Digital Signal Processing in Clinical Validation Studies with Impedance Plethysmography

Ravi Shankar, Mural Gopinathan, and John G Webster

ABSTRACT:

We provide here an automated method for calculating the volume change (ΔV) from the tissue impedance change (ΔZ) signal acquired using an impedance plethysmograph. This presents our approaches to signal processing in two of our studies: a longitudinal pathologic validation study on monkeys, and a human clinical validation study. In both these studies, plethysmographic data were acquired at three different locations simultaneously. The data were acquired for about 3 to 4 s in the case of monkeys. For human subjects, data were acquired for 3 to 6 s. Data collection was repeated multiple times with various external cuff pressures. We employed the signal averaging technique to get one average ΔZ pulse, which was used to calculate the ΔV value at each cuff pressure using Nyboer's equation. We provided a mechanism to automatically reject any noisy portion of the signal. Care was taken to minimize operator intervention in the process leading to a totally automated analysis of the signal. Such an automated system would be useful for real-time processing needed in long-term monitoring.

INTRODUCTION:

Processing of physiological signals is at the heart of any diagnostic system. Automation of such processing is complicated by physiological variability and drift, large dynamic range, motion artifacts, low signal-to-noise ratio, ac power noise, poor skin–electrode contact, and instrumentation-induced artifacts. In a clinical validation study, as with instruments used for long-term monitoring, a significant amount of data are collected and there is need for real-time quick analysis. Digital signal processing algorithms used for nonphysiological signals (as with cell phone communication) require a fundamentally different approach; their signal characteristics are more narrowly defined and may be taken for granted. For physiological signals, signal variability is the norm and may also contain useful information. Thus, a methodical approach to systematically address the challenges will not only help us obtain signal averages, but also give us new insights as we strive to acquire even weaker signals and improve instrumentation. Our experience is built on more than a decade of working with monkeys and human subjects. We noninvasively recorded impedance plethysmographic signals on the limbs in both groups of subjects, for up to 30 min, in research or clinical settings. Our raw data per subject amounted to about 20 MB from 6 data collection sequences and 3 simultaneous physiological sites. We automatically extracted from this typically 200 averaged single pulses, each representing a different site and a different external cuff pressure. These pulses had periods of about 0.5 s (in monkeys) and 1 s (in humans).

Here the physiological signal under study is the pulsatile change in tissue impedance at various cuff pressures. The acquisition and preprocessing were performed as described in our other papers on 3-channel plethysmography and software (Shankar, Shao, and

Webster, 2008; and Shankar, Martinez, and Webster, 2008). The signal was collected at three different locations simultaneously; namely, the left arm, left leg and right leg for monkeys, and the left arm, left thigh and left calf for human subjects. With each configuration, the measurement was made 2 times, leading to 6 measurements per visit on each monkey. The same procedure was followed for the human subjects. In one of the reference channels, an external pressure cuff was used to vary the transmural pressure across the arterial wall. We ignore these aspects here since they do not add value to the discussion here.

The main objective of the signal processing algorithm was to obtain one good pulse from the 5 to 6 pulses that were collected. The technique most used to do this was that of signal averaging, which lead to noise reduction. For this purpose, we needed at least three good pulses for averaging from the 5 or 6 that were obtained. The major steps involved were to identify and isolate the pulses from the raw data, and superimpose them with respect to a common reference point, and average them. The reference point chosen for this purpose was the maximum slope point of the signal. The algorithm had to automatically take care of the human and instrumentation induced interference in the signal. The overall objective was to develop a fully-automated real-time signal processing algorithm to support noninvasive and long-term acquisition physiological signals.

GLOSSARY OF TERMS

Detrending: The process of removing the signal drift seen in impedance pulse recordings.

Dicrotic Notch: An occasional second peak seen in the impedance pulsations, possibly due to pressure pulse reflection at a peripheral site in the circulation.

End Point: A foot point that immediately follows in time a valid foot point, and meets certain minimum and/or maximum time and threshold criteria.

Foot Point: The time instant at which the slope of the volume change signal has a PZC.

Foot-to-Peak Amplitude: The height of the volume change signal from the foot point to the time instant of peak amplitude obtained during systolic upswing.

Frequency Wraparound: Sampling at a frequency lower than the maximum signal (and noise) content causes frequencies higher than the sampling frequency to "wrap around" and appear as lower frequencies. This phenomenon is the same as aliasing.

Jump: A rare, instantaneous discontinuity seen in impedance pulse recordings with the system used in human subjects [Shankar, Shao, and Webster, 2008]. This results from nulling of the high frequency carrier signal to increase the modulation index.

MSP: Maximum Slope Point, occurring between a PZC and an NZC.

NZC: Negative Zero Crossing

PZC: Positive Zero Crossing

SID: Subject ID. Recorded in place of the subject's name in human data files.

Width Threshold: The minimum acceptable time interval between two consecutive MSPs in a given stage of DSP.

Z: Tissue impedance

ΔV: Change in arterial volume

ΔZ: Impedance change

METHOD

The raw plethysmographic signal (with amplitude of 0 to $\pm$ 500 μV and signal bandwidth of 0 to 25 Hz) was input to the DSP. Our plethysmograph acquired physiological data at 5 kHz sampling rate was subject to down-sampling to 1 kHz before applying DSP. The overall algorithm is shown in Figure 1. Certain subsections of the algorithm are outlined separately. Various steps of the main algorithm are keyed to entries in the description below. To keep the description simple, we assume in the discussion below that the data were acquired for a total of 6 s (that is, 6000 samples at 1 kHz sampling rate). Our algorithm automatically adapts to shorter data collection periods; however, for very short data collection periods, of less than 3 s, the algorithm invokes additional steps. Those details are not included here.

We used the following parameter values, which can be adapted to different protocols as required: The beat-to-beat period could vary ±30% with respect to each other in a given measurement; the absolute beat-to-beat period could vary over 400 ms to 1000 ms; the maximum slopes of the 3 pulses to be picked for averaging could differ by 20% (relaxed to 30% with noisy data or when 3 or fewer pulses are recorded) in their maximum slope values; their foot-to-peak amplitudes had to be within ±15% of each other; and the absolute pulse amplitude had to be ≤ 500 μl/cm.

Main Algorithm: See Figure 1.

1. First, subject the data to an anti-aliasing filter to significantly attenuate the 60 Hz ac noise and the high frequency carriers (the acquired signal amplitude modulated the carrier up to a maximum of about 30%). This was a fourth-order Butterworth filter with an attenuation of 31 dB at 62.5 Hz and a cutoff frequency of 25 Hz [Lyons, 2001]
2. Set Iteration count to one. This is useful for iteratively improving the final list of Foot Point and End Point arrays.
3. Store filtered data in a separate ('FILTERED DATA') Array. This ensures integrity and storage of the original data. All the ensuing steps, until otherwise noted, will use a working copy of this filtered data. This copy will be modified by the ensuing steps.
4. Detrend Filtered data. This is a moving-average high-pass filter with a 3 dB cutoff at 0.155 Hz. The triangular or Bartlett window used a window length of 749 points. Note: This detrending affects half the length of the window (375 points or 375 ms) at both ends of the entire signal (6000 ms). Signal in those regions is discarded in a later stage so signal averaging is not impacted.

5. Differentiate using the three-point difference method, to locate the local minimum, maximum, and deflection points [Shoup, 1983]:

   $Y'_0 = (1/2h) * (-Y_2 + 4Y_1 - 3Y_0)$ (initial point)
   $Y'_n = (1/2h) * (3Y_n - 4Y_{n-1} + Y_{n-2})$ (end point)
   $Y'_k = (1/2h) * (Y_{k+1} - Y_{k-1})$ (for $k = 1,2,...n-1$),

   where h is the distance between two consecutive points in ms, n the maximum number of data points, Y_i is the i^{th} sample point and Y_i' is the differential at that point. Due to the instrumentation and measurement procedure, the signal was always corrupted with noise which on occasion had high-frequency components. Differentiation was performed by considering only every eighth point of the signal (h = 8 ms). This effectively meant that the signal was down-sampled from a sampling frequency of 1 kHz to 125 Hz, so the maximum frequency component in the signal would have to be 62.6 Hz or less. Because of noise, this was not always the case. The frequency wrap around phenomenon occurred and the higher noise frequencies wrapped around and added to the signal frequency components. This was avoided in this implementation by the anti-aliasing filter introduced in step 1 above.
6. Filter differentiated data with a fourth order Butterworth filter with a cutoff frequency of 25 Hz, similar to the one mentioned earlier. This was needed since the differentiated data still contained some noise and the determination of MSP would be far easier without noisy peaks.
7. Find PZC (positive zero crossing) and NZC (negative zero crossing) points from the differentiated and filtered data. These provide the local minimum and maximum.
8. Remove PZCs and NZCs lying in the starting and ending regions of the sampled data, equal to ½ the window length used in the detrending step above. This equals 375 data points at 1 ms interval on both sides of the 6000 data.
9. Determine the MSP (maximum slope point) between a pair of PZC and NZC. Call these PZC and NZC arrays as FINAL ARRAYs. Use these FINAL ARRAYs in the following steps.
10. Find Valid MSPs. Reject invalid PZCs. Valid PZCs will be identified later as Foot Points of potential pulses chosen for averaging. This step (10) calls the 'Adaptive Threshold' algorithm. This algorithm will accomplish the following: (a) Start with a preset width threshold (time interval) value and automatically determine the most appropriate time window to find valid maximum slope points (MSP). The algorithm uses three width thresholds (time intervals) of 1000, 700, and 400 ms, which represent the appropriate range for human heart beat-to-beat period. (b) Remove MSPs due to high-frequency carrier and instrumentation induced high-frequency jumps due to ADC (analog-to-digital conversion). These may otherwise cause spurious PZCs. (c) Find a data-dependent earliest time instant to search from for an End Point (a near neighbor in the Foot Point array) relative to the given Foot Point, once a valid Foot Point array has been determined. There will always be a one-to-one correspondence between the Foot Point and End Point Arrays. A subject who goes to sleep or breathes improperly may cause

consecutive pulses to appear with substantially different MSPs, enough for every alternate MSP and Foot Point to be rejected. In such cases, alternate pulses may be chosen as valid ones. The result at the end of the algorithm will be an average that depicts two pulses. The algorithm finds the data-dependent earliest instant ('minimum threshold'), to search for an End Point from each valid Foot Point.

11. If there are any valid MSPs, go to step 12; otherwise, exit the analysis, since the data are not amenable to automatic processing and may be bad. Go to 30.
12. Determine valid Foot Points and End Points. A subset of PZCs will be identified as Foot Points by the end of the Adaptive Threshold Algorithm. Now, determine the End Points of the pulses to be chosen for averaging. This, in turn, may impact the Foot Point Array. The 'Robust End Point' algorithm details the steps. It enhances the robustness of End Point determination by searching for an end point starting at 200 ms less than the 'minimum threshold' set in step 10.
13. This is the start of a 'Hit-or Miss' matrix to identify quality of the data. If there are more than 2 pulses (two entries in the Foot Point and End Point arrays), go to step 14. If not skip to Step 15.
14. Remove pulses from the arrays that violate the following interpulse comparison rule: The time-width of the pulses chosen for averaging should be within ±30% of each other. Note: There are arrays for MSP, PZC, NZC, Foot Points, and End Points. We label them as FINAL ARRAYs. The time-width is computed from the time difference between respective Foot Point and End Point entries.
15. If there are no pulses left after the previous step, quit. No good data to analyze. Go to 30.
16. If not, check whether it is the first Iteration. Note: This 'Hit or Miss' matrix is used three times to refine the Foot Point and End Point arrays.
17. If it is the first Iteration, store a copy of the FINAL ARRAYs in WORKING ARRAYs. Use WORKING ARRAYs in the steps below.
18. Remove pulses from the WORKING ARRAYs with pulse widths of less than 400 ms.
19. Increment the ITERATION count by 1
20. If there are at least 3 good pulses left after this, proceed to step 25. If not, go to step 21 to better incorporate the information of pulses removed in step 18. Note that the next two steps work in the opposite direction. Given bad pulses, they remove the PZC and NZC from their respective arrays.
21. Store the Foot Points and End Points of the discarded pulses (in step 18) in separate arrays.
22. Remove corresponding PZC and NZC from their respective arrays, so the discarded pulses are no longer represented in the PZC and NZC FINAL ARRAYs.
23. If the number of ITERATIONs is 3 or less, repeat most of the process again with the new FINAL ARRAYs for PZC and NZC. Go to step 9. Otherwise, go to step 24.
24. Use the FINAL ARRAYs. Three Iterations have been completed. No improvement is possible.
25. Project the individual pulses to the time-axis. This removes any DC drift in the signal. Use the FILTERED DATA for this purpose.

26. If the number of pulses chosen is 3 or more, go to the next “Hit or Miss” matrix interpulse comparison. If not, go to step to 28
27. Remove pulses from the arrays that violate the following inter-pulse comparison rule: The amplitudes of the pulses chosen for averaging should be within ±15 % of each other. The pulse amplitude is computed by determining the signal amplitudes that correspond to the PZC and NZC entries, for valid Foot Point entries. The signal amplitudes are available in FILTERED DATA
28. If there are no pulses left, end the processing. The data is not amendable to automated processing. Go to 30
29. If not, average the good pulses. Determine the corresponding volume change in the limb segment as per the Nyboer’s equation. This is the normal end.
30. Data are not amenable to automated processing. This is the abnormal end.

Adaptive Threshold Algorithm: We briefly outline it here as a list of Stages.

1. Sort on MSP values: In this step, all the MSPs are arranged in descending order of slope values. Next find successive MSPs that are within 20% of the previous MSP. If the number of such MSPs is 3 or more, go to Stage 2. If not, relax the threshold to 30% and repeat. If the resulting number of MSPs is ≤ 2, they are due to noise. Remove them from the list and continue till the end of the MSP array. If no MSPs are left, end the processing and return to the main algorithm.
2. Sort on inter-MSP intervals: Arrange the chosen MSPs in time order. Set the width threshold (time interval) to 1000 ms. If two consecutive MSPs (in time) are separated by a width of at least the width threshold, accept the first one as a valid MSP. If not, store the first one in a reject array. If the difference between the number of chosen MSPs and rejected MSPs is ≥ 3, go to Stage 4. If not, go to Stage 3.
3. Update width threshold: Reduce the width threshold by 300 ms. If the width threshold is ≥ 400 ms, go to Stage 2. If not, end the processing and return to the main algorithm.
4. Remove MSPs due to high frequency (hf) noise: Choice of End Point is very sensitive to hf noise. The End Point corresponds to a PZC, and each PZC has a corresponding MSP. Due to the LPF used in the main algorithm, any MSP due to hf noise will have low amplitude compared to a good MSP (generated by a physiological pulse). Arrange the MSPs in descending order of slope values. Check whether there are MSPs which are lower than 50% of the first (or highest) MSP value. Remove such MSPs from the list as being due to hf noise. Remove the Foot Points corresponding to these removed/rejected MSPs and place them a reject array. Go to Stage 5.
5. Ensure Single Pulse Average: This is occasioned by a subject going to sleep or breathing improperly. Consecutive pulses may then appear with different slopes. In such cases, alternate pulses may be chosen as valid points. For example, if the heartbeat period is 750 ms, then alternate pulses chosen for averaging are separated by 1500 ms. Thus, a width threshold of 1000 ms may give rise to at least three signal strips with two pulses for averaging. Looking for an end point beyond 800 ms (i.e., width threshold – 200 ms = 1000 – 200 ms = 800 ms) will fail to pick up the actual end point that was around 750 ms, resulting in a double

pulse as the average. To avoid this, we chose to define a minimum duration from the Foot Point to search for the End Point: Determine the width between each valid MSP and the next immediate MSP not in the reject array. Find the minimum of all the widths so obtained. If this minimum width is ≤ (width threshold – 200) ms, set the width threshold to this minimum width. Return to the Main Algorithm.

Robust End Point Algorithm: This algorithm enhances the robustness of endpoint determination by searching for an End Point starting at 200 ms less than the width threshold. Here are the steps:

1. Look for a PZC immediately preceding each chosen MSP. This corresponds to the Foot Point of each chosen pulse.
2. Look ahead from this Foot Point for a PZC that is separated from the Foot Point for that pulse by at least (width threshold – 200 ms).
3. Determine whether the PZC is contained in the array of Rejected Foot Points. If so, ignore the corresponding PZC and return to step 2. If not, choose the PZC as an End Point for that pulse. Return to the Main Algorithm.

RESULTS:

Algorithmic Verification: The algorithm described above was tested before applying it to real data analysis. To this end, we generated signals which closely resemble the real data and use this algorithm to process them. This test signal was corrupted with random noise having signal-to-noise ratios of 2, 4 and 8. Each aggregate signal was processed with the algorithm described above.

The result of the analysis was one average pulse, the termination point of the Main Algorithm (step 29). We determined that noise does not lead to a bad choice of End Point or incorrect foot-to-peak value of the average, thus showing the effectiveness of the algorithm.

Human Data Analysis: We have successfully processed data on the first 30 subjects in the clinical validation study at the Bowman Gray School of Medicine, and 9 subjects in a repeatability study at our lab. Figures 2 and 3 present average pulses obtained from the same subject on two different visits. Briefly, we used a pressure cuff on subject dt on his left arm (la), left thigh (lt), and left calf (lc) and recorded 3 to 6 s of plethysmographic data at each cuff pressure decrement from about 150 to 20 mm Hg. This was repeated twice during the same visit. Each of the plots in the two figures present one averaged volume pulse plotted against time, at different cuff pressures. One may note slight differences in the maximum volume change signals obtained, especially on the thigh segment. This may represent both our inability to wrap a thigh cuff at the exact site and physiological variation. However, on the DSP side, there is no evidence of high noise corruption of the data, instrumentation induced resets in the data, and double pulses in the average. One may also note that the averaged pulses across the limb segments and across visits track each other well.

DISCUSSION:

Simultaneous recordings on two or more limb segments are feasible with our multichannel instrumentation [Shankar, Shao, and Webster, 2008]. Such simultaneous recordings can be used to advantage in DSP for cross referencing so noncompatible pulses can be rejected based on their time instances and periods. Multiple recordings on the same limb site (in our case, at different cuff pressures) also give patient-specific ranges of pulse periods, which can be exploited to enhance the reliability of the automated analysis. We have incorporated such algorithms in our DSP.

We use human experts, after DSP, to interpret the validity of the average pulse obtained by automation. It is our goal to automate even this expert interpretation. We believe the following two rules are the most relevant:

Rule 1: If two or more pulses appear in the averaged waveform, or the waveform's foot point does not start near zero, convert the validity flag in the *.fla* file to "bad". If possible, estimate the foot-to-peak amplitude from the raw signal and set the three validity flags to "good" if appropriate.

Rule 2: If the heart rate (HR) is 1/2 or 1/3 of the average value, the signal might still be good. This could happen because the one- and two-pulse periods fell within the admissible range of the DSP algorithms. If the HR is not within ±10% of 1/2 or 1/3 of the average, then the signal is "bad".

CONCLUSION:

A set of highly-sophisticated, reliable and robust algorithms have been developed to process the data acquired in the clinical validation study on the early detection of atherosclerosis using a PC-based three-channel impedance plethysmograph. Algorithms were evolved and refined over a period of six years, based on a longitudinal study on monkeys and our repeatability studies on human subjects. The processing is almost totally automated, but for the expert analysis. The latter can be incorporated easily, once all the expert rules have been formulated. We project that real-time DSP in an embedded microcontroller system could perform all the DSP algorithms in less than 10 min.

REFERENCES

Lyons, R.G., Understanding Digital Signal Processing, Prentice Hall, Upper Saddle River, NJ, 2001.

Shoup, T.E., Numerical Methods For the Personal Computer, Prentice Hall, Englewood Cliffs, NJ, 1983

Shankar, R., Shao, S.Y., and Webster, J.G., A Fully Automated Multichannel Digital Electrical Impedance Plethysmograph, In preparation, 2008.

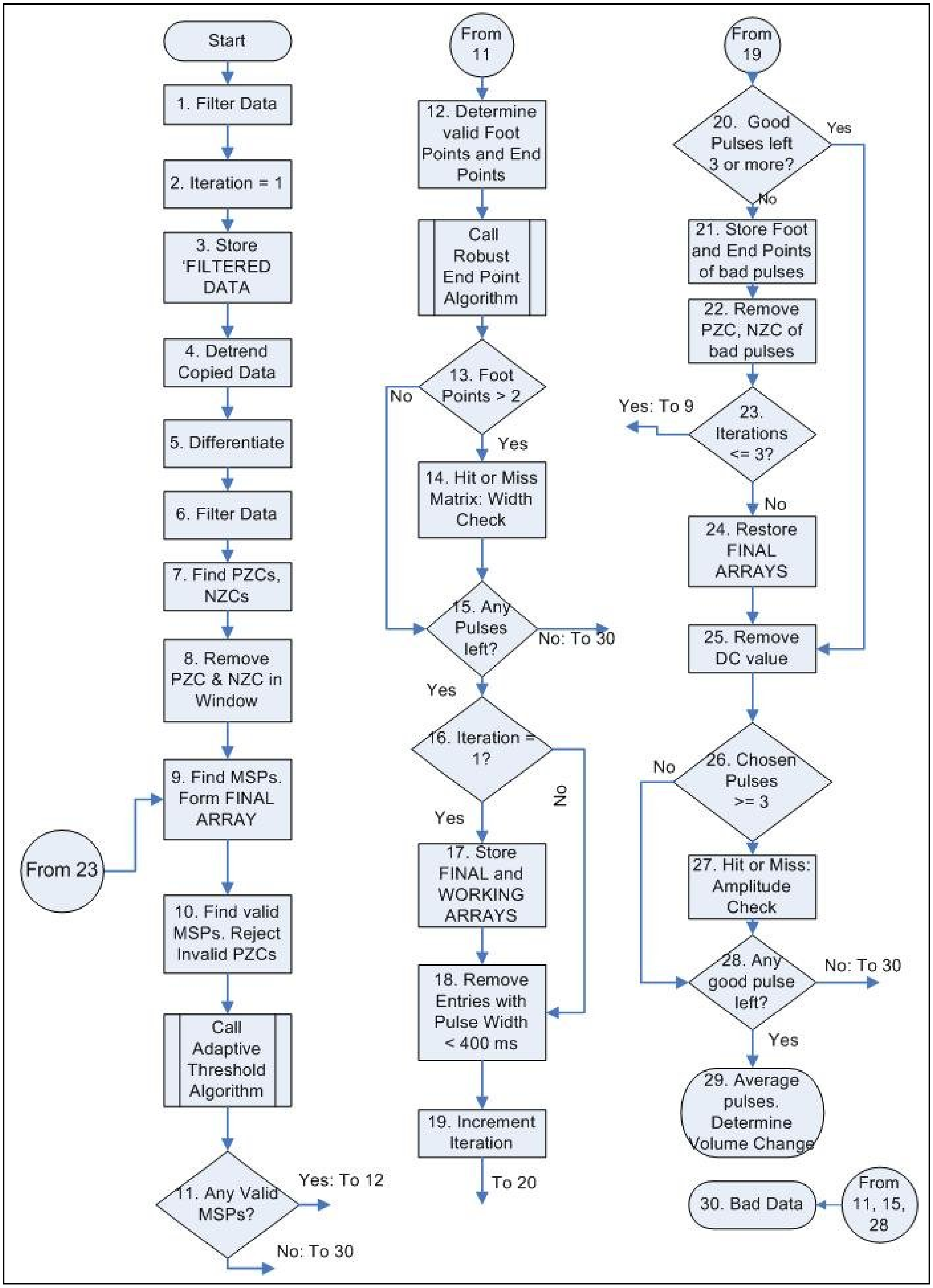
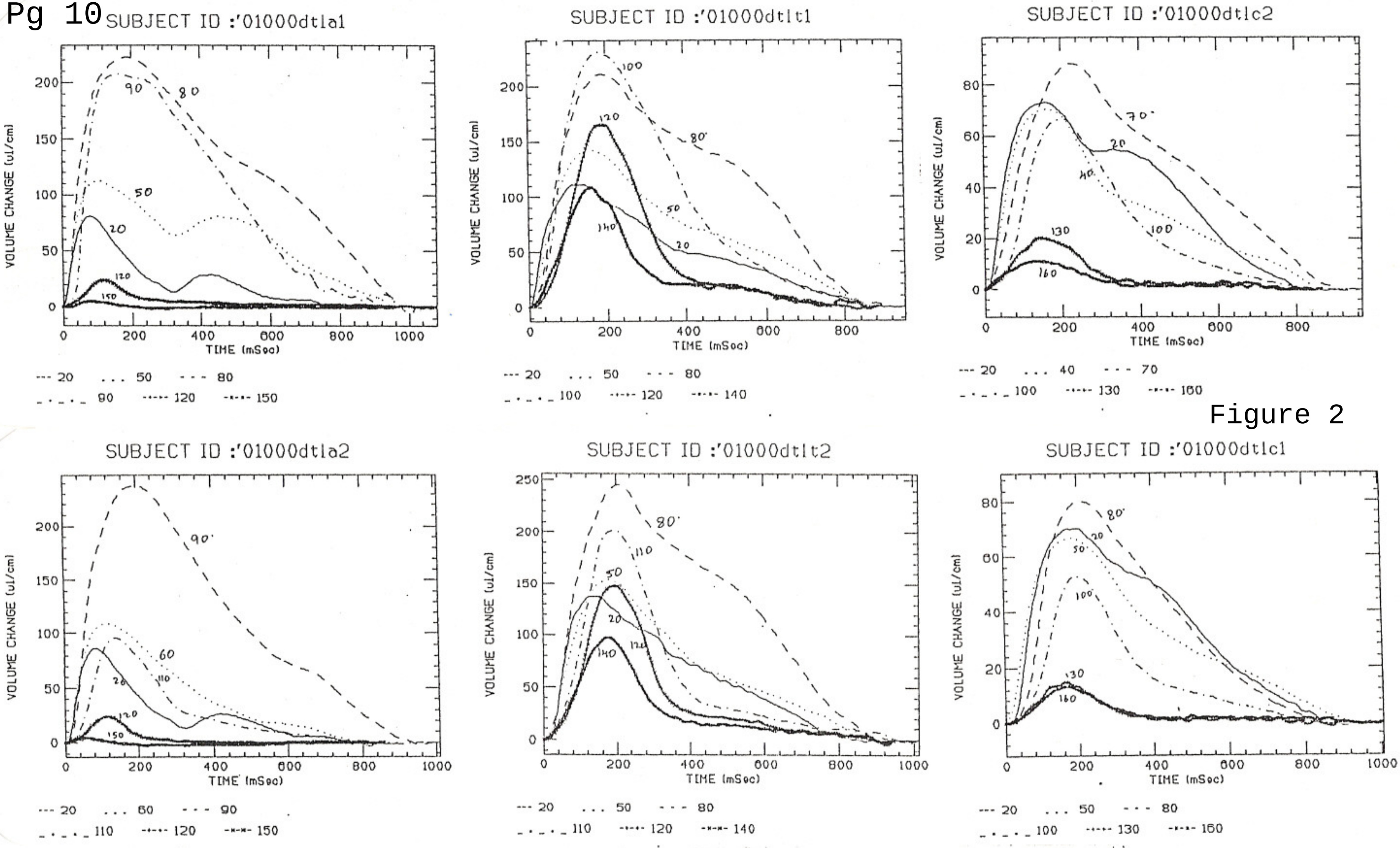

Start
1. Filter Data
2. Iteration = 1
3. Store 'FILTERED DATA
4. Detrend Copied Data
5. Differentiate
6. Filter Data
7. Find PZCs, NZCs
8. Remove PZC & NZC in Window
9. Find MSPs. Form FINAL ARRAY
From 23
10. Find valid MSPs. Reject Invalid PZCs
Call Adaptive Threshold Algorithm
11. Any Valid MSPs?
Yes: To 12
No: To 30
From 11
12. Determine valid Foot Points and End Points
Call Robust End Point Algorithm
13. Foot Points > 2
No
Yes
14. Hit or Miss Matrix: Width Check
15. Any Pulses left?
No: To 30
Yes
16. Iteration = 1?
No
Yes
17. Store FINAL and WORKING ARRAYS
18. Remove Entries with Pulse Width < 400 ms
19. Increment Iteration
To 20
From 19
20. Good Pulses left 3 or more?
Yes
No
21. Store Foot and End Points of bad pulses
22. Remove PZC, NZC of bad pulses
23. Iterations <= 3?
Yes: To 9
No
24. Restore FINAL ARRAYS
25. Remove DC value
26. Chosen Pulses >= 3
No
27. Hit or Miss: Amplitude Check
28. Any good pulse left?
No: To 30
Yes
29. Average pulses. Determine Volume Change
30. Bad Data
From 11, 15, 28

Figure 1: Main Algorithm



SUBJECT ID :'01000dtla1

SUBJECT ID :'01000dtlt1

SUBJECT ID :'01000dtlc2

Figure 2

SUBJECT ID :'01000dtla2

SUBJECT ID :'01000dtlt2

SUBJECT ID :'01000dtlc1

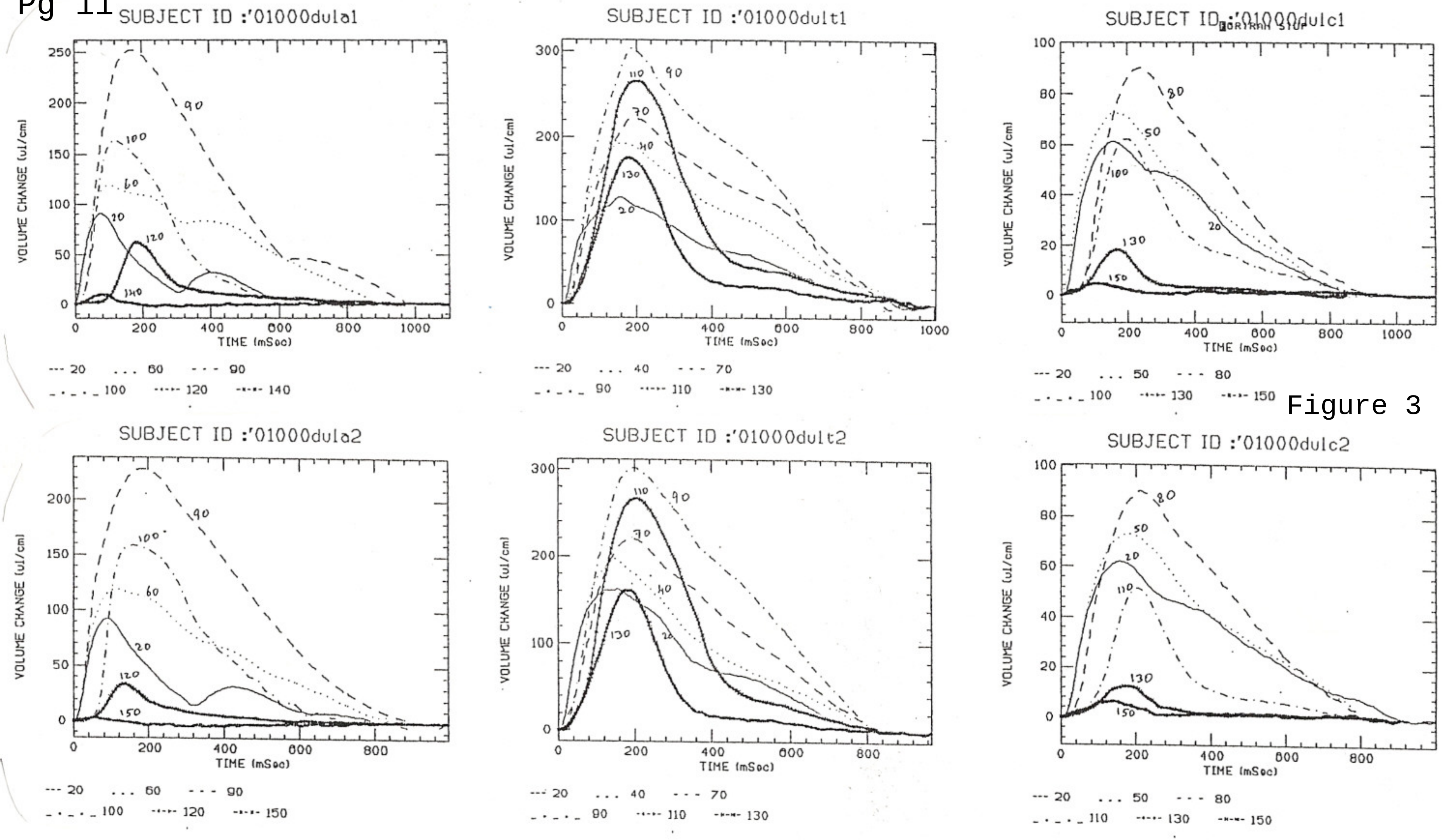

SUBJECT ID :'01000dula1
VOLUME CHANGE (ul/cm)
TIME (mSec)
90
100
60
20
120
140
--- 20 ... 60 - - - 90
_ . _ . _ 100 ----- 120 -*-*- 140
SUBJECT ID :'01000dult1
110
90
70
40
130
20
--- 20 ... 40 - - - 70
_ . _ . _ 90 ----- 110 -*-*- 130
SUBJECT ID :'01000dulc1
80
50
100
20
130
150
--- 20 ... 50 - - - 80
_ . _ . _ 100 ----- 130 -*-*- 150
SUBJECT ID :'01000dula2
90
100
60
20
120
150
--- 20 ... 60 - - - 90
_ . _ . _ 100 ----- 120 -*-*- 150
SUBJECT ID :'01000dult2
110
90
70
40
130
20
--- 20 ... 40 - - - 70
_ . _ . _ 90 ----- 110 -*-*- 130
SUBJECT ID :'01000dulc2
80
50
20
110
130
150
--- 20 ... 50 - - - 80
_ . _ . _ 110 ----- 130 -*-*- 150

Figure 3