

Implementação do time PET_Soccer_2D

Gilberto A. S. Segundo, Marcos V. B. S. de Lima, Rafael C. Favoreto e Ramon G. Schiavo

Resumo—Esse artigo descreve o time PET_Soccer_2D, que será usado na categoria RoboCup Soccer Simulation 2D, durante a LARC 2008 - V Competição Latino-Americana de Robótica a ser realizada juntamente com o 19º Simpósio Brasileiro de Inteligência Artificial. Inicialmente apresenta-se o grupo que implementou o time e posteriormente as mudanças nas estratégias principais do time. Por fim, este artigo trás as expectativas em relação a competição desse ano e perspectivas de trabalhos futuros.

I. INTRODUÇÃO

Este ano o time PET_Soccer_2D participa pela terceira vez nessa competição. Ele é desenvolvido pelos bolsistas do PET (Programa de Educação Tutorial) Engenharia de Computação, da Universidade Federal do Espírito Santo[1].

Atualmente o time é implementado basicamente por algoritmos determinísticos, mas em diversos pontos do código é feita a seleção da melhor jogada, analisando-se as diversas possíveis jogadas de acordo com as condições atuais da partida.

A. O time base UvA_trilearn_2003

Os desenvolvedores do UvA Trilearn 2003 Soccer Simulation [2], time vencedor da RoboCup-2003, disponibilizaram parte do código fonte do seu time para que novos competidores tenham um ponto de partida mais adequado e eficiente para o desenvolvimento do seu próprio time.

O time disponibilizado já é capaz de se organizar em campo, chutar a bola, fazer passes, marcar o adversário, entre outros. Mas a maioria dessas funções não são usadas e o que o time faz basicamente é correr até a bola e chutar para o gol.

A equipe desenvolvedora do PET_Soccer_2D usou esse time base para iniciar suas pesquisas nessa área. As principais modificações envolvem o uso estratégico das funções já existentes, modificações dessas e desenvolvimento de novas funções.

II. IMPLEMENTAÇÃO DO TIME

A seguir são descritas as principais alterações feitas no time desde a última competição nacional, ocorrida em outubro de 2007.

Todos os autores são estudantes de Engenharia de Computação da UFES e bolsistas do Programa de Educação Tutorial - PET Engenharia de Computação, Depto. de Informática, Universidade Federal do Espírito Santo, Av. Fernando Ferrari, nº 314, Goiabeiras - Vitória-ES, CEP: 29060-900 {gilberto.segundo, mvbasuli, rafaelcampanharo, ramon.g.schiavo}@gmail.com

A. Goleiro

Diferentemente dos demais jogadores, o goleiro deve ter sua implementação diferenciada, uma vez que possui habilidades e campo de atuação diferentes.

O goleiro fica posicionado em uma linha paralela a linha do gol, sempre entre as duas traves, sendo que sua posição exata é calculada através da posição atual da bola. São traçadas duas retas, uma entre a bola e a trave direita e outra entre a bola e a trave esquerda. O ângulo formado por essas duas retas é calculado e então uma terceira linha é criada na bissetriz desse ângulo. O ponto de posicionamento ideal do goleiro é dado pela interseção da reta paralela à linha do gol com a reta da bissetriz do ângulo entre a bola e as duas traves.

Além do posicionamento, o goleiro deve ter uma boa estratégia de interceptação da bola. Quando a bola é chutada para o gol, o goleiro deve decidir o melhor ponto de interceptação. Porém, através de diversos testes, percebeu-se que o cálculo direto do melhor ponto de interceptação é muito impreciso, uma vez que o módulo da velocidade e a direção da bola sofrem pequenas alterações randômicas no decorrer de sua trajetória. Com isso, decidiu-se calcular dois pontos principais de interceptação da bola: o primeiro sendo o ponto no qual a bola cruza a linha vertical que passa pela posição atual do goleiro; e o segundo o ponto no qual a bola cruza a linha horizontal que passa pela posição atual do goleiro. Calcula-se então quanto tempo o goleiro e a bola levarão para chegar nesses pontos e decide-se ir para o ponto no qual a folga de tempo é maior. Caso perceba-se que em ambos os pontos o goleiro chegará após a bola, o goleiro vai para o ponto que a bola cruzará o gol, numa última tentativa de interceptar a bola.

Na última versão do simulador, a distância que a bola pode estar do goleiro para que este possa pegá-la foi diminuída, de 2 metros para 1,2 metros. Então, para contornar essa situação, o goleiro agora usa o comando tackle, que pode ser interpretado como um carrinho. Esse comando só pode ser executado com sucesso quando a bola está a até 2 metros do jogador, sendo que quanto mais longe a bola estiver, menor a probabilidade do comando ser bem sucedido. Sabendo-se disso, quando o goleiro percebe que não irá conseguir interceptar a bola normalmente, ele então decide tentar o tackle. Esse comando tem como argumento a direção que a bola deverá ser chutada em relação a direção do corpo do jogador. Quanto maior o ângulo de tackle, menor a força do chute. Com isso, o goleiro decide qual ângulo deve enviar para que a bola passe o mais longe possível do gol.

B. Drible

Quando o jogador recebe a bola, sua primeira decisão é tentar conduzir a bola para a frente. Quando ele percebe que isso não é possível, ou por ter um jogador à sua frente ou por ver que perderá a posse de bola em pouco tempo, ele tenta efetuar o drible para outra direção. O drible é simplesmente a condução da bola para algum ponto perto do jogador, de modo que ele consiga recuperar a posse da bola. Os diferentes tipos de dribles implementados variam no ângulo de condução da bola e na força do chute.

Dentre os ângulos pré-estabelecidos o jogador prevê quanto tempo ele levará para alcançar a bola novamente e quanto tempo o adversário levará para alcançar a bola.

Dentre os tempos que o próprio jogador alcança a bola primeiro que o adversário, ele escolhe aquele em que o adversário demorará mais para alcançar a bola, privilegiando os pontos que fazem com que a bola se aproxime do gol adversário.

Caso o jogador perceba que algum desses pontos é favorável para fazer o drible, executa-o, sendo que o modo de chutar a bola é calculado a fim de que o jogador sempre possua uma pequena distância desta.

A força de chute também é decidida com base nos jogadores adversários próximos à bola. Caso o jogador adversário mais próximo esteja muito distante da bola, é executado um chute mais forte, que apesar de fazer com que o jogador perca mais o contato com a bola, ele consegue conduzi-la mais rapidamente.

C. Passe

Uma maneira mais direta dos agentes trabalharem em equipe é o passe. Por isso um dos algoritmos adicionados ao código é a decisão do passe. Nesse algoritmo o jogador que está com a posse da bola decide qual é o melhor jogador do seu time apto a receber o passe. Nos critérios de decisão destacam-se:

- Distância do jogador amigo até o jogador que está com a bola;
- Existência de adversárias na linha do passe;
- Probabilidade do receptor de fazer o gol.
- Existência de adversários próximos ao jogador que receberá o passe.

O jogador que está com a bola analisa essas informações a cerca de todos os jogadores do seu time e decide qual é o melhor para receber o passe. Caso esse algoritmo não consiga achar ninguém com boas possibilidades de receber a bola, o jogador atual tenta efetuar um lançamento.

D. Lançamento

O lançamento é executado como última alternativa ao jogador que está com a bola. Ele é feito chutando-se a bola para uma determinada direção na qual algum jogador amigo chegará primeiro à bola do que um jogador adversário.

O jogador que está com a bola varre uma determinada faixa de ângulo buscando a melhor direção para o lançamento. Em cada ângulo, o jogador analisa quanto tempo algum jogador amigo levará para chegar na bola e quanto

tempo algum jogador adversário levará para chegar na bola. Ele verifica então se o tempo do jogador adversário é maior que o tempo do jogador amigo. Feito isso para todos os ângulos pré-determinados, o jogador escolhe aquele no qual a folga de tempo de interceptação é maior, ou seja, aquele ângulo no qual a diferença de tempo entre o jogador adversário e o jogador amigo seja maior.

Caso o jogador não ache nenhum jogador amigo que conseguirá interceptar a bola antes do adversário, ele então chuta a bola para pontos pré-determinados, de acordo com a sua localização atual.

E. Posicionamento

Outra forma de interação entre os jogadores é o posicionamento. Com um bom posicionamento, o jogador está mais apto a receber um passe, colaborando assim com o restante do time.

Nessa versão do time os jogadores assumem o seu posicionamento com base no sistema já implementado no UvA_trilearn_2003. Porém, esse posicionamento é ajustado de acordo com as condições do jogo.

Quando o próprio time está com a bola, o jogador se posiciona para receber o passe caso este esteja perto do jogador que está com a bola. O ponto de posicionamento é decidido com base nos jogadores adversários. Caso não exista jogador adversário por perto e na linha do passe, o jogador mantém a sua posição. Caso contrário, o jogador verifica outros três pontos de posicionamento e escolhe o melhor deles para receber o passe. Caso nenhum dos pontos estejam acessíveis para a recepção do passe, o jogador mantém a sua posição, buscando não gastar energia desnecessariamente.

Caso a bola esteja com o time adversário, os jogadores que estão próximos da bola tentam marcar os adversários. Eles ficam na linha entre a bola e o jogador adversário, buscando interceptar um possível passe. Caso o jogador adversário a ser marcado esteja próximo da área defensiva, o jogador que irá marcá-lo se desloca um pouco mais para a defesa, cobrindo tanto um possível passe como um lançamento para dentro da área.

F. Ângulo de visão

No código base o jogador busca olhar para a bola o tempo todo. Percebeu-se que isso não é necessário, além de dificultar certas ações. A visão de cada jogador é delimitada por um cone e quanto mais distante o outro jogador estiver dele mais erros e menos informações sobre o jogador ele terá.

Para contornar esse problema decidiu-se fazer o jogador movimentar mais o seu “pescoço”, seguindo a ordem: bola, direita, bola, esquerda. Ou seja, o jogador não fica mais de um ciclo sem olhar para a bola, assim obtém informações atualizadas sobre ela e consegue ter um alto desempenho de decisão de ação.

O corpo do jogador fica direcionado para a bola e seu pescoço pode girar num ângulo máximo de módulo 90°. Com essa visão mais abrangente do jogo o jogador consegue ter

informações mais atualizadas dos outros jogadores, possibilitando melhores tomadas de decisão, especialmente para quem efetuar o passe.

G. Jogadores Heterogêneos

Para esta versão do time decidiu-se continuar a usar jogadores heterogêneos, tendo-se como base os ótimos resultados alcançados no ano anterior.

Esses jogadores têm algumas características diferentes dos jogadores padrões. Algumas delas são positivas outras negativas. Essas características mudam a cada vez que o simulador é iniciado. São elas:

- Aceleração
- Área de chute;
- Inércia
- Erro no chute
- Recuperação de energia

Na nova versão do simulador tem-se 18 tipos diferentes de jogadores heterogêneos. Criou-se também a limitação de se usar no máximo 1 jogador de cada tipo heterogêneo, ou seja, não pode-se ter dois ou mais jogadores no time do mesmo tipo heterogêneo.

Escolheu-se para o ataque os jogadores que têm o menor erro de chute, buscando-se um maior aproveitamento de chutes ao gol. Para os jogadores laterais escolheu-se aqueles que possuem as maiores acelerações, a fim de que possam carregar a bola mais rapidamente. Para o restante do escolheu-se aqueles que possuem melhores características em relação aos outros, excluindo-se assim as piores opções. Vale ressaltar que, para cada escolha feita, tem-se um valor mínimo para as características em geral do jogador. Por exemplo, se o jogador que possui o melhor chute também for aquele que possui a menor taxa de recuperação de energia, ele será descartado, sendo escolhido o segundo mais rápido.

H. Chute para o Gol

Quando o jogador está no campo adversário, ele avalia a sua probabilidade de gol. Essa probabilidade tem como parâmetros: a distância do jogador até o gol, o ângulo entre a bola e o gol e o posicionamento do goleiro. Cada um desses parâmetros tem um determinado peso e caso a probabilidade total do gol seja maior que 70%, o jogador efetua o chute para o gol.

A direção do chute é calculada prioritariamente com base na posição do goleiro. Decidido o ponto inicial, o jogador verifica se existem zagueiros na linha do chute. Caso afirmativo, o jogador avalia outras direções de chute, até que ache alguma direção que não existam zagueiros. Caso o jogador não encontre outro ponto, chute para o ponto mais distante do goleiro, já que é o goleiro que tem a maior probabilidade de pegar a bola.

I. Intercepção de bola

Após notar-se que muitos times conduzem a bola, percebeu-se a necessidade de criar um algoritmo que fizesse com que os jogadores roubassem a bola do adversário.

A primeira parte do algoritmo é a identificação do jogador mais rápido para a bola. Esse algoritmo é diferente do já existente no código base, que prevê a diminuição gradativa da velocidade da bola ao longo do tempo, como se ela estivesse sido chutada para longe. Quando algum jogador conduz a bola, não é isso que acontece. A bola se movimenta com uma velocidade média constante, fazendo com que o ponto de intercepção esteja mais à frente do calculado quando arbitrou-se uma diminuição gradativa da velocidade da bola.

Caso o jogador perceba que ele é o mais rápido para a bola, dirige-se ao ponto de intercepção calculado. Chegando próximo a esse ponto, o jogador decide se tenta roubar a bola normalmente ou usar o tackle.

O uso do tackle só será necessário caso o jogador perceba que não conseguirá interceptar a bola normalmente ou caso o jogador adversário tente efetuar algum drible no jogador. O tackle também só será chamado se a probabilidade dele ser bem sucedido for maior que 80%.

Decidido que deve dar tackle, o jogador avalia o melhor ângulo. Como o objetivo é parar a jogada do adversário, o jogador tenta primeiro chutar a bola para fora do campo. Caso ele perceba que isso não seja possível, tenta chutar a bola de tal modo que um jogador amigo chegue na bola antes que o jogador adversário.

J. Bola parada

Nessa nova versão do time demos atenção especial para as jogadas de bola parada, como lateral, escanteios e cobranças de faltas.

Quando a cobrança é para o adversário, o nosso time se reposiciona a fim de marcar o maior número possível de adversários e interceptar um possível chute para o gol. Para interceptar o chute para o gol é traçada uma linha ligando a bola ao centro do gol e pega-se o ponto mais perto do gol, gerado pela interseção dessa linha com um círculo de centro na posição da bola. Após escolhido o ponto, mandamos o jogador mais próximo para ele.

Já quando a cobrança é para o nosso time, o jogador mais próximo da bola se posiciona para cobrar a falta e os demais se posicionam para receber o passe. Em especial, um segundo jogador é escolhido para receber a bola prioritariamente em uma posição pré definida. Essa posição também é escolhida usando um círculo e uma linha como no caso de falta adversária. Durante a falta também é tratado problemas como impedimento e marcação adversária para definir para quem tocar.

Quando a cobrança da falta é próxima da área adversária, é tentada uma jogada ensaiada, na qual o jogador que irá receber a bola se posiciona estrategicamente para chutar a bola para o gol.

III. CONCLUSÕES E TRABALHO FUTUROS

Após alguns testes jogando-se com outros times, notou-se uma melhora no desempenho geral. Durante a 2ª SimCUP-ES (Competição Capixaba de Futebol de Robôs Simulação 2D), ocorrida em julho de 2008, o time obteve a segunda

colocação, perdendo apenas 1 jogo. Isso demonstra o crescimento da qualidade do time.

Para o futuro, esperamos implementar técnicas de inteligência artificial, começando por identificação de padrões de estratégias utilizadas por outros times, além de técnicas fuzzy para a tomada de decisões.

REFERÊNCIAS

- [1] PET Engenharia de Computação UFES - www.inf.ufes.br/~pet
- [2] UvA Trilearn 2003 -<http://staff.science.uva.nl/~jellekok/robocup/2003/>