

맵 기반의 부분시간 공통 중간주파수 제거방식을 이용한 GNSS 신호의 상관 기법

A GNSS Signal Correlation Using Map-based Partial-time Common Intermediate Frequency Removal Method

임 성 혁, 지 규 인*
(Sung-Hyuck Im and Gyu-In Jee)

Abstract : In this paper, we propose the efficient Doppler removal method using map-based partial-time common intermediate frequency removal technique. In the proposed algorithm, the 2-stage carrier removal process was used. First, the component of common intermediate frequency is removed. Next the component of Doppler was removed with averaging and approximation. For the evaluation of the proposed algorithm, The real-time software GPS L1 C/A-code receiver was implemented. When the proposed algorithms are used, 12 tracking channels with 3 track arm(early, prompt, late) is operated real-time on PC using a Intel Pentium-III 1.0GHz CPU. Also, the requirement of memory was less than 2Mbytes. The real-time software GNSS receiver using the proposed algorithms provides the navigation solution with below 10 meter rms error. Especially, in spite of using the various approximations for implementing the algorithms, the high sensitivity capability (able to track the weak signal with -159dBm) was achieved.

Keywords : software GNSS receiver, software correlator, map-based correlation, partial Doppler removal

I 서론

최근에 소프트웨어 GNSS 수신기라 불리는 하드웨어 기반의 상관기를 대체하여 소프트웨어방식으로 실시간으로 GNSS 신호의 기저대역 신호처리를 구현하는 연구가 활발히 진행되고 있다. 특히, 소형의 임베디드(embedded) 시스템에 탑재가 가능하도록 기저대역 신호처리를 위한 연산수와 메모리 요구량을 효과적으로 감소시키는 연구가 급속도로 증가하였다.

위에서 언급한 기저대역 신호처리를 효과적으로 감소시키는 연구 중에서 주목할 만한 알고리즘은 논리연산기반의 병렬처리(bit-wise parallel processing) 알고리즘이다[1]. 이 알고리즘은 GNSS RF front-end의 A/D(Analog-to-Digital)컨버터로부터 출력되는 비트열을 기존의 하드웨어 수신기가 샘플단위로 처리하던 방식을 탈피하여 병렬로 샘플을 결합한 후 미리 만들어진 기저대역 복사신호와 논리적 연산을 통하여 신호를 처리하는 방식이다.

하지만 이 방식은 두 가지의 단점이 있다. 첫째, 원하는 해상도의 기저대역 신호처리를 위해서는 맵의 저장을 위한 대형 메모리(수 메가바이트 이상)를 요구한다. 일반적인 IBM기반 PC에서는 큰 요구사항이 아니지만, 소형의 임베디드 시스템에서는 허용될 수 없는 메모리 요구량이다. 둘째, 개별적인 위성을 위한 채널간의 연산이 공유되지 않는다는 점이다. 즉, 채널간의 공통적인 연산이 없기 때문에 채널 수의 증가와 비례하여 연산량이 증가하게 된다.

위에서 언급된 두 가지 단점을 최소화한 새로운 소프트웨어 기반의 GNSS 기저대역 신호처리 방법을 제안하였다. 제안된 방법의 이름은 “맵 기반의 부분시간 중간주파수 제거방

식”이다. 이름에서 알 수 있는 바와 같이 제안된 알고리즘은 입력되는 신호열에 대해서 중간주파수(intermediate frequency)를 제거한 후 신호의 압축을 통한 처리효율을 증가시키는 방법으로 모든 채널에 공통으로 작용되는 부분을 1회의 맵핑 기능의 수행을 통해 고도의 처리시간 단축을 가능하게 하였다.

본 논문은 다음과 같이 구성되어 있다. GNSS 위성항법 신호의 구조와 기존의 처리 방식에 대해서 설명한 후 제안된 맵 기반의 부분시간 중간주파수 제거방식이 수식적으로 전개된다. 다음으로, 제안된 방식을 적용하여 실시간의 소프트웨어 GPS 수신기를 제작하여 실시간 처리성과 항법성능을 논리연산기반의 병렬처리 알고리즘[1]과 비교하여 평가하였다. 제안된 방법은 16.368Msps를 샘플링주파수를 가진 Front-end로부터 입력되는 신호에 대해서 실시간으로 12개 위성을 동시에 추적할 수 있다. 또한, 항법성능에서도 기존의 하드웨어 수신기와 동일한 성능을 도출함을 확인하였다.

II. GPS L1 C/A 신호의 일반적인 상관 방법

GNSS 신호가 RF Front-end를 통과하여 변환된 중간주파수 신호를 샘플링(sampling)하면 아래 식과 같이 표현될 수 있다.

$$\begin{aligned} s_k &= s_{IF}(t_k) \\ &= AC(t_k)D(t_k)\cos\left[2\pi\left(f_{IF} + \Delta f_k\right)\frac{k}{f_s} + \phi_0\right] + n(t_k) \\ &= AC_k D_k \cos\left[2\pi k \left(\frac{f_{IF}}{f_s} + \frac{\Delta f_k}{f_s}\right) + \phi_0\right] + n_k \\ &= AC_k D_k \cos[2\pi k N + \phi_k] + n_k \end{aligned} \quad (1)$$

$$\frac{f_{IF}}{f_s} = N, \quad 2\pi k \frac{\Delta f_k}{f_s} + \phi_0 = \phi_k$$

여기에서,

A : 신호의 진폭

* 책임저자(Corresponding Author)

논문접수 : 2007. 12. 31., 채택확정 : 2008. 4. 13.

임성혁, 지규인 : 건국대학교 전자정보통신공학과

(shim@konkuk.ac.kr/gijee@konkuk.ac.kr)

※ 본 연구는 교육자원부 BK21사업의 지원과 삼성전기 “소프트웨어 수신기” 개발과제에 의하여 연구되었음.

- C_k : k 샘플 시간에서의 코드
 D_k : k 샘플 시간에서의 데이터비트
 f_{IF} : 중간주파수 신호
 f_s : 샘플링 주파수
 Δf_k : k 샘플 시간에서의 수신기 clock 오차로 인한 도플러 및 위성파 수신기의 운동에 의한 도플러
 ϕ_0 : 초기 위상 오차
 n_k : 잡음

(1)에서와 같이 샘플링된 중간주파수 신호에서 도플러 성분을 제거하기 위해서 다음과 같이 반송파 복사 신호를 곱해야 한다.

반송파 복사 신호는 아래 식과 같이 표현할 수 있다.

$$r_k = \cos[2\pi kN + \hat{\phi}_k] + j \sin[2\pi kN + \hat{\phi}_k] \quad (2)$$

여기에서, $\hat{\phi}_k$ 는 추정된 위상 회전 값을 의미한다.

반송파 복사 신호가 곱해진 후의 신호는 다음과 같다.

$$\begin{aligned} s_k r_k &= AC_k D_k \cos[2\pi kN + \phi_k] \\ &\quad \left\{ \cos[2\pi kN + \hat{\phi}_k] + j \sin[2\pi kN + \hat{\phi}_k] \right\} \\ &= AC_k D_k \cos[2\pi kN + \phi_k] \cos[2\pi kN + \hat{\phi}_k] \\ &\quad + j AC_k D_k \cos[2\pi kN + \phi_k] \sin[2\pi kN + \hat{\phi}_k] \\ &= AC_k D_k \cos[\phi_k - \hat{\phi}_k] + j AC_k D_k \sin[\phi_k - \hat{\phi}_k] \end{aligned} \quad (3)$$

반송파 복사 신호가 곱해진 후의 신호를 I와 Q성분으로 분리하면 아래 (4)와 같이 표현될 수 있다.

$$\begin{aligned} I_k &= AC_k D_k \cos[\phi_k - \hat{\phi}_k] \\ Q_k &= AC_k D_k \sin[\phi_k - \hat{\phi}_k] \end{aligned} \quad (4)$$

반송파가 제거된 신호로부터 코드의 역-확산을 위해서 코드 복사신호(C_{rk})를 곱하는 과정은 다음과 같다.

$$\begin{aligned} I_k &= AC_k C_{rk} D_k \cos[\phi_k - \hat{\phi}_k] \\ Q_k &= AC_k C_{rk} D_k \sin[\phi_k - \hat{\phi}_k] \end{aligned} \quad (5)$$

코드 위상의 추적을 위해서 3개의 코드 복사신호를 입력 신호에 곱하면 다음과 같다. (5)의 I_k 와 Q_k 의 코드 복사신호에 대하여 코드 위상 추적을 위한 ‘ $\pm 1/2$ ’과 ‘0’ 칩의 위상차를 가진 코드 복사신호(early, prompt, late)를 생성하여 곱하면 (6)과 같이 여섯 개의 상관 값을 얻고 이것을 이용하여 코드 및 반송파 추적 루프를 이용하여 신호추적 및 항법해의 도출이 가능하게 된다.

$$\begin{aligned} I_{k,e} &= AC_k C_{rk,e} D_k \cos[\phi_k - \hat{\phi}_k] & Q_{k,e} &= AC_k C_{rk,e} D_k \sin[\phi_k - \hat{\phi}_k] \\ I_{k,p} &= AC_k C_{rk,p} D_k \cos[\phi_k - \hat{\phi}_k] & Q_{k,p} &= AC_k C_{rk,p} D_k \sin[\phi_k - \hat{\phi}_k] \\ I_{k,l} &= AC_k C_{rk,l} D_k \cos[\phi_k - \hat{\phi}_k] & Q_{k,l} &= AC_k C_{rk,l} D_k \sin[\phi_k - \hat{\phi}_k] \end{aligned} \quad (6)$$

여기에서, $C_{rk,e}$, $C_{rk,p}$, $C_{rk,l}$ 는 각각 Early, Prompt, Late 코드 복사신호를 의미한다.

위의 과정 중에서 (3)의 도플러 제거과정 때 샘플마다 코사인파와 사인신호를 연속적으로 만들어야 한다. 직관적으로 샘플링 주파수가 16.368MHz인 신호를 처리하는 경우에는 초당 약 16만번의 코사인과 사인신호를 생성해야고 각 채널이 독립적으로 수행하므로 고정된 C/A 코드열을 사용하는 소프트웨어 GPS 수신기를 구현하는 경우에 가장 많은 연산량을 차지하게 된다. 따라서, 이 과정을 간소화하는 경우 실시간 구현이 가능해진다. 다음 장에서 이 부분을 간소화 시키기 위하여 제안된 알고리즘에 대하여 설명하였다

III. 제안된 GPS L1 C/A 신호의 상관 방법

(2)에서 $2\pi kN$ 는 고정된 값이고 $\hat{\phi}_k$ 는 클록의 오차와 위성과 사용자 간의 운동에 의하여 발생한 도플러에 기인한 위상의 회전값이다. k 번째 샘플에서 $(k+1)$ 번째 샘플까지의 위상 변이는 아래 식과 같다.

$$\begin{aligned} [2\pi(k+1)N - 2\pi kN] &= 2\pi N \\ \hat{\phi}_{k+1} - \hat{\phi}_k &= 2\pi \frac{\Delta f}{f_s} \end{aligned} \quad (7)$$

일반적으로 도플러(= Δf)의 범위는 -10000Hz~10000Hz라고 할 수 있고, $\Delta f \ll f_s$ 이고 (7)은 아래 식과 같이 근사화할 수 있다.

$$\begin{aligned} [2\pi(k+1)N - 2\pi kN] &= 2\pi N \\ \hat{\phi}_{k+1} - \hat{\phi}_k &\approx 0 \end{aligned} \quad (8)$$

예를 들어 설명하면, 16.368Msps의 샘플링 주파수를 사용하고 T 가 8이고 도플러가 5000Hz를 갖는 경우에는 k 번째 샘플에서 $(k+T)$ 번째 샘플까지 위상의 변화 ($\hat{\phi}_{k+T} - \hat{\phi}_k = 2\pi T \Delta f / f_s$) 는 약 0.0154 라디안(radian)이다. 이는 ‘1’ 도(degree)보다 작은 위상 변화이며 무시할 수 있으므로 (7)을 (8)로 근사화 하는 것이 유효하다.

(8)을 사용하여 (3)에서의 순수한 중간주파수의 제거과정을 다시 쓰면 다음과 같다.

$$\begin{aligned} s_k r_{k,IF} &= AC_k D_k \cos[2\pi kN + \phi_k] \{ \cos[2\pi kN] + \sin[2\pi kN] \} \\ &= AC_k D_k \cos[2\pi kN + \phi_k] \cos[2\pi kN] \\ &\quad + j AC_k D_k \cos[2\pi kN + \phi_k] \sin[2\pi kN] \\ &= \{ AC_k D_k \cos[\phi_k] + j AC_k D_k \sin[\phi_k] \} \end{aligned} \quad (9)$$

여기에서, $r_{k,IF}$ 는 순수한 중간주파수에 의해 생성된 반송파 복사신호이다.

(9)에서 ϕ_k 에 의한 위상의 회전을 보상하기 위해서 T 샘플 동안의 위상 변화의 평균값을 보상하기 위하여 $\exp\left(\frac{1}{T} \sum_{i=0}^{T-1} \hat{\phi}_k\right)$ 를 (9)에 곱하면 아래의 식과 같이 변형된다.

$$s_k r_k = \{ AC_k D_k \cos[\phi_k] + j AC_k D_k \sin[\phi_k] \} e^{\frac{1}{T} \sum_{i=0}^{T-1} \hat{\phi}_k} \quad (10)$$

(10)은 기존의 도플러 제거과정에서 매 샘플마다 반송파 복사신호를 생성하는 과정을 T 개의 샘플마다 위상의 회전을 보상하기 위한 과정으로 변형된 형태를 나타낸다.

일반적으로 C/A 코드의 early-prompt-late의 3종의 복사신호와 샘플된 중간주파수 신호의 곱을 C/A 코드 prompt 복사신호와 입력되는 샘플된 중간주파수 신호의 개별적 지연의 곱으로 나타낼 수 있다. 이와 같은 변형은 복사신호를 메모리로부터 읽어오는 과정을 간소화하여 코드 역-확산 과정의 효율성을 증대시킨다. 따라서 (6)을 다음과 같이 변경할 수 있다.

$$\begin{aligned}
 I_{k,e} &= AC_{k+\Delta\tau} C_{rk,p} D_{k+\Delta\tau} \cos[\phi_{k+\Delta\tau} - \hat{\phi}_{k+\Delta\tau}] \\
 Q_{k,e} &= AC_{k+\Delta\tau} C_{rk,p} D_{k+\Delta\tau} \sin[\phi_{k+\Delta\tau} - \hat{\phi}_{k+\Delta\tau}] \\
 I_{k,p} &= AC_k C_{rk,p} D_k \cos[\phi_k - \hat{\phi}_k] \\
 Q_{k,p} &= AC_k C_{rk,p} D_k \sin[\phi_k - \hat{\phi}_k] \\
 I_{k,l} &= AC_{k-\Delta\tau} C_{rk,p} D_{k-\Delta\tau} \cos[\phi_{k-\Delta\tau} - \hat{\phi}_{k-\Delta\tau}] \\
 Q_{k,l} &= AC_{k-\Delta\tau} C_{rk,p} D_{k-\Delta\tau} \sin[\phi_{k-\Delta\tau} - \hat{\phi}_{k-\Delta\tau}]
 \end{aligned} \tag{11}$$

여기에서, $\Delta\tau$ 는 샘플간의 시간간격의 배수이다.

IV. 제안된 알고리즘을 이용한 소프트웨어 상관기 구현

제안된 상관 알고리즘을 실제 소프트웨어 수신기에 구현하기 위하여 몇 가지 구현 기법을 필요로 한다. 본 절에서는 제안된 상관 알고리즘을 특정 RF front-end로부터 입력되는 A/D 샘플 데이터에 적용하는 예를 통하여 구현 방법을 상세히 기술하여 이해가 쉽도록 하였다.

제안된 알고리즘을 이용하여 소프트웨어 GPS L1 C/A코드 수신기를 구현하기 위하여 사용된 RF front-end는 다음의 사양을 가지고 있다. 샘플링 주파수는 16.367MHz이고 중간주파수는 4.188MHz이며 2비트로 A/D 샘플된다. 2비트 A/D 샘플된 신호의 표현은 표 1과 같다.

표 1. A/D 샘플된 2비트 신호의 실제 값과 이진 데이터 표현 비교.

Table 1. Physical values and It's binary representation of A/D sampled 2-bit signal.

실제값	2비트 이진 데이터 표현
3	11
1	10
-3	01
-1	00

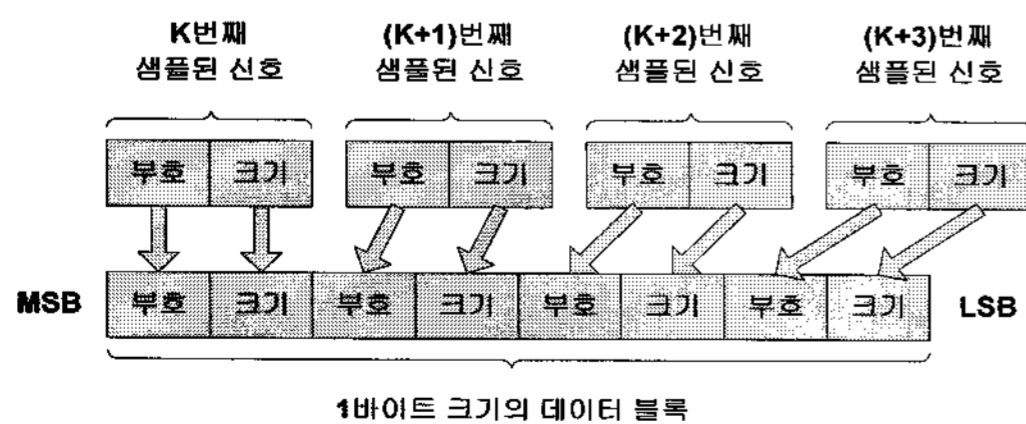


그림 1. 데이터 전송 효율을 증가시키기 위한 병렬화.
Fig. 1. Parallelization of serialized samples for increasing the transmission efficiency.

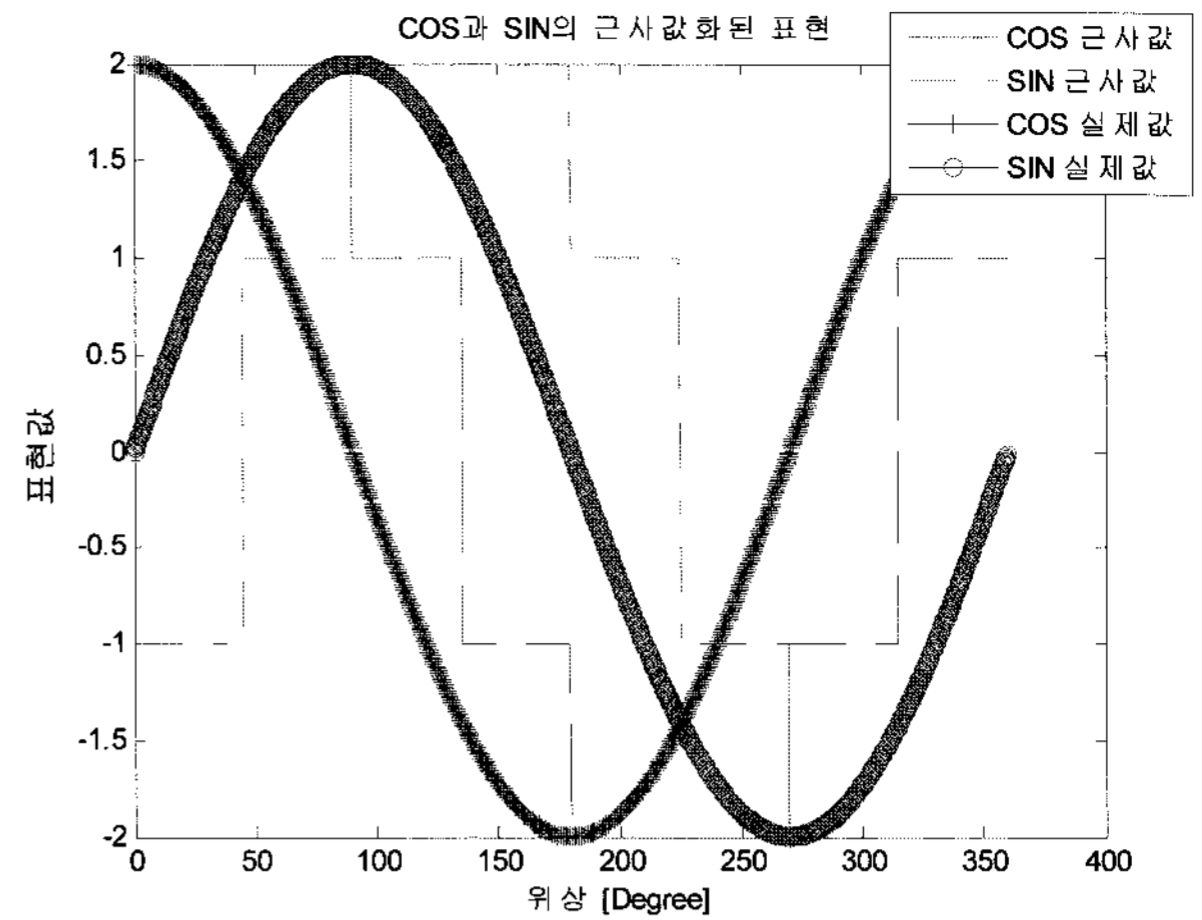


그림 2. 반송파 복사신호 생성을 위한 COS과 SIN의 근사값.
Fig. 2. Approximated values of the cosine and sine values for generating the carrier replicas.

표 2. 근사화된 COS과 SIN 값.

Table 2. Approximated cosine and sine values.

ϕ	0 ~45	45 ~90	90 ~135	135 ~180	180 ~225	225 ~270	270 ~315	315 ~360
$COS[\phi]$	2	2	1	-1	-2	-2	-1	1
$SIN[\phi]$	-1	1	2	2	1	-1	-2	-2

RF front-end로부터 host-PC로 샘플된 신호 전송의 효율을 높이기 위하여 데이터는 그림 1과 같이 신호를 병렬화하여 전송한다고 가정한다.

인터페이스를 통하여 입력된 신호는 앞에서 설명한 것과 같이 도플러 제거와 코드 역-확산 과정을 거치게 된다. 우선, 소프트웨어적인 구현의 편의성을 위하여 반송파 복사신호는 2bit의 크기를 갖는 8개 근사화된 값으로 표현할 수 있다. 앞에서와 같이 근사한 이유는 입력샘플의 비트수가 2비트이므로 반송파 복사신호도 2비트를 갖도록 하는 경우 성능 대 효율이 가장 우수하며, 2비트로 근사화된 값이 최소자승오차를 갖도록 값을 설정한다. 이와 같은 방법을 이용하여 근사화된 값은 그림 2와 표 2에 표현된다.

C/A코드 1주기인 1밀리초동안의 적산을 고려하면 (10)은 다음과 같이 다시 쓰여질 수 있다.

$$\begin{aligned}
 & \sum_{k=\alpha}^{\alpha+16366} s_k r_k \\
 &= \sum_{k=\alpha}^{\alpha+16366} \left\{ AC_k D_k \cos[2\pi kN + \phi_k] \cos[2\pi kN + \hat{\phi}_k] \right. \\
 & \quad \left. + AC_k D_k \cos[2\pi kN + \phi_k] \sin[2\pi kN + \hat{\phi}_k] \right\} \\
 &= \sum_{k=\alpha}^{\alpha+16366} \left\{ AC_k D_k \cos[2\pi kN + \phi_k] COS[2\pi kN] \right. \\
 & \quad \left. + AC_k D_k \cos[2\pi kN + \phi_k] SIN[2\pi kN] \right\} e^{j\hat{\phi}_k} \\
 &= \sum_{j=0}^{2045} \left\langle \sum_{i=0}^7 AC_{\chi+i} D_{\chi+i} \right. \\
 & \quad \left. \left\{ \cos[2\pi(\chi+i)N + \phi_{\chi+i}] COS[2\pi iN] \right. \right.
 \end{aligned} \tag{12}$$

$$+ \cos[2\pi(\chi+i)N + \phi_{\chi+i}] \sin[2\pi iN] \Big\} e^{j \sum_{i=0}^7 \phi_{\chi+i}}$$

* $\chi = \alpha + 8j$

여기에서, α 와 j 로 구성된 항을 표기의 편의를 위하여 χ 로 대체하여 명시한다.

(12)를 정리하면 아래 식과 같이 쓸 수 있다.

$$\sum_{k=\alpha}^{\alpha+16366} s_k r_k = \sum_{j=0}^{2045} \left\langle \sum_{i=0}^7 s_{\chi+i} \{ \cos[2\pi iN] + \sin[2\pi iN] \} \right\rangle e^{j \sum_{i=0}^7 \phi_{\chi+i}} \quad (13)$$

(13)에서 $\sum_{i=0}^7 \{ s_{\alpha+8j+i} \cos[2\pi iN] + s_{\alpha+8j+i} \sin[2\pi iN] \}$ 부분을

효율적으로 수행하기 위해서 맵을 이용하여 생성할 수 있다. 생성 방법은 다음과 같다. 채택된 RF front-end를 위해서 8샘플 동안의 $\cos[2\pi iN]$ 와 $\sin[2\pi iN]$ 의 값은 표 3과 같으며 그림 3은 이것을 도식한 것이다.

따라서, 입력되는 8개의 A/D 샘플과 고정된 부분적인 반송파 복사신호의 표현은 유한한 경우의 수를 갖게 되며, 그 경우의 수는 그림 4와 같이 169개로 표현된다.

그림 4는 8개의 샘플이 입력되었을 경우 고정된 반송파 복사신호와 곱해 나오는 169가지(-19-j19~19+j19)의 경우의 수를 갖는 결과를 보여주며, 이 과정은 맵을 통하여 간단하게 계산할 수 있다. 그림 5는 맵을 이용한 1차적인 중간주파수 제거과정이다. 맵을 생성할 때, -19-j19는 0으로 맵핑하고 -19-j18은 1로 맵핑하는 방법을 사용하여 169가지를 0~168의 숫자로 맵핑하여 맵을 생성한다. 그렇게 되면 1개의 변수를 이용하여 I와 Q값을 동시에 연산할 수 있다.

표 3. 고정된 부분적인 반송파 복사신호의 값.

Table 3. Values of the carrier replicas at fixed partial time.

i	0	1	2	3	4	5	6	7
$\cos[2\pi iN]$	2	1	-2	-1	2	1	-2	-1
$\sin[2\pi iN]$	-1	2	1	-2	-1	2	1	-2

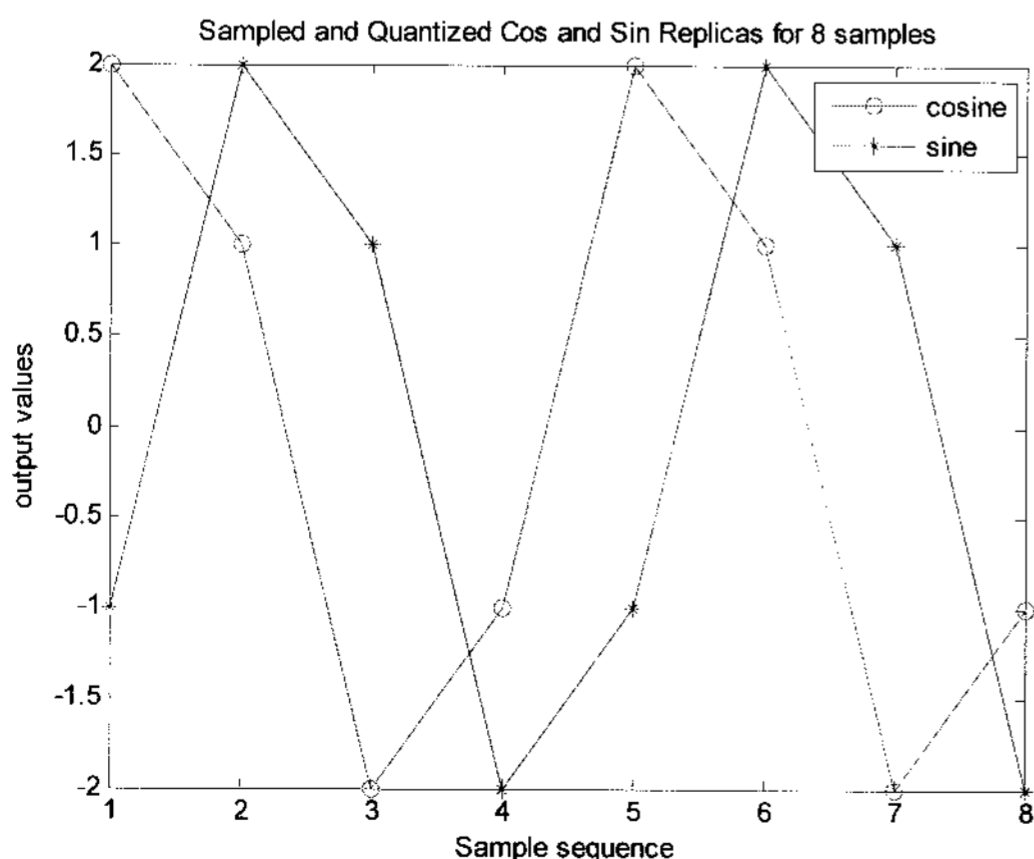


그림 3. 고정된 부분적인 반송파 복사신호의 표현.

Fig. 3. Presentations of the carrier replicas at fixed partial time.

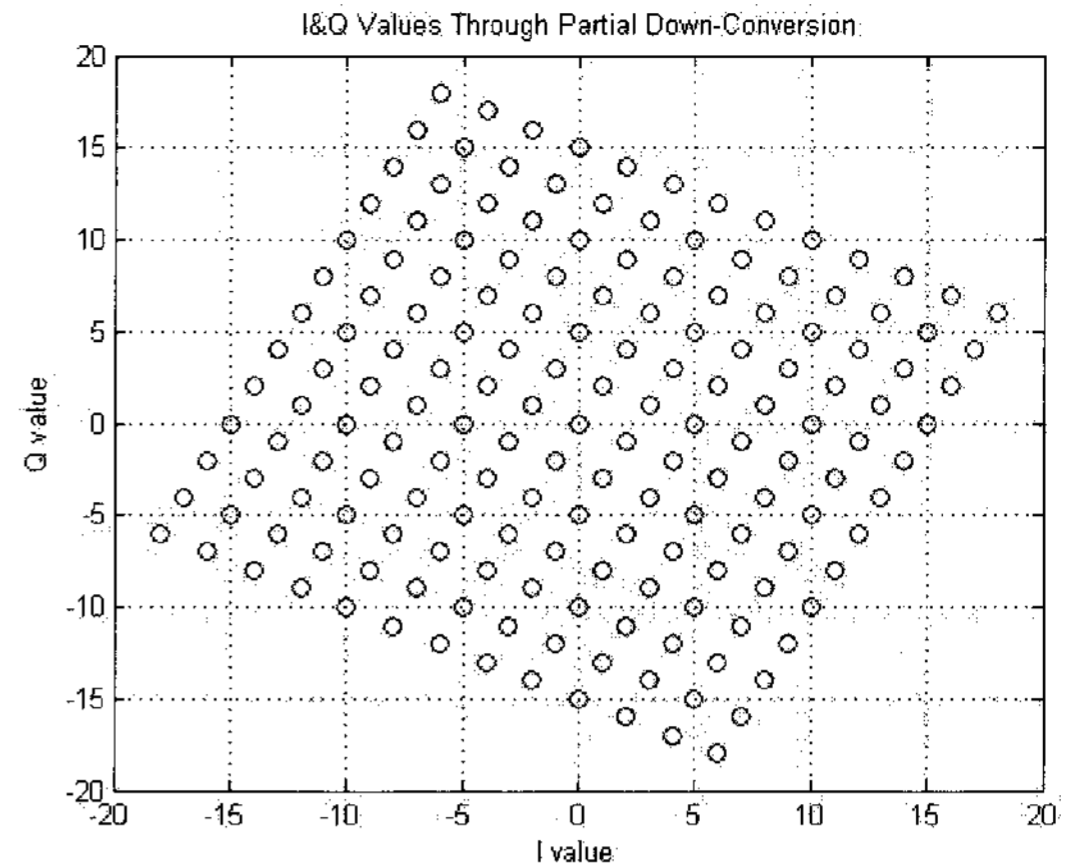


그림 4. 8샘플의 도플러가 제거된 후의 부분합.

Fig. 4. Partial summation when the 8 samples' Doppler is removed.

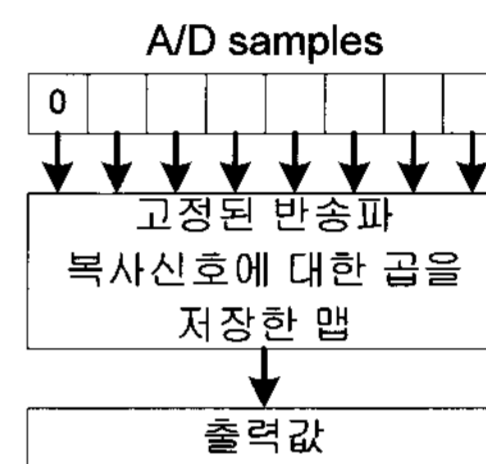


그림 5. 1차 중간주파수 제거과정.

Fig. 5. The process of the first removal of IF frequency.

그림 5의 과정은 다른 도플러를 가진 각각의 채널을 위해서 공통과정이며, 12채널을 사용할 경우 1회의 수행으로 12채널에 공통적으로 적용할 수 있다. 또한, 이 과정을 “부분적인 1차 중간주파수 제거과정”이라 지칭한다. 이 과정을 통해서 샘플링 주파수가 기존의 1/8로 감소하게 된다. 부분적인 1차 중간주파수 제거과정에 사용되는 맵을 $CR1[a,b]$ 로 지칭하고 이것이 의미하는 것은 a 번째부터 b 번째까지의 샘플에 의해서 지칭되는 출력값을 의미한다. 여기에서, $b-a=8$ 이다. $CR1[a,b]$ 맵은 입력신호가 2비트이고 8개의 샘플이므로 16비트로 표현이 가능하며, 요구되는 메모리는 출력이 1바이트(256가지수)로 지칭이 가능하므로 65536바이트이다. 이 과정을 거치면 (13)은 맵을 이용한 표현으로 다음과 같이 다시 쓸 수 있다.

$$\sum_{k=\alpha}^{\alpha+16366} s_k r_k = \sum_{j=0}^{2045} CR1[\alpha+8j, \alpha+8j+7] e^{j \sum_{i=0}^7 \phi_{\alpha+8j+i}} \quad (14)$$

$CR1[a,b]$ 맵이 지칭하는 가지 수는 169가지(8비트보다 작은 값)이며, (14)에서 8개 샘플의 평균 도플러 변화율을 맵을 통하여 생성할 수 있다. 맵을 위해서 7비트를 사용하게 되면 360/256도의 오차를 갖는 평균 도플러 변화율 보상을 위한 맵을 만들 수 있다. 이 과정을 2차 위상회전보상부분이라 지칭하며, 위에서 언급한 것과 같이 맵으로 만들 수 있다. 그림 6은 맵을 통해서 2차 위상회전보상을 수행하는 과정을 설명한 것이다. 2차 위상회전보상을 위해서 사용되는 맵을 $Rot[p][q]$ 라 지칭한다. 여기에서, p 는 $CR1[a,b]$ 에 의해서

출력되는 값이며, q 는 $\frac{1}{8} \sum_{l=0}^7 \hat{\phi}_{\alpha+8j+l}$ 에 의해서 지칭되는 값이

다. $\frac{1}{8} \sum_{l=0}^7 \hat{\phi}_{\alpha+8j+l}$ 계산의 효율을 위하여 반송파 복사신호에서

1밀리초 동안 $\hat{\phi}_{\alpha+8j+l} - \hat{\phi}_{\alpha+8j+l-1}$ 는 일정하므로 $\frac{1}{8} \sum_{l=0}^7 \hat{\phi}_{\alpha+8j+l}$ 를

$\hat{\phi}_{\alpha+8j+3}$ 으로 대체할 수 있다. 또한, $\hat{\phi}_{\alpha+8j+3}$ 를 자연수 표현으로

바꾸기 위해서 $round(256 \times \hat{\phi}_{\alpha+8j+3} / 2\pi) = \hat{\phi}_{fix,j}$ 로 표현할

수 있다. 이 식은 위에서 설명한 7비트를 이용하여 2차 위상 회전보상을 위한 맵을 생성하는 경우 해당된다. 이 과정을 포함하여 (14)를 다시 표현하면 다음과 같이 다시 쓸 수 있다.

$$\sum_{k=\alpha}^{\alpha+16366} s_k r_k = \sum_{j=0}^{2045} Rot[CR1[\alpha+8j, \alpha+8j+7]] [\hat{\phi}_{int,j}] \quad (15)$$

코드의 역-확산을 위해서 (15)를 변경할 수 있다. 코드는 1과 -1로 이루어져 있으며 $Rot[p][q]$ 맵에서 q 로 지칭되는 8 비트 중 최상위 비트에 코드의 1에 해당하는 1과 -1에 해당하는 0을 할당함으로써 가능하다. 최상위 비트에 1과 0을 할당하는 과정은 128을 더하거나 더하지 않는 과정과 같고 C/A 코드의 신호표현을 표 4와 같이 변경할 수 있다.

표 4의 표현값을 저장한 C/A코드의 맵을 $Cr_{-}[r]$ 로 지칭하며, 이 맵은 C/A 코드 한주기를 2.046MHz로 샘플링한 결과에 대하여 표4와 같은 표현값을 저장한 것이다.

C/A코드의 역확산 과정을 포함하여 (15)를 다시 표현하면 다음과 같다.

$$\begin{aligned} & \sum_{k=\alpha}^{\alpha+16366} s_k r_k \\ &= \sum_{j=0}^{2045} Rot[CR1[\alpha+8j, \alpha+8j+7]] [\hat{\phi}_{int,j} + Cr_{-}[j]] \end{aligned} \quad (16)$$

부분적인 1차 중간주파수 제거과정과 2차 위상회전보상과정 및 코드 역확산 과정을 포함하는 식은 (16)과 같이 표현될 수 있으며, 샘플링 주파수를 1/8로 감소시키며, 덧셈과 맵을 이용한 연산으로 표현될 수 있다. (16)을 (11)의 방법을 적용하여 Early-Prompt-Late 신호추적 값에 대해서 표현하면 다음과 같다.

$$\begin{aligned} Accum_{e,\alpha} &= \sum_{j=0}^{2045} Rot[CR1[\alpha+8j+1, \alpha+8j+7+1]] [\hat{\phi}_{int,j} + Cr_{-}[j]] \\ Accum_{p,\alpha} &= \sum_{j=0}^{2045} Rot[CR1[\alpha+8j, \alpha+8j+7]] [\hat{\phi}_{int,j} + Cr_{-}[j]] \\ Accum_{l,\alpha} &= \sum_{j=0}^{2045} Rot[CR1[\alpha+8j-1, \alpha+8j+7-1]] [\hat{\phi}_{int,j} + Cr_{-}[j]] \end{aligned} \quad (17)$$

(17)은 코드와 반송파의 추적 및 항법데이터를 획득하기 위하여 매 1밀리초마다 각각의 위성을 추적하기 위한 채널

표 4. C/A 코드의 실제값과 표현값.

Table 4. Physical values and representation of the C/A code.

C/A코드의 실제값	표현값
+1	128
-1	0

로부터 얻게 되는 I와 Q에 대한 상관값이다. I와 Q 값은 적산값의 상위 16비트와 하위 16비트에 분리되어 적산된다[5].

V. 구현된 알고리즘의 성능 평가

제안된 알고리즘을 앞 절에서와 같이 구현하여 실시간 소프트웨어 기반의 GPS L1 C/A 코드 수신기에 탑재한 후 성능을 평가하였다. 평가 기준은 다음과 같다. 종래의 방법인 논리연산기반의 병렬처리 기술과 상관기 동작을 위한 연산량에 대한 비교를 하였으며, 제안된 알고리즘이 탑재된 수신기가 제공하는 항법 성능을 평가하였다.

표 5에서와 같이 하드웨어기반의 상관기를 모사하여 소프트웨어로 구현한 경우와 논리연산기반 병렬처리 알고리즘의 연산량을 제시한 후 본 논문에서 제안된 알고리즘을 비교하였다. 비교 결과 제안된 알고리즘은 4개의 항목에서 가장 적은 연산량이 사용되었다. 특히 논리연산기반 병렬처리 알고리즘과 비교하는 경우 논리연산은 약 1/12, 덧셈연산은 약 1/2로 감소하였으며, 주소지정 역시 1/5로 감소하였다.

표 6과 같은 환경에서 실시간 성능평가를 수행하였다. 제안된 알고리즘을 사용한 결과 펜티엄III 1GHz CPU를 가진 시스템에서 12채널을 동작시키는데 약 80%의 리소스 점유율을

표 5. 알고리즘간의 연산량 비교.

Table 5. Comparison of calculations between algorithms.

구분	하드웨어 상관기 (3-Track Arm)	논리연산 병렬처리 (2-Track Arm)	제안된 알고리즘 (3-Track Arm)
논리연산	768	504	40
덧셈연산	1536	360	208
주소지정	768	372	74
곱셈연산	1536	-	-

표 6. 실시간 성능 평가를 위한 환경.

Table 6. Test set-up for the real-time performance.

구분	사양
운영체제	Windows XP
CPU	Pentium III 1GHz
RAM	512MB
GPS L1 RF Front-end	Nemerix NJ-1006 -Sampling Frequency : 16.367MHz -Intermediate Frequency : 4.188MHz -Quantization Bits : 2bits
데이터 전송장치	NI-6534 DAQ
데이터 전송방법	PCI
개발툴	Visual C++ 6.0
CPU의존 최적화 수행	None
위성 신호추적 채널 수	12 채널

표 7. 항법해 결과.

Table 7. Results of positioning.

구분	RMS 오차 (미터)	
	제안된 기법 사용	일반적인 기법 사용
East	2.76	3.04
North	3.34	2.96
Height	9.10	9.68

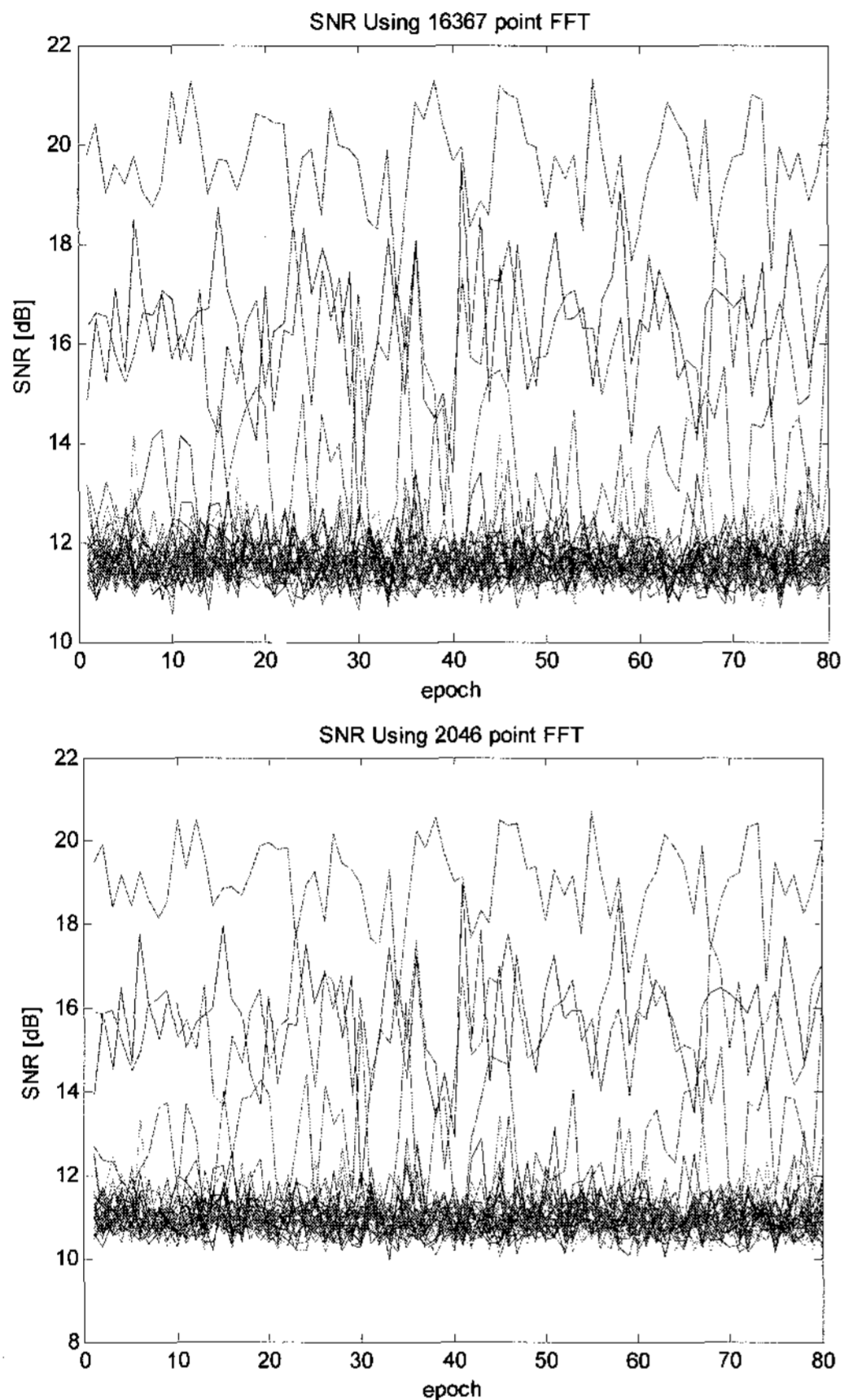


그림 6. 하드웨어 방식의 상관기와 제안된 상관기의 신호대 잡음비 비교.

Fig. 6. SNR comparison between a hardware-based correlator and the proposed correlator.

가지고 동작하는 것을 확인하였다. 실시간 동작을 위해서 구현된 프로그램은 최적화 없이 C와 C++를 혼용하여 사용하였으며, 윈도우XP 환경에서 테스트를 수행하였다.

그림 6은 제안된 알고리즘으로 상관을 수행한 경우의 신호 대 잡음비를 보여주고 있다. 기존의 상관 알고리즘(그림 6-상)과 비교하여 평균 0.6dB의 손실이 발생하는 것을 확인하였다. 하지만 많은 효율성을 위하여 근사화를 수행하여 나타난 결과인 것을 가만하면 0.6dB는 허용할 수 있는 신호의 감쇄이다.

제안된 알고리즘이 적용된 소프트웨어 GPS 수신기가 갖을 수 있는 고감도 성능을 평가하기 위하여 시뮬레이터를 이용하여 -159dBm까지 단계적으로 감소시켜 신호추적을 수행하였다. 그림 7은 신호추적 수행결과를 나타내며, 제안된 알고리즘이 적용된 GPS 수신기가 극도로 미약한 신호환경에서

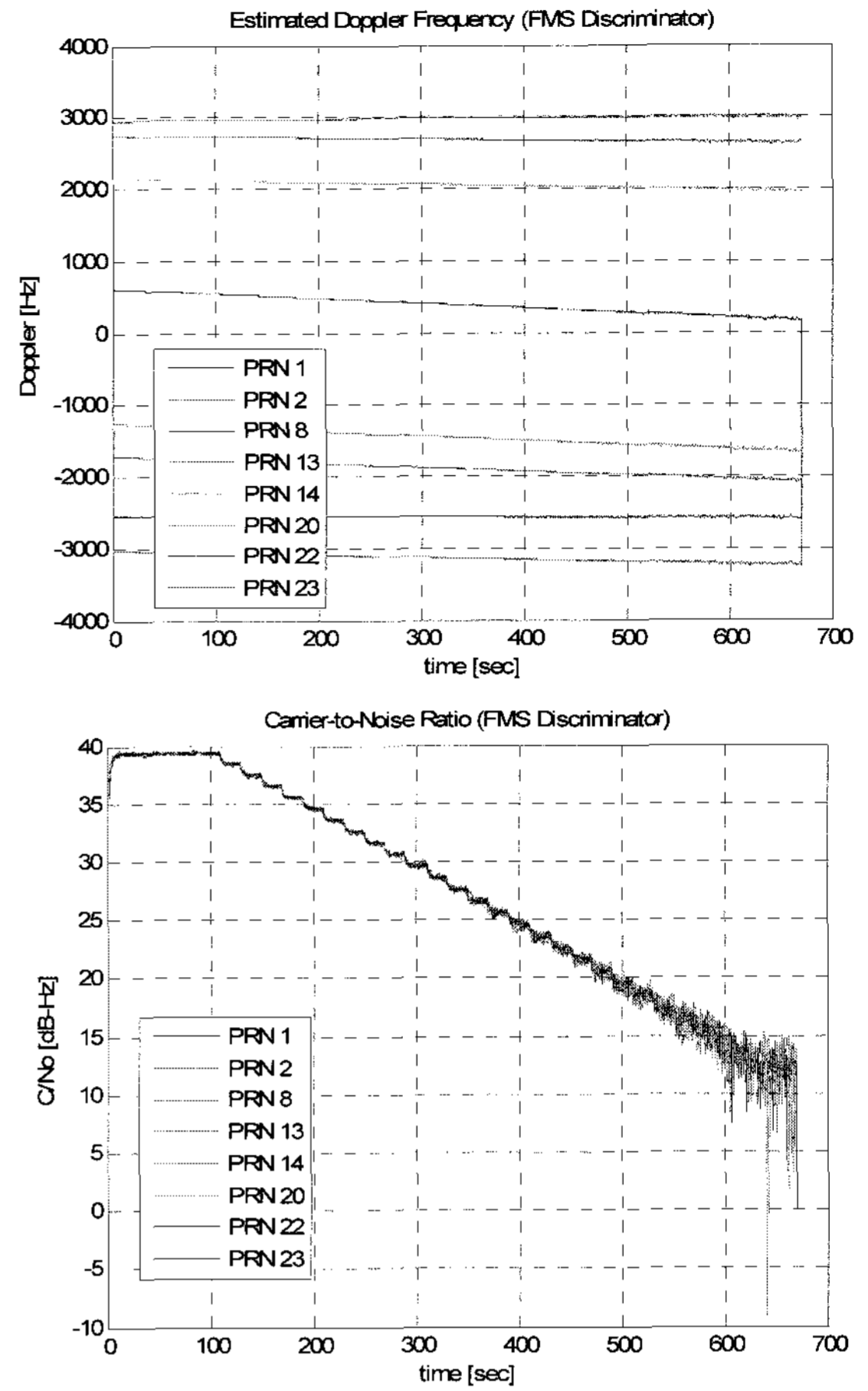


그림 7. 미약신호 추적 결과(상:도플러, 하:추정된 C/No).

Fig. 7. The results of tracking of weak signal(up:Doppler, down: estimated C/No).

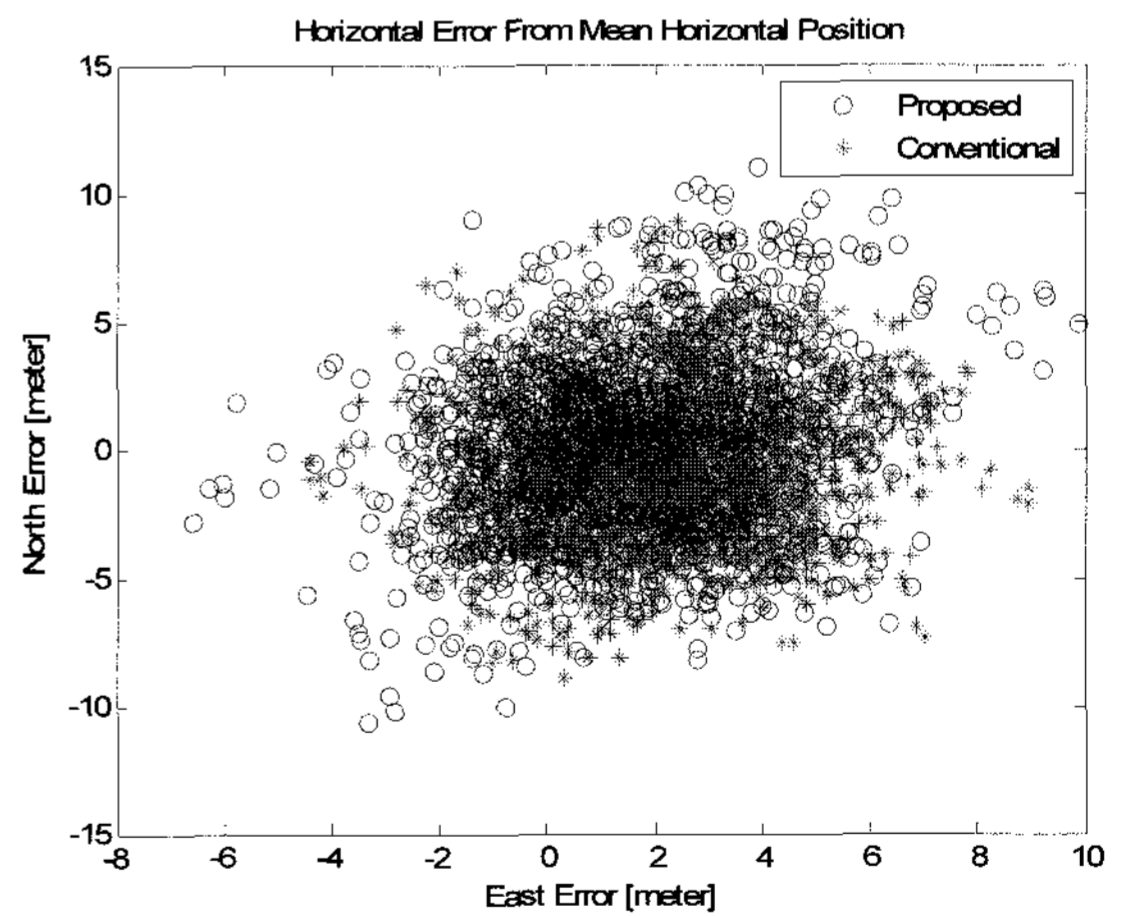


그림 8. 수평면 오차.

Fig. 8. Horizontal error.

사용될 수 있음을 확인하였다.

최종적으로 제안된 알고리즘을 이용하여 구현된 실시간 소프트웨어 GPS L1 C/A코드 수신기의 항법 성능을 평가하였다. 그림 8과 9은 구현된 수신기를 이용하여 계산된 항법해를 각각 수평면 오차와 수직 오차에 대해서 도식화 한 그림이다. 표 7은 제안된 방법이 사용된 수신기와 일반적인 하드

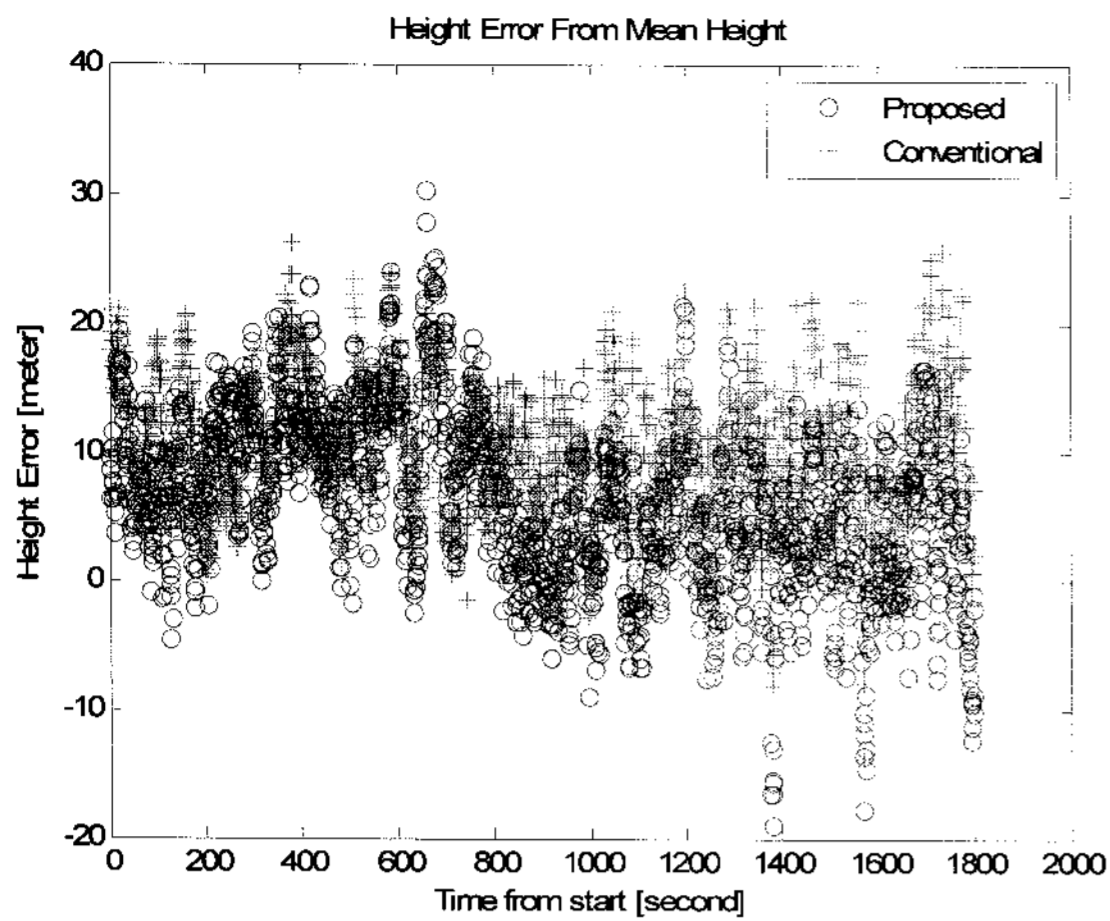


그림 9. 수직 오차.

Fig. 9. Height error.

웨어 상관기가 사용된 경우의 항법해 결과를 각각의 3차원 방향에 대한 RMS오차로 비교한 결과이다.

VI. 결론

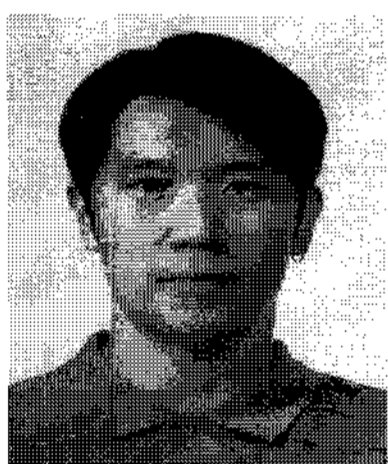
본 논문은 소형 임베디드 시스템에 적합하도록 소형의 메모리 요구량과 적은 연산량을 필요로 하는 위성항법 신호를 위한 소프트웨어 상관기 알고리즘(맵 기반의 부분시간 공통 중간주파수 제거방식)을 제안하고 실제 소프트웨어 수신기를

실시간으로 구현하여 평가하였다. 제안된 알고리즘은 2단계의 반송파 제거과정을 사용하여 채널간의 공통적인 부분을 개별적이 아닌 통합적으로 처리함으로써 효율성을 극대화시켰다.

평가결과 제안된 상관 알고리즘은 기존 하드웨어 기반의 상관 알고리즘과 비교하여 약 0.6dB의 신호 손실이 발생하였지만 항법성능은 유사하였으며, 특히 주목할만한 점은 많은 근사화를 통하여 효율성을 얻어냈음에도 불구하고 미약신호 성능에서도 상용 하드웨어 수신기가 도출하는 -159dBm의 신호추적 요구치를 제안된 알고리즘 또한 동일하게 도출함을 확인하였다.

참고문헌

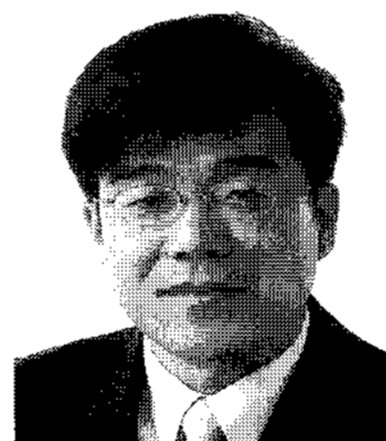
- [1] B. M. Ledvina, P. M. Kintner, and M.L. Psiak, "A 12-channel real-time GPS L1 software receiver," *ION GPS 2003*, 1990.
- [2] B. W. Parkinson and J. J. Spilker Jr, "GPS positioning system : Theory and application," *AIAA*, 1996.
- [3] E. D. Kaplan, *Understanding GPS : Principles and Applications*, Artech House Publisher, Norwood, MA, 1996.
- [4] J. B.-Y. Tsui, *Fundamentals of Global Positioning System Receivers : A Software Approach*, John Wiley&Sons INC., 2000.
- [5] S.-H. Im, G.-I. Jee, S.-D. Cho, and S.-J. Ko, "A novel software GPS receiver architecture using partial down-conversion," *ION 2007 National Technical Meeting*, 2007.



임 성 혁

2003년 건국대학교 전자정보통신공학과 졸업. 2005년 건국대학교 전자정보통신공학과(공학석사). 2006년~현재 건국대학교 대학원 전자정보통신공학과 박사과정 재학중. 관심분야는 소프트웨어 GPS, GPS anti-jamming, GPS 신호처리,

결합 항법.



지 규 인

1959년 11월 24일생. 1982년 서울대학교 제어계측공학과(공학사). 1984년 서울대학교 제어계측공학과(공학석사). 1989년 Case Western Reserve Univ. System and Control Engineering(공학박사). 1992년~현재 건국대학교 전자정보통신공학과 교수.

관심분야는 GPS/INS 결합항법, GPS 수신기 신호처리, 무선측위, 소프트웨어 GPS, GPS anti-jamming.