AFRL-RV-PS-
TR-2019-0049

GLOBAL NAVIGATION SATELLITE SYSTEM (GNSS) VECTOR CORRELATOR: HARNESSING SATELLITE DIRECTION COSINE DIVERSITY FOR ROBUST POSITIONING

Grace Gao

University of Illinois
Department of Aerospace Engineering
506 S Wright St, 364 Henry Administration Building
Urbana, IL 61801-3620

07 May 2019

Final Report

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

AIR FORCE RESEARCH LABORATORY
Space Vehicles Directorate
3550 Aberdeen Ave SE
AIR FORCE MATERIEL COMMAND
KIRTLAND AIR FORCE BASE, NM 87117-5776

DTIC COPY

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by AFMC/PA and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RV-PS-TR-2019-0049 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//SIGNED//	//SIGNED//
Dr. Lawrence M. Robertson Program Manager, AFRL/RVBY	Dr. Thomas R. Caudill, Chief AFRL Geospace Technologies Division

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
07-05-2019	Final Report	31 Jul 2015 – 07 May 2019

4. TITLE AND SUBTITLE
Global Navigation Satellite System (GNSS) Vector Correlator: Harnessing Satellite Direction Cosine Diversity For Robust Positioning

5a. CONTRACT NUMBER
FA9453-15-1-0075

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
63401F

6. AUTHOR(S)
Grace Gao

5d. PROJECT NUMBER
2181

5e. TASK NUMBER
PPM00036181

5f. WORK UNIT NUMBER
EF130536

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Illinois
Department of Aerospace Engineering
506 S Wright St, 364 Henry Administration Building
Urbana, IL 61801-3620

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Space Vehicles Directorate
3550 Aberdeen Avenue SE
Kirtland AFB, NM 87117-5776

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RVBY

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-RV-PS-TR-2019-0049

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited. (AFMC-2019-0472 dtd 01 Aug 19)

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The contractor investigated vector correlator as an alternative to the bank of parallel scalar correlators typically found in a conventional tracking receiver design. The vector correlator is different from vector tracking architectures by moving the correlation peak search into the physical domain: the three spatial dimensions and the additional time dimension. After the receiver's navigation filter has produced an estimate of the next anticipated receiver state, differential offsets from this state are generated in all four dimensions and translated into pseudorange offsets via the receiver-satellite geometry. With external knowledge of the navigation data bit stream for each satellite, precise code and carrier replicas are generated for each differential offset, and a composite replica signal is used to correlate against the raw signal arriving from the receiver front-end. Results from this spatial correlation peak search are fed back to the navigation filter, closing the loop.

15. SUBJECT TERMS
Satellite navigation, GPS, vector correlation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Dr. Lawrence M. Robertson
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(include area code)*
Unclassified	Unclassified	Unclassified	Unlimited	44	

This page is intentionally left blank.

Approved for public release: distribution is unlimited.

Table of Contents

List of Figures

List of Tables

This page is intentionally left blank.

1. INTRODUCTION

Global Positioning System (GPS), with its worldwide coverage [1] and its infrequent need for receiver calibration, has found widespread application acceptance in the aviation community [2, 3]. Conventional receiver architectures, such as the scalar tracking loop (STL) [4, 5] and the vector tracking loop (VTL) [6], are based on the two-step approach. That is, a receiver measures its ranges to the visible GPS satellites before triangulating a position- velocity-time (PVT) solution.

Popular for its simplicity and its proven service record, this two-step approach is nonetheless vulnerable in degraded signal environments [7]. More specifically, during episodes of signal multipath [8–10] or signal masking [11, 12], the signal-to-noise ratio (SNR) is reduced, rendering the range measurements unreliable or undetectable. These degrading effects often occur during critical flight phases, such as take-off and landing, due to their proximity to ground obstacles.

To address these challenges, a new GPS receiver architecture, built on the Direct Position Estimation (DPE) algorithm, has been introduced and shown to succeed in signal environments where scalar tracking-based receivers have failed [13-15]. While two-step methods such as scalar tracking generate intermediate measurements before computing navigation solutions, DPE-based approaches are one-step, directly estimating the state in the domain of navigation solutions. This is accomplished by finding the navigation-domain state whose expected signal most closely matches the signal actually received. DPE facilitates a deep coupling of the signals from different satellites, increases the effective signal power [16, 17], and utilizes weak signals that would have otherwise been discarded [7, 15].

This one-step approach is the solution of the maximum-likelihood objective function:

$$\hat{\boldsymbol{a}}_{\mathrm{ML}}, \hat{\boldsymbol{X}}_{\mathrm{ML}} = \arg\min_{\boldsymbol{a},\boldsymbol{X}} \|\boldsymbol{y} - D\boldsymbol{a}\|^2 \tag{1}$$

where $\hat{\boldsymbol{X}}_{\mathrm{ML}}$ and $\hat{\boldsymbol{a}}_{\mathrm{ML}}$ are the maximum likelihood state and amplitudes, $\boldsymbol{y}$ is the received signal, and D is the matrix of the signal expected to be received from each visible satellite at the state $\boldsymbol{X}$ with amplitude $\boldsymbol{a}$. Using the orthogonality property between different GPS Pseudo Random Noise (PRN) codes, this simplifies to finding the maximum value of the cross-correlation function $\mathcal{R}_t$ between the received signal and the superposition of the superposition of all expected satellite transmissions for the state $\boldsymbol{X}$:

$$\hat{\boldsymbol{X}}_{\mathrm{ML}} \approx \arg\max_{\boldsymbol{X}} \frac{1}{N} \boldsymbol{y}^* D D^* \boldsymbol{y} = \arg\max_{\boldsymbol{X}} \mathcal{R}(\boldsymbol{X}, \boldsymbol{t}) \tag{2}$$

Solving this objective function for $\boldsymbol{X}$ is the basis for single-receiver DPE (SR-DPE).

This effort will look at three new DPE based approaches: Multi-Receiver Direct Position Estimation (MR-DPE), GPU based Single Receiver Direct Position Estimation (SR-DPE) and a Sequential Multi-Receiver Direct Position Estimation (MR-DPE).

2. BACKGROUND AND PRIOR WORK IN DIRECT POSITION ESTIMATION

Existing works have identified the improved accuracy of DPE in degraded signal environments, using the Cramér-Rao lower bound [18] to prove the higher achievable accuracy of DPE when compared with the two-step approach. Software simulations under various propagation models have also indicated an improved accuracy performance of DPE in noisy signal environments [19–23]. These improvements have been corroborated through live-data experiments, including stationary ground stations [15, 24], a hand-held device near a residential structure [25] and receivers mounted on automobiles [26].

In addition to DPE, multi-receiver architectures have also been discussed [27–29] as means to improve GPS receiver accuracy by installing multiple receivers and their corresponding antennas on a single, rigid platform to increase effective signal power and geometric redundancy, as illustrated in Fig. 1.1. Existing works have experimented such architectures on small unmanned aircraft systems (sUAS) [27] and automobiles [28, 29], respectively using STL and VTL receivers as the constituent receivers of their networks.

However, the improved acquisition and tracking afforded by DPE comes at a cost. Solving an objective function to find the maximum likelihood state is computationally expensive [13]. This expense has been addressed by a variety of approaches ranging from more efficient navigation-domain searching to making educated approximations. Yet, the runtime expense of DPE remains a barrier to broader usage of the DPE algorithm.

3. METHODS, ASSUMPTIONS, AND PROCEDURES

This section breaks down the three methods for Multi-Receiver Direct Position Estimation (MR-DPE) and the GPU based Single Receiver Direct Position Estimation (SR-DPE).

3.1 Improved MR-DPE Fusion

In this work, we propose Multi-Receiver Direct Position Estimation (MR-DPE) to improve the measurement certainty of airborne GPS receivers in degraded signal environments by leveraging the benefits of both DPE and the multi-receiver architecture. MR-DPE uses multiple DPE receivers with known antenna baselines to form a receiver network. MR-DPE fuses the signal measurements from different receivers across the network (*i.e.* their likelihood functions with respect to the PVT domain) to generate a likelihood function for the network. An MLE for the navigation solution of the network then ensues.

3.1.1 Improved MR-DPE Fusion Motivation

In existing works that discussed multi-receiver networks [27, 29, 32-34], the SNR of each constituent receiver has not been quantitatively involved in the fusion process. In contrast, MR-DPE estimates the noise level in each constituent receiver using MLE and assigns weights to the measurements of the receiver accordingly. Therefore, a constituent receiver with decreasing SNR, which can be symptomatic of the presence of signal challenges, will have a reduced influence over the estimation of the navigation solution of the MR-DPE network.

The orientation (i.e. the attitude) of the network is essential for the fusion process as the antenna baselines are determined in the network coordinate frame (e.g. the body frame of the platform onto which the MR-DPE network is installed) rather than the global frame in which the GPS signals are measured. Multi-receiver architectures that have thus far been proposed do not consider the attitude of the platform as their operation was constrained to shorter antenna baselines [27, 29], or they have employed schemes that are ill-adapted for aerial environments [28]. Following the principle of DPE, we present a new, MLE-based algorithm that is capable of estimating the orientation of an aerial platform.

3.1.2 Formulation: State Estimation of Receivers with Respect to the Centroid

One of the key premises of MR-DPE is that the antenna baseline information is readily available to the network. For instance, a rigid platform, such as a fixed wing aircraft, would provide stationary antenna baselines that can be surveyed beforehand and known to the MR-DPE network. This assumption of known antenna baselines facilitates the coupling of the PVT coordinates of the constituent receivers via linear transformations, to wit:
where

- $\boldsymbol{X}_k \in \mathbb{R}^8$ is the PVT coordinate of the k-th constituent receiver.
- $\boldsymbol{X}_k \in \mathbb{R}^8$ is the PVT coordinate of a predetermined reference point O for the network (*e.g.* the network centroid).
- $\boldsymbol{X}_k \in \mathbb{R}^8$ is the antenna baseline of the k-th constituent receiver with respect to O, as defined in the local frame ("ℓ"-frame) of the network (*e.g.* the body frame of the platform on which the MR-DPE network is installed).
- $\boldsymbol{\phi} \in \mathbb{R}^6$ is the Euler angles (yaw, pitch, roll) and its first derivative (*i.e.* the angular rate) of the network with respect to the local tangent plane (*i.e.* the East-North-Up plane, ENU).
- $\vec{R} = \vec{R}(\boldsymbol{X}_o, \boldsymbol{\phi}) \in \mathbb{R}^{8\times3}$ is the baseline projection matrix that rotates the ℓ-frame such that its axes are aligned with the global Earth-Centered Earth-Fixed (ECEF) coordinate frame.

Then, applying this state model to the objective function of SR-DPE given in the introduction leads to the following formulation for MR-DPE:

$$\hat{\boldsymbol{X}}_{o,\mathrm{ML}}, \hat{\boldsymbol{\phi}}_{\mathrm{ML}} = \underset{\boldsymbol{X}_o, \boldsymbol{\phi}}{\arg\max} \left(\sum_{k=1}^{K} \frac{\boldsymbol{y}_k^* D_k \hat{\boldsymbol{a}}_{k,\mathrm{ML}}}{\hat{\sigma}_{k,\mathrm{ML}}^2} \right) \tag{4}$$

$$\hat{\sigma}_{k,\mathrm{ML}}^2 = \frac{\| \boldsymbol{y}_k - \hat{D}_{k,\mathrm{ML}} \hat{\boldsymbol{a}}_{k,\mathrm{ML}} \|^2}{N} \tag{5}$$

$$\hat{\boldsymbol{a}}_{k,\mathrm{ML}} = \hat{D}_{k,\mathrm{ML}}^{+} \boldsymbol{y}_k \tag{6}$$

where $\widehat{D}_{k,\mathrm{ML}}$ is the reconstructed signal for the k-th receiver at the maximum likelihood state $\widehat{\boldsymbol{X}}_{o,\mathrm{ML}}, \hat{\boldsymbol{\phi}}_{\mathrm{ML}}$ and $D^{+} = (D^{*}D)^{-1}D^{*}$ is the Moore-Penrose pseudoinverse matrix of D. These equations highlight the ability of MR-DPE to simultaneously optimize the PVT coordinate and the orientation of the network, a capability not found in prior works [27-29].

3.1.3 MR-PDE Algorithm

MR-DPE deploys k DPE receivers as its constituent receivers and organizes them into a receiver network. This network aggregates the measurements of the constituent receivers, *i.e.* their respective correlation manifolds $\mathcal{R}_k$, and derives a joint, network-level navigation solution accordingly. Note that it is assumed in this work, without loss of generality, that the constituent receivers are driven by a common clock.

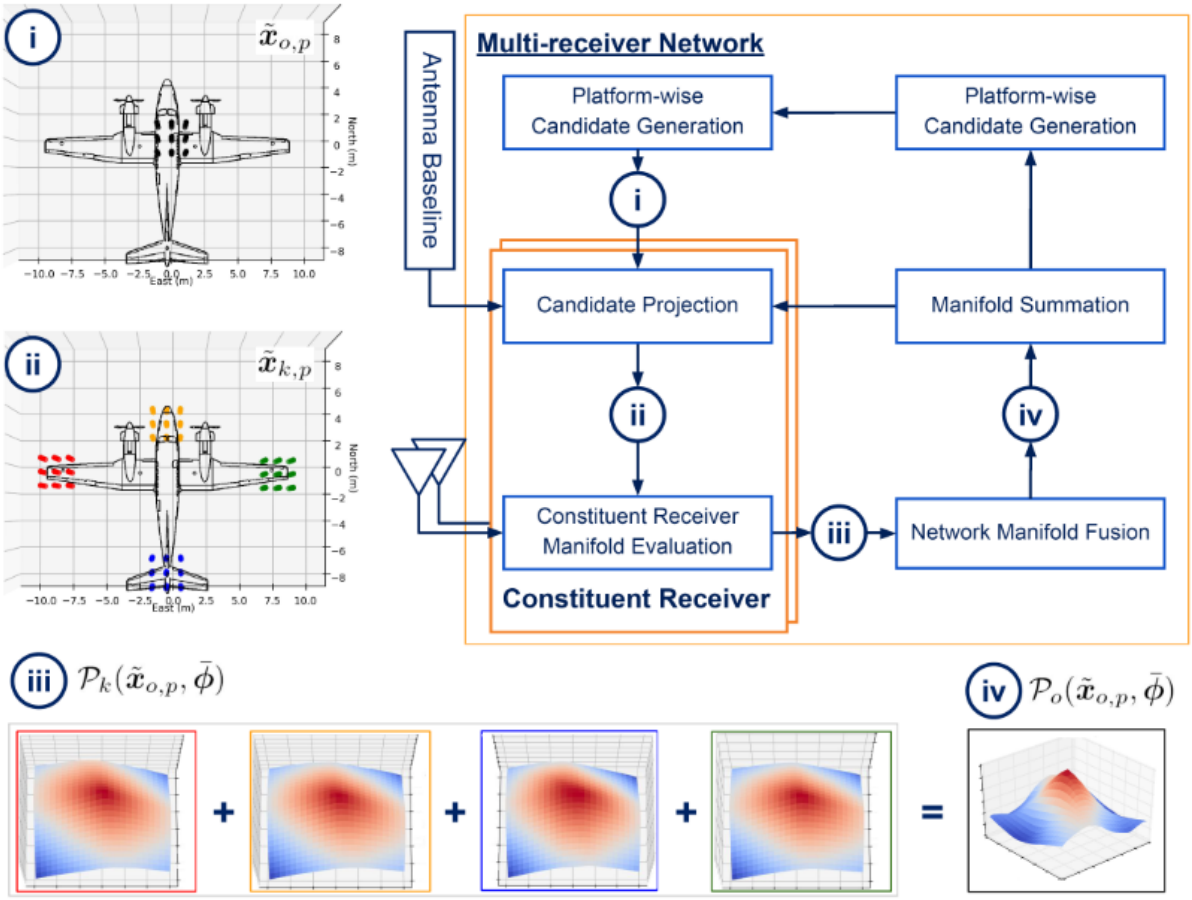


Figure 1. MR-DPE Algorithm Flow

The above figure depicts the iterative procedure of estimating the PVT coordinate of the MR-DPE network. For the purpose of clarity, the step-wise insets (i)-(iv) use an example where four color-coded antennas are positioned on the wing tips (left-red, right-green), the nose (orange) and the tail (blue) of a fixed-wing aircraft, and the centroid O is defined at the center of the fuselage. The candidate grids depicted are only in the position domain (x, y, z) for a more intuitive visualization. An iterative algorithm is devised to estimate the PVT solution of the network, as shown above. Each iteration consists of the following steps:

1. An iteration begins with the population of the zero-th and the first-order candidate grids, grids, $\{\tilde{\boldsymbol{x}}_{o,p}\}$ and $\{\tilde{\boldsymbol{x}}_{o,v}\}$, where

- $\tilde{\boldsymbol{x}}_{o,p} = \begin{bmatrix} \tilde{x}_{o,p} & \tilde{y}_{o,p} & \tilde{z}_{o,p} & c\tilde{\delta t}_{o,p} \end{bmatrix}^T$ is the p-th candidate for the network position/clock bias.
- $\tilde{\dot{\boldsymbol{x}}}_{o,v} = \begin{bmatrix} \tilde{\dot{x}}_{o,v} & \tilde{\dot{y}}_{o,v} & \tilde{\dot{z}}_{o,v} & c\tilde{\dot{\delta t}}_{o,p} \end{bmatrix}^T$ is the v-th candidate for the network velocity/clock drift.

This is shown in step (i) of the architecture, in which the network candidates are depicted as black discs surrounding the center of the fuselage. Each candidate is unique and represents a potential solution in its corresponding subspace [14, 23, 24]. MR-DPE then seeks the candidates that maximize the objective function of DPE. This numerical approach is preferred over an analytic approach as the objective function of DPE lacks closed-form solutions [13–15].

2. By the relationship between the PVT coordinates of the constituent receivers, the candidate grids $\{\tilde{\boldsymbol{x}}_{o,p}\}$ and $\{\tilde{\dot{\boldsymbol{x}}}_{o,v}\}$ are projected to the mounting points of the antennas of the constituent receivers. This creates K candidate grids for both the zeroth-order and the first-order terms:

$$\begin{bmatrix} \tilde{\boldsymbol{x}}_{k,p} \\ \tilde{\dot{\boldsymbol{x}}}_{k,v} \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{x}}_{o,p} \\ \tilde{\dot{\boldsymbol{x}}}_{o,v} \end{bmatrix} + \overset{\leftrightarrow}{R}(\bar{\boldsymbol{X}}_o, \bar{\boldsymbol{\phi}})\boldsymbol{b} \tag{7}$$

Note that this equation does not imply dependence between p-th zeroth-order candidate and the v-th first-order candidate. This step is depicted in step (ii) of the architecture. The corresponding inset shows that the black candidate grid near the center of the fuselage is now replaced by the four color-coded grids, each surrounding one of the antenna mounting points.

3. Following the projection of the candidate grids, each constituent receiver evaluates two

$$\mathcal{P}_k(\tilde{\boldsymbol{x}}_{o,p}, \bar{\boldsymbol{\phi}}, \boldsymbol{t}) \triangleq \mathcal{R}_k\left(\begin{bmatrix} \tilde{\boldsymbol{x}}_{o,p} \\ \bar{\dot{\boldsymbol{x}}}_o \end{bmatrix}, \bar{\boldsymbol{\phi}}, \boldsymbol{t}\right) \tag{8}$$

and one for the velocity/clock drift candidates:

$$\mathcal{V}_k(\tilde{\dot{\boldsymbol{x}}}_{o,v}, \bar{\boldsymbol{\phi}}, \boldsymbol{t}) \triangleq \mathcal{R}_k\left(\begin{bmatrix} \bar{\boldsymbol{x}}_o \\ \tilde{\dot{\boldsymbol{x}}}_{o,v} \end{bmatrix}, \bar{\boldsymbol{\phi}}, \boldsymbol{t}\right) \tag{9}$$

where $\mathcal{R}_k$ is the cross-correlation manifold for receiver k. Step (iii) in the architecture provides visualization for the evaluation of these receiver-level correlation manifolds.

4. The MR-DPE network then aggregates the receiver-level manifolds, as shown in step (iv) of the architecture. The resulting manifolds are the network position/clock bias manifold manifold P_o and the network velocity/clock drift manifold $\mathcal{V}_o$:

$$\mathcal{P}_o(\tilde{\boldsymbol{x}}_{o,p}, \bar{\boldsymbol{\phi}}, \boldsymbol{t}) \triangleq \mathcal{R}_o\left(\begin{bmatrix}\tilde{\boldsymbol{x}}_{o,p}\\ \bar{\dot{\boldsymbol{x}}}_o\end{bmatrix}, \bar{\boldsymbol{\phi}}, \{\bar{\sigma}_k^2\}, \boldsymbol{t}\right) = \sum_{k=1}^{K} \frac{\mathcal{P}_k(\tilde{\boldsymbol{x}}_{o,p}, \bar{\boldsymbol{\phi}}, \boldsymbol{t})}{\bar{\sigma}_k^2} \tag{8}$$

$$\mathcal{V}_o(\tilde{\dot{\boldsymbol{x}}}_{o,v}, \bar{\boldsymbol{\phi}}, \boldsymbol{t}) \triangleq \mathcal{R}_o\left(\begin{bmatrix}\bar{\boldsymbol{x}}_o\\ \tilde{\dot{\boldsymbol{x}}}_{o,v}\end{bmatrix}, \bar{\boldsymbol{\phi}}, \{\bar{\sigma}_k^2\}, \boldsymbol{t}\right) = \sum_{k=1}^{K} \frac{\mathcal{V}_k(\tilde{\dot{\boldsymbol{x}}}_{o,v}, \bar{\boldsymbol{\phi}}, \boldsymbol{t})}{\bar{\sigma}_k^2} \tag{9}$$

where the noise level $\bar{\sigma}_k^2(\boldsymbol{t})$ in each constituent receiver is approximated as constant given any two consecutive signal snapshots, $\boldsymbol{y_k(t - \Delta T)}$ and $\boldsymbol{y_k(t)}$.

5. Lastly, the PVT solution, $\widehat{\boldsymbol{X}}_{o,MR}$, is determined using maximum-likelihood estimation namely,

$$\hat{\boldsymbol{X}}_{o,\mathrm{MR}} = \begin{bmatrix}\arg\max_{\tilde{\boldsymbol{x}}_{o,p}} \mathcal{P}_o(\tilde{\boldsymbol{x}}_{o,p}, \bar{\boldsymbol{\phi}}, \boldsymbol{t})\\ \arg\max_{\tilde{\dot{\boldsymbol{x}}}_{o,v}} \mathcal{V}_o(\tilde{\dot{\boldsymbol{x}}}_{o,v}, \bar{\boldsymbol{\phi}}, \boldsymbol{t})\end{bmatrix} \tag{10}$$

3.1.4. Expansion: Noise Weighting

Note that the noise k of each constituent receiver is integrated into the fusion of the correlation manifolds $\mathcal{P}_o$ and $\mathcal{V}_o$. That is, the higher the noise level is in one receiver, the less the receiver contributes to the estimation of the network PVT solution $\widehat{\boldsymbol{X}}_{o,MR}$. Therefore, the fusion process not only utilizes the information redundancy gained by the additional signal observation, but

intelligently considers the quality of the signal from each receiver before its measurements are fused into the network. This approach is particularly important for scenarios when signal challenges are limited to certain constituent receivers within the network. In this case, the network is able to reduce the influence of these affected receivers on its navigation solution.

3.1.5 Expansion: Attitude Estimation

It is worth noting that the algorithm presented in this section requires the presence of the attitude information $\boldsymbol{\phi}$ to complete the estimation for the PVT coordinate of the network. Therefore, during each complete iteration of MR-DPE execution, the PVT estimation algorithm and the attitude estimation algorithm (presented below) proceed sequentially to supply essential information to each other. Similar to the PVT estimation algorithm introduced for the formulation of the MR-DPE algorithm, the orientation estimation algorithm for MR-DPE executes iteratively using a grid of orientation candidates, each representing a unique combination of angles and angular velocities.

First, two three-dimensional orientation candidate grids, $\{\tilde{\boldsymbol{\alpha}}_u\}$ and $\{\tilde{\dot{\boldsymbol{\alpha}}}_w\}$, are populated, where

$$\{\tilde{\boldsymbol{\alpha}}_u\} = [\tilde{\alpha}_u \quad \tilde{\beta}_u \quad \tilde{\gamma}_u]^T \tag{11}$$

$$\{\tilde{\dot{\boldsymbol{\alpha}}}_w\} = [\tilde{\dot{\alpha}}_w \quad \tilde{\dot{\beta}}_w \quad \tilde{\dot{\gamma}}_w]^T \tag{12}$$

and (11) is the u-th attitude candidate and (12) is the w-th angular-rate candidate. These two grids are centered at an initializing value, which may be obtained from various sources, *e.g.* the prediction $\bar{\boldsymbol{\phi}}_o$ based on the previous iteration.

Next, two correlation manifolds, $\mathcal{U}$ and $\mathcal{W}$, are then evaluated in a fashion similar to the position and velocity manifolds:

$$\mathcal{U}(\tilde{\boldsymbol{\alpha}}_u, \boldsymbol{t}) \triangleq \mathcal{R}_o \left(\bar{\boldsymbol{X}}_o, \begin{bmatrix} \tilde{\boldsymbol{\alpha}}_u \\ \bar{\dot{\boldsymbol{\alpha}}} \end{bmatrix}, \{\bar{\sigma}_k^2\}, \boldsymbol{t} \right) \tag{13}$$

$$\mathcal{W}(\tilde{\dot{\boldsymbol{\alpha}}}_w, \boldsymbol{t}) \triangleq \mathcal{R}_o \left(\bar{\boldsymbol{X}}_o, \begin{bmatrix} \bar{\boldsymbol{\alpha}} \\ \tilde{\dot{\boldsymbol{\alpha}}}_w \end{bmatrix}, \{\bar{\sigma}_k^2\}, \boldsymbol{t} \right) \tag{14}$$

That is, instead of projecting a PVT candidate grid to the antenna mounting points with the same network orientation, a single PVT coordinate is projected with different orientation values, and the correlation values resulting from these different projections are subsequently assessed.

Lastly, the candidates that respectively yield the highest u and w values are selected as the estimate for the network orientation, namely:

$$\hat{\boldsymbol{\phi}}_{\mathrm{MR}} = \begin{bmatrix} \arg\max_{\tilde{\boldsymbol{\alpha}}_u} \mathcal{U}(\tilde{\boldsymbol{\alpha}}_u, \boldsymbol{t}) \\ \arg\max_{\tilde{\boldsymbol{\alpha}}_w} \mathcal{W}(\tilde{\boldsymbol{\alpha}}_w, \boldsymbol{t}) \end{bmatrix} \tag{15}$$

To better understand the functioning of the orientation estimation algorithm, the following figure provides a visualization for the orientation manifold using three orientation candidates with different yaw values $\tilde{\alpha}_u$. When the values provided to the orientation candidate diverge from the true orientation of the network, as in the cases on the left and on the right of the following figure, the position of the propagated candidate drifts off the true position of the antenna. The correlation u will thus decrease accordingly.

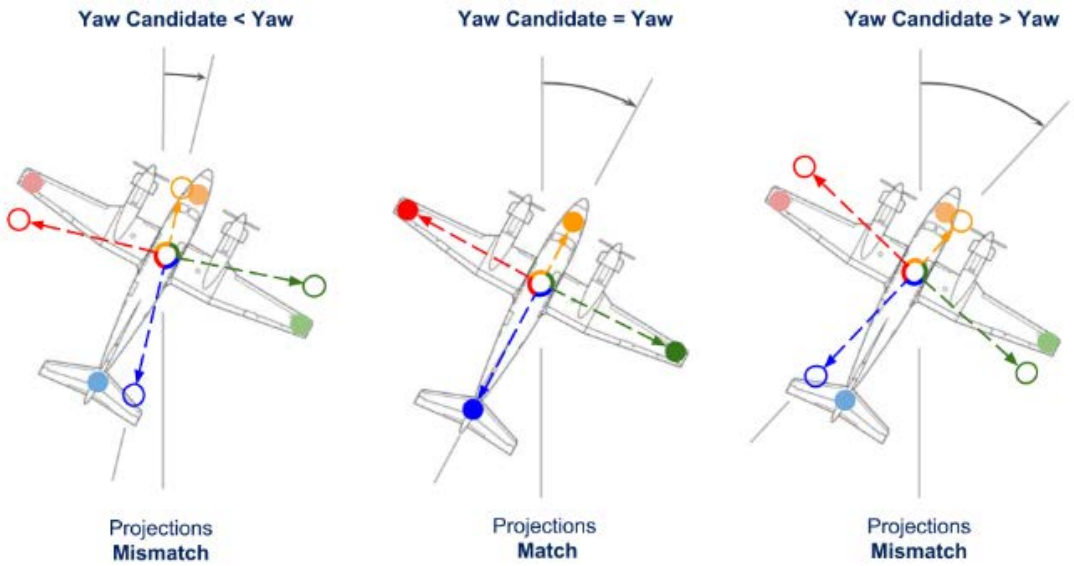
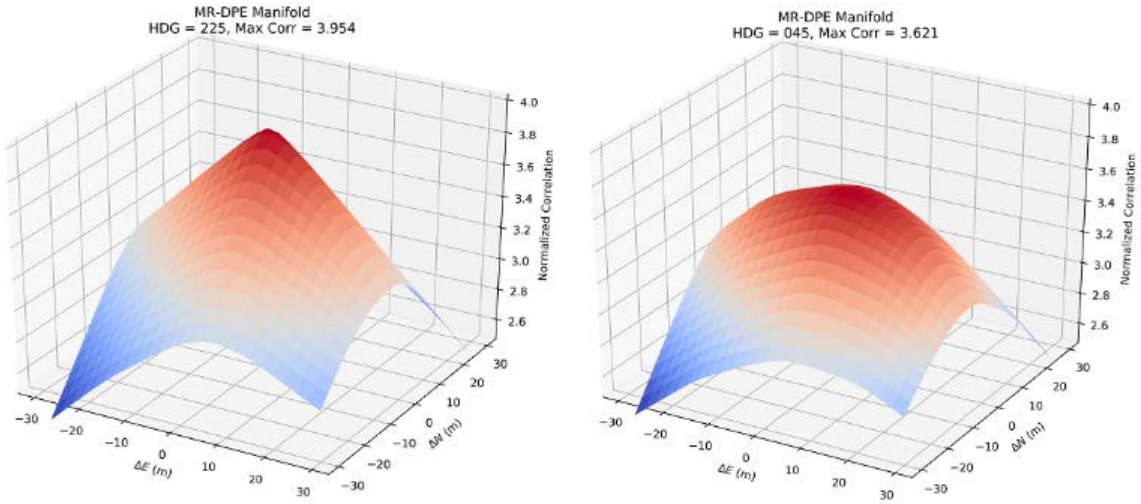


Figure 2. Candidate Yaw versus Yaw and Impact on Projections.



(a) **Matched** yaw candidate (b) **Mismatched** yaw candidate

Figure 3. Matched and Unmatched Yaw Candidate and Manifold Picture.

The manifold pictured above is constructed from real-world data. The heading of the network was at 225°, and the figure on the left depicts the case when the correct orientation value is applied when computing $\mathcal{U}$. Higher correlation values along with a sharper manifold peak are observed when compared with the figure on the right, where the orientation candidate is set with the heading of 45°. This results in much lower correlation values and a rounded appearance of the correlation manifold.

3.1.6 Theoretical Benefits: Cramér-Rao Bound Analysis

The following statements can be made both on the theoretical benefits from a Cramér-Rao Bound Analysis standpoint. Recall that for any distribution of the exponential family (e.g. the Gaussian distribution), the maximum-likelihood estimator and the minimum-variance unbiased estimator (MVUE) are identical for a given distribution parameter $\boldsymbol{\theta}$. The estimators of the MR-DPE formulation are therefore MVUE estimators to the parameter set $\boldsymbol{\theta}_o = \{\{\boldsymbol{a}_k\}_{k=1}^K, \boldsymbol{X}_o, \boldsymbol{\phi}, \{\sigma_k^2\}_{k=1}^K\}$, and their covariance is bounded by the Cramér-Rao lower bound:

$$\mathrm{Cov}[\hat{\boldsymbol{\theta}}_o | \boldsymbol{\theta}_o] \preceq I_{\boldsymbol{\theta}_o}^{-1} \tag{16}$$

where $I_{\boldsymbol{\theta}_o} \triangleq \mathbb{E}\left[-\nabla^2_{\boldsymbol{\theta}_o} \log p(\{\boldsymbol{y}_k\}_{k=1}^K | \boldsymbol{\theta}_o)\right]$ is the Fisher Information matrix of $\boldsymbol{\theta}_o$. Under the assumption of independent Gaussian processes for the noise in each constituent receiver, the conditional distribution of the received signal snapshot $\boldsymbol{y}_k$ at the k-th receiver for all receivers in K is:

$$p(\boldsymbol{y}_k | \boldsymbol{a}_k, \boldsymbol{X}_o, \boldsymbol{\phi}, \sigma_k^2) = \left(\frac{1}{\sqrt{2\pi\sigma_k^2}}\right)^N \exp\left\{-\frac{\|\boldsymbol{y}_k - D_k \boldsymbol{a}_k\|^2}{2\sigma_k^2}\right\} \tag{17}$$

which leads to the conclusion that

$$I_{\boldsymbol{\theta}_o} = \mathbb{E}[-\nabla^2_{\boldsymbol{\theta}_o} \log \prod_{k=1}^{K} p\left(\boldsymbol{y}_k | \boldsymbol{\theta}_k\right)] = \sum_{k=1}^{K} \mathbb{E}[-\nabla^2_{\boldsymbol{\theta}_o} \log p\left(\boldsymbol{y}_k | \boldsymbol{\theta}_k\right)] = \sum_{k=1}^{K} I_{\boldsymbol{\theta}_o, k} \tag{18}$$

where $\boldsymbol{\theta}_k = \{\boldsymbol{a}_k, \boldsymbol{X}_o, \boldsymbol{\phi}, \sigma_k^2\}$. That is, an increasing number of constituent receivers in the MR-DPE network will result in a corresponding lowering of the Cramér-Rao bound and, therefore, an improvement on the attainable accuracy.

3.2 Efficient GPU-Based DPE Implementation

Since each candidate PVT solution may be evaluated independently, the DPE algorithm is highly parallelizable. This work aims to demonstrate a real-time DPE-based receiver by exploiting its inherent parallelism. In this work, a software defined DPE-based GPS receiver is implemented using the CUDA C/C++ programming language. The DPE algorithm is parameterized in a manner that allows the algorithm's run speed to be tuned by a user for the hardware on which it runs. Additionally, to support future research, the receiver implementation is decomposed into seven subtasks and packaged as modules with well-defined inputs and outputs. This allows the researcher to abstract away parts of the DPE algorithm when modifying the receiver for a specific experiment. Implementation details specific to an NVIDIA Jetson TX2 GPU are provided to showcase the architecture being tuned for a dataset and hardware. The proposed architecture is evaluated for GPU usage efficiency through kernel profiling.

3.2.1 Motivation

To implement a DPE algorithm as a software-defined GPS receiver, the receiver must solve the SR-DPE objective function. Since there is no closed-form solution to this equation, the objective function must be solved numerically rather than analytically. This may be accomplished by constructing a grid of candidate receiver states and evaluating the cross-correlation between the received signal and reconstructed signal for that grid point's state [17]. The cross-correlations are

the likelihood for each grid point, meaning the grid is the discrete representation of the probabilistic manifold of PVT estimates for DPE.

Under this grid-based numerical approach, principles of parallel computing can be used to improve the efficiency of the DPE algorithm. The scores of each candidate point can be computed independently and the maximum likelihood estimate from the grid can be accumulated additively as each point is processed. This makes DPE well-suited for implementation on a GPU, as each candidate state can be assessed on separate processing threads -- each thread executing the same instructions, but operating on different memory locations corresponding to different candidate points.

3.2.2 Formulation

To provide the researcher with the benefits of abstraction when developing a receiver implementation, the software segment of a software-defined GPS receiver may be decomposed into the following four tasks:

- Acquire Samples: load samples for the current processing iteration
- Process Samples: generate a measurement from the samples
- Filter Measurement: generate PVT estimate from the measurement
- Prepare for Next Iteration: advance parameters

These four tasks would be executed sequentially for a set of samples; the processing of that set constitutes one iteration.

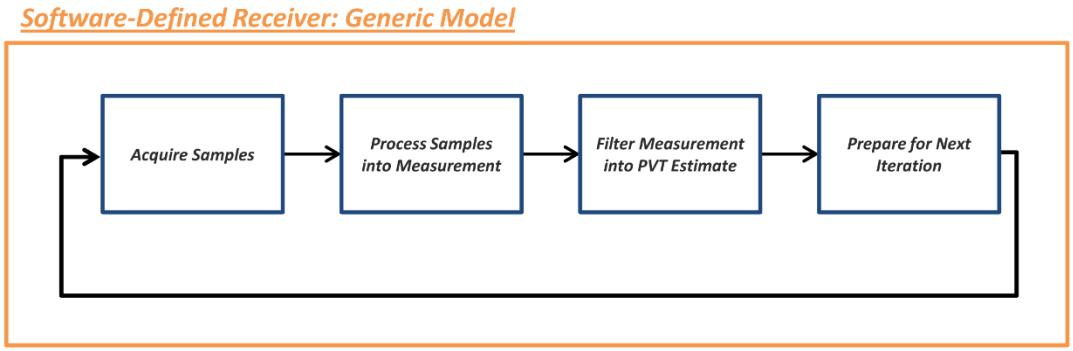


Figure 4. Generic Software-Defined Receiver Model and Flowchart

A DPE-based receiver may be implemented directly as follows from this architecture. However, this work proposes an expansion to seven tasks to improve abstraction for the researcher:

- The **Process Samples** task is where the DPE algorithm itself is implemented. This task is broken into two tandem tasks due to the complexity of the DPE algorithm and to support the objective of optimizing for GPU operation. This does not mean two steps like pseudorange measurements in scalar tracking, but is rather to allow the researcher the

flexibility through abstraction to make changes to the way in which the scores are batch-calculated and the way the manifolds are assessed. The first task will compute scores from the samples and the second will generate the measurement from the scores.

- The DPE algorithm must be initialized. Thus, a dedicated task is necessary to initialize the receiver algorithm, since initialization is performed differently than the manifold-based measurement generation.
- Result logging could be performed in the **Prepare for Next Iteration** task. However, due to the channel propagation performed at the end of an iteration of DPE, a dedicated task for data logging provides the researcher with abstraction between updating the channels and recording data.

3.2.3 Architecture

This leads to the seven-task decomposition implemented for this work:

- **Initialization:** place the grid and load channel parameters
- **Acquire Samples:** load samples for the current processing iteration
- **Batch Scoring:** compute the cross-correlation scores for the grid
- **Assess Manifold:** score the grid points and generate the measurement
- **Filter:** filter the measurements
- **Channel Propagation:** update channels and satellite states
- **Logging:** record results

Following this seven-task decomposition, the DPEFlow software-defined receiver program was created by developing the seven modules depicted below:

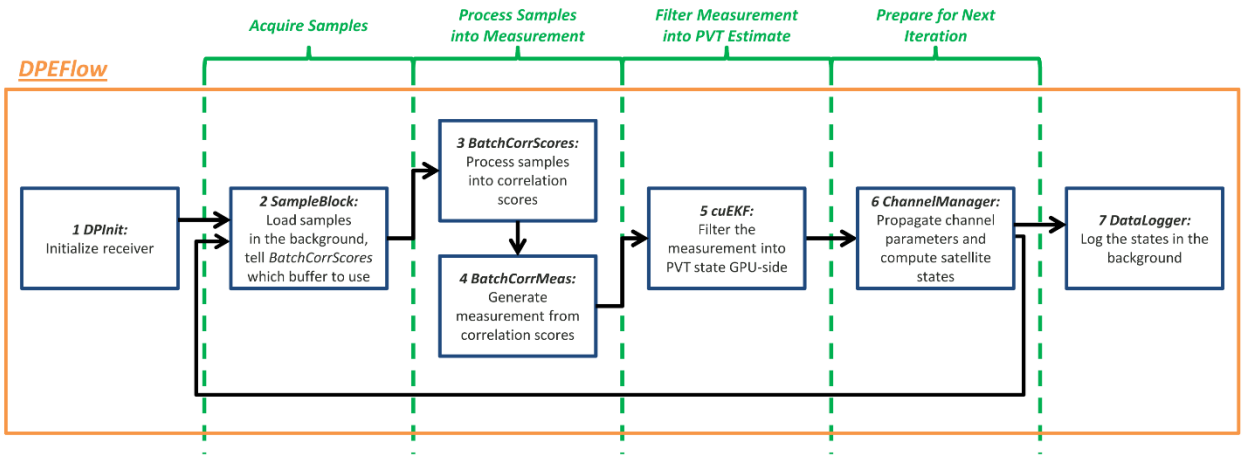


Figure 5. Generic Software-Defined Receiver Software Modules

3.2.4 Conceptual Benefits

The first of the two tandem SR-DPE algorithm modules developed for this work, the *BatchCorrScores* module computes the correlation scores for each point on the manifold grid by comparing its expected signal to the samples loaded by *SampleBlock*. This step could be implemented by constructing the expected signal for each candidate point and correlating each replica with the signal actually received. However, such an implementation is expensive, both in the amount of required operations and memory requirements. To reduce the computational complexity, two approximations have been implemented:

1. **Position and velocity manifold decoupling [35]:** This is due to the scale of the GPS with respect to the search space of an initialized receiver. For a fixed position state, the shape of the cross-correlation function with respect to velocity states will not be significantly affected by the exact position state chosen over a range of position states on the order of 100 meters [35]. Similarly, for a fixed velocity state, the shape of the cross-correlation function with respect to position states will not be significantly affected by the exact velocity state chosen over the range of reasonable receiver velocity changes [35].
2. **Batch correlation by FFT [36]:** If the PRN code chips are retrieved from the received signal, the circular cross-correlation property of the PRN codes of GPS may be leveraged to compute the cross-correlation score for all code phases [37]. This is accomplished by the circular cross-correlation property of the FFT.

To generate the position and velocity scores through batch correlation for a sample set, the parallelized implementation depicted on the following page was developed. This parallelization scheme ensures no computational work in the algorithm is redone and is parallelized through the use of three CUDA streams. Then, each point on the manifold can treat these batch-computed scores as a look-up table for different channel parameter offsets to determine the correlation score for that point.

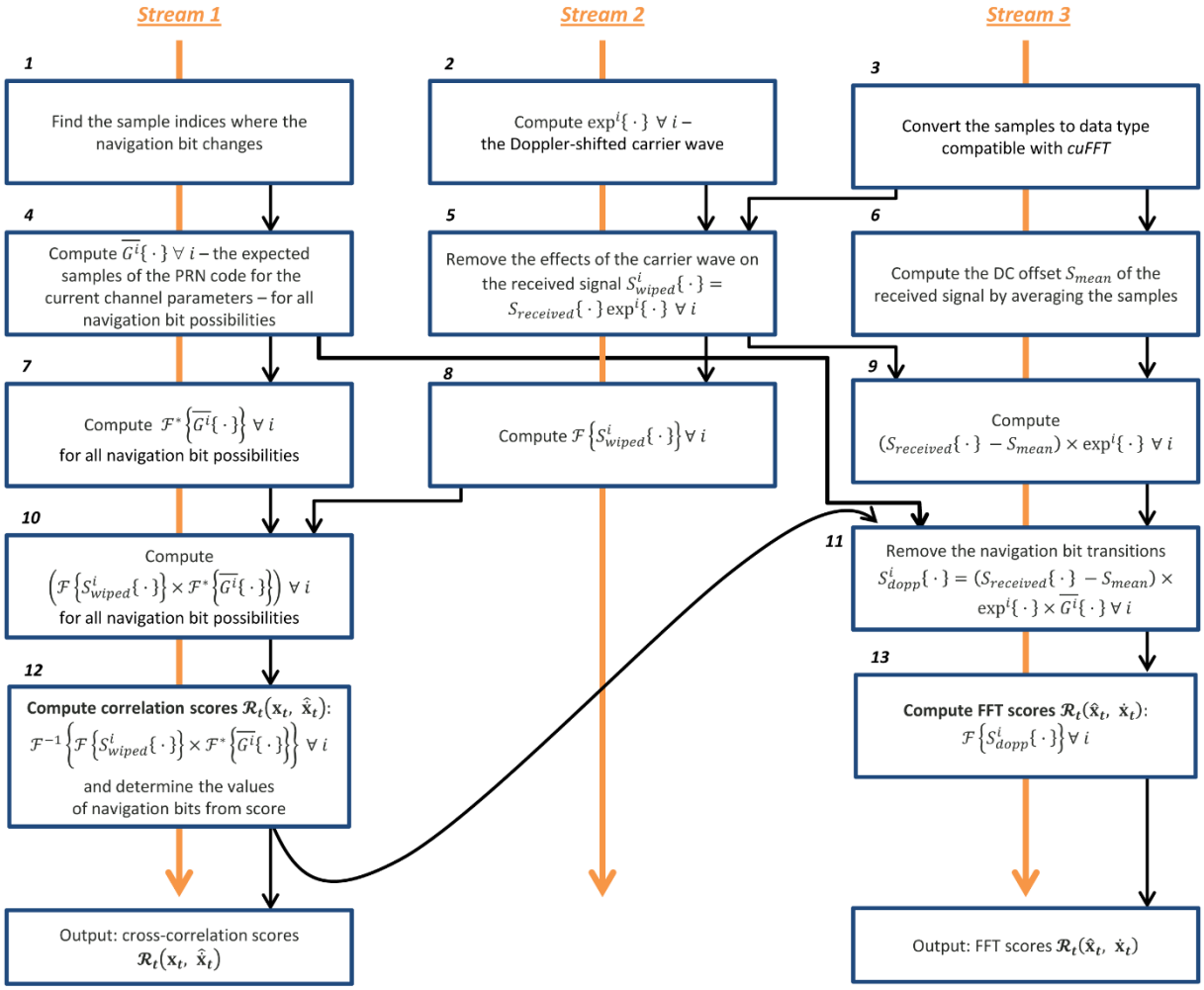


Figure 6. Parallelization Implementation of the Software

The cross-correlation functions generated by the *BatchCorrScores* module are then passed to the *BatchCorrManifold* module. The second of the two tandem DPE algorithm modules developed for this work, the *BatchCorrManifold* module determines which PVT states will comprise the manifold, computes the cross-correlation score for each point, then generates a measurement from the manifolds.

The position-time and velocity-drift manifolds are computed in parallel on separate CUDA streams. Each thread will compute the score for a state one at a time until all states have been scored. The states are chosen according to a grid initialization function and are oriented along the local ENU coordinates of the PVT state corresponding to the channel parameter estimates used when generating the batch correlation scores. To score a given state, the correlation score for each channel is found and added to a running sum for that point. Since both the position-time and velocity-drift manifolds are generated by looking up the batch correlation score for each state on

the grid, the thread-level algorithm for this underlying concept is represented in the following figure:

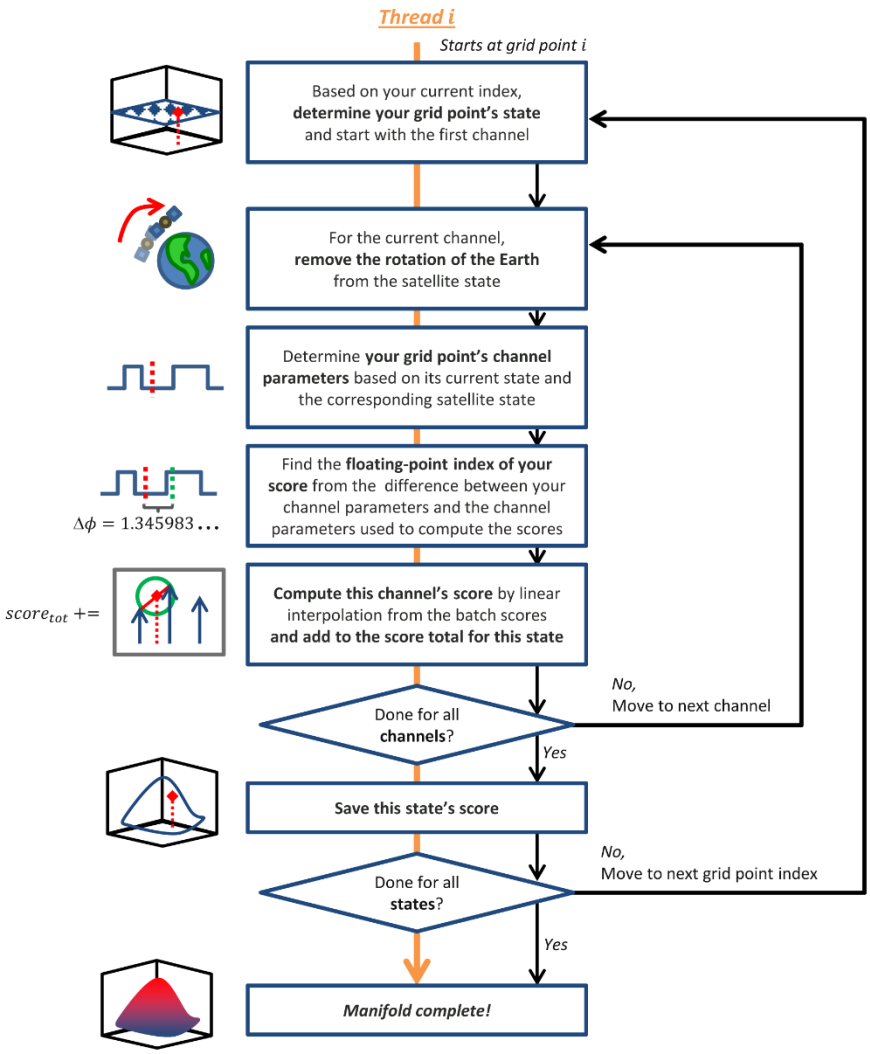


Figure 7. Maximum Likelihood

Once each point is scored, the maximum likelihood state may be found. This state is the measurement of the software-defined DPE receiver.

4. RESULTS AND DISCUSSION

Existing works have not explored the operation of DPE in aerial environments, which are characterized by a more dynamic motion profile and a more rigorous requirement for receiver performance [30]. In this work, full-scale flight tests were conducted on a fixed-wing aircraft [31] to evaluate the performance of MR-DPE in scenarios where signal multipath and masking were prevalent. For instance, in one of the test points, the aircraft flew in a river valley, below

the ridge lines, with the heights of the surrounding terrain exceeding the aircraft altitude by more than 500 meters.

This section breaks down the results for GPU based Single Receiver Direct Position Estimation (SR-DPE) and the Multi-Receiver Direct Position Estimation (MR-DPE).

4.1 Results and Discussion: GPU-DPE

The GPU used is an NVIDIA Jetson TX2. The TX2 consists of two streaming multiprocessors (SMs), each supporting up to a total of 2048 threads. CUDA kernels -- functions responsible for executing code on the GPU -- launch a specified number of blocks, each block executing a specified number of threads. Each block can support up to 1024 threads, though that number may need to be tuned smaller depending on how many registers a thread uses, as each block has a maximum of 32768 registers available to it. Thus, optimizing CUDA kernel launches for the TX2 entails choosing the number of blocks and threads to launch for each kernel so that the GPU has as many threads as possible active at a time while staying under the register cap for each kernel block launched.

The hardware segment consists of the pipeline shown below:

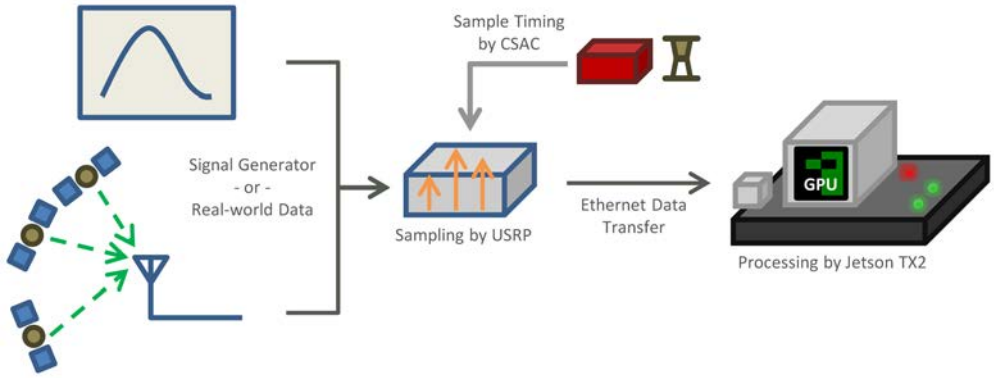


Figure 8. Hardware Segment

The sequential CPU implementation used as a benchmark is PyGNSS -- a Python-based software defined GNSS receiver developed by our research group [38]. PyGNSS performs the same computations as the GPU implementation developed for this work, but with vector operations rather than CUDA kernels and without parallelization present in *BatchCorrScores* or *BatchCorrManifold*, making it a suitable candidate for comparison to the GPU implementation. PyGNSS is run using Python 2.7 on an Asus G75VX commercially-available laptop with an Intel i7-3630QM 2.4GHz CPU and 12GB of RAM.

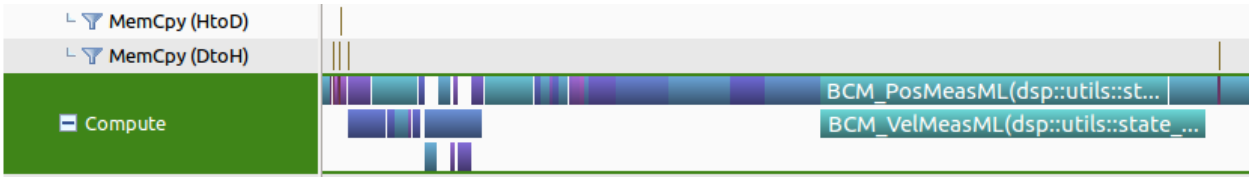
Table 1. GPU vs. CPI Processing Times

Implementation	Total Time	Time for One Iteration
CPU	2114.931s	705ms
GPU	647.555s	216ms

For this case, the parallelized GPU implementation ran approximately 3.26 times faster than the sequential CPU implementation. Furthermore, if a 20ms target is set for the real-time operation objective (due to the sample set length used for this work), the existing sequential CPU approach is approximately 35 times that value while the parallelized GPU approach improves to approximately 11 times the objective.

The module-level performance of the GPU-based SR-DPE receiver was also analyzed. The results presented are for the case of eight satellites being tracked in the open-sky stationary receiver simulated dataset. The first sixty seconds of the dataset are used to lock and track the eight satellites using scalar tracking and the second sixty seconds. The data set will processed by the DPE algorithm initialized by the channel parameters from the scalar tracking. The computational performance of the DPE algorithm is predominantly a function of the tunable parameters and less so the values of the samples, hence, cross-correlation scores need to be computed for each tracked channel, and the computation time will be dependent on the number of satellites.

The processing time taken by each GPU operation for one iteration of the DPE algorithm is shown in the figure and table below.



Module	Time	Dependent on Number of Channels
DPInit	Negligible	No
SampleBlock	Negligible	No
BatchCorrScores	113.26ms	Yes
BatchCorrManifold	100.94ms	Yes
cuEKF	1.04ms	No
ChannelManager	1.13ms	No
DataLogger	Negligible	No
Total	216.37ms	—

Figure 9. Breakdown of Timing

Other works have shown that a DPE receiver can maintain satellite track and accurate PVT estimates when duty-cycling sample-set reads [39], increasing the target time for real-time operation. The benefits of the speed-up from the GPU implementation can manifest in two ways when using this technique:

1. For the same duty-cycling interval, the GPU implementation may run manifolds with more points as compared to the sequential CPU implementation, improving the resolution of the receiver's measurements.
2. For the same number of points, the GPU implementation may run shorter duty-cycling intervals as compared to the CPU implementation, allowing the SR-DPE receiver to track higher-dynamic motion effects.

Additionally, the Jetson TX2 is an embedded GPU, and dedicated graphics cards such as the NVIDIA Titan V are significantly more powerful, and tuning the proposed architecture for such hardware is expected to further decrease processing time.

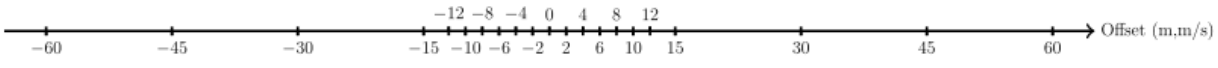
4.2 Simulated and Real World Testing

We have performed control environment and real world flight tests to verify our claim of improved position accuracy of MR-DPE compared to SR-DPE in GPS-degraded environments. We used the following candidate grid for both control environment and real-world flight tests. Control environment tests were performed to obtain the distribution of positioning errors under multiple initializations for different environments. The real-world flight tests were performed to test MR-DPE under real environments subject to actual physical signal effects, especially those prone to signal multipath and masking.

Table 2. Summary Real World Flight Tests

Domain	Axis	Span	Spacing	#/dim. [a]
Position	East, North, Up	Base grid [b] $\times 2$		21
Velocity	East, North, Up	Base grid $\div 10$		21
Time	Clock Bias δt	± 125 ns	$\frac{125}{3}$ ns	7
Time	Clock Drift $\dot{\delta t}$	2.5 ns/s	$\frac{5}{6}$ ns/s	7
Orientation	Attitude $\boldsymbol{\alpha}$	$\pm 15°$	$7.5°$	5
Orientation	Angular Rate $\dot{\boldsymbol{\alpha}}$	$\pm 15°/s$	$7.5°/s$	5

[a] Number of candidates per dimension
[b] The pattern of the base grid is depicted below



4.2.1 Simulated Datasets: Analysis of DPE Initialization Accuracy

We generated three synthetic multi-receiver GPS datasets emulating three different environmental conditions. We simulated open-sky, semi-urban and urban environments. For testing, we used one time sample data of 20 milliseconds and ran MR-DPE, SR-DPE algorithm

with multiple initial positions. These initial positions were randomly generated around ground truth. We performed 14,000 random initializations to obtain positioning error distribution. We obtained the positioning error distribution by repeating the measurement update step with different initial position, for same sample of the data. The details of each test are provided in the following sections.

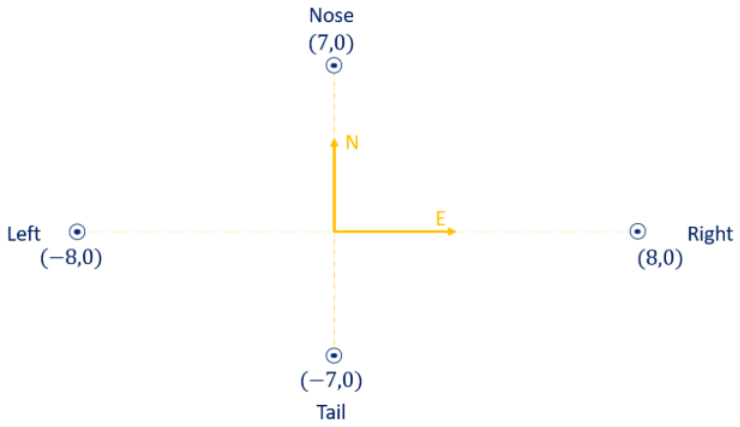


Figure 10. GPS Simulation Toolkit View

The NI GPS Simulation Toolkit with NI LabVIEW software is used to create synthetic GPS wave-forms for use with NI RF signal generators to produce customized and repeatable receiver tests. The toolkit provides the functionality of generating the signal received by an antenna subject to a given motion trajectory and adjustable satellite receive power levels.

The baseline of simulated multi-receiver configuration is depicted below. All four receivers were stationary. For simplicity, these receivers are referred as Left, Right, Nose and Tail receiver. We used all four receivers' location as known reference points, to generate synthetic data. Each receiver data was generated with a sampling frequency of f_s = 2.5 MHz and is 5 minutes long. Note that for testing, we used one time sample data of 20 milliseconds from the whole 5 minutes long data. In our simulation, based on the environment, Left and Right receiver's data was altered in the simulator by changing the received power of visible satellites. We compared SR-DPE solution of Left and Right receiver with MR-DPE solution.

4.2.2. Simulated Dataset 1: Open Sky

In this environment, all satellites were visible to all the receivers and no satellite measurements were altered. There were 8 satellites visible to all the receivers at all time. The horizontal, vertical and geometrical RMS error obtained for this test are tabulated in the conclusion to this section. The following figure shows the distribution of positioning error. It is observed that SR-DPE and MR-DPE have similar performance in open-sky environment with respect to magnitude of positioning error.

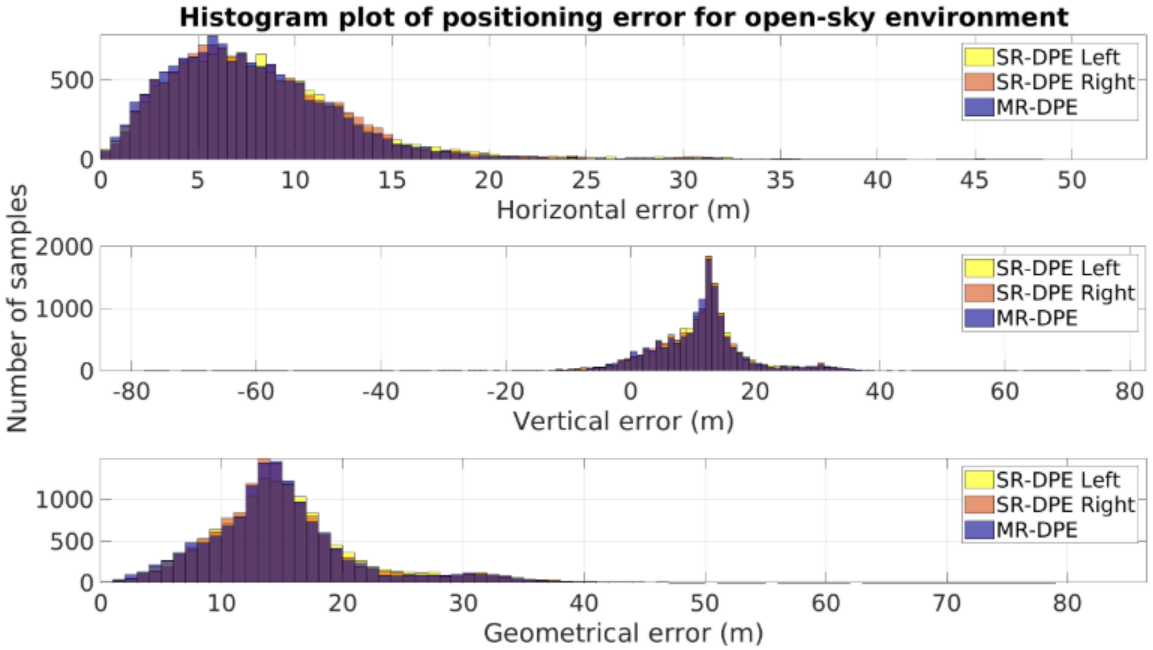


Figure 11. Simulation of Position Error Distribution for Open Sky Environment

4.2.3 Simulated Dataset 2: Semi-Urban Environment

Semi-Urban Environment: In this environment, we reduced power of satellites to simulate the effect of satellite power reduction due to buildings or foliage. For Left and Right receiver data, we reduced the power of four satellites by 15dB. The four satellites were selected based on their azimuth and elevation angle. The horizontal, vertical and geometric RMS error obtained for this test are tabulated at the conclusion of this section. The following figure shows the distribution of positioning error for this case. The spread of vertical and geometric error is smaller for MR-DPE when compared with SR-DPE. We observe more than 20% and 16% reduction in vertical and geometric error, respectively, for MR-DPE.

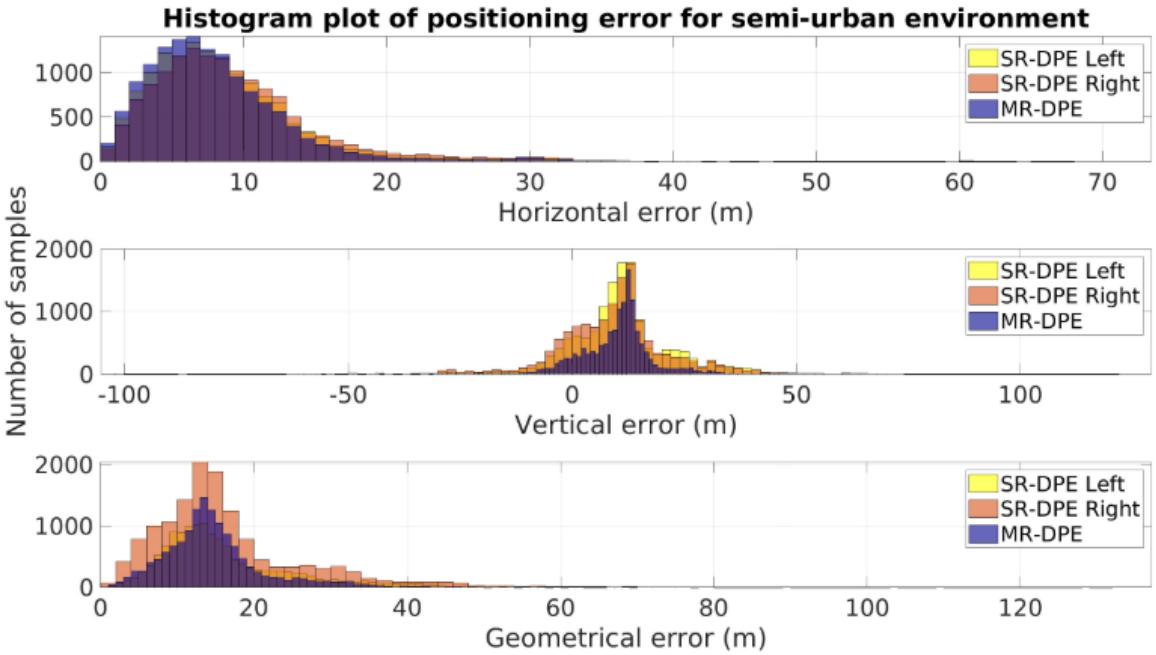


Figure 12. Simulation of Position Error Distribution in Semi-Urban Environment

4.2.4 Simulated Dataset 3: Urban Environment

In this environment, we simulated signal blockage due to urban structures. For Left and Right receiver data, four satellites were omitted entirely from the reconstructed signal. The four satellites were selected based on their azimuth and elevation angle. The horizontal, vertical and geometric RMS error obtained for this test are tabulated in the conclusion of this section. The following figure shows the distribution of positioning error for this case. The spread of vertical and geometric error is smaller for MR-DPE when compared with SR-DPE. We observe more than 27% and 24% reduction in vertical and geometric RMS error, respectively, for MR-DPE solution.

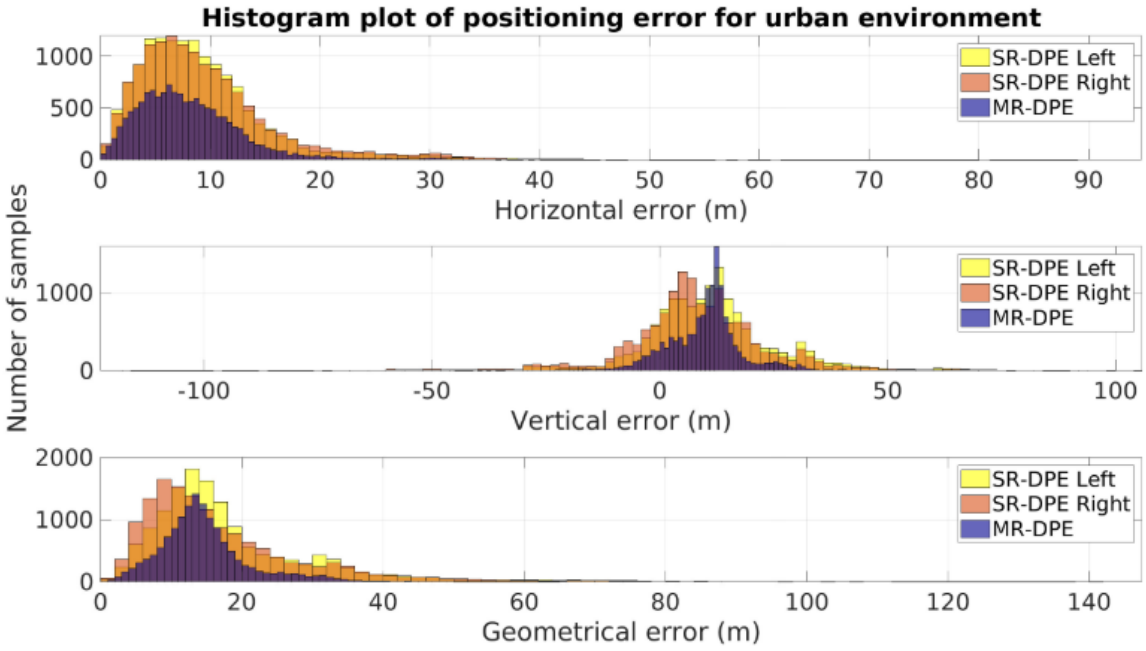


Figure 13. Simulation of Position Error Distribution in Urban Environment

4.2.5. Simulated Datasets: Conclusion

Overall, the RMS error of MR-DPE in all three environments remained similar, as shown in the following table. The RMS error of SR-DPE increases from open-sky environment to urban environment. This shows that MR-DPE is resilient to environmental changes.

Table 3. Overall GPS Simulation for all Environments

#	Test Point	Receiver	Horizontal (m)			Vertical (m)			Geometric (m)		
			SR	MR	Δ (%)	SR	MR	Δ (%)	SR	MR	Δ (%)
1	Open-Sky	Left	9.74	8.93	−8.39	13.58	13.49	−0.65	16.71	16.18	−3.21
		Right	9.48		−5.81	13.29		1.55	16.32		−0.87
2	Semi-Urban	Left	10.38	9.86	−4.98	16.47	12.81	−22.25	19.47	16.16	−16.98
		Right	11.08		−11.00	16.02		−20.07	19.48		−17.03
3	Urban	Left	10.99	9.53	−13.33	18.88	12.75	−32.35	21.84	15.92	−27.13
		Right	11.65		−18.24	17.48		−27.04	21.00		−24.22

4.3 Real-World Datasets: DPE Localization Under High-Dynamic Effects

We have performed flight tests that exposes airborne GPS receivers to signal challenges. Three flight profiles have been explored and analyzed in depth. Our experiment platform was a twin-engine, fixed-wing aircraft. Four GPS L1 active antennas were installed onto the aircraft, with one at each wingtip, one in front of the cockpit canopy and one on top of the vertical stabilizer, as shown in the following figure. Each antenna is connected to an Ettus Research Universal

Software-Radio Peripheral (USRP), a commercial off-the-shelf radio front-end, which records the raw RF samples.

Clockwise from top left: flight test aircraft, C-12C Huron; nose antenna (in front of cockpit canopy); tail antenna (on top of vertical stabilizer, not visible) and the measuring of antenna baselines using FARO® FaroArm portable coordinate measuring machine (PCMM); left-wing antenna.

Figure 14. Flight Test Configuration

The aircraft was equipped with a Time-Space Positioning Information (TSPI) system [31], [40], which deployed a kinematic, differential GPS receiver and a tactical- grade IMU [31], [41], [42] to achieve an accuracy of ±1.5 feet in position, ±0.02 feet-per-second in velocity, and 0.1° in attitude. The TSPI system therefore serves as the truth source for the position, velocity and orientation of the aircraft during the experiments. Figure 12 shows the four USRPs and the TSPI system mounted in the flight test aircraft.

Figure 15. Flight Hardware

4.3.1 Real-World Dataset 1: Tower Fly-By

The first test point (tower fly-by) was designed to replicate a signal environment that is commonly encountered by an aircraft in the stage of take-off and landing, namely, when the aircraft is traveling close to ground surface and with considerable artificial structures (e.g. hangars, towers, terminals) in its surroundings. During the experiment, the aircraft was initially positioned at 84 meters above ground level (AGL), 12 kilometers north- east to the runway, as depicted below. It then gradually descended toward the runway and reached a minimum height of 59 meters AGL, all the while traveling at approximately 108 meters per second. Upon crossing the east end of the runway, the aircraft initiated a climb-out, with the climb rate varying between 5 and 12.5 meters per second.

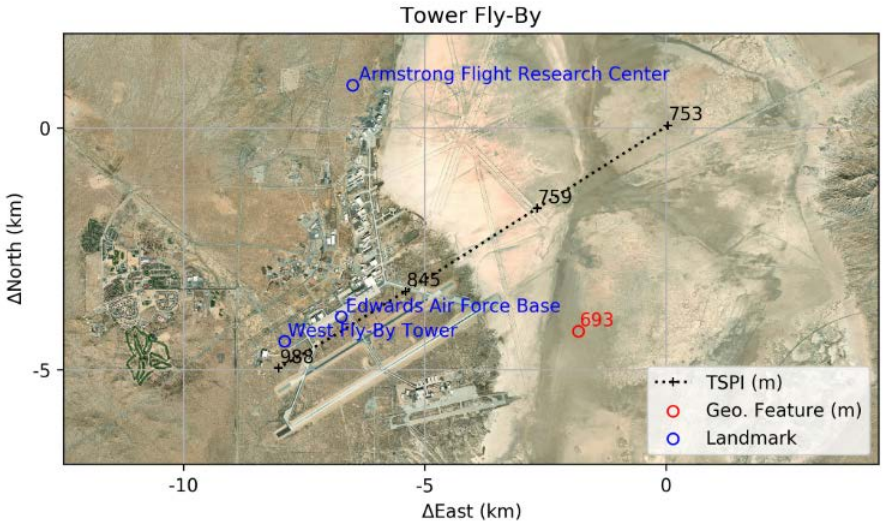


Figure 16. Tower Fly-by Route

The following figure depicts the error history of the test point; in addition, the true altitude recorded by the TSPI system is presented alongside the terrain elevation [43] to reflect the relationship between the flight path and the ground surface. The numerical results, tabulated at the conclusion of this section, indicate a 22.64% improvement over the horizontal accuracy, and a 41.91% improvement over the vertical accuracy.

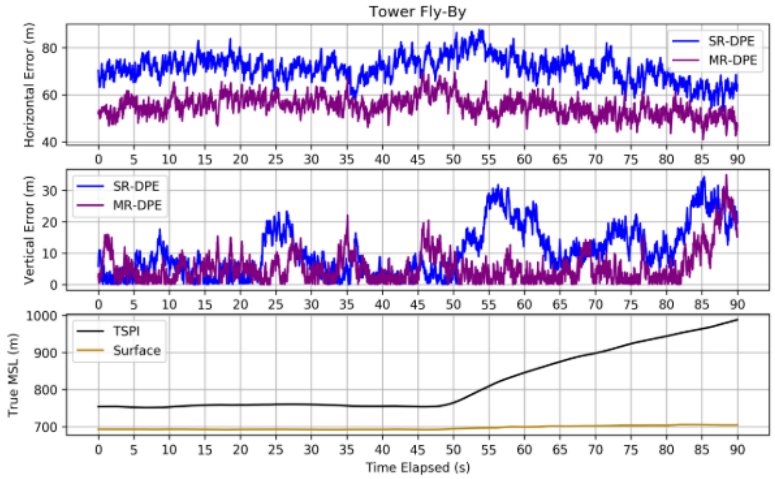
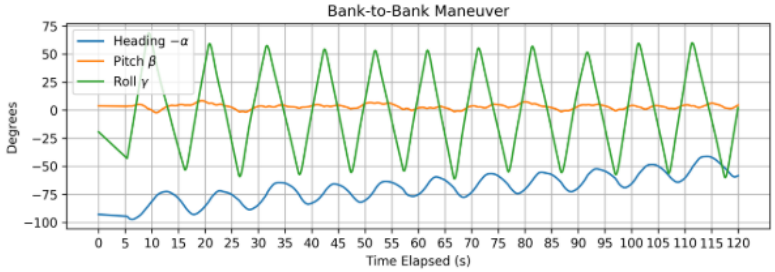


Figure 17. GPS Data: Tower Fly-By

4.3.2 Real-World Dataset 2: Bank-to-bank Maneuvers

In this test point, the aircraft was performing successive, bank-to-bank rolling maneuvers, while traveling at the ground speed of approximately 330 meters per second. The ailerons were deflected at three-fourths of their full deflection. As shown in the following figure, this resulted in the aircraft swiftly alternating between 60° banking to the left (−) and to the right (+). The average time to roll from one side to the other was 5.10 seconds, yielding an average rolling rate of 23.5°/s. Such dynamic maneuvers often result in the loss of track for receivers using the two-step approach due to the constantly changing satellite visibility [44]. To adjust for the high platform dynamics encountered during this test point, the candidate grid on the vertical axis is inflated to twice of the original spacing.



Attitude history of the bank-to-bank rolling maneuvers, as recorded by the TSPI system. Other than the swift back-and-forth of the roll angle, the heading of the aircraft exhibited similar oscillations.

Figure 18. Aircraft Data: Bank-to-Bank Maneuver

The time-domain error history for this test point is shown in the following figure. Statistically, the improvements on the horizontal and the vertical accuracy are respectively found to be 6.03% and 31.80%, as presented in the table at the conclusion of this section.

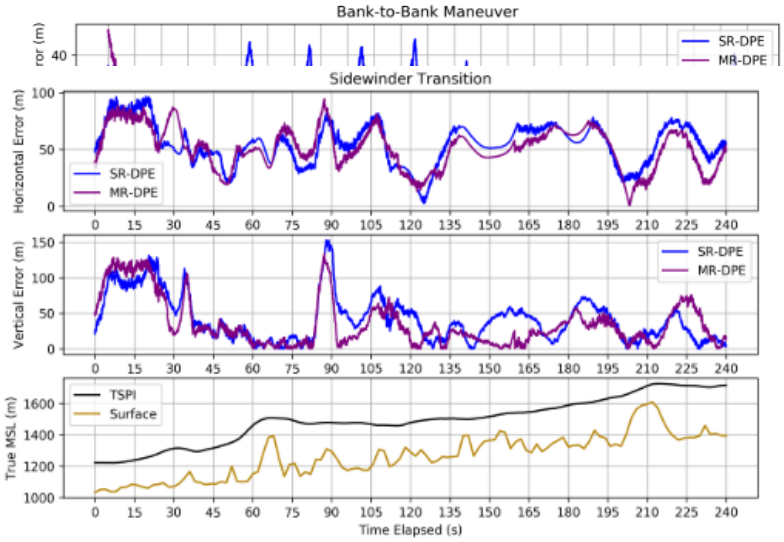


Figure 19. GPS Data: Bank-to-Bank Maneuver

4.3.3. Real-World Dataset 3: High-Terrain Environment

The third test point ("Sidewinder Transition" [45]) entailed some of the most challenging environments in which an airborne GPS receiver could be expected to operate. The aircraft was traveling in Kern River Valley, California, at less than 300 meters above ground level, while on both sides the elevation of the mountainous terrain exceeded the altitude of the aircraft by as much as 1.5 kilometers. To better illustrate the signal challenges experienced in this environment, the SNR history of the satellites is shown in below. Most notably, the SNRs of PRN 4, 14, 16 and 26 experienced a simultaneous decrease around t = 120.

An examination of the elevations and the azimuths of the four satellites at the time, as shown below, reveals that they were concentrated on the east side of the aircraft and their elevations were all below 45°. Further study to the area revealed a cluster of high-elevation geographical features to the east of the flight path during the same period of time, corroborating the presence of signal masking effects.

The improvement enabled by the multi-receiver network is observed on the temporal scale from the following figure. It is worth noting that under the heavy presence of signal masking effects, the vertical accuracy of this test point decreased considerably compared to the previous two test points. This behavior is consistent with the occlusion of the low-elevation satellites discussed earlier in this section. Nonetheless, improvements of 5.93% and 7.74% are respectively observed on the horizontal plane and on the vertical axis.

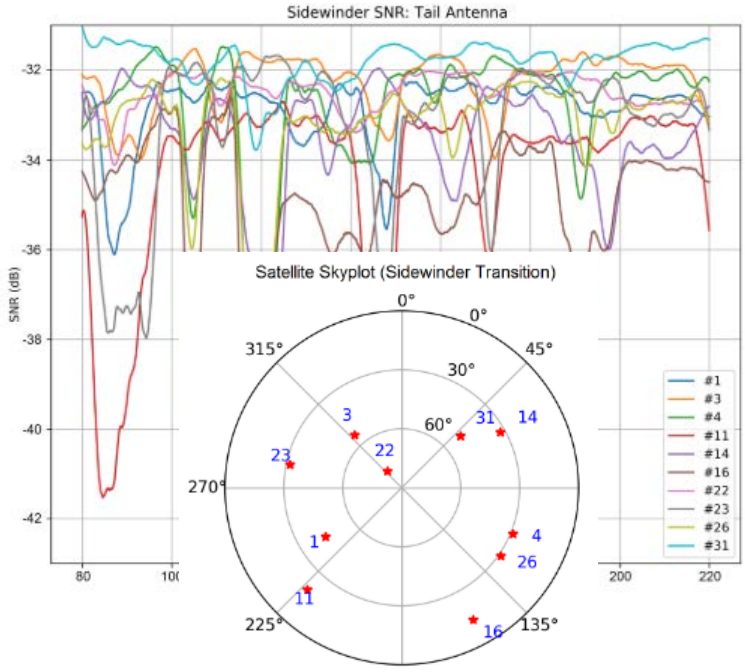


Figure 20. GPS SNR Data: High Terrain Environment

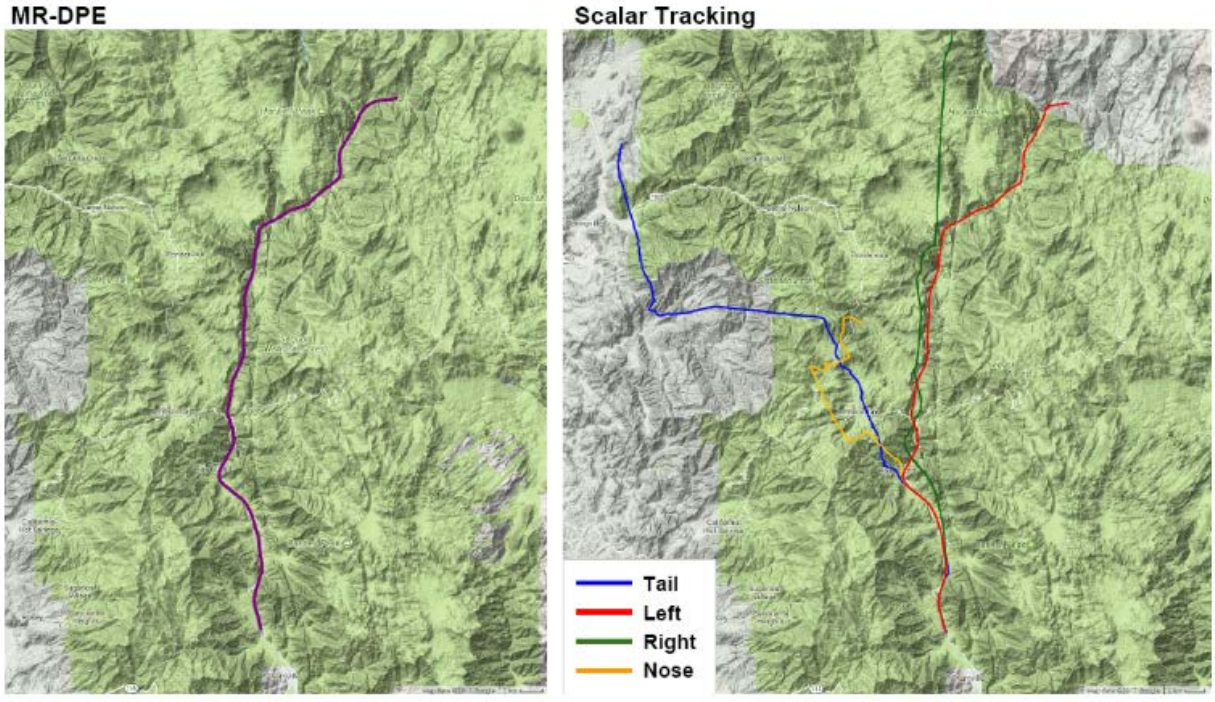


Figure 21. GPS Data: High Terrain Environment

Further, to emphasize the impact of the terrain on the received signal, the track of the MR-DPE algorithm was compared to that of a basic single-receiver scalar tracking implementation. Our MR-DPE implementation, shown on the left, maintained good tracking of the flight path throughout the test point. In contrast, three out of the four scalar tracking receivers, as shown on the right, lost track shortly after entering the high-terrain area, north of Kernville, CA.

4.3.4 Real-World Datasets: Conclusion

Overall, the RMS error of MR-DPE in all three test cases demonstrated an improvement over the RMS error of SR-DPE, as shown in the following table. In particular, the Tower Fly-By and the Bank-to-Bank Maneuver exhibited significant improvements over SR-DPE. As with the simulated data, this also shows that MR-DPE is resilient to environmental changes.

Table 4. Overall GPS Real Data for all Environments

#	Test Point	Length (s)	Horizontal (m)			Vertical (m)			Geometric (m)		
			SR	MR	Δ (%)	SR	MR	Δ (%)	SR	MR	Δ (%)
1	Tower Fly-By	90	71.33	55.19	−22.64	13.13	7.63	−41.91	72.53	55.71	−23.19
2	Bank-to-Bank Maneuver	120	16.69	15.68	−6.03	21.97	14.99	−31.80	27.59	21.69	−21.38
3	Sidewinder Transition	240	57.80	54.37	−5.93	52.50	48.44	−7.74	78.08	72.82	−6.74

5. CONCLUSIONS

To summarize, we have presented a novel Multi-Receiver Direct Position Estimation (MR-DPE) architecture for aerial GPS receivers and highlighted the advantages of MR-DPE in degraded signal environments. We have formulated MR-DPE as a maximum-likelihood (ML) estimation problem and accordingly identified the key novelties of MR-DPE, including the ML-based attitude estimation algorithm and the SNR-based weighting during the fusion of the measurements.

A seven-task decomposition of the SR-DPE algorithm was presented and implemented in seven software modules. Considerations for efficient GPU processing were also incorporated in the modules. The radio front-end acquires samples at precise intervals that are recorded to a data file. This data file is read and processed by the DPE implementation on the NVIDIA Jetson TX2 -- the computing hardware to which the implementation is tuned.

Further, we have developed an efficient GPU implementation of the SR-DPE algorithm that leverages the inherent parallelizability of the numerical solution method of the DPE objective function. This implementation is developed under a modular architecture tailored to provide abstraction in the steps of the DPE algorithm. This implementation is tuned and evaluated for

time efficiency on an embedded GPU. And, while this implementation was developed and evaluated as a SR-DPE receiver, it performs the same processing steps as one of the constituent receivers of the MR-DPE algorithm introduced in this work, setting up for expansion into an MR-DPE implementation for future work.

A sequential MR-DPE implementation was also developed to evaluate the performance of SR-DPE and MR-DPE in environments experiencing a variety of signal effects. Simulated datasets provided a control to study the accuracy of the initialization of the SR-DPE and MR-DPE algorithms under open-sky, semi-urban, and urban environments. Real-world environments provided operation scenarios of GPS-degraded environments and various flight profiles, especially those where multipath and terrain masking is prevalent, effects that are commonly encountered during critical flight phases. The localization results of this real-world dataset demonstrated the improvements provided by MR-DPE in challenging environments.

REFERENCES

[1] "Systems engineering and integration interface specification IS-GPS-200," 2015, [Online], available: https://www.gps.gov/technical/icwg/#is-gps-200.

[2] J. Beser and B. W. Parkinson, "The application of NAVSTAR differential GPS in the civilian community," *Navigation*, vol. 29, no. 2, pp. 107-136, June 1982.

[3] J. T. Eck, T. L. Bristol, and J. J. Hickey, "Nextgen priorities joint implementation rolling plan 2017-2019 executive report," September 2016, [Online], available: https://www.faa.gov/nextgen/media/NG Priorities Joint Implementation Plan.pdf.

[4] P. Ward, "The natural measurements of a GPS receiver," in *Proceedings of the 51st Annual Meeting of The Institute of Navigation*, Colorado Springs, CO, pp. 67-85, June 1995.

[5] P. Misra and P. Enge, *Global Positioning System*, Ganga-Jamuna Press, Lincoln, MA, 2006.

[6] J. J. Spilker, "Vector delay lock loop processing of radiolocation transmitter signals," U.S. Patent 5,398,034, March 14, 1995.

[7] G. X. Gao, M. Sgammini, M. Lu, and N. Kubo, "Protecting GNSS receivers from jamming and interference," *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1327-1338, June 2016.

[8] R. D. J. Van Nee, "Spread-spectrum code and carrier synchronization errors caused by multipath and interference," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 4, pp. 1359-1365, October 1993.

[9] R. L. Fante and J. J. Vaccaro, "Evaluation and reduction of multipath-induced bias on GPS time-of-arrival," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 3, pp. 911-920, July 2003.

[10] S. H. Kong, "Statistical analysis of urban GPS multipaths and pseudo-range measurement errors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 1101-1113, April 2011.

[11] J. F. McLellan, J. Schleppe, D. McLintock, and G. Deren, "GPS/barometry height-aided positioning system," in *Proceedings of 1994 IEEE Position, Location and Navigation Symposium - PLANS '94*, Las Vegas, NV, pp. 369-375, April 1994.

[12] J. McDonald and J. Kendrick, "Benefits of tightly coupled GPS/IRS for RNP operations in terrain challenged airports," in *2008 IEEE/ION Position, Location and Navigation Symposium*, Monterey, CA, pp. 294-303, April 2008.

[13] P. Closas, C. Fernández-Prades, and J. A. Fernández-Rubio, “Maximum likelihood estimation of position in GNSS,” *IEEE Signal Processing Letters*, vol. 14, no. 5, pp. 359-362, May 2007.

[14] S. Bhamidipati and G. Gao, “Improved GPS positioning accuracy using direct position estimation under multipath effects," *IEEE Transactions on Aerospace and Electronic Systems*, 2018.

[15] P. Closas and A. Gusi-Amigo, “Direct position estimation of GNSS receivers: Analyzing main results, architectures, enhancements, and challenges,” *IEEE Signal Processing Magazine*, vol. 34, no. 5, pp. 72-84, September 2017.

[16] 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, pp. 91-103, January 2007.

[17] P. Axelrad, B. K. Bradley, J. Donna, M. Mitchell, and S. Mohiuddin, “Collective detection and direct positioning using multiple GNSS satellites,” *Navigation*, vol. 58, no. 4, pp. 305-321, December 2011.

[18] P. Closas, C. Fernandez-Prades, and J. A. Fernandez-Rubio, “Cramér-Rao Bound analysis of positioning approaches in GNSS receivers,” *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 3775-3786, October 2009.

[19] O. Bialer, D. Raphaeli, and A. J. Weiss, “Maximum-likelihood direct position estimation in dense multipath,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2069-2079, June 2013.

[20] J. Liu, H. Yin, 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, pp. 3634-3643, September 2011.

[21] J. Liu, M. Lu, Z. Feng, and X. Cui, “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, August 2013.

[22] P. Closas, C. Fernández-Prades, J. Fernández-Rubio et al., “Evaluation of GNSS direct position estimation in realistic multipath channels,” in *Proceedings of the 28th International Technical Meeting of The Satellite Division of The Institute of Navigation (ION GNSS+ 2015)*, Tampa, FL, pp. 3693-3701, September 2015.

[23] Y. Ng and G. X. Gao, “Mitigating jamming and meaconing attacks using direct GPS positioning,” in *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Savannah, GA, April 2016.

[24] J. J. Brewer, "The differential vector phase-locked loop for Global Navigation Satellite System signal tracking," Ph.D. dissertation, Air Force Institute of Technology, Wright-Patterson AFB, OH, June 2014.

[25] T. Lin, J. T. Curran, C. O'Driscoll, and G. Lachapelle, "Implementation of a navigation domain GNSS signal tracking loop," in *Proceedings of the 24th International Technical Meeting of The Satellite Division of The Institute of Navigation (ION GNSS+ 2011)*, Portland, OR, pp. 3644-3651, September 2011, .

[26] Y. Ng, "Improving the robustness of GPS direct position estimation," M.S. thesis, University of Illinois at Urbana-Champaign, Urbana, IL, December 2016. [Online], available: http://hdl.handle.net/2142/95250

[27] A. Shetty and G. X. Gao, "Measurement level integration of multiple low-cost GPS receivers for UAVs," in *Proceedings of the 2015 International Technical Meeting of The Institute of Navigation*, Dana Point, CA, pp. 26-28. January 2015.

[28] Y. Ng and G. X. Gao, "Advanced multi-receiver vector tracking for positioning a land vehicle," in *Proceedings of the 28th International Technical Meeting of The Satellite Division of The Institute of Navigation (ION GNSS+ 2015)*, Tampa, FL, pp. 3148-3155, September 2015.

[29] Y. Ng and G. Gao, "GNSS multireceiver vector tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 5, pp. 2583-2593, October 2017.

[30] *Annex 10 to the Convention on International Civil Aviation*, 6th ed., International Civil Aviation Organization, Montreal, Québec, Canada, July 2006.

[31] A. G. Blackstock, J. DeMonte IV, P. J. Highland, M. S. Brodie, and J. P. Wilder, "Demonstration of aircraft state determination with blended solution of multiple GPS receivers: Project GRIFFIN," Edwards AFB, CA, June 2017.

[32] G. Lu, G. Lachapelle, M. E. Cannon, and B. Vogel, "Performance analysis of a shipborne gyrocompass with a multi-antenna GPS system," in *Proceedings of 1994 IEEE Position, Location and Navigation Symposium - PLANS '94*, Las Vegas, NV, pp. 337-343, April 1994.

[33] L. Heng, J. J. Makela, A. D. Dominguez-Garcia, R. B. Bobba, W. H. Sanders, and G. X. Gao, "Reliable gps-based timing for power systems: A multi-layered multi-receiver architecture," in *2014 Power and Energy Conference at Illinois (PECI)*, Champaign, IL, pp. 1-7, February 2014.

[34] S. Y. Lin and Z. Jiang, "GPS all in view time comparison using multi-receiver ensemble," in *2017 Joint Conference of the European Frequency and Time Forum and IEEE International Frequency Control Symposium (EFTF/IFC)*, Besançon, France: IEEE, July 2017.

[35] L. R. Weill, “A High Performance Code and Carrier Tracking Architecture for Ground-Based Mobile GNSS Receivers," pp. 3054-3068, 2010.

[36] P. Axelrad, J. Donna, and M. Mitchell, “Enhancing GNSS acquisition by combining signals from multiple channels and satellites," *22nd International Technical Meeting of the Satellite Division of the Institute of Navigation 2009, ION GNSS 2009*, vol. 5, pp. 3117-3128, 2009.

[37] K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen, *A Software- Defined GPS and Galileo Receiver: A Single-Frequency Approach*, Birkhüser Boston, Cambridge, MA, 2006.

[38] E. Wycoff and G. X. Gao, “A Python software platform for cooperatively tracking multiple GPS receivers,” in *Proceedings of the 27th International Technical Meeting of The Satellite Division of The Institute of Navigation (ION GNSS+ 2014)*, Tampa, FL, pp. 1417-1425, September 2014.

[39] Y. Ng and G. X. Gao, “Computationally efficient direct position estimation via low duty-cycling,” in *Proceedings of the 29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016)*, Portland, OR, pp. 86-91, September 2016.

[40] J. Shockley and D. Ruff, “TSPI vs. reference: What’s the difference?” in *2008 U.S. Air Force T&E Days*, Los Angeles, CA, American Institute of Aeronautics and Astronautics, February 2008.

[41] M. Veth, R. C. Anderson, F. Webber, and M. Nielsen, “Tightly-coupled INS, GPS, and imaging sensors for precision geolocation,” Wright- Patterson AFB, OH, January 2008.

[42] Honeywell International Inc., HG1700 *Inertial Measurement Unit*, Phoenix, AZ, 2016.

[43] Google Maps Elevation API, [Online], available: https://developers.google.com/maps/documentation/elevation.

[44] A. H.-P. Chu and G. X. Gao, “Multi-receiver direct position estimation tested on a full-scale fixed-wing aircraft,” in Proceedings of the 30th International Technical Meeting of The Satellite Division of The Institute of Navigation (ION GNSS+ 2017), Portland, OR, pp. 3761-3766, September 2017.

[45] “Edwards Air Force Base Instruction 13-100,” 412 OSS/OSA, October 2016.

DISTRIBUTION LIST

DTIC/OCP
8725 John J. Kingman Rd, Suite 0944 Ft
Belvoir, VA 22060-6218 1 cy

AFRL/RVIL
Kirtland AFB, NM 87117-5776 1 cy

Official Record Copy
AFRL/RVB/Dr. Lawrence M. Robertson 1 cy

This page is intentionally left blank.