A Low Complexity GNSS Array Signal Angle of Arrival (AoA) Estimation Algorithm and Validation

Boyi Wang[1,2], Yafeng Li[1,3] , Nagaraj Channarayapatna Shivaramaiah[1], and Dennis M. Akos[1]

[1]Dept. of Aerospace Engineering Sciences, University of Colorado at Boulder, Boulder, CO 80309, USA
[2]School of EIC, Huazhong University of Science and Technology, Wuhan, 430074, China
[3]School of Automation, Beijing Institute of Technology, Beijing, 100081, China

***Abstract*—Adaptive antenna array techniques are widely utilized to reject multipath and interference errors in Global Navigation Satellite System receivers (GNSS). Several methods have been proposed to tackle this issue. However, one severe problem for existing methods is the implementation complexity as GNSS receivers usually work under a high-dynamic receiving environment, thus may have the requirement of rapid angle estimates. In this paper, we propose a low complexity GNSS array signal estimation algorithm. Live data validation shows that the estimated AoA values can be used to improve the Carrier-to-Noise-Ratio by 1~4 dB-Hz.**

Keywords—AoA, GNSS, GPS, Matlab, SDR, Array Signal.

I. Introduction

With the development of Global Navigation Satellite Systems (GNSS), both high-accuracy and robust receivers are in high demand. To address the accuracy and reliability issue, sophisticated multipath rejection and high-sensitivity signal processing techniques are being incorporated into the receivers. The angle of arrival (AoA) estimation is a fundamental problem of GNSS antenna array signal processing in the same way that it is a critical task of radar, sonar and navigation systems.

Several non-adaptive AoA estimation methods have been proposed to tackle this issue. Delay-and-Sum [1] and Minimum Variance Distortion-less response Method (MVDR) [2] are two typical approaches. The basic idea behind the non-adaptive methods is to scan a beam through the signal reception space and measure the power received from each direction. Therefore the non-adaptive methods have low resolution as well as high implementation complexity. To improve the angle resolution of AoA estimation, AoA estimators that make use of the signal subspace have been proposed. Multiple Signal classification Algorithm (MUSIC) [3], root-MUSIC [4], Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) [5] are three representative algorithms. These subspace-based algorithms have high-resolution estimation capabilities. The basic idea of subspace-based algorithms is estimating the covariance of a signal plus noise model and then use it to form a matrix whose eigenstructure gives rise to the signal and noise subspaces. However, when considering the high-speed mobility of receiver and satellites for GNSS application, high complexity of subspace calculations and searches are not acceptable.

In this paper, we propose a low complexity GNSS array signal AoA estimation algorithm. The live data validation of the proposed algorithm shows that it can successfully estimate AoA of array signal. Rough analyses in terms of receiver complexity shows that the proposed algorithm has a lower implementation complexity compared to the subspace-based algorithm.

II. Antenna Array and GNSS Signal Model

A. Antenna Array Signal Model

In order to model an antenna array for GNSS signal, we make the following assumptions [6]:

1. We assume that several wavefronts are impinging on an antenna array with M isotropic sensor elements.
2. The transmission medium between the transmitter, receiver and possible scatterers is assumed linear, non-dispersive, and isotropic.
3. Radiation impinging on an array of sensor elements can be modeled as a superposition of wavefronts generated by each point source.
4. The point sources are located far from the array such that the direction of propagation is nearly equal at each sensor and the wavefronts are approximately planar (far-field assumption).

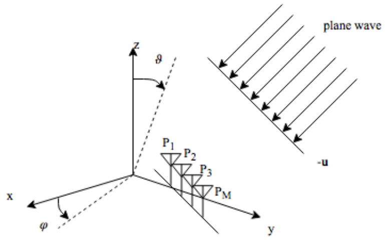


Fig. 1. Schematic of plane wave and antenna array.

As shown in Fig. 1, the array consists of isotropic sensors located at positions $\mathbf{p}_m, m = 1, \dots, M$ which spatially sample the signal filed. The array receives wave propagating in the direction **-u**. This yields a set of signals:

$$\mathbf{s}(t, \mathbf{p}) = \begin{bmatrix} S(t-\tau_1)e^{-j2\pi f_c \tau_1} \\ S(t-\tau_1)e^{-j2\pi f_c \tau_2} \\ \dots \\ S(t-\tau_1)e^{-j2\pi f_c \tau_M} \end{bmatrix}. \tag{1}$$

Submitted On: September 30, 2018
Accepted On: October 13, 2018

Here, τ_m denotes the time delays corresponding to the time of arrival at the various sensors at locations $\mathbf{p}_m$ where,

$$\tau_m = -\frac{\mathbf{u}^{\mathrm{T}}\mathbf{p}_m}{c}. \quad (2)$$

And c is the speed of the light. The vector $\mathbf{u}$ is a unit vector with:

$$\mathbf{u} = \begin{bmatrix} \sin(\vartheta)\cos(\varphi) \\ \sin(\vartheta)\sin(\varphi) \\ \cos(\varphi) \end{bmatrix} = \begin{bmatrix} \mathrm{u}_x \\ \mathrm{u}_y \\ \mathrm{u}_z \end{bmatrix}. \quad (3)$$

Where φ is the elevation angle, and ϑ is the azimuth angle. Thus, $\tau_m = -\frac{1}{c}\left(\mathrm{u}_x\mathrm{p}_{x,m}, \mathrm{u}_y\mathrm{p}_{y,m}, \mathrm{u}_z\mathrm{p}_{z,m}\right)$, with $\mathbf{p}_m = \begin{bmatrix} \mathrm{p}_{x,m} \\ \mathrm{p}_{y,m} \\ \mathrm{p}_{z,m} \end{bmatrix}$. For narrowband signals with $B\Delta T_{max} \ll 1$, where B is the signal bandwidth, $\Delta T_{max} \ll \tau_m$ is the maximum travel time between any two sensor elements in the array we can write:

$$s(t-\tau_m)e^{-j2\pi f_c\tau_m} \approx s(t)e^{-j2\pi f_c\tau_m}. \quad (4)$$

Thus, in the narrowband case the time delays τ_m can be approximated by only a phase shift and we can write:

$$\mathbf{s}(t,\boldsymbol{p}) = \begin{bmatrix} S(t-\tau_1)e^{-j2\pi f_c\tau_1} \\ S(t-\tau_1)e^{-j2\pi f_c\tau_2} \\ \dots \\ S(t-\tau_1)e^{-j2\pi f_c\tau_M} \end{bmatrix} = \boldsymbol{\alpha}(\varphi,\vartheta)s(\mathrm{t}). \quad (5)$$

Where $\boldsymbol{\alpha}(\varphi,\vartheta) \in \mathbb{C}^{M\times 1}$ is called steering vector with:

$$\boldsymbol{\alpha}(\varphi,\vartheta) = \begin{bmatrix} e^{\frac{j2\pi}{\lambda}}\left(U_xP_{x,1} + U_yP_{y,1} + U_zP_{z,1}\right) \\ e^{\frac{j2\pi}{\lambda}}\left(U_xP_{x,2} + U_yP_{y,2} + U_zP_{z,2}\right) \\ \dots \\ e^{\frac{j2\pi}{\lambda}}\left(U_xP_{x,M} + U_yP_{y,M} + U_zP_{z,M}\right) \end{bmatrix}. \quad (6)$$

Where λ is the wavelength.

B. GNSS Signal Model

The single channel GNSS RF signal can be modeled as:

$$\begin{aligned} S(t) &= \sqrt{2P}\left(s_I(t)\cos(2\pi f_c t) - s_Q(t)\sin(2\pi f_c t)\right) \\ &= \sqrt{2P}\Re\left(s(t)e^{j2\pi f_c t}\right). \end{aligned} \quad (7)$$

Where P is the power of the RF signal; $\Re(\cdot)$ represents real operator; f_c is the center frequency; S_I and S_Q are baseband I and Q signals that can be denoted as:

$$\begin{aligned} s_I(t) &= \textstyle\sum_{k=-\infty}^{+\infty} D_I(n)\mu_{T_c}(t-T_c) \\ s_Q(t) &= \textstyle\sum_{k=-\infty}^{+\infty} D_Q(n)\mu_{T_c}(t-T_c) \end{aligned}. \quad (8)$$

Fig. 2 shows the traditional GNSS receiver architecture.

The incoming RF signal, $S_{IF}(t)$, is multiplied and down converted to a baseband signal by locally the generated carrier signal. Then the baseband signal, $i(t)$ and $q(t)$ are correlated by locally generated early-, late-, and prompt-code signals. The correlated signals of prompt branches can be denoted as:

$$\begin{aligned} i_P(n) &= aD(n)R(\tau_P)\cos(\omega_e(n)t(n) + \theta_e) \\ q_P(n) &= aD(n)R(\tau_P)\sin(\omega_e(n)t(n) + \theta_e) \end{aligned}. \quad (9)$$

Where $D(n)$ is the navigation data bit; τ_p is the phase difference between locally generated prompt code and received PRN code; $R(\cdot)$ represents the normalized auto-correlation function (ACF) of PRN code; ω_e is the frequency difference between the generated carrier and the received carrier signal in rad; θ_e is the initial phase difference.

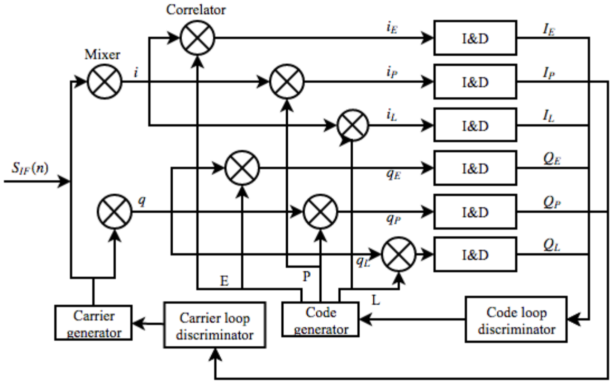


Fig. 2. Traditional GNSS receiver architecture.

The correlated signals of prompt branches are sent into integration and dumping (I&D) processor and can be denoted as:

$$\begin{aligned} I_P(n) &= aD(n)R(\tau_P)sinc(f_eT_{coh})\cos\phi_e \\ Q_P(n) &= aD(n)R(\tau_P)sinc(f_eT_{coh})\sin\phi_e \end{aligned}. \quad (10)$$

Where ϕ_e is the carrier phase difference; T_{coh} is the coherent integration time; f_e is the frequency difference in Hz. The integrator outputs of early and late branches can be denoted in a similar way. The only difference is value of the ACF of $R(\tau_E)$ and $R(\tau_L)$.

As it can be seen that the integrator outputs are determined by not only the code phase differences between local and received signals, but also by the carrier phase differences. In order to work independent from the carrier tracking loop, the code tracking loop usually uses the non-coherent discriminator to detect the correlation values between early and late branches. The traditional NELP discriminator can be denoted as:

$$V(n) = \frac{1}{2}\left(E(n) - L(n)\right). \quad (11)$$

Where,

$$\begin{aligned} E(n) &= \sqrt{I_E^2(n) + Q_E^2(n)} = aR(\tau_E)|sinc(f_eT_{coh})| \\ P(n) &= \sqrt{I_P^2(n) + Q_P^2(n)} = aR(\tau_P)|sinc(f_eT_{coh})| \\ L(n) &= \sqrt{I_L^2(n) + Q_L^2(n)} = aR(\tau_L)|sinc(f_eT_{coh})| \end{aligned}. \quad (12)$$

III. Proposed Architecture for GNSS Array Signal AOA Estimation and Beam-Steering

A. Post-Correlation AoA Estimation Architecture

According to Eq. (4), we ignore the time delay between elements. Then the GNSS array signal from one satellite can be denoted as:

$$S(t,\mathbf{p}) = \begin{bmatrix} \sqrt{2P}\left(s_I(t)\cos(2\pi f_c t) - s_Q(t)\sin(2\pi f_c t)\right)e^{-j2\pi f_c\tau_1} \\ \sqrt{2P}\left(s_I(t)\cos(2\pi f_c t) - s_Q(t)\sin(2\pi f_c t)\right)e^{-j2\pi f_c\tau_2} \\ \dots \\ \sqrt{2P}\left(s_I(t)\cos(2\pi f_c t) - s_Q(t)\sin(2\pi f_c t)\right)e^{-j2\pi f_c\tau_M} \end{bmatrix}. \quad (13)$$

Compared to the single element GNSS signal, the difference of GNSS array signal is that the received signal for every element has a corresponding phase shift $e^{-j2\pi f_c \tau_m}$. There are two approaches to estimate the corresponding phase shift angles between different elements: (1) pre-correlation estimation, and (2) post-correlation estimation. The pre-correlation estimation approach, as its name implies, is estimating AoA before the correlation operation. However, the pre-correlation GNSS signal is buried in the white noise environment, which is extremely weak for further signal processing. In this manner, it is hard to extract spatial information in weak GNSS array signal. Therefore, post-correlation processing is used to enhance the strength of the received GNSS array signal.

In Fig. 2 we can see that the correlators and I&D processors are used to enhance the signal strength. However, the carrier- and the code-phase difference between elements will be changed due to tracking loop adjustment. Therefore, to obtain correlated GNSS array signals, while maintaining the phase differences between elements, we propose the AoA estimation architecture in Fig. 3.

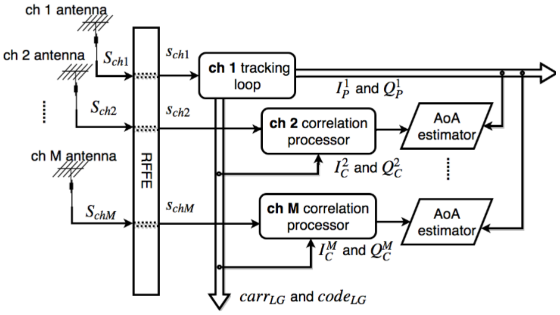


Fig. 3. Schematic diagram of the proposed AoA estimation architecture.

It can be seen that the GNSS array signals are received from antenna array for channel 1 to channel M. The received array signals are down-converted to base-band signals by the radio frequency front end (RFFE). Then the down-converted channel 1 signal is sent into a typical carrier- and code-tracking loop (described in Fig. 2) for correlation and integration processing. According to Eq. (10), the outputs of I&D for channel 1 can be denoted as:

$$\begin{aligned} I_P^1(n) &= aD(n)R(\tau)sinc(f_e T_{coh})\cos\phi_e \\ Q_P^1(n) &= aD(n)R(\tau)sinc(f_e T_{coh})\sin\phi_e \end{aligned}. \qquad (14)$$

The superscript of $I(n)$ and $Q(n)$ stands for the element numbers m, where $m = 1,2,\dots,M$.

Instead of being processed in the tracking loop, the rest of the $M-1$ channels (channel 2 to channel M) are sent into corresponding correlation processors respectively. The difference between a tracking loop and a correlation processor is: for a tracking loop the local carrier and code signals are generated based on the feedback loop adjustment. In this way, the tracking loop is able to maintain alignment of the received and generated signals. In order to adjust the loop adaptively, the early, late, and prompt branches are used.

The correlation processor, however, does not have the adaptive adjustment process. The input signals are correlated and integrated in the correlation processor. As can be seen from Fig. 3, two inputs of the $\mathrm{m_{th}}$ correlation processor are the locally generated carrier signal ($carr_{LG}$) and code signal ($code_{LG}$) from channel 1, while another input, s_{chm}, is the corresponding base-band signal down-converted from the $\mathrm{m_{th}}$ element. In this paper we assume that the tracking loop of channel 1 is in the lock condition. In this manner, both $carr_{LG}$ and $code_{LG}$ are aligned with the received $S_{ch1}(t)$. Therefore, the outputs of I&D for $m = 1,2,\dots,M$ can be denoted as:

$$\begin{aligned} I_C^m(n) &= aD(n)R(\tau)sinc(f_e T_{coh})\cos(\phi_e + 2\pi f_c \tau_m) \\ Q_C^m(n) &= aD(n)R(\tau)sinc(f_e T_{coh})\sin(\phi_e + 2\pi f_c \tau_m) \end{aligned}. \qquad (15)$$

In order to distinguish between the I&D outputs obtained from the prompt branch of channel 1 tracking loop, we use the subscript C for the I&D outputs obtained from correlation processor of the rest $M-1$ channels. As it can be seen, by correlating to $carr_{LG}$ and $code_{LG}$, the phase differences between different elements are remained. At the same time, the signal strength is improved significantly due to the correlator gain.

Hence, we can consider the correlated signals, χ_m, as an array signal that has the same spatial characteristics as the signals that arrive at the GNSS antenna array. Where χ_m can be denoted as:

$$\chi_m(n) = \begin{cases} I_P^m(n) + jQ_P^m(n), & \text{where } m = 1 \\ I_C^m(n) + jQ_C^m(n), & \text{where } m = 2,\dots,\mathrm{M} \end{cases}. \qquad (16)$$

From Eq. (16) we can see that the post-correlated array signal, χ_m, is composed of the real and imaginary parts of I&D outputs.

B. *Carrier Phase Difference Extraction Method*

Without loss of generality, we consider the carrier phase of χ_1 as the reference phase, i.e., $\tau_1 = 0$. Therefore, τ_m is the phase difference between $\mathrm{m_{th}}$ element (m ≥ 2) and channel 1. Then the relationship between χ_m and τ_m can be denoted as:

$$\chi_m(n) = aD(n)R(\tau)sinc(f_e T_{coh})e^{j(\phi_e + 2\pi f_c \tau_m)}. \qquad (17)$$

In order to extract τ_m, we propose the Carrier Phase Difference Extraction (CPDE) method. The implementation of CPDE can be expressed as:

$$\tau_m(n) = -\mathrm{angle}\{\chi_1 \times \chi_m^*\}. \qquad (18)$$

Where χ_m^* is the complex conjugate of χ_m. CPDE uses the angle information of the product between χ_1 and χ_m^* to extract τ_m. The detailed derivation of Eq. (18) can be written as:

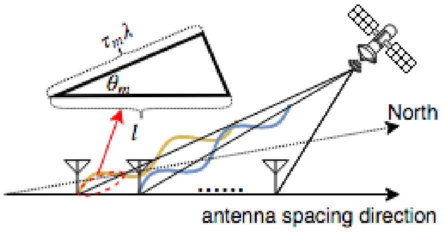


Fig. 4. Schematic of the geometrical relationship of Satellite and GNSS array signal.

$$\begin{aligned}\chi_1 \times \chi_m^* &= \left(aD(n)R(\tau)sinc(f_eT_{coh})\right)^2 \times [\cos(\phi_e + 2\pi f_c\tau_m) - j\sin(\phi_e + 2\pi f_c\tau_m)] \times [\cos(\phi_e) + j\sin(\phi_e)] \\ &= \left(aD(n)R(\tau)sinc(f_eT_{coh})\right)^2 \times [\cos(\phi_e + 2\pi f_c\tau_m)\cos(\phi_e) + j\cos(\phi_e + 2\pi f_c\tau_m)\sin(\phi_e) - j\sin(\phi_e + 2\pi f_c\tau_m)\cos(\phi_e) + \sin(\phi_e + 2\pi f_c\tau_m)\sin(\phi_e)] \\ &= \frac{1}{2}\{\cos(2\phi_e + \tau_m) + \cos(\tau_m) + j[\sin(2\phi_e + 2\pi f_c\tau_m) - \sin(\tau_m)] - j[\sin(2\phi_e + 2\pi f_c\tau_m) + \sin(\tau_m)] - [\cos(2\phi_e + 2\pi f_c\tau_m) - \cos(\tau_m)]\} \\ &= \left(aD(n)R(\tau)sinc(f_eT_{coh})\right)^2 \times \mathrm{e}^{-\mathrm{j}2\pi f_c\tau_\mathrm{m}}.\end{aligned} \tag{19}$$

Fig. 4 illustrates the geometrical relationship of Satellite and GNSS array signal. The carrier phase difference between channel 1 and channel 2 is taken as an example of how to calculate the AoA from τ_m. The AoA of m_{th} element, θ_m, can be denoted as:

$$\theta_m = \arccos(\frac{\tau_m\lambda}{l}).$$

Where λ is the wavelength of the received GNSS signal, and l is the distance between the two antennas.

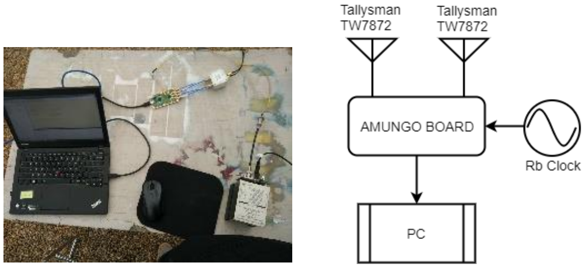
IV. Live Data Validation and Performance Analyses

In this section, we use live GPS L1 data to validate the proposed AoA estimation technique. Then we provide qualitative analyses in terms of receiver implementation complexity.

A. *Live Data Validation*

1) GPS L1 Data Collection

The actual experiment setup and the schematic are shown in Fig. 5. In the data collection configurations, two Tallysman TW7872 antennas [7] are used to collect RF data. The Amungo board [8] is used as a front-end to down-convert GPS RF signal to IF data. A Rubidium clock is used to provide a stable reference clock input for front-end sampling.



(a) (b)

Fig. 5. (a) Actual setup, and (b) data collection schematic.

Two Tallysman TW7872 dual-band GNSS antennas were used to collection GPS RF signal. The TW7872 is a dual-band, antenna for reception of GPS L1/L2, GLONASS G1/G2, BeiDou B1, Galileo E1 and is specially designed for precision dual frequency positioning. Two TW7872 antennas that used in this paper can be found in Fig. 6. In this paper, two antennas were placed with 9.5 cm spacing, which is the half-wavelength for GPS L1 carrier at 1575.42 MHz.

The Amungo board, as shown in Fig. 7 is a Multi-band RF transceivers for the L-band, that equipped with NT1065 "Nomada" as shown in Fig. 8.

Fig. 6. Tallysman TW7872 antennas.

Fig. 7. Amungo board.

NT1065 [9] is a four-channel radio frequency (RF) front-end for simultaneous reception of GPS, GLONASS, Galileo, Beidou, IRNSS, QZSS and SBAS signals of various frequency bands L1, L2, L3, L5, E1, E5a, E5b, E6, B1, B2, B3.

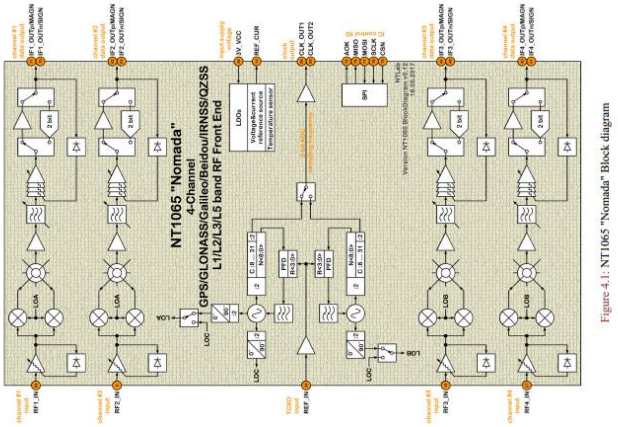

NT1065 "Nomada"
4-Channel
GPS/GLONASS/Galileo/Beidou/IRNSS/QZSS
L1/L2/L3/L5 band RF Front End
LDOs
SPI
Figure 4.1: NT1065 "Nomada" Block diagram

Fig. 8 NT1065 block diagram.

The GPS L1 data were collected on September 29th, 2017, at the Rooftop of the University of Colorado Boulder Discovery Learning Center. The selected data scenario is shown in Fig. 9.

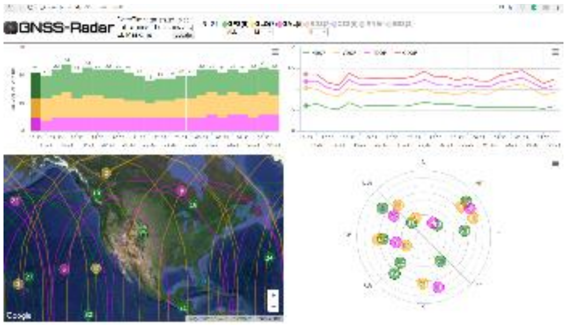


Fig. 9. Data collection scenario on September 29th, 2017, at the Rooftop of CU Boulder DLC.

2) *Post-processing for AoA Estimations*

Post-processing is a common and flexible approach for GPS data processing. In addition, the Software Defined Receiver (SDR) technique is frequently used for GPS data processing. In this paper, an open source Matlab-based SDR was used [10] to implement the proposed architecture shown in Fig. 3.

Table I is typical settings for the SDR implemented in this paper. The definations of those settings are detailed in [10], so they will not be described herein. Note that except for integration time T_{coh}, the remaining parameters are for the channel 1 tracking loop. As mentioned above, instead of generating a local carrier and code replica, the correlation processor uses the local replica from channel 1 tracking loop. Thus, it does not need to be configured with DLL and PLL settings.

TABLE I. SDR SETTINGS

Parameters	Value
DLL damping ratio	0.7
DLL damping ratio	1.5
DLL Correlator spacing	0.5
PLL damping ratio	0.7
PLL noise BW	20
Coherent integration time T_{coh}	1 ms

The correlator outputs of the prompt branch of the channel 1 tracking loop of PRN 18 is shown in Fig. 10. From Fig. 10 we can see that the magnitude of I_P is significantly higher than Q_P, which means the carrier tracking loop successfully tracked the received signal.

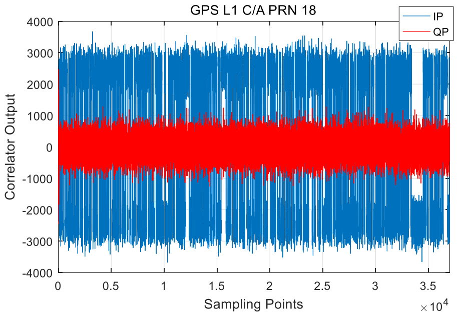


Fig. 10. Correlator output.

Positioning solutions obtained from SDR (channel 1 processing) and actual location obtained from Google Maps are shown in Fig. 11 and Fig. 12 respectively. From Fig. 11 and Fig. 12 we can conclude that the GPS L1 C/A signal has been successfully processed.

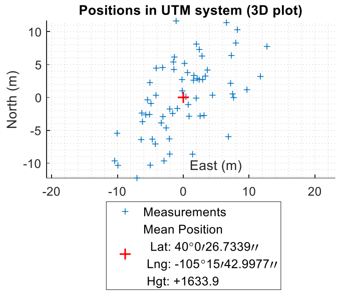


Fig. 11. Positioning solutions of SDR.

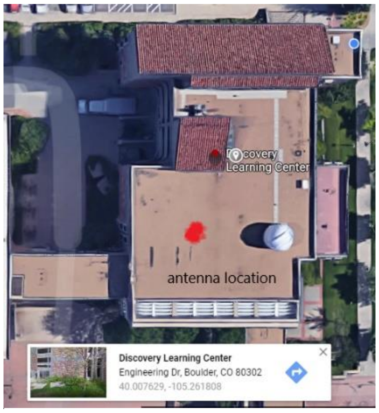

Discovery Learning Center
antenna location
Discovery Learning Center
Engineering Dr, Boulder, CO 80302
40.007629, -105.261808

Fig. 12. Screen shot from Google Maps.

To further enhance the signal strength, we should use a longer non-coherent integration time T_{ncoh}. However, in order to estimate AoA more precisely in dynamic environment, T_{ncoh} cannot be too long. Thus, we choose a intermediate value of $T_{ncoh} = 20$ ms. The process of non-coherent integration of m_{th} channel can be expressed as:

$$\chi_m|_{N_{nc}} = \frac{1}{N_{nc}} \sum_{n=1}^{N_{nc}} \chi_m(n).$$

Where N_{nc} is the non-coherent number where $N_{nc} = \frac{T_{ncoh}}{T_{coh}}$. Note that this process should be done after navigation data bit synchronization. Fig. 13 shows the CPDE AoA estimations of GPS L1 C/A signals from PRN 18, PRN 21, PRN 10, and PRN 8.

The mean elevation/azimuth angles in 37 ms for four satellites obtained from SDR processing are listed in Table II. Note that since the orientation of the two antennas is not pointing to the north, the estimated AoA angles do not match the elevation angles shown in Table II.

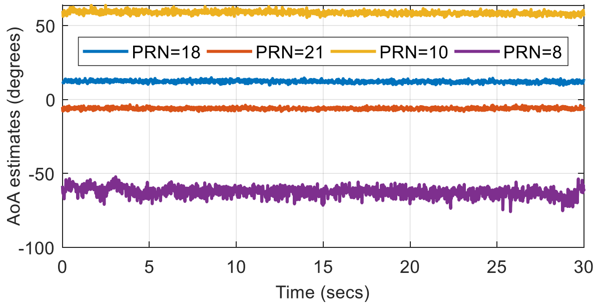


Fig. 13. AoA estimations of GPS L1 C/A using CPDE.

TABLE II. MEAN EL./AZ. ANGLES

PRN	Az.	El.
18	7.9801°	72.7560°
21	133.2669°	66.7735°
10	293.7491°	54.9897°
8	318.4287°	15.4279°

To further validate the correctness of CPDE, we plot the comparisons between CPDE and MUSIC in Fig. 14. In a MUSIC implementation, the angle resolution is set to be 0.5° and 2°. In this scenario, two settings result in 720 times and 180 times of MUSIC processing for every estimated value, respectively. Every MUSIC processing includes a 360° angle space searching, as well as MUSIC spectrum calculations.

As it can be seen that the results of CPDE and MUSIC (reso=0.5) estimates are consistent. However, for MUSIC with resolution of 2°, there is a bias of ~0.5° compared to CPDE and MUSIC (reso=0.5) for part of the estimated values.

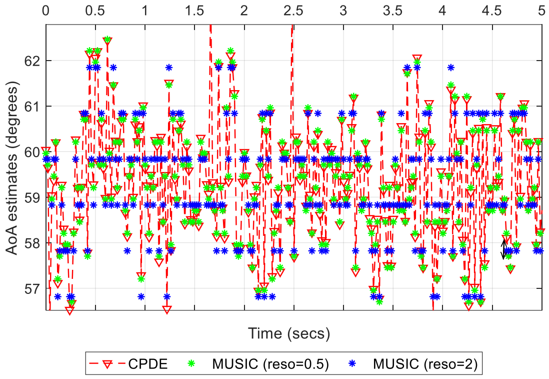


Fig. 14. AoA estimates comparisons between CPDE and MUSIC (reso = 1°) for PRN 10.

3) *Beam-steering Architecture and Validation*

A beam-steering technique is used to further validate the proposed CPDE algorithm. The pre-correlation beam-steering architecture is shown in Fig. 15. As shown, the beam-steering and combination process is done before correlator. For every estimated value, the AoA estimates stream is sent into the phase shifter. Then the phase shifter is used to steer the 2nd channel to the direction of the 1st channel. The steered 2nd channel signal and the 1st channel signal are combined together and then sent into the GNSS signal tracking loop.

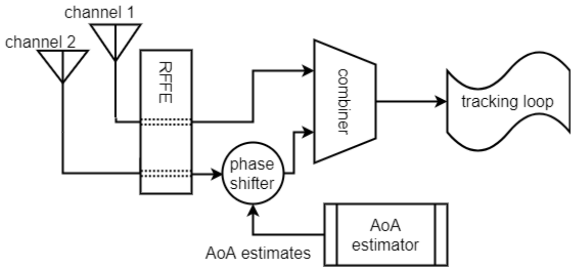


Fig. 15. Pre-correlation beam-steering architecture.

The comparison of CNo of the single channel signal and the beam-steered & combined dual-channel signal for PRN 21 is shown in Fig. 16. The averaged CNo improvement is 3.8439 dB-Hz.

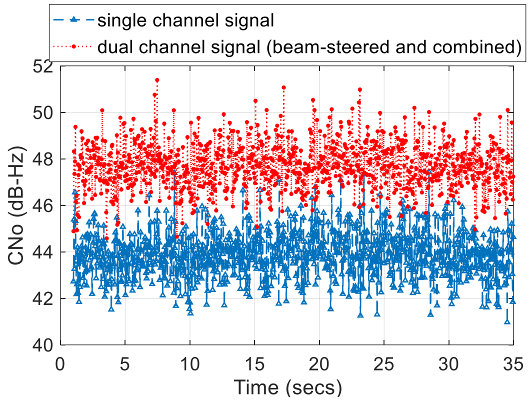


Fig. 16. CNo comparison of single and dual channel signal for PRN 21.

Fig. 17 shows the CNo improvement for PRN 18, PRN 21, and PRN 10. Note that for different elevation angles. Table III shows the Averaged CN$_0$ improvements vs. PRNs and AoA estimates. As it can be seen that the averaged improvements of beam-steering various from 1.34~3.84 dB-Hz. We can also see the nearer the AoA to 0°, the better its signal strength improvement is.

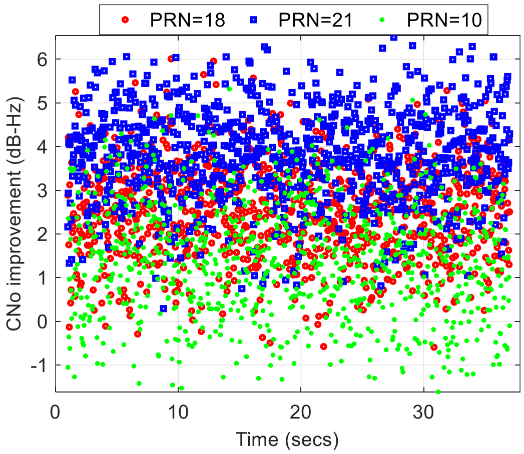


Fig. 17. CNo improvement for three satellites.

TABLE III. AVERAGED CN$_0$ IMPROVEMENT VS. PRNS AND AOA ESTIMATES

PRN	Averaged CN$_0$ Improvement (dB-Hz)	Averaged AoA Estimate (°)
18	2.4907	12.1370
21	3.8439	-6.0692
10	1.3460	58.6869

B. Analyses of Receiver Implementation Complexity

The proposed CPDE is based on extracting the phase difference between elements. Therefore, compared to a subspace-based AoA estimation algorithms that focus on searching in the signal subspace and noise subspace, CPDE has a significant lower implementation complexity. The following are two simple demonstrations of the implementation of CPDE and MUSIC for post-correlated signals. Since the detailed information and definitions of MUSIC has been documented in [3], they will not be illustrated herein.

```
CPDE Algorithm
Vector Multiplication: $\chi_1 \times \chi_m^*$
Obtaining angle: $\tau_m(n) = -\mathrm{angle}\{\chi_1 \times \chi_m^*\}$
Converting $\tau_m$ to $\theta_m$: $\theta_m = \arccos(\frac{\tau_m \lambda}{l})$
```

```
MUSIC Algorithm
Matrix Multiplication: $\mathrm{Q_n} = \frac{\begin{bmatrix}\chi_1\\ \chi_m\end{bmatrix} \times \begin{bmatrix}\chi_1\\ \chi_m\end{bmatrix}^{\mathrm{T}}}{\mathrm{n}}$
Calculating eigenvalues of $\mathrm{Q_n}$
Searching for $\theta \in (0°, 360°)$ that maximums:
    $\mathrm{P_{MUSIC}}(\theta) = \frac{1}{\boldsymbol{\alpha}^{\mathrm{H}}(\theta)\mathrm{Q_n}\mathrm{Q_n^H}\boldsymbol{\alpha}(\theta)}$
```

From the above analyses, we can see that for every estimated value, CPDE only needs $1 \times$ simple vectors multiplication, $1 \times$angle calculation, and $1 \times$ arccos operation. However, for MUSIC, it has to do $1 \times$ complicated matrix multiplication, $1 \times$ eigenvalues calculation, and searches in the whole angle space ($360°$) for the maximum value of MUSIC spectrum. As mentioned above, for every AoA estimate, MUSIC needs to search 720 or 180 times for angle resolution of 0.5° and 2°, respectively. The rough analyses show that CPDE has a lower implementation complexity than MUSIC.

V. CONCLUSIONS

In this paper, we proposed a low complexity GNSS array signal processing algorithm. The live data validation results show that the proposed algorithm can successfully estimate the AoA of array signal. More importantly, the proposed architecture is easy to implement and has a low receiver complexity.

REFERENCES

[1] L. C. Godara, "Application of antenna arrays to mobile communications. II. Beam-forming and direction-of-arrival considerations," Proc. IEEE, 85(8): pp. 1195-1245, 1997.

[2] A. Alexiou and M. Haardt, "Smart antenna technologies for future wireless systems: Trends and challenges," Communications Magazine IEEE, 42(9): pp. 90-97, 2004.

[3] R. Schmidt, "Multiple emitter location and signal parameter estimation," IEEE Transactions on Antennas & Propagation, 34(3): pp. 276-280, 1986.

[4] B. D. Rao and K. V. S. Hari, "Performance analysis of root-MUSIC," IEEE Transactions on Acoustics Speech & Signal Processing, 37(12): pp. 1939-1949, 1989.

[5] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," IEEE Transactions on Acoustics Speech & Signal Processing, 37(7): pp. 984-995, 2002.

[6] Signals and Array Signal Processing for Global Navigation Satellite Systems. Available from: https://www.msv.ei.tum.de/en/courses/master-lectures/signals-and-array-signal-processing-for-global-navigation-satellite-systems/

[7] Tallysman Home Page. Available from: http://www.tallysman.com/index.php/gnss/products/antennas-l1l2-g1g2/tw7872/

[8] Amungo Navigation. Available from: https://www.Amungo-navigation.com/nut4nt

[9] NT1065_USB3_description_v1.03.pdf. Available from: http://www.ntlab.com/IP/NT1065/NT1065_USB3_description_v1.03.pdf

[10] B. Kai, et al., A Software-Defined GPS and Galileo Receiver. Birkhäuser Boston, pp. 1632-1637, 2007.