

O Time Carrossel Caipira de Futebol de Robôs

André N. C. Silva³, Danilo W. Nunes¹, Everton K. Francisco², Marcelo Nuñez¹, Mário Bordon⁵, Mateus B. Santos¹, Matheus A. S. Viana³, Rafael T. Takagi¹, Rene Pegoraro⁴, Rodrigo B. R. R. Siqueira¹, Thiago M. Mochetti³

Resumo— Este artigo apresenta aspectos de hardware e software do time Carrossel Caipira, que representa o Departamento de Computação e o Departamento de Engenharia Elétrica da UNESP, campus de Bauru, na modalidade IEEE Very Small Size de futebol de robôs. O Hardware é composto por três robôs que utilizam Arduino para controle dos motores e recebimento de sinais de rádio, que são enviados por um computador pessoal. O Software, executado neste computador, é composto por um conjunto de módulos que inclui: Visão, Estratégia e Controle.

I. INTRODUÇÃO

O Departamento de Computação da Faculdade de Ciências da UNESP, campus de Bauru, participa de competições de futebol de robôs, na modalidade Very Small Size (atualmente IEEE Very Small), desde 1998, com a realização do 1º Campeonato Brasileiro de Futebol de Robôs – CBFR 98. A pesquisa e o desenvolvimento em futebol de robôs mantém o objetivo de incentivar o uso de inovações tecnológicas, no campo da robótica, de baixo custo e com componentes encontrados no mercado nacional. O time de futebol da UNESP de Bauru é conhecido, desde a primeira edição desta competição no Brasil, como Carrossel Caipira devido sua estratégia de jogo. O projeto atual é a sexta versão, de robôs desenvolvidos para futebol de robôs, com aprimoramentos em relação ao time de 2017.

No ambiente do futebol de robôs, nesta categoria, os robôs e a bola são identificados através de uma câmera utilizada como visão global, posicionada a 2m sobre o campo e alinhada ao seu centro, que captura imagens da arena. Estas imagens são processadas digitalmente obtendo as coordenadas dos robôs e da bola. A partir dessas coordenadas, uma estratégia escolhida e transformada em comandos que são enviados aos robôs por rádio. Os robôs recebem estes comandos e realizam as ações correspondentes, modificando a posição dos elementos presentes no ambiente real, que será

capturado novamente pela câmera. A Fig. 1 apresenta uma ilustração do ambiente do futebol de robôs.

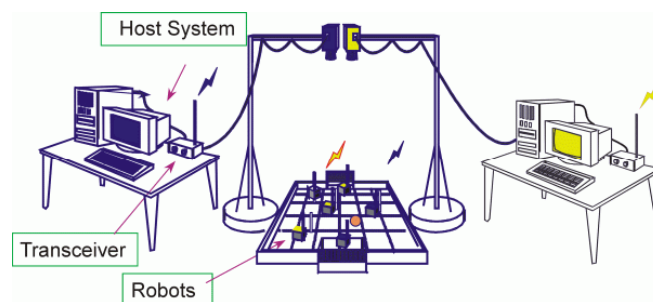


Fig. 1: Ambiente para futebol de robôs.
Fonte: www.mecatronicaatual.com.br/secoes/leitura/950 (2013)

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

O sistema deste time pode ser representado simplificada-mente através do diagrama apresentado na Fig. 2 que indica as partes principais do processamento. Estas partes são descritas, juntamente com as interações entre elas, na sequência. A CÂMERA captura uma imagem do campo, esta imagem é então processada pelo módulo de VISÃO que determinará as posições atuais dos robôs a partir de suas etiquetas coloridas e a da bola que possui cor alaranjada. A PREVISÃO, com base das informações recebidas do módulo da VISÃO, define as posições mais prováveis que os objetos em campo irão assumir alguns instantes a frente. Com estas posições (presente e futura). O módulo de ESTRATÉGIA calcula os locais do campo que os robôs do time controlado deverão se posicionar. Para realizar estes cálculos, este módulo faz uso de roteiros, que são específicos para cada robô. Basicamente, um roteiro é um conjunto de comportamentos específicos para cada robô que faz com que este assumira uma

¹Danilo W. Nunes (e-mail: daniownunes@gmail.com), Marcelo Nuñez (e-mail: marcelo.nunez@hotmail.com), Mateus B. Santos (e-mail: mateusbatistasantos@gmail.com), Rafael T. Takagi (e-mail: rafaelttakagi@gmail.com) e Rodrigo B. R. R. Siqueira (e-mail: rodrigo.buenorrs@gmail.com) são alunos de Bacharelado em Ciência da Computação.

²Everton Kevin Francisco (e-mail: everton_kelvin@hotmail.com) é aluno de Bacharelado em Sistemas de Informação.

³André N. C. Silva (e-mail: neves.andre27@gmail.com), Matheus A. S. Viana (e-mail: mathvna@gmail.com) e Thiago M. Mochetti (e-mail: thiagomochetti@gmail.com) são alunos de Engenharia Elétrica.

⁴Rene Pegoraro (e-mail: pegoraro@fc.unesp.br) é professor do Departamento de Computação.

⁵Mário Bordon (e-mail: mebordon@feb.unesp.br) é professor do Departamento de Engenharia Elétrica.

postura defensiva ou ofensiva durante uma partida. O módulo de CONTROLE, fazendo uso das posições atuais de cada robô (VISÃO) e de seus objetivos (ESTRATÉGIA), determina a maior velocidade possível que um robô pode assumir para que este consiga chegar ao seu objetivo e parar, definindo os valores de cada roda a serem enviados aos robôs via rádio, fazendo os robôs se moverem para concluir a estratégia. Todos estes módulos são executados no computador pessoal.

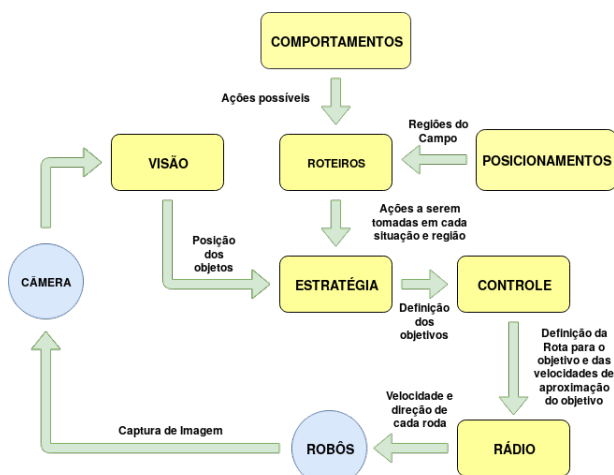


Fig. 2: Diagrama simplificado dos módulos do time Carrossel Caipira.

A. Módulo da Visão

No ambiente de futebol de robôs toda a estratégia e o controle, tanto de baixo nível quanto de alto nível, 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. 3.

O tempo de execução do ciclo de controle do sistema foi definido pela taxa de aquisição de imagens. Como o time Carrossel Caipira usa câmeras de vídeo convencionais, a taxa é limitada a 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 por uma placa de captura, que disponibiliza, 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 técnica é chamada de segmentação de cor.

B. Módulo de Estratégia

A estratégia é o módulo responsável por definir a meta de cada robô. Utilizando as coordenadas da bola, dos robôs do time e dos adversários este módulo decide qual é a posição mais indicada para cada robô do time através dos roteiros. Os roteiros são basicamente conjuntos de regras que ditam o comportamento do robô em cada situação. Com as posições pré-determinadas no módulo de posicionamentos e com os



Fig. 3: Etiqueta de identificação do robô

comportamentos padronizados no módulo de comportamentos, os roteiros são montados pensando nas situações de jogo. A vantagem deste modelo em relação aos antigos é que através da padronização dos elementos mais básicos da estratégia através destes dois novos módulos é possível montar facilmente novos roteiros, o que permite que o time possua diversas posturas às mais diversas situações de jogo, sendo mais ofensivo quando estiver atrás no placar ou mais defensivo quando se deseja tentar manter o resultado, por exemplo.

C. Módulo de Controle

A partir das coordenadas detectadas pelo módulo de Visão e das coordenadas atribuídas pelos resultados do módulo de Estratégia, o módulo de Controle deve determinar as melhores trajetórias e comandos a serem enviados aos robôs. O cálculo de uma trajetória deve levar em conta o desvio de obstáculos, evitando os outros robôs na arena. Este cálculo é necessário para levar o robô da sua posição atual até a determinada pela estratégia. No caso do time Carrossel Caipira, é empregado um método chamado de campo potencial. Os campos potenciais partem da ideia de forças imaginárias atuando sobre o robô, ideia proposta por Khatib [4], na qual 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 locais ótimos que atrapalham o desempenho do sistema para encontrar um caminho até a meta para o robô.

Para evitar esta situação, Connolly et al. [5] 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 do ambiente de futebol de robôs. As funções harmônicas utilizadas são soluções para a equação de Laplace (1).

$$\nabla = 0 \text{ para } P : R \rightarrow R. \quad (1)$$

Assim é definido um Problema de Valor de Contorno na

região de atuação do robô utilizando a condição de Dirichlet¹ com potencial alto para obstáculos e potencial baixo para a meta. Então, são extraídas as linhas de força, com base no gradiente descendente [5][6] do potencial, que direcionam o robô para sua meta, desviando-o de obstáculos.

Uma vez obtido o campo potencial, tem-se o ângulo ideal que o robô deve atingir para se deslocar até a meta, que chamamos de ângulo objetivo, basta agora calcular a direção e velocidade de cada motor.

D. Interface

Para comportar toda essas mudanças feitas em diversos módulos da equipe foi também desenvolvida uma nova interface, tendo em mente praticidade e as necessidades de comportar novas funcionalidades. A nova interface foi feita utilizando abas para as principais seções, entre a principal, configurações, outros e ajuda, como pode ser observado na Fig. 4.

Na aba principal o foco foi deixar de fácil acesso as funções indispensáveis durante o jogo, entre uma tela para mostrar o q está sendo captado pela câmera, uma tabela para logs e botões para iniciar o software além de situação em que o jogo está, como jogo normal, disputa de bola, w.o., entre outros.

Na aba de configurações se encontram as principais configurações indispensáveis para setup do software, entre escolha dos roteiros, calibração da visão entre outras.

Na aba outros, foram colocadas configurações que geralmente não são usadas, porém podem vir a ser necessárias em situações inesperadas, entre uma opção para gravação em buffer de alguns frames de jogo, uma opção para capturar uma imagem do campo vazio, entre outras.

Finalmente a aba ajuda tem por finalidade explicar as mais diversas funcionalidades do programa, através de uma estrutura separada em seções, pessoas que não conhecem o software podem aprender sobre como utilizá-lo.

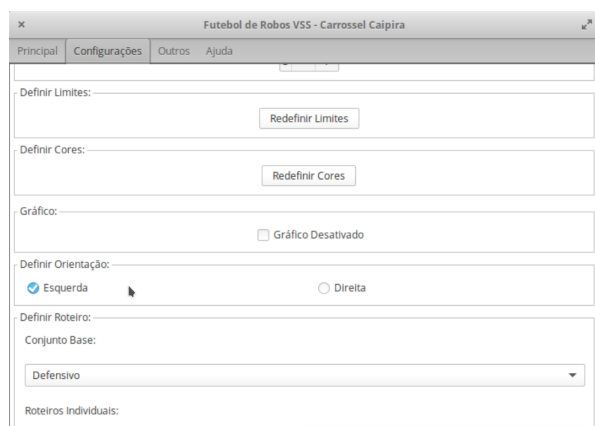


Fig. 4: Nova interface na aba de configurações.

¹A 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

III. SIMULADOR E TESTES

O desenvolvimento e os testes utilizaram o simulador no time da UNESP-Bauru previamente existente. O simulador executa o módulo de estratégia e o de controle, oriundos do software executado para o ambiente real, sem alterações nos códigos. Apenas os arquivos de códigos dos dois módulos (estratégia e controle) da pasta de fontes destinado ao ambiente real precisam ser transportados para a pasta do simulador, nenhuma outra alteração precisa ser realizada. Apesar da dinâmica ser pouco considerada neste simulador, ele simplifica a realização de testes dos algoritmos em desenvolvimento, sem a necessidade da montagem do ambiente real. Uma imagem do simulador em uma situação de jogo pode ser na vista na Fig. 5. O simulador pode também apresentar o campo potencial gerado pelo módulo de controle, Fig. 6.

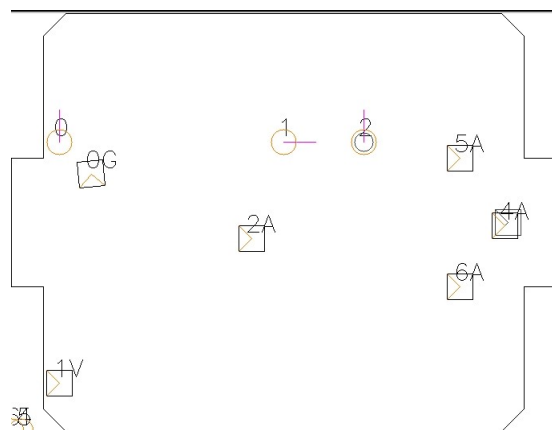


Fig. 5: Situação representada pelo simulador.

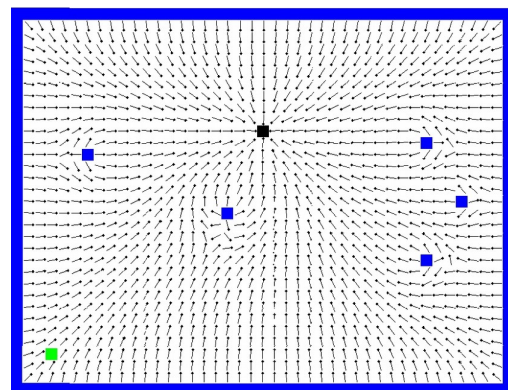


Fig. 6: Representação do campo potencial para a situação apresentada na Fig. 5 considerando o robô 1V (volante). Nos quais os quadrados preto, verde e azul representam respectivamente meta, robô e obstáculo.

Para que os teste pudessem gerar análises convincentes em relação ao desenvolvimento da estratégia e do controle, foi necessário acrescentar uma nova funcionalidade que permitisse dois times jogarem entre si. Para isso foi desenvolvido no código do simulador a possibilidade de se comunicar em rede, de tal forma que houvessem dois times clientes, cada um com sua estratégia e controle, se comunicando com o servidor do simulador, que efetiva os comandos enviados por

cada cliente, realizando a movimentação dos robôs e da bola virtualmente, e fazendo o papel da visão ao fornecer o estado de cada robô para os times clientes. Para a comunicação entre os clientes e o servidor optou-se por um protocolo de comunicação simples, o User Datagram Protocol (UDP).

Desta forma mudou-se a forma como o simulador funciona. Na Fig. 7 é apresentado o esquema, de forma simplificada, do simulador antigo, em que a estratégia recebe o estado do robô, calcula o objetivo e envia para o controle. O controle vai calcular a trajetória e o comando que será enviado para cada roda, que será enviado para o simulador que efetua a movimentação.

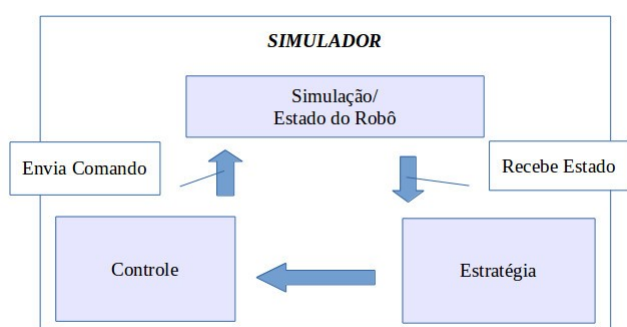


Fig. 7: Simulador antes das alterações.

Na Fig. 8 é apresentado o esquema do simulador atual, no qual o simulador torna-se um servidor, que recebe os comandos de cada roda, efetua-os e em seguida envia o estado do robô para os clientes. Os clientes recebem o estado, calculam o objetivo através da estratégia e os comandos a ser enviados para o simulador através do controle, na sequência esse comando é enviado para o servidor que efetuará a movimentação dos robôs. Além disso, tem-se a representação para a conexão de um cliente, mas a representação é a mesma para dois clientes, que é o caso do futebol de robôs.

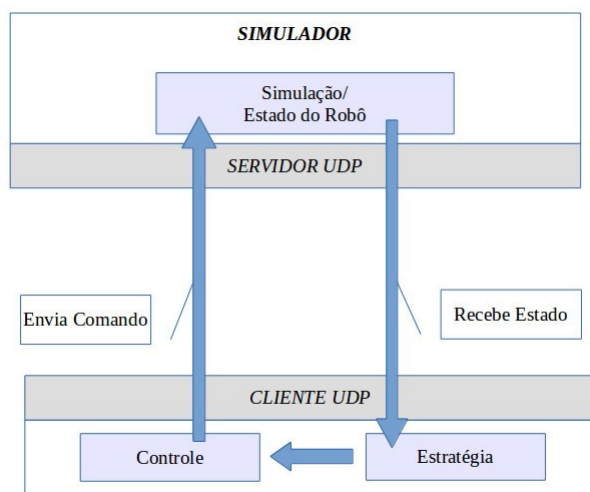


Fig. 8: Simulador após alterações, representando uma conexão.

Assim, o novo simulador permite interação entre dois

times virtuais, levando em consideração a dinâmica, cinemática e demais constantes físicas referentes ao robô do time Carrossel Caipira, possibilitando a comparação através de estatísticas que podem ser coletadas durante a execução do programa, na figura 9 temos a imagem do novo simulador.

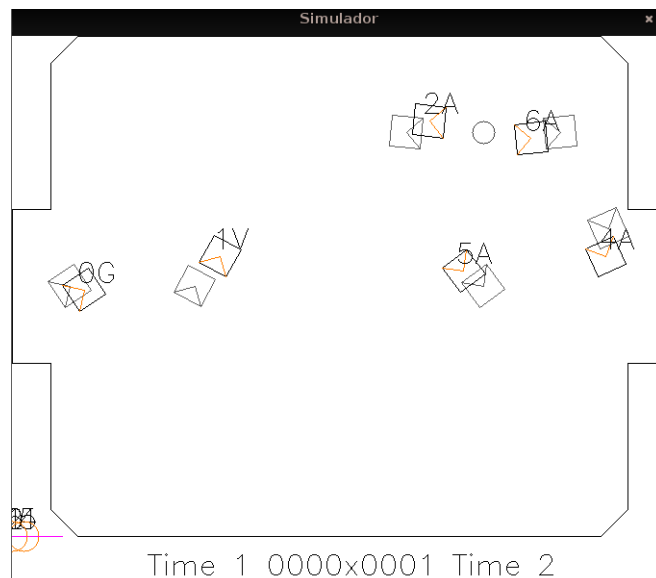


Fig. 9: Simulador após alterações, representando uma conexão.

Nesta versão, o simulador procura entender o comportamento da estratégia e controle estudados, através de situações de jogo como: placar total, gols contra, pênaltis usufruído pelo time Carrossel Caipira para o desenvolvimento das pesquisas envolvendo futebol de cometidos, pênaltis convertidos, gols de cada robô, número de free balls e tempo de jogo. Além disso, é possível verificar a evolução do placar no decorrer da simulação e avaliar o intervalo de confiança para estimar a margem de erro dos dados levantados. A análise quantitativa dos dados estimula a análise com dados reais feita no ambiente controlado do laboratório que complementa o ambiente instrumental robôs.

IV. O HARDWARE DO ROBÔ

Após o contato com outras equipes e pesquisa em relação aos materiais, decidiu-se por reformular a estrutura dos robôs. Visando a simplificação das montagens, bem como agilidade na manutenção, os robôs adquiriram um escopo modular. Dessa forma, cada conjunto de componentes responsável por um setor do robô é acoplado a um dos módulos separadamente. Assim, caso algum dos módulos apresente problemas de execução, ele pode ser trocado por outro igual reserva, sem que haja excessiva perda de tempo. Inicialmente, cogitou-se a existência de três módulos: os motores junto das rodas; a bateria; e a placa controladora com os componentes discretos. O módulo da placa controladora, que contém, entre outros componentes, o Arduino Nano[6], é agora livre de fios e possui componentes soldados diretamente ao fenolite. Novamente em busca de agilidade na manutenção, somente soquetes e caixas de pinos foram soldadas diretamente na placa, de modo que os componentes

são removíveis e de fácil substituição. As baterias, que são empregadas agora no lugar das pilhas AA de versões anteriores, ocupam sozinhas, junto de um pequeno circuito de verificação de carga, o segundo módulo. Isso ocorre devido ao trabalho necessário para carregá-las. Diferentemente das pilhas, onde bastava uma fonte de alimentação para recarregar o robô, a bateria de lítio exige um circuito próprio externo ao robô para carregar adequadamente. O circuito verificador de carga é responsável por impedir que a bateria chegue a níveis muito baixos, o que prejudicaria seu desempenho. O último dos módulos é composto pelos motores e rodas. A estrutura do robô para esse ano, além de modular, foi substituída por chapas paralelas de madeira. O acrílico previamente empregado mostrou-se dispendioso enquanto financeiro e as demais cotações, incluindo o uso de impressora 3D, apontaram para a madeira como principal opção.

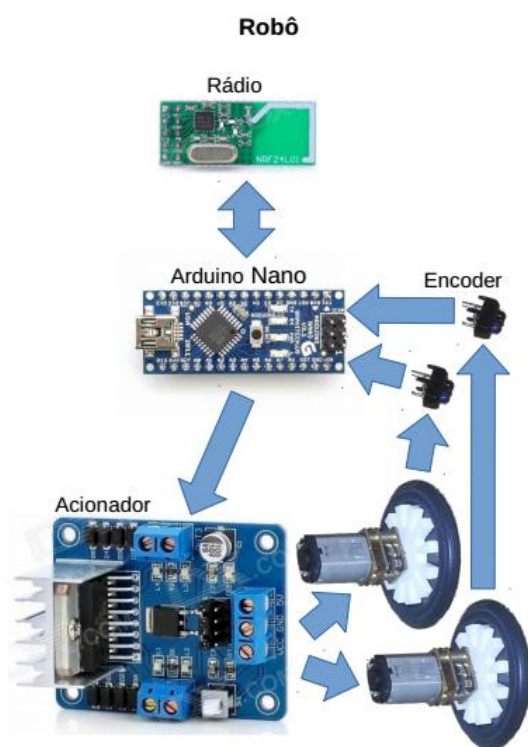


Fig. 10: Esquema dos componentes eletrônicos usados em cada robô.

O robô construído tem um Arduino Nano[6] 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 os motores utilizando modulação por largura de pulso PWM (Pulse-Width Modulation) através de uma placa equipada com o componente por um circuito impresso L293.

A rotação das rodas, e consequentemente o deslocamento do robô, são determinados 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. 10 .

Do lado do computador pessoal, o envio dos comandos também ocorre através de um circuito controlado por um Arduino Nano[6] conectado a uma interface USB. Este Arduino recebe os comandos a ser enviados aos robôs e os transmite por meio de um transceptor de rádio NRF24L01 idêntico ao utilizados nos robôs.

REFERÊNCIAS

- [1] UMBAUGH, S. E.; Digital Image Processing and Analysis: Human and Computer Vision Applications with CVIptools, Second Edition, CRC Press, November 19, 2010
- [2] 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.
- [3] COSTA, A. H. R. ; PEGORARO, R. . Construindo robôs autônomos para partidas de futebol: o time GUARANÁ. Controle & Automação, Campinas, SP, v. 11, n. 2, p. 141-149, 2000.
- [4] KHATIB, O.; Command Dynamic dans l'Espace Opérationnel des Robots Manipulateurs en Présence d'Obstacles. PhD thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse. em Francês. 1980.
- [5] CONNOLLY, C. I., BURNS, J. B.; WEISS, R. Path planning using laplaces equation. In IEEE International Conference on Robotics and Automation, 1990.
- [6] CONNOLLY, C. I.; GRUPEN, R. A. On the application of harmonic functions to robotics. Journal of Robotic Systems, 10:931-946, 1993. Placa Arduino Nano 3.0. Disponível em: <<http://arduino.cc/en/Main/ArduinoBoardNano>> Acesso em 3 de agosto de 2014.
- [7] SANTOS, Rafael, Mecanismo de Previsão de Posição para o Time de Futebol de Robôs da UNESP, Trabalho de Conclusão de Curso do Curso, UNESP - Universidade Estadual Paulista "Júlio de Mesquita Filho", Bauru, Brasil, Fev. 2017.