CORDENONSI, A. Z. ; DHEIN, Guilherme ; LOY, Alisson Moscato ; CESCA, Leone. *Framework para o Desenvolvimento de Agentes para a Robocup: o Módulo de Definição de Áreas de Atuação*. In: IV Congresso Brasileiro de Computação, 2004, Itajaí. Anais do IV Congresso Brasileiro de Computação. Itajaí : Editora Tec Art, 2004.

Framework para o Desenvolvimento de Agentes para a Robocup: o Módulo de Definição de Áreas de Atuação

Andre Zanki Cordenonsi, Guilherme Dhein, Alisson Moscato Loy, Leone Cesca

***Resumo*—Este trabalho apresenta o Módulo de Definição de Áreas de Atuação, um dos componentes de um *framework* para o desenvolvimento de agentes para a Robocup. Este módulo possui uma interface visual, que apresenta um campo de jogo sobre o qual o desenvolvedor demarca a área na qual o agente deve limitar sua atuação. O resultado é apresentado através de parâmetros que relacionam a área selecionada com os marcadores visíveis pelo jogador durante a partida. A partir da observação destes marcadores, o agente é capaz de calcular sua posição em relação a sua área. Os parâmetros do módulo configuram um agente genérico implementado em C++, que pode ser utilizado na liga de simulação da Robocup.**

***Palavras Chave*—Inteligência Artificial; Sistemas Multiagentes; Agentes Reativos; Robocup; Soccerserver; Área de atuação; Posicionamento do agente; Liga de simulação.**

I. INTRODUÇÃO

O presente trabalho tem por objetivo apresentar o módulo de definição de áreas de atuação do *framework* para o desenvolvimento de agentes que possam atuar como jogadores de futebol na liga de simulação da *Robocup*. Os agentes devem atuar em uma partida simulada de futebol de forma totalmente automatizada, sem qualquer intervenção humana e em tempo real.

Para a definição da percepção, construção de estratégias e tomada de decisão de cada agente, é necessário o desenvolvimento individual de cada agente que irá atuar no ambiente, definindo suas características básicas e seus comportamentos de acordo com a posição em que cada agente irá atuar dentro do jogo proposto. O simulador *SoccerServer* proporciona um ambiente em tempo real que possibilita analisar o desempenho de diferentes Sistemas Multiagentes.

Um time de futebol de robôs consiste de onze agentes jogadores, sendo que cada agente possui uma visão limitada, liberdade total de movimentação e uma certa quota de energia, que é gasta a cada movimento do jogador. No entanto, como os agentes deverão atuar em funções diferentes (goleiro, atacante, defensor), é necessário definir uma área de atuação para este agente impedindo, deste modo, que haja uma aglomeração de jogadores em poucas regiões do ambiente de simulação. É importante salientar que os agentes podem conversar entre si, mas que a forma de comunicação disponível pelo simulador é bastante precária e não confiável. Estas características estão definidas pelo simulador que implementa a não confiabilidade como forma de tornar o ambiente o mais próximo da realidade.

Desta forma, se torna interessante a definição de áreas fixas de atuação para cada agente, assim como no futebol real, de acordo com o objetivo programado para o agente específico. A ferramenta implementada permite ao desenvolvedor definir graficamente as áreas de atuação dos diversos agentes, sendo que este módulo é independente do comportamento definido para cada agente. Ou seja, é possível modificar a área de atuação de cada agente sem alterar o código responsável pelas demais características do comportamento do mesmo.

II. ROBOCUP

O Futebol de Robôs é uma atividade científica reconhecida pelo IEEE-*Institute of Electrical and Electronics Engineers,* que tem apoiado a realização das "Copas Mundiais" na forma de congressos científicos. Assim, os participantes exibem o desempenho de suas equipes e trocam experiências**.**

A *The Robot World Cup Initiative* (*RoboCup*) é uma motivação na área de inteligência artificial e de sistemas multiagentes (SMA), pois estimula a resolução de um problema (jogar futebol) no qual uma ampla variedade de tecnologias pode ser integrada, examinada e comparada.

O projeto *Robocup* tem como objetivos o incentivo e a popularização dos Sistemas Multiagentes como uma metodologia de resolução computacional de problemas. Por apresentar um problema padrão (times de futebol compostos unicamente por agentes autônomos) e por não restringir formas de solução, esta iniciativa permite que uma vasta gama de tecnologias sejam integradas, examinadas e comparadas.

O *The RoboCup Federation* é um órgão internacional formado por um grupo de pesquisadores que implementa e distribui as atualizações do simulador, organiza as competições e normatiza as regras para as disputas [1]. A *RoboCup* possui diversas categorias de disputa: Robôs de Pequeno Porte, Robôs de Médio Porte, Robôs de Quatro Pernas e a Liga de Simulação. Para esta última é utilizado um sistema que simula um campo de futebol (*SoccerServer*), onde os times são compostos unicamente por agentes implementados via *software*, não envolvendo a construção física de um robô. O *SoccerServer*, apresentado na Fig. 1, é um ambiente de simulação dinâmico e em tempo-real, onde os

agentes são clientes do servidor e se comunicam com este através de *sockets* UDP.

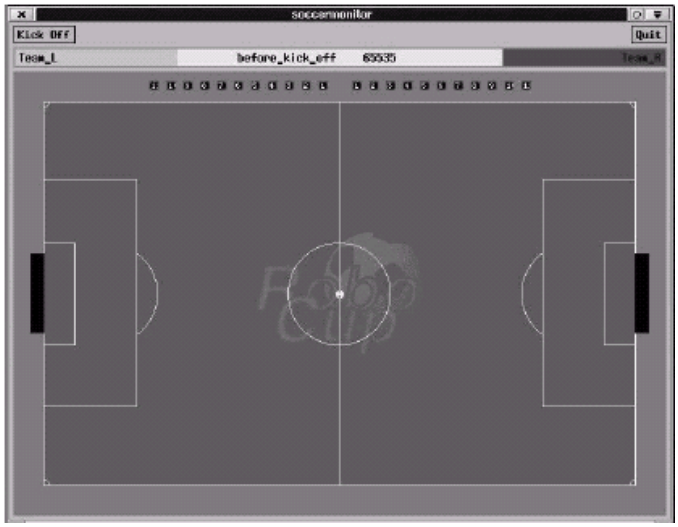


Fig. 1. Simulador *SoccerServer*

III. Agentes e sistemas multiagentes

Dentre as diversas metodologias e tecnologias disponíveis para a construção de sistemas inteligentes para a *Robocup*, se destaca a utilização de sistemas multiagentes. Um agente pode ser definido como uma entidade inteligente que apresenta, entre outras, as seguintes características: estar inserido em um ambiente e ter capacidade para perceber e atuar sobre este meio; possuir autonomia; se adaptar a novas situações; ser capaz de perceber alterações no ambiente e atuar em função destas alterações; exibir comportamento direcionado a solução dos seus objetivos; e exibir habilidade social. Além destas, as implementações de agentes em Inteligência Artificial podem incorporar outras características, como aprendizagem.

Os agentes devem raciocinar a respeito das ações e sobre o processo de coordenação em si. As suas arquiteturas são mais flexíveis e a organização do sistema está sujeita a mudanças, visando adaptar -se às variações no ambiente e/ou no problema a ser resolvido. Para [2], as arquiteturas são especificações de como podemos construir agentes a partir de módulos e como estes módulos interagem. O conjunto total de módulos e suas interações devem especificar como, a partir dos dados dos sensores e do estado interno atual do agente, são obtidos ações e o novo estado interno. Assim, cabe ao construtor da arquitetura definir a forma de dividir o problema e distribuir capacidades aos módulos do agente, além da melhor forma de construir estes módulos e de como será a sua interação.

Sistemas baseados nestas arquiteturas trabalham, geralmente, com diversos agentes que atuam concomitantemente em busca de uma solução para o problema proposto. A implementação de uma coletividade de agentes acarreta num aumento da complexidade do sistema como um todo, pois se torna necessário a elaboração de diversos mecanismos, tais como: coordenação, controle, organização e planejamento conjunto. Um conjunto de agentes trabalhando em busca de um mesmo objetivo comum é denominado Sistema Multiagente. A metáfora de inteligência utilizada pelos Sistemas Multiagentes é a *comunidade inteligente*, ou seja, o comportamento social que é base para a inteligência do sistema.

A construção de um Sistema Multiagente envolve a definição da arquitetura do agente como indivíduo e da arquitetura do sistema (ambiente e comportamento social), bem como a sua implementação. Diversos Sistemas Multiagentes têm sido propostos para a resolução de problemas: mineração de dados, controle de sistemas industriais, de tráfego aéreo, de redes de distribuição de energia elétrica, entre outros.

IV. *Framework* para o Desenvolvimento de Agentes para a *Robocup*

O desenvolvimento de agentes e de sistemas multiagentes para a *Robocup* é uma tarefa bastante complexa. O simulador *SoccerServer* foi desenvolvido com a intenção de ser um verdadeiro desafio aos seus competidores, trazendo toda a complexidade de um ambiente real para a simulação, incentivando, deste modo, a utilização de soluções criativas e inovadoras para o desenvolvimento dos times de futebol.

Além disso, é notório que qualquer sistema multiagente em desenvolvimento deve, necessariamente, passar por uma grande fase de testes, na qual as ações e os inter-relacionamentos entre os agentes são postos a prova, de tal modo a comprovar na prática se as regras comportamentais de cada agente foram definidas adequadamente.

Baseados nas asserções acima, foi definido um *framework* para o desenvolvimento de agentes para a *Robocup* com o intuito de facilitar a construção de agentes que possam atuar de forma satisfatória no ambiente proposto. A arquitetura do *framework* proposto pode ser visualizada na Fig. 2.

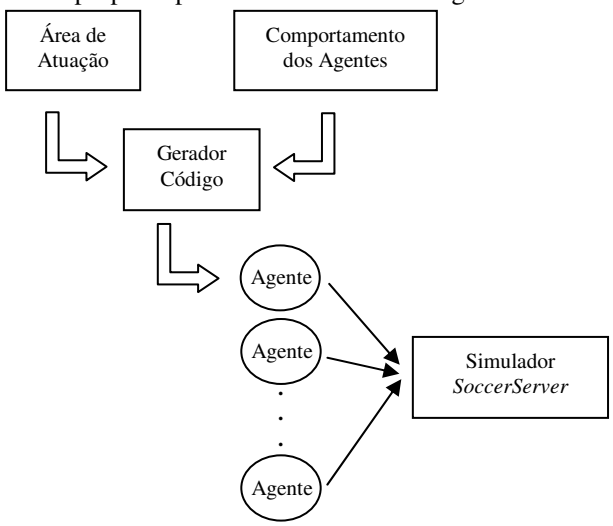


Fig..2. *Framework* para o desenvolvimento de agentes para a *Robocup*

O primeiro módulo fornece uma ferramenta para a definição da área de atuação de cada agente no ambiente de

simulação. Esta ferramenta, apresentada no próximo capítulo, permite que os agentes possam se posicionar no campo, verificando sempre se estão dentro da sua área de atuação e, caso contrário, que retornem para o seu lugar.

O segundo módulo é composto por uma ferramenta visual que implementa o comportamento individual de cada agente. Para tanto, foram implementadas diversas ações para que os agentes atuem no simulador *SoccerServer*. Estas ações podem ser separadas em dois grupos: o primeiro é composto por comportamentos individuais como correr ao encontro da bola, chutá-la, detectar o gol e correr em sua direção, chutar em gol, entre outros; o segundo é composto por capacidades de atuação em equipe, como a capacidade de realizar passes de bola, não disputar a bola com um companheiro de equipe, detectar se há companheiro mais próximo da bola, entre outros.

A arquitetura escolhida para o desenvolvimento dos agentes foi baseada na proposta de [3], que define níveis de atuação com prioridades. A integração das regras com os níveis de prioridade foi implementada através da utilização de uma árvore binária que define os comportamentos dos agentes em determinadas situações do ambiente. Os nós da árvore que não são folhas representam testes sobre as condições do ambiente, os arcos representam o resultado positivo ou negativo do teste, e as folhas são ações que devem ser executadas se há um caminho entre a raiz e aquela folha. Portanto, o ramo que parte da raiz forma o conjunto de condições do ambiente que determinam que certa ação, representada pela folha, seja realizada. Isto significa que podem ser construídos diversos testes de percepção sobre as informações visuais, de forma que apenas a ação correspondente àquela situação do ambiente é executada. A arquitetura é puramente reativa.

Tanto os testes quanto as ações são componentes visuais, o que permite a rápida geração de diferentes combinações de comportamentos. Por ser uma ferramenta visual, esta tarefa de geração é bastante simplificada, o que permite que sejam testadas uma grande variedade de combinações de comportamentos. Um exemplo de árvore binária desenvolvida pode ser visualizado na Fig. 3.

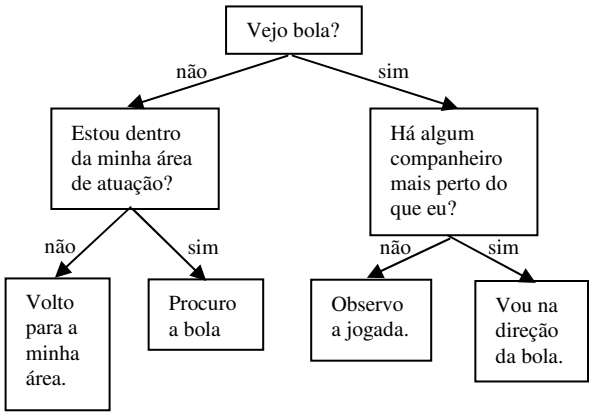


Fig. 3. Árvore binária de comportamentos

O terceiro módulo integra os resultados dos dois módulos apresentados anteriormente, e fornece como resultado um arquivo com uma série de parâmetros. É importante salientar que a definição da área implica em parâmetros que delimitam o espaço no campo dentro do qual o agente deve atuar. Mas a determinação de como será o comportamento do agente, mesmo quando este por algum motivo excedeu seus limites e percebeu estar fora de sua área, será feita através da definição de comportamentos com a árvore binária.

Cada agente jogador é implementado através de um mesmo programa em C++. Este programa lê o arquivo de parâmetros gerado pelo *framework* no momento em que é carregado, antes do início da partida. São estes parâmetros que configuram os agentes genéricos, de forma que apresentem comportamentos específicos, distintos uns dos outros.

V. Definição da área de atuação do agente

Um aspecto importante para a atuação coerente da equipe é o posicionamento dos agentes dentro do campo. A determinação do posicionamento inicial dos agentes no campo de jogo, e a manutenção de uma distribuição coerente durante o decorrer de toda a partida, são aspectos estratégicos importantes tanto para a atuação dos agentes na obtenção do objetivo do gol, quanto na sua atuação defensiva, evitando que a equipe adversária conquiste semelhante objetivo. Como resultados mais evidentes de falhas na distribuição estão o acúmulo de jogadores da mesma equipe em pequenos espaços e o surgimento de espaços desocupados.

A principal dificuldade para a delimitação da área está na forma como a percepção do agente sobre seu posicionamento dentro de campo é implementada no simulador. Em nenhum momento o agente recebe do simulador informações sobre sua posição absoluta, representada por coordenadas cartesianas. O que o agente possui é informações a respeito de sua posição com relação aos *flags* pré-estabelecidos, denominados aqui como marcadores, que estão distribuídos por todo o campo de jogo e estão identificados pelos pontos na Fig. 4.

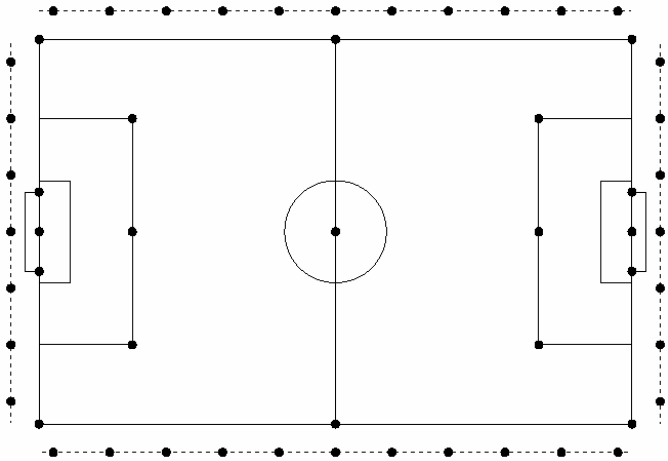


Fig. 4. O campo de jogo e os marcadores

Outro complicador está no fato de que o agente pode girar em torno do seu ponto central, e os indicadores visualizados têm sua posição determinada por um valor que indica a distância e outro que indica o ângulo deste indicador com relação ao centro do campo de visão. Por isso, um agente posicionado em um ponto do campo pode visualizar um marcador de formas diferentes, desde que a rotação do seu corpo sofra variação.

A construção de um sistema de posicionamento do agente em função de apenas um marcador, como o que marca o centro do campo, por exemplo, também é inviável, pelo motivo citado anteriormente e porque o campo de visão do agente é limitado, de forma que nem sempre a informação da posição relativa ao marcador específico está disponível nas informações sensoriais recebidas do simulador.

A cada uma das linhas laterais (no simulador aparecem como linhas horizontais) e de fundo (verticais) do campo de jogo está associado um conjunto de marcadores, colineares e posicionados cinco metros para fora de campo, onde conjunto formando uma reta paralela à margem associada. As quatro retas formadas pelos marcadores são identificadas na Fig. 4 como linhas pontilhadas.

O módulo de definição da área de atuação apresenta para o construtor do agente uma interface (Fig. 5) que representa o campo de jogo, sobre a qual deve ser desenhado um retângulo que representa a área em cujo interior o agente limitará sua atuação. Como a área marcada será sempre um retângulo, suas arestas também são paralelas às linhas que delimitam as margens do campo e, por conseqüência, às retas formadas pelos marcadores. Isto possibilita que se determine a posição do agente em relação à aresta que delimita sua área de atuação diretamente a partir dos dados referentes aos marcadores que demarcam a linha lateral paralela à aresta. O agente não percebe, durante a realização da partida, a aresta de sua área de atuação, nem as linhas que demarcam os limites do campo, apenas os marcadores. Portanto, é em função deles que a posição do agente em relação à sua aresta deve ser calculada.

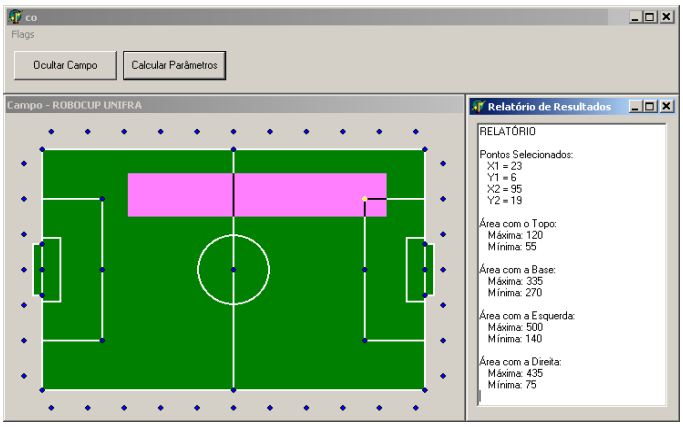


Fig. 5. Interface do Módulo de Definição de Áreas de Atuação

Para que isso seja possível, é necessário estabelecer a relação entre a aresta da área delimitada e a linha de marcadores. Como estas duas linhas são paralelas, bastaria testar a distância do agente em relação à reta de marcadores, verificando se esta distância é maior ou menor que a distância da aresta que demarca seu limite de atuação. Mas esta informação não é fornecida ao agente pelo simulador, o que torna necessária a busca por uma alternativa.

Esta alternativa encontra-se na propriedade de que, dadas duas retas paralelas r e s, todos os triângulos formados por dois pontos fixos Q e R de s e qualquer ponto de r têm área idêntica.

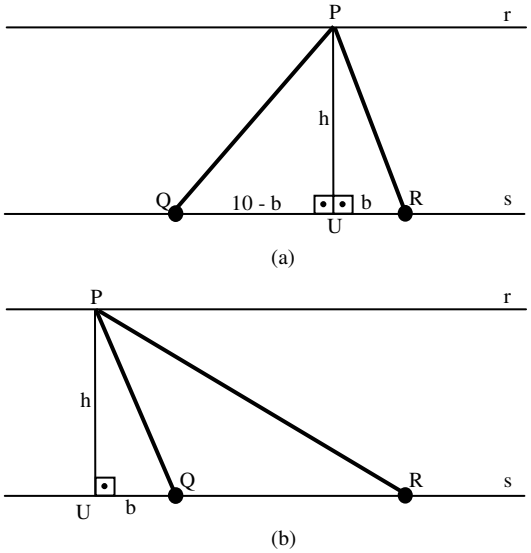
Este conceito pode ser ilustrado pela Fig. 6, na qual temos representados os dois pontos de s, Q e R, e duas possibilidades para a marcação de um ponto P sobre a reta r. Em (a), a área do triângulo PQR é a soma das áreas dos triângulos PQU e PRU:

$$Área = h.b.sen90º / 2 + h.(10 - b).sen90º / 2 = 5h \qquad (1)$$

Em (b), a área do triângulo PQR é a área do triângulo PUQ subtraída da área do triângulo PUR:

$$Área = h.(10 + b).sen90º / 2 - h.b.sen90º / 2 = 5h \qquad (2)$$

Como pode ser percebido, o fato de a distância entre os pontos Q e R ser fixa faz com que a área do triângulo seja obtida em função apenas de h, que é a distância entre as retas



paralelas, e seja idêntica para qualquer ponto colocado sobre r.

Fig. 6. Áreas dos triângulos

Ao se considerar a reta s como a linha formada pelos marcadores, sendo Q e R dois marcadores contíguos, cuja distância é fixa (10 metros), e r a aresta do retângulo que delimita a área, temos que o agente colocado sobre qualquer ponto desta aresta formará com dois marcadores contíguos um triângulo de área constante.

É esta propriedade que servirá como parâmetro para o agente delimitar sua área de atuação. A ferramenta de

demarcação de área captura o retângulo definido pelo usuário e gera oito parâmetros que relacionam cada uma das arestas do triângulo às duas retas paralelas formadas pelos marcadores. Os oito parâmetros são: dois para cada aresta horizontal do retângulo, que determinam as relações destas arestas com as retas formadas pelos marcadores das duas laterais do campo; e dois para cada aresta vertical do retângulo, que determinam as relações destas arestas com as retas formadas pelos marcadores das duas linhas de fundo do campo.

O motivo de existires dois parâmetros para cada uma das arestas do retângulo delimitador da área é a visão limitada do agente. Em um determinado momento, é bastante provável que o agente perceba apenas os marcadores de um dos lados do campo, ou de uma das linhas de fundo. O primeiro parâmetro associa a aresta do retângulo à linha de marcadores paralela mais próxima e indica a área mínima que o agente pode formar com os marcadores desta linha de forma a não transpor seu limite de atuação. A outra relaciona a aresta à linha de marcadores mais distante e indica a área máxima que o agente pode formar com os marcadores desta linha. Por outro lado, a existência de parâmetros que definem as duas arestas paralelas em função de uma mesma linha de marcadores paralela permite que o agente verifique se não transpôs nenhuma delas observando apenas esta linha. Se a aresta fosse definida apenas pela área formada por um de seus pontos com os marcadores da aresta mais próxima, a verificação da área exigiria que o agente observasse as duas linhas de marcadores ao mesmo tempo.

Para o módulo de demarcação de área, o cálculo dos oito parâmetros é bastante simples. Como a imagem apresentada para o usuário está em escala, é possível obter a distância entre as arestas e as retas diretamente das coordenadas do retângulo desenhado. Cada parâmetro é calculado como a área do triângulo formado por um ponto qualquer da aresta e dois marcadores contíguos da reta relacionada. A área é a metade do produto entre a distância entre as retas e a distância entre os marcadores contíguos, que é dez metros, por definição do simulador.

Mas para o agente jogador, que deve calcular a área formada pela sua posição em relação aos marcadores contínuos, para compará-la com os parâmetros máximo e mínimo o cálculo não é tão simples. A distância entre o ponto e a reta, equivalente a h na Fig. 4., não é uma informação disponibilizada aos jogadores pelo simulador *Soccersever*. Ao agente é disponibilizada a informação sobre a sua distância aos marcadores que estão no seu campo de visão (equivalentes às distâncias entre P e Q e entre P e R) e o ângulo em que estes se encontram com relação ao centro do campo de visão, exemplificado na Fig. 7.

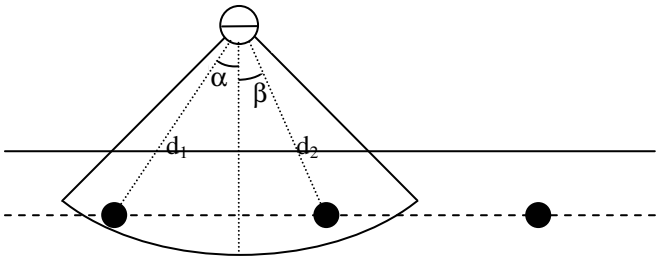


Fig. 7. A visão do agente: distâncias d1 e d2 e ângulos α e β

Os elementos capturados no campo de visão que estiverem colocados à direita do eixo têm um ângulo com valor positivo, e à esquerda negativo. Por isso, o cálculo do ângulo entre os dois marcadores deve ser calculado levando em consideração o fato de estarem do mesmo lado, possuindo ambos valores positivos ou negativos, ou estarem em lados diferentes, quando um ângulo é positivo e outro negativo. No primeiro caso, o ângulo γ entre os marcadores é dado por

$$\gamma = | \alpha + \beta | \quad (3)$$

e no segundo caso por

$$\gamma = | \alpha | + | \beta | \quad (4)$$

A área do triângulo formado pelo agente e os marcadores consecutivos é dada por

$$Área = d_1 . d_2 . sen\ \gamma / 2 \quad (5)$$

Com a área calculada, o agente verifica seu valor com relação aos dois parâmetros de áreas máxima e mínima que pode formar com a linha dos marcadores usados no cálculo. Se a área for menor que o menor dos dois parâmetros, o agente esta muito próximo dos marcadores e está fora do retângulo de sua área de atuação. Por outro lado, se a área for maior do que o maior dos dois parâmetros, o agente está muito distante dos marcadores, e também saiu de sua área, mas pelo outro lado.

VI. *CONCLUSÕES*

A verificação de área descrita apresenta o posicionamento do agente com relação a apenas uma das dimensões do retângulo, vertical ou horizontal. A posição relativa à outra dimensão deve ser calculada com relação à outra linha de marcadores, o que, em certas ocasiões, exige do agente uma movimentação que coloque marcadores desta linha no campo de visão. A realização de movimentação apenas para descobrir o posicionamento prejudica a performance geral do agente. Por isso, a verificação deve ser feita apenas em relação aos marcadores capturados no campo de visão, e os cálculos que não puderem ser realizados devem ser deixados para outro momento, quando as informações que estavam faltando

estiverem disponíveis.

É importante também que a manutenção do agente dentro da área definida não seja a maior prioridade de seu comportamento. O ideal é que a área sirva como um guia, de forma que o agente procure manter-se em seu interior, mas que tenha liberdade para sair em certas situações.

Conforme descrito, o desenvolvimento do agente no *framework* resulta em um arquivo de parâmetros que são lidos por um agente genérico implementado em C++ e definem seu comportamento. Esta forma de implementação apresenta performance inferior a agentes cujos comportamentos são implementados diretamente no programa fonte. Portanto, o *framework* apresenta uma ferramenta para a definição e teste de diferentes agentes, com as mais variadas combinações de comportamentos e abordagens frente a diferentes situações. Mas, após a escolha da melhor combinação, o agente para a competição deve ser implementado de forma a embutir diretamente no código o comportamento escolhido.

O fato de o resultado das escolhas de comportamento estarem colocadas em um arquivo de parâmetros possibilita também que o *framework* seja utilizado na definição de agentes implementados em qualquer linguagem, bastando que esteja apto a ler o arquivo.

VII. *REFERÊNCIAS*

[1] H. Kitano, "RoboCup: The Robot World Cup Initiative", in Proc. 1995 *International Joint Conference on Artificial Intelligence*.

[2] P. Maes, "The Agent Network Architecture (ANA)", *Sigart Bulletin*, vol. 2(4), p. 115-120, 1991.

[3] R. A. Brooks, "A Robust Layered Control System for a Mobile Robot". *IEEE Journal of Robotics and Automation*, vol. 2(1), p. 14-23, 1986.