Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

On Sensor Fusion in the Presence of Packet-dropping Communication Channels

Vijay Gupta, Babak Hassibi, Richard M Murray

Abstract— **In this paper we look at the problem of multi-sensor data fusion when data is being communicated over channels that drop packets randomly. We are motivated by the use of wireless links for communication among nodes in upcoming sensor networks. We wish to identify the information that should be communicated by each node to others given that some of the information it had transmitted earlier might have been lost. We solve the problem exactly for the case of two sensors and study the performance of the algorithm when more sensors are present. For the two-sensor case, the performance of our algorithm is optimal in the sense that if a packet is received from the other sensor, it is equivalent to receiving all previous measurements, irrespective of the packet drop pattern.**

I. INTRODUCTION

In recent years, there has been a lot of interest in sensor networks and sensor webs. Originally motivated by military surveillance applications, they are now being employed in a wide variety of applications (e.g., see [11], [23] and the references therein). Compared to using one big sensor, typical advantages of using sensor networks include relatively lower costs, inherent robustness and greater mobility. Moreover greater accuracy is possible not only due to more observations being taken for the same source but also since multiple types of sensors can potentially be used that generate observations related to different facets of the source. However, the price to be paid for these advantages is the need for communication and greater complexity in the estimation algorithms.

A basic problem in sensor networks is how to fuse the observation data from many nodes. The classical Kalman filter is a centralized filter that assumes all observations coming to a central computing facility. Over the years techniques have been proposed to decentralize the filter computations and to minimize the amount of information that needs to be transmitted among the nodes. An early contribution was [24] where information obtained from the local sensors is combined to generate the global estimate. However it required that data about the global estimate be sent from the fusion node to the local sensors. A similar requirement was imposed in the 'successive orthogonalization of measurement subspaces' algorithm proposed in [13]. This difficulty was first overcome in [22], [7] in which each local node sends its own local estimate based on its own data and communicates this estimate and the error covariance data to the fusion center. Similar results for continuous time systems were presented in [25]. These results were further extended in [12] where both the measurement and time update steps of the Kalman filter were decentralized. An alternative approach for data fusion from many nodes using the Federated filter was proposed in [5]. A Bayesian method was used and some algorithms presented in [14], [4] which are optimal when there is no process noise. A scattering framework [17] and algorithms based on decomposition of the information form of the Kalman filter [19], [3] have also been proposed for data fusion. A scheme based on the concept of dynamic consensus is proposed in [21], but it assumes multiple communication rounds per time step of the system evolution. For some other approaches proposed in the literature (e.g. those based on tracklets [9]), see [8], [18].

However these approaches assume a fixed communication topology among the nodes with a link, if present, being perfect. With wireless channels being used for communication between sensor nodes that are mobile, this assumption needs to be questioned. In such a case, packets of information from one node to another will be dropped randomly by the communication channel present between them. This random loss of information reintroduces the problem of correlation between the estimation errors of various nodes [1] and renders the approaches proposed in the literature as sub-optimal. We wish to address this problem of finding the optimal global estimate for each node in the case when there are communication channels present between the nodes and packets of information are being randomly dropped. We disallow approaches such as sending all the measurements taken by each node across the entire network each time communication is possible because they can potentially entail transmitting arbitrarily large amounts of data. An approach to solve this problem was proposed in [2] in the context of track-to-track fusion through exchange of state estimates based on each sensor's own local measurements but the specific scheme that was used was not proven to be optimal. Moreover, as was found in [20], the algorithm for fusing the local state estimates that was proposed is not optimal in the mean square sense. It was subsequently proved in [6], [18] that the technique was based on an assumption that was not met in general. The main contribution of this paper is to pose the problem of optimal coding for estimation in the presence of packet dropping channels and to propose an alternative algorithm which is optimal in the mean square sense for the case of two sensors being present. For more than two sensors, the algorithm no

Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125, USA {gupta,hassibi,murray}@caltech.edu Work supported in part by the AFOSR grant F49620-01-1-0460.

0-7803-9568-9/05/$20.00 ©2005 IEEE

longer remains optimal, but we present simulations to study the performance.

This paper is organized as follows. We begin with the problem formulation where we also define our notations and assumptions. Then we solve the problem for the case of two sensors. In the next section we present simulation results to study the performance of the protocol as the number of sensors is increased. We finish with conclusions and outline some avenues for future work.

II. PROBLEM FORMULATION

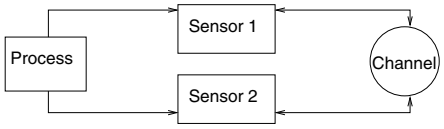

Fig. 1. Problem block diagram. Only two sensors are shown, but in general there can be N sensors.

The basic set-up of the problem is illustrated in Figure 1. Consider the discrete-time process that evolves according to

$$x_{k+1} = F_k x_k + G_k u_k, \tag{1}$$

where $x_k \in \mathbf{R}^n$ is the process state and $u_k \in \mathbf{R}^m$ is zero-mean white noise process. The initial condition x_0 is assumed to be a zero mean random variable with covariance matrix Π_0. The process is being observed by N sensors ($N = 2$ in the figure) of the form

$$y_k^i = H_k^i x_k + v_k^i, \qquad i = 1, \cdots, N, \tag{2}$$

with $v_k^i \in \mathbf{R}^{p_i}$ being zero-mean white noise processes as well. Further

$$\left\langle \begin{bmatrix} u_k \\ v_k^1 \\ \vdots \\ v_k^N \end{bmatrix} \begin{bmatrix} u_j \\ v_j^1 \\ \vdots \\ v_j^N \end{bmatrix} \right\rangle = \begin{bmatrix} Q_k & 0 & \cdots & 0 \\ 0 & R_k^1 & \cdots & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & R_k^N \end{bmatrix} \delta_{kj},$$

where the inner product $\langle x, y \rangle$ is defined in the usual way (see, e.g., [15])

$$\langle x, y \rangle = E\left[xy^T\right].$$

The sensor models and the noise statistics are known to all sensors. From now on, we use the words sensor, sensor-estimator and node interchangeably. The sensors are allowed to communicate through communication channels. The communication channels are modeled as switches with an associated drop probability. In other words, for every transmission, the channel either transmits the input to the output with a certain probability or does not produce any output. Such a channel model is a natural way to model situations where error detection coding is done for each packet of information. We assume that there is no inter-transmission coding present. Further we assume that a sufficient number of bits are allotted for each transmission so that quantization effects can be ignored.

At every time instant, each sensor-estimator thus has a pool of knowledge available to it about its own and other sensors' measurements. On the basis of this knowledge, it constructs an estimate of the process (1). The estimate of sensor i at time step k is denoted by $\hat{x}_k^i$. We assume that to generate the estimate of x_k the measurements till time step k can be used. Thus we are interested in causal estimators only. We further restrict our attention to linear estimators. The goal of each sensor-estimator i is to minimize the covariance matrix of the error defined by

$$e_k^i = x_k - \hat{x}_k^i.$$

In other words, we are interested in the linear least mean square (llms) estimator of x_k. If there are no communication channel related constraints present, each sensor can transmit the latest measurement it has observed to other sensors and at time step k, every sensor will thus have access to all the measurements $\left[\{y_j^1\}_{j=0}^k, \{y_j^2\}_{j=0}^k, \cdots, \{y_j^N\}_{j=0}^k\right]$ taken so far by the N sensors. It can then calculate the best estimate through a Kalman filter (see, e.g., [16]). Further, to reduce computation at every node, the Kalman filter can be decentralized through any of the techniques already mentioned. The key thing to note in these techniques is that all nodes are alike in the sense that they have access to an identical set of information from which to construct estimates. Thus every node can function as a central node which processes the information sent by other nodes. Further since the latest information is transmitted at every time instant, the estimation process can be recursive.

The introduction of communication channels, breaks this symmetry and introduces several interesting issues into the problem.

1) It is no longer clear what information the nodes should transmit to each other. Transmission of measurements alone might not be optimal. Consider two nodes i and j joined by a communication channel that has just dropped a packet at time step k. If the node i is sending its latest measurement alone, at time step $k+1$ the node j will have no access to the measurement y_k^i. Thus its estimate will not be optimal. On the other hand, the node i does not know that a packet has been dropped at time step k. Thus it cannot retransmit the information. If it transmits *all* the measurements it has taken so far, the amount of information needed to be transmitted will grow without bound as time k increases.
2) The conventional sensor fusion algorithms are no longer applicable. This is so since the fusion algorithms are usually recursive in either the measurements or the local estimates. It is not clear what to do in the case when the information is not delivered at a particular time step, but can *potentially* be delivered at some future time when the communication channel allows information to be transmitted.
3) It may be possible to use some routing algorithms to improve performance. Consider three nodes i, j and

k. If the communication channel between j and k drops a lot of packets, it might be useful for j to rout information to k through the node i.

We propose to look at these issues. We begin with the case of two sensors where we have tight results and then study the performance for the case of N sensors being present. We assume a broadcast medium in which each sensor can talk to every other sensor, except for packet drops. The packet dropping pattern is assumed to be independent from one time step to the next, although as we shall see, for the case of 2 sensors, our algorithm is optimal *irrespective* of the packet drop statistics or model.

III. TWO SENSORS PRESENT

Let the measurements of the sensors at time step k be given by y_k^1 and y_k^2. Let the estimates be denoted as $\hat{x}_k^1$ and $\hat{x}_k^2$ respectively. Moreover, let $\hat{a}_{k|S}$ denote the best possible estimate of random variable a given the information in the set S. Note that if there are two information sets S_1 and S_2 such that $S_1 \subset S_2$, then the error in the estimate $\hat{a}_{k|S_2}$ is less than or equal to the error in the estimate $\hat{a}_{k|S_1}$ in the mean square sense. This is so since while forming the estimate $\hat{a}_k$, we can simply not use the information from S_2.

Keeping this fact in mind, we first write down the biggest set of information which can possibly be available to a sensor on which it bases its estimate. To understand the situation a little better, we view it from the position of sensor 1. For sensor 1, we are looking to optimize the estimate $\hat{x}^1_{k+1|S}$, where S can be no bigger than the following

- If a packet has been received from sensor 2 at time step k, then $S = \left[\{y_j^1\}_{j=0}^k, \{y_j^2\}_{j=0}^k\right]$.
- If the last packet was received from sensor 2 at time step $k-t$, then $S = \left[\{y_j^1\}_{j=0}^k, \{y_j^2\}_{j=0}^{k-t}\right]$.

Similar statements can be written about sensor 2. The error covariances of these estimates will be a lower bound on all schemes that transmit different information from one sensor to the other. As an example, for a scheme that only sends the latest measurements, the information set would not contain those measurements that are dropped by the channel and would thus be a subset of S mentioned above. Thus its error covariance will be more than the one implied by this upper bound.

In the following, we propose an algorithm that achieves the upper bound. We will adopt the view-point of sensor 1 in the presentation. Consider a hypothetical centralized estimator, that has access to all the measurements from both the sensors. For this optimal centralized filter define the following quantities:

- $\hat{x}_{k|l}$: estimate of x_k based on all the measurements of both the sensors up to time step l.
- $P_{k|l}$: covariance matrix of the error corresponding to the estimate $\hat{x}_{k|l}$.

Equivalently, we can say that the optimal centralized filter is obtaining measurements from a filter of the form

$$y_k = H_k x_k + v_k,$$

where

$$H_k = \begin{bmatrix} H_k^1 \\ H_k^2 \end{bmatrix}$$

while v_k is white Gaussian noise with zero mean and covariance

$$R_k = \begin{bmatrix} R_k^1 & 0 \\ 0 & R_k^2 \end{bmatrix}.$$

Thus we can write the following equations for the time and measurement updates of the Kalman filter:

$$\begin{aligned}
\left(P_{k|k}\right)^{-1} &= \left(P_{k|k-1}\right)^{-1} + H_k^T \left(R_k\right)^{-1} H_k \\
\left(P_{k|k}\right)^{-1} \hat{x}_{k|k} &= \left(P_{k|k-1}\right)^{-1} \hat{x}_{k|k-1} + H_k^T \left(R_k\right)^{-1} y_k \\
P_{k|k-1} &= F_{k-1} P_{k-1|k-1} F_{k-1}^T + G_{k-1}^T Q_{k-1} G_{k-1} \\
\hat{x}_{k|k-1} &= F_{k-1} \hat{x}_{k-1|k-1}.
\end{aligned}$$

Now we utilize the fact that R_k is a block diagonal matrix. Thus

$$\begin{aligned}
H_k^T \left(R_k\right)^{-1} H_k &= \sum_i \left[\left(H_k^i\right)^T \left(R_k^i\right)^{-1} H_k^i\right] \\
H_k^T \left(R_k\right)^{-1} y_k &= \sum_i \left[\left(H_k^i\right)^T \left(R_k^i\right)^{-1} y_k^i\right].
\end{aligned}$$

Thus we can rewrite the equations for the optimal filter as

$$\begin{aligned}
\left(P_{k|k}\right)^{-1} &= \left(P_{k|k-1}\right)^{-1} \\
&\quad + \sum_i \left[\left(P_{k|k}^i\right)^{-1} - \left(P_{k|k-1}^i\right)^{-1}\right] \\
\left(P_{k|k}\right)^{-1} \hat{x}_{k|k} &= \left(P_{k|k-1}\right)^{-1} \hat{x}_{k|k-1} \\
&\quad + \sum_i \left[\left(P_{k|k}^i\right)^{-1} \hat{x}_{k|k}^i - \left(P_{k|k-1}^i\right)^{-1} \hat{x}_{k|k-1}^i\right] \\
P_{k|k-1} &= F_{k-1} P_{k-1|k-1} F_{k-1}^T + G_{k-1}^T Q_{k-1} G_{k-1} \\
\hat{x}_{k|k-1} &= F_{k-1} \hat{x}_{k-1|k-1}
\end{aligned}$$

The first two equations form a *measurement update* step and the last two a *time update* step. Note that the communication requirement at time step k from sensor i is as follows:

- For the covariance matrices, no communication is required. They can be calculated by either sensor, even off-line.
- For the estimates, the contribution from i-th sensor is $\left(P_{k|k}^i\right)^{-1} \hat{x}_{k|k}^i - \left(P_{k|k-1}^i\right)^{-1} \hat{x}_{k|k-1}^i$.

We can use this observation to obtain the information that sensor i should transmit to the other sensor that is used to calculate the optimal estimate. Denote

$$\Lambda_k^i = \left(P_{k|k}^i\right)^{-1} \hat{x}_{k|k}^i - \left(P_{k|k-1}^i\right)^{-1} \hat{x}_{k|k-1}^i.$$

Thus Λ_k^i is the contribution from sensor i corresponding to measurement at time step k. Thus, we can write

$$\begin{aligned}\left(P_{k|k}\right)^{-1}\hat{x}_{k|k} &= \left(P_{k|k-1}\right)^{-1}\hat{x}_{k|k-1} + \sum_i \Lambda_k^i \\ &= \left(P_{k|k-1}\right)^{-1} A\hat{x}_{k-1|k-1} + \sum_i \Lambda_k^i \\ &= \Gamma_k\left[\left(P_{k-1|k-2}\right)^{-1}\hat{x}_{k-1|k-2} + \sum_i \Lambda_{k-1}^i\right] \\ &\quad + \sum_i \Lambda_k^i\end{aligned}$$

where

$$\Gamma_k = \left(P_{k|k-1}\right)^{-1} AP_{k-1|k-1}.$$

Continuing this way, we obtain

$$\left(P_{k|k}\right)^{-1}\hat{x}_{k|k} = \Gamma_k\Gamma_{k-1}\cdots\Gamma_1 P_{0|-1}^{-1}\hat{x}_{0|-1} + \sum_i I_k^i,$$

where

$$\begin{aligned}I_k^i = \Lambda_k^i + \Gamma_k\Lambda_{k-1}^i + \Gamma_k\Gamma_{k-1}\Lambda_{k-2}^i + \cdots \\ + \left(\Gamma_k\Gamma_{k-1}\cdots\Gamma_1\right)\Lambda_0^i\end{aligned}$$

Thus, the information needed from sensor i at time step k is precisely I_k^i given in the above equation.Note that the computation required for calculating I_k^i does not grow with time since

$$I_k^i = \Lambda_k^i + \Gamma_k I_{k-1}^i.$$

This information vector 'washes away' the effect of any previous packet losses and leads to the calculation of optimal estimate at time step k as if all the measurements from both sensors were available, ie, it calculates the estimate $\hat{x}_{k|S}^i$ if the last communication from sensor j was possible at time step k. No knowledge about packet drop probabilities is required.

Thus we have obtained a way for sensor i to obtain the lowest possible error achievable when communication from sensor j is possible at time step k. If the last communication was possible at time step $k-t$, clearly, the best estimate will be obtained by finding the estimate at time step $k-t$ and then propagating it with sensor i's measurements only. Thus we obtain the following algorithm that achieves the estimate based on the set S outlined above. We summarize the algorithm in the following proposition.

Proposition 1: The algorithm outlined below achieves the optimal causal llms estimate based on all of the sensor's own measurements till time step k and the other sensor's measurements till time step $k-t$ where the last communication from the other sensor was possible at time $k-t$.

Algorithm : At each time step k,

- Each sensor takes its own measurement and runs a local Kalman filter. At time step k, it calculates
  - $\hat{x}_{k|k}^i$ and $P_{k|k}^i$ from its local Kalman filter.
  - $\Lambda_k^i = \left(P_{k|k}^i\right)^{-1}\hat{x}_{k|k}^i - \left(P_{k|k-1}^i\right)^{-1}\hat{x}_{k|k-1}^i$.
  - Global error covariance matrices $P_{k|k}$ and $P_{k|k-1}$.
  - $\Gamma_k = \left(P_{k|k-1}\right)^{-1} F_{k-1}P_{k-1|k-1}$.
  - $I_k^i = \Lambda_k^i + \Gamma_k I_{k-1}^i$.
- It transmits I_k^i and waits for the corresponding information from the other sensor.
  - If it receives I_k^j from the other sensor, it obtains the global estimate as

$$\begin{aligned}&\left(P_{k|k}\right)^{-1}\hat{x}_{k|k} \\ &\quad = \Gamma_k\Gamma_{k-1}\cdots\Gamma_1 P_{0|-1}^{-1}\hat{x}_{0|-1} + I_k^i + I_k^j.\end{aligned}$$

  - If it does not receive information from the other sensor, it uses the last global estimate it has and propagates it with its own measurements.

Note that we have made no assumptions about the statistics of the packet dropping process or even the knowledge of such statistics to the two sensors. Thus this algorithm is optimal for any packet drop pattern. We have proposed a way to encode the information to be sent for estimation purpose that is optimal under *any* channel packet drop model. In this sense, the scheme is similar in spirit to the work in [10].

IV. MANY SENSORS PRESENT

When more than 2 sensors are present, the scheme no longer remains optimal. The reason for this is the following. The problem is in calculation of the global error covariance matrix $P_{k|k}$. The equation to be used is

$$\left(P_{k|k}\right)^{-1} = \left(P_{k|k-1}\right)^{-1} + \sum_i \left(H_k^i\right)^T\left(R_k^i\right)^{-1}H_k^i.$$

The observation terms H_k^i to be included are those whose information is being fused. For the 2 sensor case, thus terms of both sensors are to be included. For the n-sensor case information from all n sensors might not be fused at *all* time steps, depending on the particular links which drop packets at any time. Thus the proposed method is only sub-optimal in this case. However, if all nodes are communicating with each other fairly regularly, the performance loss will be limited. While it will be nice to characterize this performance loss and/or come up with other algorithms that are optimal for particular packet drop patterns in (say) an expected sense, we have not been able to do that so far. Simulation results seem to suggest that the performance loss is not huge for the case of multiple sensors. In the simulation results shown below, in case a sensor fails to receive packets from all the sensors, it assumes that the contribution from that sensor is 0 and adds up the vectors I^j from the sensors that it has been succesful in communicating to. This is clearly a sub-optimal thing to do; a slightly better scheme would be to try to project the vector I_k^j based on I_{k-1}^j if communication with sensor j was not possible at time step k. However we do not look into such schemes for the present.

V. NUMERICAL EXAMPLES

We now consider some numerical examples to see the performance of the algorithm. The system that we consider is a simple scalar system, with dynamics given by

$$x_{k+1} = -1.25x_k + w_k,$$

where w_k is zero mean white Gaussian noise with variance 1. Consider initially that the system is observed by 2 sensors of the form

$$y_k^i = x_k + v_k^i, \tag{3}$$

where v_k^i is zero mean white and Gaussian with variance 1. Further the noises v_k^i and w_k are all independent of each other. The sensors are trying to obtain the best possible estimate of the state of the system x_k, while communicating over a channel which drops packets randomly with a probability $p = 0.3$. All the plots that we show below are from the viewpoint of sensor 1. Figure 2 shows the evolution of the estimate error covariance as a function of time. The packets from sensor 2 to sensor 1 are lost at times $k = 3$ and 8. The solid line represents the covariance for a hypothetical sensor that has access to all the measurements taken so far. The circles correspond to the performance of our scheme while the remaining curve shows the performance of sensor 1 taking into account its own measurements alone. It can be seen that as packets are lost, the error covariance with our scheme increases; however as soon as a single packet is received, the performance is the same as that of the hypothetical sensor which has received all the measurements so far.

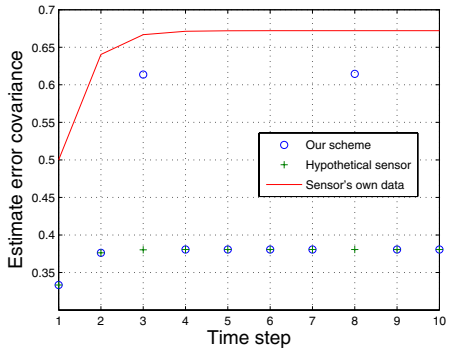

Fig. 2. Performance of estimators under three situations in presence of a packet-dropping link: 2 sensor case

We now consider the same system with measurements being taken by 4 sensors of the same form as (3). The measurement noises are still independent with each having variance 1. Figure 3 shows the estimates at sensor 1 for three schemes when the probability of packet drop in each channel is 0.2. The solid line shows the estimate if the sensor relied on its own measurements. The circles correspond to our scheme while the third curve is that of a hyptothetical sensor that has access to all the measurements. It can be seen that our scheme improves the estimates by a huge amount, though it does not approach the performance of the hypothetical ideal sensor anymore. Note that the plot shows the estimate error and not the covariance; hence the performance of our sensor can be momentarily better than the hypothetical sensor. However in a mean squared sense, the hypothetical sensor, of course, performs the best.

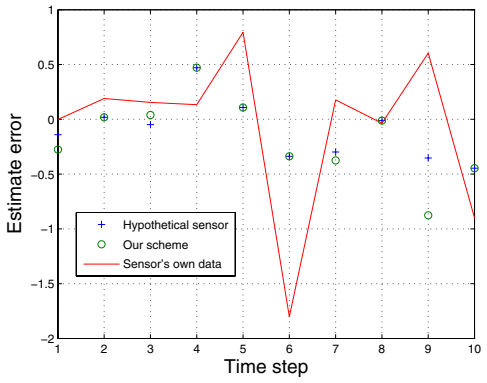

Fig. 3. Performance of estimators under three situations in presence of packet-dropping link: 4 sensor case

To compare the performance of our scheme with other schemes proposed in the literature, we use two sensors with noise variances equal to 10. We compared our scheme with a scheme in which only measurements are being exchanged and another in which local estimates are being fused according to the scheme proposed in [2]. The results are shown in Figure 4. The circles correspond to the estimate given by our scheme, the solid line from the measurement exchange scheme while the asterisk curve from the estimate fusion scheme. The remaining curve is the estimate from the hypothetical ideal sensor as discussed above. Again our scheme performs better than other schemes proposed in the literature.

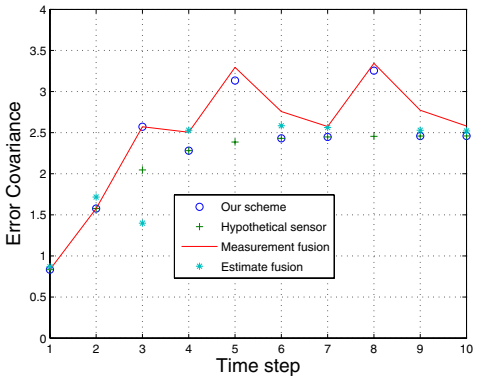

Fig. 4. Performance of estimators using different schemes in the presence of packet-dropping link: 2 sensor case

VI. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this paper we looked at the problem of optimal sensor fusion in the presence of communication channels. We modeled the communication channel as a random switch and posed the following problem: Given that the channel is dropping packets, what should the sensors communicate to each other so that the best estimate is obtained. For the case of two sensors, we solved the problem and the solution was independent of the packet drop pattern. The performance was much better than the schemes proposed in the literature. However, the scheme can not be generalized to an arbitrary number of sensors.

B. Future Work

This work can be extended in many ways. We have simply posed an interesting and relevant problem and solved it for a special case. The general solution for n sensors is still to be worked out. We can also look at more complicated channel models or coding schemes which allow some part of the information to pass through sometimes. Also we have assumed perfect knowledge about all the sensor models. It will be interesting to relax this assumption as well. Another perplexing avenue is to look at information theoretic bounds on the quality of estimate as a function of rate of communication.

VII. ACKNOWLEDGMENTS

The authors would like to thank Amir Farajidana, Haris Vikalo and Timothy Chung for useful discussions on the topic.

REFERENCES

[1] Y. Bar-Shalom. On the track-to-track correlation problem. *IEEE Trans. Automat. Contr.*, AC-26:571–572, 1981.

[2] 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.

[3] T. M. Berg and H. F. Durrant-Whyte. Distributed and decentralized estimation. In *Proc. of the Singapore International Conference on Intelligent Control and Instrumentation*, volume 2, pages 1118–1123, 1992.

[4] S. Mori C. Y. Chong and K. C. Chang. Information fusion in distributed sensor networks. In *Proc. of the American Control Conference*, pages 830–835, 1985.

[5] N. A. Carlson. Federated square root filter for decentralized parallel processes. *IEEE Transactions on Aerospace and Electronic Systems*, AES-26:517–525, 1990.

[6] 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.

[7] C. Y. Chong. Hierarchical estimation. In *Proc. 2nd MIT/ONR* $C^3 Workshop$, 1979.

[8] 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.

[9] O. E. Drummond. Tracklets and a hybrid fusion with process noise. *Proceedings of the SPIE, Signal and Data Processing of Small Targets*, 3163, 1997.

[10] V. Gupta, D. Spanos, B. Hassibi, and R. M. Murray. Optimal LQG control across a packet-dropping link. *IEEE Transactions on Automatic Control*, 2004. Submitted.

[11] D. L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[12] H. R. Hashemipour, S. Roy, and A. J. Laub. Decentralized structures for parallel Kalman filtering. *IEEE Trans. Automat. Contr.*, AC-33:88–94, 1988.

[13] M. F. Hassan, G. Salut, M. G. Singh, and A. Titli. A decentralized computational algorithm for the global Kalman filter. *IEEE Trans. Automat. Contr.*, AC-23:262–268, 1978.

[14] M. E. Liggins II, C. Y. Chong, I. Kadar, M. G. Alford, V. Vannicola, and S. Thomopoulos. Distributed fusion architechtures and algorithms for target tracking. *Proceedings of the IEEE*, 85:95–107, 1997.

[15] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, 2000.

[16] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*. John Wiley and Sons, 1972.

[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, 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.

[19] B. S. Rao and H. F. Durrant-Whyte. Fully decentralized algorithm for multi-sensor Kalman filtering. *IEE proceedings*, 138, 1991.

[20] 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.

[21] D.P. Spanos, R. Olfati-Saber, and R.M. Murray. Distributed sensor fusion using dynamic consensus. In *Proceedings of the 2005 IFAC World Congress (to be presented)*, 2005.

[22] J. L. Speyer. Computation and transmission requirements for a decentralized linear-quadratic Gaussian control problem. *IEEE Trans. Automat. Contr.*, AC-24:266–269, 1979.

[23] R. Viswanathan and P. K. Varshney. Distributed detection with multiple sensors: Part I - fundamentals. *Proceedings of the IEEE*, 85(1):54–63, 1997.

[24] D. Willner, C. B. Chang, and K. P. Dunn. Kalman filter algorithms for a multisensor system. In *Proc. of the 15th Conference on Decision and Control*, pages 570–574, 1976.

[25] 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 Trans. Automat. Contr.*, AC-27:799–813, 1982.