
ANÁLISE DE MECANISMOS DE ESCALONAMENTO PARA SISTEMAS DE TEMPO REAL TOLERANTE A FALHAS

Flávia Maristela Santos Nascimento – GSORT – IFBA *Campus* Salvador. E-mail: flaviamsn@ifba.edu.br
Franklin Lima Santos – GSORT – IFBA *Campus* Salvador. E-mail: franklin_lima@i3e.org

RESUMO: Os sistemas de tempo real têm sido amplamente utilizados na área de controle, subárea da mecatrônica caracterizada por possuir sistemas críticos, cujo funcionamento incorreto pode ocasionar catástrofes sociais, econômicas e ambientais. Ou seja, neste ambiente as falhas precisam ser controladas a ponto de não interferir (ou interferir minimamente) no funcionamento da aplicação. O estudo proposto neste trabalho visa alcançar, como um de seus resultados, uma aproximação entre os modelos teóricos e práticos, tornando mais realistas as considerações teóricas. Isso possibilitará estudos mais detalhados e uso prévio de simulações, antes que as aplicações sejam colocadas em produção. Assim, este trabalho possui impactos sociais, econômicos e ambientais na medida em que poderá minimizar os riscos de ocorrência de catástrofes.

PALAVRAS-CHAVE: Sistemas de Tempo-Real, Escalonamento, Tolerância a Falhas.

1. INTRODUÇÃO

Os sistemas mecatrônicos, como subárea da automação de sistemas, vêm ganhando destaque devido à sua principal característica, que é a capacidade de integrar componentes inteligentes e sistemas. O advento dos sistemas mecatrônicos impulsionou o uso de computadores como hospedeiros de sistemas de controle. Neste ponto, as aplicações computacionais precisaram considerar tanto os requisitos lógicos, quanto os requisitos temporais dos objetos controlados, sendo então caracterizadas como sistemas de tempo real. Para tais sistemas, a violação das restrições temporais pode incorrer em riscos ambientais ou em perda de vidas humanas.

Partindo da premissa de que todas as aplicações potencialmente falham, é preciso avaliar de que forma estas falhas podem implicar no não cumprimento dos requisitos temporais, principalmente no que diz respeito às aplicações críticas, a fim de que as ações de recuperação possam ser executadas no sentido de garantir a correção lógica e temporal de tais aplicações, mesmo na presença de falhas. As diversas soluções que consideram os aspectos de falha em sistemas de tempo real disponíveis na literatura assumem suas próprias hipóteses de falhas e restringem o modelo do sistema de maneira particular. Não há verificações de quão rea-

listas são tais hipóteses ou restrições frente à necessidade real das aplicações, ou uma efetiva comparação das diferentes abordagens de escalonamento com suporte a tolerância a falhas.

O objetivo deste trabalho é preencher esta lacuna, fornecendo uma análise comparativa entre as diferentes abordagens de escalonamento para sistemas de tempo real, ressaltando o impacto da implementação de técnicas de tolerância a falhas em tais sistemas (SMITH, 2004). Além dos resultados provenientes de uma análise qualitativa, pretende-se ainda utilizar simulações para avaliar as implementações em sistemas tolerantes a falhas, considerando as abordagens estudadas.

2. MOTIVAÇÃO

Os sistemas mecatrônicos, como subárea da automação de sistemas, vêm ganhando destaque devido a sua principal característica: capacidade de integrar componentes inteligentes e sistemas (via hardware ou software), de forma a reduzir o custo relacionado ao desenvolvimento de produtos [24, 19]. Os principais componentes de um Sistema Mecatrônico podem ser descritos como na Figura 1.

Figura 01. Esquema Geral de um Sistema Mecatrônico

O usuário é a entidade que deseja manipular um objeto controlado, que pode ser uma planta industrial, por exemplo. O objeto controlado é uma entidade do mundo real que possui requisitos lógicos, relacionados à sua correteza funcional, e requisitos temporais, relativos a prazos máximos de execução para suas operações. A manipulação deste objeto controlado é feita através de uma Interface Homem-Máquina (IHM), que tem como objetivo monitorar o funcionamento e comandar este objeto. Isto é feito através do sistema de controle, responsável por gerenciar todo o fluxo de funcionamento do objeto controlado (MiYAGI, 2004).

O sistema de controle gerencia o objeto controlado através de sensores e atuadores. Os sensores são dispositivos responsáveis por informar ao sistema de controle o estado atual do objeto controlado. Já os atuadores representam os mecanismos através do qual o sistema de controle interfere na evolução do objeto controlado (MiYAGI, 2004).

O advento dos Sistemas Mecatrônicos impulsionou o uso de computadores como hospedeiros de sistemas de controle. Isto fez com que tais sistemas evoluíssem no sentido de contemplar os requisitos lógicos e temporais dos objetos controlados. Sistemas computacionais que possuem estes dois tipos de requisitos são conhecidos como sistemas de tempo real (STANKOVIC, 1998).

Nem todas as aplicações de tempo real requerem o mesmo nível de respeito aos requisitos temporais. Considerando o conjunto dos sistemas de tempo real, é possível identificar sistemas "críticos" e "não-críticos".

Os sistemas de tempo real críticos são aqueles cujos requisitos temporais precisam ser cumpridos, de forma a evitar possíveis consequências catastróficas (FARINES, 2000), (STANKOVIV, 1998). É o caso, por exemplo,

do controle de uma planta nuclear. Para os sistemas de tempo real não-críticos, o não cumprimento dos requisitos temporais não incorre em riscos ambientais ou em perda de vidas humanas. É o que acontece com as aplicações multimídia.

É interessante observar que existe uma diferença importante entre os sistemas computacionais de propósito geral e os sistemas de tempo real. Para os primeiros, o principal objetivo da computação gira em torno de conseguir minimizar o tempo médio de resposta das tarefas e o foco está voltado para a garantia de resultados corretos (STANKOVIC, 1998). O "tempo" é apenas um fator indicativo de limites para a execução de operações. Para os sistemas de tempo real, considerar apenas os requisitos lógicos não é suficiente para garantir sua correção.

Os sistemas de tempo real são geralmente definidos em função de um conjunto de tarefas, que representam unidades de execução. Usualmente, cada tarefa possui seu próprio requisito temporal. Por exemplo, a tarefa que executa o algoritmo de controle deve prover o resultado num prazo máximo pré-estabelecido. Desta forma, é necessário estabelecer uma ordem de execução das tarefas (ou de suas operações) considerando os requisitos temporais das mesmas. Gerar tal sequenciamento (escala de execução) é papel do mecanismo de escalonamento, peça-chave na garantia da correção temporal das tarefas.

A escala de execução é gerada em função de heurísticas que definem que tarefa executar em função do tempo. Tais heurísticas são chamadas de políticas de escalonamento. Uma vez que a política de escalonamento tenha sido escolhida, é necessário determinar se existe a possibilidade de que alguma tarefa viole seus requisitos temporais.

Assim, a ordem de execução das tarefas, gerada pelo escalonador, precisa ser validada a fim de garantir correção temporal. Esta atividade é responsabilidade da análise de escalonamento, que irá avaliar o comportamento temporal do sistema considerando sua política de escalonamento, verificando se todas as tarefas irão cumprir suas restrições temporais. Se isso for verdade, o sistema é dito escalonável, de acordo com a política de escalonamento considerada.

No contexto das aplicações de tempo real, é preciso ainda considerar as falhas. Partindo da premissa de que todas as aplicações potencialmente falham (WBWER, 2010), é preciso avaliar de que forma estas falhas podem implicar no não cumprimento dos requisitos temporais, principalmente no que diz respeito às aplicações críticas. Por exemplo, considere a tarefa de controle mencionada anteriormente e suponha que ocorra um erro em sua execução causado por uma falha transiente. Neste caso, reexecutar tal tarefa pode ser satisfatório.

No entanto, o intervalo de tempo que se dispõe para a reexecução é menor do que em cenários onde não há falhas. Desta forma, não apenas o escalonador deve contemplar tais possibilidades, mas as técnicas de análise também devem ser adaptadas para considerar o efeito da falha.

Na literatura, existem várias soluções que consideram os aspectos de falha em sistemas de tempo real, inclusive avaliando tais aspectos sob a ótica do escalonamento de tarefas e da análise de escalonamento. Entretanto, cada abordagem assume suas próprias hipóteses de falhas e restringem o modelo do sistema de maneira particular.

O objetivo deste trabalho é fornecer uma análise comparativa entre as diferentes abordagens de escalonamento para sistemas de tempo real, ressaltando o impacto da implementação de técnicas de tolerância a falhas em tais sistemas. Além dos resultados provenientes de uma análise qualitativa, pretende-se ainda utilizar simulações para avaliar as implementações em sistemas tolerantes a falhas, considerando as abordagens estudadas.

3. METODOLOGIA

A evolução dos sistemas computacionais fomentou a criação de diversas áre-

as para o estudo de modelagem, projeto e implementação destes sistemas. Por outro lado, o avanço da tecnologia nos processos de fabricação e miniaturização dos componentes eletrônicos permitiu que os computadores passassem a hospedar sistemas de controle, que estão a todo tempo susceptíveis a falhas, podendo ocasionar desastres ambientais ou outros tipos de perdas (sociais e econômicas).

Alguns destes sistemas possuem, além das restrições do ponto de vista lógico e aritmético dos dados (*correctness*), também restrições do ponto de vista temporal (*timeliness*) e por isso são conhecidos como Sistemas de Tempo Real (STR). Com o objetivo de permitir que o sistema minimize os efeitos das falhas que o acometem, surgiram diversas abordagens de prevenção e tolerância a falhas.

Para a realização deste projeto, foi necessária uma ampla revisão bibliográfica de forma a fundamentar o estudo, familiarizar-se com a taxonomia utilizada na literatura e classificar as abordagens existentes.

A revisão foi feita através da leitura de artigos de periódicos (IEEE e ACM), livros e materiais disponíveis em acervos e bibliotecas virtuais. Os itens abaixo foram realizados durante o projeto e serão descritos sucintamente posteriormente.

- Conceituação e Caracterização dos Sistemas de Tempo Real em ambientes monoprocesados e multiprocessados;
- Revisão sobre aspectos gerais de tolerância a falhas;
- Desenvolvimento da ferramenta de simulação¹;
- Simulação da resiliência de um dado sistema utilizando:

- RM (*Rate Monotonic*) (LEHO-CZKY, 1989)

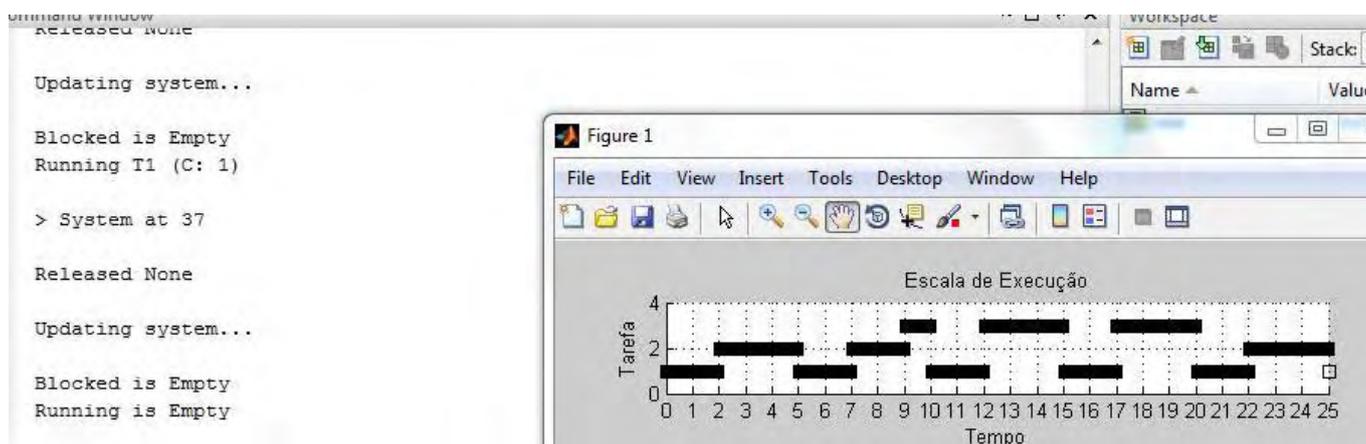
- EDF (*Earliest Deadline First*) (BUTAZZO, 2005)(BURNS, 2009).

¹ A ausência de ferramentas disponíveis fomentou a necessidade do desenvolvimento de uma aplicação capaz de realizar as simulações desejadas.

4. CENÁRIOS DE SIMULAÇÃO E RESULTADOS

A política de escalonamento EDF foi adicionada à ferramenta já desenvolvida. A Figura 2 mostra a escala de execução gerada a partir do simulador desenvolvido.

Figura 2. Visão Geral da Escala de execução para o conjunto de tarefas $\lambda 1$ através do simulador desenvolvido.



É possível também gerar a escala de execução em tempo real, como mostra a Figura 3.

Figura 3. Escala de execução sendo gerada em tempo real através do simulador desenvolvido.



4.1 INTEGRAÇÃO DA FERRAMENTA

Na última versão da ferramenta, é possível carregar um conjunto de tarefas e comparar a escalonabilidade e utilização do mesmo para o RM e o EDF. O resultado pode ser visto na Figura 4.

Figura 4. Visão do Sistema para Análise de Resiliência em Tempo-Real

```
- Simulador para Análise de Resiliência em Sistemas de Tempo Real -
  Siga as instruções atentamente!
```

```
Escolha uma opção:
```

```
[1] Gerar conjunto de tarefas
[2] Analisar conjunto gerado previamente
[3] Gerar escala de execução (RM)
[4] Gerar escala de execução (EDF)
[5] Gerar falhas no sistema
[6] Analisar resiliência do sistema (RM)
[7] Analisar resiliência do sistema (EDF)
[8] Visualizar resultados (gerar gráfico)
[9] Sair do simulador
```

```
§ 2
```

```
Analisando o conjunto gerado...
```

```
> Número de tarefas carregadas: 3
> Teste de Liu & Land: 0.78
> Utilização total do conjunto: 0.85
> Escalonabilidade (RM): Não Escalonável
> Escalonabilidade (EDF): Escalonável
> Hiperperíodo: 40
```

Exemplo 1. Seja o conjunto de tarefas periódicas e independentes $\lambda_1 = \{\tau_1, \tau_2, \tau_3\}$ com os seguintes atributos $C = (2, 5, 8)$ e $P = (5, 20, 40)$. Com isso, podemos deduzir:

- O sistema é escalonável pelo EDF, pois $U = 2/5 + 5/20 + 8/40 = 0,85 \leq 1$.
- O sistema não é escalonável pelo RM, pois $U = 2/5 + 5/20 + 8/40 > 0,78$.

5. CONSIDERAÇÕES FINAIS

Os sistemas de tempo real têm sido amplamente utilizados na área de controle, subárea da mecatrônica, caracterizada por possuir sistemas críticos, cujo funcionamento incorreto pode ocasionar catástrofes sociais, econômicas e ambientais. Ou seja, neste

ambiente as falhas precisam ser controladas a ponto de não interferir (ou interferir minimamente) no funcionamento da aplicação. O estudo proposto neste trabalho visa alcançar, como um de seus resultados, uma aproximação entre os modelos teóricos e práticos, tornando mais realistas as considerações teóricas. Isso possibilitará estudos mais detalhados e uso prévio de simulações, antes que as aplicações sejam colocadas em produção. Assim, este trabalho possui impactos sociais, econômicos e ambientais na medida em que poderá minimizar os riscos de ocorrência de catástrofes.

O desenvolvimento da ferramenta de simulação contribuirá para a análise do comportamento de sistemas de tempo real que utilizam as políticas de escalonamento RM e

EDF, permitindo a escolha de uma das políticas em relação à outra a partir do critério de resiliência.

6. AGRADECIMENTOS

Os autores agradecem à FAPESB e ao IFBA pelo apoio financeiro ao desenvolvimento deste trabalho. Este projeto não implicou em investimentos financeiros relativos à infraestrutura, pois utilizará as instalações do Grupo de Pesquisa em Sistemas Distribuídos, Otimização, Redes de Computadores e Tempo Real, além de software livre e licenciado pelo IFBA.

7. REFERÊNCIAS

Fengxiang Zhang; Burns, A., "**Schedulability Analysis for Real-Time Systems with EDF Scheduling,**" Computers, IEEE Transactions on , vol.58, no.9, pp.1250,1258, Sept. 2009 doi: 10.1109/TC.2009.58

Giorgio C. Buttazzo. **Rate monotonic vs. edf: Judgment day.** *Real-Time Syst.*, 29(1):5–26, 2005.

Jean-Marie Farines, Joni da Silva Fraga, and Rômulo Silva de Oliveira. **Sistemas de Tempo Real. Departamento de Automação e Sistemas** - Universidade Federal de Santa Catarina, Florianópolis, Santa Catarina, 2000.

John A. Stankovic. **Misconceptions about real-time computing: A serious problem for next-generation systems.** *Computer*, 21(10):10–19, 1988.

John Lehoczky, Liu Sha, and Ye Ding. **The rate monotonic scheduling algorithm: Exact characterization and average case behavior.** In *Real Time Systems Symposium, 1989., Proceedings*, pages 166–171, Santa Monica, California, USA, 1989.eraliest

Melliari-Smith, P.M.; Moser, L.E. **Progress in real-time fault tolerance.** In: *Reliable Distributed Systems, 2004. Proceedings of the 23rd IEEE International Symposium on* , vol., no., pp. 109- 111, 18-20 Oct. 2004.

Paulo Eigi Miyagi and Emilia Villani. **Mecatrônica como solução de automação.** *Revista Ciências Exatas*, 9/10(1-2):53–59, 2004.

Ren C. Luo. **Sensors and actuators for intelligent mechatronic systems.** *Industrial Electronics Society, 2001. IECON '01. The 27th Annual Conference of the IEEE*, 3:2062 – 2065, 2001.

Rolf Iserman. **Modeling and design methodology for mechatronic systems.** *IEEE/ASME Transactions on Mechatronics*, 1:16–28, 1996.

Taisy Silva Weber. **Tolerância a falhas: conceitos e exemplos.** Instituto de Informática – Universidade. 2010.