

## Desenvolvimento de rotinas para controle remoto de um robô autônomo baseado em Raspberry Pi

### Development of remote control routines for a Raspberry Pi based autonomous robot

#### RESUMO

A robótica é uma área que desde os anos 50 vem produzindo inúmeros trabalhos. Estes trabalhos abrangem diversas áreas de atuações, como na medicina, ramo industrial, segurança ou na agronomia. Sendo assim, é possível notar que a robótica trouxe inúmeros benefícios para os dias atuais. A partir disso, o grupo de pesquisa e estudos em Computação Visual Aplicada (COVAP), localizado no Laboratório de Computação Aplicada (LaCA) da UTFPR, campus Ponta Grossa, desenvolveu um robô móvel para a utilização em pesquisas relacionadas a área da robótica. Por conta disso, este trabalho tem como objetivo o desenvolvimento de uma biblioteca de controle para o robô construído.

**PALAVRAS-CHAVE:** Robótica, Robôs autônomos, Biblioteca de controle.

#### ABSTRACT

Robotics is an area that since the 1950s has been producing numerous works. These works cover several areas of expertise, such as medicine, industry, security or agronomy. Thus, it can be noted that robotics has brought numerous benefits to the present day. From this, the research and study group on Applied Visual Computing, located at the Applied Computing Laboratory of UTFPR, Ponta Grossa, developed a mobile robot for use in research related to robotics. Because of this, this work aims to develop a control library for the built robot.

**KEYWORDS:** Robotics, Autonomous robots, Control library.

#### INTRODUÇÃO

A robótica é uma área que vem sendo estudada desde o início dos anos 50, com a utilização de braços robóticos nas linhas de produção de empresas do ramo automobilístico (TZAFESTAS S. G., 2013). Além disso, com o desenvolvimento e estudo da robótica, foi possível criar soluções para diversos setores, como o militar, industrial ou da medicina (MURPHY, 2000). Exemplos de trabalhos realizados nestas áreas podem ser encontrados em Gupta et al. 2018, em que desenvolve de um robô antibomba e para controle de multidões e também em Yaghoubi et al. 2013, ao construir um robô para análise de solos e a utilização na agricultura de alta precisão.

Para que seja possível construir robôs que realizam a análise do ambiente, permitindo a tomada de decisões, é necessário que seja utilizado sensores que sejam sensíveis a determinados tipos de energia existentes no ambiente, como a luminosa, a térmica ou a sonora (WENDELING M, 2010). Com a utilização de

**Matheus Scaketti**

[mts.scaketti@gmail.com](mailto:mts.scaketti@gmail.com)

Universidade Tecnológica Federal do Paraná, Ponta Grossa, Paraná, Brasil

**Erikson Freitas de Moraes**

[emorais@utfpr.edu.br](mailto:emorais@utfpr.edu.br)

Universidade Tecnológica Federal do Paraná, Ponta Grossa, Paraná, Brasil

**Recebido:** 19 ago. 2019.

**Aprovado:** 01 out. 2019.

**Direito autorial:** Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



sensores ultrassônicos, é possível determinar a distância dos objetos que se encontram em sua frente. Além disso, a utilização de sensores *encoders* permite o conhecimento da distância percorrida por um motor. A partir do uso destes sensores, analisar as informações obtidas do ambiente se torna algo viável, fazendo com que o robô consiga realizar comandos de forma autônoma.

A partir de Murphy 2000 e com os trabalhos realizados em Gupta et al. 2018 e Yaghoubi et al. 2013, é possível notar o crescimento e a constância do estudo da robótica e o auxílio nas diversas situações do dia a dia no decorrer dos anos. Tendo em vista isso, o grupo de pesquisa e estudos em Computação Visual Aplicada (COVAP), localizado no Laboratório de Computação Aplicada (LaCA) da UTFPR, campus Ponta Grossa, desenvolveu um robô móvel para a utilização em pesquisas relacionadas a área da robótica. O robô foi construído a partir de um chassi de acrílico, possuindo quatro motores para sua movimentação e sensores *encoders* para a análise da distância percorrida do mesmo. Para seu controle, foi utilizado um microcomputador *Raspberry Pi 3B*, que possui um processador *Quad Core* com 1.2GHz e 1GB de memória RAM. Dado isso, este trabalho tem como objetivo o desenvolvimento de rotinas de controle do robô desenvolvido pelo grupo de pesquisa, para que a partir disso, seja possível a criação de novas pesquisas na área da robótica, facilitando a utilização do robô.

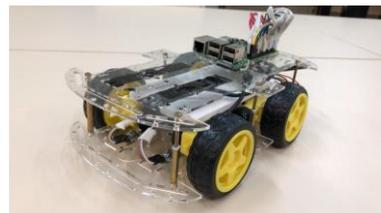
## METODOLOGIA

Nesta seção será apresentado as principais definições e materiais que foram utilizados para a realização deste trabalho, primeiramente com a especificação dos componentes do robô e o porquê da escolha de cada um e em seguida a biblioteca de controle criada para seu funcionamento.

## O ROBÔ

O robô utilizado neste trabalho foi desenvolvido pelo grupo COVAP e faz parte de um dos ramos de pesquisas que são realizados, o mesmo é apresentado na Figura 1. Foi construído utilizando componentes eletrônicos comuns que serão apresentados nesta seção.

Figura 1 – Robô móvel desenvolvido pelo grupo COVAP, com o intuito de desenvolver diversas pesquisas relacionadas a robótica.



Fonte: Autoria própria.

Para que seja possível controlar a distância percorrida por cada motor, são utilizados quatro sensores *encoders*, um para cada motor. Os sensores de

velocidade LM393, são utilizados em conjunto com diversos tipos de controladores e placas, como o Arduino, *Raspberry Pi* e PIC. Estes componentes são capazes de realizar medições de rotação de motores e controlam seu posicionamento. Sua tensão de operação é de 3,3 até 5V e possui um comparador LM393, assim como LEDs de indicação de tensão e saída digital. É comumente utilizado pelo fato de possuir facilidade de instalação e por ser constituído de uma saída digital e analógica. Além disso, foram utilizadas duas pontes H para o controle independente de cada motor.

## BIBLIOTECA DE CONTROLE

Para que seja possível a realização de diferentes tipos de pesquisa é importante que o robô possua rotinas com comandos previamente implementados, para que garanta o total foco do pesquisador na realização de sua pesquisa. Por conta disso, se tornou necessário a criação de uma biblioteca contendo diversas rotinas de comandos para o controle do robô.

Como o *Raspberry Pi* é um microcomputador e por isso, não possui um poder de processamento muito alto, devido à redução de custos para sua construção, é necessário que os programas não gerem muitas operações a serem executadas pelo processador. Tendo isso em vista, optou-se pela utilização da linguagem de programação *C++*, esta que teve início de sua criação por volta da década de 80, por Bjarne Stroustrup, tendo como objetivo melhorar uma versão do núcleo do Unix. Como esta linguagem é uma derivação da linguagem *C*, os códigos implementados tendem a serem mais otimizados em relação à outras linguagens, garantindo menos tempo de execução (SEABRA et al. 2018).

Com todas as ferramentas para a criação da biblioteca, foi definido quais operações seriam implementadas para atender o robô. A partir disso, optou-se pela criação das rotinas básicas de movimentação e rotação com um tempo definido. Também foi implementado as rotinas de movimentação a partir da distância percorrida ou ângulo para rotação.

Para as rotinas de rotação e movimentação por tempo, as funções recebem como entrada um caractere, representando o sentido a ser movimentado ou rotacionado, podendo ser frente ou para trás ou para a direita ou esquerda. Após isso, é chamado uma função para pausar a execução por um determinado período, recebido em seu parâmetro. Para que fosse possível a pausa do programa em segundos, multiplica-se o valor dado de entrada por 1000000. Sendo assim, caso a variável “tempo” seja igual a 2, ao realizar a multiplicação e passar o resultado como parâmetro desta função, o programa irá esperar dois segundos para dar sequência em sua execução. Assim como a rotina de movimentação, para que seja realizado uma rotação por tempo, basta trocar a função de movimentação.

Em seguida, foi desenvolvida as funções de movimentação e rotação por distância, cada iteração realizada dentro destas rotinas é uma chamada de função para cada motor que verifica se o sensor detectou uma alternância entre uma parede do disco e um segmento, contabilizando a distância

percorrida por um segmento até outro. Esta distância calculada é usada como critério de parada do loop, que é dividida por quatro, para realizar uma média da distância percorrida por cada motor. Assim como as rotinas por tempo, para realizar as operações de rotação, basta alterar a chamada de função de movimentação e adicionar um parâmetro que multiplica o valor da distância por 0.2, traduzindo o valor do ângulo para uma medida que o robô possa rotacionar o ângulo selecionado.

## TESTES E RESULTADOS

Nesta seção, será apresentado os resultados obtidos ao realizar testes das rotinas implementadas para a biblioteca. Para que fosse possível validar alguns pontos importantes, como distância percorrida, movimentação ou rotação correta, foi selecionado quatro testes distintos, cada um verificando determinada característica. Os testes que foram realizados são divididos em: teste de movimentação básico por tempo, teste de validação de distância percorrida 1 e 2 e por fim, teste de movimentação com distância e ângulo predefinidos.

### TESTE DE VALIDAÇÃO DE DISTÂNCIA PERCORRIDA 1 E 2

Em seguida, para verificar o funcionamento dos sensores *encoders*, utilizou-se da rotina de movimentação com distância definida. Sendo assim, o robô foi posicionado ao lado de uma régua para que fosse possível realizar uma medida aproximada da distância percorrida. Para este teste, definiu-se que o robô iria se movimentar 15cm a partir do seu ponto de origem.

Com a realização do teste, notou-se que a distância percorrida pelo robô ao final de cada execução era sempre maior do que foi definido na chamada da função, porém após diversas execuções do teste, percebeu-se que o erro gerado em todos os casos era aproximadamente igual. Tendo isso em mente, decidiu-se realizar o teste 50 vezes e anotar as distâncias finais de cada um, para verificar se este erro era constante.

Figura 2 – Gráfico de distâncias percorridas pelo robô. O robô deveria se movimentar em 15cm.



Fonte: Autoria própria.

Na Figura 2, é apresentado o gráfico com os dados obtidos ao realizar os testes. É interessante salientar que todos os dados estão dentro do intervalo de 19,6 a 20,6cm. Ao analisar o mesmo percebe-se que os dados se encontram muito próximos a medida de 20cm, sendo 5cm distante do objetivo. A média dos dados é igual a 20,15cm e os mesmos possuem um desvio padrão igual a 0,268cm, um valor relativamente baixo mesmo ao utilizar os sensores em conjunto com os motores DC, mostrando que eles são eficazes.

Em seguida, foi realizado um teste semelhante. O robô foi posicionado para se movimentar por 15cm, mas com paradas a cada 5cm.

Figura 3 – Gráfico de distâncias percorridas pelo robô. O robô deveria se movimentar em 15cm com paradas a cada 5cm.



Fonte: Autoria própria.

De forma semelhante ao teste anterior, o robô sempre se encontrava em uma distância maior em relação à distância desejada, mesmo se movimentando por distâncias pequenas. Ao analisar o gráfico obtido, nota-se que os dados estão mais dispersos em relação ao teste anterior, com um desvio padrão igual a 0,57cm e uma média de 26,85cm, gerando um erro de 11,85cm. Contudo, este erro só é maior e mais disperso pois a cada parada o robô gera um erro que é somado no decorrer do percurso, ou seja, a cada parada o robô gera em média 3,95cm de erro. Sendo assim, removeu-se o erro gerado por cada parada e realizou-se o teste novamente, resultando no gráfico apresentado na Figura 4.

Figura 4 – Gráfico de distâncias percorridas pelo robô. O robô deveria se movimentar em 15cm com paradas a cada 5cm.



Fonte: Autoria própria.

Após a tentativa de remoção do erro gerado por cada parada, notou-se que o robô manteve uma dispersão relativamente alta, semelhante ao teste com a inserção de erros. Contudo, apesar do desvio padrão encontrado ser igual a 0,62cm, a média da distância percorrida pelo robô se encontra relativamente próxima à distância desejada (15,4cm), mostrando que apesar de não garantir uma execução perfeita, o erro gerado é aceitável, visto que o robô não foi criado para realizar operações com uma precisão maior.

## CONCLUSÕES

Com a biblioteca implementada e a partir dos testes realizados, é possível verificar que com a utilização das rotinas, é possível controlar o robô de forma segura e não gerando problemas elétricos ou desligamentos repentinos. Além disso, com as rotinas criadas, será possível implementar algoritmos mais específicos dependendo da necessidade do pesquisador e a partir dos resultados analisados ao realizar o teste de movimentação por distância, nota-se que o robô pode se movimentar por uma distância predefinida com um erro relativamente baixo.

## REFERÊNCIAS

GUPTA, Varun; KUMBHARE, Amul; JAIN, Rita. Advanced Anti-Terrorism Unmanned Ground Vehicle. In: 2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS). IEEE, 2018. p. 1-5.

MURPHY, Robin; MURPHY, Robin R. Introduction to AI robotics. MIT press, 2000.

SEABRA, R. D.; DRUMMOND, I. N.; GOMES, F. C. ANÁLISE COMPARATIVA DE LINGUAGENS DE PROGRAMAÇÃO A PARTIR DE PROBLEMAS CLÁSSICOS DA COMPUTAÇÃO. Revista de Sistemas e Computação-RSC, v. 8, n. 1, 2018.

TZAFESTAS, Spyros G. Introduction to mobile robot control. Elsevier, 2013.

WENDELING, Marcelo. Sensores. 2010. Disponível em:  
<<http://www2.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendeling/4---sensores-v2.0.pdf>>.

YAGHOUBI, Sajjad et al. Autonomous robots for agricultural tasks and farm assignment and future trends in agro robots. International Journal of Mechanical and Mechatronics Engineering, v. 13, n. 3, p. 1-6, 2013.