A Graphical Approach to GPS Software-Defined Receiver Implementation

Zaher M. Kassas, Jahshan Bhatti, and Todd E. Humphreys

Radionavigation Laboratory, Wireless Networking and Communications Group

The University of Texas at Austin

Austin, Texas, USA

zkassas@ieee.org, jahshan@utexas.edu, todd.humphreys@mail.utexas.edu

Abstract—Global positioning system (GPS) software-defined receivers (SDRs) offer many advantages over their hardware-based counterparts, such as flexibility, modularity, and upgradability. A typical GPS receiver is readily expressible as a block diagram, making a graphical approach a natural choice for implementing GPS SDRs. This paper presents a real-time, graphical implementation of a GPS SDR, consisting of two modes: acquisition and tracking. The acquisition mode performs a two-dimensional fast Fourier transform (FFT)-based search over code offsets and Doppler frequencies. The carrier-aided code tracking mode consists of the following main building blocks: correlators, code and carrier phase detectors, code and carrier phase filters, a code generator, and a numerically-controlled oscillator. The presented GPS SDR provides an abstraction level that enables future research endeavors.

Index Terms—Global Positioning System (GPS), software-defined receiver (SDR), acquisition, tracking.

I. INTRODUCTION

Signal processing algorithms in software-defined receivers (SDRs) are typically implemented on general purpose digital signal processors (DSPs), with only minimal dedicated hardware components to the radio frequency (RF) front-end. Traditionally, baseband operations in Global Positioning System (GPS) receivers have been implemented using dedicated hardware due to cost, power, and speed [1]. Until recently, GPS SDRs were limited to post-processing applications operating on raw samples recorded from an RF front-end [2]. However, with modern DSPs, real-time (RT) GPS SDRs are becoming more prevalent. Such SDRs are typically implemented in high-level textual-based languages, such as C/C++. Processor-specific optimization techniques are often utilized for computationally-expensive baseband operations [3].

Graphical programming languages, such as LabVIEW® and Simulink®, are attractive choices for implementing GPS SDRs for a number of reasons. First, while the optimized C/C++ GPS SDR implementations are often portable and reusable on multi-core DSPs [4], the optimizations required for each processor in RT applications could slow development and introduce platform-specific errors. Graphical programming languages offer tools which often generate optimized implementations for multiple platforms – desktop, DSP, and field-programmable gate arrays (FPGAs) – without code modifications. Second, GPS SDRs are conceptualized as block diagram, enabling a one-to-one correspondence between the architectural conceptualization and software implementation. Third, graphical optimized routines are abundant, which could be readily exploitable by GPS SDR designers. Fourth, data-flow-based graphical implementations are easier to understand and debug, and they offer rapid access to all internal signals. Fifth, graphical tools provide attractive graphical user interfaces (GUIs), allowing designers to develop interactive panels that have the look and feel of hardware-based GPS receivers.

Graphical implementations of GPS SDRs have been the subject of a number of recent publications [5], [6]. This paper extends such previous work by outlining the GPS SDR block diagram and clearly presenting its basic building blocks and associated design alternatives. Furthermore, an implementation of a LabVIEW-based GPS SDR is discussed, serving as a guide for future implementations.

This paper is organized as follows: Section II presents the received GPS signal model. Section III gives an overview of the three main receiver stages. Section IV presents the building blocks involved in a single-channel GPS SDR. Section V illustrates the salient features of the LabVIEW-based GPS SDR. Concluding remarks are given in Section VI.

II. GPS RECEIVED SIGNAL MODEL

A model of the received GPS signal exiting the RF front-end after digitization and down-mixing to baseband is

$$x(j) = A(\tau_j)D\left[\tau_j - t_d(\tau_j)\right]C\left[\tau_j - t_s(\tau_j)\right]e^{i\,\theta(\tau_j)} + n(j) \quad (1)$$

$$D(\tau) = \sum_{k=-\infty}^{\infty} d_k\,\Pi_{T_d}(\tau - kT_d), \;\; C(\tau) = \sum_{k=0}^{N_c-1} c_k\,\Pi_{T_c}(\tau - kT_c)$$

where Π_{T_a} is the standard pulse function with support T_a, $\tau_j \triangleq jT$ is the sampling time expressed in receiver time, T the uniform sampling interval, A is the signal amplitude, D is the navigation data, d_k is a ± 1-valued navigation symbol, T_d is the navigation symbol interval, t_d is the start time of some symbol reference epoch, C is the periodic spreading (ranging) code sequence, N_c is the number of chips in one spreading code period, t_s is the code phase expressed as the maximum spreading code start time that respects the bound $t_s(\tau_j) \leq \tau_j$, c_k is a ± 1-valued spreading code chip, T_c is the chipping interval, θ is the beat carrier phase in radians, and n is a discrete-time white Gaussian noise sequence with power spectral density σ_n^2. The beat carrier phase is related to

the apparent Doppler frequency, f_D in Hertz, according to the relationship $f_D(\tau_j) = \frac{1}{2\pi} \frac{d\theta(\tau)}{d\tau}\Big|_{\tau=\tau_j}$.

A receiver tracking the GPS signal model in (1) generates a pseudorange measurement that is formed by

$$\rho(\tau_j) = c \cdot [t_s(\tau_j) - t_c(\tau_j)],$$

where c is the speed of light and t_c is the GPS satellite vehicle (SV) time associated with the current code interval as indicated by the navigation data.

A model of the pseudorange measurement is expressible as

$$\rho(\tau_j) = c \cdot \{\delta t_r(\tau_j) - \delta t_s[\tau_j - \delta t_r(\tau_j) - \delta t_{\text{TOF}}(\tau_j)]\} + c\delta t_{\text{TOF}}(\tau_j) + n_\rho(\tau_j), \quad (2)$$

where δt_{TOF} is the time-of-flight (TOF) of the transmitted signal, δt_r is the receiver clock bias, δt_s is the GPS SV clock bias (typically a second-order model with relativistic effects), and n_ρ is the pseudorange noise. The TOF model is an implicit relationship, typically solved by recursion, and is given by

$$c\delta t_{\text{TOF}}(\tau_j) = \|\boldsymbol{r}_r[\tau_j - \delta t_r(\tau_j)] - \boldsymbol{r}_s[\tau_j - \delta t_r(\tau_j) - \delta t_{\text{TOF}}(\tau_j)]\|_2 + I(\tau_j) + T(\tau_j) + B,$$

where $\boldsymbol{r}_r$ is the receiver's position, $\boldsymbol{r}_s$ is the GPS SV's position, I models the ionospheric delay, T models the tropospheric delay, and B is the receiver line bias [7].

The carrier phase measurement model is similar to (2), except that the ionosphere advances the carrier phase and an ambiguity is introduced. The carrier phase model is given by

$$-\lambda\theta(\tau_j) = c \cdot \{\delta t_r(\tau_j) - \delta t_s[\tau_j - \delta t_r(\tau_j) - \delta t_{\text{TOF}}(\tau_j)]\} + c\delta t_{\text{TOF}}(\tau_j) - 2I(\tau_j) + \lambda\theta_0 + n_\theta(\tau_j), \quad (3)$$

where λ is the wavelength of the GPS carrier signal, θ_0 is the phase ambiguity, and n_θ is the phase noise.

III. GPS Receiver Stages

A GPS receiver consists of three main stages: acquisition, tracking, and navigation solution computation.

1. Acquisition: The objective of this stage is to determine which GPS SV is visible and subsequently determine a coarse estimate of the code phase and Doppler frequency. To achieve this, the receiver generates a replica of the different pseudo random noise (PRN) sequences. Each GPS SV transmits a unique PRN. For a particular SV, acquisition amounts to a two-dimensional (2D) search over code phase and Doppler frequency. The phase is determined by aligning the received PRN code with that of the receiver-generated replica. Given the GPS SV dynamics, it is sufficient to search over $f_D \in \{-10, 10\}$ kHz. The frequency steps, Δf_D, are chosen as a fraction of the the inverse of the code interval. For a coarse acquisition (C/A) code, such interval is 1 ms; hence, it suffices to choose $\Delta f_D \sim 100 - 500$ Hz. A model of the performed acquisition accumulation operation is given by

$$\mathbf{S}_k = \sum_{j=j_k}^{j_k+N_k-1} x(j)e^{-i\hat{\theta}(\tau_j)}C\left(\tau_j - \hat{t}_{s,k}\right), \quad (4)$$

where j_k is the minimum value of j that respects the bound $\hat{t}_{s,k} \leq \tau_j$, N_k is the number of samples in the kth subaccumulation interval, and $\hat{\theta}(\tau_j)$ and $\hat{t}_{s,k}$ are the beat carrier phase and code phase estimates, respectively. If the SV is visible, a plot of $\mathbf{S}_k$ will reveal a peak rising from the noise centered at the code phase and Doppler frequency. Deciding whether to accept the rising peak as an acquired signal versus just noise is typically done through a binary hypothesis test. The mechanization of (4) can be performed via a serial search or a parallel search. The parallel search can be either over frequency space or code phase space. The code phase space search is accomplished by exploiting the tremendous computational efficiency of the fast Fourier transform (FFT) [8].

2. Tracking: Starting with the coarse estimates found by the acquisition stage, the objective of this stage is to refine these estimates and track changes to them. When the GPS signal is being tracked, the navigation data message can be demodulated to get information about the GPS SV clock, ephemeris, and almanac. A block diagram of carrier-aided code tracking is illustrated in Fig. 1, where ω_c is the GPS signal carrier frequency in radians. The various building blocks are discussed in Section IV.

3. Navigation Solution: Pseudorange and carrier phase observables can be calculated from the tracking loop. Given the calculated observables and the GPS SV position and clock bias, the receiver's position and clock bias can be solved for from the nonlinear relationship relating these quantities as described in (2) and (3).

IV. GPS SDR Signal Tracking Building Blocks

A. Correlation

The correlation block essentially correlates the incoming signal described in (1) with the receiver-generated replicas of the code and carrier. Three versions of the code replicas are used in the correlation: prompt, early, and late. The early and late code replicas are generated by shifting the code from the estimated start time by a fraction of a chip $\sim \pm 0.5$. The correlation operation is performed according to (4) yielding the prompt, early, and late complex-valued subaccumulations $\tilde{\mathbf{S}}_{\mathsf{p},k}$, $\tilde{\mathbf{S}}_{\mathsf{e},k}$, and $\tilde{\mathbf{S}}_{\mathsf{l},k}$, respectively.

B. Coherent Summation

If the received signal's carrier-to-noise ration (C/N_0) is not high enough (less than ~ 45 dB-Hz), the three subaccumulations need to be summed coherently over M accumulation periods to yield $\mathbf{S}_{\mathsf{p},k}$, $\mathbf{S}_{\mathsf{e},k}$, and $\mathbf{S}_{\mathsf{l},k}$. Note that if $M > 1$, one needs to be careful not to straddle data bit transitions, which will effectively annihilate the subaccumulations. For C/A codes, for instance, within each data bit there are 20 codes. To avoid straddling bit transition boundaries, one of two strategies is adopted: (i) achieve data bit synchronization by looking at a long series of $\mathbf{S}_{\mathsf{p},k}$ or (ii) acquire a data-free (pilot) signal.
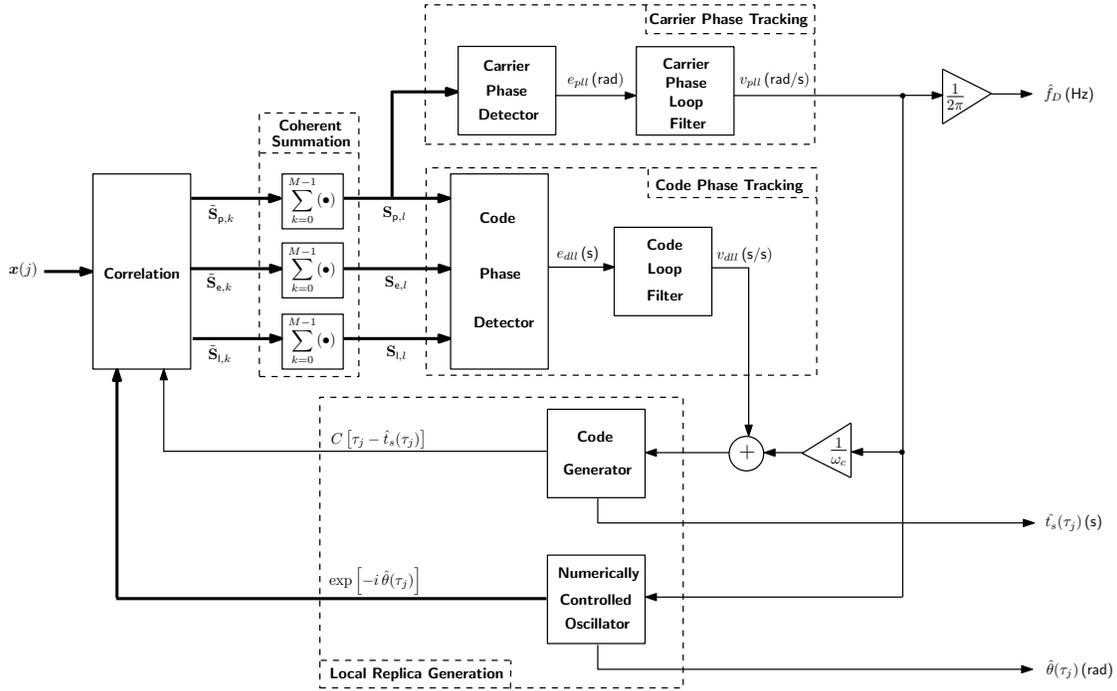
Fig. 1. Block diagram of a single-channel GPS signal tracking. Thick lines indicate complex values.

C. Carrier Phase Tracking

Carrier phase tracking is achieved via a phase lock loop (PLL) or a frequency lock loop (FLL). A PLL/FLL is composed of a carrier phase/frequency detector (discriminator) and a loop filter. Typical PLL/FLL discriminators are: arc-tangent (AT), conventional Costas (CC), decision-directed (DD), and decision-directed arc-tangent (DDAT). The DDAT discriminator can be used if the data bits in effect are known (e.g. from a side channel). Typically, the loop filter is chosen to be of second or third order. The filter's transfer function can be designed in continuous-time (s-domain) and then discretized to the loop update rate or designed directly in discrete-time (z-domain). The state space realization of second and third order loop filters consist of one and two states, respectively. To determine the PLL/FLL filter's initial state, it is assumed that the filter has reached steady-state and the filter's driving input (error signal) is zero.

D. Code Phase Tracking

Code phase tracking is achieved via a delay lock loop (DLL), consisting of a code phase detector (discriminator) and a code loop filter. DLL discriminators are categorized into coherent (early-minus-late) and non-coherent (early-minus-late power, early-minus-late envelope, and dot product) [9]. Since GPS code tracking is always aided by the PLL, a low-order loop filter with a small loop bandwidth is required. It is typical to choose such a filter to be first-order, i.e. a simple gain block.

E. Local Replica Generation

The numerically-controlled oscillator (NCO) and code generator create the local carrier and code replica signals for cor-

relation, respectively. Both the code and carrier are repetitive signals which at any instant in time can be characterized by a phase. The model for such cyclic signals is $f\left(\text{fpart}\left(\phi\right)\right)$ where ϕ is the instantaneous phase of the signal in cycles, $\text{fpart}\left(\cdot\right)$ denotes the fractional part of the argument, and $f\left(\cdot\right)$ is a function with domain $[0, 1)$. For an NCO,

$$f\left(\phi\right) = e^{-i2\pi\phi},$$

and for the code generator,

$$f\left(\phi\right) = C\left(N_c T_c \phi\right).$$

In RT applications, cyclic signals are efficiently generated using look-up tables (LUT), where the domain of f is discretized into N_{LUT} elements and N_{LUT} is typically a power of two. The LUT model for cyclic signals is

$$f\left[\frac{1}{N_{\text{LUT}}} \cdot \text{floor}\left[N_{\text{LUT}} \cdot \text{fpart}\left(\phi\right)\right]\right],$$

and for large enough N_{LUT}, the distortion is minimal.

A cyclic signal is also associated with an instantaneous frequency f in cycles per second, where $f = \frac{d\phi}{dt}$. Let ϕ_k and f_k be the phase and frequency at the beginning of the kth subaccumulation, respectively. During the subaccumulation, the phase is propagated assuming a constant frequency, which is an acceptable approximation for short subaccumulation intervals $N_k T$. At the end of the subaccumulation, the frequency may be updated by the tracking loop depending on the number of coherent summations. The phase is updated according to

$$\phi_{k+1} = \phi_k + f_k N_k T.$$

V. LabVIEW-Based GPS SDR

The acquisition and tracking modes of the GPS SDR were developed graphically in LabVIEW. Each mode was expressed as a separate so-called virtual instrument (VI), whose inputs and outputs are illustrated in Fig. 2-3. The tracking LabVIEW code closely resembled the block diagram in Fig. 1. The GPS SDR Configuration cluster contains the PRN code replicas, the designed PLL and DLL filters, and several constants (e.g. carrier frequency, nominal chipping frequency, sampling rate of received baseband in-phase and quadrature (IQ) stream, etc). The PRN number specifies the GPS SV whose signal is to be acquired and tracked. The acquisition VI outputs the 2D correlation sequence, C/N_0, and estimated Doppler frequency, local carrier phase, and code start time. The interactive front panel associated with the acquisition mode is illustrated in Fig. 4. Upon successful acquisition of a GPS signal, the Doppler frequency, local carrier phase, and code start time estimates are communicated with the tracking VI. The tracking VI tracks the GPS signal and refines such estimates. Moreover, the tracking VI computes the correlation sequence over different code replica offsets, pseudorange, and IQ data bit streams. The tracking VI outputs are shown in Fig. 5. Note that after a transient phase, the tracking loop estimates converge.
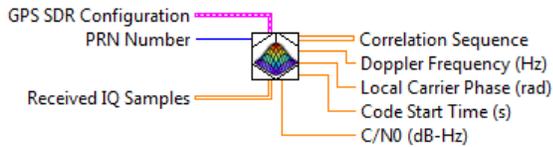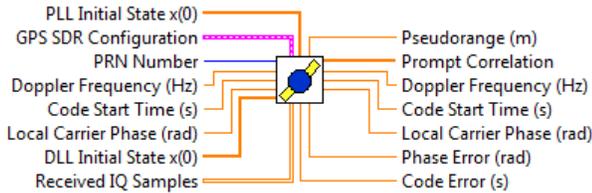

Fig. 2.   Acquisition VI

Fig. 3.   Tracking VI

VI. Conclusion

This paper presented a graphical approach to GPS SDR implementation. GPS acquisition and tracking loops are conceptualized as block diagrams, making a graphical approach a natural implementation choice. Other benefits of the graphical approach include modularity, reconfigurability, and platform independence. The developed SDR provides an abstraction level that enables future researchers to focus on studying the different building blocks separately. Also, the developed SDR could be extended to advanced navigation applications, such as multi-constellation global navigation satellite systems (GNSS) tracking, vector tracking, and signals of opportunity (SOP)-based navigation [10].

References

[1] D. M. Akos, "The role of Global Navigation Satellite System (GNSS) software radios in embedded systems," *GPS Solutions*, vol. 7, no. 1, pp. 1–4, 2003.
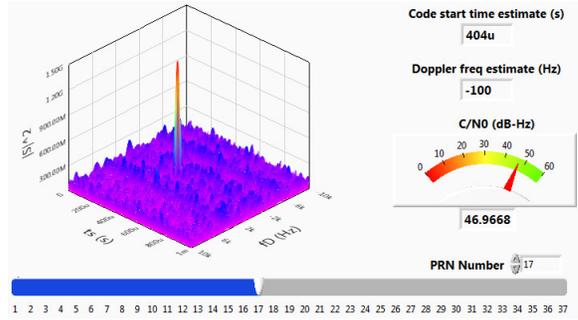
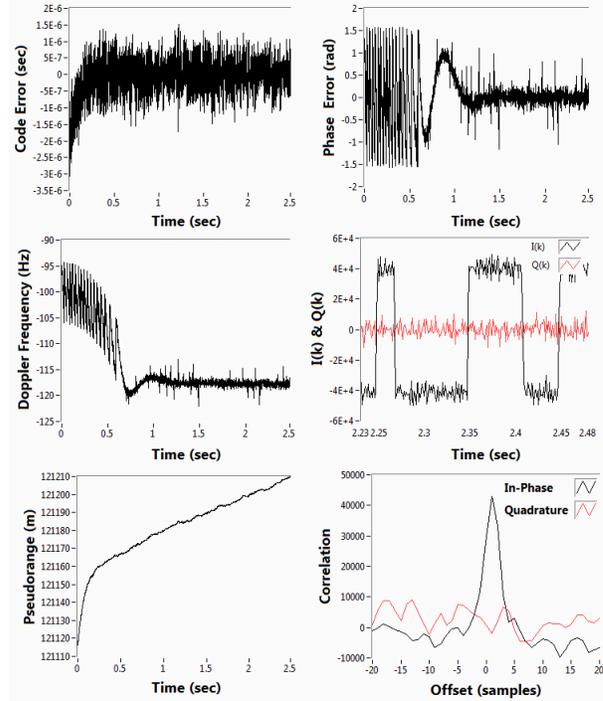Fig. 4.   Front panel of acquisition mode

Fig. 5.   Front panel of tracking mode

[2] K. Borre, D. Akos, N. Bertelsen, P. Rinder, and S. Jensen, *A Software-defined GPS and Galileo Receiver: A Single-frequency Approach.* Birkhäuser, 2007.

[3] T. Humphreys, M. Psiaki, P. Kintner, and B. Ledvina, "GNSS receiver implementation on a DSP: Status, challenges, and prospects," in *Proc. of ION GNSS*, Sep. 2006, pp. 1567–1575.

[4] T. E. Humphreys, J. Bhatti, T. Pany, B. Ledvina, and B. O'Hanlon, "Exploiting multicore technology in software-defined GNSS receivers," in *Proc. of ION GNSS*, Sep. 2009, pp. 326–338.

[5] G. Hamza, A. Zekry, and I. Motawie, "Implementation of a complete GPS receiver using Simulink," *IEEE Circuits and Systems Magazine*, vol. 9, no. 4, pp. 43–51, 2009.

[6] A. Soghoyan and D. Akopian, "A labVIEW-based GPS receiver development and testing platform with DSP peripherals: Case study with c6713 DSK," *Journal of Global Positioning Systems*, vol. 11, no. 2, pp. 127–144, 2012.

[7] M. Psiaki and S. Mohiuddin, "Modeling, analysis, and simulation of GPS carrier phase for spacecraft relative navigation," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1628–1639, 2007.

[8] D. van Nee and A. Coenen, "New fast GPS code-acquisition technique using FFT," *Electronics Letters*, vol. 27, no. 2, pp. 158–160, 1991.

[9] E. Kaplan and C. Hegarty, *Understanding GPS Principles and Applications*, 2nd ed.   Norwood, MA: Artech House, 2006.

[10] Z. Kassas, "Collaborative opportunistic navigation," *IEEE Aerospace and Electronic Systems Magazine*, vol. 28, no. 6, pp. 38–41, Jun. 2013.