo de assimilação de dados contínua. 26
2.3	Aproximação de integral de contorno para transformada de Laplace inversa. 28
2.4	Diagrama esquemático do Filtro de Kalman Linear. 31
2.5	Diagrama esquemático do filtro de Kalman que estima q_{n+1} 33
3.1	Modelo não-linear de um neurônio 38
3.2	Rede de Perceptrons com uma camada escondida e camada de saída 41
3.3	Diagrama de blocos de aprendizado supervisionado 45
4.1	Corrupção temporal da variável perturbada X no sistema de Hénon. 52
4.2	Corrupção temporal da variável perturbada Y no sistema de Lorenz. 53
5.1	Assimilação de dados no sistema de Hénon usando Filtro de Kalman Adaptativo. 64
5.2	Variância total de erro na assimilação usando Filtro de Kalman Adaptativo. 65
5.3	Comparação entre assimilações no sistema de Lorenz usando Transformada de Laplace (recursão de 1ª ordem), Filtro de Kalman Linear, Filtro de Kalman Estendido e Filtro de Kalman Adaptativo. 66

5.4	Variâncias totais de erro na assimilação usando Filtros de Kalman Linear, Estendido e Adaptativo.	67
5.5	Comparação entre assimilações usando Filtros de Kalman Estendido e Adaptativo com modelo <i>incorreto</i>	70
5.6	Variâncias totais de erro na assimilação usando Filtros de Kalman Estendido e Adaptativo com modelo <i>incorreto</i>	71
5.7	Perfis das perturbações ΔU , ΔV e $\Delta\phi$	74
5.8	Corrupção Temporal de U na simulação perturbada não-inicializada.	74
5.9	Evolução Temporal de U na assimilação usando Transformada de Laplace.	75
5.10	Evolução Temporal da diferença entre U obtida através da T. Laplace usando recursão de 2 ^a ordem e recursão de 1 ^a ordem.	76
5.11	Implementação do Filtro de Kalman	77
5.12	Evolução Temporal de U na assimilação usando Filtro de Kalman Linear.	78
5.13	Variâncias Totais de erro na assimilação usando Filtro de Kalman Linear.	79
5.14	Evolução temporal de U na assimilação usando Filtro de Kalman Estendido.	79
5.15	Variâncias totais de erro na assimilação usando Filtro de Kalman Estendido.	80
5.16	Evolução Temporal de U na assimilação usando Filtro de Kalman Adaptativo.	81
5.17	Variâncias totais de erro na assimilação usando Filtro de Kalman Adaptativo.	81

5.18	Evolução temporal de U na assimilação usando Filtro de Kalman Adaptativo com erro nos parâmetros R_O , R_F and R_β	83
5.19	Variâncias totais de erro na assimilação usando Filtro de Kalman Adaptativo com erro nos parâmetros R_O , R_F e R_β	83
5.20	Evolução temporal de U na assimilação usando Filtro de Kalman Adaptativo com adição de erro de modelagem μ'	84
5.21	Variâncias totais de erro na assimilação usando Filtro de Kalman Adaptativo com adição de erro de modelagem μ'	85
5.22	Evolução temporal de U na assimilação usando Filtro de Kalman Adaptativo com mudanças temporais nos parâmetros R_O , R_F e R_β do <i>signal verdadeiro</i>	86
5.23	Variâncias totais de erro na assimilação usando Filtro de Kalman Adaptativo com mudancas temporais nos parametros R_O , R_F e R_β do <i>signal verdadeiro</i>	87
5.24	Evolução temporal de U na assimilação usando Transformada de Laplace com mudanças temporais nos parâmetros R_O , R_F e R_β do <i>signal verdadeiro</i>	87
6.1	Assimilação de dados no sistema de Hénon usando Rede Neural.	90
6.2	Assimilação de dados no sistema de Lorenz usando Rede Neural.	92
6.3	Evolução temporal de U em assimilação usando Rede Neural com uma camada escondida e algoritmo de retropropagação adaptativo.	95
6.4	Variâncias totais de erro na assimilação usando Rede Neural com uma camada escondida e algoritmo de retropropagação adaptativo.	97

6.5	Evolução temporal de U em assimilação usando Rede Neural com duas camadas escondidas e algoritmo de retropropagação adaptativo modificado.	99
6.6	Variâncias totais de erro na assimilação usando Rede Neural com duas camadas escondidas e algoritmo de retropropagação adaptativo modificado.	99
6.7	Evolução temporal de U na assimilação usando Rede Neural com uma camada escondida, algoritmo de retropropagação adaptativo e mudanças temporais nos parâmetros R_O , R_F and R_β do <i>signal verdadeiro</i>	101
6.8	Variâncias totais de erro na assimilação usando Rede Neural com uma camada escondida, algoritmo de retropropagacao adaptativo modificado e mudanças temporais nos paraâmetros R_O , R_F e R_β do <i>signal verdadeiro</i>	101
6.9	Evolução temporal de U na assimilação usando Rede Neural com duas camadas escondidas, algoritmo de retropropagação adaptativo e mudanças temporais nos parâmetros R_O , R_F and R_β do <i>signal verdadeiro</i>	102
6.10	Variâncias totais de erro na assimilação usando Rede Neural com duas camadas escondidas, algoritmo de retropropagação adaptativo modificado e mudanças temporais nos parâmetros R_O , R_F , R_β do <i>signal verdadeiro</i>	103
6.11	Evolução da soma de erros quadráticos (e_m) no treinamento das redes.	103
6.12	Evolução da taxa de aprendizado (η_m) no treinamento das redes.	105

CAPÍTULO 1

INTRODUÇÃO

Sem uma maneira eficiente e precisa para processar as numerosas observações da natureza, os métodos da previsão numérica do tempo seriam imprecisos, pois a atmosfera é um sistema caótico em que erros introduzidos no sistema podem crescer ao longo do tempo. Como conseqüência, assimilação de dados é uma luta entre destruição caótica de conhecimento e sua restauração por novas observações [28].

Previsão numérica do tempo depende de uma representação precisa do estado inicial da atmosfera ([10], [28]). Um erro na análise do estado inicial pode ser amplificado e causar um erro drástico no estado previsto. Hoje há muitos dispositivos para medir as variáveis atmosféricas, como balões, barômetros, higrômetros, radar, e satélites. Cada forma de medida tem erros e vieses, e há áreas no globo onde é difícil coletar dados. Assimilação de dados é uma maneira de usar o máximo possível de observações, tal que, mesmo havendo erros observacionais, seja possível descobrir o estado da atmosfera que não apenas seja coerente com as observações, mas também a dinâmica da atmosfera em tempo anterior dado por um modelo prognóstico. Assim, assimilação de dados será definida como o processo de produzir conjuntos de dados, ou análises, a partir de observações cruas do estado da atmosfera em evolução e pelo estado da atmosfera dado por um modelo matemático. As análises serão o melhor ajuste tanto às observações quanto ao conhecimento prévio do estado da atmosfera, mantidos os vínculos físicos e dinâmicos do sistema.

Este trabalho teve dois objetivos. O primeiro e principal foi aplicar Redes Neurais à Assimilação de Dados. O segundo objetivo foi aplicar um novo tipo de filtro adaptativo na metodologia para Assimilação chamada Filtro de Kalman, que já inclui outros tipos de filtragem adaptativa.

Pode-se apresentar a evolução da assimilação de dados como uma cadeia de conquistas científicas iniciadas por Richardson em 1922 ([10], [28]). Richardson foi o

primeiro a tentar previsão numérica do tempo na Europa em 1922. Usou procedimentos numéricos para integrar as equações governantes da atmosfera para frente no tempo a partir do estado inicial. Seus valores iniciais de pressão e velocidade de vento foram obtidos de estações de observação irregularmente espaçadas. No entanto ele precisava de valores iniciais em uma grade regularmente espaçada. Ele calculou estas estimativas manualmente em um processo demorado chamado *análise subjetiva*. O método falhou, pois faltava ao método de Richardson inicialização, e o passo de tempo de 12h usado por Richardson era grande demais. Inicialização inclui produzir um estado analisado no qual os campos de massa e de vento estão aproximadamente equilibrados pelo balanço geostrófico. Sem este equilíbrio movimentos denominados *ondas de gravidade inerciais* podem crescer e destruir previsões.

Ondas de gravidade inerciais são ondas em que a aceleração da gravidade atua sobre a atmosfera fazendo parcelas da atmosfera oscilarem em torno de uma posição de equilíbrio [33]. As parcelas oscilam em um plano vertical em trajetórias paralelas à frente de onda. Seu comprimento de onda é da ordem de 1000 km e o período é da ordem de algumas horas. Podem ser modificadas pela rotação da Terra. Tais oscilações surgem devido ao desequilíbrio existente entre a distribuição de massa e velocidade nos dados iniciais [21]. Durante a integração do modelo atmosférico as ondas de gravidade inerciais podem amplificar-se mascarando os movimentos meteorológicos importantes como ondas de Rossby.

Com o desenvolvimento do primeiro computador, ENIAC, em 1946, o Instituto de Estudos Avançados de Princeton (IEAP) fez uma segunda tentativa de previsão numérica do tempo ([10], [28]). O grupo de Princeton usou as equações usadas por Richardson para produzir equações balanceadas que não mantivessem ondas de gravidade inerciais.

O IEAP desenvolveu também um processo automático chamado *análise objetiva* para determinar precisamente os valores na grade das variáveis de estado da atmosfera e encontrar dados para áreas esparsas interpolando observações das áreas adjacentes. Panofsky modificou a análise objetiva em 1949. Um domínio incluindo

vários pontos de grade foi estabelecido e uma expansão polinomial foi usada para ajustar todas as observações a pontos de grade dentro do domínio, usando um algoritmo de mínimos quadrados ponderados conforme a incerteza de cada observação. Um vínculo dinâmico foi usado para acoplar os campos de massa e vento. Descontinuidades entre domínios foram produzidas quando polinômios foram ajustados a cada domínio. Em 1954, Gilchrist e Cressman sugeriram usar a previsão anterior como a tentativa inicial para o estado inicial atual. A previsão inicial ficou conhecida como *estimativa a priori*. Bengthorsson e Doos em 1955 subtraíram a estimativa a priori (obtida das equações ou de um estudo climático) de cada campo de observações, produzindo uma diferença (*inovação*) observacional para produzir uma nova informação (*inovação*) de análise. O campo de inovação de análise, composto de combinações lineares de inovações observacionais da área adjacente, era adicionado à estimativa a priori para produzir a análise. Em 1960, o progresso em análise objetiva continuava e previsões numéricas se tornavam universais. Centros de meteorologia nacionais usavam procedimentos de previsão numérica operacionais. O sistema dos EUA usavam o sistema de Bergthorsson e Doos, um campo de estimativa a priori e inovações observacionais analisadas. Na década de 70, o procedimento estatístico conhecido como *interpolação ótima*, primeiramente apresentado por Eliassen em 1954 e depois novamente por Gandin em 1963, ganhou aceitação.

Interpolação Ótima (IO) faz várias suposições para torná-la computacionalmente realizável ([28]; [10]; [56]). Uma é isotropia, a suposição de que, selecionado um ponto no espaço, as funções de covariância de erro horizontais são do mesmo valor em todos os lugares equidistantes deste ponto. Outra suposição é separabilidade das funções de erro tridimensionais para que funções de correção possam ser tratadas separadamente. Supõe-se também um vínculo dinâmico entre covariâncias de erro de vento e altura. Além disso uma aproximação local é feita, significando que as análises em cada ponto de grade incorporam apenas dados observacionais em alguma vizinhança daquele ponto (se a variabilidade local da atmosfera for pequena isto pode ser um problema).

Atualmente, uma versão da IO, chamado Sistema de Análise Estatística em Espaço Físico ('Physical-space Statistical Analysis System'), é usada operacionalmente no Sistema de Assimilação de Dados do Sistema Observador da Terra de Goddard ('Goddard Earth Observing System Data Assimilation System - GEOS DAS') [28]. Este método foi projetado especificamente para o "Escritório de Assimilação de Dados" ('Data Assimilation Office - DAO') da NASA com alguns requerimentos. Entre eles, realizar a interpolação global ao invés de localmente e remover a suposição de isotropia.

Outra versão da IO é o Sistema de Análise Interpolação-Estatística Espectral ('Spectral Statistical-Interpolation Analysis System') [50], usado no Centro Meteorológico Nacional ('National Meteorological Center - NMC') dos EUA e no Centro de Previsão do Tempo e Estudos Climáticos - CPTEC do INPE. Neste método as variáveis de análise são fortemente relacionadas aos coeficientes das das expansões harmônicas da vorticidade, divergência, temperatura, razão de mistura, logaritmo da pressão superficial, todos expressos em coordenadas σ , usados pelo modelo espectral do NMC. Além disso a análise é global.

Outro método estatístico é o *Filtro de Kalman*, no qual a inovação de análise é calculada minimizando-se o erro quadrático médio estatístico ao longo do tempo ([10]; [56]). Este método ainda apresenta a vantagem de propagar a estimativa de covariância de erro de uma inserção de dados para a outra, de maneira que a confiabilidade da previsão pode ser avaliada a qualquer instante da previsão. A sua desvantagem é que devido aos produtos matriciais envolvidos o método pode se tornar lento para modelos com muitos pontos de grade. Este método será explicado mais detalhadamente no Capítulo 2. Usa-se uma formulação de estimação de processos estocásticos gerados por equações de diferença aleatoriamente perturbadas [24]. As primeiras tentativas de aplicar esta idéia a problemas meteorológicos foram feitas por Jones ([26]) e Petersen ([51]), mas elas tiveram pequeno impacto nas práticas operacionais da época. Mais recentemente, um grupo no Instituto Courant de Nova York fez uma tentativa coordenada de aplicar a metodologia de Kalman ao

problema da assimilação de dados meteorológicos (Ghil et al.[20]). Outros trabalhos importantes foram o de Cohn e Parrish ([9]), aplicando o filtro de Kalman a um modelo simples para a atmosfera sobre os E.U.A., e o de Evensen ([15]), usando abordagem de Monte Carlo para simplificar o cálculo da covariância de erro.

Um método determinístico em uso operacional é o *Método Variacional* ou *Método 3D-Var*, que pode ser considerado estimação da atmosfera por minimização do erro quadrático médio ([28]; [10]). Este método é usado pelos Centros Nacionais de Previsão Ambiental dos EUA ('National Centers for Environmental Prediction - NCEP') e pelo Centro Meteorológico Canadense ('Canadian Meteorological Centre - CMC') ([17]; [18]). Uma versão quadridimensional do *Método 3D-Var*, chamado *Método 4D-Var*, é usada pelo Centro Europeu de Previsões do Tempo de Médio Prazo ('European Centre for Medium Range Weather Forecasts - ECMWF'). Teoricamente, *4D-Var* pode ser pensado como minimização da soma das funções custo básicas *3D-Var* para todos os passos de tempo no intervalo temporal de assimilação de dados.

Lorenç [34] em 1986 usou argumentos bayesianos para derivar equações idealizadas com o objetivo de encontrar a melhor estimativa para a atmosfera prevista numericamente. Ele mostrou que *IO*, *Filtro de Kalman* e *Método Variacional* são métodos relacionados à estimativa idealizada, e portanto métodos relacionados entre si.

Outros métodos determinísticos incluem *Relaxação Dinâmica* e *Transformada de Laplace* [10]. Ambos supõem que o modelo de previsão seja perfeito. O primeiro usa o próprio modelo para amortecer e praticamente eliminar ondas indesejadas ao longo da integração. O segundo elimina diretamente no domínio espectral modos de vibração indesejados. Este último método será explicado em mais detalhes no Capítulo 2, juntamente com o Filtro de Kalman.

Uma nova abordagem para procedimento de assimilação, um filtro estendido adaptativo de Kalman (algoritmo de Kuga e Rios Neto), foi testado e comparado neste trabalho com assimilação usando Transformada de Laplace em sistemas dinâmicos

não lineares [46]: o sistema caótico de Lorenz ([35]; [36]), e o modelo meteorológico DYNAMO - uma versão simplificada das equações de água rasa ([38]; [39]).

Um novo método para calcular uma função de assimilação também foi proposto neste trabalho, onde esta função é implementada por uma Rede Neural Artificial (RNA) ([48], [47]). O uso de Redes Neurais, especificamente perceptrons multicamadas, para assimilação de dados foi desenvolvido. Entre as incontáveis aplicações do perceptron multicamadas em ciência e tecnologia em geral, as aplicações em ciências atmosféricas são relatadas em [16]:

- previsão de qualidade do ar, tempestade, chuva de monção indiana, anomalias de chuva brasileira, radiação solar diária, danificação de colheita por ozônio, dispersão atmosférica de poluentes, atividade solar, e níveis de monóxido de carbono devido a emissões de veículos numa interseção urbana;
- aproximação de funções para modelar a relação entre concentração de ozônio superficial horária e várias variáveis meteorológicas locais numa costa, funções de transferência não-lineares na recuperação de parâmetros geofísicos a partir de dados remotamente sensoriados, e a função de transferência para recuperação de velocidade de vento de superfície por sensor imageador especial de microondas;
- classificação de imagens de satélite para distinguir entre nuvens e gelo ou neve, de nuvens, padrões de circulação atmosféricos, coberturas de terra, linhas de convergência de imagens de radar, e fenômenos associados a relâmpagos.

Esta abordagem, assimilação de dados usando redes neurais, requereu treinamento de perceptron multicamadas para simular assimilação por Filtragem de Kalman. Resultados preliminares foram obtidos com sistemas caóticos de ordem baixa [47]. Três exemplos foram estudados. Primeiro, assimilação de dados usando Filtro de Kalman Estendido Adaptativo num sistema de Hénon caótico que foi usado para treinamen-

to da rede. Segundo, assimilação de dados usando Filtro de Kalman Estendido num sistema de Lorenz caótico foi usado para fornecer exemplos para treinamento da rede. Terceiro, assimilação de dados usando um Filtro de Kalman Estendido Adaptativo no modelo meteorológico DYNAMO foi usado para treinar a rede.

Nas próximas seções a tese está organizada como segue: no Capítulo 2 dois métodos típicos para assimilação contínua de dados são sucintamente revisados, nominalmente Filtragem por Transformada de Laplace e de Kalman, incluindo o Filtro de Kalman Estendido Adaptativo de Kuga-R.-Neto; no Capítulo 3 há uma revisão de redes neurais artificiais (RNA) e a apresentação de um algoritmo de treinamento modificado; no Capítulo 4 são descritos os sistemas não-lineares nos quais as novas abordagens foram testadas, sistemas caóticos de Hénon e Lorenz e o modelo meteorológico DYNAMO; no Capítulo 5 são comparados os resultados numéricos obtidos usando-se Transformada de Laplace e o Filtro de Kalman Estendido Adaptativo; no Capítulo 6 são mostrados os resultados numéricos que foram obtidos usando-se Perceptrons Multicamadas; e Capítulo 7 contém um resumo dos resultados, alguns comentários sobre as novas abordagens e sugestões para trabalhos futuros.

CAPÍTULO 2

ALGUNS MÉTODOS PARA ASSIMILAÇÃO DE DADOS METEOROLÓGICOS

2.1 Assimilação de Dados

Matematicamente, o processo de assimilação de dados pode ser descrito como tendo dois passos:

$$\text{Passo de previsão:} \quad \vec{w}_n^f = \vec{F}[\vec{w}_{n-1}^a] ;$$

$$\text{Passo de análise:} \quad \vec{w}_n^a = \vec{w}_n^f + \vec{d}_n ;$$

onde \vec{w}_n representa vetor de variáveis de estado no passo temporal n , $\vec{F}[\cdot]$ é o modelo (de previsão) matemático, sobrescritos f e a denotam valores previstos e analisados, respectivamente, e \vec{d}_n é a inovação de análise. A inovação de análise contém informação derivada de observações.

O Ciclo de Assimilação de Dados

O *ciclo de assimilação de dados* na Previsão Numérica do Tempo pode ser descrito como tendo quatro subcomponentes:

1) *Controle de qualidade*

Algoritmos de controle de qualidade são projetados para rejeitar ou modificar dados ruins. Primeiro, verifica-se se há erros de codificação e se há problemas de localização de sensores. Em seguida, examina-se se os dados são fisicamente razoáveis. O controle de qualidade remanescente é comparar cada observação com sua vizinha, requerendo-se consistência espacial e temporal.

2) *Análise objetiva*

É um procedimento automático para estimar as variáveis atmosféricas dependentes numa grade regular bidimensional ou tridimensional usando os dados disponíveis vindos da rede de observação irregularmente espaçada.

3) *Inicialização*

Calcula-se condição inicial de integração a partir do resultado da análise objetiva removendo componentes espúrias de alta frequência, procurando evitar que esta condição inicial excite oscilações espúrias (p. ex. de ondas de gravidade inerciais) na previsão do tempo.

4) *Previsão curta para preparar a estimativa a priori*

Usa-se um modelo numérico (geralmente sofisticado) chamado *modelo de assimilação* para preparar a *estimativa a priori* do próximo tempo de observação. Este modelo deve incluir as parametrizações necessárias para assegurar que, *se não for atualizado com novas observações*, o clima por ele gerado aproximar-se-á do "verdadeiro". Isso assegura que na falta persistente de dados a estimativa a priori produzida pelo modelo assimilador permaneça fisicamente plausível.

Um esboço esquemático do ciclo de assimilação de dados *intermitente* é mostrado na Figura 2.1. A cada 6 horas todas as observações num intervalo de + ou - 3 horas centrado na sexta hora são coletadas e passam pelo controle de qualidade. Estimativas a priori das variáveis de estado nos pontos de grade de análise são obtidas integrando-se o modelo assimilador, usando a análise objetiva de 6 horas atrás como condição inicial. Estimativas a priori das variáveis de estado nas estações de observação são obtidas usando-se interpolação matemática. As estimativas a priori nas estações de observação são subtraídas das observações para produzir os *incrementos observacionais*. *Incrementos de análise* na grade de análise objetiva são obtidos por análise objetiva dos incrementos observacionais. Os valores analisados são simplesmente a soma dos incrementos de análise com as estimativas a priori nos pontos de grade da análise.

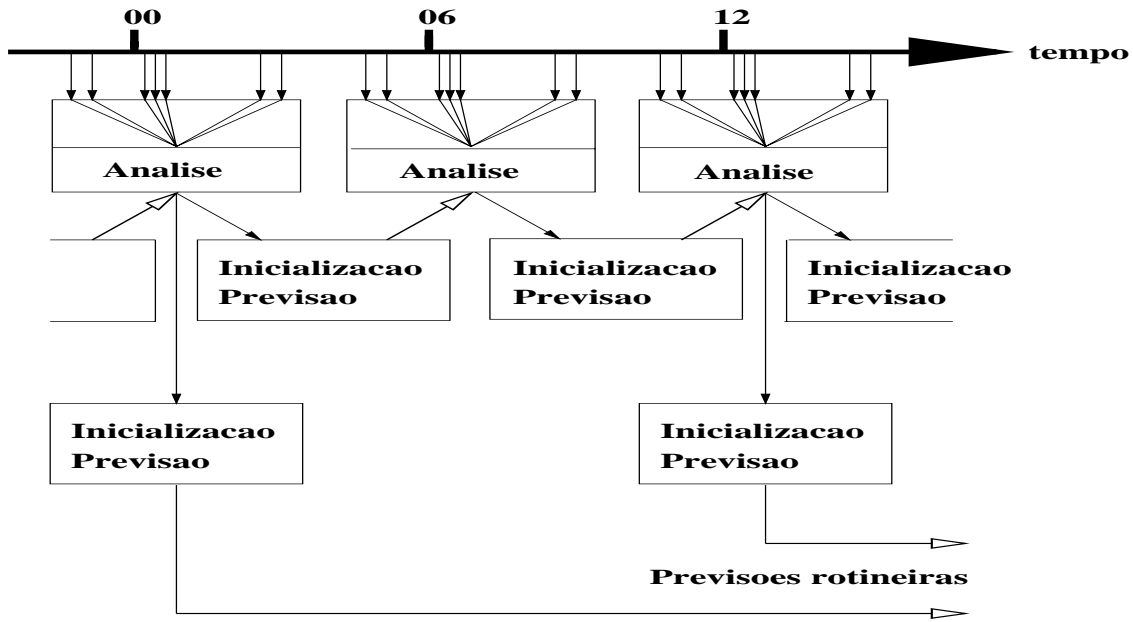


Fig. 2.1 - Diagrama esquemático do ciclo de assimilação de dados intermitente.

O passo de inicialização é então executado e o modelo de previsão é executado a partir deste estado analisado/inicializado para produzir uma previsão de 6 horas. Esta previsão de 6 horas é então usada como estimativa a priori para a próxima análise. As "previsões rotineiras" mostradas na parte inferior da Figura 2.1 são as previsões mais longas da componente da previsão e são a principal motivação para o processo todo.

O ciclo de assimilação que aqui adotamos é mostrado na Figura seguinte 2.2. Este tipo de assimilação de dados é chamada *contínua*. Nela as observações são assimiladas na mesma taxa em que são observadas mas um pouco depois do tempo real para permitir que os dados sejam comunicados e processados. O estado atmosférico simulado pelo modelo numérico é continuamente ajustado para aproximar as novas observações. Uma previsão rotineira pode começar em qualquer tempo durante o ciclo.

Pode-se agora explicar sucintamente alguns métodos de assimilação de dados, nominalmente Interpolação Ótima, Relaxação Dinâmica e o Método Variacional. Transformada de Laplace e Filtro de Kalman serão descritos mais detalhadamente porque

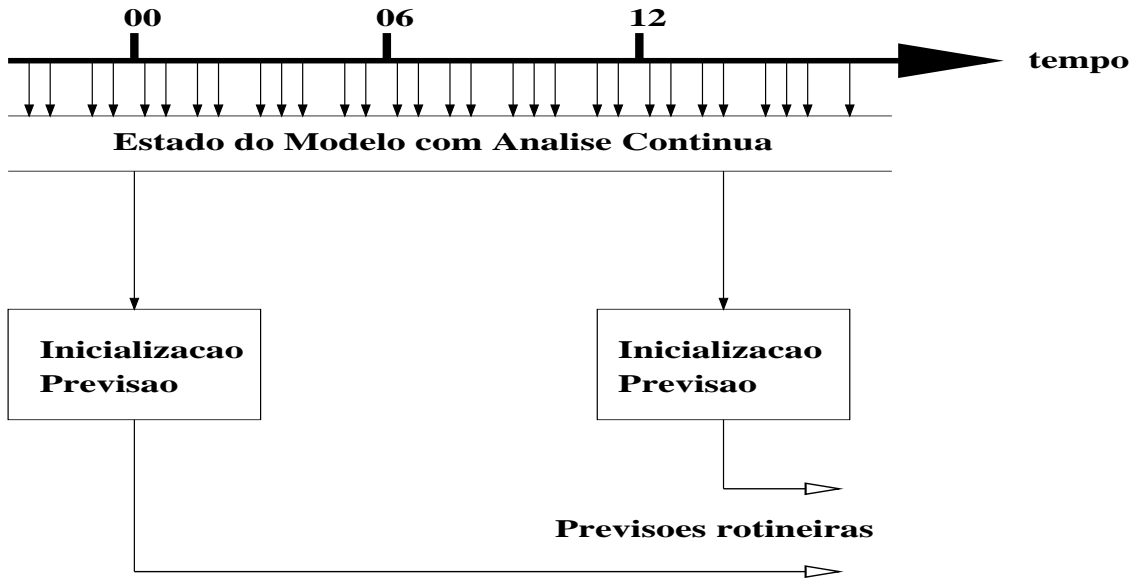


Fig. 2.2 - Diagrama esquemático do ciclo de assimilação de dados contínua.

foram implementados e comparados com o método da Rede Neural.

2.1.1 Interpolação Ótima

A correção devido à disponibilidade de observações z_n é feita estatisticamente. *Modernamente* a IO pode ser expressa como uma estimativa w_n^a de w_n inspirando-se no critério de erro com mínima variância ([10] páginas 180-184, 364-365; [56] páginas 114-119):

$$\vec{w}_n^a = \vec{w}_n^f + \hat{\mathbf{G}}_n \left(z_n - \mathbf{H}_n \vec{w}_n^f \right) \quad , \quad (2.1)$$

onde a matriz $\hat{\mathbf{G}}_n$ é calculada baseando-se em avaliações estatísticas e vínculos dinâmicos, e \mathbf{H}_n é a matriz que mapeia w_n^f na rede de observação.

A relação entre este método, *Análise Variacional em 4 Dimensões* e *Filtro de Kalman* é explicada em [34].

2.1.2 Relaxação Dinâmica

Chamada também Relaxação Newtoniana, a Relaxação Dinâmica é um procedimento de assimilação de dados contínuo no tempo ([56] páginas 111-114). Neste caso as observações são introduzidas nas equações governantes por meio de forçamento, de modo a relaxar os campos na direção das observações. Relaxação dinâmica é usada durante o período de tempo conhecido como período de pré-previsão ("pre-forecast"), de modo que ao final deste período a solução esteja o mais próximo que se queira das observações. A partir daí então será possível produzir previsões.

2.1.3 Análise Variacional em 4 Dimensões

Suponha-se que o modelo de previsão seja

$$\vec{w}_{n+1} = \mathbf{F}_n \vec{w}_n \quad \text{para } 0 \leq n \leq N - 1 \quad (2.2)$$

onde \mathbf{F}_n é matriz quadrada dependente de n . Além disso, suponha-se que o modelo de observação seja

$$\vec{z}_n = \mathbf{H}_n w_n + \mu_n \quad (2.3)$$

para todos n . A amostragem é realizada por:

$$\vec{r}(t_n) = \vec{r}_n \equiv \vec{z}_n - \mathbf{H}_n \vec{w}_n^f \quad (2.4)$$

Conforme (2.2) pode-se escrever

$$\vec{w}_{n+1}^f = \mathbf{F}_n \vec{w}_n^a \quad (2.5)$$

$$\vec{w}_{n+1}^a = \left(\prod_{k=0}^n \mathbf{F}_k \right) \vec{w}_0^a \quad (2.6)$$

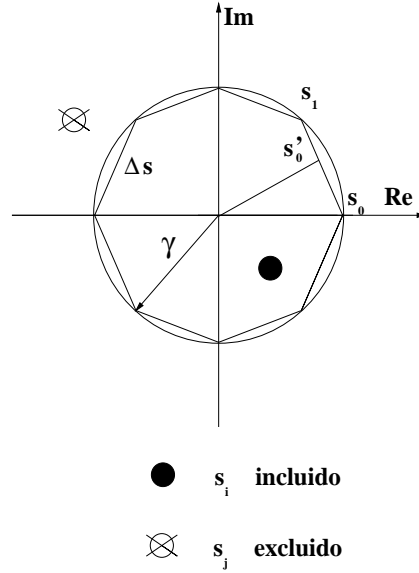


Fig. 2.3 - Aproximação de integral de contorno para transformada de Laplace inversa.

Defina-se o funcional

$$J = \frac{1}{2} \sum_{n=0}^N \vec{r}_n^T \mathbf{R}_n^{-1} \vec{r}_n \quad , \quad (2.7)$$

onde \mathbf{R}_n são matrizes representando a confiabilidade de cada inovação \vec{r}_n . O problema é encontrar \vec{w}_0^a que minimize (2.7) para então calcular \vec{w}_{n+1}^a usando (2.6).

A relação entre este método, *IO* e *Filtro de Kalman* é explicada em [34].

2.2 Transformada de Laplace

Considere o modelo de previsão seguinte:

$$\frac{d\vec{w}(t)}{dt} + \mathbf{L} \vec{w}(t) + \vec{N}[\vec{w}(t)] = 0 \quad (2.8)$$

onde \mathbf{L} é um operador linear e $\vec{N}(\vec{w})$ é uma função não-linear. Supõe-se que o termo não-linear varie devagar no intervalo de tempo $[t_n, t_n + \Delta t]$. Portanto, a transformada de Laplace (TL) deste termo é dada por

$$\mathcal{L} \{ \vec{N}[\vec{w}(t)] \} = \vec{N}(t_n) s^{-1} \quad . \quad (2.9)$$

Desta maneira a TL da Equação (2.8) é

$$\vec{W}(s) \simeq (sI + \mathbf{L})^{-1} [\vec{w}(t_n) - \vec{N}(t_n) s^{-1}] . \quad (2.10)$$

para o modelo de observação

$$\vec{z}_n = \mathbf{H}_n \vec{w}_n + \vec{v}_n , \quad (2.11)$$

onde \vec{v}_n é ruído e \mathbf{H}_n representa o sistema de observação. A assimilação é feita através da recursão de *primeira* ordem ([10] páginas 363-391, [40] páginas 4-5,[39],[38])

$$\vec{w}_{n+1} = \frac{1}{2\pi i} \oint_{C^*} (sI + \mathbf{L})^{-1} [\vec{w}_n - \vec{N}(\vec{z}_n) s^{-1}] e^{st_{n+1}} ds . \quad (2.12)$$

ou da recursão de *segunda* ordem

$$\vec{w}_{n+1} = \frac{1}{2\pi i} \oint_{C^*} (sI + \mathbf{L})^{-1} [\vec{w}_{n-1} - \vec{N}(\vec{z}_n) s^{-1}] e^{st_{n+1}} ds , \quad (2.13)$$

onde C^* é um contorno incluindo as soluções fisicamente consistentes e excluindo as soluções indesejadas. Na prática considera-se C^* um círculo de raio γ , e a integral de linha é aproximada com um polígono (veja Figura 2.3):

$$\oint_{C^*} \vec{F}(s) ds \simeq \frac{1}{\sigma} \sum_k \vec{F}(s_k) \Delta s_k . \quad (2.14)$$

onde σ é o fator de correção $\sigma = \tan(\pi/N)/(\pi/N)$ (veja [38]).

2.3 Filtro de Kalman

Uma breve descrição do Filtro de Kalman será feita aqui. Primeiramente, a versão linear do filtro será apresentada. A adaptação da técnica do filtro de Kalman para um sistema não-linear é chamada *Filtro de Kalman Estendido*, e será descrito mais adiante.

Seja o modelo de previsão

$$\vec{w}_{n+1} = \mathbf{F}_n \vec{w}_n + \vec{\mu}_n \quad , \quad (2.15)$$

onde \mathbf{F}_n é uma descrição matemática do sistema (linear), $\vec{\mu}_n$ uma forçante estocástica (chamada ruído de modelagem dinâmica) e o modelo de observação

$$\vec{z}_n = \mathbf{H}_n \vec{w}_n + \vec{\nu}_n \quad , \quad (2.16)$$

onde $\vec{\nu}_n$ é ruído. As hipóteses típicas de gaussianidade, média zero e ortogonalidade para os ruídos são adotadas. O termo \vec{w}_{n+1} é estimado através da recursão

$$\vec{w}_{n+1}^a = (\mathbf{I} - \mathbf{G}_{n+1} \mathbf{H}_{n+1}) \mathbf{F}_n \vec{w}_n^a + \mathbf{G}_{n+1} \vec{z}_{n+1} \quad (2.17)$$

onde \vec{w}_{n+1}^a é o estimador e \mathbf{G}_n é a matriz que minimiza o traço da matriz de covariância do erro de previsão, isto é, a soma dos quadrados dos erros de previsão em cada componente de \vec{w}_{n+1} ([19], páginas 108-109):

$$J_{n+1} = E \left\{ (\vec{w}_{n+1}^a - \vec{w}_{n+1})^T (\vec{w}_{n+1}^a - \vec{w}_{n+1}) \right\} \quad . \quad (2.18)$$

Este erro é denominado neste trabalho *variância total de erro*. Aqui calculou-se o Erro Quadrático Total (EQT) no instante n usando-se a expressão

$$EQT_n = (\vec{w}_n^a - \vec{w}_n)^T (\vec{w}_n^a - \vec{w}_n) \quad . \quad (2.19)$$

Para facilitar a análise dos resultados numéricos doravante o EQT também será denominado variância total de erro.

Nesta última Equação $E\{\cdot\}$ é o valor esperado estatístico. O algoritmo do *Filtro de Kalman Linear* (FKL) é mostrado na Figura a seguir 2.4, onde \mathbf{Q}_n é a covariância de $\vec{\mu}_n$, \mathbf{P}_n^f é a covariância do erro de previsão, \mathbf{R}_n é a covariância de $\vec{\nu}_n$ e \mathbf{P}_n^a é a covariância do erro de análise. Mais detalhes sobre teoria da estimação podem ser encontrados no apêndice.

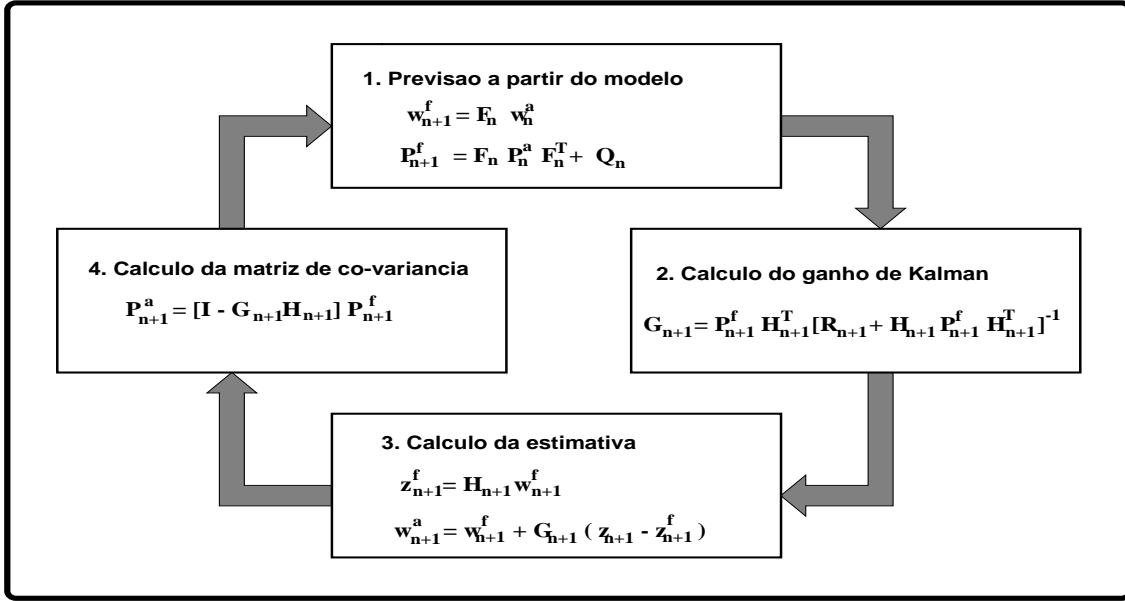


Fig. 2.4 - Diagrama esquemático do Filtro de Kalman Linear.

A assimilação é feita através da amostragem:

$$\vec{r}(t_n + \Delta t) = \vec{r}_{n+1} \equiv \vec{z}_{n+1} - \vec{z}_{n+1}^f = \vec{z}_{n+1} - \mathbf{H}_n \vec{w}_{n+1}^f \quad . \quad (2.20)$$

Note que nos passos n sem assimilação $\vec{r}_{n+1} = \vec{0}$.

A adaptação mais usada do Filtro de Kalman para uso com sistemas não-lineares é chamada *Filtro de Kalman Estendido* (EKF) ([27]; [6]). Agora \vec{w}_{n+1} será estimada minimizando a expressão (2.18) a partir do modelo

$$\vec{w}_{n+1} = \vec{f}_n(\vec{w}_n) + \vec{\mu}_n \quad , \quad (2.21)$$

$$\vec{z}_n = \vec{h}_n(\vec{w}_n) + \vec{\nu}_n \quad , \quad (2.22)$$

onde $\vec{f}_n(\vec{w}_n)$ e $\vec{h}_n(\vec{w}_n)$ são funções não-lineares de \vec{w}_n . A aproximação seguinte é usada ([1], página 195):

$$\vec{f}_n(w_n) \simeq \vec{f}_n(\vec{w}_n^a) + \mathbf{F}_n (\vec{w}_n - \vec{w}_n^a) \quad , \quad (2.23)$$

$$\vec{h}_n(\vec{w}_n) \simeq \vec{h}_n(\vec{w}_n^f) + \mathbf{H}_n (\vec{w}_n - \vec{w}_n^f) \quad , \quad (2.24)$$

onde \mathbf{F}_n e \mathbf{H}_n são dadas por

$$\mathbf{F}_n \equiv \left. \frac{\partial \vec{f}_n}{\partial \vec{w}} \right|_{\vec{w}=\vec{w}_n^a} \quad , \quad (2.25)$$

$$\mathbf{H}_n \equiv \left. \frac{\partial \vec{h}_n}{\partial \vec{w}} \right|_{\vec{w}=\vec{w}_n^f} \quad . \quad (2.26)$$

Os passos 1 e 3 do FKL são então alterados para

$$\vec{w}_{n+1}^f = \vec{f}_n(\vec{w}_n^a) \quad , \quad (2.27)$$

$$\vec{z}_{n+1}^f = \vec{h}_{n+1}(\vec{w}_{n+1}^f) \quad . \quad (2.28)$$

Na verdade a filtragem de Kalman não é feita exatamente como descrito aqui, em vista da dimensão do problema de assimilação de dados em meteorologia, que tipicamente envolve $10^6 - 10^7$ variáveis de estado. \mathbf{P}_{n+1}^f não é dada por $\mathbf{F}_n \mathbf{P}_n^a \mathbf{F}_n^T + \mathbf{Q}_n$, mas parametrizada baseando-se em avaliações estatísticas e vínculos dinâmicos ([56], páginas 114-119). Simplificações alternativas podem ser feitas na dinâmica de evolução de \mathbf{P}_n^f ou no próprio modelo de previsão [55].

Há também versões em que as matrizes \mathbf{P}_n^f e/ou \mathbf{P}_n^a são estimadas usando médias de *ensemble* ([15],[27]; [6]; [43]). Bürger e Cane [5] calcularam diretamente w_n^a usando média de *ensemble*.

Visando assimilação que incorpore não apenas a observação mais recente \vec{z}_n mas

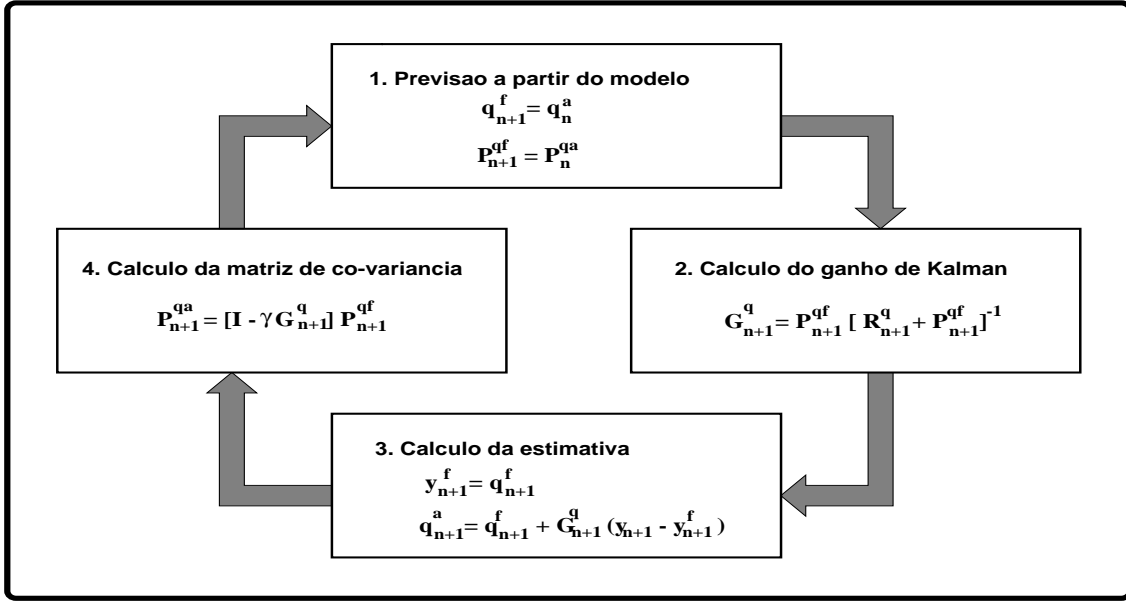


Fig. 2.5 - Diagrama esquemático do filtro de Kalman que estima

$$q_{n+1}.$$

além disso informações do passado $t_m < t_n$ foi proposta a *assimilação retrospectiva* [58], [57].

A relação entre este método, *Análise Variacional 4-D* e *IO* é explicada em [34].

2.3.1 Um Filtro de Kalman Adaptativo: Algoritmo de Kuga e Rios Neto

No FKE pode ocorrer que os erros de estimação reais excedam os valores previstos pela variância do erro após aplicar-se o filtro ([24]; [25]; [29]; [30]). Se houver a suspeita de que o erro de modelagem esteja sendo subestimado, uma solução é parametrizar \mathbf{Q}_n e usar algum método para estimar o seu verdadeiro valor. A idéia de estimar este erro, filtrar adaptativamente, foi introduzida em Assimilação de Dados por Dee et al. [12]. Naquele trabalho Dee usou um método elaborado por Bélanger baseado na hipótese de que as matrizes de covariância de erro de modelagem e observação podem ser parametrizados conforme:

$$\mathbf{Q} = \sum_{i=1}^M \alpha_i Q_i, \quad \mathbf{R} = \sum_{i=1}^M \alpha_i R_i \quad . \quad (2.29)$$

Os erros são então estimados a partir da sequência de inovações usando um filtro de Kalman. Em trabalho posterior Dee parametrizou apenas o erro de modelagem \mathbf{Q} e usou o critério da máxima verossimilhança para estimá-lo [13], [4], [14]. Outro trabalho usando este método é o de Mitchell e Houtekamer [43]. Como não será usada adaptação no filtro linear, de agora em diante qualquer FKE Adaptativo será denominado, simplesmente, *Filtro de Kalman Adaptativo* (FKA).

Aqui uma alternativa aos métodos de Bélanger [12] e Burg et al. [4] para estimar \mathbf{Q}_n é mostrada.

O requerimento básico para o procedimento alternativo é a imposição de consistência entre os resíduos e suas estatísticas. Com isto pode-se desenvolver equações para estimar a variância do ruído dinâmico. Lembrando que o resíduo previsto é definido como

$$\vec{r}_{n+1} \equiv \vec{z}_{n+1} - \vec{z}_{n+1}^f \quad , \quad (2.30)$$

onde \vec{r}_{n+1} é separado componente a componente. O ruído *verdadeiro* é agora definido como o vetor $\vec{r}(n+1 | \mathbf{R} \equiv \mathbf{0})$ (lê-se " \vec{r} no instante $n+1$ condicionado a $\mathbf{R} \equiv \mathbf{0}$ ", notação usada em probabilidade) cujos componentes são:

$$r_i(n+1 | \mathbf{R} \equiv \mathbf{0}) \equiv r_i(n+1) - \nu_i(n+1) \quad . \quad (2.31)$$

$\vec{r}(n+1 | \mathbf{R} \equiv \mathbf{0})$ representa o resíduo que seria obtido se não houvesse erros de medida. A hipótese fundamental de consistência é que o ruído $\vec{r}(n+1 | \mathbf{R} \equiv \mathbf{0})$ seja gaussiano de média nula para que também \vec{r}_{n+1} o seja. A variância do ruído dinâmico $\vec{\mu}_n$ é calculada de maneira que a probabilidade de que seja o ruído dinâmico verdadeiro seja máxima. Isto leva ao sistema de equações ([24]; [25], páginas 311-312):

$$r_i^2(n+1 | \mathbf{R} \equiv \mathbf{0}) = E \left\{ r_i^2(n+1 | \mathbf{R} \equiv \mathbf{0}) \right\} \quad . \quad (2.32)$$

Adicionalmente, as covariâncias \mathbf{Q}_n e \mathbf{R}_n são supostas diagonais. Finalmente, a variância \vec{q}_n do ruído é estimada através de um FKL secundário (Figura 2.5) usando o modelo desenvolvido por Kuga e Rios Neto [52]:

$$\vec{q}_{n+1} = \vec{q}_n \quad (2.33)$$

$$r_i(n+1) \equiv z_i(n+1) - z_i^f(n+1) \quad (2.34)$$

$$d_i(n+1) = \min \left[r_i(n+1), \alpha \sqrt{R_{ii}(n+1)} \right] \quad (2.35)$$

$$\vec{y}_{n+1} = \vec{q}_{n+1} + \nu^{\vec{q}}_{n+1} \quad (2.36)$$

$$\begin{aligned} &\equiv \begin{bmatrix} d_1^2(n+1) \\ \vdots \\ d_m^2(n+1) \end{bmatrix} + \begin{bmatrix} R_{11}(n+1) \\ \vdots \\ R_{mm}(n+1) \end{bmatrix} - \\ &\quad \begin{bmatrix} H_1(n) & \cdots & \vec{0} \\ \vdots & \ddots & \vdots \\ \vec{0} & \cdots & H_m(n) \end{bmatrix} \mathbf{F}_n \mathbf{P}_n^a \mathbf{F}_n^T \begin{bmatrix} H_1^T(n) \\ \vdots \\ H_m^T(n) \end{bmatrix} \end{aligned} \quad (2.37)$$

$$R_{ij}^q(n+1) = \begin{cases} 4 d_i^2(n+1) R_{ii}(n+1) + 2 R_{ii}^2(n+1), & \text{se } i = j \\ 0, & \text{se } i \neq j \end{cases} \quad (2.38)$$

$$\vec{q}_{n+1} = \left[q_1(n) \quad \cdots \quad q_m(n) \right]^T \quad (2.39)$$

onde \mathbf{H}_i é a i -ésima linha de \mathbf{H} , α é uma constante limitante tal que $0 < \alpha \leq 3$ determinada experimentalmente para fazer a variância do erro de assimilação previsto J_{n+1} (Equação (2.18)) ser aproximadamente igual à variância do erro verdadeiro $\|\vec{w}_{n+1}^a - \vec{w}_{n+1}\|_2$, $\nu^{\vec{q}}$ é "pseudo-ruído" com covariância $\mathbf{R}^q(n+1)$ definido para a estimação de \vec{q}_{n+1} a partir da "pseudo-observação" \vec{y}_{n+1} , e γ é constante tomando o valor 1 para a hipótese de ruído estacionário ou 0 para a contrária.

Após calcular \bar{q}_{n+1}^a no passo 1 do algoritmo principal de Kalman usando o filtro secundário (Figura 2.5) obtêm-se a matriz $\hat{\mathbf{Q}}_n$ para ser usada (no lugar de \mathbf{Q}_n^a):

$$\hat{q}_i(n+1) = \begin{cases} 0, & \text{se } q_i^a(n+1) < 0 \\ q_i^a(n+1), & \text{se } q_i^a(n+1) \geq 0 \end{cases} \quad (2.40)$$

e monta-se a matriz $\hat{\mathbf{Q}}_n$:

$$\hat{\mathbf{Q}}_n = \text{diag} \left[\hat{q}_1(n+1) \quad \cdots \quad \hat{q}_m(n+1) \right] . \quad (2.41)$$

$\hat{q}_i(n+1)$ é usada ao invés de $q_i^a(n+1)$ porque variâncias não podem ser negativas.

CAPÍTULO 3

REDES NEURAIIS MULTICAMADAS DE PERCEPTRONS

3.1 Introdução: Que é uma Rede Neural

Segundo Haykin ([22], páginas 1-6) o cérebro é um computador (sistema de processamento de informações) altamente complexo, não-linear e paralelo [3]. Na sua forma mais geral, uma rede neural é uma máquina projetada para *modelar* a maneira na qual o cérebro desempenha uma tarefa específica ou função de interesse. Nosso interesse se confina a uma classe importante de redes neurais que desempenham computações úteis por um processo de aprendizado. Pode-se então definir uma rede neural como:

Uma rede neural é um processador massivamente paralelo que tem propensão natural para armazenar conhecimento de experiência e fazê-lo disponível para uso. Lembra o cérebro em dois aspectos:

- 1) Conhecimento é adquirido pela rede através de um processo de aprendizado;*
- 2) Pesos de conexão entre os neurônios conhecidos como pesos sinápticos são usados para armazenar o conhecimento.*

O procedimento usado para desempenhar o processo de aprendizado é chamado *algoritmo de aprendizagem*, cuja função é modificar os pesos sinápticos da rede numa maneira ordenada para atingir um objetivo de projeto desejado.

A modificação de pesos sinápticos fornece o método tradicional para o projeto de redes neurais. Entretanto, é também possível que uma rede neural modifique sua própria topologia, o que é motivado pelo fato de que neurônios no cérebro humano podem morrer e que novas conexões sinápticas podem crescer.

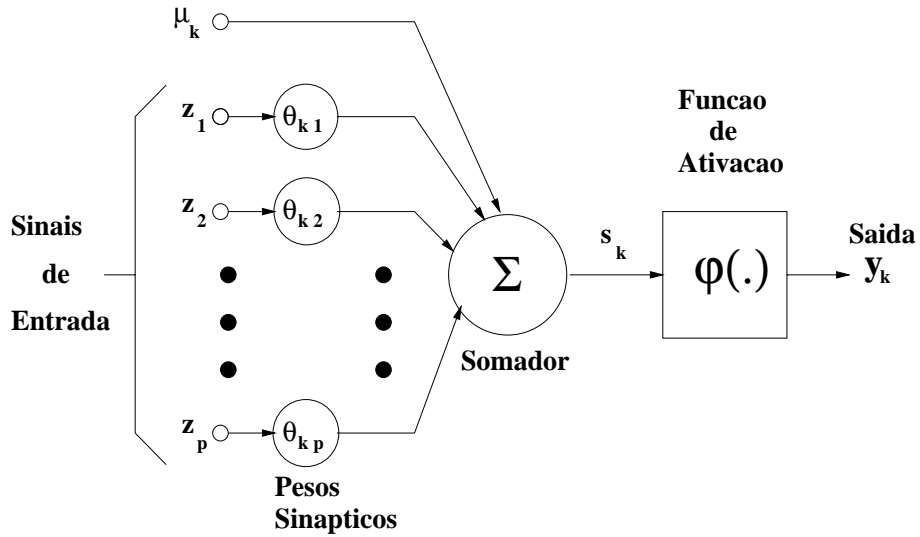


Fig. 3.1 - Modelo não-linear de um neurônio

3.2 Modelo de um Neurônio

Conforme Haykin ([22], páginas 8-13), um neurônio é uma unidade de processamento de informações que é fundamental para a operação de uma rede neural. A Figura 3.1 mostra o *modelo* para um neurônio. Podemos identificar três elementos básicos do modelo neural:

- 1) Um conjunto de *sinapses*, cada uma das quais é caracterizada por um peso correspondente. Especificamente, um sinal z_i na entrada da sinapse i conectada ao neurônio k é multiplicado pelo peso sináptico θ_{ik} ;
- 2) Um *somador* para somar os sinais de entrada, ponderados pelas sinapses respectivas do neurônios;
- 3) Uma *função de ativação* para limitar a amplitude da saída de um neurônio. Tipicamente, a faixa de amplitude da saída de um neurônio é o intervalo $[0, 1]$ ou alternativamente $[-1, 1]$.

O modelo de um neurônio mostrado na Figura 3.1 também inclui um limiar aplicado externamente μ_k que tem o efeito de acrescentar um grau de liberdade a cada neurônio.

Alguns Tipos de Função de Ativação

A função de ativação, denotada por $\varphi(\cdot)$, define a saída de um neurônio em termos do nível de atividade na sua entrada. Podemos dar como exemplo quatro tipos básicos de funções de ativação:

1) Função Linear

$$\varphi(s) = s \quad (3.1)$$

Um neurônio com esta função de ativação é chamado *linear*.

2) Função Signum

$$\varphi(s) = \begin{cases} 1, & \text{se } s > 0 \\ 0, & \text{se } s = 0 \\ -1, & \text{se } s < 0 \end{cases} \quad (3.2)$$

Um neurônio assim é chamado *signum*.

3) Função Tangente Hiperbólica

$$\varphi(s) = \tanh(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}} \quad (3.3)$$

Um neurônio assim é chamado *tangente hiperbólica*.

4) Função Logística

$$\varphi(s) = \frac{1}{1 + e^s} \quad (3.4)$$

Um neurônio assim é chamado *logístico*.

3.3 Redes de Perceptrons com Múltiplas Camadas (ou Multicamadas)

Em geral, conforme Haykin ([22], páginas 18-22) podemos identificar quatro diferentes classes de arquiteturas de rede: *Redes de Perceptrons com Camada Única*, *Redes de Perceptrons com Múltiplas Camadas (ou Multicamadas)*, *Redes Recorrentes* e *Redes Reticuladas*. Explicaremos apenas a segunda, que foi usada neste trabalho por ser a mais usada com êxito ([16],[49]). A primeira é um caso particular da segunda.

A segunda classe de rede neural se distingue por possuir uma ou mais *camadas escondidas*, cujos nós computacionais são correspondentemente chamados *neurônios escondidos*. A função dos neurônios escondidos é intervir entre a entrada externa e a saída da rede. Adicionando-se uma ou mais camadas escondidas, a rede consegue extrair estatísticas de ordem mais alta, porque (grosso modo falando) a rede adquire uma perspectiva *global* apesar de sua conectividade local em virtude do conjunto extra de conexões sinápticas e da dimensão extra de interações neurais.

Os nós fontes na camada de entrada da rede fornecem elementos respectivos do padrão de ativação (vetor de entrada), que constituem os sinais de entrada aplicados aos neurônios na segunda camada. Os sinais de saída da segunda camada são usados como entradas para a terceira camada, e assim por diante para o resto da rede. O conjunto de sinais de saída dos neurônios na camada (final) de saída da rede constitui a resposta geral da rede ao padrão de ativação fornecido pelos nós fontes na camada de entrada (primeira). A Figura 3.2 a seguir ilustra o "layout" de uma rede neural para o caso de uma única camada escondida.

Os detalhes específicos envolvidos no projeto de um perceptron naturalmente dependem da aplicação de interesse ([22], página 220). Na aproximação de funções não-lineares em particular, que foi o caso de interesse, Haykin sugere que todos os neurônios na camada de *saída* sejam lineares. Isto porque, supondo que todo o conteúdo de informação dos vetores de treinamento é transferido para os pesos sinápticos e limiares dos neurônios, o erro de aproximação seria idealmente modelado como *ruido branco gaussiano*. A propriedade de ruído branco significa que o espectro de potência do erro seja constante e portanto independente de frequência. A propriedade gaussiana significa que a amplitude do erro tenha como domínio $(-\infty, \infty)$. Portanto a faixa dinâmica da saída da rede precisa ser similarmente $(-\infty, \infty)$. Daí a sugestão para que os neurônios na camada de saída da rede sejam todos lineares.

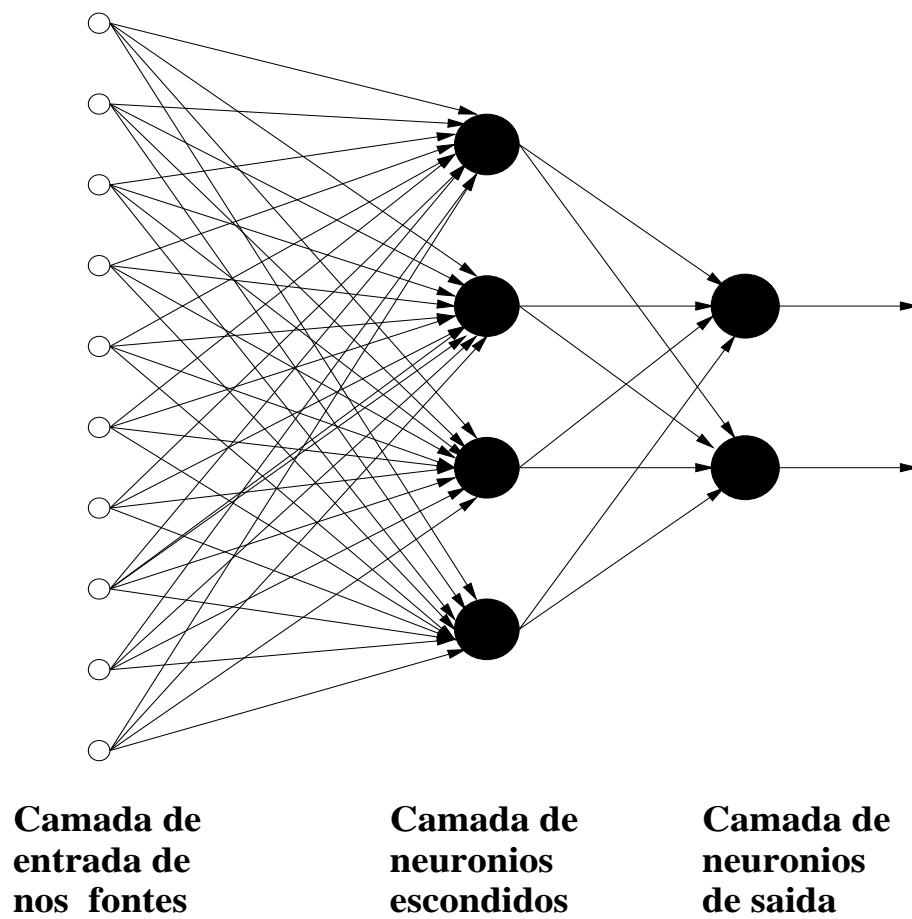


Fig. 3.2 - Rede de Perceptrons com uma camada escondida e camada de saída

3.4 Processo de Aprendizado

Introdução

Definimos aprendizado no contexto de redes neurais como se segue ([22], páginas 45-46):

Aprendizado é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo contínuo de estimulação pelo ambiente em que a rede está imersa. O tipo de aprendizado é determinado pela maneira em que as mudanças dos parâmetros ocorrem.

Esta definição do processo de aprendizado implica a seguinte seqüência de eventos:

- 1) A rede neural é *estimulada* por um ambiente;
- 2) A rede neural *muda* como resultado dessa estimulação;
- 3) A rede neural *responde de maneira nova* ao ambiente, por causa das mudanças que ocorreram na sua estrutura interna.

Para ser específico, considere um par de sinais nodais x_i e y_k conectados por um peso sináptico θ_{ik} , como mostrado na Figura 3.1. O sinal x_i representa a atividade interna do neurônio k . Seja $\theta_{ik}(m)$ o valor do peso sináptico no passo m . No passo m um ajuste $\Delta\theta_{ik}(m)$ é aplicado ao peso sináptico $\theta_{ik}(m)$, gerando o valor atualizado $\theta_{ik}(m+1)$. Podemos portanto escrever

$$\theta_{ik}(m+1) = \theta_{ik}(m) + \Delta\theta_{ik}(m) \quad (3.5)$$

onde $\theta_{ik}(m)$ e $\theta_{ik}(m+1)$ podem ser vistos como os valores antigo e novo do peso sináptico θ_{ik} , respectivamente.

Um conjunto prescrito de regras bem-definidas para a solução de um problema de aprendizado é chamado um *algoritmo de aprendizado*. Não há um algoritmo único para o projeto de redes neurais, mas uma "caixa de ferramentas" representada por uma variedade diversa de algoritmos de aprendizado, cada qual oferecendo suas

vantagens. Basicamente, os algoritmos de aprendizagem diferem uns dos outros pela maneira que o ajuste $\Delta\theta_{ik}$ ao peso sináptico θ_{ik} é formulado. Há quatro regras de aprendizado básicas ([22], páginas 46-57),

- 1) *aprendizado por erro-correção*,
- 2) *aprendizado hebbiano*,
- 3) *aprendizado competitivo* e
- 4) *aprendizado de Boltzmann*,

Sendo a rede de perceptrons multicamada a rede mais popular, será descrita apenas a regra de aprendizado geralmente empregada para seu treinamento e usada nesta tese, a primeira:

Aprendizado por Correção de Erro

Seja $w_k(n)$ uma *resposta desejada* ou *resposta-alvo* para o neurônio k no exemplo n . Seja $w_k^f(n)$ o valor fornecido por este neurônio. A resposta $w_k^f(n)$ é produzida por um vetor *estímulo* $\vec{x}(n)$ aplicado à entrada da rede em que cada neurônio está imerso. O vetor de entrada $\vec{x}(n)$ e a resposta desejada $w_k(n)$ para o neurônio k constituem um *exemplo* particular apresentado à rede no passo n . Supõe-se que este exemplo e todos os outros possíveis exemplos são gerados por um ambiente que é probabilístico em sua natureza, mas a densidade de probabilidade subjacente é desconhecida.

Tipicamente, a resposta $w_k^f(n)$ fornecida pelo neurônio k é diferente da resposta desejada $w_k(n)$. Portanto, podemos definir um *signal de erro* como a diferença entre a resposta-alvo $w_k(n)$ e a resposta obtida $w_k^f(n)$, como mostrado por

$$e_k(n) \doteq w_k^f(n) - w_k(n) \quad . \quad (3.6)$$

O objetivo da aprendizagem por correção-de-erro é minimizar uma *função-custo* baseada no sinal de erro $e_k(n)$, tal que a resposta fornecida por cada neurônio de

saida na rede aproxime-se da resposta-alvo para aquele neurônio em algum sentido estatístico. O famoso algoritmo do tipo retropropagação ('backpropagation algorithm') [22], tipicamente usado para treinar perceptrons multicamadas ([16],[49]), adota como função-custo a norma quadrática de \vec{e} média ou total. De fato, uma vez que uma função custo seja escolhida, o aprendizado erro-correção é estritamente um problema de otimização. Nesta tese, especificamente, o treinamento é feito em modo lote ('batch mode'), que usa *todos* os exemplos ao mesmo tempo. O treinamento procura um conjunto de pesos θ e limiares μ que minimize o erro quadrático total

$$e_m = \sum_{n=1}^N \left\| \vec{F}_{\text{ANN}}(\vec{X}_n, \vec{\theta}, \vec{\mu}, m) - \vec{F}(\vec{X}_n) \right\|_2 \quad (3.7)$$

onde N é número de exemplos no conjunto de treinamento, \vec{X}_n é o vetor de entrada do exemplo n , $\vec{\theta}$ e $\vec{\mu}$ são os pesos e limiares da rede, \vec{F}_{ANN} é a aproximação e \vec{F} é o vetor de saída desejado.

Outro fator a ser considerado é a maneira em que uma rede neural se relaciona com seu ambiente. Nesse último contexto, falamos de um *paradigma de aprendizagem* fazendo referência a um modelo do ambiente em que a rede opera. Das três classes básicas de paradigmas de aprendizado ([22], páginas 57-66),

- 1) *aprendizado supervisionado*,
- 2) *aprendizado por reforço* e
- 3) *aprendizado não-supervisionado*,

descreveremos a primeira por ser o paradigma do algoritmo de treinamento tipicamente usado para treinar a rede de perceptrons multicamada, e por isso adotada neste trabalho.

Aprendizado Supervisionado

Um ingrediente essencial do *aprendizado supervisionado* é a disponibilidade de um professor *externo*, como indicado na Figura que se segue 3.3. Em termos conceituais, podemos pensar no professor como tendo conhecimento sobre o ambiente que

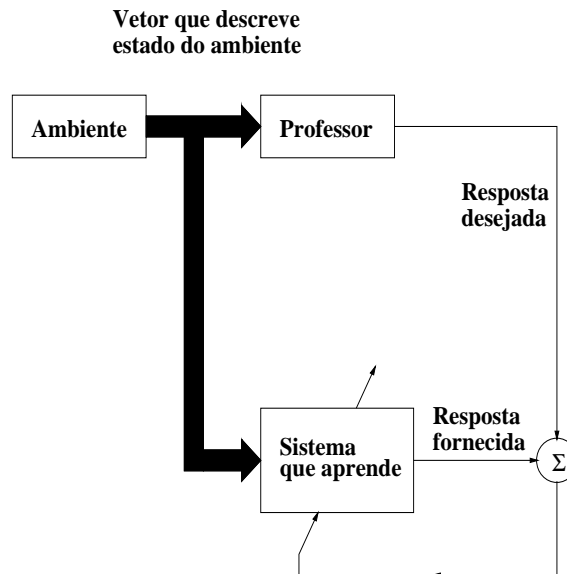


Fig. 3.3 - Diagrama de blocos de aprendizado supervisionado

está representado por um conjunto de *exemplos entrada-saida*. A rede neural, entretanto, *desconhece* o ambiente. Suponha agora que o professor e a rede neural sejam ambos expostos a um vetor (exemplo) colhido do ambiente. Por virtude de seu conhecimento, o professor é capaz de fornecer à rede uma *resposta desejada* para aquele vetor de treinamento. Os parâmetros da rede são ajustados sob a influência combinada do vetor de treinamento e do sinal de erro; o sinal de erro é definido como a diferença entre a resposta fornecida pela rede e a resposta desejada. Este ajuste é feito iterativamente passo a passo com o objetivo de eventualmente fazer a rede neural *emular* o professor; presume-se que a emulação seja ótima em algum sentido estatístico. Quando esta condição for atingida, poderemos então dispensar o professor e deixar a rede neural lidar com o ambiente daí em diante sozinha (sem supervisão).

Em função da rede, regra e paradigma de aprendizado, adotou-se o famoso algoritmo do tipo retropropagação ('backpropagation algorithm') para treinar as redes. Abaixo são descritas versões padrão e adaptativa do algoritmo [11].

A inicialização de pesos e limiares pode ser feita escolhendo-os aleatoriamente ou, por

exemplo, através do método de Nguyen-Widrow [45], que foi o caso aqui. O método lineariza a rede encontrando uma aproximação inicial por interpolação linear para a função desejada.

3.5 Algoritmos de Retropropagação

Primeiramente, o algoritmo para treinamento de retropropagação padrão é descrito [11]. Em todas as versões que se seguem y_{jn} é a entrada para o neurônio j para cada exemplo n e Δ_{in} é o gradiente do erro no neurônio i para cada exemplo n :

1. Calcule saidas da rede e e_m total;
2. Se $e_m < \epsilon$ pare;
3. Na iteração m o algoritmo de retropropagação calcula:
 - 3.1 Gradiente de erro de cada neurônio i para cada exemplo n : Δ_{in} ;
 - 3.2 Novos pesos and limiares usando, para cada neurônio i , suas entradas j e cada exemplo n
 - 3.2.1 $\Delta\theta_{ij}(m) = \eta \sum_n \Delta_{in} y_{jn}$
 - 3.2.2 $\Delta\mu_i(m) = \eta (\sum_n \Delta_{in})$
 - 3.2.3 $\theta_{ij}(m+1) = \theta_{ij}(m) + \Delta\theta_{ij}(m)$
 - 3.2.4 $\mu_i(m+1) = \mu_i(m) + \Delta\mu_i(m)$;
 - 3.3 vá para o passo 1.

3.5.1 Uma Versão Adaptativa

Um algoritmo para treinamento de retropropagação adaptativo é agora descrito [11].

1. Comece com taxa de aprendizado η_0 , constante de momentum $\alpha = 0.9$, e momentum $\alpha_0 = 0$;
2. Calcule saidas da rede e e_m total;
3. Se $e_m < \epsilon$ pare;

4. Na iteração m o algoritmo de retropropagação calcula:

4.1 O erro gradiente de cada neurônio i para cada exemplo n : Δ_{in} ;

4.2 Novos pesos e limiares tentativos usando, para cada neurônio i , suas entradas j e cada exemplo n

$$4.2.1 \quad \Delta(\theta y)_{ij} = \sum_n \Delta_{in} y_{jn}$$

$$4.2.2 \quad \Delta\theta'_{ij}(m) = \alpha_m \cdot \Delta\theta_{ij}(m-1) + (1 - \alpha_m) \cdot \eta_m \cdot \Delta(\theta y)_{ij}$$

$$4.2.3 \quad \Delta\mu'_i(m) = \alpha_m \cdot \Delta\mu_i(m-1) + (1 - \alpha_m) \cdot \eta_m \cdot (\sum_n \Delta_{in})$$

$$4.2.4 \quad \theta'_{ij}(m+1) = \theta_{ij}(m) + \Delta\theta_{ij}(m)$$

$$4.2.5 \quad \mu'_i(m+1) = \mu_i(m) + \Delta\mu_i(m);$$

5. Adapte:

5.1 Se $e_m > 1.04 \cdot e_{m-1}$ então $\eta_{m+1} = 0.7 \cdot \eta_m$ e $\alpha_{m+1} = 0$, vá para o passo 4.2;

5.2 Senão

5.2.1 se $e_m < e_{m-1}$ então $\eta_{m+1} = 1.05 \cdot \eta_m$ e $\alpha_{m+1} = \alpha$;

5.2.2 aceite pesos e limiares tentativos $\theta_{ij}(m+1) = \theta'_{ij}(m+1)$ e $\mu_i(m+1) = \mu'_i(m+1)$;

5.2.3 vá para o passo 2.

3.5.1.1 Modificação no Algoritmo

Uma modificação do Algoritmo de Retropropagação foi proposta neste trabalho [46].

No passo 4.2.3 do algoritmo 3.5.1 a fórmula para cálculo de $\Delta\mu'_i(m)$ é alterada para

$$\Delta\mu'_i(m) = \alpha_m \cdot \Delta\mu_i(m-1) + (1 - \alpha_m) \cdot \eta_m^2 \cdot \left(\sum_n \Delta_{in} \right) \quad . \quad (3.8)$$

Esta modificação tende a fazer a taxa de ajuste nos limiares mais lenta que a taxa de ajuste nos pesos θ , porque geralmente $\eta < 1$. No Capítulo de resultados numéricos (Capítulo 6) será visto que esta alteração levou a uma função de assimilação satisfatória, melhor do que o resultado obtido usando-se a fórmula original.

3.6 Aproximação de Funções

Neste trabalho o perceptron multicamada foi treinado para aproximar uma função multivariável. Aqui está um teorema que sugere como isto pode acontecer:

O *teorema de aproximação universal* ([22], página 182) afirma que

Seja φ uma função contínua não-constante, limitada e monotonicamente crescente. Seja o hipercubo unitário n -dimensional $[0, 1]^n$ denotado por I_n . O espaço de funções contínuas em I_n é denotado por $C(I_n)$. Então, dada qualquer função $f \in C(I_n)$ e $\epsilon > 0$, existe um inteiro m e conjuntos de constantes reais α_i , μ_i e θ_j^i , onde $i = 1, \dots, m$ e $j = 1, \dots, n$ tal que será possível definir

$$F(\xi_1, \dots, \xi_n) = \sum_{i=1}^m \alpha_i \varphi \left(\sum_{j=1}^n \theta_j^i \xi_j - \mu_i \right) \quad (3.9)$$

como uma aproximação da função $f(\vec{\xi})$, isto é,

$$|F(\xi_1, \dots, \xi_n) - f(\xi_1, \dots, \xi_n)| < \epsilon \quad (3.10)$$

para todos $\vec{\xi} = [\xi_1, \dots, \xi_n]^T \in I_n$.

Este teorema é diretamente aplicável a perceptrons multicamadas. Pode-se primeiramente notar que $\tanh(s)$ usada como função de ativação $\varphi(s)$ é de fato uma função não-constante, limitada, e monotonicamente crescente. Então, a Equação (3.9) pode ser encarada como a saída de um perceptron multicamada descrito como segue:

1. A rede tem n nós de entrada e uma única camada escondida consistindo de m neurônios; as entradas são denotadas por ξ_1, \dots, ξ_n ;
2. O neurônio escondido i tem pesos sinápticos $\theta_1^i, \dots, \theta_n^i$ e limiar μ_i ;
3. A saída da rede é combinação linear das saídas dos neurônios escondidos, com $\alpha_1, \dots, \alpha_m$ definindo os coeficientes dessa combinação.

O teorema declara que *uma única camada é suficiente para que um perceptron multicamada compute uma ϵ -aproximação uniforme para um dado conjunto representado*

pelo conjunto de entradas

$$\vec{\xi} = [\xi_1, \dots, \xi_n]^T$$

e uma saída

$$f(\xi_1, \dots, \xi_n) \quad .$$

Entretanto, o teorema não diz que uma camada única é ótima no sentido de tempo de aprendizado ou facilidade de implementação.

Se mais de uma camada escondida é usada, como é comum na prática, o perceptron multicamada treinado com o algoritmo de retropropagação se manifesta como um *esquema sigmoideal aninhado*, escrito na forma compacta seguinte para o caso de uma única saída ([22], p. 189):

$$F_{ANN}(\vec{\xi}, \theta, \mu) = \varphi \left(\sum_j \theta_j^o \varphi \left(\sum_k \theta_k^j \varphi \left(\sum_i \theta_i^k \xi_i - \mu^k \right) \cdots - \mu^j \right) \cdots - \mu^o \right) \quad (3.11)$$

Aqui $\varphi(s) = \tanh(s)$ foi usada como função de ativação nas camadas escondidas e a função identidade foi usada na camada de saída. Mesmo os requerimentos para $\varphi(s)$ não tendo sido obedecidos será visto que os resultados obtidos foram satisfatórios.

CAPÍTULO 4

SISTEMAS DE EVOLUÇÃO NÃO-LINEARES

Os esquemas de assimilação apresentados nos Capítulos 2 e 3 foram testados em três sistemas não-lineares dependentes do tempo: os sistemas de Lorenz [35] e Hénon [59] em regime caótico, e o modelo meteorológico DYNAMO [37].

4.1 Sistema de Hénon Caótico

Um exemplo dramático de assimilação de dados mal-sucedida pode ser dado inserindo uma perturbação no sistema caótico de tempo discreto de Hénon [59]. O sistema é:

$$X_{n+1} = 1 + Y_n - aX_n^2 \quad , \quad (4.1)$$

$$Y_{n+1} = bX_n \quad . \quad (4.2)$$

A evolução deste mapeamento adotando $a = 1.4$ e $b = 0.3$ para que o sistema esteja em estado caótico será mostrada. A unidade de tempo, mesmo sendo desnecessária, será considerada *segundos*, no sentido de que $\Delta n = (n + 1) - n = 1 \text{ s}$. As condições iniciais são

$$\begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} = \begin{bmatrix} 0.9 \\ 0.9 \end{bmatrix} \quad . \quad (4.3)$$

Duas simulações serão feitas, na segunda inserindo em $t = 50 \text{ s}$ as perturbações

$$\begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} = \begin{bmatrix} \sqrt{10^{-3}} \\ -\sqrt{10^{-3}} \end{bmatrix} \quad . \quad (4.4)$$

O resultado é ilustrado pela Figura 4.1 a seguir. A linha tracejada representa a simulação sem perturbação e a linha cheia a simulação com perturbação. A inserção

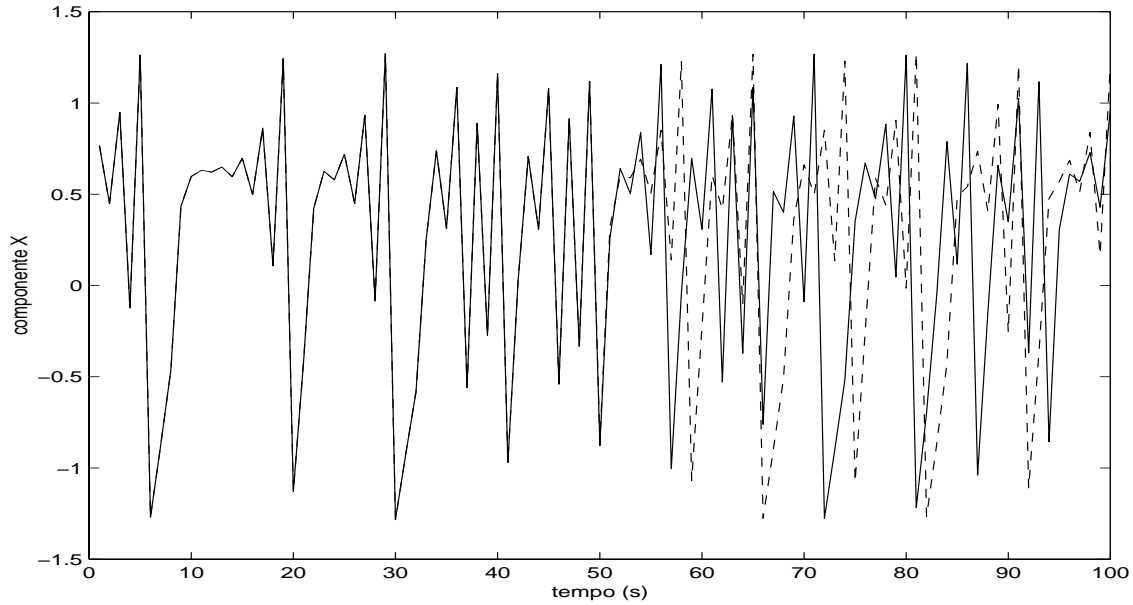


Fig. 4.1 - Corrupção temporal da variável perturbada X no sistema de Hénon.

da perturbação seriamente prejudica a previsão de X , a partir de $t = 53$ s. Aqui modificação das variáveis dependentes do tempo X e Y ou sua substituição por estimativas inapropriadas durante a simulação pode gerar soluções indesejadas.

4.2 Sistema de Lorenz Caótico

Outro sistema dinâmico não-linear caótico mais famoso é o de Lorenz. As equações do *efeito borboleta* são ([35], [36]; [42]):

$$\frac{dX}{dt} = -\sigma(X - Y) \quad (4.5)$$

$$\frac{dY}{dt} = RX - Y - XZ \quad (4.6)$$

$$\frac{dZ}{dt} = XY - bZ \quad (4.7)$$

Este sistema será integrado usando o método Euler *previsor-corretor* ('predictor-

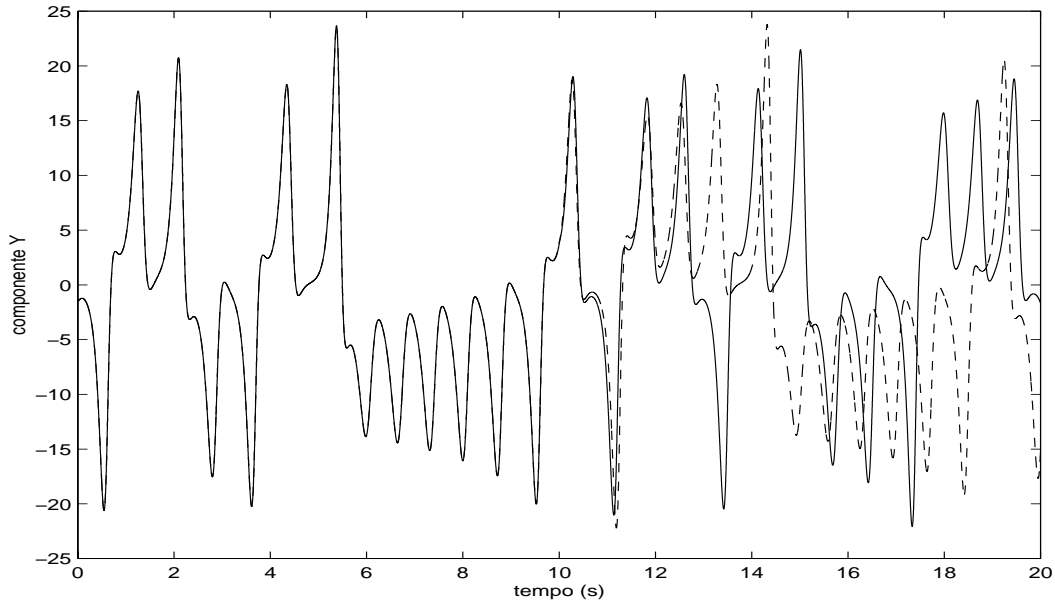


Fig. 4.2 - Corrupção temporal da variável perturbada Y no sistema de Lorenz.

corrector') adotando $\Delta t = 0.001$, $\sigma = 10$, $b = 8/3$, $R = 28$ para que o sistema esteja em estado caótico. Originalmente um método Runge-Kutta de quarta ordem era usado. Na notação adotada, o vetor de estado é

$$\vec{w} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad (4.8)$$

e as funções lineares e não-lineares são escritas como

$$\mathbf{L} = \begin{bmatrix} \sigma & -\sigma & 0 \\ -R & 1 & 0 \\ 0 & 0 & b \end{bmatrix}, \quad \vec{N}(\vec{w}) = \begin{bmatrix} 0 \\ XZ \\ XY \end{bmatrix}. \quad (4.9)$$

Usando a condição inicial

$$\vec{w}_0 = \begin{bmatrix} 1.508870 \\ -1.531271 \\ 25.46091 \end{bmatrix}. \quad (4.10)$$

Duas simulações serão feitas para ilustrar o efeito da inserção de perturbações neste sistema dinâmico. A primeira aplicando a condição inicial acima, e a segunda inserindo em $t = 7.5$ s a perturbação:

$$\Delta \vec{w} = \begin{bmatrix} 10^{-2} \\ -10^{-2} \\ 10^{-2} \end{bmatrix} . \quad (4.11)$$

O resultado é ilustrado pela Figura 4.2 apenas para a componente Y do sistema de Lorenz. A linha tracejada representa a simulação sem perturbação e a linha cheia a simulação com perturbação. A inserção das perturbações seriamente danifica a previsão de Y , de $t = 9$ s em diante. Aqui modificação do \vec{w} estado dependente do tempo ou sua substituição por estimativas apropriadas durante a integração pode gerar soluções indesejadas. Um fenômeno similar será observado na integração de um modelo meteorológico chamado DYNAMO.

4.3 O Modelo Meteorológico DYNAMO

O modelo que vai ser descrito abaixo ([7], [8]) foi concebido por P. Lynch [37], para simular movimentos atmosféricos em grande escala (a escala sinótica). Apesar de sua grande simplicidade, o código DYNAMO é capaz de reproduzir importantes fenômenos da dinâmica atmosférica, embora tenha sido imaginado para testar esquemas de inicialização.

Algumas hipóteses do modelo são:

- aproximação de água-rasa (*shallow-water equations*):
considera-se a atmosfera como um fluido incompressível e em equilíbrio hidrostático na vertical, com o campo de velocidades independente da componente z ;
- admite-se atmosfera seca;
- fronteira livre para a altura geopotencial (contorno superior);

- a superfície (fronteira no nível $z = 0$) é considerada plana;
- Assumem-se condições de contorno periódicas para todas as variáveis dependentes perturbadas.

4.3.1 Equações do Modelo

Partindo das equações de água-rasa numa camada delgada de fluido homogêneo e incompressível acima de uma superfície plana sob a ação da gravidade, num sistema de referência não inercial,

$$\frac{du}{dt} - fv + \frac{\partial \Phi}{\partial x} = 0 \quad (4.12)$$

$$\frac{dv}{dt} + fu + \frac{\partial \Phi}{\partial y} = 0 \quad (4.13)$$

$$\frac{d\Phi}{dt} + \Phi(u_x + v_y) = 0 \quad (4.14)$$

onde d/dt é a derivada substantiva; x, y são as coordenadas no sentido oeste e norte; t é o tempo; $\Phi = gh$ o geopotencial, sendo h a altura do fluido acima da superfície plana; u, v são as velocidades zonal e meridional. Considera-se o parâmetro de Coriolis na forma $f = f_0 + \beta y$, com f_0, β constantes. Por último, os subscritos significam derivação parcial.

Supondo que o vento básico (zonal) seja constante e esteja geostroficamente balanceado com o geopotencial ($f\bar{u} = -\partial\bar{\Phi}/\partial y$) e os desvios (perturbações) sejam variáveis unidimensionais

$$u = \bar{u} + U(x, t); \quad (4.15)$$

$$v = V(x, t); \quad (4.16)$$

$$\Phi = \bar{\Phi}(y) + \phi(x, t). \quad (4.17)$$

Elimina-se a dependência em y subtraindo-se a relação geostrófica da eq. (4.13), obtendo-se a equação (4.13'). As equações da vorticidade ($\zeta = v_x$) e da divergência ($\delta = u_x$) são obtidas das combinações $[(4.13')_x - (4.12)_y]$ e $[(4.12)_x + (4.13')_y]$,

respectivamente. Ou seja,

$$\zeta_t + (u\zeta)_x + \delta f + \beta V = 0 \quad (4.18)$$

$$\delta_t + (u\delta)_x - \zeta f + \beta U + \Phi_{xx} = 0 \quad (4.19)$$

$$\Phi_t + (u\Phi)_x - f\bar{u}V + \bar{\Phi} \delta = 0 \quad (4.20)$$

a equação (4.20) é obtida incorporando-se o balanço geostrófico à equação da continuidade (4.14).

O sistema (4.18)—(4.20) forma o conjunto de equações básicas do modelo, onde as variáveis prognósticas (ζ , δ , Φ) dependem somente de x e t (independentes). A dependência em y ocorre somente no parâmetro f e no geopotencial médio $\bar{\Phi}$; porém, neste modelo f e $\bar{\Phi}$ são considerados constantes onde eles não aparecem diferenciados.

O teorema de Helmholtz permite a partição do campo de vento horizontal $\vec{V}_H = U\vec{i} + V\vec{j}$ em componentes não-divergentes e irrotacionais

$$\vec{V}_H = \vec{V}_\chi + \vec{V}_\psi = \vec{k} \times \vec{\nabla}\psi + \vec{\nabla}\chi \quad ; \quad (4.21)$$

onde ψ é a função corrente e χ o potencial de velocidade. Desta maneira tem-se

$$\nabla^2\psi = \zeta \quad , \quad (4.22)$$

$$\nabla^2\chi = \delta \quad . \quad (4.23)$$

Desta forma as componentes horizontais do vento são recuperadas de

$$V = \vec{\nabla}\psi \quad , \quad (4.24)$$

$$U = \vec{\nabla}\chi \quad . \quad (4.25)$$

Então, para o presente modelo deve-se resolver, em cada passo de integração, duas equações de Poisson.

4.3.2 Adimensionalização das Equações

A conveniência de tornar as equações adimensionais, é tornar clara a importância relativa de vários termos do sistema, definindo-se escalas características para cada variável. A variável espacial é adimensionalizada por L (comprimento do canal), as velocidades por uma velocidade característica V , o tempo por f^{-1} e o geopotencial por fLV .

O uso de uma adimensionalização deste tipo nas equações do modelo, faz sobressair alguns grupos adimensionais. Definem-se então

$$R_0 \equiv \frac{V}{fL} ; \quad (4.26)$$

$$R_\beta \equiv \frac{\beta L}{f} ; \quad (4.27)$$

$$R_F \equiv \frac{\bar{\Phi}}{(fL)^2} = \left(\frac{L_R}{L}\right)^2 , \quad (4.28)$$

onde R_0 é chamado *número de Rossby*.

Deste modo as equações (4.18)—(4.20) podem ser reescritas na forma adimensional

$$\zeta_t + R_0 (u\zeta)_x + \delta + R_\beta V = 0 \quad (4.29)$$

$$\delta_t + R_0 (u\delta)_x - \zeta + R_\beta U + \phi_{xx} = 0 \quad (4.30)$$

$$\phi_t + R_0 (u\Phi)_x - R_0 u_0 V + R_F \delta = 0 . \quad (4.31)$$

Note-se que u_0 é o vento zonal médio (\bar{u}) adimensionalizado.

O sistema (4.29)—(4.31) pode ser escrito numa forma matricial prognóstica

$$\frac{\partial \vec{W}}{\partial t} + \mathbf{A} \vec{W} + \mathbf{B} \vec{X} = -R_0 \vec{N}_{\vec{W}}(\vec{W}, \vec{X}) \quad (4.32)$$

onde os vetores de estado são

$$\vec{W} = \left[\zeta \quad \delta \quad \phi \right]^T , \quad (4.33)$$

$$\vec{X} = \left[V \quad U \quad \phi \right]^T , \quad (4.34)$$

os coeficientes matriciais e o termo não linear são dados abaixo

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & \frac{\partial^2}{\partial x^2} \\ 0 & R_F & 0 \end{bmatrix} , \quad (4.35)$$

$$\mathbf{B} = \begin{bmatrix} R_\beta & 0 & 0 \\ 0 & R_\beta & 0 \\ -R_0 u_0 & 0 & 0 \end{bmatrix} , \quad (4.36)$$

$$\vec{N}_W(\vec{W}, \vec{X}) = [\vec{N}_W^\zeta \quad \vec{N}_W^\delta \quad \vec{N}_W^\phi]^T = \frac{\partial(\vec{X}_2 \vec{W})}{\partial x} . \quad (4.37)$$

A relação entre o vetor das variáveis prognósticas (\vec{W}) e o vetor das variáveis primitivas (\vec{X}) é dada por

$$\vec{W} = \mathbf{M} \vec{X} , \quad (4.38)$$

no qual \mathbf{M} é um operador diferencial linear de primeira ordem:

$$\mathbf{M} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial x} & 0 \\ 0 & 0 & 1 \end{bmatrix} . \quad (4.39)$$

Aplicando a equação (4.38) na equação (4.32), obtém-se a formulação primitiva

$$\mathbf{M} \frac{\partial \vec{X}}{\partial t} + (\mathbf{A}\mathbf{M} + \mathbf{B})\vec{X} = -R_0 \vec{N}(\vec{X}) , \quad (4.40)$$

e o termo não linear associado é

$$\vec{N}(\vec{X}) = \frac{\partial(\vec{X}_2 \mathbf{M} \vec{X})}{\partial x} . \quad (4.41)$$

Doravante denotar-se-á a matriz $(\mathbf{A}\mathbf{M} + \mathbf{B})$ pela matriz \mathbf{H} .

4.3.3 Discretização Espacial

Para integrar o sistema (4.32) ou (4.40) usou-se uma aproximação por diferenças finitas, transformando a equação num sistema de equações diferenciais ordinárias.

Como as velocidades (V, U) e as variáveis prognósticas (ζ, δ) estão relacionadas por um operador diferencial de primeira ordem, é adotada uma grade alternada [10]. As velocidades são especificadas para "semi-pontos" e as variáveis prognósticas e o geopotencial para "pontos inteiros".

Os operadores de discretização são dados por

$$(q_m)_x = \frac{1}{\Delta x}(q_{m+1/2} - q_{m-1/2}) \quad (4.42)$$

$$(\bar{q}_m) = \frac{1}{2}(q_{m+1/2} + q_{m-1/2}) \quad (4.43)$$

$$(\bar{q}_m)_x = \frac{1}{\Delta x}(q_{m+1} - q_{m-1}) \quad (4.44)$$

$$(q_m)_{xx} = \frac{1}{\Delta x^2}(q_{m+1} - 2q_m + q_{m-1}) \quad (4.45)$$

$$(Uq)_x|_m = \frac{1}{2\Delta x}[U_{m+1/2}(q_{m+1} - q_m) - U_{m-1/2}(q_m - q_{m-1})] \quad (4.46)$$

Usando estes operadores na formulação prognóstica e primitiva e supondo condições de contorno periódicas para todas as variáveis dependentes perturbadas ($Q(x+L) = Q(x)$), obtém-se

$$\frac{d\vec{W}}{dt} + \mathbf{A}\vec{W} = -[1/2 \mathbf{B} \vec{X} + R_0 \vec{N}_W(\vec{W}, \vec{X})] = -R_0 \vec{N}_B(\vec{W}, \vec{X}) \quad (4.47)$$

$$\mathbf{M} \frac{d\vec{X}}{dt} + (\mathbf{A}\mathbf{M} + 1/2 \mathbf{B}) \vec{X} = -R_0 \vec{N}(\vec{X}) \quad (4.48)$$

onde os vetores de estado e os termos não lineares tornam-se

$$\vec{W} = [\zeta_1 \ \delta_1 \ \phi_1 \ \dots \ \zeta_{N_x} \ \delta_{N_x} \ \phi_{N_x}]^T, \quad (4.49)$$

$$\vec{X} = [V_{-1/2} \ U_{-1/2} \ \phi_1 \ \dots \ V_{N_x-1/2} \ U_{N_x-1/2} \ \phi_{N_x}]^T \quad , \quad (4.50)$$

$$\vec{N}_{\vec{W}}(\vec{W}, \vec{X}) = -\frac{1}{2\Delta x} \mathbf{C} \vec{D}(\vec{X}) \mathbf{R} \vec{W} \quad , \quad (4.51)$$

$$\vec{N}(\vec{X}) = -\frac{1}{2\Delta x} \mathbf{C} \vec{D}(\vec{X}) \mathbf{R} \mathbf{M} \vec{X} \quad , \quad (4.52)$$

e os coeficientes matriciais de (4.47) e (4.48) são dados na forma

$$\mathbf{A} = \begin{bmatrix} \mathbf{E} & \mathbf{F} & & & & & & & \mathbf{F} \\ & \mathbf{F} & \mathbf{E} & \mathbf{F} & & & & & \\ & & \mathbf{F} & \mathbf{E} & \mathbf{F} & & & & \\ & & \vdots & \vdots & \vdots & & & & \\ & & & & \mathbf{F} & \mathbf{E} & \mathbf{F} & & \\ \mathbf{F} & & & & \mathbf{F} & \mathbf{E} & & & \end{bmatrix} , \quad \mathbf{B} = \begin{bmatrix} \mathbf{G} & \mathbf{G} & & & & & & & \\ & \mathbf{G} & \mathbf{G} & & & & & & \\ & & \mathbf{G} & \mathbf{G} & & & & & \\ & & \vdots & \vdots & & & & & \\ & & & & \mathbf{G} & \mathbf{G} & & & \\ \mathbf{G} & & & & & & \mathbf{G} & & \end{bmatrix} . \quad (4.53)$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{M}_2 & & & & & & & \\ & \mathbf{M}_1 & \mathbf{M}_2 & & & & & & \\ & & \mathbf{M}_1 & \mathbf{M}_2 & & & & & \\ & & \vdots & \vdots & & & & & \\ & & & & \mathbf{M}_1 & \mathbf{M}_2 & & & \\ \mathbf{M}_2 & & & & & & \mathbf{M}_1 & & \end{bmatrix} , \quad \mathbf{C} = \begin{bmatrix} -\mathbf{I} & \mathbf{I} & & & & & & & \\ & -\mathbf{I} & \mathbf{I} & & & & & & \\ & & -\mathbf{I} & \mathbf{I} & & & & & \\ & & \vdots & \vdots & & & & & \\ & & & & -\mathbf{I} & \mathbf{I} & & & \\ \mathbf{I} & & & & & & -\mathbf{I} & & \end{bmatrix} . \quad (4.54)$$

$$\mathbf{R} = \begin{bmatrix} \mathbf{I} & & & & & & & & \mathbf{I} \\ \mathbf{I} & \mathbf{I} & & & & & & & \\ & \mathbf{I} & \mathbf{I} & & & & & & \\ & \vdots & \vdots & & & & & & \\ & & & \mathbf{I} & \mathbf{I} & & & & \\ & & & & \mathbf{I} & \mathbf{I} & & & \end{bmatrix} , \quad D(\vec{X}) = \text{diag} [U_{-1/2} \mathbf{I}, U_{1/2} \mathbf{I}, \dots, U_{N_x-1/2} \mathbf{I}] . \quad (4.55)$$

Os elementos (submatrizes) das matrizes acima são expressos por

$$\mathbf{E} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -\frac{2}{\Delta x^2} \\ 0 & R_F & 0 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\Delta x^2} \\ 0 & 0 & 0 \end{bmatrix}, \quad (4.56)$$

$$\mathbf{M}_1 = \begin{bmatrix} -\frac{1}{\Delta x} & 0 & 0 \\ 0 & -\frac{1}{\Delta x} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{M}_2 = \begin{bmatrix} \frac{1}{\Delta x} & 0 & 0 \\ 0 & \frac{1}{\Delta x} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (4.57)$$

$$\mathbf{G} = \begin{bmatrix} R_\beta & 0 & 0 \\ 0 & R_\beta & 0 \\ -R_0 u_0 & 0 & 0 \end{bmatrix}, \quad \mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.58)$$

Note que apesar de as variáveis no lado direito da equação (4.47) (a parte forçante) serem \vec{W} e \vec{X} , na realidade elas dependem uma da outra, conforme (4.38). O código DYNAMO calcula $\vec{W}(t)$ por integração no tempo e a variável auxiliar $\vec{X}(t)$ resolvendo a equação $\vec{W}(t) = \mathbf{M}\vec{X}(t)$. A comunicação entre os Filtros de Kalman e a subrotina de integração é exclusivamente através do vetor $\vec{W}_n = \vec{W}(t_n)$. O filtro calcula \vec{X}_n sozinho usando (4.24-4.25).

Para recuperar os campos de vento deve-se primeiro resolver duas equações do tipo (4.23). Apresenta-se aqui somente a solução para a função corrente, visto que o caso do potencial de velocidade é análogo. Para o modelo DYNAMO a equação de Poisson discretizada é como segue

$$A_\Delta \Psi = \Xi; \quad (4.59)$$

onde $\Psi = [\psi_1, \psi_2, \dots, \psi_{N_x}]^T$, $\Xi = \Delta x^2 [\zeta_1, \zeta_2, \dots, \zeta_{N_x}]^T$ e a matriz A_Δ é dada por

$$A_\Delta = \begin{bmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & \vdots & \vdots & \vdots & \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{bmatrix}. \quad (4.60)$$

Como o problema para o potencial de velocidade é similar, a matriz do sistema é a mesma como dada em (4.59).

No próximo Capítulo serão mostrados os resultados obtidos para os três sistemas dinâmicos não-lineares na assimilação usando os métodos apresentados no Capítulo 2.

CAPÍTULO 5

RESULTADOS NUMÉRICOS DO FILTRO DE KALMAN ESTENDIDO ADAPTATIVO

Tendo apresentado no Capítulo 2 uma nova abordagem para a Assimilação de Dados, um Filtro de Kalman Adaptativo alternativo, é agora necessário testá-lo e comparar sua performance com os outros métodos típicos mencionados no Capítulo 2. O primeiro e segundo testes foram feitos usando respectivamente o sistema de Lorenz (seção 4.2) e de Hénon (seção 4.1). O terceiro teste usou o modelo meteorológico DYNAMO, apresentado na seção 4.3.

5.1 Resultados Obtidos com o Sistema de Hénon Caótico

O primeiro experimento numérico consistiu em usar o Filtro Adaptativo (subseção 2.3.1) para assimilação no sistema de Hénon da seção 4.1. O filtro foi implementado usando-se:

$$\vec{w}_{n+1} = \vec{f}(\vec{w}_n) \quad , \quad (5.1)$$

$$\mathbf{F}_n \doteq \left. \frac{\partial \vec{f}}{\partial \vec{w}_n} \right|_{\vec{w}_n = \vec{w}_n^g} \quad , \quad (5.2)$$

$$\mathbf{H}_{n+1} \doteq \mathbf{I} \quad , \quad (5.3)$$

$$\mathbf{Q}_0 = \mathbf{0} \quad , \quad (5.4)$$

$$\mathbf{R}_n \doteq \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix} \quad , \quad (5.5)$$

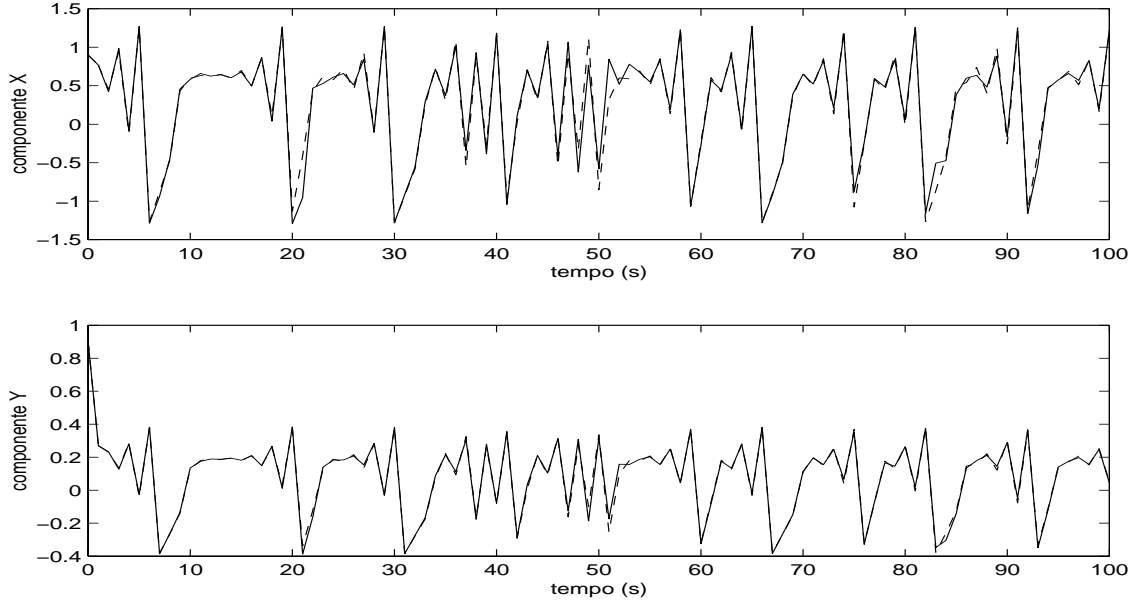


Fig. 5.1 - Assimilação de dados no sistema de Hénon usando Filtro de Kalman Adaptativo.

$$\mathbf{P}_0^a \doteq 10 \times \begin{bmatrix} X_0^2 & 0 \\ 0 & Y_0^2 \end{bmatrix} , \quad (5.6)$$

$$\mathbf{P}_0^{qa} = 10 \times \mathbf{R}_0^q , \quad (5.7)$$

$$\gamma = 1 , \quad (5.8)$$

$$\alpha = 3 . \quad (5.9)$$

$\vec{f}(\vec{w})$ descreve a dinâmica do sistema, \mathbf{H}_{n+1} é a matriz de observação, \mathbf{Q}_0 a covariância inicial do erro de modelagem, \mathbf{R}_n a covariância do erro de observação e \mathbf{P}_0^a a covariância inicial do erro de assimilação. A escolha 5.8 foi adotada para supor \mathbf{Q}_n constante, e \mathbf{P}_0^{qa} e α foram ajustados para obter-se resultado satisfatório. O resultado da estimação de estado inserindo dados a cada 2 passos de tempo é mostrado na Figura 5.1. A evolução dos erros é mostrada na Figura seguinte 5.2. A

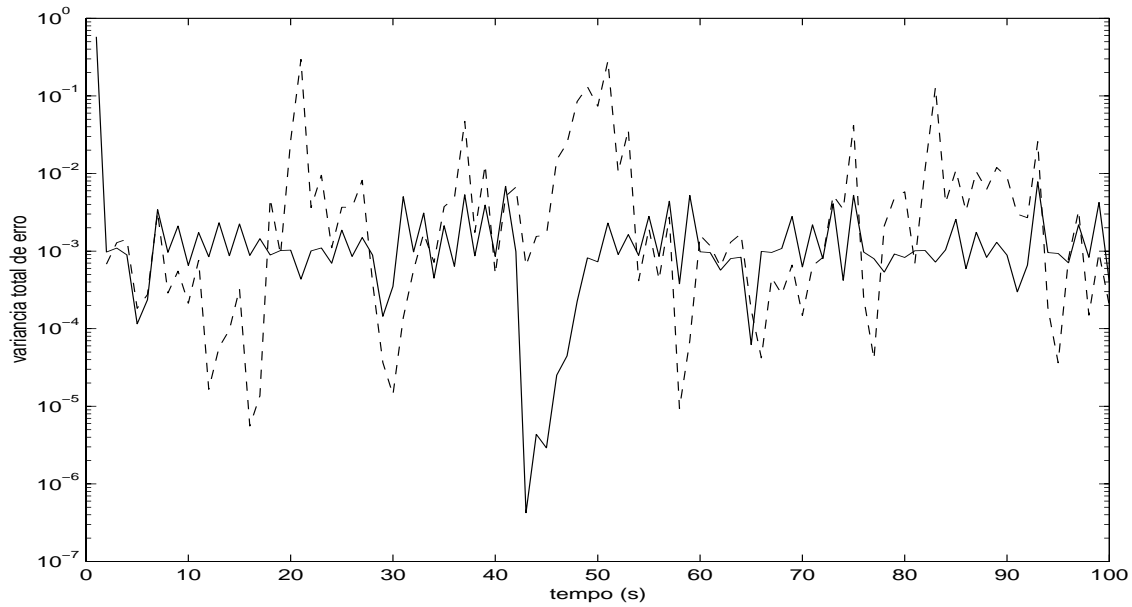


Fig. 5.2 - Variância total de erro na assimilação usando Filtro de Kalman Adaptativo.

linha cheia representa o erro previsto e a tracejada o erro verdadeiro. Embora haja discrepâncias sensíveis entre o erro previsto e o erro verdadeiro o valor médio dos dois parece ser o mesmo, da ordem de 10^{-3} .

5.2 Resultados Obtidos com o Sistema de Lorenz

O segundo experimento numérico consistiu em usar a Transformada de Laplace (seção 2.2) e os três tipos de Filtros de Kalman descritos na seção 2.3, o Linear, o Estendido e o Adaptativo (subseção 2.3.1) para realizar assimilação de dados no sistema de Lorenz.

Tentando usar a T. Laplace encontra-se o problema de que o sistema de Lorenz (4.5-4.6-4.7) é não-linear. Apenas o melhor resultado é mostrado, em que a fórmula (2.12) foi empregada usando

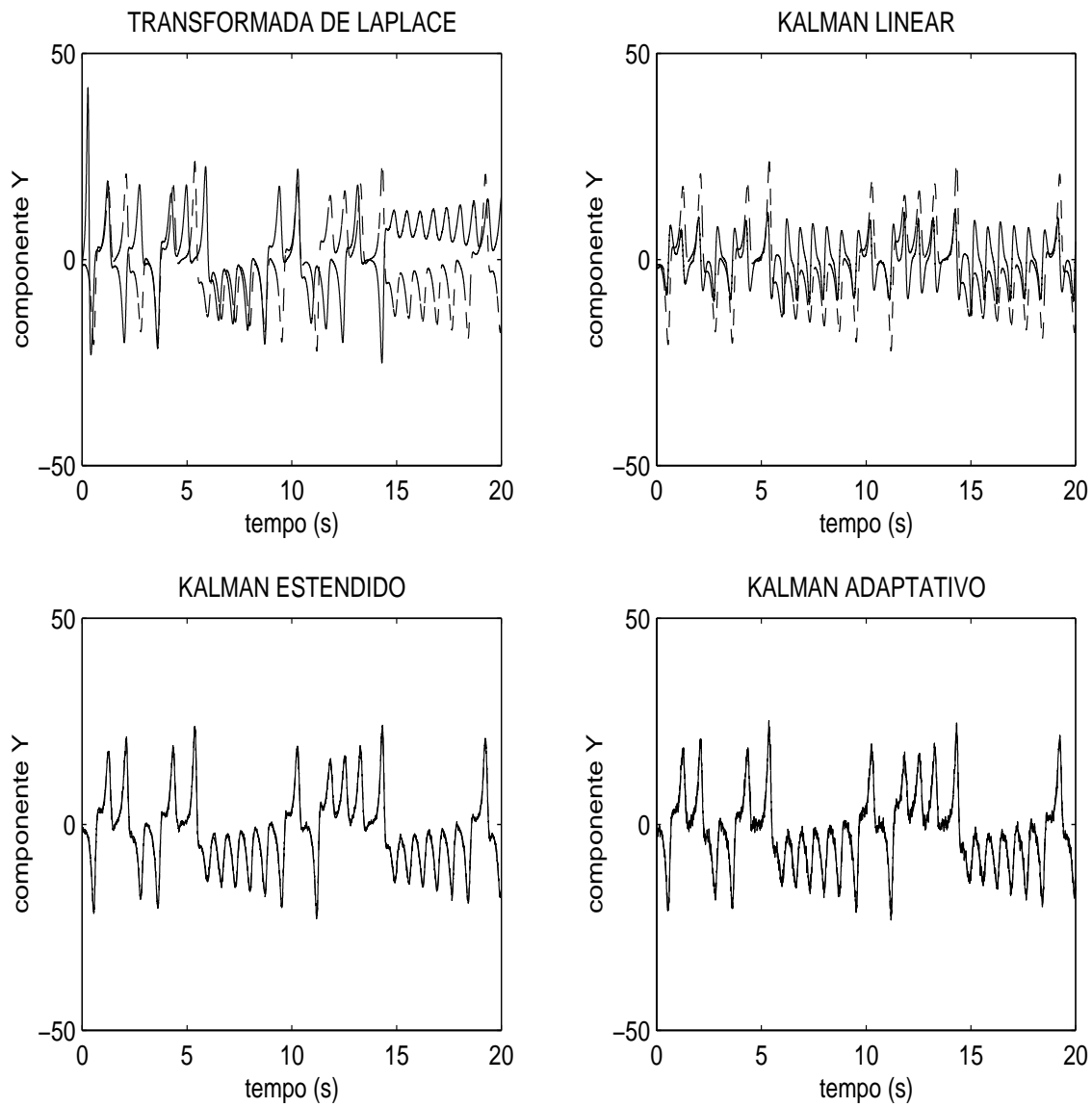


Fig. 5.3 - Comparação entre assimilações no sistema de Lorenz usando Transformada de Laplace (recursão de 1ª ordem), Filtro de Kalman Linear, Filtro de Kalman Estendido e Filtro de Kalman Adaptativo.

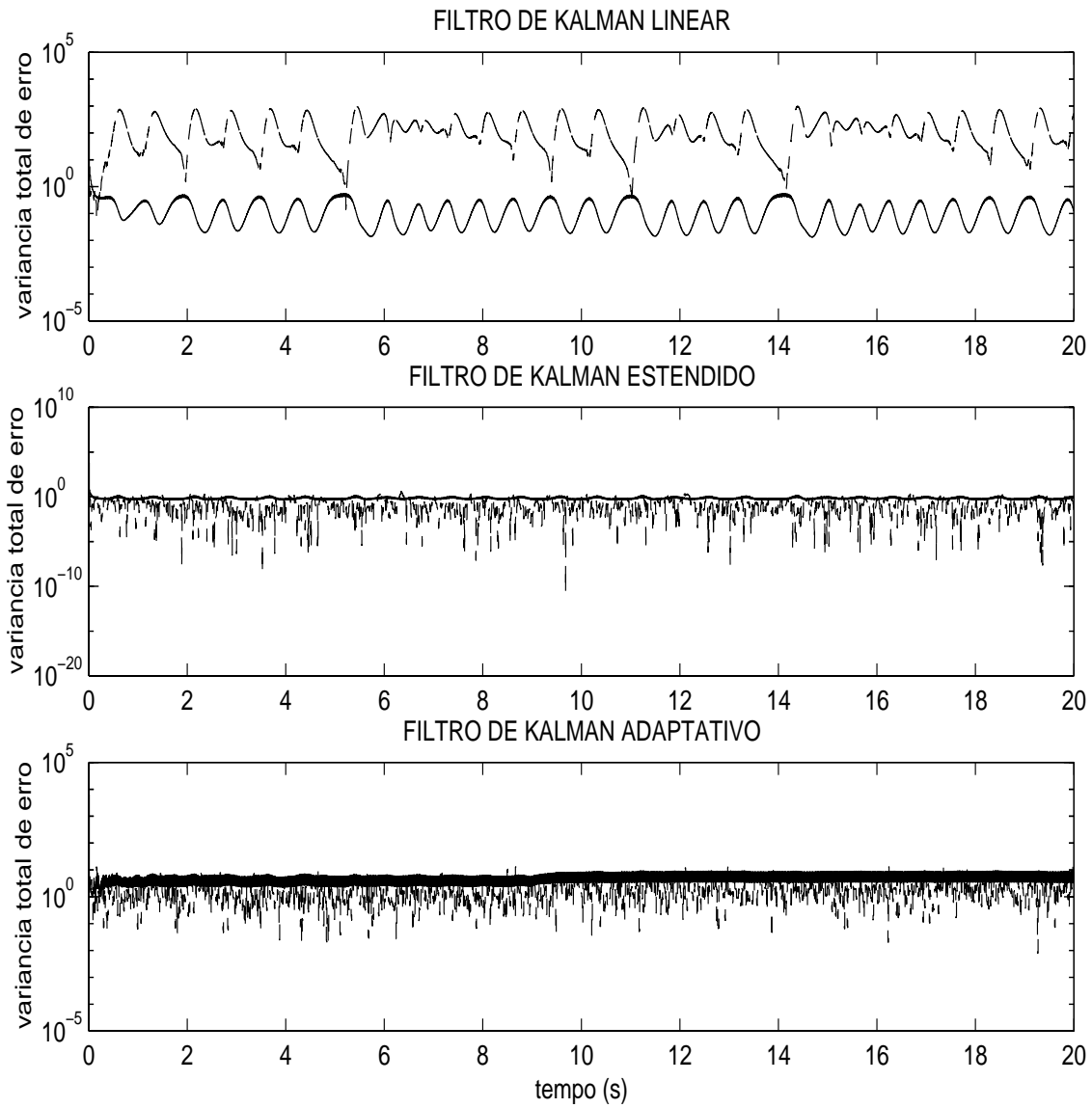


Fig. 5.4 - Variâncias totais de erro na assimilação usando Filtros de Kalman Linear, Estendido e Adaptativo.

$$\vec{w} \doteq \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} , \quad (5.10)$$

$$\mathbf{L} \doteq \begin{bmatrix} \sigma & -\sigma & 0 \\ -R & 1 & 0 \\ 0 & 0 & b \end{bmatrix} , \quad (5.11)$$

$$\vec{N}(X + \nu, Y - \nu, Z + \nu, t + \Delta t) \doteq \begin{bmatrix} 0 \\ (X + \nu)(Z + \nu) \\ -(X + \nu)(Y - \nu) \end{bmatrix} , \quad (5.12)$$

$$var\{\nu\} = 2 . \quad (5.13)$$

Os autovalores da parte linear deste sistema são $\{-22.83, 11.83, -2.67\}$, portanto esta parte não só é instável mas também "rígido" (*stiff*). Pode-se concluir que a parte não-linear do sistema estabiliza a solução completa. A solução da integração usando T. Laplace rapidamente se desvia da solução verdadeira para $t \geq 0.02$ s. Na Figura 5.3 a seguir pode ser visto que este método para assimilação de dados falha completamente para $t \geq 0.02$ s.

Escrevendo o sistema de Lorenz como

$$\frac{d\vec{w}}{dt} = \vec{g}(\vec{w}) , \quad (5.14)$$

no filtro de Kalman Linear é adotado o modelo

$$\frac{d\vec{w}}{dt} \simeq \frac{\partial \vec{g}}{\partial \vec{w}} \vec{w} \quad (5.15)$$

e usado o método de integração Euler explícito para obter \mathbf{F}_n

$$\vec{w}_{n+1} \simeq \vec{w}_n + \Delta t \left. \frac{\partial \vec{g}}{\partial \vec{w}} \right|_{\vec{w}_n = \vec{w}_n^a} \vec{w}_n = \mathbf{F}_n \vec{w}_n . \quad (5.16)$$

para calcular $\vec{w}_{n+1} = f(\vec{w}_n)$ o modelo de previsão linearizado (5.15) foi integrado usando o método Euler previsor-corretor.

O filtro foi implementado usando-se:

$$\vec{w}_{n+1} = \vec{f}(\vec{w}_n) \quad (5.17)$$

$$\mathbf{F}_n \doteq \left[\mathbf{I} + \Delta t \left. \frac{\partial \vec{g}}{\partial \vec{w}} \right|_{\vec{w}_n = \vec{w}_n^a} \right] \quad (5.18)$$

$$\mathbf{H}_{n+1} \doteq \mathbf{I} \quad (5.19)$$

$$\mathbf{Q}_n \doteq \mathbf{0} \quad (5.20)$$

$$\mathbf{R}_n \doteq \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad (5.21)$$

$$\mathbf{P}_0^a \doteq 10 \times \begin{bmatrix} X_0^2 & 0 & 0 \\ 0 & Y_0^2 & 0 \\ 0 & 0 & Z_0^2 \end{bmatrix} . \quad (5.22)$$

Note-se que (5.20) significa que a modelagem é considerada perfeita, isto é, apenas na observação \vec{z}_n há erro.

Para usar o F. Kalman Estendido e o F. Kalman Adaptativo $\vec{f}(\vec{w}_n)$ foi obtida diretamente da Equação (5.14) usando-se o método de Euler explícito:

$$\vec{w}_{n+1} = \vec{w}_n + \Delta t \cdot \vec{g}(\vec{w}_n) = \vec{f}(\vec{w}_n) \quad (5.23)$$

e, portanto, foi adotada, no lugar da expressão (5.18),

$$\mathbf{F}_n \doteq \left. \frac{\partial \vec{f}}{\partial \vec{w}_n} \right|_{\vec{w}_n = \vec{w}_n^a} , \quad (5.24)$$

Para calcular $\vec{w}_{n+1} = \vec{f}(\vec{w}_n)$ o modelo de previsão (5.14) foi integrado usando-se Euler previsor-corretor.

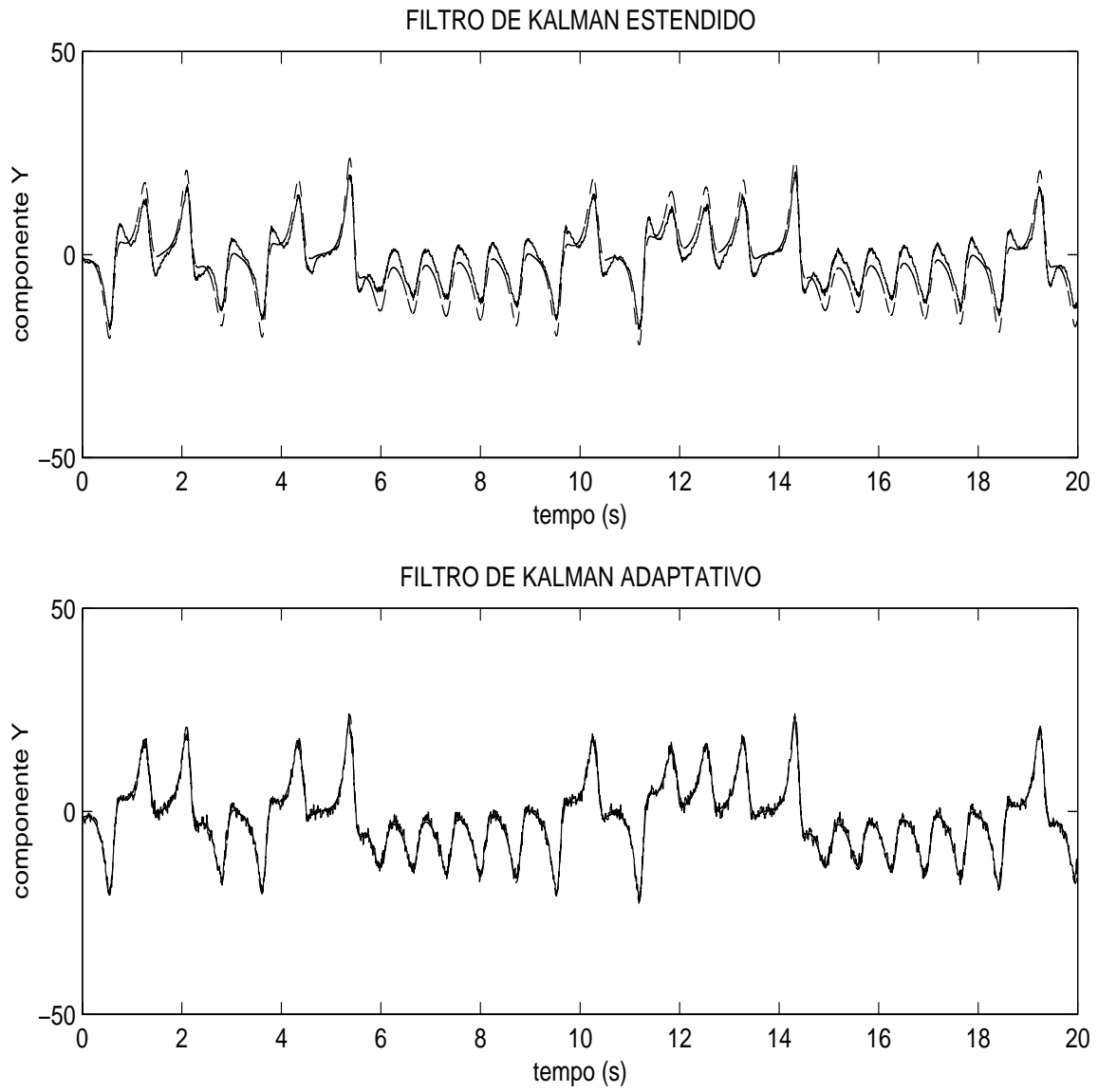


Fig. 5.5 - Comparação entre assimilações usando Filtros de Kalman Estendido e Adaptativo com modelo *incorreto*.

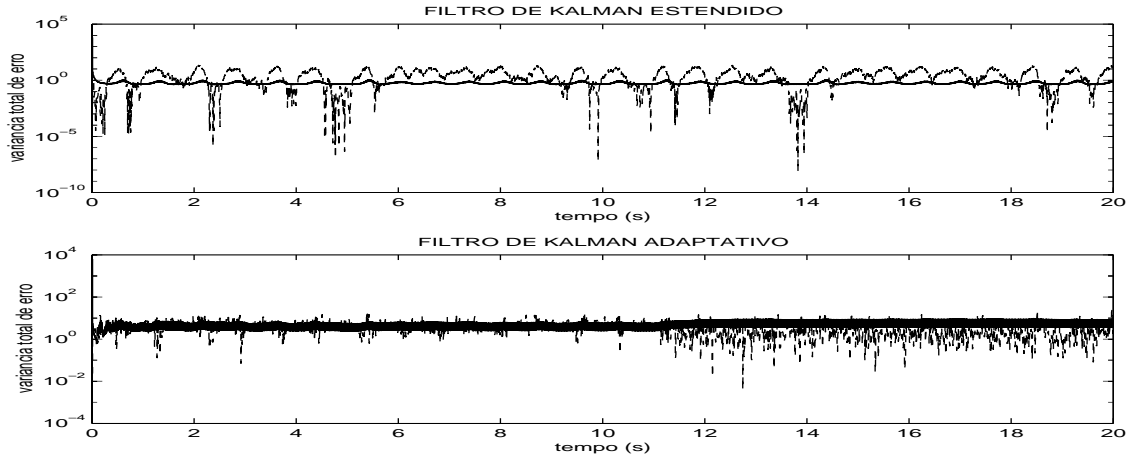


Fig. 5.6 - Variâncias totais de erro na assimilação usando Filtros de Kalman Estendido e Adaptativo com modelo *incorreto*.

No F. de Kalman Adaptativo foram adotados

$$\mathbf{P}_0^{qa} = 10 \times \mathbf{R}_0^q \quad (5.25)$$

$$\gamma = 1 \quad . \quad (5.26)$$

A escolha (5.26) supõe \mathbf{Q}_n constante. Já que todos os parâmetros do problema estocástico são estacionários pareceu desnecessário adotar erro não-estacionário. A escolha (5.25) faz \mathbf{Q}_n ser rapidamente estimada pelo algoritmo. Note-se que quando $|\mathbf{P}_0^{qa}| \gg |\mathbf{R}_0^q|$, \mathbf{G}_{n+1}^q aproxima-se da matriz identidade no passo 2 da Figura 2.5 e portanto \mathbf{P}_n^{qa} aproxima-se da matriz nula mais rapidamente conforme o passo 4. Finalmente, neste filtro foi determinado experimentalmente que resultado satisfatório é obtido adotando-se

$$\alpha = 3 \quad . \quad (5.27)$$

Na primeira parte deste experimento os desempenhos dos três Filtros de Kalman foram comparadas. Para este objetivo observações com ruído de variância 2 foram

inseridas a cada 0.012 s, integrando até $t = 20$ s. Foram obtidos os gráficos das Figuras 5.3 e 5.4. Na primeira Figura as linhas tracejadas representam o sinal verdadeiro, as linhas cheias o sinal previsto. Nas outras duas as linhas cheias representam o erro previsto pelo filtro e as linhas tracejadas os erros verdadeiros.

À primeira vista o desempenho do Filtro de Kalman Estendido foi o melhor. Na Figura 5.4 pode ser visto que os erros previstos e verdadeiros foram (em média) da mesma ordem de grandeza. Apesar de prever corretamente o erro de estimação, o Filtro Adaptativo estimou um pouco pior. O Filtro Linear teve dificuldade em rastrear o sinal original porque o modelo do sinal que usou foi muito inadequado, como pode ser visto na mesma Figura, onde o erro previsto foi muito inferior ao erro verdadeiro.

Até este momento nenhum ganho viera da abordagem adaptativa. Entretanto, ao se testar os Filtros Estendido e Adaptativo mudando um pouco os parâmetros do modelo de Lorenz, usando por exemplo

$$\sigma' = \sigma \times 0.8 \tag{5.28}$$

$$R' = R \times 0.8 \tag{5.29}$$

$$b' = b \times 0.8 \quad , \tag{5.30}$$

no lugar de σ , R e b , foram obtidas as Figuras 5.5 e 5.6.

A estimativa feita pelo Filtro de Kalman Estendido degradou muito mais que a feita pelo Filtro Adaptativo. Na assimilação com o modelo de Lorenz caótico tal robustez não é importante, já que o modelo é conhecido exatamente. Mas na assimilação de dados atmosféricos real, onde a modelagem da atmosfera é imprecisa, tal robustez é importante.

5.3 Resultados Obtidos com o Modelo DYNAMO

O terceiro experimento numérico foi com o modelo meteorológico DYNAMO. Observações da atmosfera real foram simuladas por um vetor

$$\vec{Z}_n = \vec{W}_n + \vec{v}_n \quad (5.31)$$

onde \vec{v} teve como componentes ruídos brancos gaussianos de média nula artificialmente gerados e cuja amplitude era pequena (0.1%) comparada com a magnitude de \vec{W} , representando as perturbações no processo de assimilação de dados. As perturbações foram geradas como variações de U , V e ϕ transformadas em variações de ζ , δ and ϕ através de $\Delta\vec{W} = \mathbf{M} \Delta\vec{X}$ (equation 4.38).

O perfil espacial da perturbação em cada componente do vetor é dado por

$$\Delta\phi(x) = (1 - \cos(x')) \quad , \quad (5.32)$$

$$\Delta V(x) = \frac{d\Delta\phi(x')}{dx} \quad , \quad (5.33)$$

$$\Delta U(x) = -\Delta V(x') \quad , \quad (5.34)$$

onde

$$x' = \begin{cases} 2\pi \frac{x}{L} - \frac{L}{2}, & \text{se } x \in \left[\frac{L}{2} - \frac{L}{12}, \frac{L}{2} + \frac{L}{12} \right] \\ 0, & \text{se } x \notin \left[\frac{L}{2} - \frac{L}{12}, \frac{L}{2} + \frac{L}{12} \right] \end{cases} \quad , \quad (5.35)$$

$$L = 10000 \text{ km} \quad . \quad (5.36)$$

Estes perfis são mostrados na Figura 5.7.

O resultado obtido inserindo estas perturbações *sem filtragem* numa simulação da atmosfera usando DYNAMO *sem inicialização* a cada $\Delta t = 11.1 \text{ h}$ é ilustrado através da evolução de U no ponto médio da grade na Figura a seguir 5.8. Nesta Figura a linha tracejada representa a atmosfera verdadeira e a linha sólida o resultado de simulação com perturbação.

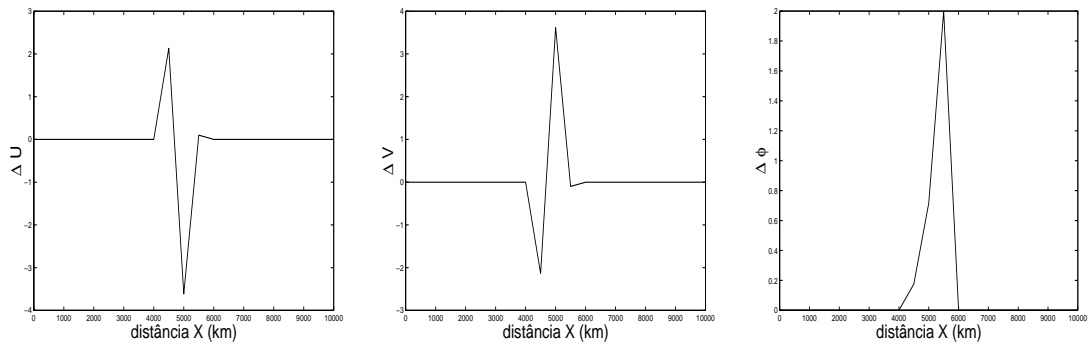


Fig. 5.7 - Perfis das perturbações ΔU , ΔV e $\Delta \phi$

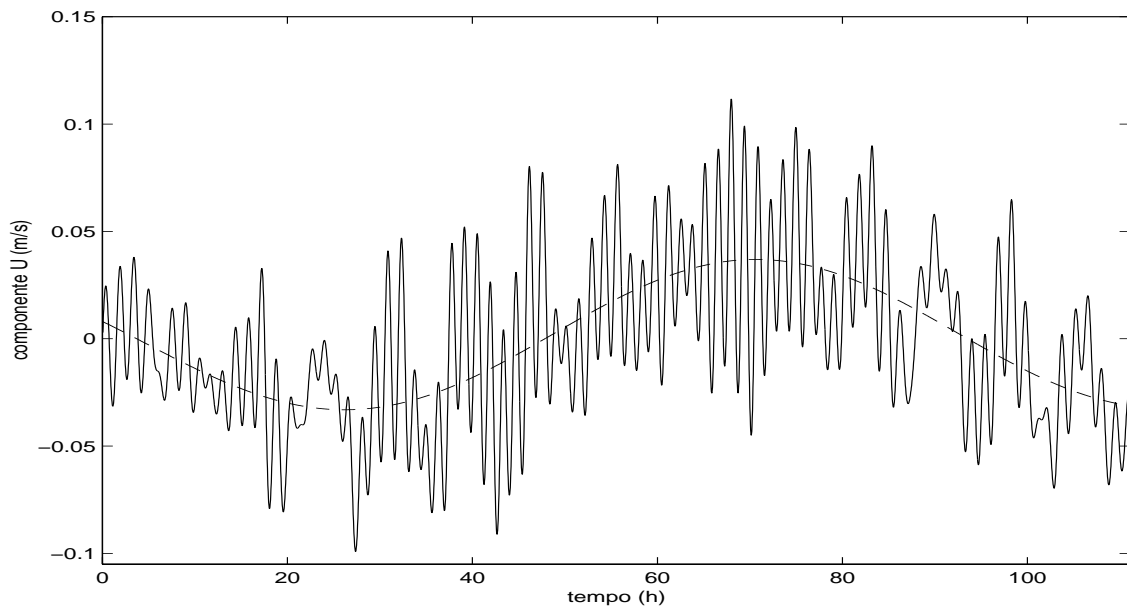


Fig. 5.8 - Corrupção Temporal de U na simulação perturbada não-inicializada.

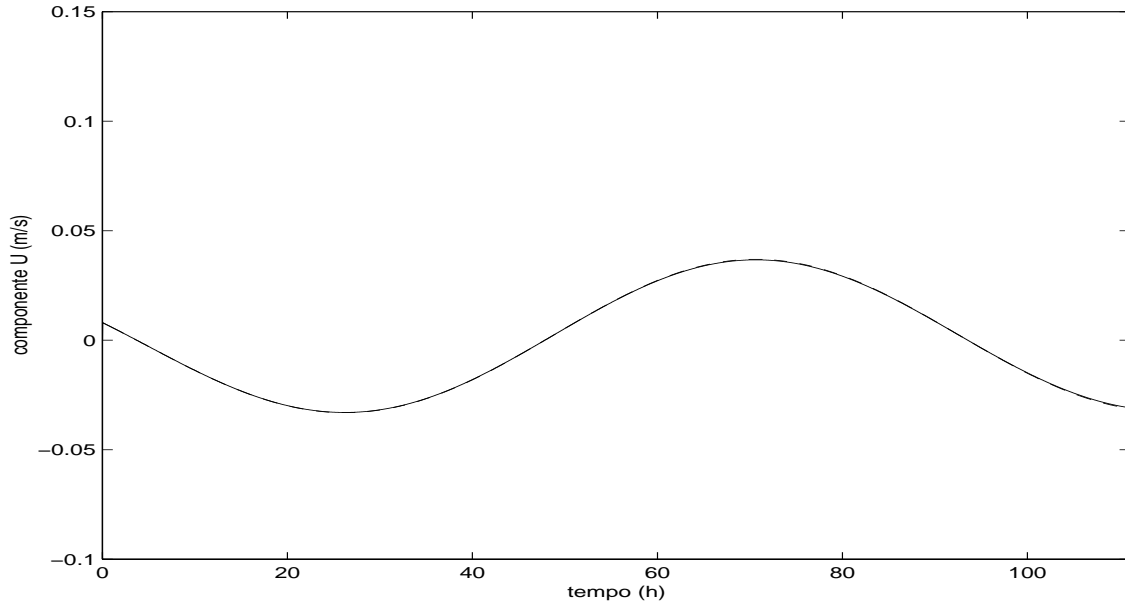


Fig. 5.9 - Evolução Temporal de U na assimilação usando Transformada de Laplace.

O segundo passo foi estudar o resultado obtido usando Transformada de Laplace (recursão de segunda-ordem - veja seção 2.2, Equação 2.13) para assimilar os dados de entrada \vec{Z}_n . A evolução de U no ponto médio da grade pode ser vista na Figura seguinte 5.9. A Figura a seguir 5.10 mostra a diferença entre U obtida através de T. Laplace usando recursão de 1^a e 2^a ordem (veja 2.2, Equações 2.12 e 2.13).

O terceiro passo foi testar o Filtro de Kalman Linear, usando as mesmas observações \vec{Z}_n . No filtro, o modelo DYNAMO foi usado com $\vec{N}_{\vec{W}}(\vec{W}, \vec{X}) = \vec{0}$. O estado verdadeiro da atmosfera foi gerado usando o modelo completo DYNAMO (com $\vec{N}_{\vec{W}}(\vec{W}, \vec{X}) \neq \vec{0}$), que representa o Sistema Dinâmico da Atmosfera. Foi suposto que o erro de previsão aumente linearmente ao longo do tempo ([43],[2]), e portanto foi adotado

$$\mathbf{F}_n = \mathbf{I} \quad . \quad (5.37)$$

A matriz de covariância de ruído de modelagem é uma matriz diagonal gerada aleatoriamente supondo que o erro de modelagem em cada componente de \vec{W} tem

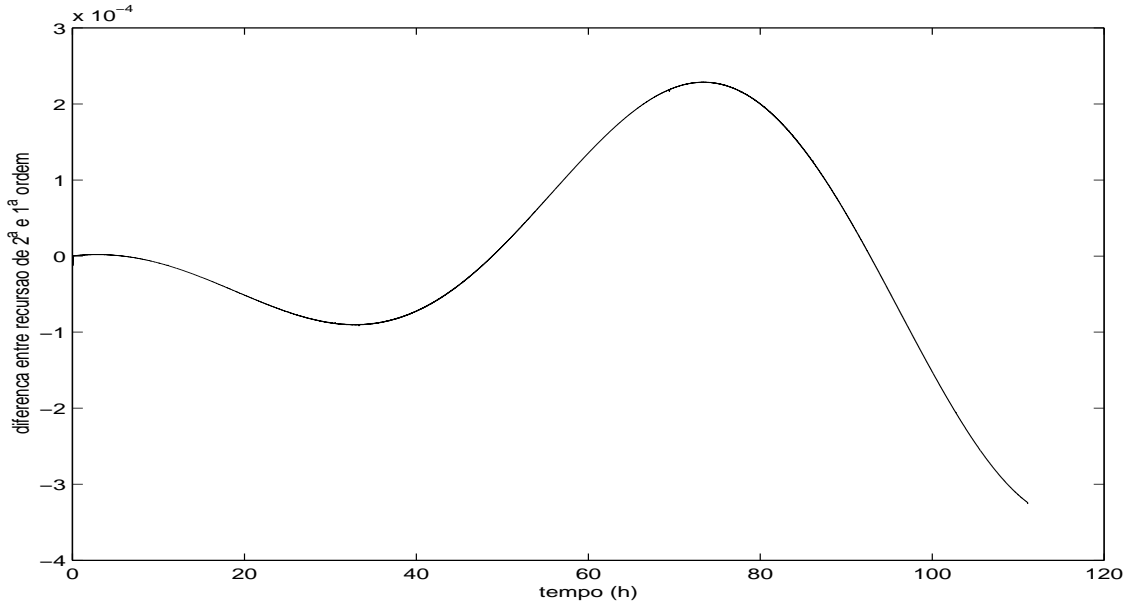


Fig. 5.10 - Evolução Temporal da diferença entre U obtida através da T. Laplace usando recursão de 2ª ordem e recursão de 1ª ordem.

ordem de grandeza 10^{-2} . A matriz de covariância de ruído de observação é formada conforme

$$\vec{r}_{n+1} = \vec{Z}_{n+1} - \vec{Z}_{n+1}^f \quad (5.38)$$

$$r_{max} = \max_i [r_i^2(n+1)] \quad (5.39)$$

$$\mathbf{R}_{n+1} = \begin{bmatrix} r_{max} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & r_{max} \end{bmatrix} . \quad (5.40)$$

A matriz de covariância de erro de assimilação a priori foi escolhida partindo da hipótese muito otimista de que o estado inicial da atmosfera fosse perfeitamente conhecido:

$$\mathbf{P}_0^a = 0 \quad . \quad (5.41)$$

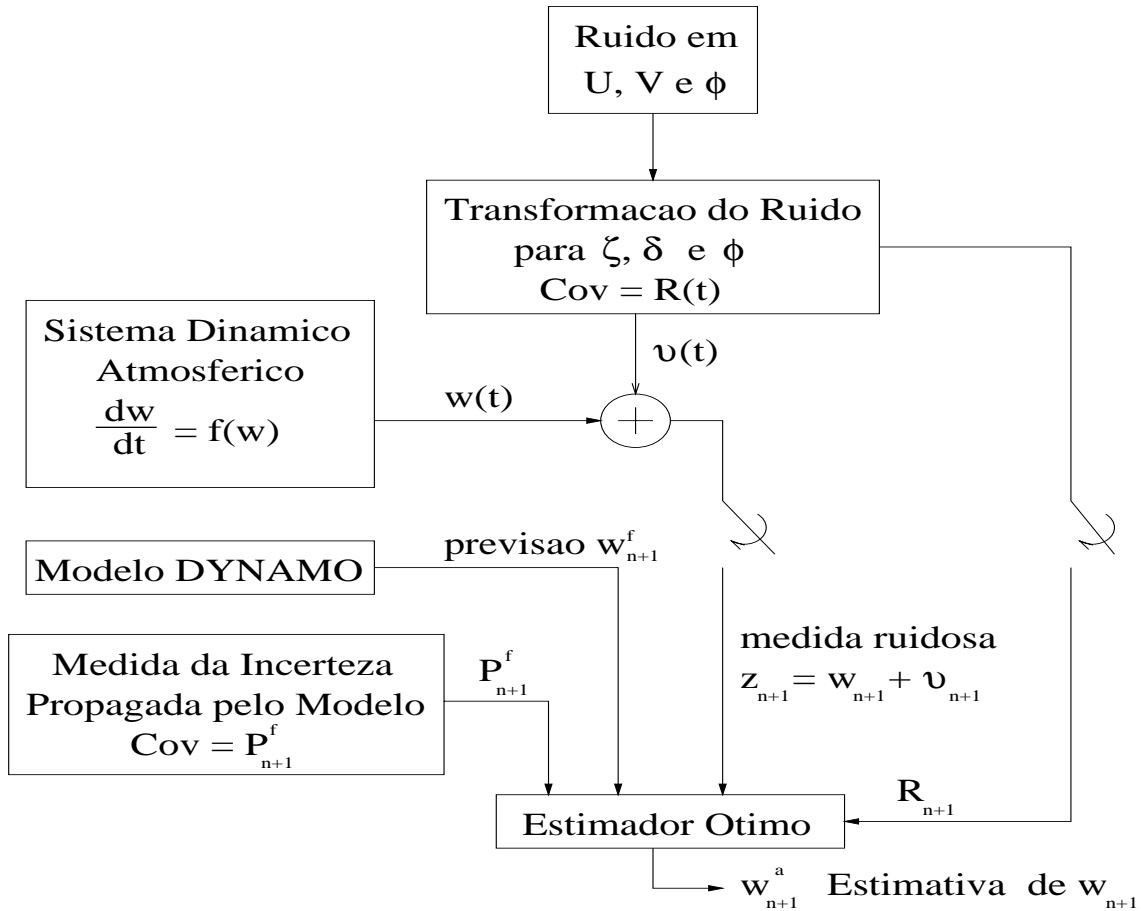


Fig. 5.11 - Implementação do Filtro de Kalman

Conforme a Equação 2.37, isto fará com que o algoritmo adaptativo procure extrair o máximo possível de informação das observações nas primeiras assimilações. A implementação do filtro é mostrada na Figura seguinte 5.11.

Em todas as Figuras seguintes descrevendo o comportamento das quantidades U , V , ϕ , ζ e δ as linhas tracejadas representam os valores da atmosfera verdadeira e as linhas cheias os resultados da assimilação. Nas Figuras subseqüentes descrevendo o comportamento dos erros as linhas representam os erros previstos e as linhas cheias os erros verdadeiros.

Os resultados obtidos no mesmo ponto ponto de grade usando o Filtro de Kalman Linear são mostrados nas Figuras a seguir 5.12 e 5.13. A Figura 5.12 mostra que a

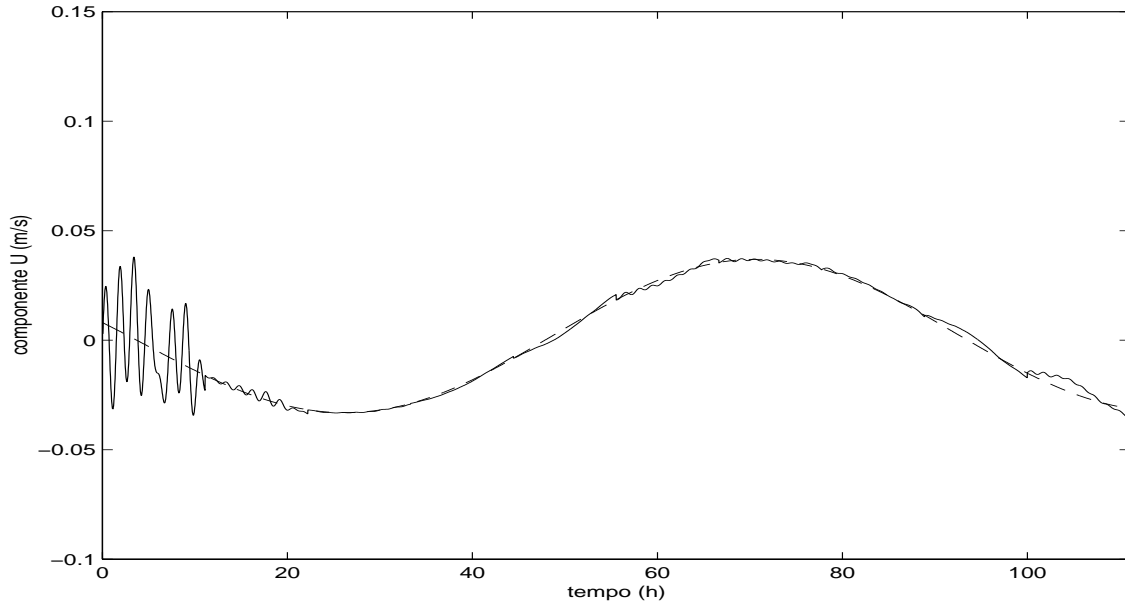


Fig. 5.12 - Evolução Temporal de U na assimilação usando Filtro de Kalman Linear.

assimilação teve problema significativo para $t \geq 90 h$. Isto também apareceu no erro verdadeiro como pode ser visto na Figura 5.13. Esta última Figura também mostra que o filtro teve dificuldade em estimar o erro de assimilação em ζ , δ e ϕ .

O quarto passo foi testar o Filtro de Kalman Estendido, usando as mesmas observações \vec{z}_n . A única diferença entre ele e o Filtro de Kalman Linear é que usa-se agora o modelo DYNAMO completo, com $\vec{N}_{\vec{W}}(\vec{W}, \vec{X}) \neq \vec{0}$.

Os resultados obtidos no mesmo ponto de grade usando o Filtro de Kalman Estendido são mostrados nas Figuras 5.14 e seguinte 5.15. A Figura 5.14 mostra que a assimilação foi aparentemente bem-sucedida. Entretanto, a Figura 5.15 mostra, primeiramente, que o filtro tem dificuldade para estimar o erro de assimilação em ζ , δ e ϕ . Segundo, e pior, as duas curvas de erro pareceram indicar que o erro poderia crescer com o tempo.

O quinto passo foi testar o Filtro de Kalman Adaptativo, usando as mesmas observações \vec{z}_n . Como no Filtro de Kalman Estendido, foi usado o modelo DYNAMO

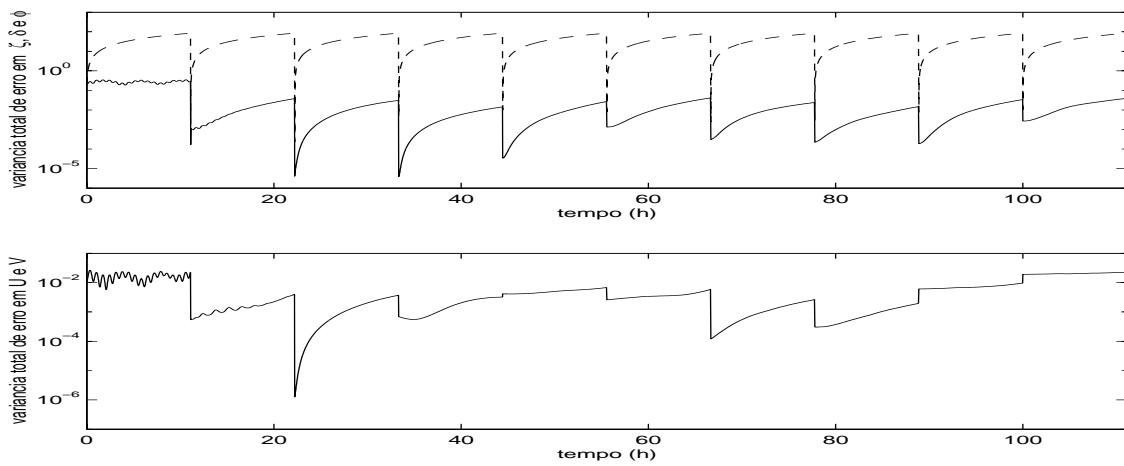


Fig. 5.13 - Variâncias Totais de erro na assimilação usando Filtro de Kalman Linear.

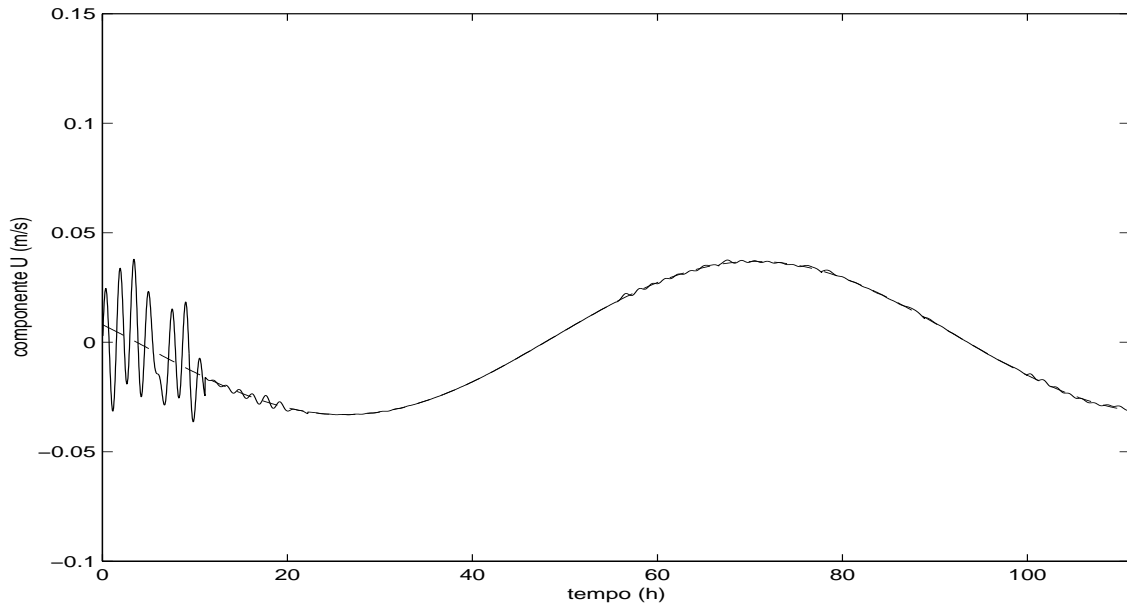


Fig. 5.14 - Evolução temporal de U na assimilação usando Filtro de Kalman Estendido.

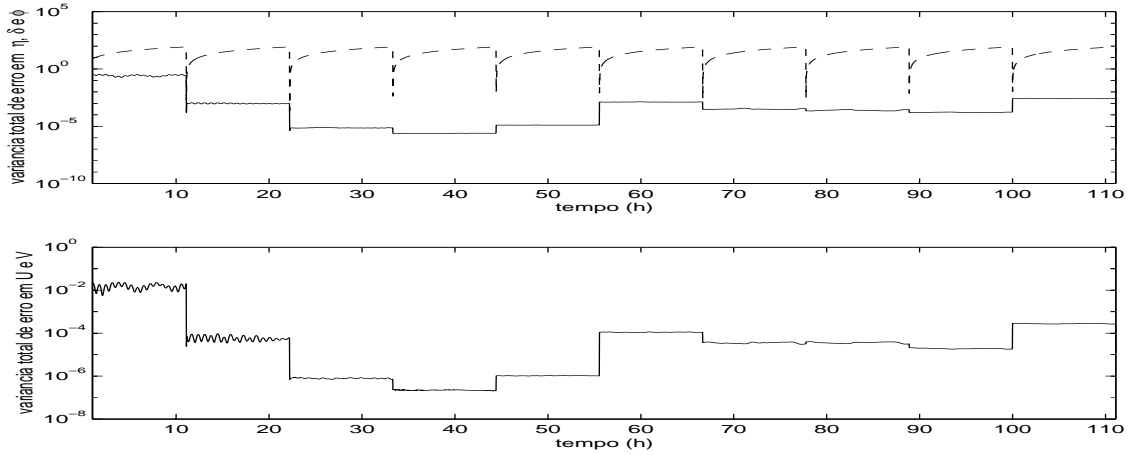


Fig. 5.15 - Variâncias totais de erro na assimilação usando Filtro de Kalman Estendido.

completo, com $N_W(W, X) \neq \vec{0}$. Adicionalmente, adotou-se:

$$\mathbf{P}_0^{qa} = \mathbf{R}_0^q \quad (5.42)$$

$$\alpha = 1 \quad (5.43)$$

$$\gamma = 0 \quad (5.44)$$

O valor de γ foi escolhido supondo-se que \mathbf{Q}_n não era estacionária, como seria no caso de previsão do tempo real. Como o mesmo objetivo a matriz \mathbf{P}_0^{qa} foi escolhida de maneira que o algoritmo lentamente estimasse \mathbf{Q}_n . \mathbf{P}_0^{qa} e α foram experimentalmente determinados para a obtenção de resultado satisfatório.

Os resultados obtidos no ponto médio da grade são mostrados nas Figuras seguintes 5.16 e 5.17. As Figuras mostram que a assimilação das grandezas U , V , ϕ , ζ and δ foi bem sucedida para $t \geq 25 h$. A Figura 5.17 mostra que para $t \geq 25 h$ o erro de assimilação foi consistente com o erro verdadeiro e que o erro nas grandezas U and V foi constante.

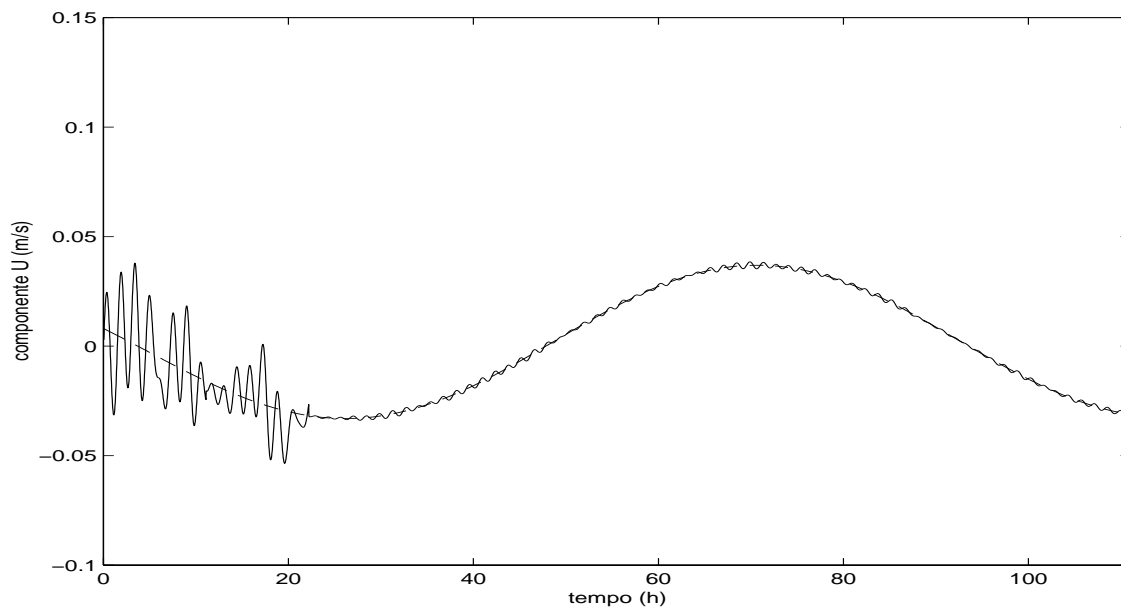


Fig. 5.16 - Evolução Temporal de U na assimilação usando Filtro de Kalman Adaptativo.

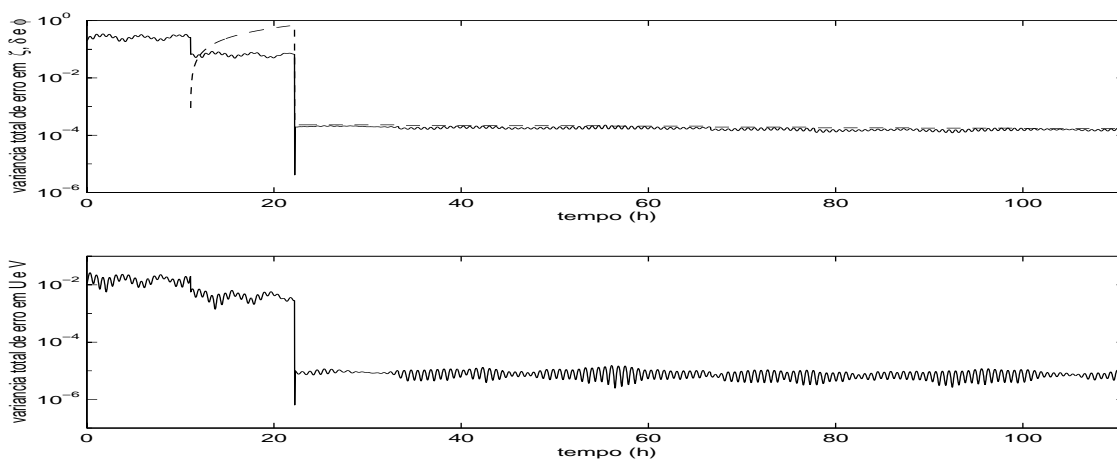


Fig. 5.17 - Variâncias totais de erro na assimilação usando Filtro de Kalman Adaptativo.

Comparando os resultados obtidos usando os Filtros de Kalman o melhor desempenho foi o do Filtro de Kalman Adaptativo.

Comparando-se as Figuras 5.9 (T. Laplace) and 5.16 (F. de Kalman Adaptativo), chegou-se à conclusão de que o melhor resultado foi obtido quando a Transformada de Laplace foi usada. Entretanto, este Filtro de Kalman Adaptativo mostrou-se uma alternativa válida aos Filtros de Kalman Linear e Estendido. Além do mais, os resultados da seção 5.2 sugerem que o desempenho da T. Laplace em sistemas dinâmicos fortemente não lineares, possíveis de ocorrer na física da atmosfera, pode não ser confiável.

Ainda foram feitos três testes de robustez com o Filtro de Kalman Adaptativo. No primeiro teste os parâmetros R_0 , R_F e R_β do modelo do filtro foram levemente mudados para

$$R'_0 = 0.9 \times R_0 \quad , \quad (5.45)$$

$$R'_F = 0.9 \times R_F \quad , \quad (5.46)$$

$$R'_\beta = 0.9 \times R_\beta \quad . \quad (5.47)$$

Os resultados obtidos no ponto médio da grade são mostrados nas Figuras seguintes 5.18 e 5.19. A Figura 5.18 mostra que a assimilação das grandezas U , V , ϕ , ζ e δ foi bem sucedida para $t \geq 25 h$. A Figura 5.19 mostra que para $t \geq 25 h$ o erro de assimilação foi consistente com o erro verdadeiro e o erro nas grandezas U e V aumentou mas tornou-se constante para $t \geq 55 h$.

No segundo teste foi adotado um ruído de modelagem gaussiano branco de média nula (veja Equação 2.15) $\vec{\mu}'_{n+1}$ de ordem de grandeza $10^{-5} \times \vec{W}_{n+1}$:

$$\vec{W}_{n+1} = \vec{F}(\vec{W}_n) + \vec{\mu}'_{n+1} \quad . \quad (5.48)$$

Os resultados obtidos no ponto médio da grade são mostrados nas próximas Figuras 5.20 e 5.21. A Figura 5.20 mostra que a assimilação teve problema significativa para $t \geq 45 h$. Isto também pareceu nos erros verdadeiros na Figura 5.21, onde pode ser

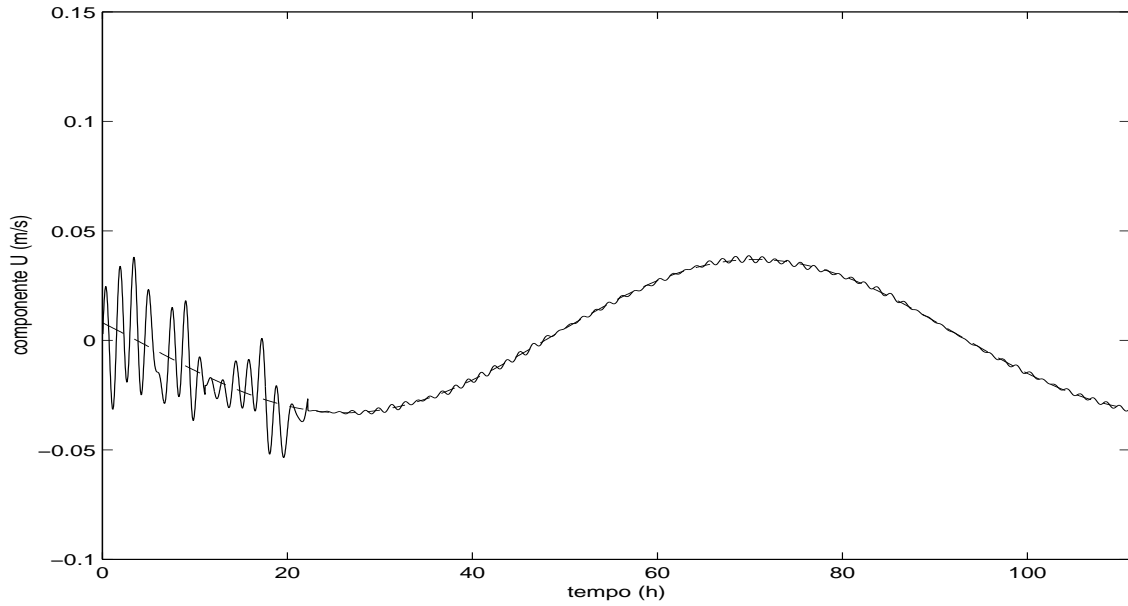


Fig. 5.18 - Evolução temporal de U na assimilação usando Filtro de Kalman Adaptativo com erro nos parâmetros R_O , R_F and R_β .

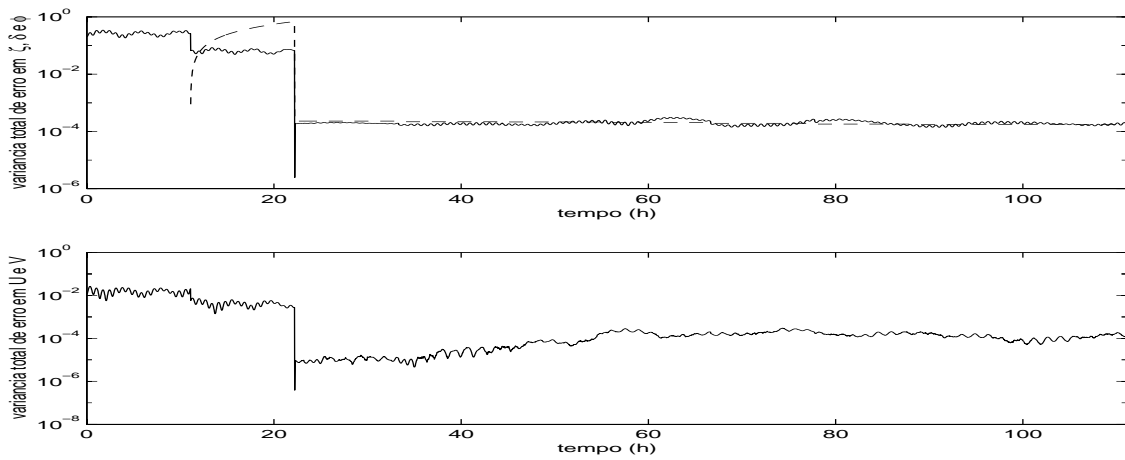


Fig. 5.19 - Variâncias totais de erro na assimilação usando Filtro de Kalman Adaptativo com erro nos parâmetros R_O , R_F e R_β .

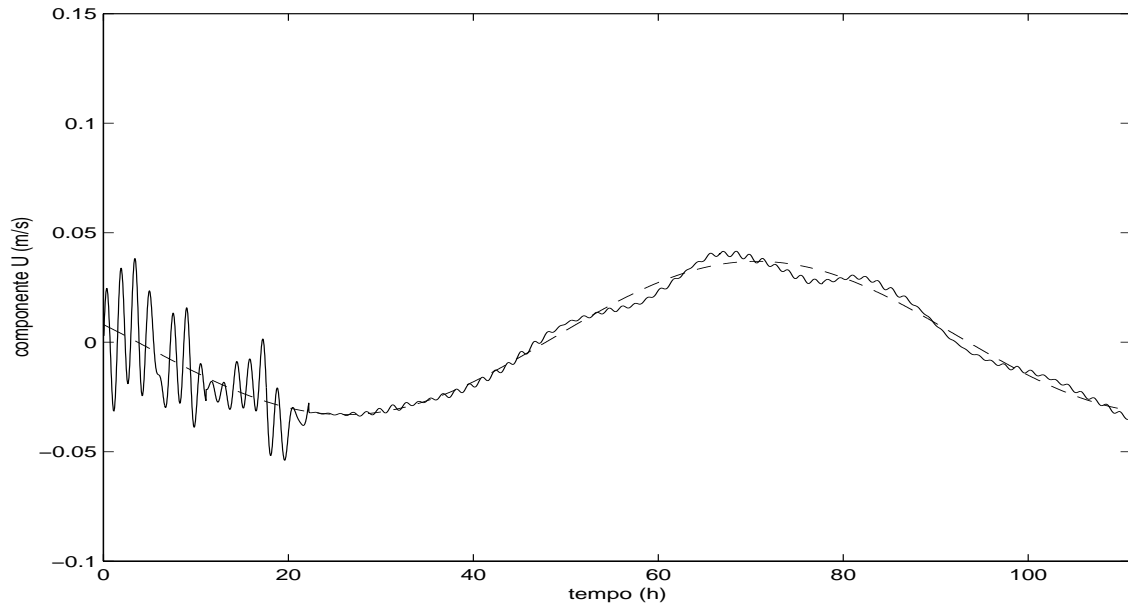


Fig. 5.20 - Evolução temporal de U na assimilação usando Filtro de Kalman Adaptativo com adição de erro de modelagem μ' .

visto que há divergência significativa entre os erros previstos e verdadeiros. Talvez neste caso o filtro precisasse receber observações em intervalos $\Delta t < 11.1 h$ para desempenhar satisfatoriamente. Ou, comparando com o primeiro teste de robustez, o erro introduzido pode ter sido grande demais em algumas componentes.

No terceiro teste de robustez os parâmetros R_0 , R_F e R_β do modelo para o sinal *verdadeiro* ("a atmosfera") foram alterados usando flutuações gaussianas brancas de média nula

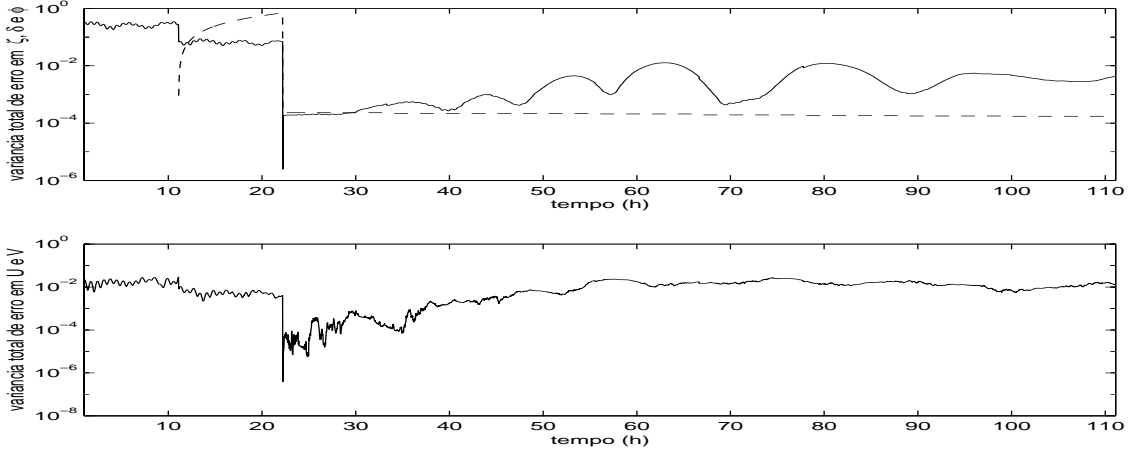


Fig. 5.21 - Variâncias totais de erro na assimilação usando Filtro de Kalman Adaptativo com adição de erro de modelagem μ' .

$$R'_0(n) = [1 + \mu_O(n)] \times R_0 \quad , \quad (5.49)$$

$$R'_F(n) = [1 + \mu_F(n)] \times R_F \quad , \quad (5.50)$$

$$R'_\beta(n) = [1 + \mu_\beta(n)] \times R_\beta \quad , \quad (5.51)$$

$$E \{ \mu_O(n) \} = E \{ \mu_F(n) \} = E \{ \mu_\beta(n) \} = 0 \quad , \quad (5.52)$$

$$E \{ \mu_O^2(n) \} = E \{ \mu_F^2(n) \} = E \{ \mu_\beta^2(n) \} = 1.6 \times 10^{-3} \quad , \quad (5.53)$$

onde $E\{x\}$ denota *valor esperado*. Tanto o sinal verdadeiro quanto a condição inicial do modelo do filtro foram inicializados. Este foi o teste mais próximo da assimilação de dados operacional, porque a atmosfera não é realmente um sistema dinâmico de parâmetros invariantes no tempo como suposto no modelo DYNAMO e, por outro lado, a condição inicial do sistema de assimilação geralmente estará inicializada.

Os resultados obtidos no ponto médio da grade são mostrados nas Figuras seguintes 5.22 e 5.23. A assimilação apenas mostrou problema significativo para $60 \leq t \leq 100 h$. Os gráficos de erro da Figura 5.23 clarificam que os erros verdadeiros atingem seu valores máximos no intervalo $60 \leq t \leq 100 h$. Comparando com o segundo teste de robustez, talvez novamente ali o erro introduzido tenha sido grande demais em

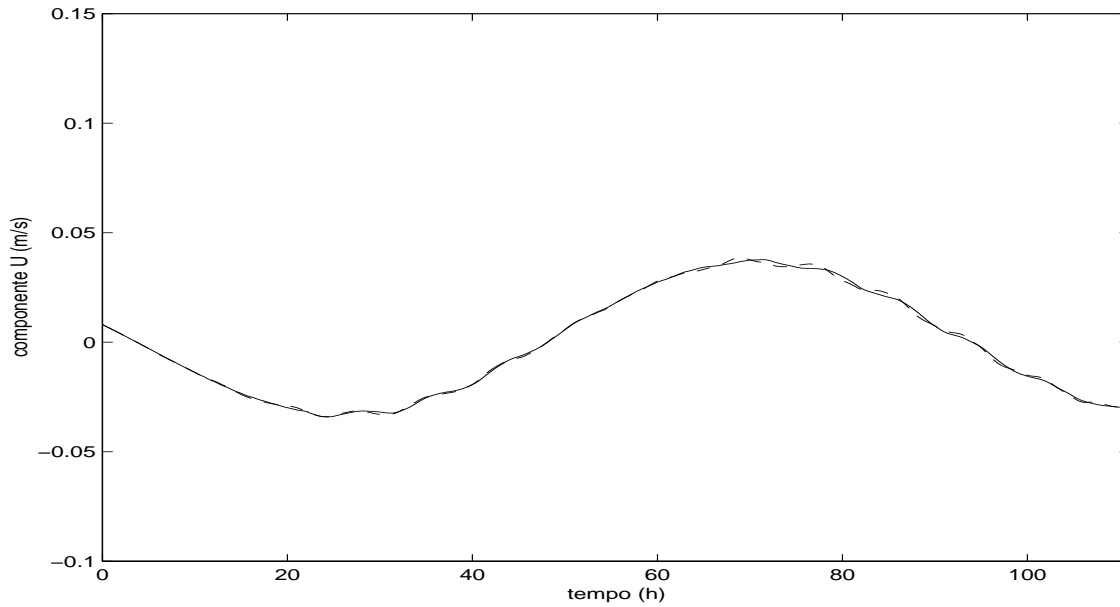


Fig. 5.22 - Evolução temporal de U na assimilação usando Filtro de Kalman Adaptativo com mudanças temporais nos parâmetros R_O , R_F e R_β do *signal verdadeiro*.

algumas componentes.

Por último, mas não menos importante, este teste de robustez foi feito com a Transformada de Laplace. Os resultados obtidos no ponto médio da grade são mostrados na última Figura deste capítulo, 5.24. A assimilação mostrou problema significativo para $20 \leq t \leq 30 h$ e $60 \leq t \leq 90h$. Convém lembrar que o sistema de assimilação deveria rastrear o *signal verdadeiro*. A integração usando Transformada de Laplace pareceu incapaz de rastrear precisamente esta atmosfera com parâmetros variantes no tempo. Provavelmente isto ocorreu porque o sinal de erro contém frequências acima dos modos do modelo atmosférico (DYNAMO), e a inversão de Laplace simplesmente elimina estas frequências (verifique fórmulas 2.12 e 2.13). O Filtro de Kalman Adaptativo, por outro lado, pôde se adaptar para assimilar informação em frequências acima dos modos do modelo escolhido. Isto é, *a assimilação de Laplace falha quando a nova informação tem frequências acima do esperado a priori*.

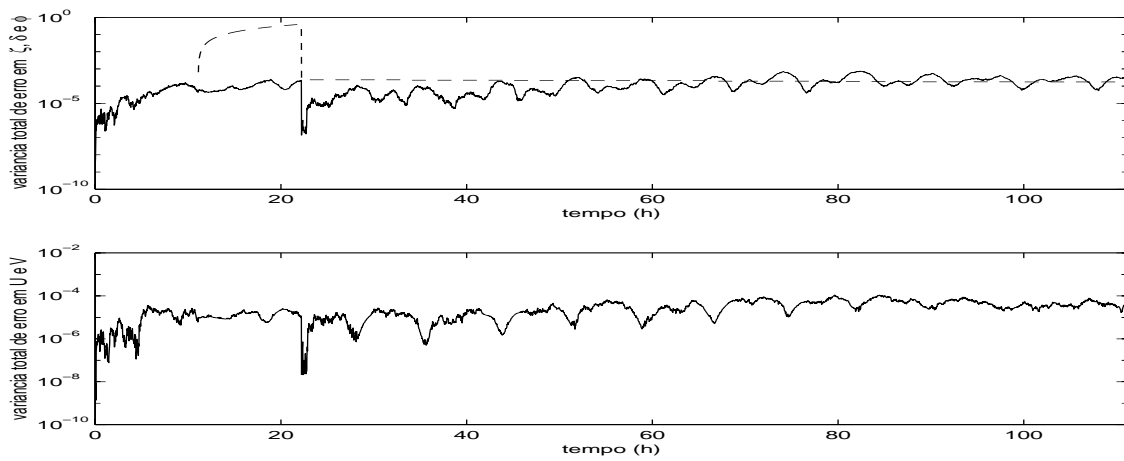


Fig. 5.23 - Variâncias totais de erro na assimilação usando Filtro de Kalman Adaptativo com mudanças temporais nos parâmetros R_O , R_F e R_β do *signal verdadeiro*.

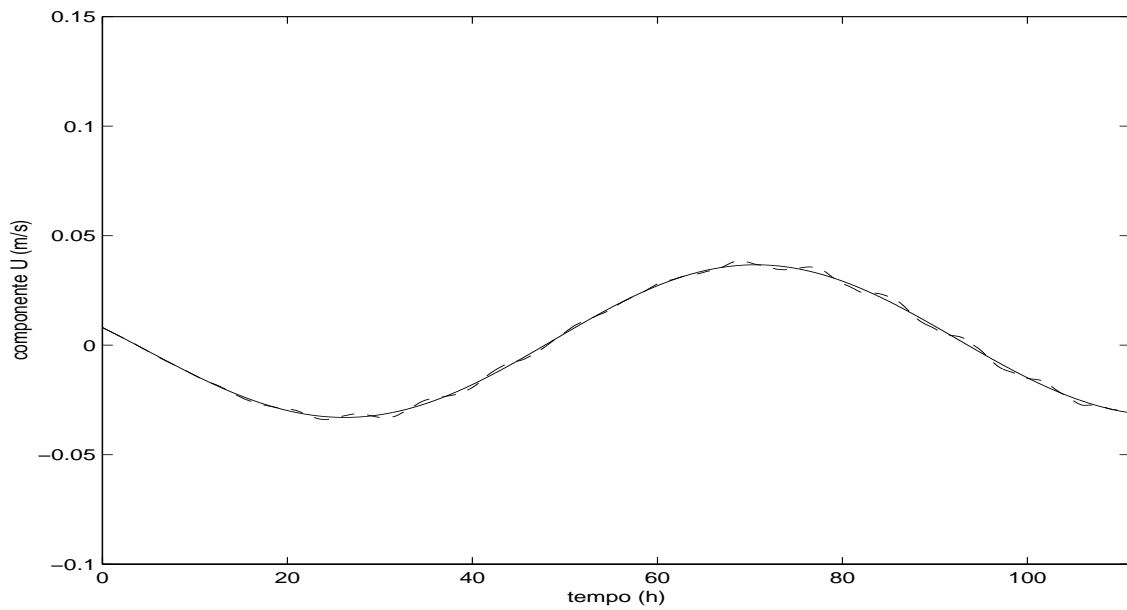


Fig. 5.24 - Evolução temporal de U na assimilação usando Transformada de Laplace com mudanças temporais nos parâmetros R_O , R_F e R_β do *signal verdadeiro*.

No próximo Capítulo serão mostrados os resultados obtidos na assimilação usando redes neurais. Os resultados daquele Capítulo serão comparados com os deste.

CAPÍTULO 6

RESULTADOS NUMÉRICOS DAS REDES NEURAIIS

O desempenho da assimilação usando Redes Neurais de Perceptrons Multicamadas também foi avaliado. O primeiro teste foi feito usando o sistema de Hénon citado na seção 4.1. O segundo teste foi feito usando o sistema de Lorenz citado na seção 4.2. O terceiro teste foi feito usando o modelo DYNAMO citado na seção 4.3.

O algoritmo de treinamento em retropropagação em modo "lote" (*batch*) foi adaptado do programa em MATLAB do pacote computacional *MATLAB Neural Network Toolbox (MNNT)* [11] para FORTRAN. O pacote computacional inclui o método de Nguyen-Widrow para inicialização de pesos e limiares que também foi adotado.

6.1 Resultados com o Sistema de Hénon Caótico

O primeiro experimento numérico consistiu em treinar uma rede neural de perceptrons multicamada como escrito na seção 3 para emular o FKA usado na seção 5.1. Um conjunto de exemplos foi gerado assimilando dados em *todos* os passos da evolução do sistema. Este conjunto foi usado para treinar uma rede consistindo de

- entradas $w_1^f, w_2^f, w_1^o, w_2^o$;
- *primeira* camada escondida tendo 2 neurônios com função de ativação $f(x) = \tanh(x)$;
- *segunda* camada escondida tendo 2 neurônios com função de ativação $f(x) = \tanh(x)$;
- camada de saída tendo 2 neurônios com função de ativação $f(x) = x$ e saídas w_1^a, w_2^a ;

foi treinada colocando nas suas camadas de entrada e saída respectivamente os pares (x_n^i, w_n^a) formados pelos vetores

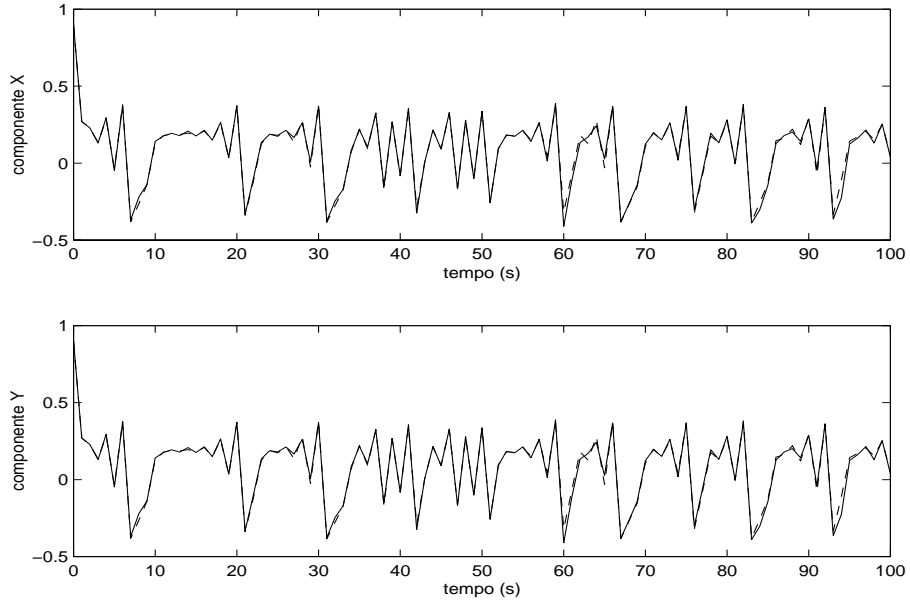


Fig. 6.1 - Assimilação de dados no sistema de Hénon usando Rede Neural.

$$\vec{x}_n^e = \begin{bmatrix} \begin{bmatrix} w_1^f(n) \\ w_2^f(n) \end{bmatrix} \\ \begin{bmatrix} w_1^o(n) \\ w_2^o(n) \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \vec{w}_n^f \\ \vec{w}_n^o \end{bmatrix} \quad (6.1)$$

$$\vec{w}_n^a = \begin{bmatrix} w_1^a(n) \\ w_2^a(n) \end{bmatrix} . \quad (6.2)$$

para todos n . A rede foi treinada com taxa de treinamento constante $\eta = 10^{-4}$ até $e_m < 0.01$. Para testar a nova abordagem colocaram-se na entrada da rede treinada

os vetores

$$\vec{x}_n^e = \begin{bmatrix} X_n^f \\ Y_n^f \\ X_n^o \\ Y_n^o \end{bmatrix} \quad (6.3)$$

onde X_n^f e Y_n^f foram preditos pelas Equações de Hénon e X_n^o e Y_n^o eram observações. Em seguida foram lidos na camada de saída os vetores de assimilação

$$\vec{w}_n^a = \begin{bmatrix} \hat{X}_n^a \\ \hat{Y}_n^a \end{bmatrix} . \quad (6.4)$$

O resultado inserindo dados a cada $\Delta t = 2 s$ pode ser visto na Figura 6.1. As linhas tracejadas representam o sinal verdadeiro, as linhas cheias o sinal estimado. Apesar de a assimilação não ter sido perfeita, o desvio caótico na Figura anterior 4.1 parece ter sido evitado. Comparando-se a Figura 6.1 com a Figura 5.1 da seção 5.1 pode-se concluir que a o desempenho da rede neural não foi pior que o do FKA.

6.2 Resultados com o Sistema de Lorenz Caótico

O segundo experimento numérico consistiu em treinar uma rede neural de perceptrons multicamada para emular o FKE usado na seção 5.2. Levando-se em consideração que os parâmetros do sistema de Lorenz não mudam ao longo do tempo, o FKE pode ser escolhido para treinamento da rede porque o algoritmo de aprendizagem necessita de exemplos precisos. Uma rede composta de

- entradas $w_1^f, w_2^f, w_3^f, w_1^o, w_2^o, w_3^o$;
- *primeira* camada escondida tendo 2 neurônios com função de ativação $f(x) = \tanh(x)$;
- *segunda* camada escondida tendo 2 neurônios com função de ativação $f(x) = \tanh(x)$;

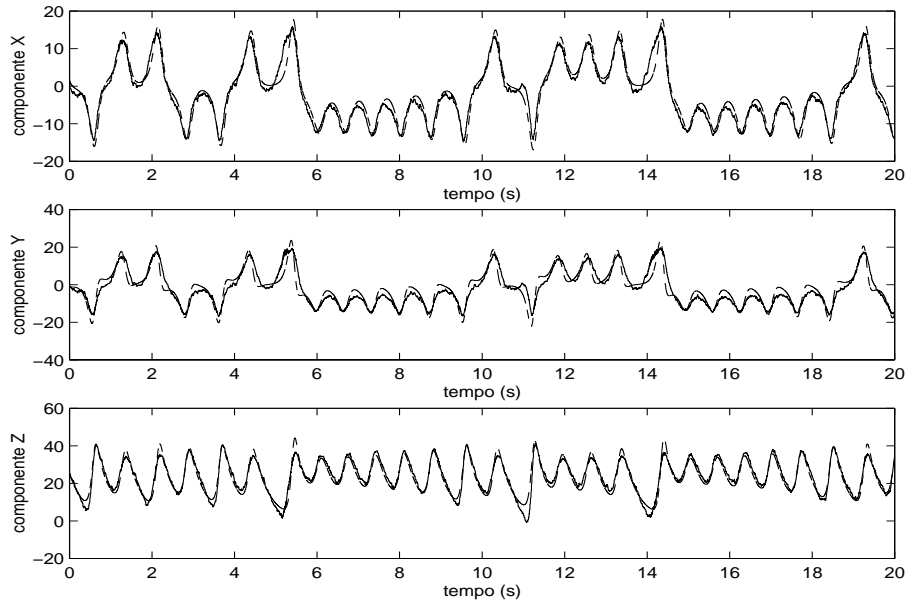


Fig. 6.2 - Assimilação de dados no sistema de Lorenz usando Rede Neural.

- camada de saída tendo 2 neurônios com função de ativação $f(x) = x$ e saídas w_1^a, w_2^a, w_3^a

foi treinada.

Para geração de exemplos, ao invés de assimilar os dados a cada 0.012 s o FKE recebeu dados a cada 0.001 s. A rede foi treinada colocando-se nas suas camadas de entrada e saída respectivamente os pares (\vec{x}_n^e, w_n^a) formados pelos vetores

$$\vec{x}_n^e = \begin{bmatrix} \begin{bmatrix} \hat{w}_1^f(n) \\ \hat{w}_2^f(n) \\ \hat{w}_3^f(n) \end{bmatrix} \\ \begin{bmatrix} \hat{w}_1^o(n) \\ \hat{w}_2^o(n) \\ \hat{w}_3^o(n) \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \hat{w}^f(n) \\ \hat{w}^o(n) \end{bmatrix} \quad (6.5)$$

$$\hat{w}_n^a = \begin{bmatrix} \hat{w}_1^a(n) \\ \hat{w}_2^a(n) \\ \hat{w}_3^a(n) \end{bmatrix} . \quad (6.6)$$

Os vetores foram normalizados para que os sinais de entrada e saída estivessem no domínio $[-1, 1]$. Assim,

$$w_{\max} = \max_{i,n} \{ |w_i^f(n)|, |w_i^o(n)|, |w_i^a(n)| \} \quad (6.7)$$

$$\hat{w}_i^f(n) = w_i^f(n)/w_{\max} \quad (6.8)$$

$$\hat{w}_i^o(n) = w_i^o(n)/w_{\max} \quad (6.9)$$

$$\hat{w}_i^a(n) = w_i^a(n)/w_{\max}, \quad i = 1, 2, 3, \quad \forall n . \quad (6.10)$$

Aqui $w_{\max} = 40$. A rede foi treinada com o algoritmo de retropropagação padrão (seção 3.5) usando os primeiros 2000 exemplos do FKE com $\eta = 10^{-4}$ constante até que $e_m < 10$ (correspondente a erro quadrático médio de 0.05). Apenas 10% dos exemplos foram usados. Para testar o novo método de assimilação colocou-se na entrada da rede treinada os vetores

$$\vec{x}_n^e = \begin{bmatrix} X_n^f \\ Y_n^f \\ Z_n^f \\ X_n^o \\ Y_n^o \\ Y_n^o \end{bmatrix} \cdot \frac{1}{w_{\max}} \quad (6.11)$$

e leram-se na camada de saída os vetores

$$\hat{w}_n^a = \begin{bmatrix} \hat{X}_n^a \\ \hat{Y}_n^a \\ \hat{Z}_n^a \end{bmatrix} . \quad (6.12)$$

X_n^f , Y_n^f e Z_n^f foram previstos pelas Equações de Lorenz e X_n^o , Y_n^o and Z_n^o eram observações. A saída da assimilação era

$$\tilde{X}_n^a = w_{\max} \times \hat{X}_n^a \quad , \quad \tilde{Y}_n^a = w_{\max} \times \hat{Y}_n^a \quad , \quad \tilde{Z}_n^a = w_{\max} \times \hat{Z}_n^a \quad . \quad (6.13)$$

O resultado inserindo dados a cada $\Delta t = 0.012$ s pode ser visto na Figura 6.2. A linha tracejada representa o sinal verdadeiro, a linha cheia o sinal estimado. Mesmo que a assimilação não tenha sido perfeita, o desvio caótico visto na Figura anterior 4.2 parece ter sido evitado em todas as componentes. Comparando com os resultados da T. Laplace e FKE (Figura 5.3) que foram mostrados na seção (5.2), o FKE apresentou um desempenho melhor. Ainda assim, a assimilação encorajou os autores a acreditar que o novo método poderia desempenhar-se aceitavelmente com sistemas que tivessem não-linearidade mais fraca e ao mesmo tempo apresentassem comportamento similar a caótico, como o modelo DYNAMO. Realmente, como será visto na próxima seção, a crença foi confirmada.

6.3 Resultados com o modelo DYNAMO

O terceiro experimento numérico consistiu em treinar redes neurais de perceptrons multicamadas para emular o FKA usado na seção 5.3.

O FKA foi usado para gerar os exemplos para uma rede composta de

- entradas \vec{U}^o , \vec{V}^o , $\vec{\phi}^o$, \vec{U}^f , \vec{V}^f , $\vec{\phi}^f$;
- uma camada escondida tendo 80 neurons com função de ativação $f(x) = \tanh(x)$;
- camada de saída tendo 60 neurônios com função de ativação $f(x) = x$ e saídas U^a , V^a , ϕ^a .

Ao invés de assimilar os dados a cada 11.1 h o filtro recebeu dados a cada 0.0278 h (um passo de tempo). A rede foi treinada colocando-se na sua entrada e saída respectivamente os pares $(\vec{x}_n^e, \vec{w}_n^a)$ formados pelos vetores

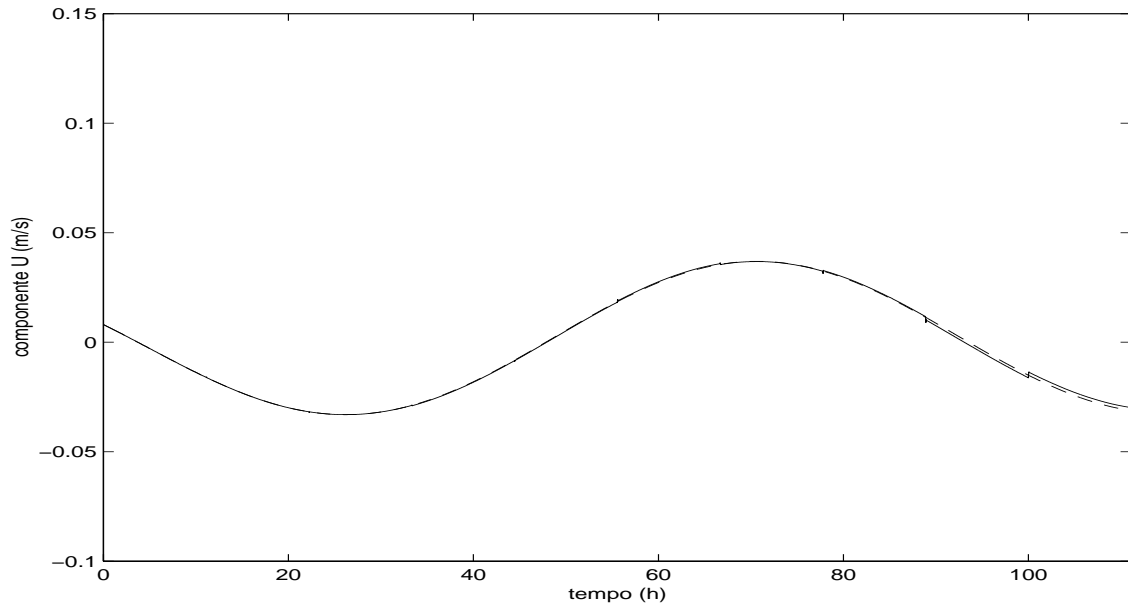


Fig. 6.3 - Evolução temporal de U em assimilação usando Rede Neural com uma camada escondida e algoritmo de retropropagação adaptativo.

$$\vec{x}_n^e = \begin{bmatrix} \begin{bmatrix} \hat{U}_n^o \\ \hat{V}_n^o \\ \hat{\phi}_n \end{bmatrix} \\ \begin{bmatrix} \hat{U}_n^f \\ \hat{V}_n^f \\ \hat{\phi}_n \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \hat{w}^f(n) \\ \hat{w}^o(n) \end{bmatrix} \quad (6.14)$$

$$\vec{w}_n^a = \begin{bmatrix} \hat{U}_n^a \\ \hat{V}_n^a \\ \hat{\phi}_n^a \end{bmatrix} . \quad (6.15)$$

Os vetores foram normalizados com escala para que os sinais de entrada e saída

fossem da mesma ordem de magnitude. Usando como referência

$$U_{\max} = 0.05, \quad V_{\max} = 2.00, \quad \phi_{\max} = 1.50, \quad (6.16)$$

respectivamente os valores absolutos máximos de \vec{U}_n^a , \vec{V}_n^a e $\vec{\phi}_n^a$ dados pelo filtro de Kalman no ponto médio da grade $i = \frac{N}{2}$ para todos n , fatores de escala

$$U^* = \alpha \cdot U_{\max}, \quad V^* = \alpha \cdot V_{\max}, \quad \phi^* = \alpha \cdot \phi_{\max} \quad (6.17)$$

foram definidos usando α determinado experimentalmente para resultado satisfatório. Neste caso $\alpha = 2.60$ e todos os sinais estiveram no domínio $[-1, 1]$. Portanto

$$U^* = 0.13, \quad V^* = 5.20, \quad \phi^* = 3.90, \quad (6.18)$$

$$\hat{U}_i^f(n) = U_i^f(n)/U^* \quad \hat{V}_i^f(n) = V_i^f(n)/V^* \quad \hat{\phi}_i^f(n) = \phi_i^f(n)/\phi^* \quad (6.19)$$

$$\hat{U}_i^o(n) = U_i^o(n)/U^* \quad \hat{V}_i^o(n) = V_i^o(n)/V^* \quad \hat{\phi}_i^o(n) = \phi_i^o(n)/\phi^* \quad (6.20)$$

$$\hat{U}_i^a(n) = U_i^a(n)/U^* \quad \hat{V}_i^a(n) = V_i^a(n)/V^* \quad \hat{\phi}_i^a(n) = \phi_i^a(n)/\phi^*, \quad (6.21)$$

para todos i e todos n no conjunto dos exemplos.

A rede foi treinada com o algoritmo de retropropagação adaptativo (seção 3.5.1) usando 100 exemplos do FKA equidistantes no tempo e $\eta_0 = 10^{-4}$ até $e_m < 0.09$. Este experimento foi denominado *experimento 1*. Tentativas de treinar perceptrons multicamadas com mais camadas escondidas usando o algoritmo de retropropagação adaptativo foram mal-sucedidas.

Para testar o novo método de assimilação, colocaram-se na entrada da rede treinada os vetores

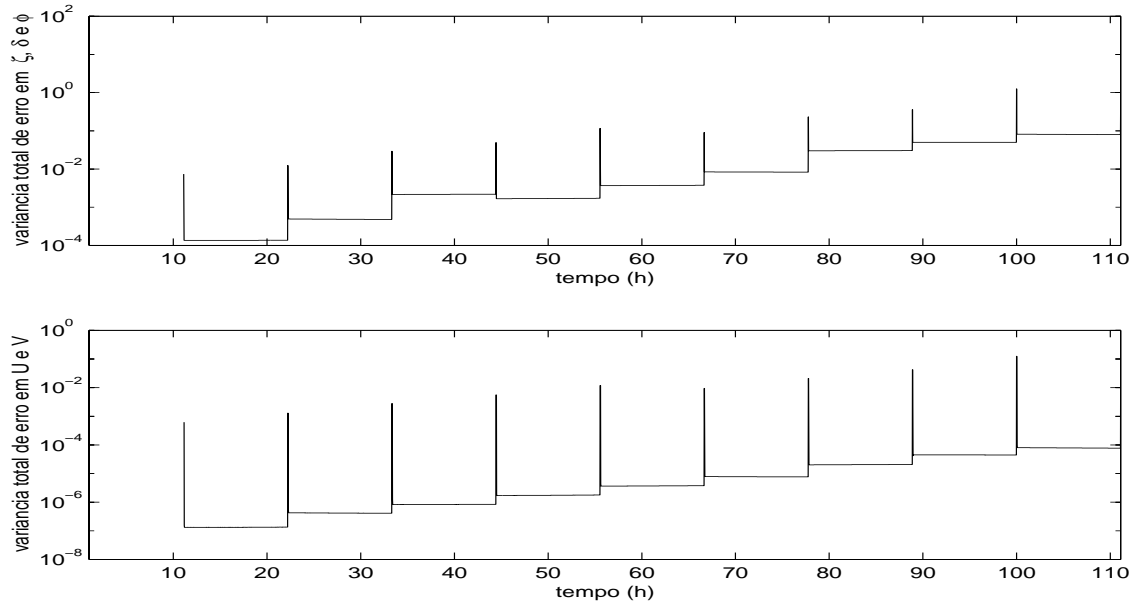


Fig. 6.4 - Variâncias totais de erro na assimilação usando Rede Neural com uma camada escondida e algoritmo de retropropagação adaptativo.

$$\vec{x}_n^i = \begin{bmatrix} \vec{U}_n^o / U^* \\ \vec{V}_n^o / V^* \\ \vec{\phi}_n^o / \phi^* \\ \vec{U}_n^f / U^* \\ \vec{V}_n^f / V^* \\ \vec{\phi}_n^f / \phi^* \end{bmatrix} \quad (6.22)$$

e leram-se na camada de saída os vetores

$$\hat{w}_n^a = \begin{bmatrix} \hat{w}_1^a(n) \\ \hat{w}_2^a(n) \\ \hat{w}_3^a(n) \end{bmatrix} . \quad (6.23)$$

\vec{U}_n^f, \vec{V}_n^f and $\vec{\phi}_n^f$ foram previstos por DYNAMO e \vec{U}_n^o, \vec{V}_n^o e $\vec{\phi}_n^o$ eram observações. A saída da assimilação seria

$$\tilde{U}_n^a = U^* \cdot \hat{w}_1^a(n) \quad , \quad \tilde{V}_n^a = V^* \cdot \hat{w}_2^a(n) \quad , \quad \tilde{\phi}_n^a = \phi^* \cdot \hat{w}_3^a(n) \quad . \quad (6.24)$$

Os resultados inserindo dados a cada $\Delta t = 11.1 h$ podem ser vistos na Figura 6.3 e 6.4. As linhas tracejadas representam o sinal verdadeiro, as linhas cheias o sinal previsto. As variáveis U, V e ϕ foram aceitavelmente assimiladas, o erro na Figura 6.3 se tornou significativo para $t \geq 85 h$. Os erros na Figura 6.4 aumentam sensivelmente ao longo do tempo indicando que a rede foi menos confiável do que o FKA que emulava. Os "picos" de erro na Figura 6.4 aparecendo nos instantes de assimilação foram possivelmente devidos à maneira como as observações foram geradas. A observação sintética foi um sinal de modelo somado a ruído (veja Equação 5.31). Este mesmo problema apareceu em todos os experimentos feitos com redes neurais. Esta descontinuidade foi fortemente suavizada na filtragem com Transformada de Laplace e Kalman porque estes métodos usam "memória", isto é, comparam novos dados com antigos para assimilar suavemente a nova informação. É possível que este problema seja resolvido usando-se rede neural recursiva [22].

Depois deste primeiro resultado, o algoritmo de retropropagação modificado (section 3.5.1.1) foi testado num perceptron com *duas* camadas escondidas. O FKA foi usado para prover os exemplos para uma rede composta agora de

- entradas $U^o, V^o, \phi^o, U^f, V^f, \phi^f$;
- *primeira* camada escondida tendo 50 neurônios com função de activação $f(x) = \tanh(x)$;
- *segunda* camada escondida tendo 50 neurônios com função de activação $f(x) = \tanh(x)$;

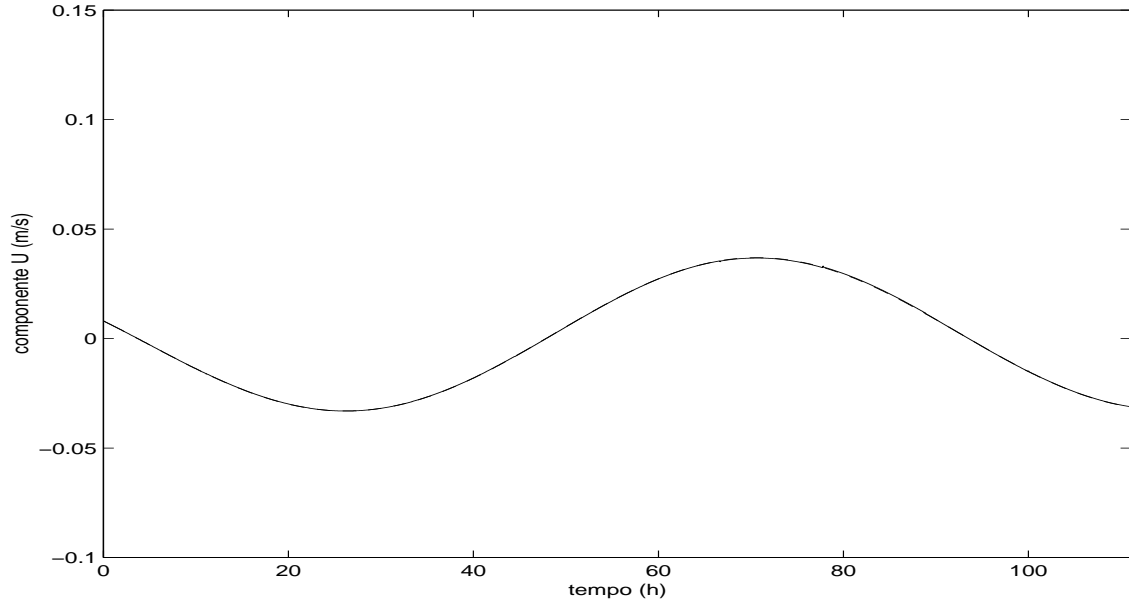


Fig. 6.5 - Evolução temporal de U em assimilação usando Rede Neural com duas camadas escondidas e algoritmo de retropropagação adaptativo modificado.

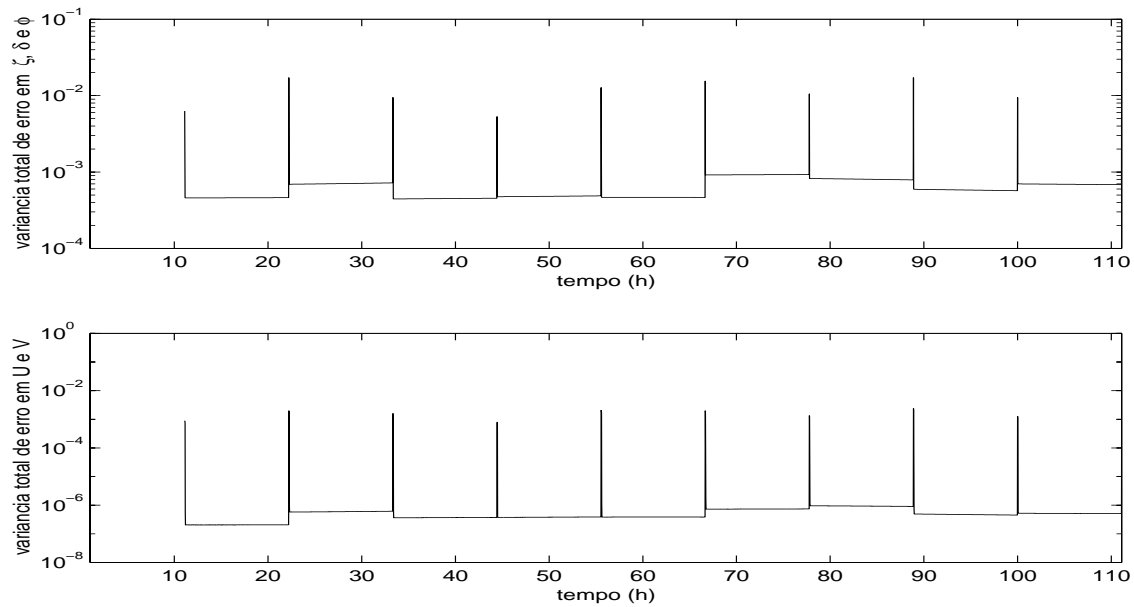


Fig. 6.6 - Variâncias totais de erro na assimilação usando Rede Neural com duas camadas escondidas e algoritmo de retropropagação adaptativo modificado.

- camada de saída tendo 60 neurônios com função de activação $f(x) = x$ e saídas U^a, V^a, ϕ^a .

Como no experimento 1, os vetores foram normalizados com escala usando $\alpha = 1$. Portanto

$$U^* = 0.05, \quad V^* = 2.00, \quad \phi^* = 1.50 \quad . \quad (6.25)$$

A rede foi treinada com o algoritmo de retropropagação adaptativo modificado (seção 3.5.1.1) usando 500 exemplos do FKA equidistantes no tempo e $\eta_0 = 10^{-3}$ até $e_m < 0.6$. A restrição seguinte precisou ser feita na camada de saída (veja passo **3.2.1** do algoritmo) para assegurar precisão numérica:

$$\Delta(\theta y)'_{ij} = \sum_n \Delta_{in} y_{jn} \quad (6.26)$$

$$\Delta(\theta y)_{ij} = \begin{cases} \Delta(\theta y)'_{ij}, & \text{se } |\Delta(\theta y)'_{ij}| \geq 10^{-6} \\ 0, & \text{senão} \quad . \end{cases} \quad (6.27)$$

Este experimento foi denominado *experimento 2*. Os resultados obtidos inserindo dados a cada $\Delta t = 11.1 h$ podem ser vistos nas Figuras 6.5 e 6.6. As linhas tracejadas representam o sinal verdadeiro, as linhas cheias o sinal previsto. As variáveis U , V e ϕ foram satisfatoriamente assimiladas. Os erros na Figura 6.6 pareceram crescer menos ao longo do tempo que no experimento anterior indicando que esta rede foi mais confiável.

Finalmente o terceiro teste de robustez da seção 5.3 foi repetido com o método da rede neural.

Os resultados obtidos no ponto médio da grade são mostrados na Figura 6.7 e nas seguintes 6.8, 6.9, 6.10. A assimilação usando rede neural com apenas uma camada

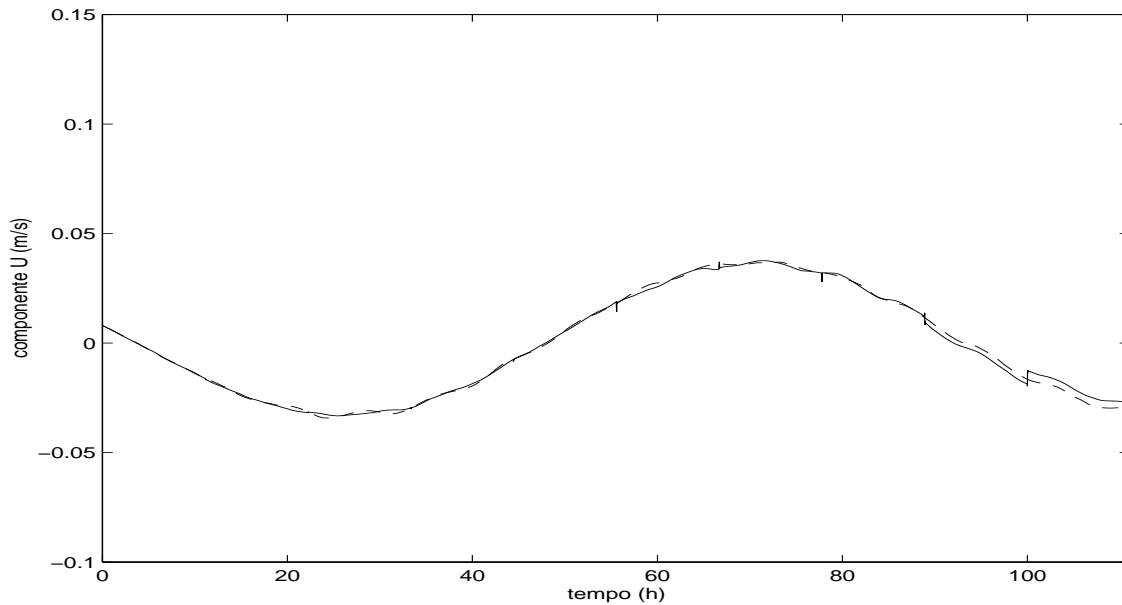


Fig. 6.7 - Evolução temporal de U na assimilação usando Rede Neural com uma camada escondida, algoritmo de retropropagação adaptativo e mudanças temporais nos parâmetros R_O , R_F and R_β do *senal verdadeiro*.

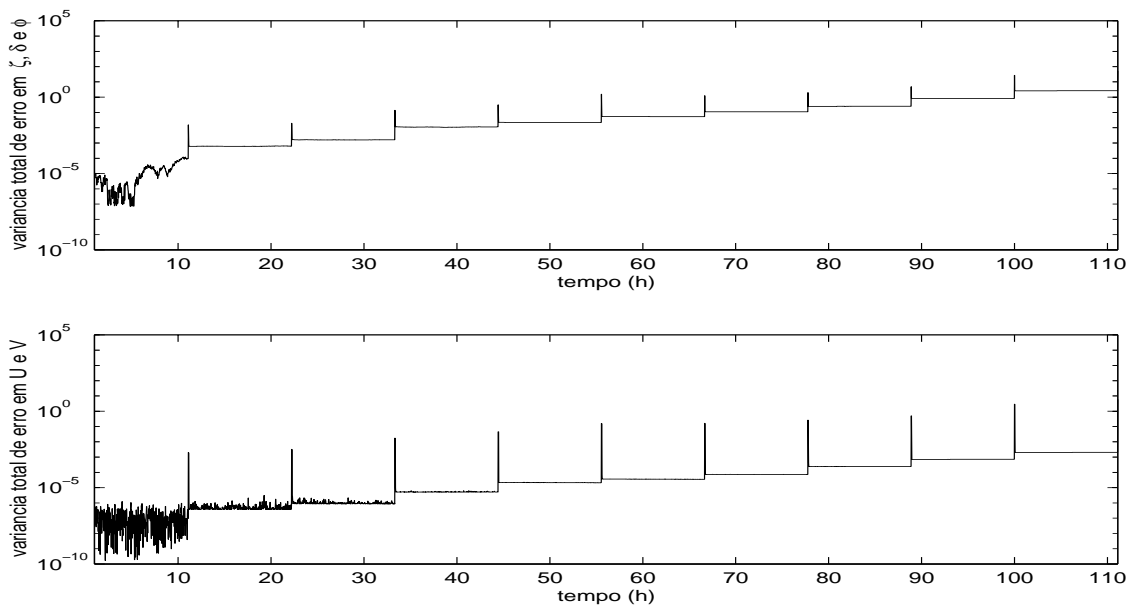


Fig. 6.8 - Variâncias totais de erro na assimilação usando Rede Neural com uma camada escondida, algoritmo de retropropagação adaptativo modificado e mudanças temporais nos parâmetros R_O , R_F e R_β do *senal verdadeiro*.

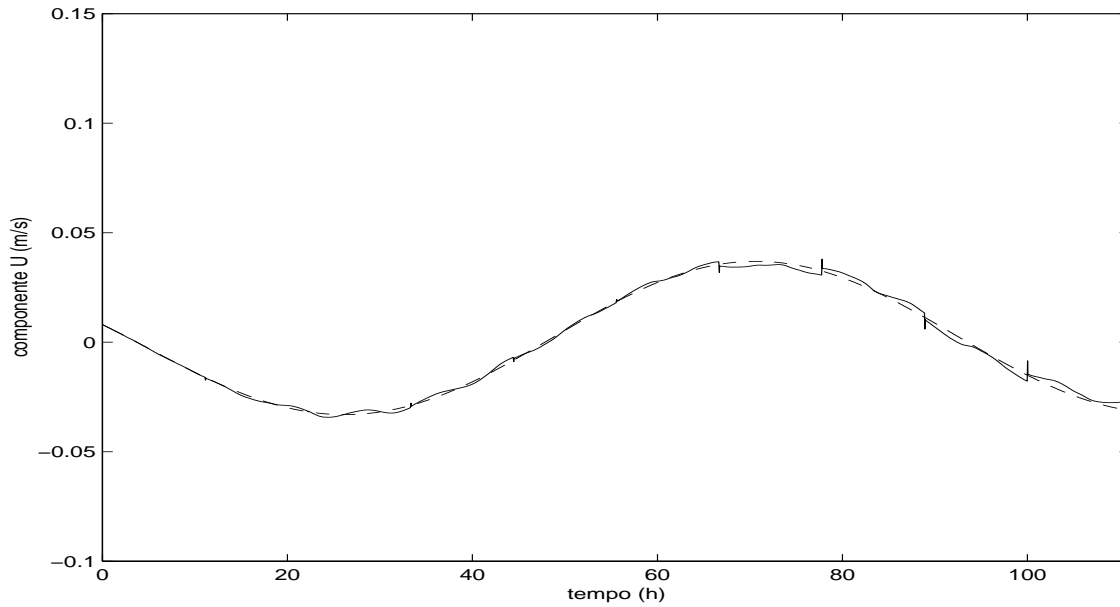


Fig. 6.9 - Evolução temporal de U na assimilação usando Rede Neural com duas camadas escondidas, algoritmo de retropropagação adaptativo e mudanças temporais nos parâmetros R_O , R_F and R_β do *signal verdadeiro*.

escondida teve problemas de descontinuidade em alguns instantes de inserção de dados, como pode ser visto na Figura 6.7. Após $t \geq 55 h$ houve problema significativo, mas somente após $t \geq 80 h$ ele começou a ficar sério. A Figura 6.8 mostra que *antes* da primeira assimilação houve erro de alta frequência significativo, que foi aparentemente eliminada pelas assimilações, apesar de que os problemas já vistos em 6.4 continuaram. A assimilação usando rede neural com duas camadas escondidas e algoritmo de retropropagação modificado também teve problemas de descontinuidade em alguns instantes de inserção de dados como visto nas Figuras 6.9 e 6.10. Novamente erro de alta frequência é visto antes da primeira assimilação e depois quase desaparece. De acordo com a Figura 6.9 para $20 \leq t \leq 45 h$ e então após $t \geq 65 h$ houve problema significativo, mas só para $t \geq 95 h$ ele se tornou sério. Comparando as Figuras 6.8 e 6.10 pode-se ver que o erro aumentou ao longo do tempo um pouco mais lentamente nessa rede que na anterior, como aconteceu nos experimentos 1 e 2. Mas em todos os casos os erros são muito menores que os da Figura 5.8.

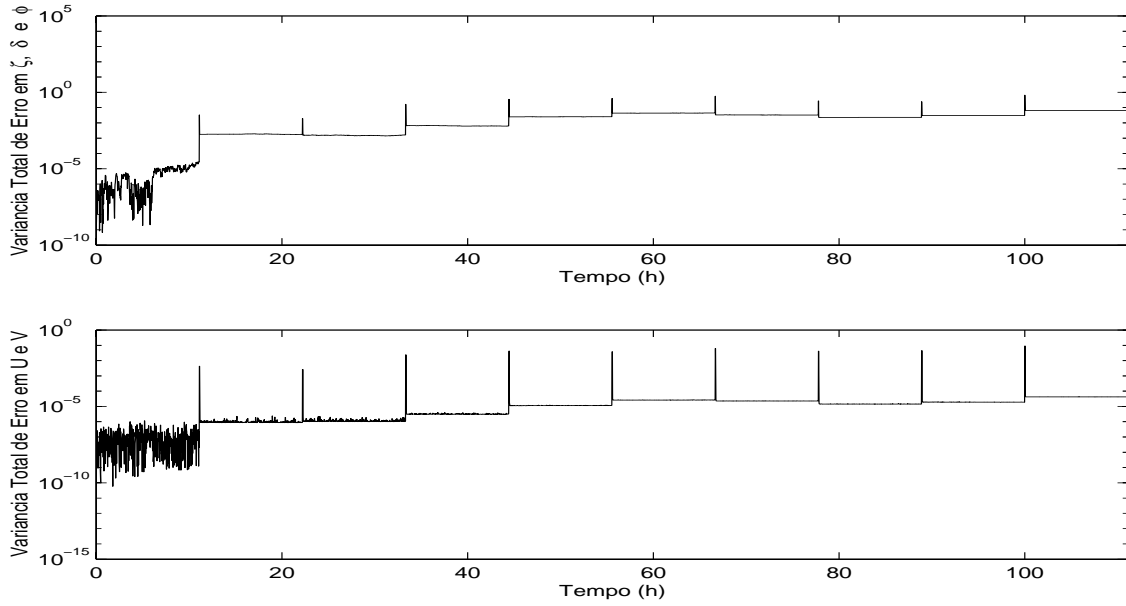


Fig. 6.10 - Variâncias totais de erro na assimilação usando Rede Neural com duas camadas escondidas, algoritmo de retropropagação adaptativo modificado e mudanças temporais nos parâmetros R_O , R_F , R_β do *signal verdadeiro*.

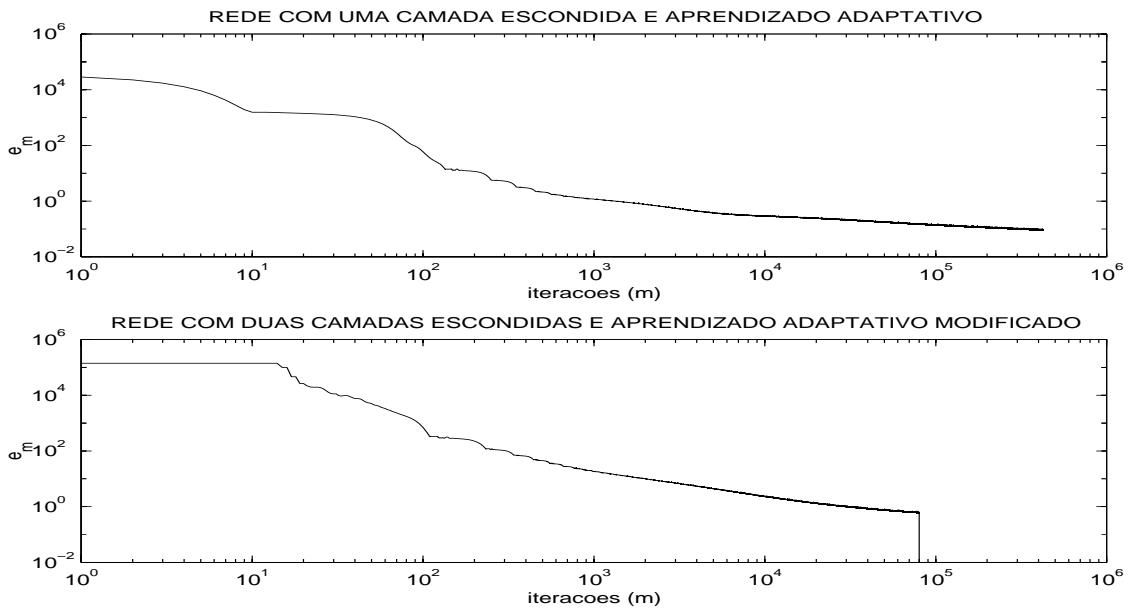


Fig. 6.11 - Evolução da soma de erros quadráticos (e_m) no treinamento das redes.

Para fechar esta seção, comparou-se o treinamento das duas redes mencionadas acima. Antes de analisá-lo deve-se ter em mente que o experimento-1 usou 100 exemplos enquanto que o experimento-2 usou 500 exemplos, e que em consequência cada iteração no experimento-2 custou consideravelmente mais tempo de processamento que no experimento-1. Verificou-se que o tempo de processamento dos dois experimentos foi similar. Nas Figuras 6.11 e 6.12 a evolução de e_m e η_m são mostrados. Todos os gráficos têm a mesma escala para o número de iterações. Nos gráficos inferiores (experimento 2) as iterações terminaram em $m = 79940$, e foram estendidas artificialmente para terem a mesma escala dos gráficos superiores. Na Figura 6.11 pode ser visto que o erro no experimento 2 caiu um pouco mais rápido que no experimento 1. O fato de que o treinamento no experimento 1 demorou mais que no experimento 2 não foi realmente conclusivo porque o alvo do experimento 1 era $e = 0.09$ enquanto que o do experimento 2 era $e = 0.6$. Em contraste, a Figura 6.12 mostra que a taxa de aprendizagem média η no experimento 1 foi maior que a no experimento 2. Confrontando este resultado com os outros resultados obtidos nos experimentos 1 e 2 sugeriria que taxa de aprendizado maior não implica necessariamente aprendizado melhor de uma função desejada.

6.4 Custo Computacional do Novo Método

O novo método para assimilação de dados foi testado treinando redes neurais para emular filtros de Kalman e comparando os resultados. Os resultados foram satisfatórios, mas para substituir filtragem de Kalman a nova abordagem precisa ser vantajosa em algum sentido. O objetivo desta seção é mostrar que fazer assimilação de dados usando uma rede neural pode ser mais barato que usando um filtro de Kalman.

Faça-se a hipótese de que uma rede com $m_0 = 2m_2$ entradas, L camadas escondidas de m_1 neurônios e uma camada de saída com m_2 neurônios tenha sido treinada para emular um Filtro de Kalman.

Em cada neurônio as m entradas alimentam a função de ativação através do produto

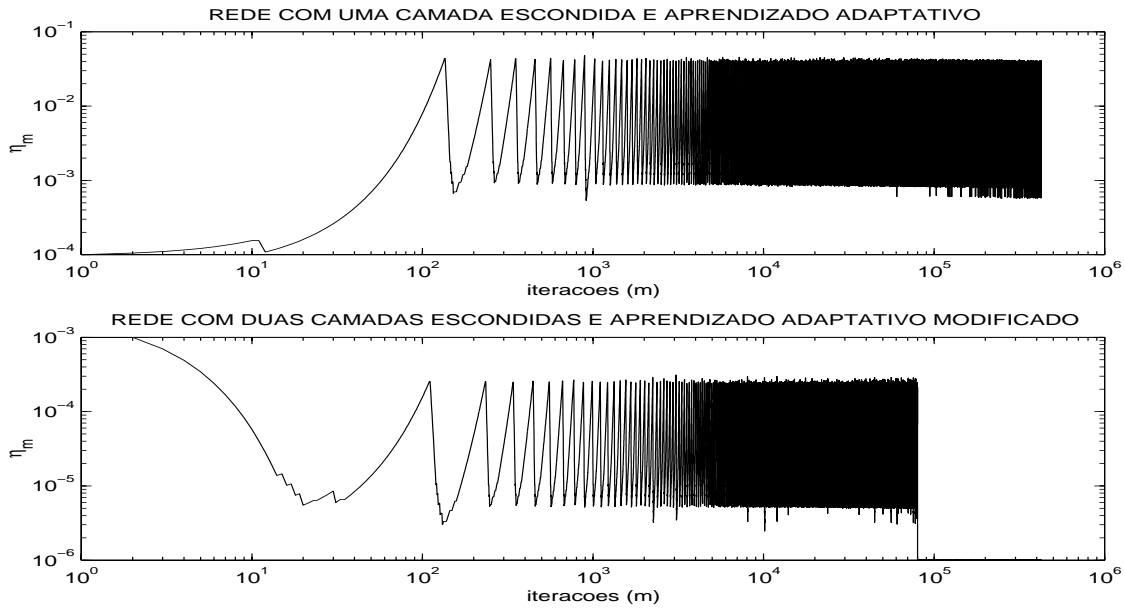


Fig. 6.12 - Evolução da taxa de aprendizado (η_m) no treinamento das redes.

interno

$$s = \sum_{i=1}^m \theta_i x_i \quad , \quad (6.28)$$

cuja complexidade é de ordem $O(m)$.

Qualquer que seja a função de ativação, suponha-se que seja computada usando uma série de Taylor truncada segundo o método da "multiplicação aninhada" (*nested multiplication*) [23]:

$$\begin{aligned} y = \psi(s) &= a_0 + a_1 s + a_2 s^2 + \dots + a_{n-1} s^{n-1} + a_n s^n \\ &= a_0 + s \left(a_1 s \left(a_2 + s^2 + \dots + s \left(a_{n-1} + a_n s \right) \right) \right) \quad . \end{aligned} \quad (6.29)$$

Este algoritmo envolve n multiplicações e n somas, sendo de ordem $O(n)$.

A função de assimilação implementada pela rede tem portanto complexidade de

ordem [54]

$$m_1 \cdot [O(m_0) + O(n)] + (L - 1) \cdot m_1 \cdot [O(m_1) + O(n)] + m_2 \cdot [O(m_1) + O(n)] \quad .(6.30)$$

Se $O(n) < O(m_1) = O(m_2)$ e $L \ll m_2$ então o algoritmo para calcular a saída da função de assimilação $w_n^a = F_{\text{ANN}}(w_n^f, w_n^o)$ terá complexidade $O(m_2^2)$.

Por outro lado, a complexidade de um Filtro de Kalman padrão com m_2 variáveis de estado e m_2 observáveis é $O(m_2^3)$ devido aos produtos matriciais a cada passo [41] (veja Figuras 2.4 and 2.5).

Pode-se concluir que quando o modelo matemático de um processo tem ordem de 10^3 ou mais variáveis de estado e requer matrizes cheias não-nulas \mathbf{F}_n e \mathbf{H}_n , uma rede neural com uma ou duas camadas treinada para desempenhar assimilação de Kalman deveria ter a possibilidade de fazê-la em tempo de computação significativamente menor. Além disso, pela própria natureza da arquitetura computacional paralela da rede, conforme salientado na seção 3.1, é possível que novos algoritmos de treinamento mais rápidos que os algoritmos usados neste trabalho sejam desenvolvidos [53]. Finalmente, esta natureza paralela torna a rede ideal para implementação *diretamente* em tecnologia de circuitos eletrônicos integrados em escala muito grande ('VLSI'), na forma de neurocomputador, e faz com que o sistema possa ser mais tolerante a falhas [22].

No caso em estudo nesta tese a filtragem de Kalman apresentou vantagens em precisão e confiabilidade. Isto porque, para o assunto em estudo nesta seção, as Equações de Água-Rasa, o modelo de observação e os parâmetros aqui escolhidos para modelar a atmosfera usaram 60 variáveis de estado e $\mathbf{F}_n = \mathbf{H}_n = \mathbf{I}$.

CAPÍTULO 7

CONCLUSÕES GERAIS

7.1 Contribuições

Esta tese apresenta três contribuições à Assimilação de Dados Meteorológicos:

- 1) Aplicação de Redes Neurais como um novo método de assimilação;
- 2) Aplicação de um Filtro de Kalman Adaptativo (algoritmo de Kuga e Rios-Neto) em assimilação;
- 3) Comparação entre métodos modernos de assimilação: Transformada de Laplace, Filtro de Kalman e Redes Neurais.

7.2 Conclusões

Neste trabalho foram testadas duas novas abordagens para assimilação de dados meteorológicos, um Filtro de Kalman Estendido Adaptativo e Redes Neurais de Perceptron Multicamadas. Na segunda abordagem usaram-se redes treinadas a partir de filtros de Kalman e testou-se uma modificação do algoritmo de aprendizado regra delta. Todos os resultados são resumidos a seguir.

Um Filtro de Kalman Estendido Adaptativo foi usado para assimilação de dados em três sistemas dinâmicos não-lineares: os sistemas de Hénon e de Lorenz em regime caótico e o modelo de água-rasa para a atmosfera DYNAMO. Esta abordagem não requereu que o erro de modelagem fosse estacionário e usou um filtro de Kalman linear para estimar este erro. Nos casos de Lorenz e DYNAMO este método foi comparado com os métodos usando Transformada de Laplace, Filtro de Kalman Linear e Estendido.

No caso do sistema de Lorenz o menor erro de assimilação ocorreu no Filtro de Kalman Estendido, mas o Filtro de Kalman Adaptativo se mostrou mais robusto a erro de modelagem.

No caso do DYNAMO, a Transformada de Laplace funcionou melhor para um modelo atmosférico com parâmetros invariantes no tempo. Os resultados também sugerem que o desempenho da Transformada de Laplace em sistemas dinâmicos fortemente não-lineares pode não ser confiável. O Filtro de Kalman Adaptativo teve desempenho satisfatório para modelos atmosférico com parâmetros variantes e invariantes no tempo. Além disso, o Filtro de Kalman Adaptativo mostrou-se uma alternativa válida ao Filtro de Kalman Estendido.

Perceptrons multicamadas foram treinados com variação de escala na normalização dos dados de entrada e saída para realizar assimilação de dados, aprendendo a emular filtros de Kalman. Os resultados foram então comparados com assimilação usando Transformada de Laplace e os próprios filtros de Kalman.

Primeiro, assimilação de dados num sistema de Lorenz caótico foi feita usando perceptron multicamadas com duas camadas escondidas, algoritmo de retropropagação com taxa de aprendizado constante e sem memória. Usaram-se 10% dos exemplos gerados por um filtro de Kalman Estendido. O desempenho da rede neural foi aceitável: a assimilação não foi perfeita, mas não ocorreu desvio caótico.

Segundo, foi feito teste de assimilação de dados no modelo DYNAMO. O perceptron multicamadas com uma camada escondida e treinado usando algoritmo de treinamento por retropropagação adaptativo desempenhou aceitavelmente. O perceptron multicamadas com duas camadas escondidas treinado usando algoritmo de retropropagação adaptativo modificado funcionou satisfatoriamente. "Picos" de erro apareceram nos instantes de assimilação usando redes neurais possivelmente devido à própria natureza dos experimentos, em que a perturbação era inserida exatamente na observação. Este mesmo problema apareceu nos outros experimentos feitos usando redes neurais. É possível que seja resolvido usando-se uma rede neural recursiva, a qual pode, em princípio, simular qualquer sistema dinâmico [22].

Quando o modelo atmosférico teve parâmetros variantes no tempo o desempenho dos perceptrons multicamadas foi aceitável, o perceptron com duas camadas escondidas

apresentou desempenho melhor.

O último resultado obtido nesta abordagem foi: quando o modelo matemático de um processo tem ordem de 10^3 ou mais variáveis de estado e requer matrizes cheias \mathbf{F}_n e \mathbf{H}_n , uma rede neural com uma ou duas camadas treinada para desempenhar assimilação de Kalman poderia fazê-la em tempo de computação significativamente menor.

Finalmente, deve-se salientar que Filtros de Kalman proveram os conjuntos de treinamento para as redes, mas outros métodos de assimilação podem ser também usados para treinar perceptrons multicamadas.

Aplicou-se um Filtro de Kalman Adaptativo que foi eficiente para promover a assimilação de dados. Redes Neurais são propostas como um método de assimilação de dados. Resultados preliminares aqui apresentados mostram que esta metodologia merece um estudo sistemático, visando sua aplicação em modelos operacionais de previsão numérica do tempo.

7.3 Trabalhos futuros

Uma variedade de trabalhos deverão dar continuidade a esta tese:

- 1) Outras funções de ativação serão usadas (já em desenvolvimento);
- 2) Outras topologias de rede serão usadas: redes com funções de base radial, correlação em cascata (já em desenvolvimento);
- 3) Os algoritmos serão paralelizados (já em desenvolvimento);
- 4) Aplicações deste filtro de Kalman adaptativo e redes neurais a modelos multidimensionais;
- 5) Redes recorrentes: introdução da memória no sistema, a principal alteração no futuro desta metodologia;
- 6) Implementação das redes neurais para assimilação de dados em hardware: neurocomputadores.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Anderson, B. D. O.; Moore, J. B. **Optimal filtering**. Englewood Cliffs, NJ: Prentice-Hall, 1979. 355 p.
- [2] Bengtsson, L.; Ghil, M.; Kälen, E. **Dynamic meteorology: data assimilation methods**. New York: Springer-Verlag, 1991. 330 p.
- [3] Braga, A. P.; Carvalho, A. C. P. L. F.; Ludermir, T. B. **Redes neurais artificiais: teoria e aplicações**. Rio de Janeiro: Editora Livros Técnicos e Científicos), 2000. 262 p.
- [4] Burg, J. P.; Luenberger, D. G.; Wenger, D. L. Estimation of structured covariance matrices. **Proceedings of the IEEE**, v. 70, n. 9, p. 963-974, September 1982.
- [5] Bürger, G.; Cane, M. A. Interactive kalman filtering. **Journal of Geophysical Research**, v. 99, n. C4, p. 8015-8031, April 1994.
- [6] Burgers, G.; van Leeuwen, P. J.; Evensen, G. Analysis scheme in the ensemble kalman filter. **Monthly Weather Review**, v. 126, p. 1719-1724, June 1998.
- [7] Campos Velho, H. F. **Matriz não-Modal em integração e inicialização num modelo barotrópico e um estudo numérico da dispersão vertical turbulenta**. Porto Alegre. 91 p. Dissertação (Doutorado em Engenharia Mecânica) - PROMEC-UFRGS, 1992.
- [8] Campos Velho, H. F.; Claeysen, J. C. R. A comprehensive analysis of a barotropic limited area model using the nonmodal matrix Technique. **Revista Brasileira de Meteorologia**, v. 12, n. 2, p. 41-50, December 1997.
- [9] Cohn, S. E., Parrish, D. The behaviour of forecast error covariances for a Kalman Filter in two dimensions. **Monthly Weather Review**, v. 119, n. 8, p. 1757-1785, August 1991.

- [10] Daley, R. **Atmospheric data analysis**. Cambridge: Cambridge University Press, 1991. 457 p.
- [11] Demuth, H.; Beale, M. **Neural network toolbox user's guide (for use with MATLAB)**. Cochituate Place, MA: The MathWorks, 1994. 700 p.
- [12] Dee, D. P.; Cohn, S. E.; Dalcher, Amnon, D.; Ghil, M. An efficient algorithm for estimating noise covariances in distributed systems. **IEEE Transactions on Automatic Control**, v. AC-3, n. 11, p. 1057-1065, November 1985.
- [13] Dee, D. P. On-line estimation of error covariance parameters for atmospheric data assimilation. **Monthly Weather Review**, v. 123, n. 4, p. 1128-1145, April 1995.
- [14] Dee, D. P.; da Silva, A. M. Maximum-likelihood estimation of forecast and error covariance parameters. Part I: methodology. **Monthly Weather Review**, v. 127, p. 1822-1834, April 1999.
- [15] Evensen, G. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. **Journal of Geophysical Research**, v. 99, n. C5, p. 10143-10162, May 1994.
- [16] Gardner, M. W.; Dorling, S. R. Artificial neural networks (the multilayer perceptron)-a review of applications in the Atmospheric Sciences. **Atmospheric Environment**, v. 32, n. 14/15, p. 2627-2636, June 1998.
- [17] Gauthier, P.; Charette, C.; Fillion, L.; Koclas, P.; Laroche, S. Implementation of a 3D Variational Data Assimilation System at the Canadian Meteorological Center. Part I: The Global Analysis. **Atmosphere-Ocean**, v. 37, n. 2, p. 103-156, June 1999.
- [18] Laroche, S.; Gauthier, P.; St-James, J.; Morneau, J. Implementation of a 3D variational data assimilation system at the Canadian Meteorological Center.

- Part II: the regional analysis. **Atmosphere-Ocean**, v. 37, n. 3, p. 281-307, September 1999.
- [19] Gelb, A. **Applied optimal estimation**. 4. ed.. Cambridge: The M.I.T. Press, 1978. 374 p.
- [20] Ghil, M.; Cohn, S.; Tavantzis, J.; Bube, K.; Isaacson, E. Applications of estimation theory to numerical weather prediction. In: Bengtsson, L.; Ghil, M.; Källén, E. ed. **Dynamic Meteorology: Data Assimilation Methods**. New York: Springer-Verlag, 1991. p. 139-224.
- [21] Harter, F. P. **Uso de filtro digital para iniciar um modelo de área limitada**. São José dos Campos. 111 p. (INPE-7251-TDI/695). Dissertação (Mestrado em Meteorologia) - Instituto Nacional de Pesquisas Espaciais, 1999.
- [22] Haykin, S. **Neural networks: a comprehensive foundation**. New York: Macmillan, 1994. 696 p.
- [23] Hoffman, J. D. **Numerical methods for engineers and scientists**. New York: McGraw-Hill, 1993. 825 p.
- [24] Jazwinski, A. H. Adaptive filtering. **Automatica**, v. 5, p. 475-485, 1969.
- [25] Jazwinski, A. H. **Stochastic processes and filtering theory**. New York: Academic Press, 1970. 376 p.
- [26] Jones, R. An experiment in non-linear prediction. **Journal of Applied Meteorology**, v. 128, p. 1971-1981, December 1965.
- [27] Keppene, C. L. Data assimilation into a primitive-equation model with a parallel ensemble Kalman filter. **Monthly Weather Review**, v. 4, p. 701-705, June 2000.
- [28] Kilcoyne, H. **Past and current methods of data assimilation at leading weather**

- and analysis centers.** [online]. < <http://www.datafront.com/paper> > .
November 1997.
- [29] Kuga, H. K. **Estimação adaptativa de órbitas aplicada a satélites a baixa altitude.** 128 p. (INPE-2316-TDL/079). Dissertação (Mestrado em Ciência Espacial) - Instituto Nacional de Pesquisas Espaciais, 1982.
- [30] Kuga, H. K. **Determinação de órbitas de satélites artificiais terrestres através de técnicas de estimação combinadas a técnicas de suavização de estado.** 249 p. (INPE-4959-TDL/388). Dissertação (Doutorado em Ciência Espacial) - Instituto Nacional de Pesquisas Espaciais, 1989.
- [31] Kurz, L. **Detection and estimation theory:** some remarks on the method of stochastic approximation: notas de aula do curso EL633 dadas na Polytechnic University of New York, New York, Primavera de 1988. 18 p.
- [32] Liebelt, P. B. **An Introduction to Optimal Estimation.** Reading: Addison-Wesley, 1967. 273 p.
- [33] Lemes, M. A. M.; Moura, A. D. **Fundamentos de dinâmica aplicados à meteorologia e oceanografia,** Instituto Nacional de Pesquisas Espaciais e Universidade do Vale do Paraíba, São José dos Campos, 1998. 484 p.
- [34] Lorenc, A. C. Analysis methods for numerical weather prediction. **Quarterly Journal of the Royal Meteorological Society**, v. 112, p. 1177-1194, October 1986.
- [35] Lorenz, E.N. Deterministic nonperiodic flow. **Journal of Atmospheric Sciences**, v. 20, p. 130-141, March 1963.
- [36] Lorenz, E.N. **The essence of chaos**, Washington: University of Washington Press, 1993. 278 p.

- [37] Lynch, P. **DYNAMO: a one-dimensional primitive equation model**. Irlanda: Irish Meteorological Service, 1984. (Irish Meteorological Service - Technical Note 44).
- [38] Lynch, P. Initialization using laplace transforms. **Quartely Journal of the Royal Meteorological Society**, v. 111, p. 243-258, January 1984.
- [39] Lynch, P. Initialization of a barotropic limited-area using the laplace transform technique. **Monthly Weather Review**, v. 113, p. 1338-1344, August 1984.
- [40] Lynch, P. **Numerical forecasting using laplace transforms: theory and application to data assimilation**. Irlanda: Irish Meteorological Service, 1986. (Irish Meteorological Service - Technical Note 48).
- [41] Mendel J. M. Computational requirements for a discrete Kalman filter. **IEEE Transactions on Automatic Control**, v. AC-16, n. 6, p. 748-758, December 1971.
- [42] Miller, R. N.; Ghil, M.; Gauthiez, F. Advanced Data Assimilation in Strongly Nonlinear Dynamical Systems. **Journal of the Atmospheric Sciences**, v. 51, n. 8, p. 1037-1056, April 1994.
- [43] Mitchell, H. L.; Houtekamer, P. L. Data assimilation using an ensemble Kalman filter technique. **Monthly Weather Review**, v. 126, p. 796-811, March 1998.
- [44] Mitchell, H. L.; Houtekamer, P. L. An ensemble adaptive Kalman filter. **Monthly Weather Review**, v. 128, p. 416-433, February 2000.
- [45] Nguyen, D.; Widrow, B. Improving the learning speed of 2-Layer neural networks by choosing initial values of the adaptive weights. In: **International Joint Conference on Neural Networks, San Diego, California, USA, 1990**. Ann Arbor, MI. IEEE Neural Networks Council, 1990. v. 3, p. 21-26.

- [46] Nowosad, A. G.; Rios Neto, A.; Campos Velho, H. F. Data assimilation using an adaptative Kalman filter and laplace transform. **Journal of Hybrid Methods in Engineering**, 2001. No prelo.
- [47] Nowosad, A. G.; Rios Neto, A.; Campos Velho, H. F. Data assimilation in chaotic dynamics using neural networks. In: International Conference on Nonlinear Dynamics, Chaos, Control and Their Applications in Engineering Sciences, 3., Campos do Jordão, SP, 2000. **Proceedings**. Rio de Janeiro: Associação Brasileira de Ciências Mecânicas, v. 6, p. 212-221.
- [48] Nowosad, A. G.; Campos Velho, H. F.; Rios Neto, A. Neural network as a new approach for data assimilation. [CD-ROM]. In: A meteorologia brasileira além do Ano 2000: Congresso Brasileiro de Meteorologia, 11. **Anais**, Campos do Jordão, SP, 2000. São Paulo: Microservice, 2000. p. 3078-3086.
- [49] Page, G. F.; Gomm, J. B.; Williams, D. **Application of neural networks to modelling and control**. Londres: Chapman and Hall. 119 p.
- [50] Parrish, D. F.; Derber, J. C. The National Meteorological Center's Spectral-Statistical Interpolation Analysis System. **Monthly Weather Review**, v. 120, n. 8, p. 1747-1763, August 1992.
- [51] Petersen, D. P. A comparison of the performance of quasi-optimal and conventional objective analysis schemes. **Journal of Applied Meteorology**, v. 7, p. 1093-1101, October 1973.
- [52] Rios Neto, A.; Kuga, H. K. Kalman filtering state noise adaptive estimation. In: IASTED International Conference on Telecommunication and Control , 2, Rio de Janeiro, 1985. **Proceedings**. São José dos Campos: INPE, 1985, p. 210-213.
- [53] Rios-Neto, A. Stochastic optimal linear parameter estimation and neural nets training in systems modelling. **Journal of the Brazilian Society of Mechanical Sciences**, v. 19, n. 2, p. 138-146, June 1997.

- [54] Terada, R. **Desenvolvimento de algoritmos e complexidade de computação**, Escola de Computação, 3., Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 1982. 231 p.
- [55] Todling, R. Suboptimal schemes for atmospheric data assimilation based on the Kalman filter. **Monthly Weather Review**, v. 122, p. 2530-2557, November 1994.
- [56] Todling, R. **Estimation theory and foundations of data assimilation**: notas de aula dadas no Laboratório Nacional de Computação Científica, Rio de Janeiro, Setembro 22-29 de 1997. 152 p.
- [57] Todling, R., Cohn, S.E.; Sivakumaran, N. Suboptimal schemes for retrospective data assimilation based on the fixed-lag Kalman smoother. **Monthly Weather Review**, v. 126, n. 8, p. 2274-2286, 1998.
- [58] Todling, R. Retrospective data assimilation schemes: fixed-lag smoothing. **Annals of the Institute of Statistical Mathematics**, n. 1, p. 1-18, 2001.
- [59] Thompson, J. M. T.; Stewart, H. B. **Nonlinear dynamics and chaos**: geometric methods for engineers and scientists. Chichester: John Wiley & Sons, 1991. 376 p.

APÊNDICE A

TEORIA DA ESTIMAÇÃO

Para melhor entender o Filtro de Kalman é interessante revisar teoria de estimação.

O problema de estimação geral pode ser formulado como se segue ([32] página 135).

Supomos que há uma relação funcional conhecida entre vetor observado \vec{z} e um vetor desconhecido \vec{w} :

$$\vec{z} = \vec{h}(\vec{w}) \quad . \quad (\text{A.1})$$

O problema de estimação não-linear consiste em estimar \vec{w} de alguma maneira baseado no conjunto das observações \vec{z} .

Técnicas Bayesianas para Resolver o Problema

Neste ponto é importante estabelecer a notação. A função $f_{\vec{w}}(\vec{w})$ denota a densidade de probabilidade da variável aleatória \vec{w} . O valor esperado, valor médio ou média de uma variável aleatória \vec{w} será denotado por $E\{\vec{w}\}$.

No caso em que \vec{w} é um parâmetro aleatório, que representa o vetor \vec{w} que se deseja estimar, pode-se usar a *Estimação Bayesiana*. No que se segue, seja $\hat{\vec{w}}$ uma estimativa de \vec{w} , e \vec{z} a observação. Então o erro na estimativa é $\vec{e} = \hat{\vec{w}} - \vec{w}$.

Neste tipo de estimação, conforme Kurz [31], supõe-se que \vec{w} seja variável aleatória e que conhecem-se

- A) $f_w(w)$;
- B) $f_{z|w}(z|w)$;
- C) Função Risco $J(\hat{\vec{w}} - \vec{w}) = J(\vec{e})$.

Conhecendo estas grandezas, minimiza-se o risco médio:

$$\vec{w}^a(\vec{z}) = \min_{\hat{\vec{w}}} E \left\{ J(\hat{\vec{w}} - \vec{w}) \right\} \quad (\text{A.2})$$

Casos Especiais Importantes:

1) **Estimador com Variância Mínima** de Erro

O estimador é:

$$\vec{w}^a(\vec{z}) = \vec{w}_{VM}(\vec{z}) = E \{ \vec{w} | \vec{z} \} \quad . \quad (\text{A.3})$$

1.1') **Estimador Linear com Variância Mínima** de Erro

$$\vec{w}^a(\vec{z}) = \vec{w}_{VML}(\vec{z}) = \mathbf{L}\vec{z} \quad . \quad (\text{A.4})$$

2) **Estimador com Probabilidade Posterior Máxima (MAP)**

Supõe-se que $J(e)$ seja uniforme (os erros não são ponderados):

$$\vec{w}^a(\vec{z}) = \vec{w}_{PM}(\vec{z}) = \max_{\hat{w}} f_{w|z}(w|z)(\vec{z}) \quad . \quad (\text{A.5})$$

3) **Estimação de Máxima Verossimilhança (MaxVer)**

Supõe-se que $f_{z|w}(z|w)$ seja conhecida. É equivalente à Estimação Bayesiana sem as suposições A) e C):

$$\vec{w}^a(\vec{z}) = \vec{w}_{MV}(\vec{z}) = \max_{\hat{w}} f_{z|w}(z|w)(\vec{z}) \quad (\text{A.6})$$

O Filtro de Kalman pode ser deduzido usando-se a técnica (1.1') para estimar os valores \vec{w}_n a partir das observações \vec{z}_n .