Distributed Estimation

Vijay Gupta

May 5, 2006

Abstract

In this lecture, we will take a look at the fundamentals of distributed estimation. We will consider a random variable being observed by N sensors. Under the assumptions of Gaussian noises and linear measurements, we will derive the weighted covariance combination of estimators. We will then touch upon the issues of distributed static sensor fusion and estimation of a dynamic random variable. Towards the end, we will look at the problem of dynamic sensor fusion, i.e., distributing a Kalman filter so that multiple sensors can estimate a dynamic random variable.

Contents

1 Preliminaries

1.1 Matrix Inversion Formula

Proposition 1. *For compatible matrices A, B, C and D,*

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B\left(C^{-1} + DA^{-1}B\right)^{-1}DA^{-1},$$

assuming the inverses exist.

Proof. Begin by considering the block matrix

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

By doing the LDU and UDL decomposition of M and equating them, we obtain

$$\begin{bmatrix} I & 0 \\ CA^{-1} & 0 \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & D - CA^{-1}B \end{bmatrix} \begin{bmatrix} I & A^{-1}B \\ 0 & I \end{bmatrix} = \begin{bmatrix} I & BD^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A - BD^{-1}C & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} I & 0 \\ D^{-1}C & I \end{bmatrix}.$$

Thus inverting both sides yields

$$\begin{bmatrix} I & -A^{-1}B \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & \left(D - CA^{-1}B\right)^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -CA^{-1} & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ -D^{-1}C & I \end{bmatrix} \begin{bmatrix} \left(A - BD^{-1}C\right)^{-1} & 0 \\ 0 & D^{-1} \end{bmatrix} \begin{bmatrix} I & -BD^{-1} \\ 0 & I \end{bmatrix}.$$

Equating the $(1,1)$ block shows

$$\left(A - BD^{-1}C\right)^{-1} = A^{-1} + A^{-1}B\left(D - CA^{-1}B\right)^{-1}CA^{-1}.$$

Finally substituting $C \to -D$ and $D \to C^{-1}$, we obtain

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B\left(C^{-1} + DA^{-1}B\right)^{-1}DA^{-1}.$$

□

1.2 Optimal mean square estimate of a random variable

We will be interested in *minimum mean square error* estimates. Given a random variable Y that depends on another random variable X, obtain the estimate $\hat{X}$ such that the mean square error given by $E\left[X - \hat{X}\right]^2$ is minimized. The expectation is taken over the random variables X and Y.

Proposition 2. *(Lemma 1 in Henrik's Kalman Filtering Lecture [1]): The minimum mean square error estimate is given by the conditional expectation* $E[X|Y = y]$.

Proof. The arguments are standard. Consider the functional form of the estimator as $g(Y)$. Let $f_{X,Y}(x, y)$ denote the joint probability density function of X and Y. Then the cost function C is given by

$$\begin{aligned} E\left[X - \hat{X}\right]^2 &= \int_x \int_y (x - g(y))^2 f_{X,Y}(x, y) dx dy \\ &= \int_y dy f_Y(y) \int_x (x - g(y))^2 f_{X|Y}(x|y) dx. \end{aligned}$$

Now consider the derivative of the cost function with respect to the function $g(y)$.

$$\begin{aligned}\frac{\partial C}{\partial g(y)} &= \int_y dy f_Y(y) \int_x 2(x-g(y)) f_{X|Y}(x|y) dx \\ &= 2\int_y dy f_Y(y)(g(y) - \int_x x f_{X|Y}(x|y) dx) \\ &= 2\int_y dy f_Y(y)(g(y) - E[X|Y=y]).\end{aligned}$$

Thus the only stationary point is $g(y) = E[X|Y=y]$. Moreover it is easy to see that it is a minimum. □

The result holds for vector random variables as well.

MMSE estimates are important because for *Gaussian* variables, they coincide with the Maximum Likelihood (ML) estimates. Of course, for non-Gaussian random variables, other notions of optimality may be better. (Recall Moving Horizon Estimation [1]).

It is also a standard result that for Gaussian variables, the MMSE estimate is linear in the state value. Proof was given in the lecture on Kalman filtering. So we will restrict our attention to linear estimates now. Also, from now on we will assume zero mean values for all the random variables. All the results can however be generalized. The covariance of X will be denoted by R_X and the cross-covariance between X and Y by R_{XY}.

Proposition 3. *The best linear MMSE estimate of X given $Y = y$ is*

$$\hat{x} = R_{XY} R_Y^{-1} y,$$

with the error covariance

$$P = R_X - R_{XY} R_Y^{-1} R_{YX}.$$

Proof. Let the estimate be $\hat{x} = Ky$. Then the error covariance is

$$\begin{aligned}C &= E[(x-Ky)(x-Ky)^*] \\ &= R_X - K R_{YX} - R_{XY} K^* + K R_Y K^*.\end{aligned}$$

Differentiating C w.r.t. K and setting it equal to zero yields

$$-2R_{XY} + 2K R_Y^{-1} = 0.$$

The result follows immediately. □

In the standard control formulations, we are also interested in measurements that are related linearly to the variable being estimated (usually the state).

Proposition 4. *Let $y = Hx + v$, where H is a matrix and v is a zero mean Gaussian noise with covariance R_V independent of X. Then the MMSE estimate of X given $Y = y$ is*

$$\hat{x} = R_X H^* (H R_X H^* + R_V)^{-1} y,$$

with the corresponding error covariance

$$P = R_X - R_X H^* (H R_X H^* + R_V)^{-1} H R_X.$$

Proof. Follows immediately by evaluating the terms R_{XY} and R_Y and substituting in the result of Proposition 3. □

2 Combining Estimators: Static Sensor Fusion

We can write the result for a linear measurement in an alternate form.

Proposition 5. *Let $y = Hx + v$, where H is a matrix and v is a zero mean Gaussian noise with covariance R_V independent of X. Then the MMSE estimate of X given $Y = y$ is*

$$P^{-1}\hat{x} = H^* R_V^{-1} y,$$

with P is the corresponding error covariance given by

$$P^{-1} = \left(R_X^{-1} + H^* R_V^{-1} H\right).$$

Proof. The expression for P follows by applying the matrix inversion lemma. For the estimate, consider

$$\begin{aligned} P^{-1}\hat{x} &= \left(R_X^{-1} + H^* R_V^{-1} H\right) R_X H^* \left(H R_X H^* + R_V\right)^{-1} y \\ &= H^* \left(H R_X H^* + R_V\right)^{-1} y + H^* R_V^{-1} H R_X H^* \left(H R_X H^* + R_V\right)^{-1} y \\ &= H^* R_V^{-1} \left(H R_X H^* + R_V\right) \left(H R_X H^* + R_V\right)^{-1} y \\ &= H^* R_V^{-1} y. \end{aligned}$$

□

This alternate form is useful because we can use it to combine local estimates directly without recourse to sending all the measurements to a central data processing unit that runs a giant estimator. This is called static sensor fusion.

2.1 Static Sensor Fusion for Star Topology

Proposition 6. *Consider a random variable X being observed by n sensors that generate measurements of the form*

$$y_i = H_i x + v_i, \qquad i = 1, \cdots, n,$$

where the noises v_i are all uncorrelated with each other and with the variable X. Denote the estimate of x based on all the n measurements by $\hat{x}$ and the estimate of x based only on the measurement y_i by $\hat{x}_i$. Then $\hat{x}$ can be calculated using

$$P^{-1}\hat{x} = \sum_{i=1}^{n} P_i^{-1}\hat{x}_i,$$

where P is the estimate error covariance corresponding to $\hat{x}$ and P_i is the error covariance corresponding to $\hat{x}_i$. Further

$$P^{-1} = \sum_{i=1}^{n} P_i^{-1} - (n-1)R_X^{-1}.$$

Proof. Denote y as the stacked vector of all the measurements y_i's, H the corresponding measurement matrix obtained by stacking all the H_i's and v the noise vector obtained by stacking all the noises v_i's. The global estimate $\hat{x}$ is given by

$$P^{-1}\hat{x} = H^* R_V^{-1} y.$$

But all the v_i's are uncorrelated with each other. Hence R_V is a block diagonal matrix with blocks R_{V_i}. Thus the right hand side can be decomposed as

$$H^* R_V^{-1} y = \sum_{i=1}^{n} H_i^* R_{V_i}^{-1} y_i.$$

But each of the terms $H_i^* R_{V_i}^{-1} y_i$ can be written in terms of the local estimates

$$P_i^{-1}\hat{x}_i = H_i^* R_{V_i}^{-1} y_i.$$

Thus

$$P^{-1}\hat{x} = \sum_{i=1}^{n} P_i^{-1}\hat{x}_i.$$

The proof for the expression for the global error covariance is similar. □

This result is useful since it allows the complexity of calculation at the fusion center to go down considerably[1]. Of course it assumes that the sensors can do some computation, but that is reasonable. The form of the global estimator shows that what we really want is a weighted mean of the local estimates. Each estimate is weighted by the inverse of the error covariance matrix. Thus more confidence we have in a particular sensor, more trust do we place in it.

2.2 Static Sensor Fusion for Arbitrary Graphs

The result above assumed the presence of a star topology in which one central node had access to local estimates from every other node. It was essentially a two step procedure then: first all the nodes transmit local estimates to the central node and then the central node calculates and transmits the weighted sum of the local estimates back. Once we realize that what is really required is a weighted average, we can generalize the approach to an arbitrary graph at the expense of more time being required. The generalization is along the lines of average consensus algorithms that have been recently considered by many people (see, e.g., [2, 3, 4]). The details of the algorithm will be covered in a later lecture. For now, I will only cover the basics.

Consider N nodes each with access to a scalar value being connected according to an arbitrary (but time-invariant) graph. Suppose we want each node to calculate the average of all the numbers. One way to do that is if each node implements the dynamical system

$$x_i(k+1) = x_i(k) + h \sum_{j:j \text{ is connected to } i} (x_j(k) - x_i(k)),$$

[1] As an exercise, compare the number of elementary operations (multiplications and additions) for the two algorithms.

where h is a small positive constant. On stacking the states of all the nodes, the entire system evolves as

$$x(k+1) = (I - hL)x(k),$$

where L is the *Graph Laplacian* matrix. If the underlying graph is connected, L has the following properties:

1. It is a symmetric positive-definite matrix. Thus the dynamics is stable (assuming h is small enough) and reaches a steady-state.

2. Each row sum is 0. Thus any vector with identical components is an equilibrium.

3. Each column sum is 0. Thus the sum of entries $x(k)$ is conserved at every time step.

Because of these three properties, it is easy to see that each entry must converge to the average of the sum of the initial conditions. This algorithm can then be readily extended for calculating *weighted* averages of *vectors* [5, 6]. If the initial values are given by the vectors $x_i(0)$, each node calculates the following:

$$x_i(k+1) = x_i(k) + hW_i^{-1} \sum_{j:j \text{ is connected to } i} (x_j(k) - x_i(k)) .$$

In our case, we let $x_i(0)$ to be the local estimate values and W_i to be inverse of the local estimation error covariance, and obtain the required weighted sum.

2.3 Sequential Measurements from One Sensor

The same algorithm can be extended to the case when there are multiple measurements from one sensor. Furthermore, the processing can be done in a sequential manner. Consider a random variable evolving in time as

$$x(k+1) = Ax(k) + w(k),$$

where $w(k)$ is white zero mean Gaussian noise with covariance matrix Q. The sensor generates a measurement at every time step according to the equation

$$y(k) = Cx(k) + v(k),$$

where $v(k)$ is again white zero mean Gaussian noise with covariance matrix R. We wish to obtain an estimate of $x(k)$ given all the measurements $\{y(0), y(1), \cdots, y(k)\}$. Suppose we divide the measurements into two sets:

1. The measurement $y(k)$.

2. The set Y of the remaining measurements $y(0)$ through $y(k-1)$.

Now note that the two sets of measurements are related linearly to $x(k)$ and further *the measurement noises are independent.* Thus we can combine the local estimates to obtain a global estimate. First we calculate the estimate of $x(k)$ based on $y(k)$. It is given by

$$M^{-1}\hat{x} = C^T R^{-1} y(k),$$

where M is the error covariance given by

$$M^{-1} = R_{x(k)}^{-1} + C^T R^{-1} C.$$

Let $\hat{x}(k-1|k-1)$ be the estimate of $x(k-1)$ based on Y and $P(k-1|k-1)$ be the corresponding error covariance. Then the estimate of $x(k)$ given Y is given by

$$\hat{x}(k|k-1) = A\hat{x}(k|k-1),$$

with the error covariance

$$P(k|k-1) = AP(k-1|k-1)A^T + Q.$$

Thus the estimate of $x(k)$ given all the measurements is given by the combination of local estimates and can be seen to be

$$P(k|k)^{-1}\hat{x}(k|k) = P(k|k-1)^{-1}\hat{x}(k|k-1) + M^{-1}\hat{x} = P(k|k-1)^{-1}\hat{x}(k|k-1) + C^T R^{-1} y(k).$$

The corresponding error covariance is given by

$$P(k|k)^{-1} = P(k|k-1)^{-1} + M^{-1} - R_{x(k)}^{-1} = P(k|k-1)^{-1} + C^T R^{-1} C.$$

These equations form the time and measurement update steps of the Kalman filter. Thus the Kalman filter can be seen to be a combination of estimators. This also forms an alternative proof of the optimality of the Kalman filter in the minimum mean squared sense under the stated assumptions.

3 Combining Sequential Measurements from Multiple Sensors: Dynamic Sensor Fusion

Suppose there are multiple sensors present that generate measurements about a random variable that is evolving in time. We can again ask the question about how to fuse data from all the sensors for an estimate of the state $x(k)$ at every time step k. This is the question of dynamic sensor fusion. We will begin by seeing why this question is difficult.

To begin with, the problem can be solved if all the sensors transmit their measurements at every time step. The central node in that case implements a Kalman filter (which we will refer to from now as the *centralized* Kalman filter). However, there are two reasons why this may not be the preferred implementation.

1. The central node needs to handle matrix operations that increase in size as the number of sensors increases. We may want the sensors to shoulder some of the computational burden.

2. The sensors may not be able to transmit at every time step. Hence we may want to transmit after some local processing, rather than transmit raw measurements.

We will initially assume that the sensors can transmit at every time step and concentrate on reducing the computational burden at the central node.

3.1 Transmitting Local Estimates

Our first guess would be to generate a local estimate at each sensor that extracts all the relevant information out of the local measurements and then to combine the estimates using methods outlined above. However, in general, it is not possible to use above method. Consider n sensors being present with the i-th sensor generating a measurement of the form

$$y_i(k) = C_i x(k) + v_i(k).$$

Suppose we denote by Y_i the set of all the measurements from the sensor i that can be used to estimate the state $x(k)$, i.e., the set $\{y_i(0), y_i(1), \cdots, y_i(k)\}$. We wish to see if the local estimates formed by the sets Y_i's can be combined to yield the optimal global estimate of $x(k)$. We can think of two ways of doing this:

1. We see that the set Y_i is linearly related to $x(k)$ through an equation of the form

$$\begin{bmatrix} y_i(k) \\ y_i(k-1) \\ \vdots \\ y_i(0) \end{bmatrix} = \begin{bmatrix} C_i \\ C_i A^{-1} \\ \vdots \end{bmatrix} x(k) + \begin{bmatrix} v(k) \\ v(k-1) - CA^{-1}w(k-1) \\ \vdots \end{bmatrix}.$$

   However we note that the process noise w appears in the noise vector. Thus even though the measurement noises $v_i(k)$'s may be independent, the noise entering the sets Y_i become correlated and hence the estimates cannot be directly combined. Of course, if the process noise is absent, the estimates can be combined in this fashion (see, e.g, [7] where the optimality in this special case was established. For a general discussion about the effects introduced by the process noise see, e.g. [8, 9, 10, 11, 12]).

2. We see that $x(k)$ can be estimated once the variables $x(0)$, $w(0)$, $\cdots$, $w(k-1)$ are estimated. Now Y_i is linearly related to these variables through

$$\begin{bmatrix} y_i(k) \\ y_i(k-1) \\ \vdots \\ y_i(0) \end{bmatrix} = \begin{bmatrix} C_i A^k & C_i A^{k-1} & \cdots & C \\ C_i A^{k-1} & \cdots & C & 0 \\ \vdots & & & \end{bmatrix} \begin{bmatrix} w(k-1) \\ w(k-2) \\ \vdots \\ x(0) \end{bmatrix} + \begin{bmatrix} v(k) \\ v(k-1) \\ \vdots \\ v(0) \end{bmatrix}.$$

   Now the measurement noises for different sensors are uncorrelated and the estimates can be combined. However, the vector being transmitted from either of the sensors is increasing in dimension as the time step k increases. Moreover the computation required is increasing since a matrix of size growing with time needs to be inverted at every time step. Hence this is not a practical solution.

Thus we see that it is not straight-forward to combine local estimates to obtain the global estimate. We can ask the question if it is possible at all to obtain the global estimate from the local estimates. Thus imagine that the local estimates $\hat{x}_i(k)$ were being combined in the optimal fashion. Is it possible to generate the global estimate $\hat{x}(k)$? As noted above, for the special case when there is no process noise, this is indeed true. However, in general, it is not possible.

Proposition 7. *(From [13]) Suppose two sets of measurements Y_1 and Y_2 are used to obtain local estimates $\hat{x}_1$ and $\hat{x}_2$. Let*

$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = L \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} \triangleq LY.$$

Then the global estimate $\hat{x}$ can be obtained from the local estimates $\hat{x}_1$ and $\hat{x}_2$ if and only if

$$R_{YY}L^T \left(LR_{YY}L^T\right)^{-1} LR_{YX} = R_{YX}.$$

Proof. The global estimate generated from the measurements is given by

$$\hat{x} = R_{XY}R_{YY}^{-1}Y.$$

If it is generated from the local estimates, it is given by

$$\hat{x} = R_{XY}L^T \left(LR_{YY}L^T\right)^{-1} LY.$$

The result is thus obvious. □

If L is invertible, the condition is satisfied and hence the global estimate can be generated from the local estimates. In general, however, L would be a fat matrix and hence the condition will not be satisfied. We thus have two options:

1. Find the best possible global estimator from the space spanned by the local estimates. This is left as an exercise.

2. Find the extra data that should be transmitted that will lead to the calculation of the global estimate. We will now describe some such schemes. For these and more such strategies see, e.g., [14, 15, 13, 16, 7, 17, 18, 19, 20, 21, 22, 23, 24, 25, 12].

3.2 Distributed Kalman Filtering

For this section we will assume that the sensors are able to transmit information to the central node at every time step. We will use the following *information form* of the Kalman filter update equations.

Proposition 8. *Consider a random variable evolving in time as*

$$x(k+1) = Ax(k) + w(k).$$

Suppose it is observed through measurements of the form

$$y(k) = Cx(k) + v(k).$$

Then the measurement updates of the Kalman filter can be given by this alternate information form.

$$\begin{aligned} P^{-1}(k|k)\hat{x}(k|k) &= P^{-1}(k|k-1)\hat{x}(k|k-1) + C^T R^{-1} y(k) \\ P^{-1}(k|k) &= P^{-1}(k|k-1) + C^T R^{-1} C. \end{aligned}$$

Proof. The equations were derived in section 2.3. □

The basic result about the requirements from the individual sensors can be derived using the above result.

Proposition 9. *The global error covariance matrix and the estimate are given in terms of the local covariances and estimates by*

$$\begin{aligned} P^{-1}(k|k) &= P^{-1}(k|k-1) + \sum_{i=1}^{N} \left(P_i^{-1}(k|k) - P_i^{-1}(k|k-1)\right) \\ P^{-1}(k|k)\hat{x}(k|k) &= P^{-1}(k|k-1)\hat{x}(k|k-1) + \sum_{i=1}^{N} \left(P_i^{-1}(k|k)\hat{x}_i(k|k) - P_i^{-1}(k|k-1)\hat{x}_i(k|k-1)\right). \end{aligned}$$

Proof. Proof follows by noting that the global estimate is given by

$$\begin{aligned} P^{-1}(k|k)\hat{x}(k|k) &= P^{-1}(k|k-1)\hat{x}(k|k-1) + C^T R^{-1} y(k) \\ P^{-1}(k|k) &= P^{-1}(k|k-1) + C^T R^{-1} C. \end{aligned}$$

Since R is block diagonal, the terms $C^T R^{-1} y(k)$ and $C^T R^{-1} C$ are decomposed into the sums

$$\begin{aligned} C^T R^{-1} y(k) &= \sum_{i=1}^{N} C_i^T R_i^{-1} y_i(k) \\ C^T R^{-1} C &= \sum_{i=1}^{N} C_i^T R_i^{-1} C_i. \end{aligned}$$

Noting the for the i-th sensor, the estimate and the error covariance are given by

$$\begin{aligned} P_i^{-1}(k|k)\hat{x}_i(k|k) &= P_i^{-1}(k|k-1)\hat{x}_i(k|k-1) + C_i^T R_i^{-1} y_i(k) \\ P_i^{-1}(k|k) &= P_i^{-1}(k|k-1) + C_i^T R_i^{-1} C_i, \end{aligned}$$

the result follows immediately. □

Based on this result we now give two architectures for dynamic sensor fusion.

1. In the first, rather obvious, architecture, the individual sensors transmit the local estimates $\hat{x}_i(k|k)$. The global fusion center combines the estimates using the theorem given above. Note that the terms $\hat{x}(k|k-1)$ and $\hat{x}_i(k|k-1)$ can be calculated by the fusion node by using the time update equation

   $$\hat{x}(k|k-1) = A\hat{x}(k-1|k-1).$$

   Similarly all the covariances can also be calculated without any data from the sensor nodes. This method is simple, especially at the sensor level. However, the fusion node has to do a lot of computation.

2. This method makes the computation at the fusion node simple at the expense of more data transmitted from the sensor node. The essential point is the observation that the term $P^{-1}(k|k-1)\hat{x}(k|k-1)$ can be written in terms of contributions from individual sensors, i.e.,

   $$P^{-1}(k|k-1)\hat{x}(k|k-1) = \sum_{i=1}^{N} z_i(k).$$

This can be proved using straight-forward algebraic manipulation as follows.

$$\begin{aligned} P^{-1}(k|k-1)\hat{x}(k|k-1) &= P^{-1}(k|k-1)A\hat{x}(k-1|k-1) \\ &= P^{-1}(k|k-1)AP(k-1|k-1)P^{-1}(k-1|k-1)\hat{x}(k-1|k-1) \\ &= P^{-1}(k|k-1)AP(k-1|k-1)\Big(P^{-1}(k-1|k-2)\hat{x}(k-1|k-2) \\ &\quad + \sum_{i=1}^{N}\big(P_i^{-1}(k-1|k-1)\hat{x}_i(k-1|k-1) \\ &\quad - P_i^{-1}(k-1|k-2)\hat{x}_i(k-1|k-2)\big)\Big). \end{aligned}$$

Thus $z_i(k)$ evolves according to the relation

$$\begin{aligned} z_i(k) = P^{-1}(k|k-1)AP(k-1|k-1)z_i(k-1) \\ + \left(P_i^{-1}(k-1|k-1)\hat{x}_i(k-1|k-1) - P_i^{-1}(k-1|k-2)\hat{x}_i(k-1|k-2)\right), \end{aligned}$$

which depends only on the i-th sensor's data. The covariances do not depend on the data and can be calculated anywhere. Hence each sensor transmits the quantity

$$\left(P_i^{-1}(k|k)\hat{x}_i(k|k) - P_i^{-1}(k|k-1)\hat{x}_i(k|k-1)\right) + z_i(k)$$

and the fusion node just calculates the sum of these quantities. Thus at expense of more data transmitted from the sensor nodes, we have made the central node very simple.

References

[1] H. Sandberg, "NCS: Kalman Filtering", available at http://www.cds.caltech.edu/∼murray/wiki/index.php?title=NCS:_Kalman_Filtering

[2] R. Olfati-Saber and R. M. Murray, "Consensus Problems in Networks of Agents with Switching Topologies and Time-delays", IEEE Transactions on Automatic Control, 49(9):1520-1533, September 2004.

[3] W. Ren, R. W. Beard, "Consensus Seeking in Multi-agent Systems Using Dynamically Changing Interaction Topologies," IEEE Transactions on Automatic Control, vol. 5, no. 5, May, 2005, p. 655-661.

[4] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules", IEEE Transactions on Automatic Control, Vol. 48, No. 6, June 2003, pp. 988-1001.

[5] D.P. Spanos, R. Olfati-Saber, and R.M. Murray, "Dynamic Consensus on Mobile Networks", IFAC 2005.

[6] L. Xiao, S. Boyd and S. Lall, "A Scheme for Robust Distributed Sensor Fusion Based on Average Consensus", IPSN'05 (International Conference on Information Processing in Sensor Networks), April 2005, p63-70.

[7] A. S. Willsky, M. G. Bello, D. A. Castanon, B. C. Levy and G. C. Verghese, "Combining and Updating of Local Estimates and Regional Maps Along Sets of One-Dimensional Tracks", IEEE Transactions on Autotmatic Control Vol. 27, No. 4, pp. 799-813. Aug 1982.

[8] Y. Bar-Shalom, "On the Track-to-Track Correlation Problem", IEEE Transactions on Automatic Control, AC-26:571-572, 1981.

[9] Y. Bar-Shalom and L. Campo, "The Effect of the Common Process Noise on the Two-Sensor Fused Track Covariance", IEEE Transactions on Aerospace and Electronic Systems, AES-22:803-805, 1986.

[10] J. A. Roecker and C. D. McGillem, "Comparison of Two-Sensor Tracking Methods Based on State Vector Fusion and Measurement Fusion", IEEE Transactions on Aerospace and Electronic Systems, AES-24:447-449, 1988.

[11] K. C. Chang, R. K. Saha and Y. Bar-Shalom, "On Optimal Track-to-Track Fusion", IEEE Transactions on Aerospace and Electronic Systems, AES-33:1271-1276, 1997.

[12] S. Mori, W. H. Barker, C. Y. Chong and K. C. Chang, "Track Association and Track Fusion with Nondeterministic Target Dynamics", IEEE Transactions on Aerospace and Electronic Systems, AES-38:659-668, 2002.

[13] C. Y. Chong, "Hierarchical Estimation," Proceedings of the 2nd MIT/ONR C^3 Workshop.

[14] D. Willner, C. B. Chang and K. P. Dunn, "Kalman Filter Algorithms for a Multisensor System", Proceedings of the 15th Conference on Decision and Control, pages 570-574, 1976.

[15] M. F. Hassan, G. Salut, M. G. Singh and A. Titli, "A Decentralized Computational Algorithm for the Global Kalman Filter", IEEE Transactions on Automatic Control, AC-23:262-268, 1978.

[16] J. L. Speyer, "Computation and Transmission Requirements for a Decentralized Linear-Quadratic Gaussian Control Problem", IEEE Transactions on Automatic Control, AC-24:266-269, 1979.

[17] B. C. Levy, D. A. Castanon, G. C. Verghese and A. S. Willsky, "A Scattering Framework for Decentralized Estimation Problem", Automatica, 19:373-384, 1983.

[18] S. Mori, C. Y. Chong and K. C. Chang, "Information Fusion in Distributed Sensor Networks", Proceedings of the American Control Conference, pages 830-835, 1985.

[19] H. R. Hashemipour, S. Roy and A. J. Laub, "Decentralized Structures for Parallel Kalman Filtering", IEEE Transactions on Automatic Control, AC-33:88-94, 1988.

[20] N. A. Carlson, "Federated Square Root Filter for Decentralized Parallel Processes", IEEE Transactions on Aerospace and Electronic Systems, AES-26:517-525, 1990.

[21] B. S. Rao and H. F. Durrant-Whyte, "Fully Decentralized Algorithm for Multi-Sensor Kalman Filtering", IEE Proceedings, 138, 1991.

[22] T. M. Berg and H. F. Durrant-Whyte, "Distributed and Decentralized Estimation", Proceedings of the Singapore International Conference on Intelligent Control and Instrumentation, volume 2, pages 1118-1123, 1992.

[23] O. E. Drummond, "Tracklets and a Hybrid Fusion with Process Noise", Proceedings of the SPIE, Signal and Data Processing of Small Targets, 3163, 1997.

[24] M. E. Liggins II, C. Y. Chong, I. Kadar, M. G. Alford, V. Vannicola and S. Thomopoulos, "Distributed Fusion Architectures and Algorithms for Target Tracking", Proceedings of the IEEE, 85:95-107, 1997.

[25] C. Y. Chong, S. Mori, W. H. Barker and K. C. Chang, "Architectures and Algorithms for Track Association and Fusion", IEEE Aerospace and Electronic Systems Magazine, 15:5-13, 2000.