# Optimal Update with Out-of-Sequence Measurements *

**Keshu Zhang** **X. Rong Li**, Senior Member, IEEE

Department of Electrical Engineering, University of New Orleans, New Orleans, LA 70148

Phone: 504-280-7416, Email: xli@uno.edu

**Yunmin Zhu**

Department of Mathematics, Sichuan University, Chengdu, Sichuan 610064, P.R. China

**Abstract:** This paper is concerned with optimal filtering in a distributed multiple sensor system with the so-called out-of-sequence measurements (OOSM). Based on BLUE (best linear unbiased estimation) fusion, we present two algorithms for updating with OOSM that are optimal for the information available at the time of update. Different minimum storages of information concerning the occurrence time of OOSMs are given for both algorithms. It is shown by analysis and simulation results that the two proposed algorithms are flexible and simple.

**Keyword:** Target tracking, out of sequence measurement, LMMSE, Kalman filter.

## 1 Introduction

In a distributed multiple-sensor tracking system, observations produced by the sensors typically arrive at a fusion center with a random time delay due to communication delay. The state equations are usually obtained in continuous time and then discretized. The sensor may provide a "time stamp" with each measurement. In centralized multi-sensor tracking systems, all these measurements are sent to the fusion center for processing. There are usually different time delays in transmitting data into the fusion center. This can lead to situations where measurements from the same target arrive out of sequence. In this case, a measurement produced at time $t_k$ is received at the fusion center and is used to produce an updated track state estimate and covariance matrix for that time $t_k$. Then, a delayed observation $z_d$ produced at a prior time $t_d$ ($t_{k-l} \leq t_d < t_{k-l+1}$, $l = 1, 2, \cdots$) is received at the fusion center. This could occur if the observation produced at time $t_d$ was subject to a longer transmission delay than the delay associated with the observation produced at the later time $t_k$.

The problem is how to use the "older" measurement from time $t_d$ to update the current state at $t_k$. There are some methods for updating the state estimate globally optimally with an out-of-sequence measurement (OOSM) within one step time delay (i.e., $l = 1$, referred to as one-step update) for a system with nonsingular

*Research Supported in part by NSF grant ECS-9734285, and NASA/LEQSF grant (2001-4)-01.

state transition matrix [1], and multi-step OOSM updating using augmented state smoothing [5][6][4][5]. Also, one-step suboptimal updating algorithms using stored information have been proposed for systems with invertible state transition matrix [14][7][3][1]; multi-step update was discussed in [11] without any discussion of the optimality. These algorithms are shown in this paper to be special cases of our proposed update algorithms. The globally optimal update algorithm and the optimal update algorithm with limited information are optimal in the linear minimum mean-square error (LMMSE) sense. When the required condition holds, our optimal update algorithm with limited information given reduces to the suboptimal algorithms of [1][11], which provides a simple proof of the optimality of these generally suboptimal algorithms.

In this paper, we first present a discussion concerning what the minimum storage at the current time is to guarantee a globally optimal update. We derive our first algorithm to give a globally optimal LMMSE update by storing all necessary information. It is general and systematic. We consider three cases of prior information about the OOSM. In each case, we try to get the minimum storage. A comparison with existing globally update algorithms in computation and storage is also discussed. Our second algorithm gives the LMMSE update by only using the information available at the current time. Although not guaranteed to be globally optimal, it is optimal for the information given. As for the first algorithm, we also consider three cases of information storage for the second algorithm. Further, we extend the above single-OOSM update algorithms to the case of arbitrarily delayed multiple OOSMs.

The results presented in this paper also demonstrate how the "static" estimation fusion formulas presented in [10] can be applied to dynamic state estimation and fusion.

This paper is a revised and extended version of the conference proceedings paper [15]. The rest of the paper is organized as follows. Section 2 formulates the problem. LMMSE update with available information is discussed in a general setting in Section 3. Two algorithms for optimal update of the state estimate with OOSM are presented in Section 4, along with associated minimum information storage. Numerical examples are provided in Section 5. Section 6 concludes the paper with a summary.

## 2 Problem Formulation

The dynamics and measurement models assumed are given by

$$x_j = F_{j,j-1}x_{j-1} + w_{j,j-1} \tag{1}$$

$$z_j = H_j x_j + v_j \tag{2}$$

where $F_{j,j-1}$ is the state transition matrix from time $t_{j-1}$ to $t_j$ and $w_{j,j-1}$ is (the cumulative effect of) the process noise for the interval $[t_{j-1}, t_j]$. The process noise $w_{j,j-1}$ and the measurement noise $v_j$ are white and have zero mean and covariances

$$C_{w_{j,j-1}} = \text{cov}(w_{j,j-1}) = Q_{j,j-1}, \;\; C_{v_j} = \text{cov}(v_j) = R_j$$

Suppose time $t_d$ is in the sampling interval $t_{k-l} \leq t_d < t_{k-l+1}$, where $l = 1, 2, \cdots$. Which means that the OOSM $z_d$ is $l$ lags behind.

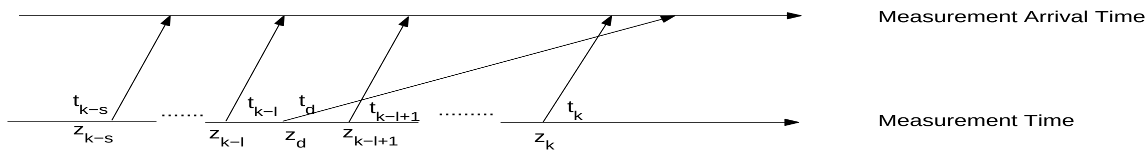


**Fig. 2.1: The OOSM $z_d$ arrives after the last processed measurement $z_k$.**

Similar to (1), we have

$$x_k = F_{k,d}x_d + w_{k,d}$$
$$z_d = H_d x_d + v_d$$

The problem is as follows: At time $t_k$ the LMMSE estimator is

$$\hat{x}_{k|k} \triangleq E^*(x_k|z^k) = \underset{\hat{x}_{k|k}=Az^k+B}{\arg\min} P_{k|k}, \quad P_{k|k} \triangleq \text{MSE}[\hat{x}_{k|k}]$$

where $\tilde{x} = x - \hat{x}$. Letting $C_{xz}$ =cov$(x, z)$ and $C_z$ =var$(z)$,

$$\hat{x} = E^*(x|z) = \bar{x} + C_{xz}C_z^{-1}(z - \bar{z}), \quad \text{MSE}(\hat{x}) = E(\tilde{x}\tilde{x}')$$

In the above, $z^k \triangleq \{z_i\}_{i=1}^k$ is the measurement sequence through $t_k$. If the inverse $C_z^{-1}$ does not exist, it can be simply replaced with the unique *Moore-Penrose pseudoinverse* (MP inverse in short) $C_z^+$. Subsequently, an earlier measurement at time $t_d$ arrives after the state estimate $\hat{x}_{k|k}$ and error covariance $P_{k|k}$ have been calculated. We want to update this estimate with the earlier measurement $z_d$, that is, to calculate the LMMSE estimator

$$\hat{x}_{k|k,d} = E^*(x_k|\Omega_k, z_d), \quad P_{k|k,d} = \text{MSE}[\hat{x}_{k|k,d}]$$

where $\Omega_k$ stands for the information available for update with the OOSM $z_d$ .

# 3 Optimal Update with Available Information

In general, we want to have the globally optimal updated estimate

$$\hat{x}_{k|k,d} = E^*(x_k|z^k, z_d)$$

And generally

$$E^*(x_k|z^k, z_d) = E^*(x_k|\hat{x}_{d|k-l}, z^{d,k})$$

Therefore, if we want to guarantee a globally optimal update, the information stored at each time $t_m$ $(k - l + 1 \leq m \leq k)$ should include at least

$$\Omega_m = \{\hat{x}_{k-l|k-l}, P_{k-l|k-l}, z^{k-l+1,m}\}$$

or its equivalent. Otherwise, no guarantee that any update is globally optimal in general. In some special cases, however, information from a smaller storage is sufficient for global optimality. Thus, all measurements, state estimates, and error covariances from the occurrence time of OOSM to its arrival time need to be saved to guarantee a globally optimal update.

In practice, an OOSM $z_d$ has a random time delay (e.g., $l$ is random). But it may not be too far before time $t_k$. It is not reasonable to store the observation at each time in order to get the optimal updated estimate $E^*(x_k|z^k, z_d)$. In fact, in each step, it is often the case that only $\hat{x}_{j|j}$ and the associated error covariance $P_{j|j}$ are stored. So at time $t_k$, the available information stored is $\Omega_k = \left\{\hat{x}_{k|k}, P_{k|k}\right\}$. Thus the optimal update is better done based on this $\Omega_k$ and OOSM. Now the update in general can be done by using formulas for BLUE fusion without prior [10], because the prior information may not be available for update with OOSM. This update is not globally optimal in general, but it is optimal for the information given.

Both algorithms presented in the next section are optimal in the LMMSE sense. They differ in that different $\Omega_k$ are used and thus they are optimal for different available information $\Omega_k$.

# 4 Optimal Update Algorithms

## 4.1 Algorithm I — Globally Optimal Update

Based on the linear dynamic model, according to recursive LMMSE, the globally optimal update can be expressed as

$$\hat{x}_{k|k,d} = E^*(x_k|z^k, z_d) = \hat{x}_{k|k} + K_d(z_d - H_d\hat{x}_{d|k}) = \hat{x}_{k|k} + K_d\tilde{z}_{d|k} \tag{3}$$

$$\text{MSE}(\hat{x}_{k|k,d}) = P_{k|k} - K_dS_dK_d' \tag{4}$$

where

$$K_d = U_{k,d}H_d'S_d^{-1}, \;\; S_d = H_dP_{d|k}H_d' + R_d, \;\; U_{k,d} = C_{x_k,\tilde{x}_{d|k}} = \text{cov}(x_k, \tilde{x}_{d|k})$$

In the above, if the inverse $S_d^{-1}$ does not exist, we can simply replace it with $S_d^+$, the MP inverse of $S_d$ =cov$(\tilde{z}_{d|k})$. $\hat{x}_{k|k}$ and $P_{k|k}$ are available in the Kalman filter. In the following we focus on other necessary information $\{\hat{x}_{d|k}, P_{d|k}, U_{k,d}\}$, which in fact exists in a recursive form (non-standard smoothing):

Let

$$\hat{x}_{d|n} = E^*(x_d|z^n), \;\; P_{d|n} = \text{MSE}(\hat{x}_{d|n}), \;\; U_{n,d} = C_{x_n,\tilde{x}_{d|n}}$$

Then, starting from $n = k - l + 1$, we have the recursion (see Appendix A)

$$\begin{aligned} \hat{x}_{d|n+1} &= \hat{x}_{d|n} + U_{n,d}'F_{n+1,n}'H_{n+1}'S_{n+1}^{-1}\tilde{z}_{n+1|n} \\ P_{d|n+1} &= P_{d|n} - U_{n,d}'F_{n+1,n}'H_{n+1}'S_{n+1}^{-1}H_{n+1}F_{n+1,n}U_{n,d} \\ U_{n+1,d} &= (I - K_{n+1}H_{n+1})F_{n+1,n}U_{n,d} \end{aligned} \tag{5}$$

with initial value

$$
\begin{aligned}
\hat{x}_{d|k-l+1} &= \hat{x}_{d|k-l} + P_{d|k-l}F'_{k-l+1,d}H'_{k-l+1}S^{-1}_{k-l+1}\tilde{z}_{k-l+1|k-l} \\
P_{d|k-l+1} &= P_{d|k-l} - P_{d|k-l}F'_{k-l+1,d}H'_{k-l+1}S^{-1}_{k-l+1}H_{k-l+1}F_{k-l+1,d}P_{d|k-l} \\
U_{k-l+1,d} &= (I - K_{k-l+1}H_{k-l+1})F_{k-l+1,d}P_{d|k-l}
\end{aligned}
\tag{6}
$$

where

$$
\hat{x}_{d|k-l} = F_{d,k-l}\hat{x}_{k-l|k-l} \tag{7}
$$

$$
P_{d|k-l} = F_{d,k-l}P_{k-l|k-l}F'_{d,k-l} + Q_{d,k-l} \tag{8}
$$

Based on the above recursion, it is easy to get that $\{\hat{x}_{d|k}, P_{d|k}, U_{k,d}\}$ are highly related with the OOSM occurrence time $t_d$ through $\{\hat{x}_{d|k-l}, P_{d|k-l}\}$ which are highly related with the state estimate of $x_d$ at that time. The key to achieve global optimality for the update lies in when and how to initialize the recursion.

Depending on different prior information about $t_d$, we consider three cases.

**Case I: Perfect Knowledge about $t_d$ at the Next Sampling Time $t_{k-l+1}$**

In this case, we know the exact sampling time at which each observation is made and supposed to arrive. Suppose $z_d$ made at $t_d$ ($t_{k-l} \le t_d < t_{k-l+1}$) has not arrived by $t_{k-l+1}$ (so we know we have an OOSM); instead, it arrives during $[t_k, t_{k+1})$ with a time stamp $t_d$. Then at the time at which $z_d$ is supposed to arrive, we can still run the Kalman filter to get prediction $\{\hat{x}_{d|k-l}, P_{d|k-l}\}$; the only difference is that there is no state update with $z_d$. Then at the next time $t_{k-l+1}$, we can initialize by (6) and run our recursion (5) until the OOSM arrives. This filter is an extension of the traditional Kalman filter by adding $\{\hat{x}_{k|n}, P_{k|n}, U_{k,n}\}$ at each recursion $n$ ($k-l+1 \le n \le k$). After receiving the OOSM, the OOSM update algorithm is globally optimal. The complete algorithm is the Kalman filter associated with the OOSM update, which is referred to as *KF-OOSM*. The *KF-OOSM* for Case I is shown in **Fig.4.1.1**.

Since the traditional Kalman filter stores $\{\hat{x}_{n|n}, P_{n|n}\}$ at each recursion, the information stored in our *KF-OOSM* at each recursion $n$ ($k-l+1 \le n \le k$) is

$$
\Omega_n = \{\hat{x}_{n|n}, P_{n|n}, \hat{x}_{d|n}, P_{d|n}, U_{n,d}\}
$$

In this case, the storage is fixed as the delay $l$ increases.

**Case II: Knowing $t_{k-l} < t_d < t_{k-l+1}$ at Time $t_{k-l+1}$**

In this case, we know exactly the sampling interval over which each observation is made and supposed to arrive; supposed $z_d$ made at $t_d$ ($t_{k-l} \le t_d < t_{k-l+1}$) has not arrived by $t_{k-l+1}$ (so we know we have an OOSM); instead, it arrives during $[t_k, t_{k+1})$ with a time stamp $t_d$. Then, at time $t_{k-l+1}$ we can not use $\{\hat{x}_{d|k-l+1}, P_{d|k-l+1}, U_{k-l+1,d}\}$ directly to initialize our *KF-OOSM* because they are all related with the state $x_d$. Without receiving the OOSM $z_d$ at time $t_{k-l+1}$, the necessary state information $x_d$ is not available at that time, but we can initialize our *KF-OOSM* using the replacement $\{y_{k-l+1}, B_{k-l+1}, U_{k-l+1}\}$, defined by

$$
y_{k-l+1} = H'_{k-l+1}S^{-1}_{k-l+1}\tilde{z}_{k-l+1|k-l}, \;\; B_{k-l+1} = H'_{k-l+1}S^{-1}_{k-l+1}H_{k-l+1}, \;\; U_{k-l+1} = I - K_{k-l+1}H_{k-l+1} \tag{9}
$$

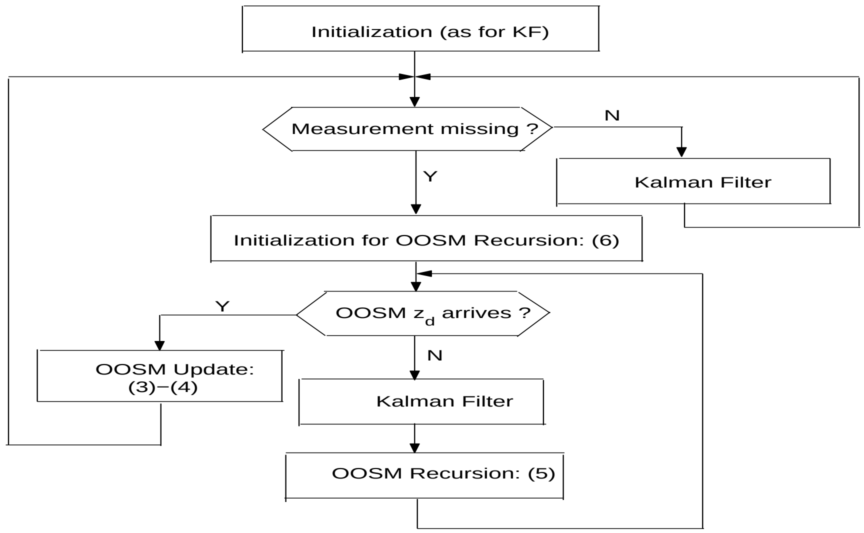


**Fig.4.1.1: Algorithm I for Case I**

None of them are related with state $x_d$ and they are generated using the information available in the traditional Kalman filter at that time. We can define the recursion for $\{y_n, B_n, U_n\}$ with $k-l+1 \le n \le k$ as

$$\begin{aligned} y_{n+1} &= y_n + U_n' F_{n+1,n}' H_{n+1}' S_{n+1}^{-1} \tilde{z}_{n+1|n} \\ B_{n+1} &= B_n + U_n' F_{n+1,n}' H_{n+1}' S_{n+1}^{-1} H_{n+1} F_{n+1,n} U_n \\ U_{n+1} &= (I - K_{n+1} H_{n+1}) F_{n+1,n} U_n \end{aligned} \quad (10)$$

Then $\{\hat{x}_{d|k}, P_{d|k}, U_{k,d}\}$ can be obtained by renewing $\{y_k, B_k, U_k\}$ once the OOSM $z_d$ arrives (see Appendix B).

$$\begin{aligned} \hat{x}_{d|k} &= P_{d|k-l} F_{k-l+1,d}' y_k + \hat{x}_{d|k-l} \\ P_{d|k} &= P_{d|k-l} - P_{d|k-l} F_{k-l+1,d}' B_k F_{k-l+1,d} P_{d|k-l} \\ U_{k,d} &= U_k F_{k-l+1,d} P_{d|k-l} \end{aligned} \quad (11)$$

The *KF-OOSM* for Case II is shown in **Fig.4.1.2**.

The information needed to be stored for our *KF-OOSM* at each recursion $n$ ($k-l+1 \le n \le k$) in this case is

$$\Omega_n = \{\hat{x}_{n|n}, P_{n|n}, y_n, B_n, U_n, \hat{x}_{k-l|k-l}, P_{k-l|k-l}\}$$

**Remark** If $l = 1$ (i.e., one-step update), there is only one recursion in our *KF-OOSM*; so the information needed to be stored is simply

$$\Omega_k = \{\hat{x}_{k|k}, P_{k|k}, y_k, B_k, \hat{x}_{k-1|k-1}, P_{k-1|k-1}\}$$

and

$$U_{k,d} = [I + (F_{k,d} P_{d|k-1} F_{k,d}' + Q_{k,d}) B_k] F_{k,d} P_{d|k-1}$$

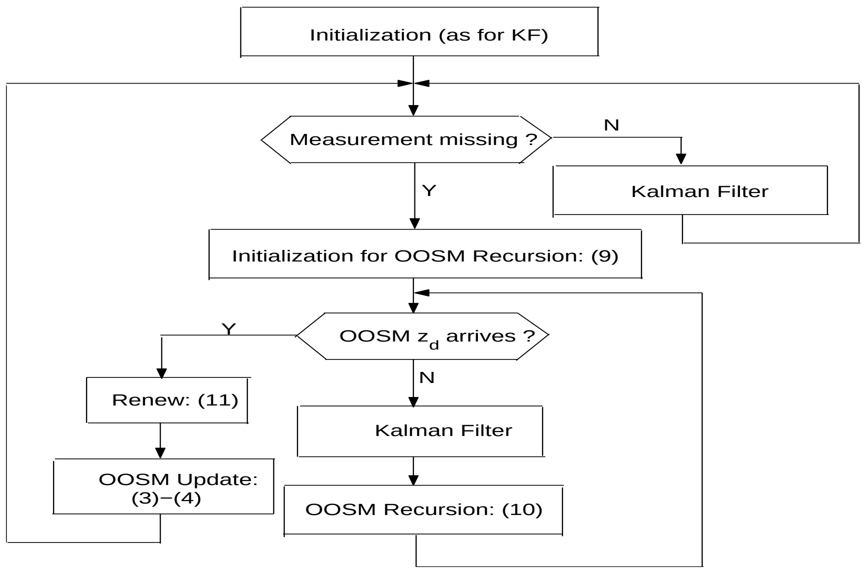


**Fig.4.1.2: Algorithm I for Case II**

In this case, the storage is also fixed as the delay $l$ increases.

**Case III: Knowing Maximum Delay $s$ of OOSM**

In this case, there is not any prior information about the OOSM $z_d$ occurrence time $t_d$ before it arrives, but we know the maximum delay $s$ for the OOSM, i.e., $t_{k-s} \leq t_{k-l} \leq t_d < t_{k-l+1} \leq t_k$ with an unknown $l$.

In this case, we do not know when to initialize our *KF-OOSM*, but we can treat each discrete time in the time window $[t_{k-s}, t_k)$ as the possible initialization point, such as

$$y_n^{(n)} = H_n' S_n^{-1} \tilde{z}_{n|n-1}, \;\; B_n^{(n)} = H_n' S_n^{-1} H_n, \;\; U_n^{(n)} = I - K_n H_n \tag{12}$$

and apply the algorithm in Case II to achieve the optimal update after the OOSM is received. The recursion for $\{y_{n+1}^{(m)}, B_{n+1}^{(m)}, U_{n+1}^{(m)}\}$ ($n > k-l+1$, $n-s < m < n$) is

$$\begin{aligned} y_{n+1}^{(m)} &= y_n^{(m)} + U_n^{(m)\prime} F_{n+1,n}' H_{n+1}' S_{n+1}^{-1} \tilde{z}_{n+1|n} \\ B_{n+1}^{(m)} &= B_n^{(m)} + U_n^{(m)\prime} F_{n+1,n}' H_{n+1}' S_{n+1}^{-1} H_{n+1} F_{n+1,n} U_n^{(m)} \\ U_{n+1}^{(m)} &= (I - K_{n+1} H_{n+1}) F_{n+1,n} U_n^{(m)} \end{aligned} \tag{13}$$

Then $\{\hat{x}_{d|k}, P_{d|k}, U_{k,d}\}$ can be obtained by renewing $\{y_k^{(k-l+1)}, B_k^{(k-l+1)}, U_k^{(k-l+1)}\}$ once the OOSM $z_d$ arrives

$$\begin{aligned} \hat{x}_{d|k} &= P_{d|k-l} F_{k-l+1,d}' y_k^{(k-l+1)} + \hat{x}_{d|k-l} \\ P_{d|k} &= P_{d|k-l} - P_{d|k-l} F_{k-l+1,d}' B_k^{(k-l+1)} F_{k-l+1,d} P_{d|k-l} \\ U_{k,d} &= U_k^{(k-l+1)} F_{k-l+1,d} P_{d|k-l} \end{aligned} \tag{14}$$

The *KF-OOSM* for Case III is shown in **Fig.4.1.3**.

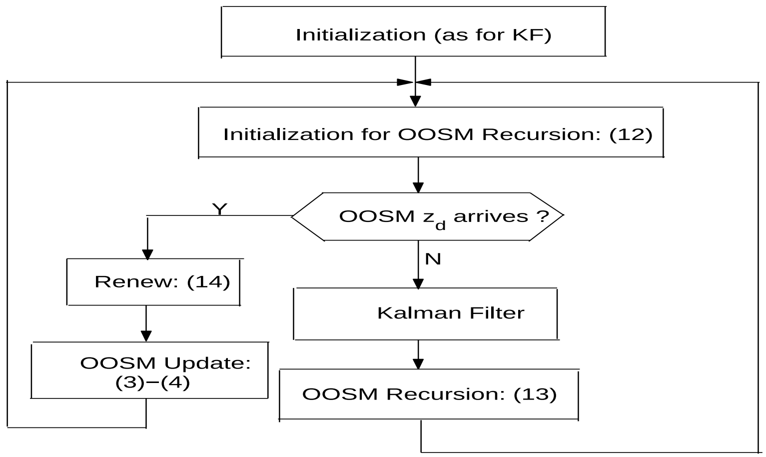


**Fig4.1.3: Algorithm I for Case III**

The information storage in our *KF-OOSM* at each recursion $n$ in this case increases linearly as time increases from $t_{k-s+1}$ to $t_k$, which is as follows:

$$\Omega_n = \{\hat{x}_{k-s|k-s}, P_{k-s|k-s}, \cdots, \hat{x}_{n|n}, P_{n|n}, y_n^{(k-s+1)}, B_n^{(k-s+1)}, U_n^{(k-s+1)}, \cdots, y_n^{(n)}, B_n^{(n)}, U_n^{(n)}\}$$

**Remark 1** If $s = 1$ (i.e., one-step update), it is just the problem considered in Case II with $l = 1$.
**Remark 2** If $t_d = t_{k-l}$, the OOSM was made exactly at a previous sampling time. With the algorithm slightly changed, the memory can be saved comparing with $t_{k-l} < t_d < t_{k-l+1}$. In this case, the flowchart of *KF-OOSM* for this case has the same structure as above, but the OOSM initialization and renewing part are respectively replaced with

$$\begin{aligned} y_n^{(n)} &= \hat{x}_{n|n} + P_{n|n} F'_{n,n-1} H'_n S_n^{-1} \tilde{z}_{n|n-1} \\ B_n^{(n)} &= P_{n|n} - P_{n|n} F'_{n,n-1} H'_n S_n^{-1} H_n F_{n,n-1} P_{n|n} \\ U_n^{(n)} &= (I - K_n H_n) F_{n,n-1} P_{n|n} \end{aligned}$$

and

$$\hat{x}_{d|k} = y_k^{(k-l+1)}, \; P_{d|k} = B_k^{(k-l+1)}, \; U_{k,d} = U_k^{(k-l+1)}$$

The information needed at each recursion $n$ $(k - s < n \leq k)$ in our *KF-OOSM* in this case is

$$\Omega_n = \{\hat{x}_{n|n}, P_{n|n}, y_n^{(k-s+1)}, B_n^{(k-s+1)}, U_n^{(k-s+1)}, \cdots, y_n^{(n)}, B_n^{(n)}, U_n^{(n)}\}$$

Depending on the uncertainty with the OOSM occurrence time, we have considered three cases of the *KF-OOSM* and the associated efficient memory structure for Algorithm I. The more uncertain the OOSM occurrence time, the more storage we need. Cases I and II have a fixed storage from time $t_{k-l}$ to $t_k$. The storage in case III increases linearly with the length of the time interval. None of these algorithms for the three cases generally have any non-singularity requirement on the transition matrix $F_{k,d}$.

## 4.2 Comparison of Globally Optimal Update Algorithms

[1, 5] present algorithms to achieve globally optimal update. [1] deals with single-step update problem. There are two major steps: retrodiction from current time to OOSM occurring time and update the estimate with the OOSM. [5] is suitable for our Case III assuming $t_d = t_{k-l}$. It is based on the method of non-standard smoothing by augmenting the state vector to include all "states" $x_{k-s}, x_{k-s+1}, \ldots, x_k$. It is conceptually elegant, but not attractive computationally. The OOSM problem is trivial without considering storage or computation. In other words, storage and computation are crucial consideration for OOSM problems. It seems impossible to have a globally optimal update with the OOSM when $t_d \neq t_{k-l}$ (i.e., $t_d$ is not exactly some sampling time instant) within this framework of state augmentation. A technique was suggested in [5] to handle the problem with $t_{k-l} < t_d < t_{k-l+1}$ by approximating $t_d$ to the nearest time $t_{k-l}$ or $t_{k-l+1}$. The approximation makes the estimation not globally optimal. The error is small when the sampling intervals are small. However, this requires more lags (i.e., large $l$) in the augmented state, and hence increases computational load to cover the same maximum time delay. This is a dilemma when one wants to have small errors and efficient computation simultaneously.

The algorithm presented here do not need to retrospect to the previous state or go back by smoothing. It is the traditional Kalman filter with a few more terms in the recursion. It requires more storage than the Kalman filter, but the extra storage used is not large. In Cases I and II, the storage is fixed at each recursion and even in Case III, the storage increases only linearly as the delay $l$ increases.

Let us simply compare the storage and computational load of our OOSM update algorithm with those of [5] in Case III assuming $t_d = t_{k-l}$. In the following, we consider the same maximum delay $s$ for the OOSM and focus on the total storage and computational burden within a certain time window. The algorithm of [5] is referred as ALG-S and our globally optimal algorithm I for Case III as ALG-I in the following:

ALG-S (Storage):

$$\begin{bmatrix} \hat{x}_{k-s|k} \\ \vdots \\ \hat{x}_{k|k} \end{bmatrix}, \quad \begin{bmatrix} P_{k-s|k} & \cdots & C_{\tilde{x}_{k-s|k},\tilde{x}_{k|k}} \\ \vdots & \ddots & \vdots \\ C_{\tilde{x}_{k|k},\tilde{x}_{k-s|k}} & \cdots & P_{k|k} \end{bmatrix}$$

ALG-I (Storage):

$$\begin{bmatrix} y_k^{(k-s+1)} \\ \vdots \\ y_k^{k-1} \\ \hat{x}_{k|k} \end{bmatrix}, \quad \begin{bmatrix} B_k^{(k-s+1)} & U_k^{(k-s+1)} \\ \vdots & \vdots \\ B_k^{(k-2)} & U_k^{(k-2)} \\ B_k^{(k-1)} & U_k^{(k-1)} \end{bmatrix}$$

Obviously, the dimension of each term of the stacked estimates and the corresponding covariances in the two algorithms is the same, which is the same as $x_k$ or $P_{k|k}$. Let the dimension of $x_k$ be $p$, and the dimension of $P_{k|k}$ be $p \times p$. The total storages for the two algorithms are:

$$\begin{array}{ll} \text{ALG-S} & sp + s^2(p \times p) \\ \text{ALG-I} & (s-1)p + 2s(p \times p) \end{array}$$

Although the storage of ALG-S can be as small as $sp + (s^2 + s)(p \times p)/2$ because of the symmetry of the covariance matrix, the storage is still quadratic in $s$. The storage of the ALG-I is linear in $s$, which means as the maximum delay $s$ increases, the storage of ALG-S will be much larger than that of ALG-I. The maximum delay $s$ could be quite large in the case of small sampling interval or large computational delay. In practice, $t_d \neq t_{k-l}$, to have good performance for ALG-S, the sampling interval must be small and thus $s$ is large. Consequently, ALG-S should have significantly larger computational complexity than ALG-I.

By analysis and comparison, we can conclude that our proposed globally optimal update algorithm has (1) an efficient memory structure; and (2) an efficient computational structure to solve the problem by storing the necessary information instead of retrodiction or augmenting the state. Also, it is globally optimal for $t_{k-l} < t_d < t_{k-l+1}$ as well as $t_d = t_{k-l}$, whereas ALG-S is globally optimal only for $t_d = t_{k-l}$. On the other hand, ALG-S is conceptually clearer and simpler than ALG-I.

### 4.3 Algorithm II — Constrained Optimal Update

Only based on information $\hat{x}_{k|k}$ and $z_d$ at the time when OOSM $z_d$ arrives, the OOSM update is the LMMSE estimation $E^*(x_k|\hat{x}_{k|k}, z_d)$. It is in general not globally optimal [i.e., $E^*(x_k|\hat{x}_{k|k}, z_d) \neq E^*(x_k|z^k, z_d)$] because the measurements $z^k$ and $z_d$ of state $x_k$ have correlated measurement noise. Of course, under some conditions, $E^*(x_k|\hat{x}_{k|k}, z_d) = E^*(x_k|z^k, z_d)$ holds. Also

$$E^*(x_k|\hat{x}_{k|k}, z_d) = \hat{x}_{k|k} + C_{x_k, \tilde{z}_{d|\hat{x}_{k|k}}} C^{-1}_{\tilde{z}_{d|\hat{x}_{k|k}}} \tilde{z}_{d|\hat{x}_{k|k}}$$

where

$$\tilde{z}_{d|\hat{x}_{k|k}} = z_d - \bar{z}_d - C_{z_d, \hat{x}_{k|k}} C^{-1}_{\hat{x}_{k|k}} (\hat{x}_{k|k} - \bar{x}_k)$$

Because $\Omega_k = \{\hat{x}_{k|k}, P_{k|k}\}$ does not sum up all prior information for this case, the LMMSE update with prior involves the prior information $\bar{x}_d, C_{x_d}$, which generally are not stored in the Kalman filter. So if we want to get the LMMSE with prior update, the information storage should increase to include the prior information. Now, not increasing our information storage in the Kalman filer, we present the LMMSE update without prior, which is derived as follows.

Let

$$z = \begin{bmatrix} \hat{x}_{k|k} \\ z_d \end{bmatrix}$$

and treat $z$ as the observation of $x_k$. Then the LMMSE estimator $\hat{x}_{k|k,d}$ of $x_k$ must be a linear combination of $\hat{x}_{k|k}$ and $z_d$, i.e., a linear function of $z$

$$\hat{x}_{k|k,d} = Kz + b$$

We obtain the optimal $K$ and $b$ by satisfying the unbiasedness assumption and minimizing the MSE matrix. According to the unbiasedness $E(x_k) = E(\hat{x}_{k|k,d})$ requirement, we have, by (1)-(2),

$$F_{k,d}\bar{x}_d = KH\bar{x}_d + b$$

where

$$H = \begin{bmatrix} F_{k,d} \\ H_d \end{bmatrix}$$

i.e.

$$(KH - F_{k,d})\bar{x}_d + b = 0$$

Since the prior information is not known, this equation must be satisfied for every $\bar{x}_d$, and so

$$KH = F_{k,d},\ b = 0 \tag{15}$$

A solution of (15) always exists, because $KH = F_{k,d}$ holds at least for $K = [I, 0]$. The next step is to obtain the optimal $K$ by minimizing the MSE matrix under linear constraint $KH = F_{k,d}$:

$$\begin{aligned} K &= \arg\min_K \mathrm{MSE}(\hat{x}_{k|k,d}) = \arg\min_K \{(K-\Gamma)R(K-\Gamma)'\} \\ s.t.\ \ KH &= F_{k,d} \end{aligned} \tag{16}$$

where the last equality in (16) follows from a tedious derivation (see Appendix C), and

$$\begin{aligned} \Gamma &= \begin{bmatrix} F_{k,d}U'_{k,d} + Q_{k,d} - P_{k|k} & 0 \end{bmatrix} R^+ \\ R &= \begin{bmatrix} F_{k,d}U'_{k,d} + U_{k,d}F'_{k,d} + Q_{k,d} - P_{k|k} & 0 \\ 0 & R_d \end{bmatrix} \end{aligned}$$

The general solution, expressed in terms of the MP-inverse, is given by:

$$K = \tilde{K} + \xi T$$

where

$$\tilde{K} = F_{k,d}H^+ + (\Gamma - F_{k,d}H^+)R(TRT)^+,\ \ T = I - HH^+$$

and $\xi$ is any matrix satisfying $\xi T R^{1/2} = 0$. So the estimate of $x_k$ is

$$\begin{aligned} \hat{x}_{k|k,d} &= Kz \\ P_{k|k,d} &= \mathrm{MSE}(\hat{x}_{k|k,d}) = (K-\Gamma)R(K-\Gamma)' + Q_{k,d} - \Gamma R \Gamma' \end{aligned}$$

Note that

$$\begin{aligned} E(\xi T z) &= E[\xi T(Hx_d - v)] = E(\xi T v) = 0 \\ \mathrm{cov}(\xi T z) &= \xi T R T \xi = 0 \end{aligned}$$

Although $K$ is not unique, the estimate of $x_k$ is unique, given by

$$\begin{aligned} \hat{x}_{k|k,d} &= \tilde{K}z \\ P_{k|k,d} &= (\tilde{K}-\Gamma)R(\tilde{K}-\Gamma)' + Q_{k,d} - \Gamma R \Gamma' \end{aligned} \tag{17}$$

**Remark 1** When $R$ matrix is non-singular, according to [9], we have

$$H^+[I - R(TRT)^+] = (H'R^{-1}H)^+ H'R^{-1}$$

Then

$$\tilde{K} = F_{k,d}(H'R^{-1}H)^{+}H'R^{-1} + \Gamma R(TRT)^{+}$$

**Remark 2** For invertible $F_{k,d}$, the LMMSE estimate of $x_k$ without prior is given by the following theorem.

**Theorem 4.3.1**: *For non-singular* $F_{k,d}$, we have

$$\hat{x}_{k|k,d} = \hat{x}^c_{k|k,d}, \;\; P_{k|k,d} = P^c_{k|k,d}$$

where

$$\hat{x}^c_{k|k,d} = \tilde{K}^c z, \;\; P^c_{k|k,d} = \tilde{K}^c R^c \tilde{K}^c$$

and

$$\begin{aligned}
&\tilde{K}^c = H^{c+}[I - R^c(T^cR^cT^c)^{+}], \;\; T^c = I - H^c(H^c)^{+} \\
&R^c = \begin{bmatrix} P_{k|k} & (P_{k|k}F^{-1\prime}_{k,d} - U_{k,d})H'_d \\ H_d(F^{-1}_{k,d}P_{k|k} - U'_{k,d}) & R_d + H_dF^{-1}_{k,d}Q_{k,d}F^{-1\prime}_{k,d}H'_d \end{bmatrix}, \;\; H^c = \begin{bmatrix} I \\ H_dF^{-1}_{k,d} \end{bmatrix}
\end{aligned}$$

When $R^c$ is invertible, by [9], it becomes

$$\begin{aligned}
\hat{x}^c_{k|k,d} &= (H^{c\prime}R^{c-1}H^c)^{-1}H^{c\prime}R^{c-1}z \\
P^c_{k|k,d} &= (H^{c\prime}R^{c-1}H^c)^{-1}
\end{aligned} \tag{18}$$

**Proof.** see Appendix D.

Obviously, for invertible $F_{k,d}$ and $R^c$, if the update is only within one step, (18) is the solution given by [7, 1]. In the multi-step update case, (18) is consistent with [11]. Thus we can say that these algorithms for update with $\Omega_k = \{\hat{x}_{k|k}, P_{k|k}\}$ and OOSM are optimal in the LMMSE sense. As such, we have proven the optimality of these existing algorithms. In Theorem 4.3.1, we have shown that $\{\hat{x}^c_{k|k,d}, P^c_{k|k,d}\}$ is a special case of our general LMMSE estimator $\{\hat{x}_{k|k,d}, P_{k|k,d}\}$ when $F_{k,d}$ is nonsingular.

In Algorithm II, the estimator contains a term $U_{k,d}$. According to Algorithm I, $U_{k,d}$ has the following recursion.

At each recursion $n$ ($n \geq k - l + 1$)

$$U_{n+1,d} = (I - K_{n+1}H_{n+1})F_{n+1,n}U_{n,d} \tag{19}$$

with initial value

$$U_{k-l+1,d} = (I - K_{k-l+1}H_{k-l+1})F_{k-l+1,d}P_{d|k-l} \tag{20}$$

where

$$P_{d|k-l} = F_{d,k-l}P_{k-l|k-l}F'_{d,k-l} + Q_{d,k-l}$$

Based on this recursion, it is easy to get that $U_{k,d}$ is highly related with the occurrence time of OOSM through $P_{d|k-l}$, which is highly related with the state estimation error covariance of $x_k$ at the OOSM occurrence time. Again, the key to achieve optimality for the update lies in when and how to initialize the recursion.

According to the uncertainty of OOSM occurrence time, we also consider above three cases of *KF-OOSM* and associated information storage as Algorithm I.

**Case I: Perfect Knowledge about $t_d$ at the Next Sampling Time $t_{k-l+1}$**

Similar to Algorithm I for Case I, the *KF-OOSM* adds a recursion for $U_{n,d}$ to the traditional Kalman filter. The flowchart of *KF-OOSM* for this case has the same structure as Algorithm I for Case I, where the OOSM initialization is given by (20), OOSM recursion is given by (19), and OOSM update by (17). The update part has the form of Remark 1 or 2 if the condition is satisfied. Since the Kalman stores $\{\hat{x}_{n|n}, P_{n|n}\}$ at each recursion, the information stored at our *KF-OOSM* at each recursion $n$ $(k-l+1 \leq n \leq k)$ is

$$\Omega_n = \{\hat{x}_{n|n}, P_{n|n}, U_{n,d}\}$$

In this case, the storage is fixed as the delay $l$ increases.

**Case II: Knowing $t_{k-l} < t_d < t_{k-l+1}$ at Time $t_{k-l+1}$**

In this case, we can initialize our *KF-OOSM* using the replacement $U_{k-l+1}$ defined by

$$U_{k-l+1} = I - K_{k-l+1} H_{k-l+1} \tag{21}$$

and $U_n$ $(n > k-l+1)$ has the recursion

$$U_{n+1} = (I - K_{n+1} H_{n+1}) F_{n+1,n} U_n \tag{22}$$

so $U_{k,d}$ can be obtained by renewing $U_k$ once the OOSM $z_d$ arrives

$$U_{k,d} = U_k F_{k-l+!,d} P_{d|k-l} \tag{23}$$

The flowchart of *KF-OOSM* for this case has the same structure as Algorithm I for Case II, where the OOSM initialization is given by (21), OOSM recursion is given by (22), renew by (23) and OOSM update by (17). The *KF-OOSM* has the following information storage structure for each $n$ $(k-l+1 \leq n \leq k)$:

$$\Omega_n = \{\hat{x}_{n|n}, P_{n|n}, U_n, P_{k-l|k-l}\}$$

As the delay $l$ increases, the storage is fixed.

**Case III: Knowing Maximum Delay $s$ of OOSM**

The method is to treat all time from $t_{k-s+1}$ to $t_k$ as a possible initialization point

$$U_n^{(n)} = (I - K_n H_n) \tag{24}$$

and applying the algorithm in Case II to achieve the optimal update with the OOSM. The recursion for $U_n^{(m)}$ $(n > k-l+1,\ n-s < m < n)$ is

$$U_{n+1}^{(m)} = (I - K_{n+1} H_{n+1}) F_{n+1,n} U_n^{(m)} \tag{25}$$

so $U_{k,d}$ can be obtained by renewing $U_k^{(k-l+1)}$ once the OOSM $z_d$ arrives, such as

$$U_{k,d} = U_k^{(k-l+1)} F_{k-l+1,d} P_{d|k-l} \tag{26}$$

The flowchart of *KF-OOSM* for this case has the same structure as Algorithm I for Case III, where the OOSM initialization is given by (24), OOSM recursion is given by (25), renew by (26), and OOSM update by (17). The information needed to store in our *KF-OOSM* at each recursion $n$ $(k-s < n \leq k)$ in this case increases linearly from time $t_{k-s+1}$ to $t_{k-1}$,

$$\Omega_n = \{\hat{x}_{n|n}, P_{k-s|k-s}, \cdots, P_{n|n}, U_n^{(k-s+1)}, \cdots, U_n^{(n)}\}$$

**Remark** If the OOSM was made exactly at a previous sampling time. With the algorithm slightly changed, the memory can be saved comparing with $t_{k-l} < t_d < t_{k-l+1}$. In this case, *KF-OOSM* flowchart structure is the same as above, except the OOSM initialization is given by

$$U_n^{(n)} = (I - K_n H_n) F_{n,n-1} P_{n|n}$$

and OOSM renew by

$$U_{k,d} = U_k^{(k-l)}$$

The memory structure of our *KF-OOSM* at each recursion $n$ $(k-s < n \leq k)$ in this case is

$$\Omega_n = \{\hat{x}_{n|n}, P_{n|n}, U_n^{(k-s+1)}, \cdots, U_n^{(n)}\}$$

Algorithm II can always give the optimal update based on the information given. Algorithm II is general, and does not have any invertibility requirement of matrix $F_{k,d}$. If $F_{k,d}$ is non-singular, the expression of the solution can be simplified. Most often $R$ and $R^c$ are invertible, which leads to even more simplified results. The algorithms of [7, 1] are special cases of Algorithm II with invertible $F_{k,d}$ and $R^c$ for the one-step update case. The algorithm of [11] solves multi-step update based on invertible matrices $F_{k,d}$ and $R^c$. Therefore, we have proven that these existing algorithms are optimal in the LMMSE sense.

The algorithm C of [1] is also a special case of Algorithm II with $w_{k,d} = 0$ and invertible $F_{k,d}$. But by setting some terms to zero in the algorithms, the new estimates may or may not be the minimizer of the original problem, i.e., MSE($\hat{x}^*_{k|k,d}$) $\leq$MSE($\hat{x}_{k|k}$) may or may not hold, where $\hat{x}^*_{k|k,d}$ is the update estimation by applying Algorithm $*$.

It can be seen from the deviation of Algorithm II that the update Algorithm actually needs more information than $\{\hat{x}_{k|k}, P_{k|k}\}$ as provided by the Kalman filter, and the OOSM $z_d$. Although at the beginning we hope to update based on the current observations, i.e., the optimal linear combination of $\hat{x}_{k|k}$ and $z_d$, we need their correlation to build the linear combination weight which include both $P_{k|k}$ and $U_{k,d}$. It tells us that the valuable update which will improve the current state estimation in general can not only based on Kalman filer.

## 4.4 Update with Arbitrarily Delayed OOSMs

In all cases discussed before, we only consider the single-OOSM update problem. But there exists arbitrarily delayed multiple OOSMs for update. The OOSMs can be the measurements of the same state or different states; the OOSMs can arrive at the same or different time. The case that any OOSM arrives before the next OOSM occurrence time belongs to the single-OOSM update problem. We can solve it by sequentially applying the single-OOSM update algorithms discussed above. But in some cases, during the period between the occurrence time and arrival time of one OOSM, other OOSMs may occur. We now consider the optimal update problem in such cases, referred to as arbitrarily delayed OOSM update. In the following, we only consider the problem of update with two OOSMs. Generalization to update with more than two OOSMs is straight forward.

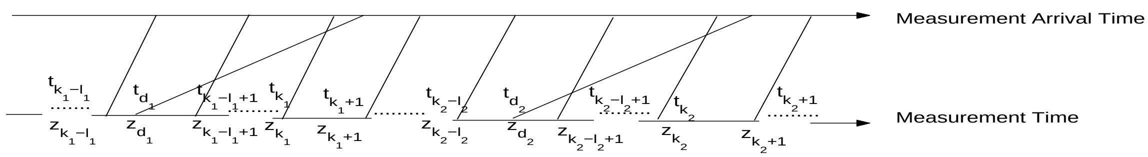


**Fig. 4.3.1 The OOSMs within the maximum delay period**

Suppose $z_{d_1}$ and $z_{d_2}$ are two OOSMs observed at $t_{k-l_i} \leq t_{d_i} < t_{k-l_i+1}$ with $1 \leq l_i < s$, $i = 1, 2$, and arrived during the time period $[t_{k_i}, t_{k_i+1})$. If $z_{d_1}$ arrives before $t_{d_2}$ (see Fig. 4.3.1), the state update with $z_{d_2}$ at its arrival time is just the single-OOSM update problem as before. $z_{d_1}$ had been used to update the state when it arrived. At $z_{d_2}$ occurrence time, there is not any other OOSMs except $z_{d_2}$. So we can directly apply Algorithm I or II for updating with the single-OOSM $z_{d_2}$ at its arrival time. If both of them arrive at the same time, although we can update the state estimate with them stacked together, computationally and operationally, it is better to update with the OOSM one by one sequentially.

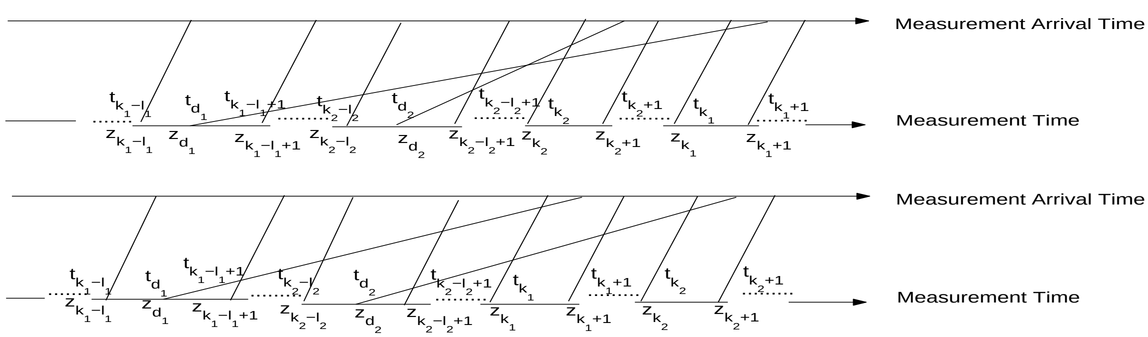


**Fig. 4.3.2 The OOSMs within the maximum delay period**

In the following, we will consider the case that $z_{d_1}$ arrives after $t_{d_2}$ (see Fig. 4.3.2). Suppose $z_{d_2}$ arrives before $z_{d_1}$, or we process $z_{d_2}$ before $z_{d_1}$ if both of them arrive at the same time. According to Algorithm I and II for single-OOSM update, we need to update the state estimation $\hat{x}_{k_2|k_2}$ and $P_{k_2|k_2}$ with $z_{d_2}$ when it arrived. At the same time, we also need to update other quantities, such as $\{\hat{x}_{d_1|k_2}, P_{d_1|k_2}, U_{k_2,d_1}\}$ for AlG-I, $\{U_{k_2,d_1}\}$ for ALG-II with $z_{d_2}$. Because $z_{d_1}$ has not arrived yet, the quantities are necessary for updating the state estimate with later arrived $z_{d_1}$. Based on different update procedures at $z_{d_2}$ arrival time, we consider two LMMSE optimal updates cases: (a) globally optimal and (b) constrained optimal.

### 4.4.1 Globally Optimal Update

In this globally optimal update case, when $z_{d_2}$ arrives, we need to update not only $\{\hat{x}_{k_2|k_2}, P_{k_2|k_2}\}$ with $z_{d_2}$, but also $\{\hat{x}_{d_1|k_2}, P_{d_1|k_2}, U_{k_2,d_1}\}$ used to update with the next $z_{d_1}$. Denote the updated quantities as $\{\hat{x}_{k_2|k_2,d_2}, P_{k_2|k_2,d_2}\}$ and $\{\hat{x}_{d_1|k_2,d_2}, P_{d_1|k_2,d_2}, U^*_{k_2,d_1}\}$. Update from $\{\hat{x}_{k_2|k_2}, P_{k_2|k_2}\}$ to $\{\hat{x}_{k_2|k_2,d_2}, P_{k_2|k_2,d_2}\}$ is trivial, it can be done by directly applying single-OOSM globally optimal update algorithm. Here we focus on the update from $\{\hat{x}_{d_1|k_2}, P_{d_1|k_2}, U_{k_2,d_1}\}$ to $\{\hat{x}_{d_1|k_2,d_2}, P_{d_1|k_2,d_2}, U^*_{k_2,d_1}\}$.

By definition

$$\hat{x}_{d_1|k_2,d_2} = E^*(x_{d_1}|z^{k_2}, z_{d_2}), \quad P_{d_1|k_2,d_2} = \text{MSE}(\hat{x}_{d_1|k_2,d_2}), \quad U^*_{k_2,d_1} = C_{x_{k_2}, \tilde{x}_{d_1|k_2,d_2}}$$

According to the recursive LMMSE, we have

$$\begin{aligned}
\hat{x}_{d_1|k_2,d_2} &= \hat{x}_{d_1|k_2} + C^{k_2}_{d_2,d_1} H'_{d_2} (H_{d_2} P_{d_2|k_2} H'_{d_2})^{-1} \tilde{z}_{d_2|k_2} \\
P_{d_1|k_2,d_2} &= P_{d_1|k_2} - C^{k_2}_{d_2,d_1} H'_{d_2} (H_{d_2} P_{d_2|k_2} H'_{d_2})^{-1} H_{d_2} (C^{k_2}_{d_2,d_1})' \\
U^*_{k_2,d_1} &= U_{k_2,d_1} + U_{k_2,d_2} H'_{d_2} (H_{d_2} P_{d_2|k_2} H'_{d_2})^{-1} H_{d_2} (C^{k_2}_{d_2,d_1})'
\end{aligned}$$

where

$$\tilde{z}_{d_2|k_2} = z_{d_2} - H_{d_2} \hat{x}_{d_2|k_2}, \quad C^{k_2}_{d_2,d_1} = C_{x_{d_1}, \tilde{x}_{d_2|k_2}}$$

Let

$$C^n_{d_2,d_1} = C_{x_{d_1}, \tilde{x}_{d_2|n}}$$

Because $C^n_{d_1,d_2} = (C^n_{d_2,d_1})'$ (see Appendix E), we can always suppose $t_{d_2} > t_{d_1}$. In this situation $C^n_{d_2,d_1}$ has a recursion for $k_2 - l_2 + 1 \leq n \leq k_2$, which has the form

$$C^n_{d_2,d_1} = C^{n-1}_{d_2,d_1} - U'_{n-1,d_1} F'_{n,n-1} H'_n S^{-1}_n H_n F_{n,n-1} U_{n-1,d_2} \tag{27}$$

with initial value

$$C^{k_2-l_2}_{d_2,d_1} = F_{d_2,k_2-l_2} U_{k_2-l_2,d_1}$$

After this update procedure, we will have $\{\hat{x}_{k_2|k_2,d_2}, P_{k_2|k_2,d_2}\}$ and $\{\hat{x}_{d_1|k_2,d_2}, P_{d_1|k_2,d_2}, U^*_{k_2,d_1}\}$. Through *KF-OOSM*, at $z_{d_1}$ arrival time, we will get $\{\hat{x}_{k_1|k_1,d_2}, P_{k_1|k_1,d_2}\}$ and $\{\hat{x}_{d_1|k_1,d_2}, P_{d_1|k_1,d_2}, U^*_{k_1,d_1}\}$, where $U^*_{k_1,d_1} = C_{x_{k_1}, \tilde{x}_{d_1|k_1,d_2}}$. Now, the single-OOSM globally optimal update Algorithm I can be directly applied to obtain $\{\hat{x}_{k_1|k_1,d_2,d_1}, P_{k_1|k_1,d_2,d_1}\}$.

It is easy to see that this OOSM update is just a sequential application of the single-OOSM globally optimal update algorithm except that at each OOSM arrival point, we need update not only the state estimate but also some other necessary quantities prepared to update other OOSMs which are not arrived yet. This update contains a new term $C_{d_2,d_1}^{k_2}$ (i.e., the correlation between two OOSMs) which fortunately has a recursive form. So very similar to Algorithm I for single OOSM update case, we can have *KF-OOSM* for the three different cases considered before.

#### 4.4.2 Constrained Optimal Update

In this constrained optimal update case, when $z_{d_2}$ arrives, we need to update not only $\{\hat{x}_{k_2|k_2}, P_{k_2|k_2}\}$ with $z_{d_2}$, but also $\{U_{k_2,d_1}\}$. Denote the updated quantities as $\{\hat{x}_{k_2|k_2,d_2}, P_{k_2|k_2,d_2}\}$ and $\{\bar{U}_{k_2,d_1}\}$. Update from $\{\hat{x}_{k_2|k_2}, P_{k_2|k_2}\}$ to $\{\hat{x}_{k_2|k_2,d_2}, P_{k_2|k_2,d_2}\}$ is trivial by directly applying single-OOSM constrained optimal update algorithm. Now, we focus on how to update from $\{U_{k_2,d_1}\}$ to $\{\bar{U}_{k_2,d_1}\}$.

As derived in Appendix F, we have

$$\bar{U}_{k_2,d_1} = C_{x_{d_1},\tilde{x}_{k_2|k_2,d_2}} = \begin{bmatrix} U_{k_2,d_1} & 0 \end{bmatrix} \tilde{K}' \tag{28}$$

where

$$\begin{aligned}
\tilde{K} &= F_{k_2,d_2}H^+ + (\Gamma - F_{k_2,d_2}H^+)R(TRT)^{-1}, \ T = I - HH^+ \\
\Gamma &= \begin{bmatrix} F_{k_2,d_2}\bar{U}_{k_2,d_2} + Q_{k_2,d_2} - P_{k_2|k_2,d_2} & 0 \end{bmatrix} R^+ \\
R &= \begin{bmatrix} \begin{matrix} F_{k_2,d_2}\bar{U}_{k_2,d_2} + \bar{U}_{k_2,d_2}F'_{k_2,d_2} + \\ Q_{k_2,d_2} - P_{k_2|k_2,d_2} \end{matrix} & 0 \\ 0 & R_{d_2} \end{bmatrix}, \quad H = \begin{bmatrix} F_{k_2,d_2} \\ H_{d_2} \end{bmatrix}
\end{aligned}$$

In fact, $\tilde{K}$ is the gain matrix for update with $z_{d_2}$ when it arrives.

Thus after this update procedure, we have $\{\hat{x}_{k_2|k_2,d_2}, P_{k_2|k_2,d_2}\}$ and $\{\bar{U}_{k_2,d_1}\}$. Through recursion, at $z_{d_1}$ arrival time, we can have $\{\hat{x}_{k_1|k_1,d_2}, P_{k_1|k_1,d_2}\}$ and $\{U^*_{k_1,d_1}\}$, where $U^*_{k_1,d_1} = C_{x_{d_1},\tilde{x}_{k_1|k_1,d_2}}$. Now, the single-OOSM constrained optimal update Algorithm II can be directly applied to obtain state estimate $\hat{x}_{k_1|k_1,d_2,d_1}$ and $P_{k_1|k_1,d_2,d_1}$.

This constrained OOSM update is just the sequential single-OOSM constrained optimal update procedure. So all previous conditions that will simplify the estimation formulas can be derived directly here. The three cases for different uncertainty of the OOSMs occurrence time can be considered in the same way as the single-OOSM case.

## 5 Numerical Examples

Several simple numerical examples are given in this section to verify the formulas presented and the existence of the optimal solution in the case where the state transition matrix is not invertible. All these examples are for the following linear system

$$x_j = F_{j-1}x_{j-1} + w_{j-1}$$
$$z_j = H_j x_j + v_j$$

where $x_j = \left[x_j^{(1)}, x_j^{(2)}\right]'$, and $w_j$ and $v_j$ are zero mean white Gaussian noise. In order to consider multi-lag delay as well as single-lag delay update, we choose a series of OOSMs $z_d$, these OOSM occurred at $d = (l+1)n$ and arrived at $(l+1)n+l$ with $n = 1, 2, \ldots$, which corresponding to $l$-lag delayed OOSMs, where $l = 1, 2, \ldots$. For example, suppose the in-sequence observation series is $\{z_1, z_2, z_3, \ldots\}$, then the observation series with OOSMs for $l = 1$ is $\{z_1, z_3, z_2, z_5, z_4, \ldots\}$ and the updated states are $x_3, x_5, x_{7,} \ldots$; the observation series with OOSMs for $l = 2$ is $\{z_1, z_2, z_4, z_3, z_5, z_7, z_8, z_6, \ldots\}$ and the updated states are $x_4, x_8, x_{11,} \ldots$; and so on. Globally optimal estimates were obtained by the Kalman filter using all observations (including OOSMs) in the right time sequence.

In the result, we use **mser**= $\frac{\text{trace}[P_{k|k,d}(\text{KF})]}{\text{trace}[P_{k|k,d}(\text{Algorithm})]}$, which is the ratio of the mean-square error of the globally optimal Kalman filter to that of the algorithm under consideration. It shows the efficiency of the algorithm. It is in the interval of $(0, 1]$. The larger the **mser** is, the better the algorithm is.

## 5.1 Nonsingular $F_{k,d}$

Consider a discretized continuous time kinematic system driven by white noise with power spectral density $q$, known as constant velocity model or white-noise acceleration model in target tracking, described by

$$F_j = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad H_j = [1, 0]$$
$$C_{w_j} = Q = \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix} q, \quad C_{v_j} = R = 1$$

where $T$ is the sampling interval. The prior information is

$$\hat{x}_{0|0} = \bar{x} = [200\ Km, 0.5\ Km/\sec]', \quad P_{0|0} = \begin{bmatrix} R & R/T \\ R/T & 2R/T^2 \end{bmatrix}$$

and the maneuver index is $\lambda = \sqrt{qT^3/R}$.

### 5.1.1 Single-Step Update ($l = 1$)

In this example, we first apply our globally optimal update algorithm in Case I and II. Then we apply Algorithm B of [1], referred to as ALG-B, to compare with our optimal update Algorithm II with limited information, referred to as ALG-II. Although there are two outputs (filtering output or smoothed output) possible from the framework of [5], smoothed output algorithm needs use future observations to estimate the current state. It is unfair to compare the smoothed output update result with that of other algorithms by update only with the observations up to current time. In order to have a fair comparison, we only apply the filtering output Algorithm of [5], referred to as ALG-S. For the case $t_{k-1} < t_d < t_k$, in order to apply ALG-S, we approximate $t_d$ to the nearest sampling time $t_{k-1}$. In Table I, we present **mser** of all algorithms at time $j = 3$. We adopt the same values of $(\lambda, q)$ as in [1], [5], [11].

Table 1: **mser**s of algorithms

| $(\lambda, q)$ | KF | KF* | ALG-I | ALG-II | ALG-B | ALG-S |
|---|---|---|---|---|---|---|
| (2,2) | 1 | 0.8786 | 1 | 0.9680 | 0.9680 | 0.7071 |
| (1,1) | 1 | 0.9121 | 1 | 0.9790 | 0.9790 | 0.8163 |
| (0.5,0.5) | 1 | 0.8787 | 1 | 0.9999 | 0.9999 | 0.9283 |

In Table 1, KF stands for the globally optimal Kalman filter; KF* stands for the Kalman filter without using OOSM. It is easy to see that our Algorithm I in Cases I and II give the globally optimal update. While Algorithm II gives the same update as Algorithm B of [1]. But the update using the algorithm of [5] is not optimal and sometimes can be worse than without using OOSM, i.e., **mser**(ALG-S) < **mser**(KF*) [see the cases $(\lambda, q) = (2, 2)$ and $(\lambda, q) = (1, 1)$]. It is caused by the error arising from the approximation of the OOSM occurrence time $t_d$. For $(\lambda, q) = (0.5, 0.5)$, the sampling interval $T$ becomes smaller, ALG-S becomes better.

In Figure 1, we show the theoretical and sample **mser** of ALG II for $(\lambda, q) = (2, 2)$. The two curves match each other very well, which verifies our update formula.

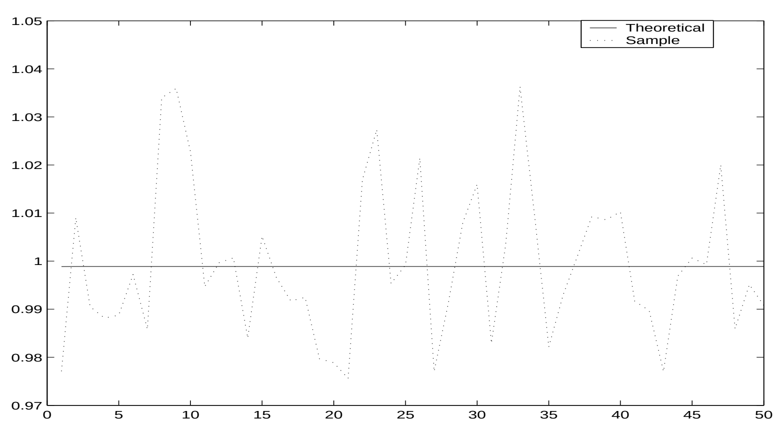


Figure 1: Theoretical and sample **mser**

It is reasonable to require that any algorithm which updates $\hat{x}_{k|k}$ with an OOSM $z_d$ to yield $\hat{x}_{k|k,d}$ should satisfy

$$\mathrm{MSE}(\hat{x}^*_{k|k,d}) \leq \mathrm{MSE}(\hat{x}_{k|k}) \tag{29}$$

Otherwise update by the algorithm is questionable. All our algorithms satisfy (29), because they are optimal estimators that minimize MSE based on the information given, but not Alg-S of [5].

### 5.1.2 Multi-Step Update

In this example, first consider $l = 2$ with $d = 3n$ and $(\lambda, q) = (2, 2)$, we apply our globally optimal update Algorithm I in all three cases and yield **mser**= 1, which verifies the global optimality of the algorithm. Also, we apply Algorithm II and compare the **mser** with the algorithm of [11], referred to as ALG-M. Figure 2 shows that the sample **mser** of ALG II matches its theoretical **mser**. ALG-II and ALG-M have the same theoretical **mser**. Meanwhile it shows the benefit of updating by comparing with KF*.

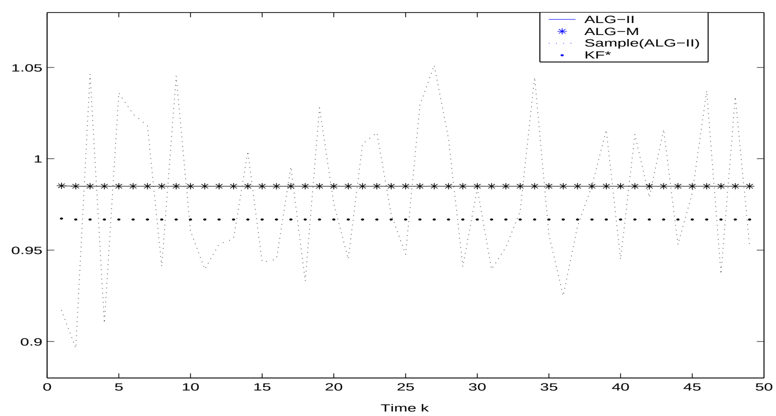


Figure 2: Comparison of **mser**

Table 2 shows the benefit of updating when $(\lambda, q) = (2, 2)$, $l = 1, \cdots, 3$ with $d = 2n, 3n, \cdots$ and $6n$ respectively, which shows that as the lag $l$ of OOSM becomes larger, the benefit of updating becomes smaller. Results of KF* for Kalman filter by ignoring OOSM show that the effect of OOSM fades quickly with the target maneuvering behaviors. It provides a strong hint for the maximum delay $s$ to consider, in addition to physical considerations.

Table 2: **mser**s of algorithms

| **mser** | $d = 2n$ | $d = 3n$ | $d = 4n$ | $d = 5n$ | $d = 6n$ |
|---|---|---|---|---|---|
| ALG II(ALG M) | 0.9680 | 0.9738 | 0.9976 | 0.9999 | 1 |
| KF* | 0.8786 | 0.9667 | 0.9972 | 0.9999 | 1 |

## 5.2 Singular $F_{k,d}$

A system with a singular state transition matrix $F_{k,d}$ is not common in practice because most discrete time systems are discretized from continuous systems. However in some cases, when a practical system is defined directly in discrete time, state transition matrix $F_{k,d}$ may be singular. The corresponding OOSM update problem needs to be considered. Also, allowing $F_{k,d}$ to be singular provides additional flexibility to handle some artificial system models, just like the study of noncausal systems, which is meaningful.

Here we consider a system with $H = [1, 1]$, $Q = 0.2I$, $R = 0.3I$ and prior information $\hat{x}_{0|0} = \bar{x} = [1, 1]'$, $P_{0|0} = 0.001I$.

### 5.2.1 Single-Step Update ($l = 1$)

In this case, we use

$$F_{2n-1} = \frac{1}{n^2}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \; F_{2n} = \begin{bmatrix} 1 & -\frac{1}{n^2} \\ -1 & \frac{1}{n^2} \end{bmatrix}$$

Algorithm I in Case I or II can always gets **mser**= 1. As shown in Figure 3, the theoretical and sample **mser** of ALG II match each other, which verifies the formula.

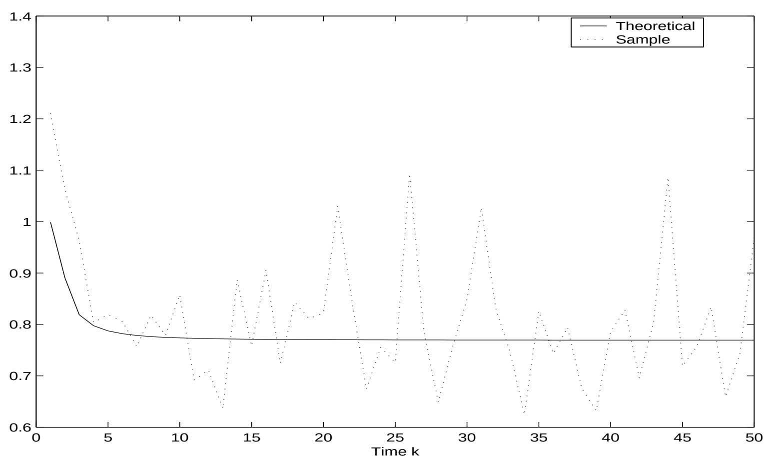


Figure 3: Theoretical and Sample **mser**

### 5.2.2 Multi- step Update ($l = 2$)

In this case, $k = 3n + 2$, we use

$$F_{3n} = \begin{bmatrix} 1 & 1/n \\ -1 & -1/n \end{bmatrix}, \; F_j = \begin{bmatrix} 1 & 1/j \\ 0 & 1 \end{bmatrix} \quad j \neq 3n$$

Algorithm I in all three cases all gave **mser**= 1. As shown in Figure 4, ALG II can give the benefit of updating, as shown by comparing **mser** with KF*.

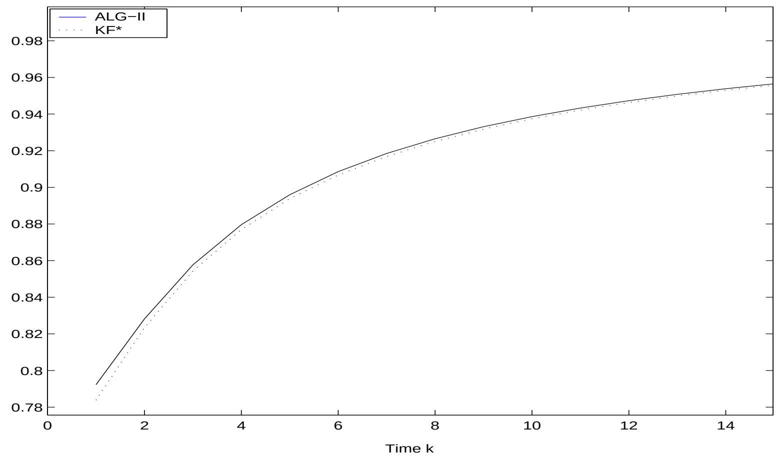


Figure 4: Comparison of **mser**

Examples in this section verify both Algorithms I and II. They show that our proposed Algorithms I and II are more general. They can solve single-step as well as multi-step update. Also, for singular state transition matrix $F_{k,d}$, they are still efficient.

# 6 Summary

We have presented two general algorithms with three cases of different information storage for state estimation update with out-of-sequence measurements. Both algorithms are optimal in the LMMSE sense for the information given and are more general than previously available algorithms. In particular, they are optimal for multiple-step as well as single-step update; they do not have any non-singularity requirement on the matrix $F_{k,d}$; they yield the best unbiased estimates among all linear estimation algorithms. Algorithm I is always globally optimal (in the LMMSE sense). Algorithm II is optimal (in the LMMSE sense) for the information given. Under the linear-Gaussian assumption of the Kalman filtering, both algorithms give the conditional mean.

Both algorithms need the smallest storage in Case I, the largest storage in Case III. The storage of Algorithm II for all cases are smaller than Algorithm I. For single-step update, the information stored is even smaller. Each item in the algorithms has a recursive form and can be computed easily, as presented. As illustrated by the simulation results, these variants in information storage complement each other in that they are suitable for different practical situations and yield the same optimal update.

Overall, both Algorithms have (1) an efficient processing structure for information update; (2) an efficient memory structure for storing historical information; (3) an efficient computational structure, and thus (4) an easy generalization for arbitrarily delayed multiple OOSMs. In this paper, we focus on the OOSM filtering problem. In future work, we would like to consider solving the multi-sensor OOSM problem in a cluttered environment [12][13][5].

**Appendix A:** Derivation of (5)-(8).

The recursion for $\{\hat{x}_{d|n}, P_{d|n}, U_{n,d}\}$ is generated as follows. For $n \geq k-l+1$

$$\begin{aligned}
\hat{x}_{d|n+1} &= E^*(x_d|z^{n+1}) = \hat{x}_{d|n} + C_{x_d,\tilde{x}_{n|n}} F'_{n+1,n} H'_{n+1} S^+_{n+1} \tilde{z}_{n+1|n} \\
P_{d|n+1} &= P_{d|n} - C_{x_d,\tilde{x}_{n|n}} F'_{n+1,n} H'_{n+1} S^+_{n+1} H_{n+1} F_{n+1,n} C'_{x_d,\tilde{x}_{n|n}} \\
U_{n+1,d} &= C_{x_{n+1},\tilde{x}_{d|n+1}} = C_{x_{n+1},\tilde{x}_{d|n} - C_{x_d,\tilde{x}_{n|n}} F'_{n+1,n} H'_{n+1} S^+_{n+1} \tilde{z}_{n+1|n}} \\
&= C_{x_{n+1},\tilde{x}_{d|n}} - C_{x_{n+1},\tilde{z}_{n+1|n}} S^+_{n+1} H_{n+1} F'_{n+1,n} C'_{x_d,\tilde{x}_{n|n}} \\
&= F_{n+1,n} U_{n,d} - K_{n+1} H_{n+1} F_{n+1,n} C'_{x_d,\tilde{x}_{n|n}}
\end{aligned} \tag{30}$$

**Theorem**

$$C_{x_n,\tilde{x}_{d|n}} = C^{'}_{x_d,\tilde{x}_{n|n}} \tag{31}$$

Proof:

$$C_{x_n,\tilde{x}_{d|n}} = C_{x_n - \hat{x}_{n|n}, x_d - \hat{x}_{d|n}} = C_{x_n - \hat{x}_{n|n}, x_d} = C_{\tilde{x}_{n|n}, x_d} = C^{'}_{x_d,\tilde{x}_{n|n}}$$

Thus (31) holds. According to the above Theorem, recursion (30) can be simplified as

$$\begin{aligned}
\hat{x}_{d|n+1} &= \hat{x}_{d|n} + U'_{n,d} F'_{n+1,n} H'_{n+1} S^+_{n+1} \tilde{z}_{n+1|n} \\
P_{d|n+1} &= P_{d|n} - U'_{n,d} F'_{n+1,n} H'_{n+1} S^+_{n+1} H_{n+1} F_{n+1,n} U'_{n,d} \\
U_{n+1,d} &= (I - K_{n+1} H_{n+1}) F_{n+1,n} U'_{n,d}
\end{aligned}$$

with initial value

$$\begin{aligned}
\hat{x}_{d|k-l+1} &= \hat{x}_{d|k-l} + P_{d|k-l} F'_{k-l+1,d} H'_{k-l+1} S^+_{k-l+1} \tilde{z}_{k-l+1|k-l} \\
P_{d|k-l+1} &= P_{d|k-l} - P_{d|k-l} F'_{k-l+1,d} H'_{k-l+1} S^+_{k-l+1} H_{k-l+1} F_{k-l+1,d} P_{d|k-l} \\
U_{k-l+1,d} &= (I - K_{k-l+1} H_{k-l+1}) F_{k-l+1,d} P_{d|k-l}
\end{aligned}$$

**Appendix B:** Relationship between $\{\hat{x}_{d|k}, P_{d|k}, U_{k,d}\}$ and $\{y_k, B_k, U_k\}$.

Now

$$\begin{aligned}
\hat{x}_{d|k} &= \hat{x}_{d|k-1} + U'_{k-1,d} F'_{k,k-1} H'_k S^{-1}_k \tilde{z}_{k|k-1} \\
P_{d|k} &= P_{d|k-1} - U'_{k-1,d} F'_{k,k-1} H'_k S^{-1}_k H_k F_{k,k-1} U_{k-1,d} \\
U_{k,d} &= (I - K_k H_k) F_{k,k-1} U_{k-1,d}
\end{aligned}$$

$$\vdots$$

$$\begin{aligned}
\hat{x}_{d|k} &= \hat{x}_{d|k-l} + U'_{k-l,d}F'_{k-l+1,k-l}H'_{k-l+1}S^{-1}_{k-l+1}\tilde{z}_{k-l+1|k-l} + \ldots + U'_{k-1,d}F'_{k,k-1}H'_kS^{-1}_k\tilde{z}_{k|k-1} \\
P_{d|k} &= P_{d|k-l} - U'_{k-l,d}F'_{k-l+1,k-l}H'_{k-l+1}S^{-1}_{k-l+1}H_{k-l+1}F_{k-l+1,k-l}U_{k-l,d} - \ldots - U'_{k-1,d}F'_{k,k-1}H'_kS^{-1}_k \\
&\quad H_kF_{k,k-1}U_{k-1,d} \\
U_{k,d} &= [(I-K_kH_k)F_{k,k-1}] \times \ldots \times [(I-K_{k-l+1}H_{k-l+1})F_{k-l+1,k-l}]U_{k-l,d}
\end{aligned}$$

also

$$\begin{aligned}
y_k &= y_{k-1} + U'_{k-1}F'_{k,k-1}H'_kS^{-1}_k\tilde{z}_{k|k-1} \\
B_k &= B_{k-1} + U'_{k-1}F'_{k,k-1}H'_kS^{-1}_kH_kF_{k,k-1}U_{k-1} \\
U_k &= (I-K_kH_k)F_{k,k-1}U_{k-1}
\end{aligned}$$

$$\vdots$$

$$\begin{aligned}
y_k &= y_{k-l} + U'_{k-l}F'_{k-l+1,k-l}H'_{k-l}S^{-1}_{k-l}\tilde{z}_{k-l+1|k-l} + \ldots + U'_{k-1}F'_{k,k-1}H'_kS^{-1}_k\tilde{z}_{k|k-1} \\
B_k &= B_{k-l} + U'_{k-l}F'_{k-l+1,k-l}H'_{k-l}S^{-1}_{k-l}H_{k-l+1}F_{k-l+1,k-l}U_{k-l} + \ldots + U'_{k-1}F'_{k,k-1}H'_kS^{-1}_kH_kF_{k,k-1}U_k \\
U_k &= [(I-K_kH_k)F_{k,k-1}] \times \ldots \times [(I-K_{k-l+1}H_{k-l+1})F_{k-l+1,k-l}]U_{k-l}
\end{aligned}$$

and for any $m \geq k-l+1$

$$\begin{aligned}
U_{m,d} &= [(I-K_mH_m)F_{m,m-1}] \times \ldots \times [(I-K_{k-l+2}H_{k-l+2})F_{k-l+2,k-l+1}]U_{k-l+1,d} \\
U_m &= [(I-K_mH_m)F_{m,m-1}] \times \ldots \times [(I-K_{k-l+2}H_{k-l+2})F_{k-l+2,k-l+1}]U_{k-l+1}
\end{aligned}$$

By comparing the initial value $\{\hat{x}_{d|k-l+1},P_{d|k-l+1},U_{k-l+1,d}\}$ and $\{y_{k-l+1},B_{k-l+1},U_{k-l+1}\}$, we have

$$U_{m,d} = U_mF_{k-l+1,d}P_{d|k-l}$$

and

$$\begin{aligned}
&P_{d|k-l}F'_{k-l+1,d}y_k = P_{d|k-l}F'_{k-l+1,d}y_{k-l+1} + \hat{x}_{d|k} - \hat{x}_{d|k-l+1} \\
&= P_{d|k-l}F'_{k-l+1,d}H'_{k-l+1}S^{-1}_{k-l+1}\tilde{z}_{k-l+1|k-l} + \hat{x}_{d|k} - \hat{x}_{d|k-l+1} \\
&= P_{d|k-l}F'_{k-l+1,d}H'_{k-l+1}S^{-1}_{k-l+1}\tilde{z}_{k-l+1|k-l} + \hat{x}_{d|k} - \hat{x}_{d|k-l} - P_{d|k-l}F'_{k-l+1,d}H'_{k-l+1}S^{+}_{k-l+1}\tilde{z}_{k-l+1|k-l} \\
&= \hat{x}_{d|k} - \hat{x}_{d|k-l}
\end{aligned}$$

$$\begin{aligned}
&P_{d|k-l}F'_{k-l+1,d}B_kF_{k-l+1,d}P_{d|k-l} = P_{d|k-l}F'_{k-l+1,d}B_{k-l+1}F_{k-l+1,d}P_{d|k-l} - P_{d|k} + P_{d|k-l+1} \\
&= P_{d|k-l}F'_{k-l+!,d}H'_{k-l+1}S^{-1}_{k-l+1}H_{k-l+1}F_{k-l+1,d}P_{d|k-l} - P_{d|k} + P_{d|k-l} - P_{d|k-l}F'_{k-l+1,d}H'_{k-l+1}S^{-1}_{k-l+1} \\
&H_{k-l+1}\ F_{k-l+1,d}P_{d|k-l} = P_{d|k-l} - P_{d|k}
\end{aligned}$$

Thus

$$\begin{aligned}
\hat{x}_{d|k} &= P_{d|k-l}F'_{k-l+1,d}y_k + \hat{x}_{d|k-l} \\
P_{d|k} &= P_{d|k-l} - P_{d|k-l}F'_{k-l+1,d}B_kF_{k-l+1,d}P_{d|k-l} \\
U_{k,d} &= U_kF_{k-l+1,d}P_{d|k-l}
\end{aligned}$$

**Appendix C:** Derivation of (16).

$$\begin{aligned}
&MSE(\hat{x}_{k|k,d}) = \mathrm{cov}(x_k - \hat{x}_{k|k,d}) = \mathrm{cov}(x_k - Kz) = \mathrm{cov}(F_{k,d}x_d + w_{k,d} - K\begin{bmatrix} \hat{x}_{k|k} \\ z_d \end{bmatrix}) \\
&= \mathrm{cov}(F_{k,d}x_d + w_{k,d} - K\begin{bmatrix} x_k - \tilde{x}_{k|k} \\ H_dx_d + v_d \end{bmatrix}) = \mathrm{cov}(F_{k,d}x_d + w_{k,d} - K\begin{bmatrix} F_{k,d}x_\tau + w_{k,d} - \tilde{x}_{k|k} \\ H_dx_d + v_d \end{bmatrix}) \\
&= \mathrm{cov}\{(F_{k,d} - KH)x_d + w_{k,d} - K\begin{bmatrix} w_{k,d} - \tilde{x}_{k|k} \\ v_d \end{bmatrix}\} = \mathrm{cov}(w_{k,d} - K\begin{bmatrix} w_{k,d} - \tilde{x}_{k|k} \\ v_d \end{bmatrix}) \\
&= Q_{k,d} - \begin{bmatrix} C_{w_{k,d},w_{k,d}-\tilde{x}_{k|k}} & 0 \end{bmatrix}K' - K\begin{bmatrix} C'_{w_{k,d},w_{k,d}-\tilde{x}_{k|k}} \\ 0 \end{bmatrix} + K\begin{bmatrix} C_{w_{k,d}-\tilde{x}_{k|k}} & 0 \\ 0 & R_d \end{bmatrix}K' \\
&= (K-\Gamma)R(K-\Gamma)' + Q_{k,d} - \Gamma R\Gamma'
\end{aligned}$$

where

$$\begin{aligned}
\Gamma &= \begin{bmatrix} F_{k,d}U'_{k,d} + Q_{k,d} - P_{k|k} & 0 \end{bmatrix}R^+ \\
R &= \begin{bmatrix} F_{k,d}U'_{k,d} + U_{k,d}F'_{k,d} + Q_{k,d} - P_{k|k} & 0 \\ 0 & R_d \end{bmatrix}
\end{aligned}$$

thus the optimization problem of $K$ is

$$\begin{aligned}
K &= \arg\min_K \mathrm{MSE}(\hat{x}_{k|k,d}) = \arg\min_K\{(K-\Gamma)R(K-\Gamma)'\} \\
&s.t.\quad KH = F_{k,d}
\end{aligned}$$

**Appendix D:** Proof of Theorem 4.2.1.

The LMMSE estimate in this case is

$$\hat{x}^c_{k|k,d} = K^cz + b^c$$

according to unbiasedness requirement, we have

$$\bar{x}_k = K^cH^c\bar{x}_k + b^c, \text{ i.e., } (K^cH^c - I)\bar{x}_k + b^c = 0$$

Without knowing the prior information $\bar{x}_k$ is equivalent to without knowing the prior information $\bar{x}_d$ because $F_{k,d}$ is invertible. The equation must be satisfied for every $\bar{x}_k$, and so

$$K^c H^c = I, \;\; b^c = 0$$

A solution always exists, because at least we can choose $K^c = [I, 0]$ to make $K^c H^c = I$ hold. The optimal $K^c$ is the solution of the following optimization problem with a linear constraint:

$$\begin{aligned} K^c &= \arg\min_K \text{MSE}(\hat{x}^c_{k|k,d}) = \arg\min_K K^c R^c K^{c\prime} \\ s.t. \;\; & K^c H^c = I \end{aligned}$$

By [10], we have $T^c = I - H^c(H^c)^+$ and $\tilde{K}^c = (H^c)^+[I - R^c(T^c R^c T^c)^+]$. On the other hand, obviously we have $H^c F_{k,d} = H$, and in view of $K^c H^c = I$, we have

$$K^c H^c F_{k,d} = F_{k,d}, \text{ i.e., } K^c H = F_{k,d}$$

Based on this, the MSE of $\hat{x}^c_{k|k,d}$ has another form

$$\text{MSE}(\hat{x}^c_{k|k,d}) = (K^c - \Gamma)R(K^c - \Gamma)' + Q_{k,d} - \Gamma R \Gamma'$$

So

$$K^c = \arg\min_K (K^c - \Gamma)R(K^c - \Gamma)', \;\; s.t. \;\; K^c H = F_{k,d}$$

Because the linear constrained optimization problem for $K^c$ is the same as that of $K$, we have

$$\hat{x}^c_{k|k,d} = \hat{x}_{k|k,d}, \;\; P^c_{k|k,d} = P_{k|k,d}$$

**Appendix E:** Derivation of (27).

$$\begin{aligned} C^n_{d_1,d_2} &= \text{cov}(x_{d_2}, x_{d_1} - \hat{x}_{d_1|n}) = \text{cov}(x_{d_2} - \hat{x}_{d_2|n}, x_{d_1} - \hat{x}_{d_1|n}) \\ &= \text{cov}(x_{d_2} - \hat{x}_{d_2|n}, x_{d_1}) = \text{cov}(x_{d_1}, x_{d_2} - \hat{x}_{d_2|n})' = (C^n_{d_2,d_1})' \end{aligned}$$

When $n \geq k_2 - l_2$

$$\begin{aligned} C^n_{d_2,d_1} &= \text{cov}(x_{d_1}, x_{d_2} - \hat{x}_{d_2|n}) = \text{cov}(x_{d_1}, x_{d_2} - \hat{x}_{d_2|n-1} - U'_{n-1,d_2} F'_{n,n-1} H'_n S^+_n \tilde{z}_{n|n-1}) \\ &= \text{cov}(x_{d_1}, x_{d_2} - \hat{x}_{d_2|n-1}) - \text{cov}(x_{d_1}, \tilde{x}_{n-1|n-1}) F'_{n,n-1} H'_n S^+_n H_n F_{n,n-1} U_{d_2,n-1} \\ &= C^{n-1}_{d_2,d_1} - U'_{n-1,d_1} F'_{n,n-1} H'_n S^+_n H_n F_{n,n-1} U_{n-1,d_2} \end{aligned}$$

with initial value

$$C^{k-l_2}_{d_2,d_1} = F_{d_1,k_2-l_2} U_{k_2-l_2,d_1}$$

**Appendix F:** Derivation of (28).

$$
\begin{aligned}
\bar{U}_{k_2,d_1} &= C_{x_{d_1},\tilde{x}_{k_2|k_2,d_2}} = cov(x_{d_1}, x_{k_2} - \hat{x}_{k_2|k_2,d_2}) = cov(x_{d_1}, x_{k_2} - \tilde{K}_{(2)} \begin{bmatrix} \hat{x}_{k_2|k_2} \\ z_{d_2} \end{bmatrix}) \\
&= cov(x_{d_1}, x_{k_2} - \tilde{K}_{(2)} \begin{bmatrix} x_{k_2} - \tilde{x}_{k_2|k_2} \\ z_{d_2} \end{bmatrix}) \\
&= cov(x_{d_1}, F_{k_2,d_2} x_{d_2} + w_{k_2,d_2} - \tilde{K}_{(2)} H_{(2)} x_{d_2} - \tilde{K}_{(2)} \begin{bmatrix} w_{k_2 d_2} - \tilde{x}_{k_2|k_2} \\ z_{d_2} \end{bmatrix}) \\
&= cov(x_{d_1}, w_{k_2,d_2} - \tilde{K}_{(2)} \begin{bmatrix} w_{k_2,d_2} - \tilde{x}_{k_2|k_2} \\ v_{d_2} \end{bmatrix}) = \begin{bmatrix} U_{k_2,d_1} & 0 \end{bmatrix} \tilde{K}'_{(2)}
\end{aligned}
$$

where $\tilde{K}_{(2)}$ is the gain matrix for update with $z_{d_2}$ at its arrival time.

# References


[1] Y. Bar-Shalom. Update with Out-of-Sequence Measurements in Tracking: Exact Solution. *SPIE Signal and Data Processing of Small Targets*, 4048:541–556, Apr. 2000.

[2] Y. Bar-Shalom, M. Mallick, H. Chen, and R. Washburn. One-Step Solution for the General Out-of-Sequence Measurements Problem in Tracking. In *Proc. 1st IEEE Aerospace Conf.*, Big Sky, MT, March 2002.

[3] S. S. Blackman and R. F. Popoli. *Design and Analysis of Modern Tracking Systems.* Artech House, Norwood, MA, 1999.

[4] S. Challa, R. Evans, and J. Legg. A fixed lag smoothing framework for OOSI problems. *Communications in Information and Systems*, 2(4):327–350, December 2002.

[5] S. Challa, R. Evans, and X. Wang. A Bayesian Solution to OOSM Problem and its approximations. *Information Fusion*, 4(3), September 2003.

[6] S. Challa, X. Wang, and J. Legg. Track-to-Track Fusion using Out-of-Sequence Tracks. In *Proc. International Conf. on Information Fusion*, pages 919–926, Annapolis, MD, July 2002.

[7] R. D. Hilton, D. A. Martin, and W. D. Blair. Tracking with Time-Delayed Data in Multisensor Systems. Technical Report NSWCDD/TR-93/351, Naval Surface Warfare Center, Dalgren, VA, Aug. 1993.

[8] X. R. Li and K. S. Zhang. Optimal Linear Estimation Fusion—Part IV: Optimality and Efficiency of Distributed Fusion. In *Fusion2001*, pages WeB1.19–WeB1.26, Montreal, QC, Canada, Aug. 2001.

[9] X. R. Li, K. S. Zhang, J. Zhao, and Y. M. Zhu. Optimal Linear Estimation Fusion–Part V: Relationships. In *Fusion2002*, pages 497–504, Annapolis, MD, USA, July 2002.

[10] X. R. Li, Y. M. Zhu, J. Wang, and C. Z. Han. Unified Optimal Linear Estimation Fusion—Part I: Unified Fusion Rules. *IEEE Transactions on Information Theory*, 49(9):2192–2208, Sep. 2003.

[11] M. Mallick, S. Coraluppi, and C. Carthel. Advances in Asynchronous and Decentralized Estimation. In *Proc. IEEE Aerospace Conf.*, Big Sky, MT, March 2001.

[12] M. Mallick, SJ. Krant, and Y. Bar-Shalom. Multi-sensor Multi-target Tracking using Out-of-Sequence Measurements. In *Proc. International Conf. on Information Fusion*, pages 135–142, Annapolis, MD, July 2002.

[13] M. Orton and A. Marrs. A Bayesian Approach to Multi-target Tracking Data Fusion with Out-of Sequence Measurements. *Proc. IEEE* (submitted), 2001.

[14] S. Thomopoulos and L. Zhang. Distributed Filtering with Random Sampling and Delay. In *Proc. 27th IEEE Conf. Decision and Control*, Austin, TX, Dec. 1988.

[15] K. S. Zhang, X. R. Li, and Y. M. Zhu. Optimal Update with Out-of-Sequence Measurements for Distributed Filtering. In *Fusion2002*, pages 1519–1526, Annapolis, MD, USA, July 2002.