Proceedings of the International Joint Conference IBERAMIA/SBIA/SBRN 2006 - 3rd Workshop on MSc dissertations and PhD thesis in Artificial Intelligence (WTDIA'2006), Ribeirão Preto, Brazil, October 23–28, 2006. CD-ROM. ISBN 85-87837-11-7

Aprendizado por Reforço Acelerado por Heurística para um Sistema Multi-Agentes

Luiz Antonio Celiberto Junior[1], Reinaldo A. C. Bianchi (orientador)

Centro Universitário da FEI – Departamento de Engenharia Elétrica
Av. Humberto de Alencar Castelo Branco, 3972 – CEP: 09850-901
São Bernardo do Campo – São Paulo – Brasil
luiz@parva.com.br, rbianchi@fei.edu.br

Resumo. O Aprendizado por Reforço é uma técnica muito conhecida para a solução de problemas quando o agente precisa atuar com sucesso em um local desconhecido por meio de tentativa e erro. Porém, esta técnica não é eficiente o bastante para ser usada em aplicações com exigências do mundo real, devido ao o tempo que o agente leva para aprender. Este artigo apresenta o uso do aprendizado por reforço, acelerado por heurística, para um sistema multi-agentes no domínio da robótica móvel. Utilizando para teste a plataforma do Robocup 2D simulação.

Palavras-chaves: Aprendizado por Reforço, Sistemas Multi-agentes, Robocup 2D.

1 Introdução

Um das características de um ser inteligente é ter a capacidade de aprender, porém o que é considerado natural para os seres humanos, para agentes robóticos é uma tarefa difícil e o sonho de muitos pesquisadores. Existem inúmeras técnicas que propõem o aprendizado para estes agentes, umas das técnicas mais conhecidas é o aprendizado por reforço.

O aprendizado por reforço (AR) é uma técnica muito atraente quando se deseja solucionar uma variedade de problemas de controle e planejamento, quando não existem modelos disponíveis *a priori*, já que seus algoritmos têm a convergência para uma situação de equilíbrio garantido, além de permitirem o aprendizado de estratégias de controles adequadas.

No aprendizado por reforço o agente aprende por meio da interação direta entre o agente e o ambiente por meio de recompensas. Estas recompensas são dadas na forma de reforços positivos e negativos, que são usadas para sinalizar ao agente se ele esta tomando as ações corretas ou incorretas. O objetivo do agente sempre será acumular o máximo reforço positivo.

Porém, somente aprender não é suficiente: o agente precisa aprender rapidamente, Bianchi [1] mostrou que a velocidade de convergência de um algoritmo de AR pode ser acelerada ao se utilizar funções heurísticas para guiar a exploração do espaço de estados-ações. Ele também propôs diversos algoritmos acelerados por heurísticas, entre eles, o Q-Learning Acelerado por Heurísticas, baseado no conhecido algoritmo Q-Learning [2, 3].

A motivação principal deste trabalho é acelerar o aprendizado por reforço através de técnicas heurísticas para sistemas multi-agentes no domínio da robótica móvel

1. Luiz Antonio Celiberto Junior é aluno no curso de mestrado em engenharia elétrica do Centro Universitário da FEI com conclusão prevista para julho/2007

usando o simulador 2D da Robocup para testes. Para tanto serão propostos métodos para criar e compartilhar as heurísticas e as recompensas entre os agentes, de maneira semelhante aos estudados por Bianchi [4] e Kok [5].

Um sistema multi-agentes (MAS) é um sistema complexo que contêm múltiplos agentes independentes, focados no gerenciamento do comportamento deste sistema [6]. MAS pode ser considerado como um sistema mais robusto se compararmos a um sistema com um único agente.

Em um sistema de um único agente, se este agente falhar, todo o sistema falha junto, porém em um sistema multi-agentes, quando se tem um agente que falha, o sistema continuará funcionando, devido a haver outros agentes trabalhando para o gerenciamento deste sistema [7].

Este artigo é organizado da seguinte maneira: na próxima seção são apresentadas, de forma sucinta, características do Robocup e no capítulo 3 será apresentado o aprendizado por reforço. A seção 4 descreve a proposta de mestrado, na seção 5 serão discutidos as experiências e os resultados obtidos e a última seção apresenta a conclusão e a proposta de trabalhos futuros.

2 Robocup

A Competição Robocup foi inicialmente proposto para ser um meio de divulgação da robótica e da pesquisa em inteligência artificial e fornecer meios para a avaliação de várias teorias, algoritmos e arquitetura, servindo também como uma ferramenta para a integração e estudos de como várias tecnologias podem trabalhar em conjunto.

Segundo Kraetzchmar [8] a Robocup deu a oportunidade dos pesquisadores trabalharem nas mais variadas áreas da inteligência artificial, como por exemplo em multi-agentes, estratégia, aprendizado, visão, redes neurais, controle e em muitas outras, pois a criação de times para o Robocup vão além da simples integração da inteligência artificial.

Atualmente o interesse pelo estudo da robótica e inteligência artificial, tem crescido em grande parte incentivados pelos campeonatos criados pelo Robocup, dando assim também um propósito educacional ao Robocup. Como o advento dos campeonatos criados do Robocup, surgiram diferentes tipos de categorias, cada um tentando focar diferentes tipos de problemas. A seguir será apresentada uma destas categorias.

2.1 Robocup 2D Simulação

A categoria o Robocup 2D simulação foi originalmente desenvolvido pelo Dr. Itsuki Nota em 1993 [9]. No Robocup 2D 11 jogadores sintéticos, criados por meio de softwares, que fazem uma partida de futebol simulado em um servidor designado por S*occerServer* e desenvolvido para ser um ambiente de simulação multi-agentes em tempo real. Sendo este um sistema incerto, com a adição de "ruídos" dependendo da distancia entre jogador-bola ou jogador-jogador.

O Robocup 2D permite a competição entre os dois times de jogadores virtuais (clientes), sendo cada jogador controlado como sendo um agente único autônomo, que irá reagir dependendo da situação na qual ele se encontra. Cada cliente envia informações ao servidor, que processa a informação recebida deste e de todos os outros clientes dos dois times. O servidor irá atualizar o ambiente e devolver a informação aos clientes através da percepção sensorial de cada agente. Esta percepção

sensorial é composta do que cada agente esta vendo, sentindo ou ouvindo.Toda comunicação entre o servidor e cliente é feito através de sockets TCP/IP.

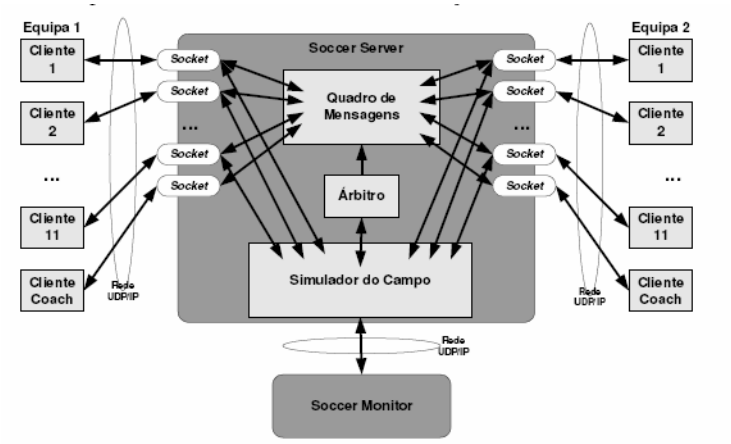

Fig. 1. Arquitetura do Sistema de Simulação SoccerServer [10]

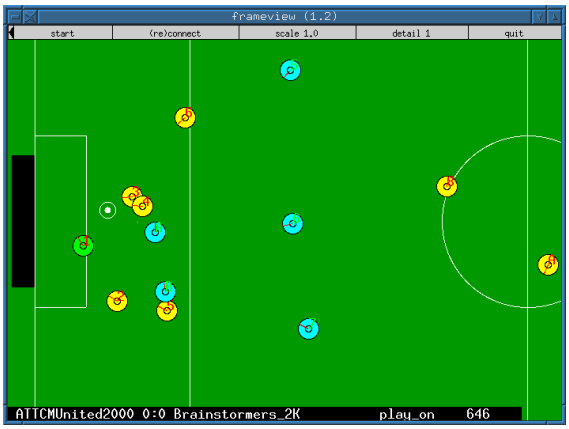

Fig. 2. Soccer Monitor

O sistema que compõem o Robocup 2D é formado por três partes principais: O servidor que irá receber todas as informações dos clientes e ira processar o jogo, os próprios clientes que fazem a comunicação com o servidor, dando as “ordens” do que tem que ser feito e o monitor do jogo, que mostra o que esta acontecendo. Na figura 1, temos uma imagem da arquitetura descrita. Toda a partida é observada através de um Soccer Monitor. Este programa é a interface gráfica, do soccer 2D, usada para observar o que está acontecendo na simulação depois de uma atualização do ambiente pelo servidor, com os dados recebidos dos clientes, como visto na figura 2.

3 Aprendizado por Reforço

No aprendizado por reforço, um agente sem conhecimentos prévios aprende através de interações com o ambiente, recebendo recompensas por suas ações e assim descobrindo a política ótima.

O aprendizado por reforço é uma técnica de aprendizado não supervisionado devido a não existência de uma representação de pares de entrada e de saída. Para cada movimentação do agente não é fornecida nenhum tipo de informação externa que ajude seu deslocamento, tirando aquela que ele mesmo percebe da sua interação com o ambiente [4].

No ambiente, em cada intervalo de tempo o agente executa uma ação (a_t). Esta ação é determinada pela política já aprendida, informando o agente para onde ele deve se locomover (s_{t+1}) e tendo em vista a recompensa ($r_{s,a}$) que irá ganhar. A recompensa pode ser dada por valores negativos ou positivos, indicando se o agente está seguindo corretamente para o objetivo ou não. A figura 1 apresenta um esboço do funcionamento do aprendizado por reforço.

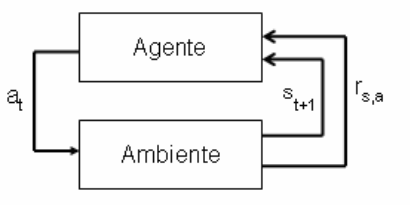

Fig 1. Aprendizado por reforço.

Dentro dos algoritmos de aprendizado por reforço, um dos mais conhecidos é o Q-Learning proposto por Watkins. No Q-Learning para cada movimento do agente é computado sua recompensa e o valor de seguir a melhor política com um desconto gama, esta política é aprendida através da interação com o ambiente e assim serão aprendidos quais os melhores movimentos para chegar a um objetivo, sendo a informação da política armazenada em uma matriz Q, que guarda os valores estimados por Q(estado, ação) [5].

O Q-Learning tem como proposta principal aprender através da interação uma política ótima π^* quando temos um desconhecido modelo de sistema. Esta política ótima é encontrada escolhendo a ação que maximiza os valores de Q, para um determinado estado. O valor de custo de um estado (função valor V(s)) também é encontrado facilmente, sendo que o custo de um estado V(s) é o valor máximo de Q(s,a) de um estado s, para todas as ações possíveis de serem executadas nesse estado.

Nesta técnica de aprendizado temos uma função de estimação Q que é calculada através da exploração on-line do agente. A função de estimação Q é computada através da equação 1:

$$\mathrm{Q(s,\ a)} = Q(s,a) + \alpha(r(s_t,a_t) + \gamma \max Q_t(s_t,a_t) - Q(s,a)) \qquad (1)$$

O parâmetro γ é compreendido entre os valores de 0 a 1. Caso o valor de γ esteja muito perto de 0 o agente tende a considerar apenas valores imediatos de recompensa,

caso os valores sejam muito perto de 1 o agente tem em vista as recompensas futuras com o maior peso.

Para a escolha das ações é usada uma estratégia conhecida como exploração aleatória ε-Greedy. O agente irá executar a ação que contenha o maior valor Q e probabilidade 1- ε e escolhe uma ação aleatória com probabilidade ε. A regra de transição de estados será dada pela seguinte equação:

$$\pi(s_t) = \begin{cases} a_{random} & \text{se q} \le \varepsilon, \\ \arg\max_{at} \hat{Q}_t(s_t, a_t) & \text{caso contrário,} \end{cases} \quad (2)$$

Sendo q um valor aleatório e com distribuição de probabilidade uniforme, ε compreendido entre os valores de 0 e 1 é o parâmetro que irá definir a taxa de exploração e de explotação e a_{random} é a ação aleatória selecionada.

O agente, através das interações com o ambiente, irá se deslocando de um estado ao outro e assim computando a função de estimação Q até conseguir chegar ao seu objetivo. Ao conseguir chegar ao seu objetivo é dito que ele completou um episódio.

A cada interação a tabela Q é atualizada. Depois de n interações é possível retirar valores condicionais desta tabela Q. Estes valores condicionais são relativos a um valor esperado da função Q seguindo uma política. Este valor condicional é chamado de função valor, que irá posteriormente influenciar na política de aprendizado e vice-versa.

Uma desvantagem do aprendizado por reforço é o tempo que o agente leva para aprender. Em muitos casos é necessário centenas de milhares de interações ou episódios para o agente poder aprender sobre onde está ou sobre o que está fazendo.

Devido a esta demora, é comum o uso de alguma forma de aceleração do aprendizado. A seção a seguir apresenta algumas técnicas que podem ser usadas para diminuir o tempo de aprendizado dos algoritmos de AR.

3.1 Aceleração do Aprendizado por Reforço

Um dos métodos de melhorar o algoritmo Q-Learning é aplicar uma heurística ou uma ajuda que acelere o aprendizado. Utilizar uma heurística em um aprendizado por reforço, por exemplo, é fornecer uma ajuda ao agente para que ele possa se locomover melhor e conseguir chegar a um objetivo mais facilmente.

Uma maneira é apresentada por Bianchi [1], que define a heurística como uma técnica que no caso médio melhora a eficiência do algoritmo. Em seu trabalho, a heurística é usada para que o agente seja influenciado a escolher as ações durante o aprendizado.

Assim esta heurística definida por $H_t(s_t,a_t)$ irá indicar a importância de executar em um estado s_t uma ação a_t. A política estará fortemente vinculada a sua função heurística. Sendo assim pode se dizer que tem se uma função heurística que é definida por uma "Política Heurística".

Devido aos algoritmos de aprendizado por reforço possuírem uma livre escolha das ações de treino, temos em uma heurística adequada uma aceleração do aprendizado e, no caso de uma heurística inadequada, temos um atraso do sistema que não o impede de convergir para um valor ótimo.

Bianchi [1] também propôs diversos algoritmos acelerados por heurísticas, entre eles, o Q-Learning Acelerado por Heurísticas (HAQL), baseado no conhecido algoritmo Q-Learning. Este algoritmo funciona de maneira similar ao Q-Learning, porém permite que sejam utilizadas heurísticas que indicam o melhor caminho para um agente aprendiz seguir, em um determinado momento. As heurísticas são integradas ao funcionamento do algoritmo modificando se a escolha das ações pelo agente. Mais detalhes podem ser encontrados em [1].

Nos últimos anos, diversas outras abordagens para aceleração do aprendizado por reforço foram desenvolvidas. Estas abordagens tentavam usar um melhor aproveitamento das experiências através de meios de generalização ou abstração. Um exemplo conhecido que usa generalização temporal é o Q(λ) [3] que estende o Q-Learning, utilizando o conceito da elegitibilidade de um par estado/ação para a propagação das atualizações.

Tem se também o aprendizado usando como técnica de aceleração a utilização de base de casos [11]. Este método faz o uso das descontinuidades da função valor aprendido e conseguindo assim definir regiões correspondentes a sub-tarefas do sistema. Ao encontrar estas sub-tarefas o aprendizado é acelerado usando informações armazenadas em uma base de casos.

4 Proposta do Mestrado

A proposta de mestrado é utilizar o aprendizado por reforço acelerado por heurísticas em um sistema multi-agentes no domínio da robótica móvel, utilizando a plataforma do Robocup 2D para fazer o aprendizado e avaliação de seu funcionamento.

O aprendizado é composto de observação e reação. Para cada local que o agente estiver, seu ângulo e da posição da bola, será considerado um estado para o agente. Cada agente informa para outros agentes onde está a bola, atualizando assim o estado constante da bola e dos estados de cada jogador. É conhecido as coordenadas de cada jogar e da bola, dentro de uma plano cartesiano, com informações de posição e ângulo.

O campo será divido em zonas, sendo estas zonas definidas dependendo das estratégias usadas, por exemplo, a zona do jogador que será o atacante sempre será o mais perto possível do gol adversário, tornando assim possível além da criação de posicionamento dos agentes um espalhamento dos jogadores, melhorando a visão global do sistema todo. Os jogadores precisarão trocar informações entre si, incluindo informação visual e sobre ações que ocorrerem no jogo.

Dentro das ações possíveis para os jogadores, forma definidas 4: correr (dash), chutar (kick), girar (turn), gritar posição da bola (say) e no goleiro foi adicionada uma quinta ação: pegar (catch). Estas ações irão determinar o que o jogador deve fazer em um determinado momento. O Robocup 2D possibilita uma variedade de forças diferentes de chutes, correr e de girar. Para as experiências iniciais foi usado apenas valores fixos para as ações, sendo estes valores futuramente expandidos.

Passes entre os jogadores são permitidos, porém não foram adicionados ao aprendizado. O passe ocorre quando um jogador com a posse da bola e estando longe do gol, percebe um adversário vindo na sua direção. Este então irá chutar a bola na direção do jogador do seu time que estiver mais perto.

A heurística para o aprendizado é obtida através das estratégias dos jogos. Kok[5] em seu trabalho definia as estratégias para cada jogador em determinadas situações e

utiliza no aprendizado dos agente. Para o trabalho aqui apresentado, foram criadas as estratégias do jogador que iram determinar o que o agente deve fazer. Estas estratégias são definidas como: goleiro, defensor e atacante. As estratégias são dadas aos jogadores dependendo da sua posição, por exemplo, se um jogador estiver em uma posição do meio do campo para frente, ele adquire a estratégia de atacante e isto irá definir que além dos movimentos já comentados ele deve também procurar o gol adversário e fazer o gol, caso o jogador possua a bola. Isto irá acelerar o aprendizado, pois ele não necessitará em saber o que é gol, ele terá a informação que deve se locomover para o local de onde deve chutar a bola para que ocorra o gol, se estiver em posição propícia para isto. Caso negativo, ele irá fazer um passe a outro jogador que também recebeu como heurística a estratégia de atacante.

Para recompensa foi fornecido um valor de + 1000 quando um jogador faz um gol e -1 para qualquer outro movimento. Para as experiências realizadas, este valor pode ser dado a apenas a um jogador ou ao time todo, tendo assim uma distribuição do aprendizado.

5 Experiências e Resultados Esperados.

Nas experiências iniciais foi utilizado o domínio desenvolvido por Littman [12], modelado a partir do futebol de robôs que propõem um jogo de Markov com soma zero entre os dois agentes. No domínio utilizado, dois jogadores chamados de A e B competem em um grade de 10x10 células. Neste jogo, cada célula pode ser ocupada por um dos jogadores, que podem executar ações de deslocamento ou ficar imóvel. A bola irá sempre acompanhar os jogadores. Quando é realizada uma ação que leva a bola a célula aonde esta um adversário, o jogador perde a bola. Quando o jogador com a bola, entre na área do gol do adversário, o time marca um gol. Quando o jogador faz uma ação que leva a bola para fora da grade, o jogador fica imóvel.

Para esta experiência foram utilizados os algoritmos Minimax-Q [12], Minimax-Q acelerado por Heurística (HMMQ) [1] e Q-Learning contra um jogador com movimentação aleatória, o resultado da média de 30 jogos pode ser observado na figura 4 e a média de saldo de gols acumulada é mostrado na tabela 1.

Para o HMMQ a heurística era definida a partir da regra: se possuir a bola, mova-se para o gol. No aprendizado foi utilizado o valore α de 1.0, taxa de exploração/explotação de 0.2 e fator de desconto γ de 0,9. O reforço utilizado foi de 1000 quando o agente chega ao gol e -1000 por gol marcado pelo adversário.

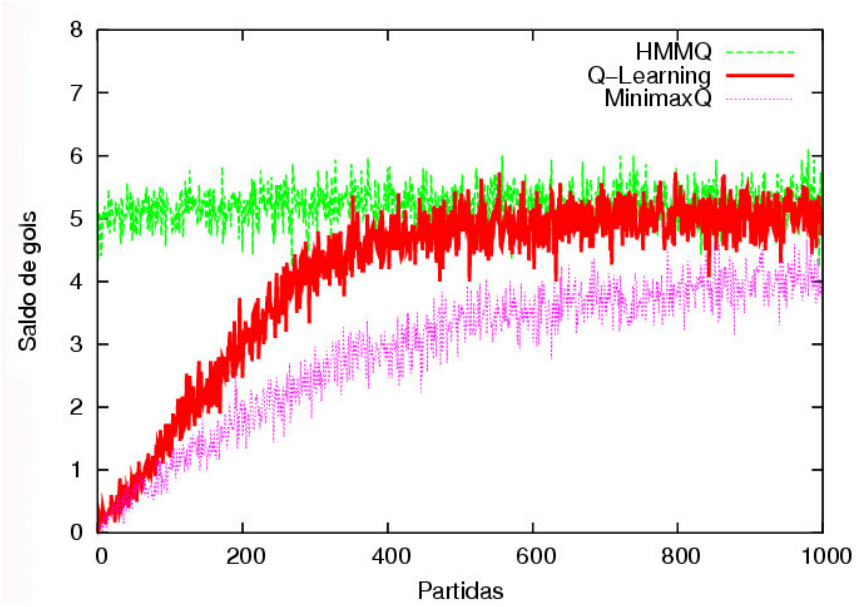

Fig.4. Resultados dos jogos.

Tabela 1.Média do saldo cumulativo de gols e do número de vitórias os jogos realizados

Partidas	**Saldo de gols cumulativo**
HMMQ x Aleatório	(5040±935) x (0.6±0.75)
MinimaxQ x Aleatório	(2784±564) x (0.63±0.65)
Q-Learning x Aleatório	(3967±748) x (0.73±0.85)
Partidas	**Vitórias**
HMMQ x Aleatório	(999±0.6) x (0.0±0.0)
MinimaxQ x Aleatório	(891±19) x (0.06±0.24)
Q-Learning x Aleatório	(937±11) x (0.06±0.24)

Os resultados demonstraram que o algoritmo com heurística, indicam ao jogador qual o melhor caminho a seguir *a priori*, acelerando assim o aprendizado em comparação a algoritmos como o MinimaxQ e o Q-Learning que levam mais partidas para aprender a melhor política.

Em outra experiência, utilizando o Robocup 2D com observação de bola e agente e com um sistema com visão global (ou seja, centrada no agente). A informação vista pelo agente era considerado um estado de aprendizado. Esta idéia a princípio foi considerada satisfatória para estudos iniciais, porém foi comprovada na prática ser ineficaz, desenvolvendo um aprendizado insatisfatório, como demonstrado experimentalmente na figura 5.

Com vista na resolução do problema, irá ser adotado um sistema de coordenadas global, assim a posição do agente e da bola independente de problemas externos, serão sempre conhecidos, gerando estados para qualquer posição dos agentes dentro do campo.

Fig.5. Aprendizado insatisfatório: Agentes aprendem a olhar a bola e não a marcar gols.

6 Conclusão e Trabalhos Futuros

Jogadores que podem compartilhar a heurística e recompensas entre si, são mais eficazes que apenas jogadores isolados com aprendizados centralizados.

Como trabalhos futuros vão ser realizadas três experiências com aprendizado, sem heurística e posteriormente com heurística, que visam tentar demonstrar qual seria o método mais eficaz de trabalhar com os agentes, de compartilhar o aprendizado e como a heurística influência no aprendizado. Estas experiências são composta de: primeiramente todos os agentes aprendem individualmente, não compartilhando com os demais o que aprendeu e a recompensa por fazer um gol. Como segunda experiência, todos os agentes compartilham o mesmo conhecimento e assim todos aprendem simultaneamente. Na terceira e última experiência, cada agente aprende de forma similar a primeira experiência, porém quando um jogador faz um gol, este passa o aprendizado do gol para os demais jogadores, concatenando sua tabela com os demais.

Existem inúmeras técnicas diferentes de aprendizado por reforço que podem fornecer resultados melhores que os comentados até o momento. Propõem se também no simulador 2D o uso do Minimax-Q ,Q(λ) e do Multi-Agent Q-Learning [13]. Um outro ponto importante que também será estudado é como melhorar a forma de troca de aprendizado e de heurísticas entre os agentes além do uso das estratégias desenvolvidas por Kok em seu trabalho adicionando-os como heurísticas para o aprendizado e comparado com seu trabalho original. Por final a tabela Q será substituída por um aproximador de funções, como uma R.N.A tipo CMAC [14], demonstrado na figura 6, com isto pretende-se diminuir o número de estados.

Todas as experiências serão comparadas com times da Robocup 2D existentes como por exemplo o UVA Trilearn e o Brainstorm [15] e seus resultados irão demonstrar como a heurística acelera o aprendizado em um sistema de mundo real.

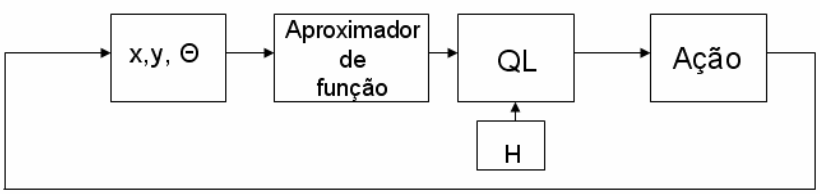

Fig 6: Aprendizado com aproximador de função.

Referências

1. Bianchi, R. A. C. Uso de Heurística para a aceleração do aprendizado por reforço. Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo, 2004.
2. Watkins, C. J. C. H. Learning from DelayedRewards. Tese (Doutorado) University of Cambridge, 1989.
3. Watkins, C. J. C. H.; Dayan, P. Q-Learning. Machine Learning, v. 8, p.279–292, 1992.
4. Bianchi, R. A. C. Uma Arquitetura de Controle Distribuída para um Sistema de Visão Computacional Propositada. Dissertacão (Mestrado) — Escola Politécnica da Universidade de São Paulo, 1998. Mestrado em Engenharia Elétrica.
5. J.R.Kok, M. T. J. Spaan, and N. Vlassis Multi-robot decision making using coordination graphs. In Proceedings of the 11th International Conference on Advanced Robotics, ICAR'03, pp. 1124–1129, Coimbra, Portugal, June 2003.
6. P. Stone and M. Veloso. Multi-Agent Systems: A Survey from a Machine Learning Perspective. Autonomous Robotics, 8(3), July 2000.
7. Kaelbling, L.P.; Littman M. L.; Moore, W.A. Reinforcement Learning: A Survey, Journal of Artificial Intelligence Research 4, May 1996, p.237-285.
8. Kraetzchmar, G. (1998) et al. "The ULM Sparrows: Research into Sensorimotor Integration, Agency, Learning, and Multiagent Cooperation". In: ROBOCUP WORKSHOP, 2, Paris. Proceedings. FIRA,. p. 459- 465.
9. Remco & Jelle, The Incremental Development of a Synthetic Multi-Agente System: The UvA Trilearn 2001 Robocup Soccer Simulation Team, 2002.
10. Luís Paulo Reis, Coordenação em Sistemas Multi-Agente: Aplicações na Gestão Universitária e Futebol Robótico, PhD Thesis/Tese de Doutoramento, Faculdade de Engenharia da Universidade do Porto, June, 2003.
11. Drummond, C.Accelerating reinforcement learning by composing solutions of automatically identified subtask, Journal of Artificial Intelligence Research, p.59-104, 2002.
12. Littman, M. L. Markov games as a framework for multi-agent reinforcement learning. In: Proceedings of the 11th International Conference on Machine Learning (ICML-94). New Brunswick, NJ: Morgan Kaufmann, 1994. p. 157–163.
13. Tan, M. Multi-agent reinforcement learning: Independent vs. Cooperative learning. In: Huhns, M. N.; Singh, M. P. (Ed.). Readings in Agents. San Francisco, CA, USA: Morgan Kaufmann, 1997. p. 487–494.
14. Carlos Henrique Costa Ribeiro.A Tutorial on Reinforcement Learning Techniques, International Joint Conference on Neural Networks (IJCNN'99), 1999, p 59,61.
15. M. Riedmiller, T. Gabel, and H. Schulz Brainstormers 2D Public Source Code Release 2005. Neuroinformatics Group Institute of Mathematics and Computer Science Institute of Cognitive Science, 2005.