

CONTROLO E ESTIMAÇÃO DO POSICIONAMENTO ABSOLUTO DE UM ROBOT OMNIDIRECCIONAL DE TRÊS RODAS

RESUMO

Este artigo aborda o tema Controlo de *Robots* Omnidireccionais. O *robot* usado como caso de estudo utiliza motores DC com *encoders* e possui 3 rodas especiais que permitem a sua omnidireccionalidade. O movimento destes *robots* não está sujeito às restrições dos *robots* mais usuais que utilizam apenas duas rodas normais, tendo como desvantagem o seu controlo ser mais complexo.

O principal objectivo deste trabalho foi o de desenvolver um controlador que permita o seguimento de trajectórias pré-definidas com velocidade controlada, estimando-se o posicionamento absoluto do *robot* baseado na odometria.

Os problemas de controlo em robótica incluem inevitáveis requisitos de tempo real muito apertados. Nesta situação particular, o cálculo da odometria e do controlo para um *robot* omnidireccional é computacionalmente muito mais pesado que para os tipicamente usados *robots* diferenciais. Deste modo é necessário ter um cuidado especial ao implementar as operações matemáticas necessárias pois o código deve ser optimizado com vista a cumprir requisitos de tempo real.

Como motivação para a utilização de *robots* omnidireccionais é apresentada a comparação desta configuração com a mais típica configuração diferencial.

1. INTRODUÇÃO

Este artigo descreve o controlo e posicionamento de um *robot* omnidireccional. Mecanicamente o *robot* possui três rodas motrizes omnidireccionais desfasadas de 120° (figura 1), permitindo movimentos em todas as direcções com velocidade controlada. O sistema de accionamento é constituído por três motores de corrente contínua com caixa redutora. Para se efectuar o controlo da velocidade de cada uma das rodas é usada modulação de largura de impulsos (PWM), controlando-se independentemente cada uma das rodas. O controlo dos motores é feito em malha fechada utilizando um micro-controlador AVR programado em C e um PC com uma aplicação em Delphi, comunicando entre si usando a norma RS232.

O odometria e o controlador são calculados no PC comunicando este com o micro-controlador. O micro-controlador envia para o PC as transições dos *encoders* relativas a cada roda, com estes dados é estimada a posição absoluta do *robot*. Após ser calculado o posicionamento baseado na odometria é calculado o sinal de controlo. O controlador tem como parâmetros de saída a velocidade a que devem rodar os motores para que o *robot* alcance um determinado objectivo. O diagrama de blocos do sistema está representado na figura 2.

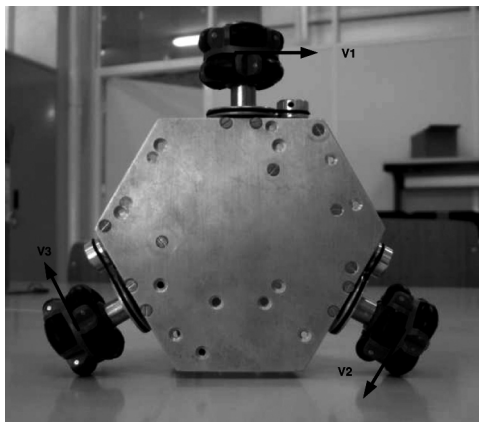


Figura 1 Vista de baixo do robot desenvolvido

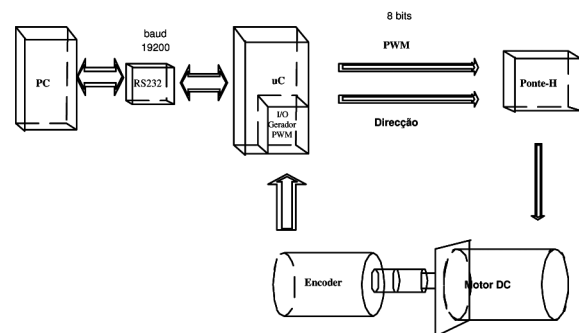


Figura 2 Diagrama de blocos do sistema

2. ROBOT OMNIDIRECCIONAL VS ROBOT DIFERENCIAL

O *robot* diferencial é provavelmente o *robot* móvel mais usado, sendo apresentado na figura 3. O *robot* diferencial é composto por duas rodas motrizes, cujos veios passam pelo mesmo eixo. O seu movimento é controlado variando independentemente a velocidade de cada uma das rodas.

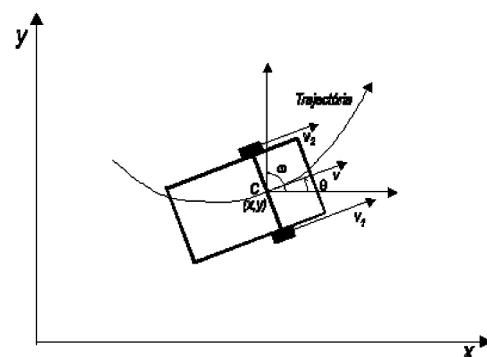


Figura 3 Robot diferencial

A estrutura do *robot* diferencial, representada na figura 3, impede que sejam feitos movimentos de translação segundo o eixo que passa pelos veios dos motores (Dudek and Jenkin, 2000). Considerando que não existe escorregamento lateral, isto é, que a velocidade das rodas no ponto de contacto com o chão seja sempre perpendicular ao eixo que passa pelas mesmas, obtemos o vector de estado representado pela equação (1).

$$X(t)^T = (x(t) \ y(t) \ \theta(t) \ v(t) \ w(t)) \quad (1)$$

Na equação (1) $x(t)$, $y(t)$ e $\theta(t)$ representam a posição do ponto C no plano e $w(t)$ representa a velocidade angular (velocidade de rotação do *robot* segundo o eixo vertical que passa por C). Uma outra possibilidade para a escolha das variáveis de estado seria a utilização da seguinte equação:

$$X(t)^T = (x(t) \ y(t) \ \theta(t) \ V_1(t) \ V_2(t)) \quad (2)$$

Neste caso $V_1(t)$ e $V_2(t)$ são velocidades medidas do ponto de contacto entre o chão e as rodas. Existem estas duas representações possíveis podendo-se passar de uma para outra usando a equação (3) e a equação (4). Na equação (4) b representa a distância entre os pontos de contacto das rodas com o chão.

$$v(t) = \frac{V_1(t) + V_2(t)}{2} \quad (3)$$

$$w(t) = \frac{V_1(t) - V_2(t)}{b} \quad (4)$$

Considerando a condição de não escorregamento a cinemática do *robot* diferencial está descrita pela seguinte equação:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} v(t)\cos(\theta(t)) \\ v(t)\sin(\theta(t)) \\ w(t) \end{pmatrix} \quad (5)$$

Esta equação permite usando as equações (3) e (4) exprimir as velocidades lineares \dot{x} , \dot{y} e a velocidade angular $\dot{\theta}$ em função das velocidades de cada uma das rodas, podendo estas ser medidas (Costa, Tese de Mestrado, Faculdade de Engenharia da Universidade do Porto, 1995).

Para colmatar a limitações do *robot* diferencial surgiu o *robot* omnidireccional, permitindo deslocações em todas as direcções (Huang et al., 2004). Para garantir a característica da omnidireccionalidade é necessário que as rodas usadas tenham pouco atrito na direcção do veio do motor, o que impediria deslocações segundo esse eixo.

Como podemos comprovar da observação da geometria de um *robot* omnidireccional com três rodas, representada na figura 4, as velocidades V_x , V_y e w variam com a velocidades lineares V_1 , V_2 e V_3 , baseando-se na equação (6) (Kalmár-Nagy et al., 2002).

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} = \begin{pmatrix} -\sin(\theta) & \cos(\theta) & L \\ -\sin(\frac{\pi}{2} - \theta) & \cos(\frac{\pi}{2} - \theta) & L \\ -\sin(\frac{\pi}{3} + \theta) & \cos(\frac{\pi}{3} + \theta) & L \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ w \end{pmatrix} \quad (6)$$

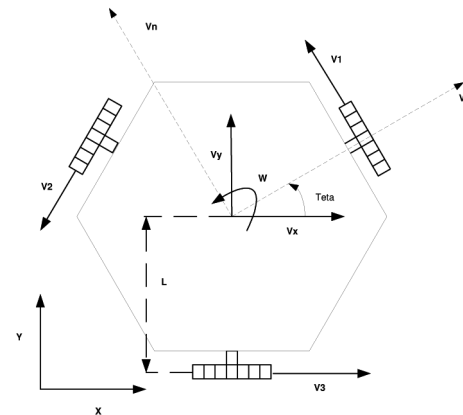


Figura 4 Geometria de um robot omnidireccional de três rodas.

O projecto do controlador é baseado no modelo apresentado na equação (6). As equações de cinemática do *robot*, poderiam ser representadas pelas equações (7) e (8) em alternativa à equação (6).

$$\begin{pmatrix} V \\ V_n \\ w \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ w \end{pmatrix} \quad (7)$$

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & L \\ -\sin(\frac{\pi}{3}) & \cos(\frac{\pi}{3}) & L \\ -\sin(\frac{\pi}{3}) & \cos(\frac{\pi}{3}) & L \end{pmatrix} \begin{pmatrix} V \\ V_n \\ w \end{pmatrix} \quad (8)$$

A equação (7) representa a velocidade (V) e a velocidade normal (V_n) do *robot* em função das velocidades V_x e V_y . A equação (8) representa as velocidades V_1 , V_2 e V_3 calculadas a partir das velocidades do *robot*.

3. CONTROLADOR

Para controlar o sistema é necessário controlar independentemente cada uma das rodas do *robot*, as quais lhe vão imprimir movimento. O controlador tem diversas hierarquias, sendo a sua função a mais baixo nível controlar independentemente cada uma das rodas e a mais alto nível determinar a que velocidade deve girar cada roda para que o *robot* se desloque com uma determinada velocidade angular e linear. O controlador a mais alto nível é efectuado baseando-se na equação (6) e num controlador de posição. A mais baixo nível cada motor é controlado em malha fechada usando-se um controlador do tipo **PI**.

O controlador tem como objectivo a deslocação do *robot* para um determinado ponto com velocidade controlada. Como parâmetros de entrada temos um objectivo que consiste em o *robot* se deslocar para um determinado ponto. Com este dado calcula-se um vector de posição que aponta para o sítio onde queremos deslocar o *robot*, esse vector depois de normalizado é convertido num vector de velocidade, passando este a ser o objectivo a alcançar. O controlador determina a que velocidade deve girar cada roda, com vista a que *robot* se desloque com uma determinada velocidade em X , Y e em θ (caso seja necessário corrigir o ângulo). A cada tempo de amostragem a posição muda, consequentemente varia o vector de posição, o vector de velocidade e a referência da velocidade de rotação dos motores. O controlador está representado no fluxograma da figura 5.

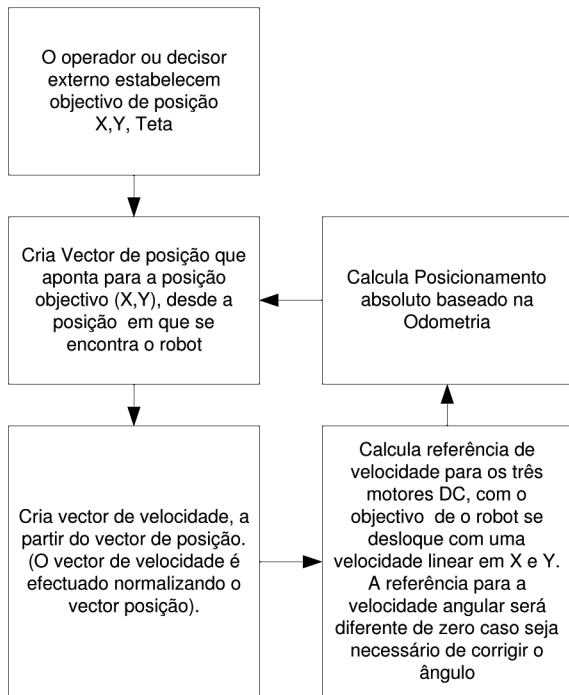


Figura 5 Controlador do robot.

3.1 Optimização do Controlador

O controlador tem como função definir qual a velocidade a que os motores devem rodar, de modo a que o *robot* rode com uma determinada velocidade angular $\dot{\theta}$ e se desloque com uma velocidade linear \dot{x} e \dot{y} .

O controlador foi optimizado desenvolvendo as funções $\sin(\theta + \frac{\pi}{3})$, $\cos(\theta + \frac{\pi}{3})$, $\cos(\theta - \frac{\pi}{3})$ e $\sin(\theta - \frac{\pi}{3})$, usando as formulas:

$$\sin(a + b) = \sin(a) \cos(b) + \cos(a) \sin(b) \quad (9)$$

$$\sin(a - b) = \sin(a) \cos(b) - \cos(a) \sin(b) \quad (10)$$

$$\cos(a + b) = \cos(a) \cos(b) - \sin(a) \sin(b) \quad (11)$$

$$\cos(a - b) = \cos(a) \cos(b) + \sin(a) \sin(b) \quad (12)$$

O objectivo de desenvolver estas funções trigonométricas foi o de o controlador se tornar computacionalmente menos pesado, pois é muito menos dispendioso a nível de tempo de cálculo calcular apenas $\sin(\theta)$ e $\cos(\theta)$, que efectuar o cálculo de vários senos e cosenos, tal como é apresentado na equação (6). Consequentemente para o cálculo das velocidades V_2 e V_3 torna-se vantajoso efectuar as alterações acima referidas.

Assim para o cálculo de V_2 , torna-se computacionalmente mais eficiente usar a equação (14) em vez da equação (13).

$$v_2 = -\sin\left(\frac{\pi}{3} - \theta\right)\dot{x} - \cos\left(\frac{\pi}{3} - \theta\right)\dot{y} + L\dot{\theta} \quad (13)$$

$$v_2 = -\left(\sin\left(\frac{\pi}{3}\right)\cos(\theta) - \cos\left(\frac{\pi}{3}\right)\sin(\theta)\right)\dot{x} - \left(\cos\left(\frac{\pi}{3}\right)\cos(\theta) - \sin\left(\frac{\pi}{3}\right)\sin(\theta)\right)\dot{y} + L\dot{\theta} \quad (14)$$

O mesmo se verifica para o cálculo de V_3 tornando-se mais eficiente usar a equação (16) que a (15).

$$v_3 = -\sin\left(\frac{\pi}{3} + \theta\right)\dot{x} - \cos\left(\frac{\pi}{3} + \theta\right)\dot{y} + L\dot{\theta} \quad (15)$$

$$v_3 = \left(\sin\left(\frac{\pi}{3}\right)\cos(\theta) + \cos\left(\frac{\pi}{3}\right)\sin(\theta)\right)\dot{x} - \left(\cos\left(\frac{\pi}{3}\right)\cos(\theta) - \sin\left(\frac{\pi}{3}\right)\sin(\theta)\right)\dot{y} + L\dot{\theta} \quad (16)$$

3.2 Controlo dos motores DC

3.2.1 Dimensionamento dos parâmetros dos controladores

O dimensionamento dos parâmetros dos controladores dos motores pode ser feito de diversas maneiras sendo apresentada uma solução possível, correspondendo a efectuar-se a identificação do pólo dominante do modelo do motor. Após a obtenção deste dado, efectua-se o cálculo de um controlador do tipo PI, colocando um zero ligeiramente maior que o pólo dominante e posteriormente variando o ganho até se obter a resposta requerida (isto é tornando o sistema o mais rápido possível sem que exista um *overshoot* significativo). O modelo de um motor de corrente contínua pode ser aproximado para um sistema de segunda ordem, tendo um pólo mecânico e um pólo eléctrico. O pólo mecânico é dominante, sendo o seu tempo de estabelecimento muito superior ao do pólo eléctrico. O tempo de estabelecimento da velocidade do motor em malha fechada é sensivelmente 5 vezes o valor da constante de tempo associada ao pólo dominante (Ogata, 1993). Desconhecendo o valor dos dois pólos é possível estimar o pólo dominante por observação da resposta a um degrau unitário do motor (figura 6). Esta opção permite que não seja necessário saber todos os parâmetros do sistema para o dimensionamento do controlador.

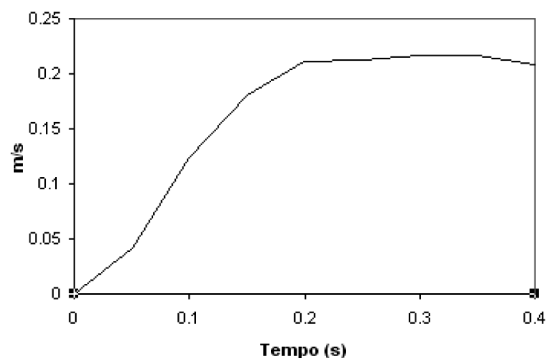


Figura 6 Resposta em malha aberta da velocidade do ponto associado à roda a um degrau unitário

Da observação dos dados da resposta ao degrau podemos constatar que o tempo de estabelecimento é sensivelmente 0,3 segundos e podemos aproximar o pólo mecânico como inverso desta constante de tempo a dividir por cinco ($1/(0,3/5)$). Estando identificado o pólo mais lento deste sistema podemos dar início ao cálculo do controlador. O controlador usado foi um PI, de maneira a ser garantido em regime permanente erro nulo à resposta a um degrau. A sua função de transferência é mostrada na equação (17).

$$G(s) = \frac{Kp(s + a)}{s} \quad (17)$$

O parâmetro a tem um valor que é o dobro do pólo mecânico estimado, e o parâmetro K_p ser a ajustado sucessivamente até se obter uma resposta temporal que seja satisfatória, isto é com o melhor regime dinâmico possível e sem erro em regime permanente. Foram efectuados diferentes testes de maneira a se obter a resposta mais satisfatória possível em malha fechada,

CONTROLO E ESTIMACÃO DO POSICIONAMENTO ABSOLUTO DE UM ROBOT OMNIDIRECCIONAL DE TRÊS RODAS

não sendo possível eliminar completamente o erro em regime permanente, devido às perturbações na carga. A resposta em malha fechada com uma referência de 0.2 m/s, está representada na figura 7.

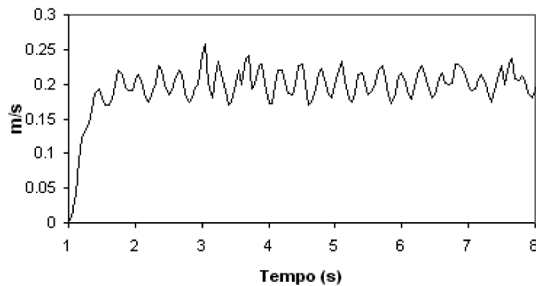


Figura 7 Resposta em malha fechada da velocidade do ponto associado à roda com uma referência de 0.2 m/s

Após se obterem os parâmetros para um controlador analógico do tipo PI que cumpre os requisitos impostos, o passo seguinte foi o cálculo dos parâmetros para um controlador digital, que emule o controlador analógico. Sendo o controlador analógico dado pela equação (18).

$$PI(s) = Kp + \frac{a.Kp}{s} \quad (18)$$

Emula-se o Controlador analógico com um controlador digital, sendo o valor T o tempo de amostragem, o qual deve ser pelo menos 10 vezes menor que o tempo de estabelecimento do polo mais rápido para que não exista *aliasing* na reconstrução do sinal erro, dado um amostrador ideal ter como propriedade replicar o espectro do sinal original centrado em $K * f$, com K pertencente a N e f igual a $1/T$ (Phillips, 1995). O controlador PID em controladores analógicos é dado pela equação (19), mas porque o controlador usado será PI apenas nos vamos debruçar sobre a equação (20).

$$m(t) = Kp[e(t) + \frac{1}{Ti} \int_0^t e(t) + Td \frac{de(t)}{dt}] \quad (19)$$

$$m(t) = Kp[e(t) + \frac{1}{Ti} \int_0^t e(t)] \quad (20)$$

Em que $e(t)$ representa o erro, sendo este a entrada do controlador e $m(t)$ a saída do controlador, sendo Kp o ganho proporcional e Ti o ganho integral. Para obter resposta a um impulso de um PI digital podemos discretizar a função erro, aproximando o integral para uma soma de trapézios, obtendo-se a equação (21), tal como é indicado na figura 8

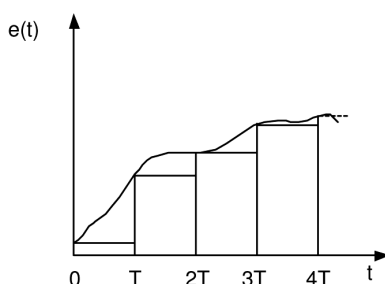


Figura 8 Discretização da função erro.

A partir da equação (17), da equação (21) e da equação (22) obtém-se a equação (23), sendo este o controlador digital que emula o funcionamento do controlador analógico (Ogata, 1995).

$$m(kT) = Kp[e(kT) + \frac{T}{Ti}(e(0) + e(T) + \dots + e(K.T))] \quad (21)$$

$$Z[\sum_{h=1}^k e(hT)] = \frac{1}{(1-z^{-1})} E(z) \quad (22)$$

$$PI(z) = Kp + \frac{aKpT}{(1-z^{-1})} \quad (23)$$

4. ODOMETRIA

As equações para o cálculo da odometria foram deduzidas através da inversão da matriz representada pelo sistema de equações (6). As equações obtidas são as seguintes:

$$V_x = (-V_2 \sin(\frac{\pi}{3}) + \sin(\frac{\pi}{3})V_3 + \sin(2\theta)V_3 - \sin(2\theta)V_1 + 2V_2 \sin(\frac{\pi}{3}) \sin(\theta)^2 + V_1 \sin(2\theta) \cos(\frac{\pi}{3}) - \cos(\frac{\pi}{3}) \sin(2\theta)V_2 - 2V_1 \sin(\frac{\pi}{3}) \sin(\theta)^2) / 2((\sin(\theta) + \sin(\frac{\pi}{3}) \cos(\theta) + \cos(\frac{\pi}{3}) \sin(\theta)) \sin(\frac{\pi}{3})) \quad (24)$$

$$V_y = -\frac{1}{2 \sin(\frac{\pi}{3})^2} (V_3 \cos(\theta) - 2V_1 \cos(\theta) + V_2 \cos(\theta) + V_2 \sin(\theta) \sin(\frac{\pi}{3}) - V_3 \sin(\theta) \sin(\frac{\pi}{3}) - V_3 \cos(\theta) \cos(\frac{\pi}{3}) - V_2 \cos(\theta) \cos(\frac{\pi}{3}) + 2V_1 \cos(\theta) \cos(\frac{\pi}{3})) \quad (25)$$

$$w = (-2 \cos(\theta)V_1 + \cos(\theta)V_3 + V_2 \cos(\theta) - \cos(\theta) \cos(\frac{\pi}{3})V_2 + \sin(\theta)V_2 \sin(\frac{\pi}{3}) + \sin(\theta)V_3 \sin(\frac{\pi}{3}) + 2V_1 \sin(\frac{\pi}{3}) \cos(\frac{\pi}{3}) \sin(\theta) + 2V_1 \sin(\frac{\pi}{3})^2 \cos(\theta) + 2 \cos(\theta) \cos(\frac{\pi}{3})V_1 - \cos(\theta) \cos(\frac{\pi}{3})V_3) / (2L \sin(\frac{\pi}{3})(\sin(\theta) + \sin(\frac{\pi}{3}) \cos(\theta) + \cos(\frac{\pi}{3}) \sin(\theta))) \quad (26)$$

O cálculo destas equações torna-se bastante dispendioso computacionalmente e como o objectivo é utilizá-las para efectuar o controlador estas devem ser o mais optimizadas possível, com vista a cumprir requisitos de tempo real.

O cálculo da odometria pode ser optimizado utilizando a equação (27).

$$2 \sin(\theta) \cos(\theta) = \sin(2\theta) \quad (27)$$

O objectivo de usar esta função trigonométrica foi optimizar o Controlador do robot representado na figura 5, no qual está implícito o cálculo da odometria. Deste modo calcula-se apenas uma vez $\sin(\theta)$ e $\cos(\theta)$, evitando o cálculo de vários senos e cosenos.

Nestas condições torna-se mais eficiente para o cálculo da odometria usar

a equação seguinte em detrimento da apresentada anteriormente:

$$\begin{aligned}
 V_x = & (-V_2 \sin(\frac{\pi}{3}) + \sin(\frac{\pi}{3})V_3 + 2 \sin(\theta) \cos(\theta)V_3 \\
 & -2\sin(\theta) \cos(\theta)V_1 + 2V_2 \sin(\frac{\pi}{3})\sin(\theta)^2 + V_1 2 \sin(\theta) \\
 & \cos(\theta)) \cos(\frac{\pi}{3}) - \cos(\frac{\pi}{3})2 \sin(\theta) \cos(\theta)V_2 \\
 & -2V_1 \sin(\frac{\pi}{3}) \sin(\theta)^2)/2((\sin(\theta) \\
 & + \sin(\frac{\pi}{3}) \cos(\theta) + \cos(\frac{\pi}{3}) \sin(\theta)) \sin(\frac{\pi}{3}))
 \end{aligned} \quad (28)$$

4.0.1. Calibração da odometria

A cada roda está associado um *encoder*, este sensor permite saber em que sentido está a rodar cada roda e a que velocidade. Para se efectuar a calibração da odometria é necessário dar valores aos parâmetros que relacionam o número de transições que cada roda fornece com a distância percorrida. A maneira como é transmitido movimento da roda para o *encoder* está representado na figura 9. Sendo estes valores diferentes de uma roda para a seguinte, o procedimento para a calibração consiste em primeiro lugar em efectuar o cálculo da equação (29), com o objectivo de nos aproximarmos do valor pretendido.

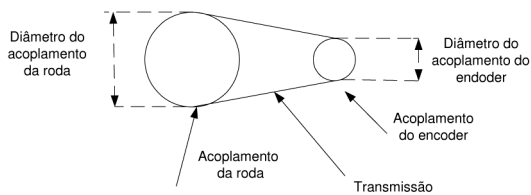


Figura 9 Transmissão entre a roda e o *encoder*.

$$D = \frac{D1 \pi \cdot Diamr}{D2 \cdot tranv} \cdot tran \quad (29)$$

- D Distância percorrida
- $D1$ Diâmetro do acoplamento do encoder
- $D2$ Diâmetro do acoplamento da roda
- $Diamr$ Diâmetro da roda
- $tran$ Transições registadas num Tempo de amostragem
- $tranv$ Transições registadas numa volta

Após termos efectuado o cálculo da equação (29) obtemos um factor que converte transições em distância percorrida, podemos proceder então a uma afinação deste factor para cada uma das rodas, fazendo com que percorram distâncias e comparando-as com as que a fórmula fornece.

Além de ser calibrado para cada uma das rodas um factor que converte transições em distância percorrida, é necessário calibrar também um outro factor que é o parâmetro L , estando este ilustrado na figura 4. Para se efectuar esta calibração coloca-se o *robot* a rodar sobre si mesmo com vários valores para este parâmetro. O objectivo desta operação é comparar o ângulo estimado pelos cálculos da odometria e o medido com um transferidor, quando estes valores estiverem próximos podemos considerar que o parâmetro se encontra calibrado.

5. VALIDAÇÃO DO CONTROLADOR E DA ODOMETRIA

Com o objectivo de analisar o desempenho do controlador e avaliar os erros

de odometria foi realizada uma corrida, estando descrita pelo fluxograma da figura 10. Durante a corrida foram registadas a posição estimada pelo cálculo da odometria e a posição absoluta real. Para se obter a posição absoluta real foi utilizado um sistema de visão para localização em tempo real de múltiplos robots (Costa, Tese de Doutoramento, Faculdade de Engenharia da Universidade do Porto, 1999), utilizado pela equipa 5DPO na *Small Size League* (Costa et al., 2004) do *Robocup* (Robocup, 2005).

Como podemos observar na figura 10 o *robot* desloca-se passando por vários pontos efectuando deste modo uma trajectória. A estimação do posicionamento usando-se visão externa contem um erro absoluto não cumulativo, permitindo saber com erro menor que 1 cm o posicionamento absoluto do *robot* em qualquer instante. Por outro lado o posicionamento estimado pelo cálculo da odometria contem erros cumulativos, os quais decorrido algum tempo tornam a estimativa muito desfasada da realidade. Deste modo podemos tirar conclusões quanto às possíveis fontes de erro na estimativa do posicionamento, analisando-se a sua evolução, utilizando como referência o sistema de visão externo. A trajectória real (visão) e a estimada (odometria) estão representadas na figura 11.

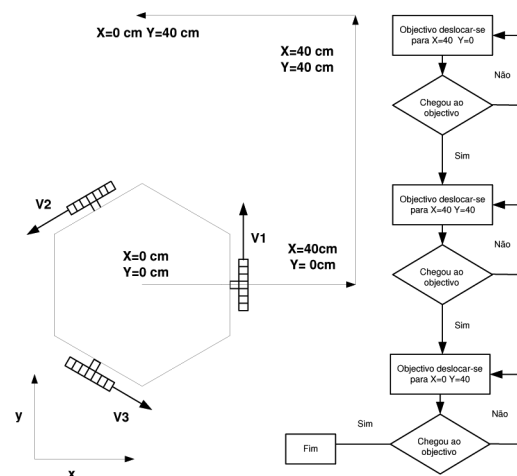


Figura 10 Fluxograma da corrida efectuada pelo *robot*.

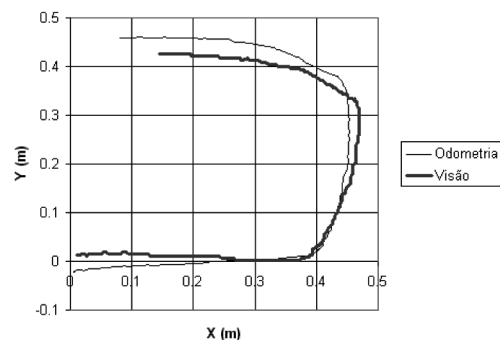


Figura 11 Posicionamento absoluto do *robot* baseado em visão externa e no cálculo da odometria.

Nas figuras 12, 13 e 14, é apresentado o posicionamento absoluto do *robot*, registado com visão externa, sendo o *robot* colocado num valor de posicionamento absoluto indicado pela visão como sendo próximo da origem. Como podemos observar dos gráficos relativos à trajectória do *robot*, este vai efectuar duas mudanças bruscas de direcção, tendo este facto conse-

CONTROLO E ESTIMAÇÃO DO POSICIONAMENTO ABSOLUTO DE UM ROBOT OMNIDIRECCIONAL DE TRÊS RODAS

quências a nível da estimação do posicionamento do *robot* baseado no cálculo da odometria, pois os cálculos estão a ser efectuados com base em parâmetros que variam. Nas situações em que existem variações bruscas de direcção é necessário ter em conta que as rodas podem plissar, que existem folgas e o facto da polia que transmite movimento da roda para o *encoder* ter elasticidade. Além disso o parâmetro L varia dinamicamente, dada a estrutura física das rodas.

Um facto importante que deve ser realçado do gráfico da figura 11, é que quando se espera que o *robot* passe pelo ponto $X = 40\text{ cm}$ e $Y = 0\text{ cm}$, o *robot* começa a deslocar-se para o ponto seguinte antes de passar por este. Isto acontece porque o objectivo de chegar a um ponto é atingido se o erro em X e em Y forem menores que 2 cm , evoluindo a máquina de estados para o objectivo seguinte, que será o ponto $X = 40\text{ cm}$ e $Y = 40\text{ cm}$.

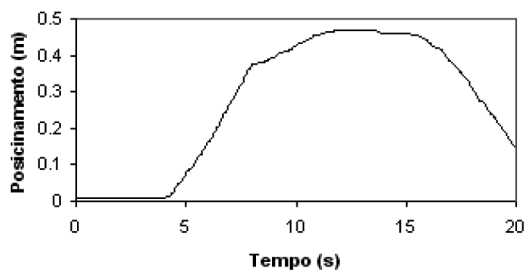


Figura 12 Posicionamento absoluto em X do *robot* baseado em visão externa.

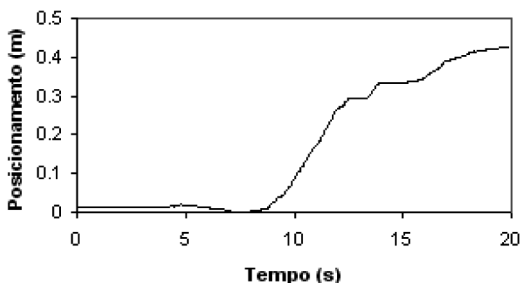


Figura 13 Posicionamento absoluto do *robot* em Y baseado em visão externa.

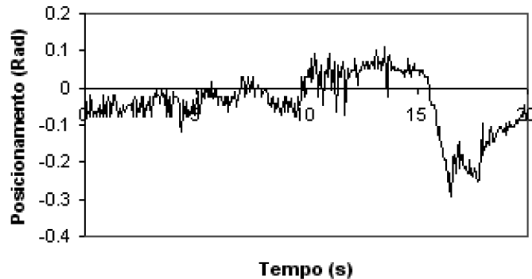


Figura 14 Posicionamento absoluto do *robot* em teta baseado em visão externa.

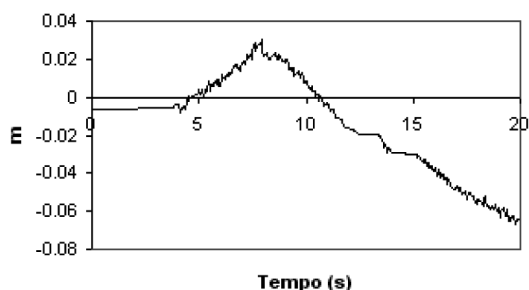


Figura 15 Erro absoluto de posicionamento em X do *robot*.

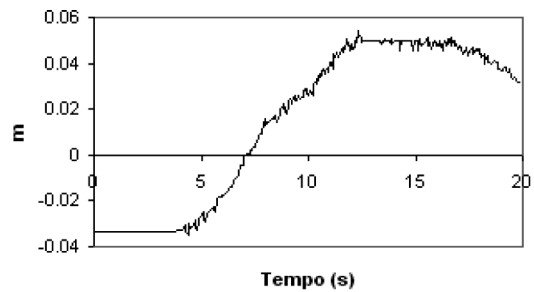


Figura 16 Erro absoluto de posicionamento *robot* em Y.

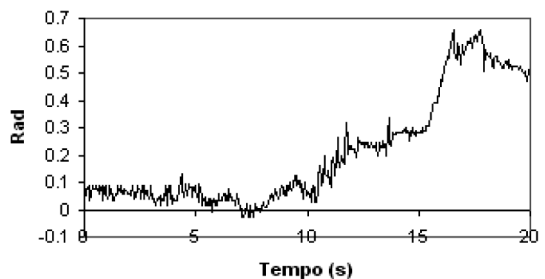


Figura 17 Erro absoluto de posicionamento do *robot* em teta.

Da observação das figuras 15, 16 e 17 podemos afirmar que existe um pequeno erro em X , Y e $teta$ na posição inicial. Após o início da corrida observou-se que para uma trajectória rectilínea não existem variações significativas no erro do ângulo, mas existindo variações no erro em X e Y devido ao erro inicial no ângulo. O erro no ângulo sofre um incremento significativo quando existem mudanças bruscas de direcção, existindo incremento aos 11 e 16 segundos. A este incremento no erro em $teta$ corresponde um conseqüente acumular do erro em X e Y , o que leva a que a estimação do posicionamento baseado na odometria se torne inevitavelmente cada vez mais desfasado da realidade.

6. CONCLUSÕES

Ao ser monitorizada a trajectória seguida pelo *robot* e comparando-se esta com a estimativa do posicionamento pelo cálculo da odometria podemos tirar algumas conclusões face às fontes de erro da odometria. Observou-se que numa situação em que existe uma variação brusca de direcção temos que ter em conta que as rodas podem plissar, que existem folgas e o facto da polia que transmite movimento da roda para o *encoder* ter elasticidade. Este facto implica que hajam nestas situações um erro acrescido na estimação da posição, acumulando-se essencialmente um maior erro no ângulo, o que vai fazer com que o erro posteriormente se acumule também em X e Y .

Verifica-se que se torna insuficiente o cálculo da odometria para o posicionamento absoluto de um *robot*, pois passado algum tempo o posicionamento estimado encontra-se completamente desfasado da realidade. Deste modo como alternativa o posicionamento pode ser estimado usando múltiplas fontes ou usando-se um sistema que não contenha um erro cumulativo, sendo um exemplo o sistema de visão usado para comparar a trajectória real com a estimada.

Os problemas de controlo em robótica incluem inevitáveis requisitos de tempo real muito apertados. Nesta situação particular o cálculo da odometria

e do controlo de um *robot* omnidireccional é computacionalmente muito mais pesado que para os tipicamente usados *robots* diferenciais, como pode ser constatado após inspecção das equações apresentadas. Deste modo é necessário ter um cuidado especial ao serem implementadas as operações matemáticas necessárias, pois o código deve ser optimizado com vista a cumprir requisitos de tempo real.

7. REFERÊNCIAS

Costa, Paulo (Tese de Doutoramento, Faculdade de Engenharia da Universidade do Porto, 1999). Localização em tempo real de múltiplos *robots* num ambiente dinâmico.

Costa, Paulo (Tese de Mestrado, Faculdade de Engenharia da Universidade do Porto, 1995). Identificação, modelização e controlo de um veículo móvel.

Costa, Paulo, António Moreira, Paulo Marques, Armando Sousa, Pedro Costa and Susana Gaio (2004). 5dpo robotic soccer team for year 2004. RoboCup2004, Lisbon, Portugal.

Dudek, Gregory and Michael Jenkin (2000). *Computational Principles of Mobile Robotics*. Cambridge University Press.

Huang, L., Y. S. Lim, David Lee and Christopher E. L. Teoh (2004). Design and analysis of a four-heel omnidirectional mobile robot. *2nd International Conference of Autonomous Robots and Agents*.

Kalmár-Nagy, Tamás, Raffaello D'Andrea and Pritam Ganguly (2002). Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle. Sibley School of Mechanical and Aerospace Engineering Cornell University Ithaca, NY 14853, USA.

Ogata, Katsuhiko (1993). *Engenharia de controlo Moderno*. Prentice Hall do Brasil LDA.

Ogata, Katsuhiko (1995). *Discrete-time control systems*. Prentice-Hall, Inc.

Phillips, Charles L. (1995). *Digital Control Systems, analysis and design*. Prentice-Hall, Inc.

Robocup (2005). <http://www.robocup.org/>.

1/2 pg publicidade

1 pg publicidade