KALMAN FILTER ALGORITHMS FOR A MULTI-SENSOR SYSTEM*

D. Willner, C. B. Chang, and K. P. Dunn
Massachusetts Institute of Technology
Lincoln Laboratory
P. O. Box 73
Lexington, Massachusetts 02173

Abstract

The purpose of this paper is to examine several Kalman filter algorithms that can be used for state estimation with a multiple sensor system. In a synchronous data collection system, the statistically independent data blocks can be processed in parallel or sequentially, or similar data can be compressed before processing; in the linear case these three filter types are optimum and their results are identical. When measurements from each sensor are statistically independent, the data compression method is shown to be computationally most efficient, followed by the sequential processing; the parallel processing is least efficient.

1. Introduction

The purpose of this paper is to introduce discrete Kalman filter algorithms [1,2] for a multiple sensor system (e.g., multiple radars at different locations observe a re-entry body). Computationally efficient filter configurations are needed when the number of sensors becomes large as may happen with a highly redundant sensor system. This paper is a simplified version of the Lincoln Laboratory Report [3] prepared by the same authors. Detailed discussions such as algorithm derivation are included in that reference.

The paper is organized as follows. In Section 2 the problem is defined in more detail. In Section 3 filters for both synchronously and randomly (in time) collected data are described. The computational efficiencies of the filters are compared in Section 4 followed by the conclusions in Section 5.

2. Problem Statement

In the multisensor system under consideration, several sensors (e.g., radars) at different locations make measurements of the same object (e.g., airplane, re-entry body). The sensor accuracies are known, their sampling times may be synchronized or random. The problem is particularly interesting as the number of sensors grows and the computational requirements increase.

For the synchronous data collection system, three types of filters are described:

1. Parallel filter where the filter is updated by processing all measurements in parallel.
2. Sequential filter, where the filter is updated sequentially with the blocks of statistically independent data.
3. Data compression filters, where similar data are compressed and the filter is then updated only once with the compressed data.

In the case of linear systems the above three filters are all equivalent and optimum. The proof is shown in detail in Reference 3. They are compared in terms of computational requirements by counting the number of multiplications for one complete filter cycle using all available data. These numbers are also derived in Reference 3.

3. Filter Derivations

In Section 3.1, the filters for synchronously collected data are derived. The filter for randomly collected data is discussed in Section 3.2.

3.1. Synchronously Collected Data

Let $z_{k+1,i}$, denote the measurement taken at time t_{k+1} from i-th sensor with a total of I sensors. Then

$$z_{k+1,i} = h_i(x_{k+1}) + v_{k+1,i} \qquad i=1,\dots.I \tag{3.1}$$

where $\{v_{k+1,i}\}$ is a white Gaussian noise sequence with zero mean and covariance $R_{k+1,i}$.

*This work was sponsored by the Department of the Army.

All measurement vectors may be used to form a new measurement vector z_{k+1}.

$$z_{k+1} = \left[z^T_{k+1,1}\ z^T_{k+1,2}, \ldots . z^T_{k+1,I} \right]^T \quad (3.2)$$

If each $z_{k+1,i}$ is an m-vector, then z_{k+1} is an M=mxI vector (otherwise $M = m_1 + m_2 + \ldots + m_I$). If the measurement noise for different sensors are uncorrelated, the covariance of z_{k+1} is

$$R_{k+1} = \left[\begin{array}{c|c|c|c|c|c} R_{k+1,1} & 0 & \cdot & \cdot & \cdot & \\ \hline 0 & R_{k+1,2} & & & & \\ \hline \cdot & & & & & \\ \cdot & & & & & \\ \cdot & & & & & \\ \hline & & & & & R_{k+1,I} \end{array} \right] \quad (3.3)$$

There are three options in processing these measurements by a Kalman filter. They are discussed individually below.

3.1.1. Parallel Filter

Using (3.2) and (3.3) in the filter update equation and after a few manipulations, we obtain:

(State)
$$\hat{x}_{k+1/k+1} = \hat{x}_{k+1/k} + \sum_{i=1}^{I} K_{k+1,i}(z_{k+1,i} - h_i(\hat{x}_{k+1/k})) \quad (3.4)$$

(Gain)
$$K_{k+1,i} = P_{k+1/k+1} H^T_{k+1,i} R^{-1}_{k+1,i} \quad (3.5)$$

(Covariance)
$$P^{-1}_{k+1/k+1} = P^{-1}_{k+1/k} + \sum_{i=1}^{I} (H^T_{k+1,i}\ R^{-1}_{k+1,i}\ H_{k+1,i}) \quad (3.6)$$

where $H_{k+1,i}$ is the Jacobian matrix of $h_i(x_{k+1})$ at $x_{k+1/k}$. Notice that the inverse covariance matrix equation is used in Eq. (3.6). This form is more convenient for discussing filter equivalence. This algorithm is depicted in Fig. 3.1.

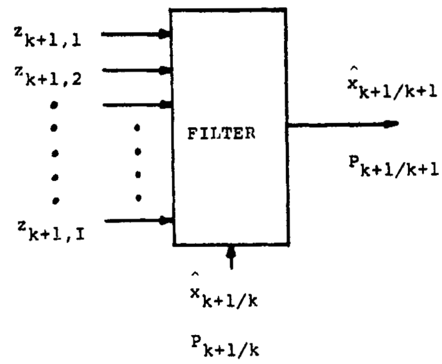

Fig. 3.1. Parallel Filter

3.1.2. Sequential Filter

Each measurement $z_{k+1,i}$ for i=1,...,I is treated as a new measurement with zero prediction time. The state estimate, gain and covariance are updated sequentially.

(State)
$$\hat{x}_{k+1/k+1,i} = \hat{x}_{k+1/k+1,i-1} + K_{k+1,i}(z_{k+1/i} - h_i(\hat{x}_{k+1/k+1,i-1})) \quad (3.7)$$

(Gain)
$$K_{k+1,i} = P_{k+1/k+1,i-1}\ H^T_{k+1,i} (H_{k+1,i} P_{k+1/k+1,i-1} H^T_{k+1,i} + R_{k+1,i})^{-1} \quad (3.8)$$

or

$$K_{k+1,i} = P_{k+1/k+1,i}\ H^T_{k+1,i}\ R^{-1}_{k+1,i} \quad (3.9)$$

(Covariance)

$$P_{k+1/k+1,i} = P_{k+1/k+1,i-1} - K_{k+1,i} H_{k+1,i} P_{k+1/k+1,i-1} \tag{3.10}$$

or

$$P^{-1}_{k+1/k+1,i} = P^{-1}_{k+1/k+1,i-1} + H^T_{k+1,i} R^{-1}_{k+1,i} H_{k+1,i} \tag{3.11}$$

for

$i = 1,2,\ldots,I$

where

$$\hat{x}_{k+1/k+1,0} = \hat{x}_{k+1/k}, \quad \hat{x}_{k+1/k+1} = \hat{x}_{k+1/k+1,I} \tag{3.12}$$

$$P_{k+1/k+1,0} = P_{k+1/k}, \quad P_{k+1/k+1} = P_{k+1/k+1,I} \tag{3.13}$$

The measurement of the i-th sensor is used to update the state estimate at the i-th step. After I steps the state estimate computes to:

$$\hat{x}_{k+1/k+1} = \hat{x}_{k+1/k} + \sum_{i=1}^{I} K_{k+1,i} (z_{k+1,i} - h_i(\hat{x}_{k+1/k+1,i-1})) \tag{3.14}$$

This algorithm is depicted in Fig. 3.2.

3.1.3. Data Compression

All similar measurements may first be combined to form a pseudomeasurement (data compression). With the introduction of fake measurements, i.e., with extremely large covariances, even nonsimilar data sets can be made to look similar.

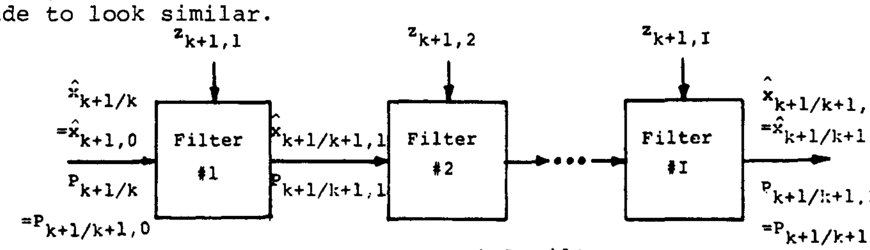

Fig. 3.2. Sequential Filter

$$z_{k+1} = R_{k+1} \left(\sum_{i=1}^{I} R^{-1}_{k+1,i} z_{k+1,i} \right) \tag{3.15}$$

$$R^{-1}_{k+1} = \sum_{i=1}^{I} R^{-1}_{k+1,i} \tag{3.16}$$

In order to use (3.15) and (3.16) all measurement vectors have to be transformed to a common coordinate system. Using the above results, the update equations are those of the standard Kalman filter. This algorithm is shown in Fig. 3.3.

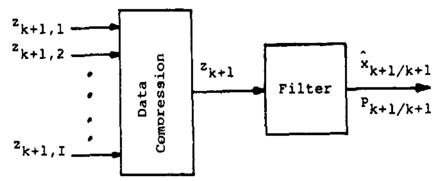

Fig. 3.3. Data Compression Filter

3.2. Non-Synchronously Collected Data

The filter prediction and update process is carried out according to the availability of new data sets. Suppose at time t_{k+1} that the only available data is from radar i and let it be denoted by $z_{k+1,i}$ and $R_{k+1,i}$, the update is performed based upon this available data.

(State)

$$\hat{x}_{k+1/k+1} = \hat{x}_{k+1/k} + K_{k+1,i}(z_{k+1,i} - h_i(\hat{x}_{k+1/k})) \tag{3.17}$$

(Gain)

$$K_{k+1,i} = P_{k+1/k+1} H^T_{k+1,i} R^{-1}_{k+1,i} \tag{3.18}$$

(Covariance)

$$P^{-1}_{k+1/k+1} = P_{k+1/k} + H^{T}_{k+1,i} R_{k+1,i} H_{k+1,i} \qquad (3.19)$$

If the filter is restricted to accept data only in a fixed measurement coordinate system, $z_{k+1,i}$ and $R_{k+1,i}$ must be first transformed into that coordinate system. The resulting update equations are the same as for the standard Kalman filter.

The draw back of a nonsynchronous data collection system is in its high computational requirements. This is caused by the fact that the filter must be updated sequentially (as in Section 3.1.2) and the data compression scheme cannot be applied.

Two alternatives exist. The first one is simply to insist on a synchronous data collection system. This is possible if sufficient communication exists between the sensors so that data can be collected synchronously. The second alternative is to preprocess the data for time alignment A polynomial data smoother could be used as a data processor for data-time-alignment similar to the one discussed in Ref. 4

4. Algorithm Comparison

In the case of a linear system it can be shown [3] that the resulting state estimates for the parallel filter, sequential filter, and data compression for synchronously collected data are identical and optimal. In Table 4.1 a cost comparison is shown (in terms of multiplications per step).*

The comparison of the algorithm is demonstrated for a particular example (n=7, m=3, I=3), where n denotes the dimension of the state vector, m the dimension of the measurement vector per sensor, and I the number of independent sensors. The randomly collected data filter requires a much larger amount of computer resources than the synchronously collected data filters. The data compression method is computationally more efficient than all the others. Although it requires that all measurements be transformed to a common coordinate, the filter needs to be updated only once. The computational requirements of the parallel filter per update are larger than those of

*The number of multiplications per filter cycle is derived in Reference 3.

TABLE 4.1

COMPARISON OF ALGORITHM EFFICIENCY

(Number of Multiplications)

DATA COLLECTION	SYNCHRONOUSLY			RANDOMLY
	Parallel Filter	Sequential Filter	Data Comp.	Sequential Filter
COMPUTATIONAL REQUIREMENTS	Moderate-High	Moderate	Low	High
	Pred. 588 Update 2181 2769* Via Eq. (3.4 - 3.6) Pred. 588 Update 1330 1918*	Pred. 588 Update 945 1533*	Pred. 588 Data Comp. 208 Update 394 1190*	Pred. 1764 Update 945 2709

*Linear Case N=7, m=3, I=3, M=Im=9.

the sequential filter. The sequential filter efficiency is computed under the assumption that all measurements are statistically independent.

5. Conclusions

Several Kalman filter configurations are described and compared for a multiple sensor system. They are the parallel, sequential, and data compression filter for synchronous data collection and the sequential filter with prediction between updates for the random data collection. It is shown that the randomly collected data systems require much higher computer resources than the synchronously collected data systems. In the latter group the parallel filter requires the most computation and the data compression filter is most efficient.

Acknowledgements

The authors would like to thank Drs. Steve Weiner and John Tabaczynski for their helpful suggestions and to Kathie O'Connell for typing and preparing the manuscript.

References

1. R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problem," Trans. ASME, J. Basic Engr., Ser. D, Vol. 82 (March 1960).

2. A. H. Jazwinski, Stochastic Processes and Filtering Theory, (Academic Press New York, New York, 1970).

3. D. Willner, C. B. Chang, and K. P. Dunn, "Kalman Filter Configurations for Multiple Radar Systems," M.I.T. Lincoln Laboratory Report No. TN-1976-21 (14 April 1976).

4. Y. Bar-Shalom, "Redundancy and Data Compression in Recursive Estimation," IEEE Trans. Automatic Control AC-18, 684-689, (1973).