



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

# **Estudo de Redes Neurais Artificiais na Identificação de Alvos Móveis com Auxílio do Filtro de Kalman**

**RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA  
(PIBIC/CNPq/INPE)**

**Bolsista: Cassiano Rodrigo Silva (FEG/UNESP)  
ele04335@feg.unesp.br**

**Orientador: José Ernesto de Araújo Filho (LIT/INPE)  
ernesto@lit.inpe.br**

**Junho de 2004**

# ÍNDICE

<b>ÍNDICE DAS FIGURAS .....</b>	<b>3</b>
<b>1 – INTRODUÇÃO.....</b>	<b>4</b>
1.1 - OBJETIVO DO TRABALHO .....	5
1.2 – DESENVOLVIMENTO REALIZADO.....	6
<b>2 - HISTÓRICO DA IDENTIFICAÇÃO DE ALVOS MÓVEIS .....</b>	<b>8</b>
<b>3 - REDES NEURAIAS KOHONEN E FILTRO DE KALMAN NA IDENTIFICAÇÃO DE ALVOS MÓVEIS ...</b>	<b>11</b>
3.1 – APLICANDO AS REDES NEURAIAS KOHONEN E O FILTRO DE KALMAN.....	13
PREDIÇÃO OU PROPAGAÇÃO NO TEMPO .....	13
ATUALIZAÇÃO OU CORREÇÃO DA ESTIMATIVA .....	14
3.2 –EXEMPLOS DA VALIDAÇÃO DO ALGORITMO.....	16
<b>4 – ANÁLISE DOS RESULTADOS.....</b>	<b>20</b>
4.1 – VALIDAÇÃO E RESULTADO DO ALGORITMO EM PONTOS.....	20
4.1.1 – Rastreamento do Objeto Sobre a Reta.....	20
4.1.2 – Rastreamento do Objeto Sobre o Logaritmo.....	22
4.1.3 – Rastreamento do Objeto Sobre o Seno.....	22
4.1.4 – Rastreamento do Objeto com Mais de Uma Trajetória .....	23
4.2 – VALIDAÇÃO E RESULTADOS DO ALGORITMO EM SEQUÊNCIAS DE IMAGENS .....	26
<b>5 – CONCLUSÃO E SUGESTÃO PARA TRABALHOS FUTUROS.....</b>	<b>29</b>
5.1 – CONCLUSÃO .....	29
5.2 – SUGESTÕES PARA TRABALHOS FUTUROS .....	30
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>32</b>

## ÍNDICE DAS FIGURAS

Figura 1- Rastreamento com Falha no Treinamento.....	6
Figura 2 - - Falha no Rastreamento com Imagem.....	7
Figura 3- Fluxograma do algoritmo de Kohonen e Filtro de Kalman.....	15
Figura 4- Nuvem de pontos e posições iniciais dos neurônios na Reta.-.....	16
Figura 5 - Nuvem de pontos e posições iniciais dos neurônios no Logaritmo.....	17
Figura 6- Nuvem de pontos e posições iniciais dos neurônios no Seno.....	17
Figura 7- Seqüência de imagens mostrando a trajetória do avião.....	18
Figura 8- Resultado do Algoritmo com Trajetória Retilínea.....	21
Figura 9- Resultado do Algoritmo com Trajetória Logarítmica.....	22
Figura 10- Resultado do Algoritmo com Trajetória Senoidal.....	23
Figura 11 Dois Objetos com Trajetórias Diferentes.....	24
Figura 12- Rastreamento mal sucedido de dois objetos.....	25
Figura 13- Três Objetos com Trajetórias Diferentes.....	25
Figura 14- Rastreamento mal sucedido de três objetos.....	26
Figura 15 - Rastreamento da Imagem com Avião.....	27
Figura 16- - Rastreamento do avião na seqüência de imagens.....	28

# 1 – Introdução

A identificação de alvos móveis utilizando Redes Neurais Artificiais e o Filtro de *Kalman* é apresentada neste trabalho. Para a identificação da trajetória do objeto-alvo foram utilizadas redes neurais artificiais do tipo *Kohonen*, enquanto o filtro de Kalman prediz o comportamento futuro no movimento.

Redes neurais artificiais são técnicas computacionais que utilizam como modelo, o modelo de funcionamento do cérebro humano, em que neurônios são treinados de forma a responder aos padrões de entrada, podendo ser utilizadas no reconhecimento de padrões. As redes neurais, atualmente, estão sendo utilizadas para solucionar diversos problemas de engenharia, como em processamento de imagens e robótica, sendo de grande interesse em diversas áreas da engenharia. As principais características que tornam desejáveis o estudo e o uso das redes neurais artificiais são a capacidade das mesmas de aprender através da experiência, a capacidade de se adaptar a situações adversas e a sua tolerância a ruídos. No campo das redes neurais existem diferentes técnicas de treinamento dos neurônios, mas neste trabalho foram utilizadas técnicas de aprendizado não supervisionado, onde os neurônios não necessitam que o usuário forneça uma saída para a rede.

Conhecida também como Camada de *Kohonen*, nesta rede os neurônios disputam entre si pelo treinamento de suas sinapses, apresentando desta forma um aprendizado competitivo. Esta característica competitiva fornece à rede neural também a classificação de redes neurais do tipo *winner-takes-all*. Nas redes Kohonen o neurônio vencedor é treinado e tem sua sinapse alterada, aproximando, a cada passo de treinamento, seus valores dos padrões de entrada. Estas técnicas são utilizadas na identificação do objeto-alvo em seqüências de imagens, e na identificação do movimento de forma que os neurônios sejam treinados para se aproximarem (rastream) dos objetos-alvo, que neste caso são os padrões de entrada. Neste trabalho, os neurônios perdedores também recebem um treinamento. Os neurônios perdedores têm seu peso alterado de forma contrária do neurônio vencedor, de forma a se afastarem do objeto-alvo, reduzindo assim, a chance de atrapalharem o neurônio rastreador.

Visto que as redes neurais Kohonen, geram suas próprias saída a partir dos padrões de entrada, elas funcionam como um sistema dinâmico de tal forma que a entrada (p.ex., padrões, alvos etc) e a saída (sinapses treinadas) do sistema estão disponíveis. Assim, o filtro de Kalman é utilizado para estimar a saída da rede em um instante posterior àquele em que há o treinamento da rede Kohonen. Em casos em que o neurônio vencedor é alterado, ou seja, há uma falha no rastreamento, o filtro de Kalman é utilizado para estimar a posição do alvo, e corrigir o erro da rede neural.

O Filtro de Kalman é um estimador de estados de sistemas dinâmicos, baseando-se nos estados

anteriores do sistema e na teoria da probabilidade. Através de um conjunto de equações, o filtro estima os estados futuros do sistema. Basicamente, o Filtro de Kalman realiza o processo de estimação em duas fases: Predição, ou propagação no tempo, e Atualização, ou correção da estimativa. Na fase de predição, o filtro utiliza os estados anteriores do sistema dinâmico para propagar no tempo uma estimativa *a priori* do estado futuro e uma matriz de covariância de erro. Já na fase de atualização, a saída do sistema se torna disponível e o filtro a utiliza como parâmetro para corrigir a predição dos estados e a matriz de covariância de erro, gerando uma estimação *a posteriori*, ou corrigida dos estados do sistema. O filtro realiza este processo diversas vezes, reduzindo, a cada passo, o erro de estimativa. O Filtro de Kalman tem sido utilizado como rastreador em diversas áreas da engenharia. Todavia, neste trabalho o Filtro de Kalman é uma “ferramenta complementar” na qual irá auxiliar no rastreamento realizado pela rede neural.

### 1.1 - Objetivo do Trabalho

Neste trabalho foram utilizadas técnicas de Redes Neurais Artificiais como ferramenta para realizar o rastreamento de alvos móveis com o objetivo de analisar o comportamento da mesma. Para tal tarefa foram utilizadas redes neurais do tipo *Kohonen* que, com aprendizado não supervisionado, é treinada quando os padrões de entrada são fornecidos à entrada da rede. A vantagem das redes neurais *Kohonen* é que a mesma não necessita de um treinamento prévio, podendo ser treinada quando os padrões aparecem na entrada da rede, realizando assim um rastreamento em tempo real.

Foram pesquisados em livros e artigos publicados por pesquisadores da área, para formar uma base de conhecimento sobre redes neurais. Após o estudo sobre redes neurais e a implementação de alguns algoritmos, foi realizada uma pesquisa sobre o Filtro de Kalman. Este também foi pesquisado em livros e artigos. Para tal pesquisa, foram utilizadas as seguintes bibliotecas: (1) Instituto Nacional de Pesquisas Espaciais; (2) Departamento da Universidade Estadual Paulista – Campus Guaratinguetá. Os artigos e periódicos foram pesquisados na internet, como as bases de dados: (1) Web of Science – [www.webofscience.com](http://www.webofscience.com), e (2) Capes – [www.periodicos.capes.gov.br](http://www.periodicos.capes.gov.br). Em sites de busca: (1) yahoo (Brasil) – [www.yahoo.com.br](http://www.yahoo.com.br), e robôs de busca: (1) radix – [www.radix.com.br](http://www.radix.com.br), (2) uol – [www.radaruol.com.br](http://www.radaruol.com.br), e (3) google – [www.google.com](http://www.google.com); o robô (Metamáquina) de busca [www.metaminer.com](http://www.metaminer.com); (4) scirus – [www.scirus.com](http://www.scirus.com); e (5) Web Of Science – [www.webofscience.com](http://www.webofscience.com).

## 1.2 – Desenvolvimento Realizado

Na primeira fase do projeto foram implementados algoritmos de redes neurais para realizar o rastreamento de alvos utilizando o software Matlab. Foram utilizados pontos para representar o objeto. Inicialmente, a validação se deu com pontos estáticos e a rede neural foi treinada de forma a se aproximar do ponto, identificando os padrões. Após esta implementação, foram simuladas trajetórias para este ponto, descritas por funções matemáticas na seguinte seqüência: (1) Reta; (2) Função Logarítmica e (3) Função Senoidal. Em todos os casos foi observado que a rede neural realizou o rastreamento com precisão, conseguindo identificar a forma da trajetória do objeto-alvo. Foram simuladas, também, movimentos com mais de um objeto, descritos com funções diferentes, de forma a se cruzarem. Foi observado que os neurônios realizavam o rastreamento com algumas falhas. A Figura 1 ilustra esta situação. Estas falhas se davam quando as trajetórias dos objetos se cruzavam, o que confundia os neurônios rastreadores.

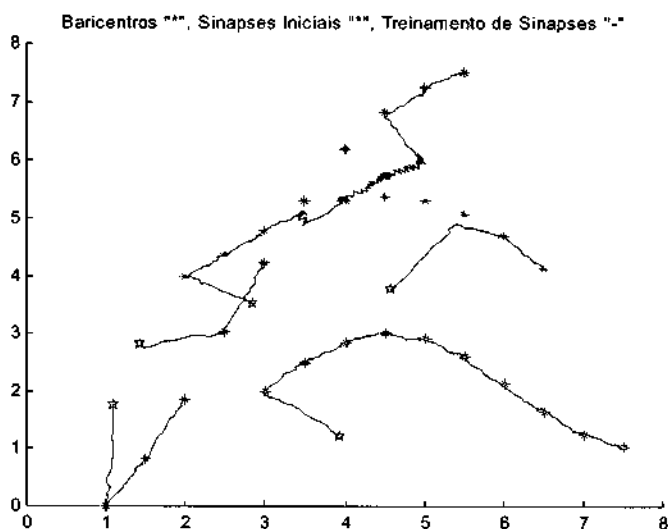
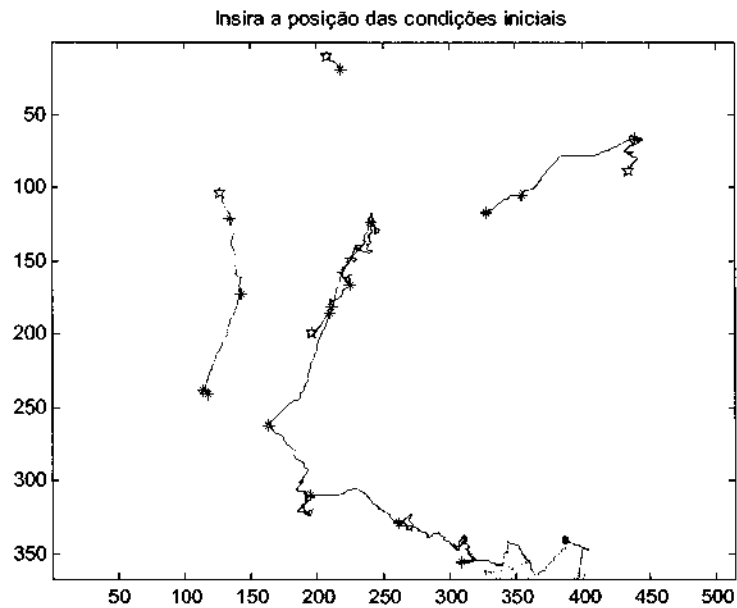


Figura 1- Rastreamento com Falha no Treinamento

O algoritmo foi validado também com seqüência de imagens digitais. Foi observado que o rastreamento foi bom, de forma que a rede neural identificava a trajetória do objeto. Através dos testes foi possível analisar a importância, para o sucesso do treinamento, das condições ou posições iniciais dos neurônios, da quantidade de neurônios a serem treinados e da posição dos alvos a serem identificados. Através dos testes com pontos móveis e seqüência de imagens, pode-se também analisar a influência da mudança de posição e do cruzamento dos alvos durante o rastreamento. Foram identificadas falhas no algoritmo no caso de mais de um neurônio estar próximo a um alvo, ou no caso em que mais de um alvo se aproximam e se cruzam, gerando confusão entre os

neurônios rastreadores. Esta situação é mostrada na Figura 2.



**Figura 2 -- Falha no Rastreamento com Imagem**

Na segunda fase do projeto foi implementado o Filtro de Kalman na rede neural com a intenção de estimar a posição do neurônio rastreador e identificar a forma do rastreamento. Pelos testes realizados, pode-se observar que o Filtro de Kalman realizou uma estimativa satisfatória da posição da rede neural, identificando seu comportamento dinâmico.

Este trabalho é, na verdade, uma continuação de um trabalho de iniciação científica [42] que utiliza redes neurais *Kohonen* para realizar o rastreamento. O incremento aqui colocado é o Filtro de Kalman, com a intenção de melhorar o rastreamento nas falhas identificadas.

## 2 - Histórico da Identificação de Alvos Móveis

A identificação, ou rastreamento de alvos móveis, tem sido de grande interesse de pesquisadores para aplicação em diversas áreas, como por exemplo, em sistemas de vigilância utilizados para segurança de algum local ou utilização de radar. O fato de o rastreamento ser em tempo real, requer um algoritmo rápido e robusto. Nos rastreamentos em tempo real, o algoritmo deve realizar o rastreamento em cada instante de tempo, ou cada imagem retirada da seqüência de imagens.

Atualmente, vários pesquisadores têm utilizado técnicas de inteligência artificial combinadas às métodos de processamento digital de imagens para o reconhecimento de padrões e rastreamento de alvos em seqüência imagens. Entre as técnicas de inteligência artificial utilizadas, destacamos: redes neurais artificiais, lógica difusa, sistemas baseados em regras e sistemas híbridos. O objetivo, neste caso, é gerar algoritmos que possam identificar a trajetória do alvo. Wu e Lin [2] utilizaram redes neurais artificiais em conjunto com sistemas nebulosos como um método de rastreio num campo aproximado de um alvo em movimento. A rede neural do tipo *feedforward* foi empregada por Schram, VanderLinden, Krose e Groen [3] na identificação de alvos estacionários e móveis para o controle de robôs através do uso de derivadas no tempo da posição do alvo e do manipulador. Gaudiano, Zalama e Coronado [4] utilizam uma rede neural como controlador de um robô móvel. Esse controlador aprende de forma autônoma a indicar as distâncias percorridas de forma direta e inversa pelo robô através de um ciclo de treinamento não-supervisionado. Como o aprendizado é constante e não-supervisionado, perturbações, ou ruídos, significativas (como a mudança da planta do robô) levam a uma recalibração interna gradual de forma automática do controle do robô. Nas simulações realizadas, o robô foi capaz de rastrear alvos estacionários e móveis de forma correta dentro do erro mínimo estabelecido, graças à tolerância a ruídos da rede neural. Um sistema de composição de dados para uma câmera situada no braço do robô SCORBOT-ER VII foi apresentado por Chen e Hsueh [5]. O sistema é capaz de realizar o rastreamento visual e a interceptação de alvos móveis utilizando uma rede neural Elman. A rede neural Elman é usualmente uma rede neural de duas camadas (a camada de entrada e a camada de saída, não possuindo a camada escondida) com um retorno de realimentação (feedback) da camada de saída para a camada de entrada. Nesta abordagem, o valor da saída é armazenado para ser processado no passo seguinte juntamente com a próxima entrada. Assim, se duas redes Elman idênticas (com os mesmos valores para os pesos) receberem uma mesma entrada em um dado passo, suas saídas poderão ser diferentes devido a estados de retorno diferentes. Ma e Teng [6] exploram o problema do rastreamento de um alvo móvel em um campo aproximado usando redes neurais nebulosas. O alvo móvel irradia ondas de banda curta que incidem sobre uma série de sensores passivos. A posição do alvo é encontrada



através de vários estimadores de ângulo e distância. Yu e Azimi-Sadjadi [7] desenvolveram uma rede neural para a classificação de alvos explorando o comportamento dinâmico do alvo. O sistema consiste de um extrator de características, uma rede neural geradora de probabilidades condicionais diretas e um novo classificador Bayes seqüencial. Yang e Meng [8] propõem uma rede neural artificial para a geração de trajetórias em um ambiente não-estacionário. A rede é usada para gerar trajetórias livres de colisões, evitar obstáculos em forma de U e rastrear alvos móveis em ambientes com obstáculos. Cameron, Grossberg e Guenther [9] utilizam uma RNA que recebe como entrada uma imagem e a posição do ponto de onde se está visualizando a imagem. Como saída a rede fornece a representação da direção, da profundidade do cenário e da localização de objetos móveis. Essas representações são usadas para a navegação em simulações envolvendo o desvio de obstáculos e o rastreamento de um alvo móvel. Existem abordagens para a detecção e rastreamento de alvos inspirados em técnicas de rastreamento encontrados nos sistemas visuais de organismos vivos. Missler e Kamangar [10] apresentam uma rede neural artificial que detecta e rastreia um objeto se movendo dentro de seu campo de visão. O trabalho é inspirado em funções de processamento observadas no sistema visual das moscas. A rede é capaz de detectar mudanças de intensidade de luz, determinar a movimentação de um alvo e fornecer as informações necessárias para realizar o rastreamento. Dror, Florer, Rios e Zagaeski [11] realizaram estudos sobre a utilização de uma rede neural para reconhecimento de padrões que recebe dados em um formato similar ao do sonar de um morcego. A rede foi utilizada com sucesso no reconhecimento de rostos e para determinar a velocidade de um alvo que se movia em direção à câmera. Navabi e Agarwal [12] apresentam um modelo artificial de organização de resposta neural de retina que pode servir como um quadro para a análise da resposta dinâmica em redes neurais artificiais, especialmente nos sistemas visuais. Yi Li, Songde Ma, Hanqing Lu [14], desenvolveram um sistema de reconhecimento de postura do corpo humano utilizando métodos morfológicos e o Filtro de Kalman. A filtragem de um sinal com ruídos, medido da saída de um conversor A/D não muito preciso, com ruídos internos foi realizada por Pereira, Guilherme A. S.[15] através da filtragem linear utilizando para o mesmo o Filtro de Kalman. Wann e Thomopoulos [22] utilizam uma rede neural artificial de aprendizado não-supervisionado no desenvolvimento de um sistema de composição de dados. Os dados são aplicados em problemas de detecção de alvos em um sistema multicanal de indicação de alvos móveis. As características dos dados recebidos de três canais de radares diferentes são extraídas através de técnicas de processamento de sinais digitais. Shams [30] utiliza redes neurais no rastreamento de alvos utilizando sensores passivos. Cada sensor fornece o ângulo formado pela linha estimada entre o alvo e o sensor. A partir dos dados de todos os sensores, a rede neural é capaz de estimar a posição dos alvos. A rede neural proposta é uma rede auto-associativa chamada de ‘módulos elásticos múltiplos’.

Como mostrado anteriormente, vários rastreadores têm sido desenvolvidos utilizando técnicas de inteligência artificial isoladamente, ou combinando-as. Este trabalho possui uma abordagem diferente. Aqui são empregadas duas ferramentas: Redes Neurais Artificiais do tipo Kohonen e o Filtro de Kalman.

### 3 - REDES NEURAIS KOHONEN E FILTRO DE KALMAN NA IDENTIFICAÇÃO DE ALVOS MÓVEIS

Esta parte do projeto destina-se à implementação de um algoritmo baseado nas técnicas de redes neurais artificiais do tipo *Kohonen*, e em técnicas de filtragem linear, o Filtro de Kalman, para identificar e rastrear a trajetória de um alvo qualquer que pode ser fixo ou móvel.

As Redes Neurais *Kohonen* são um tipo de rede neural, chamadas de redes de aprendizado não supervisionado, que identificam e agrupam os padrões de entrada a partir de sua proximidade geométrica, ou seja, o quão próximos estão os padrões de entrada dos pesos dos neurônios. A vantagem que as redes *Kohonen* nos oferece é que a mesma não necessita que seja fornecida a saída esperada para um dado padrão de entrada. Apenas com a apresentação dos dados de entrada à rede, esta gera sua própria saída e treina os neurônios alterando seus pesos de forma a agrupar os dados de entrada que são similares. As redes *Kohonen* são, muitas vezes, chamadas de redes neurais de aprendizado competitivo, ou sem professor, *winner-takes-all*. Este nome se dá pelo fato de que, quando os padrões de entrada são apresentados à rede, os neurônios disputam entre si pelo treinamento de seus pesos, ou seja, é realizada uma disputa em toda a rede para determinar qual(is) neurônio(s) terá(ão) seus pesos alterados de forma a se aproximar dos dados de entrada. Uma rede neural do tipo *Back-Propagation*, por exemplo, necessita que informemos os padrões de entrada e a saída desejada à rede. Com esta característica das redes neurais *Kohonen*, precisamos apenas informar à rede os padrões de entrada, e a mesma realiza processo de determinação do neurônio que terá o peso alterado, e treina o mesmo para agrupar os padrões de entrada. A cada padrão de entrada apresentado à rede, apenas um neurônio é treinado para agrupá-los, e este neurônio é denominado neurônio vencedor. O neurônio vencedor é o que está mais próximo dos padrões de entrada, portanto, é o mais apto a ser treinado para o dado agrupamento.

O Filtro de Kalman é um filtro que realiza a estimação dos estados futuros de um sistema dinâmico. Técnicas de filtragem linear têm sido muito utilizadas por pesquisadores de vários países, sendo o Filtro de Kalman uma das mais conhecidas. A função do Filtro de Kalman é, a partir de estados anteriores, gerar uma estimativa dos estados futuros de um sistema dinâmico. Basicamente, este filtro realiza o processo de estimação em duas fases: (1) Predição, ou propagação no tempo, e (2) Atualização, ou correção dos estados a partir da medida atual disponível. Uma característica interessante deste filtro é que o mesmo suporta ruídos branco Gaussiano, ou seja, de média zero. Estes ruídos são originados de dispositivos de medição, como um sensor, por exemplo. O que precisamos informar ao filtro são os estados anteriores do sistema, a covariância inicial de erro dos estados e os ruídos de processo e de medição. Todos estes parâmetros são armazenados em matrizes

e vetores para que sejam manipulados ao mesmo tempo. Este filtro baseia-se na Teoria da Probabilidade, e possui equações que realizam a estimação dos estados de um sistema dinâmico, e reduz a covariância de erro de estimação. Na fase de predição, o filtro realiza a propagação no tempo da estimativa dos estados do sistema e da matriz covariância de erro. Neste momento, possuímos uma estimativa dos estados do sistema com um certo erro. Já na fase de atualização, as medidas corretas dos estados do sistema se tornam disponíveis, com os devidos ruídos de medição. Tendo isto disponível, o filtro corrige os estados estimados e a covariância de erro baseando-se nas medidas disponíveis. O filtro realiza este processo várias vezes, reduzindo a covariância de erro e estimando um valor cada vez mais próximo dos estados do sistema.

Observando as características descritas acima, podemos analisar como estas duas ferramentas podem ser combinadas de forma a realizar um rastreamento. Para um dado padrão de entrada, a rede neural *Kohonen* realiza a competição entre os neurônios e determina qual será o neurônio vencedor. Feito isto, esta treina o neurônio para cada dado nos padrão de entrada, alterando os pesos dos neurônios de forma a aproximá-los dos padrões de entrada. Com este processo, os neurônios são aproximados dos dados de entrada. Se este padrão de entrada representar um objeto-alvo no espaço, os neurônios irão se aproximar deste objeto. No caso deste objeto ser móvel, ou seja, alterar sua posição no espaço, e conseqüentemente os valores dos padrões de entrada, o neurônio vencedor terá seu peso treinado novamente de forma a se aproximar deste novo padrão de entrada. Se tivermos  $N$  mudanças no padrão de entrada, o neurônio será treinado para cada uma das  $N$  mudanças, realizando assim, um rastreamento, ou seja, o mesmo irá identificar a trajetória do objeto no espaço. Quando existirem mais de um padrão de entrada, mais de um neurônio será treinado, sendo cada neurônio pertencente a um destes. No caso destes objetos se cruzarem em suas trajetórias, pode ocorrer uma confusão entre os neurônios, resultando em uma troca de rastreamento, ou seja, os neurônios mudam de objeto-alvo, passando a rastrear o outro objeto que se aproximou. Uma possível solução para este problema está no Filtro de Kalman. Sendo a rede neural um sistema dinâmico, seus estados (posição  $X$  e  $Y$  do neurônio no futuro passo de treinamento) podem ser estimados pelo Filtro de Kalman. Na primeira fase do filtro, uma estimativa dos estados, ou posição futura da rede neural é gerada, baseada na posição anterior da mesma e é chamada de estimativa *a priori*. Quando o neurônio é treinado, temos disponível sua posição, e o filtro utiliza a mesma para corrigir os estados estimados e a covariância de erro da estimativa. Estes estados corrigidos são denominados "estimativas" *a posteriori*. Deste modo, o Filtro de Kalman nos indica a posição atual da rede neural. Sendo assim, mesmo que os neurônios rastreadores de dois objetos que cruzam as trajetórias se enganem, o filtro identifica a trajetória do objeto. Esta é a forma como um rastreamento é realizado quando combinadas às ferramentas: Redes Neurais Kohonen e Filtro de Kalman.

### 3.1 – Aplicando as Redes Neurais Kohonen e o Filtro de Kalman

Como dito anteriormente, as redes neurais *Kohonen* realizam um agrupamento dos padrões de entrada em classes, sendo assim, denominada uma rede Classificadora. Quando um padrão de entrada é apresentado à rede, esta entrada  $x^k$  terá uma saída  $y^k$  da rede neural, sendo que  $y_i^k \in \{-1, +1\}$ , e para essa saída, somente um neurônio está ativo, ou seja, está recebendo treinamento, então: ( $y_j^k = +1$ ). Os demais neurônios são treinados de forma a se afastarem do neurônio rastreador. O  $j$ -ésimo neurônio ativo indica que a entrada  $x^k$  pertence à classe  $j$ . Deste modo, a rede neural classifica os padrões de entrada em classes. Neste projeto, as entradas são bidimensionais, então:  $x^k = [x_1^k, x_2^k]$ . Para cada entrada, um neurônio vencedor  $w^j$  é selecionado, sendo:

$$\|x^k - w^j\|^2 < \|x^k - w^i\|^2, \forall i \neq j \quad (1)$$

Com estes dados, a rede neural treina o neurônio vencedor  $w^j$  de forma a aproxima-lo das entradas com a seguinte equação:

$$W_{\text{novo}} = W_{\text{anterior}} + \beta (X - W_{\text{anterior}}) \quad (2)$$

sendo:  $\beta$  o Passo de Treinamento da Rede Neural (*Learning Rate*)

A função do Filtro de Kalman neste projeto é estimar a posição futura da rede neural baseando-se em estados anteriores e fazer com que o rastreamento seja identificado pela forma do filtro. Visto que o Filtro de Kalman realiza a estimativa em duas fases, as equações do filtro são:

#### Predição ou Propagação no Tempo

$$X_{k+1} = \text{PHI} * X_k + \text{GAMMA} * U_k + W_k \quad (3)$$

$$P_{k+1} = \text{PHI} * P_k * \text{PHI}^T + G * Q * G^T \quad (4)$$

Sendo:  $X_k$  é o vetor de estados anteriores

$X'_{k+1}$  é o vetor de estados estimados *a priori*

$P'_{k+1}$  é a matriz de covariância de erro estimado *a priori*

### Atualização ou Correção da Estimativa

$$K = P_{k+1} * C * 1 / (C * P_{k+1} * C^T + R) \quad (5)$$

$$X_{k+1} = X_{k+1} + K * (Z - C * X_{k+1}) \quad (6)$$

$$P_{k+1} = (1 - K * C) * P_{k+1} \quad (7)$$

sendo: K é o ganho de Kalman

Z é o vetor de medida dos estados

$X_{k+1}$  é o vetor de estados corrigidos ou estimados *a posteriori*

$P_{k+1}$  é a matriz covariância de erro corrigida

As matrizes C, W, G e Q são ruídos de processo e de medição, PHI é a matriz que reporta o estado anterior ao estado futuro da rede neural, e U é uma entrada de controle.

Antes do treinamento e da determinação do neurônio vencedor da rede neural, todas as entradas e todos os neurônios devem ser normalizados, ou seja:

$$\|x^k\|^2 = \|w^i\|^2 = 1, \forall k, i \quad (8)$$

Isto é obtido da seguinte forma [13]:

$$M = \max \{\|x^k\|^2\}, \forall k \quad (9)$$

sendo M, o máximo da norma de X.

A normalização de  $x^k$  é obtida através do aumento da dimensão de  $x^k$  em uma unidade, com a introdução de uma nova componente:

$$x_{n+1}^k = (M - \sum_{i=1}^n (x_i^k)^2)^{1/2} \quad (10)$$

e o uso como entrada do vetor  $x^{nk} = M^{-1/2} [x_1^k, x_2^k, \dots, x_n^k, x_{n+1}^k]$ , que tem norma unitária. Os neurônios também devem ter seus valores normalizados através do mesmo processo.

Para todos os valores dos padrões de entrada, a rede neural é treinada, sendo que em cada passo de treinamento, apenas um neurônio é ativado. O parâmetro que define o quanto o neurônio vai se aproximar do valor de entrada é o Larning Rate. O Filtro de Kalman é aplicado a cada passo de treinamento do neurônio na rede neural. Agora, podemos construir um fluxograma do algoritmo para um melhor entendimento do processo. O fluxograma é mostrado na Figura 3

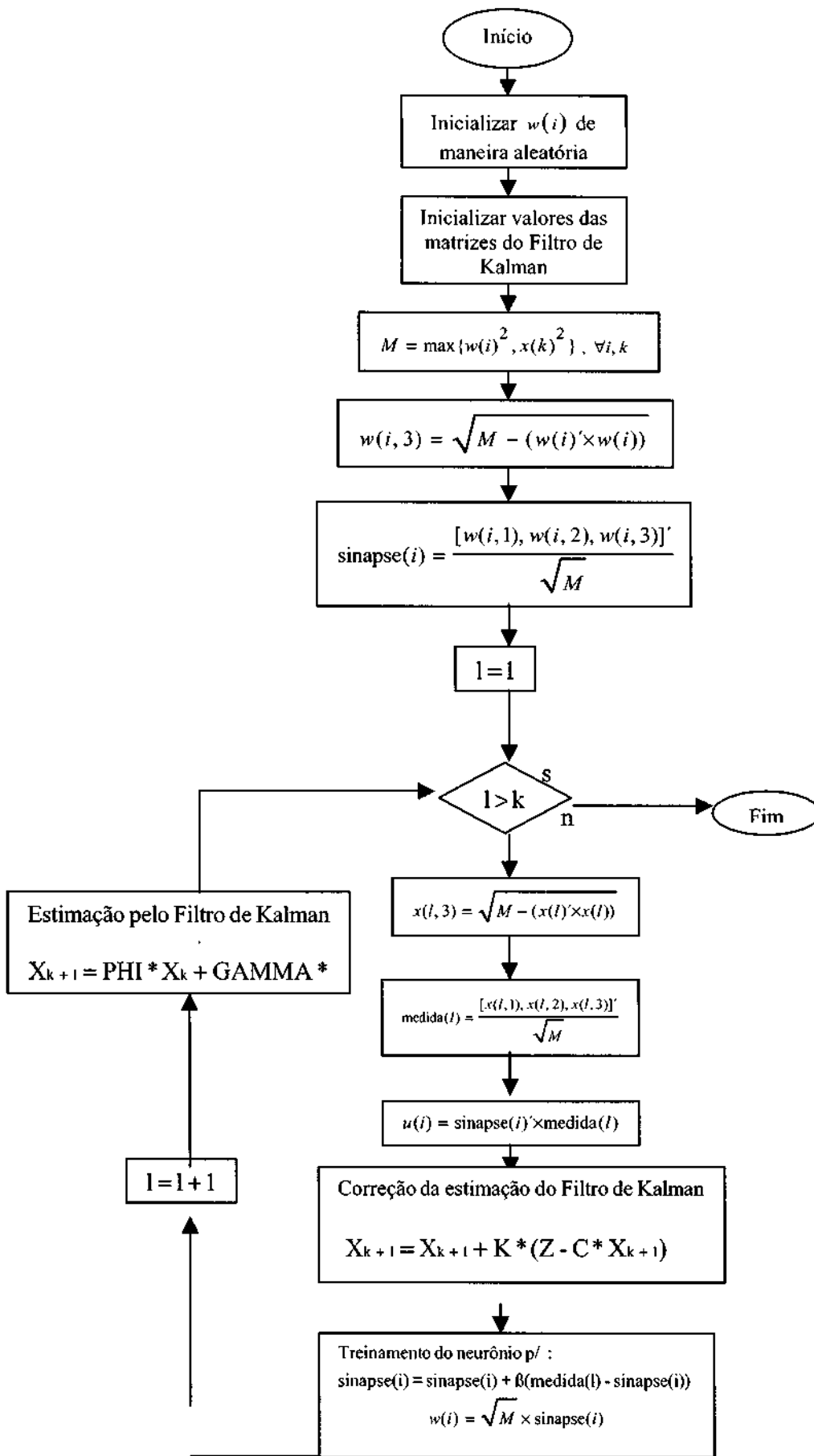


Figura 3- Fluxograma do algoritmo de Kohonen e Filtro de Kalman.

### 3.2 –Exemplos da Validação do Algoritmo

Nesta seção, são mostrados três exemplos de rastreamento com trajetórias simuladas por equações matemáticas diferentes. Nestes exemplos, o algoritmo foi validado com um conjunto de pontos se deslocando sobre as seguintes trajetórias: Reta, Logaritmo Natural e Seno, como um alvo móvel. A geração de pontos foi feita ao redor de alguns pontos escolhidos sobre as trajetórias. Na tentativa de observar o comportamento da rede neural, estes conjuntos de pontos foram fornecidos como valores de um padrão de entrada para a mesma. Três neurônios foram criados em posições aleatórias para disputarem entre si no rastreamento do objeto. A Figura 4 mostra pontos azuis como os valores dos padrões de entrada, denominada aqui como nuvem de pontos, o ponto original está em ciano e os neurônios foram representados como pequenas estrelas. A Figura 5 e Figura 6 mostram os padrões de entrada, ponto sobre a curva e posições iniciais dos neurônios como na Figura 4, porém com trajetórias simuladas pelas curvas Logaritmo e Seno, respectivamente.

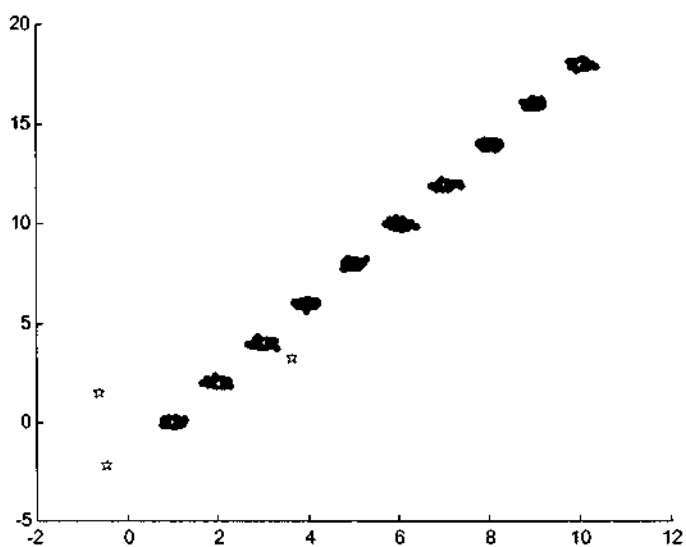
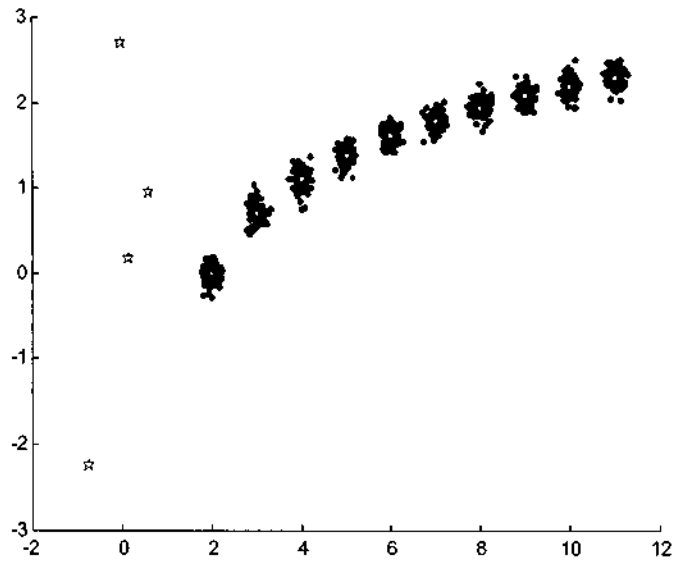
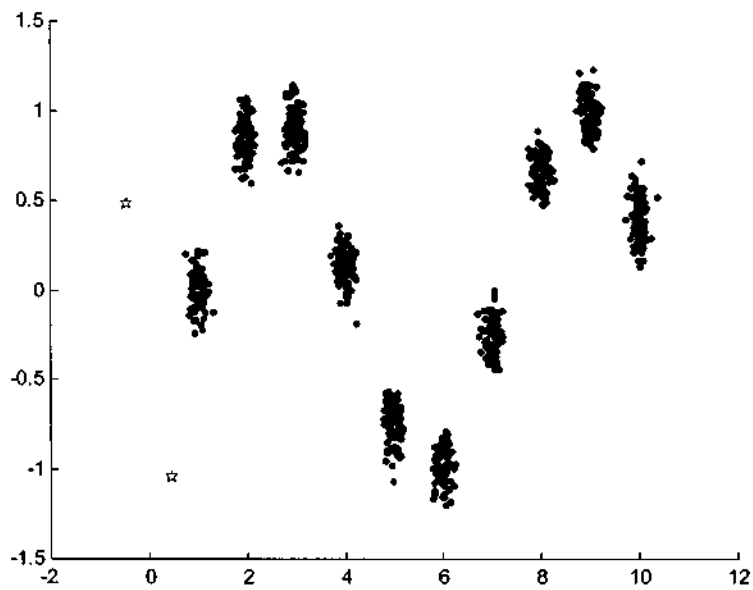


Figura 4- Nuvem de pontos e posições iniciais dos neurônios na Reta.-



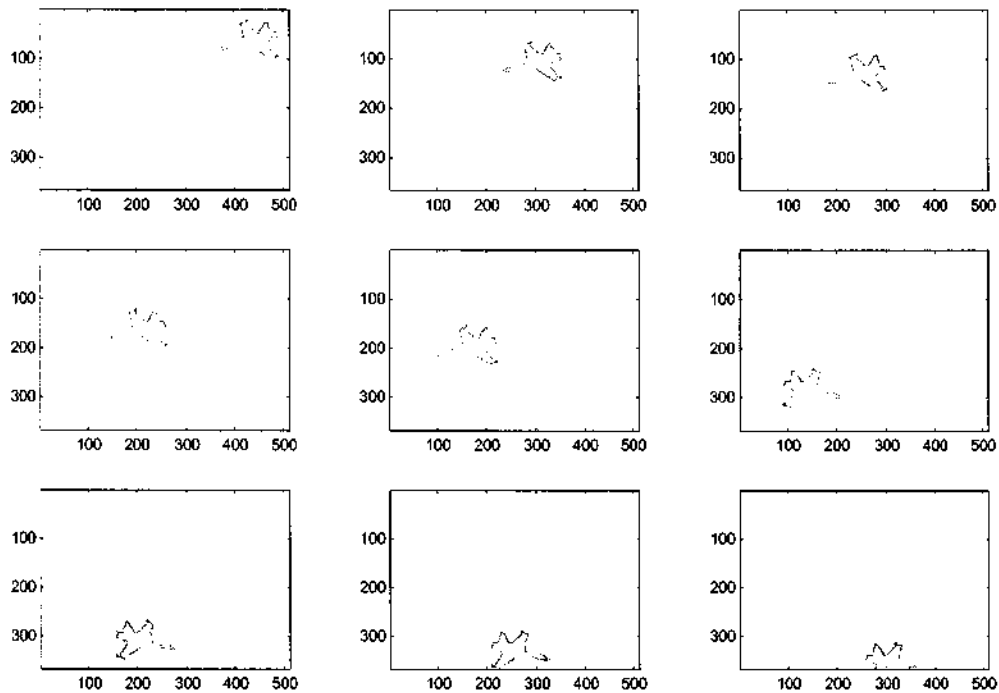


**Figura 5 - Nuvem de pontos e posições iniciais dos neurônios no Logaritmo..**



**Figura 6- Nuvem de pontos e posições iniciais dos neurônios no Seno.**

Agora, será mostrado um exemplo da aplicação do algoritmo em seqüências de imagens. A Figura 7 mostra a seqüência de imagens contendo o objeto que será rastreado pela rede neural. Este objeto, um avião, possui uma trajetória tal que o objeto muda de posição em cada uma das dez imagens.



**Figura 7- Seqüência de imagens mostrando a trajetória do avião.**

O algoritmo foi validado da seguinte maneira: As imagens foram dadas uma de cada vez como padrões de entrada para a rede neural. Depois de realizadas as normalizações dos neurônios e das entradas, a rede neural determinou o neurônio vencedor, ou seja, o que estava mais próximo da posição inicial do avião. Fazendo uma comparação com as simulações anteriores, os padrões de entrada eram um conjunto de pontos gerados em torno de um baricentro, ou seja, um ponto que estava sobre a curva. Aqui, este conjunto de pontos é dado pelos pontos diferentes da cor branca, já que o fundo do plano possui cor branca. Este conjunto de pontos foi apresentado à rede neural como valores do padrão de entrada, e assim foi realizado o rastreamento. Identificado o neurônio vencedor, o mesmo foi treinado de forma a se aproximar do padrão de entrada, deste modo realizando o rastreamento e identificando a forma da trajetória do avião. Todos os outros neurônios foram treinados de forma a serem afastados do neurônio vencedor. O Filtro de Kalman, aplicado à rede neural, realizou a estimação da posição da mesma em cada passo de treinamento.

Independentemente de o algoritmo ser aplicado a trajetórias simuladas por equações matemáticas ou ser aplicado a seqüências de imagens, o mesmo sempre possui os mesmos passos. Estes passos são descritos a seguir:

- 1 - São estabelecidas condições iniciais para os pesos dos neurônios  $w'$  de forma aleatória.
- 2 - É estabelecido um quociente de aprendizado  $\beta$  para os pesos. Inicialmente, foi definido  $\beta=0,01$ .
- 3 - São estabelecidas condições iniciais para as matrizes do Filtro de Kalman, baseado nos dados do sistema dinâmico em questão, ou seja, a Rede Neural.
- 4 - Normaliza-se as entradas e os pesos de forma que, fique como a equação (8). Os resultados da normalização serão os vetores sinapses (no caso dos pesos dos neurônios) e os vetores medidas (no caso dos padrões de entrada).
- 5 - Dado uma entrada, é calculado um neurônio vencedor, o qual possui a maior variável interna  $u(i)$ , que será o neurônio que mais se aproxima da posição atual da entrada.
- 6 - O peso do neurônio vencedor deverá ser atualizado da seguinte forma:

$$\text{sinapse}(i) = \text{sinapse}(i) + \beta(\text{medida}(po) - \text{sinapse}(i))$$

Sendo que “i” representa o índice da sinapse vencedora e “po” a entrada apresentada para a rede, ambas normalizadas no passo 4.

- 7 - Realiza-se a correção da estimativa do Filtro de Kalman com a equação (6):
- 8 - Retorna-se ao passo 4, até que todas as entradas tenham sido apresentadas à rede.

## 4 – ANÁLISE DOS RESULTADOS

Na seção anterior o algoritmo foi validado para pontos se movendo sobre trajetórias diferentes e seqüências de imagens, e foram descritos os funcionamentos da rede neural como ferramenta para realizar o rastreamento e o Filtro de Kalman como estimador da posição da rede neural em cada passo de treinamento da mesma. Nesta seção, o algoritmo será aplicado de forma a nos mostrar o comportamento das duas ferramentas utilizadas, para podermos analisar a eficiência de cada uma destas. Simulações realizadas mostraram a eficiência da rede neural no rastreamento e do Filtro de Kalman como estimador. Enquanto a rede neural realizava o rastreamento do objeto alvo, pode ser observado que, ao mesmo tempo, o Filtro de Kalman fazia a estimação da posição da mesma em cada ponto, convergindo para esta.

### 4.1 – Validação e Resultado do Algoritmo em Pontos

Na seção anterior, os baricentros das classes foram gerados a partir de funções  $f(x)$ , determinando-se funções diferentes para cada baricentro, e a partir daí foram gerados pontos próximos a cada baricentro. Nesta seção, após cada etapa de treinamento da rede, o  $x$  de cada baricentro será atualizado, gerando outros conjuntos de pontos que serão apresentados à rede para um novo treinamento, e, em cada atualização, o conjunto de pontos atual é normalizado. Com o treinamento, a posição do neurônio vencedor é informada ao Filtro de Kalman, o qual realiza a correção da estimação da mesma. As condições iniciais dos neurônios nesse novo treinamento serão iguais às condições finais do treinamento anterior, ou seja, o neurônio vencedor é inicializado de forma aleatória somente antes de qualquer treinamento.

Aqui, o algoritmo será validado de forma a realizar a identificação da trajetória do objeto alvo pela rede neural e a estimação da posição do neurônio vencedor pelo Filtro de Kalman. Serão simuladas três trajetórias distintas. As trajetórias serão as da seção anterior, com as equações das seguintes curvas: Reta, Logaritmo e Seno, como mostrado na Figura 4, Figura 5, e Figura 6, respectivamente.

#### 4.1.1 – Rastreamento do Objeto Sobre a Reta

A primeira simulação de rastreamento será a do objeto alvo se movendo com uma trajetória segundo uma reta. A Figura 4 mostra a evolução dos pontos sobre a trajetória. Foram plotados os pontos para cada baricentro sobre a reta, para que fique explícita a trajetória no qual o objeto alvo se desloca.

O baricentro foi inicializado na posição (1,0). A cada passo, o  $x$  foi incrementado em 1 para que os pontos pudessem se deslocar sobre a reta em questão, e o baricentro foi atualizado de acordo com a seguinte função:

$$F(x) = 2 * X \quad (11)$$

Pela teoria desenvolvida na seção 3, o neurônio vencedor deve ser capaz de realizar o rastreamento dos pontos, enquanto os perdedores devem se afastar do neurônio vencedor. Já o Filtro de Kalman deve identificar a forma da rede neural.

A Figura 8 mostra todo o comportamento da rede neural e do Filtro de Kalman. As estrelas pretas representam as condições iniciais das sinapses, os pontos em ciano representam os baricentros e os traços azuis representam o treinamento das sinapses. Os pontos em azul representam a nuvem de pontos e a linha tracejada em vermelho representa a estimativa do Filtro de Kalman.

Pela figura, podemos observar que a rede neural realiza o rastreamento do objeto alvo corretamente, enquanto o Filtro de Kalman faz a estimativa da mesma, como se havia previsto.

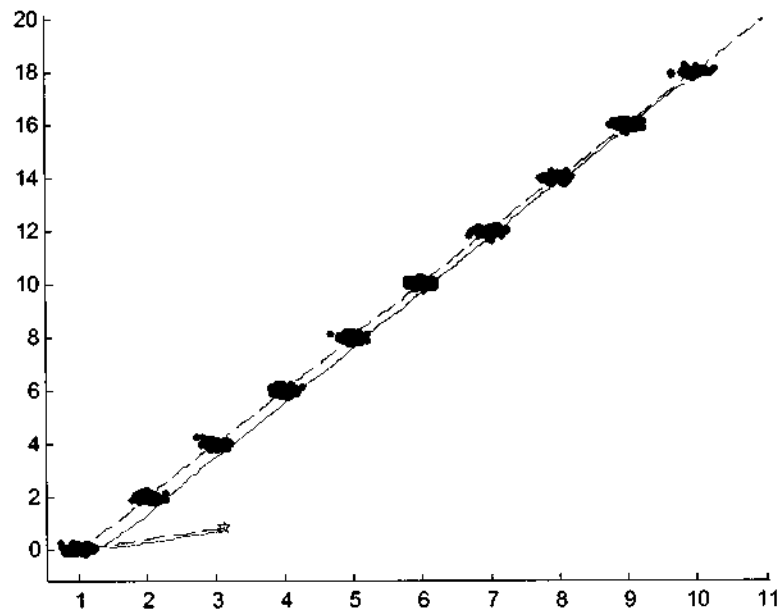


Figura 8– Resultado do Algoritmo com Trajetória Retilínea

#### 4.1.2 – Rastreamento do Objeto Sobre o Logaritmo

Nesta simulação, o rastreamento será a do objeto alvo se movendo com uma trajetória segundo uma função logarítmica. A Figura 5 mostra a evolução dos pontos sobre a trajetória. Foram plotados os pontos para cada baricentro sobre a reta, para que fique explícita a trajetória no qual o objeto alvo se desloca, como no caso anterior.

A cada passo, o  $x$  foi incrementado em 1 para que os pontos pudessem se deslocar sobre a função logarítmica, e o baricentro foi atualizado de acordo com a seguinte função:

$$F(x) = \text{LN}(X) \quad (12)$$

A Figura 9 mostra os resultados da validação do algoritmo, e, a partir da mesma podemos analisar o comportamento do sistema.

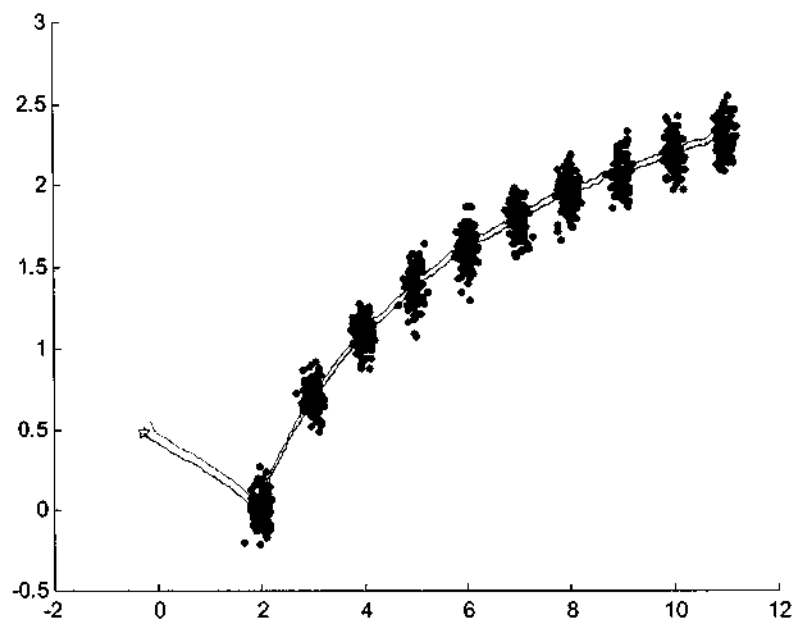


Figura 9– Resultado do Algoritmo com Trajetória Logarítmica

Pela figura, vemos que tanto a rede neural quanto o Filtro de Kalman se comportam como havíamos previsto. Apesar de esta ser uma trajetória curva, o algoritmo funcionou como esperado.

#### 4.1.3 – Rastreamento do Objeto Sobre o Seno

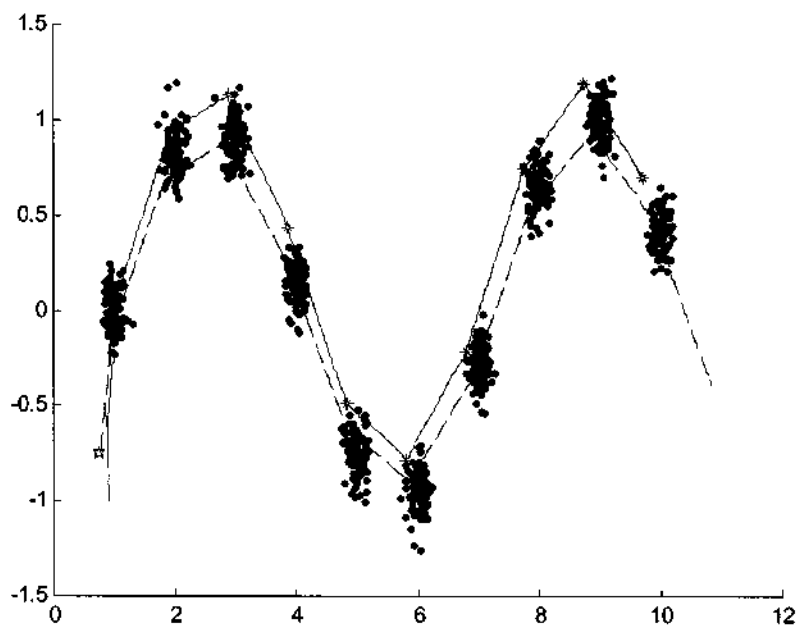
Nesta simulação, o rastreamento será a do objeto alvo se movendo com uma trajetória segundo uma função senoidal. A Figura 6 mostra a evolução dos pontos sobre a trajetória desta simulação.

Observe que esta figura foi plotada como a anterior, de forma que as mesmas características se aplicam aqui, exceto a forma da trajetória. A função utilizada foi a seguinte:

$$F(x) = \text{Sen}(X) \quad (13)$$

A Figura 10 mostra os resultados do sistema. Apesar de esta ter sido uma trajetória com grandes mudanças, a rede neural e o Filtro de Kalman obtiveram um desempenho satisfatório.

A intenção destas três simulações foi verificar o comportamento do sistema para o rastreamento de um objeto com tipos diferentes de trajetórias, desde movimentos suaves a movimentos com grandes variações no espaço.



**Figura 10– Resultado do Algoritmo com Trajetória Senoidal**

#### **4.1.4 – Rastreamento do Objeto com Mais de Uma Trajetória**

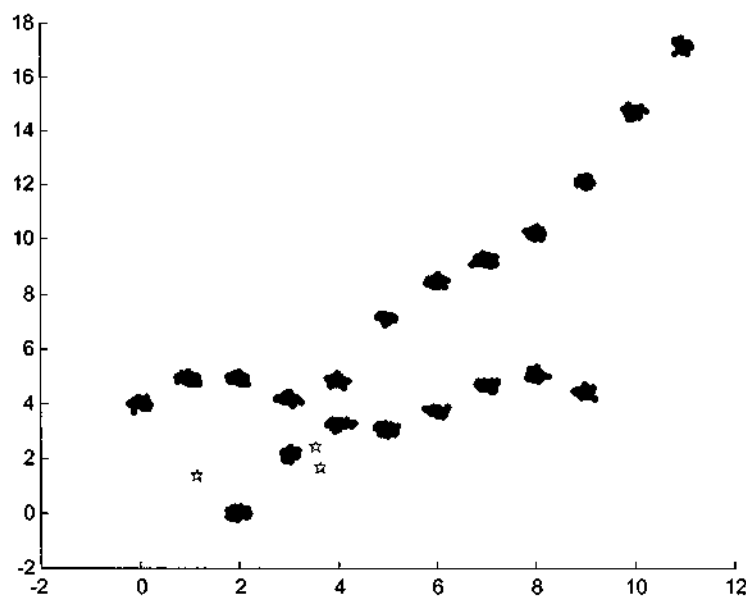
Nas três seções anteriores, o algoritmo obteve bons resultados. Mas o que acontece quando existe mais de um objeto? Pode acontecer que, durante o cruzamento das trajetórias, os neurônios se confundam e não realizem um bom rastreamento. Nesta seção são simuladas trajetórias com mais de um objeto, dadas por funções matemáticas diferentes. Vamos validar o algoritmo em duas situações diferentes: Uma com duas e outra com três trajetórias diferentes.

Na validação com duas trajetórias, as funções utilizadas para simulá-las foram as seguintes:

$$F1(x) = 1 + (1.7 * x - \text{Cos}(x)) \quad (14)$$

$$F2(x) = 4 + (\text{Seno}(x)) \quad (15)$$

A evolução dos baricentros e das nuvens é mostrada na Figura 11. É mostrada a posição de cada objeto em todos os pontos para que se possa observar a trajetória descrita pelo mesmo.



**Figura 11 Dois Objetos com Trajetórias Diferentes**

O resultado obtido está na. Figura 12 Analisando-a podemos ver que o rastreamento não é bom. Isto se deve ao fato de os dois neurônios rastreadores se aproximarem no cruzamento das trajetórias. Como o objeto se aproxima de outro neurônio, este passa a ser o neurônio rastreador deste ponto, enquanto o mesmo acontece com o outro, ou seja, há uma troca no rastreamento.

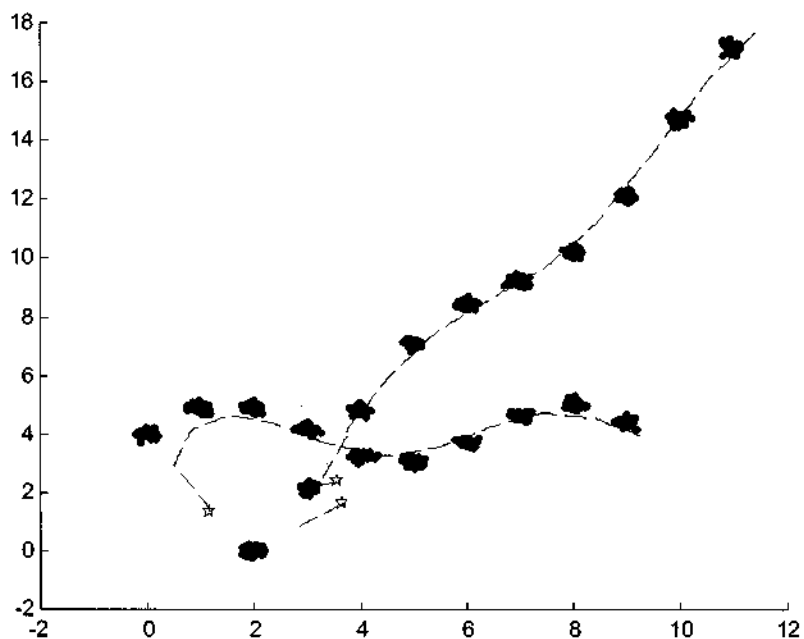
Na validação com três trajetórias, as funções utilizadas foram as seguintes:

$$F1(x) = 1 + (1.4 * x - \text{Cos}(x)) \quad (16)$$

$$F2(x) = 4 + (x^{1.2} - (x/2)^3) \quad (17)$$

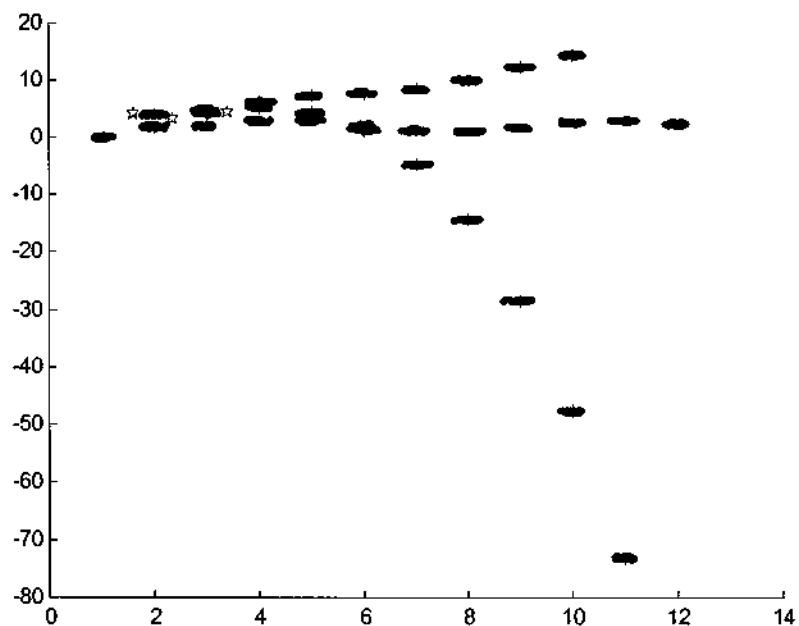
$$F3(x) = 2 + (\text{Seno}(x)) \quad (18)$$





**Figura 12– Rastreamento mal sucedido de dois objetos**

A Figura 13 mostra a evolução dos baricentros e suas respectivas nuvens, como na simulação anterior.



**Figura 13– Três Objetos com Trajetórias Diferentes**

Como pode ser observado na Figura 14, o treinamento dos neurônios vencedores não foi bom. O motivo é o mesmo do caso anterior. Já o Filtro de Kalman consegue estimar a trajetória de cada neurônio.

Assim, percebe-se que a ocorrência de cruzamento entre os alvos é um fator que pode levar a falhas no rastreamento. Outro fator importante para determinar o sucesso no rastreamento é o salto dado pelas nuvens a cada intervalo de iterações. Quanto menor é o deslocamento das nuvens, mais fácil se torna o rastreamento.

Através de diversas simulações foi possível confirmar que a posição inicial dos neurônios é crítica para o sucesso do treinamento [13]. Uma forma de minimizar esse problema seria aumentar o número de neurônios a serem treinados, esperando-se que dessa forma sempre exista um deles próximo o suficiente a uma classe para ser atraído apenas por ela. Desta forma, existe um neurônio próximo ao ponto que será treinado para fazer o rastreamento, enquanto todos os outros serão afastados do neurônio rastreador.

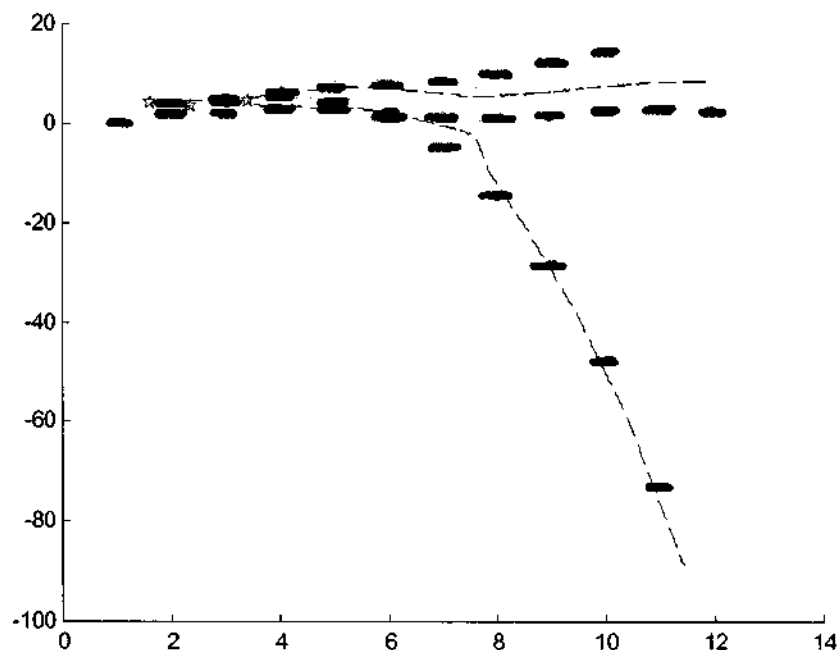


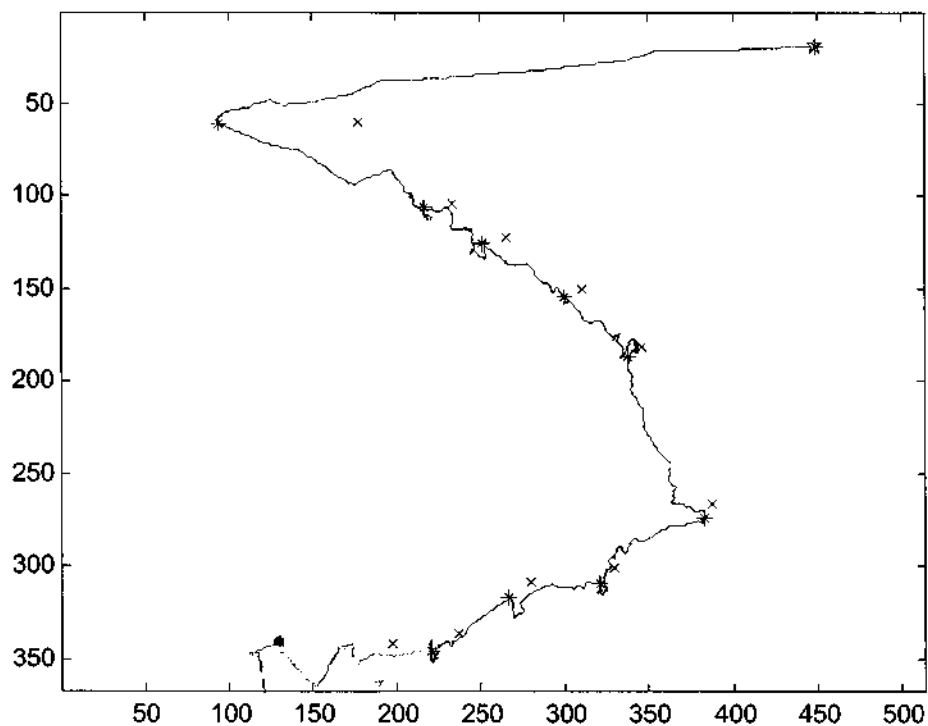
Figura 14- Rastreamento mal sucedido de três objetos

#### 4.2 – Validação e Resultados do Algoritmo em Sequências de Imagens

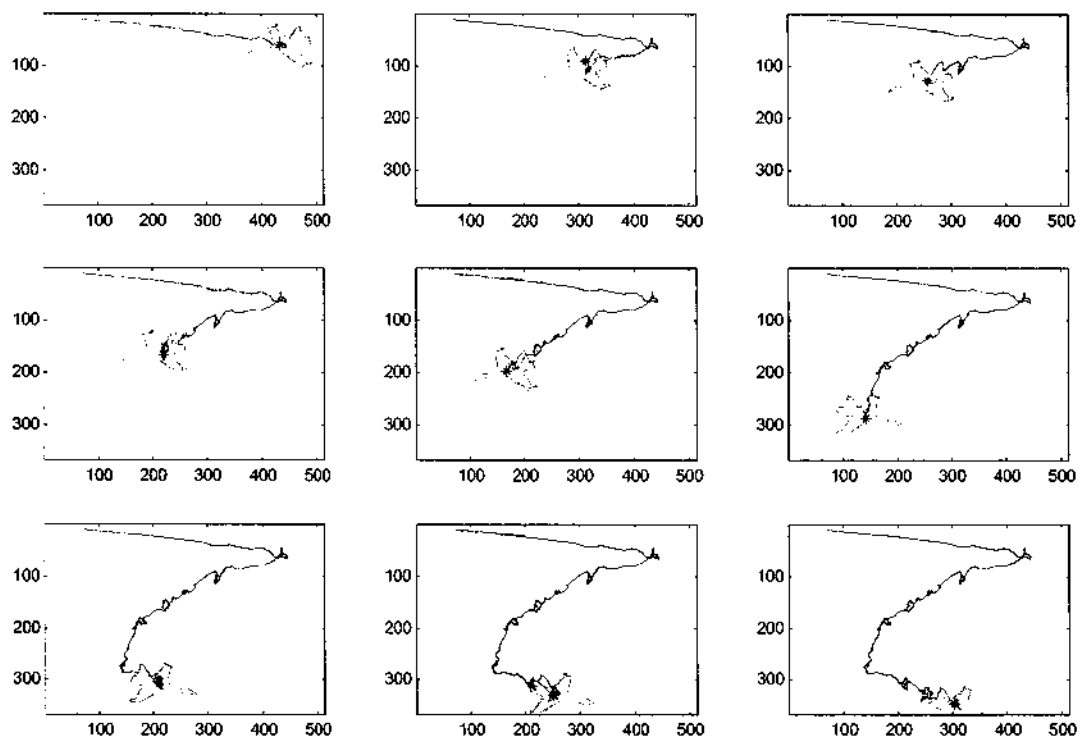
Nesta seção, o algoritmo desenvolvido será aplicado às sequências de imagens contendo como

objeto alvo em movimento, um avião. Uma das dificuldades em se rastrear alvos em imagens digitais consiste na identificação dos *pontos válidos* contidos na imagem. Estes *pontos válidos* são o conjunto de pixels na imagem que compõe o alvo a ser rastreado.

A Figura 7 mostra uma seqüência de dez imagens para visualização do movimento do avião. Em cada imagem, o neurônio vencedor é treinado para cada um dos pixels que compõem o avião. Desta forma, o neurônio se aproxima do avião, realizando o rastreamento do mesmo. O Filtro de Kalman também foi aplicado a este algoritmo, de forma a estimar a posição do neurônio rastreador, a cada passo de treinamento. Como dito anteriormente, o conjunto de pixels do avião, em cada imagem, é equivalente ao conjunto de pontos que formam a nuvem nas simulações das seções anteriores. Os resultados da validação do algoritmo são mostrados na Figura 15.



**Figura 15 – Rastreamento da Imagem com Avião**



**Figura 16- - Rastreamento do avião na seqüência de imagens.**

Analisando a Figura 15 podemos observar que o neurônio vencedor realiza o rastreamento do avião, e o Filtro de Kalman consegue estimar a posição da mesma, como foi previsto. Aqui, o Filtro de Kalman é representado por um “x” em preto. Como nos casos das simulações realizadas com pontos representando objetos móveis, a quantidade de neurônios disponíveis, as posições iniciais dos mesmos e seus respectivos Learning Rates têm influências críticas no rastreamento.

## **5 – CONCLUSÃO E SUGESTÃO PARA TRABALHOS FUTUROS**

Nesta seção será feita uma conclusão, ou seja, uma dissertação sobre os resultados obtidos e um comentário sobre o funcionamento das ferramentas utilizadas como rastreador.

### **5.1 – Conclusão**

No início da primeira fase do projeto foi feito um estudo sobre redes neurais artificiais através de livros e artigos apresentados pelo orientador. Logo após, foi feita uma pesquisa sobre a aplicação das redes neurais artificiais no rastreamento de alvos e no reconhecimento de padrões, além de um estudo sobre outras técnicas de inteligência artificial utilizadas para resolver o problema em abordado.

Após a pesquisa, foram feitos o estudo e a implementação do algoritmo baseado nas redes neurais artificiais Kohonen para a identificação de alvos. Inicialmente, o algoritmo foi validado através de simulações de rastreamento de alvos representados por grupos de pontos fixos e móveis. A rede foi então treinada em diversas condições distintas para analisar a influência das variáveis envolvidas no sucesso do treinamento. Estas variáveis são: Learning Rate da rede neural, numero de neurônios na rede, posição inicial dos mesmos, numero de objetos e suas respectivas trajetórias. Segue-se uma pequena conclusão sobre algumas destas variáveis:

Com as simulações verificou-se que a posição inicial dos neurônios é um fator de importância para o sucesso do treinamento, pois se existir algum neurônio próximo ao objeto, mais rápido o neurônio irá convergir para o objeto.

Verificou-se também que a adição de neurônios aumenta a probabilidade de sucesso, apesar de acarretar maior custo computacional, ou seja, o tempo de resposta do algoritmo é maior, e isto é prejudicial ao sistema, já que o mesmo deve realizar o rastreamento em tempo real.

Além disso, comprovou-se que o cruzamento dos alvos móveis pode dificultar o rastreamento, pois as distâncias entre os alvos e os neurônios rastreadores podem ser diferentes no cruzamento, o que gera uma troca de neurônio rastreador nos alvos.

Após isto, o algoritmo foi implementado com seqüências de imagens digitais, na tentativa de realizar o rastreamento do objeto que se movia. O algoritmo foi implementado de forma que, a cada

imagem retirada da seqüência, o neurônio recebia um treinamento para cada pixel que representava o objeto na imagem. O rastreamento realizado pela rede neural apresentou resultados satisfatórios de tal forma que conseguiu identificar a trajetória do objeto-alvo.

Na segunda fase do trabalho, foi implementado o Filtro de Kalman no algoritmo da rede neural, na tentativa de estimar a posição da mesma em cada passo de seu treinamento.

Primeiramente, o filtro foi aplicado aos algoritmos com trajetórias simuladas por equações matemáticas e objetos representados por pontos. Simulações mostraram que o filtro apresentou uma boa estimativa da rede neural, conseguindo identificar sua forma.

Posteriormente, o filtro foi aplicado aos algoritmos com seqüências de imagens digitais. Verificou-se que o filtro realizou uma estimativa satisfatória da rede neural.

Entretanto, outros critérios de validação podem ser adotados na implementação do algoritmo rastreador, utilizando para as mesmo outras técnicas existentes, ou simplesmente modificando a forma como as ferramentas foram utilizadas neste trabalho.

## **5.2 – Sugestões Para Trabalhos Futuros**

Outros trabalhos podem ser realizados utilizando as ferramentas ou até mesmo a idéia empregada aqui. A utilização das mesmas, podem ser feitas com variações nas pesquisas e nas implementações, de forma a minimizar alguns erros gerados pela rede neural. Seguem-se aqui algumas sugestões:

1. A pesquisa de como minimizar a chance de um neurônio rastrear dois alvos diferentes, no caso do treinamento de apenas um neurônio envolvendo mais de um alvo.
2. O estudo e implementação de técnicas que evitem que neurônios troquem de alvos quando ocorre o cruzamento dos mesmos durante o movimento, no caso do treinamento de mais de um neurônio.
3. A pesquisa e validação de outras formas de implementação do Filtro de Kalman como estimador da rede neural ou como um rastreador propriamente dito.
4. Estudo teórico e implementação prática de formas variadas do Filtro de Kalman, como o Filtro de Kalman Estendido, Filtro de Kalman-Bucy e outras variações.

5. Implementação de um algoritmo que simule trajetórias tridimensionais para realizar o rastreamento do mesmo, ou generalizar o caso, desenvolvendo um sistema que rastreie o objeto alvo, qualquer que seja sua trajetória ou dimensão..

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Krose, Ben; Smagt, Patrick Van Der – **An Introduction to Neural Networks**, eighth edition.
- [2] Wu, S. J., Lin, C. T., **Global Optimal Fuzzy Tracker Design Based on Local Concept Approach**, IEEE Transaction On Fuzzy Systems, 10 (2): 128-143, 2002.
- [3] Schram, G., vanderLinden, F.X., Krose, B.J.A., Groen, F.C.A., **Visual tracking of moving objects using a neural network controller**, Robotics And Autonomous Systems, 18 (3): 293-299 AUG 1996.
- [4] Gaudiano, P., Zalama, E., Coronado, J.L., **An unsupervised neural network for low-level control of a wheeled mobile robot: Noise resistance, stability, and hardware implementation**, IEEE Transactions On Systems Man And Cybernetics Part B-Cybernetics, 26 (3): 485-496 JUN 1996.
- [5] Chen, Y. M., Hsueh, C.S., **Complementary Data Fusion in Vision-guide and Control of Robotic Tracking**, Robotica, 19: 53-58 Part 1, 2001.
- [6] Ma, C.W., Teng, C.C., **Tracking a near-field moving target using fuzzy neural networks**, Fuzzy Sets And Systems, 110 (3): 365-377 MAR 16 2000.
- [7] Yu, X., Azimi-Sadjadi, M.R., **Neural network directed bayes decision rule for moving target classification**, IEEE Trans. on Aerospace and Electronic Systems, 36 (1): 176-188, 2000.
- [8] Yang, S. X., Meng, M., **Neural Network Approaches to Dynamic Collision-Free Trajectory Generation**, IEEE Transaction on System Man and Cybernetic Part B-Cybernetics, 31 (3): 302-318, 2001.
- [9] Cameron, S., Grossberg, S., Guenther, F.H., **A self-organizing neural network architecture for navigation using optic flow**, Neural Computation, 10 (2): 313-352, 1998.
- [10] Missler, J.M., Kamangar, F.A., **A Neural-Network For Pursuit Tracking Inspired By The Fly Visual-System**, Neural Networks, 8 (3): 463-480 1995.
- [11] Dror, I. E., Florer, F. L., Rios, D., Zagaeski, M., **Using Artificial Bat Sonar Neural Networks for Complex Pattern Recognition: Recognizing Faces and the Speed of a Moving Target**, Biological Cybernetics, 74 (4): 331-338, 1996.
- [12] Navabi, H., Agarwal, A., **Adaptative Response Organizer Network for Space-time Patterns in Low Level Vision**, Neural Networks, 11 (5): 825-836, 1998.
- [13] Calôba, Luis P., **Introdução à Computação Neuronal**, IX Congresso Brasileiro de Automação, 1992, Vitória, ES. Mini-curso 2. p. 25-38.
- [14] Yi Li, Songde Ma, Hanqing Lu, **Human posture recognition using multi-scale morphological method and Kalman motion estimation**, Proceedings of the IEEE International Conference on Pattern Recognition, 1998, pp. 175-177.
- [15] Pereira, Guilherme A. S., **Filtro de Kalman: Teoria e Aplicações**, CPDEE - UFMG
- [16] Nascimento Jr., C. L.; Yoneyama, Takashi – **Inteligência Artificial em Controle e Automação**, Ed. Edgard Blucher Ltda, 2000.
- [17] Kovács, Zsolt L. – **Redes Neurais Artificiais: Fundamentos e Aplicações**, 2ª ed São Paulo: Edição Acadêmica, 1996.
- [18] Masters, Timothy – **Practical Neural Network Recipes in C++**, Academic Press.



- [19] LiMin Fu – **Neural Networks in Computer Intelligence**, University of Florida, Gainesville.
- [20] Rojas, Raúl – **Neural Networks: A Systematic Introduction**, Springer.
- [21] Balakrishnan, S. N.; Weil, R.D. – **Neurocontrol: A Literature Survey**, Elsevier Science Ltd, 1996.
- [22] Agbinya, Johnson I.; Rees, David – **Multi-Object Tracking in Video**, Academic Press, 1999.
- [23] Matthews, N.D., Na, P.E., Charnley, D., Harris, C.J., **Vehicle detection and recognition in greyscale imagery**, *Control Engineering Practice*, 4 (4): 473-479 APR 1996.
- [24] Kocjancic, R., Zupan, J. **Application of a feed-forward artificial neural network as a mapping device**, *Journal Of Chemical Information And Computer Sciences*, 37 (6): 985-989 Nov-Dec 1997.
- [25] Tipping, M.E., Lowe, D., **Shadow targets: A novel algorithm for topographic projections by radial basis functions**, *Neurocomputing*, 19 (1-3): 211-222 Mar 1998.
- [26] Amoozegar, F., **Neural-network-based target tracking state-of-the-art survey**, *Optical Engineering*, 37 (3): 836-846, 1998.
- [27] Wann, C.D., Thomopoulos, S.C.A., **Application of self-organizing neural networks to multiradar data fusion**, *Optical Engineering*, 36 (3): 799-813, 1997.
- [28] Cameron, S., Grossberg, S., Guenther, F.H., **A self-organizing neural network architecture for navigation using optic flow**, *Neural Computation*, 10 (2): 313-352, 1998.
- [29] Yu, N., Wu, H., Wang, R.S., **Morphological neural networks with applications to image target detection**, *Chinese Journal Of Electronics*, 10 (1): 85-90, 2001.
- [30] Shams, S. **Neural Network Optimization for Multi-Target Multi-Sensor Passive Tracking**, proceedings of the IEEE, vol. 84, No. 10, October, 1996.
- [31] Gurney, K., **Neural Nets**, Dpt. Human Sciences, Brunel University, Uxbridge, Middx, UK.
- [32] Fu, LiMin, **Neural Networks in Computer Intelligence**, c1994 by McGraw-Hill, Inc.
- [33] Junior, B.M., Anido, R.O., **Object Detection with Multiple Cameras**, CAPES and PRONEX-SAI, grant number 2346.
- [34] Fioretti, S., **Signal processing in movement analysis (a state-space approach)**, Ancona: Elsevier Science B.V., c1996.
- [35] Fagiani, C., **An Evaluation of Tracking Methods for Human-Computer Iteration**, Senior Thesis 2002, Boston College – Computer Science Department.
- [36] Roux, J. Le, **An introduction to Kalman filter**, University of Nice, 3 janiver 2003.
- [37] Çavusoglu, M.C., Tendick, F., **Kalman Filter Analysis for Quantitative Comparison of Sensory Schemes in Bilateral Teleoperation Systems**, Presence, December 2002.
- [38] Welch, G., Bishop, G., **An Introduction to the Kalman Filter**, Copyright 2001 by ACM, Inc. Disponível em: <http://www.cs.unc.edu/~welch>.
- [39] Gelb, A., **Applied Optimal Estimation**, M.I.T. Press 1996.
- [40] Brown, R.G., Hwang, P.Y.C., Wiley, J., and Sons, **Introduction to Random Signals and Applied Kalman Filtering**, Inc. New York, 1992.
- [41] Shet, K.C., Rao, B.V., **An Overview Of The Kalman Algorithm**, *International Journal Of Electronics*, 59 (6). 657-665 Dec. 1985.
- [42] Garcia, Francis, Filho, José E. de A., - **Estudo de Redes Neurais Artificiais na Identificação de Alvos Móveis em Processamento Digital de Imagens**, LIT, INPE, Maio de

2003.