S & M 1502

Efficient Folding Processor for Direct Precision Code Acquisition in Global Positioning System Sensors

Keunhong Chae and Seokho Yoon*

College of Information and Communication Engineering, Sungkyunkwan University,
300 Cheoncheon-dong, Jangan-gu, Suwon-si, Gyeonggi-do 16419, Korea

(Received July 31, 2017; accepted November 2, 2017)

Keywords: GPS sensor, direct acquisition, P code, folding

Although the folding processor enables the global positioning system (GPS) sensor to directly acquire the correct phase of the received precision code (P code) without any prior information on the phase from auxiliary codes such as the coarse acquisition code, the output of the folding processor essentially involves considerable out-of-phase components as a result of multiple phases being used all together in the folding processor, and consequently, it could result in a poor acquisition performance. Thus, in this study, a novel folding processor is designed to reduce the number of out-of-phase components while maintaining the desirable in-phase components, thus achieving a better acquisition performance. In addition, the proposed folding processor does not require the parallel correlator used in the conventional folding processor, thus lowering the complexity in implementation. Numerical results demonstrate that the proposed folding processor provides a significant improvement in acquisition over the conventional folding processor, with a lower complexity in implementation.

1. Introduction

The global positioning system (GPS) provides location-based services based on the positioning signals transmitted from satellites.(1–3) To estimate the distance from the satellites, the GPS sensor performs code acquisition. In the GPS, the precision code (P code) is employed to provide a high degree of positioning accuracy owing to its long period (one week).(4,5) However, the long period of the P code makes its acquisition by the GPS sensor very challenging. Conventionally, P code acquisition is achieved using prior information on the phase from auxiliary codes such as the coarse acquisition code (C/A code). However, the C/A code is very vulnerable to channel distortions such as jamming and spoofing.

The advent of direct P code acquisition techniques, such as extended replica folding acquisition search technique (XFAST)(6) and dual folding (DF),(7) have obviated the need for the auxiliary codes in P code acquisition. In the XFAST and DF schemes, the special signal processor called the folding processor has been employed. "Folding" refers to the signal processing technique that divides the P code into multiple groups with a much shorter length

*Corresponding author: e-mail: syoon@skku.edu
http://dx.doi.org/10.18494/SAM.2018.1761

ISSN 0914-4935 © MYU K.K.
http://myukk.org/

than the period of the P code, and then, accumulates the entire group. The folding processor not only allows the GPS sensors to directly acquire the P code without any prior information, but also allows the GPS sensors to scan multiple code phases simultaneously with just a single operation. However, the folding processor essentially causes considerable out-of-phase components in the correlation outputs because of multiple phases being used all together in the folding processor. Since the out-of-phase components degrade the reliability of the threshold test that determines whether or not the P code is acquired, it could result in a poor acquisition performance. Although another direct P code acquisition technique called zero-padding (ZP),[8] which does not fold the P code, exists, ZP causes a loss of information. Thus, the acquisition performance of ZP is inferior to those of the folding-processor-based techniques.

Thus, in this paper, we propose a novel folding processor that reduces the number of out-of-phase components. In the conventional folding processor, folding is performed on both local and received codes, causing a considerable number of out-of-phase components. To solve this problem, we fold the local code only, significantly decreasing the number of out-of-phase components. However, the number of desirable in-phase components also decreases when folding is performed on only the local code. Thus, we obtain multiple correlation results and accumulate them to maintain the number of desirable in-phase components. Although the proposed idea can be enacted through the conventional folding processor using the parallel correlator, the parallel correlator increases the complexity in implementation. Thus, we present a folding processor architecture that does not require the parallel correlator. Reducing the number of out-of-phase components and obviating the need for the parallel correlator, the proposed folding processor can significantly improve the robustness of the GPS sensor to internal and external disturbances, and consequently, it allows us to employ relatively lower-cost-materials in making sensors to achieve a target performance (i.e., it provides us with a cost-effective sensor technology).

The rest of this paper is organized as follows. Section 2 describes the concept of the proposed folding processor. Section 3 describes the operation of the proposed folding processor, and presents the proposed folding processor architecture. Section 4 confirms that the proposed folding processor outperforms the conventional ones. Finally, Sect. 5 concludes this paper.

2. Concept of Proposed Folding Processor

Figure 1 illustrates the idea of the proposed folding processor based on the conventional folding processor architecture. First, the local P code generator generates the P code of an arbitrary phase. Then, the local P code is folded through the group divider and the adder. On the other side, the antenna and the low-pass filter (LPF) obtains the baseband received signal. Then, the sampler obtains discrete received samples, and the samples are divided into the same length with the folded local signal through the group divider for the correlation with the folded local signal. We generate a plurality of the local and received signals described above, and correlate the received signals with the local signals in parallel. Here, each correlator consists of two fast Fourier transform (FFT) modules, one inverse fast Fourier transform (IFFT) module, one multiplier, and one conjugate operator.[9] The outputs of the parallel correlator are

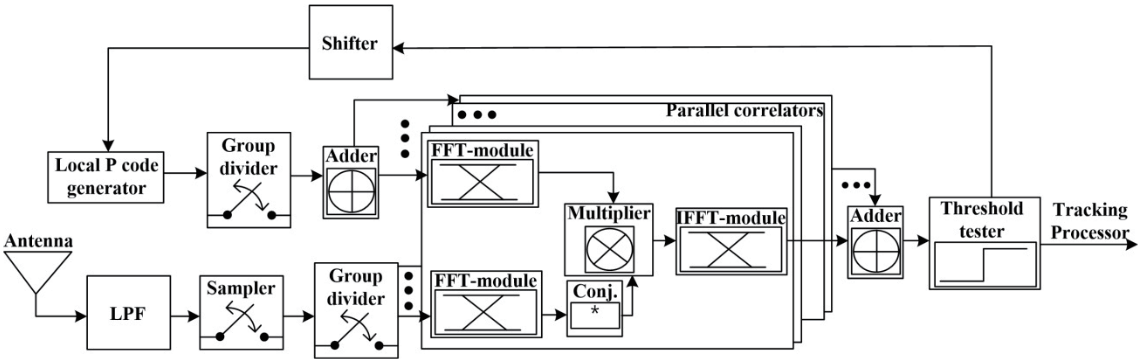

Fig. 1. Proposed idea implemented on the basis of the conventional folding processor.

accumulated to obtain the test statistic. The test statistic is compared with the predetermined threshold at the threshold tester. If the test statistic is larger than the threshold, it is determined that the P code is acquired. If the test statistic is smaller than the threshold, the process described above is repeated.

As shown in Fig. 1, we can see that the conventional folding processor requires the parallel correlator that increases the complexity in implementation. In Sect. 3, we describe the proposed idea in detail and present the proposed folding processor architecture without the parallel correlator.

3. Proposed Folding Processor

3.1 Signal model and FFT-based correlation

Before describing the proposed folding processor, we describe the signal model and the FFT-based correlation method. The ith sample of the received P code $\vec{x} = [x[0], x[1], x[2], ..., x[N-1]]$ is expressed as

$$x[i] = AC_R[i] \quad (i = 0, 1, ..., N-1), \tag{1}$$

where A is the amplitude of the received P code, N is the length of the received P code, and $C_R[i] \in \{-1, 1\}$ is the ith received P code chip.

The FFT-based correlation operation of two code vectors $\vec{a}$ and $\vec{b}$ is expressed as

$$\vec{R} = IFFT(FFT(\vec{a}) \cdot FFT^*(\vec{b})), \tag{2}$$

where $\vec{a} = [a[0], a[1], a[2], ..., a[N-1]]$, $\vec{b} = [b[0], b[1], b[2], ..., b[N-1]]$, $IFFT(*)$ and $FFT(*)$ mean the IFFT and FFT operations, $\vec{A} \cdot \vec{B}$ means the inner product operation of $\vec{A}$ and $\vec{B}$, and $()^*$ means the conjugate operation.[9]

3.2 Operation of conventional folding processor

In this section, we explain the operation of the conventional folding processor,(7) and show that the conventional folding processor causes a considerable number of out-of-phase components. In the conventional folding processor, the GPS sensor first generates the P code with the length of *MN*, where *M* is the number of local code folding. Then, the local P code is folded, yielding the folded code with the length of *N* as

$$f_j[i] = C_L[jW + i] + C_L[jW + N + i] + \ldots + C_L[jW + (M - 1)N + i], \tag{3}$$

where $f_j[i]$ is the ith sample of the folded local P code for the jth search, W is the phase difference between the local codes used in the current search and the subsequent search, and $C_L[i]$ means the ith P code chip of the local P code. Similarly, the received P code with the length of *KN* is also folded as

$$g[i] = AC_R[i] + AC_R[N + i] + \ldots + AC_R[(K - 1)N + i] \quad (i = 0,\ 1,\ \ldots,\ N - 1), \tag{4}$$

where $g[i]$ is the ith sample of the folded received P code $\vec{g}$, and K means the number of received code folding. Then, these two folded codes are correlated as

$$\vec{R}_j = IFFT(FFT(\vec{f}_j) \cdot FFT^*(\vec{g})). \tag{5}$$

Then, $\vec{R}_j$ is compared with the predetermined threshold to decide whether the received P code sequence is included in the local P code. Figure 2 shows an example of the operation of the conventional folding processor when $M = 3$ and $K = 2$. In Fig. 2, the folding processor can scan five code phases simultaneously, $[a_1\ b_1 \cdots c_2\ d_2]$, $[b_1\ c_1 \cdots d_2\ a_3]$, $[c_1\ d_1 \cdots a_3\ b_3]$, $[d_1\ a_2 \cdots b_3\ c_3]$, and $[a_2\ b_2 \cdots c_3\ d_3]$. The number of phases that can be scanned simultaneously is expressed with M and K as $MN–KN + 1$.(7) Thus, if the received code sequence is not included in the local

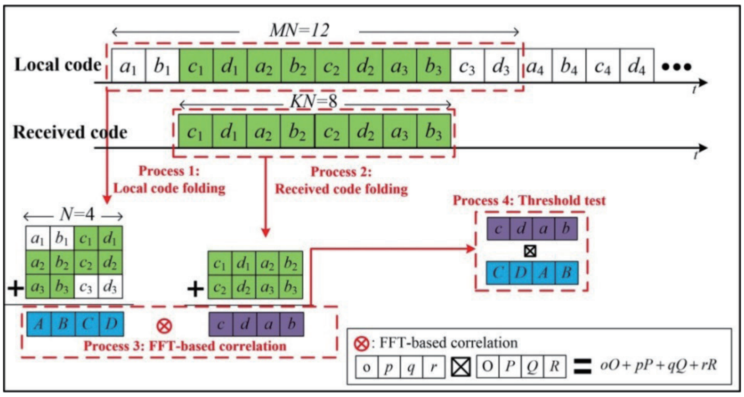

Fig. 2. (Color online) Operation of the conventional folding processor.

code, the local code is regenerated after shifting the phase of the local P code generator by $MN - KN + 1$ or less (i.e., $W = MN - KN + 1$ or less).

Figure 3 shows the in-phase and out-of-phase components in the correlation output of the conventional folding processor. As shown in Fig. 3, the correlation of the folded codes can be interpreted as the sum of the correlations of the unfolded codes which are represented by the red and green lines. Although the in-phase components, $\left(c_1^2 + d_1^2 + a_2^2 + b_2^2\right) + \left(c_2^2 + d_2^2 + a_3^2 + b_3^2\right) + \cdots + \left(c_4^2 + d_4^2 + a_5^2 + b_5^2\right)$, are included in the correlation output, it contains the much larger number of out-of-phase components.

3.3 Main idea of proposed folding processor

Figure 4 shows the main idea of the proposed folding processor. From Fig. 4, we can see that the number of out-of-phase components can be reduced significantly while maintaining the number of in-phase components compared with the case in Fig. 3, by folding on the local code only and accumulating multiple correlation results. The operation of the proposed folding processor is as follows.

In the proposed folding processor, the GPS sensor first generates the P code with the length of $(K' + 1)N$, where K' is the number of received code groups (i.e., $K' = 4$ in Fig. 4). The folding is performed on only the local code as

$$f'_{j,k}[i] = C_L[jW + (k-1)N + i] + C_L[jW + kN + i] \quad (i = 0,\ 1,\ ...,\ N-1), \tag{6}$$

where $\vec{f}'_{j,k}$ is the kth local code for the jth search.
The received code is evenly divided into K' groups with the length of N as

$$g'_k[i] = AC_R[(k-1)N + i] \quad (i = 0,\ 1,\ ...,\ N-1), \tag{7}$$

where $\vec{g}'_k$ is the kth received code. Then, the local and received codes are correlated as

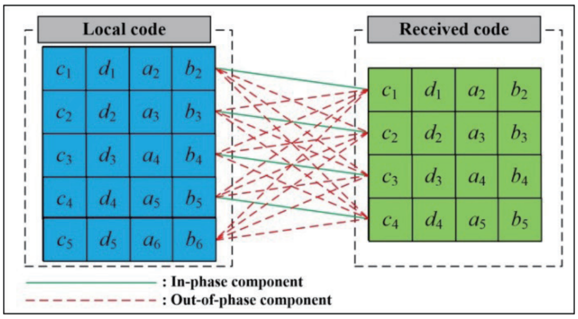

Fig. 3. (Color online) In-phase and out-of-phase components in the correlation output of the conventional folding processor.

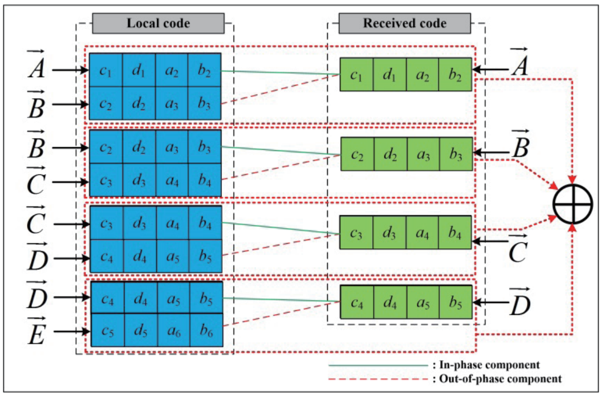

Fig. 4. (Color online) Main idea of the proposed folding processor.

$$\vec{R}'_{j,k} = IFFT(FFT(\vec{f}'_{j,k}) \cdot FFT^{*}(\vec{g}'_{k})), \tag{8}$$

where $\vec{R}'_{j,k}$ is the correlation result of the kth local code and the kth received code for the jth search. Finally, the entire results of $\vec{R}'_{j,k}$ are accumulated as

$$\vec{R}'^{pro}_{j} = \sum_{k=1}^{K'} \vec{R}'_{j,k}. \tag{9}$$

Then, $\vec{R}^{pro}_{j}$ is compared with the threshold to decide whether the received P code sequence is included in the local P code. Figure 5 shows the operation of the proposed folding processor when $K' = 2$. Since the proposed folding processor can be interpreted as the parallel execution of the conventional folding processor when $M = 2$ and $K = 1$, W can be expressed as $N + 1$ or less.

3.4 Proposed folding processor architecture

If the idea described in Sect. 3.3 is enacted with the conventional folding processor, the folding processor requires the parallel correlator, as shown in Fig. 1. Thus, we propose a novel folding processor architecture that does not require the parallel correlator. For the implementation of the proposed idea, we use the mathematical properties of FFT and IFFT:

$$FFT(X[k]) + FFT(Y[k]) = FFT(X[k] + Y[k]) \tag{10}$$

and

$$IFFT(X[k]) + IFFT(Y[k]) = IFFT(X[k] + Y[k]). \tag{11}$$

To explain the proposed folding processor architecture, we assume the correlation operation depicted in Fig. 4. As depicted in Fig. 4, we name the code sequences [c_1 d_1 a_2 b_2], [c_2 d_2 a_3 b_3] ...,

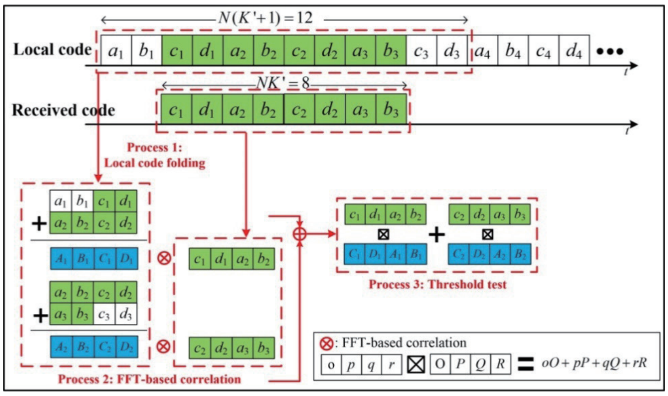
N(K'+1) = 12
Local code
a1 b1 c1 d1 a2 b2 c2 d2 a3 b3 c3 d3 a4 b4 c4 d4 •••
t
NK' = 8
Received code
c1 d1 a2 b2 c2 d2 a3 b3
t
Process 1: Local code folding
a1 b1 c1 d1
+ a2 b2 c2 d2
A1 B1 C1 D1
a2 b2 c2 d2
+ a3 b3 c3 d3
A2 B2 C2 D2
⊗
c1 d1 a2 b2
c2 d2 a3 b3
Process 2: FFT-based correlation
⊕
c1 d1 a2 b2 ⊠ C1 D1 A1 B1 + c2 d2 a3 b3 ⊠ C2 D2 A2 B2
Process 3: Threshold test
⊗: FFT-based correlation
o p q r ⊠ O P Q R = oO + pP + qQ + rR

Fig. 5. (Color online) Operation of the proposed folding processor.

$[c_5\, d_5\, a_6\, b_6]$ as $\vec{A}$, $\vec{B}$, ... , $\vec{E}$, respectively. The output of the folding processor can be represented as

$$\begin{aligned} &= IFFT\left[FFT(\vec{A}+\vec{B})\cdot FFT^*(\vec{A})\right] + IFFT\left[FFT(\vec{B}+\vec{C})\cdot FFT^*(\vec{B})\right] \\ &\quad + IFFT\left[FFT(\vec{C}+\vec{D})\cdot FFT^*(\vec{C})\right] + IFFT\left[FFT(\vec{D}+\vec{E})\cdot FFT^*(\vec{D})\right]. \end{aligned} \tag{12}$$

From Eqs. (10) and (11), Eq. (12) can be rewritten as

$$\begin{aligned} &= IFFT\left[\begin{array}{l} FFT(\vec{A}+\vec{B})\cdot FFT^*(\vec{A}) + FFT(\vec{B}+\vec{C})\cdot FFT^*(\vec{B}) \\ FFT(\vec{C}+\vec{D})\cdot FFT^*(\vec{C}) + FFT(\vec{D}+\vec{E})\cdot FFT^*(\vec{D}) \end{array}\right] \\ &= IFFT\left[\begin{array}{l} \left\{FFT(\vec{A}) + FFT(\vec{B})\right\}\cdot FFT^*(\vec{A}) + \left\{FFT(\vec{B}) + FFT(\vec{C})\right\}\cdot FFT^*(\vec{B}) \\ \left\{FFT(\vec{C}) + FFT(\vec{D})\right\}\cdot FFT^*(\vec{C}) + \left\{FFT(\vec{D}) + FFT(\vec{E})\right\}\cdot FFT^*(\vec{D}) \end{array}\right]. \end{aligned} \tag{13}$$

If the GPS sensors decide that the P code is not acquired, the local code is delayed and regenerated. If we set the phase delay as *N*, the FFT results used in the current search, $FFT(\vec{B})$, $FFT(\vec{C})$, $FFT(\vec{D})$, and $FFT(\vec{E})$, can be used one more time in the subsequent search as

$$\begin{aligned} &= IFFT\left[\begin{array}{l} FFT(\vec{B}+\vec{C})\cdot FFT^*(\vec{A}) + FFT(\vec{C}+\vec{D})\cdot FFT^*(\vec{B}) \\ FFT(\vec{D}+\vec{E})\cdot FFT^*(\vec{C}) + FFT(\vec{E}+\vec{F})\cdot FFT^*(\vec{D}) \end{array}\right] \\ &= IFFT\left[\begin{array}{l} \left\{FFT(\vec{B}) + FFT(\vec{C})\right\}\cdot FFT^*(\vec{A}) + \left\{FFT(\vec{C}) + FFT(\vec{D})\right\}\cdot FFT^*(\vec{B}) \\ \left\{FFT(\vec{D}) + FFT(\vec{E})\right\}\cdot FFT^*(\vec{C}) + \left\{FFT(\vec{E}) + FFT(\vec{F})\right\}\cdot FFT^*(\vec{D}) \end{array}\right]. \end{aligned} \tag{14}$$

where $\vec{F}$ is the local code vector subsequent to $\vec{E}$. From Eqs. (13) and (14), we can see that only one FFT operation [i.e., $FFT(\vec{F})$] and one IFFT operation are needed to perform the operation of Eq. (14).

Figure 6 shows the direct P code acquisition architecture incorporating the proposed folding processor. As depicted in Fig. 6, the conventional folding processor with the parallel correlator can be replaced by the proposed folding processor. In the proposed folding processor, the FFT is performed first as in Eq. (13) without the FFT after the addition (i.e., folding), as in Eq. (12). Since the FFT results are used repeatedly, they are stored in memory. After performing the FFT, the FFT results are accumulated or a conjugate of the FFT results is taken. Then, the multiplication, addition, and IFFT operations are performed sequently, as in Eq. (13), to obtain the test statistic.

4. Numerical Results

In this section, we compare the acquisition performances in terms of mean acquisition time (MAT) and mean phase delay (MPD) of the proposed direct P code acquisition scheme based on the proposed folding processor and the conventional schemes. In the simulation, we consider the following parameters: Code period of about 100 ms (10^5 chips), false alarm probability

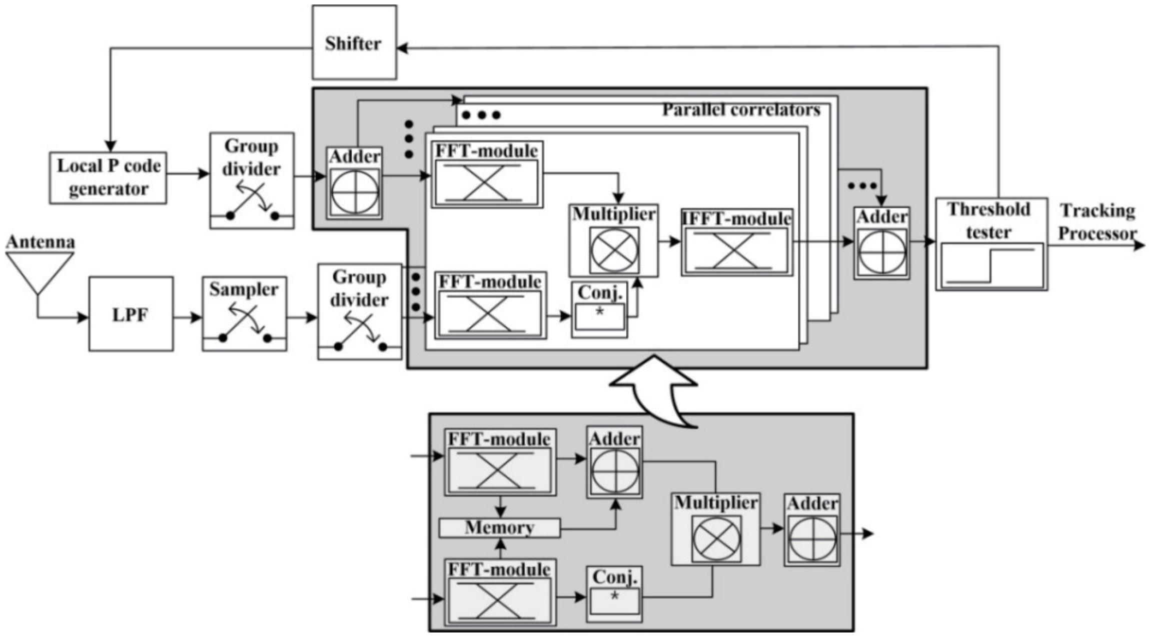

Fig. 6. Direct P code acquisition architecture incorporating the proposed folding processor.

$P_{fa} = 10^{-4}$, and $N = 1024$. For the FFT-based correlation, we set the time for one FFT operation as 100 ms.(10) For the conventional schemes, we set $M = 2$ for the XFAST and $M = 3$, $K = 2$ for the DF (here, it should be noted that the number of folding of the conventional schemes are set to have the same W with the proposed folding processor scheme), and the length of zeros in the ZP is set to $N/2$. For the proposed folding processor, K' is set to 10.

Figure 7 shows the MAT performances of the proposed and conventional schemes. Here, the MAT performance of the serial search scheme is included for reference. From the figure, we can clearly see that the proposed scheme gives a better MAT performance at low signal-to-noise ratios (SNRs) of −20–−10 dB than those of the conventional schemes. This is because the proposed folding processor reduces the number of out-of-phase components compared with that of the conventional folding processor. In addition, we can see that the proposed scheme shows a similar performance to the DF and the XFAST in the SNR range of −10–0 dB. This is because the conventional schemes can also acquire the P code correctly if the SNR is sufficiently high.

Figure 8 shows the MPD performances of the proposed and conventional schemes. In the simulation of Fig. 8, we set the maximum number of false alarms to 10 (i.e., even if the P code is not acquired, the code acquisition process is terminated if 10 false alarms occur). From the figure, we can clearly see that the proposed scheme acquires the P code more accurately than the conventional schemes in the SNR range of −22–−10 dB. This means that the proposed scheme is more likely to succeed in code acquisition than the conventional schemes.

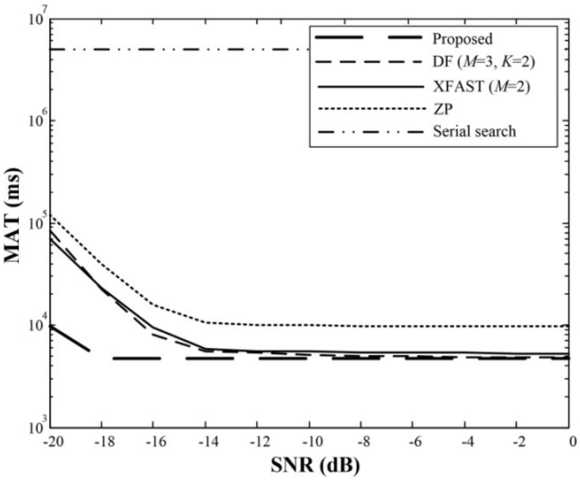

Fig. 7. MAT performance of the proposed and conventional schemes as a function of SNR.

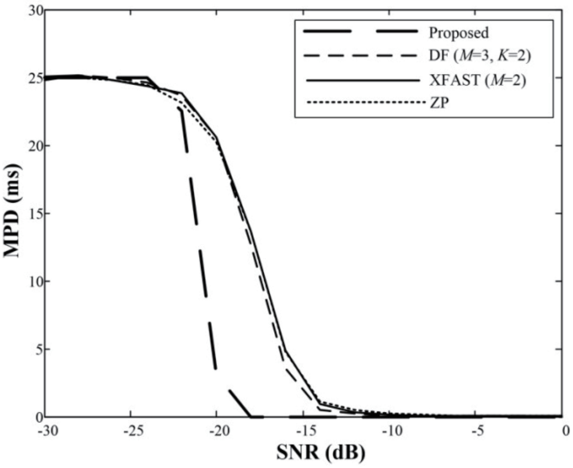

Fig. 8. MPD performance of the proposed and conventional schemes as a function of SNR.

5. Conclusions

In this paper, we proposed an efficient folding processor for direct P code acquisition. Specifically, we folded only the local code and accumulated the multiple correlation results to reduce the number of out-of-phase components and to maintain the number of in-phase components, respectively. For the implementation of the proposed folding processor, we designed a folding processor architecture that does not require a parallel correlator, and thus, results in lower implementation complexity. In numerical results, it has been confirmed that the use of the proposed folding processor significantly improves the acquisition performance.

Acknowledgments

This work was supported by Sungkyun Research Fund, Sungkyunkwan University, 2016.

References

1 B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins: Global Positioning System: Theory and Practice (Springer Science & Business Media, New York, 2012).
2 V. U. Zavorotny and E. Cardellach: IEEE Geosci. Remote Sens. Mag. **2** (2015) 8.
3 J. Nurmi, E. S. Lohan, S. Sand, and H. Hurskainen: GALILEO Positioning Technology (Springer, New York, 2015).
4 B. W. Parkinson and S. W. Gilbert: Proc. IEEE **71** (1983) 1177.
5 E. D. Kaplan and C. H. Hegarty: Understanding GPS: Principles and Applications (Artech House, Norwood, 2006).
6 C. Yang, J. Chaffee, J. Abel, and M. J. Vasquez: Proc. ION GPS 2000 (ION, 2000) 19.
7 H. Li, X. Cui, M. Lu, and Z. Feng: IEEE Trans. Wireless Commun. **7** (2008) 5286.
8 H. Li, M. Lu, and Z. Feng: IEEE Trans. Wireless Commun. **8** (2009) 2866.
9 D. J. R. Van Nee and A. J. R. M. Coenen: Electron. Lett. **27** (1991) 158.
10 P. Duhamel: IEEE Trans. Acoust. Speech Signal Process. **34** (1986) 285.

About the Authors

Keunhong Chae received his B.S.E. degree in information and communication engineering from Sungkyunkwan University, Suwon, Korea, in 2014. He is currently working toward his Ph.D. degree in the College of Information and Communication Engineering, Sungkyunkwan University, Suwon, Korea. His research interests include mobile communications, satellite communications, and detection and estimation theory.

Seokho Yoon received his B.S.E. (summa cum laude), M.S.E., and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1997, 1999, and 2002, respectively. From March 2002 to June 2002, he was with the Department of Electrical Engineering and Computer Sciences, Massachusetts Institute of Technology, Cambridge, MA, and from July 2002 to February 2003, he was with the Department of Electrical Engineering, Harvard University, Cambridge, MA, as a postdoctoral research fellow. In March 2003, he joined the College of Information and Communication Engineering, Sungkyunkwan University, Suwon, Korea, where he is currently a professor. His research interests include spread spectrum and orthogonal frequency division multiplexing systems, mobile/satellite communications, detection and estimation theory, and statistical signal processing. Dr. Yoon is a senior member of the Institute of Electrical and Electronics Engineers, a member of the Institute of Electronics, Information and Communication Engineers, and a lifetime member of the Korean Institute of Communication Sciences.