Aprendizado e Controle de Robôs Móveis Autônomos Utilizando Atenção Visual

Milton Roberto Heinen [1]
Paulo Martins Engel [1]

Resumo: Este artigo descreve um modelo de aprendizado por reforço capaz de aprender tarefas de controle complexas utilizando ações e estados contínuos. Este modelo, que é baseado no ator-crítico contínuo, utiliza redes de funções de base radial normalizadas para aprender o valor dos estados e das ações, sendo capaz de configurar a estrutura destas redes de forma automática durante o aprendizado. Além disso, um mecanismo de atenção visual seletiva é utilizado para perceber o ambiente e os estados. Para a validação do modelo proposto, foi utilizada uma tarefa relativamente complexa para os algoritmos de aprendizado por reforço: conduzir uma bola até o gol em um ambiente de futebol de robôs simulado. Os experimentos realizados demonstram que o modelo proposto é capaz realizar a tarefa em questão com bastante sucesso utilizando somente informações visuais.

Abstract: This paper describes a reinforcement learning model which is able to learn complex control tasks using continuous states and actions. This model, which is based on continuous actor-critic model, uses normalized radial basis function networks to learn the value function of states and actions, and is able to configure the network structure in an automatic way during the learning process. Besides, a visual selective attention mechanism is used to perceive the environment and the states. To validate the proposed model, a relatively complex task for reinforcement learning algorithms was used: to guide a ball to the goal in a robot soccer simulated environment. The described experiments shows that the proposed model is able to accomplish the task in a very successful way using visual information only.

1 Introdução

Os algoritmos de aprendizado por reforço tradicionais [39] geralmente assumem a existência de um conjunto finito de estados disjuntos, o que é bastante válido para acelerar o aprendizado e permitir o emprego de ferramentas estatísticas com forte base teórica. No

[1] Instituto de Informática, UFRGS, Caixa Postal 15064
{mrheinen,engel@inf.ufrgs.br}

[2] Este trabalho foi parcialmente publicado no SCA 2009, tendo sido selecionado como um dos três melhores artigos do evento.

entanto, esta representação enfrenta dificuldades quando as variáveis de estado são contínuas, como geralmente acontece no mundo real [3, 36]. Em [7, 8] é apresentada uma formulação contínua do algoritmo de diferença temporal $TD(\lambda)$ [38]. Esta formulação utiliza redes de funções de base radial (RBF) [16] normalizadas para aproximar os valores dos estados e aprender as ações. Segundo [8, 33], as redes RBF são mais adequadas de serem utilizadas em aplicações de aprendizado por reforço que as redes MLP [16, 35] pois apresentam uma codificação local dos campos receptivos de entrada, o que evita que o aprendizado em uma região do espaço de entrada destrua o conhecimento adquirido de outras áreas [3]. Entretanto, no algoritmo descrito em [7, 8], as funções radiais são simplesmente distribuídas de modo uniforme no espaço de entradas e mantidas fixas durante o treinamento, o que exige a utilização de informações a priori para a configuração das mesmas.

Neste artigo é proposto um modelo de aprendizado por reforço, inspirado no ator-crítico contínuo de Doya, que consegue criar e posicionar as funções radiais de forma automática e incremental durante o aprendizado. Para mostrar a robustez desta abordagem, o modelo é testado em uma tarefa relativamente complexa: conduzir uma bola até o gol em um ambiente de futebol de robôs simulado. Esta tarefa é especialmente complexa para os algoritmos de aprendizado por reforço porque, além de possuir ações e estados contínuos, a percepção destes últimos ocorre de forma ruidosa e indireta. De fato, nos experimentos realizados o robô recebe informações do ambiente utilizando somente informações visuais fornecidas pelo NLOOK [17–19, 23, 24], que é um modelo de atenção visual capaz de detectar as regiões de interesse do campo visual de forma rápida e eficiente.

Este artigo está estruturado da seguinte forma: a Seção 2 descreve alguns trabalhos relacionados à área em questão; a Seção 3 descreve o ator-crítico contínuo de Doya; a Seção 4 descreve o modelo proposto neste artigo; a Seção 5 descreve os experimentos realizados e os resultados obtidos; e por último a Seção 6 descreve as conclusões finais e perspectivas futuras.

2 Trabalhos relacionados

Esta seção descreve diversos trabalhos da área de aprendizado por reforço que utilizam ações e estados contínuos e/ou informações visuais para a percepção dos estados. Em [2] foram utilizadas informações visuais, em conjunto com o aprendizado por reforço, para o controle de um robô não holonômico (tipo carro) na condução de uma bola até o gol em um ambiente de futebol de robôs. Neste modelo os espaços de estados e ações foram discretizados, e o aprendizado foi conduzido através de missões fáceis (*Learning from Easy Missions* – LEM), o que simplificou em muito o problema original. Em [1], um robô humanóide foi utilizado nesta mesma tarefa, sendo que o controle das juntas realizado através de geradores centrais de padrões (*Central Pattern Generator* – CPG), cabendo ao aprendizado por reforço apenas modular os parâmetros do CPG.

Em [36,37] é proposto o chamado LWR (do inglês *Locally Weighted Regression*), que é uma técnica que permite a utilização ações e estados contínuos em algoritmos de aprendizado por reforço. Apesar de parecer bastante promissor, o LWR é uma técnica de *lazzy learning* ("aprendizado preguiçoso"), pois todos os pontos de dados vistos anteriormente precisam ser armazenados e posteriormente consultados na hora da tomada de decisão. Assim, o LWR não escala muito bem em aplicações de robótica móvel, pois neste tipo de aplicação os dados são abundantes e as decisões precisam ser tomadas com bastante frequência.

Em [32], o modelo de atenção visual VOCUS [12] foi utilizado para a localização de uma bola em um ambiente de futebol de robôs. Em [13–15], este mesmo modelo foi utilizado para a detecção de *landmarks* em tarefas de localização e mapeamento simultâneos (SLAM – *Simultaneous Localization and Mapping*). Além da detecção de *landmarks*, o VOCUS conseguiu controlar de forma automática a câmera móvel do robô, permitindo que a mesma fosse dirigida para as regiões de interesse do campo visual.

Em [30,31], uma rede neural auto-organizável do tipo GTSOM [4] foi utilizada para a categorização das informações sensoriais de um robô móvel (sensores de distância simulados), permitindo assim a utilização de técnicas aprendizado por reforço tradicionais (*Q-Learning*) em um ambiente com estados contínuos simulado. Neste modelo, o robô possui três ações possíveis (virar para a esquerda, virar para a direita ou ir para frente), e uma nova ação é selecionada somente quando ocorre uma transição de estados, indicando que a ação anterior foi concluída. Nos experimentos descritos em [30], este modelo foi utilizado no controle de um robô móvel simulado em um ambiente bidimensional (um labirinto em forma de cruz), no qual o robô conseguiu aprender a melhor trajetória até um determinado alvo. Para evitar problemas de ambiguidade, este modelo precisou utilizar informações de odometria juntamente com um mapa de grade.

Já o modelo proposto neste artigo, descrito na Seção 4, expande a abordagem utilizada em [30,31] em vários aspectos, pois incorpora um modelo de atenção visual na percepção do ambiente e elimina a necessidade de uma mapa de grade e informações de odometria, fazendo com que o robô simulado perceba o ambiente somente através de informações visuais. Além disso, os espaços de estados e ações são ambos contínuos, e o ambiente simulado é muito mais realístico do ponto de vista físico.

3 Ator-crítico contínuo

O aprendizado por reforço tradicional foi fundamentado sobre processos de decisão de Markov e com uma representação finita de estados desconectados [39]. Nesta representação, a estimativa da função de valor de estado normalmente é implementada de forma tabular. Quando se lida com espaços contínuos, a função de valor de estado deve ser implementada por um aproximador que permita generalizar o valor da função para os infinitos estados de

entrada [3].

O ator-crítico [39] é um método de aprendizado por reforço que utiliza dois elementos neurais: um ator e um crítico. O ator implementa a função de controle do agente, e o crítico realiza a estimativa da função de valor de estado. Em [7, 8] é proposta uma versão do ator-crítico na qual as ações e os estados são codificados de forma contínua através de redes de funções de base radial (*Radial Basis Function* – RBF) normalizadas [8], e a interpretação do tempo também é contínua. Nesta versão do ator-crítico, o valor do estado $v(t)$ é aproximado pelo crítico de acordo com a Equação 1, onde $b^V(\cdot)$ são as funções radiais normalizadas do crítico, B^V é o número de funções radiais e w^V são os parâmetros livres do crítico, ou seja, os pesos sinápticos da camada de saída da rede RBF normalizada do crítico.

$$v(t) = \sum_{j=1}^{B^V} w_j^V b_j^V(\mathbf{x}(t)) \tag{1}$$

O crítico é ajustado pelo erro da diferença temporal $\delta(t)$ no instante t através da Equação 2, onde η^V é o passo de atualização do crítico e $e_i(t)$ é o traço de elegibilidade exponencial do peso i, calculado pela Equação 3, na qual k é o passo de desconto da elegibilidade. No ator-crítico de contínuo de Doya, somente os pesos sinápticos da camada de saída são ajustados durante o treinamento.

$$\dot{w}_i^V = \eta^V \delta(t)\, e_i(t) \tag{2}$$

$$\dot{e}_i = \frac{1}{tau^e}\left(\frac{\partial v(t)}{\partial w_i(t)} - e_i(t)\right) \tag{3}$$

O erro da diferença temporal $\delta(t)$ é calculado através da Equação 4, onde $r(t)$ é a recompensa obtida pelo agente ao realizar a ação $\mathbf{u}(t)$ e τ^r é o passo de desconto das recompensas.

$$\delta(t) = r(t) + \tau^r \hat{v}(t) - \hat{v}(t) \tag{4}$$

O sinal de controle $\mathbf{u}(t)$ é calculado pela Equação 5, onde $a(t)$ é a ação gulosa gerada pelo ator no instante t, $\epsilon(t)$ é o termo de exploração, utilizado para lidar com o dilema da exploração-aproveitamento [39], e $\mathbf{n}(t)$ é um vetor de ruído gaussiano normal utilizado para guiar a busca no espaço de estados [8].

$$\mathbf{u}(t) = \tanh\left(a(t) + \epsilon(t)\,\mathbf{n}(t)\right) \tag{5}$$

A ação gulosa $a(t)$ é calculada pela Equação 6, onde $\mathbf{x}(t)$ é o vetor de sinais de estado observado pelo agente no instante t, $b^A(\cdot)$ são as funções radiais normalizadas do ator, B^A é

o número de funções radiais e w^A são os parâmetros livres do ator.

$$a(t) = \sum_{j=1}^{B^A} w_j^A b_j^A(\mathbf{x}(t)) \tag{6}$$

Os parâmetros livres do ator são ajustados através da Equação 7, onde η^A é a taxa de aprendizado do ator e $\mathbf{n}(t)$ é o mesmo vetor de ruído gaussiano utilizado na Equação 5.

$$\dot{\mathbf{w}}^A(t) = \eta^A \delta(t) \frac{\partial \mathbf{u}^A(t)}{\partial \mathbf{w}^A(t)}^T \mathbf{n}(t) \tag{7}$$

Uma das limitações do ator-crítico contínuo de Doya é que somente os pesos sinápticos da camada de saída podem ser ajustados durante o aprendizado. Assim, as funções radiais precisam ser previamente posicionadas de modo a cobrir uniformemente o espaço de entradas, e isto gera os seguintes problemas: (i) o número de funções de base radial precisa ser determinado a priori para que o modelo funcione adequadamente; (ii) os intervalos de valores das entradas precisam ser previamente conhecidos e não podem mudar com o tempo; e (iii) os recursos computacionais não são aproveitados da melhor forma possível, ocorrendo um desperdício em regiões uniformes e pouco relevantes, bem como escassez em outras regiões do espaço de entradas. Já o modelo proposto neste artigo, descrito na próxima seção, consegue criar e posicionar funções radiais de forma automática durante o aprendizado, e assim consegue aproveitar de forma mais eficiente os recursos computacionais sem a necessidade de utilizar conhecimentos a priori.

4 Modelo proposto

A Figura 1 mostra a arquitetura geral do modelo proposto neste artigo, que é baseado no ator-crítico contínuo [7, 8], descrito na seção anterior. Seu funcionamento ocorre da seguinte forma: inicialmente as imagens fornecidas por um dispositivo de carga acoplado (*Charge-Coupled Device* – CCD) são enviadas para o NLOOK (Subseção 4.1), que extrai os pontos mais relevantes (focos de atenção – FOAs) destas imagens. A partir destes FOAs são extraídas informações utilizando a codificação angular de cores [11], na qual os pontos de interesse são codificados utilizado apenas seis valores numéricos bastante descritivos e robustos a mudanças de iluminação.

As informações visuais obtidas com a codificação angular de cores são enviadas para o IGMM (Subseção 4.2), que realiza a formação de agrupamentos sobre os dados de entrada de forma incremental e contínua. Os parâmetros dos agrupamentos (média μ e desvio padrão σ) são utilizados para posicionar as funções radiais do ator e do crítico (cada agrupamento

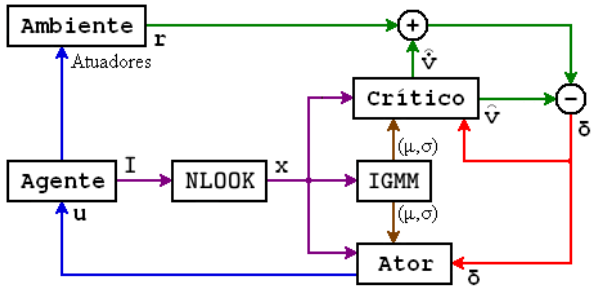


Figura 1. Arquitetura do modelo proposto

corresponde a uma função radial no modelo e no crítico). As redes RBF normalizadas do ator e do crítico são então ativadas, e o robô realiza a ação $\mathbf{u}$ fornecida pelo ator e recebe do ambiente a recompensa imediata r. O erro da diferença temporal $\delta(t)$ (Equação 4 é então calculado e utilizado para ajustar os parâmetros livres do ator e do crítico. O processo se repete até que: (i) o robô consiga conduzir a bola até o gol (sucesso); (ii) a bola saia fora do campo (falha); ou (iii) o tempo de simulação exceda um valor máximo t_{max}.

A função de recompensas é dada pela Equação 8, onde $d_{rb}(t)$ é a distância do robô até a bola e $d_{bg}(t)$ é a distância da bola até o gol no instante t. Os parâmetros $a = 1/4C$ e $b = 2/C$ (onde C é o comprimento do campo) servem para modular a influência dos dois termos da função de recompensas. Além disso, toda vez que o robô fizer um gol, o episódio termina com uma recompensa $r(t) = 1$ por um segundo, e se a bola sair fora do campo o episódio termina com uma recompensa de $r(t) = -1$ por um segundo.

$$\begin{array}{ll} r(t) = a(-d_{rb}(t)) + b(-d_{bg}(t)) & \text{se } d_{bg}(t) > 0 \\ r(t) = 1 & \text{se } d_{bg}(t) \leq 0 \\ r(t) = -1 & \text{se a bola sair fora do campo} \end{array} \quad (8)$$

Como foi descrito anteriormente, em [7, 8] as funções radiais foram uniformemente distribuídas no espaço de entradas, o que torna o aprendizado linear e garante a convergência do aprendizado. Mas de acordo com [3], se as funções radiais forem modificadas de modo significativo durante o treinamento, não é possível garantir a convergência do algoritmo. Entretanto, o modelo proposto neste artigo consegue criar e posicionar as funções radiais de modo adequado durante o treinamento devido às características de aprendizado do IGMM, que fazem com que ele consiga estimar a estrutura dos dados de entrada de forma rápida e eficiente. Em outras palavras, o IGMM consegue aprender de forma incremental, agressiva e não destrutiva, o que segundo [36] é essencial para que um aproximador de funções possa ser utilizado para aprender as funções de valor de estado.

Em [20, 21] uma versão prévia do modelo proposto, baseada na hipótese de independência das variáveis de entrada (*Naïve Bayes*), foi utilizada para conduzir uma bola até o gol utilizando informações fornecidas por sonares simulados. Já o modelo proposto neste artigo, que segue a formulação Bayesiana completa (multivariada), utiliza como dados de entrada informações visuais fornecidas pelo NLOOK. As próximas subseções descrevem cada uma dessas etapas em detalhes.

4.1 NLOOK

O NLOOK [19, 23, 24] é um modelo de atenção visual especialmente desenvolvido para aplicações de robótica móvel. Este modelo, que possui uma excelente desempenho computacional, é bem menos sensível a transformações afins que a maioria dos modelos de atenção existentes, em especial o NVT [27]. Além disso, o NLOOK consegue selecionar a escala aproximada dos focos de atenção (FOAs) em conjunto com as coordenadas espaciais, e isso faz com que ele possa ser utilizado como um *front-end* em aplicações de visão computacional e robótica.

O funcionamento do NLOOK ocorre da seguinte forma: Inicialmente a imagem de entrada é decomposta em três conjuntos de mapas de características (intensidade, cores e orientações) que operam em paralelo sobre toda a cena visual. Para a criação dos mapas de intensidade, a imagem original é inicialmente convertida em imagem em tons de cinza I, e diferenças de gaussianas (DoG) são geradas utilizando *scale-spaces* [29, 40]. Para a criação dos mapas de cores, inicialmente são gerados quatro *scale-spaces* para os canais de cores R (vermelho), G (verde), B (azul) e Y (amarelo). Em seguida são geradas diferenças de gaussianas entre diferentes *scale-spaces*, ou seja, $R - B$ e $R - B$. Para a criação dos mapas de orientação são utilizados filtros de Gabor [6] com orientações preferenciais $\theta \in \{0°, 45°, 90°, 135°\}$. Em [17, 23, 24] é descrita a criação destes mapas em maiores detalhes.

Após a criação dos mapas características, eles são normalizados e unidos (somados) em um único *scale-space* de saliências. Os mapas deste *scale-space* são então redimensionados para a escala 0 (tamanho original da imagem) e somados, formando assim um único mapa de saliências. Este mapa de saliências é então percorrido pelo foco de atenção da seguinte forma: inicialmente o ponto mais saliente do mapa é encontrado, e o *scale-space* de saliências é analisado para se descobrir a escala característica deste FOA [28]. O raio da região de interesse é então calculado através da equação [5]:

$$r_{roi} = 2^{(o-1)} \times k_s \times b^{(s+\hat{s})} \tag{9}$$

onde o e s são o oitavo e a escala característica do FOA, e a constante $k_s = 1{,}6$ é um fator de correção empírico para a escala dado pela progressão geométrica com base $b = \sqrt{2}$. Um mecanismo de inibição de retorno, que possui o formato de uma gaussiana invertida, é então aplicado sobre o mapa de saliências único, sendo o diâmetro calculado pela Equação 9.

Através de diversos experimentos utilizando imagens reais e sintéticas, descritos detalhadamente em [17, 22–24], foi verificado que o NLOOK é capaz de selecionar as regiões mais relevantes do campo visual de forma bastante precisa, sendo bastante robusto a transformações afins. Além disso, o NLOOK consegue selecionar a escala aproximada dos FOAs, o que o torna bastante útil em aplicações de visão computacional e robótica.

4.2 IGMM

O IGMM (*Incremental Gaussian Mixture Model*) [10] é um algoritmo baseado em técnicas de aprendizado não-supervisionado incremental para formação de conceitos a partir de instâncias do domínio descritas por atributos contínuos. O algoritmo IGMM opera sucessivamente sobre cada dado, mantendo estimativas atualizadas dos modelos dos agrupamentos correntes. Usando o modelo corrente, o algoritmo decide se é necessário criar um novo agrupamento para o dado apresentado ao sistema. O foco do IGMM é o chamado aprendizado incremental, utilizado principalmente em tarefas que lidam com dados que estão disponíveis apenas instantaneamente para o sistema de aprendizado. Neste caso, o sistema de aprendizado deve agir imediatamente, levando em consideração o dado atual para atualizar o seu modelo. Assim, o IGMM propõe uma solução para o problema do aprendizado incremental, considerando-o como uma aproximação para os métodos de aprendizado que dispõem do conjunto completo de dados no início do processo de aprendizado.

Em [9] é apresentada a primeira versão anterior do IGMM, chamada de INBC, que seguia a hipótese Bayesiana ingênua (*Naïve Bayes*). Já a versão do IGMM utilizada neste artigo segue a hipótese bayesiana completa, ou seja, utiliza gaussianas multivariadas para a definição dos agrupamentos de dados. Nesta abordagem, a tarefa de formação de agrupamentos é formulada como um problema de identificação das probabilidades de um modelo de mistura particular, formado por uma combinação linear de k probabilidades correspondentes a processos probabilísticos independentes, $prob(\mathbf{x}, j)$:

$$prob(\mathbf{x}) = \sum_{j=1}^{k} prob(\mathbf{x}|j)p_j \tag{10}$$

Os parâmetros p_j são chamados de parâmetros de mistura e estão relacionados com a probabilidade a priori de $\mathbf{x}$ ter sido gerado pela componente j da mistura. Para atributos contínuos, cada componente $\mathbf{x}$ de uma distribuição j é modelada por uma gaussiana multidimensional caracterizada pela média $\boldsymbol{\mu}$ e a matriz de variâncias/covariâncias $\mathbf{C}$ conforme:

$$p(\mathbf{x}|j) = \frac{1}{(2\pi)^{D/2}\sqrt{|\mathbf{C}_j|}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T\mathbf{C}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)\right\} \tag{11}$$

Uma importante contribuição do IGMM está na formulação de um procedimento incremental para a atualização dos parâmetros do modelo de mistura que representa o problema de aprendizado, que é vista como um processo de aproximação dos estimadores estatísticos. Além disso, o IGMM não necessita que os mesmos dados sejam apresentados de forma repetida, ou seja, ele consegue aprender com apenas uma iteração sobre os dados.

Um outro aspecto importante do IGMM é a formação incremental de agrupamentos. A cada apresentação de um vetor de dados ao sistema, o algoritmo utiliza o modelo probabilístico corrente para decidir se o novo dado deve ser incorporado à configuração de agrupamentos atual, ou se este dado deve originar um novo agrupamento. A decisão é tomada em relação a um limiar de probabilidade mínima aceitável para que um vetor de dados seja considerado como pertencente a um dos componentes da mistura. A condição para criação de uma nova componente da mistura é: se $prob(\mathbf{x}|j) < \tau_{nov} \ \ \forall j$, então uma nova componente é criada. O limiar de probabilidade mínima aceitável τ_{nov}, cuja principal função é definir a granularidade do modelo, é o único parâmetro que precisa ser configurado no IGMM. Valores pequenos ($\tau_{nov} << 0{,}01$) fazem com que poucos agrupamentos sejam criados. Valores mais elevados ($\tau_{nov} >> 0{,}01$) podem levar a criação de muitas componentes espúrias. Na prática o valor $\tau_{nov} = 0{,}01$ pode ser utilizado sem problemas na maioria dos casos.

As principais vantagens do IGMM que o tornam útil para o problema em questão são: (i) o IGMM consegue criar os agrupamentos de forma incremental e contínua sem a necessidade de se apresentar previamente todo o conjunto de dados de treinamento; (ii) os parâmetros do modelo de mistura são ótimos do ponto de vista probabilístico, ou seja, a hipótese fornecida é sempre a de maior verossimilhança em relação aos dados fornecidos até o momento; (iii) sempre que novos dados estiverem disponíveis, estes podem ser apresentados ao modelo sem que os conhecimentos adquiridos anteriormente sejam perdidos; (iv) o IGMM consegue aprender as distribuições dos dados de entrada de forma agressiva, sem a necessidade de analisar os mesmos de forma repetida; e (v) o IGMM possui um ótimo desempenho computacional que o torna adequado de ser utilizado em aplicações de tempo real. Mais detalhes sobre o IGMM podem ser encontrados em [9, 10].

4.3 Robô e ambiente modelados

Os experimentos descritos neste artigo (Seção 5) foram realizados através da utilização de um robô e um ambiente simulados. Para que uma simulação de robôs móveis seja realista, diversos elementos do mundo real precisam estar presentes no modelo de simulação, para que os corpos se comportem de forma similar à realidade. Para que isto ocorra, é necessário que as leis da física sejam modeladas no ambiente de simulação (gravidade, inércia, fricção e colisão) [25, 34]. Atualmente existem várias bibliotecas de software disponíveis para a implementação de simulações baseadas em física. Após o estudo de diversas possibilidades, optou-se pela utilização de uma biblioteca de código aberto e gratuita chamada *Open*

Dynamics Engine (ODE)[2], que permite a realização de simulações da dinâmica de corpos rígidos articulados com bastante realismo. Para a movimentação das rodas de um robô, é possível que sejam utilizados os motores angulares que estão disponíveis no ambiente ODE.

O ambiente de simulação utilizado segue as regras da liga de robôs de pequeno porte (F180 *League* 2009) da Robocup[3], como mostra a Figura 2. O campo possui 605cm de comprimento por 405cm de largura, a goleira possui 18cm de altura por 70cm de largura, as traves possuem um diâmetro de 2cm e a bola possui 4,3cm de diâmetro. Conforme as especificações da liga, o campo possui uma textura verde similar a um gramado, as goleiras são brancas e a bola de cor alaranjada. O robô modelado possui o formato de uma caixa com 5,5cm de comprimento e largura e 3cm de altura, duas rodas de 1,25cm de diâmetro e cinemática diferencial. Estas dimensões são similares às do robô móvel Khepera, porém o corpo foi modelado no formato de uma caixa (o Khepera possui um corpo cilíndrico) para facilitar a tarefa de conduzir a bola ao gol. No topo do robô existe uma câmera simulada que envia imagens do ambiente para os módulos de visão e controle. Estas imagens foram sintetizadas com o auxílio da biblioteca OpenGL, e possuem diversos elementos realísticos como sombras, texturas do gramado e de nuvens em movimento, permitindo assim um certo realismo das imagens, porém sem chegar ao nível fornecido por uma câmera real. A Figura 2 mostra três ângulos diferentes do ambiente e do robô simulados. Em [26] uma versão anterior deste simulador foi utilizada para o controle do caminhar de robôs com pernas.

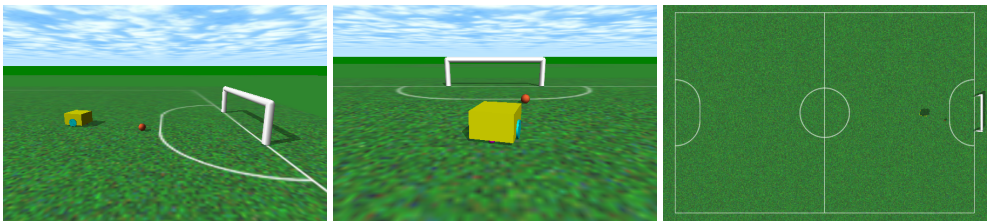


Figura 2. Detalhes do ambiente e do robô simulado

5 Experimentos realizados e resultados obtidos

Esta seção descreve os resultados obtidos com o protótipo do modelo proposto na tarefa de conduzir a bola até o gol em um ambiente de futebol de robôs simulado. Nos experimentos realizados, o aprendizado ocorre em 2500 episódios distintos. Cada episódio inicia com o robô sempre na mesma posição, mas para evitar que o robô simplesmente "decore"

[2]ODE – `http://www.ode.org`
[3]Robocup – `http://www.robocup.org/`

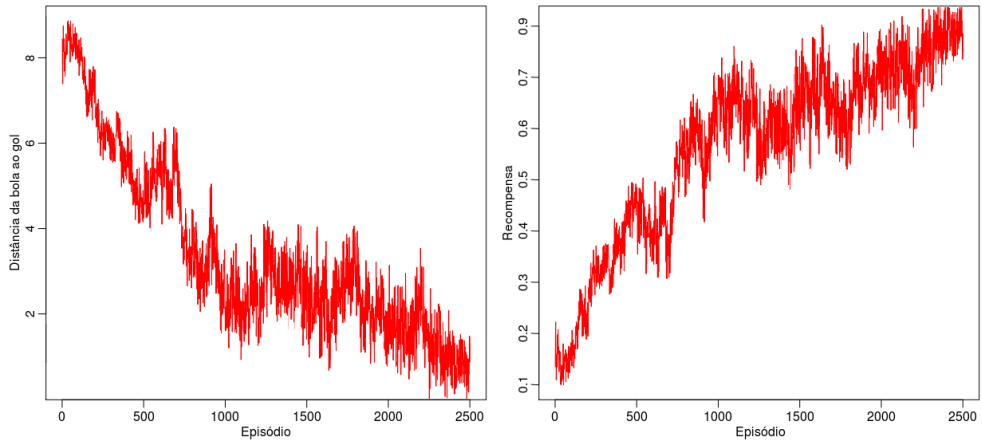
a trajetória desejada sem fazer uso das informações sensoriais, a posição da inicial da bola varia a cada episódio. Assim, para obter sucesso na tarefa o robô precisa: (i) identificar e localizar a bola utilizando somente informações visuais; (ii) se deslocar em direção à bola; e (iii) conduzir a bola até o gol sem deixar a mesma escapar. Um episódio termina sempre que a bola sair fora do campo, o robô conseguir fazer um gol ou o tempo de simulação exceder o limite t_{max}. Neste caso, um novo episódio se inicia com a bola posicionada em outro local e o robô colocado de volta à posição original.

Com relação às redes RBF do ator e do crítico, foram utilizadas seis entradas (que recebem os seis valores numéricos da codificação angular de cores), uma saída no crítico ($\hat{v}(t)$) e duas no ator, que correspondem as ativações dos motores das duas rodas laterais do robô. O número de funções radiais varia durante o aprendizado, começando com apenas uma função radial e aumentando sempre que necessário. Os parâmetros do aprendizado por reforço utilizados são os mesmos descritos em [8], ou seja: $\tau^r = 1$; $\tau^e = 0{,}1$; $\tau^n = 1$; $epsilon_0 = 0{,}5$; $v_0 = 0$; $v_1 = 1$; $\eta^V = 1$; e $\eta^A = 5$. O tempo máximo t_{max} é de 150 segundos, e o passo de tempo Δt é de 0,05 segundos.

Para a avaliação dos resultados, duas medidas de desempenho foram utilizadas: (a) a distância da bola até o gol ao final do episódio (zero quando o robô fizer um gol); e (b) recompensa imediata recebida ao final do episódio (1 quando fizer o gol). Devido a natureza estocástica do modelo proposto, cada experimento foi replicado 20 vezes usando números aleatórios diferentes, e a Figura 3 mostra a média dos resultados obtidos nestes experimentos. O número médio de categorias geradas pelo IGMM foi de 173,51. Pelo fato do modelo proposto ser de aprendizado perpétuo, os resultados apresentados levam em conta a performance do modelo durante o aprendizado, pois de fato o modelo não possui duas fases distintas de treinamento e utilização.

Analisando o gráfico da Figura 3(a), percebe-se que a média das distâncias da bola até o gol se reduzem gradativamente, o que indica que ao final do treinamento o robô foi capaz de conduzir a bola até o gol na maioria dos episódios (as variações se devem tanto à inicialização aleatória da bola quanto ao ruído de exploração). Já o gráfico da Figura 3(b) mostra que a recompensa recebida pelo robô ao final de cada episódio se eleva gradativamente, o que indica que o robô foi capaz de aprender uma política capaz de maximizar as recompensas recebidas, o que é um claro indício da convergência do modelo proposto.

A Figura 4 mostra a visão que o robô tem durante a execução da tarefa, bem como os dez pontos de maior interesse (FOAs) a cada dois segundos. A título de demonstração, a Figura 5 mostra este mesmo experimento visto de um ângulo diferente que o robô não tem acesso (ele só percebe o ambiente através de sua própria câmera), no caso com a câmera virtual localizada na lateral do campo. O local da bola é mostrado no primeiro quadro com um círculo em vermelho.



(a) Distância da bola ao gol em metros

(b) Recompensa média recebida pelo robô

Figura 3. Resultados obtidos nos experimentos

Apesar do modelo proposto ter conseguido realizar a tarefa em questão com bastante sucesso, cabe ressaltar que as imagens obtidas em um ambiente real são muito mais complexas, o que pode dificultar o processo de aprendizado. Entretanto, acredita-se que em um ambiente interno controlado (que é o caso no futebol de robôs) o modelo proposto ainda consiga realizar a tarefa com sucesso, pois o NLOOK foi amplamente testado com imagens reais e sintéticas [22–24], tento apresentado excelentes resultados em ambos os casos.

Cabe salientar que o principal objetivo destes experimentos não é simplesmente criar um jogador de futebol de robôs eficiente (para isto uma abordagem pré-programada seria suficiente), mas provar: (i) a eficiência do aprendizado por reforço utilizando ações e estados contínuos; e (ii) que um modelo de atenção visual pode ser utilizado de forma efetiva em aplicações de robótica móvel. Com relação aos tempos de execução, o modelo proposto é bastante eficiente, conseguindo realizar cada experimento completo (todos os 2500 episódios) em aproximadamente 34,5 minutos em um computador típico[4].

6 Conclusões e perspectivas

Este artigo descreveu um modelo de aprendizado por reforço que é capaz de realizar tarefas de controle complexas utilizando ações e estados contínuos. Este modelo utiliza redes

[4]Computador Dell Optiplex 755, Processador Intel(R) Core(TM)2 Duo CPU 2,33GHz, 1,95GB de memória RAM e Sistema Operacional Debian Linux.

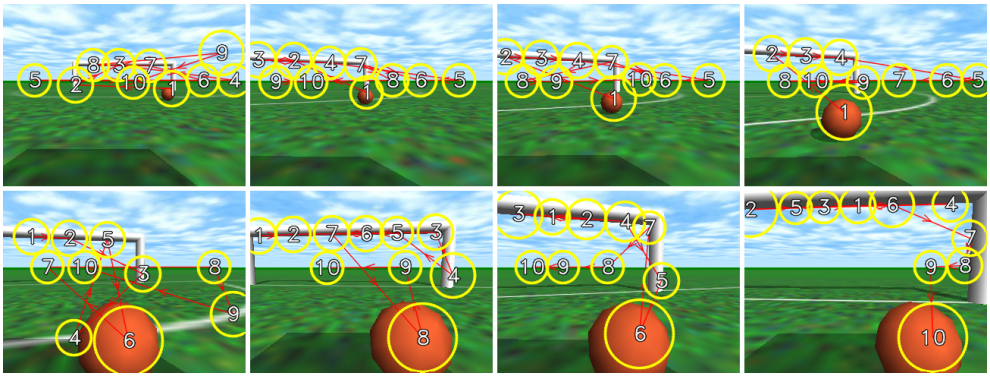


Figura 4. Visão obtida pelo robô no ambiente simulado

RBF normalizadas para aprender os valores dos estados e as ações, conseguindo inclusive alterar a estrutura das redes durante o aprendizado sem a necessidade de utilizar informações a priori. Além disso, o modelo proposto consegue realizar o aprendizado utilizando apenas informações visuais obtidas através de um mecanismo de atenção visual seletiva. Para a validação do modelo proposto, foi utilizada uma tarefa de controle bastante complexa para os algoritmos de aprendizado por reforço: conduzir uma bola até o gol em um ambiente de futebol de robôs simulado. Os resultados obtidos demonstram que o robô foi capaz aprender a tarefa de forma bastante eficiente. As perspectivas futuras incluem substituir o ambiente simulado por um ambiente real de futebol de robôs da liga de tamanho médio (*Middle Size Robot League*) da Robocup, bem como utilizar um robô real do tipo Pioneer 3-DX e imagens obtidas a partir de uma câmera na realização da tarefa em questão.

Agradecimentos

Agradecemos ao apoio do CNPq que tornou possível a realização deste trabalho.

Referências

[1] M. Asada, Y. Katoh, M. Ogino, and K. Hosoda. A humanoid approaches to the goal - reinforcement learning based on rhythmic walking parameters. In *Proc. 7th Int. RoboCup Symposium*, volume 3020 of *LNCS*, pages 344–354, Padua, Italy, July 2003. Springer-Verlag.

[2] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, 23:279–303, 1996.

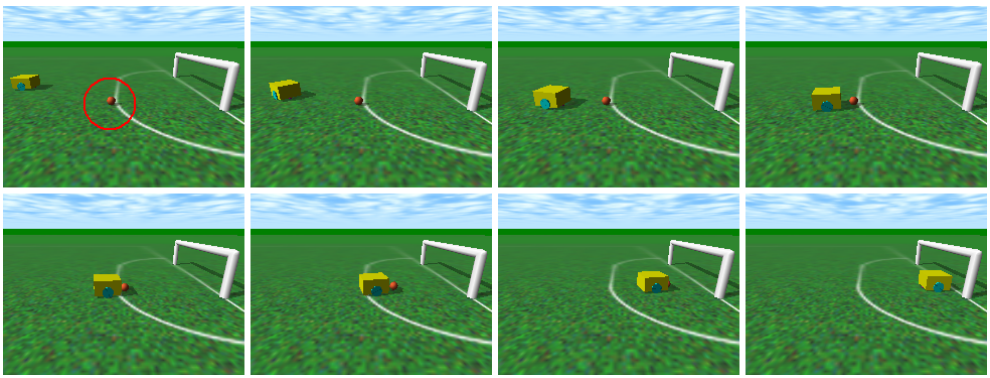


Figura 5. Trajetória do robô vista por uma câmera lateral

[3] E. W. Basso and P. M. Engel. Reinforcement learning in non-stationary continuous time and space scenarios. In *Anais do VII Encontro Nacional de Inteligência Artificial (ENIA)*, Bento Gonçalves, RS, 2009. SBC Editora.

[4] E. N. F. Bastos. Uma rede neural auto-organizável construtiva para aprendizado perpétuo de padrões espaço-temporais. Master's thesis, UFRGS, Porto Alegre, RS, Aug. 2007.

[5] J. Crowley, O. Riff, and J. Piater. Fast computation of characteristic scale using a half octave pyramid. In *Proc. Int. Workshop on Cognitive Vision*, Zurich, Switzerland, 2002. Springer-Verlag.

[6] J. G. Daugman. Complete discrete 2-d gabor transforms by neural networs for image analysis and compression. *IEEE Trans. Acoustics, Speech, and Signal Proc.*, 36(7):1169–1179, July 1988.

[7] K. Doya. Temporal difference learning in continuous time and space. *Advances in Neural Information Processing Systems*, 8:1073–1079, 1996.

[8] K. Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12(1):219–245, 2000.

[9] P. M. Engel. INBC: An incremental algorithm for dataflow segmentation based on a probabilistic approach. Technical Report RP-360, UFRGS, Porto Alegre, RS, May 2009.

[10] P. M. Engel and M. R. Heinen. IGMM: An incremental approach for learning gaussian mixture models from data flows. 2009. In preparation.

[11] G. D. Finlayson, S. S. Chatterjee, and B. V. Funt. Color angular indexing. In *Proc. 4th European Conf. in Computer Vision (ECCV'96)*, pages 16–27, Cambridge, UK, 1996. Springer-Verlag.

[12] S. Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*. Ph.d. dissertation, Rheinische Friedrich-Wilhelms-Universitat Bonn, Bonn, Germany, Jan. 2006.

[13] S. Frintrop and P. Jensfelt. Active gaze control for attentional visual SLAM. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA'08)*, Pasadena, CA, May 2008. IEEE Press.

[14] S. Frintrop and P. Jensfelt. Attentional landmarks and active gaze control for visual SLAM. *IEEE Trans. Robotics, Special Issue on Visual SLAM*, 24(5), Oct. 2008.

[15] S. Frintrop, P. Jensfelt, and H. Christensen. Simultaneous robot localization and mapping based on a visual attention system. *Attention in Cognitive Systems*, 4840, 2007.

[16] S. Haykin. *Neural Networks and Learning Machines*. Prentice-Hall, Upper Saddle River, NJ, 3 edition, 2008.

[17] M. R. Heinen and P. Engel. Avaliação de modelos de atenção visual em relação a transformações afins. In *Proc. IV Workshop de Visão Computacional (WVC)*, Bauru, SP, Nov. 2008. IEEE Press.

[18] M. R. Heinen and P. M. Engel. NLOOK: Modelo de atenção visual relativamente insensível a transformações afins. *Hífen*, 32(62):270–277, 2008.

[19] M. R. Heinen and P. M. Engel. Visual selective attention model for robot vision. In *Proc. 5th IEEE Latin American Robotics Symposium (LARS'08)*, Salvador, BH, Oct. 2008.

[20] M. R. Heinen and P. M. Engel. Aprendizado autônomo de robôs móveis simulados em ambientes contínuos. In *Proc. XXXV Latin American Informatics Conf. (CLEI)*, Pelotas, RS, Sept. 2009.

[21] M. R. Heinen and P. M. Engel. Aprendizado de robôs móveis autônomos em ambientes simulados contínuos. In *Anais do IX Congr. Brasileiro de Redes Neurais / Inteligência Computacional (CBRN)*, Ouro Preto, MG, Oct. 2009.

[22] M. R. Heinen and P. M. Engel. Categorização de objetos utilizando atenção visual. In *Anais do IX Congr. Brasileiro de Redes Neurais / Inteligência Computacional (CBRN)*, Ouro Preto, MG, Oct. 2009.

[23] M. R. Heinen and P. M. Engel. Evaluation of visual attention models under 2d similarity transformations. In *Proc. 24rd ACM Symposium on Applied Computing (SAC 2009) – Special Track on Intelligent Robotic Systems*, pages 1156–1160, Honolulu, Hawaii, Mar. 2009. ACM press.

[24] M. R. Heinen and P. M. Engel. NLOOK: A computational attention model for robot vision. *Journal of the Brazilian Computer Society (JBCS)*, page 15, Sept. 2009.

[25] M. R. Heinen and F. S. Osório. Co-evolução da morfologia e controle de robôs móveis simulados utilizando realidade virtual. In *Proc. IX Symposium on Virtual and Augmented Reality*, pages 187–196, Petrópolis, RJ, May 2007. SBC Editora.

[26] M. R. Heinen and F. S. Osório. Evolving gait control of physically based simulated robots. *Revista de Informática Teórica e Aplicada (RITA)*, XVI(1):119–134, 2007.

[27] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, Nov. 1998.

[28] T. Lindeberg. Feature detection with automatic scale selection. *Int. Journal of Computer Vision*, 30(2):79–116, 1998.

[29] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, Jan. 2004.

[30] M. Menegaz. Aplicação da rede GTSOM para navegação de robôs móveis utilizando aprendizado por reforço. Master's thesis, Instituto de Informática – UFRGS, Porto Alegre, RS, Mar. 2009.

[31] M. Menegaz and P. M. Engel. Using the gtsom network for mobile robot navigation with reinforcement learning. In *Proceedings of the IEEE Int. Joint Conf. Neural Networks (IJCNN)*, Atlanta, GA, June 2009. IEEE Press.

[32] S. Mitri, S. Frintrop, K. Pervolz, H. Surmann, and A. Nuchter. Robust object detection at regions of interest with an application in ball recognition. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 122–131, Barcelona, Spain, Apr. 2005. IEEE Press.

[33] J. Morimoto and K. Doya. Robust reinforcement learning. *Neural Comp.*, 17(2):335–359, 2005.

[34] F. Osório, S. Musse, R. Vieira, M. R. Heinen, and D. Paiva. *Increasing Reality in Virtual Reality Applications through Physical and Behavioural Simulation*, volume 2 of *Research in Interactive Design – Proc. Virtual Concept Conf. 2006*, pages 1–45. Springer-Verlag, Berlin, Germany, 2006.

[35] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning Internal Representations by Error Propagation*. MIT Press, Cambridge, MA, 1986.

[36] W. D. Smart. *Making Reinforcement Learning Work on Real Robots*. PhD thesis, Brown Univ., Providence, Rhode Island, May 2002.

[37] W. D. Smart and L. P. Kaelbling. Practical reinforcement learning in continuous spaces. In *Proc. 17th Int. Conf. Machine Learning (ICML'2000)*, pages 903–910, 2000.

[38] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

[39] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[40] A. P. Witkin. Scale-space filtering. In *Proc. Int. Joint Conf. Artificial Intelligence*, pages 1019–1022, Karlsruhe, Germany, 1983. Morgan Kaufman.