

Inteligência Artificial para Robótica Móvel:

CT-213

Instituto Tecnológico de Aeronáutica (ITA)

Relatório do Laboratório 1 - Máquina de Estados

Finita e *Behavior Tree*

Leonardo Peres Dias

17 de março de 2025





Sumário

1 Breve Explicação em Alto Nível da Implementação	3
1.1 Máquina de Estados Finita	3
1.2 <i>Behavior Tree</i>	4
2 Figuras Comprovando Funcionamento do Código	5
2.1 Máquina de Estados Finita	5
2.2 <i>Behavior Tree</i>	5

1 Breve Explicação em Alto Nível da Implementação

1.1 Máquina de Estados Finita

FiniteStateMachine

Esta classe mantém o estado atual e atualiza o agente executando os métodos `check_transition` e `execute` do estado ativo. A transição entre estados é realizada pelo método `change_state`.

MoveForwardState

O agente avança com velocidade constante (`FORWARD_SPEED`). Em `check_transition`, se o sensor de colisão (`get_bumper_state()`) detectar um obstáculo, ocorre a transição para `GoBackState`; se o número de ciclos (contados por `decisions`) exceder

$$\frac{\text{MOVE_FORWARD_TIME}}{\text{SAMPLE_TIME}},$$

a transição é para `MoveInSpiralState`. O contador é incrementado a cada ciclo.

MoveInSpiralState

Se houver colisão, a FSM transita para `GoBackState`; após

$$\frac{\text{MOVE_IN_SPIRAL_TIME}}{\text{SAMPLE_TIME}},$$

a máquina retorna para `MoveForwardState`. Em `execute`, calcula-se o tempo, o raio (como $\text{INITIAL_RADIUS_SPIRAL} + \text{SPIRAL_FACTOR} \cdot t$) e a velocidade angular ($\frac{\text{FORWARD_SPEED}}{r}$).

GoBackState

O agente recua com velocidade `BACKWARD_SPEED`. Após

$$\frac{\text{GO_BACK_TIME}}{\text{SAMPLE_TIME}},$$

o estado muda para `RotateState`.

RotateState

Um ângulo aleatório entre $-\pi$ e π é amostrado e o tempo de rotação é

$$\frac{\text{angle}}{\text{ANGULAR_SPEED}}.$$

Ao completar esse período, o estado retorna para **MoveForwardState**. Durante a rotação, o agente aplica velocidade angular de **ANGULAR_SPEED**.

1.2 Behavior Tree

BehaviorTree & TreeNode

A classe **BehaviorTree** possui uma raiz (do tipo **TreeNode**) e executa o método **execute** deste nó. **TreeNode** é abstrata e define os métodos **enter** e **execute**.

RoombaBehaviorTree

Implementa uma árvore para o robô Roomba com um nó seletor contendo duas sequências:

- **Sequência 1:** **MoveForwardNode** seguido de **MoveInSpiralNode**.
- **Sequência 2:** **GoBackNode** seguido de **RotateNode**.

Leaf Nodes

MoveForwardNode: Comanda o robô para avançar com velocidade constante; retorna **FAILURE** em caso de colisão e **SUCCESS** após um tempo pré-definido.

MoveInSpiralNode: Executa movimento em espiral calculando o raio e a velocidade angular; falha se houver colisão e retorna **SUCCESS** após o tempo estipulado.

GoBackNode: Comanda o recuo com velocidade definida; retorna **SUCCESS** após um período determinado.

RotateNode: Realiza rotação com ângulo aleatório (entre $-\pi$ e π) e retorna **SUCCESS** após completar o tempo de rotação.

2 Figuras Comprovando Funcionamento do Código

2.1 Máquina de Estados Finita

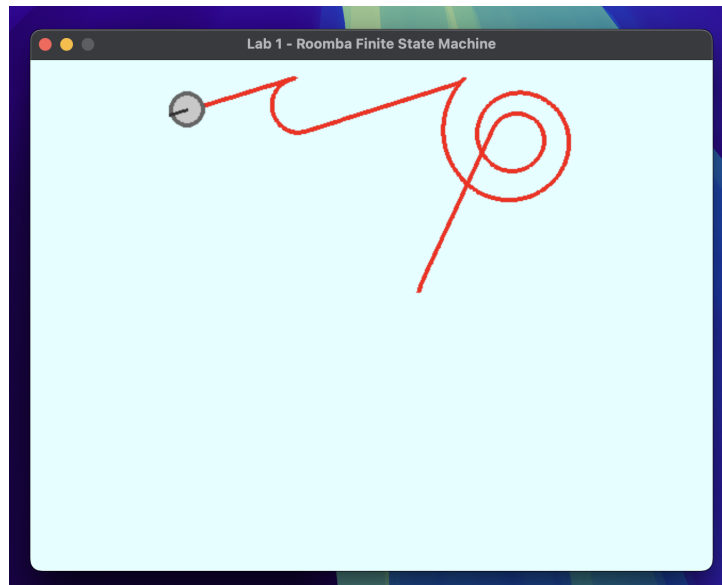


Figura 1: Máquina de Estados Finita

2.2 *Behavior Tree*

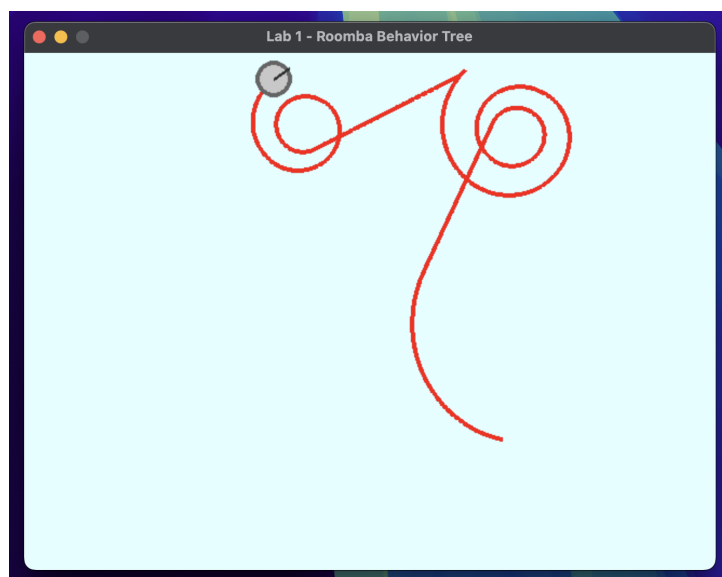


Figura 2: *Behavior Tree*