

# Uma arquitetura de hardware e software para competições em robótica móvel cooperativa

Diego F. de Almeida, Thiago N. A. do Amaral, Luiz R. L. Rodrigues,  
Felipe M. C. Marlière, Ronan A. da Paixão, Erbene de C. M. Júnior,  
Bruno Eduardo Madeira, Paulo F. F. Rosa

**Resumo**—Serão apresentados as principais implementações do projeto nas áreas de eletrônica embarcada e visão, software e mecânica. O controle é feito por um microcontrolador ATMEGA168, programado na linguagem C. O firmware foi implementado para controlar até quatro motores de corrente contínua de 12V e três servomotores. O software é dividido basicamente nos módulos: WorldModel, Inteligência/Execução e Comunicação.

## I. INTRODUÇÃO

Está é a primeira participação do time RobotIME em competições. A maior parte das pessoas envolvidas no projeto são integrantes do Laboratório de Robótica e Inteligência Computacional do Instituto Militar de Engenharia (IME).

## II. ELETRÔNICA EMBARCADA E VISÃO

A eletrônica embarcada das plataformas terrestres é responsável pelo controle dos motores e por toda parte de comunicação entre a estação base e os robôs. O controle é feito por um microcontrolador ATMEGA168 trabalhando a 16MHz, programado na linguagem C, utilizando o ambiente de desenvolvimento AVR Studio 4, ambos fabricados pela empresa Atmel.

O firmware foi implementado para controlar até quatro motores de corrente contínua de 12V e três servomotores. No protótipo mecânico apenas dois motores de corrente contínua são utilizados. Não são utilizados os dispositivos de chute e de dribble. Para a plataforma realizar alguma ação, pulsos PWM são enviados do microcontrolador aos motores, passando por um dispositivo eletrônico intermediário, L298, que possibilita que um par de conjunto de motores de corrente contínua possam ser controlados tanto no sentido horário quanto no sentido anti-horário. Essas ações são transmitidas da estação base via USB-Serial e recebidas por um transceptor RFM12B,

acoplado ao circuito das plataformas, que opera na faixa 433MHz. Além das ações, são recebidos um identificador, que informa para qual robô a ação está sendo transmitida, e o checksum, para verificar a validade do pacote. O software

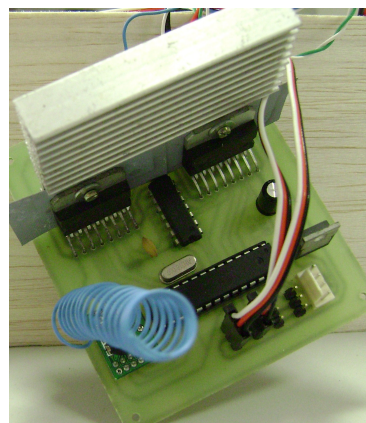


Figura 1: Eletrônica Embarcada - Placa com ATMEGA168.

*SSL-Vision* é compatível apenas com câmeras firewire. Sendo assim, foi utilizada a câmera *FireWire CCD Color Camera DFK 31BF03*. Para captura das imagens da câmera foi utilizada uma *Placa Firewire Pci - IEEE 1394 com 03 Entradas Externas e 01 Interna*. Para conexão entre a câmera e a placa foram utilizados: 1 Cabo Firewire 6x6 pinos com 1,8 metros de comprimento e 1 Cabo 16' Firewire / i.Link Active Extension / Repeater Cable 6x6 pinos com 5 metros de comprimento. Segundo as dimensões do campo (3 metros x 2 metros), onde foram realizados os testes, e a altura da câmera (4 metros) foi utilizada a lente *Vari-Focal Manual Iris Lens 3.5-8MM FS-LV0358* configurada com distância focal próximo de 4.5 mm.



Figura 2: Protótipo do robô.

Diego Felix de Almeida, Seção de Engenharia de Computação  
diegobill@gmail.com

Thiago Navarro Arruda do Amaral, Seção de Engenharia de Computação  
navarro.ime@gmail.com

Luiz Renault Leite Rodrigues, Seção de Engenharia Elétrica  
luiz@luizrenault.com

Felipe Mundy Condé Marlière, Seção de Ensino Básico  
felipemundy@gmail.com

Ronan Alves da Paixão, Seção de Ensino Básico  
ronanpaixao@gmail.com

Erbene de Castro Maia Júnior, Seção de Ensino Básico  
ecmaiaj@hotmail.com

Paulo Fernando Ferreira Rosa, Seção de Engenharia de Computação  
rpaulo@ime.eb.br

Bruno Eduardo Madeira, Seção de Engenharia de Computação  
madeira@ime.eb.br

### III. IMPLEMENTAÇÃO DO SOFTWARE

O software é dividido basicamente nos módulos: World-Model, Inteligência/Execução e Comunicação. Os módulos WorldModel e Comunicação foram implementados na IDE Visual Studio 2008 que permitiu em uma única solução a integração de projetos em diversas linguagens de programação (CSharp, C++ e Basic) tornando o projeto mais flexível para que outros programadores deem continuidade à implementação. O módulo Inteligência/Execução foi implementado na linguagem Java. Os módulos se comunicam entre si via UDP socket o que permite uma maior modularidade ao projeto.

#### A. WorldModel

Com a padronização dos programas *SSL-Vision* e *Referee-Box* para categoria F-180 foi necessária a criação de um software que fosse capaz de se comunicar com esses programas.

Foram implementados dois clientes (um para o *SSL-Vision* e o outro para o *Referee-Box*) que rodam suas rotinas de recebimento de dados em threads separadas sendo assim executadas concorrentemente.

Foram utilizadas as bibliotecas da plataforma .NET para implementação dos clientes. Nesse módulo do software é armazenado o máximo de informações que possam ser extraídas dos dados fornecidos pelo *SSL-Vision* e o *Referee-Box*. Foi criado um banco de dados na memória do computador de modo a armazenar um conjunto de dados (e.g., informações das posições, orientações, identificadores dos robôs) em função do tempo de utilização, principalmente, no módulo de inteligência e na execução.

Foram feitas algumas filtragens dos dados da visão para que se aproximem ao máximo da realidade. As principais filtragens são: remoção de robôs que possuem distância menor que duas vezes o raio menos uma margem de erro; e remoção de robôs que excedem a quantidade de 5 robôs em campo. Essa última filtragem é feita levando em consideração a variável *confidence* que é fornecida pelo *SSL-Vision* e levando em conta também a associação que é feita entre os dados do frame atual e dos anteriores que foram salvos no banco de dados do *WorldModel*.

A variável *confidence* mede o grau de confiabilidade daquele dado. Ela é calculada através de um filtro gaussiano e é formada pelas características dos blobs (padrões de cores), ou seja, é uma medida instantânea, somente válida para aquele frame.

#### B. Inteligência/Execução

Após passarem pelo módulo *WorldModel*, os dados são enviados via UDP Socket para o módulo de inteligência que funciona de forma integrada com o módulo de simulação que é baseado no aplicativo TeamBots. O aplicativo simula a dinâmica e as dimensões de uma partida de robôs da categoria *Small Size Robot League*. Os agentes jogadores são baseados por objetivo e é utilizado o algoritmo minimax para auxiliar a tomada de decisão. A função objetivo leva em consideração inúmeros fatores: distância da bola ao gol

adversário, distância da bola ao gol aliado, distância da bola aos jogadores oponentes, entre outros.

#### C. Comunicação

Após a tomada de decisão pelo módulo de inteligência são enviados via UDP Socket um pacote contendo 10 valores de dezesseis bits que correspondem aos parâmetros de envio de velocidades das rodas esquerda e direita de cada um dos 5 robôs da equipe. Esse parâmetro varia de 0 a 255 sendo que de 0 a 127 o valor da velocidade da roda cresce girando num sentido e de 128 a 255 decresce no outro sentido. A função de envio de comando para o robô possui como argumentos o identificador do robô e os valores de velocidade das rodas esquerda e direita.

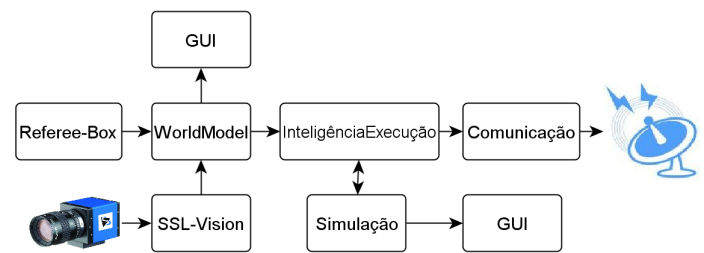


Figura 3: Diagrama em blocos do funcionamento do software.

### IV. O PROJETO MECÂNICO

Foram utilizadas as plataformas *Arduino Powered 2WD Mobile Robot Platform* para os robôs da equipe. Em relação à plataforma que vinha sendo utilizada, esta fornece um maior torque dos motores transmitindo assim uma maior velocidade ao robô. Ela também possibilita que mais dispositivos sejam embarcados, pois possui 170mm de diâmetro e furos passantes para fixação dos dispositivos. A carenagem do robô foi confeccionada com placas de PV (polivinil).

Foram realizados testes experimentais para correspondência das velocidades das rodas em milímetros por segundo em função do parâmetro de envio de velocidade (esse parâmetro é descrito em III-C) para dadas condições do teste. Os testes consistiam em calcular a velocidade do robô através dos dados de captura da visão, dado que uma roda ficava parada enquanto a outra variava de 0 a 255. A partir daí a velocidade da roda em movimento seria o dobro da velocidade do robô, devido à distância da roda parada até a roda em movimento ser o dobro da distância da roda parada até o centro do robô.

O gráfico de um dos experimentos é apresentado na figura 4. Neste gráfico é possível verificar que devido a ruídos da visão, mesmo para parâmetros de velocidade que não eram capazes de movimentar a roda (valores entre 0 e 15 e valores entre 242 e 255) foi obtido valor não nulo para as velocidades. O gráfico também mostra que os valores de velocidade para o motor girando em um sentido não é equivalente no outro sentido, como pode ser visto em função da descontinuidade na mudança do parâmetro de velocidade de 127 (máximo para frente) para 128 (máximo para trás).

Esse experimento teve como objetivo final o levantamento do modelo dinâmico do robô para que ele estivesse casado com o modelo da simulação, apesar de que estes valores ficam susceptíveis às condições do experimento (nível de bateria, superfície de contato, entre outros).

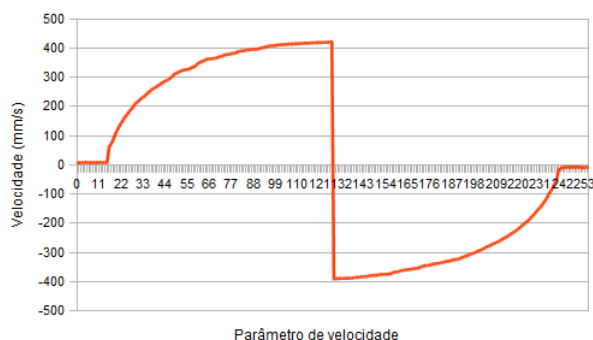


Figura 4: Gráfico da velocidade da roda esquerda versus o parâmetro de envio de velocidade durante um dado experimento.

## V. CONCLUSÕES

O futebol de robôs se mostrou uma excelente plataforma experimental para desenvolvimento de aplicações voltadas as diversas engenharias. Possibilita a aplicação e validação imediata de técnicas de várias áreas do conhecimento humano. Sobretudo, para implantar um ambiente robótico para este fim, diversos são os sistemas fundamentais ao funcionamento de uma equipe de futebol de robôs para atuar especificamente na *Small Size League* da RoboCup, desde a implementação de softwares complexos até a construção física de robôs.

## VI. AGRADECIMENTOS

Esta pesquisa recebeu o apoio parcial do Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação (PIBITI-CNPq/IME).

## REFERÊNCIAS

- [1] A. T. R. da Silva, "Comportamento social cooperativo na realização de tarefas em ambientes dinâmicos e competitivos," Master's thesis, Instituto Militar de Engenharia, Rio de Janeiro, 2006.
- [2] M. A. F. de Sousa, "Uma plataforma para cooperação autônoma de múltiplos robôs," Master's thesis, Instituto Militar de Engenharia, Rio de Janeiro, 2008.
- [3] "Small size robot league," <http://small-size.informatik.uni-bremen.de/start/>, 2010, visitada em 25 de março de 2010.
- [4] "Teambotstm domain: Soccerbots," <http://www.cs.cmu.edu/~trb/TeamBots/Domains/SoccerBots/index.html>, 2010, visitada em 25 de março de 2010.
- [5] "Arduino powered 2wd mobile robot platform," [http://www.seecstudio.com/depot/2wd-arduino-mobile-platform-p-657.html?cPath=119\\_125](http://www.seecstudio.com/depot/2wd-arduino-mobile-platform-p-657.html?cPath=119_125), 2010, visitada em 25 de março de 2010.
- [6] I. M. de Engenharia, "Robotime," <http://www.youtube.com/watch?v=7cPcnFvh8D0>, 2010, visitada em 25 de março de 2010.
- [7] R. Paixão, "Pathfinder," <http://www.youtube.com/watch?v=-cvBo8xL5wE>, 2010, visitada em 25 de março de 2010.
- [8] T. I. Source, "Firewire ccd color camera dfk 31bf03," [http://www.theimagingsource.com/en\\_US/products/cameras/firewire-ccd-color/dfk31bf03/](http://www.theimagingsource.com/en_US/products/cameras/firewire-ccd-color/dfk31bf03/), 2010, visitada em 25 de março de 2010.

- [9] Robocup, "Ssl-vision," <http://code.google.com/p/ssl-vision/>, 2010, visitada em 25 de março de 2010.
- [10] —, "Referee-box," <http://small-size.informatik.uni-bremen.de/referee-start>, 2010, visitada em 25 de março de 2010.
- [11] Atmel, "Atmega168," [http://www.atmel.com/dyn/products/product\\_card.asp?part\\_id=3303](http://www.atmel.com/dyn/products/product_card.asp?part_id=3303), 2010, visitada em 25 de março de 2010.
- [12] S. Zickler, T. Laue, O. Birbach, M. Wongphati, and M. Veloso, "SSL-Vision: The Shared Vision System for the RoboCup Small Size League," *RoboCup 2009: Robot Soccer World Cup XIII*, pp. 425–436, 2009.
- [13] Robocup, "Laws of the robocup small size league 2010," [http://small-size.informatik.uni-bremen.de/\\_media/rules:ssl-rules-2010.pdf](http://small-size.informatik.uni-bremen.de/_media/rules:ssl-rules-2010.pdf), 2010, visitada em 10 de agosto de 2010.