



---

# Robótica Móvel e Inteligente

## Assignment 2

Robotic challenge solver using the CiberRato simulation environment

88771 João Génio  
89185 Ruben Menino

Janeiro 2022

---

# Movimento e Localização

$$out_t = \frac{in_i + out_{t-1}}{2}$$

- Célula a célula.
- Orientado para o centro da próxima célula.
- No centro, rodar para  $0^\circ$ ,  $90^\circ$ ,  $\pm 180^\circ$  ou  $-90^\circ$ .

$$x_t = x_{t-1} + lin * \cos(\theta_{t-1})$$

$$y_t = y_{t-1} + lin * \sin(\theta_{t-1})$$

$$lin = \frac{out_t^l + out_t^r}{2}$$

$\theta$ , obtido através da bússola.

Fig. 1 - Equações do modelo de movimento usado

# Calibração

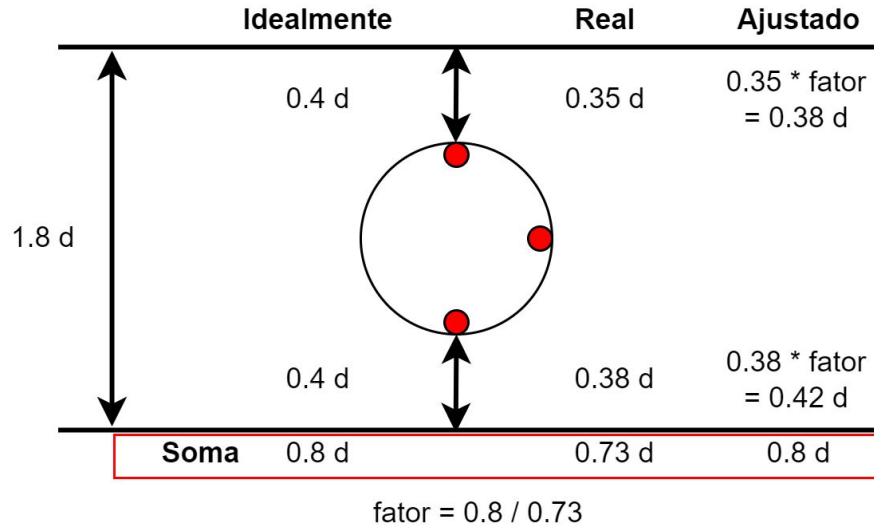


Fig. 2 - Diagrama de calibração usando dois sensores

- **Centro da célula:**  
- 0 ou 1 sensor (frente).
- **Navegação entre células:**  
- 0~2 sensores (laterais).

# Mapeamento

## (Navegação Base)

- Método usado na navegação “cega”.
- O seu uso é interrompido quando o robô revisita uma célula.

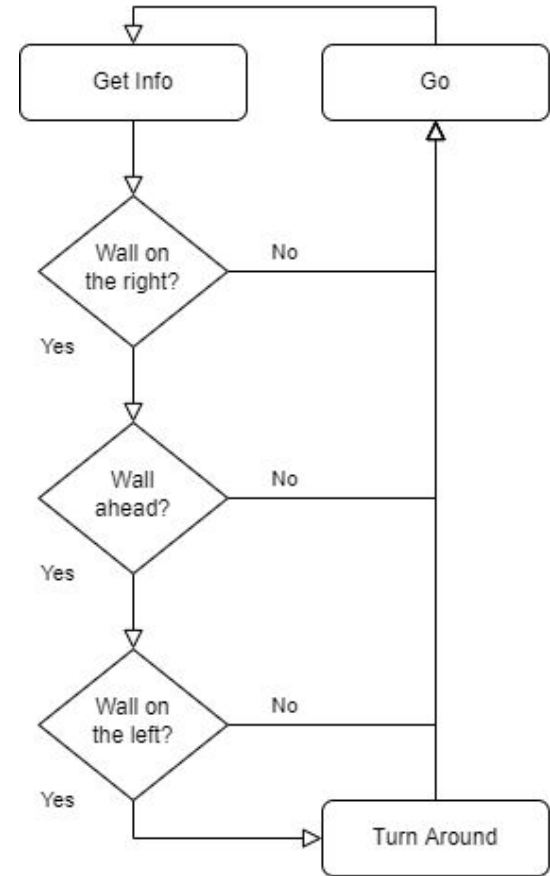
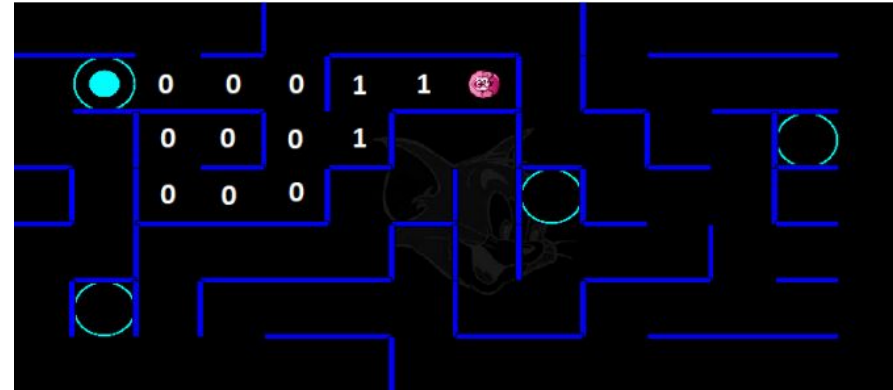
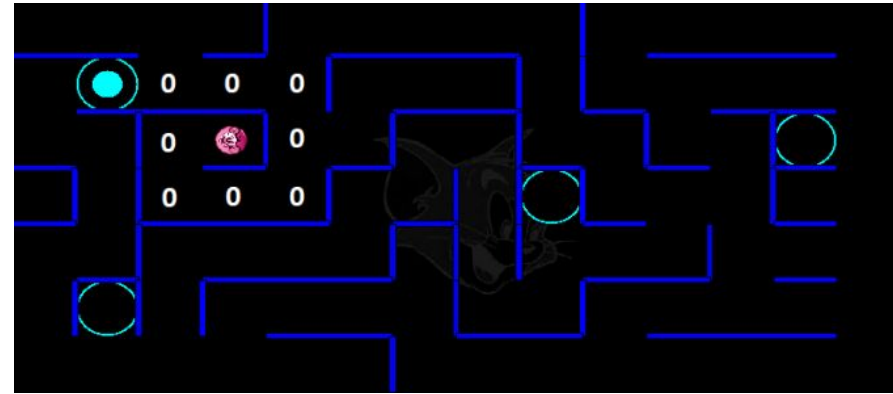


Fig. 3 - Diagrama de fluxo da decisão base de navegação

# Mapeamento

## (Circuitos)

- Circuitos são construídos usando a Navegação Base.
- Noção de células **conhecidas** mas não visitadas.

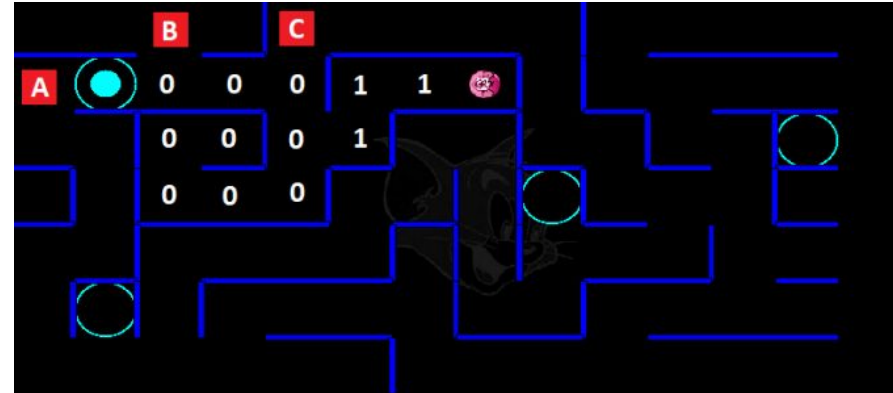
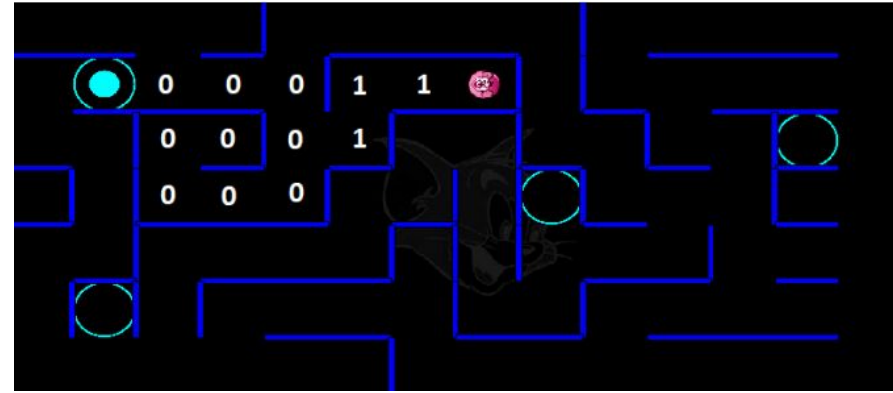


Figs. 4 e 5 - Exemplo de navegação de 2 circuitos

# Mapeamento

## (Circuitos)

- **Algoritmo de Dijkstra** para determinar o circuito (desconhecido) mais próximo.
- **Regressar à célula inicial** (e terminar o programa) quando não existirem candidatos.



Figs. 6 e 7 - Exemplo de candidatos a novo circuito.

# Planeamento

- **Algoritmo recursivo** que procura o caminho mais curto entre os beacons e a célula inicial (Dijkstra).
- **Grafo** com todas as células navegáveis.
- **Guardado o primeiro caminho com menor custo.**

```
allpaths
[78, 0, 1, 2, 3, 0]
[62, 0, 1, 3, 2, 0]
[62, 0, 2, 1, 3, 0]
[62, 0, 2, 3, 1, 0]
[62, 0, 3, 1, 2, 0]
[78, 0, 3, 2, 1, 0]
bestpath [62, 0, 1, 3, 2, 0]
```

Fig. 8 - Exemplo de todos os caminhos mais curtos possíveis para o desafio C4 (custo e ordem de beacons).

# Resultados e Conclusões

- **Mapeamento:**
  - Dependente da localização
  - Explora qualquer mapa por completo.
- **Planeamento:**
  - Não tem em conta número de rotações num percurso.
  - Dependente do mapeamento.
- **Movimento:**
  - Ponto fraco do sistema.
  - Pára em células já visitadas.
  - Velocidade maior resultaria em erros.
- **Localização:**

Pode falhar em “tick rates” mais rápidos, ou em máquinas fracas.