https://eventos.utfpr.edu.br//sicite/sicite2019

Desenvolvimento de uma ferramenta para a obtenção automática das equações utilizadas na simulação dinâmica

Development of a tool for automatic obtaining of the equations used in dynamic simulation

André Luís Henchenski
andrehenc@gmail.com
Universidade Tecnológica Federal do Paraná, Medianeira, Paraná, Brasil

Diogo Marujo
diogomarujo@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Medianeira, Paraná, Brasil

Recebido: 19 ago. 2019.
Aprovado: 01 out. 2019.

RESUMO

O amplo uso da energia elétrica impõe a necessidade de se ter um sistema elétrico confiável capaz de operar dentro de limites pré-estabelecidos e suportar perturbações. Para garantir a eficiência dos dispositivos de proteção, expansão do sistema e a estabilidade elétrica, diversos estudos dinâmicos são necessários. A maioria desses estudos são feitos através de simulações por meio da solução numérica das equações que descrevem os elementos que compõe o sistema. Para isso são utilizados diversos softwares de simulação dinâmica que são responsáveis por resolver conjuntos algébrico-diferenciais de equações que representam o sistema. As equações provêm da modelagem matemática desses elementos levando em consideração os efeitos relevantes para o tipo de estudo desejado. O mesmo elemento pode ter diferentes equações a depender do intervalo de tempo do estudo. A inclusão ou modificação de novos modelos implica em novas equações que podem ser trabalhosas de se obter e implementar. Neste trabalho desenvolve-se uma ferramenta para obtenção das equações necessárias para a modelagem de dispositivos na simulação dinâmica. A ferramenta visa padronizar uma entrada, evitando a tarefa de inserir novos modelos por linhas de códigos, descrevendo um diagrama de blocos do qual as equações necessárias para descrição do modelo são obtidas.

PALAVRAS-CHAVE: Estabilidade. Dinâmica. Modelagem.

ABSTRACT

The wide use of electric energy imposes the need to have a reliable electrical system capable of operating within pre-established limits and withstanding disturbances. To ensure the efficiency of protection devices, system expansion and electrical stability, several dynamic studies are required. Most of these studies are done through simulations by performing the numerical solution of the equations that describe the elements that make up the system. For this sake, several dynamic simulation software are used, which are responsible for solving algebraic-differential sets of equations that represent the system. The equations come from the mathematical modelling of these elements taking into account the effects relevant to the type of study desired. The same element may have different equations depending on the time interval of the study. The inclusion or modification of new models implies new equations that can be difficult to obtain and implement. In this work a tool is developed to obtain the necessary equations for the modelling of devices in the dynamic simulation. The tool aims to standardize an input, avoiding the task of inserting new models by code lines, describing a block diagram from which the equations necessary to describe the model are obtained.

KEYWORDS: Stability. Dynamic. Modelling.

INTRODUÇÃO

Dada a complexidade e volatilidade do Sistema Elétrico de Potência (SEP) se faz necessário utilizar simulações para conduzir estudos que promovam um melhor desempenho operacional e econômico desse sistema. Tais estudos contribuem essencialmente para o gerenciamento do fluxo de potência, chaveamentos, otimização e gerenciamento dos sistemas de geração e transmissão, projeto de sistemas de proteção, robustez do sistema perante faltas e frente possível operação ilhada Saadat (1999).

Para a simulação dinâmica são necessários modelos matemáticos que representem os dispositivos e são descritos por funções de transferência ou equações de estado. A representação do modelo matemático pode ser feita a partir de diagrama de blocos ou diagrama de fluxo de sinais Nise (2015).

Em muitos softwares para simulação dinâmica de sistema de potência, a inserção de novos modelos de controladores do sistema é feita via linhas de código. Tal inconveniente poderia ser evitado se existisse uma interface para a inclusão de novos modelos via diagramas de blocos por meio de um conjunto de instruções padrão das quais a plataforma seria capaz de extrair as equações algébrico-diferenciais que ditam o comportamento do elemento modelado.

Em Ananias (2016) o autor flexibiliza a modelagem matemática para simulação dos componentes do SEP através de uma ferramenta programada em Python. As informações referentes aos modelos são obtidas a partir dos arquivos do CDU do ANATEM que são convertidos nos blocos básicos e nos grafos orientados para que seja possível a interpretação pela ferramenta.

Em Manzoni (2005) o autor cria uma ferramenta que através de programação orientada a objetos permite a implementação e flexibilização de novos modelos definidos pelo usuário às demais ferramentas de simulação dinâmica, fluxo de potência e análise modal desenvolvidas. Também foi proposta uma modificação nos métodos de simulação rápida de médio e longo prazo para expandir os tipos de fenômenos suportados pelo simulador.

Em Marujo (2017) o autor desenvolve uma ferramenta computacional em Matlab para análise de estabilidade de SEP. A ferramenta possui seis módulos responsáveis pela análise dinâmica, análise quase-dinâmica, fluxo de potência monofásico e trifásico, fluxo de potência unificado e um módulo híbrido com a análise quase-dinâmica e o fluxo de potência unificado. Os módulos fazem a simulação através da resolução de sistemas algébricos ou sistemas algébrico-diferenciais. As equações algébrico-diferenciais são solucionadas separadamente através de integração explícita ou transformando as equações diferenciais em equações algébricas através de um método de integração implícita, como o trapezoidal. Contudo, na abordagem apresentada, o usuário deve adicionar manualmente novos modelos, por meio de *scritps*.

Diante do exposto, nota-se que a inserção de novos modelos de controladores do sistema de potência via linhas de código pode tornar-se uma tarefa árdua para o usuário, principalmente para aqueles que não possuem maior facilidade com programação. Tal dificuldade poderia ser superada se existisse uma interface para a inclusão de novos modelos via diagramas de blocos, de maneira análoga ao Simulink Mathworks (2019), toolbox do Matlab. Assim, o foco deste trabalho é o desenvolvimento de uma ferramenta em Matlab que possibilite a construção de novos controladores utilizados em SEP. Neste sentido, após a descrição dos diagramas de blocos por meio de um conjunto de instruções padrão (e não a programação propriamente dita), isto é, uma entrada padronizada, a plataforma será capaz de extrair equações algébrico-diferenciais que ditam o comportamento do elemento modelado, manuseá-las e determinar as condições inicias necessárias para sua resolução, de maneira análoga ao que ocorre com o programa CDUEdit do CEPEL (2019), que trabalha de maneira conjunta com o ANATEM CEPEL(2015).

O objetivo deste trabalho é desenvolver uma ferramenta em Matlab, com programação orientada a objetos, que auxilie na obtenção das equações algébrico-diferenciais utilizadas na simulação dinâmica, permitindo a descrição dos modelos através de entradas padronizadas gerando um diagrama de blocos que dará origem as equações algébrico-diferencias que descrevem os controladores.

MÉTODOS

O programa a ser desenvolvido utilizará a programação em Matlab permitindo a implementação através de scripts, funções e classes. Scripts são a forma mais simples de programação no Matlab e são úteis para automatizar comandos que devem ser executados sequencialmente durante alguma operação. Funções são um conjunto de scripts necessários para facilitar a implementação de procedimentos que devem ser repetidos em momentos diferentes e sobre diversas variáveis sem que haja a necessidade de se repetir as linhas de código.

Classes descrevem um conjunto de objetos com características comuns armazenando dados específicos e as funções relevantes facilitando a manipulação do conjunto. Esses dados e funções são chamadas de propriedades e métodos e ditam o comportamento interno e frente outros objetos.

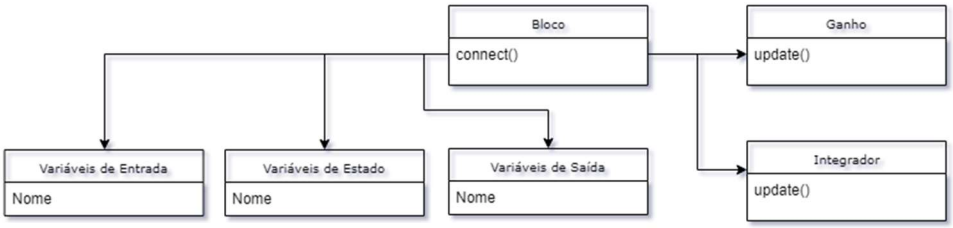
Ao definir uma classe, portanto, é necessário nomeá-la, definir as propriedades que ela contém e os métodos que se aplicam a ela. Uma classe pode ainda herdar ou repassar suas características a uma outra classe facilitando o reuso das técnicas já utilizadas até então e eliminando a necessidade de reescrever partes do código.

Assim, a programação orientada a objeto é uma forma simples de gerenciar funções de um programa complexo com um grande número de funções encapsulando os dados e suas operações nos objetos que interagem entre si pela interface de objetos.

O diagrama de blocos será interpretado pelo programa através da descrição de alguns objetos com suas respectivas propriedades e métodos. Cria-se a classe **Bloco** contendo nos parâmetros um vetor de entrada e de saída e nos métodos a função ***connect()***, responsável por representar as conexões.

A partir desse objeto **Bloco** os outros blocos com funções mais específicas, como ganho e integrador, são herdados e possuem um método ***update()***, responsável por relacionar a entrada com a saída utilizando os parâmetros através da equação correspondente ao bloco. As variáveis de estado e saída servem de ponto de partida para a varredura do diagrama e os blocos integradores armazenam a equação obtida correspondente a uma equação de estado.

Figura 1 – Fluxograma das relações entre os objetos



Fonte: Autoria própria.

O programa varre os blocos através de uma lista encadeada atualizando a saída dos blocos que possuem as entradas disponíveis. Ao receber a entrada o bloco realiza a manipulação necessária nela e propaga sua saída continuando em uma reação em cadeia

até que uma variável de estado, saída ou algébrica seja encontrada. Quando todos os blocos receberem uma atualização na entrada a varredura está completa e o sistema de equações de estado pode ser obtido.

Desta maneira, os blocos são os responsáveis pela manipulação algébrica das variáveis. As equações armazenadas nos blocos integradores são as equações linearmente independentes necessárias para descrever o diagrama de blocos por completo através de um sistema de equações algébrico-diferenciais. Através desse sistema de equações será possível descrever os modelos de controladores desejados na forma de equações de estado.

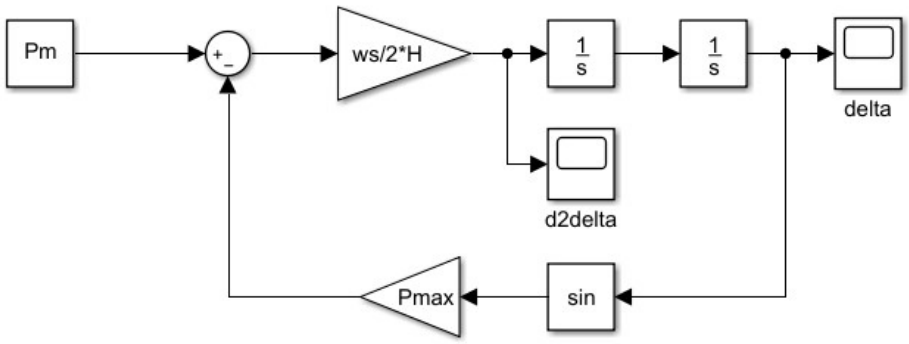
RESULTADOS E DISCUSSÕES

Uma versão inicial da ferramenta proposta já foi desenvolvida e algumas equações já podem ser obtidas na forma de variáveis simbólicas do Matlab. A automatização da varredura do diagrama ainda não foi implementada. O formato das equações resultantes não é padronizado, mas os blocos e as conexões conseguem operar as variáveis algebricamente e retornar equações mais simples.

Para demonstrar a atual capacidade do programa considere o diagrama de blocos da Figura 2, descrito no Simulink, que representa a equação de *swing* da máquina síncrona, Eq. (1).

$$\frac{d^2\delta}{dt^2} = \frac{\omega_s}{2H}(P_m - P_e) \quad (1)$$

Figura 2 – Diagrama de blocos no Simulink da equação swing



Fonte: Autoria própria.

O diagrama foi transcrito para o programa através das linhas de códigos apresentadas na Figura 3, já que a leitura de dados ainda não foi implementada.

As variáveis foram declaradas e posicionadas nos respectivos blocos. Foram dados os comandos de conexão dos blocos e atualização das saídas, entretanto, os blocos integradores não são capazes ainda de associar a saída com a sua derivada na entrada, sendo necessário declarar as derivadas como variáveis, mas conseguem dar origem as equações necessárias.

A equação de *swing* obtida pela execução do programa encontra-se na entrada do primeiro integrador e o programa retornou a Eq. (2):

$$\mathrm{d2delta} == (\mathrm{ws} * (\mathrm{Pm} - \mathrm{Pmax} * \sin(\mathrm{delta})))/(2 * \mathrm{H}) \quad (2)$$

onde d2 representa a segunda derivada e os dois sinais de iguais são naturais da manipulação de variáveis simbólicas no Matlab.

A equação obtida é só um exemplo para mostrar as atuais capacidades do programa. Os resultados atingidos não correspondem a versão final da ferramenta, mas servem para demonstrar o que se pretende atingir e verificar as suas possíveis limitações. A ferramenta mostra-se capaz de obter as equações diferenciais correspondentes ao diagrama de blocos descrito.

Figura 3 – Linhas de código utilizadas

```
%%Cria os blocos
syms ws H Pm Pmax delta ddelta d2delta ; %Declaração das variáveis do
problema
Gain1 = Ganho(ws/(2*H));
Gain2 = Ganho(Pmax);
Sum1 = Somador;
Int1 = Integrador;
Int2 = Integrador;
Trig1 = Seno;

%Define-se as variaveis de estado como entradas dos blocos a que estão
%conectadas
Int1.varEst = d2delta;
Int2.varOut = delta;
Sum1.varIn = Pm;
Sum1.sinal = '+';

%Conectam-se os blocos
connect(Trig1, Int2)
update(Trig1)

connect(Gain2,Trig1)
update(Gain2)

connect(Sum1, Gain2, '-')
update(Sum1)

connect(Gain1, Sum1)
update(Gain1)

connect(Int1,Gain1)
update(Int1)

%Equações
Int1.eq
```

Fonte: Autoria própria.

CONCLUSÃO

A ferramenta apresentada facilita a obtenção das equações que descrevem os modelos matemáticos dos dispositivos possibilitando a implementação de novos modelos de controladores em estudos de simulação dinâmica.

Em seu estado atual a ferramenta já é capaz de obter equações mais simples. A programação orientada a objetos facilita a inserção de novos blocos com novas operações matemáticas para descrição de sistemas mais complexos.

Um desafio no desenvolvimento da ferramenta é a correta manipulação algébrica das variáveis frente as operações o que dificulta a automatização do processo de varredura do diagrama.

A seguir são apresentados os próximos passos para o desenvolvimento da ferramenta:

a) Padronizar as entradas;

b) Declarar as variáveis como blocos;

c) Implementar a automatização da varredura do diagrama.

REFERÊNCIAS

SAADAT, Hadi. **Power System Analysis.** New York: McGraw-Hill, 1999.

NISE, Norman S**. Control Systems Engineering.** Pomona: Wiley, 2015.

ANANIAS, João Paulo Silva. **Metodologia para Modelagem e Simulação de Curto e Longo Termo em SIstemas Elétricos de Potência.** 2016. 222 f. Tese (Mestrado em Engenharia Elétrica), Universidade Federal de Juiz de Fora, Juiz de Fora, 2016.

MANZONI, Alessandro. **Desenvolvimento De Um Sistema Computacional Orientado A Objetos Para Sistemas Elétricos De Potência**: Aplicação A Simulação Rápida E Análise Da Estabilidade De Tensão. 2005. 165 f. Tese (Doutorado em Ciências em Engenharia Elétrica), Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.

MARUJO, Diogo. **Estabilidade de Sistemas Elétricos de Potência com a Presença de Rede de Distribuição Ativas.** 2017. 199 f. Tese (Doutorado em Ciências em Engenharia Elétrica), Universidade Federal de Itajubá, Itajubá, 2017.MATWORKS. Documentação do Simulink. Disponível em <https://www.mathworks.com/help/simulink/index.html>. Acesso em: 15 mai. 2019.

CEPEL. **CDUEdit**: Editor de Controladores Definidos pelo Usuário. Disponível em: <http://www.cepel.br/pt_br/produtos/cduedit-editor-de-controladores-definidos-pelo-usuario.htm>. Acesso em: 15 jun 2019.

CEPEL. **Programa ANATEM**: Manual do Usuário. Rio de Janeiro: CEPEL, 2015.