

BOXER: An Interactive Analysis System for  
Cardiovascular Physiology Data

Jesse W. Hartley  
Dalton Research Center  
University of Missouri-Columbia  
Columbia, MO 65211

## INTRODUCTION

The data gathered during a cardiovascular physiology experiment consists of multiple channels of time varying signals, including pressures recorded at various points inside the heart and circulatory system, and blood flow in various arteries supplying the heart. During an experiment, the raw data is recorded on an analog FM magnetic tape recorder. Later, the analog tape is replayed onto a multi-channel chart recorder for detailed analysis. Traditionally, additional derived variables such as derivatives, products, and integrals of various channels would be computed using analog hardware. A typical experiment will require 8 minutes of 5 channels of raw data and the computation of 7 additional derived variables. The bandwidth of each channel is less than 100 Hz. The presentation of the data in graphical form is crucial.

The purpose of this analysis system is to enhance the previous analog instrumentation used for data analysis with a digital system. Initial acquisition of the analog data is done by the program TODISK. The core program, BOXER, displays waveforms and allows the user to select points from those waveforms. This automates counting boxes on chart paper and recording the values. The computation of derived variables is done by the program COMPUTE. These programs have the capability of handling up to 32 channels of data. The programs are written in Fig-Forth and assembly language.

## DATA ACQUISITION

The data is digitized using an ITC Model 9279 12 bit, analog to digital converter (A/DC) with 8 channel multiplexer that directly drives a digital 9-track tape recorder. In this application, each channel is sampled at 200 samples/sec. The digital tapes are transferred to a DEC PDP-11/34A with VT-125 graphics terminal for subsequent processing. Using TODISK, the calibration information for each channel is displayed and extracted from the tape and the experiment data is transferred to disk as a standard named RSX-11M disk file. The data display and menu presentations are identical to those of BOXER and will be discussed later.

## FILE FORMAT

The file format consists of a header followed by the sampled data. The header contains a channel calibration table, and a time calibration table for the scaling of data in the user's units. For the user's convenience, the header contains text strings for an experiment title, channel names and units.

Each channel calibration table contains the origin and range for each of the three coordinate systems used; the A/DC, video and user data spaces. The A/DC produces numbers between +/-2048. The video display uses values from 0 to 480. The user will specify his own values to correspond to the A/DC and video values. Conversion from

the A/DC data space to the user data space is accomplished with the following formula:

$$\text{UserValue} = \left[ \frac{(\text{SampledValue} - S_{\text{offset}})}{S_{\text{range}}} * U_{\text{range}} \right] - U_{\text{offset}}$$

where 'SampledValue' is the value in the A/DC data space, and 'UserValue' the corresponding value in the users data space. The 'offset' subscripts denote a 'zero' value in the data space and the 'range' subscript is the scale factor for conversion. A similar equation converts the A/DC data to the video data space.

The time calibration table contains the sampling rate, the origin of the left side of the display screen and the scale factor of the screen in both units of the sample number and in the user's units. Using an equation similar to that previously discussed, we can convert from time in seconds as specified by the user to the sample number from the beginning of the file. Using the video origin and range the sample number can be converted to a screen location.

The sampled data is stored in integer form to minimize storage requirements in the format as shown in Figure 1.

```
Channel-1(t),Channel-2(t),....Channel-N(t),Channel-1(t+1),...
```

Figure 1. Data Format

#### COMPUTING NEW VARIABLES

The computation of new variables by the program COMPUTE is directed by a user-written script from a limited set of Forth words. This script specifies how each channel of the output file will be

```
2 #OUTPUT-CHANNELS
: OUTPUT-1 INPUT-1 ;
" Left Vent. Press." OUT-TITLE-1
" Torr" OUT-UNITS-1

: OUTPUT-2 INPUT-1 INPUT-2 * ;
" Power" OUT-TITLE-2
" Torr*mm/sec" OUT-UNITS-2

END
```

Figure 2. Example Computation Script

computed along with text for the name and units of the channel. Figure 2 is an example of a script to copy one channel from the input file to the output file and compute the second output channel as the product of two input channels. The words, INPUT-x, are predefined to fetch the value of the input channel at the current time. Each output channel definition leaves one floating point number on the stack. The output definitions are put into an execution table to avoid scanning the dictionary during execution. COMPUTE cycles through the channel definitions in turn for each input file sample, rescaling the floating point numbers into integers and writing them to the output file. These calculations are done in floating point because the user's range of numbers is not

predictable and the user's notion of where 'zero' is necessary for the product function.

The derivative function is a 11 pole finite-impulse-response filter. This type of filter has no phase distortion, but requires input samples in advance of the output point being calculated. For this reason, only channels from the input file may serve as arguments. This filter was designed using the EQFIR program, and has a pass band up to 60 Hz, a transition band from 60 Hz to 99 Hz, and a stop band from 99 Hz to 100 Hz.

COMPUTE determines the output calibrations for each channel by loading the user script into two vocabularies. The first vocabulary creates the user offset calibration table entry for each channel. The second vocabulary creates the user range entry. In the example of Figure 1, 'OUTPUT-1' is defined as input channel 1. During calibration, the user script is executed from the 'offset calibration' vocabulary, and 'INPUT-1' fetches the user offset calibration value for channel 1 from the input file. This value is passed through the user specified function to arrive at the user offset value for the output file. The range calibration is determined by passing an upper calibration point through the user specified function, then recomputing the output range.

The functions are redefined to facilitate calibration. Addition and subtraction functions do not act on both calibration values but rather use the first value specified. This is done so that the two channels that are subtracted don't have a calibration scale of 'zero'. Scaling division is also a problem. For applications where I have used the division function, it has been adequate to define the upper calibration point as value of the first channel divided by 1/3 of the second channel.

#### MENU SYSTEM

The user is presented with a menu interface for the specification of file names, channel selection, and channel positioning in all three programs. This menu system is table driven, and re-entrant. The menu generator table specifies title text for the top of the menu screen, text for each item of the menu, text for an answer after each item, and an execute address for the action word of each menu item. On the selection of an item, the action word receives the text typed by the user. The action word may modify the answer text in the menu generation table, or the entire screen. The menu system will regenerate the menu display as necessary upon return from the action word.

#### POINT SELECTION & DATA DISPLAY

The primary operations that must be efficiently handled by BOXER are to allow the user to locate a point on a waveform, such as those in Figure 3, and store the value as a text string in a separate file for statistical analysis. In many instances, points from several different waveforms need to be taken at the same point in time. These operations also need to be done with a minimum of keystrokes.

This is done in BOXER using a table driven keystroke decoder. The keypad of the VT-125 terminal generates escape sequences that are decoded to direct execution of a Forth word for each key. Through the cursor keys, the user controls the position of a graphics cursor that is constrained to follow the waveform.

Moving up or down jumps the graphics cursor to the next waveform, keeping the same time position. Constrained movement is more appropriate here than unconstrained movement typical of mouse cursors. Keystrokes for moving the graphics cursor are generated at very high speeds so keystroke words are optimized for speed. Separate keys are encoded for the entry of a point into the text buffer, the flushing of the text buffer to the output file, and the integration between specified limits. The text output file is compatible with most statistics packages.

The generation of the graphical display is done using the asynchronous I/O of the operating system, RSX-11M, for speed. Time-critical functions are written in assembly language.

## CONCLUSIONS

This system has been in use for over two years and is in a state of continual growth. Speed, both in computation of new variables and in generating graphics is a continuing problem and is primarily due to the particular hardware used. A Commodore Amiga system for the graphic display is currently under development. Data storage is also a problem, though it is much the nature of the types of experiments done. A typical user will transiently use 40 to 50 Mbytes of disk space. Archival storage on 9-track has proven satisfactory.

The general construction of this system has been very flexible. Most of the code is shared between these three programs and other more specialized ones not described here. This ability to tailor the software to the task is extremely important in the research situation.

## REFERENCES

1. Programs for Digital Signal Processing, IEEE Press, New York, 1979, pp. 5.1-1 - 5.1-13.

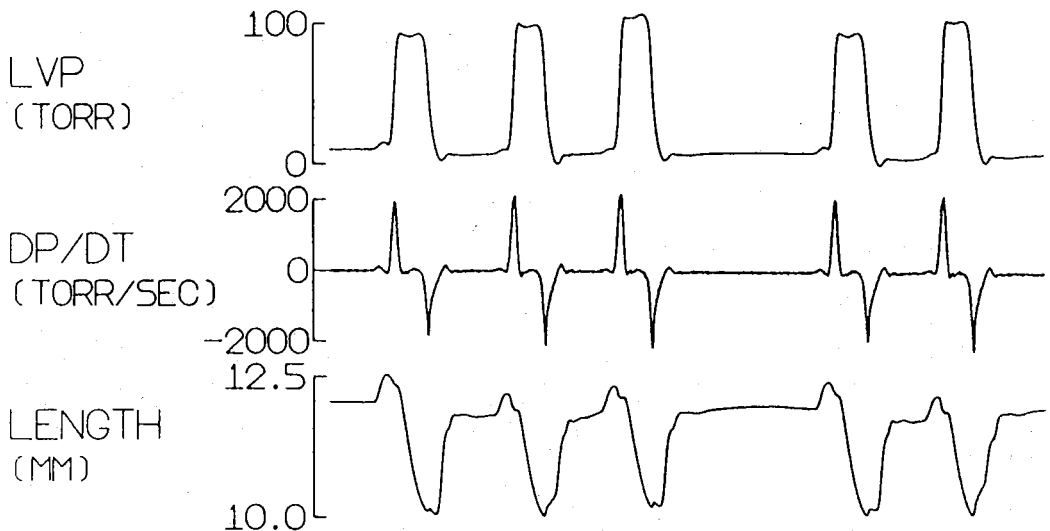


Figure 3. Typical Waveforms