

Circuitos Elétricos II – 2º semestre de 2014 - Trabalho

Prof. Antônio Carlos Moreirão de Queiroz

Fazer um programa que analise circuitos no tempo, e que aceite, no mínimo, os elementos:

- Fontes de corrente e de tensão independentes (DC, pulso, senóide).
- Capacitores e indutores lineares invariantes no tempo.
- Acoplamentos entre indutores.
- Resistores, possivelmente polinomiais.
- As quatro fontes controladas, possivelmente polinomiais.
- Amplificadores operacionais ideais, de 4 terminais.

O programa deve ler um netlist, e realizar uma análise transiente, com parâmetros dados por uma linha de comando no netlist. O resultado deverá ser uma tabela em arquivo, tendo o tempo como primeira coluna, e todas as tensões nodais e correntes nas fontes de tensão nas outras colunas, plotável com outro programa. A primeira linha da tabela deve listar os nomes de todas as variáveis calculadas, com correntes citadas como “j” seguidas do nome do elemento onde estão. Ex: t 1 2 3 4 5 6 jH1 jF2.

O método de integração numérica a usar é o “backward” de Euler. Inicialmente deverá ser feita uma análise de ponto de operação para a determinação das condições iniciais, com capacitores em aberto e indutores em curto-circuito. (Melhor colocar resistores grandes e pequenos nos lugares de capacitores e indutores.)

A análise deverá ser feita usando análise nodal modificada, com correntes em indutores calculadas. O programa pode ser baseado no programa exemplo mna1, que implementa o algoritmo para um circuito resistivo linear: <http://www.coe.ufrj.br/~acmq/cursos/mna1.zip>

Formato para o netlist:

O netlist pode ser gerado pelo programa Edfil, e deve ser compatível com o aceito pelo programa exemplo mnae. Veja os programas em <http://www.coe.ufrj.br/~acmq/cursos>

Primeira linha: Comentário, ignorar (O editor Edfil coloca o número de nós nesta linha).

Linhas seguintes: Descrição do circuito, com um elemento por linha.

Resistor: R<nome> <nó1> <nó2> <Resistência>

Indutor: L<nome> <nó1> <nó2> <Indutância>

Capacitor: C<nome> <nó1> <nó2> <Capacitância>

Acoplamento: K<nome> <L1> <L2> <coeficiente de acoplamento>

Fonte de tensão controlada a tensão: E<nome> <nóV+> <nóV-> <nóv+> <nóv-> <Av>

Fonte de corrente controlada a corrente: F<nome> <nóI+> <nóI-> <nói+> <nói-> <Ai>

Fonte de corrente controlada a tensão: G<nome> <nóI+> <nóI-> <nóv+> <nóv-> <Gm>

Fonte de tensão controlada a corrente: H<nome> <nóV+> <nóV-> <nói+> <nói-> <Rm>

Fonte de corrente: I<nome> <nó+> <nó-> <Parâmetros>

Fonte de tensão: V<nome> <nó+> <nó-> <Parâmetros>

Amplificador operacional ideal: O<nome> <nó saída+> <nó saída-> <nó entrada+> <nó entrada->

Comentário: *<comentário>

O programa exemplo permite nomes nos nós. O programa feito pode continuar permitindo usando o mesmo algoritmo, ou admitir apenas números. Neste caso a primeira linha gerada pelo editor Edfil pode ser usada.

As direções para fontes são de acordo com a ordem dos nós e as direções convencionais associadas, sendo o primeiro nó o positivo. Os parâmetros para as fontes podem ser:

DC <valor>

SIN <nível contínuo x_0 > <amplitude A > <frequência f (Hz)> <atraso t_0 > <atenuação α > <ângulo ϕ (graus)> <número de ciclos>

PULSE <amplitude 1> <amplitude 2> <atraso> <tempo de subida> <tempo de descida> <tempo ligada> <período> <número de ciclos>

A fonte senoidal vale: $x(t) = x_0 + Ae^{-\alpha(t-t_0)} \text{sen}(2\pi f(t-t_0) + (\pi/180)\phi)$. Antes de $t = t_0$ ou depois do número de ciclos vale: $x(t) = x_0 + A \text{sen}((\pi/180)\phi)$.

A fonte pulsada começa na amplitude 1, e fica aí até o fim do tempo de atraso. Então muda para a amplitude 2 variando linearmente dentro do tempo de subida, fica na amplitude 2 durante o tempo ligada, volta à amplitude 1 dentro do tempo de descida, e repete tudo com o período e o número de ciclos especificados. Termina na amplitude 1. Tempos de subida e descida nulos podem ser substituídos pelo tempo do passo interno.

Para os elementos R, E, F, G e H, em vez do valor pode estar uma lista de até 8 valores, definindo um polinômio. Um resistor, por exemplo, poderia ser descrito como:

RX a b 1 -1 3 1 1 1 1 1

O que deve ser interpretado como uma regra $j = f(v) = 1 - v + 3v^2 + v^3 + v^4 + v^5 + v^6 + v^7$

Se existir apenas um parâmetro, ele é a resistência de um resistor linear. Uma descrição como:

RY a b 0 10

Significa uma condutância linear de 10 S. Se não existir o 0, é uma resistência de 10 Ω .

O programa deve ler as instruções de como tratar o netlist de uma linha de comando no próprio netlist, no formato abaixo. Não deve ser necessário fornecer qualquer outro parâmetro ao programa além do arquivo de entrada. Os passos internos permitem aumentar a precisão da análise, com alguns passos entre os valores que vão à tabela de saída. O passo interno usado é então o passo dado dividido pelo número de passos internos. Colocar um só passo interno significa que todos os valores calculados vão estar na tabela gerada.

.TRAN <tempo final> <passo> BE <passos internos>

O programa mnae pode ser usado para plotar os gráficos de saída e para verificação. Ele implementa os elementos polinomiais conforme o especificado.

O programa deve contar quantas vezes o ciclo de Newton-Raphson é executado, e se o número passar de um valor razoável, tentar outra aproximação inicial para a solução. Deve contar também quantas vezes faz isto, e se o número passar de um valor razoável, abortar a análise.

O programa deve ser escrito em uma linguagem compilada como C, C++ ou Pascal. O programa deve rodar em ambiente gráfico Windows (Windows 7, 32 bits). Um arquivo .zip com tudo o que for necessário para o programa, inclusive fontes, arquivo executável, documentação e exemplos não deve ter mais de 3 Mbytes. Evite-se sistemas de desenvolvimento que requeiram grandes instalações onde o programa for apenas rodar. É recomendado o uso de uma interface gráfica. O mnae está escrito com o Borland C++ builder 6.0 (atual Embarcadero).

Adicionais que podem ser incluídos: Gráficos, diodos com descrição polinomial (algo como um diodo Zener), transistores bipolares usando estes diodos, chaves, etc.

Grupos de 3 alunos, no máximo. O programa deverá ser apresentado e demonstrado por todo o grupo, e entregue com um relatório com comentários e exemplos significativos e originais verificados, até uma semana antes da segunda prova.