Anais do IX Congresso Brasileiro de Redes Neurais / Inteligência Computacional (IX CBRN)
Ouro Preto 25-28 de Outubro de 2009 

# APRENDIZADO DE ROBÔS MÓVEIS AUTÔNOMOS EM AMBIENTES SIMULADOS CONTÍNUOS

MILTON ROBERTO HEINEN*, PAULO MARTINS ENGEL*

*Instituto de Informática, Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15064, 91501-970 Porto Alegre, RS, Brasil

Emails: mrheinen@inf.ufrgs.br, engel@inf.ufrgs.br

**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 a função de valor dos estados e as ações. Além disso, o modelo proposto consegue configurar a estrutura das redes de forma automática durante o aprendizado. 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 resultados demonstram que o modelo é capaz realizar a tarefa utilizando apenas informações sensoriais.

**Palavras-chave—** Robôs Móveis Autônomos, Aprendizado de Máquina, Aprendizado por Reforço.

## 1 Introdução

Os algoritmos de aprendizado por reforço tradicionais (Sutton and Barto, 1998) 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 entanto, esta representação enfrenta dificuldades quando as variáveis de estado são contínuas, como geralmente acontece no mundo real (Basso, 2009). Em (Doya, 1996; Doya, 2000) é apresentada uma formulação contínua do algoritmo de diferença temporal $TD(\lambda)$ (Sutton, 1988). Esta formulação utiliza redes de funções de base radial (RBF) (Haykin, 2001) normalizadas para aproximar os valores dos estados e aprender as ações. As redes RBF são mais adequadas às técnicas de aprendizado por reforço que as redes MLP (Rumelhart et al., 1986; Haykin, 2001) 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 (Basso, 2009). Entretanto, no algoritmo descrito em (Doya, 1996; Doya, 2000), 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, mas que consegue criar e posicionar as funções radiais de modo automático 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 espaços contínuos, a percepção dos estados utilizada é imprecisa e indireta, ou seja, o agente precisa estimar os mesmos a partir de dados ruidosos fornecidos pelos seus próprios sensores (Asada et al., 2003). Este artigo está estruturado da seguinte forma: a Seção 2 descreve o ator-crítico contínuo; a Seção 3 descreve o modelo proposto neste artigo; a Seção 4 descreve os resultados obtidos; e a Seção 5 descreve as conclusões e perspectivas.

## 2 Ator-crítico contínuo

O aprendizado por reforço tradicional foi fundamentado sobre processos de decisão de Markov e com uma representação finita de estados desconectados (Sutton and Barto, 1998). 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 (Basso, 2009).

O ator-crítico (Sutton et al., 1983) 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 (Doya, 1996; Doya, 2000) 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 (RBF) normalizadas (Doya, 2000), 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.

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

O crítico é ajustado pelo erro da diferença temporal $\delta(t)$ no instante $t$ através da Equação 2, onde $\eta^V$ 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 $\tau^r$ 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 (Sutton and Barto, 1998), e $\mathbf{n}(t)$ é um vetor de ruído gaussiano normal utilizado para guiar a busca no espaço de estados (Doya, 2000).

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

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 $\eta^A$ é a taxa de aprendizado do ator e $\mathbf{n}(t)$ é o mesmo vetor de ruído da 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 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 conhecimento a priori.

## 3 Modelo proposto

A Figura 1 mostra a arquitetura geral do modelo proposto neste artigo, que é baseado no ator-crítico contínuo (Doya, 1996; Doya, 2000) descrito na seção anterior. Inicialmente os dados sensoriais $\mathbf{x}$ fornecidos por um robô simulado são enviados para o INBC (Subseção 3.1), que realiza a formação de agrupamentos sobre os dados de entrada de forma incremental e contínua. Os parâmetros dos agrupamentos (média $\mu$ e desvio padrão $\sigma$) são utilizados para posicionar as funções radiais do ator e do crítico (cada agrupamento 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 (término por sucesso); (ii) a bola saia fora do campo (término por falha); ou (iii) o tempo de simulação exceda um valor máximo $t_{max}$.

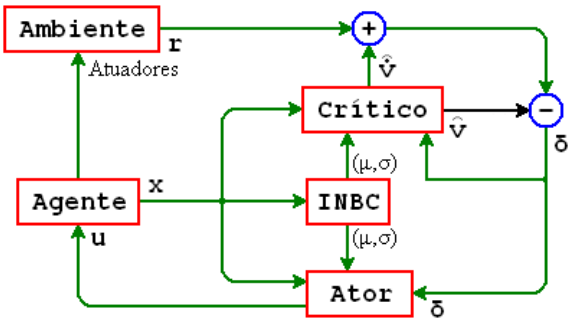


Figura 1: Arquitetura do modelo proposto

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 $r(t) = 10$ por um segundo, e se a bola sair fora do campo o episódio termina com uma recompensa de $r(t) = -10$ por um segundo.

$$\begin{array}{ll} r(t) = a(-d_{rb}(t)) + b(-d_{bg}(t)) & \text{se } d_{bg}(t) > 0 \\ r(t) = 10 & \text{se } d_{bg}(t) \leq 0 \\ r(t) = -10 & \text{se bola fora} \end{array} \tag{8}$$

Como foi descrito anteriormente, em (Doya, 1996; Doya, 2000) as funções radiais foram uniformemente distribuídas no espaço de entradas, o que torna o aprendizado linear e assim garante a convergência do aprendizado. Mas de acordo com (Basso and Engel, 2009; Basso, 2009), 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 INBC, que fazem com que ele consiga estimar a estrutura dos dados de entrada de forma rápida e eficiente. Estas características são descritas na próxima subseção.

### *3.1 INBC*

O INBC (*Incremental Naïve Bayes Clustering*) (Engel, 2009) 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 e discretos. O algoritmo INBC 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. A formulação do algoritmo está baseada na hipótese da independência entre as variáveis que descrevem o domínio, equivalente à hipótese bayesiana ingênua, ou *Naïve Bayes* (NB).

O foco do INBC 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 INBC 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. 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 independentes, $prob(\mathbf{x}, j)$:

$$prob(\mathbf{x}) = \sum_{j=1}^{k} prob(\mathbf{x}|j)p_j \qquad (9)$$

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 $j$. Para atributos contínuos, cada componente $x_i$ de uma distribuição $j$ é modelada por uma gaussiana unidimensional.

Uma importante contribuição do INBC está na formulação de um procedimento incremental para a atualização dos parâmetros do modelo de mistura que representa o problema de aprendizado. A atualização dos parâmetros é vista como um processo de aproximação dos estimadores estatísticos levando em conta a hipótese da independência das variáveis. Além disso, o INBC não necessita que os mesmos dados sejam apresentados de forma repetida para que ocorra o aprendizado, ou seja, ele aprende com apenas uma iteração sobre os dados.

Um outro aspecto importante do INBC 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 $prob(\mathbf{x}_t|j) < \eta \ \ \forall j$, então uma nova componente é criada. O limiar de probabilidade mínima aceitável $\eta$ é o único parâmetro que precisa ser configurado no INBC, e de acordo com (Engel, 2009), na maioria dos casos o valor padrão de 0,05 pode ser utilizado sem maiores problemas. Mais informações sobre o INBC são encontradas em (Engel, 2009).

As principais vantagens do INBC que o tornam útil para o problema em questão são: (i) o INBC 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 INBC consegue aprender as distribuições dos dados de entrada sem a necessidade de analisar os mesmos de forma repetida; e (v) o INBC possui um ótimo desempenho computacional que o torna adequado de ser utilizado em aplicações de tempo real.

Assim, quando utilizado em conjunto com o ator-crítico contínuo, o INBC consegue rapidamente criar e posicionar as funções radiais nos locais mais adequados do espaço de entradas sob o ponto de vista estatístico, e assim consegue aproveitar melhor os recursos de processamento das redes RBF. Além disso, a sua rápida convergência evita que as funções radiais se alterem de forma significativa durante o treinamento, e isto torna possível a convergência do aprendizado por reforço. De fato, utilizando o INBC as funções radiais estabilizam muito antes do modelo aprender a função de valor $V(\cdot)$, o que evita que o conhecimento adquirido seja destruído durante o aprendizado.

### *3.2 Robô e ambiente modelados*

Os experimentos descritos neste artigo (Seção 4) foram realizados através da utilização de um robô e ambiente simulados. Para que uma simulação de robôs móveis seja realista, é necessário que as leis da física (gravidade, inércia, fricção e colisão) sejam modeladas no ambiente de simulação (Osório et al., 2006), o que é

possível através da biblioteca ODE[1] (*Open Dynamics Engine*), que é uma biblioteca de simulação baseada em física que permite a construção de ambientes simulados com bastante realismo do ponto de vista físico (Heinen and Osório, 2006; Heinen and Osório, 2007).

O ambiente de simulação utilizado segue as regras da liga de robôs de médio porte da Robocup[2]. O campo possui 18 metros de comprimento por 12 metros de largura, a goleira possui 1 metro de altura por 2 metros de largura, as traves possuem um diâmetro de 12,5cm, e a bola possui 70cm de circunferência e 450 gramas de peso. 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. Para facilitar a percepção dos limites do campo utilizando sensores do tipo sonar, foram instaladas paredes de um metro de altura a um metro de distância dos limites do campo. O robô modelado possui o formato de uma caixa com 44,5cm de comprimento, 39,3cm largura e 23,7cm de altura, 9kg de peso, duas rodas de 19,53cm de diâmetro e cinemática diferencial. Estas dimensões são equivalentes às do robô móvel Pioneer 3-DX, presente em nossos laboratórios e que será utilizado futuramente na realização da tarefa. Uma versão prévia deste ambiente de simulação foi utilizada em (Heinen and Engel, 2009).

Para a percepção do ambiente, foram utilizados sensores do tipo sonar simulados com bastante realismo utilizando uma técnica de RayCast. Ao todo foram utilizados 8 sensores, posicionados nos mesmos locais dos sonares robô Pioneer 3-DX: um em cada lado e seis na frente, dispostos em intervalos de 20°. A faixa de sensibilidade dos sensores varia de dez centímetros até quatro metros. Todos os demais parâmetros sensoriais foram configurados de acordo com as especificações do robô Pioneer 3-DX presentes em seu manual técnico[3]. Através de diversos experimentos preliminares, não descritos aqui por questões de espaço, foi constatado que os dados fornecidos pelos sensores simulados são bastante próximos aos obtidos com os sensores reais, o que garante a confiabilidade dos resultados obtidos, descritos na próxima seção.

## 4 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 1000 episódios distintos. Cada episódio inicia com a bola posicionada aleatoriamente (mas dentro do raio de alcance dos sensores do robô), de modo a evitar que o robô simplesmente decore a trajetória desejada sem fazer uso das informações sensoriais. Assim, para obter sucesso na tarefa o robô precisa: (i) localizar a bola utilizando somente informações sensoriais; (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 oito entradas (uma para cada sonar do robô), uma saída no crítico ($\hat{v}(t)$) e duas saídas no, 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 (Doya, 2000), 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 300 segundos, e o passo de tempo $\Delta t$ é de 0,05 segundos.

Para a avaliação dos resultados foi utilizada a distância da bola até o gol ao final do episódio (zero quando o robô fizer um gol). Devido a natureza estocástica do modelo proposto, cada experimento foi replicado 30 vezes, e a Figura 2 mostra a média dos resultados obtidos nos experimentos. O número médio de categorias geradas pelo INBC foi de 138,32. 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 fases distintas de treinamento e utilização.

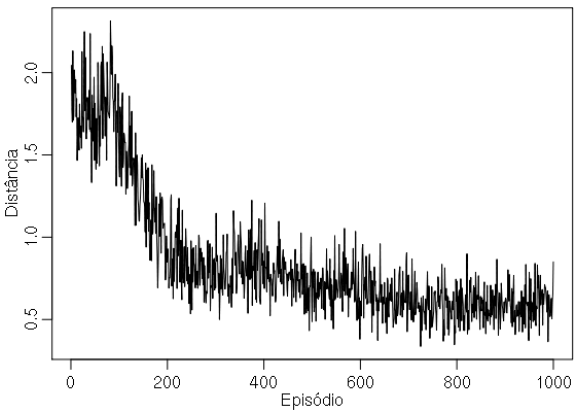


Figura 2: Distância da bola ao gol

Analisando o gráfico da Figura 2, percebe-se que nos primeiros episódios o robô não conseguiu chegar até a bola (as variações se devem a inicialização aleatória da bola). Entretanto, a partir do episódio 100 as distâncias começam a se reduzir, e a partir do episódio 750 elas estabilizam próximo de 0,6 metros, o que indica que o robô conseguiu levar a bola até o gol em boa parte dos episódios. A título de demonstração, a Figura 3 mostra a trajetória do robô durante a execução da tarefa. O

[1]ODE – http://www.ode.org
[2]Robocup – http://www.robocup.org/
[3]Pioneer 3 Operations Manual – http://www.mobilerobots.com

local da bola é mostrado no primeiro quadro com um círculo em vermelho.

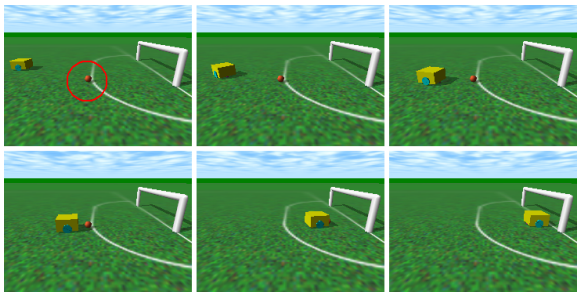

Figura 3: Exemplo de uma trajetória do robô

Com relação aos tempos de execução, o modelo proposto é bastante eficiente, conseguindo realizar cada experimento completo (todos os 1000 episódios) em aproximadamente 2,5 horas em um computador típico, o que o torna bastante adequado de ser utilizado em aplicações de controle em tempo real.

## 5 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 RBF normalizadas para aprender os valores dos estados e as ações, conseguindo inclusive alterar a estrutura das redes RBF durante o aprendizado sem a necessidade de utilizar informações a priori. Além disso, o modelo proposto consegue realizar o aprendizado utilizando apenas informações obtidas a partir de sensores do agente. 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 na maioria dos experimentos, o que demonstra a validade do modelo proposto. As perspectivas futuras incluem a utilização de um robô Pioneer 3-DX real na realização da tarefa.

## Agradecimentos

Agradecemos ao apoio do CNPq à este trabalho.

## Referências

Asada, M., Katoh, Y., Ogino, M. and Hosoda, K. (2003). A humanoid approaches to the goal - reinforcement learning based on rhythmic walking parameters, *Proc. 7th Int. RoboCup Symposium*, Vol. 3020 of *LNCS*, Springer-Verlag, Padua, Italy, pp. 344–354.

Basso, E. W. (2009). *Detecção de contexto em ambientes contínuos*, Master's thesis, UFRGS, Porto Alegre, RS, Brazil.

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

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

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

Engel, P. M. (2009). INBC: An incremental algorithm for dataflow segmantation based on a probabilistic approach, *Technical Report RP-360*, UFRGS, Porto Alegre, RS, Brazil.

Haykin, S. (2001). *Redes Neurais: Princípios e Prática*, 2 edn, Bookman, Porto Alegre, RS, Brazil.

Heinen, M. R. and Engel, P. M. (2009). Aprendizado e controle de robôs móveis autônomos utilizando atenção visual, *Anais do Simpósio de Computação Aplicada (SCA)*, Passo Fundo, RS, Brazil.

Heinen, M. R. and Osório, F. S. (2006). Applying genetic algorithms to control gait of physically based simulated robots, *Proc. IEEE Congr. Evolutionary Computation (CEC)*, Vancouver, Canada.

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

Osório, F., Musse, S., Vieira, R., Heinen, M. and Paiva, D. (2006). *Increasing Reality in Virtual Reality Applications through Physical and Behavioural Simulation*, Vol. 2 of *Research in Interactive Design*, Springer-Verlag, Berlin, Germany, pp. 1–45.

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

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

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

Sutton, R. S., Barto, A. G. and Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems, *IEEE Trans. Systems, Man and Cybernetics* **13**(5): 834–846.