1. Introduction

Cerebral blood-flow is defined as the measure of the rate of delivery of arterial blood to a capillary bed in a tissue [1]. Algorithms have been proposed to estimate this velocity value using the Doppler Effect. By processing the phase shift between reflected ultrasound signals emitted at the cerebrum, it is possible to gain crucial real-time insight into the health of a patient. Regulated blood-flow is essential to provide the required oxygen levels to tissues, handling changes in blood-pressure in the body. Analyzing deviations in measurements and having standard values for healthy patients could pave the way towards efficient methods of diagnosis by ultrasound scanning.

Measuring the **Doppler velocity** of blood-flow requires an intersection between medical imaging, programming, engineering and biophysics. Research in this novel field can be considered a response to the various healthcare challenges and limitations of other imaging modalities. With increasing pressure on hospitals, medical ultrasound succeeds in providing hypothetical solutions in accessibility, real-time imaging, safety and cost reductions.

For example, the portability of ultrasound devices provides patients in rural clinics with access to imaging [2]. Despite 9.7 million people living in rural areas, there are only 1000 clinics, resulting in one clinic for every 9700 people. Access to time efficient imaging is essential. Many patients today are wary of the ionizing radiation that CT scans use, where ultrasound could provide a non-invasive and non-ionizing alternative to ease such patients. Surgeons may also benefit - real-time image-guided surgery has become valuable tool in neurosurgery [3], in turn reducing wait times for patients in other clinical scenarios. In the current economy, it is becoming increasingly important to deliver new cost-efficient technology. To bring ultrasound from research labs into clinical settings requires in-depth collaboration among researchers, and detailed technical documentation of presented ideas.

'**The Complex Autocorrelation**' algorithm (Kasai et al. 1985) [4] and '**The Subsample Volume Doppler**' (SDopp) algorithm (Hoeks et al. 1994) [5] are two velocity functions implemented in the novel graphical user interface (GUI): BrainTV.

BrainTV is a software used to extract and visualize features, such as CO2 levels, blood pressure, and the aforementioned Doppler velocity of each patient. The connection between ultrasound technology, the signal processing required, and the implementation of the two algorithms in the MATLAB software has not been well documented prior. Additionally, there exist several concerns with obtaining reliable results via ultrasound, such as the effect of artefacts and noise at deeper depths; data is often discarded or unreliable and hence the clinical applications are not yet realized.

This paper is therefore a technical documentation of the two algorithms and their implementation within BrainTV. It aims to improve the understanding of the signal processing involved as well as discuss the challenges in obtaining reliable and meaningful measurements in the context of healthcare.

Chapter 2 will serve as background literature to introduce the reader to medical ultrasound and contextualize the use of the Autocorrelation and SDopp algorithms. Chapter 3 will begin to focus on the algorithms. Here, their implementations will be verified and

assessed in relation to noise and other artefacts. Then, Chapter 4 will be a technical documentation of parameters, data flow and code structure. Chapter 5 shall be a discussion and comparison of the two algorithms to determine which one should be maintained within BrainTV. Following on, Chapter 6 will look to address the challenges of artefacts and noise. Lastly, Chapter 7 will be a conclusion to summarise the goals achieved in this project and future research that must be done to develop BrainTV further.

2. Literature Review: From Ultrasound Scan to Velocity Estimation

2.1. Characteristics of Ultrasound

Ultrasound is a high frequency sound wave used to image the body and brain non-invasively. It can provide real-time insight into moving structures, such as tissues or blood flow in vessels. Frequency of emitted ultrasound waves is typically 2-10MHz, where higher frequencies (5-10MHz) are more commonly used in high resolution recordings of tissues and vessels. However, this range results in less penetration into soft tissues due to attenuation. Ultrasound relies on the emitted signal being reflected as an echo, back to the same probe (a transducer) [6].

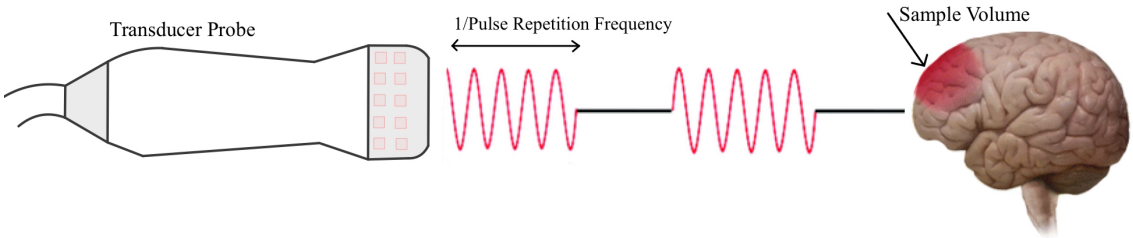


Figure 1: Probe Emits Pulse at a Sample Volume.

The level of reflection is dependent on the difference of acoustic impedance in two mediums. Acoustic impedance can be approximated as the product of the density with the velocity of the sound wave through the particular medium [7]. At large differences between two mediums (air and bone), almost all of the energy of the sound wave is reflected back. Therefore, it is most common to use ultrasound for imaging soft tissues such as the brain.

Although this is one example where attenuation is useful in obtaining meaningful measurements, there are other forms which can produce artefacts. Attenuation produces energy losses through reflection, refraction, scattering and absorption. Absorption occurs due to the friction between oscillating particles and is the largest cause of energy loss, in the form of heat. In the context of brain tissue imaging, high-frequency ultrasound results in high absorption, and moreover unreliable measurements at deeper depths. It is most prevalent to observe noise and artefacts at deeper gates.

2.2. Scan Procedure

Nevertheless, measurements can be effectively made at smaller depths. The user may select the area of interest, known as a sample volume or '**gate**'. In research, the sonographer typically sets the probe against a patient's temple to access the temporal lobe. The probe is activated by briefly passing an electric current, to which its many piezoelectric elements emit a pulse towards a tissue, in the form of a point on a scan line

[8]. The pulse is reflected as it travels to the focal point, back to the probe. Once all the waves return, the current is passed again to transmit new sound waves towards a new point along the scan line. This is an iterative process as shown in Figure 2. Once all the points along the scan line are measured, a new scan line is selected and the process continues.

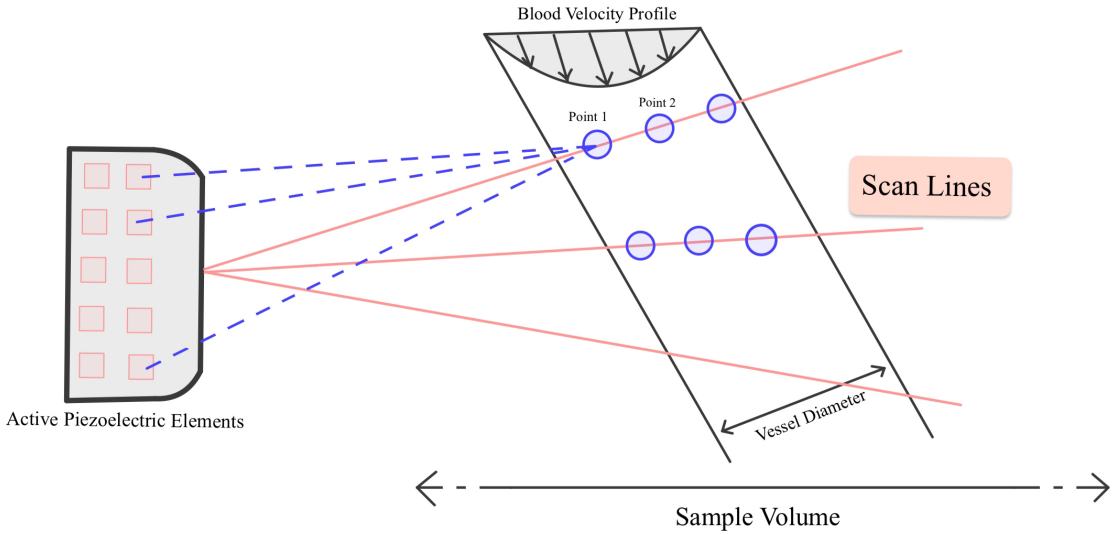


Figure 2: Sample Volume is Imaged via Points on Scan Lines

The geometry of the transducer in relation to the direction of blood-flow is incredibly important in gathering high-fidelity data. Ultrasound beams aligned at 90° relative to direction of blood-flow will produce the lowest quality results, while beams aligned parallel will produce high frequency images. This phenomenon can be explained by the Doppler frequency shift:

$$f_d = \frac{2 \cdot f_0 \cdot v \cdot cos(\theta)}{c} \tag{1}$$

- f_d - Doppler shift (Hz)
- f_0 - Pulse frequency (kHz)
- v - Blood velocity (μm/sec)
- θ - Incident angle between velocity and beam vectors

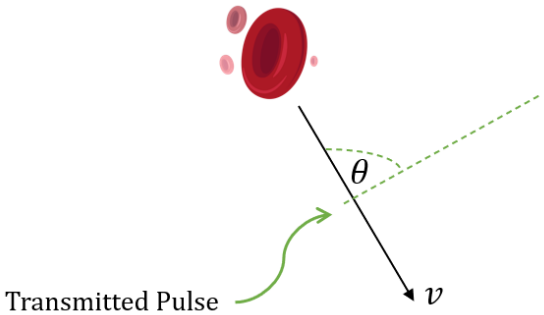


Figure 3: Geometry of transducer emitted pulse and blood velocity vectors

- c - Speed of sound in tissue (1.54 mm/μsec)

Using a pulse-wave (PW) system, a sample volume may be interrogated by measuring the relative time difference of the reflected signals arriving at the transducer. This is analogous to computing the phase shift. Consequently, it is possible to then estimate the Doppler velocity, which is proportional to the change in phase shift of the signal at that particular gate/depth:

$$v(m,n) = \frac{\lambda_n}{2}\left(\frac{\phi(m,n) - \phi(m,n-1)}{2\pi T}\right) \tag{2}$$

- $v(m,n)$ - Velocity from m^{th} depth and n^{th} pulse.
- $\phi(m,n)$ - Phase of signal returned from m^{th} depth and n^{th} pulse.
- λ_n - Nyquist sampling rate (m)
- T - reciprocal of the Pulse Repetition Frequency (PRF), or simply called Time Interval (s)

2.3. Harmonics & Aliasing

The required delay time between emitted and reflected signals is an important quality of such PW systems. PRF refers to the number of pulses sent over a period of one second, which requires some consideration of the motion velocity of the tissue of interest. The ratio between emission and waiting time is around 0.1% to 99.9%, which implies a maximum limit to the PRF at which, beyond, the echoes from the gate no longer have enough time to return before the next pulses are emitted [9]. Therefore PW systems are strongly limited by their sampling frequency.

It is important to understand that an ultrasound pulse can be defined by its center frequency f_0, but it is composed of a range of frequencies called the bandwidth [10].

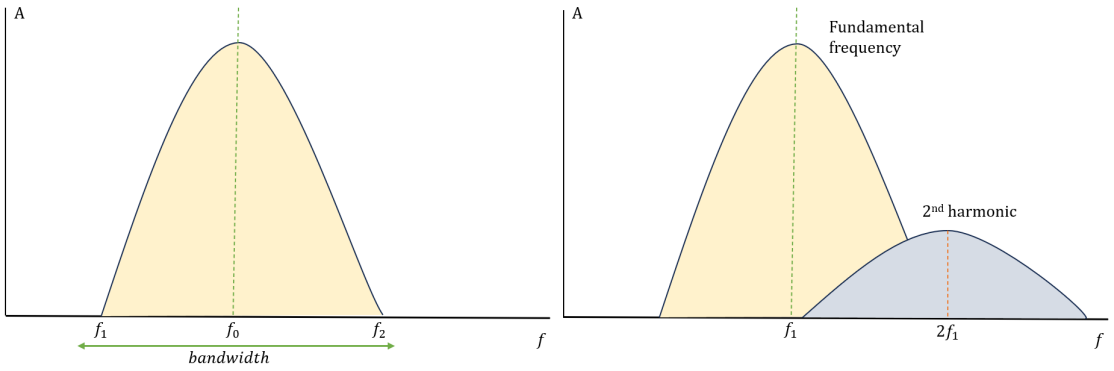


Figure 4: Left shows fundamental frequency; Right shows second harmonic (e.g. echo pulse) overlapping with the fundamental frequency.

As a generated pulse passes through tissue, the speed of sound increases slightly causing the wave to travel at a higher rate. This occurs due to the natural characteristic of sound waves creating areas of high and low compression. In low compression areas of the

brain tissue, the wave travels slower. This contrast in wave velocity creates harmonic frequencies, which return as echoes to the transducer.

These harmonic frequencies are higher than the central frequency of the transducer. This greatly improves Signal-to-Noise Ratio (SNR) as the beam becomes focused - higher frequency harmonics increase lateral resolution and reduce backscattered signals from artefacts around the sample volume. Ultrasound probes transmit very short pulses which in turn can create a large bandwidth - causing overlap of central and harmonic frequencies. This overlap causes the aliasing effect.

The Nyquist Frequency dictates the highest fidelity the transducer may reliably image, which is half of the sampling frequency PRF [11]. In the case of arterial blood velocity and the angle θ yielding a higher Doppler frequency value than $\frac{PRF}{2}$, aliasing will occur, creating ambiguity in the displayed signals. Aliasing creates a difficult and dynamic problem where parameters must be optimized to the various biological processes in the body. Examples include changes in arterial pressure due to cardiac pulsations, brain metabolism where velocity may differ in the right and left hemisphere [12], as well as increased tissue motion amplitude in deep structures of the brain compared to the cortex [13].

Conclusively, the maximum velocity that can be reliably measured is:

$$v_{max} = \frac{c}{4} \cdot \frac{PRF}{f_0} \tag{3}$$

And the Nyquist Rate is:

$$\lambda_n = \frac{c}{4 \cdot 1000} \cdot \frac{1}{f_0} \tag{4}$$

In addition to correct parameters, the sample volume must also be carefully selected during imaging to account for tissue movements due to cerebral pulsations. If the enclosed volume changes during the scanning process, the scattered signal may be further attenuated and create misleading results.

2.4. Processing Reflected Signals

Understanding the physics of PW Doppler systems is merely the initial step in obtaining phase and velocity values. This section shows the signal processing required to contextualize velocity estimators.

Once an RF (radio frequency; ultrasound) signal is at the receiver, it is first converted from a real to a complex signal. The resultant complex signal holds **phase** and **amplitude** information that is needed to compute the phase shift and recover the envelope, and thus also computing blood velocity. This process is done via. standard quadrature demodulation as shown in Figure 5.

There are two additional signals used for demodulation, the inphase D_I signal and the quadrature D_Q signal. Both share the same center frequency, w_c, as the RF signal; the quadrature signal is shifted by $\frac{\pi}{2}$ relative to the inphase signal. The following equations show the process of demodulation mathematically [14].

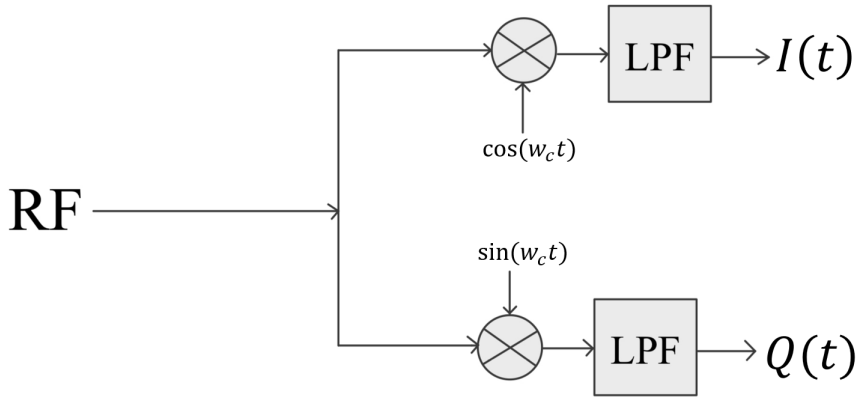


Figure 5: Standard RF Quadrature Demodulation procedure

It is common to represent an RF signal at the receiver as:

$$\begin{aligned} S(t) &= A(t)cos(w_c t + \phi(t)) \\ &= A(t)cos(w_c t)cos[\phi(t)] - A(t)sin(w_c t)sin[\phi(t)] \end{aligned} \tag{5}$$

where $A(t)$ is the amplitude and $\phi(t)$ is the phase at a given time. In order to extract information from the signal, it must be demodulated. For this we require an inphase D_I and a quadrature Q_I signal:

$$D_I(t) = cos(w_c t) \tag{6}$$

$$D_Q(t) = sin(w_c t) \tag{7}$$

Multiplying the RF signal by the inphase demodulating signal gives:

$$S(t)D_I(t) = \frac{1}{2}A(t)\{cos(2w_c t + \phi(t)) + cos[\phi(t)]\} \tag{8}$$

Multiplying the RF signal by the quadrature demodulating signal gives:

$$S(t)D_Q(t) = \frac{1}{2}A(t)\{sin(2w_c t + \phi(t)) - sin[\phi(t)]\} \tag{9}$$

By applying a low-pass filter (LPF) with a cutoff frequency at $\leq w_c$, all the terms with $2w_c$ are removed. As a result, this yields the respective inphase (real) and quadrature (imaginary) components of the complex signal:

$$I(t) = \frac{K}{2}A(t)cos[\phi(t)] \tag{10}$$

$$Q(t) = -\frac{K}{2}A(t)sin[\phi(t)] \tag{11}$$

where K is the gain due to LPF. Therefore, the complex signal can be shown as so, using Euler's identity:

$$\begin{aligned} S'(t) &= I(t) + jQ(t) \\ &= A(t)e^{-j\phi(t)} \end{aligned} \tag{12}$$

Obtaining the complex IQ signal from a reflected RF pulse is crucial in extracting features such as the phase and envelope. This is best shown by an IQ graph in figure 6. Consequently, the phase can be expressed as:

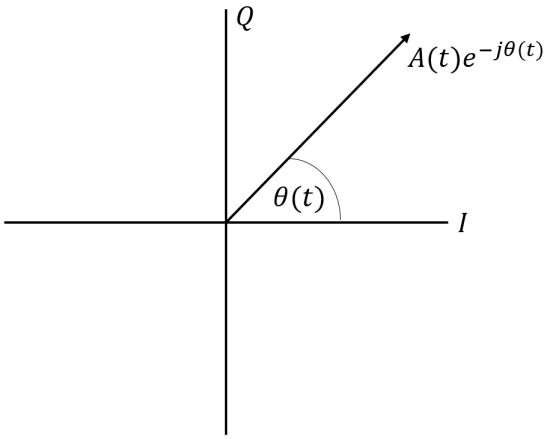


Figure 6: IQ vector has phase and amplitude components based on its I and Q parts.

$$-\phi(t) = \arctan\left(\frac{Q(t)}{I(t)}\right) \tag{13}$$

The envelope may also be extracted by taking the absolute value of the IQ vector:

$$E = \sqrt{I(t)^2 + Q(t)^2} = \frac{K}{2}A(t) \tag{14}$$

As a result of quadrature demodulation, the frequency spectrum of the RF signal is shifted towards the base-band. Then, the LPF erases the high frequency component to produce the base-band signal. This can be seen in Figure 7.

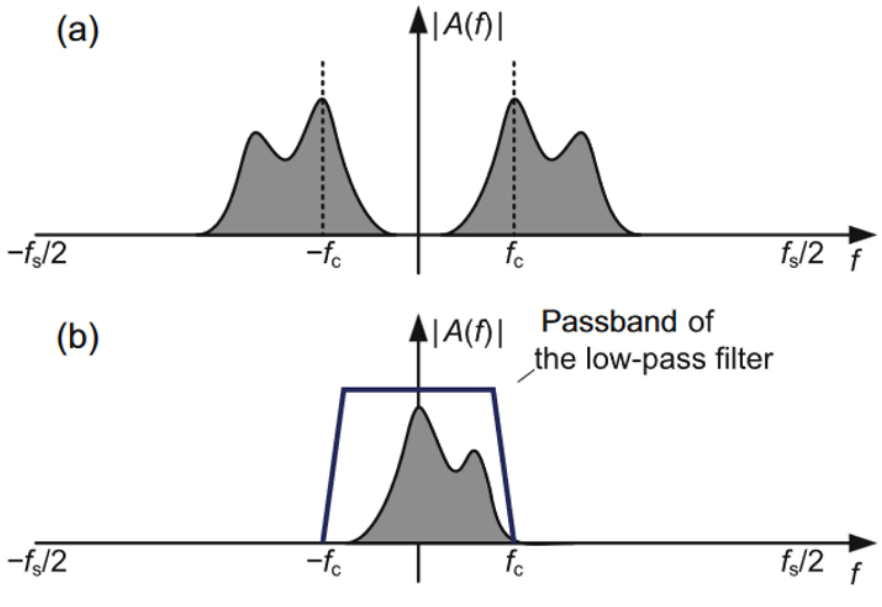


Figure 7: **(a)** Shows the frequency spectrum of an RF signal; **(b)** Shows spectrum after quadrature demodulation and low pass filtering. Adapted from [15]

Equation 12 expresses one RF signal at the receiver after demodulation. However, because the goal is calculating the phase shift which requires more than a single period, it is important to account for multiple RF signals as well as the digitization process.

Moreover, the complex IQ signal is better expressed as:

$$S'(m,n) = A(m,n)e^{-j\phi(m,n)} \tag{15}$$

where parameters m and n are introduced to index the signal by depth and pulse number respectively.

Once the phase can be extracted from the taken recordings, it is now possible to utilize the Doppler Effect shown in Equation 2 to compute the velocity of blood-flow within the targeted tissue. It is now possible to introduce our two algorithms.

2.5. Algorithm 1: The Complex Autocorrelation Estimator

The Complex Autocorrelation Estimator, developed by Kasai et al. (1985) [16], is a well known and commonly utilized phase-shift function for the velocity estimation of cerebral blood-flow. It computes the mean velocity of the reflected RF signals within a single sample volume over a series of pulse cycles:

$$v(m,n) = \frac{\lambda_n}{4\pi T} arg \left\{ \sum_{b=2-PL}^{0} S'(m,n-b)S'^{*}(m,n-b-1) \right\} \tag{16}$$

where PL is the package length, or the number of pulses the mean velocity is estimated over. $S'(m,n)$ is the digitized signal in Equation 15 and $S'^{*}(m,n)$ refers to the conjugate of this RF signal.

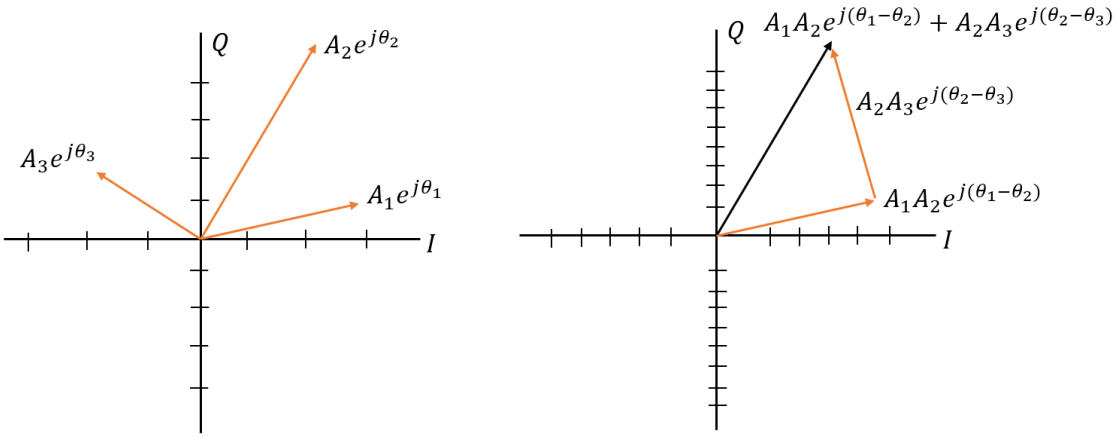


Figure 8: (a) Shows 3 IQ vectors after quadrature demodulation. (b) Shows the Autocorrelator process: the black vector is the resultant vector estimating the mean phase-shift between ultrasound echoes.

2.6. Algorithm 2: The Subsample Volume Doppler Estimator

A major assumption of most PW systems is that the phase of the received echo changes linearly with depth, though the spatial distribution of reflected pulses suggests this is not the case (Hoeks et al. 1994). The result of using PW systems with this assumption is the present variance in measured velocities. The solution to this is to split the observation window into smaller 'subsample' volumes and then find the mean velocity using the sum of these subsamples. This can be shown mathematically by the estimator known as The Subsample Volume Doppler (SDopp) algorithm:

$$v(m,n) = \frac{\lambda_n}{4\pi T} arg \left\{ \sum_{a=2-ND}^{0} \sum_{b=2-PL}^{0} S'(m-a,n-b)S'^{*}(m-a,n-b-1) \right\} \tag{17}$$

where ND is the number of subsamples over which the velocity is estimated.

This is the velocity estimator implemented in BrainTV in the file VSDopp_v2_BrainTV.m and is used to display arterial velocities from recorded transcranial Brain Tissue Pulsations (BTP) [13]. Consequently, in the case of no subsamples (ND = 1), SDopp reduces back to the Autocorrelation function. This method of estimation theoretically reduces the variance in recorded BTP measurements.

3. Verification

Implementing algorithms within a healthcare-related software is a great technical challenge. The already developed hardware and modules must be flexibly compatible while utilizing correct parameters and computing high-precision accurate data that may, in the future, be used as a basis to dictate the health of a patient. The functionality and usability of the BrainTV GUI relies heavily on the implementation of the SDopp estimator. This has not been verified prior to this project. Additionally, the mitigation of noise and artefacts is an equally crucial aspect in Doppler imaging. Biomedical imaging is often limited by the generalisations of typical signal processing studies, which assume most signals are affected by only Gaussian random noise - this is often not the case. It is essential to instill confidence in the sonographer that their recordings are clean of noise and therefore they are able to make accurate conclusions from signal analysis.

For small observation windows (e.g. one pulse), the velocity output computed by the SDopp algorithm should be exact to the velocity output computed by the Autocorrelation algorithm [5]. Therefore, a method to verify the implementation of the SDopp algorithm could be to program the Autocorrelation algorithm and compare results. This section looks to verify the SDopp algorithm by obtaining similar velocity values through the use of a second algorithm. It will also highlight Signal-to-Noise Ratio (SNR) challenges of Doppler transcranial measurements made at deeper gates/depths.

3.1. Verifying The SDopp Algorithm

The SDopp estimator can be verified by implementing a second algorithm to test for similar velocity values. The algorithm follows a two-step approach wherein it first averages over 1 period, and subsequently averaging over time and depth. This is done by setting the m depth parameter to 1 at the first iteration, and then averaging at further depths from 2 to 34 (the total number of gates).

To make this test fair, the temporal resolution was made exact by utilizing the same Package Length parameter. All header information read from raw Nihon Kohden files (containing additional parameters such as the Nyquist Rate and IQ Data) is managed by a separate file (RPData_BrainTV.m) and was not altered to further allow an accurate comparison. The Autocorrelator was newly implemented in the file VAuto_BrainTV.m.

It is visible from Figures 9, 10 that the implementation of the SDopp algorithm can be verified by the Kasai Autocorrelator for small observation windows, where velocity output is almost always exactly the same for gates below 20. At greater observation windows, the performance of the SDopp estimator begins to suffer and results may differ (Hoeks et al. 1994).

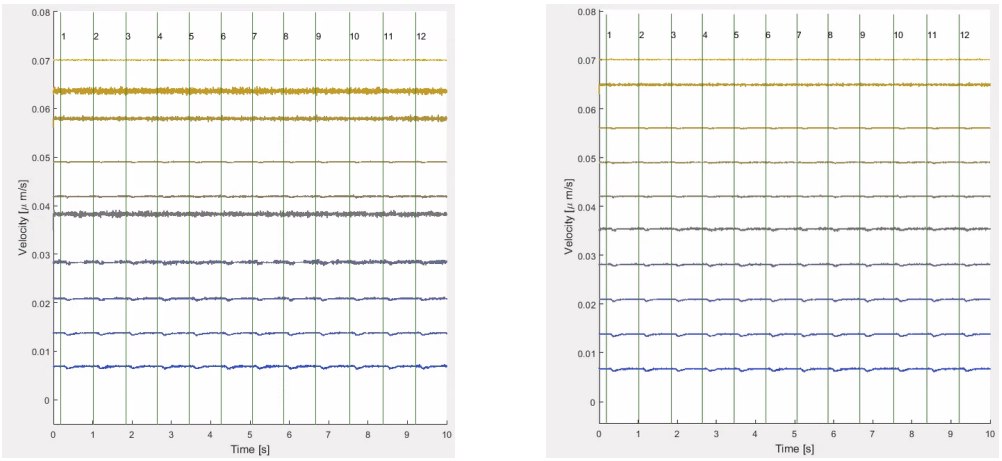


Figure 9: Velocity Output Gates 1-10; Left graph utilizes Autocorrelator; Right graph utilizes SDopp. Maximum deviation is +/- 0.003 $\mu m/s$. Autocorrelator appears to be slightly more noisy than SDopp estimated velocities.

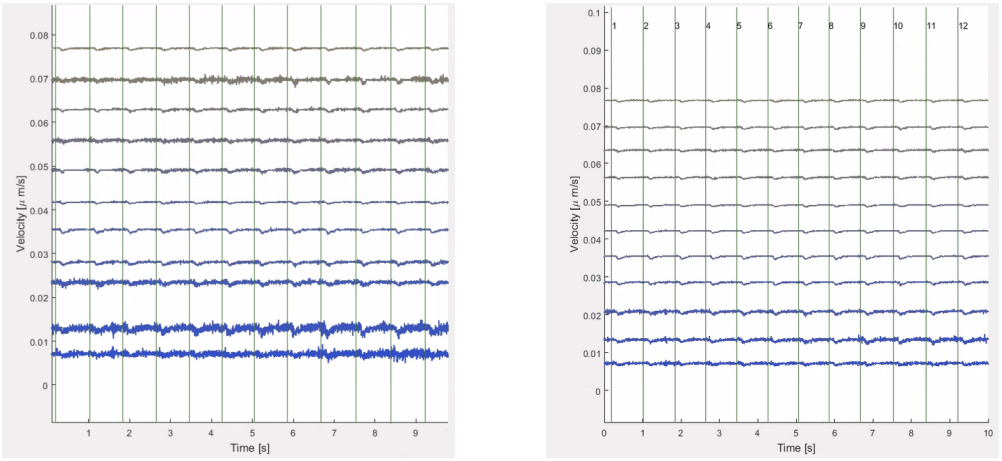


Figure 10: Velocity Output Gates 10-20; Left graph utilizes Autocorrelator; Right graph utilizes SDopp. Autocorrelator is once again outputs more noisy signals, however velocity values are the same.

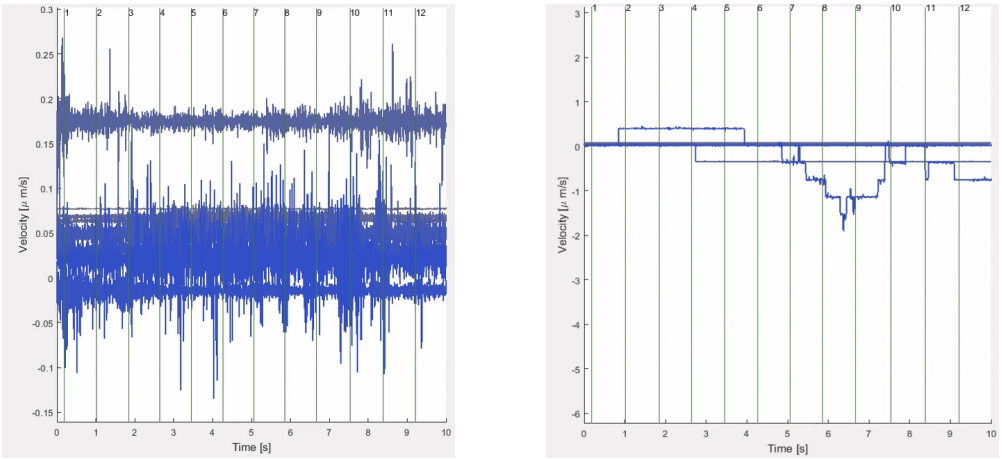


Figure 11: Velocity Output Gates 20-30; Left graph utilizes Autocorrelator; Right graph utilizes SDopp. Autocorrelator is very noisy, however still produces velocities greater than $0\mu m/s$ at deeper gates. SDopp has no meaningful velocity value, occasionally oscillating below $0\mu m/s$, implying a negative blood-flow.

The algorithm can be pseudo-coded as follows:

Algorithm 1 Kasai Autocorrelation Estimator

function VAUTO_BRAINTV(T, $avgIQ\_data$, $nlambda$, $GATE$)
  for $m \leftarrow 1$ **to** $GATE$ **do**
    for $n \leftarrow 1$ **to** length$(avgIQ_data) - 2$ **do**
      $autocorrelationSum \leftarrow 0$
      for $j \leftarrow 0$ **to** $PL - 1$ **do**
        $autocorrelation \leftarrow avgIQ\_data[n+2-j, m] \times \text{conj}(avgIQ\_data[n+1-j, m])$
        $autocorrelationSum \leftarrow autocorrelationSum + autocorrelation$
      end for
      $phaseDifference \leftarrow \text{angle}(autocorrelationSum)$
      $autocorrelationVelocity[n, m] \leftarrow \frac{nlambda}{2\pi \times T} \times phaseDifference$
    end for
  end for
  return $autocorrelationVelocity$
end function

3.2. Artefacts & Signal-to-Noise Ratio

Albeit, a second conclusion can also be made from the displayed data that both algorithms are not appropriate to compute velocities at deep depths in the brain. The Autocorrelator output shows heavily attenuated results at which the SNR worsens exponentially compared to measurements made near the cortex. The SDopp output appears to be heavily malfunctioned and aliased. In both cases, BTP data must be discarded to avoid ambigious conclusions. The effect of artefacts at deep gates is one of the most difficult challenges in ultrasound imaging.

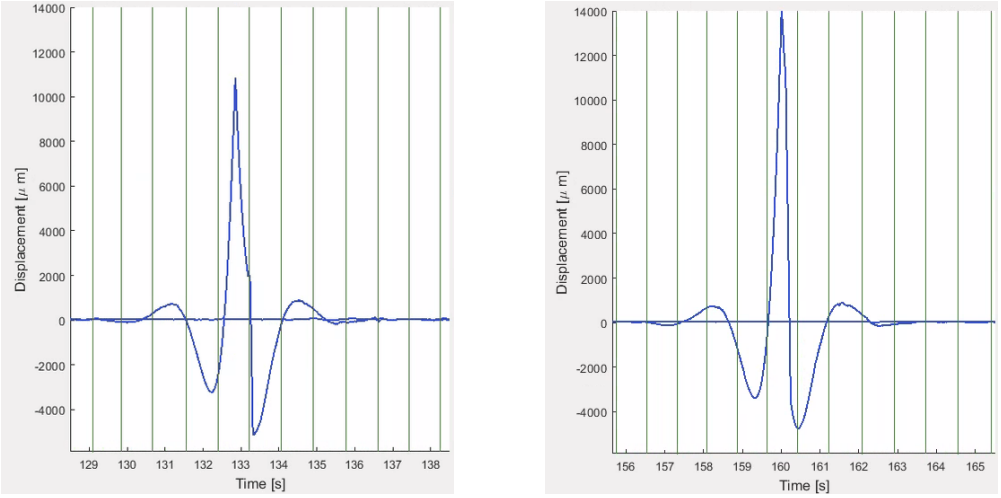


Figure 12: Gate 24; Figures show constructive interference in brain tissue displacement. The gradient increases at times 132.5 seconds and 160 seconds respectively, while remaining at zero motion at other recording times. Signals at deep gates must be discarded.

To image blood-flow at deep depths reliably, high frequency ultrasound is required. High frequency ultrasound however is limited by energy absorption at shallower tissues. A

second and third layer of complexity occurs due to aliasing and the effect of physical artefacts; superposition of signals may occur when PRF is not dynamically optimized to handle the required time-delay between emitted pulses. Certain scatterers may be amplified at certain depths when the Nyquist Rule and thereby signal interference is not considered. This can be observed in Figure 12.

In the BrainTV file VSDoppExt_BrainTV, an averaging/smoothing filter processes the Raw Velocity output from the SDopp estimator.

```
    for count = 1:gate
        Velocity{j,1}(:,count) = smooth(Velocity{j,1}(:,count));
    end
```

This uses the Local Regression Method [17], an averaging filter.

Averaging filters are sensitive to outliers in the BTP data, potentially caused by artefacts or noise and thus creating an ambiguous output particularly at deeper gates. SNR visibly worsens at deeper gates and its calculation is not rudimentary due to the correlation of signal and noise. The question arises: can an accurate blood velocity be measured in deep brain structures when the signal will become increasingly attenuated due to high-frequency absorption and can the noise be suppressed using standard signal processing assumptions such as that noise is Gaussian?

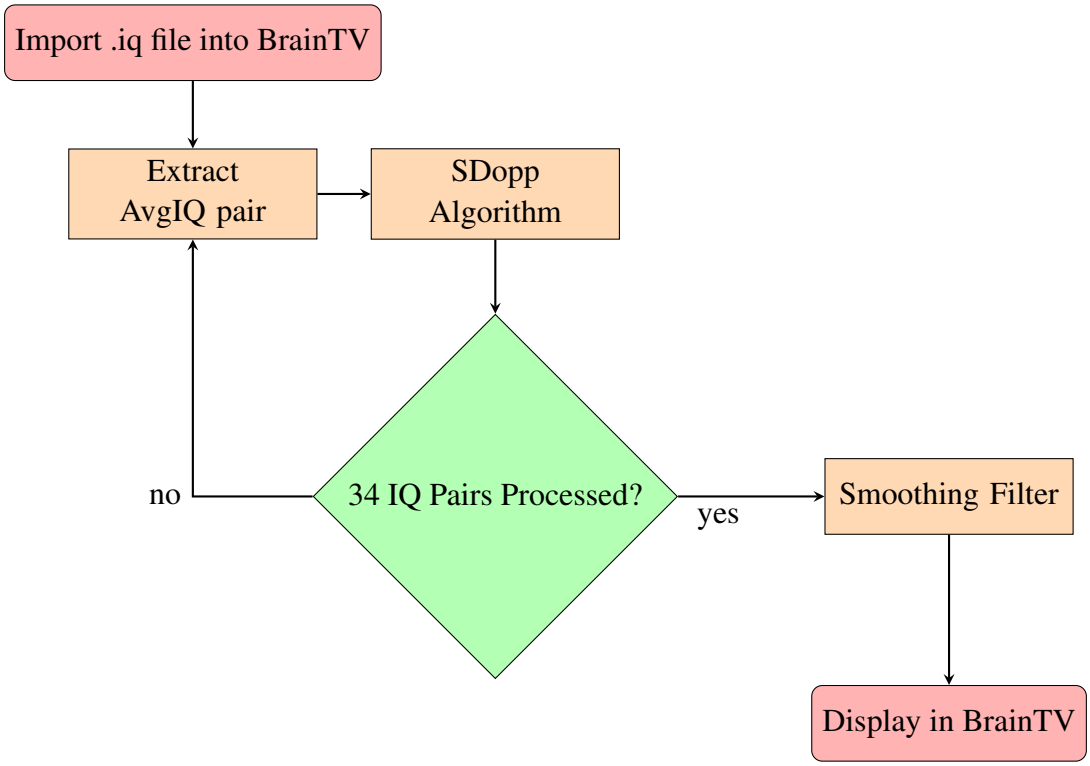
Furthermore, the displayed velocity values in brainTV are currently not in accordance to other blood-flow imaging techniques such as MRI and in-vitro studies investigating blood flow. Standard rates in cerebral capillaries range between 0.5-1.5mm/sec, whereas BrainTV displays values ranging from 0.04-0.18μm/sec [18].

This range confusingly implies values a factor of 1000 off the standard by plotting micrometers rather than millimetters - whether the units are incorrectly used remains unclear. Moreover, assuming millimeter/second rates, the values fall flat under the assumption that velocity should increase at deeper depths/gates. The velocity displayed instead decreases at deeper gates. This is a worrying result that may be due to the absorption artefact and should be addressed quickly.

4. Documentation

This section focuses on the SDopp and the newly implemented Kasai Autocorrelator algorithm. It is a technical documentation on the NK1 data format, how files are generally processed, relevant parameters, as well as the code functionality and improvements.

4.1. Data Format & Processing



In BrainTV, raw Nihon Kohden (NK) data files are directly imported via the user interface in the form of **.iq** files. This file holds 117 samples of complex IQ data, used for extracting features such as blood velocity, CO2 levels, blood pressure, and more.

The processing of NK data for velocity computation can be summarised by the diagram above. Files are read in the MATLAB code RPData_BrainTV.m and BrainTV_gui.m, where their header information (Abstract A) is extracted, along with IQ pairs. This IQ data is averaged and stored in a second NK **.ai** file.

NK files are stored in the temporal Year/Month/Day/Hour/Minute/Second format:

	IQ File	**AvgIQ File**
Ch1	YYYYMMDDhhmmss.iq1	YYYYMMDDhhmmss.ai1
Ch2	YYYYMMDDhhmmss.iq2	YYYYMMDDhhmmss.ai2

Table I: NK File Formats

Where the numbers following the file type represent one of the 2 channels. Only channel 1 was utilized in this project. Furthermore, it is the average IQ (AvgIQ) data that is processed in the two implemented velocity functions within BrainTV.

The data stored in the AvgIQ file is shown in Table II: Each array element stores 4 bytes (32 bits) of data. There are 34 gates / sample volumes recorded, and in turn 34

ch1	I0	Q0	I1	Q1	. . .	I33	Q33
AD	icp1	bp1	icp2	bp2	. . .		
ch2	I0	Q0	I1	Q1	. . .	I33	Q33
AD	icp1	bp1	icp2	bp2	. . .		

Table II: AvgIQ Data Format, each array element holds 4 bytes of data.

sample pairs of AvgIQ data. Therefore the total amount of data stored in each row can be computed:
2 $\times$ **4 bytes** $\times$ **34 samples = 272 bytes** (each channel)

In addition to IQ data, .ai files include AD data, which include features such as gyroscope, accelerometer, heart rate. There are 17 arrays of AD data for each patient:
4 bytes $\times$ **17 features = 68 bytes** (each channel)

Conclusively, for each channel, the size sums to:
272 bytes + 68 bytes = 340 bytes

And, the total sample size:
2 $\times$ **34 IQ samples + 17 AD features = 85 samples of data** stored in each data file.

These NK files are processed under fileID_IQ using the format given above. As the IQ data come in their quadrature pairs (e.g. I0/Q0 or I1/Q1), these elements must be extracted from the data rows individually and passed onto the velocity function in the file VSDopp_v2_BrainTV.m. Therefore, the software first selects one of the 2 channels, and then secondly extracts the AvgIQ pairs - following the derived quadrature demodulation process and iterating until all 34 samples have been processed.

```
RawVel{k} = VSDopp_v2_BrainTV(T,av_data{k,1},nlambda,col);
```

Finally, the raw velocities are filtered for noise using a MATLAB averaging filter and displayed.

4.2. Parameters

In this section of the project, new parameters will be explained as they appear in the following equations. The basis of both velocity functions, as in Equations 16 & 17, are formed by the Doppler Shift:

$$v(m,n) = \frac{\lambda_n}{4\pi T}\left(\phi(m,n) - \phi(m,n-1)\right) \tag{18}$$

- λ_n - Nyquist rate, used as a limit for sampling frequency (metres, m)
- T - Time Interval / Inverse of PRF, used for calculating the mean (seconds, s)
- m - Depth index (metres)
- n - Pulse number index

- $\phi(m,n)$ - Phase of n^{th} signal at m^{th} depth
- $\phi(m,n-1)$ - Phase of the previous signal at the same depth

Substituting the phase difference for the Kasai Autocorrelator estimator, velocity is computed by:

$$v(m,n) = \frac{\lambda_n}{4\pi T} arg\left\{ \sum_{b=2-PL}^{0} S'(m,n-b)S'^{*}(m,n-b-1) \right\} \tag{19}$$

Autocorrelation Parameters: Three parameters introduced

- S'- Complex IQ signal
- S'^{*} - Complex Conjugate of IQ signal
- PL - Package Length, number of pulses the mean velocity is estimated over; dictates the temporal resolution.

Substituting the phase difference for the SDopp algorithm, velocity is computed by:

$$v(m,n) = \frac{\lambda_n}{4\pi T} arg\left\{ \sum_{a=2-ND}^{0} \sum_{b=2-PL}^{0} S'(m-a,n-b)S'^{*}(m-a,n-b-1) \right\} \tag{20}$$

SDopp Parameters: One additional parameter introduced

- ND - Number of subsamples; dictates the length of observation window.

By comparison, it is clear to see that the difference in the algorithms appears in how the phase-shift is computed. The SDopp algorithm is, in fact, merely an expansion of the Autocorrelator algorithm by introducing a single additional variable to allow the observation window to be split into subsamples. The complete parameter list can be found in Abstract A.

4.3. Code & Improvements

A challenge of developing software is providing thorough documentation and comments to allow future changes and progress. Parameter names should match those of the algorithm, and any introduced variables must be explained. Prior to this project, the SDopp code was not clearly implemented. The following code snippet displays the algorithm before changes were made to address the many problems.

```
VD = zeros(length(av_data)-2,length(a));
v = zeros(length(av_data)-2,2);
V_dop = zeros(length(av_data)-2,mm);

for n = 1:length(av_data)-2
    m = 1;
    for k = 1:length(a)
        VD(n,k) = 0;
        for j = 1:length(b)
```

```
10
11                  if n==1 && b(j) == 0
12
13                      v(j,k) = av_data(n+2-b(j),...
14                          m-a(k)).*conj(av_data(n+1-b(j),m-a(k)));
15                      VD(n,k) = VD(n,k)+ v(j,k);
16                  elseif n==1 && b(j) ==1
17                      v(j,k) = av_data(n-1+b(j),...
18                          m-a(k)).*conj(av_data(n-1 + b(j)+1,m-a(k)));
19                      VD(n,k) = VD(n,k) + v(j,k);
20                  else
21                      v(j,k) = av_data(n-b(j)+1,...
22                          m-a(k)).*conj(av_data(n-b(j),m-a(k)));
23                      VD(n,k) = VD(n,k)+ v(j,k);
24                  end
25          end
26          V_dop(n,m) = V_dop(n,m) + VD(n,k);
27      end
28  end
29  ...
30  V_SDopp = (nlambda/(pi*T)).*unwrap(angle(V_dop));
```

This snippet shows only the first iteration of the two-step approach, implemented in the VSDopp_v2_BrainTV.m code. Visibly, there is a lack of commenting which makes the readability of this implementation difficult.

Moreover, it does not follow the original parameter names in the Hoeks et al. 1994 paper, but a PhD thesis by Dr. Kucewicz [14]. Secondly, it is not clear what the for-loops are doing to calculate the phase difference due to the similarly-named variables: **VD, v, V_dop, V_SDopp**. The lack of good comments leaves the process of velocity estimation ambiguous as it is difficult to differentiate as to what each variable is storing, and thus far it has not been certain whether the code was correctly implemented to begin with. The correct implementation was only verified in this project.

In order to address the many concerns of this implementation, debugging methods such as placeholder MATLAB disp() functions were utilized at sections of the code to understand the introduced variables. This method was successful, making it easy to match parts of the code to the SDopp algorithm. This debugging method allowed for creating useful comments as well as revising variable names for added clarity. The MATLAB code was updated to work as usual with the improvements made. The following snippet shows some of the comments and updated variable names.

```
1  %Initialize matrix for storing phase value from Algorithm output.
2  %Size must fit pair of IQ signals.
3  phaseIQ = zeros(length(avgIQ_data)-2,2);
4
5  %Initialize matrix for storing phase values.
6  %Size must fit pair of IQ signals and allow for looping
7  %For 4 subsamples, there are 3 columns.
8  phaseIQ_Matrix = zeros(length(avgIQ_data)-2,length(a));
9
10 %Initialize matrix for storing the final phase values
11 ...
```

The variable changes are summarised in Table III:

Old	Revised
VD	phaseIQ
v	phaseIQ_Matrix
V_dop	phaseDifference_Matrix
V_SDopp	velocityEstimate
av_data	avgIQ_data
mm	GATE

Table III: Variable name changes in VSDopp_v2_BrainTV.m

Similarly, the implemented Kasai Autocorrelator in the VAuto_BrainTV.m was programmed with this clarity in mind. Variable names have been chosen for equal readability, and remain the same as in the SDopp implementation where possible. This can be observed in the following code snippet.

```
for m = 2:GATE
    % Iterate over IQ data points
    for n = 1:length(avgIQ_data)-2
        % Initialize sum for averaging
        autocorrelationSum = 0;

        % Average autocorrelation over PL pulse lengths
        for j = 0:PL-1
            % Calculate autocorrelation
            autocorrelation = avgIQ_data(n+2-j,m).* ...
            conj(avgIQ_data(n+1-j,m));

            % Add to sum
            autocorrelationSum = autocorrelationSum + autocorrelation;
```

5. Discussion

The proved functionality of both the Hoeks SDopp and Kasai Autocorrelator functions raises the question: with future development of BrainTV, which algorithm should be maintained? As observed in previous figures of the **Verification** section, it is clear that at gates <20, the velocity estimations are almost always exact for small observation windows. Therefore the choice must be made on other parameters. This section is a comparison of the two algorithms, investigating theoretical limitations and creating a choice for future research.

The primary assumption made during the verification stage was that the velocity estimations should be similar, albeit **only at small observation windows**. This is an avenue that may be investigated by changing the relevant parameter. In the study by Hoeks et al. (1994), it is stated that performance drastically drops at larger observation windows. Increasing the observation window (ND = 8 then 3) and the package length (PL = 3 then 2) seemed to have a slight worsening effect on velocity output at deeper gates, more easily seen by its first-order displacement form in Figure 13. However, because this data is usually discarded due to lack of mitigation strategies, this effect can be considered negligible for the current time. It is clear that larger observation windows have

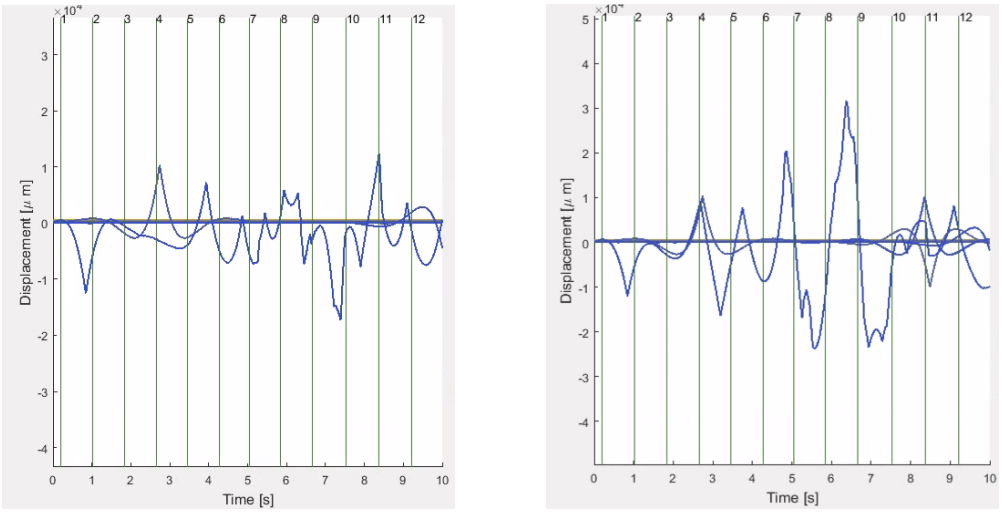


Figure 13: Tissue displacement, Gates 1-30; Left figure shows displacement at normal parameters. Right figure shows displacement at larger observation windows. Signals at gates 25-30 are amplified.

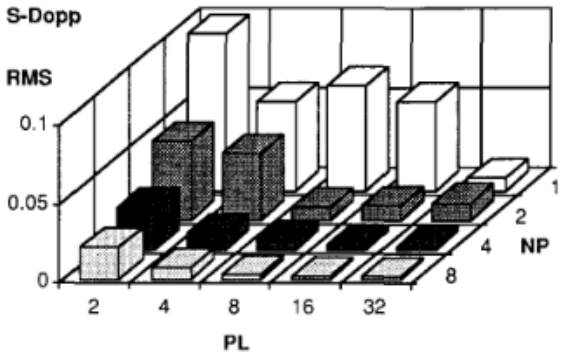


Figure 14: RMS error in SDopp velocity at increasing observation windows. Interestingly, when both NP and PL increase simultaneously, error improves. Adapted from [5]

an amplifying effect on the noise of the signal in the SDopp algorithm. Autocorrelator does not utilize subsampling and therefore this amplification is not observed. This general effect can be seen in Figure 14, where temporal resolution and length of observation window may be interchanged.

Furthermore, it is assumed that reducing the contribution of noise is dependent on the signal bandwidth and in turn the quality factor of noise Q_n and signal Q_s:

$$Q = \frac{w_0}{BW} \tag{21}$$

where w_0 is defined as the center angular frequency of the transmitted pulse and BW is the noise and signal bandwidth.

Due to the requirement of imaging small blood vessels, the pulse frequency must be high. The bandwidth of an ultrasound pulse is directly proportional to the frequency deviation of the modulating signal and the carrier frequency. This inevitably implies wide RF bandwidths (and small Q_s), which increases potential for aliasing, increased bias and ambiguous results. This is in line for theoretical expectations of overlapping harmonics - RF signals must ideally have narrow bandwidths - however this would require the pulse frequency to be decreased, in turn paradoxically reducing resolution of sample volumes.

Nevertheless, the carrier frequency must be optimized dynamically at different sample volumes in both software and hardware; transmit and receive circuits within the probe have potential to generate random noise and increase bandwidths. Because both the SDopp and Autocorrelation algorithms are limited by their physical constraints, methods of filtering must become more complex. Neither algorithm significantly outperforms due to this technical challenge.

In the evaluation of the SDopp estimator, SNR have been computed under certain assumptions: "a specified SNR value was achieved by randomly selecting a signal segment with a Gaussian spectral distribution and adding to composite signal. It was assumed that the signal bandwidth was governed by the emitted signal and the frequency characteristics of the transducer" (Hoeks et al. 1994). It is an assumption in signal processing and Hoeks' study that noise is Gaussian. For biomedical signals, this does not generalize well. Noise may be caused by artefacts and therefore it is arguable that the signal is **inherently** correlated with its noise. Thus the process of smoothing/filtering may amplify noise levels, further yielding misleading velocity values. Additionally, in the case of both algorithms, the phase relationship of the emitted ultrasound signals vanishes with increased depth/gates; causing amplitude and noise to add up randomly.

SNR can be defined as:

$$SNR = 10\log_{10}\frac{M_{blood}}{M_{noise}} \tag{22}$$

where M_{blood} and M_{noise} is the mean amplitude of the blood-flow signal and noise respectively. Calculating the amplitude of noise does not appear trivial.

The resultant SNR is independent of the length of the sample volume depth. It may be possible to improve the SNR using signal averaging as an additional filter, among other more advanced techniques such as the 'matching pursuit' method [19] and Deep Learning. Nevertheless, averaging repetitions of 100 can improve SNR by a factor of 10 [20].

$$SNR_m = \frac{m \cdot M_{blood}}{\sqrt{m} \cdot \sigma_n} = \sqrt{m} \cdot SNR \tag{23}$$

where SNR_m is the improved SNR after m repetitions, and σ_n is the Root-Mean-Square (RMS) error.

The severity of RMS error in the SDopp and Autocorrelation algorithms have been studied. For low SNR values (e.g. SNR = 5 dB), greater errors can be observed using the SDopp estimator. The performance of both algorithms is equal at shallow depths near the cortex of the brain - yielding the same velocity values. At deeper gates, both functions perform poorly. It can be assumed that the performance relies primarily on the severity of attenuation.

In summary, at identical parameter combinations the velocity outputs of both the SDopp algorithm and Autocorrelation algorithm are close to identical. This project suggests that the choice of algorithm is completely dependent on the level of documentation available. In practice, it appears that SDopp is superior at handling noise for sample volumes near the cortex, whereas the autocorrelation algorithm may be more appropriate at imaging deeper brain structures. Both algorithms are limited by bandwidth, the design of the

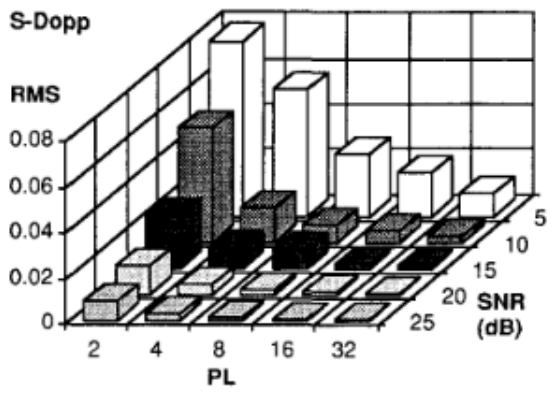


Figure 15: RMS error in SDopp at increasing SNR values. Adapted from [5]

probe and physical artefacts. All of these issues must be addressed in order to improve the quality of the velocity estimate.

The Kasai Autocorrelator does have a greater level of documentation available in research. On the other hand, in the context of healthcare, velocity values that are less prone to noise, have greater resolution and accuracy may be more appropriate for BrainTV. As both algorithms are limited by extraneous constraints, it is recommended to explore novel ultrasound imaging methods and compare performance with different sets of observation windows, temporal resolutions, SNR values, among other parameter combinations in order to make an informed choice.

6. Challenges

This section summarises the various technical problems encountered during the course of the project, as well as general issues in medical ultrasound imaging literature. It may be used as reference for the basis of future studies and projects.

6.1. Code Inconsistencies & Crashes

The scope of this project focused on precisely evaluating the SDopp algorithm and its relevant code modules. It is therefore still uncertain how all data stored within NK files is processed. This leads onto some code inconsistencies, namely one being the problem of the Time Interval variable, T. In the following snippet from the file RPData_BrainTV.m, the SDopp function is called to compute the velocity:

```
RawVelocity{i,1} = VSDopp_v2_BrainTV(PRF_new,IQavgch_ds{i,1},nlambda..)
```

The inputs to this function are the PRF, the pair of demodulated IQ signals, λ_n, and the number of gates. The issue arises in the velocity function:

```
function velocityEstimate = VSDopp_v2_BrainTV(PRF,avgIQ_data,...)
velocityEstimate = (nlambda/(2*pi*T)).*unwrap(angle(phaseDif...));
```

It is clear that the PRF value (500kHz) is used to compute the Doppler velocity, rather than its inverse, which would lead to incorrect units. This inconsistency can be shown by the Doppler Shift Equation 2. There is no clear indication as to why this code appears this way. When adjusting the code to its supposed correct form, BrainTV crashes after importing a data file. However, the code of BrainTV_gui.m conflicts this issue with the function being called correctly:

```
RawVel{k} = VSDopp_v2_BrainTV(T,av_data{k,1},nlambda,col);
```

Though it could be implied that the RPData_BrainTV.m is therefore not utilized to compute the velocity - this is not the case. When changing the called SDopp function to the implemented Autocorrelator function, BrainTV displays different velocity and displacement graphs. It is not clear what the intention behind this discrepancy is and should be addressed.

Additionally, in the verification stage of the study, blood velocity values in BrainTV were compared to standard values. It was shown that the μm/sec units displayed in BrainTV heavily disagree with the usual reported mm/sec values. Standard rates in cerebral capillaries range between 0.5-1.5mm/sec, whereas BrainTV displays values ranging from 0.04-0.18μm/sec [18]. This suggests that velocities are a factor of 1000 off. This issue must be addressed.

6.2. Attenuation

In the study by Turner et al. (2020) [13], the researchers had to discard 5% of the BTP data due to artefacts in the form of motion (coughing, blinking, probe movement) and noise at deeper depths (82-86mm). Although this seems like a relatively low discard rate, there are implied concerns regarding standardization and replicability of the data collection methodology. The lack of standardization and mitigation strategies to tackle artefacts prevents the development of clinical diagnostic applications, particularly at deeper gates.

It has been made clear through this project that the main limitations of Doppler PW systems are shared with physical limitations of ultrasound signals. Optimizing the carrier frequency of the transmitted pulse at different depths of the brain is a crucial aspect in reducing aliasing and the effect of absorption. Unfortunately, high frequency ultrasound is a must to access the small target vessels in the cerebrum.

Future research must be done to help validate the current findings and techniques, possibly through direct replication. Researchers have a responsibility to clearly list their equipment apparatus to make this accessible.

Attenuation in general is a topic of debate in literature. Some researchers highlight the benefits of acoustic impedance in probe fabrication. In this project it has been made apparent that the hardware of the probe affects the bandwidth of the ultrasound beam, and therefore the SNR at deeper sample volumes. In contrast, some studies report on the importance of achieving high resolution with the trade off with high absorption. Exploring the impact of attenuation on pulse wave systems should be essential - with a heavy focus on understanding the reason of physical artefacts. This will require greater collaboration with researchers specialized in brain anatomy.

6.3. Biased samples & Ethics

In the study by Ince et al. (2020) [21], the researchers perform a systematic review of 10 BTP papers. Most studies, including the one that collected BrainTV data, currently claim to only sample healthy volunteers. A lack of standardization of the sampling process raises ethical concerns, particularly what makes a healthy 'sample'. Researchers must be transparent and report their procedures to ensure integrity of this novel research and ensure replicability of this sampling process to form a coherent agreement in

methodology among researchers.

There is also a lack of consideration in terms of extraneous variables such as undiagnosed medical conditions or the apparatus used. It is important to account for confounding factors to ensure reliability of their results. Access to this information would have been beneficial in this project, having a broader context as to why certain artefacts may occur.

Furthermore, emphasis must be put on the level of training of the operator during data collection, as well as making training standardized to ensure consistent data collection and replicability of results.

7. Conclusion

This project looked to evaluate the SDopp algorithm and contextualize its use by providing sufficient background literature of the scanning process, required ultrasound physics and signal processing. The paper begun by introducing ultrasound as a soundwave, providing standard parameters in Doppler imaging and mentioning the challenge of attenuation. Following on, the method of PW Doppler scanning was explored with the aid of the Doppler Frequency equation - this further allowed a conversation on sound wave harmonics and their importance in understanding signal aliasing and noise. The method of calculating velocity by the Doppler Shift was shown. To expand on the process of gathering BTP data, quadrature demodulation and its affect on the transmitted frequency spectrum was derived and discussed. Finally, this gave context to introducing the two algorithms of this project: the SDopp algorithm and the Autocorrelator algorithm.

Furthermore, the SDopp algorithm was verified through the implementation of the Autocorrelator algorithm in BrainTV. Values were compared to those of standard values in accepted MRI imaging studies and found to be problematic in scale. Nevertheless, the SDopp algorithm was shown be correctly implemented.

The technical documentation of this project was the main focus - allowing researchers to gain an overall understanding of the algorithms through background context as well as explained data and file formats, parameters, and code. The programming involved in this project improved the readability of the code through in-depth commenting as well as revised variable names for added clarity, reflective of the knowledge gained through this research.

The knowledge gained through this project allowed for an additional discussion of the two algorithms. It was shown that both algorithms are limited by the same physical constraints; albeit the SDopp algorithm performed better at its original parameters, producing velocity signals with less noise compared to the Autocorrelation algorithm. The topic of attenuation and signal noise was given greater focus, conclusively showing that Doppler velocity functions are limited by assumptions of Gaussian noise, frequency characteristics of the traunsducer probe and physical artefacts.

Finally, challenges encountered during the project were addressed. These are issues related to code inconsistencies in the BrainTV_gui.m and RPData_BrainTV.m MATLAB modules of BrainTV, as well as the aforementioned issue of scaling in displayed velocity graphs. Additionally, challenges in literature, such as data discard rates, sampling

methodologies and ethical concerns were assessed in relation to difficulties encountered in this project.

In conclusion, this project's scope was achieved and surpassed through verifying the SDopp implementation. Noise mitigation strategies such as averaging repetitions were suggested for further exploration. Albeit, plans to suppress noise through the use of supervised learning methods were erased after re-assessment of the project goals: labelling a large IQ dataset and training a model to identify the many artefacts would have had a time-cost beyond the deadline of this project, while requiring a greater understanding of brain anatomy and signal attenuation to discriminate between meaningful and inaccurate blood velocities. Additionally, a Github repository was not needed, as all MATLAB code were archived within the relevant BrainTV folder repository.

It is recommended that future research should focus on the topic of noise suppression in biomedical signal processing and explore promising novel artefact mitigation methods such as Deep Learning methods [22].