

6

Análise dos resultados obtidos

O capítulo anterior apresentou a solução a questão principal de pesquisa, obtida pelo fracionamento da mesma em questões secundárias e sua solução em passos. Estas questões secundárias foram respondidas conforme a pesquisa avançava.

Para comprovar a eficácia das soluções propostas, foram realizados experimentos quantitativos, cujos resultados foram analisados graficamente e pelos valores de saída.

As seções a seguir apresentam estes experimentos:

6.1

Suporte a VES implemetadas com MABS dirigidas por tempo

A pesquisa iniciou por responder as questões secundárias abaixo:

QS3: *Obter uma solução que atenda o controle de atraso em tempo de simulação através do estabelecimento de uma taxa estável para o avanço do tempo de simulação;*

QS4: *Obter uma solução que atenda o controle de atraso em tempo de simulação e que seja capaz de perceber tanto a necessidade de reduzir a taxa de avanço quanto a existência de recursos que permitam acelerar a execução sem comprometer a taxa máxima de atraso admissível*

A abordagem inicial não incluía o suporte a avanço de tempo de simulação dirigido por eventos ou a questões associadas a representações geoespacial, e o algoritmo de cálculo de f estava em uma versão que foi posteriormente evoluída.

Para comprovar a habilidade em atender as questões, um experimento foi executado para obtenção de dados quantitativos. O experimento constituía-se em uma simulação com um conjunto de atores simulados em um ambiente virtual. Estes atores representavam navios e submarinos, possuindo comportamentos associados à detecção e engajamento. Tempos simulados de carga de processamento foram inseridos no código do simulador a fim de permitir um experimento controlado.

Para redução de erros decorrentes de interferências externas, todas as execuções foram realizadas no mesmo ambiente de software e hardware.

Os agentes da simulação possuíam comportamentos individuais, e a simulação utilizou a abordagem de avanço do tempo de simulação em passos, centrados nos agentes e seus comportamentos. Para o experimento foi estabelecido que um atraso superior a 4% do tempo esperado para execução de um comportamento agendado seria crítico, e que atrasos maiores que 0,5% deveriam disparar o mecanismo de controle, alimentando o algoritmo que atua sobre o cálculo de f . O valor inicial de f utilizado foi 100 (i.e, cada segundo de tempo real correspondia a um avanço de 100 segundos no tempo da simulação).

6.1.1

Amostra de controle

Inicialmente, o experimento foi executado sem ativar o controle sobre o avanço do tempo da simulação. Esta abordagem otimista supunha que os agentes poderiam executar suas ações respeitando os limites estabelecidos para *tardiness* com os recursos computacionais disponíveis. Esta corrida ofereceu resultados que foram utilizados como referência, ie. amostra de controle.

Nesta primeira simulação foram utilizados 10 agentes e um passo de atualização correspondente a 10 segundos de tempo de simulação (fig. 6.1). Cada evento executado foi registrado em um arquivo de log, e o valor de *tardiness* é apresentado como uma cruz vermelha no gráfico.

A *tardiness* foi medida de acordo com o apresentado em 4.2.2, ou seja:

$$Tardiness = \frac{T_{executed}}{T_{programed}} - 1$$

Desta forma, *tardiness* representa o quanto a duração de uma ação executada excedeu o instante esperado para sua execução, em percentagem medida considerando a duração do passo para esta ação como referência.

A figura 6.1 apresenta que o valor de *tardiness* variou de 0,2 a 15, o que significa exceder o período previsto com valores que variaram entre 20% e 1500% o passo esperado em tempo de simulação. Estes valores indicam que esta simulação não representa de forma consistente o sistema real simulado, pois agentes não podem garantir que as ações agendadas ocorrerão quando previsto, com precisão no tempo. A imagem apresenta áreas de concentração de medidas que visualmente se assemelha a três linhas. Isto ocorre porque os comportamentos ocorrem em períodos diferentes e seu *tardiness* é proporcional a esse período.

O segundo experimento foi também executado sem ativar o controle do valor de f , porém o passo de execução foi aumentado de 10 segundos de tempo de simulação para 2 minutos. Este novo passo de execução deveria prover mais tempo

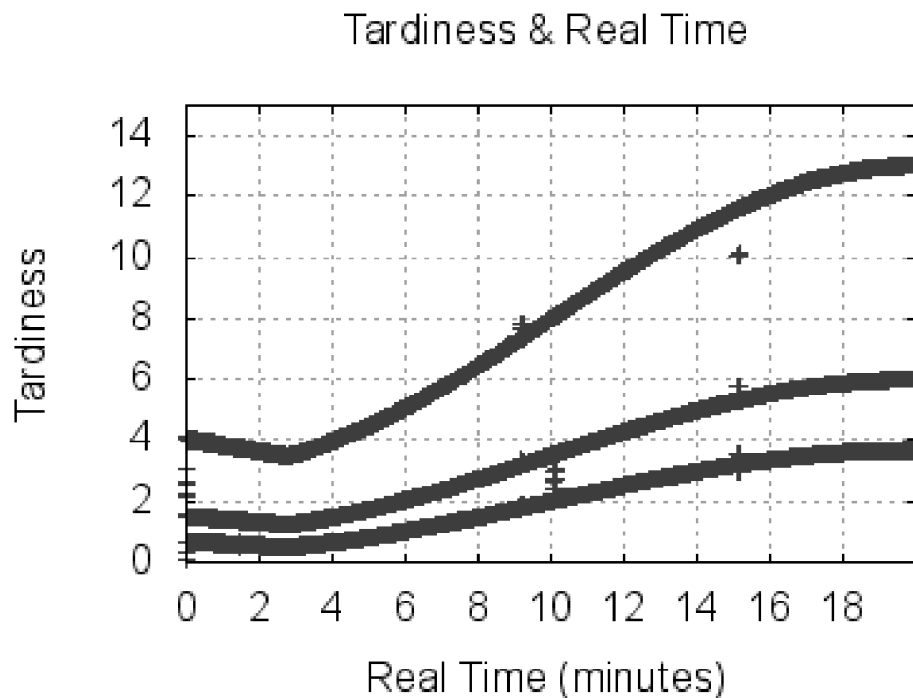


Figura 6.1: Tardiness - 10 agentes de simulação executando sem controle de f (10 seg. de passo)

real para os agentes executarem suas ações agendadas e reduzir o valor de *tardiness*. Como esperado, o valor de *tardiness* reduziu em relação ao observado no primeiro experimento, contudo os valores ficaram acima de 0,04, estabelecido como maior erro aceitável, atingindo 0.82 nos momentos de pico (fig. 6.2).

Os dois experimentos apresentados processaram 2.000 segundos de tempo de simulação, usando o valor de $f = 100$. Contudo os dados de saída destas simulações não são confiáveis, devido aos altos valores de *tardiness*.

6.1.2

Utilizando o controle de f

Os próximos gráficos apresentam *tardiness* em relação ao avanço do tempo de simulação, quando usando o controle automático da progressão do tempo de simulação através da atuação sobre o valor de f , conforme proposto por este trabalho. O experimento apresentado na figura 6.3 é a mesma instância de simulador apresentada no primeiro experimento, mas agora ativando o controle automático do valor de f .

Como pode ser observado no gráfico, o nível de *tardiness* fica dentro do padrão estabelecido após o período de carga. O custo para tal é a redução do tempo de simulação que o sistema foi capaz de processar para o mesmo tempo real.

A figura 6.4 apresenta o avanço do tempo da simulação em relação ao tempo

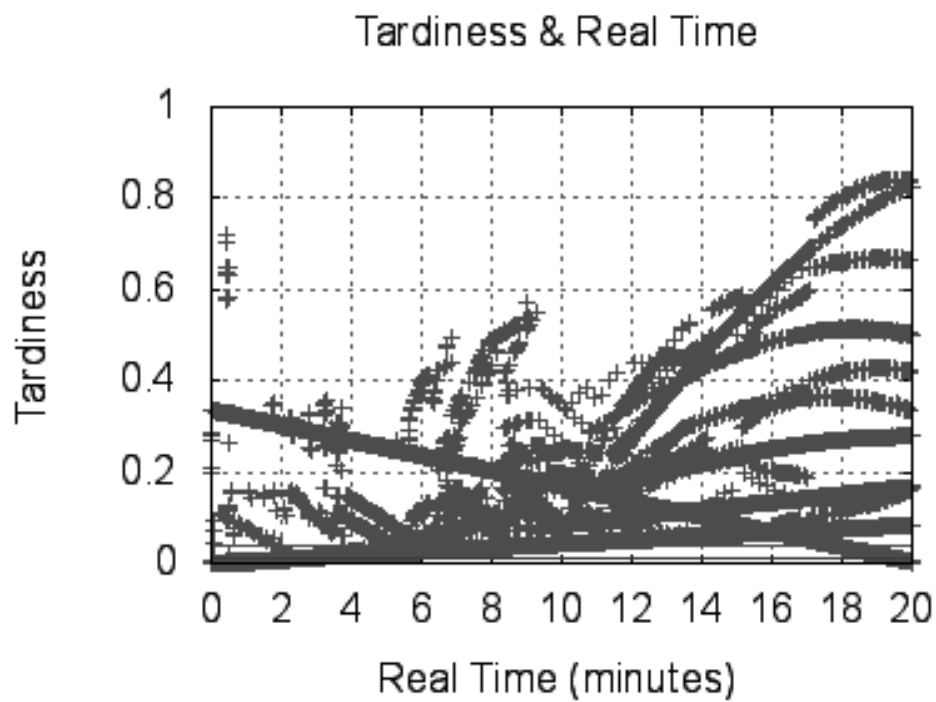


Figura 6.2: Tardiness - 10 agentes de simulação executando sem controle de f (120 seg. de passo)

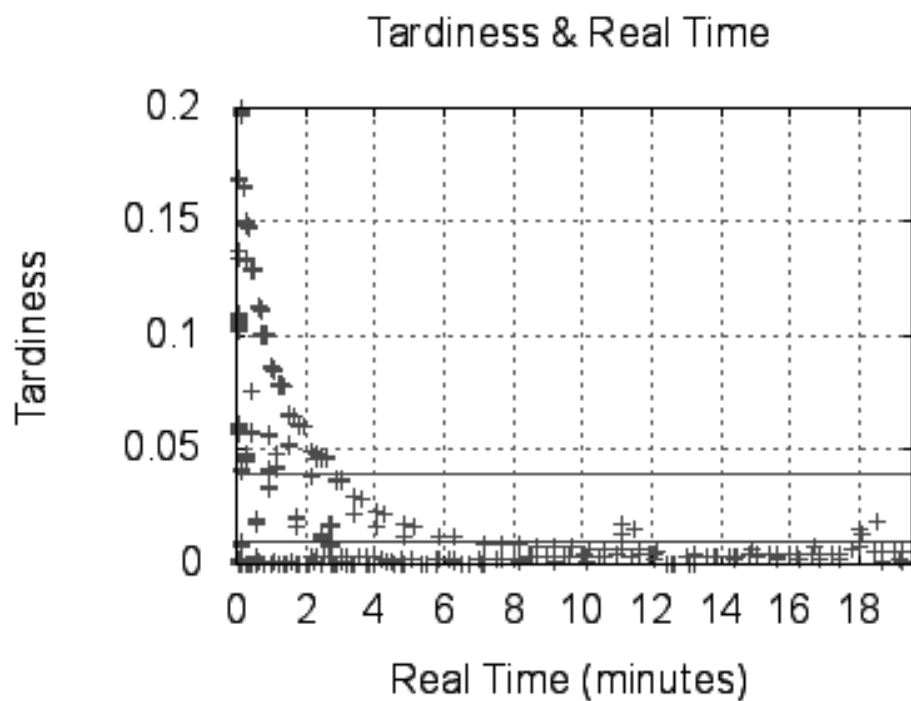


Figura 6.3: Tardiness - 10 agentes de simulação executando com controle de valor de f (10 seg. de passo)

real quando o mecanismo de controle esteve ativado. A simulação foi capaz de executar apenas 150 minutos de tempo de simulação, porém neste caso foi possível garantir uma execução na qual os agentes não excediam os níveis máximos de *tardiness* estipulados.

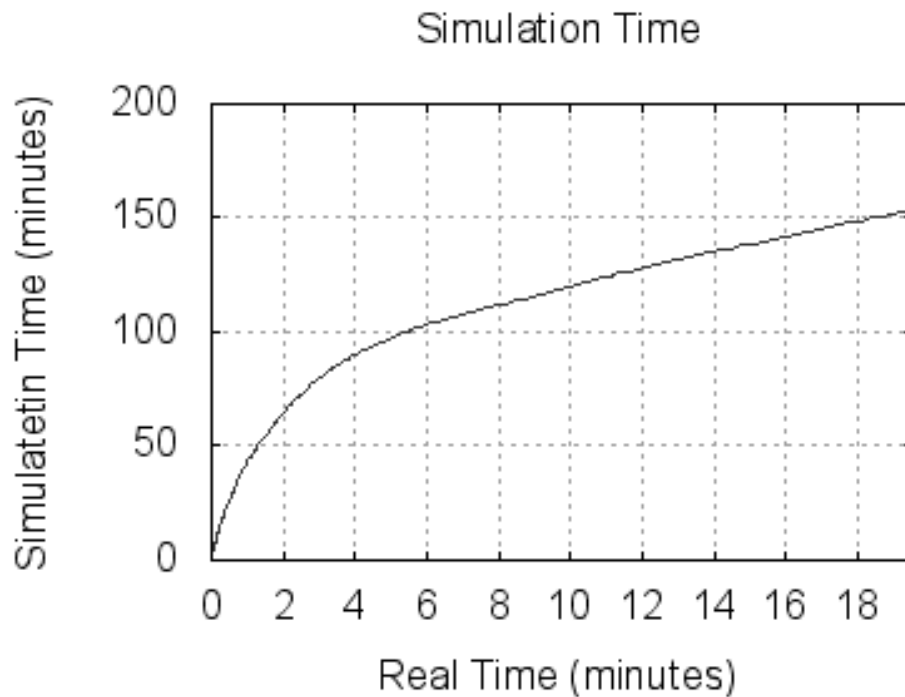


Figura 6.4: Avanço do Tempo de Simulação - MABS com 10 agentes de simulação e com controle de valor de f

A variação do valor de f ao longo do tempo real é apresentada na figura 6.5. O algoritmo de controle força a manutenção do nível de *tardiness* a um valor que não exceda o máximo estipulado, e depois procura um valor ótimo, variando com a carga do sistema em tempo de execução.

Outros experimentos foram executados para testar a abordagem proposta. A figura 6.6 apresenta uma simulação de 10 agentes utilizando um passo de tempo de simulação de 2 minutos e com controle de valor de f ativado. Outro exemplo é apresentado na figura 6.7, que apresenta a execução de uma simulação de 100 agentes com a duração do passo de 2 minutos e também com o controle de valor de f ativado.

É necessário destacar que nos exemplos apresentados nas figuras 6.3, 6.6 e 6.7, o nível de *tardiness* somente foi estabilizado abaixo do nível crítico após 5 minutos de simulação (tempo real). Esta situação indesejável pode ser evitada quando se inicia a simulação com um valor de f adequado aos recursos disponíveis. O ciclo de aquecimento pode ser utilizado para determinar este nível. Nos experimentos,

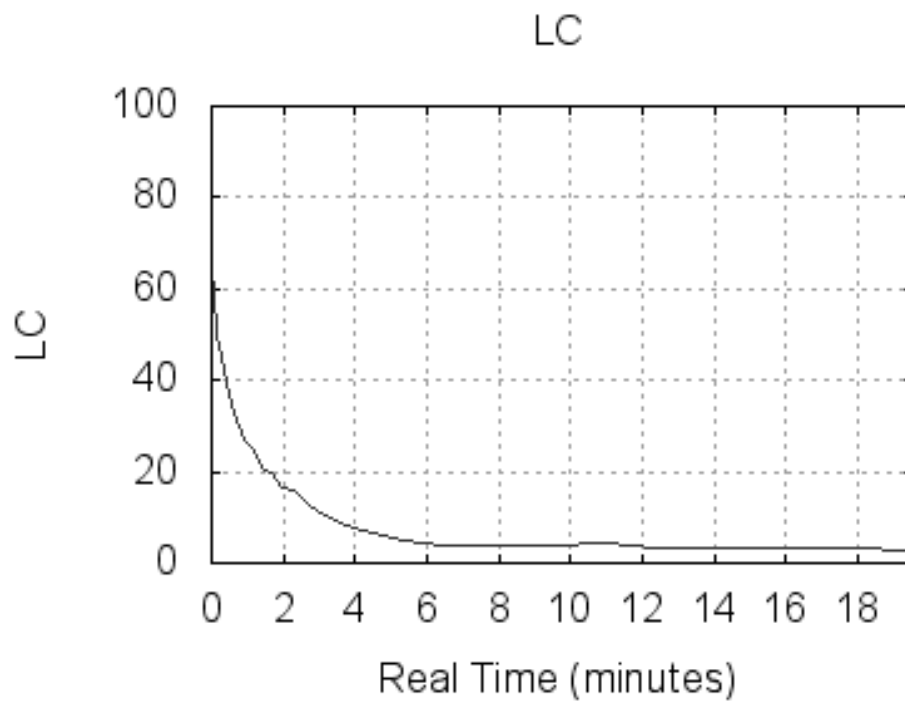


Figura 6.5: Variação do valor de f - MABS com 10 agentes de simulação e com controle de valor de f

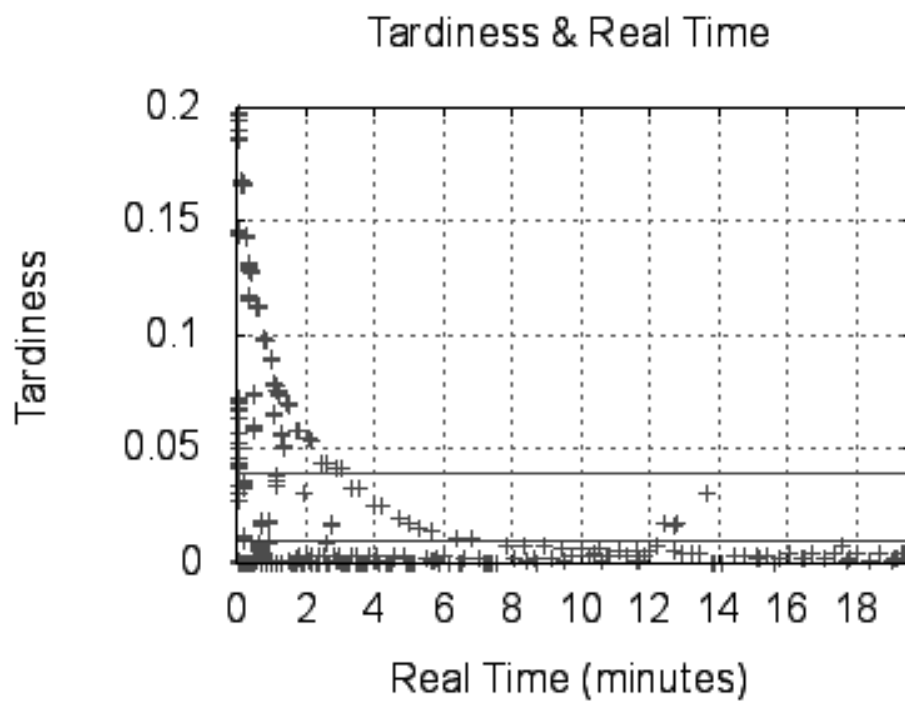


Figura 6.6: Tardiness em MABS executando com controle de valor de f (2 min de passo - 10 agentes)

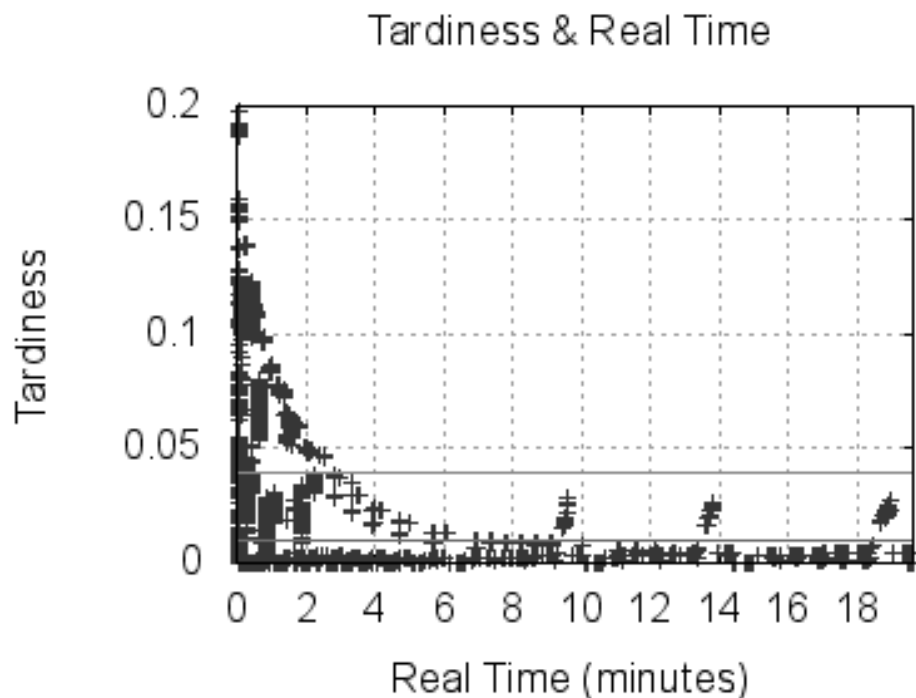


Figura 6.7: Tardiness em MABS executando com controle de valor de f (2 min de passo - 100 agentes)

optou-se por não usar o ciclo de aquecimento do MASP por fim didático, pois assim a ação do controle do valor de f é destacada.

6.2

Suporte a simulações georeferenciadas

A fase seguinte no desenvolvimento da solução foi destinada a responder a questão secundária abaixo:

QS5: *Obter uma solução que atenda o controle de atraso em tempo de simulação e que admita que os agentes variem os períodos de execução de comportamentos em tempo de simulação*

A resposta a questão acima permitiu o uso de simulações georeferenciadas. As modificações na abordagem original consistiram em alterar o algoritmo de controle, de forma a detectar e se adaptar as alterações dos períodos de execução. Os demais componentes foram alterados adequarem-se aos modificações inseridas.

6.2.1

Descrição do experimento

Para validar as modificações, foi executado o experimento descrito a seguir: um cenário com 500 agentes de simulação representando atores foi instanciado, com diferentes limites de velocidade e de aceleração para os grupos de agentes. Os valores, todos descritos no Sistema Internacional (SI), são apresentados na tabela 6.2.

Tabela 6.1: Cenário da Simulação

Tipo de Agente	Veloc Min	Veloc Max	Acel Max
<i>Avião de Caça</i>	150	300	2.0
<i>Helicoptero</i>	5	60	0.8
<i>Fragata</i>	4	20	0.3
<i>Submarino</i>	1	10	0.2
<i>Submarino (Nuclear)</i>	4	20	0.2

Para cada tipo de agente de simulação, 100 instâncias foram criadas em tempo de execução, com um comportamento de velocidade e rumo variáveis e aceleração estocástica, porém respeitando os limites estabelecidos inicialmente e que são coerentes com o domínio de guerra naval.

Para a simulação foram estabelecidos os seguintes parâmetros:

- Erro espacial máximo: 500 metros
- Máximo valor de tardiness = 10%
- Valor de tardiness para iniciar o controle = 2%
- $f_{(inicial)} = 5$

Os comportamentos associados ao movimento no espaço virtual adaptam seu loop de controle (ie. passo de execução) para atualizarem sua posição em saltos menores que o erro máximo.

O experimento gerou duas amostras, a de controle e a de dados para análise.

A amostra de controle foi executada sem interferência, mantendo $t_s = 5.t_r$ durante toda execução. A amostra de coleta de dados utilizou $f_{(inicial)} = 5$, porém com o controle do valor de f variando pela atuação do algoritmo. O MASP foi utilizado para o experimento, e seu mecanismo de registro de dados gerou a massa de dados em arquivos comma-separated files (.csv), que depois foram manipulados utilizando o software R (R Development Core Team, 2011).

As execuções foram realizadas no mesmo ambiente: um computador pessoal com processador i7, com 8GB de RAM DDR3. O sistema operacional usado foi o Opensuse 12.1 (64bit), ie. que é uma versão de Linux. Serviços como o de

impressão e de rede foram desabilitados fisicamente para minimizar interferências externas.

Os resultados são apresentados sob forma gráfica com comentários a seguir.

6.2.2

Avanço do tempo de simulação

O custo da abordagem é a redução do tempo de simulação executado: a amostra de controle, que usou $f = 5$, avançou 125 minutos de tempo de simulação em 25 minutos de tempo real de execução (fig. 6.8).

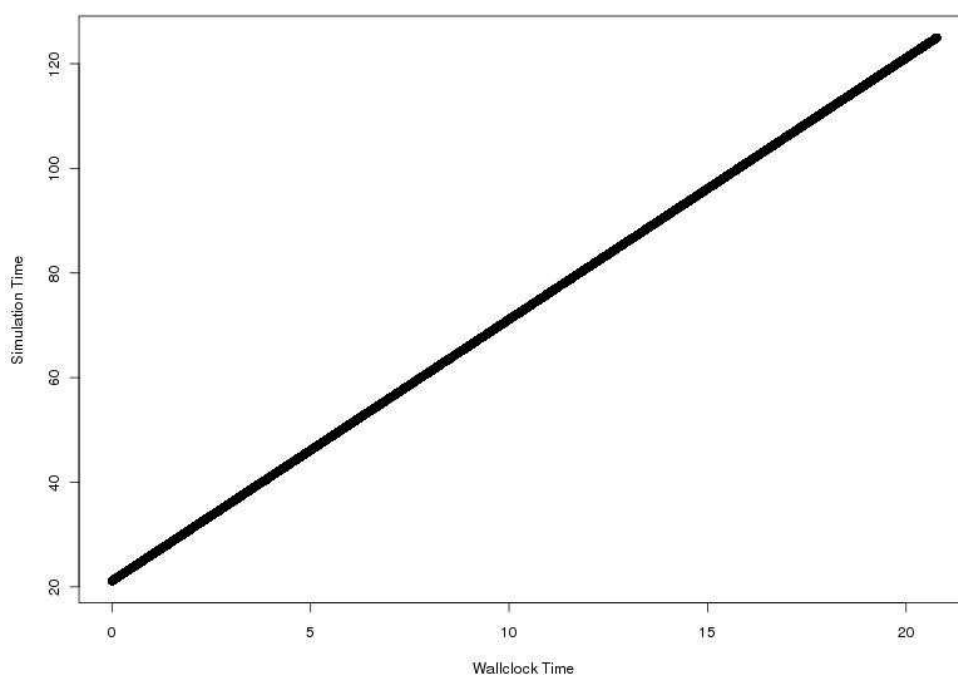


Figura 6.8: Amostra de controle – tempo de simulação x tempo real

Em contraste, a amostra que utilizou a abordagem foi capaz de executar somente 2,3 minutos de tempo de simulação no mesmo período, conforme apresentado na figura 6.9).

Esta diferença ocorre devido a redução automática do valor de f , que inicialmente era $f = 5$. O gráfico apresentado na figura 6.10 mostra que o valor de f , alterado pela ação do algoritmo de controle, decaiu rapidamente para valor próximo de 0,3, estabilizando próximo a este nível.

6.2.3

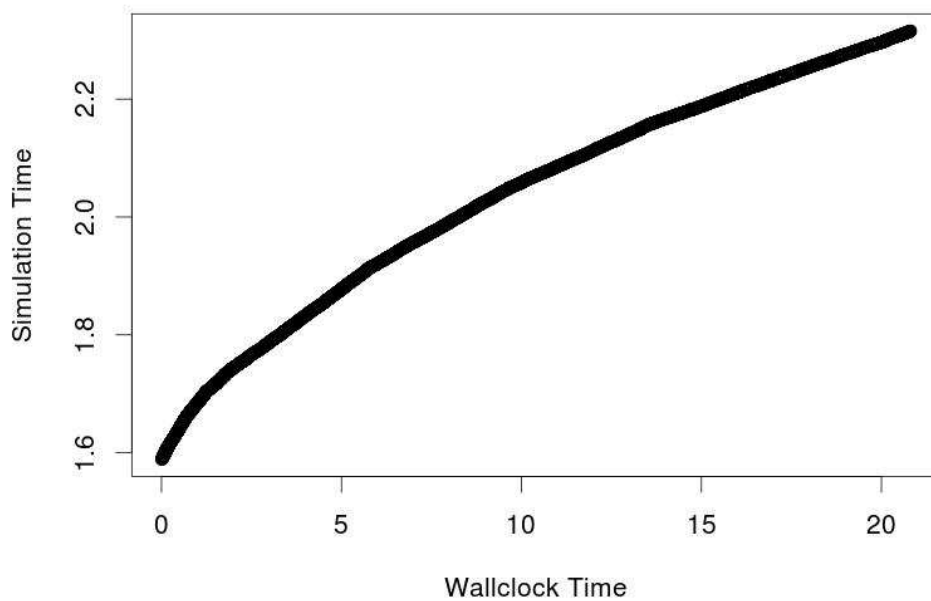


Figura 6.9: Amostra principal – tempo de simulação & tempo real

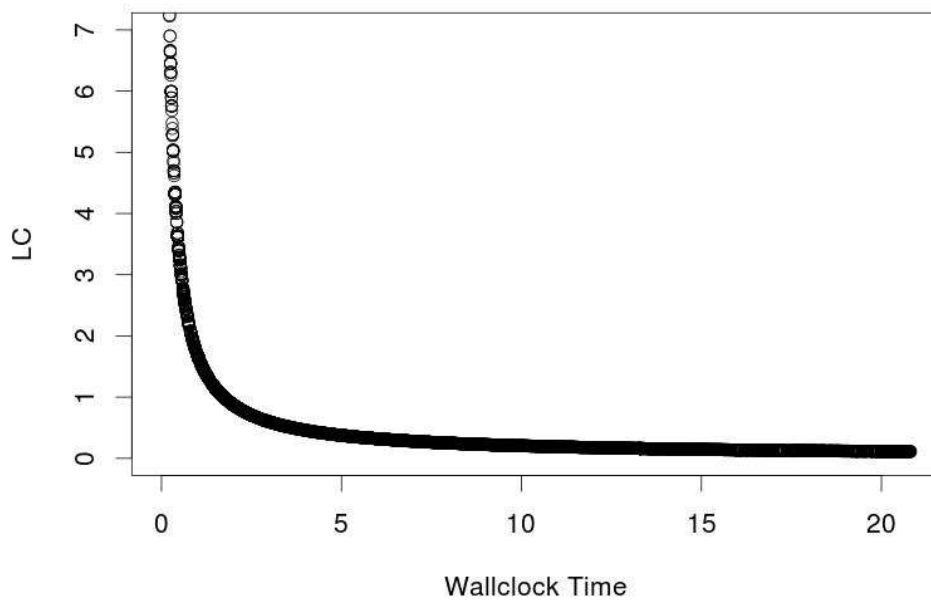


Figura 6.10: Amostra principal – valor de f x tempo real

Nível de tardiness

A massa de dados foi também verificada quanto ao controle do nível de *tardiness*, a fim de garantir que as modificações no algoritmo não haveria degradado

a abordagem para este item.

A amostra de controle é apresentada nos gráficos 6.11 e 6.12. O primeiro apresenta o histograma de *tardiness* e o segundo apresenta o comportamento do *tardiness* ao longo do tempo real, com o valor constante de $f = 5$.

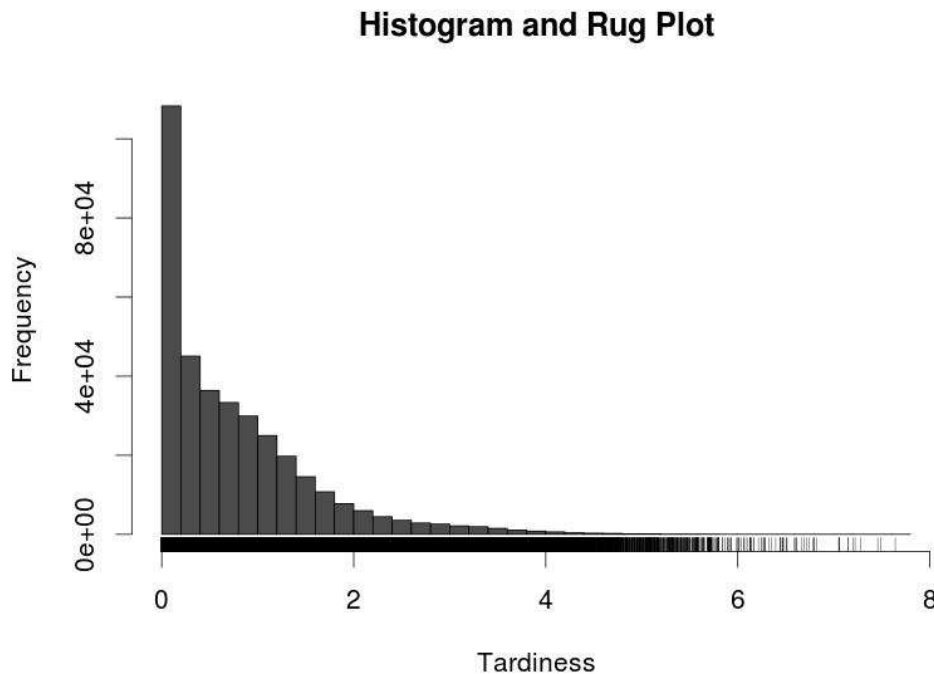


Figura 6.11: Amostra de controle – histograma de *tardiness*

Conforme observado no histograma, o valor de *tardiness*, sem o controle automático de f , varia aproximadamente de 0 a 6 em uma distribuição que graficamente é semelhante a exponencial. Adicionalmente o gráfico 6.12 permite observar que o *tardiness* não possui distribuição normal. É apresentado o *boxplot*, no qual os quartis são nitidamente diferentes do que se esperaria nesta distribuição. Esta informação concorda com o histograma. O gráfico também apresenta que os valores de *tardiness* ficam contidos em valores menores que 2,3, exceto pela ocorrência de *outlines*.

Os próximos dois gráficos (fig. 6.13 e 6.14) apresentam a amostra principal, como mecanismo de controle automático de f ativado. O primeiro gráfico apresenta um histograma de comportamento exponencial similar ao da amostra principal, contudo os valores medidos são muito menores, com a maioria abaixo de 0,01 de *tardiness*, ou seja, um atraso inferior a 1%.

Outra observação é que o valor de *tardiness* reduziu ao longo da execução até estabilizar, em contraste com o observadona amostra de controle. Isto ocorre devido a variação de f , que estabiliza-se após atingir nível adequado. Este comportamento

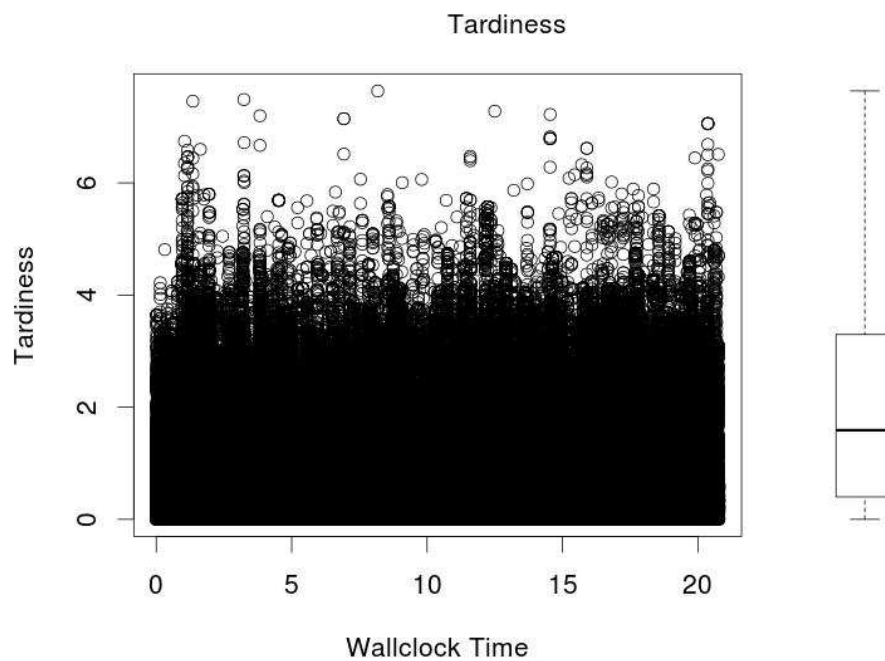


Figura 6.12: Amostra de controle – tardiness & tempo real / Boxplot

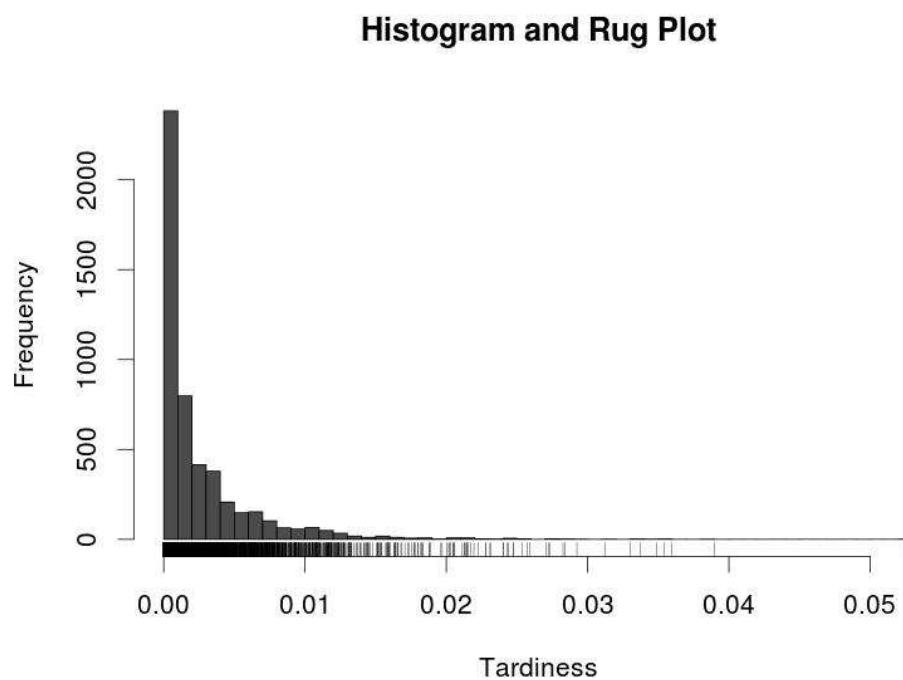


Figura 6.13: Amostra principal – histograma de tardiness

poderia ser reduzido com uso de ciclo de aquecimento para a simulação a fim de avaliar qual o valor de f adequado para o início da simulação.

O *boxplot* da fig. 6.14 permite observar que os maiores valores de tardiness

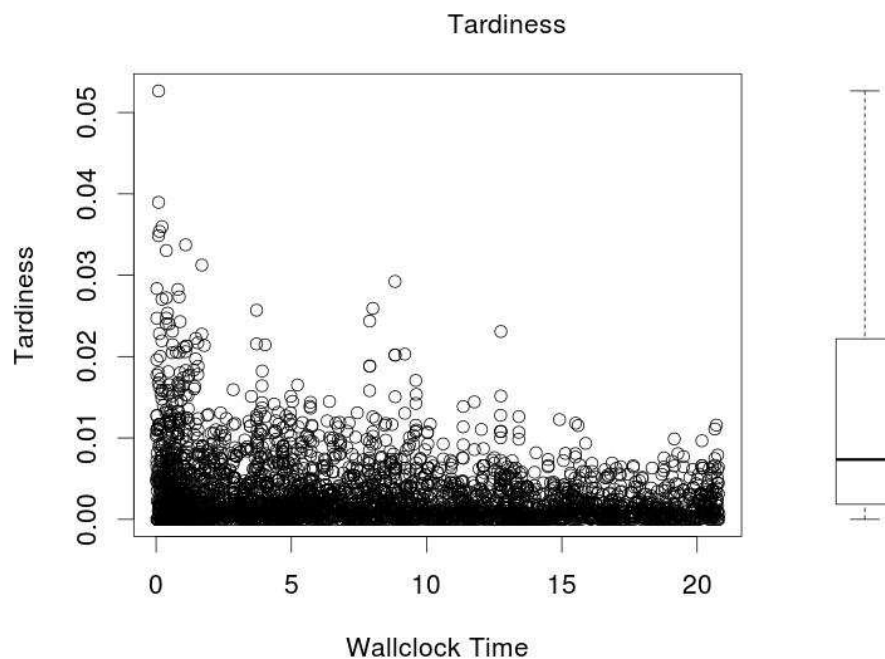


Figura 6.14: Amostra principal – tardiness & tempo de simulação / Boxplot

forma inferiores a 0,06, abaixo do parâmetro estabelecido de 0,1 (ie. 10%).

6.2.4

Erro espacial

As figuras 6.15 e 6.16 apresentam o erro espacial da amostragem de controle.

O histograma da amostra de controle apresenta erros superiores a 4.000 metros. O comportamento do erro espacial observado ao longo do tempo real apresenta a maioria dos valores abaixo de 1.500 metros. Destaca-se que aproximadamente 25% dos erros espaciais possuem valores entre 500 e 1.500 metros no espaço virtual, valores maiores que o erro máximo de 500 metros determinado no início da simulação.

Os valores observados em 6.15 e 6.16 comprometem a consistência da simulação, uma vez que, se o contexto espacial afeta o processo de decisão dos agentes de simulação, estes decidirão incorretamente em consequência do erro espacial.

A figura 6.17, obtida da amostra principal, mostra que o erro espacial foi controlado e mantido em valores entre 0 e 250 metros, abaixo do nível crítico do experimento.

Ocorreu concentração de erros em determinados valores, próximos a 20, 50 e 100 metros. Não foi possível determinar a causa da concentração, contudo em outros experimentos esta concentração se repetiu. Conquanto não afete a simulação

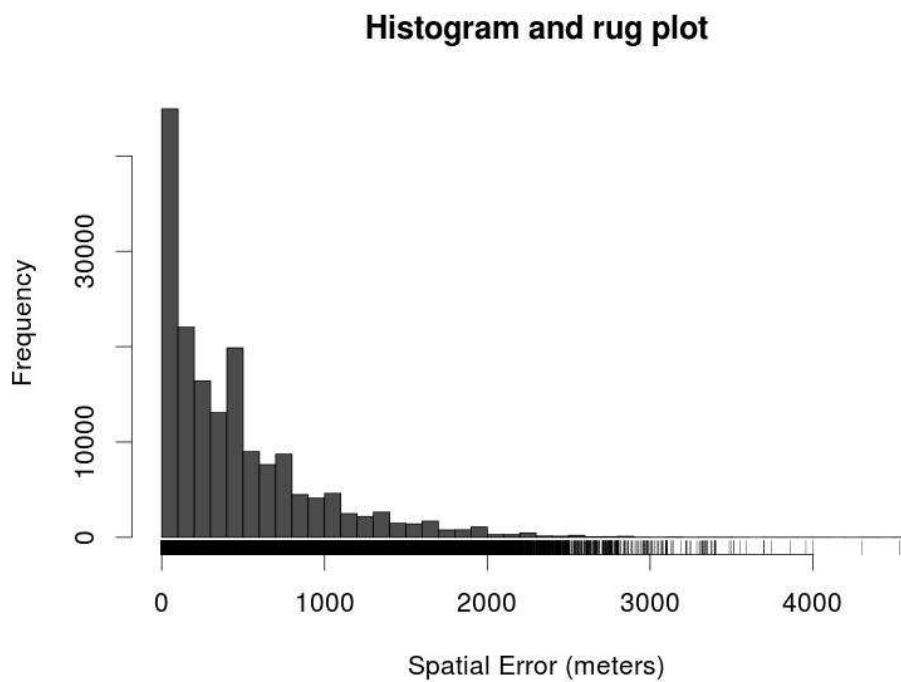


Figura 6.15: Amostra de controle – histograma do erro espacial

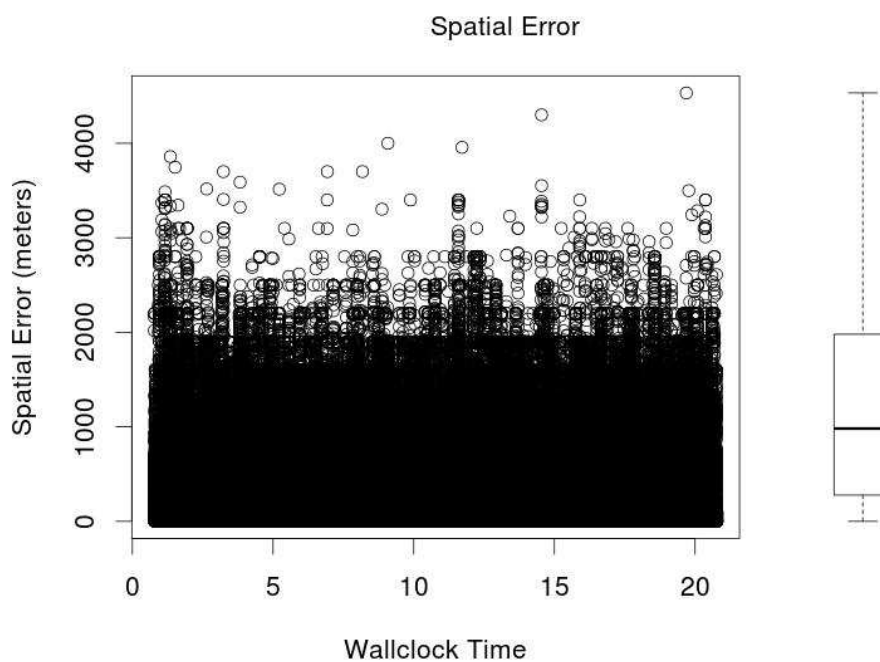


Figura 6.16: Amostra de controle – erro espacial & tempo real / Boxplot

por ocorrer abaixo do limite estabelecido, este é um fator que merece atenção em estudos futuros.

A figura 6.18 oferece outra visão sobre o erro espacial da amostra principal.

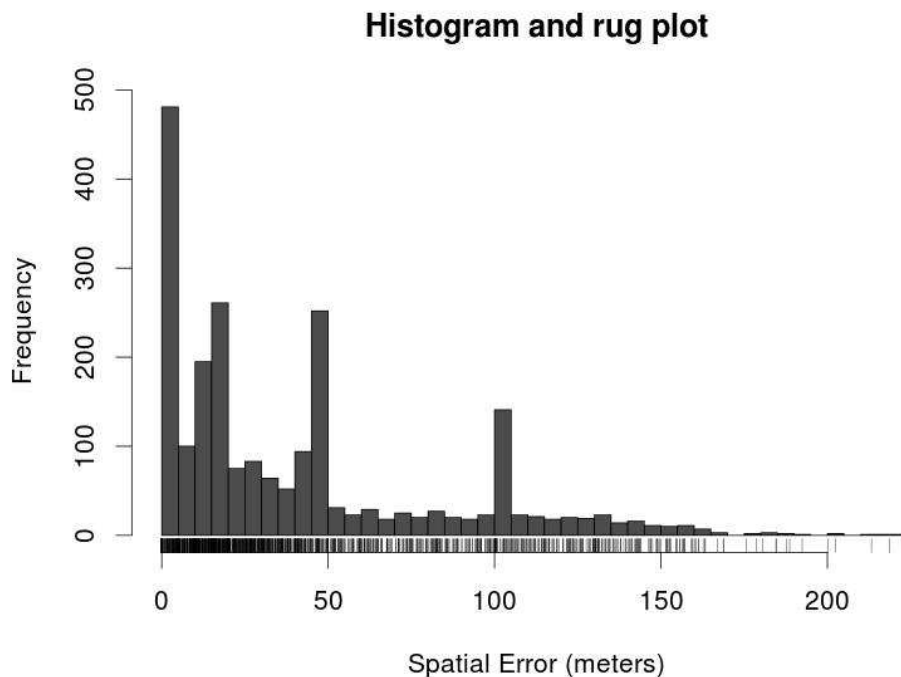


Figura 6.17: Amostra principal – histograma do erro espacial

O gráfico apresenta o *boxplot*, com *outlines* menores que 200 metros em valor absoluto. A distribuição das medidas pelos quartis foi considerada adequada aos objetivos da pesquisa. O erro espacial apresentou uma distribuição regular ao longo do tempo real de execução. Uma análise visual permite observar que a concentração de erros espaciais próximos a 100 e 50 metros, já discutida na figura 6.17, é distribuída ao longo de todo o tempo de execução da simulação.

Ambos os gráficos indicam a ocorrência de erro espacial, porém dentro dos limites estabelecidos (o limite foi informado eu diretiva pré-execução no MASP) .

6.3

Suporte a simulações dirigidas por eventos

A última parte no desenvolvimento da solução foi dedicada a obter a resposta par a questão secundária abaixo:

QS6: *Estender a solução para simulações que usem o formalismo DEVS, nas quais o controle do avanço do tempo é realizado ao próximo evento.*

Esta última resposta completou o conjunto de questões secundárias e permitiu responder a questão principal de pesquisa de forma completa, quando considerada em conjunto com as respostas anteriores:

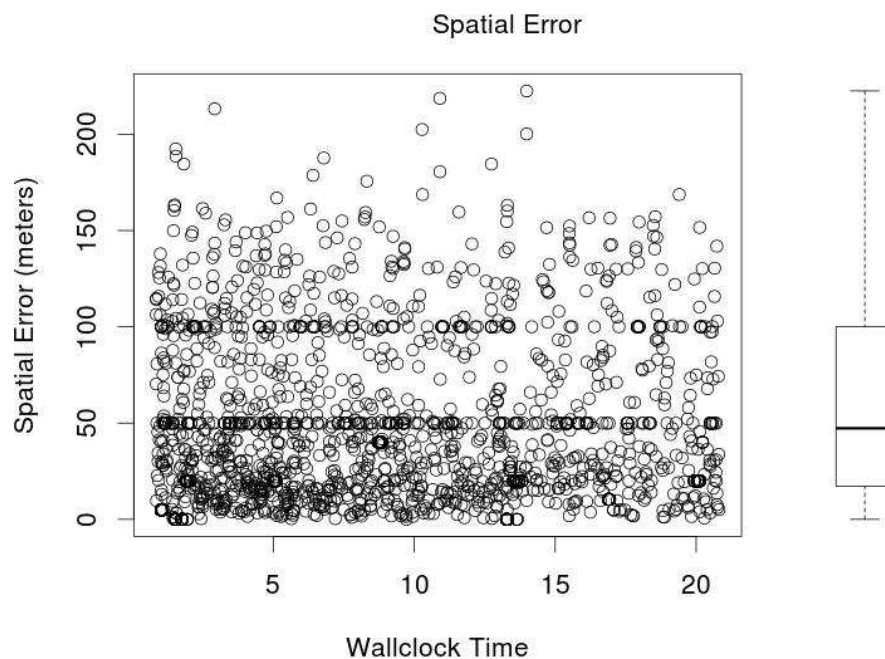


Figura 6.18: Amostra principal – erro espacial & tempo de simulação / Boxplot

QP: *Obter uma solução que permita a execução de simulações baseadas em sistemas multiagentes nas quais os atrasos medidos em tempo de simulação sejam controlados a níveis pré-estabelecidos e nas quais cada agente possua o controle de sua linha de execução.*

6.3.1

Descrição do experimento

Para testar as modificações realizadas na abordagem a fim de suportar simulações com avanço de tempo ao próximo evento foi conduzido um teste que utilizou o modelo conceitual apresentado na figura 6.19. O modelo de execução, como nos experimentos anteriores, foi instanciado no MASP.

O modelo contempla a existência de 300 agentes de simulação que se deslocam no espaço virtual, executando comportamentos de verificação de contexto (sonar) e movimento (kinematics). Para avaliação o modelo foi instanciado em dois modelos de execução:

- o primeiro utilizando avanço do tempo de simulação ao próximo evento; e
- o segundo outro avanço por passo de tempo.

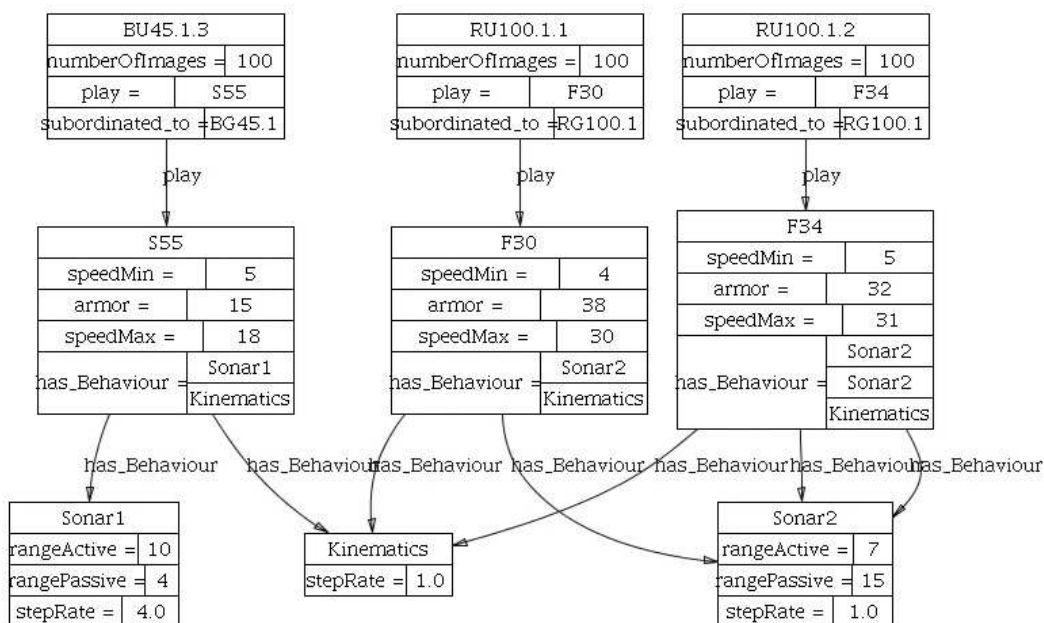


Figura 6.19: Modelo conceitual da simulação apresentado em diagrama de classes e indivíduos de ontologia

O avanço do tempo de simulação em ambos os modelos é gerenciado individualmente pelos agentes da simulação. Cada modelo de execução foi executado duas vezes, a primeira com o controle de *tardiness* proposto pela abordagem ativado e a segunda sem o controle.

Os parâmetros iniciais da simulação foram:

- valor inicial de $f = 10$;
- maior valor de *tardiness* previsto igual a 10%;
- *tardiness* mínima para iniciar controle do avanço igual a 1%;
- distancia mínima de detecção no espaço virtual = 1000 metros (erro espacial máximo 500 metros);
- período utilizados pelos agentes para enviar informação periódica de *tardiness* (vide diagrama de sequência 5.3) foi 1000 milissegundos; e
- tempo real de execução para cada simulação = 120 minutos.

Para cada execução de um comportamento de simulação os dados de *tardiness* foram registrados em log, sob a forma de arquivos csv. Posteriormente estes dados foram analisados utilizando o software R (R Development Core Team, 2011).

Não foram considerados os dados relativos aos comportamentos de controle, pois estes não pertencem ao modelo que representa o sistema real simulado, mas sim à infraestrutura de apoio da engine.

Os resultados são apresentados a seguir.

6.3.2

Controle de tardiness

O resumo dos resultados coletados sobre este parâmetro são apresentados na tabela 6.2.

Tabela 6.2: Tardiness

Avanço de Tempo	Ocorrências	Ocorrências > 0, 1 (%)	Ocorrências > 0, 3 (%)
<i>Próximo Evento</i>	435.910	3.424 (0,78%)	236 (0,054%)
<i>Avanço por passo</i>	414.896	2.605 (0,62%)	70 (0,016%)
<i>Próximo Evento*</i>	1.272.127	68.947 (5,41%)	28.017 (2,20%)
<i>Avanço por passo*</i>	1.271.636	69.447 (5,46%)	28.874 (2,27%)

* controle automático do valor de f desabilitado

As duas primeiras linhas apresentam os resultados obtidos com o controle de tempo de simulação ativado, para simulação utilizando avanço de tempo de simulação ao próximo evento e por passo de tempo respectivamente.

O número semelhante de execuções no período indica que o valor médio de f , calculado e aplicado ao sistema dinamicamente, foi similar em ambos os casos.

Considerando o nível de aceitabilidade de *tardiness* estabelecido ($< 0, 1$), aproximadamente 99,2% das ocorrências foram controladas no primeiro caso e 99,4% no segundo. Se considerarmos um nível maior ($< 0, 3$), o nível de ocorrências controladas ultrapassa 99,9%.

Sem o controle ativado, conforme se observa nas duas últimas linhas da tabela, o nível de ocorrências controladas é menor que 95%, e a queda não é tão brusca se considerarmos o valor de 0,3 como limite.

Através de uma análise empírica dos resultados apresentados podemos afirmar que a abordagem tem sucesso no controle do *tardiness*.

Para apoiar esta análise, apresentamos os gráficos de histograma de *tardiness* para avanço de tempo de simulação ao próximo evento, sendo que a figura 6.20 apresenta a situação com controle ativado e a figura 6.21 sem este controle.

É importante salientar que estes gráficos e os números apresentados não podem ser utilizados de forma absoluta, pois variações nos parâmetros iniciais, tais como o valor de *tardiness* limite, erro espacial ou valor inicial de f , influenciam diretamente nos resultados obtidos. Outros parâmetros, de ajuste do algoritmo de controle, também tem influência nos resultados.

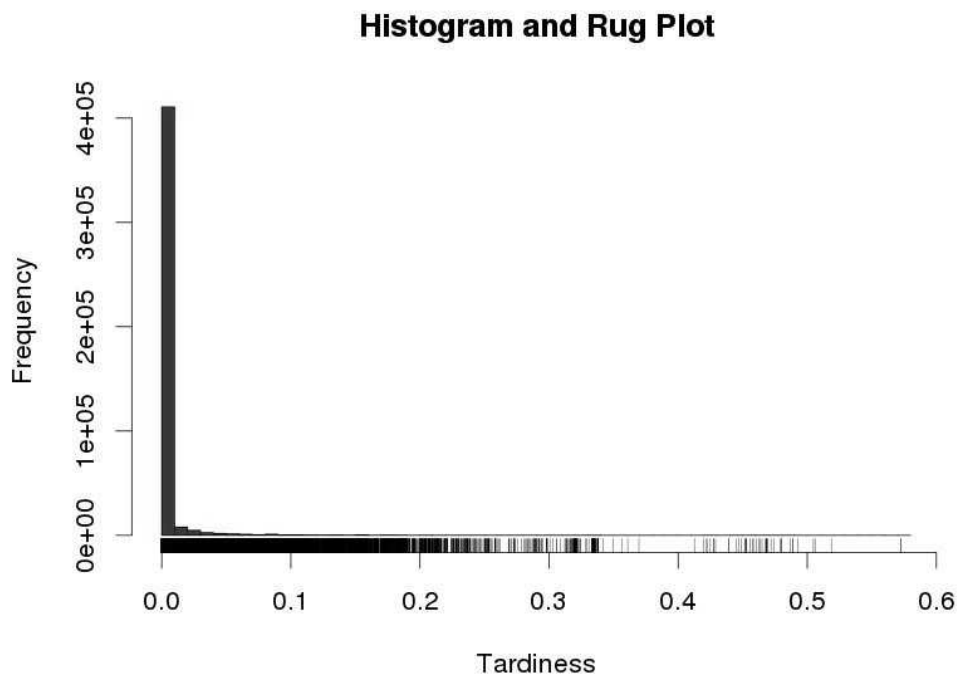


Figura 6.20: Histograma de tardiness - Avanço ao Próximo Evento

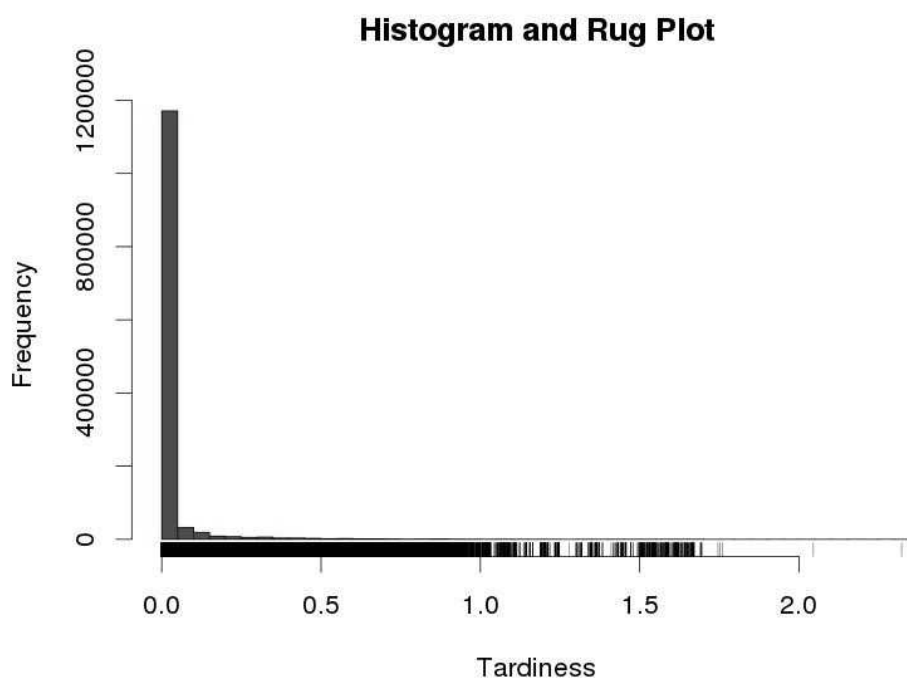


Figura 6.21: Histograma de tardiness - Avanço ao Próximo Evento (sem controle)

6.4

Considerações sobre generalização da solução para outras tecnologias

Como explicitado em 1.1, o uso de Java foi uma restrição a pesquisa. Esta restrição deve-se a diversos fatores, tais como: existência de ferramental disponível em código aberto, que facilita a pesquisa; a ampla base instalada em Java; e disponibilidade de mão-de-obra. Contudo, ao longo do trabalho ocorreram questionamentos sobre a aplicabilidade da solução em outras tecnologias.

A discussão a seguir é conceitual, uma vez que não foram realizados experimentos para apoiar o posicionamento com dados:

Qualquer linguagem de programação, seja interpretada ou compilada e dependente de plataforma, executa sobre uma arquitetura de camadas. O acesso do interpretador ou do programa binário ao processador se faz através do sistema operacional (SO), que gerencia o uso da CPU, da memória e dos demais dispositivos. O SO realiza o agendamento preemptivo de *thread*, e pode alterar os estados dos processos associados. Este cenário é responsável pela existência de *tardiness* em qualquer programa em execução, sejam simuladores ou não. Java apresenta particularidades que ampliam estas questões, relacionadas a sua JRE, ao GC e a sua gerência de *thread* Java e o mapeamento para *thread* de SO.

A opinião do autor é que a solução se aplica às demais tecnologias, contudo é necessário instanciar a mesma para apoiar essa posição por medições e verificar se a relação custo/benefício seria adequada.

6.5

Conclusão do capítulo

Este capítulo apresentou os resultados de experimentos quantitativos realizados para validar a abordagem apresentada no capítulo 5. Cada um destes experimentos gerou uma massa de dados grande, que permitiu realizar a avaliação gráfica dos resultados.

Pode-se observar que a abordagem não conseguiu excluir a existência de *outlines*, o que fica claro na análise da tabela 6.2. Conquanto isto não seja desejável, a existência destes envolve uma relação de custo-benefício: eles podem ser reduzidos com o estabelecimento de um nível menor para acionamento do controle de f , contudo a consequência será a redução do valor médio de f durante a execução e um menor avanço do tempo de simulação. Esta decisão cabe aos projetistas do modelo de execução ao estabelecer os parâmetros de corte para a abordagem. Um exemplo é o ajuste das diretivas de pré-execução no arquivo de propriedades do MASP (vide A.4.3).

Os experimentos não consideraram a interação entre os agentes da simulação, característica comum em simulações baseadas em MAS. Esta decisão foi assumida para reduzir o ruído do sistema, gerado por variáveis não controladas. Contudo, é

importante ressaltar que estas mensagens são implementadas como comportamentos imediatos, e o *tardiness* que estes causem poderá ser medido sobre os demais, sobre os quais a abordagem atua. Desta forma, o atraso decorrente destes é controlado indiretamente.