# SELECTIVE FUSION OF DELAYED MEASUREMENTS IN FILTERING

[1]Dimitris K. Tasoulis, [2]Niall M. Adams, and [1,2]David J. Hand

[1]Institute for Mathematical Sciences, [2] Department of Mathematics
Imperial College London, South Kensington Campus
London SW7 2AZ, United Kingdom

## ABSTRACT

Delayed measurements can create difficulties for discrete time filtering. A number of procedures have been proposed to handle such delayed observations. A common characteristic of these methods is the policy of always incorporating, or fusing, delayed measurements at the time they are finally available. In this paper, we consider such methods, and develop thresholding techniques that result in delayed measurements being selectively fused. This selection is driven by an assessment of the utility of incorporating delayed measurements. With simulated and real data, we show that selective fusion can reduce computational costs while maintaining near optimal performance.

## 1. INTRODUCTION

Real-world applications of filtering are often subject to a variety of sources of uncertainty and ambiguity. One such problem is that data corruption mechanisms can often manifest as non-negligible delays in the transmission or receipt of the measurements. Here, we picture an agent receiving continuous valued, discrete time measurements from decentralized sensors. For a variety of reasons, some measurements arrive significantly later than their time stamp, that is, they are delayed. To this end the classical assumption that measurements are available immediately is easily violated [1, 2]. A number of methods have been proposed to address the problem of delayed measurements. All such methods can be characterised as attempting to incorporate, or *fuse*, delayed measurements when they are finally received. However, no consideration has been given to the important question of determining whether dedicating computational resources to fuse delayed measurements is worthwhile. The objective of this paper is to explore precisely this issue.

This research was undertaken as part of the ALADDIN (Autonomous Learning Agents for Decentralised Data and Information Systems) project and is jointly funded by a BAE Systems and EPSRC (Engineering and Physical Research Council) strategic partnership, under EPSRC grant EP/C548051/1. We are grateful to Dr Alex Rogers, University of Southampton, for introducing us to the local weather analysis problem, and providing software for data acquisition.

In this paper we restrict attention to discrete time system with state equation:

$$x_t = \Phi_t x_{t-1} + \eta_t, \tag{1}$$

and measurement equation:

$$y_t = C_t x_t + \varepsilon_t. \tag{2}$$

The random vectors $\varepsilon_t, \eta_t$ are assumed to be Gaussian and independent with $E[\varepsilon_t] = 0$, $E[\eta_t] = 0$, and $E[\varepsilon_t \varepsilon_t^\top] = R_t$, $E[\eta_t \eta_t^\top] = Q_t$. $R_t$ and $Q_t$ are the measurement and system covariance matrices, respectively. For convenience, and to isolate the character of the delay problem, both of these matrices are treated as known throughout most of this paper.

A useful concept for such a system is the signal-to-noise ratio (SNR), defined as the ratio of a given signal to the background noise of the transmission medium. In the system above the we will define the SNR as $\frac{\|Q_t\|}{\|R_t\|}$, where $\|\cdot\|$, is the Frobenius matrix norm, $\|T\| = \text{trace}\,(A^\top A)$, for matrix $T$.

The Kalman filter estimates the process $x$, by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of the (noisy) measurements $y$. We can therefore describe the Kalman filter operation with two steps: the *prediction* step and the *correction* step [3]. Time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the a priori estimates for the next time step:

**Prediction Step**

$$\hat{x}_{t|t-1} = \Phi_t \hat{x}_{t-1|t-1} \tag{3}$$

$$P_{t|t-1} = \Phi_t P_{t-1|t-1} \Phi_t^\top + Q_t \tag{4}$$

where $P_{t|t-1}$ denotes the a priori estimate for the error covariance. The measurement update equations are responsible for fusing the new measurement into the a priori esti-

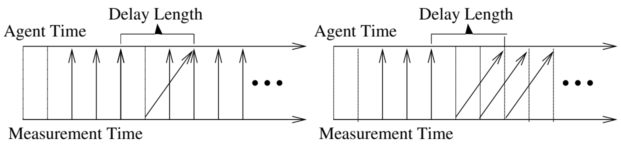


**Fig. 1**. Delay mechanics. Left: random delay. Right: Constant delay

mates to obtain improved a posteriori estimates:

$$\textbf{Correction Step}$$
$$K_t = P_{t|t-1}C_t^\top(C_tP_{t|t-1}C_t^\top + R_t)^{-1} \quad (5)$$
$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - C_t\hat{x}_{t|t-1}) \quad (6)$$
$$P_{t|t} = (I - K_tC_t)P_{t|t-1} \quad (7)$$

The difference $(y_t - C_t\hat{x}_{t|t-1})$ in Eq.6, is called the measurement innovation, or the residual, and reflects the discrepancy between the predicted $(C_t\hat{x}_{t|t-1})$ and the actual measurement $(y_t)$. The $K$ matrix is called the *Kalman gain* or *blending factor*, and it is chosen as such to minimise the a posteriori error covariance.

Our primary focus in this paper is those situations in which entire measurement vectors $y_t$ are delayed. To incorporate delay, consider the measurement vector $y_t$, delayed (or lagged) by time $\delta_t$, that becomes available to the agent at time $t + \delta_t$. We refer to the manner in which measurements are delayed as the *delay mechanics*. The delay mechanics provide the following fundamental distinction

- Constant (deterministic) delays.
- Random delays.

The problem of *constant delay* involves every measurement vector being deterministically delayed by the same constant lag. Such a delay mechanism never causes measurement vectors to be observed out-of-sequence, they are simply and consistently late. Such behaviour could be induced, for example, by a constant bandwidth restriction on a sensor network. In contrast, *random delays* provides for a large number of possibilities, including that measurements are delayed with a constant probability but fixed lag, or constant probability and random lag. Such problems could arise as a result of intermittent bandwidth restrictions on a sensor network. All modes of random delay have the potential to cause *out-of-sequence* measurements, such that $t + \delta_t$ is not constrained to be less than $(t + s) + \delta_{(t+s)}$. Both constant and random delay mechanics are illustrated in figure 1.

One simple characterisation of homogeneous delayed measurements is as follows. First, consider constant delay. The measurement vector at time $t$, $y_t$, is not available to the agent until $t + \delta$, for $\delta \in \{0, 1, 2, \ldots\}$. Note that framework includes both the non-delayed situation and the problem of missing values as special cases. Next, consider random delays. We still have $y_{t+\delta_t}$, but $\delta_t$ is a realisation of some lag random variable $D$. Different modes of random delay arise from different probability distributions for $D$. For example, consider the Bernoulli random variable $D$, with range $\{0, \delta\}$, for constant $\delta$ and all $t$, and

$$P(D = 0) = \theta \qquad P(D = \delta) = 1 - \theta$$

This is a random probability of delay, with a constant lag. Alternatively, consider $D \in \{0, 1, 2, \ldots\}$, with some probability mass function $p$ providing $P(D = \delta) = p(\delta; \theta)$. This is an example of random delay, with random lag.

Here, we restrict attention to homogeneous delay mechanics. Note that, in principle non-homogeneous cases, in which the parameters $\theta$ governing the delay also have a time-index, can be handled by the methods explored here. Random delays also have the potential to cause no measurement to be observed at some particular time point, and in consequence, more than one measurement to be observed at a later time point. For convenience, we ignore the potential operational problems this could imply.

## 2. METHODS

It is striking that all methods proposed to handle delayed measurements, including [1, 2, 4, 5, 6], have in common that delayed measurements are *always* ultimately incorporated into the filtering process. We have extensive simulation evidence that suggests these methods often provide similar empirical performance for a variety of delay mechanics. Unsurprisingly, the performance of these methods is heavily dependent on the SNR. More importantly, our simulation results suggest that when the SNR is high there is little value in incorporating delayed measurements. We should anticipate this property, since it is simply a consequence of the system changing sufficiently rapidly that delayed measurements are no longer relevant to the current situation.

Perhaps the crudest way of handling delayed measurements is simply to use the $k$-step ahead forecast equations to fill in delayed measurements, and never subsequently incorporate them. For example, if a single measurement is delayed, the 1-step ahead forecast is used, meaning the whole standard correction step (Eq.6) can be skipped, and the operation of the filter continues normally to the next prediction step. For the general case of $k$ consecutive delayed measurements, a $k$-step ahead forecast is performed, until new measurements are received. We refer to this simple, missing value approach as MKF and use it is as a baseline for performance assessment in experiments, since we certainly need to out-perform at least this simple approach.

We consider three methods proposed for handling delayed measurements. The first, the so-called measurement extrapolation Kalman filter (MEKF) is described in [7]. The

MEKF filter essentially computes and maintains a correction term that is added to the state estimation when the delayed measurement arrives.

The second algorithm, due to [8], operates in a similar fashion to MEKF. This approach, that we abbreviate ZKF, is designed to have minimum storage requirements, in response to the large storage requirements many delay-handling methods.

The final method, is the augmented state Kalman filter (ASKF). This algorithm uses state augmentation, considered the classical approach for delay problems [9]. State augmentation proceeds by increasing the state space representation to accommodate delayed measurements. Specifically, the state of the system is represented as the augmented state vector $x_t^\alpha$

$$x_t^\alpha = [x_t^\top, x_{t-1}^\top, \ldots, x_{t-\Delta}^\top],$$

where $\Delta$ is an upper bound for the delay. If we suppose that $\delta_t$ is the delay of the $t$th measurement, the measurement vector at each time $t$ becomes $Y_t = (y_{s \in S(t)})$, where $S(t) = \{s : \delta_s + s = t\}$. This augmentation procedure leads to augmented forms for the matrices $C, \Phi, R, Q$ as described by [9], and leads to an algorithm similar to the standard Kalman filter. Note also that state augmentation provides a natural means for considering components of the measurement vector as differentially delayed.

Of the methods we consider, the ASKF using state augmentation is the most general approach, since it can handle all types of delay mechanics. Additionally, the whole algorithmic description is quite clear, a fact that allows easy implementation. However the computational complexity and storage requirement of state augmentation increase linearly with with the lag. The ZKF algorithm, by construction, has a steady storage requirement, irrespective of the lag. However both the storage and the computational requirement grow quadratically with the number of simultaneous delayed measurements. MEKF, using measurement extrapolation, can also be regarded as efficient with respect to the extra storage it requires, as it is steady and unaffected by the delay length, due to its recursive nature. However, the method is not generalised by [7] to the case of simultaneously delayed measurements.

The procedures considered here, and others, are similar in that they all attempt to incorporate or fuse delayed measurements, when they finally become available. As such the methods retain certain optimality characteristics. However, as we have noted, there are resource consequences for fusing delayed measurements. Our concern in this paper is exploring the impact of selectively fusing delayed measurements on both prediction performance and resource utilisation.

In the context of known measurement and system covariance matrices, we propose thresholding procedures, such that delayed measurements that do not significantly alter the estimates are disregarded. This decision can be made *before* a delayed measurement is received by examining the variance of the innovations for the delayed measurements, since this is connected to the impact of delayed measurements on filter performance.

We modify each of the three methods described above by introducing a threshold, called the *fusion threshold* (FT). The norm of the variance of delayed measurement innovations is compared with FT. Fusing only occurs when this norm exceeds FT, meaning that it will significant impact on the filtering procedure. We treat FT as a free parameter and explore its impact for a range of values in Section 3.

For the MEKF algorithm, let $T_{\mathrm{MEKF}}$ denote the covariance matrix of the delayed measurement to be incorporated at time step $t + \delta_t$, for a previously delayed measurement $y_t$, and following the notation of [7]:

$$\begin{aligned} T_{\mathrm{MEKF}} &= Var(K(y_s^{\mathrm{int}} - C_k \hat{x}_k)) \\ &= K(C_k P_k C_k^\top + R_k) K^\top. \end{aligned} \tag{8}$$

This matrix is updated from the moment a measurement is delayed up until either the measurement arrives or the threshold condition is satisfied The matrix $K$ is computed as follows

$$K_{\delta_t} = M_* P_{s|s-1} C_s^\top [C_s P_{s|s-1} C_s^\top + R_k]^{-1} \tag{9}$$

where, following [7]:

$$M_* = \prod_{i=0}^{d-1} (I - K_{k-i} C_{k-i}) \Phi_{k-i-1}. \tag{10}$$

The derivation of $y_t^{\mathrm{int}}$ is also described in [7]. The thresholding can be applied each time the matrix $T_{\mathrm{MEKF}}$ is updated, using the respective components of the product in Eq. (10), to calculate $K$. At each time step that the delayed measurement has not arrived, if $\|T_{\mathrm{MEKF}}\| \leqslant FT$ then the fusing procedure is dropped. Thus, all subsequent computations to update $T_{\mathrm{MEKF}}$ are saved, and the operation of the filter continues normally.

Following the notation of [8], the corresponding matrix for the ZKF algorithm is:

$$\begin{aligned} T_{\mathrm{ZKF}} &= Var(K_s(z_s - C_s \hat{x}_{s|k}) \\ &= K_s(C_s P_{s|k} C_s^\top + R_s) K_s^\top \end{aligned} \tag{11}$$

where the $K_s, C_s$ and $P_{s|k}$ matrices are computed through the recursion proposed in [8]. For ZKF we determine whether to fuse delayed measurements according to the rule $\|T_{\mathrm{ZKF}}\| \leqslant FT$.

For the ASKF algorithm, however, we can implement a thresholding technique by restricting the augmentation size to a fixed value. In this way, only those delayed measurements with lag smaller than the current augmentation size are incorporated. Delayed measurements with greater lag will be automatically disregarded as missing values and handled as in the MKF.

In the following section we describe experiments designed to determine how the threshold value for MEKF and ZKF, and the augmentation size for ASKF, affects the performance of the algorithms.

## 3. EXPERIMENTAL ANALYSIS

Since our primary interest is complete measurement vectors being delayed, it is sufficient to restrict attention to the case of a univariate system equation, evolving as a random walk, and a univariate measurement equation, both as per equations (1) and (2) with known error covariances. Our experiments have constant delay probability, $\theta = 1/2$, and the lag for delayed measurements is uniformly distributed on the integers 1 to 10.

There are two aspects of performance to consider here. First we need to measure the performance of the filtering method. In our simulation results, we refer to the *error* of a delay algorithm as the mean absolute difference between the state estimates of the delay algorithm and the state estimates of the baseline Kalman filter, estimated on the true non-delayed data. Our performance metric then computes the ratio of this error measure to the corresponding error measure of the MKF. The second aspect we have to account for is the impact of incorporating delayed measurements. We account for this by tracking the proportion of delayed measurements that are actually incorporated.

The plots in Figure 2 display the ratio of the performance measure on the vertical axis and the ratio of delayed measurements on the horizontal axis. The left-hand side plot refers to SNR=0.05, while the right-hand side refers to SNR=0.25. In general, we see that performance degrades when delayed measurements are not incorporated. However, the nature of this degradation appears to be related to the SNR. For the smaller SNR, all algorithms exhibit similar behaviour, and we see that 80% performance can be obtained incorporating as few as 60% of delayed measurements. For the greater SNR the degradation of performance is less marked although the ZKF generally degrades faster than the other methods. However, 80% performance can still be achieved incorporating as few as 50% of delayed measurements.

### 3.1. Selective Fusion in Multivariate Systems

The merits of not always incorporating delayed measurements can be further extended to a multivariate system, by selecting to not fuse only those parts of the delayed measurement vector that have minimal impact.

Consider a multivariate system, with diagonal matrices $Q$ and $R$ of size $p \times p$ for measurement and system covariance respectively. In this case, we can define the per-variate signal to noise ratio $SNR_i$, by the fraction $SNR_i = \frac{q_i}{r_i}$, where $q_i, r_i$, is the $i$th diagonal element of $Q$, and $R$ respectively. We restrict attention to diagonal cases since they allow the immediate derivation of the $SNR_i$ quantity that describes the SNR of each variate individually. Otherwise more complicate procedure should be employed.

As noted above, state augmentation is a very general procedure, and enables us to modify the ASKF to handle components of a measurement vector individually. For the ASKF algorithm we can augment the system partially for those components of the delayed measurement for which the $SNR_i$ has a sufficiently high value to further reduce the computational burden. We call this technique *reduced*-ASKF (R-ASKF).

To demonstrate the computation saving the R-ASKF algorithm can achieve, we conducted a simulation of a 20 dimensional system, where delays occur randomly with a probability of $\theta = 0.5$, with lag between 1 and 10 uniformly likely. The SNR was constructed as $SNR_i = 10.1$ for $i \leqslant 10$, and $SNR_i = 0.1$ for $i \geqslant 11$. This way half of the variates should have more impact on the algorithm's performance, since their evolution is much more difficult to track. This can be seen in Fig. 3, which illustrates the performance of ASKF algorithms for different augmentation sizes, plotted against the required computational effort. The latter is determined by the augmentation size itself, since it dictates the size of each matrix involved in the filter operation. The matrix size affects not only the computation but also the memory requirements of the whole methodology. As anticipated, the best performance is achieved by a procedure that augments only the first 10 variables since those variables are the most important to incorporate when they are delayed.

### 3.2. Local Weather Data Analysis

Local weather data analysis is an interesting application that provides a good test case for selective delay fusion since delays are inherent in this context. There are a vast number of automated weather stations, both commercial and private, that obtain and report a stream of measurements on a set meteorological characteristics. These stations typically report measurements at fixed intervals, but because of imprecise timing, the measurements are not synchronised. We have used 5 such automated weather stations distributed around Southampton, in the UK. These stations were queried for measurements of *wind speed* and *air temperature* every minute between $1/02/2006$ and $3/02/2006$.

For simplicity we assume that measurements originating from different stations are uncorrelated, whereas measurements from sensors in the same station are considered fully correlated. Furthermore we do not make any assumptions

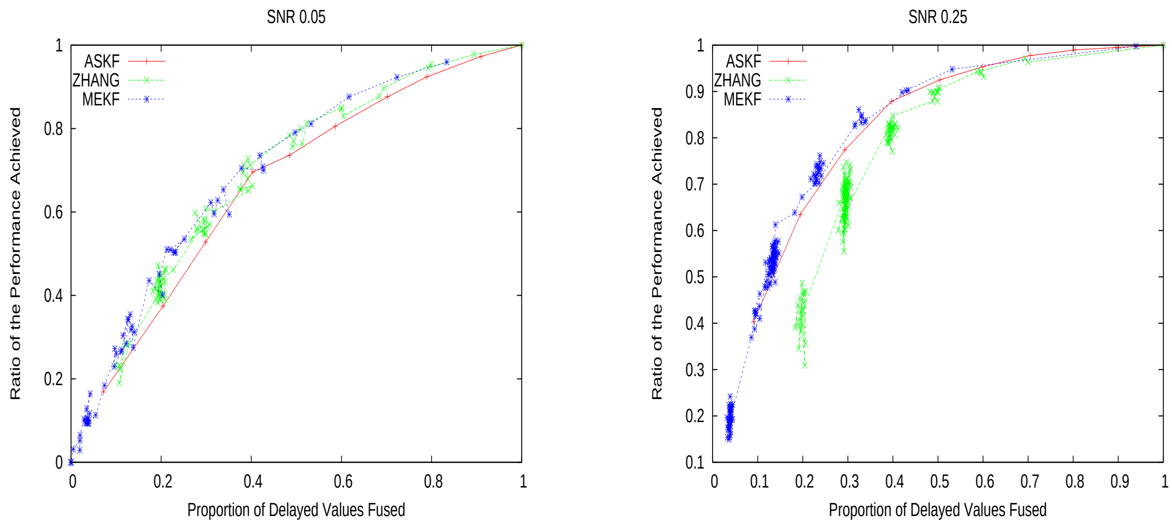


**Fig. 2**. The effect of ratio of the delays fused on the algorithms performance for a random delay length between 1 and 10.

for the latter correlation structure, but instead prefer to learn it from historic data. As such we employ the Expectation Maximization (EM) algorithm [10] on a Kalman filter operating on the initial 200 measurements obtained from station $s_i$ for $i = 1, \ldots, 5$, to estimate the matrices $Q_i$ and $R_i$ for each station. These estimates of $Q_i, R_i$ are stacked together into the block diagonal matrices $Q_f$ and $R_f$ respectively, and then embedded in the filtering procedure of the system consisting of all five stations together. In real life applications like this, fusing delays can be very important since it will often be the case that gross system changes coincide with delays. To illustrate this, in Fig. 4 we present an example of filtering results on temperature measurement for one of the stations, when filtering is done across all stations as described above. In this case delays start to occur exactly when there is a big jump in the obtained measurements (at the 9th time step). We see that the ASKF approach dapts to this large change more rapidly than MKF. The per station SNR can be computed by $SNR_{s_i} = \frac{\|Q_{s_i}\|}{\|R_{s_i}\|}$ where $\| \cdot \|$ is a matrix norm.

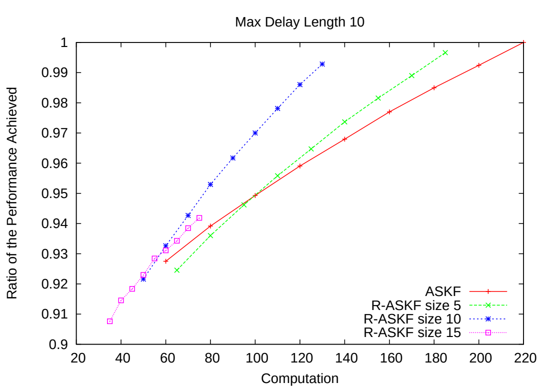


**Fig. 3**. ASKF performance comparison between different augmentation strategies.

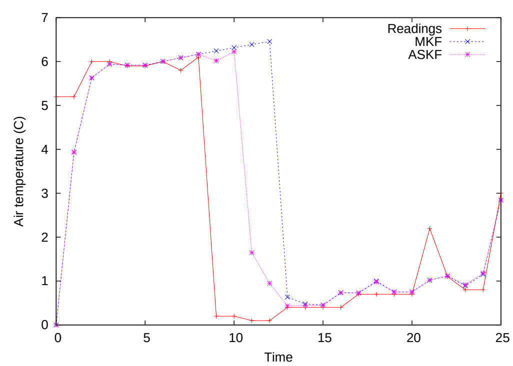


**Fig. 4**. An example of filtering on Temperature measurements.

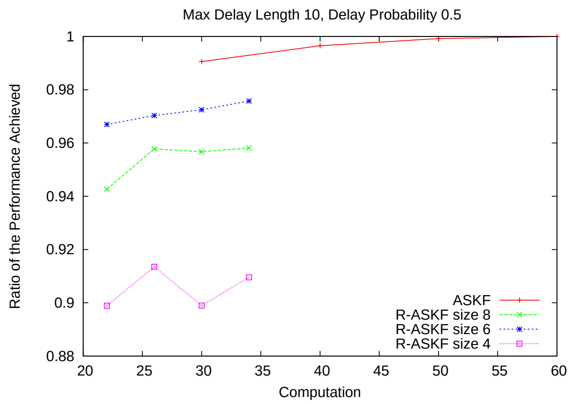


**Fig. 5**. The performance of the R-ASF strategy on the Weather Data Stream.

From the estimated values for each $Q_{s_i}$ and $R_{s_i}$, we have the following estimated $SNR_i$:

$$SNR = \{19.3339, 8.46144, 0.37551, 0.114866, 0.025552\}$$

Experiments with different size of partial and full augmentation produced the results in Fig. 5. The performance curve for R-ASKF with augmentation size 8 that partially augments (and subsequently fuses) only the first 8 components of the system ($SNR_{s_i} \geqslant 0.1$).

As shown in Fig. 5 while full augmentation can maintain quite high levels of performance, it is not possible to reduce the computation effort by more than half. However, if we employ the partial augmentation strategy of R-ASKF, by choosing not to always fuse measurements originating from stations 4 and 5, (as indicated by the estimated SNR above) the computational requirements can be reduced down to one third. In all cases performance remains at very high levels.

## 4. CONCLUSIONS

In this paper we have examined the problem of fusing delayed measurements in discrete time filtering. A notable common feature of all the methods proposed to handle the delay problem is that delayed measurements are always incorporated when they finally arrive.

Our experimental analysis draws attention to an important issue that has not been fully explored to date. Specifically, we have investigated the relationship of the signal-to-noise ratio to the relevance of delayed measurements. This relationship can have significant impact on the performance of the considered methods, and since they have different characteristics, probably extends to all similar approaches.

Our results show that fusing delayed measurements can sometimes be worthless. To address this problem, we propose a thresholding technique to selectively incorporate only those delayed measurements that will improve the performance of the filter. In both simulated and real world scenarios, we show that most of the filtering performance of delay-handling algorithms can be maintained by fusing few delayed measurements, especially as the SNR ratio increases. In future work we intend to develop methods for the automatically determining the fusion threshold.

## 5. REFERENCES


[1] A. Ray, "Output feedback control under randomly varying distributed delays," *J. Guidance, Control Dynamics*, vol. 17, no. 4, pp. 701–711, 1994.

[2] K. Yamanaka, K. Uchida, and E. Shimemura, "Optimal control of systems with random delay," *Internat. J. Control*, vol. 29, no. 3, pp. 489–495, 1979.

[3] R.E. Kalman, "A new approach to linear filtering and prediction problems," *Transaction of the ASME–Journal of Basic Engineering*, pp. 35–45, 1960.

[4] Y. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*, Academic Press, New York, 1988.

[5] S. Challa, R.J. Evans, and X. Wang, "A Bayesian solution and its approximations to out-of-sequence measurement problems," *Information Fusion*, vol. 4, pp. 185–199, 2003.

[6] A.S. Matveev and A.V. Savkin, "The problem of state estimation via asynchronous communication channels with irregular transmission times," *IEEE Transactions on Automatic Control*, vol. 48, no. 4, pp. 670–676, 2003.

[7] T. Larsen, N. Poulsen, N. Andersen, and O. Ravn, "Incorporation of time delayed measurements in a discrete-time Kalman filter," in *CDC '98, Tampa, Florida, .*, 1998.

[8] K. Zhang, X.R. Li, and Y. Zhu, "Optimal update with out-of-sequence measurements," *IEEE Transactions on Signal Processing*, vol. 53, no. 6, pp. 1992– 2004, 2005.

[9] X. Lu, H.S. Zhang, W. Wang, and K.L. Teo, "Kalman filtering for multiple time-delay systems," *Automatica*, vol. 41, no. 8, pp. 1455–1461, 2005.

[10] R. H. Shumway and D. S. Stoffer, "An approach to time series smoothing and forecasting using the EM algorithm," *Journal of Time Series Analysis*, vol. 3, no. 4, pp. 253–263, 1982.