

Uma Arquitetura para Sistemas Multiagentes e sua Aplicação no Futebol de Robôs

Claudio J. Biazus, Mauro Roisenberg

Universidade Federal de Santa Catarina - UFSC, Departamento de Informática e Estatística - INE, Florianópolis, SC Brasil, Caixa Postal 476- 88040-900
<http://www.inf.ufsc.br/~mauro>

Abstract. Dentre as principais dificuldades encontradas para a construção de sistemas multiagentes em que existe a disponibilidade de um sistema de visão global, como é o caso de algumas categorias de futebol de robôs, pode-se destacar: a necessidade de resposta em tempo real para identificação dos objetos em cena, conhecimento do ambiente, distribuição das competências de controle entre os comportamentos reativos a cargo de cada agente e os comportamentos deliberativo e estratégico, a cargo da entidade central. Este trabalho descreve a implementação de uma arquitetura para controle de sistemas multiagentes equipados com sistema de visão global e dotados de sensores e visão local, e sua aplicação em ambientes de futebol de robôs. A arquitetura é composta pela integração de modelos de arquitetura deliberativa e reativa. Esta arquitetura é distribuída em duas partes. A primeira parte é implementada em uma entidade central e possui os níveis: estratégico e de ação. Por outro lado, a segunda parte da arquitetura é implementada diretamente nos robôs EyeBot (embarcada), e possui os níveis: de comportamento e de execução. Assim, este novo modelo de arquitetura proposta distribui as competências de forma que tarefas relacionadas aos níveis estratégicos e de ação, tais como reconhecimento do ambiente, dos agentes que fazem parte da equipe, da equipe adversária e bola foi realizado por um sistema de processamento de imagens em uma entidade central. Por outro lado, informações de natureza reativa, tais como controlar a bola, vagar pelo ambiente, desviar de obstáculos são realizados por um sistema de processamento embarcado.

Key words: Sistemas multiagentes, Processamento de imagens, Futebol de robôs.

1 Introdução

O desenvolvimento da robótica móvel se ocupa do estudo e da construção de robôs móveis inteligentes que possam perceber o ambiente, contendo uma alta quantidade de ruídos e, ao mesmo tempo, possam atuar neste ambiente desconhecido de maneira eficiente. Para que sejam atendidos estes e outros quesitos, tais como, tempo de reação adequado e segurança, em robótica móvel, a multidisciplinaridade se fez necessária. A aplicação de recursos mecânicos, elétrico-eletrônicos

e de computação é necessária para que possam contribuir no desenvolvimento e melhorias desta área (Pazos, 2002).

Muitas dessas tecnologias têm sido desenvolvidas de maneira fortemente acoplada com o aparato sensório (Pazos, 2002). No caso de haver um dispositivo de visão global, tal como uma câmera, posicionada sobre o campo de jogo, as estratégias de implementação utilizadas remetem ao desenvolvimento de robôs móveis inteligentes com baixo grau de autonomia e que têm no “técnico”, um sistema de controle centralizado e de caráter deliberativo, localizado junto ao dispositivo de visão global, a sua principal fonte de estratégia e controle. Este técnico gera comandos que dirigem os robôs móveis inteligentes em suas tarefas. Em configurações que não contam com o sistema de visão global, todo o aparato sensório e “decisório”, e está, nos próprios robôs móveis inteligentes, que contam então com câmeras de visão local, sensores de proximidade e sensores de toque (Brady, 1992). Neste caso, as equipes normalmente não têm um “técnico”, os robôs tendem a ser totalmente autônomos. Os comportamentos e estratégias normalmente têm caráter fortemente reativo e são implementados diretamente nos robôs móveis inteligentes e, a cada momento, um comportamento pode emergir sem que haja interferência externa (Arkin, 1998) e (Botelho, 2003).

Obviamente, quando consideradas de maneira isolada, cada uma dessas estratégias apresenta vantagens e desvantagens no atendimento das restrições de tempo de resposta, capacidade de processamento dos sinais sensórios e de geração de estratégias (Shim, 2000).

Em um modelo tradicional de arquitetura de controle centralizada para futebol de robôs, todo o processamento das informações é realizado em uma entidade central. Neste modelo, os robôs móveis inteligentes atuam no ambiente de acordo com informações enviadas por este sistema. Esta arquitetura tem como vantagem a facilidade para reconhecimento do ambiente e definição de estratégias, porém, em certos casos, não oferece um poder reativo em tempo real (Schwartz, 2003).

Já em um modelo de arquitetura local ou embarcada, todo o controle, comportamento e ações do agente encontram-se no sistema, que está concentrado no próprio robô e, portanto, ele é bastante eficiente para desvio de obstáculos e condução da bola, porém, sofre com as restrições de capacidade de memória, processamento e visão incompleta do ambiente. Neste caso, a única forma de obter informações globais do ambiente é através da cooperação entre robôs (Shim, 2000).

Uma grande dificuldade no desenvolvimento deste tipo de arquitetura é a definição precisa de que competências colocar no controle central e de que competências colocar nos robôs móveis inteligentes locais de forma a minimizar o tamanho e a quantidade de mensagens de comunicação trocadas entre a entidade central e os robôs móveis inteligentes. (Biazus, 2008).

O cenário utilizado para testes, é de acordo com as regras da liga Small Size Robot League (F-180) (RoboCup, 2007). O modelo de robô SoccerBot Plus (Eye-Bot) utilizado apresenta as seguintes características: microcontrolador MC68332 Motorola de 32 bits com 25 MHz, 1 MB de memória RAM, 512 Kb de memória Flash-ROM, display de LCD com resolução de 128x64 pixels para apresentação

de gráficos com baixa resolução, portas paralelas e seriais, entradas e saídas digitais e analógicas.

Este artigo está estruturado da seguinte maneira: Após esta introdução, na seção 2 é descrita a arquitetura para sistemas multiagentes. Na seção 3 descrevemos os experimentos. A seção 4 apresenta os resultados e finalmente na seção 5 as conclusões.

2 Arquitetura para Sistemas Multiagentes

A principal razão pela qual este modelo de arquitetura pode ser aplicado ao futebol de robôs está na maneira com que combinam-se recursos de baixo desempenho computacional, sendo que é preciso obter o melhor desempenho possível para os robôs, no menor tempo possível em função da dinâmica de jogo. Dentre os recursos de hardware disponíveis, podem-se destacar: a câmera CCD com placa de aquisição de imagens com uma taxa de 29.97 fps; câmera embarcada com processamento local e um conjunto de sensores com necessidade de processamento imediato, sendo estes recursos responsáveis pelo processo de captura das informações do ambiente. Dessa forma, a integração destes recursos para capturar informações do ambiente e o imediato processamento destas informações em pontos distribuídos torna o sistema mais rápido, visto que não têm a necessidade de transmissão destas informações para uma entidade externa de processamento. Assim, evita-se a necessidade de utilizar o sistema de comunicação para enviar as informações para serem processadas e obter como resposta as mensagens de comandos para os atuadores dos robôs, tornando o sistema com um menor desempenho computacional e poder de precisão. Isto justifica a necessidade e a aplicação desta arquitetura. Portanto, uma arquitetura híbrida distribuída pode ser aplicada ao futebol de robôs para que seja possível obter um melhor desempenho, por parte dos robôs que estão atuando em um ambiente dinâmico.

Com base nos modelos de arquiteturas que foram propostos e apresentados no capítulo anterior pelos diversos pesquisadores (Shim, 2000), (Vadakkapat, 2006) e (Wang, 2003), propõe-se um modelo que integre tanto o modelo de arquitetura remota sem informação quanto o modelo de arquitetura embarcada.

Neste novo modelo, pretende-se integrar o que cada modelo de arquitetura apresenta de melhor. Isto significa dizer que se propõe um sistema distribuído onde parte das informações, tais como, reconhecimento do ambiente, dos agentes que fazem parte da equipe, da equipe adversária e bola é realizado por um sistema de processamento de imagens em uma entidade central. Por outro lado, informações de natureza reativa, tais como, controlar a bola, vagar pelo ambiente, desviar de obstáculos e decisão de chute são realizados por um sistema de processamento embarcado.

O sistema embarcado recebe informações através do conjunto de sensores e da câmera, processa as informações capturadas no próprio sistema embarcado do robô e imediatamente envia comandos para seus atuadores. A vantagem em construir um modelo de arquitetura distribuída consiste no fato de que os recur-

sos disponíveis são mais bem aproveitados. Pois o uso de forma independente, tanto da arquitetura embarcada quanto da arquitetura centralizada, não oferece a melhor solução para o problema. Dessa forma, é necessário que haja cooperação e negociação entre os sistemas, deliberativos e reativos. Assim, agentes individuais podem ignorar comandos vindos do sistema deliberativo quando o sistema reativo oferece uma melhor solução para o problema, por exemplo, quando o sistema de visão local indicar a posição em que a bola se encontra. Por outro lado, os agentes também podem ignorar comandos do sistema reativo quando o sistema de visão local não indicar a presença da bola e o sistema deliberativo indicar que o sentido de deslocamento do agente deve ser um sentido oposto ao deslocamento atual do robô. Diante de tudo isso, o novo modelo procura atender as reais necessidades dos objetivos, que é construir um modelo de arquitetura híbrida para sistemas multiagentes que possa ser aplicado ao futebol de robôs. A Figura 1 apresenta o modelo da arquitetura para sistemas multiagentes.

2.1 Módulo da entidade central

O módulo da entidade central é composto por uma câmera e os níveis estratégico e de ação. O nível estratégico é composto por quatro módulos: sistema de exploração do ambiente, sistema de definição de estratégia, sistema de definição de trajetória e sistema de controle de navegação. Já o nível de ação corresponde à ação, que é atribuída aos robôs de forma independente. No módulo da entidade, é considerada a importância hierárquica da mesma forma como está sendo aplicado no módulo do agente. No entanto, é necessário que haja informações suficientes nos sistemas de exploração do ambiente, estratégico, de trajetória e de navegação para que possa ser definida qual deve ser a ação que é executada pelo agente.

Como se pode observar, a arquitetura apresentada está dividida em duas partes (na divisão não é considerado o módulo de comunicação). A primeira parte da arquitetura é composta por um módulo, que é implementado em uma entidade central. Este módulo possui os níveis: estratégico e de ação. Por outro lado, a segunda parte da arquitetura é composta pelo módulo do agente que é implementado nos robôs (embarcado). Este módulo contém os níveis de comportamento e de execução. Além disso, tem-se um módulo de comunicação, que é responsável por ligar as duas partes da arquitetura.

2.2 Módulo de comunicação

Para que haja troca de informações entre a entidade central e os robôs, faz-se necessário um módulo de comunicação. O conjunto de robôs possui uma unidade de comunicação sem fio para realizar a troca de mensagens. Desta forma, permite estabelecer a comunicação em uma determinada frequência estabelecida, tanto para a recepção das mensagens de comandos por parte dos robôs quanto para transmissão de informações entre entidade central e robôs.

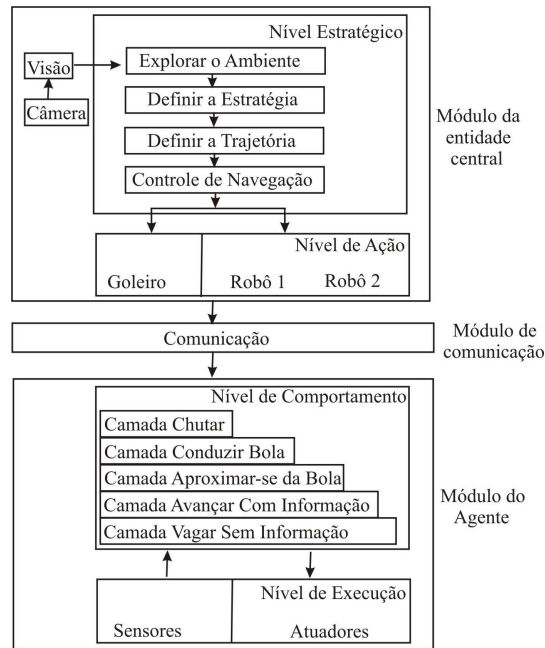


Fig. 1. Arquitetura para sistemas multiagentes.

O próprio sistema operacional do robô é quem gerencia este meio de comunicação através de uma rede virtual do tipo token ring. Este sistema de comunicação transmite a uma taxa de 9.600 bps; também apresenta a característica de tolerância a falhas. Esta rede virtual, denominada de EyeNet, permite conectar diversos robôs com a mesma estação de comunicação. A comunicação é realizada através do envio de pacotes em forma de comandos da entidade central para os robôs (Eyebot, 2007).

2.3 Módulo do agente

Ao definir a arquitetura para os agentes locais, optou-se pela implementação de uma arquitetura puramente reativa, baseada na Arquitetura de Subsumption, proposta por (Brooks, 1985). Este modelo reativo apresenta muitas propriedades que são consideradas adequadas. Dentre as que se pode citar está a capacidade de execução de forma rápida e eliminar a necessidade de planejamento.

Os comportamentos foram montados conforme a necessidade de fornecer informações para as camadas superiores. A arquitetura proposta contém cinco camadas que são: vagar sem informação, avançar com informação, aproximar-se da bola, conduzir bola e chutar, conforme se pode observar na Figura 1. As camadas que compõem a arquitetura do robô representam uma estrutura hierárquica, sendo que a camada que se encontra em mais alto nível assume a

atividade que foi designada à camada inferior, quando uma determinada configuração dos sensores indicar uma situação que seja considerada propícia para sua atuação.

3 Implementação

Dentre os recursos utilizados para a implementação desta arquitetura, tem-se: um microcomputador com sistema operacional Windows XP, onde foi implantado o módulo da entidade central que é responsável pelo controle do sistema de visão, pelo nível estratégico, nível de ação e pelo sistema de comunicação; o sistema de visão composto por uma câmera de vídeo colorida JVC com saída S-VHS. As imagens capturadas são digitalizadas através de uma placa de captura PCI da PixelView PV-TV304P. A imagem digitalizada apresenta uma resolução de 640 x 480 pixels em formato RGB, com taxa de captura de 29.97 fps. Três robôs SoccerBot (EyeBot) contendo sistema de visão local, sensores e atuadores. A seguir, são apresentadas as estratégias utilizadas bem como os resultados que foram obtidos no desenvolvimento desta aplicação, utilizando as ferramentas de desenvolvimento Dev-C++ e a biblioteca de visualização OpenCV.

3.1 Implementação do módulo da entidade central

O sistema que corresponde ao módulo da entidade central é responsável pelo controle do sistema de visão e dos níveis estratégico e de ação. Cabe ao sistema alocado na entidade central explorar, planejar, definir e deliberar a estratégia que será executada pelos robôs. A definição de qual deve ser a estratégia que o robô deve executar é dada a partir do momento em que o sistema de exploração do ambiente, juntamente com o sistema de visão global, indicar a posição em que a bola e os robôs se encontram no ambiente. É a partir da informação que corresponde à coordenada da posição da bola, que o sistema de definição de estratégia, define uma trajetória que será executada pelo robô.

3.2 Implementação do sistema de visão

O sistema de visão utilizado neste módulo é um sistema de captura global, que recebe um conjunto de pixels que representam uma imagem capturada do campo, com os robôs e a bola, que estão em movimento em um dado instante. Este processo de captura é realizado por uma câmera que está posicionada a uma distância de 4m acima do campo, tendo este sistema de visão a função de capturar a posição dos robôs e da bola (Robocup, 2007).

3.3 Implementação do sistema de exploração do ambiente

O processo de exploração do ambiente consiste em localizar cada objeto que está inserido no ambiente. Para localizar cada objeto o sistema desenvolvido recebe um quadro em formato RGB, converte esta imagem para o formato HLS,

separa esta imagem em três canais, CanalH, CanalL, CanalS, onde o CanalH representa o tom da imagem. O CanalL representa o nível de luminosidade e o CanalS representa o nível de saturação. Após a divisão da imagem em três canais, é possível aplicar um limiar de corte, que é responsável pelo intervalo aceito no processo de classificação de um determinado objeto. Dependendo da posição em que o objeto se encontra, o nível de luminosidade que incide sobre este objeto é variável. Sendo assim, o sistema aceita como limiar de corte uma faixa de valores que corresponde ao intervalo aceito no processo de classificação do objeto. Após ter sido selecionada, a cor que corresponde a um objeto determinado, o algoritmo realiza o processo de erosão e dilatação dos objetos. O processo de erosão ao redor dos objetos é necessário para que seja possível remover pontos isolados que representam falsos objetos, já a dilatação serve para uma melhor visualização do objeto na imagem.

3.4 Implementação do sistema de definição de estratégia

A implementação do módulo de estratégia consiste na definição de qual deve ser o comportamento que é executado pelo robô. Para a aplicação em questão, e considerando que a equipe possui três robôs EyeBot, adotaram-se dois comportamentos: um para o goleiro e outro para os jogadores de linha. Para os jogadores de linha, tanto eles podem assumir o comportamento de defesa quanto de ataque.

A definição da estratégia a ser adotada utiliza como base as informações fornecidas pelo sistema de visão global, que apresenta o ambiente, e pelo sistema de exploração do ambiente, que apresenta o estado de cada agente (robôs e bola) no ambiente. Para definir uma estratégia é necessário identificar todos os objetos inseridos no ambiente, e transformar a informação que corresponde à posição do objeto em uma nova informação que corresponde ao vetor de distância que está associado ao estado de cada agente. Dessa forma, um conjunto de estados e transições (ações) é definido.

3.5 Implementação do sistema de definição de trajetória

A implementação do módulo de definição de trajetória depende do comportamento que é assumido pelo robô. Assim, definiu-se três comportamentos, do goleiro, de defesa e de ataque. É necessário conhecer o conjunto de ações válidas para o comportamento. Dessa forma, é preciso saber a posição alvo que corresponde ao ponto para onde o robô deve se deslocar e o ponto onde o robô se encontra.

Para este algoritmo, e sendo que as informações necessárias para o deslocamento do robô são coletadas pelo conjunto de sensores e processadas pela camada vagar sem informação no próprio EyeBot, utilizou-se então um método que corresponde apenas à definição das ações válidas, isto é, qual deve ser a ação necessária para que seja possível o robô atingir o ponto alvo. Foi implementado, dessa forma, dada à característica da arquitetura que consiste em deixar os níveis estratégico e de atribuição de comportamento (ação), na entidade central, e comportamentos reativos no próprio robô.

3.6 Implementação do sistema de navegação

A implementação do sistema de navegação é responsável pelo controle das mensagens de comando que correspondem ao vetor de velocidade que é enviado para os atuadores do EyeBot. Este sistema recebe o vetor de velocidade que representa à velocidade linear e angular do sistema de definição de trajetória, processa estas informações, e verifica se realmente é necessário enviar esta mensagem para o robô. O processo de verificação consiste em saber se realmente a informação coletada pelo sistema visão global e processada pela entidade central é importante para o deslocamento do robô. Todas as mensagens transmitidas da entidade central para o robô são enviadas diretamente para a camada avançar com informação. Estas mensagens possuem ordem de prioridade maior e sendo assim sempre são executadas pela camada avançar com informação. Esta prioridade é atribuída em função de que o sistema de visão global possui a característica de poder visualizar todo o ambiente do jogo; já a camada avançar com informação é responsável pela precisão no controle da navegação do robô.

3.7 Implementação do módulo de comunicação

A implementação do módulo de comunicação é responsável pela troca de mensagens entre a entidade central e os robôs. O rádio do EyeBot verifica se existe mensagem disponível e identifica se esta mensagem pertence ao robô. Quando existem mais do que uma mensagem disponível para o mesmo robô, somente a última mensagem enviada é executada; o restante é descartado. Para o caso em que não existem mensagens disponíveis, o sistema de comunicação do robô fica aguardando uma nova mensagem. Após o recebimento de uma mensagem de comando, o sistema de comunicação do robô envia para a entidade central a confirmação do recebimento dessa mensagem.

3.8 Implementação do módulo do EyeBot

O módulo do EyeBot é composto por uma câmera CCD, três sensores e cinco comportamentos separados em forma de camadas, além dos atuadores. A implementação do módulo de inicialização do EyeBot consiste no processo de checagem dos componentes de hardware do robô e da integração de todas as camadas que fazem parte do software embarcado. A checagem dos componentes de hardware do robô é responsável por indicar se existe algum problema com relação ao rádio, à câmera, aos motores e ao mecanismo de chute. É este módulo que informa caso um desses componentes não esteja conectado ou não funcionando. Já o processo de integração das camadas, une todas as camadas que fazem parte do sistema embarcado, bem como da liberação de memória após a execução de cada camada.

3.9 Implementação do sistema de visão

A implementação do sistema de visão para o robô SoccerBot (EyeBot) foi desenvolvida com o objetivo de que o robô reconheça apenas três cores: laranja, azul

e amarelo. A cor laranja corresponde à cor da bola, e as cores azul e amarelo representam as cores das metas. A forma utilizada na implementação consiste em procurar pelo tom da cor correspondente e armazenar este valor em uma variável. Após a análise de todo o quadro da imagem, o algoritmo traça um histograma para o eixo horizontal e outro, para o eixo vertical e combina os histogramas. Dessa forma, é possível obter um valor aproximado do centro da imagem que corresponde à posição aproximada do objetivo. O valor obtido da variável que armazena o ponto da coordenada correspondente ao centro da imagem, adequa-se ao valor da distância em que a bola se encontra do robô, tanto para o eixo horizontal quanto para o eixo vertical.

3.10 Implementação da camada vagar sem informação

O sistema implementado para a camada vagar sem informação foi desenvolvido com o objetivo de que o robô possa vagar pelo ambiente a partir das informações adquiridas do seu conjunto de sensores. Sua aplicação se faz necessária para que o robô possa vagar pelo ambiente independente de possuir a informação da posição em que a bola se encontra e ao mesmo tempo de forma reativa. O funcionamento da camada vagar sem informação verifica a distância em que um obstáculo se encontra do robô. Cada robô é equipado com um conjunto de três sensores, e a distância entre o robô e os obstáculos é fornecida através da leitura destes sensores. O sistema desenvolvido verifica se os sensores: frontal, à direita e à esquerda possuem a informação de presença ou ausência de obstáculo. Com a informação coletada pelo sensor frontal, a máquina de estados define qual ação deve ser executada.

3.11 Implementação da camada avançar com informação

O sistema implementado para a camada avançar com informação foi desenvolvido com o objetivo de que esta camada somente seja acionada a partir do momento em que o robô possui a informação da posição em que a bola se encontra. O funcionamento da camada avançar com informação é iniciado no instante em que o sistema de visão embarcado indicar a posição da bola. Dessa forma, o algoritmo desenvolvido executa uma ação, que é responsável por calcular a trajetória de direção para o robô. O cálculo da trajetória de direção toma como referência o centro da posição da bola. A partir deste ponto de referência, o algoritmo desenvolvido verifica esta posição para saber se este ponto se encontra: no centro da imagem; ou à esquerda; ou à direita. Caso o ponto de referência pertença ao intervalo considerado como sendo o centro da imagem, a mensagem de comando enviada aos atuadores consiste simplesmente em avançar. Já para o caso onde a posição do ponto de referência não pertence ao intervalo considerado como sendo o centro da imagem, independente deste ponto estar à esquerda ou à direita, é necessário calcular a nova trajetória de direção para o robô.

3.12 Implementação da camada aproximar-se da bola

O sistema implementado para a camada aproximar-se da bola foi projetado com o objetivo de atender à deficiência que o robô tem no momento em que passa do estado em que não está de posse da bola para o estado em que está de posse da bola. A abordagem do robô em relação à bola deve ser o mais suave possível, pois uma abordagem em alta velocidade pode resultar em uma colisão entre o robô e a bola, fazendo com que a bola seja lançada para uma posição mais distante do robô. No entanto, seu comportamento somente é acionado a partir do momento em que o robô está se deslocando em direção à bola. Este comportamento consiste em verificar a distância em que a bola se encontra do robô, e a partir do instante em que o robô vai se aproximando da bola, e encontra-se em uma distância considerada como sendo um intervalo aceito para aplicar o comportamento de redução de velocidade. Um intervalo aceito para aplicar o processo de redução consiste em um valor entre 65mm à 50mm. Dentro deste intervalo definido, é possível aplicar o processo de redução de velocidade que consiste na mudança do valor linear de 0.4 para 0.2. A partir do momento em que a abordagem aconteceu, a velocidade do robô retorna às condições normais de navegação.

3.13 Implementação da camada conduzir bola

O sistema implementado para a camada conduzir bola é responsável pelo controle que o robô exerce na condução da bola. Esta camada controla o robô nos momentos em que a bola está sob o seu domínio. O controle exercido por este algoritmo é aplicado no instante em que a bola está escapando do domínio do robô. Sendo assim, o algoritmo verifica em qual das extremidades do mecanismo de chute a bola se encontra e ajusta a nova trajetória do robô, para que a bola possa ficar no centro do mecanismo de chute. Assim, é possível que o robô possa conduzir a bola até o seu objetivo, que é a meta da equipe adversária.

3.14 Implementação da camada chutar

O sistema implementado para a camada chutar é responsável pelo movimento do mecanismo de chute. Dentre os movimentos executados pelo sistema que controla o movimento do mecanismo de chute está: levantar, abaixar e levantar de forma moderada o mecanismo de chute.

O mecanismo de chute somente é acionado em duas condições, quando o robô está em condições de chutar a bola na meta da equipe adversária ou quando pretende passar a bola para um robô da mesma equipe. Para o caso onde o robô chuta a bola em direção à meta da equipe adversária, o robô deve estar em condições de chute. Estas condições são: visão da meta da equipe adversária e posição do robô considerada como sendo próxima da meta da equipe adversária. Já para o caso onde o robô passa a bola para um outro robô da mesma equipe, esta condição depende apenas da dinâmica do jogo.

A decisão que envolve o acionamento do mecanismo de chute é atribuída tanto ao robô quanto à entidade central. Para o caso onde a decisão parte da entidade central, é necessário enviar uma mensagem através do sistema de comunicação, que corresponde ao comando responsável pelo acionamento do mecanismo de chute. Este comando é interpretado pela camada chutar e executado pelo agente, que é responsável pelo controle do mecanismo de chute.

4 Resultados

Todos os resultados descritos nesta seção foram obtidos através da observação feita no display de LCD dos robôs EyeBots. Para se chegar a cada resultado apresentado, foi necessário configurar, no sistema desenvolvido dos robôs, comandos de impressão das mensagens de execução. Assim, foi possível observar o desempenho dos robôs no ambiente e realizar estes ensaios. Os resultados obtidos com o desenvolvimento dessa aplicação demonstraram um desempenho satisfatório, pois o sistema desenvolvido que está hospedado na entidade central é consistente em relação à localização da bola e dos robôs. Este sistema processa as imagens em tempo real e não apresenta um melhor desempenho em função do sistema de captura utilizado, que opera a uma taxa de 29.97 fps.

Em relação à definição de deslocamento envolvendo os níveis, estratégico e de comportamento, percebeu-se que em função da dinâmica de jogo, o resultado obtido para o deslocamento do robô em relação a sua direção, em geral, converge para o mesmo ponto. Dessa forma, a mensagem de comando enviada pelo nível estratégico é simplesmente descartada pelo robô. Por outro lado, quando os níveis estratégicos e de comportamento não convergem para um ponto em comum, e sendo que o robô não possui a informação que correspondem à posição da bola, ele descarta as mensagens enviadas pelo nível de comportamento do sistema embarcado e executa as mensagens enviadas pelo nível estratégico que correspondem à informação enviada pela entidade central. Assim, este modelo de arquitetura atende aos objetivos que consistem na construção de um modelo de arquitetura híbrida e distribuída, atuando em tempo real.

Com relação aos resultados obtidos com o desenvolvimento do sistema embarcado, constatou-se que realmente os robôs atuam no ambiente conforme foi proposto. A implementação das camadas com seus estados e ações são considerados suficientes para que o robô vague pelo ambiente sem colidir com obstáculos e, simultaneamente, ao encontro da bola, tendo como objetivo capturar e conduzir a bola até à meta da equipe adversária.

5 Conclusões

Neste trabalho, descreveu-se e analisou-se o desenvolvimento de uma arquitetura híbrida para sistemas multiagentes e sua aplicação no futebol de robôs. Esta arquitetura é adaptada com um sistema de visão global e dotada de sensores e visão local para captura das informações. O processamento das imagens é realizado tanto nos robôs quanto na entidade central.

A arquitetura híbrida distribuída para processamento das informações permitiu tratar as informações em locais estratégicos e possibilitou um ganho com relação ao desempenho do sistema. Assim, tanto o sistema de visão global quanto o sistema de visão local foram beneficiados, pois esta arquitetura permitiu os níveis estratégicos e de ação serem precisos e, ao mesmo tempo, os níveis de comportamento e execução se beneficiarem com relação ao domínio do ambiente.

Esta arquitetura foi testada na V Competição Brasileira de Robótica (CBR-2007). Apesar dos resultados obtidos com relação à classificação na competição não terem sido dos melhores, a participação serviu de experiência para testar a implementação da arquitetura proposta em um ambiente oficial da RoboCup. As falhas ocorridas durante a competição são atribuídas ao baixo desempenho oferecido pelo robô EyeBot, principalmente, com relação ao desgaste das baterias e à arquitetura oferecida pelo hardware, pois a arquitetura do hardware não permite retirar o atrito produzido pelo ponto de apoio do robô. Assim, este modelo de robô apresenta um desempenho considerado fraco para uma competição oficial da RoboCup. Por outro lado, o software desenvolvido passou por uma série de melhorias, tendo como objetivo ajustar pontos considerados relevantes.

References

1. Arkin, R. C.: Behavior-Based Robotics. MIT Press (1998)
2. Biazus, C. J., Roisenberg, M.: The Development of a Hybrid, Distributed Architecture for Multiagent Systems and its Application in Robot Soccer. IEEE World Congress on Computational Intelligence. Hong Kong (2008).
3. Botelho, S., et al.: FURGBOL Construindo Robôs Autônomos Holonômicos para Jogar Futebol. VI Simpósio Brasileiro de Automação Inteligente. Bauru, Setembro (2003)
4. Brady, J. M., Wang, H.: Vision for mobile robots, Proceeding of Royal Society, London (1992)
5. EyeBot.: Online Documentation. Disponível por www em. <http://robotics.ee.uwa.edu.au/eyebot/> (2007)
6. Menezes, P. B.: Linguagens Formais e Autômatos. 5 ed. Editora Sagra Luzzatto, Porto Alegre (2005)
7. Pazos, F.: Automação de Sistemas e Robótica. Editora Axcel (2002)
8. RoboCup.: Disponível por www em. <http://www.robocup.org>
9. Shim, H. S. et al.: A Hybrid Control Structure for Vision Based Soccer Robot System. International J. Intelligent Automation and Soft Computing, pp. 89–101 (2000)
10. Vadakkepat, P. et al.: Evolution of fuzzy behaviors for multi-robotic system. Robotics and Autonomous Systems. Singapore, July (2006)
11. Wang, Z. D. et al.: LOGUE: an architecture for task and behavior object transmission among multiple autonomous robots. Robotics and Autonomous Systems 44, pp. 261–271 (2003)
12. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195–197 (1981)