Computationally Efficient Direct Position Estimation via Low Duty-Cycling

Yuting Ng and Grace Xingxin Gao
University of Illinois at Urbana-Champaign

BIOGRAPHIES

Yuting Ng is a graduate student in the Aerospace Engineering Department at the University of Illinois at Urbana-Champaign. She received her B.S. degree in electrical engineering, graduating with university honors, from the same university. Her research interests are in advanced signal tracking, navigation, control, robotics, RADAR and UAVs.

Grace Xingxin Gao received her B.S. degree in mechanical engineering and her M.S. degree in electrical engineering from Tsinghua University, Beijing, China in 2001 and 2003. She received her PhD degree in electrical engineering from Stanford University in 2008. From 2008 to 2012, she was a research associate at Stanford University. Since 2012, she has been an assistant professor in the Aerospace Engineering Department at University of Illinois at Urbana-Champaign. Her research interests are systems, signals, control, and robotics. She is a senior member of IEEE and a member of ION.

Abstract—**Direct Position Estimation (DPE) is an unconventional GPS positioning technique that directly estimates the GPS navigation solution from the GPS raw signal. In our prior work, we have proposed and implemented a novel DPE receiver architecture that efficiently estimates and tracks a comprehensive underlying signal and navigation parameter set of three-dimensional (3D) position, clock bias, 3D velocity and clock drift without additional aiding information from an external source.**

To further reduce the computational load of DPE, we propose low duty-cycling of our DPE receiver architecture. Our duty-cycled DPE receiver algorithm consists of a computationally efficient DPE measurement update and a DPE time update that reduces the accumulation of signal tracking errors. Our DPE measurement update optimizes over navigation parameter subsets, combines and computes batch signal replica generation and correlation using Fast Fourier Transforms and estimates the navigation solution using a correlation-weighted mean. Our DPE time update iteratively predicts and updates the signal code phase and carrier doppler frequency parameters using updated satellite positions, velocities, clock biases and clock drifts calculated using the satellite broadcast ephemerides.

We implemented our duty-cycled DPE receiver architecture using a commercial frontend and our software platform - PyGNSS. We conducted both static and dynamic open-sky experiments. From the signal tracking results of the static experiment, we demonstrate that our duty-cycled DPE receiver, with duty-cycling as low as 2%, shows similar performance to continuous DPE. From the positioning results of the dynamic experiment, we demonstrate that our duty-cycled DPE receiver, with duty-cycling as low as 2%, successfully tracks a moving vehicle; with an accuracy that outperforms continuous vector tracking under signal attenuation.

Yuting Ng and Grace Xingxin Gao are with the Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA. E-mails: yng5@illinois.edu, gracegao@illinois.edu.

I. INTRODUCTION

Direct Position Estimation (DPE) is an unconventional GPS positioning technique that directly estimates the GPS navigation solution from the GPS raw signal [1], [2]. DPE works by performing correlations of the received signal against the expected signal reception at multiple candidate navigation points. The navigation solution is then estimated as the navigation candidate with the highest vector correlation, a correlation result accumulated across multiple satellite signals, as shown in Fig. 1.

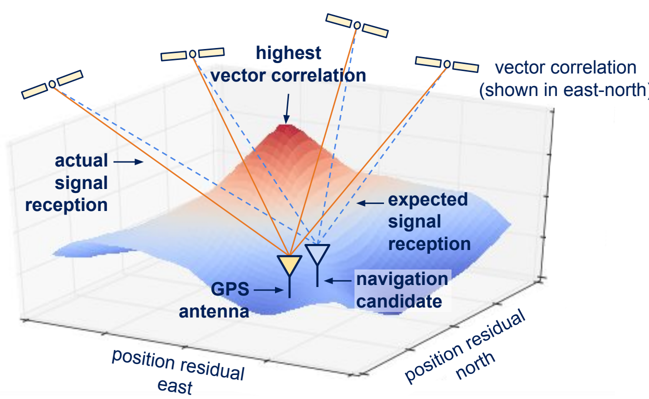

Fig. 1. Vector correlation distribution shown in the east-north navigation search dimensions. For each navigation candidate, the vector correlation is generated as an accumulation of signal correlations across all satellites in view. The navigation solution is estimated as the navigation candidate with the highest vector correlation.

In our prior work, we have proposed and implemented a novel DPE receiver architecture that efficiently estimates and tracks a comprehensive underlying signal and navigation parameter set of three-dimensional (3D) position, clock bias, 3D velocity and clock drift without additional aiding information from an external source [3], [4]. However, this implementation still requires significant computational time and is thus only suitable for delay tolerant applications. To further reduce the computational load of DPE, we propose low duty-cycling of our DPE receiver architecture.

The rest of this paper is organized as follows. Section II discusses related work on improving DPE's computational efficiency and related work on duty-cycling of conventional GPS positioning. Section III describes our approach, computationally efficient DPE via low duty-cycling. Section IV describes our implementation using our software platform - PyGNSS. Section V presents our experiment results. Through our experiments, we show that our duty-cycled DPE receiver with duty-cycling as low as 2% demonstrates similar performance to continuous DPE under a static, open-sky scenario. In addition, our duty-cycled DPE receiver tracks a moving vehicle under a dynamic, open-sky scenario. Finally, Section VI summarizes the paper.

II. Related Work

This section describes related work on improving DPE's computational efficiency and related work on duty-cycling of conventional GPS positioning.

A. Improving DPE's Computational Efficiency

Related work on improving DPE's computational efficiency include use of sparsely-located navigation candidates [5], reducing the number of required navigation candidates [6]–[8], efficient techniques for calculating multiple correlations [9] and efficient techniques for estimating the navigation solution given the vector correlation distribution [5].

To improve efficiency of initialization, coarse-grid search [5] and initialization using Assisted-GPS (A-GPS) [10] was proposed.

To improve efficiency after initialization near the main vector correlation peak, optimization over navigation subsets was proposed. One such implementation used parameter grouping by position-clock-bias and velocity-clock-drift [5], [6]. In this manner, parameters within subsets are strongly correlated while parameters across subsets are weakly correlated. The subsets can then be separately optimized. Other subset optimization implementations used parameter grouping where each parameter formed its own subset [11], [12], Space-Alternating Generalized Expectation-Maximization (SAGE) [7] and Accelerated Random Search (ARS) [8]. These subset optimization methods reduced the number of required navigation candidates and thus the overall computational load. However, they were sensitive to initialization [7]. Subset optimization required initialization near the main navigation peak. To increase the chances of initialization near the main navigation correlation peak for subsequent DPE measurement updates, motion model filtering was performed between DPE measurement updates [6], [13], [14].

For faster replica signal generation and correlation against the received signal, the processes were combined and batch performed using Fast Fourier Transforms (FFTs) [9].

Finally, instead of performing a computationally intensive search for the vector correlation peak, using a correlation-weighted mean improves computational efficiency [5].

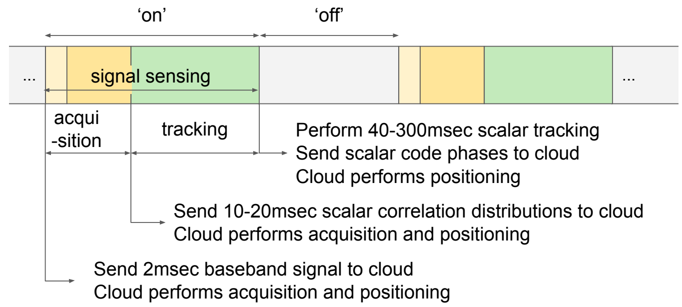

Fig. 2. State-of-the-art architectures duty-cycling the GPS receiver's signal tracking and positioning processes. Most implementations employ open-loop snapshot processing with cloud offloading [22], [25], [26].

B. Duty-Cycling of Conventional GPS Positioning

Conventional methods of duty-cycling the GPS receiver switched the entire GPS receiver 'on' and 'off' [15] based on the signal environment [16], [17] and receiver dynamics [18]–[20].

State-of-the-art implementations duty-cycle the GPS receiver's signal tracking and positioning processes, as shown in Fig. 2. Most implementations employ open-loop snapshot processing with cloud offloading [21]. Implementations differ in the partitioning of the GPS processes between the GPS receiver and the cloud. With different partitioning, the data communicated with the cloud differs.

In one implementation, the GPS receiver performs only signal sensing. The GPS receiver then sends a two millisecond baseband signal to the cloud [22]. The cloud uses A-GPS and Coarse Time Navigation (CTN) [23] to complete acquisition and positioning [22], [24].

To reduce the amount of data transmitted, another implementation proposed sending the scalar correlograms from acquisition to the cloud [25]. The cloud then completes acquisition and positioning [25].

To increase positioning precision and further reduce the amount of data transmitted, another implementation proposed preliminary scalar tracking then sending the tracked scalar code phases to the cloud [26]. With ten milliseconds of preliminary scalar tracking, the average positioning error is around 40 meters. With another 300 milliseconds of preliminary scalar tracking, the average positioning error is comparable to that of a continuously tracking GPS receiver [26]. This however offers minimal benefit as 300 milliseconds is on the order of the initial time taken to lock onto a satellite signal from acquisition.

Without preliminary scalar tracking, duty-cycling implementations based on conventional GPS processing are unable to achieve the accuracy and precision comparable to that of a continuously tracking GPS receiver [27].

Conventional signal tracking and navigation schemes such as scalar and vector tracking use intermediate measurements of scalar code and carrier discriminations [28], [29]. These lead to a combination of higher measurement noise, limited discriminator range and more stringent loop stability criteria [30] that result in the signal tracking loops of scalar and vector tracking receivers being unsuitable for duty-cycling. A recent

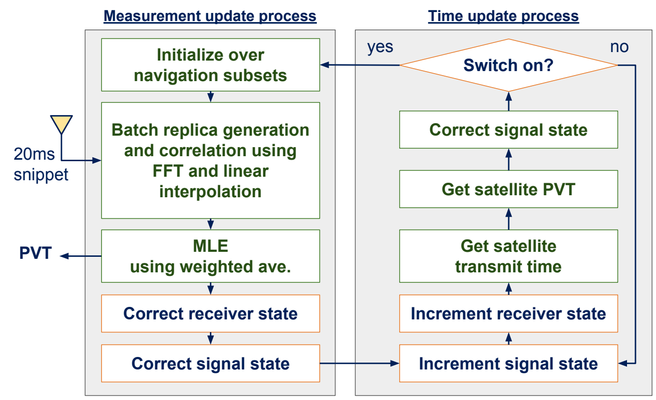

Fig. 3. Block diagram of our duty-cycled DPE receiver architecture.

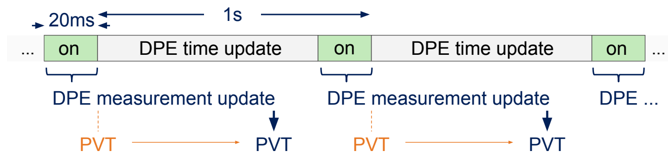

Fig. 4. Overview of our duty-cycled DPE receiver architecture with our DPE measurement update and our DPE time update.

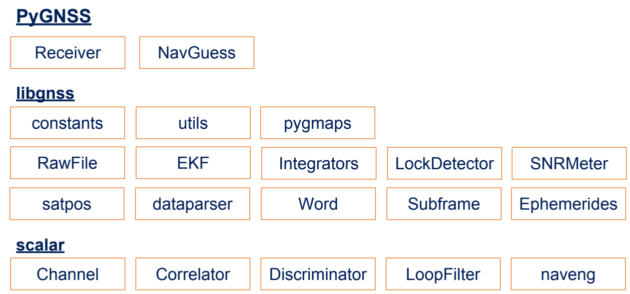

Fig. 5. Our software platform, PyGNSS, is modular and comprehensive. It performs signal conditioning, tracking and positioning.

result showed that for a 1Hz positioning update rate, to achieve accuracy comparable to a continuous scalar tracking receiver, the scalar tracking receiver may only be duty-cycled with a duty-cycling factor of 80% [27].

On the other hand, DPE jointly processes available satellite signals to directly estimate the navigation solution [31]. This leads to a combination of lower measurement noise and no linearisation of navigation equations, making DPE receiver architectures suitable for duty-cycling. Using DPE, we perform duty-cycling as low as 2% while providing position updates at a 1Hz rate, with accuracy comparable to a continuous DPE receiver.

III. Our Approach: Duty-Cycled Direct Positioning

We propose a novel duty-cycled DPE receiver architecture with accuracy comparable to that of a continuously tracking DPE receiver. Our duty-cycled DPE receiver architecture consists of a computationally efficient DPE measurement update and a DPE time update that mitigates the accumulation of signal tracking errors.

Our computationally efficient DPE measurement update [3], [4] uses optimizations over navigation parameter subsets, combines and computes batch signal replica generation and correlation using Fast Fourier Transforms and estimates the navigation solution using a correlation-weighted mean instead of a computationally intensive peak search. The exact computation time depends on signal sampling parameters and the number of satellites in view.

During the time required to compute the DPE measurement update, we iteratively predict satellite movements, using the satellite broadcast ephemerides, and update the signal code phase and carrier doppler frequency parameters. This DPE time update reduces the accumulation of signal tracking errors during the time in which the DPE receiver is not performing any DPE measurement update.

The block diagram showing the processes within the DPE measurement update and the DPE time update is given in Fig. 3.

IV. Implementation Using PyGNSS

We implemented our duty-cycled DPE receiver architecture with the following parameters:

- perform 1 DPE measurement update every 1 sec on a 20 msec GPS raw signal snippet during which the GPS receiver is 'on'.
- perform 49 DPE time updates during the 980 msec in which the GPS receiver is 'off'.

Our computationally efficient DPE measurement update takes approximately 1 sec to perform. During this time, we perform our DPE time updates to mitigate the accumulation of signal tracking errors. Using our duty-cycled DPE receiver architecture, we provide positioning updates at a 1 Hz rate with a duty-cycling factor of 2%. We mitigate the time delay required to obtain the positioning update through extrapolating the navigation results by the time delay. An overview of our duty-cycled DPE receiver architecture is shown in Fig. 4.

We implemented our duty-cycled DPE receiver using commercially available signal sensing frontend components and our software platform - PyGNSS. Available modules within PyGNSS are shown in Fig. 5. Our duty-cycled DPE receiver architecture and its accompanying computer algorithm are shown in Fig. 6.

Referring to the receiver architecture in Fig. 6, our duty-cycled DPE Receiver object has a NavGuess object, a RawFile object and multiple Channel objects, with each Channel having its own Correlator and Ephemerides objects.

Using NavGuess, the Receiver initializes 25^4 evenly-spaced navigation guesses in the position-clock-bias navigation subset around its current navigation state. These navigation guesses are separated by 50 m within each navigation search dimension. Using NavGuess, the Receiver also initializes 25^4 evenly-spaced navigation guesses in the velocity-clock-drift navigation subset around its current navigation state. These

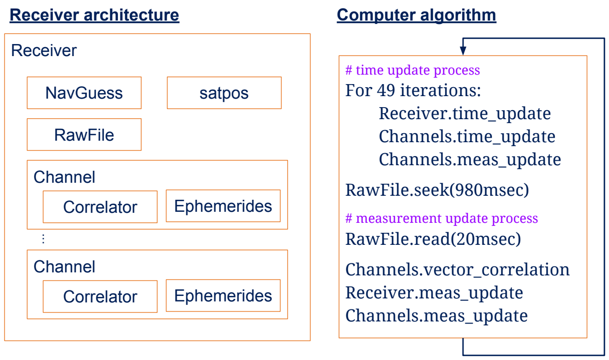

Fig. 6. Duty-cycled DPE receiver architecture and accompanying computer algorithm implemented using PyGNSS.

guesses are separated by 2.5 ms^{-1} within the three velocity search dimensions and 0.03 ms^{-1} within the clock drift search dimension.

Our duty-cycled DPE Receiver also has a satpos module that calculates satellite positions, clock biases, velocities and clock drifts based on the Channel's Ephemerides and a satellite transmit time. The satellite transmit time is obtained from the signal code phase attribute maintained within each Channel.

Referring to the computer algorithm in Fig. 6, our duty-cycled DPE Receiver cycles between a DPE time update process and a DPE measurement update process.

Within the DPE time update process, the duty-cycled DPE Receiver performs a 'Receiver.time_update', a 'Channels.time_update' and a 'Channels.meas_update'. 'Receiver.time_update' increments the Receiver's position-clock-bias based on its velocity-clock-drift using a constant acceleration motion model. 'Channels.time_update' increments the signal code phases by the signal carrier doppler frequencies. The incremented signal code phases are translated into updated satellite transmit times which we use to calculate updated satellite positions, clock biases, velocities and clock drifts. The updated satellite navigation states are used to update signal code phases and carrier doppler frequencies in 'Channels.meas_update'.

'RawFile.seek(980msec)' skips a 980 msec GPS raw signal snippet. 'RawFile.read(20msec)' reads a 20 msec GPS raw signal snippet. The duty-cycled DPE Receiver then cycles into the DPE measurement update process.

Within the DPE measurement update process, 'Channels.vector_correlation' performs correlations against the received signal and accumulates the correlations across satellites to form the vector correlation. 'Receiver.meas_update' calculates the DPE navigation solution using the vector correlation and updates the Receiver's navigation parameters. 'Channels.meas_update' then updates the Channel's signal parameters based on the updated Receiver's navigation parameters.

The duty-cycled DPE receiver repeatedly cycles between the DPE measurement update and DPE time update.

V. Experiment Results

With our duty-cycled DPE receiver, we conducted both static and dynamic open-sky experiments to verify its feasibility and validate its accuracy. To avoid effects of increased tracking robustness and noise reduction associated with filtering, we did not smooth our navigation results with a navigation filter.

The static open-sky experiment was conducted on the roof of Talbot laboratory, University of Illinois at Urbana-Champaign (UIUC). We used an AntCom 3GNSSA4-XT-1 GNSS antenna [32], an Ettus Research N210 Universal Software Radio Peripheral (USRP) [33], equipped with a DBSRX2 daughterboard [34], triggered by a Microsemi Quantum SA.45s Chip Scale Atomic Clock (CSAC) [35]. We used zero-IF complex sampling with a 5.0MHz sampling frequency.

The dynamic open-sky experiment was conducted south of campus, UIUC. Apart from switching to a low-cost small ANT-555 magnetic mount active patch antenna [36], the signal sensing frontend components and signal sampling parameters are the same.

A. *Static Experiment Results*

The signal tracking results from the static, open-sky experiment show similar performance to continuous DPE. The signal tracking residuals obtained by performing a difference of code phase and carrier doppler frequency results from 2% duty-cycled DPE against continuous DPE is noisy but unbiased, as shown in Fig. 7.

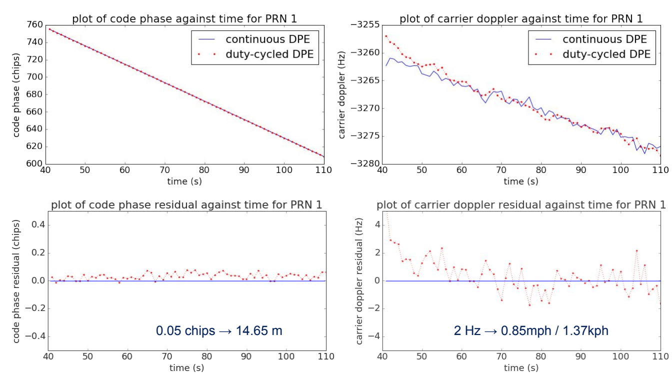

Fig. 7. Code phase and carrier doppler frequncy tracking results for PRN 1. 2% duty-cycled DPE demonstrate similar performance to continuous DPE.

B. *Dynamic Experiment Results*

The results from the dynamic, open-sky experiment show 2% duty-cycled DPE tracking a moving vehicle. Shown in Fig. 8 are the positioning results in the navigation domain. Each position marker on the Google Maps [37] plot is a DPE measurement update, each update separated by 1 second. 2% duty-cycled DPE accurately tracked the navigation states of a dynamic receiver.

The same experiment was repeated under signal attenuation of approximately 15dB per channel. Under signal attenuation, our duty-cycled DPE receiver outperforms continuous vector tracking, as shown in Fig. 9.

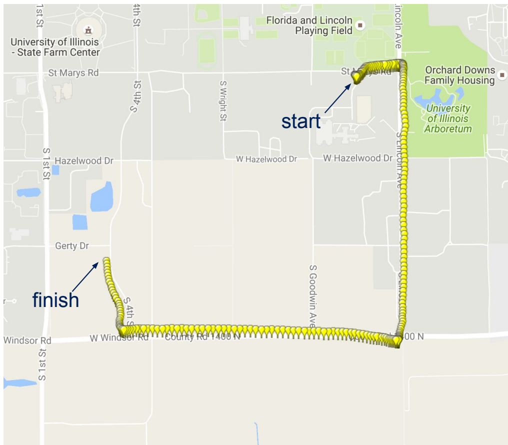

Fig. 8. Our duty-cycled DPE receiver, with duty-cycling as low as 2%, accurately tracked the route taken by a moving vehicle.

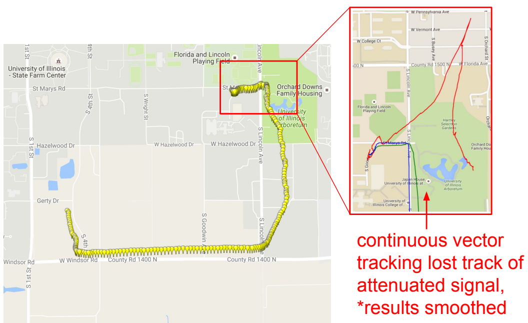

Fig. 9. Under signal attenuation, our duty-cycled DPE receiver outperforms continuous vector tracking.

VI. Conclusion

In summary, we improved the computational efficiency of DPE via low duty-cycling. Our duty-cycled DPE receiver architecture uses a computationally efficient DPE measurement update and a DPE time update that mitigates accumulation of signal tracking errors. We implemented our duty-cycled DPE receiver architecture using our software platform - PyGNSS. We conducted experiments in both static and dynamic receiver scenarios. Our experimental results show that we have achieved duty-cycling at low as 2% while maintaining similar positioning performance to a continuous DPE receiver.

References

[1] P. Closas, C. Fernández-Prades, and J. Fernández-Rubio, "On the maximum likelihood estimation of position," in *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006), Fort Worth, TX*, 2006, pp. 1800–1810.

[2] P. Axelrad, J. Donna, and M. Mitchell, "Enhancing GNSS acquisition by combining signals from multiple channels and satellites," in *Proceedings of the 22nd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2009), Savannah, GA*, 2009, pp. 2617–2628.

[3] Y. Ng and G. X. Gao, "Mitigating jamming and meaconing attacks using direct GPS positioning," in *Proceedings of IEEE/ION PLANS 2016, Savannah, GA*. IEEE/ION, 2016, pp. 1021–1026.

[4] Y. Ng, A. H.-P. Chu, and G. X. Gao, "Efficient GPS direct position estimation," *IEEE Transactions on Aerospace and Electronic Systems*, Submitted.

[5] B. Bradley, P. Axelrad, J. Donna, and S. Mohiuddin, "Performance analysis of collective detection of weak GPS signals," in *Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2010), Portland, OR*, 2010, pp. 3041–3053.

[6] L. R. Weill, "A high performance code and carrier tracking architecture for ground-based mobile GNSS receivers," in *Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2010), Portland, OR*, 2010, pp. 3054–3068.

[7] P. Closas, C. Fernández-Prades, and J. Fernández-Rubio, "ML estimation of position in a GNSS receiver using the SAGE algorithm," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP'07*, vol. 3. IEEE, 2007, pp. 1041–1045.

[8] P. C. Gómez, "Bayesian signal processing techniques for GNSS receivers," Ph.D. dissertation, Universitat Politecnica de Catalunya, 2009.

[9] J. W. Cheong, J. Wu, A. G. Dempster, and C. Rizos, "Efficient implementation of collective detection," in *International Global Navigation Satellite Systems Society (IGNSS Symposium 2011)*, 2011, pp. 15–17.

[10] ——, "Assisted-GPS based snap-shot GPS receiver with FFT-accelerated collective detection: Time synchronisation and search space analysis," in *Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012), Nashville, TN*, 2012, pp. 2357–2370.

[11] J. Liu, X. Cui, M. Lu, and Z. Feng, "A direct position tracking loop for GNSS receivers," in *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011), Portland, OR*, 2011, pp. 3634–3643.

[12] ——, "Direct position tracking loop based on linearised signal model for global navigation satellite system receivers," *IET Radar, Sonar & Navigation*, vol. 7, no. 7, pp. 789–799, 2013.

[13] R. DiEsposti, "GPS PRN code signal processing and receiver design for simultaneous all-in-view coherent signal acquisition and navigation solution determination," in *Proceedings of the 2007 National Technical Meeting of The Institute of Navigation, San Diego, CA*, 2007, pp. 91–103.

[14] F. D. Nunes, J. Marcal, and F. M. Sousa, "Low-complexity VDLL receiver for multi-GNSS constellations," in *2010 5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*. IEEE, 2010, pp. 1–8.

[15] C. Y. Lau, D. G. Farmer, K. A. Martin, and E. B. Rodal, "GPS receiver having a low power standby mode," Jan. 7 1997, US Patent 5,592,173.

[16] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive GPS-based positioning for smartphones," in *Proceedings of the 8th international conference on mobile systems, applications, and services*. ACM, 2010, pp. 299–314.

[17] R. Jurdak, P. Corke, D. Dharman, and G. Salagnac, "Adaptive GPS duty cycling and radio ranging for energy-efficient localization," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2010, pp. 57–70.

[18] Z. Jia, S. A. Kurethaya, and C.-S. Wang, "Navigational signal tracking in low power mode," Dec. 7 2010, US Patent 7,847,726.

[19] M. Leclercq, I. Spyropoulos, A. K. Anandakumar, and C. Ling, "Intermittent tracking for GNSS," Sep. 10 2013, US Patent 8,531,333.

[20] Y. Tawk, P. Tomé, C. Botteron, Y. Stebler, and P.-A. Farine, "Implementation and performance of a GPS/INS tightly coupled assisted PLL architecture using MEMS inertial sensors," *Sensors*, vol. 14, no. 2, pp. 3768–3796, 2014.

[21] N. U. Linty, "Snapshot estimation algorithms for GNSS mass-market receivers," Ph.D. dissertation, Politecnico di Torino, 2015.

[22] J. Liu, B. Priyantha, T. Hart, H. S. Ramos, A. A. Loureiro, and Q. Wang, "Energy efficient GPS sensing with cloud offloading," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 2012, pp. 85–98.

[23] F. v. Diggelen, *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech House, Boston, 2009.

[24] D. Borio, "A statistical theory for GNSS signal acquisition," Ph.D. dissertation, Politecnico Di Torino, 2008.

[25] P. K. Misra, W. Hu, Y. Jin, J. Liu, A. Souza de Paula, N. Wirstrom, and T. Voigt, "Energy efficient GPS acquisition with sparse-GPS," in *Proceedings of the 13th international symposium on Information processing in sensor networks*. IEEE Press, 2014, pp. 155–166.

[26] H. S. Ramos, T. Zhang, J. Liu, N. B. Priyantha, and A. Kansal, "LEAP: a low energy assisted GPS for trajectory-based services," in *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 2011, pp. 335–344.

[27] V. Bellad, "Intermittent GNSS signal tracking for improved receiver power performance," Ph.D. dissertation, University of Calgary, 2015.

[28] P. W. Ward, J. W. Betz, and C. J. Hegarty, *Understanding GPS: Principles and Applications Second Edition*. Artech House, Boston, 2006, ch. Satellite signal acquisition, tracking, and data demodulation, pp. 153–240.

[29] J. J. Spilker, “Fundamentals of signal tracking theory,” *Progress in Astronautics and Aeronautics*, vol. 163, pp. 245–328, 1996.

[30] P. W. Ward and T. D. Fuchser, “Stability criteria for GNSS receiver tracking loops,” *Navigation*, vol. 61, no. 4, pp. 293–309, 2014.

[31] P. Axelrad, B. K. Bradley, J. Donna, M. Mitchell, and S. Mohiuddin, “Collective detection and direct positioning using multiple GNSS satellites,” *NAVIGATION, Journal of The Institute of Navigation*, vol. 58, no. 4, pp. 305–321, 2011.

[32] Antcom Corporation. (2014, jun) 3.5in dia. GNSS antenna for global navigation satellite systems (GNSS) applications P/N: 3GNSSA4-XT-1 (front/back fixed mount). [Online]. Available: http://www.antcom.com/documents/catalogs/Page/3GNSSA4-XT-1_GNSSAntennas1.pdf

[33] Ettus Research. (2012, jun) USRP n200/n210 networked series. [Online]. Available: https://www.ettus.com/content/files/07495_Ettus_N200-210_DS_Flyer_HR_1.pdf

[34] ——. (2012, jun) DBSRX2 800-2300 mhz rx. [Online]. Available: http://www.ettus.com/product/details/DBSRX2

[35] Microsemi Corporation. (2012, jun) Quantum SA.45s chip scale atomic clock (CSAC). [Online]. Available: http://www.microsemi.com/document-portal/doc_download/133305-quantum-sa-45s-csac

[36] On Shine Enterprise Co. Ltd. (2012, jun) ON SHINE G.P.S. antenna ANT-555. [Online]. Available: http://www.onshine.com.tw//www/ANT555-2010.pdf

[37] Google Developers. (2016, jan) Google Maps Javascript API. [Online]. Available: https://developers.google.com/maps/documentation/javascript/