JPNT 7(4), 189-203 (2018)
https://doi.org/10.11003/JPNT.2018.7.4.189

A Design and Implementation of Software Defined Radio for Rapid Prototyping of GNSS Receiver

Kwi Woo Park[1], Jin-Mo Yang[2], Chansik Park[1†]

[1]Department of Electronics Engineering, Chungbuk National University / Research Institute for Computer and Information Communication, Cheongju 362-763, Korea
[2]Agency for Defense Development, Daejeon 305-600, Korea

ABSTRACT

In this paper, a Software Defined Radio (SDR) architecture was designed and implemented for rapid prototyping of GNSS receiver. The proposed SDR can receive various GNSS and direct sequence spread spectrum (DSSS) signals without software modification by expanded input parameters containing information of the desired signal. Input parameters include code information, center frequency, message format, etc. To receive various signal by parameter controlling, a correlator, a data bit extractor and a receiver channel were designed considering the expanded input parameters. In navigation signal processing, pseudorange was measured based on Coordinated Universal Time (UTC) and appropriate navigation message decoder was selected by message format of input parameter so that receiver position can be calculated even if SDR is set up various GNSS combination. To validate the proposed SDR, the software was implemented using C++, CUDA C based on GPU and USRP. Experimentation has confirmed that changing the input parameters allows GPS, GLONASS, and BDS satellite signals to be received. The precision of the position from implemented SDR were measured below 5 m (Circular Error Probability; CEP) for all scenarios. This means that the implemented SDR operated normally. The implemented SDR will be used in a variety of fields by allowing prototyping of various GNSS signal only by changing input parameters.

Keywords: software defined radio, GNSS receiver, multi-GNSS, rapid prototyping, implementation

1. INTRODUCTION

A variety of global navigation satellite systems (GNSSs) have been operated in recent years such as global positioning system (GPS) in the US, GLObal Navigation Satellite System (GLONASS) in Russia, Beidou Navigation Satellite System (BDS) in China, and Galileo in the European Union (EU). Each of the systems transmits signals with various modulation schemes to avoid the interference with the same band and adjacent band signals (Söderholm et al. 2016). Recently, Japan and India have developed a regional satellite navigation system to transmit satellite signals (GPSworld 2018). In addition, GPS and GLONASS are preparing system modernization to provide better services. Attention of the GNSS receiver based on software defined radio (SDR) has been increased with coming of multi-constellation environment which various satellite navigation systems exist. Most GNSS receivers are implemented with super-heterodyne type (Kaplan & Hegarty 2006). The receiver is divided into RF front-end that consists of down converter and analog to digital converter (ADC) and a digital signal processing unit that calculates navigation solution and processes baseband signals. The SDR is a receiver in which the signal processing unit uses the calculation by software based on general-purpose processors (Söderholm et al. 2016, Yang et al. 2017). The signal processing unit in a GNSS receiver is composed of the following functions: signal acquisition, signal tracking, bit synchronization and extraction, navigation data reconstruction, satellite position estimation, and receiver position estimation (Tsui 2005). To receive each

Received Sep 18, 2018 **Revised** Oct 23, 2018 **Accepted** Oct 25, 2018

[†]**Corresponding Author**
E-mail: chansp@cbnu.ac.kr
Tel: +82-43-261-3259 Fax: +82-43-268-2386

http://www.ipnt.or.kr Print ISSN: 2288-8187 Online ISSN: 2289-0866

of the GNSS signals, the algorithm of the signal processing unit should also be implemented suitably according to the corresponding signal structure. Since SDR can change the digital signal processing unit by software change, its maintenance is simple and development time can be shortened. More recently, as parallel processing techniques such as multi-core, graphic processing unit (GPU), and single instruction multiple data (SIMD) were applied, a number of development cases of real-time operable GNSS SDR have been published (Suzuki & Kubo 2014, Xu et al. 2016, Xu et al. 2017).

The Finnish Geospatial Research Institute (FGI) in Finland developed an SDR to compare the performance of navigation results estimated by the various combination of GNSS systems. The developed SDR supports the reception of GPS, Galileo, BDS and a fusion navigation using them (Söderholm et al. 2016). The Centre Tecnològic Telecomunicacions Catalunya (CTTC) in Spain applied SDR in the ionospheric scintillation and interference signal observation. The receiving band can be changed without hardware change using the universal software radio peripheral (USRP). Since most functions can be controlled by software, receivers can be remotely maintained. Observations that are difficult for human to access can be run efficiently through remote maintenance (Cristodaro et al. 2018). Tokyo University of Marine Science and Technology (TUMSAT) in Japan developed and published the GNSS-SDRLIB that added SDR functionality to the RTK-LIB that an open source library for precise point positioning (PPP). The development is underway, aiming for supporting various commercial GNSS RF front-end products and PPP algorithm tests (Suzuki & kubo 2014). Tsinghua University in China developed an anti-jamming performance evaluation platform based on their developed GNSS SDR. This platform can receive signals of GPS L1 C/A and BDS B3, and signals from eight antennas can be received through the eight-channel data acquisition device. The array antenna-based anti-jamming algorithms such as spatial time adaptive processing and spatial frequency adaptive processing were developed using the SDR platform and its performance evaluations were preformed (Xu et al. 2016).

The SDR in GNSS field has been utilized for algorithm development, evaluation, and observation. The reason for SDR's utilization in various studies was because it enabled a rapid prototyping of receivers with low cost (Goverdovsky et al. 2016). Prototyping is needed to measure the necessary cost for achieving the requirements of the final version and verify the performance. It is not restricted to development language or environment tools. Thus, accurate and fast modifications at low cost are required with regard to the requirements (Kordon & Luqi 2002). The SDR should be designed with more flexible structure to respond to the requirements and various objectives quickly.

This paper proposed and implemented SDR structure for rapid prototyping of GNSS receivers. The proposed SDR can receive various GNSS and DSSS signals without software modification by changing of input parameters containing the signal's structure information. These input parameters include Code information, center frequency, and navigation message format and so on. The receiver channel of SDR was designed to demodulate various GNSS and DSSS signals according to the input parameters. A pseudorange was measured based on coordinated universal time (UTC) to calculate a position even if the setup was set to receive different satellite systems, and a navigation signal processing module that selected navigation message decoder with the message type information was designed. To verify the validity of the proposed SDR structure, USRP and GPU-based hardware was configured, and the proposed SDR software was implemented. The GPS, GLONASS, and BDS satellite signals were received with various combinations by changing the implemented SDR's input parameters, and the correct operation and validity were verified through comparison of other SDR results and precision analysis of the measured positions.

2. CASE STUDY OF SDR DEVELOPMENTS

The GNSS SDRs developed by CTTC and TUMSAT are distributed as an open source (GitHub-taroz 2018, GNSS-SDR 2018). The meaning of input parameters of the other GNSS SDRs was analyzed by referring to the open source code and user manuals. The results can verified that which SDR could support a rapid prototyping without additional software development in various situations. Table 1 summarizes the input parameters that can control GNSS-SDR of CTTC and GNSS-SDRLIB of TUMSAT. The input parameters defined in each software are summarized by RF front-end, receive channel, and navigation signal processing categories.

The GNSS-SDR that utilizes USRP can control sampling frequency, local oscillator (LO) of down-converter, and filter coefficient. It has a high flexibility because it can receive signals from various bands using the same hardware according to input parameters. In contrast, for the GNSS-SDRLIB, there is a selection parameter of RF front-end type to supports various commercial GNSS RF front-ends. The GNSS-SDR of CTTC can respond more rapidly if it receives

Table 1. Input parameters of open source GNSS SDR.

	GNSS-SDR	GNSS-SDRLIB
RF Front-end	- Sampling frequency - Local oscillator frequency - Filter coefficient - Real or Complex	- Real or Complex - Front-end type
Receiver channel	- Number of L1 Channels - Number of E1B Channels - Noise bandwidth of PLL/DLL - Loop filter order - Integration time - Doppler search range - Doppler search step - Threshold	- PRN No. - System type - Code type - Noise bandwidth of PLL/DLL - Number of correlators - Chip spacing - Doppler search range - Doppler search step - Threshold
Navigation signal processing	- System selection - Average window size - Output rate - Display rate	- Output rate

GNSS or DSSS signals transmitted from the bands that commercial RF front-end does not support.

In GNSS-SDR, the numbers of GPS L1 and Galileo E1 are inputted when setting up the receive channel. The integration time, loop filter order, and noise bandwidth of the receiver channel can be controlled. In contrast, PRN No. system type, and code type are inputted in GNSS-SDRLIB. Users can also configure the channel in a detailed manner. Although the integration time during channel operation cannot be controlled, a loop filter order and noise bandwidth as well as the number of correlators and chip spacing of DLL can be controlled so that a multi-correlator configuration is possible. Both of the SDRs equally support the Doppler frequency search range and step, and threshold control in relation to signal acquisition. The above results imply that when using GNSS-SDRLIB, receive performance tests with more detailed scenarios could be possible.

The GNSS-SDR is more detailed in terms of navigation signal processing. In addition to navigation operation and data output cycles, selection results of systems preferred by users can be found as well. Moreover, GNSS-SDR can cope with the receiver's dynamic environments by controlling the integration time. On the other hand, only the output cycle can be set up in GNSS-SDRLIB.

The analysis results of the input parameters exhibit that both SDRs can perform a performance comparison according to the receive channel configuration. The performance evaluation according to RF front-end design such as Intermediate frequency (IF) and sampling frequency can be conducted without additional developments if GNSS-SDR is employed. The GNSS-SDRLIB is suitable to receive performance evaluation according to a multi-correlator.

More flexible structure of SDR can be designed if several parameters are added more to the input parameters supported by both SDRs. In particular, if signal's structure information such as code information, center frequency, modulation method, data rate, and navigation message format is manually inputted, the DSSS signals can be received via input parameter change without software modification. Signals of currently operating GNSS as well as new signals of satellite navigation systems that will be developed in future can be received with fewer cost and time. Fig. 1 shows the proposed configuration of GNSS SDR and types of input parameters.

Digital IF signals outputted in RF-front-end are computed by software. Here, RF front-end-related parameters are referred to the functions that manage signal data or control RF front-end. These parameters include sampling frequency, LO frequency of down-converter, amplifier gain, and real or complex signal selection. The receive performance according to various IF setups can be verified by controlling these four parameters (Park et al. 2017). Moreover, it can receive signals transmitted from other bands that are not operated by the current GNSS.

For the receiver channel, carrier signal and PRN code are removed and data bits are extracted from digital IF signals, which is signal demodulation. There are two DSSS methods used in satellite navigation systems: code division multiple access (CDMA) that uses different codes and frequency division multiple access (FDAM) that uses different frequencies. To receive DSSS without distinction in the receiver, users should directly input code information and center frequency. Thus, primary code, secondary code, code frequency and code length, signal's center frequency, data rate, and message format are added to the input parameter used in existing SDRs. Users can do prototyping of receivers on new signals quickly without software modification by inputting signal structure information to the receive channel in a more detailed manner.

The pseudorange and satellite position can be calculated by data bits, transmission and reception time extracted from N receiver channels. If the measurements from four or more satellites are obtained, receiver's position and time can be calculated (Kaplan & Hegarty 2006). This function is performed in the navigation signal processing unit. The mainly referred parameters are operation cycle, output rate, window size of moving average filter, and GNSS system selection. The core parts of the proposed SDR are the receiver channel whose flexibility is improved through the input parameter extension and the the navigation signal processing part that calculates a navigation solution using measurements in mutually different systems.

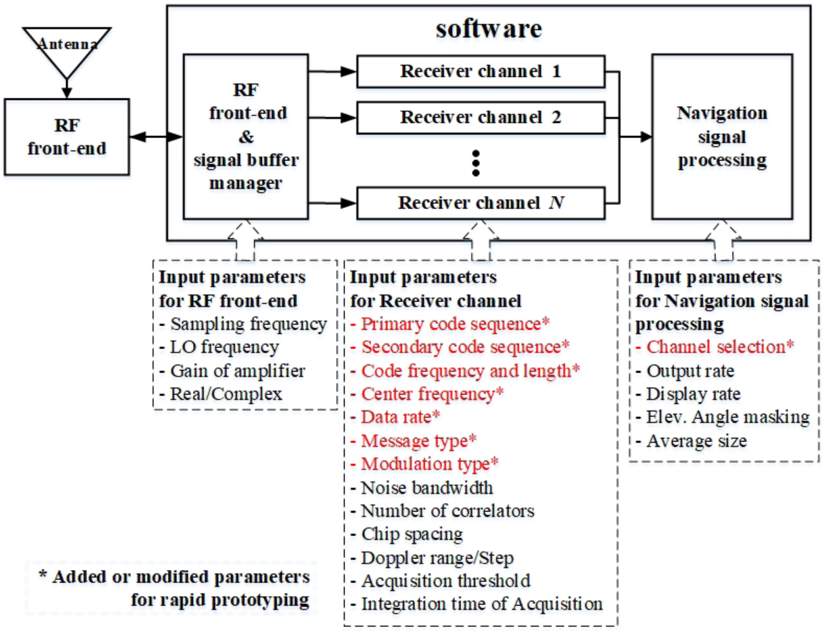

Fig. 1. A concept of proposed SDR and configuration of input parameters.

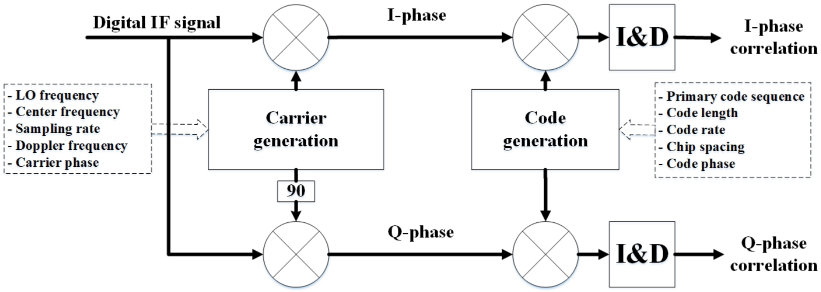

Fig. 2. A block diagram of correlator in signal acquisition and tracking.

3. DESIGN OF GNSS SDR CONSIDERING RAPID PROTOTYPING

3.1 Design of the Correlator

To acquire data from DSSS signals modulated with code and carrier, replica signals that have the same code phase, frequency, and carrier phase with those of the input signals should be generated, and a correlation between input signals and replica signals should be performed. If the code phase, carrier frequency, and carrier phase are matched, only the data included in the signals are left. However, noise, propagation delay, Doppler frequency, carrier phase, and code phase in transmitted signals vary over time. Methods to search and track these are called signal acquisition and signal tracking. Since signal acquisition and tracking are based on a correlation value, a design of correlator is important in design of SDR.

For the proposed SDR, the correlator should generate replica signals of the input signals using only parameter input regardless of a type of satellite system. Therefore code frequency, code length, and PRN code are entered by a user, there is no need to generate the code. However, it is necessary to re-arrange input PRN code considering the sampling frequency and measured code phase. Rearrangement function is performed in the code generator. In the meantime, the replica carrier is generated using the center frequency and LO frequency entered through user interface. The configuration is shown in Fig. 2

The carrier generator generates a replica carrier signal utilizing the inputted LO frequency f_{Lo}, center frequency f_r, sampling frequency f_s, measured Doppler frequency $\hat{f}_d$, and measured carrier phase $\hat{\theta}_d$. The process is presented in Eq. (1). The replica carrier at the carrier generator are produced as I-phase signal and 90º delayed Q-phase signals, which are employed to generate I-phase and Q-phase correlation

https://doi.org/10.11003/JPNT.2018.7.4.189

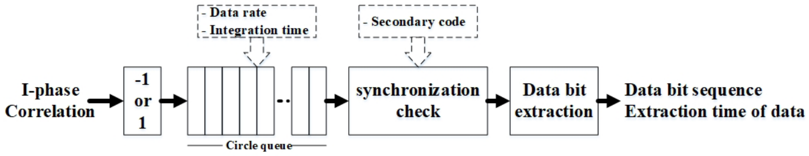

Fig. 3. A concept of bit synchronization for both tiered and non-tired signal

values, respectively.

$$r[k] = \cos(2\pi[f_r - f_{Lo} - \hat{f}_d]k / f_s + \hat{\theta}_r) \qquad (1)$$

The code generator generates replica signals by referring to the primary code information. The reason for the co-input of secondary code information in SDR is due to the consideration of tiered code. The secondary code is designed to include code one-chip to the primary code's one cycle. If a correlator is operated at one code cycle, there is no need to consider the secondary code. The secondary code is not needed to be considered if duration without sign change is integrated when the coherent integration time is increased. For example, coherent integration can be done using 5 msec signal for the NH code with consecutive "1" values for 5 msec (Bhuiyan et al. 2014). By utilizing code length l_c, code frequency f_c, and measured code phase $\hat{\theta}_c$, the code generator searches for the index j of PRN code sequence Γ that corresponds to k. Here, j can be calculated by using Eq. (2).

$$j = \mathrm{mod}(l_c,\ \mathrm{ceil}(\hat{f}_c k / f_s + \delta p \theta_{offset})) \qquad (2)$$

In Eq. (2), $\hat{f}_c$ refers to the measured code frequency, which can be calculated utilizing the measured code phase. A ratio between code frequency and sampling frequency refers to a frequency per signal sample, and by multiplying k, it can be the code position. The early, prompt, and late signals of the code tracking loop can be generated by adding the multiplying result of the next correlator distance δp and current correlator's offset θ_{offset}. In Eq. (2), mod refers to a modulo operator and ceil refers to a rounding-up operator. Using j calculated by the above equation, the code that corresponds to k in the inputted primary code can be acquired through $c[k] = \Gamma[j]$. $r[k]$ produced from the carrier generator, $c[k]$ produced from the code generator, and input signal become a correlation value through Integration and Dump (I&D). The calculated correlation value is utilized in bit synchronization, extraction, and signal tracking.

3.2 Bit Synchronization and Extraction Design

The I-phase correlation value produced from the correlator becomes baseband signal when the phase and frequency between input and replica signals is matched. Generally, data rate is slower than the coherent integration time of correlator. Thus, a single data bit is composed of many I-phase correlation values. For example, 20 correlation values become a single data bit when correlating GPS signals that transmit data at 50 bps with 1 msec coherent integration time. As described in the above, a process of conversion of correlation value into data bit is called bit extraction, in which data rate and coherent integration time should be considered.

When the baseband signal is produced for the first time as a result of signal phase and frequency match, the receiver does not know the starting point of the data bit. To find the starting point of the data bit, correlation values are collected as many as the data transmission cycle from the point where the sign of data bit changes. If the sign of the acquired correlation values are matched and the sign changes according to the data rate, it means bit synchronization occurs normally. Data bits are extracted by checking the correlation value and the sign from the starting point once the bit synchronization occurs.

There are signals of BDS B1I D1NAV and Galileo E1 that employ the tiered structure, and signals of GPS L1 C/A and GLONASS L1 that employ the non-tiered structure. In the previous correlator, a correlation value was calculated without consideration of secondary code. In this circumstance, the baseband signals that make up a single data bit of B1I D1NAV are represented with a sequence of Neuman Hoffman (NH) code, which is a secondary code. The non-tiered signal is the same if the secondary code is "1". Considering this, the bit extractor was designed as shown in Fig. 3.

The core of the designed bit synchronization is the collection of correlation value's signs using the circle queue, and correlation between the collected correlation values and secondary code. The GNSS signals that transmit binary data such as 1 or -1 can verify bit signs through the sign of the I-phase correlation value. To find the starting point of the bit, a size of the circle queue is determined using data rate and coherent integration time. When the data is full in the circle queue, it means the baseband signals that

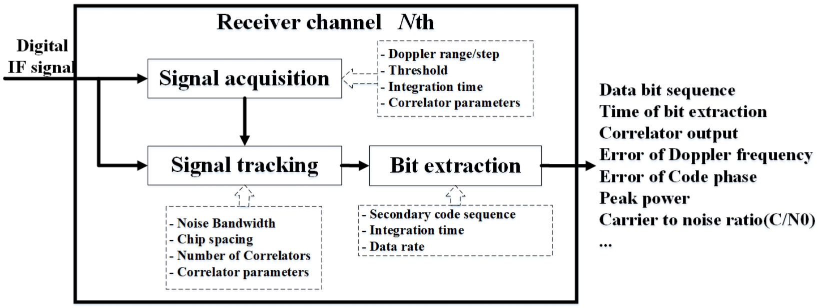

Fig. 4. A block diagram of receiver channel considering input parameters.

make up a single data bit are collected. To verify whether synchronization is complete, user inputted secondary code and bit sequence stored in the circle queue are correlated. If the correlation size is the same as the circle queue size, it means that bit synchronization is complete, and the sign indicates the data sign. Then, the data are restored by gathering the output of the I-phase correlation values from the starting point. The time to take data reconstruction is also recorded. Here, time starts from the start of the receiver operation. The measured time information is used to measure pseudorange between satellites and receiver.

3.3 Design of the Receiver Channel

From a single receiver channel, demodulation of single satellite signal is performed. Without prior information, signal acquisition that searches for approximate Doppler frequency and code phase, signal tracking that tracks Doppler frequency, carrier phase, and code phase precisely, and bit extraction and synchronization function (Tsui 2005). Therefore, the design of receiver channel includes the aforementioned correlator, bit synchronization, and extractor. For the receiver channel, a signal of single satellite is demodulated based on user-configured information from the inputted IF signal. As a result, navigation message, bit reception time, correlation value of each correlator, Doppler tracking error, code phase tracking error, and carrier to noise ratio (C/N0) are calculated. The above information can be categorized into information required for receiver position and time estimation, and information that can verify the receive status of satellite signals. Fig. 4 shows the receive channel designed considering the proposed input parameters.

The IF signals are inputted to signal acquisition and signal tracking units, respectively. With prior information available, changing Doppler and code phases are measured through signal tracking, and I-phase correlation values are produced. Without prior information, approximate Doppler frequency and code phase are searched through signal acquisition. Here, search range of Doppler frequency, search step, threshold that determines whether signal is detected, and coherent integration time of correlator are inputted. The search range of Doppler frequency may vary depending on the dynamic environment of receiver, but it does not exceed 10 kHz (Kaplan & Hegarty 2006). The Doppler frequency search step is closely related to a coherent integration time of correlator. If a coherent integration time is set to 1 msec, Doppler frequency should be searched in every 500 Hz interval, and if it is set to 20 msec, the search should be done at least in every 25 Hz interval. Most SDR-based receivers are designed to control this by user input. The integration time is closely related to the receive sensitivity, and with the increase in integration time, the receive sensitivity improves, but the calculation time increases (Tsui 2005).

During signal tracking, changes in Doppler frequency and code phase are measured utilizing PLL and DLL. Here, the initial value can be obtained through signal acquisition. PLL and DLL are composed of a discriminator that measures an error between input signal and replica signal, a loop filter, and a correlator. The noise bandwidth is a loop filter-related parameter, which is related to convergence rate and jitter during signal tracking. The chip distance and the number of correlators are related to a code tracking loop using DLL. Generally, three correlators (early, prompt, and late) are used to calculate an error of code phase in general GNSS receivers. Early and prompt correlators, and late and prompt correlators are generally set up with 0.5 chip spacing (Tsui 2005). If a chip spacing is narrower, it becomes a code tracking loop of narrow correlator mode. If more correlators are utilized to measure a code phase error, it becomes multi-correlator mode. If a multi-correlator is used, it is known to be robust to multi-path errors (Kaplan & Hegarty 2006).

The I-phase correlation value calculated in signal tracking is transferred to bit synchronization and extraction units. If bit synchronization occurs normally, it produces a data bit that makes up the navigation message. *N* measurements can be obtained from *N* receiver channels

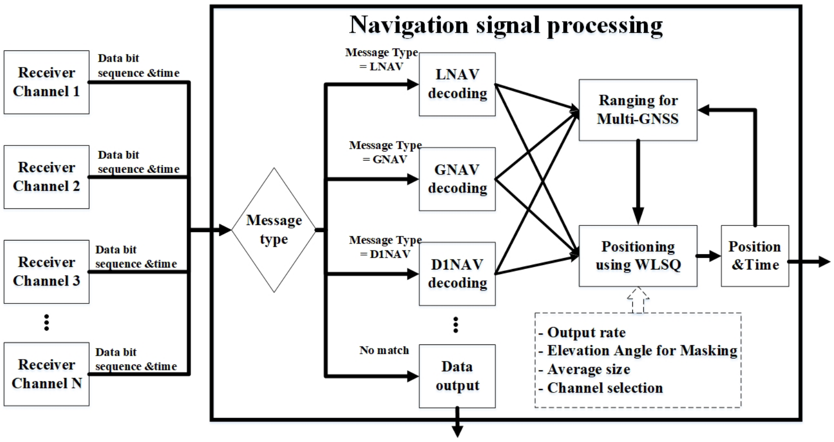

Fig. 5. A block diagram of positioning module.

configured by a user. The navigation signal processing unit in the receiver collects measurements from each receiver channel and performs operations required for receiver position estimation. The detailed explanation is described in the next section.

3.4 Design of Navigation Signal Processing

If the GNSS receiver acquired measurements from four or more receiver channels, a receiver position can be calculated. The measurements here refer to a data bit demodulated from the signal and reception time when the data bit is extracted. The data bit can be restored to information that indicates ephemeris, almanac, and transmission time by referring to the pre-defined message format. This information is defined as the navigation message. For example, GPS employs the message structure such as LNAV and CNAV (GPS ICD 2013). The BDS employs D2NAV structure transmitted from geostationary orbit (GEO) satellites, and D1NAV structure transmitted from inclined geosynchronous orbit (IGSO) or medium earth orbit (MEO) satellites (BDS ICD 2013). A navigation message decoder that matches the message structure applied to the signal should be used to decode navigation message from the received satellite signals.

A navigation message decoder can extract parameters required to calculate satellite positions. Each GNSS has a different method to calculate a satellite position using parameters. For GPS and BDS, satellite positions that correspond to the system time can be calculated by users through providing parameters in the orbit equation (GPS ICD 2013, BDS ICD 2013). The GLONASS provides satellite position, velocity, and acceleration in every 15 min (GLONASS ICD, 2008). A GLONASS satellite position at the current time is calculated by an extrapolation method utilizing the above information. Thus, each of the satellite position calculator should be implemented according to the navigation message format.

In the navigation message transmitted from satellites, satellite position information as well as transmission time information are included. Typically, time of week (TOW) in GPS, second of week (SOW) in BDS, and transmission time in GLONASS are found. GPS time is expressed by TOW count, which is the time duration since the last GPS week start/end to the beginning of the next frame in units of 6 sec (GPS ICD 2013). BDS time is expressed by SOW count, in which a time in a week is defined by one count per sec (BDS ICD 2013). A time of GLONASS is synchronized with UTC, and transmission times are provided by hour, min, and sec format (GLONASS ICD 2008).

When calculating a pseudorange between satellite and receiver, the data bit extraction time measured in the receiver channel and the data transmission time included in the navigation message are utilized. The position estimation method using the distance calculated by the above method is only valid based on the assumption that the time between satellites are synchronized. The time synchronization between satellites should be considered when receiver channels are configured to receive signals from different systems. This is because each system employs its own unique system clock.

For GNSS SDR to have a flexible structure that is changed according to the input parameters, it should be able to determine the position regardless of the channel configured by a user considering navigation message structure, satellite position calculation method, and system time difference, which are different from system to system. As shown in Fig. 5, this study designed navigation signal processing utilizing the message type parameter included in each channel. Since the satellite position determination method is different depending on a message format, a user inputs the message format that is matched with the corresponding receive channel. For example, satellite position and pseudorange can be calculated normally when the message format is inputted as LNAV if the channel receives GPS PRN No. 3 signal.

Each channel inputs data bits extracted by referring to the message format using an appropriate decoder. The parameter extracted through the decoder is inputted to the pseudorange and position estimation modules. Here, the pseudorange module compensates the aforementioned system time difference and calculates a pseudorange. The position estimation module calculates a satellite position of each channel through the parameters received from the decoder, and combines calculation result with the pseudorange to calculate receiver position and receiver's clock bias error. The receiver's bias error is delivered to the pseudorange module again to compensate the receiver's clock. The receiver's clock is also synchronized with the navigation system's clock in this process.

The designed SDR solved the clock difference by converting different system clocks into UTC. UTC is the world time standard used since January 1 in 1972. Different system times can be converted to UTC times using Eq. (3), and from which, pseudorange measurement can be done. In Eq. (3), t_{gnss} refers to system reference time that can be obtained in each of the navigation systems. t_{offset} refers to a time difference between UTC and system and information that is not changed, which is defined in the Interface Control Document (ICD) of each satellite system (GLONASS ICD 2008, GPS ICD 2013, BDS ICD 2013). t_{leap} refers to a leap second, which can vary over time. The calculation is done by the module to convert time information provided in the system into second unit of daily basis. Utilizing this equation, since GLONASS manages to maintain a three-hour time difference with UTC, it satisfies t_{offset} = 10800 and the leap second becomes 0. In contrast, t_{offset} is 0 and the leap second occurs in GPS and BDS. The leap second in GPS and BDS can be obtained in the navigation message.

$$t_{utc} = \mathrm{mod}(86400, t_{gnss} + t_{offset} - t_{leap}) \tag{3}$$

The receiver's position can be calculated through the weighted least square technique when pseudorange and satellite position are acquired from four or more satellites (Kaplan & Hegarty 2006). Here, the position is calculated considering output rate, moving average size, channel selection, and elevation masking. The output rate is related to a cycle that measures position velocity and timing (PVT) and pseudorange. The moving average size refers to a window size of the low pass filter to remove high-frequency noise in the produced PVT information. The output rate can be set up to 1 Hz - 100 Hz and the moving average size to 1 - 100 depending on the receiver's use environment. The channel selection parameter is a function that is used by a user to select the measurement that will be utilized in satellite position calculation by channel unit. It enables observations of more various scenarios than the system selection provided by CTTC. It was designed that even the DSSS signals where navigation messages were not implemented in the software can produce data measured at the receive channel. Although the measurement of the signal is not utilized in PVT calculation, it enables performance evaluation of receive according to changes in signal structure, code, and band.

4. IMPLEMENTATION OF GNSS SDR

4.1 Hardware Configuration

The SDR can be utilized in performance evaluation on algorithms as well as observation tools under special circumstances such as the ionosphere, interference, and spoofing if it is run in real time. Currently, most developed SDRs support real-time operation based on a graphics processing unit (GPU), and the proposed SDR also referred to studies on GPU-based real-time SDR implementation (Park et al. 2015). The hardware configuration is shown in Fig. 6. USRP is a well-known universal RF front-end developed by Ettus (Thompson et al. 2012, Park et al. 2017). When used USRP, its the LO frequency of down-converter, sampling frequency of ADC, and amplifier's gain can be changed, therefore, bandwidth and center frequency of the receiver can be easily changed. Since various wireless signals can be received without hardware change, USRP has been widely used in SDR implementation (Goverdovsky et al. 2016). However, USRP has no automatic gain control because it is a general-purpose device. A user should determine the input gain by external amplifier to match the signal power that is received to the resolution of ADC in USRP (Park et al. 2017). In addition, additional power supply is needed to utilize active mode antennas used in GNSS receivers. Considering this, 20 dB amplifiers with DC input function in the antenna direction were used between the antenna and USRP (GPS Networking 2007).

For USRP, X310 product was used, and WBX-40MHz was used in the daughter board. The USRP consists of mother and daughter boards. In the mother board, FPGA and ADC are arranged so that analog signals are converted into digital

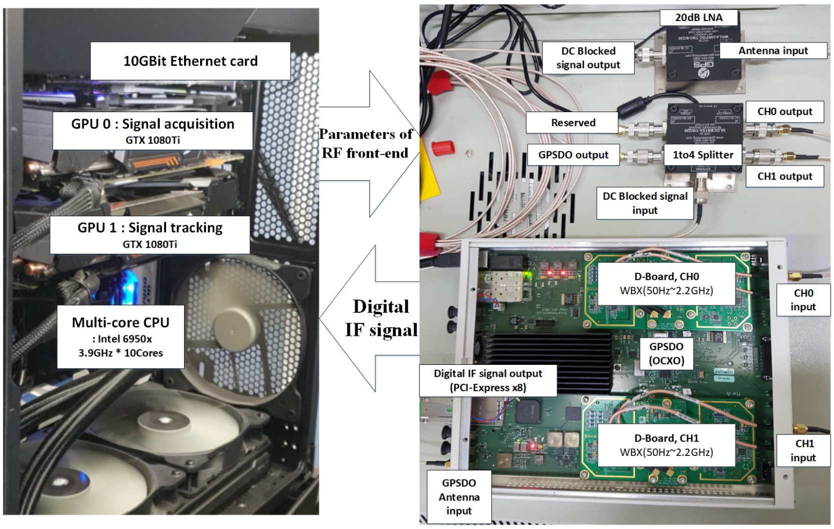

Fig. 6. A hardware configuration of proposed SDR using USRP and GPU.

signals, which are then delivered to PC. In the daughter board, a down-converter and amplifier are arranged. Two daughter boards can be mounted in a single mother board in X310 product (Ettus research 2018). Since a down-converter is presented in each of the daughter boards, different LO frequency can be set up respectively. That is, a single USRP can receive satellite signals at the upper-L and lower-L bands, which bandwidth is about 300 MHz, simultaneously. Since a single mother board is used, the same sampling frequency must be applied to both daughter boards. However, users do not need to consider additional clock synchronization due to the operation by a single clock source (Park et al. 2017).

A signal splitter is added to input the analog GNSS signals obtained through the antenna and amplifier to USRP of two channels (GPS Networking 2008). The splitter distributes a single signal to four signals as shown in Fig. 6. The two outputs are inputted to Channel 1 and 2 of the USRP. One of the remaining two outputs is inputted to a GPS disciplined oscillator of the USRP. The frequency stability of the oscillator can be improved up to 0.02 part per million (ppm) utilizing the in-built GPS receiver (Ettus research 2018). The remaining output was left to receive signals simultaneously with a commercial receiver.

The data size of digital IF signals generated in USRP varies depending on sampling frequency, the number of quantization bits, and complex signal presence. The sampling frequency can be selected limitedly from 1 MHz to 200 MHz in X310. A sample of signal data is represented with 16 bits. If both I-phase and Q-phase signals are received, its size becomes twice. If 50 Msps sampling frequency is applied to receive GNSS signals and complex signals are used, the data amount would be 1.6 Gbit per sec approximately. Thus, Ethernet interface that supports 10 Gbit bandwidth was used to transfer data in real time from USRP to PC (Ettus research 2018).

The receiver channel of SDR performs operations with regard to a large number of digital signals such as correlation computation of signal acquisition and signal tracking, fast Fourier transform, and carrier waves and code generation. PC was configured using two GPUs to process the operations in real time. The software was implemented that each of two GPUs handle signal acquisition and tracking, respectively. For the GPU, GTX 1080Ti product was used.

In USRP, data at a rate of 2.4 Gbit per sec were generated and transferred to PC. The data bus was configured with the Peripheral Component Interconnect Express (PCI-Express) 3.0 specification in the PC to receive data without bottleneck. Moreover, data transfer between CPU and GPU is also done through the bus configured with the PCI-Express 3.0 specification. It is important to select CPUs and main board that support a sufficient amount of PCI-Express bus in order to prevent bottleneck during signal processing operation and signal data acquisition. One GPU occupies 16 PCI-Express buses. Since two GPUs are used, 32 buses are assigned. In addition, one Ethernet card occupies eight buses. Thus, CPU should support at least 40 PCI-Express buses. In this study, Intel 6950x product was used in consideration of the bus and multithread operations. The used CPU supports 40 buses and 10 multicores.

4.2 Implementation of Signal Processing Software

The aforementioned designed receiver channel and navigation modules become a receiver by adding a USRP

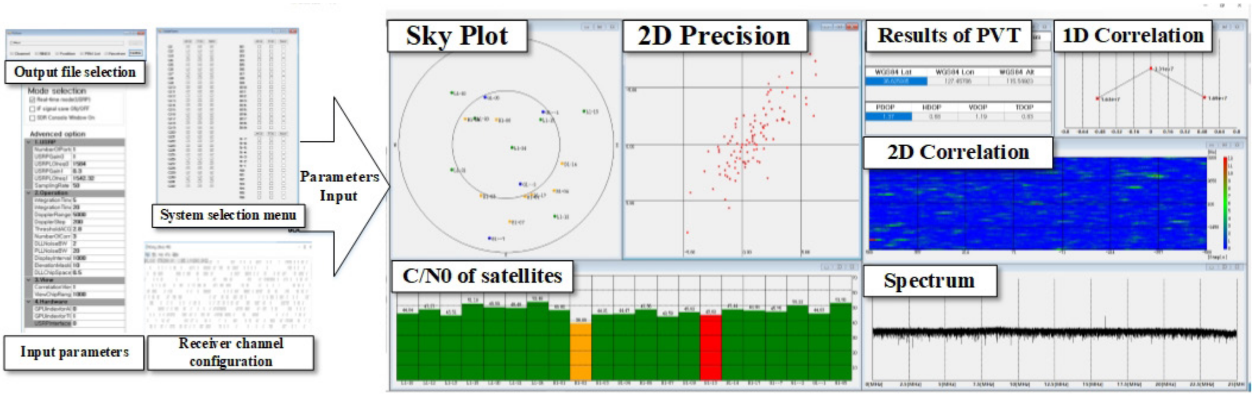

Fig. 7. A graphic user interface of implemented SDR.

control function, receiver channel management, signal buffer management modules. In this study, SDR software was implemented using C++ and compute unified device architecture (CUDA). The function that set the USRP parameters and received data at the PC was implemented utilizing USRP hardware driver. In addition, a buffer for signal management was designed and implemented to transfer signals to the signal acquisition and tracking units reliably. Here, the signal management module consists of arrangement that collects signal at 1 msec unit, queue buffer that manages the signal arrangement by first input first output architecture, and signal tracking and acquisition signal arrangements that are referred to in the signal acquisition and tracking phases, respectively. For detailed information, a study on signal data buffer design for real-time SDR implementation by Park et al. (2016a) was referenced.

The receiver channel was implemented by modularization using C++ class as objects. In addition, objects were created in accordance with the receiver channel configuration and the number of channels entered by a user during SDR operation, and parameters were initialized. The receiver channel can be changed into following states: channel creation signal acquisition (ACQ), code and carrier synchronization (CCLOCK), bit synchronization (BITLOCK), frame synchronization (FRAMELOCK), and navigation availability (NAVIGATION). In addition, a memory was assigned to record the number of channels and channel number for each state thereby implementing the record of current status.

The ACQ state is a channel that acquired prior information during signal acquisition. The ACQ state channel becomes the signal tracking target channel. If the Doppler frequency, carrier phase and code phase are synchronized in signal tracking, the channel state become CCLOCK. Whether the synchronization is complete can be verified by a ratio of correlation value size of I-phase and Q-phase signals (Tsui 2005). The CCLOCK state channel is transitioned into BITLOCK state when bit synchronization is conducted, and more than 10 bits are extracted as the channel is matched with the secondary code sequence. BITLOCK is then changed into FRAMELOCK once the navigation message is decoded and preamble test, parity test, and time information and sub-frame validity test are passed. Once FRAMELOCK state channel collects navigation message sufficiently thereby obtaining satellite position and transmission time, the state is changed to NAVIGATION. If more than four channels at NAVIGATION state are present, the receiver performs receiver position calculation.

Each implemented functions are divided into four threads: signal data management, satellite signal acquisition, signal processing and navigation, and data output to perform operation concurrently. The signal data management performs a function that fetches signal data from USRP and stores the data in the signal buffer. The signal acquisition is a function that searches prior information of satellite signal, in which user-defined channels are searched from Channel No. 1 sequentially. Here, the searched satellite signal is recorded in CCLOCK channel management memory, and the signal processing and navigation thread of the receiver channel object performs signal tracking, bit synchronization and extraction, navigation message decoding, and satellite position determination. The data output thread outputs the calculated PVT, computation results at each receiver channel, and receiver operation status according to user-defined output interval. This was referred to from the study on multithread-based SDR implementation (Park et al. 2016b).

For SDR in consideration of rapid prototyping, the design of user interface is important as well. Since all of code information, center frequency, and message format are inputted for N channels to configure the receive channels, user interface should be easier than the software modification. Fig. 7 shows the user interface implemented in this study. The information required for receiver operation are RF front-end related parameters, receive channel-related parameters, and navigation signal processing-related parameters. The RF front-end and data output, and some receive channel parameters were included in the interface to be changed even while the receiver was operated. The remaining receive channel parameters that were related to signal structure and difficult to be changed were implemented to be entered through a separated setup file.

The user interface was designed to control USRP's LO frequency, sampling frequency, and amplifier gain. Among the receive channel-related parameters, integration time, Doppler search range and step, signal acquisition threshold, number of correlators, and noise bandwidth were also included in the user interface. Since the parameters can be modified during receiver operations, changes in receiver states according to parameter change can be observed in real time. The output rate, elevation angle, and channel selection functions are also included in the user interface.

The information that is put through a file includes a name of each channel, center frequency, code, code frequency, code length, message format, and modulation mode. Each receive channel was separated by a line feed character, and parameters were separated by a comma.

A receiver generates and initializes a receive channel according to user input parameters. The information displayed in the user interface includes sky plot, position precision, PVT results, C/N0 of each channel, signal's spectrum, and correlation results of signal tracking and acquisition in real time. Moreover, PVT measurement results, channel number list used in PVT measurement, USRP's operation status, receiver operation status, and computation results at each channel are recorded in the file along with measurement time. Among them, the computation results at each channel are produced at a rate of 1 to 1,000 Hz, and include receive status, channel status, correlator status, extracted bits, and transmission and receive clock information. The output file type can be selected in the user interface.

5. EXPERIMENT AND ANALYSIS

5.1 Experiment Method and Setup of SDR Input Parameters

The position measurement experiments were conducted using live GNSS signals to verify PVT performance according to the configuration of various receive channels set by a user. The SDR was initialized with the five channel configurations, and each configuration is used to receive GNSS signal and measure receiver's positions. Table 2 presents each channel configuration and parameter setup. The receiver's bandwidth and center frequency were set up by changing LO frequency and sampling frequency, and to receive GNSS signal that were located inside receiver's bandwidth, receiver channels were configured.

Table 2. Input parameters of SDR with each scenario.

Input parameter	Scenario No.				
	1	2	3	4	5
Sampling frequency (Msps)	10	20	25	40	50
LO frequency (MHz)	1572	1557	1567	1595	1584
Number of channels	32	19	51	14	65
Gain of amplifier (dB)	0				
Real/Complex	Real				
Number of correlators	3				
Chip spacing	0.5				
Doppler range/step (Hz)	5000/100				
Integration time (msec)	5				
Output rate/Display rate (Hz)	1/1				
Elev. angle masking (degree)	5				
Average size	10				
Noise bandwidth of PLL (Hz)	20				
Noise bandwidth of DLL (Hz)	2				

Scenario Nos. 1, 2, and 4 were setups to receive GPS L1 C/A, BDS B1I, and GLONASS G1 OS, respectively. The LO frequency in GPS L1 C/A signal that approximately occupied 2 MHz bandwidth around 1,575.42 MHz was set to 1,572 MHz so that the center frequency of GPS L1 C/A be become 3.42 MHz. Then, GPS signals can be received by sampling at 10 Msps. Here, the number of receiver channels was set to 32, in which information such as code from GPS PRN No. 1 to 32 and center frequency is inputted.

BDS B1I signals are transmitted at 1561.098 MHz. It occupies about 4 MHz bandwidth. The BDS B1I signals that are located at about 4.098 MHz can be received by LO frequency at 1,557 MHz and sampling at 20 Msps. The number of the receive channels was 19, in which information from BDS PRN No. 1 to 19 was inputted.

GLONASS G1 OS occupies about 8 MHz bandwidth of 14 signals at 562.5 kHz interval from Channel No. -7 to 6. No. -7 satellite employs 1598.0625 MHz and No. 6 satellite uses 1605.375 MHz. The LO frequency was set to 1,595 MHz for the frequency crossed over from the negative region not to be overlapped and the sampling frequency was set to 40 Msps. When the receiver operates to receive GLONASS signal, 14 receiver channels are assigned based on the channel number from -7 to 6 rather than the number of satellites.

Scenario Nos. 3 and 5 are experiments that combine two or more systems. In Scenario No. 3, LO frequency was set to 1,567 MHz to receive GPS and BDS signals simultaneously. In GPS and BDS, signals will be located at 8.42 MHz and 5.902 MHz, respectively. By sampling at

Table 3. Configuration parameters of channel No. 1, 33, 39, 52 in scenario No. 5.

Input parameter	Channel No.			
	1	33	39	52
Channel name	GPSL1CA01	BDSB1IG01	BDSB1IM07	GLOG1-7
Center frequency (Hz)	1575420000	1569098000	1569098000	1598062500
PRN No.	001	001	001	-07
Modulation type	001 = BPSK	001	001	001
Message type	000 = LNAV	002 = D2NAV	003 = D1NAV	001 = GNAV
Data rate (bps)	0050	0500	0050	0100
Code frequency	01023000	02046000	02046000	00511000
Code length	001023	002046	002046	000511
Primary code sequence	11-1-11-1-1-1...	1-1-11-1-11-1...	1-1-11-11-1-1...	1111111-1...
Secondary code sequence	1...	1...	-1-1-1-1-11...	1...

25 Msps considering the bandwidth, GPS and BDS signals were included in a single channel of USRP. Here, the receive channels were configured that front numbers were assigned to 32 GPS satellite signals and following numbers were assigned to 19 BDS satellite signals. Thus, a total of 51 receiver channels were operated.

Scenario No. 5 was a setup that can receive three GPS, BDS, and GLONASS signals simultaneously. By setting up the sampling frequency to 50 Msps, GPS at 8.58 MHz, GLONASS at 18 MHz, and BDS at 22.902 MHz can be received simultaneously. The cut-off frequency in the low pass filter in the USRP used in this study was 20 MHz. Some BDS and GLONASS signals may be lost. The satellite signals of GPS, BDS, and GLONASS were configured sequentially in the receive channels, and 65 receive channels were run in total.

The same values were applied to the rest of the parameters in the scenarios except for sampling frequency, LO frequency, and receiver channel configuration. The amplifier of RF front-end was 0 dB. The simplest type of receiver was set up by using only real signals, three correlators, and 0.5 of chip spacing. Since coherent integration time of signal acquisition is 5 msec, 5 kHz Doppler was searched at 100 Hz interval. Both of the position calculation and output cycles were 1 Hz, in which a moving average filter was applied to 10 measurements. In the setup, satellite signals whose elevation angle was less than 5° was removed from the positioning. The PLL noise bandwidth of the carrier tracking loop was set to 20 Hz, and DLL noise bandwidth of the code tracking loop was set to 2 Hz.

Table 3 presents some of the parameters in the receiver channel configuration set in Scenario No. 5. In receive channel No. 1, GPS L1 C/A PRN No. 1 signal was defined. In Receive Channel No. 33, BDS B1I PRN No. 1 was assigned, which was GEO satellite signal. In receive channel No. 39, BDS B1I PRN No. 7 was assigned, which was MEO satellite signal. Receive channel No. 52 was set to receive GLONASS G1 OS No. -7 satellite signals. Each of the receiver channel names can be defined by users arbitrarily. In the center frequency, the center frequency of each system was inputted. In the center frequency of No. 52 channel that received FDMA signals, the center frequency of the No. -7 GLONASS signal was inputted. The PRN number is used to distinguish the receive channel in the receiver. The definition of the modulation mode was enumerated type. 001 and 002 refer to binary phase shift keying (BPSK) and binary offset carrier (BOC), respectively. The current implemented SDR can receive BPSK signals only. However, BOC signals can be received if the correlator of BOC signals is added later since the chip spacing and the number of correlators can be changed.

The message format and data rate were set according to the satellite system received by each receiver channel. The data rate of GLONASS was 50 bps but it was set to 100 bps considering the meander (GLONASS ICD 2008). The code rate, code length, and code sequence of each system are inputted as the code-related information. The code was represented with 8 bit per chip in the implementation so that codes other than binary code can be supported in future. The secondary code was all set to “1” except for signals that used a tiered mode. In current scenarios, only BDS signals that used D1NAV message structure employed the secondary code. The MEO and IGSO satellites belonged to this among BDS satellites.

5.2 Experimental Results and Analysis

Signals were received by GPS-702-GGL antenna installed at the rooftop in Chungbuk National University in September 7, 2018 using implemented SDR initialized with the input parameters of five scenarios (NovAtel 2018). The antenna was fixed. Signals were received for about 9 min for each scenario to compare the scenarios even after the experiment at the similar satellite arrangement environment. Since it was run without prior information, position estimation could take up to 40 sec. The performance was verified through 500 navigation solutions

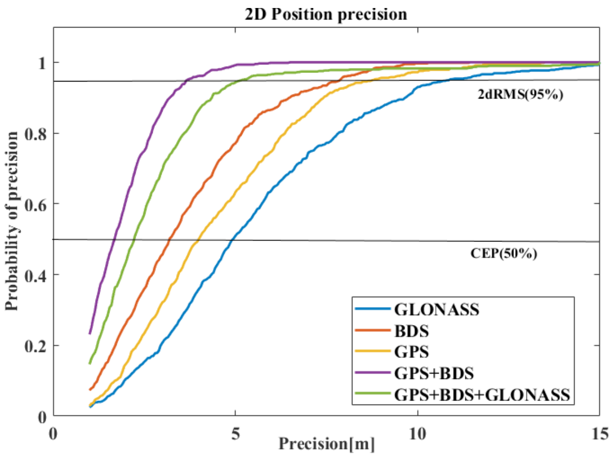

Fig. 8. Precision of position with various combination of GNSS that is computed by implemented SDR.

measured for 9 min in each scenario, and the stability was verified based on computation time per second, buffer management status, and memory leakage status.

Scenario No. 5, which had the highest computation load among the five scenarios, received up to 25 satellite signals, and it was utilized in navigation solution computation. Here, the computation time per signal data for one sec was up to 0.88 sec. The results verified that the implemented SDR could receive up to three systems simultaneously in real time. During the execution of No. 5 scenario, signal data buffer size for about 9 min was maintained uniformly below about 1.2 Mbyte (approximately 6 msec signal length), and assigned memory was maintained below 288 Mbyte. The system was verified that there was no problem in more than 9 min of long-time operation as the buffer size was maintained and memory leakage did not occur.

Approximately 500 position results acquired per each scenario were expressed via longitude, latitude, and altitude values. These values were converted to the North East Down coordinate and then two-dimensional (2D) position was calculated with values in the N and E axes. In addition, means and errors of the calculated 2D positions were calculated. The calculated errors were summarized with the distribution according to the error size as shown in Fig. 8, and the size was normalized. In these results, the precision was summarized based on the Circular Error Probability (CEP). Table 4 presents the summary.

For single GPS, it had an error of 4.0 m. For BDS and GLONASS, they had errors of 3.2 m and 4.9 m, respectively. When GPS and BOS were combined, the best performance was exhibited as an error of 1.7 m, and when combining GPS, BDS, and GLONASS, it had an error of 2.25 m. When using the multi-systems, all of them showed at least 1.4 times better performance than using a single system. In the case of GLONASS, only six satellites were received on average, which was the smallest number, and the system was known as the degradation of performance due to the difference in satellite position calculation method compared to other systems. Thus, when combining GLONASS, degradation in precision occurred so that results in Scenario Nos. 3 and 5 could occur.

Table 4. Comparison of positioning precision with results of other SDR

System	Precision of 2D position (m)	
	Söderholm et al. (2016)	Implemented SDR
GPS	2.98	4.0
BDS	3.41	3.2
GLONASS	-	4.9
GPS+BDS	2.28	1.7
GPS+BDS+GLONASS	-	2.25

Compared to other SDR results in other study by Söderholm et al (2016), there are differences of 1.02 m for GPS, 0.21 m for BDS, and 0.58 m for GPS+BDS. GPS is more precise when measured in other study, but the proposed SDR is more accurate when using BDS or BDS+GPS. Considering the different reception environments, the performance of the implemented SDR is similar to that of other SDR.

The live signal reception experiment results verified that SDR configured with only changes in input parameters received GPS, BDS, and GLONASS signals, and correct navigation solution were measured. The present receiver implemented LNAV, GNAV, D1NAV, and D2NAV so that the receiver channel configuration of GPS, BDS, and GLONASS can be changed easily. Since software modification is not required additionally when configuring the receiver channel, more rapid response can be achieved than existing SDR does. Also DSSS signals that are transmitted to different code or different center frequency can be received by change of the receiver channel file to set up. Thus, rapid prototyping with regard to various signals can be achieved with more flexible structure than existing SDR does.

6. CONCLUSIONS

This paper proposed and implemented a SDR architecture for rapid prototyping of GNSS receivers. The proposed SDR supports the input of signal structure information of desired signal such as code sequence, code frequency, code length, center frequency, message format, and data rate through user interface. The correlator and bit extractor considering the defined input parameters were designed, and utilizing them, a receiver channel that can demodulate various DSSS signals was designed. In addition, navigation signal processing that can calculate a receiver position was designed and implemented even if the receiver channel was configured with different GNSSs. The pseudorange can be measured without system distinction by converting different system clocks into UTC-based clock. Appropriate navigation message decoders can be applied by inputting the message format of the receiver channel. Even if navigation message decoders are not implemented, receiving performance can be evaluated by providing data bit, C/N0, correlation value of each correlator, and Doppler and code phase tracking information.

To verify the validity of the proposed SDR architecture, the hardware was configured with USRP and GPU-based PC. The configured hardware is for real-time operation SDR. With the real-time operation, the SDR can be utilized as evaluation tool and as real-time observation tool. The designed receiver channel and navigation signal processing module were implemented by C++ and CUDA C language.

To verify the performance of the implemented SDR, experiments were conducted with five scenarios: GPS, BDS, GLONASS, GPS+BDS, and GPS+BDS+GLONASS. The experimental results verified that the position precisions of all combination scenarios were within 5 m of the CEP. Among them, GPS result was similar to that of other SDR result, and when combining GPS+BDS, the precision was 1.7 m, which exhibited the best precision. All the satellite system combinations in the experiment can be configured simply with changes in input parameters. The implemented SDR is expected to be utilized in various fields due to the advantage of rapid prototyping with regard to various GNSS and DSSS signals.

ACKNOWLEDGMENTS

This work has been supported by the National GNSS Research Center program of Defense Acquisition Program Administration and Agency for Defense Development.

REFERENCES

BDS ICD 2013, Beidou Navigation Satellite System in Space Interface Control Document

Bhuiyan, M. Z. H., Söderholm, S., Thombre, S., Ruotsalainen, L., & Kuusniemi, H. 2014, Overcoming the Challenges of BeiDou Receiver Implementation, Sensors, 14, 22082-22098. https://doi.org/10.3390/s141122082

Cristodaro, C., Dovis, F., Linty, N., & Romero, R. 2018, Design of a Configurable Monitoring Station for Scintillations by Means of a GNSS Software Radio Receiver, IEEE Geoscience and Remote Sensing Letters, 15, 325-329. https://doi.org/10.1109/LGRS.2017.2778938

Ettus research, USRP Hardware Driver and USRP Manual [Internet], cited 2018 Sep 12, available from: http://files.ettus.com/manual/index.html

GitHub-taroz, GNSS-SDRLIB [Internet], cited 2018 Sep 12, available from: https://github.com/taroz/GNSS-SDRLIB

GLONASS ICD 2008, GLONASS Interface Control Document, Version 5.1

GNSS-SDR [Internet], cited 2018 Sep 12, available from: https://gnss-sdr.org/

Goverdovsky, V., Yates, D., Willerton, M., Papavassiliou, C., & Yeatman, E. 2016, Modular software-defined radio testbed for rapid prototyping of localization algorithms, IEEE Transactions on Instrumentation and Measurement, 65, 1577-1584. https://doi.org/10.1109/TIM.2016.2540998

GPS ICD 2013, Global Positioning Systems Directorate System Engineering & Integration Interface Specification IS-GPS-200H

GPS Networking 2007, LA20RPDC Line Amplifier 20dB Gain Technical Product Data

GPS Networking 2008, LDCBS1X4 Passive 1X4 GPS Splitter Technical Product Data

GPSworld, India successfully launches IRNSS-1I navigation satellite [Internet], cited 2018 Sep 12, available from: http://gpsworld.com/india-successfully-launches-irnss-1i-navigation-satellite/

Kaplan, E. D. & Hegarty, C. J. 2006, Understanding GPS: Principles and Applications. 2nd ed. (Boston: Artech House Inc.)

Kordon, F. & Luqi, 2002, An introduction to rapid system prototyping, IEEE Transactions on Software Engineering, 28, 817-821. https://doi.org/10.1109/TSE.2002.1033222

NovAtel, GPS-701-GGL & GPS-702-GGL [Internet], cited 2018 Sep 12, available from: https://www.novatel.com/products/gnss-antennas/high-performance-gnss-antennas/gps-702-ggl/

Park, K. W., Chae, J. G., Song, S. P., Son, S. B., Choi, S., et al. 2017a, A Performance Analysis of Multi-GNSS Receiver with Various Intermediate Frequency Plans Using Single RF Front-end, Journal of Position, Navigation, and Timing, 6, 1-8. https://doi.org/10.11003/JPNT.2017.6.1.1

Park, K. W., Lee, M. J., & Park, C. 2017b, Implementation of a Multi-frequency, Multi- constellation and Real-time GNSS Software Receiver Using Dual Channel

USRP, Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2017), Portland, OR, 25-29 Sep 2017, pp.3710-3717

Park, K. W., Lee, S., Lee, M. J., Kim, S. & Park, C. 2015, An accelerated signal tracking module using a heterogeneous multi-GPU platform for real-time GNSS software receiver, 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP 2015), Orlando, FL, 14-16 Dec 2015. https://doi.org/10.1109/GlobalSIP.2015.7418431

Park, K. W., Lee, S., Lee, M. J., & Park, C. 2016a, Design of a buffer management algorithm for USRP based real-time GNSS SDR, in Proc., Korea GNSS Society conference 2016, Jeju, 02-04 Nov 2016

Park, K. W., Suh, J.-W., Seo, B.-S., Lee, M. J., & Park, C. 2016b, Design of signal acquisition and tracking process based on multi-thread for real-time GNSS software receiver, in Proc. 2016 International Conference on Localization and GNSS (ICL-GNSS), 28-30, Jun 2016, Barcelona, Spain. https://doi.org/10.1109/ICL-GNSS.2016.7533688

Söderholm, S., Bhuiyan, M. Z. H., Thombre, S., Ruotsalainen, L., & Kuusniemi, H. 2016, A multi-GNSS software-defined receiver: design, implementation, and performance benefits, Annals of Telecommunications, 71, 399-410. https://doi.org/10.1007/s12243-016-0518-7

Suzuki, T. & Kubo, N. 2014, GNSS-SDRLIB: An Open-Source and Real-Time GNSS Software Defined Radio Library, Proceedings of The 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2014), Tempa, FL, 8-12 Sep 2014, pp.1364-1375

Thompson, E. A., Clem, N., Renninger, I., & Loos, T. 2012, Software-defined GPS receiver on USRP-platform, Journal of Network and Computer Applications, 35, 1352-1360. https://doi.org/10.1016/j.jnca.2012.01.020

Tsui, J. B. 2005, Fundamentals of global positioning system receivers: a software approach (New York: John Wiley & Sons)

Xu, H., Cui, X., & Lu, M. 2016, An SDR-based real-time testbed for GNSS adaptive array anti-jamming algorithms accelerated by GPU, Sensors, 16. https://doi.org/10.3390/s16030356

Xu, L., Ziedan, N. I., Niu, X., & Guo, W. 2017, Correlation acceleration in GNSS software receivers using a CUDA-enabled GPU, GPS Solutions, 21, 225-236. https://doi.org/10.1007/s10291-016-0516-2

Yang, X., Huang, Z., Han, B., Zhang, S., Wen, C.-K., et al. 2017, RaPro: A Novel 5G Rapid Prototyping System Architecture, IEEE Wireless Communications Letters, 6, 362-365. https://doi.org/10.1109/LWC.2017.2692780

Kwi Woo Park received his B.S. and M.S. degrees from Chungbuk National University, Korea in 2013 and 2015, respectively. He is currently a Ph.D. candidate in Department of Control and Robotics Engineering at the same university. His research interests include GNSS signal processing, SDR and navigation.

Jin-Mo Yang received his B.S. and M.S degrees from Ajou University, Korea in 1988 and 1991, respectively. And He received his Ph.D degrees in Electrical Engineering from Chungnam National University in 2006. He is currently a senior researcher in Agency for Defense Development, Korea, since 1991. His research interests include super high frequency system and phased array radar system

Chansik Park received his B.S., M.S., and Ph.D. degrees in Electrical Engineering from Seoul National University in 1984, 1986 and 1997, respectively. He was a Professor with the School of Electronics Engineering, Chungbuk National University, Cheongju, Korea, since 1997. His research interests include GNSS, PNS, SDR, integer ambiguity resolution algorithm and Error Analysis.