



## **SUPERANDO O MITO DE FASORES E SÉRIES DE FOURIER**

**Thiago C. C.** – tccesar@ieee.org

Universidade Federal de Juiz de Fora, Faculdade de Engenharia.

Campus da UFJF – Bairro Martelos

36.036-330 – Juiz de Fora – MG

**Francisco J. G.** – chicogomes@terra.com.br

**Resumo:** Simulações constituem poderosa ferramenta no estudo de sistemas na área de engenharia, pois possibilitam ao usuário a visualização preliminar de resultados e análise de sua dinâmica. Um exemplo bastante ilustrativo desta situação é o estudo do comportamento de circuitos elétricos para entradas periódicas, onde a utilização de fasores é um método altamente poderoso e necessário ao aluno. O trabalho desenvolvido constitui um ambiente computacional que funciona como ferramenta didática para auxílio no estudo e compreensão da álgebra fasorial e de sua imensa aplicabilidade. O ambiente desenvolvido opera tanto com, entradas pré-configurados, bem como possibilita a definição de esquemas fasoriais pelo usuário. O ambiente fornece ao aluno uma visão da composição fasorial de diferentes frequências, no plano complexo, e sua resposta real no tempo através da saída de um circuito elétrico. Permite ainda a inserção de elementos lineares passivos no circuito padrão utilizado, possibilitando alteração dos fasores e geração da saída do circuito, quando excitado por uma fonte com a forma da entrada escolhida.

**Palavras-chave:** *Fasores, Séries de Fourier, Circuitos lineares, Ambiente computacional*

### **1. INTRODUÇÃO**

A simulação dos diversos sistemas, nos projetos de Engenharia, constitui atualmente uma etapa obrigatória em praticamente todas suas etapas. Desenvolvimento necessita de simulações computacionais antes de sua fase de implementação e para tanto é necessária a utilização de várias ferramentas matemáticas e técnicas de programação. A área de sistemas lineares tem uma grande importância nos dias de hoje, e é de grande utilidade em análise de circuitos elétricos, sistemas de controle, etc.

Sinais periódicos são extensivamente utilizados na área de engenharia elétrica, principalmente na eletrônica, telecomunicações e processamento de sinais. O estudo de sinais periódicos ocorre, em grande parte, pela utilização de funções periódicas, que podem ser expressas na forma das séries de Fourier. No estudo de sistemas lineares, especialmente circuitos elétricos, a utilização das séries de Fourier, bem como a aplicação da álgebra fasorial constitui um conhecimento obrigatório para o estudante, ferramentas estas com as quais necessita estar profundamente familiarizado e apto a utilizá-las de forma objetiva. Contudo, esta percepção da aplicabilidade da álgebra fasorial, e mesmo das funções de transferência, nem sempre é perceptível ao aluno. A regra geral, facilmente observável, é que os alunos manipulam mecanicamente estes conceitos e suas aplicações, sem entendimento concreto de suas relações, sem percepção exata de suas correlação com as

respostas temporais e mesmo sem saberem como extrair informações adicionais destes sistemas, a partir destas estruturas.

A proposta deste trabalho consistiu no desenvolvimento de uma ferramenta que possibilite ao aluno formas mais concretas de contato com este ferramental teórico, verificando sua aplicabilidade em um circuito elétrico. O sistema permite que o aluno efetue a construção de um sinal desejado, a partir de funções senoidais de diferentes módulos, fases e frequências quantizadas, que se expressam através de suas harmônicas. A partir desta base, o sistema possibilita a visualização da contribuição de cada uma destas características das harmônicas na função resultante, no domínio do tempo. O ambiente, adicionalmente à apresentação da resposta temporal da função, possibilita também a visualização do comportamento dos fasores associados, no plano complexo, em função do tempo, deixando claro o efeito da superposição de fasores, de diferentes frequências, e seu giro, à medida que ocorre variação do tempo.

Adicionalmente, o ambiente possibilita ao aluno um melhor entendimento do funcionamento básico de simuladores de circuitos utilizando-se o método das séries de Fourier. Para isto, abre a possibilidade do usuário alterar os parâmetros de uma malha de um circuito RLC: o ambiente determina a função de transferência do circuito selecionado, excitando-a com os fasores do sinal de entrada selecionado, mostrando ao usuário a corrente que flui no circuito em regime permanente, que é apresentada no domínio do tempo e no plano complexo. As funções utilizadas como sinal de entrada foram extraídas da forma complexa da série de Fourier; as operações com as exponenciais complexas foram efetuadas utilizando-se técnicas de álgebra fasorial, implementadas na linguagem computacional Java.

O trabalho apresenta de forma detalhada todas as etapas do desenvolvimento na seção 2, os resultados obtidos, alguns exemplos e imagens do software são apresentados na seção 3. Na seção 4 tem-se as conclusões finais e comentários gerais sobre o trabalho desenvolvido.

## **2. IMPLEMENTAÇÃO**

### **2.1 A linguagem**

A escolha da linguagem de programação foi efetuada baseando-se em dois pré-requisitos mínimos que foram a facilidade de trabalhar as interfaces gráficas e sua característica de orientação a objetos. A importância do primeiro requisito, a interface gráfica, fica clara quando se coloca a necessidade de interação com o usuário, com geração de saídas em forma gráfica, de forma clara, compreensível e amigável para o usuário. Já a característica de orientação a objetos torna-se uma necessidade para facilidade de implementação, possibilitando o desenvolvimento em módulos separados, totalmente estanques, mas ao mesmo tempo com extrema portabilidade. Considerando-se que o ambiente opera com métodos e ferramentas bastante diferenciadas, que devem ser simples, porém totalmente entrelaçadas e interativas, verifica-se que a orientação a objetos é de importância crucial para o resultado efetivo da proposta.

Dentre as linguagens possíveis para a tarefa, que satisfaziam os requisitos colocados, podem ser citadas: C++, Delphi, Java, Visual Basic, entre outras. A opção final foi pela utilização da linguagem Java da Sun Microsystems, por diversos motivos, explicados na sequência. Inicialmente, além de possuir estrutura de orientação a objetos completa e

rígida, Java apresenta a vantagem de ser multiplataforma, sem necessidade de qualquer mudança de código ou de compilação, tornando sua portabilidade insuperável. Esta característica decorre do fato que seu compilador gera outra linguagem, o *bytecode*, que é interpretada pela máquina virtual Java (JVM), possibilitando que o programa seja utilizado em qualquer sistema operacional, bastando para isso apenas ter a JVM instalada. A opção por multiplataformas evita que o ambiente fique preso a sistemas proprietários, sendo totalmente compatível com a escolha do sistema operacional efetuada pelo usuário, inclusive utilizando "software livre". A escolha da linguagem Java possibilita ainda que o ambiente possa ser facilmente disponibilizado e utilizado em rede Web, bastando para isto efetuar seu encapsulamento em um *Applet* apropriado. A proposta de se utilizar formato Web é extremamente atraente, haja vista que o aluno, enquanto navega pela Internet, possa encontrar o ambiente desenvolvido e, sem maiores empecilhos, utilizá-lo através de seu navegador.

Contudo, deve ser citado um aspecto negativo da escolha, que é o fato do Java não apresentar características de linguagem científica, possuindo apenas funções matemáticas básicas. Contudo, dada a opção pelo orientação a objeto, este problema foi facilmente contornado, não acarretando problemas de maior monta e nem envolvendo cálculos extremamente complexos.

## 2.2 Modelo Matemático

O modelo utilizado para representar os sinais periódicos foi o da forma complexa da série de Fourier, que pode ser encontrado no trabalho de SPIEGEL (1976),

$$s(t) = \sum_{n=-\infty}^{\infty} C_n e^{in\omega t} \quad (1)$$

onde as constantes complexas  $C_n$  podem ser determinadas pela seguinte integral.

$$C_n = \frac{1}{T} \int_{-T/2}^{T/2} s(t) e^{in\omega t} dt \quad (2)$$

Na forma fasorial, os termos da série de Fourier podem ser escritos como

$$F = C_n \cdot e^{i\omega t} \quad (3)$$

$$F = |F| \cdot e^{i\omega t} \cdot e^{-\phi} \quad (4)$$

$$\Re\{F\} = |F| \cdot \cos(\omega t - \phi) \quad (5)$$

$$x(t) = \Re\left\{\sum_{-\infty}^{\infty} F\right\} \quad (6)$$

Verifica-se assim que é possível obter-se o resultado da série domínio do tempo somando-se os termos reais das equações de Fourier da equação (1). Pode-se assim exibir o giro dos fasores concatenados utilizando-se a forma da série truncada, com o número de harmônicos que for necessário às necessidades do usuário.

Os procedimentos de simulação utilizados para os circuitos lineares pode ser exemplificado com a situação descrita a seguir.

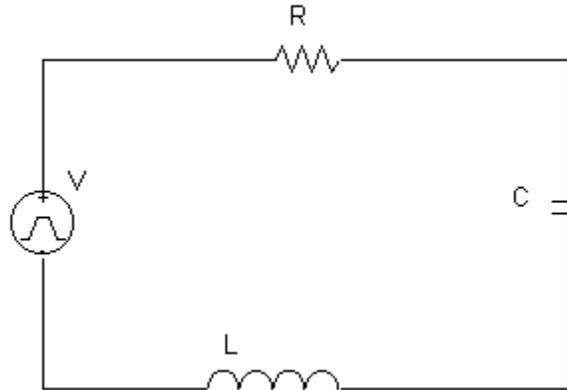


Figura 1 - Circuito RLC utilizado na simulação

A determinação da corrente que circula na malha RLC foi possível utilizando-se sua função de transferência, aplicada a cada uma das frequências referentes aos harmônicos da função de entrada, através da seguinte relação, que pode ser encontrada no livro de CLOSE (1975) ou de QUEVEDO (1988).

$$Z(i\omega) = \sqrt{R^2 + \left(\omega L - \frac{1}{\omega C}\right)^2} \quad \angle \phi_c \quad (7)$$

$$\phi_c = \arctan\left(\frac{\omega L - \frac{1}{\omega C}}{R}\right) \quad (8)$$

$$I = \frac{E}{Z} = \frac{|E|}{|Z|} \cdot e^{i\omega t} \cdot e^{\phi - \phi_c} \quad (9)$$

### 2.3 Estrutura do sistema

A estrutura básica do software está ilustrada na figura 2, de acordo com a metodologia apresentada por DEITEL e DEITEL (2003). A classe *Applet* é a principal do sistema, constituindo o programa em si. Ela contém todas as informações de *layout*, os botões, os *labels* os painéis, e os controles dos eventos principais da tela do programa. Para utilizar os componentes que a linguagem Java disponibiliza, foi necessário o acesso à documentação dos mesmos no *site* da Sun Microsystems (2005).

A classe *GenericGraph* é uma classe genérica para geração de gráficos, efetuando ainda o interfaceamento entre a forma matemática de se gerar um gráfico e a maneira como estes gráficos serão gerados em Java. Foram também implementados, nesta classe, alguns controles para otimização do processo, evitando que haja a necessidade, a cada ocorrência de algum evento, de se efetuar o redesenho de toda a tela: esta função otimizatória guarda os pontos a serem plotados, evitando que sejam recalculados. Trata-se de uma superclasse visível no programa. A *Graph* é a classe responsável por ilustrar o sinal no domínio do tempo e herda propriedades e métodos de *GenericGraph*. *ComplexPlane*, que também herda atributos da *GenericGraph*, é a responsável pela disponibilização dos fasores no plano complexo, inclusive quando estes estiverem girando à medida que o tempo passa. Existe, neste aspecto, a implementação de um importante evento, que permite ao usuário, no modo *Desenho de Fasores*, marcar os fasores diretamente no gráfico exibido pelo ambiente. As duas classes têm um relacionamento de agregação com a classe *Applet*, pois são componentes *ativos* do sistema.

A classe *FourierSerie* é a principal responsável pelos cálculos do programa. Trata-se de uma superclasse que, para cada tipo de sinal periódico que se deseja representar, cria as subclasses *SquareSerie* e *SawSerie*, que herdam atributos - como nas exemplificadas anteriormente. A informação que é específica de cada subclasse é a regra de formação de cada fasor, os coeficientes  $C_n$  da equação (1), que devem ser calculados separadamente para cada novo sinal que for solicitado pelo usuário. Esta classe agrega também a classe *Phasors*, que registra as informações referentes aos fasores, (módulo, harmônico e fase). Como a superclasse *FourierSerie* possui informações de todos fasores, é ela quem fornece o valor da série truncada para o sistema, para um determinado instante solicitado.

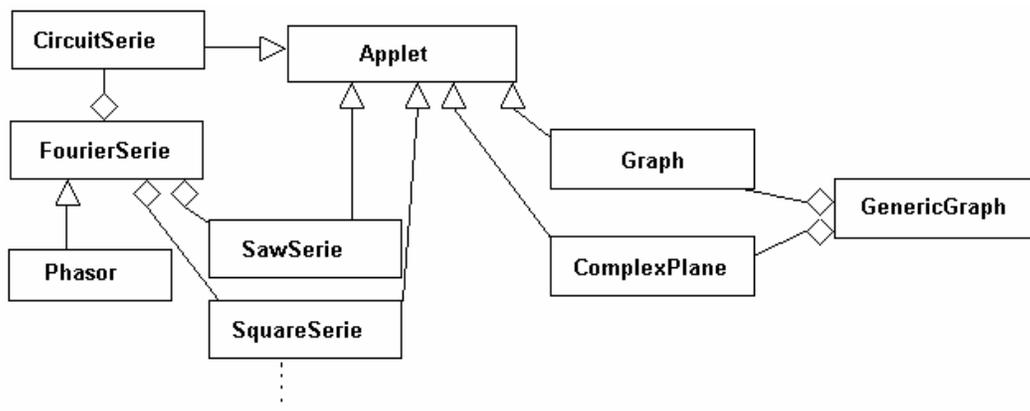


Figura 2 - Estrutura da programação do ambiente desenvolvido

A classe *CircuitSerie* é a responsável pela simulação de circuitos lineares, herda também atributos da *FourierSerie* e está agregada à *Applet*. Quando os parâmetros do circuito estão definidos e a simulação se inicia, o programa seleciona um objeto da classe *CircuitSerie*, copiando todos os fasores da série atual mas configurada para o novo objeto. O método *addPhasor* (que é responsável por adicionar um fasor a uma série) é sobrescrito de forma a passar o fasor pela função de transferência de circuito para que ele passe a representar não a entrada (sinal) mas sim a série de saída do circuito (corrente). A partir daí

a simulação continua igualmente como se fosse uma *FourierSerie* comum. Esta é outra vantagem de se utilizar a Orientação a Objetos no desenvolvimento do ambiente.

### 3. Resultados Obtidos

A tela inicial para o ambiente está ilustrada na figura 3. No topo tem-se a região aonde serão gerados os gráficos no tempo e no plano complexo. No meio, uma tabela com as propriedades dos fasores gerados. Neste ponto do programa o usuário deve fazer as suas escolhas para gerar o sinal que deseja.

Os botões rádio, são as opções de ondas que já existem implementadas, A opção *Desenhar Fasores* permite que se marque, no plano complexo, os fasores desejados, possibilitando ao usuário acompanhar o giro dos fasores, bem como o gráfico temporal do sinal. A opção *Inserir Manualmente*, permite, ao se pressionar o ícone *Editar*, a inserção das propriedades dos fasores (módulo, fase e harmônico) através de caixas de texto.

A figura 3 exibe os ícones de ação para a simulação. Pressionando-se *Play* a simulação terá início, com as opções selecionadas. O botão *Circuito* permite alterar a tela para simular a resposta de um circuito RLC, para o sinal de entrada atual. Tem-se, adicionalmente, um campo de texto que permite a especificação do número de harmônicos que se deseja para esta simulação.

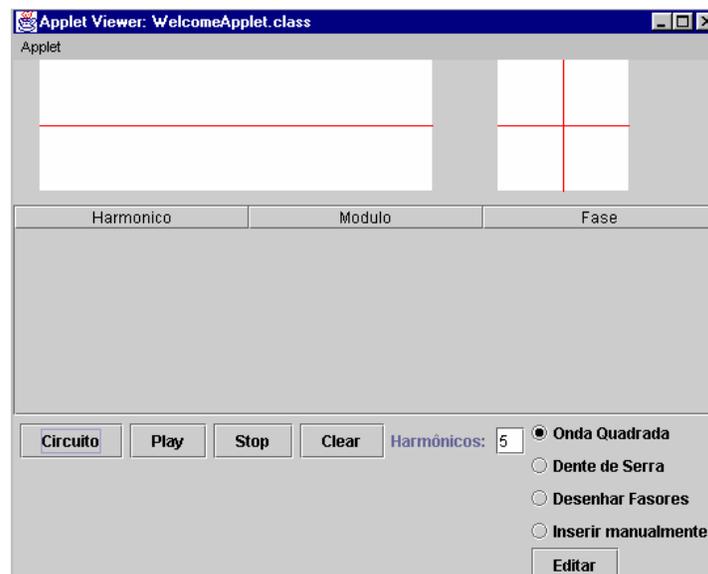


Figura 3 - Tela inicial do ambiente

O primeiro exemplo - figura 4 - mostra a situação onde é possível visualizar a geração de sinais periódicos. O sinal, neste caso, é uma onda quadrada, com ciclos positivos e negativos. No topo da tela podem ser visualizados, à esquerda, a variação da função no tempo e à direita o comportamento dos fasores. As propriedades de cada fasor utilizado na geração do sinal são mostradas na parte central da tela.

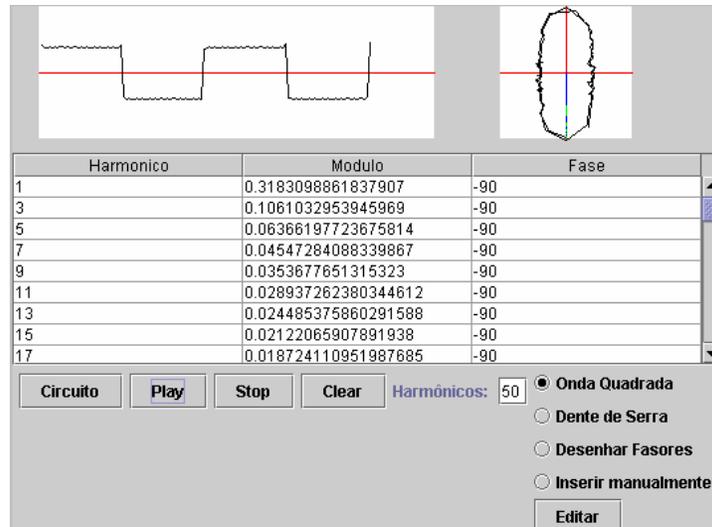


Figura 4 - Exemplo da construção de funções

O ambiente mostra claramente que o gráfico temporal da função é a projeção, no eixo real, da composição existente no plano imaginário, construída pelo diagrama fasorial. O ambiente mostra ainda que nas partes laterais do diagrama os fasores apresentam um giro com um período de maior duração, resultando que sua projeção no eixo real é uma constante; contudo, durante a transição do estado positivo para o negativo, os fasores giram muito rapidamente, formando pequenas figuras no topo e base do diagrama, ocasionando que o eixo real faça uma alternância quase instantânea entre os semiplanos positivo e negativo do eixo real. Variando-se o número de harmônicos percebe-se, claramente, a convergência da série truncada; a figura 5 mostra exemplo de duas situações, para a mesma função, com 5 e 50 termos na série de Fourier, e as aproximações conseguidas em cada caso.

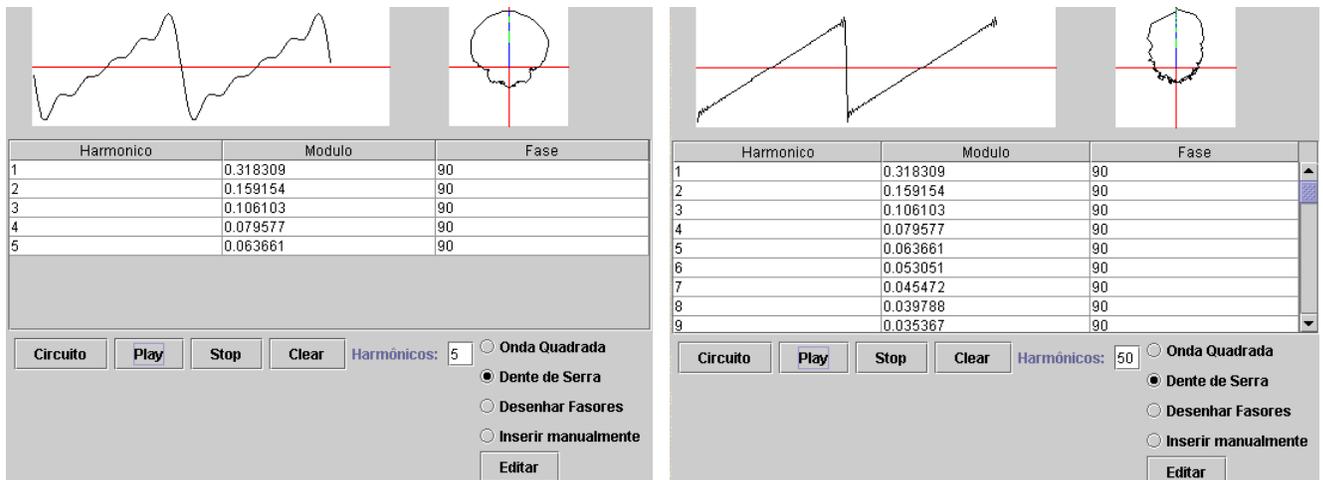


Figura 5 - Aproximações da função com 5 e 50 termos na série

A figura 6 mostra a simulação de um circuito RLC utilizando-se o ambiente desenvolvido. Vê-se, na tela, em sua parte central, o circuito que será simulado, com as respectivas entradas para os parâmetros de configuração do circuito. Neste caso, foram

adotados uma resistência de 100 ohms, uma indutância de 10 mH e um capacitor de 100  $\mu$ F. Pressionando-se o botão *Circuito*, volta-se à tela anterior. Mostra-se, na parte superior da tela, as formas de onda temporal e fasorial para o circuito RLC, com as configurações escolhidas.

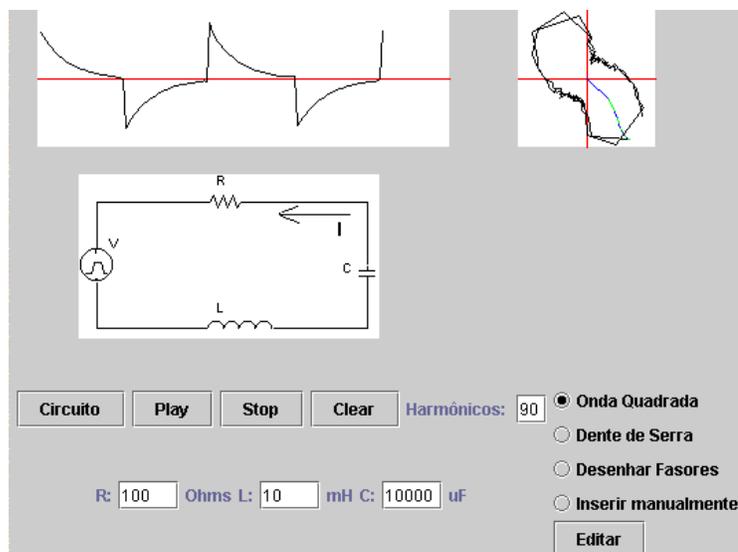


Figura 6 - Simulação de um circuito RLC

Para efeitos comparativos, é possível ver, na figura 7, os resultados obtidos na simulação deste mesmo circuito no Pspice, com uma frequência de 1rad/s, que é a frequência definida no programa para as ondas geradas.

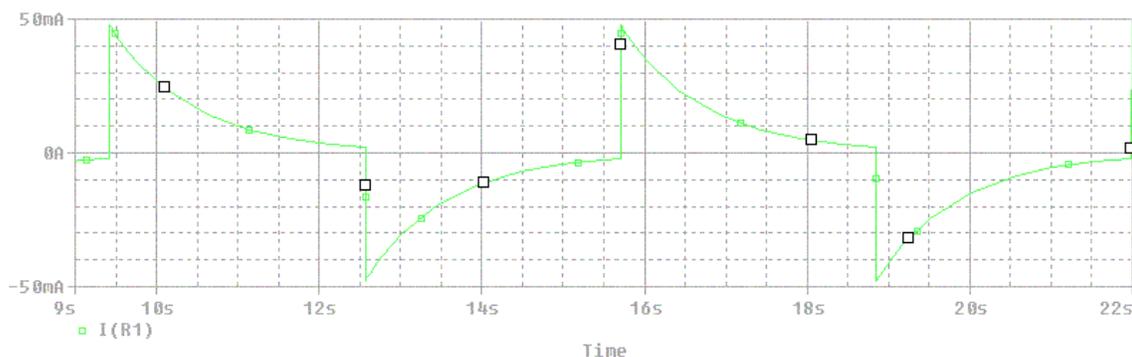


Figura 7 - Resultados obtidos no Pspice para o mesmo circuito RLC

Fica claro que os dois gráficos, qualitativamente, são bastante semelhantes. Verifica-se apenas uma variação na fase das duas ondas, visto que no Pspice o eixo do tempo foi adiantado para evitar os transientes de corrente, isto porque a análise por séries de Fourier nos permite analisar os circuitos apenas em regime permanente.

#### 4. Considerações Finais

O emprego de sinais periódicos é, atualmente, fundamental para a engenharia, especialmente nas áreas de Eletrônica, Processamento de sinais e Telecomunicações. Mas,

adicionalmente, o conhecimento e utilização da álgebra fasorial, bem como das séries de Fourier, é condição fundamental para a Engenharia Elétrica. Com a finalidade de tornar esta ferramenta mais palpável desenvolveu-se o ambiente apresentado, que deve funcionar como ferramenta didática de suporte ao aprendizado do estudante ligado às áreas da engenharia elétrica, eletrônica e de sinais em geral.

A linguagem de programação utilizada, que foi o Java, Orientada a Objetos, apesar de não ser uma linguagem científica, apresentou bons resultados nos cálculos necessários à montagem do ambiente. O fundamental na utilização do Java foi a boa estrutura montada para integração entre as classes do projeto. Ressalte-se que a maior dificuldade encontrada no desenvolvimento esteve relacionada com a parte gráfica do ambiente, pois como o compilador gera código com características multiplataforma, não permite muita flexibilidade na disposição de objetos na interface, pois utiliza um posicionamento relativo.

Analisando os resultados obtidos gráficos verifica-se que as séries apresentaram boa convergência para os harmônicos utilizados, mesmo sem utilizar técnicas de convergência, como os janelamentos encontrados no site da Johns Hopkins University (2005) que utiliza multiplicadores para os módulos dos fasores, possibilitando uma melhorar a convergência de séries truncadas.

O projeto no nível educacional é bastante atraente, pois é fato conhecido que muitos alunos não conseguem a percepção necessária, nos cursos básicos, da álgebra fasorial, com seus fasores girantes e que se somam, com comportamento diferenciado em frequências diferenciadas, bem como o desenvolvimento da série de Fourier. O ambiente desenvolvido possibilita esta visualização, associando-a a uma situação real que é a análise de um circuito RLC.

Pelas características apresentadas, vê-se que este trabalho pode funcionar como importante ferramenta didática no ensino da Engenharia Elétrica, em suas diversas modalidades.

### ***Agradecimentos***

Ao Programa Especial de Treinamento – PET/SESu/UFJF pela oportunidade de desenvolvimento deste trabalho.

### **REFERÊNCIAS BIBLIOGRÁFICAS**

CLOSE, C. M., Circuitos Lineares, Rio de Janeiro: LTC, 1975.

DEITEL, H. M & DEITEL, P. J. Java Como Programar. Porto Alegre: Bookman, 2003, 4º Edição.

QUEVEDO, Carlos P., Circuitos Elétricos, LTC Editora, (1988), Rio de Janeiro, Brasil.

SPIEGEL, Murray, Análise de Fourier, Coleção Schaum, McGraw-Hill do Brasil, (1976), São Paulo, Brasil.

<http://www.jhu.edu/~signals>

<http://java.sun.com/j2se/1.3/docs/api/>

### **OVERCOMING THE MYTH OF PHASORS AND FOURIER SERIES**

**Abstract:** *Simulation constitutes a very powerful tool in the study of systems in engineering, therefore they make possible to the user the preliminary visualization of results and analysis of its dynamics. A good illustrative example is the analysis of linear circuits for periodic signals, where the use of phasors is a great and necessary method to the student. The project developed here consists of a computer environment that works as a didactic tool for aid in the study and understanding of the use of phasors and its immense applicability. The environment operates in such a way with previous configured entrances, as well as makes possible the user to define other phasorial schemes. The software supplies to the student an illustration of the composition of phasors of different frequencies, in the complex plane, and its real time response through the response of an linear circuit. It still allows the user to enter passive linear elements in the default circuit, making possible alteration of the phasors and generation of the circuit response when excited by the chosen source.*

**Key-words:** *Phasors, Fourier Series, Linear Circuits, Computer Environment.*