SICITE XVII SEMINÁRIO DE INICIAÇÃO CIENTÍFICA E TECNOLÓGICA DA UTFPR

APLICAÇÃO DE TESTES UTILIZANDO LÓGICA CLÁSSICA NA PROGRAMAÇÃO DE PLATAFORMA ROBÓTICA MÓVEL

Thomas Massaru Okuyama[PIBIC][1], Márcio Mendonça[Orientador] [2], Keriton Lopes [Colaborador] [3]

[1] Programa Institucional de Bolsas de Iniciação Científica UTFPR

[2]Departamento de Engenharia Elétrica

[3] Departamento de Engenharia de Controle e Automação

Campus Cornélio Procópio
Universidade Tecnológica Federal do Paraná - UTFPR
Avenida Alberto Carazzai, 1640 CEP 86300-000 - Cornélio Procópio - PR – Brasil
Telefone Geral +55 (43) 3520-4000 - Fax: +55 (43) 3520-4010

thhomas@gmail.com, mendonca@utfpr.edu.br, kerinton@hotmail.com

Resumo - Este artigo apresenta uma avalição da plataforma robótica Curumim, para aplicações em robótica autônoma utilizando algoritmos inteligentes programados em linguagem c. Foram avaliados resultados iniciais demonstrando o comportamento dinâmico do robô. Utilizando algoritmos clássicos, construídos em blocos e linguagem c.

Palavras-chave: Plataforma Curumim; Navegação Robótica; Lógica Clássica.

Abstract - This article presents a review of Curumim robotic platform for applications in autonomous robotics using intelligent algorithms programmed in C language. We evaluated the initial results demonstrating the dynamic behavior of the robot. Using classical algorithms, built in blocks and c language.

Keywords: Platform Curumim; Navigation Robotics; Classical Logic.

INTRODUÇÃO

Robótica é uma área em crescimento nos dias atuais. Uma das áreas de pesquisa é a robótica autônoma, na qual robôs tem uma capacidade de serem independentes no quesito de suas funcionalidades, ou seja, tendo rara ou nem uma intervenção humana em suas funções [1] e [2].

A proposta dessa pesquisa é utilizar algoritmos baseados em lógica *fuzzy* e redes neurais artificiais em atividades autônomas, a lógica *fuzzy*, comumente conhecida como logica nebulosa, tem como proposito a modelagem do raciocínio humano de forma aproximada, a fim de desenvolver sistemas computacionais para o processo de tomada de decisão em ambientes incertos [3] e [4]. Outra possível contribuição é a de se utilizar algoritmos de aprendizagem por reforço para aumentar interação com o ambiente [5]. Entretanto, antes de se utilizar algoritmos de maior complexidade deve-se reconhecer a plataforma robótica e aplicar algoritmos baseados em tomadas de decisões a partir de blocos lógicos.

A plataforma robótica utilizada é o robô curumim, *opensource*, oferecendo ao usuário total liberdade em desenvolver aplicações.

A plataforma robótica (kit Curumim) é composta por:

1 robô Curumim
1 par de baterias para o robô
1 carregador das baterias
1 fonte de alimentação de 24Vcc para o carregador de bateria
1 rádio base e seu cabo USB
1 receptor de imagem *wireless* da câmera do robô
1 fonte de alimentação de 12Vcc para o receptor de imagem wireless
1 cabo RCA para o receptor *wireless.*

O robô contem sensores infravermelho, câmera sem fio e transceiver de radio.O sistema de locomoção é composto por três motoredutores *encoders*, controlados por uma unidade exclusiva(placa driver) capaz de verificar velocidade e posição dos motoredutores, realizando assim movimentos precisos. A comunicação é efetuada através de conexão *wireless* de 2.4GHz. Com a comunicação estabelecida entre robô e computador, o algoritmo assume o controle da plataforma móvel via *software.*

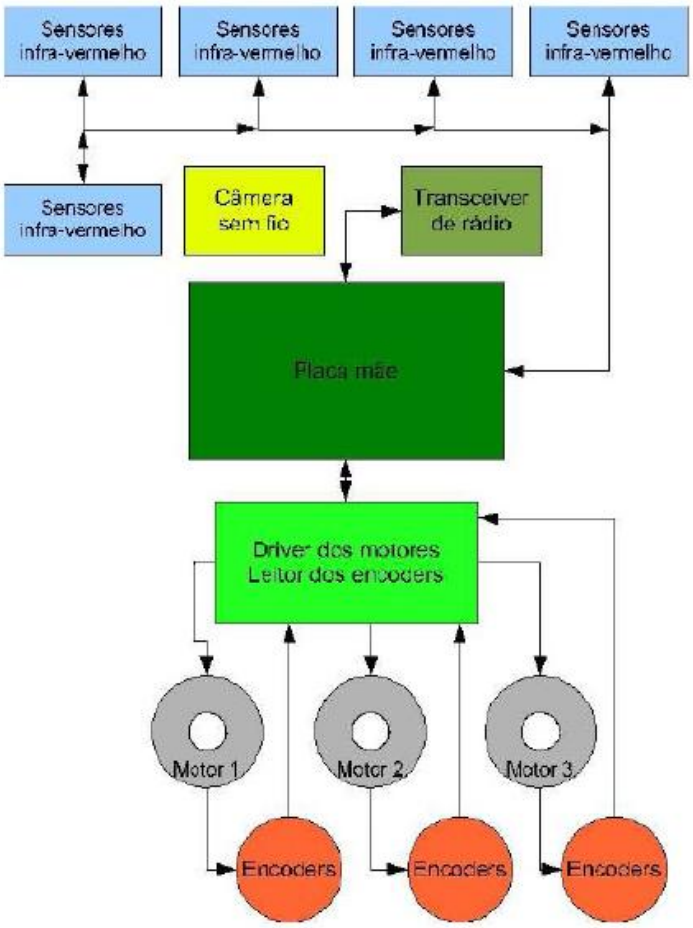

Figura 1. Diagrama da unidade móvel.

METODOLOGIA

A partir de dois trajetos, avaliamos os métodos de programação, estudando seu comportamento diante de obstáculos e movimentos distintos, possibilitando manobras evasivas com o propósito de evitar colisões.

Trajeto

Utilizando cones de sinalização como obstáculos foi possível montar trajetos de forma aleatória, com o objetivo de testar os sensores e lógica clássica na plataforma robótica.

Figura 2. Curumim entre obstáculos.

Programação

Diagrama dos blocos. Na figura abaixo temos um exemplo do diagrama dos blocos, onde todos os comandos estão representados graficamente. O algoritmo é montado alinhando os blocos de acordo com cada trajeto. Foi proposta uma trajetória testando: movimentos em linha reta para frente e para trás, laço para, giros e desvio para evitar possível colisão.

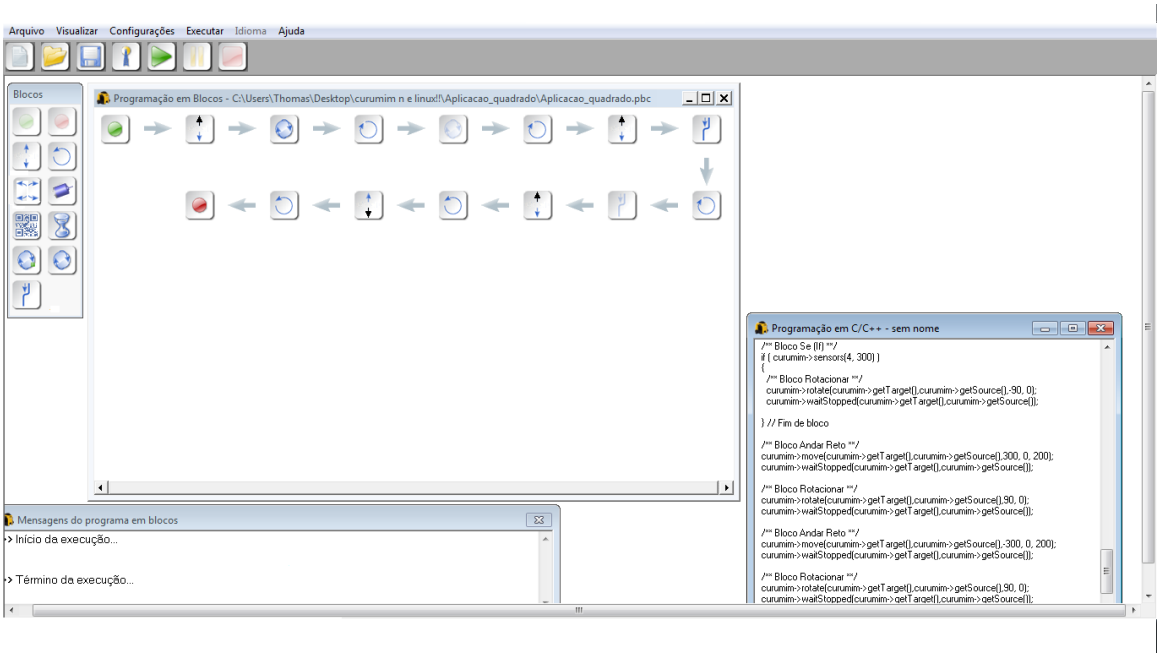

Figura 3. Diagrama dos blocos.

Linguagem c. Os algoritmos em linguagem c são escritos através de linhas com comandos especificos, após verificação de erros, o algoritmo é enviado ao robô via conexão sem fio, controlando-o de forma autônoma conforme algoritmo montado. Devido a maior compatibilidade entre *software* e *hardware*, utilizamos o IDE (*Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado) *Microsoft Visual c++*, que acompanha a plataforma com todas as bibliotecas e *softwares* adicionais necessários para executar corretamente todas as funções do curumim. No algoritmo teste foi compilada a seguinte logica:

Enquanto sensor frontal não captar obstáculos por 40cm, anda 20cm.
Se existir obstáculo na frente, voltar e vira a esquerda
Enquanto existir obstáculo nos outros sensores o robô vai realizar, ações de acordo com o sensor que está o obstáculo.
Se existir obstáculo na frontal esquerda, volta e vira a direita
Se existir obstáculo na frontal direita, volta e vira a esquerda.
Se existir obstáculo na traseira direita, volta e vira a esquerda.
Se existir obstáculo na traseira esquerda, volta e vira a direita.

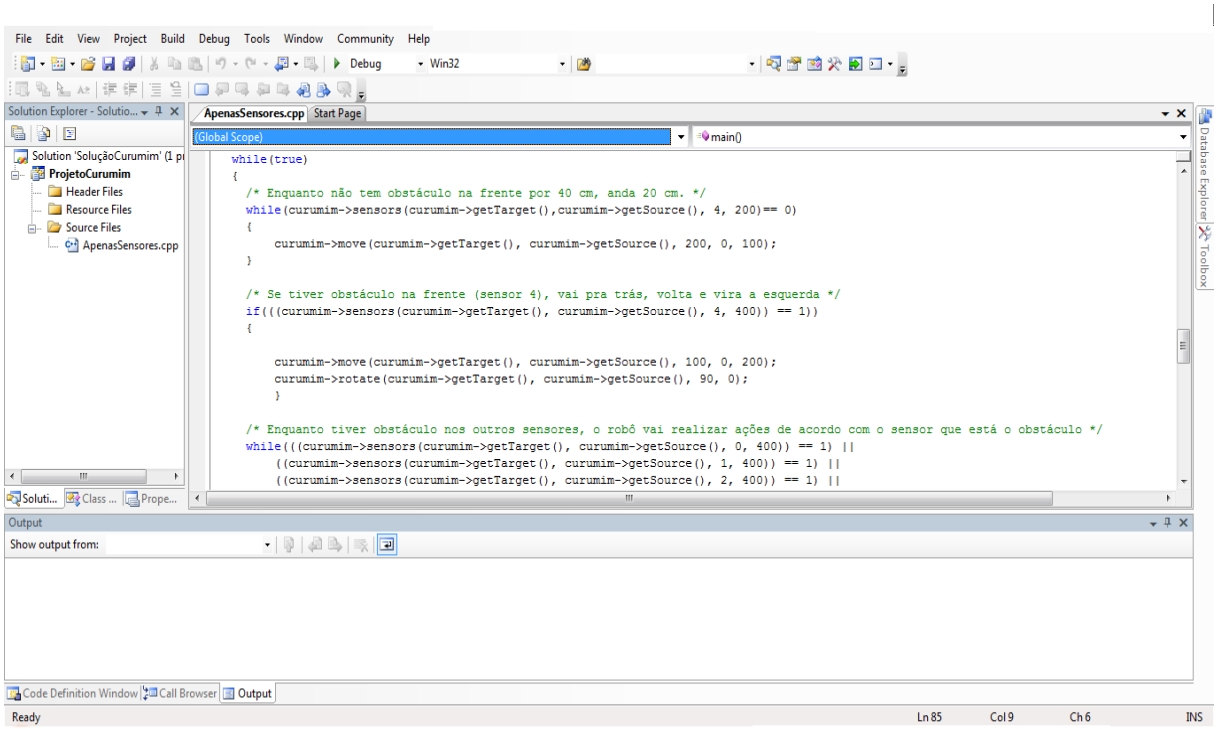

Figura 4. *Microsoft Visual c++*.

RESULTADOS E DISCUSSÃO

Obtivemos resultados satisfatórios, uma vez que a plataforma conseguiu de forma autônoma transpassar objetos. Quando compilado um conjunto de regras utilizando logica clássica através de um algoritmo em linguagem c. Por meio de diagrama dos blocos também obtivemos resultado satisfatórios, visto que a planta respondeu corretamente as instruções contidas no diagrama. Deve-se observar que por meio de diagrama dos blocos é necessário a reprogramação para cada cenário diferente.

Figura 5. Vista da câmera do curumim transpassando obstáculos.

Em termos de *hardware*, os resultados foram relevantes, demonstrando o equilíbrio entre suas partes de comando/controle e potência. As rodas omnidirecionais em conjunto com combinação entre motores mostraram grande versatilidade, podendo andar em quase todos os sentidos e direções.

É importante observar cuidadosamente as portas de comunicação ao ligar a base de transmissão de dados no computador. Para endereçamento correto no *software*, visto que há possibilidade de troca de porta ao reconectar a base. Deve-se evitar o uso da plataforma para desviar de superfícies espelhadas ou com alto grau de refletância, uma vez que os sensores funcionam de forma irregular com os mesmos.

CONCLUSÕES

Inicialmente utilizando logica clássica e diagrama dos blocos para programação, obtivemos resultados satisfatórios, mas devido a complexibilidade dos futuros trabalhos estaremos utilizando a linguagem c, pelo maior suporte com modelos matemáticos e também maior interação em quesito *hardware/software.* Concluímos também que a plataforma atende as condições de mobilidade, comunicação e interação necessários para futuros trabalhos propostos. A partir dos resultados obtidos e avaliação do funcionamento dinâmico da plataforma. Novos projetos endereçam a aplicação de algoritmos baseados em técnicas computacionais inteligentes, como por exemplo, Lógica *Fuzzy*, Redes Neurais Artificiais, Mapas Cognitivos *Fuzzy* e/ou sistemas Neuro/*fuzzy*. Outra possível contribuição será a utilização da câmera do robô curumin para reconhecimento dinâmicos de objetos, como alvos, por exemplo, ou na construção de trajetória através da aquisição e tratamento de imagens.

AGRADECIMENTOS

Agradecemos a UTFPR pela oportunidade e bolsa PIBIC.

REFERÊNCIAS

[1] FIGUEIREDO, M. VII escola de informática da sbc. VII Escola de Informática da SBC, v .38, p. 74–106, 1999.
[2] FRACASSO, P. T., C. A. H. R. Navegação relativa de robótica moveis utilizando logica nebulosa com regras ponderadas. simpósio brasileiro de automação¸ ao inteligente,7.; ieee latin-american robotics symposium, v. 7, 2005.
[3] FIGUEIREDO, K. VELLASCO, M. P. M. S. F. Modelo neuro- fuzzy hierárquico politree com aprendizado por reforço para agentes inteligentes. SBA Controle & Automação, v. 18, p. 234–250, 2007.
[4] ZADEH, L. Fuzzy sets. Information and Control. , v. 8, p. 338–353, 1965.
[5] SMART, W. D.; KAELBLING, L. P. Reinforcement learning for robot control. MOBILE ROBOTS XVI PROCEEDINGS, v. 4573, p. 187–194, 2001.
COLEMAN, M.; GRAF, J.; PAINTER, P. Specific interactions and the miscibility of polymer blends. 1st. ed. CRC Press, 1995.