

O AGENTE TREINADOR E O CÁLCULO DE ESTATÍSTICAS DO JOGO NO FUTEBOL ROBÓTICO SIMULADO

Rui Sampaio^{1,2}, Luís Paulo Reis^{1,2}, Nuno Lau^{3,4}

ei00058@fe.up.pt, lpreis@fe.up.pt, lau@det.ua.pt

¹FEUP – Faculdade de Engenharia da Universidade do Porto, Portugal

²LIACC – Laboratório de Inteligência Artificial e Ciência de Computadores, Universidade de Aveiro, Portugal

³DET – Departamento de Electrónica e Telecomunicações, Universidade de Aveiro, Portugal

⁴IEETA - Instituto de Engenharia Electrónica e Telemática de Aveiro, Portugal

Resumo: A liga de simulação de futebol robótico é uma competição simulada de robôs virtuais que jogam futebol. Um desses robôs virtuais é o treinador da equipa, elemento omnisciente que tem como função dar conhecimento aos jogadores da melhor estratégia e comportamento a seguir. O treinador tem uma série de mecanismos que lhe permitem analisar o jogo sob vários pontos de vista, de modo a poder tomar uma decisão estratégica para a comunicar aos jogadores.

Este artigo apresenta uma forma de analisar um jogo de futebol robótico simulado por meio do cálculo de estatísticas relacionadas com as diversas ocorrências do jogo. Estas estatísticas permitem ao treinador analisar o jogo, definir uma tática e transmiti-la aos jogadores.

Keywords: Artificial Intelligence, RoboCup, Opponent Modeling, Coaching, Game Analysis, Simulation, Robotic Soccer.

1. INTRODUÇÃO

Este artigo apresenta uma abordagem para a criação de um agente treinador capaz de treinar uma equipa de futebol robótico que compete na liga de simulação do RoboCup.

Na liga de futebol robótico simulado, cada agente simula o comportamento de um jogador, ou de um treinador de futebol. Os agentes são programas que comunicam entre si partilhando informação, de forma a coordenarem as suas acções para se obter o melhor resultado possível.

O principal interesse do futebol robótico consiste na análise das percepções que cada agente capta e na escolha inteligente da melhor acção de forma a otimizar o desempenho da equipa. O treinador tem um papel muito importante no comportamento da equipa, pois tem um conhecimento global do jogo melhor que o dos jogadores, podendo extrair mais informação do jogo e ao mesmo tempo informar os

jogadores da melhor forma de actuar tanto a nível global como individual.

2. LIGA DE SIMULAÇÃO DO ROBOCUP

A liga de simulação do RoboCup é baseada num simulador chamado *soccer server*. Os jogos podem ser observados usando o *soccer monitor* em tempo real, ou vendo uma repetição de um jogo através de *logfile*s.

Um *logfile* é criado sempre que se disputa um jogo, armazenando toda a informação respeitante aos objectos em campo (posição, velocidade, direcção, energia...), assim como as mensagens emitidas pelos jogadores e árbitro.

O simulador funciona usando uma arquitectura cliente servidor, em que o servidor é o simulador e os clientes são cada jogador e treinador de cada equipa. A comunicação é realizada usando sockets UDP, e cada cliente liga-se ao servidor através de uma porta, através da qual comunica com o servidor. O servidor

envia informação sensorial aos jogadores e estes, por sua vez, enviam as acções que desejam executar ao servidor.

Cada jogador tem uma representação do estado do mundo com as posições dos outros jogadores, da bola e direcções. O estado do mundo é actualizado consoante as percepções que o jogador recebe. Ter um bom estado do mundo é fundamental para o bom funcionamento da equipa.

A unidade de tempo mínima do simulador, um ciclo, corresponde a 100 ms, por isso, cada jogador tem que decidir neste intervalo de tempo a acção que deseja executar consoante o seu estado do mundo.

As sensações que os agentes jogadores recebem podem ser de três tipos: auditiva, visual e física.

2.1 Sensações

2.1.1 Sensações auditivas

As sensações auditivas provêm da comunicação oral de um outro qualquer jogador ou do treinador. Se a mensagem enviada for proveniente de um adversário, o mais certo é o jogador não saber descodificá-la, de modo que só interessam as mensagens do treinador e dos colegas de equipa. As mensagens orais têm limite de tamanho de 10 bytes e cada jogador só recebe uma mensagem de cada equipa em cada ciclo, o que significa que se muitos agentes estiverem a falar, a maioria das mensagens não serão recebidas. Os jogadores podem dar especial atenção a outro jogador usando o comando *AttentionTo*, que lhes permite ouvir preferencialmente as mensagens que esse jogador transmita. Caso o jogador não transmita nada, é como se o comando não tivesse sido executado. Cada jogador recebe do servidor uma sensação auditiva e o seu conteúdo se estiver no raio de audição da mensagem comunicada. Esta só é recebida no ciclo seguinte a ter sido enviada.

Com estas condicionantes a comunicação oral fica muito limitada e deve ser usada só em situações relevantes.

2.1.2 Sensações visuais

As sensações do tipo visual são recebidas, por defeito, em intervalos de 150 ms, com a informação sobre a posição de cada objecto e da sua respectiva direcção que esteja no campo de visão do jogador.

Cada jogador pode definir o ângulo de visão e a qualidade da informação a receber. O ângulo de visão diz respeito à quantidade de área a abranger e a qualidade da informação relaciona-se com os erros de exactidão que estão associados a esta percepção.

A qualidade pode ser boa ou má e o ângulo de visão pode ser estreito, normal ou largo. Por defeito o ângulo é normal e a qualidade é boa, sendo a informação recebida em intervalos de 150 ms. Quanto pior for a qualidade, menor é o intervalo de tempo, e quanto maior a abertura do ângulo de visão, também maior será o intervalo de tempo.

2.1.3 Sensações físicas

As sensações físicas dizem respeito ao estado físico do jogador: ângulo do pescoço do jogador em relação ao corpo, a sua energia, velocidade e qualidade e ângulo de visão. Esta informação é recebida todos os ciclos.

2.2 Acções dos jogadores

Durante o jogo os jogadores podem executar vários tipos de acções. Num ciclo podem ser executadas uma ou mais acções dependendo da acção escolhida.

As acções que os jogadores podem executar são:

- Dash – é utilizado para acelerar um jogador na direcção do seu corpo com uma dada potência.
- Turn – este comando serve para o jogador rodar todo o seu corpo num determinado ângulo, mudando a sua direcção.
- Move – um jogador apenas se pode mover para uma posição qualquer do campo instantaneamente no início de cada metade do jogo, após um golo de uma equipa ou pelo guarda-redes após agarrar a bola.
- Kick – o comando kick serve para o jogador chutar a bola numa determinada direcção com força controlada
- Tackle – esta acção serve para tentar tirar a bola ao adversário. Quanto mais perto estiver a bola maior a probabilidade de retirar a bola do adversário. O jogador permanece em seguida 10 ciclos sem se mover.
- Catch – esta acção pode apenas ser executada pelo guarda-redes, e para conseguir apanhar a bola esta tem que estar suficiente próxima.
- Turn_neck – permite a um jogador virar o pescoço numa outra direcção para poder observar o que se passa nas várias zonas do campo.
- Change_view – permite aos jogadores mudarem os parâmetros de visão: a qualidade e ângulo de visão.
- Attentionto – usando este comando um jogador pode dar atenção a outro jogador, ouvindo as mensagens enviadas por este preferencialmente.
- Say – permite aos jogadores comunicarem entre si via oral, transmitindo mensagens de 10 bytes.
- Pointto – é o comando que se usa quando um jogador que apontar numa determinada direcção.

Os jogadores apenas podem executar uma acção durante um ciclo se esta for do tipo dash, turn, move, kick, tackle ou catch

2.3 Decisões dos jogadores

Das várias possibilidades de acções que o jogador pode executar, o jogador escolhe uma consoante o estado actual do mundo e a estratégia que a equipa segue.

Se o jogador tem a posse da bola, pode seguir com ela, rematar ou passar. Se não tem a posse da bola verifica se quem tem a posse de bola é um jogador da sua equipa ou não, comportando-se de forma ofensiva ou defensiva, posicionando-se consoante o tipo de jogador que é. No modo defensivo um jogador pode cobrir um adversário ou tentar tirar-lhe a bola se for o jogador mais próximo dela.

3. TREINADOR

O simulador de futebol robótico permite a utilização de dois tipos de treinadores: o treinador *off-line* e o treinador *on-line*.

Ambos os treinadores recebem do servidor toda a informação sem erros respeitante aos objectos em campo. Recebem também as mensagens trocadas entre os jogadores e podem comunicar oralmente com estes.

O treinador *off-line* é usado apenas como auxiliar ao desenvolvimento da equipa, podendo exercer maior controlo na equipa e no jogo, controlando o modo de jogo, desactivando o árbitro artificial se necessário. Pode ainda movimentar os jogadores de ambas as equipas e a bola para qualquer ponto do campo, assim como atribuir velocidades e direcções aos jogadores, o que é extremamente útil para testar e ensaiar diversas situações.

O treinador *on-line* é usado nas competições e jogos e as suas capacidades são muito limitadas de modo a não controlar o modo de jogo ou os jogadores.

O treinador pode influenciar o comportamento da equipa globalmente e de cada jogador caso não esteja a ter um comportamento desejado.

Para fazer a gestão do comportamento da equipa de forma correcta, o treinador necessita de ter informações sobre o jogo que está a decorrer, e da forma como os jogadores se comportam. A esta informação chamamos estatísticas de jogo.

O treinador é capaz de calcular as estatísticas em tempo real, ou seja enquanto o jogo decorre, ou mesmo através de uma gravação prévia do jogo, de modo a poder analisar o que correu bem ou mal e a preparar-se a si e à equipa de uma forma mais eficaz.

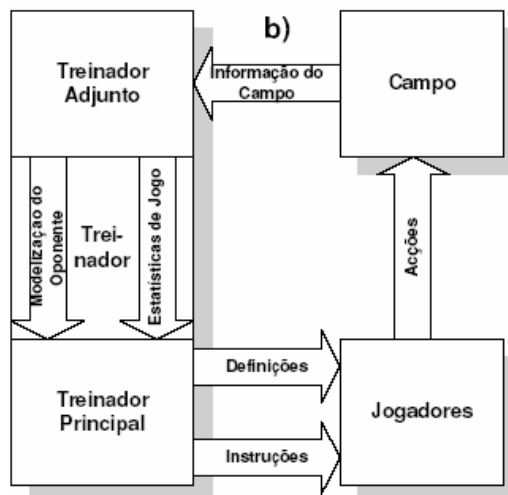


Fig. 1. Arquitectura do agente treinador (Reis, 2003B)

4. ANÁLISE DO JOGO

O treinador pode fazer a análise do jogo em duas situações distintas: em tempo real, ou através de *logfiles*.

A análise em tempo real dá-se ao mesmo tempo que decorre um jogo. O treinador recolhe a informação à medida dos acontecimentos, analisa-a e comunica com os jogadores.

A análise do jogo através da leitura de *logfiles* permite ao treinador fazer uma análise de vários jogos diferentes e com diversas equipas adversárias, de modo a obter informação mais detalhada sobre o comportamento da equipa com determinada equipa ou em situações mais variadas. Os *logfiles* de um jogo contêm mais informação do que a obtida por análise em tempo real. O cálculo das estatísticas é mais simples e estas podem ser mais precisas devido à maior facilidade de cálculo.

5. COMPETIÇÃO DE TREINADOR

Alem da competição de simulação, em que é posto à prova o desempenho da equipa, no qual o treinador tem um papel preponderante, existe uma competição de treinadores. É formada uma equipa constituída por jogadores de equipas diversas, e o treinador tem que gerir e dar instruções aos vários jogadores com inteligências diferentes. A comunicação standard é realizada na linguagem *CLang*, que supostamente todos os jogadores conhecem e sabem interpretar. Ganha o treinador que conseguir obter o melhor resultado final do jogo. É fundamental que o treinador neste tipo de competição não deixe os jogadores actuarem como lhes apetece, pois irá haver uma grande divergência de raciocínio e estratégia individual. O treinador é o elemento unificador da equipa e tem como função pôr os jogadores a trabalhar em equipa eficazmente.

6. ESTATÍSTICAS E CÁLCULO

O treinador apenas possui informação sobre a posição e direcção dos objectos em campo em cada ciclo de jogo e das mensagens verbais emitidas, esta informação não é suficiente para se perceber o estado do jogo de modo a perceber as falhas e oportunidades para definir uma estratégia vencedora. Com base na informação dada, o treinador necessita de calcular as estatísticas do jogo, de forma a filtrar aquelas que se mostram relevantes e alterar a estratégia da equipa ou de apenas um jogador.

As estatísticas relevantes para uma boa análise do jogo são as que estão definidas na linguagem standard **COACH UNILANG** (Reis e Lau, 2001) e são: resultado do jogo, tempo de jogo parado, ocorrências (cantos, foras...), oportunidades de golo, posse de bola, posição dos jogadores, posse de bola, ataques, assistências a remates, perdas e recuperação de bolas, passes e remates à baliza. Cada estatística contém informação sobre as regiões, os jogadores intervenientes e o tempo em que ocorreu no caso de ser relevante.

6.1 Estatísticas sem cálculos

As estatísticas sem cálculos são aquelas que se podem obter directamente por observação do jogo. Estas não requerem nenhum cálculo adicional. Nestas estatísticas estão englobadas: o tempo de jogo, o tempo parado, as ocorrências, a posição do jogador e a posição da bola.

6.2 Estatísticas com cálculos

Estas estatísticas são as que requerem algum cálculo e uma definição formal, por serem às vezes um pouco ambíguas.

Estas estatísticas são: oportunidades de golo, posse de bola de um jogador, ataques, assistências a remates, perdas e recuperação de bola, passes e remates

6.3 Definição formal de estatísticas de jogo

De seguida estão apresentadas as estatísticas formalmente definidas na **COACH UNILANG** (Reis e Lau, 2001)

```
<ACTION> ::= shoot | pass | forward | dribble
           | run | hold | clear | any
<RECOVERY> ::= interception | tackle |
            mark_pass_line | mark_player |
            goto_ball | strategical_move | any
<ACTION_RESULT> ::= success | out |
                 interception | stolen | fail |
                 opp_goalie_catch | our_goalie_catch |
                 any
<REC_RESULT> ::= <ACTION_RESULT> |
                stolen_other | interception_other

<GAME_PLAYMODE> ::=
  playon |
```

```
corner |
offside |
kickin |
kickoff |
free_kick |
goalie_free_kick |
goal_kick |
any
```

```
<STATISTICS> ::=
(clear) |
(game_time <TIME>) |
(game_result <PERIOD> [integer] [integer]) |
(stopped_time <PERIOD> <TIME>) |
(game_occurrence <GAME_PLAYMODE> <REGION>
 <TEAM> <PERIOD> <COUNT>) |
(action <ACTION> <REGION_FROM> <REGION_TO>
 <TEAM> <PLAYER> <PERIOD> <ACTION_RESULT>
 <COUNT>) |
(recovery <RECOVERY> <REGION> <TEAM>
 <PLAYER> <PERIOD> <INT_RESULT>
 <COUNT>) |
(ball_possession <REGION> <TEAM> <PLAYER>
 <PERIOD> <COUNT>) |
(player_position <REGION> <TEAM> <PLAYER>
 <PERIOD> <COUNT>) |
(attack <REGION_FROM> <REGION_TO> <TEAM>
 <PERIOD> <NPASSES> <ACTION_RESULT>
 <COUNT>) |
(assist <REGION_FROM> <REGION_TO> <TEAM>
 <PLAYER> <PLAYER_TO> <PERIOD> <ACTION>
 <ACTION_RESULT> <COUNT>) |
(ball_losses <ACTION> <REGION> <TEAM>
 <PLAYER> <PERIOD> <COUNT>) |
(ball_recoveries <RECOVERY> <REGION> <TEAM>
 <PLAYER> <PERIOD> <COUNT>) |
(action_to_player <ACTION> <REGION_FROM>
 <REGION_TO> <TEAM> <PLAYER> <PLAYER_TO>
 <PERIOD> <ACTION_RESULT> <COUNT>)
```

6.4 Definição algorítmica de estatísticas de jogo

Oportunidades de golo ocorrem quando um jogador tem a posse da bola e se encontra numa área perigosa, tipicamente a grande área.

A posse de bola de jogador é a situação em que um jogador se encontra com a bola na sua área de chuto (ou perto), controlada (pouca velocidade) e sem adversários na mesma situação. Também se considera posse de bola o momento do passe quando executado com sucesso.

Um ataque dá-se quando a bola se desloca em direcção à baliza adversária por meio de passes entre jogadores, ou por meio de um jogador que segue sozinho com a bola. A distância percorrida tem que ser superior a um parâmetro ou a bola tem que passar um limite definido, controlada pelos jogadores para o ataque ocorrer.

Uma assistência é um passe feito para um jogador criando uma oportunidade de golo, ou seguindo-se de um remate.

A perda de bola dá-se quando um jogador possui o controle da bola e a perde sem ter feito um passe ou passando a posse da bola a pertencer ao adversário.

A recuperação da bola dá-se quando um jogador tem posse de bola e esta não era da sua equipa no momento anterior, é o inverso da perda de bola. A recuperação de bola de uma equipa representa a perda da bola para a outra.

O passe dá-se quando um jogador chuta a bola com força suficiente na direcção de um jogador da mesma equipa. O passe tem sucesso se a bola passa fica na posse desse jogador.

Um remate é um chute com força suficiente na direcção (ou quase) da baliza adversária.

7. ARMAZENAMENTO INTERNO DAS ESTATÍSTICAS

O armazenamento das estatísticas é efectuado através da definição de estruturas em C. Cada estrutura contém a informação necessária para a definição de uma estatística. A informação de cada estatística é armazenada integralmente num vector de estruturas.

7.1 Estatísticas discretas

Uma estatística discreta é uma ocorrência que acontece em ciclos do jogo pontuais. Para estas estatísticas, cada elemento do vector contém informação sobre cada ocorrência da estatística.

Por exemplo, no vector dos passes, cada elemento é uma estrutura contendo informação relevante do passe.

```
typedef struct
{
    Vector point_from;
    Vector point_to;
    int team;
    int cycle;
    int passes;
    bool result;
} attack;

typedef struct
{
    Vector point_from;
    Vector point_to;
    int team;
    int player;
    int player_to;
    int cycle;
} assist;

typedef struct
{
    Vector point;
    int team;
    int player;
    int cycle;
} loss;

typedef struct
{
    Vector point;
    int team;
    int player;
    int cycle;
} recoverie;
```

```
typedef struct
{
    Vector point_from;
    Vector point_to;
    int team;
    int player;
    int player_to;
    int cycle;
    bool result;
} pass;

typedef struct
{
    p_mode pmode;
    Vector point;
    int team;
    int cycle;
} occurrence;

typedef struct
{
    Vector point;
    int team;
    int cycle;
} oportunit;
```

7.2 Estatísticas contínuas

Uma estatística contínua é constantemente alterada em cada ciclo do jogo, de modo que seria impraticável o seu armazenamento da mesma forma que as estatísticas discretas. Assim, o armazenamento destas estatísticas é efectuado, não guardando a informação relevante da estatística, mas sim da informação que permita o seu cálculo de forma eficaz.

Por exemplo, para a estatística de posse e bola guarda-se apenas em cada ciclo do jogo a posição da bola e dos jogadores, de forma a conseguir-se calcular a posse de bola de um determinado jogador, ou numa determinada região do campo num determinado período de tempo.

```
int game_time;
int time_stopped;

typedef struct
{
    int cycle;
    int myresult;
    int theirresult;
} result;

typedef struct
{
    int cycle;
    int time;
} stopped;

int positions [78000][4];
```

8. ARMAZENAMENTO EXTERNO DAS ESTATÍSTICAS

No final de cada jogo o treinador guarda as estatísticas do jogo que se pretendam seleccionar num ficheiro no formato da **COACH UNILANG**.

Para cada jogo umas estatísticas podem revelar-se mais interessantes que outras, e noutros jogos o contrário pode verificar-se. Assim, no final de cada

jogo (em tempo real ou não), o utilizador pode escolher de todas as estatísticas calculadas, as que pretende consultar no momento e guardá-las para futura consulta. As estatísticas tanto podem ser globais como referentes a determinado jogador ou região do campo num certo intervalo de tempo.

9. CONCLUSÕES

O treinador embora não seja indispensável para uma equipa de futebol robótico, pode ser um factor de grande optimização do funcionamento da equipa. Para isso, o treinador não pode ser apenas um espectador mas deve intervir na decisão de táticas e comportamentos dos jogadores. De forma a conseguir realizar decisões táticas, o treinador necessita de analisar correctamente o jogo, sendo para tal indispensável calcular um elevado conjunto de estatísticas do jogo.

AGRADECIMENTOS

Este projecto tem o apoio da Fundação para a Ciência e a Tecnologia FCT-POSI/ROBO/43910/2002 Projecto – “FC Portugal – Novas Metodologias de Coordenação Aplicadas na Liga de Simulação”.

REFERÊNCIAS

- Reis L. P.e Lau, N., (2001)
COACH UNILANG – A Standard Language for Coaching a (Robo) Soccer Team, in Andreas Birk, Silvia Coradeschi e Satoshi Tadokoro, editors, RoboCup-2001: Robot Soccer World Cup V, Springer Verlag Lecture Notes in Artificial Intelligence, Berlin.
- Reis, L. P., (2003A)
Coordenação em Sistemas Multi-Agente: Aplicações na Gestão Universitária e Futebol Robótico, Análise do Domínio de Aplicação: Futebol Robótico (Páginas 239-298),
- Reis, L. P., (2003B)
Coordenação em Sistemas Multi-Agente: Aplicações na Gestão Universitária e Futebol Robótico, Coordenação em SMA no Futebol Robótico (Páginas 299-404), PhD
- Reis L. P.e Lau, N, “FC Portugal Team Description: Robocup 2000 Simulation League Champion”, Robocup-2000: Robot Soccer World Cup IV, Peter Stone, Tucker Balch, e Gerhard Kraetzschmar editors, LNAI 2019, 29-40, Springer Verlag, Berlin, 2001.
- UvA Trilearn 2003 coach framework source code
http://carol.wins.uva.nl/~jellekok/robocup/2003/index_en.html
- Robocup simulation league website
<http://sserver.sourceforge.net/>
- Soccer Server manual
<http://sserver.sf.net/docs/manual.pdf>