# Target Tracking with Distributed Sensing and Optimal Data Migration.

Vahram Stepanyan *
*NASA Ames Research Center, KBR Wyle Services, Moffett Field, CA 94035*

Keerthana Kannan†
*NASA Ames Research Center, KBR Wyle Services, Moffett Field, CA 94035*

Evan T. Kawamura‡
*NASA Ames Research Center, Moffett Field, CA 94035*

Thomas Lombaerts§
*NASA Ames Research Center, KBR Wyle Services, Moffett Field, CA 94035*

Corey Ippolito ¶
*NASA Ames Research Center, Moffett Field, CA 94035*

**The paper presents an Extended Kalman Filter based framework for airborne target tracking using adaptive information fusion from multi-modal multi-rate distributed sensors network. First, the tracking algorithm execution location is determined using an optimal data migration strategy, which also computes the associated delays for each sensor data to arrive at the computing location. Next, the fast (zero-delay) sensors information is dynamically fused in the filter correction procedure at the arrival instance of each valid sensor reading. Finally, the target estimation is updated based on the valid slow (delayed) data, which are grouped according to the delay-time steps before application of the Larsen's method. This approach is applied to the synthetic sensor data generated by means of the ground based radar and camera models for the simulated target flight in Reflection simulation environment.**

## I. Introduction

Obtaining accurate and in-time information about the air vehicles states and surrounding environment is crucial for safe autonomous operations in air transportation systems for passengers and cargo in dense and highly dynamic metropolitan areas. This is one of the focus areas of the NASA Advanced Air Mobility (AAM) Mission, which is designed to support a variety of on-board, ground-piloted and increasingly autonomous flight operations by the exploration of new sensing, perception, and collision avoidance technologies capable of replacing onboard pilot's functions [1–4].

Both ground based and airborne object detection and tracking are integral parts of many collision avoidance solutions. Traditionally, ground based radars have been used to detect terrains and air vehicles for safe airspace operations. Recently, they found widespread applications in autonomous driving and driver assistive systems along with other type of sensors such as, visual, LIDAR or infrared. However, size, weight, power and cost limitations create challenges in their onboard applications to unmanned aircraft systems [5].

In many target tracking and navigation applications, the problem is considered in a mixed continuous-discrete time domain, where the target dynamics are in continuous-time and the measurements are in discrete-time. While the target motion is commonly considered as a continuous time coordinate-uncoupled white-noise acceleration model for a point-mass object [6], which is linear in states, the measurements are in discrete time domain, and are nonlinear in majority of cases. Therefore the Extended Kalman Filter (EKF) or its variants make a suitable framework for target tracking in Cartesian coordinates with measurements linearization [7].

*Technical Advisor, KBR Wyle Services, Intelligent Systems Division, Mail Stop 269-1, email: vahram.stepanyan@nasa.gov
†Aerospace Research Engineer, KBR Wyle Services, Intelligent Systems Division, Mail Stop 269-1, email: keerthana.kannan@nasa.gov
‡Aerospace Research Engineer, Intelligent Systems Division, email: evan.t.kawamura@nasa.gov
§Aerospace Research Engineer, KBR Wyle Services, Intelligent Systems Division, Mail Stop 269-1, email: thomas.lombaerts@nasa.gov
¶Aerospace Research Engineer, Intelligent Systems Division, Mail Stop 269-1, email: corey.a.ippolito@nasa.gov

To improve the situational awareness, distributed homogeneous or heterogeneous sensors network is used in various applications for airspace and ground transportation systems (see for example [8–10] and referenced therein), where the information from all sensors are transmitted to a computing center for processing. However, because of the diverse sensor sampling rates, different initial measurements processing times at sources and network communication delays, sensor data from the same target arrive at the tracking center out-of-sequence, that is data acquired at earlier time arrives at the fusion center after the tracks were already updated at the current time [11–13].

The existing literature on applications of the Kalman filter with delayed measurements can be classified into two main methods: augmenting the current state with the appropriate past information and fusing the delayed measurements on arrival [14]. The first group of methods achieve optimal estimates at the expense of increased system's size, hence the computational load [15, 16]. Therefore, they can be effectively applied to cases of small number sample time delays or cases when computational burden is not an issue [17, 18]. The second group of methods can handle large and variable delays with a reasonably low computational load, although sometimes achieve only sub-optimal estimates. Some methods, such as the filter recalculation method, are optimal although have high computational load for nonlinear systems [19, 20], some provide a trade-off between optimality and computational load [11, 21, 22]. In any case, they can process interim measurements while incorporating the delayed ones, which makes them suitable for autonomous navigation [23].

As airspace operations density increases, it becomes challenging to make rapid decisions that use geo-diverse and dynamic data because of communication rate, bandwidth and execution hardware and power limitations. Therefore, it is necessary to automatically learn where to execute the tracking algorithm to meet users' latency, cost and power consumption requirements. There are several computing models and platforms such as Edge Computing [24], Fog Computing [25, 26], and Mobile Edge Computing (MEC) [27, 28] to meet these challenges. Some of these approaches are devoted to substantially reducing the operational costs and satisfy the quality of service requirements in Internet of Vehicles (IoV) or vehicular networks [29–31]. Utilizing edge computing infrastructure can be instrumental for safety and reliability of increasingly autonomous aircraft operations. Therefore, migrating the data and estimation algorithm between computing centers can play a critical role in in-time target tracking for airspace operations needs, such as real-time trajectory re-planning for sense and avoid or other contingencies handling [32].

In the paper, we consider an an Extended Kalman Filter based framework for airborne target tracking using adaptive information fusion from multi-modal multi-rate distributed sensors network. First, the tracking algorithm execution location is determined using an optimal data migration strategy, which also computes the associated delays for each sensor data to arrive at the computing location. Next, the fast (zero-delay) sensors information is dynamically fused in the filter correction procedure at the arrival instance of each valid sensor reading, similar to approach in Ref. [33]. Finally, the target estimation is updated based on the valid slow (delayed) data, which are grouped according to the delay-time steps before application of the Larsen's method. This approach is applied to synthetic sensor data generated by means of the ground based radar and camera models for the simulated target flight in Reflection simulation environment.

## II. Problem Formulation

We consider a target tracking problem using information from ground based distributed multi-modal multi-rate sensors $S_i$, $i = 1, \ldots, N_s$, using Extended Kalman Filter (EKF) framework. We assume that the sensors' positions $\boldsymbol{p}_i^s$, $i = 1, \ldots, N_s$ in a Cartesian ENU (East-North-Up) frame $F_0$ and the rotation matrices $\mathbb{R}_i$, $i = 1, \ldots, N_s$ from sensor-attached frames $F_i$, $i = 1, \ldots, N_s$ to $F_0$ are known. Sensors transmit the target's information over the wired or wireless network as soon as it becomes available. This information can be in any form, partially or completely processed, which may include data classification, compression, encryption, etc. depending on the type of sensor. For example, in the case of radar, this processing may include obtaining target's range, azimuth and elevation information from the sensor's raw readings. In any case, we assume that the sensors data include the time stamp of the readings, validity flag indicating that the target is in the sensors field of view, the targets information in some form, and that the size of data to be transmitted by each sensor is known.

We assume that the sensors are capable of transmitting data through an existing wired or wireless communication infrastructure such as cell towers or internet connections, or portable transmitters, which are deployed on-demand taking into account local specifics. In any case, we assume that the transmitter/receiver locations and communication characteristics from sensors to network infrastructure and in the network are known. Let the transmitting towers $T_i$, $i = 1, \ldots, N_t$ be located at $\boldsymbol{p}_i^t$, $i = 1, \ldots, N_t$ in the frame $F_0$ and have transmit power of $p_i^t$, $i = 1, \ldots, N_t$ (in watts) for wireless communication. Based on these powers and the adopted wireless communication model, a maximum transmit distance $d_i^m,\ i = 1, \ldots, N_t$ for each tower can be computed from the perspective of acceptable received signal to noise ratio. Additionally, we assume that the sensors have transmit power of $p_i^s,\ i = 1, \ldots, N_t$, which is enough to reach the closest tower. All wired connection legs have a uniform rate of $\omega$ (in bits per second).

The data are received and processed at computing centers $C_i,\ i = 1, \ldots, N_c$ using information fusion and target tracking algorithms. It is assumed that the locations $\boldsymbol{p}_i^c\ i = 1, \ldots, N_c$ in the frame $F_0$ and computing capacities of these centers are known. The computing centers are connected via wired or wireless network, depending on the available infrastructure and the regional data processing demand. Since data sources have diverse locations, the information fusion and target tracking algorithms have to deal with the latency in communications and processing.

The objective is to design algorithms for migrating data to computing centers with minimum end-to-end latency, for fusion of information from all sensors as it becomes available, and for tracking the target based on dynamically resizing observations.

## III. Optimal Data Migration

We estimate the elapsed time based on the available best knowledge of the locations, data processing capabilities at transmit towers, communication networks transmission rates and bandwidths, and execution capabilities at computing centers.

To this end, we first map the sensors to communication towers based on highest data rate, which is equivalent to the smallest distance for wireless connection. We will denote $S_i \in H(T_m)$ to indicate that the $i-th$ sensor has a wired or wireless connection with the tower $T_m$. Likewise, we map the computing centers to closest towers and denote $C_j \in H(T_k)$ to indicate that the $j-th$ computing center has a wired or wireless connection with the tower $T_k$. To streamline the derivations, we introduce quantities $s_{im} = 1,\ i = 1, \ldots, N_s$ to indicate that the sensor $S_i$ has a wired connection with the associated tower $T_m$ and $s_{im} = 0$ otherwise; $c_{jk} = 1,\ j = 1, \ldots, N_c$ to indicate that the computing center $C_j$ has a wired connection with the associated tower $T_k$ and $c_{jk} = 0$ otherwise; $w_{km} = 1,\ k, m = 1, \ldots, N_t$ to indicate that the tower $T_m$ has a wired connection with the tower $T_k$ and $w_{km} = 0$ otherwise.

Let the data of size $D(S_i)$ bits be transmitted from the $i-th$ sensor $S_i \in H(T_m)$ to the $j-th$ computing center $C_j \in H(T_k)$. If $S_i$ has a wired connection with $T_m$, then according the model introduced in Ref. [34] the average communication latency of this data transmission is given by

$$\tau_{wr}(S_i, T_m) = \frac{d(S_i, T_m)}{c} + \frac{D(S_i)}{\omega}\,, \tag{1}$$

where $d(S_i, T_m) = \|\boldsymbol{p}_i^s - \boldsymbol{p}_m^t\|$ is the Euclidean distance between $S_i$ and $T_k$, $c$ is the propagation speed (in meter/second) of electromagnetic waves. Otherwise, we assume a non-orthogonal multiple access (NOMA) protocol for the wireless data transmission from $S_i$ to $T_k$ [35, 36], which allows all users to simultaneously utilize the entire system bandwidth with a multiple access interference inside the coverage area, but neglects the interference between adjacent coverage areas, and compute the up-link transmission rate as follows

$$\rho_u(S_i, T_m) = B_{sys}^m \log_2\left(1 + \frac{p_i^s g(s_i, T_m)}{\sigma_a^2 + \sum_{h=1,\ h \neq i, S_h \in H(T_m)}^{h=N_m} p_h^s g(S_h, T_m)}\right), \tag{2}$$

where $N_m$ is number of sensors in $H(T_m)$, $B_{sys}^m$ is the network bandwidth associated with the tower $T_m$, $p_i^s$ is the $i-th$ sensor's transmit power, $\sigma_a^2$ is the power of ambient white Gaussian noise at the receiver and $g(s_i, T_m)$ is the channel power gain, which depends on $d(S_i, T_m)$ as

$$g(s_i, T_m) = \frac{g_0}{d^2(S_i, T_m)}\,, \tag{3}$$

where $g_0$ is the reference gain [37]. Then corresponding up-link data transmission latency is computed for data volume $D(S_i)$ as

$$\tau_u(D(S_i), T_m) = \frac{D(S_i)}{\rho_u(S_i, T_m)}\,. \tag{4}$$

Thus the latency of $i-th$ sensor's data to reach the associated tower $T_m$ can be expressed as

$$\tau(D(S_i), S_i, T_m) = s_{im}\tau_{wr}(S_i, T_m) + (1 - s_{im})\tau_u(D(S_i), T_m)\,. \tag{5}$$

We assume that the communication tower $T_m$ processes data as they arrive trough a wireless channel with the rate of $\lambda_T$ in terms of CPU cycles per second. Then the processing latency is computed as

$$\tau_{pr}(D(S_i),T_m) = \frac{D(\cdot)}{\alpha_{Tm}\lambda_{Tm}}, \tag{6}$$

where $\alpha_T$ stands for the number of bits processed per one CPU cycle.

Next, we consider transmission of data $D(S_i)$ from $T_m$ to $T_k$ in the following cases.

*Case 1.* There is a wired connection between $T_m$ and $T_k$.

In this case the communication latency is computed similar to equation (1) as

$$\tau(D(S_i),T_m,T_k) = \frac{d(T_m,T_k)}{c} + \frac{D(S_i)}{\omega}. \tag{7}$$

*Case 2.* There is no wired connection between $T_m$ and $T_k$ and $d(T_m,T_k) <= d_m^t$.

We use orthogonal frequency-division multiple access (OFDMA) protocol [38] for the wireless communication between the towers. Then the data transmission latency is computed as follows

$$\begin{aligned} \rho_d(T_m,T_k) &= B_{sys}^k \log_2\left(1 + \frac{p_m^t g_0}{\sigma_a^2 d^2(S_i,T_m)}\right) \\ \tau(D(S_i),T_m,T_k) &= \frac{D(S_i)}{\rho_d(T_m,T_k)}, \end{aligned} \tag{8}$$

where $p_i^t$ is the $m-th$ tower's transmit power,.

*Case 3.* There is no wired connection between $T_m$ and $T_k$ and $d(T_m,T_k) > d_m^t$.

In this case the data are transmitted from $T_m$ to $T_k$ using other towers in the network for relay. Our goal is to find the optimal path, that is to find sequence of towers $T_m = T_{m_1}, T_{m_2}, \dots, T_{m_h} = T_k$ which provide the minimum latency for data transmission. This is a variant of the Traveling Salesman Problem (TSP) [39] with the starting node at $T_m$, ending node at $T_k$ and additional constraint on the wireless transmission legs length. For the simplicity of formulation we introduce the nodes $v_1,\dots,v_n$ such that $v_1 = T_m$, $v_n = T_k$, $n = N_t$. The cost of travel $c_{ic}$ between nodes $v_i$ and $v_j$ is the latency of data transfer from $v_i$ to $v_j$, which is computed according to equation (7) if there is a wired connection from from $v_i$ to $v_j$ and according to equations (8) otherwise.

$$c_{ij} = w_{ij}\left[\frac{d(T_m,T_k)}{c} + \frac{D(S_i)}{\omega}\right] + (1-w_{ij})\frac{D(S_i)}{\rho_d(v_i,v_j)}, \tag{9}$$

where $w_{ij} = 0$ if there is a a wired connection from from $v_i$ to $v_j$ and $w_{ij} = 1$ otherwise. Then the TSP is formulated as a mixed-integer linear program (MILP) for the following decision variables: binary variables $x_{ij} = 1$ if a visit to node $i$ is followed by a visit to node $j$ and $x_{ij} = 0$ otherwise; integer variables $u_i$, which are used to prevent sub-tours; $i, j = 1,\dots,m$. The optimization problem is formulated as:

$$\text{minimize} \quad f = \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}t_{ij} \tag{10}$$

$$\text{subject to:} \quad \sum_{j=2}^{n} x_{1j} = \sum_{i=1}^{n-1} x_{in} = 1 \tag{11}$$

$$\sum_{j=2}^{n} x_{ij} \le 1,\ i = 2,\dots,n-1 \tag{12}$$

$$\sum_{j=2}^{n} x_{ji} = \sum_{i=1}^{n-1} x_{ij},\ i = 2,\dots,n-1 \tag{13}$$

$$\sum_{j=1}^{n} x_{ij}(1-w_{ij})d(v_i,v_j) \le d_m^i,\ i = 1,\dots,n \tag{14}$$

$$2 \le u_i \le n,\ i = 2,\dots,n \tag{15}$$

$$u_i - u_j + 1 \le (n-1)(1-x_{ij}),\ i,j = 1,\dots,n, \tag{16}$$

where constraints (11) guarantee that each path starts at $v_1$ and ends at $v_n$, constraints (12) ensure that every node is visited at most once, constraints (13) guarantee that, if a node is visited, it is preceded and followed by exactly one other visit, constraints (14)) ensure the distance between two nodes with wireless connection on the path is within the maximum allowable limits, and constraints (15) and (16) are necessary to prevent sub-tours according to Miller–Tucker–Zemlin (MTZ) formulation for the traveling salesman problem (TSP) [34].

The solution of MILP (10) is a sequence of relay nodes $v_{m_1}, v_{m_2}, \ldots, v_{m_h}$ for the data transfer from $T_m$ to $T_k$. The corresponding elapsed time is computed as

$$\tau(D(S_i), T_m, T_k) = \sum_{i=1}^{h-1} t_{m_i, m_{i+1}}, \tag{17}$$

assuming that the data package is being transferred as is without any additional processing. However, it is being processed at the destination tower $T_k$ before transmitting to the computing center $C_j$. The processing time is computed according to equation (6), replacing index $m$ with $k$.

The last data transfer leg is from tower $T_k$ to computing center $C_j$. It is computed similar to (5)

$$\tau(D(S_i), T_k, C_j) = c_{jk}\tau_{wr}(T_k, C_j) + (1 - c_{jk})\tau_d(D(S_i), T_k), \tag{18}$$

where $\tau_{wr}(T_k, C_j)$ is computed according to equation (1) with understandable substitution of $T_k$, $C_j$ for $S_i$, $T_m$, and $\tau_d(D(S_i), T_k)$ is computed according to equation (8) with substitution of $T_k$, $C_j$ for $T_m$, $T_k$.

For the execution time estimation, we assume that the computing center $C_j$ has a data processing capability of $\lambda_{Cj}$ cycles per second and $\alpha_{Cj}$ bits per one CPU cycle. Then the center $C_j$ will spend

$$\tau_{pr}(D(S_i), C_j) = \frac{D(S_i)}{\alpha_{Cj}\lambda_{Cj}} \tag{19}$$

seconds to process a data of size $D(S_i)$, which may include for example image processing in case of raw camera data, and

$$\tau(A, C_j) = \frac{f(A)}{\lambda_{Ci}} \tag{20}$$

second to execute the estimation algorithm, which is assumed to have a size of $f(A)$ in terms of CPU cycles. Then the total elapsed time for the $S_i$ sensor's data $D(S_i)$ to reach to the computing center $C_j$ and to ready for the estimation algorithm can be estimated as

$$\begin{aligned} \tau(S_i, C_j) &= \tau(D(S_i), S_i, T_m) + (1 - s_{im})\tau_{pr}(D(S_i), T_m) + \tau(D(S_i), T_m, T_k) \\ &+ (1 - c_{jk})\tau_{pr}(D(S_i), T_k) + \tau(D(S_i), T_k, C_j) + \tau_{pr}(D(S_i), C_j). \end{aligned} \tag{21}$$

Let the data package of the estimation algorithm of size $D(A)$ be originally stored at computing center $C_l$. The elapsed time of its transfer to $C_j$ is computed similar to any sensor data as

$$\begin{aligned} \tau(A, C_j) &= \tau(D(A), C_l, T_m) + (1 - c_{lm})\tau_{pr}(D(A, T_m) + \tau(D(A), T_m, T_k) \\ &+ (1 - c_{jk})\tau_{pr}(D(A), T_k) + \tau(D(A), T_k, C_j) + \tau_{pr}(D(A), C_j). \end{aligned} \tag{22}$$

Obviously $\tau(A, C_j) = 0$ if $j = l$.

The estimation algorithm will be executed if all available sensor data and the algorithm data reach to the computing center $C_j$. That is the total time of the algorithm execution at $C_j$ with all input data is

$$\tau(C_j) = \max\left[\tau(S_1, C_j), \ldots, \tau(S_{N_c}, C_j)\right] + \tau(A, C_j). \tag{23}$$

We notice that data compression, encoding and decoding or other type of processing can be computed as in (6).

Now we formulate an optimization problem to determine the data and estimation algorithm migration policy. Our goal is to design a decision making algorithm to send the data and estimation algorithm itself to the computing center while minimizing the latency $\tau(C_j), j = 1, \ldots, N_c$ as estimated in (23). To this end, we introduce binary variables

$x_i, i = 1, \ldots, N_c$, which take on value $x_j = 1$ when the target tracking is placed in the computing center $C_j$ for execution and zero otherwise. To make sure that only on location can be selected, we impose a constraint

$$\sum_{i=1}^{N_c} x_i = 1 \tag{24}$$

on the binary variables $x_i$. The cost function to be minimized is expressed as

$$f = \sum_{i=1}^{N_c} x_i \tau(C_i) \, . \tag{25}$$

The user imposed latency upper limit, if any, is enforced via equation

$$\sum_{i=1}^{N_c} x_i \tau(C_i) \leq \tau_{\max} \, . \tag{26}$$

The optimization problem is a mixed-integer linear program

$$\begin{array}{ll} \text{minimize} & f \\ \text{subject to:} & 0 \leq x_i \leq 1, \; i = 1, ..., N_c, \; and \; (24), \; (26) \, . \end{array} \tag{27}$$

with possible additional constraints on available computational, storage and power resources, which are not detailed here. It determines the optimal execution computing center $C_j$. Therefore, equation (21) represents the data transfer latency for each sensor $S_i, i = 1, \ldots, N_s$.

## IV. Extended Kalman Filter Framework

### A. Filter State Description.

We use a conventional continuous time coordinate-uncoupled white-noise acceleration model for a point-mass object to describe the target's motion in Cartesian frame $F_0$ [6]

$$\begin{aligned} \ddot{x}(t) &= n_x(t) \\ \ddot{y}(t) &= n_y(t) \\ \ddot{z}(t) &= n_z(t) \, , \end{aligned} \tag{28}$$

where $n_x(t)$, $n_y(t)$, $n_z(t)$ are zero-mean, Gaussian distributed, and uncorrelated processes with variances $\sigma_x^2$, $\sigma_y^2$, $\sigma_z^2$. To apply EKF, we discretize the target dynamics with a time step $\delta t$. Then the filter's state is described by the equation

$$\boldsymbol{s}(k) = \Phi(k-1)\boldsymbol{s}(k-1) + G(k-1)\boldsymbol{w}(k-1) \, , \tag{29}$$

where the vector $\boldsymbol{s}(k) = [\hat{x}(k) \;\; \hat{y}(k) \;\; \hat{z}(k) \;\; \dot{\hat{x}}(k) \;\; \dot{\hat{y}}(k) \;\; \dot{\hat{z}}(k)]^\top$ represents the target's states estimate, $\boldsymbol{w}(k) = [n_x(k) \;\; n_y(k) \;\; n_z(k)]^\top$ represents the process noise vector, and

$$\Phi(k) = \begin{bmatrix} \mathbb{I}_{3\times3} & \mathbb{O}_{3\times3} \\ \mathbb{O}_{3\times3} & \delta t \mathbb{I}_{3\times3} \end{bmatrix}, \; G(k) = \begin{bmatrix} 1/2\delta t^2 \mathbb{I}_{3\times3} \\ \delta t \mathbb{I}_{3\times3} \end{bmatrix}$$

are the state transition and noise input matrices.

### B. Observation Description.

The observation models are sensor specific and have variable sizes. In the case of sensor $S_i$ is a radar, the observation model is given in $3D$ spherical coordinate system $\{r, \, \theta, \, \phi\}$ as

$$\begin{aligned} r_i^m &= r_i + b_i r_i + n_{ri} \\ \theta_i^m &= \theta_i + n_{\theta i} \\ \phi_i^m &= \phi_i + n_{\phi i} \, , \end{aligned} \tag{30}$$

where superscript $m$ indicates the measured values of the corresponding quantities, $n_{ri},\ n_{\theta i}\ \ n_{\phi i}$ are assumed to be zero-mean, Gaussian distributed, and uncorrelated processes with variances $\sigma^2_{ri},\ \sigma^2_{\theta i},\ \sigma^2_{\phi i}$, and $b_i$ is assumed to be an unknown constant [41]. This measurements relate to the target's coordinates in the sensor frame $F_i$ as

$$\begin{aligned} r_i &= \sqrt{x_i^2+y_i^2+z_i^2} \\ \theta_i &= tan^{-1}\left(\frac{y_i}{x_i}\right) \\ \phi_i &= tan^{-1}\left(\frac{z_i}{r_{ih}}\right), \quad r_{ih}=\sqrt{x_i^2+y_i^2}\,, \end{aligned} \tag{31}$$

and $[x_i\ y_i\ z_i]^\top = R_i^\top\left([x\ y\ z]^\top - \boldsymbol{p}_i^s\right)$. To take into account the effects of unknown constants $b_i, i = 1,\ldots,N_r$ in the case of radar sensors, the vector $\hat{\boldsymbol{b}}(k) = [\hat{b}_1(k)\ \ \ldots\ \ \hat{b}_{N_r}(k)]^\top$ representing the unknown constants estimates is appended to the filter's state $\boldsymbol{s}(k)$ as $[\boldsymbol{s}^\top(k)\ \ \hat{\boldsymbol{b}}^\top(k)]^\top$. The process noise vector $\boldsymbol{w}(k)$ is augmented accordingly with a vector $[n_{b_1}(k)\ \ \ldots\ \ n_{b_{N_r}}(k)]^\top$ representing the driving noises for the estimates of the unknown constants, which are assumed to be zero-mean, Gaussian distributed, and uncorrelated processes with variances $\sigma^2_{b_1},\ \ldots,\ \sigma^2_{bN_r}$. The state and noise input matrices are augmented by diagonal blocks $\mathbb{I}_{N_r\times N_r}$ and $\delta t\mathbb{I}_{N_r\times N_r}$, thus resetting the filter's dimension to $n = 6 + N_r$. With a slight abuse of notations we denote the filter's augmented state and process noise vectors by $\boldsymbol{s}(k)$ and $\boldsymbol{w}(k)$, and state and noise matrices by $\Phi(k)$ and $G(k)$ respectively. With this augmentation the process noise covariance matrix is defined as

$$\Gamma = diag\left(\sigma_x^2,\ \sigma_y^2,\ \sigma_z^2,\ \sigma^2_{b_1},\ \ldots,\ \sigma^2_{bN_r}\right).$$

In the case of camera sensor $S_i$, the observation model is given in $2D$ image coordinate system $\{x_p,\ y_p\}$ as pixel measurements

$$\begin{aligned} x^m_{pi} &= x_{pi} + n_{pxi} \\ y^m_{pi} &= y_{pi} + n_{pyi}\,, \end{aligned} \tag{32}$$

where superscript $m$ indicates the measured values of the corresponding quantities, $n_{pxi},\ n_{pyi}$ are assumed to be zero-mean, Gaussian distributed, and uncorrelated processes with variances $\sigma^2_{pxi},\ \sigma^2_{pyi}$. This pixel coordinates relate to the target's position in the sensor frame $F_i$ as

$$\begin{aligned} x_{pi} &= -\frac{x_i}{z_i}\frac{f_i}{dp_i} + \frac{W_i}{2} \\ y_{pi} &= -\frac{y_i}{z_i}\frac{f_i}{dp_i} + \frac{H_i}{2}\,, \end{aligned} \tag{33}$$

where $f_i,\ dp_i,\ W_i,\ H_i$ are the camera focal length, pixel size, camera width and camera height in pixels respectively, and $[x_i\ y_i\ z_i]^\top = R_i^\top\left([x\ y\ z]^\top - \boldsymbol{p}_i^s\right)$.

The rest of the paper will discuss target tracking using only radar and camera sensors data. The cases of other sensors such as LIDAR, sonar, etc. can be handled similarly. Since the sensor models (30) - (31) and (32) - (33) are nonlinear in target's position, we linearize them using the best available estimate at each time instance these sensor's data arrive to the selected by migration algorithm computing center.

### C. Measurements Grouping.

Let at time instance $k$ the data of $n(k)$ sensors $S_{i_1},\ldots,S_{i_{n(k)}}$ become available to the estimator, first $n_f(k)$ of which are fast data, and $n_s(k) = n(k) - n_f(k)$ are slow data with latencies computed according to the previous section. Here we emphasize that the number of available sensor data sets is dynamically changing by furnishing it with time argument. Let $n_f^v(k)$ stets of available fast data are valid, that is contain useful information about the target's position. Without loss of generality we assume that first $n_r^v(k)$ sets are valid radar measurements and next $n_c^v(k) = n_f^v(k) - n_r^v(k)$ sets are valid camera measurements, and compose an observation vector $\boldsymbol{h}^f(k,x,y,z)$ stacking up the valid in-time measurements, that is

$$\boldsymbol{h}_f(k,x,y,z) = \left[\boldsymbol{y}_1^\top(k)\ldots\boldsymbol{y}^\top_{n_{fv}(k)}(k)\right]^\top \tag{34}$$

$$\boldsymbol{y}_i^\top = \left[r_i^m(k)\ \ \theta_i^m(k)\ \ \phi_i^m(k)\right],\ i=1,\ldots,n_r(k),\ \ \boldsymbol{y}_i^\top = \left[x^m_{pi}(k)\ \ y^m_{pi}(k)\right],\ i = n_r(k)+1,\ldots,n_{fv}(k)\,.$$

Similarly, we compose the valid fast measurements error covariance matrix as

$$R_f(k) = blockdiag\left(R_1^m \dots R_{n_{fv}(k)}^m\right)^\top$$
$$R_i^m = diag\left(\sigma_{ri}^2\ \sigma_{\theta i}^2\ \sigma_{\phi i}^2\right),\ i = 1,\dots,n_r(k),\ \ R_i^m = diag\left(\sigma_{pxi}^2\ \sigma_{pyi}^2\right),\ i = n_r(k)+1,\dots,n_{fv}(k)\,. \tag{35}$$

On the other hand, the valid delayed measurements are grouped based on the number of delayed time steps. Let there be $n_s^v(k)$ valid measurements with $d_i, i = 1,\dots,n_s^v(k)$ seconds delays respectively. The corresponding delay steps are defined as $k_i^d = round(d_i/\delta t), i = 1,\dots,n_s^v(k)$. If $k_i^d = 0$, we count the corresponding measurements for fast ones, therefore $k_m^d >= k_i^d >= 1$ for all $i$, where $k_m^d$ denotes the largest delay step. Let there be $n_g(k)$ group of valid delayed measurements with $n_i^g(k), i = 1,\dots,n_g(k)$ measurements in each group such that $n_1^g(k) + \dots + n_{n_g}^g(k) = n_s^v(k)$ and $k_1^d < k_2^d < \dots < k_{n_g(k)}^d <= k_m^d$. In each group of $n_i^g(k)$ delayed measurements there may be readings from both type of sensors, the first $n_{ri}(k)$ sets of which are assume to be radar data and the next $n_{ci}(k) = n_i^g(k) - n_{ri}(k)$ sets valid camera data. The delayed for $k_i^d$ steps observations that arrive at time $k$ are denoted as $\boldsymbol{h}_i^s(k, k_i^d, x, y, z)$ and are defined with corresponding error covariance matrices $R_i^s(k, k_i^d)$ similar to (34) and (35) respectively for each $i = 1,\dots,n_g(k)$ .

**D. Filter Implementation.**

At time $k$, the prediction steps of the state estimate and error covariance matrix are implemented as usual

$$\begin{aligned} \hat{\boldsymbol{s}}(k|k-1) &= \Phi(k)\hat{\boldsymbol{s}}(k-1|k-1) \\ P(k|k-1) &= \Phi(k)P(k-1|k-1)\Phi^\top(k) + Q(k)\,, \end{aligned} \tag{36}$$

where $Q(k) = G(k)\Gamma(k)G^\top(k)$, but the correction steps depends on the available sensor data.

Let $n_r^v(k)$ fast radar measurements, $n_c^v(k)$ fast camera measurements, $n_{ri}(k)$ delayed radar measurements, and $n_{ci}(k)$ delayed camera measurements with delay steps $k_i^d$ for each $i = 1,\dots,n_g(k)$ be available at time $k$.

First we process the fast measurements. The target's predicted position at time $k$,

$$\hat{\boldsymbol{s}}_p(k|k-1) = [\hat{x}(k|k-1)\ \ \hat{y}(k|k-1)\ \ \hat{z}(k|k-1)]^\top$$

is expressed in each senor's frame using corresponding rotation matrices as

$$\boldsymbol{p}_t^{s_i} = \mathbb{R}_i^\top\left(\hat{\boldsymbol{s}}_p(k|k-1) - \boldsymbol{p}_i^s\right)\,. \tag{37}$$

The fast radar measurements predictions are computed using the equations

$$\begin{aligned} \hat{r}(k) &= \sqrt{\left(p_{t_1}^{s_i}\right)^2 + \left(p_{t_2}^{s_i}\right)^2 + \left(p_{t_3}^{s_i}\right)^2} \\ \hat{\theta}(k) &= \arctan 2\left(p_{t_2}^{s_i}, p_{t_1}^{s_i}\right) \\ \hat{\phi}(k) &= \arctan 2\left(p_{t_3}^{s_i}, \sqrt{\left(p_{t_1}^{s_i}\right)^2 + \left(p_{t_3}^{s_i}\right)^2}\right)\,. \end{aligned} \tag{38}$$

Thus at time $k$, the $i-th$ radar measurement prediction is

$$\hat{\boldsymbol{y}}_i^r(k) = \left[(1 + \hat{b}_i(k|k-1))\hat{r}(k)\ \ \hat{\theta}(k)\ \ \hat{\phi}(k)\right]^\top\,, \tag{39}$$

and the corresponding Jacobean is computed as

$$H_i^r(k) = \begin{bmatrix} (1+\hat{b}_i)\cos(\hat{\phi})\cos(\hat{\theta}) & (1+\hat{b}_i)\cos(\hat{\phi})\sin(\theta) & (1+\hat{b}_i)\sin(\hat{\phi}) \\ -\frac{\sin(\hat{\theta})}{\hat{r}\cos(\hat{\phi})} & \frac{\cos(\hat{\theta})}{\hat{r}\cos(\hat{\phi})} & 0 \\ -\frac{\sin(\hat{\phi})\cos(\hat{\theta})}{\hat{r}} & -\frac{\sin(\hat{\phi})\sin(\hat{\theta})}{\hat{r}} & \frac{\cos(\hat{\phi})}{\hat{r}} \end{bmatrix}\mathbb{R}_i + \begin{bmatrix} 0 & \dots & \hat{r} & \dots & 0 \\ 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & \dots & 0 \end{bmatrix}, \tag{40}$$

where time dependencies are omitted for the notation convenience. For the camera sensors without measurement delays, the predictions are computed as follows

$$
\begin{aligned}
\hat{x}_{pi}(k) &= -\frac{p_{t_1}^{s_i}}{p_{t_3}^{s_i}}\frac{f_i}{dp_i} + \frac{W_i}{2} \\
\hat{y}_{pi}(k) &= -\frac{p_{t_2}^{s_i}}{p_{t_3}^{s_i}}\frac{f_i}{dp_i} + \frac{H_i}{2} \\
\hat{\boldsymbol{y}}_i^c(k|) &= \left[\hat{x}_{pi}(k) \;\; \hat{y}_{pi}(k)\right]^\top ,
\end{aligned} \tag{41}
$$

and the corresponding Jacobean matrices are computed as follows

$$
H_i^c(k) = \frac{f_i}{dp_i}\begin{bmatrix} -\frac{1}{p_{t_3}^{s_i}} & 0 & \frac{p_{t_1}^{s_i}}{\left(p_{t_3}^{s_i}\right)^2} \\ 0 & -\frac{1}{p_{t_3}^{s_i}} & \frac{p_{t_2}^{s_i}}{\left(p_{t_3}^{s_i}\right)^2} \end{bmatrix}\mathbb{R}_i + \mathbb{O}_{2\times(3+N_r)}. \tag{42}
$$

Therefore the no-delay observation prediction and the corresponding Jacobean at time $k$ are

$$
\hat{\boldsymbol{h}}_f(k) = \begin{bmatrix} \hat{\boldsymbol{y}}_1^r(k) \\ \vdots \\ \hat{\boldsymbol{y}}_{n_r^v(k)}^r(k) \\ \hat{\boldsymbol{y}}_1^c(k) \\ \vdots \\ \hat{\boldsymbol{y}}_{n_c^v(k)}^c(k) \end{bmatrix}, \;\; H_f(k) = \begin{bmatrix} H_1^r(k) \\ \vdots \\ H_{n_r^v(k)}^r(k) \\ H_1^c(k) \\ \vdots \\ H_{n_c^v(k)}^c(k) \end{bmatrix}. \tag{43}
$$

We notice that at each time step $k$, observation vector $\boldsymbol{h}_f(k,x,y,z)$ and its prediction $\hat{\boldsymbol{h}}_f(k)$ have time dependent dimensions of $l(k) = 3n_r^v(k) + 2n_c^v(k)$, and the corresponding Jacobean $H^f(k)$ has dimensions of $l(k) \times n$.

The Kalman gain for the fast measurement processing is computed according to equation

$$
K_f(k) \;\; = \;\; P(k|k-1)H_f^\top(k)\left[H_f(k)P(k|k-1)H_f^\top(k) + R_f(k)\right]^{-1} , \tag{44}
$$

and has time varying dimensions of $n \times l(k)$, which corresponds to the $l(k)$ - dimensional residue $\boldsymbol{e}(k) = \boldsymbol{h}_f(k,x,y,z) - \hat{\boldsymbol{h}}_f(k)$. The concluding steps are updates of target's state estimate and the error covariance matrix

$$
\begin{aligned}
\hat{\boldsymbol{s}}(k|k) &= \hat{\boldsymbol{s}}(k|k-1) + K_f(k)\boldsymbol{e}(k) \\
P(k|k) &= \left[\mathbb{I}_{n\times n} - K_f(k)H_f(k)\right]P(k|k-1).
\end{aligned} \tag{45}
$$

Next we process the delayed measurements grouped as $\boldsymbol{h}_i^s(k,k_i^d,x,y,z)$, $i = 1,\ldots,n_g(k)$ corresponding to delay steps $k_i^d$ using Larsen's approach [21, 23]. First we compute the corresponding Jacobean matrices $H_i^s(k_i)$ and $H_i^s(k)$, $i = 1,\ldots,n_g(k)$ at the propagated filter state $\hat{\boldsymbol{s}}(k_i|k_i-1)$ and $\hat{\boldsymbol{s}}(k|k-1)$ according to equations (40) and (42) and stack them similar to (43). Here, we denote $k_i = k - k_i^d$. Then extrapolate the delayed measurements for $k_i^d$ steps using already computed estimates $\hat{\boldsymbol{s}}(k_i|k_i-1)$ and $\hat{\boldsymbol{s}}(k|k-1)$ as follows

$$
\boldsymbol{h}_i^e(k,x,y,z) = \boldsymbol{h}_i^s(k,k_i^d,x,y,z) - H_i^s(k_i)\hat{\boldsymbol{s}}(k_i|k_i-1) + H_i^s(k)\hat{\boldsymbol{s}}(k|k-1),\; i = 1,\ldots,n_g(k).
$$

The Larsen's correction term is computed as

$$
M_i^*(k) = \prod_{j=1}^{k_i^d}\left[\mathbb{I}_{n\times n} - K_f(k_i+j)H_f(k_j+j)\right]\Phi(k_i+j), \tag{46}
$$

which results in the Kalman gain

$$
K_i^*(k) = M_i^*(k)P(k_i|k_i-1)H_i^{s\top}(k_i)\left[H_i^s(k_i)P(k_i|k_i-1)H_i^{s\top}(k_i) + R_i^s(k_i)\right]^{-1}. \tag{47}
$$

for the delayed measurement incorporation, which is in fact the Kalman gain that could be computed at time instance $k_i$ if the measurements $\boldsymbol{h}_i^s(k, k_i^d, x, y, z)$ were available, pre-multiplied by $M_i^*(k)$ [23]. The state and error covariance matrix estimates at time $k$ are corrected based on $K_i^*(k)$ and the extrapolated delayed measurement $\boldsymbol{h}_i^e(k, x, y, z)$ as

$$\begin{aligned} \hat{\boldsymbol{s}}(k|k) &= \hat{\boldsymbol{s}}(k|k) + K_i^*(k)\left[\boldsymbol{h}_i^e(k, x, y, z) - H_i^s(k)\hat{\boldsymbol{s}}(k|k)\right] \\ P(k|k) &= P(k|k) - K_i^*(k)H_i^s(k)P(k_i|k_i - 1)M_i^{*\top}(k)\,. \end{aligned} \tag{48}$$

When there are no interim measurements, that is no fast measurements arrive at the computing center from time $k_i$ to $k$, then there are no interim Kalman gains, hence the Larsen's correction term reduces to

$$M_i^*(k) = \prod_{j=k_i+1}^{k} \Phi(j)\,, \tag{49}$$

and the Larsen's corrections (48) are applied to the open loop estimates of state $\hat{\boldsymbol{s}}(k|k)$ and error covariance matrix $P(k|k)$ , which are computed as follows

$$\begin{aligned} \hat{\boldsymbol{s}}(j|j) &= \hat{\boldsymbol{s}}(j|j-1) = \Phi(j)\hat{\boldsymbol{s}}(j-1|j-1) \\ P(j|j) &= P(j|j-1) = \Phi(j)P(j-1|j-1)\Phi^\top(j) + Q(j)\,. \end{aligned} \tag{50}$$

for all $j = k_i + 1, \ldots, k$.

# V. Simulation Results

We evaluate the approach in desktop simulations using synthetic data generated for a target UAV of a size of DJI M600 flying along a planned trajectory for the upcoming flight test at NASA Ames Research Center's DART facility. Target motion was simulated in Reflection environment and recorded at 100 $Hz$. The flight environment, path and used ground based infrastructure are displayed in Fig. 1.

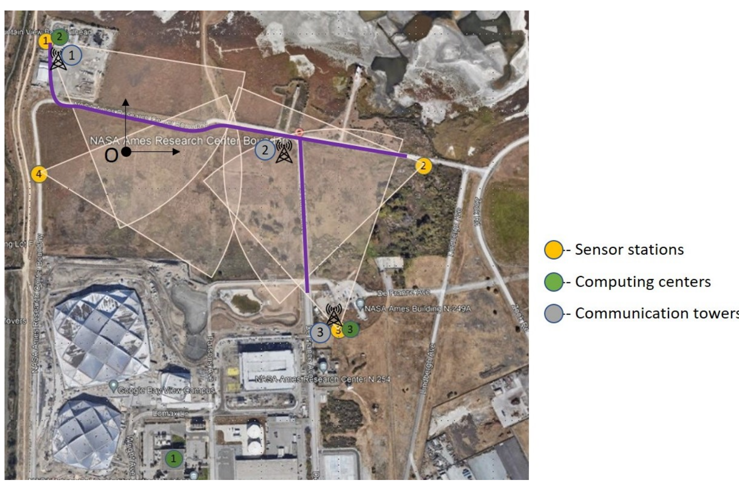
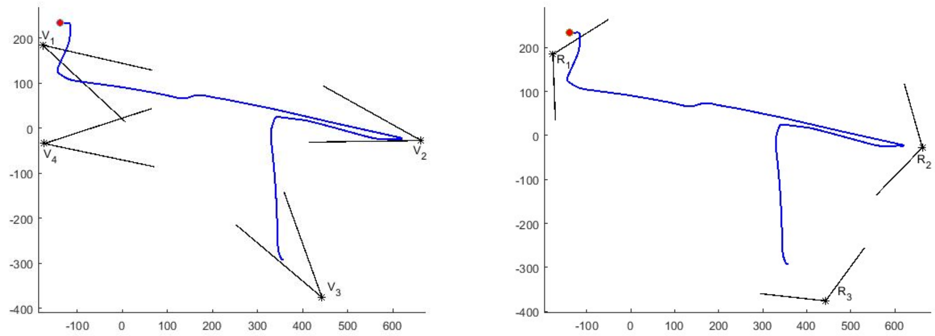



**Fig. 1 Target's flight path, Sensor stations, Computing canters and Communication towers.**

The recorded flight data are used to generate sensors readings using in-house developed radar and camera models with selected parameters and noise characteristics. Using Google Earth, we allocate three identical ground based radars and four identical ground based visual sensors (cameras) in the DART site, the locations, directions and field-of-views of which are displayed in Fig 2.

For this simulation study, we assume that the cameras provide target's information at a rate of 30 $Hz$ in the form of compressed frames of size 0.4 $Mbts$ and the radars provide information at a rate of 10 $Hz$ in the form of azimuth, elevation and range measurements of size 20 $Kbts$. According to sensors allocations in Fig. 1, visual sensors $V_1$, $V_3$ and radars $R_1$, $R_3$ are wired connected to communication towers $T_1$, $T_3$ respectively, and $V_2$, $V_4$ and $R_2$ have wireless communication with towers $T_1$, $T_3$ respectively.




**(a) cameras** **(b) radars**

**Fig. 2 Target's flight path, sensors locations, directions and fields of view**

| Camera parameters | | | | | | |
|---|---|---|---|---|---|---|
| $Resolution$ | $Size$ | $H - FOV$ | $F - length$ | $Visible\ size$ | $\sigma_{p_x}$ | $\sigma_{p_y}$ |
| [1920, 1080] | [7.68, 4.32] | $[-30°, 30°]$ | $14.33\ mm$ | $7\ pixel$ | 0.83387 | 0.4690 |
| Radar parameters | | | | | | |
| $H - FOV$ | $V - FOV$ | $Range$ | $\sigma_\phi$ | $\sigma_\theta$ | $\sigma_r$ | $b$ |
| $[-60°, 60°]$ | $[-40°, 40°]$ | [20, 2000] | 1° | 3° | 3.25 | 0.05 |
| Communication parameters | | | | | | |
| $\omega$ | $c$ | $p^s$ | $p^t$ | $B_{sys}$ | $\sigma^2$ | $g_0$ |
| $200\ Mbts$ | $299.79e6\ m/s$ | $0.2\ W$ | $0.5\ W$ | $5MHz$ | $8e - 13$ | $5.4250e - 04$ |
| Computation parameters | | | | | | |
| $\lambda_T$ | $\lambda_{C_1}$ | $\lambda_{C_2},\ \lambda_{C_3}$ | $\alpha_T$ | $\alpha_{C_1}$ | $\alpha_{C_2}$ | $\alpha_{C_3}$ |
| $2.2e6$ | $2.2e9$ | $2.2e9$ | 32 | $32 \times 4$ | $32 \times 4$ | $32 \times 1048$ |

**Table 1 Simulation Parameters.**

Two computing centers $C_2,\ C_3$ are laptops at sensors stations $S_1,\ S_3$, wired connected to communication tower $T_1$ $T_3$ respectively, and $C_1$ is the NASA super-computing center, which is wired connected to communication tower $T_3$. All simulation parameters are are presented in Table 1.

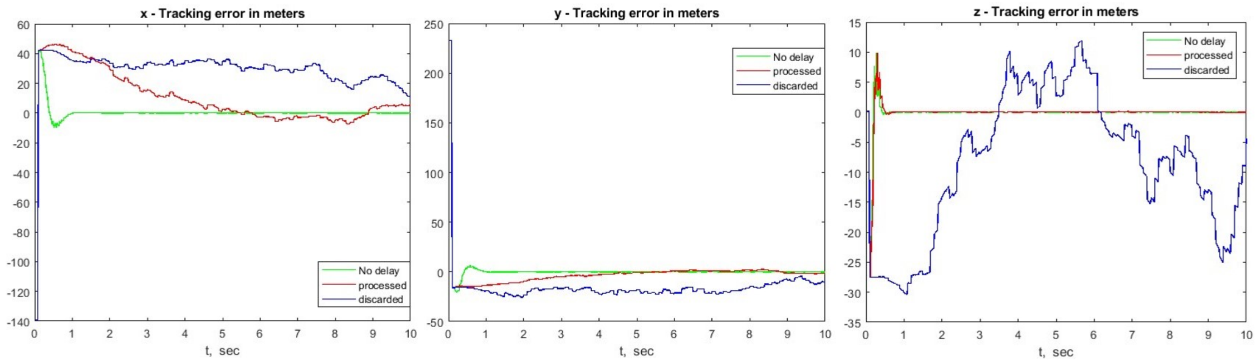
The maximum distance of reliable wireless transmission was set to 500 $m$ for this simulation, which results in the relay optimization selecting tower $T_2$ for transmitting data between the towers $T_1$ and $T_3$. The optimal data migration algorithm selects the computing center $C_1$ for the target tracking execution with the information delays presented in Table 2 as number of time steps.

| $V_1$ | $V_2$ | $V_3$ | $V_4$ | $R_1$ | $R_2$ | 3 |
|---|---|---|---|---|---|---|
| 3 | 2 | 0 | 4 | 0 | 0 | 0 |

**Table 2 Sensors data latency in time steps.**

It can be observed that although the radar data are generated at slower rate, but are received by the tracking algorithm practically without any delay. On the other hand, the camera data are generated at faster rate, but are received by the tracking algorithm with delays, except for the third one, which has fast wired connection with the selected computing center $C_1$.

We start simulation at time instance $k_0$ when the first radar measurement becomes available and set the filter position state at this radar measurement translated to the inertial frame, velocity state to zero, the unknown constant to zero,




**(a) x-error** **(b) y-error** **(c) z-error**

**Fig. 3 Target tracking errors during takeoff.**

which is the same for all radars, and the tracking error covariance matrix to $Q(k_0)$ with $\Gamma = diag\left(50^2,\ 50^2,\ 50^2,\ 0.05^2\right)$
.

For comparison purposes, we ran simulations for the ideal case setting all delays to zero, with discarded delayed measurements, and with processed delayed measurements. Table 3 represents root-mean-square values of tracking errors in $x,\ y,\ z$ directions and trace of the tracking error covariance matrix. Obviously, this values are smallest for the ideal case, get larger when the delayed measurements are ignored, and approach to the ideal case when they are processed.

| *Tracking case* | $RMS(e_x)$ | $RMS(e_y)$ | $RMS(e_z)$ | $RMS(P_{tr})$ |
|---|---|---|---|---|
| *No delayed measurements* | 2.3380 | 3.6198 | 0.3731 | 14.9456 |
| *Slow measurements discarded* | 5.9840 | 4.9586 | 4.1202 | 88.3366 |
| *Slow measurements processed* | 4.0234 | 3.7562 | 0.4136 | 18.3418 |

**Table 3 Root-mean-square values of the tracking errors and the trace of covariance matrix.**

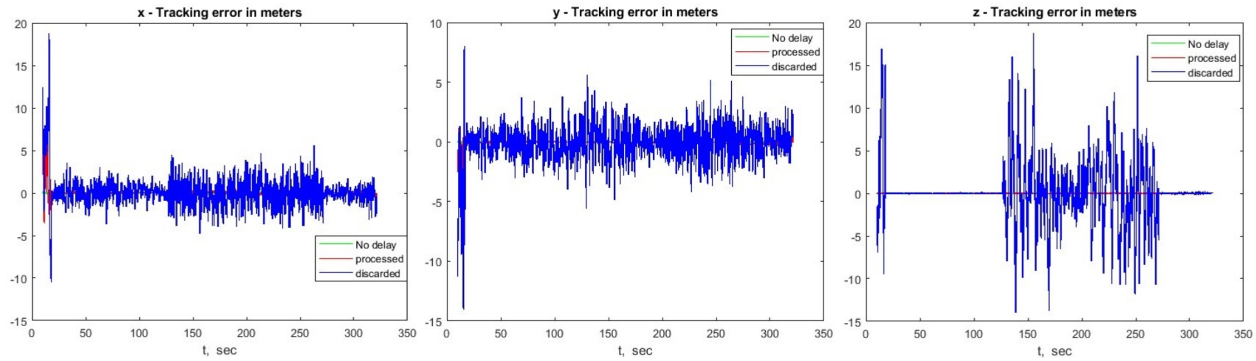
For better understanding of the tracking error evolution during the simulation stages, namely at the initial stage when the target takes off, when the EKF produces larger tracking errors due to inaccurate initialization, at the cruising stage and at the landing stage when the target descends almost vertically. The corresponding root-mean-square values are presented in Table 4. As we expected, initial large errors are decreased as the simulation progresses and are slightly increased at landing because of information richness loss.

| *Time inreval* | $RMS(e_x)$ | $RMS(e_y)$ | $RMS(e_z)$ | $RMS(P_{tr})$ |
|---|---|---|---|---|
| $0 < t \leq 10$ | 23.2430 | 21.8229 | 2.4052 | 52.7483 |
| $10 < t \leq 321$ | 0.4975 | 0.2005 | 0.0163 | 15.4578 |
| $321 < t \leq 331$ | 0.5317 | 0.4193 | 0.0079 | 26.2248 |

**Table 4 Error root-mean-square values for takeoff, cruise and landing stages.**

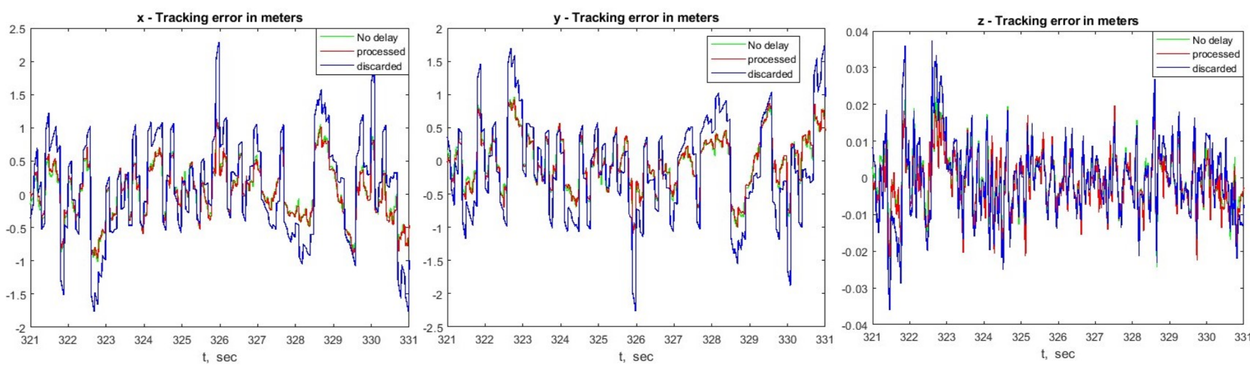
Figures 3, 4, and 5 represent tracking errors behaviors in details. It can be observed in Figure 3 that the large initialization errors quickly vanish for the ideal and processed delay cases. When the delayed measurements are ignored, the racking error converges slowly in $z$-direction, because the target is in the FOV of only sensors $V2$ (ignored), $R_2, R_3$, and the radar measurements are noisy in elevation ($z$- direction).

Figure 4 displays the tracking errors for the cruising flight, where the bursting at about 15 $sec$ can be attributed to targets turning maneuver, while large errors in $z$-direction for the delay discarded case result from the target going out of FOV of $V_3$, leaving only vertically noisy radar measurements for tracking. In other two cases visual measurements are able to drastically decrease the tracking errors.




**(a) x-error** **(b) y-error** **(c) z-error**

**Fig. 4 Target tracking errors during cruising.**




**(a) x-error** **(b) y-error** **(c) z-error**

**Fig. 5 Target tracking errors during landing**

Tracking of the target landing is displayed in Figure 5. The large errors observed in $x$ and $y$ direction can be attributed to lack of the useful information in the sensors readings in $x$ and $y$ direction.

# VI. Conclusion

We have presented an Extended Kalman Filter based framework for airborne target tracking using adaptive information fusion from multi-modal multi-rate distributed sensors network. To take into account the effects of information transmission delays, we first determine the best location for the tracking algorithm execution using an optimal data migration strategy. The associated delays for each sensor are used to determine the order of data fusion at the selected computing center. The fast (zero-delay) sensors information is processed by dynamically resizing the observation vector based on the validity and dimensions of received measurements, which respectively resizes the Jacobean and Kalman gain matrices in the implementation of EKF. The resulting filter state is updated based on the valid slow (delayed) data, which are grouped according to the delay-time steps before application of the Larsen's method. This approach is applied to synthetic sensor data generated by means of the ground based radar and camera models for the simulated target flight in Reflection simulation environment.

Future research will include mobile sensors, antennas and computing centers, which will bring more complexity in the network design from the perspective of optimal navigation and control of mobile assets to improve the tracking accuracy.

## Acknowledgment

This work was performed under the NASA Aeronautics Research Mission Directorate (ARMD), Transformative Tools and Technologies (TTT) Project's Autonomous Systems activity.

## References

[1] G. Fasano, D. Accado, A. Moccia, and D. Moroney. Sense and Avoid for Unmanned Aircraft Systems, *I*EEE Aerospace and Electronic Systems Magazine, Vol. 31, No. 11, 2016, pp. 82-110.

[2] B. Holmes, R. Parker, D. Stanley, L. Garrow, P. Masson and J. Olcott. NASA Strategic Framework for On-Demand Air Mobility. *Techreport, National Institute of Aerospace, Jan. 2017, http://www.nianet.org/ODM/reports/ODMStrategicFramework-Final170308.pdf*.

[3] A. Straubinger, R. Rothfeld, M. Shamiyeh, K.-D. Büchter, J. Kaiser, and K. O. Plötnera. An overview of current research and developments in urban air mobility – Setting the scene for UAM introduction, *J*ournal of Air Transport Management, Volume 87, August 2020, https://doi.org/10.1016/j.jairtraman.2020.101852.

[4] K. H. Shish, N. B. Cramer, G. Gorospe, T, Lombaerts, V. Stepanyan and K. Kannan. Survey of Capabilities and Gaps in External Perception Sensors for Autonomous Urban Air Mobility Applications, *I*n Proc. of the AIAA Scitech 2021 Forum, 11–15 & 19–21 January 2021, Virtual Event, https://doi.org/10.2514/6.2021-1114.

[5] C. Serres and M. Wu. Low C-SWaP Detect and Avoid: Defining Well Clear, MIT Lincoln Labs Air Traffic Control Workshop, Washington DC, November 2019, URL https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20200000623.pdf.

[6] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software. *New York: Wiley, 2001*

[7] G. Fasano, D. Accado, A. Moccia, and L. Paparone. Airborne Multisensor Tracking for Autonomous Collision Avoidance, *I*EEE 9th International Conference on Information Fusion, 2006, doi:10.1109/icif.2006.301724.

[8] K. Da , T. Li , Y. Zhu , H. Fan , Q. Fu. Recent advances in multisensor multitarget tracking using random finite set, *F*ront. Inf. Technol. Electron. Eng., 22 (1), 2021, pp. 5-24.

[9] L. Yan , B. Liu , D. Zhou. Asynchronous multirate multisensor information fusion algorithm, *I*EEE Trans. Aerosp. Electron. Syst., 43 (3), 2007, pp. 1135-1146.

[10] G. Fasano , D. Accardo , A. E. Tirri , A. Moccia , E. De Lellis. Radar/electro-optical data fusion for non-cooperative UAS sense and avoid, *A*erosp. Sci. Technol. Vol. 46, 2015, pp. 436-450 .

[11] Y. Bar-Shalom. Update with out-of-sequence measurements in tracking: exact solution, *I*EEE Trans. Aerosp. Electron Syst. 2012, 38(3), pp. 769-778.

[12] W. Koch. Tracking and Sensor Data Fusion: Methodological Framework and Selected Applications, *S*pringer: Berlin, 2014.

[13] M. Liu,T. Huai, R. Zheng and S. Zhang. Multisensor Multi-Target Tracking Based on GM-PHD Using Out-Of-Sequence Measurements, *S*ensors 2019, 19, 4315.

[14] A. Gopalakrishnan, N. S. Kaisare, and S. Narasimhan. Delayed and Infrequent Measurements in Extended Kalman Filter Based Nonlinear State Estimation. *Journal of Process Control, vol. 21, no. 1, pp. 119-129, 2011, https://doi.org/10.1016/j.jprocont.2010.10.013*.

[15] D. Simon. Kalman, H Infinity, and Nonlinear Approaches. *Wiley, Hoboken, NJ, 2006, pp. 274-279*.

[16] A. Gelb. Applied Optimal Estimation. *MIT Press, Cambridge, MA,1974, pp. 173-176.*.

[17] F. H. Hsiao and S.-T. Pan. Robust Kalman Filter Synthesis for Uncertain Multiple Time-Delay Stochastic Systems. *Journal of Dynamic Systems, Measurement, and Control, Vol. 118, No. 4, 1996, pp. 803-808. https://doi.org/10.1115/1.2802363*.

[18] E. Kaszkurewicz and A. Bhaya. Discrete-Time State Estimation with Two Counters and Measurement Delay. *Proceedings of 35th IEEE Conference on Decision and Control, Vol. 2, IEEE, New York, 1996, pp. 1472-1476. https://doi.org/10.1109/cdc.1996.572723*.

[19] V. Prasad, M, Schley, L. P. Russo, and B. W. Bequette. Product Property and Production Rate Control of Styrene Polymerization. *Journal of Process Control, Vol. 12, No. 3, 2002, pp. 353-372. https://doi.org/10.1016/S0959-1524(01)00044-0*.

[20] R. Li, A. B. Corripio, M. A. Henson and M. J. Kurtz. On-Line State and Parameter Estimation of EPDM Polymerization Reactors Using a Hierarchical Extended Kalman Filter. *Journal of Process Control, Vol. 14, No. 8, 2004, pp. 837-852. https://doi.org/10.1016/j.jprocont.2004.03.002*.

[21] T. D. Larsen, N. A. Andersen,O. Ravn, and N. K. Poulsen. Incorporation of TimeDelayedMeasurements in a Discrete-Time Kalman Filter. *Proceedings of the 37th IEEE Conference on Decision and Control, Vol. 4, 1998, pp. 3972-3977. https://doi.org/10.1109/cdc.1998.761918*.

[22] H. L. Alexander. State Estimation for Distributed Systems with Sensing Delay. *Data Structures and Target Classification, Vol. 1470, SPIE, Bellingham, WA, 1991, pp. 103–111. https://doi.org/10.1117/12.44843*.

[23] A. Comellini, D. Casu, E. Zenou, and C. Espinosa. Incorporating Delayed and Multirate Measurements in Navigation Filter for Autonomous Space Rendezvous. *Journal of Guidance, Control, and Dynamics,Vol. 43, No. 6, June 2020, pp. 1164-1172. https://doi.org/10.2514/1.G005034*.

[24] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal, Vol. 3, No. 5, October 2016, pages 637–646*.

[25] A. V. Dastjerdi and R. Buyya. Fog computing: Helping the Internet of Things realize its potential. *IEEE Computer, Vol. 49, No. 8, August 2016, pages 112-116, DOI: 10.1109/MC.2016.245*.

[26] P. Yang, N. Zhang, Y. Bi, L. Yu, and X. S. Shen. Catalyzing cloud-fog interoperation in 5G wireless networks: An SDN approach. *IEEE Network, Vol. 31, No. 5, September 2017, pages 14-20, DOI: 10.1109/MNET.2017.1600078*.

[27] P. Mach and Z. Becvar. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials, Vol. 19, No. 3, 2017, pages 1628–1656*.

[28] H. Yamanaka, E. Kawai, Y. Teranishi, and H. Harai. Proximity-aware IaaS in an edge computing environment with user dynamics. *IEEE Transactions on Network and Service Management, Vol. 16, No. 3, 2019, pages 1282–1296*.

[29] C. Yu, B. Lin, P. Guo, W. Zhang, S. Li, and R. He. Deployment and dimensioning of fog computing-based Internet of Vehicle infrastructure for autonomous driving. *IEEE Internet of Things Journal, Vol. 6, No. 1, Feb. 2019, pages 149–160*.

[30] X. Hou, Z. Ren, J. Wang, W. Cheng, Y. Ren, K.-C. Chen, and H. Zhang. Reliable Computation Offloading for Edge-Computing-Enabled Software-Defined IoV. *IEEE Internet of Things Journal, Vol. 7, No. 8, August 2020, pages 7097-7111, DOI: 10.1109/JIOT.2020.2982292*.

[31] A. Ullah, X. Yao, S. Shaheen, and H. Ning. Advances in position based routing towards ITS enabled fog-oriented VANET: A survey. *IEEE Trans. Intelligent Transportation Systems, Vol. 21, No. 2, Feb. 2020, pages 828–840*.

[32] V. Stepanyan, S. Schuet and K. S. Krishnakumar. An Approach to Reasoning Service Migration in Data and Reasoning Fabric (DRF) Implementation. *AIAA SciTech Forum, Jan 2022. https://doi.org/10.2514/6.2022-0787*.

[33] T. Lombaerts, K. H. Shish, G. Keller, V. Stepanyan, N. B. Cramer and C. A. Ippolito. Adaptive Multi-Sensor Fusion Based Object Tracking for Autonomous Urban Air Mobility Operations. *AIAA SciTech Forum, Jan 2022. https://doi.org/10.2514/6.2022-0362*.

[34] K. Miller. Communication Theories: Perspectives, Processes, and Contexts. *New York, NY, USA: McGraw-Hill, 2005*.

[35] L. Dai, B. Wang, Y. Yuan, S. Han, C. -L. I, and Z. Wang. Non-orthogonal multiple access for 5G: Solutions, challenges, opportunities, and future research trends. *IEEE Commun. Mag., vol. 53, no. 9, pp. 74–81, Sep. 2015*.

[36] S. Hu and G. Li. Dynamic Request Scheduling Optimization in Mobile Edge Computing for IoT Applications. *IEEE Internet of Things Journal, Vol. 7, No. 2, February 2020*.

[37] S. Jeong, O. Simeone, and J. Kang. Mobile edge computing via a UAV mounted cloudlet: Optimization of bit allocation and path planning. *IEEE Trans. Veh. Technol., vol. 67, no. 3, pp. 2049–2063, Mar. 2018*.

[38] T. X. Tran and D. Pompili. Joint task offloading and resource allocation for multi-server mobile-edge computing networks. *IEEE Trans. Veh. Technol., vol. 68, no. 1, pp. 856–868, Jan. 2019*.

[39] G. Gutin and A. P. Punnen. The Traveling Salesman Problem and Its Variations. *New York, NY, USA: Springer, 2007*.

[40] P. Vansteenwegen, W. Souffriau, G. V. Berghe, D. V. Oudheusden. A guided local search metaheuristic for the team orienteering problem. *European Journal of Operational Research, Vol. 196, pages 118–127, 2009, DOI: 10.1016/j.ejor.2008.02.037*.

[41] V. Stepanyan, T. Lombaerts, C. Dolph, N. B. Cramer and C. A. Ippolito. Estimation With Range Depended Sensor Model. *AIAA SciTech Forum, Jan 2022. https://doi.org/10.2514/6.2022-0494*.