Correlation Agnostic Fusion for Enabling Drone Flights in Challenging Environments

Aryan Naveen

Autonomy and Navigation Technology (ANT) Center
Air Force Institute of Technology
United States of America

October 27, 2020

1 Introduction

Unmanned aerial vehicles (flying drones) are especially serviceable for completing tasks that are not feasible by humans or highly dangerous; from search and rescue in hazardous environments, delivery services in response to natural disasters, or even exploration and data-collection in untraveled locations. Not to even mention the other benefits that drones provide such as lower fuel consumption and less environmental footprint. The wide expanse of applications for autonomous drones will result in a requirement for *robustness* in a wide variety of environments that may each present a unique potential failure or obstacle. In order for drones to operate safely, accurate pose (position, velocity, and attitude) estimation must be performed for effective path-planning and following to take place. The variety of environments that drones may be required to operate in introduces various challenges for the drone's ability to accurate estimate it's pose. For instance, the most frequently used and publicly-available resource for pose-determination for drones is the Global Positioning System (GPS), but there are a wide variety of challenges with this system including but not limited to the following:

- Urban areas cause several problems for GPS, including blocked satellites and multi-path interference from satellite signals reflecting off buildings
- Frequency jamming and spoofing, which can cause the drone to no longer receive accurate estimates. [1]

The above limitations of GPS frequently result in a vulnerable state for the drone as it can no longer determine it's position and velocity. The attitude of the drone can drift without bound without GPS as well. A well studied solution to the challenges presented by GPS is the use of all Global Navigation Satellite Systems (GNSS). GNSS possibilities include not only GPS, but also Russia's GLONASS, European Union's Galileo and China's Beidou. Because each of these satellites utilize different frequencies and signal processing techniques for data signals, the localization becomes more robust to multi-pathing and jamming. It also becomes more difficult to spoof as not just one but several systems have to be spoofed. In addition, with more systems, the the number of satellites used for localization purposes can be much larger, increasing the ability to continue to provide robust pose estimations in challenging environments.

Using GNSS to localize an autonomous drone is a well studied field, and most of the previous research focuses on utilizing kalman filters which fuse together the GNSS signals through bayesian fusion by using information/kalman filters [2] [3] [4]. While these papers, focus on producing accurate pose estimates our research attempts to further improve both pose estimates as well as the uncertainty estimates, which is a significant challenge to the safe and accurate control for a unmanned aerial vehicle.

We illustrate the importance of accurate uncertainty estimates with a simple example. Envision a scenario wherein a drone is attempting to autonomously navigate between 2 building 10 meters apart. In the case, where the estimated uncertainty is 0.5 meters whilst the true uncertainty is 0.5 meters as well, the drone would be able to safely navigate between the two buildings. However, if the uncertainty estimate is overconfident and estimates an uncertainty of 0.5 meters when the true uncertainty is 10 meters, then the drone would believe itself to be able to safely fly between those 2 buildings when in reality it is extremely vulnerable to dangerous situations due to the overconfident estimate on the pose.

Current practices in GNSS pose estimation involve assuming independence between all satellites and implementing Bayesian (naive) data fusion. However, we hypothesize that the satellites are not independent of each other as there are a multitude of reasons that can cause correlation among a group of satellites including:

- There are multiple frequencies that broadcast from the same satellite so if there is an error they may be common between the frequencies
- Satellites that are in the same constellation (system) may have the same type of errors exhibited in them
- The terrain itself may introduce errors, inducing correlated errors between satellites

In this paper we have two main contributions: (1) the implementation of correlation agnostic fusion for GNSS localization and (2) a novel correlation agnostic fusion technique yielding extremely accurate uncertainty estimates in this application.

This paper is organized as follows. In Section 2, we mathematically define the data fusion problem and introduce the notation that will be utilized for the remainder of this paper. We also review the Ellipsoidal Intersection and Covariance Intersection approach to data fusion to provide further context for current correlation-agnostic techniques. In Section 3, we introduce our novel correlation data fusion technique based on probablistically constraining the amount of common information between the input distributions. Furthermore we also provide simulation results to quantitatively demonstrate the degree of optimization provided through the use of the novel technique. Section 4 outlines both background information about the mathematical model used to preprocess the satellite signals as well as the actual algorithm used to update the current estimates with new information from each satellite. Section 6 concludes the paper and discusses future improvements to our method.

2 Correlation Agnostic Fusion Review

The fusion problem considered in this paper is when two sources (satellites) of information about a quantity (estimated UAV pose) are available, each providing a probability distribution function (PDF) representing its knowledge, labeled P_a and P_b. When the two PDFs are (conditionally) independent, the fused PDF (P_f) can be obtained as:

$$P_f(x) = \frac{P_a(x)P_b(x)}{\int_S P_a(x)P_b(x)dx} \, \forall x \in S \tag{1}$$

where S is the region of support for both input PDFs (i.e. $P(x) = 0$ for all $x \notin S$).

When the PDFs P_a and P_b contain some common information, we can write the inputs in terms of the common and independent information as:

$$P_a(x) \simeq P_{a\backslash c}(x)P_c(x) \tag{2}$$

$$P_b(x) \simeq P_{b\backslash c}(x)P_c(x), \tag{3}$$

where $\simeq$ denotes equality up to a normalization factor. The *optimal* data fusion result in this case is:

$$P_\phi(x) \simeq P_{a\backslash c}(x)P_{b\backslash c}(x)P_c(x). \tag{4}$$

Because we are focusing on Gaussian distributions in this paper, the input, common, and fused distributions can be completely represented by the mean (μ) and covariance matrix (C) associated with that distribution. Therefore, we have (μ_a, C_a) and (μ_b, C_b) representing the information to be fused and the common information represented by (μ_c, C_c). The input distributions can be decomposed into independent estimates $(\mu_{a\backslash c}, C_{a\backslash c})$, $(\mu_{b\backslash c}, C_{b\backslash c})$, and (μ_c, C_c) according to

$$\begin{cases} \mu_a & = C_a(C_{a\backslash c}^{-1}\mu_{a\backslash c} + C_c^{-1}\mu_c) \\ C_a^{-1} & = C_{a\backslash c}^{-1} + C_c^{-1} \end{cases} \tag{5}$$

and

$$\begin{cases} \mu_b & = C_b(C_{b\backslash c}^{-1}\mu_{b\backslash c} + C_c^{-1}\mu_c) \\ C_b^{-1} & = C_{b\backslash c}^{-1} + C_c^{-1} \end{cases} \tag{6}$$

Using this specialization of (2) and (3), we can obtain the optimal fusion result (denoted by

ϕ) of

$$\mu_{\phi} = C_{\phi}(C_{a\backslash c}^{-1}\mu_{a\backslash c} + C_{b\backslash c}^{-1}\mu_{b\backslash c} + C_{c}^{-1}\mu_{c}) \tag{7}$$

$$C_{\phi}^{-1} = C_{a\backslash c}^{-1} + C_{b\backslash c}^{-1} + C_{C}^{-1} \tag{8}$$

Note that in all of these equations, all covariance matrices are required to be positive semi-definite (p.s.d.). A matrix (M) is p.s.d. if

$$x^{\top}Mx \geq 0 \; \forall x. \tag{9}$$

Ensuring all matrices are p.s.d. leads to the following definition:

Definition 1. *A common covariance matrix C_c is* admissible *if the matrices $C_c - C_a$ and $C_c - C_b$ are p.s.d.*

2.1 Covariance Intersection Fusion Algorithm

To review, Covariance Intersection (CI) [5] is the convex combination of the initial distribution's mean and covariances represented in the information space. Given the input distributions of P_a and P_b, we can find the fused distribution based on CI using the following equations.

$$C_{fus}^{-1} = \omega_a C_a^{1} + \omega_b C_b^{1} \tag{10}$$

$$\mu_{fus} = C_{fus}(\omega_a C_a^{-1}\mu_a + \omega_b C_b^{-1}\mu_b) \tag{11}$$

$$\omega_a + \omega_b = 1 \tag{12}$$

where ω represents the weight for each input distribution. The selection of ω is based on the following optimization, where CI tries to find the appropriate weightage on each distribution to minimize either the determinant or trace- in this paper determinant is used- of C_{fus}.

$$\omega := \underset{w}{\operatorname{argmin}} \, |C_{fus}| \tag{13}$$

2.2 Ellipsoidal Intersection Fusion Algorithm

To review, the Ellipsoidal Intersection [6] algorithm attempts to solve the optimization problem

$$\begin{aligned} C_c = & \underset{\Upsilon}{\operatorname{argmin}} \, |\Upsilon| \\ & s.t. \; x^\top(\Upsilon - C_a)x \geq 0 \; \forall x. \\ & \quad x^\top(\Upsilon - C_b)x \geq 0 \; \forall x. \end{aligned} \tag{14}$$

where $|\Upsilon|$ represents the determinant of the matrix and the constraints express the admissible constraint of Definition 1.

To find the matrix that has the minimum determinant, given the constraints, ellipsoidal geometry is used to derive the closed form expression

$$\begin{aligned} D_{C_c} &= \begin{cases} \max(D_2,\,1) & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \\ C_c &= E_a D_a^{0.5} Q_b D_{C_c} Q_b^T D_a^{0.5} E_a^T + \epsilon I \end{aligned} \tag{15}$$

where ϵ is a small numerical constant used for numerical stability when computing the mean and

$$\begin{aligned} C_a &= E_a D_a E_a^T \\ Q_b D_b Q_b^T &= D_a^{-0.5} E_a^T C_b E_a D_a^{-0.5} \end{aligned} \tag{16}$$

From the calculated C_c, EI finds the common mean as

$$\mu_c = (C_a^{-1} + C_b^{-1} - 2C_c^{-1})^{-1}(C_{a\backslash c}\mu_a + C_{b\backslash c}\mu_b) \tag{17}$$

Given an estimated common mean and covariance, value, the fused estimate (μ_f, C_f) is found similarly to Equations (7) and (8).

3 Probablistically Conservative Data Fusion

Note that EI, CI and other correlation agnostic fusion techniques have, in essence, used the covariance matrices of the inputs to impose constraints on what the fused result can be. By finding the maximum amount of common information that could be present, *within the constraints imposed by the input covariance matrices*, a correlation agnostic fusion technique is achieved. We believe that the means of the input distributions can also be used to indicate how much common information is present in the inputs to a fusion algorithm. Conceptually, if the means of two input distributions are different, even if their covariance is the same, that is an indicator of independent information being present in the inputs.

We assume that the means of each input distribution can be constructed as

$$\mu = x + \nu \tag{18}$$

where μ is the mean of the input distribution, x is the true value of the quantity being estimated, and ν is a random variable sampled from a zero mean Gaussian variable with covariance C. When an input distribution is received by the fusion algorithm, it receives both μ and C. Because x from Equation (18) is assumed constant across all inputs to a fusion algorithm, this leads to the following key proposition:

Proposition 2. *If multiple Gaussian PDFs express uncertainty about a single estimate, the differences in their means are due solely to the presence of noise.*

Corollary 3. *A difference of means between two inputs implies the presence of independent information in the inputs.*

Based on Proposition 2 and assuming $\mu_{a\backslash c}$, $\mu_{b\backslash c}$, and μ_c are estimates of the same quantity, then the differences in mean value should be due solely to noise, i.e. different estimates of the quantity being estimated. This proposition can be expressed as a null hypothesis test

$$H_0 : \mu_{a\backslash c} - \mu_{b\backslash c} = 0 \tag{19}$$

$$H_A : \mu_{a\backslash c} - \mu_{b\backslash c} \neq 0 \tag{20}$$

where the alternative hypothesis is that $\mu_{a\backslash c}$ and $\mu_{b\backslash c}$ are estimates of different quantities.[1] If the alternative hypothesis were to be accepted, that would indicate that the two input data sources were actually measurements of different quantities and should not be fused together. This leads to the following definition:

Definition 4. *An estimate of common information is* probablistically admissible *if there is not sufficient evidence to reject the null hypothesis that they are estimates of the same quantity.*

[1] We have selected $\mu_{a\backslash c}$ and $\mu_{b\backslash c}$ because the most likely μ_c will always be "in-between" these two μ values. Therefore, using $\mu_{a\backslash c}$ and $\mu_{b\backslash c}$ will give us the most restrictive constraints on common covariance.

By constraining the common information estimates to probabilisticly admissible estimates, the estimated quantity of common information will be reduced, leading to fusion results that are less conservative than previous techniques.

For Gaussian distributions, the null hypothesis test can be expressed as a threshold on the Mahalanobis distance

$$\Psi(C_c) = (\mu_{a\backslash c} - \mu_{b\backslash c})^\top (C_{a\backslash c} + C_{b\backslash c})^{-1} (\mu_{a\backslash c} - \mu_{b\backslash c}) \tag{21}$$

where $C_{a\backslash c}$, $C_{b\backslash c}$, $\mu_{a\backslash c}$, and $\mu_{b\backslash c}$ are all functions of C_c using Equations (5) and (6). The threshold value γ is derived from a user selected level of significance $\alpha = p(\Psi \geq \gamma | H_0)$, or the probability of rejecting the null hypothesis when the null hypothesis was actually true. Given α, the threshold for the Mahalanobis distance γ can be computed from a chi-squared distribution of degree equal to the number of dimensions in the state space. If the Mahalanobis distance (Ψ) is greater than γ, the null hypothesis is rejected.

The central idea behind Ellipsoidal Intersection is a constrained optimization problem (Equation (14)) wherein the determinant of the common covariance is the optimization variable to be minimized, while constraining the common covariance to *admissible* values. Similarly, we can create a probabilisticly constrained (PC) fusion technique described using the following optimization problem:

$$C_c = C_{c,EI} + SS^T \tag{22}$$

$$\begin{aligned} S :=& \operatorname*{argmin}_{\Upsilon} |\Upsilon| \\ s.t.\ & \Psi(C_{c,EI} + \Upsilon\Upsilon^T) \leq \gamma \end{aligned} \tag{23}$$

where Υ is a lower triangular matrix, γ is the chi-square threshold selected by the user for a problem of dimension N, $C_{c,EI}$ is the common covariance computed by ellipsoidal intersection, and $\Psi(...)$ is described by (21). This optimization problem finds the smallest matrix SS^T (where small is defined by its determinant) required to make C_c meet the probabilistic constraint. We use the SS^T parameterization of symmetric matrices because (1) this ensures that we are generating a p.s.d. matrix and (2) the determinant of a triangular matrix is just the product of its diagonals, making it easy to compute. Note that because SS^T is added to the common covariance from ellipsoidal intersection, the computed common covariance will be both admissible *and* probabilisticly admissible.

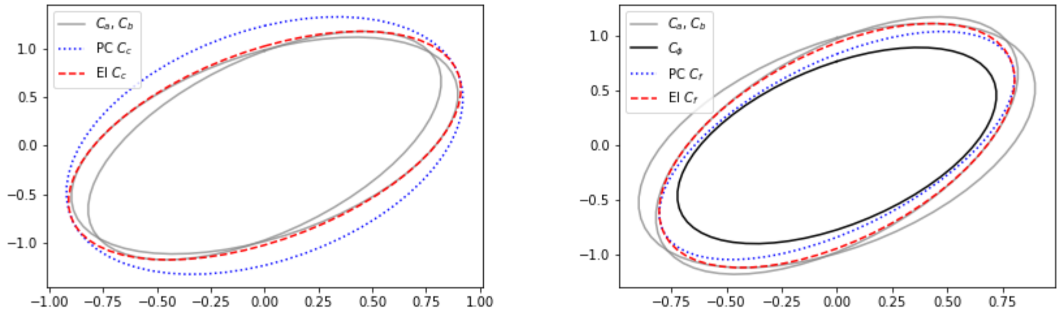
3.1 Simulation Results

To evaluate the effectiveness of the probabilisticly constrained fusion algorithm, we consider the scenario where two distributions are being fused together and evaluate both the output covariance and mean squared error when using the probabilisticly constrained fusion method (PC) and the traditional ellipsoidal intersection (EI) method.

3.1.1 Two input distributions

In this section, we analyze the result performing fusion on two input Gaussian distributions. We first analyze a single instance to show the potential results of PC fusion, followed by some Monte Carlo simulations showing more general performance.

In Figure 1, we show, for a single instance, the covariance output from both PC and EI fusion. In subfigure (a), we plot the level set of the common covariance matrix derived by both EI and the PC techniques. Note that the estimated common covariance of the PC method is larger, meaning it estimates less common information between the two inputs. This decrease in common information leads to the fused covariance outputs illustrated in subfigure (b). Note that the fused covariance when probabilisticly constrained is significantly smaller and closer to the truth than when using EI.



(a) Common covariances

(b) Fused covariances

Figure 1: Comparison of PC (blue dotted) and EI (red dashed) fusion approaches. Light gray ellipses denote the input covariances and dark black the optimal fusion result (subfigure (b)). Note that PC has a larger estimated common covariance (a) leading to a smaller fused covariance output (b).

While the results illustrated in Figure 1 are promising, we wanted to know how often PC leads to a significant improvement in fusion compared with EI. To perform this analysis, we ran Monte Carlo simulations over 35 different sets of 2 input distributions. To generate input distributions, we used an Inverse Wishart distribution to generate three covariance matrices corresponding with $C_{a\backslash c}$, $C_{b\backslash c}$, and C_c. A random sample from a Gaussian with the corresponding covariance is then drawn to generate $\mu_{a\backslash c}$, $\mu_{b\backslash c}$, and μ_c. These values are then used to generate the inputs (μ_a,C_a) and (μ_b,C_b) according to formulas (5) and (6). To evaluate the effect of how similar the input distributions are, we varied the degrees of freedom of the Inverse Wishart distribution used to generate the covariances over the values 2, 10, and 100 (leading to a total of 105 sets of input distributions.) As the degrees of freedom increased, the sampled covariances will approach the two-dimensional identity matrix (i.e. all covariance matrices are 2 × 2.) Note that with higher degrees of freedom, all three covariances will be more similar, making C_a and C_b more similar as well. Each set of input distributions was fused with the proposed technique (PC fusion) at α values of .05 and .01 and with EI fusion.

Table 1: Monte-carlo inverse wishart simulation result comparison between EI and PC for different degrees of freedom.

	PC, α =0.05	**PC, α =0.01**	**EI**
	df=2		
Avg. Determinant	0.7229	0.7418	0.7946
% smaller than EI	9.02%	6.64%	
MSE	1.927	1.9442	2.019
% smaller than EI	4.56%	3.70%	
	df=10		
Avg. Determinant	0.415	0.435	0.485
% smaller than EI	14.47%	10.28%	
MSE	0.574	0.588	0.640
% smaller than EI	10.32%	8.09%	
	df=100		
Avg. Determinant	0.537	0.565	0.651
% smaller than EI	17.62%	13.24%	
MSE	0.672	0.684	0.867
% smaller than EI	22.46%	21.15%	

The effectiveness of the different fusion techniques were evaluated using two metrics: (1) the determinant of the output (fused) covariance and (2) the mean squared error (MSE) of the fused means. The goal of using the PC is to compute fused covariances that are less conservative than EI while still being conservative compared to the optimal fusion result. Therefore, we would expect the determinants of the fused covariances from the proposed technique to be smaller than the EI-derived covariances. The error used to compute MSE is defined as $\mu_{f,PC} - \mu_\phi$, where $\mu_{f,PC}$ is the mean of the fused distribution using PC and μ_ϕ is the optimal mean as defined in Equation (8). The MSE due to EI was computed similarly.

The results of the Monte Carlo simulations are summarized in Table 1. Note that:

- Using the PC leads to a significant reduction in the size of the fused covariance determinant, an indicator that it is less conservative.
- The MSE is also significantly smaller with PC fusion
- As the α value gets smaller, the PC approaches the EI technique, as would be expected.
- As the input covariances become more similar (the degrees of freedom increase), the advantage of using PC fusion increases.

Figure 2 also shows a box and whisker plot showing the percentage decrease in determinate using PC fusion at $\alpha = .01$ compared with EI fusion for the same monte-carlo simulations described previously. This chart demonstrates that (1) PC fusion is always as good as and often better than EI fusion (no negative percentages) and (2) as the input covariances become more similar, the advantage of PC fusion over EI fusion is amplified.

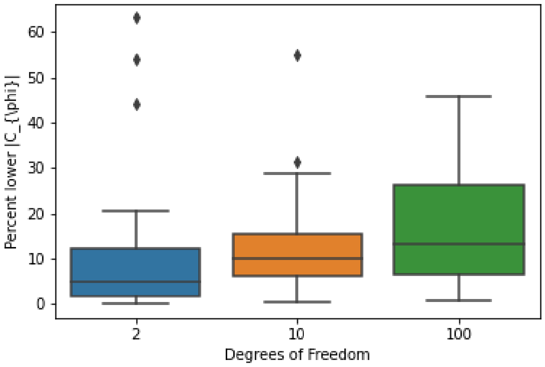


Figure 2: Decrease in fused covariance determinate (y-axis) using PC ($\alpha = 0.01$) as opposed to EI fusion at different inverse wishart degrees of freedom (x-axis). Each degree of freedom had 35 runs of data.

4 Fusion of Multi-GNSS Signal for UAV Localization

While the results in section 3.1 demonstrate the advantages of PC fusion, this paper is focused specifically on how to apply correlation agnostic fusion to pose estimation for drones using multiple GNSS systems. In this section, we specifically focus on the estimation of velocity and performing correlation-agnostic fusion to obtain accurate uncertainty estimates for the velocity as well. Note that the general technique for estimating velocity using multiple

GNSS systems was previously developed in [7], but using correlation-agnostic fusion is a novel contribution of this paper.

In the following subsections, we first review the estimation of velocity from satellite signals. We then describe how to modify this procedure to utilize correlation agnostic fusion techniques to estimate velocity and its uncertainty.

4.1 Multi-GNSS velocity estimation

Following the procedure outlined in [7], we estimate the velocity of a multi-GNSS receiver using a Time-differenced Carrier Phase (TDCP) solution. While there are several possible procedures for estimating velocity from GNSS systems, we selected this particular technique because it (a) was designed specifically for multi-GNSS systems and (b) intrinsically (by using a differencing method) eliminates many possible sources of error, including those that typically cause disagreement between different GNSS systems (e.g., GPS and Galileo).

TDCP is used to calculate the displacement of a GNSS antenna (and the change in receiver clock) between two time stamps [8][9], which are t_0 and t_1 in Figure 3.

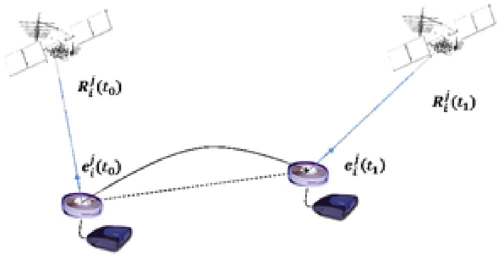


Figure 3: Displacement with Time Differenced Carrier Phase

Let U_i be the location of the i^{th} antenna/reciever, and S^j be the location of the j^{th} satellite. As illustrated in Figure 3, e_i^j represents the unit vector pointing from user antenna to satellite, and R_i^j is the range between them. A raw user position estimate is needed to compute e_i^j.

$$R_i^j(t_0) = \left(SV^j(t_0) - U_i(T_0)\right) \cdot e^j(t_0) \tag{24}$$

The same thing can be found for the range observed at t_1, from which the LOS range difference between both epochs can be found using

$$\begin{aligned} \Delta R_i^j &= R_i^j(t_1) - R_i^j(t_0) = (SV^j(t_1) - U_i(t_1)) \cdot e_i^j(t_1) - (SV^j(t_0) - U_i(t_0)) \cdot e_i^j(t_0) && (25) \\ &= SV^j(t_1) \cdot e_i^j(t_1) - SV^j(t_0) \cdot e_i^j(t_0) - U_i(t_0) \cdot e_i^j(t_1) + U_i(t_0) \cdot e_i^j(t_0) - \Delta U \cdot e_i^j(t_1) && (26) \end{aligned}$$

where $\Delta U = U_i(t_1) - U_i(t_0)$, which is the user displacement vector. It can be computed from multiple observations of ΔR_i^j in the following steps. First, a GNSS satellite Doppler correction term and a geometric correction term are defined.

$$corr_{sv} = SV^j(t_1) \cdot e_i^j(t_1) - SV^j(t_0) \cdot e_i^j(t_0) \tag{27}$$

$$corr_{geo} = U_i(t_0) \cdot \left(e_i^j(t_0) - e_i^j(t_1)\right) \tag{28}$$

By rearranging equation (25), ΔR_i^j can be related to displacement ΔU via

$$\Delta R_i^j - corr_{sv} - corr_{geo} = -\Delta U^\top e_i^j(t_1) \tag{29}$$

To measure R_i^j, the accumulated Doppler range (ADR) can be used in a receiver. Errors in the ADR can may include (but are not limited to) satellite clock error ($B_i^{j,k,l}$); ionosphere (T_i^j) and troposphere ($I_i^{j,k,l}$) effects; ground and airborne multipath effects ($M_i^{j,k,l}$); carrier phase cycle slip ($N_i^{j,k,l}$); time-variant receiver clock bias (b_i [8]); and noise and any other unmodeled errors ($\epsilon_i^{j,k,l}$). The notation ^{j,k,l} refers to the j-th satellite in the k-th constellation broadcasting on the l-th channel. Including all these error sources leads to the following model for ADR:

$$ADR_i^{j,k,l} = R_i^{j,k,l}(t) + T_i^j(t) - I_i^{j,k,l}(t) + B_i^{j,k,l}(t) + b_i(t) + N_i^{j,k,l} + M_i^{j,k,l}(t) + \epsilon_i^{j,k,l}(t) \tag{30}$$

Assuming a small time difference and displacement, differencing two ADR measurements eliminates the effect of several of these noise terms ($T,\ I,\ B,\ N$), while the remaining terms can either be estimated directly (Δb_i), or assumed to add a small amount of additional noise (M). This leads to the following expression for the difference in ADR measurements:

$$ADR_i^{j,k,l}(t_1) - ADR_i^{j,k,l}(t_0) = \Delta R_i^{j,k,l} + \Delta b_i + \epsilon_i^{j,k,l}(t) \tag{31}$$

$$ADR_i^{j,k,l}(t_1) - ADR_i^{j,k,l}(t_0) - corr_{sv} - corr_{geo} = \Delta U^\top e_i^j(t_1) + \Delta b_1 + \epsilon_i^{j,k,l}(t) \tag{32}$$

Rewriting this equation in matrix form yields:

$$y_i^{j,k,l} = [-e_i^j(t_1); 1]^\top x + \epsilon_i^{j,k,l} \tag{33}$$

where the state vector is defined with $x = [\Delta U; \Delta b_i]$, and the accumulated doppler effect is $y_i^{j,k,l} = ADR_i^{j,k,l}(t_1) -_i^{j,k,l} (t_0) - corr_{sv} - corr_{geo}$. Multiple observations from the satellites can be stacked to form a linear system, based on which x can be estimated.

$$y = Hx + \epsilon \tag{34}$$

where y and ϵ are column vectors that include individual elements of $y_i^{j,k,l}$ and $\epsilon_i^{j,k,l}$, respectively.

4.2 Methodology

In this subsection we introduce the algorithm we developed to estimate the drone's velocity in a correlation agnostic manner based on the data received from each satellite. Our correlation agnostic fusion technique is based on the PC fusion method described in section 3, but had to be modified to enable 1D measurements to be applied to a multi-dimensional (4D in our case) state space. This modification is described in section 4.2.1 below. This approach, however, assumes a pre-existing state and covariance estimate. In section 4.2.2 we describe how the initial state and covariance estimate is generated.

4.2.1 Correlation Agnostic Satellite Data Fusion

To begin fusion, we assume that an estimate of the current velocity (μ) and its covariance (C) is available. These estimates, however, are stored in "information form" where an information vector (i) and matrix (I) are related to the current state as:

$$I = C^{-1} \tag{35}$$

$$i = I\mu \tag{36}$$

If the information was independent (had no correlation), then traditional Bayesian fusion could be performed using an information form of the Kalman filter as:

$$i^+ = i^- + H_i^\top R_i^{-1} y_i \tag{37}$$

$$I^+ = I^- + H_i^\top R_i^{-1} H_i \tag{38}$$

where both I^+ and i^+ denote the new estimate after adding the new satellite's information to the current estimate, I^- and i^-, and where y_i, R_i, and H_i are the accumulated Doppler effect, standard deviation of accumulated Doppler effect, and the geometric matrix respectively for satellite i. Note that these formulas essentially consist of taking the pre-existing information (i^-, I^-) and adding in new information from the measurements, projected into the higher dimensional space using the H_i matrices to perform the projection.

In Algorithm 1, we describe a similar procedure for adding each new satellite information to the current estimate, but rather than just "adding" information, the information is fused in a correlation-agnostic way. To enable correlation agnostic fusion techniques to be used, the current estimate's uncertainty is projected (marginalized) into the same space as the measurements as shown in the *Project current estimates step* in Algorithm 1. After *fusion* (one of the many possible fusion techniques, including regular Bayesian fusion), the new information that can be added to the current state estimate is computed. This new information is then added to the current state to generate the new state estimate.

Algorithm 1: Satellite Fusion Algorithm

```
Result: $i^+$, $I^+$
Input: $i^-$, $I^-$
Initialization: $i$, $I$
for $i$=$n$...$M$ do
    State space conversion step:
        $C^- \leftarrow (I^-)^{-1}$
        $\mu^- \leftarrow Ci^-$
    Project current estimates step:
        $\mu_a \leftarrow H_i\mu^-$
        $C_a \leftarrow H_iC^-H_i^\top$
    Compute new information step:
        $\mu_f, C_f \leftarrow \textbf{fusion}(\mu_a, C_a, y_i, R_i)$
        $C_d \leftarrow (C_f^{-1} - C_a^{-1})^{-1}$
        $\mu_d \leftarrow C_d(C_f^{-1}\mu_f - C_a^{-1}\mu_a)$
    $I^+ \leftarrow I^- + H^\top C_d^{-1}H$
    $i^+ \leftarrow i^- + H^\top C_d\mu_d$
end
```

4.2.2 Initialization

Algorithm 1, as described above, assumes that an initial I matrix is available. Furthermore, this matrix must be full-rank. To handle this the first n satellites, where n is the number of states, are fused together using a basic information filter as discussed in [10]. Therefore the initialization of I as well as i is dictated by the following equations.

$$I = \sum_{i=0}^{n} R_i^{-1} H_i^\top H_i \tag{39}$$

$$i = \sum_{i=0}^{n} h_i^\top R_i^{-1} y \tag{40}$$

With regards to the selection of satellite signals used to construct the initial I matrix, it is ensured that they are unique in order to make sure the matrix is full rank.

5 Performance Evaluation

When using GNSS satellite signals to estimate pose, we have two primary goals: (1) to accurately estimate the velocity and (2) to accurately estimate the uncertainty of the velocity estimate. Therefore, we will evaluate two corresponding metrics for each technique: Mean Squared Error (MSE) and Average Normalized Estimation Error Squared (ANEES) [11]. These metrics are reviewed in subsection 5.1, and the dataset used to test the methods is described in [7] and reviewed in subsection 5.2.

5.1 Evaluation Metric Overview

5.1.1 Mean Squared Error

The mean squared error is a commonly used metric for evaluating the accuracy of an estimate with regards to the truth. This metric is computed using the following expression:

$$MSE = \frac{1}{N} \sum_{n=1}^{N} ||\hat{x}_i - x_i||^2 \tag{41}$$

where $\hat{x}_i$ is the estimated vector and x_i is the true vector for data sample i, N is the number of data samples, and the operator $|| \cdot ||^2$ finds the squared magnitude of a vector. If the estimates were perfectly accurate, the MSE would be zero. Therefore, the smaller MSE is, the better the estimation method is performing.

5.1.2 Average Normalized Estimation Error Squared (ANEES)

ANEES, first introduced in [12, 13], is a metric utilized to evaluate the accuracy of an estimator's uncertainty. As outlined in [11], ANEES first computes λ_i for each data point which is the error in the estimate ($e_i = \hat{x}_i - x_i$) normalized by the predicted uncertainty (C_i):

$$\lambda_i = e_i^\top C_i^{-1} e_i \tag{42}$$

Using the values of λ, we compute ANEES as follows:

$$\Lambda = \frac{1}{N}\sum_{i=1}^{N}\lambda_i \tag{43}$$

If the uncertainty is accurately estimated ($C_i = \hat{C}_i$), then the values λ_i should be distributed according to a chi-squared distrbution of degree l, where l is the length of the state vector. The average value of λ, Λ, should converge to l as N becomes larger. If $\Lambda > l$, C_i understates the true uncertainty, while if $\Lambda < l$, C_i overstates the true uncertainty. While the state vector being estimated in (34) and in Algorithm 1 is of length four, we only have truth for the velocity portion and not the receiver clock information. Therefore, we will only be evaluating ANEES (and MSE) on the velocity estimate. This means that the optimal ANEES value is 3 for showing that uncertainty estimates are accurate.

5.2 Experimental Results

To collect data to test out the different fusion techniques, GNSS measurements were collected during a highway drive test near Greenville, North Carolina. The path taken is shown in Figure 4 and went through a suburban/rural area, which is similar to the open sky condition most of the time. The average speed of the test vehicle was 45 mi/hr, or 20 m/s.

During the drive test, satellite position and clock, and satellite ephemeris (which were used to compute y, H, and R) were recorded with a NovAtel OEM® 6 receiver for GPS, GLONASS, GALILEO and BEIDOU. To enable accurate truth estimation, the GNSS antenna was shared with a NovAtel SPAN® A1 GPS/IMU unit. In this case, the truth reference was obtained from post-processed GPS-IMU velocity using NovAtel Waypoint® Inertial Explorer software. The standard deviation reported for the "truth" reference is [0.004 0.004 0.003] m/s in east, north, and up respectively.

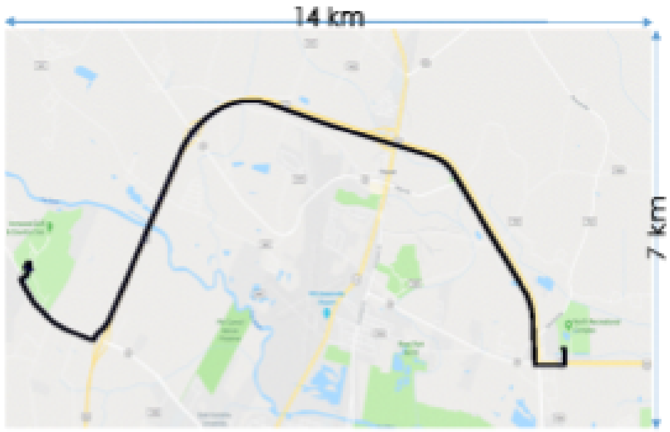


Figure 4: Ground trajectory of highway drive test in Greenville, NC. Map from google.com

Using Algorithm 1, four different fusion techniques were evaluated on this data set. The fusion techniques evaluated (with their respective **fusion** function) are:

- **Bayesian (naive)**. This assumes the new data is independent from previous data. **fusion**(μ_a, C_a, y_i, R_i) is expressed as:

$$C_f^{-1} = R_i^{-1} + C_a^{-1} \tag{44}$$

$$\mu_f = C_f(R_i^{-1} y_i + C_a^{-1} \mu_a) \tag{45}$$

- **Covariance Intersection**. A correlation agnostic technique. **fusion**(μ_a, C_a, y_i, R_i) expressed as:

$$C_f, \mu_f = \begin{cases} R_i, y_i & : R_i < C_a \\ C_a, \mu_a & : \text{otherwise} \end{cases} \tag{46}$$

- **Ellipsoidal Intersection**. Another (prior) correlation agnostic technique **fusion**(μ_a, C_a, y_i, R_i) expressed as:

$$C_c = \epsilon + \begin{cases} R_i & : R_i > C_a \\ C_a & : \text{otherwise} \end{cases} \tag{47}$$

$$C_f^{-1} = C_a^{-1} + R_i^{-1} - C_c^{-1} \tag{48}$$

$$\mu_f = (C_a^{-1} + R_i^{-1} - 2C_c^{-1})^{-1} \left((C_a^{-1} - C_c^{-1})\mu_a + (R_i^{-1} - C_c^{-1}) y_i\right) \tag{49}$$

- **Probabilisticly Conservative**. Our novel method for correlation-agnostic fusion, depending on both the covariances and the means. To compute **fusion**(μ_a, C_a, y_i, R_i), we start with the C_c computed for EI and find the minimal variance increase that makes the Mahalanobis distance (**??**) drop below the threshold for making a false positive less than 5% likely.

Each of these techniques is run on the same data, and the metrics outlined in section 5.1 computed based on their results. The results of this experiment can be seen in Table 2. To provide a single metric combining both MSE and ANEES, the *distance from optimal* ($D_{optimal}$) is computed by finding the squared distance between the computed MSE and 0 and the computed ANEES and 3. The results are also shown in Figure 5 in a more graphical format, where the y-axis is the MSE and the x-axis is the ANEES.

Table 2: GNSS dataset velocity/uncertainty estimation performance for each fusion technique

	$MSE(*10^{-4})$	$ANEES$	$D_{optimal}$
Bayesian (naive)	5.97	9.578	43.268
Ellipsoidal Intersection	149.13	2.888	0.0128
Covariance Intersection	6.64	1.817	1.3999
Probablistic Conservative ($\alpha = 0.05$)	6.45	3.088	0.0076

Several important things can be observed from the presented results.

1. While the Bayesian technique gives the best MSE results, this technique significantly under-estimates the uncertainty in its results, demonstrating the need for correlation-agnostic fusion results

2. Of the three correlation-agnostic fusion techniques, the novel PC technique is most accurate in both MSE and ANEES, leading to the smallest $D_{optimal}$.

3. Both CI and PC fusion have MSE results that are very close to the Bayesian technique. EI fusion, while significantly larger, still has a very small MSE.

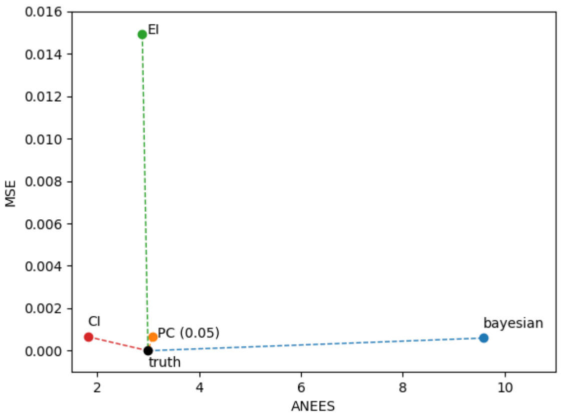


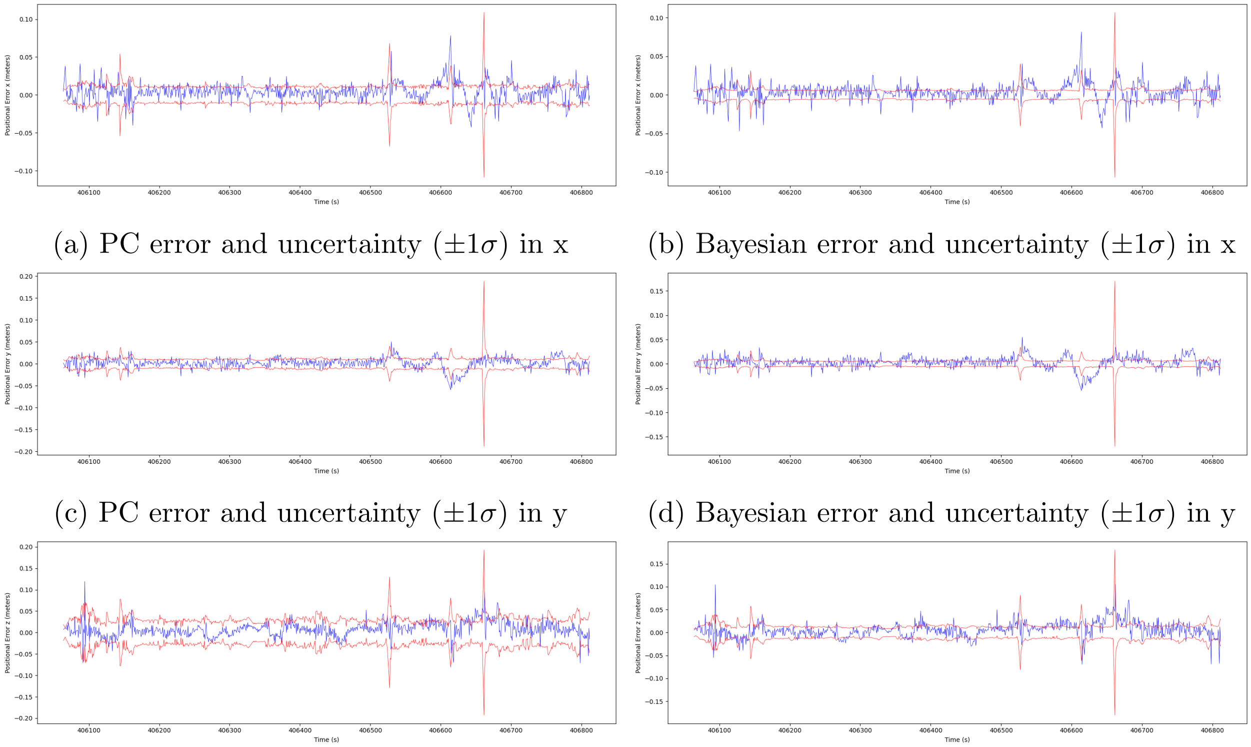
Figure 5: Visual plotting of various fusion technique's evaluation metrics in comparison to desired point.

To demonstrate the effect of the large ANEES observed by the Bayesian fusion method, Figure 6 has three plots of the errors, one for each axis, and the corresponding 1-sigma bounds over time. In each plot, the blue line is the difference between the estimated value (in that dimension) and the truth, while the two red lines are the computed (1-sigma) standard deviations for that axis. Although both PC as well as Bayesian have similar error (blue line) note that the computed covariances for PC bound the errors much more accurately than the covariances computed by Bayesian technique.

6 Conclusion

This paper has outlined the application of correlation-agnostic fusion to GNSS-based pose estimation. This paper introduces a novel correlation-agnostic fusion technique and applies this technique to GNSS-based velocity estimation. In a GNSS drone localization solution, failure to account for possible correlation among satellites could result in severe underestimation of true errors which is dangerous for drone navigation in cluttered environments. Based on datasets collected in real-world scenarios, we have demonstrated that by deploying correlation-agnostic fusion techniques we can provide equally accurate velocity estimates while also accurately predicting the uncertainty in the estimates.

In the future, we hope to find closed form solutions to the PC fusion optimization problem. Currently, the fusion is based on a constrained optimization framework, which may not be feasible computationally for all applications (particularly higher-dimesional problems). If a closed-form solution were to be discovered then the computational requirements issue would be minimized.



(a) PC error and uncertainty ($\pm 1\sigma$) in x

(b) Bayesian error and uncertainty ($\pm 1\sigma$) in x

(c) PC error and uncertainty ($\pm 1\sigma$) in y

(d) Bayesian error and uncertainty ($\pm 1\sigma$) in y

(e) PC error and uncertainty ($\pm 1\sigma$) in z

(f) Bayesian error and uncertainty ($\pm 1\sigma$) in z

Figure 6: Uncertainty estimate comparison between PC and Bayesian in each dimension.

The big picture hope for this research is that by enabling more effective control of drones in a wider-expanse of environments the full potential for autonomous drones with regards to applications will be achieved. The contributions of this paper provide a fundamental solution for drone localization that can be applied to a multitude of real-world challenges.

References

[1] O. a recent assignment to Moscow, "Getting lost near the kremlin? russia could be 'gps spoofing'." [Online]. Available: https://money.cnn.com/2016/12/02/technology/kremlin-gps-signals/

[2] A. Patrik, G. Utama, A. A. S. Gunawan, A. Chowanda, J. S. Suroso, R. Shofiyanti, and W. Budiharto, "Gnss-based navigation systems of autonomous drone for delivering items," *Journal of Big Data*, vol. 6, no. 1, 2019.

[3] G. Zhang and L.-T. Hsu, "A new path planning algorithm using a gnss localization error map for uavs in an urban area," *Journal of Intelligent and Robotic Systems*, vol. 94, no. 1, p. 219–235, 2018.

[4] F. Li, Q. Li, and W. Qi, "Combining carrier-phase and doppler observations for precise velocity estimation with a stand-alone gps receiver," *Electromechanical Control Technology and Transportation*, p. 133–137, 2017.

[5] "General decentralized data fusion with covariance intersection (ci)," *Multisensor Data Fusion*, p. 269–294, 2001.

[6] J. Sijs and M. Lazar, "State fusion with unknown correlation: Ellipsoidal intersection," *Automatica*, vol. 48, no. 8, p. 1874–1878, 2012.

[7] Z. Zhu, E. Vinande, and M. U. D. Haag, "Multi-constellation time-differenced carrier phase solution with protection from multiple failures," *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2018.

[8] G. Panel, "Phase ii of the gnss evolutionary architecture study," 2010.

[9] J. Cohenour, *Global Positioning System Clock and Orbit Statistics and Precise Point Positioning.* Ohio University, 2009. [Online]. Available: https://books.google.com/books?id=Ohi4jwEACAAJ

[10] P. Zarchan, H. Musoff, A. I. of Aeronautics, and Astronautics, *Fundamentals of Kalman Filtering: A Practical Approach*, ser. Progress in astronautics and aeronautics. American Institute of Aeronautics and Astronautics, Incorporated, 2000. [Online]. Available: https://books.google.com/books?id=AQxRAAAAMAAJ

[11] C. N. Taylor and S. Lubold, "Verifying the predicted uncertainty of bayesian estimators," *Geospatial Informatics, Motion Imagery, and Network Analytics VIII*, 2018.

[12] Y. Bar-Shalom and X.-R. Li, “Estimation and tracking- principles, techniques, and software,” *Norwood, MA: Artech House, Inc, 1993.*, 1993.

[13] D. Lerro and Y. Bar-Shalom, “Tracking with debiased consistent converted measurements versus ekf,” *IEEE transactions on aerospace and electronic systems*, vol. 29, no. 3, pp. 1015–1022, 1993.