論 文
52D-12-7

근 반응제거를 위한 적응필터 설계와 FPGA 구현

Design of Adaptive Filter for Muscle Response Suppression and FPGA Implementation

廉 晧 準* · 朴 榮 喆** · 尹 亨 老***
(Ho-Joon Yeom · Young-Cheol Park · Hyung-Ro Yoon)

Abstract - The surface EMG signal detected from voluntarily activated muscles can be used as a control signal for functional electrical stimulation. To use the voluntary EMG signal, it is necessary to eliminate the muscle response evoked by the electrical stimulation and enable to process the algorithm in real time. In this paper, we propose the Gram-Schmidt(GS) algorithm and implement it in FPGA(field programmable gate array). GS algorithm is efficient to eliminate periodic signals like muscle response, and is more stable and suitable to FPGA implementations than the conventional least-square approach, due to the systolic array structure.

Key Words : EMG controlled stimulator, Gram-Schmidt, muscle response, FPGA

1. 서 론

C5-C6 완전 척추손상(SCI)이나 불완전 척추손상환자들의 경우 한손 또는 양손의 기능을 상실하게 된다. 이러한 환자에게 손의 기능을 회복시키는 것은 일상생활에서 예를 들어 먹고 마시고 글을 쓰게 하는 등등의 일들을 가능하게 해주어 좀더 독립적인 삶을 살아갈 수 있게 한다. 대부분 척추손상환자의 경우 뇌 또는 척추 연결에 문제가 발생하더라도 말초신경이나 근육은 정상적으로 남아있다. 이러한 이유로 신경이 마비된 근육에 전기자극을 가함으로써 자발적인 운동기능을 부분적으로 회복시키는 것이 가능하며 이러한 기능적전기자극에 대한 연구가 지난 30여년 동안 이루어 졌다[1-7].

그중에서 최근에 관심이 되고 있는 것은 불완전 척추손상환자의 마비된 팔에 남아있는 미세한 근전도 신호를 검출하여 이 신호의 크기에 따라 전기자극의 강도를 조절하여 동일근육에 전기자극을 가함으로써 부드럽게 손을 쥐고 펴는 동작을 하게 하는 것이다[3]. 이러한 방식의 기능적 전기자극기을 근전도제어 전기자극기(EMG controlled stimulator)라고 하며 이 경우 근전도를 1초에 20번에서 30번정도 검출한 후에 그때마다 전기자극을 주어 환자로 하여금 정상인처럼 부드럽게 쥐고 펴는 동작을 하게 한다.

그러나 이 경우 크게 두 가지 문제점이 발행한다. 먼저 전기자극은 20V에서 90V까지, 또는 환자에 따라 더 높은 자극을 가해야 하는데 이러한 자극 전압 때문에 근전도 증폭단이 포화되어 적절한 측정을 방해하게 된다. 이때 전기자극에 의해 나타나는 포화전위를 자극잡음(stimulation artifact)이라고 한다. 그러나 이것은 그 기간이 짧기 때문에 blanking 회로나 감쇄회로로 차단하면 된다. 또 다른 문제점은 전기자극이 피부조직과 근육에서 반응하여 나타나는 근반응(muscle response, M-wave) 신호인데 이 신호의 크기는 자발적인 근전도에 비해 일반적으로 20dB 이상 크고 주파수대역이 자발적인 근전도와 겹치기 때문에 근전도 증폭단에서 검출되는 신호의 대부분을 차지하게 되고 일반적인 밴드제한필터로는 제거할 수 없다.

효과적인 근전도 제어 기능적전기자극을 위해서는 근반응에 섞여 있는 자발적인 근전도 신호만을 추출하기 위한 신호처리 기술을 필요로 한다. 이를 위해 사용할 수 있는 가장 간단한 방법이 콤(comb) 필터를 이용하는 것이다[3]. 이 방법은 근반응이 주기적인 자극에 의해 발생한 주기신호라는 점에 착안하여 신호의 특성이 장시간 일정하게 유지된다는 가정하에 이용되는 방법이다. 이 방법이 간단하기는 하나 근반응은 시간에 따라 그 크기나 형태가 일정하게 발생되는 신호가 아니므로 성능에는 상당한 한계를 가지고 있다. 이러한 문제를 해결하기 위해 사용된 방법이 적응 필터를 사용하여 변화하는 근반응의 특성을 적응적으로 추정하여 제거하는 방법이다[4][5][6]. 이중에서도 가장 적합한 적응 알고리듬이 바로 최소 자승법(LS: least squares)이다[6]. 왜냐하면 근반응의 특성을 보면 자극주기에 따라 비록 그 크기는 가변이 되지만 이전 자극주기와 상당히 높은 상호연관성(correlation)을 갖게 된다. 이에 비해 자발적인 근전도는 자극주기에 대해 낮은 연관성을 갖는다. 또한 자극이 시작되는 시점에서는 신호가 급격하게 변하는 이산적인 특성을 최소평균자승법(least mean square algorithm)과 같은 샘플처리(sample processing)로 계수를 업데이트하는 알고리듬

* 正 會 員 : 延世大 保健科學大 醫工學科 博士課程
** 正 會 員 : 延世大 保健科學大 醫工學科 敎授
*** 正 會 員 : 延世大 文理大 情報處理學部 助敎授
接受日字 : 2003年 9月 24日
最終完了 : 2003年 10月 21日

보다는 최소자승법과 같은 블록처리(block processing)로 계수를 업데이트하는 것이 계수가 빠르게 수렴할 뿐 아니라 수렴후에도 계수의 변화가 거의 없다. 그러나 최소자승법 적응알고리듬은 연산과정이 복잡하여 하드웨어 구현에는 적합하지 않다는 단점이 있다.

본 논문에서는 기존의 최소자승법알고리듬과 같은 성능을 가지면서 하드웨어 구현에 적합한 적응 필터링 기술을 이용하여 근반응을 제거하는 알고리듬으로 Gram-Schmidt 알고리듬을 제안한다. 또한 이 알고리듬이 하드웨어 구현에 용이한 것을 보이기 위해 근반응제거용 FPGA를 설계한다.

2. 본 론

2.1 근전도 제어 전기자극기

마비환자에게 손의 기능을 향상시키기 위한 다양한 기능적 전기자극기가 소개되어 왔다. 이러한 기능적 전기자극기에서 가장 관심이 되는 것은 자극을 시작하고 자극의 크기를 조절하는 제어신호에 있다. 손을 쥐기 위한 제어신호로서 현재 사용하고 있는 기능적 전기자극 시스템 중에 대표적인 것으로는 푸시버튼과 어깨위치센서 그리고 손목의 신전 굴절센서를 이용하는 것이 있다[7].

푸시버튼방법은 가장 단순하고 초기에 나온 것이다. 그러나 이것은 단지 손을 쥐고 펴는 제어신호일 뿐 손을 쥐는 힘의 조절은 가능하지 않다. 어깨에 센서를 부착하여 제어신호로 사용하는 시스템의 경우 손을 쥐는 시점과 강도를 조절할 수 있지만 아주 부자연스러운 상체의 움직임을 해야만 한다. 손목의 신전 굴절센서를 이용하는 경우 위 두 방법보다는 자연스러운 동작을 할 수 있지만 이 방법은 C6 또는 그 아래의 척추손상 환자만이 사용할 수 있다는 한계를 갖는다. 이러한 제어시스템의 대체방안으로 최근에 관심이 되고 있는 것은 척추손상환자에게 마비된 팔에 남아있는 그러나 근수축을 하기에는 부족한 근전도 신호를 검출하여 이 신호의 상대적인 크기에 따라 전기자극의 강도를 조절하여 부드럽게 손을 쥐고 펴는 동작을 하게 하는 것이다. 이러한 근전도제어 전기자극은 환자가 기존의 방법들처럼 부자연스러운 상체움직임을 한다거나 극히 제한된 동작만을 하지 않을 뿐더러 환자가 마비된 근육을 스스로 움직인다는 만족감을 얻으므로서 단순히 기능성전기자극뿐 아니라 유용한 재활치료방법으로 제시되고 있다. 또한 자발적인 근전도 신호를 검출하기 위해 표면전극을 사용하므로 앞에서 제시한 방법과 같이 환자에게 기계적인 센서를 이식할 필요도 없다. 이러한 근전도제어 전기자극기의 일반적인 구성도가 그림 1에 나와 있다.

제어신호로서 근전도 신호를 사용하기 위해서는 다음과 같은 요구를 만족해야 한다. 첫째 제어근육은 최소한의 자발적인 제어신호가 발생되어야 한다. 둘째 제어근육은 다른 정상적인 상지 운동시에는 활성화 되지 않아서 원치 않는 자극이 발생되는 경우가 없어야 한다. 이러한 시스템의 표면전극에서 얻어지는 신호가 그림 2에 나와 있다. 그림 2(a)는 4차례의 전기자극이 가해진 후에 그림 1과 같은 근전도 증폭단을 통해 얻어지는 원신호(raw data)이다. 이 원신호에는 네 개의 근반응과 환자가 스스로 움직이려고 하는 자발적인 근기전이 섞여져 있다. 그림 2(b)는 2(a)에서 근반응만을 따로 분리한 것이다. 그림 2(a)와 그림 2(b)를 비교하여 보면 우리가 근전도 증폭단에서 얻는 신호에서 보여지는 대부분은 근반응임을 알 수 있다. 그리고 그림 2(c)는 (a)신호에서 자발적인 근전도만을 나타내고 있다.

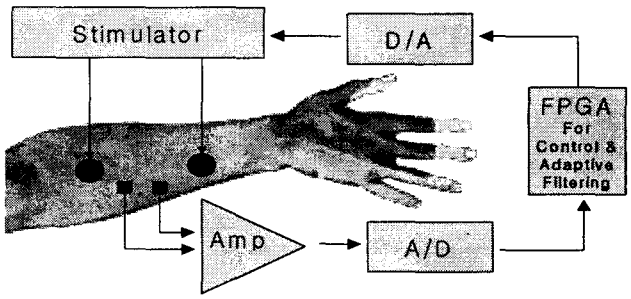
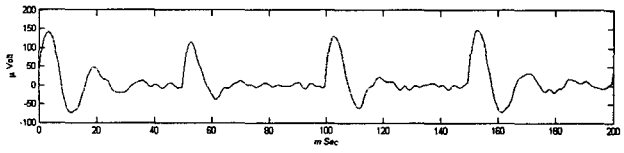
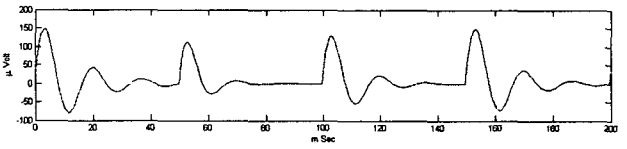


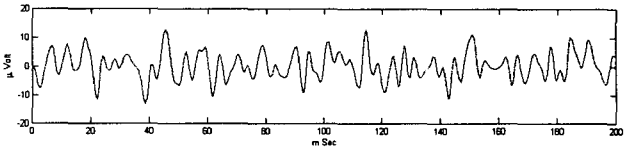
그림 1. 근전도제어 전기자극기 구성도.
Fig. 1. Architecture of an EMG controlled stimulator.



(a) 근반응과 자발근전도가 섞인 신호



(b) 근반응



(c) 자발적인 근전도

그림 2. 근전도기록을 통해 마비환자의 근육에서부터 얻을 수 있는 신호와 그 요소들
Fig. 2. Signals in EMG recordings from the muscles of a tetraplegic individual

2.2 근반응 제거를 위한 최소자승법

근반응은 자극잡음에 비해 상대적으로 길기 때문에 blanking 회로로 제거할 수 없다. 다른 연구에서는 이 근반응을 stationary하다고 간주하고 comb filter를 사용하였다[3]. 그러나 근반응은 자극강도, 근피로도, 근육의 수축정도에 따라 nonstationary하게 변하는 신호이다[6]. 이러한 근반응을 제거하기 위해서는 적응필터의 사용이 적절하다. 적응알고리듬 중에서도 자극주기에 따라 상호연관도가 높은 특성을 보이는 근반응의 경우 블록처리를 하는 최소자승법(LS : least square)이 우수한 결과를 보여주고 있다[6]. 최소자승법의 기본구조가 그림 3에 나와 있다. 여기서 L은 자극이

시작되어서 다음 자극이 발생되기 전까지 하나의 구간에 존재하는 샘플수를 나타내며 L은 샘플링주파수(Fs)와 자극주파수(F_{STIM})의 비(Fs/F_{STIM})로 얻어진다.

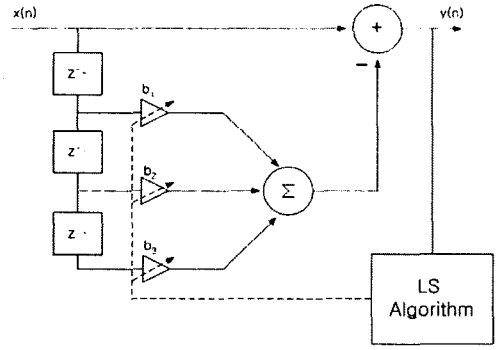


그림 3. 최소자승법을 이용한 적응필터의 구조
Fig. 3. Architecture of an adaptive filter using least square algorithm

그림 3과 같은 M차 트랜스버설 필터에서 벡터 $\underline{y}$ 를 필터에 의해 얻어진 한주기 출력이라고 한다면, 벡터 $\underline{y}$는 다음과 같이 표현할 수 있다.

$$\underline{y} = \underline{x}_0 - \underline{X}\underline{b} \tag{1a}$$

$$\underline{y} = [y(n)\ \ y(n-1)\ \cdots\ y(n-L+1)]^T \tag{1b}$$

$$\underline{X} = [\underline{x}_1\ \underline{x}_2\ \cdots\ \underline{x}_M] \tag{1c}$$

$$\underline{x}_i = [x(n-iL)\ \ x(n-iL-1)\ \cdots\ x(n-L(i+1)+1)]^T \tag{1d}$$

$$\underline{b} = [b_1\ \ b_2\ \cdots\ b_M]^T \tag{1e}$$

필터의 계수 벡터 $\underline{b}$를 적응 추정하기 위해 최소자승 알고리듬을 사용하면 최적 추정치는 아래와 같이 구해진다.

$$\underline{b} = \underline{\Phi}^{-1}\underline{\Theta} \tag{2a}$$

$$\underline{\Phi} = \underline{X}\ \underline{X}^T = \begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \cdots & \phi_{1,M} \\ \phi_{1,2} & \phi_{2,2} & \cdots & \phi_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{1,M} & \phi_{2,M} & \cdots & \phi_{M,M} \end{bmatrix} \tag{2b}$$

$$\underline{\Theta} = \underline{X}^T\underline{x}_0 = [\phi_{0,1}\ \ \phi_{0,2}\ \cdots\ \phi_{0,M}]^T \tag{2c}$$

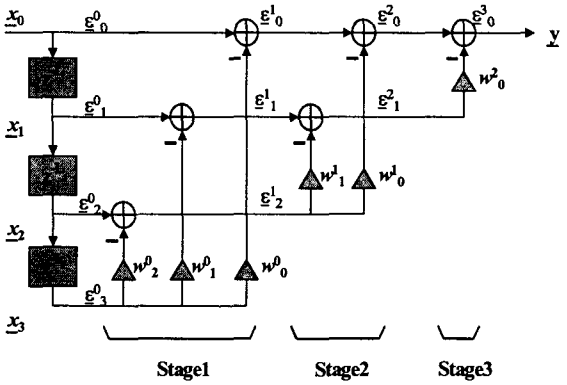
$$\phi_{r,s} = \sum_{i=0}^{L-1} x(n-i-rL)x(n-i-sL)\ , \quad 1 \le r, s \le M \tag{2d}$$

$\underline{\Phi}$는 입력신호행렬 $\underline{x}$ 에 의한 MxM 자기상관행열이고 $\underline{\Theta}$는 Mx1 교차상관벡터이다.

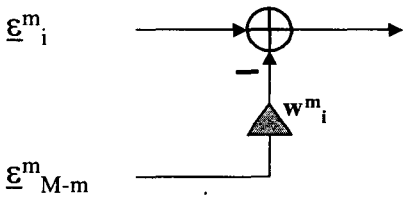
최소자승법은 한 주기의 근반응에 대하여 최적의 계수를 얻을 수 있는 방법이지만, 위의 식(2)에서 나타나 있듯이 역행렬 연산을 통해 최적 계수를 얻기 때문에 계산과정이 복잡하고, 특히 신호의 특성이 ill-conditioned 일 경우 연산과정이 안정적이지 않다. 결국 이러한 문제들은 실시간구현을 위한 하드웨어 설계에 있어 상당한 장애요소로 작용한다. 본 논문에서는 최소자승법과 같은 최적 결과를 얻을 수 있으며 위의 문제점들을 해결할 수 있는 Gram-Schmidt(GS) 적응필터를 사용하여 근반응을 제거하는 방법을 제안한다.

2.3 근반응 제거를 위한 Gram-Schmidt (GS)알고리듬

Gram-Schmidt(GS) 알고리듬은 GS 직교화 과정을 구조화한 방법으로서 다른 적응 알고리듬에 비해 산술적 효율성, 안정성에서 우수한 결과를 보이는 것으로 알려져 있다[9]. GS알고리듬에서는 트랜스버설 구조를 이용한 LS알고리듬처럼 자기상관행렬을 따로 구하거나 직접적인 역행렬연산을 수행하지 않기 때문에 동반되는 연산 과정이 매우 안정적이다.



(a). 3차의 Gram-Schmidt 필터



(b). 1차의 Gram-Schmidt 필터

그림 4. Gram-Schmidt 필터
Fig. 4. Gram-Schmidt filter

그림 4에 3차 GS 필터의 구조를 예시하였다. L 샘플로 구성된 한 주기의 근반응을 각 채널에 가해지는 입력 신호라 할 때, 필터 출력은 다음 과정을 거쳐 얻어진다. 먼저, 식(1)에 정의된 입력 벡터 $\underline{x}_0$, $\underline{x}_1$, $\underline{x}_2$, $\underline{x}_3$ 가 각각 0번째 1번째 2번째 3번째 채널에 가해 진다. 예시된 GS 구조에서는 입력 벡터 $\underline{x}_i$ 를 $\underline{\epsilon}_i$ 로 치환하여 사용하고 있다. $\underline{\epsilon}_i^m$에서 위첨자 m은 스테이지를 나타내고 아래첨자 i 는 해당 스테이지에서의 계수 인덱스를 나타낸다. 첫 단계에서는 $\underline{\epsilon}_3^0$의 자기상관값을 구하고 $\underline{\epsilon}_3^0$와 $\underline{\epsilon}_2^0$의 교차상관값을 구한 뒤 교차상관값을 자기상관값로 나누어 준 값을 계수 w_2^0로 사용한다.

마찬 가지 방법으로 계수 w_1^0와 w_0^0를 구하고 나면 첫 번째 스테이지의 출력을 계산한다. 첫번째 스테이지에서는 이러한 과정을 거쳐 ϵ_3^0가 ϵ_2^0 , ϵ_1^0 , ϵ_0^0 와 직교화 하게 된다. 이 과정을 M차 필터에 대해 정리하면 다음과 같다.

$$\underline{\epsilon}_i^0 = \underline{x}_i, \qquad i = 0, 1, \cdots M \tag{3a}$$

$$w_i^m = \frac{\underline{\epsilon}_i^{mT}\underline{\epsilon}_{M-m}^m}{\left\|\underline{\epsilon}_{M-m}^m\right\|^2}, \qquad \underline{\epsilon}_i^{m+1} = \underline{\epsilon}_i^m - w_i^m\underline{\epsilon}_{M-m}^m$$

$$m = 0, 1, \ldots, M-1, \quad i = 0, \ldots, M-m-1 \tag{3b}$$

GS 필터의 최종 출력은 $\underline{y} = \underline{\epsilon}_0^M$으로 주어지며, 출력을 입력과의 관계식으로 나타내면 다음과 같다.

$$\underline{y} = \underline{X}\,\underline{L}\,\underline{b} \tag{4}$$

위 식에서 _는 식(1)에 정의된 바와 같이 입력 신호로 구성된 (LXM)행렬이며, $\underline{L}$은 GS 필터의 전달함수, $\underline{b} = [\,1\ 0\ \ldots\ 0\,]^T$이다. GS 필터의 전달 함수는 다음과 같이 모듈화된 형태로 표현할 수 있다.

$$\underline{L} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ w_{M-1}^0 & w_{M-2}^0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ w_{M-2}^1 & w_{M-3}^1 & \cdots & 0 \\ 0 & 0 & \cdots & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & 0 & \cdots & 0 \\ w_0^0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{5}$$

GS 필터 구조에서 첫 번째 단계에서 나온 출력은 두 번째 단계에서의 입력신호로서 작용하게 되고 최종출력이 생성되기 까지 이러한 분할과정은 계속된다. 그리고 모든 연산이 하나의 계수를 갖는 적응 필터 구조가 반복되는 형태로 진행되는 systolic array 구조를 갖고 있다.

GS 알고리듬이 LS알고리듬과 등가의 구현법이라는 사실은 과거 연구에서 증명된 바와 같다[9]. GS 알고리듬이 LS 알고리듬에 비해 갖는 장점은 연산 구조가 단순하다는 것과 상대적으로 안정하다는 것, 그리고 하드웨어 구현에 용이한 구조를 갖고 있다는 점을 들 수 있다. LS 알고리듬은 역행렬을 따로 연산한 후, 역행렬 연산을 시행하여 최적 계수를 얻고, 얻어진 계수로 다시 입력을 필터링을 해서 최종 출력을 얻는 반면, GS 알고리듬은 공분산 행렬을 만들거나, 직접적인 역행렬연산을 하지 않고 직교화 과정을 통해 결과를 바로 얻어내기 때문에 연산과정이 LS 알고리듬에 비해 매우 단순하다. LS 알고리듬을 QR 분할을 사용한 고속 과정을 거쳐 연산을 하는 경우[6]에 비하더라도 GS는 계산상의 잇점을 가지고 있으며, LS 과정에서 자기 상관 행렬이 ill-conditioned인 경우에 발생하는 수치적인 불안정성도 극복될 수 있다[9]. 또한 GS 필터의 systolic array 구조는 하드웨어로 구현하는 경우 상당한 잇점을 제공한다.

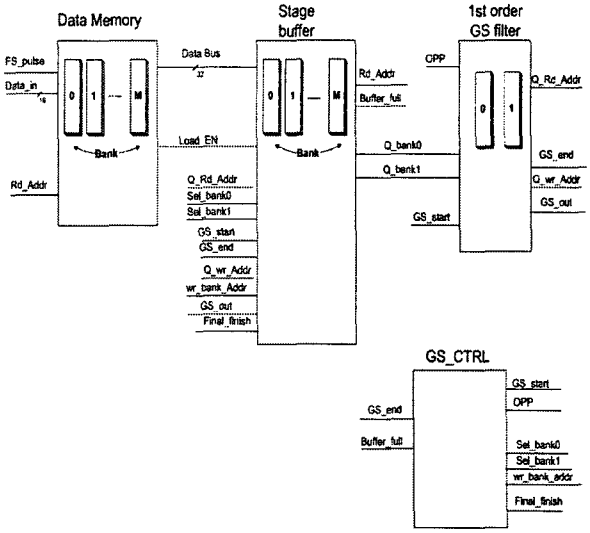


그림 5. M차의 GS 필터가 구현된 FPGA 블록도

Fig 5. FPGA block diagram implementing M th order GS filter

표 1. 단정도 형식의 실수형 산술연산기의 사용게이트수

Table 1. Gate count for single precision floating point ALU

	Gates for design
adder & subtractor	9,156
multiplier	17,819
divider	17,806

2.4 GS 알고리듬이 구현된 FPGA

GS 알고리듬이 실시간 구현에 적합한 것을 보이기 위해 FPGA에 알고리듬을 구현하였다. 필터차수는 1차에서 15차까지 임의로 설정할 수 있도록 설계하였다. 알고리듬 구현을 위한 산술연산은 신호처리의 정밀도를 높이기 위해 IEEE754형식의 단정도 부동소수점 연산방법을 선택하였고 이에 필요한 산술연산기를 FPGA로 구현하였다. 산술연산기에 사용된 게이트 수는 표 1에 나와 있다. 본 연구에서는 GS 필터의 systolic array 구조를 이용하여 1차의 GS 필터를 설계하고 여기에 입력채널만 바꾸어 주면서 높은 차수의 필터링을 수행하도록 FPGA를 구성하였다.

FPGA에서 구현된 GS 필터의 블럭도가 그림 5에 나와 있다. 먼저 A/D된 16비트 데이터는 Memory bank & Input data 부(component)에 적재된다. 하나의 메모리 뱅크크기는 근반응의 한주기에 해당하는 채널크기와 동일하며 전체 메모리뱅크는 필터차수보다 하나 많은 수의 메모리 뱅크로 구성된다. 1차 GS filter부에서는 1차의 GS필터링을 하고 이 결과를 Stage buffer부에 저장한다. 이러한 과정은 필터차수에 따라 반복되며 GS_CTRL에서 전체제어가 이루어 진다.

그림 6은 1차 GS filter의 내부도이며 그림 4(b)에 있는 1

차의 GS 필터구조를 구현한 것이다. 그림 6에서 보면 두 입력벡터 $\underline{\epsilon}^m_{M-m}$과 $\underline{\epsilon}^m_i$가 MAC (multiplier accumulator)에 입력되어 진다. MAC에 $\underline{\epsilon}^m_{M-m}$ 와 $\underline{\epsilon}^m_{M-m}$가 입력되면 자승의 합의 형태가 되어 우리가 구하고자 하는 자기상관값이 얻어지고 레지스터 AC에 저장된다. 또한 $\underline{\epsilon}^m_{M-m}$와 $\underline{\epsilon}^m_i$가 입력되면 결국 벡터의 내적이 연산되어 상호상관값이 얻어지고 레지스터 CC에 저장된다. 자기상관값은 제수, 상호상관값은 피제수가 되어 Divider부 즉 나눗셈기에 입력이 되어 나온 값은 w^m_i에 저장되면 이는 1차 GS 필터의 계수가 되어 w^m_i와 $\underline{\epsilon}^m_{M-m}$가 곱셈기인 multiplier부를 통과하여 나온 결과와 $\underline{\epsilon}^m_i$를 빼주게 되면 결국 1차의 GS 필터의 출력을 얻게 된다.

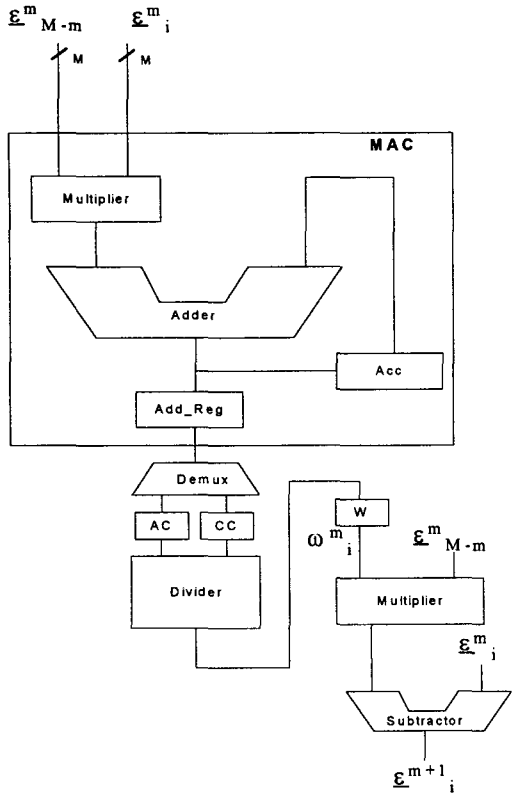


그림 6. FPGA로 구현된 1차의 GS 필터.
Fig. 6. 1st order GS filter implemented in FPGA

3차의 필터링 과정은 1차 GS filter부에 두 입력벡터 $\underline{\epsilon}^m_{M-m}$과 $\underline{\epsilon}^m_i$가 m과 i에 순차적으로 입력되어 최종 필터출력을 얻게 된다. 이러한 systolic 과정을 표 2에서 나타내었다. 먼저 state1에서는 $\underline{\epsilon}^0_3(\underline{x}_3)$의 자기상관값을 구한 뒤 레지스터 AC 레지스터에 저장한고 state2에서는 $\underline{\epsilon}^0_3(\underline{x}_3)$와 $\underline{\epsilon}^0_2(\underline{x}_2)$의 상호상관값을 구하여 레지스터 CC에 저장한다. 이 때 필터계수 w^0_2가 Divider를 거쳐 나오면 state3에서는 $\underline{\epsilon}^0_3(\underline{x}_3)$와 $\underline{\epsilon}^0_1(\underline{x}_1)$의 상호상관값을 MAC에서 구하는 동안 Multiplier와 Subtractor 에서는 앞에서 구한 w^0_2를 가지고 필터출력 $\underline{\epsilon}^1_2$를 얻게 된다.

표 2 FPGA에 구현된 3차의 GS 필터링 과정(MAC : Multiplier & Accumulator, Div : Divider, Dvd : Dividend, Dvs : Divisor Mult : Multiplier, Sub : subtractor)

Table 2 3 order Filter processing implemented FPGA

Stage0

	state1	state2	state3	state4
MAC in1	$\underline{\epsilon}^0_3(\underline{x}_3)$	$\underline{\epsilon}^0_3(\underline{x}_3)$	$\underline{\epsilon}^0_3(\underline{x}_3)$	$\underline{\epsilon}^0_3(\underline{x}_3)$
MAC in2	$\underline{\epsilon}^0_3(\underline{x}_3)$	$\underline{\epsilon}^0_2(\underline{x}_2)$	$\underline{\epsilon}^0_1(\underline{x}_1)$	$\underline{\epsilon}^0_0(\underline{x}_0)$
MAC out	AC_3	CC_{32}	CC_{31}	CC_{30}
Div. Dvd.		CC_{32}	CC_{31}	CC_{30}
Div. Dvs.		AC_3	AC_3	AC_3
Div. out		w^0_2	w^0_1	w^0_0
Mult. in1			w^0_2	w^0_1
Mult. int2			$\underline{\epsilon}^0_3$	$\underline{\epsilon}^0_3$
Mult. out			$w^0_2 \cdot \underline{\epsilon}^0_3$	$w^0_1 \cdot \underline{\epsilon}^0_3$
Sub. in1			$w^0_2 \cdot \underline{\epsilon}^0_3$	$w^0_1 \cdot \underline{\epsilon}^0_3$
Sub. in2			$\underline{\epsilon}^0_2$	$\underline{\epsilon}^0_1$
Sub. out			$\underline{\epsilon}^1_2$	$\underline{\epsilon}^1_1$

Stage1 ‖ stage2

	state5	state6	state7	state8	state9	state10
MAC in1	$\underline{\epsilon}^1_2$	$\underline{\epsilon}^1_2$	$\underline{\epsilon}^1_2$	$\underline{\epsilon}^2_1$	$\underline{\epsilon}^2_1$	
MAC in2	$\underline{\epsilon}^1_2$	$\underline{\epsilon}^1_1$	$\underline{\epsilon}^1_0$	$\underline{\epsilon}^2_1$	$\underline{\epsilon}^2_0$	
MAC out	AC_2	CC_{21}	CC_{20}	AC_1	CC_{10}	
Div. Dvd.		CC_{21}	CC_{20}		CC_{10}	
Div. Dvs.		AC_2	AC_2		AC_1	
Div. out		w^1_1	w^1_0		w^2_0	
Mult. in1	w^0_0		w^1_1	w^1_0		w^2_0
Mult. int2	$\underline{\epsilon}^0_3$		$\underline{\epsilon}^1_2$	$\underline{\epsilon}^1_2$		$\underline{\epsilon}^2_1$
Mult. out	$w^0_0 \cdot \underline{\epsilon}^0_3$		$w^1_1 \cdot \underline{\epsilon}^1_2$	$w^1_0 \cdot \underline{\epsilon}^1_2$		$w^2_0 \cdot \underline{\epsilon}^2_1$
Sub. in1	$w^0_0 \cdot \underline{\epsilon}^0_3$		$w^1_1 \cdot \underline{\epsilon}^1_2$	$w^1_0 \cdot \underline{\epsilon}^1_2$		$w^2_0 \cdot \underline{\epsilon}^2_1$
Sub. in2	$\underline{\epsilon}^0_1$		$\underline{\epsilon}^1_1$	$\underline{\epsilon}^1_0$		$\underline{\epsilon}^2_0$
Sub. out	$\underline{\epsilon}^1_0$		$\underline{\epsilon}^2_1$	$\underline{\epsilon}^2_0$		$\underline{\epsilon}^3_0$

state9에서는 $\underline{\epsilon}^2_1$와 $\underline{\epsilon}^2_0$의 상호 상관값을 구함으로서 MAC에서의 연산은 끝나지만 여기서 구한 상호상관값을 가지고 구한 필터계수 w^2_0를 가지고 필터링을 해야하므로 하나의 state가 추가되어 모두 10번의 state가 지나야 3차의 GS 필터링이 끝나게 된다. 여기서 보여지는 것처럼 자기상관값이나 상호상관값을 구하는 MAC과 그 외 필터링을 하는 요소는 FPGA 특성상 병렬처리를 하므로 상용 DSP칩이나 일반 프로세서에 비해 2배의 속도향상을 가져온다. 또한 구조면에서 보면 그림 6에서 나타낸 것처럼 MAC(multiplier, adder, accumulator), divider, multiplier, substractor가 각

각 1개로 구성되어 있는 비교적 단순한 구조이지만 고차의 적응필터링을 할 수 있는 것은 GS 알고리즘이 systolic 특성을 가지고 있기 때문이다.

2.5 FPGA 검증

환자로부터 얻은 A/D(analog to digital)된 신호를 FPGA에 입력하여 FPGA에서 구현된 GS 알고리듬에 대한 검증(verification)을 하였다. 필터는 VHDL(very high speed integrated circuit hardware description language)로 설계하였으며 합성은 Xilinx사의 ISE 5.2i를 사용하였고 검증은 Model Sim XE 5.6a를 가지고 하였다. 근반응을 제거하기 위해 GS 필터링에 소요되는 시간 T_{GS}는 수식 (6)와 같다.

$$T_{GS} = (D_{state} \times N_{state} \times T_{CLK}) \quad (6)$$

단 $D_{state} = D_{MAC} \times$

$$N_{state} = \sum_{k=1}^{M+1} k$$

D_{state}는 하나의 state에서 소요되는 사이클수로 이는 MAC에서 하나의 샘플이 처리되는 사이클수 D_{MAC}과 근반응주기에 해당하는 벡터크기 L의 곱만큼 지연된다. 본 연구에서 개발한 MAC의 경우 D_{MAC} 은 부동소수점 연산을 하기 때문에 8 클락이 소요된다. L은 자극주기와 샘플링에 의해 결정되므로 경우에 따라 가변적이다. N_{state}는 한주기의 근반응을 제거하기 위해 필요한 state의 수이다. 표 2에서 필터차수가 3차이므로 M에 3을 대입하면 이때 필요한 state의 수는 10으로 얻어진다. T_{CLK}는 FPGA에서 사용되는 주클락(Main clock)이다. 예를 들어 근반응의 한주기인 L이 10샘플이고 주클락 10ns인 경우 D_{MAC}이 8 사이클, N_{state}는 6차의 경우 28 states이므로 이때 총 연산시간 즉 T_{GS}는 22.4 μsec가 소요된다. 이 경우 L이 10이므로 10개의 샘플이 채워지면 그때마다 한번씩 GS 필터링을 하고 다음 10개 샘플이 채워지기 전에 처리가 끝나야 하므로 최대 샘플링주파수는 T_{GS}에 10이 나누어진 2.24μsec의 역수인 447 kHz가 된다.

3. 실 험 방 법

GS 알고리듬의 성능을 평가하기 위해 시뮬레이션 프로그램으로 모의데이터와 실제 환자 데이터를 가지고 각각 실험을 하였다.

3.1 모의데이터

자극된 근육에서 근전도신호를 기록할 때 자극에 의해 생성되는 근반응과 자발적인 근전도를 필터링 전단계에서 분리하는 것은 불가능하다. 또한 근반응과 자발적인 근전도의 상대적인 크기에 따라 필터의 성능이 달라질 수 있기 때문에 실제 근전도데이터를 가지고 필터성능을 비교하는 것보다 모의데이터를 사용함으로써 필터수행능력을 보다 객관적으로 평가할 수 있을 것이다. 자극강도가 증가하는 동안 근반응의 변화는 크기와 모양의 변화로 표현될 수 있다. 참고문헌[6]에서는 실제 근반응 측정결과를 토대로 근반응은 지수함수적으로 감소되는 사인파로 가정하고 크기와 시정수를 가변함으로써 근반응의 크기와 모양을 조절하였다. 또한 자발적인 근전도는 400Hz 저대역필터로 밴드제한된 가우시안 잡음으로 가정되었다. 이 경우 근전도의 j번째 채널은 아래와 같이 나타낼 수 있다.

$$s_j(i) = a_j \, exp(-i/\tau_j)^* \, sin(2\pi 3\, i/N) + v(i), \quad i = 0, 1, \cdots N-1 \quad (7)$$

N은 4000Hz의 표본주파수와 20Hz의 자극주파수에 따라 200으로 고정되었다. $a_j \, exp(-i/\tau_j)^* \, sin(2\pi 3\, i/N)$는 근반응을 나타내며 $v(i)$는 자발적인 근전도를 나타낸다. 파라메터 a_j는 j번째 근반응의 크기를 결정하는 크기조정 요소이고 τ_j는 모양을 결정하는 요소이다.

3.2 측정 근전도데이터

측정데이터를 얻기 위해 BioPacSystem사 제품의 MP150을 사용하였다. 증폭단은 UIM100C를 연결하였고 자극은 STM100과 STMISOB를 사용하였다. 실험은 1000Hz의 샘플링을 하였고 자극주파수는 20Hz이므로 각 자극에 대한 근반응의 주기는 50샘플이고 6차의 GS 알고리듬을 적용하였다. 자극의 크기는 20Volt에서 70Volt사이의 전압을 인가하였다. 실험은 전기자극이 시작된 3초후에 환자에게 부저소리(beep sound)와 함께 자발적인 근수축을 실시하게 한 뒤 부저소리가 끝나면 자발적인 근수축을 멈추도록 하였다.

4. 실 험 결 과

4.1 모의데이터

GS 알고리듬을 구현한 필터의 성능을 필터 차수에 따라 평가하기 위해 샘플링주파수는 4000Hz, 채널당 샘플은 200개인 모의데이터에 대해 56개의 0을 추가하여 256 포인트 FFT를 수행한 뒤 각 구간마다 평균을 취한 결과가 그림 7에 나와 있다. 필터의 차수는 1차, 3차, 6차, 12차에 대해 수행하였다. 또한 많은 계산량과 연산결과의 불안정성을 갖는 최소자승법알고리듬과의 비교가 표 3에 나와 있다. 먼저 그림 7에서 보면 차수가 증가하면서 근전도신호의 크기에 근접하는 결과를 볼 수 있다. 1차 필터링에서는 물론 크기가 감소되어 근반응이 일부 제거되었지만 근반응이 여전히 크게 존재하고 있다. 그러므로 과거에 사용하였던 comb필터로는 근반응을 적절하게 제거할 수 없음을 알 수 있다. 6차의 경우 대부분의 근반응은 제거되고 자발근전도의 크기는 감소하지 않았으며 12차의 경우 필터의 출력과 자발적인 근전도의 크기가 거의 같음을 알 수 있다.

표 3에서는 이러한 결과를 수치적으로 보여주고 있다. 근반응과 근전도가 섞인 신호가 LS와 GS 에 동일하게 42.796의 크기로 입력된다. 이때의 근전도의 크기는 0.11이다. 여기서의 수치는 모두 RMS(root mean square)값들이다. 두 알

고리즘에 의한 RMS값은 1차, 3차, 6차 12차에 대해 서로 동일한 결과를 얻는다. 1차에서는 42.796의 크기에서 6.3605로 근반응을 많이 줄였으나 자발근전도보다는 상대적으로 큰 크기를 가지므로 근전도 신호를 제어신호로 사용하기에는 false-positive가 빈번하게 발생할 수 있는 결과이다. 차수가 증가하면서 점차 근전도 신호의 크기에 근접하는 것을 볼 수 있다. 12차의 경우 0.10879로 0.11에 근접한 수치를 보이고 있다. 그러나 6차 필터링 결과 0.21895의 값은 12차의 결과에 비해 상대적으로 큰 차이가 보이지 않는 것을 알 수 있다. 차수가 증가하면 증가할수록 필터의 성능은 좋아지지만 실제 회로구현에서 있어 연산속도의 증가와 메모리자원의 사용증가로 이어지기 때문에 6차의 필터가 적절하다고 판단된다.

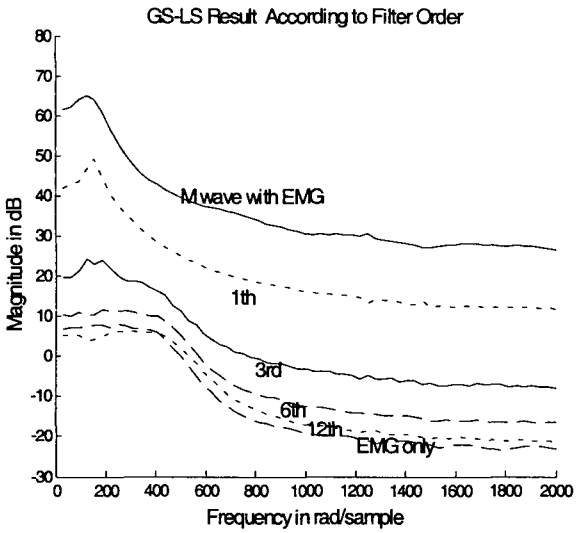


그림 7. 모의데이터를 필터 차수별로 GS 필터링한 후 스펙트럼분석 결과

Fig. 7. Spectra before and after GS filtering according to filter order

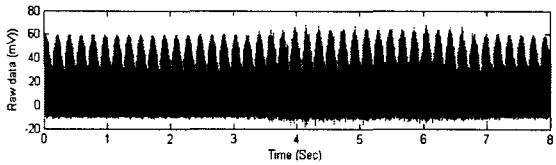
표 3. 필터차수에 따른 LS 알고리듬과 GS 알고리듬의 RMS 비교

Table 3. Root mean square comparison between least square algorithm and GS algorithm

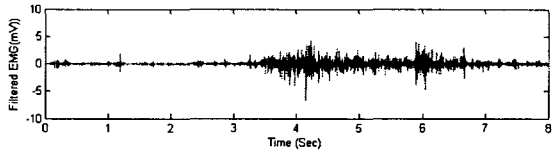
	LS	GS
M wave with EMG	42.796	42.796
EMG	0.11	0.11
1 th	6.3605	6.3605
3 rd	0.77115	0.77115
6 th	0.21895	0.21895
12 th	0.10879	0.10879

4.2 측정데이터

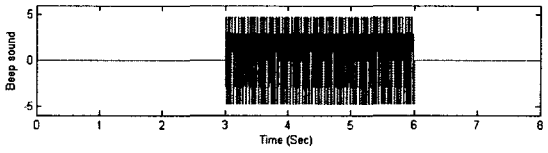
자발적인 근수축이 있기 전에 전기자극에 의한 근반응의 제거정도와 근수축이 있을 때 근반응의 제거정도를 보고 이때 남아 있는 자발근전도의 크기를 보기 위한 실험의 결과가 그림 8에 나와 있다. 그림 8(a)는 GS 알고리듬을 적용하기 전이고 그림 8(b)는 6차의 GS 알고리듬을 적용한 후의 결과이다. 그림 8(c)는 환자에게 근수축을 지시하는 들려주는 부저음이다. 그러므로 환자는 이 부저음이 발생되는 시간동안 근수축을 하게 된다. 그림 8(a)를 그림 8(c)의 부저음을 기준으로 보면 자발근전도가 발생되는 시점과 그전 시점에서 별다른 차이를 볼 수 없다. 그러나 필터링을 한 그림 8(b)에서 보면 자발근전도가 발생되기 이전시점에서는 신호가 대부분 감쇄되었고 자발근전도가 발생되는 시점에서 보면 근전도 신호만이 검출되는 것을 볼 수 있다. 그림 8(b)와 그림 8(c)를 비교해 보면 부저음이 울리기 시작하면서 근수축이 일어나서 점점 증가하다가 감소하는 결과를 볼 수 있다.



(a)근전도증폭단에서 얻어지는 신호



(b) GS 제거기로 필터링한 데이터



(c)환자에게 근수축을 지시하는 부저음

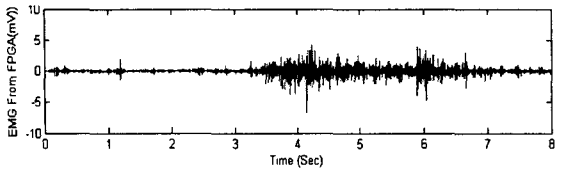
그림 8 환자로부터 측정된 근전도 데이터의 분석

Fig. 8 Anaysis of EMG data

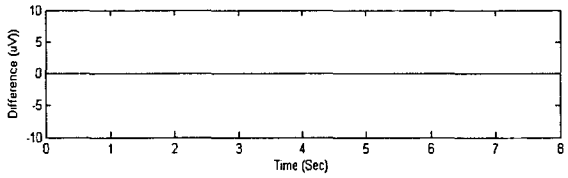
이러한 근수축의 정도에 따라 자극강도를 크기를 변화한다면 환자는 원하는 만큼 근육을 움직일 수 있게 된다. 실제 환자데이터를 통한 실험결과에서 알 수 있듯이 6차의 필터링 후 근전도 신호를 제어신호로 쓸 수 있을 정도도 필터링이 잘 되어있음을 알수 있다.

4.3 FPGA결과와 시뮬레이션결과 비교

그림 9는 그림 8에서 실시한 동일데이터에 대해 FPGA에 구현된 6차의 GS 필터를 MODEL SIM XE 5.6a으로 검증(verification)한 결과이다. 그림 8(b)에 나타난 시뮬레이션 결과는 64bit 배정도 부동소수점연산결과이고 그림 9(a)는 32bit 단정도 부동소수점연산을 FPGA에 구현한 결과이며 그림 9(b)는 시뮬레이션과 FPGA 결과의 차이값이다. 두 결과의 비교를 통해 본 연구에서 FPGA에 구현하기 위해 개발한 단정도 부동소수점연산 산술연산기로도 충분한 연산결과를 얻을 수 있음을 보여 주고 있다.



(a) FPGA출력



(b) 시뮬레이션결과 그림 8(b)와 FPGA출력의 차이값

그림 9. FPGA에서 구현된 GS 필터결과

Fig. 9. Results of the GS filter implemented in FPGA

5. 결 론

본 연구의 목적은 사지마비환자가 일상생활의 일부를 보다 자연스럽게 영위할 수 있게 하는 것이다. 손을 쥘 수 있는 경우 그렇지 않은 경우에 비해 생활을 하는데 있어 많은 불편함을 해소할 뿐 아니라 환자나 가족에게도 큰 정신적 위안이 된다. 이에 근전도제어 전기자극기에서 가장 문제가 되는 근반응 제거에 대해 GS 알고리듬을 제안하였고 나아가 임상적용을 위해 FPGA에 구현 가능한 6차의 GS 필터를 설계하였다. 일반적으로 최소자승법알고리듬은 연산이 복잡하고 근반응의 경우 샘플간에 상호연관성이 적기 때문에 RLS(Recursive Least Square)와 같이 계산량을 줄이는 알고리듬을 적용할 수 없다. 이에 반해 GS 알고리듬은 블록처리로 자극구간마다 필터계수를 구하면서 출력을 얻을 수 있고 동일 구조가 반복적으로 나타나기 때문에 단순한 구조의 설계로 고차의 확장성이 용이하며 계수를 구하는 처리와 필터링처리가 병렬연산이 되기 때문에 적은 처리시간을 갖는다. 또한 샘플링주파수 증가와 필터차수가 증가하여 실시간 처리가 어려울 경우 1차 GS필터의 기본구조를 추가하여 병렬배치하면 추가할 때 마다 2배의 처리시간증가를 갖는다. 현재 개발한 근반응제거용 FPGA를 이용한 근전도제어 기능적전기자극기를 개발하고 있으며 추후 임상적용을 할 예정이다.

감사의 글

본 연구는 과학기술부 한국과학재단 지정 연세대학교 의용계측 및 재활공학 연구센터의 지원에 의한 것입니다.

참 고 문 헌

[1] L.Vodovnik, C.Long, J.B.Reswick, A.Lippay, and D.Starbuck, "Myo-electric control of paralyzed muscles", IEEE Trans. Rehab. Eng., vol.12, pp. 169-172, July/Oct. 1965.

[2] R.H.Nathan, "An FNS-based system for generating upper limb function in the C4 quadriplegic", Med.Bio.Eng.Comput., vol.27, pp. 549-556, Nov. 1989.

[3] C.Frigo, M.Ferrarin, W.Frasson, E.Pavan, and R.Thorsen, "EMG signal detection and processing for on-line control of functional electrical stmulation", J Electromyo. and kid., pp. 351-360, Oct. 2000.

[4] R.I. Barbosa-Mier, and A.C. Chica, "Comparison of adaptive system for suppression of stimulation artifacts in EMG signals", IEEE pp. 144-148, 1995

[5] W.Peasgood, et al. "EMG-controlled closed loop electrical stimulation using a digital signal processor" , Electronics letters, vol.36, No.22, pp. 1832-1833, Oct. 2000.

[6] S.Sennels, et al. "Functional Neuromuscular Stimulation Controlled by Surface Electromyographic Signals Produced by volitional activation of the same muscle", IEEE Trans. Rehab. Eng., vol. 5, No. 2, pp. 195-295 June. 1997.

[7] T.Keller, A.Curt, M.R. Popovic, "Grasping in high lesioned tetraplegic subjects using the EMG controlled neuroprosthesis", J. Neuro Reh vol 10, pp. 251-255 , 1998.

[8] R.Merletti, M.Knaflitx, and C.J. Deluca, "Electrically evoked myoelectric signal", Crit.Rev.Biomed.Eng., vol19, No. 4, pp. 293-340, 1992.

[9] K.Gerlach and F.F.Kretschmer Jr., "Convergence Properties of Gram-Schmidt and SMI Adaptive Algorithms", IEEE Trans. Aerospace &Electronic System, vol AES-26, No. 1, pp. 44-56, Jan. 1990.

저 자 소 개

염 호 준 (廉 晧 準)

1969년 2월 12일생. 1995년 연세대학교 의공학과 졸업. 1997년 동 대학원 의용전자공학과 졸업(석사). 1997년~현재 동 대학원 의공학과 박사과정

Tel : 033-760-2807

Fax : 033-760-2197

E-mail : hojoon@bme.yonsei.ac.kr

박 영 철 (朴 榮 喆)

1964년 2월 26일생. 1986년 연세대 전자과 졸업. 1988년 동 대학원 전자공학과 졸업(석사). 1993년 동 대학원 전자공학과 졸업(공박). 2002년~현재 연세대 정보처리학부 조교수

Tel : 033-760-2744

E-mail : young00@dragon.yonsei.ac.kr

윤 형 로 (尹 亨 老)

1949년 2월 17일생. 1972년 연세대학교 전기공학과 졸업.1974년 동 대학원 전기공학과 졸업(석사). 1986년 동 대학원 전자공학과 졸업(공박). 1980년~현재 연세대학교 의용전자공학과 교수

Tel : 033-760-2850, Fax : 033-763-1953

E-mail : hryoon@dragon.yonsei.ac.kr