Arterial Vascular Impedance Calculation

AIM: To calculate Arterial Vascular Impedance, $\boldsymbol{Z}[\boldsymbol{k}] = \frac{\boldsymbol{P}[\boldsymbol{k}]}{\boldsymbol{Q}[\boldsymbol{k}]}$, where ***P*** is intra-arterial pressure (in mmHg) and ***Q*** is blood volume-flow (in ml/s).

Currently, the data (1,2,3) can be collected simultaneously using CMCdaq without any time- synchronization issue.

STEP1: The following are the data collected simultaneously:

1. Intra-arterial pressure (***p[n]***) at 4 KHz.

2. Video recorded at 30fps. 3. Audio recorded at 4KHz	$\Longrightarrow$	From Ultrasound Machine

STEP2: Conversion to frequency domain:

2.1. ***p[n]*** **→** $\boxed{\textbf{DFT}}$ **→** ***P*** $[\boldsymbol{k}]$ **, Sampling rate= 4000 samples/s.**

2.2. Calculation of volume-flow: ***q[n]***

2.2.1. Frame-wise area calculation: ***a[n], n= frame no***.

2.2.2. For one frame duration, no: of audio samples = 4000/30 = 133 samples/frame: ***x[n], N=133***

2.2.3. ***x[n]*** → $\boxed{\textbf{DFT}}$ → ***X*** $[\boldsymbol{k}]$ → $\boxed{\textbf{\textit{Averaging}}}$ = average doppler shifted freq ($\boldsymbol{f_D}$) → conversion to average velocity, ***v[n]***, using, $\mathbf{f_D} = \frac{\mathbf{2}\,.\boldsymbol{f}.\boldsymbol{v}\cos\boldsymbol{\theta}}{\boldsymbol{c}}$, f is the probe txn freq=11MHz, c is velocity of soud in blood=1580 m/s, $\theta = 60^{\circ}$.

.

2.2.4. Hence, calculation of volume flow, ***q[n]= a[n] . v[n]***

2.3. Conversion of q[n] to frequency domain:

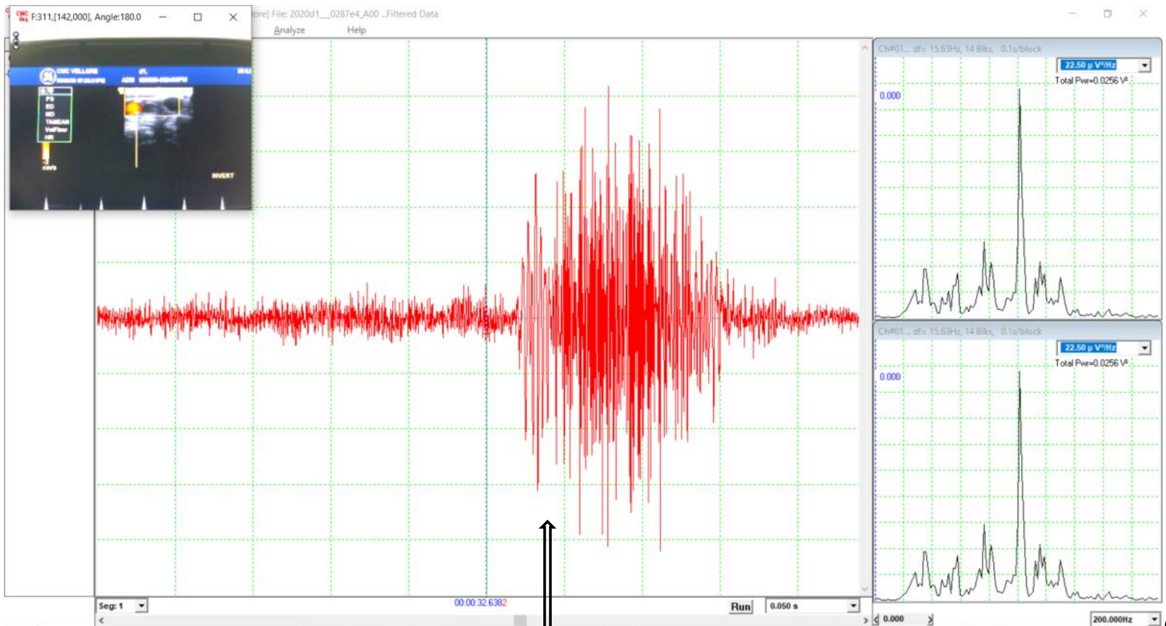
q[n] **→** $\boxed{\textbf{DFT}}$ **→** ***Q [k]*** **→** $\boxed{\textbf{\textit{Zero padding to 4000 samples}}}$ **→** $\boxed{\textbf{IDFT}}$ **→** ***q[n], N=4000***

2.4. Arterial Vascular Impedance, $\boldsymbol{Z}[\boldsymbol{k}] = \frac{\boldsymbol{P}[\boldsymbol{k}]}{\boldsymbol{Q}[\boldsymbol{k}]}$

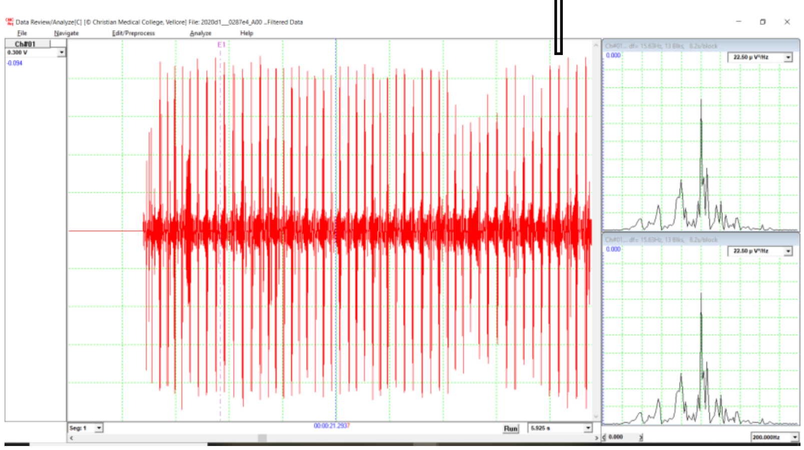
Summary of Calculations from Video (10fps) and audio (4KHz) collected simultaneously:

Data collection:

1. The doppler sound was recorded at 4KHz and video was recorded at 10fps , using CMCdaq, simultaneously. **Extraction**: The doppler signal was low pass filtered at 31 Hz cut off. Also, 50Hz noise was filtered; the negative time (in the extracted text file) was corrected so that the video and doppler signal starts at 0 time instant.



one out of many bursts shown



2. Intra-arterial pressure – to be collected simultaneously along with video and doppler on CMCdaq.
3. Volume Flow = (Mean flow velocity). (Area of cross-section)
4. Area of cross-section calculation:

4.1. **Area calculation Algorithm**:

Principle: Covert to binary weighted image and calculate the area based on the number of pixels within the boundary detected. This is converted to standard units based on PPI -value of ultrasound machine.

Initial Image processing techniques:

Input Image= A
Laplacian =B
A+B=C
Sobel (C) =D
Avg(D)=E
C∗E = F
A+F =G

Histogram plotting for the pixels.

Determine the threshold for conversion to binary weighted image.

Non-maximum suppression.

Clear the lighter connected structures within the boundary.

Calibrate area ‘a’ to standard units (sq.cm) based on positioning and ppi of cam used.

5. Mean flow velocity frame-wise:

Extract frames from videos:

For each frame count ⇩ ***+ Area calculation Algorithm (‘a’ sq.cm)***

Get the frame current time: t2

⇩

Get start time, t1, for this frame: t2-(1/frame rate)

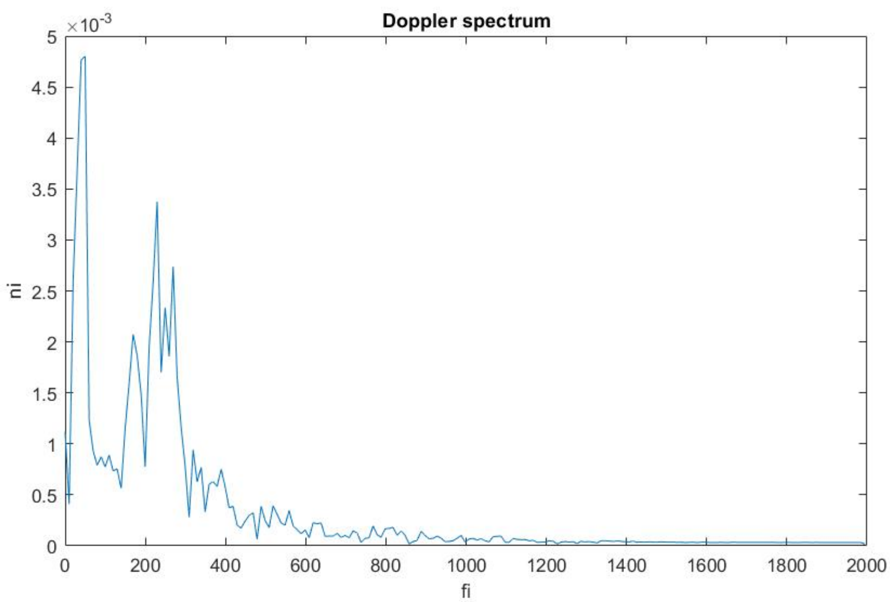
⇩

Extract the doppler signal data between t1 and t2

⇩

Determine the L-point DFT spectral density of this L-point signal segment

⇩



⇩

$$Mean\ doppler\ frequency, f_D = \frac{\sum_{i=0}^{\frac{L}{2}-1}(f_i\ .n_i)}{\sum_{i=0}^{\frac{L}{2}-1}(n_i)}$$

$f_i : i'th\ discrete\ frequency\ component\ from\ the\ spectrum$

$n_i\ : the\ power\ spectral\ density\ of\ the\ i'th\ frequency\ component$

⇩

Mean doppler shifted frequency, $f_D = \dfrac{2.f.v\cos\theta}{c}$

Hence, Mean velocity, $v = \dfrac{f_D.\,c}{2.f.\cos\theta}$

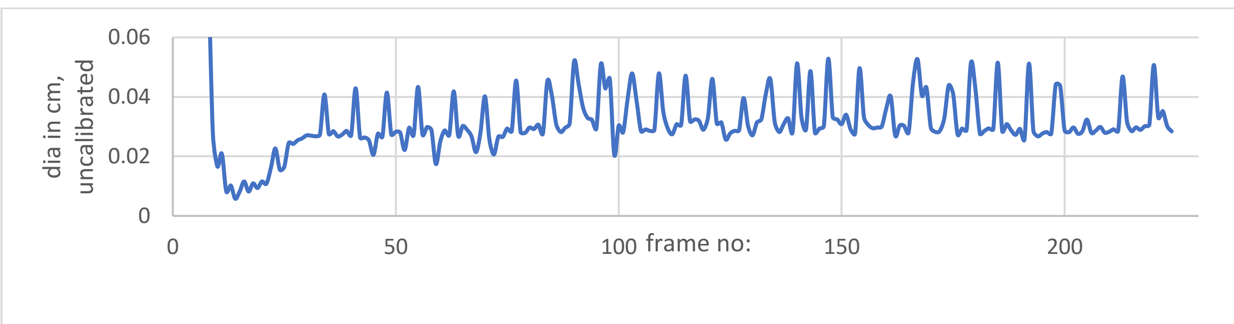
Where, velocity of sound in blood, c = 1580m/s ; θ = *angle of insonation* ; f = *transmitted frequency* = 8MHz; *(convert to cm/s)*

For this frame, calculate volume flow, $q = a.v$ *; in cc/s*

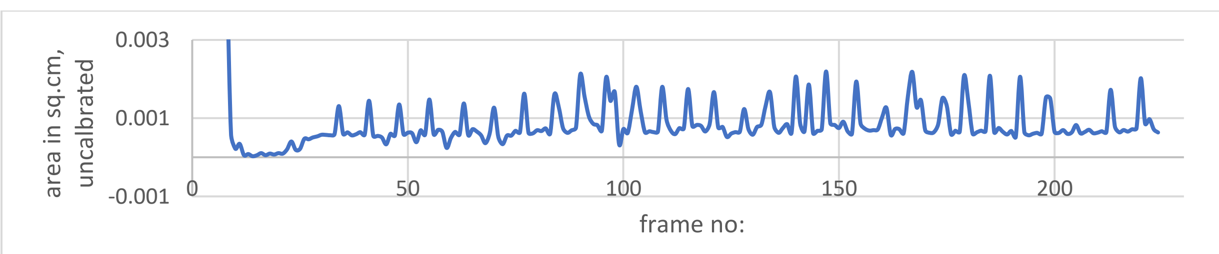
Repeat over all frames

Results:

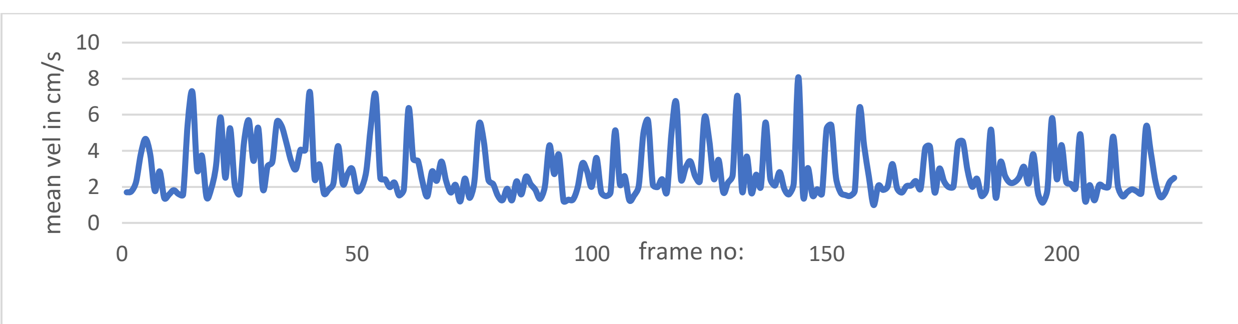
Diameter vs frame no. (NB: need to calibrate for correct dia measurement based on the webcam position which was not done for this data).



Area vs frame no:



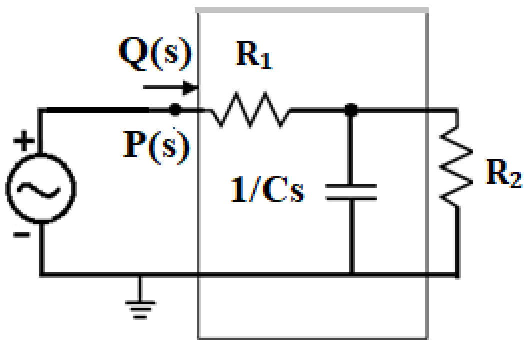
Velocity vs frame no:



Hence, Volume flow will be: *area. Velocity*

6. **As mentioned in page 1, 2.4, arterial vascular impedance can be calculated in frequency domain.**

Physiological Interpretation:
The arterial lumped model can be simply modeled as a low pass filter as shown below, where, R_1 and C: compliant vessels; R2: (non-compliant vessels + veins + capillaries).



In Frequency domain, net arterial impedance = Z(s) =

$$\frac{P(s)}{Q(s)} = \frac{(R_1.\, R_2.\, Cs\, +\, R_2\, +\, R_1)}{(R_2.Cs\, +1)}$$

$$\frac{P(j\omega)}{Q(j\omega)} = \frac{(R_1.\, R_2.j\omega C\, +\, R_2\, +\, R_1)}{(R_2.j\omega C\, +1)} = Z(j\omega)$$

$$|\,Z(\omega)\,| = \frac{\sqrt{(R_1^2 + R_2^2\, + 2\, R_1.R_2\, +\, R_1^2.R_2^2.\omega^2.C^2)}}{\sqrt{(1\, +\, R_2^2.\omega^2.C^2)}}$$

$$\angle\, Z(\omega) = \tan^{-1}\left(\frac{\omega.C.R1.R2}{R2\, +\, R1}\right) - \tan^{-1}(\omega.C.R2)$$
$$= \angle(\text{Pressure}) - \angle(\text{Flow})$$

→ **Arterial Vascular Impedance, $|\,Z(\omega)\,| . \angle\, Z(\omega)$**

From the **phase response** of Arterial Vascular Impedance, the nature of R and C of the model can be discussed.

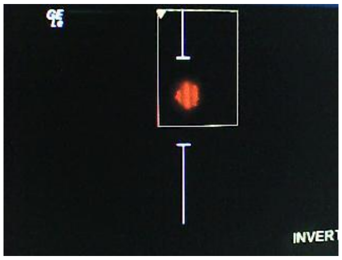
For instance, a phase lag of 90° between voltage and current would imply a pure capacitive component, while an in-phase V and I denotes purely resistive circuit. Voltage and current electrical equivalence explains how pressure and flow would work in the lumped model for arterial tree. More the compliance factor, we would expect a phase difference between pressure and flow (leading, analogous to current) waveform. Lesser the compliance or more the resistant / stiffer the vessels are, minimum (or no) phase difference would be expected between pressure and flow. Area of cross-section is constant for a stiffer vessel. In a compliant system, the phase would be negative between 0° and 90°. In a resistant system, it would be zero. Hence, it is crucial to preserve the phase information of all these signals through synchronous data collection which can be achieved using CMCdaq. Lead II ECG data along with (doppler+video+pressure) channels would give additional information on cardiac cycle timing information for any validation.

From the **frequency spectrum**, solving for 3 equations from 3 impedance values, R1, R2 and C1 can be resolved.

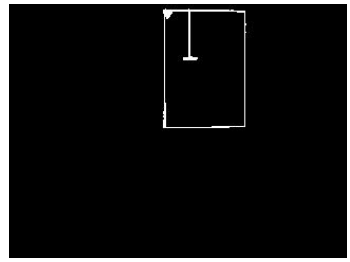
Recent updated code:

Problems with the previous analysis:

1. Recording of doppler imaging mandates use of a window of interest and this would interfere with the thresholding used in the program. Some frames analyzed does not detect the vessel correctly. Instead it generates a frame with window frame detected.



→



Original frame

Wrong output frame

2. Currently, the program has been modified for resolving this issue:

Extracting frames from the video (30fps)

⇩

CLAHE method of histogram equalization

⇩

Binarize the image frame

⇩

Split to R, G, B layers

⇩

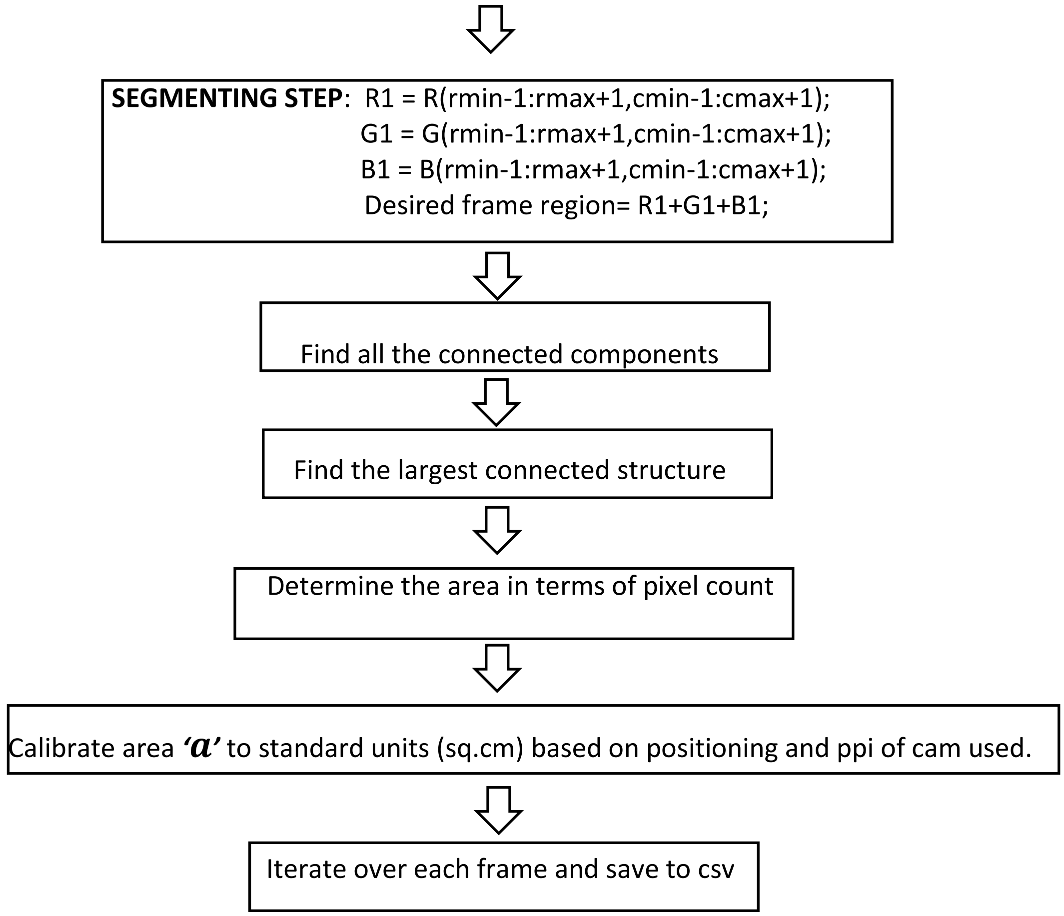
Eliminate the background- doppler window + patient details + any other legends from Ultrasound screen: F = R-G

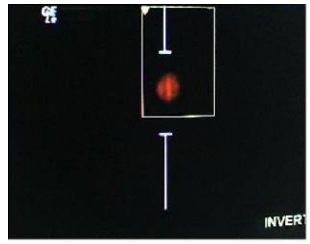
⇩

Iterate over (5x5) pixel neighborhood to detect the boundary for F based on the white pixel count

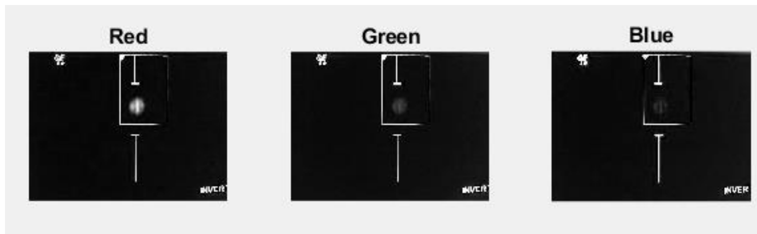
⇩

Find row_max and row_min;
col_min and col_max of F. This would be the region of interest for segmentation of the image frame.

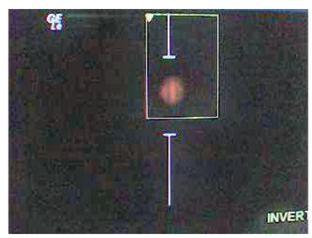

SEGMENTING STEP: R1 = R(rmin-1:rmax+1,cmin-1:cmax+1);
G1 = G(rmin-1:rmax+1,cmin-1:cmax+1);
B1 = B(rmin-1:rmax+1,cmin-1:cmax+1);
Desired frame region= R1+G1+B1;
Find all the connected components
Find the largest connected structure
Determine the area in terms of pixel count
Calibrate area 'a' to standard units (sq.cm) based on positioning and ppi of cam used.
Iterate over each frame and save to csv



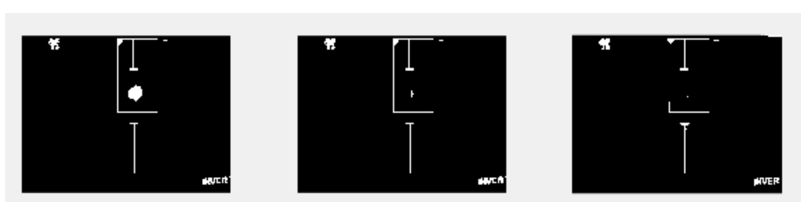
Original Image



RGB Layers

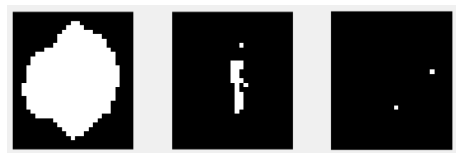


CLAHE EQUALIZED

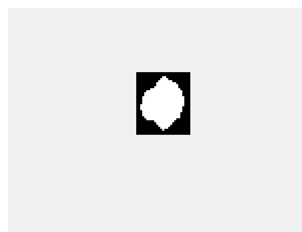


Binarized and then RGB layer split

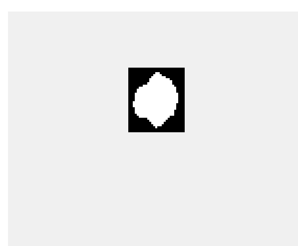
R_G= F; for windowing



R1, G1, B1 layers after windowing R,G,B with the row and column indices obtained from F



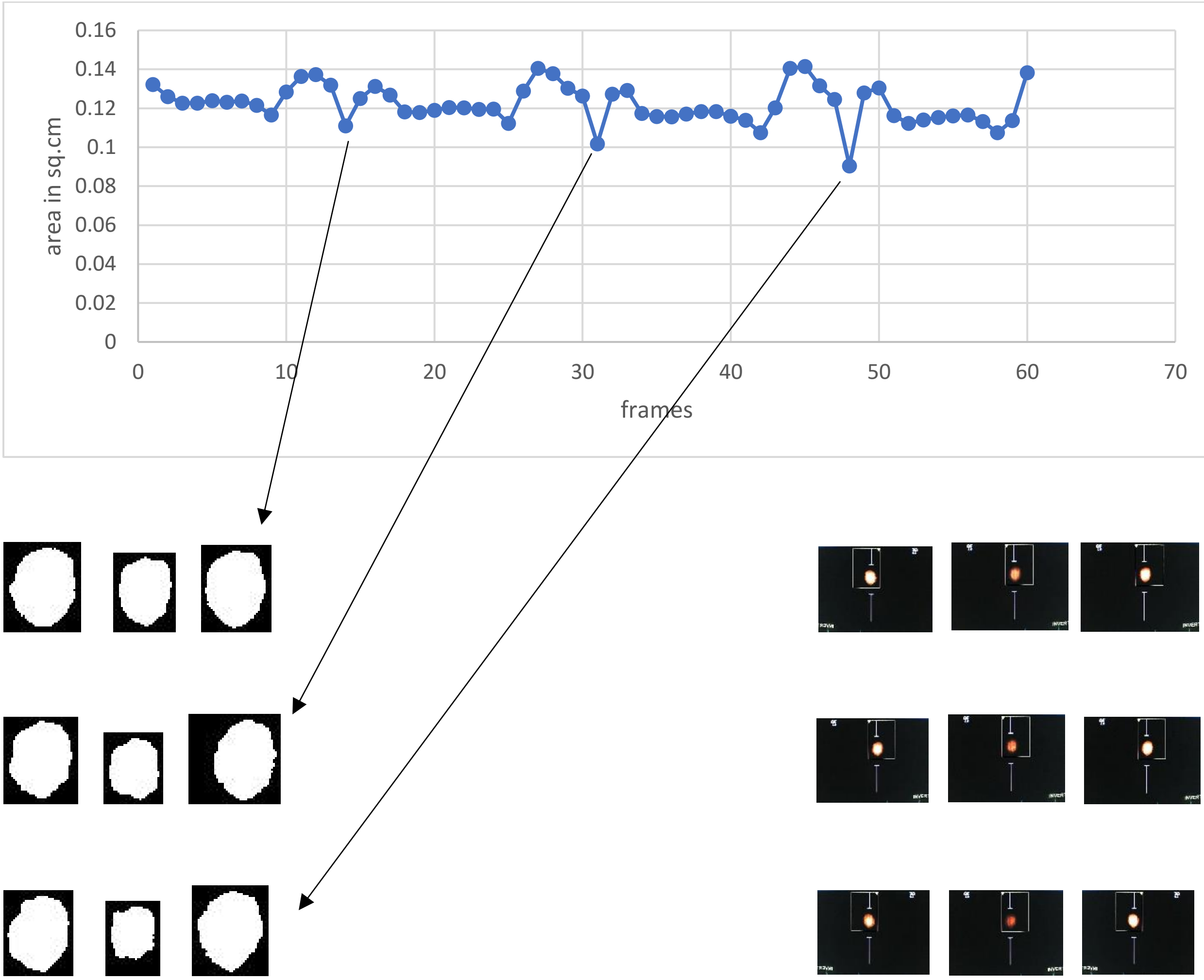
R1+G1+B1



Largest connected component and area calculation

NB: Without CLAHE, output is reduced in area

Results:



Drawbacks identified:

Some frames (482,483,484) were detected like these:

The windowing issue had to be sorted for 1st and 3rd image frames. Frame 2, though had detected the area coresponding to the central region of higher velocity alone, the area needs to be corrected for the complete outer lighter periphery which is not detected. Such frames were identified based on the count of white pixels and for these frames, thresholding was done before segmenting the region of interest. This resolves the area calculation and hence all those points corresponding to a sudden drop in area.

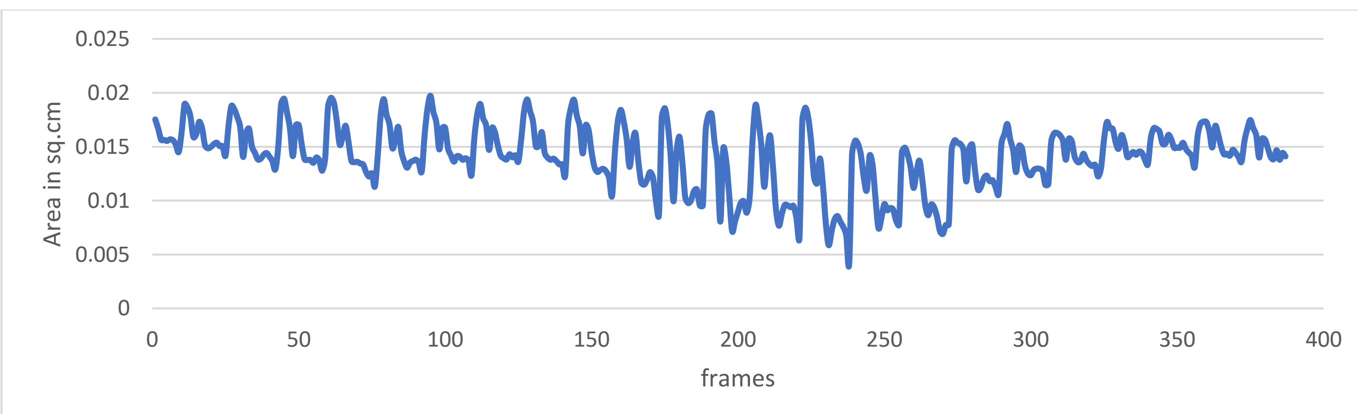
The modifications in algorithm:

i. Perform till the SEGMENTING STEP as in the above flowchart.
ii. Detect the frames based on the count of white pixels in the 5x5 neighbourhood scan of the image.
iii. Perform SEGMENTING on the R-layer which is thresholded to detect the entire periphery of the vessel and not just the brighter centre.

```
R = img1(:,:,1);      // R,G,B from original image
G = img1(:,:,2);
B = img1(:,:,3);
R = R<threshold;  //thresholding
ar1 = R(rmin-1:rmax+1,cmin-1:cmax+1);  //segmentation using the window range identified from step-i.
ar1=imcomplement(ar1);
```

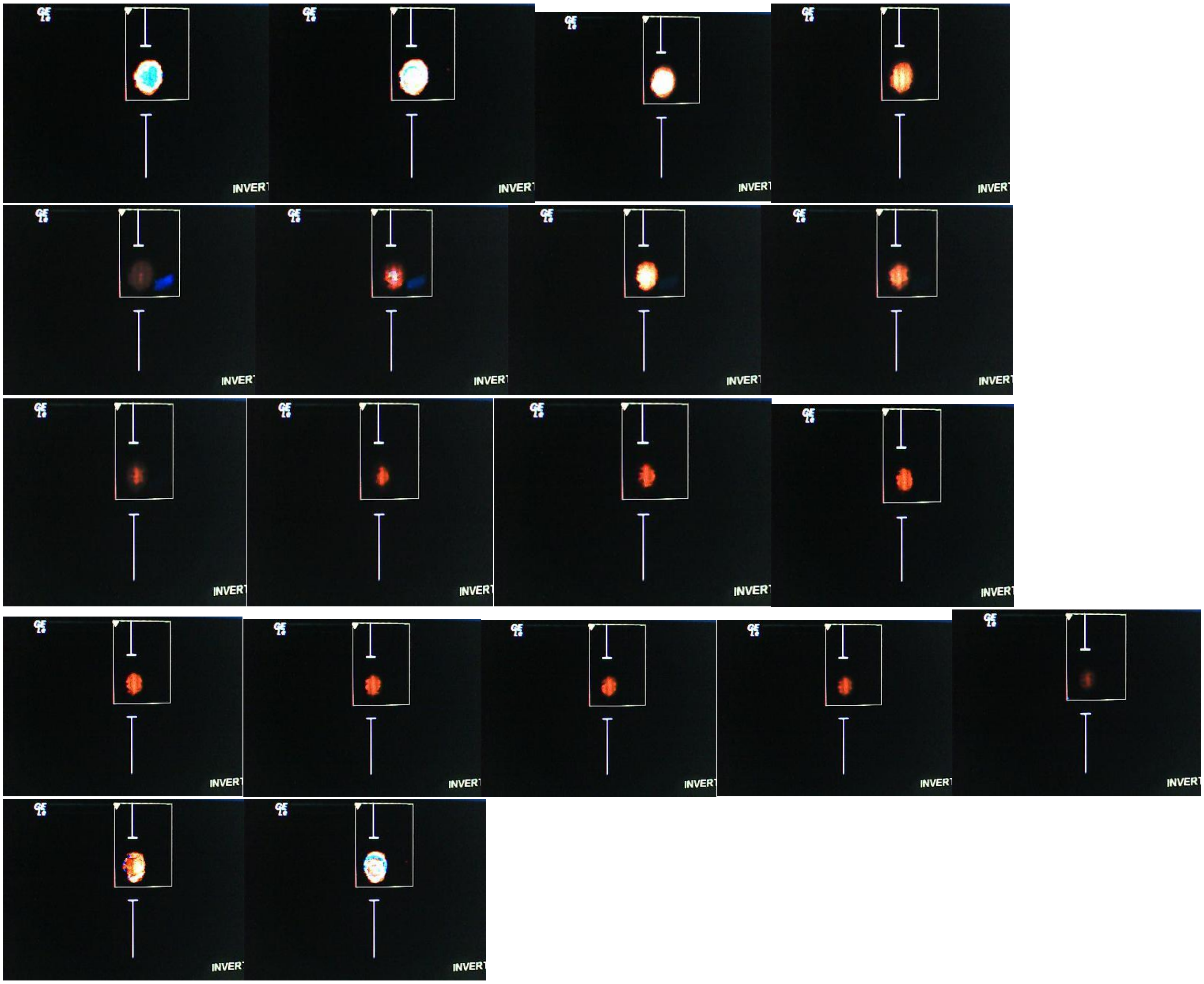
iv. Now determine the largest connected component and the area.

Results from this method which is the latest version used for area calculation:

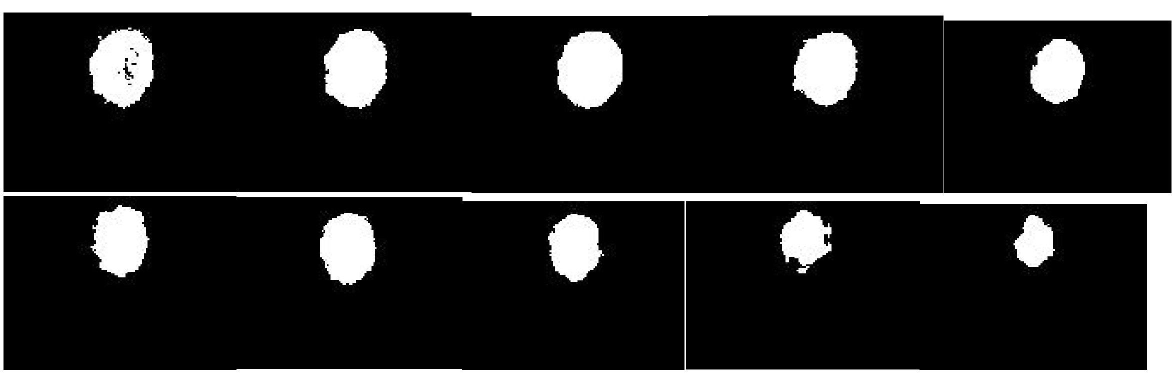


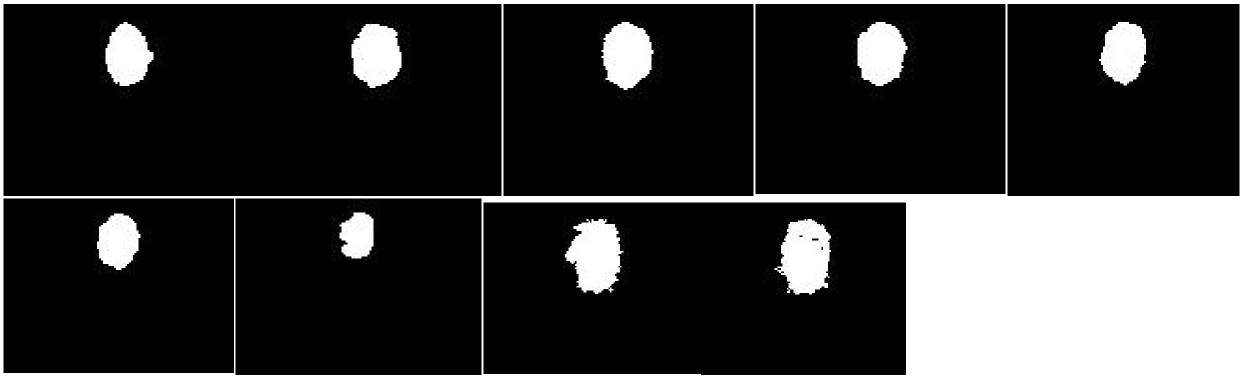
Verification:

Frames: 222 to 240, the frame areas seem to drop a lot. Hence those frames were cross-verified with the corresponding detected outputs.



These frame areas are detected correctly:





Hence, verified.