



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-12979-PRE/8256

**REDES NEURAIIS E FILTRO DE KALMAN PARA
RASTREAMENTO DE ALVOS COM TRAJETÓRIAS
TRIDIMENSIONAIS**

Cassiano Rodrigo Silva*

*Bolsista UNESP

Relatório Final de Projeto de Iniciação Científica (PIBIC/CNPq/INPE), orientado pelo
Dr. José Ernesto de Araújo Filho



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Redes Neurais e Filtro de Kalman para Rastreamento de Alvos com Trajetórias Tridimensionais

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 2005

ÍNDICE

ÍNDICE DAS FIGURAS	3
1 INTRODUÇÃO	4
1.1 OBJETIVO DO TRABALHO	5
1.2. DESENVOLVIMENTO REALIZADO	6
2 SISTEMAS INTELIGENTES NA IDENTIFICAÇÃO DE ALVOS MÓVEIS	9
3 REDES NEURAIS KOHONEN E FILTRO DE KALMAN	13
3.1. IMPLEMENTANDO REDES NEURAIS E O FILTRO DE KALMAN	15
<i>Predição ou Propagação no Tempo</i>	17
<i>Atualização ou Correção da Estimativa</i>	17
3.2. EXEMPLOS DA VALIDAÇÃO DO ALGORITMO	20
4 RESULTADOS E DISCUSSÕES	22
4.1 – RASTREAMENTO DO OBJETO SOBRE A RETA	22
4.2 – RASTREAMENTO DO OBJETO COM TRAJETÓRIA LOGARÍTMICA	23
4.3 – RASTREAMENTO DO OBJETO COM TRAJETÓRIA SENOIDAL	24
4.4 – RASTREAMENTO DE MÚLTIPLOS OBJETOS-ALVO	25
5 5– CONCLUSÕES	31
5.2 – SUGESTÕES PARA TRABALHOS FUTUROS	31
6 REFERÊNCIAS BIBLIOGRÁFICAS	33

ÍNDICE DAS FIGURAS

Figura 1 - Rastreamento com Falha no Treinamento.....	7
Figura 2 - Rastreamento bem sucedido no treinamento.....	7
Figura 3 - Esquema geral de rastreamento.....	9
Figura 4 - Curva padrão de treinamento da rede neural Kohonen.....	16
Figura 5 - Curva de treinamento da rede neural Kohonen utilizada neste projeto.....	16
Figura 6 – Predição e atualização dos estados no Filtro de Kalman.....	17
Figura 7 - Convergência da Matriz de Covariância, P.....	18
Figura 8 - Fluxograma do algoritmo de Kohonen e Filtro de Kalman.....	19
Figura 9 - Nuvem de pontos e posições iniciais dos neurônios na Reta.....	20
Figura 10 - Nuvem de pontos e posições iniciais dos neurônios no Logaritmo.....	20
Figura 11 - Nuvem de pontos e posições iniciais dos neurônios no Seno.....	21
Figura 12 - Resultado do Rastreamento com Trajetória Retilínea.....	23
Figura 13 - Resultado do Algoritmo com Trajetória Logarítmica.....	24
Figura 14 - Resultado do Algoritmo com Trajetória Senoidal.....	25
Figura 15 - Dois Objetos com Trajetórias Diferentes.....	26
Figura 16 - Rastreamento Mal Sucedido com Dois Objetos.....	27
Figura 17 - Três Objetos com Trajetórias Diferentes.....	28
Figura 18 - Rastreamento Mal Sucedido de Três Objetos.....	28
Figura 19 - Rastreamento Bem sucedido de Dois Objetos.....	29
Figura 20 - Rastreamento Bem sucedido de Três Objetos.....	30

1 INTRODUÇÃO

A identificação e rastreamento de alvos móveis com trajetórias tridimensionais utilizando Redes Neurais Artificiais e o Filtro de *Kalman* é apresenta neste trabalho. Para a identificação da trajetória do objeto-alvo foram utilizadas redes neurais artificiais do tipo *Kohonen*, enquanto o filtro de Kalman é empregado para prever o comportamento futuro do movimento.

As redes neurais têm sido utilizadas para solucionar diversos problemas de engenharia como, por exemplo, em processamento digital de imagens e robótica e processamento digital de sinais. Redes neurais artificiais são técnicas computacionais que utilizam como modelo o funcionamento do cérebro humano, onde os neurônios são treinados de forma a responder somente aos padrões de entrada, podendo ser utilizadas no reconhecimento de padrões sem a necessidade de um treinamento prévio. 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. Redes neurais supervisionadas - por exemplo, *feed-forward*, *back-propagation*. – requerem que o usuário forneça os dados de saída desejada e os dados de entrada para que a rede neural realize o treinamento. Redes neurais não-supervisionadas é outra técnica de treinamento no qual os neurônios não necessitam que o usuário forneça uma saída para a rede. Mas neste trabalho foram utilizadas técnicas de aprendizado não supervisionado do tipo Kohonen.

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. Neste trabalho, estas técnicas são utilizadas na identificação do objeto-alvo em seqüências de imagens e seqüência de sinais elétricos, e na identificação do movimento de forma que os neurônios sejam treinados para se aproximarem dos objetos-alvo (rastreamento), que neste caso são os padrões de entrada. Adicionalmente, os neurônios perdedores também recebem um treinamento; têm seu peso alterado de forma contrária do neurônio vencedor. Eles afastam-se 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 para análise do comportamento dinâmico e para serem

utilizados em sistemas em tempo real.. 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 (há uma falha no rastreamento) o filtro de Kalman é utilizado para estimar a posição futura do alvo, e corrigir um possível erro de trajetória da rede neural.

A função do Filtro de Kalman é estimar os estados futuros de um sistema dinâmico, e para isso baseia-se nos estados anteriores medidos neste sistema e O Filtro de Kalman realiza o processo de estimação em duas fases: (1) Predição, ou propagação no tempo, e (2) Atualização, ou correção das estimativas. Na fase de predição, o filtro utiliza os estados anteriores do sistema medidos para propagar no tempo uma estimativa *a priori* do estado futuro e uma matriz de covariância de erro. As estimativas *a priori* do estado futuro e da covariância de erro são chamadas de predição. Na fase de atualização, as saídas do sistema se tornam disponíveis e o filtro as utiliza como parâmetros para corrigir a predição dos estados e a matriz de covariância de erro, gerando uma estimação *a posteriori* (corrigida) dos estados do sistema. O filtro realiza este processo diversas vezes reduzindo, a cada passo, o erro de estimativa. Em algumas aplicações o Filtro de Kalman tem sido utilizado como o sistema que se comporta como o próprio rastreador, mas, 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 trajetórias tridimensionais com o objetivo de analisar o comportamento da mesma. Com o objetivo de auxiliar o rastreamento da rede neural, o filtro de *Kalman* foi utilizado. Para a tarefa de rastreamento 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. Já uma rede neural do tipo *back-propagation* necessitaria que informássemos a ela as saídas desejadas, para poder realizar o treinamento.

A pesquisa foi feita através de estudo de livros, apostilas e artigos publicados por pesquisadores da área, para formar uma base de conhecimento sobre redes neurais. Após isto, foram implementados alguns algoritmos com o intuito de observar na prática o funcionamento das redes neurais. Depois deste estudo sobre redes neurais, foi realizada uma pesquisa sobre o

Filtro de Kalman. Também foi pesquisado em livros e artigos e depois alguns algoritmos selecionados foram implementados para se observar seu funcionamento e a redução do erro de estimação. Para tal pesquisa, foram utilizadas as seguintes bibliotecas: (1) Instituto Nacional de Pesquisas Espaciais; (2) 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, e (2) Capes – www.periodicos.capes.gov.br. Em sites de busca: (1) yahoo (Brasil) – www.yahoo.com.br, e robôs de busca: (1) radix – www.radix.com.br, (2) uol – www.radaruol.com.br, e (3) google – www.google.com; o robô (Metamáquina) de busca www.metaminer.com; (4) scirus – www.scirus.com; (5) Jarbas – www.jarbas.com.br. Foram feitas pesquisas também nas bibliotecas eletrônicas, via internet, da Universidade de São Paulo (USP) e da Universidade Estadual de Campinas (Unicamp).

1.2. Desenvolvimento Realizado

Na primeira fase do projeto foram implementados algoritmos de redes neurais para realizar o rastreamento de alvos móveis utilizando o software Matlab. A fim de simular uma situação real no qual seriam utilizados sensores elétricos de posição para obter medidas discretas no tempo da posição do alvo, foram utilizados conjuntos de pontos para representar o objeto-alvo. Inicialmente, a validação se deu com pontos estáticos e a rede neural foi treinada de forma a se aproximar do ponto, identificando os padrões. Assim, foi possível verificar a identificação do objeto-alvo pela rede neural, realizada pela sua convergência para o conjunto de pontos. Após esta implementação, foram simuladas trajetórias para este conjunto de pontos, 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 sucesso, conseguindo identificar a forma da trajetória do objeto-alvo. Foram simuladas, também, movimentos com múltiplos objetos-alvo de forma a se cruzarem e assim acrescentar um grau de dificuldade maior para a validação do sistema ao mesmo tempo em que se verificou a possibilidade de condições reais para este tipo de aplicação.

Ao utilizar-se somente a rede neural kohonen para a identificação da trajetória, 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, ou quando um alvo se aproximava de mais de um neurônio, o que confundia os neurônios rastreadores.

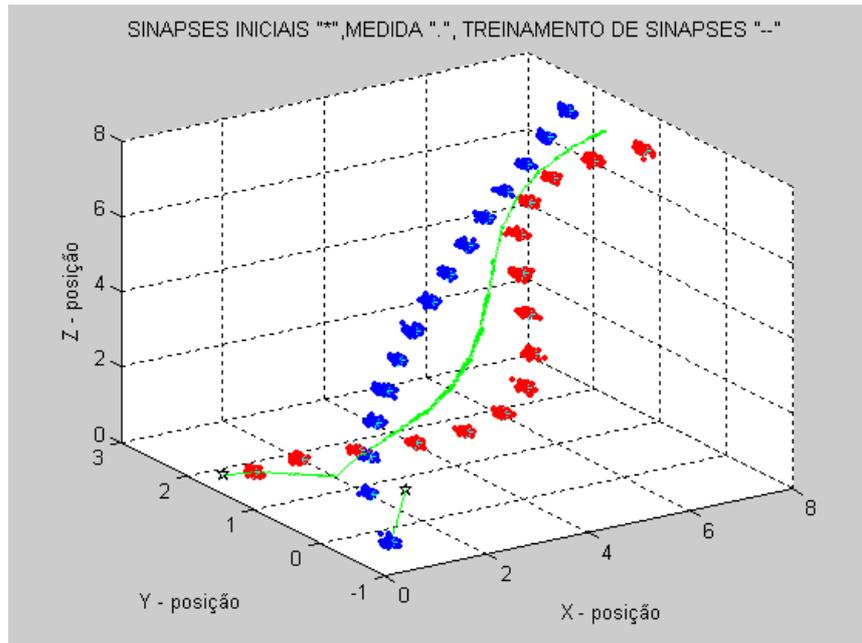


Figura 1- Rastreamento com Falha no Treinamento

Como uma tentativa de minimizar estes efeitos indesejáveis, o Filtro de Kalman foi incorporado à rede neural como uma ferramenta auxiliar com a intenção de estimar a posição futura do neurônio rastreador e para que não ocorresse às falhas no rastreamento. Pelos testes realizados, pode ser observado que o Filtro de Kalman realizou uma estimativa satisfatória da posição da rede neural, identificando seu comportamento dinâmico (Figura 2).

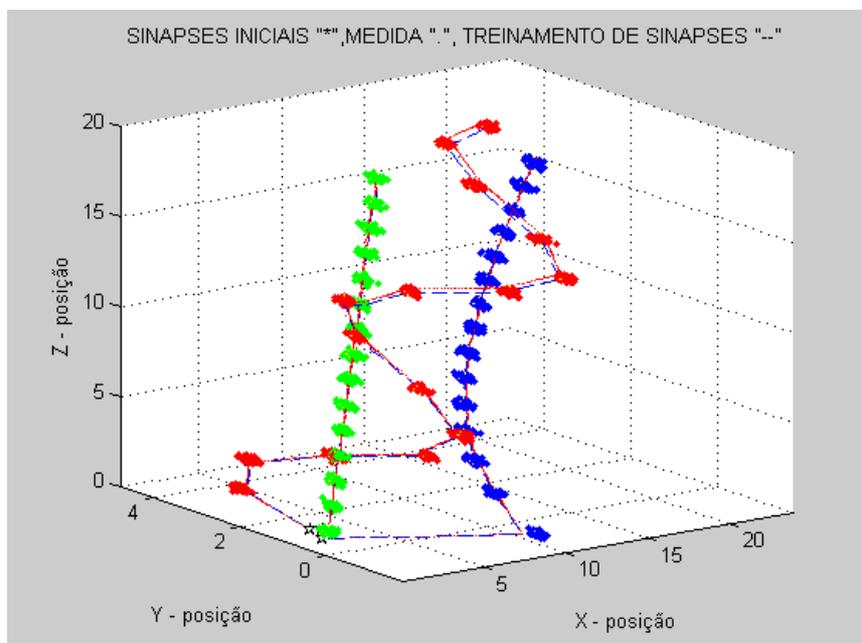


Figura 2 - Rastreamento bem sucedido no treinamento.

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 pode-se também analisar a influência da mudança de posição e do cruzamento dos alvos durante o rastreamento.

2 SISTEMAS INTELIGENTES NA IDENTIFICAÇÃO DE ALVOS MÓVEIS

A identificação de alvos móveis pode ser representada, de um modo geral, como mostrada na Figura 3. Neste artigo a ênfase é no rastreamento do objeto-alvo e na identificação de sua trajetória.

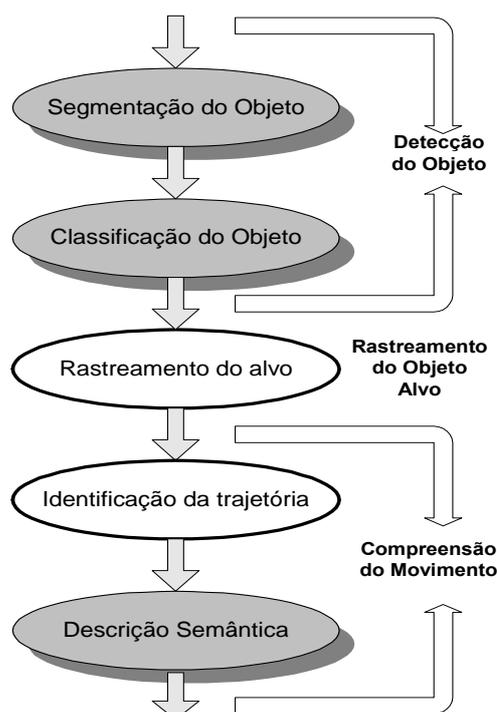


Figura 3 - Esquema geral de rastreamento.

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. Nos rastreamentos em tempo real, o algoritmo deve realizar o rastreamento em cada instante de tempo, ou a cada amostragem realizada por um conjunto de sensores elétricos de posição.

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 rastreamento 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. 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 à 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. 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 [27] 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’. D. Willett, C. Busch, e F. Seibert [43] utilizaram redes neurais do tipo *Kohonen* para realizar um algoritmo rápido para fazer análise de imagens. Este algoritmo poderia ser aplicado nas áreas médicas para segmentar objetos anatômicos, tais como tumores cerebrais de dados de ressonância magnética. Pode também ser aplicado a classificações de imagens de satélite.

Danil Prokhorov [44] desenvolveu um modo auxiliar para o treinamento de redes neurais recorrentes, pelo fato de elas estarem sendo utilizadas para uma ampla variedade de aplicações práticas e abstratas. Prokhorov realizou o treinamento da rede neural baseando-se no Filtro de *Kalman* Estendido. H.D. Cheng, X. H. Jiang, Y. Sun, e Jingli Wang [45] fizeram um resumo das técnicas de segmentação de imagens coloridas que são utilizadas para realizar sua análise através de redes neurais. Gary Bradski e Stephen Grossberg utilizaram uma família de redes neurais auto-organizáveis chamada VIEWNET para realizar o reconhecimento de objetos tridimensionais a partir de seqüências de suas visualizações bidimensionais. Sung-Bae Cho [46] utilizou um método híbrido que combinava algoritmos genéticos com redes neurais para realizar

reconhecimento de padrões. O algoritmo produziu melhores resultados que alguns métodos convencionais. Várias aplicações de rastreamento se referem a imagens digitais, como descrito anteriormente.

Mas, e se quisermos representais tais imagens tridimensionalmente? Para o campo de representação de imagens digitais tridimensionais Sargur N. Srihari [47] fez um levantamento dos métodos de representação de dados espaciais tridimensionais. A representação tridimensional de imagens digitais é encontrada numa variedade de problemas, tais como tomografia computadorizada, modelagem biológica e visão computacional. Assim podemos realizar rastreamento de objetos em imagens de uma forma tridimensional. Rajani Srinivasan e Jason M. Kinser [48] apresentaram uma técnica humana de reconhecimento de imagens utilizando uma rede neural do tipo “pulse-coupled”. A técnica funciona da seguinte maneira: Nossos olhos reúnem informações sobre diferentes pontos de interesse dentro da imagem e assimila esta informação para identificá-la. Estes pontos são geralmente as bordas ou cantos dos segmentos dentro da imagem. Este tipo de rede neural tem a habilidade de segmentar a imagem, onde as bordas e cantos dos segmentos são similares aos pontos reunidos pelos olhos. A aplicação conseguiu reconhecer um caractere escrito “à mão”.

Brijnesh e Wysotzki [49] apresentaram um fundamento teórico para aprendizado competitivo no domínio de gráficos de um modo conexionista. Eles programaram um modelo de rede neural para aprendizado competitivo no domínio de gráficos.

Em [50] foi proposto um rastreamento utilizando redes neurais e o filtro de Kalman. Eram retiradas medidas de um radar, estas medidas representavam as coordenadas da posição do alvo. Este radar realizava medições em coordenadas esféricas, ou seja, media o raio, ou alcance r , o ângulo de *azimuth* Θ , e o ângulo de elevação \varnothing . Estas medidas eram passadas para um Filtro de Kalman, o qual realizava as estimativas. Estas estimativas continham um certo erro, os quais eram passados para uma rede neural do tipo *Back-Propagation*. Esta rede neural reduzia os erros, e o sistema adicionava a saída da rede neural com os estados estimados do filtro de Kalman, obtendo assim um rastreamento mais preciso. O algoritmo realizava estimativas da posição, velocidade e aceleração do alvo. Simulações mostraram que o algoritmo se mostrou eficiente, realizava um rastreamento preciso, em tempo real, e com baixo custo computacional.

3 REDES NEURAIAS KOHONEN E FILTRO DE KALMAN

Aqui é dada uma teoria e implementação de um algoritmo baseado em 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. A teoria é considerada já sendo dada uma idéia de como aplicar a rede o filtro em conjunto para realizar o rastreamento.

As Redes Neurais *Kohonen* são um tipo de rede neural 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. Existem as redes neurais com aprendizado supervisionado e não supervisionado. As redes supervisionadas possuem como uma das características, o fato de precisarmos informar uma saída para a rede, ou seja, os valores desejados na saída da rede. Assim, o treinamento deste tipo de rede é baseado nestes valores informados pelo usuário.

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, o que é conhecido como agrupamento. 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*, é necessário 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. Pensando em rastreamento no espaço físico, temos um alvo e as coordenadas da posição deste alvo servirão como valores de entrada para a rede neural. A partir destes é determinado o neurônio vencedor, e este neurônio é o que está mais próximo, geometricamente, deste alvo, sendo assim o mais apto a ser treinado para rastrear este alvo, pois já que está mais próximo, converge mais rápido.

A função do Filtro de Kalman é realizar 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 do tipo branco Gaussiano, ou seja, de média zero. Estes ruídos são originados de dispositivos de medição, como um sensor elétrico de posição, por exemplo. O que precisamos informar ao filtro são os estados anteriores do sistema, a covariância de erro inicial 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 futuros 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, a medida correta 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. Fazendo isso, o filtro de *Kalman* converge para os estados reais do sistema dinâmico, e a partir deste momento, as estimativas dos estados futuros se tornam muito mais precisas. É exatamente quando o filtro realiza a convergência que ele é aplicado à rede neural.

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, que são as coordenadas da posição do alvo, 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 padrões 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, realizando o rastreamento e identificando a trajetória do mesmo.

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. Se olharmos para o caminho que o neurônio percorreu, conseguimos identificar a trajetória do(s) alvo(s). Porém, com este problema da troca de rastreamento, os neurônios podem perder os alvos, não rastreá-los mais, e assim não conseguimos identificar as trajetórias dos alvos.

Uma possível solução para este problema está no Filtro de Kalman. Sendo a rede neural um sistema dinâmico, seus estados (posição X, Y e Z 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, os dados atualizados na saída da rede estão disponíveis, os quais representam as coordenadas da posição da rede neural, e o filtro utiliza estas medidas para corrigir os estados estimados e a covariância de erro da estimativa. Estes estados corrigidos são denominadas “estimativas” *a posteriori*.

Deste modo, o Filtro de *Kalman* nos indica as coordenadas da posição futura da rede neural. Estas estimações do filtro são utilizadas pela rede neural como um dos parâmetros de treinamento da rede. Tendo esta informação a mais, a chance de os neurônios se enganarem é bem menor. Esta é a forma como um rastreamento é realizado quando combinadas às ferramentas Redes Neurais Kohonen e Filtro de *Kalman*.

3.1. Implementando Redes Neurais 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: $(y_j^k = +1)$. Mas o que acontece com os outros neurônios da rede? A curva padrão de treinamento da rede *Kohonen winner-takes-all* é a mostrada na Figura 4. Nesta curva, o neurônio vencedor recebe um treinamento e os outros também recebem, de acordo com a distância do neurônio vencedor. A curva utilizada neste projeto é representada na Figura 5. Nesta curva, o neurônio vencedor recebe um treinamento e os demais recebem um treinamento contrário ao vencedor, de forma a se afastar do mesmo.

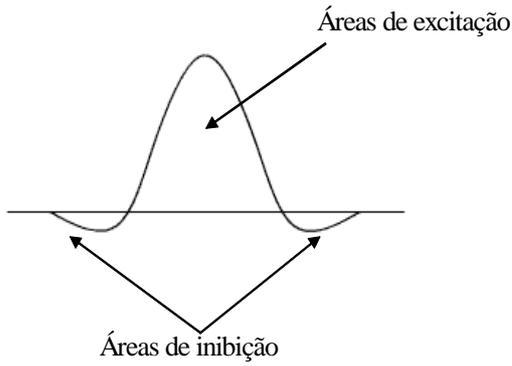


Figura 4 - Curva padrão de treinamento da rede neural Kohonen.

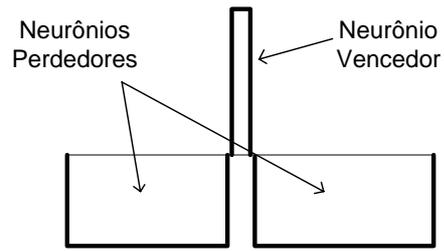


Figura 5 - Curva de treinamento da rede neural Kohonen utilizada neste projeto.

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 tridimensionais, então: $x^k = [x_1^k, x_2^k, x_3^k]$. 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 aproximá-lo das entradas com a seguinte equação:

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

sendo: α o Passo de Treinamento da Rede Neural (*Learning Rate*).

A rede neural é treinada desta maneira para cada padrão de entrada, ou seja para cada coordenada das posições do alvo, realizando assim o rastreamento.

A função do Filtro de *Kalman* neste projeto é estimar a posição futura da rede neural baseando-se em estados anteriores (Figura 6) e fazer com que o rastreamento seja corrigido pelo filtro.

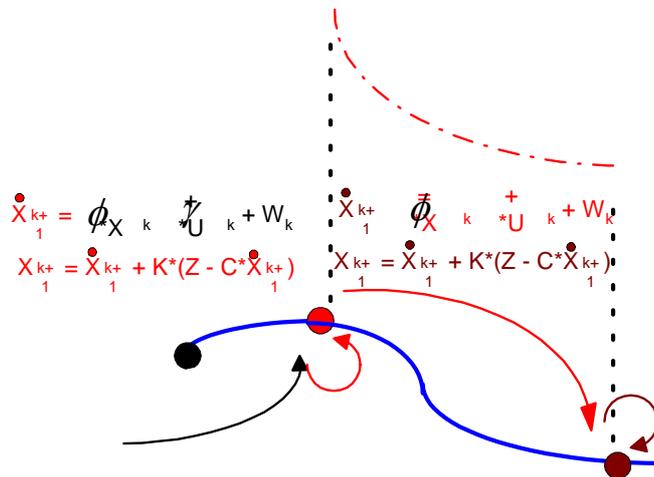


Figura 6 – Predição e atualização dos estados no Filtro de Kalman.

Visto que o Filtro de *Kalman* realiza a estimativa em duas fases , as equações do filtro são:

Predição ou Propagação no Tempo

$$\hat{X}_{k+1} = \text{PHI} * \hat{X}_k + \text{GAMMA} * U_k + W_k \quad (3)$$

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

Sendo: X_k é o vetor que contém os estados anteriores

X'_{k+1} é o vetor que contém os estados estimados *a priori*

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

Atualização ou Correção da Estimativa

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

$$\hat{X}_{k+1} = \hat{X}'_{k+1} + K * (Z - C * \hat{X}'_{k+1}) \quad (6)$$

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

sendo: K é o ganho de Kalman

Z é o vetor que contém as medidas reais 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 opcional.

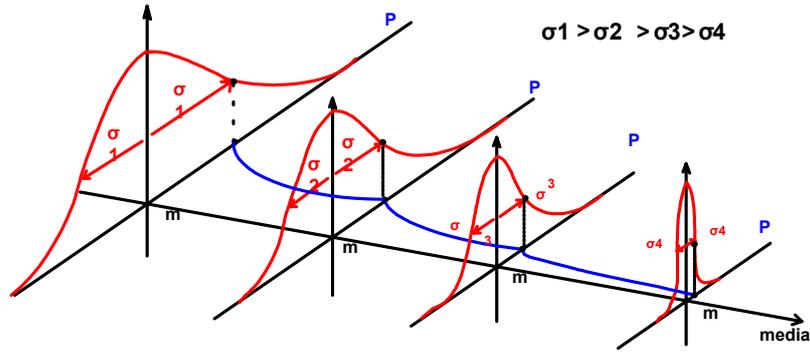


Figura 7 - Convergência da Matriz de Covariância, P.

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 $x^{tk} = M^{-1/2}[x_1^k, x_2^k, \dots, x_n^k, x_{n+1}^k]$, 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, o neurônio vencedor é treinado de forma a se aproximar do alvo, e os demais se afastam. O parâmetro que define o quanto o neurônio vai se aproximar do valor de entrada é o Learning Rate. O Filtro de Kalman é aplicado a cada passo de treinamento do neurônio na rede neural. O fluxograma do algoritmo para um melhor entendimento do processo é mostrado na Figura 8:

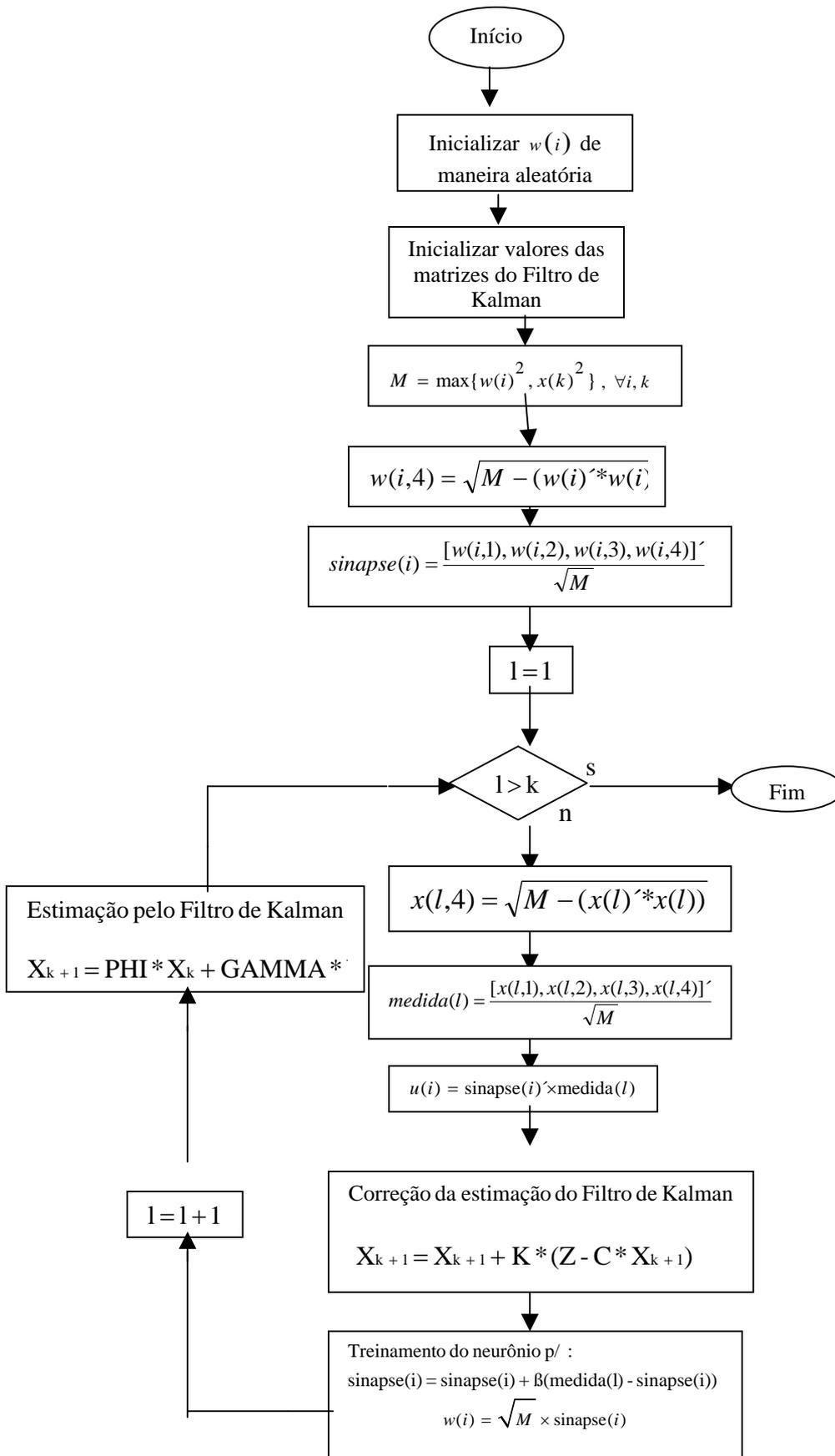


Figura 8 - Fluxograma do algoritmo de Kohonen e Filtro de Kalman.

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. A intenção destes pontos é simular o caso em que medidas são retiradas de sensores elétricos de posição, e representadas por estes pontos.

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 9 mostra pontos azuis como os valores dos padrões de entrada, denominada aqui como nuvem de pontos, o ponto original está em vermelho, e as posições iniciais dos neurônios foram representados como pequenas estrelas. A Figura 10 e a Figura 11 mostram os padrões de entrada, ponto sobre a curva e posições iniciais dos neurônios como na Figura 9, porém com trajetórias simuladas pelas curvas Logaritmo e Seno, respectivamente.

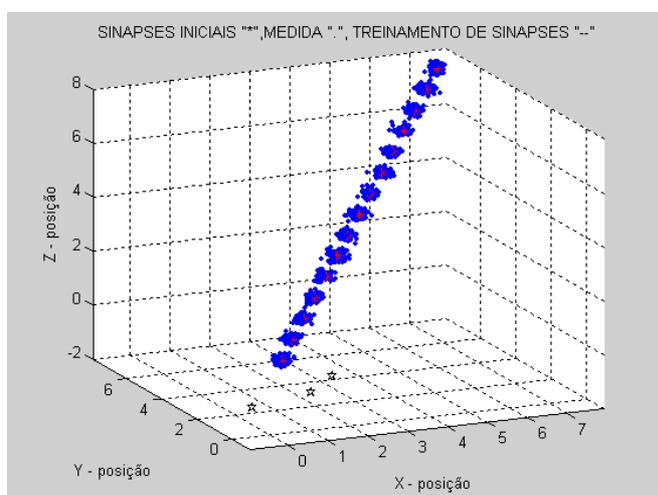


Figura 9 - Nuvem de pontos e posições iniciais dos neurônios na Reta.

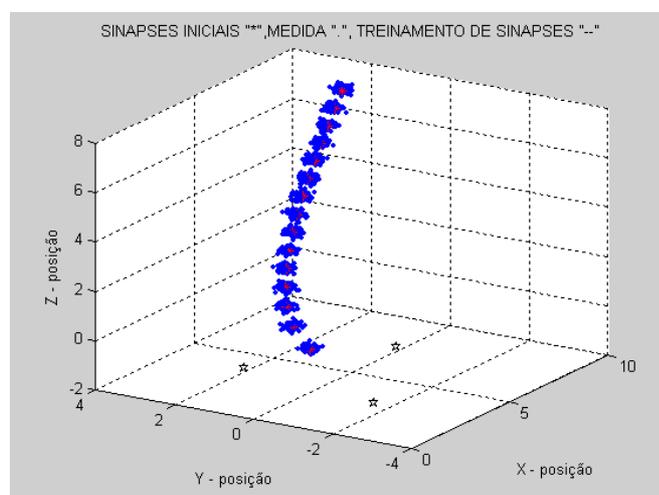


Figura 10 - Nuvem de pontos e posições iniciais dos neurônios no Logaritmo.

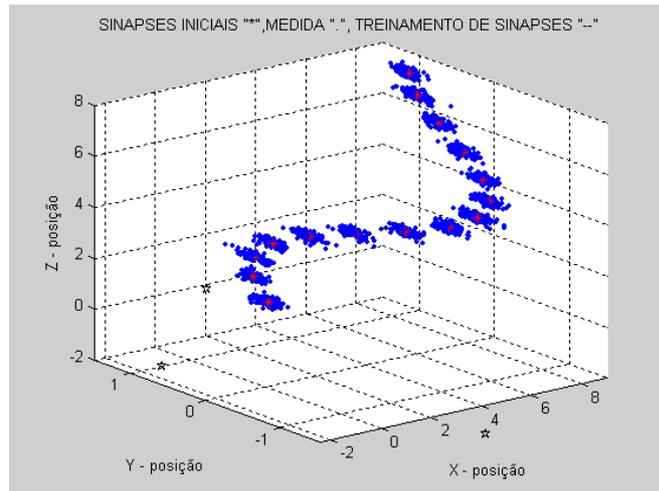


Figura 11 - Nuvem de pontos e posições iniciais dos neurônios no Seno.

Os passos para o treinamento da rede neural com o filtro de Kalman estão descritos abaixo:

1. São estabelecidas condições iniciais para os pesos dos neurônios w^i de forma aleatória
2. É estabelecido um quociente de aprendizado α para os pesos. Inicialmente, foi definido $\alpha=0,01$. Posteriormente este parâmetro foi sendo variado, para analisar sua influência no treinamento da rede;
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. Normalizam-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:

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

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 RESULTADOS E DISCUSSÕES

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 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 estimativa 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 estimativa da posição do neurônio vencedor pelo Filtro de Kalman. Serão simuladas três trajetórias distintas. As trajetórias são as descritas na seção anterior, com as equações das seguintes curvas: Reta, Logaritmo e Seno, como mostrado na Figura 9, na Figura 10, e na Figura 11, respectivamente.

4.1 – Rastreamento do Objeto Sobre a Reta

A primeira simulação de rastreamento feita foi a do objeto alvo se movendo com uma trajetória segundo uma reta. A Figura 9 mostra a evolução dos pontos sobre a trajetória. Foram gerados os pontos para cada baricentro sobre a reta, para que fique explícita a trajetória no qual o objeto alvo se desloca. Em torno destes baricentros foram criados conjuntos de pontos para representar medidas retiradas de sensores elétricos de posição. A cada passo, o parâmetro “T” 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:

$$X(T) = T \qquad Y(T) = 2 * T \qquad Z(T) = T \qquad (11)$$

Foram utilizadas funções vetoriais onde “T” é um parâmetro, que aqui representa o tempo.

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 estimar as coordenadas da posição futura da rede neural.

A Figura 12 mostra o resultado do rastreamento. As estrelas pretas representam as condições iniciais das sinapses, os pontos em vermelho 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. Pode ser observado que, no início, o filtro de Kalman possui estimativas com um certo grau de erro mas, conforme é aplicado, ele converge para a rede neural.

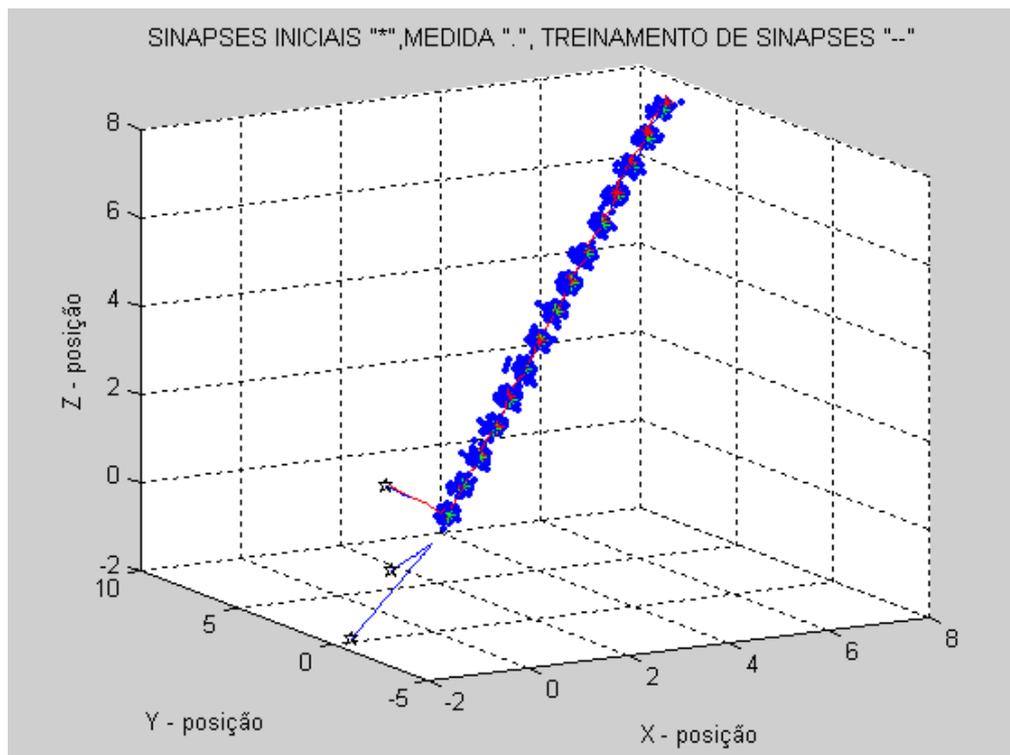


Figura 12 - Resultado do Rastreamento com Trajetória Retilínea.

4.2 – Rastreamento do Objeto com Trajetória Logarítmica

Nesta simulação, o rastreamento será a do objeto alvo se movendo com uma trajetória segundo uma função logarítmica. A Figura 6 mostra a evolução dos pontos sobre a trajetória. Foram gerados 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 baricentro foi incrementado para que os pontos pudessem se deslocar sobre a função logarítmica. O baricentro foi atualizado de acordo com a seguinte função:

$$X(T) = T \quad Y(T) = \text{LN}(T) \quad Z(T) = T \quad (12)$$

A Figura 13 mostra o resultado da validação do algoritmo, e, a partir da mesma podemos analisar o comportamento do sistema.

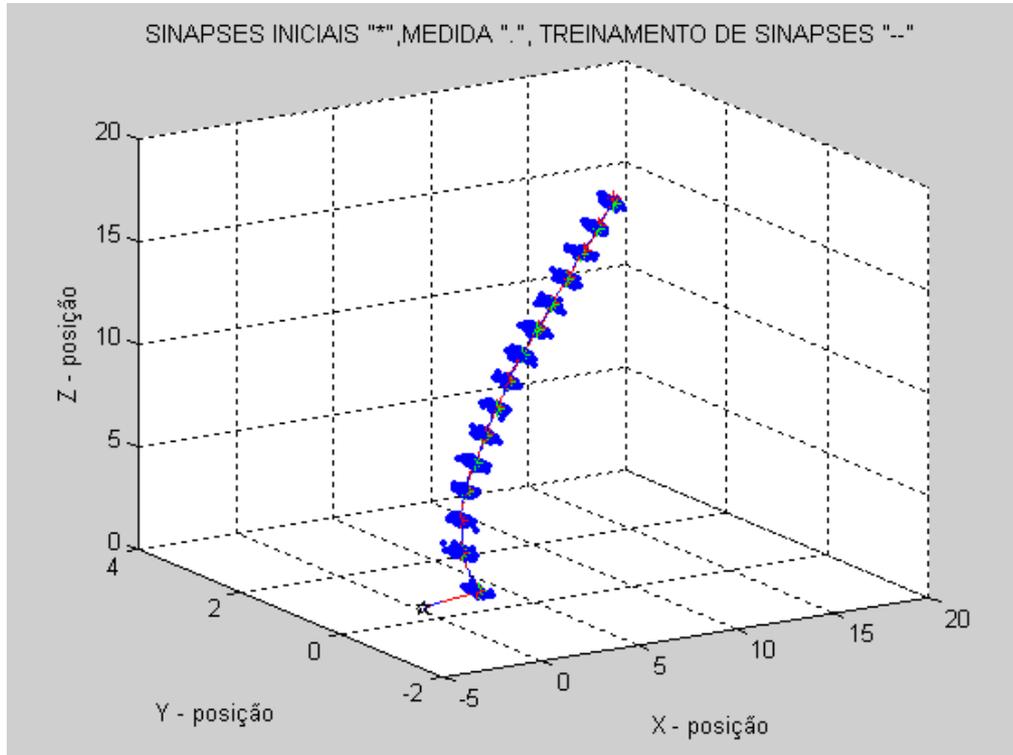


Figura 13 - Resultado do Algoritmo com Trajetória Logarítmica.

Pela figura, vemos que tanto a rede neural quanto o Filtro de Kalman se comportam como havíamos previsto. O fato de a trajetória utilizada ser curva pode prejudicar um pouco o desempenho do filtro de Kalman, mas como neste caso a curva é suave, o filtro obteve um bom comportamento, convergindo para a rede neural.

4.3 – Rastreamento do Objeto com Trajetória Senoidal

Nesta simulação, o rastreamento será a do objeto alvo se movendo com uma trajetória segundo uma função senoidal. A Figura 10 mostra a evolução dos pontos sobre a trajetória desta simulação. Observe que esta figura foi gerada como a anterior, de forma que as mesmas características se aplicam aqui, exceto a forma da trajetória. A função utilizada foi a seguinte:

$$X(T) = T \quad Y(T) = \text{Sen}(T) \quad Z(T) = T \quad (13)$$

A Figura 14 mostra os resultados do sistema. Apesar de esta ter sido uma trajetória com grandes variações de direção, a rede neural e o Filtro de Kalman obtiveram um desempenho satisfatório. Pela figura podemos observar que o filtro perde um pouco a convergência nas regiões com grandes variações de direção, ou seja, nos chamados extremos da função. Já nas partes de subida e descida, onde a trajetória é aproximadamente retilínea, o filtro obtém um bom desempenho.

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.

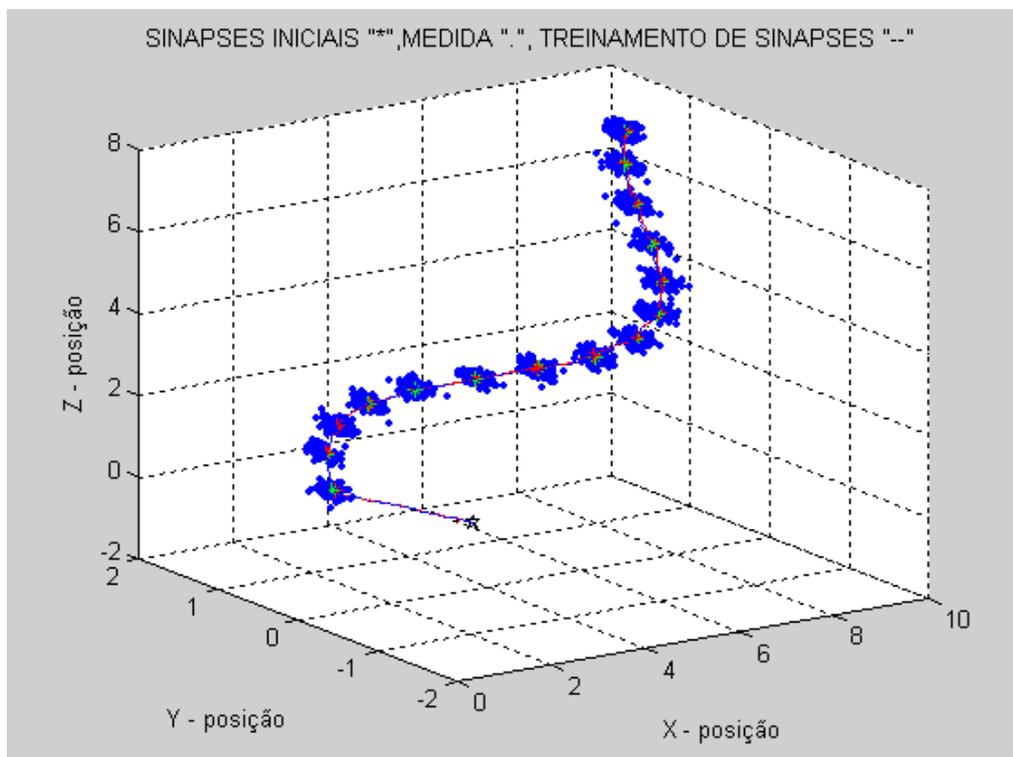


Figura 14 - Resultado do Algoritmo com Trajetória Senoidal.

4.4 – Rastreamento de Múltiplos Objetos-alvo

Nos três exemplos anteriores, o algoritmo obteve bons resultados. Todavia, 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 diferentes funções vetoriais. 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:

$$X1(T) = 1 + T \quad Y1(T) = 2 * \text{sen}(T) + 1 \quad Z1(T) = T \quad (14)$$

$$X2(T) = T \quad Y2(T) = 0,3 * T \quad Z2(T) = T \quad (15)$$

A evolução dos baricentros e das nuvens é mostrada na Figura 15. É mostrada a posição de cada objeto em todos os instantes de tempo para que se possa observar as trajetórias descritas por eles. Aqui o filtro de Kalman foi aplicado somente para convergir para a rede, e não foi utilizado como auxílio para a rede neural. A idéia é ver o que acontece nos cruzamentos quando o rastreamento é feito somente com a rede neural.

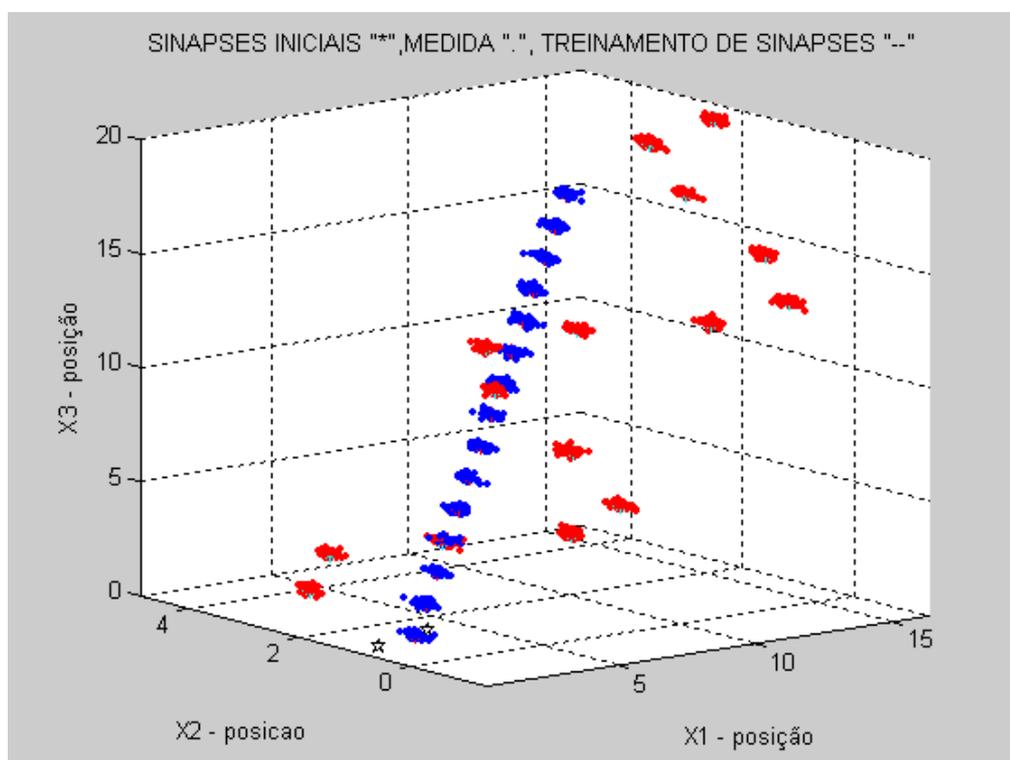


Figura 15 - Dois Objetos com Trajetórias Diferentes.

O resultado obtido está na Figura 16. 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 alvo, enquanto o mesmo acontece com o outro, ou seja, há uma troca no rastreamento. Com esta troca, não conseguimos identificar as trajetórias dos dois alvos.

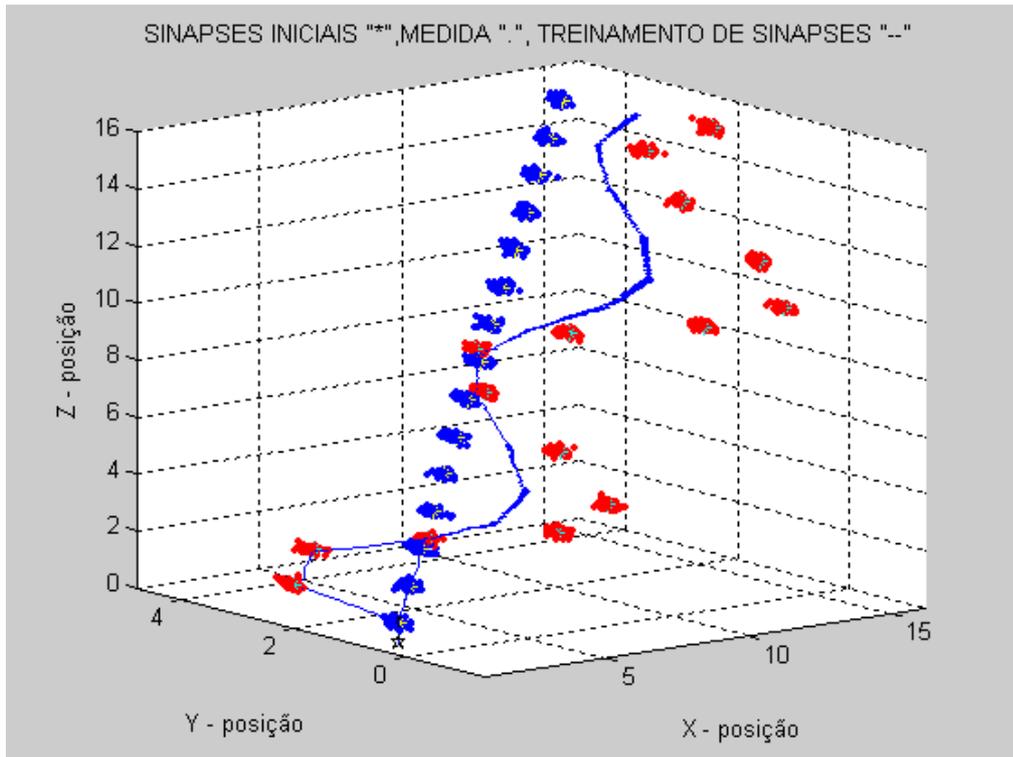


Figura 16 - Rastreamento Mal Sucedido com Dois Objetos.

O mesmo processo foi realizado com uma simulação de três alvos. Novamente, a idéia aqui é observar o desempenho da rede neural rastreando sozinha, e a convergência do filtro de Kalman. Nestes casos, as funções vetoriais utilizadas foram as seguintes:

$$X1(T) = 1 + T \quad Y1(T) = 2 * \text{sen}(T) + 1 \quad Z1(T) = T \quad (16)$$

$$X2(T) = T \quad Y2(T) = 0,3 * T \quad Z2(T) = T \quad (17)$$

$$X3(T) = T + 8 \quad Y3(T) = 2 * \log(T) - 1,5 \quad Z(T) = T \quad (18)$$

A evolução dos baricentros e das nuvens é mostrada na Figura 17. Foram mostrados os três alvos se deslocando.

Como pode ser observado na Figura 18, o treinamento dos neurônios vencedores não foi bom. Houve troca de neurônios rastreadores, e não conseguimos identificar a trajetória dos alvos. O motivo é o mesmo do caso anterior.

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.

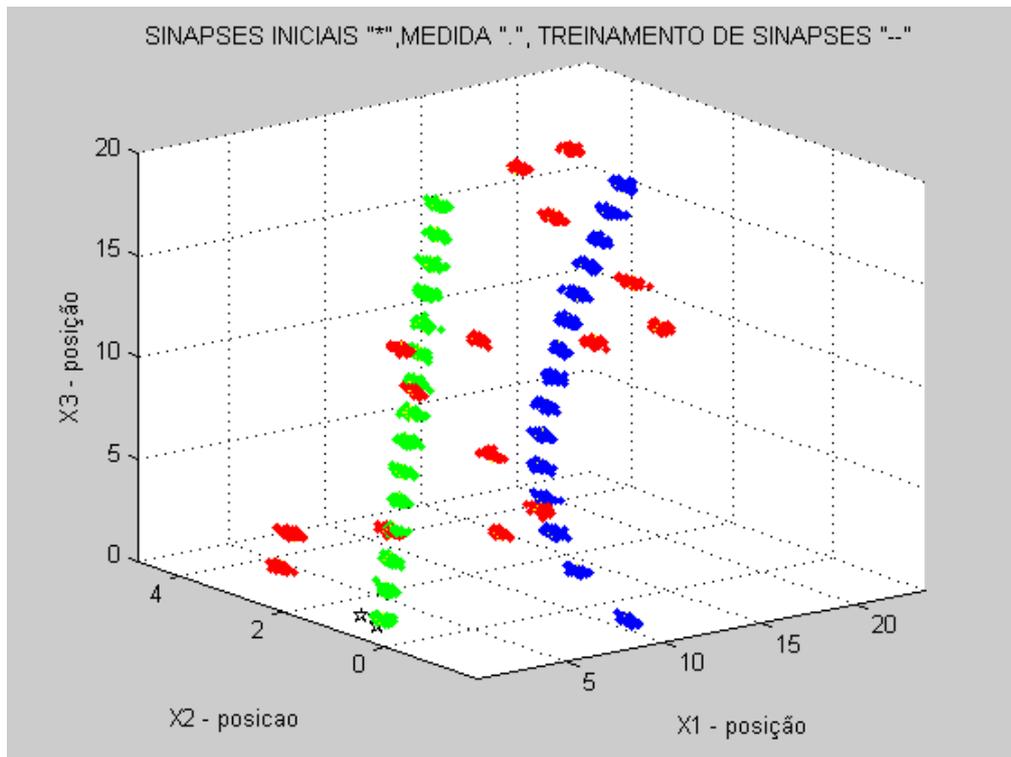


Figura 17 - Três Objetos com Trajetórias Diferentes.

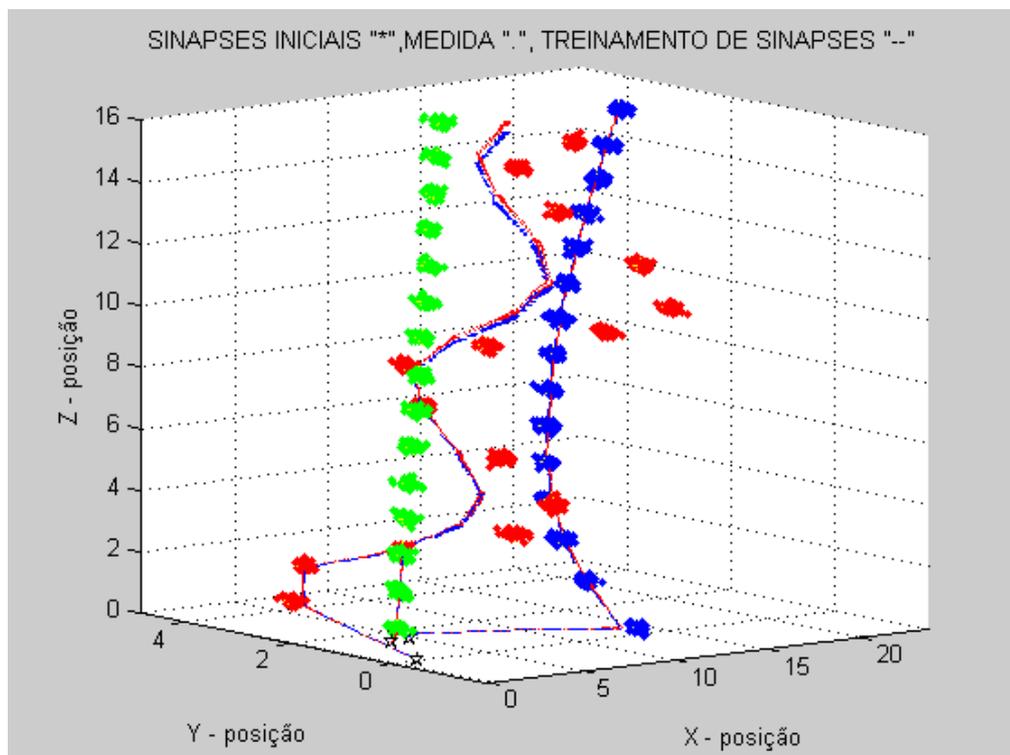


Figura 18 - Rastreamento Mal Sucedido de Três Objetos.

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.

Como o filtro de Kalman realizou a convergência desejada, podemos aplicá-lo como um dos parâmetros de treinamento para a rede neural. Aqui, os estados futuros estimados pelo filtro servem como alguns dos padrões de entrada para o treinamento da rede neural. A seguir analisamos o efeito do filtro de Kalman na rede neural nos casos de dois e três alvos que geraram os erros mostrados anteriormente. Os mesmos rastreamentos de dois e três objetos mostrados anteriormente são mostrados agora com o filtro de Kalman nas Figura 19 e Figura 20.

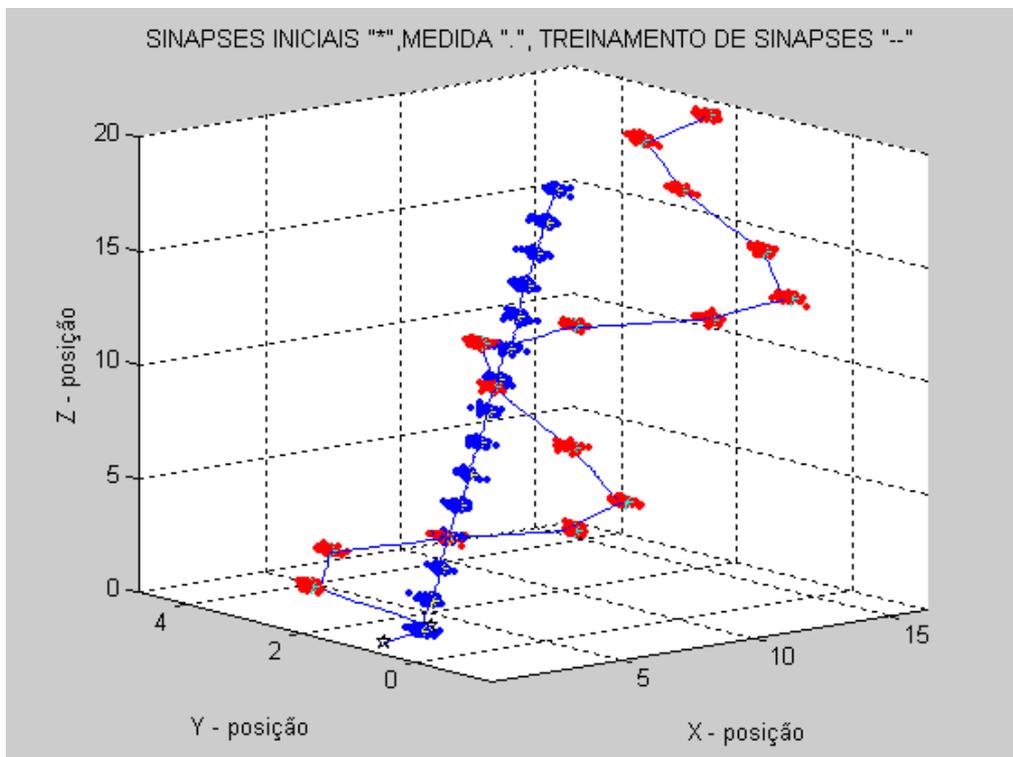


Figura 19 - Rastreamento Bem sucedido de Dois Objetos.

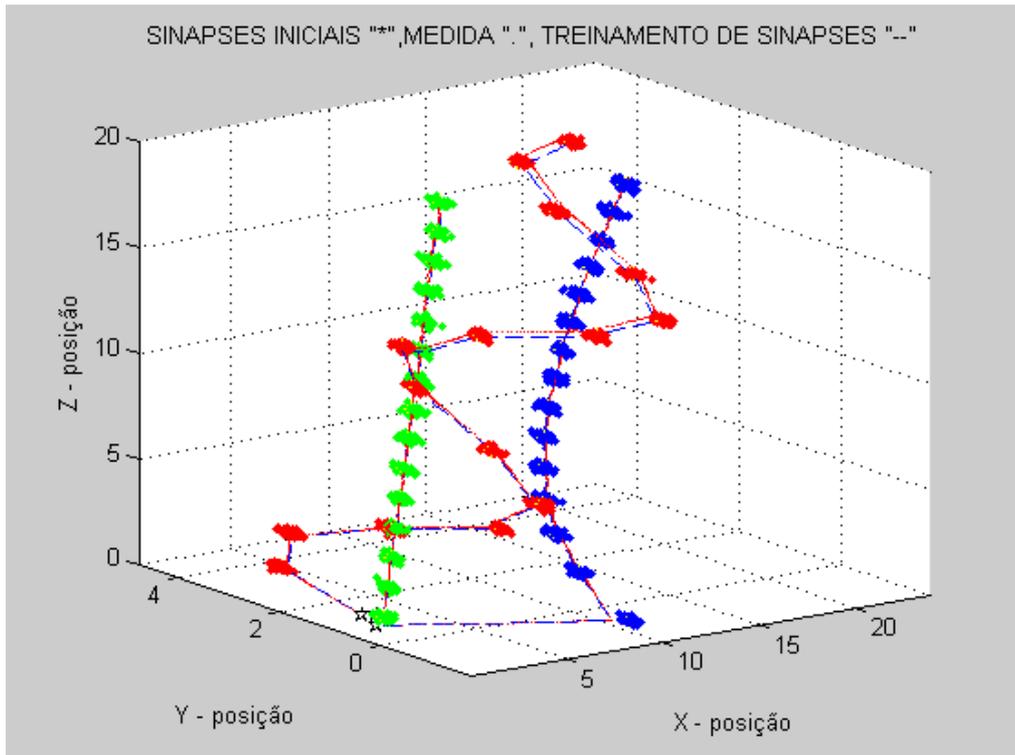


Figura 20 - Rastreamento Bem sucedido de Três Objetos.

5 5– CONCLUSÕES

Este projeto tem por objetivo estudar e desenvolver um algoritmo para rastreamento de alvos móveis utilizando redes neurais Kohonen e o Filtro de Kalman.

Foi realizado um estudo sobre redes neurais, os tipos existentes, formas de treinamento, aplicações, e foi pesquisado em artigos, o que já existe de projetos sobre o assunto com esta abordagem.

Foram realizadas também pesquisas sobre o Filtro de Kalman e sua utilização, para assim poder-se implementar a rede neural em conjunto com o filtro de Kalman para obter um algoritmo rápido e eficiente para satisfazer os seus fins.

Foram simulados casos com somente um alvo móvel, e foi observado que a rede neural realizava o rastreamento, enquanto o filtro de Kalman realizava as estimativas das coordenadas da posição futura da rede neural e convergia para a mesma. Em casos simulados com mais de um alvo móvel observou-se que, nos cruzamentos das trajetórias, ocorriam trocas de neurônios rastreadores, o que prejudicava o rastreamento.

Após isto, as estimativas do Filtro de Kalman foram utilizadas como alguns dos padrões de entrada para a rede neural, com a intenção de corrigir o rastreamento nos cruzamentos dos alvos. Através da análise dos resultados das simulações, foi observado que, com o uso do filtro de Kalman, os problemas de rastreamento não ocorriam, sendo que a rede neural conseguia realizar o rastreamento mesmo quando na presença de mais de um alvo móvel.

Outro parâmetro importante que foi observado na rede neural, é a taxa de aprendizagem, ou *Learning Rate*. Testes foram realizados com diversos valores para a mesma situação de rastreamento. Foi observado que, com valores muito baixos, como 0,01, o rastreamento fica ruim, pois o passo dado pela rede é muito pequeno, então o neurônio mal consegue alcançar o alvo. Quando aumentamos o valor da taxa de aprendizagem, o rastreamento melhora, sendo que o passo dado pelo neurônio é maior e ele consegue alcançar o alvo com facilidade.

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, pode ser feita com variações nas pesquisas e nas implementações. Seguem-se aqui algumas sugestões:

1. 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. O Filtro de Kalman Estendido trata de sistemas não lineares, o que é interessante nos casos de alvos com trajetórias com grandes variações de direção.
2. O estudo e implementação de tipos diferentes de redes neurais, como *back-propagation*, perceptron, e outras para testar seus desempenhos para o problema de rastreamento.
3. Utilização de diferentes técnicas de inteligência artificial, como a lógica *fuzzy*, combinadas com a rede neural para realizar o rastreamento.

6 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.
- [43] Willet, D., Busch, C., Siebert, F., - **Fast Image Analysis Using Kohonen Maps**, Darmstadt Computer Graphics Center.
- [44] Feldkamp, L. A., Prokhorov, D. V., Eagen, C. F., Yuan, F. – **Enhanced Multi-Stream Kalman Filter Training for Recurrent Networks**, For Research Laboratory, Michigan, USA.
- [45] Cheng, H.D., Jiang, X.H., Sum, Y, Wang, J. – **Color image segmentation: advances and prospects**, Department of Computer Science, Utah State University, Logan, USA.
- [46] Cho, Sung-Bae – **Pattern Recognition With Neural Networks Combined by Genetic Algorithm**, Department of Computer Science, Yonsei University, Seoul, South Korea.
- [47] Srihari, S. N., **Representation of Three-Dimensional Digital Images**, Department of Computer Science, State University of New York at Buffalo.
- [48] Srinivasan, R., Kinser, J. M. – **A Foveating Fuzzy Scoring Target Recognition System**, Center for Applied Optical Sciences, Department of Physics, Alabama A&M University.
- [49] Brijnesh, J. Jain, Wysotzki, Fritz, **Competitive Winner-Takes-All Clustering in the Domain of Graphs**, Department of Computer Science, Technical University of Berlin, Germany.
- [50] Vaidehi, V., Chitra, N., Chokkalingam, M., Krishnan, C.N. – **Neural Network Aided Kalman Filtering for multitarget target applications**, School of Instrumentation and Electronics, Madras Institute of Tecnologi, India. 20 May 1999.