

Controle de Movimentos de um Robô Móvel Aplicado a um Laboratório Remoto em AVA

Iago P. Gomes¹, Raphael C. dos S. Oliveira², Pablo S. da Silva², Jenifer de J. Jang², Cássio M. Silva², Gustavo L. de Oliveira², Roque M. P. Trindade³, Alzira F. Silva³, Maísa S. dos S. Lopes³

1. Estudante de Ciência da Computação - UESB; *iago.pg00@gmail.com
2. Estudante de Ciência da Computação - UESB;
3. Professor do Departamento de Ciência Exatas e Tecnológicas - UESB

Palavras Chave: controle de robôs móveis; PID; odometria

Resumo

O L1R2 é um robô móvel que faz parte do experimento de robótica móvel do LARA (Laboratório Remoto em Ambiente Virtual de Aprendizagem), que possui topologia diferencial. Esta topologia é uma das mais fáceis de se controlar, entretanto, conta com a desvantagem dos desvios laterais que aumentam proporcionalmente a distância percorrida pelo robô. Esta problemática não é desejada para o L1R2, pois dificulta sua utilização. Para minimizar esses desvios foram utilizados na biblioteca de programação do robô, os algoritmos de PID e odometria. Os resultados mostram que com determinados valores para as constantes do primeiro algoritmo, o desvio do robô diminuiu consideravelmente, entretanto nenhuma combinação o levou a zero. A abordagem para calibração do PID foi a empírica, que apesar de funcional é exaustiva, assim, o estudo de novas técnicas de escolha de constantes do PID podem melhorar a eficiência do controle aplicado.

Introdução

Os robôs móveis são uma classificação de robôs que distingue-se das demais por serem capazes de se locomover (ROMERO et al., 2014). Para tanto, segundo Siegwart, Nourbakhsh e Scaramuzza (2011) é preciso conhecer os parâmetros da definição cinemática e mecânica do veículo, uma vez que tais quesitos estão diretamente ligados a forma como esses robôs realizam suas manobras. O LARA (Laboratório Remoto em Ambiente Virtual de Aprendizagem) é um projeto da Universidade Estadual do Sudoeste da Bahia - UESB, que disponibiliza experimentos de robótica para auxiliar o ensino/aprendizagem de programação (LOPES et al., 2016).

O laboratório possui um experimento voltado à robótica móvel em que o aluno pode controlar pela internet o robô chamado de Lara Remote Robot (L1R2). Este possui uma cinemática diferencial, que é formada por duas rodas tracionadas, e uma roda livre, que garante a estabilidade do chassi. Conforme Muniandy e Muthusamy (2012) este modelo cinemático, é um dos mais fáceis de controlar, entretanto possui como desvantagem o desvio lateral que acentua-se com a distância percorrida pelo robô. Assim, movimentos como “frente” e “ré”, chamados de longitudinais, não são executados corretamente, ocorrendo, em alguns casos, um desvio tão grande que faz o robô traçar curvas ao invés do que fora pretendido (como retas, no caso das manobras citadas).

Esse evento não é desejado no experimento do LARA, pois acrescenta muito dificuldade nos desafios propostos aos alunos do laboratório. Desta forma, um dos principais requisitos funcionais do desenvolvimento do

L1R2, levantado na definição da arquitetura do projeto, é que o mesmo deve realizar as manobras corretamente, ou seja, caso seja solicitado que o robô ande para frente, este deve percorrer essa trajetória minimizando ao máximo os desvios laterais. Deste modo, como forma de prover tal funcionalidade, optou-se por utilizar os algoritmos de P.I.D (Proporcional, Integral e Derivativo) e a odometria do veículo, para corrigir os desvios em tempo de execução.

Este trabalho descreve a abordagem utilizada na biblioteca de controle do L1R2, para corrigir o problema dos desvios laterais nas manobras longitudinais desempenhadas pelo mesmo, em experimentos realizados pelo LARA.

Metodologia

O robô móvel do LARA, L1R2, possui uma topologia diferencial, que conta como desvantagem os desvios laterais que tornam-se mais significativos à medida que o robô desloca-se. Isso ocorre porque nesta topologia, cada roda é controlada independentemente, assim, mesmo que seja aplicado a mesma tensão, a velocidade resultante será diferente, e por consequência o robô tenderá para uma direção específica.

Como um dos requisitos funcionais classificados como essenciais no LARA é que o robô deve percorrer suas trajetórias corretamente, assim, foi necessário assumir uma abordagem que minimizem ao máximo as inconsistências provenientes de sua cinemática e mecânica. Por conta disso, optou-se por utilizar os algoritmos de P.I.D. e odometria.

O P.I.D ou controle Proporcional (P), Integral (I) e Derivativo (D), é um algoritmo amplamente usado no gerenciamento de processos que, segundo Novus (2003), faz com que o erro seja corrigido proporcionalmente ao seu aumento, bem como ter seu desvio característico de um controle P corrigido pela ação integral que age em função do tempo, e ter sua velocidade corretamente gerida pela função derivativa.

Por sua vez, a odometria é uma técnica de localização relativa de robôs móveis que determina a posição cartesiana e orientação destes, a partir de dados aferidos por sensores acoplados às suas rodas, geralmente usa-se encoders (BORENSTEIN et al. 1996). A Tabela 1 apresenta as principais equações odométricas para a topologia diferencial, e suas aplicações. Na tabela, ‘*d*’ representa o deslocamento do robô, ‘*d_{left/right}*’ a distância percorrida para cada roda, e ‘*d_{baseline}*’ a distância entre os pontos de contato das rodas.

Conforme Lee, Chung e Yoo (2010), quando os encoders são usados para prover informações para a odometria, utiliza-se as seguintes equações para calcular

Tabela 1. Equações Odométricas.

Equação	Aplicação
$d = \frac{d_{left} + d_{right}}{2}$	calcula o deslocamento do robô
$\phi = \frac{d_{right} - d_{left}}{d_{baseline}}$	calcula a orientação momentânea do robô
$\theta' = \theta + \phi$	calcula a orientação do robô
$x' = x + d \cos(\theta)$	calcula a posição x do robô
$y' = y + d \sin(\theta)$	calcula a posição y do robô

Fonte: o autor.

o deslocamento individual das rodas (*d_{left}* para a esquerda, e *d_{right}* para a direita):

$$d_{left} = \frac{P_{left} N_{left}}{Re_{left}} \quad (1)$$

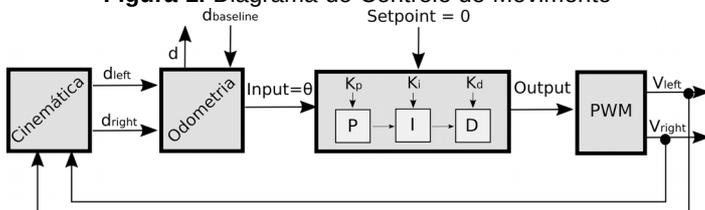
$$d_{right} = \frac{P_{right} N_{right}}{Re_{right}} \quad (2)$$

Onde, *P_{left/right}* é o perímetro das rodas, *N_{left/right}* é a quantidade de leituras realizadas pelos encoders em um intervalo de tempo, e *Re_{left/right}* é a resolução dos sensores por uma revolução completa das rodas.

Assim, na abordagem assumida para o controle do L1R2, a distância *d* é usada nas funções 'frente' e 're' para controlar o tanto que o robô deve deslocar-se, uma vez que os usuários têm a opção de determinar a distância em centímetros para os deslocamentos longitudinais.

Para minimizar os desvios laterais, a angulação θ é usada como entrada do controle PID. Neste controle o 'setpoint', ou seja, o valor desejado para a entrada do sistema é configurado como 0, assim, sempre que θ for diferente deste valor, um valor de correção é calculado pelo algoritmo e usado para alterar as velocidades das rodas (*V_{left}* para a esquerda, e *V_{right}* para a direita), ajustando a manobra do robô. O valor de θ é zerado a cada chamada das funções. A Figura 1, apresenta o diagrama do controle aplicado, sendo possível notar a forma com que a orientação do veículo é usado como entrada do controle e como a saída ajusta as velocidades das rodas. Como as velocidades são determinadas por um sinal PWM (*Pulse Width Modulation* - Modulação de Largura de Pulso), então uma função de conversão é utilizada para garantir que o valor da velocidade esteja nos limites corretos (no caso do microcontrolador arduino, utilizado no L1R2, o limite é 0-255).

Figura 1. Diagrama do Controle de Movimento



Fonte: o autor.

Resultados e Discussão

Para verificar a eficiência do controle implementado, foram realizados repetições dos movimentos longitudinais, e conferindo a orientação final do veículo calculada pela odometria. Desta forma é possível notar o quanto de erro ainda encontrava-se presente no robô após a realização da manobra.

Assim, utilizou-se a função 'frente' para fazer o robô deslocar-se por 1 metro. Este teste foi executado 10 vezes, em cada execução, as constantes do PID (*K_p*, *K_i* e *K_d*) foram ajustadas para melhorar o resultado final. A Tabela 2 apresenta os dados colhidos. É importante salientar que os valores escolhidos para as constantes foram definidos empiricamente, baseando-se apenas nas definições de cada unidade do sistema (Proporcional, Integral e Derivativo).

Tabela 2. Dados obtidos pela função 'frente'.

	1ª	2ª	3ª	4ª	5ª	6ª	7ª	8ª	9ª	10ª
K_p	10	50	20	5	10	10	50	50	50	50
K_i	0	0	0	0	5	10	10	50	50	10
K_d	0	0	0	0	0	0	0	0	0.5	0.05
θ (°)	4	-0.57	1.72	-2.86	-7.45	-6.88	-1.15	2.3	0.56	1.15
V_{left}	200	200	206	200	209	200	200	218	200	200
V_{right}	200	200	200	200	200	206	206	200	204	200

Fonte: o autor.

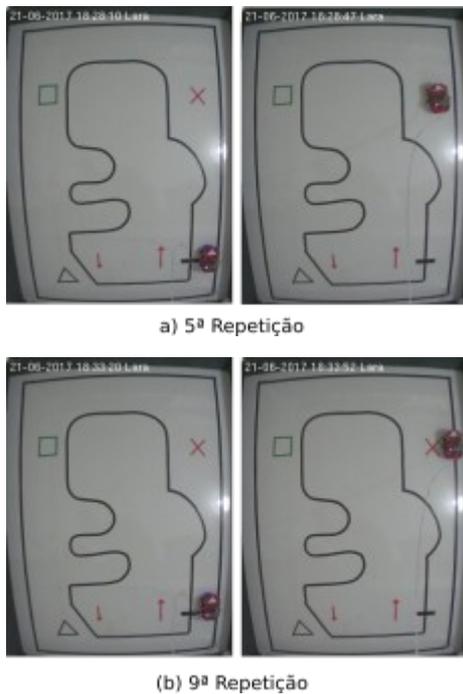
É possível notar que ao ajustar os valores de *K_p*, *K_i* e *K_d*, o desvio lateral diminuiu ou aumentou, a depender da combinação que foi usada. Isso só enfatiza a importância de uma boa escolha de valores para melhorar a eficiência do controle, como ocorreu na 2ª e 9ª repetição, cujos desvios foram os menores entre os obtidos. Na 2ª repetição, foi aplicado um controle apenas proporcional (P) ao erro, em que as velocidades das rodas são alteradas apenas com base na diferença entre a orientação do robô e o setpoint do algoritmo. Já na 9ª repetição, foi utilizado um controle PID completo. Entretanto, não foi possível definir valores que levassem o erro a 0, assim, é necessário encontrar outro meio para calibração do PID, uma vez que a escolha empírica mostrou-se exaustiva.

Além disso, os ruídos presentes nos parâmetros cinemáticos do robô (como o diâmetro absoluto das rodas, ou seja, o diâmetro real de cada roda, que podem ser diferentes entre si), usados nas equações cinemáticas para calcular a orientação e deslocamento do referido, bem como as contrapartidas da odometria (deslize de uma roda, por exemplo) aumentam a imprecisão no resultado final.

Na Figura 2 são apresentado a posições resultantes dos testes 5 e 9 (antes do movimento e depois do movimento), após a execução do programa. Em Fig. 2a, é possível notar o tamanho do desvio com o controle PI do teste 5, e em Fig. 2b resultado do controle PID. É perceptível que para o laboratório remoto não é permitido que o robô atue como em 5, pois isso dificultaria a manipulação do mesmo, e, conseqüentemente, na realização das atividades. Já no caso do controle da 9ª repetição, apesar do erro não ter sido zerado, ainda

assim, o desvio é muito pequeno, por hora podendo ser considerado aceitável.

Figura 2: Posição do Robô nos Testes 5 e 9. (a) posição no teste 5; (b) posição no teste 9.



Fonte: o autor.

Conclusões

Por mais que a topologia diferencial seja uma das mais fáceis de se utilizar, a principal desvantagem são os desvios laterais que aumentam conforme o robô desloca. Esta problemática não é aceitável para o L1R2, pois dificulta sua utilização. Por conta disso utilizou-se na biblioteca de controle do robô os algoritmos de P.I.D. e odometria para corrigir esses desvios durante a execução de movimentos longitudinais pelo robô.

Esta abordagem apresentou resultados aceitáveis, pois reduziu os desvios a valores pequenos, apesar de não tê-lo zerado. A utilização do PID é bastante comum quando objetiva-se controle de processos, entretanto a escolha de suas constantes modificam bastante os resultados. A escolha empírica, pode surtir efeito, entretanto é muito exaustiva, por conta disso, utilizar um método de calibração como os citados em LOURENÇO (1997) facilitará a definição desses valores, melhorando os resultados finais.

Outros fatores relevantes aos resultados são a cinemática do veículo, que pode apresentar algumas diferenças, como no diâmetro das rodas, e os erros odométricos, principalmente quando usa-se os encoders, por exemplo, quando uma roda desliza, assim o sensor não realiza nenhuma leitura apesar do deslocamento. Borenstein et al. (1996) apresenta um teste chamado *UMBmark test*, que realiza a calibração de alguns parâmetros cinemáticos do robô, entre eles o diâmetro das rodas, para diminuir a influência destes na estimação odométrica. Certamente, quando convier este teste será realizado para diminuir ainda mais o desvio do robô, e aumentar a precisão da posições e orientação do robô utilizando as equações odométricas.

Referências bibliográficas

- BORENSTEIN, J. et al. Where am I? Sensors and methods for mobile robot positioning. University of Michigan, 1996. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.444.5014&rep=rep1&type=pdf>>.
- LOPES, M. et al. Web environment for programming and control of mobile robot in a remote laboratory. IEEE Transactions on Learning Technologies, IEEE, 2016. Disponível em: <<http://ieeexplore.ieee.org/document/7740908/>>.
- LOURENÇO, João. "Sintonia de controladores PID." *Escola superior de tecnologia*. 1997. Disponível em: <<http://todi.est.ips.pt/smarques/CS/Pid.pdf>>
- MUNIANDY, M.; MUTHUSAMY, K. An innovative design to improve systematic odometry error in non-holonomic wheeled mobile robots. *Procedia Engineering*, Elsevier, v. 41, p. 436–442, 2012. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877705812025957>>.
- NOVUS. Controle PID básico. 2003. Disponível em: <<http://www.novus.com.br/downloads/Arquivos/artigopidbasi.co.pdf>>.
- ROMERO, R. A. F. et al. *Robótica Móvel*. 1st edition. ed. Rio de Janeiro: LTC, 2014.
- SIEGWART, R.; NOURBAKHSI, I. R.; SCARAMUZZA, D. *Introduction to autonomous mobile robots*. [S.l.]: MIT press, 2011.