

Laboratório 1 - Sistema Logístico Autônomo de Transporte

Universidade de Brasília
Departamento de Ciência da Computação
CIC0248 – Sistemas de Tempo Real – 2022/2
21 de janeiro de 2023

Davi Salomão Soares Corrêa

18/0118820

davi.salomao@aluno.unb.br

Francisco Henrique da Silva Costa

18/0120174

francisco.henrique@aluno.unb.br

Matheus Teixeira de Sousa

18/0107101

teixeira.sousa@aluno.unb.br

1. Objetivos

Este laboratório é desenvolvido no contexto da disciplina de Sistemas de Tempo Real da Universidade de Brasília e tem por objetivo explorar os conhecimentos adquiridos até então sobre sistemas de tempo real através de uma aplicação prática no projeto de um sistema logístico autônomo de transporte. No laboratório 1, escopo deste trabalho, será tratado da definição da situação problema, das tarefas envolvidas, dos requisitos temporais e da validação lógica do projeto.

2. Introdução

A logística é uma área da administração responsável pela gestão, armazenamento e distribuição de recursos para uma determinada atividade [3]. Dentro do contexto da engenharia, essa área pode envolver situações como o planejamento de compra e de distribuição de material, definição de escalas de manutenção de máquinas, planejamento da produção. É com base nessa ideia que o projeto proposto busca adaptar um sistema logístico autônomo de transporte como empregado na montadora da Porsche [1].

Durante o processo de montagem do Porsche 911, um veículo autônomo faz o transporte dos motores produzidos na oficina de montagem para a linha de montagem principal, em outro ponto da fábrica. Para isso, o veículo conta com um sistema de orientação baseado em ímãs colocados abaixo do piso de concreto, de modo que o sistema consegue se orientar e encontrar o caminho correto. Além disso, como o ambiente da fábrica é altamente colaborativo, cada veículo deve ter uma série de mecanismos de segurança para evitar colisões.

No entanto, antes de efetivamente tratar do detalhes do

projeto proposto, as próximas seções da introdução são dedicadas à revisão de conceitos importantes da teoria aplicada na modelagem do problema e à descrição do UPPAAL, *software* que será utilizado para validação. Depois, na seção Metodologia, tanto o modelo quanto as tarefas propostas serão detalhados. Então, por meio de simulação, cada tarefa será analisada e os resultados serão apresentados e discutidos nas seções Resultados e Discussão e conclusões, respectivamente.

2.1. Sistemas de tempo real

Os sistemas computacionais com requisitos de tempo de real, ou sistemas de tempo real, são aqueles submetidos a requisitos de natureza temporal não triviais, ou seja, aqueles que possuem parâmetros como um prazo máximo para execução de uma tarefa ou um intervalo máximo entre duas ações [4]. Dessa forma, os resultados fornecidos por esse tipo de sistema devem estar tanto lógicos quanto temporalmente corretos. Dentre os exemplos existentes, é possível destacar o sistema de defesa anti-míssil, que deve ser capaz de detectar, rastrear e interceptar o míssil dentro de um intervalo de segundos.

2.2. Tarefas e suas propriedades

Para fins de projeto e análise, um sistema pode ser dividido em unidades fundamentais, denominadas tarefas, que representam um bloco de código com atributos temporais próprios. Cada tarefa possui um instante máximo desejado para sua conclusão, chamado de *deadline*. O *deadline* é requisito temporal mais relevante na análise de um sistema e pode ser dividido em dois tipos, relativo e absoluto. Este define um instante para conclusão em relação ao UTC (Tempo Universal Coordenado), enquanto aquele define o intervalo

para conclusão a partir do instante de chegada da tarefa.

Ainda em relação ao *deadline*, é possível definir outra classificação, quanto a criticidade da perda do prazo de conclusão da tarefa. Se a perda do *deadline* resulta em consequências catastróficas, este é chamado de *hard*. No entanto, se a perda não tem consequências catastróficas e a conclusão da tarefa tem valor, mesmo com atraso, este é chamado de *soft*. Por fim, se a perda não tem consequências catastróficas, mas a conclusão da tarefa não tem valor fora do prazo, este é chamado de *firm* [4].

Quanto a periodicidade, cada tarefa pode ser classificada em três tipos, periódica, esporádica e aperiódica. A periódica é ativada (chamada para execução) em um intervalo conhecido e constante de tempo, chamado período. No entanto, quando a tarefa não possui um instante de ativação conhecido, mas possui um intervalo mínimo entre uma ativação e outra, esta é chamada de esporádica. Por fim, quando a restrição de intervalo mínimo não existe, a tarefa é chamada de aperiódica e não se conhecem informações temporais sobre a ativação [4].

Ainda é possível classificar a tarefa por meio do grau de importância na execução ou pela prioridade. Usualmente, a prioridade é um valor inteiro maior ou igual a zero e tarefas com valores de prioridade menores são mais importantes que aquelas com valores maiores. Além disso, outras propriedades temporais relevantes são: tempo de execução no pior caso (WCET), tempo de bloqueio no pior caso, interferência e tempo de resposta máximo.

Primeiro, o tempo de execução é o tempo necessário para a conclusão de uma tarefa, desconsiderando a interferência de outras tarefas. Depois, o tempo de bloqueio no pior caso compreende o tempo em que uma pode passar bloqueada por outra de maior prioridade em função do acesso a recursos compartilhados. Então, a interferência ocorre quando uma tarefa é atrapalhada por outra de maior prioridade. Por fim, o tempo de resposta máximo é o intervalo de tempo desde a chegada da tarefa até a sua conclusão [4].

2.3. Microkernel

A parte principal do sistema operacional é chamada de kernel, responsável pelo controle direto dos recursos do *hardware*. No contexto da disciplina, o serviço mais importante oferecido pelo sistema operacional é a gerência dos recursos do processador por parte das tarefas e o tratamento de interrupções. Quando o kernel fornece apenas os serviços mais básicos de gerência do processador, ele é chamado de microkernel. No microkernel, as tarefas de tempo real são implementadas como *threads*.

Cada *thread*, forma na qual um processo divide a si mesmo em duas ou mais tarefas para execução concorrente no processador, solicita um serviço ao microkernel por meio de chamadas de sistema na forma de interrupções de *software*. Neste momento, o microkernel gerencia e responde



Figura 1: Veículo para transporte do motor [6]

a solicitação recebida. Este tipo de sistema é chamada de sistema reativo, porque não faz nada (permanece em estado idle) até que ocorra um evento via interrupção de software (por meio de uma chamada de sistema de uma *thread*) ou via interrupção de *hardware* [4].

2.4. UPPAAL

O UPPAAL é um *software* de desenvolvimento integrado para modelagem, validação e verificação de sistemas de tempo real modelados na forma de redes autômatas temporizadas [5]. No contexto do laboratório 1, o UPPAAL será utilizado para projeto e validação lógica das tarefas propostas para o sistema logístico projeto.

3. Metodologia

A partir da inspiração do modelo implementado na Porsche, apresentado na introdução, o projeto proposto visa simular a atuação de controle durante o deslocamento de um veículo autônomo (figura 1) cuja função é transportar o motor da área de fabricação para a linha de montagem principal. Ao longo do deslocamento, o ambiente da fábrica impõe estímulos que exigem atuação dinâmica baseada em requisitos temporais, como mudanças de trajetória e detecção de obstáculos.

Em uma situação real, a perda dos requisitos temporais podem gerar problemas de diferentes níveis, como atraso na produção, colisão com operador, danificação do motor. A figura 2 apresenta o mapa proposto para a fixação dos ímãs, os quais podem ser distribuídos de forma aleatória no mapa conforme seja conveniente no *layout* de fábrica. O único critério a ser obedecido é que a distância máxima seja, no máximo, de cinco metros entre eles.

Sensores de proximidade, tais como sensor de efeito Hall, capacitivo e indutivo, geralmente apresentam faixa de

detecção na ordem de milímetros, nesse sentido foi adotado o sensor magnetômetro em conjunto com ímãs super potentes para que seja compatível à faixa de medição na ordem de metros. Os campos magnéticos emitidos pelos ímãs são os referenciais do veículo para ajuste da direção entre dois *checkpoints*. No ajuste de rota, os ruídos do ambiente e o modelo cinemático do veículo implicam em trajetória não-linear. As tarefas propostas para o projeto estão resumidas na tabela 1.

Tarefas	Prioridade	Deadline	Período
Ajustar rota	3	0,5 s	esporádica
Alarme	1	1 s	aperiódica
Detectar obstrução	2	0,5 s	aperiódica
Garra	1	20 s	esporádica
Seguir rota	4	10 min	periódica

Tabela 1: Tarefas e seus requisitos temporais

3.1. Descrição das tarefas

A tarefa **Ajustar rota** é uma tarefa que ocorre quando o veículo passa por um dos *checkpoints*, é definida esporádica, porque sua chamada depende da velocidade do veículo e da ocorrência de outros eventos. Apesar disso, é de conhecimento que o menor intervalo de ativação ocorre quando o veículo emprega velocidade máxima no deslocamento e não ocorrem outros eventos entre dois *checkpoints*.

A tarefa **Alarme** ocorre quando uma obstrução superior a três unidades de tempo é detectada. Como depende de uma obstrução, esta tarefa é aperiódica. Esta tarefa serve para informar que ocorreu algum problema durante o transporte e é necessário uma intervenção manual para identificar a causar e tratar da solução.

A tarefa **Detectar obstrução** é uma tarefa aperiódica, uma vez que não se sabe quando um obstáculo se colocará diante do veículo. Esta tarefa é ativada através de um sensor ultrassônico que alerta para a presença de obstáculos usando duas medições consecutivas. É importante que o tempo de resposta do sistema à esta tarefa seja curto para garantir a segurança das pessoas na fábrica e evitar colisões.

A tarefa **Garra** ocorre quando o veículo chega no local para buscar ou deixar uma peça. É definida como esporádica, porque o intervalo de ativação não é conhecido, mas é possível definir um intervalo mínimo. Quando uma peça fica pronta para montagem, esta tarefa é ativada e só será ativada novamente depois do veículo levar a peça até o destino e retornar para base. O menor intervalo ocorre quando o veículo emprega velocidade máxima e não ocorrem eventos durante o transporte.

Por fim, a tarefa **Seguir rota** é ativada quando o veículo é chamado para buscar uma peça. Esta tarefa é definida como periódica, porque a fábrica possui uma vazão de produção conhecida. Se uma peça é fabricada a cada 12 minutos,

então a cada 12 minutos o veículo será chamado para realizar o transporte.

3.2. Abordagem

Em função da necessidade de um sistema reativa para lidar com as requisições para transporte de peças e com as interrupções causadas pelos sensores de navegação, a abordagem escolhida é a implementação em um microkernel simples por meio de *threads*. Além disso, no laboratório 2, o projeto deve ser implementado em ambiente computacional para simulação por meio do CoppeliaSim por permitir a simulação de sistemas robóticos com suporte para C e Python [2].

4. Resultados

O sistema autômato que representa o mapeamento da fábrica (figura 3) inicia no estado de requisição, que solicita a atuação da garra (figura 8) para pegar o item. Quando ocorrem transições do tipo "logística" (figura 12a), o veículo se encontra parado e seguirá a rota apenas quando terminar o processo de pegar ou entregar o item (figura 9). Nesse sentido, variáveis *booleanas* são adicionadas para evitar que o veículo se movimente e para impedir interrupção de *deadline*, que ocorre quando há alternância entre os estados "inicio_fabrica/destino" e "logística".

Durante o percurso, no mapa da fábrica são executadas instruções síncronas "rota?" que esperam a resposta do autômato (figura 5) para ajustar rota (responsável pelo controle de direção do veículo e identificação de obstruções no caminho). Na tarefa de ajustar rota, utiliza-se o sensor ultrassônico (figura 4) para detecção de obstruções (outros veículos, objetos e trabalhadores).

O sensor ultrassônico aguarda ser solicitado para medir a distância de obstruções à frente do veículo, e, quando ocorre, ele executa a leitura da medição por meio da instrução de tipo *select*, que gera um número aleatório no intervalo definido e com inclusão dos extremos. A partir do valor de leitura do ultrassônico, abstraímos os cálculos aritméticos e a saída expressa a diferença entre medições consecutivas, que representa a velocidade discretizada. Ao final, a leitura é decrementada para representar velocidade negativa, que corresponde a uma obstrução se aproximando do veículo.

Na parte de ajuste da rota, após a requisição, são executadas instruções de buscar a rota na fábrica e ligar motor no seguir rota. Depois, o controle se comunica com o sensor ultrassônico e permanecerá parado se na detecção ocorrer obstrução ou estiver em sinal de emergência. Caso contrário será direcionado para o sensor magnetômetro, que, caso não esteja executando o processo de logística, definirá a direção que o veículo deve percorrer conforme o campo magnético dos ímãs. Esta leitura é representada por meio de três caminhos aleatórios de virar à esquerda, virar à direita ou seguir

em linha reta.

Em relação ao autômato seguir rota (figura 7), este se encontra desligado no início e espera a requisição para ligar o motor. Após ligá-lo, ele seguirá a rota normalmente (figura 11b), enquanto o sensor ultrassônico não detectar obstrução. Caso seja detectada obstrução, o veículo permanecerá parado e iniciará o contador denominado "demora". Caso este contador supere três unidades de tempo, será acionado o autômato de alarme (figura 6 e figura 10b).

Para sair do estado parado e retornar ao seguir de forma normal, será necessário permanecer mais tempo acelerando do que ficou parado no passado (figura 11a). Esse efeito é acumulativo e recursivo para a flexibilidade e a adaptação do sistema nos casos em que ocorre congestionamento de rota. No entanto, obstruções consecutivas (figura 10a) são eventos incomuns, sendo a probabilidade de ocorrer uma obstrução de uma em sete. Assim, é necessário que ocorra o fator multiplicativo de três obstruções consecutivas para que o evento alarme seja disparado.

Por fim, o autômato seguir rota apresenta a instrução de desligar motor e o autômato ajustar rota a instrução de fim de rota, que correspondem ao fim do mapa fábrica (figura 12b). Nesse caso, o veículo retornará ao estado inicial de requisição e aguardará a instrução de solicitação de um novo fluxo logístico.

5. Discussão e conclusões

A implementação do sistema de logística com veículo autônomo apresentou tarefas periódicas, aperiódicas e esporádicas ao longo do percurso, com estimativas dos parâmetros de *deadline* e período. As relações entre as tarefas também foram definidas de acordo com as prioridades estabelecidas, permitindo definir ordenação de tarefas e garantir que o sistema possa lidar com imprevistos e adaptar a rota do veículo de forma eficiente.

Os resultados obtidos através da simulação lógica simplificada no *software* UPPAAL demonstram que o objetivo de criar um sistema de tempo real que atendesse aos requisitos temporais não só foi alcançado, como também foi possível observar a viabilidade de utilizar o *software* de interface para robótica CoppeliaSim em conjunto com FreeRTOS para simulações mais próximas do ambiente industrial real.

Para se obter uma simulação mais realista, devem ser incluídas mais obstruções no caminho, aumentando o caos no sistema dinâmico, para que o veículo autônomo possa lidar com situações mais diversas. Consequentemente, serão necessárias mais variáveis de controle de interrupções, aumentando a adaptabilidade e a flexibilidade do sistema para imprevistos, de modo que o objetivo final ainda seja cumprido dentro dos requisitos temporais especificados.

Referências

- [1] Carros Cascavel. Mega fábricas porsche natgeo dublado port. Disponível em: <https://www.youtube.com/watch?v=J38S7zZuOkA>. Online; Acesso em 18 de jan. de 2023.
- [2] CoppeliaSim. Robot simulator coppeliasim: create, compose, simulate, any robot. Disponível em: <https://www.coppeliarobotics.com/>. Online; Acesso em 19 de jan. de 2023.
- [3] J. Crestani. Logística: o que é, por que é importante e quais os principais tipos. Disponível em: <https://www.mutuus.net/blog/logistica-o-que-e/>, 2021. Online; Acesso em 18 de jan. de 2023.
- [4] R. S. de Oliveira. *Fundamentos dos Sistemas de Tempo Real*. Edição do Autor, 1ª edição, 2018.
- [5] UPPAAL. Home. Disponível em: <https://uppaal.org>, 2019. Online; Acesso em 16 de jan. de 2023.
- [6] WELT Documentary. Building a porsche 911 - legend on 4 wheels — full documentary. Disponível em: <https://www.youtube.com/watch?v=mOB13tcGILA>. Online; Acesso em 18 de jan. de 2023.

Anexos

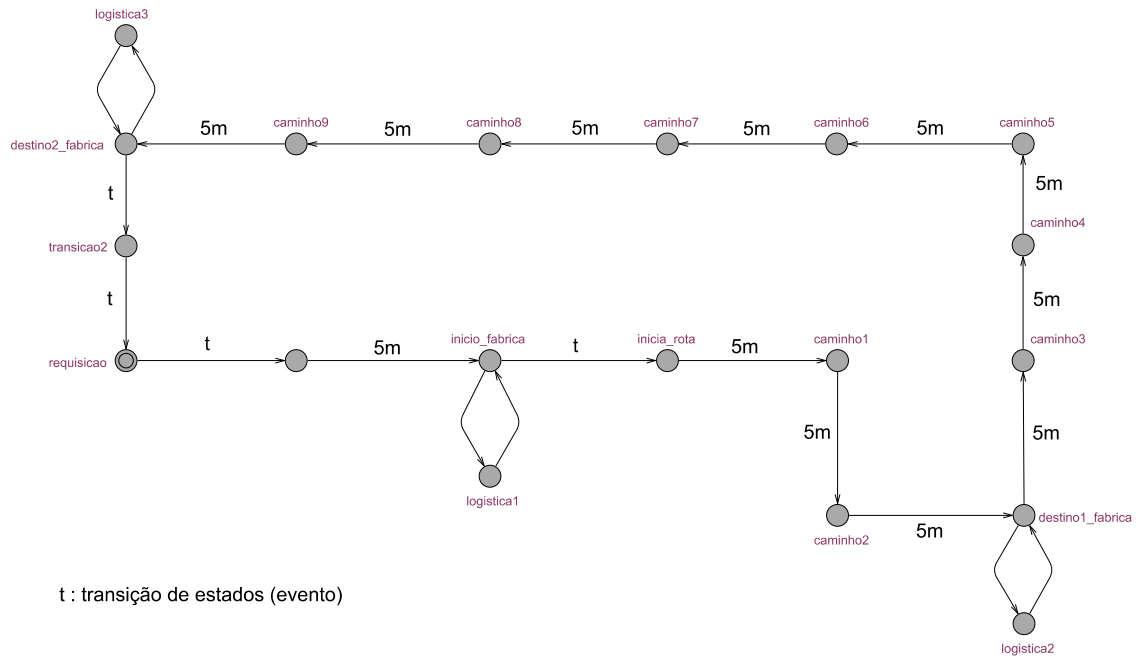


Figura 2: Mapa da fábrica com a disposição dos ímãs

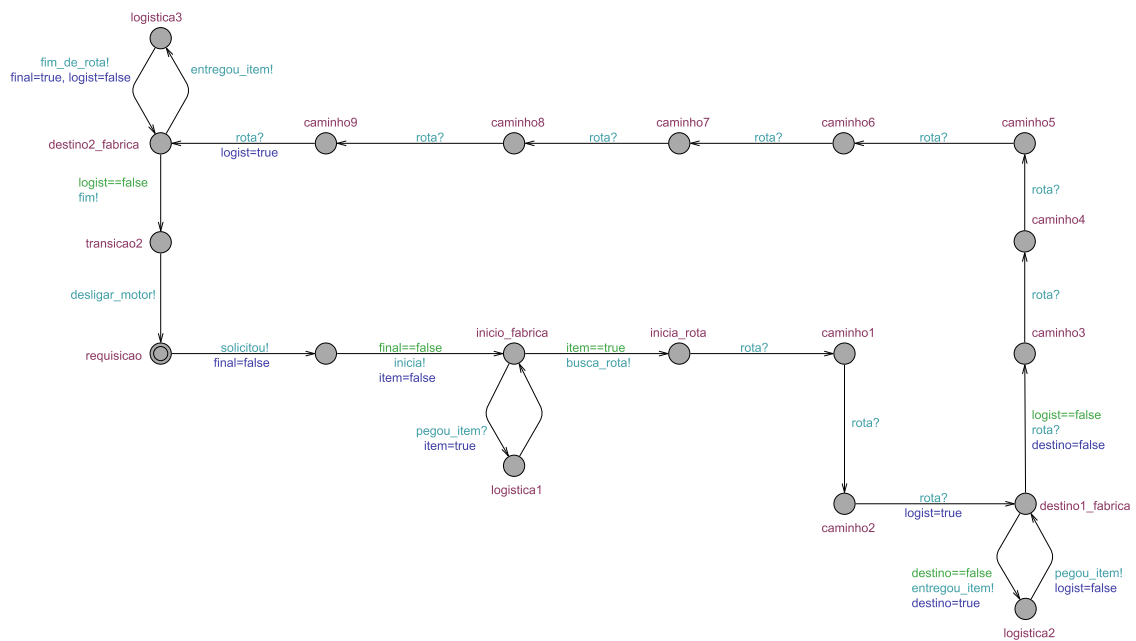


Figura 3: Autômato do mapa da fábrica

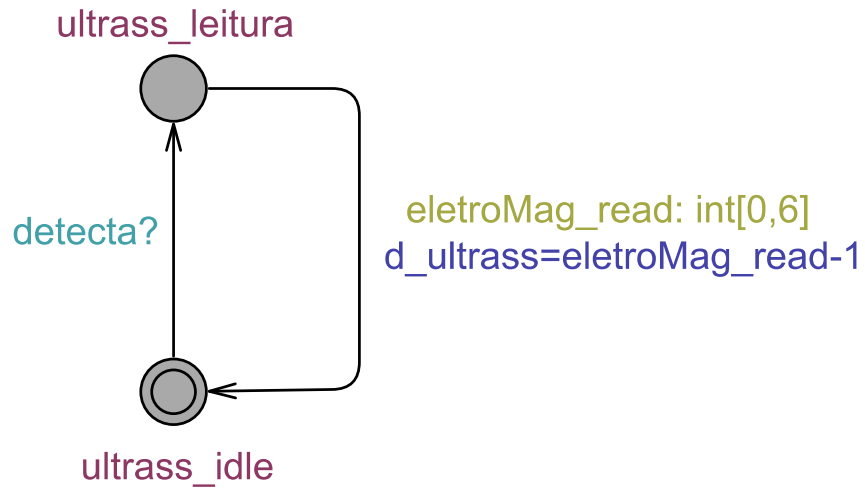


Figura 4: Autômato para detectar obstrução

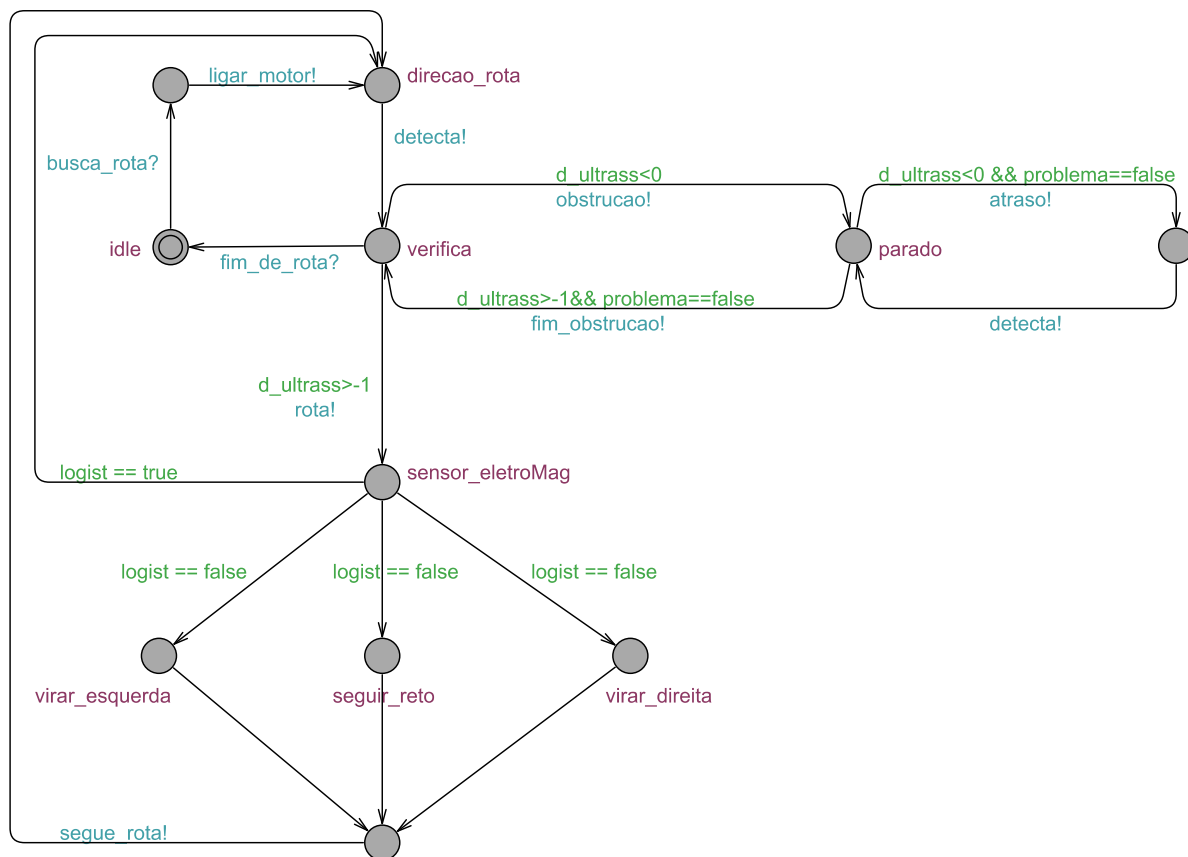


Figura 5: Autômato para ajustar rota

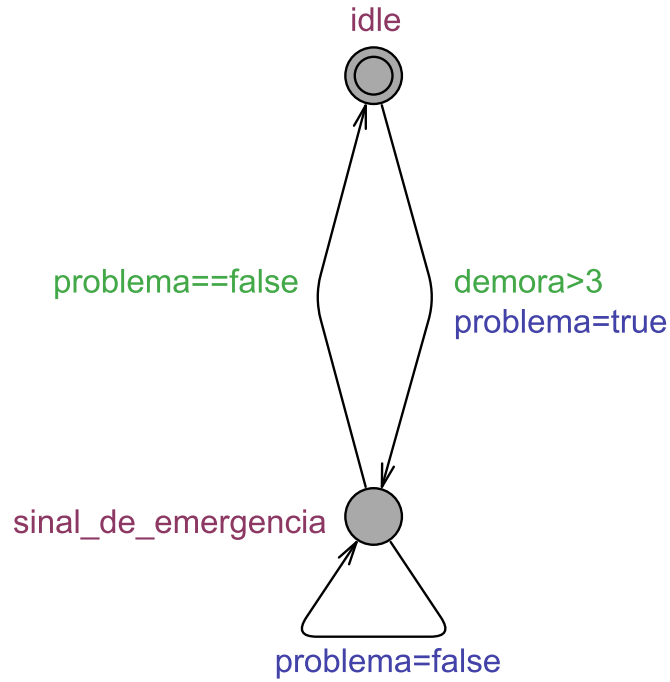


Figura 6: Autômato do sinal de alarme

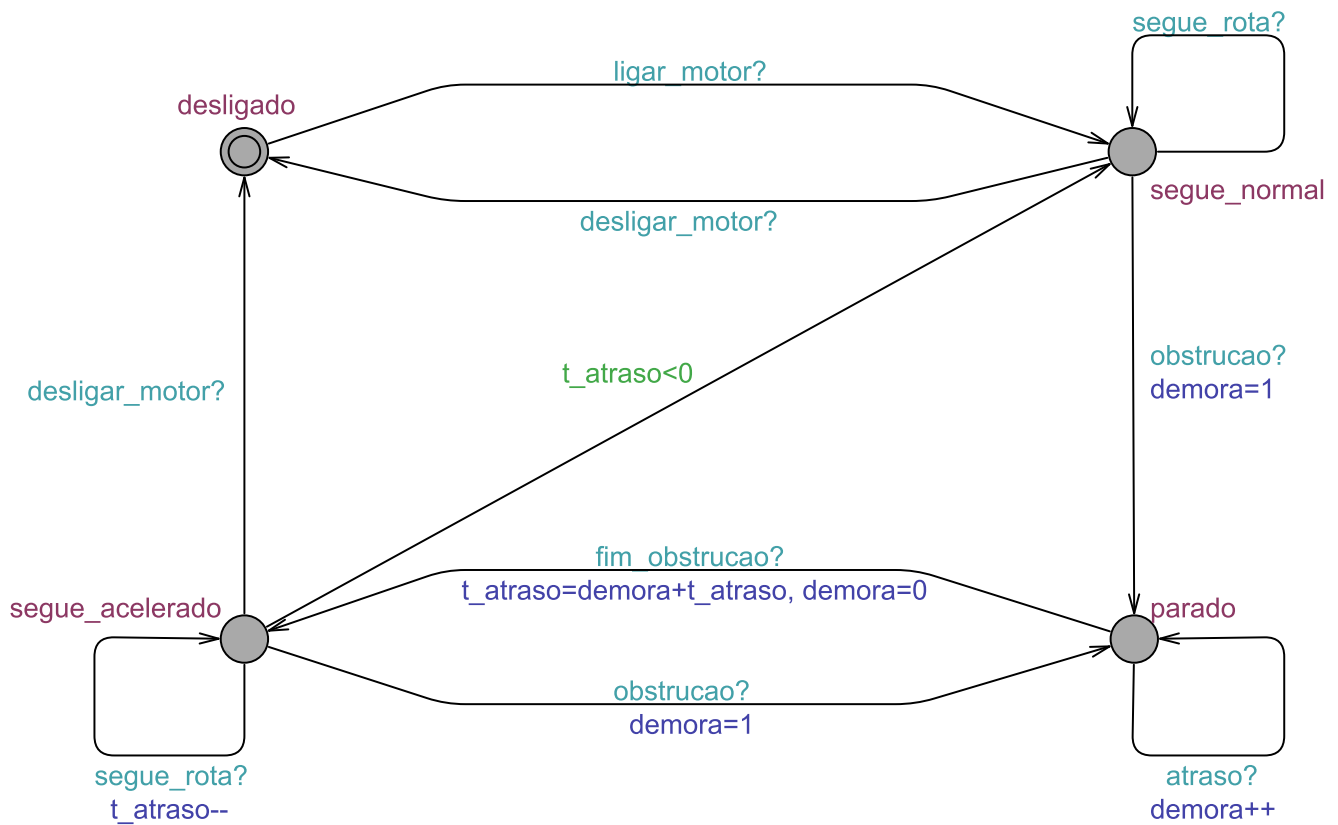


Figura 7: Autômato para seguir rota

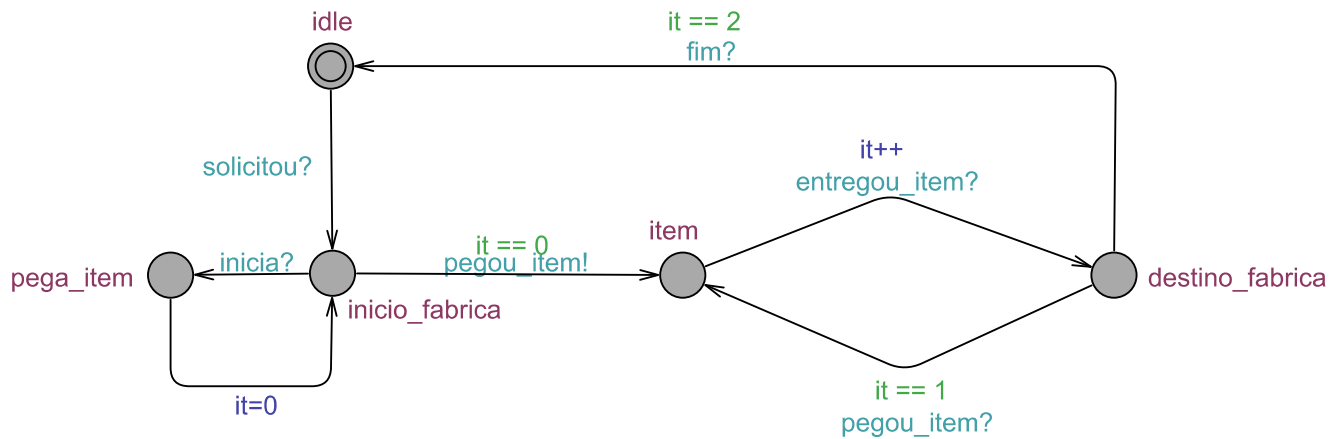


Figura 8: Autômato do atuador garra

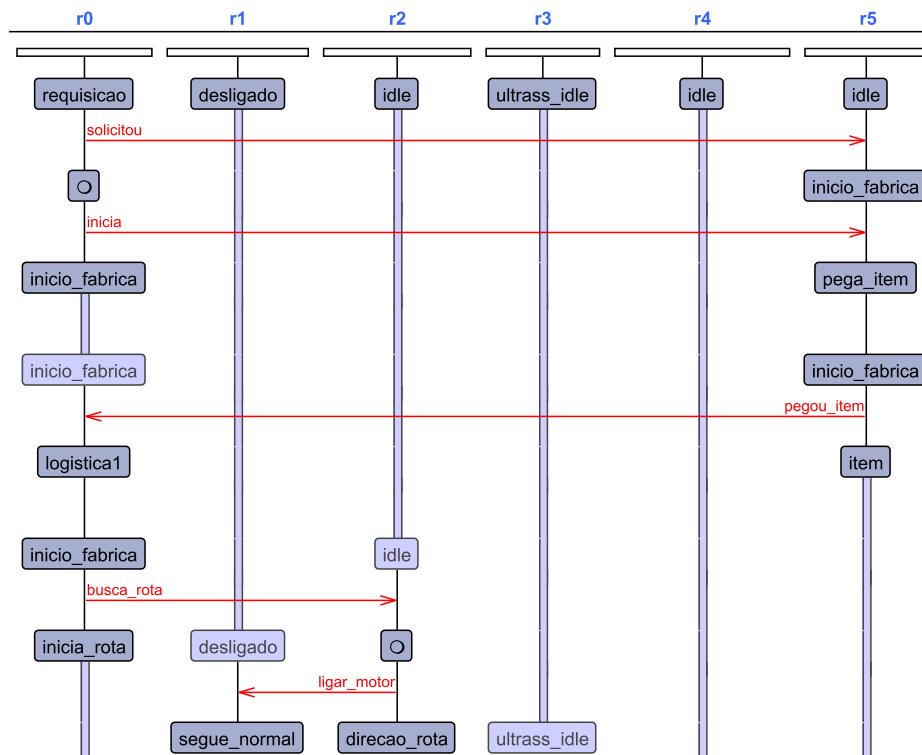
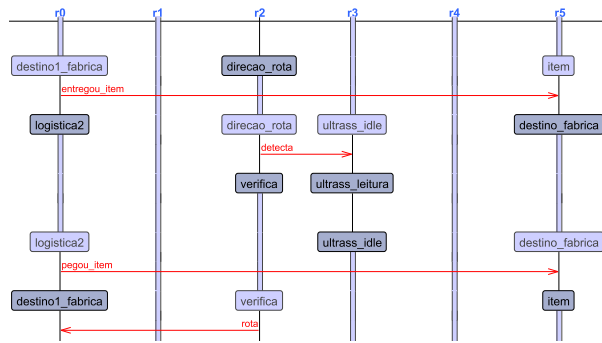
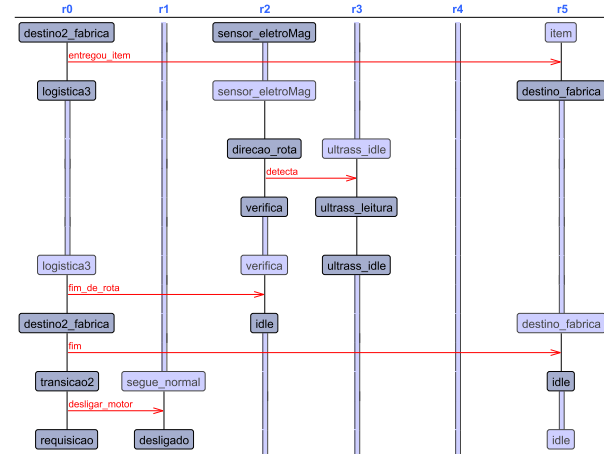


Figura 9: Comportamento no início do sistema



(a) Interações intermediárias



(b) Fim do percurso

Figura 12: Comportamento do sistema