

실시간처리를 위한 태아심음 추출 알고리즘

°이 응 구*, 이 용 희*, 김 선 일**, 이 두 수*
 한양대학교 전자공학과*, 계량의학과**

Fetal heart rate estimation algorithm for real-time processing

* Dept. of Electronic Engineering, Hanyang University
 ** Dept. of Medical Information and Management University

ABSTRACT

Despite the simplicity of processing, a conventional autocorrelation function(ACF) method for the precise determination of fetal heart rate (FHR) has many problems. In case of weak or noise corrupted Doppler ultrasound signal, the ACF method is very sensitive to the threshold level and data window length. It is very troublesome to extract FHR when there is a data loss. To overcome these problems, the high resolution pitch detection algorithm was adopted to estimate the FHR. This method is more accurate, robust and reliable than the ACF method. With a lot of calculation, however, it is impossible to process real time FHR estimation. This paper is presented a new FHR estimation algorithm for real time processing.

I. 서 론

태아의 박동수 검출 방법은 고전적으로 산모의 체외로 전달되는 태아심음을 이용하는 방법과 자궁속의 태아에 직접 전극을 삽입하여 얻어지는 전기적인 신호의 주기를 측정해서 얻는 방법등이 있다. 그러나 후자의 직접적인 방법은 침습적인 방법(invasive method)으로서 복잡한 시행과정과 위험을 수반하기 때문에 특별한 경우 외에는 행하지 않고 체외에서 측정하는 간접적인 방법이 많이 쓰여지고 있다. 이 간접적인 방법에는 태아심음의 직접적인 증폭을 통한 방법과 초음파의 도플러 효과를 이용하는 방법이 있으나 도플러 효과를 이용한 방법이 최근에는 주종을 이루고 있다.

근간에는 자기상관 함수(Autocorrelation function)을 이용하는 방법이 많이 쓰여지고 있다.^[1-7] 그러나 이 방법은 자기상관 함수에서 심음주기를 찾을 때 일정한 레벨 이상의 피크중 첫번째 피크를 찾는 방법이기 때문에 레벨의 설정에 따라 또는 태아심음의 상태에 따라 심음주기 측정오차가 나타나며 창의크기(Window size)

가 길어야 정확한 주기를 구할 수 있으나 변화에 대한 응답시간이 길어지는 단점이 있다. 그러므로 태아심음주기의 미세한 변화를 처음으로서 궁극적으로 태아의 건강상태도 알 수 있는 데에는 미치지 못하며 또한 전체적인 심음주기 측정오차도 피할 수 없다. 또한 적당한 신호대 잡음비(SNR)에서는 자기상관 함수법으로도 정확한 심음주기를 찾을 수 있으나, SNR이 낮거나 신호의 손실이 있을때 ACF는 부적합하다. 불행히도 이와같은 상황이 임상에서는 빈번히 발생한다. 특히 임신 34주 이내의 태아에서 태아의 움직임으로 FHR의 변화가 심할때, 신호의 손실이 많이 나타난다. 기존 ACF의 단점을 요약하면 다음과 같다. 1) 성능평가는 Data window length와 신호의 SNR에 의존한다. Window length가 길면 길수록 좋은 결과를 가져오나 주기 변화에 따른 응답이 늦어지게 된다. 2) 적당한 Window 설정에 어려움이 존재한다. 어떠한 Window를 설정하더라도 Window의 효과가 ACF Index가 증가함에 따라 점차 영으로 감소하게 된다. 3) 주기결과는 데이터 손실에 큰 영향을 받는다. 4) 첫번째 피크는 특히 신호의 SNR이 낮을때 Threshold level에 민감하다.

이와같은 문제점을 해결하고 정확한 태아심음을 추출하는데 고해상 피치검출 알고리즘을 적용한다.^[8-10] 이 방법은 ACF 보다 정확한 피치주기를 찾고 있으나 연산량이 많아 실시간 처리에 부적합하다. 고해상 피치검출 알고리즘에서 연산량을 줄이기 위한 새로운 알고리즘을 제안하고 이 알고리즘이 기존의 신호처리 전용 프로세서에 적합한 알고리즘임을 보인다.

II. 고해상 태아심음 추출

태아심음 샘플 $S[n_1:n_2]$ ($1 \leq n_1 < n_2$)의 벡터를 다음과 같이 정의한다.

$$S[n_1:n_2] = (S_{n_1}, \dots, S_{n_1+k}, \dots, S_{n_2})^T; k=0,1, \dots, n_2-n_1$$

$$S_i = S(t+t_0)|_{t=(i-1)T} \quad ; i=1,2, \dots \quad (1)$$

두개의 n 차원 벡터 $X_n(i_0) = (x_1, \dots, x_j, \dots, x_n)^T$ 와 $Y_n(i_0) = (y_1, \dots, y_j, \dots, y_n)^T$ 로 변형해서 표현한다.

$$X_n(i_0) = S[1:n], \quad Y_n(i_0) = S[n+1:2n] \quad (2)$$

여기서 i_0 는 시간첨자 t_0 에 대한 샘플첨자이다. $t=t_0$ 에서 최적 정

실시간처리를 위한 태아심음 추출 알고리즘

수주기 N_0 는 정규이산자승오차함수의 최소화로 구해진다.

$$N_0 = \underset{n, a(t_0) > 0}{\operatorname{argmin}} \left\{ C = \frac{\sum_{j=1}^n [x_j - a(t_0)y_j]^2}{\sum_{j=1}^n x_j^2} \right\} \quad (3)$$

실질적인 주기값 N_0 는 $N_{\min} \leq n \leq N_{\max}$ 범위에 대하여 구한다. 태아심음주기가 1.5-3 Hz 사이의 값을 갖으므로 샘플주기 $F_s=2400\text{Hz}$ 일때, N_{\min} 은 800(2400/3)이고 N_{\max} 는 1600(2400/1.5) 이다. 식(3)의 최적화는 X_n 과 Y_n 의 상호상관계수 $\rho_n(X_n, Y_n)$ 를 $[N_{\min}, N_{\max}]$ 에서 최대값으로 구한다.

$$N_0 = \underset{n}{\operatorname{argmax}} \rho_n(X_n, Y_n) \quad N_{\min} \leq n \leq N_{\max} \quad (4)$$

$\rho_n(X_n, Y_n) = (X_n \cdot Y_n) / |X_n| \cdot |Y_n|$ 이고, $(X_n \cdot Y_n)$ 는 벡터의 내적이며, $|X_n|$ 와 $|Y_n|$ 는 벡터의 Norm을 나타낸다.^[10] 상호상관계수 $\rho_n(X_n, Y_n)$ 은 n 이 N_{\min} 에서 N_{\max} 까지 벡터의 내적과 Norm을 계산하는데 이때 내적과 Norm값은 매 샘플 마다 반복해서 계산해야 한다.

III. 실시간 태아심음 추출 알고리즘

II장에서 구한 태아심음은 반복 연산으로 계산량이 많아지게 된다. 이와같은 반복 연산을 효율적으로 줄이기 위한 알고리즘을 제시한다. 처음 $2 N_{\max}$ 의 데이터로 부터 피치를 구할 때, N_{\min} 에서 N_{\max} 까지 $(X_n \cdot Y_n)$, $|X_n|$, 그리고 $|Y_n|$ 의 모든 값을 표1과 같이 저장한다. 새로운 데이터가 입력되면 $(X_n \cdot Y_n)$, $|X_n|$, 그리고 $|Y_n|$ 의 값을 N_{\min} 에서 N_{\max} 까지 계산하게 된다. 이 때 $(X_n \cdot Y_n)$, $|X_n|$, 그리고 $|Y_n|$ 의 값은 새로운 데이터를 제외한 나머지 부분에서 이미 앞에서 구한 것과 중복된다. 이와같은 중복된 연산을 줄이기 위하여 이전에 구한 값들을 저장하고 새로 갱신되는 값만 계산하여 저장한다.

N_{\min}	$N_{\min+1}$...	$N_{\max-1}$	N_{\max}
$(X \cdot Y)_{N_{\min}}$	$(X \cdot Y)_{N_{\min+1}}$...	$(X \cdot Y)_{N_{\max}}$	$(X \cdot Y)_{N_{\max}}$
$ X _{N_{\min}}$	$ X _{N_{\min+1}}$...	$ X _{N_{\max-1}}$	$ X _{N_{\max}}$
$ Y _{N_{\min}}$	$ Y _{N_{\min+1}}$...	$ Y _{N_{\max-1}}$	$ Y _{N_{\max}}$

표 1. 메모리 데이터 할당

Table 1. Assignment of memory data

갱신된 값을 계산하는 식은 다음과 같다. 그림 1과 같이 입력된 도플러 신호를 이전에 계산된 세그먼트 X, Y와 현재 세그먼트 X', Y'로 표시한다.

그림 1에서 현재의 벡터내적 $(X \cdot Y)'$ 은 식(5)와 같이 이전 벡터내적 $(X \cdot Y)$ 값에서 첫번째 값 A·B를 제외한 나머지 값은 동일하므로 이전의 $(X \cdot Y)$ 값에서 A·B를 빼고 새로운 값 B·C를 더해서 구한다.

$(X \cdot Y)' = (X \cdot Y) - A \cdot B + B \cdot C = (X \cdot Y) + B(C - A)$ (5)
또한 Norm $|X|'$ 과 $|Y|'$ 은 같은 방법으로 식(6)과 식(7)로 표현되며, 이전의 값에서 제외되는 부분을 빼고 새로운 값을 더함으로써 구한다.

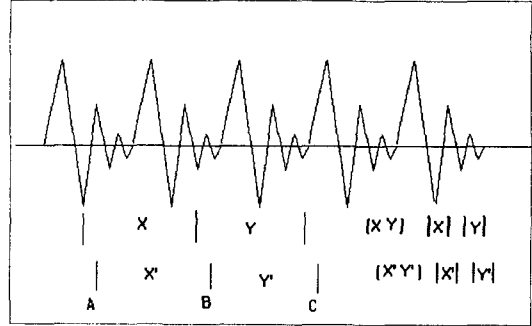


그림 1. 도플러 입력신호

Fig. 1. Doppler input signal

$$|X|' = \sqrt{|X|^2 - A^2 + B^2} \quad (6)$$

$$|Y|' = \sqrt{|Y|^2 - B^2 + C^2} \quad (7)$$

갱신된 새로운 값에 대한 상호상관계수 값은 식(8)로 구한다.

$$\rho' = \frac{(X \cdot Y)'}{|X|' \cdot |Y|'} \quad (8)$$

입력된 데이터는 $2 N_{\max}$ 크기의 버퍼에 저장하고 처음 $2 N_{\max}$ 데이터에 대하여 세그먼트 X, Y의 크기 N_{\min} 에서 N_{\max} 까지 변화하면서 각각의 벡터내적 $(X \cdot Y)$, Norm $|X|$ 와 $|Y|$ 를 계산하여 표 1에서와 같이 메모리에 저장한다. 그림 2와 같이 입력 데이터 버퍼를 할당하면 해석구간을 $2 N_{\max}$ 로 하고 매 번 데이터가 입력될 때 전체 데이터를 이동하지 않고 데이터 포인터 X_p 를 증가 시키면서 입력된 데이터를 저장한다. 데이터 포인터 X_p 는 $2 N_{\max}$ Modulo 연산을 한다.

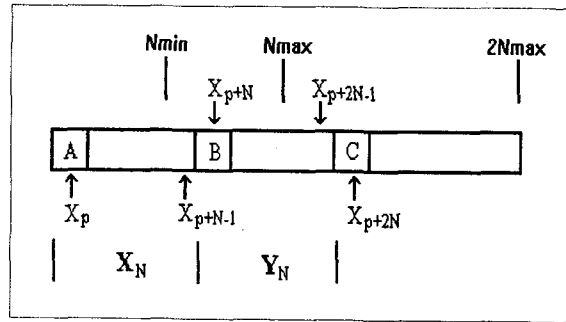


그림 2. 입력 데이터 버퍼 할당

Fig. 2. Assignment of input data buffer

$2 N_{\max}$ 입력 데이터 버퍼로부터 N_{\min} 에서 N_{\max} 까지 갱신된 값을 계산하는 수식은 다음과 같이 표현된다.

$$(X \cdot Y)'_N = (X \cdot Y)_N + X_{p+N}(X_{p+2N} - X_p) \quad (9)$$

$$|X|'_N = \sqrt{|X|_N^2 - X_p^2 + X_{p+N}^2} \quad (10)$$

$$|Y|'_N = \sqrt{|Y|_N^2 - X_{p+N}^2 + X_{p+2N}^2} \quad (11)$$

입력신호 한 샘플이 $2 N_{\max}$ 입력 버퍼에 들어 올 때 마다 식(9)에서 식(11)로부터 N_{\min} 에서 N_{\max} 까지 갱신한 값을 표1과 같이 메모리에 저장하고 식(8)로 각각의 상호상관계수를 구하고 그 중 가장 큰 값으로 태아 심음주기를 구한다.

태아심음주기를 찾는 실시간처리 알고리즘의 흐름도는 그림3과

같다. 입력 데이터 $2 N_{max}$ 의 해석구간에서 N_{min} 에서 N_{max} 까지 각각의 벡터내적 $(X \cdot Y)$, Norm $|X|$ 와 $|Y|$ 값을 메모리 할당 영역에 저장하고, 각각의 상관계수값을 구하여 그중 최대값을 갖는 N 값으로 초기의 심음주기를 찾는다. 이 후 새로운 입력 데이터가 들어오면 초기 심음주기를 구하는 것과 같이 벡터내적 $(X \cdot Y)$, Norm $|X|$ 와 $|Y|$ 를 모든 데이터에 대하여 값을 구하는 것이 아니라, 메모리에 할당된 각각의 값들을 이용하여 실시간처리 알고리즘에 따라 새로운 벡터내적 $(X \cdot Y)'$, Norm $|X|'$ 와 $|Y|'$ 값들을 각각 구하고 메모리에 Update 한다.

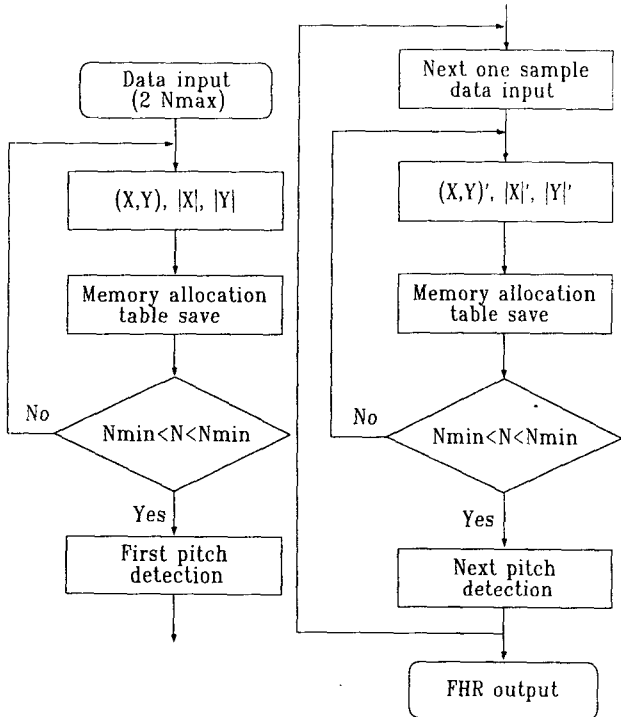


그림 3. 실시간 태아심음주기 추출 알고리즘의 흐름도
Fig. 3. Flowchart of real-time FHR detection algorithm

IV. 실험 및 검토

기존의 고해상 피치 알고리즘과 제안된 알고리즘 사이의 계산량을 비교하기 위하여 $2 N_{max}$ 데이터 구간에서 곱셈과 덧셈에 계산량을 각각 구한다. 두 벡터 세그먼트 X, Y의 Norm 과 상호상관계수의

		N_{min}	N_{min+1}	...	N_{max-1}	N_{max}
$(X \cdot Y)$	MUL	N_{min}	N_{min}	...	N_{max-1}	N_{max}
	ADD	N_{min}	N_{min}	...	N_{max-1}	N_{max}
$ X $	MUL	N_{min}	N_{min}	...	N_{max-1}	N_{max}
	ADD	$N_{min}-1$	$N_{min+1}-1$...	$N_{max-1}-1$	$N_{max}-1$
$ Y $	MUL	N_{min}	N_{min}	...	N_{max-1}	N_{max}
	ADD	$N_{min}-1$	$N_{min+1}-1$...	$N_{max-1}-1$	$N_{max}-1$

표2. 고해상 피치검출 알고리즘

Table 2. High resolution pitch detection algorithm

값은 두 세그먼트의 길이 N_{min} 에서 N_{max} 까지 기존 방법과 제안된 알고리즘을 표2와 표3에 나타냈다.

		N_{min}	N_{min+1}	...	N_{max-1}	N_{max}
$(X \cdot Y)$	MUL	1	1	...	1	1
	ADD	2	2	...	2	2
$ X $	MUL	3	3	...	3	3
	ADD	2	2	...	2	2
$ Y $	MUL	3	3	...	3	3
	ADD	2	2	...	2	2

표3. 실시간 고해상 피치검출 알고리즘

Table 3. Real-time high resolution pitch detection algorithm

표2와 표3에 나타난 것과 같이 기존의 알고리즘은 벡터 세그먼트가 증가함에 따라 연산량도 증가하는 반면, 실시간처리 알고리즘은 각각의 벡터 세그먼트에서 일정한 연산량을 갖으므로 실시간처리 알고리즘이 기존의 비해 계산량이 많이 줄어들음을 알 수 있다. 그러나 표2와 표3에 나타난 값은 한 해석구간($2 N_{max}$)에서만 비교한 값이며 샘플 입력이 계속될 때는 곱셈에 관한 비교는 그림4와 같다. 그림4에서 보는 바와 같이 기존의 알고리즘에 비해 제안된 실시간처리 알고리즘이 현저히 줄어들음을 알 수 있다.

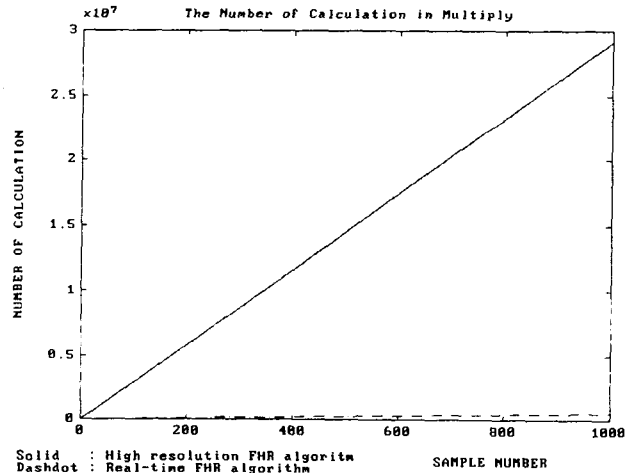


그림4. 연속 데이터의 곱셈 계산량

Fig. 4. Multiply calculation number of continuous data

V. 결론

본 연구에서는 기존의 태아심음 알고리즘의 문제점을 보완한 고해상 태아심음 추출의 실시간처리 알고리즘에 관하여 연구하였다. 이 알고리즘은 고해상 태아심음 알고리즘의 과도한 연산량을 효율적으로 줄일 수 있는 것으로 기존의 신호처리 전용프로세서를 이용하여 실시간처리가 가능하고, 현재 실시간처리를 위한 신호처리 전용 시스템 설계 및 제작이 연구 진행되고 있다. 데이터 메모리의 효율적 관리와 태아심음의 주기적 특성을 고려하여 고속 퍼스널컴퓨터에서도 실시간처리가 가능하도록 연구되어야 할 것이다.

참 고 문 헌

- [1] M. J. Ross et al., "Average magnitude difference function pitch extractor," *IEEE Trans., Acoust., Speech, Signal Processing*, vol. ASSP-22, pp.353-362, Oct. 1974.
- [2] L. R. Rabiner et al., "A coparative performance study of several pitch detection algorithms," *IEEE Trans., Acoust., Speech, Signal Processing*, vol. ASSP-24, no.5, Oct. 1976.
- [3] L. R. Rabiner, "On the use of autocorrelation analysis for pitch detection," *IEEE Trans., Acoust., Speech, Signal Processing*, vol. ASSP-25, no.1, Feb. 1977.
- [4] Divon MY et al., "Autocorrelation techniques in fetal monitoring," *Am J Obstet & Gynecol*, vol.151, pp2-6, 1985.
- [5] Amato JC, "Fetal heart rate monitoring," *Am J Obstet & Gynecol*, vol.147, pp.967-969, 1983.
- [6] Lawson GW, Belcher R, Dawes GS, Redman CWG, "A comparison of ultrasound(with autocorrelation) and direct electrocardiogram fetal heart rate detector system," *Am J Obstet & Gynecol*, vol.147, pp.721-722, 1982.
- [7] Favret AG, "Autocorrelation techniques applied to the fetal electrocardiogram," *In: Frommer PL, ed. Digest of the fourth international conferance on medical electronics*, Princeton, New Jersey, 1961.
- [8] Y. Medan et al., "Super resolution pitch determination of speech signals," *IEEE Trans., Signal Processing*, vol.39, no.1, pp.40-48, Jun. 1991.
- [9] Sun I. Kim, Eung G. Lee, Yong H. Lee and Doo S. Lee, "Fetal heart rate estimation using high resolution pitch detection algorithm," *Proceedings of the 15th anual international conference of the IEEE EMBS*, part 2, pp.783-784, oct, 1993.
- [10] 이 용 구, 이 두 수, "태아심음주기 검출을 위한 고해상 피치 알고리즘," *대한전자공학회 논문집 제31권 B편 제2호*, 대한전자공학회, pp.194-201, 1994.