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

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

Thus inverting both sides yields

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

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 $E\left[X|Y = y\right]$.*

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

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

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

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

Thus the only stationary point is $g(y) = E\left[X|Y = y\right].$ 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

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

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

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

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*

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

with the corresponding error covariance

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

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

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

[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:

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

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*

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

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

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

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

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

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.

Introduction to Distributed Control

Vijay Gupta

May 7, 2006

Abstract

In this lecture, we take a look at the problem of distributed control. We will begin by seeing why the problem is hard. Then we will look at one obvious approach towards solving the problem. Other approaches to the problem will also be mentioned.

Contents

1 Introduction

Distributed Control is a very widely used and ill-defined term. We will consider one possible way of defining such systems.

Conventional controller design problem assumes that all the controllers present in the system have access to the same information. Thus, typically, the controller design problem if to design a controller K for a plant P such that some performance specification min $\| f(P,K) \|$ is met[1]. As an example, the classical LQG problem can be stated as follows. Given a plant P of the form

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

design a controller that generates control inputs $u(k)$ as a causal function of the measurements

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

[1] There is usually also an additional specification that K should stabilize the plant.

and minimizes a quadratic cost function of the form

$$J = E \sum_{k=0}^{K} \left[x(k)^T Q x(k) + u(k)^T R u(k) \right] .$$

The noises $w(k)$ and $v(k)$ are assumed white and Gaussian.

In the corresponding distributed control problem, multiple plants of the form

$$x_i(k+1) = A_i x_i(k) + \sum_{i \neq j} A_{ij} x_j(k) + B_i u_i(k) + w_i(k)$$

are present. If the terms A_{ij} are all zero, the plants are said to be *dynamically uncoupled*. Each plant i has access to (possibly noisy) observations about the states of a set of other agents. We refer to this set as the out-neighbors of the agent i and denote it as $\mathcal{N}_i$[2]. For simplicity, throughout this lecture we will assume that each agent can access the states of all its out-neighbors perfectly. Denote by $x(k)$ the vector formed by stacking the states of all the individual agents $x_i(k)$'s and define vectors $u(k)$ and $w(k)$ similarly. The aim is to design the control laws of the individual agents to minimize (say) a quadratic cost function of the form

$$J = E \sum_{k=0}^{K} \left[x(k)^T Q x(k) + u(k)^T R u(k) \right] ,$$

where in general Q and R are full. The additional constraint is that each control input $u_i(k)$ can only depend on the states of agents in the set $\mathcal{N}_i$. If we try to minimize the cost function directly, we will come up with a control law of the form $u(k) = F(k)x(k)$ where the matrix $F(k)$ is full in general and thus does not satisfy this *topology constraint*. Solving the problem in the presence of this constraint is a much harder problem.

Thus, in general, a distributed control problem can be stated in the form [22]

$$\begin{aligned} &\text{minimize } \parallel f(P, K) \parallel \\ &\text{subject to } K \text{ stabilizes } P \\ &\qquad\qquad K \in S, \end{aligned} \tag{1}$$

where S is a subspace[3]. For a general linear time-invariant plant P and sub-space S, there is no known tractable algorithm for computing the optimal K. In the next section, we try to see why this problem is hard. We will restrict ourselves to the case of linear plants and quadratic costs from now on.

2 Information Pattern

While the problem stated in 1 is tractable (at least for the special LQ case we are concentrating on) if the subspace constraint is not present, imposing the constraint that K lie only in the subspace S renders the problem open in general. One of the earliest works that pointed out that just the

[2] By convention we assume that $i \in \mathcal{N}_i$.

[3] The way we have defined the problem makes it very similar to the problem of finding a *structured controller* for a plant.

assumptions of a linear plant, quadratic cost and Gaussian noises are not sufficient to obtain the solution was the famous counter-example provided by Witsenhausen [1] (see also [3]). The problem was originally posed in terms of two stages. We can view them as two agents in our setting. Consider $x(0)$ and v to be two independent scalar random variables. At the first stage, the random variable $x(0)$ is viewed. Thus the output equation is

$$y(0) = x(0).$$

Based on this observation, a control input $u(0)$ is calculated and applied. The state then evolves to

$$x(1) = x(0) + u(0).$$

At the next stage, the output equation is

$$y(1) = x(1) + v.$$

A control input $u(1)$ that depends on $y(1)$ is then calculated and applied to obtain

$$x(2) = x(1) - u(1).$$

The objective is to minimize the cost function given by

$$J = k^2 u(0)^2 + x(2)^2.$$

The admissible controllers are

$$\begin{aligned} u(0) &= \gamma_0 \left(y(0)\right) \\ u(1) &= \gamma_1 \left(y(1)\right), \end{aligned}$$

where γ_0 and γ_1 are Borel functions.

Note that if $u(1)$ were allowed the knowledge of $u(0)$, the problem can be solved using LQG like methods. However, in the present case, there is information to signal and the observation $y(1)$ can be used to signal that information. There is a trade-off between maximizing the information available (signaling) and minimizing the use of control at the first stage. Note the form of the cost function. At the second stage, all we are penalizing is $x(2)$ which is calculated as

$$x(2) = x(1) - u(1).$$

The controller needs to estimate $x(1)$ from $y(1)$ as best as it can, so that it can set $u(1)$ close to $x(1)$. On the other hand, at the first stage, we do not penalize the state $x(1)$ and hence the controller can choose $u(0)$ arbitrarily without worrying about $x(1)$. Thus we are asking for $x(1)$ to be

1. low entropy, so that it can be easily predicted.

2. high energy, so that the noise v does not affect it much.

Affine controllers would mean Gaussian random variables and for Gaussian variables these two aims are in direct opposition. Non-linear controllers thus can achieve better performance.

While it is known that a nonlinear controller can achieve much better performance than any linear controller, the optimal controller of this problem is still unknown. As an instance, for $k = 0.1$, the best possible affine control law gives a cost of 0.99, while non-linear controllers are possible which drive the cost as close to zero as desired. It can also be shown that the cost function is no longer convex in the controller variables, hence the problem is hard to solve numerically.

This simple counterexample is important since it shows that even for linear plants, quadratic costs and Gaussian noises, linear controls may not be optimal and the problem may be very difficult to solve. The additional piece that makes the conventional control problem simple is that of the *information pattern.* Informally, the information pattern is a representation of the information set that each decision maker in the problem (e.g. the controller) has access to at every time step when it makes the decision (e.g. calculates the control input). As an example, in the conventional LQG control problem, the controller at time step k has access to all the measurements $y(0)$, $y(1)$, $\cdots$, $y(k-1)$ as well as all the previous control inputs $u(0)$, $u(1)$, $\cdots$, $u(k-1)$. This is called a *classical information pattern*[4]. As Witsenhausen's counterexample shows, a non-classical information pattern can render a control problem intractable. Since in a distributed control problem, different controllers have access to different information sets, the information pattern is not classical and hence the problem is inherently difficult. It can be shown [4, 5], e.g., that the problem of finding a stabilizing decentralized static output feedback is NP-complete.

Since the general problem is difficult, there are two main approaches that have been proposed:

1. Identifying sub-optimal solutions.
2. Identifying special conditions or information patterns under which the problem can be solved.

We now look at these approaches in a bit more detail.

3 Sub-optimal Controller Synthesis

In this section, we will take a look at some of the approaches that have been suggested to implement sub-optimal controllers for arbitrary interconnection topology (and hence arbitrary sub-space constraints) on the controller.

Perhaps the approach that is most easy to understand is the one inspired by the design of reduced-order controllers (e.g., [6]). This approach was used to obtain numerical algorithms for solving the optimal linear control with arbitrary number of free parameters for the infinite horizon case in, e.g., [7, 8]. We will consider the version presented in [9].

Consider N dynamically uncoupled agents evolving as

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k),$$

where the control of the i-th agent can depend linearly on its own state value and the states of a specified set of other agents. On stacking the states of all the agents, the system evolves as

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ u(k) &= Fx(k), \end{aligned}$$

[4] Alternatively, the information pattern has *total recall.*

where F is a matrix that incorporates the interconnection information. In particular, F has a block structure, with the (i, j)-th block zero if agent i cannot obtain the information about agent j to calculate its state value. Thus F is constrained to lie in a particular space. Assume that the initial condition $x(0)$ is random and Gaussian with mean zero and covariance $R(0)$. We wish to find the constrained control law F that minimizes the cost function

$$J = E\left[\sum_{k=0}^{\infty}\{x^T(k)Qx(k) + u^T(k)Ru(k)\}\right].$$

Assume that a F exists in the required space, such that $A + BF$ is stable. Then, for that F, the cost function is given by

$$J = E\left[x^T(0)Px(0)\right],$$

where P satisfies the discrete algebraic Lyapunov equation

$$P = (Q + F^T RF) + (A + BF)^T P(A + BF).$$

Thus the cost is given by $J = \text{trace}(PR(0))$ with $R(0)$ as the initial covariance.

The case when noise is present can also be expressed similarly. Suppose that the system evolves as

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k) \\ u(k) &= Fx(k), \end{aligned}$$

where F is chosen to minimize the cost function

$$J = \lim_{k\to\infty} E\left[x^T(k)Qx(k) + u^T(k)Ru(k)\right].$$

As an exercise, prove that the cost can now be written as $J = \text{trace}(PR_w)$ where R_w is the covariance of noise $w(k)$. Note that because F is stable, the initial condition $R(0)$ would not affect the cost function.

3.1 Stabilizability

Two questions arise immediately:

1. Is it possible to stabilize the system using information from other agents when the agents are individually not stable. In other words, if an agent is unstable, can the system be stabilized by the exchange of information between different agents?

2. Are some topologies inherently unstable in that even if the agents are stable, the information flow will always make it impossible to stabilize the formation?

The following result [9, 10]) answers these questions.

Proposition 1. *Consider a system of interconnected dynamically uncoupled agents as defined above.*

1. *The system is controllable if and only if each individual agent is controllable.*

2. The system is stabilizable if and only if each individual agent is stabilizable.

Proof. We present the proof for the case of identical agents. The case of non-identical agents is similar and is left as an exercise. Suppose there are N agents each with state-space dimension m with state matrices Φ and Γ. Thus the entire system has state-space dimension Nm and system matrices

$$\begin{aligned} A &= I \otimes \Phi \\ B &= I \otimes \Gamma, \end{aligned}$$

where I is the identity matrix of suitable dimensions and $\otimes$ represents the Kronecker product. For controllability of the system, we thus want the following matrix to have rank Nm

$$M_1 = \left[\begin{array}{ccccc} I \otimes \Gamma & (I \otimes \Phi)(I \otimes \Gamma) & (I \otimes \Phi)^2(I \otimes \Gamma) & \cdots & (I \otimes \Phi)^{Nm-1}(I \otimes \Gamma) \end{array} \right].$$

Using the standard property of Kronecker product

$$(a \otimes b)(c \otimes d) = ac \otimes bd,$$

we can rewrite M_1 as

$$M_1 = \left[\begin{array}{ccccc} I \otimes \Gamma & (I \otimes \Phi\Gamma) & (I \otimes \Phi^2\Gamma) & \cdots & (I \otimes \Phi^{Nm-1}\Gamma) \end{array} \right].$$

This matrix has rank Nm if and only if the following matrix has rank m

$$M_2 = \left[\begin{array}{ccccc} \Gamma & \Phi\Gamma & \Phi^2\Gamma & \cdots & \Phi^{Nm-1}\Gamma \end{array} \right].$$

Since Φ is an $m \times m$ matrix, the equivalent condition is that the matrix

$$M_3 = \left[\begin{array}{ccccc} \Gamma & \Phi\Gamma & \Phi^2\Gamma & \cdots & \Phi^{m-1}\Gamma \end{array} \right]$$

has rank m. But M_3 being rank m is simply the condition for the individual agent being controllable. Thus the system is controllable if and only if each individual agent is controllable. This proves the first part. The proof of the second part is similar. The subspace not spanned by the columns of M_1 is stable if and only if the subspace not spanned by the columns of M_3 is stable. □

3.2 Numerical Algorithms

In this section we obtain necessary conditions for the optimal solution that we can numerically solve. We wish to find

$$F = \sum_{i=1}^{n} \alpha_i \Phi_i$$

such that trace$(PR(0))$ is minimized, where

$$P = (Q + F^T RF) + (A + BF)^T P(A + BF). \tag{2}$$

For a critical point

$$\text{trace}\left(\frac{\partial P}{\partial \alpha_i} R(0)\right) = 0, \qquad \forall i = 1, 2, \cdots, n.$$

Let us define

$$\Sigma_i = \Phi_i^T \left[RF + B^T P(A+BF) \right]. \tag{3}$$

Differentiating (2) with respect to α_i, we obtain

$$\frac{\partial P}{\partial \alpha_i} = (A+BF)^T \frac{\partial P}{\partial \alpha_i}(A+BF) + \Sigma_i + \Sigma_i^T.$$

Thus the cost is given by

$$\text{trace}\left(\frac{\partial P}{\partial \alpha_i} R(0)\right) = \text{trace}\left(\left((A+BF)^T \frac{\partial P}{\partial \alpha_i}(A+BF) + \Sigma_i + \Sigma_i^T\right) R(0)\right).$$

Now the covariance of the state at time k evolves as

$$R(k+1) = (A+BF)R(k)(A+BF)^T.$$

Thus

$$\text{trace}\left((A+BF)^T \frac{\partial P}{\partial \alpha_i}(A+BF)R(0)\right) = \text{trace}\left(\frac{\partial P}{\partial \alpha_i} R(1)\right).$$

Using this relation k times, we obtain

$$\text{trace}\left(\frac{\partial P}{\partial \alpha_i} R(0)\right) = \text{trace}\left(\left(\frac{\partial P}{\partial \alpha_i} R(k) + \Sigma_i X(k) + \Sigma_i^T X(k)\right) R(0)\right),$$

where

$$X(k) = R(0) + R(1) + \cdots + R(k).$$

But if $(A+BF)$ is stable, $R(k)$ would be approximately be a zero matrix for sufficiently large values of k. Thus if we denote

$$X = R(0) + R(1) + \cdots,$$

we see that X satisfies the Lyapunov equation

$$X = R(0) + (A+BF)X(A+BF)^T, \tag{4}$$

we obtain the following necessary condition for a critical point. We want

$$\text{trace}\left(\Sigma_i X + \Sigma_i^T X\right) = 0 \qquad \forall i = 1, \cdots, n,$$

where

$$F = \sum_{i=1}^{n} \alpha_i \Phi_i,$$

P satisfies (2), Σ_i is defined by (3) and X satisfies (4). This equation can either be solved iteratively or a gradient descent method can be used to obtain the control law.

As an exercise, show that

1. For the case when F has no restrictions on its structure, we obtain the usual condition

$$B^T P(A+BF) + RF = 0.$$

2. If the initial conditions of the agents are independent, then for the completely decentralized case (when the control law of each agent can depend only on its own state value), only the diagonal terms of cost matrices Q and R are important. Note that this is *not* true is general. Even if agent i cannot access the state of agent j, the (i, j)-th block of matrices Q and R are still important.

The algorithm we have discussed is for the infinite horizon case. For the finite horizon case, a similar algorithm can be applied as described, e.g., in [11]. However, as [12] pointed out, there are computational difficulties arising out of solving a large number of coupled matrix equations. A sub-optimal algorithm to get around this difficulty was proposed in [10] in which as opposed to NT coupled matrix equations (where T is the time horizon and N agents are present), N equations need to be solved T times.

3.3 Other Approaches

We have described one particular approach towards obtaining sub-optimal algorithms for the distributed control problem. Many other approaches have been proposed in the literature. We do not have time to go through them in any detail. However we summarize a couple of approaches here.

The problem of synthesizing a constrained controller while minimizing a H_2 performance criterion was considered in [13]. The analysis problem (for a given controller) was shown to be convex. However the it was shown that for the synthesis problem, enforcing the topology constraint typically destroys convexity. A method to retain convexity at the expense of sub-optimality was presented.

The problem of synthesizing a distributed controller achieving H_∞ performance was considered in [14]. They used tools inspired by dissipativity theory and derive sufficient LMI conditions on which performance constraints can be imposed. The controller structure that they come up with has the same interconnection topology as the plant interconnection topology. The tools have been extended to the case of lossy communication links in [15].

These are but two particular approaches. The problem can also be looked at in context of Receding Horizon Control. Distributed Receding Horizon Control will be covered in detail next week. There is also extensive work on many other approaches including those inspired by Game Theory, potential fields and so on.

4 Identifying Solvable Information Patterns

As we saw, the general problem of distributed control is very difficult and the optimal controller is not known for arbitrary information patterns. In particular, optimal controllers are not linear or even numerically easy to calculate in general. There have been numerous efforts to classify what information patterns lead to linear controllers being optimal and in what cases can the optimal linear controller be cast as a convex optimization problem. Witsenhausen [2] in a survey paper summarized several important results. He gave sufficient conditions under which the standard LQG theory could be applied and thus the optimal controller would be linear. Another important early contribution was [16] which showed the optimal controller to be linear for a class of information structures that they called *partially nested*. A partially nested information structure is one in which the *memory communication structure* is the same as the precedence relation in the *information structure diagram*. Informally, this means that a controller A has access to all the information that another controller B has access to, if the decision that B makes can affect the information set of

A. Thus once the control laws are fixed, any controller can deduce the action of all the controllers *precedent* to it.The only random effects are due to the structure of the external disturbances which are not control-law dependent.

As an example, consider a system where two agents evolve according to

$$\begin{aligned} x_1(k+1) &= A_1x_1(k) + B_1u_1(k) + w_1(k) \\ x_2(k+1) &= A_2x_2(k) + A + 12x_1(k) + B_2u_2(k) + w_2(k), \end{aligned}$$

where $w_1(k)$ and $w_2(k)$ are white uncorrelated zero mean Gaussian noises. Further let the initial conditions $x_1(0)$ and $x_2(0)$ be independent. Suppose the cost function to be minimized is

$$J = E\left[\sum_{k=0}^{K}\{x_1^T(k+1)Q_1x_1(k+1) + x_2^T(k+1)Q_2x_2(k+1) + u_1^T(k)R_1u_1(k) + +u_2^T(k)R_2u_2(k)\}\right].$$

The agents are being observed through measurements of the form

$$\begin{aligned} y_1(k) &= C_1x_1(k) + v_1(k) \\ y_2(k) &= C_2x_2(k) + v_2(k), \end{aligned}$$

with the usual assumptions on the noises. Obviously if both the agents have access to all previous control inputs $u_i(0)$, $u_i(1)$, $\cdots$, $u_i(k-1)$ and the measurements $y_i(0)$, $y_i(1)$, $\cdots$, $y_i(k)$ at any time step k, the information structure is classical. The problem then admits of unique optimal control inputs $u_i(k)$. Further they are linear in the measurements and can be obtained, e.g., using the LQG theory. However now consider an information pattern in which agent 1 has access to its own previous controls $u_1(0)$, $u_1(1)$, $\cdots$, $u_1(k-1)$ and its own measurements $y_1(0)$, $y_1(1)$, $\cdots$, $y_1(k)$. The agent 2 has access to its own control inputs but measurements from both agents. In this case, the information pattern is partially nested. Even for this information structure, the optimal control inputs are unique and linear in the measurements. This is so because agent 1 can choose its control input without worrying about agent 2's decision. Agent 2 can reconstruct agent 1's control input if it knows the control *law* followed by agent 1 even if it does not have access to the control *input* directly. Thus it can also solve for the same control input as in the classical information pattern case.

There has been a lot of work on particular information patterns. For instance the one-step delayed information sharing pattern assumes that each controller has, at the current time, all the previously implemented control values, all the observations made anywhere in the system through, and including the previous time, and its own observation at the current time. Hence current observations are not shared. Recursive solutions for this problem with a quadratic cost were provided using dynamic programming in [18], an exponential cost by [19] and with H_2, H_∞ and L_1 costs by [21]. Some other structures that are tractable have been identified, e.g., in [17, 20]. A property called *quadratic invariance* was defined in [22] and it was shown that it is necessary and sufficient for the constraint set to be preserved under feedback, and that this allows optimal stabilizing decentralized controllers to be synthesized via convex programming.

References

[1] H. S. Witsenhausen, “A Counterexample in Stochastic Optimum Control,” SIAM Journal of Control, vol. 6, no. 1, pp. 131-147, 1968.

[2] H. S. Witsenhausen, "Separation of Estimation and Control for Discrete Time Systems," Proceedings of the IEEE, vol. 59, no. 11, pp. 1557-1566, Nov. 1971.

[3] A. Sahai and S. Mitter, "Information and Control: Witsenhausen Revisited," Lecture Notes in Control and Information Sciences, vol. 241, pp. 639-654, 1999.

[4] C. H. Papadimitriou and J. N. Tsitsiklis, "Intractable Problems in Control Theory," SIAM Journal of Control and Optimization, vol. 24, pp. 639-654, 1986.

[5] V. Blondel and J. Tsitsiklis, "A Survey of Computational Complexity Results in Systems and Control", Automatica, vol. 36, no. 9, pp. 1249-1274, 2000.

[6] W. S. Levine, T. L. Johnson and M. Athans, "Optimal Limited State Variable Feedback Controllers for Linear Systems", IEEE Transactions on Automatic Control, AC-16, pp. 785-793, 1971.

[7] C. J. Wenk and C. H. Knapp, "Parameter Optimization in Linear Systems with Arbitrarily Constrained Controller Structure", IEEE Transactions on Automatic Control, AC-25(3), pp. 496-500, 1980.

[8] T. Soderstrom, "On Some Algorithms for Design of Optimal Constrained Regulators", IEEE Transactions on Automatic Control, AC-23, pp. 1100-1101, 1978.

[9] V. Gupta, B. Hassibi and R. M. Murray, "On the Synthesis of Control Laws for a Network of Autonomous Agents", Proceedings of the American Control Conference, 2004, Vol. 6, pp. 4927-4932, 2004.

[10] V. Gupta, B. Hassibi and R. M. Murray, "A Sub-optimal Algorithm to Synthesize Control Laws for a Network of Dynamic Agents", International Journal of Control, vol. 78, no. 16, pp. 1302-1313, November 10, 2005.

[11] B. D. O. Anderson and J. B. Moore, "Linear Optimal Control", Prentice Hall, 1971.

[12] D. L. Kleinman, T. Fortmann and M. Athans, "On the Design of Linear Systems with Piecewise-Constant Feedback Gains", IEEE Transactions on Automatic Control, AC-13(4), pp. 354-361, 1968.

[13] G. A. de Castro, "Convex Methods for the Design of Structured Controllers," Ph D Thesis, University of California, Los Angeles, 2001.

[14] C. Langbort, R. S. Chandra and R. D'Andrea, "Distributed Control Design for Systems Interconnected over an Arbitrary Graph," IEEE Transactions on Automatic Control, vol. 49, no. 9, pp. 1502-1519, September 2004.

[15] C. Langbort, V. Gupta and R. M. Murray, "Distributed Control over Failing Channels", Workshop on Networked Embedded Sensing and Control, October 2005. To appear as a volume in the Lecture Notes in Control and Information Sciences series published by Springer.

[16] Y. C. Ho and K. C. Chu, "Team Decision Theory and Information Structures in Optimal Control Problems - Part I," IEEE Transactions on Automatic Control, vol. AC-17, no. 1, pp. 15-22, Jan. 1972.

[17] B. Bamieh and P. G. Voulgaris, “Optimal Distributed Control with Distributed Delayed Measurement”, In Proceedings of the IFAC World Congress, 2002.

[18] N. Sandell and M. Athans, “Solution of some Nonclassical LQG Stochastic Decision Problems,” IEEE Transactions on Automatic Control, vol. AC-19, pp. 108-116, April 1974.

[19] C. Fan, J. L. Speyer and C. R. Jaensch, “Centralized and Decentralized Solutions of the Linear-Exponential-Gaussian Problem”, IEEE Transactions on Automatic Control, vol. 39, no. 10, pp. 1986-2003, October 1994.

[20] X. Qi, M. Salapaka, P. Voulgaris and M. Khammash, “Structured Optimal and Robust Control with Multiple Criteria: A Convex Solution,” IEEE Transactions on Automatic Control, vol. 49, no. 10, pp. 1623-1640, October 2004.

[21] P.G. Voulgaris, “Optimal control of systems with delayed observation sharing patterns via input-output methods,” Systems and Control Letters, Vol. 50, pp. 51-64, 2003.

[22] M. Rotkowitz and S. Lall, “A Characterization of Convex Problems in Decentralized Control”, IEEE Transactions on Automatic Control, vol. 51, no. 2, pp. 274-286, February 2006.