Modern AI Approaches for Building Robust FDIR Components

Matti Paakko(2) , Petri Myllymäki(1), Niklas Holsti(2) and Henry Tirri(1)

(1)Complex Systems Computation Group
P.O.Box 26, Department of Computer Science
FIN-00014 University of Helsinki, Finland
Email: cosco@cs.Helsinki.FI

(2)Space Systems Finland Ltd.
Kappelitie 6, FIN-02200 Espoo, Finland
Email: matti.paakko@ssf.fi, niklas.holsti@ssf.fi

INTRODUCTION

Space is a harsh and a changing environment in which spacecraft have to survive many years. The irregularities of this environment, radiation and the possible threat of debris impact means the spacecraft has to be provided with adequate hardening. Defining the level of protection necessary is of prime importance. In addition, there are very limite possibilities to perform physical repair operations after the launch. This imposes strict requirements for the quality of the design and testing of the spacecraft. Nevertheless, faults and failures are unavoidable.

Failure tolerant technology involves a variety of techniques intended to maintain an entire system in normal working conditions even if one of its components fails, by means of failure detection, isolation and recovery (FDIR). Such technology is critical in space where it is almost impossible to repair a damaged device if a failure takes place. A commonly adopted approach to failure tolerance is to provide for redundant device configurations.

However, obviously devices and subsystems cannot always be duplicated, and there are strong weight and power restrictions imposed on a payload. More advanced technology and research aim towards an improved failure tolerant design based on the concept of higher efficiency and new ideas. Typically the new methods are able to detect failures at a higher level, and thus try to detect failures in subsystems, where such failures are difficult to diagnose by only monitoring the independent components. These methods try to predict dangerous trends that could lead to catastrophic results.

New failure-tolerant technology, based on new failure detection methods, promises efficient failure-tolerance and will thus, hopefully, reduce the cost and increase the reliability of space systems. Advanced autonomous systems detect failures and are able to restore themselves (or at least try to continue the mission), albeit with degraded performance. In the next section we discuss in more detail what type of requirements one can set to the on-board FDIR procedures of such systems. We then proceed by considering te problem, which approaches look most promising for fulfilling these requirements from the artificial intelligence research point of view. Finally, we briefly describe a concrete demonstration project where these approaches will be implemented and tested.

REQUIREMENTS FOR ROBUST AUTONOMOUS SYSTEMS

A *fault* is a physical condition that causes a device, a component, or an element to fail to perform in a required manner, such as an incorrect step, process, or data definition in a computer program [1]. A fault is a hardware or software problem that can be activated and can lead to an error that can eventually be detected. A *failure* is defined as the inability of a system or component to perform its required functions within specified performance requirements. Failure tolerance is defined as the ability of a system or subsystem to perform its function(s) in the presence of failures within its hardware or software [2]. The system can be built fault or failure tolerant or both. If the system, including software, handles most probable, and some less probable but possibly hazardous faults, it will be able to preclude most relevant failure scenarios and gain better failure tolerance [3].

Robustness is often described as the system's ability to continue operating sensibly even under a load that is higher or different than specified. This "load" can include faults, failures - including design failures and environmental disturbances (beyond specification). Thus "robustness" is considered equivalent to "operation over and beyond the call of duty" [4][5].

What factors can make a spacecraft fail to operate, or fail to be robust? At least the following are possible:

- A physical fault in a critical component, whether internal (manufacturing fault, wear) or external (e.g. debris impact).
- A specification or design error that leads to erroneous use of healthy components, e.g. a software error.
- An unforeseen environmental factor, e.g. an unexpectedly high radiation level, exceeding the operational range of a component or of the design.
- Operator error, e.g. a harmful telecommand.

A more systematic and extensive classification of faults is given in [6]. Still, the prime players are evident from the above examples: the spacecraft's components, its design, its environment, and its operation. Any factor that threatens the functioning or correctness of one of these components is a potential cause of failure. Often, combinations of threats are especially dangerous. For example, the failure of a component may force the system to operate in an abnormal mode, which may expose design errors and operational problems that were not detected in the testing phase.

As the duration of space missions increases – reaching a decade or more – unforeseen changes in operational conditions can cause severe threats, such as specification faults or faults in interfacing systems, e.g. faults in the ground segment. Human operator errors are also not uncommon in space mission disasters.

Today, the environmental conditions that will be met by most spacecraft can be predicted and considered in the design. However, a failure in one part of a spacecraft can change the environment for other parts, perhaps exceeding the design range. For example, a failure of attitude control can change the temperature balance within the spacecraft.

Consequently, in order to build a robust autonomous system, one needs to design software that is capable of functioning in a dynamically changing reactive environment, and working with uncertain (noisy or missing) information. In case of unexpected events, the performance of the system should degrade gracefully, and the system should be able to estimate the risks associated with each possible action. In addition, the system needs to be able to function within the limited computational resources available in an autonomous spacecraft. In the following section, we consider these requirements from the artificial intelligence research point of view. While our focus is on FDIR, similar requirements apply to autonomous reactions to any type of event, whether it represents a success, failure, or an opportunity or threat.

ARTIFICIAL INTELLIGENCE IN ROBUST AUTONOMOUS SYSTEMS

From the software point of view, FDIR involves *classification* (of events or situations) and *decision* (whether to react and how to react) problems. When considering suitable computational methods for FDIR, the following aspects can be observed. First of all, it is evident that the chosen method has to be able to process uncertain information: the dependencies between the model components, effects of different actions, as well as the observations about the status of the components are usually not deterministic (certain), but stochastic (uncertain) in nature. This rules out solutions based on purely deterministic rules and procedures, such as most model-based reasoning or procedural reasoning systems.

A consistent calculus for uncertain reasoning is one for which the result is always unambiguous: no matter how the calculus is implemented, when starting from the same set of hypotheses, the result (the calculated uncertainty of an event) remains always the same. Probability calculus, being provably consistent, can hence be considered more reliable in this respect than non-axiomatic models, such as neural networks, or models based on calculus that has not been proven to be consistent, such as various rule-based or fuzzy systems. In fact, construction of a consistent, non-probabilistic calculus has been shown to be very difficult (see [7] for a more detailed discussion), and from this uncertain reasoning viewpoint many existing rule-based or fuzzy systems can be seen as only approximations of more sophisticated probabilistic systems such as Bayesian networks.

Flexibility and Generality

Any FDIR method should be flexible in the sense that the direction of reasoning should not be fixed in advance - we cannot know in advance where the possible faults will occur, or where the effects of the faults appear. Spacecraft subsystems interact with each other, so a fault in one system may be detected by its effect on another, or vice versa. Bayesian networks [8] provide a flexible methodology as they model the joint probability distribution of the domain variables, and hence can be used for any type of prediction tasks, i.e. for computing the predictive distribution for a freely chosen attribute or attribute set, given the observations concerning some other set of attributes. Most neural network and fuzzy logic systems are typically designed for a specific static prediction task in mind, where the input attributes and output attributes are determined as a first step of the system construction process, and cannot be changed afterwards.

Supporting Recovery Decisions

As fault recovery is an important aspect of the FDIR process, the chosen uncertainty calculus should be able to deal with decisions, and provide means for estimating the possible risks associated with alternative actions. Probability calculus is unique in the sense that it offers a decision theoretical framework for modeling the effects of possible actions, or more precisely, for estimating the expected risks or utilities associated with the actions [9]. What is more, it has been shown that building a system based on the probabilistic decision theory framework produces a coherently rational agent, which means that it is not possible to construct a sequence of events (a so called "Dutch book") so that, in an imaginary betting situation (where the bets correspond to utilities associated with the possible actions), the agent will eventually lose all its betting money (see e.g., [9][10][11]). Coherence seems as a natural requirement for a rational decision maker, but perhaps somewhat surprisingly, it can be also shown that agents based on Bayesian decision theory are the only coherent rational decision-makers [12]. This means that decision theory offers an elegant theoretical framework for optimal decision making. As decision theory can only be applied with probabilistic models, in this respect it seems natural that probabilistic models, such as Bayesian networks, are the models to be used in decision making domains such as FDIR.

Transparency and Understandability

Another important requirement is that there is a need to be able to construct the FDIR module based on available expert knowledge: although the module should be at least to some extent adaptive, the system cannot depend on sample data only, if this type of data is not available. As the parameters of probabilistic models such as Bayesian networks have a concise semantic interpretation, these models can be constructed directly from expert knowledge, so that the produced model reflects the structure of the physical domain to be modeled. This is in contrast to for example neural networks, which are black box systems that are learned from sample data, and if no such data exists, the model cannot be constructed. Although some attempts towards incorporating prior expert knowledge in neural network constructing have been made (see e.g. [13] and the references therein), this approach cannot be considered very promising.

It should also be noted that there exist several algorithms for learning Bayesian networks from sample data [14], which means that if such data becomes available later on as a result of a simulated or real-world use of the modeled system, the data can be used for improving the accuracy of the Bayesian network model.

Computational Resources

Finally, the chosen method needs to be computationally efficient so that it can be implemented and used within the processor performance and memory storage resources of a spacecraft.

In this respect neural networks and fuzzy systems may at first glance seem to be more promising than Bayesian networks: as the number of parameters (the number of hidden nodes in a neural networks) or the exact form of the uncertainty processing calculus of a fuzzy system are basically freely modifiable by the user, the systems can of course be made as computationally efficient as needed. However, it is obvious that this type of simplification affects the quality of the results (i.e. predictions). Besides, this same trivial observation applies also for the Bayesian networks: if it is necessary to reduce the memory storage requirements, the number of parameters of a Bayesian network can be decreased by making more radical independence assumptions. This would probably decrease the accuracy of the predictions made by the model, but it should be noted that this approach offers a theoretical framework for playing with the trade-off between storage requirements and model performance.

On the other hand, the speed of the Bayesian network uncertain reasoning module is usually not a problem since the model can be "pre-compiled" through a so-called triangulation process to a computationally efficient data structure where the computations can be performed rapidly [15][16].

BAYESIAN NETWORKS FOR ROBUST FDIR

Bayesian Modeling

Probability is a mathematical construct that behaves in accordance with certain consistent rules and can be used to represent uncertainty. In order to be able to do robust (plausible) inference under uncertainty, probability needs to be interpreted somehow. Depending on this interpretation, we end up in different inference frameworks; the *classical statistical inference* is based on a long-run frequency interpretation of probability, and the *Bayesian inference* is based on the “degree of belief” interpretation.

The long-run frequency interpretation of probability requires a sequence of repeatable experiments. The probability of an observable event is the long run proportion of the time it happens compared with the total number of observations—where long-run means in the limit as the total number of observations tends to infinity. This definition of probability is natural for example in the context of gambling. However, even in such domains things are not necessarily straightforward. More importantly, no frequency interpretation can be given to a new, unseen event that will be take place sometimes in future.

Alternatively probability can be defined as a subjective assessment concerning whether the event in question will occur. At first sight it might be quite astonishing that anything useful comes from a definition that involves subjective assessment. However, one should remember that scientists in general entertain various theories of the world, but no one knows which theory is correct. Some scientists have strong beliefs in particular theories, and sometimes two different scientists have strong beliefs in theories that contradict each other. Consequently “from a belief to a theory" is common practice in all science. Degree of belief depends on the person who has the belief, as well as on the event on question. In plausible inference, this person could be any observer, in particular it can be an autonomous system.

Bayesian inference is based on the degree of belief interpretation of probability. Since all Bayesian probabilities depend on the available information, they actually are mathematical concepts known as conditional probabilities, and are denoted P(A | I), where I represents the information affecting the probability assignment. The right hand side of the bar “|” is called the conditioning part.

Let us now suppose that we have some data, and denote this data by D. In addition we have several unknown things, which we happen to denote by M. Typical examples of unknown things could be:

- Values related to the model structure we have chosen. These values are needed to uniquely specify the model, and they are called the parameters of the model. The model structure could be e.g. a normal distribution.
- Missing information in the data.
- Events that were not observed directly or exactly.

In the the FDIR context, D represents both known facts about the spacecraft health, and observations such as sensor readings or low-level failure alarms raised by threshold monitoring, built-in tests or the like. M represents both missing observations (perhaps due to sensor failures) and the unknown faults or other events we wish to detect. Since the Bayesian inference uses conditional probabilities to represent uncertainty, we are interested in P(M | D,I)—the probability of unknown things (M) given the data (D) and background information (I). Now the essence of Bayesian inference is in the rule that tells us how to update our initial probabilities P(M | I) if we see data D, in order to find out P(M | D,I). This means that we could “update” our beliefs. This update rule is known as Bayes’ theorem and can be formally expressed as follows:

$$P(M \mid D,I) = \frac{P(D \mid M,I)P(M \mid I)}{\int P(D \mid M,I)P(M \mid I)dM}$$

Consequently Bayesian inference briefly comprises the following principal steps:

- Obtain the initial probabilities P(M | I) for the unknown things. These probabilities are called the prior (distribution).
- Calculate the probabilities of the data D given different values for the unknown things, i.e., P(D | M,I). This function of the unknowns is called the likelihood.
- Finally the probability distribution of interest, P(M | D,I), is calculated using the Bayes' theorem given above. This so called posterior (distribution) will then express what is known about M after observing the data.

Bayes' theorem can be used sequentially, i.e. if we first receive some data D, and calculate the posterior P(M | D,I), and at some later point in time receive more data D', the calculated posterior can be used in the role of prior to calculate a new posterior P(M | D,D',I) and so on. The posterior P(M | D,I) expresses all the necessary information to perform predictions. (The case of making decisions is more complicated, as it involves the notion of a utility or cost associated with a decision. We do not address decision theoretical questions in this paper.) The more data we get, the more certain we will become of the unknowns. It should be observed that in Bayesian inference there is no such thing as "too little data"- even having one data element is better than having none.

If this is all there is to it at this abstraction level, where are the difficulties? Putting the philosophical debates aside, to implement the theory above requires considerable effort in practice. First, deriving proper priors to initialize the reasoning process is not an easy task. Second, calculating the likelihood, and in more complex cases also the denominator integral in a closed form can be difficult. One special model class where these implementation problems can be solved in an elegant manner, is the class of Bayesian network models discussed in the next section.

Bayesian Networks

A Bayesian (belief) network [8] models a problem domain through a set of probabilistic relations between the relevant domain variables. The structure of a Bayesian network model is given as a directed acyclic graph (DAG), where each node in the network represents a domain variable, which are usually assumed to be discrete. The arcs of the network represent probabilistic dependencies between the variables. More precisely, a missing arc represents an independence, through a somewhat complex graph-theoretic definition. The probabilistic relations associated with each variable (node) are expressed as conditional probability distributions, where the conditioning is done according to variables associated with the parent nodes in the underlying graph. At intuitive level, any observation (observed data) can then be evaluated "against" the model, i.e., one can calculate the likelihood of the observation with respect to various possible states of the system (variable values), and consequently use the Bayes´ theorem to find the probability of each possible cause for the observations.

More technically, a Bayesian network is a model of the joint probability distribution of the problem domain. Thus the observed facts (or beliefs) about the values of some of the domain variables can be entered into the model as input by setting the states of the corresponding nodes, and the model then updates the beliefs about the unobserved variables by computing the posterior probability distribution for each node (or a set of nodes), given the input. One of the main reasons for the recent success of Bayesian network models lies in the fact that researchers have been able to develop efficient algorithms for performing this type of probabilistic inference within this framework by exploiting the characteristics of Bayesian network models. An additional advantage with the Bayesian network approach is that the probability calculus is robust in the sense that the outcome does not fluctuate radically as a consequence of a minor change in the model parameters [17]. It should be noted that most fuzzy logic systems are not proved to be consistent, and can possibly be quite unrobust in practice.

The probabilistic calculus used in the Bayesian network approach offers the important possibility to apply decision theory in estimating the possible risks associated with each of the possible future actions. For this reason, models based on probability theory are especially important for safety-critical domains, such as spacecraft FDIR. A decision network (or influence diagram) [18] is an extension of a standard Bayesian network, containing decision nodes and utility nodes for utilizing the decision-theoretic framework. Decision nodes correspond to controllable variables related to possible actions, while utility nodes correspond to the resulting quantities (risks or utilities) to be optimized. "Solving" the decision network means finding values for the decision nodes so that the expected utility is maximized (or equivalently, the expected risk is minimized), given the observed facts/beliefs.

Although the variables (nodes) of a Bayesian network model are usually assumed to be discrete, it should be noted that the modeling framework extends also to cases with continuous variables. What is more, Bayesian networks offer a theoretically consistent framework for handling domains with both discrete and continuous variables. Many alternative approaches, such as most neural network models, are inherently designed for the continuous case, and handling discrete variables can be very problematic.

A CONCRETE EXAMPLE

Advanced FDIR

As an example of Bayesian networks in spacecraft FDIR, we briefly present the Advanced FDIR (AFDIR) project. The goals of this project are:

- To implement modern failure-detection and diagnosis methods, including Bayesian networks.
- To improve mission survival and utility by graceful recovery.
- To provide reusable FDIR software components.

The project is Call-Off-Order 5 of the ESA frame project “Software For an Autonomous Satellite” (SFAS). Space Systems Finland Ltd. is tasked with the major part of the AFDIR work, with Astrium SAS as the Prime Contractor (managed by Mr. Philippe David) and ESTEC as the ultimate customer (managed by Mr. Eric Bornschlegl). The Complex Systems Computation Group (CoSCo) from the University of Helsinki provides expertise in Bayesian methods.

Failure-Detection and Diagnosis Methods

In addition to the traditional failure-detection methods, such as limit monitoring, correlation tests, transient filtering, etc., the AFDIR includes:

- Kalman filtering, to compute least-squares estimates of the real state from noisy or drifting sensors [2].
- Weighted Sum-Squared Residual (or Chi-Squared) test [20], which accumulates Kalman-filter residuals over a time-window and is more sensitive to small errors than a single-sample test.
- Parity-vector methods such as the Generalized Likelihood Test (GLT) [20][21], in which data from multiple, redundant sources (sensors) are cross-checked for consistency, and the aberrant data can be identified.
- Voting techniques based on the Random Sample Consensus method [22]. As GLT, this cross-checks redundant values, but is designed for errors with non-normal distribution (frequent occurrence of very large errors).
- Various spacecraft simulations for computing “expected” values. This includes attitude dynamics, reaction-wheel dynamics, magnetotorquer dynamics, orbit propagation, and a model of the Earth's magnetic field.

The above methods detect discrepancies between actual values (e.g. from sensors) and expected values (e.g. from models or redundant sensors). They can sometimes also identify the value or part that is in error. Failures often affect many different observable values and behaviors. The task of diagnosis is to consider all of this “evidence” jointly, and infer the most likely common cause: the part that has failed. The AFDIR includes two “integrative” methods that can deduce underlying failures from multiple, superficial indications or symptoms:

- Probabilistic reasoning using Bayesian Networks.
- Model-based diagnosis using Causal Networks.

Bayesian Networks

The input nodes (variables) of the Bayesian networks are discrepancy-measures from lower-level failure-detection methods. The output nodes give probabilities for spacecraft part failures. For FDIR purposes, two notable advantages of Bayesian networks are that they can deal with *missing* data and with *probability* data. The first property makes it easy to use the same network in different configurations, for example with different sensors active. The second property means that the basic failure-detection methods, such as the GLT, do not have to discretize their error-measure against a fixed threshold. Instead, the probability of the error-measure (e.g. chi-squared) can be fed as such into the network.

The AFDIR uses the Probability Propagation in Trees of Clusters (PPTC) approach [23] to pre-process the Bayesian networks on the ground and to evaluate them on-board as part of FDIR execution.

As a pilot application we have defined a hypothetical satellite called ASOS, for "Advanced Smart Observation Satellite". ASOS resembles the small Earth-orbiting satellites being built today, with an attitude-control system using autonomous star cameras, a GPS receiver, magnetometers, magnetotorquers and reaction wheels.

Fig. 1 shows a diagnostic Bayesian network for ASOS attitude sensors. It contains input nodes for failure-detection tests (two bottom rows), part health (top row) and on/off-status (second row). The middle row output nodes indicate probabilities for part failures. For example, the autonomous star camera 1 (ASC1) failure depends only on the ASC1 health. On the other hand, ASC1 failure and on/off status affect the ASC1 built-in test (BIT), several correlation tests (CT) and a GLT test.

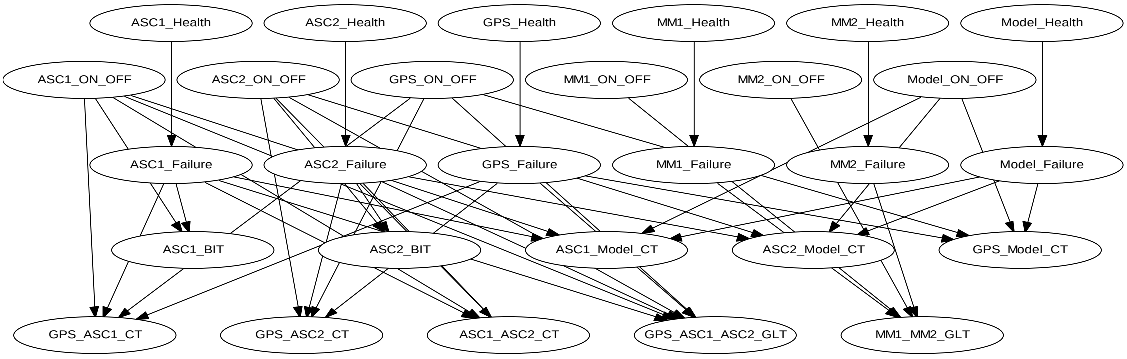

Fig. 1. A Bayesian Network for Attitude Sensor Diagnostics

Causal Networks

A Causal Network (CN) [24] models a system as a set of connected parts, where each part has zero or more Boolean inputs and computes one Boolean output. The parts are connected in a directed acyclic graph where a node represents the output of a part. Error propagation is modelled by equations of the form "c_in ∧ c_ok ⊃ c_out", i.e. if the input to part C is correct, and C works, then the output from C should be correct. From observed errors, model-based diagnosis can infer the smallest set of faulty parts.

In ASOS, Causal Networks will model the TC/TM subsystem, which has redundant, cross-connected encoders, decoders, transmitters, receivers and antennae. A single observed failure (lack of TC, for example) usually cannot resolve the fault to one component, but the Causal Network can guide a sequence of trial-and-error reconfigurations intelligently, until the faulty components are isolated.

Reusability

The AFDIR is divided into several levels, from simple mathematical algorithms up to the Bayesian and causal network levels. Each AFDIR level is divided into two layers: an *algorithm layer* of purely callable subprograms, and a *structure layer* with a more convenient declarative interface. The latter is designed for use in a *signal-processing network* 0. It is similar to the data-flow block diagrams used with MATLAB and other tools, and consists of signals and signal-operators that are connected into a data-flow network. A *signal* represents a stream of time-stamped values, e.g. sampled sensor readings. A *signal-operator* is a processing function that takes some signals as inputs and produces some signals as outputs. All the AFDIR failure-detection and diagnosis methods, including Bayesian networks, are provided as signal-operators.

CONCLUSIONS

Our analysis of the problem of FDIR for autonomous spacecraft indicates that probabilistic reasoning, and in particular its realization as Bayesian networks, is a suitable computational basis for FDIR. This conclusion is supported by the extensive use of Bayesian methods in ground-based systems, and by the success of the Advanced FDIR project with the ASOS example. However, since Bayesian networks are a very flexible method, much work remains to discover how they can best be applied in the varied contexts and designs of current and future spacecraft.

REFERENCES

[1] The IEEE Standard Dictionary of Electrical and Electronics Terms, 6th Ed., IEEE inc., http://standards.ieee.org/, 10 December 1996.

[2] Software Safety, NASA Technical Standard, NASA-STD-8719.13A, http://satc.gsfc.nasa.gov/assure/distasst.pdf, 15 Sep. 1997.

[3] NASA Guidebook for Safety Critical Software – Analysis and Development, NASA Lewis Research Center, Office of Safety and Mission Assurance, NASA-GB-1740.13-96, http://www.ivv.nasa.gov/SWG/docs/safety/-main_body.pdf, 13 July 1995.

[4] Cassini Spacecraft Design Features, JPL, http://www.jpl.nasa.gov/Cassini/Spacecraft/design_feats.html.

[5] Telecommunications: Glossary of Telecommunication Terms, National Communications System Technology & Standard Division, Federal Standard 1037C, Superseding Fed-Std-1037B, http://www-library.itsi.disa.mil/org/fed_std/html/fs-1037c.htm, June 1991.

[6] Fault models and Associated Fault Detection Techniques, Software for an Autonomous Spacecraft, Matra Marconi Space, ILB.NT.HS.2885.00, Issue 0, Rev. 0, 20 January 2000.

[7] H.Tirri, Plausible Prediction by Bayesian Inference, Ph.D. Thesis, Report A-1997-1, Department of Computer Science, University of Helsinki, 1997.

[8] R. Cowell, P.A. Dawid, S. Lauritzen and D. Spiegelhalter, *Probabilistic Networks and Expert Systems*, Springer,New York, 1999.

[9] J. O. Berger, Statistical Decision Theory and Bayesian Analysis, Springer-Verlag, New York, 1985.

[10] J. M. Bernardo, A. F. M. Smith, Bayesian theory, John Wiley, 1994.

[11] M. H. DeGroot, Optimal statistical decisions, McGraw-Hill, 1970.

[12] J. Paris, The Uncertain Reasoner's Companion: A Mathematical Perspective, Volume 39, Cambridge Tracts in Theoretical Computer Science, CambridgeUniversity Press, Cambridge, 1994.

[13] P. Myllymäki, Massively Parallel Probabilistic Reasoning with Boltzmann Machines, Applied Intelligence 11, pp. 31-44, 1999.

[14] D. Heckerman, D. Geiger, D. M. Chickering, Learning Bayesian Networks: The Combination of Knowledge and Statistical Data, Machine Learning 20, pp. 197-243, 3 September 1995.

[15] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[16] F. Jensen, An Introduction to Bayesian Networks, UCL Press, London, 1996.

[17] M. Henrion, M. Pradhan, B. Del Favero, K. Huang, Why is diagnosis using belief networks insensitive to imprecision in probabilities?, Uncertainty in Artificial Intelligence, Proceedings of the Twelfth Conference, edited by E. Horvits and F.Jensen. Morgan Kaufmann Publishers, pp. 307-314, 1996.

[18] R. D. Shachter, Probabilistic Inference and Influence Diagrams, Operations Research 36, 4 (July-August), pp. 589-604, 1988.

[19] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," University of North Carolina at Chapel Hill, TR 95-041, February 8, 2001. http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html.

[20] J. Borde, "Implementation of AOCS Configurations," *Technical Note 5: Fault Management System and Redundancy (WP 250),* Matra Marconi Space, reference S260/JB/NT/17.96, Issue 1, 6/26/96.

[21] K.C. Daly, E. Gai. and J.V. Harrison, "Generalized Likelihood Test for FDI in Redundant Sensor Configuration," *Journal of Guidance and Control*, Vol 2, No 1, Jan.-Feb. 1979.

[22] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis," *Communications of the ACM*, Vol 24, No 6, pp. 381-395, June 1981.

[23] C. Huang and A. Darwiche, "Inference in Belief Networks: A Procedural Guide," *International Journal of Approximate Reasoning* 11:1-158, 1994. http://www.cs.ucla.edu/~darwiche/cs262a/hd_inference_ijar.ps.

[24] A. Darwiche, "Model-Based Diagnosis under Real-World-Constraints," *AI Magazine,* Vol 21, No 2, pp. 57-73, Summer 2000. http://www.cs.ucla.edu/~darwiche/realize.ps.

[25] N. Holsti and M. Paakko, "Towards Advanced FDIR Components," *Data Systems in Aerospace, DASIA 2001*, Nice, France, May 2001.