

Máquina de Estados Finita e Behavior Tree

Samara Ribeiro Silva

Instituto Tecnológico de Aeronáutica, Laboratório de Inteligência Artificial para Robótica Móvel (CT-213). Professor Marcos Ricardo Omena de Albuquerque Máximo, São José dos Campos, São Paulo, 15 de março de 2021.

1. Máquina de Estados Finita

Nas figuras 1 e 2 é possível observar o comportamento do robô utilizando a máquina de estados finita.

Figura 1 : Trajetória do robô (Move Forward → Move in Spiral → Move Forward → Move in Spiral (incompleta) para a state machine.

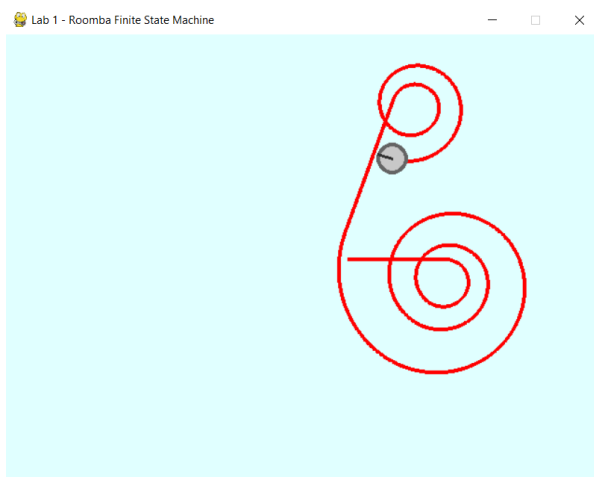
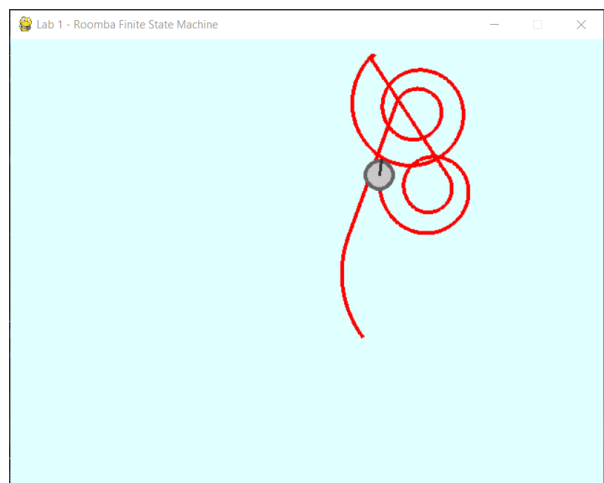


Figura 2 : Trajetória do robô (Move in Spiral → Go back → Rotate → Move Forward → Move in Spiral para a state machine.



Para a máquina de estados finitos no método *check_transition* de cada classe dos estados foram checadadas as condições para transições de estado: **tempo** para todos os estados e **colisão** para os estados *Move Forward* e *Move in Spiral*. Foram utilizados os valores de tempo sugeridos para *Move Forward*, *Move in Spiral* e *Go Back*. Já para o *Rotate*, o tempo limite foi calculado foi $t_R = \frac{[angulo]}{ANGULAR\ SPEED}$. Já no método *execute* foram realizados os cálculos da velocidade angular para os estados *Move in Spiral* e *Rotate* e para todos os estados foi definida a velocidade do robô através da função *set_velocity* e incrementado o contador de tempo.

2. Behavior Tree

Nas figuras 3 e 4 é possível observar o comportamento do robô utilizando a behavior tree. Note que as figuras 1 e 2 possuem um comportamento semelhante devido a aleatoriedade do movimento ser inserida apenas após a primeira colisão.

Figura 3 : Trajetória do robô (Move in Spiral → Move Forward → Move in Spiral (incompleta) para a behavior tree.

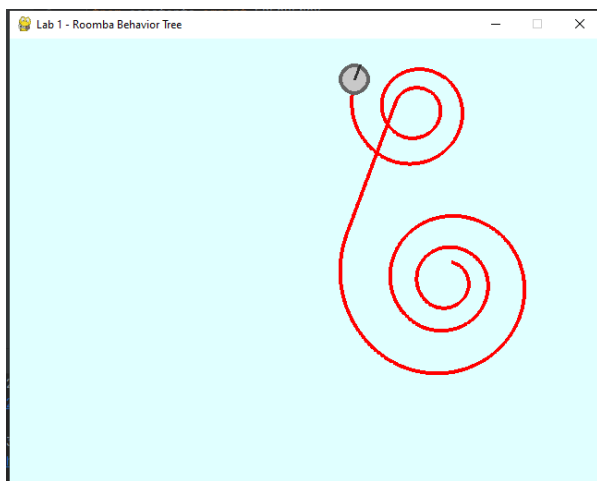
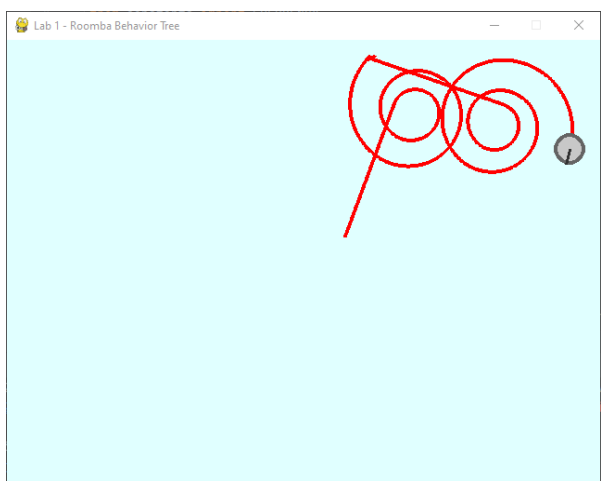


Figura 4 : Trajetória do robô (Move Forward → Move in Spiral → Go back → Rotate → Move Forward → Move in Spiral para a behavior tree.



Para a Behavior Tree no método *enter* de cada classe das folhas as variáveis foram reiniciadas: o contador de tempo para todas as classes, a velocidade angular e o raio para a *Move in Spiral* e o ângulo e o tempo de rotação para a *Rotate*. Já no método *execute* foram checadas as condições de retorno (SUCCESS, RUNNING ou FAILURE) que são:

| | <i>Move Forward</i> | <i>Move in Spiral</i> | <i>Go Back</i> | <i>Rotate</i> |
|---------|---------------------|-----------------------|----------------|---------------|
| SUCCESS | $t > t_1$ | $t > t_2$ | $t > t_3$ | $t > t_R$ |
| RUNNING | $t < t_1$ | $t < t_2$ | $t < t_3$ | $t < t_R$ |
| FAILURE | $colidiu = True$ | $colidiu = True$ | - | - |

Antes de retornar RUNNING são realizados os cálculos necessários e a velocidade é inserida através da função *set_velocity*.

3. Equações utilizadas

Raio no estado *Move in Spiral*

$$r = r_o + bt$$

Velocidade Angular no estado *Move in Spiral*

$$\omega = \sqrt{\frac{v^2 - b^2}{r^2}}$$

Tempo de rotação no estado *Rotate*

$$t_R = \frac{|\alpha|}{ANGULAR\ SPEED}, \text{ onde } -\pi < \alpha \leq \pi$$