

A Comparative Study of Different Kalman Filtering Methods in Multi Sensor Data Fusion

Mohammad Sadegh Mohebbi Nazar

Abstract- In this paper two different techniques of Kalman Filtering and their application in multi sensor data fusion are studied. Two different Kalman filtering techniques such as CKF and DKF are modeled and the effectiveness of their application in multi sensor data fusion is compared with each other. All of the kalman filter models have been simulated in MATLAB and simulation environment.

Index Terms—Kalman filtering - Data fusion - Multi sensor data fusion - DKF - CKF

I. INTRODUCTION

Multisensor data fusion is an evolving technology, concerning the problem of how to fuse data from multiple sensors in order to make a more accurate estimation of the environment [1, 2, 3]. Applications of data fusion cross a wide spectrum, including environment monitoring, automatic target detection and tracking, battlefield surveillance, remote sensing, global awareness, etc. They are usually time-critical, cover a large geographical area, and require reliable delivery of accurate information for their completion.

Sensor fusion is the combining of sensory data or data derived from sensory data from disparate sources such that the resulting information is in some sense better than would

II. KALMAN FILTER

The Kalman filter, explained in detail in [4], is an optimal linear estimator based on an iterative and recursive process. It is used in a wide variety of applications, and applies particularly well to sensor fusion. It recursively evaluates an optimal estimate of the state of a linear system. In our case, the state of the system is the pose of the vehicle. At each iteration of the filter, a new estimate of the state (pose) is evaluated, using the new information (measurements) available to the filter.

Mohammad Sadegh Mohebbi Nazar is with Iranian Offshore Oil Company (I.O.O.C), Tel: +98918150898, Fax: +988118275210 email: M.S.Mohebinazar@gmail.com

be possible when these sources were used individually. The term better in that case can mean more accurate, more complete, or more dependable, or refer to the result of an emerging view, such as stereoscopic vision (calculation of depth information by combining two-dimensional images from two cameras at slightly different viewpoints).

The data sources for a fusion process are not specified to originate from identical sensors. One can distinguish direct fusion, indirect fusion and fusion of the outputs of the former two. Direct fusion is the fusion of sensor data from a set of heterogeneous or homogeneous sensors, soft sensors, and history values of sensor data, while indirect fusion uses information sources like a priori knowledge about the environment and human input.

Sensor fusion is a term that covers a number of methods and algorithms, including: Kalman filter, Bayesian networks and Dempster-Shafer. In this paper we only utilize Kalman filters for multi sensor fusion.

The paper is organized as follows. Section 2 describes the Kalman filter, which is the most widely used estimator in sensor fusion. However, the Kalman filter also suffers from certain shortcomings. In section 3 Kalman filters fusion and performance is tested, followed in Section 4 by comparative results between the two mentioned kalman filter approaches. Finally the general results presented in the conclusion.

The Kalman filter process consists of two sub-processes, repeated iteratively at a time step of dt : the time update and the measurement update. An easy way to explain the Kalman filter is to say that it incorporates two sets of sensor measurements, one in each sub-process. In the time update process, a prior estimate $X_{prior}(k)$ is computed based on the previous state estimate $X(k-1)$ and sensors indirectly related to the state (e.g. accelerometers or velocity sensors when the state is position, also called dead-reckoning sensors). Then, in the measurement update process, this prior estimate is blended with direct measurements of the state (position) coming from other sensors, thus obtaining the new updated state estimate $X(k)$.

First of all, we will assume a mathematical model of a plant defined by equations of discrete system dynamics. To get the equations of the optimum estimator, i.e., the KF,

suppose that the plant of system dynamics are designed by the (possibly time-varying) general model of linear finite-dimensional stochastic system, see below; [5], [6].

$$X(n+1) = Ax(n) + Bw(n) \quad (1)$$

A control input $u(n)$ of plant is included in process noise $w(n)$, where n refers to discrete time. The $w(n)$ noise is not necessary white but it should be a zero mean noise.

$$y(n) = Cx(n) + v(n), \quad v = Cx + 0 \quad n \geq n_0 \quad (2)$$

The $n_0 = 0$ is the initial time. The equation (1) is called stochastic state transition or system model equation, and (2) is called the observation equation of stochastic model, [7]. In the terminology, A is state transition matrix, B is input coupling vector, C is observation vector, x is called state vector and $yv(n)$ is plant observation and finally $v(n)$ is called sensor or observation noise. The $x(n_0)$ has a mean x_0 and state covariance P_0 matrix and

$$\text{Where } w \sim N(0, Q(n)), \quad (4)$$

$$\text{and } v \sim N(0, R(n)). \quad (5)$$

Here we denote equation (4) for (1), and (5) for (2). The Q is called process noise covariance matrix and R is observation noise covariance matrix.

The relation of $w(n)$, $v(n)$ and $Q(n)$, $R(n)$ is respectively defined by (4), (5). Also we assume the ideal non-noisy observations

$$y(n) = Cx(n). \quad (6)$$

The discrete plant model was already described by the applied process noise $w(n)$. Observations are given by equation (2). Although the observation noise $v(n)$ is generated by sensor but not by a plant. The equations (1) and (2) both describe state-space model. Below we mention some conditions been valid for a KF such as an optimal state estimator and its equations, [5] and [6].

1. The sampled white noise has a mean of zero:

$$E[w(n)] = 0; \quad E[v(n)] = 0.$$

2. The $w(n)$ and $v(n)$ are uncorrelated for $n \neq k$, i.e.:

$$S = E[v(n) w^T(k)] = 0, \text{ for } n \neq k, \text{ where } k \text{ refers to a discrete time.}$$

3. Noise variances are

$$E[w(n) w^T(k)] = \begin{cases} Q(n), & \text{for } n = k \\ 0, & \text{for } n \neq k \end{cases}$$

$$E[v(n) v^T(k)] = \begin{cases} R(n), & \text{for } n = k \\ 0, & \text{for } n \neq k \end{cases}$$

4. $E\{[x(n) - x(n|n)]v^T(n)\} = 0$, where $x(n)$ is state vector of plant, $x(n|n)$ is estimated state.

5. State error covariance matrix $P(n)$, innovation Kalman filter gain $M(n)$, A , C , R , Q and $S = 0$ are

independent of observations sequence $Y(n-1) = \{y_v(n-1), y_v(n-2), \dots, y_v(0)\}$.

6. We shall specify new symbol, namely $r(n)$ to the error of $y_v(n) - Cx(n|n-1)$ the innovations sequence (residuals), where $E\{r(n)|Y(n-1)\} = 0$ and $x(n|n-1)$ is the state time update in Kalman filter. Nonzero mean of a noise is not our case of study and tests. In Kalman filtering we consider $Q(n) < x(n)x^T(n)$ and $R(n) < y_v(n)y_v^T(n)$. We assume that the system output can be predicted and a white noise of innovations sequence-residuals $r(n)$, Q and R are correctly estimated. Unfortunately, this is not the reality. Supposing those assumptions, the optimality of state estimation is achieved when an algebraic constraint on derived equations is used in Model of Kalman filter below. Solving the estimation problem, KF minimizes the state error covariance matrix in optimal linear filtering. The innovations sequence named by $r(n)$ is useful for the reason of state estimation in Kalman filtering.

A. CENTRALIZED KALMAN FILTER

In centralized Kalman filtering, signals of sensors are transferred through the communication network to the central processor to generate the optimal central estimate $x(n|n)$. The all information is sent to the fusion centre to yield $x(n|n)$ and minimize state estimation error. The models are built according to [8].

All sensors are measuring outputs of plant. Of course, all the plant is controlled by one actuator which gives $u(n)$ value the same for CKF estimator input. For example the $y_{v1}(n)$ may represent the first CCD camera observation, $y_{v2}(n)$ represents the microphone observation and $y_{v3}(n)$ represents the mechanical sensor of a track and position estimation of a train in a tunnel, [9]. Observations $y_{vi}(n)$ are modeled as filtered $x(n)$ through observation vectors C_i , where $x(n)$ is only one. Subscript $i = 1, 2, 3$ refers to the particular sensors.

Model of centralized Kalman filter. A timing diagram of this model is shown in Table 1 as a special case of Model of Kalman filter with sensor fusion. In other words, signals are combined to get state estimation in fusion. Initial program starts at n_0 and prepares (7) – (9).

The concept of CKF technique is analyzed in papers [8], [9] and [10].

B. DECENTRALIZED KALMAN FILTER

The decentralized Kalman filter processes data from many sensors to provide a global state estimation in multi-sensor fusion. A DKF model was built according to references [9], [11] - [13].

Every DKF contains a local and a global filter that emphasizes double-estimation in a node. The local filter uses its own data and observation $y_v(n)$ to perform an optimal local estimates $P(n|n)$ and $x(n|n)$.

These estimates are obtained in a parallel processing mode implicitly. Thus, each node takes

observation (possibly asynchronously) from a plant of an environment. With this observation (and its associated variance) the DKF is able to compute a partial state estimate. Then each node broadcasts one vector and a matrix of error information to the others and it receives other information being broadcasted to it. Those two (one vector and a matrix) as state error information SEI(n) and variance error information VEI(n) are used for global state and covariance estimate in global filter of every node. Finally, all nodes performed same global state estimates $x(n|n)$ because SEI(n) and VEI(n) matrices are fused in the same way.

Generally, the number of sensors is not restricted. We will describe the functionality of DKF system by the following flowchart in Table 2.

The local filter of the i -th node computes its own local estimate updates (6) and (7) using residuals (5) and an innovation KF gain (4), respectively. The time updates (8) and (9) are also performed by a local filter of each i -th DKF node. The local filter of the i -th DKF node works independently and indirectly from the other nodes and all their residuals differ. After the local filter was processed, the SEI(n) of (11) and VEI(n) of (10) are computed by assimilation equations. So, every nodal DKF computes its own part such as the i -th part of either (10) or (11) that is called fractional matrix. These fractional matrices are sent both to other nodes where are collected to get global SEI(n) and VEI(n). The fractional matrices are meant to be expressed such as $x_i(n|n)P_i(n|n-1) - x_i(n|n-1)P_i(n|n-1)-1$ for SEI(n), and $P_i(n|n-1) - P_i(n|n-1)-1$ for VEI(n), where the subscript i denotes the i -th node and filter. The global filter of the i -th node computes global estimates by received SEI(n) of (12) and VEI(n) of (13). There the time updates of global state and covariance matrix are computed in (16) and (15), respectively.

In each processor (node), a feedback process is running, where the global filter sends a global updated estimates $x(n|n)$ and $P(n|n)$ covariance matrix into its own local filter. This way, the $x(n|n)$ and $P(n|n)$ are interchanged with $x_i(n|n)$ and $P_i(n|n)$, respectively. It refers to (17) and (18). Finally, the state estimate updates are evaluated in the same manner in all nodes. This fact tends to DKF data fusion, where the states are computed optimally. The global output can be obtained in (19).

Each DKF estimator can be embedded in a sensor. Besides, in CKF structure all observations y_{vi} need to be broadcasted to the corresponding i -th nodes. The advantage of DKF against CKF is an embedded processor of DKF in sensor; hence no central fusion is needed, see [12]. In DKF node the observations are used directly. In sense of estimation, the advantage of DKF is also the less sensitive estimator to corrupted SEI(n) and VEI(n) when corrupted broadcasting happens. In other words, each sensor node has its own processing element, and its own communication facilities.

The DKF algorithm has a number of important features [11]:

- Global estimates by all nodes are guaranteed to be exactly the same as those obtained by CKF.
- A failure of any broadcasted signal tends to estimate degradation. But it will not result a whole system failure. This fact works in opposite to centralized fusion.
- A small additional computation required.
- Low communication overhead than similar structures.

Each node must communicate one SEI(n) vector and one matrix VEI(n) to each other node. Assuming there are N_0 sensing nodes and each node estimates a full state vector of dimension l , then a total of $(l^2 + l)(N_0 - 1)$ numbers need to be communicated in each cycle. In CKF systems (central fusion) only N_0 numbers are equal to the number of sensors need to be broadcasted, [9].

III. TESTING THE KALMAN FILTERS FUSION AND PERFORMANCE

In this section mentioned Kalman filters have been tested in data fusion.

A. 3.1 Testing the CKF fusion and performance

The CKF model is built according to equations of Table 1. In Figure 1, the filtered outputs of CKFs and CoKF are overlapped without any difference. In other words, the outputs of CKF and CoKF are the same quality and mathematically equivalent in this test.

The Figure 2 shows the priori error variance of filtered outputs computed by $C P(n|n-1) CT$. Finally, all curves tend asymptotically to constant. On the basis of comparability with all models, the covariance time update performs quite the same. The singularity problem has no longer effect on state estimation performance. In the case of CoKF estimator, the MSE is shown in Table 1 in second row and the relative MSE in last one. The power of state $x(n)$ is equal to 1.076.

IV. CENTRALIZED AND DECENTRALIZED KALMAN FILTER COMPARISON

In this section we will dedicate the effort to explore identity between decentralized, DKF, and centralized, CKF, Kalman filter which can be measured by mean square error MSE in state estimates in an experimental simulation study. The MSE values of CKF and DKF models will be compared although both models are derived mathematically see [11]. Both models are created according to Table 1 and 2 as a flow of control, and articles [9], [11-13].

Table 1 Timing diagram of Model of CKF.

Initial Program	
Initial Time n = 0	
Initial error covariance:	$P(n n - 1) = B Q d B^T$, $Q d$ is defaulted $Q(0) > 0$ (7)
Initial condition on the state:	$x(n n - 1) = 0$ (8)
Initial condition on Filter Output:	$y_{ei}(n) = 0$ (9)
Iteration Time n = 1,2,3,...	
<i>Observation update :</i>	
Error Covariance Update:	$P(n n) = \left[I - \frac{1}{N_0} \sum_{i=1}^{N_0} M_i(n) C_i(n) \right] P(n n - 1)$ (10)
State Estimate Update:	$x(n n) = x(n n - 1) + \frac{1}{N_0} \sum_{i=1}^{N_0} M_i(n) r_i(n), N_0 \geq 1$ (11)
Estimated Filter Output:	$y_{ei}(n) = C_i x(n n)$ (12)
Error Covariance:	$\text{error cov}_i = C_i P(n n) C_i^T$ (13)
<i>Time update :</i>	
State Time Update:	$x(n+1 n) = Ax(n n) + Bu(n)$ (14)
Error Covariance Time Update:	$P(n+1 n) = AP(n n)A^T + BQ(n)B^T$ (15)

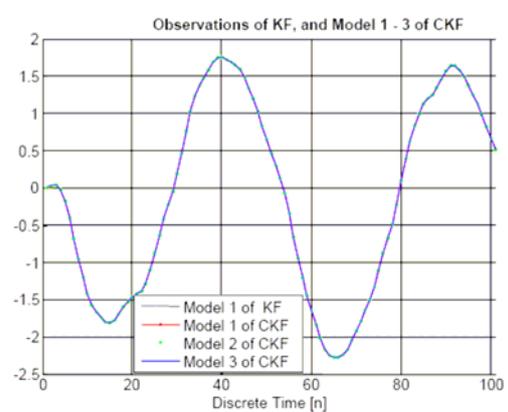


Figure 1 The CKF and KF observations of first output.

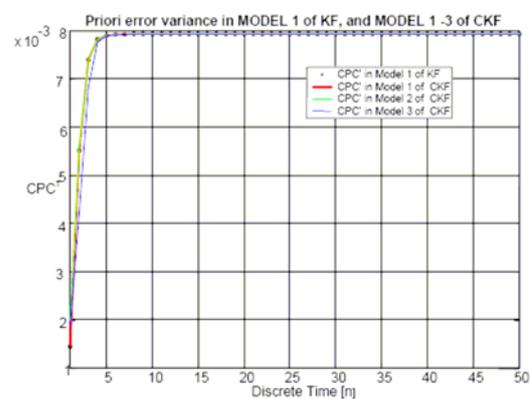


Figure 2 Prior error variance $CP(n|n-1)C_T$ of filtered outputs in CKF and CoKF.

Table 2 Timing diagram of DKF.

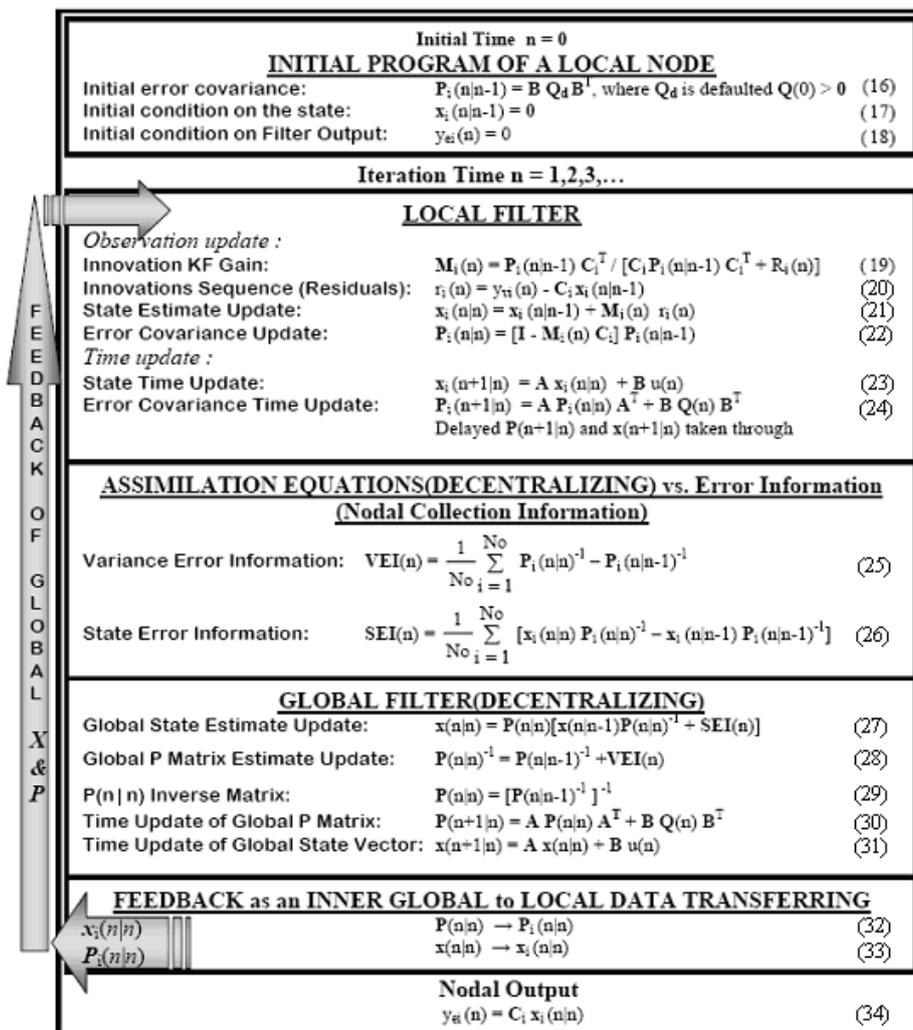


Table 3 MSE of DKF; MSE of CKF; MSE of estimated state x(n|n) (DKF-CKF) comparison.

$(MSE \text{ of Model of CKF}) / (\bar{x}^2(n) \text{ of Plant})$	$3.420 \cdot 10^{-4}$
$(MSE \text{ of DKF}) / (\bar{x}^2(n) \text{ of Plant})$	$3.416 \cdot 10^{-4}$
$(DKF \bar{x}(n n) - Model \text{ of CKF } \bar{x}(n n))^2 / (\bar{x}^2(n) \text{ of Plant})$	$3.429 \cdot 10^{-7}$

Table 4 State vectors of Model of CKF, DKF and plant at n = 2

$x(n n)$ of Model of CKF	[-1.457; -0.719; -0.061]
$x(n n)$ of DKF	[-1.457; -0.719; -0.061]
$x(n)$ of Plant	[-1.471; -0.729; -0.057]

Table 5 DKF verification toward to CKF result based on stated measurement.

$\sum [(DKF1 \bar{x}(n n) - CKF \bar{x}(n n))^2] / (\bar{x}^2(n) \text{ of Plant})$	$3.429 \cdot 10^{-7}$
$\sum [(DKF2 \bar{x}(n n) - CKF \bar{x}(n n))^2] / (\bar{x}^2(n) \text{ of Plant})$	$3.429 \cdot 10^{-7}$

Additionally, a relative MSE of a difference between DKF and CKF state estimates $x(n|n)$ is shown at last row of the following table. The value in first row is related to difference between $x(n)$ and $x(n|n)$ of CKF state estimates. The second item explains a relative MSE of DKF computed in such way as the previous MSE in the first row.

Both values are very similar to satisfy the numerical identity among CKF and DKF. The error comparison is not crucial in performance comparison of both estimators that is still to come. To make relevant comparison on estimators is to compute a difference between state vectors $x(n|n)$ of DKF and CKF, see the last value. This relative MSE is about 103 times lower than the first two values. Actually, this value is expected to be small, because the DKF and CKF provide mathematically equal states. As an example, the Table 4 shows some sampled values of state vectors at $n = 2$.

One can see no numerical difference in state elements of both estimators due to low precision. The last item shows the $x(n)$ of plant that is on numerical comparability with $x(n|n)$ of CKF and DKF.

In our simulation, two sensors and two nodes of DKF estimator called DKF1 and DKF2 are shown. As shown in the last item of Table 1, the relative MSE is measured in the case of DKF1 and DKF2 and shown below in Table 5.

So, the DKF and CKF operate well despite very small difference measured in state estimates.

V. CONCLUSIONS

We have measured the relative MSE about 10-2 in state estimates of CKF and DKF. Measured MSE of $x(n|n)$ difference between DKF1, DKF2 and CKF is about 103 smaller than the value mentioned firstly. These facts can be deeply clarified via in Table 1 and 5. In the practical point, we can say that both estimators perform on the same state estimation despite the unimportant difference in our MATLAB simulations. This fact is caused by a computational problem of covariance matrix singularity that can be overcome using the improvements. Said the improvements are needed however causing a little inconvenience as a numerical problem. Without the improvements either the DKF or CKF can not start up, that usually gives a singularity problem. This is certain constraint. Toward to the power of $u(n)$ and $x(n|n)$, the DKF is designed so that this error 3.429×10^{-7} is negligible. In our simulations all signals are broadcasted without any unknown latency or transmission failure in multi-sensor fusion.

REFERENCES

- [1] D. N. Jayasimha, S. S. Iyengar, and R. L. Kashyap. Information integration and synchronization in distributed sensor networks. *IEEE Trans. Syst., Man, Cybern.*, SMC-21(21):1032–1043, Sept./Oct. 1991.
- [2] A. Knoll and J. Meinkoehn. Data fusion using large multi-agent networks: an analysis of network structure and performance. In Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pages 113–120, Las Vegas, NV, Oct. 2-5 1994. IEEE.
- [3] L. Prasad, S. S. Iyengar, R. L. Kashyap, and R. N. Madan. Functional characterization of sensor integration in distributed sensor networks. *IEEE Trans. Syst., Man, Cybern.*, SMC-21, Sept./Oct. 1991.
- [4] MOHINDER S. GREWAL, and ANGUS P. ANDREWS, (1993). "KALMAN FILTERING Theory and Practice," PRENTICE-HALL, INC., A Simon & Schuster Company Englewood Cliffs, New Jersey 07632.
- [5] GOODWIN Clifford G., and SIN K. S. (1984). "Adaptive Filtering Prediction and Control," PRENTICE-HALL, INC., Englewood Cliffs, New Jersey 07632.
- [6] FRIEDLAND B. (1996). "Advanced Control System Design," PRENTICE-HALL, INC., A Simon & Schuster Company Englewood Cliffs, New Jersey 07632
- [7] RAO B. S., and DURRANT-WHYTE H. F., (September 1991). "Fully decentralized algorithm for multisensor Kalman filtering," IEE PROCEEDINGS-D, Vol. 138, No. 5, pp. 413-420.
- [8] WEI M., and SCHWARTZ K. P., (20-23 March 1990). "Testing a decentralized filter for GPS/INS integration," Position Location and Navigation Symposium, 1990. Record. 'The 1990's - A Decade of Excellence in the Navigation Sciences'. IEEE PLANS '90, IEEE , pp. 429 – 435.
- [9] STROBEL N., SPORS S. and RABENSTEIN R., (5-9 June 2000). "Joint audio-video object localization using a recursive multi-state multi-sensor estimator," Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference, Vol.6, pp.2397 - 2400 vol.4.
- [10] PAO L. Y., and BALTZ N. T. (June 1999). "Control of Sensor Information in Distributed Multisensor Systems," Proc. American Control Conference, San Diego, CA, pp. 2397-2401.
- [11] DURRANT-WHYTE H. F., and LEONARD J. J., (13-18 May 1990). "Toward a fully decentralized architecture for multi-sensor data fusion," Robotics and Automation, 1990.Proceedings., IEEE International Conference, Vol.2, pp. 1331 – 1336.
- [12] DURRANT-WHYTE H. F., RAO B. Y. S., and HU H., (4 Feb 1991). "Toward a fully decentralized architecture for multi-sensor data fusion," Principles and Applications of Data Fusion, IEE Colloquium, pp.2/1 - 2/4.
- [13] DURRANT-WHYTE H. F., (19 Feb 1991). "Elements of sensor fusion," Intelligent Control, IEE Colloquium, pp.5/1 - 5/2.