

CONTROLE DE ROBÔ JOGADOR DOTADO DE SENSORES

LUÍS HENRIQUE CLAUDINO SILVA*, LARISSA FERNANDES MARÃES*, FÁBIO IAIONE*, IGOR SANTO ANDREA VISIOLI*, ANDRÉA TERESA RICCIO BARBOSA*, EDSON TAKASHI MATSUBARA*

**Laboratório de Inteligência Artificial - Faculdade de Computação
Universidade Federal do Mato Grosso do Sul
Campo Grande, Mato Grosso do Sul, Brasil*

lhcs10@gmail.com, larissa.maraes@gmail.com, iaione@facom.ufms.br,
igor@santoandrea.com.br, aricciobarbosa@yahoo.com.br, edsontm@facom.ufms.br

Abstract— Robot soccer competition is an activity used to promote education and research on robotics. This activity is relatively complex and involves knowledge and technology from various areas. Among the categories, there is the IEEE Very Small Size Soccer, whose robots execute only commands received by radio and, therefore, the control is concentrated on the computer. This approach requires a camera and computer with high-performance. Otherwise, there is an input lag between image capture and the decision making during a movement. This paper suggested that the use of a magnetic compass to determine robot's direction, which data could be used for the implementation of a discrete PID controller to perform circular movements, walk straight and rotate on the axis itself. Many tests with the ball located in different positions on the field, with the robot positioned in different directions. The results obtained were satisfactory and promising for use in a future team. It is believed that this work contributed to improvements in the VSSS category robots, which have the same construction and control for many years.

Keywords— Robotic, Control Theory and Applications, Robots Soccer.

Resumo— A competição de futebol de robôs é uma atividade utilizada para promover a educação e pesquisa em robótica. Esta atividade é relativamente complexa e envolve conhecimentos e tecnologias em várias áreas. Entre as categorias, há o IEEE Very Small Size Soccer, cujos robôs executam apenas os comandos recebidos pelo rádio e, portanto, todo o controle está concentrado no computador. Esta abordagem requer que a câmera de vídeo e o computador tenham um alto desempenho. Caso contrário, há um atraso entre a captura de imagem e algumas decisões durante seu movimento. Este trabalho propôs-se o uso de bússola magnética para determinar a direção do robô, cujos dados foram utilizados na implementação de um controlador PID discreto para realização de movimentos circulares, andar em linha reta e rotacionar no próprio eixo. Foram realizados vários testes com a bola em diferentes posições do campo, com o robô posicionado em sentidos diferentes. Os resultados obtidos foram satisfatórios e promissores para utilização em um time futuro. Acredita-se que este trabalho deu uma contribuição para melhorar os robôs usados na categoria VSSS, que têm a mesma construção e controle há vários anos.

Palavras-chave— Robótica, Teoria de Controle e Aplicações, Futebol de Robôs.

1 Introdução

As competições de futebol de robôs são realizadas para o desenvolvimento de pesquisas e educação, além de promover conhecimento em diversas áreas envolvidas, tais como: Informática, mecânica, eletrônica, teorias de otimização, entre outros. Desta forma, através da resolução de problemas no futebol de robôs, várias teorias, algoritmos e arquiteturas, além de diversas tecnologias, podem ser integrados e analisados.

Com esse objetivo, a Competição Latina Americana de Robótica (LARC) é realizada a cada dois anos e cada edição é sediada em um país da América Latina. Nota-se que esta é uma competição acadêmica composta por diversas categorias, sendo uma delas a categoria de futebol de robôs denominada IEEE Very Small Soccer (IEEE, 2014).

A Figura 1 apresenta o sistema geral do campo pronto para uma partida. Em cima do campo existe uma câmera de cada time, que coletam as imagens do jogo. Essas imagens são enviadas a um computador que identifica a posição

dos robôs (inclusive a dos robôs adversários) e da bola (cor laranja). As posições dos robôs e da bola são enviadas a um programa computacional de estratégia que decide qual movimento cada um dos robôs deve executar. Os robôs de cada time são controlados, portanto, por um computador totalmente autônomo, não podendo ter intervenção humana em hipótese alguma. A estratégia é transmitida aos robôs por um sistema de transmissão de radiofrequência. Os robôs são formados por um sistema receptor que converte as ondas de rádio em sinais elétricos para os motores, que movimentam os robôs (IEEE, 2014).

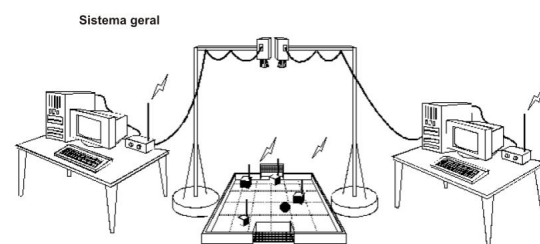


Figura 1: Organização do Sistema Geral de uma partida, retirada de (IEEE, 2014).

Os robôs utilizados pelas equipes possuem dois motores que movimentam uma roda de cada lado. Desta forma, é possível fazer o robô ir para frente, para trás ou girar, conforme o sentido de rotação e a velocidade angular de cada motor. Normalmente, as equipes utilizam robôs que recebem do computador a velocidade angular (rotação) e o sentido de rotação que devem ser colocados em cada um dos motores. Observa-se que com essa técnica há a vantagem do robô não precisar de uma elevada capacidade de processamento de dados, deixando assim o robô simples e barato.

Essa técnica, entretanto, exige uma câmera e um computador com desempenhos elevados para realizar o processamento de imagem. E, para o robô se locomover corretamente, o sistema deve ter alta taxa de atualização de sua posição, o que exige uma câmera que forneça cada quadro de imagem rapidamente (mais de 30 quadros por segundo). Além disso, deve haver um computador que consiga processar a imagem rapidamente, extraindo a posição dos robôs e da bola e determinando qual deve ser o sentido e velocidade de rotação de cada motor dos três robôs do time. Devido ao alto desempenho exigido, a câmera e o computador normalmente apresentam um custo elevado.

A proposta desse trabalho é construir um sistema de controle, que envie comandos para um robô dotado de uma bússola magnética que auxilia na sua movimentação. Dessa forma, o computador pode transmitir comandos estratégicos ou de movimentação mais simples, com uma taxa de transmissão menor, pois o próprio robô controlará a velocidade e sentido de rotação de seus motores.

2 Hardware do Robô

De acordo com as regras, os robôs devem ter tamanho máximo de um cubo com lado de 8 cm. A estrutura física e mecânica do robô deve ser rígida o suficiente para suportar eventuais colisões com outros robôs durante o jogo, além de prover locais para o armazenamento de seus componentes eletrônicos. Desta forma, utilizou-se uma estrutura de encaixe de placas, separadas por espaçadores de nylon que são colocados com parafusos. Foram utilizadas placas de fibra de vidro e de fenolite sendo estes materiais usados na fabricação dos circuitos impressos (Barbosa et al., 2016).

Os dispositivos selecionados para construção do sistema embarcado de controle do robô foram: microcontrolador C8051F310 (Silicon Labs, 2006), sensor de campo magnético modelo HMC5883L (Honeywell, 2013), conversor CC-CC boost U3V12F9 (Pololu, 2014b), regulador linear de tensão 78L33 (ST Microeletronics, 2014), módulo bluetooth HC-05 (Itead Studio, 2010), ponte H DVR-8835 (Pololu, 2014a), e dois micromotores metal gearmotor HP 100:1. A Figura 2 mostra a montagem do robô (Barbosa et al., 2016).

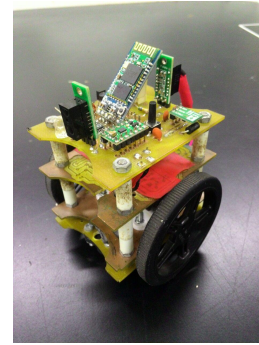


Figura 2: Foto do robô construído dotado de bússola magnética.

3 Firmware

O microcontrolador 8051F310 do robô foi programado utilizando-se a IDE de programação gratuita, fornecida pelo fabricante, e o compilador SDCC open source (Nicolosi and Bronzeri, 2005; SDCC, 2014). Com isso, foi implementado um controlador PID (Proporcional-Integral-Derivativo) discreto no firmware do robô (Wescott, 2000) para o controle da movimentação em linha reta, em curva e giro no próprio eixo, diminuindo assim o processamento realizado pelo computador e a necessidade de elevadas taxas de frames da câmera.

O sistema proposto utiliza apenas leituras de uma bússola magnética para manter a trajetória de deslocamento, um método diferente do encontrado em trabalhos semelhantes como (Seong and Kim, 1996) e (Sandi et al., 1998), que utilizam mais sensores e a bússola apenas como auxiliar na movimentação do robô.

O controlador PID, que é executado em uma rotina de interrupção chamada a cada 20 milissegundos (Figura 3), necessita de um valor de referência, neste caso, o ângulo desejado que o robô deve estar direcionado e o valor do ângulo atual que o robô está direcionado, no momento da leitura da bússola, para gerar o valor de erro (valor de referência menos o valor atual). Esse erro é utilizado pelo controlador para criar um valor de correção posteriormente aplicado aos PWM's dos dois motores do robô. Ou seja, o controlador PID discreto lê o ângulo da bússola do robô, verifica a diferença deste ângulo com o ângulo desejado e ajusta a velocidade de rotação das duas rodas do mesmo, a fim de manter a direção de deslocamento com a velocidade solicitada pelo computador, este procedimento é apresentado na Figura 4. Quando o robô se movimenta em trajetórias curvilíneas, o ângulo de referência é incrementado periodicamente com um valor diferente para cada raio de curvatura desejado, como o fluxograma da Figura 5.

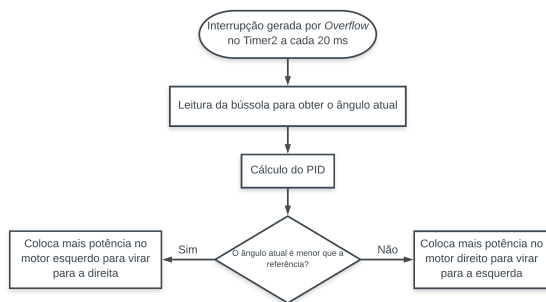


Figura 3: Fluxograma da função de interrupção periódica.

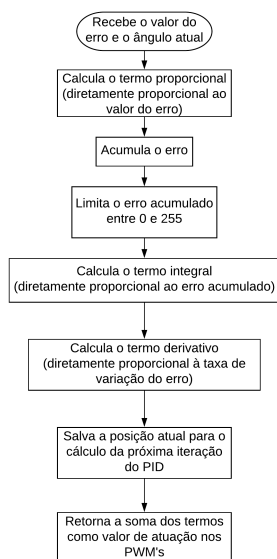


Figura 4: Fluxograma da leitura do ângulo da bússola.

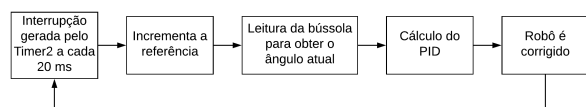


Figura 5: Fluxograma do método utilizado para fazer o robô deslocar-se em trajetórias curvas.

Foi definido um protocolo que será usado pelo computador para controlar o robô andando reto, na realização de curvas e giro no eixo, e também facilitar a integração com o software no computador. Desta forma, foi decidido que o pacote de dados recebido pelo robô conterá duas informações, a velocidade e um raio/ângulo. A Tabela 1 mostra o protocolo criado para controlar o robô.

Tabela 1: Relação das informações enviadas e ações realizadas.

	Reto	Giro	Curva	Parado
Velocidade	-255,255	0	-255,255	0
Raio / Ângulo	0	-180°,180°	-320mm,320mm	0

3.1 Comando para Andar Reto

Para que o robô realize uma trajetória reta, o pacote irá conter informações da velocidade, que pode ser positiva, para o robô ir para frente, ou negativa, para ir para trás, e o valor do raio em zero, pois deseja-se manter a direção do robô. Um pacote com ambas as informações em zero é enviado quando se deseja parar o robô.

3.2 Comando para Girar no Próprio Eixo

As informações necessárias para um giro no próprio eixo são a velocidade com valor zero, pois o giro é realizado em uma velocidade fixa, e o ângulo, que representa um valor em graus que deve ser variado a partir de seu ângulo inicial. O ângulo pode ser negativo, caso se queira girar no sentido anti-horário, e positivo caso se queira girar no sentido horário.

A bússola magnética é sensível e sujeita a interferências externas e do próprio robô (campo magnético dos motores). A solução encontrada para corrigir as distorções do campo magnético captado pela bússola foi comparar as medidas da bússola do robô com as medidas de um transferidor. Assim, fixou-se o 0 do robô com o 0 do transferidor, e foram calculadas as médias de 5 medidas realizadas a cada 10 graus de giro. Com isso, foi obtido um gráfico dos pontos coletados, a fim de realizar uma regressão polinomial, de forma que a curva dessa regressão fosse muito próxima dos pontos do gráfico. Com a equação obtida por meio da regressão, pode-se realizar a correção das medidas obtidas da bússola.

Com a regressão de grau 4, obteve-se a curva com o melhor ajuste sobre os pontos coletados, como mostra a Figura 6. Com isso, foi obtida a função para esta curva, de modo que aplicando-se os valores da bússola do robô nessa função, resulta o ângulo real.



Figura 6: Regressão polinomial de grau 4. O eixo horizontal mostra as medidas obtidas por meio da bússola do robô e o eixo vertical as medidas obtidas por meio do transferidor.

3.3 Comando para Realizar uma Circunferência

Para curvas, são enviados parâmetros de velocidade, que pode ser positiva para que a trajetória seja para frente, e negativa para trás. Para o

mesmo movimento, também é transmitida a informação do raio, que indica o raio da circunferência a ser realizada, que quando é negativo indica curva para a esquerda e quando é positivo indica curva para a direita.

Para a criação deste comando, foi necessário encontrar o fator de conversão de pixels para centímetros, pois as técnicas de medidas realizavam contagens de pixels em uma imagem, para uma trajetória circular realizada pelo robô. Para isso, foi colocada uma régua na arena e a partir da imagem obtida pela câmera, foi realizada a contagem de pixels correspondente à dimensão da régua. Então, verificou-se o fator de conversão de aproximadamente 4,2 milímetros por pixel, como mostra a Figura 7.

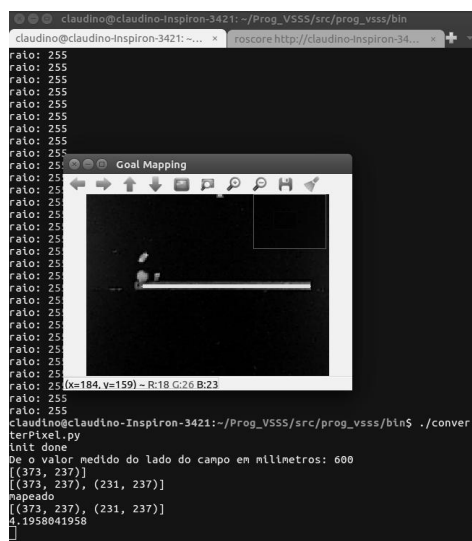


Figura 7: Cálculo do fator de conversão de pixels para milímetros pelo *software*.

Para se encontrar a relação entre o raio da trajetória circular do robô e o incremento do ângulo de referência, utilizou-se alguns valores de incremento (1.0, 1.5, 2.0, 2.5, 3.0 e 3.5 graus) e registrou-se, por meio do software de visão computacional, as coordenadas XY (em pixels) da circunferência descrita pelo robô. Em seguida, por meio do maior e do menor valor em cada eixo, calculou-se a diferença e obteve-se o valor do diâmetro das circunferências nos eixos x e y, em pixels. Para cada um dos valores de incremento testados, foram realizados 4 testes, a fim de se obter uma média para cada um deles. Por fim, os valores médios encontrados foram transformados para centímetros com a proporção encontrada anteriormente. Com os resultados obtidos, foi gerado um gráfico com pontos representando o diâmetro das circunferências e seu valor de incremento. Com isso foi realizada uma regressão polinomial de grau 4, mostrada na Figura 8.

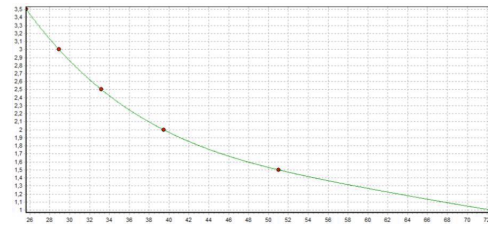


Figura 8: Cálculo do fator de conversão de pixels para milímetros pelo *software*.

A curva gerada pela regressão se apresenta bem ajustada aos pontos, de forma que ao se aplicar um valor de diâmetro, é retornado o valor de incremento a ser utilizado para que o robô realize uma curva com o raio desejado.

4 Software

O *software* que extrai dados de posição da imagem e envia comandos para os robôs foi feito em um *framework* flexível chamado *Robot Operating System* (ROS), o que ajudou na modularização do programa. A Figura 9 mostra como o algoritmo de estratégia funciona quando o robô está depois da bola, ou seja, quando o valor do Y da posição do robô é menor que o da bola, sendo a referência no canto superior esquerdo da imagem. Após isso verifica se o robô está a direita ou a esquerda da bola, dado o X da bola, para definir o ponto A atrás, do lado em que está posicionado o robô, evitando que ele se choque com ela e acabe chutando a bola contra o próprio gol. Depois, alinha a frente do robô com esse ponto girando no próprio eixo e anda reto para posicionar-se atrás da bola (Ponto A).

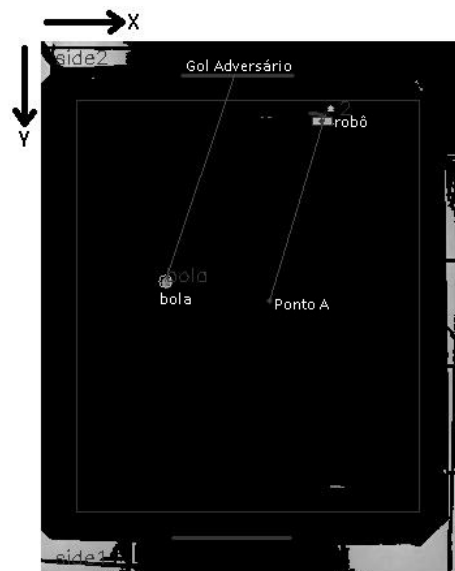


Figura 9: Estratégia quando o valor de Y da posição do robô é menor que o da bola.

A Figura 10 mostra como o algoritmo de estratégia funciona quando o robô está atrás da

bola, ou seja, quando o valor do Y da posição do robô é maior que o da bola. Neste caso é calculado o raio da circunferência, que passa pelas posições do robô e da bola de forma que uma reta tangente dessa circunferência liga o centro da bola ao centro do gol adversário. Após isso, o robô gira no próprio eixo para alinhar com a reta que tangencia a circunferência na posição do robô e realiza o movimento de curva para chutar ao gol.

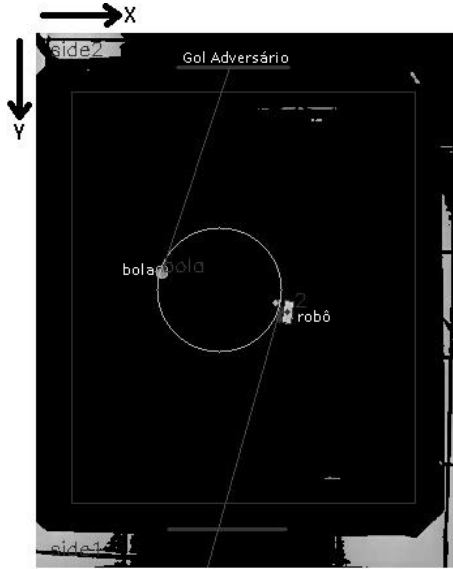


Figura 10: Estratégia quando o valor de Y da posição do robô é maior que o da bola.

Com base na Figura 11, a estratégia citada funciona de forma que uma reta bg é traçada entre o ponto do gol adversário (ponto G) e o ponto da bola (ponto B), obtida pela Equação (1). É utilizada a Equação (2) para obter o coeficiente angular (m) da reta bg por meio dos pontos G e B. Em seguida, uma outra reta p é traçada de forma que a mesma seja perpendicular à reta bg, ou seja, a condição de perpendicularidade dessas retas satisfaz a Equação (3), e que contenha o ponto da bola.

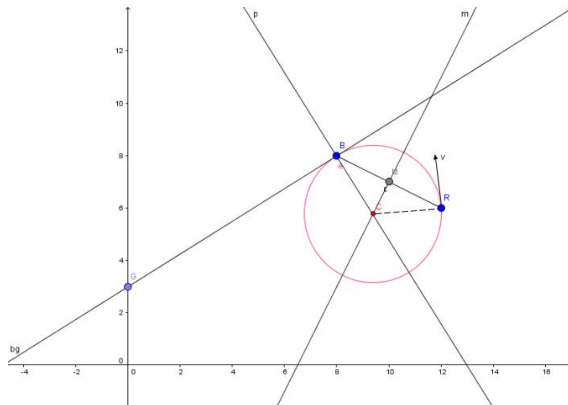


Figura 11: Estratégia para "chutar" a bola em direção ao gol do time adversário realizando movimentos circulares (obtida pelo programa GeoGebra).

$$y_b - y_g = m_{bg}(x_b - x_g) \quad (1)$$

$$m = \frac{\Delta x}{\Delta y} \quad (2)$$

$$m_{bg} \times m_p = -1 \quad (3)$$

A circunferência é obtida através das retas p e m, onde m é a reta perpendicular ao segmento de reta RB (R é a posição do robô) e que contém o ponto médio (ponto M) desse segmento, obtido pela Equação (4). Da intersecção dessas duas retas, obtém-se o centro (ponto C) da circunferência, de modo que a distância entre o centro e a bola seja igual à distância entre o centro e o robô, que pode ser calculado por (5). Essa distância é o raio da circunferência. Com o valor do coeficiente angular do segmento de reta entre C e R (segmento CR), encontra-se o ângulo de direção que o robô deve ter (perpendicular à CR) para tangenciar a circunferência no ponto R. Esta direção é dada pelo vetor v. A partir desse ângulo, calcula-se quantos graus o robô deve girar no próprio eixo para ficar com o ângulo de direção correto. Depois, o robô realiza uma curva com o raio obtido anteriormente. Esta curva é dada pela circunferência c, obtida por (6).

$$M = \left(\frac{x_b + x_r}{2}, \frac{y_b + y_r}{2} \right) \quad (4)$$

$$d_{cr} = \sqrt{(x_r - x_c)^2 + (y_r - y_c)^2} \quad (5)$$

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \quad (6)$$

5 Resultados

Para verificar a eficácia dessa estratégia utilizando o robô com bússola foram realizados testes para cada uma das seguintes posições da bola:

- Bola posicionada próxima a borda direita do campo;
- Bola posicionada no centro campo;
- Bola posicionada próxima a borda esquerda do campo;

Para os testes com a bola posicionada próxima as laterais do campo, como mostra a Figura 12, o robô foi posicionado em um ponto a frente da bola e realizados trinta testes com a frente do robô virada para a borda do gol adversário (Norte), depois com ele virado para a borda esquerda (Leste), direita (Oeste) e a do próprio gol (Sul). A intenção foi verificar a eficiência da estratégia em retirar a bola das laterais do campo, algo que ocorre com frequência durante uma partida. Na Figura 14, os pontos vermelhos mostram o caminho do robô e os amarelos a trajetória da bola. Com a bola

localizada no centro do campo, o robô foi posicionado em oito posições diferentes em volta, como mostra a Figura 13, e foram feitos os mesmos testes anteriormente descritos para cada posição. A Figura 15 mostra alguns exemplos.

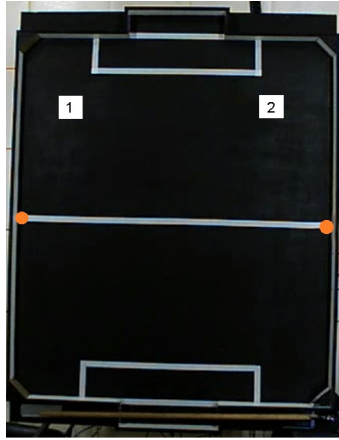


Figura 12: Posições do robô nos testes com a bola, representada pelo círculo laranja, localizada na Borda Esquerda, ou seja, na posição 1, e na Borda Direita na posição 2.

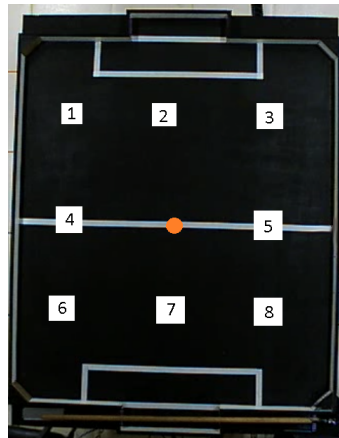


Figura 13: Posições do robô nos testes com a bola localizada no centro do campo.

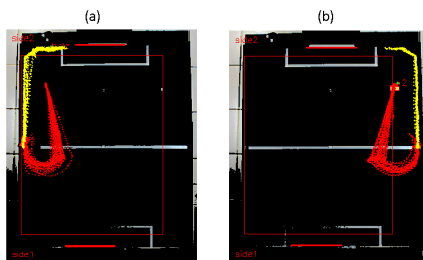


Figura 14: Exemplo de 30 testes para chutar a bola encostada na borda esquerda (a) e na borda direita (b).

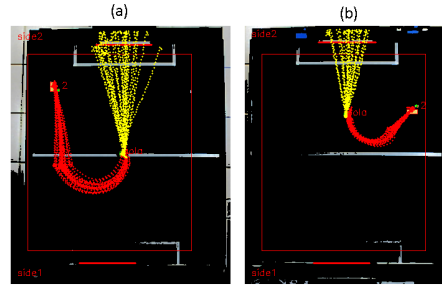


Figura 15: Exemplo de 30 testes para chutar a bola com o robô posicionado na frente (a) e ao lado (b).

Devido ao problema do robô ficar preso na fita do meio campo ao rotacionar no próprio eixo, os testes ao lado da bola foram feitos um pouco a frente da posição original, como mostra na Figura 15 (b). A Tabela 2 mostra para cada um dos testes a média e o desvio padrão do erro em milímetros entre o caminho percorrido pelo robô e o caminho ideal feito pelo computador.

Tabela 2: Resultados do erro obtido entre o caminho do robô, o caminho ideal para chute e dos chutes errados que não acertaram o gol e a taxa de acertos no gol.

Teste		Em milímetros		Chutes errados durante 30 testes	Taxa de acertos (%)
		Média	Desvio Padrão		
Borda Esquerda	Norte	20,929198	25,793181	1	96,67
	Sul	14,589354	15,222578	1	96,67
	Leste	20,550811	21,804478	1	96,67
	Oeste	18,750361	24,970354	0	100,00
Borda Direita	Norte	24,263608	30,304457	2	93,33
	Sul	25,960841	41,913359	2	93,33
	Leste	30,522414	39,278007	3	90,00
	Oeste	25,286892	46,365094	3	90,00
Posição 1	Norte	24,155129	40,551945	4	86,67
	Sul	24,687274	33,620193	4	86,67
	Leste	26,915599	45,229259	4	86,67
	Oeste	37,994402	43,962587	6	80,00
Posição 2	Norte	27,291064	37,442717	4	86,67
	Sul	29,296831	39,139522	3	90,00
	Leste	37,786656	48,883496	3	90,00
	Oeste	26,519264	31,948582	2	93,33
Posição 3	Norte	17,307849	20,141362	2	93,33
	Sul	14,001600	15,396646	3	90,00
	Leste	18,881052	25,082171	1	96,67
	Oeste	25,591377	22,569289	5	83,33
Posição 4	Norte	10,231623	10,430864	0	100,00
	Sul	7,303785	6,404134	0	100,00
	Leste	13,359021	15,960420	1	96,67
	Oeste	9,576696	8,753990	0	100,00
Posição 5	Norte	11,247976	14,218307	1	96,67
	Sul	10,637078	10,442434	0	100,00
	Leste	9,602964	9,171312	1	96,67
	Oeste	8,609925	7,895140	0	100,00
Posição 6	Norte	14,933692	15,417593	1	96,67
	Sul	12,503279	13,172343	2	93,33
	Leste	15,661218	18,167593	4	86,67
	Oeste	9,456739	11,829779	1	96,67
Posição 7	Norte	12,409062	14,401398	2	93,33
	Sul	13,844835	16,433210	6	80,00
	Leste	13,028786	18,573730	4	86,67
	Oeste	17,131361	20,386354	6	80,00
Posição 8	Norte	8,815816	9,415815	3	90,00
	Sul	8,855794	8,697382	1	96,67
	Leste	12,902912	11,749129	3	90,00
	Oeste	9,053243	9,563042	4	86,67
Média da Taxa de acertos (%)					93,3333

6 Conclusão

De acordo com (Lacerda et al., 2017), que realizou uma pesquisa analisando dados encontrados nos noventa TDPs submetidos pelas equipes participantes na Competição Brasileira de Robótica de 2013 à 2016, foi concluído que muitos times

tentaram usar sensores como giroscópio, acelerômetro e magnetômetro, porém pararam de utilizá-los por não encontrarem nenhum resultado expressivo. Com a utilização da bússola, o próprio robô controla a sua movimentação, em linha reta, fazendo curvas ou girando no próprio eixo, e isso resulta em uma diminuição de cálculos e envio de comandos de controle pelo computador, que passa a ter uma função de monitoramento com intervalo de tempo maior que nos sistemas convencionais (quadro a quadro), permitindo o uso de computadores e câmeras com desempenho menor e barateando o investimento nas equipes. Com os dados mostrados na Seção 5, percebe-se que a técnica apresentou bons resultados, encorajando a implementação futura de um time de robôs com bússolas magnéticas para utilização em jogos oficiais.

7 Trabalhos Futuros

Algumas observações foram feitas durante a realização de testes, resultando nas seguintes propostas de trabalhos futuros:

- Melhoria do código de curva no firmware do robô para considerar a velocidade juntamente com o raio, tornando o robô mais dinâmico e preciso;
- Realização de uma análise da resposta do PID para um ajuste melhor dos ganhos e obter um controle mais preciso;

Agradecimentos

Os autores agradecem à Universidade Federal de Mato Grosso do Sul (UFMS) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo auxílio financeiro.

Referências

- Barbosa, A. T. R., Visioli, I. S. A., Silva, L. H. C., Marães, L. F. and Iaione, F. (2016). Construção de um Robô Jogador com Sistema Embarcado e Sensores, *Atas das Conferências IADIS Ibero Americanas WWW/Internet 2016 e Computação Aplicada 2016*, Conferência IADIS Ibero Americana em Computação Aplicada, Lisboa, pp. 213–220.
- Honeywell (2013). Datasheet: 3-axis Digital Compass IC HMC5883L.
- IEEE (2014). Rules for the IEEE Very Small Competition Version 1.0.
- Itead Studio (2010). Bluetooth Datasheet: HC-05 - Bluetooth to Serial Port Module iteadstudio.
- Lacerda, L. P., de Oliveira, A. C. and Todt, E. (2017). Survey on IEEE Very Small League Robots.
- Nicolosi, D. E. C. and Bronzeri, R. B. (2005). *Microcontrolador 8051 com linguagem C: Prático e Didático - Família AT89S8252 Atmel*, 1 edn, Editora Érica, São Paulo.
- Pololu (2014a). Datasheet: DRV8835 dual motor driver carrier.
- Pololu (2014b). Datasheet: Pololu 9V step-up voltage regulator U3V12F9.
- Sandi, F. A. L., Hemerly, E. M. and Lages, W. F. (1998). Sistema para Navegação e Guiagem de Robôs Móveis Autônomos, **9**(3): 107–118.
- SDCC (2014). Small Device C Compiler.
- Seong, P. H. and Kim, J. H. (1996). Experiments on Orientation Recovery and Steering of an Autonomous Mobile Robot Using Encoded Magnetic Compass Disc, **45**(1): 271–274.
- Silicon Labs (2006). Datasheet: C8051F310/1/2/3/4/5/6/7/8/16 kB ISP Flash MCU Family.
- ST Microelectronics (2014). Datasheet: L78L Positive voltage regulators, DocID2145 Rev 22.
- Wescott, T. (2000). PID Without a PhD. Embedded Systems Programming.