

Controle de um Sistema de Robôs Autônomos em um Armazém

Data de entrega: até 14 de março de 2025.

Loca: Embedded, sala 105.

Deverá ser feita uma defesa oral.

Descrição geral

Em um armazém automatizado, dois robôs móveis transportam caixas de insumos entre um Buffer de Entrada (BE) e máquinas de processamento. Cada robô tem uma rota específica e restrições operacionais que garantem a segurança e eficiência do transporte.

Os robôs seguem trajetórias predefinidas, e eventos como solicitações de transporte, falhas ou congestionamentos podem ocorrer. A modelagem em Supremica ajudará a gerar um supervisor que controla os robôs garantindo operação segura e eficiente.

Objetivos

O projeto tem como objetivo desenvolver um **modelo de controle supervisionado** para um sistema de robôs móveis em um armazém utilizando **autômatos finitos** e a ferramenta **Supremica**. Os principais objetivos são:

- 1. Modelar o sistema de transporte de insumos**
 - a. Representar o comportamento dos robôs usando autômatos finitos.
 - b. Definir estados e transições para o transporte de caixas entre o **buffer de entrada** e as **máquinas de processamento**.
- 2. Garantir que apenas um robô tem acesso ao buffer de entrada**
 - a. Restringir o acesso simultâneo de mais de um robô ao buffer de entrada.
 - b. Implementar eventos de espera para evitar conflitos de acesso.
- 3. Evitar colisões entre os robôs**
 - a. Definir regras que impeçam que os robôs tentem acessar a mesma área ao mesmo tempo.
 - b. Criar restrições que forcem um robô a aguardar caso outro já esteja na região crítica.

4. Implementar a especificação e gerar o supervisor

- a. Modelar **restrições de segurança** usando **autômatos de especificação**.
- b. Utilizar o **Supremica** para a **síntese do supervisor**, garantindo que o sistema opere corretamente.

5. Simular e validar o comportamento do sistema

- a. Testar diferentes cenários de transporte para verificar se o supervisor evita colisões e bloqueios.
- b. Avaliar a performance do sistema quanto à entrega eficiente dos insumos às máquinas.

Deverão ser aplicados os **conceitos de controle supervisionado, eventos controláveis e não controláveis**, e **síntese de supervisores**.

Funcionamento do sistema

Principais componentes

- a. Número de robôs: 3 robôs autônomos (R1, R2 e R3).
- b. Número de máquinas: 4 máquinas de processamento (M1, M2, M3, M4).
- c. Buffer de Entrada (BE): local onde os robôs retiram os insumos antes de entregá-los às máquinas.

Regras do operacionais

1. Rotas dos robôs

- a. R1: Transporta caixas do Buffer de Entrada (BE) para as máquinas M1 e M2.
- b. R2: Transporta caixas do Buffer de Entrada (BE) para as máquinas M3 e M4.
- c. R3: substitui R1 ou R2 em caso de falha.

2. Buffer de Entrada (BE)

- a. Apenas um robô pode acessar o BE por vez para evitar conflitos na retirada de insumos.
- b. Se um robô estiver retirando um insumo, o outro deve aguardar.

3. Evitar colisões (opcional)

- a. Como ambos os robôs passam pelo BE, é necessário um controle para evitar colisões na área.
- b. Os robôs devem aguardar se o outro estiver no buffer.
- c. **Se essa regra não for implementada, a regra do buffer de entrada (2) se torna desnecessária.**

Eventos

A seguir você tem uma lista de possíveis eventos para esse sistema. Dependendo de seu modelo, a lista de eventos (ou até mesmo os parâmetros dos eventos) pode ser ligeiramente diferente. Por exemplo, como só temos um BE, talvez não faça sentido colocar o BE como parâmetro nos eventos.

Eventos controláveis

- $move(R_i, BE, M_x)$: R_i transporta uma caixa do BE para a máquina M_x .
- $wait(R_i)$: R_i aguarda caso o BE esteja ocupado.
- $unload(R_i, M_x)$: R_i entrega a caixa na máquina M_x .

Eventos não controláveis

- $request(BE, M_x)$: uma nova solicitação de transporte surge.
- $robot_fault(R_i)$: o robô R_i apresenta falha.
- $robot_reset(R_i)$: o robô R_i volta a funcionar normalmente.
- $collision_risk(BE)$: o sistema detecta risco de colisão no BE.

Entregáveis

Você deverá entregar 3 artefatos:

1. Arquivo com o modelo Supremica do projeto;
2. Documentação;
3. Vídeo no YouTube com demonstração e explicação do código.

Você deverá enviar um link para o repositório no GitHub com seu modelo. **Adicione o usuário kyllercg@gmail.com com permissão de leitura ao repositório. Você só precisa enviar esse link.** Todos os demais itens devem estar contidos no repositório. **Envie 'link para kyller@dee.ufcg.edu.br.**

A documentação do projeto deverá estar no GitHub (README). Nela, você deve explicar o que foi feito, como o sistema funciona. Adicione figuras para ajudar na compreensão do modelo.

O vídeo de demonstração não deverá exceder mais do que 10 minutos.

Outras coisas

1. Pode ser feito em grupos de até 3 pessoas.
2. Onde baixar o Supremica?
 - [Release Waters/Supremica IDE 2.7.1 · robimalik/Waters · GitHub](#)

3. Lembre-se que você precisa instalar também:

- [Java | Oracle](#)
- [Graphviz](#)

4. Mini tutorial do Supremica

- [GitHub - kyllercg/tutorial-supremica: Minitutorial sobre como usar o Supremica.](#)