

Sistema de Controle de Empacotamento de Latas

Cristóvão Bartholo Gomes
140135081

cristovao@live.com

Matheus Escovedo da Costa
140155317

matsrei@gmail.com

Allan Jhonny
140128468

allanjhonny55@gmail.com

Abstract

This paper presents a simulation of a Real Time System of a quality control and packing of a fictional product. Here we develop and test the model using computer softwares and Real Time theory modelling.

1. Objetivo

Desenvolver uma aplicação que use os pré-requisitos de sistemas de tempo real, aplicando os conceitos a eles atribuídos como deadlines, períodos e tempo de computação de cada tarefa.

2. Introdução

Sistemas computacionais com requisitos de Tempo Real são aqueles que são submetidos a requisitos de natureza temporais não triviais, ou seja, são definidos pelo ambiente da própria aplicação. Quanto aos resultados de um sistema em tempo real eles devem estar corretos tanto na lógica quanto temporalmente e é necessária a análise desses sistemas para ter a segurança de que todos os requisitos temporais sejam cumpridos, uma vez que alguns desses sistemas, em casos de falhas temporais, podem ter consequências catastróficas, estes são chamados de tempo real crítico. Em outros casos, o requisito temporal descreve apenas o comportamento a ser executado, sendo chamados de tempo real não-crítico. [2]

Quando se fala em tarefas de tempo real algumas definições são de extrema importância, que são as definições e tipos de tarefas, deadlines, tempos de execução e períodos de execução.

Tarefas são segmentos de código cuja execução possui um atributo temporal próprio, que são o tempo de computação, períodos de execução e os deadlines. As tarefas são classificadas em periódicas, aperiódicas e esporádicas. Uma tarefa periódica é aquela que é ativada a cada P unidades de tempo, o instante de chegada pode ser calculado a partir do inicial. Uma tarefa esporádica é aquela em que os instantes de chegadas não são conheci-

dos, porém, é conhecido um intervalo mínimo entre suas execuções. Por fim, uma tarefa aperiódica é aquela em que nada se sabe quanto as informações temporais de ativação da tarefa.

Deadline é o instante máximo desejado para a conclusão de uma tarefa, são definidos em relativo ou absoluto, relativo é quando o deadline é determinado em relação ao tempo de início da tarefa, já o absoluto o deadline é mensurado em relação ao tempo universal de início da execução das tarefas.

O tempo de execução é o tempo que o processador gasta com uma determinada tarefa, este tempo não inclui as interferências causadas por outras tarefas, ou seja, é apenas o tempo gasto pelo processador para executar todas as linhas de código da tarefa, infelizmente, o tempo de execução é variável, essa variância é causada devido a “interferências” causadas pelo software e pelo hardware do sistema, assim, é importante em sistemas de tempo real abordar o tempo de execução de uma forma pessimista, ou seja, considerar que dentro as variâncias do tempo de execução sempre ocorrerão o pior caso de tempo de execução (WCET), dessa forma, se todos os requisitos temporais forem garantidos em tais circunstâncias é seguro afirmar que o sistema sempre respeitará os requisitos temporais e funcionará corretamente. [2].

2.1. Laço principal com tratador de interrupção

Dentre os diversos sistemas de tempo real o laço principal está entre uma das estratégias mais simples, portanto apresenta vantagens de ser fácil de implementar e custo computacional baixo. Esse sistema consiste em concatenar as tarefas organizadas da maneira desejada de maneira que os requisitos temporais sejam cumpridos. Entretanto essa estratégia apresenta algumas desvantagens. Dentre elas está o fato de que essa estratégia não lida bem com tarefas esporádicas, ou seja, tarefas que ocorrem raramente. Como solução para esta situação desenvolveu-se a ideia do laço principal com tratador de interrupções, essa estratégia consiste em reservar um espaço a mais no ciclo principal para que quando haja interrupção as tarefas não violem seus requisitos temporais.

2.2. Troca de mensagens direta

Dentre as diversas formas de realizar comunicação está a troca de mensagem direta, ela é uma estratégia simples ideal para sistemas simples que consiste em simplesmente o envio de uma mensagem, por exemplo, um sensor enviar periodicamente o seu valor lido para a tarefa responsável pelo controle do sistema.

3. Materiais e Métodos

3.1. Metodologia

3.1.1 Laboratório 1:

Para o laboratório 1 a metodologia empregada foi, após a definição do tema via reuniões em grupo, delimitar requisitos temporais que contemplasse o escopo da disciplina e, após isso, modelar e simular o sistema com requisitos lógicos e temporais via plataforma UPPAAL.

3.1.2 Laboratório 2:

Para o laboratório 2 foi ponderado todas as argumentações feitas pelo professor, assim corrigindo erros cometidos e tentando melhorar a ideia do projeto como um todo, após as correções, foi feito a estimativa dos valores de WCET das tarefas e, por fim, a simulação do sistema completo via plataforma Unity.

3.2. Materias e Equipamentos

Para a elaboração da simulação 1 do projeto proposto foi utilizado o software UPPAAL, do qual, é referência em simulação de aplicações de tempo real, fornecendo o estado lógico da aplicação e fluxograma temporal. Para simulação final do projeto foi utilizado o software Unity, do qual, permite a modelagem física do sistema para simulação de lógica e requisitos temporais. [1]

4. Resultados

4.1. Arranjo Experimental

4.1.1 Resumo do projeto

A ideia do projeto é implementar o serviço de empacotamento de latas por meio do sistema de LAÇO PRINCIPAL COM TRATADORES DE INTERRUPÇÕES, onde tarefas periódicas estão no laço principal e as tarefas aperiódicas são tratadas como interrupções via software. O processo do sistema passa por 2 etapas, a primeira etapa é responsável por realizar o controle de qualidade do produto, por meio de visão computacional será possível reconhecer qualquer avaria na estrutura física da lata, a segunda etapa é separar as latas de qualidade aceitável das latas danificadas, as primeiras vão ser direcionadas para o empacotamento, já as latas com problemas serão direcionadas para o descarte.

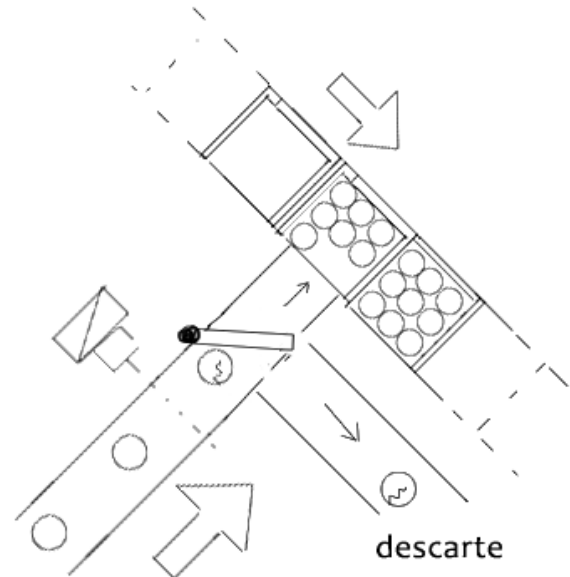


Figure 1. Representação do Sistema

Esse sistema utiliza duas esteiras automatizadas, a primeira esteira carrega as latas que serão selecionadas, o período de chegada entre as latas é de 1,655s. A segunda esteira carrega as caixas onde as latas estão sendo empacotadas, cada caixa suporta um total de 9 latas. A representação do sistema pode ser vista na figura 1.

4.1.2 Tarefas do Sistema

Tarefa 1: A tarefa tem a responsabilidade de fazer a identificação da lata, via câmera, cujo objetivo é avaliar se a lata não possui avarias em sua estrutura física, desse modo, estando apta para prosseguir o seu processo de empacotamento. É responsável por atualizar a variável global do número total de latas empacotadas.

Tarefa 2 e 3: Estas tarefas são responsáveis enviar o sinal de aprovação ou falha no teste que a lata é submetido, atualizando a informação para as tarefas 3 e 4.

Tarefa 4 e 5: Estas tarefas recebem o sinal das tarefas 2 e 3 que informam se a lata é apta ou não para o empacotamento, após isso, processa esse resultado e emite, caso necessário, um sinal para o atuador da porta seletora, após isso, quando a tarefa 5 é executada ela sempre verifica o número total de latas que em caso de 9 envia um sinal para a tarefa de número 6.

Tarefa 6 e 7: Estas tarefas são responsáveis por ativar e parar a esteira 2, respectivamente. A tarefa 6 recebe o sinal da tarefa 5, assim, iniciando o movimento da esteira 2. Após finalizar, a tarefa 6 chama a tarefa 7 que realiza a paragem da esteira e reinicia o contador de latas.

4.1.3 Relacionamento entre tarefas

-A tarefa 1 precisa ser executada primeiramente, pois ela quem vai determinar o resultado das tarefas seguintes 2 e 3, portanto a tarefa 1 não exige precedência. -As tarefas 2 e 3 precisam ser executadas após a leitura da câmara, portanto têm precedência dessa tarefa.

-As tarefas 4 e 5 necessitam dos processamentos advindos das tarefas 2 e 3, portanto estas as precedem. -A tarefa 6 é precedida pela tarefa 5, que determina se a interrupção ocorre ou não baseado no número de latas. -A tarefa 7 é precedida pela tarefa 6.

4.1.4 Estimativas de WCET

Para realizar as estimativas dos WCET's de todas as tarefas definidas foi utilizado o site [3], do qual, a partir de um código em linguagem C de entrada, fornece o assembly do código para alguns tipos de compiladores, para o projeto foi utilizado o do Arduino Uno(1.8.9) para garantir que a simulação baseia-se em um sistema factível. O arduino possui um processador Uniciclo e faz 20 MIPS, assim, foi colocado o código C de cada uma das tarefas e gerado o seu assembly, após isso, foi considerado o pior caminho de execução, por serem tarefas simples é possível utilizar este tipo de abordagem, foi contabilizado o total de tempo de cada tarefa com base nos modelos ministrados em outras disciplinas como OAC e, por fim, adicionado uma margem de segurança de 20% prática muito utilizada na indústria.

4.1.5 Características das tarefas

Tarefas	Numeração	Tipo
Leitura	1	Periódica
Passou_Testes	2	Periódica
Falhou_Testes	3	Periódica
Abre_Porta	4	Periódica
Fecha_Porta	5	Periódica
Anda_Esteira	6	Aperiódica
Para_esteira	7	Aperiódica

Requisitos da Tarefa 1:

Período = 1,655 s;

Deadline = 1,655 s;

Tempo de execução = 755 ms;

Requisitos das Tarefas de 2 a 5:

Período = 1,655 s;

Deadline = 1,655 s;

Tempo de execução = 3,45 μ s;

Requisito das Tarefas 6 e 7:

Intervalo mínimo = 14,895 s;

Deadline = 122 ms;

Tempo de execução = 3,45 μ s;

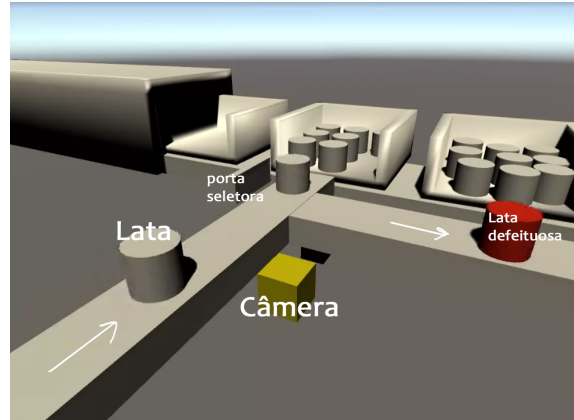


Figure 2. Simulação no Unity.

4.2. Implementação

Foi implementado a partir do modelo teorizado o modelo no programa computacional UPPAAL, dessa forma pôde-se fazer as análises apropriadas a partir de simulações feitas pelo programa. Foi modelada a tarefa da câmera que atua sobre as latas da forma como é mostrado na figura 3, ela atua decidindo quais latas são defeituosas ou não e sinaliza o controlador do servo-motor, que emite comandos para o seu atuador (fig.6), direcionando as latas para o empacotamento ou para o descarte. Eventualmente, nove latas sem defeito terão passado para as caixas, esse valor é armazenado pela variável "latas". Quando esse valor é atingido ativa-se a esteira 2 para que entre uma nova caixa vazia e repete-se o ciclo. A partir da análise realizada pôde-se concluir através de simulação que não ocorrem deadlocks para os requisitos temporais estabelecidos. O fluxograma obtido pode ser visto na figura 7, e o caso em que ocorre a troca de caixa pode ser observado na figura 8.

Além disso, foi implementada uma simulação do modelo físico no Unity para que dessa forma pudéssemos verificar a validade do sistema, já que não foi possível realizar a implementação prática devido à pandemia. O Unity é uma plataforma para desenvolvimento de jogos e é uma ferramenta bastante intuitiva para criação e simulação de ambientes físicos de corpo rígido. A visualização da simulação pode ser vista na figura 2.

Para esta simulação, os valores de tempo de execução foram estabelecidos utilizando um valor fixo de tempo ao qual foi adicionado um valor aleatório com alcances definidos para simular a variância do tempo de resposta sistema.

A distribuição utilizada foi a uniforme, pois mesmo que seja mais provável que a distribuição real do sistema seja aproximadamente uma gaussiana, para o propósito desse trabalho a distribuição utilizada foi suficiente para simular o comportamento do sistema.

Para os valores estabelecidos, teve-se um sistema sem falhas para uma simulação de 30 minutos, o que mostra, para essa análise, que a simulação validou o sistema em questão nesse experimento.

5. Discussão e Conclusão

Tarefas aperiódicas elevam bastante o grau de dificuldade da implementação/simulação de um sistema, pois mesmo quando definido o seu intervalo mínimo a lógica torna-se mais complexa para garantir o funcionamento do sistema, além disso, a definição exata do que é cada tarefa é fundamental para um bom sistema de requisito temporal, tarefas visualizadas de maneira macro pode camuflar complexidades internas gerando valores incorretos de WCET e tarefas pequenas demais podem tornar o sistema muito trabalhoso e repetitivo em sua elaboração.

Por meio da análise do problema pôde-se observar a importância de uma análise mais minuciosa do sistema de tempo real a ser implementado. Em teoria é sempre fornecido os valores de execução em pior caso, período e deadline de cada tarefa, porém, já na simulação de uma nova aplicação é perceptível que estes valores não são facilmente obtidos na vida real, sendo necessário estipular valores que nem sempre correspondem com os valores reais.

Através das simulações realizadas no UPPAAL pôde-se concluir que a modelagem desenvolvida apresentou resultados coerentes com o esperado na teoria, assim, evidenciando a qualidade do software em garantir requisitos lógicos e temporais de sistemas.

Por meio da simulação no Unity foi possível validar a solução proposta para o problema, já que em uma simulação em que ocorreram a passagem de mais de 1000 latas não ocorreu nenhuma falha. Entretanto, para diferentes valores de frequência de chegada das latas ou da velocidade da porta seletora o sistema simulado perde os deadlines e ocorre a falha do sistema, que é exatamente o esperado, sendo que por vezes ocorre a queda de latas e por outras latas são selecionadas de maneira errada.

References

- [1] M. Checking. https://www.youtube.com/watch?v=tUSxi_rSXwo
- [2] R. S. de Oliveira. *Fundamentos dos Sistemas de Tempo Real*. Amazon Digital Services LLC, 2018.
- [3] C. Explorer. <https://godbolt.org/>

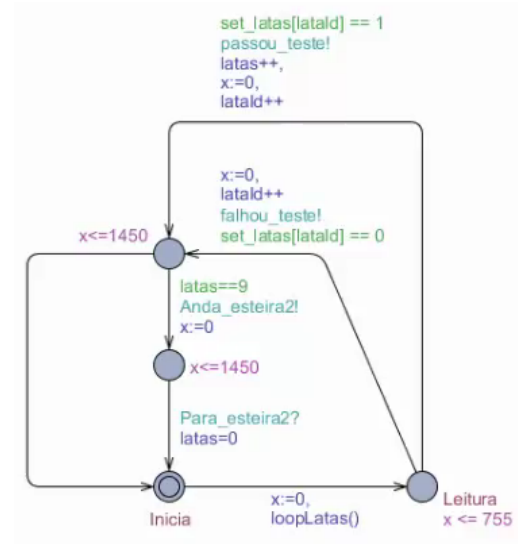


Figure 3. Camera

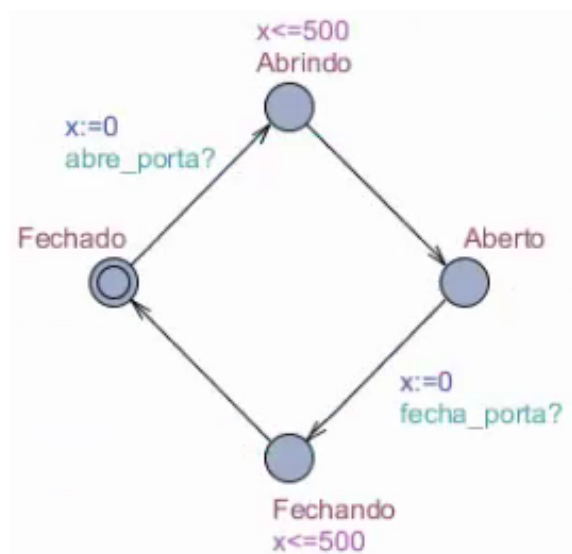


Figure 4. Atuador do servo-motor

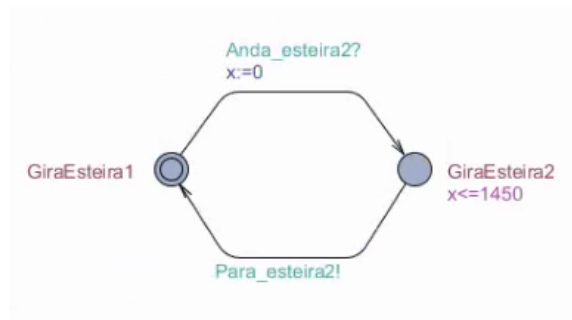


Figure 5. Controlador das Esteiras

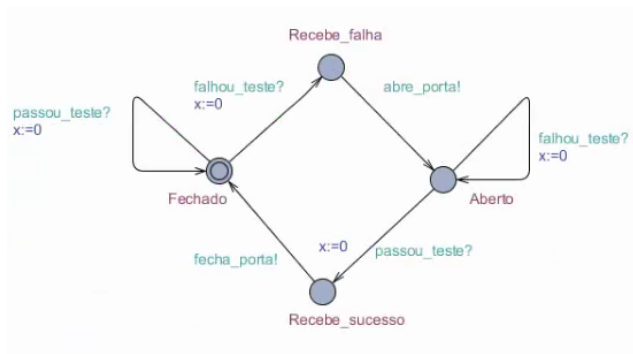


Figure 6. Controlador do Servo Motor

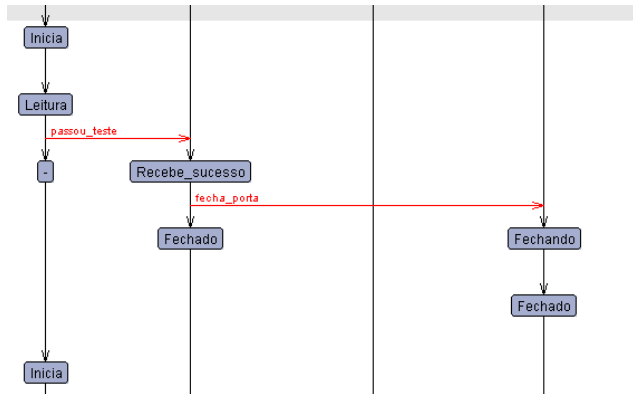


Figure 7. Grafo resultante da chegada de uma lata boa porém com a portinhola fechada.

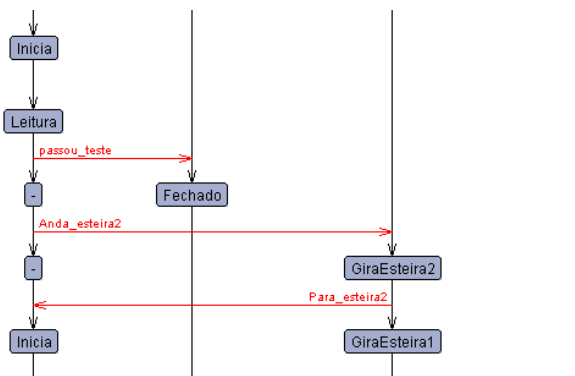


Figure 8. Grafo resultante da movimentação da segunda esteira.