Circuitos Elétricos II
Trabalho III - 2023.2

Universidade Federal do Rio de Janeiro
Departamento de Engenharia Eletrônica e de Computação
Professora: Fernanda Oliveira

Enunciado do terceiro trabalho de simulação de circuitos elétricos. Para o trabalho, os alunos devem desenvolver um programa em Python (versão 3 ou maior). Os pacotes Numpy e Scipy são permitidos. O Matplotlib pode ser usado para testar o trabalho, mas a chamada da função do Matplotlib não deve ser incluída dentro da main. Para utilizar outros pacotes, converse com a professora. O trabalho é individual. Por favor não enviem ou recebam códigos de outros alunos.

Instruções:

1. O trabalho consiste no desenvolvimento de um programa em Python para o cálculo da reposta DC ou transiente de um circuito utilizando a análise nodal modificada, o método dos trapézios e o método de Newton Raphson para componentes não lineares (diodos).

2. Para esse trabalho, modifique o código do trabalho 2 de forma que a função main calcule e retorne arrays que representam as tensões nodais DC ou ao longo de um intervalo de tempo definido por argumentos da função. Para a montagem do sistema deve ser utilizada a análise nodal modificada. A análise no tempo deve ser realizada utilizando o método dos trapézios com passo fixo. O método de Newton-Raphson (NR) deve ser utilizado para a análise de circuitos com componentes não-lineares.

3. O trabalho pode ser escrito em um ou mais arquivos ".py" (veja, abaixo, a padronização dos nomes dos arquivos) e pode usar funções ou classes, de acordo com a preferência do aluno, mas deve, necessariamente, conter uma função principal cujo nome é main. A função main recebe quatro valores por argumento, na seguinte ordem: uma string com o nome do arquivo; uma string igual a "DC" ou "TRAN"; uma lista de números inteiros que representam os nós desejados; uma lista com parâmetros de simulação. Para a simulação "DC" a lista de parâmetros contém: um float que representa a tolerância do NR; uma lista de valores iniciais para cada nó da netlist que devem ser usados como aproximação inicial do NR. No caso de uma netlist que não contenha componentes não lineares, esses valores devem ser ignorados. Para tais netlists, a tolerância será zero e a lista possuirá somente números iguais a zero. Para a simulação "TRAN" a lista de parâmetros contém: um float que representa o tempo total de simulação em segundos (a simulação deve começar de 0 s e ir até o tempo total); um float que representa o passo, também em segundos; um float que representa a tolerância do Newton-Raphson; uma lista de valores iniciais para cada nó da netlist. A função deve ser definida da seguinte forma: def main(arqNetlist, tipo, nos, parametros), onde parametros é no formato: [tol, v0] para a simulação "DC"; ou [tSim, passo, tol, v0] para a simulação "AC". Dessa forma, ao fazer a chamada da função dentro de um ".py" os valores devem ser definidos na chamada, ex.: main('netlist1.txt', 'DC', [1,2], [1e-9, [0,1,0]]) ou main('netlist1.txt', 'TRAN', [1,2], [1e-3, 10e-6, 1e-9, [0,1,0]]). Reforçando, a função será chamada por um arquivo ".py" e não diretamente pelo terminal.

4. No caso de circuitos com componentes não lineares, a análise "DC" ou o cálculo do ponto de operação deve considerar como aproximação inicial as tensões nodais passadas por argumento. Para o cálculo "DC" ou cálculo do ponto de operação, os capacitores devem ser considerados como circuitos abertos, os indutores como curto-circuito, as fontes de tensão e corrente senoidais devem ter seus valores modificados para zero. No caso da análise transiente, as tensões encontradas no cálculo do ponto de operação são as tensões no instante $t = 0$. Estas tensões devem ser utilizadas como aproximação inicial para o primeiro instante de tempo diferente de zero. Para os demais instantes de tempo, o valor encontrado no instante de tempo anterior sempre deve ser usado como aproximação inicial.

5. Para a simulação "DC", a função main deve retornar um array com as tensões nodais referente aos nós indicados por argumento.

6. Para a simulação "TRAN", a função main deve retornar dois arrays. O primeiro deve conter os instantes de tempo utilizados na simulação (definidos de acordo com o tempo de simulação e com o passo). E segundo array retornado pela função main deve conter uma quantidade de linhas igual ao tamanho da lista com os nós desejados (terceiro argumento), e a quantidade de colunas igual à quantidade de instantes de tempo utilizados na simulação. Considere que a lista de nós desejados sempre estará na ordem crescente, que os nós indicados na lista existirão na netlist e que os valores da lista serão inteiros maiores que zero.

7. O arquivo ".py" que contém a função main deve ter o nome seguindo o seguinte padrão: trab3nomesobrenome.py , onde nome deve ser substituído pelo seu primeiro nome e sobrenome pelo seu último sobrenome. Utilize somente minúsculas e não utilize caracteres especiais. Por exemplo: trab3fernandaoliveira.py

8. Para garantir que não haverá arquivos de alunos diferentes com o mesmo nome, os nomes de todos demais arquivos ".py" utilizados (caso o aluno utilize mais de um arquivo ".py" além daquele que contém a função main) devem conter as iniciais do aluno.

9. Utilize o modelo exponencial do diodo. Para facilitar a convergência limite a tensão direta sobre o diodo em 1 V e a tensão reversa em 20 V. Isto é, se a tensão do anodo menos a tensão do catodo for maior que 1 V, ela deve ser definida igual a 1 V. Por outro lado, se a tensão do anodo menos a tensão do catodo for menor que -20 V, ela deve ser definida igual a -20 V.

10. A análise AC do trabalho 2 pode ser deixada no código, de forma que a string "AC" também seja aceita.

Cada linha da netlist indica um novo componente e as informações necessárias sobre ele. A netlist pode conter comentários, que devem ser ignorados pelo programa, conforme indicado abaixo. Para resistores, fontes independentes ou controladas, capacitores, indutores, transformadores e diodo, o seguinte padrão deve ser seguido:

- Comentários, linhas que não são usadas pelo programa: *<comentário>
- Resistor: R<identificação> <nó1> <nó2> <valor da resistência>
- Fonte de corrente controlada por tensão: G<identificação> <nóI(fonte drena a corrente desse nó)> <nóI(fonte injeta a corrente nesse nó)> <nóv(positivo)> <nóv(negativo)> <valor da transcondutância Gm>
- Indutor: L<identificação> <nó1> <nó2> <valor da indutância>

- Capacitor: C<identificação> <nó1> <nó2> <valor da capacitância>
- Transformador: K<identificação> <nó a> <nó b> <nó c> <nó d> <valor da indutância no primeiro enrolamento> <valor da indutância no segundo enrolamento> <valor da indutância mútua>
- Fonte de corrente controlada por corrente, considerando que a corrente de controle passa por um curto do nó C ao nó D: F<identificação> <nóI(fonte drena a corrente desse nó)> <nóI(fonte injeta a corrente nesse nó)> <nó de controle C> <nó de controle D> <valor do ganho de corrente B>
- Fonte de tensão controlada por tensão: E<identificação> <nó positivo> <nó negativo> <nó de controle positivo> < nó de controle negativo> <valor do ganho de tensão A>
- Fonte de tensão controlada por corrente, considerando que a corrente de controle passa por um curto do nó C ao nó D: H<identificação> <nó positivo> <nó negativo> <nó de controle C> <nó de controle D> <valor da transresistência Rm>
- Fonte de corrente DC: I<identificação> <nó cuja corrente é drenada pela fonte> <nó cuja corrente é injetada pela fonte> DC <valor>
- Fonte de tensão DC: V<identificação> <nó positivo> <nó negativo> DC <valor>
- Fonte de corrente senoidal considerando a expressão i(t) = <amplitude> * cos (2*π*<frequência>*t + <fase>*π/180): I<identificação> <nó cuja corrente é drenada pela fonte> <nó cuja corrente é injetada pela fonte> SIN <valor DC> <amplitude> <frequência em Hz> <fase em graus>
- Fonte de tensão senoidal considerando a expressão v(t) = <amplitude> * cos (2*π*<frequência>*t + <fase>*π/180): V<identificação> <nó positivo> <nó negativo> SIN <valor DC> <amplitude> <frequência em Hz> <fase em graus>
- Diodo: D<identificação> <nó positivo> <nó negativo> <I_s > <ηV_T>

Tudo o que está entre os caracteres < e > deve ser substituído na netlist e os caracteres < e > em si também não aparecem na netlist.

Observação: um mesmo circuito pode ter fontes AC e DC. Em uma simulação DC as fontes AC devem ser consideradas iguais a zero; em uma simulação AC as fontes DC devem ser consideradas iguais a zero.