

# ESTRATÉGIAS DE ACOMPANHAMENTO DE MÚLTIPLOS ALVOS: UMA COMPARAÇÃO ENTRE AS ABORDAGENS DE ASSOCIAÇÃO DE DADOS MHT E REDES NEURAIIS

**Manuel R. Vargas A.**

Universidade Federal de Rio de Janeiro  
manuelvargas760@gmail.com

**Cachimo C. Assane**

Universidade Federal de Rio de Janeiro  
cachimo.assane@gmail.com

**Rubens L. de Oliveira**

Instituto de Pesquisas da Marinha  
rubensliv@yahoo.com

**Beatriz P. de Lima**

Universidade Federal de Rio de Janeiro  
bia@coc.ufrj.br

## Resumo:

Neste trabalho desenvolvemos uma estratégia de acompanhamento de múltiplos alvos que trafegam em um ambiente marítimo a partir de informações provenientes de um único sensor radar. Utilizamos Redes Neurais artificiais como metodologia de associação de dados e comparamos o seu desempenho com o da abordagem clássica baseada no algoritmo de associação de dados MHT. A metodologia proposta permitiu a redução do tempo de execução do algoritmo comparando com a abordagem clássica.

Quando em alguns cenários são introduzidas perdas de detecção, a metodologia clássica apresenta dificuldades em manter o acompanhamento de alguns alvos, o que pode ser considerado um dos seus pontos fracos. Em relação ao número de falsos acompanhamentos criados, a metodologia clássica apresenta um número maior em relação à abordagem proposta, provavelmente devido à sua complexidade na estratégia de busca (combinatória) e o processo de inicialização.

**Palavras-Chaves:** Algoritmos de acompanhamento, algoritmos de associação de dados, algoritmo MHT, Redes Neurais Artificiais, filtragem.

## Abstract:

In this paper, we develop a multiple target tracking strategy for tracking targets in a maritime environment from one sensor (radar) measurement data. We use the Artificial Neural Network as a data association method and we compare its performance with the classical approach based on Multiple Hypothesis Tracking (MHT) algorithm. The proposed algorithm execution time is smaller than the execution time MHT algorithm. When in some scenarios we have considered some no-detections, the conventional method presents difficulties to maintain some tracking targets. This fact may be considered one of their weaknesses. Regarding the number of existing

false tracks, the classical method indicates a number larger than the proposed approach, probably because of its complexity in the combinatorial search strategy and its initialization process.

**Keywords:** Tracking algorithms; data association algorithms; MTH algorithm; Artificial Neural Networks; filtering.

# 1 Introdução

Os algoritmos de acompanhamento de múltiplos alvos (*Multiple Target Tracking-MTT*) são cada vez mais utilizados em sistemas de navegação e vigilância civis e militares, permitindo a identificação e rastreamento de ameaças em tempo real. Entretanto, acompanhar um alvo que trafega num ambiente com alta densidade de *clutter* ou falsos alarmes é um processo complexo que exige algoritmos de associação de dados sofisticados capazes de lidar com um grande volume de dados. Diversas técnicas alternativas eficientes têm sido desenvolvidas para resolver esta problemática, especialmente em aplicações militares como o controle tático e de armamentos.

No contexto deste trabalho, entenda-se por contato, medida ou observação qualquer ponto fornecido pelo sensor cuja fonte pode ser um alvo ou *clutter*. Define-se por alvo qualquer objeto de interesse e *clutter* um contato gerado devido às condições do mar, objetos como boias ou erros internos do sensor entre outros fatores. *Tracks* ou acompanhamentos referem-se ao conjunto de parâmetros ou medidas que podem representar o movimento de um alvo. Falsos acompanhamentos referem-se a acompanhamentos compostos em sua maioria por medidas geradas por *clutter*. *Scan* refere-se à varredura completa da antena do radar. Finalmente, *tracking* ou rastreamento é o processo de estimação dos parâmetros da cinemática do alvo de interesse.

O processo de associação de dados é a etapa primordial de um sistema de acompanhamento de múltiplos alvos. Dele depende a correta atualização do *track* e o número de falsos acompanhamentos criados. A presença de *clutter* dificulta a associação dos dados na medida em que se torna praticamente impossível decidir, com certeza, se um contato se refere ao alvo de interesse ou simplesmente é *clutter*. Associar estas observações espúrias (*clutter*) ao alvo de interesse pode resultar na perda do acompanhamento. Além disso, a alta densidade de *clutters* pode produzir maior quantidade de falsos acompanhamentos, comprometendo o desempenho computacional dos algoritmos.

O algoritmo clássico de associação amplamente utilizado no MTT é o *Multi Hypothesis Tracking* (MHT), o qual se baseia na criação de hipóteses para obter o caminho mais provável do alvo. No entanto, perante um cenário com alta densidade de *clutters*, o número de hipóteses pode crescer significativamente, aumentando o custo computacional deste algoritmo. Como alternativa, neste trabalho é proposta uma abordagem baseada em Redes Neurais visando contornar as dificuldades (de otimização combinatória) associadas ao algoritmo MHT.

As redes neurais são popularmente conhecidas pela recursividade dos seus modelos, seu baixo custo computacional após o seu treinamento, capacidade de se adaptar às mudanças no ambiente, robustez na presença de dados ruidosos e potencial para sua computação massiva paralela [1, 2]. Por estas razões esta ferramenta é recomendada para aplicações que requerem resultados em tempo real. Algumas aplicações

de Redes Neurais em problemas de associação de dados são apresentados em [3, 4, 5] e [6].

Após associar uma nova medida ao alvo em acompanhamento e tendo em conta às incertezas em relação à origem desta medida, tanto o algoritmo MHT quanto a abordagem de Redes Neurais precisam de uma etapa de filtragem como forma de estimar a trajetória “real” do alvo. Uma ferramenta amplamente utilizada para tratar da etapa de filtragem é o algoritmo de filtro de Kalman, que parte dos pressupostos da linearidade e normalidade dos modelos de evolução de estado e de medição [7].

Na prática, a maioria dos modelos não são lineares pelo que o uso de filtro de Kalman estendido é necessário. Além disso, dada a possibilidade de um alvo experimentar diferentes padrões de comportamento ao longo do tempo, a implementação de um único filtro de Kalman se torna ineficiente. Uma abordagem alternativa eficaz é o estimador Múltiplos Modelos Integrantes (IMM) [8, 9] cuja vantagem reside no fato de permitir a utilização de mais de um modelo de KF em paralelo.

O objetivo deste trabalho é desenvolver uma estratégia de acompanhamento de múltiplos alvos a partir de informações provenientes de um único sensor radar em ambientes marítimos usando Redes Neurais artificiais como metodologia de associação de dados e comparar o seu desempenho com o da metodologia clássica baseada no algoritmo de associação de dados MHT. Pretende-se também estudar diferentes algoritmos de treinamento, visando a encontrar aquele que apresente a melhor solução para o nosso problema.

## 2 Método clássico

O método clássico de acompanhamento de múltiplos alvos, pode ser decomposto em dois processos fundamentais:

- **Inicialização:**

Criação do track preliminar:

Uma medida não associada a um *track* existente é considerada como um “iniciador”, usado para produzir uma “tentativa de *track*”. Na varredura, após a detecção de um iniciador, é construída entorno dele uma janela de associação circular com limites estabelecidos por

$$D.min = (V_{min} * t_r) - (3 * \sigma_d) \quad (1)$$

$$D.max = (V_{max} * t_r) + (3 * \sigma_d), \quad (2)$$

onde  $D.min$  e  $D.max$  são o alcance mínimo e máximo da janela respectivamente,  $V_{min}$  e  $V_{max}$  representam as velocidades mínima e máxima definidas para o tipo de alvo de interesse,  $t_r$  é o tempo de rotação de antena do Radar e  $\sigma_d$  é o desvio padrão do ruído de medição na distância.

Se existe uma nova medida dentro dessa janela o conjunto dos dois contatos formam um “*track* preliminar”. Caso não haja alguma medida que possa ser associada, a tentativa de *track* é descartada.

Atualização dos *tracks* preliminares:

Para isto utiliza-se num primeiro momento um filtro de kalman padrão com modelo de cinemática retilíneo uniforme, para prever a posição futura do alvo

em acompanhamento (*track* preliminar), seguidamente, com centro na posição prevista pelo filtro é criada uma janela de precorreção utilizando as equações 1 e 2. Finalmente, no caso de houver mais de um contato nesta janela é empregado o método de associação pelo vizinho mais próximo, o critério considerado neste caso é a distancia euclidiana.

Confirmação ou eliminação do *track*:

A partir da terceira varredura, a lógica de  $M$  detecções em  $N$  scans é utilizada para decidir se o *track* preliminar é confirmado ou eliminado. Esta lógica é nomeada critério  $M/N$ . Se no final das  $N + 2$  varreduras o critério lógico for satisfeito, o *track* preliminar é agora chamado por “*track* confirmado”, de outra maneira o *track* é eliminado. Esta lógica é executada como apresentado em [8, 10]. Para o nosso caso, segundo a densidade de clutter na área de vigilância, os parâmetros  $M$  e  $N$  podem assumir valores inteiros entre 3 e 9.

- **Manutenção:**

O primeiro passo deste processo é o de prever a posição futura do alvo para isso utiliza-se o algoritmo de filtragem IMM o qual também é utilizado para fazer o alisamento do acompanhamento. Entorno da posição prevista é criada uma janela de precorreção utilizando as equações 1 e 2. Finalmente, como algoritmo de associação é empregado o MHT orientado a tracks (Track-oriented MHT, TOMHT). A descrição deste algoritmo é apresentada em [11] e [12].

### 3 Metodologia

O método proposto consiste basicamente em três etapas: na primeira, os acompanhamentos preliminares são criados, em seguida, ao receber novas medidas, os mesmos podem ser atualizados mediante uma rede neural e por último um processo de filtragem é aplicado com o objetivo de encontrar a aproximação da trajetória real do alvo que é de interesse para o usuário do sistema. O esquema funcional do método desenvolvido neste trabalho é apresentado na figura 1

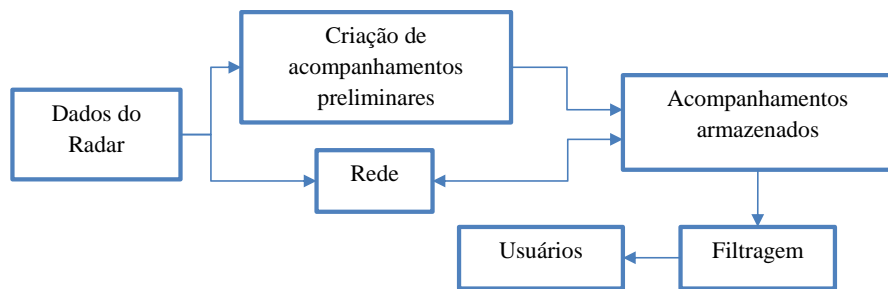


Figura 1: Método Proposto

Os dados utilizados para o treinamento da rede neural e a validação do método proposto, assim como as etapas que compõem este método e as métricas utilizadas para compara-lo com o método clássico são apresentados a seguir.

### 3.1 Medidas do sistema Radar

Neste trabalho é utilizado o sistema Radar de pulsos de busca de superfície, um dos mais populares na navegação marítima. Este sistema fornece uma medição da localização de um objeto em coordenadas polares a cada instante  $k$  de tempo discreto. O tempo entre as medições é uma variável que depende da velocidade, direção do navio e do tempo de rotação de antena [13]. Esta medição é perturbada por ruídos assumidos independentes e estacionários com distribuição gaussiana de média zero e variâncias constantes [14]. Assim, esta medição é representada em forma de vetor

$$Z(k) = \begin{bmatrix} d(k) \\ \theta(k) \end{bmatrix} = \begin{bmatrix} d_r(k) \\ \theta_r(k) \end{bmatrix} + e(k) = \begin{bmatrix} d_r(k) \\ \theta_r(k) \end{bmatrix} + \begin{bmatrix} e_d(k) \\ e_\theta(k) \end{bmatrix}, \quad (3)$$

onde  $d(k)$  e  $d_r(k)$  são respectivamente as distâncias medida e real (em metros) do alvo ao observador,  $\theta(k)$  e  $\theta_r(k)$  representam o azimuth medido e real (em graus) com respeito ao norte geográfico,  $e_d(k)$  é o ruído na distância com variância  $\sigma_d^2$  e  $e_\theta(k)$  é o ruído no azimuth com variância  $\sigma_\theta^2$ .

### 3.2 Geração de dados

Uma vez que não temos à disposição dados provenientes do sensor radar para realizar o treinamento da rede neural e validação do método proposto, foi necessário utilizar um gerador artificial de dados. Assim, foram empregados dois programas nomeadamente gerador de trajetória de alvos e gerador de cenários. Estes programas foram fornecidos pelo Instituto de Pesquisa da Marinha (IPqM) apenas para fins de realização deste trabalho.

O gerador de trajetória permite criar trajetórias coerentes de alvos com diferentes padrões de movimentos. É importante destacar que, inicialmente, as trajetórias aqui geradas não apresentam ruído de processo ou de medição. Para o nosso caso, foram geradas trajetórias com uma velocidade máxima de 200 m/seg, velocidade mínima de 70 m/seg, aceleração tangencial máxima de 5 m/seg<sup>2</sup> e taxa de guinada máxima de 5 grau/seg.

No programa gerador de cenários são acrescentados tanto os ruídos de processo como os de medição. O ruído de processo é visto como uma perturbação na aceleração e supõem-se que tem média zero e covariância constante. Em relação ao ruído de medição, o programa permite ajustar o desvio da distribuição dos ruídos segundo as características do sensor que se quer modelar. Além disso, permite definir parâmetros como alcances máximo e mínimo do Radar, a quantidade de clutter por varredura, tempo entre varreduras, número de sensores, número de alvos no cenário, entre outros. Assim, neste trabalho, foram estabelecidos os seguintes parâmetros: um desvio da distribuição dos ruídos de processo de 2 m/seg<sup>2</sup> para as componentes  $x$  e  $y$ , um desvio da distribuição dos ruídos de medição de 30 m na distância e de 3 mrad no azimuth, alcances máximo e mínimo de 30 Km e 5 Km respectivamente, tempo de rotação da antena de 2 seg e um máximo de 100 pontos de clutters por varredura, sendo que foi utilizado apenas um sensor por cenário.

A cada momento em que o programa é executado, um cenário diferente com ruído aleatório é gerado. Como resultado da geração do cenário, é formada uma matriz que contém informações referentes à distância e azimuth de cada um dos contatos contidos no cenário e os seus respectivos tempos de detecção. Adicionalmente, é

construído um vetor que contém informação sobre o tipo de contato, i.e, se o contato é clutter ou se é gerado pelo alvo.

Uma situação a ser levado em conta na geração dos dados é a possibilidade de acontecer perdas de detecção do alvo por algumas varreduras. Assim, para o nosso caso, opta-se por gerar trajetórias com três perdas consecutivas no máximo. A escolha deste número de perdas foi intuitiva pelo fato de que quanto maior o número de perdas consecutivas maior será a incerteza na associação dos contatos.

A partir desses dois programas foram gerados 17 cenários para o treinamento da rede neural, compostos por alvos que descrevem diversas trajetórias visando incorporar grande parte dos tipos de manobras que acontecem na realidade. Para este caso, foram gerados cenários com um único alvo.

A título de exemplo, destacaremos apenas um desses cenários, o qual é apresentado na figura 2(a). Neste cenário considera-se um alvo que descreve um movimento retilíneo com velocidade constante de 99 m/seg durante os primeiros 80 seg, seguido de um movimento curvilíneo com aceleração normal de 5 m/seg<sup>2</sup> (taxa de giro de 2.89 graus/seg) durante 100 / seg e finalmente o alvo descreve um movimento retilíneo com velocidade constante de 99 m/seg durante 80 seg. Além disso, também foram inseridas perdas de detecção nos intervalos de tempo conforme apresentados na tabela 1. Finalmente, para efeito de visualização, na figura 2(b) é apresentada a trajetória resultante deste cenário na sua dimensão ampliada omitindo o clutter.

Tabela 1: Intervalos de tempo e varreduras com perda de detecção no cenário

Tempo (seg)	Varredura
[12 – 18]	7 a 8
[28 – 36]	15 a 17
[46 – 52]	24 a 25
[58 – 62]	30
[70 – 78]	36 a 38
[82 – 88]	42 a 43
[92 – 96]	47
[100 – 108]	51 a 53
[198 – 206]	100 a 102
[228 – 232]	115

Uma etapa fundamental neste trabalho é a validação do método proposto. Para o efeito, foram gerados 6 cenários compostos por múltiplos alvos, os quais incluem situações de alvos navegando próximos um dos outros e apresentando diferentes tipos de manobras, visando incorporar situações frequentes em áreas como portos ou baías. Em jeito de ilustração, apresentamos e descrevemos apenas um dos cenários para validação, o qual é apresentado na figura 3(a).

As quatro trajetórias que compõem o cenário gerado podem ser observadas na figura 3(b). Importa destacar que as trajetórias mostradas nesta figura não são afetadas por ruídos de processo e de medição.

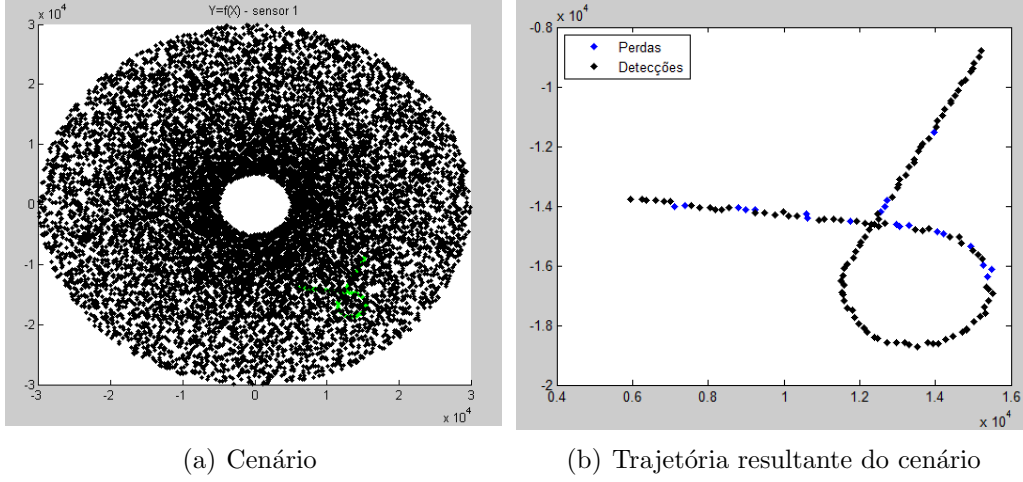


Figura 2: Geração de cenário

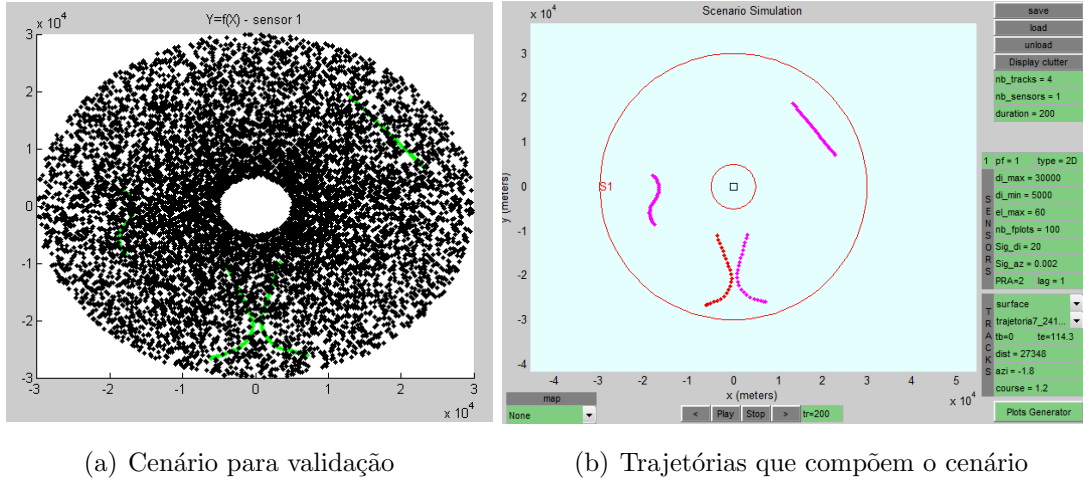


Figura 3: Geração de cenário para validação

### 3.3 Criação dos acompanhamentos preliminares

Nesta etapa é criada uma janela circular entorno de qualquer medida não associada a um acompanhamento existente. Esta janela terá limites estabelecidos por alcances mínimo de 70 metros e máximo de 490 metros. Se na varredura seguinte aparece uma medida dentro da janela criada, é inicializado um acompanhamento. Devido à densidade de clutter, é possível que dentro de uma mesma janela apareçam mais do que uma medida, criando acompanhamentos com medidas em comum, os quais, no decorrer do tempo, muito provavelmente serão excluídos, permanecendo unicamente aqueles que representam um objeto de interesse. Os acompanhamentos criados são armazenados e atualizados nas seguintes varreduras por meio da rede neural.

### 3.4 Rede neural

Tendo em conta os diferentes tipos de movimento que um alvo marítimo pode experimentar e as informações provenientes do sensor Radar, foram seleccionadas como variáveis de entrada,

$$\Delta V(t_k) = V(t_k) - V(t_{k-1}) \quad (4)$$

$$\Delta \alpha(t_k) = \frac{\alpha(t_k) - \alpha(t_{k-1})}{\Delta T} \quad (5)$$

$$\Delta T = t_k - t_{k-1} \quad (6)$$

onde  $V(t_k)$  é a velocidade necessária para percorrer a distancia entre dois contatos,  $\alpha(t_k)$  representa a variação do rumo da trajetória do alvo em graus e  $\Delta T$  é o tempo entre medições. O cálculo destas variáveis é feito para cada um dos contatos disponíveis para associação com um acompanhamento existente. No caso apresentado na figura 4, os contatos 1 a 3 representam um acompanhamento existente e os contatos 4 a 5, gerados na mesma varredura, são aqueles que estão em avaliação para serem associados ao acompanhamento. Só um destes contatos será seleccionado para associação.

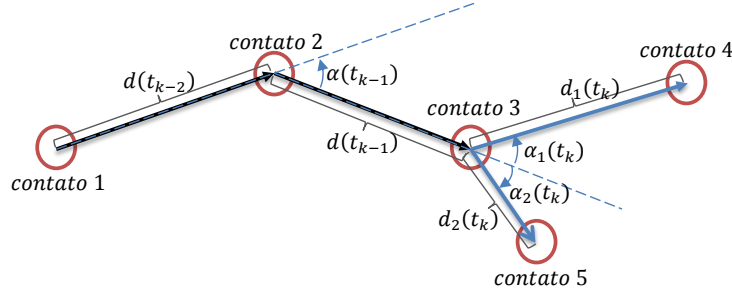


Figura 4: Exemplo do calculo das entradas da RNA

Devido à quantidade maior de contatos gerados numa única varredura e à grande área de vigilância do radar, optamos por dividir a área de vigilância em 16 setores, o que implica uma separação mínima entre dois setores não vizinhos de aproximadamente 1.96 km. Observe que é impossível percorrer esta distância em 8 seg (tempo máximo entre medições considerando as perdas consecutivas) tomando como referência a velocidade máxima dos alvos estabelecida neste trabalho. Portanto, para não testar contatos fora de alcance do alvo que esta sendo acompanhado, definimos uma heurística segundo a qual, todas associações entre acompanhamentos e contatos em setores não vizinhos são ignoradas.

Uma vez seleccionados os contatos em setores vizinhos, calcula-se a velocidade necessária para percorrer a distância entre o acompanhamento (track) e cada um destes contatos. A partir desta informação, são seleccionados candidatos à associação apenas os contatos que tenham velocidade coerente com a descrita pelo alvo. Desta forma, foi estabelecido que um alvo com velocidade superior a 400 m/seg ou inferior a 30 m/seg não pode ser considerado candidato à associação.

A geração das entradas para a aplicação do treinamento da rede, é feita assumindo que as medidas fornecidas pelo sensor até o instante de tempo  $k - 1$  pertencem



à trajetória de um determinado alvo e não a clutter correlacionado (falsos acompanhamentos).

A informação do tipo de contato, fornecida pelo programa gerador de cenário, onde o valor 1 indica que o contato é gerado por um alvo e  $-1$  indica que o contato é clutter, é utilizada para compor o vetor de saídas desejadas. É assumido que o contato identificado como tipo 1 é o mais adequado ou coerente para atualizar o acompanhamento existente do mesmo alvo.

A partir dos cenários gerados para o treinamento da rede e da descrição sobre como é calculado cada um dos pares entrada-saída de treinamento, obtém-se 2766 pares de dados dos quais, 831 representam a associação do acompanhamento com o contato gerado pelo mesmo alvo e 1935 representam associação com clutter. Estes dados são pre-processados fazendo a normalização dos dados. Esta etapa é feita utilizando a toolbox de pre-processamento de dados do *software* Matlab. Finalmente, o conjunto de entradas é dividido em 3 subconjuntos nomeadamente treinamento, validação e teste, definidos aleatoriamente com proporções de 60%, 20% e 20% respectivamente. Cada um destes subconjuntos é avaliado no processo de treinamento para evitar a perda de generalização da rede neural.

A rede construída utiliza o algoritmo de treinamento *backpropagation* com otimização Levenberg-Marquart o qual é um dos algoritmos que apresentam melhor desempenho no treinamento de redes neurais alimentadas adiante (*feedforward*) [15]. Embora este algoritmo misture alta velocidade de convergência com estabilidade, seu custo computacional pode ser elevado, por ser um método de segunda ordem. A camada intermediária é ajustado em cinco neurônios tipo tangente hiperbólico e na camada de saída é utilizado um único neurônio do mesmo tipo.

A implementação da rede neural foi feita utilizando a *toolbox* de redes neurais artificiais que integra o *software* Matlab.

### 3.5 Confirmação e eliminação de acompanhamentos

Para a confirmação ou eliminação do acompanhamento é utilizado numa fase temprana o critério  $M/N$  descrito na seção 2, onde  $M$  e  $N$  são empiricamente definidos com valores iguais a 3 e 4 respectivamente. Se o critério é satisfeito e o acompanhamento confirmado, é verificado se no decorrer do tempo este acompanhamento apresenta mais do que três perdas consecutivas. Se assim acontecer, o mesmo é considerado como perdido e posteriormente eliminado.

### 3.6 Filtragem

Nesta etapa é utilizado o algoritmo de Múltiplos Modelos Interagentes (IMM), composto por dois filtros de Kalman estendido (EKF). A ideia é atingir um desempenho satisfatório quando a trajetória do alvo experimenta diferentes padrões de comportamento ao longo do tempo [9].

O EKF é um algoritmo subótimo de estimação de estado para sistemas não lineares. Sua derivação depende da linearização do modelo de espaço de estado definido como

$$X(k+1) = f(k, X(k)) + W(k) \quad (7)$$

$$Z(k+1) = fh(k+1, X(k+1)) + e(k+1), \quad (8)$$

onde  $X(k)$  é o vetor de estados,  $f(k, \hat{X}(k|k))$  é a função vetorial não linear de transição de estados,  $W(k)$  é o vetor de ruído associado ao processo,  $Z(k+1)$  é o vetor de medidas,  $fh(k+1, \hat{X}(k+1|k))$  é a função vetorial não linear que relaciona as variáveis de estado às medidas, e  $e(k+1)$  é o vetor de ruído associado à medida.

O conjunto de equações que formam a solução recursiva do EKF é apresentado a seguir [16]:

a) **Previsão do estado**

Estimativa a priori do estado:

$$\hat{X}(k+1|k) = f(k, \hat{X}(k|k)). \quad (9)$$

Estimativa a priori da medida:

$$\hat{Z}(k+1|k) = fh(k+1, \hat{X}(k+1|k)). \quad (10)$$

Matriz de covariância dos erros da estimativa a priori do estado:

$$P(k+1|k) = F(k) * P(k|k) * F(k)^T + Q(k). \quad (11)$$

Ganho do filtro:

$$K(k+1) = P(k+1|k) * H(k+1)^T * \left( H(k+1) * P(k+1|k) * H(k+1)^T + R(k+1) \right)^{-1}. \quad (12)$$

b) **Atualização da estimativa:**

Estimativa a posteriori do estado:

$$\hat{X}(k+1|k+1) = \hat{X}(k+1|k) + K(k+1) * (\nu(k+1)), \quad (13)$$

Matriz de covariância dos erros da estimativa atualizada:

$$P(k+1|k+1) = (I - K(k+1) * H(k+1)) * P(k+1|k). \quad (14)$$

Nas equações acima,  $\hat{X}(k+1|k+1)$  é sua estimativa a posteriori,  $F(k)$  é a matriz Jacobiana da função  $f(k, \hat{X}(k|k))$ ,  $H(k+1)$  é a matriz jacobiana da função  $fh(k+1, \hat{X}(k+1|k))$ ,  $Q(k)$  é matriz de covariância associada ao ruído do processo,  $P(k|k)$  é a covariância dos erros da estimativa a posterior do estado no instante  $k$  e  $\nu$  é o vetor de inovação ou medida residual. No caso em que  $f(k, \hat{X}(k|k))$  é linear,  $F(k)$  é simplesmente uma função matricial que, multiplicada por o vetor de estado  $\hat{X}(k|k)$ , obtém-se  $f(k, \hat{X}(k|k))$ .

Para o nosso caso foram selecionados dois modelos de filtro nomeadamente o modelo de Movimento Retilíneo Uniforme (MRU) e modelo de Giro Coordenado (*Coordinated Turn Model*, CT) [17, 18, 19]. A escolha dos modelos MRU e CT se justifica ao fato de que um navio, na maior parte do tempo, efetua um movimento parcimonioso que permite economizar o consumo de combustível, exceto em alguns casos em que há necessidade de alterar o movimento para executar alguma manobra [16], nos quais o MRU não seria o modelo adequado para descrever a cinemática do alvo.

Assim, para esses modelos definimos os seguintes parâmetros:

Modelo MRU	Modelo CT
$X(k) = \begin{bmatrix} x(k) \\ V_x(k) \\ y(k) \\ V_y(k) \end{bmatrix}$	$X(k) = \begin{bmatrix} x(k) \\ V_x(k) \\ y(k) \\ V_y(k) \\ \omega(k) \end{bmatrix}$

- **Vetor de estados:**

- **Estimativa inicial do estado**

Para calcular a estimativa inicial do estado, deve-se lembrar que  $d_{m(i)}$  e  $\theta_{m(i)}$  são as medições de localização feitas pelo sensor Radar. Assim para o modelo MRU teremos,

$$\hat{X}_1(0|0) = \begin{bmatrix} x_{m(2)} = d_{m(2)} * \cos \theta_{m(2)} \\ \frac{x_{m(2)} - x_{m(1)}}{T} \\ y_{m(2)} = d_{m(2)} * \sin \theta_{m(2)} \\ \frac{y_{m(2)} - y_{m(1)}}{T} \end{bmatrix}, \quad (15)$$

e para o modelo CT,

$$\hat{X}_2(0|0) = \begin{bmatrix} \hat{X}_1(0|0) \\ 0 \end{bmatrix}. \quad (16)$$

- **Função vetorial de transição de estados:**

Pela simplicidade de notação, nas equações abaixo omitiu-se o índice  $(k|k)$ .  
Modelo MRU:

$$f(k, \hat{X}(k|k)) = \begin{bmatrix} \hat{x} + \hat{V}_x * T_k \\ \hat{V}_x \\ \hat{y} + \hat{V}_y * T_k \\ \hat{V}_y \end{bmatrix}. \quad (17)$$

Como neste caso  $f(k, \hat{X}(k|k))$  é linear, teremos que:

$$F(k) = \begin{bmatrix} 1 & T_k & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_k \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (18)$$

Modelo CT:

$$f(k, \hat{X}(k|k)) = \begin{bmatrix} \hat{x} + \hat{V}_x * T_k - \hat{V}_y * \hat{\omega} * \frac{T_k^2}{2} \\ \hat{V}_x - \hat{V}_x * \hat{\omega}^2 * \frac{T_k^2}{2} - \hat{V}_y * \hat{\omega} * T_k \\ \hat{y} + \hat{V}_y * T_k + \hat{V}_x * \hat{\omega} * \frac{T_k^2}{2} \\ \hat{V}_y - \hat{V}_y * \hat{\omega}^2 * \frac{T_k^2}{2} + \hat{V}_x * \hat{\omega} * T_k \\ \hat{\omega}(k) \end{bmatrix}. \quad (19)$$

Assim, aplicando o jacobiano de  $f(k, \hat{X}(k|k))$  temos que:

$$F(k) = \begin{bmatrix} 1 & T_k & 0 & -\hat{\omega} * \frac{T_k^2}{2} & -\hat{V}_y * \frac{T_k^2}{2} \\ 0 & 1 - \hat{\omega}^2 * \frac{T_k^2}{2} & 0 & -\hat{\omega} * T_k & -\hat{V}_y * T_k - \hat{V}_x * \hat{\omega} * T_k^2 \\ 0 & \hat{\omega} * \frac{T_k^2}{2} & 1 & T_k & \hat{V}_x * \frac{T_k^2}{2} \\ 0 & \hat{\omega} * T_k & 0 & 1 - \hat{\omega}^2 * \frac{T_k^2}{2} & \hat{V}_x * T_k - \hat{V}_y * \hat{\omega} * T_k^2 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (20)$$

- **Matriz de covariância associada ao ruído do processo:**

Modelo MRU:

$$Q(k) = \begin{bmatrix} \frac{T_k^4}{4} \sigma_q^2 & \frac{T_k^3}{2} \sigma_q^2 & 0 & 0 \\ \frac{T_k^3}{2} \sigma_q^2 & T_k^2 \sigma_q^2 & 0 & 0 \\ 0 & 0 & \frac{T_k^4}{4} \sigma_q^2 & \frac{T_k^3}{2} \sigma_q^2 \\ 0 & 0 & \frac{T_k^3}{2} \sigma_q^2 & T_k^2 \sigma_q^2 \end{bmatrix}, \quad (21)$$

onde  $\sigma_q = 3.5 \frac{m}{seg^2}$ .

Modelo CT:

$$Q(k) = \begin{bmatrix} \frac{T_k^4}{4} * \sigma_q^2 & \frac{T_k^3}{2} * \sigma_q^2 & 0 & 0 & 0 \\ \frac{T_k^3}{2} * \sigma_q^2 & \frac{T_k^2}{2} * \sigma_q^2 & 0 & 0 & 0 \\ 0 & 0 & \frac{T_k^4}{4} * \sigma_q^2 & \frac{T_k^3}{2} * \sigma_q^2 & 0 \\ 0 & 0 & \frac{T_k^3}{2} * \sigma_q^2 & \frac{T_k^2}{2} * \sigma_q^2 & 0 \\ 0 & 0 & 0 & 0 & \frac{T_k^2}{2} * \sigma_\omega^2 \end{bmatrix}, \quad (22)$$

onde  $\sigma_q = 5 \frac{m}{seg^2}$  e  $\sigma_\omega = 0.044 \frac{rad}{seg}$ .

- **Função vetorial não linear que relaciona as variáveis de estado às medidas:**

Esta equação é a mesma para os dois modelos devido a que ambos recebem a mesma informação em coordenadas cartesianas. Assim,

$$fh(k+1, X(k+1)) = \begin{bmatrix} fh_1 = \sqrt{x(k+1)^2 + y(k+1)^2} \\ fh_2 = \arctan\left(\frac{x(k+1)}{y(k+1)}\right) \end{bmatrix}. \quad (23)$$

Por ser esta função não linear, é calculada a sua matriz jacobiana, pelo que teremos para o modelo MRU

$$H_1(k+1) = \begin{bmatrix} \frac{\hat{x}(k+1|k)}{\sqrt{(\hat{x}(k+1|k)^2 + \hat{y}(k+1|k)^2)}} & 0 & \frac{\hat{y}(k+1|k)}{\sqrt{(\hat{x}(k+1|k)^2 + \hat{y}(k+1|k)^2)}} & 0 \\ \frac{\hat{y}(k+1|k)}{\hat{x}(k+1|k)^2 + \hat{y}(k+1|k)^2} & 0 & -\frac{\hat{x}(k+1|k)}{\hat{x}(k+1|k)^2 + \hat{y}(k+1|k)^2} & 0 \end{bmatrix}, \quad (24)$$

e para o modelo CT

$$H_2(k+1) = \begin{bmatrix} H_1(k+1) & \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (25)$$

- **Matriz de covariância da estimativa inicial:**

Modelo MRU:

$$P_1(0|0) = \begin{bmatrix} \sigma_x^2 & \frac{\sigma_x^2}{T} & \sigma_{xy}^2 & \frac{\sigma_{xy}^2}{T} \\ \frac{\sigma_x^2}{T} & \frac{2\sigma_x^2}{T^2} & \frac{\sigma_{xy}^2}{T} & \frac{2\sigma_{xy}^2}{T^2} \\ \sigma_{xy}^2 & \frac{\sigma_{xy}^2}{T} & \sigma_y^2 & \frac{\sigma_y^2}{T} \\ \frac{\sigma_{xy}^2}{T} & \frac{2\sigma_{xy}^2}{T^2} & \frac{\sigma_y^2}{T} & \frac{2\sigma_y^2}{T^2} \end{bmatrix}. \quad (26)$$

Modelo CT:

$$P_2(0|0) = \begin{bmatrix} P_1(0|0) & \mathbf{0} \\ \mathbf{0} & P_\omega \end{bmatrix}, \quad (27)$$

onde  $\sigma_d = 30m$ ,  $\sigma_\theta = 3mrad$ ,  $P_\omega(0|0) = 0.0076$  e

$$\sigma_x^2 = \sigma_d^2 * \cos^2 \theta_{m(2)} + \sigma_\theta^2 * d_{m(2)}^2 * \sin^2 \theta_{m(2)} \quad (28)$$

$$\sigma_{xy}^2 = (\sigma_d^2 - \sigma_\theta^2 * d_{m(2)}) * \cos \theta_{m(2)} * \sin \theta_{m(2)} \quad (29)$$

$$\sigma_y^2 = \sigma_d^2 * \sin^2 \theta_{m(2)} + \sigma_\theta^2 * d_{m(2)}^2 * \cos^2 \theta_{m(2)} \quad (30)$$

O IMM é um algoritmo proposto por Blom e Bar-shalom em 1988 [20], cuja principal característica é a habilidade de estimar o estado de um sistema dinâmico e sua respectiva matriz de covariância através da soma ponderada das estimativas de N modelos de filtro executados em paralelo [7, 21].

Os detalhes do ciclo prático do algoritmo IMM são apresentado em DANG *et al.* [22]. Para sua execução é necessário atribuir uma condição inicial para alguns parâmetros associados ao algoritmo. Neste trabalho o valor inicial das probabilidades de que o movimento do alvo seja caracterizado por um dos modelos assumidos é  $\mu_1(0) = \mu_2(0) = 0.5$  e a matriz de transição da cadeia de Markov é definida segundo o processo descrito em [8] e [23], desta forma teremos

$$P = \begin{bmatrix} 0.99 & 0.01 \\ 0.1 & 0.9 \end{bmatrix}. \quad (31)$$

## 4 Resultados

Na tabela acima, M1 indica o número de contatos gerados pelos alvos, M2 o número de acertos, M3 perdas de detecção, M4 clutter's associados aos acompanhamentos, M5 taxa de acerto (%), M6 tempo médio da Iteração(seg) e M7 número de acompanhamentos confirmados. Por tempo de iteração entende-se o tempo que o programa leva em analisar os dados contidos numa única varredura de antena.

Dos resultados apresentados, pode-se observar que nos cenários 1, 2 e 3 o método clássico apresenta uma taxa de acerto um pouco superior ao método proposto, isto se deve ao fato de o método proposto ser mais restritivo na hora de selecionar os contatos, o que gera um número maior de perdas de detecção e menor número de associações com clutter em comparação com o método clássico em todos os cenários. É importante destacar que apesar de apresentar maior número de perdas isto não se traduz na perda do acompanhamento dos alvos.

Nos cenários onde foram introduzidas as perdas de detecção (cenários 3 e 5) o método clássico não conseguiu manter o acompanhamento de dois dos alvos, o que pode ser considerado como um ponto fraco deste método.

Tabela 2: Comparação de desempenho entre os métodos proposto e clássico para diferentes cenários

Cenário 1							
	M1	M2	M3	M4	M5	M6	M7
Método Clássico	266	266	0	1	99.63	1.96281	6
Método Proposto	266	258	8	0	96.99	0.14504	5
Cenário 2							
Método Clássico	281	279	2	2	98.59	2.21144	8
Método Proposto	281	262	18	1	93.24	0.15713	4
Cenário 3							
Método Clássico	265	261	4	4	97.03	2.0066	10
Método Proposto	265	256	8	2	96.24	0.16002	4
Cenário 4							
Método Clássico	201	199	0	4	98.03	1.79256	8
Método Proposto	201	198	2	1	98.51	0.11164	4
Cenário 5							
Método Clássico	188	184	2	8	94.85	1.8421	11
Método Proposto	188	182	5	2	96.29	0.1078	4
Cenário 6							
Método Clássico	308	307	0	4	98.71	2.20885	9
Método Proposto	308	305	3	0	99.03	0.17404	6

A grande diferença entre os dois métodos é o tempo médio de execução de uma iteração. Pode-se observar que o tempo médio gasto pelo método proposto é inferior a 1/10 do tempo gasto pelo método clássico, o que fornece uma grande vantagem para o método proposto na hora da sua implementação em aplicações em tempo real. Conforme discutido em [11], este resultado justifica-se pelo fato de que um cenário com alta densidade de clutter gera um número elevado de hipóteses, o que aumenta o custo computacional deste método.

Em relação ao número de acompanhamentos confirmados que não representam um alvo de interesse (falsos acompanhamentos), o método clássico apresenta um número maior em comparação com o método proposto, isto é devido à busca intensiva feita pelo método clássico e ao processo de inicialização descrito na seção 2. Embora estes acompanhamentos sejam eliminados nas varreduras seguintes, os mesmos podem causar uma queda do desempenho, aumentando o tempo de execução por iteração.

## 5 Conclusões

O método proposto permite abordar problemas de acompanhamento envolvendo um número maior de alvos para diferentes cenários com alta densidade de clutters, uma vez que o seu desempenho reduz significativamente o custo computacional em comparação com o método clássico. O algoritmo de filtragem construído para as duas metodologias possui a capacidade de ponderar dois modelos dinâmicos, um para alvos em movimento retilíneo uniforme e outro para alvos que efetuam manobras, e se adapta melhor às variações de cinemática referente a trajetória do alvo que está sendo acompanhado.

**Agradecimentos:** Ao IPqM e FUNDEP.

## Referências

- [1] WASSERMAN, P. D., 1989, *Neural Computing Theory and Practice*. Van Nostrand Reinhold: New York, USA.
- [2] HERTZ, J., KROGH, A., PALMER, R., 1991, *Introduction to the Theory of Neural Computation*. 1st ed. Addison-Wesley Publishing Company: Reading, Massachusetts.
- [3] WINTER, M., FAVIER, G., “A neural network for data association”, *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1999. Proceedings of the IEEE, 1999*, v. 2, pp. 1041–1044, Mar. 1999.
- [4] WINTER, M., FAVIER, G., “A neural solution for multitarget tracking based on a maximum likelihood approach”, *IEEE International Conference on Acoustics, Speech and Signal Processing, 1998. Proceedings of the IEEE*, v. 2, pp. 1141–1144, May. 1998.
- [5] LIN, Y. J., CHEN, H. T., SU, J. D., et al., “Applying the Competitive Hopfield Neural Network to Multiple Target Tracking Systems”, *Automatic Control Conference Tainan, Taiwan*, Nov. 2005.
- [6] RICHARD, M. D., LIPPMANN, R. P., “Neural network classifiers estimate Bayesian a posteriori probabilities”, *Neural computation*, v. 3, n. 4, pp. 461–483, 1991.
- [7] GUANG-YUAN, Z., FU-JUN, W., ZHEN-SHENG, W., “Interacting Multiple Model Algorithm Used In Multi-Sensor Fusion System”, *8th International Conference on Electronic Measurement and Instruments*, , n. 4, pp. 4–135 4–139, 2007.
- [8] BAR-SHALOM, Y., LI, X.-R., 1995, *Multitarget-Multisensor Tracking: Principles and Techniques*. 3rd ed. Yaakov Bar-Shalom.
- [9] KIRUBARAJAN, T., BAR-SHALOM, Y., “Kalman Filter vs. IMM Estimator: When Do We Need the Latter?” *IEEE Transactions on Aerospace and Electronic Systems*, v. 39, n. 4, pp. 1452 1457, Oct. 2003.
- [10] PETERS, D. J., *A Practical Guide to Level One Data Fusion Algorithms*, Tech. Rep. DREA TM 2001-201, Defence Research Establishment Atlantic, 2001.
- [11] AMDITIS, A., THOMAIDIS, G., MAROUDIS, P., et al., “Multiple Hypothesis Tracking Implementation”, In: RODRIGUEZ, J. A. M. (ed), *Laser Scanner Technology*, chap. 10, InTechOpen, 2012.
- [12] BAR-SHALOM, Y., 1990, *Multitarget-multisensor tracking: advanced applications*. v. 1. Artech House: Norwood, MA.
- [13] MIGUENS, A. P., 1996, *Navegação a Ciência e a Arte: Navegação Costeira, Estimada e em Águas Restritas*. v. 1. Rio de Janeiro: DHN.
- [14] KANG, E. W., 2008, *Radar system analysis, design, and simulation*. Artech House.

- [15] HAGAN, M. T., MENHAJ, M. B., “Training feedforward networks with the Marquardt algorithm”, *Neural Networks, IEEE Transactions on*, v. 5, n. 6, pp. 989–993, Nov. 1994.
- [16] DE OLIVEIRA, J. R. P., *Acompanhamento de Alvos Radar Utilizando Filtragem de Kalman e Vetor de Estados com Dimensão Variável*, Tese de mestrado, PEE - COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2005.
- [17] BAR-SHALOM, Y., WILLETT, P. K., TIAN, X., 2011, *Tracking and Data Fusion a Handbook of Algorithms*. Yaakov Bar-Shalom.
- [18] LI, X.-R., JILKOV, V. P., “Survey of Maneuvering Target Tracking: Dynamic Models”. In: *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, v. 4048, pp. 212–235, Orlando, FL, USA, April 2000.
- [19] YUAN, X., LIAN, F., HAN, C., “Models and Algorithms for Tracking Target with Coordinated Turn Motion”, *Mathematical Problems in Engineering*, v. 2014, n. 649276, pp. 1–10, 2014.
- [20] BLOM, H. A. P., BAR-SHALOM, Y., “The interacting multiple model algorithm for systems with Markovian switching coefficients”, *IEEE Transactions on Automatic Control*, v. 33, n. 8, pp. 780–783, 1988.
- [21] ALY, S. M., FOULY, R. E., BARAKA, H., “Extended Kalman Filtering and Interacting Multiple Model for Tracking Maneuvering Targets in Sensor Networks”, *Seventh Workshop on Intelligent solutions in Embedded Systems*, pp. 149–156, Jun. 2009.
- [22] DANG, H., HAN, C., GRUYER, D., “Combining of IMM filtering and DS data association for multitarget tracking”, *Fusion 2004: Seventh International Conference on Information Fusion*, 2004.
- [23] BLACKMAN, S. S., POPOLI, R., 1999, *Design and analysis of modern tracking systems*. Artech House.