

Faculdade de Engenharia da Universidade do Porto



**Implementação de rotinas básicas de controlo do
movimento de um robô (GoToXYTheta)**

Sandro Augusto Costa Magalhães
Tiago José Ferreira Mendonça

VERSÃO FINAL

Trabalho realizada no âmbito da unidade curricular de
Sistemas Robóticos Autónomos

Docentes: Vitor Pinto e António Moreira

14 de Outubro de 2017

1 Introdução

Com este trabalho, pretende-se implementar um algoritmo básico que seja capaz de transportar um robô móvel de tracção diferencial até uma determinada pose final, tipicamente designado por *Go To XY Theta*. O movimento de um robô de uma determinada pose para outra é facilmente feito em três passos:

1. Rodar o robô para a posição final
2. Seguir até à posição final
3. Rodar o robô com o ângulo final

2 Cálculo da velocidade de cada roda

Dada a natureza do robô, a velocidade linear e angular final deste tem de ser traduzida em equivalentes velocidades para cada roda. As equações 1 e 2 calculam a velocidade v_1 e v_2 de cada roda, assumindo b como metade da distância entre rodas.

$$v_1 = v + \omega \cdot b \quad (1)$$

$$v_2 = v - \omega \cdot b \quad (2)$$

Assim sendo, criou-se um procedimento *VelocidadeRodas* responsável por calcular a velocidade de cada roda.

```
procedure VelocidadeRodas (v, vw: double);
var
  v1, v2, b: double;

begin
  b := 0.1;

  v1 := v + vw*(b/2);
  v2 := v - vw*(b/2);

  SetRCValue(12, 1, 'v1');
  SetRCValue(13, 1, 'v2');
  SetRCValue(12, 2, format('%.3g', [v1]));
  SetRCValue(13, 2, format('%.3g', [v2]));

end;
```

Note-se que no procedimento implementado, a b corresponde a toda a distância entre as rodas, como tal, sente-se a necessidade de dividi-lo por dois.

3 Implementação do procedimento *Go To XY Theta*

Para a implementação do algoritmo desejado, sumariamente descrito na secção 1, recorreu-se à máquina de estados representada na figura 3.1. É de notar que em relação ao anteriormente afirmado, sentiu-se a necessidade de implementar dois estados extra que têm como principal objectivo actuar na proximidade à posição final, sendo que nestes momentos o robô move-se a uma velocidade inferior à nominal.

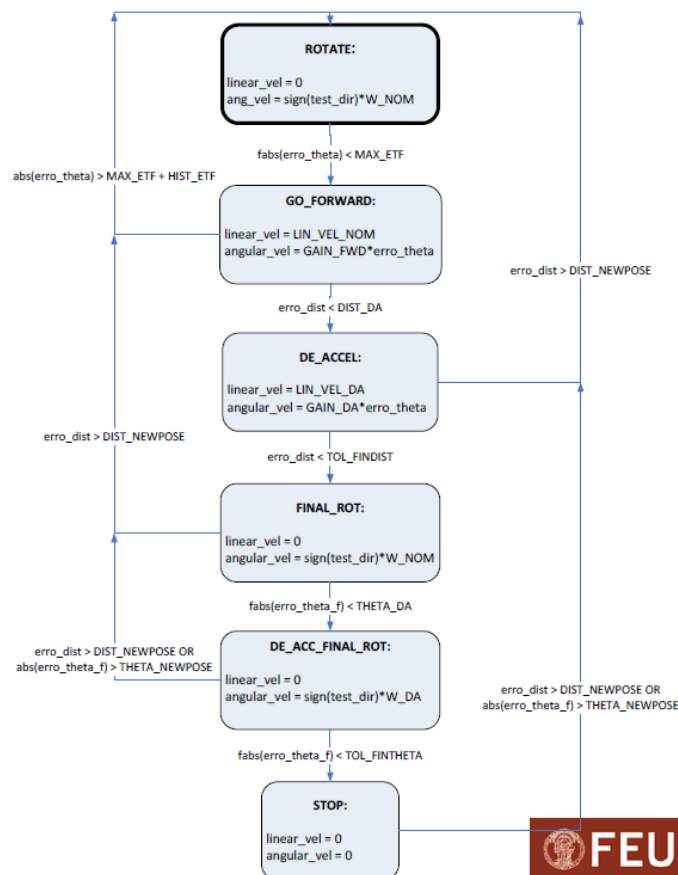


Figura 3.1 Máquina de estados que modela o procedimento *Go To XY Theta* (figura retirada da página 52 do conjunto slides Cinemática Dinâmica)

Na implementação da máquina de estados, e uma vez que estamos numa situação de simulação onde os componentes são mais precisos e possuem menos ruído, introduziu-se exigiu-se uma elevada precisão ao sistema. Para atingir esta precisão, modelou-se as variáveis referentes às tolerâncias do sistema, tal como demonstrado na tabela 3-I.

Procedeu-se de forma semelhante para os ganhos e velocidades nominais do sistema (tabela 3-II) e para os erros máximos admitidos nos estados intermédios (tabela 3-III).

Nome da variável	Tolerância	Tolerância máxima admitida
TOL_FINTHETA	Tolerância em relação ao valor final de theta	0.01
TOL_FINDIST	Tolerância em relação à posição final objectivo	0.05

Tabela 3-I Tolerâncias máximas admitidas

Nome da variável	Descrição	Valor
LIN_VEL_NOM ou VNOM	Velocidade linear nominal	10
W_NOM ou WNOM	Velocidade angular nominal	20
GAIN_FWD	Ganho do erro angular em relação à posição que quer atingir que modela a velocidade angular quando o robô percorre à velocidade nominal	-0.5
GAIN_DA	Ganho do erro angular em relação à posição que quer atingir que modela a velocidade angular quando o robô percorre à velocidade de desaceleração	0.05
V_DA	Velocidade linear de desaceleração	0.25
W_DA	Velocidade angular de desaceleração	5

Tabela 3-II Velocidade e ganhos das velocidades

Nome da variável	Tolerância	Tolerância máxima admitida
MAX ETF	Tolerância máxima do erro angular em relação à posição final	0.1
HIST ETF	Histerese de MAX ETF	0.3
DIST_DA	Distância a partir da qual o robô entra no estado de desaceleração linear	0.1
THETA_DA	Erro angular a partir do qual o robô entra no estado de desaceleração angular	0.1
DIST_NEWPOSE	Distância à posição final	0.21
THETA_NEWPOSE	Distância ao theta final	0.15

Tabela 3-III Erros máximos admitidos dos estados de transição

O código, referente à máquina de estados implementada, segue num ficheiro de texto em anexo.

De seguida, apresenta-se um exemplo de execução do procedimento. A título de exemplo, escolheram-se duas posições arbitrárias para demonstrar o seu correto funcionamento. Assim, como posição inicial considerou-se $(x, y, \theta) = (-0.5, -1, -1.54)$ (ângulo em radianos) e como posição final $(x, y, \theta) = (0.5, 0, 0)$. Tal como pretendido, num primeiro instante o robot orienta-se de forma a ficar direcionado com posição final (2), percorre uma trajetória retilínea até atingir as coordenadas (x, y) que lhe estão associadas e, por último, reorienta-se de acordo com o ângulo indicado (3).

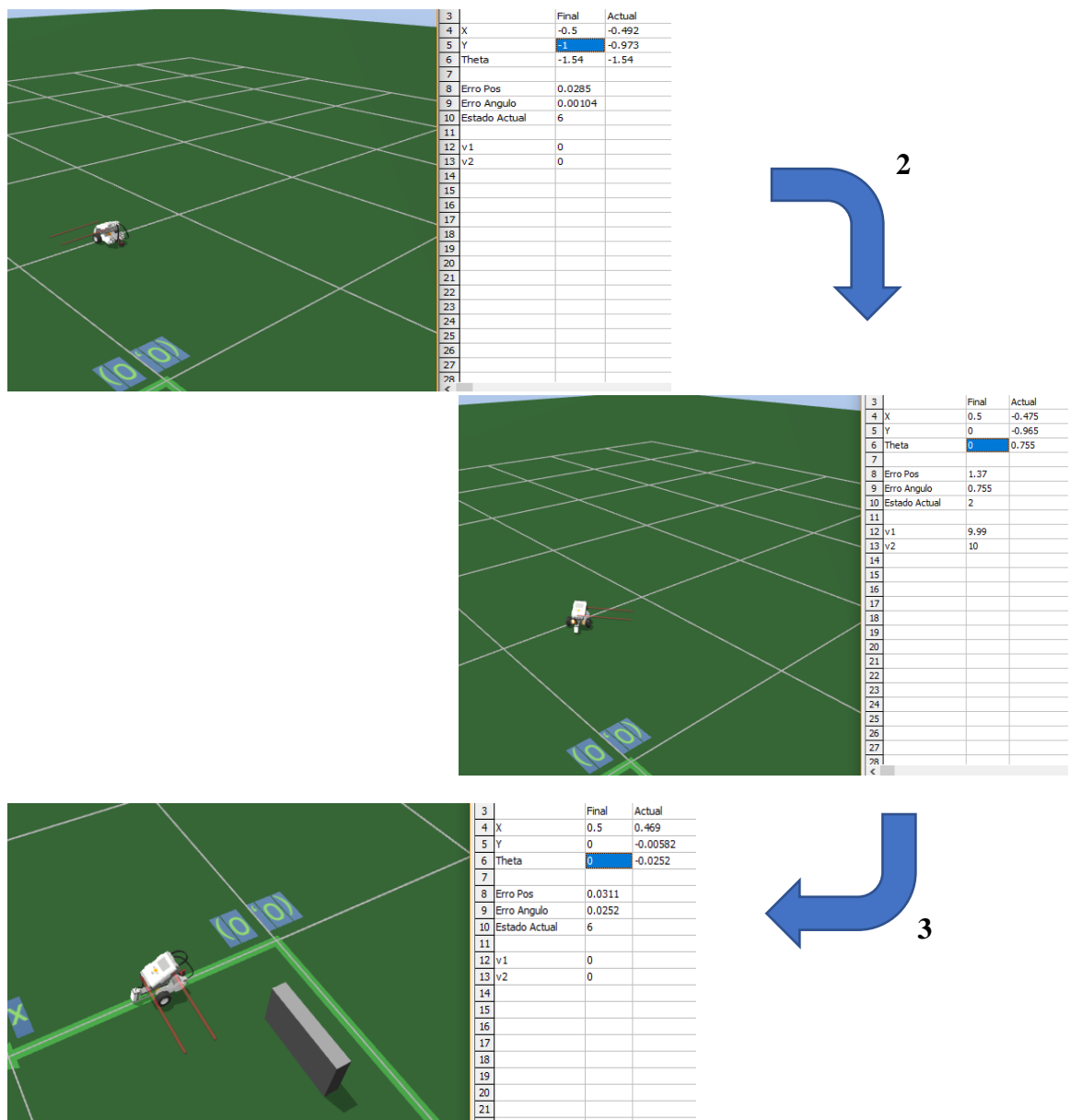


Figura 3.2 – Sequência de movimentos do Robot na deslocação entre as posições inicial e final

4 Limitação Taxas Máximas de Aceleração/Desaceleração Angular e Linear

Neste momento, pretendia-se a limitação da taxa de variação de velocidade de forma a que esta não fosse instantânea, mas estivesse sujeita a um valor máximo. Assim, para reproduzir este comportamento, definiram-se duas novas variáveis (DVMAX e DWMAX) que representariam, respetivamente, o limite imposto em termos lineares e angulares. Desta forma, nos diferentes estados do procedimento de controlo, em vez de se atribuir um valor imediato à velocidade, o conteúdo atual da mesma seria incrementado ou decrementado (conforme se trata de uma aceleração ou desaceleração, respetivamente), em termos absolutos, por uma constante de forma iterativa, até se atingir a velocidade pretendida. Constatou-se que uma menor taxa de variação implica uma maior distância de travagem por parte do robot, implicando, por isso, um maior número de movimentos, quando comparado com uma situação de variação momentânea, até se obter uma posição final de acordo com as tolerâncias estabelecidas. Após várias simulações, considerou-se razoável a estipulação dos seguintes valores: DVMAX = 2 e DWMAX = 8.

De seguida, apresenta-se um exemplo de execução do procedimento de acordo com os parâmetros indicados. Neste caso, partindo de uma posição inicial $(x, y, \theta) = (0,0,0)$, pretendia-se que o robot atingisse a posição final $(x, y, \theta) = (-0.5, -0.5, 0)$. Num primeiro instante, o robot rodou sobre si próprio de forma a ficar direcionado para a localização da sua posição final (2). Posteriormente, percorreu uma trajetória retilínea até à posição final, onde novamente rodou sobre si próprio, localizando-se de acordo com a orientação pretendida (3).

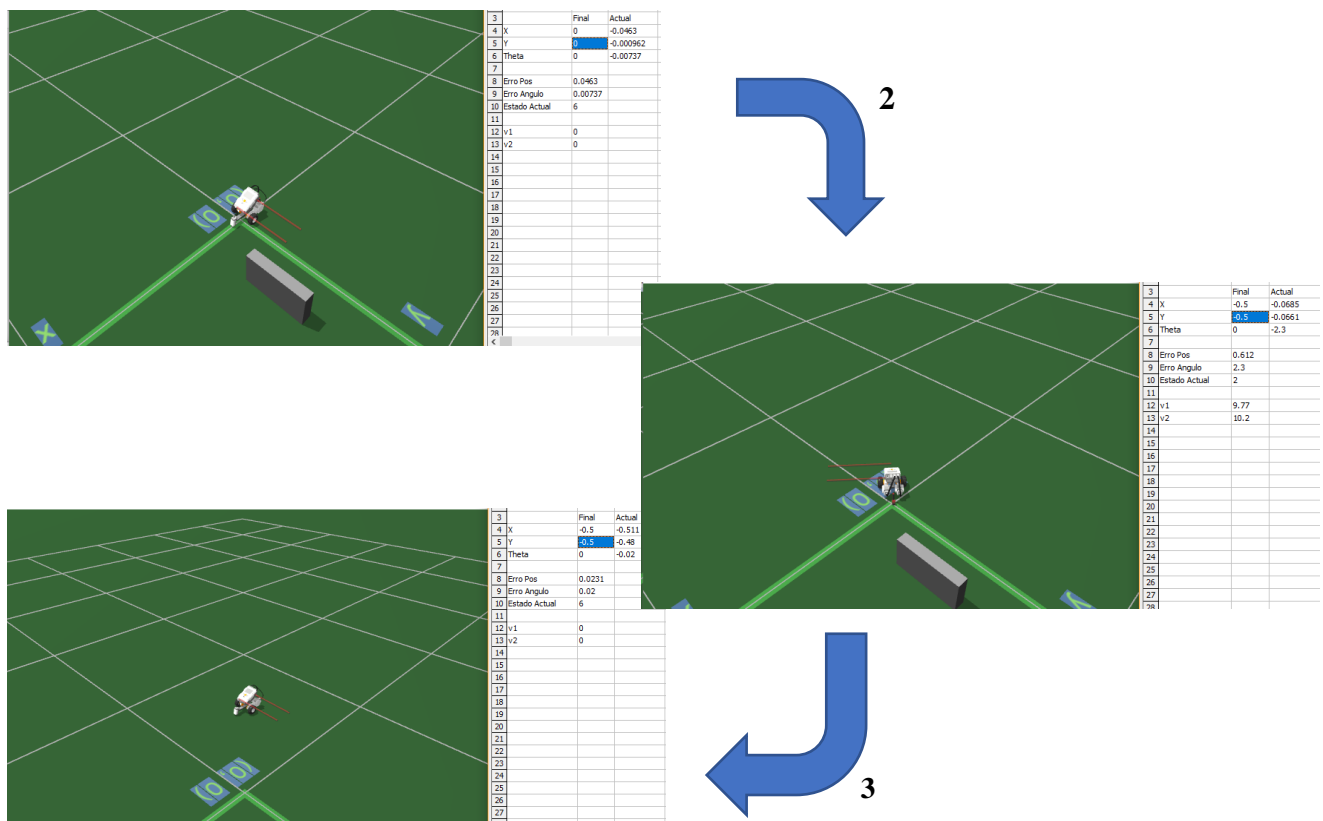


Figura 4.1 – Sequência de movimentos do Robot na deslocação entre as posições inicial e final com taxas máximas de aceleração/desaceleração