Universidade Federal do Rio de Janeiro - UFRJ

Departamento de Engenharia Eletrônica e de Computação

Circuitos Elétricos II - Prof. Antônio Carlos Moreirão de Queiroz

Segundo semestre de 2016 - Entrega: 9 de dezembro de 2016

Grupo: Alice Fontes, Camyla Romão, Paulo Valente

Relatório

**1. Introdução**

Este trabalho visa a construção de um programa que analisa circuitos no domínio do tempo, utilizando análise nodal modificada e o método dos trapézios junto com o método de Newton-Raphson.

O programa analisa circuitos compostos pelos seguintes elementos: resistores, capacitores, indutores, fontes de corrente e de tensão independentes, os quatro tipos de fontes controladas, amplificadores operacionais ideais de 4 terminais, transformadores ideais, diodos ideais e chaves ideais controladas por tensão.

**2. Funcionamento básico do simulador**

Para utilizar o simulador, pode-se executar o programa diretamente. Nesse caso, o programa pedirá um arquivo de netlist no formato SPICE. Além disso, em ambiente Windows, pode-se executar o programa “arrastando” um arquivo de netlist “para cima” do executável. Nesse caso, o programa analisa o arquivo dado. Do mesmo modo, aceita um argumento de linha de comando, que será utilizado como nome de arquivo para analisar.

Primeiramente, o programa faz a leitura do netlist, separando os componentes em três categorias: componentes fixos, componentes variantes e componentes não-lineares. A cada categoria corresponde uma lista de objetos que guardam a informação necessária para a simulação.

Após a leitura do netlist, começam a ser criadas as variáveis correspondentes aos componentes fixos, como é o caso de resistores, transformadores e amp-ops. Então, programa executa uma análise de ponto de operação, também utilizada para simulação DC.

Feito isso, caso tenha sido escolhido o método de simulação TRAP, na linha .TRAN do netlist, descrita na seção seguinte, é feita a análise no tempo através do método dos trapézios. Por fim, os resultados são escritos em um arquivo de texto, no qual a primeira linha especifica a variável correspondente a cada coluna e as demais linhas, os valores. As colunas são separadas por espaços.

**3. Formato do netlist**

O netlist do circuito a ser analisado contém uma linha de título, além dos seguintes elementos:

Especificação de análise transiente:

.TRAN <tempo final> <passo> <método> <passos por ponto>

Comentários: \*<comentário>

Resistor: R<nome> <valor>

Indutor: L<nome> <valor>

Capacitor: C<nome> <valor>

Fontes controladas: <Tipo da fonte><nome> <saída+> <saída-> <entrada+> <entrada->

Fonte de tensão controlada a tensão: E

Fonte de corrente controlada a corrente: F

Fonte de corrente controlada a tensão: G

Fonte de tensão controlada a corrente: H

Fonte de corrente: I<nome> <nó a> <nó b> <valor>

Fonte de tensão: V<nome> <nó a> <nó b> <valor>

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

Diodo ideal: D<nome> <nó a> <nó b>

Chave ideal: $<nome> <nó a> <nó b> <nó controle+> <nó controle-> <Vlimite>

Transformador ideal: K<nome> <nó in+> <nó in-> <nó out+> <nó out-> <ganho>

**4. Entendimento detalhado do programa**

Para o funcionamento do programa, utilizamos as funções citadas abaixo, seguidas da explicação de seu funcionamento:

**resolverSistema**

Função para a resolução de sistema de equações lineares. Utiliza o método de Gauss-Jordan com condensação pivotal.

**numero**

Rotina que conta os nós do sistema a ser resolvido e atribui números a eles.

**leituraNetlist**

Realiza a leitura da netlist entregue e organiza as informações para a posterior resolução do sistema.

**adicionarVariaveis**

Define variáveis adicionais relativas às correntes que precisam ser calculadas no sistema.

**listarVariaveis**

Expõe as variáveis, relacionando seus nomes aos seus números no sistema.

**mostrarNetlist**

Organiza a netlist com os dados e as variáveis do sistema.

**montarSistemaDC**

Nesta função, identificamos os tipos dos elementos e montamos as estampas para cada elemento linear do sistema que não varia com o tempo, combinando-as.

**mostrarSistema**

Função opcional: Mostra o sistema resolvido.

**adicionarEstampasComponentesVariantes**

Adiciona as estampas para cada um dos componentes com estampas que variam no tempo presentes no sistema.

**resolverNewtonRaphson**

Função que executa o algoritmo de Newton Raphson para resolução do circuito, adicionando as estampas dos componentes não lineares antes de resolver o sistema.

**simulacaoTrapezios**

Esta função monta o sistema DC. A cada iteração, adiciona estampas dos componentes variantes no tempo, resolve o sistema através do método de Newton Raphson e guarda a solução da iteração na linha de uma matriz.

**resolverPontoOperacao**

Calcula o ponto de operação do sistema a ser resolvido de acordo com os componentes presentes.

**converterExtensao**

Adiciona a extensão correta ao arquivo produto.

**escreverResultadosNoArquivo**

Função que grava os resultados em um arquivo de texto com nome definido de acordo com a função converterExtensao. Ex: netlist: circuit.net, arquivo: circuit.res.

**5. Exemplos**