Model Based Information Fusion in Sensor Networks

Alriksson, Peter; Rantzer, Anders

2008

Model Based Information Fusion in Sensor Networks

Peter Alriksson and Anders Rantzer

Department of Automatic Control, Lund University, Sweden

Abstract: In this paper, a model based sensor fusion algorithm for sensor networks is presented. The algorithm, referred to as distributed Kalman filtering is based on a previously presented algorithm with the same name. The weight selection process has been improved yielding performance improvements of several times for the examples studied. Also, solutions to both optimization problems involved in the iterative off-line weight selection process are given as closed form expressions. The algorithm is also demonstrated on a typical signal tracking application.

Keywords: Distributed control and estimation, Sensor networks

1. INTRODUCTION

In recent years the increases in battery and processing power of sensor nodes has made a wide range of sensing applications possible. However as the number of sensors in a network increase the need for efficient data aggregation becomes more and more evident. For a small sensor network routing measurements to a central node using for example Ad hoc On Demand Distance Vector (AODV) routing might be feasible, see Perkins et al. [2003]. However as the network grows, the computational- and network load both in the central node and in bottleneck nodes throughout the network will be a major problem. Also these nodes will drain their energy resources unnecessarily fast.

There are numerous data fusion techniques in the sensor network literature, but most fall into two categories: data driven and model based. An example of a data driven technique would for example be finding the maximum temperature in an area. Each node compares its temperature with its neighbors and only the maximum is transmitted. In this paper we will focus on a model based approach. One simple example would be to estimate the mean temperature in an area. The temperature could then be modeled as a constant quantity that is observed through a number of noisy sensors. In the model based approach the quantity of interest is not required to be directly measurable but can be estimated from previous measurements using a model.

2. PREVIOUS WORK

The technique used in this paper is often referred to as distributed Kalman filtering. In a distributed Kalman filter, nodes exchange estimates of the quantity of interest possibly together with the their local measurements.

An early reference is Durrant-Whyte et al. [1990] where a decentralized Kalman filter was proposed. However, this algorithm requires every node to be able to communicate with every other node, which is not possible in the setup studied here.

One common technique is to apply consensus filters, see Olfati-Saber et al. [2007], on various quantities such as the measurements, covariances and/or state estimates. These consensus filters usually operate at a faster rate than the sampling rate, thus allowing the network to reach an agreement before the state estimate is updated. Under this assumption the choice of Kalman gain can be treated in the same way as a centralized Kalman filter. Recent papers in this area include Olfati-Saber [2007], Spanos et al. [2005] and Xiao et al. [2005]. In Carli et al. [2007] it was noted that if the assumption of agreement is not fulfilled the optimal Kalman gain for a centralized filter does not coincide with that of a distributed. This issue was also addressed in Schizas et al. [2007]. In Speranzon et al. [2006] the scalar case was studied under the assumption that nodes communicate only once between each measurement.

In Alriksson and Rantzer [2006] a two step procedure for distributed Kalman filtering was developed. This algorithm consists of one part that is done online and one offline part where parameters for the online part are selected. This paper aims at improving the parameter selection step of that paper.

This paper is organized as follows. In Section 3 we present the mathematical problem studied and give necessary assumptions. In Section 4 the online part of the algorithm is given for clarity. Section 5 presents the improved offline parameter selection process and in Section 6 three numerical examples are given.

3. PROBLEM FORMULATION

Consider the following discrete-time linear system

$$x(k+1) = Ax(k) + w(k) \tag{1}$$

where $x(k) \in \mathbf{R}^n$ is the state of the system and $w(k) \in \mathbf{R}^n$ is a stochastic disturbance. The disturbance is assumed to be a white zero mean Gaussian process with covariance defined in (3).

The process is observed by N agents each with some processing and communication capability. The agents are

labeled $i = 1, 2, \ldots, N$ and form the set $\mathcal{V}$. The communication topology is modeled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the edge (i, j) is in $\mathcal{E}$ if and only if node i and node j can exchange messages. The nodes to which node i communicates are called neighbors and are contained in the set N_i. Note that node i is also included in the set N_i.

Each node observes the process (1) by a measurement $y_i(k) \in \mathbf{R}^{m_i}$ of the form

$$y_i(k) = C_i x(k) + e_i(k) \tag{2}$$

where $e_i(k) \in \mathbf{R}^{m_i}$ is a white zero mean Gaussian process. The process- and measurement disturbances are correlated according to

$$\mathbf{E} \begin{bmatrix} w(k) \\ e_1(k) \\ \vdots \\ e_N(k) \end{bmatrix} \begin{bmatrix} w(l) \\ e_1(l) \\ \vdots \\ e_N(l) \end{bmatrix}^T = \begin{bmatrix} R_w & 0 & \ldots & 0 \\ 0 & R_{e11} & \ldots & R_{e1N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & R_{eN1} & \ldots & R_{eNN} \end{bmatrix} \delta_{kl} \tag{3}$$

where $\delta_{kl} = 1$ only if $k = l$. Note that this is a heterogeneous setup where each agent is allowed to to take measurements of arbitrary size and precision. Further the disturbances acting on the measurements are allowed to be correlated.

Each node is only allowed to communicate *estimates* with its neighbors and only once between each measurement. Further the only assumption made on the graph structure is that it has to be connected, other assumptions such as requiring it to be loop free are not necessary. No node is superior to any other and thus no central processing is allowed after deployment. This setup is somewhat different from the setup used in for example distributed control problems where each node in the graph also has dynamics associated with it. The reader should think of the problem studied here as for example a network of sensors trying to estimate the position of an external object they observe.

The goal is to make sure that every node in the network has a good estimate $\hat{x}_i(k)$ of the state $x(k)$.

4. ONLINE COMPUTATIONS

The algorithm consists of the two traditional estimation steps measurement update and prediction, together with an additional step where the nodes communicate and merge estimates. We will refer to an estimate after measurement update as local and after the communication step as regional.

(1) **Measurement update**
The local estimate $\hat{x}_i^{local}(k|k)$ is formed by the predicted regional estimate $\hat{x}_i^{reg}(k|k-1)$ and the local measurement $y_i(k)$

$$\hat{x}_i^{local}(k|k) = \hat{x}_i^{reg}(k|k-1) + K_i[y_i(k) - C_i\hat{x}_i^{reg}(k|k-1)] \tag{4}$$

where K_i is computed off-line. The predicted estimate at time zero is defined as $\hat{x}_i^{reg}(0|-1) = \hat{x}_0$ where $\hat{x}_0$ is the initial estimate of $x(0)$.

(2) **Merging**
First the agents exchange their estimates over the communication channel. This communication is assumed to be error and delay free. The merged estimate $\hat{x}_i^{reg}(k|k)$ in node i is defined as a linear combination of the estimates in the neighboring nodes N_i.

$$\hat{x}_i^{reg}(k|k) = \sum_{j \in N_i} W_{ij}\hat{x}_j^{local}(k|k) \tag{5}$$

The weighting matrices W_{ij} are computed off-line by the procedure described in Section 5.

(3) **Prediction**
Because the measurement- and process noises are independent the prediction step only includes

$$\hat{x}_i^{reg}(k+1|k) = A\hat{x}_i^{reg}(k|k) \tag{6}$$

5. OFFLINE PARAMETER SELECTION

As the best estimate will be available after the merging step we will focus on minimizing the estimation error covariance after this step. First, let the estimation error in node i be denoted

$$\tilde{x}_i(k|k) = x(k) - \hat{x}_i^{reg}(k|k) \tag{7}$$

and its (cross)covariance

$$P_{ij}^{reg}(k|k) = E\tilde{x}_i(k|k)\tilde{x}_j^T(k|k) \tag{8}$$

Next introduce the stacked estimation error

$$\tilde{x}(k|k) = \begin{bmatrix} \tilde{x}_1(k|k) \\ \vdots \\ \tilde{x}_N(k|k) \end{bmatrix} \tag{9}$$

with covariance $P^{reg}(k|k)$. Using (5) and requiring that

$$W_{ij} = 0 \text{ if } (i, j) \notin \mathcal{E} \tag{10}$$

the covariance after step 2) can be written as

$$P^{reg}(k|k) = WP^{local}(k|k)W^T \tag{11}$$

To keep the estimate unbiased we also need to require that

$$\sum_{j \in N_i} W_{ij} = I \quad \forall i \in \mathcal{V} \tag{12}$$

The covariance $P^{local}(k|k)$ after step 1) can be expressed as

$$P^{local}(k|k) = \begin{bmatrix} I & \tilde{K} \end{bmatrix} F \begin{bmatrix} I & \tilde{K} \end{bmatrix}^T \tag{13}$$

where

$$F = \begin{bmatrix} I \\ -\tilde{C} \end{bmatrix} P^{reg}(k|k-1) \begin{bmatrix} I \\ -\tilde{C} \end{bmatrix}^T + \begin{bmatrix} 0 & 0 \\ 0 & R_e \end{bmatrix} \tag{14}$$

and

$$\tilde{K} = \begin{bmatrix} K_1 & & \\ & \ddots & \\ & & K_N \end{bmatrix} \quad \tilde{C} = \begin{bmatrix} C_1 & & \\ & \ddots & \\ & & C_N \end{bmatrix} \tag{15}$$

After step 3) each block of the covariance matrix is updated as

$$P_{ij}^{reg}(k|k-1) = AP_{ij}^{reg}(k-1|k-1)A^T + R_w \tag{16}$$

with $P_{ij}^{reg}(0|-1) = P_0$ where P_0 is the initial estimation error covariance. Equations (11), (13) and (16) form an iterative procedure for computing the steady state covariance as time approaches infinity for given values of W and $\tilde{K}$.

Ideally, we would like to find values W and $\tilde{K}$ that minimizes the steady state value of tr $P^{reg}(k|k)$ as $k \to \infty$ subject to the constraints (10) and (12). This non-convex problem will be approximated in two steps. Instead of minimizing the steady state covariance directly, an approximate iterative procedure in analogy with the standard Kalman filter will be used.

Combining (13) and (11) the minimization problem to be solved in each iteration can be written as

$$\begin{array}{ll}\min\limits_{\tilde{K},W} & \text{tr } W\begin{bmatrix} I & \tilde{K}\end{bmatrix} F \begin{bmatrix} I & \tilde{K}\end{bmatrix}^T W^T \\ \text{s.t} & (10) \text{ and } (12)\end{array} \tag{17}$$

This problem will be solved using an alternating minimization type method.

The algorithm is divided into three steps. The first two steps correspond to the minimization problem (17) and in the third step the covariance is updated. The procedure is then iterated until convergence.

(1) **K-step**

$$\tilde{K}(k) = \arg\min_{\tilde{K}} \text{ tr } W(k-1)\begin{bmatrix} I & \tilde{K}\end{bmatrix} F \begin{bmatrix} I & \tilde{K}\end{bmatrix}^T W^T(k-1)$$

(2) **W-step**

For all $i \in \mathcal{V}$

$$\begin{array}{ll} W_{i\cdot}(k) = \arg\min\limits_{W_{i\cdot}} & \text{tr } W_{i\cdot}\begin{bmatrix} I & \tilde{K}(k)\end{bmatrix} F \begin{bmatrix} I & \tilde{K}(k)\end{bmatrix}^T W_{i\cdot}^T \\ \text{s.t} & (10) \text{ and } (12)\end{array}$$

(3) **P-step**

$$P^{reg}(k|k) = W(k)\begin{bmatrix} I & \tilde{K}(k)\end{bmatrix} F \begin{bmatrix} I & \tilde{K}(k)\end{bmatrix}^T W^T(k)$$
$$P_{ij}^{reg}(k+1|k) = AP_{ij}^{reg}(k|k)A^T + R_w \qquad \forall i,j \in \mathcal{V}$$

The algorithm is initialized with $P_{ij}^{\text{reg}}(0|-1) = P_0$ and $W(0) = I$. Note that in step 2) the minimization with respect to $W_{i\cdot}$, that is block row i, for each $i \in \mathcal{V}$ is equivalent to minimization with respect to the full matrix W.

Compared to the algorithm proposed in Alriksson and Rantzer [2006] the K-step now takes into account the fact that the estimates will be merged. Both the K- and W-step are quadratic minimization problems with explicit solutions which allows the algorithm to be applied to large scale systems.

5.1 The K-step

In this section the optimization problem from the K-step will be studied. To simplify notation time indices are dropped:

$$\min_{\tilde{K}} \text{ tr } W\begin{bmatrix} I & \tilde{K}\end{bmatrix} F \begin{bmatrix} I & \tilde{K}\end{bmatrix}^T W^T \tag{18}$$

In this section linear conditions that the optimal $\tilde{K}$ have to fulfill will be derived. First partition F as in

$$\begin{bmatrix} I & \tilde{K}\end{bmatrix} \begin{bmatrix} F_{11} & F_{12} \\ F_{12}^T & F_{22}\end{bmatrix} \begin{bmatrix} I & \tilde{K}\end{bmatrix}^T \tag{19}$$

To isolate the free parameters in $\tilde{K}$ it is expressed as a sum

$$\tilde{K} = \sum_{i=1}^{N} U_i^T K_i V_i \tag{20}$$

where

$$\begin{aligned} U_i &= \begin{bmatrix} 0_{n\times n(i-1)} & I_{n\times n} & 0_{n\times n(N-i)}\end{bmatrix} \\ V_i &= \begin{bmatrix} 0_{m_i\times l_i} & I_{m_i\times m_i} & 0_{m_i\times \tilde{l}_i}\end{bmatrix}\end{aligned} \tag{21}$$

$$l_i = \sum_{j=1}^{i-1} m_j \text{ and } \tilde{l}_i = \sum_{j=i+1}^{N} m_j \tag{22}$$

Using the decomposition (20) of $\tilde{K}$, conditions for optimality of (18) are given by

$$U_i W^T W \begin{bmatrix} I & \tilde{K}\end{bmatrix} \begin{bmatrix} F_{12} \\ F_{22}\end{bmatrix} V_i^T = 0 \quad \forall i \in \mathcal{V} \tag{23}$$

To simplify notation first introduce

$$G_{ij} = U_i W^T W U_j^T \text{ and } H_{ij} = V_j F_{22} V_i^T \tag{24}$$

$$Q_i = U_i W^T W F_{12} V_i^T \tag{25}$$

Now (23) can be rewritten as

$$\sum_{j=1}^{N} G_{ij} K_j H_{ij} = -Q_i \ , \ \forall i \in \mathcal{V} \tag{26}$$

To solve this set of matrix equations vectorization of the matrices will be used:

$$\sum_{j=1}^{N} (H_{ij}^T \otimes G_{ij})\bar{K}_j = -\bar{Q}_i \tag{27}$$

where

$$\bar{K}_j = \text{vec}(K_j) \text{ and } \bar{Q}_i = \text{vec}(Q_i) \tag{28}$$

This can be written in matrix form as

$$\begin{bmatrix} H_{11}^T \otimes G_{11} & \cdots & H_{1N}^T \otimes G_{1N} \\ \vdots & \ddots & \vdots \\ H_{N1}^T \otimes G_{N1} & \cdots & H_{NN}^T \otimes G_{NN}\end{bmatrix} \begin{bmatrix} \bar{K}_1 \\ \vdots \\ \bar{K}_N\end{bmatrix} = - \begin{bmatrix} \bar{Q}_1 \\ \vdots \\ \bar{Q}_N\end{bmatrix} \tag{29}$$

Thus we have derived linear equations for the optimal $\bar{K}_i$ which gives the optimal $\tilde{K}$.

5.2 W-Step

Introducing the sparsity constraint (10) is equivalent to removing rows and columns corresponding to weights that are required to be zero. Thus for each i the optimization problem can be written as

$$\begin{array}{ll}\min\limits_{\tilde{W}} & \text{tr } \tilde{W}\tilde{P}\tilde{W}^T \\ \text{s.t.} & \tilde{W}e = I_n\end{array} \tag{30}$$

where $e = [I_n \dots I_n]^T$. Here $\tilde{W}$ contains the non-zero blocks of $W_{i\cdot}$ and $\tilde{P}$ the corresponding elements of the matrix $\begin{bmatrix} I & \tilde{K}(k)\end{bmatrix} F \begin{bmatrix} I & \tilde{K}(k)\end{bmatrix}^T$. Using Lagrange multipliers it can be shown, see Sun and Deng [2004], that conditions for optimality are

$$\underbrace{\begin{bmatrix} \tilde{P} & e \\ e^T & 0\end{bmatrix}}_{G} \begin{bmatrix} \tilde{W}^T \\ \Lambda\end{bmatrix} = \begin{bmatrix} 0 \\ I_n\end{bmatrix} \tag{31}$$

The equation system (31) is in general underdetermined, so to get a unique solution the following minimization problem is introduced

$$\begin{array}{ll}\min\limits_{\tilde{W}} & \text{tr } \tilde{W}\tilde{W}^T \\ \text{s.t.} & (31)\end{array} \tag{32}$$

All solutions satisfying (31) can be parametrized in terms of V as

$$\begin{bmatrix} \tilde{W}^T \\ \Lambda\end{bmatrix} = \underbrace{G^\dagger \begin{bmatrix} 0 \\ I_n\end{bmatrix}}_{d} + G^0 V \tag{33}$$

where $G^\dagger$ denotes the Moore-Penrose pseudo inverse and G^0 a matrix of vectors spanning the null space of G. Now (32) can be rewritten as

$$\min_V \operatorname{tr} \begin{bmatrix} V \\ I \end{bmatrix}^T \begin{bmatrix} I & (G_1^0)^T d_1 \\ d_1^T G_1^0 & d_1^T d_1 \end{bmatrix} \begin{bmatrix} V \\ I \end{bmatrix} \tag{34}$$

where d_1 and G_1^0 are the parts corresponding to $\tilde{W}^T$. The solution to this unconstrained quadratic minimization problem is given by $V = -(G_1^0)^T d_1$. Thus the optimal $\tilde{W}$ is given by

$$\tilde{W} = d_1^T (I - G_1^0 (G_1^0)^T) \tag{35}$$

6. NUMERICAL EXAMPLES

In this section three numerical examples will be studied. The first example is chosen to illustrate the performance improvement gained by modifying the K-step compared to the algorithm presented in Alriksson and Rantzer [2006]. The second example illustrates how varying the communication topology influences achieved performance. The third example illustrates an application where a scalar time varying signal, such as for example the temperature in an area, is measured by a sensor network.

6.1 Performance Comparison

Ideally, when comparing two suboptimal algorithms one would like to compare them to the results of the optimal solution. However in this case the $\tilde{K}$ and W yielding the optimal covariance P^{opt} can only be computed for very simple systems with special structure.

As a comparison we will use an observer scheme that relies on communication of measurements rather than estimates, but respects the imposed communication topology. For all nodes to be able to maintain an optimal estimate, they must have access to all measurements. The amount of communication required to achieve this greatly exceeds the required communication for the scheme in Section 4 and in most cases is practically impossible. This scheme will however yield a lower estimation error covariance P^{meas} than P^{opt} because having access to all measurements is clearly at least as good as having access to estimates generated by these measurements. Thus we have

$$P^{\text{meas}} \leq P^{\text{opt}} \leq P \tag{36}$$

where P refers to the scheme presented in Section 4.

As communication is only allowed to occur once every sample, the communication topology will impose a delay equal to the graph distance to a particular node. If measurement noise in different nodes is assumed independent, delayed measurements can be incorporated in the current estimate by extending the state space.

Note that the covariance will be different in different nodes due to the imposed communication topology. To evaluate overall performance the mean over all nodes in the network will be used.

In Fig. 1 the relative performance $\frac{\operatorname{tr} P - \operatorname{tr} P^{\text{meas}}}{\operatorname{tr} P^{\text{meas}}}$ is plotted as a function of the process noise R_w for both weight selection algorithms. The performance was evaluated for 843 randomly generated second order systems with a communication topology described by graphs with 10 nodes and 5.95 neighbors on average. The shaded regions

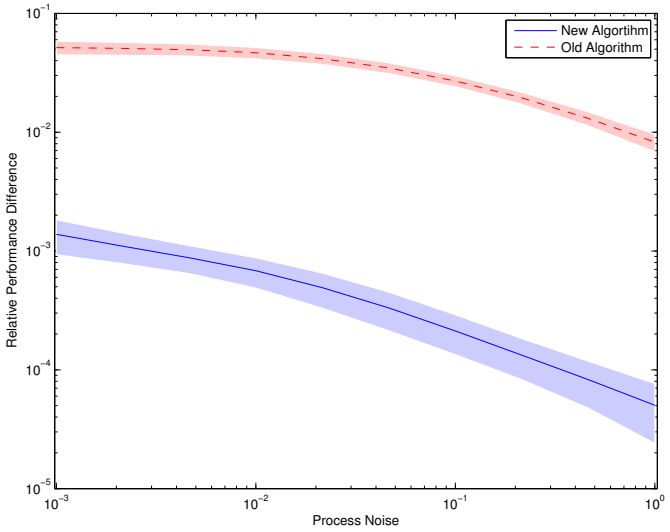

Fig. 1. Comparison of the relative difference $\frac{\operatorname{tr} P - \operatorname{tr} P^{\text{meas}}}{\operatorname{tr} P^{\text{meas}}}$ for the algorithm presented in Alriksson and Rantzer [2006] and the one presented here for 843 randomly generated second order systems on a graph with 10 nodes with 5.95 neighbors on average. The shaded regions are 95% confidence intervals for the mean over all 843 systems.

are 95% confidence intervals for the mean over all 843 systems. The measurement noise covariance matrix R_e was chosen as the identity matrix.

Because estimates are used as information carriers and communication is only allowed to take place once every sampling interval the process noise parameter R_w determines the effective distance from which a node collects information. Therefore one would expect the suboptimal solution to deteriorate as R_w is decreases, this is also confirmed by the results in Fig. 1.

6.2 Connectivity Dependencies

The effects on estimation performance of 1620 randomly generated communication topologies with 20 nodes was studied for a system with integrator dynamics. As a measure of connectivity the average number of neighbors was used. An alternative measure would be the algebraic connectivity of the associated graph. Both these measures give similar results but the average number of neighbors is more intuitive.

As mentioned in Section 6.1 the effective radius from which information is used increases as R_w decreases. Therefore, choosing a small value, such as $R_w = 0.001$, of the process noise parameter will make effects caused by different communication topologies more evident.

In Fig. 2 the variance is plotted as a function of the average number of neighbors for each of the 1620 topologies using the algorithm presented in Alriksson and Rantzer [2006], the one presented in Section 5 and the scheme with delayed measurements presented in Section 6.1. The improvement compared to the previous weight selection algorithm is more evident for strongly connected graphs. However even for very sparse graphs the improvement is more than 50%.

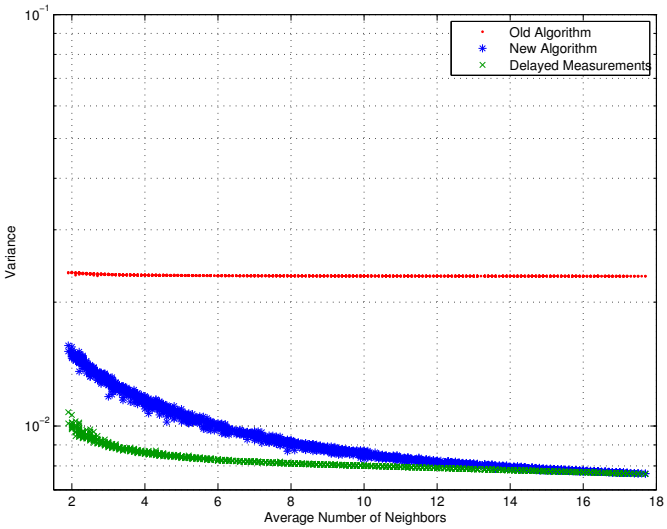

Fig. 2. Variance plotted as a function of the average number of neighbors for 1620 randomly generated graphs of size 20 using the algorithm presented in Alriksson and Rantzer [2006], the one presented in Section 5 and the scheme with delayed measurements presented in Section 6.1.

6.3 Signal Tracking

This example aims at demonstrating how the proposed estimation scheme can be used in a situation where a sensor network is used to estimate the mean of a time varying signal in an area. Here 50 sensors are used to measure a signal described by

$$x(k) = \sin\left(\frac{2\pi}{100}k\right) + \sin\left(\frac{4\pi}{100}k\right)$$

Each node measures $x(k)$ corrupted by Gaussian white noise with unit variance. Further, the noise is assumed independent between nodes.

Two different signal models will be used: an integrator and a double integrator. The reason for not using a fourth order model capable of fully describing $x(k)$ is that in general, an exact model of the signal studied is hardly ever available.

Four different estimation schemes will be compared:

Centralized refers to a scenario where measurements are fused in a central node without any communication delay.

Delayed Measurements refers to the scenario described in Section 6.1.

Distributed refers to the scheme described in Section 4 with the weight selection procedure of Section 5.

Local refers to a scenario where no communication is used. Here each node runs a Kalman filter based on local information only.

As both the integrator and double integrator model differs from the true model describing $x(k)$ the choice of process noise covariance R_v is crucial for the performance. Here two different ways of choosing R_v will be used. The first involves making a maximum likelihood estimate of the process noise covariance R_v for the centralized case and then using that estimate as the true value. In the case of a double integrator model the optimal ML-estimate is

$$\hat{R}_v = \begin{bmatrix} 0 & 0 \\ 0 & 0.001 \end{bmatrix}$$

and for the case of an integrator model $\hat{R}_v = 0.03$.

The second approach aims at making a fair comparison between the four schemes. To this end, both the process noise covariance and the model structure will be optimized to yield the best performance (measured as the root mean square (RMS) of the estimation error). The optimal configurations are summarized in Table 1.

	Signal Model	$\hat{R}_v$
Centralized	Double Integrator	$\begin{bmatrix} 0 & 0 \\ 0 & 0.001 \end{bmatrix}$
Delayed Measurements	Integrator	0.01
Distributed	Double Integrator	$\begin{bmatrix} 0 & 0 \\ 0 & 0.002 \end{bmatrix}$
Local	Integrator	0.09

Table 1. Optimal configuration for the four schemes.

The performance, measured as RMS of the estimation error, for the four different schemes in the three different model setups is presented in Fig. 3. In the two middle cases the estimation performance will vary depending on which node is studied, this is represented as shaded boxes. As expected, using a more complex model generally improves performance except for the case with delayed measurements. A possible explanation for this is that to make use of old measurements the model must be used heavily, thus making the scheme very sensitive to modeling errors.

In the distributed case the double integrator model improves performance significantly. This shows the importance of allowing a more complex model structure than a simple first order model that is often assumed.

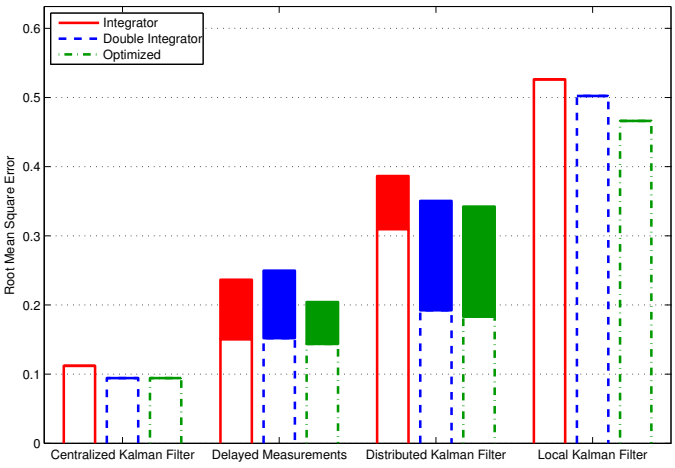

Fig. 3. RMS of the estimation error, for the four different schemes in three different setups. In the cases referred to as integrator and double integrator, the process noise parameter was chosen as the ML-estimate without time delays. In the optimized case, both the model structure and parameters were optimized for the specific estimation scheme. The shaded regions represent max- and minimum values among all nodes.

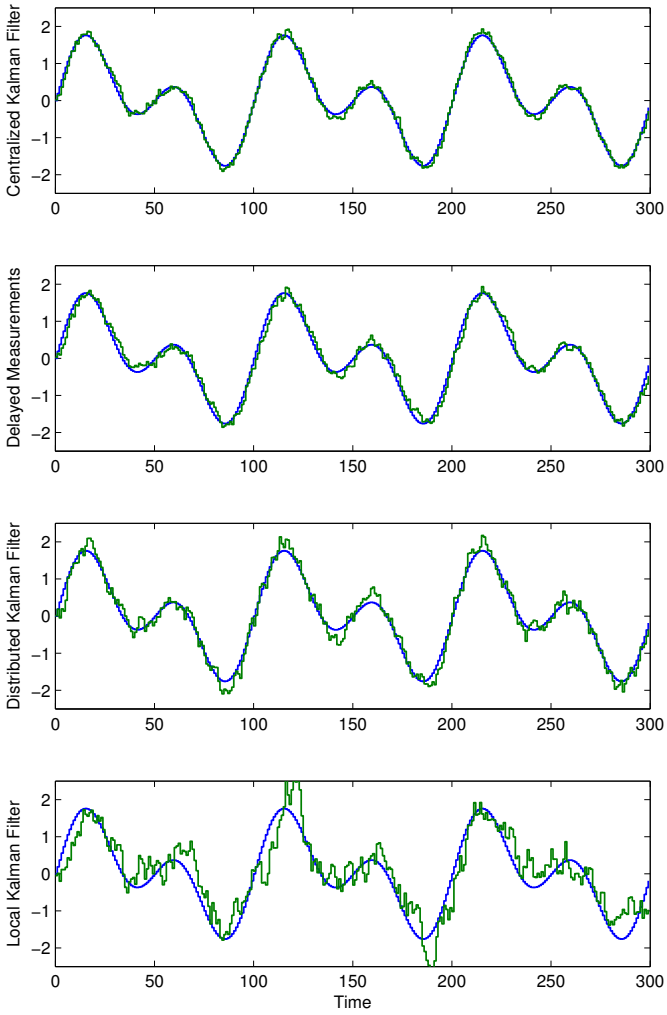

Fig. 4. Typical trajectories for the four different estimation schemes together with the true value of $x(k)$.

In Fig. 4 typical trajectories are shown for the four different estimation schemes together with the true value of $x(k)$.

7. CONCLUSIONS

In this paper an enhanced weight selection algorithm for the distributed Kalman filter algorithm presented in Alriksson and Rantzer [2006] has been presented. Improvements in terms of covariance reduction of several times have been noticed for the examples studied. The algorithm relies on the assumption that both the dynamics and communication topology are time-invariant and known at deployment. Slow variations in the communication topology and dynamics can be handled by recomputing the parameters on a regular basis. Fast variations in the communication topology can be treated as packet loss, against which the algorithm has proved robust.

Ideally both the weights for neighboring estimates, W, and local measurements, $\tilde{K}$, should be optimized jointly. However this is a non convex problem in general. Instead of a joint optimization in W and $\tilde{K}$, W is held constant equal to the value from the previous iteration when $\tilde{K}$ is optimized and $\tilde{K}$ is held constant while W is optimized. This reduces both optimization problems to quadratic optimization problems for which expressions in closed form are derived. Compared to the previous algorithm, fewer but bigger optimization problems are now solved.

The second contribution of this paper is to evaluate performance of the estimation algorithm on a number of numerical examples. The first two numerical Monte Carlo studies conclude that a significant performance improvement has been gained through the new weight selection algorithm. In the third numerical example the importance of allowing a more complex signal model than for example the commonly used integrator model is highlighted.

ACKNOWLEDGEMENTS

This work was partially funded by the RUNES Integrated Project contract IST-2004-004536.

REFERENCES

Peter Alriksson and Anders Rantzer. Distributed Kalman filtering using weighted averaging. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan, July 2006.

Ruggero Carli, Alessandro Chiuso, Luca Schenato, and Sandro Zampieri. Distributed Kalman filtering using consensus strategies. In *Proceedings of the 46th Conference on Decision and Control*, pages 5486–5491, New Orleans, LA, USA, December 2007.

H.F. Durrant-Whyte, B.Y.S. Rao, and H. Hu. Toward a fully decentralized architecture for multi-sensor data fusion. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1331–1336 vol.2, 1990.

R. Olfati-Saber. Distributed Kalman filtering for sensor networks. In *Proceedings of the 46th Conference on Decision and Control*, pages 5492–5498, New Orleans, LA, USA, December 2007.

R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing, 2003.

Ioannis D. Schizas, Georgios B. Giannakis, Stergios I. Roumeliotis, and Alejandro Ribeiro. Anytime optimal distributed Kalman filtering and smoothing. *IEEE/SP 14th Workshop on Statistical Signal Processing, 2007. SSP '07*, pages 368–372, 2007. doi: 10.1109/SSP.2007.4301282.

D. P. Spanos, R. Olfati-Saber, and R. M. Murray. Distributed sensor fusion using dynamic consensus. In *Proccedings of the 16th IFAC World Congress*, July 2005.

A. Speranzon, C. Fischione, and K. H. Johansson. Distributed and collaborative estimation over wireless sensor networks. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 1025–1030, San Diego, December 2006.

S.-L. Sun and Z.-L. Deng. Multi-sensor optimal information fusion Kalman filter. *Automatica*, 40(6):1017–1023, 2004.

L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the Information Processing for Sensor Networks (IPSN'05)*, 2005.