

Descrição da Equipe SIRSoccer para a Categoria IEEE Very Small Size

Johnathan Fercher, Oscar Neiva, Hebert Luiz, Lucas Marques, Lucas Borsatto,
Alberto Angonese, Eduardo Krempser
Laboratório de Sistemas Inteligentes e Robótica – SIRLab
Faculdade de Educação Tecnológica do Estado do Rio de Janeiro – FAETERJ/Petrópolis

Abstract—Este artigo descreve o resultado do trabalho feito pela equipe SIRSoccer de futebol de robôs, categoria IEEE Very Small Size do Laboratório de Sistemas Inteligentes e Robótica (SIRLab) da Faculdade de Educação Tecnológica do Estado do Rio de Janeiro (FAETERJ/Petrópolis). Aqui estão descritos os principais componentes dos robôs, suas arquiteturas, o mecanismo de transmissão adotado, assim como a estratégia elaborada para a participação na XII Competição Brasileira de Robótica e na XIII Competição Latino Americana de Robótica. Além disso, descreve brevemente os testes feitos com os protótipos físicos e virtuais, e as avaliações tomadas como base no aperfeiçoamento do projeto.

I. INTRODUÇÃO

A Faculdade de Educação Tecnológica do Estado do Rio de Janeiro - FAETERJ/Petrópolis, localizada no município de Petrópolis - RJ, vinculada à Fundação de Apoio à Escola Técnica do Estado do Rio de Janeiro (FAETEC) possui o curso superior de Tecnologia da Informação e da Comunicação, em que além das disciplinas tecnológicas previstas no currículo, os alunos também participam de eventos e cursos, visando a contribuição na formação profissional e acadêmica. Dessas iniciativas surgiu o Laboratório de Sistemas Inteligentes e Robótica (SIRLab), com a intenção de proporcionar um ambiente para o ensino da computação física, robótica e inteligência computacional.

A equipe SIRSoccer foi formada no ano de 2013, ano em que teve a sua primeira participação em um campeonato, a Competição Brasileira de Robótica - CBR[12]. No ano de 2014 a equipe teve como objetivo aperfeiçoar sua plataforma de futebol de robôs para participar da *IEEE Very Small Size na Latin American Robotic Competition*[10].

A *IEEE Very Small Size*, é uma categoria de futebol de robôs[6] que proporciona o estudo e a aplicação em diversas áreas, como por exemplo: robótica, visão computacional, transmissão de dados e estratégia de controle, que são empregadas no desenvolvimento de sistemas cooperativos que devem atuar em um ambiente altamente dinâmico, utilizando sistemas híbridos de controle.

O sistema computacional utiliza uma câmera fixada a 2m acima do campo dos robôs e da bola no campo. O sistema é executado em um computador localizado fora do campo, o qual envia comandos de arbitragem e posicionamento para os robôs, baseando-se na informação visual, sendo o computador responsável pelo processamento necessário para a coordenação e controle dos robôs. A comunicação

é sem fio e utiliza transmissor/receptor de rádio frequência. O sistema de visão funciona como o receptor de informações do jogo, diferenciando seus elementos (bola, robôs e campo) pelas suas cores. A divisão de estratégia elabora quais serão as decisões tomadas pelos robôs, para as mais diversas situações que podem ser enfrentadas. Após tomada a decisão e realizada a atuação por parte de cada um dos jogadores, outra posição é captada pela câmera, e o ciclo de processo de informações ocorre novamente.

Este trabalho está organizado em 5 outras seções. Na seção 2 e 3 o sistema de visão e o de controle são descritos, respectivamente. E na seção 4 e 5, serão apresentados nesta ordem, o sistema de comunicação e a arquitetura dos agentes. Por fim na 6 o trabalho é concluído.

II. VISÃO COMPUTACIONAL

A visão computacional foi a parte a qual mais se deu atenção, e foi também por onde iniciou-se os trabalhos da construção do time de futebol autônomo. Considera-se este, o artefato mais importante para o funcionamento de todo o sistema SIRSoccer, que precisa estar bastante acurado, por ser o início de toda uma cadeia de processos que também devem estar bem integrados. Para o desenvolvimento do sistema de visão computacional foram estudados outros sistemas, como por exemplo, o RoboCup SSL[13] e o Multi-Cam, porém a equipe preferiu não fazer uso desses sistemas ou adaptá-los, optando por desenvolver seu próprio programa, com base em uma biblioteca de visão computacional.

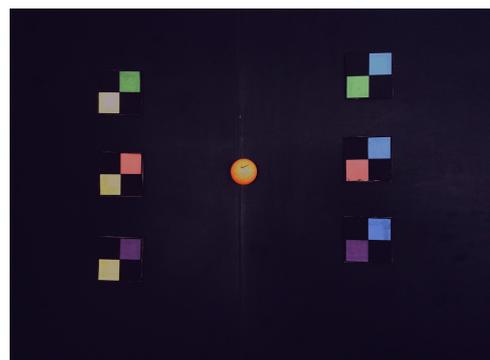


Fig. 1. Padrão de cor SIRSoccer

A equipe utiliza a biblioteca OpenCV[11] e faz uso do padrão de cores RGB para captação e filtragem de imagens. A OpenCV é uma biblioteca livre, sob a licença BSD, incluindo diversos algoritmos de visão. A biblioteca pode ser implementada em Python e C++, porém, dado que todo o restante do sistema criado pela equipe está implementado em C++, optou-se por essa linguagem. O programa localiza na imagem as cores desejadas, e por meio disso faz a distinção dos objetos do jogo, por exemplo, a cor laranja esta destinada a fazer referência ao objeto bola (Figura 1). E para uma melhor captação das imagens são usados alguns filtros da biblioteca.

O sistema obteve avanços em relação ao ano de 2013, uma vez que agora é possível ser feito o reconhecimento de um objeto com área de apenas $5cm^2$, enquanto que com o antigo sistema apenas áreas de no mínimo $20cm^2$ conseguiam ser detectadas. Os novos aprimoramentos foram feitos de forma a captar padrões de cores cada vez menores, e também de forma a melhorar o método de calibragem, agora mais preciso e robusto o suficiente para minimizar as diferenças de variação da luz em todo o campo. Obtem-se agora, em média, uma faixa de erro nas imagens obtidas de um grau para mais ou para menos para a angulação do robô em relação ao eixo x, e de dois pixels em x e em y para o posicionamento.

Quanto ao equipamento usado para captação de imagens, a equipe faz uso de uma câmera Life Cam HD5000 da Microsoft, com retorno de uma quantidade 30 quadros por segundo, porém testes com outras câmeras tem sido feitos na busca de atingir melhores resultados.

III. SISTEMA DE CONTROLE

O trabalho desenvolvido para criação do sistema de controle dos agentes foi dividido em quatro partes. Primeiramente serão apresentadas técnicas usadas para aperfeiçoamento da movimentação, técnicas clássicas como o controle proporcional, integral e derivativo. Para a estratégia de controle, tomou-se como base a metodologia de campos potenciais, em que criam-se campos de atração e repulsão para que o agente diferencie seus oponentes de um ponto objetivo. Também foram usadas técnicas de computação evolutiva para criar uma estratégia onde os parâmetros utilizados nas outras metodologias pudessem ser otimizados. E, por fim, é apresentado um pouco do simulador que está sendo criado pela equipe.

O controle proporcional, integral e derivativo - PID[9] é usado para correção de erros, que são ocasionados por ruídos no sistema. Os ruídos podem ser ocasionados por diversos fatores, seja por interferência eletromagnética entre os componentes, ou até mesmo por desgaste das catracas dos micro motores. O controle PID está dividido em três métodos, o proporcional que é parametrizado por uma constante K_p , o derivativo de constante K_d , e o integrativo com constante K_i , que normalmente vem como último a ser implementado. Dada uma função $u(t)$ e uma função erro $e(t)$, o cálculo do PID é feito da seguinte forma: Primeiramente é feita a estimação do erro, essa função esta presente em todas

as três componentes do método. No método derivativo faz-se a derivada do erro em função do tempo, e no integral o somatório de todas as funções erros do instante zero ao t. E em cada uma das componentes tem-se as constantes proporcional, derivativa e integral.

$$u(t) = k_p \cdot e(t) + k_d \cdot \frac{de(t)}{dt} + k_i \cdot \int_0^t e(t) dt \quad (1)$$

Após as informações do atual estado do agente terem passado pelo algoritmo de controle PID, os valores corrigidos devem ser encaminhados a controladora lógica programável - CLP. É então enviado um vetor de caracteres para a CLP, no qual dois dos cinco caracteres representam os valores de modulação de pulso - PWM[2] a ser enviado aos micro motores.

Antes da implementação da estratégia foi necessário a obtenção de alguns parâmetros básicos do jogo, como algumas distâncias e angulações. Como por exemplo a distância entre a bola e o robô, sendo (X_b, Y_b) as coordenadas da bola e (X_r, Y_r) as coordenadas do robô, tem-se a seguinte equação para a distância d entre esses dois pontos.

$$d = \sqrt{(X_b - X_r)^2 + (Y_b - Y_r)^2} \quad (2)$$

E a angulação Θ do robô dado um certo referencial (X_o, Y_o) no campo, é dada por.

$$\Theta = \tan^{-1} \frac{Y_r - Y_o}{X_r - X_o} \quad (3)$$

Feito isso, como base para a estratégia de controle foi usada a técnica de campos potenciais, e para a implementação baseou-se no que é proposto em[8]. Essa metodologia surge como ideia de se imaginar forças atuando sobre um robô. A técnica é conhecida como Campo Potencial por constituir uma função potencial aplicada aos agentes, seus oponentes e a meta (bola). Os oponentes tem potencial repulsivo e as metas potenciais atrativos, assim seguindo-se a direção do gradiente descendente chega-se a regiões de baixo potencial, longe de obstáculos e cada vez mais próximo da meta.

Além da aplicação de campos potenciais, avaliou-se a aplicação de técnicas da Computação Evolutiva, tais como a Evolução Diferencial[14] para a construção de estratégias mais robustas, especialmente na otimização dos parâmetros envolvidos na aplicação do Campo Potencial aplicado. Essa aplicação conjunta de técnicas permitiu uma busca mais ampla por soluções que caracterizassem a estratégia final adotada. Obviamente que, a ampliação da busca por soluções que definem a estratégia geram a necessidade de avaliação de cada uma das estratégias produzidas, fato que demandaria um exacerbado tempo de implementação e execução dos robôs reais. Portanto, a equipe vem trabalhando num simulador para testar o uso de novas estratégias. O simulador tem sido feito por meio do uso de duas bibliotecas, a Glut[7] e a Bullet[3]. Ambas as bibliotecas são livres, a Bullet detém todo o caráter físico a ser implementado, de colisões até resistência de materiais. A Glut, foi usada para criação de uma

interface gráfica, sendo possível desabilitar a representação para tornar o processo menos custoso.

IV. TRANSMISSÃO DE DADOS

A forma de transmissão de comandos para o robô foi a parte menos alterada de todo o sistema, mantendo-se todos os equipamentos utilizados em 2013. Porém um protocolo de comunicação serial foi criado, de forma a otimizar a recepção de dados feita pelos módulos XBees[5] dos agentes.

Para que seja feita a transmissão dos comandos utiliza-se o micro transmissor XBee, que pode operar enviando ou recebendo dados. Ele trabalha em uma frequência de 2.4GHz e os dados são enviados via socket e broadcast.

O protocolo de comunicação SIR2014, criado durante os trabalhos deste ano, faz com que cada robô reconheça a parte de uma informação referente a si, por meio de um único pacote que é enviado por um *XBee Explorer* (Figura 2). O pacote transmitido é um vetor com três blocos de seis caracteres, que representam uma função específica. O primeiro dos caracteres, representa a id do Robô, os dois seguintes representam a direção de cada roda, seguido de outros dois que apontam o Tamanho do Pulso de Modulação (Pulse-wide Modulation - PWM), e por último um para representar a quantidade de vezes que o comando deve ser executado.



Fig. 2. Explorer usado para enviar as informações

V. ARQUITETURA DOS AGENTES

A arquitetura dos agentes passou por diversas mudanças, desde a primeira participação da equipe na CBR em 2013, até se chegar a uma plataforma mais estabilizada. A elaboração da arquitetura foi feita de forma a minimizar interferências eletromagnéticas entre os componentes. A equipe teve um primeiro contato com Shields da DF Robot[4], placas Arduino Uno[1] e receptor XBee[16], porém com objetivo de se minimizar a arquitetura dos Robôs foram feitos estudos para a criação de uma única placa. O objetivo de se desenvolver uma placa especificamente para o robô vem pela necessidade de ter-se um hardware mais aplicado as questões da equipe na competição (Figura 3).

Analisando os resultados do desempenho da arquitetura usada pela equipe na CBR 2013, pode-se notar diversos pontos da parte física que poderiam ser melhorados, esses que

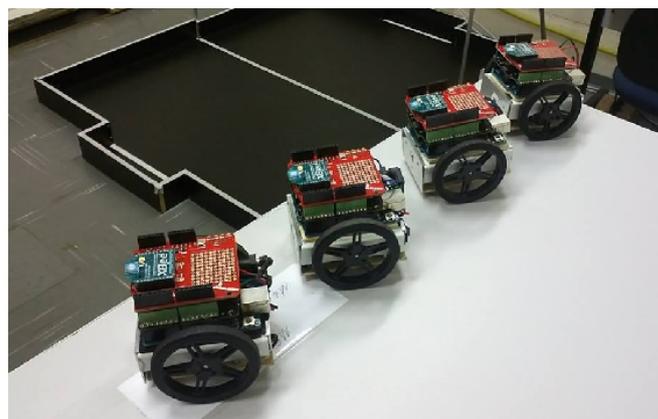


Fig. 3. Robôs da equipe SIRSoccer

serão enumerados a seguir. O primeiro deles seria um melhor aproveitamento do espaço, uma vez que a pinagem dos Shields e da placa controladora Arduino não eram totalmente utilizados. O segundo seria a diminuição da transmissão utilizada nos motores de marca Pololu[12], que apresentaram torque elevado e por consequência baixa velocidade, portanto substituiu-se as caixas de transmissão de 1:75 para 1:30 (Figura 4).



Fig. 4. Motor Pololu 1:30

Após diversos testes e pesquisas feitas sobre o funcionamento de *drivers* e CLPs, chegou-se a um consenso dos dispositivos a serem usados no hardware do robô. Alguns componentes são críticos para o funcionamento, por exemplo, o *driver* de motor a ser usado. No caso da equipe SIRSoccer optou-se pelo uso do Toshiba TB6612FNG[15]. O tamanho pequeno do circuito integrado é uma boa característica, o que sempre levou-se em consideração na escolha das peças, com o objetivo de chegar-se a uma arquitetura mais leve. As outras especificações, também atendem a necessidade do projeto, tanto em corrente máxima e tensão máxima. A comunicação é tratada pelo módulo Xbee, o que simplifica bastante em termos de manter algo já usado na arquitetura anterior. O micro-controlador é o Arduino Nano (Figura 5), e para a escolha deste integrante apenas preocupou-se com o número de entradas e saída que o projeto necessita.

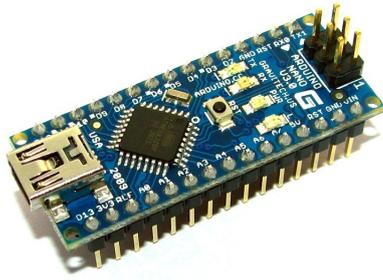


Fig. 5. Arduino Nano

VI. CONCLUSÃO

Cada vez mais a equipe SIRSoccer vem apreendendo conhecimentos nas áreas de pesquisas influentes em problemas de futebol de robôs autônomos, e observado o quanto vasto é esse campo de estudos. Por meio do que foi estudado para a *IEEE Very Small Size*, a equipe adquiriu experiências descritas neste artigo. Muitos avanços foram obtidos desde a participação da equipe na CBR 2013, antigos programas foram aperfeiçoados e novos tiveram que ser criados para compor o novo sistema. Além disso, novos planos são traçados para que no futuro melhores resultados sejam atingidos. Assim, espera-se obter resultados satisfatórios com o novo sistema na referida competição.

REFERENCES

- [1] ARDUINO: Arduino HomePage. Disponível em: "http://www.arduino.cc". acessado em: 20 de maio de 2013.
- [2] ARDUINO: Arduino Learning. <http://arduino.cc/en/Tutorial/PWM>.
- [3] BULLET: Physics Library. <http://bulletphysics.org/wordpress/>.
- [4] DFRobots HomePage. Disponível em: <http://www.dfrobot.com/index.php>. acessado em: 22 de junho de 2013.
- [5] Digi. *XBee/XBee-PRO RF Modules*, 2011.
- [6] E. B. Ferreira Filho. Desenvolvimento de um time de futebol de robôs categoria ieeee very small size. 2013.
- [7] GLUT - The OpenGL Utility Toolkit. <http://www.opengl.org/resources/libraries/glut/>.
- [8] M. A. Goodrich. Introduction potential fields tutorial, 2002.
- [9] W. A. Guerra. Implementação de controle proporcional, integral e derivativo digital em controladores lógico programáveis. 2009.
- [10] LARC:Latin American Robot Competition. <http://cbrobotica.org/>.
- [11] OpenCV. <http://docs.opencv.org/>.
- [12] Pololu: Robotics & Eletronics. Disponível em: <http://www.pololu.com/catalog/product/2361>. acessado em: 22 de junho de 2013.
- [13] Z. Stefan, L. Tim, B. Oliver, W. Mahisorn, and V. Manuela. Ssl-vision: The shared vision system for the robocup small size league. In *RoboCup 2009: Robot Soccer World Cup XIII*, pages 425–436. Springer, 2010.
- [14] R Storn and K.V. Price. Differential evolution – a simple and efficient heuristic for global optimization over continous spaces. *Journal of Global Optimization*, 11:341–359, 1997.
- [15] Toshiba. *Toshiba Bi-CD Integrated Circuit*, 2012.
- [16] ZigBee Alliance. www.zigbee.org.