



## SPARK: Missão *Micro-Rato* '99

Anabela A. Duarte, Paulo Jorge F. Peixoto

### I. INTRODUÇÃO

Este artigo descreve o robot SPARK baseado numa plataforma móvel circular com duas rodas motrizes e uma roda livre. Tem 3 sensores de obstáculos na parte frontal e um sensor de detecção da zona de chegada na parte inferior. É equipado com *encoders* ópticos incrementais nas rodas motrizes e uma unidade rotativa na parte superior para a detecção da posição do farol de IV.

Este robot é capaz de conciliar a execução de missões pré-programadas com um comportamento reactivo de busca do farol e desvio de obstáculos inesperados.

### II. DESCRIÇÃO DO HARDWARE

O robot é constituído, pela plataforma standard que a organização do concurso fornece como base. Possui uma armação metálica para suporte das várias placas de hardware, terminada por um painel de chapa onde estão montados os botões de controlo e o *display*. Este painel serve também de apoio à caracterização do robot, que é feita através de um pequeno boneco de plástico rígido, com cabeça rotativa.

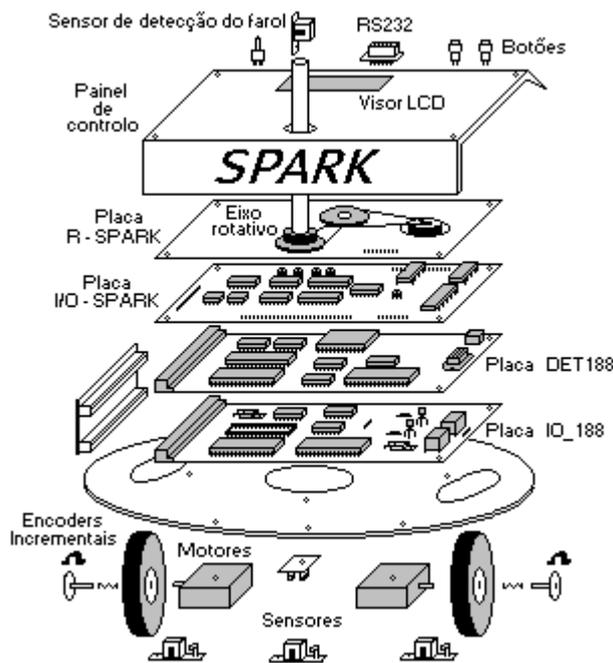


Fig. 1 – Constituição do robot Spark

IO\_188 fornecidas pela organização e nas placas específicas R-SPARK e I/O-SPARK.

A placa R-SPARK contém os mecanismos relacionados com a estrutura rotativa do sensor do farol. O movimento rotativo é conseguido com um pequeno motor com desmultiplicação que é controlado por dois relés e alimentado por um regulador independente com offset que permite variar a tensão de saída.

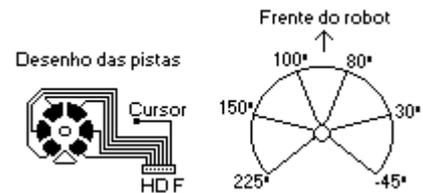


Fig. 2 – Detecção da direcção do farol.

A direcção do sensor é obtida por contacto deslizante de um cursor com uma pista seccionada, gravada em placa de circuito impresso, como mostra a Figura 2. A respectiva resolução é de 6 posições entre  $-45^\circ$  e  $225^\circ$ . Variando a velocidade do motor pode-se controlar o tempo de varrimento que poderá ir até 2 seg sem deterior o comportamento do robot.

A placa I/O-SPARK comporta a electrónica relativa aos *encoders*, *display* e entradas digitais para ligação à placa R-SPARK (fig. 3). Os *encoders* são realizados com sensores ópticos retirados de um *serial mouse* de PC, que disponibilizam cada um dois pares emissor-receptor. Assim, é possível não só obter os impulsos para contagem de incrementos como também um sinal indicador de sentido de rotação.

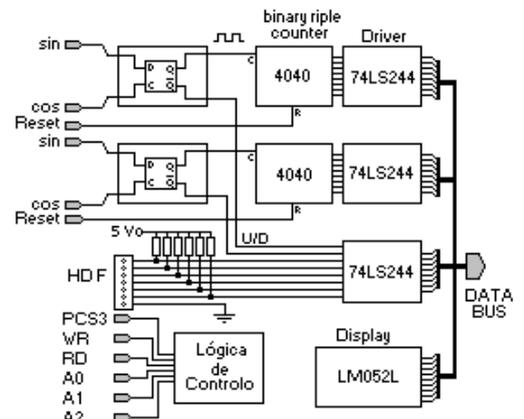


Fig. 3 – Diagrama da placa I/O-SPARK

A electrónica do robot é baseada nas placas DET188 e

### III. DESCRIÇÃO DO SOFTWARE

O software está dividido em dois grupos: as funções de interface com o hardware e as funções de controlo. A simulação do multiprocessamento necessário para este tipo de sistemas é feita através da chamada cíclica das várias funções que são modulares e apenas comunicam via variáveis globais (fig. 4).

#### A. Funções de interface com o hardware

Estas funções interagem directamente com o hardware e actualizam as variáveis globais que contêm as leituras dos sensores e botões de controlo.

**Função *init\_spark*:** Faz a inicialização do hardware e das estruturas de dados. É invocada apenas no início.

**Função *sensores*:** Lê os sensores IV de obstáculos e da zona de chegada, bem como o estado dos botões de controlo, e guarda os respectivos valores numa estrutura de dados.

**Função *le\_farol*:** Actualiza a variável que indica qual a direcção do farol. São guardados os 6 valores medidos nas várias direcções (indicadas na fig. 2) e seleccionada a direcção com maior intensidade medida pelo sensor. Este processo é relativamente robusto a falsas medidas provocadas por reflexões já que a direcção do farol só é actualizada ao fim de uma pesquisa completa (6 medidas). Esta função é também responsável pela comutação do sentido de rotação do motor.

**Função *executor*:** Decide quais as velocidades a impor a cada roda com base nas velocidades sugeridas pelas várias funções de controlo. Efectua uma filtragem simples das velocidades a impor às rodas para suavizar os movimentos do robot. Esta função é também responsável pela actualização das variáveis de posição, obtidas através dos *encoders*, que contêm a distância percorrida por cada roda.

#### B. Funções de controlo

O algoritmo de controlo do robot é composto por duas partes: por um lado, a execução de uma missão composta por tarefas isoladas, por outro lado, a execução continua de um processo de procura do farol desviando-se dos obstáculos.

##### Tarefas possíveis para execução de uma missão:

*go(distância, velocidade)*, percorre em linha recta uma distância em centímetros, com uma determinada velocidade., (representa o *offset* imposto nos motores, não sendo possível obter uma função de transferência onde seja possível calcular a velocidade exacta em metros por segundo, devido à existência de factores não equacionáveis).

*roll(theta)*, faz uma rotação segundo o ângulo em graus que é o parâmetro de entrada.

*segue\_parede(distância, lado)*, esta função implementa o seguimento de uma parede, (circula o mais perto possível da parede tentando não colidir) durante a

distância em centímetros, do lado (direito ou esquerdo).

*gotofarol()*, esta função é baseada num algoritmo cíclico de constante tentativa de progressão na direcção do farol, desviando-se dos obstáculos.

Estas funções podem ser invocadas no ciclo de uma missão mas não têm em conta a zona de chegada. Esse factor é da responsabilidade de uma função específica (*fimProva*) que analisa constantemente as variáveis necessárias e reunidas as condições imobiliza o robot sinalizando o fim de prova. Esta imobilização é efectuada tendo em conta que o robot terá que estar completamente dentro da zona de chegada.

De modo semelhante, existe uma função que detecta situações de emergência e que entra em actividade sempre que as condições de aproximação excessiva a obstáculos, ou de ciclos repetitivos na busca do farol, se verificarem. Esta função é também responsável pela monitorização do estado do botão STOP, que caso seja premido, interrompe o ciclo principal e aguarda que seja premido o botão START. No caso de uma aproximação excessiva esta função força o robot a recuar alguns centímetros retomando depois a execução do ciclo principal. Para garantir uma resposta imediata, esta função actua directamente nos motores, passando por cima da função *executor*.

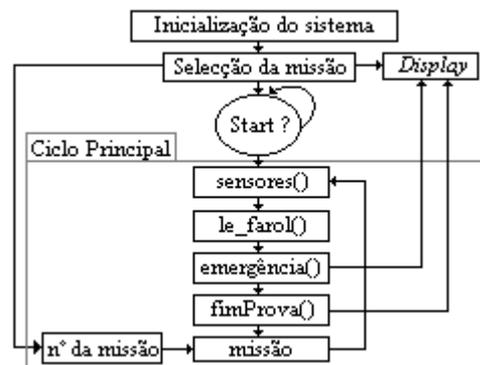


Fig. 4 – Diagrama de blocos do software

Como mostra o diagrama da figura 4, o programa principal é composto por 3 partes: inicialização, selecção da missão através dos botões no painel de controlo, e ciclo principal que inclui a execução das várias funções residentes, simulando assim o multiprocessamento.

### IV. CONCLUSÕES

Uma das grandes vantagens do sistema desenvolvido é a sua versatilidade que permite ao robot, por um lado, efectuar tarefas ou conjuntos de tarefas pré-definidas, por outro lado e especificamente para o concurso Micro-Rato, seguir um farol desviando-se dos obstáculos, sem uma trajectória pré-definida.

Refira-se, no entanto, que a originalidade do projecto e sobretudo a causa do seu êxito, é a conjugação de ambas as estratégias, ou seja, um algoritmo baseado no sistema

odométrico para pequenos percursos (pois este tem erros cumulativos) e um algoritmo baseado em sensores (i.e., reactivo) para que o objectivo final seja atingido eficazmente mesmo com obstáculos imprevistos.

Um outro aspecto de alguma importância é a hierarquização da estrutura global do algoritmo. Usando funções modulares consegue-se esconder do nível de controlo os procedimentos de baixo nível ligados ao hardware, e.g. a obtenção da direcção do farol, ou a leitura da distância percorrida por cada roda.