



**SNPTEE
SEMINÁRIO NACIONAL
DE PRODUÇÃO E
TRANSMISSÃO DE
ENERGIA ELÉTRICA**

GAT - 06
14 a 17 Outubro de 2007
Rio de Janeiro - RJ

GRUPO IV

GRUPO DE ESTUDO DE ANÁLISE E TÉCNICAS DE SISTEMAS DE POTÊNCIA - GAT

INTEGRAÇÃO ENTRE OS PROGRAMAS ATP E MATLAB/SIMULINK EM TEMPO DE EXECUÇÃO APLICÁVEL A ESTUDOS DE TRANSITÓRIOS ELETROMAGNÉTICOS DEVIDOS À MANOBRA DE EQUIPAMENTOS

Henildo Medeiros de Barros*

ONS – Operador Nacional do Sistema Elétrico

Carlos F. Teodósio Soares

Programa de Engenharia Elétrica - COPPE/UFRJ

Fábio Domingues de Jesus

ENELTEC – Engenharia Elétrica e Tecnologia

RESUMO

Este trabalho descreve o esforço empreendido e os resultados auferidos no desenvolvimento de um ambiente de simulação integrado no qual estão presentes as funcionalidades de ferramentas padrão empregadas por engenheiros e especialistas em estudos de transitórios eletromagnéticos resultantes de manobras em equipamentos: Os programas ATP (*Alternative Transients Program*) e MATLAB/Simulink.

O principal motivador dessa integração é a viabilização do uso seqüencial ou simultâneo do motor de processamento/solução de cada ferramenta no decorrer de uma mesma simulação. Tal faculdade possibilita, numa primeira aplicação, o uso da moderna interface gráfica do MATLAB/Simulink para apresentação e processamento de resultados durante a simulação. Em modo mais sofisticado, a integração visa também à utilização conjunta das bibliotecas de modelos disponíveis em ambas as ferramentas. Tais bibliotecas são caracterizadas, no caso do ATP, pela robustez e versatilidade e são caracterizadas, no caso do MATLAB/Simulink, pela especialização e facilidade de programação.

Inicialmente, a exemplo de trabalhos similares disponíveis na literatura, foi empregado o mecanismo de integração padrão disponível no MATLAB/Simulink – a biblioteca *MATLAB Engine*. Apesar de permitir o controle de execução do Matlab por meio de um programa externo, esse mecanismo mostrou-se mais adequado às tarefas de pré e pós-processamento de resultados devido ao seu baixo desempenho computacional intrínseco.

Posteriormente, foram investigadas diversas alternativas de integração de diferentes processos computacionais em ambiente MS-Windows, visando à identificação do mecanismo cujo requisito de desempenho computacional fosse o mais adequado ao uso combinado das ferramentas ATP e MATLAB/Simulink para a realização de estudos de transitórios eletromagnéticos.

A disponibilidade de uma interface para a integração de dois ambientes de simulação distintos e complementares – o ATP e o MATLAB/Simulink –, adequados ao uso nos estudos de transitórios eletromagnéticos, possibilita ao ONS (Operador Nacional do Sistema Elétrico) a utilização de um ferramental de simulação, no qual estão incorporadas as melhores funcionalidades de cada uma dessas duas ferramentas. A interface desenvolvida pelo especialista em engenharia de sistemas computacionais, apesar da complexidade associada ao controle de funções ao nível de sistema operacional, foi realizada de forma a ser transparente ao usuário final, ou seja, o especialista em estudos de transitórios eletromagnéticos.

PALAVRAS-CHAVE

Estudos de Transitórios Eletromagnéticos, ATP, MATLAB/Simulink, Ambiente de Simulação Integrado.

1.0 - INTRODUÇÃO

O programa ATP é uma ferramenta computacional largamente utilizada na simulação de transitórios eletromagnéticos em sistemas de potência. Apresenta-se como ferramenta de uso livre que é adotada, sem restrições, pelas empresas do setor elétrico brasileiro e pelo ONS devido a sua robustez e generalidade. O ambiente MATLAB/Simulink é uma ferramenta computacional concebida para modelagem e simulação de sistemas dinâmicos, a qual oferece maior potencial para representação de equipamentos com eletrônica de potência e dispositivos de controle e processamento de sinais. Apresenta-se como ferramenta comercial de largo espectro de utilização, devido à facilidade de uso do seu ambiente de programação e à existência de inúmeras bibliotecas – *Toolboxes* – especializadas.

Modernamente, em vista da consolidação de algumas ferramentas de simulação como padrão aplicável a estudos de transitórios eletromagnéticos no domínio do tempo – ATP, PSCAD/EMTDC e o EMTP-RV – e de ferramentas matemáticas de simulação de sistemas dinâmicos e discretos no tempo – MATLAB/Simulink –, há uma tendência natural à integração do primeiro grupo de programas com o segundo grupo, visando, principalmente, a fazer uso das melhores características de cada uma das ferramentas numa mesma simulação. Nessa vertente, o suporte a mecanismos de comunicação entre processos (*Interprocess Communication* – IPC) nos diversos sistemas operacionais, bem como a disponibilidade de arquiteturas computacionais de baixo custo, dotadas de múltiplos processadores, provê os subsídios necessários à integração de processos computacionais identificados com cada uma dessas ferramentas não somente de forma seqüencial, mas também em modo simultâneo durante o tempo de execução da simulação. Nesse último caso, além de as ferramentas de simulação atenderem a determinados protocolos de interface, é necessário também que seja garantida a sincronização entre os processos identificados com cada uma das mesmas.

De fato, tanto o PSCAD/EMTDC quanto o EMTP-RV já contam, em maior ou menor grau, com tal funcionalidade em seu ambiente de simulação. No PSCAD/EMTDC, o usuário invoca uma interface FORTRAN, especialmente desenvolvida para comunicação *on-line* como ambiente MATLAB/Simulink, através de ícone disponível em sua interface gráfica PSCAD. Nesse caso, a integração é realizada de forma transparente ao usuário e requer um mínimo de esforço do mesmo. No EMTP-RV, a interface é realizada por meio de chamadas diretas a bibliotecas do tipo DLL (*Dynamic Link Library*) geradas por *Toolboxes* especializadas do ambiente MATLAB/Simulink. Nesse último caso, a interação entre as ferramentas dá-se apenas no instante de geração do código executável, não havendo interação durante o tempo de execução da simulação, como no primeiro caso.

O ATP, pela filosofia adotada em sua distribuição, demanda, entretanto, que o próprio usuário seja o provedor das rotinas específicas para a comunicação do processo computacional a ele associado com os processos associados às ferramentas complementares. Obviamente, essa é uma tarefa que extrapola a área de conhecimento de um especialista em estudos de transitórios eletromagnéticos, o que representa um primeiro obstáculo a ser superado, visando à integração do ATP a outras ferramentas. A despeito dessa desvantagem, o ATP preenche um requisito básico, o qual permite a sua interação com outras ferramentas em tempo de execução, que é justamente a existência de interface própria para a inclusão de modelos externos definidos pelo usuário.

2.0 - HISTÓRICO DE INTEGRAÇÃO COM O ATP

O programa ATP dispõe de duas linguagens especializadas denominadas TACS e MODELS, com funções pré-definidas que facultam o desenvolvimento de modelos complexos de equipamentos e/ou sistemas de controle. Contudo, a ausência de uma interface homem-máquina moderna, a inexistência de bibliotecas de modelos específicas e a necessidade de conhecimento especializado fazem do uso dessas linguagens uma tarefa desestimulante para uma representação mais completa e aprimorada da rede sob estudo. Por conta dessa dificuldade e da necessidade de atualização do código frente às demais ferramentas comerciais, o desenvolvedor do ATP dotou o programa de algumas funcionalidades que permitem a incorporação de modelos externos ao ATP, descritos em linguagem de alto nível (FORTRAN, C e C++), e viabilizam, com maior grau de técnica e criatividade, o controle de processos computacionais externos ao ambiente de simulação do próprio ATP.

As primeiras iniciativas de integração entre os programas ATP e MATLAB/Simulink exploraram a automação de modificações da base de dados do ATP, através de scripts de comandos em ambiente MATLAB adequados à programação dos casos correspondentes a uma seqüência de simulações. Nesse caso específico, a interação ocorre apenas no instante de execução do ATP, através de uma chamada ao DOS *shell* – comandos DOS ou ! (*shell scape*) –, via janela de comandos do MATLAB. A referência [1] apresenta um exemplo de interação dessa natureza, correspondente à avaliação de desempenho de relés de proteção. Entretanto, a literatura técnica internacional contempla poucos artigos que detalham a utilização direta de modelos externos descritos em linguagem padrão do ambiente MATLAB/Simulink, em programas de simulação de transitórios eletromagnéticos. No caso do programa ATP, existe um registro de aplicação razoavelmente bem sucedida com uso simultâneo dos motores de processamento/solução dessa ferramenta integrada com o MATLAB/Simulink [2]. Nesse trabalho, o motivador maior de tal aplicação foi a representação da rede elétrica e seus sistemas de proteção digital em malha fechada, para a análise do desempenho dos respectivos relés.

Em [3], os autores apresentam uma simulação na qual o sistema de potência e os relés de proteção do tipo digital foram representados, respectivamente, no ATP e no ambiente gráfico do MATLAB/Simulink. Nessa simulação, a solução é realizada pelos dois motores de processamento simultaneamente, através de uma estratégia denominada *buffer de interação de processos*. Em outras palavras, os autores sugerem a utilização de um bloco de memória, reservada para o armazenamento temporário dos dados intercambiados entre as duas ferramentas durante o intervalo de simulação. Na seqüência de solução, isto é, a cada passo de integração, os sinais de entrada dos relés – tensões e correntes no secundário dos transformadores de instrumento – são computados pelo ATP e disponibilizados na área de intercâmbio. No mesmo passo de integração ou em um múltiplo desse, o MATLAB/Simulink avalia a resposta do relé de proteção, a partir dos dados disponibilizados na área de intercâmbio, e atualiza o sinal de disparo do mesmo, disponibilizando-o também nessa área. No passo de integração seguinte, o ATP atualiza o estado das chaves de acordo com o sinal de disparo gerado no passo anterior e calcula a nova solução da rede. De fato, os autores utilizam, nessa aplicação, o mecanismo IPC conhecido como *Shared Memory*.

Posteriormente, em trabalho desse mesmo grupo [2], é feita uma reavaliação da metodologia utilizada no trabalho anterior, tendo em vista o baixo desempenho computacional da solução integrada. Nessa oportunidade, os autores indicam como melhor opção à modelagem dos relés via linguagem gráfica do Simulink, a sua representação por modelo externo ao ATP, descrito em linguagem de alto nível (C++). Obviamente, essa solução representa um retrocesso, pois se trata da renúncia às facilidades computacionais de representação e modelagem oferecidas pelo ambiente MATLAB/Simulink.

3.0 - INCORPORAÇÃO DE MODELOS EXTERNOS AO ATP

3.1 Interface de modelos descritos em linguagem de alto nível

No uso tradicional do programa ATP, o especialista descreve a lógica representativa de sistemas de monitoramento/controle no domínio contínuo ou discreto do tempo através das instruções existentes nas linguagens proprietárias TACS e MODELS. Tal procedimento o obriga a converter o diagrama funcional do controle – ou o código fonte de rotina executável em um microprocessador – para uma dessas linguagens, fazendo as devidas adaptações e simplificações em função dos blocos e instruções disponíveis nas mesmas. Por essa razão, o uso de tais linguagens é mais apropriado à representação de sistemas de monitoramento/controle relativamente simples a serem utilizados no processamento matemático dos valores instantâneos das grandezas elétricas da rede e na operação de chaves ou fontes controladas.

3.2 Interface MODELS

A utilização da interface externa via linguagem MODELS evita que o usuário tenha de converter um código já existente em linguagem de alto nível (FORTRAN, C ou C++) para uma das linguagens proprietárias do ATP, o que poderia tornar-se inviável a depender da complexidade do que se quer representar. A interface externa disponível na MODELS possibilita a programação da rotina necessária a compatibilização das grandezas que serão intercambiáveis entre o ATP e o código externo. Por outro lado, é exigido como requisito básico que o código fonte externo – *foreign function* – possa ser tratado pelo mesmo compilador da versão disponível do ATP.

Essa interface é especialmente útil quando se necessita realizar a interação do ATP com controladores e sistemas de proteção desenvolvidos pelo próprio usuário, além do que possibilita um melhor desempenho computacional da simulação visto que o código externo passa a ser embarcado (*embeeded*). Numa aplicação mais avançada, permite, inclusive, o controle da execução de um outro programa ou ambiente de simulação que seja apto a realizar operações tão complexas quanto às realizadas pelo ATP. Tal tipo de aplicação será explorado nos itens seguintes deste artigo.

3.3 Elemento Tipo 94 (TYPE-94)

A utilização adequada desse componente, contemplando as suas três variantes, abre uma possibilidade de uso integrado do ATP com o MATLAB/Simulink, que faculta ao usuário a escolha da parcela de solução a ser executada por cada processo associado. Mais especificamente, o uso elemento Tipo 94 em conjunto com rotinas externas desenvolvidas pelo usuário, possibilita que outro processo correspondente a programa ou ambiente de simulação distinto se integre ao processo correspondente ao ATP no intervalo de tempo de simulação, de forma executar uma parcela do cálculo das tensões nodais empregando seus próprios modelos matemáticos e algoritmos de solução. Tais rotinas externas, acessíveis a partir da interface MODELS, são especializadas para o controle e gerenciamento da comunicação entre os referidos processos.

Esse elemento corresponde à chave para a utilização integral das funcionalidades existentes nas ferramentas ATP e MATLAB/Simulink, ampliando a perspectiva de uso da interface externa via linguagem MODELS no aprimoramento de aspectos de pré-processamento de dados e pós-processamento de resultados para uma evolução na quantidade de opções disponíveis para a modelagem de equipamentos individuais ou porções da rede elétrica, bem como uso efetivo de arquiteturas multiprocessadas e ambiente de processamento distribuído.

Em última análise pode-se lançar mão da subdivisão implícita ao algoritmo de solução do ATP, adotada toda a vez que o processo associado a essa ferramenta trata de rede na qual está presente um ou mais componentes Tipo 94, para distribuir parte da solução entre o próprio ATP e o MATLAB/Simulink. Pode-se até mesmo utilizar o processo correspondente ao MATLAB/Simulink como um escalonador de diversos processos correspondentes a simulações de subsistemas em ATP integrantes de uma mesma rede.

4.0 - REPRESENTAÇÃO DE MODELOS EM LINGUAGEM DE ALTO NÍVEL VIA TOOLBOXES MATLAB/SIMULINK

A principal vantagem das ferramentas de modelagem disponibilizadas pelo pacote Matlab/Simulink é a grande facilidade com que o usuário final pode construir modelos de equipamentos, através da simples construção de diagramas de blocos em uma interface gráfica amigável. Uma vez que o modelo esteja pronto, emprega-se a ferramenta *Real-Time Workshop* (RTW), do próprio MATLAB/Simulink, para produzir um conjunto de funções, escritas em linguagem C, que descrevem o comportamento dinâmico do modelo em questão. Tais funções podem ser chamadas por uma *foreign function*, associada à interface MODELS de uma simulação ATP. Dessa forma, o modelo construído com o MATLAB/Simulink pode ser utilizado em uma simulação ATP.

4.1 Bibliotecas do Ambiente Simulink

O MATLAB/Simulink disponibiliza uma variedade de bibliotecas – conhecidas como *Toolboxes* –, contendo blocos que desempenham determinadas operações, que podem ser utilizados na construção dos modelos de equipamentos. Por exemplo, a biblioteca *Math Operands* contém blocos que realizam operações matemáticas fundamentais com sinais, como adição, multiplicação e a aplicação de funções matemáticas – como exponenciais e trigonométricas.

Para que o RTW possa produzir um código em linguagem C referente a um determinado modelo, é necessário que cada bloco tenha, associado a ele, um arquivo com o código referente a apenas o modelo do próprio bloco separadamente. Assim, o que efetivamente o RTW realiza é combinar os trechos de código associados a cada um dos blocos do modelo, de acordo com as interconexões dos blocos fornecida pelo usuário.

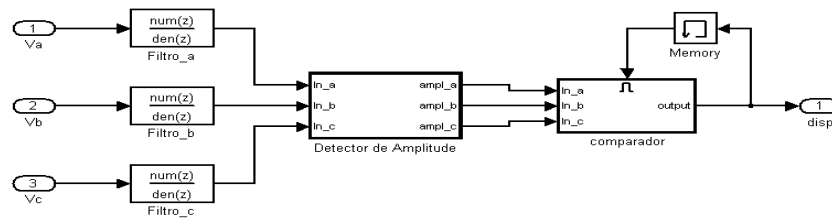
Infelizmente, nem todos os blocos disponibilizados pelas bibliotecas do MATLAB/Simulink possuem, associado a eles, um trecho de código em linguagem C. Um exemplo é justamente a biblioteca *SimPower Systems*, que contém modelos de componentes e equipamentos empregados na simulação de sistemas de potência. Devido a essa limitação, o RTW não pode produzir um código em linguagem C que descreva um modelo que contenha blocos da biblioteca *SimPower Systems*, o que é uma grande desvantagem para os usuários interessados na simulação de sistemas elétricos de potência.

Portanto, caso o usuário deseje utilizar um modelo desenvolvido com o MATLAB/Simulink em uma simulação ATP, mas esse modelo contenha blocos sem compatibilidade com o RTW, será necessário realizar uma simulação integrada, onde ambos os softwares – ATP e MATLAB/Simulink – executam o processamento numérico concorrentemente durante o tempo de execução da simulação, ou seja, o ATP fica responsável pelo processamento dos transitórios eletromagnéticos e o MATLAB/Simulink fica responsável pela simulação do modelo auxiliar. A desvantagem desse método é a perda de velocidade de processamento devido ao emprego de processos concorrentes para a execução de uma tarefa.

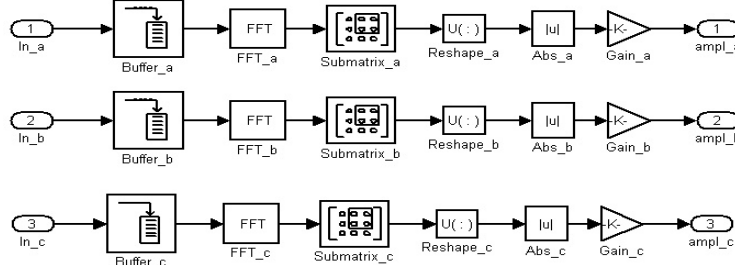
4.2 O Modelo do Relé Digital e a Ferramenta Gráfica para Automação da Codificação

Um relé de proteção do tipo sobretensão, por exemplo, pode ser modelado no MATLAB/Simulink empregando apenas blocos compatíveis com o RTW, de forma que seja possível produzir um código em linguagem C que descreva o comportamento dinâmico do equipamento. Um modelo desse tipo é apresentado na Figura 1.

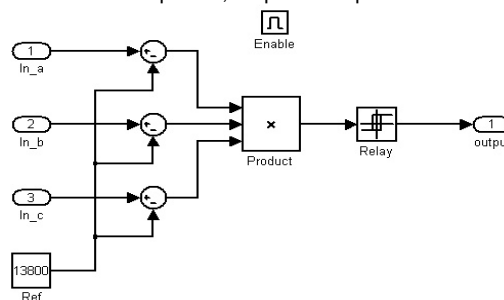
As funções escritas em linguagem C, produzidas pelo RTW, não podem ser diretamente empregadas pelo ATP como uma *foreign function*. Antes, é necessário escrever uma única função que realize a chamada, na ordem correta, das funções produzidas pelo RTW. Esta última função é que será, finalmente, empregada pelo ATP como uma *foreign function*. Entretanto, esse procedimento requer que o usuário conheça o tipo de código gerado pelo RTW e saiba como as funções geradas por essa ferramenta devem ser chamadas. Com o objetivo de tornar todo esse processo transparente para o usuário, foi desenvolvida uma ferramenta de software, com uma interface gráfica amigável, responsável por automatizar todo o processo de adequação do código produzido pelo RTW para ser utilizado em uma simulação ATP. Além disso, a ferramenta desenvolvida também produz automaticamente o código, em linguagem MODELS, a ser adicionado à simulação ATP, contendo a chamada à *foreign function*. Um arquivo MAKE também é criado automaticamente, contendo as instruções para a compilação de todos os arquivos de código produzidos.



(a) Modelo MATLAB/Simulink para um relé de sobretensão simples.



(b) Modelo do subsistema Detector de Amplitude, responsável pelo cálculo da amplitude das tensões medidas.



(c) Modelo do subsistema Comparador, responsável por comparar as amplitudes das tensões medidas com o limiar de sobretensão do relé – nesse caso o limiar é de 13800 V.

Figura 1 – Modelo do relé de sobretensão construído a partir de blocos do MATLAB/Simulink.

No caso, a partir do modelo MATLAB/Simulink do relé, produziu-se um código escrito em linguagem C que foi incorporado à simulação ATP via interface externa MODELS, o qual demonstrou correto funcionamento da lógica de detecção de defeitos para as simulações realizadas. No entanto, nem sempre esse tipo de incorporação é possível. Nas próximas seções, abordaremos o problema de estabelecer um canal de comunicação entre o ATP e o MATLAB/Simulink para realizar o intercâmbio de dados e permitir a execução conjunta de uma simulação.

5.0 - OS MECANISMOS DE IPC E A INTEGRAÇÃO ATP – MATLAB/SIMULINK

5.1 O MATLAB Engine

O software MATLAB disponibiliza uma biblioteca – denominada *MATLAB Engine* – a qual permite que um programa externo execute o MATLAB, envie dados numéricos, execute comandos e resgate os resultados. Dessa forma, pode-se desenvolver um programa que faça uso do MATLAB como um motor de processamento numérico de dados. Essa flexibilidade do MATLAB representa uma enorme vantagem, tendo-se em vista que esse software possui uma grande variedade de ferramentas de cálculo já implementadas. Além disso, a biblioteca *MATLAB Engine* representa uma porta de comunicação por onde um software externo – o ATP, por exemplo – pode utilizar todas as vantagens do MATLAB para a execução de uma simulação conjunta.

Um exemplo da aplicação dessa técnica é encontrado na referência [5]. No referido trabalho, a biblioteca *MATLAB Engine* é utilizada para permitir que o EMTP-EPRI/DCG faça uso de modelos de sistemas criados com o MATLAB/Simulink. Nesse trabalho, foi possível realizar uma simulação conjunta com ambos os softwares de simulação, onde o EMTP-EPRI/DCG responde pelo processamento numérico do sistema elétrico e o MATLAB/Simulink responde pelos cálculos relativos ao modelo de um relé de proteção. Apesar de a solução ser funcional, há uma significativa perda de desempenho quando a simulação é realizada com ambos os softwares operando concorrentemente.

Em simulação com rede representada por três nós e quatro elementos de circuito o teste de desempenho do relé de sobretensão apresentada na seção 4.2, empregando-se a estratégia de integração apresentada naquela seção leva a um tempo total de execução de aproximadamente um segundo. Por outro lado, usando a estratégia de integração similar à descrita em [5] – via funções da biblioteca *MATLAB Engine* –, a mesma simulação demorou

cerca de um minuto para ser completamente executada. Essa enorme perda de desempenho motivou a investigação de um mecanismo de comunicação entre os processos do ATP e do MATLAB/Simulink, em tempo de execução, que fosse mais eficiente que a biblioteca *MATLAB Engine*.

5.2 Os Mecanismos IPC do sistema operacional MS-Windows

De acordo com a documentação do MATLAB, a biblioteca *MATLAB Engine* realiza a comunicação – no sistema operacional Windows - através de um mecanismo conhecido como COM (*Component Object Model*). De acordo com a documentação disponibilizada pela própria Microsoft [6], existem nove diferentes meios para realizar a comunicação entre processos concorrentes: *Clipboard*, COM (*Component Object Model*), *Data Copy*, DDE (*Dynamic Data Exchange*), *File Mapping*, *Mailslot*, *Pipes*, RPC (*Remote Procedure Call*) e *Windows Sockets*.

De acordo com [6], a principal vantagem da ferramenta COM é a capacidade de realizar a troca de dados entre uma enorme variedade de aplicações, devido a sua generalidade. Devido a essa generalidade, esse mecanismo de comunicação é bastante adequado para a biblioteca *MATLAB Engine*, já que o seu objetivo é exatamente o de fornecer ao programador uma interface de comunicação com o MATLAB que possa ser incluída em qualquer aplicação que esteja sendo desenvolvida. Por outro lado, por ser um mecanismo bastante geral, esse meio de comunicação deve incluir dados adicionais às mensagens trocadas pelos processos e ainda deve executar operações de manipulação desses dados que sobrecarregam o andamento da simulação. Tal mecanismo não é adequado para integração de aplicações que necessitem de bom desempenho computacional.

Inicialmente, a semelhança da implementação realizada em [2] optou-se por explorar o uso do mecanismo de comunicação *File Mapping*. O sistema de *File Mapping* nada mais é do que a comunicação entre processos realizada através de arquivos mapeados em áreas de memória compartilhada. Em testes realizados com o exemplo do relé de sobretensão da seção anterior, o desempenho desse tipo de comunicação foi significativamente pior que o obtido com a biblioteca *MATLAB Engine*.

Dentre os demais mecanismos disponíveis o *Pipes* foi considerado a melhor alternativa para a viabilização da integração, tendo em vista ser simples o suficiente para que o processo de comunicação entre as duas ferramentas não representasse ônus em demasia ao desempenho computacional final. Um *pipe* é uma espécie de canal de comunicação entre dois processos, por onde ambos podem transferir dados entre si. De acordo com [6], há dois tipos de *pipes* disponíveis: os *Named Pipes* e os *Anonymous Pipes*. Os *Anonymous Pipes* são empregados exclusivamente para estabelecer a comunicação entre dois processos relacionados entre si, por exemplo, uma comunicação entre um processo pai e o seu processo filho, ou então entre processos filhos de um mesmo pai. Já os *Named Pipes* podem ser empregados na comunicação entre dois processos que não apresentam nenhuma relação entre si. Nesse caso, o *pipe* deve receber um nome que o identifique, para que ambos os processos envolvidos possam acessá-lo. Como os *Named Pipes* apresentam maior flexibilidade, optou-se por utilizá-los no presente projeto.

Além disso, os *pipes* podem realizar a transferência de dados entre processos através de dois mecanismos diferentes: mensagens ou uma corrente (*stream*) de bytes. A comunicação via mensagens possui o mesmo inconveniente das informações adicionais que devem ser encapsuladas na mensagem juntamente com os dados. Já na comunicação via corrente de bytes, os dados são transmitidos através do *pipe* em sua forma bruta (bytes), sem nenhuma informação adicional. Nesse último tipo de transferência de dados, ambos os processos devem saber como os dados estão dispostos na corrente de bytes e quantos bytes devem ser lidos ou escritos. Por outro lado, como esse tipo de comunicação é bastante simples, isso o torna mais rápido que os demais.

Para realizar a comunicação entre dois processos via *pipes*, deveremos ter um processo realizando o papel de servidor e o outro o papel de cliente. Essa distinção é feita porque o processo responsável por criar o *pipe* é necessariamente o servidor. Ao cliente resta apenas a função de se conectar a um *pipe* já existente. Quanto ao sentido do fluxo de dados, podemos ter três tipos diferentes de *pipe*: aquele em que o servidor apenas envia dados para o cliente, aquele em que o servidor apenas lê os dados enviados pelo cliente e aquele em que é permitida a comunicação bidirecional, onde ambos – servidor e cliente – podem escrever e ler dados. Em uma simulação conjunta entre o ATP e o MATLAB/Simulink, podemos definir o processo MATLAB/Simulink como sendo o servidor que disponibiliza modelos de dispositivos e equipamentos. Já o processo ATP seria encarado como um cliente que solicita ao servidor MATLAB a resposta de um determinado modelo com respeito a um conjunto de entradas definidas ao longo do tempo de simulação.

6.0 - MELHORIA DE DESEMPENHO COM O EMPREGO DE PIPES

A comunicação através de *pipes* é realizada através da chamada de funções escritas em linguagem C, disponíveis em bibliotecas fornecidas pela Microsoft [6]. Dessa forma, tanto o ATP quanto o modelo MATLAB/Simulink deverão ser capazes de executar rotinas escritas nessa linguagem, que realizam chamadas às funções responsáveis pelas operações sobre os *pipes* (criar, escrever, ler e remover).

No caso do ATP, essas chamadas podem ser realizadas pela própria *foreign function*, chamada pela interface externa MODELS – exatamente como foi feito na Seção 4. No caso do modelo MATLAB/Simulink, será necessário incluir blocos denominados *S-Function* ao diagrama de blocos do modelo. Um bloco *S-Function* realiza uma operação sobre os seus sinais de entrada para produzir suas saídas de acordo com um conjunto de rotinas escritas em linguagem de alto nível – como C. Dessa forma, é possível incluir em um modelo MATLAB/Simulink as chamadas de funções que realizam as operações relativas à manipulação de *pipes*. Dessa forma, o modelo do relé apresentado na Figura 1 deve ser modificado de forma a incluir uma *S-Function* para receber os dados do ATP e entregá-los ao modelo na forma de sinais de entrada, e uma outra *S-Function* ficará responsável por receber a resposta do modelo e devolvê-la ao ATP através de um *pipe*. Na Figura 4, temos o diagrama de blocos do modelo do mesmo relé incluindo as duas *S-Functions*.

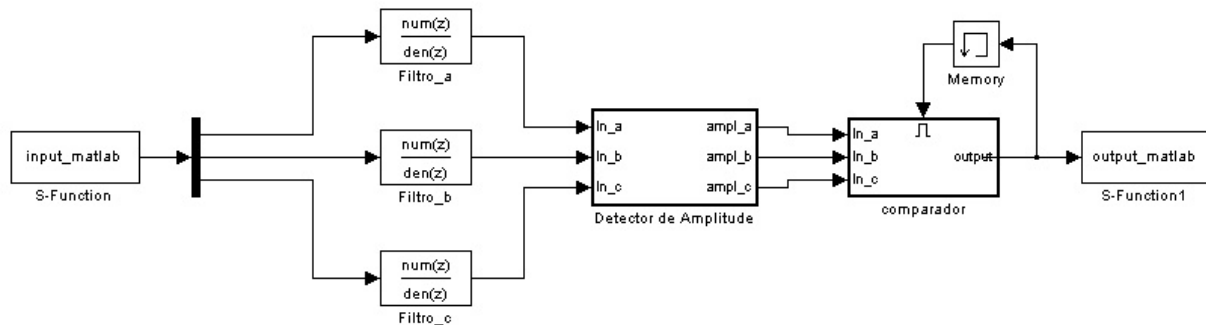


Figura 4 – Modelo MATLAB/Simulink do relé de sobretensão incluindo S-Functions.

Portanto, com base nas considerações feitas acima, o esquema de comunicação entre o ATP e o MATLAB é ilustrado através do diagrama apresentado na Figura 5. De acordo com esse diagrama, há dois *pipes*: um para que o cliente (ATP) envie dados para o servidor (a *S-Function* de entrada do modelo Simulink) e um outro para que o servidor (a *S-Function* de saída do modelo Simulink) envie dados para o cliente (ATP). Ambos os *pipes* são unidirecionais e devem ser do tipo *blocking*, isto é, toda vez que um processo for realizar a leitura dos dados do *pipe* e tais dados ainda não foram disponibilizados pelo outro processo concorrente, a execução do processo que deseja fazer a leitura é imediatamente suspensa pelo sistema operacional. Dessa forma, o processador ficará livre para executar o outro processo concorrente e, assim, produzir os dados aguardados pelo primeiro. Assim que os dados estiverem disponíveis no *pipe*, o sistema operacional imediatamente escala o processo em estado de espera para entrar em execução. Portanto, o mecanismo de comunicação através de *pipes* apresenta uma vantagem adicional em relação a outros métodos IPC: o tempo desperdiçado pela espera é eliminado e o processo em estado de espera apenas irá aguardar o tempo necessário para que o outro processo realize a sua tarefa e disponibilize os dados no *pipe*.

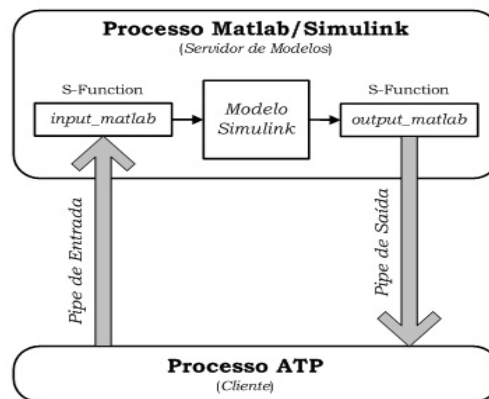


Figura 5 – Esquema de comunicação entre o ATP e um modelo MATLAB/Simulink através de *pipes*.

Baseado na Figura 5, a comunicação entre os processos ATP e Matlab/Simulink ocorre da seguinte forma:

1. O processo ATP realiza suas tarefas numéricas enquanto o processo da *S-Function input_matlab* está em estado de espera, aguardando que seus dados de entrada estejam disponíveis no *pipe* de entrada.
2. O processo ATP finalmente produz os dados a serem enviados ao modelo Simulink e os disponibiliza no *pipe* de entrada. Depois disso, o processo ATP entra em estado de espera, aguardando que a resposta do modelo Simulink seja disponibilizada no *pipe* de saída.
3. Com o processo ATP em estado de espera, o processo Matlab/Simulink entra em execução e a *S-Function input_matlab* realiza a leitura dos dados no *pipe* de entrada.

4. A nova entrada é, então, processada pelo modelo Simulink e a resposta é disponibilizada no *pipe* de saída pela *S-Function output_matlab*.
5. Novamente a *S-Function input_matlab* entra em estado de espera, aguardando uma nova amostra do sinal de entrada no seu respectivo *pipe*.
6. O processo ATP entra novamente em execução, realiza a leitura dos dados no pipe de saída e continua o seu processamento até o fim dos cálculos necessários ao atual passo de integração.
7. Ao iniciar um novo passo de integração, o processo ATP volta a repetir todo o processo deste o item 1.

Executando a mesma simulação do item 5.1 (rede representada por três nós e quatro elementos de circuito), os resultados obtidos foram idênticos aos anteriores. Entretanto, diferentemente da simulação onde a integração entre ATP e MATLAB/Simulink fora realizada através da biblioteca *MATLAB Engine* – a qual durou cerca de um minuto –, o tempo de simulação usando a interface com *pipes* durou apenas três segundos. Apesar de ser um tempo de execução maior que aquele verificado quando o modelo do relé foi convertido em um conjunto de rotinas codificadas em linguagem C, o desempenho foi significativamente melhor que o verificado em outras tentativas de ser realizar uma simulação integrada entre o ATP e o MATLAB/Simulink, operando simultaneamente durante o tempo de execução [2] e [5].

7.0 - CONCLUSÕES E PERSPECTIVAS

Através de projetos realizados em parceria, contemplados nos planos de ação bienais, o ONS vem viabilizando a incorporação de melhorias ao ferramental computacional empregado em seus estudos de transitórios eletromagnéticos, com o objetivo de aprimorar as respectivas análises e agilizar execução dos mesmos. Esse ferramental, o qual compreende: pré-processadores, pós-processadores e integrador de processos, vem sendo utilizado em caráter preliminar nos estudos pré-operacionais e de recomposição do Sistema Integrado Nacional (SIN) realizados pelo próprio operador.

No que diz respeito à interface de integração do ambiente de cálculo e simulação associado ao programa ATP à sua contrapartida associada ao software MATLAB/Simulink, mostrou-se como é possível auferir ganhos expressivos em qualidade e flexibilidade de representação de equipamentos e controladores, através do uso simultâneo e/ou compartilhado das bibliotecas de ambas as ferramentas de simulação. Somando-se a isso foi demonstrada, também, a possibilidade de visualização e pós-processamento dos gráficos das formas de onda do ATP simultaneamente às simulações, empregando as facilidades e recursos do ambiente gráfico MATLAB/Simulink numa forma bastante eficiente.

Através do suporte necessário ao desenvolvimento dessa interface para integração de processos associados a ambientes de simulação distintos, o ONS passou a contar com um ferramental de simulação de transitórios eletromagnéticos no qual estarão incorporadas as melhores funcionalidades do ATP e do MATLAB/Simulink. Funcionalidades essas caracterizadas no caso do ATP, pela robustez do algoritmo de solução e versatilidade de representação e no caso do MATLAB/Simulink pela oferta de bibliotecas especializadas, facilidade de programação e modernidade da interface homem-máquina.

O desdobramento futuro deste trabalho é o desenvolvimento de uma ferramenta gráfica amigável que possibilite aos especialistas em transitórios eletromagnéticos do ONS realizar uma simulação integrada entre os softwares de simulação sem a necessidade de se ater aos detalhes de implementação da interface de comunicação entre os programas. Pretende-se, através da metodologia desenvolvida, poder viabilizar a integração do ATP com outras ferramentas de simulação como, por exemplo, o PSCAD/EMTDC. Isso deverá tornar desnecessária a conversão ao ATP de modelos utilizados pelos fabricantes de equipamentos nos estudos com aquela ferramenta, bastando apenas que os mesmos atendam a um protocolo específico a ser definido oportunamente.

8.0 - REFERÊNCIAS BIBLIOGRÁFICAS

- [1] N. Zhang, M. Kezunovic. "Implementing an advanced simulation tool for comprehensive fault analysis". IEEE/PES Proc. on Transmission and Distribution Conference and Exhibition: Asia and Pacific, pp. 1-6, 2005.
- [2] M. Kezunovic, "A novel approach for interactive protection system simulation". IEEE Transactions on Power Delivery, Vol. 12, No. 2, pp. 668-674, April 1997.
- [3] X. Luo, M. Kezunovic. "Interactive protection system simulation using ATP MODELS and C++", PES TD 2005/2006, pp. 874-879, May 21-24 2006.
- [4] H. M. Barros, R. M. Fernandes. "Implementação de biblioteca de modelos genéricos para simulação de relés de proteção digital no programa de transitórios eletromagnéticos ATP – Alternative Transients Program", VIII STPC (Seminário Técnico de Proteção e Controle), Rio de Janeiro, junho/julho 2005.
- [5] J. Mahseredjian, G. Benmouyal, X. Lombard, M. Zouiti, B. Bressac, L. Gérin-Lajoie. "A link between EMTP and MATLAB for user-defined modeling". IEEE Transactions on Power Delivery. pp. 667-674, vol. 13, n. 2, April, 1998.
- [6] <http://msdn2.microsoft.com/en-us/library/aa365574.aspx>