Diagnostic Alarms in Anaesthesia

Bhupendra Gohil

Thesis submitted in partial fulfilment of the degree of Master of Engineering

AUT University Auckland

February 2007

This thesis contains confidential material. The thesis shall not be used, copied, given or conveyed to anyone who is not directly involved in the examination of this work.

Acknowledgements

I would like to extend my gratitude to my supervisors and mentors, Prof. Hamid Gholam Hossieni and Prof. Michael Harrison who gave me the opportunity to work in this particular field of research and for patiently reviewing my thesis. Both of my supervisors have been very approachable, considerate and have always expressed a keen attention to the problems I encountered during the course of my research.

I seize this opportunity to thank Prof. Ahmed Al-Jumaily for his guidance and excellent support from my colleagues at BIOMEC, AUT. I am grateful to Dr. Andrew Lowe for timely support, guidance and for providing technical expertise with his DOMonitor application which smoothed the entire data collection and testing phase of the project. I would like to thank Dr. Khee Lim and librarian Emma McFadgen for supporting and encouraging my writing, and for challenging me to strive for excellence.

My deepest thanks go to my family who have been very supportive and always dedicated to my academic success.

Abstract

Smart computer algorithms and signal processing techniques have led to rapid development in the field of patient monitoring. Accelerated growth in the field of medical science has made data analysis more demanding and thus the complexity of decision-making procedures. Anaesthetists working in the operating theatre are responsible for carrying out a multitude of tasks which requires constant vigilance and thus a need for a smart decision support system has arisen. It is anticipated that such an automated decision support tool, capable of detecting pathological events can enhance the anaesthetist's performance by providing the diagnostic information to the anaesthetist in an interactive and ergonomic display format.

The main goal of this research was to develop a clinically useful diagnostic alarm system prototype for monitoring pathological events during anaesthesia. Several intelligent techniques, fuzzy logic, artificial neural networks, a probabilistic alarms and logistic regression were explored for developing the optimum diagnostic modules in detecting these events. New real-time diagnostic algorithms were developed and implemented in the form of a prototype system called real time – smart alarms for anaesthesia monitoring (RT-SAAM). Three diagnostic modules based on, fuzzy logic (Fuzzy Module), probabilistic alarms (Probabilistic Module) and respiration induced systolic pressure variations (SPV Module) were developed using MATLAB^{TM} and LabVIEW^{TM}. In addition, a new data collection protocol was developed for acquiring data from the existing S/5 Datex-Ohmeda anaesthesia monitor in the operating theatre without disturbing the original setup.

The raw physiological patient data acquired from the S/5 monitor were filtered, pre-processed and analysed for detecting anaesthesia related events like absolute hypovolemia (AHV) and fall in cardiac output (FCO) using SAAM. The accuracy of diagnoses generated by SAAM was validated by comparing its diagnostic information with the one provided by the anaesthetist for each patient. Kappa-analysis was used for measuring the level of agreement between the anaesthetist's and RT-SAAM's diagnoses.

In retrospective (offline) analysis, RT-SAAM that was tested with data from 18 patients gave an overall agreement level of 81% (which implies substantial agreement between SAAM and anaesthetist). RT-SAAM was further tested in real-time with 6-patients giving an agreement level of 71% (which implies fair level of agreement). More real-time tests are required to complete the real-time validation and development of RT-SAAM.

This diagnostic alarm system prototype (RT-SAAM) has shown that evidence based expert diagnostic systems can accurately diagnose AHV and FCO events in anaesthetized patients and can be useful in providing decision support to the anaesthetists.

Table of Contents

List of Figures

Statement of Originality

‘I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning, except where due acknowledgment is made in the acknowledgments.’

Signed: ……………………

Date: …………………….

Intellectual Property Rights

This thesis contains confidential material. This thesis shall not be given to anyone who is not directly involved in the examination of the thesis. The MATLAB^{TM} and LabVIEW^{TM} pseudo code in the appendix section and the accompanying CD-ROM is property of the researcher and co-researchers. For commercial reasons the program code in this thesis needs to be kept confidential for a period of 3 years from the date of publication of this thesis.

List of Abbreviations

BP/P1	Invasive blood pressure
SAP	Systolic arterial pressure
MAP	Mean arterial pressure
HR	Heart rate
SV	Stroke Volume
P2	Central venous pressure
PV	Pulse volume / Plethysmography
$ETCO_2$	End tidal (exhaled) carbon-dioxide
ICO_2	Inspired (inhaled) carbon-dioxide
SPV	Respiration-induced systolic pressure variations
AHV	Absolute hypovolaemia
RHV	Relative hypovolaemia
FCO	Fall in cardiac output
MH	Malignant hyperpyrexia
ANN	Artificial Neural Networks
FFT	Fast Fourier transform
SAAM	Smart alarms for anaesthesia monitoring
RT-SAAM	Real-time smart alarms for anaesthesia monitoring
mmHg	Millimetres of mercury
S/5	Datex-Ohmeda, GE S/5 anaesthesia monitor
IDAS	Injectable drug administration and automation
OOP	Object Oriented Programming

Chapter 1 Introduction

1.1. Background

Last two decades have witnessed a tremendous growth in the field of biomedical engineering. Another domain of interest is intelligent computer systems which have been successfully applied in various applications from aircraft controllers [1] to automated control processes in many industrial applications [2, 3] and smart patient monitoring systems. These intelligent systems are also referred as “Expert Systems” because they mimic the operation of the expert human operator in their respective fields. Expert Systems can be used to provide decision support to human operator by integrating and displaying information in an easy to grasp format and thereby improve the operational performance of the operator [4, 5].

During a routine surgery an anaesthetist is in charge of maintaining patient’s physiological state in sound condition through vigilant analysis of multiple channels of patient data [6]. Deneault [7] compared anaesthetists’ task to the complex decision making process executed by an aircraft controller or a nuclear plant operator. Patient management and monitoring are some of the key tasks that an anaesthetist performs during routine surgical procedures in the operation theatre. Thus an anaesthetist is required to maintain very high level of vigilance at all times, analysing patient information from a vast array of monitoring equipments present in the operation theatre and make appropriate decisions. The anaesthetists’ task is further complicated by the diversity of the display formats from the monitoring equipments manufactured by different vendors [7]. Anaesthetists are often overloaded with data from the monitoring equipment and thus lead to anaesthesia related mishaps. Human errors contribute to a large portion of the anaesthesia related mishaps and they can be easily prevented by providing decision support to the anaesthetists [8, 9].

Many expert decision support systems have been proposed in the past to enhance the anaesthetists’ performance [5, 10]. Decision support systems can provide diagnostic information for critical events during anaesthesia, thus aiding the anaesthetist and, in some cases, even outperforming the anaesthetist. For instance in retrospective analysis an expert system called “SENTINEL” [5, 11] diagnosed the onset of malignant hyperpyrexia about 10 minutes before the anaesthetist. Numerous techniques have been proposed in the past and some were employed for developing diagnostic alarm systems [12-16]. Most of the expert diagnostic systems for detecting anaesthesia related events were designed for offline/retrospective analysis [6, 11, 17, 18]. This had prompted the development of a diagnostic alarm prototype of clinical significance which could provide decision support to the anaesthetists and tested in real-time. Having redundant (extra) diagnostic information provides more concrete evidence and thus increases the confidence in one particular diagnosis [15]. Hence it was proposed that the diagnostic alarm prototype developed in this research project should have multiple diagnostic modules so that redundant diagnosis could be generated for each critical event. During this research, several different techniques were studied through intensive literature review and the most suitable techniques were selected for developing new algorithms which were implemented for generating diagnoses.

1.2. Motivation

Digital signal processing is a rapidly expanding field and has numerous real-life applications ranging from medical imaging and patient monitoring [19-22] to telecommunication-based applications [23, 24]. In this research project, new signal processing scripts were required for filtering and analysing the raw patient data before generating diagnostic information from the analysed data. The development of the proposed diagnostic system prototype was a practical application of the signal processing techniques and has provided a challenging research project, on part of a signal processing engineer.

Knowledgeable project mentors have provided the researcher a good understanding in the field of signal processing and anaesthesiology. This project was a brain child of external project supervisor Dr. Michael Harrison, Anaesthetist, Auckland City Hospital. Dr. Harrison had worked on several projects on anaesthesia alarm development in the past and has strong knowledge in the field of anaesthesia.

The motivation for this research project was the development of a new computer based prototype which could improve patient safety by reducing human error. The main objective of this research was to develop a diagnostic alarm prototype capable of generating diagnosis for anaesthesia related critical events and present the diagnosis to the anaesthetist on an interactive and ergonomic user interface on the computer screen.

1.3. Literature Review

With the accelerated growth in the field of medical science it became evident that experts had to cope with large volumes of data and decision making processes [25]. Thus the need for expert systems in assisting the decision making task became even more apparent. Technological advances in clinical science and engineering had given birth to a whole new era of computer assisted systems called “smart/expert systems”. Research into computer assisted decision making tools began as early as 1950s [25]. Expert systems are capable of mimicking the performance of a human operator by using a computer based decision making algorithm. Expert systems have been widely applied in decision making processes for various medical applications [26-30] as well as non-medical applications [31-33]. In this research project we focus our concentration on expert systems applied to anaesthesia and their applications for monitoring pathological conditions during the administration of anaesthesia.

1.3.1. Expert Systems in Anaesthesia Monitoring

Anaesthesia is one of the most demanding branches of medical science. During routine surgical procedures in the operating theatre, anaesthetists have to execute several critical tasks, some of which are listed below:

1) Analyse large amount of data presented by various monitoring equipments [8].
2) If a problem is detected, it must be diagnosed as a matter of urgency.
3) Execute various patient management protocols [34].

In certain circumstances anaesthetists may be overloaded by patient data from the monitoring equipments and the analysis of these data. The anaesthetists could potentially be too occupied to diagnose the patient’s condition [35] and this could lead to potentially grave consequences [11]. The common anaesthesia mishaps that occur during day-to-day anaesthesia causing these disastrous errors in anaesthesia administration were classified and studied by Reason [36]. Human errors in anaesthesia account for 82% of the preventable mishaps [9]. The raw information given to the anaesthetist can be analysed and presented to anaesthetist in an ergonomic form which helps to reduce data overloading and hence the associated errors during anaesthesia administration. It was suggested by van den Eijkel and co-workers [8] that such conditions can be overcame by using a knowledge based anaesthesia monitor (Expert System). Evidence/knowledge based anaesthesia monitor which analyses the data and presents the results to the anaesthetist can help the anaesthetist to carry out effective diagnosis and hence prevent such mishaps.

Another approach towards aiding the anaesthetists' task include providing the information to the anaesthetist in an integrative display format which reduces the response time of the anaesthetist in detecting pathological events [7]. Thus it can be inferred that, an expert anaesthesia monitor should analyse the patient data from the monitoring equipment in the operating theatre and present the diagnosis to the anaesthetists in an explicit way on an integrative, ergonomic user interface.

The existing anaesthesia monitors in the operating theatre are capable of monitoring raw physiological data with no diagnostic alarm capabilities. Present anaesthesia monitoring systems are capable of monitoring numerous physiological parameters like:

1) Blood pressure (BP)
2) Plethysmography/Pulse volume (PV)
3) End tidal (exhaled) carbon dioxide ($ETCO_2$)
4) Inspired (inhaled) carbon dioxide (ICO_2)
5) Pulse oximetry (oxygen saturation level of haemoglobin in blood) (SPO_2), etc.

The majority of expert systems have been developed for offline/retrospective analysis of clinical data and hence cannot be tested in real-time [5, 6, 37, 38]. Some recent works [11, 13, 37, 39, 40] in the field have given way to new online monitoring techniques like fuzzy logic based diagnosis, artificial neural network based diagnosis, logistic regression based trend detection and statistical/probability based techniques.

Sukuvaara et al. [39] developed an online expert system that acquires data from an anaesthesia monitor and analyses the acquired data to give two level (alarm and alert) warnings to the anaesthetist. This alarm detects hypovolaemic, hyperdynamic and hyperventilation states in postoperative cardiac surgical patients. Another monitoring algorithm developed by Krol et al. [37] detects low anaesthesia level and unstable blood pressure (lability).

However, it has been established that overlapping parameters can improve the accuracy of the diagnosis and by giving more weight to a closely related parameter can further enhance the correct diagnosis. For instance the diagnosis of a fall in cardiac output (FCO) is likely to be more accurate if $ETCO_2$ is given more weight than the other parameters in the rule base used for detection of CO [11]. Anaesthetists make decisions in complex circumstances by monitoring various parameters like BP, SpO_2, $ETCO_2$ and reacting to an abnormal stimulus (abnormal patterns in the signals being monitored) [8]. The response of the anaesthetists is based on the available clinical evidence. Anaesthetist execute the decision making task by forming a mental map of the available evidence and identifying the disease patterns (stimulus) in the data signals. For example use of the central venous pressure (CVP) monitoring during the preanaesthetic period is essential to the anaesthetist for evaluation of the patients state who has been suffering massive blood loss[41]. Quantitative measures like thermodilution CO measurement by pulmonary artery catheterisation [42, 43] and central venous pressure can also be useful for monitoring CO and FCO but might not be always available, for example in patients undergoing non-cardiac surgeries.

An expert diagnostic system mimics an expert's behaviour by executing a series of smart/intelligent algorithms which scan the available data for stimulus. The expert knowledge implemented by anaesthetists for making the diagnosis exists in the form of linguistic rules. These linguistic rules are required to be converted into programmable sets of rules for the development of smart computer algorithms. Fuzzy logic based algorithms and probabilistic alarm algorithms discussed in the following sections have the potential for implementing these linguistic rules into logical algorithms.

By using a fuzzy logic based algorithm, expert diagnostic systems can be developed to match or exceed the performance of the anaesthetists as discussed by Grant and Naesh [44]. These authors have also pointed out some applications for control of depth of anaesthesia where decision making algorithms have outperformed the anaesthetist. It was also suggested that the expert diagnostic systems are more reliable than manual interventions. The SENTINEL system developed by Lowe [11] is a prime example of how fuzzy logic and knowledge based systems can be developed for detection of adverse condition during anaesthesia administration.

Expert systems based on fuzzy logic and artificial neural networks (ANN) can also be used for detecting rare pathological conditions like malignant hyperpyrexia (MH). For instance a fuzzy logic based algorithm was developed by Lowe and Harrison [5] for detecting MH. In offline validation of the algorithm, the system detected MH nine minutes before the anaesthetist diagnosed it. The work done by Lowe and Harrison exemplifies how expert systems can be implemented to facilitate and enhance anaesthetists' performance in the clinical environment and thus improving patient safety. The algorithms developed by Lowe and Harrison (SENTINEL) are offline test algorithms and therefore were tested retrospectively. These algorithms need to be redeveloped as real-time test modules so that they can be used for real-time diagnostics during a clinical procedure. The evolution of relevant technologies and innovations has generated a whole new breed of intelligent algorithms and techniques. The following literature discusses various computer based techniques which can be used for implementing an expert diagnostic alarm system.

1.3.1.1. **Fuzzy and Crisp Sets**

Sets are ubiquitous in all logic based systems. Sets theory was proposed by Cantor [45] in late 1890s. Sets are defined as logical groups of abstract elements. For example consider the linguistic diagnostic rule for absolute hypovolaemia (AHV) defined by the anaesthetist which can be expressed in the form of following logical statement.

> *"If blood pressure is very low AND pulse volume is very low then absolute hypovolaemia is very likely."*

Now using the classical crisp set theory we can define a low blood pressure set (LBP) and low pulse volume set (LPV) as in equations 1.1 and 1.2 respectively, see the graphical illustration. Further crisp set operators like AND, OR, NOR, etc. operator can be used for combining two or more sets to create more complex sets (refer equation 1.3), this is graphically illustrated in the following Venn diagram Figure 1.3.1-1.

$$\textbf{Set LBP} = \{X \in BP\_dataset \mid 95mmHg \geq X\} \qquad (1.1)$$

$$\textbf{Set LPV} = \{X \in PV\_dataset \mid 3Litres \geq X\} \qquad (1.2)$$

Set AHV = Set LBP AND Set LPV (1.3)

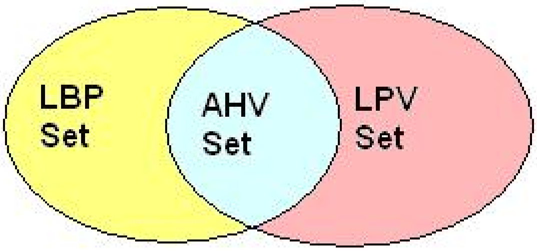


Figure 1.3.1-1. Venn diagram for AHV.

Fuzzy sets can be used for representing information where the boundaries of the set are not clearly defined [46]. In the above diagnostic rule, blood pressure (BP) of 94.9999 mmHg will be classified as either high BP (1 or ON) or low BP (0 or OFF) depending upon what limiting condition is set for the crisp set (refer Figure 1.3.1-2). However it is very likely that an expert would classify BP values of 94.9999 mmHg and 95.0001 mmHg as high BP when the threshold is set at 95 mmHg. Thus the vagueness and uncertainty in linguistic rules cannot be clearly represented if the BP is expressed using a crisp set. For instance very high BP, very low BP, moderately high BP cannot have fractional/partial membership to the crisp set.

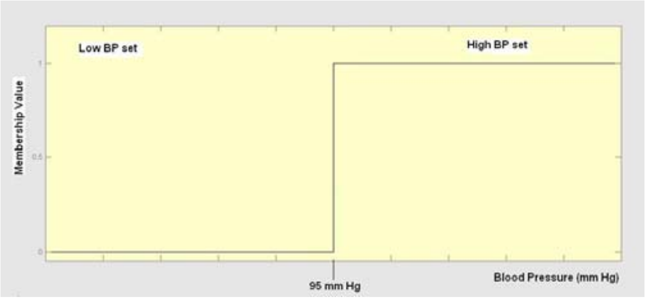


Figure 1.3.1-2. Crisp Sets.

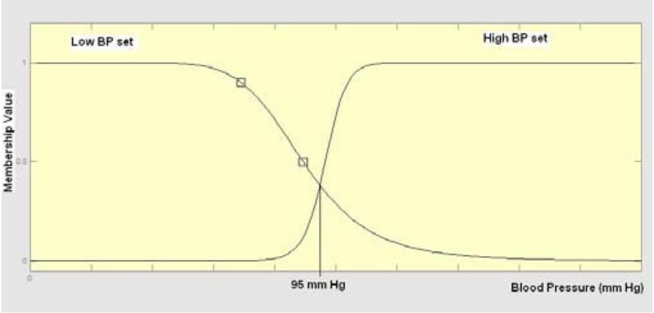
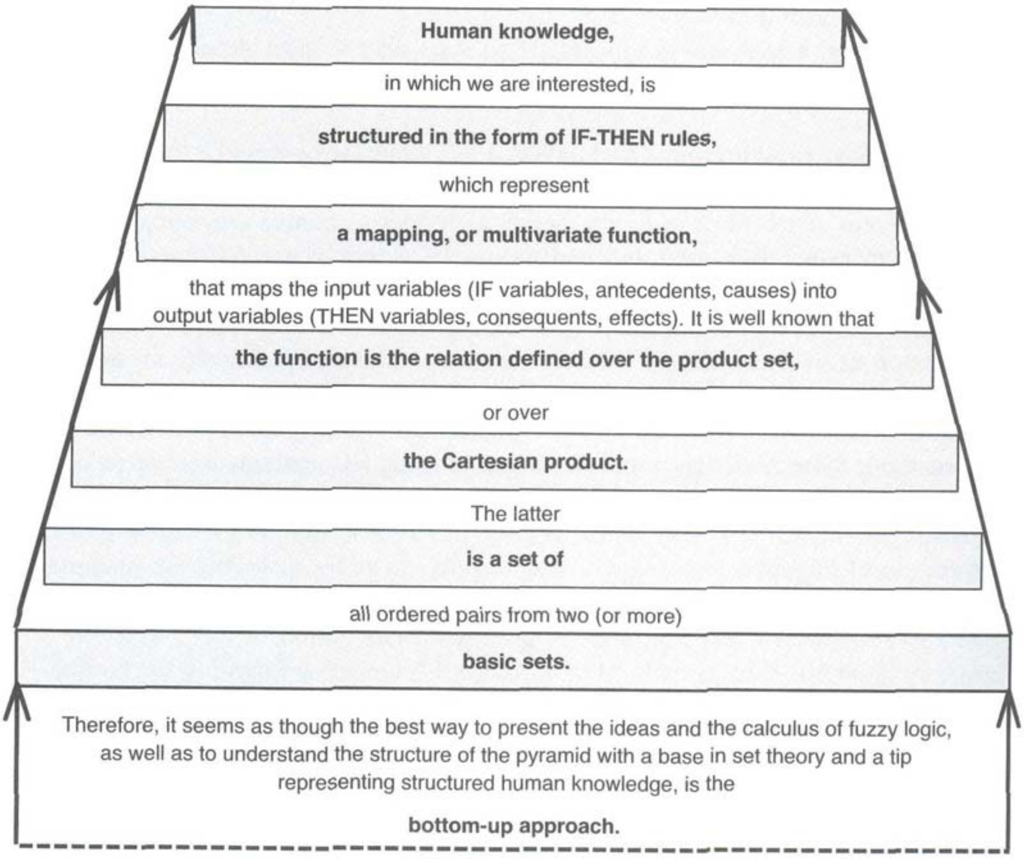


Figure 1.3.1-3. Fuzzy Sets.

Fuzzy sets were first proposed by Zadeh [47] in 1965. Fuzzy sets can be useful when defining such sets where the boundaries of the sets are imprecisely defined. Figure 1.3.1-3 shows the fuzzy set representation for the blood pressure set where the fuzzy variable (BP in the above example) has a degree of membership in the fuzzy set. In fuzzy sets the variable (BP) is mapped into a fuzzy domain through a fuzzy membership function. Theoretically fuzzy set is defined as,

$$\mu s(x) = \begin{cases} >0 & if \quad x \in S \\ 0 & if \quad x \notin S \end{cases} \tag{1.4}$$

Where S is the fuzzy BP_set, $\mu_S(x)$ is the fuzzy membership function which maps the crisp blood pressure values of X on [0, 1], that is, the degree of belonging of element x to the universe X can be any number $0 \le \mu_S(x) \le 1$.



Human knowledge,

in which we are interested, is

structured in the form of IF-THEN rules,

which represent

a mapping, or multivariate function,

that maps the input variables (IF variables, antecedents, causes) into output variables (THEN variables, consequents, effects). It is well known that

the function is the relation defined over the product set,

or over

the Cartesian product.

The latter

is a set of

all ordered pairs from two (or more)

basic sets.

Therefore, it seems as though the best way to present the ideas and the calculus of fuzzy logic, as well as to understand the structure of the pyramid with a base in set theory and a tip representing structured human knowledge, is the

bottom-up approach.

Figure 1.3.1-4. Pyramid of structured human knowledge in the world of fuzzy logic [48].

Due to this capability of fuzzy logic to express linguistic rules in terms of degrees of membership, the vagueness and uncertainties can be easily handled. Fuzzy logic based methods can be used for designing expert systems on the basis of knowledge expressed in a common language [48]. The relationship between fuzzy logic and linguistic knowledge can be graphically illustrated by Figure 1.3.1-4. The capability of fuzzy logic to easily map human knowledge into algorithmic rules has allowed the application of fuzzy logic in many expert diagnostic systems.

Fuzzy logic has been successfully applied in many decision and control applications. A very relevant application of fuzzy logic to this research project was for diagnosis of a rare pathological condition called malignant hyperpyrexia by Lowe and Harrison [5]. Some more publications by Zadeh were studied for more thorough understanding of fuzzy logic and its applications in diagnostic analysis. In another publication [49] titled "Fuzzy Logic = Computing with words" Zadeh demonstrated how robust fuzzy algorithms could be created by exploiting the ability of fuzzy sets to handle imprecise linguistic rules. Zadeh [50] also discussed the use of fuzzy logic and its impact in the domain of soft computing. Soft computing is real life application of computer based methodologies aimed at accommodating the imprecision in real world computer applications. Similar to crisp set operators like AND, OR and NOR, etc., fuzzy operators are defined for fuzzy sets. These fuzzy operators can be used for forming a complex diagnostic rule like the AHV rule mentioned above.

Diagnostic alarm development for anaesthesia monitoring is one such application wherein soft computing methods can be applied. Some other soft computing techniques which were reviewed are discussed below.

1.3.1.2. **Artificial Neural Networks**

Artificial neural networks have found applications in a large number of biomedical applications. Some of these applications include anaesthesiology, radiology, cardiology, psychiatry and neurology [51]. Naguib et al. [51] have summarised an impressive collection of articles on practical application of ANN including patient monitoring. ANN is particularly useful in applications where the problem to be solved is not clearly defined and the development of algorithmic solution is not feasible. ANN finds its application in areas of biomedical application where the knowledge is shallow and it is not feasible to form an algorithmic solution by formulating rules from the expert’s knowledge. One typical application for neural networks is feature extraction [19] for biomedical signal/images wherein the embedded features in the signal/images are unknown. It is therefore considered as the black box approach when employed in a smart computational application [48] like the one proposed in this research project. ANNs are self learning modules and take decisions by learning from examples through training sessions. Neural networks learn by forming internal linkages corresponding to the examples they are trained with. Thus sufficient sample data sets need to be available for each critical condition that needs to be diagnosed using the ANN based algorithms. However for some rare critical conditions for example malignant hyperpyrexia sufficient data might not be available beforehand. Also “The lack of a readily understandable explanation for the ‘black box’ operation of neural networks prohibits some potential further deployment”[52]. Due to these limitations ANN cannot be directly used for diagnostic applications in anaesthesia.

1.3.1.3. **Logistic Regression**

Logistic regression is a statistical tool for finding the best relationship between an outcome (dependent variable, diagnosis) and a set of independent variables (patient data), the outcome variable is binary (‘0’ or ‘1’). These set of independent variables are also called covariates [53]. Thus logistic regression can be used for determining the outcome (i.e. diagnosis/decisions) from multiple independent variables (like blood pressure, pulse volume, heart rate, etc). Some logistic regression based applications are mentioned below.

Diagnosing pathological events during anaesthesia monitoring is analogous to fault detection and diagnosis (FDD) processes in industrial applications [11]. Logistic regression is widely used in many FDD processes. Sutanto [54] discussed one such industrial application wherein logistic regression-based FDD technique was used for predicting future faults in a printing process. In this article Sutanto suggested that logistic regression analysis with successive data collection (as in anaesthesia monitoring) could result in an unstable process with incomprehensible results. Such instability should be anticipated as regression analysis is very sensitive to the nature of the data collection and the outliers present in the input data set. Physiological data from the patient monitors are very likely to be artifact and noise prone and thus a regression-based diagnostic module is very likely to suffer instability.

An expert system called "TrenDx" developed by Haimowitz et al. [16, 55] was a regression-based system which was capable of detecting stimulus (clinically significant trends) in patient data. TrenDx has successfully diagnosed paediatric growth trends in the patient data achieving same diagnostic performance as the experts. However, TrenDx was in early stages of the development and needed further work before it could be used for real-time application. Moreover the diagnostics in anaesthesia are far more rigid in time than paediatric and intensive care monitoring for which TrenDx was designed.

1.3.1.4. **Statistics based Physiological Alarms**

For diagnosing anaesthesia related critical events disease patterns in the patient data need to be identified. Identification of these disease patterns involves detecting excessive changes in the physiological variables. Many statistical tools like logistic regression discussed above and clustering analysis [56] were available. Harrison and Connor [13] [In press] suggest that a fall in BP of 10 mmHg from a previous value of 70 mmHg had far greater significance than a fall of 10 mmHg from 150 mmHg. Thus it becomes difficult to develop a simple algorithm for detection of adverse changes in physiological variables. Probabilistic alarms from sequential physiological measurements by Harrison and Connor is mathematically straightforward and is a rapid statistical tool for detecting clinically adverse physiological variations. The methodology of this probabilistic alarm algorithm is explained in detail in Chapter 3. The adverse changes in physiological variables (i.e. the output of probabilistic alarm algorithm) are expressed in multiples of standard deviations (SD) and the sensitivity of the probabilistic alarms can be tuned by specifying an appropriate alarm limit. For instance an alarm limit of 2SD can be set for BP. Any adverse changes in BP data which generate a probabilistic alarm level greater than 2SD could be seen as a strong evidence for a clinically significant change.

A probabilistic alarm algorithm is logically straight forward and simple for the anaesthetist to understand. The reasoning behind the alarm level will become more transparent to the anaesthetists if this algorithm were to be employed in an expert diagnostic system. Probabilistic alarms also have the advantage of being computationally less demanding compared to other complex diagnostic algorithms like fuzzy logic and ANN. Harrison and Connor have also described the heuristic relationships patterns for some common critical states which might arise during anaesthesia administration.

These heuristic relationships (refer Table 1) combine the transformation in observable physiological variables like blood pressure, heart rate, pulse-volume and end-tidal CO_2 to reveal the patterns for clinical pathological states. For instance the heuristic relationship for absolute hypovolaemia is identified by *an increase in HR and decrease in BP and decrease in PV and variable* $ETCO_2$.

Table 1. Heuristic relationships between deviations in observable physiological variables and clinical pathological states [13].

	Heart Rate (HR)	Blood Pressure (BP)	Pulse Volume (PV)	End Tidal CO2 (ETCO2)
Relative hypovolemia	~	↓	↑	~
Absolute hypovolemia	↑	↓	↓	~
Sympathetic response	↑	↑	↓	↑
Fall in cardiac output	↑↓	↓	↓	↓

Where,

↑	indicates increase in the physiological parameter
↓	indicates decrease in the physiological parameter
↑↓	indicates increase or decrease in the physiological parameter
~	indicates a variable physiological parameter.

1.3.1.5. **Selection of Diagnostic Techniques**

After consultation with the external project supervisor Dr. Harrison (Anaesthetist), it was decided that multiple diagnostic modules should be employed for generating diagnostics in the diagnostic alarm prototype. In order to reduce visual clutter on the user interface of RT-SAAM the number of diagnostic modules were limited to three. Based on the literature review presented above it was decided that a fuzzy logic based module would be used for developing the diagnostic alarm prototype proposed in this research project. Fuzzy logic was chosen because of its successful offline application in the field of anaesthesia monitoring and its capability to handle the linguistic complexity in the expert knowledge [11]. The probabilistic alarm algorithm was selected as the second module. It was anticipated that computational simplicity of the probabilistic alarm algorithm and the transparency of the reasoning behind the probabilistic alarm results would give the anaesthetist more confidence when using the evidence presented by the prototype system. The third diagnostic module (SPV based diagnostic module) selected for the prototype system requires understanding of some clinical knowledge which is presented in the following section.

1.4. Clinical Literature

Human anatomy and physiology is complex and covering the minute details is beyond the scope of this text, we concentrate our attention on the cardio-pulmonary system and the associated physiological parameters which are important from this research point of view. The human cardio-pulmonary system consists of the cardiovascular system and the respiratory system.

1.4.1. Cardiovascular System

From the moment its starts beating until the moment it stops the heart works tirelessly providing the power needed for life. The heart undergoes fairly complicated movements as it alternately contracts (systole), forcing the blood out of its chambers, and relaxes (diastole), refilling its chambers with blood [57]. The cardiac cycle of the heart consists of all the events associated with the heart during one complete heart beat. Figure 1.4.1-1 shows the cardiac cycle and describes the events occurring in the left side of the heart, we focus our attention only to events significant from the project point of view.

During ventricular systole as the atria relax, the ventricles contract and their walls close in on the blood in their chambers and the pressure steeply rises (phase 2a in Figure 1.4.1-1). The blood pressure in the heart is at its peak. This is called **systolic pressure**. During relaxation, the atrioventricular valves are open and the ventricles relax causing the pressure to drop. The backflow of the blood in the aorta and pulmonary trunk closes the semilunar valves causing the pressure to rise briefly. This is shown in Figure 1.4.1-1 as the dicrotic notch. The ventricles relax completely and this marks the **diastolic pressure**. The blood pressure is always expressed in pairs (systolic pressure and diastolic pressure). The typical range for systolic and diastolic pressure is in the range 110 to 130 mmHg and 70 to 80 mmHg respectively [57]. The mean of the blood pressure waveform computed over a fixed interval of time is called mean arterial pressure (**MAP**). The systolic pressure value derived from invasive pressure waveform (BP/P1) is also called systolic arterial pressure (**SAP**).

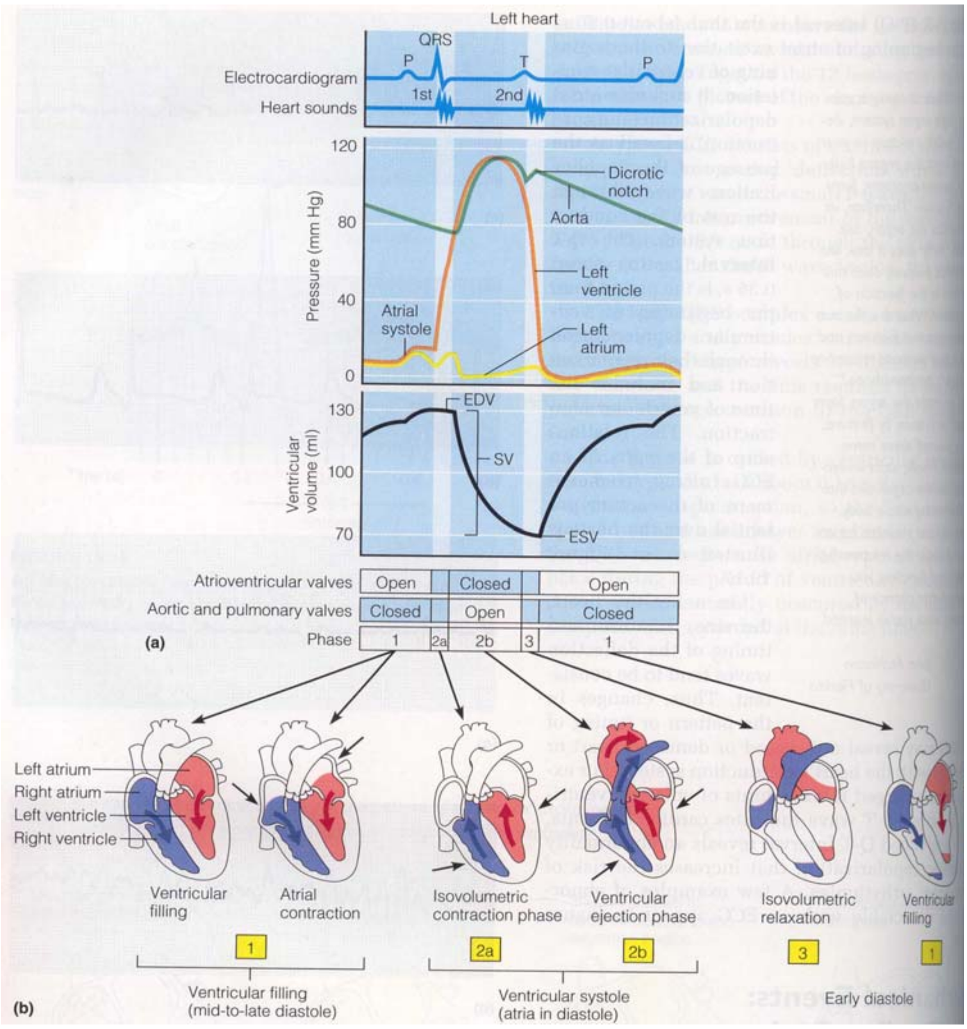


Figure 1.4.1-1. Summary of events occurring in the heart during the cardiac cycle [57].

(a) Events in the left side of the heart. An ECG tracing is superimposed on the graph (top) so that pressure and volume changes can be related to electrical events occurring at any point. Time occurrence of heart sounds is also indicated.

(b) Events of phases 1 through 3 of the cardiac cycle are depicted in diagrammatic views of the heart.

1.4.1.1. **Cardiac Output**

Cardiac output (**CO**) is defined as the amount of blood pumped out by each ventricle in one minute. **Heart rate** (**HR**) is calculated as the number of contractions (beats) of the heart in one minute. The unit of HR is thus beats per minute (bpm). The average heart rate for an adult male is 75 bpm. CO can be expressed as the product of HR and **stroke volume** (**SV**). Stroke volume is defined as the amount of blood pumped out by a ventricle with each beat. The average SV for an adult male is 70 ml per beat [57]. Thus,

CO (ml/min) = HR (75 beats/min) x SV (70 ml/beat)

∴ **CO = 5250 ml/min (5.25L/min).**

Cardiac output may be reduced due to numerous reasons. The fall in cardiac output (**FCO**) can be defined as state where the compensatory mechanisms in the human body are insufficient to maintain the blood circulation.

1.4.1.2. **Pulse volume (PV)**

Plethysmography is a non-invasive method for measuring the blood flow within the blood vessels, or arteries [58]. Pulse oximetry is primarily for the measurement of oxygen saturation of haemoglobin but as a by-product there is a unitless ‘pulse volume’ output. It can be used as one of the parameters for monitoring patient’s hemodynamic state during anaesthesia.

1.4.2. Respiratory System

During inspiration the diaphragm contracts, which results in the expansion of the thoracic cavity (refer Figure 1.4.2-1). As the thorax expands the lungs get pulled along with it and intra-alveolar and pleural pressure fall below atmospheric level (about -3mm Hg). Air rushes into the lungs due to the pressure gradient which now exists.

During natural breathing (unventilated) the pressure in the pleural cavity (between the lungs and thoracic cage) is negative (about -3mm Hg). During positive pressure ventilation expansion of the lungs occurs with positive pressure and increases the pleural pressure, this reduces the flow of venous blood into the chest and therefore reduces venous return to the heart (refer Figure 1.4.2-2). In the presence of absolute hypovolaemia (blood loss) the venous return to the heart is markedly reduced. Venous return will vary throughout the respiratory cycle (because of intra-thoracic pressure changes) and thus the stroke volume will vary and this will influence the systemic arterial pressure

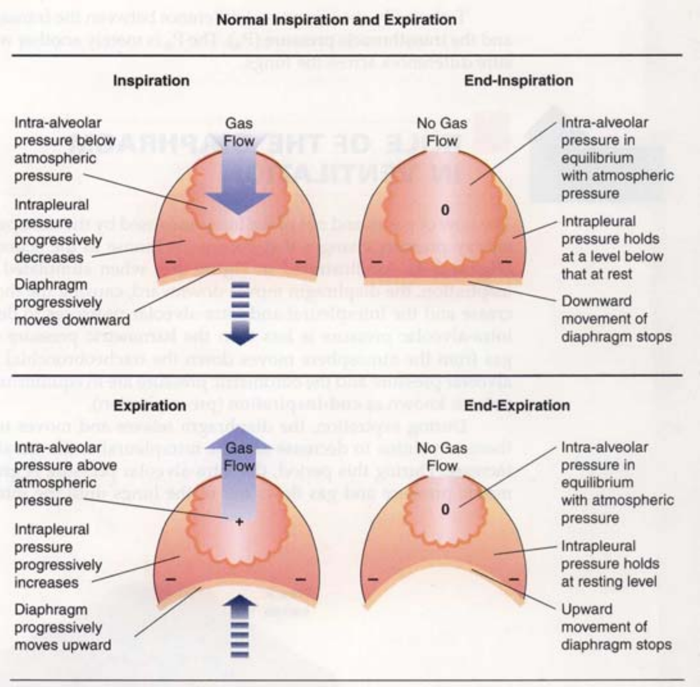

Normal Inspiration and Expiration
Inspiration
End-Inspiration
Intra-alveolar pressure below atmospheric pressure
Gas Flow
No Gas Flow
Intra-alveolar pressure in equilibrium with atmospheric pressure
0
Intrapleural pressure progressively decreases
Intrapleural pressure holds at a level below that at rest
Diaphragm progressively moves downward
Downward movement of diaphragm stops
Expiration
End-Expiration
Intra-alveolar pressure above atmospheric pressure
Gas Flow
No Gas Flow
Intra-alveolar pressure in equilibrium with atmospheric pressure
+
0
Intrapleural pressure progressively increases
Intrapleural pressure holds at resting level
Diaphragm progressively moves upward
Upward movement of diaphragm stops

Figure 1.4.2-1. Natural respiration [59].

.

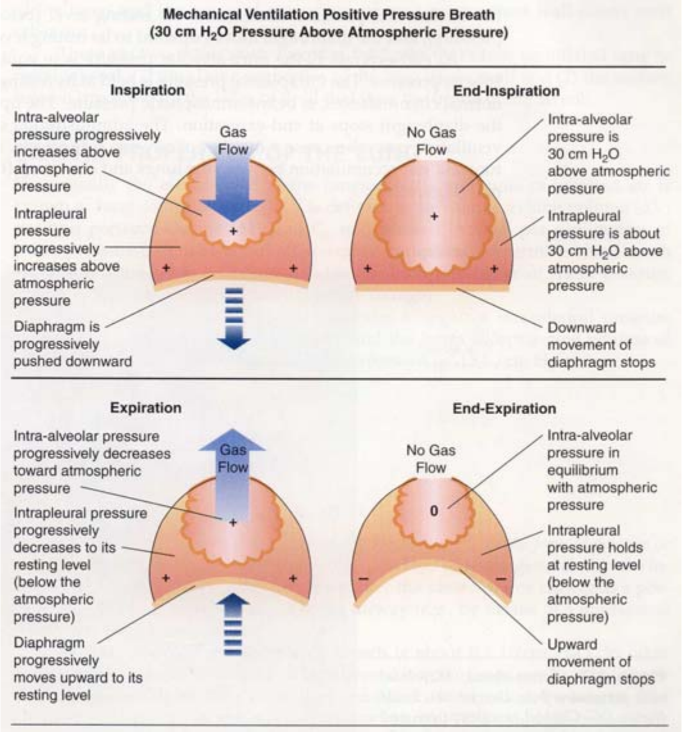

Mechanical Ventilation Positive Pressure Breath
(30 cm H_2O Pressure Above Atmospheric Pressure)
Inspiration
End-Inspiration
Intra-alveolar pressure progressively increases above atmospheric pressure
Gas Flow
No Gas Flow
Intra-alveolar pressure is 30 cm H_2O above atmospheric pressure
+
Intrapleural pressure progressively increases above atmospheric pressure
Intrapleural pressure is about 30 cm H_2O above atmospheric pressure
Diaphragm is progressively pushed downward
Downward movement of diaphragm stops
Expiration
End-Expiration
Intra-alveolar pressure progressively decreases toward atmospheric pressure
Gas Flow
No Gas Flow
Intra-alveolar pressure in equilibrium with atmospheric pressure
0
Intrapleural pressure progressively decreases to its resting level (below the atmospheric pressure)
Intrapleural pressure holds at resting level (below the atmospheric pressure)
Diaphragm progressively moves upward to its resting level
Upward movement of diaphragm stops

Figure 1.4.2-2 . Positive pressure (mechanical) respiration [59].

Theoretically a ventilation failure will occur if the load placed on the respiratory system exceeds its capacity. The conditions responsible for increasing the respiratory load are also responsible for reducing ventilatory capacity and will place the individual at the risk of respiratory insufficiency [60]. In the event of absolute hypovolaemia blood loss occurs reducing the venous supply to the respiratory system and hence reducing the drive capacity. Mechanical ventilation changes the pleural pressure in the thoracic cavity and thus the relationship of the intra-thoracic and extra-thoracic structures by changing the right and left ventricular loading. By monitoring the respiration induced systolic pressure variations (**SPV**), hemodynamic changes (particularly AHV) can be detected. SPV can be used for monitoring sympathetic regulation in anaesthetized patients under positive-pressure (mechanical) ventilation [61].

Rook et al.[62] and Magder [63] have indicated in their studies that SPV provides a useful measure for fluid responsiveness. The changes in SPV are related to the blood volume capacity of the patient, for instance increasing SPV indicates increasing hypovolaemia. Ornstein and co-workers [64] suggested that SPV is a better indicator than loss in blood volume.

Dalibon et al. [65] studied 67 subjects for assessing the accuracy of the SPV for detecting low ventricular preload in non-hypotensive patients. Through this study they had concluded that in anaesthetized non-hypotensive patients SPV does not reflect low left ventricular preload. Thus in such patients SPV was not an accurate measure for hypovolaemia. But patients undergoing major surgical procedures are very likely to suffer blood loss and hence majority of the patients under anaesthesia monitoring are likely to be hypotensive. A SPV based algorithm was chosen as the third diagnostic algorithm for the proposed diagnostic alarm prototype.

1.4.2.1. **End-Tidal CO_2 and Inspired CO_2**

Carbon-dioxide (CO_2) produced by cellular metabolism is transported to the right ventricle by the venous system. It is hence pumped to the lungs by the heart and diffuses out into the exhaled air. The concentration at the end of the expiration cycle is called End tidal CO_2 ($ETCO_2$), which reflects the metabolism, circulation and ventilation [66].

1.4.2.2. Inspired CO_2

Inspired carbon-dioxide (ICO_2) is the carbon-dioxide breathed in by the patient and is normally zero. If rebreathing occurs then the ICO_2 may be greater than zero. A raised CO_2 will affect the patient's hemodynamic state [67].

SPV values reflect changes in blood volume capacity only if patient is under positive pressure ventilation [61]. In the SPV based diagnostic module proposed in section 1.4.2, ICO_2 and $ETCO_2$ values are to be used for determining if the patient is breathing with or without positive pressure (mechanical) ventilation. The detailed functionality of the SPV algorithm is explained in the following chapter.

1.5. Research Contribution

During the course of research, new filtering algorithms, signal processing algorithms and diagnostic modules were developed. An interactive user interface was programmed after consultation with the anaesthetists. The diagnoses generated by the diagnostic alarm prototype were compared with the diagnoses generated by the expert (anaesthetist). The objective was to obtain high level of agreement between the two diagnoses thereby validating the reliability of the diagnostic algorithms.

The real-time version of the proposed system is called "Real-Time Smart Alarms for Anaesthesia Monitoring (RT-SAAM)" and it required new communication protocol to be developed for testing RT-SAAM. This setup will be explained in detail in Chapter 4. It is anticipated that this communication protocol can be used for other anaesthesia related research wherein data collection is required.

This research has identified probability (statistic) based alarms, fuzzy logic based alarms and systolic variations in blood pressure with respiration (SPV) based alarms as the three diagnostic alarm modules for generating diagnoses. The development of these three modules and their application for anaesthesia monitoring will be explained in the following chapters. This research forms a small component of the ongoing development for the proposed diagnostic alarm prototype which will be further refined through more rigorous real-time testing.

1.6. Thesis Outline

Chapter 1: This chapter presents the background, motivations, literature review and contributions of the research documented in this thesis. The theoretical development of the research concepts and the diagnostic process are also discussed. The following paragraphs briefly describe the contents of other chapters of this thesis.

Chapter 2: Data Analysis and Algorithm Development discusses the data simulation, relevant signal processing and diagnostic theory. The analytical working of the diagnostic modules is also explained, followed by the real-time algorithm development which added functionality to the proposed diagnostic alarm prototype.

Chapter 3: Real Time System Development presents the communication protocol developed for conducting real-time testing of the RT-SAAM. An overview of the real-time system is given and the real-time implementation of RT-SAAM is discussed in detail.

Chapter 4: Diagnostics and Testing describes the data collection process wherein the actual experimental setup is shown and explained. The system validation section in this chapter reviews the diagnostic method used and the offline and real-time analysis test results.

Chapter 5: Discussion and Conclusions summarises the major aspects of the research outcomes and possible future works in refining the prototype that was developed in the previous chapters.

1.7. Summary

The aim of this thesis is to extend the work of Andrew Lowe and Michael Harrison [5, 11, 68, 69] by developing real time algorithms for analysis of the hemodynamic data from the patient and providing a warning/alarm to the anaesthetist. The alarm system prototype developed in the following chapters uses the new fuzzy logic algorithms for real-time diagnosis; they are however based on same underlying principles as in SENTINEL by Lowe. The newly developed probabilistic alarm algorithms and SPV based diagnostic alarm algorithms were also discussed.

A major part of this thesis is the amalgamation of the results from these three algorithms in such a way that the anaesthetist was presented with a real time interactive display, this allowed the anaesthetists to respond to the diagnosis (agree/disagree) in real time.

Chapter 2 Data Analysis and Algorithm Development

2.1. Introduction

Digital signal processing (**DSP**) has been successfully applied in many biomedical applications including patient monitoring and diagnostic systems [5, 38]. This chapter introduces some of the techniques which have been used in SAAM. Furthermore, we discuss development of the signal processing algorithms which form the foundation blocks for the proposed alarm system. Based on the theory and techniques reviewed in Chapter-1 and Chapter-2 a modular alarm system is proposed. This chapter provides with an in depth description of LabVIEW^{TM} and MATLAB^{TM} algorithms employed in SAAM. Some of the topics discussed in this chapter are noise filtering techniques, signal processing algorithms, diagnostic techniques followed by the diagnostic modules.

2.2. Signal Processing Theory

SAAM uses physiological data derived from the output port of the Datex Ohmeda S/5 anaesthesia monitor. There are various physiological signals which are available at the output port of the S/5 anaesthesia monitor. The signals of our interest are invasive pressure waveform (**P1**), plethysmography (Pulse volume-**PV**) and end tidal CO_2 (**$ETCO_2$**). These signals represent the analogue waveform information for the P1, PV and $ETCO_2$ signals in a digital time-series format. Digital signals are derived by sampling the analogue signals coming from various sensors at a particular sampling rate. For instance the analogue signal coming from the pulse oximetery probe is sampled at a frequency of 100 Hz, i.e. 100 digital samples are recorded in a buffer at every 1 second interval. Since the digital representation of the analogue signal is discrete and not continuous in time, these digital signals are also called ‘discrete-time series signal’.

The raw signals acquired from the S/5 anaesthesia monitor have embedded noise/ artifacts due various pneumatic, hydraulic, optical and electronic devices. These noise/artifacts need to be discarded; this is called pre-processing of digital signals. Pre-processing of the signals involve filtering and smoothing of the original signal to make it noise-free. Numerous signal processing techniques were employed for filtering the individual signals. This section describes the theory and algorithm development for each of the filtering technique that has been employed in SAAM.

2.2.1. Noise and Artifact Rejection

Noise/artifacts in the raw physiological signals have a different frequency component and thus cannot be eliminated by employing a single band stop filter. Various filtering techniques with different specifications were tried for achieving the desired level of filtering. The cleaned signal waveforms obtained through noise filtering were to be used for diagnosis.

2.2.1.1. Spectral Analysis of the Raw Signals

The physiological signals have specific frequency components which represent the rate of changes of the physiological variables. The frequency spectrum of the signals can be used for distinguishing the clean signal samples from the noisy samples.

In order to define filter specification for a signal it is essential to distinguish between the useful frequencies and the noisy frequencies in the raw signals obtained from the anaesthesia monitor. This process of analysing the frequency content in the signal is called spectral analysis. To obtain the frequency content of a discrete time signal we need to transform the time domain signal into the frequency domain. The DSP tool used for transforming the time domain signal into the frequency domain signal is called discrete-time Fourier transform (DTFT) [70].

2.2.1.2. Discrete-Time Signals

In digital signal processing, signals are represented as sequences of numbers called samples. A sample value of a typical discrete-time signal or sequence is denoted as *x[n]* where the argument *n* is an integer between $-\infty$ and ∞. It should be noted that *x[n]* is defined only for integer values of n and is undefined for non-integer values of the argument *n* [70].

2.2.1.3. Discrete-Time Fourier Transform (DTFT)

The discrete-time Fourier transform (DTFT) of a discrete-time sequence *x[n]* is a complex exponential representation $\{e^{-j\omega n}\}$ where ω is a real frequency variable.

The discrete-time Fourier transform $X\left(e^{j\omega}\right)$ of a discrete-time sequence *x[n]* is defined by Eq.2.1.

$$X\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} x[n]e^{-j2\omega n} \qquad \textbf{2.1}$$

2.2.1.4. **Discrete Fourier Transform (DFT)**

For a finite-length sequence *x[n],* $0 \le n \le N-1$ there exists a simpler relation between the sequence and its discrete-time Fourier transform $X\left(e^{j\omega}\right)$. For a length-N sequence, only N values of $X\left(e^{j\omega}\right)$, called the frequency samples, at N distinct frequency points, $\omega = \omega_k$, $0 \le n \le N-1$, are sufficient to determine *x[n]*, and hence, $X\left(e^{j\omega}\right)$, uniquely. This frequency domain representation of a finite-length sequence is called discrete Fourier transform (DFT). The DFT is one of the most commonly used DSP tool for analysing the frequency content of a time-series signal. By definition the DFT of a sequence *x[n]* is given by following equation (refer to Eq.2.2).

$$X[k] = X\left(e^{j\omega}\right)\Big|_{\omega=2\pi k/N} = \sum_{n=0}^{N-1} x[n]e^{-j2\pi\frac{k}{N}n} \qquad 0 \le n \le N-1 \qquad \textbf{2.2}$$

The DFT of a sequence can be computed efficiently in practice using a fast Fourier transform (FFT) algorithm. MATLAB$^{\text{TM}}$ provides built-in functions for the computation of the DFT. The power spectrum of the signal gives the measurement of power in a signal at different frequencies. These functions were implemented during the data analysis for identifying the frequency spectrum of the clean physiological data in the raw signals. After performing the DFT of the BP and PV signals a power spectrum of the signals was plotted. The clean signals are identified by the high power band in the frequency spectrum. Figure 2.2.1-1 shows the power spectrum of the BP signal transformed to the frequency domain using FFT algorithm. It illustrates that the frequency of clean BP data is in the range 0 Hz to 2 Hz (high power content).

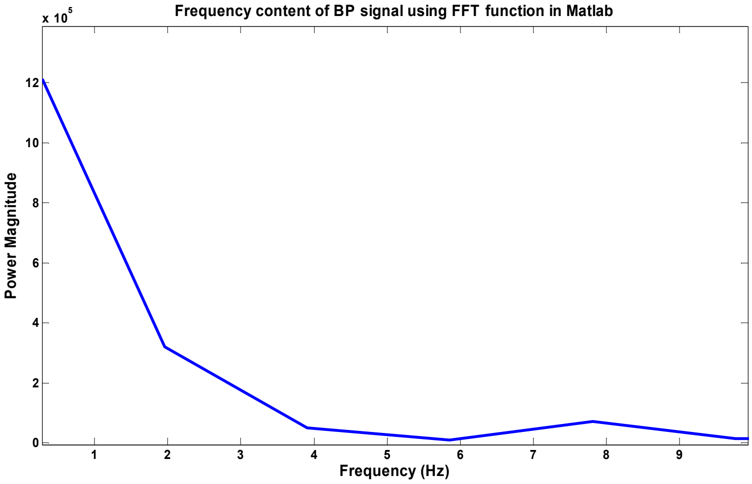


Figure 2.2.1-1. Power spectrum of BP signal.

2.2.1.5. **Lowpass Filter**

A lowpass filter passes the signals in the lower spectrum of the frequency and blocks the signals in the higher frequency range. Some signal sources generate interference/noisy signals and these sources are ambient sources producing noisy signal of much higher frequency than the useful signals. For instance, noise due to cauterization procedure usually has a frequency much higher than the signal under consideration. Other artifacts are embedded in the signal from the sensor itself. For example touching or moving the invasive pressure sensor will generate noise, resulting in a P1 signal which is highly contaminated. By filtering the signal with a lowpass filter noise from high frequency noise sources can be eliminated. For in depth theory on filter design refer to [70].

MATLAB^{TM} has built in filter modules that generate filter coefficients automatically for user specified filter specifications. Once the frequency range is identified for the filter the MATLAB^{TM} FDAtool can be used for generating appropriate filter coefficients. Figure 2.2.1-2 shows a standard low pass filter specification and illustrates the symbols used in this context.

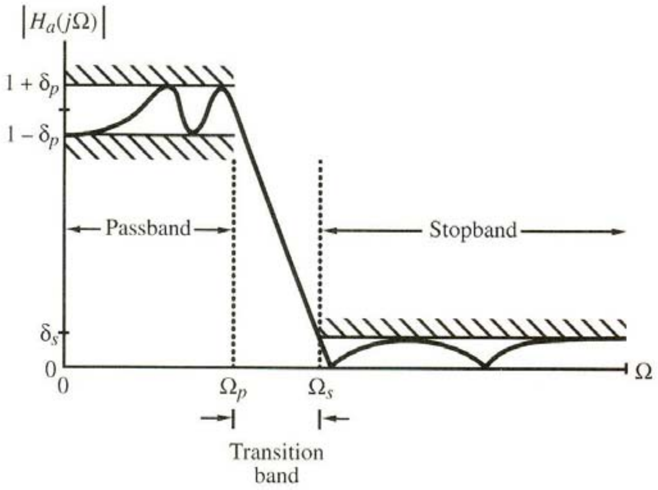


Figure 2.2.1-2. Lowpass Filter Specification [70]

Where,

Ω_p ➔ Passband edge frequency.

Ω_s ➔ Stopband edge frequency.

δ_p ➔ Passband ripple.

δ_s ➔ Stopband ripple.

$|H_a(j\Omega)|$ ➔ Magnitude of the filter.

All signals having frequencies in the range of 0 to Ω_p pass through filter without any attenuation while the signals with frequencies greater Ω_p are highly attenuated. Ideally a low pass filter should have a transition band of infinitesimally small band width (i.e. $\Omega_s \approx \Omega_p$), which requires a very high filter order and thus increases the computation requirement for the executing the filter algorithm. In practice filter has a transition band of moderate width which is a compromise between the ideal filter specification and the available computational resources. Also the magnitude response of the passband and stopband is not constant and specified with some tolerance limits in terms of the ripple content. Figure 2.2.1-3 shows the practical magnitude response of a lowpass filter for invasive blood pressure (P1) waveform.

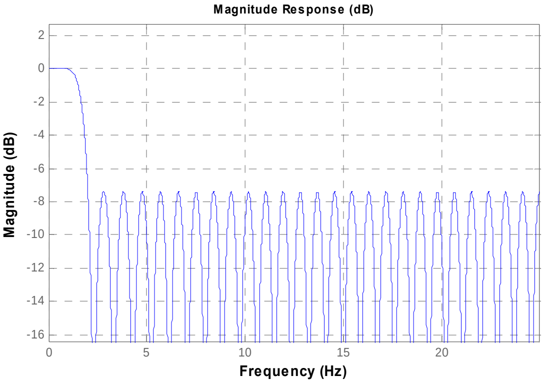


Figure 2.2.1-3. Magnitude response of a lowpass filter for invasive pressure (P1) waveform.

2.2.1.6. **Adaptive filtering**

Adaptive filters are basically digital filters with self-adjusting characteristics (specifications). An adaptive filter consists of two distinct parts: a digital filter with adjustable coefficients, and an adaptive algorithm which modifies the coefficients of the filter depending upon the noise characteristics of the contaminated signal. An adaptive filter was also tested with the sample BP and PV data but successful filtering could not be achieved due to the high variability of the noise in the input signals [71].

2.2.1.7. **Variance Based Filtering**

Variance and standard deviation are extensively used in the following algorithms and it is worthwhile to take a brief look at these terms. In statistics theory, the variance is defined as the measure of statistical dispersion of a random variable. Variance of a random number indicates how dispersed the population sample is for the random variable in the given data samples. The square root of the variance is called standard deviation.

Standard deviation measures how widely spread the values in a data set are. If the data points are all close to the arithmetic mean, then the standard deviation is close to zero. If data points are spread far from the mean, then the standard deviation is far from zero.

The variance based filtering technique discussed in this chapter was used for eliminating major artifacts. The variance based filtering algorithm developed for SAAM analyses variance of data-batches, which represent a segment of the whole time series data, and filters the time-series data on a batch-by-batch basis. The functionality of this filtering algorithm is illustrated by the block diagram shown in Figure 2.2.1-4 and explained below.

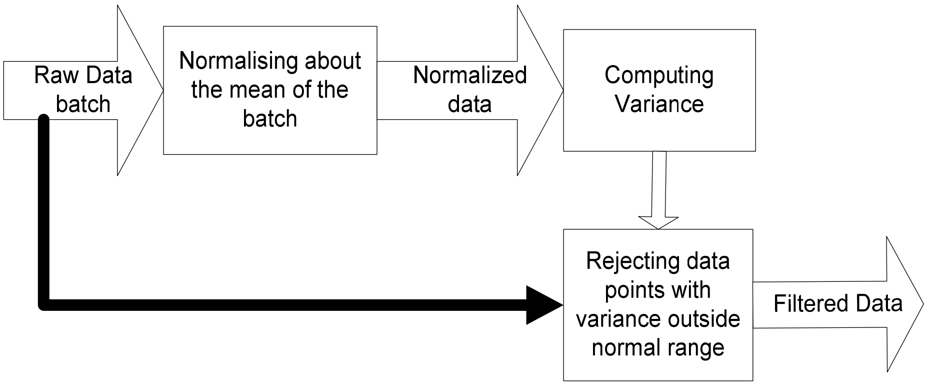

Raw Data batch
Normalising about the mean of the batch
Normalized data
Computing Variance
Rejecting data points with variance outside normal range
Filtered Data

Figure 2.2.1-4. Block diagram for variance based filtering technique.

Variance based filtering algorithm:

1. The mean of each batch of input data is computed.
2. The computed mean is then subtracted from the original batch. This is termed as ‘normalization of the data about the mean value’.
3. Variance of the normalized data is computed.
4. The data batches which have embedded artifacts have a variance which lies outside the normal variance thresholds and these batches of data are rejected from the raw input data.
5. The data set resulting from step-4 is filtered data with all major artifacts rejected.

As evident from Figure 2.2.1-6 the high frequency component is eliminated through lowpass filtering from the raw BP signal shown in Figure 2.2.1-5. Further filtering using a variance based filtering algorithm eliminates the noisy signals (refer to Figure 2.2.1-7). The highlighted portion (in red) of the waveform shows the high frequency noise before and after performing lowpass filtering. After performing the lowpass filtering and variance based filtering the resultant P1 waveform (Figure 2.2.1-7) shows that majority of the artifacts are completely eliminated.

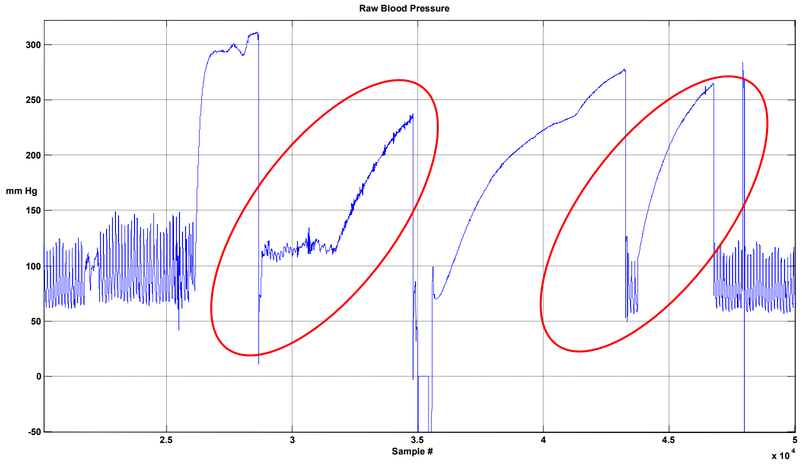


Figure 2.2.1-5. Raw P1 waveform.

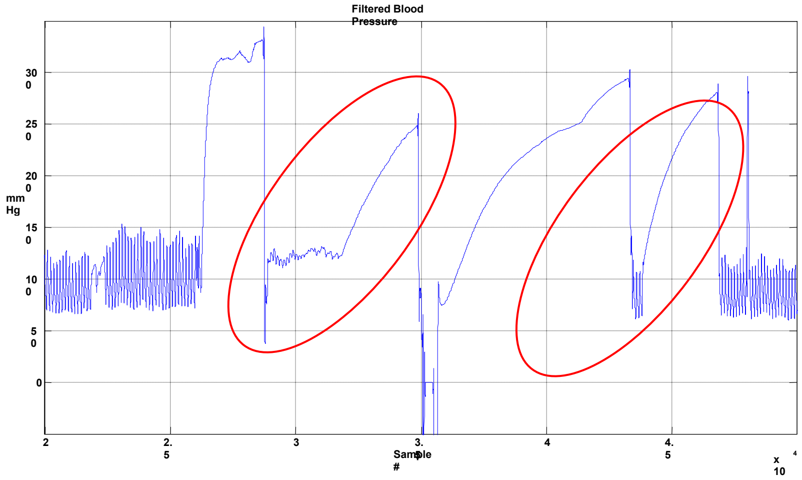


Figure 2.2.1-6. Lowpass filtered P1 waveform.

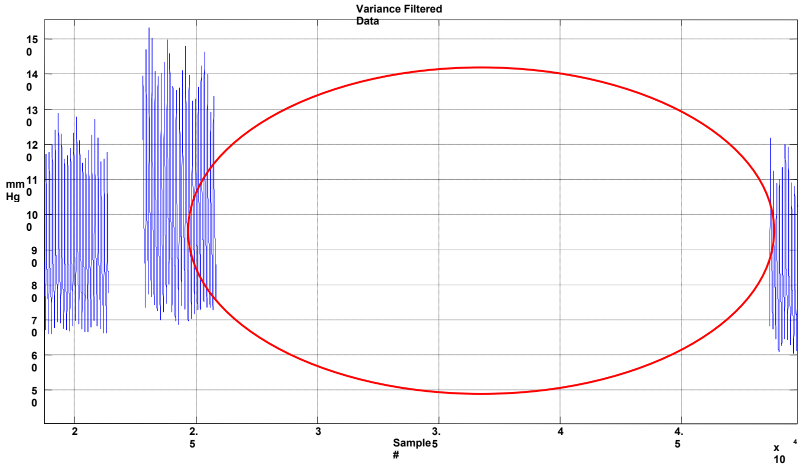


Figure 2.2.1-7. Filtered data after variance based filtering.

2.2.2. Peak (or Trough) Detection

Some of the diagnostic techniques described in this section and the following section require peak (and/or trough) detection from the invasive blood pressure waveform (P1) and the plethysmographic waveform (PV). For instance the peak detection from the P1 waveform is essential for computing the heart rate (HR). SAAM employs two separate peak (or trough) detection algorithms for P1 and PV waveforms which differ slightly in the operation but use the same underlying principle.

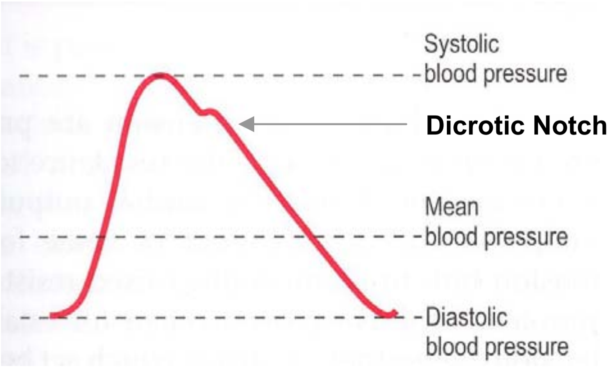


Figure 2.2.2-1. Blood Pressure (P1) waveform [72]

Several algorithms were studied [73-75] and tried for peak detection of P1 and PV waveforms, some of these algorithms are listed below with their advantages and disadvantages with respect to the signals under study:

2.2.2.1. **Slope-based peak detection:**

This technique simply check for a change in slope from positive (rising) slope to negative (falling) slope for detecting the peaks in the P1 and PV waveforms. For detection of troughs in the peak it checks for change in slope of the waveforms from negative slope to positive slope. The slope based detection by itself is not very reliable because some of spurious peaks or even the peaks due to the dicrotic notch get detected as a systolic pressure peak (Refer Figure 2.2.2-1) in case of P1 waveforms.

2.2.2.2. **Maxima/Minima-based peak (or trough) detection:**

Jacobson [74] had proposed a periodic Maxima-detection based method. This algorithm simply computes the maximum and minimum values for a window of data with length equal to one cycle length of the waveform under consideration. The data samples with the maximum and minimum values are then detected as peaks and troughs for each window of the data.

Peak (or trough) detection algorithms used in SAAM are a hybrid of the basic peak (or trough) detection techniques described above and supplementary threshold limits on the input waveform. These peak (or trough) detection techniques are more signal specific and optimum for the P1 and PV waveforms which are used for generating the diagnosis.

Listed below are the criterions which need to be satisfied for a peak in the P1/PV waveform to be qualified as a genuine peak:

1) There should be a change in the sign of the derivative (slope-change).
2) The data point on the waveform should be a local maxima.
3) There should not be an already detected peak in the previous data segment equal to half the periodic length of the waveform. For instance P1 waveform sampled at 100 Hz has a period of 100 data points. If a peak was detected in the P1 waveform at 25th sample then no genuine peak will be detected for the next 50 data points (until the 75th sample). This avoids spurious peaks or peaks due to the dicrotic notch.
4) Threshold limiting condition is imposed on peaks, to filter out the spurious peaks. For e.g. the P1 value is limited by limiting values of 50 and 220 mmHg.

In the following sections we first introduce the three diagnostic models which generate the diagnosis in SAAM and then we apply all the techniques discussed above for analyzing the three input signals and generating an alarm level based on the diagnosis generated by the algorithms.

2.3. Model Overview

For a diagnostic system to be of any clinical significance the system needs to be reliable and accurate. Both these requirements can be met by implementing multiple diagnostic modules (redundancy). The proposed diagnostic alarm system therefore employs the following three modules for generating a diagnosis.

- Statistic (Probability based) module.
- Systolic variation in arterial pressure with respiration (SPV) based module and
- Fuzzy logic module.

The modular structure for proposed alarm is shown in Figure 2.2.2-1 below.

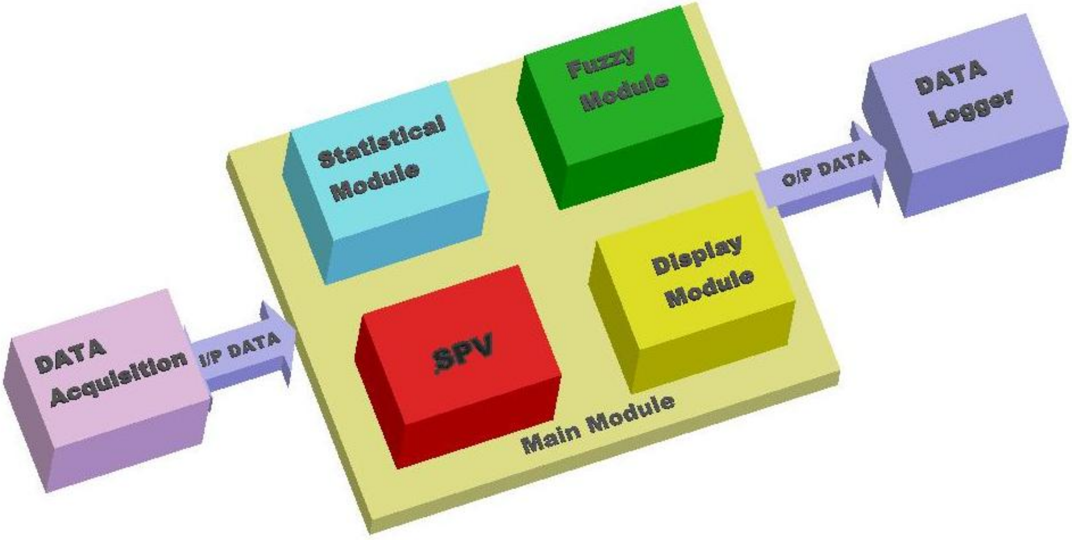


Figure 2.2.2-1. Block Diagram of the Alarm Model.

2.3.1. Probabilistic Model

The probabilistic alarms based model is a real time application of statistics based physiological alarms proposed by Harrison and Connor [13]. In this model the various input parameters to the system such as Blood Pressure (BP), Heart Rate (HR), Pulse Volume (PV) and End Tidal CO_2 ($ETCO_2$) are normalised about the mean data collected from a cluster of population and they are then expressed in terms of multiples of standard deviations in respect to the population data of same age group as the current patient being monitored. Thus all the input parameters are expressed in terms of a common scale of standard deviations. The alarm levels are then computed for each of the four signals being monitored. For in depth explanation of the probabilistic alarm algorithm refer to section 2.4.1.2. Alarm levels for individual signals are then merged together to give the Statistical Alarms based on the expert rules for Absolute Hypovolaemia (AHV) and Fall in Cardiac Output (FCO) provided by anaesthetists.

2.3.2. Systolic Variations in Pressure with Respiration (SPV) Based Model

In case of anaesthetised patients on mechanical ventilation (positive pressure ventilation) the systolic pressure variations with respiration can provide a useful measure for loss in blood volume [63, 64, 76]. The SPV model uses this information embedded in the invasive blood pressure signal (BP/P1) for diagnosing AHV and FCO. The first step in the SPV model is to determine if the patient is breathing with or without positive pressure ventilation. The range of $ETCO_2$ and ICO_2 values for a patient with positive pressure ventilation are generally in the range 0 to 40 for $ETCO_2$ and < 1 for ICO_2. We use these threshold conditions for assessing if the patient is breathing with or without the positive pressure ventilation. For patients on positive pressure ventilation the SPV module calculates the percentage change in SPV from the arterial pressure signal. Any change in percentage SPV over 16% signifies AHV. The sensitivity of SPV module can be changed by changing the alarm level (which is currently 16%), this value will be optimised through offline and online testing so as to obtain accurate diagnosis.

2.3.3. Fuzzy Model

Diagnosing critical conditions during anaesthesia is analogous to detecting a fault in a complex industrial process and these faults can be identified by monitoring the system variables. In anaesthesia management the critical events are monitored by monitoring the physiological signals from the anaesthesia monitor. The anaesthetist diagnoses a critical event by monitoring the available clinical data/waveforms from the screen of the anaesthesia monitor and by forming a mental map of all the clinical information that is available to him/her. The diagnosis generated by the anaesthetist (expert) has an inherent vagueness and uncertainty, fuzzy logic based alarms can mimic the expert diagnoses.

The fuzzy logic module is based on the fuzzy logic based alarm module proposed by Lowe and Harrison, [11, 34, 40, 68]. For an in depth theoretical background on application of fuzzy logic for diagnostic applications refer to [11].

The fuzzy model is subdivided into various fuzzy templates each of which monitors a specific critical condition. The output of this fuzzy templates is a fuzzy membership value, the membership values indicates how well the patients current physiological state is following or deviating from the fuzzy course. The fuzzy template which monitors AHV, for instance will monitor the four input signals (BP, HR, PV and $ETCO_2$) and generate an output membership value. This membership value indicates how well the patient's physiological state is following / deviating from the AHV course. The basis of such a fuzzy course is human knowledge represented as linguistic rules. For example an expert may express the diagnosis for AHV in the form of linguistic statements, as follows (refer to Eq.2.3 and Eq.2.4);

If heart rate increases AND blood pressure decreases
AND Pulse Volume decreases then **2.3**
absolute hypovolaemia is LIKELY.

If heart rate is very high AND blood pressure is low
AND Pulse Volume is very low then **2.4**
absolute hypovolaemia is VERY LIKELY.

Linguistic interpretation of the expert's diagnostic rules cannot be clearly expressed using crisp logic because there is no provision for accommodating degree of agreement/disagreement. For instance in crisp logic heart rate (HR) can be expressed as high (1) or low (0) but the there is no provision for expressing very high HR, very low HR and likely absolute hypovolaemia. However fuzzy logic can be used to complex rules with degree of agreement/disagreement and thereby the uncertainty and vagueness in the expert linguistic rules can be easily programmed in the form of a fuzzy course. Refer to [11] for a conceptual explanation for applying fuzzy theory in fault detection and diagnosis (FDD). The relevant real time algorithms used for generating the anaesthesia related diagnostics are explained in section 2.4.8.

2.4. Real Time Algorithms

Real time algorithms for SAAM were developed with modular programming approach, i.e. each block of algorithm is functionally independent from the other. The incoming signals namely P1/BP, Pleth/PV and $ETCO_2$ are each pre-processed prior to analysing these signals for generating any diagnostic information. In the following stage the signals are further processed to compute specific information like MAP, SPV or HR changes. Figure 2.3.3-1 shows how the incoming data flows through the various algorithms and the logical sequence of the algorithm modules. Each of these algorithm blocks are explained in details in the sub-sections that follow.

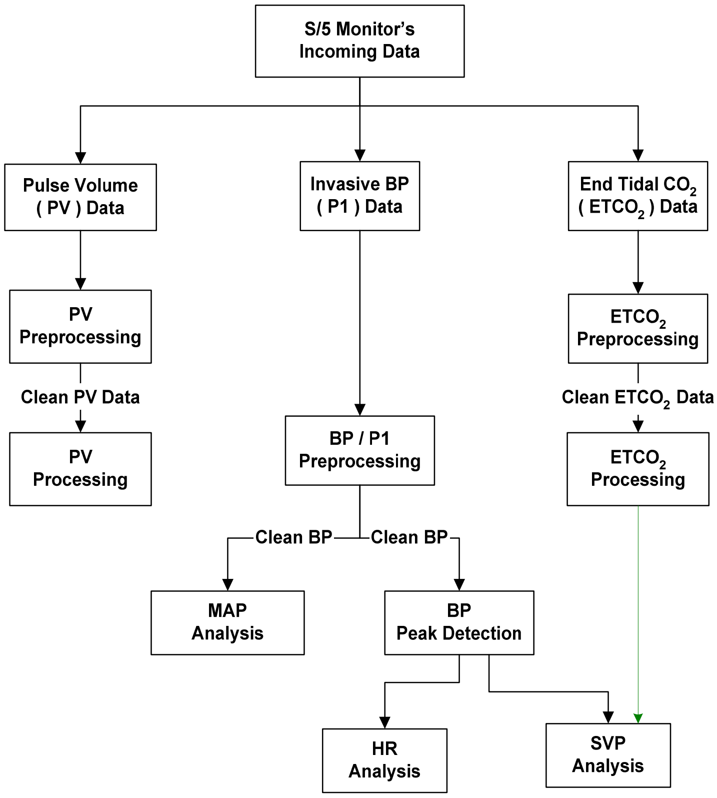


Figure 2.3.3-1. Real-Time Algorithm's Hierarchical Layout.

2.4.1. Pulse Volume/Plethysmography (PV) based Algorithms

2.4.1.1. PV Pre-Processing Algorithm

The PV pre-processing algorithm performs the following tasks sequentially:

1) It performs lowpass filtering of the PV waveform using a lowpass PV-filter. The low pass PV-filter has the following filter specifications. These filter specifications were chosen after analysing the sample data during the algorithm development stage by trial and error so as to attain maximum noise attenuation from the sample data. MATLAB^{TM} Filter Design and Analysis FDA-tool (Filter Design and Analysis Tool) was used to determine the optimal parameter values for the filters applied in this research.
   a. Ω_p ➔ Passband edge frequency = 1 Hz.
   b. Ω_s ➔ Stopband edge frequency = 2 Hz.
   c. δ_p ➔ Passband ripple = 0.6 dB.
   d. δ_s ➔ Stopband gain = 100 dB.
2) It then performs variance based filtering of the PV waveform, variance based filtering algorithm is explained in section 2.2.1.6. The PV variance based filtering algorithm processes a batch of 1000 data samples (i.e. 10-

seconds data) at a time. If the batch has variance which is beyond the normal range then the whole batch of data is removed from the PV-waveform.

3) The resultant PV waveform will have clean PV signal with majority of the low-frequency noise eliminated and all major artifacts removed. We call the resultant PV-waveform '**clean PV**'.

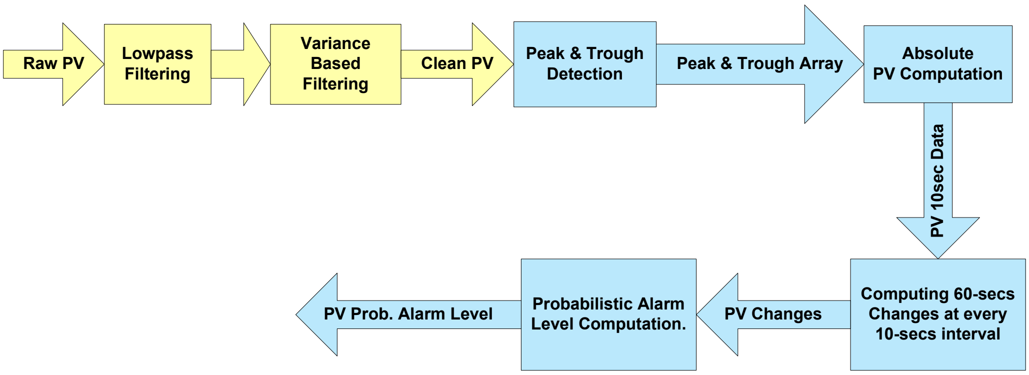


Figure 2.4.1-1. Block diagram for PV-processing algorithm.

2.4.1.2. PV-Processing Algorithm

The PV-processing algorithm processes the clean PV data to obtain the intermittent PV-value (absolute) at a regular interval of 1000 data sample (10-seconds). This is accomplished by first computing the peak and trough value for each PV waveform segment of length 10-seconds and then subtracting the PV trough value from the PV peak value. The PV value so obtained is called 'Absolute PV value'. The PV-Processing algorithm thus generates a series of absolute PV-values at every 10-seconds interval and these values are stored in an array called "PV-10 sec data". Refer to Figure 2.4.1-1 for a graphical illustration.

We then compute the changes in absolute PV over every 60-seconds period. These changes are computed intermittently at every 10-seconds interval and used for computing probabilistic alarm level for the PV. For example if we are currently processing the PV which was recorded with a time stamp of 9:15:50am then we compare the absolute PV value for 9:15:40am (current) with the absolute PV value at 9:14:40am (60 seconds back in time from the current value). Next we compare the absolute PV value for 9:15:50am (current) with the absolute PV value at 9:14:50am (60-seconds back in time from the current value) and so on. This iteration is performed for every data element in the PV-10 sec data array. This computation gives us the change in absolute PV values at every 10-seconds interval (obtained by comparing PV values at every 60-seconds interval).

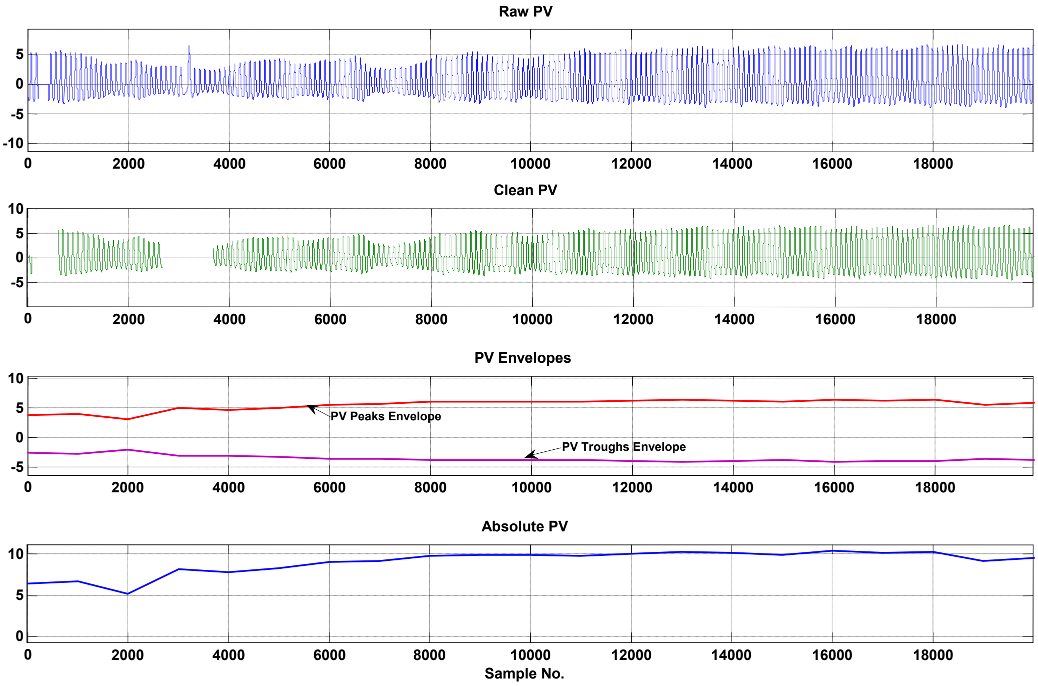


Figure 2.4.1-2. PV processing Algorithms Output.

The above algorithm is implemented in the form of a moving window of length 60-seconds (6000 sample) moving in steps of 10-seconds (1000 samples) as illustrated in Figure 2.4.1-3 below. The resultant 10-second PV data is further analysed by the probabilistic alarm algorithm for computing the probabilistic alarm level of PV-signal.

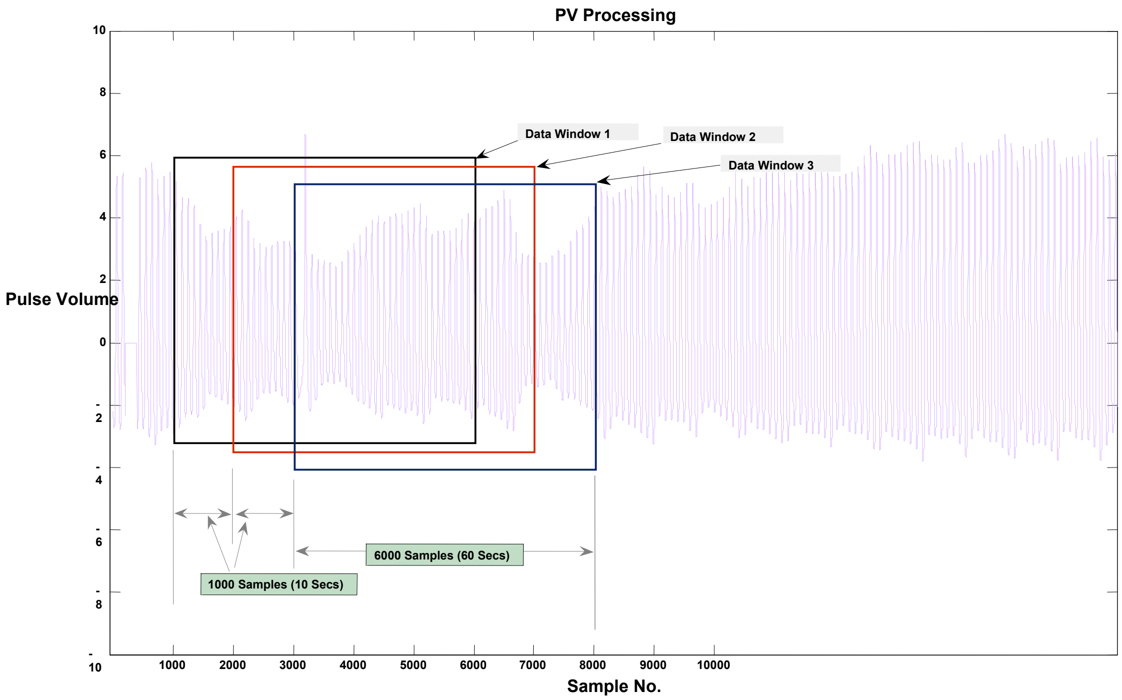


Figure 2.4.1-3. PV Processing by Moving Window.

The probabilistic alarm for PV is generated by comparing the current PV value and the change in absolute PV value over a fixed interval of time (i.e. 60-seconds in the above case) with a cluster of population data. The probabilistic alarm algorithm implements the following mathematical operations for generating the probabilistic alarm level:

1) For each data-point, the current absolute PV (x) is plotted against the corresponding change in PV value (y) over the last 60-seconds time interval. See Figure 2.4.1-4.(a).
2) Plot obtained in step-1 is centred about the origin by subtracting the population average values from each data-point $(x-\bar{x})$ vs. $(y-\bar{y})$ as shown in Figure 2.4.1-4.(b). This is called 'centring of the cluster'. It should be noted that in Figure 2.4.1-4 (b) $(x-\bar{x})$ signifies the PV values normalised about the population average (and not about the patient's average PV value). The PV values for this particular patient are much higher than the population average and hence the $(x-\bar{x})$ values for the data points are all positive.
3) The x and y-coordinates for each data point are then divided by standard deviation of PV and standard deviation (SD) and change in PV respectively $(x-\bar{x})/s_x$ vs. $(y-\bar{y})/s_y$, as shown in Figure 2.4.1-4.(c). These standard deviation values are obtained from the population data. The average (SD)

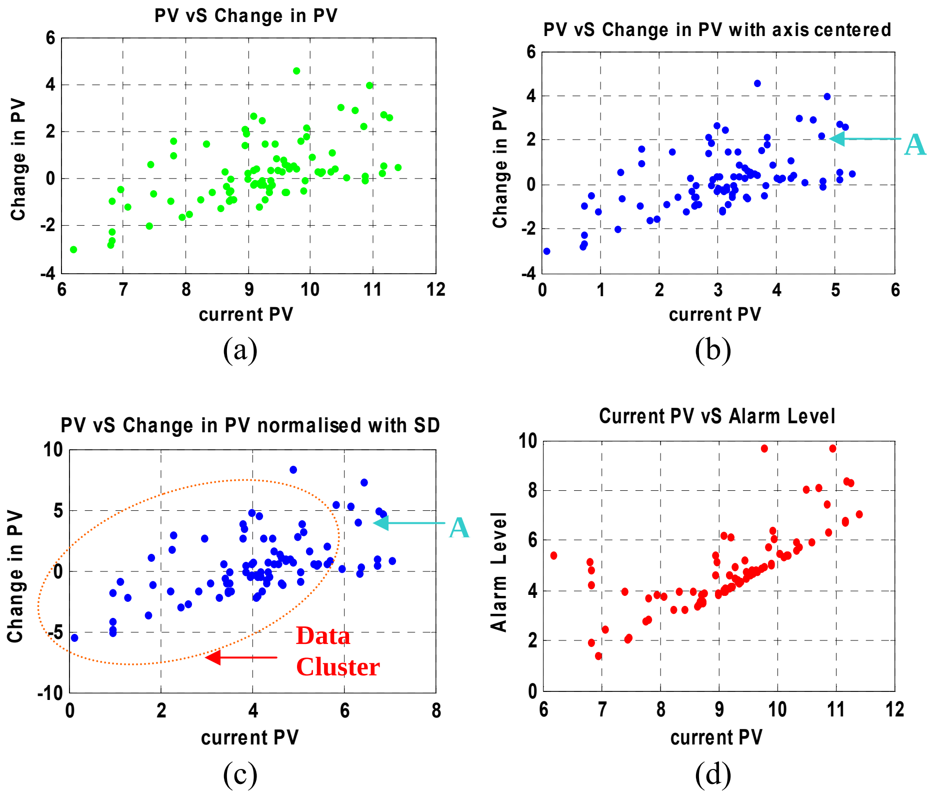
PV for the population data was 6.096 (0.75), the average (SD) change in PV is 0.01 (0.55). The distance of any data point from the origin gives the probabilistic alarm level corresponding to that particular point. For instance consider the point A in Figure 2.4.1-4.(c) with current PV of 5, and a recent change of 2.1.

(Current PV – Average) / SD = (5-6.096)/0.75 = -1.46; value on x-axis
(Change in PV – Average) / SD = (2.1-0.01)/0.55 = 3.8; value on y-axis

4) The probabilistic alarm level for each data-point in Figure 2.4.1-4.(c) is computed by calculating the corresponding distance from the origin. The values above are combined using Pythagoras theorem for computing the overall distance of point A from the mean value in terms of standard deviations, see Eq.2.5. The alarm levels so obtained are plotted in Figure 2.4.1-4.(d).

$$\sqrt{-1.46^2 + 3.8^2} = 4.07 \text{ SD from the mean} \qquad 2.5$$

Here point A represents a single data-point on the PV-waveform. Thus the integration of the probabilistic alarm levels for all the points on a PV-signal segment (say of length 10-seconds) gives the PV probabilistic alarm for the 10-seonds waveform segment.



(a) (b)

(c) (d)

Figure 2.4.1-4. PV probabilistic alarm processing.

2.4.2. Blood Pressure (BP / P1) based Algorithms

The BP pre-processing algorithm performs the following operations sequentially:

1. It performs lowpass filtering of the BP waveform using a lowpass BP-filter. The low pass BP-filter has the following filter specification.
    a. Ω_p ➔ Passband edge frequency = 1 Hz.
    b. Ω_s ➔ Stopband edge frequency = 2 Hz.
    c. δ_p ➔ Passband ripple = 0.0345 dB.
    d. δ_s ➔ Stopband ripple = 0.00001 dB.
2. It then performs variance based filtering of the BP waveform as explained in section 2.2.1.6. The BP variance based filtering algorithm processes a batch of 1000 data samples (i.e.10-seconds data) at a time. If the batch has variance which is beyond the normal range then the whole batch of data is removed from the BP-waveform.
3. The resultant BP waveform will have clean BP signal with majority of the low-frequency noise eliminated and all major artifacts removed. We call the resultant BP-waveform ‘**clean BP**’.

2.4.2.1. MAP Processing Algorithm

The mean arterial pressure (MAP) is obtained by simply computing the mean of the clean BP waveform over a fixed time interval (window of data). We compute the MAP value at every 10-seconds (i.e. 1000 samples at 100 Hz) interval by using a moving window of length 60 seconds (i.e. 6000 samples). This is graphically illustrated in Figure 2.4.2-1. Figure 2.4.2-2 shows the MAP envelope obtained after applying MAP algorithm to a clean BP waveform. Thus we get a series of MAP values at 10-seconds intervals which are further processed to obtain the changes in BP over 60-seconds interval, computed intermittently at every 10-seconds interval. Functionally this is similar to PV-processing algorithm except the methodology which computes the MAP value from the clean BP waveform.

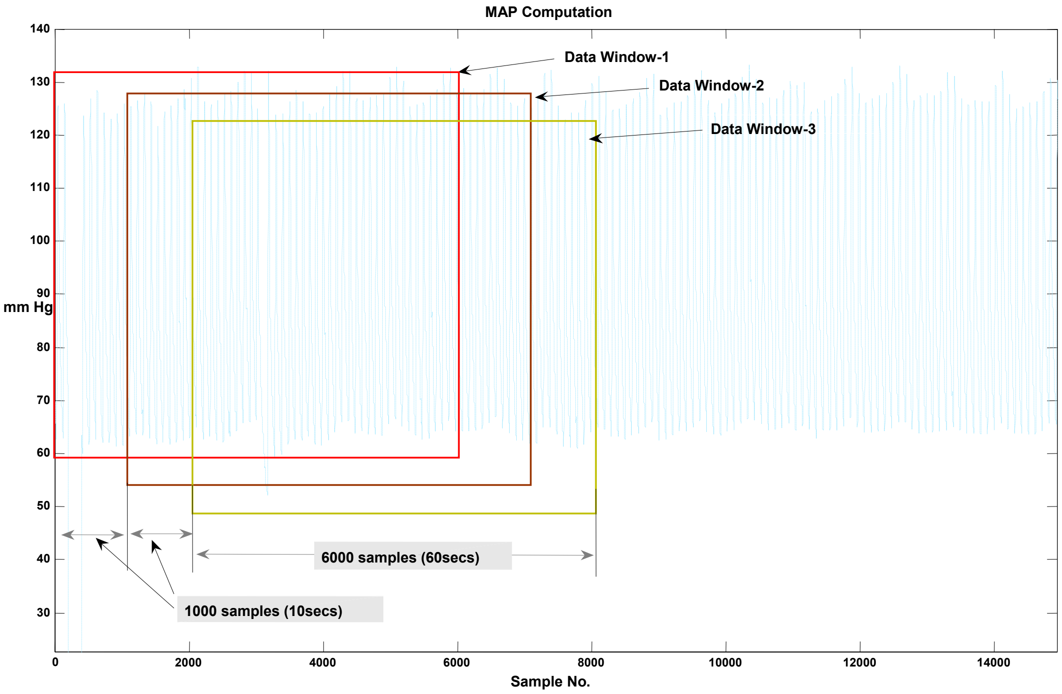


Figure 2.4.2-1. MAP computation by moving data window.

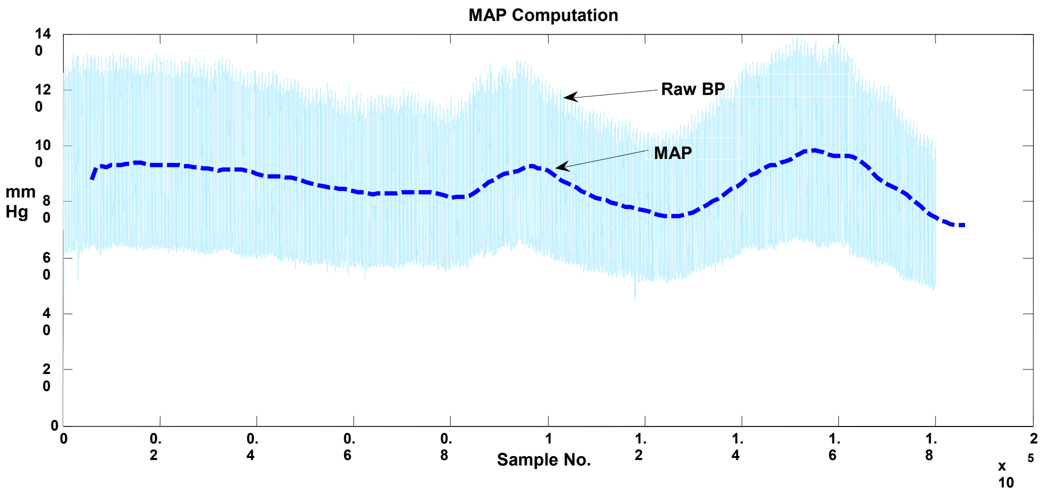


Figure 2.4.2-2. MAP Envelope obtained by applying MAP algorithm.

We then compute the changes in MAP over every 60-seconds period. These changes are computed intermittently at every 10-seconds interval and used for computing probabilistic alarm level for MAP. The MAP probabilistic alarm algorithm is functionally similar to the PV probabilistic alarm algorithm. The MAP processing algorithm can be summarised in the following logical steps:

1. The MAP value is computed from the clean BP waveform at every 10-seconds interval.
2. The MAP values at 10-seconds interval are then evaluated to compute the change in MAP over a 60-seconds period, this value is intermittently obtained for every 10-second interval using a 60-seconds data window advancing in increments of 10-seconds.
3. The changes in MAP over 60-seconds are stored in an array and used for computing the probabilistic alarm level for MAP. This step is identical to the PV probabilistic alarm computation (refer section 2.4.1.2). Figure 2.4.2-4 illustrates the probabilistic alarm computation for MAP. It should be noted that in Figure 2.4.2-4 (b) $\left(x-\bar{x}\right)$ signifies the MAP values normalised about the population average (and not about the patient's average MAP value). The MAP values for this particular patient are much higher than the population average and hence the $\left(x-\bar{x}\right)$ values for the data points are all positive.

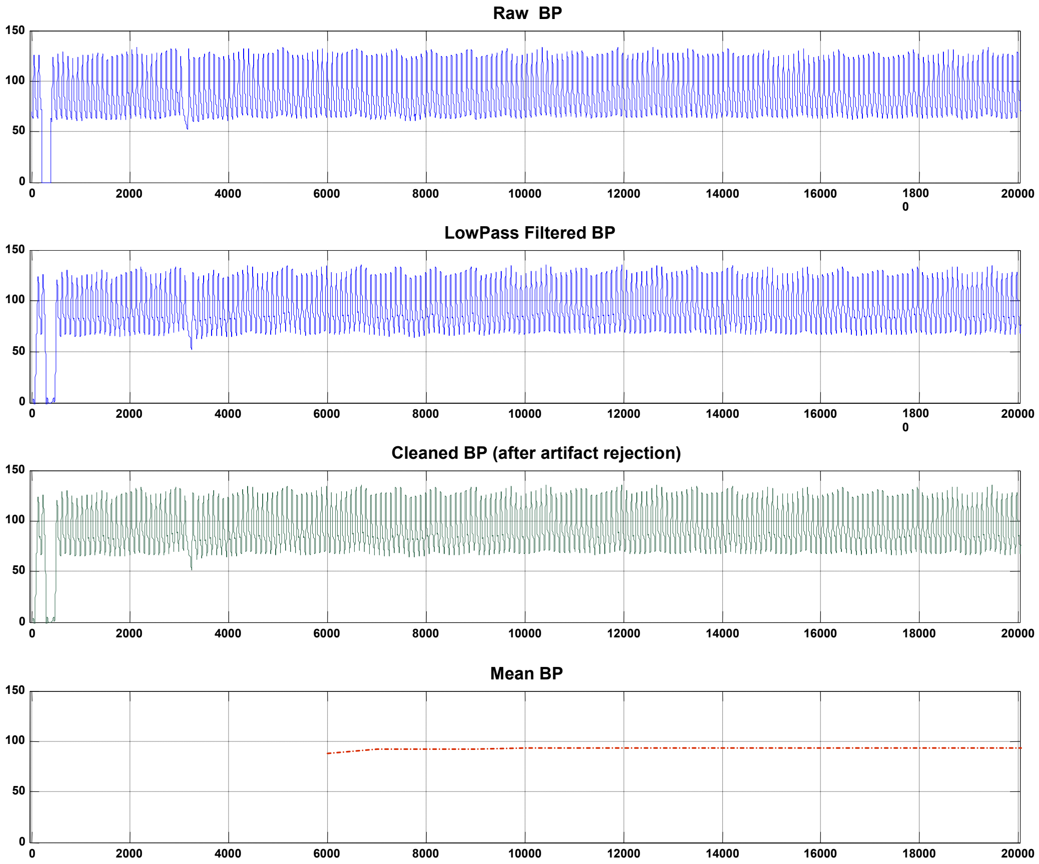
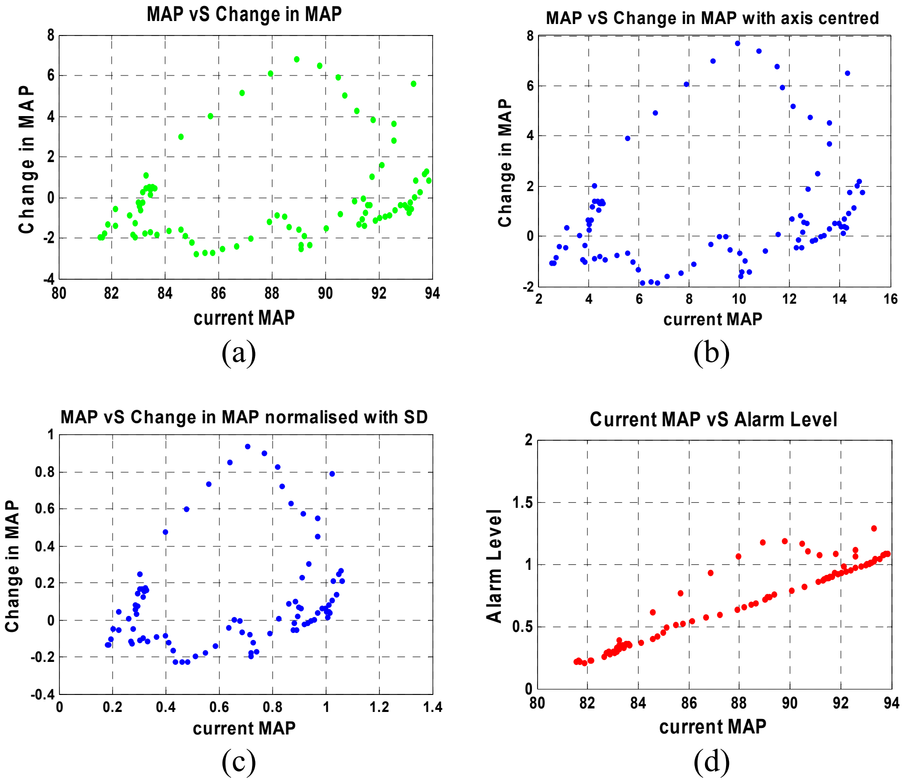


Figure 2.4.2-3. MAP processing algorithms output.



(a)

(b)

(c)

(d)

Figure 2.4.2-4. MAP probabilistic alarm processing.

2.4.3. Systolic Pressure Algorithm

Systolic pressure is obtained from the clean BP waveform by computing the average value of the peak BP values for every 10-seconds long batch of the data. Thus the first step in the systolic pressure algorithm is to obtain the peak envelope of the BP waveform. This is accomplished by using a BP peak detection algorithm which is functionally similar to the peak detection used for PV peak detection. The criterion for a BP peak to qualify as a genuine peak when implementing the BP peak detection algorithm is listed below:

1. The peak BP data point should be local maxima in the current data window.
2. There should be a change in slope from positive slope (rising) to negative slope (falling).
3. There should be no previously detected peak at any of the previous 50 data points.
4. The current BP value should be within the threshold range 50 to 220 mm Hg.

The resultant BP peak envelope is then further processed to obtain the following information:

a) **Heart Rate (HR) value**: The HR value is obtained by counting the number of systolic peaks in every 60-seconds data batch. The HR values obtained are expressed in beats per minute. It also checks if there is any gap in the current cleaned BP-data batch. If any gap is detected than the HR value for that particular data batch is recorded as NaN value (Not a number).

$$\boxed{\text{HR} = \text{No. of systolic peaks (in 60-seconds data window)}}$$

b) **SPV value**: Systolic variation in BP due to respiration (**SPV**) value is obtained by computing the difference between the maximum and minimum values in the peak BP envelope over each 30-seconds long data segment. Figure 2.4.3-1 shows the logical flow of data between the various BP based algorithms used in SAAM.

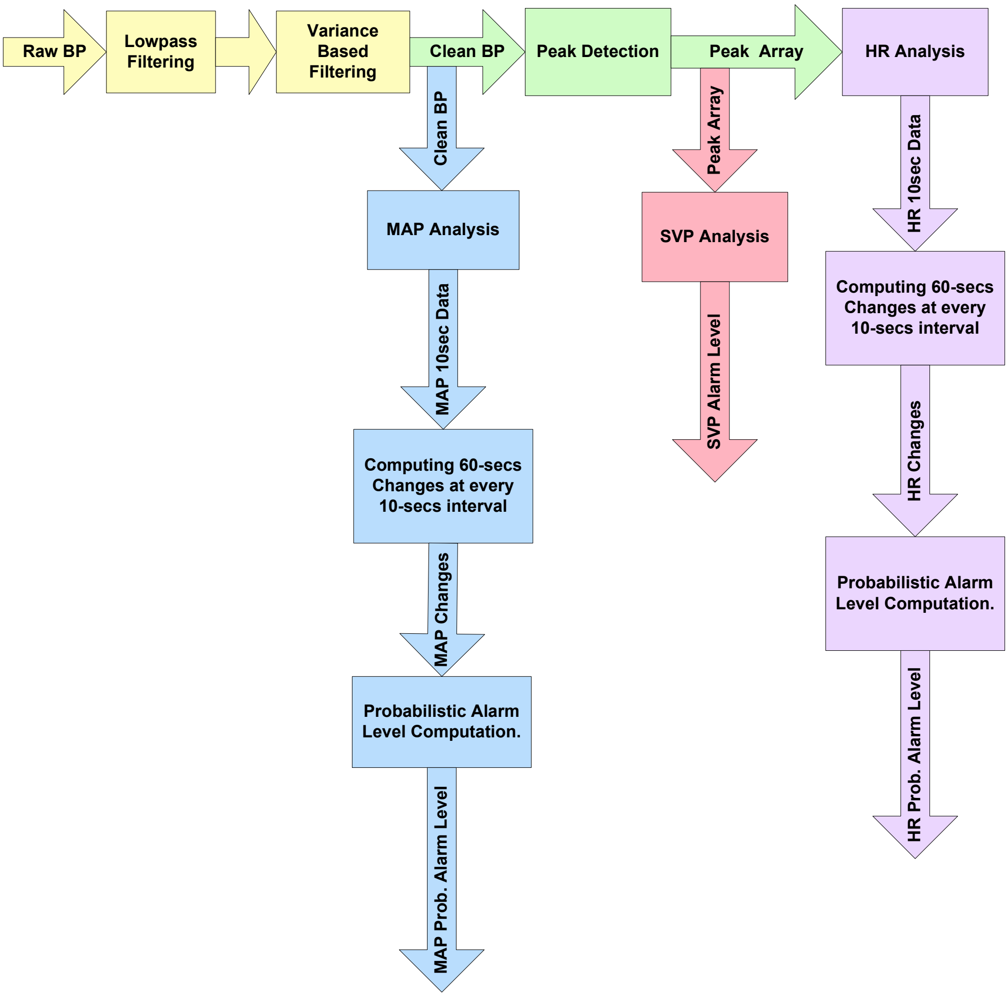


Figure 2.4.3-1. Block diagram for BP (P1)-processing algorithm.

2.4.4. HR Algorithm

The HR algorithm computes the HR values from the peak BP envelope by counting the number of BP peaks in each 10-seconds data batch. The resultant HR values are labelled “10-sec HR data”. The 10-sec HR data is then further processed using the moving data window approach identical to PV and MAP processing to obtain the changes in HR values over 60-seconds duration, computed intermittently at every 10-seconds interval. The changes in HR computed in the previous step are then used for computing the probabilistic alarm level of HR. The probabilistic alarm algorithm for PV, MAP, HR and ETCO$_2$ is the same, refer to section 2.4.1.2 for detailed description on the probabilistic alarm algorithm.

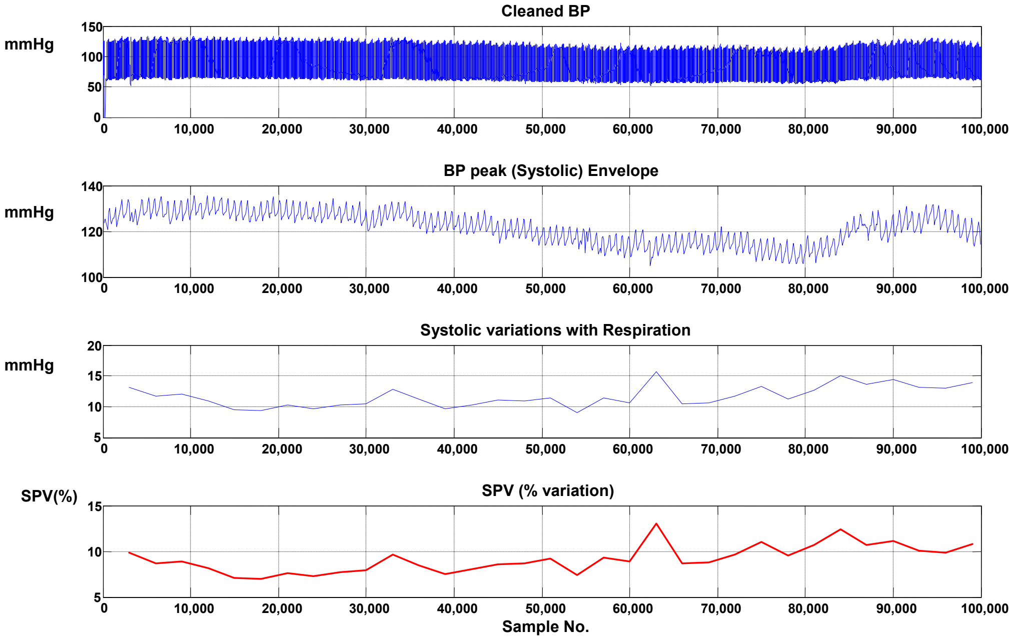


Figure 2.4.4-1. Output of the SPV algorithm.

2.4.5. SPV Algorithm

SPV represents the change in the BP peak (Systolic BP) values due to respiration. The SPV algorithm computes the SPV value from the BP peak envelope simply by computing the difference between maximum and minimum BP value for each 30-seconds data window. The SPV is then expressed as percentage of the maximum Systolic BP in the current 30-seconds data window. The SPV algorithm updates the display module with a new SPV (%) value after every 30-seconds interval. Figure 2.4.4-1 above shows the output of the SPV algorithm.

2.4.6. ETCO$_2$ Algorithm

The raw ETCO$_2$ signal is the cleanest signal out of the three input signals that are used in SAAM. Thus the ETCO$_2$ signal does not require any pre-processing. The raw ETCO$_2$ waveform is processed to obtain the maximum and minimum values for each 10-second ETCO$_2$ waveform segment. The maximum value for each 10-seconds segment gives the ETCO$_2$ value and the corresponding minimum value identifies the inspired CO$_2$ (ICO$_2$). A limiting threshold condition is enforced on computed ETCO$_2$ value to ensure that the raw ETCO$_2$ is genuine. The computed ETCO$_2$ values (10-seconds ETCO$_2$ data) have to be greater than a threshold value of 3 to qualify as a genuine ETCO$_2$ signal. The 10-seconds ETCO$_2$ data is then further processed using the 60-seconds moving window algorithm for computing the changes in $ETCO_2$ values over 60-seconds duration, computed intermittently at every 10-seconds interval. Next, the changes in $ETCO_2$ computed in the previous step are used for computing the probabilistic alarm level for $ETCO_2$. A copy of the array with 10-seconds data for $ETCO_2$ and ICO_2 is also passed to the SPV algorithm where these values are used for determining if patient is breathing with or without mechanical ventilation.

2.4.7. Probabilistic Module Algorithm

The probabilistic module basically integrates the four probabilistic alarm levels viz. MAP alarm level, HR alarm level, PV alarm level and $ETCO_2$ alarm level to generate an integrated probabilistic alarm level which is displayed in the display module. The probabilistic module simply takes an average of the four alarm levels and updates the display module after every 10-seconds.

2.4.8. Fuzzy Module Algorithm

The fuzzy logic algorithm executes a hierarchy of fuzzy functions for evaluating the membership values for each data-set over the specified fuzzy course. The hierarchical structure for the AHV fuzzy module is illustrated in Figure 2.4.8-1 below. The data flow to the fuzzy functions and flow of the membership values between the fuzzy functions is shown in Figure 2.4.8-2-(a) below. The flow chart illustrated in Figure 2.4.8-2-(b) shows the analytical fuzzy steps being executed by the corresponding functional blocks in the fuzzy algorithm.

2.4.8.1. **Fuzzification:**

A fuzzy interval is the basic building block of a fuzzy template. A fuzzy interval performs the task of mapping each data point to the fuzzy domain. A trapezoidal membership function is used for evaluating the membership value of each data sample. The fuzzification of the data samples is achieved by:

1) Evaluating the membership value of the data samples to the fuzzy signal level interval and the fuzzy time-change interval.
2) The membership values obtained in the previous step are than evaluated by the fuzzy course function and a fuzzy course membership value is generated.

A fuzzy course membership is thus generated for each data-sample. This is graphically illustrated in Figure 2.4.8-3 above.

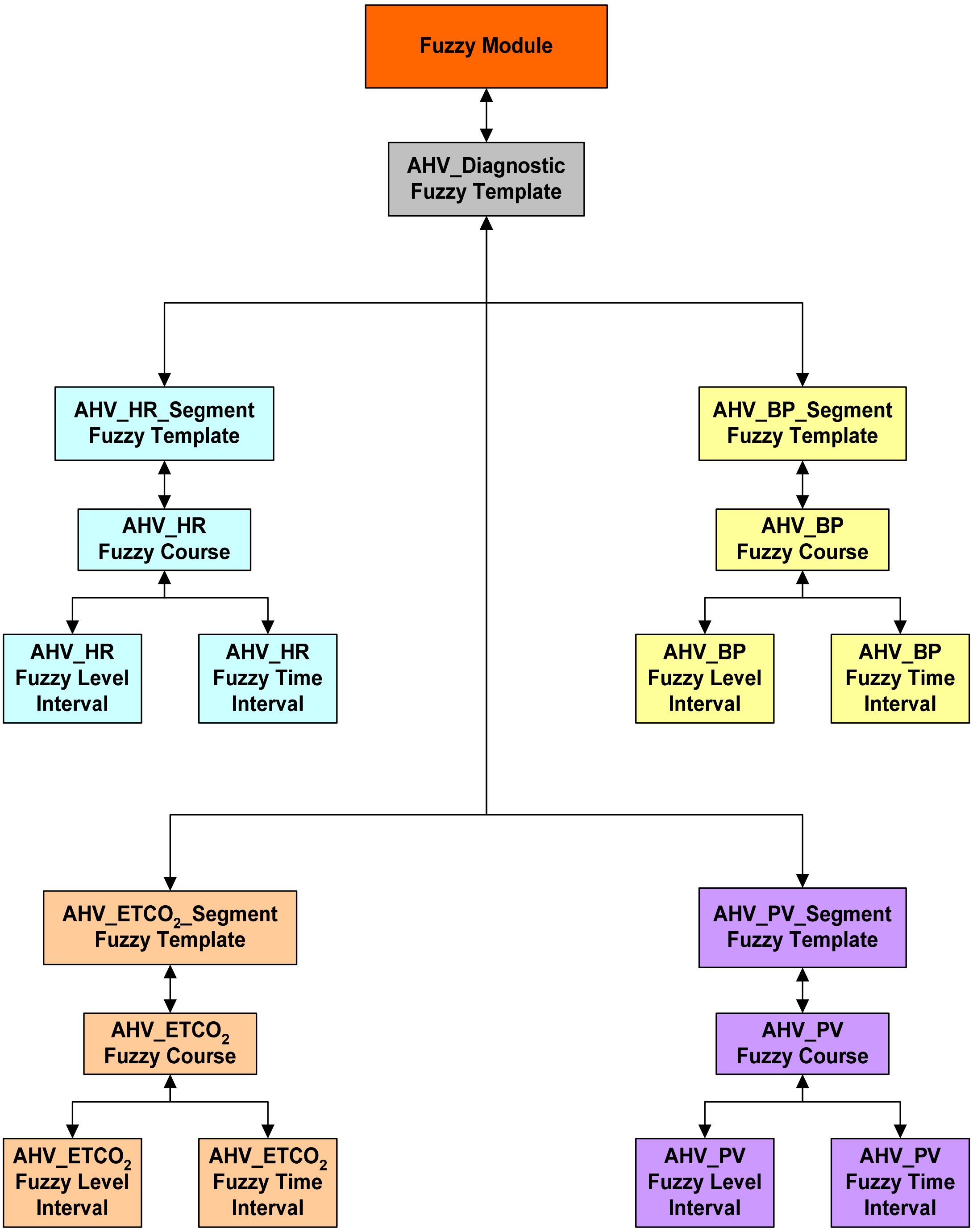

Fuzzy Module
AHV_Diagnostic Fuzzy Template
AHV_HR_Segment Fuzzy Template
AHV_HR Fuzzy Course
AHV_HR Fuzzy Level Interval
AHV_HR Fuzzy Time Interval
AHV_BP_Segment Fuzzy Template
AHV_BP Fuzzy Course
AHV_BP Fuzzy Level Interval
AHV_BP Fuzzy Time Interval
AHV_ETCO$_2$_Segment Fuzzy Template
AHV_ETCO$_2$ Fuzzy Course
AHV_ETCO$_2$ Fuzzy Level Interval
AHV_ETCO$_2$ Fuzzy Time Interval
AHV_PV_Segment Fuzzy Template
AHV_PV Fuzzy Course
AHV_PV Fuzzy Level Interval
AHV_PV Fuzzy Time Interval

Figure 2.4.8-1. AHV Fuzzy Logic Algorithms' Hierarchical Layout.

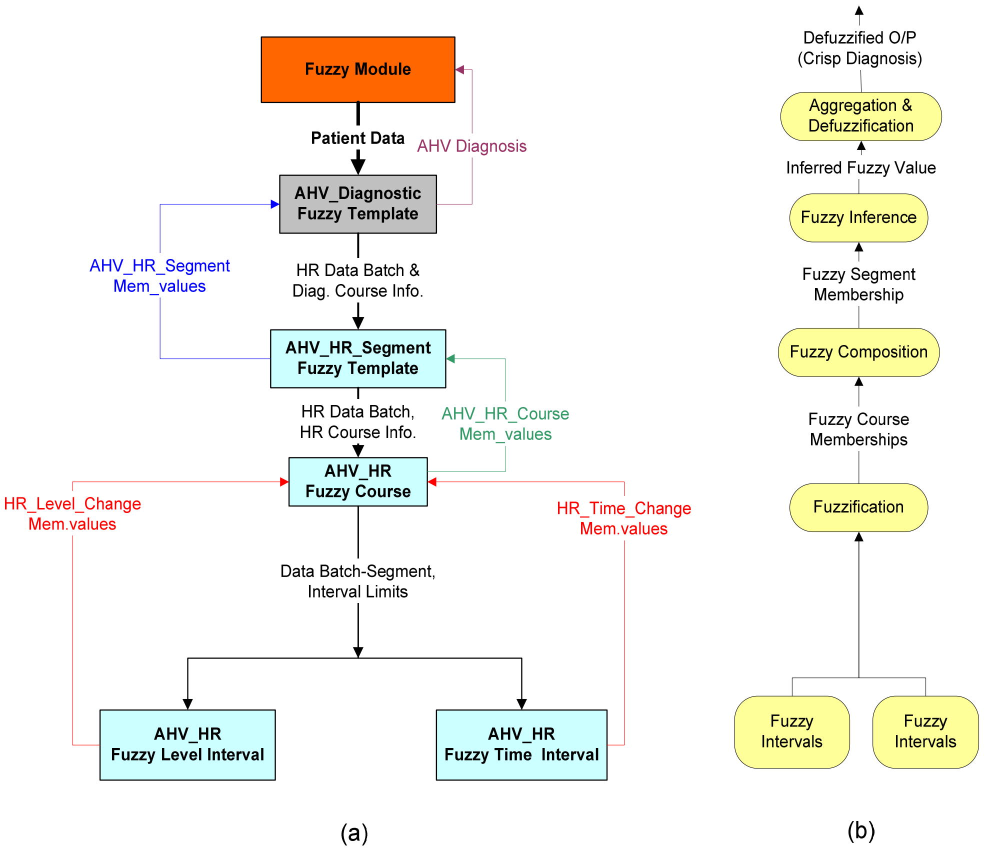

Fuzzy Module
Patient Data
AHV Diagnosis
AHV_Diagnostic Fuzzy Template
AHV_HR_Segment Mem_values
HR Data Batch & Diag. Course Info.
AHV_HR_Segment Fuzzy Template
HR Data Batch, HR Course Info.
AHV_HR_Course Mem_values
AHV_HR Fuzzy Course
HR_Level_Change Mem.values
HR_Time_Change Mem.values
Data Batch-Segment, Interval Limits
AHV_HR Fuzzy Level Interval
AHV_HR Fuzzy Time Interval
(a)
Defuzzified O/P (Crisp Diagnosis)
Aggregation & Defuzzification
Inferred Fuzzy Value
Fuzzy Inference
Fuzzy Segment Membership
Fuzzy Composition
Fuzzy Course Memberships
Fuzzification
Fuzzy Intervals
Fuzzy Intervals
(b)

Figure 2.4.8-2. AHV_HR Algorithm's Flow Chart.

(a) Fuzzy AHV_HR algorithm, (b) Fuzzy methodology flowchart.

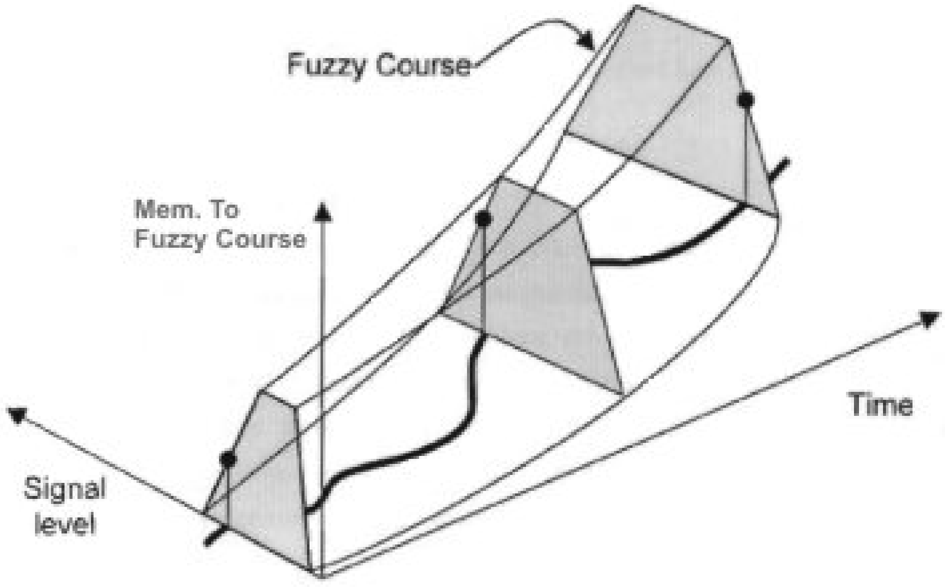


Figure 2.4.8-3. Membership of a signal waveform (data-set) to a fuzzy course is aggregation of membership of data-samples in the signal waveform to individual slices of the fuzzy course (trapezoidal function) [11].

2.4.8.2. **Fuzzy Composition:**

The fuzzy course memberships for all the data-samples in a waveform segment (data-set) are then aggregated by using a fuzzy MIN or fuzzy MAX operator to generate a fuzzy segment membership value. This is called ‘Fuzzy Composition’. Similarly fuzzy segment membership values are computed for AHV_BP_Segment, AHV_PV_Segment and AHV_ETCO$_2$_Segment.

2.4.8.3. **Fuzzy Inference:**

The four Signal_Segment membership values are then evaluated using fuzzy inference rule, for example the following inference rule (Eq.2.6) is used by the AHV_Diagnostic module.

If (AHV_HR_Segment is high) AND
(AHV_BP_Segment is low) AND **2.6**
(AHV_PV_Segment is low) then (absolute hypovolaemia is LIKELY).

The fuzzy statements (enclosed within brackets) in Eq.2.6 before ‘then’ are called antecedents and the fuzzy statement after ‘then’ is called the consequent. The value of the consequent depends on the overall value of the antecedent. Eq.2.6 forms a fuzzy relation which maps the four Signal_Segment membership values from antecedent domain (fuzzy) to consequent domain (fuzzy). The resultant fuzzy value is called inferred fuzzy value.

2.4.8.4. **Fuzzy Aggregation**

Fuzzy modules can have more than one fuzzy inference rule for generating the same diagnosis. This improves the accuracy of the fuzzy module by providing redundant diagnosis for the same critical event. For instance we can have multiple inference rules for obtaining multiple AHV diagnoses. These diagnoses are in the form of fuzzy values and need to be aggregated prior to performing defuzzification. However in SAAM we are implementing one inference rule per diagnosis and therefore fuzzy aggregation is not required.

2.4.8.5. **Defuzzification**

The fuzzy aggregated fuzzy value obtained after defuzzification is then mapped onto the crisp domain by using a defuzzification function. For instance if fuzzy inference value (after defuzzification) is high then the corresponding crisp diagnosis will be AHV very likely. Thus defuzzication maps the fuzzy inference value onto a crisp domain. The output obtained from the defuzzication function is generally a linguistic statement, but the resultant can also be expressed in the form of a number.

The fuzzy module algorithm was programmed with LabVIEW^{TM} graphical script and has a similar structure as explained in this section of the text.

2.5. Algorithms Overview and Description

All the algorithms described above are developed using MATLAB^{TM} programming language. Data simulation and analysis can be easily carried out in MATLAB^{TM}. All the algorithms are tested repetitively so as to optimise their performance and accuracy. Offline data previously collected from the S/5 monitor was used for data simulation and analysis in MATLAB^{TM}. Each of these algorithms are written in separate MATLAB^{TM} (**.m) files. Offline data simulation and algorithm testing gave satisfactory outputs.

2.6. Final Developments

Since the application of SAAM in real-time needs data communication with external devices, a more robust development platform, LabVIEW^{TM} was chosen. The following factors influenced the selection of LabVIEW^{TM} as the development tool for developing the real-time version of SAAM:

1. The MATLAB^{TM} algorithm can be easily reused in LabVIEW^{TM} using the Mathscript block. This ensures that the accuracy of the algorithm is not affected due to migration across different programming platforms.
2. LabVIEW^{TM} provides user friendly communication tools which were used for acquiring data in real time.
3. LabVIEW^{TM} provides a user friendly tool for building executable version of the final application, thus eliminating the need of installing LabVIEW^{TM} application on the test bench in the operating theatre where RT-SAAM was tested in real time.
4. The modular programming approach can be easily implemented in LabVIEW^{TM} by using the hierarchical Virtual Instruments (VIs) and sub-VIs.

2.7. Chapter Summary

SAAM derives physiological waveform data from the S/5 monitor. The waveform data is filtered, processed, analysed using modular algorithms which process the waveform data to generate diagnostic information. There are three diagnostic modules in SAAM viz. probabilistic module, fuzzy module and SPV module which generate three different alarm levels which are displayed on a display module. The display is updated regularly and provides the front-end information to the end-user (i.e. the anaesthetist). These algorithms were developed in MATLAB^{TM} and were later reused in LabVIEW^{TM}. The following chapter discusses how these algorithms operate and interact in real time to generate the diagnostic alarm levels and interact with the anaesthetist.

Chapter 3 Real Time System Development

3.1. Introduction

The ultimate goal of this research project was to develop a prototype alarm system which could be tested and validated through clinical testing. Diagnostic algorithms discussed in the previous chapters were used as foundation blocks for constructing a Real Time version of SAAM (**RT-SAAM**). A major hurdle encountered during the real-time system development was the acquisition of data from the S/5 Datex-Ohmeda anaesthesia monitor without affecting the existing setup in the operating theatre. The serial and TCP communication protocols which were implemented to overcome this issue are described in section 3.2 of this chapter.

The real-time system was programmed by taking into account the communication issue and was built using the state-engine / state-machine technique. Thus SAAM was developed using the state-engine approach where the system control is passed from one state to another state as data progresses through the various stages in the system. The state-engine approach neatly integrates the different blocks of the real-time algorithm whilst maintaining the integrity of individual algorithms. The state-engine thus promotes a lucid modular programming approach for an otherwise complex system. Following sections in this chapter explain the real time system development and the various algorithms which were employed for implementing the real-time state engine in SAAM.

3.2. Data Acquisition

In order to comprehend how SAAM communicates with S/5 monitor and to recognize the problems that were involved in testing SAAM it is essential to understand the existing data collection setup in the operating theatre (refer to Figure 2.4.8-1). The computer labelled ‘IDAS’ is a data logging computer used by the Auckland City Hospital’s computerised database. The S/5 monitor continuously relays the waveform data and the physiological data through a single serial port at the back of the S/5 monitor. The IDAS system acquired the patient data from the S/5 monitor via the serial port and saved it into the patient record keeping system on the hospital’s server. To accomplish this serial communication between S/5 and IDAS various handshake signals were continuously exchanged between S/5 monitor and the IDAS system.

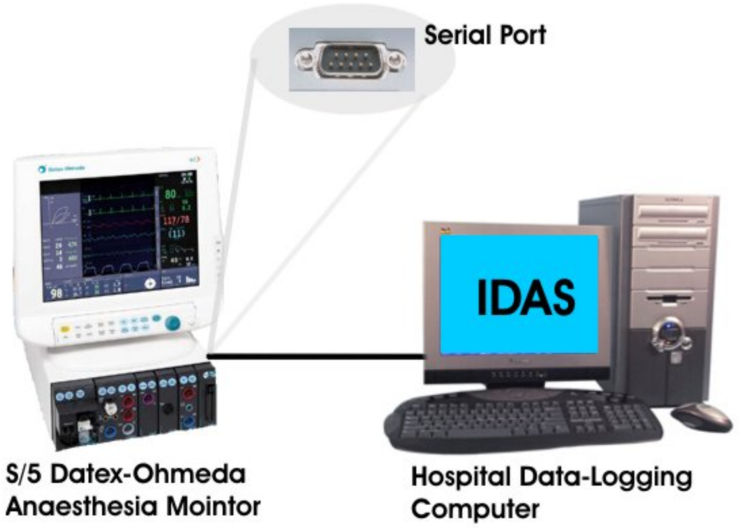


Figure 2.4.8-1. Original data collection setup in the operating theatre (Setup-A).

For testing the prototype system (SAAM) it was crucial that the inflow of data to IDAS was maintained whilst simultaneously acquiring and feeding the data from S/5 to SAAM. The Following communication setups were studied for data acquisition and testing of SAAM without affecting the setup-A described in Figure 2.4.8-1.

3.2.1. Using Different Data Channels for IDAS and SAAM

The technical manual for Datex-Ohmeda S/5 monitor was referred to for investigating alternative ports at the back of the S/5 monitor which could be used for data collection and testing. Datex-Ohmeda, GE technical support was contacted to investigate the availability of using alternate communication channel for acquiring the data signals for testing SAAM. Through this exercise it was learned that S/5 monitor relays all the data signals only via a single serial port and has no other provision for data acquisition. Thus it became evident that the same serial port had to be used for data acquisition (which was currently occupied by IDAS). This required a data relaying solution which could acquire the data from the serial port on the S/5 monitor and simultaneously relay the acquired data to SAAM and IDAS. Therefore other options were tried considering the new technical requirements for data collection setup.

3.2.2. Implementing Y-Splitter Serial Cable (Setup-B)

This setup implemented a Y-splitter serial cable (illustrated in Figure 3.2.2-1) with one serial connector at one end (connected to S/5) and two serial connectors at the other end (connected to IDAS and SAAM). However a single serial port can exchange handshake signals only with one device at a time. In this case the serial port was receiving handshake signals from two devices and therefore resulted in a hardware conflict. Thus setup-B did not meet the objective.

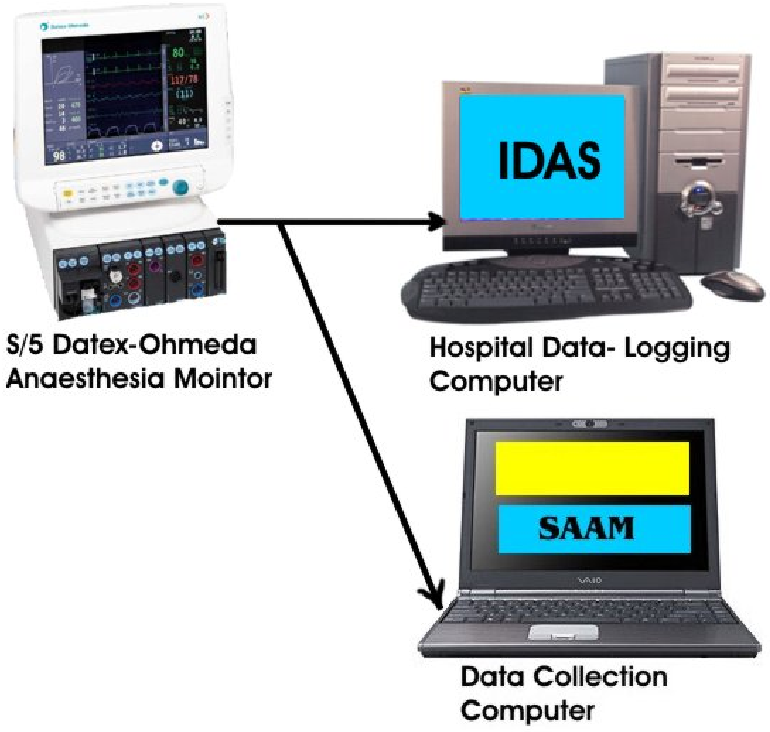


Figure 3.2.2-1. Serial Communication using a Y-splitter serial cable (Setup-B).

3.2.3. Virtual Serial Splitter (Setup-C)

Virtual splitter is a software based approach towards resolving the hardware conflict caused in the previous setup (setup-B). Figure 3.2.3-1 below graphically illustrates how setup-C operates. A virtual serial splitter is a computer application which assists multiple devices to simultaneously communicate with a single serial port, it does so by giving each device a turn to communicate with the serial port. A virtual splitter virtually splits the serial port between several devices by handling the handshake signals on behalf of each device [77].

A virtual serial splitter was tested for data collection with SAAM but it failed as IDAS did not receive all the requested data from the S/5 monitor.

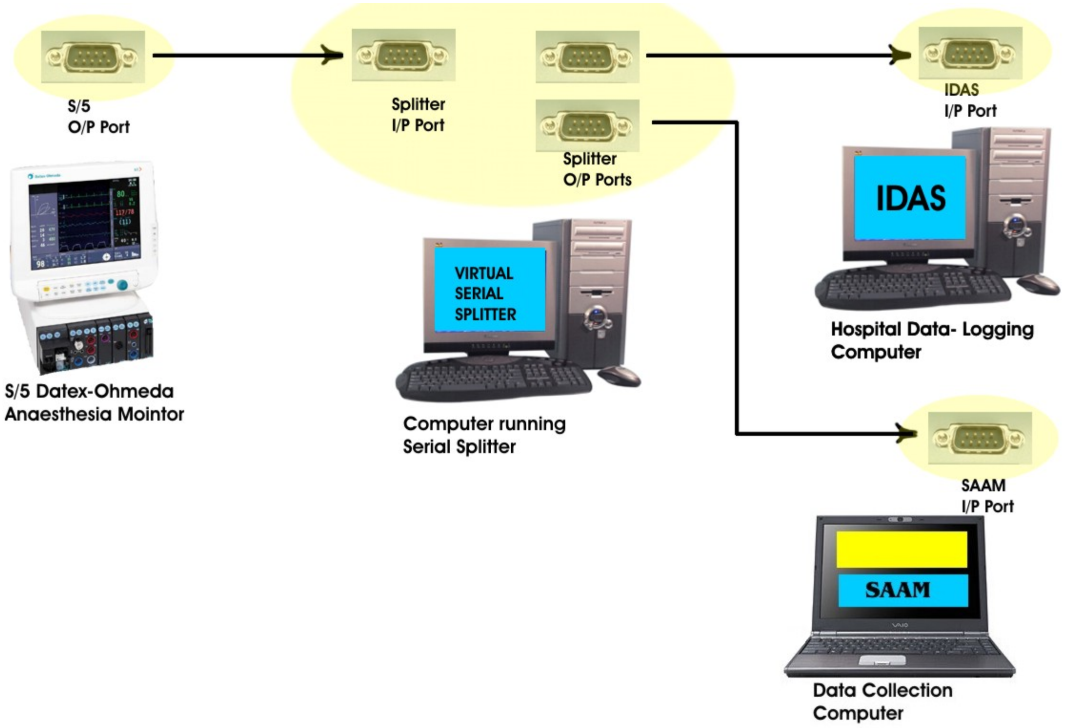


Figure 3.2.3-1. Virtual Serial Splitter (Setup-C).

3.2.4. Data Collection using S/5 Collect (Setup-D):

S/5 Collect is a data collection software from GE Healthcare (the manufacturer of S/5 monitor). S/5 Collect however can only collect data from the S/5 monitor into the data collection computer, the acquired data can be saved into a digital data file which can be used for offline analysis. It has no provision for relaying data to any other device or application and thus it could not be used for real time data collection and testing. In the past S/5 collect was used for collecting data from the anaesthesia monitor but then the serial port was not occupied by IDAS and this data was collected for offline testing only. Some of the data which was collected using S/5 collect in the past was used for offline testing of SAAM.

3.2.5. Communication using DOMonitor Application (Setup-E)

DOMonitor is a JAVA^{TM} based data collection application developed by Dr. Andrew Lowe, Ilixir Limited. Originally DOMonitor was used for acquiring data from the S/5 monitor, saving acquired data to a digital file and simultaneously relaying the data over another serial port. The digital file saved by DOMonitor can be used for offline analysis. The hardware setup for SAAM when using DOMonitor for data collection is illustrated in Figure 3.2.5-1.

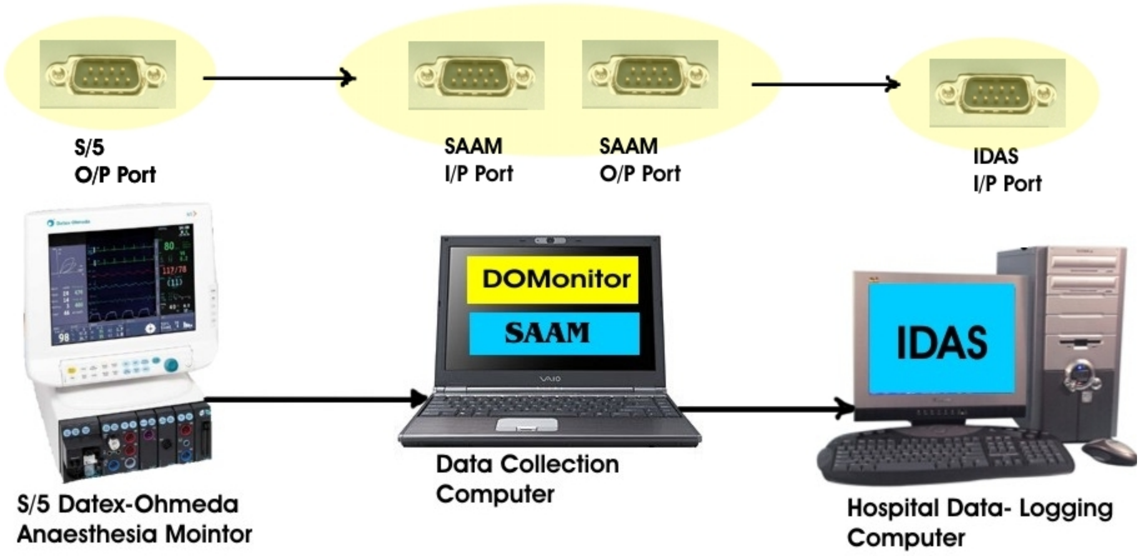


Figure 3.2.5-1. Communication using DOMonitor Application (Setup-E).

DOMonitor had to be modified so that the acquired data could be relayed to the real-time version of SAAM developed using LabVIEW^{TM}. DOMonitor served as a very handy tool for testing SAAM as it performed the following tasks simultaneously which smoothed the whole testing process;

1) Acquires data from the S/5 monitor.
2) Relays acquired data to IDAS over another serial port on the back of the data collection computer.
3) Relays the required data signals to SAAM over a Transmission Control Protocol (TCP) port.
4) Saves the selected waveform data to a digital file which can be accessed in offline mode for retrospective analysis.

DOMonitor runs on the same data collection computer which runs SAAM. During the trial runs DOMonitor successfully communicated with both SAAM and IDAS. The following sections elucidate the analytical grouping and interactions of the different modules used in SAAM and explain the state-engine based algorithm which was developed using LabVIEW^{TM}.

3.3. Real Time System Overview

SAAM was required to execute the following tasks in real time:

1.) TCP-reading: Read the data being relayed by DOMonitor over the TCP port. These data were in the form of data packets which had to be retrieved, identified and sorted into different data buffers before initiating data processing.

2.) Checking if sufficient data had been read into the input buffer.

3.) If sufficient data were read, the control would activate the data processing state which pre-processes the input data and then further processes it to generate the diagnostic alarms.

4.) Data processing state then passes the control to display state which updated the SAAM's front panel with the most recent diagnosis and generates a voice prompt if necessary.

5.) This operation would be continued in a closed loop, until the user exited SAAM.

6.) Comment Logging: At the end of each data-collection session when the user exited SAAM, the user comments would be logged into a file for retrospective analysis of SAAM's performance.

3.3.1. TCP Read

The data sent by DOMonitor to SAAM was in the following data-packet format:

```
[wave:4,100,false:13947,14185,14094,13787,13389,12969,12560,12151,11741,11343,10968,106
16,10263,9900,9536,9161,8775,8422,8070,7740,7468,7252,7070,6888,6717,6581,6490,6422,636
5,6319,6274,6228,6217,6217,6206,6194,6171,6149,6126,6092,6012,5956,5933,5876,5830,5796,
5728,5671,5614,5535,5501,5944,7524,9638,11139,12014,12742,13492,14071,14287,14174,1385
6,13458,13037,12605,12185,11787,11389,11003,10650,10298,9945,9616,9275,8911,8558,8206,
7865,7581,7354,7161,6979,6808,6649,6524,6456,6399,6331,6285,6251,6229,6217,6194,6183,61
60,6115,6069,6035,5967,5910,]

[wave:8,100,false:-185,-193,-199,-204,-210,-216,-221,-225,-229,-231,-233,-236,-237,-238,-238,-
233,-223,-205,-175,-133,-81,22,38,98,154,204,245,278,304,323,335,341,341,336,326,313,297,
279,259,   236,212,186,159,131,102,73,45,18,-7,-33,-57,-80,-101,-118,-134,-148,-160,-172,-180,-
187,194,-199,-204,-210,-216,-221,-225,-229,-231,-233,-235,-236,-237,-234,-225,-206,-177,-135,-
84,-26,33,92, 148, 197,240,276,303,322,333,338,338,333,324,311,296,279,259,236, 212,185,]

[wave:5,100,false:1483,1508,1520,1513,1499,1494,1511,1550,1594,1637,1674,1704,1737,1773,
1801,1817,1829,1836,1845,1857,1868,1866,1849,1824,1808,1820,1868,1946,2040,2135,2204,22
43,2259,2261,2257,2252,2248,2245,2243,2231,2208,2174,2132,2123,2192,2277,2273,2171,1996
,1766,1582,1520,1545,1589,1614,1614,1614,1631,1654,1667,1663,1651,1649,1670,1700,1730,1
758,1783,1808,1847,1882,1912,1932,1928,1930,1944,1953,1964,1978,1974,1969,1971,1988,
2040,2109, 2183,2270,2346,2385,2406,2420,2408,2385,2372,2349, 2321,2303,2273,2218,2162,]

[phdb:basic]
```

Each data-packet has following basic format:

[

Datatype of the relayed data: ……waveform (wave) or physiological data (phdb)

Wave-number identifying the signal, …… (4/5/8 or 9)

Data length, …… (100 or 25)

Gap in data, …… (true of false)

Actual data to be read into the data buffers ……data to be read

]

Here the first component (in pink) in the data packet identifies if it is a waveform data-packet or physiological data-packet. Wave-number (in grey) sorts the data packets to one of the four waveform signals. These wave-numbers and the implied waveforms are explained below. The next component (in yellow) indicates the length of data in the packet followed by the gap identifier (in turquoise) which indicates if there is any gap in the data, followed by the actual data.

4→ Invasive blood pressure waveform1.

5→ Invasive blood pressure waveform2 (central venous pressure waveform).

8→ Plethysmography waveform.

9→ End Tidal CO_2 waveform.

TCP-read algorithm reads and sorts these data and puts them in appropriate data buffers. The data buffers can be either read or written to but do not support simultaneous read and write operations on the data buffers. Therefore the TCP-read algorithm needs to check for any current read operation on the data buffer and acquire a write control whenever the buffer is in idle state.

3.3.2. State-Engine in RT-SAAM

This section explains the fundamental concepts which control the execution of the state-engine and thus the program flow in the **RT-SAAM**.

State-Engine: A state-engine is an advanced computer programming tool available in LabVIEW^{TM} wherein a computer application is divided into logical blocks of program code called states. Only one state can be active (running) at any instant of time. The word ‘state’ is often referred to as “waiting” in context to state-engine, which implies that the system waits in the current state until the next event occurs and triggers a state change. Events are external occurrences to which the state-engine responds by taking appropriate action. For instance RT-SAAM is divided into following six states:

1. Start / Initialising State.
2. Pre-File-Read State.
3. File-Read State.
4. Analysing/Processing State
5. Display State and
6. Exit State.

The program control is passed from one state to another in response to an event. For example data received from the TCP-port-1467 would trigger an event which causes the program control to be passed from pre-file-read to file-read state. Another integral concept to a state-engine is an action. Actions are the responses generated by a state machine whenever an event occurs. Action can be in the form of a simple state change or activating an external device. The action generated in response to the event depends upon the current state and the event that has occurred. Thus a state engine passes the program control between various states in the state engine depending upon its interactions with the external occurrences (signalled by corresponding event). In the case of SAAM, external events that trigger state-changes are those events which result from TCP-read and the size of the buffer variables. The state-engine algorithm employed in RT-SAAM is explained in detail in section 3.4.3. For more theory on state-engine refer to [78].

3.3.3. Comment Logging

SAAM has a comment section on the user interface which provides various buttons for logging the anaesthetists' response. For example the anaesthetist can press the "Yes" button in the absolute hypovolaemia comment section when the patient is showing symptoms of possible hypovolaemia. These responses from the anaesthetists are saved into a digital file at the end of each data collection to be used later for offline analysis of SAAM's performance. The real-time version of SAAM is also responsible for handling this commenting section.

3.4. Real Time Algorithm Development

RT-SAAM was developed using MATLAB^{TM}, LabVIEW^{TM} and Java^{TM} scripts. Most of the algorithms implemented in RT-SAAM^{TM} were in math-script programming language (executable MATLAB^{TM} code) implemented using the mathscript block in LabVIEW^{TM}. The State-Engine, TCP-Read and Comment-logging scripts were implemented in LabVIEW^{TM} in the form of LabVIEW^{TM} graphical code and math-script.

RT-SAAM has to simultaneously perform following three operations;

1. Read data packets and save them to appropriate data buffers (TCP-read).
2. Process data.
3. Log/save the user comments (Comment-logging).

Considering the system requirements, the state-engine entity was a realistic approach towards developing the complex system in an efficient form. The following flowcharts summarised the state-engine algorithm along with the two auxiliary algorithms for performing TCP-read and Comment-logging. In RT-SAAM flowchart (Figure 3.3.3-1) the blocks highlighted in blue represent the TCP-read algorithm, the green blocks represent the Comment-Logging algorithm and the blocks in black represent the state-engine algorithm. The TCP-read algorithm has a sub-module called Temp-Data Logging which performs the actual operation of sorting and logging the data-packets into appropriate buffer variable, refer flowchart in Figure 3.3.3-1. State-engine algorithm is graphically illustrated in Figure 3.4.3-1.

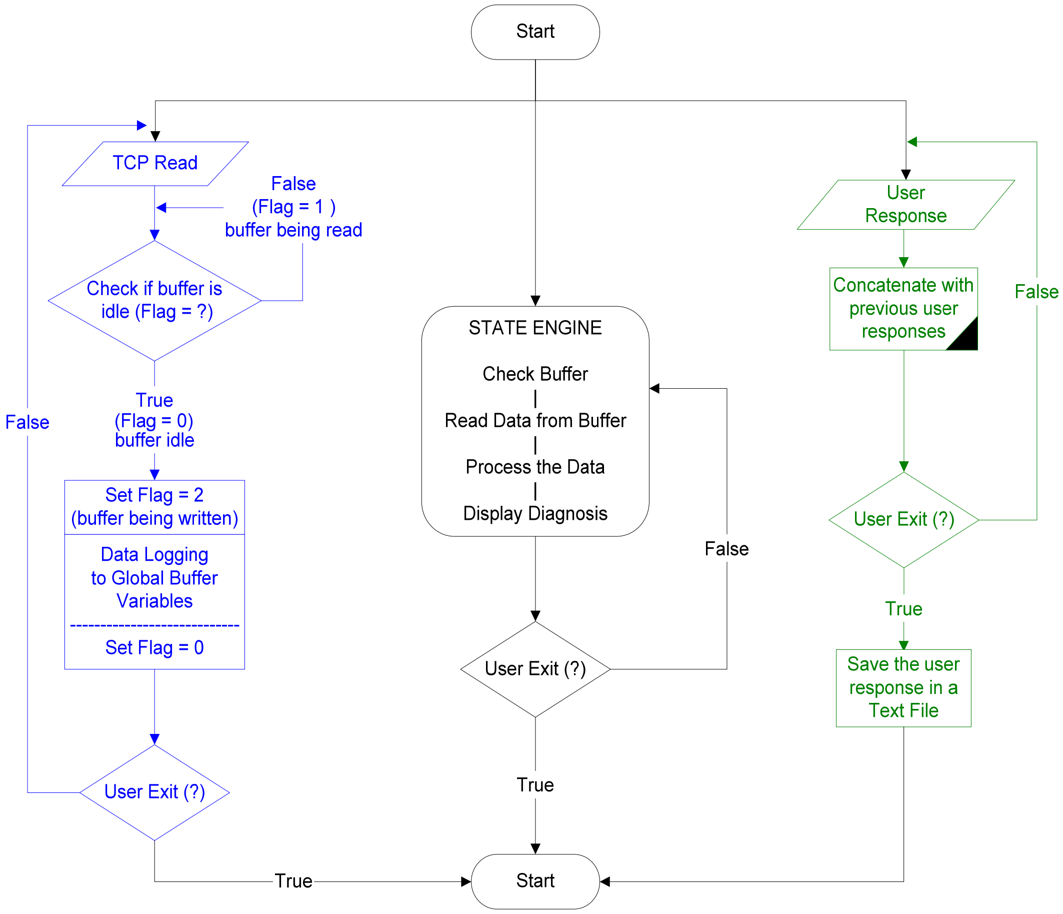


Figure 3.3.3-1. RT-SAAM Flowchart.

RT-SAAM is implemented in the form of three closed loops illustrated in Figure 3.3.3-1 above:

1. Data acquisition loop (in blue) reads data from the TCP-port and logs them into appropriate data-buffers.
2. Comment-logging loop (in green) is responsible for logging user comments/diagnoses into a text file.
3. State-engine loop (in black) which reads the data from the data-buffers and processes the data to generate the diagnostic alarms and audio-visual prompts.

Each of these three loops executes repetitively and simultaneously until the user exits SAAM by pressing the "STOP" button on the front panel.

3.4.1. Data Acquisition Loop (Algorithm)

Data acquisition loop executes the following tasks:

1.) TCP connection is opened on TCP-port-1467. DOMonitor relays data on port-1467.

2.) A while loop is executed which repetitively reads a character from the established TCP connection.

3.) If a character is read successfully then it concatenates the most recently read character to the previously read characters from the current data-packet.

4.) Step 2 and step 3 are repeated in a nested while loop until a ']' is encountered in the data-packet, indicating the end of the data-packet is reached.

5.) The wave number, gap information and the packet length are extracted from the read data characters. This information is used for identifying and sorting the data-packets.

6.) The buffer status is checked by checking the buffer flag. If the buffer is idle a write lock is acquired on the buffer.

7.) Based on the wave number, appropriate data logging block is executed as illustrated in the last section of the temp-data logging flowchart (see Figure 3.4.1-1).

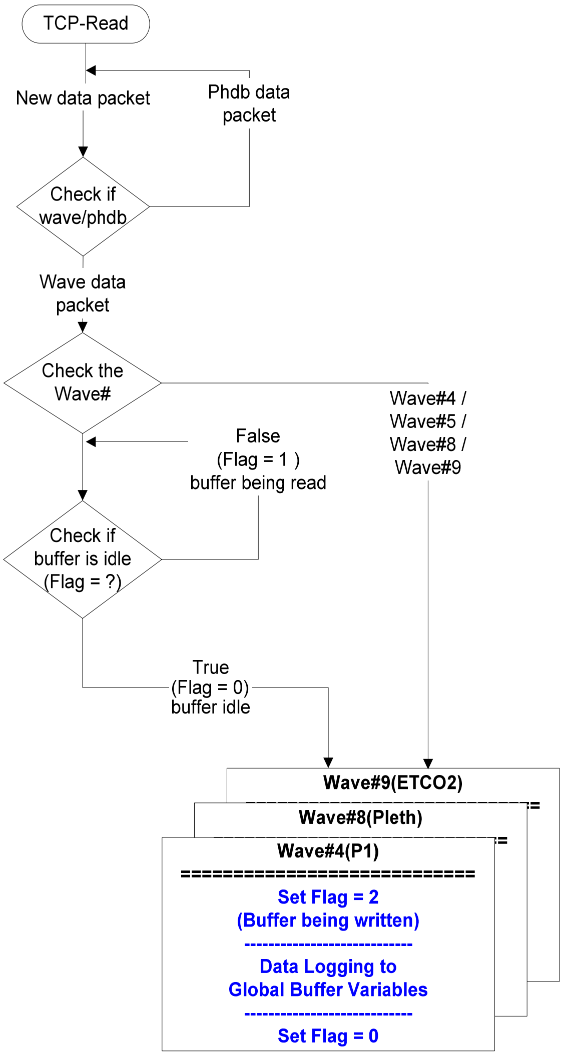


Figure 3.4.1-1. Temp-Data Logging Flowchart.

3.4.2. Comment-Logging Loop/Algorithm

The comment logging loop is implemented in the form of a LabVIEW^{TM} event structures nested within a while loop. Event structures are equivalent to action listeners in Java^{TM} and the primary function of an event structure is to capture any user generated events. For instance if the user presses 'YES' button in the FCO comment section of the SAAM's front panel (see Figure 3.4.2-1) then a button press event is triggered. The event structure captures this event and executes appropriate action code which is programmed in the event structure.

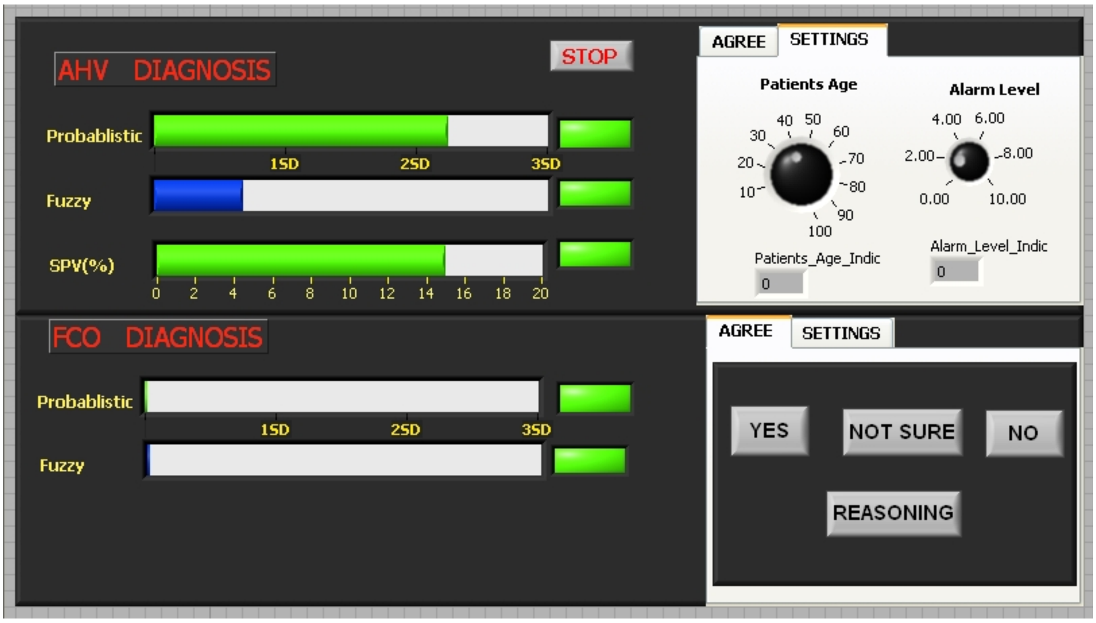


Figure 3.4.2-1. RT-SAAM Front Panel.

The event structures in the comment logging loop captures all the button press events generated in response to the user pressing the buttons. The time-stamp (date and time) of each button-press event is captured. All the captured events are logged into a text file. Comment logging loop executes repetitively until the 'STOP' button is pressed by the user. A sample comment logging output is illustrated below. These comment logs can be used for checking the anaesthetists' diagnoses against the diagnoses generated by SAAM.

```
27/12/2006 12:08:18 p.m.  AHV_Yes
27/12/2006 12:08:18 p.m.  AHV_Not_Sure
27/12/2006 12:08:23 p.m.  AHV_No
27/12/2006 12:08:20 p.m.  FCO_Yes
27/12/2006 12:08:23 p.m.  FCO_No
```

3.4.3. State-Engine Loop/Algorithm

The state-engine loop in RT-SAAM is programmed in LabVIEW^{TM} by nesting a case-structure inside a while loop. The state-engine flowchart in Figure 3.4.3-1 graphically illustrates how the program control is passed between the six states in RT-SAAM. The external events which trigger state changes and actions are user generated events like button press and the events generated by data acquisition blocks. The blocks which trigger state changes are highlighted in yellow. The data acquisition blocks continuously update the global buffer variable with new data-packets. The current status (represented by the flag value) changes from '0' which indicates the buffer is idle to ‘2’ when it is being written by the temp data-logging algorithm. Each of these states is explained in detail in the following text.

3.4.3.1. **Start-State**

When RT-SAAM is executed the system is in start-state, all the input and output files are loaded into the computers memory. The control automatically passes from start-state to init-state (simultaneously). The data-acquisition loop and comment-logging loop are also activated simultaneously.

3.4.3.2. **Init-State (Initialising state)**

The init-state is executed only once during a single data-collection session. Initialising state’s main task is to initialise all the global data buffer variables and flag to zero. There are six global variables and a global buffer flag which indicates the buffer status. The three global buffer variables store the data temporarily before it is transferred to corresponding global data-batch variables (three). The global buffer variables can be accessed only when they are idle (i.e. flag = 0). After initialising the global variables and flag to zero value the control is passed to the pre-file-read-state. To visualise the behaviour of global variables in context to SAAM refer to Figure 3.4.3-2.

3.4.3.3. **Pre-File-Read-State**

Pre-file-read-state checks if data read into the buffer is ≥ 10-second data (1000 data samples) for each of the four data-waveforms required by SAAM. Once the data in the buffer exceeds the 1000-sample threshold for all four-signals, the control is passed to the file-read state. The pre-file-read state is timed out after every 1 millisecond interval if it is waiting for more data to be read into the buffer. This prevents the race condition in the state-engine, waiting for other events.

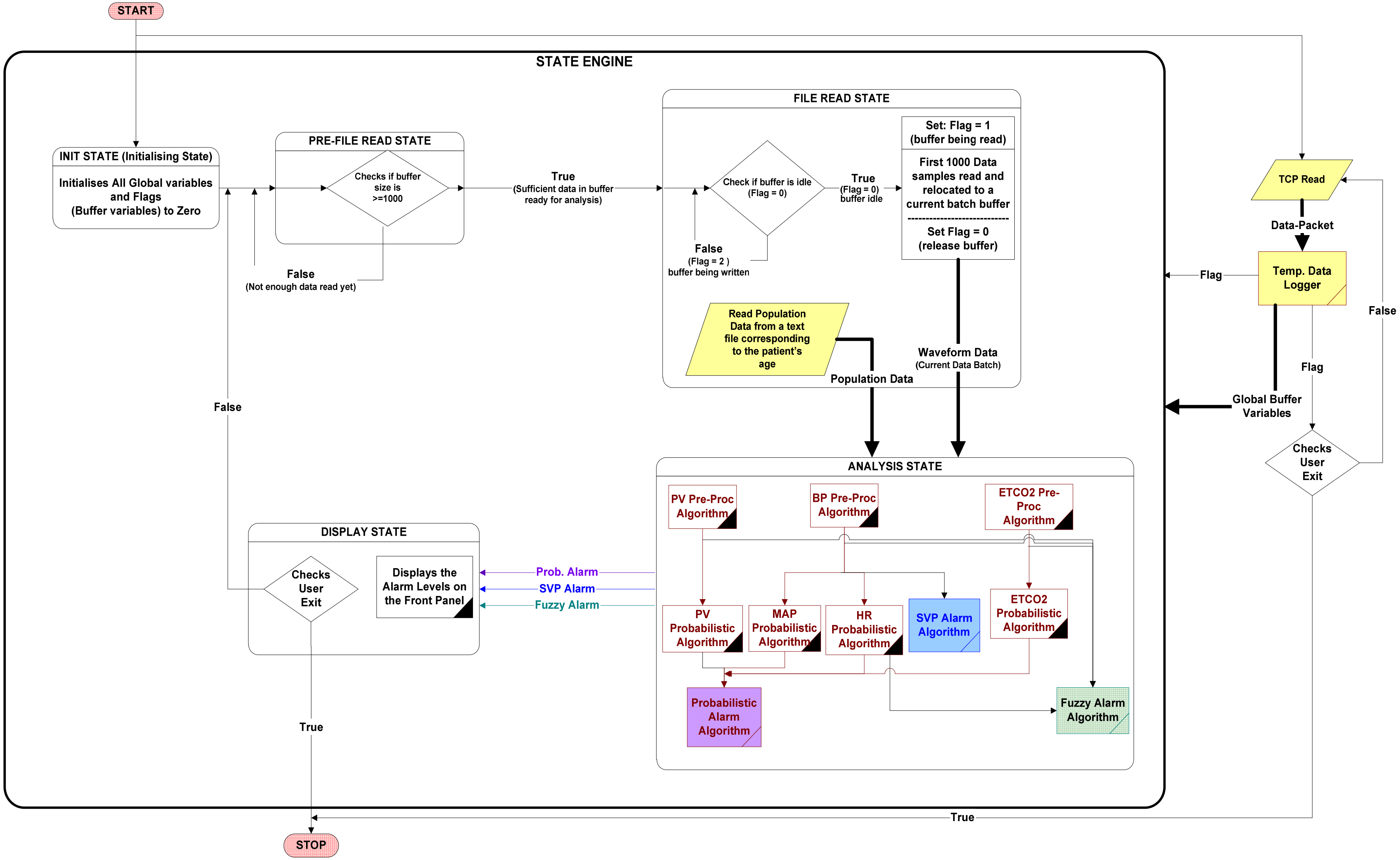

START
STATE ENGINE
INIT STATE (Initialising State)
Initialises All Global variables and Flags (Buffer variables) to Zero
PRE-FILE READ STATE
Checks if buffer size is >=1000
False (Not enough data read yet)
True (Sufficient data in buffer ready for analysis)
FILE READ STATE
Check if buffer is idle (Flag = 0)
True (Flag = 0) buffer idle
False (Flag = 2) buffer being written
Set: Flag = 1 (buffer being read)
First 1000 Data samples read and relocated to a current batch buffer
Set Flag = 0 (release buffer)
Read Population Data from a text file corresponding to the patient's age
Population Data
Waveform Data (Current Data Batch)
False
ANALYSIS STATE
PV Pre-Proc Algorithm
BP Pre-Proc Algorithm
ETCO2 Pre-Proc Algorithm
PV Probabilistic Algorithm
MAP Probabilistic Algorithm
HR Probabilistic Algorithm
SVP Alarm Algorithm
ETCO2 Probabilistic Algorithm
Probabilistic Alarm Algorithm
Fuzzy Alarm Algorithm
DISPLAY STATE
Checks User Exit
Displays the Alarm Levels on the Front Panel
Prob. Alarm
SVP Alarm
Fuzzy Alarm
True
TCP Read
Data-Packet
Temp. Data Logger
Flag
False
Flag
Global Buffer Variables
Checks User Exit
True
STOP

Figure 3.4.3-1. State Engine Flowchart.

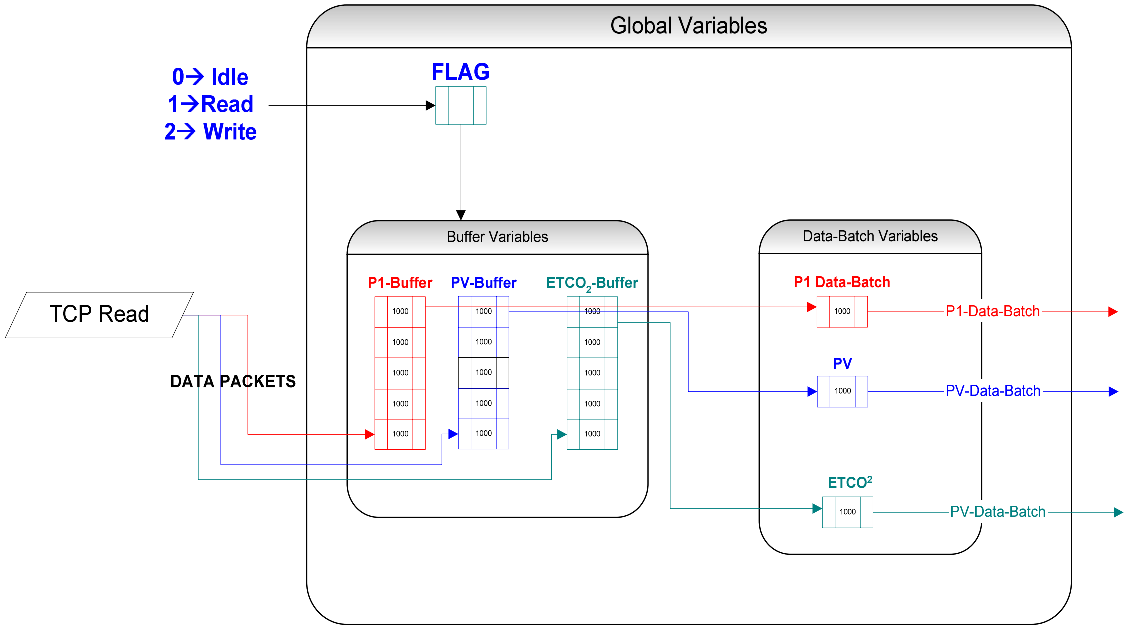


Figure 3.4.3-2. Global variables in RT-SAAM.

3.4.3.4. File-Read-State

In the file-read-state the population reads data from the text-file corresponding to the patient age selected in the settings panel on the user interface. For instance if the patient's age is 43 years, file-read-state reads the file containing the population statistics and fetches the data corresponding to the age group of 40 to 50 years. File-read-state then checks for the global buffer status. If the buffer is idle the file-read-state acquires a read lock on the global variable (by setting the global flag to '1'). In the event that the buffer is not idle the file-read state waits for 1-millisecond for the buffer to become idle, otherwise the file-read state is timed-out. This prevents the occurrence of race condition in the state-engine. The first 1000 data samples (10-second data) are transferred from the global buffer variables to global data-batch variables (see Figure 3.4.3-2). Thus the data in the global buffer variables are stored and disposed based on first-in-first-out (FIFO) principle. The global variable lock is released at the end of the file-read-state by setting global flag to '0'.

3.4.3.5. Analysis-State

Analysis-state performs the actual data processing of the waveform signal to generate diagnostic alarm levels. Firstly the analysis-state checks if the length of the data-set from the previous iteration is ≥ 2 minutes (13000-samples) for each of the waveform signals. Some diagnostic algorithms discussed in the previous chapters require a minimum of 2-minute long data for generating diagnosis.

If the data-set from the previous iteration is ≥ 13000-samples long, then the first 1000-samples (10-second long data) are discarded and the new data-batch (1000-samples) is added to the data from the previous iteration. Thus in every run of the state engine, the data in the state-engine is updated with a new set of data as the data-collection progresses. If the data from the previous iteration is < 13000-samples then the new 1000-samples are simply appended to the data from the previous iteration until the state-engine accumulates sufficient data (≥ 13000-samples).

The updated data-set is then supplied to the probabilistic alarm algorithm, fuzzy algorithm and the SPV algorithm. These processing algorithms then generate the corresponding alarm levels. Finally the program control and alarm levels are transferred to the display-state.

3.4.3.6. **Display-State**

The display-state generates appropriate audio-visual alarms after taking into account the sensitivity of the alarm selected by the user. The sensitivity of the probabilistic alarm can be varied by adjusting the alarm level control provided in RT-SAAMs' settings panel. Display-state checks if the 'STOP' button is pressed by the user. If the 'STOP' button is pressed, the control is passed to the exit-state otherwise the control is passed back to pre-file-read-state and so on.

3.4.3.7. **Exit-State**

The exit-state simply exits the RT-SAAM application and simultaneously the TCP-connection is closed and all open input/output files are closed. The program control can be transferred from any other state in the state-engine to the exit-state at any instant by pressing the 'STOP' button.

3.5. Chapter Summary

The RT-SAAM program communicates with the S/5 monitor via the DOMonitor. DOMonitor maintains the data transmission to IDAS so that the original communication setup in the hospital is not affected. Using this communication setup RT-SAAM can be used for data-collection and testing. Following chapters will discuss real-time data collection and the performance of the system in offline and online testing.

Chapter 4 Diagnostics and Testing

4.1. Introduction

SAAM provides a test platform for testing the prototype diagnostic alarm system proposed in Chapter 1. With ethics approval from local ethics committees and informed patient consent, physiological data was collected in real-time. SAAM was tested out extensively through offline (retrospective) testing and a few real-time clinical test sessions. The clinical tests were conducted in operating theatres on Level-8, Auckland City Hospital. Dr. Michael Harrison, Anaesthetist, Auckland City Hospital provided his expertise and valuable time during the data-collection and data-analysis phases. The diagnoses generated by SAAM in offline and real-time tests were evaluated against the anaesthetists' diagnoses. SAAM was then further refined to minimise the number of false negatives and false positives. This Chapter concludes with a result summary and a brief discussion on the fine-tuning of the offline and real-time versions of SAAM.

4.2. Ethics Approval and Patient Consent

Since physiological data collection from the patient monitor would require ethical approval from the local ethics committees, ethics approvals were obtained from the Northern-X Regional Ethics Committee and from the Auckland University of Technology Ethics Committee (AUTEC). Copies of these ethics approval letters from Northern-X and AUTEC can be found in Appendix B1 and B2 respectively. As part of the ethics approval, an informed written consent was required from each patient who participated in this research. Before signing the consent form, each patient was informed about the project by the anaesthetist and the participant was given a patient information sheet to understand the purpose of the research. Appendix B3 and Appendix B4 respectively show a copy of the patient information sheet and the consent form respectively.

4.3. Data Collection

The physiological data were directly collected from the S/5 Datex-Ohmeda (GE, Datex-Ohmeda, Helsinki, Finland) anaesthesia monitor in the operating theatre during a variety of operative procedures. A large amount of patient data used for offline (retrospective) analyses were collected over many years by the projects' external supervisor, Dr. Michael Harrison. These data were collected from patients in UK and New Zealand with the respective local ethical approvals obtained by Dr. Michael Harrison for the use of these data in developing an anaesthesia alarm systems like SAAM. Some of these data were used for generating the population statistics which is used by the probabilistic module. These data were collected using a software program called S/5 Collect from GE Healthcare Limited. S/5 Collect, however, supports only data-logging to a digital file and does not relay the data to any other device or application. Therefore S/5 Collect can not be used for real-time testing.

Real-time data-collection involves tracking and recruiting patients for data-collection and setting up the desktop computer running SAAM and DOMonitor in an operating theatre prior to every data-collection session. The patients recruited for real-time data collection were patients who were likely to suffer moderate to major blood-loss due to the nature of the surgery. Another important criterion to be considered during patient recruitment was that, only those patients who needed an arterial pressure line (for measuring BP/P1 signal invasively) during the surgery could be recruited for data-collection.

4.3.1. Setup

Figure 4.3.1-1 to Figure 4.3.1-4 show the practical setup for the real-time test bed during a real-time data collection and testing session at the Auckland City Hospital. The goal of the real-time data collection was to capture the anaesthetists' diagnosis and the patient data with the correct time-stamp and to get suggestions from the anaesthetist for making the prototype alarm more ergonomic. The alarm-levels generated by SAAM would be compared with the anaesthetists' diagnosis to determine the level of agreement between SAAM and the anaesthetists. This comparison was carried out in offline mode.

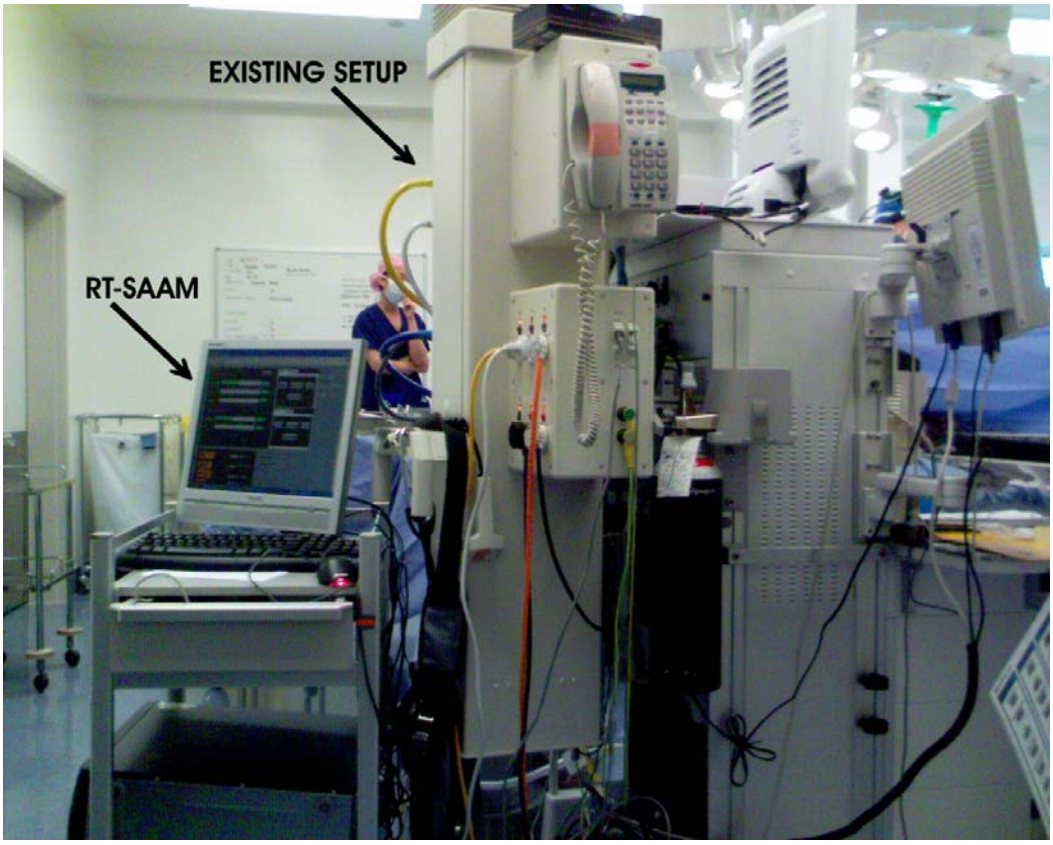


Figure 4.3.1-1. RT-SAAM Setup in the Operating Theatre.

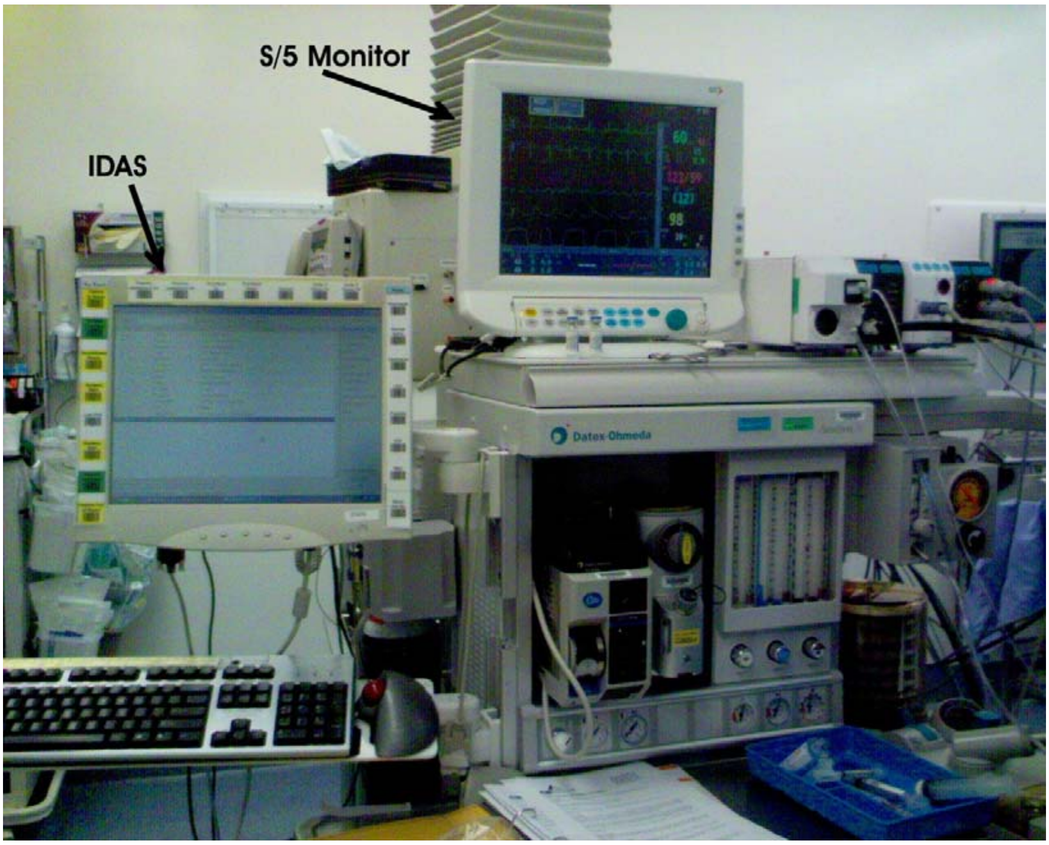


Figure 4.3.1-2. Existing Setup in the Operating Theatre.

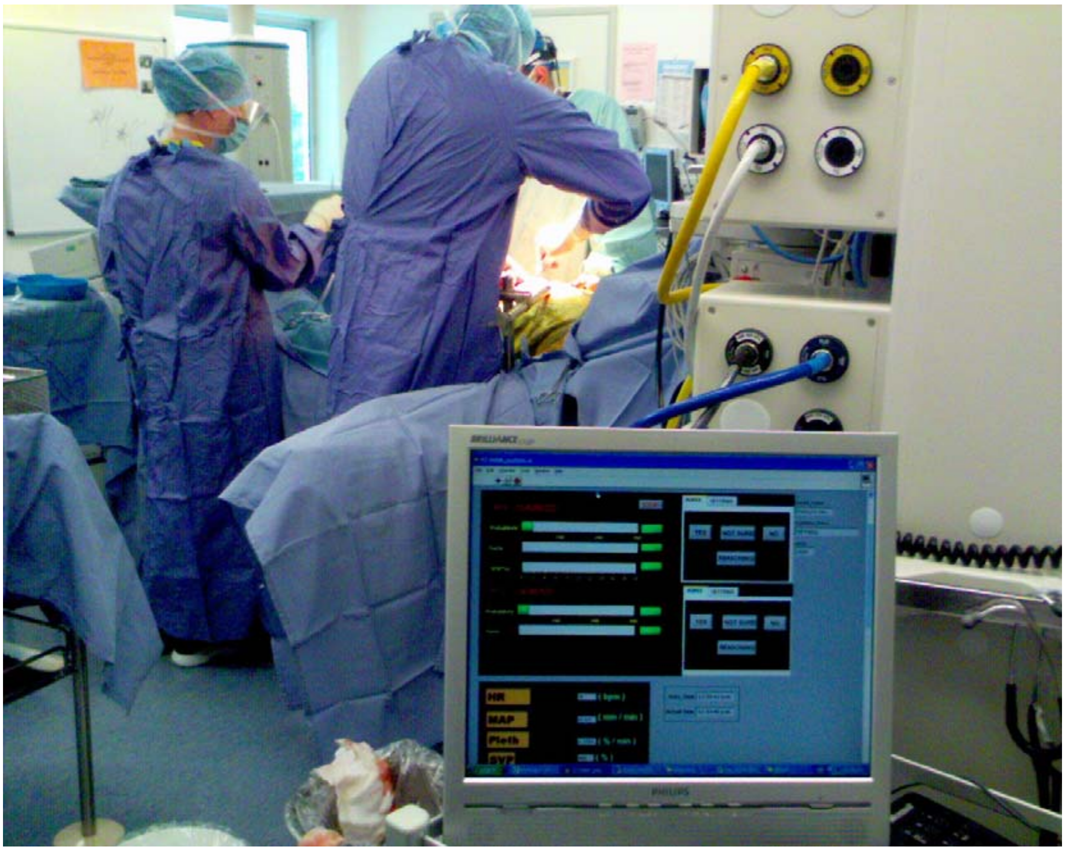


Figure 4.3.1-3. Screen-Shot of the RT-SAAM During a Real-Time Data-Collection Session.

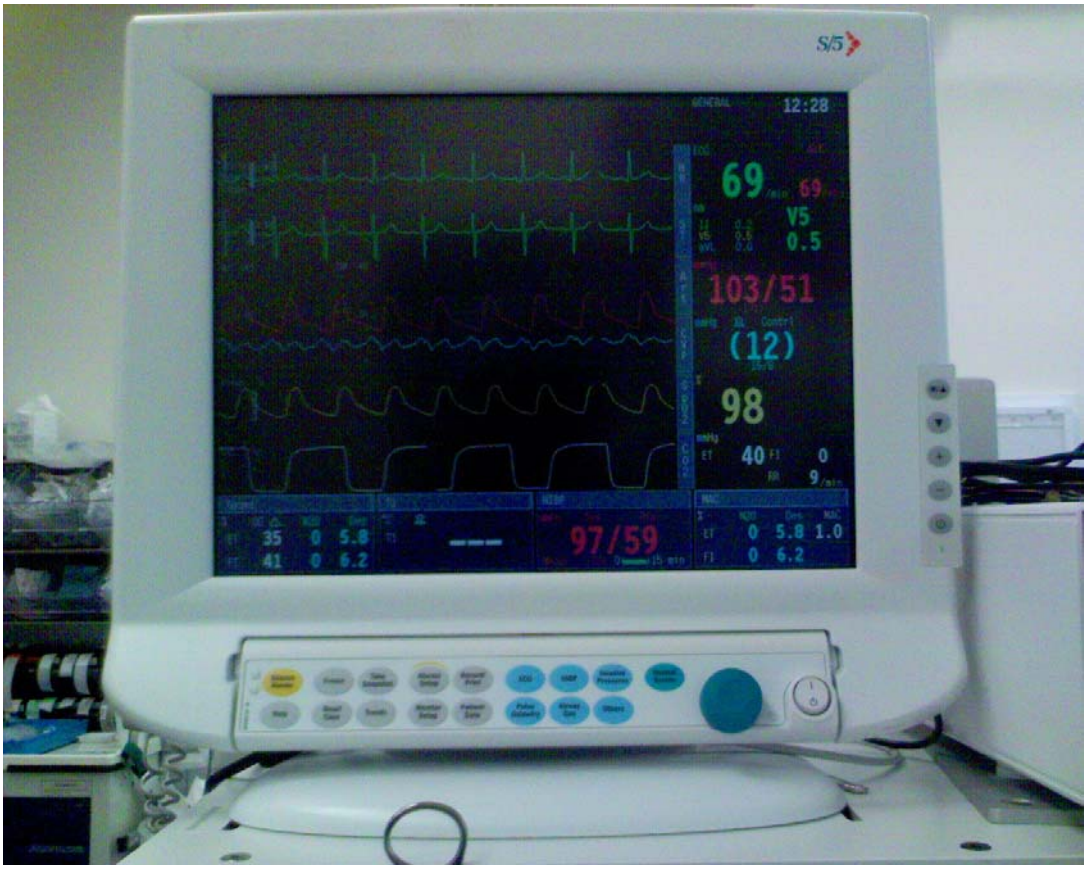


Figure 4.3.1-4. Screen-Shot of the Datex-Ohmeda S/5 Monitor During a Real-Time Data-Collection Session.

4.3.2. Challenges

Prior to the development of the real-time system it was proposed to use a PCI (Peripheral Component Interconnect) based data-acquisition card for data collection from the older **S/3** Datex-Ohmeda anaesthesia monitor. The S/3 monitor relays the required analogue data waveforms to a 44-pin port in addition to the serial port which exists on the newer S/5 monitor. A PCI-based data-collection approach was proposed to be used for data-acquisition via a 44-pin port at the back of the S/3 anaesthesia monitor. However the anaesthesia monitors in the operation theatre were recently upgraded to the latest S/5 series Datex-Ohmeda models. The S/5 monitors deprecated (rendered obsolete) the 44-pin port which existed in the older S/3 monitors. In the new S/5 monitors all the data are relayed out of the monitor via single serial port, thus new means for real-time data collection had to be developed as discussed in Chapter 4. Establishing the new communication protocol for real-time data collection was time-consuming as it required several visits to the operation theatre and liaising with Dr. Andrew Lowe for his expertise in data-collection. This communication issue was not anticipated earlier and it took more than two months to resolve this issue, thus system validation had to be postponed. After resolving the communication issue RT-SAAM required a couple of trial runs for debugging the new setup before successfully establishing data communication with DOMonitor and S/5 in real-time.

4.4. System Validation

Complete real-time testing of RT-SAAM is beyond the scope of this project as it involves recruiting suitable patients which is a time-consuming task. Also the exact number of real time test sessions required for gauging system performance and refining the system can not be predicted beforehand. It is a rough estimation that at least 30-test sessions will be required for successful testing and refining of RT-SAAM. The real time testing phase will follow the study methodology illustrated in Figure 4.3.2-1. Initially 10 real-time test sessions will be conducted. Following this part of the study, the process will be repeated either once or twice for further refining the system. It is proposed that after every test batch of 10-patients the analyzing algorithm will be fine tuned to make RT-SAAM's diagnoses more accurate. The system performance however could be validated through a few trial real-time test sessions and extensive offline tests.

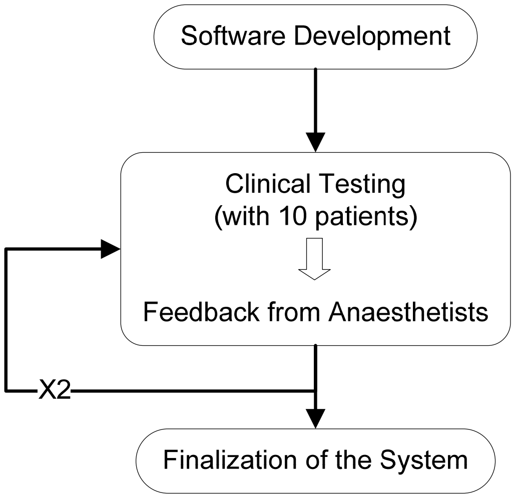


Figure 4.3.2-1. Flow Chart for Real-Time Test Methodology.

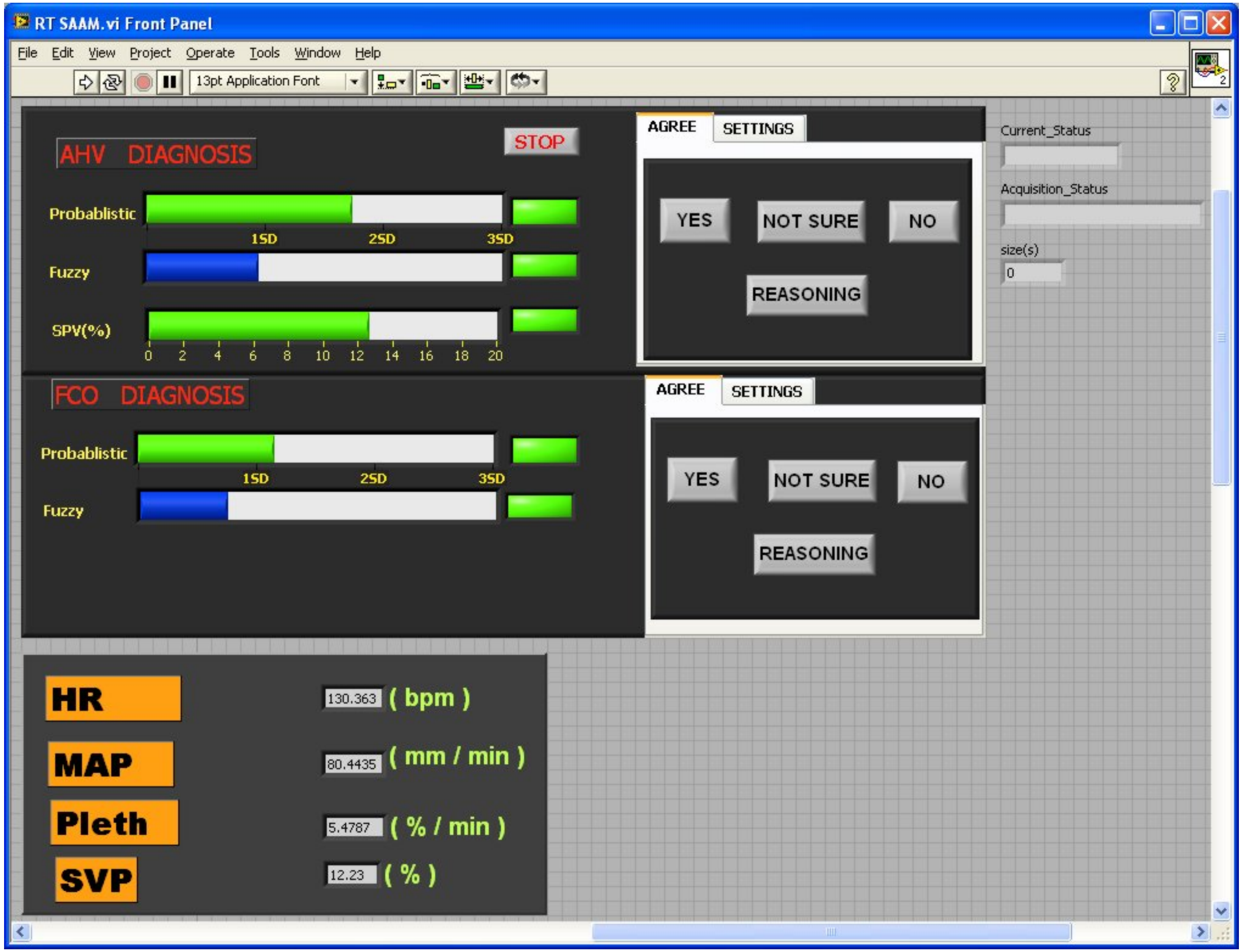
At the time of publishing this thesis RT-SAAM had been successfully tested with 6-patients (14 hours worth data) in real-time tests and 18-patients (51 hours worth data) in offline tests. Real-time performance of the system can not be validated through 6-patients and thus more real-time testing is required. Real-time data collection is a very slow process as it involves recruiting sufficient number of suitable patients. The performance of RT-SAAM therefore could only be validated through offline analysis which will be described in detail in this section. The offline test results still hold true for both (offline and real time) versions of SAAM because both employ the same diagnostic algorithms. The real-time testing was carried out on the test-bed shown in Figure 4.3.1-1. The graphical interface provided to the anaesthetists is shown in Figure 4.4.1-1.

4.4.1. RT-SAAM's User Interface

RT-SAAM's front panel is divided into following three sections:

4.4.1.1. **AHV Diagnosis Section**

AHV diagnosis section has three progress bars on the left hand side which indicate the AHV alarm levels from the three (Fuzzy, Probabilistic and SPV) diagnostic modules. The right hand side of the AHV diagnosis section has four buttons using which the anaesthetists interacts with RT-SAAM. The settings tab in AHV-diagnosis lets the user select the patient's age and sensitivity of the probabilistic alarm module.



size(s) 0

HR 130.363 (bpm)

MAP 80.4435 (mm / min)

Pleth 5.4787 (% / min)

SVP 12.23 (%)

Figure 4.4.1-1. RT-SAAM's Front Panel.

4.4.1.2. **FCO Diagnosis Section**

FCO diagnosis section has two progress bars on the left hand side which indicate the FCO alarm levels from the two (Fuzzy and Probabilistic) diagnostic modules. The right hand side of the FCO diagnosis section has four buttons using which the anaesthetists interacts with RT-SAAM.

4.4.1.3. **Raw Data Display Section**

The raw data display section located in the bottom portion of the RT-SAAM's front panel displays four variables namely, heart rate (HR), mean arterial pressure (MAP), plethysmography (Pleth) and respiration induced systolic pressure variations (SPV).

4.4.2. Diagnostic Method

Anaesthetists can record their response by clicking one of the buttons on the front panel. For instance if the anaesthetists detect absolute hypovolaemia by monitoring the patient information on the S/5 monitor (conventional method for diagnosis) then he/she can respond by pressing the 'YES' button on the AHV comment panel. After consultation with anaesthetists it was decided that the anaesthetists should be presented with an audio prompt at 15 minutes intervals, asking him/her to respond. For instance the AHV module, generates an audio prompt after every 15 minutes, asking the user, "IS HYPOVOLAEMIA POSSIBLE ?". The anaesthetist can respond by pressing one of the buttons on RT-SAAMs' comment panel. RT-SAAMs' performance can be reviewed by checking the anaesthetists' response to the audio prompt. The diagnostic alarm levels generated by various modules in RT-SAAM are evaluated for reliability by verifying them against the clinical evidence and the anaesthetists' response. This comparison is carried out for every 15-minutes epoch. The level of agreement between the computer generated (SAAMs') diagnosis and the experts' (anaesthetists') diagnosis gives a performance indicator for the prototype diagnostic system.

Prior to proceeding to the next section it is imperative to acknowledge the fact that is very likely that there might be some variability regarding the diagnosis among a group of anaesthetists [15].

Experts handle each clinical event in real-time by monitoring the patient's hemodynamic state and studying all the evidence (cues) at hand. Anaesthetists take a therapeutic decision by forming a mental map of the available clinical evidence.

Diagnosis of any critical events during anaesthesia depends on all the available information (evidence) and the ability of the expert to recognise patterns of disease behaviour.

Expert diagnosis may vary between anaesthetists and authenticity of the diagnosis depends on the skills and experience of the anaesthetists. Thus there is no right or wrong diagnosis as even the anaesthetists' diagnoses has an uncertainty attached to them. Thus for evaluating SAAMs' diagnostic performance the level of agreement between SAAM and the anaesthetists is used as measure of how accurately SAAM can mimic anaesthetists performance.

4.4.3. Offline Analysis

Sufficient historical patient data collected by co-researchers over the past years were available and could be used for testing this prototype system. These data were used for offline data-simulation and analysed retrospectively by revisiting all the available patient information for each patient record. The analysis of this data set in offline mode is identical to the real-time analysis as same diagnostic algorithms were used for offline analysis as well as online analysis. Physiological data from 18-patients (divided into 204 epochs of 15-minutes duration each) was used for offline analysis. These data-sets had incomplete expert diagnoses (anaesthetists' diagnosis), as some of the 15-minutes epoch expert diagnoses were not available. Dr. Harrison studied the clinical evidence for each of these 204-epochs and generated the corresponding diagnoses for the incomplete epochs. However, it should be noted that as the clinical evidence relevant to this offline data was in the form of short notes / logs. Hence it would be difficult for another anaesthetist to correctly interpret offline data with reference to the accompanying clinical evidence. Due to this reason the offline data was analysed only by one anaesthetist i.e. Dr. Harrison.

All the clinical events and procedures that took place during each of these historical data collection sessions had been saved into text files and these files provided the clinical evidence which could be used by an expert for retrospectively analysing each 15-minutes epoch. A sample log file and the corresponding physiological data-waveforms can be found in Appendix C1.

The expert diagnosis was generated for each 15-minutes epoch from the entire physiological data-set. Even though retrospective expert diagnosis can not be as accurate as the real-time diagnosis given by the anaesthetist but it may be considered very close to the real-time diagnosis. Expert opinion in anaesthesia varies between anaesthetists and there is no gold standard for the expert diagnosis. The accuracy of the expert diagnosis in case of anaesthesia related critical events depends totally on the expertise of the anaesthetist and his past experience. Dr. Harrison has been working as anaesthetist for last 30-years and his expertise and experience provided the diagnoses for the 15-minutes epochs with missing diagnoses.

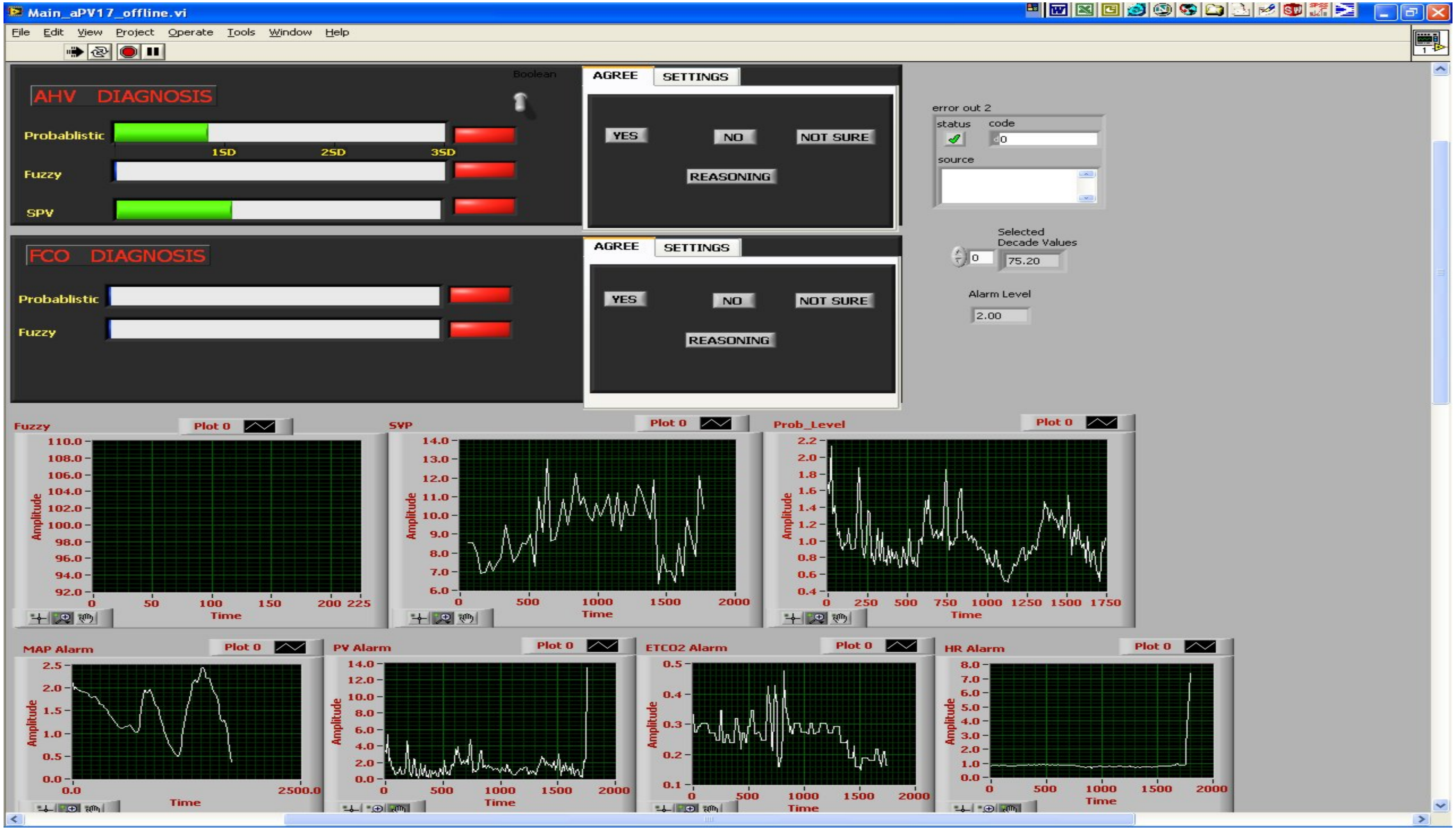
SAAM (offline version of the prototype alarm system) is capable of reading the binary patient data files and it analyses the data-waveforms in these files and generates the alarm levels for the entire length of the data-set. The data from the digital patient files was fed to SAAM and the corresponding computer generated diagnostic alarm levels were produced by the three diagnostic modules. The diagnostic alarm levels are in the form of a continuous waveform for the duration of the data-set and divided into 15-minutes epochs for comparison with the expert diagnoses.

Figure 4.4.3-1 shows the front panel of SAAM after the diagnosis is generated. In the figure the x-axis on the waveforms labelled Probabilistic, Fuzzy and SPV display the time elapsed since the first data sample was recorded in patient file which is currently being analysed. For instance ‘100’ on x-axis indicates 100-seconds since the first recorded sample in that file. Thus the alarm level for each 15-minutes epoch was manually noted down into a comparison chart (refer to Table 2 and Table 3 below) by checking the corresponding time interval on these waveforms on the front panel.

For each 15-minutes epoch the following rules were used for classifying the computer generated alarm levels on the front panel as positive and negative diagnosis:

1. SPV alarm level with a value ≤ 16% and ≥ 15% was considered possible AHV (**P**).
2. SPV alarm with a value > 16% was considered very likely AHV (**V**).
3. SPV alarm with a value < 15% was considered AHV not likely (**N**).
4. For probabilistic alarm level with a value ≤ 1.8 S.D. and ≥ 2 S.D. was considered possible AHV (**P**).

5. Probabilistic alarm with a value > 2 S.D. was considered very likely AHV (**V**).
6. Probabilistic alarm with a value < 1.8 S.D. was considered AHV not likely (**N**).
7. Any 15-minute epoch which had a ‘possible AHV (**P**)’ / ’very likely AHV (**V**)’ diagnosis was classified as positive diagnosis.
8. Any 15-minute epoch which had an ‘AHV not likely (**N**)’ was classified as negative diagnosis.



error out 2

status ✓ code 0

source

Selected Decade Values 0 75.20

Alarm Level 2.00

Figure 4.4.3-1. SAAM’s front panel.

Based on these classification rules each of the 204-epochs was studied and a comparison chart was constructed. Table 2 shows a sample comparison chart for patient #10. Similar comparison charts were constructed for all the 18 patients. Based on the agreement between the expert's diagnosis and SAAM's diagnosis, each of the 204-epochs was classified into four categories (TruePOS, TrueNEG, FalsePOS and FalseNEG).

Table 3 shows a sample classification for patient #10, this classification will be explained in detail in the next section. The results from this comparison were evaluated using an statistical test called Kappa, refer Kundel [79]. Kappa gives a statistical measure for evaluating inter-observer variability, that is, how often two or more observers agree/disagree in their interpretation.

Various literatures were surveyed to determine the validation approaches used for evaluating similar systems [5, 30, 34, 80-82]. However due to the simplicity of the kappa analysis and transparency of this assessment method it was selected as a tool for determining the inter observer variability. In this research project the expert (anaesthetist) and SAAM are the two observers interpreting the diagnoses for the pathological events from the physiological data. The value of kappa computed in the following section indicates the level of agreement/disagreement between the expert and SAAM.

4.4.4. Kappa Analysis for Offline Tests

For validating the performance of the alarm system prototype it was required to determine the reliability of SAAM's diagnoses for absolute hypovolaemia diagnosis. According to Kundel and Polansky [79], measurement of the level of agreement between the expert diagnoses and the computer generated diagnoses can provide;

1. A measurement for reliability of the diagnostic system.
2. A measure of specificity and sensitivity of the diagnostic system.
3. A measure for consistency of the diagnostic system.

Table 2. Offline Test - Diagnosis Outcome (for Patient #10).

Patient (10)	Filename	80104				80104-2					80104-3			80104-4			80104-5			80104-6		
	Time	**8:45a.m.**	9am	9:15	9:30	**9:47**	10am	10:15	10:30	10:45	**11:05**	11:15	11:30	**11:45**	12p.m.	12:15	**12:23**	12:45	1pm	**1:08**	1:30	1:45
	Slot	0-15	16-30	31-45	46-60	61-75	76-90	91:105	106-120	121-135	136-150	151-165	166-180	181-195	196-210	211-225	226-240	241-255	256-270	271:285	286-29(	291-305
SAAM Diag	Prob.	--	V	N		P	N	N	--		V	N	N	V	V	N	N	V	N	N	V	N
	SVP.	--	V	N		N	V	N	--		P	N	N	V	P	N	V	P	N	V	V	N
		--							--													
Expert Diag	Dr. Harrison	--	P	N		P	V	N	--		P	P	V	V	V	N	V	V	N	P	P	N

🟪	=	Positive Diagnoses
🟦	=	Negative Diagoses
🟥	=	Unsure
--	=	Major Artefact
□	=	No data Avail.
V	=	AHV Very Likely
P	=	AHV Possible
N	=	No AHV

Table 3. Offline Test – Agreement Chart (for Patient #10).

Patient No.	Time Slot	SAAM Diag.	Expert Diag.	Expert Comments		TruePOS	TrueNEG	FalsePOS	FalseNEG
						Agreement SAAM +ve Expert +ve	Agreement SAAM -ve Expert -ve	Disagreement SAAM +ve Expert -ve	Disagreement SAAM-ve Expert +ve
Patient 10	9:00	V	P	BP unstable		1			
	9:15	N	N				1		
	9:47	P	P	BP unstable		1			
	10:00	V	V	0953-1025	CVP fell 17 - 8	1			
	10:15	N	N				1		
	11:05	V	P	BP down, HR up		1			
	11:15	N	P	RASVP					1
	11:30	N	V	1121-1136	metaraminol, CVP 5				1
	11:45	V	V	1144-1213	CVP 6, MAP v variable	1			
	12:00	V	V	1144-1214		1			
	12:15	N	N	BP unstable but going up			1		
	12:23	V	V	1230-1244	Colloid, CVP 7,	1			
	12:45	V	P	BP down, HR up		1			
	13:00	N	N				1		
	13:08	V	P	1314-1337	MAP down to 67mmHg	1			
	13:30	V	P			1			
	13:45	N	N				1		
Total						10	5	0	2

For computing the level of agreement between the expert's diagnosis and SAAM's diagnosis, each of the diagnoses was explicitly expressed as positive or negative. Based on the positive or negative diagnosis generated by SAAM and the diagnosis by expert there were four possible permutations for AHV diagnosis;

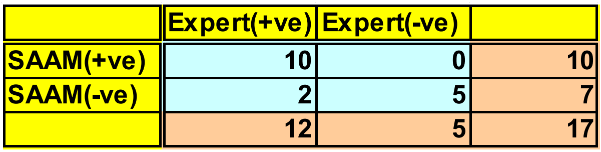
a) Both SAAM and Expert agree that AHV exists (TruePOS).
b) Both SAAM and Expert agree that AHV does not exist (TrueNEG).
c) SAAM gives positive diagnosis while Expert gives negative diagnosis (FalsePOS).
d) SAAM gives negative diagnosis while Expert gives positive diagnosis (FalseNEG).

A total of 204-epochs from the offline data-sets were identified and classified into the four classes described above. From Table 3, Table 4 was developed for computing the various statistical entities for kappa analysis and represents the epoch's corresponding to patient #10.

Similar classification was carried out for the entire data-set, refer Appendix C2 for offline testing – agreement chart for the entire data-set. The resulting statistical information for the entire data-set is summarised in the

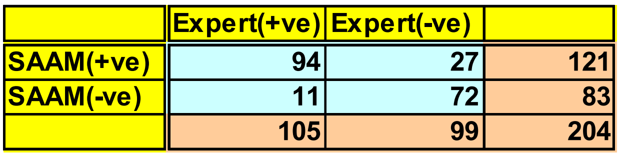
Table 5 below.

Table 4. Kappa Table (for Patient #10).



	Expert(+ve)	Expert(-ve)	
SAAM(+ve)	10	0	10
SAAM(-ve)	2	5	7
	12	5	17

Table 5. Kappa Table for Offline-Analysis of the Complete Data-Set.



	Expert(+ve)	Expert(-ve)	
SAAM(+ve)	94	27	121
SAAM(-ve)	11	72	83
	105	99	204

Based on the data from

Table 5 the positive agreement (P_{pos}) and negative agreement (P_{neg}) indices were calculated as follows:

$$P_{pos} = \frac{94+94}{(94+27)+(94+11)} = 0.83 \tag{4.1}$$

$$P_{neg} = \frac{72+72}{(27+72)+(11+72)} = 0.79 \tag{4.2}$$

The third index of agreement gives the overall agreement (P_o) level between the expert and SAAM; it was calculated with the following equation:

$$P_o = \frac{94+72}{204} = 0.81 \tag{4.3}$$

Agreements between the two diagnoses may be affected by chance. For instance if the expert had any prior knowledge that most of the epoch's had a positive diagnosis, then he would have had adopted a strategy of reporting a positive diagnosis whenever he had a doubt. Kappa (k) is a measurement of agreement between the expert and SAAM which has been corrected for error by chance. Kappa (k) is calculated by subtracting the proportion of readings that are expected to agree by chance (P_e) from the overall agreement (P_o) and dividing the remainder by the number of cases on which agreement is not expected to occur by chance.

$$k = \frac{P_o - P_e}{1 - P_e} \tag{4.4}$$

The agreement expected by chance (P_e) was calculated as follows:

$$P_e = \left(\frac{121}{204} \cdot \frac{105}{204}\right) + \left(\frac{83}{204} \cdot \frac{99}{204}\right) = 0.50 \tag{4.5}$$

$$\therefore \quad k = \frac{0.81 - 0.5}{1 - 0.5} = 0.62 \tag{4.6}$$

The standard error (SE) of k for a 2x2 table could be estimated with Eq.4.7:

$$SE = \sqrt{\frac{P_o(1-P_o)}{n(1-P_e)^2}} \tag{4.7}$$

$\therefore \quad SE = \sqrt{\frac{0.81(1-0.81)}{204(1-0.5)^2}} = 0.055$ CIs for k were $0.62 + 1.96 \cdot 0.055$ = **0.73** and $0.62 - 1.96 \cdot 0.055$ = **0.51.** The kappa analysis for offline tests is summarised in Table 6.

4.8

The 95% confidence intervals (CIs) for k could be calculated with the following equation (see Eq.4.9):

$$CI_{95\%} = k \pm 1.96 \cdot SE \qquad \textbf{4.9}$$

Thus the 95%

Table 6. Offline analysis results.

Overall Aggreement	Positive Agreement	Negative Agreement	Agreement by chance	Standard Error	95% Confidence Intervals for k
P_o	P_{pos}	P_{neg}	P_e	SE	$CI_{95\%}$
0.81	0.83	0.79	0.50	0.06	0.73 and 0.51

The value of k represents the strength of agreement between the two diagnoses; Table 7 below illustrates how a range of k-values are related to the corresponding strength of agreement beyond chance.

Table 7. k-Values Expressed as Strength of Agreement, [79].

k-value	Strength of Agreement beyond chance
<0	Poor
0-0.2	Slight
0.21-0.40	Fair
0.41-0.6	Moderate
0.61-0.8	Substantial
0.81-1	Almost perfect

Thus for the offline-tests (retrospective analysis) performed, kappa based statistical analysis showed substantial level of agreement (k = 0.62) between the expert's and SAAM's diagnoses.

4.4.5. Real Time Analysis

Real-time test result carries much more weight than the offline test result for two reasons. First, expert diagnosis is based on more clinical evidence than the retrospective expert diagnosis which was generated for the offline analysis. Second, the real-time testing simulates the actual operating environment for RT-SAAM.

At the time of publication of this thesis, RT-SAAM had been tested with 6-patients in real-time and the data from these tests were analysed retrospectively using the kappa analysis, as in the offline-tests. To obtain a reliable measure of agreement for real-time analysis of the system, a sizeable amount of patient data is required. Thus more data collection is required and therefore testing of the RT-SAAM prototype will be continued after the completion of this M.E. thesis. Further real-time tests will span over a period of 3 months and will be tested with up to 30 more patients.

RT-SAAM uses the communication protocol developed in Chapter 3 for data-acquisition. The acquired data is processed by RT-SAAM in real-time to generate the alarm levels as the time progresses. The alarm levels generated by RT-SAAM are saved into a text file (alarm log file) at every 10-seconds interval. The content of the alarm log file for patient RT02 is shown in Table 8 below. These alarm levels were later compared with the expert comments/opinions which were logged by pressing the comment buttons in the comment section of the front panel. As in offline analysis, the real-time test results were further evaluated using kappa analysis.

Table 8. Alarm Log File for Patient # RT02.

```
15/12/2006 10:07:27 a.m. AHV Prob. Alarm Level = 1.2326
15/12/2006 10:07:27 a.m. AHV SPV Alarm Level = 10.5632
15/12/2006 10:07:27 a.m. AHV Fuzzy Alarm Level = 2.1213
15/12/2006 10:12:20 a.m. AHV Prob. Alarm Level = 1.3412
15/12/2006 10:12:20 a.m. AHV SPV Alarm Level = 11.5637
15/12/2006 10:12:20 a.m. AHV Fuzzy Alarm Level = 2.1324
15/12/2006 10:17:31 a.m. AHV Prob. Alarm Level = 1.2567
15/12/2006 10:17:31 a.m. AHV SPV Alarm Level = 11.0234
```

4.4.6. Kappa Analysis for Real-Time Tests

The expert diagnosis and the RT-SAAM diagnosis were compared using the following excel spreadsheet (Table 9). During each data collection session a manual record of clinical events was also kept for backup. For example Table 10 shows a manual record for the first real time patient (patient # RT02). Such a record would prove useful in the unlikely event that the expert comments were not entered or saved due to some reason.

Kappa analysis was carried out for the complete data-set (refer to Appendix C3) from the real-time test and the results obtained are tabulated below (refer to Table 11). The real-time test kappa analysis showed that there was moderate level of agreement (k = 0.42) between the expert and RT-SAAM.

Table 9. Real-Time test – Diagnosis Outcome (for Patient # RT02).

Patient	File	151206-1	151206-2								151206-2									
RT-02	Time	9:50	10:05	10:20	10:35	10:50	11:05	11:20	11:35	11:50	12:05	12:20	12:35	12:50	1:05	1:20	1:35	1:50	2:05	2:20
	Slot	0-15	16-30	31-45	46-60	61-75	76-90	91:105	106-120	121-135	136-150	151-165	166-180	181-195	196-210	211-225	226-240	241-255	256-270	271:285
Alarm	Prob	--	N	N	N	N	N	N	V	N	N	N	V	V	--	--	--	--	--	
	SVP	--	N	N	V	N	N	N	N	N	V	V	V	V	V	V	N	V	V	
Mike																				
		N	N	N	N	N	N	N	P	P	P	P	P	V	V	N	N	V	N	

🟪	=	Positive Diagnoses
🟦	=	Negative Diagoses
🟥	=	Unsure
--	=	Major Artefact
□	=	No data Avail.
V	=	AHV Very Likely
P	=	AHV Possible
N	=	No AHV

Table 10. Manual Record for Clinical Events.

Patient # RT02	Date & Time: 9:50 am 151206	Status	
Time	Comments	AHV	FCO
9:50:00 a.m.	File start, Surgery starts		
9:55:00 a.m.	Local anes in epidural	N	N
10:00:00 a.m.	Stable(slight BP fluct bcoz of epidural)	N	N
10:15:00 a.m.	Data collection(DC) going normal	N	N
10:30:00 a.m.	DC going normal	N	N
10:45:00 a.m.	This pat. has an irreg HR rhytm…investigate	N	N
11:00:00 a.m.		N	N
11:15:00 a.m.		N	N
11:30:00 a.m.		N	N
11:45:00 a.m.	A little resp change	P	N
12:00:00 p.m.	IVC is clamped	P	N
12:15:00 p.m.	IVC released(12:24)	P	N
12:30:00 p.m.	IVC surgical compr….AHV likely	P	N
12:45:00 p.m.	Blood is being given…	V	P
1:00:00 p.m.	Minor fall in FCO, Definite AHV(1pm & 1:11pm	V	V
1:15:00 p.m.	Stable (break)	N	N
1:30:00 p.m.	Stable (break)	N	N
1:45:00 p.m.	Surgery resumed(bp, etco2 going down)	V	P
2:00:00 p.m.	Surgery ends.	N	N

Table 11. Real-Time Analysis Results.

Overall Aggreement	Positive Agreement	Negative Agreement	Agreement by chance	Standard Error	95% Confidence Intervals for k
P_o	P_{pos}	P_{neg}	P_e	SE	$CI_{95\%}$
0.75	0.56	0.82	0.50	0.13	0.68 and 0.15

4.5. Refining the Algorithms

Most of the algorithm refining and fine-tuning were carried out during the algorithm development cycle, for example the filter specifications were selected after evaluating and simulating the sample data sets from the historical offline data that was available during the initial stages of this research project. During the development phase, each of the algorithms was evaluated with sample test-data and the algorithms were fine-tuned for optimal performance, refer to Section 2.4 in Chapter 2. However it is projected that the real-time testing phase and the application of the prototype into real-life simulations may require further fine-tuning for improving the system performance. Some of the threshold values set for the pre-processing and the processing algorithms need to be further refined as more data is collected and evaluated during the system validation. For example the ETCO$_2$ threshold value which is currently set at 3 can be optimised to a more specific value as more data is collected. This will improve the diagnostic specificity and accuracy of the ETCO$_2$ algorithm. Also the filter specifications and the threshold values for individual algorithms can be made accurate as more data is collected and evaluated during the clinical testing and validation of RT-SAAM.

4.6. Chapter Summary

SAAM was successfully tested in both online and offline modes with ethical consent obtained from local ethics committees. From offline analysis it was evident that the pre-processing algorithms and diagnostic modules in the prototype system achieved satisfactory performance. Prior to being implemented in SAAM, each of the pre-processing and the diagnostic modules was tested individually with a large amount of patient data. The algorithms were refined over a period of 3 months before their implementation into SAAM. Both offline and real-time versions of SAAM used the same diagnostic algorithms and thus the system performance realized through offline tests was used as a performance indicator for the both the real-time and offline diagnostic versions.

The offline and real-time testing of the diagnostic alarm system prototype generated satisfactory values for kappa (k) indicating that there was substantial to moderate level of agreement between the expert and the computer generated diagnosis. Further real-time testing is required for making the real-time test results more credible. More real-time tests will be conducted in the near future and further fine-tuning of the system may be required for improving the system accuracy.

Chapter 5 Discussions and Conclusions

5.1. Introduction

This chapter presents discussion, conclusions and recommendations for future work. A concise and detailed discussion is presented on some of the major developmental stages of this research project. The conclusions section provides a summary of the research outcome and the conclusions reached. Finally recommendations are made for possible future works for further refining the performance of RT-SAAM and other possible applications for RT-SAAM.

5.2. Discussions

5.2.1. Data Simulation

Data simulation in MATLAB^{TM} was one of the milestones in this research project. The resultant algorithms developed during data simulation provided the basic foundation for SAAM and RT-SAAM. MATLAB^{TM} programming codes and techniques were learned and understood before starting with the data-simulation phase.

Data simulation for the sample physiological signals was carried out in MATLAB^{TM} environment. During data simulation it was learnt that the blood pressure signals (BP/P1) and the pulse volume signals (PV) were highly corrupted with noise signals and interference due to a variety of intra-operative events and procedures. Due to the random nature of these noisy physiological signals conventional filtering techniques could not achieve satisfactory noise cancellation as described in Chapter 2. Thus new filtering and pre-processing algorithms were developed for BP and PV signals. The noise filtering capability of these algorithms was further assessed through more rigorous data simulation. From the output of the BP and PV filtering algorithms it was observed that successful noise filtering could be achieved with the developed pre-processing algorithms (see Figure 5.2.1-1 to Figure 5.2.1-4). The variance based filtering algorithm eliminates segments of the data-waveforms which were highly corrupted by noise, it uses a DFT based low pass filter for attenuating the low-frequency noise component.

With the variance threshold values set at 50 to 1000 for the BP signal and 0.1 and 30 for the PV signal the noisy signal segments (with variance values exceeding these threshold values) were eliminated from the BP and PV waveforms. The limits for the variance based filter were generalised during the data-analysis stage by trial and error analysis of the sample physiological data. It was observed that the variance threshold values for PV signal had much wider inter-patient variability and therefore threshold values were not uniform for the different patient samples. After analysing data from 15 patients the limits were set at 0.1 and 30. It is anticipated that as more physiological data are collected and analysed the limits of the variance-based filtering will be fine tuned to reduce the inter-patient variability and thereby making the filtering process more accurate for the PV signal.

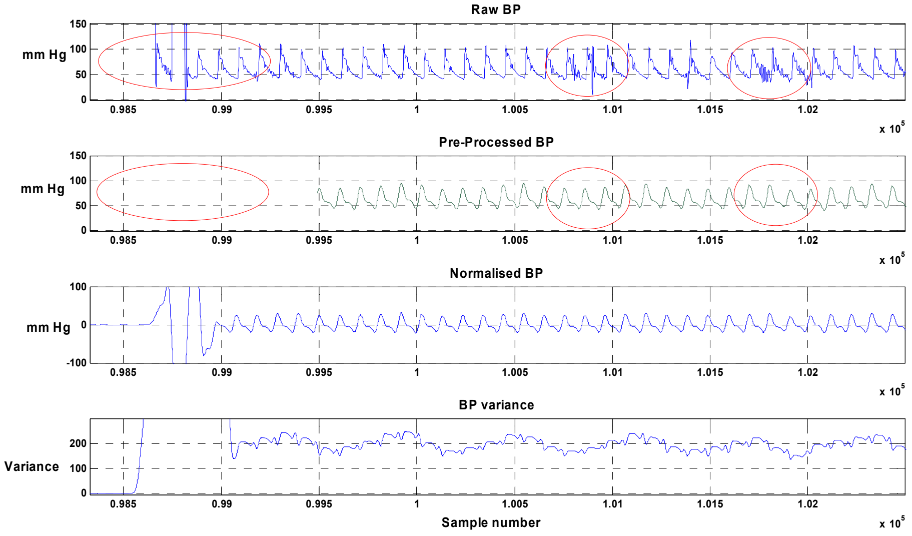


Figure 5.2.1-1. BP Pre-Processing Algorithm's Output.

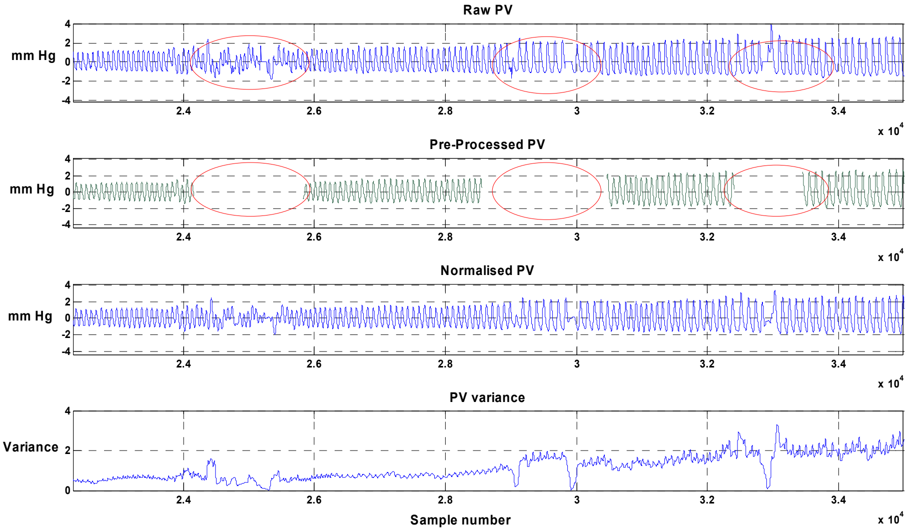


Figure 5.2.1-2. PV Pre-Processing Algorithm's Output.

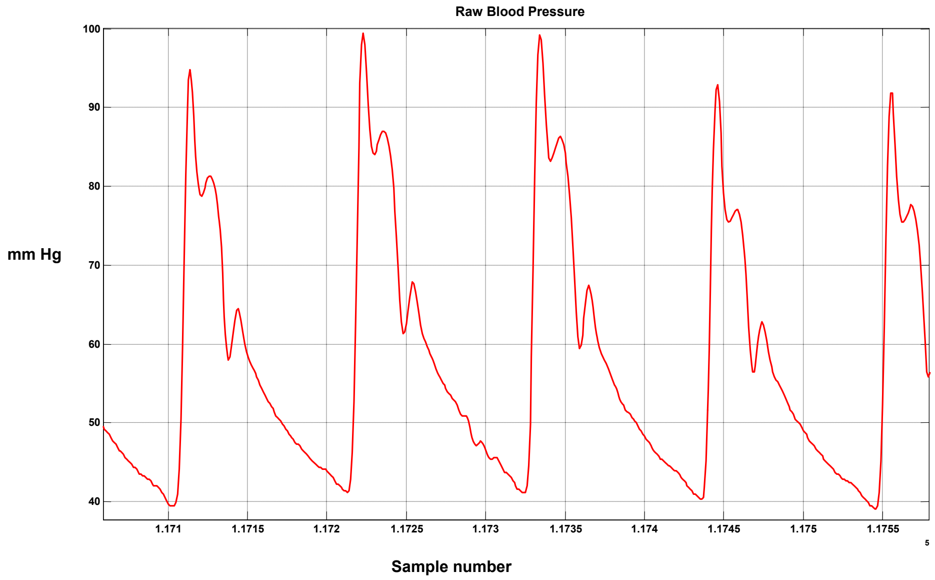


Figure 5.2.1-3. Raw Blood Pressure Waveform.

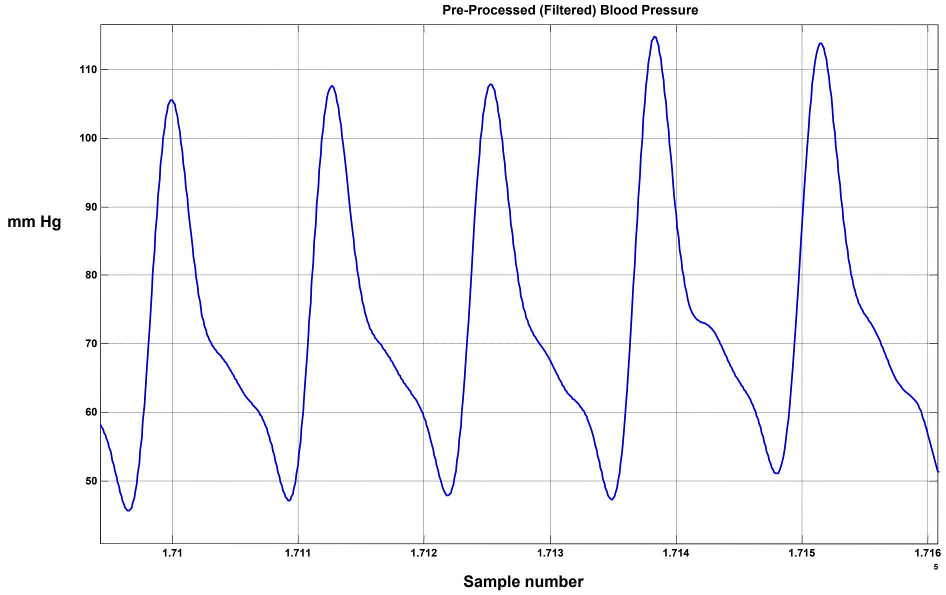


Figure 5.2.1-4. Filtered Blood Pressure Waveform.

Prior to implementing the variance based algorithm, conventional filtering algorithms (discussed in Chapter 2) were tested for their possible application in this research project but the results obtained were unsatisfactory. For instance the low pass BP signal shown in Figure 5.2.1-5 had a random noise component which had the same frequency component as the useful signal component and therefore the low pass filtering algorithm could not reject the random noise component. On the other hand the BP pre-processing algorithm developed in this research successfully eliminated such random noise components and simultaneously filtered the signal for high frequency noise (refer to Figure 5.2.1-1).

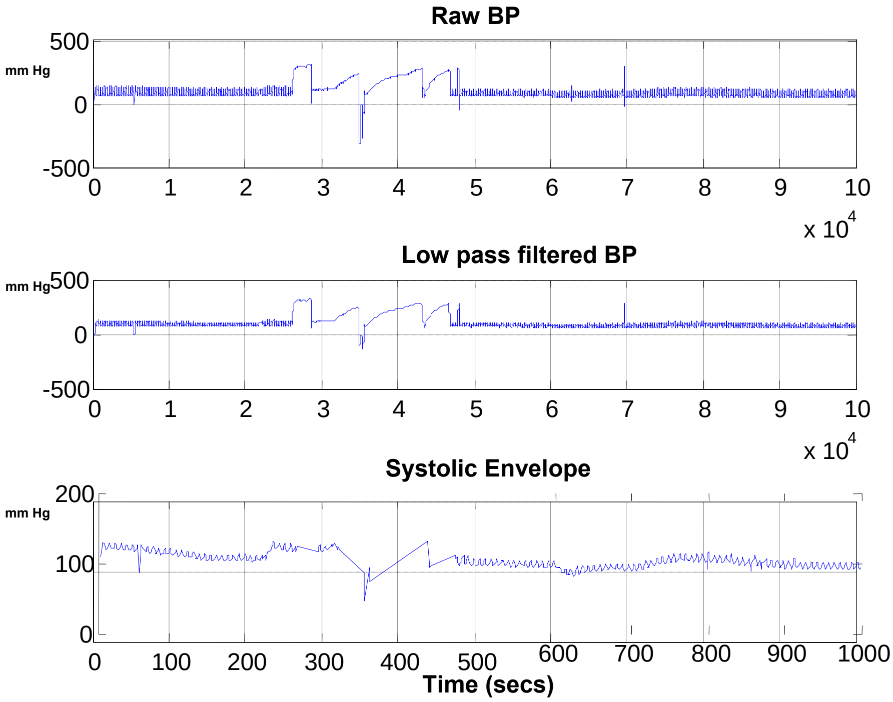


Figure 5.2.1-5. Lowpass filtered BP signal.

5.2.2. Algorithm Development

After data simulation in MATLAB^{TM} the diagnostic algorithms were developed by selecting the most appropriate signal processing techniques and several new algorithms were developed. A signal processing technique called windowing was then used for extracting the required information from the segments of the pre-processed physiological signals. For instance the HR, MAP and SAP were derived from the pre-processed BP signal using the BP-processing algorithm.

For detecting HR and SAP values from the raw BP waveform requires detecting the highest peaks (systolic pressure peaks) from the BP waveform. The number of systolic pressure peaks in the BP waveform per minute gives the HR value. Several conventional peak detection algorithms [73-75] like slope-based peak detection, DFT based peak detection and maxima-minima based peak detection were studied for this application. The presence of spurious peaks, for example dicrotic notches and their associated peaks, and the noise induced peaks in the raw BP waveform made peak detection a challenging task (refer to Figure 5.2.1-3). The conventional peak-detection algorithms were not effective when applied as separate algorithms and hence a new hybrid peak-detection algorithm was developed. The new peak-detection algorithm successfully extracted the genuine systolic peaks from the BP waveforms (see Figure 5.2.2-1). The SAP values were used for computing the respiration induced variations in the systolic pressure in the SPV module. With slight modification the same peak-detection algorithm was used for detecting the peaks and trough values from the filtered PV-signal. The PV peak and trough values were then further processed for computing the PV-value. The new peak detection algorithm proved to be very efficient at detecting peaks and troughs from the BP and PV waveforms.

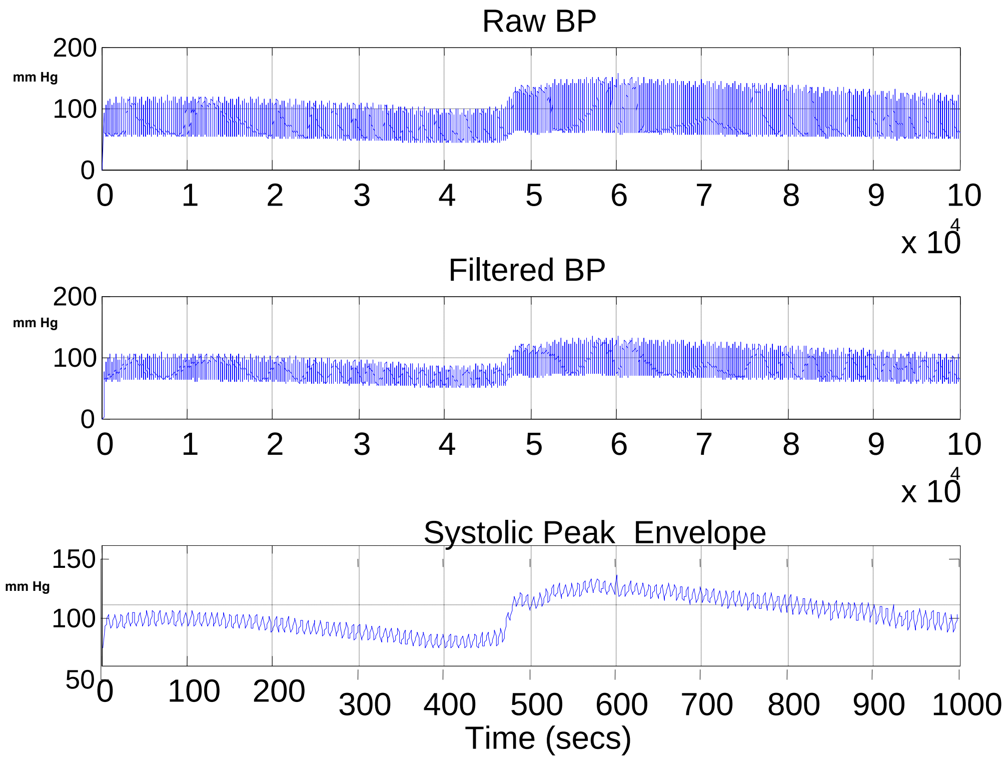


Figure 5.2.2-1. Systolic Peak detection from the BP Waveform.

The probabilistic alarm module generates the probabilistic alarm level by comparing the changes in each signal sample with similar changes from a population cluster. For implementing the probabilistic alarm module the most essential set of data required were the statistical data (i.e. average and standard deviation values) for each of the four physiological parameters (i.e. MAP, HR, PV and $ETCO_2$). Dr. Harrison had provided these statistical values for MAP, HR and $ETCO_2$ from a relevant study [13] by analysing a cluster of physiological signals from a large population sample. The statistical values were not available for the PV signal. During the data simulation stage these statistical data values were generated for PV signal from a relatively smaller population sample. Hence the PV probabilistic algorithm using these values was relatively less accurate as compared to the probabilistic algorithms for the other three signals. In the current form the PV signal generated a higher probabilistic alarm level as compared to the other three signals and thus the probabilistic alarm value from the PV module was scaled down. The resulting PV alarm level could then be used in conjunction with the probabilistic alarm values from the other three signals for computing the over all probabilistic alarm level for absolute hypovolaemia. It was expected that as more physiological data was collected the PV statistical values will be recalculated from a larger population sample. Thus the accuracy of the PV module could be further improved.

The diagnostic algorithms initially developed with MATLAB^{TM}, were later reprogrammed as LabVIEW^{TM} code. LabVIEW^{TM} was chosen because of its data acquisition, signals analysis and real-time application development capabilities. Initially Java^{TM} was considered for developing RT-SAAM but due to the compatibility issues with MATLAB^{TM} code and the lack of built-in data acquisition and signal processing capabilities it was decided that LabVIEW^{TM} was a more practicable approach for developing RT-SAAM. Familiarity and expertise was gained by self learning the required program codes and programming techniques in LabVIEW^{TM}.

The fuzzy logic algorithm was initially programmed in MATLAB^{TM} but LabVIEW^{TM} did not have a compatible object oriented programming (OOP) toolbox and hence the fuzzy logic module could not be reprogrammed in LabVIEW^{TM}. The complexity of the fuzzy module and the computational overhead without the OOP implementation would produce a large delay in the real-time analysis and hence it was not feasible to implement the fuzzy module in RT-SAAM. However using third party implementation of OOP for LabVIEW^{TM} it would be possible to do this. The learning curve for the third party OOP toolbox would require additional time and additional simulated testing and thus could not be accommodated within the allotted time duration of this thesis. The exclusion of the fuzzy logic from RT-SAAM had very little effect on the diagnostic capability of RT-SAAM as a whole. However reprogramming the fuzzy logic module would provide a redundant diagnosis thereby improving the overall strength of the diagnosis.

5.2.3. Real-Time System Development

The diagnostic alarm prototype implemented in the RT-SAAM is described in Chapter 4. RT-SAAM was developed using a modular programming approach in LabVIEW^{TM} environment, the programming block was sub-divided into modular blocks of code called Sub-Virtual Instruments (sub-VIs). The modular layout of SAAM and RT-SAAM facilitated addition of new program-blocks and depreciation of existing program-blocks with little or no effect on the functionality of the system as a whole.

The RT-SAAM was also provided with a data-acquisition module (for collecting data from the S/5 anaesthesia monitor) and a data relaying module (for relaying the data to other equipment i.e. IDAS) in the operating theatre. These capabilities of RT-SAAM enabled the data-collection and testing without affecting the operation of the existing equipment in the operating theatre. The development of this communication setup was one of the most time-consuming phases of this research project. Several communication setups were tried before successfully implementing the DOMonitor based communication protocol. A new communication module was added to RT-SAAM to enable it to communicate with the DOMonitor program. After implementing the communication protocol RT-SAAM was tested in an empty operating theatre before its successful implementation for real-time data collection.

In the current form RT-SAAM can be tested in real-time in the operating theatre and its clinical usefulness has been assessed in real-time tests. The real-time test capability and clinical usefulness of RT-SAAM distinguishes it from other diagnostic systems [6, 11, 17, 18] in anaesthesia. The results obtained from a the offline (retrospective) analysis and real-time analysis of the RT-SAAM during the system validation phase showed that there was a high level of agreement between the anaesthetist and the RT-SAAM pertaining to the diagnosis of the critical events. Section 5.2.5 in this chapter provides a detailed discussion on system performance, accuracy of diagnoses and the problems encountered during the system validation phase. However, it should be noted that the evaluation of RT-SAAM from computation cost point of view will be a big research objective in itself and is beyond the scope of this thesis. Once RT-SAAM's performance is validated and refined in clinical context then the computational utilisation of this diagnostic system can be evaluated and optimised.

5.2.4. Audio-Visual Display Design

One of the goals of this research project was to provide the diagnostic information on an ergonomic, interactive display to the users (i.e. the anaesthetists). Therefore anaesthetist's feedback was obtained while designing the front panel of the RT-SAAM. The aim was to minimise the visual clutter on the front panel and to display the diagnostic information in an easy to grasp format. Thus progress bars were used for indicating the alarm levels and waveform graphs were avoided. To notify about any critical events the alarm level progress bars were provided with blinking property and an audio alarm to get the user's attention. An audio alarm was implemented in the form of a human voice that asked the anaesthetist *"IS HYPOVOLEMIA POSSIBLE ?"*. Thus the anaesthetists were prompted to provide their diagnosis based on the present state of the patient. The interactive capability of RT-SAAM was useful for logging anaesthetists' responses and thus evaluating the system performance. Using LabVIEW^{TM} graphically rich, interactive displays can be easily produced and modified.

5.2.5. System Performance

For testing of the diagnostic system prototype in the operation theatre required ethical consent from the local ethics committees. Ethical approval was obtained from the Northern-X Regional Ethics Committee and Auckland University of Technology Ethics Committee (AUTEC). An informed, written consent was obtained from each patient participating in this research and strict clinical guidelines were followed during the real-time testing so that little or no interference was caused to the routine clinical proceedings.

The prototype system could be tested only with those patients who were likely to suffer moderate to major blood-loss and who required an arterial pressure line during the surgery. Emergency procedures were avoided because informed consent is more difficult in such cases. The recruitment of suitable patients was a time consuming task. At the time of publication of this thesis only 6 suitable research participants could be recruited and during the first clinical trial the data relaying module in DOMonitor did not function as anticipated and thus the data collection could not be continued. Hence RT-SAAM could be successfully tested with only 5 patients in real-time.

The prototype system was tested with physiological data from 18-patients in offline (retrospective) analysis and 5 patients in real-time analysis. The physiological data were divided into 15-minutes epochs and the diagnoses for each of these epochs were evaluated. For some of the 15-minutes epochs the data were noisy and thus the diagnoses were unavailable. A total of 259 epochs (i.e. approximately 65 hours of physiological data) were analysed. The useful diagnoses from the expert (anaesthetist) and the expert system prototype (RT-SAAM) were compared to measure the level of agreement between the two observers. Kappa analysis was used for computing the level of agreement between the two independent observers (i.e. Expert and RT-SAAM in the above case). During the offline tests and real-time tests substantial and moderate levels of agreements were achieved respectively. For some 15-minutes epochs RT-SAAM made diagnosis 5-10 minutes before the anaesthetist.

During the analysis of some 15-minutes epochs from the real-time data it was observed that the diagnostic system prototype generated false positive alarms. These epochs were re-analysed to determine the cause of these false positive alarms. During the re-analysis of the RT-SAAM system it was found that the data-communication module appended a zero value at the end of each data batch resulting in a large variation in the signals and thus generating false positive alarms. After removing the appended zeros from the data batches the false positive alarms were minimised.

Kappa analysis used in this research is a statistics based analysis that does not pre-judge that either of the observers is correct, simply measuring the level of agreement between the two observers. It is important as this research aims to improve on anaesthetists' diagnoses. It is likely that RT-SAAM may make the diagnosis when the clinician does not, in this case RT-SAAM could be correct and yet there is disagreement. This shows that even though kappa statistics can provide a measure of agreement between the two observers but does not completely represents the actual performance of RT-SAAM. However the diagnoses from the expert and RT-SAAM cannot be studied independently and hence the kappa analysis provides incomplete yet most reliable performance indicator.

Some important factors to be noted about the diagnostic analysis are listed below:

- In an ideal world the individual assessments should be made in different patients so that the data is completely independent. This would mean only having one assessment per patient which is infeasible, thus statistical results should be interpreted with this in consideration.
- The data-set analysed for timeslot 1-15 minutes is not completely independent of the data-set for timeslot 16-30 minutes (refer to Table 1 and Table 8 in Chapter 4). Thus the alarm level in any particular 15-minute epoch could be related occurring in the adjacent epochs, resulting in a disagreement.
- During the real-time tests it was observed that the normal changes in BP, HR, $ETCO_2$ were not detected as significant changes by the diagnostic algorithms. Hence it could be inferred that the diagnostic algorithms detected only the significant (genuine) changes in the physiological parameters as critical events.
- The measurement of actual blood volume is extremely difficult and complex and therefore we have had to depend on clinical expertise for an assessment. Thus there was no gold standard measure for comparing the output of the diagnostic algorithm.

It is expected that more clinical trials will be conducted in the operating theatre in the near future and the performance of the system will be further refined. The results from the clinical trials will be used for making appropriate fine-tuning of the system components for making the diagnosis more accurate.

5.3. Conclusion

In summary, this research has developed a clinically useful diagnostic alarm for anaesthesia monitoring by investigating various topics in digital signal processing, data communication techniques, data collection from anaesthetised patients in the operating theatre. The complete clinical usefulness of this system however needs to be validated through more rigorous testing in the real test environment and is a time consuming process. Fuchs-Buder[83] discusses evaluation guidelines for monitoring/diagnostics systems like RT-SAAM. Another aspect of the system validation which needs further investigation is the comparison in the performance of the various diagnostic modules (Probabilistic, SPV and Fuzzy). This will give more insight into the accuracy of the individual algorithms and hence the better understanding of the overall accuracy of RT-

SAAM. In its current state the system is ready for clinical use as a test bed for validating the diagnostic algorithms presented in this thesis. The complete validation of RT-SAAM as a clinically useful diagnostic alarm system and evaluation of the diagnostic accuracy and specificity of the individual algorithm is a time consuming undertaking and will be continued.

The area that had been covered during this research project included:

- Biomedical signal processing algorithms that used principles of digital signal processing in DFT, FFT, Windowing.
- Understanding the physiological signals and the anaesthesia monitoring techniques used by the anaesthetists.
- Real-time algorithm development using the state-engine approach.
- Diagnostic algorithms based on fuzzy logic, statistics and the respiratory induced systolic pressure variations (SPV).
- Protocols for collecting and relaying the physiological signals.

The results produced during this research showed that the developed diagnostic system (RT-SAAM) is capable of diagnosing the pathological events with substantial to fair level of agreement between RT-SAAM and the anaesthetist. Thus RT-SAAM is potentially promising and with further research the monitoring system could be commercialised.

5.4. Future Work

This research project achieved approximately 80% of the research objectives. The recommendations for future work provided in the following paragraphs are towards improving the RT-SAAM developed during this research project followed by other possible applications to which RT-SAAM can be extended.

5.4.1. Fuzzy Logic Module

Fuzzy logic can be implemented in RT-SAAM using third party OOP toolbox for LabVIEW^{TM}.

5.4.2. Refining RT-SAAM

Further real-time testing and data collection of RT-SAAM will provide more fruitful results and constructive feedback on the basis of which the system can be fine tuned for obtaining optimal performance. As more data is available after more data-collection sessions the BP and PV variance threshold values could be reanalysed thereby improving the accuracy of the pre-processing algorithms in RT-SAAM. The average and standard deviation values for the PV signal also need to be obtained for a larger population sample and with future data collection and clinical trials this can be easily achieved.

5.4.3. Expanding RT-SAAM for Monitoring Additional Events

After complete validation of RT-SAAM for diagnosing absolute hypovolaemia, the system can be further expanded for monitoring more anaesthesia related events like relative hypovolaemia, sympathetic response and malignant hyperpyrexia.

5.4.4. Data Acquisition, Apparatus and Setups

A touch screen implementation of RT-SAAM with wireless communication capability can make the system more ergonomic and reduce the complexity involved in deploying the setup in the operating theatre. Using touch screen and wireless setup will reduce the clutter in the operating theatre creating a safer work place.

5.4.5. Patient Database for Physiological Data and Events

Developing a database of the patient data for future research in the field of patient monitoring could highly benefit future projects in this field.

5.4.6. Application to General Patient Monitoring

Some of the signal processing algorithms developed during this project can be used in other patient monitoring systems where similar physiological signals need to be processed. Thus it is possible to extend the diagnostic capability of RT-SAAM to other patient monitoring applications. More research can be done in this area.

5.4.7. Comparative Performance Validation of Diagnostic Algorithms

A comparative validation and more intense performance study of the diagnostic algorithms will prove helpful in recognising and selecting more accurate and specific algorithm. Such a study will prove to be a useful aid in identifying and refining these algorithms and thereby improving the RT-SAAM diagnostic system as a whole.

5.4.8. Complete Performance Validation of the System

A complete, intensive performance validation of RT-SAAM is required for determining for proving the clinical usefulness of this diagnostic system. It is proposed that such a validation should be carried out to determine RT-SAAMs performance with reference to sensitivity, real-time analysis, hum-factors engineering, etc.

References

[1] K. Krishnamurthy and D. T. Ward, "Intelligent systems for autonomous aircraft," 2000.

[2] J. A. Momoh, D. Sobajic, and J. Dolce, "An evaluation of intelligent systems for fault diagnosis," presented at IEEE International Conference on Systems, Man, and Cybernetics, 1994.

[3] T. Fukuda and N. Kubota, "Computational intelligence in robotics and mechatronics," presented at 23rd International Conference on Industrial Electronics, Control and Instrumentation, IECON 97, 1997.

[4] R. R. Hoffman and J. F. Yates, "Decision making [human-centred computing]," *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications]*, vol. 20, pp. 76-83, 2005.

[5] A. Lowe and M. J. Harrison, "Computer-enhanced diagnosis of malignant hyperpyrexia," *Anaesthesia and Intensive Care [NLM - MEDLINE]*, vol. 27, pp. 41, 1999.

[6] S. R. Hameroff, M. J. Navabi, R. C. Watt, and K. C. Mylrea, "Smart Alarms In Anesthesia Heart Rate And ECG Monitoring And Event Recognition Using Neural Network And Algorithmic Methods," presented at International Conference of the IEEE Engineering in Medical and Biological Society, 1990.

[7] L. G. Deneault, C. M. Lewis, A. Debons, K. L. Stein, and A. M. Dewolf, "An integrative display for patient monitoring," presented at Proceedings of the IEEE International Conference in Systems, Man and Cybernetics, Los Angeles, CA 1990.

[8] G. C. Van den Eijkel and E. Backer, "Knowledge acquisition using a fuzzy machine-learning algorithm for a knowledge-based anesthesia monitor," presented at 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Amsterdam, 1996.

[9] B. C. Jeffrey, S. N. Ronald, D. L. Charlene, and M. Bucknam, "Preventable anesthesia mishaps: A study of human factors," *Quality & Safety in Health Care*, vol. 11, pp. 277, 2002.

[10] K. Becker, H. Kasmacher, G. Rau, G. Kalff, and H. J. Zimmermann, "A fuzzy logic approach to intelligent alarms in cardioanesthesia," presented at Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Orlando, FL 1994.

[11] A. Lowe, "Evidential Inference for Fault Diagnosis," in *Engineering*. Auckland: University of Auckland, 1998, pp. 217.

[12] F. Steimann, "The interpretation of time-varying data with -1," *Artificial Intelligence in Medicine*, vol. 8, pp. 343-357, 1996.

[13] C. C. MJ Harrison, "Probabilistic alarms from sequential physiological measurements " *Medical Sciences Congress* pp. 61, 2005.

[14] W. W. Melek, Z. Lu, A. Kapps, and W. D. Fraser, "Comparison of trend detection algorithms in the analysis of physiological time-series data," *IEEE Transactions on Biomedical Engineering*, vol. 52, pp. 639-651, 2005.

[15] M. J. Harrison, "The enhancement of intra-operative diagnostics and decision-making using computational methods," in *Department of Anaesthesiology*. Auckland: University of Auckland, 2005.

[16] I. J. Haimowitz, "Intelligent diagnostic monitoring using trend templates," *Proceedings of the 18th Annual Symposium on Computer Applications in Medical Care*, pp. 702-8, 1994.

[17] M. Elkfafi, J. S. Shieh, D. A. Linkens, and J. E. Peacock, "Intelligent signal processing of evoked potentials for anaesthesia monitoring and control," *Control Theory and Applications, IEEE Proceedings*, vol. 144, pp. 354-360, 1997.

[18] E. M. Koski, T. Sukuvaara, A. Makivirta, and A. Kari, "A knowledge-based alarm system for monitoring cardiac operated patients-assessment of clinical performance," *International Journal of Clinical Monitoring and Computing*, vol. 11, pp. 79-83, 1994.

[19] H. G. Hosseini and Z. Krechowec, "Facial expression analysis for estimating patient's emotional states in RPMS," presented at 26th Annual International Conference of the IEEE in Engineering in Medicine and Biology Society 2004.

[20] H. G. Hosseini, M. Fatemi, and A. Alizad, "Registration of Vibro-acoustography Images and X-ray Mammography," presented at
27th Annual International Conference of the IEEE in Engineering in Medicine and Biology Society, 2005.

[21] N. V. Kalpakam and S. Venkataramanan, "EEG signal processing for modern wireless patient monitoring," presented at 27th Annual International Conference in Engineering in Medicine and Biology Society, 2005.

[22] R. C. Watt, E. S. Maslana, and K. C. Mylrea, "Alarms and anesthesia: challenges in design of intelligent systems for patient monitoring," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 12, pp. 34-41, 1993.

[23] C. Chi-Hao, "Signal processing for optical communication," *Signal Processing Magazine, IEEE*, vol. 23, pp. 88-96, 2006.

[24] H. He, C. Shijie, Z. Youbing, and J. Nguimbis, "Home network power-line communication signal processing based on wavelet packet analysis," *IEEE Transactions on Power Delivery*, vol. 20, pp. 1879-1885, 2005.

[25] E. H. Shortliffe, B. G. Buchanan, and E. A. Feigenbaum, "Knowledge engineering for medical decision making: A review of computer-based clinical decision aids," *Proceedings of the IEEE*, vol. 67, pp. 1207-1224, 1979.

[26] J. Xue and M. Krajnak, "Fuzzy Expert Systems For Sequential Pattern Recognition For Patient Status Monitoring in Operating Room," presented at 28th Annual International Conference of the IEEE in Engineering in Medicine and Biology Society, 2006.

[27] S. M. Mohiddin, "Medical differential diagnostic system (DDS) using fuzzy inferencing methods," presented at AI in Medical Decision Making, IEE Colloquium, 1990.

[28] D. Anthony, E. Hines, D. Taylor, and J. Barham, "An investigation into the use of neural networks for an expert system in nuclear medicine image analysis," presented at Third International Conference in Image Processing and its Applications, 1989.

[29] E. Czogala, J. Chmielniak, A. Brodziak, and W. Siler, "Decision making by means of probabilistic sets in medical expert systems," presented at Proceedings of the Annual International Conference of the IEEE in Engineering in Medicine and Biology Society, 1988.

[30] D. A. Linkens and M. F. Abbod, "Intelligent control of anaesthesia," presented at Intelligent Methods in Healthcare and Medical Applications (Digest No. 1998/514), IEE Colloquium, York 1998.

[31] "IEEE Colloquium on `Expert Systems for Fault Diagnosis in Engineering Applications' (Digest No.64)," 1989.

[32] S. Mahalingam, "CSRI-an expert system tool and its applications," presented at Proceedings of the IEEE 1989 National Aerospace and Electronics Conference, 1989.

[33] K. Kaarela, R. Korhonen, H. Huotari, and K. Autio, "An embedded expert system for performance monitoring of process stations in a distributed process automation system," presented at Proceedings of the International Workshop of the IEEE in Artificial Intelligence for Industrial Applications, 1988.

[34] R. W. Jones, M. J. Harrison, and A. Lowe, "Computerised anaesthesia monitoring using fuzzy trend templates," *Artificial Intelligence in Medicine*, vol. 21, pp. 247-251, 2001.

[35] G. C. van den Eijkel, van der Lubbe, C.A., and E. Backer, "Fuzzy Incremental Learning of Expert Rules for a Knowledge-Based Anaesthesia Monitor.," presented at Fourth European Congress on Intelligent Techniques and Soft Computing, Germany, 1996.

[36] J. Reason, "Safety in the operating theatre - Part 2: Human error and organisational failure," *Quality Safe Health Care*, vol. 14, pp. 56-60, 2005.

[37] M. Krol and D. L. Reich, "The algorithm for detecting critical conditions during anesthesia," presented at 12th IEEE Symposium on Computer-Based Medical Systems (CBMS'99), 1999.

[38] M. Imhoff, Kuhls, S. , "Alarm Algorithms in Critical Care Monitoring," *International Anesthesia Research Society*, pp. 1525–1537, 2006.

[39] T. Sukuvaara, E. M. Koski, A. Makivirta, and A. Kari, "A knowledge-based alarm system for monitoring cardiac operated patients--technical construction and evaluation," *International Journal of Clinical Monitoring and Computing*, vol. 10, pp. 117-26, 1993.

[40] M. T. K. M.J. Harrison, N.N. Robertson, "The relationship between change in blood pressure, blood pressure and time.," *Anaesthesia* vol. 55, pp. 385-387, 2000.

[41] G. S. Leonard C. Jenkins, "Central Venous Pressure Monitoring in Anaesthesia," *Canadian Anaesthetic Society Journal*, vol. 13, 1966.

[42] J. M. Capek and R. J. Roy, "Noninvasive measurement of cardiac output using partial CO2 rebreathing," *IEEE Transactions on Biomedical Engineering*, vol. 35, pp. 653-661, 1988.

[43] S. Faddy, J. Boland, and D. W. M. Muller, "Accuracy and reliability of non-invasive cardiac output: the future in cardiology?," presented at Computers in Cardiology, 2003.

[44] P. Grant and O. Naesh, "Fuzzy logic and decision-making in anaesthetics," *Journal of the Royal Society of Medicine*, vol. 98, pp. 7-9, 2005.

[45] G. Cantor, "German mathematician and philosopher.," (1845-1918).

[46] R. Thomopoulos, P. Buche, and O. Haemmerle, "Fuzzy Sets Defined on a Hierarchical Domain," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, pp. 1397-1410, 2006.

[47] L. A. Zadeh, "Fuzzy Sets," *Information and Control*, vol. 8, pp. 338-353, 1965.

[48] V. Kecman, *Learning and Soft Computing, Support Vector Machines, Neural Networks and Fuzzy Logic*. Cambridge, Massachusetts: The MIT Press, 2001.

[49] L. A. Zadeh, "Fuzzy logic = computing with words," *IEEE Transactions on Fuzzy Systems*, vol. 4, pp. 103-111, 1996.

[50] L. A. Zadeh, "Soft computing and fuzzy logic," *Software, IEEE*, vol. 11, pp. 48-56, 1994.

[51] R. N. Naguib, Sherbet, G.V.,, *Artificial Neural Networks in Cancer Diagnosis, Prognosis and Patient Management*. London: CRC, 2001.

[52] Y. Yingjie, C. Hinde, and D. Gillingwater, "A new method for explaining neural network reasoning," presented at Proceedings of the International Joint Conference on Neural Networks, 2003, 2003.

[53] S. L. David Hosmer, *Applied Logistic Regression*, vol. 2. New York, 2000.

[54] E. Sutanto, K. Warwick, and M. Griffin, "Application of logistic regression for fault analysis in an industrial printing process," Proceedings of the Fourth Annual IEEE Symposium in Computer-Based Medical Systems, 1992.

[55] L. P. Haimowitz IJ, Kohane IS. , "Clinical monitoring using regression-based trend templates.," *Artificial Intelligence in Medicine*, vol. 7, pp. 473-96.

[56] T. Tang and T. Wang, "An ANN-based clustering analysis algorithm with dynamic data window," presented at International Conference in Control and Automation, 2005., 2005.

[57] E. N. Marieb, *Human Anatomy and Physiology*, vol. 4, 1998.

[58] I. NorthPoint Domain, "Pulse Volume Recording," vol. 2007, 2007.

[59] T. D. Jardins, *Cardiopulmonary Anatomy & Physiological*, 4 ed, 2001.

[60] *Non-Invasive Respiratory Support*. London, UK: Chapman and Hall Medical, 1996.

[61] Y. C. Lai HY, Cheng CF, Huang FY, Lee Y, Shyr MH, Kuo TB., "Effect of esmolol on positive-pressure ventilation-induced variations of arterial pressure in anaesthetized humans," *Clinical science (London, England : 1979)*, pp. 303-308.

[62] S. H. Rooke GA, Shapira Y., "The effect of graded hemorrhage and intravascular volume replacement on systolic pressure variation in humans during mechanical and spontaneous ventilation.," *Anesthesia Analgesia*, pp. 670.

[63] S. Magder, "Clinical Usefulness of Respiratory Variations in Arterial Pressure," *American Journal of Respiratory and Critical Care Medicine*, vol. 169, pp. 151-155, 2004.

[64] M. Ornstein PhD, Associate Professor of Clinical Anesthesiology, Eugene, S. L. i. A. Eidelman MD, Leonid A., S. L. i. A. Drenger MD, Benjamin, L. i. S. Elami MD, Amir, and S. L. i. A. Pizov MD, Reuven, "Systolic Pressure Variation Predicts the Response to Acute Blood Loss," *Journal of Clinical Anesthesia*, vol. 10, pp. 137, 1998.

[65] N. Dalibon, T. Guenoun, D. Journois, J. Frappier, D. Safran, and M. Fischler, "The clinical relevance of systolic pressure variations in anesthetized nonhypotensive patients," *Journal of Cardiothoracic and Vascular Anesthesia*, vol. 17, pp. 188, 2003.

[66] M. S. Bhende, "End-tidal carbon dioxide monitoring in pediatrics: concepts and technology.," vol. 47, 2001, pp. 153-156.

[67] S. M. Bradley, Simsic, J.M., ATZ Andrew M., Myers J.L.,, "Hemodynamic effects of inspired carbon dioxide after the Norwood procedure. Commentary," in *The Annals of thoracic surgery*, vol. 72, 2001, pp. 2088-2094

[68] A. L. MJ Harrison, RW Jones, "Assessment of heart function using fuzzy templates and evidence based reasoning to process data during anaesthesia," presented at ICSC Symposia On Engineering of Intelligent Systems, 2000.

[69] M. Harrison, "Accentuating the (true) positives and eliminating the (false) negatives in intra-operative real time diagnostics and decision-making." Auckland, New Zealand, 2003.

[70] S. K. Mitra, *Digital Signal Processing*, Second ed, 2001.

[71] E. C. Ifeachor, *Digital Signal Processing* vol. Second Edition. London: Prentice Hall, 2002.

[72] A. Noble, R. Johnson, A. Thomas, and P. Bass, *The Cardiovascular System*. Edinburg ; New York: Elsevier Churchill Livingstone, 2005.

[73] V. X. Afonso, Tompkins, W.J., Nguyen, T.Q., Shen, Luo, , "ECG beat detection using filter banks " *IEEE Trans. Biomedical Engineering*, vol. 46, pp. 192 - 202, 1999.

[74] A. L. Jacobson, "Auto-threshold peak detection in physiological signals," 2001.

[75] B. Köhler, Hennig, C., Orglmeister, R.,, "The Principles of Software QRS Detection," *IEEE ENGINEERING IN MEDICINE AND BIOLOGY*, pp. 42-57.

[76] W. E. M. Beverly C.M., Thomas F.H., Edward W.C., Warren G.G., , "Hemodymnamic effects of intermittent positive pressure ventilation respiration," *Anesthsiology*, pp. 584-590, 1966.

[77] Eltima.com, "Eltima Serial Splitter ", 2006.

[78] R. Bitter, Mohiuddin, T., Nawrocki M., *LabVIEW: Advanced Programming Techniques*: CRC, 2000.

[79] H. L. Kundel, Polansky, M., "Measurement of observer agreement.," *Radiology*, pp. 303-308, 2003.

[80] T. Sukuvaara, A. Makivirta, A. Kari, and E. Koski, "An intelligent intensive care alarming system," presented at Proceedings of Computers in Cardiology, 1989.

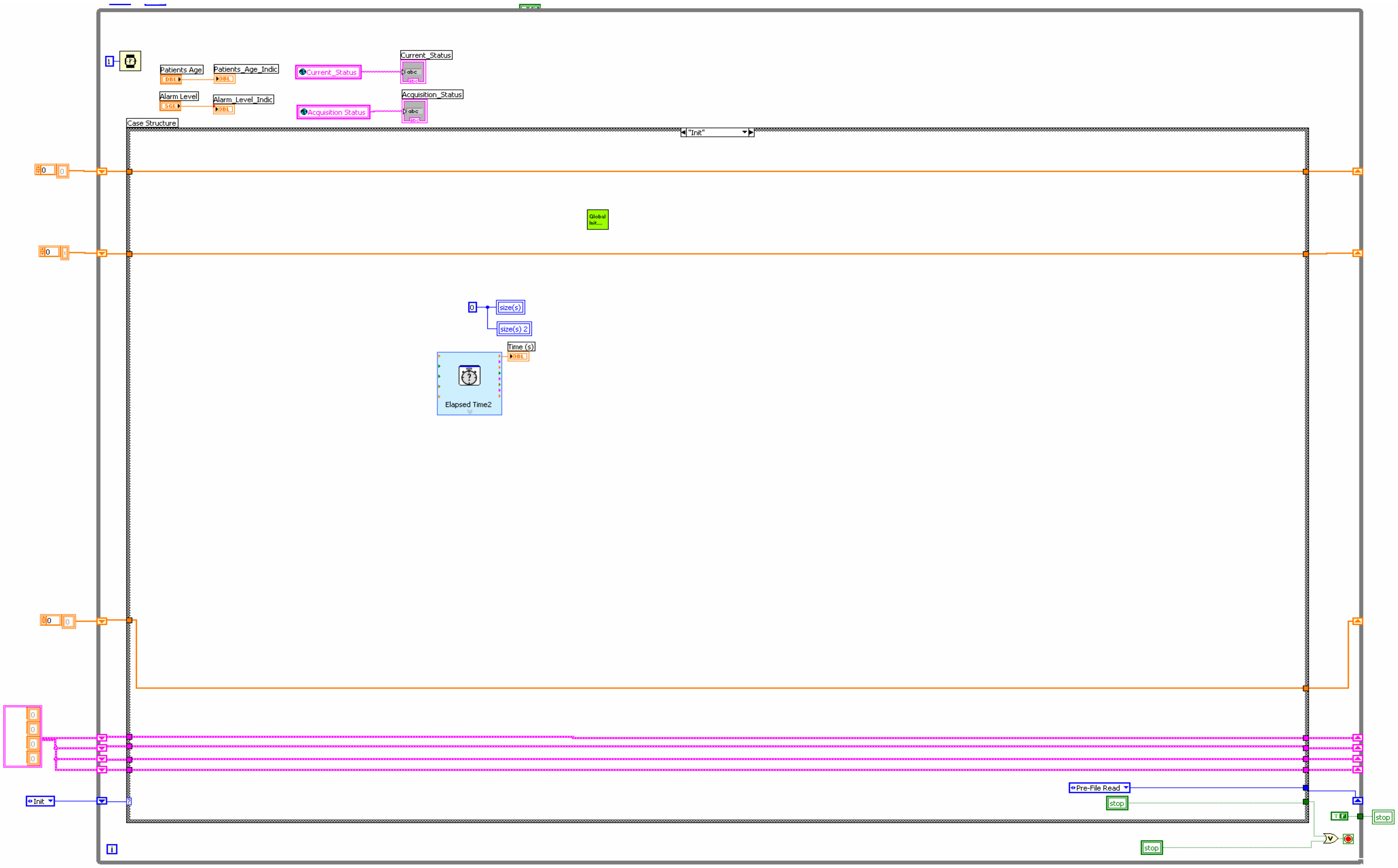
[81] K. G. Landis JR, "The measurement of observer agreement for categorical data.," *MEDLINE*, vol. 33(1), pp. 159-174, 1977.

[82] J. Pandey and D. Bajpai, "Developmental design of a rule based expert system for diagnosis," presented at Engineering in Medicine and Biology Society, 1995 and

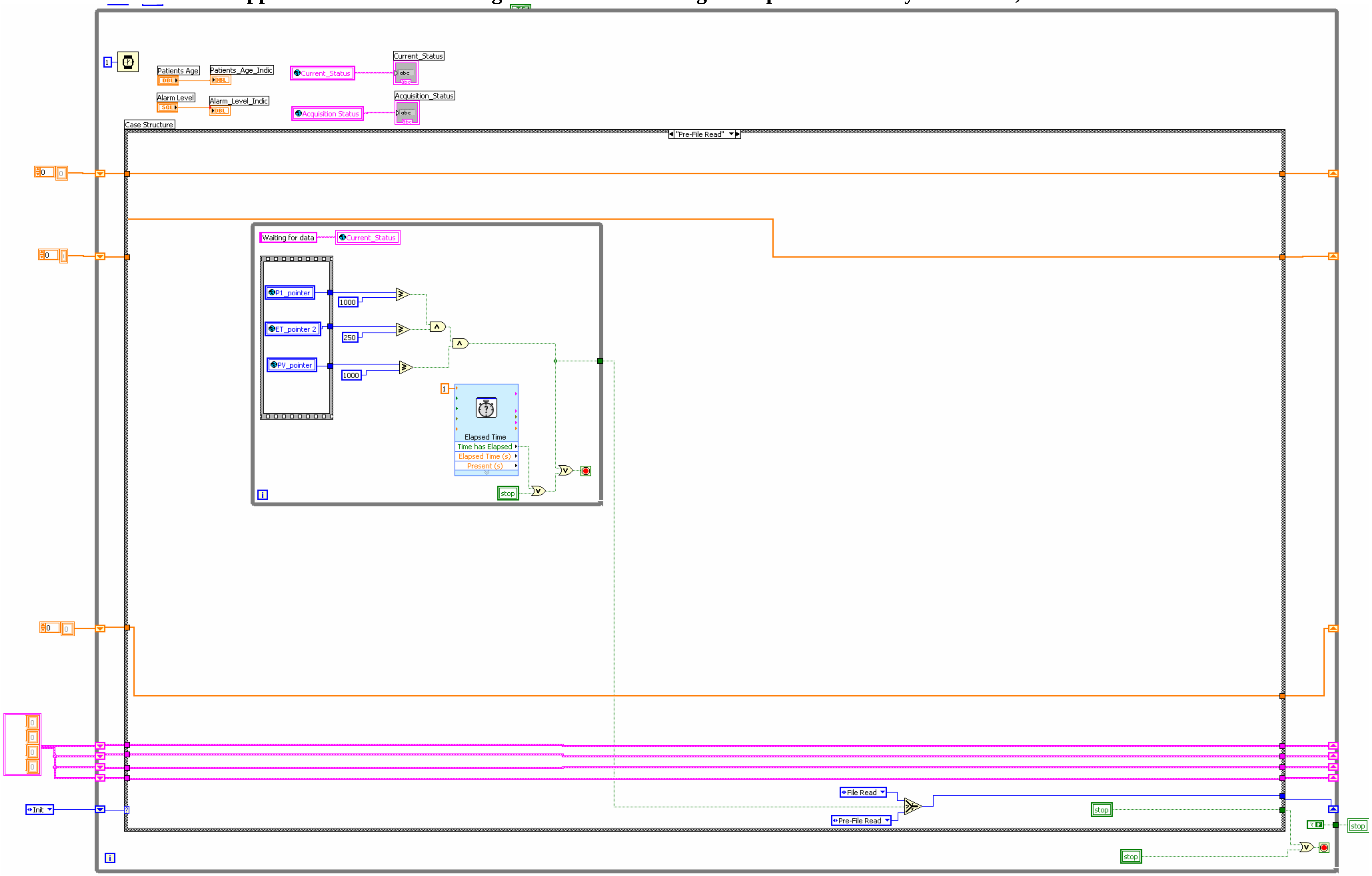
14th Conference of the Biomedical Engineering Society of India. An International Meeting, Proceedings of the First Regional Conference., IEEE, 1995.

[83] F.-B. Thomas, "Intraoperative Monitoring: What is necessary.," in *European Soceity of Anaesthesiologists.*, 2003.

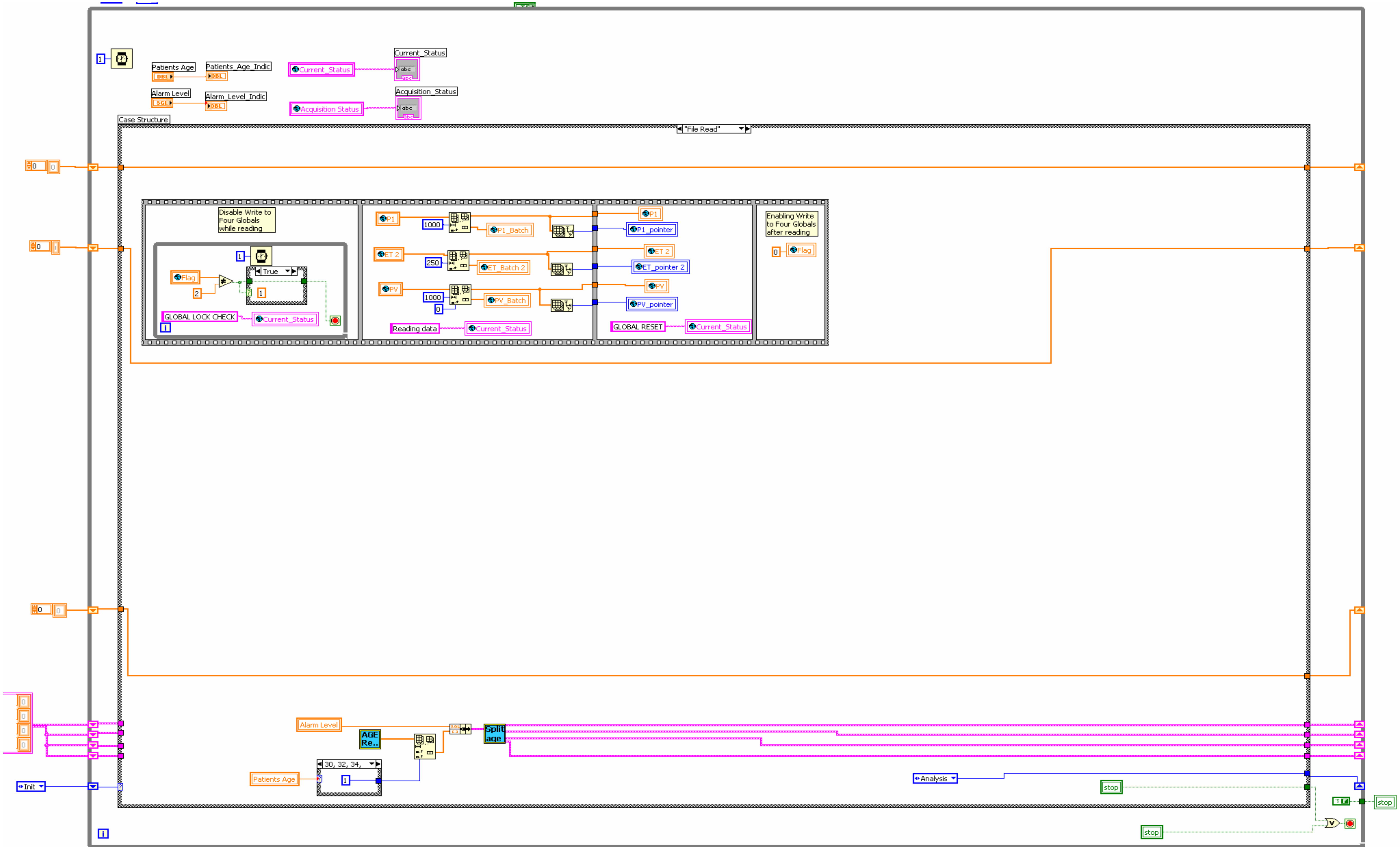
Appendix A1 - Real-Time Algorithm with State-Engine Implementation by LabVIEW, Init-State.



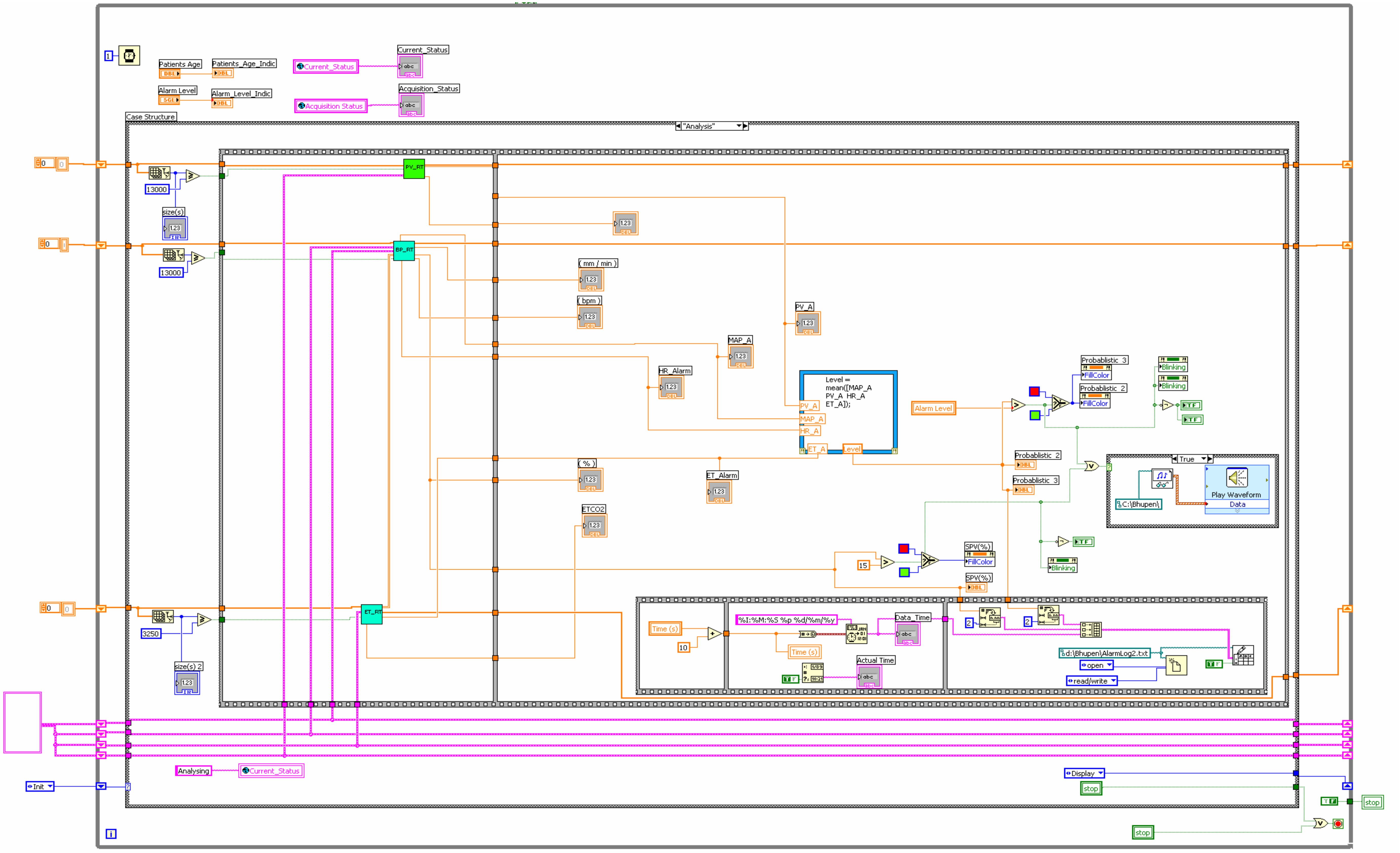
Appendix A2 - Real-Time Algorithm with State-Engine Implementation by LabVIEW, PreFileRead-State.



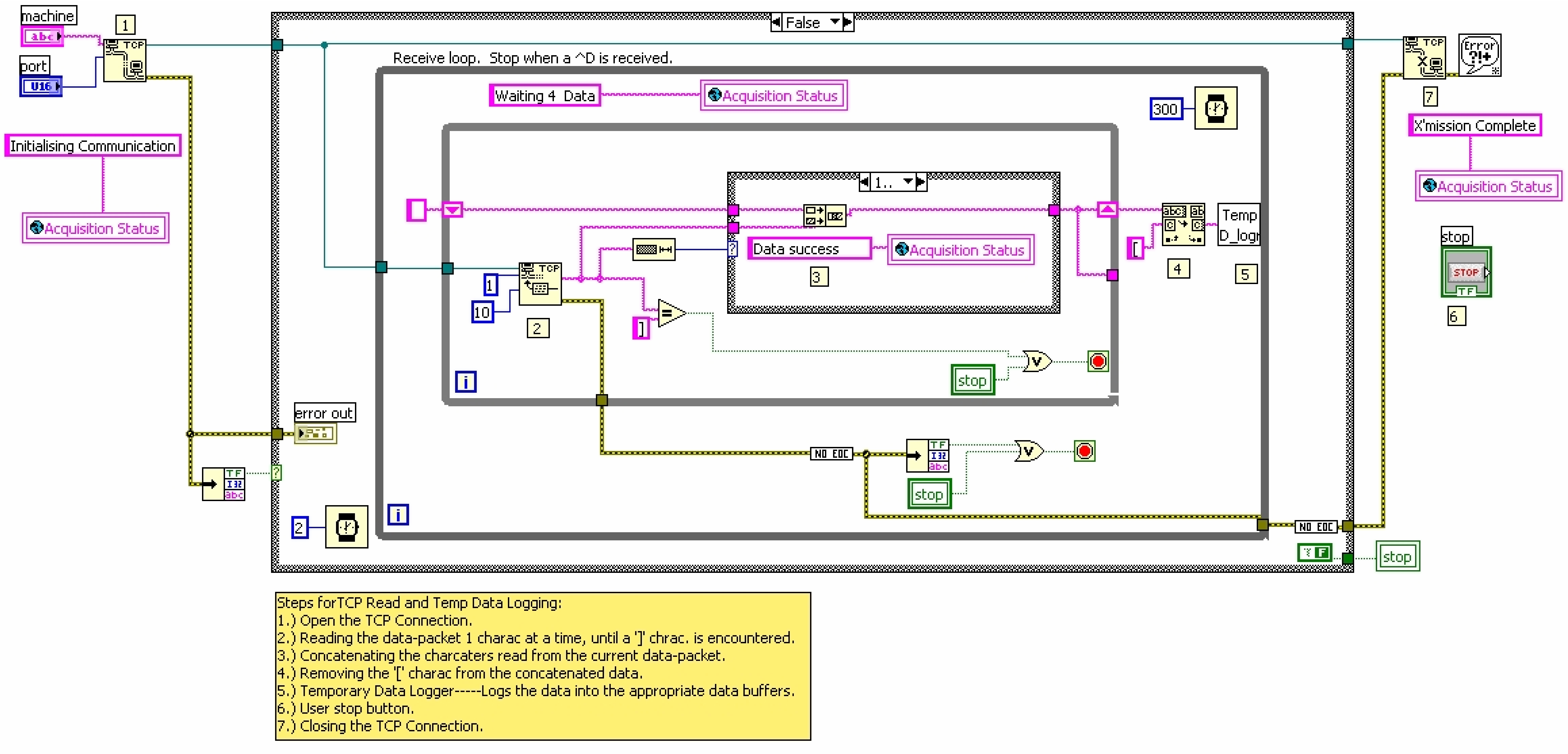
Appendix A3 - Real-Time Algorithm with State-Engine Implementation by LabVIEW, FileRead-State.



Appendix A4 - Real-Time Algorithm with State-Engine Implementation by LabVIEW, Analysis-State.

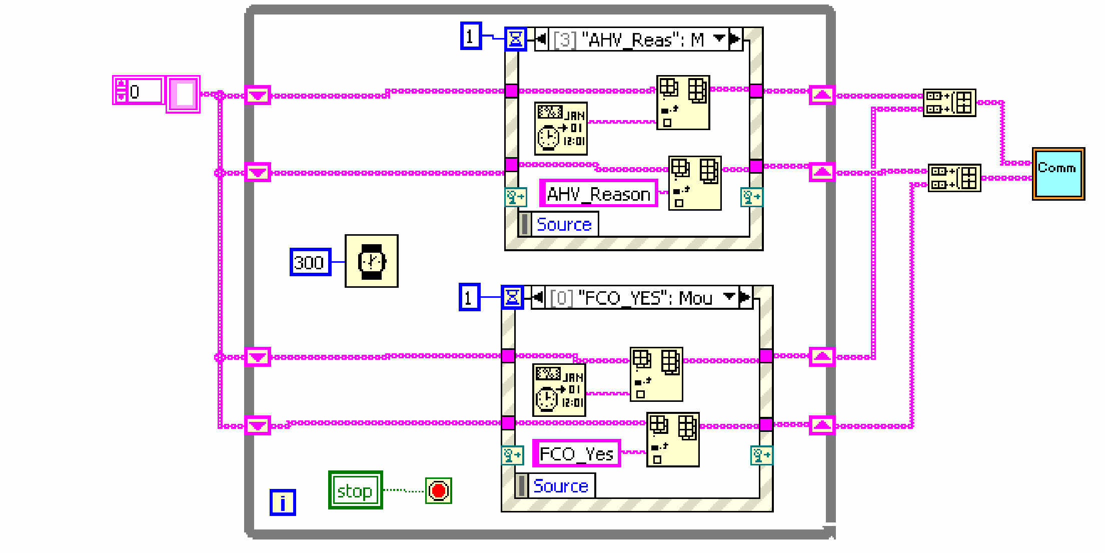


Appendix A5 - Real-Time Algorithm for TCP-Read by LabVIEW.

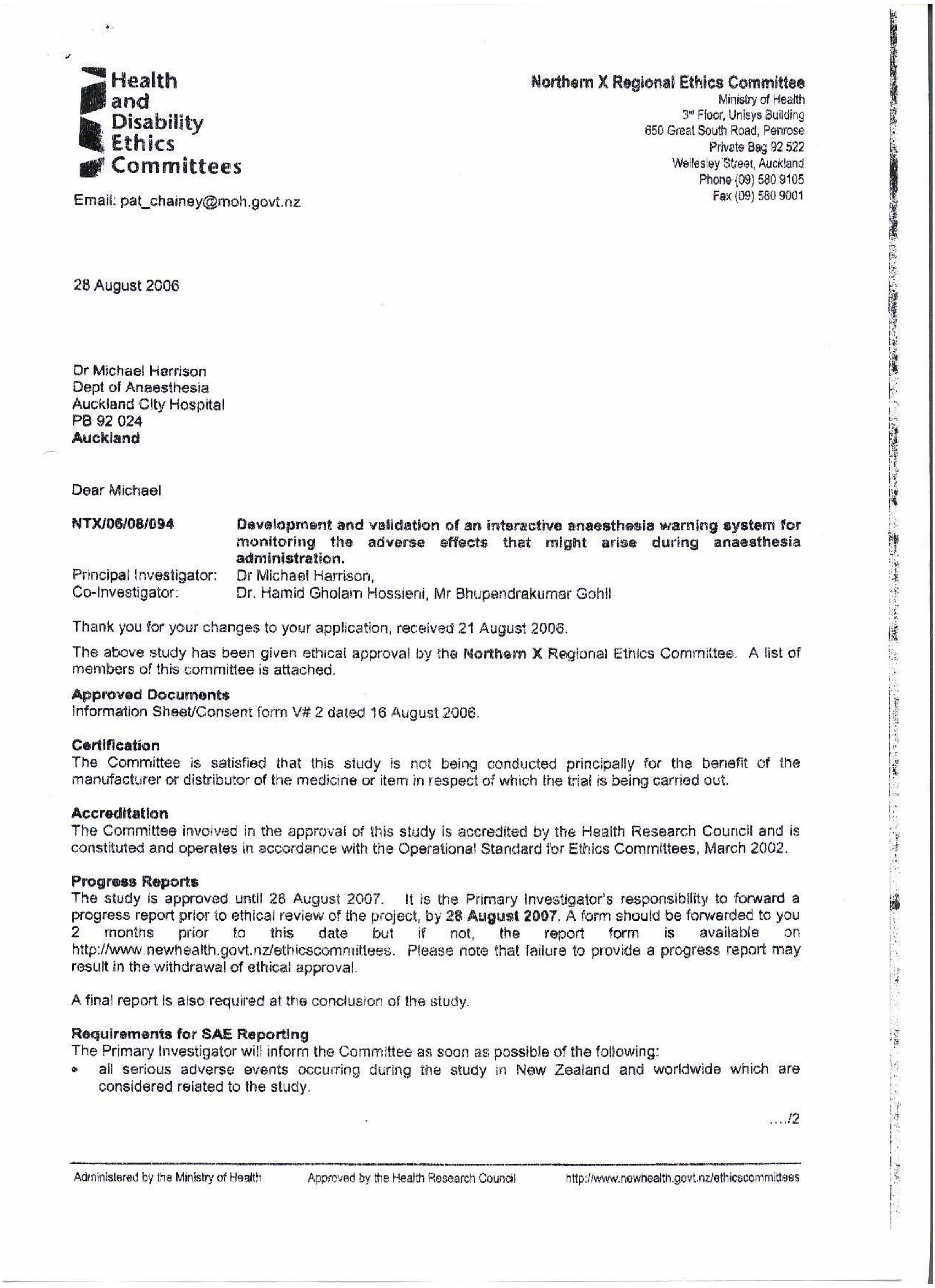


Steps forTCP Read and Temp Data Logging:
1.) Open the TCP Connection.
2.) Reading the data-packet 1 charac at a time, until a ']' chrac. is encountered.
3.) Concatenating the charcaters read from the current data-packet.
4.) Removing the '[' charac from the concatenated data.
5.) Temporary Data Logger-----Logs the data into the appropriate data buffers.
6.) User stop button.
7.) Closing the TCP Connection.

Appendix A6 - Real-Time Algorithm for Comment Logging by LabVIEW.



Appendix B1 - Northern X-Ethics Approval Letters.



Health
and
Disability
Ethics
Committees

Email: pat_chainey@moh.govt.nz

Northern X Regional Ethics Committee
Ministry of Health
3rd Floor, Unisys Building
650 Great South Road, Penrose
Private Bag 92 522
Wellesley Street, Auckland
Phone (09) 580 9105
Fax (09) 580 9001

28 August 2006

Dr Michael Harrison
Dept of Anaesthesia
Auckland City Hospital
PB 92 024
Auckland

Dear Michael

NTX/06/08/094 **Development and validation of an interactive anaesthesia warning system for monitoring the adverse effects that might arise during anaesthesia administration.**

Principal Investigator: Dr Michael Harrison,
Co-Investigator: Dr. Hamid Gholam Hossieni, Mr Bhupendrakumar Gohil

Thank you for your changes to your application, received 21 August 2006.

The above study has been given ethical approval by the **Northern X** Regional Ethics Committee. A list of members of this committee is attached.

Approved Documents
Information Sheet/Consent form V# 2 dated 16 August 2006.

Certification
The Committee is satisfied that this study is not being conducted principally for the benefit of the manufacturer or distributor of the medicine or item in respect of which the trial is being carried out.

Accreditation
The Committee involved in the approval of this study is accredited by the Health Research Council and is constituted and operates in accordance with the Operational Standard for Ethics Committees, March 2002.

Progress Reports
The study is approved until 28 August 2007. It is the Primary Investigator's responsibility to forward a progress report prior to ethical review of the project, by **28 August 2007**. A form should be forwarded to you 2 months prior to this date but if not, the report form is available on http://www.newhealth.govt.nz/ethicscommittees. Please note that failure to provide a progress report may result in the withdrawal of ethical approval.

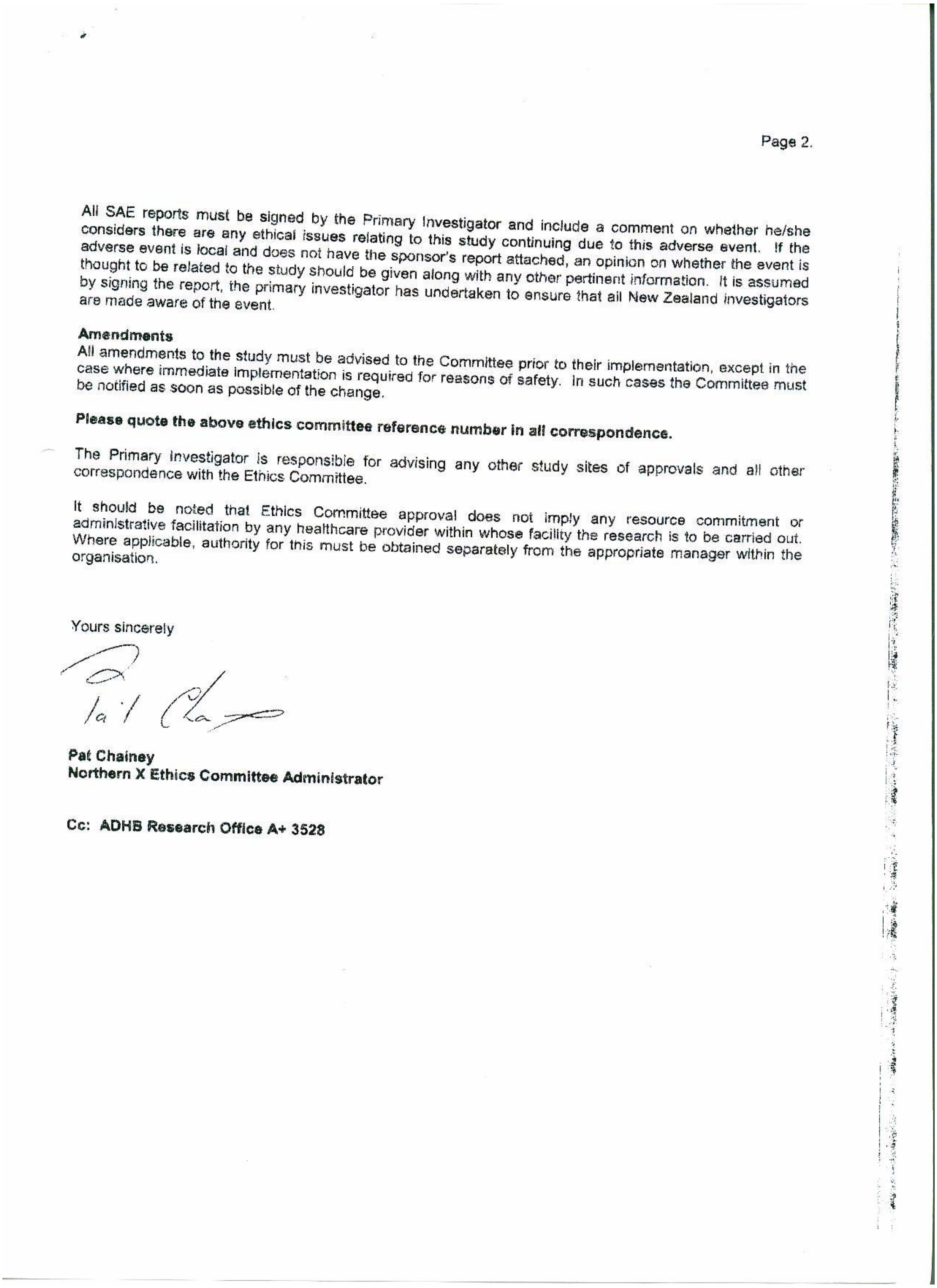
A final report is also required at the conclusion of the study.

Requirements for SAE Reporting
The Primary Investigator will inform the Committee as soon as possible of the following:

- all serious adverse events occurring during the study in New Zealand and worldwide which are considered related to the study.

…./2

Administered by the Ministry of Health Approved by the Health Research Council http://www.newhealth.govt.nz/ethicscommittees



Page 2.

All SAE reports must be signed by the Primary Investigator and include a comment on whether he/she considers there are any ethical issues relating to this study continuing due to this adverse event. If the adverse event is local and does not have the sponsor's report attached, an opinion on whether the event is thought to be related to the study should be given along with any other pertinent information. It is assumed by signing the report, the primary investigator has undertaken to ensure that all New Zealand investigators are made aware of the event.

Amendments
All amendments to the study must be advised to the Committee prior to their implementation, except in the case where immediate implementation is required for reasons of safety. In such cases the Committee must be notified as soon as possible of the change.

Please quote the above ethics committee reference number in all correspondence.

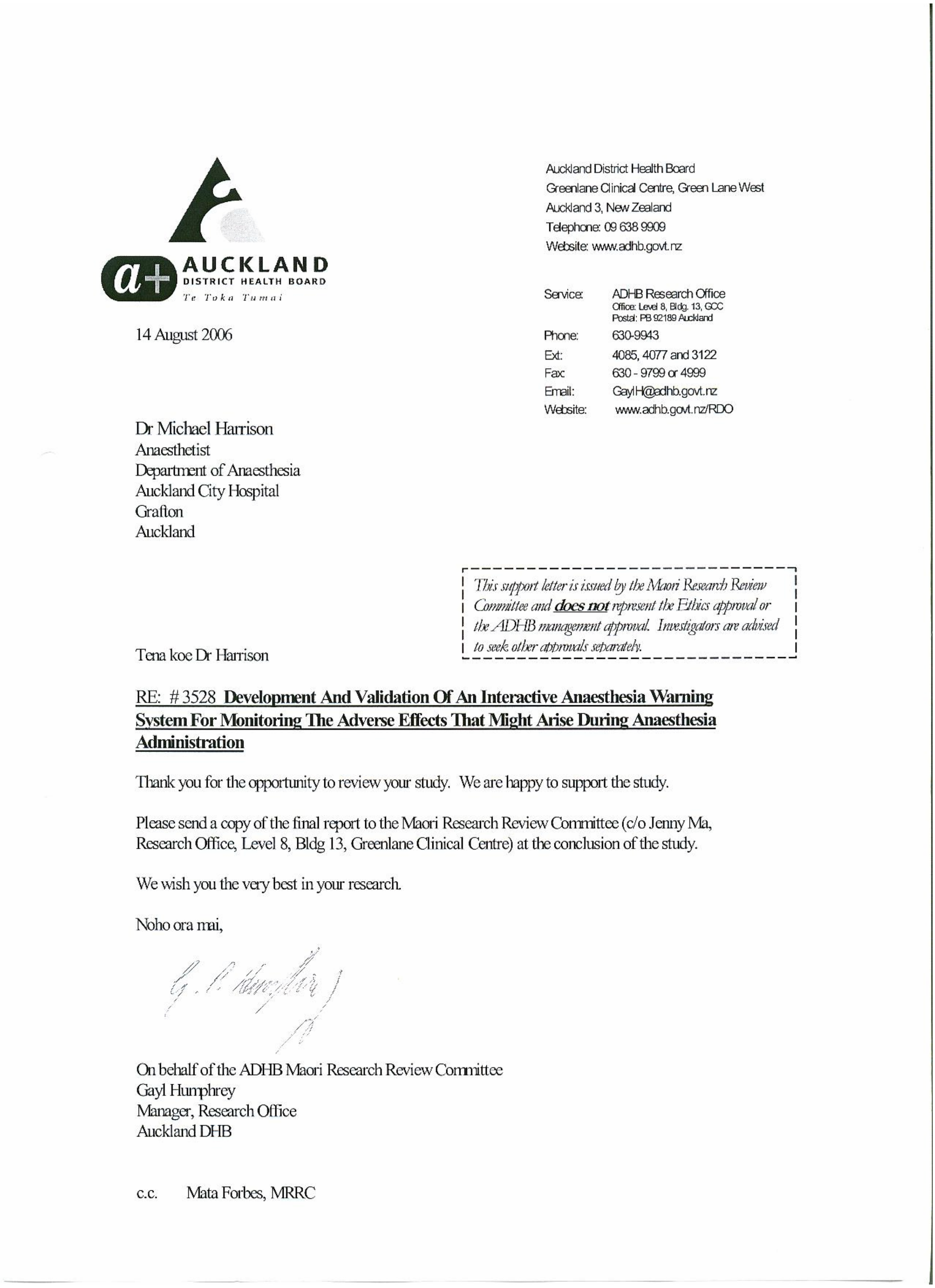
The Primary Investigator is responsible for advising any other study sites of approvals and all other correspondence with the Ethics Committee.

It should be noted that Ethics Committee approval does not imply any resource commitment or administrative facilitation by any healthcare provider within whose facility the research is to be carried out. Where applicable, authority for this must be obtained separately from the appropriate manager within the organisation.

Yours sincerely

Pat Chainey
Northern X Ethics Committee Administrator

Cc: ADHB Research Office A+ 3528



a+ AUCKLAND
DISTRICT HEALTH BOARD
Te Toka Tumai

Auckland District Health Board
Greenlane Clinical Centre, Green Lane West
Auckland 3, New Zealand
Telephone: 09 638 9909
Website: www.adhb.govt.nz

Service:	ADHB Research Office Office: Level 8, Bldg. 13, GCC Postal: PB 92189 Auckland
Phone:	630-9943
Ext:	4085, 4077 and 3122
Fax:	630 - 9799 or 4999
Email:	GaylH@adhb.govt.nz
Website:	www.adhb.govt.nz/RDO

14 August 2006

Dr Michael Harrison
Anaesthetist
Department of Anaesthesia
Auckland City Hospital
Grafton
Auckland

This support letter is issued by the Maori Research Review Committee and ***does not*** *represent the Ethics approval or the ADHB management approval. Investigators are advised to seek other approvals separately.*

Tena koe Dr Harrison

RE: #3528 **Development And Validation Of An Interactive Anaesthesia Warning System For Monitoring The Adverse Effects That Might Arise During Anaesthesia Administration**

Thank you for the opportunity to review your study. We are happy to support the study.

Please send a copy of the final report to the Maori Research Review Committee (c/o Jenny Ma, Research Office, Level 8, Bldg 13, Greenlane Clinical Centre) at the conclusion of the study.

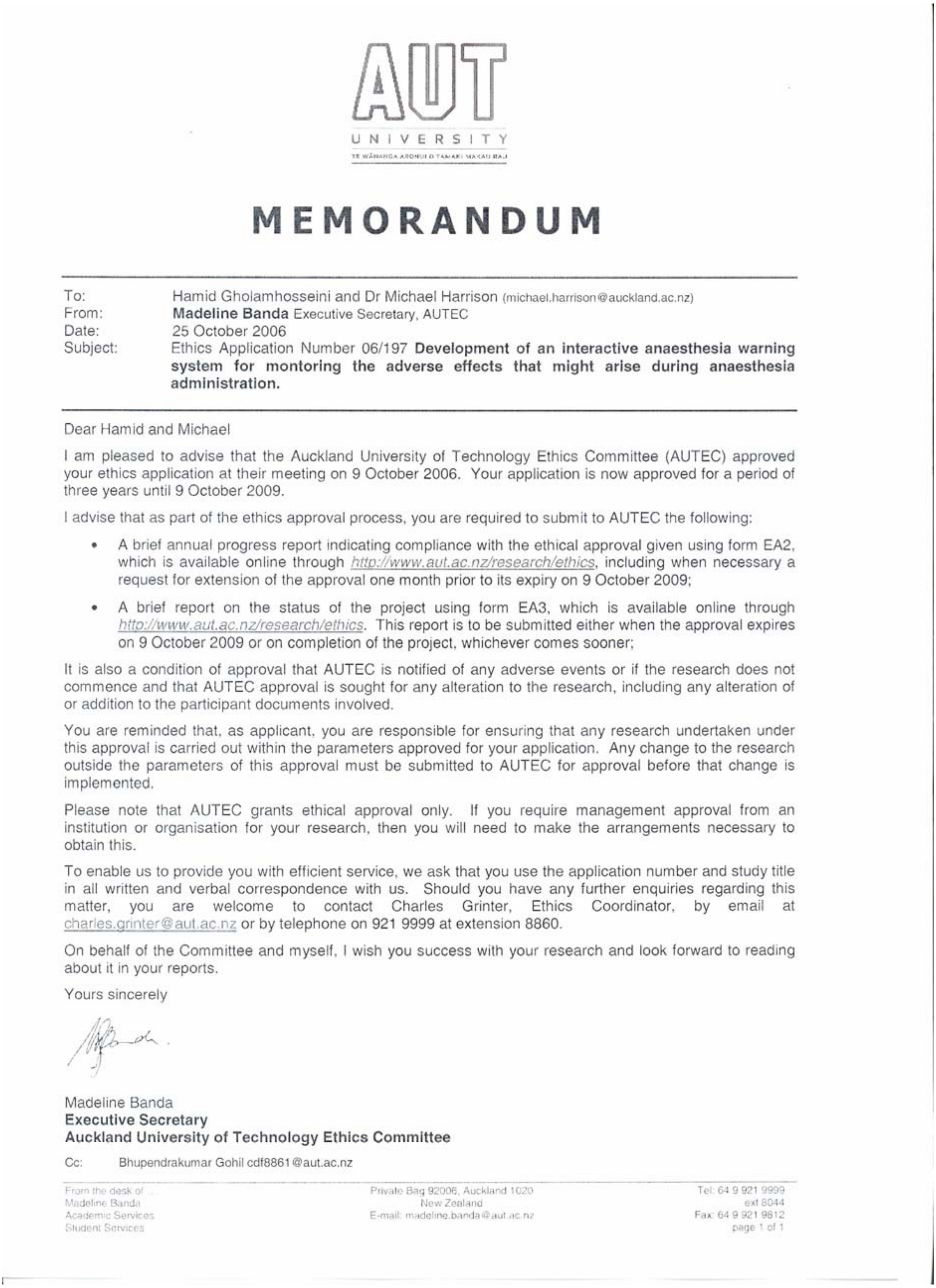
We wish you the very best in your research.

Noho ora mai,

On behalf of the ADHB Maori Research Review Committee
Gayl Humphrey
Manager, Research Office
Auckland DHB

c.c. Mata Forbes, MRRC

Appendix B2 - AUTEC X-Ethics Approval Letter.



AUT
UNIVERSITY
TE WĀNANGA ARONUI O TAMAKI MAKAU RAU

MEMORANDUM

To: Hamid Gholamhosseini and Dr Michael Harrison (michael.harrison@auckland.ac.nz)
From: **Madeline Banda** Executive Secretary, AUTEC
Date: 25 October 2006
Subject: Ethics Application Number 06/197 **Development of an interactive anaesthesia warning system for montoring the adverse effects that might arise during anaesthesia administration.**

Dear Hamid and Michael

I am pleased to advise that the Auckland University of Technology Ethics Committee (AUTEC) approved your ethics application at their meeting on 9 October 2006. Your application is now approved for a period of three years until 9 October 2009.

I advise that as part of the ethics approval process, you are required to submit to AUTEC the following:

- A brief annual progress report indicating compliance with the ethical approval given using form EA2, which is available online through http://www.aut.ac.nz/research/ethics, including when necessary a request for extension of the approval one month prior to its expiry on 9 October 2009;
- A brief report on the status of the project using form EA3, which is available online through http://www.aut.ac.nz/research/ethics. This report is to be submitted either when the approval expires on 9 October 2009 or on completion of the project, whichever comes sooner;

It is also a condition of approval that AUTEC is notified of any adverse events or if the research does not commence and that AUTEC approval is sought for any alteration to the research, including any alteration of or addition to the participant documents involved.

You are reminded that, as applicant, you are responsible for ensuring that any research undertaken under this approval is carried out within the parameters approved for your application. Any change to the research outside the parameters of this approval must be submitted to AUTEC for approval before that change is implemented.

Please note that AUTEC grants ethical approval only. If you require management approval from an institution or organisation for your research, then you will need to make the arrangements necessary to obtain this.

To enable us to provide you with efficient service, we ask that you use the application number and study title in all written and verbal correspondence with us. Should you have any further enquiries regarding this matter, you are welcome to contact Charles Grinter, Ethics Coordinator, by email at charles.grinter@aut.ac.nz or by telephone on 921 9999 at extension 8860.

On behalf of the Committee and myself, I wish you success with your research and look forward to reading about it in your reports.

Yours sincerely

Madeline Banda
Executive Secretary
Auckland University of Technology Ethics Committee

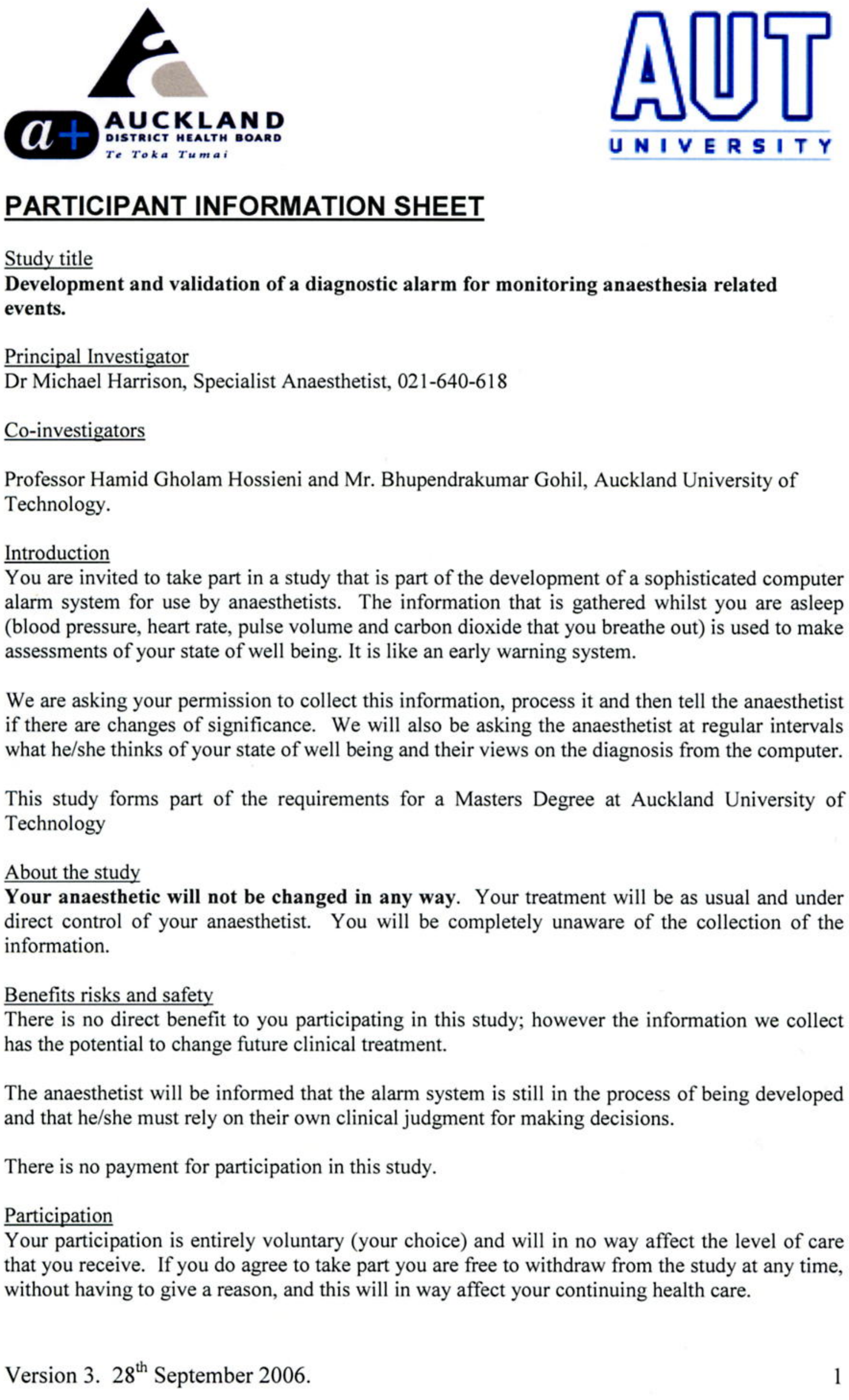
Cc: Bhupendrakumar Gohil cdf8861@aut.ac.nz

From the desk of ...
Madeline Banda
Academic Services
Student Services

Private Bag 92006, Auckland 1020
New Zealand
E-mail: madeline.banda@aut.ac.nz

Tel: 64 9 921 9999
ext 8044
Fax: 64 9 921 9812
page 1 of 1

Appendix B3 - Patient Information Sheet.



AUCKLAND
DISTRICT HEALTH BOARD
Te Toka Tumai

AUT
UNIVERSITY

PARTICIPANT INFORMATION SHEET

Study title
Development and validation of a diagnostic alarm for monitoring anaesthesia related events.

Principal Investigator
Dr Michael Harrison, Specialist Anaesthetist, 021-640-618

Co-investigators

Professor Hamid Gholam Hossieni and Mr. Bhupendrakumar Gohil, Auckland University of Technology.

Introduction
You are invited to take part in a study that is part of the development of a sophisticated computer alarm system for use by anaesthetists. The information that is gathered whilst you are asleep (blood pressure, heart rate, pulse volume and carbon dioxide that you breathe out) is used to make assessments of your state of well being. It is like an early warning system.

We are asking your permission to collect this information, process it and then tell the anaesthetist if there are changes of significance. We will also be asking the anaesthetist at regular intervals what he/she thinks of your state of well being and their views on the diagnosis from the computer.

This study forms part of the requirements for a Masters Degree at Auckland University of Technology

About the study
Your anaesthetic will not be changed in any way. Your treatment will be as usual and under direct control of your anaesthetist. You will be completely unaware of the collection of the information.

Benefits risks and safety
There is no direct benefit to you participating in this study; however the information we collect has the potential to change future clinical treatment.

The anaesthetist will be informed that the alarm system is still in the process of being developed and that he/she must rely on their own clinical judgment for making decisions.

There is no payment for participation in this study.

Participation
Your participation is entirely voluntary (your choice) and will in no way affect the level of care that you receive. If you do agree to take part you are free to withdraw from the study at any time, without having to give a reason, and this will in way affect your continuing health care.

Version 3. 28th September 2006. 1

Participation in this study will be stopped should any harmful effects appear or if the doctor feels it is not in your best interests to continue.

Frequently asked questions

What will happen at the end of the study?

The results will be collected, anonymised, analysed and published in an anaesthetic journal.

Where can I get more information about the study?

Feel free to contact any of the researchers for further information about the study.

If I need an interpreter, can one be provided?

As informed consent is a requirement this will be provided as necessary.

You may have a friend, family or whanau support to help you understand the risks and/or benefits of this study and any other explanation you may require.

If you have any queries or concerns regarding your rights as a participant in this study you may wish to contact a Health and Disability Advocate, telephone

Northland to Franklin	0800 555 050
Mid and lower North Island	0800 42 36 38 (4 ADNET)

For Auckland District Health Board Maori health support, please contact Mata Forbes, RGON; Co-ordinator/Advisor , Maori Health Services, Auckland Hospital, Grafton. Phone (09)3074949 ext. 7292 or (021)348432.

Confidentiality

No material which could personally identify you will be used in any reports on this study. Records are stored in a locked cabinet and within a secure computer server.

Results

The results from this research will be published in a scientific journal. No individual will be identified. There will be no reference to you or your name in any publications. The delay between starting the study and publishing results may be over two years. However, should you wish to receive a copy of the results please indicate this on the consent form.

Compensation

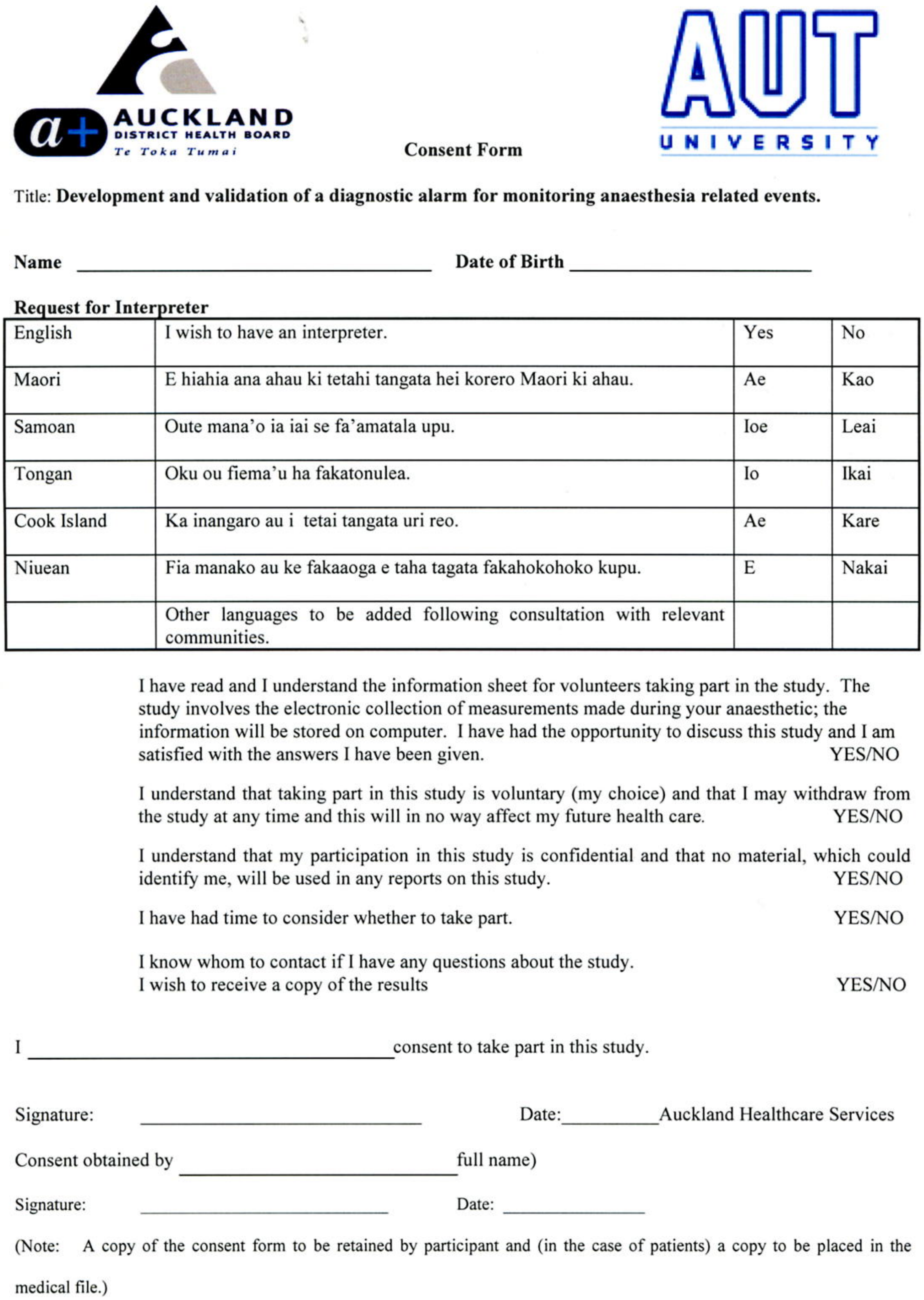
In the very unlikely event of a physical injury as a result of your participation in this study, you may be covered by ACC under the Injury Prevention, Rehabilitation and Compensation Act. ACC cover is not automatic and your case will need to be assessed by ACC according to the provisions of the 2002 Injury Prevention Rehabilitation and Compensation Act. If your claim is accepted by ACC, you still might not get any compensation. This depends on a number of factors such as whether you are an earner or non-earner. ACC usually provides only partial reimbursement of costs and expenses and there may be no lump sum compensation payable. There is no cover for mental injury unless it is a result of physical injury. If you have ACC cover, generally this will affect your right to sue the investigators.

If you have any questions about ACC, contact your nearest ACC office or the investigator.

This study has received ethical approval from the Northern X Regional Ethics Committee.

Please feel free to contact the researcher if you have any questions about this study.
021- 640- 618

Appendix B4 - Consent Forms.



AUCKLAND DISTRICT HEALTH BOARD
Te Toka Tumai

Consent Form

AUT UNIVERSITY

Title: **Development and validation of a diagnostic alarm for monitoring anaesthesia related events.**

Name ______________________________ **Date of Birth** ____________________

Request for Interpreter

English	I wish to have an interpreter.	Yes	No
Maori	E hiahia ana ahau ki tetahi tangata hei korero Maori ki ahau.	Ae	Kao
Samoan	Oute mana'o ia iai se fa'amatala upu.	Ioe	Leai
Tongan	Oku ou fiema'u ha fakatonulea.	Io	Ikai
Cook Island	Ka inangaro au i tetai tangata uri reo.	Ae	Kare
Niuean	Fia manako au ke fakaaoga e taha tagata fakahokohoko kupu.	E	Nakai
	Other languages to be added following consultation with relevant communities.		

I have read and I understand the information sheet for volunteers taking part in the study. The study involves the electronic collection of measurements made during your anaesthetic; the information will be stored on computer. I have had the opportunity to discuss this study and I am satisfied with the answers I have been given. YES/NO

I understand that taking part in this study is voluntary (my choice) and that I may withdraw from the study at any time and this will in no way affect my future health care. YES/NO

I understand that my participation in this study is confidential and that no material, which could identify me, will be used in any reports on this study. YES/NO

I have had time to consider whether to take part. YES/NO

I know whom to contact if I have any questions about the study.
I wish to receive a copy of the results YES/NO

I ______________________________ consent to take part in this study.

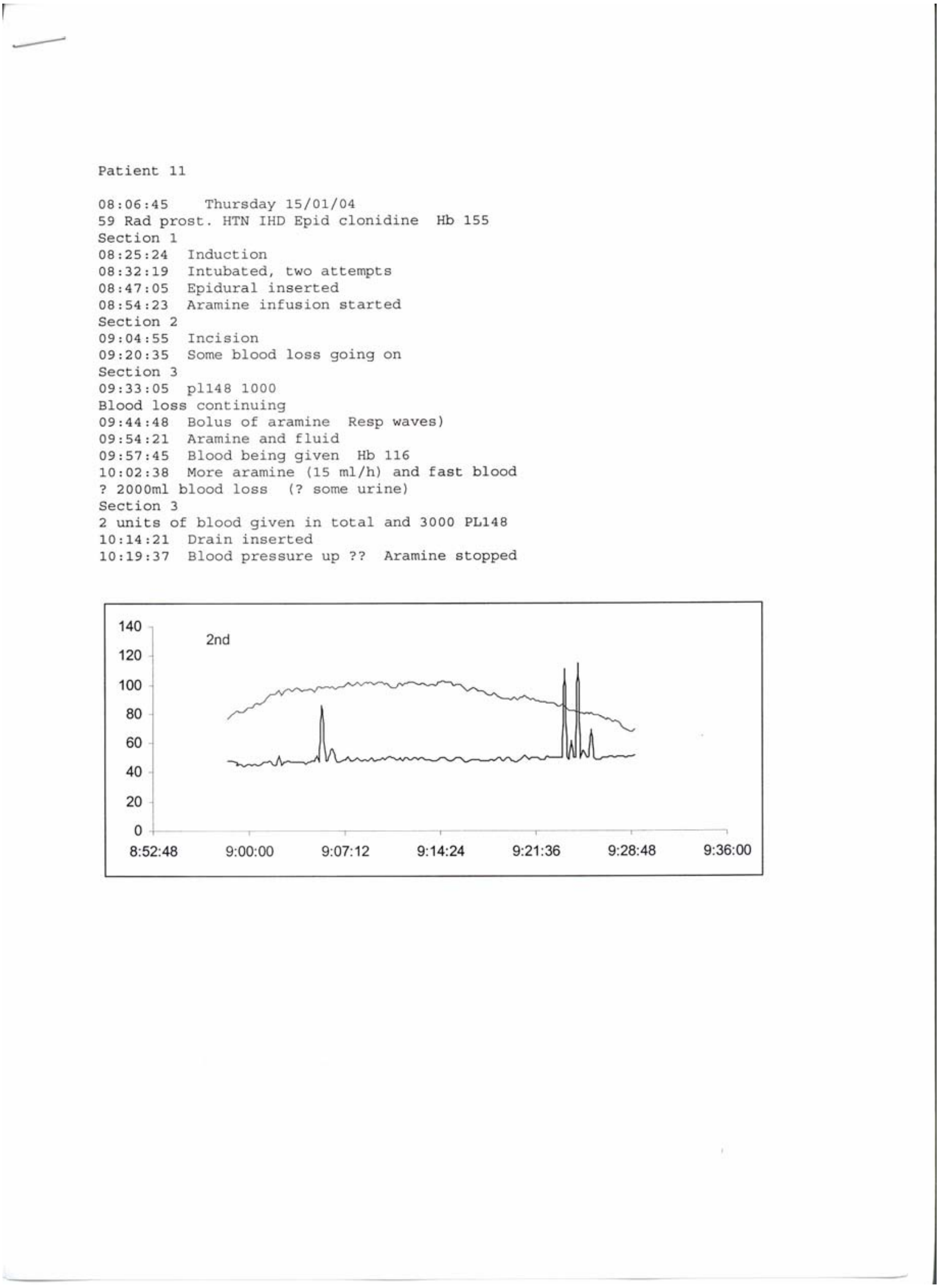
Signature: ______________________ Date:__________Auckland Healthcare Services

Consent obtained by ______________________ full name)

Signature: ______________________ Date: ____________

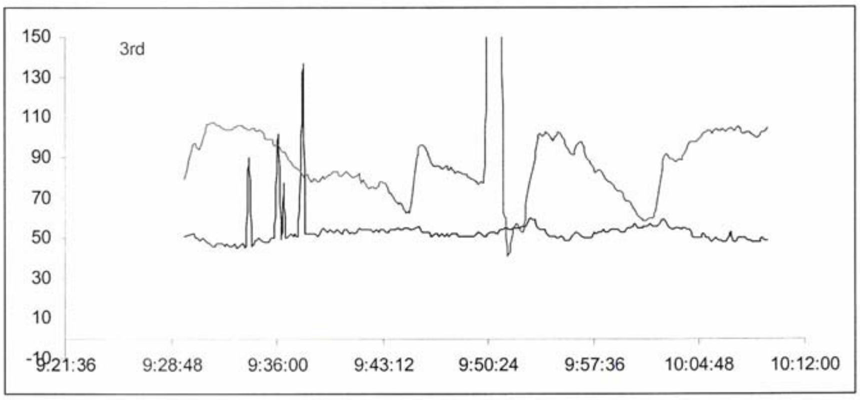
(Note: A copy of the consent form to be retained by participant and (in the case of patients) a copy to be placed in the medical file.)

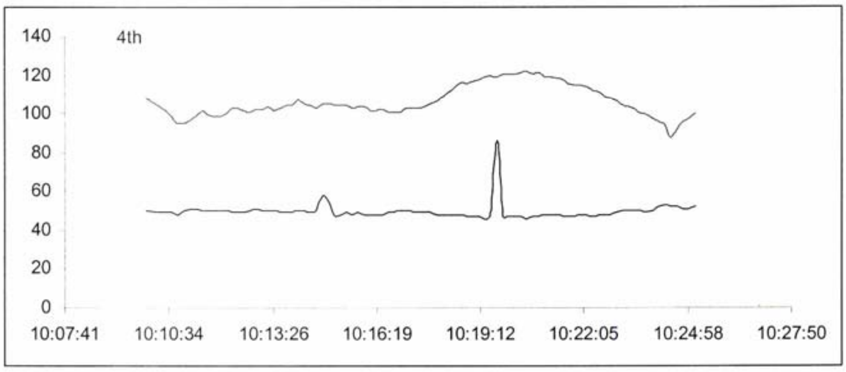
Appendix C1 - Patient Log.



```
Patient 11

08:06:45    Thursday 15/01/04
59 Rad prost. HTN IHD Epid clonidine  Hb 155
Section 1
08:25:24  Induction
08:32:19  Intubated, two attempts
08:47:05  Epidural inserted
08:54:23  Aramine infusion started
Section 2
09:04:55  Incision
09:20:35  Some blood loss going on
Section 3
09:33:05  pl148 1000
Blood loss continuing
09:44:48  Bolus of aramine  Resp waves)
09:54:21  Aramine and fluid
09:57:45  Blood being given  Hb 116
10:02:38  More aramine (15 ml/h) and fast blood
? 2000ml blood loss  (? some urine)
Section 3
2 units of blood given in total and 3000 PL148
10:14:21  Drain inserted
10:19:37  Blood pressure up ??  Aramine stopped
```


3rd
150
130
110
90
70
50
30
10
-10
9:21:36
9:28:48
9:36:00
9:43:12
9:50:24
9:57:36
10:04:48
10:12:00


4th
140
120
100
80
60
40
20
0
10:07:41
10:10:34
10:13:26
10:16:19
10:19:12
10:22:05
10:24:58
10:27:50

Appendix C2 - Offline Analysis Results.

Patient No.	Timeslot	SAAM Diag.	Expert Diag.	TruePOS	TrueNEG	FalsePOS	FalseNEG
				Agreement	Agreement	Disagreement	Disagreement
				SAAM +ve	SAAM -ve	SAAM +ve	SAAM-ve
				Expert +ve	Expert -ve	Expert -ve	Expert +ve
Patient 1	7:52	V	P				
	8:00	N	N		1		
	8:36	N	N		1		
	8:45	N	P				1
	9:19	V	N			1	
	9:30	N	N		1		
	9:50	V	N			1	
	10:00	P	N			1	
	10:21	P	P	1			
	10:30	N	N		1	1	
	10:52	V	N			1	
	11:00	V	N				
Patient 3	9:47	N	P				1
	10:00	P	P	1			
	10:34	P	P	1			
	10:45	V	N			1	
	11:37	N	N		1		
	11:45	N	P				1
	12:00	P	P	1			
	12:09	V	V	1			
	12:30	V	V	1			
	13:09	P	P	1			
	13:30	N	N		1		
Patient4	11:00	VP	P	1			
	11:15	NN	P				1
	11:30	VV	V	1			
	11:45	NV	P	1			
	12:00	NN	N		1		
	12:15	VV	V	1			
	12:30	NN					
	12:45	VV	p	1			
	1:00	VN	P	1			
	1:15	VN	P	1			
	1:30	VN	P	1			
Patient 5	10:24	V	P	1			
	10:52	P	N			1	
	11:19	N	N		1		
	11:37	N	N		1		
	11:45	N	N		1		
	12:12	N	N		1		

	Timeslot	SAAM	Expert	TruePOS	TrueNEG	FalsePOS	FalseNEG
	12:30	N	N		1		
	12:45	V	N			1	
	13:00	V	N			1	
	13:15	V	N			1	
	13:23	N	N		1		
	13:45	V	P	1			
Patient 6	10:45	P	N			1	
	1100	N	N		1		
	1115	VP	N			1	
	11:32	NV	P	1			
	1145	PV	P	1			
	1204	NN	N		1		
	1215	NV	P	1			
	1230	NN	N		1		
	1251	NN	P	1			
	1300	NV	V				
Patient 7	9:23	NN	N		1		
	9:30	VV	N			1	
	10:29	NV	P	1			
	10:45	PV	P	1			
	11:05	NN	N		1		
	11:38	NN	N		1		
	12:00	NN	N		1		
Patient 8							
	10:19	V	N			1	
	10:30	V	P	1			
	10:45	N	N		1		
	11:00	V	P	1			
	11:13	V	N			1	
	11:30	V	V	1			
	11:45	V	P	1			
	11:56	P	P	1			
	12:15	N	N		1		
	12:27	N	N		1		1
	12:45	N	P				1
	13:00	N	N		1		
Patient 9	11:25	VN	N			1	
	11:45	NV	P	1			
	12:37	NN	N		1		
	12:45	NN	N		1		
	13:00	NN	N		1		
	13:15	NP	P	1			
	13:30	NN	N		1		
	13:45	NN	N		1		
	14:09	NV	N			1	
	14:15	NN	N		1		

	Timeslot	SAAM	Expert	TruePOS	TrueNEG	FalsePOS	FalseNEG
Patient 10	9:00	V	P	1			
	9:15	N	N		1		
	9:47	P	P	1			
	10:00	N	V				1
	10:15	N	N		1		
	11:05	V	P	1			
	11:15	N	P				1
	11:30	N	V				1
	11:45	V	V	1			
	12:00	V	V	1			
	12:15	N	N		1		
	12:23	V	V	1			
	12:45	V	P	1			
	13:00	N	N		1		
	13:08	V	P	1			
	13:30	V	P	1			
	13:45	N	N		1		
Patient 11	8:58	PN	P	1			
	9:15	NN	N		1		
	9:29	VV	V	1			
	9:45	VV	V	1			
	10:00	VV	V	1			
	10:09	VN	P	1			
Patient 13	9:15	NV	????				
	9:44	N	N		1		
	10:00	NV	P	1			
	10:18	NN	N	1			
	10:30	NN	N	1			
	10:49	NN	N	1			
	11:05	NV	N			1	
Patient 15	8:30	NV	P	1			
	8:45	NV	P	1			
	9:00	VN	P	1			
	9:26	VV	V	1			
	9:45	VV	V	1			
	10:00	NP	P	1			
	11:15	NV	V	1			
	11:30	NN	P				1
	11:45	VN	N			1	
Patient 16	9:00	NN	N		1		
	9:17	NN	N		1		
	9:30	VN	P	1			
	9:45	NN	N		1		
	9:58	VV	P	1			
	10:12	PV	V	1			
	10:30	NV	P	1			
	10:50	NV	V	1			
	11:00	NV	V	1			

	Timeslot	SAAM	Expert	TruePOS	TrueNEG	FalsePOS	FalseNEG
	11:15	PN	V	1	1		
	11:35	NN	N		1		
	11:45	NN	P				
Patient 17	8:45	NN	N		1		
	9:45	PN	N			1	
	9:58	NN	N		1		
	10:19	NN	N		1		
	10:30	VV	N	1			
	10:50	NN	N		1		
	11:00	NP	P	1			
	11:15	NN	N		1		
	11:28	VN	P	1			
	11:45	NN	N		1		
	12:01	NN	N		1		
	12:15	NN	N		1		
	12:51	PP	P	1			
	13:15	NN	N		1		
	13:25	NN	N		1		
	13:45	VV	P	1			
	14:00	NN	P				1
	14:08	VN	N			1	
	14:30	NN	N		1		
	14:43	NN	N		1		
	15:00	NN	N		1		
Adden 2	9:17	VV					
	9:30	VV					
	9:45	NV					
	10:00	VV					
	10:15	NN					
	10:30	NN	N		1		
	11:27	VV	P	1			
	11:45	VN	P	1			
	12:00	VN	P	1			
	12:15	VV	P	1			
	12:29	PV	P	1			
	12:45	VV	P	1			
	13:00	NN	N		1		
Adden 3	9:17	VV	P	1			
	9:30	VV	P	1			
	9:45	VV	N			1	
	10:02	VV	V	1			
	10:15	VV	PV	1			
	10:30	VV	V	1			
	10:45	VV	V	1			
	11:00	VV	V	1			
	11:15	VV	V	1			
	11:30	VV	V	1			
	11:45	VV	V	1			
	12:00	VV	V	1			

	Timeslot	SAAM	Expert	TruePOS	TrueNEG	FalsePOS	FalseNEG
	12:15	VV	P	1			
	12:30	VN	P	1			
Adden 4	14:24	PN	P	1			
	14:43	NN	N		1		
	15:00	VN	N			1	
	15:15	NN	N		1		
	15:27	NN	N		1		
	15:45	PN	P	1			
	16:00	PN	P	1			
	16:15	NN	N		1		
	16:34	VN	P	1			
	16:45	NV	N			1	
	17:00	VN	P	1			
	17:15	NV	P		1		
	17:30	NN	N				
Adden 5	9:59	NN	N		1		
	10:15	VN	P	1			
	10:30	NN	N		1		
	10:45	NN	N		1		
	10:59	VN	N			1	
	11:15	NV	N			1	
	11:30	VN	P	1			
	11:45	NN	N		1		
	12:01	NN	N		1		
	12:15	NN	N		1		
	12:30	NV	N			1	
	12:45	NN	N		1		
	13:00	NN	P				1
	13:15	NN	N		1		
	13:30	NN	N		1		
	13:45	NN	N		1		
Total				93	72	27	12

Appendix C3 - Real-Time Test Results

Patient No.	TimeSlot	SAAM Diag.	Expert Diag.	TruePOS	TrueNEG	FalsePOS	FalseNEG
				Agreement	Agreement	Disagreement	Disagreement
				SAAM +ve	SAAM -ve	SAAM +ve	SAAM-ve
				Expert +ve	Expert -ve	Expert -ve	Expert +ve
Patient 1	9:50	--	N				
	10:05	NN	N		1		
	10:20	NN	N		1		
	10:35	NV	N			1	
	10:50	NN	N		1		
	11:05	NN	N		1		
	11:20	NP	N			1	
	11:35	PP	P	1			
	11:50	NN	P				1
	12:05	NV	P	1			
	12:20	NV	P	1			
	12:35	VV	P	1			
	12:50	VV	V	1			
	1:05	OV	V	1			
	1:20	OV	N			1	
	1:35	NN	N		1		
	1:50	OV	V	1			
	2:05	OV	N			1	
Patient 2	9:00	NN	N		1		
	9:15	NN	N		1		
	9:30	NN	N		1		
	9:45	NV	N			1	
	10:00	NN	N		1		
	10:15	NV	N			1	
	10:30	NV	N			1	
	10:45	NN	N		1		
	11:00	NN	N		1		
	11:15	NN	N		1		
	11:30	NP	P	1			
	11:45	PP	P	1			
	12:00	NP	N			1	
	12:15	NN	N		1		
	12:30	NN	N		1		
Patient 4	9:00	--	--				
	9:15	--	N				
	9:30	NN	N		1		
	9:45	NV	N			1	
	10:00	NV	N			1	
	10:15	NN	N		1		
	10:30	NN	N		1		
	10:45	NN	N		1		
	11:00	NN	N		1		

	11:15	NN	N		1		
	Timeslot	SAAM	Expert	TruePOS	TrueNEG	FalsePOS	FalseNEG
Patient 5	12:00pm	--	--				
	12:15	--	--				
	12:30	NN	N		1		
	12:45	VV	N		1		
	1:00	NV	N			1	
	1:15	NN	N		1		
	1:30	NV	N			1	
	1:45	NN	N		1		
Patient 6	9:00	--	--				
	9:15	NN	N		1		
	9:30	NN	N		1		
	9:45	NN	N		1		
	10:00	NN	N		1		
	10:15	NN	N		1		
	10:30	NN	N		1		
	10:45	NN	N		1		
	11:00	VN	N			1	
	11:15	NN	N		1		
Total				9	32	13	1