THIS ARTICLE PRESENTS A DETAILED description of the various modules needed for the implementation of a global positioning system (GPS) software receiver. The concept of software-defined radios is gaining a lot of momentum in the telecommunications industry due to the rapid growth in the capabilities of programmable logic devices and the need for multistandard, multiband receiver terminals.

GPS is a satellite-based navigation system that is widely used in civil and military location-based services. The location estimates of users on earth are determined by analyzing the incoming satellite signals in the user receiver. Current GPS receivers capture the radio frequency modulated signals at the Link 1 (L1) frequency of 1.57542 GHz, down convert them to an intermediate frequency (IF), digitize them, and perform signal processing to extract the location information. The driver behind this architecture is the high frequency of the incoming signal and the limited sampling rate of the available analog-to-digital converters (ADCs). This architecture is demonstrated in Fig. 1.

The design of a heterodyne receiver has limited flexibility. Once the receiver is designed, the user has limited options in the radio frequency (RF) tuning and digital signal processing (DSP) portions. This limitation can be overcome by the utilization of software-based receivers that rely on the theory inside band-pass sampling as discussed by Vaughan. In this architecture, the RF signal is digitized without the need for down conversion, and the bandwidth of interest is captured. Then, the DSP part can be reconfigured to perform different signal processing tasks. The DSP can be reprogrammed to give different results for different standards. Field programmable gate array based software receivers and interfaces for GPS signal processing have been addressed in some literature, such as in Cummings and Hein. Figure 2 shows the architecture of an ideal software receiver. The main limitation in this design is the constraint of the incoming signal bandwidths and the high dynamic range needed. For the ADC to cope with these two varying parameters (since a software radio is supposed to capture signals from multiple standards with different power levels and frequency ranges), it has to have a larger number of bits and over-sampling ratios. A more practical design architecture that can reduce the high requirements on the ADC would be to have a programmable IF stage and perform partial band digitization.

The application of the software receiver approach or the so-called software-defined radio approach on GPS receivers gives the algorithm designer, as well as the receiver architect, some definite advantages while testing the performance of various implementations. The advantage of reconfiguring the functionality via programming yields a better time to market and lower debugging and hardware costs. The software radio approach applied to GPS receivers was first presented in 1996, according to Thor and Akos.

GPS signals

The GPS signal consists of the pseudo random noise (PRN) code with a frequency of 1.023 MHz and a period of 1023 b (i.e., 1 µs per code chip). The PRN code spreads the navigation message, which has a frequency of 50 Hz. The resultant signal denoted as the coarse acquisition (C/A) signal is then modulated with a carrier frequency of 1575.42 MHz (L1—link 1) and is then sent through the communication chan-
nel. The PRN code is generated using linear feedback shift registers and belongs to the gold code class of maximum length sequences.

The multiplication of the 50-Hz navigation signal with the PRN will spread its power spectral density over a wider range of frequencies, a type of spread spectrum communications. Because of the long traveling path from the satellite to the receiver on earth, the power levels of the signal will be lower than the noise floor, making the signal invisible. Several amplification stages in the receiver as well as the utilization of the correlation property of the PRN code will bring back the signal and make it detectable by the receiver, and the navigation data is processed.

GPS receiver modules

Radio frequency front end

After the RF signal has been collected by the antenna, it gets amplified using a low-noise amplifier (LNA) (pre-amplifier) that has a low noise figure. The signal is then filtered with a bandpass filter (BPF) to reduce band noise. Such filters should possess sharp transitions. Surface acoustic wave (SAW) filters are widely used in current receivers. The signal is then down converted to an IF via an RF-IF mixer. This process is needed to lower the frequency of the signal for the ADC process. The RF L1 signal needs to be lowered to some 100s of MHz to be able to use commercial ADC converter chips. This is the task of the down conversion section. Some receivers utilize single or multiple IF stages based on the frequency plan of the receiver architecture.

ADC

The ADC will transform the analog IF signal into a digital IF signal for baseband processing. In low-end receivers, single bit (hard limiters) ADCs are used in narrow bandwidths (2 MHz). High-end receivers use up to 3 b (eight levels) of sampling in bandwidths from 2–20 MHz. If a multibit ADC is used, automatic gain control (AGC) is utilized on the incoming signal before entering the ADC to adjust the signal amplitudes based on the ADC dynamic range.

DSP

Signal acquisition

The purpose of acquisition is to find the visible satellites and coarse values of carrier frequency and code phase of their signals. The satellites are differentiated by 32 PRN sequences. The code phase is the time alignment of the incoming PRN code in the current block of data with that of the locally generated one. The code phase is necessary to generate a local PRN code that is perfectly aligned with the incoming code. Only when this is the case can the incoming code be removed from the signal. Another parameter that is important to estimate is the Doppler shift. The line-of-sight velocity of the satellite causes a Doppler effect resulting in a higher or lower frequency value for the incoming signal. In the worst case, the frequency can deviate up to ±10 kHz. It is important to know the frequency of the signal to be able to generate a local carrier signal. This signal is used to remove the incoming carrier from the signal. In most cases, it is sufficient to search the frequencies in steps of 500 Hz.

Two types of signal acquisition are usually adopted in GPS receiver implementations:

1) Serial search acquisition is performed by stepping the code phase of the locally generated PRN to cover the length of the sequence (i.e., 1023) to find the correlation peak and find the code phase offset between the locally generated and incoming GPS signal.

2) Parallel search acquisition is performed by eliminating the full stepping of a whole PRN sequence by utilizing circular correlation between the incoming signal and the locally generated one. This will require less steps (i.e., correlation time) since circular correlation can be performed in the frequency domain utilizing the fast Fourier transform (FFT). Then an inverse FFT is performed to get the signal back into the time domain, but now the correlation peak has been identified. This method is much faster than the serial search method, but the latter can operate on weaker signals.

Tracking

After the acquisition, the frequency and code offset parameters of a satellite signal are roughly known. The main purpose of tracking is to refine these
values and keep track and demodulate the navigation data for the specific satellite. First, the input signal is multiplied with a carrier replica. This is done to wipe off the carrier wave from the signal. Next, the signal is multiplied with a code replica and the output of this multiplication gives the navigation message. So the tracking module has to generate two replicas, one for the carrier and one for the code, to perfectly track and demodulate the signal of one satellite.

The goal for a code-tracking loop is to keep track of the code phase of a specific code in the signal. The output of such a code tracking loop is a perfectly aligned replica of the code. The code tracking loop in the GPS receiver is a delay lock loop (DLL) called an early-late tracking loop. The idea behind the DLL is to correlate the input signal with three replicas of the code, the early, prompt, and late. The three replicas are often generated with a chip spacing of ±0.5. The three outputs are then integrated over a whole C/A code period and stored. The output of these integrations is a term indicating how much the specific code replica correlates with the code in the incoming signal.

To demodulate the navigation data successfully an exact carrier wave replica has to be generated. To track a carrier wave signal, phase lock loops (PLL) or frequency lock loops (FLL) are often used. A loop discriminator block is used to find the phase error on the local carrier wave replica. The output of the discriminator, which is the phase error, is then filtered and used as a feedback to the numerically controlled oscillator (NCO), which adjusts the frequency of the local carrier wave. In this way, the local carrier wave could be an almost precise replica of the input signal carrier wave.

**GPS software receiver implementation**

As shown in Fig. 2, the software receiver starts after the ADC. The raw GPS data tested was collected using a data collection setup that consisted of a rooftop right-hand circularly polarized (RHCP) GPS antenna, an RF front-end with a total gain of about 90 dB, and a data acquisition card that sampled the data and stored it on a hard drive.

The stored signal was then passed to the software receiver implemented using MATLAB. The signal analyzed was at digital IF. The receiver has to generate a replica of the PRN code and then using the code tracking loop, align the locally generated sequence with the received one to be able to despread the signal to yield the navigation data. Figure 3 shows a block diagram of the GPS software receiver implemented with the carrier and code tracking loops interconnected.

In the GPS software receiver implemented in MATLAB, the gold code generator can generate any of the 32 GPS satellite PRN codes. It can be called as a function, and it will generate the 1,023 bits of the desired PRN sequence. These bits have to be up-sampled to be used with the sampled raw data. The up-
sampling factor is given by \( \frac{T_{C/A}}{T_s} \), where \( T_{C/A} \) is the C/A code bit period, and \( T_s \) is the sampling time. The first stage of the software receiver is to perform acquisition on the raw data. This will identify the visible satellites and will provide the code and Doppler shifts. The two-dimensional space that finds the code and carrier phases is shown in Fig. 4. The peak will give the current code and carrier (Doppler) phases for the acquired signal.

Once the visible satellites are identified and the code phase and Doppler shift are stored, the tracking loop (for both code and carrier) tracks the changes in the C/A code and carrier shifts. As can be seen in Fig. 3, the prompt outputs of the code tracking loop feeds the carrier tracking loop, while the carrier tracking loop feeds the NCO used to create the \( I \) and \( Q \) components used by the code-tracking loop. The analysis is conducted on a block by block basis, where each block is worth 20 ms of data corresponding to a single navigation data bit. The data within the block is processed as shown in Fig. 3, where the digital IF data is multiplied by the sine and cosine functions to form the \( I \) and \( Q \) streams, then the data is correlated within the code tracking loop to form the early, prompt, and late components. These three are used within the DLL to speed up or slow down the internally generated C/A code to despread the signal. The integration time used in this work was over 1 ms, which is suitable for tracking good signals. If weak signals are to be detected, the integration time should be increased accordingly. Within the same tracking loop, the PLL will track the carrier phase shift depending on the prompt value of the internally generated C/A code. This carrier phase adjustment is supplied to the sine and cosine functions (NCOs) that are used to wipe off the carrier component of the incoming signal.

The \( I \) and \( Q \) values coming out of the tracking loop are presented in Fig. 5. The navigation data bits are primarily being detected on the \( I \) component of the output. This is due to the fact that the local carrier wave is in phase with the input signal, thus all the energy will be in the \( I \) component.

Figure 6 shows the error of the locally generated code frequency and the error in the locally generated sine and cosine function frequencies. The plots are generated by subtracting the known frequencies of the generated C/A code and IF carrier from the original values of 1.023 MHz and 4.333211 MHz for the code and IF, respectively. The code error was within \( \pm 3 \) Hz while the IF frequency error was within \( \pm 4 \) Hz. These numbers show that the implemented code and carrier loops are tracking the incoming signal with pretty good accuracy.
Conclusions
Software-based GPS receivers are the focus of current research because of their reprogramming ease and flexibility. Sampling an incoming wideband signal will allow the receiver to capture more navigation signals from different standards. This task is very important for more accurate position determination. This article explains the various components needed to implement a GPS-based software receiver and shows the results of one implemented in MATLAB.

Read more about it

About the authors
Oleksiy V. Korniyenko is currently pursuing his Ph.D. in systems engineering at Oakland University, Rochester, Michigan. His areas of interest include, software defined radios, navigation systems and filter design for position estimation.

Mohammad S. Sharawi obtained his B.S. degree in electronics engineering with highest honors from Princess Sumaya University for Technology, Amman, Jordan in 2000. He was an exchange student during the academic year of 1997/1998 at the University of Illinois at Urbana-Champaign. He received his M.S. degree in electrical and computer engineering and Ph.D. degree in Systems Engineering from Oakland University, Rochester, Michigan, in 2002 and 2006, respectively. He is with the Computer Engineering Department at Philadelphia University, Amman, Jordan.