A Novel Statistically-Aided Learning Framework for Precise Localization of UAVs

Akash Kumar Mandal[a], Jun-Bae Seo[b], Swades De[a], Ajay K. Poddar[c], and Ulrich Rohde[d]
[a]Department of Electrical Engineering and Bharti School of Telecommunication, IIT Delhi, New Delhi, India
[b]Department of Information and Communication Engineering, Gyeongsang National University, South Korea
[c]Synergy Microwave Corporation, Paterson, NJ, USA
[d]Faculty of Informatics, Federal University of the Joint Forces, Germany

***Abstract*—The accuracy of localization using global positioning system (GPS) data plays a key role in reliable positioning and control of unmanned aerial vehicles (UAVs). This paper proposes a novel statistically-aided earning-based localization approach, called filtered neural network (FNN) for high-precision localization of UAVs. The proposed FNN framework utilizes an entropy adaptive Kalman filter to fine-tune the inputs to a recurrent neural network, which works in a loop with the filter to generate subsequent robust position estimates. The proposed framework outperforms the state-of-the-art techniques with an nRMSE of $\approx 10^{-6}$, $\approx 97\%$ reduced estimation delay, $\approx 73\%$ reduced modeling time, ≤ 100 lag samples for FNN training, and only 4-6 overall model retraining instances per flight trajectory. The results are verified over a wide range of mean GPS noise power.**

***Index Terms*—Entropy adaptive Kalman filter, filtered neural network (FNN), long short-term memory, UAV localization**

I. Introduction

Fast growing interest of precise navigation, reliable communication, and surveillance using aerial methods have increased the demand for location-intensive services [1], [2]. Global positioning system (GPS) has gained widespread importance for localization as it provides absolute positions to the receiving units unlike the source localization systems. However, the accuracy of GPS is affected by measurement techniques, wireless channel characteristics, and position estimation approaches. These factors assume significance when the GPS receiver is highly mobile, as in unmanned aerial vehicles (UAVs). Furthermore, the received GPS signals can be severely impaired by the involved plasma channel, affecting three-dimensional localization. Inaccurate localization affects aerial system performance in many ways, e.g., in object monitoring integrity, actuation accuracy, multi-UAV coordination, etc. [3]. Therefore, developing robust navigation filter to fine-tune the UAV location estimates from the noisy GPS data is of high interest [4].

A. Literature Review and Motivation

The work thus far in literature can be put into three sets. The first set [5], [6] uses statistical tools, such as adaptive Kalman filter (AKF), extended Kalman filter, correntropy Kalman filter (CKF) etc., in achieving UAV localization. Though these methods provide some order of reliability in generating UAV localization, none of them provide highly accurate positioning solutions. Further, the time complexity of these tools is not suitable in most practical deployment scenarios [7]. The work in [8] alleviates this issue by utilizing improved Kalman variants, and tag-based visual localization aids in generating rapid and robust position estimates. However, the signal from a GPS transmitter is heavily impaired by the involved plasma channel, imparting a non-Gaussian noise characteristic to the received data [9], [10]. Therefore, traditional filtering and regression approaches fail to operate in a time-constricted framework [11].

The second set [12], [13] uses learning-based approaches in generating UAV position estimates. The work in [14] utilizes genetic algorithms for UAV localization and clustering in a densely deployed scenario. Though these approaches are relatively faster in generating subsequent position estimates, they do not consider statistical properties of the noise. This makes the UAV localization less accurate, which is particularly detrimental in UAV swarm setup. The work in [15] tried to overcome such aspects using a prior knowledge of UAV trajectory. However, such approaches fail when the trajectory of the UAV is dynamically adaptive. Furthermore, because of no statistical aid in the learning module, none of these approaches adapt well with the plasma channel induced aberrations [16].

The final set [17], [18] provide UAV positioning strategies in GPS-denied environments. The work in [19] utilizes multi-UAV coordination for precised UAV localization in extreme conditions, such as under energy constraint. Though these strategies show acceptable performance in scenarios requiring less precision, they are not sufficient for UAV-swarm networks. The research in [20] proposes a maximum likelihood Kalman filter-aided learning for UAV localization in a multi-UAV scenario. However, the inability of the maximum likelihood framework to handle non-Gaussian noise profiles renders this strategy unsuitable in presence of heavy-tailed noise [21].

As noted above, there is a need for robust, fast, and low-cost UAV localization framework that is capable of generating accurate position estimates using the noisy estimates from the basic receiver module. Such a navigational filter becomes especially important as the UAV mobility increases, with the measurements impaired by non-Gaussian noise profiles of the plasma channel. To this end, this paper proposes a statistically-aided learning-based framework, called filtered neural network (FNN), for fast and reliable estimation of UAV position from the GPS data that is impaired by non-Gaussian channel noise.

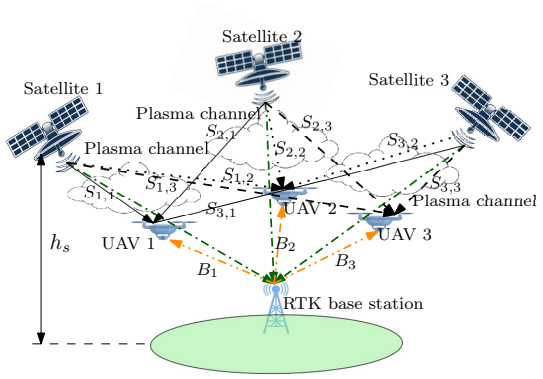


Fig. 1: System model for localization of UAV.

B. Contributions and Significance

The major contributions are as follows: 1) An entropy adaptive Kalman filter (EAKF) is proposed to generate UAV position estimates using data transmission through channels with non-Gaussian noise profiles. 2) A novel FNN algorithm is introduced that uses EAKF and long short-term memory (LSTM) network interaction. 3) Computational complexity of the FNN algorithm is noted to be lesser than that of the state-of-the art navigational filters. 4) Finally, an energy efficient retraining strategy is suggested for the FNN algorithm.

Simulation results under noisy GPS measurements demonstrate a considerably improved localization with the proposed FNN framework compared to the existing pure statistical and pure learning-based frameworks, with a normalized root mean square error (nRMSE) of $\approx 10^{-6}$, in 60 training epochs with a retraining frequency of 4, 97.83% faster estimation, and 72.81% reduced modeling time. *The proposed framework can also be utilized in positioning and control of unmanned ground vehicles (UGVs), military surveillance, marine applications, etc.*

Section II outlines the system model; Section III presents the proposed FNN framework; Section IV discusses the performance results, followed by the conclusion in Section V.

II. System Model

Fig. 1 demonstrates the system model envisaged in formulating the proposed FNN-based UAV localization algorithm. A multi-UAV scenario, pertaining to UAV swarm architecture is presented. Signal from the i-th satellite to the j-th UAV is marked as $S_{i,j}$, while the error signal from the real-time kinematic (RTK) receiver to the j-th UAV is denoted as B_j. The RTK generates error signals B_j $\forall j$ by comparing the real-time estimates of its position using the signals from multiple satellites and its precise location calculated through various high-precision measurement techniques. Information of the error signal B_j provides a knowledge of the process noise.

Since, these satellites are at least at a height $h_s = 160$ km, these signals traverse through a plasma sheath of thickness $t_p \approx h_s - 80$ km, contributing to a non-Gaussian noise to the measurements generated using this RTK-GNSS assembly. Therefore, to provide reliable real-time localization of the UAV, a FNN-based UAV localization algorithm is implemented at the UAV in a data-aided approach. The proposed algorithm is implemented on a quad-core 64-bit BCM2837 ARM Cortex-A53 SoC processor running at a clock frequency of 1.2 GHz.

III. FNN-Based UAV Localization

This section explains the FNN algorithm utilized in achieving high-precision localization. Let the measurement estimated from the GPS readings at time k be $\mathbf{x}(k) = \tilde{\mathbf{x}}(k) + \boldsymbol{\epsilon}(k)$, where $\tilde{\mathbf{x}}(k)$ is the true reading and $\boldsymbol{\epsilon}(k)$ is the error in the true measurement with variance σ_ϵ^2. On this model, we implement a novel entropy adaptive Kalman filter followed by LSTM network in precise localization of UAV, as discussed next.

A. EAKF Formulation

Consider a linear system described by the equations

$$\begin{aligned}\mathbf{x}(k) =& \mathbf{F}(k-1)\mathbf{x}(k-1) + \mathbf{q}(k-1)\\ \mathbf{y}(k) =& \mathbf{H}(k)\mathbf{x}(k) + \mathbf{r}(k)\end{aligned} \tag{1}$$

where $\mathbf{x}(k) \in \mathbb{R}^n$ is the n-dimensional state vector, $\mathbf{y}(k) \in \mathbb{R}^m$ is the m-dimensional measurement vector at k-th instant. $\mathbf{F}$ and $\mathbf{H}$ denote the state transition matrix and observation matrix at the reference instances. Also, $\mathbf{q}(k-1)$ and $\mathbf{r}(k)$ represent zero-mean and mutually uncorrelated process and measurement noise statistics, respectively, with the covariance defined as

$$\mathbb{E}[\mathbf{q}(k-1)\mathbf{q}^T(k-1)] = \mathbf{Q}(k-1);\ \mathbb{E}[\mathbf{r}(k)\mathbf{r}^T(k)] = \mathbf{R}(k). \tag{2}$$

From (1), the prior estimate $\hat{\mathbf{x}}(k|k-1)$ and prior covariance $\mathbf{P}(k|k-1)$ are written as

$$\begin{aligned}\hat{\mathbf{x}}(k|k-1) &= \mathbf{F}(k-1)\hat{\mathbf{x}}(k-1|k-1)\\ \mathbf{P}(k|k-1) &= \mathbf{F}(k-1)\mathbf{P}(k-1|k-1)\mathbf{F}^T(k-1) + \mathbf{Q}(k-1).\end{aligned} \tag{3}$$

Consequently, the conventional Kalman gain is written as

$$\begin{aligned}\mathbf{K}(k) =& \mathbf{P}(k|k-1)\mathbf{H}^T(k)\\ &\times [\mathbf{H}(k)\mathbf{P}(k|k-1)\mathbf{H}^T(k) + \mathbf{R}(k)]^{-1}\end{aligned} \tag{4}$$

Based on the Kalman gain, the posterior state estimate $\hat{\mathbf{x}}(k|k)$ and posterior covariance from the prior state estimate and prior covariance, respectively, is given as

$$\begin{aligned}\hat{\mathbf{x}}(k|k) =& \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(k)(\mathbf{y}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k|k-1))\\ \mathbf{P}(k|k) =& (\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k))\mathbf{P}(k|k-1)(\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k))^T\\ &+ \mathbf{K}(k)\mathbf{R}(k)\mathbf{K}^T(k).\end{aligned} \tag{5}$$

For the linear model described above, a-priori and measurement estimates are collectively written as

$$\begin{bmatrix}\hat{\mathbf{x}}(k|k-1)\\ \mathbf{y}(k)\end{bmatrix} = \begin{bmatrix}\mathbf{I}\\ \mathbf{H}(k)\end{bmatrix}\mathbf{x}(k) + \boldsymbol{\nu}(k) \tag{6}$$

where $\nu(k) = [-(\mathbf{x}(k) - \hat{\mathbf{x}}(k|k-1))\ \ \mathbf{r}(k)]^T$. Further, we define

$$\begin{aligned}\mathbb{E}[\nu(k)\nu^T(k)] =& \begin{bmatrix}\mathbf{B}_p(k|k-1)\mathbf{B}_p^T(k|k-1) & 0\\ 0 & \mathbf{B}_r(k)\mathbf{B}_r^T(k)\end{bmatrix}\\ =& \mathbf{B}(k)\mathbf{B}^T(k)\end{aligned} \tag{7}$$

Algorithm 1: EAKF initial estimation algorithm

Result: Coarse location estimate: $\hat{\mathbf{x}}_c \rightarrow \{x_c, y_c, z_c\}$
1. Initialize EAKF at $k = 1$, kernel bandwidth σ_i, $\varepsilon \rightarrow 0$, initial estimate $\hat{\mathbf{x}}(0|0)$, and initial covariance matrix $\mathbf{P}(0|0)$
2. Generate $\hat{\mathbf{x}}(k|k-1)$ and $\mathbf{P}(k|k-1)$ using (5) and $\mathbf{B}_p(k|k-1)$ using (7)
3. Set $t = 1$ and $\hat{\mathbf{x}}(k|k)_0 = \hat{\mathbf{x}}(k|k-1)$, where $\hat{\mathbf{x}}(k|k)_t$ is the estimated state at fixed-point iteration t
4. Estimate $\hat{\mathbf{x}}(k|k)_t$ and update $\mathbf{K}(k)$ using Lemma 1
5. **If** $\frac{\|\hat{\mathbf{x}}(k|k)_t - \hat{\mathbf{x}}(k|k)_{t-1}\|}{\|\hat{\mathbf{x}}(k|k)_{t-1}\|} \leq \varepsilon$; set $\hat{\mathbf{x}}(k|k) \rightarrow \hat{\mathbf{x}}(k|k)_t$ and update $\mathbf{P}(k|k)$ with $\mathbf{K}(k)$ from step 4 in (5)
6. **Else**, repeat step 4

where $\mathbf{B}(k)$ is obtained by the Cholesky factorization of $\mathbb{E}[\nu(k)\nu^T(k)]$. Multiplying (6) by $\mathbf{B}^{-1}(k)$, we get $\mathbf{Z}(k) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{e}(k)$, where $\mathbf{Z}(k) = \mathbf{B}^{-1}(k)[\hat{\mathbf{x}}(k|k-1) \;\; \mathbf{y}(k)]^T$, $\mathbf{A}(k) = \mathbf{B}^{-1}(k)[\mathbf{I} \;\; \mathbf{H}(k)]^T$, and $\mathbf{e}(k) = \mathbf{B}^{-1}(k)\nu(k)$. Further, since $\mathbb{E}[\mathbf{e}(k)\mathbf{e}^T(k)] = \mathbf{I}$, $\mathbf{e}(k)$ is a white noise process. To account for the heavy tailed noise profile from the intermediate plasma channel, we use the adaptive entropy function defined as $\mathscr{L}(\mathbf{x}(k)) = \sum_{i,j=1}^{N,M} \frac{\omega_i}{M} G_{\sigma_i}(\mathbf{e}_j(k))$, where $G_{\sigma_i}(\mathbf{e}_j(k)) = \exp\left(-\frac{\|\mathbf{e}_j(k)\|_2^2}{2\sigma_i^2}\right)$, with $\mathbf{e}_j(k) = z_j(k) - a_j(k)\mathbf{x}(k)$, $z_j(k)$ being the j-th element of $\mathbf{Z}(k)$, $a_j(k)$ the j-th row of $\mathbf{A}(k)$, and ω_i denoting the normalized weight, s.t. $\sum_{i,j=1}^{N,M} \frac{\omega_i}{M} = 1$. Thus, based on the adaptive entropy function, the optimal estimate of $\mathbf{x}(k)$ can be obtained by maximizing $(\mathbf{P}_1)$.

$$(\mathbf{P}_1) : \hat{\mathbf{x}}(k) = \arg\max_{\mathbf{x}(k)} \mathscr{L}(\mathbf{x}(k)) \tag{8}$$

Lemma 1. *The fixed-point iterative solution of* $(\mathbf{P}_1)$ *is given by* $\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(k)\left(\mathbf{y}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k|k-1)\right)$ *where* $\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{H}^T(k)\left(\mathbf{H}(k)\mathbf{P}(k|k-1)\mathbf{H}^T(k) + \mathbf{R}(k)\right)^{-1}$, $\mathbf{P}(k|k-1) = \mathbf{B}_p(k|k-1)\mathbf{G}_x^{-1}(k)\mathbf{B}_p^T(k|k-1)$, *and* $\mathbf{R}(k|k-1) = \mathbf{B}_r(k)\mathbf{G}_y^{-1}(k)\mathbf{B}_r^T(k)$.

Proof. See Appendix A. □

B. EAKF-LSTM Position Estimation Framework

Let the previous position estimates for j time horizons, $j \in \{1, \cdots, k-1\}$, be stored in $\mathbf{h}_{k-1} = [\mathbf{h}_{k-2}^T, \hat{\mathbf{x}}^T(k-1)]^T \in \mathbb{R}^{(k-1)\times n}$, where $\hat{\mathbf{x}}(k-1)$ is generated at time step $k-1$ using Lemma 1. $\mathbf{h}_{k-1}$ is pre-processed to generate a vector of dimension $\mathbb{R}^{(k-1)n\times 1}$. The LSTM network takes the flattened $\mathbf{h}_{k-1}$ along with the current measurement $\mathbf{y}(k)$ generated using (1), to attain an input vector of dimension $\mathbb{R}^{kn\times 1}$. Thus, the hidden state is updated based on the previous state $\mathbf{h}_{k-1}$ and the input $\mathbf{y}(k)$, using the forget gate $\mathbf{f}_k$, the input gate $\mathbf{i}_k$, the cell state $\mathbf{c}_k$, and the output gate $\mathbf{o}_k$. Mathematically, we have

$$\begin{aligned}
&\mathbf{f}_k = \delta\left(\mathbf{W}_f[\mathbf{h}_{k-1}, \mathbf{y}(k)] + \mathbf{b}_f\right);\; \mathbf{i}_k = \delta\left(\mathbf{W}_i[\mathbf{h}_{k-1}, \mathbf{y}(k)] + \mathbf{b}_i\right)\\
&\mathbf{c}_k^{'} = \tanh\left(\mathbf{W}_c[\mathbf{h}_{k-1}, \mathbf{y}(k)] + \mathbf{b}_c\right);\; \mathbf{c}_k = \mathbf{f}_k\mathbf{c}_{k-1} + \mathbf{i}_k\mathbf{c}_k^{'}\\
&\mathbf{o}_k = \delta\left(\mathbf{W}_o[\mathbf{h}_{k-1}, \mathbf{y}(k)] + \mathbf{b}_o\right);\; \mathbf{h}_k = \mathbf{o}_k \tanh(\mathbf{c}_k)
\end{aligned} \tag{9}$$

where $\mathbf{h}_k \in \mathbb{R}^{d\times 1}$ is the hidden state vector, $\mathbf{W}$ are the weight matrices of order $\mathbb{R}^{(kn+d)\times d}$, and $\mathbf{b} \in \mathbb{R}^{d\times 1}$ are the bias vectors. $\delta(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function, and $\tanh(\cdot)$ is the hyperbolic tangent function. Finally, the L-step ahead updated state estimate is given by $\hat{\mathbf{x}}_k = \mathbf{W}_x\mathbf{h}_k + \mathbf{b}_x$, where $\mathbf{W}_x$ is the transformation matrix and $\mathbf{b}_x$ is the corresponding offset. The network parameters are optimized based on the mean squared error (MSE) loss function, evaluated over M training instances, given by: $\mathscr{L}_{\text{LSTM}} = \frac{1}{R}\sum_{i=1}^{R} \|\tilde{\mathbf{x}}_{k+i} - \hat{\mathbf{x}}_{k+i}\|_2^2$. The detailed localization procedure using the FNN, i.e., EAKF-LSTM network, is outlined in Algorithms 1 and 2. Computational complexity of the proposed framework is presented next.

Algorithm 2: FNN-based UAV localization algorithm

Result: True location: $\mathbf{x} \rightarrow \{x_T, y_T, z_T\}$
1. Initialize EAKF using Algorithm 1
2. Train LSTM network using the coarse estimates $\hat{\mathbf{x}}_c$
3. For each new input, invoke Algorithm 1 to obtain the state estimates
4. Use the LSTM network to make predictions based on the state estimates produced Algorithm 1
5. Repeat steps 3 and 4 to update predictions over time

C. Analysis of Computational Complexity

In this subsection, we analyze the computational complexity of the proposed FNN algorithm in terms of floating point operations (FLOPS). The aggregate number of FLOPS involved in the FNN algorithm include the ones in (3), Lemma 1 and the LSTM framework. The total number of FLOPS in (3) and Lemma 1 are: $4n^3 + n^2 - n + \mathscr{O}(m^3)$ and $(2\kappa+8)n^3 + (4\kappa+6)n^2m + (2\kappa-1)n^2 + (4\kappa+2)nm^2 + (3\kappa-1)nm + (4\kappa-1)n + 2\kappa m^3 + 2\kappa m + \kappa\mathscr{O}(n^3) + 2\kappa\mathscr{O}(m^3)$ respectively, where κ is the average number of iterations required for convergence. Next, for a training length of l_t, and establishing $l_t \ll D^2$, the number of FLOPs in the RNN part of the framework is $\mathscr{O}(l_t D^2)$. Thus, the total involved FLOPS in the FNN framework is

$$\begin{aligned}
\mathscr{C}_{FNN} = {} & 4n^3 + (2\kappa+8)n^3 + (4\kappa+6)n^2m + 2\kappa n^2 + \mathscr{O}(l_t D^2)\\
& + (4\kappa+2)nm^2 + (3\kappa-1)nm + 2(2\kappa-1)n\\
& + 2\kappa m^3 + 2\kappa m + \kappa\mathscr{O}(n^3) + (2\kappa+1)\mathscr{O}(m^3).
\end{aligned} \tag{10}$$

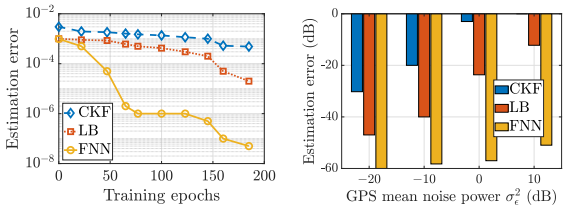
It is notable that, because of multi-threading property in most moderately powered processors, the dimensions of the input and output vectors to and from the Kalman filter reduces to unity. Thus, the complexity in (10) reduces to $\mathscr{C}_{FNN} = 25\kappa + 18 + \mathscr{O}(l_t D^2) \approx \mathscr{O}(l_t D^2)$. This is mostly governed by the optimal training length requirement by RNN. In the proposed FNN, the training length l_t reduces because of multiple recycling of the EAKF-LSTM network. This significantly reduces the computational complexity compared to the orthodox learning algorithms. Performance results are presented next.

IV. Results and Discussions

This section presents the relative performance of the proposed FNN framework and the state-of-the-art statistical (CKF) [4] and learning-based (LB) [13] frameworks. The data of a dual frequency GPS receiver located at IISc Bangalore is taken from the Scripps orbit and permanent array center to validate the proposed algorithm. Table I lists the used parameter values.

TABLE I: FNN parameters

Activation function	Hidden layers	Learning rate	Batch size	Training length	Training epochs	Optimum lag samples, k
ReLU	32	10^{-3}	500	15000	65	100



(a) (b)

Fig. 2: Comparison of estimation error with (a) training epochs, $\sigma_\epsilon^2 = 0.01$ and (b) average GPS noise power, 185-th training epochs.

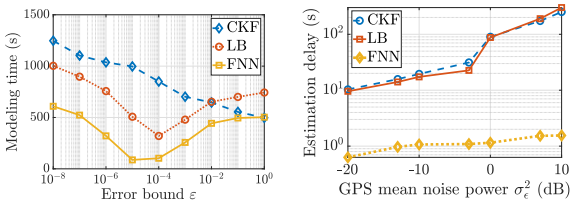
A. Localization Error Performance

Fig. 2(a) presents the error performance of the proposed FNN algorithm and compares it with the CKF and LB algorithms. It can be noted that the proposed algorithm achieves an nRMSE of the order 10^{-6} in about 60 training epochs. From the square and diamond-marked curves in Fig. 2(a), we conclude that none of the navigation filters in literature can achieve a precision of similar order. Moreover, the LB filter performs better as it has a statistical aid from the involved AKF. However, the non-Gaussian noise impairments in the data makes this filter inefficient in handling localization in such setups.

Denoting i-th training epoch as t_{e_i} with corresponding error e_i, the average estimation error is $\langle e \rangle = \frac{\sum_i t_{e_i} e_i}{\sum_i t_{e_i}}$. Numerically, the average errors in CKF, LB, and FNN frameworks are respectively 1.12×10^{-2}, 2.05×10^{-4}, and 5.05×10^{-7}. Thus, the proposed framework achieves 99.75% more accuracy than the state-of-the-art. From Fig. 2(b), we note that the proposed framework is fairly robust, with an nRMSE of $\approx 10^{-6}$ at all noisy GPS measurements, while the performance of the competitive frameworks massively deteriorate.

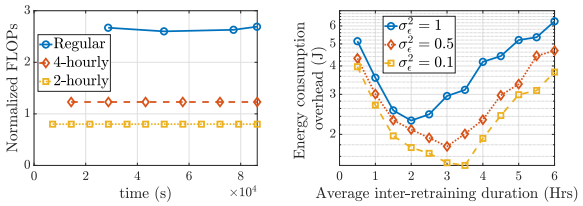
B. Complexity Analysis

Fig. 3(a) depicts the modeling time of the competitive localization algorithms. We observe that there exists an optimal error bound ε for all the learning-based algorithms. For a stricter error-bound, the learning module over-trains, leading to over-fitting, while a more relaxed error-bound leads to under-training, implying under-fitting. Such a phenomenon is not



(a) (b)

Fig. 3: Comparison of (a) modeling time versus error bound at $\sigma_\epsilon^2 = 0.1$ and (b) estimation delay versus error variance σ_ϵ^2 at $\varepsilon = 10^{-4}$.



(a) (b)

Fig. 4: (a) Normalized FLOPs vs. retraining instances; (b) energy overhead vs. average retraining interval at $\sigma_\epsilon^2 = 0.1$, and $\varepsilon = 10^{-4}$.

observed in the CKF statistical filter. As a consequence, we note that the LB localization framework achieves optimal modeling time of 320 s for $\varepsilon = 10^{-4}$. The proposed FNN framework outperforms LB with a modeling time of 87 s, which is 72.81% less than LB, while achieving an error bound of 10^{-6}.

From Fig. 3(b), we note that the delay in extracting subsequent position estimates is least in the proposed FNN algorithm. Notably, the estimation delay increases with the GPS measurement noise power. Similar to the estimation delay in the previous subsection, the average estimation delay with CKF, LB, and FNN algorithms are respectively 46.3 s, 45.94 s, and 0.995 s. Thus, a 97.83% reduction in delay is achieved. We implemented the proposed algorithm in an E3-1285 CPU @4.10 GHz, which showd a modeling time of 40 ms. Further, an estimation delay of ≈ 0.3 ms was noted on this processor.

Remark 1. *Learning-based navigation filters outperform the statistical filters by providing ahead-of-time prediction of subsequent UAV position coordinates.*

C. Retraining Strategy for the FNN Model

Fig. 4 analyses different retraining strategies for the proposed algorithm for practical deployment in energy-constrained setups. From Fig. 4(a) we observe that under 'Regular' retraining, which allows retraining of the proposed model only when the ε bound is not met, the normalized number of FLOPs (cf. Section III-C) involved in the retraining process are fairly high. This is because of the requirement to repeat the whole EAKF-LSTM network in this FNN algorithm. However, when a pre-emptive retraining policy is utilized, the number of FLOPs involved in the process is reduced by a factor of ≈ 3.12, as mostly only the neural network retraining is required. Further, under these strategies, fewer retraining data points are required as the estimates generated by the FNN algorithm are already with the ε bound, as defined in Algorithm 1. Further, from the simulation results in Fig. 4(b) we concluded that retraining the FNN every 3.5 hours gives the best result in terms of energy saving, which is an important factor in most practical UAVs. This majorly stems from exhausting more energy in retraining more often against waiting till the estimates cross the ε bound.

V. Conclusions

The proposed statistically-aided learning-based FNN framework aimed at estimating precise position of UAVs from the noisy GPS measurements in real-time. From the localization estimation studies under varying GPS noise power and precision

requirement, it was concluded that the EAKF-LSTM network (FNN) offers a faster, more robust, and more precise localization compared to the most competitive techniques in literature. The statistical filtering of non-Gaussian noise led to a decreased modeling and estimation delay with a lesser estimation error resulting from multiple recycling of this network. Furthermore, it was demonstrated that, pre-emptive retraining proves fruitful in energy constrained scenarios with reduced FLOPs and energy overhead. As an immediate extension, we plan to utilize the findings for UAV swarm mobility optimization.

ACKNOWLEDGEMENT

This work was supported in parts by the Science and Engineering Board, Dept. of Science and Technology (DST), Government of India, under Grant CRG/2019/002293, by the International Cooperation Division, DST, INT/Italy/P-34/2022, by the Indian National Academy of Engineering (INAE) through the Abdul Kalam Technology Innovation National Fellowship, and by the Prime Minister's Doctoral Research Fellowship.

APPENDIX

A. Proof of Lemma 1

Differentiating (8) with respect to $\mathbf{x}(k)$ we get

$$\frac{\partial \mathscr{L}(\mathbf{x}(k))}{\partial \mathbf{x}(k)} = \sum_{i=1}^{N}\sum_{j=1}^{M}\frac{\omega_i}{M}G_{\sigma_i}(\mathbf{e}_j(k))a_j^T(k)\,(z_j(k) - a_j(k)\mathbf{x}(k)) = 0 \tag{A1}$$

On solving (A1), we get $\mathbf{x}(k) = \left[\sum_{i=1}^{N}\sum_{j=1}^{M}\omega_i G_{\sigma_i}(\mathbf{e}_j(k))\, a_j^T(k)a_j(k)\right]^{-1}\left[\sum_{i=1}^{N}\sum_{j=1}^{M}\omega_i G_{\sigma_i}(\mathbf{e}_j(k))a_j^T(k)z_j(k)\right]$. On manipulation, we get

$$\mathbf{x}(k) = \left(\mathbf{A}^T(k)\mathbf{G}(k)\mathbf{A}^T(k)\right)^{-1}\left(\mathbf{A}^T(k)\mathbf{G}(k)\mathbf{Z}(k)\right) \tag{A2}$$

where $\mathbf{G}(k) = \text{diag}\left(\mathbf{G}_x(k), \mathbf{G}_y(k)\right)$, $\mathbf{G}_x(k) = \text{diag}\left(\sum_{i=1}^{N}\omega_i G_{\sigma_i}(\mathbf{e}_1(k)), \cdots, \sum_{i=1}^{N}\omega_i G_{\sigma_i}(\mathbf{e}_n(k))\right)$, and $\mathbf{G}_y(k) = \text{diag}\left(\sum_{i=1}^{N}\omega_i G_{\sigma_i}(\mathbf{e}_{n+1}(k)), \cdots, \sum_{i=1}^{N}\omega_i G_{\sigma_i}(\mathbf{e}_{n+m}(k))\right)$. The inverse term in (A2) can be simplified as

$$\begin{aligned}&\left(\mathbf{A}^T(k)\mathbf{G}(k)\mathbf{A}^T(k)\right)^{-1}\\ &= [(\mathbf{B}_p^{-1}(k|k-1))^T\mathbf{G}_x(k)\mathbf{B}_p^{-1}(k|k-1)\\ &+ \mathbf{H}^T(k)(\mathbf{B}_r^{-1}(k))^T\mathbf{G}_y(k)\mathbf{B}_r^{-1}(k)\mathbf{H}(k)]^{-1}.\end{aligned} \tag{A3}$$

Using matrix inversion lemma in (A3), we get

$$\begin{aligned}&\left(\mathbf{A}^T(k)\mathbf{G}(k)\mathbf{A}^T(k)\right)^{-1} = \mathbf{S}^{-1}(k) - \mathbf{S}^{-1}(k)\mathbf{H}^T(k)\\ &\times\left(\mathbf{U}^{-1}(k) + \mathbf{H}(k)\mathbf{S}^{-1}(k)\mathbf{H}^T(k)\right)^{-1}\mathbf{H}(k)\mathbf{S}^{-1}(k)\end{aligned} \tag{A4}$$

where $\mathbf{S}(k) = (\mathbf{B}_p^{-1}(k|k-1))^T\mathbf{G}_x(k)\mathbf{B}_p^{-1}(k|k-1)$, and $\mathbf{U}(k) = (\mathbf{B}_r^{-1}(k))^T\mathbf{G}_y(k)\mathbf{B}_r^{-1}(k)$. Further, the next product term of (A2) is simplified as

$$\mathbf{A}^T(k)\mathbf{G}(k)\mathbf{Z}(k) = \mathbf{S}(k)\hat{\mathbf{x}}(k|k-1) + \mathbf{H}^T(k)\mathbf{U}(k)\mathbf{y}(k). \tag{A5}$$

Therefore, using (A4) and (A5) in (A2) we get $\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}_G(k)\left(\mathbf{y}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k|k-1)\right)$, where $\mathbf{K}_G(k) = \mathbf{P}(k|k-1)\mathbf{H}^T(k)\left(\mathbf{H}(k)\mathbf{P}(k|k-1)\mathbf{H}^T(k) + \mathbf{R}(k)\right)^{-1}$, $\mathbf{P}(k|k-1) = \mathbf{B}_p(k|k-1)\mathbf{G}_x^{-1}(k)\mathbf{B}_p^T(k|k-1)$, and $\mathbf{R}(k|k-1) = \mathbf{B}_r(k)\mathbf{G}_y^{-1}(k)\mathbf{B}_r^T(k)$. □

REFERENCES

[1] S. Aggarwal, N. Kumar, and S. Tanwar, "Blockchain-envisioned UAV communication using 6G networks: Open issues, use cases, and future directions," *IEEE IoT J.*, vol. 8, no. 7, pp. 5416–5441, 2021.

[2] N. Varshney and S. De, "Design optimization for UAV aided sustainable 3D wireless communication at mmWaves," *IEEE Trans. Veh. Tech.*, 2022.

[3] N. Zhu, J. Marais, D. Bétaille, and M. Berbineau, "GNSS position integrity in urban environments: A review of literature," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 2762–2778, 2018.

[4] S. Bhattacharya and D. Mute, "Kalman filter-based RAIM for reliable aircraft positioning with GPS and NavIC constellations," *Sensors*, vol. 22, no. 1, pp. 6606–6637, 2020.

[5] J. Zhang, W. Zhou, and X. Wang, "UAV swarm navigation using dynamic adaptive Kalman filter and network navigation," *Sensors*, vol. 21, no. 16, p. 5374, 2021.

[6] B. Gong, S. Wang, M. Hao, X. Guan, and S. Li, "Range-based collaborative relative navigation for multiple unmanned aerial vehicles using consensus extended Kalman filter," *Aeros. Science Technol.*, vol. 112, p. 106647, 2021.

[7] X. Zhou, W. Sun, R. He, and H. Liu, "Unscented Kalman filter for UAV real-time localizating dynamic target on the ground," in *Adv. Intell. Informat. Hiding Multim. Signal Process.: Proc. IIH-MSP 2021 & FITAT 2021, Kaohsiung, Taiwan, Volume 2*. Springer, 2022, pp. 55–65.

[8] B. Yang, E. Yang, L. Yu, and C. Niu, "Adaptive extended Kalman filter-based fusion approach for high-precision UAV positioning in extremely confined environments," *IEEE/ASME Trans. Mechatron.*, 2022.

[9] L. Shi, B. Guo, Y. Liu, and J. Li, "Characteristic of plasma sheath channel and its effect on communication," *Progress In Electromagn. Res.*, vol. 123, pp. 321–336, 2012.

[10] A. K. Mandal and S. De, "Analysis of wireless communication over electromagnetic impulse noise channel," *IEEE Trans. Wireless Commun.*, vol. 22, no. 2, pp. 1187–1200, 2023.

[11] X. Fan, G. Wang, J. Han, and Y. Wang, "A background-impulse Kalman filter with non-gaussian measurement noises," *IEEE Trans. Syst. Man. Cybernet.: Syst.*, 2022.

[12] M. M. U. Chowdhury, F. Erden, and I. Guvenc, "RSS-based Q-learning for indoor UAV navigation," in *Proc. IEEE Military Commun. Conf. (MILCOM)*. IEEE, 2019, pp. 121–126.

[13] H. Luo, Y. Luo, B. Han, and M. Zeng, "A learning-based noise tracking method of adaptive Kalman filter for UAV positioning," in *IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*. IEEE, 2022, pp. 440–445.

[14] M. Y. Arafat and S. Moh, "Localization and clustering based on swarm intelligence in UAV networks for emergency communications," *IEEE IoT J.*, vol. 6, no. 5, pp. 8958–8976, 2019.

[15] D. Ebrahimi, S. Sharafeddine, P.-H. Ho, and C. Assi, "Autonomous UAV trajectory for localizing ground objects: A reinforcement learning approach," *IEEE Trans. Mob. Comput.*, vol. 20, no. 4, pp. 1312–1324, 2020.

[16] B. Brik, A. Ksentini, and M. Bouaziz, "Federated learning for UAVs-enabled wireless networks: Use cases, challenges, and open problems," *IEEE Access*, vol. 8, pp. 53 841–53 849, 2020.

[17] H. Qin, Z. Meng, W. Meng, X. Chen, H. Sun, F. Lin, and M. H. Ang, "Autonomous exploration and mapping system using heterogeneous UAVs and UGVs in GPS-denied environments," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1339–1350, 2019.

[18] Y. Tang, Y. Hu, J. Cui, F. Liao, M. Lao, F. Lin, and R. S. Teo, "Vision-aided multi-UAV autonomous flocking in GPS-denied environment," *IEEE Trans. Indus. Electron.*, vol. 66, no. 1, pp. 616–626, 2018.

[19] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14 413–14 423, 2020.

[20] W. Wang, D. Marelli, and M. Fu, "Multiple-vehicle localization using maximum likelihood kalman filtering and ultra-wideband signals," *IEEE Sensors J.*, vol. 21, no. 4, pp. 4949–4956, 2020.

[21] B. Chen, X. Liu, H. Zhao, and J. C. Principe, "Maximum correntropy Kalman filter," *Automat.*, vol. 76, pp. 70–77, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S000510981630396X