12th International Conference on Information Fusion
Seattle, WA, USA, July 6-9, 2009

A Composite Trust Model and its Application to Collaborative Distributed Information Fusion

Ion Matei John S. Baras Tao Jiang
Institute for Systems Research
and Department of Electrical and Computer Engineering
University of Maryland
College Park, MD
{imatei,baras,tjiang}@umd.edu

Abstract – *We consider the distributed state estimation of a linear dynamic system, observed by various sensors, as a problem in information fusion. We introduce a novel model of trust, using weights on the graph links and nodes that represent the sensor network. These weights can represent several interpretations of trustworthiness in sensor networks. We describe two algorithms that integrate distributed Kalman filtering with these trust weights. We consider two interpretations of these trust weights as information accuracy and reliability. We show that by appropriate use of these weights the distributed estimation algorithm avoids using information from untrusted sensors. Simulation experiments further demonstrate the behavior of these algorithms.*

Keywords: Trust, Distributed, Filtering

1 Introduction

Sensor networks consist of spatially distributed autonomous sensors to cooperatively monitor physical or environmental conditions. Sensors self-organize to derive appropriate inferences from the information gained in real-time. Emerging technologies for sensing and pervasive computing have extended the scope of information fusion for distributed sensor networks from a simple merger of multiple sensor inputs to fusion of data and knowledge from multiple perspectives [11]. In other words, sensors in the emerging sensor networks not only sense, disseminate and fuse information, they also function as communication nodes, decision makers, application operators, etc.. Therefore, theories, models, analysis and synthesis methods we propose to study sensor networks must be hybrid and composite with considerations of various domains.

The hybrid and composite characteristics of information fusion in sensor networks pose more challenges on reach accurate and reliable decisions. For instance, sensors make use of wireless channels for communication. Wireless networks are typically plagued with unreliable communication: link qualities can change over time, and (potentially deliberate) interference can complete block communication. In sensor networks, these problems are exacerbated, since sensors have limited storage, computing power and energy.

One crucial question is: how a user can trust the information provided by the sensor network? Our research efforts are motivated by two key observations. First, due to the autonomous and dynamic nature of sensor networks, uncertainty of information accuracy has to be taken into consideration. Second, sensor networks often operate unattended in physically insecure environments, and are designed with an emphasis on numbers and low cost which makes measures such as tamper-proof hardware not cost effective. Therefore, we cannot only resort to costly cryptography to design secure sensor networks. In this paper, we propose a composite trust model based on the *valued directed graphs with weighted nodes* [15] as a methodology to improve the quality of information fusion. We further apply this trust model to the distributed estimation problem to show how our trust model can be effectively used in various problems arising in sensor networks.

2 Problem formulation

We consider a sensor network with N sensors, indexed by i. The network is used for the state estimation of a linear random process given by:

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

where $x(k) \in \mathbb{R}^n$ is the state vector and $w(k) \in \mathbb{R}^n$ is the state noise, assumed Gaussian with zero mean and covariance matrix Q. The initial state x_0 has a Gaussian distribution, with mean μ_0 and covariance matrix P_0.

We assume each sensor has a linear sensing model given by:

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

where $y_i(k) \in \mathbb{R}^{p_i}$ is the observation of the state $x(k)$ made by sensor i and $v_i(k) \in \mathbb{R}^{p_i}$ is the measurement noise assumed Gaussian with zero mean and covariance matrix R_i.

The goal of each sensor i is to compute an accurate estimation of the state $x(k)$, using the local measurements $y_i(k)$, the information received from the sensors in its communication neighborhood (e.g. measurements and estimates) and the confidence in the information received from

other sensors provided by the trust model described in the following sections.

Each sensor i has a communication neighborhood containing sensors with whom the sensor can exchange information. Let $\mathcal{N}_i$ denote such communication neighborhood:

$$\mathcal{N}_i = \{j \mid i \text{ exchanges information with } j\}.$$

The communication neighborhoods of the sensors determine a communication graph with N vertices, such that a link from i to j exists if sensor i sends information to sensor j.

We attach a positive value T_{ij} to each link (j, i) which represents the confidence value that sensor i places on the information coming from sensor j. The value T_{ij} represents a measure of the trust sensor i has in the information received from sensor j.

There are many different definitions of "trust" depending on the particular domains. An operational definition of "trust" for information, mainly considers two aspects: information *accuracy* and *reliability*. Accuracy reflects the deviation of the information from truth and reliability is confidence in the assessment of accuracy. In this paper, we apply trust weights to the distributed estimation problem where these two aspects of trust are investigated separately.

3 Composite Trust Model

Trust appears in sensor networks in various ways and meanings. Thus one can refer to the reduced trustworthiness of a sensor, meaning that the sensor may have been compromised. Or one refers to the trustworthiness of the data transmitted by a sensor. Or one can refer to a compromised link due to jamming, which reduces the trustworthiness of the link. Thus trust in sensor networks, and more generally in hybrid networks consisting of collaborating humans and automated agents (sensors, actuators, computers) is a composite entity, represented by several metrics and/or parameters.

Due to the composite nature of trust, it is not enough to analyze models of trust based on an individual domain. Rather, we must develop models and methodologies that can represent and analyze the effects of trust across domains. For example, as was shown in our recent work [17], one can analyze jointly the effects of trust coming from a social network perspective, on a communication network (supporting the social network) performance, by using trust related weights on the nodes, and by extending the recently developed network utility maximization (NUM) approach [18] to systematically develop cross-domain design of high performance communication network protocols that are security or trust aware. In this paper, we introduce the valued directed graph with weighted nodes [15] as our first level model (in terms of complexity and sophistication) of composite trust. We integrate the valued directed graph with weighted nodes in the distributed estimation problem as an application of the composite trust model. Our ultimate research goal is to extend the valued directed graph with weighted nodes composite trust model to include not only numerical weights, but also numerical constraints, logical variables, logical constraints and other forms of metrics, which better capture the hybrid nature of information fusion in the emerging sensor networks. This ultimate goal is not addressed in this paper, but it represents our future work.

Trust, as a composite concept, consists of components that are derived from different domains. We represent the composite trust as *trust weights* on nodes or links. A trust weight is a numerical representation of trust, which represents reliability of a node or a link to conduct certain function/ action and risks for others to cooperate with the node or the link in such function/action. These weighted nodes and links form the *trust graph*, which is a valued directed graph with weighted nodes.

Two types of trust we are considering in our composite model: global trust and local trust. The global trust weight assigns to a node (trustee) a unique trust value, independently of the node (trustor) that is evaluating the trustee's trustworthiness. On the other hand, a local trust weight (personalized trust weight) provides a personalized trust value that depends on the point of view of the trustor. Global trust weights are good when nodes have the same criterion on trustworthiness. For instance, in Section 5.1, the trust of the information provided by a specific node depends on the estimation accuracy of this node, which is globally identical regardless of the trustor. Therefore, we assign a value T_j to node j which is the inverse of estimation error and for any j's neighbor, say node i, its trust value on node j is $T_{ij} = T_j$. The local trust depends on each trustor's preference, and the trust values are assigned as weights on links. For instance, in Section 5.2, the trust of the information provided by a specific node depends on estimation reliability, which is compared with the current estimate of the trustor. Apparently, this notion of trust is a personalized concept. Therefore, we assign a weight T_{ij} to each link ij. Notice that more than one weights are allowed on a node or a link to represent multiple trust relations in different domains.

In order to represent trustworthiness well, trust establishment schemes must ensure that these trust weights are the best evaluation of trustworthiness of a node or a link. In centralized networks, with the help of cryptography and trusted third part, trust weights are proven to be able to correctly represent the true trustworthiness of agents [10]. However, sensor networks with decentralized control pose more difficult challenges on trust establishment. In this case, the realization of robust and accountable trust establishment is based on healthy member cooperation, namely, community monitoring. The community is a set of nodes that watch each other execute all network protocols, such as the generation or forwarding of data or control packets. The fact that the medium is wireless makes this intentional or unintentional collaboration real. Nodes watch others activities by overhearing their wireless signals or sending out probes with secured acknowledgement (probes to nodes with shared key). The trustworthiness of a node or a link is evaluated based on opinions of the community. A high level view of our general model for composite trust gener-

ation and management is shown in Figure 1. The detailed

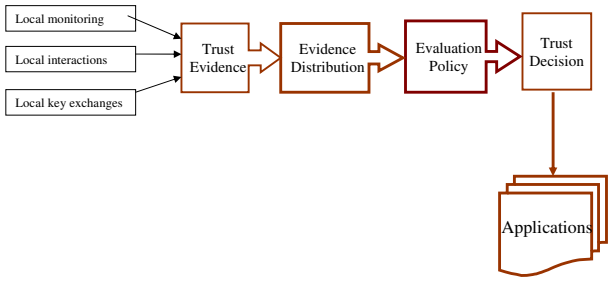

Figure 1: Abstract view of the general model for composite trust generation and management

schemes for trust evidence distribution and distributed trust evaluation are beyond the scope of this paper and we refer to our previous work on trust [16, 13, 14, 6].

There are various ways to numerically represent trust weights. In some trust schemes, continuous or discrete numerical values are assigned to measure the level of trustworthiness. For example, in [5], an entitys opinion about the trustworthiness of a certificate is described by a continuous value in [0, 1]. In [6], a 2-tuple in $[0,1]^2$ describes the trust opinion. In [12], the metric is a triplet in $[0,1]^3$, where the elements in the triplet represent belief, disbelief, and uncertainty, respectively. Trust can also be interpreted as probability. In [7], subjective probability is defined, while objective probability is used in [8]. As a concept of uncertainty, entropy in information theory is a natural measurement of trust as well [9]. In the extreme case, trust can be binary: trust (trust weight=1) or distrust (trust weight=0) because either there is 100% security in the network or the approach to evaluate trust is very coarse. There is no absolutely right or wrong for these representations. All the aforementioned numerical representations are suitable for different environments and management requirements.

In the rest of the paper, we incorporate trust weights into a particular application of information fusion in sensor networks: the distributed estimation problem. We show that the algorithm with trust "avoids" using information from sensors that are not trusted.

4 Distributed Kalman filtering

Distributed estimation and tracking are two of the most fundamental collaborative information processing problems in wireless sensor networks. The main idea behind distributed estimation, found in most of the papers addressing this problem, consists of using a standard Kalman filter locally, together with a consensus step in order to ensure that the local estimates agree [1, 2, 3]. In what follows, we use a simplified version of the algorithm proposed in [1].

For simplicity we omitted the time index in Algorithm 1. Notice that with the exception of line 4, the above algorithm is the standard linear Kalman filter. In line 4, the local information is linearly combined with information received from

Algorithm 1: Distributed Kalman Filtering algorithm with consensus step on estimates [1]

Input: μ_0, P_0

1 Initialization: $\xi_i = \mu_0$, $P_i = P_0$

2 **while** new data exists

3 Compute the intermediate Kalman estimate of the target state:

$$M_i = P_i^{-1} + C_i' R_i^{-1} C_i$$
$$L_i = M_i C_i R_i^{-1}$$
$$\varphi_i = \xi_i + L_i(y_i - C_i \xi_i)$$

4 Estimate the state after a consensus step:

$$\hat{x}_i = \varphi_i + \epsilon \sum_{N_i \cup \{i\}} (\varphi_j - \varphi_i)$$

5 Update the state of the local Kalman filter:

$$P_i = A M_i A' + Q$$
$$\xi_i = A \hat{x}_i$$

neighbors. We will refer to line 4 as either the *information fusion step* or the *consensus step*. We will focus our analysis on the values of the weights w_{ij} . In fact they will play the role of the confidence values introduced in the previous section. Unlike the original algorithm [1], we assume that only local estimates are exchanged and not output measurements as well.

5 Distributed Kalman filtering with trust dependent weights in the consensus step

In this section we develop the distributed filtering equations that take into account the confidence (trust) of the sensors. We address two cases with respect to what the confidence values represent. In the first case, we assume that the weights w_{ij} are a measure of the *information accuracy*, i.e. the larger the value of w_{ij} is, the more accurate the information received by i from j is. In the second case, the weights w_{ij} are a measure of the *trustworthiness* of the data received by sensor i from sensor j. It may be the case that either a sensor or a link were compromised, so that the information received from the respective sensor or through the respective link is not trustworthy.

5.1 Distributed Kalman Filtering with accuracy dependent consensus step

We attach to each sensor a trust value. In this subsection, the trust refers to the accuracy of information. The larger the trust value is, the more accurate the information received from the respective sensor is. The information exchanged between sensors is represented by estimates. As previously mentioned, we denote by T_{ij} the trust sensor i has in information received from sensor j. We propose to

choose the trust values to be inversely proportional to the estimation error, according to the formula:

$$T_{ij} = \frac{1}{trace(M_j)},\ j \in \mathcal{N}_i, \tag{3}$$

where M_j represents the covariance matrix of the estimation error from the standard Kalman filter step. The properties of this matrix will be affected by how *observable* the state is from sensor j, (such as the rank of matrix C_j) and how noisy the measurements are, i.e. the variance of the measurements noise R_j. We can expect that the variance of the estimation error, given by the trace of M_j, to be small for highly observable measurements with low noise. Therefore, we computed the weight values in the information fusion step, by normalizing the trust values T_{ij}:

$$w_{ij} = \frac{T_{ij}}{\sum_k T_{ik}}. \tag{4}$$

Algorithm 2: Distributed Kalman Filtering Algorithm with accuracy dependent consensus step on estimates

Input: μ_0, P_0

1 Initialization: $\xi_i = \mu_0$, $P_i = P_0$

2 **while** new data exists

3 Compute the intermediate Kalman estimate of the target state:

$$M_i = P_i^{-1} + C_i' R_i^{-1} C_i$$
$$L_i = M_i C_i R_i^{-1}$$
$$\varphi_i = \xi_i + L_i(y_i - C_i \xi_i)$$

4 Compute the consensus weight values:

$$T_{ij} = \frac{1}{trace(M_j)}$$

$$w_{ij} = \frac{\bar{w}_{ij}}{\sum_k \bar{w}_{ik}}$$

5 Estimate the state after a consensus step:

$$\hat{x}_i = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \varphi_j$$

6 Update the state of the local Kalman filter:

$$P_i = A M_i A' + Q$$
$$\xi_i = A \hat{x}_i$$

This way, we assign a larger influence to the more accurate estimates, directing the resulting average towards estimates with high accuracy. Note however that the matrix M_j is not the actual covariance matrix of the estimation error for the current estimate $\hat{x}_j$, but the covariance error given by the standard Kalman filter. Int does however reflect the observability properties of the sensor, making it a good candidate for constructing the weight values. We summarize the idea introduced above in Algorithm 2.

5.2 Distributed estimation with reliability dependent consensus step

In this subsection we propose a distributed estimation scheme where the averaging operation depends on the reliability of the sensors. We assume that sensors may be compromised and may send data aimed at modifying the result of the estimation process. The update mechanism for the trust values T_{ij} is based on the notion of *belief divergence* [4]:

$$d_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \|\hat{x}_i - \hat{x}_j\|^2, \tag{5}$$

where we denoted by $\hat{x}_i$ the current estimates.

The belief divergence d_i, gives to sensor i a measure of how different its own estimate is with respect to the estimates of the other sensors within its communication neighborhood.

Since the sensors exchange only state estimates, every sensor will compute a belief divergence, d_{ij}, for each sensor in his neighborhood, according to the formula:

$$d_{ij} = \frac{1}{\mathcal{N}_i - 1} \sum_{k \in \mathcal{N}_i} \|\hat{x}_j - \hat{x}_k\|^2. \tag{6}$$

This metric shows how far a received estimate is from the other received estimates in some neighborhood. Note that in the fusion step, estimates far from their real values are prone to hurt more. However, if enough neighbors provide reliable information, the belief divergence for a sensor sending false information is going to by high. We use the locally computed belief divergence metric, to update the trust values T_{ij}. We first choose a positive constant c_i, satisfying:

$$c_i > \max\{d_{ij} \mid j \in \mathcal{N}_i\}.$$

We use the constant c_i in the following formula for updating the trust values:

$$T_{ij} = c_i - d_j,\ j \in \mathcal{N}_i \tag{7}$$

Notice that the parameters c_i were chosen so that the trust value T_{ij} are nonnegative. Moreover, c_i are discriminating in the sense that they influence the ratios T_{ij}/T_{ik}. Typically, the smaller c_i, the more sensors with large values of the belief divergence are penalized. From (7) we notice that we favor the sensor whose estimate is close to the other estimates in his neighborhood, in a sense 'accelerating convergence' to consensus. We denote by p_{ij} the normalized versions of the trust values T_{ij}, computed according to the formula:

$$p_{ij} = \frac{T_{ij}}{\sum_{k \in \mathcal{N}_i} T_{ik}}, \tag{8}$$

which may be interpreted as the "probability the data received by sensor i from j are accurate". Note from the above formulas that, although small, the normalized trust values are not necessarily zero for sensors with large belief divergence. Therefore if the value of a false estimate is large compared with the others, it will still influence negatively the

information fusion step. That is why we introduce a thresholding scheme on the normalized trust values. Let p_i^{min} be the minimum value accepted for p_{ij}. If $p_{ij} < p_i^{min}$ the trust value T_{ij} will be set to zero, hence filtering out information that is not considered sufficiently trustworthy. The lower bound p_i^{min} should be chosen to be inversely proportional to the size (cardinality) of the neighborhood.

Algorithm 3: Distributed Kalman Filtering Algorithm with a reliability dependent consensus step on estimates

Input: μ_0, P_0

1 Initialization: $\xi_i = \mu_0$, $P_i = P_0$

2 **while** new data exists

3 Compute the intermediate Kalman estimate of the target state:

$$M_i = P_i^{-1} + C_i' R_i^{-1} C_i$$
$$L_i = M_i C_i R_i^{-1}$$
$$\varphi_i = \xi_i + L_i(y_i - C_i \xi_i)$$

4 Compute locally the belief divergence:

$$d_{ij} = \frac{1}{\mathcal{N}_i - 1} \sum_{k \in \mathcal{N}_i} \|\varphi_j - \varphi_k\|^2$$

5 Compute the trust values:

$$T_{ij} = c_i - \bar{d}_{ij},\ j \in \mathcal{N}_i$$

6 Compute the normalized trust values:

$$p_{ij} = \frac{T_{ij}}{\sum_k T_{ik}}$$

7 Eliminate insufficiently accurate data by setting T_{ij} to zero if $p_{ij} < p_i^{min}$

8 Compute the consensus weight values:

$$w_{ij} = \frac{T_{ij}}{\sum_k T_{ik}}$$

9 Estimate the state after a consensus step:

$$\hat{x}_i = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \varphi_j$$

10 Update the state of the local Kalman filter:

$$P_i = A M_i A' + Q$$
$$\xi_i = A \hat{x}_i$$

The updated trust values are next used to compute the weights in the consensus step:

$$w_{ij} = \frac{T_{ij}}{\sum_{k \in \mathcal{N}_i} T_{ik}}, \tag{9}$$

The distributed estimation algorithm with a reliability dependent averaging scheme is presented in Algorithm 3. The intuition behind our proposed algorithm is that if a node j sends false data, the other nodes will compute large belief divergence values, and hence low trust values, which together with the thresholding scheme will eliminate the node from the information flow. The consensus step has the role of producing a new state estimate by averaging the estimates on neighborhoods. If an estimate is not accurate enough, it may drag the updated estimate towards the wrong direction. By computing the consensus weight values using a trust dependent mechanism, we try to minimize the possibility of an estimate update moving in the wrong direction. By adjusting the minimum accepted value for the normalized trust values, p_i^{min} , the sensors can control their sensibility with respect to the received data.

6 Simulations

We consider a perturbed oscillatory linear system:

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

where,

$$A = \begin{pmatrix} 0.9996 & -0.03 \\ 0.03 & 0.9996 \end{pmatrix}$$

and $w(k) \in \mathbb{R}^2$ is a white, Gaussian noise, with covariance matrix $Q = 0.15 I_2$. Each sensor has a sensing model of the form:

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

where the observation matrices C_i are chosen at random to be $[0, 1]$ or $[1, 0]$ with the same probability. The measurement noise $v_i(k) \in \mathbb{R}$ is assumed white and Gaussian with variance $R_i = \sigma_v \sqrt{i}$ and $\sigma_v = 30$. We consider a network with seven sensors. Six sensors have each three neighbors. The seven sensor, communicate with all others.

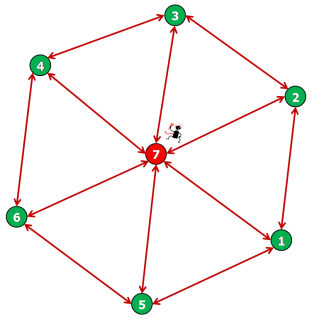

Figure 2: Sensor network

We first test Algorithm 2 against Algorithm 1. For computing the weights w_{ij} in Algorithm 1 we used the original scheme proposed in [1], the value for ϵ being chosen such that the average estimation error per node was as small as posssible. More precisely we want to compare the average estimation errors per node, given by the two algorithms. Since the trust weights are computed such that more weight is given to information coming from sensors with smaller variance of the estimation error, we would expect Algorithm

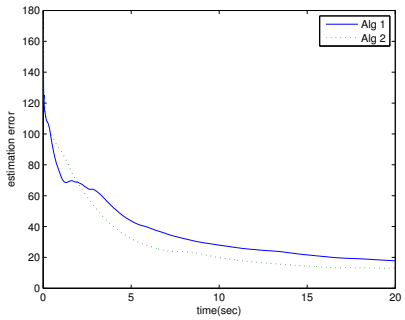

Figure 3: Comparison of estimation error given by Alg 1 and Alg 2 respectively

2 to perform better, in the sense that the average estimation error per node should converge to a smaller value..

We observe form Figure 3 that Algorithm 2, as expected, performs better. This is mainly due to the fact that in the estimation fusion step, we move the updates estimate closer to the local estimate with better observability and lower measurement noise.

For testing Algorithm 3, we assume that sensor 7 was compromised and sends false information to all the other sensors. The goal of sensor 7 is to shift the estimates of other nodes away from their true values. We consider first the case when sensor 7 sends to its neighbors a constant value as estimate, $\varphi_7 = -2\xi_0$. The centered sensor has to potential to do a lot of damage since it is connected to all other sensors. We first test this scenario using Algorithm 1.

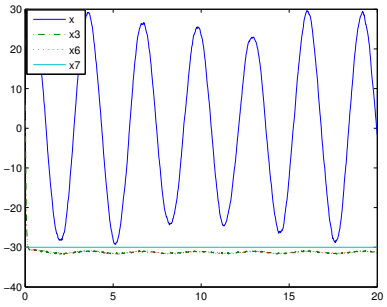

Figure 4: Distributed Kalman filtering with constant false information

In Figure 4 we plot the first entry of the state vector and the estimates of sensor 3, 6 and 7. The solid line depicts the trajectory of the first entry of the state vector. Notice how the estimates of the other sensors are driven away from the state values, due to the malevolent influence of sensor 7. We repeat the simulations, when the node 7 sends to its neighbors sinusoidal values as state estimates values, $\varphi_7 = [-20\sin(2\pi T(k) + 0.2); 10\sin(2\pi T(k))]$.

As in the previous case, the false data infused in the network, shifts the other estimates away from their correct values.

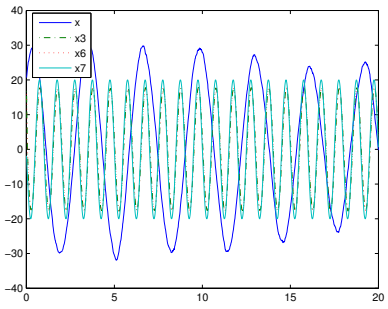

Figure 5: Distributed Kalman filtering with sinusoid constant false information

We used Algorithm 3 to test if sensor 7 is detected and eliminated. If this is the case, the other estimates should follow closely the state vector.

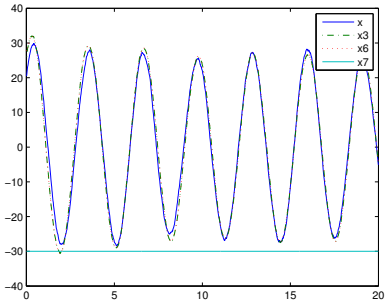

Figure 6: Distributed Kalman filtering with trust dependent consensus step and constant false information

Although is seems that the false data does have an influence on how fast the estmates track the state in the beginning, since the false data is not immediately detected and rejected, the sensors are able to compute state estimates that are close to the state values (Figures 6 and 7).

7 Conclusions

As presented in this article, data fusion involves the integration and application of many disciplines, including communication and decision theory, epistemology and uncertainty management, estimation theory, digital signal processing, computer science, and artificial intelligence. In this paper, we proposed two modified Distributed Kalman Filtering algorithms, which incorporate the notion of trust. Two operational interpretations of trust were used: accuracy and reliability, respectively. When using the first interpretation, we proposed un update scheme for the trust values based on the estimation error computed by the standard Kalman filtering; a metric which incorporates a qualitative description

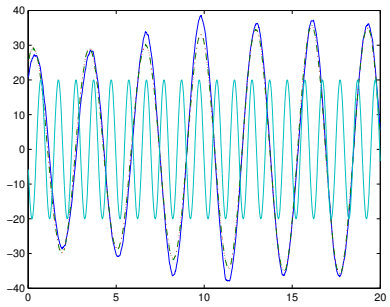

Figure 7: Distributed Kalman filtering with trust dependent consensus step and sinusoid false information

of a sensor in terms of observability and measurement noise. When we interpreted trust in terms of reliability, we used the belief divergence metric and a thresholding scheme to compute the trust values. In both cases, we used the normalized trust values as weights in the information fusion step, the resulting updated estimates leaning towards estimates with high trust values. Finally we tested our proposed algorithms via simulations.

Acknowledgement

This research is supported by award MURI W911-NF-0710287 from the Army Research Office. It is also prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011.

References

[1] R. Olfati-Saber, "Distributed Kalman Filtering for Sensor Networks", *Proceedings of the* 46^{th} *IEEE Conference on Decision and Control*, pages 5492-5498, 2007

[2] A. Speranzon, C. Fischione, K. H. Johansson and A. Sangiovanni-Vincentelli, "A Distributed Minimum Variance Estimator for Sensor Networks", *IEEE Journal on Select Areas in Communication*, vol. 26, no. 4, pages 609-621, May 2008.

[3] R. Carli, A. Chiuso, L. Schenato, Member and S. Zampieri, "Distributed Kalman Filtering Based on Consensus Strategies", *IEEE Journal on selected area in communication*, vol. 26, no. 4, pages 622-633, May 2008.

[4] C. De Kerchove and P. Van Doren, "Iterative filtering for a dynamical reputation system", arXiv, 2007.

[5] U. Maurer, "Modelling a Public-Key Infrastructure", *Proceedings of 1996 European Symposium on Research in Computer Security – ESORICS'96*, 1996, pages 325-350.

[6] G. Theodorakopoulos and J.S. Baras, "On Trust Models and Trust Evaluation Metrics for Ad-Hoc Networks", *IEEE Journal on Selected Areas in Communications, Security in Wireless Ad-Hoc Networks*, vol. 24, no. 2, Feb. 2006, pages 318-328.

[7] A. Josang, A. and R. Ismail, "The Beta Reputation System", *Proceedings of the 15th Bled Conference on Electronic Commerce*,Bled, Slovenia, 2002.

[8] S.D. Kamvar, M.T. Schlosser and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks", *Proceedings of the 12th Initernational World Wid Web Conference*, 2003, pages, 640-651, Budapest, Hungary.

[9] Y.L. Sun, Z. Han and W. Yu and K.J.Ray Liu, "A Trust Evaluation Framework in Distributed Networks: Vulnerability Analysis and Defense Against Attacks", *INFOCOM 2006: Proceedings of 25th IEEE International Conference on Computer Communications*, April 2006, pages 1-13.

[10] B. Lampson, M. Abadi, M. Burrows and E. Wobber, "Authentication in Distributed Systems: Theory and Practice", *Proceedings of the 13th ACM Symposium on Operating Systems Principles*, Oct. 1991, pages 265-310.

[11] K.B. Pratik and H. Sajid, "Special issue on information fusion in distributed sensor networks", *Information Fusion*, vol. 9, no. 3, pages 330-331, 2008.

[12] A. Jsang, "A Logic for Uncertain Probabilitie", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. 3, pages 279-311, June, 2001.

[13] T. Jiang and J. Baras, "Trust Evaluation in Anarchy: A Case Study on Automomous Networks", *Proceedings of the 25th IEEE Conference on Computer Communications (Infocom06)*, Barcelona, Spain, April 25-27, 2006.

[14] T. Jiang and J. S. Baras, "Trust Credential Distribution in Autonomic Networks", *Proceedings of the IEEE Global Communications Conference (IEEE Globecom 2008)*, pp. 1-5, New Orleans, LA, November 30 - December 4, 2008.

[15] V. Buskens, "Social networks and Trust", Kluwer, 2002.

[16] J.S. Baras and T. Jiang, "Cooperative, Trust and Games in Wireless Netwroks", in *Advances in Control, Communication Networks and Transportation Systems: in Honor of Pravin Varaya*, E.H. Abed (Edt.), Systems and Control: Foundations and Applications Series, pp. 183-202, Birkhauser, Boston, 2005.

[17] J.S. Baras, T. Jiang and P. Purkayastha, "Constrained Coalitional Games and Networks of Autonomous Agents", Invited paper, Proceedings of Third International Symposium on Communications, Control and Signal Processing, pp. 972-979, St. Julian, Malta, march 2008.

[18] M.Chiang, S.H. Low, A.R. Calderbank and J,C. Doyle, "Layering as Optimization Decomposition: A Mathematical Theory of Network Arhitectures", *Proceedings of the IEEE*, vol. 95, no. 1, pp. 255-312, January 2007.