

# O uso da plataforma de desenvolvimento OMAP-L137/TMS320C6747, de processamento digital de sinais, na implementação de filtros digitais

Thamyres T. C. Palitó <sup>[1]</sup>, Caio S. Florentino <sup>[2]</sup>, Silvana Cunha Costa <sup>[3]</sup>, Suzete E. N. Correia <sup>[4]</sup>, Michel C. Dias <sup>[5]</sup>

[1] thamyrestamulla@gmail.com, [2] caiosf1@gmail.com, [3] silvanacunhacosta@gmail.com, [4] suzete.correia@gmail.com, [5] diasmichel@gmail.com. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. Av. 1º de Maio, 720, Jaguaribe, João Pessoa - PB, Brasil, CEP: 58015-430.

## RESUMO

Nas últimas décadas a tecnologia tem evoluído rapidamente com a grande demanda por equipamentos para processamento digital de sinais. O Digital Signal Processor (DSP) é um microcontrolador especializado, com alto poder de processamento, utilizado normalmente no tratamento de sinais de áudio e vídeo. Neste trabalho, é apresentada uma ferramenta didática para a implementação de filtros digitais Finite Impulse Response (FIR) no módulo de avaliação OMAP-L137/TMS320C6747. Para o projeto do filtro é utilizada a interface gráfica fdatool do MATLAB e para a implementação o software Code Composer Studio v3.3. Um filtro passa-baixas foi implementado e testado no processo de filtragem de um sinal de voz.

**Palavras-chave:** Processamento Digital de Sinais. Filtros Digitais FIR. DSP. OMAP-L137/TMS320C6747.

## ABSTRACT

*In the last few decades the technology has evolved fastly because of the high demand for equipments to digital signal processing. The Digital Signal Processor (DSP) is a specialized microcontroller with high processing capability usually employed in the processing of audio and video. In this paper, we present a methodology for the implementation of Finite Impulse Response (FIR) digital filters in the development kit OMAP-L137/TMS320C6747. The filter design is made at fdatool MATLAB GUI and its implementation with the Code Composer Studio v3.3 software. A low-pass filter was implemented and tested in the filtering process of a speech signal.*

**Keywords:** Digital Signal Processing. FIR Digital Filters. DSP. OMAP-L137/TMS320C6747.

## 1 Introdução

Os filtros digitais constituem uma das aplicações mais comuns no processamento de sinais de áudio e vídeo, em sistemas de telecomunicações e controle, devido as suas vantagens em relação aos filtros analógicos, tais como maior imunidade a ruídos, uso de tecnologia mais avançada e maior grau de automação (facilidade de projeto/síntese).

Entretanto, a tarefa do processamento digital de sinais é muito cara do ponto de vista computacional. Filtros digitais *Finite Impulse Response* (FIR), por exemplo, chegam a representar 80% do poder de computação requerido pela maioria das aplicações de telecomunicações. Consequentemente, os circuitos digitais que atendem esse tipo de aplicação devem cumprir requisitos de alto desempenho (BELLILE *et al.*, 1998).

Dessa forma, a presença de circuitos digitais otimizados para o processamento de sinais nos sistemas atuais é de relevada importância. Verifica-se, portanto, uma crescente demanda por circuitos cada vez mais densos e mais complexos, com altíssimas velocidades e implementados em tamanhos cada vez mais reduzidos. Nesse contexto, o *Digital Signal Processor* (DSP) surge como uma alternativa para o desenvolvimento de sistemas digitais complexos, permitindo uma nova maneira de abordar temas relacionados ao processamento digital de sinais, tais como o processamento de sinais de voz e imagens (AVELINO, 2009).

O DSP é um processador de uso específico de alto desempenho. Sua arquitetura é projetada para efetuar operações e manipulações matemáticas de dados de forma rápida, sendo capaz de gerar saídas em tempo real. Estas características do DSP permitem a implantação de algoritmos de filtragem com a flexibilidade de alteração dos parâmetros de controle por *software*, dispensando a interferência de parâmetros físicos dos componentes do processamento e fazendo com que os custos dessa tecnologia passem a ser atrativos (SILVA, 2008; GONÇALVES *et al.*, 2002; KEHTARNAVAZ, 2005).

Neste artigo é apresentada uma ferramenta didática para apoio à síntese automática de filtros digitais FIR, com base nas especificações dos filtros no domínio da frequência, e a subsequente implementação no OMAP-L137/TMS320C6747. Para o desenvolvimento da aplicação é utilizada, na etapa de projeto, uma interface gráfica disponível no MATLAB. Na etapa de

implementação é utilizado o Code Composer Studio v3.3, que é um ambiente de desenvolvimento da Texas Instruments para os DSPs de sua fabricação.

Este artigo está organizado da seguinte forma: na seção 2, encontram-se os métodos usados para a síntese e implementação de filtros digitais FIR na plataforma utilizada; na seção 3, os resultados experimentais são mostrados; e por fim, na seção 4, as considerações finais.

## 2 Materiais e métodos

Nesta seção são apresentados os aspectos teóricos referentes aos filtros digitais FIR e a metodologia empregada para a sua implementação na plataforma OMAP-L137/TMS320C6747.

### 2.1 Filtros Digitais FIR

A designação de filtro, habitualmente usada em referência aos sistemas lineares e invariantes no tempo (LIT), deriva da possibilidade dos sistemas eliminarem ou atenuarem fortemente certas harmônicas ou bandas de frequência. O sistema funciona, nesse caso, como um seletor de frequências. A banda onde o filtro praticamente não altera o sinal chama-se banda passante, em oposição a banda de rejeição, onde o sinal é eliminado ou fortemente atenuado.

Um filtro ideal é um sistema que seleciona perfeitamente as frequências desejadas e rejeita perfeitamente as frequências indesejadas em um sinal de entrada. No caso dos filtros não ideais há uma banda intermediária entre a banda de passagem e a de rejeição, chamada de banda de transição (HAYKIN, 2001).

Um sistema LIT pode ser descrito por uma equação de diferenças com coeficientes constantes, da forma:

$$\sum_{k=0}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k] \quad (1)$$

em que  $a_k$  e  $b_k$  são os coeficientes do filtro e  $N$  e  $M$  determinam a ordem do filtro.

Se o valor de  $N$  na Equação (1) for zero, a saída do sistema dependerá somente do valor atual e de valores anteriores da entrada, isto é:

$$y[n] = \sum_{k=0}^M b_k x[n-k] \quad (2)$$

Nesse caso, o sistema não contém termos recursivos e o filtro é denominado FIR (*Finite Impulse Response*) ou de resposta ao impulso finita. Filtros FIR são sempre estáveis e podem apresentar características de fase linear.

Uma forma bastante direta de obter um filtro FIR é truncando a resposta ao impulso infinita de um filtro ideal. Seja  $h_d[n]$  a resposta à amostra unitária de um filtro seletivo em frequência ideal de fase linear, então tem-se que:

$$H_d(e^{j\omega}) = A(e^{j\omega})e^{-j(\alpha\omega - \beta)} \quad (3)$$

Como em geral o comprimento de  $h_d[n]$  é infinito, é necessário encontrar uma aproximação finita para  $H_d(e^{j\omega})$ . Existem diversas técnicas para a síntese de filtros FIR. O método das janelas é um dos mais empregados, por sua simplicidade e por resultar em um filtro de desempenho muito bom (HAYES, 2006).

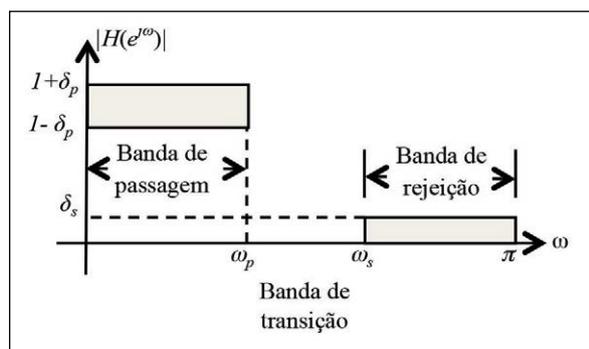
No projeto com janelas, o filtro FIR é obtido multiplicando-se a resposta à amostra unitária por uma janela,

$$h[n] = h_d[n]\omega[n] \quad (4)$$

em que  $\omega[n]$  é uma janela de comprimento finito que assume o valor zero fora do intervalo  $0 \leq n \leq N$  e é simétrica a seu ponto médio.

Para que um filtro passa-baixas seja projetado, é necessário definir um conjunto de especificações que incluem a frequência de corte da banda de passagem,  $\omega_p$ , a frequência de corte da banda de rejeição,  $\omega_s$ , a atenuação da banda de passagem,  $\delta_p$ , e a atenuação da banda de rejeição,  $\delta_s$ . O intervalo  $[\omega_p, \omega_s]$  é a banda de transição. A Figura 1 ilustra as especificações de um filtro passa-baixas.

Figura 1 – Especificações de um filtro passa-baixas.



Fonte: HAYES, 2006.

A janela de Kaiser é uma janela otimizada no sentido que fornece a menor ordem. Sua forma pode ser modificada pelo ajuste de um parâmetro  $\beta$ . Dessa forma, dependendo da aplicação, pode-se modificar a forma da janela para controlar a perda espectral. A Equação (5) define uma janela de Kaiser para  $0 \leq n \leq N-1$  amostras,

$$\omega[n] = \frac{I_0 \left[ \beta \left( 1 - \left[ 1 - \frac{2n}{N-1} \right]^2 \right) \right]}{I_0(\beta)} \quad (5)$$

em que  $\alpha = N/2$  e  $I_0$  é uma função de Bessel de ordem zero de primeira espécie que pode ser gerada facilmente usando uma expansão em série de potência, dada por:

$$I_0(x) = 1 + \sum_{k=1}^{\infty} \left[ \frac{(x/2)^k}{k!} \right]^2 \quad (6)$$

Há duas relações desenvolvidas para a janela de Kaiser que facilitam o seu uso no projeto de filtros digitais FIR. A primeira estabelece uma relação entre a atenuação da banda de rejeição de um filtro passa-baixas  $\delta_s$  (dB) e o parâmetro  $\beta$ ,

$$\beta = \begin{cases} 0,1102(\delta_s - 8,7), & \delta_s \text{ (dB)} > 50 \\ 0,5842(\delta_s - 21)^{0,4} + \\ 0,07886(\delta_s - 21), & 21 \leq \delta_s \text{ (dB)} \leq 50 \\ 0, & \delta_s \text{ (dB)} \leq 21 \end{cases} \quad (7)$$

A segunda relaciona  $N$  com a largura de transição ( $\Delta f$ ) e com a atenuação da banda de rejeição  $\delta_s$  dado por:

$$N = \frac{\delta_s - 7,95}{14,36\Delta f}, \quad \text{se } \delta_s > 21 \text{ dB} \quad (8)$$

## 2.2 OMAP-L137/ TMS320C6747

O módulo de avaliação OMAP-L137/ TMS320C6747 é uma plataforma de desenvolvimento de baixo custo, da *Spectrum Digital Inc.*, que permite aos usuários avaliar e desenvolver aplicações de alta precisão para o processador OMAP-L137.

Essa plataforma integra: um dispositivo OMAP-L137, da *Texas Instruments*, com um processador em ponto flutuante DSP C674xVLIW; um processador ARM926EJ-S com frequência de funcionamento de até 300 MHz; um codificador estéreo TLV320AIC3106, da *Texas Instruments*, para a entrada e saída de áudio; 64 MB de memória SDRAM; um emulador JTAG, para melhorar o desempenho da comunicação; e 64 MB de memória SDRAM. A comunicação pode ser feita através de interfaces seriais USB e RS-232 ou Ethernet (TEXAS INSTRUMENTS, 2012).

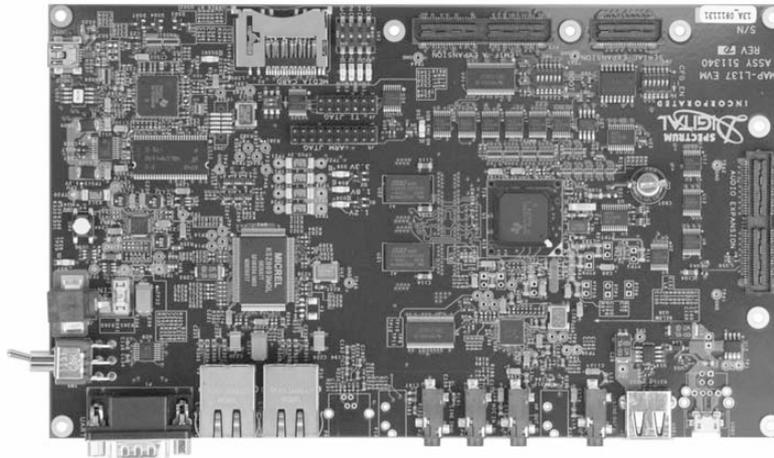
Na Figura 2 é ilustrada uma foto da placa original do módulo de avaliação OMAP-L137/TMS320C6747 e na Figura 3 seu diagrama em blocos com os principais componentes de *hardware*.

### 2.3 Projeto e Implementação de Filtros Digitais FIR

A metodologia empregada para o projeto e implementação de filtros digitais na plataforma de desenvolvimento OMAP-L137/TMS320C6747 consiste nas etapas descritas a seguir:

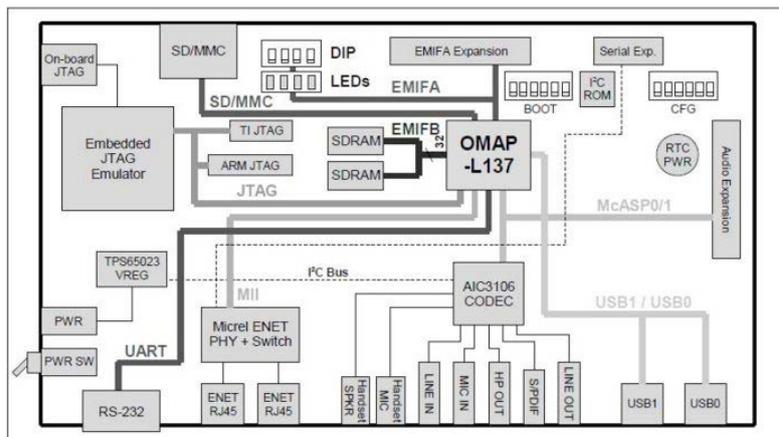
- I. Especificação das frequências e atenuações (dB) da banda de passagem e da banda de rejeição de acordo com a aplicação;
- II. Projeto do filtro através do método das janelas;
- III. Transcrição do projeto para *software* e *hardware*.

Figura 2 – Módulo de Avaliação do OMAP-L137/TMS320C6747.



Fonte: TEXAS INSTRUMENTS, 2012.

Figura 3 – Diagrama em blocos com os principais componentes de hardware do OMAP-L137/TMS320C6747.



Fonte: TEXAS INSTRUMENTS, 2012.

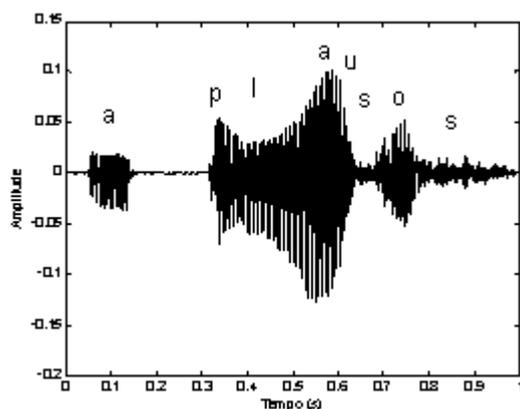
Para o projeto do filtro foi empregada a ferramenta *fdatool* do MATLAB, que possui uma interface gráfica para projeto de filtros digitais. Nesse aplicativo pode-se escolher o tipo de filtro digital a ser projetado (FIR ou IIR), o método empregado para o projeto, a ordem requerida do filtro e as especificações desejadas das atenuações e frequências das bandas de passagem e rejeição. Como resposta, é fornecida a ordem e os coeficientes para a implementação do filtro em *software* ou *hardware*.

Um código na linguagem de programação C foi desenvolvido para calcular a equação de diferenças da Equação (2), empregando os coeficientes e a ordem do filtro obtidos através da ferramenta *fdatool* do MATLAB. Esse código é integrado ao *software* de edição *Code Composer Studio (CCS) v3.3*, que compila, adiciona as bibliotecas necessárias e gera um código executável que é embarcado no módulo de avaliação OMAP-L137/TMS320C6747. A partir de então, o mesmo será executado inteiramente na placa da plataforma OMAP-L137. No entanto, é possível acessar o processo em execução através do depurador do CCS.

### 3 Resultados e Discussão

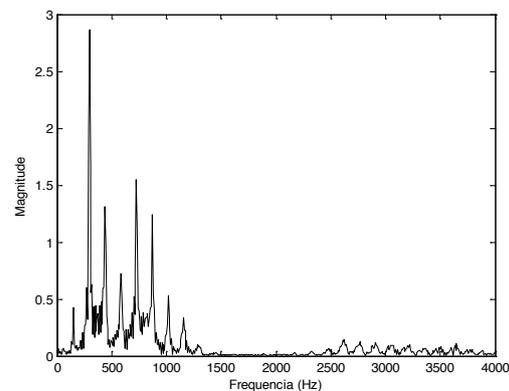
Como exemplo, foi projetado e implementado um filtro digital FIR passa-baixas para filtrar um sinal de voz, referente à palavra aplausos, amostrado a uma taxa de 8000 amostras/s e com amplitude normalizada entre [-1,1]. As Figuras 4 e 5 ilustram o sinal de voz no domínio do tempo e o espectro da sua magnitude, respectivamente. O sinal de voz foi previamente capturado usando o *software GoldWave*, em sua versão demo.

Figura 4 – Sinal de voz no domínio do tempo.



Fonte: Elaborada pelo autor.

Figura 5 – Espectro da magnitude do sinal de voz.



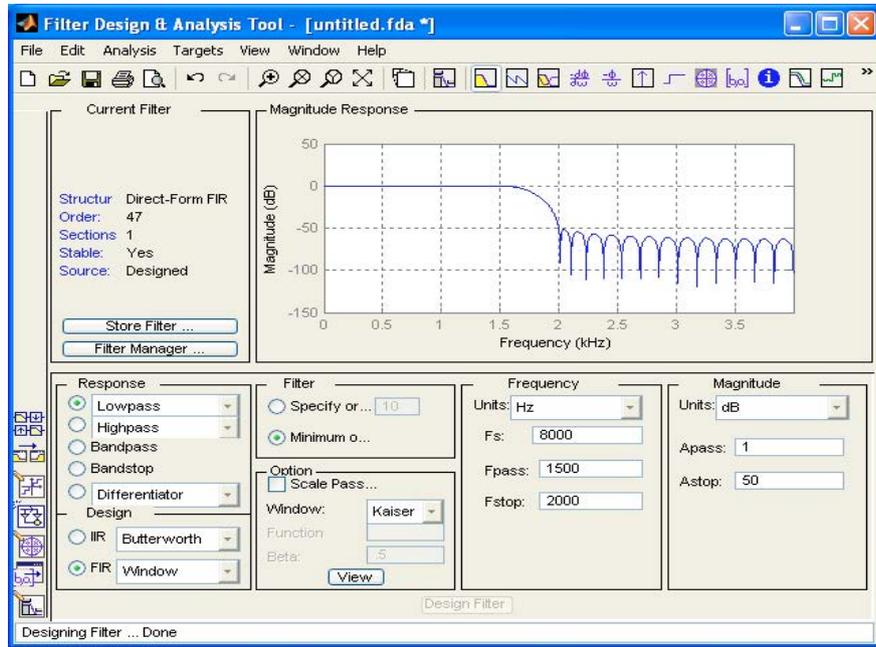
Fonte: Elaborada pelo autor.

Observa-se, na Figura 4, a presença de harmônicas indesejadas na frequência acima de 2,5 kHz, provenientes de ruído ambiente introduzido durante o processo de aquisição. Com o intuito de eliminar as harmônicas indesejadas, foi especificado um filtro passa-baixas FIR com uma frequência de 1,5 kHz para a banda de passagem, 2 kHz para a banda de rejeição e atenuações de 1 dB e 50 dB, para as bandas de passagem e rejeição, respectivamente. Foi escolhido para o projeto do filtro o método das janelas, empregando-se a janela de Kaiser. Na Figura 6 é ilustrada a interface do *fdatool* do MATLAB para o filtro passa-baixas implementado.

O programa gerou um filtro FIR de ordem 47 e seus coeficientes foram extraídos e dados como entrada no código em C desenvolvido para implementar o filtro especificado no *software* de edição *Code Composer Studio v3.3*. A Figura 7 apresenta a janela do CCS com o código em C do filtro passa-baixas. O vetor *entrada* recebe os valores do sinal de voz digitalizado, o vetor *b* os coeficientes do filtro e o vetor *saída* recebe o sinal de voz filtrado.

Após a compilação, o código gerado pelo CCS foi embarcado no DSP. Os valores armazenados no endereço de memória do vetor de saída foram salvos em um arquivo e plotados no MATLAB. As Figuras 8 e 9 ilustram o sinal obtido com a filtragem e o espectro da sua magnitude, respectivamente. Observa-se que o sinal obtido no domínio do tempo apresenta similaridade com o sinal original. A eficácia do filtro é mais perceptível observando-se o espectro do sinal filtrado, tendo em vista que o ruído de alta frequência foi eliminado. O fato é percebido, ainda, no teste de escuta em que foi percebida uma melhor qualidade no sinal resultante.

Figura 6 – Interface da ferramenta fdatool do MATLAB para o filtro passa-baixas implementado.



Fonte: TEXAS INSTRUMENTS, 2012.

Figura 7 – Código em C do filtro digital FIR no Code Composer Studio v3.3.

```

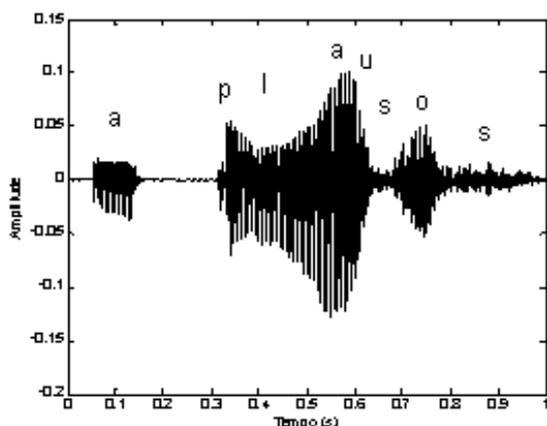
#include <stdio.h>

float entrada[8199]; /* vetor de entrada */
int N=8199; /* número de amostras do sinal de voz */
float saida[8199]; /* vetor de saída */
int ordem=47; /* ordem do filtro */
float b[48]={0.00058154,-0.00054671,-0.0015972,0.00022492,0.0030375,0.0011492,-0.0044403,
-0.0039981,0.0049405,0.008399,-0.0033526,-0.01385,-0.0016362,0.01911,0.011257,-0.022119,
-0.026644,0.019762,0.049577,-0.0064398,-0.086566,-0.036132,0.18563,0.4035,0.4035,0.18563,
-0.036132,-0.086566,-0.0064398,0.049577,0.019762,-0.026644,-0.022119,0.011257,0.01911,
-0.0016362,-0.01385,-0.0033526,0.008399,0.0049405, -0.0039981, -0.0044403, 0.0011492,
0.0030375,0.00022492,-0.0015972,-0.00054671,0.00058154}; /* coeficientes do filtro */
int i=0;
float yo=0;
int M;
int k=0;

void main()
{
while(1)
{
for(i=0;i<N;i++){
yo=0;
if(i<ordem+1){
M=i;
}
else{M=ordem+1;}
for(k=0;k<M;k++){
yo=yo+b[k]*entrada[i-k];
saida[i]=yo;
}
}
}
}
    
```

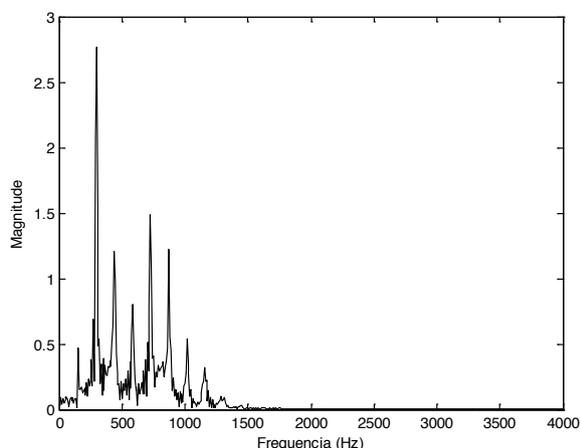
Fonte: TEXAS INSTRUMENTS, 2012.

**Figura 8** – Sinal de voz filtrado no domínio do tempo.



Fonte: Elaborada pelo autor.

**Figura 9** – Espectro da magnitude do sinal de voz filtrado.



Fonte: Elaborada pelo autor.

## 4 Considerações finais

O projeto de filtros digitais, em relação aos filtros analógicos, usando DSP, apresenta como vantagem a possibilidade de alteração do projeto, sem a necessidade de mudança de *hardware*. Além disso, com a grande demanda por sistemas embarcados, o DSP vem sendo amplamente empregado nas mais diversas aplicações, tais como o processamento de sinais de voz e imagens. A maior contribuição deste trabalho consiste em apresentar uma metodologia para o desenvolvimento de projetos de filtros digitais na plataforma OMAP-L137/TMS320C6747. O método apresentado pode ser facilmente adaptado e aplicado

a outros tipos de projetos que demandem processamento digital de sinais, a exemplo de sistemas de reconhecimento de voz e sistemas de identificação de locutor, que necessitam de sinais livres do ruído ambiente para um melhor desempenho.

## REFERÊNCIAS

- AVELINO, M. A. **Processamento Embarcado aplicado a um Sistema de Detecção de Vazamentos**. 2009. Dissertação (Mestrado em Engenharia Elétrica e Computação) – Universidade Federal do Rio Grande do Norte, Natal, 2009.
- BELLILE, O.; DUJARDIN, E. **Architecture of a Programmable FIR Filter Co-Processor**. In: Proceedings of IEEE International Symposium on Circuits and Systems, Monterey, CA, 1998, p. 433-436.
- GONÇALVES, J.; MARQUES, L.; SALVADOR, J.; SOUSA, L. **Ferramenta de Síntese e Implementação de Filtros em DSP**. In: Actas da CCTE'2002 – Conferência Científica e Tecnológica em Engenharia, ISEL, Lisboa, 2002.
- HAYES, M. H. **Processamento Digital de Sinais – Coleção Schaum**. Porto Alegre: Bookman, 2006, 466 p.
- HAYKIN, S; BARRY, V. **Sinais e Sistemas**. Porto Alegre: Bookman, 2001, 668 p.
- KEHTARNAVAZ, N. **Real-Time Digital Signal Processing based on the TMS320C6000**. Elsevier: Newnes, 2005.
- SILVA, A. P. D. **Uso do DSP na Instrumentação Associada à Medição Distribuída de Temperatura**. Monografia (Especialização em Engenharia de Instrumentação) – UFPE, Recife, 2008.

TEXAS INSTRUMENTS. **Code Composer Studio Development Tools v3.3 - Getting Started Guide**. Disponível em <<http://www.ti.com/general/docs/litabsmultiplefilelist.tsp?literatureNumber=spru509h>>. Acesso em: ago. 2012.

## AGRADECIMENTOS

Os autores agradecem ao IFPB pelo suporte técnico e ao CNPq pelo apoio financeiro.