Fusion of State Estimates Over Long-haul Sensor Networks with Random Loss and Delay

Qiang Liu, *Student Member, IEEE,* Xin Wang, *Member, IEEE,* and Nageswara S. V. Rao, *Fellow, IEEE*

***Abstract*—In long-haul sensor networks, remote sensors are deployed to cover a large geographical area, such as a continent or the entire globe. Related applications can be found in military surveillance, air traffic control, greenhouse gas emission monitoring, and global cyber attack detection, among others. In this work, we consider target monitoring and tracking using a long-haul sensor network, wherein the state and covariance estimates are sent from the sensors to a fusion center that generates a fused state estimate. Long-haul communications over submarine fibers and satellite links are subject to long latencies and/or high loss rates, which lead to lost or out-of-order messages. These in turn may significantly degrade the fusion performance: fusing fewer state estimates may compromise the accuracy of the fused state, whereas waiting for all estimates to arrive may compromise its timeliness. We propose an online selective linear fusion method to fuse the state estimates based on projected information contribution from the pending data. Using both prediction and retrodiction techniques, our scheme enables the fusion center to opportunistically make decisions on when to fuse the estimates, thereby achieving a balance between accuracy and timeliness of the fused state. Simulation results of a target tracking application show that our scheme yields accurate and timely fused estimates under variable communications delay and loss conditions.**

***Index Terms*—State estimation, long-haul sensor networks, delay and loss, online selective fusion, projected information gain, prediction and retrodiction.**

I. INTRODUCTION

Many networked sensor systems are deployed for detecting and/or monitoring the states of dynamic targets. In particular, long-haul sensor networks, where sensors span a large geographical area, can be found in many real-world applications, such as monitoring of greenhouse gas emissions using airborne and ground sensors [8], processing of global cyber events using cyber sensors distributed over the Internet [15], space exploration using a network of telescopes [20], and target detection and tracking for air and missile defense [4]. The response time requirements of such long-haul sensor networks can vary from a few seconds in detecting cyber attacks on critical infrastructures to years in detecting global trends in greenhouse gas emissions. In this work, we focus on a particular class of long-haul sensor networks that are deployed to detect and track events and/or targets within a timescale of seconds.

Q. Liu and X. Wang are with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY, 11794 USA e-mail: {qiangliu, xwang}@ece.sunysb.edu.

Nageswara S. V. Rao is with Oak Ridge National Laboratory, Oak Ridge, TN, 37831 USA email: raons@ornl.gov.

We consider a long-haul sensor network wherein the sensors generate, in their field of view, state estimates (such as position and velocity) of certain targets, and send them to a remote fusion center that generates global state estimates. With perfect communications, i.e., without any communication loss and delay, the fused estimate would normally achieve an accuracy level far superior to those of the individual sensor estimates. However, this is hardly the case for state estimation over a long-haul sensor network, where the connection distances to the fusion center are in the range of several to tens of thousands miles. In these networks, the state estimates are sent via satellite links or a combination of submarine and terrestrial connections to a fusion center with round trip time (RTT) of hundreds of milliseconds or more. Fig. 1 illustrates the infrastructure of a satellite-based long-haul sensor network. The combination of long latency and high loss over such links [16] often leads to an insufficient number of state estimates reaching the fusion center, thereby affecting the quality of the fused estimate produced by the fusion center.

As with most sensing applications, there are two competing requirements for the fused estimates generated over long-haul sensor networks. First, the accuracy of the fused estimate must exceed that of any single sensor and the estimation error should also be below a predefined maximum tolerable threshold. This generally requires the incorporation of most, if not all, sensor data into the fused estimate. On the other hand, the fused estimate must be generated within a certain, often tight, deadline. For instance, in some military applications, the position of an aircraft or missile is required to be reported within a few seconds. Unfortunately, over long-haul links with severe loss and delay, data from sensors may even fail to arrive at the fusion center by the reporting deadline.

Faced with these challenges, the fusion center can take two types of "extreme" actions. If it simply combines only the data that have arrived, the quality of the fused state is often suboptimal. On the other hand, waiting for all the data to arrive could compromise the timeliness requirement. Accounting for the trade-offs between estimation accuracy and reporting timeliness, in this work, we design an *online selective fusion* scheme that enables the fusion center to opportunistically make decisions on when to fuse the estimates to achieve a balance between the two.

Our study has a few salient features compared to the related works (See Section II):

(1) This work is among the very first to address state estimation with severe data loss/delay and strict enforcement on estimation accuracy and timeliness. The delay performance has often been overlooked in previous studies. The fusion

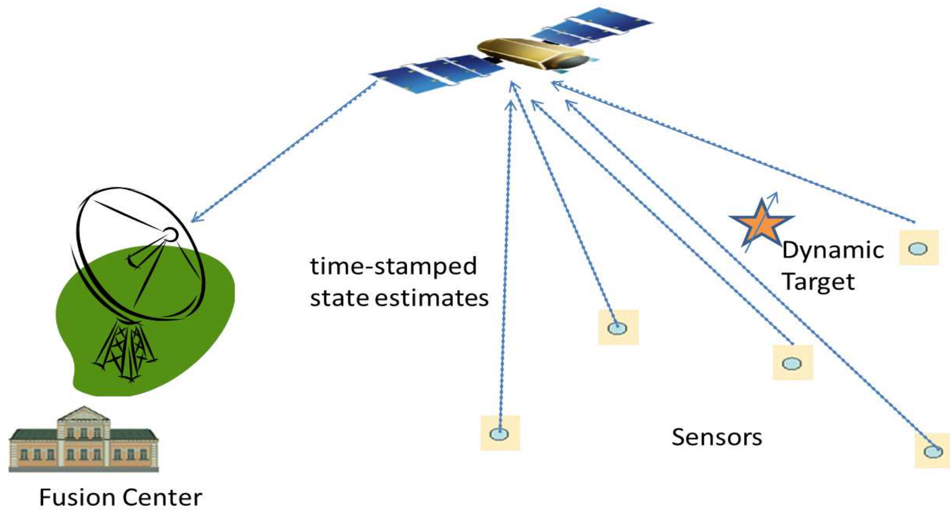


Fig. 1. Sensor estimate fusion over satellite links: the sensors generate their state estimates of the dynamic target(s), which are sent over long-haul satellite links to a remote fusion center for fusion.

center must balance the pros and cons of early versus late reporting to meet the dual requirements on timeliness and accuracy.

(2) A fusion center in long-haul state estimation applications often possesses sufficient storage and computation capabilities to handle large amounts of data sent from the sensors that can be queried and retrieved easily. However, from our performance study results, the storage/computation requirements on the fusion center are fairly minimal because the waiting period is often short following our information-gain-based fusion algorithm.

(3) In our study, prior statistical knowledge about delay and loss is not required. Although such information can be learned over a long period of time by the fusion center, and, once obtained, will prove to be useful, it is not a prerequisite of our selective fusion scheme. Therefore, our scheme can be applied more universally.

(4) We highlight the "online" nature of our scheme as the fusion center determines the best strategy on the go, based on its current accuracy and delay performance of the fused estimates. Combining this and the "selective" criteria whereby the fusion center determines the potential contribution from each missing estimate, our scheme works well even under a high degree of system uncertainty.

For the convenience of presentation, in this work we mainly consider tracking of a target with its dynamics being characterized by a linear system with zero-mean Gaussian measurement and process noise processes. Nevertheless, the ideas introduced here, especially the information-based online selective fusion, can be extended to somewhat more complex target models as well, with higher implementation costs. In addition, the results can be easily extended to the scenario when multiple distributed fusion centers are deployed in a long-haul sensor network.

The major contributions of this work are listed as follows:

- We study the impact of communication delay and loss on the accuracy of the fused estimate and provide the motivation for selective fusion;
- We propose a novel information metric that the fusion center can utilize for online selective fusion decisions based on the projected differential information contribution, which effectively reduces the reporting delay while meeting the accuracy requirement;
- We apply retrodiction to the selective fusion process so that the recovery of the missing information can be *proactively* expedited and the estimation accuracy improved within the same amount of time under variable degrees of data loss and delay;
- Although the loss and noise profiles are not essential to the online decision making process, we carry out some analysis of the estimation performance from known or learned link loss and delay statistics;
- We perform extensive simulations of a tracking example to demonstrate the effectiveness of our online scheme.

The rest of the paper is organized as follows. We briefly discuss a list of related works in Section II. The optimal fusion rule with fully received state estimates from the sensors is first reviewed in Section III, before we discuss design considerations for a selective fusion algorithm in Section IV. We then propose our selective fusion scheme based on the projected differential information gain metric and backward retrodiction in Section V and provide a complete algorithmic description. In Section VI, as an extension to our scheme, the notion of expected information contribution is discussed and its impact on fusion performance is explored. Simulation results are presented and analyzed in Section VII before we conclude the work in Section VIII.

II. Related Work

There has been growing research interest in state estimation and fusion under uncertainty. Fixed or relatively stable arrival delay can be easily handled by the technique of *state augmentation* [17], where adjacent states in time are grouped together to form a "super-state" with a higher dimension. Nevertheless, this can inflate the computation overhead significantly with longer delay. More importantly, with highly random arrival delay, it is difficult to effectively apply this approach as the dimension of the augmented state would keep changing. On the other hand, some literature studies have considered independent packet losses (i.e., a packet either arrives on time or is permanently lost) for one sensor-estimator. For example, in [6], an upper bound of the packet loss rate is derived above which the estimation error will go unbounded.

In multi-sensor state estimation problems, fusion schemes have been proposed under the condition that all packets arrive on time; see [7] and the references therein. The authors in [14] have attempted to address fusion by combining various sources of degradation (delay, loss, and packet drop) in a probabilistic manner. Their scheme calls for highly intensive computation to run on multiple sensors because of the high dimensionality of the augmented states on top of the already complicated one-sensor case. Besides, the underlying solution requires that the probabilities of different types of degradation are known a priori, which apparently is a very unrealistic assumption. In [11], a message-level retransmission mechanism is implemented, where under certain conditions, a sensor retransmits a message deemed as lost. At the cost of increased reporting delay, the chance that a message containing the state estimate is eventually delivered to the fusion center is improved over time. More recently, a staggered estimation scheduling scheme is proposed in [12] that aims to explore the temporal domain relationships of adjacent data within an estimation interval to improve the estimation and fusion performance.

There have been a series of studies addressing the out-of-sequence-measurement (OOSM) issues. In these problems, due to random latency, sensor data – often in the form of raw measurements – arrive out of order. One of their main focuses is on how to re-incorporate late arrivals. The initial one-step lag problem [1] has been extended to the multi-lag case [2], and the single-OOSM problem in [1], [2] has been extended to the multi-OOSM case [22] as well. Whereas a similar concept of "selective" information processing based on thresholding is proposed in [19], the focus is mainly on the re-incorporation of OOSMs (without loss) over the time domain, rather than information fusion from multiple sensors. Still, there are few multi-sensor studies under adverse link conditions, and the time-domain constraints – in the form of a reporting deadline – have not been accounted for in any of the above works.

Compared to the related studies, our design is not confined to a particular type of packet delay/loss. It is neither possible nor necessary that the fusion center can ascertain why a packet is missing. Instead, we focus on the impact of missing[1] packets on current estimation accuracy and timeliness, regardless of the delay/loss patterns. In this light, our approach is more of an online decision-making process under tight constraints on accuracy and delay.

This paper is an expansion of and an improvement upon the early version [10] in the following aspects: We now have more thorough discussions on the design considerations of selective fusion and the rules that we proposed, the latter of which are collectively shown here in an algorithmic format. Analytical studies are also carried out on the information gain probabilistically, where probabilities of delivering the state estimates by a certain reporting time are used to calculate the expected information contribution. In addition, we also compare the performance of our selective fusion scheme with two OOSM schemes in the literature. Simulation results demonstrate the advantages of our design in terms of improved reporting accuracy and latency performances over the other schemes.

III. OPTIMAL FUSION WITH FULLY RECEIVED STATE ESTIMATES FROM THE SENSORS

A. System Model and Kalman Filtering (KF) Basics

We consider the following multi-sensor discrete linear system (the subscript k and superscript i are time and sensor indices respectively):

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_k, \qquad E[\mathbf{w}_k\mathbf{w}_l^T] = \mathbf{Q}\delta_{k-l}, \tag{1}$$

$$\mathbf{y}_k^i = \mathbf{H}^i\mathbf{x}_k + \mathbf{v}_k^i, \qquad E[\mathbf{v}_k^i(\mathbf{v}_l^i)^T] = \mathbf{R}^i\delta_{k-l}, \tag{2}$$

where $\mathbf{F}$ is the state transition matrix and $\mathbf{H}$ is the measurement matrix. These matrices are often known from the underlying system and time-invariant in tracking applications. The vector $\mathbf{x}$ denotes the state of the target and $\mathbf{y}$ the sensor measurement. The process noise $\mathbf{w}$ and measurement noise $\mathbf{v}$ are white and independent (δ is the Kronecker delta function), whose covariances are $\mathbf{Q}$ and $\mathbf{R}$ respectively. While $\mathbf{Q}$ measures the internal system uncertainty, $\mathbf{R}$ measures the imperfect performance of the sensors as exhibited by errors in their measurements. Note that here these noise statistics are considered as time-invariant[2].

The well-known Kalman filter (KF) operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying dynamic system state. At sensor i, the filter evolves recursively according to the following set of equations:

$$\hat{\mathbf{x}}_{k|k-1}^i = \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1}^i \tag{3}$$

$$\mathbf{P}_{k|k-1}^i = \mathbf{F}\mathbf{P}_{k-1|k-1}^i\mathbf{F}^T + \mathbf{Q} \tag{4}$$

$$\mathbf{K}_k^i = \mathbf{P}_{k|k-1}^i(\mathbf{H}^i)^T(\mathbf{H}^i\mathbf{P}_{k|k-1}^i(\mathbf{H}^i)^T + \mathbf{R}^i)^{-1} \tag{5}$$

$$= \mathbf{P}_{k|k}^i(\mathbf{H}^i)^T(\mathbf{R}^i)^{-1} \tag{6}$$

$$\hat{\mathbf{x}}_{k|k}^i = \hat{\mathbf{x}}_{k|k-1}^i + \mathbf{K}_k^i(\mathbf{y}_k^i - \mathbf{H}^i\hat{\mathbf{x}}_{k|k-1}^i) \tag{7}$$

$$\mathbf{P}_{k|k}^i = (\mathbf{I} - \mathbf{K}_k^i\mathbf{H}^i)\mathbf{P}_{k|k-1}^i \tag{8}$$

$$= ((\mathbf{P}_{k|k-1}^i)^{-1} + (\mathbf{H}^i)^T(\mathbf{R}^i)^{-1}\mathbf{H}^i)^{-1} \tag{9}$$

In these equations, $\hat{\mathbf{x}}_{k|k-1}$ and $\hat{\mathbf{x}}_{k|k}$ denote respectively the a priori and a posteriori estimate at time k. These notations also apply to $\mathbf{P}$, the *error covariance matrix* of the estimate, defined as $\mathbf{P}_k^i = E[(\hat{\mathbf{x}}_k^i - \mathbf{x}_k)(\hat{\mathbf{x}}_k^i - \mathbf{x}_k)^T]$. Each of the diagonal element in $\mathbf{P}$ measures the mean-square-error (MSE) of the corresponding element state estimate. Eqs. (3)–(4) form

[1] We use "missing" here since an unavailable message could either be lost or delayed.

[2] An analysis on bounds of errors under varying noise statistics can be similarly derived as in [18].

the *prediction* step and Eqs. (5)–(9) the *correction* step. The Kalman filter gain matrix $\mathbf{K}$ determines the relative weights of the historical data and a new measurement. In Eq. (9), $\mathbf{P}^{-1}$ is often called the *information matrix*. Kalman filters are minimum-mean-square-error (MMSE)-optimal as the trace of $\mathbf{P}$ – characterizing the estimation error – at each step is minimized. A more thorough discussion of the Kalman filter basics can be found in [17].

B. Optimal Fusion of Fully Received Local State Estimates

Let us focus on the information form of the KF correction step Eq. (9). From the equation, we have

$$\mathbf{J}_k^i \triangleq (\mathbf{P}_{k|k}^i)^{-1} - (\mathbf{P}_{k|k-1}^i)^{-1} = (\mathbf{H}^i)^T(\mathbf{R}^i)^{-1}\mathbf{H}^i, \tag{10}$$

where $\mathbf{J}_k^i$ is defined as the *information gain matrix* from the sensor i at time k. This $\mathbf{J}$ matrix measures the increase of the information matrix $\mathbf{P}^{-1}$, and indirectly measures the reduction of error $\mathbf{P}$. Similarly, we can define

$$\mathbf{j}_k^i \triangleq (\mathbf{P}_{k|k}^i)^{-1}\hat{\mathbf{x}}_{k|k}^i - (\mathbf{P}_{k|k-1}^i)^{-1}\hat{\mathbf{x}}_{k|k-1}^i = (\mathbf{H}^i)^T(\mathbf{R}^i)^{-1}\mathbf{y}_k^i \tag{11}$$

as the *information gain vector* – the difference between the prior and posterior *weighted* estimates. It can be shown that $\mathbf{J}_k^i$ of a real system is semi-positive definite; besides, it is also stable since both $\mathbf{H}$ and $\mathbf{R}$ are time-invariant[3].

In multi-sensor fusion, the updates from the KFs run by the individual sensors are sent to the fusion center so that the global fusion can be performed. If we define the above parameters for the fusion center in a similar manner ("*G*" denotes "global"), the optimal fusion rule is simply to add up the information gain terms from the n sensors during the correction step:

$$\begin{aligned}\mathbf{J}_k^G &\triangleq (\mathbf{P}_{k|k}^G)^{-1} - (\mathbf{P}_{k|k-1}^G)^{-1}\\ &= \sum_{i=1}^{n}\left((\mathbf{P}_{k|k}^i)^{-1} - (\mathbf{P}_{k|k-1}^i)^{-1}\right), \qquad (12)\\ \mathbf{j}_k^G &\triangleq (\mathbf{P}_{k|k}^G)^{-1}\hat{\mathbf{x}}_{k|k}^G - (\mathbf{P}_{k|k-1}^G)^{-1}\hat{\mathbf{x}}_{k|k-1}^G\\ &= \sum_{i=1}^{n}\left((\mathbf{P}_{k|k}^i)^{-1}\hat{\mathbf{x}}_{k|k}^i - (\mathbf{P}_{k|k-1}^i)^{-1}\hat{\mathbf{x}}_{k|k-1}^i\right). \qquad (13)\end{aligned}$$

Such results have been shown in works such as [5] and [21]; and in [3], this is named the centralized measurement fusion (CMF), as the estimate fusion here is in effect equivalent to the centralized scheme in which the fusion center has received all the raw measurements.

In our setting, the state estimates and their corresponding error covariances, both prior and posterior, are sent to the fusion center. Since the raw measurement data usually come in much larger volumes, sending them directly to the fusion center not only consumes more bandwidth but may also renders transmission more prone to the link delay and loss.

[3]This time-invariance serves as the basis for our selective fusion algorithms in Section V, as the fusion center initially needs to use this steady information gain to determine the potential contribution of a missing packet.

Combining the above global correction step with the global prediction step, which has the very same form as in the one-sensor case (Eqs. (3) and (4)), we have the evolution of $\mathbf{P}^G$ and $\hat{\mathbf{x}}^G$ at the fusion center as follows:

$$\hat{\mathbf{x}}_{k|k-1}^G = \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1}^G \tag{14}$$

$$\mathbf{P}_{k|k-1}^G = \mathbf{F}\mathbf{P}_{k-1|k-1}^G\mathbf{F}^T + \mathbf{Q} \tag{15}$$

$$(\mathbf{P}_{k|k}^G)^{-1} = (\mathbf{P}_{k|k-1}^G)^{-1} + \sum_{i=1}^{n}\left((\mathbf{P}_{k|k}^i)^{-1} - (\mathbf{P}_{k|k-1}^i)^{-1}\right) \tag{16}$$

$$\begin{aligned}(\mathbf{P}_{k|k}^G)^{-1}\hat{\mathbf{x}}_{k|k}^G &= (\mathbf{P}_{k|k-1}^G)^{-1}\hat{\mathbf{x}}_{k|k-1}^G\\ &+ \sum_{i=1}^{n}\left((\mathbf{P}_{k|k}^i)^{-1}\hat{\mathbf{x}}_{k|k}^i - (\mathbf{P}_{k|k-1}^i)^{-1}\hat{\mathbf{x}}_{k|k-1}^i\right) \qquad (17)\end{aligned}$$

These equations constitute the optimal global fusion rule, again, when all the estimates generated by the sensors are successfully received by the fusion center.

IV. Selective Fusion: Design Considerations

With the remote sensors' state estimates being fully received by the fusion center, the estimation error of the fused estimate is often much lower than that of the state estimates provided by individual sensors. However, severe delay and loss inherent over the long-haul links may significantly limit the fusion gain. In this section, we propose selective fusion as a capable solution to balance the dual requirements of reporting accuracy and timeliness.

With incomplete data, Eqs. (16) and (17) can be rewritten as

$$(\mathbf{P}_{k|k}^G)^{-1} = (\mathbf{P}_{k|k-1}^G)^{-1} + \sum_{i=1}^{n}\mathbb{I}_k^i\mathbf{J}_k^i, \tag{18}$$

$$(\mathbf{P}_{k|k}^G)^{-1}\hat{\mathbf{x}}_{k|k}^G = (\mathbf{P}_{k|k-1}^G)^{-1}\hat{\mathbf{x}}_{k|k-1}^G + \sum_{i=1}^{n}\mathbb{I}_k^i\mathbf{j}_k^i, \tag{19}$$

where $\mathbb{I}_k^i = \{0,1\}$ is the indicator function that describes whether the actual packet sent by Sensor i for time k – which is denoted by P_k^i – is delivered to the fusion center on time and hereby contributes to the final fusion. One option is that the fusion center simply ignores those missing packets. As a result, with incomplete observation, fewer than n terms are incorporated in Eqs. (18)-(19) during the correction step and the resulting a posteriori $\mathbf{P}$ is higher – in terms of the elevated diagonal elements in P – than that in the full-observation case. Alternatively, the fusion center can substitute one- or multi-step predicted values – that is, the a priori estimates – for the missing data. However, the effect is exactly the same as that of simply ignoring the missing packets: $\mathbf{P}_{k|k}^i = \mathbf{P}_{k|k-1}^i$ and $\hat{\mathbf{x}}_{k|k}^i = \hat{\mathbf{x}}_{k|k-1}^i$ leading to $\mathbf{J}_k^i = 0$ and $\mathbf{j}_k^i = 0$, respectively (the same can be said for multi-step predictions). Consequently, prediction alone is equivalent to having zero information gain, and may cause the error variance of (some elements of) the state estimates to shoot up within a short amount of time if there are multiple missing packets. Also the unavailability of certain components in (18)-(19) results in sub-optimality of the fuser [3].

Fortunately, the fusion center is often allowed to delay its reporting, up till the reporting deadline, by waiting for the delayed data to arrive. The reporting deadline D_{max} is often set in such a way to reflect the worst-case delay performance the system could tolerate. The fusion center can certainly hold off the finalization of the global estimate for an earlier time till all the packets from that time have arrived; however, the average waiting time could well approach D_{max} whenever there is a single packet loss. For many applications, in which near real-time performance is called for, it is desirable for the fusion center to report its fused estimate as early as possible before the deadline. Not only does waiting passively for any lost and/or delayed packets significantly increase the reporting delay, but there is also a good chance that some of the packets being awaited carry little information to improve the existing level of estimation accuracy.

On the other hand, if the fusion center prematurely terminates the process of waiting for some of the packets with a higher potential to reduce the estimation error, the fused estimate obtained may deviate too much from the true state to be acceptable. Therefore, to maintain a decent accuracy level while not incurring a long reporting delay, it would be more viable for the fusion center to selectively wait for some missing packets before it decides to finalize the estimate of an earlier time instant. This can be stated in an alternate way: at any time step, the fusion center should decide whether each delayed packet is still "worth" waiting for. By making such online decisions for the pending data, the fusion center can dynamically balance the need for both reporting accuracy and timeliness to finalize the state estimate.

We note that such "*wait-versus-disregard*" decisions should be made online because they are largely determined by the current accuracy level at the fusion center. Generally speaking, if the current error covariance is higher, it is preferable that the fusion center waits longer for the recovery of those missing packets to achieve a higher confidence before sending out the final estimate. Suppose that the elevated error covariance due to an earlier missing packet P_k^i leads the fusion center to initially make a "waiting" decision, as other packets are subsequently received (including the packets from Sensor i itself but generated at different time instants other than k and those sent by other sensors), the information loss from P_k^i may be offset by the gain from these other packets. At a certain point, even when the said packet is still missing, the fusion center may decide to discontinue the wait.

For any missing packet, only by further observing the subsequent arriving packets can the fusion center *opportunistically* decide the cost vs. benefit of deferring its final reporting for one more time step[4]. In the next section, we propose a selective fusion scheme that enables the fusion center to make its fusion decisions based on the current level of estimation accuracy.

A final note is in place before we end this section: Although we loosely use "information gain" and "error reduction" interchangeably as the same concept, quantitative calculations should always follow the latter because in the KF evolution the trace of the error variance matrix is minimized, *not* that the trace of the information matrix is maximized.

[4]This can be regarded as one instance of the principle of *diminishing information return.*

V. Selective Fusion With Projected Differential Information Contribution (PRODIC) and Retrodiction

The fusion center apparently cannot predict whether a missing packet will eventually be received; it can, however, *project* the packet's past information gain to the current time and decide whether the potential information contribution to the current time still warrants further waiting for the missing packet. In other words, the information gain past due is measured from the perspective of the current time so that the decrease of information gain as time progresses, due to the arrival of other packets, is accounted for.

In this section, we propose using an information metric to guide the fusion center through the selective waiting and fusion process. In particular, we combine forward projected information gain and backward retrodiction so that the fusion center can obtain an accurate estimate much faster.

A. PRODIC: the Information Metric

For notational simplicity, we define a function h that links two successive a posteriori $\mathbf{P}$ together:

$$h(\mathbf{X}, \mathbf{Y}) \triangleq (\mathbf{F}\mathbf{X}^{-1}\mathbf{F}^T + \mathbf{Q})^{-1} + \mathbf{Y}. \quad (20)$$

Then from Eqs. (15) and (16), we have

$$\mathbf{P}_{k|k}^{-1} = h(\mathbf{P}_{k-1|k-1}^{-1}, \mathbf{J}_k). \quad (21)$$

We name the information metric Projected differential information contribution (PRODIC). It measures the potential information contribution of a delayed packet should it return now. The following steps calculate $\Delta\mathbf{P}_{k,k-d}^i$ at time k, which is the PRODIC of the missing packet from Sensor i with the time-stamp $k-d$, that is, P_{k-d}^i:

Step 1: Add the information gain[5] $\mathbf{J}_{k-d}^i$ of the missing packet to the information matrix $\mathbf{P}_{k-d}^G$:

$$(\mathbf{P}_{k-d,temp}^G)^{-1} = (\mathbf{P}_{k-d}^G)^{-1} + \mathbf{J}_{k-d}^i; \quad (22)$$

The "temp" in this and the following equations denotes that the associated $\mathbf{P}^G$ is only updated temporarily to obtain the PRODIC of the missing packet which has not actually arrived.

Step 2: Recursively propagate the change of $\mathbf{P}_{k-d}^G$ in Step 1, through the intermediate steps, to the current time k. From time $T_n = k-d+1$, $k-d+2$, ..., up to k, calculate

$$(\mathbf{P}_{T_n,temp}^G)^{-1} = h\left((\mathbf{P}_{T_n-1,temp}^G)^{-1}, \mathbf{J}_{T_n}^G\right); \quad (23)$$

In this step, all the existing $\mathbf{J}^G$ values of these intermediate time steps remain the same.

Step 3: Calculate the differential information gain.

$$\Delta\mathbf{P}_{k,k-d}^i = \mathbf{P}_k^G - \mathbf{P}_{k,temp}^G. \quad (24)$$

After the recursion in Step 2 proceeds to the current time k, Eq. (24) measures the difference between $\mathbf{P}_{k,temp}^G$ – the

[5]From now on, the conditions in the subscripts are dropped for simplicity since all the variables are considered a posteriori: e.g., $k|k$ is shortened to k.

updated $\mathbf{P}_k^G$ with the *supposed arrival* of the missing packet – and the current $\mathbf{P}_k^G$. Note that $\mathbf{P}_k^G$ has been calculated according to the existing arrivals; no matter how small the original information gain from the missing estimate is, the aggregate error in $\mathbf{P}_{k,temp}^G$ will be better than that in $\mathbf{P}_k^G$. In other words, $\Delta\mathbf{P}_{k,k-d}^i \succ 0$.

The lag d is in general a random variable that depends on the actual arrival history of the recent data; however, in the following analysis, we consider it as the *existing* lag of the next global estimate to be obtained. The PRODIC for each delayed packet of time $k-d$ is calculated separately. At any time instant, it is computationally expensive to consider all the possible packet arrival patterns across multiple time steps for all the currently missing messages; there are as many as 2^{dn} different patterns in a d-lag enumeration. By singling out each sensor's contribution, we have reduced the complexity from a brute-force search to a linear order dn. After considering the PRODIC of one missing packet, the fusion center has found the least information to be gained among all the possible arrival patterns that include at least this particular pending packet. Therefore, the PRODIC is actually a conservative measure of the potential information gain from awaiting a particular pending packet.

Following the above calculation, the fusion center should compare the PRODIC value with a cutoff threshold th. If the PRODIC value is larger than the threshold, the fusion center still considers the information carried by the packet important for reducing the estimation error and will continue to wait for it. As long as the reporting deadline has not been reached, the pending global estimate for a time instant will be finalized only when the fusion center decides not to wait for any of the pending packet which contains the corresponding sensor estimate.

Determining the value of the threshold th is again a process of balancing the dual requirements of high reporting accuracy and low latency. In general, the higher the existing error variance is, the more potential information gain can be expected from the same missing packet. The threshold thus in principle should adapt according to varying levels of $\mathbf{P}^G$. For the convenience of implementation, however, it is better to normalize the threshold to a fixed value. We choose the desired proportion of error reduction by the fusion center as the threshold at any time step, so that

$$\frac{tr(\Delta\mathbf{P}_{k,k-d}^i)}{tr(\mathbf{P}_k^G)} = 1 - \frac{tr(\mathbf{P}_{k,temp}^G)}{tr(\mathbf{P}_k^G)} > th \tag{25}$$

implies the packet P_{k-d}^i can potentially improve the current $\mathbf{P}_k^G$ more than the threshold level and thus will be awaited.

The online decisions are largely affected by the availability of the packets from the sensors with better accuracy guarantees. When the fusion center has received all or most packets from these sensors, further improvement from the missing ones becomes increasingly small and thus unnecessary beyond a certain point. On the other hand, with the data from these better sensors missing, $\mathbf{P}^G$ can quickly inflate, thereby elevating the normalized PRODIC of these packets to a higher level; what often ensues is the decision to continue waiting for these missing packets. Hence, with heterogeneous sensors, a decision for a sensor with higher accuracy (e.g., "to wait") often overrules that from another one with lower accuracy (e.g., "to disregard") when the two decisions are contradictory.

B. Information Gain from Retrodiction

From the last subsection, sensors that yield more accurate estimates usually predominate the fusion center's selective waiting decisions. Because of their larger potential information gain, the fusion center generally has to wait longer in case an estimate sent from any of these sensors is delayed. In order to guarantee timely reporting, it is critical for the fusion center to reduce such passive waiting.

Estimation of a target state at a particular time based on measurements collected beyond that time is generally called retrodiction (or smoothing). Traditionally, an earlier *existing* estimate is retrodicted using subsequent measurements so that its accuracy is improved [17]. In this work, we propose a novel use of retrodiction to proactively *interpolate* intermediate missing data. Once one or more subsequent packets of an unavailable one have been received, the fusion center uses them to retrodict the preceding missing one. While waiting for missing packets, the fusion center applies retrodiction backward, from the current time k to time $k-d$ whose globe estimate is to be reported next, for *all* the packets – including the available ones – in between. This idea is illustrated in Fig. 2, where the current lag is $d = 2$. There are a total of three sensors. Since applying prediction-only estimates results in a higher error variance, to reduce the performance degradation, the fusion center may decide to wait for the delayed packet while applying retrodiction scheme if subsequent packets arrive. Note both available and unavailable estimates are retrodicted during waiting. The retrodiction process is always run backward to the time instant for the next pending global estimate.

We apply the fixed-interval Rauch-Tung-Streibel (RTS) retrodiction algorithm [17], which is known to be computationally efficient. Besides, since measurements do not appear in the equations, the algorithm is especially suited for our scenario in which state estimates rather than raw data are sent directly to the fusion center. The following iterative steps propagate the newly gained information due to the on-time arrival of packet P_k^i backward to time $k-d$.

Step 0: Initialize the backward smoother.

$$\mathbf{P}_{k,retr}^i = \mathbf{P}_{k|k}^i; \tag{26}$$

$$\hat{\mathbf{x}}_{k,retr}^i = \hat{\mathbf{x}}_{k|k}^i; \tag{27}$$

From time $T_n = k$, $k-1$, ..., up to $k-d$, recursively calculate the values through the following three steps:

Step 1: calculate the backward smoothing gain

$$\mathbf{G}_{T_n-1}^i = \mathbf{P}_{T_n-1|T_n-1}^i \mathbf{F}^T (\mathbf{P}_{T_n|T_n-1}^i)^{-1}; \tag{28}$$

Step 2: calculate the error $\mathbf{P}^i$ of the smoothed estimate

$$\begin{aligned}&\mathbf{P}_{T_n-1,retr}^i \\ =&\mathbf{P}_{T_n-1|T_n-1}^i - \mathbf{G}_{T_n-1}^i(\mathbf{P}_{T_n|T_n-1}^i - \mathbf{P}_{T_n,retr}^i)(\mathbf{G}_{T_n-1}^i)^T;\end{aligned} \tag{29}$$

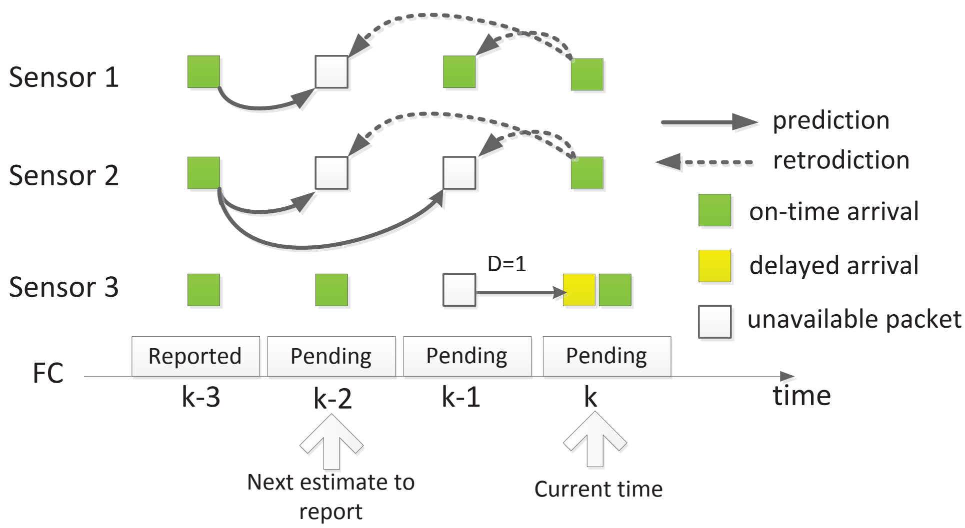


Fig. 2. Selective fusion example with three sensors and forward prediction and backward retrodiction

Step 3: find the smoothed estimate

$$\hat{\mathbf{x}}^i_{T_n-1,retr} = \hat{\mathbf{x}}^i_{T_n-1|T_n-1} + \mathbf{G}^i_{T_n-1}(\hat{\mathbf{x}}^i_{T_n,retr} - \hat{\mathbf{x}}^i_{T_n|T_n-1}). \quad (30)$$

In these equations, $\mathbf{P}^i_{retr}$ denotes the a posteriori $\mathbf{P}^i$ after retrodiction. The algorithm is applied to each sensor separately, so that the process is also in line with the packet-level PRODIC calculation. As mentioned earlier, retrodiction in our scheme has the dual benefits of improving the accuracy of existing estimates and interpolating missing ones; consequently, the process has its distinct features, some of which are absent from those conventional retrodiction studies.

(1) *Interpolation of the missing estimates and the chain effect.* Retrodiction is only meaningful when a later estimate has an accuracy level at least as good as that of an earlier estimate. According to the equations, an unavailable packet itself has no impact on its preceding estimate during RTS retrodiction (that is, retrodiction is only effective with the presence of an actual estimate); however, if this unavailable estimate has been retrodicted from a subsequently available one, it can in turn improve its preceding one as well. This can be regarded as the "chain effect" of retrodiction. As a result, a string of missing estimates can be improved by just one single estimate subsequent to them all. For example, in Fig. 2, at time k, having been retrodicted by packet P^2_k, P^2_{k-1} can further retrodict P^2_{k-2}.

(2) *Handling the a priori mismatch as a result of loss.* Both the a priori estimates and their error covariances appear in the RTS algorithm. In the full-observation case, sending a posteriori values is often enough, since the fusion center can always derive one-step predicted values for a priori values of the next step that are congruent with the predictions at the sensors. However, with missing data, the a priori values at the sensors and those at the fusion center may not match. The predicted values at the fusion center can be much more error-prone after multi-step losses. This has an effect on RTS retrodiction. For instance, when an actual estimate is received following at least a missing one, there exists a *mismatch* between the a priori values calculated from prediction and those (if any) sent directly by the sensors. We let the fusion center take the prior values from earlier prediction for the following two reasons: first, the a priori values may not actually be sent by the sensors in a real system or can be lost separately from the a posteriori data; second, the predicted values at the fusion center is a true reflection of accuracy level evolution associated with the global estimate.

(3) *Handling the delayed estimates.* What if the original estimate sent from a sensor arrives when the estimate has been interpolated from subsequent packet(s) but before the global estimate for the time of interest has been finalized? For simplicity, in our design, we have the original estimate replace the corresponding "partially retrodicted estimate(s)". Still better yet, the fusion center may reprocess the estimates so that both the original and its subsequent estimates are combined to yield a "fully retrodicted estimate", possessing an even better accuracy level than the original itself.

C. Selective Fusion Algorithm – PRODIC-RTS

Now we are ready to present the complete selective fusion algorithm, in which we have incorporated the RTS retrodiction into our recursive PRODIC calculation.

In a dynamic decision-making process like ours, there is more than one way to implement the selective fusion scheme. While we hope to improve both reporting accuracy and delay performances at the same time, the algorithm should be implemented as easily as possible. In our design, only arrivals from within the waiting window are considered; that is, delayed arrivals are disregarded if the corresponding global estimate has already been finalized and reported. Although it is possible that the fusion center still considers incorporating

Algorithm 1 Selective Fusion Algorithm with PRODIC and RTS Retrodiction

1: **Initialize**: At time k, the next estimate to be reported is $\hat{\mathbf{x}}^G_{k-d}$
2: **for** $T_n := d$ to 0 with decrement 1 **do**
3: Collect arriving packets $\mathcal{N}_k = \bigcup_{T=0}^{d} \mathcal{N}_{k,k-T}$.
4: Update $\mathcal{W}_{k,k-Tn}$ and $\mathcal{F}_{k,k-Tn}$:
5: $\mathcal{W}_{k,k-Tn} \leftarrow \mathcal{W}_{k-1,k-Tn} - \mathcal{N}_{k,k-Tn}$;
6: $\mathcal{F}_{k,k-Tn} \leftarrow \mathcal{F}_{k-1,k-Tn} \bigcup \mathcal{N}_{k,k-Tn}$;
7: **end for**
8: **for** $\forall i \in \mathcal{S}$ **do**
9: Calculate $\mathbf{P}^i_{k-d,retr}$ and $\hat{\mathbf{x}}^i_{k-d,retr}$ using Eqs. (26)-(30);
10: **for** $T_n := d$ to 0 with decrement 1 **do**
11: Calculate $\mathbf{J}^i_{k-T_n}$ using Eq. (10) (with retrodicted values);
12: $\mathbf{J}^G_{k-T_n} = \sum_{\text{packet } \mathsf{P}^i_{k-T_n} \in \mathcal{F}_{k,k-T_n}} \mathbf{J}^i_{k-T_n}$;
13: $(\mathbf{P}^G_{k-T_n|k-T_n})^{-1} = h((\mathbf{P}^G_{k-T_n-1|k-T_n-1})^{-1}, \mathbf{J}^G_{k-T_n})$;
14: **end for**
15: **end for**
16: **if** $\mathcal{W}_{k,k-d} = \emptyset$ or $d = D_{max}$ **then**
17: Finalize $\hat{\mathbf{x}}^G_{k-d}$ using Eqs. (17) with all the elements in $\mathcal{F}_{k,k-d}$ and $\mathbf{P}^G_{k-d}$;
18: $d \leftarrow d - 1$;
19: **if** $d > 0$ **then**
20: Go to Line 8;
21: **end if**
22: **else**
23: **for** $\forall j \in \mathcal{S}$ such that $\mathsf{P}^j_{k-d} \in \mathcal{W}_{k,k-d}$ **do**
24: Calculate $\Delta\mathbf{P}^i_{k,k-d}$ using Eqs. (22)-(24);
25: Calculate the difference between the normalized PRODIC and the threshold th $Diff^j_{k-d} = \frac{tr(\Delta\mathbf{P}^j_{k,k-d})}{tr(\mathbf{P}^G_k)} - th = 1 - \frac{tr(\mathbf{P}^{G,j}_{k|k,temp})}{tr(\mathbf{P}^G_k)} - th$;
26: **end for**
27: **if** $\forall i \in \mathcal{S}$, $Diff^i_{k-d} \leq 0$ **then**
28: $\mathcal{W}_{k,k-d} \leftarrow \emptyset$;
29: Go to Line 16;
30: **else**
31: $d = d + 1$;
32: **end if**
33: **end if**

delayed arrivals (e.g., up to D_{max}) in this scenario, doing so may increase the computational cost as the size of the historical data grows. In addition, the benefit of incorporating the estimates with such a long delay is often negligible: the PRODIC criterion has already dictated that the previously disregarded packets carried little information, and even more so after the global estimate has been finalized, due again to the diminishing information over time.

Algorithm 1 describes the selective fusion rule performed by the fusion center at time k for a global estimate at time stamp $k - d$ (i.e., the current lag is d). The following sets are defined:

- $\mathcal{S}$ is the set of all sensors;
- $\mathcal{P}_k$ is the set of all the packets with time stamp k;
- $\mathcal{N}_{k,k-d}$ is the set of new arrivals at time k with time-stamp $k - d$;
- $\mathcal{W}_{k,k-d}$ consists of all the packets with time-stamp $k - d$ that the fusion center is still expecting at time k;
- $\mathcal{F}_{k,k-d}$ is the set of packets with time-stamp $k - d$ that are ready at time k for fusion.

In addition, we let $\mathcal{W}_{k,k-d} = \mathcal{P}_{k-d}$ and $\mathcal{F}_{k,k-d} = \emptyset$ if $d < 0$. Apparently, if at time k, $\mathcal{W}_{k,k-d} = \emptyset$, the fusion center is ready to finalize $\hat{\mathbf{x}}^G_{k-d}$ using all the elements in $\mathcal{F}_{k,k-d}$. Besides, a packet appearing in $\mathcal{N}_{k,k-d}$ ($d > 0$) must have appeared in $\mathcal{W}_{k-1,k-d}$ too. Therefore, the selective fusion decision-making process evolves as follows. After updating newly received packets, the fusion center first retrodicts the past estimates using newly received data; afterward, it calculates the PRODIC of each packet in $\mathcal{W}_{k,k-d}$ and decide whether further waiting is necessary. Only when all the packets in the set have been received, or when the pending packets are deemed no longer worthy of further waiting can the fusion center finalize $\hat{\mathbf{x}}^G_{k-d}$.

In the algorithm, the instantaneous $\mathbf{P}^G$ can be calculated immediately, as shown in Lines 8-12, after the fusion center has incorporated the packets that have just arrived and has retrodicted the estimates. Afterward, the PRODIC of all the pending packets is computed and normalized for comparison with a cutoff threshold th (Lines 23-26). There are two possible scenarios when the next pending estimate can be finalized: all the packets have been received (Line 16) or all the remaining pending packets are no longer deemed necessary and will be disregarded (Lines 27-28). Besides, when one global estimate is finalized, the above procedure is repeated immediately for the next pending estimate (Line 21) – for a lag of $d - 1$ in this case – as more than one pending estimate may be finalized at the same time step.

Thanks to the information gain from retrodiction, the PRODIC of a missing packet becomes smaller and the fusion center can often terminate its waiting for the pending packets much earlier compared to the case without retrodiction. The level of improvement will be demonstrated via simulation studies in Section VII.

VI. Expected Information Gain with Link Loss and Delay Profiles

Our earlier discussions highlighted the proposed selective waiting and fusion algorithm as an online scheme since the fusion center makes its decisions based solely on the actual message arrival instances. In this section, we consider the scenario where the fusion center has some knowledge about the packet loss and delay statistics. The deterministic information gain from a pending estimate can then be extended to the expected information gain for different time periods. Substituting the expected information gain for the original will also result in different fusion performance as to be discussed below.

A. Loss and Delay Profiles

The packet loss and delay characteristics are largely determined by the long-term condition of the long-haul link. In a real system, even when the fusion center is initially oblivious to these characteristics, the empirical packet arrival patterns can be measured and recorded over time and thus approximate profiles of link loss and delay can be constructed. For ease of exposition, here we assume that each packet sent by a sensor is lost during transmission with probability p independently of other packets. A pdf $f(t)$ and the corresponding cdf $F(t) = \int_0^t f(u)\,\mathrm{d}u$ can model the overall delay t – a continuous random variable – that a packet experiences to be successfully delivered to the fusion center. Additionally, the loss and delay are regarded as two independent processes.

B. Expected Information Gain

If we let time zero denote the time of interest, that is, the time when the packet containing the state estimate is generated and sent out by a sensor, then the probability that this packet is delivered by time t is $(1-p)F(t)$. Consider a pending packet with a current lag of d steps – in other words, a delay of at least dT, where T is the estimation interval – and the probability that the packet will be delivered within the next interval $[dT, dT+T]$ can be calculated as

$$\begin{aligned} & p_{del,dT\to(d+1)T} \\ & \triangleq \Pr\{t \le dT+T | t > dT\} \\ & = \frac{\Pr\{dT < t \le dT+T\}}{\Pr\{t > dT\}} \\ & = \frac{(1-p)[F(dT+T)-F(dT)]}{1-(1-p)F(dT)}. \end{aligned} \tag{31}$$

Eq. (31) describes the probability that the information contribution from a pending estimate will be realized in the next estimation interval. Then the *expected* information contribution would be

$$\mathbf{J}_{k-d,E} = p_{del,dT\to(d+1)T}\mathbf{J}_{k-d}. \tag{32}$$

C. Effect of the Expected Gain on Selective Fusion

Eq. (32) is a statistical measure of the potential information gain in the first step of our selective fusion algorithm. We consider its effect for both heterogeneous and homogeneous link conditions.

1) Heterogeneous Link Loss and Delay Profiles: Since both the packet-level loss rate p and the delay distribution $F(t)$ appear in Eq. (32), any variations in either of them would cause a shift in the expected information gain even for the same value of $\mathbf{J}$. It is easy to show that for the same delay distribution, an increase in the packet loss rate would result in the decrease of the expected information gain at any step (that is, Eq. (31) monotonically decreases with p.); on the other hand, with the same loss rate, the expected information gain depends on the specific shape of the pdf $f(t)$. In our context, then, the fusion center would find it difficult to realize the potential information gain promptly for a link with a high loss rate and/or a long average arrival delay.

2) Homogeneous Link Loss and Delay Profiles: Even when the loss and delay statistics are homogenous across different communication links, differences in fusion performance still exist if the fusion center opts to use Eq. (32) as the potential information gain. In particular, since Eq. (31) effectively serves as a discount factor, the perceived information gain at each step becomes smaller than the original $\mathbf{J}$. Recall from the last section that the fusion center uses a cutoff threshold th to decide whether to continue waiting for a pending packet. As the information gain term to be plugged in the initial step becomes smaller, so does the one propagated to the current time. As such, for the same th, the fusion center may alter some of the would-be "wait" decision so that the packet is then disregarded. This would help reduce the reporting delay significantly at the cost of slightly higher estimation errors.

VII. Performance Evaluation

We evaluate the performance of our selective fusion algorithm through extensive simulations. We first introduce the target motion model, and then compare a few on-line fusion algorithms; and finally, we compare our scheme with two other OOSM fusion schemes that also exploit retrodiction. Despite our discussions on expected information gain in the last section, in our simulation studies, the fusion center does not assume any statistical knowledge about the loss and delay characteristics in its online decision making.

A. Simulation Setup

1) Target Motion Model: We consider tracking of a target whose motion follows the near-constant-acceleration model [9]; that is, the trajectory of the moving target follows Newtonian laws with independently incremented acceleration. In particular, we consider the white-noise jerk version of the model, in which the acceleration derivative (i.e., the "jerk") is an independent white noise process.

The target in general moves within the three-dimensional free space, but the trajectory can be mapped to orthogonal axes (e.g., the commonly used "east – north – up"). And here we single out the effect of one dimension (e.g., east) by mapping the trajectory onto this axis only. The target state then consists of the position r, velocity v, and acceleration a on this axis and the discretized state evolution is given by

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_k,$$

where

$$\mathbf{x}_k = \begin{bmatrix} r_k \\ v_k \\ a_k \end{bmatrix}, \mathbf{F} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix},$$

and

$$\mathbf{Q} = \mathrm{cov}(\mathbf{w}_k\mathbf{w}_k^T) = S_w \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix},$$

where S_w is the power spectral density of the continuous-time white noise, and T is the sampling/estimation period. Besides, the common measurement matrix at the individual sensors is

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

2) Parameters and Performance Metrics: The following table contains the list of our default parameters.

TABLE I
SIMULATION PARAMETERS DEFAULT SETUP

Parameter	Symbol	Value
sampling period (s)	T	0.5
process noise PSD ($\mathrm{m}^2/\mathrm{s}^3$)	S_w	0.5
loss rate	P_{loss}	0.1
normalized arrival delay	D_{arv}	3
reporting deadline	D_{max}	10
no. of sensors	n	3
measurement noise s.t.d. (m)	$\sqrt{R}$	50
PRODIC threshold	th	5%

As our default setup, there are a total of three sensors, whose measurement noise standard deviations are all 50 m. The process noise PSD is 0.5 $\mathrm{m}^2/\mathrm{s}^3$; the sampling/estimation period $T = 0.5$ s and the associated time measures are subsequently "normalized" relative to this time (without units); e.g., the normalized sampling time is 1, whereas the normalized deadline is set as 10 (which is really 5 s). The PRODIC threshold is set to 5%. Again, in our study, packet loss and delay are generated as two independent processes. While the packet loss is generated as an independent Bernoulli process, delays follow (memoryless) exponential distribution. The default loss rate and normalized arrival delay are set to be 10% and 3, respectively.

The loss and delay statistics for a *given* long-haul network should be fairly stable over a period of time (that's the very reason that they can be learned over time). Nevertheless, we are interested in studying the impact on fusion performance from variable loss and delay profiles, for instance, when different types of networks are compared with one another. We study the impact of each factor separately by varying its values while keeping all other parameters at their default values. In addition, although we expect a long-haul network is well designed for scalability, we study the scenario when the number of deployed sensors is small (no more than five) as long-haul sensors are generally expensive to deploy; also, the system is more vulnerable to failure and suboptimal performance with an insufficient number of sensors and thereby warrants special attention.

B. Comparison of Different Online Fusion Schemes

We compare the following schemes in our first group of study:

- Maximum waiting ("wait"): the FC finalizes the estimate after all missing estimates arrive or the reporting deadline is reached, whichever comes first;
- Selective fusion based on PRODIC but without retrodiction ("PRODIC"):
- Selective fusion based on PRODIC with modified RTS retrodiction[6] ("PRODIC-retr");
- We also consider the full-observation case ("ideal") as a baseline scenario for comparison with other schemes, in which the reporting delay is always zero and hence there is no retrodiction being implemented. In other words, this ideal scenario is not realistic since it always assumes perfect communications and instant data processing and reporting.

[6]This has been denoted as "PRODIC-RTS" earlier, which will be used in subsequent analysis as well.

Overall the goal is to reduce both the estimation MSE and reporting time, where the former is measured as the trace of the error covariance matrix.

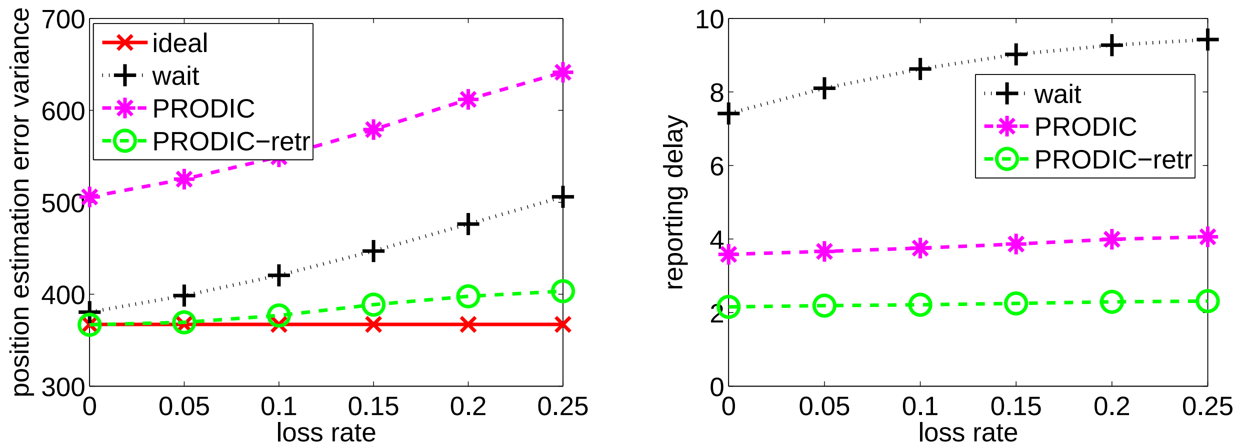
1) Loss Rate: The packet loss rate is varied from 0 to 0.25, and the results are shown in Fig. 3. With its average reporting time approaching the deadline, the "wait" scheme takes advantage of the extra waiting time to collect delayed estimates and thus reduces the estimation error[7]. The PRODIC scheme is seen to effectively reduce the reporting delay; however, the estimation error is still relatively high. Compared to other schemes, PRODIC-RTS is less sensitive to changes in the loss rate. At the highest loss rate 25% in our study, the position error variance increases only by about 7% from the zero-loss case. This demonstrates the effectiveness of our design, which exploits retrodiction to reduce the estimation error upon packet loss. One of the main reasons for this improvement has been shown earlier: as long as one most recent packet is received, all the preceding missing estimates can be retrodicted. Besides, the change in reporting delays as the loss rate increases is negligible (it stays slightly above 2). From this perspective, our PRODIC-RTS is robust to packet loss.

2) Arrival Delay: We vary the normalized packet arrival delay from 0 to 6, and observe similar trends in Fig. 4 as in the previous case, with some minor exceptions. When there is no arrival delay, no retrodiction is performed so that the estimates can be reported immediately; as the arrival delay goes up to 1, the error variance decreases following PRODIC-RTS thanks to retrodiction. And then, as the arrival delay continues to increase, the error variance also increases, though not quite as fast as in other schemes. As can be seen again, retrodiction has effectively reduced the estimation error when there are significantly long delays.

Although it may first seem surprising that the reporting delay is well below the average arrival delay (e.g., when the arrival delay equals 6, the reporting delay is about 2.5), the result is attributed to both the randomness of the arrival delay and the PRODIC-based selective fusion process. There exist packets whose arrival delay is smaller than that of others and hence can retrodict other missing ones comparatively faster. With the improved estimates following retrodiction, the fusion center may decide to terminate its waiting much earlier by disregarding all the remaining pending packets. In contrast, the reporting delay in the "waiting" case, even under moderate loss and delay profiles, almost always approaches the reporting deadline D_{max} due to the nearly constant presence of missing packets.

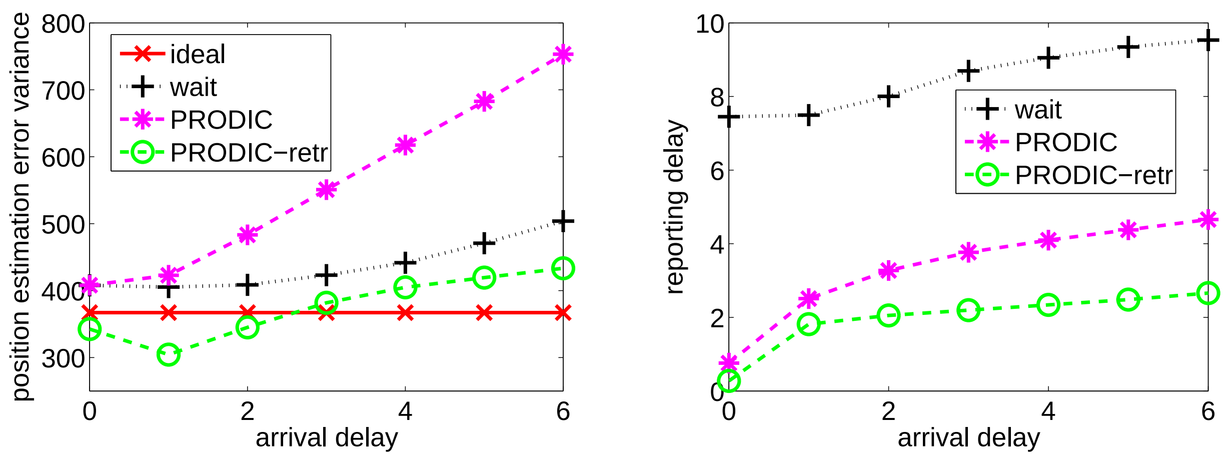
3) Number of Sensors: We vary the number of sensors from 1 to 5 to test its impact on fusion performance. From Fig. 5,

[7]The unit for this and all subsequent position error variances is m^2.



(a) (b)

Fig. 3. Loss rate vs. (a) position error variance and (b) normalized reporting delay



(a) (b)

Fig. 4. Normalized arrival delay vs. (a) position error variance and (b) normalized reporting delay

we observe that when the number of sensors is small, the effect of retrodiction is more prominent. Because the error variance is usually much higher with an insufficient number of sensors – at the risk of violating the maximum tolerable errors – the targeted potential information gain (as specified by PRODIC threshold) from a missing packet can be realized only by waiting longer so that the actual packet or subsequent packets following this missing one can be received. If there is only one sensor, all the gains from the retrodiction would benefit the final estimate because no other sources – that is, packets from other sensors – exist that can compensate for the information loss due to an unavailable packet. On the other hand, the relative information gain from retrodiction with an increasing number of sensors being present becomes smaller and the advantage of applying PRODIC-RTS in terms of improved accuracy diminishes.

As for the reporting delay, opposite trends are observed. The “waiting” scheme is subject to the increase of waiting time when more sensors are present, as there is a higher possibility that a packet is absent. In contrast, PRODIC-based schemes experience reduced waiting time, thanks to the selective waiting process, in which the information gain from a missing packet diminishes as more sensors contribute to the final fusion.

4) PRODIC Threshold: In the above simulations, we have kept the PRODIC threshold at a conservative 5%; that is, only when a pending packet can potentially reduce the current estimate error by more than 5% would the fusion center decide to wait for it. In reality, the fusion center can tune the threshold according to the current accuracy level. In Fig. 6, the threshold is set to vary from 0% (i.e., to wait for all pending packets) to 20%. The plots can be easily interpreted: When the threshold goes up, the requirement on each packet is relaxed, fewer packets need to be awaited, and reduced accuracy and reporting delay would follow; and vice versa. It is interesting to note that when the threshold is zero, the PRODIC scheme is reduced to the “waiting” case; PRODIC-RTS, on the other hand, does not have inflated waiting time thanks to the proactive nature of retrodiction.

C. Performance Comparison with OOSM Algorithms

We also compared our scheme with Al1 [2] and Al1-RTS [13] algorithms, both of which are designed to address

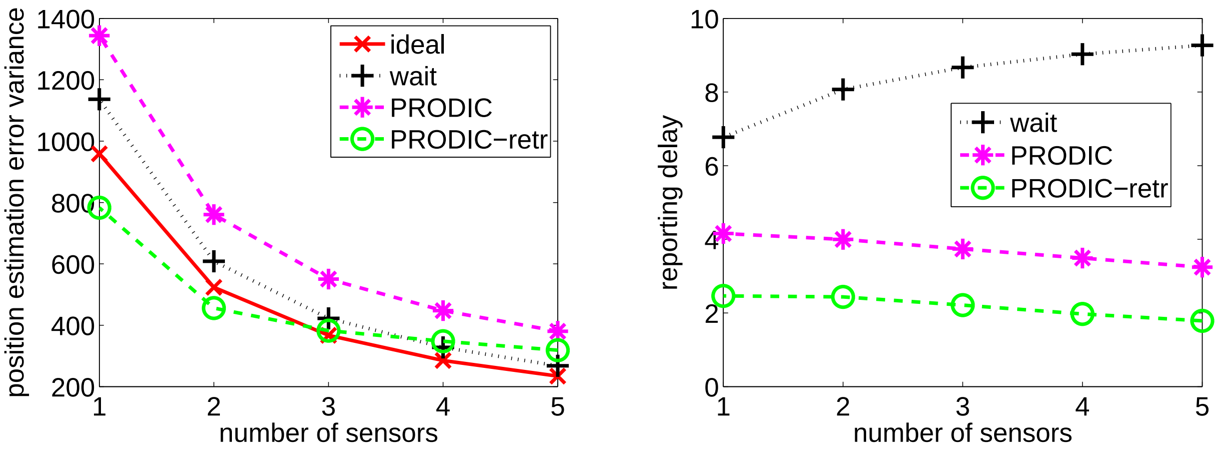


Fig. 5. Number of sensors vs. (a) position error variance and (b) normalized reporting delay

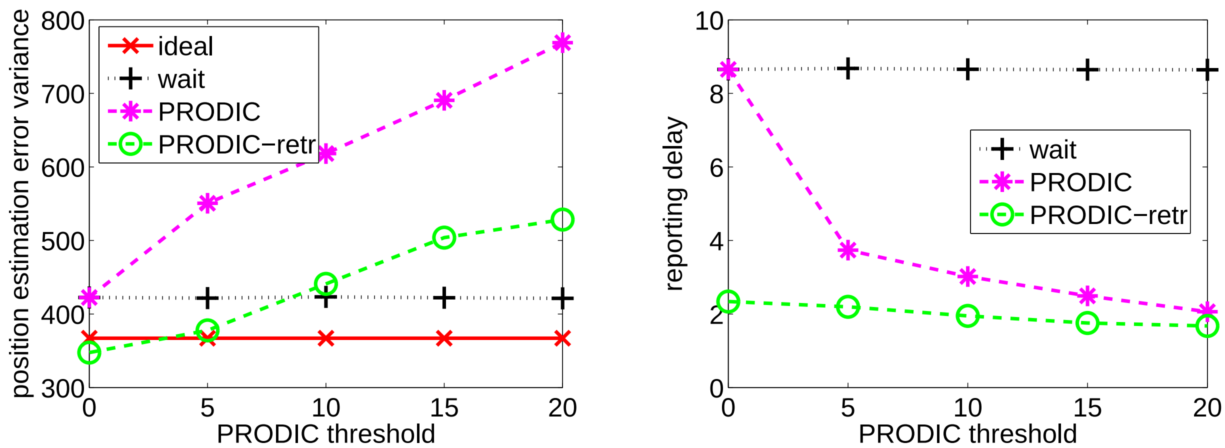


Fig. 6. PRODIC threshold vs. (a) position error variance and (b) normalized reporting delay

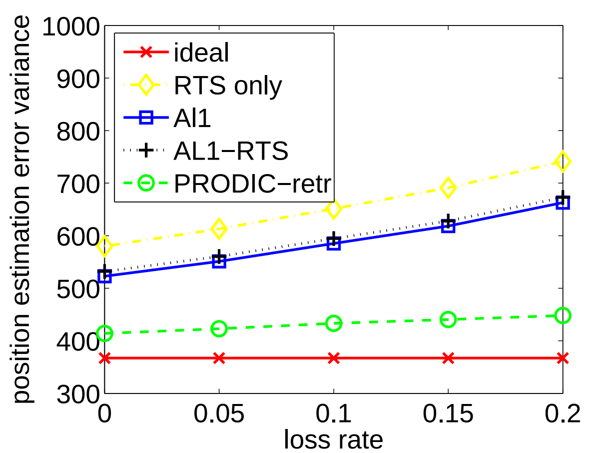


Fig. 7. Loss rate vs. position error variance (with normalized reporting delay = 2)

OOSM issues. The main feature of these algorithms is that retrodiction is performed only after an OOSM actually arrives and the goal is to correct only the current estimate. The difference within the two cited algorithms exists during the retrodiction process: while Al1 uses an "equivalent measurement" to perform one-step retrodiction, Al1-RTS applies the standard RTS retrodiction algorithm.

The original schemes are proposed under relatively simple circumstances: single sensor, single l-step delay, no data loss, and no reporting deadline, among others. However, it is easy to extend the schemes to multi-OOSM [22] and multi-sensor case with data loss and reporting deadline. Besides, we also allow non-zero reporting delay so that the correction after retrodiction applies to an intermediate step as well, thereby improving the accuracy performance of not just the current estimate. This can somewhat guarantee fair comparisons among the algorithms.

Because all the schemes share retrodiction as a common means to reduce the error variance of an earlier estimate, we also compare with an "RTS-only" scheme in which no OOSM is processed; that is, data are either received on time or smoothed with subsequent ones whenever available. We test how the position error variance varies with different parameters (loss rate, arrival delay, and number of sensors) with a *given* normalized reporting delay of two (since the

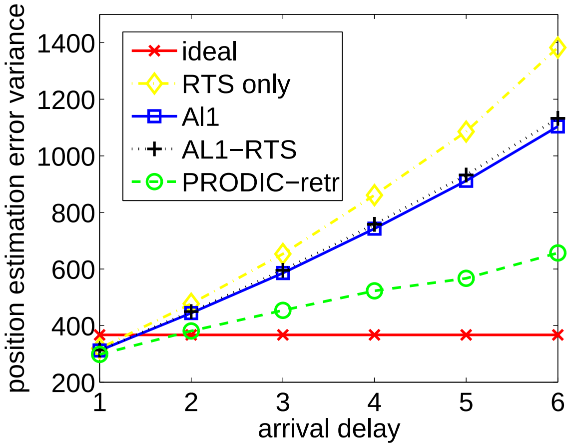


Fig. 8. Normalized average arrival delay vs. position error variance (with normalized reporting delay = 2)

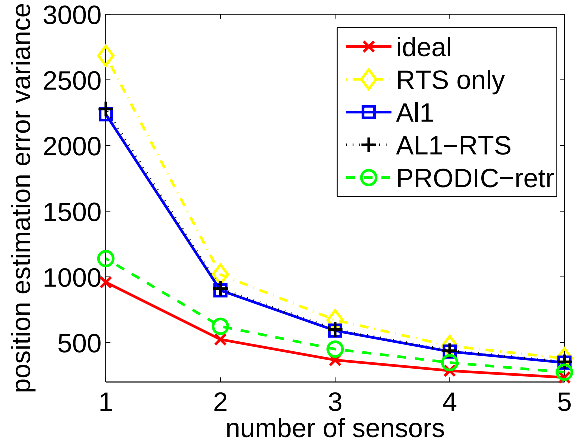


Fig. 9. Number of sensors vs. position error variance (with normalized reporting delay = 2)

reporting delay in other schemes needs to be pre-assigned to run the algorithms). Note that the thresholds have been adjusted in our PRODIC-RTS scheme to guarantee the same reporting delay.

1) Loss Rate: From Fig. 7, as the loss rate increases, the increase in the position estimation error of PRODIC-RTS is less significant compared to others. When the loss rate is 20%, the error variances of both A*l*1 and A*l*1-RTS are 50% more than that of PRODIC-RTS. On the other hand, while the two OOSM algorithms reduce the error variance by about 10% compared to the case where OOSMs are simply ignored ("RTS only"), our scheme reduces the error by about 40%. The reason is fairly simple: In other schemes, retrodiction is performed only when the missing packets actually arrive later, which may not be effective when the loss rate is high, resulting in larger estimation errors.

2) Arrival Delay: As can be seen in Fig. 8, with the increasing mean arrival delay, the position error variance in PRODIC-RTS does not shoot up as much as those in other schemes. When the arrival delay is 4, for example, A*l*1 already has its error variance nearly 50% higher than that of PRODIC-RTS; as it further reaches 6, the error variances of the two OOSM schemes are more than 75% higher. Similar to the case with varying loss rates, as the arrival delay increases, the fusion center has to wait longer to perform retrodiction. Meanwhile, the reporting deadline is fixed, allowing fewer estimates to be effectively retrodicted in time for the final reporting, thus leading to a higher error variance.

3) Number of Sensors: Again, we can see from Fig. 9 that the benefit of retrodiction is more significant when the number of sensors is small for all cases. With fewer sensors, all the other schemes suffer from much higher error variances than PRODIC-RTS. Our scheme benefits from the proactive retrodiction where the correction doesn't have to occur only after an OOSM actually arrives. The other schemes must wait significantly longer to perform retrodiction, which may not be possible within the deadline if the missing packets do not arrive in time, and this becomes even more challenging when fewer sensors exist. The difference is especially significant when there is only one sensor.

To sum up the above results, our PRODIC-RTS scheme, combining features such as information gain projection, selective waiting, and proactive retrodiction, often yields accuracy performance comparable to that under the full-observation case while incurring very little reporting latency, demonstrating its robustness against degradation in transmission links such as severe loss and arrival delay.

VIII. CONCLUSION

State estimation and fusion is required in many real-world applications. In this work, we have considered state estimation and fusion over a long-haul sensor network. To meet the stringent requirements on fused state accuracy and timeliness, while accounting for long latency and high loss inherent over long-haul links that exert a negative impact on fusion performance, we have proposed an information metric (PRODIC) and a modied application of the RTS retrodiction algorithm, so that the fusion center can make its online decisions on when to fuse the information contributed by the remote sensors. Simulation results of a target tracking application have validated the advantages of our design under variable link loss and delay profiles.

ACKNOWLEDGMENT

This work is funded by the Mathematics of Complex, Distributed, Interconnected Systems Program, Office of Advanced Computing Research, U.S. Department of Energy, and SensorNet Project of Office of Naval Research, and is performed at Stony Brook University under Contract No. DE-AC05-00OR22725.

REFERENCES

[1] Y. Bar-Shalom, "Update with out-of-sequence measurements in tracking: exact solution," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 769–778, 2002.

[2] Y. Bar-Shalom, H. Chen, and M. Mallick, "One-step solution for the general out-of-sequence measurement problem in tracking," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 40, no. 1, pp. 27–37, 2004.

[3] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms.* YBS Publishing, Storrs, CT, 2011.

[4] W. Boord and J. B. Hoffman, *Air and Missile Defense Systems Engineering*. CRC Press, 2013.

[5] H. Chen and X. R. Li, "On track fusion with communication constraints," *Proc. 10th International Conference on Information Fusion*, Quebec, Canada, Jul. 2007.

[6] A. Chiuso and L. Schenato, "Information fusion strategies from distributed filters in packet-drop networks," *Proc. Control Decision Conference*, Cancun, Mexico, Dec. 2008.

[7] Z. Duan and X. R. Li, "Lossless linear transformation of sensor data for distributed estimation fusion," *IEEE Trans. on Signal Processing*, vol. 59, no. 1, pp. 362–372, 2011.

[8] A. Khan, D. Schaefer, L. Tao, et al., "Low Power Greenhouse Gas Sensors for Unmanned Aerial Vehicles," *Journal of Remote Sensing*, vol. 4, no. 5, pp. 1355-1368, 2012.

[9] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. part I: dynamic models," *IEEE Trans. on Aerospace and Electronics Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.

[10] Q. Liu, X. Wang, and N. S. V. Rao, "Fusion of state estimates over long-haul sensor networks under random delay and loss," *Proc. 31st IEEE International Conference on Computer Communications (INFOCOM 2012), Mini-conference*, pp. 2968-2972, Orlando, FL, Mar. 2012.

[11] Q. Liu, X. Wang, N. S. V. Rao, K. Brigham, and V. Kumar, "Performance of state estimate fusion in long-haul sensor networks with message retransmission," *Proc. 15th International Conference on Information Fusion*, Singapore, Singapore, Jul. 2012.

[12] Q. Liu, X. Wang, and N. S. V. Rao, "Staggered scheduling of estimation and fusion in long-haul sensor networks," *Proc. 16th International Conference on Information Fusion*, Istanbul, Turkey, Jul. 2013.

[13] M. Mallick and K. Zhang, "Optimal multiple-lag out-of-sequence measurement algorithm based on generalized smoothing framework," *Proc. SPIE, Signal and Data Processing of Small Targets*, San Diego, CA, Apr. 2005.

[14] M. Moayedi, Y. K. Foo, and Y. C. Soh, "Adaptive Kalman filtering in networked systems with random sensor delays, multiple packet dropouts and missing measurements," *IEEE Trans. on Signal Processing*, vol. 58, no. 3, pp. 1577–1588, 2010.

[15] R. N. Murty and M. Welsh, "Towards a dependable architecture for internet-scale sensing," *Proc. 2nd Workshop on Hot Topics in System Dependability - Volume 2*, Seattle, WA, Nov. 2006.

[16] D. Roddy, *Satellite Communications.* McGraw-Hill, New York, NY, 2006.

[17] D. Simon, *Optimal State Estimation: Kalman, H-Infinity, and Nonlinear Approaches.* Wiley & Sons, Hoboken, NJ, 2006.

[18] L. Shi, K. H. Johansson and R. M. Murray, "Kalman filtering with uncertain process and measurement noise covariances with application to state estimation in sensor networks," *Proc. IEEE Intl' Conference on Control Applications*, Singapore, Singapore, Oct. 2007.

[19] D. K. Tasoulis, N. M. Adams, and D. J. Hand, "Selective fusion of out-of-sequence measurements," *Journal of Information Fusion*, vol. 11, pp. 183-191, 2010.

[20] U. of Southampton, "Global network of new-generation telescopes will track astrophysical events as they happen, ScienceDaily," Jan. 2011.

[21] A. Willsky, M. Bello, et al., "Combining and updating of local estimates along sets of one-dimensional tracks," *IEEE Trans. on Automatic Control*, vol. AC-27, pp. 799–813, 1982.

[22] S. Zhang and Y. Bar-Shalom, "Optimal update with multiple out-of-sequence measurements," *Proc. SPIE, Signal and Data Processing of Small Targets*, Orlando, FL, Apr. 2011.