Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

SENSOR FUSION USING FUZZY INTEGRAL AND DIVERSE BAYESIAN NETWORKS

Jamie Walls, Albert Esterline, Abdollah Homaifar

Author Autonomous Control and Information Technology Center (ACIT), Electrical & Computer Engineering Department
North Carolina Agricultural & Technical State University, Greensboro, NC

Abstract: This paper investigates and contrasts the use of different Bayesian networks and a fuzzy integral for real-time sensor fusion using sonar and rangefinder laser values on an ActivMedia robot. Bayesian networks have become increasingly popular because of their ability to capitalize on the conditional probabilities present in an influence chain. The Choquet fuzzy integral, which has primarily been used for statistical analysis, has a great power of description. Comparison of the two methods indicates that noise within the sensor network can drastically affect the accuracy of the results, especially those obtained using the Bayesian network. *Copyright © 2008 IFAC*

Keywords: Information and sensor fusion, Autonomous robotic systems, Data-fusion, Multi sensor systems, Guidance navigation and control, Intelligent robotics.

1. INTRODUCTION

Optimizing data from multiple sensors is a problem that can be approached in many ways. This is especially true in a dynamic environment, where it is necessary to have good and reliable results while providing redundancy. There is no universal data-fusion technique that works for all sensor networks because diverse sensors and sensor networks have unique specifications and operating limitations that may yield different types of data. Thus, in general, sensor fusion algorithms must be specially designed for applications.

Difficulties from corrupt or seemingly unreliable data can often make formulation and design difficult and possibly require a computationally expensive algorithm. Therefore, it is important to design the algorithm around the given system constraints and with sufficient levels of fault detection. There are many unique methods for sensor fusion because different methods are needed to handle the different constraints and issues that commensurate and non-commensurate sensor networks pose (Klein, 1999).

Bayesian networks provide an increasingly popular method for information fusion. These networks not only use probability inference to combine different types of sensor data but also give the designer a better grasp of the relations between the different sensors and features. Fuzzy integrals provide a non-traditional optimization method that is based on a generalization of one of the most important concepts in analysis, measure. The integrals are based on non-additive measures, which capture not only positive but also negative relationships among the multiple inputs to a system. (Garbisch, et al., 2000).

2. SENSOR FUSION

Several problems may arise with physical sensor measurements: sensor deprivation, limited spatial coverage, limited temporal coverage, imprecision, and uncertainty. Sensor, information or data fusion allows compensation for the limitations of individual sensors. There are many approaches to information fusion (Elmenreich, 2004), and there are certain advantages that can be expected from the fusion of sensor information (Klein, 1999): robustness and reliability, extended spatial and temporal coverage, increased confidence, reduced ambiguity and uncertainty, robustness against interference, as well as improved resolution. Sensor fusion also has its limitations and constraints, such as processing time and data corruption, which have to be accounted for when designing any sensor fusion algorithm.

Several types of general sensor configurations are available for complementary and competitive sensor networks (Klein, 1999). In a complementary configuration, data from heterogeneous sensors is combined for a more complete set of data, while a competitive/redundant sensor configuration tries to combine only the "best" information from multiple homogeneous sensors to support an optimal decision (Borenstein et al., 1996).

2.1 Bayesian Network

Bayesian networks exploit the conditional independence present in an influence chain. This method of sensor fusion allows the designer to form a network in an intuitive manner instead of using strict constraints and formats. Its flexibility allows the designer to determine what structures to explore to produce a network that is both efficient and accurate (Neapolitan, 2004). A Bayesian network is represented as a DAG (Directed Acyclic Graph). Each node X_i has a conditional probability distribution, written schematically as $P(X_i|Y_1,\ldots,Y_n)$, where Y_j, $1 \le j \le n$, are the parents of X_i. Every entry in the joint probability distribution can be calculated as the product of the appropriate elements in the conditional probability tables in the Bayesian network. Conditional independence allows each node to have only a bounded number of parents and children. A parent can generally be seen as a cause of each of its children. Diagnostic reasoning is pursued in a child-to-parent direction with Bayes' rule. Unconditional probabilities are represented by the source nodes (nodes with no parents).

Bayesian networks support searching DAGs. This is done by establishing a basic structural model for the proposed network and defining a set of criteria to determine which DAG produces a highest score. This score indicates how close the probabilities the DAG gives are to the actual measured probabilities. Any DAG with the highest score is considered to be among the most suitable Bayesian network models for supporting decisions.

There are many algorithms that can be used to find a DAG that gives accurate results efficiently. The type of algorithm needed is determined by the basic structure and the number of relationships that need to be explored. Two of the most common techniques for finding a high-scoring DAG are the K2 search and DAG search algorithms (Neapolitan, 2004). Both allow the designer to identify influences that must or must not be present, and both use an established schema for learning the structure of a Bayesian network with n variables. K2 search imposes an upper bound on the number of parents any node may have, while DAG search does not explore combinations with different numbers of parents (Neapolitan, 2004). Our future use of Bayesian networks for sensor fusion will include the use of the K2 search algorithm because it considers a more diverse set of configurations. (Demircioglu and Osadciw, 2006) (Hall and Llinas, 2001).

2.2 Fuzzy Integral

A fuzzy integral is similar to a LeBegues integral but uses fuzzy measures. For a given *measure*, or *set function* (function whose domain is a set of sets), ξ and topological space X, if (1) holds for ξ and any subsets A, B of X, then ξ is said to be *monotonic*.

$$\xi(A) \le \xi(B), \quad A \subset B \tag{1}$$

An *additive measure* ξ has the property expressed in equation (2) for disjoint subsets A and B over universe X.

$$\xi(A) + \xi(B) = \xi(A \cup B) \tag{2}$$

A *signed measure* can assume either positive or negative values while an *unsigned measure* can assume only positive values. A measure is additive if and only if it is monotonic and unsigned. Traditional measures are additive. Fuzzy measures are in general monotonic but need not be strictly additive. Weaker conditions make these measures more complex but also give the fuzzy integral a greater power of description. This is because fuzzy measures allow an integral to capture positive and negative interactions between different inputs, while the unsigned quality of a classical measure only captures positive interactions.

There are many kinds of fuzzy measures; some of the more popular are λ–fuzzy measures (the most general), possibility and necessity measures, and t-conorm and decomposable measures (Garbisch, et al., 2000). The λ–fuzzy measure was developed by Sugeno to reduce complexity and is defined, for topological space X, as a normalized set function g_λ defined on 2^X (the set of all subsets of X) where, for every pair of disjoint subsets A and B of X, equation (3) is satisfied (Garbisch, et al., 2000).

$$g_\lambda(A \cup B) = g_\lambda(A) + g_\lambda(B) + \lambda g_\lambda(A) g_\lambda(B), \quad -1 \le \lambda \le \infty \tag{3}$$

There are four basic types of fuzzy integrals, the Choquet integral, the Šipoš integral, the Sugeno integral, and the t-conorm integral (Garbisch, et al., , 2000). Of these four, the Choquet integral is applied to complex problems because it is the fuzzy integral that most resembles the LeBegues integral. On the other hand, the Sugeno integral is the simplest of the four.

The familiar Riemann sum is defined by expression (4). The equation shows the Riemann sum of function f on an open convex subset (x_o, x_n) of the real line partitioned into finitely many segments at the partition points $x_1, \ldots, x_{n-1}$, and with a finite sequence of numbers $t_o, \ldots, t_{n-1}$, where t_i is a value between x_i and x_{i+1}.

$$\sum_{i=0}^{n-1} f(t_i)(x_{i+1} - x_i) \tag{4}$$

$$\sum_{i=1}^{n} [f(x_i) - f(x_{i-1})] \cdot m(\{x \mid f(x) \geq f(x_i)\}) \tag{5}$$

The LeBegues sum is defined by expression (5). This sum is the sum of values of f on an open convex subset (x_o, x_n) of the real line partitioned into finitely many segments at $x_1, \ldots, x_{n-1}$. Each term in the sum is the difference between the function values at adjacent partition points, x_{i-1} and x_i, multiplied by the value of the measure m of the set of all x for which $f(x)$ is at least as great as $f(x_i)$.

In the LeBegues sum, the measure m must be additive, and, in the Riemann sum, the measure is simply the distance between adjacent partition points, which in fact defines an additive measure. The Riemann integral and the LeBegues integral are the limits of their respective sums as the distances between the partition points approach zero.

The Choquet integral is generally used only when there are finitely many function values, in which case it is equal to a finite sum as shown in equation (6). This sum is reminiscent of the LeBegues sum, but there are additional stipulations on the values of the function f, as shown in (7).

$$(C)\int f \quad dm = \sum_{i=1}^{n} [f(x_i) - f(x_{i-1})] \cdot m(\{x \mid f(x) \geq f(x_i)\}) \tag{6}$$

$$\begin{aligned} & f(x_1) \leq f(x_2) \leq \ldots \leq f(x_n), \\ & f(x_0) = 0 \end{aligned} \tag{7}$$

For this integral, the function f, defined on the real numbers, must be monotonic and such that $f(x_0) = 0$.

Figure 1 portrays the evaluation of the Choquet integral that is expanded in equation (8). Here the space X is partitioned in the same manner as for the Lebegues integral, but the λ–fuzzy measure is no longer additive.

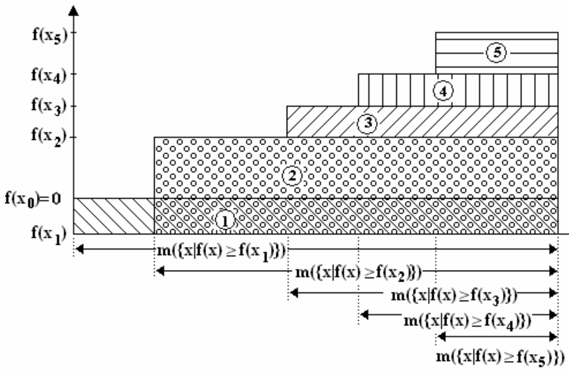


Figure 1: The Choquet integral of a function f

$$\begin{aligned} & (C)\int f \quad dm = \\ & f(x_1) - f(x_0) * m(X) \\ & + f(x_2) - f(x_1) * m(\{x \mid f(x) \geq f(x_2)\}) \\ & + \ldots \\ & + f(x_5) - f(x_4) * m(\{x \mid f(x) \geq f(x_5)\}) \\ & = (1) + (2) + (3) + (4) + (5) \end{aligned} \tag{8}$$

There are two major ways to find λ–fuzzy measures, with optimization methods and with constraint satisfaction methods. The most commonly used optimization method is sequential quadratic programming (SQP). To present this method, we let g be the value of the Choquet integral (equation (6)) of f on interval (x_o, x_n) with fuzzy measures $m = [m_1, m_2, \ldots, m_n]$ and function values $f(x_i)$, $1 \leq i \leq n$. (In our application, the function values are the sensor values.) SQP is a minimization technique formulated as in (9) to find the value of m that minimizes the error function $q(m) = (g-y)^2$, where y is the measured value, subject to the set of constraints (10). For r_e training inputs, the A_i and A_j, $1 \leq i \leq r_e$, $r_e + 1 \leq j \leq 2 r_e$, are the training inputs (in our case, the vectors of sensor values) and the b_i and b_j are the expected training outputs of the system for inputs A_i and A_j, respectively. The values b_j define the system's upper limits, and values b_i define the desired results for the system. In our case, the b_i and b_j are the true distance values. b_k and b_{k+re}, $1 \leq k \leq r_e$, are not necessary the same values but (as in our case) may be. Likewise for the A_k and A_{k+re}, $1 \leq k \leq r_e$.

$$\min_{m \in \Re^n} \quad q(m) = \frac{1}{2} m^T H m + c^T m \tag{9}$$

$$\begin{aligned} & A_i m = b_i \qquad i = 1, \ldots, r_e \\ & A_j m \leq b_j \qquad j = r_e + 1, \ldots, r \end{aligned} \tag{10}$$

Equation (9) shows the minimization of the function $q(m)$ with respect to the vector of fuzzy measures m and the Hessian matrix H (second order partial derivative of q with respect to m). The minimization is usually accomplished by calculating the positive definite quasi-Newton approximation of the Hessian H using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method.

BFGS is a quadratic modeling method derived from gradient decent algorithms (Walter, 1976) (Lunenberg, 2001) (Garbisch et al., 2000) (Fletcher, 1987).

3. TEST-BED SET-UP

The Bayesian network and fuzzy Integral algorithms outlined above are tested on the ActivMedia robot shown in figure 2.

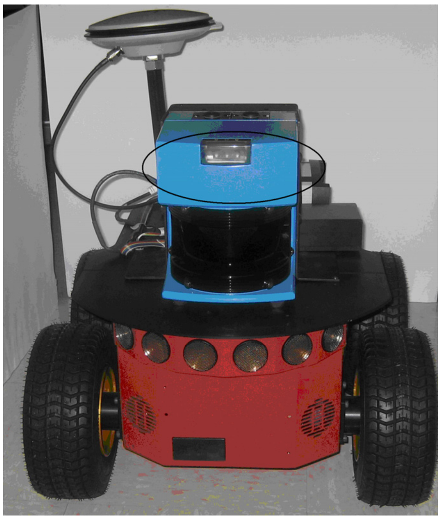
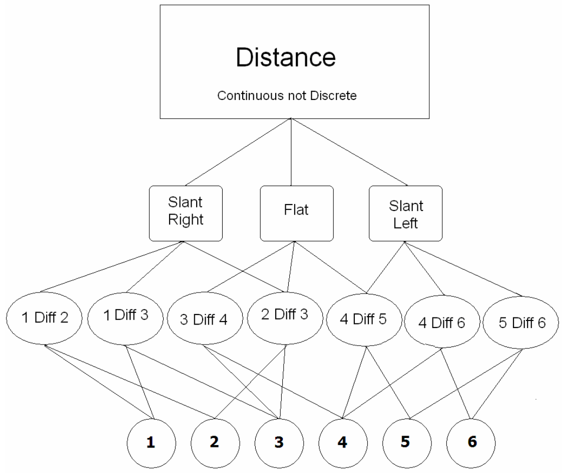


Figure 2: ActivMedia robot with 16 sonars and a rangefinder laser

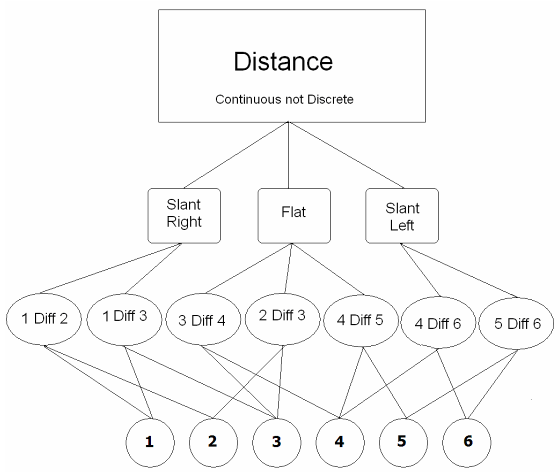
This system is programmed with C++ on a Linux operating system. The robot is equipped with 16 sonar sensors (8 on the front and 8 on the rear) and a laser with a 180° measurement range. The outputs of the six sonar sensors on the front of the system are combined using the two methods described in the above sections. The laser located directly above the two center sonars provides true distance values, against which the sonar values may be validated. The robot is programmed to do a wall following task using either one of the Bayesian networks shown in figure 3 or our fuzzy integral algorithm with measures defined using the SQP method.

The Bayesian network structures shown in figure 3 use the probability of the distance for each of the six sonar sensors. It also exploits the conditional independence between the probabilities of the differences between the six sensors. There are 15 different combinations of differences, but only the seven that demonstrate the strongest relationships are used. The third level looks at the probability that the object that the robot is approaching is either flat, slanted to the left, slanted to the right, or a combination. In addition, the top level is a continuous probability of the distance based on all the other probabilities. Figures 3a and 3b have different numbers of casual edges; 3a has the most, causal edges and shows a stronger relationship between the 'difference' level and the third level.

Since all computation is in real time, the sample set is kept minimal (ten). Since neither the Bayesian network nor the fuzzy integral requires separate training and testing sets, any of the ten original sensor samples can also be used to test the accuracy of the algorithm. The fuzzy integral evaluated here uses the basic Choquet integral with the SQP algorithm to determine the fuzzy measures. Within the minimization function, the samples are entered as the criteria shown in (10) above. The sample sets are evaluated within the fuzzy integral equation then entered as the upper bounds and the target equalities for the inequalities and equations in (10).



(a)



(b)

Figure 3: Bayesian network structures (a) and (b)

The Hessian here is a 6 x 6 matrix of second order partial derivatives of the error squared. The fuzzy measures are determined for the Choquet integral using the function values and the same ten samples used to determine the probabilities in the Bayesian network. After the fuzzy measures are calculated, the fuzzy integral is tested for accuracy using any of the members of the sonar sample set.

4. SIMULATION RESULTS

Figures 4, 5, and 6 show the results of running each of the algorithms while the system is performing a wall-following routine. These results are based on forty runs of the two algorithms using the same sensor sample sets. Figure 4 shows how the fuzzy integral results compare to the measured values, found using the rangefinder laser. Note that most of the values are quite accurate.

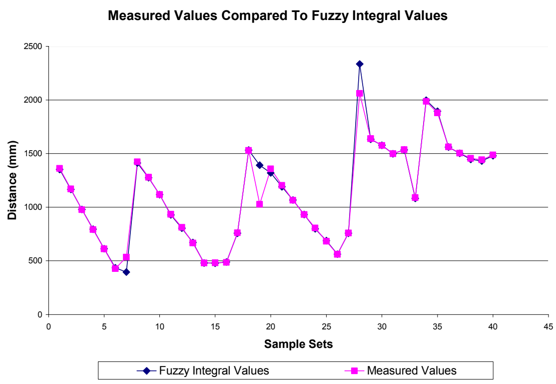


Figure 4: Algorithm results of preliminary runs using a fuzzy integral

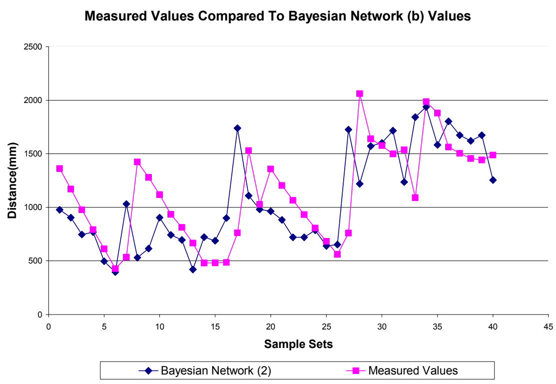


Figure 5: Algorithm results of preliminary runs using Bayesian network structure (3a)

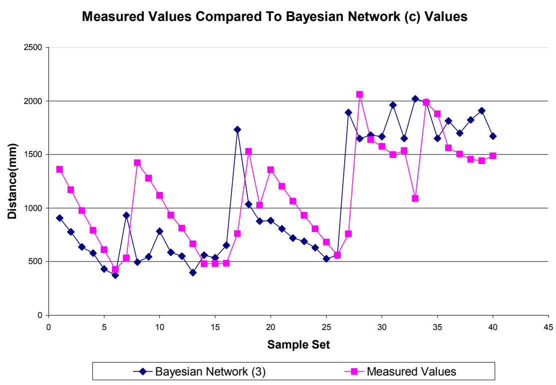


Figure 6: Algorithm results of preliminary runs using Bayesian network structure (3b)

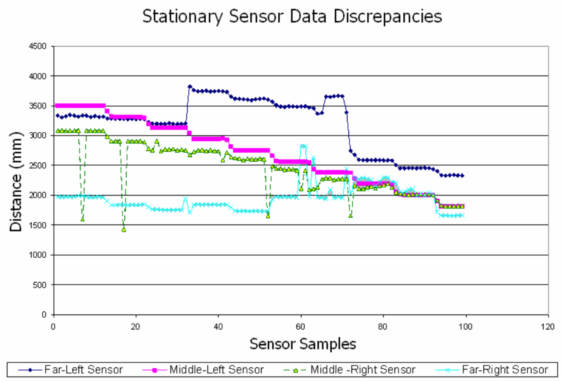


Figure 7: Sensor results of noisy simulations

Figures 5 and 6 show how the Bayesian network results compare to the measured values and how different structures deliver different results. All of the Bayesian network methods tend to over- or under-estimate the measured values for most cases, but the structure shown in figure 3a, which contains the most causal edges, gives the best results of the two Bayesian networks. Overall, the fuzzy integral is more accurate than the current Bayesian network structures.

The amount of noise in the sensor measurements is evident in figure 7, which plots raw sensor values over time. There is not only a drift in the sensor values but also some randomness. These 400 sensor measurements (100 for each sensor 1, 3, 4, 6) were taken while the robot was stationary. Only four of the six sensors are sampled, to provide an un-crowded image that demonstrates the sensor noise.

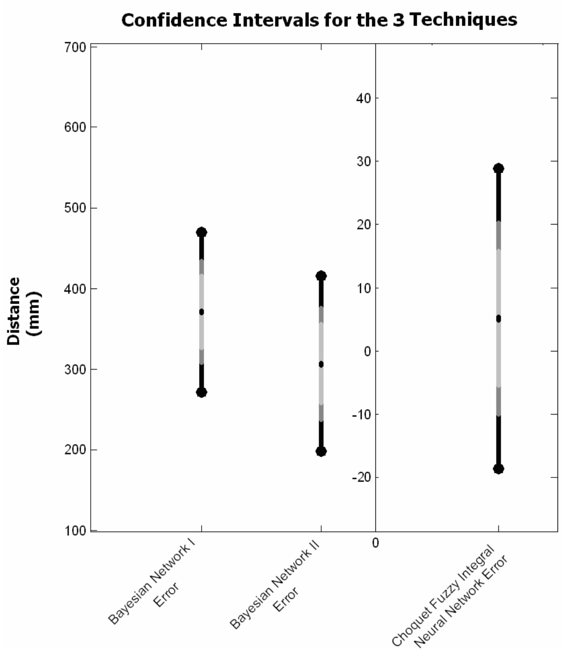


Figure 8: The confidence intervals for the error between the accepted values and the results found using the different techniques.

The error confidence intervals for each technique are shown in figure 8. The reliability of a technique is indicated by the size of the areas that have 75 (light grey), 90 (dark grey), and 99 (black) percent of the error values in reference to the mean error for each technique (table 1).

The Bayesian networks have similar error confidence intervals, although the Bayesian network with more causal links has slightly better results. The technique that calculates the most accurate distance values is the Choquet fuzzy integral.

Technique	STD	Mean Error	75%	90%	99%
Choquet Fuzzy Integral	65.1	5.1	10.6	15.1	23.7
Bayes Network I	271.3	370.9	44.1	63.1	98.8
Bayes Network II	298.0	370.9	48.5	69.3	108.6

Table 1: Summary of results for the techniques and their confidence intervals

Clearly, some of the sensors are more stable than others. In this case, the far-left sensor is the most unreliable, but this is not always so. Depending on the trajectory of the robot relative to the walls, other sensors can provide values that are inaccurate as well. Figure 7 also indicates that, as time passes, the sonars tend to converge to a more stable set of values. If time were not an issue in a dynamic system, more accuracy could be achieved by waiting longer before acquiring the data used with the Bayesian networks and with the fuzzy integral.

5. CONCLUSION

The Bayesian networks gave some good and some bad results, yet the results seem to shadow the measured distance values. Bayesian networks worked to some degree for this lower-level sensor fusion problem. Bayesian network structure 3a has slightly better results than structure 3b (which has fewer causal links). Generally, Bayesian networks are better at higher-level decisions and require more lower-level information than fuzzy integrals. This is why we plan to look at incorporating Bayesian networks as a higher-level sensor fusion technique.

The fuzzy integral estimation using SQP to determine the non-additive fuzzy measures provides an even more accurate estimation of the actual distance the robot is from the wall. The SQP algorithm yields an accurate result even in an environment with noisy sensors, as in these experiments. Using the SQP method to determine a set of fuzzy measures is an adequate method to assist the robot at wall following.

Fuzzy integrals can be powerful with an accurate set of fuzzy measures because they can handle interactions between inputs in a non-additive manner. The fuzzy measure's ability to show constructive and destructive interaction between the sensor values allows the fuzzy integral to give a good approximation of the actual measured values when suitable measures are found. A fuzzy integral, however, is only as accurate as its measures. Since the set of fuzzy measures found by the SQP algorithm, though adequate for this problem, was not optimal, other methods will be investigated. We intend to investigate a hybrid algorithm that uses a neural network to find the fuzzy measures and Bayesian networks to assist in higher-level decision-making processes. Better fuzzy measures and combining these methods should improve the reliability of the system's decisions.

REFERENCES

Klein, L. Sensor and Data Fusion Concepts andApplications, Bellingham, Washington SPIE Optical engineering Press, 1999

Garbisch, M., Murofushi, T,. and Sugeno, M., Fuzzy Measures and Integrals, New York, Physica-Verlag Heidelberg, 2000.

Elmenreich, W. "Sensor Fusion in Time-Triggered Systems", *Dissertation, the department of Technical Science and Information Technology, Technical University at Vienna*, October, 2004.

Neapolitan, R., Learning Bayesian Networks, New Jersey, Pearson Prentice Hall and Llinas, 2004.

Demircioglu, E., Osadciw, L., "A Bayesian network sensor manager for heterogeneous radar suites", *2006 IEEE Radar Conference*, 24-27 April 2006.

Hall, D., Llinas, J., Handbook of Multisensor Data Fusion, New York, CRC Press LLC, 2001.

Walter, Rudin, Principles of Mathematical Analysis, 3rd Ed., McGraw-Hill, New York, NY, 1976.

Lunenberg, D.G. Linear and Nonlinear Programming 2ed. Massachusetts, Kluwer Academic Publishers, 2003.

Fletcher, R., Practical Methods of Optimization 2ed. New York, John Wiley and Sons, 1987.

Borenstein, J., Everett, H.R., Liqiang, F., Navigation Mobile Robot Systems and Techniques. : Massachusetts: AK Peters, Ltd., 1999