



G2A Localization: Aerial Vehicles Localization Using a Ground Crowdsourced Network

Hazem Sallouha¹, Alessandro Chiumento², and Sofie Pollin¹

¹ Department of Electrical Engineering – TELEMIC, KU Leuven, Belgium

² CONNECT Centre, Trinity College Dublin, Ireland

*Email: hazem.sallouha@esat.kuleuven.be

Abstract

In this paper, we address the ground-to-air (G2A) localization problem using a crowdsourced network with a mix of synchronized and unsynchronized receivers. First, we use a dynamic model to represent the offset and the skew of the unsynchronized receivers. This model is then used with a Kalman filter (KF) to compensate for the drifts of the unsynchronized receivers' clocks. Subsequently, the location of the aerial vehicle (AV) is estimated using another KF with the multilateration (MLAT) method and the dynamic model of the AV. We demonstrate the performance advantages of our method through a dataset collected by the OpenSky network. Our results show that the proposed dual KF method decreases the average localization error by orders of magnitude compared with a solo multilateration method. In particular, the proposed method brings the average localization error from tens of kilometers down to hundreds of meters, based on the considered dataset.

Keywords— Localization, multilateration, synchronization, Kalman filter, dynamic clock model, TDoA, unmanned aerial vehicle (UAV)

1 Introduction

Unmanned aerial vehicles (UAVs) are projected to outgrow manned aerial vehicles (MAVs) by several orders of magnitude over the next 20 years [1]. With UAVs entering the civil airspace, air traffic control (ATC) must expand to handle their coexistence with MAVs. However, many technical issues arise in current ATC systems such as sparse coverage and vulnerability to spoofing devices [2,3]. Moreover, unlike the MAVs safety system, which includes onboard navigation aids, a pilot to intervene, and ground ATC, UAVs safety systems depend primarily on ground ATC due to size and power limitations. To this end, crowdsourced ATC networks have gained considerable focus over the past decade [3–5].

Crowdsourced ATC uses widely distributed off-the-shelf sensors to record broadcast messages from aerial vehicles (AVs) [4]. Capturing AV broadcasts at multiple ground receivers, allows one to localize the AV using multilateration (MLAT) [6]. Conventional MLAT methods are based on received signal

*This work is part of a project that has received funding from the SESAR Joint Undertaking (JU) under grant agreement No. 763702. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the SESAR JU members other than the Union.

strength (RSS), time of arrival (ToA), or time differences of arrival (TDoA). Both RSS and ToA based methods require knowledge about some of the transmitter's characteristics, such as the transmit power and the local clock [7–9]. Such information is not necessary for TDoA-based methods, making it a favorable choice with MLAT [6,10]. However, localization using MLAT with TDoA requires the clocks of all receivers to be synchronized [9]. Unfortunately, this is not the case in the current crowdsourced ATC networks, as they include many unsynchronized receivers [11].

In this work, we address the ground-to-air (G2A) localization problem using a crowdsourced network with a mix of synchronized and unsynchronized sensors. Firstly, we model the clock's dynamic behavior of the unsynchronized receiver. This model is then used with a Kalman filter (KF) to compensate for the drift of the unsynchronized clocks. Subsequently, the location of the AV is estimated using another KF with the TDoA-based MLAT and the dynamic model of the AV.

The rest of the paper is organized as follows. In Section 2, we introduce system assumptions and the TDoA method. Section 3 presents the dual KF method with both the dynamic clock model and the AV dynamic model used. Subsequently, we present our experimental results in Section 4. Finally, the conclusion is presented in Section 5.

Notation: Italic letters, simple bold letters, and capital bold letters represent scalars, vectors, and matrices, respectively. We use (a_1, a_2, \dots) to represent a sequence and $[a_1, a_2, \dots]^T$ to represent a column vector, with $[.]^T$ being the transpose operator. We use \hat{x} to denote the estimate of x and x^- to denote the *a priori* of x .

2 System Model

Consider a ground-based crowdsourced ATC network deployed to localize and track aerial vehicles, as illustrated in Fig. 1. We assume that AVs send periodic radio frequency (RF) signals which are received by N receivers, denoted by Rx_1, Rx_2, \dots, Rx_N . The receivers register the signal's ToAs and, subsequently, forward them to a centralized station where the location of the corresponding AV is estimated. Consider MLAT with TDoA, the TDoA associated with the i -th receiver, Rx_i and the j -th receiver, Rx_j , is $t_j - t_i$, where t_i and t_j are the ToAs at Rx_i and Rx_j . Accordingly, one can define the difference in distance d_{ij} as

$$\begin{aligned} d_{ij} &:= d_i - d_j \\ &= (t_i - t_o)c - (t_j - t_o)c = (t_i - t_j)c, \end{aligned} \quad (1)$$

where c is the speed of light, t_o is the AV clock time, and d_i and d_j are the distances from the AV to Rx_i and Rx_j , respectively. The distance from the AV to the i -th receiver is expressed as

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2}, \quad (2)$$

where (x, y, z) is the AV position and (x_i, y_i, z_i) is the position of Rx_i with $i = 1, 2, \dots, N$. Without loss of generality, the origin of the Cartesian coordinate system is set at Rx_1 , i.e., $(x_1, y_1, z_1) = (0, 0, 0)$. Consequently, using (2), d_1 can be written as

$$d_1 = \sqrt{x^2 + y^2 + z^2}. \quad (3)$$

Now, following [9,12], the position of the AV, defined as $\tilde{\mathbf{q}} := [\tilde{x}, \tilde{y}, \tilde{z}]^T$, can be calculated from

$$\tilde{\mathbf{q}} = \left(\mathbf{B}^T \mathbf{B} \right)^{-1} \mathbf{B}^T (d_1 \mathbf{A} + \mathbf{D}). \quad (4)$$

In (4), \mathbf{B} , \mathbf{A} , and \mathbf{D} are matrices expressed as

$$\mathbf{B} = \begin{bmatrix} x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ \vdots & \vdots & \vdots \\ x_N & y_N & z_N \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} -d_{21} \\ -d_{31} \\ \vdots \\ -d_{N1} \end{bmatrix}, \quad \mathbf{D} = \frac{1}{2} \begin{bmatrix} D_2^2 - d_{21}^2 \\ D_3^2 - d_{31}^2 \\ \vdots \\ D_N^2 - d_{N1}^2 \end{bmatrix}$$

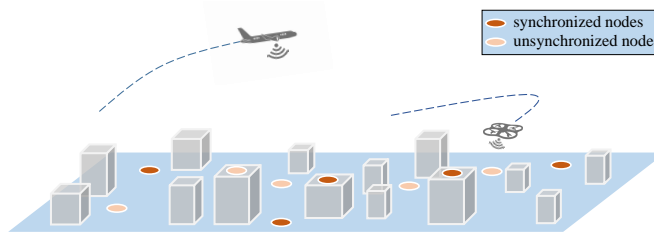


Figure 1: Localizing and tracking of aerial vehicles using a ground-based crowdsourced network.

where

$$D_i^2 = x_i^2 + y_i^2 + z_i^2. \quad (5)$$

Equation (4) contains d_1 , which is unknown. Substituting (4) into (3) yields a quadratic equation in d_1 . Solving for d_1 and substituting the positive root back into (4) gives the final solution for $\tilde{\mathbf{q}}$. In the case of two positive solutions for d_1 , we choose the one that lies in the domain of interest, e.g., the one that presents a positive altitude [12]. To obtain an accurate position, all receivers need to be synchronized. However, in crowdsourced ATC networks, receivers are of two categories: synchronized receivers, with clocks that are constantly GPS synchronized, and unsynchronized receivers, with clocks that are subject to drifts. Therefore, we introduce our dual KF method, which is detailed in the following section.

3 Dual Kalman Filter Method

In this section, we introduce our dual KF method. The first KF uses a dynamic clock model to compensate for the clock drifts of the unsynchronized receivers. The second one uses the AV dynamic model to localize and track the targeted AV.

3.1 Receiver's Clock Dynamic Model

When unsynchronized receivers are involved in the localization process, TDoA produces a localization error that is proportional to the clock drifts. In order to compensate for the clock drifts in unsynchronized receivers, we use a dynamic clock model [13,14]. The time reported by a clock is characterized by an offset $\theta_{(k)}$ and the skew $\alpha_{(k)}$. The skew is the offset's rate of change, i.e., the slope of the offset. Accordingly, the clock dynamic model can be written as

$$\theta_{(k)} = \theta_{(k-1)} + \alpha_{(k-1)}\tau_{(k-1)} + \eta_{\theta(k)} \quad (6)$$

where $\tau_{(k)}$ the sampling period at the k -th sample, and $\eta_{(k)}$ is a zero-mean normally distributed noise. In order to use the model in (6) to synchronize clocks, one needs to estimate $\alpha_{(k)}$. The time-varying skew can be modeled using an auto-regressive (AR) process. Consider an AR model with order M , the skew can be written as [14]

$$\alpha_{(k)} = \sum_{i=1}^M b_i \alpha_{(k-i)} + \eta_{\alpha(k)} \quad (7)$$

where b_i 's are the AR coefficients, and $\eta_{\alpha(k)}$ is a zero-mean Gaussian noise. In order to estimate the coefficients b_i 's, training data is needed. This training data can be obtained from a known AV.

Ideally, a clock model can estimate the clock behavior accurately. However, clocks are affected by other factors such as temperature and humidity. These factors have a nondeterministic influence on clock behavior, which prevents an exact clock model from tracking clock behaviors precisely. Therefore, we propose a KF to track the time-varying skew and offset using the model presented in (6).

3.2 Aerial Vehicles Dynamic Model

The movement of the AV is modeled as a dynamic system. Given the periodic transmissions from an AV, the state vector at the k -th transmission is written as

$$\mathbf{s}(k) = [x(k), y(k), z(k), \dot{x}(k), \dot{y}(k), \dot{z}(k)]^T \in \mathbb{R}^6, \quad (8)$$

where the sequence $(x(k), y(k), z(k))$ denotes the position of the AV, and $(\dot{x}(k), \dot{y}(k), \dot{z}(k))$ represents its velocity. Moreover, the input of this dynamic system is represented by the AV's acceleration which, at the k -th transmission, is given by

$$\mathbf{u}(k) = [\ddot{x}(k), \ddot{y}(k), \ddot{z}(k)]^T \in \mathbb{R}^3, \quad (9)$$

where the sequence $(\ddot{x}(k), \ddot{y}(k), \ddot{z}(k))$ represents the acceleration of the AV. Now, assuming four receivers are available, one can use (4) to calculate the AV's position. Consequently, we express the k -th measurement vector $\tilde{\mathbf{q}}(k)$, which is the calculated position from (4), as

$$\tilde{\mathbf{q}}(k) := [\tilde{x}(k), \tilde{y}(k), \tilde{z}(k)]^T \in \mathbb{R}^3. \quad (10)$$

Finally, the dynamic system can be written as

$$\mathbf{s}(k) = \mathbf{\Phi}_{(k-1)} \mathbf{s}(k-1) + \mathbf{\beta}_{(k-1)} \mathbf{u}(k-1) + \mathbf{w}(k-1), \quad (11)$$

$$\tilde{\mathbf{q}}(k) = \mathbf{H} \mathbf{s}(k) + \mathbf{v}(k), \quad (12)$$

where $\mathbf{\Phi}_{(k)} \in \mathbb{R}^{6 \times 6}$ is the state transition matrix, $\mathbf{\beta}_{(k)} \in \mathbb{R}^{6 \times 3}$ is the input matrix, $\mathbf{u}(k) \in \mathbb{R}^3$ is the model input vector, and $\mathbf{H} \in \mathbb{R}^{3 \times 6}$ represents the measurement matrix. Moreover, in (11) and (12), vectors $\mathbf{w}(k)$ and $\mathbf{v}(k)$ represent the model and measurement noise, respectively. They are assumed to be independent and normally distributed, i.e.,

$$\begin{aligned} \mathbf{w}(k) &\sim \mathcal{N}(0, \mathbf{Q}_{(k)}) \text{ with } \mathbf{Q}_{(k)} = \mathbb{E}[\mathbf{w}(k)\mathbf{w}(k)^T], \\ \mathbf{v}(k) &\sim \mathcal{N}(0, \mathbf{R}_{(k)}) \text{ with } \mathbf{R}_{(k)} = \mathbb{E}[\mathbf{v}(k)\mathbf{v}(k)^T], \end{aligned}$$

where $\mathbb{E}[\cdot]$ denotes the expected value. Now, the KF formulation including the predictions and the updates is expressed as [15]

Predict:

$$\hat{\mathbf{s}}_{(k)}^- = \mathbf{\Phi}_{(k-1)} \hat{\mathbf{s}}_{(k-1)} + \mathbf{\beta}_{(k-1)} \mathbf{u}_{(k-1)} \quad (13)$$

$$\mathbf{P}_{(k)}^- = \mathbf{\Phi}_{(k-1)} \mathbf{P}_{(k-1)} \mathbf{\Phi}_{(k-1)}^T + \mathbf{Q}_{(k-1)} \quad (14)$$

Update:

$$\mathbf{G}_{(k)} = \mathbf{P}_{(k)}^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_{(k)}^- \mathbf{H}^T + \mathbf{R}_{(k)})^{-1} \quad (15)$$

$$\hat{\mathbf{s}}_{(k)} = \hat{\mathbf{s}}_{(k)}^- + \mathbf{G}_{(k)} (\tilde{\mathbf{q}}_{(k)} - \mathbf{H} \hat{\mathbf{s}}_{(k)}^-) \quad (16)$$

$$\mathbf{P}_{(k)} = (\mathbf{I} - \mathbf{G}_{(k)} \mathbf{H}) \mathbf{P}_{(k)}^- \quad (17)$$

where \mathbf{I} is an identity matrix, $\mathbf{P}_{(k)} \in \mathbb{R}^{6 \times 6}$ is the state error covariance matrix, and $\mathbf{G}_{(k)} \in \mathbb{R}^{6 \times 3}$ is the KF gain. The KF presented in equations (13)-(17) is able to overcome the Gaussian noise with the estimated locations.

A summary of the dual KF method is presented in Algorithm 1. The algorithm works as follows. With each broadcast from AVs, the algorithm collects the ToA from all available receivers (*currentReceivers*). Subsequently, it checks the type of each receiver. For each unsynchronized receiver (i.e., type == unsynced), it runs the first KF (KF₁) with the corresponding clock model to compensate for the clock offset. The ToAs are used to calculate TDOAs, which are subsequently used to estimate the location using MLAT. The estimated location is then processed using a second KF (KF₂) with the AV dynamic model to obtain the final location estimate.

Algorithm 1 Localizing and tracking of AVs using dual KF method

```

1: Input: AV, AVDynamicModel, clockDynamicModel, receivers:  $Rx_1, Rx_2, \dots$ 
2: Output: Estimated location of the AV ( $locationEstimate$ )

3: while True do
4:    $Msg \leftarrow newBroadcastMessage(AV)$ 
5:    $currentReceivers \leftarrow receivers.hasReceived(Msg)$ 
6:    $N \leftarrow count(currentReceivers)$ 
7:    $ToAs \leftarrow currentReceivers.getToAs(Msg)$ 
8:   for  $\forall currentReceivers$  where type == unsynced do
9:      $ToAs \leftarrow KF_1(ToAs, clockDynamicModel)$  # using (6), (7)
10:  end for
11:   $TDoAs \leftarrow calculateTDoAs(ToAs)$ 
12:  if  $N \geq 4$  then
13:     $locationEstimate \leftarrow MLAT(TDoAs, currentReceivers.location)$  # using (4)
14:     $locationEstimate \leftarrow KF_2(locationEstimate, AVDynamicModel)$  # using (13)-(17)
15:  end if
16: end while

```

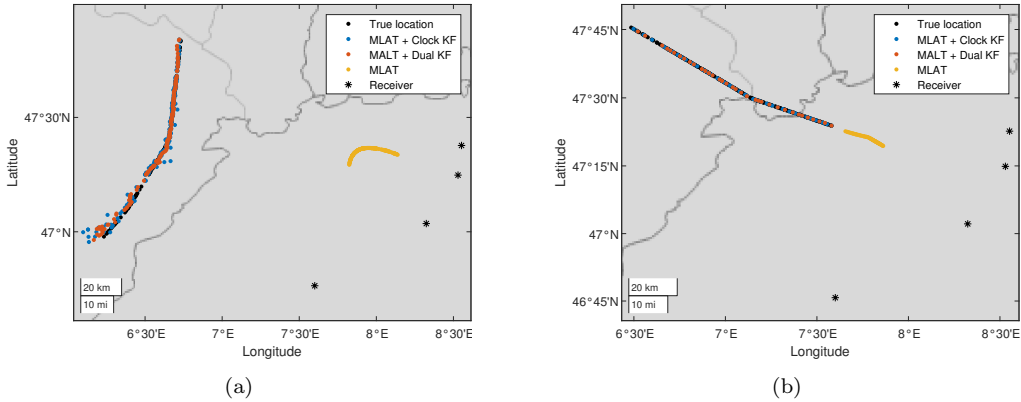


Figure 2: A Comparison between the proposed dual KF and other tracking methods. (a) AV's trajectory is parallel to the receivers' positions. (b) AV's trajectory is perpendicular to the receivers' positions.

4 Experimental Results

In this section, we assess the performance of the proposed method when applied on a dataset collected by the OpenSky Network¹. The dataset consists of ADS-B messages sent from aircraft and received using distributed synchronized and unsynchronized receivers. Although the dataset is based on MAV, the performance analysis is also valid for other AV. This due to the fact that the line of sight channels are rather dominant at altitudes above 200 m [7], leading to a steady TDoA performance above this altitude. The dataset includes the locations of the receivers, the ToA at reach receiver, the aircraft ID, and the true position of the aircraft. The true position is used to calculate the localization error. Considering a 3D Cartesian coordinate system, we defined the localization error as

$$\epsilon = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2 + ((z - \hat{z})/10)^2}, \quad (18)$$

¹<https://opensky-network.org>

Table 1: Average localization error in Figure 2a

Method	Average localization error [m]
MLAT	104×10^3
MLAT + KF with clock model	1300
Dual KF method	800

Table 2: Average localization error in Figure 2b

Method	Average localization error [m]
MLAT	60×10^3
MLAT + KF with clock model	200
Dual KF method	140

where (x, y, z) is the true position of the AV and $(\hat{x}, \hat{y}, \hat{z})$ is the estimated one.

Figure 2a presents the performance of the dual KF method in comparison with MLAT method and MLAT with a single KF for the clock model. In this scenario, we have 4 receivers, one of which is unsynchronized. The average localization errors, over time, for all three methods are summarized in Table 1. As shown in the table, using the clock model with a KF to compensate for the drifts of the clock decreases the error by orders of magnitude. Moreover, applying the second KF with the AV dynamic model further decreases the localization error from an average of 1300 m to 800 m.

The performance of MLAT based on TDoA method is greatly influenced by the target’s position relative to the receivers’ positions [4,9]. In Figure 2b, we present another scenario where the AV’s trajectory is perpendicular to the locations of the receivers. As shown in Table 2, better performance is achieved with the same set of receivers and the same clock model. In particular, using the dual KF method, an average localization error of 140 m is achieved.

5 Conclusion

The localization of AVs using a crowdsourced network has been investigated. Particularly, a crowdsourced network with a mix of synchronized and unsynchronized receivers has been considered. We proposed a dual KF method to localization and track AVs. The first KF is used to compensate for the offsets of the unsynchronized clocks. Subsequently, we used a MLAT method to estimate the location of the targeted AV. The second KF is introduced to further improve the estimated location while considering the AV dynamic model. The proposed method has been validated using an OpenSky dataset, where it showed a notable improvement in the localization accuracy.

References

- [1] U. D. of Transportation, “Unmanned aircraft system (UAS) service demand 2015-2035, Tech. Rep., 2013.,” [Online]. Available: <https://fas.org/irp/program/collect/service.pdf>, 2013.
- [2] J. D. Yoo and S. Devasia, “On-demand conflict resolution procedures for air-traffic intersections,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1538–1549, 2014.
- [3] K. Jansen, M. Schäfer, D. Moser, V. Lenders, C. Pöpper, and J. Schmitt, “Crowd-gps-sec: Leveraging crowdsourcing to detect and localize gps spoofing attacks,” in *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 1018–1031, IEEE, 2018.

- [4] M. Strohmeier, I. Martinovic, and V. Lenders, "A k-nn-based localization approach for crowd-sourced air traffic communication networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 3, pp. 1519–1529, 2018.
- [5] G. Liu, R. Zhang, C. Wang, and L. Liu, "Synchronization-free gps spoofing detection with crowd-sourced air traffic control data," in *2019 20th IEEE International Conference on Mobile Data Management (MDM)*, pp. 260–268, IEEE, 2019.
- [6] I. A. Mantilla-Gaviria, M. Leonardi, G. Galati, and J. V. Balbastre-Tejedor, "Localization algorithms for multilateration (mlat) systems in airport surface surveillance," *Signal, Image and Video Processing*, vol. 9, no. 7, pp. 1549–1558, 2015.
- [7] M. M. Azari, H. Sallouha, A. Chiumento, S. Rajendran, E. Vinogradov, and S. Pollin, "Key technologies and system trade-offs for detection and localization of amateur drones," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 51–57, 2018.
- [8] H. Sallouha, M. M. Azari, and S. Pollin, "Energy-constrained uav trajectory design for ground node localization," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, IEEE, 2018.
- [9] A. H. Sayed, A. Tarighat, and N. Khajehnouri, "Network-based wireless location: challenges faced in developing techniques for accurate wireless location information," *IEEE Signal Processing Magazine*, vol. 22, pp. 24–40, July 2005.
- [10] H. Seo, H. Kim, J. Kang, I. Jeong, W. Ahn, and S. Kim, "3d moving target tracking with measurement fusion of tdoa/fdoa/aoa," *ICT Express*, vol. 5, no. 2, pp. 115–119, 2019.
- [11] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm, "Bringing up opensky: A large-scale ads-b sensor network for research," in *Proceedings of the 13th international symposium on Information processing in sensor networks*, pp. 83–94, IEEE Press, 2014.
- [12] H. Schau and A. Robinson, "Passive source localization employing intersecting spherical surfaces from time-of-arrival differences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 8, pp. 1223–1225, 1987.
- [13] W. Masood, J. F. Schmidt, G. Brandner, and C. Bettstetter, "Disty: Dynamic stochastic time synchronization for wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1421–1429, 2017.
- [14] H. Kim, X. Ma, and B. R. Hamilton, "Tracking low-precision clocks with time-varying drifts using kalman filtering," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 1, pp. 257–270, 2012.
- [15] G. Welch and G. Bishop, "An introduction to the kalman filter," *Dept. Comput. Sci., Univ. North Carolina, Chapel Hill, NC, USA, Tech. Rep.*, 2006.