

Circuitos Elétricos II – 1º semestre de 2014 - Trabalho

Prof. Antônio Carlos Moreirão de Queiroz

Fazer um programa que analise circuitos lineares invariantes no tempo, encontrando a resposta em frequência.

O programa deverá analisar um circuito composto pelos elementos:

- Fontes de corrente e de tensão independentes.
- Resistores, capacitores, indutores e transformadores.
- As quatro fontes controladas.
- Amplificadores operacionais ideais, de 4 terminais.
- Transistores bipolares, com modelo de pequenos sinais.

O programa deve ler um netlist, e realizar uma análise nodal modificada no estado permanente senoidal para cada frequência em um conjunto dentro da faixa especificada. Uma linha de comando no netlist define os parâmetros necessários. O resultado deverá ser uma tabela em arquivo, tendo a frequência como primeira coluna, e todas as tensões nodais e correntes nas fontes de tensão nas outras colunas, em módulo e fase, plotável com outro programa. A primeira linha da tabela deve listar os nomes de todas as variáveis na tabela, com correntes citadas como “j” seguidas do nome do elemento onde estão e sufixos m e f para indicar módulo e fase. Ex: f 1m 1f 2m 2f 3m 3f jH1m jH1f jF2m jF2f.

O programa pode ser baseado no programa exemplo MNA1, que implementa a análise nodal modificada para um circuito resistivo linear. Uma versão gráfica (para o Borland C++ builder 6) está também disponível:

<http://www.coe.ufrj.br/~acmq/cursos/mna1.zip>
<http://www.coe.ufrj.br/~acmq/cursos/mna1gr.zip>

Formato para o netlist:

O netlist pode ser gerado pelo programa Edfil. 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. A primeira letra determina o tipo de elemento.

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

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

Acoplamento entre indutores: K<nome> <La> <Lb> <k> (La e Lb já declarados.)

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

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-> <módulo> <fase (graus)>

Fonte de tensão: V<nome> <nó+> <nó-> <módulo> <fase (graus)>

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

Transistor bipolar: Q<nome> <coletor> <base> <emissor> <hfe> <hie> <hre> <hoe>

Comentário: *<comentário>

(Notar que <xxx> significa colocar o valor xxx sem <>.)

O programa exemplo permite nomes nos nós. O programa feito pode continuar permitindo usando o mesmo algoritmo (código no programa MNA1), 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.

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.

```
.AC <LIN ou OCT ou DEC> <pontos> <início> <fim>
```

Como no SPICE, o número de pontos é por década ou por oitava com as opções DEC e OCT. Senão é o total com a opção LIN. As frequências são em Hz.

Opcionalmente, o próprio programa pode plotar seus gráficos. O programa MNAE pode ser usado para plotar os gráficos (embora não seja adequado para plotar em escala não linear, na versão atual). Programas como o gnuplot são adequados para isto também.

O programa deve ser escrito preferencialmente em uma linguagem compilada como C, C++ ou Pascal. O programa deve preferencialmente rodar em ambiente gráfico Windows, 32 bits. Um arquivo .zip com tudo o que for necessário para o programa, inclusive fontes, arquivo executável, documentação, bibliotecas (dll) e exemplos não deve ter mais de 3 Mbytes. Evite sistemas de desenvolvimento que requeiram extensas bibliotecas instaladas.

Sugere-se partir do programa exemplo MNA1, que já tem o algoritmo completo da análise nodal modificada, implementar a análise no estado permanente senoidal, operando com variáveis complexas, implementar a geração da tabela de saída, e então implementar os componentes reativos.

Grupos de 3 alunos, no máximo. O programa deverá ser apresentado e demonstrado, completamente funcional, por todo o grupo, e entregue com um relatório com comentários e exemplos significativos e originais verificados, até (entenda-se antes de) duas semanas antes da segunda prova.

Note-se que o trabalho é bastante extenso, e deve ser começado imediatamente.