D. Dinis: um Robô com Sentido de Orientação

Luís Seabra Lopes, José Nuno Lau, Luís Paulo Reis*

* Lab. de Inteligência Artificial e Ciências da Computação, Universidade do Porto

I. INTRODUÇÃO

Neste artigo descreve-se o D. Dinis, o robô que, além de ter ficado classificado em 2º lugar, ganhou o Prémio Inovação no Micro-Rato'99. A principal inovação deste robô é a utilização de uma bússola digital para orientação e cálculo da posição. O artigo descreve resumidamente o hardware e a arquitectura de controlo deste robô.

II. DESCRIÇÃO DO HARDWARE

O D. Dinis foi construído utilizando o hardware básico fornecido pela organização do concurso Micro-Rato. Todo o controlo é efectuado a partir da placa MPR11, a qual inclui o micro-controlador 68HC11. Esta placa funciona em íntima ligação com a placa ME11, onde se encontra a interface com os motores e 32KB de memória.

Para a detecção de obstáculos foram incluídos três sensores de proximidade baseados em infra-vermelhos. Estes sensores, depois de devidamente calibrados, permitem a detecção de obstáculos a distâncias até 25 cm. A partir desta distância, a presença de obstáculos pode ser confundida com o sinal do farol ou até com câmaras de filmar. Para a localização do farol e para a detecção da zona de chegada, foram incluídos dois outros sensores de infra-vermelhos. O sensor do farol está fixo, orientado para a frente e aproximadamente à mesma altura do farol. A este sensor aplicou-se ainda um canudo de protecção por forma a anular interferências da luz ambiente e de outras fontes de infra-vermelhos. O interface com todos estes sensores é feito através da placa IO_NOVA, desenvolvida no DETUA.

A principal novidade no hardware do D. Dinis é a utilização de uma bússola electrónica (Vector 2X da Precision Navigation). Esta bússola tem, para uma resolução de 1º, uma precisão de 2º (desvio padrão). Permite uma ligação SPI (master/slave). Esta interface é também fornecida pelo 68HC11 mas, infelizmente, um dos pinos estava a ser utilizado nas placas MPR11/ME11 para controlar a direcção de um dos motores. Para utilizar a SPI do 68HC11 directamente, a direcção do motor passou a ser controlada através de outro pino (livre). Esta alteração envolveu mudanças no software: a biblioteca com as rotinas de controlo dos motores foi adaptada; e no hardware: a ligação MPR-ME deixou de ser efectuada directamente mas sim através de um "flat-cable" no qual duas linhas foram trocadas. A bússola foi utilizada como "slave" do SPI. A rotina de leitura de valores da bússola foi implementada como uma máquina de estados. A execução desta rotina é desencadeada por um temporizador que dispara com uma frequência de 1000Hz.

III. CÁLCULO, VALIDAÇÃO E CALIBRAÇÃO DA POSIÇÃO

Em vários algoritmos de controlo do D. Dinis, o conhecimento da posição desempenha um papel importante. Durante os movimentos, o robô vai actualizando duas variáveis globais, x e y, que representam a sua posição no recinto. Essa actualização é feita com base na direcção do movimento, θ, dada pela bússola, na velocidade, v, e no tempo decorrido desde a última actualização, Δt. A velocidade e o tempo permitem calcular a distância percorrida, Δs, e, a partir daqui, o cálculo da nova posição é directo:

$$\Delta s = v \times \Delta t$$
$$x_{i+1} = x_i + \Delta s \times \cos(\theta)$$
$$y_{i+1} = y_i + \Delta s \times \sin(\theta)$$

O valor da velocidade utilizado nestes cálculos resulta de uma interpolação efectuada a partir de experiências realizadas para vários níveis da bateria e três valores da velocidade nominal (25, 50 e 75). Trata-se, pois, de uma interpolação em duas variáveis. Por exemplo, para valores médios da bateria, a velocidade nominal de 75 (próxima da velocidade máxima, devido a uma diferença entre os motores), traduz-se numa velocidade real do robô de cerca de 0.8 pés/segundo (23 cm/s, sendo1 pé = 29 cm).

Este método, para ter precisão suficiente, exige uma actualização frequente da posição. Uma vez que a bússola fornece cerca de 10 leituras por segundo, foi utilizada uma frequência semelhante para a actualização da posição. Isto significa que, à velocidade máxima, se pode actualizar a posição a cada 0.08 pés (2.3 cm).

O cálculo da posição segundo este método tem uma precisão, em movimento livre de obstáculos, de cerca de 1 pé (é claro que para grandes distâncias, isto é, distâncias numa ordem de grandeza acima das dimensões do recinto, a precisão será menor). Quando o robô enfrenta situações em que tem que evitar outros robôs ou sair de becos, ele perde precisão na posição. Convém, assim, que o cálculo da posição possa ser validado por outras vias.

Uma das formas de validação que foi considerada baseia-se nas dimensões do recinto: quando a posição calculada ao longo do movimento está fora dos limites do recinto, então a posição passa a ser a posição mais próxima que esteja dentro do recinto.

Contudo, existe no concurso uma fonte complementar de informação para o cálculo da posição: o farol. Por um lado, verificou-se que o sinal do farol permite calcular a distância do robô ao farol com uma precisão de cerca de 1 pé (Fig. 1). Por outro lado, a bússola permite obter a direcção do farol relativamente ao robô. Com base na direcção, na distância e nas coordenadas do farol, facilmente se calcula um valor alternativo, (x_f,y_f), para a posição do robô.

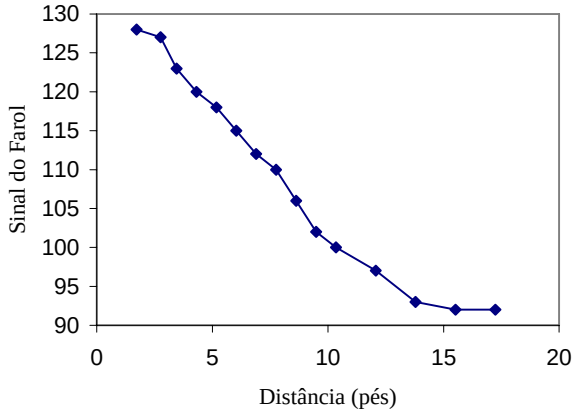

Fig. 1 – Sinal do farol versus distância

A calibração da posição é efectuada com base numa medida de confiança, $\xi \in [0,1]$, que vai sendo actualizada ao longo da execução. Essa medida é aumentada sempre que se faz uma calibração e diminuída sempre que o robô encontra situações que tipicamente reduzem a precisão na posição. Finalmente, a calibração da posição consiste em redefinir a posição actual (x,y) da forma seguinte:

$$(x, y) \leftarrow (x \times \xi + x_f \times (1-\xi), y \times \xi + y_f \times (1-\xi))$$

ou seja, o novo valor da posição passa a ser a média pesada entre o valor anterior e o valor calculado a partir do farol. O peso do valor anterior é a confiança, ξ.

IV. COMPORTAMENTOS E META-COMPORTAMENTOS

Foram definidas e implementadas diversas estratégias capazes de levar o robô desde o ponto de partida até ao farol. Estas estratégias baseiam-se na combinação de comportamentos básicos e são, por isso, designadas de meta-comportamentos.

Todos os comportamentos básicos seguem o algoritmo geral da Fig. 2.

```
inicialização
repeat
     ler sensores
     actualizar estado
     actuar motores
until condições de paragem
terminação
```

Fig. 2 – Algoritmo geral dos comportamentos

Os comportamentos implementados foram os seguintes:

- `mover_para_parede(direcção)`
- `seguir_parede(lado, direcção de terminação)`
- `mover_para_farol(direcção inicial)`
- `movimento_guiado(posição desejada)`

Para cada comportamento foram definidas as condições de paragem adequadas, sendo que todos os comportamentos terminam por limite de tempo (de modo a evitar ciclos "infinitos"). A condição de paragem de um comportamento é um dos factores que os meta-comportamentos levam em conta para seleccionar os comportamentos seguintes.

No caso de chegarem à conclusão que as suas premissas já não estão de acordo com a realidade, nomeadamente analisando as condições de terminação dos comportamentos, os meta-comportamentos terminam sem que o objectivo tenha sido atingido. Neste caso o robô passará a executar um meta-comportamento genérico.

Experimentaram-se três tipos de meta-comportamentos:

Meta-comportamento genérico - tem a pretensão de resolver "qualquer" labirinto. Não utiliza informação sobre qual o labirinto em que vai actuar. Baseia-se no algoritmo da Fig. 3.

```
repeat
   procurar direcção farol → dir
   mover para farol
   if parede
     then segue parede até voltar à direcção
dir
until área de chegada
```

Fig. 3 – Meta-comportamento genérico

O seguimento da parede tem um tempo limite que varia aleatoriamente entre valores ajustáveis. Naturalmente que o algoritmo apresentado é uma simplificação em que não é considerado o caso de o farol não estar visível.

Meta-comportamentos sequenciais – são definidos como uma sequência de comportamentos adaptada a um labirinto específico. Este tipo de meta-comportamentos utilizam pouco o comportamento `movimento_guiado` e mais os comportamentos que dependem da estrutura do labirinto nas suas condições de terminação. Foi verificado que os meta-comportamentos deste tipo são robustos e eficientes (em termos de tempo) quando o robô actua isolado. No entanto, facilmente confundem os robôs adversários com paredes, o que pode provocar terminações antecipadas dos comportamentos que os constituem e a consequente aplicação dos comportamentos seguintes em condições diferentes das previstas.

Meta-comportamento baseado em trajectórias – esta estratégia baseia-se na definição de um percurso que liga a zona de partida ao farol. Este percurso é definido pelos elementos da equipa pouco antes da prova começar. Tudo o que é necessário indicar é uma sequência de pontos, através do preenchimento de um *array* global. No caso do labirinto do Micro-Rato'99, o percurso continha apenas 4

pontos intermédios. Entre dois pontos consecutivos do percurso, o robô executa o comportamento básico `movimento_guiado`, que será descrito em mais pormenor na secção seguinte. Foi esta a estratégia utilizada no concurso. A Fig. 4 mostra a trajectória por nós especificada para o labirinto do Micro-Rato'99.

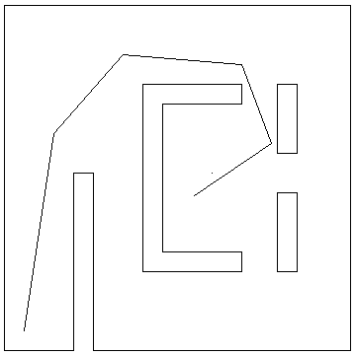

Fig. 4 – Labirinto do Micro-Rato'99 e trajectória seguida

Uma vez que duas das estratégias desenvolvidas se baseiam na especificação de planos, convinha dispôr de uma ferramenta de validação desses planos. Por isso, foi desenvolvido um simulador que permite visualizar graficamente o percurso resultante da execução de um dado meta-comportamento num dado labirinto. Isto foi utilizado no concurso e mostrou a sua utilidade.

V. MOVIMENTO GUIADO

O comportamento `movimento_guiado(posição desejada)` leva o robô desde uma dada posição inicial até à `posição desejada` (em coordenadas absolutas). Se não existirem obstáculos pelo caminho (por exemplo outros robôs), a trajectória aproximar-se-á da linha recta. Em qualquer caso, a actualização da posição baseia-se sempre na direcção efectivamente seguida.

Este comportamento segue o algoritmo geral da Fig. 2. Quando em movimento livre de obstáculos, o robô segue à velocidade máxima. Com o objectivo de manter o robô permanentemente orientado para a posição final, aplica-se uma correcção às velocidades das rodas que é proporcional à diferença entre a direcção actual e a direcção da posição final.

Quando o robô encontra obstáculos, reduz a velocidade linear e deixa de aplicar a correcção referida, adoptando uma atitude reactiva que eventualmente o leva a contornar esses obstáculos. Caso os obstáculos sejam outros robôs (ou de tamanho comparável), o D. Dinis consegue sempre contorná-los de forma eficiente. Ultrapassado o obstáculo, o robô volta a encaminhar-se para a posição final, retomando a estratégia de movimento livre.

Os obstáculos encontrados pelo robô podem ser paredes mais ou menos longas. Isto acontece, por exemplo, quando o robô, depois de ter acumulado erros nas variáveis de posição, não consegue acertar com uma passagem estreita. Se o robô começar a contornar a parede pelo lado errado, a estratégia de evitar obstáculos facilmente o leva a afastar-se da sua trajectória e, por vezes, a meter-se em becos de que é difícil sair.

Para tratar este tipo de situações, foram implementadas duas heurísticas. Uma delas impede o robô de se afastar da sua trajectória ideal (isto é, a linha recta que liga as posições inicial e final) mais do que 2 pés. Quando ele atinge este limite, roda até à direcção ortogonal à trajectória ideal. Provavelmente, continuará a ver o obstáculo que o levou a afastar-se, mas irá contorná-lo no sentido oposto.

A outra heurística baseia-se em decompor a trajectória entre dois pontos em vários segmentos. Esta heurística destina-se a obrigar o robô a seguir uma trajectória tão próxima quanto possível da trajectória ideal e, em particular, a evitar becos.

VI. RESULTADOS E CONCLUSÕES

A arquitectura de controlo do D. Dinis permite, quer a execução de estratégias puramente reactivas, quer a execução de planos definidos por um agente externo. A execução de um plano previamente optimizado para atingir um dado objectivo é uma estratégia claramente mais eficaz do que a estratégia reactiva. No entanto, a execução de um plano exige que se vá mantendo actualizada uma descrição do estado do mundo. No caso da estratégia baseada em trajectórias, essa descrição inclui a posição do robô no labirinto.

A grande inovação do D. Dinis foi a utilização de uma bússola digital para orientação (quer no movimento guiado quer nos outros comportamentos básicos) e para actualização contínua da posição.

O D. Dinis foi o robô mais regular no Micro-Rato'99, tendo terminado todas as provas. Os resultados são apresentados na Tabela I.

Tabela I – Resultados do D. Dinis no Micro-Rato'99

		Tempo	Lugar
1ª Manga	1ª Prova	1'38"	7º
	2ª Prova	43''	2º
	Resultado	43''	2º
2ª Manga		42''	3º
Final		46''	2º

AGRADECIMENTOS

Agradecemos muito os ensinamentos e apoio dos técnicos do DET Sr. Mário e Sr. Fernando.

REFERÊNCIAS

[1] *Assembly Manual Mekatronix ME11 Expansion Board for the MC68HC11 EVBU*, Mekatronix, 1997.

[2] F. Martin, *The 6.270 Robot Builder's Guide*, http://www.cs.fsu.edu/courses/ROBOTS/manual.html .

[3] H.R. Everett, *Sensors for Mobile Robots: Theory and Applications*, A.K. Peters Ltd., 1995.

[4] *MC68HC11A8: HCMOS single chip microcontroller*,

Motorola Inc., 1996

[5] *Vector™ electronic modules – Application notes*, version 1.07, Precision Navigation Inc., January 1998.

O *Concurso Micro-Rato'99* é uma iniciativa do
Departamento de Electrónica e Telecomunicações da
Universidade de Aveiro

Apoios:

Reitoria da Universidade de Aveiro
Câmara Municipal de Aveiro
Ordem dos Engenheiros – Região Centro
Fundação Luso-Americana para o Desenvolvimento
Ministério da Ciência e Tecnologia – Unidade Ciência Viva
Associação Académica da Universidade de Aveiro
Associação dos Antigos Alunos da Universidade de Aveiro
IEEE – Ramo Estudantil da Universidade de Aveiro
Diário de Aveiro
Rádio Regional de Aveiro

Patrocínios:

Telecel
Microsoft
Mekatronix
Micro-I/O