

Geração de imagem de ultrassom em Modo B usando o Raspberry Pi

B-mode ultrasound imaging using the Raspberry Pi

RESUMO

Renan Antonio Corrêa Medeiros
renan4_acm@hotmail.com
Universidade Tecnológica Federal
do Paraná, Curitiba, Paraná, Brasil

Amauri Amorin Assef
amauriassef@utfpr.edu.br
Universidade Tecnológica Federal
do Paraná, Curitiba, Paraná, Brasil

Joaquim Miguel Maia
joaquim@utfpr.edu.br
Universidade Tecnológica Federal
do Paraná, Curitiba, Paraná, Brasil

A técnica do ultrassom em Modo B representa uma das principais modalidades de geração de imagem para auxílio ao diagnóstico médico. Para melhorar a qualidade da imagem gerada, são esperadas novas abordagens e técnicas de processamento digital de sinais baseadas em plataformas de hardware e software. Este artigo apresenta a implementação e avaliação dos algoritmos de processamento digital de sinais no sistema embarcado Raspberry Pi utilizando a linguagem de programação Python para reconstrução de imagens de ultrassom em Modo B. As etapas propostas incluem filtragem digital, atraso de focalização, somatório coerente, demodulação com detecção de envoltória e compressão logarítmica. Para validar os algoritmos implementados, foram utilizados dados de ultrassom com 12 bits amostrados com frequência de 40 MHz. As análises qualitativas e quantitativas utilizando a função de custo da raiz quadrada do erro quadrático médio normalizado (NRMSE), demonstram que os resultados são compatíveis com o modelo de referência adotado no Matlab e validado em estudos prévios. Todos os resultados do NRMSE foram menores que 5%, indicando uma excelente concordância com o modelo de referência do Matlab.

PALAVRAS-CHAVE: Ultrassom. Imagem em Modo B. Raspberry Pi.

Recebido: 19 ago. 2020.

Aprovado: 01 out. 2020.

Direito autoral: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



ABSTRACT

B-Mode ultrasound imaging represents one of the main imaging methods in medical diagnosis. To improve the quality of generated images, new approaches and techniques for digital signal processing based on hardware and software platforms are being introduced nowadays. This article shows the implementation and evaluation of digital signal processing algorithms on the Raspberry Pi using Python programming language for B-mode image reconstruction. The proposed steps include digital filtering, focusing delay, coherent summation, demodulation with envelope detection, and logarithmic compression. To validate the implemented algorithms, 12-bit sampled ultrasound data at frequency of 40 MHz were used. Qualitative and quantitative analyses using the Normalized Root Mean Squared Error (NRMSE) cost function show results compatible with the reference model in Matlab and validated in previous studies. All NRMSE results were less than 5%, indicating excellent agreement with the reference Matlab model.

KEYWORDS: Ultrasound. B-mode imaging. Raspberry Pi.



INTRODUÇÃO

No Brasil existe um grande potencial para desenvolver no âmbito acadêmico o setor de equipamentos de imagem médica e, conseqüentemente, contribuir para o desenvolvimento científico, tecnológico e inovador na área de Engenharia Biomédica, especificamente no diagnóstico por imagem a partir do ultrassom (US). Nesse sentido, o crescente desenvolvimento tecnológico computacional pode auxiliar na implementação de novos algoritmos de processamento digital em novas plataformas de hardware e software, permitindo aprimorar a qualidade e a pesquisa dos métodos de reconstrução de imagens de US.

Neste trabalho é apresentada a implementação e avaliação do processamento digital de sinais de US para geração de imagem no Modo B, utilizando um módulo Raspberry Pi e a linguagem de programação Python. Durante o desenvolvimento da pesquisa foi abordado o estudo das características dos sinais de US para formação da imagem em Modo B e as etapas do processamento digital para reconstrução da imagem de US, destacando-se a filtragem digital, demodulação com detecção de envoltória, compressão logarítmica e conversão de varredura.

MATERIAL E MÉTODOS

Para o desenvolvimento do trabalho foi utilizado um Raspberry Pi 3 Modelo B, ilustrado na Figura 1. Esse dispositivo foi desenvolvido pela organização inglesa sem fins lucrativos Fundação Raspberry Pi, cujo principal objetivo é estimular o ensino da ciência da computação. O módulo apresenta as seguintes especificações principais: Processador Quad Core de 1,2 GHz Broadcom BCM2837, CPU (*Central Processing Unit*) de 64 *bits* e Memória RAM (*Random Access Memory*) de 1 GB (RASPBERRYPI, 2020).

Figura 1 – Módulo Raspberry Pi 3 Modelo B.



Fonte: Raspberry Pi (2020).

Os dados de US utilizados para avaliação das etapas de processamento descritas a seguir foram capturados e digitalizados com uma frequência de amostragem de 40 MHz em 12 *bits* por um sistema proprietário (ASSEF, 2013) e enviados para um computador com o *software* Matlab instalado. Esses sinais foram adquiridos através de um *phantom* de US modelo 84-317 e um transdutor multielemento convexo AT3C52B.

A primeira etapa descrita no algoritmo desenvolvido foi a filtragem digital, que é utilizada para eliminar a componente contínua e os ruídos da radiofrequência de eco (JINBO, 2007). Utilizou-se neste trabalho um filtro de

resposta ao impulso finita (FIR – *Finite Impulse Response*), no qual a resposta ao impulso tem um número finito de amostras que são referidas como os coeficientes do filtro (MADISETTI; WILLIAMS, 1998). Os filtros FIR são representados matematicamente por uma sequência de multiplicações e somas, conforme a Eq. (1):

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

no qual $x[n]$ é o sinal de entrada, $y[n]$ é o sinal de saída, n é o índice da amostra, a_k são os coeficientes do filtro e M é a ordem do filtro (JI-FENG et al., 2015).

O projeto do filtro digital foi desenvolvido com a ferramenta FDATool do Matlab. As especificações utilizadas são apresentadas na Tabela 1, considerando a frequência central do transdutor igual a 3,2 MHz. Para a avaliação do filtro, utilizou-se um sinal do tipo *chirp* com frequência de 1 MHz a 10 MHz.

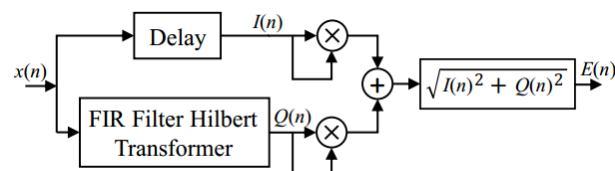
Tabela 1 – Especificações do filtro digital.

| Parâmetro | Especificação |
|---------------------------------|------------------------------|
| Frequência de amostragem | 40 MHz |
| Método do projeto | FIR - Generalized Equiripple |
| Tipo de resposta | Passa-baixa |
| Frequência da banda passante | 3,2 MHz |
| Frequência da banda de rejeição | 8,0 MHz |
| Atenuação na banda passante | -1 dB |
| Atenuação na banda de rejeição | -50 dB |

Fonte: Autoria própria (2020).

Para geração de imagens por US em Modo B, cujo brilho é modulado de acordo com a amplitude do sinal em Modo A, são necessárias as operações de demodulação seguida pela detecção de envoltória para extrair do sinal soma coerente a linha de varredura final da faixa de frequência da portadora do eco (JINBO, 2007; ASSEF, 2013). Um dos métodos de demodulação é através da Transformada de Hilbert baseada no filtro FIR, que foi utilizada neste trabalho por ser mais eficiente e precisa (LEVESQUE; SAWAN, 2009). A envoltória do sinal é obtida através do valor absoluto da Transformada de Hilbert do sinal que é calculado pela raiz quadrada da soma dos quadrados das componentes em fase $I(n)$ e em quadratura $Q(n)$ (OPPENHEIM; SCHAFER, 1975). O diagrama em blocos, ilustrado na Figura 2, representa o funcionamento dessa etapa.

Figura 2 – Diagrama de blocos para detecção de envoltória.



Fonte: Assef et al. (2019).

O projeto da etapa de demodulação também foi desenvolvido com a ferramenta FDATool do Matlab. As principais especificações utilizadas são

apresentadas Tabela 2. Nas demais especificações necessárias, foram utilizados os valores padrão da ferramenta.

Tabela 2 – Especificações do filtro FIR utilizando a Transformada de Hilbert.

| Parâmetro | Especificação |
|-------------------|----------------------------|
| Método do projeto | FIR – <i>Equiripple</i> |
| Tipo de resposta | <i>Hilbert Transformer</i> |
| Ordem do filtro | 32 |

Fonte: Autoria própria (2020).

A compressão logarítmica que é utilizada para adequar a faixa dinâmica da linha de varredura nos níveis da escala de cinza para visualização no Modo B, melhorando o contraste da imagem (FERREIRA, 2017). Na compressão logarítmica, os valores dos dados de US são mapeados por uma função logarítmica, através da Eq. (2), na qual $E(n)$ é o sinal da envoltória e $Ec(n)$ é o sinal da envoltória comprimido, dado em decibéis (dB) (JINBO, 2007).

$$Ec(n) = 20 \log_{10}(E(n)) \quad (2)$$

Devido a incompatibilidade do sistema de coordenadas da varredura, normalmente utilizada nos sistemas de US, com o sistema de coordenadas de visualização, é necessário a realização da etapa de conversão de varredura. Neste trabalho, a correção de coordenadas foi realizada em um computador com o *software* Matlab e a demais etapas no Raspberry Pi.

Ao final, os resultados do processamento digital no Raspberry Pi foram comparados com os resultados de referência apresentados previamente através de *scripts* do Matlab no trabalho de Assef (2013). Além da análise qualitativa, a análise quantitativa foi realizada utilizando a função de custo NRMSE (*Normalized Root Mean Squared Error*). O NRMSE é considerado excelente para valores menores de 10%, bom para 11% a 20%, justo para 21% a 30%, e ruim para valores acima de 30%, sendo calculado por:

$$NRMSE = \sqrt{\frac{\sum_{n=0}^{M-1} |Y(n) - hy(n)|^2}{\sum_{n=0}^{M-1} |hy(n) - \overline{hy}|^2}} * 100, \quad (3)$$

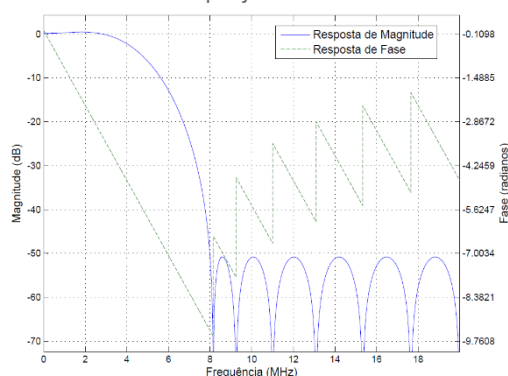
no qual n é o índice de amostra, M é o número de amostras, $Y(n)$ é a informação obtida por meio do *script* de referência do Matlab, $hy(n)$ é a resposta do Raspberry e \overline{hy} é o valor médio de $hy(n)$ (ASSEF et al., 2019).

RESULTADOS E DISCUSSÃO

A partir das especificações filtro digital (Tabela 1), determinou-se um filtro com a menor ordem. As respostas de magnitude e fase são ilustradas na Figura 3. Para verificar o funcionamento do filtro projetado, os valores dos 16 coeficientes foram exportados para um algoritmo desenvolvido em Python na plataforma Raspberry Pi. No algoritmo são implementados os somatórios e as multiplicações necessárias para descrever a estrutura do filtro FIR.

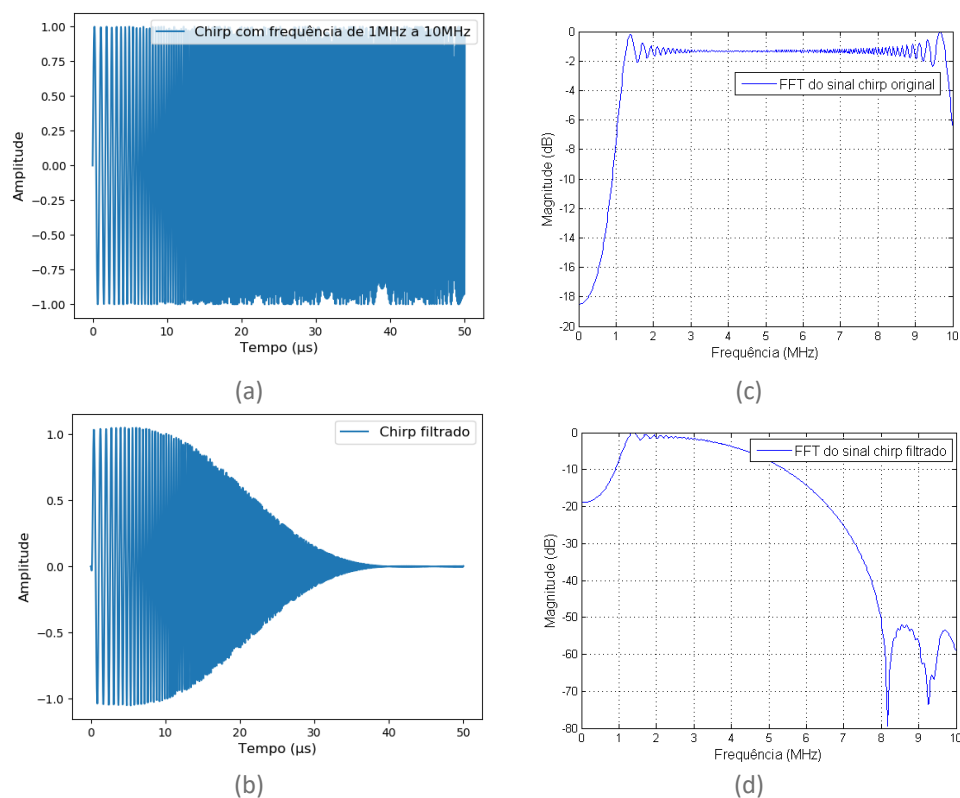
Na Figura 4(a) e Figura 4(c) são apresentados o sinal do tipo *chirp* com frequência de 1 MHz a 10 MHz e a respectiva FFT. Na Figura 4(b) e Figura 4(d) são mostrados o resultado da filtragem digital do sinal *chirp* e a respectiva FFT.

Figura 3 – Respostas em frequência da magnitude e fase do filtro FIR passa-baixa projetado.



Fonte: Autoria própria (2020).

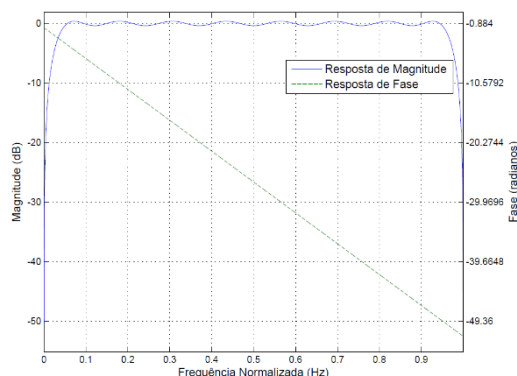
Figura 4 – (a) Sinal do tipo *chirp* original e (b) filtrado. (c) FFT do sinal *chirp* original e do (d) sinal *chirp* filtrado.



Fonte: Autoria própria (2020).

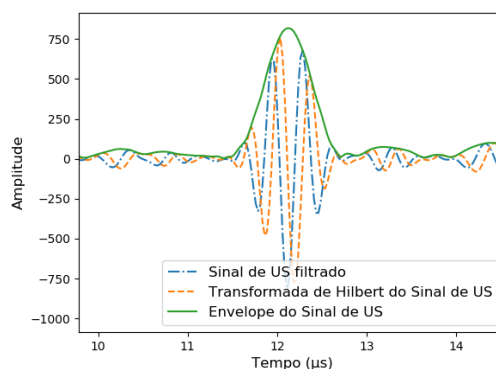
A partir das especificações do filtro FIR usando a Transformada de Hilbert (Tabela 2), obteve-se as respostas de magnitude e fase para a etapa de demodulação, conforme a Figura 5. Para verificar o funcionamento da detecção de envoltória, os valores dos coeficientes foram exportados para o mesmo algoritmo desenvolvido no Raspberry Pi em Python do filtro digital, dando continuidade ao processamento. No algoritmo, implementou-se a estrutura da Transformada de Hilbert baseada no filtro FIR e o equacionamento matemático necessário. Como exemplo de resultado, na Figura 6 são ilustrados o sinal de US filtrado, a Transformada de Hilbert do sinal de US e o envelope resultante do processamento.

Figura 5 – Respostas em frequência da magnitude e fase do filtro FIR usando a Transformada de Hilbert.



Fonte: Autoria Própria (2020).

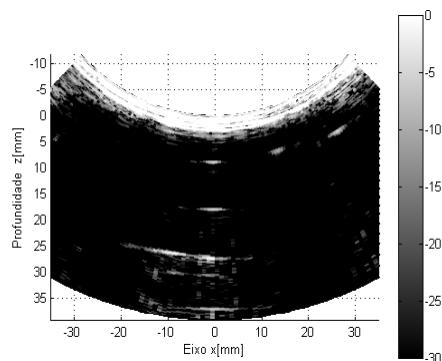
Figura 6 – Envoltória do sinal de ultrassom obtida a partir da Transformada de Hilbert no Raspberry Pi.



Fonte: Autoria Própria (2020).

Após os testes individuais das etapas de filtragem digital e detecção de envoltória, realizou-se o processamento completo dos dados de US para geração de imagem. Com os dados brutos de US processados completamente no Raspberry Pi, os sinais resultantes foram exportados para a área de trabalho do Matlab, no qual realizou-se a etapa final de conversão de varredura. Na Figura 7 é mostrada a imagem final de US com a conversão de varredura. Conforme a barra de cor lateral, a faixa dinâmica é de -30 dB.

Figura 7 – Imagem final de ultrassom após a conversão de varredura.



Fonte: Autoria Própria (2020).

O resultado do cálculo da função de custo NRMSE para comparação entre o resultado do processamento embarcado no Raspberry Pi, proposto neste trabalho, com o modelo usado de referência foi de 3,43e-10% para a etapa de soma coerente, 3,58% para a detecção de envoltória e 4,35% para a compressão logarítmica.

O algoritmo implementado no Raspberry Pi que realiza a etapa de filtragem digital, envolveu a aplicação de um filtro FIR passa-baixa com 16 coeficientes. O resultado foi satisfatório, apresentando atenuação das componentes com frequência superior à frequência da banda de rejeição. Verificou-se principalmente a atenuação acima de 50 dB da componente de 8 MHz, conforme é apresentado na Figura 4. No algoritmo que descreve a etapa de detecção de envoltória baseada no filtro FIR a partir da Transformada de Hilbert, o envelope do sinal de US foi obtido com sucesso no Raspberry Pi (Figura 6).

Através dos resultados, constatou-se que não é possível a geração de imagens em tempo real, devido às limitações do Raspberry Pi 3 Modelo B, principalmente a frequência de operação e capacidade de memória. O tempo total para todas as etapas de processamento propostas e embarcadas no Raspberry Pi foi de aproximadamente 27 minutos.

CONCLUSÃO

Conclui-se que o objetivo principal deste trabalho, que foi realizar o processamento digital de sinais para geração de imagem no Modo B utilizando um módulo Raspberry Pi 3 Modelo B, foi obtido com sucesso. Ambas as análises qualitativas e quantitativas, utilizando a função de custo NRMSE, demonstram que o algoritmo em Python implementado no Raspberry Pi, apresenta resultados compatíveis com o modelo de referência adotado e validado em estudos prévios (ASSEF, 2013). Todos os resultados do NRMSE foram menores que 5%, indicando uma excelente concordância entre os modelos. Apesar o tempo de processamento não ser adequado para geração de imagens com elevada taxa de quadros, abre-se uma nova possibilidade de processamento de sinais de US embarcado no Raspberry Pi com a utilização da linguagem de Python, haja vista que não foram encontrados na literatura e demais fontes pesquisadas, até o momento, códigos de programação para essa finalidade. Dessa forma, este trabalho apresenta resultados que podem ser explorados em outros trabalhos futuros pela comunidade científica.

AGRADECIMENTOS

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), à Financiadora de Estudos e Projetos (FINEP), à Fundação Araucária e à UTFPR pelo apoio financeiro no desenvolvimento da pesquisa.

REFERÊNCIAS

ASSEF, A. A. **Arquitetura de Hardware Multicanal Reconfigurável com Excitação Multinível para Desenvolvimento e Testes de Novos Métodos de Geração de**

Imagens por Ultrassom. Tese (Doutorado em Ciências) – Universidade Tecnológica Federal do Paraná, Curitiba, 2013.

ASSEF, A. A.; OLIVEIRA, J. DE.; MAIA, J. M.; COSTA, E. T. **FPGA implementation and evaluation of an approximate Hilbert Transform-based envelope detector for ultrasound imaging using the DSP builder development tool.** 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, p. 2813-2816, 2019.

FERREIRA, BRENO M. A. **Modelagem e Implementação de um Sistema de Processamento Digital de Sinais baseado em FPGA para Geração de Imagens por Ultrassom usando o Simulink.** Dissertação (Mestrado em Sistemas de Energia) – Universidade Tecnológica Federal do Paraná (UTFPR), Curitiba, 2017.

JI-FENG, D.; SHUANG, X.; ZHANG, J. X.; YA-NING, Y. **Filter Design Based on DSP Builder.** Open Electrical & Electronic Engineering Journal, v. 9, p. 15-21, 2015.

JINBO, M. **Imagens ultrassônicas em modo-B com técnicas de abertura focal sintética – SAFT.** Dissertação (Mestrado em Engenharia) – Universidade de São Paulo (USP), São Paulo, 2007.

LEVESQUE, P.; SAWAN, M. **Real-Time Hand-Held Ultrasound Medical-Imaging Device Based on a New Digital Quadrature Demodulation Processor.** IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control, v.56, n. 8, p. 1654-1665, 2009.

MADISETTI, V. Y. K.; WILLIAMS, D. **Digital Signal Processing Handbook.** Boca Raton, Fla.: CRC Press, 1998.

OPPENHEIM, A. V.; SCHAFER, R. W. **Digital signal processing.** New Jersey: Prentice-Hall, 1975.

RASPBERRYPI. **Raspberry Pi 3 Model B.** Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. Acesso em: 31 ago. 2020.