신호 분석을 위한 시각 프로그래밍 시스템

*김형진, **박승훈, **우응제
*삼성종합기술원 의료기기 연구팀,**건국대학교 의과대학 의학공학과

A Visual System for Signal Analysis

*Hyung-Jin Kim, **Seung-Hun Park, **Eung-Je Woo
*Medical Electronics Team , SAIT, **Dept. of Biomedical Eng., Kon Kuk Univ.

Abstract

We present a visual programming system for signal analysis; it allows the user to construct signal processing algorithms by assembling visually fundamental signal processing blocks, and to observe the processed signals as well as the original signals in detail by magnifying a portion and measuring the time interval and amplitude between two points. Each fundamental signal processing block is implemented as an independent dynamically linked library module. Therefore, the user can expand the system processing capability by simply adding dynamically linked library modules without restructuring the entire system.

서 론

대부분의 신호 수집 및 처리 시스템은 응용 분야에 한정된 신호 처리 기법을 사용하여 실시간으로 신호를 처리하거나, 저장 매체나 종이에 기록한 다음 나중에 육안이나 컴퓨터를 사용하여 처리한다. 의료, 산업, 교육 등 각 분야에서 사용되는 이러한 신호 분석 시스템들은 특정 응용 분야에 적합하도록 개발되어, 사용자의 요구가 달라졌을 때, 요구에 적합하도록 수정하기가 힘들다. 또한, 그 시스템을 사용하기 위해서는 미리 많은 준비와 교육이 필요하다. 신호 처리를 담당하는 개발자들은 시스템을 설계할 때 문제점을 발견하기 위해 시간이 많이 드는 구현 과정을 거친 후에야 비로소 설계의 문제점을 발견할 수 있다.

특정한 신호 성분을 검출하는 알고리즘을 개발하거나, 어떤 알고리즘에서 적합한 계수들의 값을 찾아야 하는 경우, 시행 착오에 의한 방법을 많이 사용한다. 즉, 계수들을 바꾸어 결과를 본 후, 가장 좋은 계수의 조합을 선택한다. 이러한 시행 착오 방식은 많은 시간과 노력을 필요로 한다. 따라서, 이러한 작업을 손쉽게 하기 위한 소프트웨어적인 작업 환경에 대한 연구가 수행되어 왔다. 생체 신호의 관찰 및 알고리즘의 최적 계수들의 집합을 육안 관찰에 의해 찾을 수 있는 작업 환경에 대한 연구와 여러 채널의 뇌파를 관찰하고, 저장하며, 나중에 검색할 수 있는 시스템이 개발되었다[1, 2, 3]. 이러한 시스템들은 고정된 메뉴나 정해진 철자법에 의해 원하는 신호 처리 알고리즘을 표현하며, 변경이나 확장이 어렵다. 따라서, 사용자는 프로그래밍 하지 않는 대신 신호 처리 알고리즘 표현 언어를 별도로 배워야 한다.

최근 연구가 활성화 되고 있는 시각 프로그래밍 방법은 사용자가 프로그래밍 언어나 컴퓨터에 대한 지식 없이도 자신이 원하는 것을 손쉽게 나타낼 수 있는 융통성 있는 작업 환경을 제공한다. 본 연구에서는 이러한 시각 프로그래밍 기술을 신호 처리 및 분석 환경에 적용하여, 각종 신호 처리 알고리즘들을 그래픽 아이콘들을 간단히 조합하여 구현할 수 있고, 그 처리 결과를 즉시 관찰할 수 있는 신호 분석 환경을 개발하였다.

시각 프로그래밍을 이용한 신호 분석

어떤 신호를 수집하여 분석하기 위해서는 여러 종류의 작업을 수행해야 한다. 우선, 신호를 수집하기 위한 데이터 수집 장치의 제어 프로그램을 작성해야 하며, 수집된 데이터를 처리하기 위한 신호 처리 알고리즘을 설계해야 하며, 직접 프로그래밍하거나 라이브러리로 만들어진 코드를 이용하여 구현한 다음, 그 알고리즘의 성능을 시험해야 한다. 이러한 작업은 원하는 성능이 나올 때까지 반복된다. 컴퓨터 프로그래밍에 익숙치 못한 사용자라면 많은 시간을 투자하여, 먼저 프로그래밍 언어를 습득해야 할 것이다. 이렇듯, 신호 수집 및 처리를 위해서는 본래의 목적과는 상관없는 부수적인 작업에 더 많은 시간을 투자해야 한다.

시각 프로그래밍이란 정해진 철자법을 갖는 언어를 사용하지 않고, 사용자가 시각적인 직관으로 자신의 의도를 쉽게 표현할 수 있는 작업 방법이다. 따라서, 프로

그래밍 언어를 배우지 않고서도 원하는 작업을 쉽게 할 수 있다. 본 연구에서 개발한 Signal Path Designer (SPD)는 시각 프로그래밍 개념을 신호 처리 알고리즘 설계 작업에 적용한 것으로, 시스템에서 제공하고 있는 기본적인 신호 처리 요소들을 그래픽 편집기와 같은 방법으로 조합하여, 복잡한 알고리즘을 구성할 수 있게 한다. SPD는 개념적으로 각종 신호처리 요소들을 대상으로 한 작업 공간이라 할 수 있다. 사용자가 SPD에서 사용할 수 있는 신호처리 요소들은 신호 수집 요소, 신호 시각화 요소들, 각종 필터 , 미분기, 적분기, logic 처리 요소 등 모듈화 할 수 있는 여러 종류의 처리 요소들을 포함하고 있다.

SPD 내에서 신호처리 요소들은 작은 비트맵으로 표현되고, 각 신호처리 요소들 간의 연결은 마우스를 이용하여 두 비트맵 간에 선을 연결함으로써 이루어지며, 관련된 연결 정보는 해당 신호 처리 요소에 저장된다. 여러 개의 SPD를 사용하여, 동시에 여러 작업을 수행할 수도 있으며, 이미 완성된 설계를 편집할 수 있어서, 기존 알고리즘에 바탕을 둔 새로운 알고리즘 새로운 알고리즘을 쉽게 설계할 수 있다.

SPD에서 사용할 수 있는 시각화 요소는 신호의 출력 기능 뿐 아니라, 신호에서 추출한 사건 정보를 신호와 중첩하여 출력할 수 있는 기능을 구비하고 있다. 신호의 특정 부분을 확대하는 기능, 신호의 chart speed, sensitivity 등을 조절할 수 있는 기능들을 포함하고 있다. 신호의 특정 부분을 블록으로 지정하여 그 부분의 최대, 최소값과 시간 정보, peak-to-peak의 진폭 및 시간 간격 등을 측정할 수도 있다. 신호의 양이 많을 때에는 시간과 진폭을 이용하여 자동으로 검색할 수도 있다.

동적 링크를 이용한 기능의 확장

각종 신호 처리 요소들은 *DLL (Dynamic Link* Library) 형태로 독립적으로 구현할 수 있기 때문에 쉽게 기능을 확장할 수 있다. DLL은 주 프로그램과는 별도로 존재하는 일종의 실행 라이브러리로서, 주 프로그램이 실행되는 시점에 동적으로 연결되는 모듈이다. 많은 신호처리 알고리즘을 제공해야 하는 신호 분석 시스템인 경우에는 상황에 따라 필요한 DLL 들만을 메모리에 올려, 효율적으로 메모리를 관리할 수 있다. 또한, 특정 알고리즘을 수정하기 위해 전체 시스템을 다시 컴파일하고 링크해야 하는 불편을 없앨 수 있다.

SPD는 처음 실행될 때 각종 신호 처리 요소에 대한 정보를 정보 파일로부터 읽어, 해당 DLL들을 메모리로 올린다. 따라서, 사용자에 따라 다른 종류의 신호 처리 요소들을 한군데 모아 둠으로써, 각자에 적합한 작업 환경을 구축할 수 있다. 새로운 신호 처리 요소가 필요한 사용자는 본 시스템에서 정해 놓은 일정한 형식에 맞추어 자신의 알고리즘을 구현하여, 주 프로그램에 쉽게 등록할 수 있다.

신호와 관련된 사건 정보의 관리

신호와 관련한 사건 정보란 신호에 포함된, 진단에 중요한 근거가 될 수 있는 정보를 의미한다. 예를 들어, ECG에 존재하는 비정상적인 파형은 의사가 심장 질병을 진단하는데 중요한 실마리를 제공할 수 있다. 본 연구에서는 신호 처리 결과로 찾아낸 이러한 사건 정보를 관리할 수 있는 사건 정보 관리자를 구현하고 있다. 사건 정보 관리자는 특정한 시점의 사건 정보들을 시간 정보와 함께 저장하여, 나중에 신호를 화면에 출력할 때 중첩하여 출력함으로써 육안으로 검사할 수 있게 한다. 사용자는 마우스를 이용하여 관심 있는 특정 부분에 주석을 달 수 있으며, 이미 기록된 사건 정보를 삭제하거나 수정할 수 있다. 이러한 사건 정보들은 사용자가 원하는 경우 문서화할 수 있다. 수치정보, 메시지 정보 등 여러 종류의 사건 정보가 존재하기 때문에, 이들을 모두 표현할 수 있는 구조를 갖도록 하였다.

객체 지향 개념을 이용한 시스템의 구현

객체 지향 개념을 적용하여 구축한 시스템은 구성하고 있는 여러 객체들과 그 객체들 사이의 통신으로 나타낼 수 있다. 객체들 간의 통신은 메시지 교환을 통해 이루어진다. 본 연구에서 개발한 시스템의 전체 구조는 그림 1에 나타나 있다. Kernel은 각 객체들 간의 통신을 중계하고 작업 모드를 결정하며, 객체들의 동작을 제어한다. File Manager는 생체 신호의 Data File을 관리하며, Data Manager는 실행하는 중에 생체 신호와 사건 정보를 관리한다. 사용자가 SPD를 사용해 설계한 알고리즘을 실행시키면, Kernel 은 Signal Processing Manager로 하여금 Data Manager에 저장된 신호들에 알고리즘이 포함하고 있는 신호 처리 요소들을 규정된 작업 순서로 적용한다. 신호 처리의 결과는 Visualization Manager를 통해 Display Window에 출력되는데, 사용자가 정한 옵션에 따라 다양한 출력 형태를 갖는다. Visualization Manager 는 Kernel을 통해 데이터 베이스 정보를 출력하기도 하고 화면에 출력된 신호에 대하여 확대/축소, 신호의 검색, 블록 정보, 블록 복사 등 사용자가 원하는 특정 작업들을 수행하기도 한다.

결 론

본 연구에서는 사용자가 원하는 신호 처리 알고리즘을 손쉽게 설계할 수 있고, 그 성능을 여러 종류의 시각적인 방법으로 검증할 수 있는 신호 분석을 위한 시각 프로그래밍 시스템을 개발하였다. 알고리즘의 설계는 기본적인 신호 처리 요소들을 나타내는 그래픽 아이콘들을 선택하여 위치시킨 다음, 입력과 출력들을 연결하여 행한다. 처리된 결과는 구비된 다양한 측정 기능을 이용하여 육안으로 관찰하거나 정량적인 방법으로 검증할 수 있다. 동적인 링크를 통해 신호 처리 요소들을 실행하기 때문에, 전체 시스템을 수정하지 않고 신호 요소들을 독립적으로 추가하여 시스템의 기능을 확장할 수 있다. 관련된 신호 처리 요소들을 모아 관리함으로써 사용자의 용도에 맞는 작업 환경을 구축할 수 있다.

신호 처리 알고리즘들에 의해 검출된 여러 종류의 정보들을 데이터베이스화 하여 생체 신호와 같이 관리함으로써, 육안에 의한 검증이나 재검토를 신속히 수행할 수 있다.

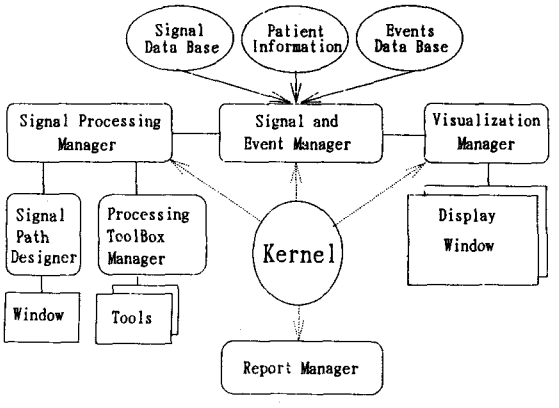

그림 1

전체 시스템의 구조

참 고 문 헌

[1] Seung-Hun Park, J. C. Principe, J. R. Smith, and Steven A. Reid, "TDAT - Time Domain Analysis Tool for EEG Analysis," IEEE Trans. on Biomed. Eng., vol. 37, pp. 803-811, 1990.

[2] Thomas F. Collura, Ernest C. Jacobs, David S. Braun, and Richard C. Burgess, "Eview - A Workstation-Based Viewer for Intensive Clinical Electroencephalography," IEEE Trans. on Biomed. Eng., vol. 40, no. 8, pp. 803 - 811, 1993.

[3] 김형진, 박승훈, 우응제, "SiMACS에서의 생체 신호 해석을 위한 Workstation,"의용 생체 공학회 춘계학술대회 논문집, 제16권, 제1호, pp. 60 -62, 1994.