Cursor Control Using EMG Signals and Morse Code

Grant Connell

4/21/2008

I. Introduction

Cursor control is easy for most computer users. One simply moves the mouse and the cursor moves in the desired direction. For others, cursor control is more difficult due to physical limitations. These individuals may have only speech, eye movement, or head movement as a source of control. Athough applications have been developed to aid these individuals in computer cursor control and text generation, many are usually expensive and are slow and cumbersome to use. This paper describes a project that uses facial EMG signals and Morse Code encoding to achieve efficient cursor control and also to provide for fast text generation independent of soft keyboards. Text generation is necessary for password login to many web sites and for file saving and retrieving.

Previous work on using EMG signals[1] showed that Morse code encoding of facial EMG signals was reliable. Hardware implementation was a simple sound card EEG unit with associated processing software. Although cursor control was good (using a template matching approach) there were several short comings:

1. The speed of the cursor was either too slow or too fast for a given setting. Either one could zero in on a target easily at slow speed or one could move across the screen at high speed but not land on a target easily.
2. Using a soft keyboard, text generation took a very long time. There was considerable time spent moving the cursor from letter to letter. There was also no way to "paste" in passwords into text fields when logging on to an application.
3. Swallowing would occasionally generate false characters that would have to be erased.
4. The epoch time for each input command was a minimum of one second. This allowed both long and short commands to be processed by the template matching software. However, the maximum character rate was limited to one per second.

To overcome these limitations it was decided to implement a continuous decoder that would interpret the input signal stream and generate letter commands for cursor control and text for input to other applications. Morse Code was selected as the coding scheme for the commands as it is one of the most efficient text codes implemented. This type of coding requires the minimum code sequences to generate the commands and text.

II. Equipment:

The hardware was a four channel AM sound card EEG unit (only one channel was used) coupled to the input of a 3.2 GHz laptop computer running WindowsXP, SP2. The electrodes were AgCl type and placed on either side of the face by the jaw muscles. Figure 1 shows the location of the electrodes. The DRL was connected to a wrist strap.

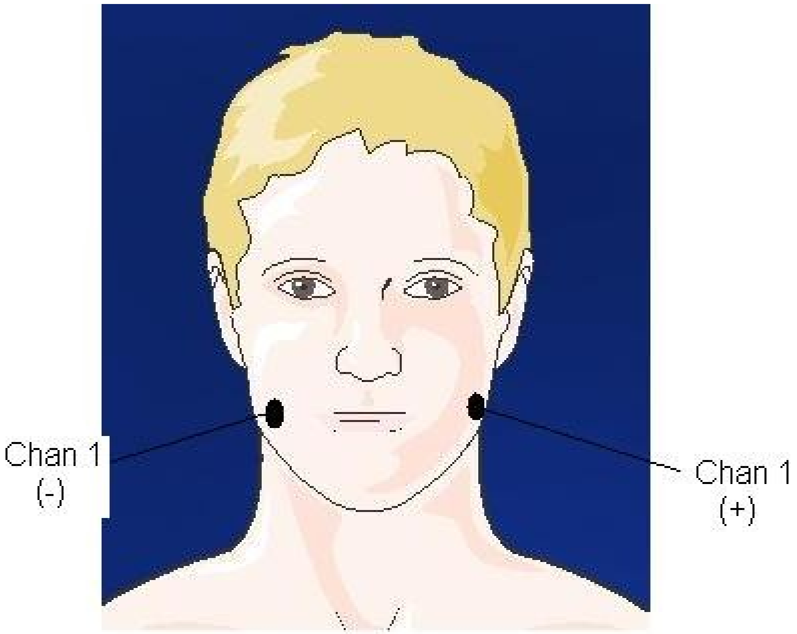


Figure 1, Electrode Placement for EMG Cursor Control

The software used was the NeuroProbe application (version 1.18 Beta) developed specifically for sound card EEG units. A processing design was created to pre-process the EMG signals before sending it to the Voice Cursor element. The design operated at a 1024 Hz sample rate with band pass filtering between 70 Hz and 500 Hz. Additional elements were added for observing the waveforms for debug and further analysis. Only three elements are actually needed, the EEG Input element, the Bandpass Filter element, and the Voice Cursor element. A diagram of the design used for testing is shown in figure 2.

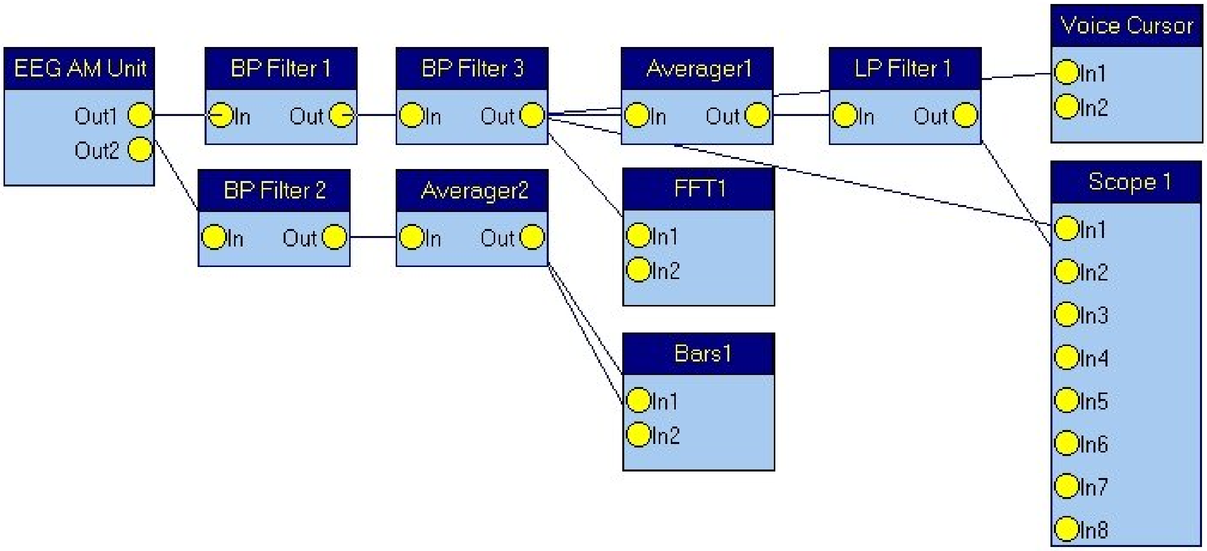
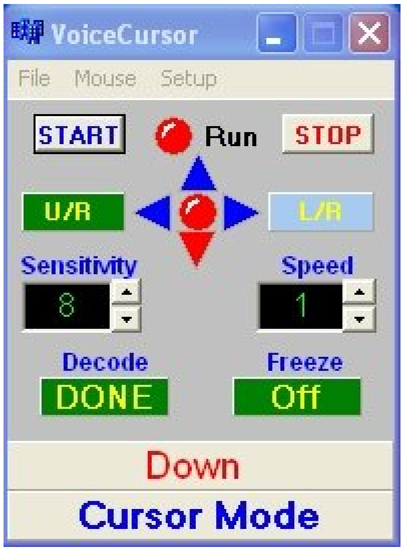


Figure 2, Facial EMG Design

The "Voice Cursor" element was modified to have three modes of operation; a general voice command input, a facial EMG input with template matching for cursor movement, and a facial EMG input using Morse encoding for both command and text input. The latter mode continuously processes the EMG signals rather than operating on a one-second data capture. The Voice Cursor screen has a secondary screen that allows template generation and also shows the input waveforms after processing. The Voice Cursor screen also shows the commands or text that have been decoded and whether the operation is in the command or text input modes. Figure 3 shows the modified Voice Cursor screen.



Sensitivity 8

Speed 1

Decode DONE

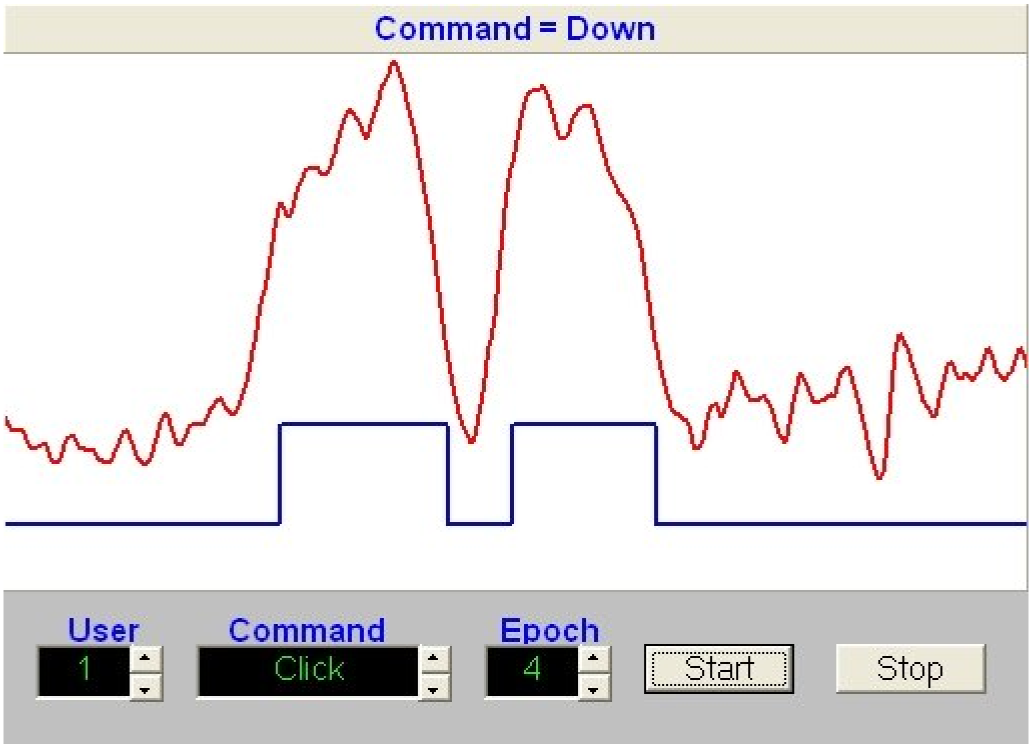
Freeze Off

Down

Cursor Mode

Figure 3, Voice Cursor Screen

The modified Vocabulary screen now shows the processed waveforms on a timeline. Previous displays only showed template and one second data captures. The Vocabulary screen is needed for user training to display a timeline to the user depicting the spacing of the EMG bit stream. By observing the spacing of the on and off signal duration, the user can modify EMG generation to conform to the required ones and zeros needed for decoding. Figure 4 shows the Vocabulary screen with processed data. The upper trace shows the EMG signal after applying DSP, the lower trace shows the bit stream to the decoder after threshold detection.


Command = Down

User 1

Command Click

Epoch 4

Figure 4, Vocabulary screen

III. Signal Processing:

The facial EMG signals are considered "noisy" as compared to other signals with good signal to noise ratios (SNR). These EMG signals, although strong, are comprised of many neurons firing very quickly. The resultant signal has great variations in amplitude and looks more like "shot noise". The digital signal processing has to level the amplitude of the signal and still provide for reliable threshold detection. Several different processing algorithms were tried with a final configuration, shown in figure 5, achieving the most reliable decoding.

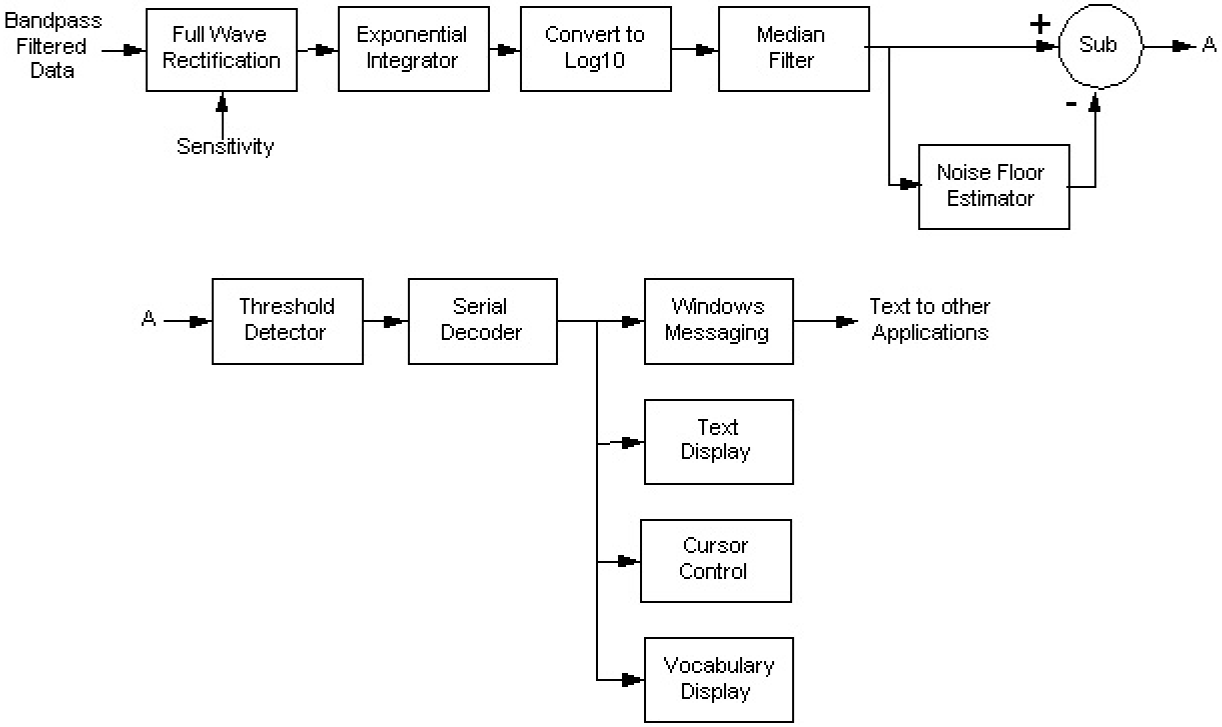


Figure 5, DSP of the EMG Signals

IV. Morse Code Implementation:

The software was designed to operate in one of two modes, a cursor control mode, and a text mode. The cursor control mode used short code sequences for the command codes. The shortest codes were used to stop the cursor movement. In the cursor mode of operation the following codes were used:

1. Up command = dit dit, letter "i"
2. Down command = dah-dah, letter "m"
3. Left command = dit-dah, letter "a"
4. Right command = dah-dit, letter "n"
5. Stop command = dit, letter "e" or dah, the letter "t"
6. Click command = dit-dit-dit, letter "s"
7. Double-click command = dah-dah-dah, letter "o"
8. Switch modes = dah-dah-dah-dah, no printable letter, switching between cursor and text modes
9. Freeze command = dit-dah-dit, letter"r".. used to freeze the decoding for 3 seconds (for swallowing)

In the text mode of operation, several text functions such as spaces, backspaces, and caps toggling were needed for efficient text input to other applications. The command codes for these functions were:

1. Space character = dit-dit-dah-dah, non-printable Morse character
2. Caps toggling = dit-dah-dit-dah, non-printable Morse character
3. Backspace = dit-dit-dah-dit-dit, non-printable Morse character
4. Switch modes = dah-dah-dah-dah, no printable letter, switching between cursor and text modes

To overcome the fast/slow cursor speed dilemma, a progressive speed control was implemented. When a command to move the cursor in a given direction was generated, the cursor speed was set to the slowest setting. A command to move the cursor in the same direction would increment the speed of the cursor. This sequence could be repeated up to 7x speed. A stop command would reset the speed to the slow setting again. Close in cursor movement would consist of a command sequence such as up… stop…down… stop… left…stop, etc. This allows precise movement of the cursor to a target location.

V. Testing:

Control of the cursor and text generation depended on EMG signal extraction and code generation. Many different DSP schemes were implemented with varying degrees of success. The most typical problem was multiple triggering on the rise and fall times of the dit and dah pulses. Although most of the problems were solved through careful signal conditioning and thresholding, some user training was necessary to insure a positive signal with faster on and off times.

The decoding algorithm was extracted from another application designed for amateur radio use. The algorithm worked well for code that was keyed by hand. After much testing it was determined that generation of the ones and zeros by the jaw EMG signals contained different timing elements than hand generated ones. The spacing between dits and dahs typically was longer and the dits were usually shorter than the 1 to 3 ratio between dits and dahs. Once recognized, the decoder algorithm was modified to be tolerant of the new spacing. Code generation became considerably easier and less frustrating. The algorithm was also modified to operate in a limited speed range compatible with jaw generated EMG signals. The speed would change dramatically during long signal up times when the user swallowed. A freeze command was also implemented when the cursor mode was enabled to limit the decoder speed pulling. After the letter "r" was decoded, the decoding was disable for three seconds.

Due to the "noisy" nature of the EMG signals and the extreme DSP filtering, the maximum code rate was about 10 words per minute (WPM). Using the cursor movement only and a soft keyboard, the text rate was about 2 WPM. This resulted in a worthwhile factor of 5 improvement in the text generation rate.

Learning to generate Morse code with facial EMG muscles had the same learning curve as learning to send Morse code by hand. Pulse duration timing and learning to send the correct code sequence improved with practice.

VI. Conclusions:

Continuous and reliable decoding of facial EMG signals was achievable. The use of the technique proved to be relatively easy (once the Morse code was learned) and was not very tiring even after a couple of hours of use.

Necessity is the mother if invention. Most of the special command codes such as back space, caps lock, etc. were added during use of the technique while web surfing. Without these commands, the soft keyboard would have been necessary to accomplish the same control.

The new timeline display in the Vocabulary screen was necessary for user training and to maintain correct bit steam timing. Instead, future designs of the sound card EEG units could have an audible tone to allow the user to hear the sequence. This has proven to be effective in learning to send Morse code by hand.

As in the previous study using EMG signals for voice decoding and facial decoding with templates, it would be desirable to use electrodes that did not require paste.

VII. References:

1. "Cursor Control Using Voice and Facial EMG Signals", by Grant Connell