

**Construção de robô autônomo com o kit lego para modalidade prática OBR
- fase regional**

**Construction of autonomous robot with the lego kit for practical OBR -
regional phase**

DOI:10.34117/bjdv6n12-503

Recebimento dos originais:11/11/2020

Aceitação para publicação:20/12/2020

Eduardo Rodrigues Amaral

Técnico em Informática

Instituição de atuação atual: CEFET-MG

Endereço :Rua Alvares de Azevedo, 400, Bela Vista, Divinópolis, MG

E-mail: eduardoroamaral@gmail.com

Henrique Silva Rabelo

Técnico em Informática

Instituição de atuação atual: CEFET-MG

Endereço: Rua Alvares de Azevedo, 400, Bela Vista, Divinópolis, MG

E-mail: rabelohenry@gmail.com

Igor Eduardo Teodoro

Técnico em Informática

Instituição de atuação atual: CEFET-MG

Endereço :Rua Alvares de Azevedo, 400, Bela Vista, Divinópolis, MG

E-mail: ieteodoro050115@gmail.com

Márcio Ribeiro de Oliveira

Técnico em Informática

Instituição de atuação atual: CEFET-MG

Endereço: Rua Alvares de Azevedo, 400, Bela Vista, Divinópolis, MG

E-mail: marcioribeiro7390@gmail.com

Alisson Marques da Silva

Doutorado

Instituição de atuação atual: CEFET-MG

Endereço :Rua Alvares de Azevedo, 400, Bela Vista, Divinópolis, MG

E-mail: alisson@cefetmg.br

RESUMO

Este trabalho descreve o desenvolvimento de um robô utilizando o Kit Lego EV3 para a modalidade prática da OBR. Foi desenvolvido um robô capaz de seguir linha, ultrapassar redutores, subir rampas, tomar decisões e desviar de obstáculos. Para desenvolvimento foi utilizado o *software* livre EV3dev juntamente com a linguagem Python e a biblioteca ev3devpython. Além disso, foi utilizado um controlador PID (Proporcional-Integrativo-Derivativo). Foram realizados testes em laboratório para ajuste dos parâmetros e, posteriormente, o robô desenvolvido participou da Etapa Regional da OBR e obteve o segundo lugar, classificando a equipe para a Etapa Estadual. Este artigo visa descrever o processo de desenvolvimento do robô, apresentar os resultados obtidos e apresentar as dificuldades encontradas. Espera-se que este possa incentivar e contribuir no desenvolvimento de projetos relacionados à robótica.

Palavras Chaves: Robótica, Competição, OBR, Mecânica, Programação, PID, EV3, EV3dev.

ABSTRACT

This paper describes the development of a robot using the Lego EV3 kit for the practical modality of the Brazilian Robotic Olympics. It was developed a robot capable of following lines, overcome reducers, climb ramps, make decisions and dodge obstacles. For the development was used the free software EV3dev in combination with the python language and it's library ev3devpython besides concepts of PID controller that use integral, derivative and proportional. The tests were satisfactory and the result in the competition positive taking us to the next stage (state stage). This report has the objective to inform people interested in developing robotic related projects.

Keywords: Robotic, Competition, OBR, Mechanic, Programming, PID, EV3, EV3dev.

1 INTRODUÇÃO

A robótica vem com uma base sólida que é também utilizada para o desenvolvimento da interdisciplinaridade (dentro da robótica) servindo como ferramenta eficiente neste conceito de estudo (NETO et al., 2015). O CEFET-MG Campus V explora estes benefícios através de projeto voltado aos alunos do Curso Técnico em Informática e Técnico em Mecatrônica. No projeto os alunos são divididos em equipes e estas tem o objetivo de desenvolver um robô para participar na Olimpíadas Brasileira de Robótica, Modalidade Prática. Robô este que será descrito neste trabalho. A Olimpíada Brasileira de Robótica (OBR) é uma das olimpíadas científicas brasileiras que utiliza-se da temática da robótica. Tem o objetivo de estimular os jovens às carreiras científico-tecnológicas, identificar jovens talentosos e promover debates e atualizações no processo de ensino-aprendizagem brasileiro (OBR, 2018).

Este artigo está organizado de forma que na seção 2 é descrito os procedimentos necessários para o desenvolvimento do robô. Na seção 3 são apresentados os testes realizados. Na seção 4 é descrito as discussões acerca do desenvolvimento e os resultados encontrados. Já na seção 5 encontra-se as considerações finais.

2 DESENVOLVIMENTO DO ROBÔ

O robô foi desenvolvido de maneira a concluir com sucesso algumas das tarefas propostas na OBR 2018, das quais podemos citar seguir linhas, desviar de obstáculos, realizar decisões, ultrapassar redutores e subir rampas. Inicialmente algumas tarefas propostas na competição foram descartadas para a Etapa Regional e serão desenvolvidas para a Etapa Estadual, como por exemplo o resgate das vítimas.

2.1 MECÂNICA

Para o desenvolvimento do robô foi utilizado o Kit Lego EV3. A estrutura do robô foi totalmente construída com peças de Lego e foi levado em consideração o peso para ajudar na tração dianteira e assim facilitar subir rampas e ultrapassar redutores. Foram utilizados dois servo motores para tração dianteira e uma esfera na traseira. As dimensões do robô, posicionamento das rodas e dos sensores foram todas construídas em função da programação e não o inverso e ao longo da programação o robô foi adaptado mecanicamente para atender as necessidades. A Figura 1 ilustra o robô visto de frente enquanto a Figura 2 o mostra visto por trás.

Figura 1 – Robô – Visto pela frente



Figura 2 – Robô – Visto por trás



2.1.1 Sensores

Foram utilizados dois sensores de cor, para seguir linha, além de um sensor ultrassônico para medir a proximidade dos obstáculos. Estes sensores podem ser vistos na Figura 3.

Figura 3 – a) Sensor Ultrassom – b) Sensor RGB/Luminância



2.1.2 Motores

Foram utilizados dois servo motores para tração das rodas dianteiras. O exemplo de um servo motor pode ser visto na Figura 4.

Figura 4 – Servo Motor



2.2 PROGRAMAÇÃO

Para o desenvolvimento do algoritmo responsável pelo controle do robô, foi utilizado o software livre ev3dev (EV3DEV, 2018) juntamente à biblioteca ev3devpython (EV3DEVPYTHON, 2018). A linguagem Python foi escolhida pelo seu amplo suporte ao EV3 devido à biblioteca dedicada a ele, além de sua grande facilidade de escrita, compreensão e sua versatilidade no que se diz respeito à resolução de problemas. O grande suporte encontrado pela comunidade e o alto nível da linguagem também foi considerado na escolha. Foi utilizado o editor de texto NANO para edição dos códigos.

2.2.1 Seguidor de Linha

O seguidor de linha foi desenvolvido com base em controlador PID, sendo que a proporcional, derivativa e integral influenciam na rotação das rodas, tendo assim um sistema diferencial. Dentre as vantagens do controle PID cita-se (ALMEIDA; COELHO, 2001):

- i) o conjunto de regras é simples e intuitivo;
- ii) é completamente auto-ajustável, necessitando apenas dos parâmetros de um controlador PID, não necessariamente bem sintonizado;
- iii) os limites de estabilidade são facilmente estabelecidos através do teorema do pequeno ganho.

As constantes k_i , k_p , k_d são definidas previamente. Inicialmente é feito uma calibração dos valores preto e branco de acordo com a luminosidade do ambiente. Calcula-se então o *offset* através de uma média simples entre os valores do preto e branco. A cada leitura calcula-se a diferença entre a leitura atual e o valor do *offset* obtendo assim o erro. A partir dos valores da integral, derivada e do erro (proporcional) calcula-se a velocidade final (em porcentagem) do motor através da seguinte equação:

$velocidade_final = k_p * erro + k_d * derivada + k_i * integral + velocidade_constante;$

Executa-se então os cálculos necessários para atualizar os valores da derivada e da integral.

$integral = soma_dos_erros / numero_de_leituras;$

$derivada = proporcional_anterior;$

Essa rotina é realizada para os dois motores, cada um com seu respectivo sensor de luminância. Através desse método obtém-se um seguidor de linha suave e preciso. Além disso foi criado uma rotina específica para obedecer curvas de noventa graus e realizar algumas decisões.

2.2.2 Desvio de Obstáculos

Para o desvio de obstáculos utilizou-se leituras do sensor de ultrassom que a partir de uma certa distância iniciava-se uma rotina padrão para desvio de obstáculos. Porém esse método se provou falho em obstáculos encontrados logo após uma curva (detalhes na seção 4).

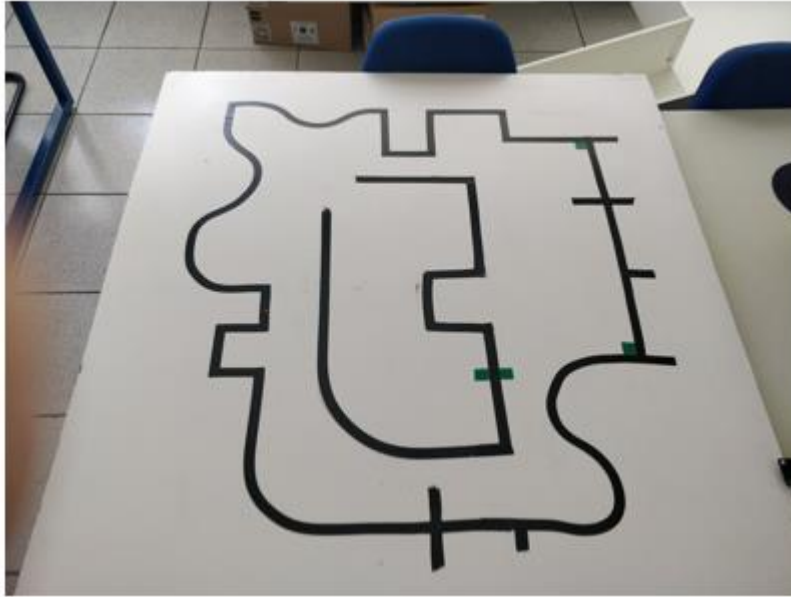
2.2.3 Redutores e rampas

Devido à falta de tração e velocidade suficiente para ultrapassar alguns redutores e subir rampas, implementou-se uma rotina para tal. Ao detectar múltiplas leituras de luminância de valores semelhantes, assume-se que o robô está parado em algum obstáculo, então o recua aproximadamente 1 cm, aumenta-se sua velocidade por um curto período de tempo e então volta-se à rotina PID.

3 PROCEDIMENTOS PARA TESTE

Para o teste do robô foi utilizado pistas inspiradas em edições passadas da OBR, as quais continham, rampas, área de resgate e trajetos construídos com fita isolante preta e verde. A análise visual foi utilizada para determinar o sucesso do robô, e os testes executados múltiplas vezes para confirmação do resultado. As figuras 6 e 7 mostram as pistas de teste.

Figura 6 – Pista usada para seguir linha



4 DISCUSSÃO SOBRE OS RESULTADOS

O resultado final foi satisfatório e alcançou-se pontuações suficientes para classificação para Fase Estadual da OBR em segundo lugar. Porém foram encontradas algumas falhas descritas a seguir.

4.1 FALHA NO DESVIO DE OBSTÁCULOS

O desvio de obstáculos se mostrou falho ao encontrar obstáculos logo após curvas, devido à imprecisão da leitura da distância do obstáculo logo após a curva, falhando em detectar a curva. Em algumas situações de objetos após a curva a rotina se iniciava porém o procedimento padrão falhava em retornar o robô ao trajeto. Uma solução para tornar a leitura mais precisa neste caso seria recuar o sensor para o meio do robô.

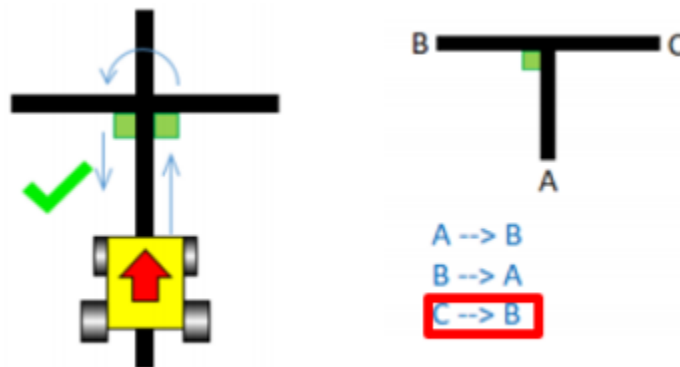
Figura 7 – Rampa utilizada



4.2 FALHA NA TOMADA DE DECISÕES

A rotina PID era incapaz de tomar decisões duplas ou ignorar decisões quando não estava no sentido correto para a tomada da mesma. Mostrou-se a necessidade de outra rotina computacional para tal. Esta rotina deve ser capaz de detectar verdes duplos e diferenciar fielmente verdes subsequentes à pretos de verdes subsequentes à brancos. Os dois casos de falha podem ser vistos nas Figura 8.

Figura 8 – Caso 1 falho – b) Caso 2 falho



4.3 DESVANTAGENS NA SOLUÇÃO PARA REDUTORES E RAMPAS

A solução implementada para ultrapassar redutores e subir rampas que utilizava de software e aumentava a velocidade quando necessário (citada em 2.2.3) mostrou a desvantagem de atrasar o

percurso e em algumas retas longas acelerar erroneamente e tirar o robô do trajeto. Concluiu-se então a vantagem de uma solução mecânica para o problema em futuras implementações.

5 CONSIDERAÇÕES FINAIS

Após o desenvolvimento conclui-se a eficiência do sistema PID para seguir linhas de maneira eficiente e suave. Concluiu-se a necessidade de novas rotinas e adaptações mecânicas para correção de falhas no desvio de obstáculo e tomada de decisões. Encontrou-se também dificuldade em detectar a cor verde através dos sensores de cor, algo a ser considerado para próxima fase. Pretende-se também ampliar o projeto para a realização do resgate, para isso utilizaremos uma garra já construída e é necessário criar uma rotina para detecção da vítima e seu resgate.

REFERÊNCIAS

ALMEIDA, O. M.; COELHO, A. A. R. Controlador pid com escalonamento nebuloso dos ganhos: Auto-sintonia, análise e implementação. SBAI, 2001.

EV3DEV, E. ev3dev is your EV3 re-imagined. 2018. Ev3dev Website. Disponível em: <<https://www.ev3dev.org/>>. Acesso em: 14.8.2018.

EV3DEVPYTHON, E. Pure python bindings for ev3dev. 2018. Ev3devpython Github Page. Disponível em: <<https://github.com/ev3dev/ev3dev-lang-python>>. Acesso em: 14.8.2018.

NETO, R. P. B. et al. Robótica na educação: Uma revisão sistemática dos últimos 10 anos. Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE 2015), p. 390, 2015.

OBR, E. O que é a OBR. 2018. Website OBR. Disponível em: <<http://www.obr.org.br/o-que-e-a-obr/>>. Acesso em: 14.8.2018.