

ARTIGO DE DESCRIÇÃO DA ESTRATÉGIA DA EQUIPE UFSC-TEAM NA LIGA SMALL-SIZE (F-180) DA ROBOCUP

CLAUDIO J. BIAZUS* DÊNIS S. STRASSMANN* MAURO ROISENBERG* GUILHERME BITTENCOURT†
LUCAS C. NEVES† GUSTAVO U. COMERLATTO† MIGUEL Z. AVEIRO† MARCELO UEDA† CONRADO
NUNES†

**Departamento de Informática e Estatística (INE)*
Universidade Federal de Santa Catarina (UFSC) Caixa Postal 476- 88040-900
Florianópolis, SC, Brazil

†*Departamento de Automação de Sistemas (DAS)*
Universidade Federal de Santa Catarina (UFSC) Caixa Postal 476- 88040-900
Florianópolis, SC, Brazil

Email: cbiazus@inf.ufsc.br deniss@inf.ufsc.br mauro@inf.ufsc.br gb@das.ufsc.br
lucascasagranden@hotmail.com gustcomer@gmail.com mzaveiro@yahoo.com.br
ueda@das.ufsc.br conradnunes@gmail.com

Abstract— This article describes the control system implementation of the EyeBot robots applied in the robot soccer team that intends to be used by UFSC-team in the RoboCup small-size (F-180) league. The team is composed by three robots which one of them is the goal-keeper. Each robot is equipped with a local vision system, three proximity sensors and a kick mechanism. The team also count with a global vision system located above the field. The need for real-time responses is one of the most difficult constraints when developing soccer robot systems. In order to satisfy this constraint and achieve a greater reaction power and autonomy, we adopted a distributed control strategy based on the Brooks subsumption architecture for the on-board system and deliberative and strategic behaviors in the central entity. In this work we describe the digital image processing techniques, the structure of the distributed control architecture as well as the preliminary results.

Keywords— Distributed control architecture, Image processing, Robotic soccer.

Resumo— O presente artigo descreve a implementação adotada para o controle dos robôs *EyeBot*, sendo aplicado no time de futebol de robôs, utilizado pela equipe *UFSC-Team*, na liga *Small-Size* (F-180) da *RoboCup*. A equipe é composta por três robôs *EyeBot*, sendo um robô, o goleiro. Cada robô *EyeBot* é equipado com um sistema de visão *on-board*, três sensores e um mecanismo de chute, também contam com um sistema de visão global. Dentre as principais dificuldades encontradas para a construção de sistemas aplicados ao futebol de robôs, está a necessidade de resposta em tempo real. Desta forma, tendo como objetivo oferecer um maior poder de reação e autonomia aos robôs *EyeBot* em uma partida de futebol, adotou-se uma estratégia de controle distribuída, sendo esta baseada na arquitetura de subsunção de Brooks para o sistema *on-board* e com comportamentos deliberativos e estratégico, a cargo da entidade central. Neste trabalho nós descrevemos as técnicas utilizadas no processamento digital de imagens, bem como a estrutura da arquitetura de controle distribuída e os resultados que foram obtidos com o desenvolvimento desta arquitetura relacionada ao futebol de robôs.

Keywords— Arquitetura de controle distribuída, Processamento de imagens, Futebol de robôs.

1 Introdução

O futebol, envolvendo agentes autônomos inteligentes, foi proposto na década de 90, tendo como princípio criar um novo desafio para a Inteligência Artificial (Kim, 1998), (Kitano, 1997) e (Sanderson, 1997). Conforme (Costa, 2000), a adoção deste problema está relacionada ao grande poder de aplicação envolvendo diversas áreas de pesquisa que utilizam o desenvolvimento e o desempenho de algoritmos e arquiteturas diversas.

De acordo com (Botelho, 2003), diversas tecnologias têm sido desenvolvidas e empregadas em sistemas envolvendo o futebol de robôs. Conforme (Schwartz, 2003), estas tecnologias estão fortemente acopladas com o aparato sensório que está disponível em cada robô. No entanto, estes sistemas procuram construir robôs que sejam capazes de jogar futebol de forma totalmente autônoma. Neste caso, os comportamen-

tos e estratégias normalmente têm caráter fortemente reativo e são implementadas diretamente nos próprios robôs, sendo assim a qualquer momento um novo comportamento pode emergir, sem que haja a interferência externa. Por outro lado às equipes que dependem de um computador para enviar mensagens de comandos para seus robôs, tem no “técnico”, um sistema de controle centralizado e ao mesmo tempo de caráter deliberativo (Silva, 2006).

Obviamente, quando consideradas de maneira isolada, cada uma destas 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, que possui um sistema de

visão, responsável pela captura das imagens, pré-processamento destas informações, e envio de comandos através do controle de velocidade para cada robô. Neste modelo os robôs atuam no ambiente de acordo com as informações enviadas por este sistema. Este modelo de arquitetura tem como vantagem a facilidade para reconhecimento do ambiente e definição de estratégias, porém não oferece um poder de expressividade reativa em tempo real.

Para que seja possível atender às restrições relacionadas ao ambiente, um agente deve observar a relação do grau de complexidade deste ambiente, envolvendo sua limitação de comunicação e, ao mesmo tempo oferecendo um poder de expressividade reativa em tempo real (Pio, 2006).

Já em um modelo de arquitetura local ou embarcada, cada robô é suficientemente autônomo ao ponto de poder tomar suas próprias decisões, diante de informações que foram coletadas pelos seus próprios sensores (Shim, 2000). Neste modelo de arquitetura todo o controle, comportamento e ações do agente encontram-se no sistema que está concentrado no próprio robô e, portanto, é bastante eficiente para desviar de obstáculos e condução da bola, porém, sofre com as restrições de capacidade de memória, processamento e visão incompleta do ambiente.

Assim, o objetivo deste trabalho é descrever a metodologia empregada no desenvolvimento da arquitetura de controle distribuída e hierárquica proposta pela equipe UFSC-Team, na liga *Small-Size* (F-180) da *RoboCup*, aplicada ao futebol de robôs. Tem como princípio atender aos requisitos necessários de poder reativo imediato e tomada de decisões estratégicas.

Este artigo está estruturado da seguinte maneira: após esta introdução, na seção 2 são descritas as características do cenário e do *EyeBot*. Na seção 3 a descrição da arquitetura de controle distribuída e hierárquica. Na seção 4, a descrição dos experimentos realizados. Na seção 5, os resultados obtidos e finalmente na seção 6, a conclusão deste trabalho.

2 Características do Cenário e do EyeBot

2.1 Cenário

O cenário utilizado para testes, na liga *Small-Size* (F-180) da *RoboCup*, é composto de um campo plano, de cor verde com dimensões de 4,9m de comprimento x 3,4m de largura (RoboCup, 2007). Para capturar as imagens do ambiente, é utilizado um sistema de visão global e também uma câmera digital colorida acoplada a cada robô *EyeBot*. A bola utilizada é de golfe, de cor laranja.

2.2 EyeBot

O modelo de robô *SoccerBot Plus (EyeBot)*, apresentado na figura 1, é utilizado para pesquisa e destinado a competições de futebol de robôs. Os robôs apresentam 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.

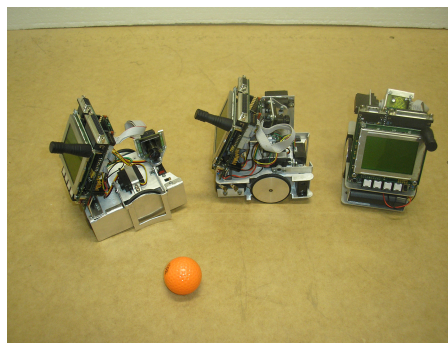


Figura 1: Robô *EyeBot*

Além dos controladores, este modelo de robôs apresenta também dois motores de passo, dois servos motores, dois *encoders* acoplados a cada uma das rodas, três sensores de aproximação infravermelho, bateria recarregável com indicador de nível, câmera digital colorida de 24 *bits* com resolução de 80x60 *pixels*, comunicação via rádio com velocidade 9.600bps; atuando em uma frequência de 433 MHz, protocolo de tolerância à falha e configuração de rede automática, mecanismo de chute posicionado na parte da frente (EyeBot, 2007).

3 Arquitetura de controle distribuída e hierárquica

O mecanismo de percepção-ação que foi implementado utilizando a Arquitetura de *Subsumption* descarta a representação do conhecimento como sendo uma base fundamental dos sistemas artificiais inteligentes (Brooks, 1986).

Em um modelo tradicional de arquitetura para futebol de robôs, todo o processamento das informações é realizado em um computador servidor, sendo que este possui um sistema de visão, responsável pela captura das imagens, pré-processamento destas informações, e envio de comandos através do controle de velocidade para cada robô (Velooso, 1998).

Tanto o modelo de arquitetura tradicional quanto o modelo de arquitetura embarcada apresentam características consideradas essenciais para arquiteturas aplicadas ao futebol de robôs.

Contudo, estes modelos apresentam limitações diante da escolha do modelo adotado. O modelo de arquitetura tradicional apresenta certa deficiência com relação ao poder de expressividade reativa, tais como, evitar obstáculos e na condução da bola. Por outro lado o modelo de arquitetura embarcada possui deficiência com relação à visão completa do ambiente e na definição da estratégia a ser adotada.

Diante de tudo disso, vamos propor um modelo que possa atender às reais necessidades de nosso objetivo que é construir um modelo de arquitetura de controle distribuída e hierárquica que possa ser aplicado ao futebol de robôs. A figura 2 apresenta o modelo da arquitetura distribuída e hierárquica de controle.

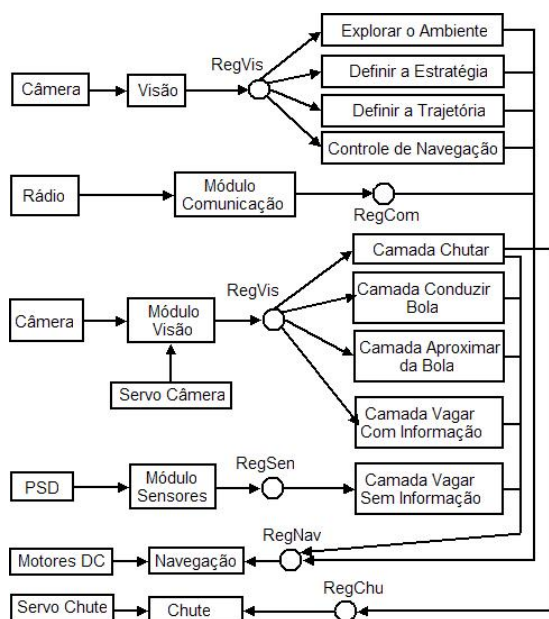


Figura 2: Modelo da arquitetura de controle distribuída e hierárquica

Como podemos observar a arquitetura proposta recebe informações através de uma câmera global processa-as em uma entidade central e envia mensagens de comando utilizando um sistema de comunicação para os atuadores do *EyeBot*. Por outro lado, um conjunto de sensores e uma câmera *on-board* recebem e processam estas informações localmente, no próprio *EyeBot* que imediatamente envia mensagens de comandos para seus atuadores.

3.1 Módulo da entidade central

O sistema que corresponde ao módulo da entidade central é responsável pela exploração do ambiente e definição da estratégia adotada. Cabe ao sistema alocado no servidor 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 mo-

mento em que o sistema de exploração do ambiente, juntamente com o sistema de visão indicar a posição exata em que a bola e os robôs se encontram. É a partir da informação que corresponde à coordenada da bola que o sistema de definição de trajetória escolhe qual é o robô que receberá a mensagem de comando que indica o caminho a ser seguido em direção à bola, ou até mesmo, qual deve ser a posição que o robô deve ocupar em situações onde o robô faz parte da equipe de defesa. Nesse módulo também estão comandos que correspondem ao início (iniciar e reiniciar a partida), comandos de parada (gol, faltas, cobrança de faltas, pênaltis e finalizar a partida).

3.2 Módulo de comunicação

Para que haja troca de informações entre o computador e os *EyeBots*, faz-se necessário um módulo de comunicação. O conjunto de robôs *EyeBots* possui uma unidade de comunicação sem fio para realizar a troca de mensagens entre *EyeBots*. Dessa forma, permite-nos estabelecer a comunicação em uma determinada frequência estabelecida, tanto para recepção das mensagens de comandos, quanto para transmissão de informações entre computador e *EyeBots*.

O próprio sistema operacional do *EyeBot* é quem gerencia este meio de comunicação através de uma rede virtual do tipo *token ring*, sendo que esta apresenta características de tolerância a falhas. A esta rede virtual, que podemos denominá-la de *EyeNet*, permite-nos conectar diversos *EyeBots* com a mesma estação de comunicação.

3.3 Módulo do EyeBot

O módulo do *EyeBot* é composto por um módulo de câmera, Esta câmera possui um servo motor para controle de movimentos, um módulo de sensores, sendo estes sensores posicionados em posições estratégicas para coleta de dados, cinco comportamentos separados em forma de camadas além do módulo dos atuadores, responsáveis pelo deslocamento dos robôs.

Cada comportamento foi montado respeitando a arquitetura de subsunção, sendo que sua posição está de acordo com a importância na arquitetura e ao mesmo tempo a necessidade de fornecer informações para as camadas superiores. Os comportamentos são: vagar sem informação, vagar com informação, aproximar-se da bola, conduzir bola e chutar, conforme podemos observar na figura 2. Os módulos que compõem a arquitetura do *EyeBot* representam uma estrutura hierárquica.

4 Experimentos

A seguir são apresentadas as estratégias adotadas no desenvolvimento desta aplicação utilizando a

ferramenta de desenvolvimento Dev-C++.

A estratégia adotada para a realização deste trabalho teve como iniciativa a divisão dos recursos que compõe este sistema. Esta divisão se faz presente no modelo da arquitetura proposta, ou seja, implementamos um sistema de arquitetura baseada na arquitetura de subsunção de *Brooks* (Brooks, 1986).

O princípio deste mecanismo é baseado na iniciativa de que quando a inteligência é abordada através de um princípio incremental, ou seja, através de camadas adicionais, cada camada deve produzir suas próprias atividades ou comportamentos de forma específica e completa. O presente trabalho se refere a uma atividade completa como sendo uma atividade produzida por uma única camada e sendo esta completa ao mesmo tempo em que é independente das atividades produzidas envolvendo outras camadas.

4.1 O Sistema de Visão

Na liga *Small-Size* (F-180) da *RoboCup* dois sistemas de visão podem ser utilizados: sistema de visão global ou sistema de visão local (RoboCup, 2007). Para o controle dos robôs da equipe *UFSC-Team*, nós utilizamos o sistema de visão local.

O sistema de visão local do *EyeBot* é composto por uma câmera acoplada que está posicionada na parte superior do *EyeBot*, responsável pelo processo de captura das imagens do ambiente. O algoritmo de processamento de imagens desenvolvido além de capturar as imagens é também responsável pelo processamento destas imagens. O algoritmo desenvolvido apresenta a característica de diferenciação de cores. No entanto, é necessário apenas identificar três cores: a bola que possui a cor laranja, e as metas, uma possuindo a cor azul e a outra a cor amarela. A figura 3 apresenta no *display* de LCD do *EyeBot*, a imagem da bola capturada após o processamento.



Figura 3: Imagem da bola no *display*

4.2 O Sistema de sensores

Cada *EyeBot* possui três sensores de distância, sendo estes posicionados à esquerda, à direita e

na parte da frente, para os robôs que atuam na linha. No entanto para o goleiro estão posicionados à esquerda, na parte da frente e na parte de traz. A função principal dos sensores é fornecer informações para a camada vagar sem informação. É através do conjunto de sensores que a camada vagar sem informação atua no ambiente. A utilização deste conjunto de sensores se faz necessária para um maior poder de controle reativo do agente na navegação. Este conjunto de sensores é responsável por detectar obstáculos que possam surgir na dinâmica do jogo. Dentre as mensagens utilizadas no algoritmo desenvolvido está a indicação de que um obstáculo pode estar: à frente, à esquerda ou à direita do robô e, ao mesmo tempo a certa distância. Deste modo se faz necessário separar a informação da existência de um obstáculo em faixas consideradas aceitáveis para a navegação, sendo estas faixas pertencentes a três valores: perto, médio e longe.

4.3 O Sistema de navegação

O sistema de navegação dos robôs *SoccerBot Plus* (*EyeBot*) é composto por dois motores de passo, que são controlados pelo módulo de PWM do próprio microcontrolador *EyeBot*. Cada motor está conectado a um *encoder* que é responsável pela realimentação de velocidade, juntamente com um controlador PI do sistema operacional. Sendo assim é possível realizar o controle da velocidade que é exercida sobre cada motor. No entanto, os robôs *EyeBot* possuem duas rodas com características diferentes, isto é, uma é responsável pelo controle angular e a outra pelo controle linear. Dessa forma, não é possível ter um grau de certeza com relação ao controle de posição dos *EyeBots* confiando unicamente nos dados coletados pelos sensores.

4.4 Camada vagar sem informação

A camada vagar sem informação representa o comportamento do robô de mais baixo nível na arquitetura proposta. Esta camada consiste do comportamento que corresponde a vagar pelo ambiente desviando dos obstáculos que podem surgir, sendo obstáculos as laterais do campo e os robôs.

Esta camada foi desenvolvida com a aplicação de um autômato que corresponde a uma máquina de estados. Dentre os estados possíveis para esta camada nós temos: Vagar, Tem obstáculo, Desviar, Retornar, Girar à direita e Girar à esquerda. O estado corresponde à posição em que o robô se encontra; já a mudança de estados ocorre através da ação a ser executada.

O estado vagar corresponde ao instante em que não existe obstáculo na frente do robô, ou o robô está vendo a bola e desta forma o algoritmo executa a ação navegar para frente. Por outro lado o estado tem obstáculo corresponde ao instante

em que um dos sensores está indicando a presença da parede ou de um robô. Assim, o algoritmo desenvolvido permite que o robô troque de estado podendo desviar deste obstáculo, à ação a ser executada é a ação desviar. Desse modo é possível que o robô gire à direita ou à esquerda. No entanto, para o instante em que todos os sensores estão indicando a presença de obstáculos à ação a ser executada é a ação retornar.

4.5 Camada vagar com informação

A camada vagar com informação inicia sua execução a partir do momento em que o sistema de visão do robô está de posse da informação que corresponde à posição em que a bola se encontra no ambiente. Isto significa dizer que esta camada parte do princípio de que o robô está vendo a bola.

O sistema implementado para a camada vagar com informação foi desenvolvido com o objetivo de que esta camada somente é acionada a partir do momento em que o robô possui a informação da posição em que a bola se encontra. A informação da posição em que a bola se encontra é fornecida através do sistema de visão embarcado. É necessário informar que, para o caso onde o sistema de visão global indicar a posição da bola, a informação da posição é enviada em forma de mensagem de comando diretamente para os atuadores.

A partir do momento em que o robô não está mais de posse da informação que corresponde à posição em que a bola se encontra, a camada vagar com informação pára de ser executada e retorna o controle do robô para a camada vagar sem informação.

4.6 Camada aproximar-se da bola

A camada aproximar-se da bola é executada a partir do momento em que o robô está se aproximando da bola. É esta camada a responsável pelo momento de aproximação do robô com relação à posição em que a bola se encontra. A camada aproximar-se da bola se faz necessária em função da necessidade que envolve o instante de transição do estado em que o robô não está de posse da bola e o instante em que o robô está de posse da bola, ou seja, é necessário que esta transição de estado seja a mais suave possível, visto que, a chegada na bola em um velocidade constante pode resultar em um toque na bola de forma que a mesma seja conduzida para uma posição distante do robô. O autômato da camada aproximar-se da bola possui três estados que são: Longe, Intermediário, Próximo à bola.

O Estado Longe da Bola é executado até que o robô chegue a uma determinada posição considerada próxima a bola, este estado é executado com uma velocidade linear constante, e até o momento em que o robô está se aproximando da bola.

Já o Estado Intermediário à Bola consiste na redução da velocidade linear para que o robô possa se aproximar da bola de forma suave. A partir de certo ponto de aproximação, o estado volta a ser executado com uma velocidade linear constante. Por outro lado o Estado Próximo à Bola consiste em permanecer vagando de forma lenta ou retornar a velocidade linear constante.

4.7 Camada conduzir bola

A camada conduzir bola somente é executada a partir do instante em que o robô está de posse da bola. No entanto, esta camada é responsável pelo controle que o robô exerce na condução da bola, isto é, estando o robô de posse da bola e necessitando conduzi-la até à meta da equipe adversária, esta camada é acionada. O controle exercido por este algoritmo é aplicado nos momentos em que a bola está escapando do domínio do robô, ou seja, quando a bola está em uma das extremidades do mecanismo de chute, que é externo ao robô. Dessa forma, faz-se necessário controlar a bola em direção ao seu objetivo, que é a meta da equipe adversária. O autômato da camada conduzir bola possui três estados que são: Seguir em Frente, Seguir à Direita e Seguir à Esquerda.

O algoritmo que implementa o comportamento conduzir bola consiste apenas em verificar se a posição da bola se encontra no centro do mecanismo de chute. Caso não esteja, ajusta a trajetória do robô para que a bola permaneça no centro do mecanismo de chute.

4.8 Camada chutar

A camada chutar é executada a partir do momento em que o robô está de posse da bola. No entanto, o mecanismo de chute possui três estados possíveis: Abaixado, Passe, Levantado. O primeiro estado Abaixado consiste em permanecer com o mecanismo de chute em uma posição abaixada. Este estado permanece nessa posição enquanto o robô não está de posse da bola, ou quando não se encontra em condições de passar a bola para o seu companheiro de equipe, ou em condições de chutar no gol. Já o estado de Passe consiste em um movimento intermediário, isto é, o mecanismo de chute é acionado através de um movimento considerado intermediário, que resulta em um movimento suave que é responsável por movimentar a bola de forma lenta. Por outro lado, o estado Levantado consiste no movimento de chute. Este estado é executado quando o robô pretende chutar a bola no gol ou quando pretende passar a bola para o seu companheiro de equipe que se encontra em uma posição mais distante.

5 Resultados

A seguir são apresentados os resultados que foram obtidos até o presente momento com o desenvolvimento desta aplicação.

Com relação ao sistema de visão desenvolvido, constatamos que este sistema é consistente em relação à localização da bola. No entanto, o tempo de processamento de uma imagem é de aproximadamente 0.23 de um segundo. Este tempo é atribuído aos recursos de hardware oferecidos pela *EyeBot*. Pois somente o processo de captura da imagem pela câmera do *EyeBot* necessita de um tempo aproximado de 0.2 de um segundo. Dessa forma torna-se difícil melhorar o sistema de visão desenvolvido, visto que a margem de melhoria possível é de 0.03 de um segundo.

O sistema desenvolvido para os sensores dado sua simplicidade, que consiste apenas em indicar a presença de um obstáculo e sua distância, comportou-se de forma eficiente. No entanto, atribuído a característica do conjunto dos sensores *EyeBot* que estamos utilizando, percebemos que em situações onde o robô se choca contra à parede, e sendo este choque fruto de uma colisão imprevista causada por outro robô ou qualquer outro problema que possa surgir, o sensor do *EyeBot* não consegue indicar que o robô está em colisão, ou seja, para uma distância menor de 5 centímetros, o sensor do *EyeBot* indica ausência de obstáculo.

Já para o conjunto que envolve as camadas da arquitetura de subsunção, percebemos que realmente os resultados obtidos refletem o comportamento que foi proposto, mesmo considerando situações onde há certo atraso na máquina de estados desenvolvida. Esta deficiência de atraso está sendo analisada para uma possível correção e ao mesmo tempo para obter um melhor desempenho dos robôs *EyeBots*.

6 Conclusões

Neste artigo, descrevemos e analisamos o desenvolvimento de uma arquitetura de controle distribuída e hierárquica, cuja sua aplicação é voltada para o futebol de robôs, adaptado com um sistema de visão local. E sendo o processamento das informações capturadas realizado no próprio robô.

Através da análise de comparação entre o modelo comportamental proposto para cada robô, e obtido através do comportamento apresentado pelos robôs no conjunto de testes realizados, observa-se que os robôs refletem o comportamento desejado. No entanto ainda se faz necessário realizar novos testes para se obter um melhor desempenho.

Com base em observações realizadas, uma próxima etapa deste trabalho é uma melhor

otimização de cada módulo, cujo objetivo é desenvolver um sistema consistente.

Referências

- Botelho, S. e Araújo, R. e N. R. e T. L. e C. R. P. D. (2003). *FURGBOL Construindo Robôs Autônomos Holonômicos para Jogar Futebol. VI Simpósio Brasileiro de Automação Inteligente. Bauru, Setembro.*
- Brooks, R. (1986). *A Robust Layered Control System for a Mobile Robot, IEEE Journal of Robotics and Automation, p.14-23, March.*
- Costa, H. e Pegoraro, R. (2000). *Construindo Robôs Autônomos para Partidas de Futebol. O Time Guaraná. SBA Controle e Automação v. 11, n. 3, Dezembro.*
- EyeBot (2007). *EyeBot - Online Documentation. (<http://robotics.ee.uwa.edu.au/eyebot/>), Último acesso em Junho.*
- Kim, J. H. (1998). *FIRA Robot World Cup Initiative and Research Directions. Int. J. of Robotics and Automation Systems.*
- Kitano, H. (1997). *RoboCup: A challenge Problem for AI. AI Magazine, v. 18, Spring.*
- Pio, F. V. e Wihibi, N. e. G. E. M. N. e. B. S. S. C. (2006). *Arquitetura FURGBOL aplicada ao SoccerServer. EnRI, III Encontro de Robótica Inteligente.*
- RoboCup (2007). *Robot Soccer World Cup. The rules. (<http://natalnet.dca.ufrn.br/>), Último acesso em Junho.*
- Sanderson, A. (1997). *MicroRobot World Cup Soccer Tournament (MiroSot). IEEE Robotics and Automation Magazine, pg. 15.*
- Schwartz, W. (2003). *Reconhecimento em tempo real de agentes autônomos em futebol de robôs. VI Simpósio Brasileiro de Automação Inteligente - SBAI.*
- Shim, H. S. e Vadakkepat, P. (2000). *A Hybrid Control Structure for Vision Based Soccer Robot System. International Journal Intelligent Automation and Soft Computing.*
- Silva, F. A. e Strassmann, D. S. e. C. J. G. F. e. L. R. G. e. B. G. e. R. M. (2006). *Estratégia para o Controle dos Robôs EyeBot do UFSC-Team: Categoria Small Size do Futebol de Robôs.*
- Veloso, M. (1998). *CMUnited97 RoboCup97 SmallRobot World Champion Team. February.*