System for Remote Multi-Channel Real-Time Monitoring of ECG via the Internet

M Oefinger[1], GB Moody[2], M Krieger[3], RG Mark[2]

[1]MIT Department of Electrical Engineering & Computer Science, Cambridge MA, USA
[2]Harvard-MIT Division of Health Sciences & Technology, Cambridge MA, USA
[3]MIT Department of Biology, Cambridge MA, USA

Abstract

Continuous physiological data monitoring of subjects has heretofore required the researcher to be in immediate proximity to the subject. Such 'bedside monitoring' is impractical in long-term studies, as often the subject and research facilities are in geographically disparate locations. Another limitation of existing technologies prohibits long-term data archiving without substantial user intervention. We constructed a novel hardware and software platform allowing us to visualize, via the Internet, encrypted real-time ECG signals from mice used in preclinical cardiovascular studies. As the design is highly modular, it can easily be adapted for use on different subjects and/or different signals of interest. The hardware consists of analog signal-conditioning PCBs (schematics on Physionet) [1] and an A/D board with open-source APIs for digitizing data. In its current incarnation, the system simultaneously monitors ECGs of 16 mice with implanted electrodes, streaming the data to the network for remote visualization and to a RAID partition on the acquisition server for archiving purposes. As the mouse heart rate is roughly 10 times that of a human, the processing power and network bandwidth required of the system is equivalent to that needed to monitor 160 humans simultaneously. The system can collect data on multiple mice, with no user intervention, for months at a time. Data acquisition is controlled through an intuitive web interface that permits simple starting and stopping of acquisition while requiring the user to enter data into a customizable HTML record that is written to an SQL database for accurate record-keeping. In the past 8 months we have collected and analyzed more data, with far less effort, than the formerly-used commercial system allowed in over two years. The remote visualization capability of our system has provided us unprecedented intuition and insight into long-term trends of cardiovascular disease of pre-clinical subjects. The database-driven web record entry has provided a paperless environment, replacing extensive manual logbook entries that once prevailed prior to long-term data collection capabilities.

1. Introduction

The foundation of any physiological study is a system for collecting and organizing data. One may often take for granted the underlying data capture mechanism in an algorithmic study or other pure data collation effort. However, in our recent preclinical studies utilizing mouse ECG, we encountered roadblocks imposed by existing commercial systems that prohibited simple data collection for long-term, high-throughput studies. Beginning with the design of a high-fidelity front-end analog instrument for signal conditioning and ending with fine-tuning for network security and optimization of server performance, we considered many details in the data pipeline to maximize the system's utility and minimize the threat of data loss. Our fundamental design goals accounted for two perspectives: that of the clinician or technician who would operate the data collection server and that of the researcher or collaborator who would interpret the vast quantities of data. The perspective of the technician mandated that the data acquisition server provide an intuitive interface for beginning and ending data collection with no need for intervention between. The researcher's perspective required that the data be easily accessible via the web, automatically analyzed (to the extent possible) with no third-party software, and visualizable in real-time via the web to enable interventional studies. Using our analog hardware design and eventually employing a parallel computing environment to realize more complex features of our goals, we designed and implemented a system that has brought to our studies the liberty we had hoped we might take from the beginning: that of knowing that a robust data collection and collation tool lies at the foundation of our data-intensive research efforts.

2. Methods

2.1. Signal conditioning hardware

The resting mouse heart rate is roughly 600 beats per minute and 99.9% of its spectral energy resides in the 0-

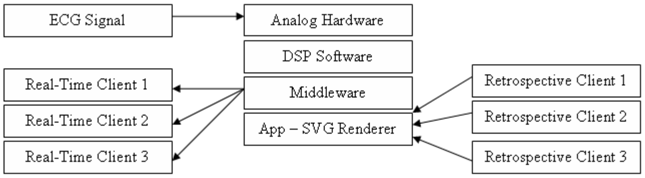

Figure 1. The system consists of hardware and software that permits simple collection of, and real-time and retrospective visualization of, large quantities of data.

1kHz range [2]. We therefore began our hardware design with thorough consideration of the merits of different filter types. Given the fairly wide spectral content of mouse ECG and the fact that important clinical events occur both at very low frequencies (i.e. ST segment depression) and very high frequencies (i.e. fine details of QRS shape), we sought to preserve the relative time ordering of different spectral content in an optimal sense. That is, we wanted to minimize phase distortion. We therefore selected Bessel filter types, which have a “maximally flat” group delay plot, or said another way they minimize phase distortion with the near-linear phase. The penalty incurred for this near-constant group delay comes in the form of a slower rolloff in the transition region than one would see for a Butterworth, Chebychev, or other common filter types. There are two means of compensating for a slow rolloff filter type: increase the filter order or perform oversampling in the digital domain. We utilized both techniques for maximum efficacy (section 2.2 will discuss the latter). We used 8th-order Bessel lowpass and high-pass filters to create a bandpass, and with the goal of modularity in mind we had these filters custom-built and packaged (details on Physionet) [1] as individual chips to be incorporated into our printed circuit board (PCB). Utilizing the same footprint but different cutoff frequency or filter-type characteristics one could easily adapt the boards for use on different subjects or for different physiological signals of interest.

Of additional concern for our neonatal murine subjects, in which the electrodes are implanted subcutaneously and therefore have minimal contact resistance, was a micro-shock scenario. Given the size of the subjects (approximately 12g) micro-shock currents on the order of tens of microamps are sufficient to cause fibrillation [3]. Our design uses an optical isolation amplification stage incorporating an extremely high dielectric to mitigate potential leakage currents through the leads. We additionally use DC/DC isolation between the input and output stages of circuitry to prevent any circuit-damaging voltage spikes at the output stage from coupling into the expensive server DSP card.

Schematics and PCB layout information for our design are available on Physionet [1].

2.2. Digital signal processing software

Using Microstar’s[4] model DAP4200A PCI data acquisition board on our main server, we are able to collect a single lead of ECG data from each of 16 subjects simultaneously. With the analog bandpass set at 0-1kHz, we require a sampling frequency of at least 2kHz per the Nyquist criterion. However our analog filters, in spite of their very high order, have a finite rolloff that could lead to aliasing if we do not sample a fair amount faster than the Nyquist rate. We therefore oversample each channel at 10kHz (5x oversampling) then use a 256-tap symmetric FIR low-pass filter followed by 5x decimation to achieve an alias-free signal sampled at the Nyquist rate. Given that our use of Bessel filters in the analog circuitry was motivated by a desire to minimize phase delay, it was important that we use an FIR symmetric filter, which is inherently endowed with perfectly linear phase, in our digital signal processing scheme.

2.3. Server daemons for data management and network streaming

Several server daemons are constantly active in the background managing data. As data samples become available on the DSP card, these daemons are collectively responsible for streaming the data to a RAID partition on disk and invoking an ’on-the-fly’ QRS detection algorithm. QRS detection is performed using *wqrs* from Physionet [1].

Other daemons are involved with buffering several hundred milliseconds of data at a time, encapsulating it in network packets, and streaming the packets to the server’s network interface via multicast for a connection-less real-time communication between the server and an unlimited number of clients. We utilize a connection-less multicast networking approach because the alternative, that of serving a stream on a per-client request basis (unicast), would occupy substantially more network bandwidth and server I/O resources. Because a multicast (broadcast) stream requires special router configurations and tight control over the TTL (time-to-live) field of UDP packets, the solution is not generally applicable to all network environments.

Of most notable concern is the expense of owning a ’true’ Class D IP address that would allow a global multicast to stream beyond the walls of our institution. To enable pseudo-multicast communication to select collaboration sites we have recently added an additional layer in the network architecture. This layer utilizes an application server on the same subnet as the data acquisition server, with both behind a firewall that blocks multicast traffic entirely. Local authorized users at our insitution log into the application server to view the real-time ECG. This same application server securely transmits the streams (via unicast SSL connection) to gateway machines at our collaborators' institutions, where they have the choice of utilizing the gateway as an application server or as a multicast transmitter for their local network.

2.4. Clinician user interface

The details of low-level digital signal processing need not burden the clinician or technician charged with collecting physiological data for studies. With simplicity in mind, we designed an intuitive GUI (graphical user interface) in HTML that controls which channels of the DSP card are actively collecting data, processing it, and streaming raw data and analysis results to disk. We expressly worked with HTML in our GUI design because the power of CGI (Common Gateway Interface) coupled with HTML allows for a robust interaction among HTML forms and back-end SQL databases, providing the power for a dynamic, data-driven interface.

If an authorized user chooses to begin acquisition, he will be provided with a list of available channels (i.e. those that are not currently on) along with a form to enter information relevant to the study. This form is completely dynamic and is configured by an authorized user. Consider, for example, a preclinical study in which genotype, diet and gender are important fields. An authorized user, using simple HTML forms to add or delete desired field entries, creates corresponding fields in the SQL database. All active fields in the database then become requisite for the onset of data acquisition. That is, when the clinician chooses to begin data collection on a channel, a dynamically generated HTML form with client-side JavaScript data verification routines is presented to the user. This HTML form requests that the user enter the channel number to use for collection and relevant subject parameters, which are checked for proper formatting prior to the creation of a new record in the SQL database and the onset of data collection on the requested channel. The user may end data acquisition with equal ease.

2.5. Researcher user interfaces

2.5.1 Real-time ECG visualization

The real-time visualization tool allows a user to select from any one of the 16 available channels to view the real time stream. The user may additionally change the amplitude or time scaling of the signal and toggle the grid and baseline graphics on and off. An additional indicator panel shows each channel's status (on or off) as well as the genotype, diet, elapsed time, and elapsed age of the currently selected subject. This web-accessible real-time stream is particularly useful to collaborators with no physical access to the animal facilities housing live subjects.

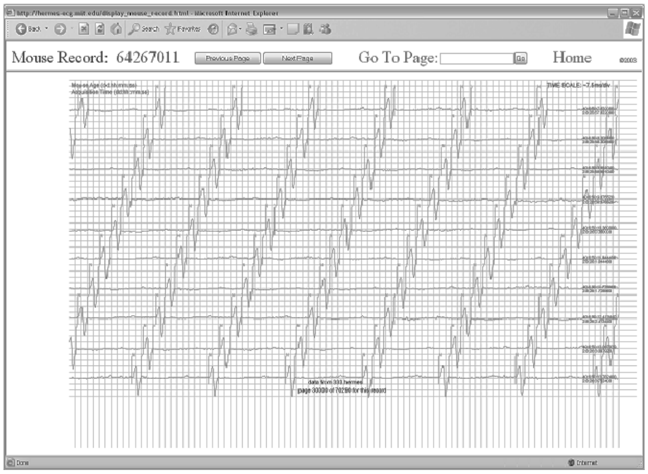

Figure 2. A sample screen shot of retrospective visualization. Using a web browser with an SVG plug-in, the user may visualize ECG records 1.) by moving forward or backward one page with the 'previous' and 'next' buttons or 2.) by entering a specific page number to view.

2.5.2 Retrospective ECG visualization

While certainly an authorized user with access to the archived data can download and format the data for visualization, such a process is cumbersome. Our system allows users to peruse extremely long ECG charts (our longest chart has over 5 billion samples, representing more than 30 days of continuous ECG). The disk space required for encoding data slices in .jpg, .gif or another traditional graphics format would far exceed the disk space occupied by the raw binary waveform data! We therefore use back-end CGI engines to render annotated ECG plots from the raw data in SVG (Scalable Vector Graphics) format. SVG is a graphics subset of XML that a browser with the appropriate plug-in may render. Because multiple users may look at separate records simultaneously and completely asynchronously, the complexities of requiring that the server maintain state are far too great. Nonetheless it is important that the user have a feeling of state; for example, a user should be able to click on a 'previous' or 'next' button to view the corresponding data slice. The server's CGI engines wrap the SVG-encoded images in an HTML file with state variables that the client browser can store using JavaScript. Thus the user's computer maintains the state, relieving the server of this burden. A sample screen shot is shown in figure 2.

2.6. Parallel computing environment for high throughput analysis

With high data throughput requirements and our need to perform 'on-the-fly' QRS detection to coalesce an otherwise disjoint process comes an extraordinary burden on the server CPU and I/O controllers. As we have continued to add more algorithms the demand on the main server has exceeded its capabilities, as evidenced by several main server crashes early in our algorithm development cycle. We therefore decided to provide additional servers operating in parallel with the main server to test and run algorithms. Using a parallel computer architecture in which the majority of the computational burden rests on the second (non-acquisition) server reduces the probability of overloading and crashing the main, "mission-critical" server.

3. Results

In our original efforts we used a commercial system for the acquisition of data, which required a team of personnel and approximately two years of effort to acquire, convert, format and plot multiple 8-12 hour segments of data into coherent long-term trend plots. Using this work flow we acquired and converted approximately 20 records ranging in length from several days to five weeks.

Using our novel system we acquired approximately 30 continuously-recorded ECG records, ranging in length from one to four weeks, over the course of 9 months. Data acquisition was contiguous, requiring no user intervention for the entire duration of collection. QRS detection and annotation were performed 'on-the-fly' and automatically by the acquisition server, and time-averaged long term heart rate averaging and other algorithms were performed automatically in our parallel computing environment. Using our real-time web visualization tool as well as retrospective web-based SVG plots that are created dynamically by CGI scripts, we have begun to shift the emphasis of our lab efforts toward interpreting data and away from manual efforts associated with its collection and collation.

4. Discussion

The system we developed serves as a foundation for preclinical studies in MIT's Department of Biology by providing a high-throughput data management and visualization platform that is quite intuitive. Future efforts will be spent on improvements to the existing visualization tools by porting them to Java. Preliminary efforts in this direction have led us to believe that we can create a much more powerful interface with similar functionality to existing Holter software. We will additionally use the existing system as a foundation for real-time algorithm development, specifically ST segment and arrhythmia analysis. With the existing software hooks available for instantaneous emailing upon server scripting errors, we will incorporate a notification system to alert the technician 'on-call' at the time to events of interest. These events of interest, namely arrhythmias and significant ST segment changes that could indicate infarction or ischemia, when detected in real time, will provide our laboratory with unprecedented interventional research capabilities.

5. Conclusion

Preclinical and clinical studies alike require a robust system for the collection, visualization, and analysis of physiological data. Long-term, high-throughput studies have, to date, required a manually-intensive work flow for data acquisition, formatting, collation, and visualization, most often relying on multiple serial processes and several software packages. Our system consolidates the traditional cumbersome work flow into an easy-to-use, automated parallel computing platform with fault-tolerant RAID data storage, SQL subject record management, and real-time and retrospective visualization capabilities via the Internet. While this system is currently in use for preclinical studies on mice, it can easily be adapted for use in any long-term data-logging application.

References

[1] http://www.physionet.org/.

[2] Ai H, Cui X, Tang L, Zhu W, Ning X, Yang X. Studies on the time domain and power spectrum of high frequency ecg in normal mice. Sheng Li Xue Bao Oct. 1996;512.

[3] Feinberg BN. Applied Clinical Engineering. Prentice-Hall, 1986. Pp. 459-465.

[4] http://www.mstarlabs.com/.

Address for correspondence:

Matthew B Oefinger
Laboratory for Computational Physiology
Harvard-MIT Division of Health Sciences & Technology
Rm E25-505, 45 Carleton St.,
Cambridge MA 02142 USA
oefinger@mit.edu

This research was supported by NIH grant PO1HL66105.