

Interceptação Visual de Alvos 3D usando Redes Neurais de Aprendizado Não Supervisionado

Sidney Antonio A. Viana, Cairo L. Nascimento Jr., Jacques Waldmann

Divisão de Engenharia Eletrônica

Instituto Tecnológico de Aeronáutica

12228-900 - São José dos Campos - SP - Brasil

E-mails: sidney@ele.ita.cta.br, cairo@ele.ita.cta.br, jacques@ele.ita.cta.br

Abstract

This paper concerns the application of two neural network architectures to solve the problem of visual interception of a stationary 3D target by a stereo (binocular) vision system. The neural networks are trained using competitive and unsupervised learning. The first neural architecture uses two Fuzzy-ART pattern-clustering neural networks which are combined with a linear "representation layer" to act as a fast open-loop neural controller and provides a rapid and coarse positioning of the visual system. The second neural architecture is a Kohonen network that acts as a closed-loop neural controller and provides a slower but more accurate positioning of the visual system. We show how to use these two neural architectures separately and combined to solve the visual interception problem. Simulation results of some interception tasks are presented and show that the better results are obtained when both neural architectures are combined.

1. Introdução

Um problema de interesse fundamental em muitos sistemas que utilizam câmeras de visão como elemento sensor é a interceptação visual ou o rastreamento visual de objetos. Essas são capacidades importantes de muitos animais em suas interações com o meio externo. Para os animais, por exemplo, um objeto pode significar alimento ou perigo. No caso de uma plataforma móvel com sensor de visão, um objeto pode significar uma chance de colisão ou uma referência para navegação visual.

Este artigo apresenta a implementação de duas metodologias de interceptação visual baseadas em redes neurais. O sistema de visão é computacionalmente simulado e usado como plataforma para a implementação e estudo das propriedades das abordagens neurais de interceptação visual de alvos estacionários no espaço 3D. As redes neurais em questão fazem uso de aprendizado competitivo [1;2] e não-supervisionado, o que lhes confere uma capacidade de auto-organização.

2. Modelo do Sistema de Visão

O sistema de visão foi simulado com base no modelo de um sistema real (TRC, *Helpmate Robotics Inc.*) existente no Laboratório de Visão Ativa (Div. de Engenharia Eletrônica / ITA). A Figura 1 esquematiza o sistema, que possui três graus de liberdade rotacionais para as câmeras: *pan* (rotação em azimute), *tilt* (rotação em elevação) e vergências esquerda e direita. As câmeras foram consideradas com resolução de $320 \times 240 \text{ pixels}^2$ e com distância focal de 2,0 cm.

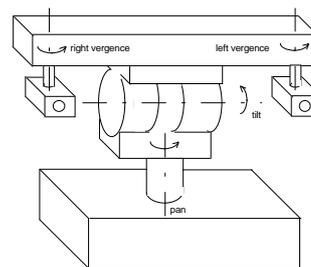


Figura 1: Ilustração do sistema de visão simulado.

A dinâmica de posicionamento angular em um sistema rotacional pode ser caracterizada por uma função de transferência da forma $G(s) = a/s(s+b)$, de acordo com o diagrama de blocos mostrado a seguir.

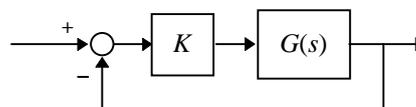


Figura 2: Diagrama de um sistema de posicionamento angular.

Objetivando-se rapidez de resposta, reduzido sobressinal e erro estacionário nulo, a dinâmica rotacional de todas as juntas (*pan*, *tilt* e vergências) do sistema foram consideradas como:

$$G(s) = \frac{289}{s(s+13,6)} ; K = 0,24 \quad (1)$$

permitindo que o sistema se estabilize adequadamente na atitude fornecida como referência.

3. Aprendizado Competitivo

Em certas arquiteturas de redes neurais, os pesos de todos os neurônios de uma camada são ajustados paralelamente durante o processo de aprendizado (p.e. redes de perceptrons, treinadas pelo algoritmo *Backpropagation*). Um diferente tipo de aprendizado é conhecido como **aprendizado competitivo**, no qual os neurônios de uma camada competem entre si pelo privilégio de permanecerem ativos, tal que o neurônio com maior atividade seja o único a participar do processo de aprendizado. Em termos de neurodinâmica, tal comportamento competitivo é descrito por equações diferenciais não-lineares, envolvendo conexões auto-excitatórias e inibitórias entre os neurônios da camada, conforme mostrado na Figura 3 [3]. As unidades com menores atividades serão mais inibidas que a unidade com maior atividade, de modo que esta torna-se vencedora da competição, sendo a única a participar do aprendizado. Este tipo de competição é conhecido como *winner-take-all* (vencedor-leva-tudo) [2;3;4] e possui grande importância na formação de redes neurais auto-organizáveis.

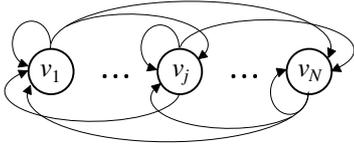


Figura 3: Camada competitiva de neurônios.

Seja \mathbf{I} um padrão de entrada e \mathbf{w}_j o vetor de pesos associado ao neurônio v_j da camada competitiva, define-se a entrada-total da unidade v_j por:

$$net_j = \mathbf{w}_j^T \cdot \mathbf{I} = \sum_{i=1}^M w_{ij} I_i \quad (2)$$

Para efeito de implementações computacionais não é necessária a resolução das equações diferenciais de competição envolvendo as conexões interneuronais. É suficiente apenas simular seus efeitos, isto é, verificar qual a unidade apresentando maior atividade, em decorrência da aplicação de uma entrada \mathbf{I} , e tomá-la como unidade vencedora. Assim, denotando-se a unidade vencedora pelo índice J , esta é tal que:

$$net_J = \max\{net_j\}, \forall j \quad (3)$$

O aprendizado competitivo pode então ser resumido aos seguintes passos:

loop: para cada padrão de entrada \mathbf{I} :

1. Fornecer o padrão de entrada corrente à rede.

2. Calcular a entrada-total de cada unidade na camada competitiva, de acordo com (2), e identificar a unidade vencedora J por (3).
3. Proceder ao processo de aprendizado da unidade J .

end loop

O passo 3 refere-se ao ajuste dos pesos da rede, que consiste em aumentar a similaridade entre o vetor de pesos da unidade vencedora e o padrão de entrada. Isso é usualmente realizado através da **regra delta** [1;3]:

$$\Delta \mathbf{w} = -\lambda (\mathbf{Y}_D - \mathbf{Y}) \quad (4)$$

onde λ é a taxa de aprendizado, \mathbf{Y} é a saída decorrente da entrada \mathbf{I} , e \mathbf{Y}_D é a saída desejada. Como consequência desse aumento de similaridade, e em decorrência das equações (2) e (3), entradas semelhantes tenderão a ativar uma mesma unidade na camada competitiva, que passa então a funcionar então como uma camada auto-organizável de classificação de padrões ou **camada de categorias**.

4. Controlador Neural Open-Loop Baseado em Redes Fuzzy-ART

Uma importante teoria de redes neurais, envolvendo aprendizado competitivo, é conhecida como Teoria de Resonância Adaptativa (**ART** – *Adaptive Resonance Theory*). ART constitui uma família de arquiteturas neurais, dentre as quais citam-se [5;6]: ART1, ART2 e Fuzzy-ART, possuindo como característica fundamental a capacidade de **aprendizado incremental**, isto é, a capacidade de aprender novas informações sem prejuízo de informações aprendidas no passado [5]. A rede ART1 realiza o processamento apenas de dados binários, ao passo que as redes ART2 e Fuzzy-ART são aptas ao processamento de dados contínuos.

Com relação ao sistema de visão citado na Seção 2, um dado ponto 3D imageado $\mathbf{P} = [x \ y \ z]$ pode ser representado em termos dos ângulos de *tilt* (ϕ) e vergências esquerda e direita (θ_L e θ_R) das câmeras, mantendo-se fixo o grau de liberdade *pan*. Define-se o vetor de atitude \mathbf{A} como

$$\mathbf{A} = [\alpha_1 \ \alpha_2 \ \alpha_3] = [\theta_L \ \phi \ \theta_R] \quad (5)$$

e o vetor de coordenadas visuais \mathbf{V} por

$$\mathbf{V} = [v_1 \ v_2 \ v_3 \ v_4] = [c_L \ r_L \ c_R \ r_R] \quad (6)$$

onde c e r denotam, respectivamente, a coluna e a linha (em *pixels*) do ponto-imagem de \mathbf{P} .

Para um dado ponto \mathbf{P} imageado, existem diversas combinações $\{\mathbf{A}_i, \mathbf{V}_i\}$ de vetores de atitude e correspondentes vetores visuais associadas a \mathbf{P} .

A primeira arquitetura neural utilizada para este fim é mostrada na Figura 7, sendo constituída por duas subredes Fuzzy-ART (uma para cada câmera) e uma camada de representação. A rede é treinada de modo a desenvolver para as diversas combinações $\{\mathbf{A}_i, \mathbf{V}_i\}$ de um mesmo ponto \mathbf{P} uma única representação interna $\Psi = [\psi_1 \ \psi_2 \ \psi_3]$.

Os dados de treinamento são constituídos por um conjunto aleatoriamente gerado de $N=20000$ dados $\{\mathbf{A}_i, \mathbf{V}_i\}$ correspondentes ao mesmo ponto \mathbf{P} .

Inicialmente as subredes Fuzzy-ART são treinadas, independentemente, com base nos dados visuais \mathbf{V}_i . As camadas F_0 recebem as entradas $[v_1 \ v_2]$ e $[v_3 \ v_4]$, convertendo-as para a forma de código complementar $\mathbf{I}^L = [v_1 \ v_2 \ 1-v_1 \ 1-v_2]$ e $\mathbf{I}^R = [v_3 \ v_4 \ 1-v_3 \ 1-v_4]$, que são passadas às camadas F_1 [6]. O treinamento é do tipo competitivo tal que, para cada entrada \mathbf{I}_i , a rede Fuzzy-ART determina a unidade vencedora J na camada F_2 de acordo com a **função de escolha de categoria**:

$$T_j = \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|} \quad ; \quad net_j = \max\{T_j\}, \forall j \quad (7)$$

onde α é um fator constante e “ \wedge ” é o operador *fuzzy-and*. O aprendizado ocorre desde que a unidade vencedora satisfaça o critério de casamento:

$$\frac{|\mathbf{I} \wedge \mathbf{w}_j|}{|\mathbf{I}|} \geq \rho \quad (8)$$

onde ρ é o **coeficiente de vigilância** da rede Fuzzy-ART. Se o critério (8) não é satisfeito, a unidade J corrente é desativada, e uma nova unidade vencedora é determinada por (7). Se (8) é satisfeito pela unidade J , diz-se que a rede alcançou um **estado ressonante** [6], e o aprendizado ocorre de acordo com

$$\mathbf{w}_j^{(\text{nov})} = \mathbf{I} \wedge \mathbf{w}_j^{(\text{atual})} \quad (9)$$

A denominação “Fuzzy” da rede Fuzzy-ART advém do operador “ \wedge ”. O processo de busca por uma adequada unidade J faz com que a rede Fuzzy-ART desenvolva de forma incremental uma categorização (*clustering*) auto-organizada dos dados de entrada.

A Figura 4 mostra as classes formadas pela rede Fuzzy-ART esquerda, subentendidas pelos pesos \mathbf{w}_j^L (vide [6] para detalhes sobre a representação gráfica das classes).

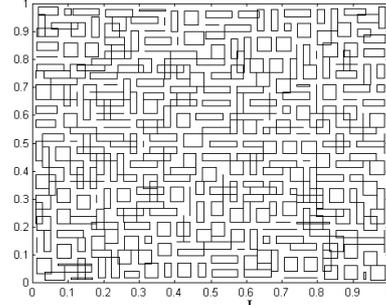


Figura 4: Categorias (pesos \mathbf{w}_j^L) desenvolvidas pela subrede Fuzzy-ART esquerda.

Após o treinamento dos módulos Fuzzy-ART, procede-se ao treinamento da camada de representação. Os dados visuais \mathbf{V}_i são fornecidos às redes Fuzzy-ART e os pesos \mathbf{z}_J e \mathbf{z}_K das unidades (categorias) ativadas são combinados com os dados de atitude \mathbf{A} como:

$$\Psi = \mathbf{A} + \mathbf{z}_J + \mathbf{z}_K \quad (10)$$

O objetivo é fazer com que a representação seja única para todos os dados de treinamento, isto é, $\Psi_i = \Psi$. Para tanto, os N dados de treinamento são tomados aos pares, $\{\mathbf{A}_1, \mathbf{V}_1\}$ e $\{\mathbf{A}_2, \mathbf{V}_2\}$, aleatoriamente, calculando-se suas respectivas representações Ψ_1 e Ψ_2 , de acordo com (9), e usando-se a seguinte regra de aprendizado no ajuste dos pesos da camada de representação:

$$\Delta \mathbf{z}_{J_1} = -\varepsilon (\Psi_1 - \Psi_2) \rho_L \quad (11a)$$

$$\Delta \mathbf{z}_{K_1} = -\varepsilon (\Psi_1 - \Psi_2) \rho_R \quad (11b)$$

$$\Delta \mathbf{z}_{J_2} = -\varepsilon (\Psi_2 - \Psi_1) \rho_L \quad (11c)$$

$$\Delta \mathbf{z}_{K_2} = -\varepsilon (\Psi_2 - \Psi_1) \rho_R \quad (11d)$$

onde ε é a taxa de aprendizado, ρ_L e ρ_R são os coeficientes de vigilância das redes Fuzzy-ART, J_1 e K_1 são os neurônios ativados por \mathbf{V}_1 , e J_2 e K_2 são os neurônios ativados por \mathbf{V}_2 . Observe-se que o treinamento é não-supervisionado porque o erro $\pm(\Psi_1 - \Psi_2)$ usado no treinamento é gerado pela própria rede neural com base apenas nos dados de treinamento, e não a partir de dados fornecidos por um supervisor externo. A Figura 5 mostra o decaimento do erro de representação, indicando a convergência da representação para um valor médio Ψ comum a todas as categorias.

Uma vez concluído o treinamento da camada de representação, a rede neural pode então ser usada na determinação da atitude necessária à interceptação do ponto \mathbf{P} . Considere-se que o sistema esteja a uma atitude inicial \mathbf{A}_0 , resultando em um vetor visual \mathbf{V}_0 . O alvo \mathbf{P} deve ser interceptado visualmente em uma posição \mathbf{V}_d . Para isso, \mathbf{V}_d é fornecido à rede e os pesos \mathbf{z}_{J_d} e \mathbf{z}_{K_d} das categorias ativadas são lidos. \mathbf{V}_0 e \mathbf{A}_0 são fornecidos à rede neural, obtendo-se uma representação

Ψ , de (10). Como a representação Ψ é única, a atitude de interceptação \mathbf{A}_d é então calculada como

$$\mathbf{A}_d = \Psi - \mathbf{z}_{Jd} - \mathbf{z}_{Kd} \quad (12)$$

A atitude \mathbf{A}_d é usada como comando de posição angular para as juntas de *tilt* e vergências do sistema.

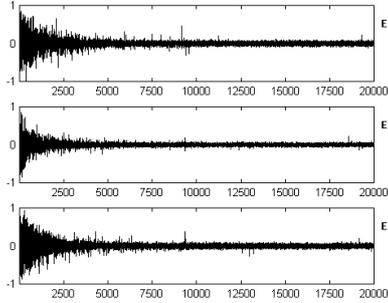


Figura 5: Evolução das três componentes do erro de representação $\mathbf{E} = \Psi_1 - \Psi_2$ durante o treinamento. O treinamento foi realizado por dois *epochs*. Em cada *epoch* são utilizados $N/2$ pares de treinamento.

A Figura 8(a) mostra um exemplo de interceptação usando a rede neural descrita. O ponto de interceptação desejado consistiu no ponto central de imagem em ambas as câmeras, isto é, $\mathbf{V}_d = [160 \ 120 \ 160 \ 120]$ *pixels*. Devido ao inerente desvio-padrão não-nulo de Ψ (Figura 5), resulta um pequeno erro residual de interceptação.

A ação de controle \mathbf{A}_d fornecida pela rede neural é calculada em um único passo, de modo que a rede funciona como um controlador em malha aberta (*open-loop*).

5. Controlador Neural *Closed-Loop* Baseado em Rede de Kohonen

Uma segunda arquitetura neural utilizada no problema de interceptação visual faz uso de uma Rede de Kohonen em associação com uma camada de saída, conforme mostrado na Figura 6.

Na fase de treinamento, a rede é exercitada através de trajetórias suaves aleatórias entre uma atitude inicial \mathbf{A}_0 e uma atitude final \mathbf{A}_F , escolhidas aleatoriamente. A geração dessas trajetórias foi feita com um gerador 4-3-4 [7], que desenvolve curvas formadas por três segmentos polinomiais contínuos entre si, de graus 4, 3 e 4, respectivamente. A cada instante de tempo k , a atitude corrente \mathbf{A}^k e as variações posicionais no plano-imagem $\Delta \mathbf{V}^k = \mathbf{V}^k - \mathbf{V}^{k-1}$ são fornecidas como entrada ao controlador neural. A correspondente variação de atitude, $\Delta \mathbf{A}^k = \mathbf{A}^k - \mathbf{A}^{k-1}$, é fornecida como saída desejada. Portanto, o treinamento faz com que a rede desenvolva um mapeamento auto-organizado entre variações $\Delta \mathbf{V}$ nas coordenadas-imagem e variações $\Delta \mathbf{A}$ nos ângulos de atitude.

Durante o treinamento, o movimento angular das câmeras é realizado pelo gerador de trajetórias. O erro de similaridade entre o vetor de entrada e os pesos \mathbf{w}_j de cada unidade j da rede de Kohonen é medido por:

$$e_j = \|\mathbf{u} - \mathbf{w}^j\|_2 = \left[\sum_{i=1}^4 (\Delta v_i - w_i^j)^2 + \sum_{i=1}^3 (\alpha_i - w_{i+4}^j)^2 \right]^{1/2} \quad (13)$$

sendo usado para definir a unidade vencedora como:

$$e_J = \min\{e_j\}, \forall j \quad (14)$$

A medida de vizinhança entre as unidades da rede de Kohonen é dada pela distância radial d entre as unidades, consideradas alocadas em um plano. O aprendizado competitivo e não-supervisionado da rede de Kohonen é baseado na regra:

$$\Delta w_i^J = \lambda e^{-\left(\frac{d}{R}\right)^2} (u_i - w_i^J) \quad (15)$$

onde J indica a unidade vencedora, λ é a taxa de aprendizado, d é a distância entre a unidade j e a unidade vencedora J , e R é o coeficiente de vizinhança em torno do neurônio J . O termo exponencial em (13) faz com que o grau de ajuste dos pesos seja maior para a unidade vencedora, diminuindo em função da distância até as demais unidades. Isto significa que a camada competitiva é do tipo “*winner-take-most*”, ao invés de “*winner-take-all*”, pois os pesos de todas as unidades são ajustados, sendo que o ajuste é mais intenso para a unidade vencedora.

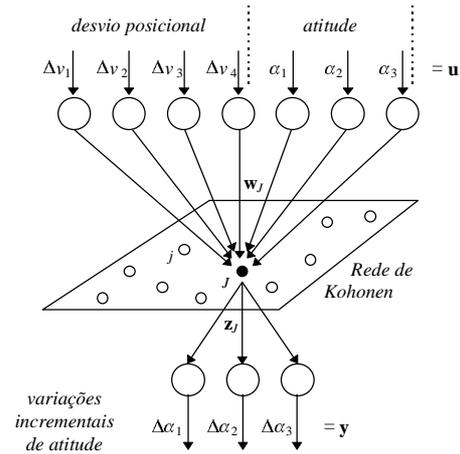


Figura 6: Rede neural para controle em malha fechada

Uma vez determinada a unidade vencedora J para uma entrada \mathbf{u} , o peso de saída \mathbf{z}_J é adaptado de acordo com a seguinte regra delta:

$$\Delta \mathbf{z}^J = \lambda (\Delta \mathbf{A} - \mathbf{z}^J) \quad (16)$$

onde λ é a taxa de aprendizado. Os parâmetros λ e R são feitos inicialmente com valores elevados, de modo a permitir uma rápida adaptação inicial da rede de Kohonen ao tamanho do espaço de entradas. A medida que o aprendizado prossegue, os valores de λ e R são gradativamente reduzidos usando-se

$$\lambda_t = \lambda_0 \left(\frac{\lambda_F}{\lambda_0} \right)^{\frac{t}{t_{\max}}} \quad R_t = R_0 \left(\frac{R_F}{R_0} \right)^{\frac{t}{t_{\max}}} \quad (17)$$

onde λ_0 e λ_F representam os valores inicial e final de λ (idem para R_0 e R_F), e t_{\max} é o tempo máximo de treinamento de cada iteração. Assim, a representação distribuída das entradas na rede de Kohonen é mapeada às correspondentes variações de atitude.

Uma vez concluído o treinamento, o processo de interceptação é realizado aplicando-se a entrada $\mathbf{u} = [\Delta \mathbf{V} \ \mathbf{A}]$ corrente à rede, que então fornece como saída uma variação incremental de atitude $\mathbf{y} = \Delta \mathbf{A}$, obtida dos pesos sinápticos da unidade J ativada. Isso é feito de forma retroativa, durante um número de iterações suficiente para que as imagens do alvo tenham convergido para o ponto de interceptação \mathbf{V}_d desejado.

A rede de Kohonen foi implementada tendo 15×15 unidades. O treinamento foi realizado com 500 trajetórias, com incrementos angulares de $1/50$ grau. Os valores dos parâmetros de aprendizado foram: $\lambda_0=0,99$; $\lambda_F=0,01$; $R_0=100$; e $R_F=0,1$.

A Figura 8(b) mostra um exemplo de interceptação realizada pela rede. As variações incrementais de atitude foram calculadas pela rede por 80 iterações, resultando em um menor erro de interceptação, em comparação com a metodologia anterior.

Uma vez que as variações de atitude são calculadas de forma retroativa, a rede funciona como um controlador em malha fechada (*closed-loop*).

6. Utilização Combinada das Redes

A variação da posição do alvo nas imagens, para os dois exemplos de interceptação anteriormente descritos, é mostrada na Figura 9. Observando-se que a primeira rede neural apresenta maior rapidez, porém menor precisão que a segunda. A utilização em sequência das duas arquiteturas permite combinar as vantagens de cada arquitetura neural, obtendo-se maior rapidez e precisão de posicionamento do que seria obtido com cada arquitetura separadamente, conforme mostrado na Figura 10.

7. Conclusões

As redes neurais utilizadas na realização das tarefas de interceptação visual apresentam certas diferenças de

desempenho entre si. A primeira arquitetura possui a vantagem de ser mais rápida devido ao fato de a ação de controle \mathbf{A}_d ser calculada em um único passo.

A segunda arquitetura apresenta menor rapidez em virtude de realizar a interceptação de forma incremental, porém permitindo maior exatidão. Simulações adicionais permitiram verificar, ainda, que a primeira arquitetura apresenta bom desempenho apenas na interceptação de pontos do espaço próximos do ponto \mathbf{P} utilizado na geração dos dados de treinamento, sugerindo que a rede neural realiza uma generalização local. Este problema não ocorre para a segunda arquitetura, em virtude de a mesma operar retroativamente com variações incrementais de dados, o que assegura uma melhor convergência. A combinação em sequência das duas redes permite obter melhores resultados.

Maiores detalhes sobre as metodologias aqui apresentadas podem ser encontrados em [11].

Referências

- [1] C. L. Nascimento Jr. and T. Yoneyama. *Inteligência Artificial em Automação e Controle*. Edgard-Blücher. 1999. (a ser publicado).
- [2] E. Rich and K. Knight. *Inteligência Artificial*. Makron Books. 1994.
- [3] S. Haykin. *Neural Networks - A Comprehensive Foundation*. McMillan. 1994.
- [4] R. P. Lippmann. An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine*, pp.4-22. 1987.
- [5] J. A. Freeman and D. M. Skapura. *Neural Networks - Algorithms, Applications and Programming Techniques*. Addison-Wesley. 1991.
- [6] G. A. Carpenter, S. Grossberg and D. B. Rosen. Fuzzy-ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System. *Neural Networks*, vol.04, pp.759-771. 1991.
- [7] K. S. Fu, R. C. Gonzalez and C. S. G. Lee. *Robotics: Control, Sensing, Vision and Intelligence*. McGraw-Hill. 1987.
- [8] J. Huang and M. Georgiopoulos. Properties of Learning in Fuzzy-ART. 1994 *IEEE International Conference on Neural Networks*. pp.756-761. 1994.
- [9] N. Srinivasa and R. Sharma. SOIM: A Self-Organizing Invertible Map with Applications in Active Vision. *IEEE Transactions on Neural Networks*, vol.08, nº 03, pp.758-773. 1997.
- [10] N. Srinivasa and R. Sharma. Execution of Saccades for Active Vision Using a Neurocontroller. *IEEE Control Systems Magazine*, vol.17, nº 02, pp.18-29. 1997.
- [11] S. A. A. Viana. *Estudo sobre Redes Neurais Não Supervisionadas Aplicado a Simulações de Interceptação Visual de Alvos 3D*. Tese de Mestrado. Departamento de Sistemas e Controle, Divisão de Engenharia Eletrônica, Instituto Tecnológico de Aeronáutica (ITA-IEE-IEES). 1999.

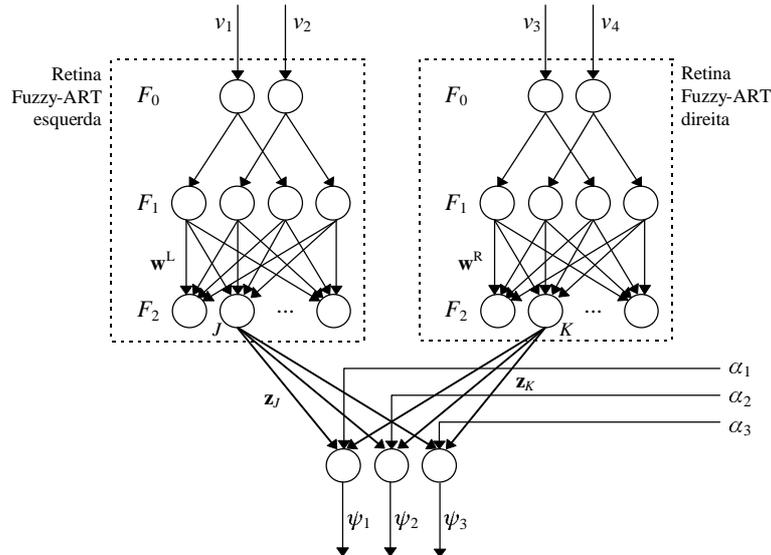


Figura 7: Rede neural para controle em malha aberta.

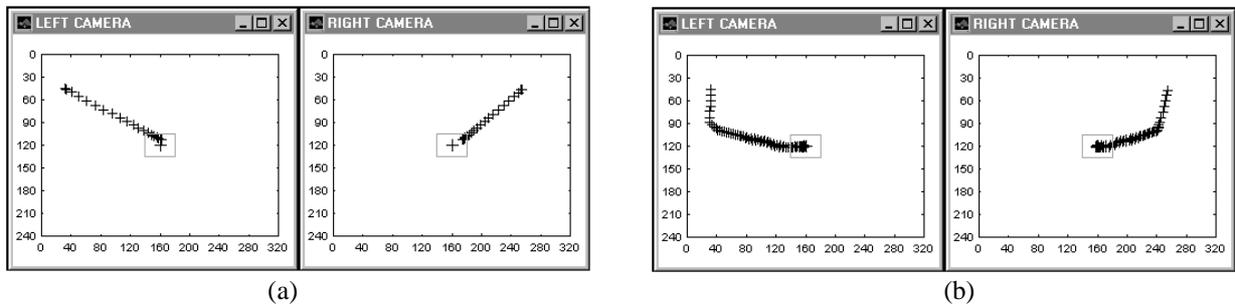


Figura 8: (a) Intercepção em malha aberta a partir da atitude inicial $\mathbf{A}_0 = [-20,70 \ -6,50 \ 18,80]^\circ$, à qual correspondem as coordenadas iniciais $\mathbf{V}_0 = [32 \ 45 \ 225 \ 46] \text{ pixels}$. A atitude de intercepção calculada pela equação (10) foi $\mathbf{A}_F = [-9,66 \ -0,78 \ 11,90]^\circ$, resultando nas coordenadas visuais finais $\mathbf{V}_F = [163 \ 113 \ 174 \ 113] \text{ pixels}$. (b) Intercepção em malha fechada, a partir da mesma atitude inicial anterior. Após 85 iterações, a atitude final de intercepção foi $\mathbf{A}_F = [-9,99 \ -0,20 \ 10,60]^\circ$, resultando $\mathbf{V}_F = [159 \ 120 \ 158 \ 120] \text{ pixels}$.

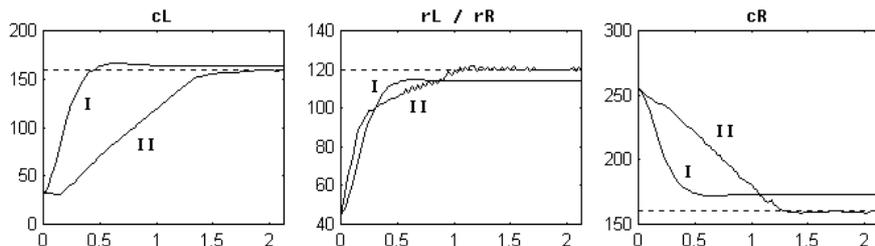


Figura 9: Variação das coordenadas do alvo nas imagens para as tarefas de intercepção da Figura 8(a) (I) e 8(b) (II). A primeira rede neural tende a fornecer uma intercepção mais rápida, porém menos precisa que a segunda.

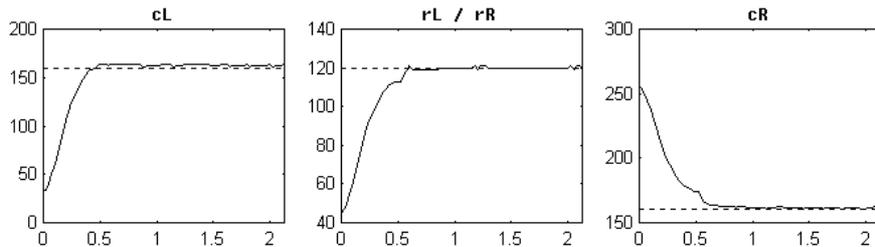


Figura 10: Intercepção usando as duas redes neurais sequencialmente. O sistema é inicialmente submetido (nos primeiros 0,5 segs) à ação de controle \mathbf{A}_d fornecida pela primeira rede, visando obter uma rápida intercepção inicial e em seguida o controle é chaveado para a segunda rede. Isso permite obter uma intercepção rápida e precisa.