Evaluation of Singular Value Decomposition (SVD) for Clutter Suppression Implemented in Simulink for Commercial Ultrasound

Tarek Kaddoura[1], Mayara Nascimento de Oliveira[1] and Gonzalo Rubio[2]

[1] Department of Electrical and Computer Engineering, University of Alberta

[2]Department of Physics, University of Alberta

April 9th, 2018

ABSTRACT

A Simulink model for clutter-suppression of in-vivo ultrasound data for blood flow imaging is presented. The proposed system compares the performance of the Singular-Value-Decomposition (SVD) filter against the Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters. We acquire 3D ultrasound data and apply temporal filters as well as a spatiotemporal filter as a means to supress clutter. The isolated blood signal is then used to calculate the signal-to-clutter ratio to evaluate the performance of the filters. We demonstrate a considerable increase in SCR of 10 dB when applying SVD clutter filtering versus applying conventional high-pass clutter filters. The results obtained could potentially have substantial implications in real-time blood flow imaging of areas of interest where tissue motion is no longer negligible, thus making temporal filters inadequate. In this study, both 2D spatial and temporal coherence are used to filter unwanted signals. This process can potentially have application in imaging microvasculature of skin, muscles or tumors; as well as in areas with considerable tissue motion such as cardiac or abdominal imaging.

Contents

INTRODUCTION 1
Background 1
Related work 1
THEORY 2
Singular Value Decomposition (SVD) Filter 2
Finite Impulse Response (FIR) Filters 3
Infinite Impulse Response (IIR) Filters 3
METHODS 4
Data Acquisition 4
Simulink Implementation 4
Singular Value Decomposition 4
FIR and IIR Filters 8
RESULTS and DISCUSSION 9
FIR and IIR Filter Parameters 9
Blood Power Images Comparison 10
Performance Metrics 11
Application: Calculating heart rate from blood power images 12
CONCLUSION 12
REFERENCES 12

INTRODUCTION

Background

Ultrasound imaging consists of basic ultrasound propagation within a medium. When a signal (high frequency compressional waves) is transmitted from a transducer into the human body, the signal will be reflect back to the receiver from different internal structures [1]. The reflections received on a receiver are then processed to form an image of the reflected structures. The common carotid arteries (carotids) are two main arteries that supply oxygenated blood to the head and neck of the human body and are of high clinical significance. For instance, they are commonly used in measuring the pulse in patients who lack a detectable pulse in other arteries of the body, evaluation of systolic blood pressure and diagnosis of carotidnya and atherosclerosis. Carotid ultrasound imaging is done by propagating sound waves with a standard frequency range of 0.5MHz to 20MHz, typically through the patient's neck, recording and processing the signal to then produce series of images (or a video) of the carotid artery. This procedure is commonly done for screening of blockages, narrowing of the carotid arteries and blood clots, conditions which could severely increase a patient's risk of stroke. Since ultrasound images are captured in real-time, they can show the structure of dynamic and static components of the body's internal organs. In other words, we can obtain an image of the tissue surrounding the carotid, as well as the blood flow through it.

Over the past three decades, various filtering techniques have been developed and tested to remove unwanted tissue signal from both stationary and slow-moving tissue to obtain higher quality images of blood flow in an area of interest. In blood flow imaging, this can be an incredibly challenging task because both tissue and blood scatterers can be almost identical in extreme cases such as when blood flow velocities are considerably slow or when the tissue surrounding a vein or vessel has considerably motion when compared to the blood flow velocity [2]. The substantial number of different filters make it difficult for researchers to decide which one to use. In this study we aim to compare three different models of clutter-suppression filters for detecting blood flow, (1) the singular value decomposition (SVD) filter, (2) the high-pass FIR filter, and (3) the high-pass IIR filter. By implementing the SVD and other filters in Simulink, we want to be able to effectively suppress stationary tissue signal and image the blood flow through the carotid [2].

Related work

Our research is inspired by past studies on clutter filters [2-4], and is novel in the sense that it is implemented in Simulink. Additionally, we do not consider microvasculature like in [3] but rather we investigate the carotid arteries in the neck which are major arteries in the human body. The study performed by [2] uses a much higher frequency (24 MHz) and conversely, we used a central frequency of 5 MHz. The same study also performs Higher Order Singular Value Decomposition (HOSVD) using a 4D array of echo data. In this study, we perform conventional SVD on 3D data. The study performed by R. Fox [3] analyzes the frequency spectrum of the signal obtained from a transducer to design a high-pass temporal filter in MATLAB, whereas our study visualized the spectrum and designed the high-pass clutter filter in Simulink. Furthermore, [4] performs SVD filtering in a phantom study and detected previously undetected blood flows. In contrast with their study, our system is not tested on a phantom study and is instead tested in-vivo, meaning that the exact locations and dimensions of our targets are not exactly known. Owing to our use of Simulink, our approach allows for flexible control over filter design parameters using Simulink blocks and top-level implementations that can be visualized in real-time. This research is the first we know of to implement the SVD filter on in-vivo ultrasound data using the Simulink platform.

THEORY

In this section, we explain the theory behind the three different clutter filters we compare.

Singular Value Decomposition (SVD) Filter

A matrix with full rank can be approximated by another with lower rank through Singular Value Decomposition (Eckart and Young, 1936). Assume that we have a matrix $\boldsymbol{A}$ with rank r that can be approximated to one of lower rank $\widetilde{\boldsymbol{A}} = \boldsymbol{U}_s\boldsymbol{\Sigma}_s\boldsymbol{V}^{\dagger}, s < r$, where † denotes the conjugate transpose, $\boldsymbol{U}_s$ and $\boldsymbol{V}_s$ are orthogonal matrices and $\boldsymbol{\Sigma}_s$ is a diagonal matrix of singular values σ_i sorted in descending order. The $\widetilde{\boldsymbol{A}}$ matrix is the best approximation of $\boldsymbol{A}$ in the least-squares sense, $\|\boldsymbol{A} - \widetilde{\boldsymbol{A}}\|_2^2$.

For the implementation of the SVD filter, one needs to first reshape the acquired 3D data to a spatiotemporal matrix form [4]. This is achieved by transforming the 3D data set consisting of dimensions (N_z, N_x, N_t) into a 2D spatiotemporal matrix of dimensions $(N_z \times N_x, N_t)$. One can now apply singular value decomposition (SVD) to this matrix to deconstruct it to its eigenimages.

It is commonly assumed that tissue scattering is more echogenic than blood scattering (Kim et al., 2017). In other words, tissue has a greater ability to reflect the incoming signal than blood. It is also assumed that tissue and blood contribute more to the variance than noise. Therefore, we can assume that the first eigenvalues in $\boldsymbol{\Sigma}_s$ belong to the tissue, followed by blood and noise. We can obtain an image of the blood signal by simply zeroing the tissue and noise eigenvalues. Thus, the lower rank approximation matrix containing only the eigen-images belonging to the blood space can be represented by:

$$\widetilde{\boldsymbol{A}} = \sum_{i=c+1}^{c+d} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\dagger} \tag{1}$$

Where c and d represent the rank of the clutter and blood signal. This can be thought of as decomposing our spatiotemporal matrix into a weighted, ordered sum of separable matrices A_i (eigenimages). In other words, the SVD filter can be used to find the decomposition of an ultrasonic dataset into spatial and temporal filters, which in theory should be superior to FIR and IIR filters which only take into consideration temporal variations. Here u_i and v_i are the ith columns of the corresponding SVD matrices, σ_i are the ordered singular values. It is important to keep in mind that each column v_i corresponds to a temporal signal with length N_t and each column u_i corresponds to a spatial signal with length N_t.

To implement equation (1) in Simulink, we use the matrix form,

$$\widetilde{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{V}\boldsymbol{I}^{\boldsymbol{f}}\boldsymbol{V}^{\dagger} = \boldsymbol{U}\tilde{\boldsymbol{I}}^{\boldsymbol{f}}\boldsymbol{V}^{\dagger} \tag{2}$$

Where $\boldsymbol{I}^{\boldsymbol{f}}$ is a filter operator, in this case the identity matrix with zeros for the first diagonal elements and $\tilde{\boldsymbol{I}}^{\boldsymbol{f}}$ is the truncated diagonal matrix corresponding to the removal of tissue motion and noise.

As mentioned above, this representation of our data contains tissue movement in the first singular values (and vectors) due to their large spatiotemporal coherence. However, lower singular values

(and vectors) represent the blood signal since they are not highly spatiotemporally coherent. It is also widely known that the last eigenvalues (and eigenvectors) typically represent noise in the acquisition due to their very low spatiotemporal coherence [9].

Finite Impulse Response (FIR) Filter

As the name implies, FIR filters have a finite length. The output of the FIR filter with order (K-1) can be expressed as the finite convolution sum [8]:

$$y[n] = \sum_{k=0}^{k-1} h[n]x[n-k] = \sum_{k=n-k-1}^{n} h[n-k]x[k] \tag{3}$$

It can be shown that the frequency response of the FIR filter is given by $H_0[\omega] = |H[\omega]|^2$, where $H[\omega]$ is the Fourier Transform of the impulse response $h[n]$ [8]. Since the output does not depend on past inputs or outputs, the FIR filter is an example of a memory-less system.

There are four types of FIR filters with linear phase or constant group delay with impulse response of length M. These can be implemented on Simulink using the "Filter Design" block. A linear phase filter can be defined as a filter with frequency response of the form [8]:

$$H[\omega] = G[\omega] * e^{j(k_1+k_2\omega)} \tag{4}$$

where k_1 and k_2 are constants and $G[\omega]$ is a real function. Linear phase filters are advantageous in the sense that all the frequency components of the input signal are delayed by the same amount. Therefore, the input signal will not be distorted [8]. This is favourable since in the passband, the frequency response can be approximated by $H[\omega] \sim e^{j(k_1+k_2\omega)}$. However, the linear phase filter has a constraint, the impulse response must be symmetric. This is not true for IIR filters.

In some applications where only the amplitude response is considered, it is possible to use the minimum phase FIR filter, which has all the zeros inside the unit circle. For a clutter filter, FIR filters with minimum phase are proven to be more efficient [8]. However, this filter is not being implemented on this project since we are using the "FDA toolbox" in Simulink that designs linear phase FIR filters.

Infinite Impulse Response (IIR) Filter

IIR filters of order K are defined by the difference equation:

$$y[n] = \sum_{k=1}^{K} a_k y[n-k] + \sum_{k=0}^{K} b_k x[n-k] \tag{5}$$

where the output depends on present and past input samples and on past output samples making it not a memory-less system.

In Simulink, when using the "Filter Design" block, the frequency response of IIR filters are considered as its steady state magnitude response. Butterworth, Chebyshev type I and II, and Elliptic are the most common IIR filters, and are available on the "Filter Design" toolbox [8]. It is

also known that the most efficient IIR filter is the Elliptic. However, if the input signal is finite or contains discontinuities, the transient response may be important (there is always a transient response before the steady state). Moreover, the Butterworth filter is the one that has the smallest transient response at the cost of longer transition region [8]. For this project, only Elliptic IIR filters will be considered.

METHODS

Data Acquisition

Ultrasound data of a human carotid was acquired and processed using a programmable ultrasound system (Vantage 256, Verasonics, Redmond, WA) and a 5 MHz 128-element imaging transducer array (L7-4, Philips ATL, WA). RF channel data was acquired and RF beamformed images were reconstructed offline using well-known dynamic receive beamforming algorithms. All processing of the data was performed on MATLAB (The MathWorks Inc., Natick, MA). Cross-sectional and lateral images of the carotid were acquired at 500 frames-per-second for a period of 2.5 seconds, giving 1250 image-frames for each view. Each image frame is of data-type double and has a size of $N_z x N_x$, corresponding to the two spatial dimensions of the image, the axial dimension ($\boldsymbol{z}$) and the lateral dimension ($\boldsymbol{x}$). In this experiment, $N_z = 128$ and $N_x = 64$. The reconstructed images were then used in the Simulink models described below.

Simulink Implementation

Singular Value Decomposition

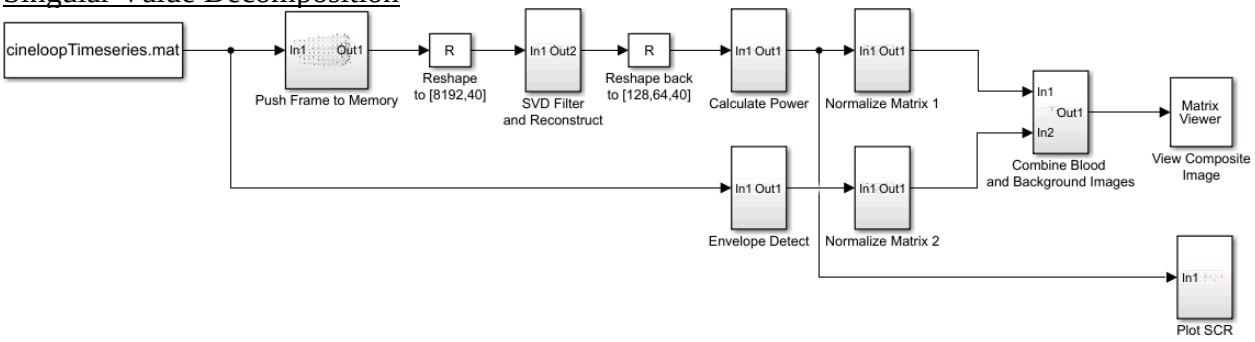


Fig. 1. High-level Simulink model of the SVD filter implementation

To implement the SVD filter in Simulink, we first need to collect several frames and stack them in a matrix to process with the SVD filter as outlined above. To achieve this, we first model our input data source as a MATLAB **timeseries** object that has a sample rate of 500 Hz. The **timeseries** object allows us to specify each 2D image frame as a single sample and the corresponding sample time associated with that frame. The **timeseries** object for the 1250 frames is created using a sample-rate of 500 Hz and stored in a file that is used as a data-input source in the Simulink model as shown in Fig. 1.

From a high-level perspective, we first wait for a frame that arrives every 2 ms. The image frame is then pushed onto a memory stack that holds the previous 40 frames stacked in the third dimension. The output of the memory stack gives us a matrix of size $[\boldsymbol{128, 64, 40}]$ which is then reshaped into the $[\boldsymbol{128x64, 40}]$ or $[\boldsymbol{8192, 40}]$ spatiotemporal form needed for the SVD processing. The SVD filter is then applied on this spatiotemporal matrix to remove the tissue clutter and then the matrix is reshaped back to a 3D $[\boldsymbol{128, 64, 40}]$ matrix to calculate the power image of the blood. The generated blood image and the enveloped ultrasound image are then overlayed to

produce one output that is shown on the screen through a "Matrix Viewer" Simulink block. Each block is described in detail below.

Memory Block

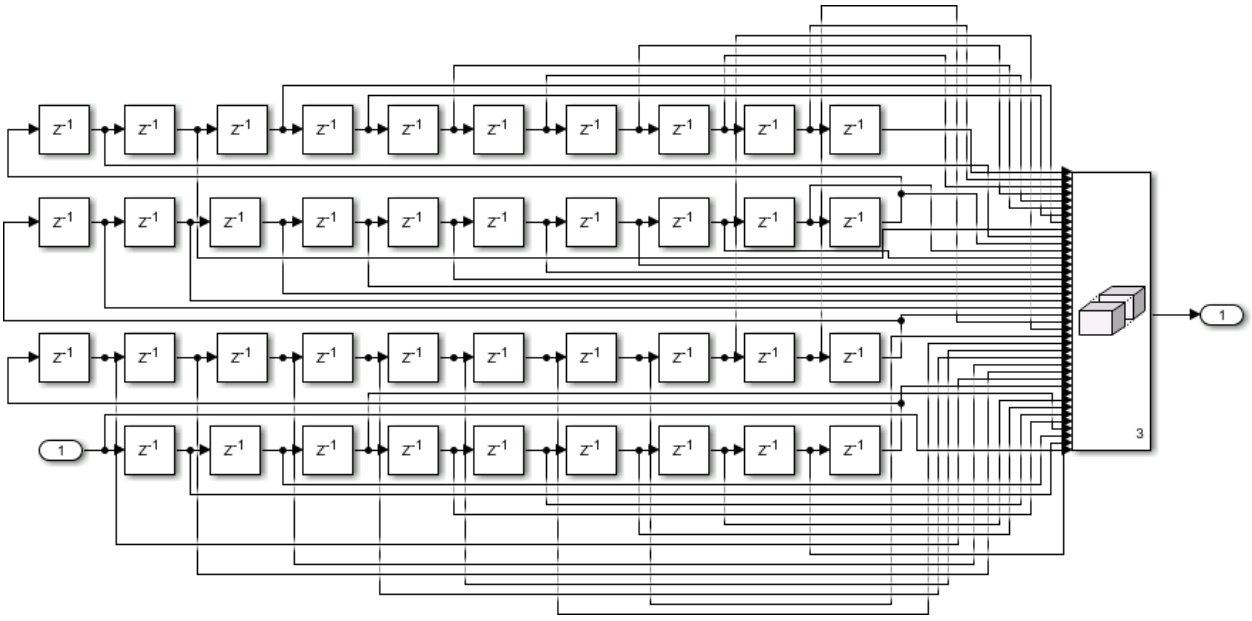


Fig. 2. Simulink model of a 40-element memory block

The memory block consists of a simple implementation of a 40-element memory stack. The memory stack is implemented using 40 unit-delay blocks, with each delay block's output connected to the input of the next delay block. The outputs of the delays blocks are then connected in reverse-order to a Matrix Concatenator, which stacks their outputs along the third dimension resulting in a 3D matrix whose size is $[\mathbf{128}, \mathbf{64}, \mathbf{40}]$.

When the simulation starts, each delay block is initialized to 0. This means that at the first time-step (2 ms) when the first frame is pushed onto the memory stack, the stack will be all empty except for the last frame. By the 40^{th} time-step (80 ms), the memory stack will have stored all the previous 40 frames, with the oldest frame now stored in the first index in the third dimension, and the newest frame stored in the last index. For each time-step after the 40^{th}, the oldest frame will be pushed out of the stack, and the newest frame will be pushed into it. Hence, at the 40^{th} time-step, the memory stack will hold frames 1-40 of our timeseries; at the 41^{st} time-step, the memory stack will hold frames 2-41; at the 42^{nd} time-step, the memory stack will hold frames 3-42; and so on.

Note that since we have to wait for 40 frames to fill up the buffer, the Simulink model will have a transient response that we are not interested in during the first 39 time-steps. This transient response is truncated in the results shown below.

SVD Filter and Reconstruct

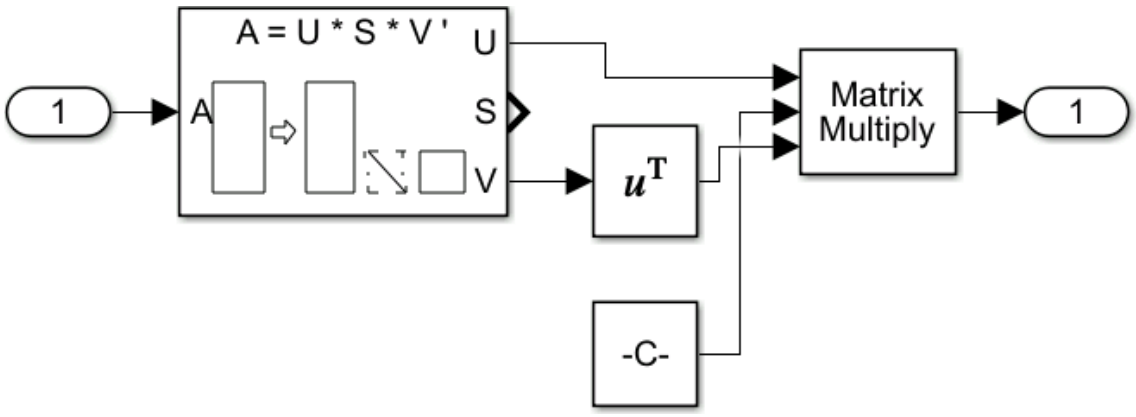


Fig. 3. Simulink model of the SVD Filter and Reconstruct block

Once the 40-frame data is reshaped to the spatiotemporal form, it is then passed through a "Singular Value Decomposition" block in Simulink. This block decomposes the matrix to its basis' **U** and **V** matrices which are then used to reconstruct the matrix with the required eigenvalues according to Equation (2). The reconstruction operation is a matrix multiplication of $\boldsymbol{U\tilde{I}^f V^{\dagger}}$, where $\boldsymbol{\tilde{I}^f}$ is a truncated identity matrix that defines the eigenvalues we wish to keep. As previously mentioned, the blood signal contributes to more spatiotemporal variation compared to the tissue signal. This implies that the first eigenvalues are related to the clutter subspace while the last eigenvalues are related to the blood subspace. Hence, the constant **C** in the model above was chosen as an identity matrix whose diagonal elements are all zero except for 10 last diagonal elements. This ensures that only the 10 fastest varying spatiotemporal subspaces of our matrix are reconstructed, which should be the blood flow.

Calculate Power

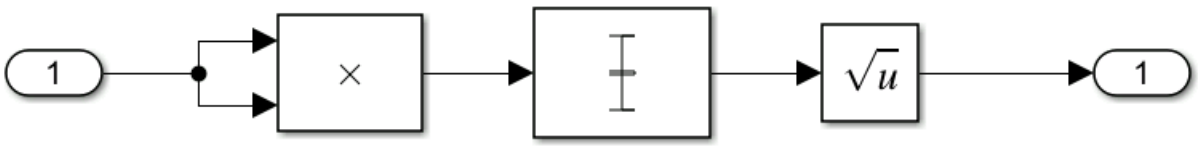


Fig. 4. Simulink model of the power calculation block

Once the 40-frame data has been clutter-filtered with the SVD block and reshaped back into a 3D $N_z x N_x x N_t$ matrix, we then calculate the power of the matrix to generate a blood image. Given a 3D matrix represented by **S**, the power is defined as $\sqrt{mean(S.\hat{}2)}$. This operation averages the power of the pixels across the third dimension and results in a $N_z x N_x$ or $128x64$ matrix in our case. This resulting matrix is referred to as the blood image.

Normalize Matrix

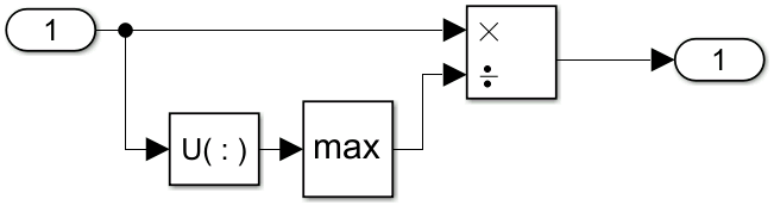


Fig. 5. Simulink model of a 2D matrix normalizer

Before overlaying the blood and ultrasound images, the 2D images are first normalized to 1 to simplify processing. This is implemented in Simulink as a simple operation that divides the input matrix by its maximum value.

Envelope Detect

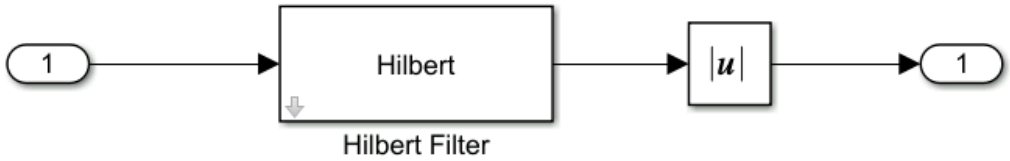


Fig. 6. Simulink model of a 2D matrix envelope detector

Before displaying the ultrasound image, an envelope-detection technique is applied on the image to convert from sinusoidal data to intensity data. The envelope-detection is implemented as a 31-order IIR Hilbert filter with a normalized transient width of 0.1 using the "Hilbert" block in Simulink. The absolute value of the Hilbert-filtered result is then obtained to produce the envelope-detected image.

Combine Blood and Background Images

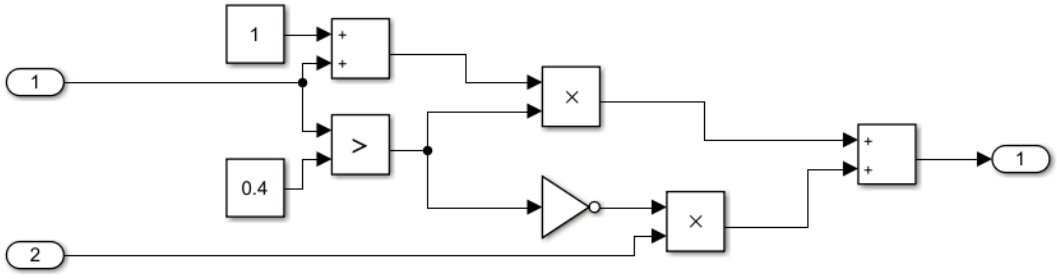


Fig. 7. Simulink model of the "Combine Blood and Background Images" block

The normalized blood and background images are then combined according to the diagram above. The blood image is first thresholded by selecting only the regions of the blood image that are above a selected power threshold, 0.4 in this example. This ensures that only strong blood signals (and not noise) are overlayed in the resulting image. The thresholded mask is also used to select the part of the background image that does not include the blood pixels that were chosen for the blood image by passing the mask through a NOT operator. To show the background and

the blood in different colors, we add 1 to the result of the blood image so that its values now lie between 1 to 2 rather than 0 to 1. The two masked matrices are then summed to generate the combined image. In the Matrix Viewer block, the colormap is set to **[gray; hot]** so that values of 0 to 1 (the background tissue) are displayed using a gray colormap, and the values between 1 and 2 (the blood) are displayed using a hot colormap.

Plot SCR

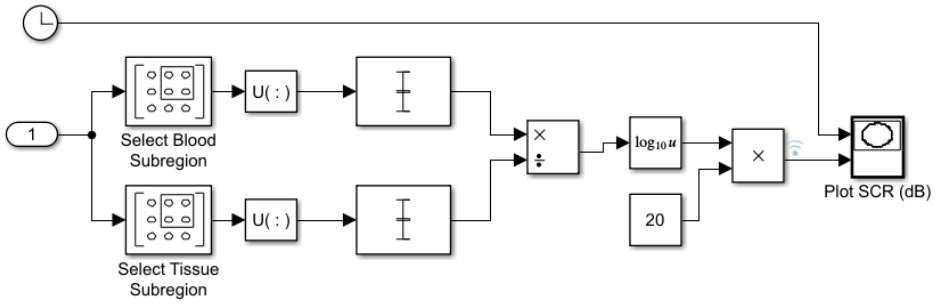


Fig. 8. Simulink model of the signal-to-clutter ratio calculation

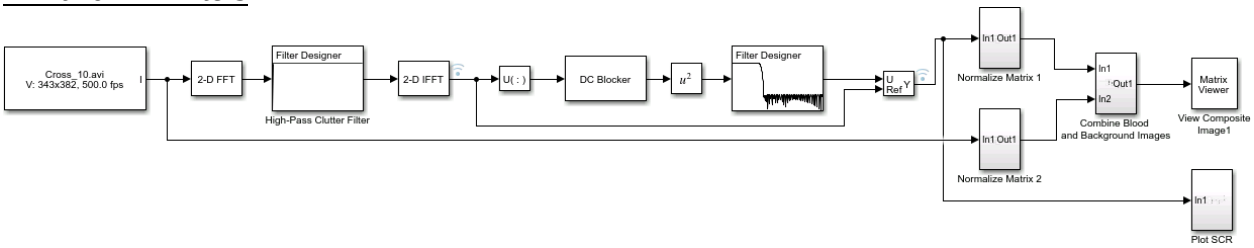
For each power image that is generated, the signal-to-clutter ratio (SCR) is calculated to evaluate the performance according to the equation,

$$SCR = 20\log_{10}\frac{P_f}{P_c} \tag{6}$$

where P_f is the power of the blood signal and P_c is the power of the clutter signal or background tissue.

The blood and tissue subregions are first selected from the power image using the Submatrix block in Simulink. The submatrix indices were found by viewing the ultrasound image and recording the relevant indices of the tissue region and the expected blood region. The indices were then hardcoded into the Submatrix blocks in Fig. 8 above. The averages of those subregions are then calculated and the SCR ratio is finally found in decibels according to Equation (6). The calculated SCR value is then output to a XY Plot with the current clock time for a real-time plot of the SCR as the simulation progresses.

FIR and IIR Filters



The implementation of both FIR and IIR filters were identical in Simulink except for the type of filter selected in the “Filter Design” block and the cut-off frequencies. It is important to note that the main challenge throughout the implementation of the filters was to realize how to treat the data in Simulink. First, it was thought to consider each column of our matrix signal (that represents each frame) as a single channel through the entire process. However, calculating the power this

way would be complicated. Therefore, we decided to consider each pixel as an independent channel to calculate the average power of the signal. This was achieved by reshaping each matrix or frame into a vector.

Before doing any processing, the input signal is viewed through the "Spectrum Analyzer" block to look at its Frequency-Power spectrum. At this point, each frame is treated as a single row vector. This is extremely important because our objective is to identify and mute or attenuate unwanted temporal frequencies. In this case, we are performing a high-pass filter due to the higher frequency of the blood signal.

The "2-D FFT" block applies a 2D Fourier Transform to a frame in the input signal. Afterwards, using the "Filter Design" block, a high-pass filter (FIR or IIR) is implemented to attenuate the clutter signal. The specific stop and pass frequencies for each filter are explained in the results section of this report. The filtered signal is transformed back to the time domain using the "2-D IFFT" block. It is important to mention that until this point every column of the matrix input signal is still being treated as a channel to make it compatible with the "Filter Design" block.

Furthermore, to calculate the average power, the filtered matrix signal is reshaped into a vector. The DC offset is subtracted using the method "subtract the mean", each pixel is then squared using the "Math function" block and finally averaged using a low-pass filter. The processed signals are reshaped back to the original matrix size, combined with the background ultrasound image, and then displayed in the "Matrix Viewer" window.

RESULTS and DISCUSSION

FIR and IIR Filter Parameters

Information about our clutter signal was learned by looking at the frequency spectrum of the signal through the "Spectrum Analyzer" tool of Simulink. Fig. 9 shows the frequency spectrum of our signal before high-pass filtering (left), and after high-pass filtering (right). We can see how based on our filter parameters, the frequency spectrum of the high-pass filtered signal on the right effectively attenuates the lower frequencies which correspond to the tissue signals.

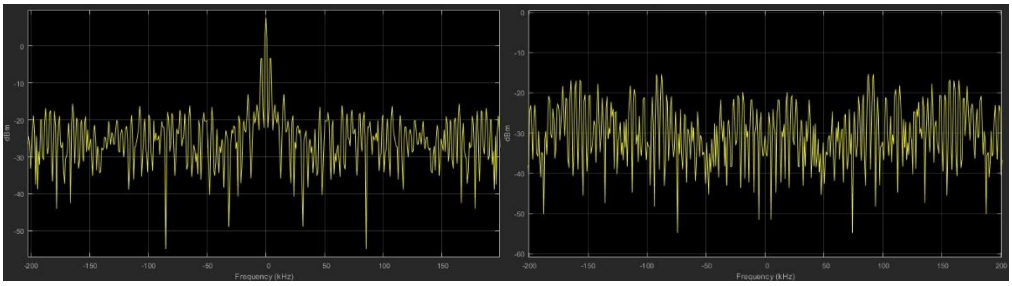


Fig. 9. (Left) Spectrum of the input signal where it is possible to see the clutter signal at the lower frequencies. (Right) Spectrum of the filtered signal where it is possible to see that the clutter signal has been attenuated.

It is known from the digital filtering theory that the best FIR filter is the one designed using the Equiripple method. Therefore, this is the only method used for the FIR filters. Using the "Filter Design" block and the information from the spectrum of the input signal, a high-pass filter was designed with a normalized stopping frequency of 0.004 (π rad/sample) and a normalized passing

frequency of 0.0079. The filter with the minimum order has a constant phase and group delay of 858 samples.

Like the FIR filter, the elliptic IIR filter is Equiripple in both stop and passband. It is also known that no other filter of the same order can have steeper transition between the pass and stop band. As the ripple in the pass and stop band approaches zero, the filter becomes a Butterworth filter. The filtered signal is almost identical to the one obtained from using a FIR filter. Using the "Filter Design" block and the information from the spectrum of the input signal, a high-pass Elliptic filter was designed with a normalized stopping frequency of 0.003 (π rad/sample) and a normalized passing frequency of 0.005.

The lowpass filter with minimum order used in the process of calculating the average power for the FIR/IIR filters has a passband normalized frequency of 0.3, and a stopband normalized frequency of 0.35 π rad/sample. The FIR filter presented a behavior closer to the ideal lowpass filter for this application, with constant group and phase delay of 50 samples.

Blood Power Images Comparison

From the Simulink simulation, we obtain more than 1200 frames of blood flow from each filter used. Here we only show a comparison of a selected single frame between the SVD and the FIR/IIR filters. Videos obtained of all the frames for each filter model are available upon request. Fig. 10. below shows a comparison of the obtained blood flow power images for (a) SVD filter applied in the cross-sectional view of the carotid, (b) FIR filter applied in the cross-sectional view, (c) IIR filter applied in the cross-sectional view, (d) SVD filter applied in the axial view, (e) FIR filter applied in the axial view, and (f) IIR filter applied in the axial view.

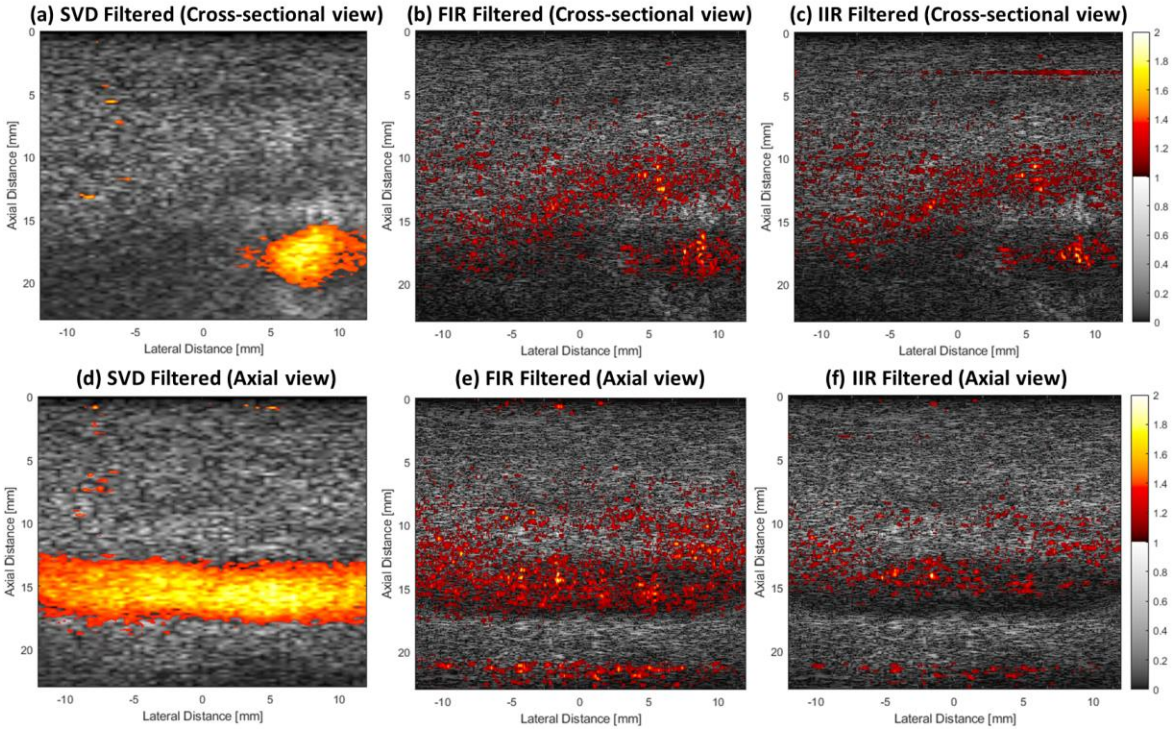


Fig. 10. Comparison of the blood flow images obtained using the three different filters.

First, we observe that the SVD filter outperformed the FIR and IIR filters in all cases. For the SVD filtered results, the blood power is much more intense and more uniform across the region of the carotid. Compared to the SVD filter, the FIR and IIR filters have a much weaker blood signal that is sparse and discontinuous rather than uniform. This shows that the SVD method is more sensitive to blood flow compared to the other filters.

We also observe that the FIR and IIR filters have much more noise (red speckles) outside of the carotid, while the SVD filter has almost no noise outside the carotid region. Since the SVD filter results in a higher intensity blood signal, we can choose a high power threshold to discard low-intensity blood signals (or noise) when we overlay the blood and background tissue images. However, in the case of the IIR and FIR filters, the blood signal has a much lower intensity, closer to the noise level of the system, and therefore we have to choose a low threshold to ensure that the blood signal is separated from the tissue signal. However, the low threshold means that some noise will likely be picked up as well.

Even though the FIR and IIR filters performed much worse than the SVD filter, we observe that with the filter parameters chosen for the FIR and IIR filters, we are indeed able to separate the blood and tissue signals as shown in Fig. 10 (b-c) and (e-f), showing that the FIR and IIR filter approach does work for clutter suppression. Compared to each other, the FIR filter outperformed the IIR filter in terms of signal-to-noise and uniformity of the blood flow in both views of the carotid.

Performance Metrics

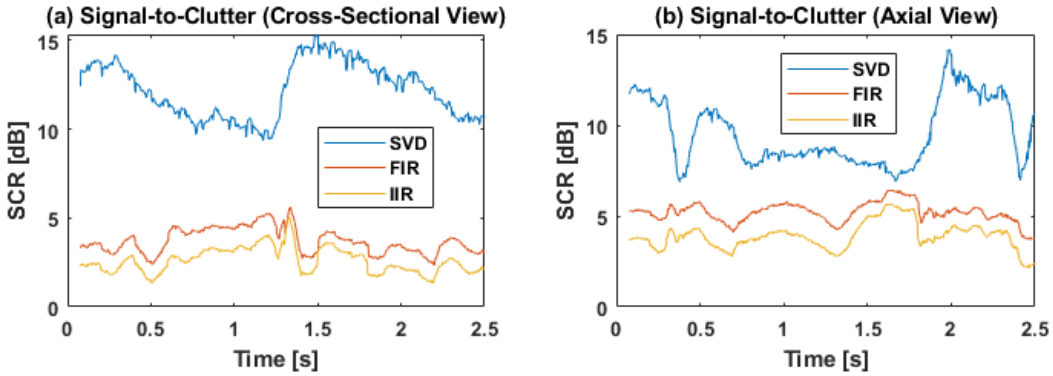


Fig. 11. Comparison of the SCR metric for the thee different filters in the (a) cross-sectional view, and (b) axial view.

To evaluate the performance of the system with the three different filters, we calculated the signal-to-clutter ratio for each blood image generated as shown above. This resulted in 1250 samples of SCR values representing the evolution of the SCR parameter as the simulation progresses for each filter implementation.

Fig. 11 above shows a comparison of the SCR metric for the three different filters used in (a) the cross-sectional view of the carotid, and (b) the axial view of the carotid. Note the first 40 samples were discarded due to the transient nature of the SVD filter implementation in Simulink.

As seen visually in the previous results, the SVD filter outperforms all filters in terms of signal-to-clutter. The SCR of the SVD approach ranges between 10-15 dB throughout the simulation in the cross-sectional view and 7-14 dB in the axial view while the FIR and IIR filters are at the 3-5 dB level in both views. Hence, the SVD filter has a 10 dB maximal increase in SCR compared to the FIR and IIR filters.

Compared to each other, the FIR and IIR filters perform very similarly. As previously seen with the visual result and confirmed with the above SCR metrics, the FIR filter performs better than the IIR filter by almost 1 dB.

Application: Calculating heart rate from blood power images

The blood flow images and the SCR values shown above in Fig. 11 tell much more of a story than just the intensity of the blood flow. From the trends of the SCR data, we can deduce several hemodynamic parameters such as heart rate from the cyclic nature of the intensity levels, or rate of ejection of blood flow from the slopes of the trends.

We expect in a breathing, living subject that the blood flow will rise in intensity and then decrease in intensity, representing the diastole and systole phases of the cardiac cycle. If we obtain a good estimate of the blood flow and a clean SCR metric over time, we can roughly approximate the heart rate from the subsequent maximas in the SCR metric.

For example, let us consider the SCR trend of the SVD filter in the cross-sectional view (Fig. 11a) since that is the cleanest one we obtained with the most visible trends. We observe there are two maximas in the graph, representing two cardiac cycles. The maximas occur at a time of 0.4 s and 1.2 s, or a time difference of 0.8 seconds. This translates to a heart rate of 60/0.8 = 75 beats-per-minute. Typical adult heart rates are around 60-100 bpm, so an estimate of 75 bpm falls well within the range of physiological heart rates.

CONCLUSION

We have successfully shown that the SVD filter can be implemented in Simulink, and that such a filter is superior to temporal IIR and FIR filters for detecting blood flows in-vivo. Although present ultrasound literature does not explore the implementation of clutter filters using Simulink, this study shwows that Simulink is an appropriate tool for implementing clutter-suppression techniques for blood flow imaging using ultrasound data in real time. We found that the SVD filter performed best when compared against the other two types of temporal filters as expected because of its spatiotemporal characteristic. The system could be further extended and explored implementing the Higher Order SVD (HOSVD) filter which gives even more information, and investigating the effects of different frame rates in the input ultrasound data.

REFERENCES

[1] D. Asiedu, K. Lee. “A Non-Invasive Detection and Monitoring of Intercranial Pressure Using Ultrasound Sensors.” *Journal of Neurology Research,* vol. 5, pp. 167-180, Apr 2015.

[2] M. Kim, et al. "Expanding Acquisition and Clutter Filter Dimensions for Improved Perfusion Sensitivity." *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 64, No. 10, Oct 2017

[3] R. Fox. "Optimization of Digital Clutter Filter for Microvascular Volume Quantification using Power Doppler Ultrasound." *MBP 3970Z Six Week Project*, Department of Biophysics Western University, 2012.

[4] C. Demené, et al. “Spatiotemporal Clutter Filtering of Ultrafast Ultrasound Data Highly Increases Doppler and fUltrasound Sensitivity.” *IEEE Transactions on Medical Imaging*, vol. 34, No. 11, Nov 2015

[5] S. Bjaerum, et al. “Clutter Filter Design for Ultrasound Color Flow Imaging.” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 49, No. 2, Feb 2002

[6] C. Eckart and G. Young. “The approximation of one matrix by another of lower rank.” *Psychometrika*, vol. 1 No.3, pp 211-218

[7] Park, G., et al., “New Adaptive Clutter Rejection Based on Spectral Analysis for Ultrasound Color Doppler Imaging: Phantom and In vivo Abdominal Study”, IEEE Transactions on Biomedical Engineering (Volume 61, NO. 1, Jan. 2014).