

Circuitos Elétricos II – 2º semestre de 2013 - Trabalho

Prof. Antônio Carlos Moreirão de Queiroz

Fazer um programa que analise circuitos lineares invariantes no tempo, encontrando a resposta no tempo a um sinal periódico através de análise no estado permanente senoidal.

O programa deverá analisar um circuito composto pelos elementos:

- Fontes de corrente e de tensão independentes, contínuas, senoidais e pulsadas.
- Resistores, capacitores, indutores e transformadores.
- As quatro fontes controladas.
- Amplificadores operacionais ideais, de 4 terminais.

O programa deve ler um netlist, e realizar uma análise nodal modificada no estado permanente senoidal para cada componente da série de Fourier de cada fonte independente, e ao fim da análise gerar uma tabela com a soma das respostas no tempo correspondentes aos fasores calculados, o que deve ser uma aproximação da resposta em regime permanente do circuito. Uma linha de comando no netlist define os parâmetros necessários. 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.

As séries de Fourier podem ser calculadas a partir de expressões analíticas para os termos das séries (melhor) ou por integrações numéricas.

O programa pode ser baseado no programa exemplo MNA1, que implementa a análise nodal modificada 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. A primeira letra (ou símbolo) 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-> <Parâmetros>

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

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

Comentário: *<comentário>

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

Os programas exemplo permitem 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. Os parâmetros para as fontes podem ser:

DC <valor>

SIN <nível contínuo> <amplitude> <frequência (Hz)> <atraso*> <atenuação*> <ângulo> <número de ciclos*>

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

Os termos marcados com * devem existir, para compatibilidade com outros programas, mas são ignorados.

A fonte senoidal vale: nível contínuo + amplitude $\times \sin(2 \times \pi \times \text{frequência} \times t + \text{ângulo} \times \pi/180)$.

A fonte pulsada começa na amplitude 1, 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 dado. Os tempos de subida e de descida podem ser nulos. Pode ser tratado também o atraso, mas então ele vale em todos os ciclos.

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.

.TRAN <tempo final> <passo> [<número de termos nas séries>]

Se o número de termos não for dado, considere-se que a máxima frequência dos componentes não pode ser maior que $1/(2 \times \text{passo})$.

O programa MNAE pode ser usado para plotar os gráficos de saída e para comparação com uma análise no tempo calculada de outra forma (melhor usar o método dos trapézios ou o de Gear de 2ª. ordem). O programa Fourier faz o cálculo como pedido.

O programa deve ser escrito preferencialmente em uma linguagem compilada como C, C++ ou Pascal. O programa deve preferencialmente rodar em ambiente gráfico Windows. 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. O programa MNAE foi escrito usando o Borland C++ Builder.

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 as fontes DC, as senoidais, capacitores, indutores e transformadores, e por fim as fontes pulsadas.

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 (entenda-se antes de) duas semanas antes da segunda prova. Note que o trabalho é bastante extenso, e deve ser começado imediatamente.