Descrição do Time Carrossel Caipira de Futebol de Robôs da Unesp de Bauru*

Rodrigo César Barboza Rossetti[1] João Vitor Mariano Correia[1] Pedro Henrique Alves de Castro[1]
Gustavo Ribeiro Montes[1] Artur Kyung Min Lee[1] Renê Pegoraro[1] e Mateus Batista[1]

***Resumo*— Este artigo apresenta aspectos de software do time Carrossel Caipira, que representa o Departamento de Computação da UNESP, campus de Bauru, na modalidade IEEE Very Small Size Soccer de futebol de robôs. Adaptações no hardware e nos módulos de visão, estratégia, controle e rádio foram realizadas a fim de possibilitar a participação na modalidade que após dois anos volta a acontecer de forma presencial.**

I. INTRODUÇÃO

O Departamento de Computação da Faculdade de Ciências da UNESP, campus de Bauru, (DCo) participa de competições de futebol de robôs, na modalidade de micro robôs (chamada atualmente IEEE *Very Small Size Soccer*), quando alcançou o vice-campeonato na 1a. Copa Brasil de Futebol de Robôs - CBFR'98, realizada na Escola Politécnica da USP (EPUSP), em abril de 1998. Ainda em 1998, foi criado o time Guaraná resultado da união entre as equipes do time do DCo e do time FUTEPOLI da EPUSP, que também participou da CBFR'98. O time Guaraná obteve o título de vice-campeão mundial no futebol de robôs na categoria MIROSOT (Micro-Robot Soccer Tournament) da FIRA (Federation of International Robot-soccer Association) nas competições da FIRA, realizadas na França em 1998. Apesar de diversos períodos de ausência, o time de futebol de robôs do Departamento de Computação da UNESP de Bauru é conhecido, desde a CBFR'98, como Carrossel Caipira devido sua estratégia de jogo.

O ambiente real do futebol de robôs, nesta categoria, usa uma arena plana horizontal de 150 cm x 130 cm, onde robôs interagem, tentando conduzir a bola até o gol adversário e, ao mesmo tempo, defender o próprio gol de ataques. Neste ambiente, para identificar o estado do jogo, os robôs e a bola são identificados através de visão global usando uma câmera posicionada a 2 m sobre a arena e alinhada ao centro dela. As imagens capturadas por esta câmera são processadas digitalmente obtendo as coordenadas dos robôs e da bola. Com essas coordenadas, uma estratégia é escolhida e transformada em comandos que são enviados aos robôs via rádio. Os robôs recebem esses comandos e realizam as ações correspondentes, alterando o estado do jogo que será capturado novamente pela câmera. A Fig. 1 apresenta uma ilustração do ambiente do futebol de robôs. Durante as disputas, cada time normalmente participa com três integrantes, um "gerente" que coordena o time, um "técnico" que opera o computador interagindo com o software de controle e um "treinador" que manipula os robôs em campo. Porém, neste período de pandemia, onde o contato social deve ser restrito, a permanência de três pessoas por time na área de jogo poderia ter consequências indesejadas.

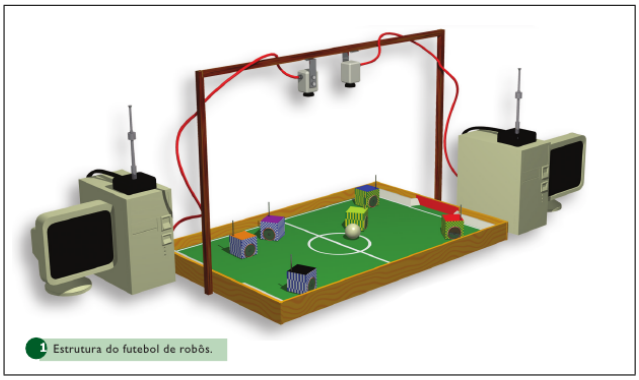

Fig. 1. Ambiente para futebol de robôs.

O futebol robótico abrange diversas áreas do conhecimento. Na construção do robô são aplicados conceitos de mecânica, eletrônica e sistemas embarcados. Do ponto de vista do software, executado no computador pessoal, estão envolvidos elementos de processamento de imagens, inteligência artificial e teoria de controle. Essa abrangência faz desta modalidade de futebol uma ferramenta pedagógica com possíveis aplicações na graduação. Esse projeto busca incentivar e facilitar o desenvolvimento da robótica, para isso o artigo faz uma apresentação das tarefas realizadas, enfatizando a melhoria aplicada recentemente na arquitetura do software como um todo e no novo sistema de controle.

II. SISTEMA DE SOFTWARE DO FUTEBOL DE ROBÔS

Para o robô locomover-se autonomamente o sistema precisa saber "responder" às seguintes questões: "onde está?", "para onde vai?" e "como vai?". Cada módulo do sistema "responde" às perguntas para conseguir a "resposta" e fazer com que o robô entenda o comando final.

O *software* que controla o time pode ser representado através do diagrama apresentado na Fig. 2 que indica as partes principais do processamento. Estas partes são descritas, juntamente com as interações entre elas. A CÂMERA

*O time Carrossel Caipira foi apoiado pela Agência Unesp de Inovação - AUIN

[1]Os autores são do Departamento de Computação, Faculdade de Ciências, Universidade Estadual Paulista "Júlio de Mesquita Filho" - UNESP, Bauru, SP, 17063-170, Brasil. (e-mails: rodrigo.rossetti@unesp.br, mariano.correia@unesp.br, p.castro@unesp.br, gustavo.montes@unesp.br, a.lee@unesp.br, rene.pegoraro@unesp.br, mateusbatistasantos@gmail.com)

captura e digitaliza uma imagem do campo a cada 33ms que é processada pelo módulo VISÃO, calculando, a partir das cores, as poses (x, y, ângulo) dos robôs e da bola em campo. As poses estão em um plano cartesiano retangular, sendo que a abscissa 0 indica o fundo do gol que o time está defendendo Fig. 3. Com estas poses, o módulo ESTRATÉGIA determina as poses de destino onde os robôs do time devem estar para realizar a estratégia definida de defesa e ataque. O módulo CONTROLE, transforma as poses observadas dos robôs pelo módulo VISÃO e das poses de destino determinadas pelo módulo ESTRATÉGIA em comandos de acionamento das rodas a serem enviados aos robôs via rádio, fazendo com que os robôs se movam para realizar a estratégia. Todos estes módulos são executados no computador pessoal.

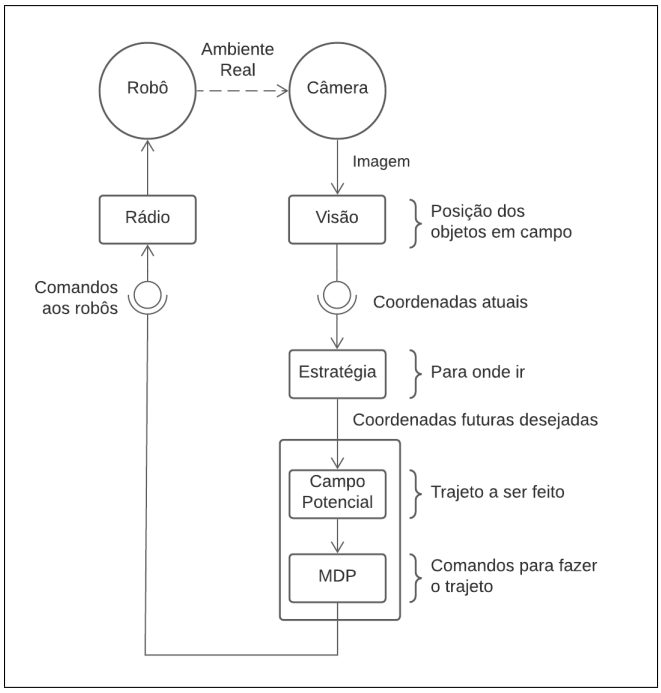

Fig. 2. Diagrama simplificado dos módulos do time Carrossel Caipira no ambiente real.

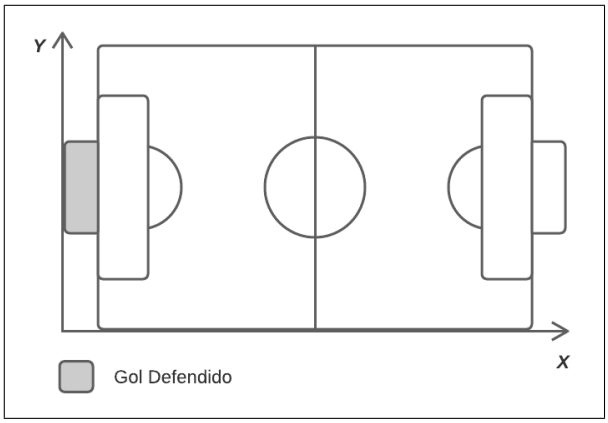

Fig. 3. Sistema de coordenadas globais utilizadas pelo time.

A. A. Módulo da VISÃO

No ambiente de futebol de robôs toda a estratégia e o controle são baseados na interpretação das imagens captadas pela câmera. Para que isso seja realizado, etiquetas de cores em destaque localizadas no topo dos robôs identificam cada um deles, em relação a seu time e possivelmente sua função, conforme demonstra a Fig. 4.

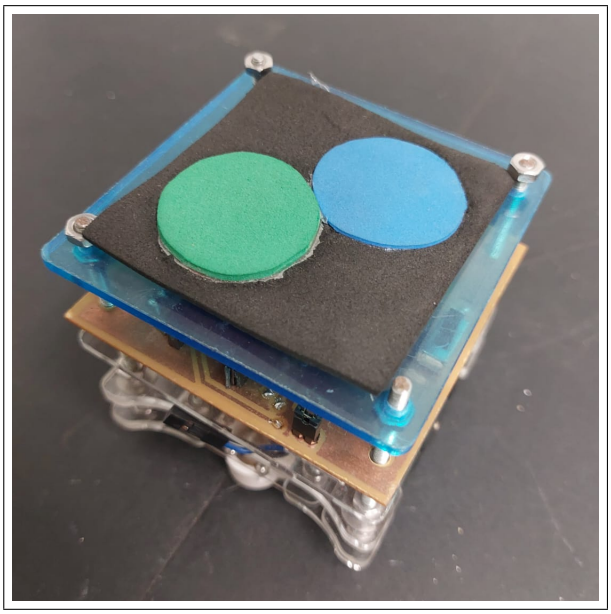

Fig. 4. Etiqueta de identificação do robô. Fonte: Alves et al. [6]

Cada jogador carrega pelo menos uma etiqueta indicando a cor do seu time (azul ou amarelo) e, frequentemente, outra etiqueta de identificação do robô. Nos robôs do Carrossel Caipira, cada etiqueta é formada por duas áreas circulares de 2 cm de raio. Através destas etiquetas o sistema de visão pode localizar os robôs espacialmente sobre o campo.

O tempo de execução do ciclo de controle do sistema foi definido pela taxa de aquisição de imagens. Como o time Carrossel Caipira usa uma taxa de 30 quadros por segundo. Portanto, a cada período de 33 ms, uma nova imagem refletindo o estado atual do campo torna-se disponível ao computador para processamento. Cada uma dessas imagens é capturada e digitalizada na forma de uma matriz com dimensões 640x480 *pixels* com três canais (componentes *Red*, *Green*, *Blue* - RGB). Cada um desses *pixels* deve ser analisado, quanto a sua cor, para identificar se é uma cor de importância ao sistema.

Esta análise é realizada através da verificação de faixas dos ângulos e da intensidade obtidos da transformada dos valores de RGB em coordenadas esféricas (*Spherical Coordinate Transform*, SCT), como descrito em [1]. Esta conversão é importante pois o sistema de cor obtido com a transforma SCT é pouco sensível a variações de intensidades de luz e cores que o RGB. Assim, as cores esperadas para as etiquetas dos robôs e a bola são identificadas e unidas com suas semelhantes. Dessa união, obtém-se um mapa digital dos *pixels* adjacentes de mesma cor na imagem que são chamados

de *blobs*. Para melhorar a robustez, apenas os *blobs* com dimensões aceitáveis (i.e. próximos a 4 cm de diâmetro) e capazes de formar uma etiqueta existente em um robô são considerados. Uma vez que uma etiqueta é identificada como válida, o centro dos dois *blobs*, C_0 e C_1 são utilizados para identificar a posição e a orientação do robô. Em seguida, C_0 e C_1 são convertidos para coordenadas cartesianas, tendo como unidade centímetros.

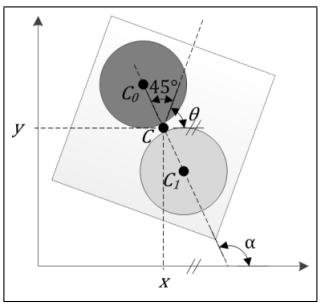

Fig. 5. Posicionamento do robô reconhecido pelo sistema de visão global. Fonte: Alves et al. [6]

Com as coordenadas cartesianas de C_0 e C_1, calcula-se as coordenadas de C, dado pelo ponto médio entre C_0 e C_1, e o ângulo θ do robô, dado por α – 45°, onde α é o ângulo formado entre o eixo das abscissas e o segmento de reta terminado em C_0 e C_1, Fig. 5. Este método é descrito por Alves et al. [6]

B. Módulo ESTRATÉGIA

ESTRATÉGIA é o módulo responsável por definir os objetivos de cada robô. Utilizando as coordenadas atuais da bola e de todos os robôs em campo, este módulo decide qual é a posição mais indicada para cada robô do time, de tal forma que corresponda ao comportamento que os robôs devem desempenhar na partida. Neste ano, o time utiliza uma adaptação dos comportamentos usados em 1998 pelo Carrossel Caipira, quando alcançou o vice-campeonato no CBFR'98, os mesmos comportamentos usados também pelo time Guaraná [3] que recebeu o título de vice-campeão mundial na categoria MIROSOT da FIRA.

O módulo ESTRATÉGIA foi desenvolvido com base no paradigma de programação orientada a autômatos – Os três comportamentos dos robôs: goleiro, volante e atacante são representados por autômatos finitos determinísticos (AFD). Cada estado desses autômatos descreve um objetivo para os robôs, as arestas conectando os estados são as condições de troca de estados, e em cada instante um único estado deve estar ativo.

Estruturalmente, esses três autômatos podem ser considerados "comportamentos" dos robôs, representando super-estados, e a estratégia em si, uma máquina de estados hierárquica que descreve qual comportamento ativo de cada robô, com arestas representando condições de troca de comportamento dos robôs.

Esses AFDs são classes que herdam de uma classe abstrata que representa uma máquina de estados. Utilizando estruturas de seleção (*switches*) e enumerações, os autômatos são implementados proceduralmente através de três métodos herdados:

- inicia() – inicia o autômato, ativando um estado em específico.
- transicao() – verifica as condições das arestas e realiza transições do estado ativo.
- acao() – define o objetivo do robô a ser alcançado no estado ativo.

O códigos das classes desses autômatos são escritos em linguagem C++ e seguem algumas regras que permitem que um programa separado possa ler esse código e gerar representações gráficas dos comportamentos para análise e depuração da estratégia. Estas representações dos autômatos e as descrições dos comportamentos dos robôs são apresentados a seguir.

1) GOLEIRO: Basicamente, o comportamento do goleiro está descrito no diagrama de transições de estados da Fig. 6, o qual consiste de três estados: estado $PosicionaNoCentroDoGol$ - posicionar goleiro no centro do gol sobre a linha defensiva; estado $PosicionaOndeABolaChegaNoGol$ - posicionar goleiro na interseção da direção da bola com a linha defensiva do gol e estado $PosicionaNoXYDaBola$ - posicionar goleiro alinhado à coordenada y_b da bola na linha defensiva. A Fig. 6 representa o comportamento nas diversas situações de jogo, sempre considerando a posição da bola.

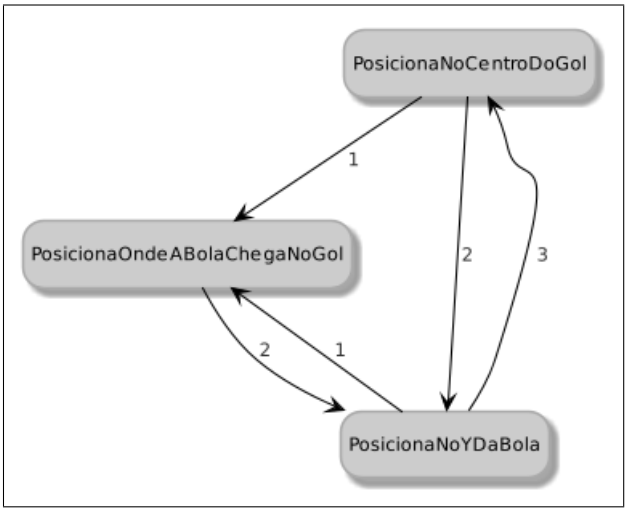

Fig. 6. AFD que descreve o comportamento do goleiro.

2) VOLANTE: Como representado na Fig. 7, o comportamento do volante é descrito por três estados: estado $BloqueiaBola$ - posicionar o robô volante na coordenada y da bola, em uma faixa de defesa distanciada de 30cm a 60cm do gol; estado $BloqueiaChuteAdversario$ - acontece se o robô está a frente da bola, neste caso ele é posicionado entre o robô adversário e a bola, obstruindo um possível chute do adversário; $LiberaCaminhoParaAtacante$ - este estado é sucessor do estado $BloqueiaChuteAdversario$ e ocorre quando o atacante está próximo e atrás da bola, fazendo o volante sair do caminho da bola para permitir que o atacante leve a bola para longe da área de defesa.

Quando o volante e a bola estão em uma situação favorável com respeito ao gol adversário, ocorre uma permuta do volante com o do atacante, simplificando e agilizando uma situação de ataque ao gol adversário. As situações favoráveis acontecem quando o volante é o robô mais próximo à bola e, além disso, ele se encontre alinhado com o gol adversário e a bola.

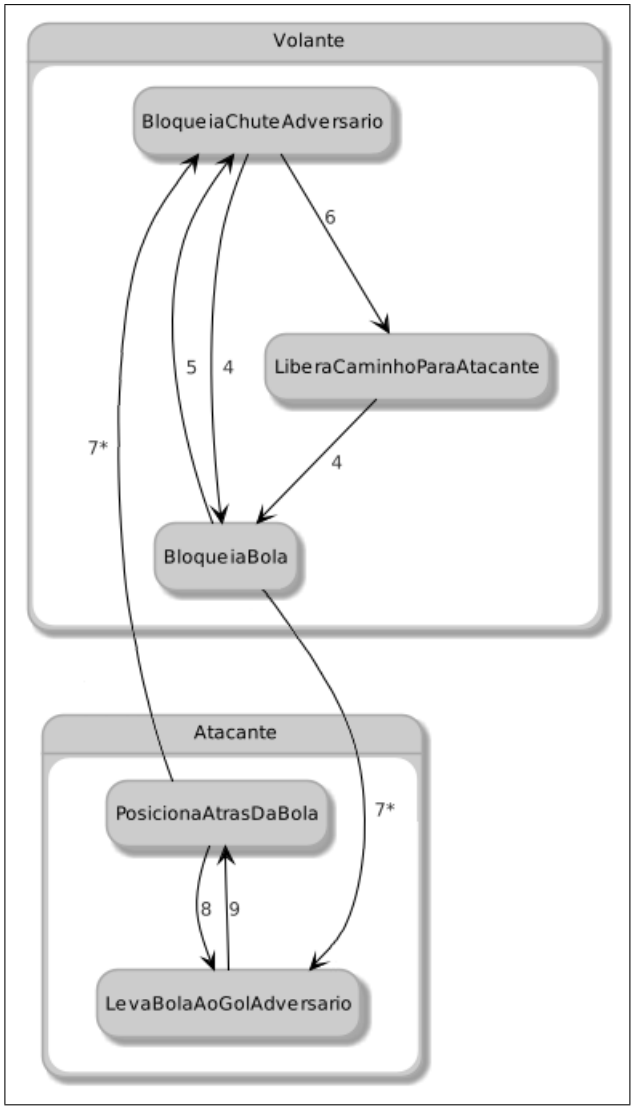

Fig. 7. AFD que descreve o comportamento do volante e do atacante.

3) ATACANTE: O atacante pode estar em dois estados distintos: estado $PosicionaAtrasDaBola$ - modo de posicionamento, neste modo o objetivo do robô é se deslocar para se posicionar atrás da bola, numa posição que permita a condução da bola ao gol adversário ou o bloqueio da bola empurrada por um atacante adversário; estado $LevaBolaAoGolAdversario$ - modo de condução, neste modo o robô se desloca e corrige sua trajetória – alterando o sua posição objetivo para conduzir a bola de forma a alcançar o gol adversário. A Fig. 7 representa o comportamento do robô atacante.

Os eventos que disparam as transições entre os estados, representados por números nas Fig. 6 e Fig. 7, são relacionados com a suas descrições na Tabela I.

TABELA I
CONDIÇÕES DE TROCAS DE ESTADOS UTILIZADAS PELA ESTRATÉGIA.

Condição	Descrição
1	Bola em direção ao gol defendido no intervalo estimado
2	Bola em direção ao gol adversário mas ainda com risco
3	Não há perigo iminente de gol
4	Bola virtualmente posicionada atrás de nosso volante
5	Bola virtualmente posicionada a frente de nosso volante
6	Atacante atrás da bola e do volante
7*	Troca de comportamento entre atacante e volante
8	Bola a frente do atacante e em direção ao gol adversário
9	Atacante virtualmente a frente da bola

C. Módulo CONTROLE

Com as coordenadas detectadas pelo módulo de visão e pelas atribuídas pelos comportamentos do módulo de estratégia, o módulo CONTROLE deve determinar as trajetórias e os melhores comandos a serem enviados aos robôs.

O cálculo de uma trajetória evitando obstáculos utiliza campos potenciais para determinar o melhor caminho. Os campos potenciais partem da ideia de forças imaginárias atuando sobre o robô, ideia proposta por Khatib [4], onde a "força causada" pelos obstáculos é de caráter repulsivo e pela meta, de caráter atrativo. Porém a interferência das "forças" geradas a partir de vários obstáculos podem produzir ótimos locais que atrapalham o desempenho do sistema para encontrar um caminho até a meta. Para evitar esta situação, Connoly et al. [1] solucionaram o problema utilizando funções harmônicas para o cálculo do campo potencial de ambientes nos quais as posições das paredes, objetos e metas sejam conhecidas, que é o caso no de futebol de robôs. As funções harmônicas utilizadas são soluções para a equação de Laplace.

Assim, é definido um Problema de Valor de Contorno na região de atuação do robô utilizando a condição de Dirichlet[1] com potencial alto para obstáculos e potencial baixo para a meta. Então, são extraídas as linhas de força, com base no gradiente descendente [1] [2] do potencial, que direcionam o robô para sua meta, desviando de obstáculos.

Da mesma forma que os robôs reais, os robôs simulados dispões de duas rodas de tração, utilizando o sistema de tração diferencial (*differential drive*). A tração diferencial consiste em ter o acionamento de cada uma das rodas do robô por comandos independentes o que permite manobras. Para manter a compatibilidade com os robôs reais do time, as velocidades enviadas aos robôs no simulador dispõem de 7 velocidades para frente e para trás.

Uma vez obtido o Campo Potencial, tem-se o ângulo ideal que o robô deve atingir para deslocar-se até a meta, então

[1]A condição de contorno de Dirichlet (ou de primeiro tipo) é um tipo de condição de contorno, nomeada em homenagem a Johann Peter Gustav Lejeune Dirichlet (1805-1859). Quando aplicada sobre uma equação diferencial ordinária ou parcial, especifica os valores que uma solução necessita para tomar-se sobre o contorno do domínio.

as velocidades são escolhidas para cada motor de forma a deslocar o robô seguindo este ângulo.

III. O Hardware dos Robôs

O desenvolvimento do robô levou em conta as premissas de baixo custo, disponibilidade de materiais, simplicidade de construção e facilidade na replicação, permitindo que outros robôs possam ser construídos nos mesmos moldes. O robô foi construído com um Arduino Nano [7] como módulo de controle. Este módulo recebe, via um transceptor de rádio NRF24L01, os comandos de velocidade desejados para cada uma das rodas. O Arduino aciona, proporcionalmente, usando modulação por largura de pulso PWM (*Pulse-Width Modulation*), os motores através do módulo TB6612FNG, que por sua vez aciona os motores. A rotação das rodas, e, consequentemente, o deslocamento do robô, são percebidos por sensores TCRT5000, que codificam o movimento de dentes reflexivos fixados às rodas em pulsos enviados ao Arduino. Esses pulsos são contados e servem de realimentação ao acionamento direcionado a cada motor. Um esquema representativo da eletrônica do robô pode ser apreciado na Fig. 8.

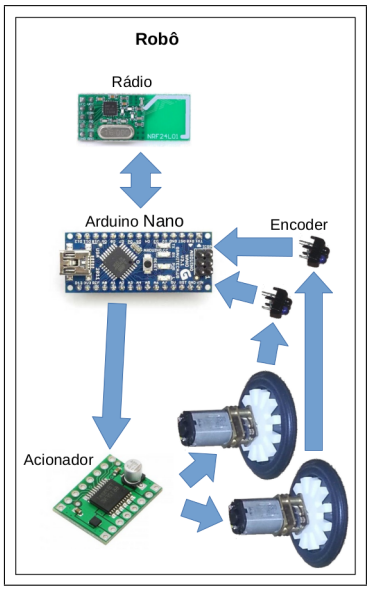

Fig. 8. Esquema dos componentes eletrônicos usados em cada robô.

Do lado do computador pessoal, o envio dos comandos ocorre também através de um circuito controlado por um Arduino Nano conectado a uma interface USB. Este Arduino recebe os comandos a serem enviados aos robôs e os transmite através de um transceptor de rádio NRF24L01 idêntico ao utilizados nos robôs.

References

[1] CONNOLLY, C. I., BURNS, J. B.; WEISS, R. Path planning using laplaces equation. In IEEE International Conference on Robotics and Automation, 1990.

[2] CONNOLLY, C. I.; GRUPEN, R. A. On the application of harmonic functions to robotics. Journal of Robotic Systems, 10:931-946, 1993.

[3] COSTA, A. H. R., PEGORARO, R. Construindo Robôs Autônomos Para Partidasde Futebol: O Time Guaraná. SBA Controle Automação Vol. 11 no. 03 / Set., Out., Nov., Dezembro de 2000.

[4] KHATIB, O.; Command Dynamic dans l'Espace Opérationnel des Robots Manipulateurs en Présence d'Osbtacles. PhD thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse. em Francês. 1980.

[5] WATKINS, J. C. H. Learning from Delayed Rewards. Tese (Doutorado), Universidade de Cambridge, Inglaterra, 1989.

[6] ALVES, S. F. R.; FERASOLI FILHO, H.; PEGORARO, R.; CALDEIRA, M. A. C.; ROSARIO, J. M.; YONEZAWA, W. M.; Proposal of educational environments with mobile robots; Conference on Robotics, Automation and Mechatronics (RAM), 2011.

[7] UMBAUGH, S. E.; Digital Image Processing and Analysis: Human and Computer Vision Applications with CVIPtools, Second Edition, CRC Press,November 19, 2010.

[8] Placa Arduino Nano 3.0. Disponível em: ¡http://arduino.cc/en/Main/ArduinoBoardNano¿ Acesso em 3 de agosto de 2014.