A computer-based system for quantitatively monitoring respiratory depression induced by incremental levels of halothane anesthesia

by

Mark Munas Mehl

A Thesis Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE

Major: Biomedical Engineering

Signatures have been redacted for privacy

Signatures have been redacted for privacy

Iowa State University
Ames, Iowa

1985

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

The Problem with Anesthesia

Relative to other drugs, general anesthetics have been very difficult to administer safely--principally because of their low therapeutic indices. This means that an anesthetic dosage which is correct for one individual may be lethal to another. As a result, depth of anesthesia must be quantitatively monitored throughout surgery so that the level of anesthesia can be titrated to the individual's need and the surgical requirement. Therefore, the anesthesiologist must estimate the level of anesthesia accurately enough to maintain sufficient analgesia and muscle relaxation without endangering the patient's life.

Unfortunately, there exists no quantitative methodology to determine the ideal anesthetic level for a patient. Presently, this ideal level is surmised by judging qualitative signs. To judge anesthetic depth more precisely, a computer-based system could be developed which quantitatively determines the level of anesthesia from selected physiological parameters. This thesis proposes an apparatus to measure many of these physiological parameters in relation to incremental levels of halothane anesthesia. Future work may use such a system to determine the relationship--and validity--that these parameters have relative to varying degrees of halothane anesthesia. These results can then serve as a basis for the development of an automated anesthetic monitoring system.

Preliminary Considerations Based on Earlier Work

General anesthetics are primarily targeted at the central nervous system (CNS); therefore, in order to precisely characterize anesthetic depth, one must measure physiological parameters that directly reflect CNS activity. Mori and his associates [1] performed such physiological measurements on the midbrain in the reticular formation--which is partly responsible for consciousness and autonomic control--by characterizing its unit activity while subjected to different levels of these anesthetics. Not only did their work show that reticular unit activity directly reflects depth of anesthesia for several general anesthetics, but it also showed that this unit activity is quantized in distinct steps which exactly correspond to the different stages and planes of anesthesia.

Although reticular formation unit activity may accurately quantitate depth of anesthesia for many volatile anesthetics, it is not suitable for the clinical environment since it requires special electrode implantation. From this point on, this paper will focus on non-invasive approaches for quantizing depth of anesthesia.

One such method studied by many authors [2-9] is electroencephalography (EEG). Examining this approach uncovers two major problems. First, quantitative analysis of EEGs on a continuous real-time basis--which would be necessary in clinical practice--requires fairly fast and sophisticated equipment. Moreover, EEG data processed via Fourier transform produces results which are not reproducible with time [9]. In other words, a subject maintained at 1% halothane anesthesia will have a dif-

ferent EEG power spectrum at the onset of anesthesia as opposed to two hours later. Differences become even more pronounced with prolonged periods of anesthesia. This is probably because the CNS can eventually compensate for some of the effects that halothane brings upon the brain.

A better method for monitoring CNS activity would involve monitoring specific areas of the brain. To perform this non-invasively (without penetrating the scalp), however, would require signal averaging of evoked-potentials. Work concerning both auditory and visually evoked-potentials was perused. Work dealing with evoked-potentials in the somatosensory system was not seriously investigated.

With regard to visually evoked-potentials, the electroretinogram (ERG) was one of the earliest techniques to be explored in relationship to halothane anesthesia. Raitta et al. [10] observed a very strong change in the amplitude of the ERG with increasing dosages of halothane; however, Gerritsen [11] elegantly demonstrated that this was not a direct effect produced by halothane. By controlling expired pCO_2, Gerritsen showed that increasing halothane levels gave rise to higher expired pCO_2, and it is the higher pCO_2 which led to the amplitude changes in the ERG.

Both above authors also investigated visual evoked-potentials (VEP) with halothane anesthesia, but arrived at conflicting conclusions. Raitta et al.'s work [10] in man indicated that halothane only slightly depressed VEPs; whereas, Gerritsen's work [11] in rabbits reported marked changes in the VEP waveform, but he also pointed out that the observed changes were not characterizable between his experiments in spite of taking great lengths to control experimental conditions. Curiously, more

recent work by Uhl et al. [12]--apparently unaware of any earlier studies--showed that visual evoked responses (VER) produced fairly good results by increasing the latencies of VERs with higher levels of halothane anesthesia. The vast discrepancies observed in all three studies can best be explained by the positioning differences of the recording electrodes. Uhl's impressive results were obtained by placing the recording electrodes between the Oz and Cz scalp positions while using the ear lobe as ground.

Based on the propitious results presented by Uhl, visual evoked responses deserve further serious consideration; nevertheless, the VER approach still has several shortcomings. Johnson et al. [13] pointed out that long-term anesthesia results in degenerative changes in visual receptors and pigment epithelium as shown by electron microscopy. Also, positional changes in the eye relative to the light source--during the experiment--may influence the VER. In conclusion, more control on future investigations is necessary to improve reproducibility of the VER approach. Also, since recording electrode placement is extremely critical for good results, special effort should be taken to evaluate several electrode combinations and their relative performances.

Click-evoked potentials (CEP or CER) were used to study the effects of halothane anesthesia on the auditory system by several researchers [14-16]. Sasa et al. [14] characterized CERs in the inferior colliculus and auditory cortex of the cat. They also characterized the unitary discharges in the inferior colliculus following each CER. They observed that halothane markedly suppressed unitary discharges primarily within

20-msec after click stimulation in contrast to other volatile anesthetics (ether, chloroform, nitrous oxide) where this depression was most pronounced 20 to 40-msec after stimulation.

Sasa et al. also noted that the amplitudes of the CERs in both the inferior colliculus and auditory cortex were depressed. This is in agreement with Celesia and Puletti's work [15] with man using a halothane/nitrous-oxide mixture. Not only did they see a depression in the CER amplitude for the auditory cortex, but they also found a significant increase in the latencies of the CER components. They concluded that there are several pathways which are evoked by click stimulation and that all these pathways meet at the auditory cortex, in turn, creating the polyphasic waveform with individual component latencies which vary independently of one another.

Kitahata [16] studied click-evoked responses in the auditory cortex of the cat. He too noted a depression in CER with halothane anesthesia; however, he pointed out that this depression was not a direct effect of halothane when he cited Beecher et al. [17]. Beecher et al. observed that high concentrations of halothane cause vascular hypotension which in turn caused the amplitude depression in the CER seen by earlier authors. As a result, Kitahata avoided using response amplitudes alone, but rather characterized the latencies of the responses which--in agreement with Celesia and Puletti--increased with halothane anesthesia.

Kitahata's most innovative contribution, however, came from the use of "recovery cycles" to characterize halothane anesthesia. Kitahata recognized that halothane affects the amount of habituation seen within the

brain. As a result, he stimulated with twin click pulses and measured the relative amplitude changes seen between components of the first and second response. He referred to these measures of relative change as the CER "recovery cycle". His recovery cycle technique established the best correlation with depth of anesthesia that has ever been reported in the literature. Certainly this approach deserves more serious investigation.

All the evoked-potential studies discussed above used invasive procedures to collect their results. More clinical, non-invasive procedures could be developed using a real-time moving average to average the response waveforms over time while controlling the level of halothane anesthesia. Unfortunately, the necessary high speed equipment for accomplishing this was not available; therefore, respiratory parameters (which respond slower) were studied instead. Before dismissing the subject of evoked-potentials, several points are worth re-emphasizing for future investigations in this area:

1. It is the unitary activity in the reticular formation which provides the absolute measure of brain activity and depression as induced by halothane anesthesia.
2. It is the rate of rise (first derivative) of the post synaptic potentials (not their amplitudes) that characterize the amount of unit activity entering a cortical synapse. As a result, all post synaptic responses should be characterized by their first derivative in order to determine how many units are firing in synchrony to produce them.

3. Habituation within the cortex is markedly affected by anesthesia. (Perhaps this can be explained from the role the hippocampus plays in mediating sensory activity at the cortices. The hippocampus is well-known for its habituating effects.) In light of this, all future evoked-potential studies dealing with anesthesia should stimulate with twin pulses, or a train of pulses, so that the recovery cycle or similar measure of habituation can be evaluated.
4. Exact recording electrode placement is extremely critical. It is essential to study several electrode placements and their relative performances in order to establish the best combination.

With the above points in mind, evoked-potentials are likely to play a very important role in monitoring the level of anesthesia for future clinical practice.

Respiratory Depression as a Measure of Anesthetic Depth

Since the ultimate goal in characterizing depth of anesthesia involves monitoring brain stem reticular activity, and since there is not a direct, non-invasive means of accomplishing this, an alternative approach must be sought. This involves indirectly characterizing brain stem activity via monitoring respiratory control. In other words, it is proposed that the level of anesthesia can be estimated by directly characterizing respiratory depression over incremental levels of halothane anesthesia in order to indirectly characterize the effects of halothane on the brain stem which regulates respiration.

This scheme is flawed, however, since respiratory depression is not solely caused by depression of the brain stem. For example, strong evidence [18] suggests that halothane affects cerebral vascular regulation of brain pO_2 by increasing its oxygen "set point", thereby causing vasodilation and increasing the cerebral blood flow (CBF), so that the brain encounters blood-gas concentrations which are different from those in the body. Moreover, the change in CBF itself is suspected to have some affect on chemoreceptor firing [19] further complicating the picture. Nevertheless, although respiratory depression may not directly correlate with brain stem depression induced by halothane, a very close correlation still exists as supported by the following discussion.

Respiratory control is directed by both pO_2 and pCO_2 concentrations in the blood. The effects of hypoxia on respiration under several levels of halothane anesthesia were studied by two groups with similar results [19, 20]. Weiskopf et al. [19] observed that 1.1% halothane severely depressed respiratory drive, as measured by minute volume, to different hypoxic concentrations of oxygen. In fact, it was not until arterial pO_2 reached 60-torr that minute volume began to rise significantly. They indicated that pO_2 drive was essentially abolished in the dog at 1.7% halothane which is in contrast to Knill and Gelb's work [20] in man who reported total loss of pO_2 respiratory drive at 1.1% halothane, if not lower. Knill suggested these discrepancies could be caused by species differences--between dog and man--relative to the amount of influence chemoreceptors have on respiratory control and the amount of pharmacological depression halothane inflicts on the chemoreceptor itself.

Since light halothane anesthesia abolishes pO_2 drive altogether, monitoring respiratory depression in response to pO_2 and halothane shows little promise. Moreover, if hypoxia causes only slight O_2 respiratory drive under light anesthesia, then hyperoxia--typically used during general anesthesia--will eliminate O_2 drive altogether. As a result, one must turn to CO_2 concentration and its influence, or lack of influence, on respiration in response to different levels of halothane anesthesia.

As expected, both of the above studies [19, 20] demonstrated that halothane affected respiratory drive relative to arterial pCO_2. Although CO_2 respiratory drive is somewhat depressed, it still remains strong enough to reliably characterize as well as to maintain sufficient spontaneous ventilation in the dog--during normal anesthesia. Further work supports end-tidal pCO_2 as the predominant parameter influencing respiratory behavior with halothane. As a final note, since arterial pCO_2 has been shown to be only 1.7-torr higher than end-tidal pCO_2 in normal man [23], both methods of measure can be considered to produce similar results.

Previous work [19-22] has found minute volume an excellent measure of respiratory activity with varying levels of halothane anesthesia. Considering the importance of CO_2 in respiratory regulation, minute pCO_2 volume may provide an even more important correlation with depth of anesthesia and respiratory depression. Minute volume alone, however, may not necessarily be an accurate measure of respiratory center depression with halothane anesthesia. Derenne et al. [24] presented clear evidence indicating that depression of minute volume with methoxyflurane and increas-

ing pCO_2 is not so much due to neural depression as it is to an increase in respiratory elastance; that is, the respiratory system becomes "stiff" with anesthesia. Derenne et al. reasoned that this increase in stiffness is not produced by segmental reflexes since spinal epidural anesthesia produces no change in effective elastance. Rather, they believe the stiffness is caused by an increase in the elastance of the lung itself. Whether halothane increases respiratory elastance thereby decreasing minute volume is not completely established. Nevertheless, some caution should be taken when characterizing respiratory center activity solely through minute volume.

Several authors [20, 24] observed that tidal volume and breaths per minute (BPM) also varied with both halothane and methoxyflurane anesthesia in man. More importantly, increasing anesthesia causes tidal volume to decrease and BPM to increase such that the product of the two (i.e., minute volume) would be only slightly depressed. This project will further study the relationship of these three parameters.

Derenne et al. [24] suggested that the degree of respiratory center activity is related to the initial force of contraction (actually the initial inspiratory pressure) of the diaphragm. In light of this, they noted that the first 0.1 seconds of inspiratory pressure (during occlusion) and the initial inspiratory flow rate (during normal breathing) most accurately characterize respiratory activity with methoxyflurane anesthesia in man. To further examine this in the dog, this study will measure peak inspiratory flow--which generally occurs at the onset of inspiration--and the net slope of the inspiratory response--which approx-

imates the mean inspiratory flow rate--from the onset to the termination of normal inspiration (figure 1).

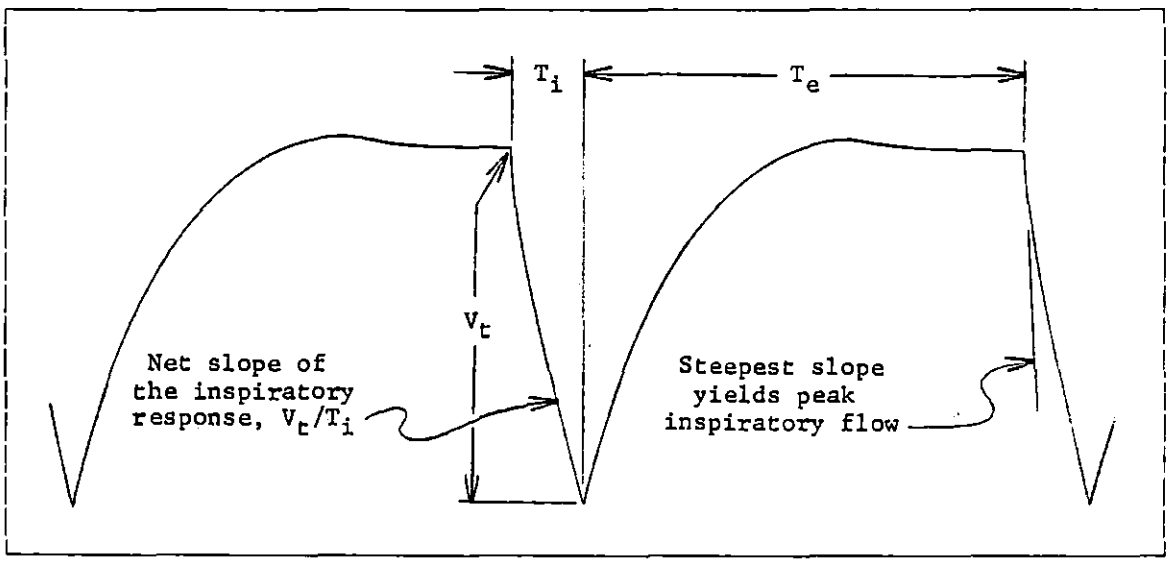


FIGURE 1. Position of inspiratory measurements to be taken

Other variables to be measured in this experiment are respiratory quotient (RQ) and end-tidal pCO_2/pO_2. Since CO_2 production and concentration generally rise and fall with O_2 production and concentration, one would expect their ratios--RQ and pCO_2/pO_2--to remain constant. If they, instead, vary with halothane anesthesia, they they may serve to characterize respiratory depression and depth of anesthesia.

MATERIAL AND METHODS

The overall apparatus for monitoring respiratory and cardiovascular parameters as well as controlling the depth of halothane anesthesia is shown in figure 2. Except for the 150-ml/minute of breath which is continually sampled by the infrared (IR) analyzers and replenished with 100% oxygen, the anesthetic loop is a rebreathing circuit.

The anesthetic loop consists of an 8-liter Godart-Statham spirometer (type 16003) which contains a blower to mix and circulate the air in the anesthetic circuit, a CO_2 absorber cartridge to remove expired CO_2, and a bell with a pulley connected to a 3-turn potentiometer (Bourns# 3511S-1-103, infinite resolution, conductive plastic element) which gauges bell displacement. The halothane vaporizer is composed of a 500-ml, three-neck round bottom flask. The two side necks are connected to the anesthetic circuit. Reaching 8-cm into the center neck is a 2-foot, 18-gauge, stainless steel tube which delivers liquid halothane. At the opposite end of this tube is a Harvard (series 940) syringe-type infusion pump. A Digital Equipment Corporation (DEC) PDP-8 minicomputer steps this infusion pump thereby causing a 20-ml glass syringe to inject a measured amount of liquid halothane into the bottom of the flask. Copper metal cuttings at the bottom of the flask, gently warmed by a heating mantle, then vaporize the liquid. The halothane vapor is then evenly distributed throughout the anesthetic loop by the spirometer blower.

Monitoring of respiratory parameters begins with a long Silastic microsampling tube which enters the anesthetic loop at the inhaler Y that

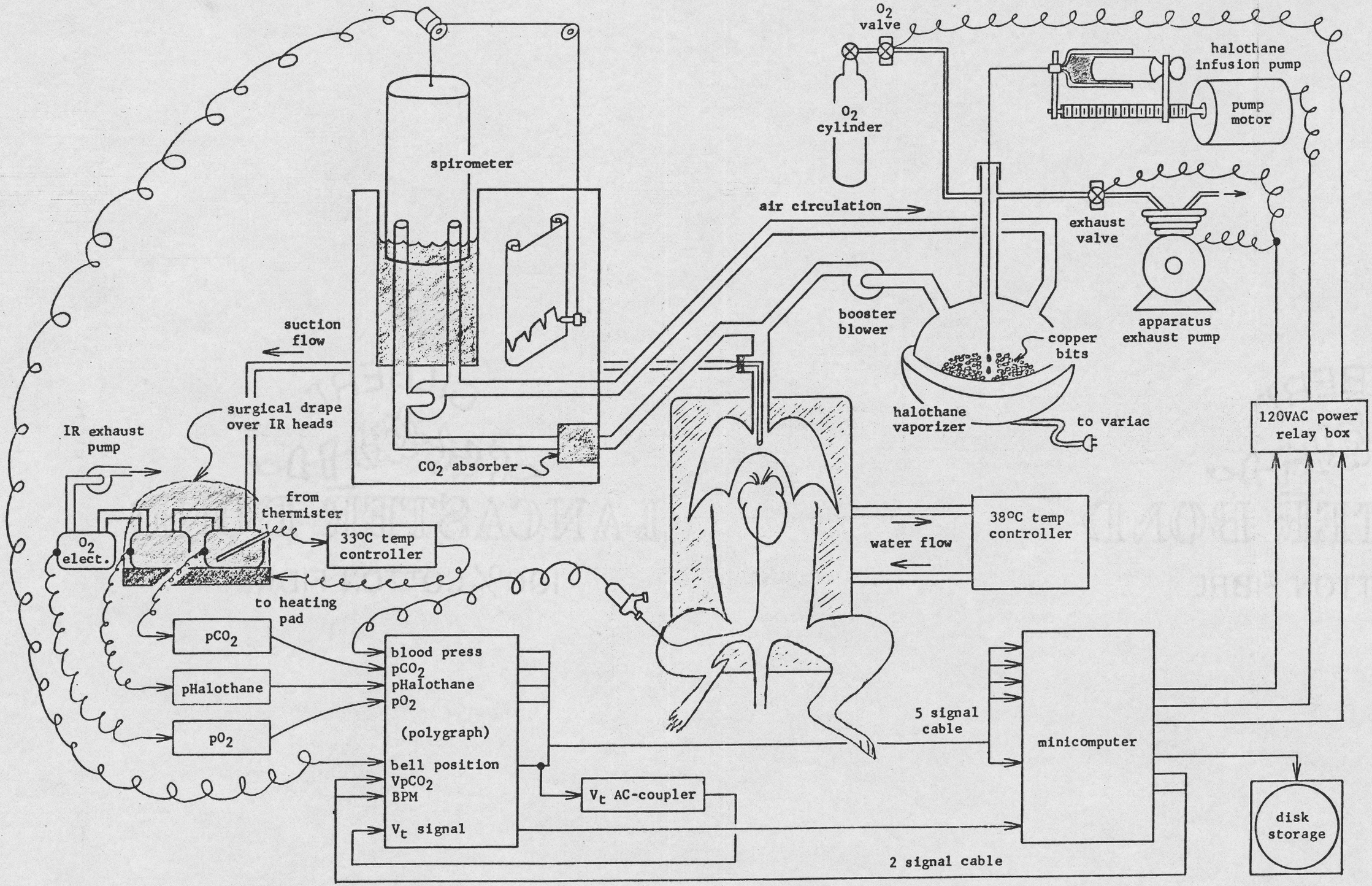


FIGURE 2. Layout of the apparatus

connects the dog to the loop. This sampling tube runs through the full length--and 3-cm beyond--the endotracheal tube and continuously samples end-tidal breath at 150-ml/minute. The other end of this tube--at the inhaler Y--is attached to the 3-foot "heated sample inlet" (Beckman Instruments# 149573) which is heated to prevent condensation of the moist expired breath. This heated inlet supplies three gas analyzer heads connected in series with Beckman tubing (Beckman# 673868) cut into 10-cm lengths. Series, as opposed to parallel, connections are used to reduce the required gas volume for continual analysis.

The first two IR heads are Beckman LB-2 units. They continually measure the partial pressure of CO_2 and halothane respectively. The partial pressure of O_2 is measured last with a Beckman OM-11 analyzer using a gold/silver KCl electrode. The output of all three analyzers goes to the PDP-8 computer via a Beckman R411 polygraph which appropriately amplifies and offsets the signals. An exception is made for the pO_2 signal, which is first attenuated by a resistor pad preceding connection to the 10-bit analog to digital (A/D) converters in the computer.

Calibration of the gold/silver oxygen electrode is complicated by its additional sensitivity to both water and halothane. Westenskow and associates [25] reported that the reading of the Beckman OM-10 oxygen electrode fell as much as 19% upon long term exposure to saturated water vapor. Similarly, halothane also appears to have a slow term affect on all types of gold/silver oxygen electrodes [26-30]. Depending on the electrode construction and the halothane concentration involved, halothane has been found to increase oxygen readings from 10% to 40% over a

10 minute [26] to 3 hour [30] period. Dent and Netter [27] showed that this halothane effect is nearly proportional to its concentration in saline. This is in contrast to gaseous halothane where Bates et al. [26] reported this effect to dwindle at approximately 105-torr.

A partial solution to this halothane interference problem, however, lies with an observation made by Maekawa, Okuda, and McDowall [28]. In spite of the 40% drift (over 45-minutes) they initially observed in their electrodes, they reported that--after recalibration--their electrodes drifted by less than 10% over an hour. Apparently, after the oxygen electrode "adapts" to the halothane, it becomes much more stable.

In light of these findings, the oxygen electrode in this experiment was calibrated only after first being exposed to a 1% halothane/oxygen mixture, which was saturated with water, for an hour. This allowed the oxygen electrode time to adapt to both water and halothane prior to calibration, thereby minimizing possible drift later on. Once again, since Maekawa et al. indicated that--in spite of late calibration--halothane still shifted the oxygen reading up by 10%, the oxygen readings gathered in this experiment may not be absolutely accurate; nevertheless, they should still demonstrate relative changes and anesthetic effects within a given 40 minute run.

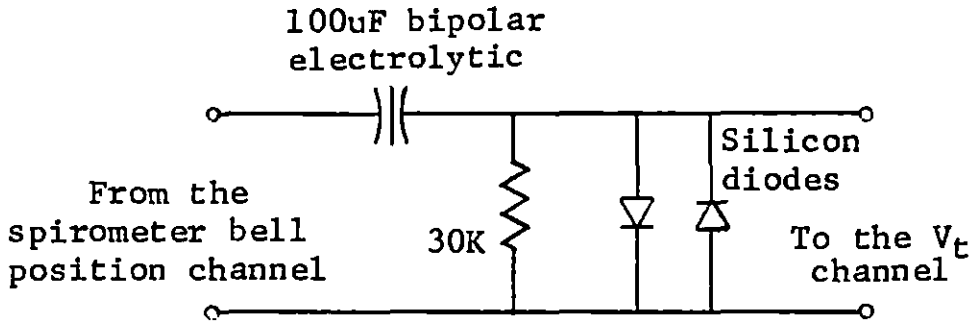
Unlike the pO_2 head which has internal temperature control, the pCO_2 and halothane IR heads do not. Hence, these two IR heads are placed on a heating pad; this assembly is then wrapped in a surgical drape. A YSI (model 63RC) on/off temperature controller maintains the inside of this assembly at 35°C so that the inside of the IR heads remains around 39°C.

This heating arrangement prevents the moisture in the expired breath from condensing inside the heads as it passes from one head to another. Both the heater and the three gas analyzers are allowed to equilibrate overnight prior to each experiment in order to prevent calibration drift during the experiment.

The bell displacement signal, transduced by the Bourns potentiometer on the spirometer, is preamplified by a channel on the Beckman polygraph prior to entering the computer which maintains the travel of the spirometer bell within the high/low limits. If the bell is too low, the computer switches oxygen into the anesthetic loop via a pneumatic valve; if it is too high, the computer enables an exhaust pump to remove the excess air.

The output of the bell displacement channel on the polygraph is also fed back into the input of a second polygraph channel through the AC coupling circuit shown on figure 3. Two back-to-back diodes serve to reduce circuit recovery time whenever a large bell displacement occurs; for instance, when oxygen is added or air is exhausted from the anesthetic loop. The output of this second, AC coupled channel (the tidal volume channel) is sent to the computer so that tidal volume, BPM, and expired/inspired instantaneous flow rates can be determined.

Blood pressure information is obtained from a cannula inserted in the right femoral artery and pushed up to the level of the heart. A Statham P23D pressure transducer is used to make the actual measurement. The output is processed by a Beckman (type 9853A) voltage/pulse/pressure coupler on the Beckman polygraph prior to sending the final pressure



This circuit removes the DC component from the bell displacement channel so that only the AC signal enters the highly sensitive V_t channel.

FIGURE 3. AC-coupling circuit for V_t channel

waveform to the computer. Signals from the blood pressure channel are then used to compute systolic, diastolic, pulse, and mean pressures as well as heart rate. Mean pressure is computed by the method outlined in Spooner [31].

Three signals--generated by the computer--are sent to the polygraph so that these computed results can be evaluated during the experiment. These parameters are integrated expired $VpCO_2$, end-tidal pCO_2/pO_2, and BPM. Short markers are superimposed on the BPM waveform by the computer--either up or down--to indicate either a crest or trough has been detected by the peak recognition software. This marking technique gives the operator a permanent record of the exact points the computer sampled crest or trough data on the input channels. This allows the operator to

make both an on-line and off-line performance evaluation of the peak recognition software. This is essential since there is no way of pretesting the peak recognition software on all abnormal breathing patterns that are encountered during an experiment.

All waveform data are recorded on a Honeywell 5600 instrumentation tape recorder. If any of the detection software is malfunctioning, the tape recording can be played back sometime after the software has been enhanced to determine if the enhancements can effectively analyze these unusual waveforms. In this way, the software can be enhanced, over time, to handle all possible waveforms encountered including the abnormal ones.

No premedications are given to the dogs before the experiment in order to prevent non-halothane side effects from influencing the results. Anesthesia is first induced with halothane, intubation performed, then the dog is oriented in dorsal recumbency for the remainder of the experiment. The control program is in the standby overlay, HALWT, during this time with halothane anesthesia maintained around 1.1%. After the blood pressure cannula is inserted and all preliminary checks and connections are made, the program is switched from the standby mode (HALWT overlay) to the initialization mode (HALINT overlay); this begins the first data collection run of the experiment.

Software Monitoring and Control

Software overview

The software for monitoring and controlling this experiment is compiled by DEC OS/8 BASIC (version 5B). To conserve core memory, the pro-

gram is split into five independent subprograms called overlays. Only one of these overlays may reside in memory at a time. When a residing overlay terminates, it issues a CHAIN command to instruct BASIC to load--from disk--and execute the superseding overlay.

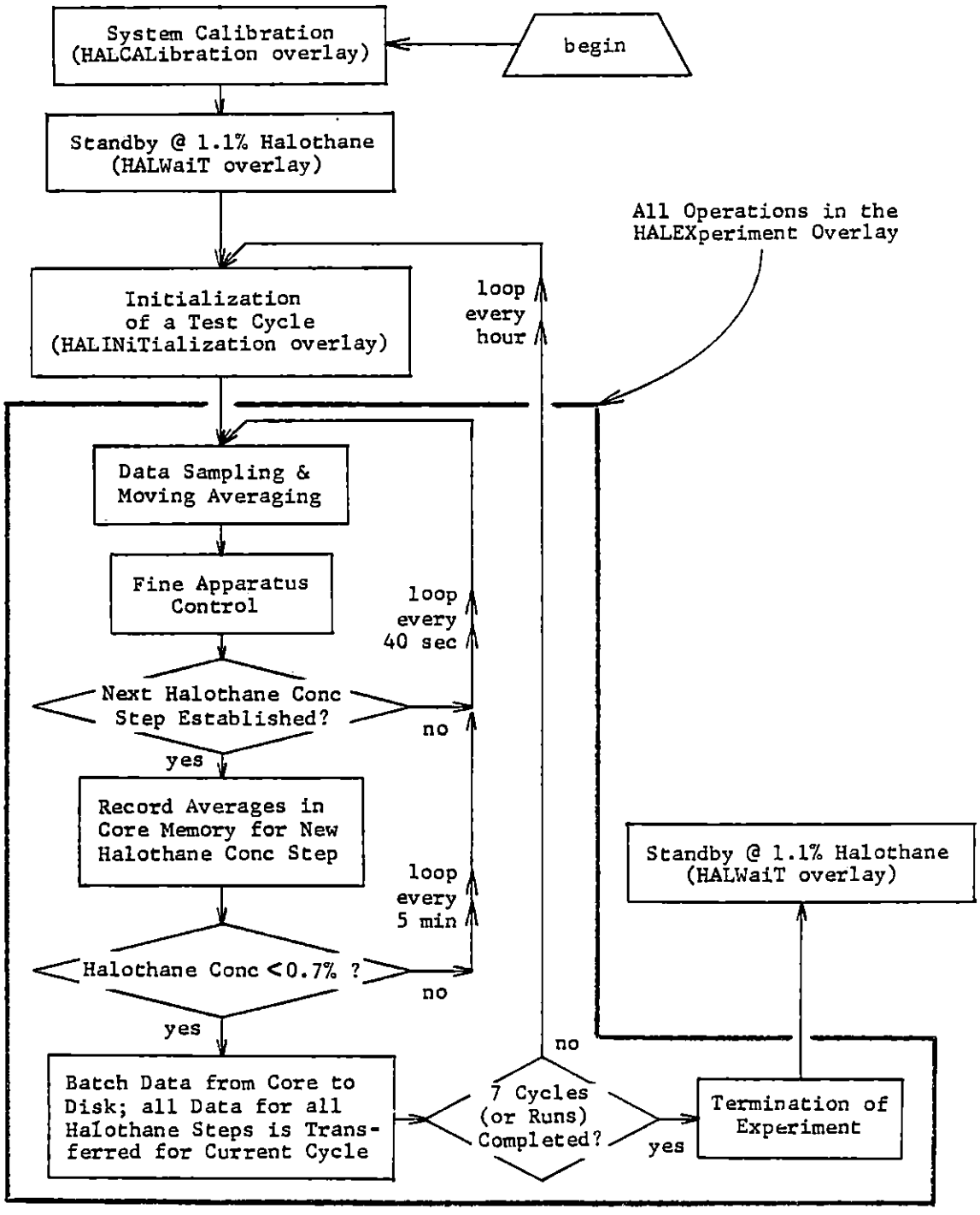
The first overlay executed, HALCAL, calibrates the monitoring and recording equipment with the A/D inputs and D/A outputs of the computer. Calibration of each channel is performed at three points: high, middle, and low. A linear regression technique is then used to determine the slope, m, and y-intercept, b, for each input channel. These slopes and intercepts are then used in the equation

$$(\text{real units}) = (\text{A/D units})(m) + (b)$$

to convert all digitally sampled data into real units for the program.

After calibrations are complete, HALCAL chains to HALWT which places the apparatus in standby mode so that the subject may be connected to the system. The primary purpose of the HALWT overlay is to maintain maximal halothane concentration at 1.1% thereby allowing both subject and apparatus to stabilize before beginning the experiment. HALWT is also responsible for maintaining the spirometer bell between its upper and lower limits via exhausting excess air or adding oxygen to the apparatus accordingly. Once the halothane concentration in the apparatus becomes stable, HALWT will have oxygen and halothane systematically added to the apparatus to replenish those gases as they are continuously removed by the Beckman gas analyzers.

Once everything has stabilized and the recording equipment is running, a sense switch is manually set on the computer to cause HALWT to



This flow diagram illustrates the interrelationship of each software module which all work in conjunction to drive the apparatus.

FIGURE 4. Overall software layout

chain to HALINT so that it can initialize the upcoming data collection run. HALINT controls the high/low limits of the spirometer bell much like HALWT. The halothane concentration, however, will be increased so that the inspired level climbs just over 2.2%. When this goal is achieved, HALINT will chain to the HALEX overlay.

HALEX (halothane experiment) will first wait until the expired halothane reaches 2.0% before sampling data. During the course of data collection, it simultaneously attempts to maintain the halothane expiratory/inspiratory gradient at 0.05% while allowing the halothane concentration to drop from 2.0% to 0.65% over an approximate half-hour period. There are two reasons for precisely controlling the halothane gradient at 0.05%. First, the gradient must remain small so that expired, end-tidal halothane accurately reflects arterial halothane--the most significant parameter. Second, by accurately maintaining a constant halothane gradient, the offset between expired halothane and arterial halothane will also remain constant [32] so that true arterial halothane can be approximated.

Throughout the experiment, all parameters are sampled during special collection intervals which span from 7 to 15 seconds. Moving averages are then used to smooth out results collected over four of these intervals. As the expired halothane decreases from 2.0% to 0.65% over the half-hour run, these moving averages for all parameters are cached in memory for every 0.05% incremental drop of expired halothane. After expired halothane reaches 0.65%, all cached data are transferred to disk for further off-line analysis, and the run is declared over. HALEX then

begins a new run by incrementing the run counter and chaining back to HALINT so that the halothane concentration can be re-established at its initial high level. Runs are continually performed in this way until seven acceptable runs are completed. Runs which are prematurely aborted by the operator are not counted; their results are discarded into a special debug file, and HALEX chains back to HALINT without incrementing the run counter. After the seventh run, HALEX chains to HALWT instead of HALINT so that the apparatus finishes in the standby mode.

The high/low limits of the spirometer bell are maintained by HALEX just as they are in HALWT and HALINT. The halothane concentration, however, is controlled somewhat differently. HALEX controls the halothane expired/inspired gradient, not the absolute halothane concentration as in HALWT and HALINT.

Because of the extreme complexity of the HALEX overlay, a special section--which follows--is devoted entirely to it. Also, because of their complexity, special sections on halothane control and signal peak detection are discussed separately. Finally, a special error trap routine--used by HALWT, HALINT, and HALEX--is presented as optional reading.

Details of HALEX

The main overlay, HALEX, is composed of several individual sections. The first section initializes HALEX and is executed only once per run. The other overlays--HALWT and HALINT--have similar sections. Here special functional definitions are performed for interfacing the laboratory devices, taking moving averages, and converting digital units to

real units for each channel. Error conditions and their respective messages are then defined for subroutines ERROR and PEAK. Subroutine RANGE is then initialized--discussed further in the peak detection section. A sampling rate of 20 samples per second is then selected.

Lastly, two data files are read into memory. The first file is a read-only file generated by HALCAL. It contains all calibrations values to transform digital units to real units and vice versa. The second file is a read/write file which HALINT and HALEX use to interchange experimental information such as run count.

Initialization is then suspended until the subject's expired halothane level reachs 2.0% at which time initial crest/trough range limits for peak recognition are established for the tidal volume and pCO_2 channels via subroutines PRE-RANGE and RANGE. This ends the initialization process and begins the main "run loop". Each cycle of the main run loop (1) decreases the halothane set point by 0.05%, (2) readjusts peak recognition range limits for the tidal volume and pCO_2 channels, and (3) flushes the blood pressure cannula for half-a-second via a heparinized saline pump. Afterwards, the main loop enters the "monitor/control loop" contained within itself. The "monitor/control loop" returns to the main loop whenever the moving average of expired halothane declines by 0.05% (plus or minus 0.01%) at which time experimental data in the moving average array (ring buffer) for all parameters are cached in a storage array for that particular halothane concentration. As mentioned earlier, the main loop continues to repeat this data caching process until the expired halothane level declines to 0.65%, at which time the main loop terminates, and the run termination section begins.

The run termination section is executed only once per experimental run. It simply transfers the cached data from its storage array to disk under a particular file name for that specific run. It will either chain to HALWT if seven good runs have been performed leaving the apparatus in the standby mode; otherwise, it will increment the run counter (except on run abortions), save program variable parameters (such as run number) on the disk, then chain to HALINT to begin another run.

It is the monitor/control loop--contained within the main loop--which contains the bulk of the program. It is composed of two major sections--a data acquisition section and a fine apparatus control section--which execute every 7 to 15 seconds comprising one "collection pass". Results are sampled and apparatus control is adjusted once during each of these passes.

During data acquisition, whenever a collected parameter is made available to the acquisitioning loop by a lower software layer (such as a peak detection subroutine), that parameter is immediately saved in the four element input ring buffer associated with that sampled or computed parameter. The input ring buffer pointer is then advanced accordingly at the end of the monitor/control loop so that the next data collection pass stores its data in the oldest of four buffer locations for each parameter. An exception is made if an error condition occurs during the collection pass; in most of these cases, the ring pointer is not advanced for any of the parameters (all parameters use the same pointer) and the pass is restarted at the top of the monitor/control loop as soon as subroutine ERROR flags a message to the operator.

When it is time for the main run loop to cache parameter data for a given 0.05% incremental level of halothane, it simply averages the four values in each ring buffer for each parameter and saves them in a storage array for later transfer to disk. In this way, moving averages are continually being updated and saved while results from problem passes are purged.

The data acquisition section idles until the subject begins a breath--identified as a crest in the tidal volume (V_t) channel. It records the time, then detects the end of the breath--identified as a trough. During this period of expiration, it samples spirometer bell displacement on the V_t channel every 0.05 seconds. Beginning at the trough point, it now searches back through this collected data using the "slope difference" method (discussed later) to select a "near crest" point which is regarded as the true beginning of inspiration. It then uses this interval to determine peak inspiratory flow, inspiratory volume (regarded as V_t), and inspiratory time (T_i). Inspiratory volume is then divided by inspiratory time to estimate the mean inspiratory flow rate (V_t/T_i).

Next, the start of alveolar expiration--trough pCO_2--is detected. Unfortunately, trough V_t does not occur simultaneously with trough pCO_2 because there is a variable delay (a function of sample flow rate) associated with pumping the expired gas to the gas analyzers. As a result, this delay must be measured and the appropriate offset in the flow rate ring buffer must be applied when integrating minute pCO_2 and minute pO_2. The flow rate buffer can handle 36 samples. Sampling every 0.05 seconds,

this allows the buffer to accommodate an offset as large as 1.8 seconds between the V_t and pCO_2 channels.

At the trough point of pCO_2, this offset is computed and the integration of minute volumes for pCO_2 and pO_2 begins using real-time values of pCO_2 and pO_2 and buffered values for expiratory flow rate. These integrations occur once every 0.05 seconds. One should note that the same buffered offset applied to pCO_2 is also applied to pO_2 even though the sampled gas reaches the pO_2 head somewhat later. The problem is this: the pCO_2 head comes first so that a precise measure of crest pCO_2 can be taken prior to the gas dispersing the crest. The halothane head comes next so that a reasonably precise measure of end-tidal halothane can be taken before the crest is dispersed even further. This means the pO_2 head, the last head, is presented with a broader crest with much more offset delay. This introduces some error in the minute volume calculation for pO_2 while minimizing it for pCO_2. In summary, pCO_2 and halothane are felt to be more important than pO_2; therefore, the pO_2 head is placed in the poorest position.

During the initial phase of the minute volume integration, real-time expiratory flow rate is continually being sampled until it crests marking the end of expiration. At this point, flow rate is no longer sampled and integration is allowed to continue only as long as the buffered offset dictates. Afterwards, pCO_2 is allowed to crest--if it has not already done so--peak pCO_2 is sampled, and halothane gradient and pCO_2/pO_2 are computed; pCO_2/pO_2 is then outputted to the Beckman polygraph via the D/A channel.

Next, the end of inspiration is detected--as characterized by a trough on the V_t channel. Tidal volume is now computed based on the time between the last end of inspiration that occurred earlier along with the time-corrected form of minute pCO_2; both are plotted on the the polygraph. Minute volume and RQ are also calculated at this time.

Lastly, the cardiovascular parameters are sampled. Because of its averaging and superior crest/trough peak sampling capability, subroutine RANGE is used to find systolic and diastolic pressures from the blood pressure channel. The program then measures the time for another crest followed by a trough on the blood pressure channel to determine systolic period. Its reciprocal is then recorded as heart rate.

Now the monitor/control loop enters its fine apparatus control section. This section begins with upper limit spirometer bell control where an exhaust pump is turned on if the 8-liter bell is over the 3-liter mark. The pump remains on until the bell falls below the 1.5-liter mark--allowing 1.5-liters for hysteresis.

The halothane expired/inspired gradient control algorithm executes next; its discussion is left for a later section. Upon exiting this control algorithm, a time variable is passed to the following oxygen input algorithm which proceeds to open the pneumatic oxygen input valve for the prescribed time, thereby maintaining a steady decline in the halothane concentration of the apparatus and an uniform expired/inspired halothane gradient for the subject. This completes the monitor/control loop. If mean expired halothane is not a 0.05% multiple, the program will return to the beginning of the loop for another 7 to 15 second collection pass;

otherwise, it will fall through to the main run loop to cache parameter data as discussed earlier.

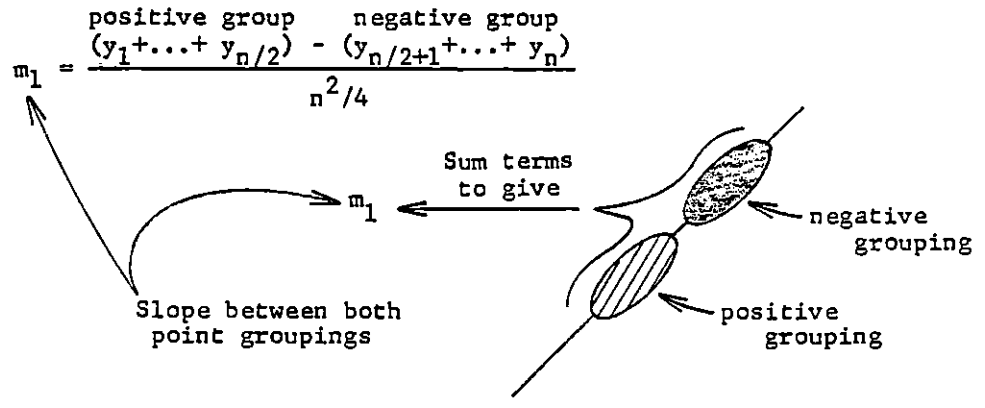
The dual-slope difference algorithm

The noise and overall shape of the tidal volume waveform complicates identifying the onset of inspiration. The tidal volume waveform is nearly flat until the point of inspiration when it suddenly drops; therefore, the onset of inspiration is best depicted at the point where the waveform takes on a sudden change in slope. In light of this, an algorithm was developed that senses the point where slope shifts from slight to very negative. Moreover, to prevent noise from confusing the determination, 8 points are used to evaluate each segment of the waveform.

The dual-slope difference algorithm begins by evaluating the slope of two adjacent 8-point line segments beginning at one end of the waveform. Next, it computes the difference between the slopes of these two line segments. It repeats this process all along the waveform, shifting its point of reference one point at a time, until all possible slope differences have been evaluated. Finally, it selects the intersection point between the two 8-point line segments that have the greatest slope difference as the onset point of inspiration.

Initially, a least-squares method was used to perform the slope calculation; however, this method took too much CPU time and was too sensitive to outliners (discordant points) [33] causing it to be too slow and unreliable in high noise environments. Since the y-coordinates are sampled as a time-series, a new method was developed that would take advan-

tage of this attribute to improve computational speed while, at the same time, avoiding a "squaring technique" to reduce the sensitivity to outliners. Figure 5 illustrates this new technique. Here the slope of a time-series of contiguous points is evaluated by summing them together into two large adjacent groups, then taking the difference between these groups. Although, the result must finally be divided by 0.25*N, where N is the number of points (8 in this experiment) that make up the line segment, this additional step can be skipped in this "maximal dual-slope difference" application since only relative maximal differences are characterized here.

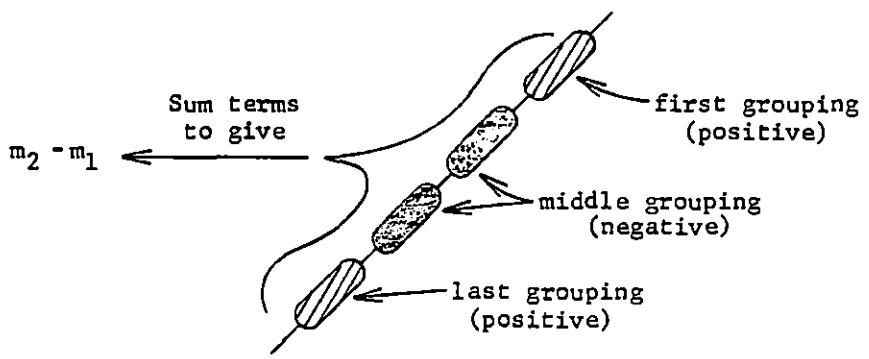


The above formula computes the slope of a time-series without using a least-squares method.

FIGURE 5. Slope calculation for a time-series

Using this new slope method, the slope equation in figure 5 can be subtracted from a second slope equation like itself to arrive at the dual-slope differences equation shown in figure 6. Here the time-series data is split into three separate groups such that, if N equals 8, the 8 points in the two outlining groups (4 points/group) are subtracted from the 8 points in the interior group to compute the slope difference. Once again, the division by 0.25*N can be skipped since only relative maximal differences are evaluated to determine the onset of inspiration.

$$m_2 - m_1 = \frac{\overbrace{(y_1 + \ldots + y_{n/2})}^{\text{first grouping}} - \overbrace{(y_{n/2+1} + \ldots + y_{3n/2})}^{\text{middle grouping}} + \overbrace{(y_{(3n/2)+1} + \ldots + y_{2n})}^{\text{last grouping}}}{n^2/4}$$



The above formula computes the difference in slopes between two separate parts of a time-series.

FIGURE 6. Dual-slope difference calculation

Peak detection routines

The peak recognition algorithm is often the most sophisticated part of a data acquisition system. But, in spite of its popularity in the literature, most papers fail to discuss real-time peak recognition, and the few that do only analyze the waveform after it has been completely sampled rather than in real-time. Granted, this is the most accurate approach since the program has total access to the entire waveform, thereby attaining more precise peak selection, but this method will not suffice in this research because other real-time data must be collected in conjunction with the peak point. If the program paused to collect the entire waveform prior to beginning peak analysis, valuable real-time data on other channels--which correlates with that peak event--would be missed before peak analysis has even begun. Alternately, the program could cache all the real-time data that correlates with an anticipated peak event until total waveform peak analysis is completed; however, considering the number of channels involved and the data sampling rate of those channels, the memory of the computer would soon be overwhelmed with data--only a small fraction of which coincides with a peak event and is worth saving. In summary, to prevent overflowing computer memory with extraneous data from many channels all anticipating a peak event, a real-time peak recognition routine is required for triggering mass data collection the instant the monitored peak event occurs.

Such a swift recognition routine must be able to anticipate the behavior of the monitored waveform based upon past experience. There are several guidance parameters set by both the operator and the program

which assist in characterizing the past behavior of each peak event. The operator sets a maximal second derivative noise limit parameter (f" limit) for each channel being monitored. Samples on a data stream which exceed this f" limit are immediately rejected as noise spikes. The operator also sets a time limit for peak searches; if a peak fails to occur within this time limit, a warning condition is flagged and the operator is alerted. (See the "exception handling" section for further discussion.)

The program adjusts one parameter type on its own; these are the crest/trough range limits or "range limits" for short. They give the program the edge in discriminating the valid peaks from the others. All legitimate crests must be greater than the crest range limit, and all legitimate troughs must be less than the trough range limit. There are separate range limits for each channel since crest/trough values vary from channel to channel. The setting of these range limits is accomplished via subroutines PEAK DET, PRE-RANGE, and RANGE which are discussed below. (PEAK DET is read as "peak detect".)

Subroutine RANGE performs the actual range limit calculations; however, it basis these initial calculations on the four most prominent crests and troughs recorded by subroutine PRE-RANGE. Hence, subroutine PRE-RANGE is called prior to RANGE to establish initial range limits. For subsequent range limit adjustments, PRE-RANGE is no longer required providing that PEAK DET is called periodically. In contrast to PRE-RANGE, subroutine PEAK DET updates the peak history of the most recently monitored channels for RANGE. In summary, PRE-RANGE initializes

the peak history; whereas, PEAK DET updates it. The RANGE subroutine does the actual range limit calculations; therefore, RANGE must be called whenever it is time to establish or update the range limits. This is why PRE-RANGE always calls RANGE upon completion.

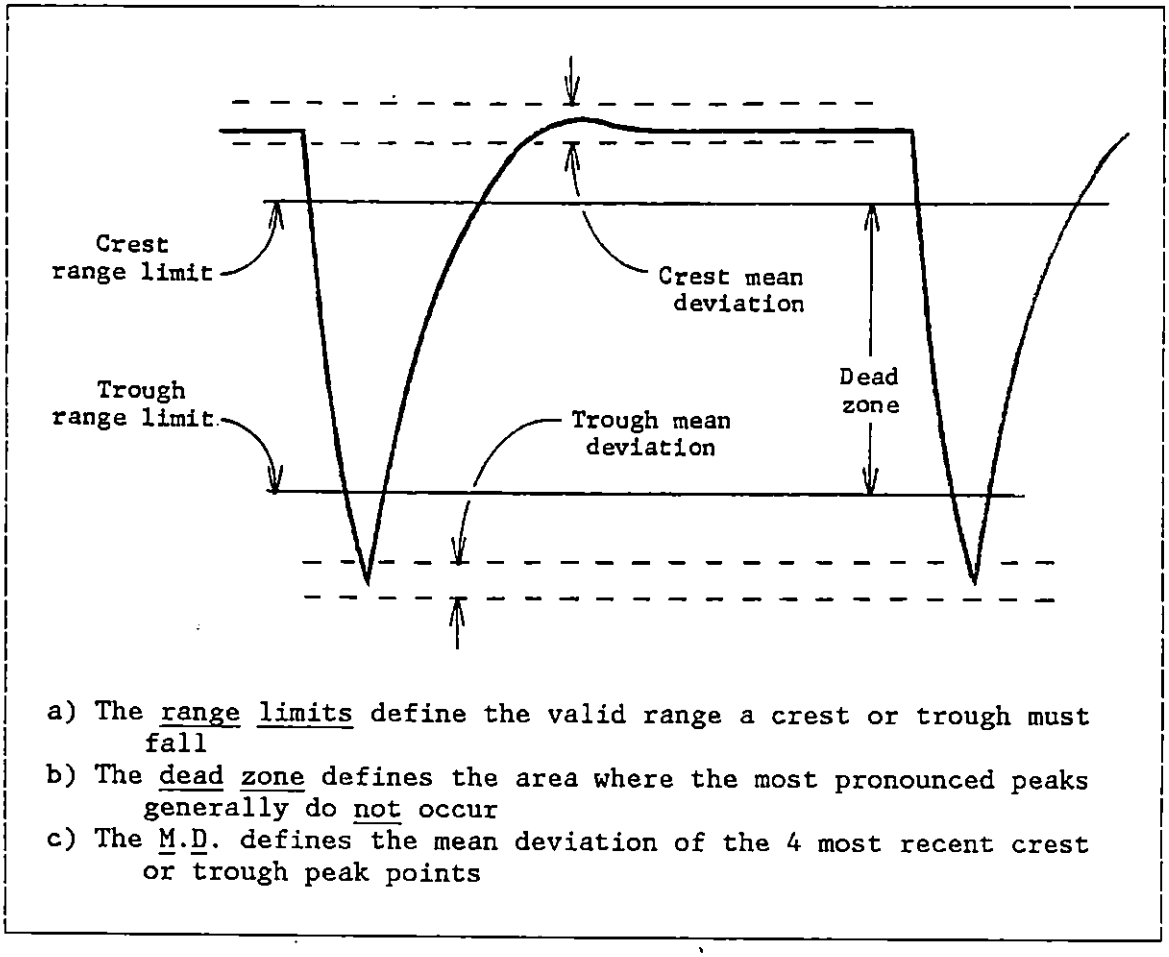


FIGURE 7. Waveform attributes scrutinized by the peak detection routines

Subroutine PRE-RANGE begins by nullifying the range limits so that every peak--regardless of how large or small--is recognized by PEAK DET

provided that they meet the other noise guidelines established by subroutine PEAK DET and the operator. (Optionally, PRE-RANGE can use the old range limits when establishing the new ones, but this approach runs the risk of disregarding the entire waveform if it falls inside the old range limits known as the "dead zone".) Afterwards, PRE-RANGE repetitively calls PEAK DET for a predetermined time. During this time, PEAK DET reports every crest and trough encountered to PRE-RANGE which in turn saves the values of the four highest crests and four lowest troughs encountered.

This stored crest/trough data is next passed to subroutine RANGE for computing the means and mean deviations (M.D.s) for both the four crests and troughs. Next, RANGE computes the crest/trough range limits as described below, then it returns to the main program.

For crest searches: (crest limit) = (crest mean) - 8(crest M.D.)
For trough searches: (trough limit) = (trough mean) + 8(trough M.D.)

Like subroutine PRE-RANGE, subroutine PEAK DET also logs crest/trough data; however, rather than logging the four most pronounced crests and troughs as PRE-RANGE does, PEAK DET logs the four most recent crest and trough values. This allows the main program to call RANGE without first calling PRE-RANGE if it wants the new range limits to be computed from most recent data rather than maximal crest/trough data. This is the preferred method since it yields more realistic range limits and is much faster. In contrast, calling PRE-RANGE first is slower because this routine waits for several new peaks to occur beforing exiting; whereas, calling RANGE directly eliminates this wait since current data on past

peak behavior has already been saved by subroutine PEAK DET. (Note that if PEAK DET has not been recently called for the channel of interest, then it is better to call PRE-RANGE prior to calculating new range limits with RANGE so that fresh peak data is available to RANGE.)

The primary purpose of subroutine PEAK DET is to detect peaks. Prior to beginning a peak search, PEAK DET is passed several initial parameters pertinent to the search. These parameters include the sampling rate, maximum noise limit, current range limits, channel number to be sampled, and whether its a crest or trough search.

After these guidelines are established, PEAK DET samples the appropriate channel and computes the first and second derivatives (f' and f") of the input data stream. Afterwards, it performs a sequential series of tests and operations to determine if the current sample is a valid peak. As soon as a collected data point fails a test, the program returns to the beginning of the search loop and waits to collect and analyze the next incoming data point. The first test confirms that the f" is within the set f" limit for that particular channel. The most pronounced peak value (for the immediate peak search) is next updated. This would be the maximum encountered value for a crest search (X_{max}) or the minimum encountered value for a trough search (X_{min}).

In order for a peak to be considered valid, it must double back by one-half the value of the current second derivative (f"/2). This is accomplished by checking to see if the following corresponding equality is satisfied:

For a crest search: (X_{max}) - (newest value) > (f"/2)
For a trough search: (newest value) - (X_{min}) > (f"/2)

Finally, PEAK DET verifies if the candidate peak meets the range limit requirements. This means the suspected peak value must be over the crest range limit for a crest search or under the trough range limit for a trough search. If this requirement is met, PEAK DET marks the location of the peak on the Beckman polygraph (via the D/A converters), then updates the most recent peak and f" values for later use by subroutine RANGE in making range limit adjustments. (Note that this last step is skipped whenever PRE-RANGE is the calling routine since PRE-RANGE modifies these parameters itself and slightly differently than PEAK DET.)

If, over the specified sampling period, none of the sampled data points fell below the f" limit or doubled back enough, a warning message is issued indicating that no recognizable peaks occurred within the allocated period for the monitored channel. If the evaluated points meet both the f" limit and double back requirements but failed the range limit requirements, a warning message is sent to the operator indicating that the peaks are too small after which PEAK DET modifies the corresponding range limit value just as RANGE does only using X_{max} (crest search) or X_{min} (trough search) instead of mean values:

For crest searches: (new crest limit) = (X_{max}) - 8(crest M.D.)

For trough searches: (new trough limit) = (X_{min}) + 8(trough M.D.)

This adjustment by PEAK DET accounts for a drift in one of the range limits but not the other. As a result, if the new range limit overlaps with opposing range limit, then the opposing limit is also adjusted. For example, if a new crest limit falls below the current trough limit, then the trough limit is also adjusted. The equations below describe this second, conditional adjustment:

Adjustment on an overlap during a crest search--
(new trough limit) = (crest limit) - (dead zone)
Adjustment on an overlap during a trough search--
(new crest limit) = (trough limit) - (dead zone)

In making the above adjustments, the algorithm assumes a simple shift in both crest and trough peaks so that the dead zone (and the size of the waveform) remains the same. On the other hand, if the waveform is shrinking, then the dead zone area must shrink as well. To facilitate this, the dead zone value is cut in half for each iteration that requires opposing limit readjustment. In this way, crest and trough range limits are both dynamically translated up and down as well as moved together so as to best forecast the next peak value. Upon completion of these limit adjustments, subroutine PEAK DET exits with the appropriate warning code to the calling routine.

There is one precautionary note to consider before leaving the topic of peak recognition. In evaluating the relationships for judging whether or not a waveform doubles back sufficiently, the attentive reader would have noticed that these relations--reproduced below--could be inadver-

For a crest search: (X_{max}) - (newest value) > (f"/2)
For a trough search: (newest value) - (X_{min}) > (f"/2)

tently satisfied upon random entry into a waveform. For example, upon beginning a crest search, if PEAK DET starts by sampling the negative going edge of a waveform, the first relation would be immediately satisfied before a valid crest was encountered. This is because PEAK DET looks at the sign alone of f' (which is negative at the onset of a crest) when the real crest occurs on the sign change of f' from positive to negative. To avoid this potential problem, subroutine PRE-PEAK DET is

called instead whenever asynchronous entry is made into a waveform. PRE-PEAK DET simply synchronizes the calling of PEAK DET to the waveform by first calling PEAK DET to search for the opposing peak. For instance, during a crest search, the trough (whether it be the valid trough or not) is first found to insure the waveform is positive going when PRE-PEAK DET calls PEAK DET a second time to find the valid crest. In summary, for peak searches, PRE-PEAK DET is called for asynchronous entry into the waveform of interest; whereas, PEAK DET is called for synchronous entry into the waveform. It should be noted that PRE-PEAK DET disables the peak history updating function of PEAK DET on the first call since the first peak detected by PEAK DET is erroneous. Also, since PRE-RANGE reports only the highest crests and lowest troughs encountered out of the numerous peaks detected, it automatically disregards all erroneous peaks; therefore, it always calls PEAK DET directly even on asynchronous waveform entry. It is this ability to reject erroneous peaks which makes PRE-RANGE and RANGE the most powerful subroutines in this peak recognition package.

The regulation of halothane concentration

Although the iterative steps employed for halothane maintenance in all three overlays--HALWT, HALINT, HALEX--are similar, there are several differences. For instance, HALWT and HALINT regulate halothane concentration directly; when halothane concentration is too low, they add more halothane (via the infusion pump) to the system. In contrast, HALEX trys to reduce--not increase--halothane concentration steadily. It enacts

this reduction by maintaining a constant halothane gradient between expired and inspired halothane gas by adding oxygen to the system whenever the gradient is too small.

Other differences in halothane control for these overlays lie in their initialization protocol. This protocol will be examined next for HALWT and HALEX. HALINT will be excluded from any further discussion since its control algorithms are exactly identical to that of HALWT. The sole difference lies in their set points. HALWT is set to regulate inspired halothane at 1.1%; whereas, HALINT regulates at 2.2%.

HALWT begins by sampling crest halothane concentration. If a crest is not detected within the "time out" period, subroutine ERROR flags a "subject is not breathing" error, and HALWT must then use the most recently sampled halothane concentration instead. Next, HALWT branches to either the initialization or maintenance mode depending on whether or not inspired halothane is less than 0.8%. (Note that HALWT can fall back from the maintenance mode to the initialization mode if a transient drop below 0.8% halothane occurs.) In the initialization mode, HALWT will begin to fill the apparatus with both oxygen and halothane. Oxygen is switched off when the spirometer bell reaches 3 liters and halothane is switched off when it reaches 0.8%. The operator adjusts the rate of infused halothane to 0.4944-ml/minute so that the halothane concentration just exceeds 0.8% during this initialization.

HALEX initializes somewhat differently. It begins by branching to either the initialization or maintenance mode based on whether the expired/inspired halothane gradient is less than 0.01% or not. Normally,

expired halothane is higher than inspired (in HALEX) so that the halothane gradient is positive and halothane concentration slowly declines. When first entering HALEX from HALINT, however, this gradient is negative since HALINT was previously trying to raise halothane concentration. HALEX recognizes this by noting that the gradient is less than 0.01% and therefore branches into the initialization mode. In the initialization mode, HALEX tries to reverse the gradient by introducing oxygen into the apparatus for 10 seconds so that halothane concentration begins to decline. This initialization process may repeat itself on forthcoming collection passes if the halothane gradient still has not reversed. The rate of oxygen introduction is adjusted to 4-L/minute by the operator so that 10 seconds of flow normally delivers just enough oxygen to reverse the gradient.

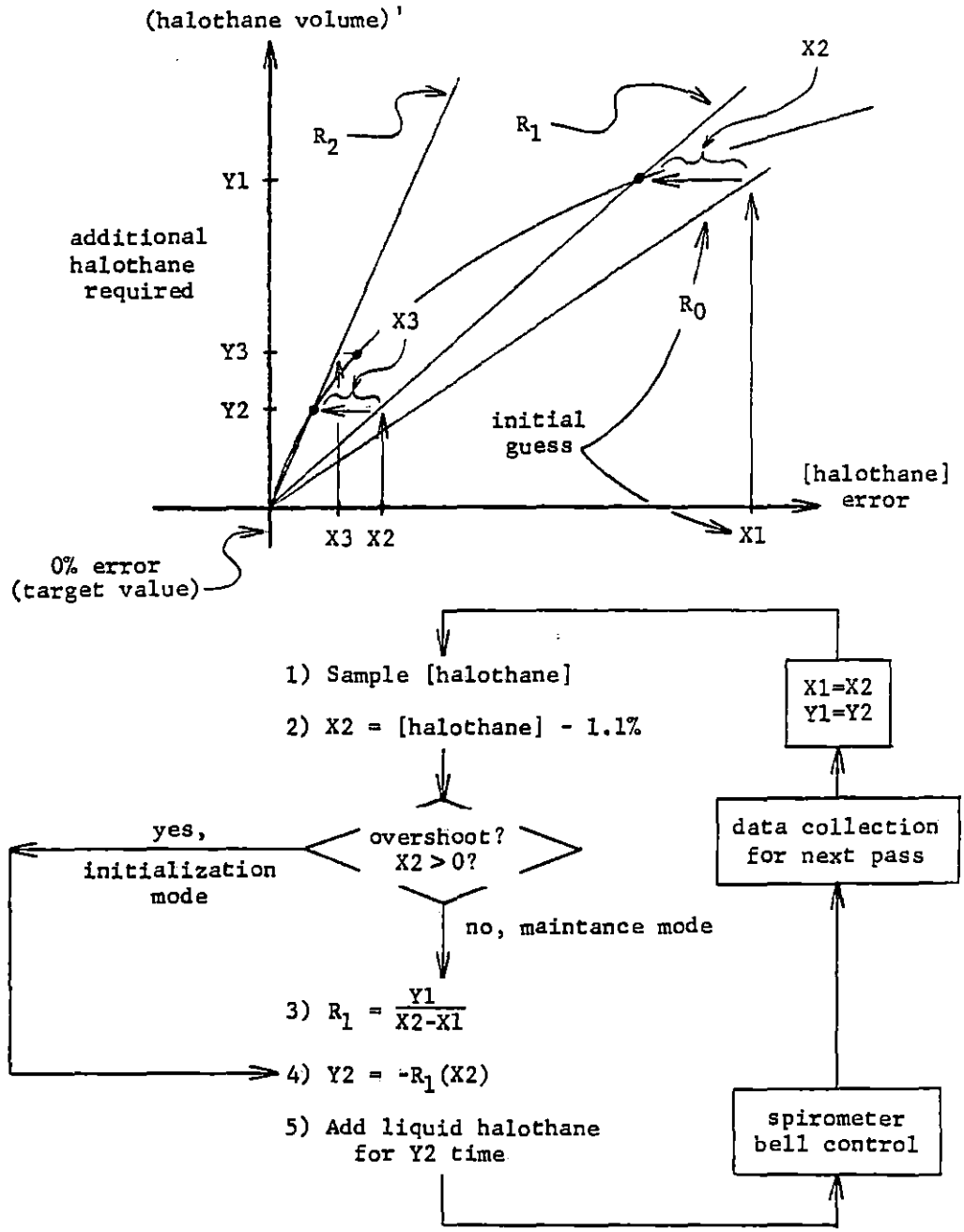
The basic steps for the halothane maintenance mode are very similar for all program overlays. The major differences are:

1. HALWT (and HALINT) introduces more halothane into the system thereby raising its concentration to a preset level; whereas, HALEX introduces oxygen into the system in order to reduce halothane concentration and maintain a preset halothane gradient.
2. HALWT (and HALINT) uses a simplified version of the Regula Falsi method [34] for its control management; whereas, HALEX uses the secant method [34]--which is a variation of Newton's method--for its control.

The halothane maintenance control steps for HALWT (and HALINT) and HALEX, shown in figures 8 and 9, are discussed below:

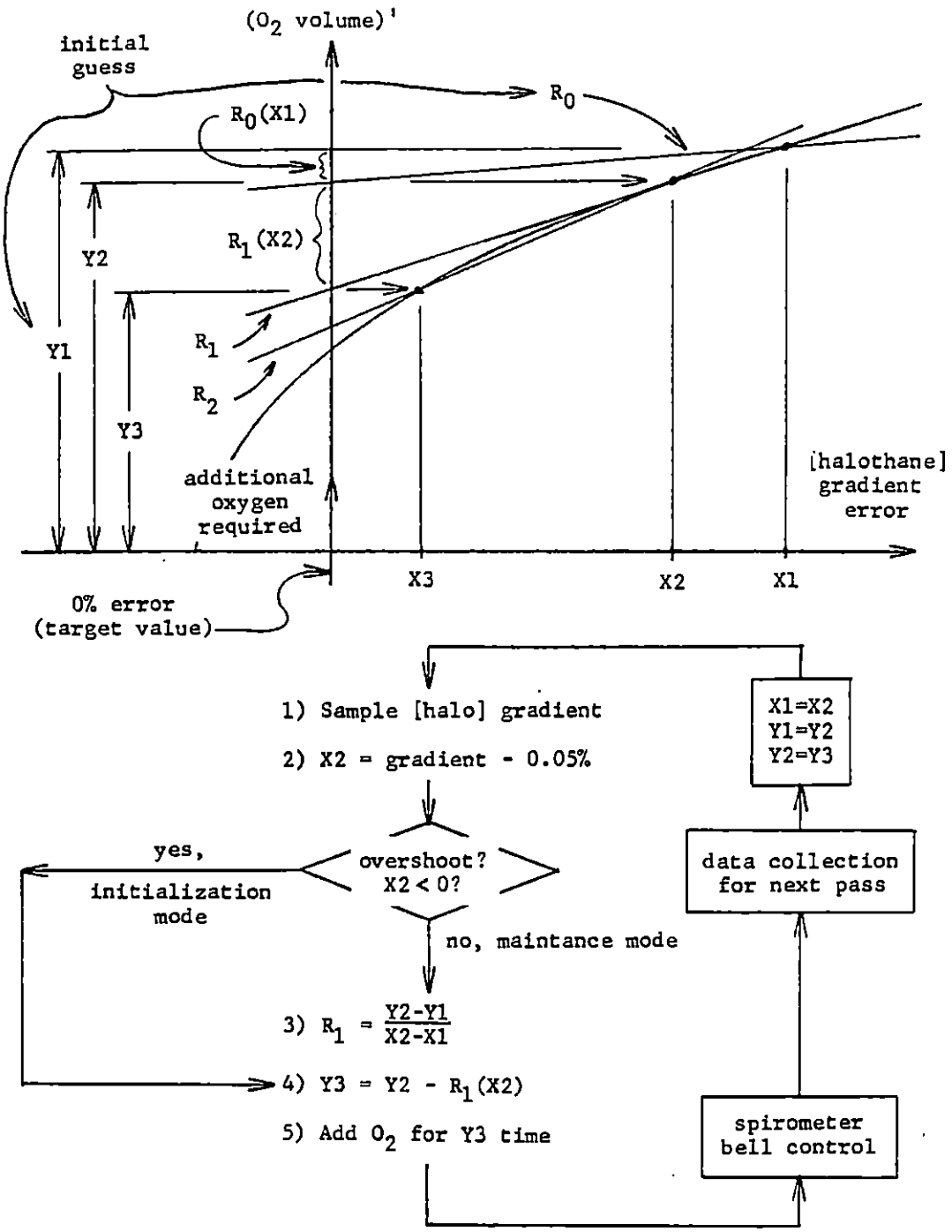
1. Inspired halothane concentration (HALWT) or its gradient (HALEX) are first sampled.
2. A numerical calculation is then performed so that the iterative method converges to the appropriate preset point.
3. A decision is then made concerning whether or not to recalculate the slope of the tangent line, R. This depends on whether the Y2 variable collected from the previous iteration is meaningful (positive and exists). For the first iteration in the maintenance mode, Y2 does not exist; therefore, the R calculation is skipped and an initial guess for R--defined at the start of the overlay--is used to complete the calculation. If the control algorithm has exceeded the set point so that negative halothane or negative oxygen must be added to the system, then Y2 is also negative and the R calculation must be skipped.

   If Y2 exists and is positive, then the slope of the tangent line, R, is recalculated so that it more precisely models the halothane control function. The slope, R, has units of {ml-halothane / %halothane} for HALWT (and HALINT) and {ml-oxygen / gradient halothane} for HALEX.
4. Next, the time required for infusing more halothane or oxygen into the apparatus, Y3, is computed.



This algorithm is employed by both HALWT and HALINT for respectively maintaining and boosting the halothane level of the system.

FIGURE 8. The modified Regula Falsi method for halothane control



This algorithm is employed by HALEX for monitoring a steady concentration gradient between expired and inspired halothane thereby creating a stable decline in the halothane level of the system.

FIGURE 9. The secant method for halothane control

5. Finally, either liquid halothane is infused for time Y3 (HALWT and HALINT) or oxygen is introduced into the apparatus for time Y3 (HALEX). If control exceeds the set point so that Y3 is negative, this step is skipped; the apparatus is then allowed to fall passively below the set point with time.

This completes the halothane control protocol. The program then proceeds to the spirometer control section of the monitor/control loop.

Notice that more precise controlling could be realized in step 5 whenever there is an overshoot by adding oxygen when halothane dosage is overestimated or adding halothane when oxygen introduction is overestimated. This enhancement, however, would require a dual-slope approach so that two Rs must be computed--one for oxygen input and one for halothane input. Due to this additional complexity, allowing time to passively correct overestimates was considered adequate.

Exception handling

Several different trouble conditions may occur during the course of an experiment that require immediate action by the apparatus and/or operator in order to preserve the integrity of the results and the life of the subject. As a result, special exception handling procedures are used to inform the operator of any life-threatening complications as well as to suspend data collection (in some instances) and allow the main program to proceed without hang-up.

Initially, exception handling proceeds as follows: First, an alert condition is flagged by a lower software layer such as the peak recogni-

tion routines. An error code is set to identify the condition, the operator is alerted with a terminal bell sound, and program control is passed back to the higher software layer. On mild warnings, this intermediate layer will intercept the alert condition and restart its processing as though nothing happened; otherwise, on more severe warnings, it will return control to the main program which must take further action.

The main program handles warnings in three steps. First it determines the nature of problem based upon the kind of warning that occurred in context to its current operation. For example, if a "no peak detected" warning occurs while measuring tidal volume, then the main program would conclude that the subject has ceased breathing and would modify the error code accordingly. Second, the main program calls subroutine ERROR to report the problem to the operator. Finally, the main program either continues from where it left off if the error is not severe, or it restarts the current collection pass at the top of the monitor/control loop (ignoring results sampled on the current pass) if the problem could have adversely affected results collected on the current collection pass.

Alert conditions may be flagged by either subroutines PEAK or RANGE. For each input channel that PEAK samples, there is an upper limit parameter for second derivative noise and maximal waiting time. If this time out period expires without detecting a peak or if all detected peaks exceed the second derivative noise maximum, PEAK flags a "no peak within time window" alert. If peaks did exist during the sampling period, but were too small to meet the minimal size requirements established earlier

by subroutine RANGE, then a "peaks are too small" alert is flagged; PEAK then relaxes these size requirements as discussed in the peak detection section.

When RANGE is called, it expects to encounter four peaks within its own time out period. If it does not, it generates a "peaks occurring too infrequently" warning.

Any of these warning/alert conditions can cause the main program to flag a trouble condition. The main program also checks for numerous other trouble conditions on a routine basis. For example, whenever the offset delay between the tidal volume channel and the pCO_2 channel is too high, a warning is either given to "change the air filter" in the pCO_2 analyzer or "abort the run" if the offset becomes so high that it overflows the offset ring buffer.

Cardiovascular parameters are also routinely examined by the main program. Heart rate and blood pressure (pulse pressure included) are monitored and reported if they drop below a set limit.

RESULTS

Much was learned from the first experiment which can be applied to the design of similar experiments in the future. The following is a summary of some of the problems encountered before and during the first animal run.

By itself, the spirometer blower failed to distribute the infused halothane through the system rapidly enough. This sluggish response caused the computer to infuse too much anesthetic before perceiving the anesthetic level as too high. This problem was significantly alleviated by adding a small Air Marine blower (serial A6816) to the anesthetic circuit between the vaporizer and inhaler Y.

The most serious problem encountered in the system concerned the high-pass filter which coupled the spirometer channel to the tidal volume channel. The RC time constant for this filter was too long to allow for rapid enough recovery after the spirometer bell was abruptly shifted up or down to either introduce oxygen or exhaust excess air from the system. As a result, the computer could not immediately sample meaningful expiratory flow and tidal volume data from the V_t channel which constantly went into long-term saturation following each spirometer-bell adjustment. During the experiment, an attempt was made to shorten the time constant of the filter, but this only served to distort the tidal volume waveform.

The heating mantle for the halothane vaporizer produced enough heat--even at low levels--to evaporate the liquid halothane within the delivery tube. As a result, when the computer briefly invoked the

halothane infusion pump to deliver more halothane, fewer drops than expected came out. This in turn caused the halothane control algorithm to erroneously increase the (halothane concentration)/(infusion time) ratio, R. Moreover, this excessive R adversely increased the halothane delivery period on the succeeding iteration causing the halothane level to overshoot the target concentration. This problem was lessened by three modifications: First, all further R value adjustments were only based upon iterations that infused halothane for more than four seconds, thereby allowing enough time to purge any evaporated halothane out of the delivery tube. Second, the temperature of the heating mantle was decreased to 38°C. Third, the protrusion of the delivery tube into the hot vaporizer was reduced in length to 1-cm.

After the system was initialized with 2.2% halothane by the run initialization overlay, HALINT, the animal's expiratory concentration never reached 2% as anticipated by the following overlay, HALEX. To overcome this and continue the experiment, the halothane concentration of the system was manually raised to 2.7%. This problem can be alleviated by having HALINT initially raise the inspired concentration to 2.7%, or so, then wait for the expired breath to reach 2% before invoking the HALEX overlay.

After the end of the first experimental run, the animal was left so lightly anesthetized--at 0.65% halothane--that he woke up before the beginning of the second run. In the future, the anesthesia should remain well above 0.65% halothane.

Generally speaking, the peak recognition routines performed well; however, when the signal on an input channel ceased completely, subroutine PEAK DET would invoke the appropriate warning message, then (without further consideration) proceed to readjust the range limits to suit the signal in error. Later, when the signal recovered, these fallacious range limits would cause PEAK DET to select smaller premature peaks on the waveform rather than waiting for the highest crest or lowest trough. This difficulty was remedied by selectively preventing PEAK DET from adjusting any range limits whenever both crest and trough limits needed to be jointly adjusted to prevent overlap. Such a condition would imply a sudden cessation of the input signal. Whenever this condition occurred, the operator would be warned. If the operator saw that the input waveform in question was reasonable, he would momentarily flip a sense switch on the computer giving PEAK DET the go ahead to adjust both range limits; otherwise, PEAK DET would proceed without doing so. This maneuver was found only to be necessary two or three times during this experiment when the system was malfunctioning.

The sampling rate for the peak detection routines in the HALCAL calibration overlay was initially set at 5 samples/second. This created as much as 30-ml error in the tidal volume calibration. When the sampling rate was increased to 20 samples/second, the error in tidal volume calibration was reduced to 2-ml. Apparently, the sampling rate for measuring tidal volume must be at least 20 to insure reasonable accuracy.

The higher sampling rate of 20 samples/second worked well for all aspects of data collection except the real-time integration. Here the

computer needed to compute flow and sample both pCO_2 and pO_2 while performing real-time integration for minute pCO_2 and pO_2 20 times a second; it simply could not keep up. An attempt was later made to slow the sampling rate to 10 samples/second at the expense of losing some accuracy for the tidal volume mensuration, but this was still not slow enough. Moreover, halving the sampling rate also halved the number of points used to evaluate the onset of inspiration via the dual-slope difference method. (Initially, this pattern recognition algorithm used two 8-point line segments; whereas; now it was using two 4-point segments.) This reduction in resolution caused the success level of the recognition algorithm to fall from 100% to 95%. This performance was unexceptable.

DISCUSSION

Performance of the Apparatus

The tidal volume channel continually went into saturation whenever the spirometer bell was filled or emptied. This occurred because the coupling capacitor (shown in figure 3) became sufficiently charged during each bell adjustment to drive the V_t channel into saturation. There are two approaches to solving this problem. One approach involves clamping the AC output signal so that it lies within the operational range of the amplifier. Currently, full deflection of the V_t channel occurs at plus or minus 0.1-volts; therefore, AC signals exceeding 0.1-volts would need to be clamped.

Unfortunately, the above approach would cause a slight offset in the AC-coupled signal each time the bell was adjusted. This means the range limits for the V_t channel must be re-evaluated for each spirometer adjustment--a time consuming process. An alternative approach would involve estimating the DC offset produced when filling or emptying the bell, and applying this offset signal to the minus side of the differential amplifier for the V_t channel. Hence, the offset produced by the bell adjustment would be canceled out. This alternate approach assumes the average flow rate of gases to and from the spirometer will be stable enough to permit accurate estimations of these DC offsets. Whether these estimations are sufficiently precise to make this second approach more attractive than the first remains to be seen.

As mentioned earlier, the liquid halothane remaining in the supply tube for the halothane vaporizer slowly evaporated during long periods of disuse. As a result, on the following halothane infusion, air was being infused rather than liquid halothane which made regaining smooth halothane control more difficult. Although reducing the heat on the vaporizer diminished this problem, some alternative solutions may prove to be better.

One alternative solution involves infusing the liquid halothane directly into the squirrel-cage blower casing for vaporization. This would allow the extensive turbulence within the blower to promote immediate halothane vaporization with minimal heating, thereby reducing halothane evaporation in the delivery tube. Naturally, a "cool running" blower is desirable for this application; the blower employed in this experiment would not be appropriate because it gets too hot.

Another method, employed by Fukui et al. [35], would introduce the halothane as a concentrated gas--via a commercial vaporizer--rather than a liquid. A pneumatic valve can then be used to control the oxygen going into the vaporizer thereby giving the computer the means to regulate halothane introduction.

Discussion Concerning the Software Design

As mentioned earlier, evaporation of the halothane liquid in the halothane delivery tube caused the halothane control algorithm to set R--the halothane infusion ratio--excessively large to compensate. This, however, caused the halothane concentration to overshoot its target con-

centration on the following iteration. One of the methods employed for preventing R from becoming erroneously large involved recomputing R only after long infusions of halothane in order to insure that all halothane gas is purged from the delivery tube. There are, however, two purposed enhancements concerning the R computation which may further reduce this problem:

The first enhancement would place an upper limit on all R adjustments. Often times, when R is misestimated, R will be left artificially large because of the negligible increase in halothane concentration following an infusion of gaseous halothane from a nearly empty delivery tube. A simple limit test on R, rejecting the obvious overestimates, would improve the stablity of halothane control.

Finally, the computed R values from the most recent and acceptable iterations could be averaged to produce a highly stable "effective R". As its stands now, the value of the computed Rs oscillate; taking a moving average would level out these oscillations and further stabilize halothane control.

Development of the peak detection routines was probably the single most time-consuming venture of this project. In the final analysis, these peak detection routines performed well; however, there is still room for several enhancements. One such enhancement would use A/D units rather than real-units for the initial data sampling and data screening functions.

Using A/D units instead of real-units in the lower software layers would greatly accelerate the data sampling capability of the peak detec-

tion routines. For example, integer arithmetic--which executes 8 to 10 times faster than floating point arithmetic--could be employed in the data screening routines which must process all points being sampled regardless of fate. Since only 5% of the points sampled are actually peaks, 95% of the scaling multiplications that convert A/D units to real-units could be eliminated if the conversion was postponed until it was firmly established that the candidate peak was truly a peak.

Also, using A/D units in the lower software layers would allow expressing offsets and noise derivative limits (f" limits) in terms of A/D units. For instance, the program could verify that all valid crests and troughs detected were separated by at least 1% of the deflection for all A/D channels by verifying that the crests and troughs differed by at least 10 A/D units. Similarly, all the input channels could be programmed to reject noise that is greater than 1% of the deflection by setting the f" limit to 10 A/D units.

Data collection rates could be further increased by rewriting part of the peak detection routines in assembler. The main peak detection routine, subroutine PEAK DET, is the major bottle-neck; writing its peak processing algorithm in assembler could halve peak detection time.

Real-time integration of minute pCO_2 and pO_2 could not be performed at 20 times/second as desired because the computer was not fast enough. This problem may be alleviated by speeding up the peak detection routines as discussed above and caching the expiratory flow, pCO_2, and pO_2 data for later processing. In this way, the actual integration calculations can be performed later in the program when time allows. Unfortunately,

this data caching approach requires enough computer memory to store three parameters sampled 20 times/second for up to 10 seconds. Whether the computer can accommodate this 2k-word requirement is uncertain.

Enhancements to the system software

The present software design was made needlessly complicated by the inability of the OS/8 operating system to implement foreground/background program execution. As a result, the main program and its associated layers had to use many watchdog time loops for determining whether to continue waiting for the anticipated event or to issue an error message (e.g., "Waiting time expired, dog not breathing.") and then proceed on. Moreover, these watchdog time loops are a wasteful use of computer time. Conceivably, the computer could be monitoring and controlling several events simultaneously if it were not hung-up in a time-checking loop most of the time.

Foreground/background programming works by allowing the data sampling and fine control routines to operate in the background as an interrupt driven process while the main program runs in the foreground. For example, a clocked interrupt routine may be called 100 times a second to take samples from 6 A/D channels and then place these samples in a data queue for the main program to access when convenient. The background job could also maintain the level of the spirometer bell by adding or subtracting gas from the system as necessary to keep the spirometer bell within its limits of travel. Halothane control could also be maintained in a similar fashion by making it part of the background job. These background activities would all be transparent to the main program.

The main program and its exception handling routines (error traps) would run as the foreground job. It would be primarily concerned with processing the data sampled by the background job as well as issuing any error messages and acting on any error conditions within its control. It is this kind of system hierarchy that would have made the main program easier and quicker to write, debug, and maintain as well as improving the quality of the research by more productively using CPU time for sampling more channels simultaneously and at a higher rate.

Data sampling and control operations often require access to the interrupt system of the computer so that event handling routines can be invoked as soon as the interrupting event occurs. Using interrupts usually means writing in assembly language, which is very difficult to debug. In recent years, however, a new computer language called Modula-2 has been developed which allows access to the interrupt system without the use of assembler. This new language would be ideal for developing background routines for experimental control and data collection where interrupt driven events are involved. Moreover, its modular design and structure (considered superior to its forerunner, Pascal) would lend itself to the development of the main program as well. It is this author's expectation that Modula-2 will become an important software enhancement for many laboratory computer systems.

The ultimate system for computerized monitoring and control of laboratory experiments would support event driven control and data collection. This feature is commonly found in event-type simulation languages such as GASP IV [36]. In such simulation languages, timed events can be

"scheduled" to begin after other timed or conditional events. For example, the main program could schedule an event to begin filling the spirometer bell for 10 seconds after the bell has reached a specified low mark. The beginning of this event could then automatically schedule a another event that would then add enough halothane to reach a specific level. Finally, the conclusion of this last event could reschedule the initial event to refill the spirometer, and so forth, thereby creating an event-controlled control loop. Using this scheme, an event-type simulation language can simulate a chain of reoccurring events (all of which are dependant upon one another) in order to define a process. Similarly, this same event/time driven method could be employed to control a laboratory process and sample experimental data thereby simplifying the development of the main control program.

The main program for the experiment would be composed like an event simulation model. The main program would schedule events--either timed or conditional--to begin relative to the beginning or ending of other timed or conditional events. These events would in turn schedule still other events for data collection and control until the entire experiment was defined. The events currently pending (scheduled) or in progress would be placed in an event queue. The event manager (and interpreter), which would be part of the control language and separate from the main program, would periodically examine the events in the queue to determine if the condition or time is right for some of the events to begin, to end, or to schedule new events. In summary, the event manager would be a generalized monitor that is periodically invoked by an interrupt-driven

clock routine to examine, update, and carry out the scheduled events as defined by the main program. It would remain unchanged from one experiment to another; whereas, the main program would be unique from one experiment to another since it defines the experiment to be performed by defining the sequence of events that are to occur (and reoccur) throughout the progression of the experiment.

CONCLUSIONS

Since this apparatus, which measures depth of halothane anesthesia via monitoring respiratory depression, never achieved full functionality due to developmental problems, final conclusions concerning its overall performance are difficult to draw. Nevertheless, following several modifications, most individual functions of the system appeared to perform well. The major problem still remaining in the system concerns the signal conditioning of the V_t channel. Somehow, this signal needs to be offset for each spirometer adjustment to prevent the V_t channel from going into saturation.

Surprisingly, the PDP-8 computer is not fast enough to perform the integration of minute pCO_2 and pO_2 in real-time. Caching of the intermediate data will be necessary so that the integration can be postponed until a more opportune time. Also, rewriting parts of the peak detection algorithm in assembly language and avoiding needless scaling multiplications (converting A/D units to real-units) may be necessary to facilitate real-time processing.

Several innovative algorithms were developed and implemented, for this project, with good success. The most complex comprises the peak detection routines which perform a peak recognition function during simultaneous data collection. This is partly possible through the use of "range limits" and "mean peak deviations" that record the past peak behavior of each input channel giving the peak detection routines the foresight to assess whether the current incoming data stream depicts a

crest or trough. Although this approach may not be as precise as an after-the-fact analysis, it does make very reliable real-time predictions.

The dual-slope difference algorithm performed well during the experiment provided that at least 16 points were used for the pattern recognition analysis. Its high processing speed and noise immunity proved it an invaluable tool for determining the "true" onset of inspiration.

The numerical methods for controlling halothane anesthesia performed reasonably well throughout the experiment. Although the secant method--employed in HALEX--permitted the halothane gradient to drift by 0.03% over a run, both methods still maintained good control.

There remain several practical problems concerning the overall implementation of this project. Current constraints require sequential execution of each operation by the computer. This is somewhat unfeasible since it allows one very long event in the sequence to hang-up the entire process. As a result, all control and much of the data collection for the experiment halts. Moreover, this sequential design allows only one attribute to be analyzed at a time so that only a small sample of the available data is ever collected by the computer. A considerably more feasible design would allow each control and data sampling function to execute independently of the other in an interrupt, clock-driven environment while the main program could gather, process, and store the results as time permits. This design hierarchy is made possible with the Unix operating system where the control and data collection programs operate in the background environment oblivious to one another and the main pro-

gram. They all pass their data, via a message queue, to the main program which is executing in the foreground at a lower priority. The main program then retrieves the data from the message queue for analysis and storage. Moreover, the main program also retrieves operator instructions from the console, thereby giving him full control over the experiment. Such a design architecture as this is needed to make this project more feasible to develop, debug, maintain, and control.

BIBLIOGRAPHY

1. Mori, K., W. D. Winters, and C. E. Spooner. 1968. Comparison of reticular and cochlear multiple unit activity with auditory evoked responses during various stages induced by anesthetic agents. II. Electroenceph. Clin. Neurophysiol. 24:242-248.

2. Clark, D. L., and B. S. Rosner. 1973. Neurophysiologic effects of general anesthetics: I. The electroencephalogram and sensory evoked responses in man. Anesthesiology 38(6):564-582.

3. Tinker, J. H., F. W. Sharbrough, and J. D. Michenfelder. 1977. Anterior shift of the dominant EEG rhythm during anesthesia in the java monkey: correlation with anesthetic potency. Anesthesiology 46(4):252-259.

   Yanagida, H. 1978. EEG changes during anesthesia. Anesthesiology 48(3):229-230. (Letter of correspondence for above article)

4. Luque, J., D. L. Sherrill, R. H. Jones, G. D. Swanson, and R. W. Virtue. 1979. Anesthetic depth transition by EEG analysis. Anesthesiology 51(3):S2.

5. Harter, M. R. 1967. Effects of carbon dioxide on the alpha frequency and reaction time in humans. Electroenceph. Clin. Neurophysiol. 23:561-563.

6. Darbinyan, T. M., K. Y. Bogdanov, and S. I. Plehotkina. 1977. Synaptic transmission in different neuronal pathways and general anesthesia. Anaesthesiology: Proceedings of the VI world congress of anaesthesiology, Mexico City, April 24-30, 1976. Enrique Hulsz, Jose Antonio Sanchez-Hernandez, Guillermo Vasconcelos, J. N. Lunn, eds. Excerpta Medica, Amsterdam-Oxford. pp. 60-66.

7. Darbinyan, T. M., and K. Y. Bogdanov. 1977. Comparative significance of evoked responses, high frequency EEG and mean frequency EEG during general anesthesia. Anaesthesiology: Proceedings of the VI world congress of anaesthesiology, Mexico City, April 24-30, 1976. Enrique Hulsz, Jose Antonio Sanchez-Hernandez, Guillermo Vasconcelos, J. N. Lunn, eds. Excerpta Medica, Amsterdam-Oxford. pp. 77-81.

8. Takeshita, H., T. Sakabe, and T. Kuramoto. 1977. Cerebral oxygen consumption studies related to the electroencephalogram during anesthesia. Anaesthesiology: Proceedings of the VI world congress of anaesthesiology, Mexico City, April 24-30, 1976. Enrique Hulsz, Jose Antonio Sanchez-Hernandez, Guillermo Vasconcelos, J. N. Lunn, eds. Excerpta Medica, Amsterdam-Oxford. pp. 90-97.

9. Smith, N. T., I. J. Rampil, F. J. Sasse, B. H. Hoff, and D. C. Flemming. 1979. EEG during rapidly changing halothane or enflurane. Anesthesiology 51(3):S4. (Abstr.)

10. Raitta, C., U. Karhunen, A. M. Seppalainen, and Marja Naukkarinen. 1979. Changes in the electroretinogram and visual evoked potentials during general anaesthesia. Albrecht V. Graefes Arch. Klin. Exp. Ophthal. 211:139-144.

11. Gerritsen, B. G. 1971. The effect of anaesthetics on the electroretinogram and the visually evoked response in the rabbit. Documenta Opthalmologica 29(2):289-330.

12. Uhl, R. R., K. C. Squires, D. L. Bruce, and A. Starr. 1979. The visual evoked response--index of anesthetic potency? Anesthesiology 51(3):S18. (Abstr.)

13. Johnson, N. F., T. M. Wilson, and R. Strang. 1973. Retinal fine structure after long-term anaesthesia. Exp. Eye Res. 15:127-139.

14. Sasa, M., Y. Nakai, and S. Takaori. 1967. Effects of volatile anesthetics on the evoked potentials and unitary discharges in the central auditory system caused by click stimuli in cats. Jpn. J. Pharmacol. 17:364-380.

15. Celesia, G. G., and F. Puletti. 1971. Auditory input to the human cortex during states of drowsiness and surgical anesthesia. Electroenceph. Clin. Neurophysiol. 31:603-609.

16. Kitahata, L. M. 1968. The effects of halothane and methoxyflurane on recovery cycles of click-evoked potentials from the auditory cortex of the cat. Anesthesiology 29(3):523-532.

17. Beecher, H. K., F. K. McDonough, and A. Forbes. 1938. Effects of blood pressure changes on cortical potentials during anesthesia. J. Neurophysiol. 1:324.

18. Smith, A. L. 1973. Dependence of cerebral venous oxygen tension on anesthetic depth. Anesthesiology 39(3):291-298.

19. Weiskopf, R. B., L. W. Raymond, and J. W. Severinghaus. 1974. Effects of halothane on canine respiratory responses to hypoxia with and without hypercarbia. Anesthesiology 41(4):350-360.

20. Knill, R. L., and A. W. Gelb. 1978. Ventilatory responses to hypoxia and hypercapnia during halothane sedation and anesthesia in man. Anesthesiology 49(4):244-251.

21. Knill, R. L., J. L. Clement, and A. W. Gelb. 1978. Ventilatory responses mediated by peripheral chemoreceptors in anesthetized man; regulation of respiration during sleep and anesthesia. Advances Exp. Med. Biol. 99:67-77.

22. Duffin J., A. Triscott, and J. G. Whitwam. 1976. The effect of halothane and thiopentone on ventilatory responses mediated by the peripheral chemoreceptors in man. Br. J. Anaesth. 48:975-981.

23. Whitesell, R., C. Asiddao, D. Gollman, and J. Jablonski. 1981. Relationship between arterial and peak expired carbon dioxide pressure during anesthesia and factors influencing the difference. Anesthesia and Analgesia 60:508-512.

24. Derenne, J. P., J. Couture, S. Iscoe, W. A. Whitelaw, and J. Milic-Emili. 1976. Occlusion pressures in men rebreathing CO_2 under methoxyflurane anesthesia. J. Applied Physiology 40(5):805-814.

25. Westenskow, D. R., W. S. Jordan, R. Jordan, and S. T. Gillmor. 1977. Evaluation of oxygen monitors for use during anesthesia. Anesthesia and Analgesia 60:53-56.

26. Bates, M. L., A. Feingold, and M. I. Gold. 1975. The effect of anesthetics on an in-vivo oxygen electrode. A.J.C.P. 64:448-451.

27. Dent, J. G., and K. J. Netter. 1976. Errors in oxygen tension measurements caused by halothane. Br. J. Anaesth. 48:195-197.

28. Maekawa T., Y. Okuda, and D. G. McDowall. 1980. Effect of low concentrations of halothane on the oxygen electrode. Br. J. Anaesth. 52:585-587.

29. Brooks, W. N., C. E. W. Hahn, P. Foex, P. Maynard, and W. J. Albery. 1980. On-line pO_2 and pN_2O analysis with an in vivo catheter electrode. Br. J. Anaesth. 52:715-721.

30. Eberhard P., W. Fehlmann, and W. Mindt. 1979. An electrochemical sensor for continuous intravascular oxygen monitoring. Biotelemetry Patient Monitg. 6:16-31.

31. Spooner, R. B., ed. 1977. Hospital Instrumentation Care and Servicing for Critical Care Units. Instrument Society of America, Pittsburgh, Pennsylvania. p. 32.

32. Eger II, E. I., and S. H. Bahlman. 1971. Is the end-tidal anesthetic partial pressure an accurate measure of the arterial anesthetic partial pressure? Anesthesiology 35(3):301-303.

33. Hoaglin, D. C., Frederick Mosteller, and John W. Turkey, eds. 1983. Understanding Robust and Exploratory Data Analysis. John Wiley & Sons, Inc., New York.

34. Johnson, L. W., and R. D. Riess. 1982. Numerical Analysis. 2nd ed. Addison-Wesley Publishing Company, Reading, Massachusetts.

35. Fukui, Y., N. T. Smith, and R. A. Fleming. 1982. Digital and sampled-data control of arterial blood pressure during halothane anesthesia. Anesthesia and Analgesia 61(12):1010-1015.

36. Pritsker, A. 1974. The GASP IV Simulation Language. John Wiley & Sons, Inc., New York.

* * * * * * *