

MECANISMO DE UM ESCALONADOR DE TAREFAS BASEADO EM SMP2

Leandro Toss Hoffmann e Leonel Fernando Perondi

Instituto Nacional de Pesquisas Espaciais
Programa de Pós-graduação em Engenharia e Tecnologia Espaciais
Área de concentração em Engenharia e Gerenciamento de Sistemas Espaciais
São José dos Campos - SP
hoffmann@dss.inpe.br, perondi@las.inpe.br

Resumo. *Dentro do escopo da missão CBERS-3 está previsto o desenvolvimento de um simulador computacional do satélite, o SIMC3. Esse simulador já conta até o presente momento com dois protótipos implementados e na sua versão final terá como objetivo atender as necessidades de treinamento dos operadores do centro de controle, bem como auxiliar em análises de missão, mediante simulação de cenários de operação do CBERS-3. Atualmente, o SIMC3 é um simulador de tempo real, sendo o tempo de simulação sincronizado com o relógio do computador. Contudo, para atender aos requisitos de uso para análise de missão é necessário que o SIMC3 permita que o tempo de simulação seja acelerado, de modo que o usuário possa obter o resultado de uma simulação de várias órbitas, o mais breve possível. Visando sanar essa carência, o presente trabalho propõe um mecanismo de escalonamento de tarefas que possa ser incorporado na arquitetura de software existente do SIMC3. Para tanto, a concepção do escalonador é baseada no padrão de ambiente de simulação SMP2. O uso de padrões no desenvolvimento de sistemas de dados espaciais é uma tendência entre as principais agências espaciais, pois reduzem o custo da missão, através do reuso de componentes e flexibilização para a portabilidade e a interoperabilidade do software. Como produto deste trabalho, chegou-se a uma especificação de classes que implementam os serviços de escalonamento indicados pelo SMP2, mas que ao mesmo tempo minimiza o impacto de modificações na arquitetura atual do SIMC3. O conjunto de classes foi validado por meio de simulações realizadas com um protótipo implementado em C++. Os resultados demonstraram a viabilidade de uso do mecanismo de escalonamento de processos no SIMC3.*

Palavras-chave: *simulador de satélites, escalonador de tarefas, SMP2, Padrões de sistemas de dados.*

1 INTRODUÇÃO

O Simulador do CBERS-3 (SIMC3) é um projeto que vem sendo desenvolvido pela Divisão de Desenvolvimento de Sistemas de Solo (DSS) no INPE, dentro do escopo de trabalho da missão CBERS-3. O seu objetivo é implementar uma ferramenta computacional de tempo real que permita simular o funcionamento operacional do CBERS-3, com fidelidade suficiente para atender aos usos de: treinamento de operadores do centro de controle de satélites; e análise de missão, mediante diferentes cenários de operação do satélite. Os cenários de operação podem envolver uma determinada configuração de equipamentos ligados ou desligados, variações de parâmetros orbitais e atitude e ainda, para um simulador de grande fidelidade, situações de falhas de equipamentos, entre outros.

Neste escopo, e visando a criação de uma arquitetura de software flexível que possa ser gradativamente construída e refinada, os dois primeiros protótipos do SIMC3 utilizam um conjunto central de classes para suportar o desenvolvimento de modelos e gerenciamento da simulação, que será apresentada em maiores detalhes na seção seguinte.

Ainda que a arquitetura de classes seja flexível, uma de suas limitações atuais consta que o mecanismo de execução dos modelos de simulação está baseado em uma classe de temporização, que é atrelada ao relógio do computador. Isto implica que o relógio de simulação, estando sincronizado a este relógio, não pode realizar rodadas de simulação com o tempo de simulação

acelerado ou retardado. Ainda que essa limitação não impeça o uso do simulador para cenários com requisito de tempo real (por exemplo, treinamento de operadores), acaba invalidando a sua aplicação em análises de missão, onde validar cenários que envolvam várias órbitas se torna inviável, à medida que o analista deve aguardar um tempo desnecessariamente grande para obter o resultado da simulação.

Visando contornar essa limitação, o presente trabalho propõe um mecanismo alternativo para a execução dos modelos de simulação, como uma alternativa ao sistema de temporização que atualmente vem sendo empregado na arquitetura do SIMC3. Este mecanismo consiste em um escalonador de tarefas, que além de gerenciar a fila de processos a ser executados e despachá-los, também permite o controle da velocidade de avanço do relógio de simulação.

O uso de escalonadores de processos em ambiente de simulação computacional é recomendado pelo padrão Portabilidade de Modelo de Simulação (SMP – *Simulation Model Portability*) (ESOC, 2005), e por se tratar de um padrão amplamente utilizado nos projetos de simulação de espaçonaves da Agência Espacial Européia (ESA – *European Space Agency*), será consultado no desenvolvimento deste trabalho.

A busca por padronização no desenvolvimento de sistema de dados espaciais tem sido muito praticada e seu uso tem se demonstrado uma tendência entre as principais agências espaciais. Entidades como o Comitê Consultor para Sistemas de Dados Espaciais (CCSDS – *The Consultative Committee for Space Data Systems*) (CCSDS, 2010) e a Cooperação Européia para Padronização Espaciais (ECSS – *European Cooperation for Space Standardization*) (ECSS, 2010) avançam na padronização de atividades em vários níveis da atividade espacial. O objetivo macro dessas iniciativas de padronização é buscar por soluções de implementação que primam pelo reuso, portabilidade e interoperabilidade de sistemas, trazendo assim economia de recursos nos processos de ciclo de vida da missão. Além disto, os padrões documentam caminhos para implementação de soluções que já foram acertadas e validadas, reduzindo os esforços no desenvolvimento de produtos semelhantes.

O SMP, agora na sua segunda versão SMP2, segue este caminho e aponta padrões de interface e serviços no âmbito do desenvolvimento de simuladores de satélites, orientado ao desenvolvimento baseado em modelos e em componentes. Neste contexto, o escalonador proposto neste trabalho é especificado como um conjunto de classes, cujos serviços são acessíveis por meio de interfaces bem definidas. A validação do modelo de classes é realizada através de simulações computacionais, executadas por um protótipo implementado em C++.

O restante deste trabalho está organizado da seguinte maneira: na seção 2, a arquitetura de software atual do SIMC3 é apresentada; a seguir, na seção 3, os conceitos do SMP2 são sumarizados; na seção 4 o mecanismo proposto de escalonamento de tarefas para o SIMC3 é discutido, na seção 5; a validação do modelo pela implementação do protótipo, bem como os resultados obtidos por meio de simulações são apresentados na seção 5; finalmente, na seção 6, as conclusões e sugestões de trabalhos futuros são expostas.

2 A ARQUITETURA DO SIMC3

A arquitetura central do Simulador do CBERS-3 baseia-se em um conjunto de classes representado no diagrama da Fig. 1, no qual se destacam inicialmente as classes *CSIMController* e *CSIMAbstractModel*. A *CSIMController* é uma classe fundamental da arquitetura, pois é responsável pelo controle da simulação, envolvendo operações de inicialização, execução e finalização da simulação. Já a classe *CSIMAbstractModel* é uma classe abstrata que especifica pontos de entrada de execução comuns a todos os modelos do satélite que serão implementados. Portanto, modelos como o térmico, do computador de bordo, do ambiente, entre outros, devem especializar essa classe, cujo ponto de entrada é o método *Behavior*.

De forma genérica, todos os pontos de entrada são escalonados para execução através de um temporizador do sistema operacional Windows. Por decisões de projeto e reuso de software, os temporizadores são implementados por uma classe *CSPTTimer*, desenvolvida no escopo de outro projeto da divisão (CARDOSO et. al., 2006). Essa classe também utiliza um mecanismo de herança para definir o ponto de entrada. Sendo assim, cada modelo codificado pode especificar e

implementar uma classe derivada *CSPTTimer* para utilizá-la como escalonador exclusivo. Por exemplo, na Fig. 2, o controlador da simulação agenda a execução de um modelo genérico através do objeto de sua classe de temporização *CSIMGeneralTimer*.

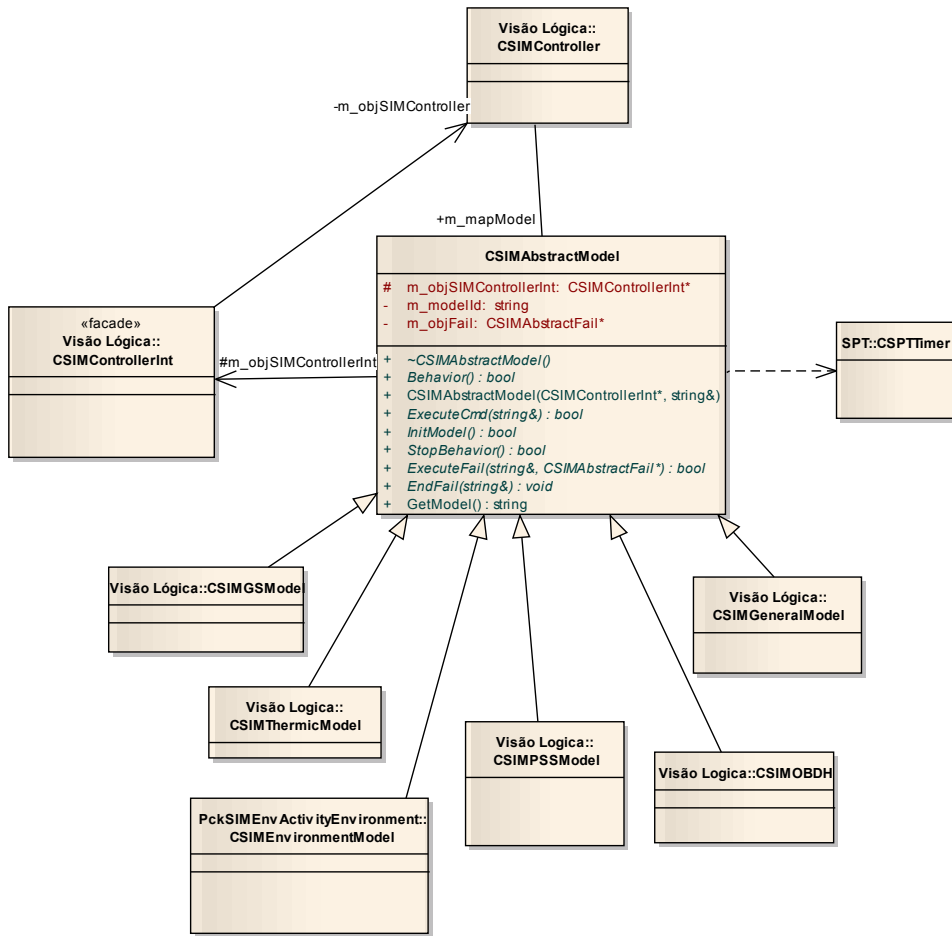


Figura 1. Diagrama de Classes simplificado da arquitetura núcleo do SIMC3.

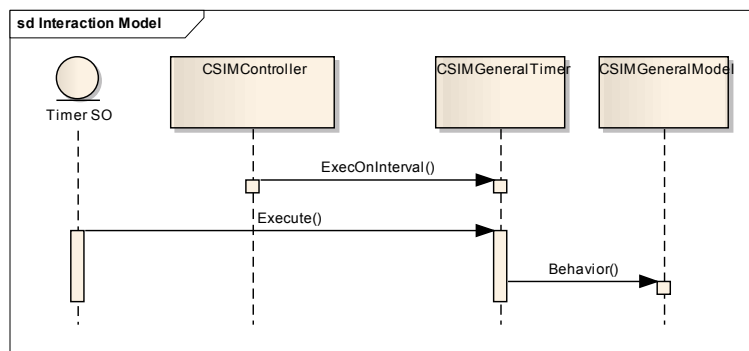


Figura 2. Diagrama de interação para o cenário de temporização da execução de modelos.

Este mecanismo sugere o SIMC3 como um simulador inerentemente de eventos discretos, uma vez que a execução dos modelos é agendada e controlada por um temporizador. Contudo, não há impedimentos para que os modelos simulem internamente sistemas contínuos, desde que estes respeitem limites temporais para sua execução. (Se o tempo de execução de um modelo ultrapassar os intervalos de disparo de seu temporizador, a simulação se torna inválida.)

Ao se utilizar a temporização do sistema operacional, via implementação do *CSPTTimer*, para disparar diretamente a execução dos modelos, a evolução do tempo de simulação fica naturalmente sincronizada ao relógio do PC, conforme dito anteriormente. Assim, busca-se em padrões como o SMP2, descrito a seguir, modelos de escalonadores que possam contornar essa limitação.

3 SIMULATION MODEL PORTABILITY 2 (SMP2)

O reuso de modelos e implementações tem sido o grande motivador para o desenvolvimento de padrões de desenvolvimento de sistemas de dados espaciais. Na área de simulação, uma dessas iniciativas é o *Simulation Model Portability* (SMP) proposto e utilizado pela Agência Espacial Européia (ESA – *European Space Agency*) (ESOC, 2005).

Esse padrão visa promover principalmente a portabilidade de modelos entre diferentes ambientes de simulação e sistemas operacionais, em uma filosofia de desenvolvimento orientada a modelos (*Model Driven Architecture*). Na sua segunda versão, SMP2, o padrão faz uso extensivo do conceito de orientação a objetos e representação de metadados por XML.

3.1 Visão de Alto Nível do SMP2

Para atender aos seus objetivos, o SMP2 parte de uma visão de alto nível (Fig. 3) dividida em camadas que representam:

- o mundo físico (o sistema de engenharia modelado);
- o sistema modelado para simulação; e
- o sistema de software que implementa o modelo de simulação.

O SMP2 trata da segunda e terceira camada, mas, por utilizar como entrada dados de engenharia, interage fortemente com a primeira camada (ESOC, 2005).

A segunda camada representa um Modelo Independente de Plataforma (PIM – *Platform Independent Model*), na qual o objetivo é especificar os modelos a serem simulados da melhor maneira possível, sem se preocupar como os mesmos serão implementados ou executados. Nessa etapa, modelos genéricos já existentes podem servir de base para a especialização de modelos específicos. A terceira camada trata do Modelo Específico de Plataforma (PSM – *Platform Specific Model*), ou seja, o código objeto dos modelos para serem executados em uma determinada arquitetura de computação.

Conforme representado na Fig. 3, a visão de alto nível ainda é diferenciada em duas colunas, correspondendo a definição de artefatos à coluna da esquerda, e a instâncias desses artefatos à coluna da direita.

Essa visão viabiliza, além dos objetivos já expostos do SMP2, criar ambientes de simulação flexíveis, baseados em modelos previamente desenvolvidos para a engenharia, e aplicáveis ao ciclo de vida do processo e produto.

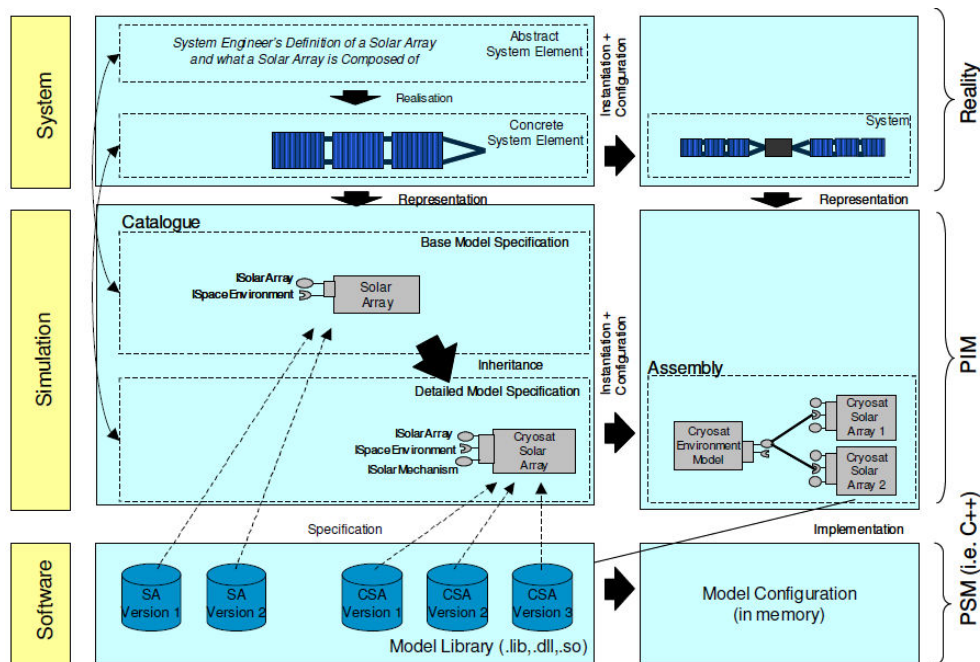


Figura 3. Visão de Alto Nível do SMP2. FONTE: (ESOC, 2005).

3.2 Arquitetura de Simulação

O SMP2 promove uma arquitetura baseada em componentes, onde todos os elementos de simulação são pequenas porções de software independentes. Nesta abordagem, os componentes oferecem pontos de conexão para a prestação de serviços e a comunicação entre eles se dá por meio dessas interfaces. Por se tratarem de interfaces padrões, a substituição de modelos e demais elementos da simulação pode ser efetuada em tempo de execução de maneira bastante dinâmica.

Utilizando esse conjunto de interfaces base, a arquitetura típica de um simulador implementando o padrão SMP2 está ilustrada na Fig. 4. O simulador é um conjunto de componentes que descreve o comportamento de modelos. Esses modelos interagem entre si e com o ambiente de simulação. O ambiente de simulação, por sua vez, consiste em uma camada de adaptação do SMP2 ao ambiente nativo de simulação, além de um conjunto de componentes que implementam serviços comuns, presente na maioria das simulações, conforme descrito a seguir.

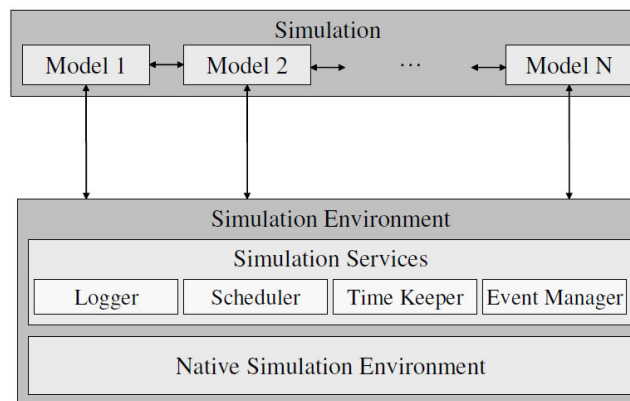


Figura 4. Arquitetura típica do SMP2. FONTE: [ESOC, 2005].

3.3 Serviços comuns de um ambiente de simulação

Quatro serviços mandatórios são definidos pelo SMP2:

- *Logger*: permite arquivar mensagens de *log*, tais como eventos, alertas e erros;
- *Scheduler*: é um serviço de escalonamento de eventos que devem ser executados em instantes de tempo ou de forma cíclica;
- *Time Keeper*: provê diferentes formatos de tempo;
- *Event Manager*: implementa mecanismos para registro e sinalização de eventos globais, que serão distribuídos aos modelos.

Os serviços de escalonamento (*Scheduler*) e acompanhamento do tempo (*Time Keeper*) estão intimamente relacionados e são de maior interesse para o escopo deste trabalho.

As interfaces do serviço escalonador permitem adicionar, alterar e remover tarefas a ser executadas em um determinado instante de tempo (ESOC-b, 2005). Para a adição, os componentes registram pontos de entrada (*entry points*) que serão invocados pelo escalonador no momento certo. Esses pontos de entrada são referências para operações (p.e. métodos de classes) que não devem ter argumentos e nem retornar valores.

As referências de tempo utilizadas pelo escalonador são manipuladas pelo serviço *Time Keeper* e podem ser representadas em 4 tipos de tempo (ESOC, 2005):

- **Tempo de simulação**: é um tempo relativo discreto utilizado somente no escopo da simulação. Iniciado com o valor zero e progride positivamente. Pode assumir os seguintes modos:
 - *Tempo real*: o tempo de simulação avança em tempo real, tipicamente definido pelo relógio do computador (PC).
 - *Acelerado*: o tempo de simulação avança de forma relativa ao tempo real, utilizando um fator constante de aceleração (um fator multiplicador). Se esse fator

é < 1 , o tempo simulado progride mais lento que o tempo real; ou mais rápido se for > 1 . Caso o fator seja igual a 1, equivale ao tipo *tempo real*;

- *Execução livre*: o tempo progride tão rápido quanto possível, limitado apenas pelo tempo de execução das tarefas disparadas pelo escalonador;
 - *Depuração*: a simulação é executada passo-a-passo, utilizando pontos de parada (*break points*), de forma a permitir a inspeção de valores de dados e rastrear chamadas na simulação.
- **Tempo da Época**: é um tempo de referência absoluto, utilizado para expressar o tempo de simulação em unidades de data e hora.
 - **Tempo de Missão**: é semelhante ao tempo de simulação, contudo o seu valor inicial está relacionado ao instante de tempo de início da missão.
 - **Tempo Zulu**: é o tempo relativo ao relógio do computador, ou seja, ao tempo civil. O nome Zulu refere-se a letra Z, designada tradicionalmente para referenciar o tempo de Greenwich.

A seguir, o serviço de escalonamento do SMP2, implementado no projeto da Infraestrutura de Software para Modelagem de Satélites (SIMSAT - *Software Infrastructure for Modelling of SATellites*) do Centro Europeu de Operações Espaciais (ESOC – *European Space Operations Centre*), é discutido em maiores detalhes.

Adicionalmente, informações sobre um componente de modelo que define um conjunto padrão de serviço de simulação está descrito no documento ESOC-b, 2005. Seu mapeamento para linguagem C++ é apresentado em ESOC-c, 2005. Além disto, um metamodelo para definição de modelos SMP2, por meio de uma Linguagem de Definição de Modelos de Simulação (SMDL – *Simulation Model Definition Language*) é provida em ESOC-d, 2005.

3.4 O serviço de escalonamento do SIMSAT

Com o intuito de suportar o desenvolvimento e o uso de simuladores de espaçonaves de tempo real, o Centro Europeu de Operações Espaciais (ESOC – *European Space Operations Centre*) concebeu o a Infraestrutura de Software para Modelagem de Satélites (SIMSAT - *Software Infrastructure for Modelling of SATellites*) (ESA, 2007). Esse produto foi especificamente criado à luz do SMP2, suportando então os seus modelos e implementando os seus serviços padronizados.

O serviço de escalonamento de execução de modelos do SIMSAT provê uma resolução de nano segundos, suportando uma simulação de tempo real *soft*. Para tanto, o seu mecanismo de implementação define um limite de latência na sincronização entre o relógio do PC e o relógio de simulação. Essa latência é o intervalo máximo que o escalonador pode estar à frente ou atrás da progressão do tempo real.

O escalonador também implementa um mecanismo para a execução de tarefas mais rápida ou mais devagar que o tempo real. Este ajuste é feito por um fator de velocidade da simulação, neste trabalho chamado de fator multiplicador, que se trata da razão da progressão do tempo de simulação pelo relógio da parede (tempo Zulu). É uma funcionalidade bastante útil para se cobrir rapidamente períodos longos de baixa atividade de simulação ou analisar detalhadamente períodos curtos com muitos eventos rápidos.

O SIMSAT provê ainda recursos de escalonamento para a execução dos processos passo-a-passo, de modo que o desenvolvedor possa analisar sua execução, além da execução de tarefas paralelas, utilizando um escalonador concorrente. Esses recursos adicionais não serão considerados no escopo desse trabalho.

Um exemplo de operação do escalonador do SIMSAT, para simulação de tempo real *soft*, é ilustrado na Figura 5. Nela um gráfico mostra a relação entre o tempo de simulação (representado pelo eixo horizontal) e o tempo Zulu (representado pelo eixo vertical). A progressão do tempo é mostrada pela linha preta. Quando a linha avança no sentido horizontal, significa que o escalonador avança o tempo de simulação de um instante discreto para o próximo. Quando a linha avança no sentido vertical, significa que uma tarefa está executando, logo o tempo de simulação permanece constante, enquanto que o tempo do computador avança.

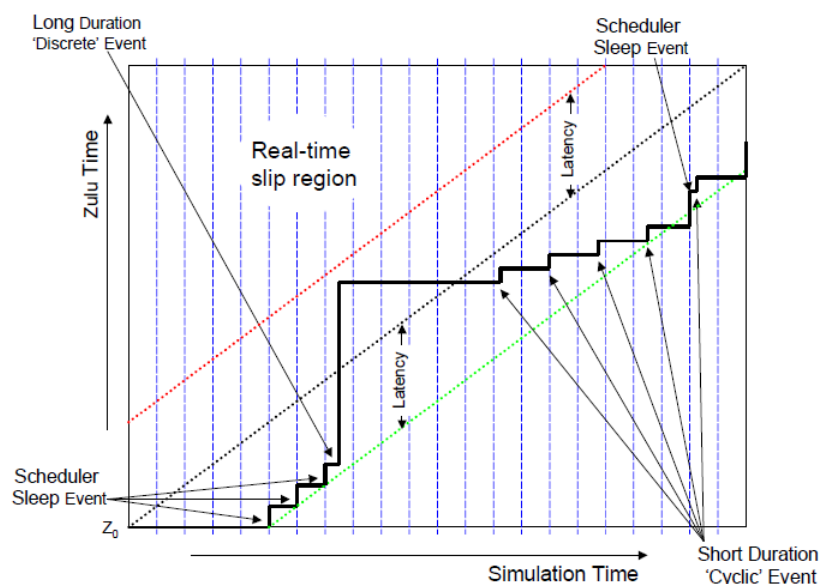


Figura 5. Gráfico ilustrativo da operação do escalonador de tempo real *soft* do SIMSAT.
FONTE: [ESA, 2007].

A situação ideal de sincronismo entre o tempo de simulação e o tempo do PC é representada no gráfico pela linha preta pontilhada. As linhas pontilhadas vermelha e verde representam, respectivamente, os limites superiores e inferiores de latência, que neste caso é 500 ms. O escalonador executa um ciclo de sincronização a cada 10 ms (representado pelas linhas verticais pontilhadas em azul). Neste ciclo, a diferença entre o tempo de simulação e o tempo do PC é calculada e utilizada para decidir se o próximo processo será executado ou um *delay* (*scheduler sleep event*) será introduzido na simulação.

Ainda no exemplo da Figura 5, nos primeiros 500 ms da simulação (que é igual ao tempo de latência), o tempo de simulação é rapidamente avançado e eventos de *sleep* são executados pelo escalonador. Aproximadamente no tempo de simulação 750 ms um evento de grande duração é executado, que devido à folga de latência, permanece dentro dos limites tolerados. Em seguida, pela ausência de novos processos por um período de simulação, o escalonador avança o tempo de simulação para o instante onde se inicia a execução da série de eventos cíclicos programados. Esses eventos executam tão rápido que logo o escalonador intercepta novamente a linha verde. Trabalhando com esse mecanismo, o escalonador pode utilizar a execução de processos leve para manter uma folga de tempo que será utilizada para a execução de processos longos. Ainda assim, caso o um processo leve muito tempo para ser executado, de forma que o limite da linha vermelha seja ultrapassado, o escalonador deverá resincronizar as tarefas. Esta situação é bastante indesejada, especialmente se existir outros sistemas externos que dependam de informações geradas pelos modelos da simulação.

4 ESCALONADOR DE PROCESSOS PROPOSTO

Uma arquitetura de componentes e serviços, nos moldes daquele apresentado pelo SMP2, é apropriada para um simulador do tipo do SIMC3. Porém, sua implementação nos conformes do padrão SMP2 levaria a um esforço indesejado de modificação da arquitetura existente do SIMC3.

Para solucionar o problema de flexibilidade na aceleração do tempo de simulação, sem trazer grandes impactos ao código atual do SIMC3, um mecanismo de escalonamento de processos é empregado. Este escalonador é inspirado nos serviços de escalonamento do SMP2, mas propondo adequações a arquitetura existente do SIMC3, conforme ilustrado no diagrama de classes da Fig. 6. A classe *CSIMScheduler* é responsável em gerenciar e despachar os processos e eventos a ser executados. Como o SMP2 prevê, essa classe define interfaces para serviços de manipulação da sua fila de processos (como por exemplo, a adição de novos processos *AddOnInterval* e *AddOnTime*)¹.

¹ a rigor deve prover também serviços de remoção e alteração de processos.

A fila de processos é implementada no *CSIMScheduler* por um *container* de *CJob*, que é uma classe que armazena a referência para o ponto de entrada do processo, o *timestamp* de simulação que deverá ser executado, além do número e intervalo entre as execuções.

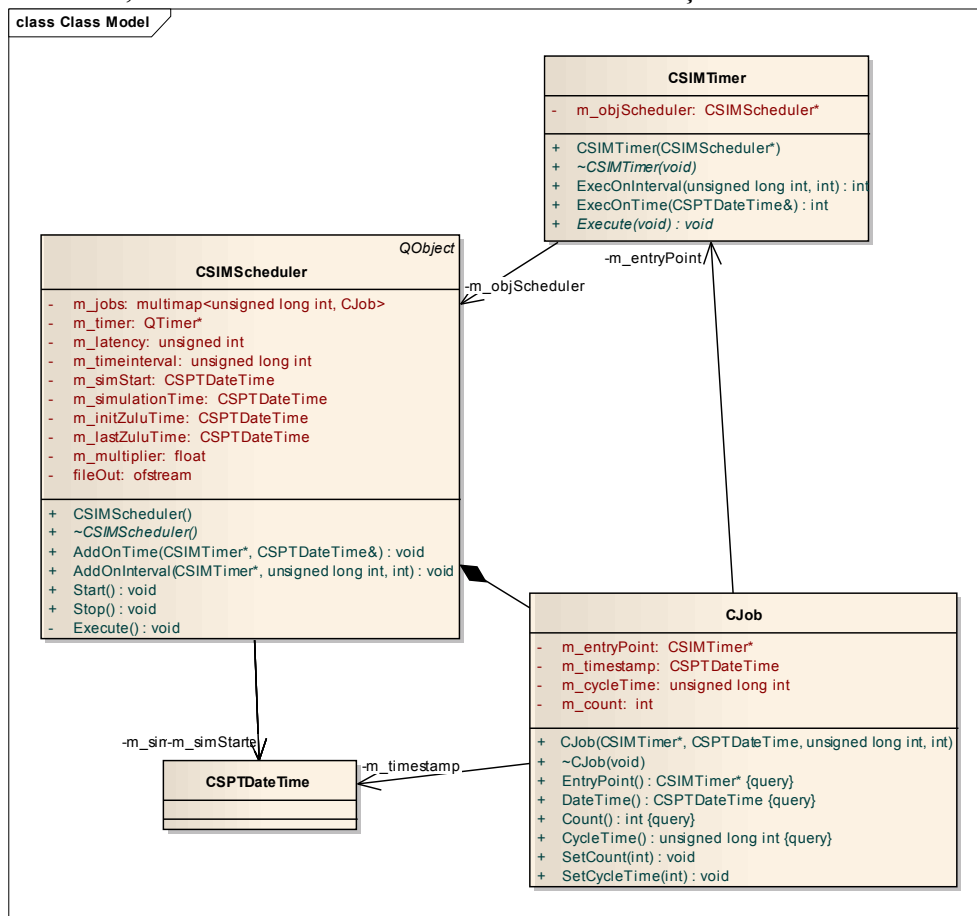


Figura 6. Diagrama de Classes da proposta de escalonador de processos para o SIMC3.

Ambas as classes *CSIMScheduler* e *CJob* utilizam a classe *CSPTDateTime* para manipulação de formato de tempos e recuperação do tempo do computador.

Para minimizar o impacto de modificações na codificação atual do SIMC3, os modelos que implementavam classes derivadas da *CSPTTimer*, agora devem derivar da classe *CSIMTimer*. Essa classe nova respeita as interfaces dos serviços da *CSPTTimer*, repassando a função de agendamento de tarefas a *CSIMScheduler* e tendo o serviço virtual *Execute* como ponto de entrada padrão. A execução de um modelo genérico da arquitetura do SIMC3, incorporando esse mecanismo, seria efetuada nos moldes do diagrama de interação da Fig. 7.

Portanto, o escalonador é executado baseado em uma referência externa, provida também por um temporizador do sistema operacional. A sua decisão em despachar ou não um processo se dá na mesma forma da estratégia definida para o escalonador do SIMSAT (ESA, 2007): deve-se avançar o tempo de simulação e executar um processo, caso esse tempo causa um desincronismo em relação ao tempo do computador (tempo Zulu), dado um intervalo arbitrário de latência.

Logo, uma possível regra de despacho, proposta neste trabalho, pode ser dada pela equação

$$TS_{job} - l \leq (TS_Z - TS_{Z0})m, \quad (1)$$

onde:

TS_{job} é o *timestamp* de simulação do próximo processo,

l é o tempo de latência,

TS_Z é o *timestamp* corrente do relógio do PC,

TS_{Z0} é o *timestamp* do relógio do PC no início da simulação e

m é o fator multiplicador.

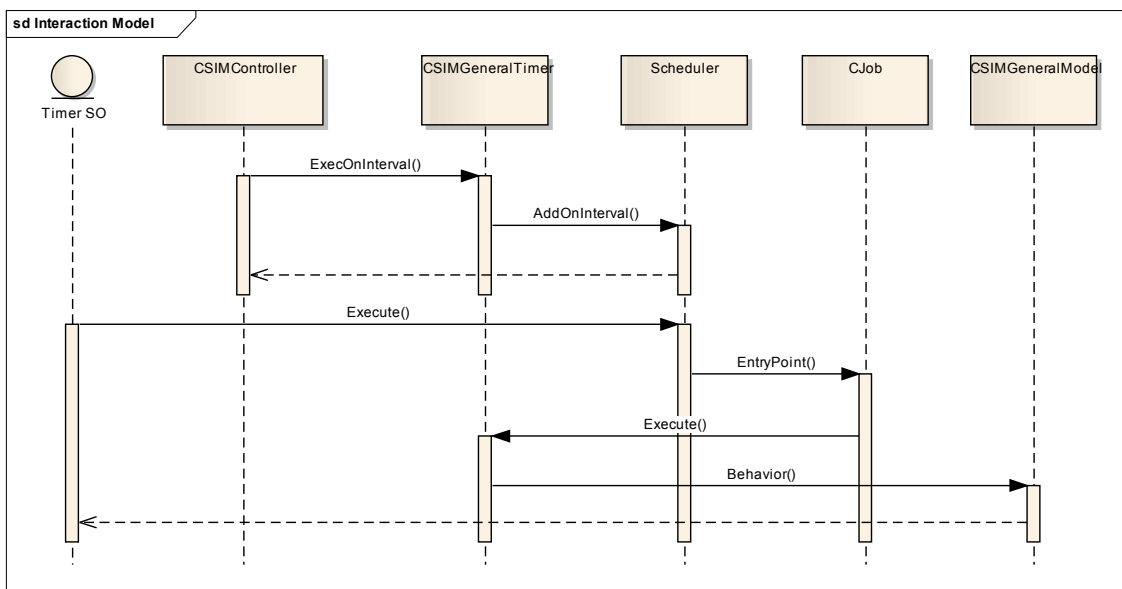


Figura 7. Diagrama de interação para o cenário de temporização da execução de modelos, pelo mecanismo de escalonamento de processos.

Quando a regra de despacho é satisfeita, o escalonador avança o relógio de simulação para TS_{job} . Caso contrário o tempo de simulação é sincronizado com o tempo do computador acrescido do tempo de latência, permanecendo o escalonador em modo *sleep* por 10 ms.

O fator multiplicador $m > 0$ permite acelerar ou retardar o tempo de simulação, alterando a regra de despacho e, portanto, a forma de sincronização do relógio de simulação com o relógio do computador. Para $m = 1$, a simulação corre em velocidade compatível com o relógio do computador, para $m > 1$, a simulação corre mais rápida e para $0 < m < 1$, mais lenta.

5. SIMULAÇÕES E RESULTADOS

Para validação do modelo do escalonador proposto, foram realizadas simulações computacionais, por meio de um protótipo implementado em C++. Esse protótipo especificou uma classe *CProcess*, derivada da *CSIMTimer*, responsável em executar uma carga de processamento. Essa classe tem como propósito emular a carga de processamento de modelos implementados no SIMC3.

A implementação foi codificada em Microsoft Visual Studio 2003, no qual foi feito uso da biblioteca Qt 4.3. para agendamento de temporizações. Na execução do programa, empregou-se um microcomputador PC Pentium 4 2.26 GHz, com 1 GB de memória RAM, rodando Windows XP.

Ao todo, foram executadas 3 seções de simulação, cada uma visando uma análise diferente, empregando-se o tempo de latência igual a 500 ms:

- Seção 1: tarefas semelhantes aquelas apresentadas na Figura 5 foram escalonadas, com o intuito de se comparar os resultados com a operação descrita em ESA, 2007;
- Seção 2: tarefas com duração e início aleatórios foram escalonadas em um horizonte de aproximadamente 10 segundos, verificando-se a robustez do escalonador em respeitar os limites de latência;
- Seção 3: o mecanismo de aceleração ou retardo da simulação, por meio de uso de fator multiplicador, foi analisado.

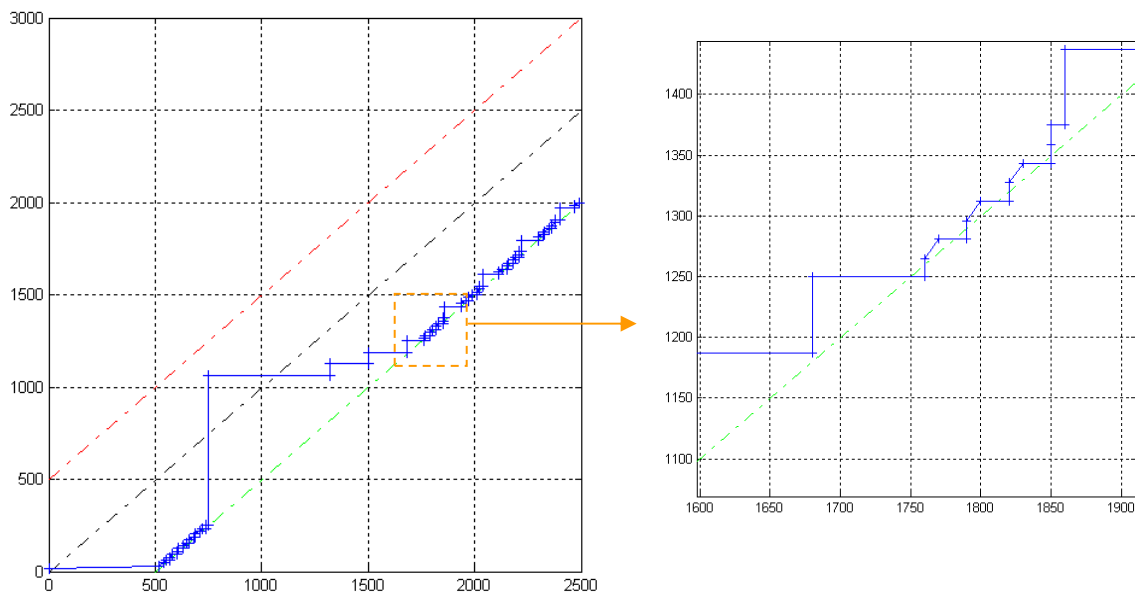
5.1. Seção de simulação 1

Procurou-se nesta simulação escalonar processos que tivessem características aproximadas daqueles apresentados na Figura 5, conforme relacionado na Tab. 1. Observa-se pelos resultados dispostos na Tab. 1 que os *timesteps* de execução dos processos e os tempos de execução e intervalo estão dentro do esperado para o conceito de tempo real *soft*, com a latência de 500 ms adotada. Essa evidência é melhor visualizada por meio dos gráficos da Fig. 8, nos quais empregou-se a mesma notação da Figura 5: o eixo horizontal representa o tempo de simulação em ms, enquanto que o eixo vertical representa o tempo medido do PC; a tracejada preta representa a situação ideal de

sincronismo entre o tempo simulado e do PC, enquanto que as linhas vermelha e verde representam os limites de latência adotados. Como era de se esperar, o tempo simulado ficou dentro dos limites de latência, próximos ao limite inferior, sempre que possível. No *timestamp* simulado 750, o processo (1) entra em execução, e por ter uma carga computacional prevista de 800 ms, o tempo do PC avança de 250 ms para 1062 ms, enquanto que o tempo de simulação permanece constante. Ao término do processamento da tarefa 1, o escalonador rapidamente dispara a execução do segundo processo, avançando o tempo de simulação para 1320 ms. Com isso, consegue trazer a diferença de sincronismo entre os tempos simulado e do PC, para próximo ao limite inferior de latência. No detalhe do gráfico, na Fig. 8-b, a folga de processamento no *timestamp* do PC 1250 ms permite ao escalonador introduzir *delays* na simulação.

Tabela 1. Relação de simulações efetuadas e resultados obtidos.

Proc.	Timestamp (ms)		N.º de Exec.	Duração (ms)		Intervalo Ciclo (ms)	
	Simulado	Zulu		progr.	medido	progr.	medido
1	750	250	1	800	812	-	-
2	1320	1062	1	50	63;	-	-
3	1500	1125	6	50	62; 63; 62; 63; 62; 62.	180	63; 62; 188; 171; 188; 172.



(a) horizonte completo para a seção de simulação (b) detalhe aumentado do gráfico (a)

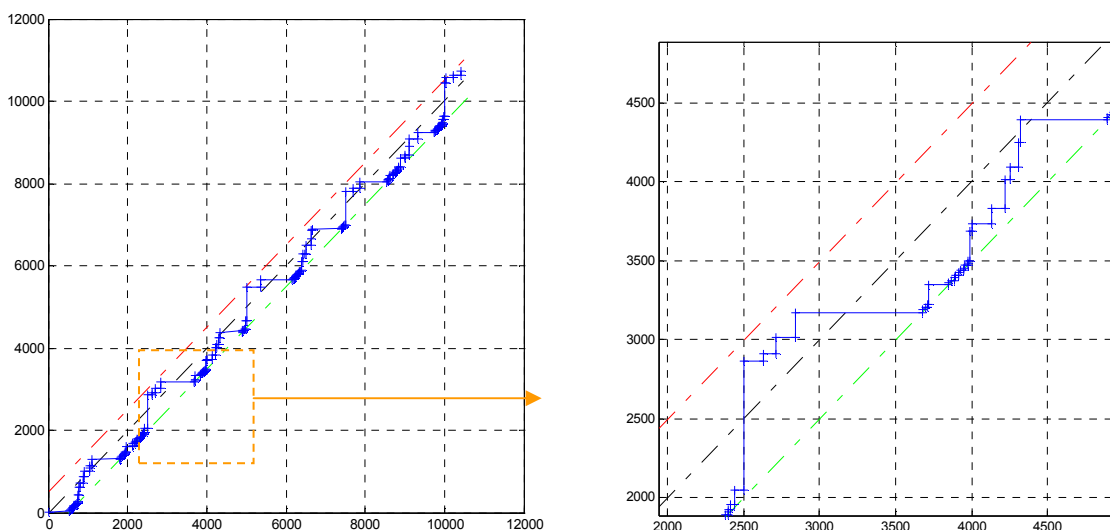
Figura 8. Progressão do tempo de simulação o horizonte de 2500 ms. A linha azul representa a relação entre o tempo simulador (eixo horizontal) e o tempo do PC (eixo vertical).

5.2. Seção de simulação 2

Para melhor avaliar o comportamento do escalonador, frente a processos com intervalo e duração aleatórios, nesta seção de simulação foram introduzidos:

- 1 processo cíclico executando a cada 2500 ms, com duração de processamento programada para 800 ms.
- Processos com uma única execução, intercalados em um intervalo aleatório com limites entre 0-500 ms, e carga computacional aleatória entre 0-200 ms.

O resultado da simulação para um horizonte de aproximadamente 10.000 ms, ilustrado na Fig. 9, indica a capacidade do escalonador em manter o sincronismo entre os relógios, dentro dos intervalos de latência de 500 ms. Contudo, observa-se que essa capacidade só é mantida, devido as características dos processos agendados. É evidente que processos com tempos computacionais elevados levariam a uma perda de sincronismo.



(a) horizonte completo para a seção de simulação (b) detalhe aumentado do gráfico (a)

Figura 9. Progressão do tempo de simulação o horizonte de 10.000 ms. A linha azul representa a relação entre o tempo simulador (eixo horizontal) e o tempo do PC (eixo vertical).

5.3. Seção de simulação 3

Nesta seção final de simulação, objetivou-se avaliar o mecanismo de variação de velocidade do relógio de simulação, pelo uso do fator multiplicador. Para tanto, 4 rodadas de simulação foram realizadas, variando-se o fator multiplicador nos seguintes valores: 1,0; 1,5; 2,0 e 0,5 vezes. Para todas as rodadas, os processos escalonados foram os mesmos, e suas características estão sumarizadas na Tabela 2.

Tabela 2. Características dos processos escalonados nas rodadas da seção de simulação 3.

Processo	Timestamp simulação (ms)	Duração programada (ms)	Intervalo de ciclo programado (ms)	Número de Execuções
1	750	800	-	1
2	1320	50	180	10
3	2400	500	-	1
4	3100	50	300	5
5	3050	40	200	5

Rodada 1,0x

Nesta rodada, utilizando-se o fator multiplicador de 1x, o resultado obtido assemelha-se aos já apresentados nas seções de simulação anteriores, uma vez que 1,0x não implica atraso nem aceleração do relógio de simulação. Através do gráfico da Fig. 10, observa-se que o escalonador pôde manter o sincronismo entre os tempos de simulação e do PC.

A última execução do processo 4 é iniciada no tempo simulado 4300 ms. Ao término dessa execução (o que estava programado para durar 50 ms) o *timestamp* do relógio do computador marcava 3875 ms. Esse fato está dentro da normalidade, provendo uma folga de 425 ms, devido ao tempo de latência empregado.

Rodada 1,5x

Da mesma forma que na rodada anterior, a última execução do processo 4 é iniciada no tempo de simulação 4300 ms. Por se utilizar um fator multiplicador de aceleração do tempo de 1,5 vezes, espera-se que o tempo de execução seja aproximadamente 33% menor ($100 - 100/1,5$). De fato, conforme pode ser visto no gráfico da Fig. 11, ao término da simulação, o tempo medido do relógio do computador marcava 2656 ms de simulação transcorrida. Esse valor, corrigido pelo fator

multiplicador, equivale a ter acelerado o tempo do computador até 3984 ms (o que está dentro do tempo de latência aceitável).

Contudo, tendo o processo (1) uma duração de processamento programada para 800 ms, o escalonador não pode garantir o sincronismo dentro dos limites de latência estipulados, para todo horizonte de simulação. Isto é observado no *timestamp* de simulação 750 ms, no momento da execução do processo (1). Ainda assim, o escalonador pôde recuperar o sincronismo, na sequência.

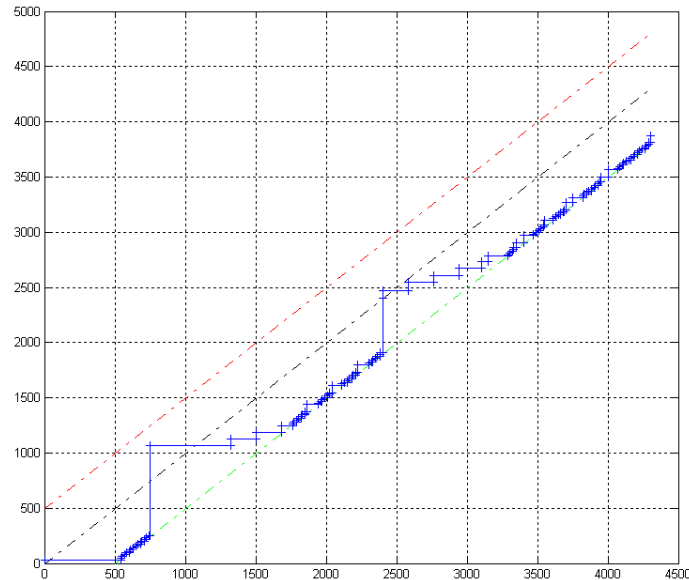


Figura 10. Progressão do tempo de simulação o horizonte de 4300 ms, empregando-se o fator multiplicador 1x. A linha azul representa a relação entre o tempo simulador (eixo horizontal) e o tempo do PC (eixo vertical).

Rodada 2,0x

Nesta rodada, utilizando-se um fator multiplicador igual a 2 vezes o tempo real, o esperado é que o tempo de execução fosse metade do tempo, em relação a simulação com fator 1x, ou seja, aproximadamente 1937 ms. Contudo, o resultado obtido foi de 2610 ms, o que significa 5220 ms se esse valor fosse corrigido pelo fator multiplicador. Este fato se deve porque utilizando-se um fator multiplicador de 2x, mas mantendo-se a complexidade computacional dos processos, não foi possível manter o sincronismo do tempo simulado com o tempo real acelerado, dentro dos valores de latência. Esse resultado fica claro através da Fig. 12, onde percebe-se que a linha azul ficou acima da linha vermelha. Devido à complexidade computacional dos processos, 2610 ms é o tempo mínimo necessário para processamento. De fato, ainda pela Fig. 12, nota-se que a partir do *timestamp* de simulação 750 ms, em nenhum momento foi necessário executar um processo de *delay* do escalonador, sendo todas tarefas disparadas na sequência, tão logo fosse possível.

Rodada 0,5x

Utilizando um fator multiplicador 0,5 vezes, significa retardar o relógio de simulação, de modo que o tempo total de simulação seja o dobro do tempo real. De fato, esse resultado foi observado ao final da última execução do processo (4), onde o tempo medido do relógio do PC foi de 7672 ms. Aplicado o fator multiplicador, o tempo do PC equivale a 3836 ms, o que está dentro do valor esperado para os limites de latência. O sincronismo dos tempos de simulação é mostrado no gráfico da Fig. 13, no qual se observa a folga de recurso computacional, surgindo então a necessidade do escalonador introduzir *delays* no escalonamento de processos.

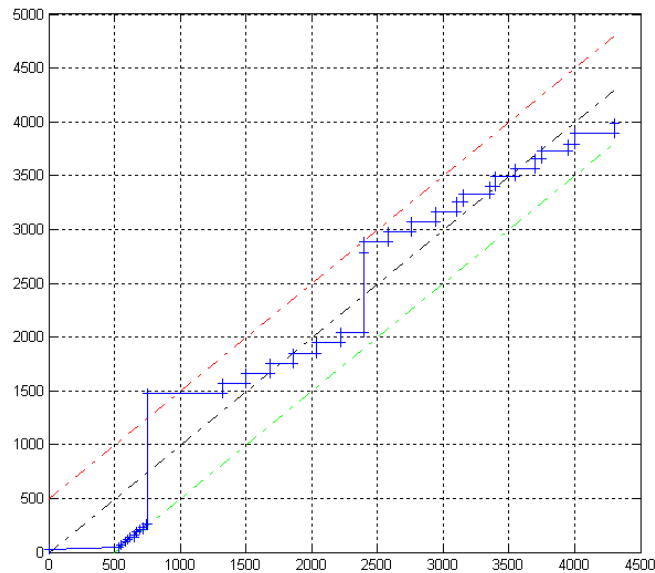


Figura 11. Progressão do tempo de simulação o horizonte de 4300 ms, empregando-se o fator multiplicador 1,5x.

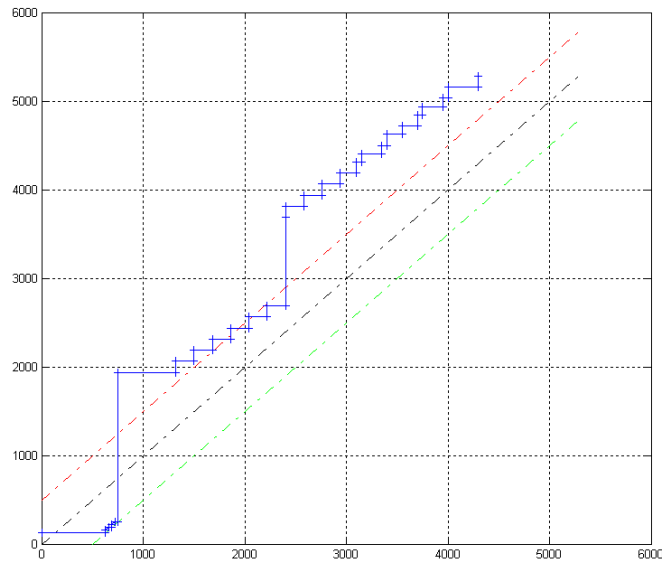


Figura 12. Progressão do tempo de simulação o horizonte de 4300 ms, empregando-se o fator multiplicador 2,0x.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou um mecanismo de escalonamento de processos que pode ser incorporado na arquitetura de software do SIMC3. O mecanismo é baseado nas interfaces de serviço de escalonamento definidas pelo padrão SMP2 e permite a aceleração e retardo do relógio de simulação.

Os resultados obtidos por meio da implementação de um protótipo de simulação demonstraram a sua viabilidade de uso para tempo real *soft*, desde que a carga computacional dos processos seja pequena. Através de uma análise para identificação do perfil dessa carga computacional, é possível se ajustar o tempo de latência do escalonador, bem como seu intervalo de espera (*sleep*), mas adequado para as necessidades do SIMC3. Adicionalmente, um tempo de latência adaptativo pode ser empregado, conforme discutido em ESA, 2007.

O impacto na implementação atual do SIMC3 é pequeno, sendo necessárias algumas modificações no uso da classe *CSPTTimer* pela *CSIMTimer*, e os respectivos construtores de suas classes derivadas. Dependente da estratégia utilizada para integração do escalonador na arquitetura de software do SIMC3, a classe *CSIMScheduler* poderá ser implementada com o padrão *singleton* ou

uma única referência de sua instância poderá ser mantida por alguma classe do simulador (como por exemplo, a *CSIMAbstractModel*). Além disso, a classe *CSIMDateTime*, que detém atualmente o tempo de simulação, poderá ter o seu serviço *SetComputerCurrentTime()* como um mecanismo para recuperação do tempo de simulação armazenado pelo *CSIMScheduler*.

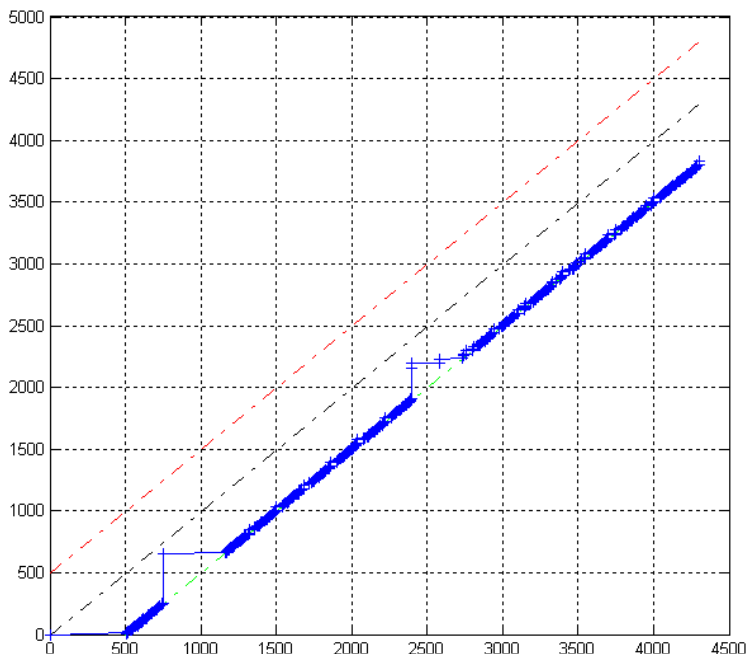


Figura 13. Progressão do tempo de simulação o horizonte de 4300 ms, empregando-se o fator multiplicador 0,5x.

7 AGRADECIMENTOS

O autor agradece a colaboração do professor Dr. Walter Abrahao dos Santos no desenvolvimento deste trabalho e da professora Dra. Ana Maria Ambrósio, gerente do pacote de trabalho SIMC3.

8 REFERÊNCIAS

- CARDOSO, Paulo E. ; BARRETO, Joaquim P. ; DOBROWOLSKI, Kléber M. ; CARDOSO, L. S.; HOFFMANN, L. T., 2006. “A Ground Control System for CBERS 3 and 4 Satellites”, In: Anais do SpaceOps, Roma, p. 6
- CCSDS, The Consultative Committee for Space Data Systems, 2010. “CCSDS.org”, online: <http://www.ccsds.org>, acessado em fev-2010.
- ECSS, European Cooperation for Space Standardization, 2010. “Collaboration website of the European Cooperation for Space Standardization”, online: <http://www.ecss.nl>, acessado em fev-2010.
- ESA, European Space Agency, 2007. “SIMSAT 4.0 Simulator Designer's User Manual”, EGOS-SIM-SIM-SUM-1001, p.301.
- ESOC, European Space Operations Centre, 2005. “SMP 2.0 Handbook”, EGOS-SIM-GEN-TN-0099, Issue 1, Revision 2, p. 134.
- ESOC-b, European Space Operations Centre, 2005. “SMP 2.0 Component Model”, EGOS-SIM-GEN-TN-0101, Issue 1, Revision 2, p. 158.
- ESOC-c, European Space Operations Centre, 2005. “SMP 2.0 C++ Mapping”, EGOS-SIM-GEN-TN-0102, Issue 1, Revision 2, p. 96.
- ESOC-d, European Space Operations Centre, 2005. “SMP 2.0 Metamodel”, EGOS-SIM-GEN-TN-0100, Issue 1, Revision 2, p. 125.