

Universidade Federal do Rio de Janeiro
(UFRJ)

Escola Politécnica
Engenharia Eletrônica e de Computação

Simulador de Circuito de Análise Transiente

Circuitos Elétricos II - 2014/2
Prof. Antônio Carlos Moreirão de Queiroz

Dhiana Deva
Felipe De Leo
Silvino Vieira

Sumário

- 1 - Objetivo**
- 2 - Desenvolvimento**
- 3 - Elementos Polinomiais**
- 4 - Circuitos de Exemplo**
- 5 - Conclusão**

1 - Objetivo

O objetivo desse trabalho é elaborar um simulador de circuitos elétricos capaz de analisar circuitos no tempo, e simular pelo menos os seguintes elementos listados na tabela abaixo:

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

A análise deverá ser feita usando análise nodal modificada, com correntes em indutores calculadas.

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.

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 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.

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

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).

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.

Formato para o Netlist:

- **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>

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 x0> <amplitude A> <frequência f (Hz)> <atraso t0> <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 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.

2 - Desenvolvimento

Utilizamos diferentes técnicas, especificações e ferramentas para o desenvolvimento do nosso simulador.

Dentre as especificações técnicas principais do programa estão:

Linguagem de Programação	C++
Sistemas Operacionais (testados em operação)	<ul style="list-style-type: none">• Linux• Windows 7• Windows 8
Ambientes de Desenvolvimento (Com seus respectivos compiladores)	<ul style="list-style-type: none">• Visual Studio 2013 Ultimate• Visual Studio 2013 Express• Qt• Clang / G++

Além disso utilizamos algumas ferramentas auxiliares:

Github	<ul style="list-style-type: none">• Repositório para versionamento do programa• Sincronização com diversos dispositivos e membros da equipe• Comentários em mudanças no código• Commits - Envio de atualizações no código a cada sucesso• Monitoramento do andamento do trabalho com realces a cada mudança no código• Possibilidade de Integração com outras ferramentas
Travis CI (Plataforma de Integração Contínua)	<ul style="list-style-type: none">• Permite a execução de testes automatizados conforme o software está sendo desenvolvido.• Notifica qualquer problema de Build detectado nos testes.• Integrado diretamente com o GitHub
Google Test	Framework do Google para rodar testes. Com ele, nós podemos especificar nossos

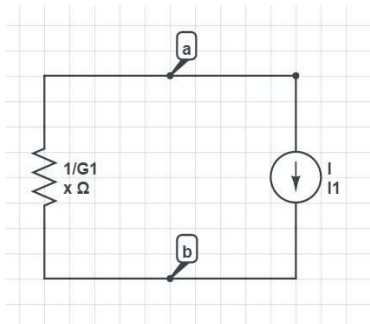
	testes a serem realizados.
--	----------------------------

3 - Elementos Polinomiais

Resistor Polinomial

Estampa do resistor polinomial para método de Newton-Raphson:

$j = f(v) = a_0 + a_1 v + a_2 v^2 + a_3 v^3 + a_4 v^4 + a_5 v^5 + a_6 v^6 + a_7 v^7$
$G_1 = f'(v) = a_1 + 2a_2 v + 3a_3 v^2 + 4a_4 v^3 + 5a_5 v^4 + 6a_6 v^5 + 7a_7 v^6$
$I_1 = a_0 - a_2 v^2 - 2a_3 v^3 - 3a_4 v^4 - 4a_5 v^5 - 5a_6 v^6 - 6a_7 v^7$

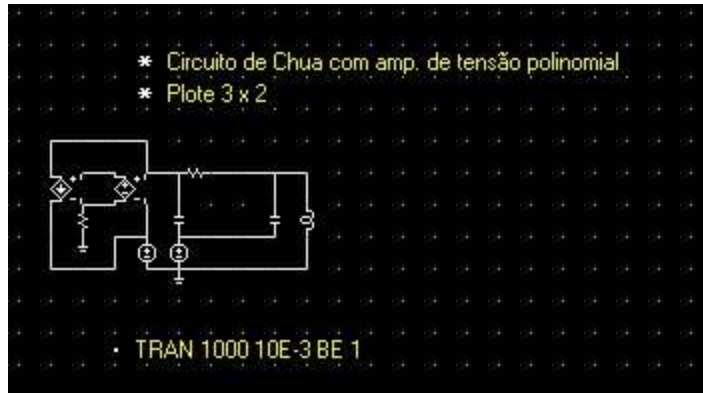


$$\begin{bmatrix} +G_1 & -G_1 \\ -G_1 & +G_1 \end{bmatrix} \begin{bmatrix} e_1 \\ e_{2n+1} \end{bmatrix} = \begin{bmatrix} -I_1 \\ +I_1 \end{bmatrix}$$

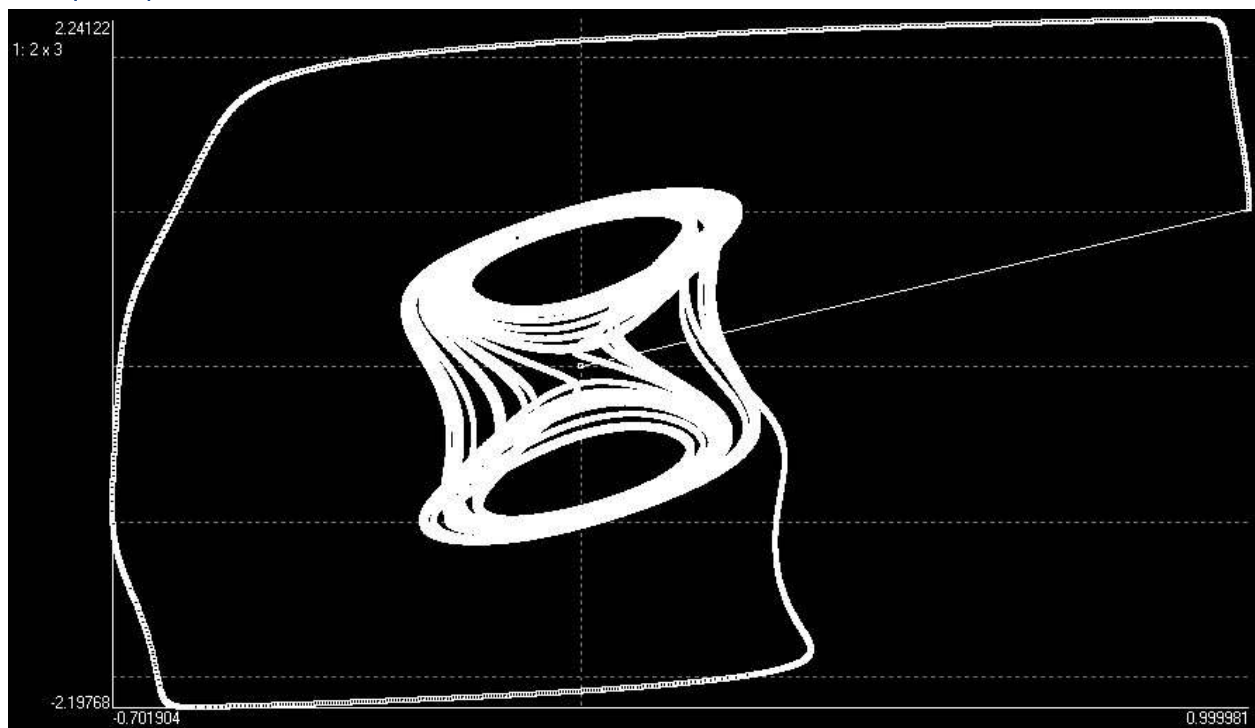
As estampas dos demais elementos polinomiais estão explícitas nos arquivos de teste e podem ser visualizadas no resultado dos testes em anexo.

4 - Circuitos de Exemplo

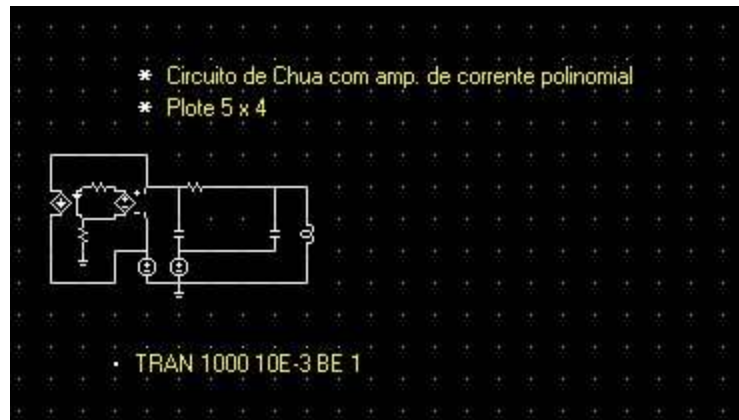
1 - Chuapole.net



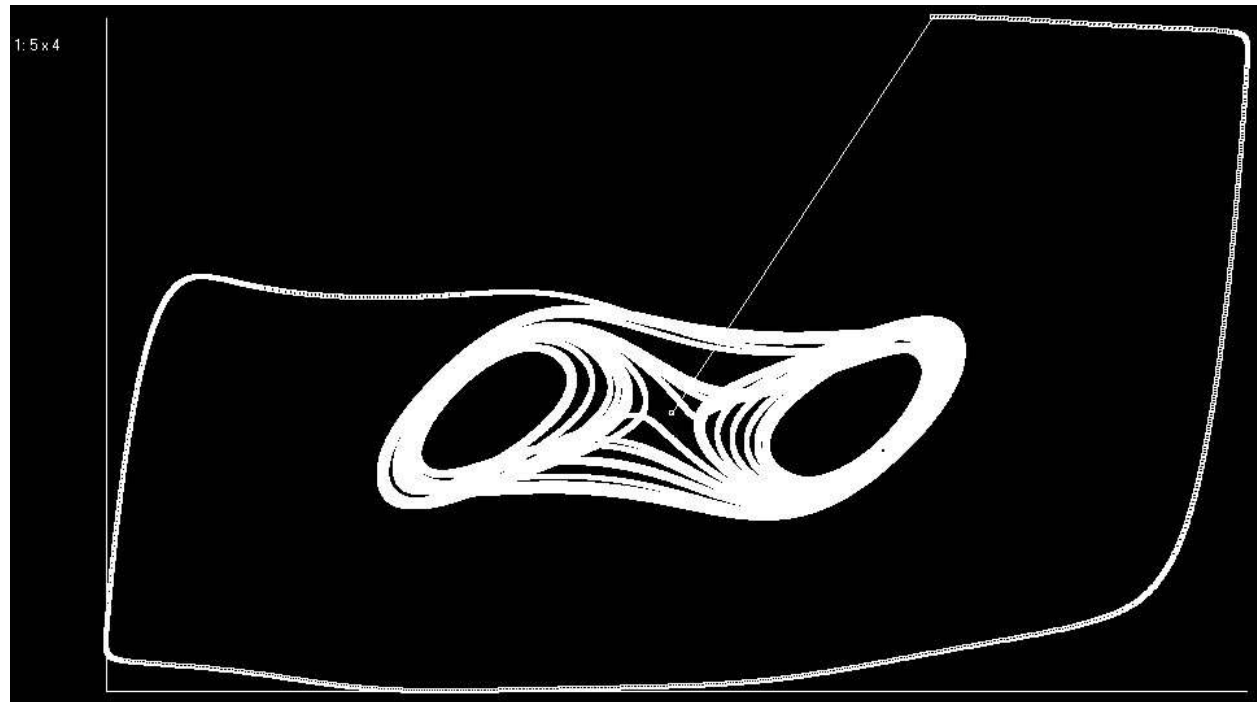
Plot (2 x 3)



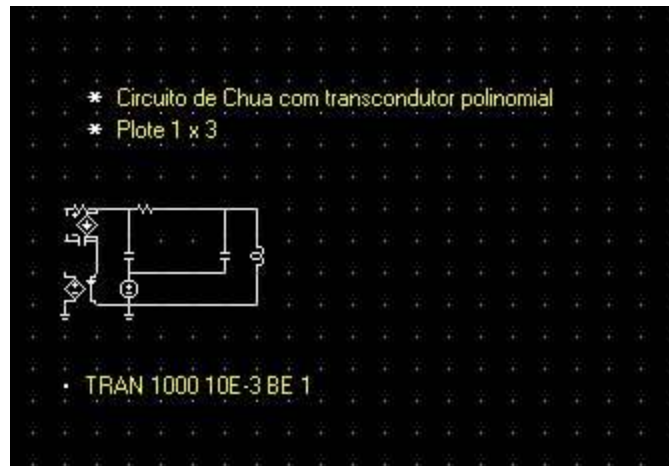
2 - Chuapolf.net



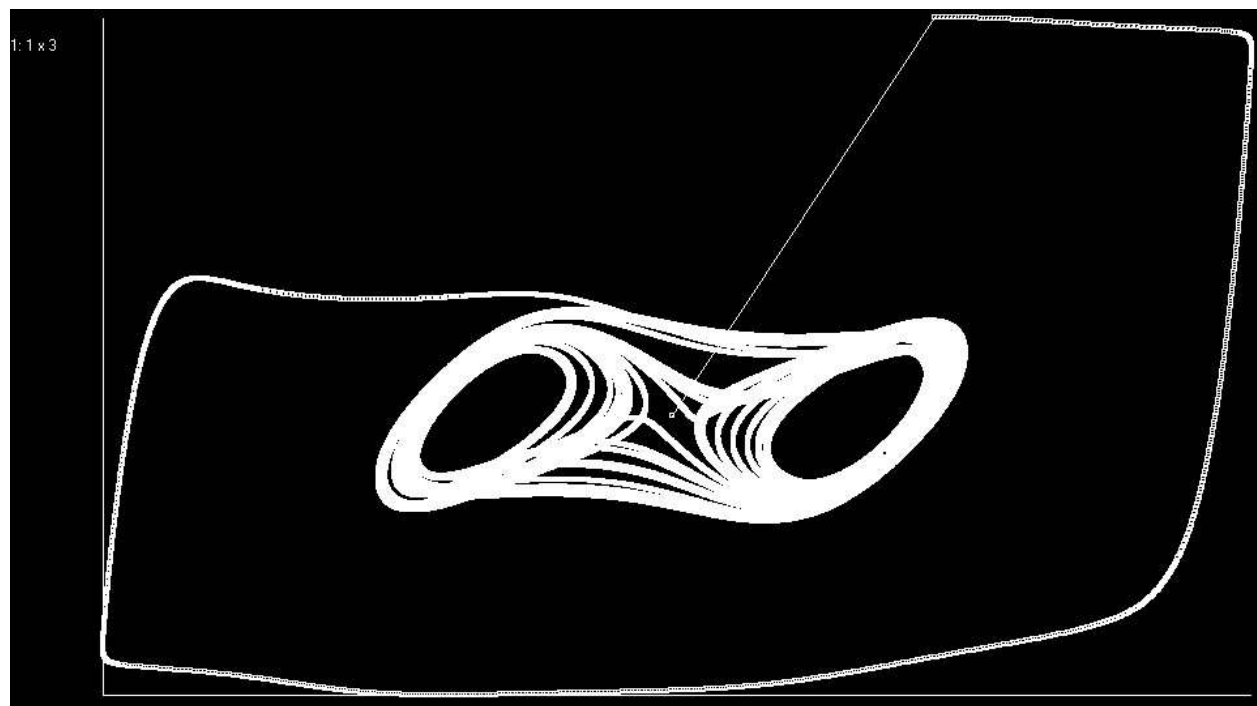
Plot (5 x 4)



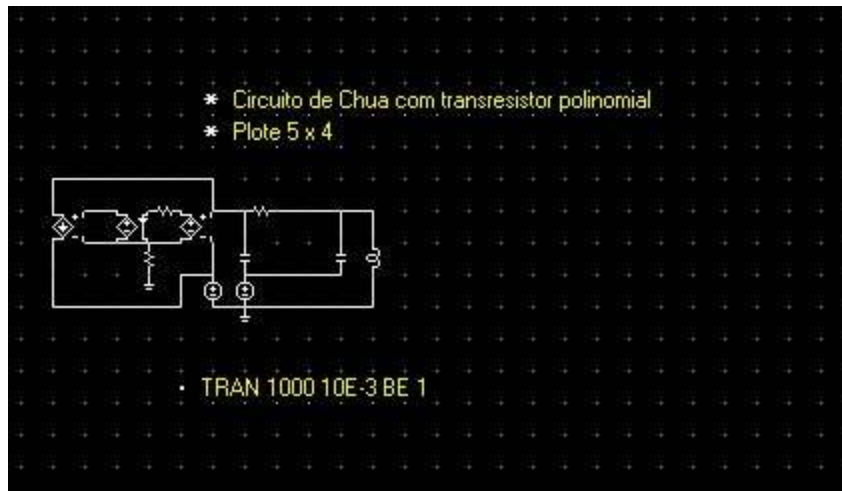
3 - Chuapolg.net



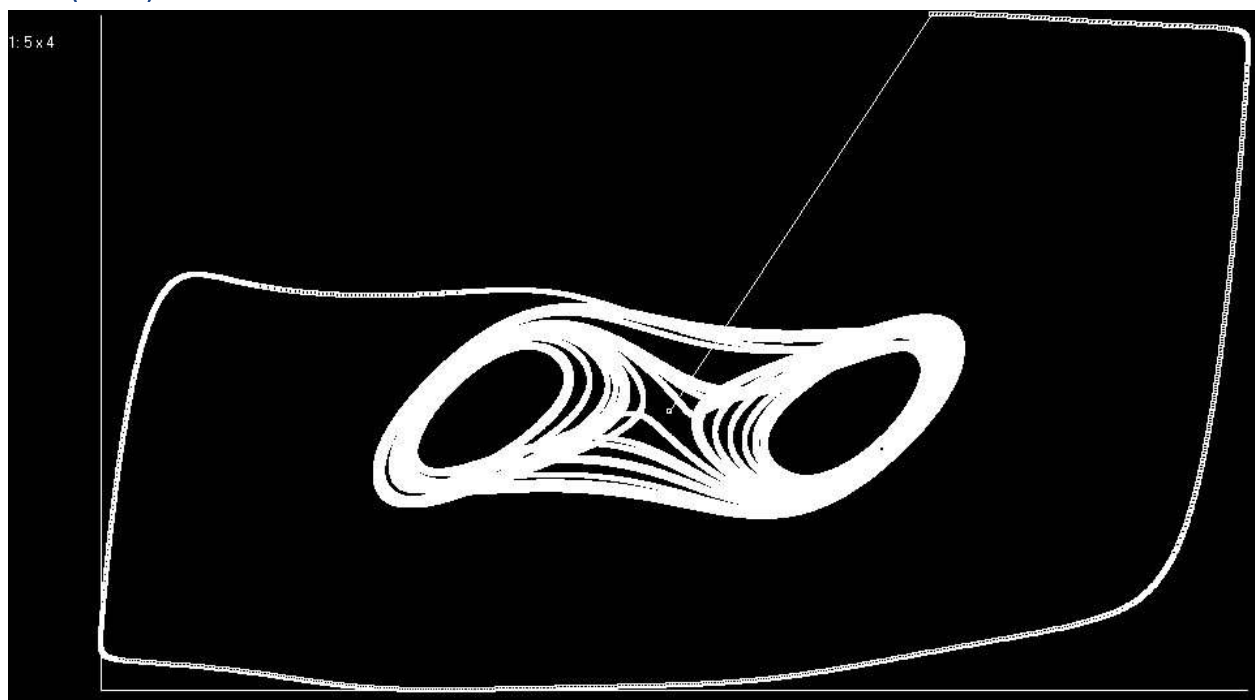
Plot (1 x 3)



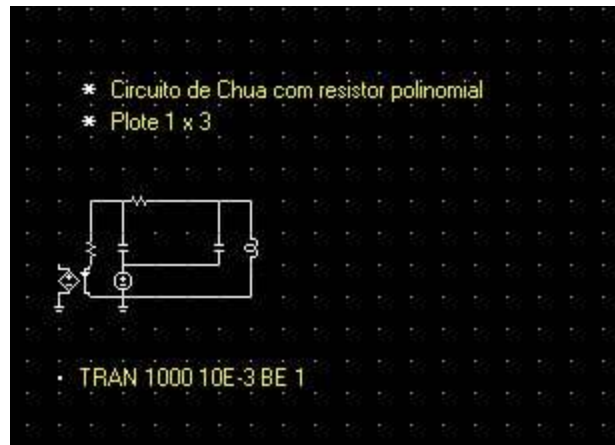
4 - Chuapolh.net



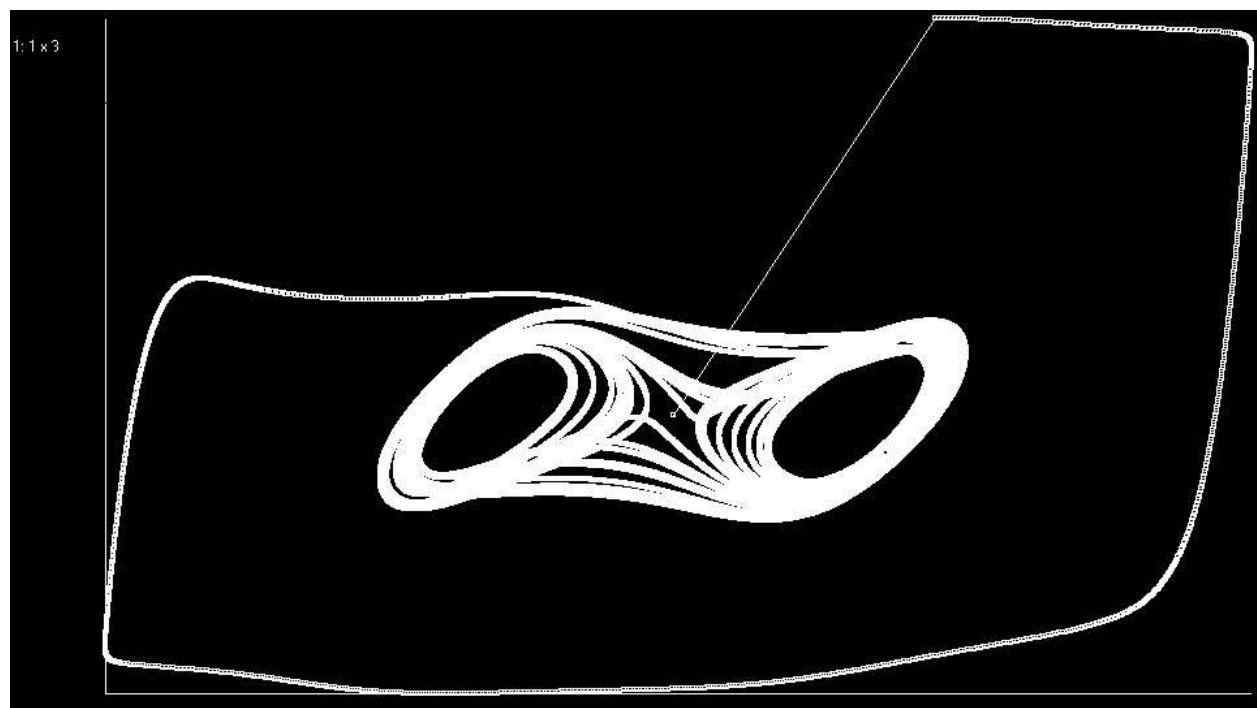
Plot (5 x 4)



5 - Chuapolr.net



Plot (1 x 3)



6 - Diodop1.net



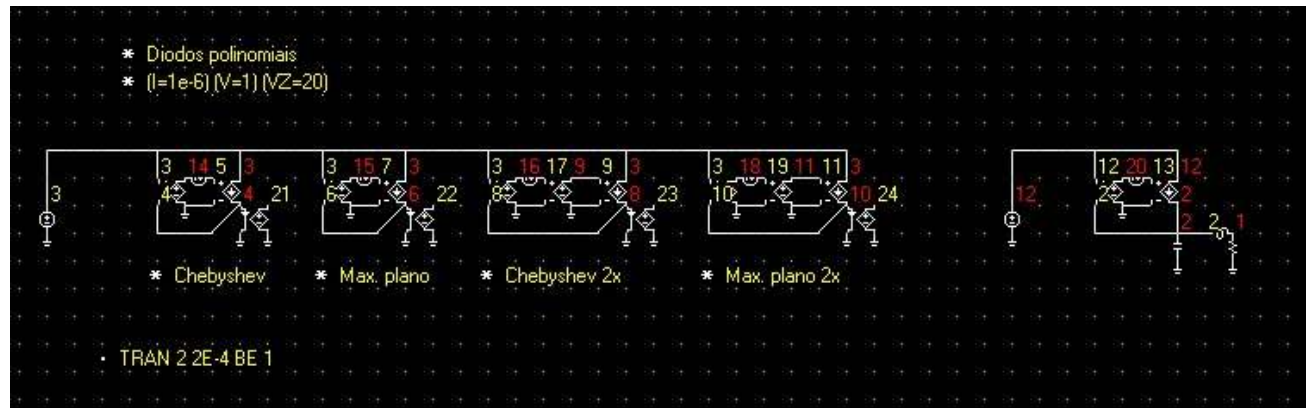
Plot (t x jyH1300)



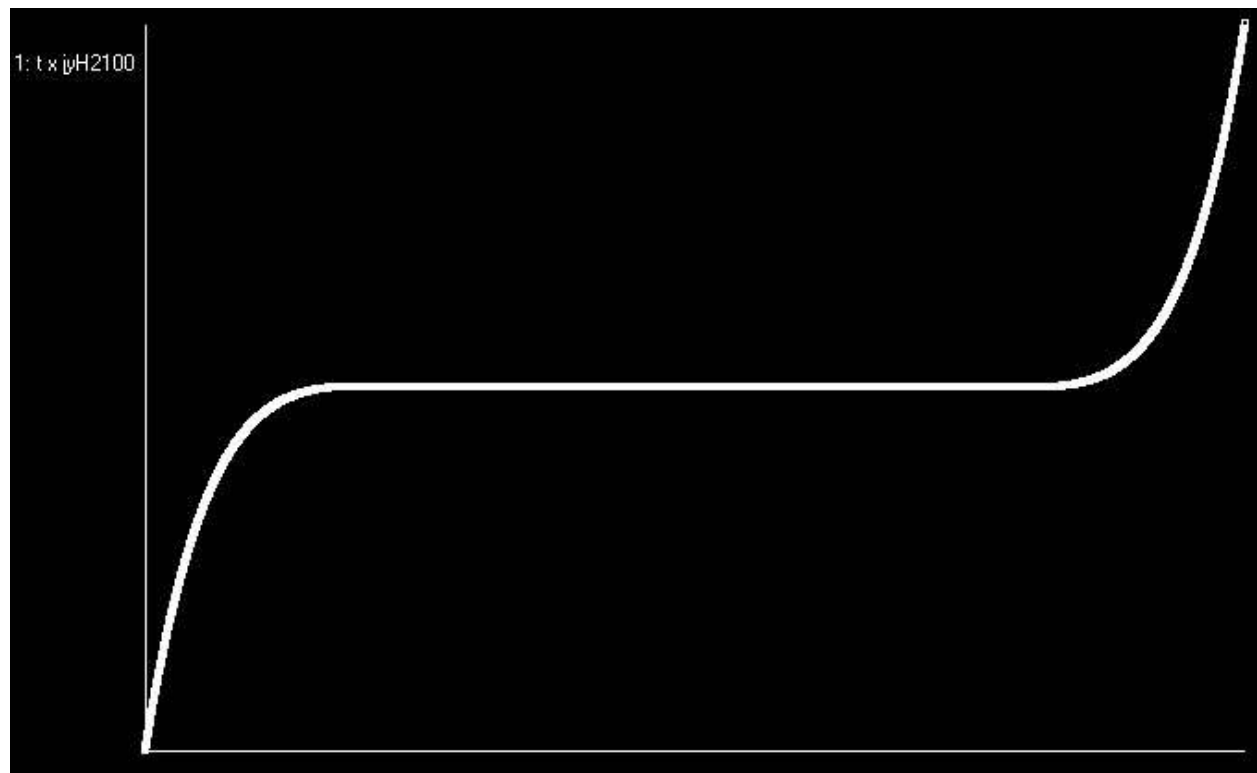
Plot (t x jyH1400)



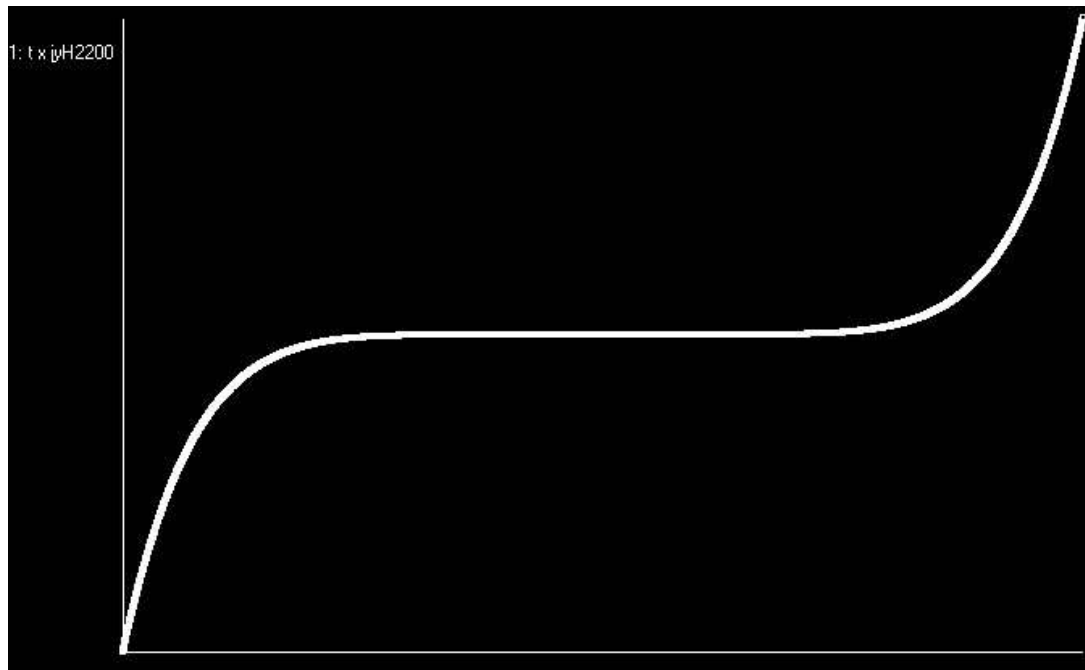
7 - Diodop2.net



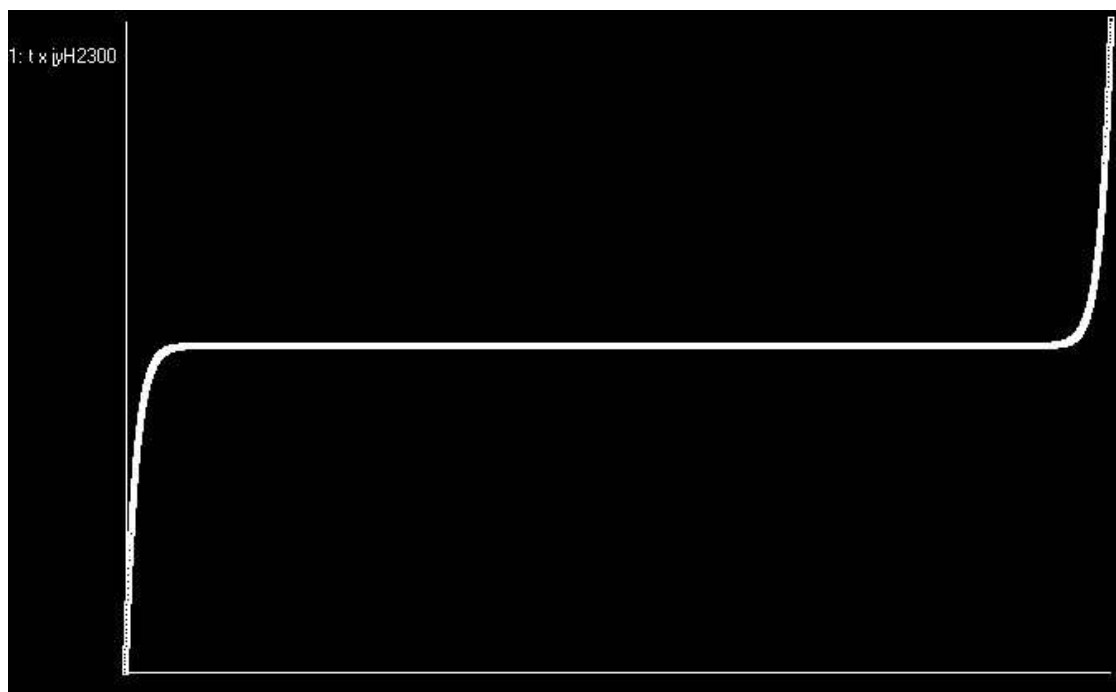
Plot (t x jyH2100)



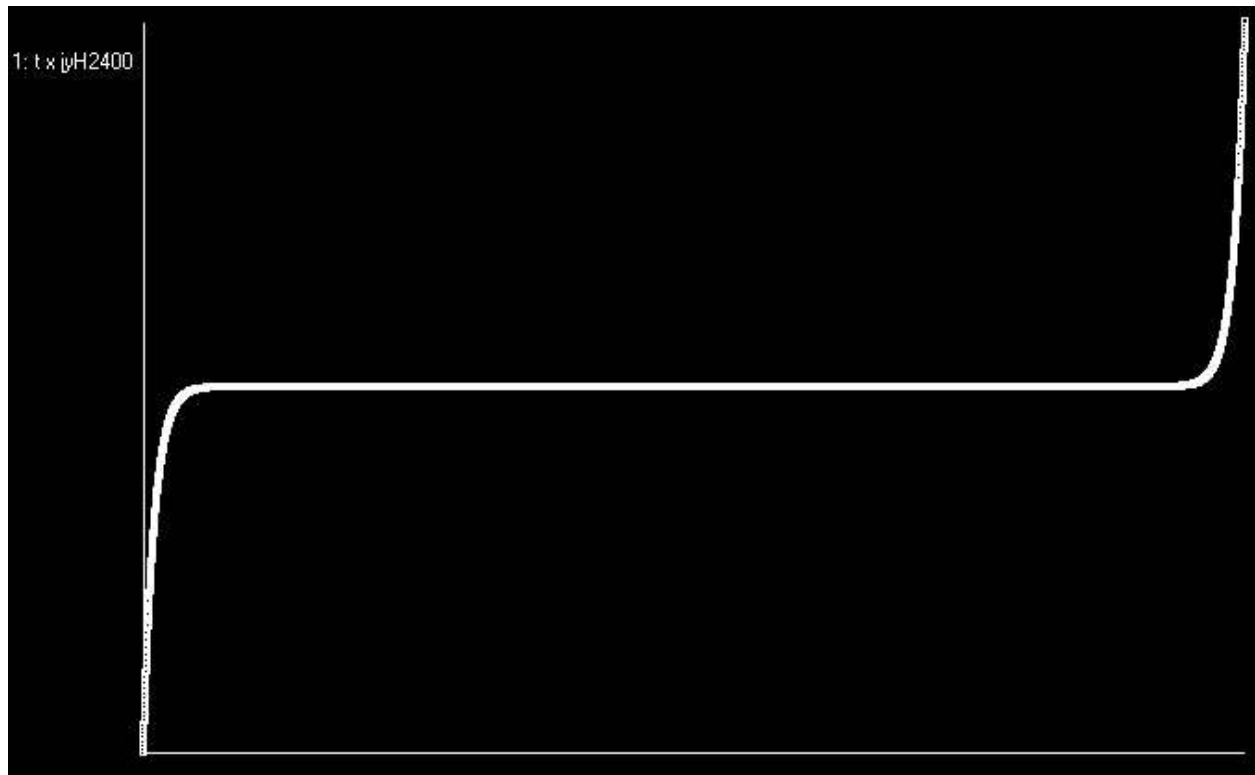
Plot (t x jyH2200)



Plot (t x jyH2300)

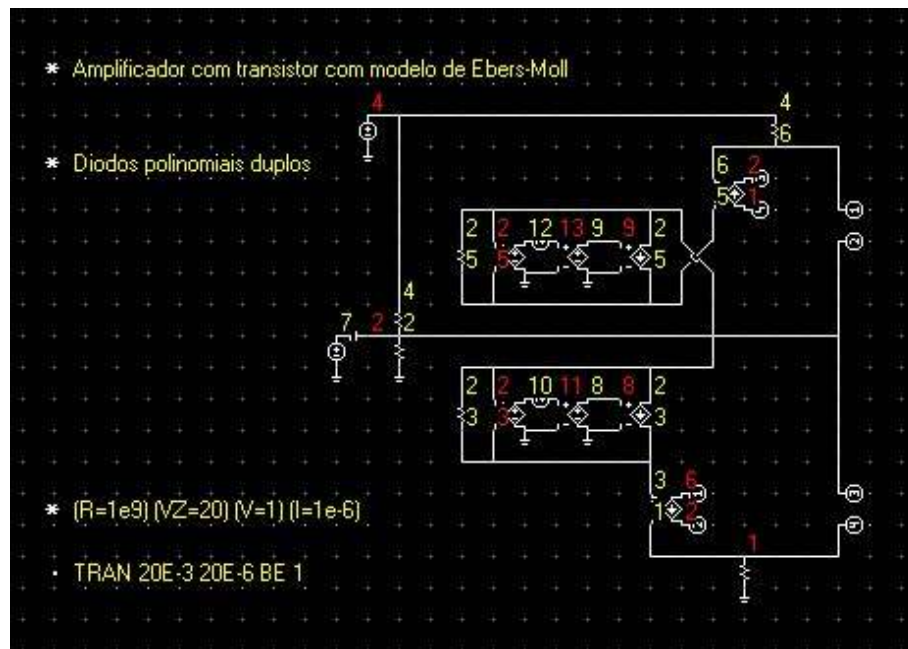


Plote (t x jyH2400)

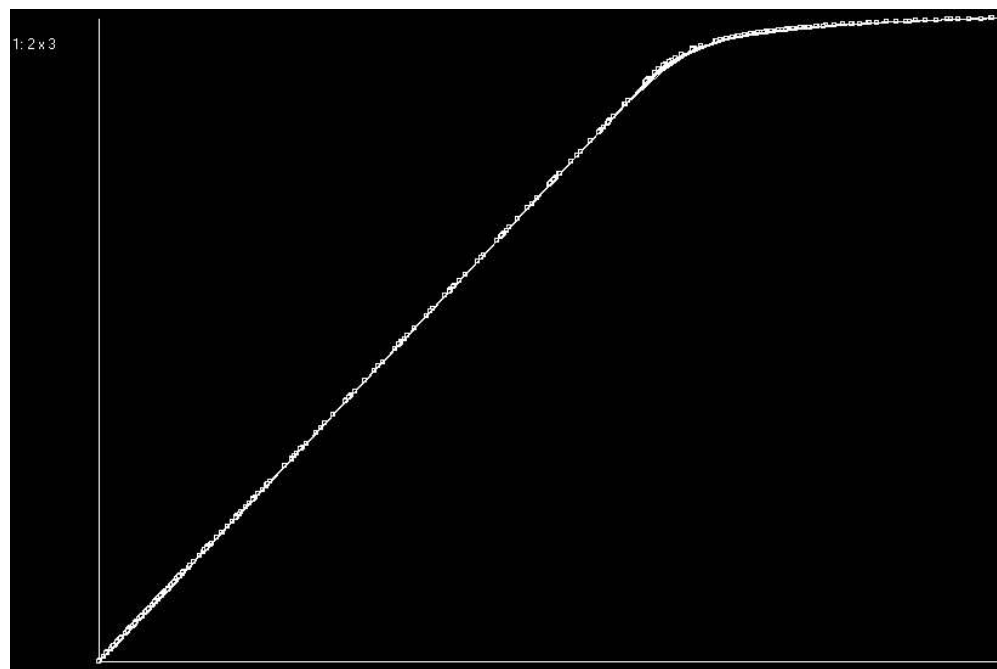


8 - Ebersmoll.net

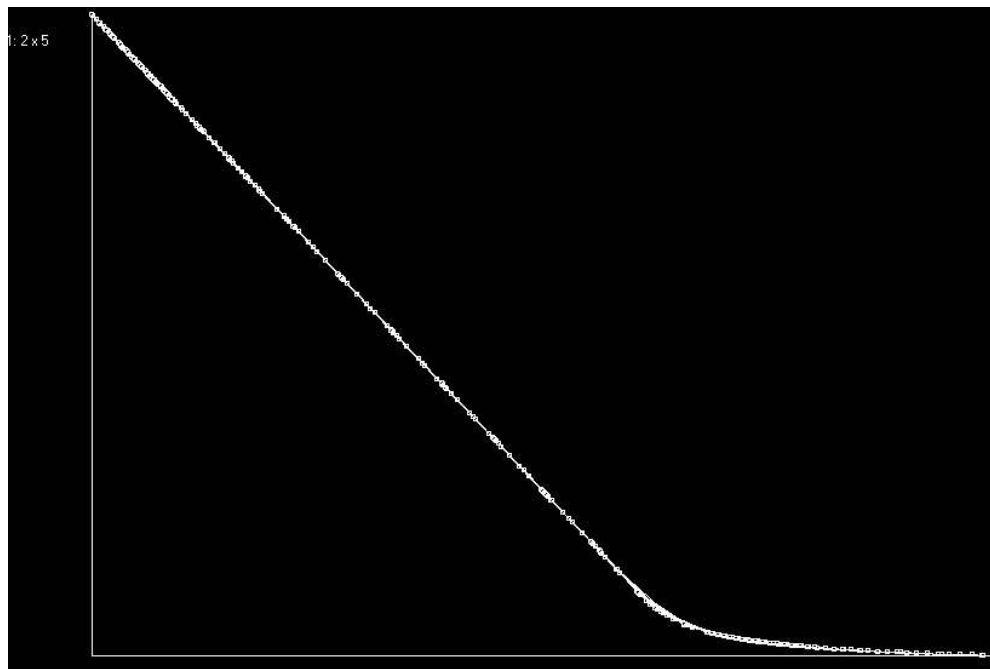
Circuito



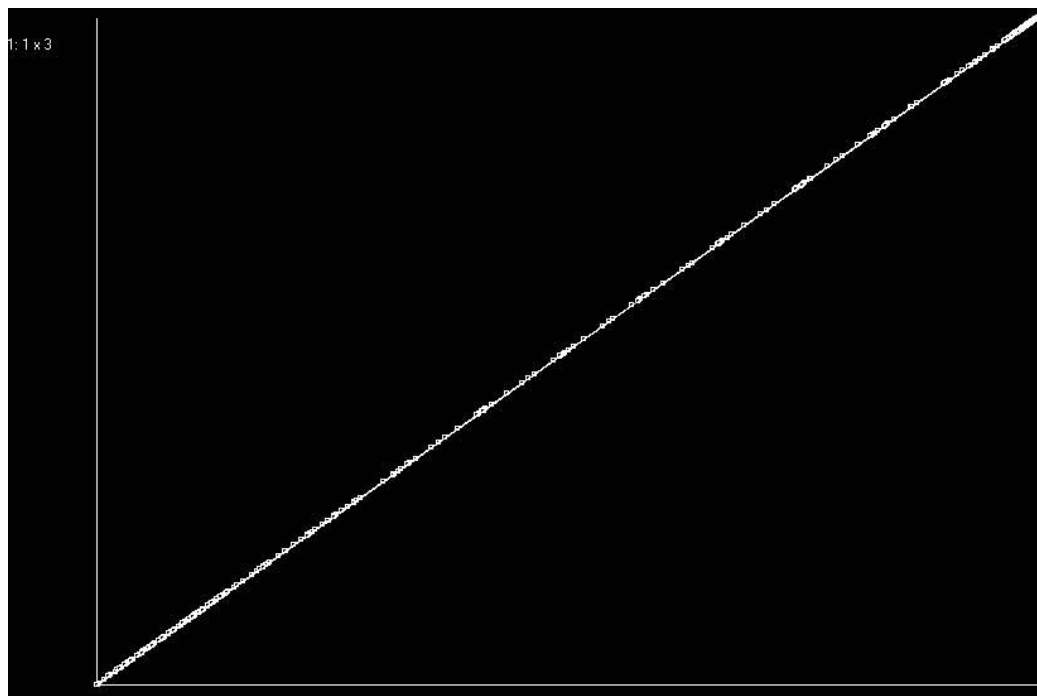
Plot (2 x 3)



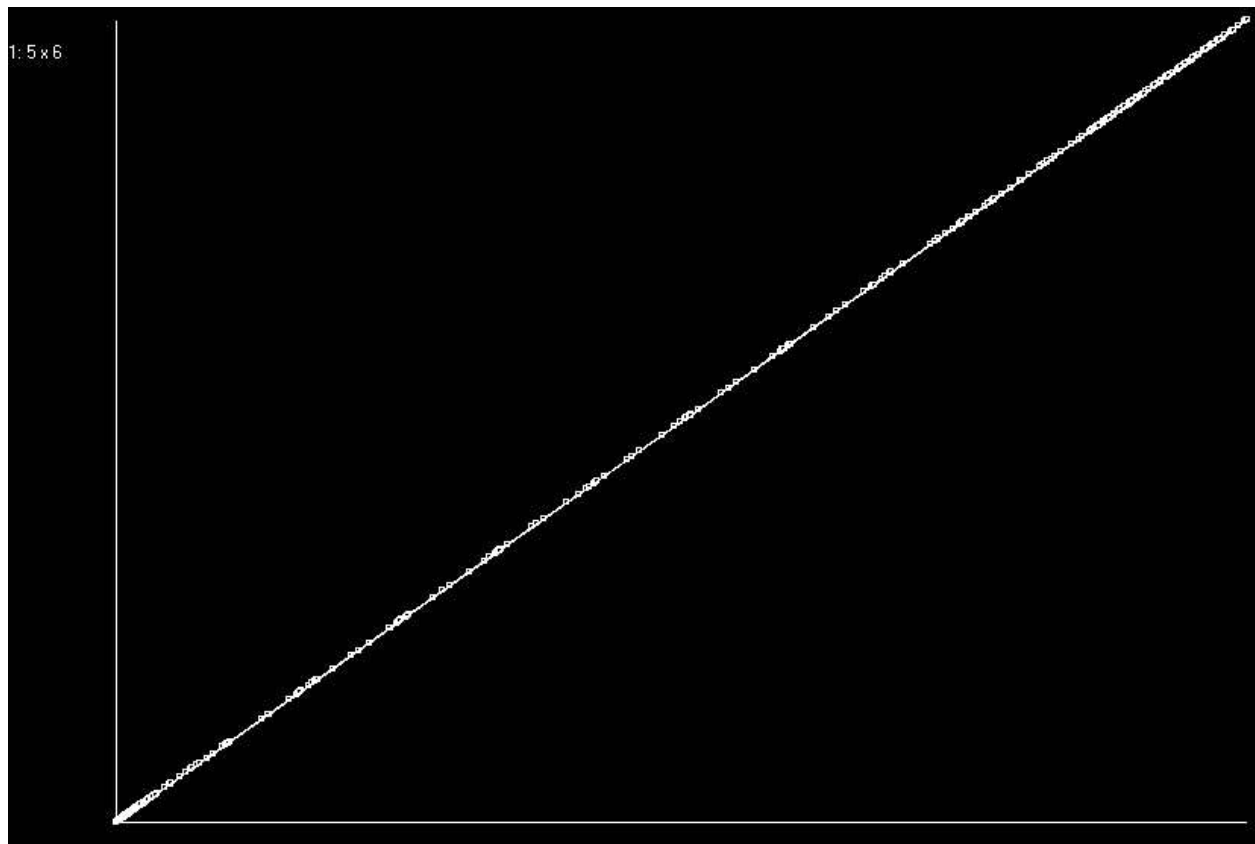
Plot (2 x 5)



Plot (1 x 3)

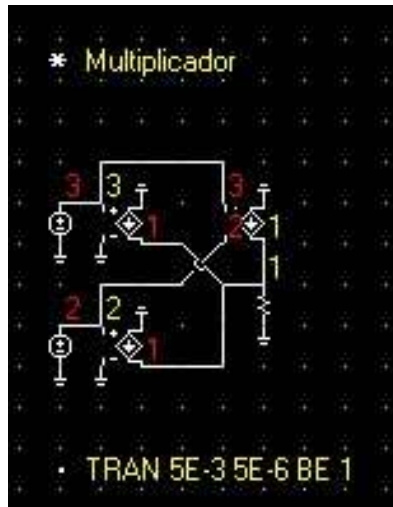


Plot (5 x 6)

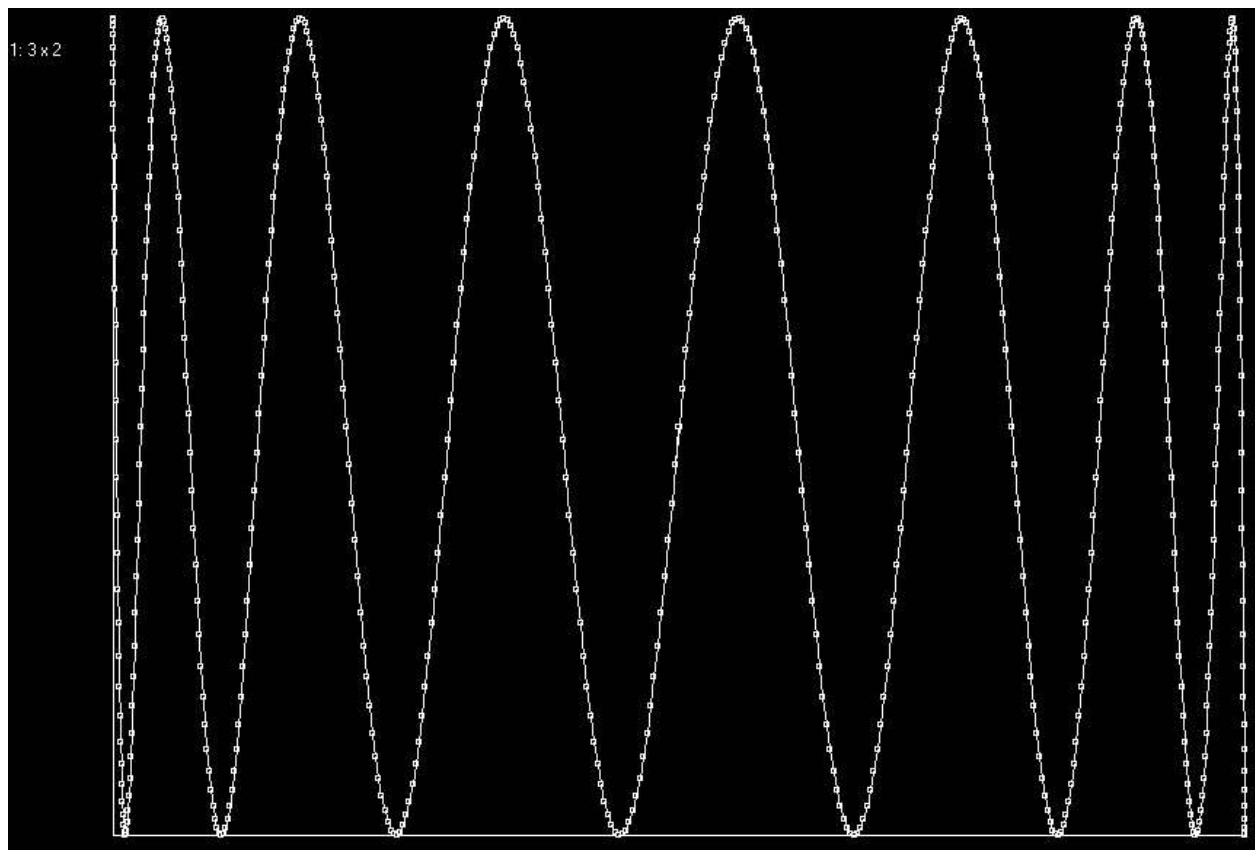


9 - Mult.net

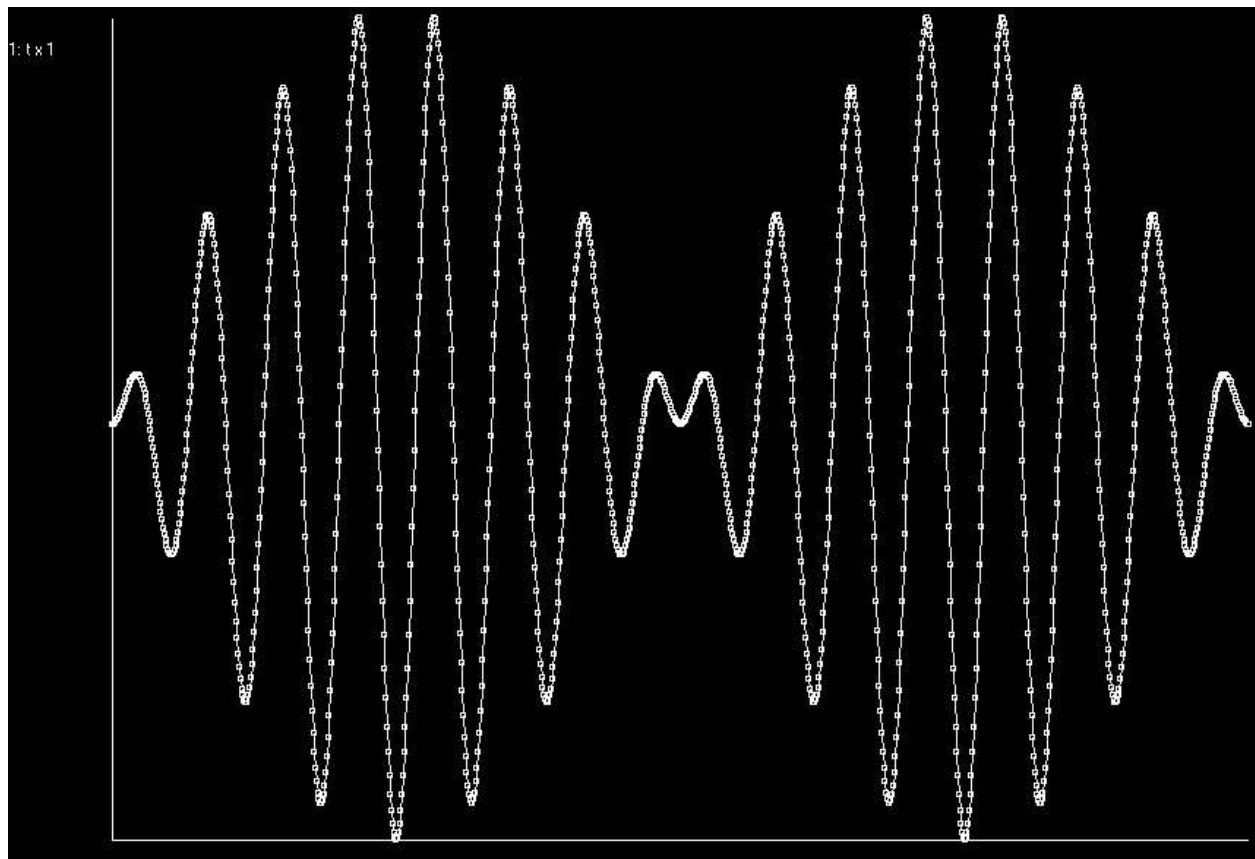
Circuito



Plot (3 x 2)

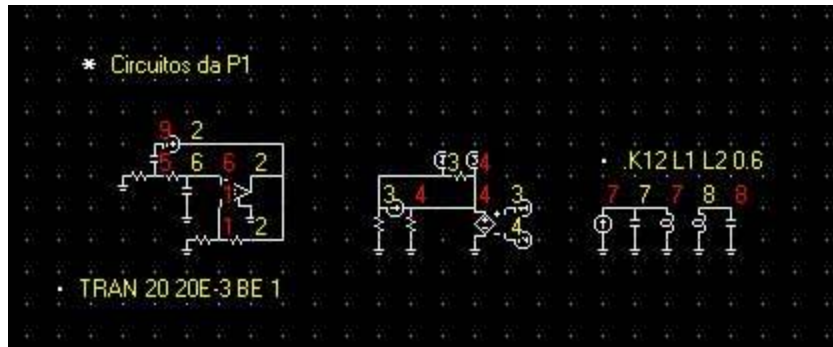


Plot (t x 1)

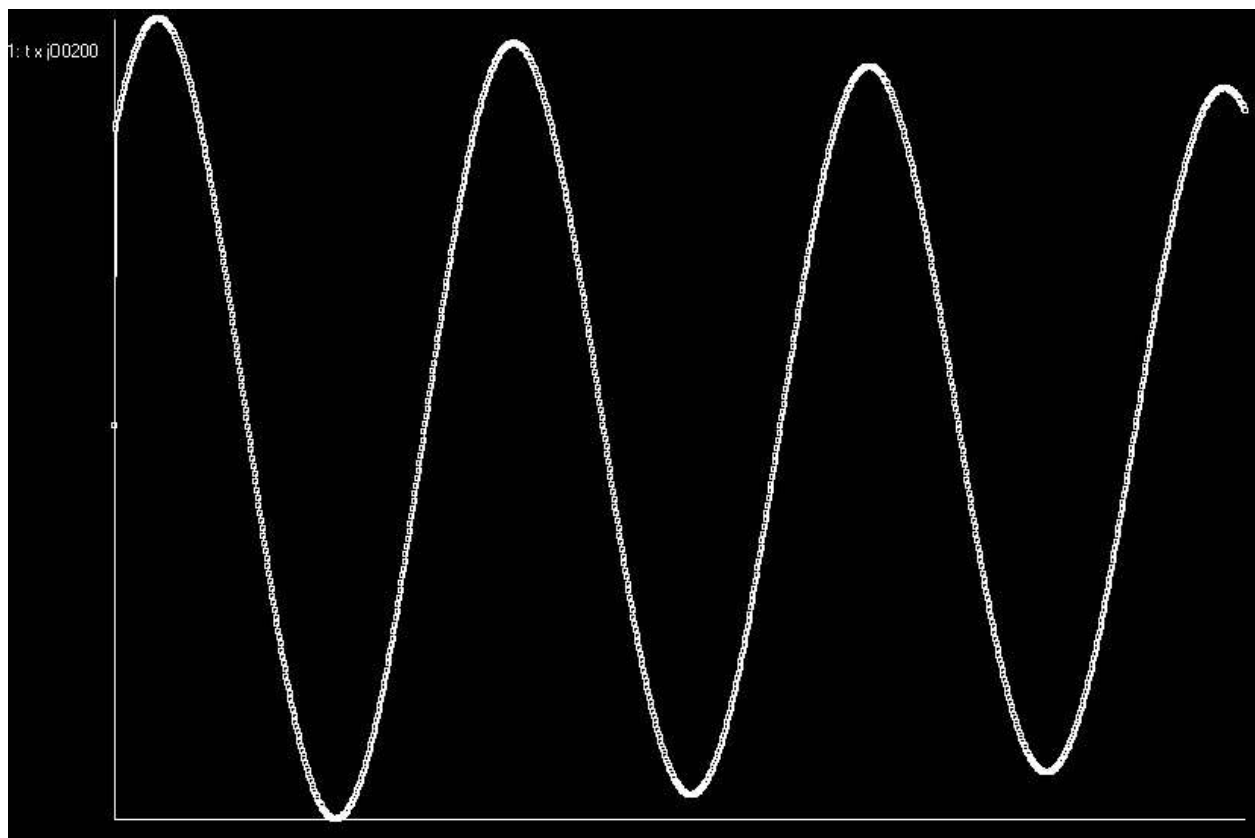


10 - Prova.net

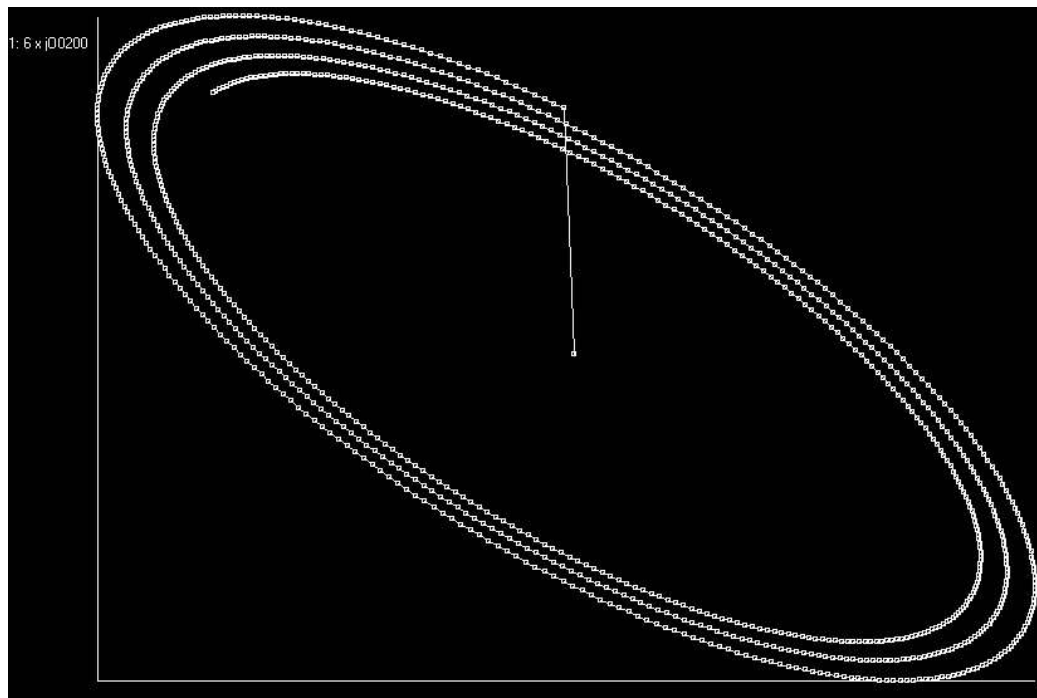
Circuito



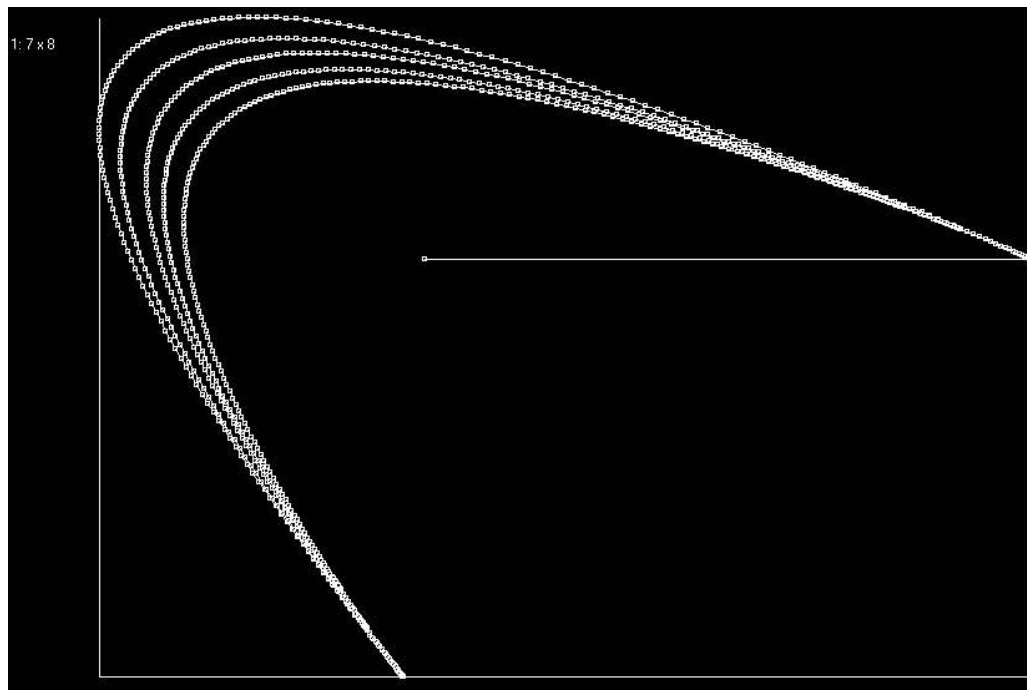
1) Plot (j00200) - Corrente na entrada do AMPOP



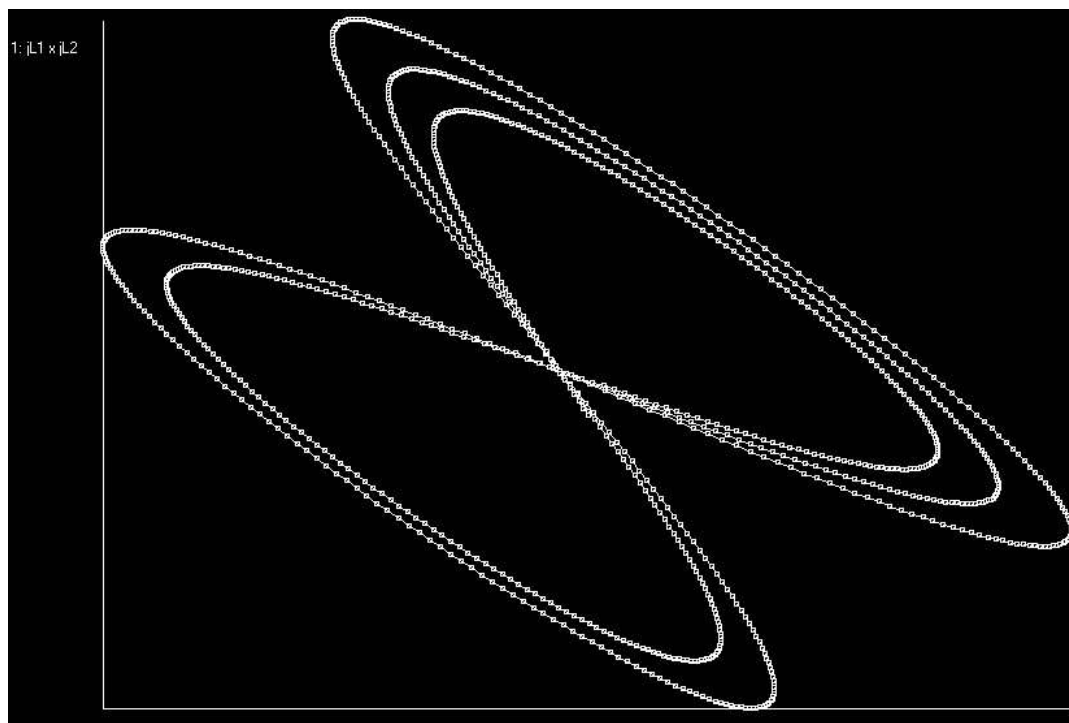
Plot (6 x j00200) - Relação da Tensão de Entrada no Ampop e a corrente



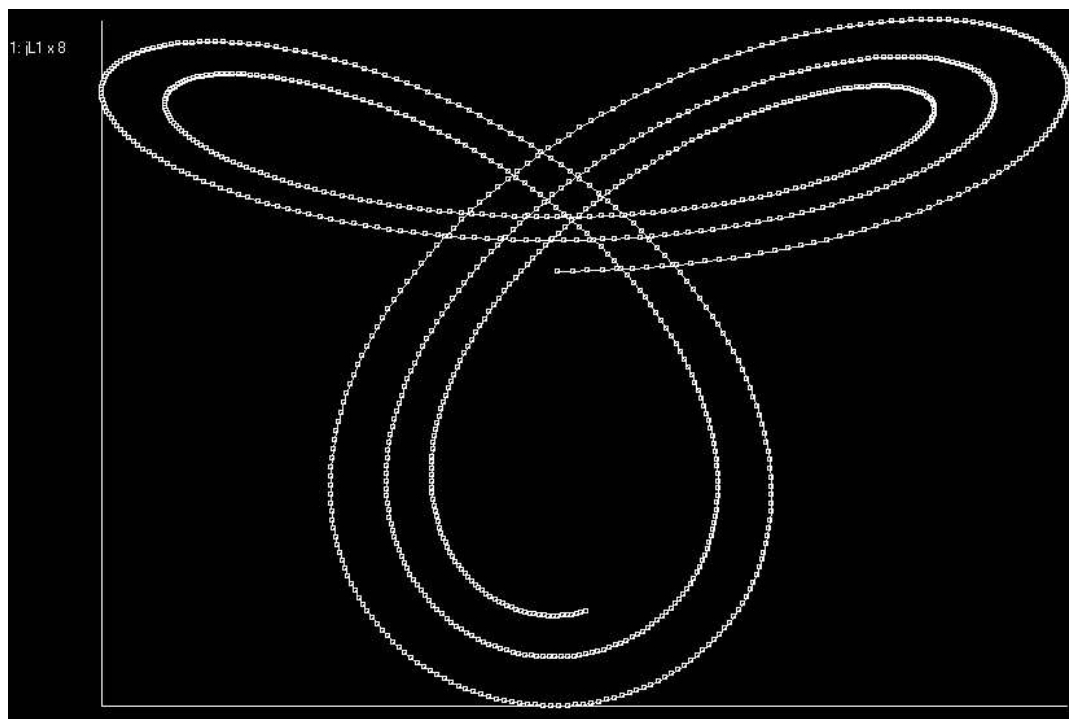
3) Plot (7 x 8)



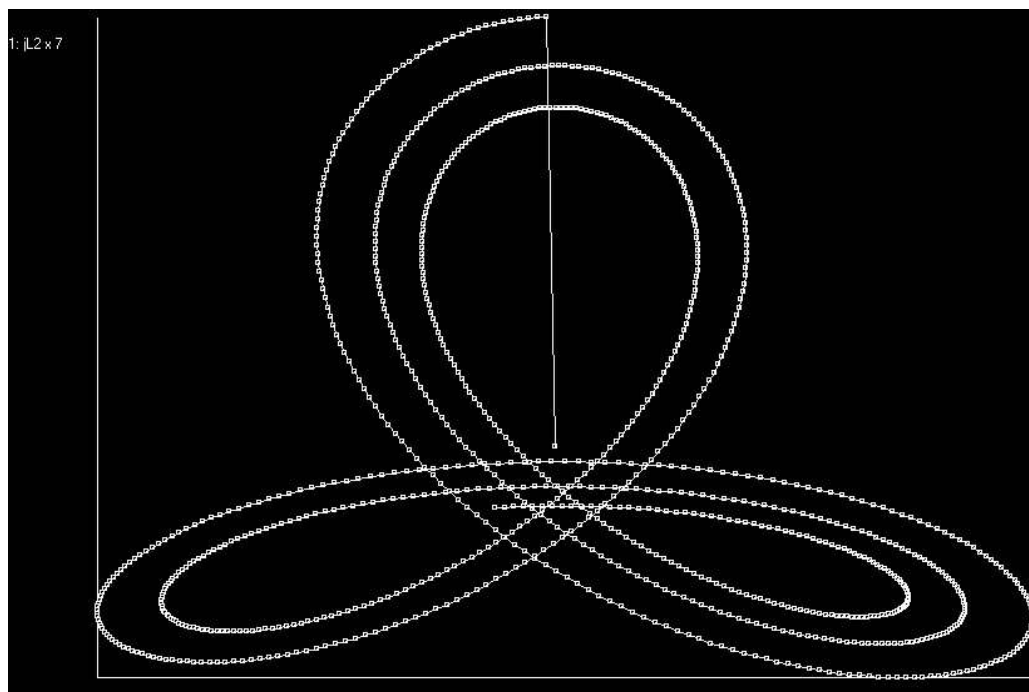
Plot (jL1 x jL2) - Correntes dos Indutores



Plot (jL1 x 8) - Corrente Indutor 1 x Tensao Indutor 2



Plot (jL2 x 7) - Corrente Indutor 2 x Tensao Indutor 1

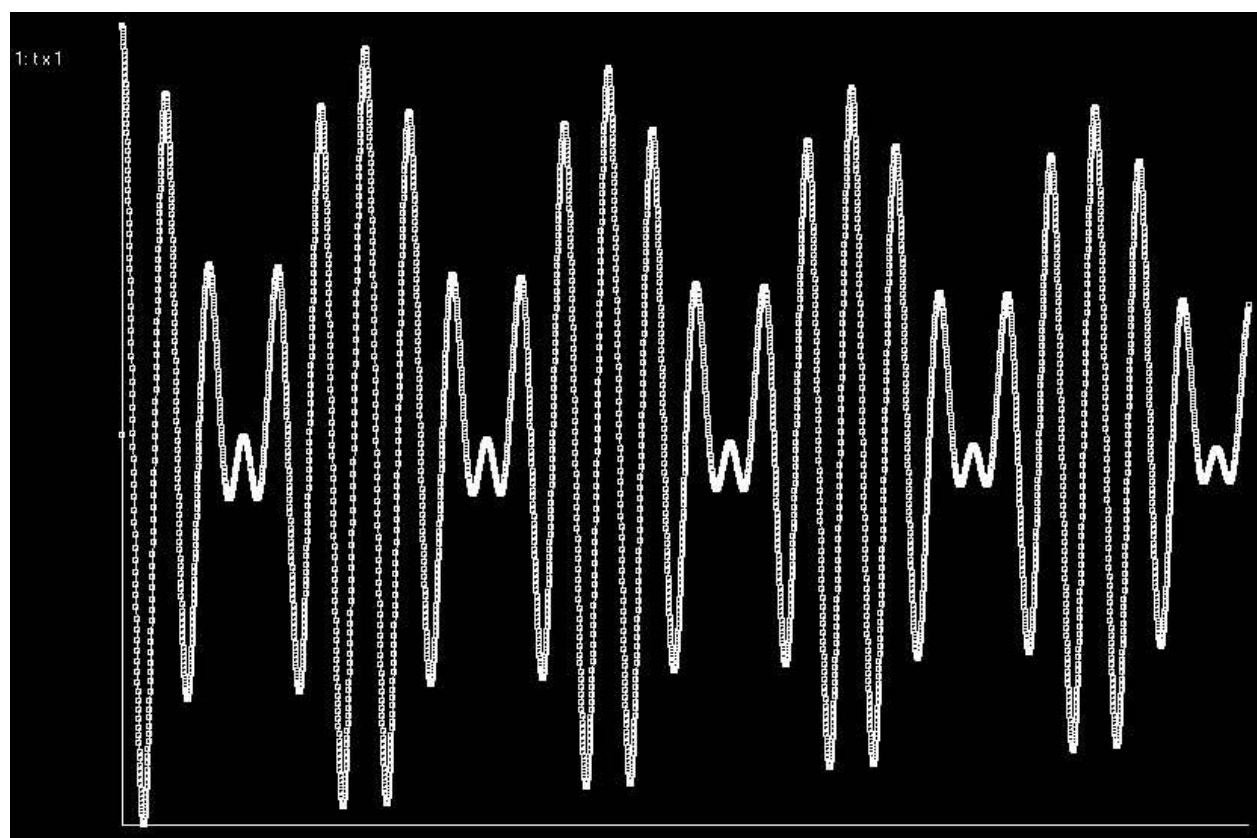


11 - Teslak.net

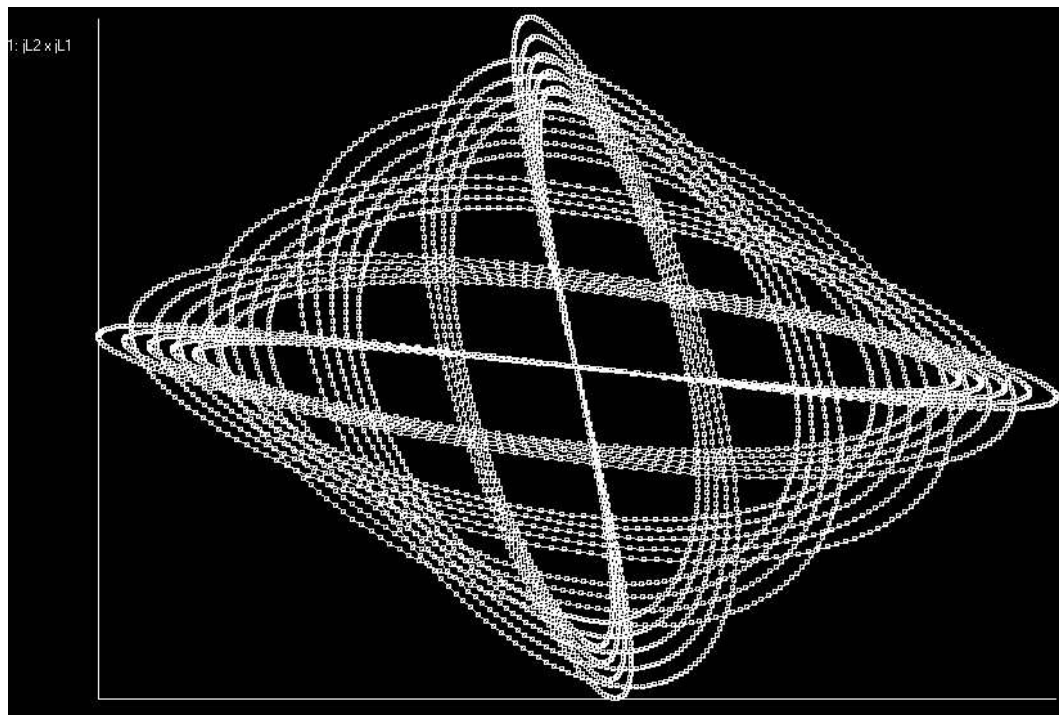
Circuito



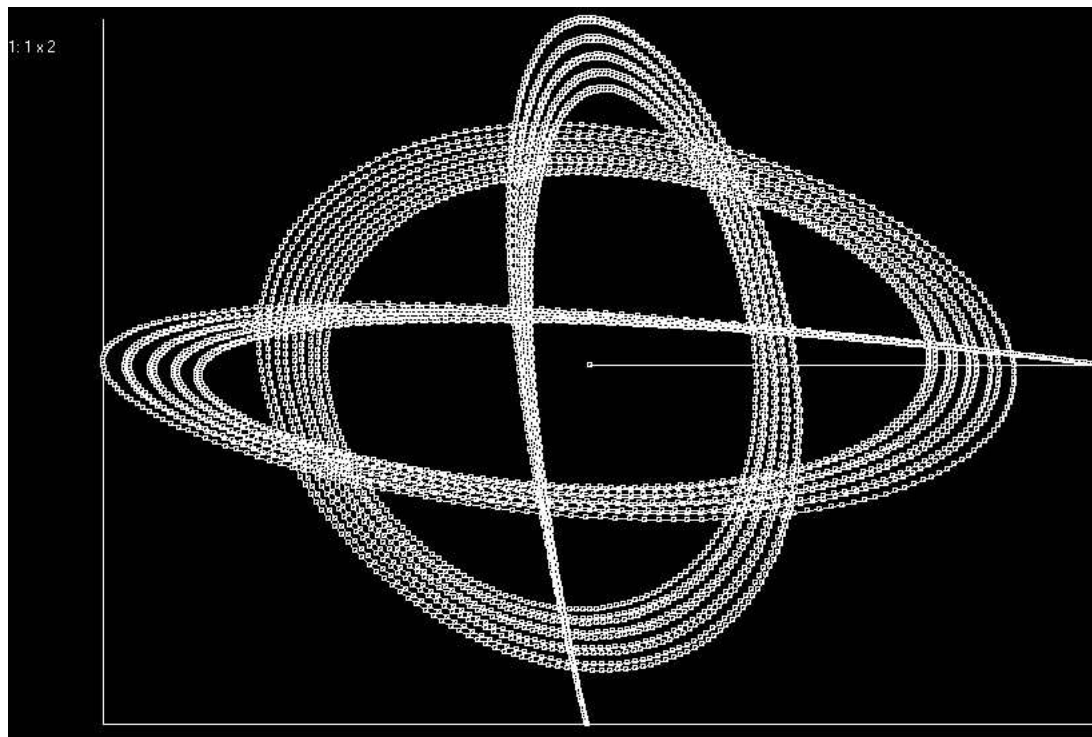
Plot (t x 1)



Plot (jL2 x jL1)



Plot (1 x 2)

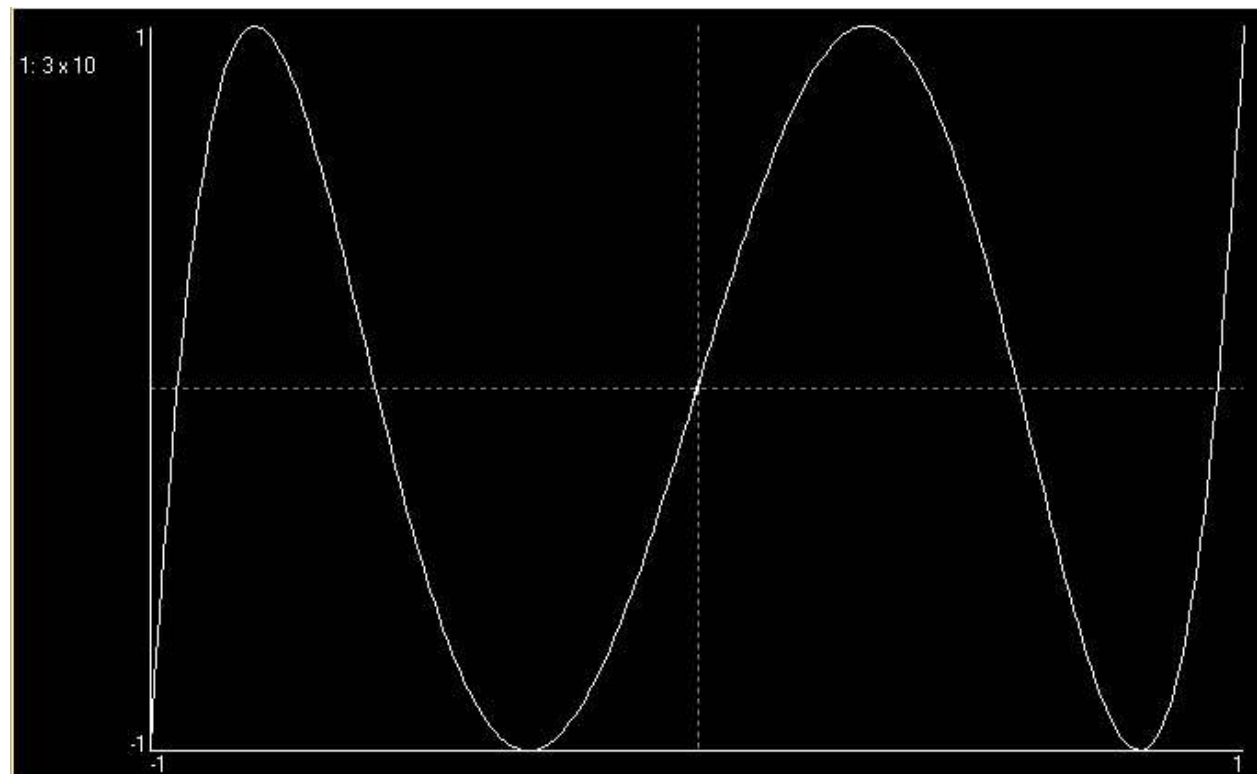


12 - Tracpol.net

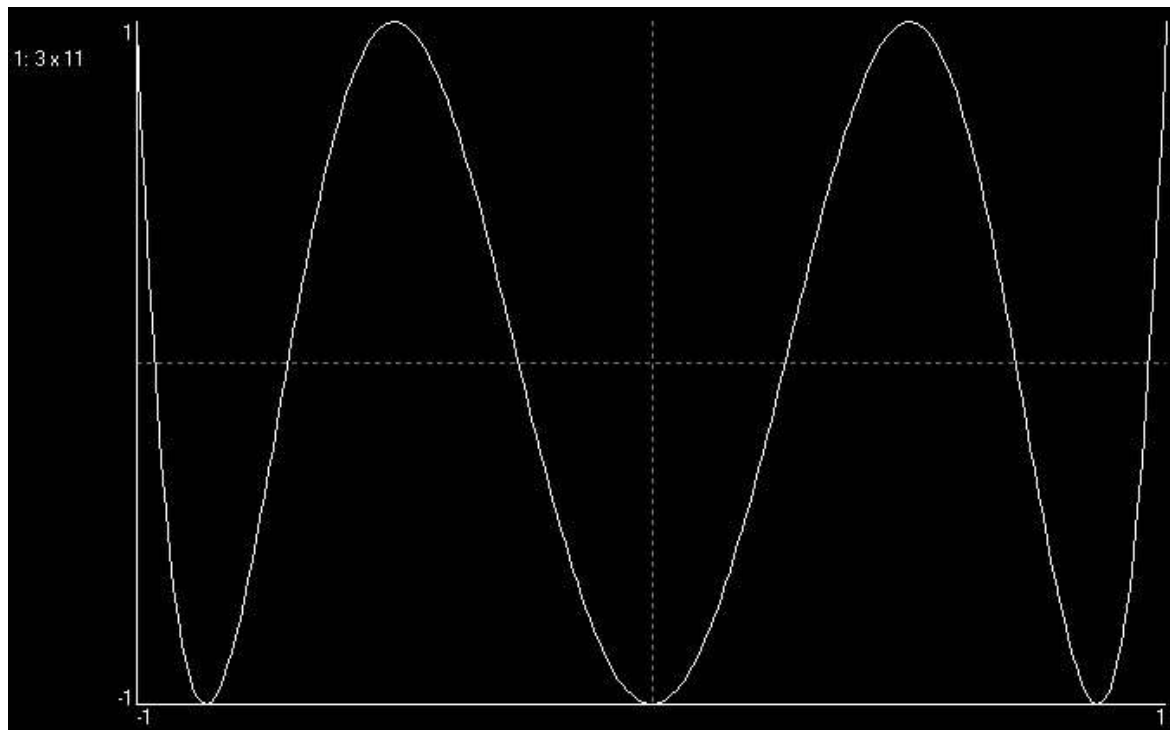
Circuito



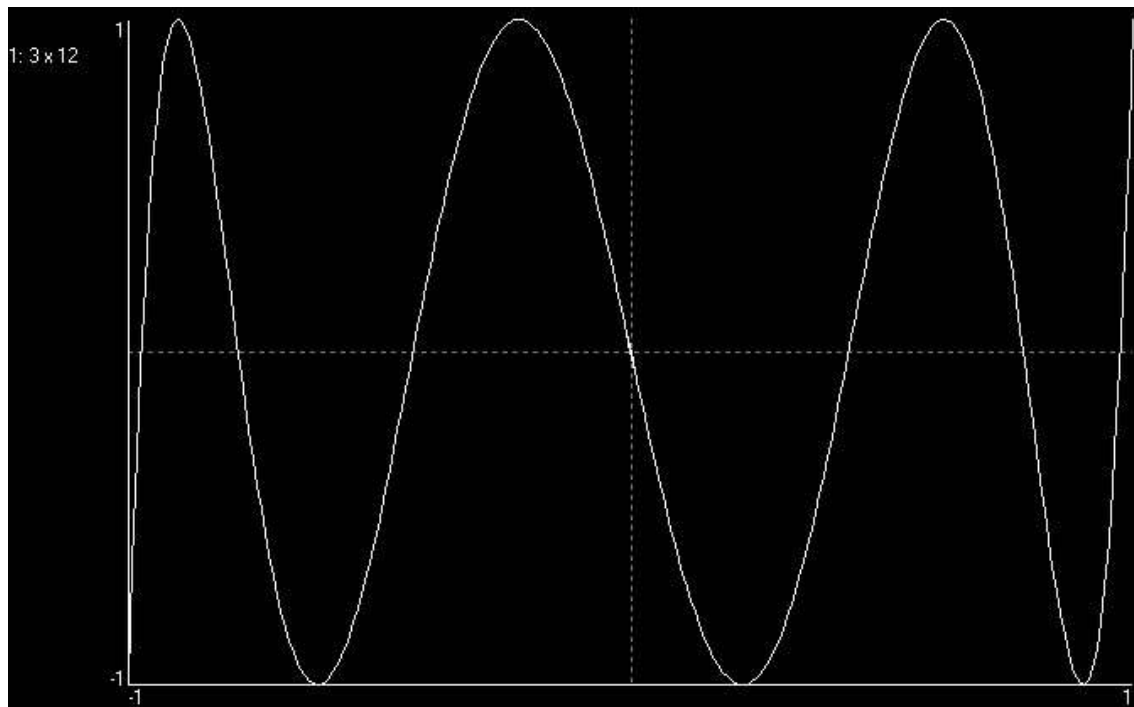
Plot (3 x 10)



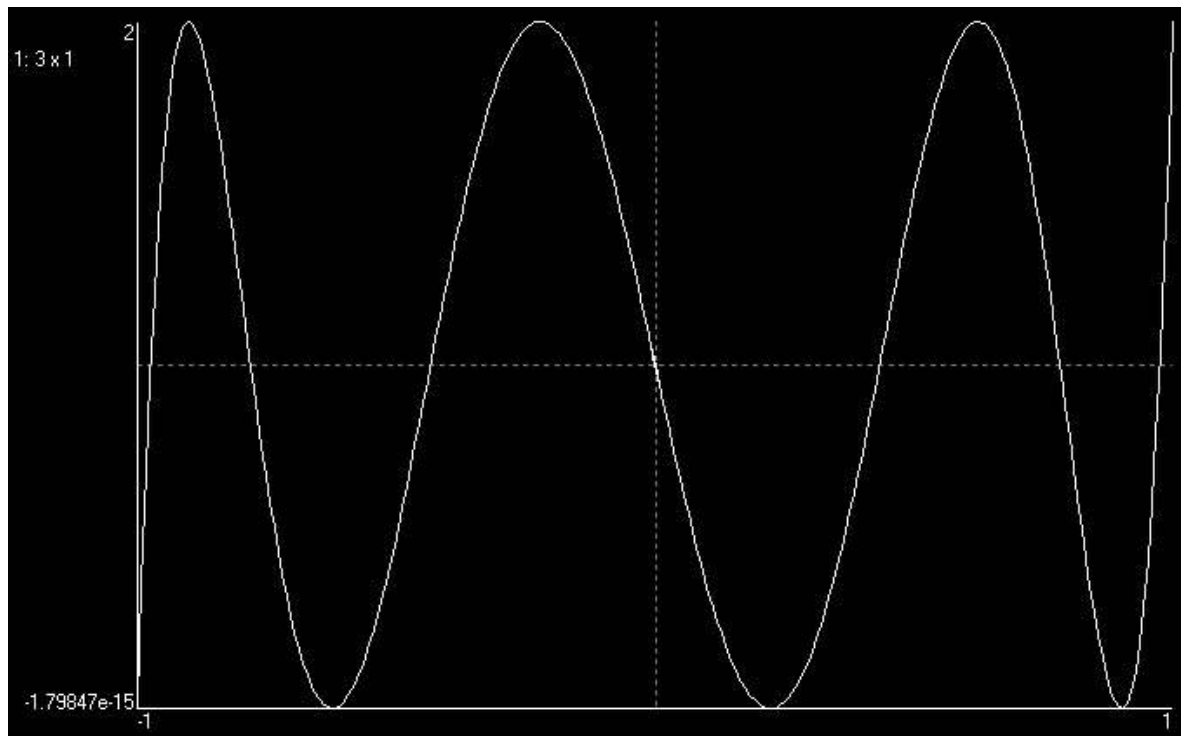
Plot (3 x 11)



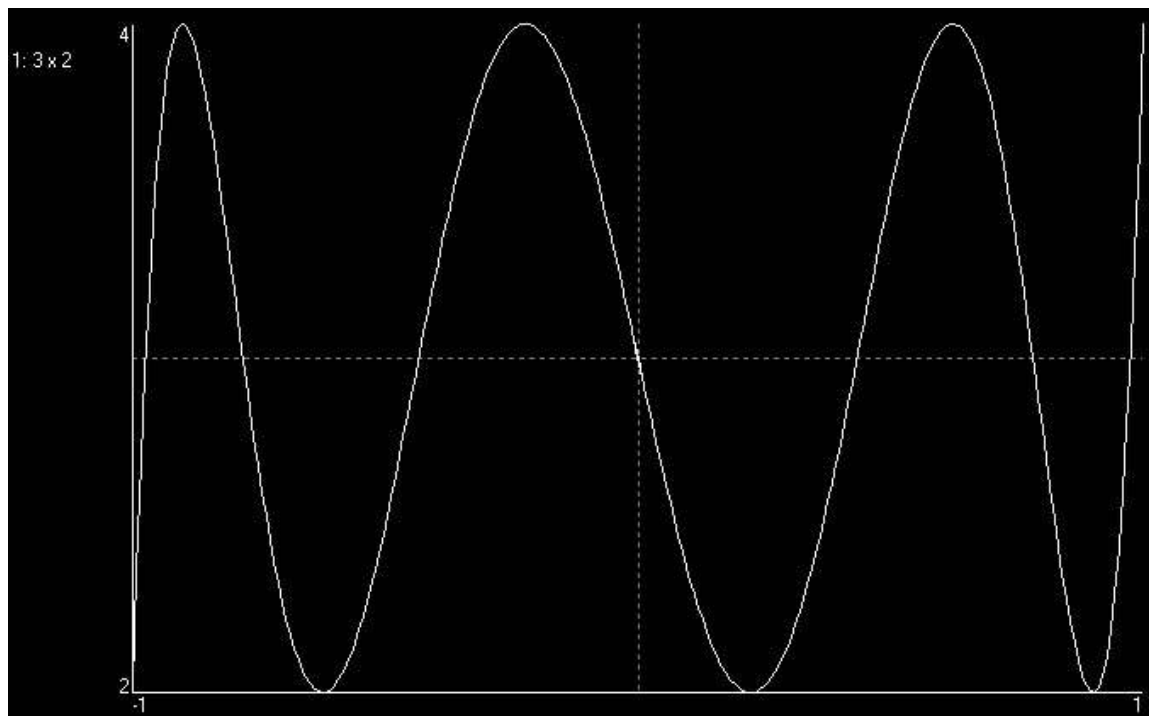
Plot (3 x 12)



Plot (3 x 1)

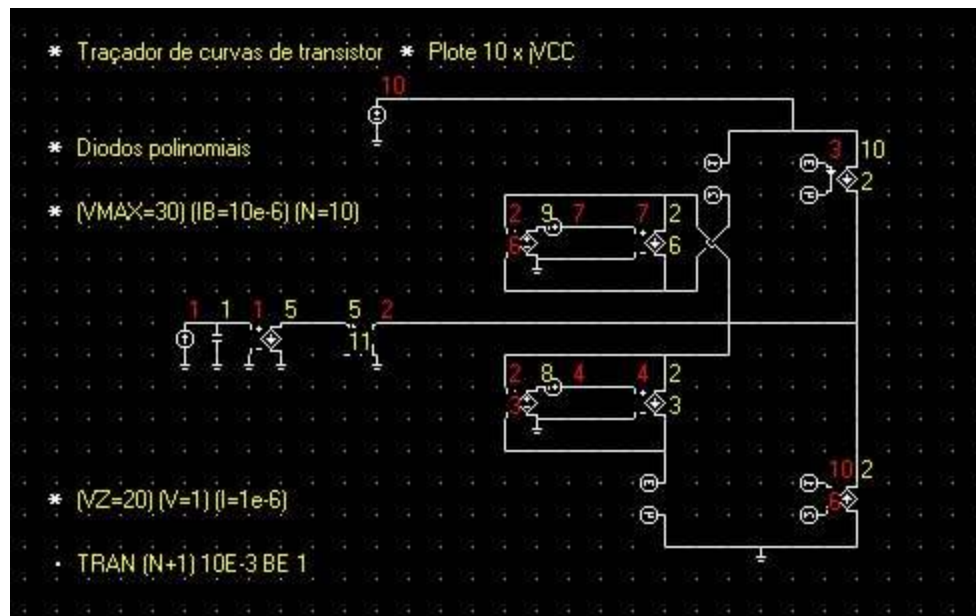


Plot (3 x 2)

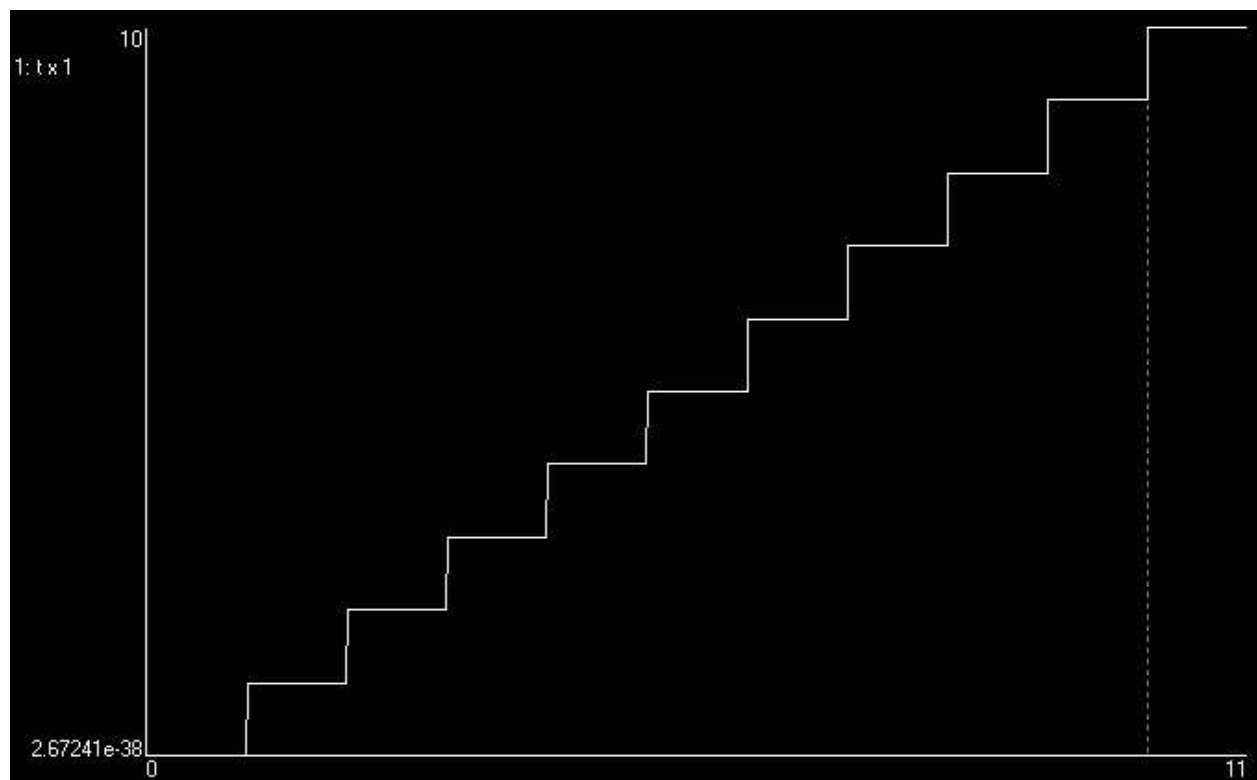


13 - Tractran.net

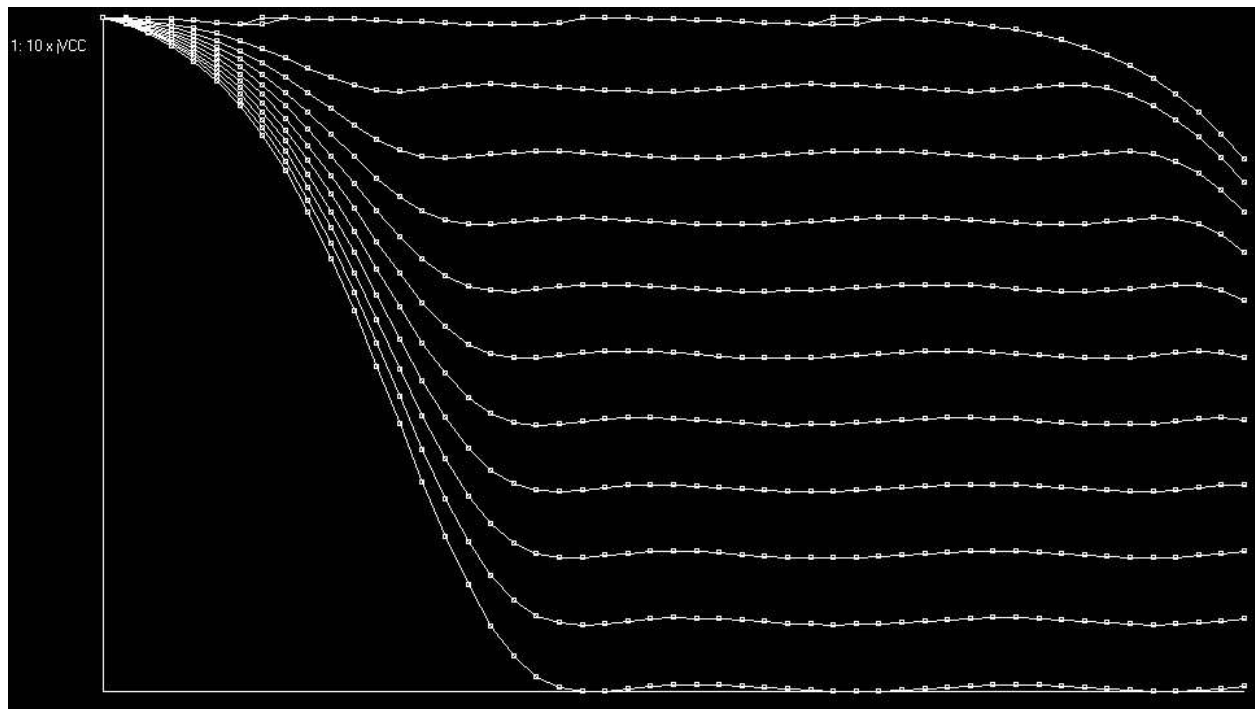
Circuito



Plot (t x 1)



Plot (10 x VCC)



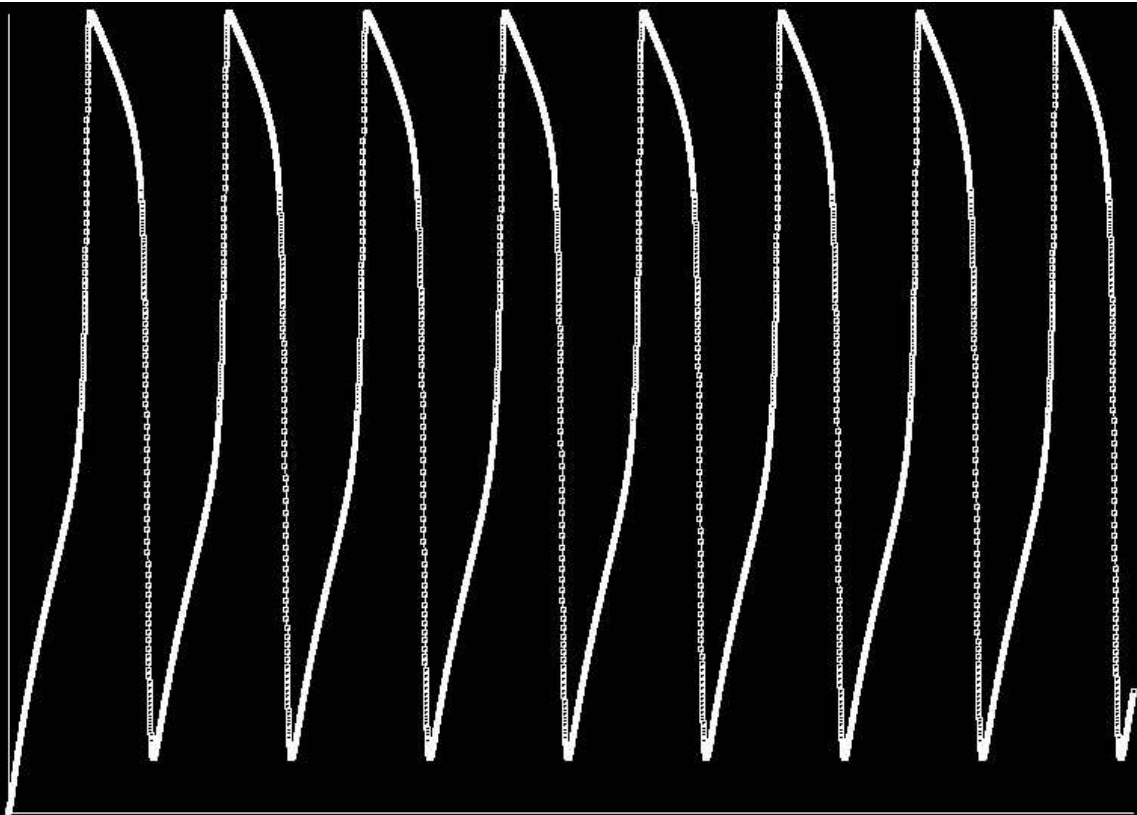
14 - Tunel3.net

Circuito

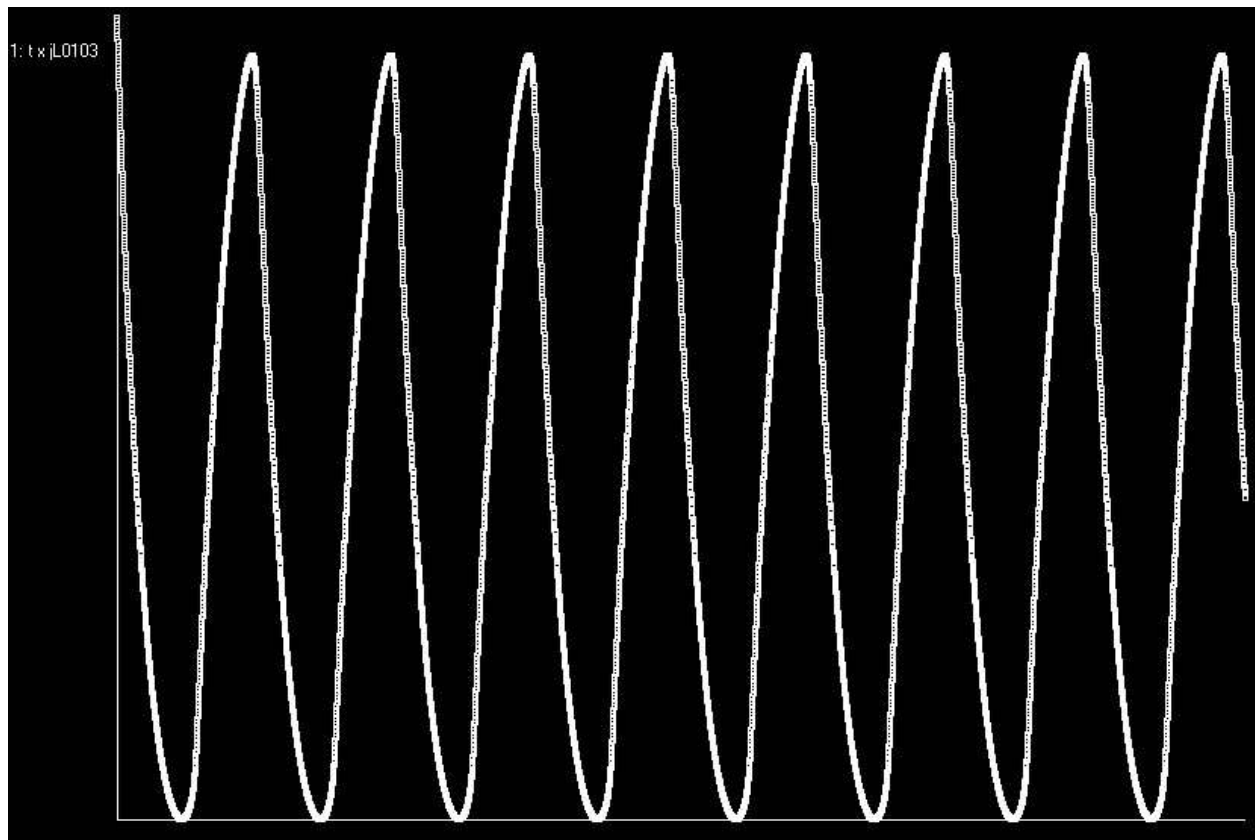


Plot (t x 1) - Tensão em 1 (V1)

1: t x 1



Plot (t x jL0103) - Corrente no Indutor



5 - Conclusão

Desenvolvemos um simulador para análise no tempo de circuitos elétricos. O programa envolve integração numérica, tratamento de elementos não-lineares e análise nodal modificada por estampas.

A experiência durante a execução do trabalho foi muito gratificante. Observar os resultados das simulações dos circuitos foi importante para reforçar conceitos aprendidos no curso.

Também foi importante observar na prática os efeitos da aplicação errada de conceitos da teoria de circuitos, investigar, identificar o problema e corrigir. Estes eventuais erros que encontramos durante o desenvolvimento do trabalho, foram fundamentais para solidificar nossos conhecimentos.

Através da visualização de gráficos gerados a partir das tabelas criadas pelo nosso programa, pudemos observar o comportamento de diversos circuitos. Essa percepção visual agregou muito ao nosso conhecimento e, frente a gráficos muito interessantes, foi uma importante fonte de motivação durante o desenvolvimento deste trabalho.

Por fim, vale ressaltar que somando nossa análise criteriosa e a ajuda do professor, foi possível simular com sucesso todos os circuitos de exemplo.

Anexo A

Em anexo, resultados dos testes unitários que compreendem nossas expectativas sobre as estampas dos elementos.

Anexo B

Também em anexo, resultados de análises comparando as diferenças entre as saídas retornadas pelo nosso programa e a saída esperada, obtida pelo programa MNAE.

Anexo C

Também em anexo, planilha com o resumo dos resultados obtidos para cada circuito de teste.

Anexo D

Código (inclui testes) do programa desenvolvido pelo grupo.

Anexo E

Planilha com itens de projeto organizados em um backlog .