

## Bulldozer: Um robô que reconhece o seu passado

Valter Filipe Silva<sup>1</sup>, Frederico Miguel Santos<sup>2</sup>

1 – Escola Superior de Tecnologia de Castelo Branco / DETUA

2 – Instituto Superior de Engenharia de Coimbra / DETUA

**Resumo** – Este artigo descreve o robô Bulldozer IV vencedor do concurso Micro-Rato 2001. Trata-se de uma evolução das versões anteriores do Bulldozer, tendo sido adicionado um novo micro-controlador, uma bússola electrónica e um rato óptico de computador. Todo o software e hardware existente foi aproveitado, existindo apenas pequenos ajustes. Serão descritos ao longo do artigo alguns aspectos importantes, tanto na construção do hardware, como do desenvolvimento do software. É dada especial atenção ao cálculo da posição global e regresso ao ponto de partida, sem informação prévia.

**Abstract** – This article describes the Bulldozer IV robot, winner of the Micro-Rato 2001 contest. It is an evolution of previous versions to which a new micro-controller, an electronic compass and an optical computer mouse were added. All the hardware and software of the previous versions of Bulldozer are reused, with just minor changes. In this paper some important hardware and software issues are discussed, particularly global positioning in the maze and returning to initial position without any previous information.

### I. INTRODUÇÃO

O concurso Micro-Rato realiza-se na Universidade de Aveiro desde 1995. A equipa Bulldozer participa no evento desde 1998, tendo efectuado a sua 4ª participação na edição deste ano, o Micro-Rato 2001. Para tornar o concurso mais aliciante, a organização resolveu alterar as regras nesta edição, havendo agora dois objectivos a cumprir.



Fig. 1 – O aspecto físico do Bulldozer

O Robô Bulldozer (fig. 1), construído segundo as novas exigências das regras do concurso Micro-Rato, é capaz de atingir a área preta escondida no meio de um labirinto com 5x10m e assinalada por um emissor de infra-vermelhos (1º objectivo), e de seguida regressar ao ponto de partida (2º objectivo) apenas com a informação recolhida no caminho para o 1º objectivo.

Em termos técnicos, o Bulldozer IV reaproveitou todo o hardware e software existente das versões anteriores, tendo sido acrescentado uma bússola electrónica, um rato óptico de computador e um micro-controlador adicional.

### II VERSÕES ANTERIORES DO BULLDOZER

O robô Bulldozer participou nas três edições anteriores do concurso Micro-Rato em que os robôs teriam que ser capazes, no interior de um labirinto com 5 x 5 m, encontrar um área de chegada (área preta), assinalada com um farol emissor de infravermelhos, sem colidir com os obstáculos e no menor tempo possível.

As versões anteriores do Bulldozer apenas usavam a placa DET188 com o processador Intel 188 e uma placa de expansão (I/O 188) para fazer o interface com os diversos periféricos (motores, botões, sensores, etc.). Um esquema geral é apresentado na figura 2. O comportamento das versões anteriores é explicado mais à frente neste artigo.

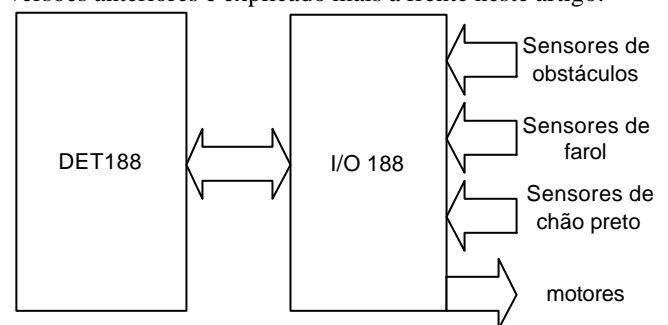


Fig. 2 – O hardware antigo

### III. DESCRIÇÃO DO HARDWARE

O hardware do Bulldozer baseia-se, tal como os anteriores, no conjunto DET188 e placa de expansão I/O188, fornecido pela organização [8]. Este conjunto possui um interface directo para 3 sensores de obstáculos, 2 sensores de farol e 1 sensor de área de chegada. A

construção de cada sub-sistema do Bulldozer IV é descrita de seguida.

*A. Detecção de Farol*

A detecção de farol é feita através de um sistema de semi-rotação [9], com cerca de 270° de amplitude. A sua posição angular é determinada através de um potenciómetro acoplado ao sistema e ligado a uma ADC. Para o controlo do motor usado na detecção de farol foi construído um circuito electrónico [1] que inverte o sentido de rotação do motor sempre que é atingida uma posição extrema.

Dada as novas dimensões do labirinto (5m x 10m) foram utilizados dois sensores de farol solidários. Um destes sensores colocado a 39cm de altura servia apenas para avistar o farol a longas distâncias (>5m), o outro montado a 28cm de altura permite a detecção precisa da localização do farol (<5m). Esta precisão é aumentada com recurso a um pedaço de tubo preto que limita o ângulo de detecção do sensor.

*B. Detecção de obstáculos*

Relativamente à detecção de obstáculos, as respectivas dimensões mínimas (10 cm) dificultam a detecção por meio de um par simples LED+sensor. Assim, para melhorar a detecção de obstáculos decidiu-se sobre-illuminar as zonas frontal e lateral. Na figura 3 é apresentada a disposição dos LED e sensores utilizados.

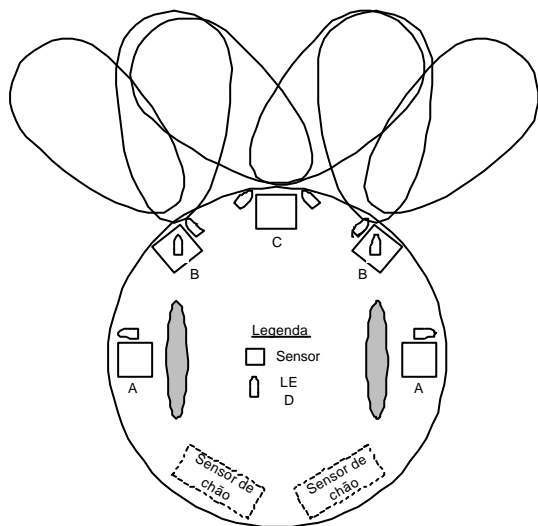


Fig. 3 – Implantação dos sensores de obstáculos

A disposição dos conjuntos B e C melhoram significativamente o comportamento perante situações de mais difícil detecção como por exemplo, esquinas.

*C. Detecção de área preta*

Para a detecção do chão preto foram utilizados dois sensores, colocados simetricamente na parte traseira do robô, para garantir a correcta imobilização na área preta, ver a figura 3.

*D. Sistema de localização*

Para permitir ao Bulldozer o regresso ao ponto de partida, recorreu-se a uma bússola electrónica (Dinsmore 1655) [2], que permite obter em cada instante a direcção do movimento com uma precisão de 1°.

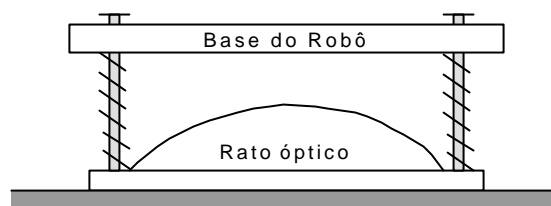


Fig. 4 – fixação do rato óptico

A distância percorrida é obtida com a ajuda de um rato óptico (Microsoft Wheel Mouse Optical), que tem uma precisão de 1/16mm, tendo sido programado para funcionar a uma frequência de 40Hz. Este componente tem diversas vantagens em relação a um odómetro de encoders, descrito em [4], entre elas:

- Não existe escorregamento do sistema em relação ao solo.
- Elevada precisão.
- Adaptação mecânica fácil.

O rato óptico foi acoplado ao robô através de molas que garantem um perfeito contacto com o solo necessário para o bom funcionamento do mesmo, ver figura 4.

Para tratar os dados provenientes dos dois periféricos atrás mencionados e para reaproveitar na totalidade o código existente de edições anteriores, foi adicionado um novo micro-controlador (Microchip PIC16F877) [3].

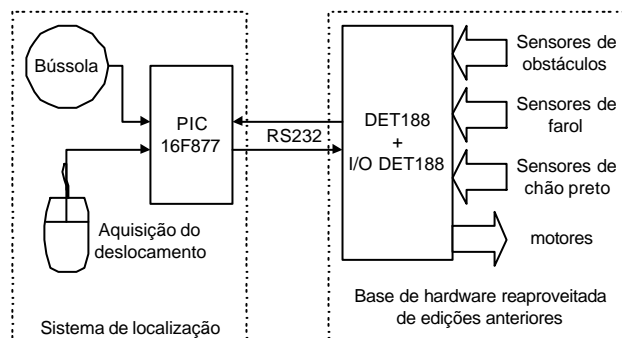


Fig. 5 – Esquema geral do hardware

O reaproveitamento de código é feito porque o algoritmo para atingir o primeiro objectivo é o mesmo que para atingir o segundo objectivo. A comunicação entre os dois processadores é feita utilizando RS232 a 9600bps. Um esquema geral do circuito utilizado é apresentado na figura 5.

### E. Bússola

O sensor utilizado tem duas saídas analógicas em forma de seno-coseno, tal como apresentado na figura 5. Estes sinais estão ligados a duas entradas analógicas do PIC.

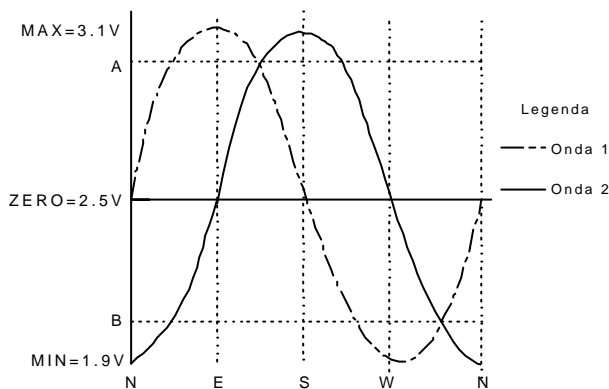


Fig. 6 – Ondas de saída da bússola

Para a obtenção de uma maior precisão utilizam-se apenas as zonas mais lineares das ondas apresentadas, entre os níveis A e B.

O código utilizado para a obtenção do ângulo  $\alpha$  (ângulo do robô em relação ao Norte) é o seguinte:

```

if (Onda1 >= A)
     $\alpha = \text{acos}((\text{Onda2} - \text{ZERO})/(\text{MAX}-\text{ZERO}));$ 
else
if (Onda2 >= A)
     $\alpha = \text{PI}-\text{asin}((\text{Onda1} - \text{ZERO})/(\text{MAX}-\text{ZERO}));$ 
else
if (Onda1 <= B)
     $\alpha = 2*\text{PI}-\text{acos}((\text{Onda2} - \text{ZERO})/(\text{MAX}-\text{ZERO}));$ 
else
if (Onda2 <= B)
{
     $\alpha = \text{asin}((\text{Onda1} - \text{ZERO})/(\text{MAX}-\text{ZERO}));$ 
    if ( $\alpha < 0$ )
         $\alpha = 2*\text{PI} + \alpha;$ 
}

```

Fig. 7 – Cálculo de  $\alpha$  (em radianos)

Apesar de se ter seguido as indicações do fabricante do componente para o cálculo do ângulo, este apresentava alguns erros, devido a: (1) as ondas não estavam desfasadas de 90°; (2) a tensão pico-a-pico era diferente.

Assim, foram feitas correcções por software baseadas em medições práticas.

Para minimizar interferências na bússola foram evitados componentes ferromagnéticos na construção do robô. No caso particular dos motores teve-se cuidado de colocar a bússola suficientemente afastada destes. Dada também a boa blindagem magnética dos motores, a influência causada por estes era praticamente nula.

### F. Rato óptico

Este componente baseia-se num circuito integrado HDNS2000 da Agilent [5], que adquire 1500 imagens por segundo, comparando a actual com as últimas para determinar a direcção e a velocidade do movimento.

O rato disponibiliza duas interfaces de comunicação – PS2 e USB, tendo sido utilizada a primeira.

O protocolo PS2 especifica uma comunicação síncrona bidireccional de 8 bits, conforme descrito em [6] e [7].

Para inicializar e configurar o rato é necessário começar por enviar alguns comandos que estão descritos em [6].

A tabela 1 apresenta a trama que é enviada pelo rato com a informação relativa ao último movimento.

	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1.	Y <sub>0</sub>	X <sub>0</sub>	Y <sub>S</sub>	X <sub>S</sub>	1	M	R	L
2.	X <sub>7</sub>	X <sub>6</sub>	X <sub>5</sub>	X <sub>4</sub>	X <sub>3</sub>	X <sub>2</sub>	X <sub>1</sub>	X <sub>0</sub>
3.	Y <sub>7</sub>	Y <sub>6</sub>	Y <sub>5</sub>	Y <sub>4</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
M	Estado do botão do meio (1=premido)							
R	Estado do botão da direita (1=premido)							
L	Estado do botão da esquerda (1=premido)							
X <sub>0</sub> -X <sub>7</sub>	Movimento na direcção X							
Y <sub>0</sub> -Y <sub>7</sub>	Movimento na direcção Y							
X <sub>S</sub> -Y <sub>S</sub>	Sentido do movimento (1=negativo)							
X <sub>0</sub> -Y <sub>0</sub>	Overflow (1=ocorreu overflow)							

Tabela 1 – Trama de comunicação

Sabendo que a precisão do rato é de 16 incrementos por milímetro e que entre cada trama podem ocorrer 256 contagens sem overflow (em cada sentido), utilizando uma frequência de 40 tramas por segundo, possibilita uma velocidade máxima de 64cm/seg para o robô.

Para se identificar o byte dentro da trama é utilizado um mecanismo de sincronismo, baseado no byte 1 que apresenta o seguinte valor: XXXX1000, já que os botões nunca são premidos.

## IV. CÁLCULO DA POSIÇÃO

Para saber a posição do robô em qualquer instante é utilizado um referencial ortogonal, com um dos eixos orientado para o norte magnético. A sua posição de arranque é assumida como (0, 0), havendo duas variáveis que mantêm a sua distância à origem em cada eixo.

Tomando como exemplo a figura 8, e assumindo o deslocamento  $\vec{D}$ , que é obtido através da informação retirada do rato (componentes  $A'$  e  $B'$ , respectivamente Byte2 e Byte3 da tabela 1), temos no referencial absoluto um deslocamento  $(A, B)$ .

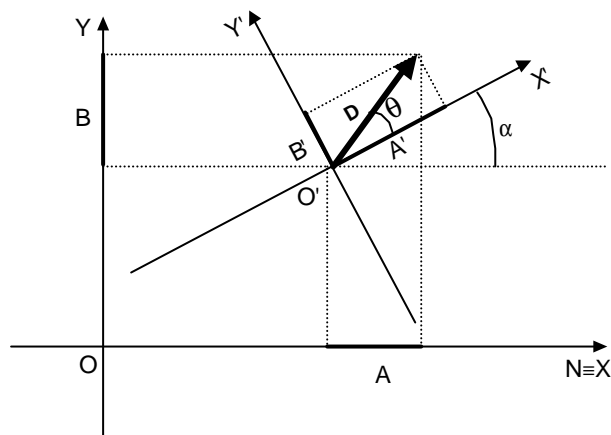


Fig. 8 – Cálculo do deslocamento

O referencial  $O'$  é o referencial do robô, fazendo um ângulo  $\alpha$  com o Norte ( $N \equiv X$ ), ângulo este obtido através da bússola – ver figura 7. Assim, o deslocamento, segundo o nosso referencial absoluto ( $A$  e  $B$ ) é dado pelas seguintes expressões:

$$q = \arctan\left(\frac{B'}{A'}\right) \quad \|\vec{D}\| = \sqrt{B'^2 + A'^2}$$

$$A = \|\vec{D}\| \times \cos(q + \alpha) \quad B = \|\vec{D}\| \times \sin(q + \alpha)$$

$$X_i = X_{i-1} + A \quad Y_i = Y_{i-1} + B$$

Os valores  $X_i$  e  $Y_i$  representam a posição actual em relação à origem  $O$ .

### V. SOFTWARE

O software que comanda o Bulldozer, está distribuído por dois processadores (Intel '188 e PIC).

O processador PIC é responsável pela cálculo da localização espacial do robô, enquanto que o Intel '188 é responsável pelas tarefas de controlo.

Toda e qualquer operação que o robô desempenha é controlada por um temporizador específico, que não deixa o robô executar a mesma tarefa infinitamente. Assim, por exemplo, o robô não anda eternamente a seguir uma parede, ou não anda eternamente a virar-se na mesma direcção à procura do farol. Desta forma o comportamento do robô é melhorado, aumentando a aleatoriedade do sistema, por vezes necessária ao bom desempenho em concurso, para evitar ciclos viciosos.

### A. Primeiro Objectivo

Para atingir a área preta (primeiro objectivo) foi utilizado o software para o '188, existente de outras participações (e já com provas dadas!) [1]. Este baseia-se no kernel de tempo real ReTMiK fornecido pela organização e disponível em [8].

O kernel gere a execução de várias tarefas cíclicas, controlando automaticamente a sua activação. Assim, o utilizador apenas tem de especificar o código e frequência de execução (que define também a prioridade) de cada tarefa.

Existem seis tarefas definidas (ordenadas por prioridade):

- STOP – período 20ms, atende os botões de arranque e paragem e é responsável pela imobilização do robô em qualquer situação;
- FAROL – período 20ms, responsável pela leitura dos sensores de farol e respectiva posição angular;
- EVITA OBSTÁCULOS – período 50ms, toma decisões relativamente à direcção a seguir;
- CONTROLO MOTORES – período 50ms, aplica aos motores a direcção obtida na tarefa anterior;
- SENSOR CHÃO PRETO – período 100ms, lê a informação dos sensores de chão preto e decide sobre a correcta posição dentro da área preta;
- RAND – período 3s, esta tarefa gera um valor aleatório.

A tarefa FAROL executa cerca de 30 leituras por varrimento, sendo cada varrimento dividido em 5 partes, conforme a figura 9.

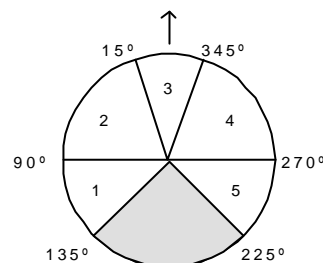


Fig. 9 – Divisão do campo de detecção de farol

Para existir uma maior precisão no alinhamento para o farol, a área 3 é a que tem menor ângulo.

O Bulldozer atinge a área preta usando o algoritmo de seguimento de paredes, apresentado na figura 10. Como se pode ver, apenas se inicia o seguimento de paredes quando o farol se encontra por detrás destas, melhorando-se o comportamento do robô, não permitindo que este comece a seguir uma parede desnecessariamente, por exemplo, a parede do labirinto. O fim do contorno é feito quando o obstáculo acaba, ou seja ao ser atingida uma esquina.

O Bulldozer consegue distinguir, em certas situações, quando um obstáculo é uma parede ou outro robô. Isto é possível, quando por exemplo se contorna uma parede pela direita e existe um obstáculo na esquerda. Esta

distinção é importante para decidir o que fazer perante tal cenário. Por exemplo, esperar que o obstáculo saia da frente em vez de se desviar deste.

Além deste algoritmo, o robô dispõe de software capaz de reconhecer certas configurações de obstáculos, conseguindo assim situar-se melhor no labirinto e obter uma melhor performance. Através deste método, o Bulldozer resolve algumas configurações de obstáculos propícias a ciclos viciosos. Na figura 11 são apresentadas duas destas configurações, e que o Bulldozer é capaz de resolver. Nesta figura é apresentada a trajectória mais provável e que leva a uma situação de ciclo vicioso.

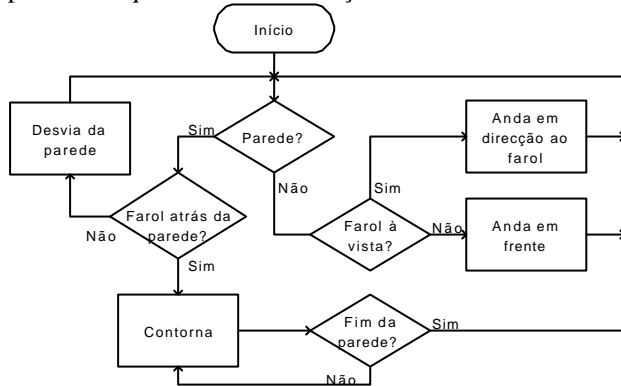


Fig. 10 – Algoritmo de contorno de parede

### B. Segundo Objectivo

Para atingir o segundo objectivo, a novidade no concurso deste ano, foi acrescentado outro processador, responsável pelo cálculo da posição (descrito em IV) e pela gestão de um sistema de faróis virtuais. Este sistema assegura a transparência para o software do DET188 (descrito no ponto anterior) na ida para a área preta, ou no regresso para o ponto de partida. Um farol virtual consiste num ponto no espaço, onde na realidade não existe um farol, mas que o DET188 interpreta como tal. O ponto (0,0), ponto de partida do robô, é um farol virtual e é o objectivo final do robô. Sempre que acaba uma parede é fixado um farol virtual, num máximo de 16, devido às limitações de memória da PIC. No percurso de regresso o robô vai tendo como objectivos os faróis virtuais fixados na ida, mas atingindo-os pela ordem inversa. Assim, e passando por todos os faróis virtuais, consegue-se uma distância mais curta no caminho de regresso, em relação ao caminho percorrido para atingir o primeiro objectivo.

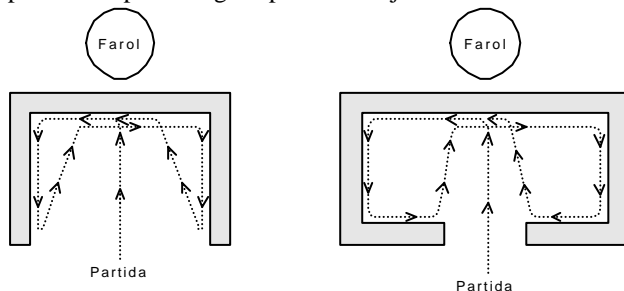


Fig. 11 – Duas configurações de obstáculos difíceis

Na figura 12 é mostrado um exemplo de funcionamento do sistema de faróis virtuais. No arranque o robô guarda o farol  $F_0$  (0,0) e tenta atingir o farol real (área preta). Neste percurso são colocados no fim de cada parede os faróis  $F_1$ ,  $F_2$  e  $F_3$ . Ao ser atingida a área preta, o robô desloca-se para a partida tendo que passar por  $F_3$ ,  $F_2$  e  $F_1$ , dirigindo-se de seguida para  $F_0$ .

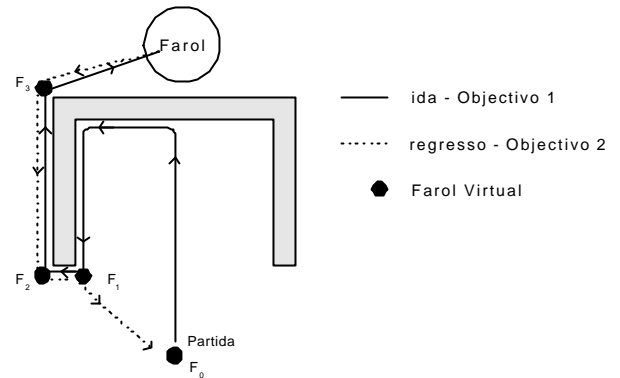


Fig. 12 – Exemplo de um percurso

Considera-se que um determinado farol virtual foi atingido quando o robô está dentro de um círculo com raio de  $\approx 0.5m$  ( $\approx 1m$  para o farol  $F_0$  (0,0)) e a distância do robô ao farol virtual é maior que a calculada no momento anterior. Isto permite que o robô se aproxime o máximo do ponto alvo mesmo para o caso em que o farol virtual se encontra num ponto inacessível (por exemplo, num obstáculo), o que pode ser possível devido a erros acumulados. O algoritmo utilizado para o uso do sistema de faróis virtuais é apresentado na figura 13.

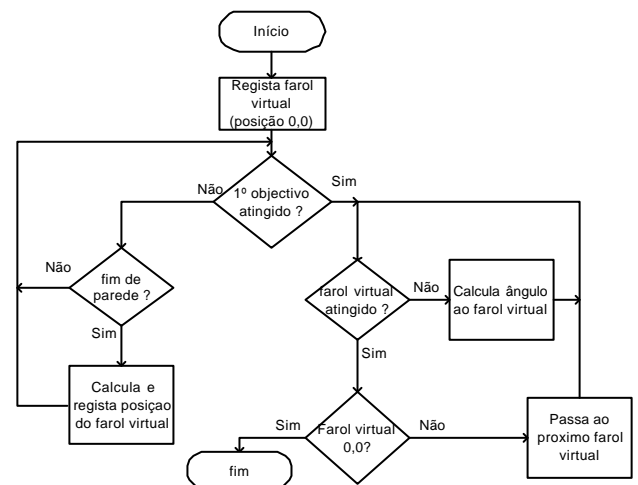


Fig. 13 – Algoritmo de aplicação de faróis virtuais

Como já foi referido anteriormente todo o software de gestão da posição do robô e dos faróis virtuais é executado no PIC, tendo este que informar, no regresso, o Intel '188 de qual a direcção angular correspondente (figura 9) em que se encontra o farol virtual. Tomando como exemplo a figura 14, pretende-se saber a direcção

angular ( $\beta$ ) na qual se encontra o farol virtual, em relação ao norte. Esta direcção é dada por:

$$\mathbf{b} = \arccos \left( \frac{X_r - X_f}{\|\vec{D}\|} \right)$$

Através desta formula determina-se o ângulo entre a direcção norte e o vector que une o robô ao farol virtual. Uma vez que a função *arccos* devolve um ângulo entre 0 e  $\pi$ , é necessário fazer uma correcção para o caso do ângulo ser superior a  $\pi$ . Esta correcção é feita com o seguinte código:

```
if (Yr - Yf < 0)
    beta = 2 * PI - beta;
```

O ângulo que a frente do robô faz com o farol virtual é dado por:

$$ang\_farol = \mathbf{b} - \mathbf{a}$$

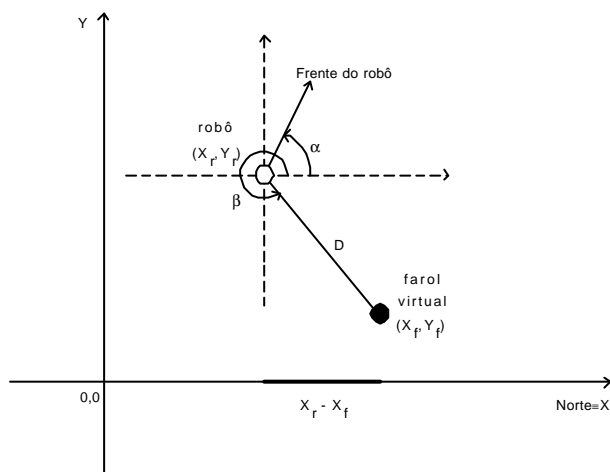


Fig. 14 – Cálculo da direcção do farol virtual

O ângulo calculado (*ang\_farol*), permite saber em que direcção angular é que se encontra o farol virtual relativamente ao robô. A informação sobre a área correspondente é enviada para o Intel '188, que a processa como se do farol real se tratasse. O envio desta informação apenas se realiza depois de atingido o primeiro objectivo. Até então o PIC vai sendo informado dos vários eventos ocorridos, por exemplo, o início da prova e o fim de parede para marcação dos faróis virtuais. Quando o segundo objectivo é atingido o '188 é informado, por forma a parar e assinalar este facto.

Um exemplo da comunicação utilizada entre os dois processadores é apresentado na figura 15.

Toda a programação do PIC foi efectuada em C com o compilador da Ht-soft [11]. Em particular, as funções trigonométricas fazem parte da respectiva livreria matemática.

## VI. RESULTADOS

A disposição dos LEDs/sensores de obstáculos e chão preto revelou-se eficaz ao longo dos 4 anos de participações, fazendo com que o Bulldozer evite os obstáculos mais problemáticos, e se consiga imobilizar na área preta sem sofrer penalizações.

A detecção de farol, também utilizado desde 1998, é muito precisa, de tal forma que o robô dirige-se para o farol sempre orientado para este.

O sistema adoptado para retornar ao ponto de partida revelou-se eficaz e de construção muito simples.

Nesta edição, o Bulldozer foi o único robô capaz de atingir o ponto de partida, segundo as novas exigências das regras do concurso.

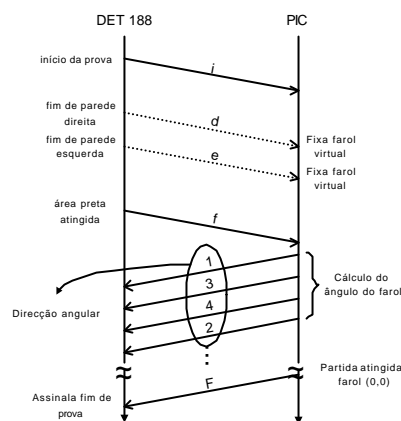


Fig. 15 – Comunicação entre os processadores

Nas quatro mangas disputadas pelo robô este conseguiu sempre atingir o farol e, em duas delas, voltou ao ponto de partida. Numa destas duas vezes o robô imobilizou-se exactamente no ponto de partida, enquanto na segunda ficou a cerca de 20 cm deste, depois de ter percorrido mais de 25m.

Nos testes efectuados antes do concurso o robô conseguiu sempre regressar ao ponto de partida, com erros de cerca de 2%, não o conseguindo em todas as mangas do concurso devido à sua velocidade baixa de locomoção e limitação do tempo de prova.

O desenvolvimento deste sistema só foi possível porque a equipa já tinha o sistema de atingir o primeiro objectivo completamente desenvolvido e testado.

## VII. CONCLUSÕES

O software do Intel '188, foi utilizado pela primeira vez no Bulldozer 98, tendo obtido o 1º lugar, e desde então sofreu apenas ligeiros ajustes. Este tem revelado uma excelente *performance* para as exigências do concurso ao longo destes anos. Este software permitiu que nas 15 mangas disputadas pelo Bulldozer (em 4 participações), apenas não terminasse uma manga, permitindo ao robô uma elevada percentagem de sucesso.

Na edição de 2001, com as novas exigências do concurso, o Bulldozer, foi o único robô a cumprir integralmente os objectivos propostos pela organização. A bússola electrónica e o rato óptico revelaram-se determinantes para a obtenção de tal feito. O uso destes dois componentes juntamente com uma arquitectura multiprocessador valeram ao Bulldozer o prémio inovação e o primeiro lugar absoluto.

O palmarés do Bulldozer nas quatro participações é indicado na tabela 2.

Edição	Competição	Prémios de Mérito
1998	1º	Engenharia e DETUA
1999	4º	
2000	2º	DETUA
2001	1º	Inovação

Tabela 2 - Palmarés

Todas as informações relativas ao Bulldozer podem ser consultadas em <http://bulldozer.4mg.com>.

#### VIII. AGRADECIMENTOS

A equipa Bulldozer agradece a todas as pessoas ou instituições que de forma directa ou indirecta contribuíram para o sucesso da equipa nas várias participações.

#### REFERÊNCIAS

- [1] Valter Silva, Telmo Silva, Frederico Santos, "Bulldozer: O Valor da Aleatoriedade", *Electrónica e Telecomunicações*, Vol. 2, Nº 6, pp. 804-805, 1999.
- [2] "Analog Hall-Effect sensor; part no. 1655", Dinsmore Instrument Company, 1998.
- [3] "PIC 16F87x Data Sheet", Microchip, 1999.
- [4] Edwin Wise, "Applied Robotics", Prompt Publications, 1999.
- [5] "HDSN2000 Technical Data", Agilent Technologies Inc., 2000.
- [6] "Synaptics TouchPad Interfacing Guide", Synaptics Inc., 1998.
- [7] Tomi Engdahl, "PC Mouse Information", disponível em: <http://digitais.ist.utl.pt/ec-asc/proj2/mouse.html>.
- [8] Concurso Micro-Rato, Departamento de Electrónica e Telecomunicações da Universidade de Aveiro, <http://microrato.ua.pt/>
- [9] João Carlos Capucho, José António Parente, "Detecção do Farol: Estudo para Aplicação no Robot *Tripé II*", Vol 2, Nº 6, pp. 806-807, 1999.
- [10] J. Borenstein, H.R. Everett, L. Feng, "Where am I? Sensors and Methods for Mobile Robot Positioning", University of Michigan, April 1996, disponível em <http://www.eecs.umich.edu/~johannb/pos96rep.pdf>
- [11] Ht-Soft Web site, [www.htsoft.com](http://www.htsoft.com)