

CyberMouse: Robô móvel modular para aplicações didácticas

Abel Mendes, Lino Marques e Urbano Nunes

{abfm,lino,urbano}@isr.uc.pt

Instituto de Sistemas e Robótica

Departamento de Engenharia Electrotécnica e de Computadores

Universidade de Coimbra

3030-290 Coimbra, Portugal

Resumo – Neste artigo é descrito um robô móvel de pequenas dimensões, desenvolvido no âmbito do concurso MicroRato. Serão abordadas quatro áreas: estrutura mecânica; unidade electrónica de controlo e sensores; software, baseado em programação comportamental, segundo o modelo de submissão de Brooks [1]; controlo de seguimento de trajectórias, com apresentação de resultados dos controladores de Kanayama [2] e Yang [3].

Abstract – This paper presents a mobile robot of small dimensions developed within the MicroRato contest. The paper focus four main areas of interest: the mechanical structure; the unit controller and sensors; the software, based on behaviour programming, following Brooks subsumption architecture model; trajectory tracking, based on Kanayama and Yang controllers. Simulation results will be also presented.

Palavras chave – Microcontrolador, lógica programável, sensores reflectivos, estrutura de controlo baseada em comportamentos, controlo de trajectórias, comportamentos reactivos.

I. INTRODUÇÃO

Este artigo tem a finalidade de apresentar um robô modular não holónimo, construído no âmbito do concurso MicroRato.

No decurso da prova, o robô deve ir desde a área de partida até à área de chegada sem colidir com os obstáculos com que se depara ao longo do percurso. A área de chegada do percurso de ida é um círculo de cor preta, onde se encontra um farol de infravermelhos. Posteriormente, o robô terá de fazer o percurso inverso até chegar de novo à casa de partida.

De forma a que o CyberMouse possuísse as funcionalidades necessárias a resolver o labirinto e voltar à área de partida, foram desenvolvidos os seguintes sensores: sensores de distância, detector da localização do farol de infravermelhos e detector da direcção do farol. Para integrar todos os sensores e decidir o caminho a seguir, construiu-se uma placa controladora baseada no microcontrolador MC68HC912D60 da Motorola e no dispositivo de lógica programável XC95108 da Xilinx.

O modelo de programação utilizado baseia-se em comportamentos reactivos e permite um desenvolvimento modular de um conjunto de comportamentos extremamente simples

e independentes, que permitem levar a cabo um modelo coerente e eficaz de navegação local [1].

Para levar o robô a regressar à casa de partida, o procedimento baseia-se no seguimento de uma trajectória obtida aquando do trajecto de ida. O controlo de seguimento de trajectórias de robôs móveis não holónomos é um problema que tem vindo a ser alvo de estudo em anos recentes. Para o CyberMouse, vamos analisar o comportamento dos controladores de Kanayama [2] e Yang [3] que requerem da parte do robô pequena capacidade de processamento.

II. ESTRUTURA MECÂNICA

A plataforma do robô é construída em acrílico, por se tratar de um material leve, de fácil maquinação e bastante resistente. Do conjunto alargado de configurações que os robôs móveis podem possuir, provavelmente a mais usual é a configuração diferencial. A versatilidade desta configuração permite efectuar rotações e translações puras, bem como um misto rotação/translação, possibilitando que o robô se desloque com flexibilidade.

As baterias foram colocadas estrategicamente, de forma a que o centro de massa do robô ficasse situado entre o eixo das rodas motrizes e o *castor*¹.

A tracção do robô, é realizada por dois motores DC Pittman GM9234, alimentados a 24 V, acoplados a uma caixa de redução de 11.5 : 1, que fornecem à saída 535 rpm. Como as rodas usadas possuem um raio de 7.6 cm, o robô pode atingir uma velocidade máxima de 213 cm/s.

III. ELECTRÓNICA

A. Unidade controladora

O hardware de controlo é constituído por um processador e vários periféricos necessários à leitura de sensores e controlo de actuadores. De forma a dotar o CyberMouse com as funcionalidades exigidas pela tarefa a realizar, incluíram-se os seguintes componentes e periféricos:

- Microprocessador
- Memória
- Comunicação série
- Geradores de PWM
- Descodificadores de codificadores das rodas
- Conversores A/D
- Entradas e saídas digitais.

¹Roda livre, tipo roda de pé de mesa

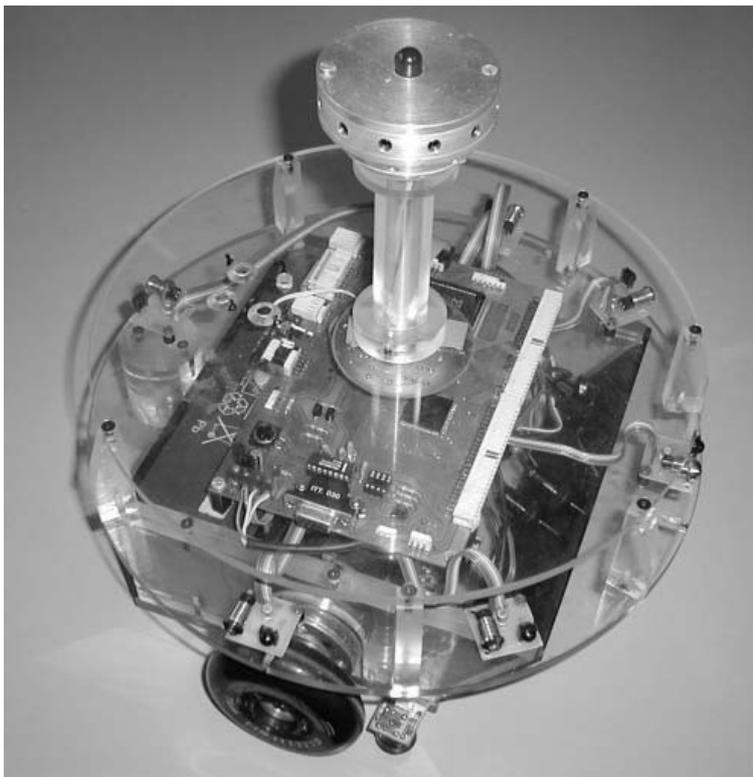


Figure 1 - Fotografia do CyberMouse

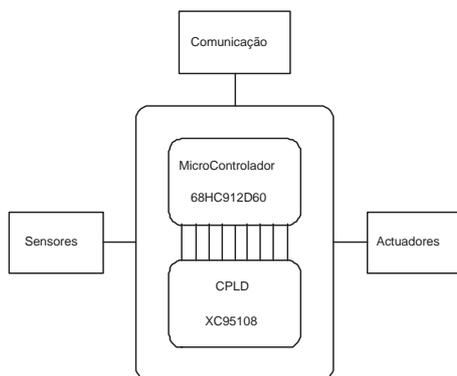


Figure 2 - Diagrama de blocos da placa controladora

Com o objectivo de construir uma placa que satisfizesse as necessidades actuais do robô, limitámo-nos a incluir apenas os periféricos indispensáveis e deixar uma elevada capacidade de expansão, para futuramente poderem ser desenvolvidos módulos extra dedicados a novas funcionalidades. Os dispositivos fundamentais que compõem esta unidade, são o microcontrolador *HC912D60* da Motorola e a CPLD² *XC95108* da Xilinx. Na CPLD foram implementados os contadores de pulsos dos codificadores ópticos das rodas, bem como a descodificação de endereços para vários periféricos. Os restantes blocos da placa controladora (veja-se a figura 2), destinam-se a implementar os restantes periféricos.

²Complex Programmable Logic Device

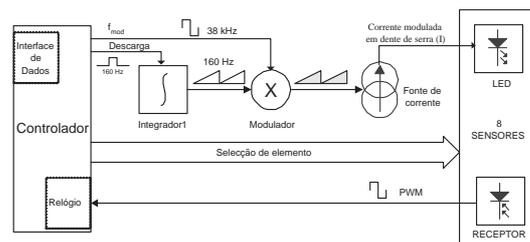


Figure 3 - Sensor de distâncias, baseado em potência de emissão variável

B. Sensores

Para que o robô possa movimentar-se em ambientes desconhecidos, tem necessidade de adquirir informação sobre o ambiente. Para desempenhar esse papel poderíamos optar por uma de várias abordagens, tais como: sensores de ultrasons; sensores de imagem (câmaras); sensores de infravermelhos; etc... A nossa escolha recaiu nos sensores de infravermelhos reflectivos, porque apesar de não fornecerem dados muito precisos devido à sua sensibilidade em relação às características reflectivas e orientação dos obstáculos, têm a vantagem de serem compactos e utilizar hardware simples.

B.1 Distância

O método tradicional usado para medir distâncias através da optoelectrónica, utiliza um emissor de potência constante, modulado à frequência do receptor. Sendo o receptor um sensor analógico, fornece uma tensão cujo valor depende da energia luminosa reflectida, logo da distância ao obstáculo. Na figura 3, apresenta-se a arquitectura concep-

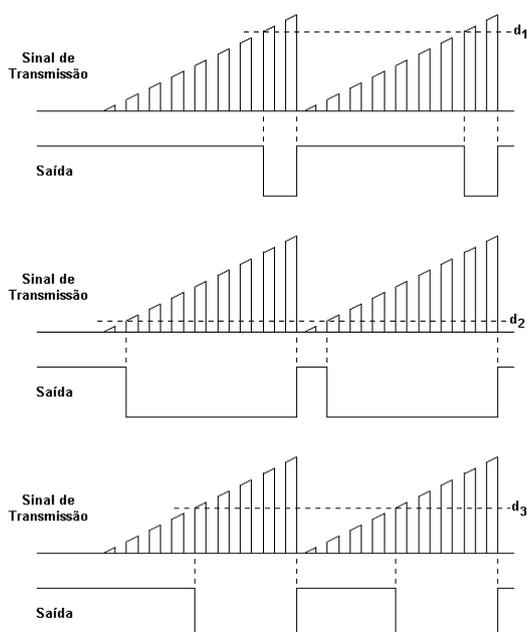


Figure 4 - Exemplos de medições de distâncias através de um receptor digital

tual do sensor realizado, baseado numa nova técnica que usa sensores que fornecem uma informação ON/OFF. O método utilizado para medir distâncias com estes sensores baseia-se na reflexão de um feixe de luz com uma potência de emissão variável [4]. Assim, se a potencia do sinal emitido tiver uma envolvente do tipo dente de serra, obtém-se à saída do sensor digital, um pulso com uma largura proporcional à distância ao obstáculo. Na figura 4, são apresentados exemplos de medições onde se pode verificar que o sinal obtido a partir do sensor representa a distância ao obstáculo. A reflexão originada pelo feixe de luz emitida no início do ciclo não é captada pelo receptor devido à fraca potência de emissão, obtendo assim o valor lógico alto na saída do sensor. Com o aumento da potência de emissão aumenta também a reflexão. Quando essa reflexão atinge o limiar de captação do sensor, este coloca a saída a zero (estado activo). Desta forma, é possível medir a distância ao objecto com base no instante de início da detecção da reflexão do feixe luminoso. Para realizar o sensor descrito, utilizamos o seguinte par emissor-receptor: emissor de infravermelho HIRL5010 da Rodan, com uma potência máxima de emissão de 120 mW e ângulo de abertura de 12° ; detector de infravermelho IS1U60 da Sharp, com uma frequência central de 38 kHz [5].

A caracterização do sensor descrito baseou-se nas respostas obtidas para uma superfície lisa pintada de branco com uma área de 1600 cm^2 que faz um ângulo θ com a perpendicular à linha de emissão do sensor. Foram realizadas medidas de 5 em 5 cm, até ao limite para o qual os valores deixavam de ser consistentes (veja-se a figura 5). Os resultados obtidos para as diferentes situações de luminosidade ambiente não são apresentados, uma vez que os testes revelaram uma grande independência nas medições em função da luz ambiente. Esta situação deve-se ao facto dos sensores utilizarem luz com comprimento de onda na

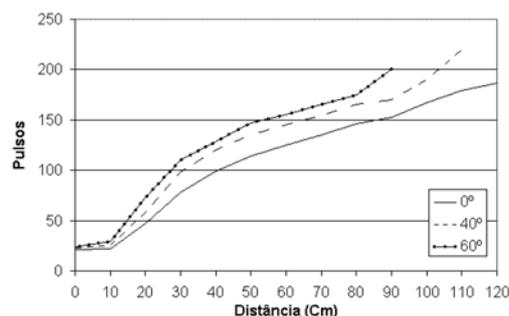


Figure 5 - Resultados obtidos com o sensor de distância, para diferentes inclinações da superfície a medir

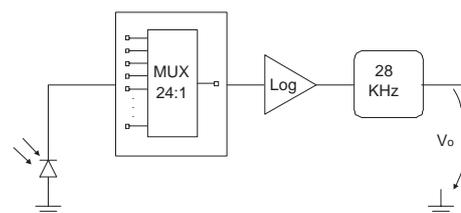


Figure 6 - Diagrama de blocos do Detector de Farol

gama dos infravermelhos modulada e respectiva filtragem óptica e eléctrica na detecção.

Analisando o gráfico da figura 5, verificamos que se pode dividir as respostas do sensor em dois intervalos: $[0..10]$ em que a resposta é aproximadamente linear e de baixa variação; $[10..max]$ onde a resposta pode ser aproximada por uma função polinomial de segunda ordem.

B.2 Detector de farol

Para detectar a frequência de 28 kHz do farol, foi desenvolvido um sistema composto por vários fotodíodos, multiplexadores, um amplificador logarítmico, um filtro passa-banda, um filtro passa-baixo de 8^a ordem e um circuito retificador (ver figura 6).

Uma vez que os sinais provenientes dos fotodíodos são correntes de valor muito baixo, é crucial que os multiplexadores tenham correntes de fuga muito reduzidas, por forma a não degradar o sinal de entrada. Tendo em conta estas considerações, a nossa selecção recaiu sobre o MAX336 (16 : 1) e o MAX338 (8 : 1) da Maxim. Por vezes é desejável obter valores de tensão, que sejam proporcionais ao logaritmo de outros sinais. No caso deste sensor, uma vez que os sinais provenientes dos fotodíodos, são bastante pequenos, podendo variar numa gama muito elevada, interessa-nos amplificar mais os sinais de amplitudes reduzidas, que os de “elevada” amplitude, pois é crucial que este sensor seja capaz de detectar sinais a grandes distâncias, daí a implementação de um amplificador logarítmico. Para que se possam processar os dados provenientes dos vários fotodíodos, é necessário enviar essa informação ao microcontrolador. À saída do andar anterior vamos ter uma onda sinusoidal de valor médio nulo. Para uma correcta leitura do valor eficaz dessa onda, procede-se à sua rectificação.

Para caracterizar este sensor, foram realizados os seguintes testes:

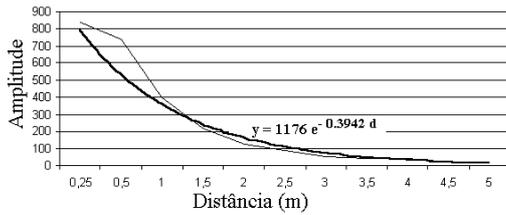


Figure 7 - Resposta para diferentes distâncias ao farol com emissão a 28 kHz

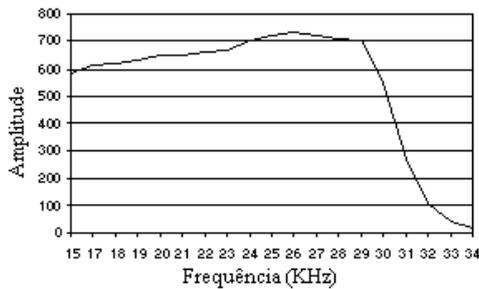


Figure 8 - Resposta para diferentes frequências emitidas pelo farol

1. Colocação do farol alinhado com o sensor, aumentando progressivamente a distância entre eles.
2. Medição da amplitude do sinal com a variação da frequência de emissão.

Para o primeiro teste foram efectuadas medições de 50 em 50 cm. Realizou-se uma primeira medição a 25 cm, servindo esta como valor máximo de referência.

No segundo teste, colocou-se um emissor a cerca de 50 cm de distância do receptor alinhados frente a frente, procedendo-se de seguida à variação da frequência de emissão.

Na figura 8, pode ver-se a capacidade de rejeição de sinais com frequências acima dos 30 kHz. Esta resposta fica a dever-se em grande parte ao bom comportamento do filtro passa-baixo descrito anteriormente.

B.3 Detector de Contraste

A estes sensores dá-se o nome de fim de curso, dada a sua aplicação no concurso MicroRato. A função destes sensores é a detecção da área de chegada (círculo de cor preta), informando assim o robô que atingiu a zona do farol (final da primeira etapa) e que deve voltar para o ponto de partida do labirinto. Estes sensores são constituídos por elementos emissores e um receptor que mede a reflexão da luz no pavimento. Por este motivo houve necessidade de os colocar o mais próximo possível do solo e isolar da luz ambiente, de modo a melhorar a recepção por parte dos elementos detectores. A emissão de luz, é feita à custa de 4 leds vermelhos de alto brilho, enquanto que o elemento receptor é um fototransistor.

A reflexão da luz, varia de superfície para superfície e de cor para cor. Sendo assim para cores diferentes, o fototransistor irá apresentar resultados diferentes, podendo

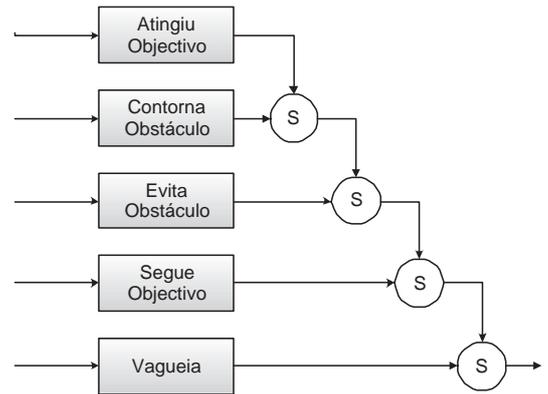


Figure 9 - Diagrama de programação por comportamentos, segundo a arquitectura de submissão

então detectar-se diferenças entre uma cor escura e outra qualquer mais clara.

B.4 Codificadores de posição

São utilizados codificadores ópticos devido ao seu baixo custo e elevada resolução. O método de codificação, consiste num disco com um padrão de sectores opacos e transparentes montado no eixo de rotação do motor, um emissor de luz e um fotodetector. Com a rotação do disco, a luz emitida é interrompida pelo padrão do disco, formando assim uma onda quadrada. De notar, que se o disco possuir dois padrões iguais desfasados 90°, fornecem duas ondas em quadratura e a resolução pode ser aumentada até 4 vezes.

A descodificação dos codificadores ópticos foi realizada na CPLD. Com a desmultiplicação de 11.5 : 1 da caixa de redução acoplada aos motores e com uma resolução dos codificadores de 500 PPR³, obtemos com apenas um canal 5750 pulsos por cada volta completa da roda do robô.

IV. PROGRAMAÇÃO BASEADA EM COMPORTAMENTOS

O método de programação usado, foi proposto por Rodney Brooks [1], denominado por *Subsumption Architecture*. É um método puramente baseado em comportamentos, separados por camadas, com execução assíncrona e com objectivos individuais. Cada camada produz uma acção e é responsável pela percepção do mundo na área em que executa. Cada comportamento possui dois estados, quando está activo, compete com os outros comportamentos no ataque aos motores, quando está inactivo não entra em luta para o controlo dos motores. Em cada instante, apenas um dos comportamentos activos ataca os motores. A selecção é feita através de um esquema de prioridades, definido segundo a importância de cada comportamento [6] e [7].

A programação comportamental implementada no Cyber-Mouse, encontra-se esquematizada no diagrama de blocos da figura 9. De seguida, apresenta-se uma breve descrição de cada comportamento:

Vagueia Tem o objectivo de impedir a paragem do robô, quando perde o contacto visual com o farol durante o percurso de ida. Tem a prioridade inferior a todos

³Pulse Per Revolution

os restantes comportamentos e está permanentemente activo.

Segue Objectivo Durante o trajecto de ida, utiliza o sensor detector de farol para determinar qual a direcção desejada e torna-se activo sempre que o farol seja detectado. No trajecto de volta, o objectivo é seguir uma trajectória obtida durante o percurso de ida. Neste caso, está sempre activo, porque conhece sempre a trajectória desejada.

Evita Obstáculo Visa impedir que o robô colida com os obstáculos à medida que se desloca em direcção ao objectivo. Torna-se activo quando algum sensor de distância atinge um limite mínimo de distância a um obstáculo e tem prioridade sobre o *Segue Objectivo* e o *Vagueia*.

Contorna Obstáculo Tem o objectivo de fazer com que o robô siga uma parede, mantendo uma distância mínima de segurança. Este comportamento é útil para resolver determinadas configurações de obstáculos, que bloqueariam o robô apenas com o *Evita Obstáculo*. Torna-se activo sempre que o *Segue Objectivo* e o *Evita Obstáculo* estejam simultaneamente activos e o obstáculo e o farol se localizem à frente do robô.

Atingiu Objectivo Este comportamento, tem a função de realizar as tarefas de sinalização no final de cada objectivo. No percurso de ida, baseia-se no detector de contraste para se tornar activo e consequentemente dar por concluído o primeiro objectivo. No percurso de volta, torna-se activo quando o robô se encontrar suficientemente perto da posição de partida, dando assim por terminado o segundo objectivo.

O último elemento desta arquitectura é o módulo **Escalonador**, que tem o objectivo de seleccionar o comportamento que vai actuar sobre os motores. A escolha é efectuada sobre os comportamentos activos, tendo em atenção as prioridades de cada um. O comportamento situado na camada mais elevada possui a prioridade mais elevada e vai decrescendo à medida que se vai descendo nas camadas.

V. SEGUIMENTO DE TRAJECTÓRIAS

Para o robô diferencial, a trajectória é descrita num espaço a três dimensões (x, y, θ) , havendo unicamente duas entradas de controlo. Além deste facto acresce que se têm restrições não holónomas. Isto significa que o conjunto de trajectórias de estado admissíveis é restrito e que o planeamento e controlo dessas trajectórias não é tarefa trivial.

Os métodos de seguimento de trajectórias podem basear-se no tempo ou em eventos, ou seja, a próxima posição desejada pode ou não ter em conta restrição temporal. No caso em estudo, vamos apenas concentrar-nos no método baseado em eventos. Assim, o planeador de trajectória terá que ser realimentado com a posição actual do robô P_r (ver figura 10), de forma a poder calcular a distância euclidiana do robô ao ponto desejado e decidir quando deve indicar como posição desejada a posição seguinte da trajectória discretizada.

O problema do seguimento de trajectória tem vindo a ser alvo de intensa investigação nos anos recentes, continuando

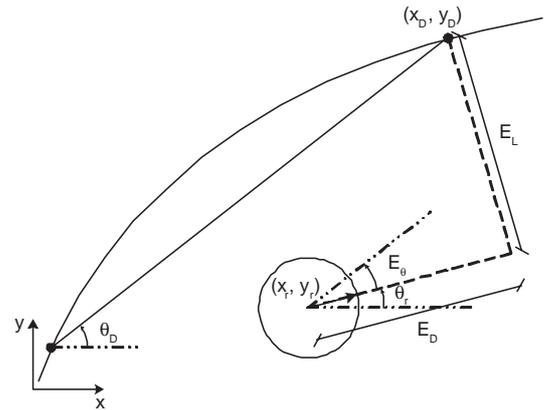


Figure 11 - Variáveis do controlo de trajectória

a ser uma questão em aberto da robótica, sendo necessário desenvolver técnicas com aplicabilidade e que sejam robustas às perturbações inerentes a estes sistemas. Do conjunto de técnicas variadas existentes, podemos referir as seguintes: [8]

- Controlador de Kanayama [2];
- Controlador de Yang [3];
- Controlador em modo deslizante;
- Controlador baseado em lógica difusa;

Neste artigo são analisados os controladores de Kanayama e Yang, por serem métodos que não exigem grandes necessidades de processamento, o que é ideal para robôs controlados por um simples microcontrolador tais como o CyberMouse. De forma a simular esses controladores, foi utilizado um programa desenvolvido em ambiente *Matlab/Simulink* [9].

A. Condução

Os controladores de condução que vamos analisar, geram o vector de velocidades (v_c, w_c) , a partir do qual, por cinemática inversa se obtém as velocidades para as rodas $(\dot{\theta}_r, \dot{\theta}_l)$. Na figura 10, apresenta-se o sistema de controlo de trajectória baseado em eventos, onde o vector P_d representa a posição e orientação desejada que é definido pelo módulo *Planeador de Trajectória*, tendo este como base a posição actual do robot (P_r) e o erro espacial máximo admitido E_{max} , que é a distância entre o robot e o ponto desejado, para definir quando deve passar para a posição seguinte. A diferença entre P_d e P_r , dá-nos os erros de posição e orientação no referencial do mundo. Através da transformação definida pela matriz T_E esses erros são transformados para o referencial do robô eq. (1), obtendo-se assim, os erros longitudinal (E_d), lateral (E_l) e de orientação (E_θ) (veja-se a figura 11).

$$E_p = T_E(P_d - P_r)$$

$$\begin{bmatrix} E_d \\ E_l \\ E_\theta \end{bmatrix} = \begin{bmatrix} \cos \theta_r & \sin \theta_r & 0 \\ -\sin \theta_r & \cos \theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x_r \\ y_d - y_r \\ \theta_d - \theta_r \end{bmatrix} \quad (1)$$

Com os erros de posição no referencial do robô definidos anteriormente e com as velocidades linear e angular de re-

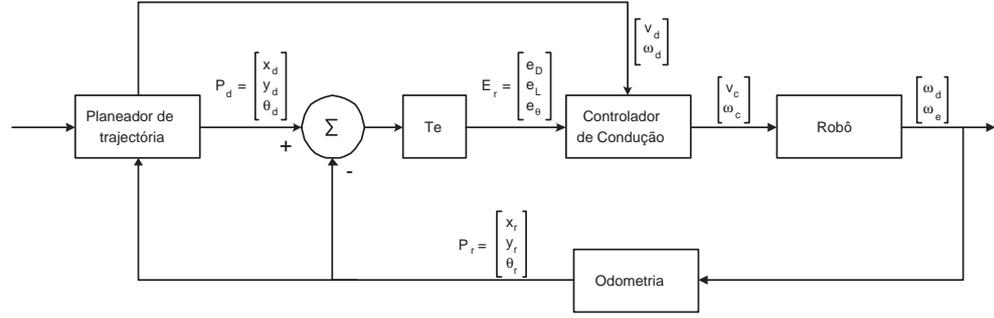


Figure 10 - Diagrama de blocos do controlador de condução de trajetórias

ferência (v_r, w_r) , obtidas do módulo *Planeador de Trajetórias*, o controlador de condução gera as velocidades lineares e angulares de controlo (v_c, w_c) para o robot, como se verá nas secções seguintes.

A realimentação de posição efectua-se com as equações de odometria em eq. (2), com base nos dados dos codificadores ópticos localizados nas duas rodas motrizes, que fornecem o deslocamento efectuado em cada período de amostragem (N_d - número de pulsos da roda direita; N_e - número de pulsos da roda esquerda).

$$\begin{cases} x[N] = x[N-1] + \Delta s \cos\left(\theta[N-1] + \frac{\phi}{2}\right) \\ y[N] = y[N-1] + \Delta s \sin\left(\theta[N-1] + \frac{\phi}{2}\right) \\ \theta[N] = \theta[N-1] + \phi \end{cases} \quad (2)$$

$$\begin{cases} \Delta s = K_r (N_d + N_e) \\ \phi = \frac{K_r}{L} (N_d - N_e) \\ K_r = \frac{\pi r}{N_{rev}} \end{cases} \quad (3)$$

Com L metade da distância entre as rodas, r o raio das rodas e N_{rev} o número de pulsos por rotação completa da roda.

B. Controlador de Condução de Kanayama

Em 1991, Kanayama propôs o uso do erro local no referencial do robô E_p , a partir da posição desejada P_d e a posição do robot P_r (figura 10) e um algoritmo de controlo proporcional para o controlo das velocidades linear e angular $[v_c \ w_c]^T$. Nas equações de controlo propostas por Kanayama eq. (4), obtidas a partir da análise da dinâmica do erro de um robô não holónomo, a estabilidade é garantida para quaisquer constantes de controlo K_x , K_y e K_θ , pelo critério de estabilidade de Liapunov [2].

$$\begin{bmatrix} v_c \\ w_c \end{bmatrix} = \begin{bmatrix} v_d \cos E_\theta + K_x E_d \\ w_d + v_d(K_y E_l + K_\theta \sin E_\theta) \end{bmatrix} \quad (4)$$

Usando as equações do factor de amortecimento ζ e frequência natural ξ deduzidas em [2],

$$\begin{aligned} \zeta &= \frac{K_\theta}{2\sqrt{K_y}} \\ \xi &= v_d \sqrt{K_y} \end{aligned} \quad (5)$$

podemos estimar os parâmetros para obter um controlador que conduz a uma resposta transitória com um bom compromisso rapidez/sobrelevação máxima.

Resultados de simulação e ajuste dos parâmetros estimados para o CyberMouse, demonstram que o conjunto de constantes que melhoram o comportamento da condução

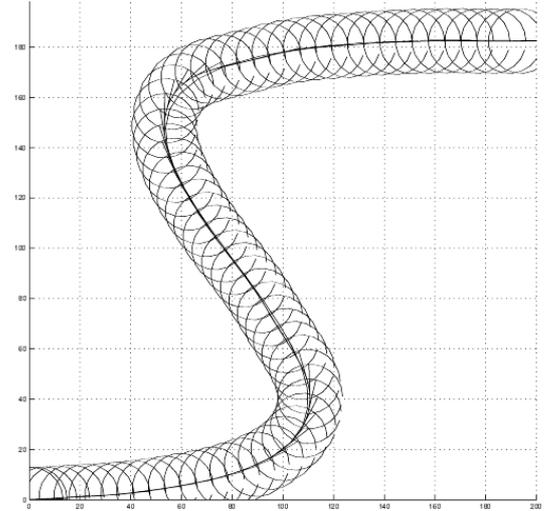


Figure 12 - Simulação do controlador de Kanayama

são: $v_d = 30 \text{ cm/s}$, $w_d = 0 \text{ rad/s}$, $K_y = 6.4 \times 10^{-3}$, $K_x = 0.1$ e $K_\theta = 0.16$.

A figura (12), apresenta o controlo de condução de uma trajetória definida por um spline cúbico, onde se pode observar que os parâmetros do controlador de Kanayama conduzem a um seguimento sem oscilações. No entanto, como este controlador é do tipo proporcional, gera sinais de comando que exigem muito esforço dos motores para pequenos erros de orientação e distância ao objectivo, ver figura (14). Este problema, pode ser parcialmente solucionado com a inclusão de um módulo que limita as velocidades linear e angular. Ficando no entanto, a persistir o problema das variações bruscas dessas mesmas velocidades.

C. Controlador de Condução de Yang

Com o objectivo de atenuar o comportamento indesejado de variações bruscas detectadas nas velocidades do controlador de Kanayama, devido a pequenos erros de orientação, Yang [3] propôs um novo controlador baseado num modelo neuronal biológico denominado *Shunting Model*, desenvolvido por Hodgkin e Huxley [10] e descrito pela seguinte equação:

$$\frac{d\varepsilon_i}{dt} = -A\varepsilon_i + (B - \varepsilon_i)S_i^e(t) - (D + \varepsilon_i)S_i^i(t) \quad (6)$$

onde ε_i é o potencial ou “actividade neuronal” do neurónio

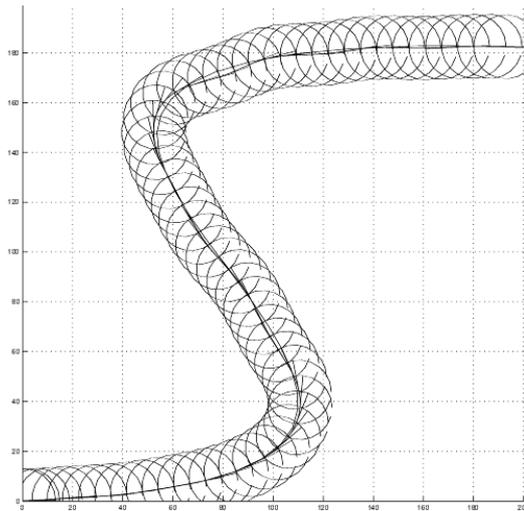


Figure 13 - Simulação do controlador de Yang

i ($\varepsilon_i \in [-D, B]$) e A , B e D são constantes não negativas que representam respectivamente a taxa de decaimento passiva, o limite superior e inferior da actividade neuronal. Os termos $S_i^e(t)$ e $S_i^i(t)$ são entradas de excitação e de inibição do neurónio i . Com inspiração no modelo da eq. (6), foi proposto o controlador de condução composto pelas seguintes equações:

$$\frac{dv_s}{dt} = -Av_s + (B - v_s)[e_d]^+ - (D + v_s)[e_d]^- \quad (7)$$

$$\frac{d\beta}{dt} = -A\beta + (B - \beta)[e_\theta]^+ - (D + \beta)[e_\theta]^- \quad (8)$$

$$\begin{cases} v_c = v_s + v_d \cos \beta \\ w_c = w_d + v_d(K_y e_l + K_\theta \sin \beta) \end{cases} \quad (9)$$

No modelo neuronal descrito, as funções que representam de excitação e inibição dos neurónios são definidas por $[x]^+ = \max\{x, 0\}$ e $[x]^- = \max\{-x, 0\}$.

Na grande maioria dos modelos baseados em *backstepping*, a velocidade linear v_d é definida como constante v_0 e a velocidade angular w_d nula. Com a finalidade de obter pequenas variações da velocidade no início e no fim do percurso, Yang definiu a velocidade linear inicial como $v_d = v_0(1 - e^{-\frac{t}{\tau_1}})$ e final $v_d = v_0(1 - e^{-\frac{s}{\tau_2}})$, onde τ_1 e τ_2 são constantes e s é a distância da posição do robô até à posição final.

A simulação do controlador de condução de Yang para uma trajetória definida por um spline cúbico (figura (13)), apresenta, tal como o controlador de Kanayama um bom comportamento, mas com a vantagem de gerar esforços de comando muito inferiores, como se pode verificar, comparando as velocidades lineares de ambos os controladores para a mesma trajetória (figura (14)). As constantes utilizadas para a simulação são: $v_d = 30 \text{ cm/s}$; $w_d = 0 \text{ rad/s}$; $K_y = 8.5 \times 10^{-3}$; $K_\theta = 0.8$; $A_1 = 20$; $B_1 = 5$; $D_1 = 5$; $A_2 = 10$; $B_2 = \frac{\pi}{2}$; $D_2 = \frac{\pi}{2}$;

D. Resultados práticos do controlador de Kanayama

A análise dos controladores de condução, termina com a implementação prática do controlador de Kanayama no

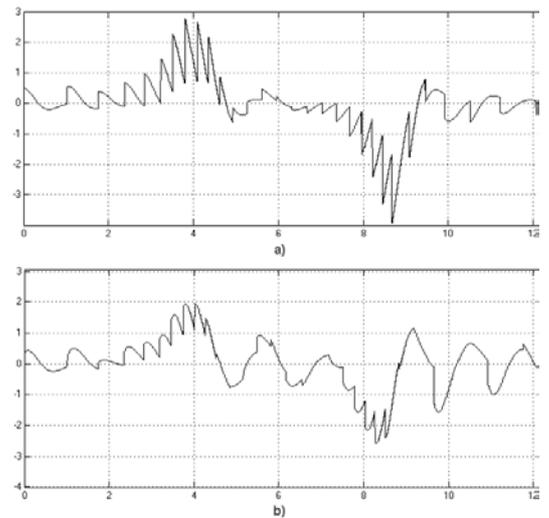


Figure 14 - Gráfico das velocidades angulares de cada controlador para a trajetória de simulação: a) Kanayama ; b) Yang

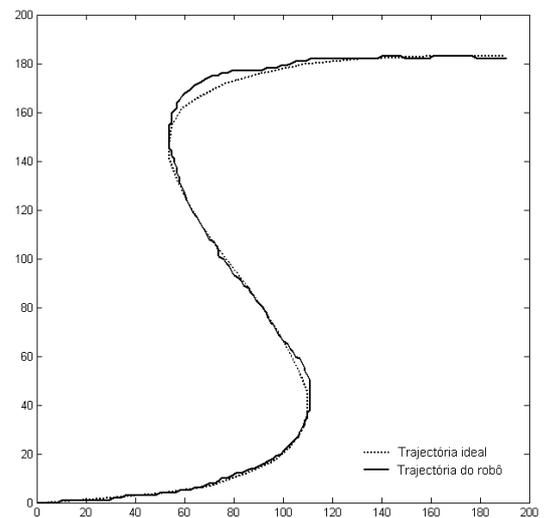


Figure 15 - Trajetória obtida, com o controlador de Kanayama implementado no CyberMouse

robô CyberMouse.

Partindo dos parâmetros obtidos por simulação para o controlador de Kanayama, verificámos que o comportamento do robô não correspondia exactamente ao desejável. Tal facto, deve-se fundamentalmente a imprecisões nos modelos dos motores na simulação. Fazendo alguns reajustes, obtivemos um seguimento de trajetória aceitável, com os seguintes parâmetros: $v_d = 30 \text{ cm/s}$, $w_d = 0 \text{ rad/s}$, $K_y = 0.01$, $K_x = 0.1$ e $K_\theta = 0.3$.

VI. CONCLUSÕES

Este artigo descreve um robô de pequenas dimensões, para o qual também foi desenvolvido um ambiente de simulação. Este conjunto, robô real e ambiente de simulação, constituem uma ferramenta de grande utilidade no âmbito do ensino de matérias básicas de robótica e mecatrónica.

Os comportamentos reactivos implementados, revelam que a abordagem proposta por Brooks é muito eficiente e pode ser executado por um simples microcontrolador.

Os controladores de condução simulados, baseados na cinemática, apresentam ambos bom desempenho. No entanto, o uso da aproximação a uma dinâmica neuronal usada por Yang, atenua o esforço de comando exigido comparativamente com o método de Kanayama.

Além dos métodos de controlo de condução referidos neste artigo, estão a ser desenvolvidos e melhorados métodos baseados em lógica difusa [11] e na teoria do controlo de sistemas não lineares.

REFERENCES

- [1] R. A. Brooks, "A robust layered control system for a mobile robot", *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.
- [2] Y. Kanayama, Y. Kimura, e F. Miyazaki, "A stable tracking control method for a non-holonomic mobile robot", *IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS '91*, pp. 1236–1241, 1991.
- [3] S. X. Yang, G. Yuan, M. Meng, e G. S. Mittal, "Real-time collision-free path planning and tracking control of a nonholonomic mobile robot using a biologically inspired approach", *IEEE International Conference on Robotics and Automation*, pp. 3402–3407, 2001.
- [4] L. Marques, U. Nunes, e A. T. de Almeida, "Modulation Method for Low-cost Implementation of Optoelectronic Reflective Sensors", *Proc. 10th Int. Conf. on Sensors, Transducers & Systems*, 2001.
- [5] H. R. Everett, *Sensors for Mobile Robots*, AK Peters, Ltd, 1995.
- [6] B. A. Seiger J. L. Jones e A. M. Flynn, *Mobile Robots*, A K Peters, second edição, 1999.
- [7] R. C. Arkin, *Behavior-Based Robotics*, The MIT Press, 1998.
- [8] U. Nunes e P. M. M. Coelho, "Robôs móveis: Dinâmica e controlo de trajectórias", *Actas do 2^o Colóquio de Física do Instituto Politécnico de Tomar*, pp. 99–132, 2001.
- [9] A. Mendes, "Simulador de robôs móveis em ambiente de concurso microrato", Tech. Report No. ISR-LCIR 2002/01, 2002.
- [10] A.L. Hodgkin e A.F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve", *J. Physiol. Lond.*, vol. 117, pp. 500–544, 1952.
- [11] L. Bento, "Simulador de robchair: Arquitectura comportamental baseada em lógica difusa", Tech. Report No. ISR-LCIR 2002/02, 2002.