

Faculdade de Engenharia da Universidade do Porto



Odometria e equações de movimento de um robô

Sandro Augusto Costa Magalhães
Tiago José Ferreira Mendonça

UP201304932
UP201305394

Relatório realizado no âmbito da unidade curricular de Sistemas Robóticos
Autónomos

Docentes: Vítor Pinto e António Moreira

28 de setembro de 2017

1. Introdução

Este relatório surge no âmbito do trabalho prático para a unidade curricular de Sistemas Robóticos Autónomos, realizado em sala de aula no dia 27 de setembro de 2017, que teve como principal intuito incentivar os estudantes a estudar o comportamento de uma robot de tração diferencial com duas rodas e calcular alguns parâmetros das características inerentes.

Como ambiente de trabalho para a realização do exercício, recorreu-se ao simulador *SimTwo* que disponibiliza um modelo do robot supra indicado. Este desloca-se num plano horizontal intermédio de duas rodas motrizes diferenciais e com uma terceira roda livre de apoio.

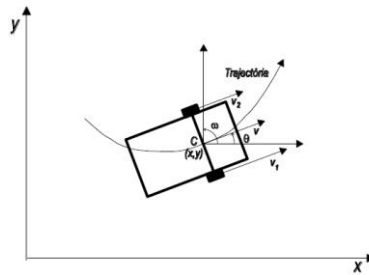


Figura 1 Esquemática de um robot de tração diferencial com duas rodas (o efeito da roda livre é menosprezado nos cálculos)

Como ponto de partida, teve-se em consideração as equações base que permitem o controlo da velocidade linear (v) e velocidade angular (ω), por controlo da velocidade dos motores de cada roda (v_1 e v_2).

$$v = \frac{v_1 + v_2}{2} \quad (1)$$

$$\omega = \frac{v_1 - v_2}{b} \quad (2)$$

2. Determinação da constante K_{imp}

Neste ponto, pretendeu-se determinar a constante K_{imp} que corresponde à relação entre os impulsos gerados pelos sensores nas rodas e o deslocamento linear de cada roda.

Para tal, começou-se por analisar o código já pré-implementado no *SimTwo* e constatou-se que este permitia um deslocamento linear do robot por pressão das teclas *up* e *down* do teclado, uma vez que estas estavam pré-configuradas com igual velocidade em cada motor, eliminando a velocidade angular (como se comprova pela equação 1).

Primeiramente, sentiu-se a necessidade de implementar dois acumuladores que permitem contar o número de impulsos dos sensores de cada motor.

```
Odometro1 := Odometro1+odo1;  
Odometro2 := Odometro2+odo2;  
  
SetRCValue(1,2, format('%.3g', [Odometro1]));  
SetRCValue(2,2, format('%.3g', [Odometro2]));
```

Ambas as variáveis, *Odometro1* e *Odometro2*, foram declaradas como do tipo inteiras e globais.

Uma vez obtido o valor acumulado dos odômetros, é possível obter o valor da constante K_{imp} .

$$d = \frac{odo_1 + odo_2}{2} \cdot K_{imp} \quad (3)$$

Por análise dos dados obtidos do *SimTwo*, sabemos que o robot, neste caso, percorreu linearmente $0,452\text{ m}$ e que cada sensor dos motores contou 978 e 985 impulsos, respetivamente. Assim, resolvendo a equação acima em ordem a K_{imp} , temos que $K_{imp} = 0,000462 \approx 0,0005\text{ m/impulso}$.

3. Cálculo da distância entre rodas

Para o cálculo da distância entre rodas (b), procedeu-se de forma similar à situação anterior. Novamente, por análise do código pré-implementado, constatou-se que a pré-configuração das teclas *left* e *right* do teclado possibilitava ao robot realizar apenas o movimento angular, uma vez que $v_1 = -v_2$, anulando a equação 1.

Sabe-se que o movimento angular do robot é dado, em radianos, por

$$\theta = \frac{odo_1 - odo_2}{b} \cdot K_{imp} \quad (4)$$

Mais uma vez, aproveitando o acumulador anteriormente implementado, verificou-se que cada sensor contou -206 e 207 impulsos, respetivamente. Constatou-se que o robot rodou 123° , isto é $\theta = 2,1468\text{ rad}$. Então, resolvendo a equação 4 em ordem a b , considerando $K_{imp} = 0,000462\text{ m/impulso}$, conclui-se que $d = 0,088664 \approx 0,1\text{ m}$.

4. Cálculo simultâneo de K_{imp} e da distância entre rodas

Neste momento, o objetivo consistia na realização de um único movimento composto, incluindo rotação e translação, através do qual é possível determinar a constante K_{imp} bem como a distância entre rodas (b). Para o efeito, procedeu-se à alteração do código previamente implementado, alterando a velocidade associada a cada uma das rodas nas instruções *up* e *down*. Assim, configurou-se uma razão entre velocidades de $\frac{v_1}{v_2} = 2$, de forma a que o robot realizasse um movimento circular. Seguidamente, partindo da origem do referencial, forçou-se o mesmo a descrever uma trajetória equivalente a um deslocamento angular de 180° , isto é, $\pi\text{ rad}$, registando-se os seguintes valores:

$$x = -0.326\text{ m}; y = -0.050\text{ m}; \Delta\theta = 180^\circ$$

$$Odometro_1 = -1290\text{ impulsos}; Odometro_2 = -669\text{ impulsos}$$

Sendo a trajetória circular e o deslocamento angular de 180° , imediatamente se infere que o deslocamento linear será dado pela metade do perímetro de uma circunferência de diâmetro igual à distância entre a posição final e a origem. Sendo assim:

$$d = \pi \times \frac{\sqrt{(-0.326)^2 + (-0.050)^2}}{2} \approx 0.518\text{ m}$$

$$\theta = \pi\text{ rad}$$

Com estes dados e substituindo-os nas equações inicialmente apresentadas, obtém-se $K_{imp} = 0.000529 \approx 0.0005 \text{ m/impulso}$ e $b = 0.104 \approx 0.1 \text{ m}$ (com $K_{imp} = 0.000529$), valores consistentes face aos resultados das alíneas anteriores.

5. Implementação das Equações de Movimento

Nesta etapa, o propósito residia na implementação de um procedimento adicional capaz de estimar as coordenadas (x, y, θ) da posição do robot no referencial. Nesse sentido, recorreu-se ao método de discretização por diferenças centradas, regido pelas seguintes equações:

$$x(i+1) = x(i) + d(i) \cdot \cos(\theta(i) + \frac{\Delta\theta(i)}{2}) \quad (5)$$

$$y(i+1) = y(i) + d(i) \cdot \sin(\theta(i) + \frac{\Delta\theta(i)}{2}) \quad (6)$$

$$\theta(i+1) = \theta(i) + \Delta\theta(i) \quad (7)$$

Para a sua implementação, concebeu-se o seguinte procedimento que recebe como argumentos os valores lidos, em cada iteração, do odómetro no *encoder* de cada uma das rodas:

```
procedure alineaD(odo2, odo1:double);

var
  d_at:double;
  delta_theta:double;

begin
  d_at := (Odo1 + Odo2)/2*0.0005;
  delta_theta := (Odo1 - Odo2)*0.0005/0.1;
  theta := theta + delta_theta;

  if theta > (2*3.14159265359) then begin
    theta := theta - 2*3.14159265359;
  end;

  if theta < (-2*3.14159265359) then begin
    theta := theta + 2*3.14159265359;
  end;

  x_atual := x_atual + d_at*cos(theta + delta_theta/2);
  y_atual := y_atual + d_at*sin(theta + delta_theta/2);

  SetRCValue(1,3, format('%.3g', [x_atual]));
  SetRCValue(2,3, format('%.3g', [y_atual]));
  SetRCValue(3,3, format('%.3g', [theta*180/3.14159265359]));
end;
```

6. Testes

Para aferir o correto funcionamento do procedimento, realizaram-se alguns movimentos com o robot e comparou-se a posição indicada pelo simulador com a estimada. Inicialmente, constatou-se uma elevada similaridade entre as coordenadas, sendo o erro praticamente nulo. Contudo, aquando da realização de movimentos mais bruscos (acelerações e paragens repentinas), era perceptível uma crescente divergência entre os valores. Esta divergência tornava-se ainda mais clara quando se colidia o robot com a barreira existente no ambiente de simulação, uma vez que a posição estimada do robot crescia continuamente, em termos absolutos, enquanto que a sua posição real se mantinha constante, reproduzindo-se num aumento gradual do erro acumulado. Este efeito está relacionado com uma das limitações associadas à estimação por odometria, pois assume-se que todo o movimento rotacional das rodas é convertido em deslocamento linear, o que não é assegurado em todas as circunstâncias. Em determinadas condições existe deslizamento entre as rodas e o solo, gerando erros de estimação, tal como os observados no presente exemplo.