Diagnosis for Control and Decision Support in Complex Systems

Blanke, Mogens; Hansen, Søren; Blas, Morten Rufus

Published in:
Proceedings Volume from the Special International Conference on Complex Systems

Publication date:
2011

Document Version
Publisher's PDF, also known as Version of record

Proceedings Volume from the Special International Conference on Complex Systems:
Synergy of Control, Communications and Computing - COSY 2011
Hotel Metropol Resort, Ohrid, Republic of Macedonia, September, 16–20 2011

Diagnosis for Control and Decision Support in Complex Systems

Mogens Blanke [*,**] **Søren Hansen** [*,***] **Morten Rufus Blas** [****]

[*] *Automation & Control Group, Department of Electrical Engineering, Technical University of Denmark, Kgs. Lyngby, Denmark, (e-mail: mb@elektro.dtu.dk)*
[**] *CeSOS, Norwegian Univ. of Science and Technology, Trondheim, Norway*
[***] *Danish Forces Joint UAV Team, Gnibenvej 55, Sjællands Odde, Denmark.*
[****] *CLAAS Agrosystems, Bøgeskovvej 6, Kvistgård, Denmark*

Abstract: Diagnosis and, when possible, prognosis of faults are essential for safe and reliable operation. The area of fault diagnosis has emerged over three decades. The majority of studies related to linear systems but real-life systems are complex and nonlinear. The development of methodologies coping with complex and nonlinear systems have matured and even though there are many un-solved problems, methodology and associated tools have become available in the form of theory and software for design. Genuine industrial cases have also become available. Analysis of system topology, referred to as structural analysis, has proven to be unique and simple in use and a recent extension to *active structural* techniques have made fault isolation possible in a wide range of systems. Following residual generation using these topology-based methods, deterministic and statistical change detection has proven very useful for on-line prognosis and diagnosis. For complex systems, results from non-Gaussian detection theory have been employed with convincing results. The paper presents the theoretical foundation for design methodologies that now appear as enabling technology for a new area of design of systems that are reliable in practise. Yet they are also affordable due to the use of fault-tolerant philosophies and tools that make engineering efforts minimal for their implementation. The paper includes examples for an autonomous aircraft and a baling system for agriculture.

Keywords: Fault diagnosis, Fault-tolerant Control, Change Detection, Complex Systems.

1. INTRODUCTION

Diagnosis of faults and active accommodation of faults are tools to prevent that faults develop into failure. Diagnosis is needed for fault-tolerant control where the diagnostic information is used without operator intervention to handle a fault or it is used by a human supervisor in support for fault-tolerant operation.

The theory of fault diagnosis has a long history. Early papers that helped the directions in this field included seminal results on generation of residuals using parity space approaches Chow and Willsky (1984), combined overview and research articles Gertler (1988), Horak (1988), Gertler (1988), Gertler (1997), Isermann (1997). Early applications appeared in Patton (1991) for diagnosis of flight control systems, in Dunia et al. (1996) for chemical processes, in Isermann (1998). The theory for fault-diagnostic observers was pioneered in Frank (1990), Frank (1994), and generation of robust residuals in Frank and Ding (1994), Chen and Patton (1996), Romano and Kinnaert (2006) with extension to nonlinear systems using geometric theory in de Persis and Isidori (2001). Extension to fault-tolerant control emerged in Blanke (1996), Blanke et al. (1997), Wu and Klir (2000). The important issue of threshold selection for diagnosis was treated in papers by Emami-Naeini et al. (1988) for the uncertain deterministic case and for the statical case by Basseville (1988). Essential books in the area include the early edited volume Patton et al. (1989), textbooks in the area Zhang (1989), Basseville and Nikiforov (1993), Gertler (1998), Mangoubi (1998), Chen and Patton (1999), Blanke et al. (2003, 2nd ed. 2006), Isermann (2005), Noura et al. (2009), Ducard (2009). Several aspects of control and diagnosis of complex systems appeared in ÅstrÖm et al. (2000) and recent applications include

Coping with complex systems is a challenge. Dimensionality of the problems met in real life is one challenge, complexity and nonlinearity are others. The ideas of using graph-analysis concepts to help solving complex set of equations were studied early in the applied mathematics community with the result of Dulmage and Mendelsohn (1959) being instrumental for the area. Further theoretical developments in Dulmage and Mendelsohn (1963) and Hopcroft and Karp (1973) made the analysis of the structure of a set of equations or of a system described by such equations a feasible task, even for large systems of nonlinear equations. Structural analysis as this area is called, has been used intensively in Chemical Engineering for solving large sets of equations and issues on solvability

have been pursued in a number of publications, see Unger et al. (1995) and Leitold and Hangos (2001) and the references herein. The structural approach and the features it offers for analysing monitoring and diagnosis problems was first introduced in Staroswiecki and Declerck (1989) and further developed in Staroswiecki et al. (1993). Extensions to analysis of reconfigurability and fault-tolerance emerged in Staroswiecki et al. (1999) and Staroswiecki and Gehin (2000). The structural analysis approach was brought into a digested form in Blanke et al. (2003, 2nd ed. 2006), Travé-Massuyès et al. (2006). Structural analysis has hence evolved during several decades. However, the salient features of the theory and the possibilities it offers have only become apparent to a larger community in the field of automation and automatic control over the last few years. Reasons for the slow penetration into applications origin mainly in the lack of widely available tools to support the structural analysis method for automated industrial systems. Software tools appeared in Düştegör et al. (2004), Blanke and Lorentzen (2006). An approach to highly efficient algorithms was developed in Krysander (2006).

When considering diagnosis for control, the safety, from a structural point of view, depends on services offered by a systems not only in normal operation but more important, after reconfiguration of the system to accommodate one or more faults. Blanke and Staroswiecki (2006) considered the safety of fault-tolerant control schemes when multiple faults could be present. It was shown how structural analysis could be applied to analyse cases of multiple faults and to synthesise residual generators. Fault isolation, which is instrumental for correct fault handling, was addressed and active isolation was introduced from a structural point of view.

This paper revisits the theory for graph-based analysis of systems, introduces the notion of structural active isolability. It first reviews the concept of behaviours and shows how the behaviour of a system is equally well applied on the services offered by hardware and software components. It then interprets the impact on safety of a system that is supposed to work under conditions of multiple faults. While it is well established that structural analysis is very useful for residual generation of technical processes Düştegör et al. (2006), it is less obvious that the generic technique is also very applicable on complex system, even generation of residuals for diagnosis in a natural environment. The combination with change detection techniques is highlighted, and the techniques needed for nonlinear systems, when non-Gaussian residuals occur, are demonstrated. Two real cases are inclded for illustration of the complete design procedure, an aeroplane speed sensor fault diagnosis problem and a case of baler control for agriculture. The latter includes vision senor techniques and very recent results show how the diagnostic concepts can also be applied to enhance the robustness of a vision-based control system.

2. GRAPH-BASED ANALYSIS

Fault-tolerant control uses control or sensor reconfiguration to accommodate failures in instruments, plant components or actuators. Aiming at utilising existing redundancy in instrumentation and control devices as far as possible, fault-tolerant control can be applied to minimise the hazards associated with malfunction, even when several sensors or actuators fail, but several modifications need be made to the usual single fault FTC schemes in order to achieve the necessary level of safety.

2.1 Reconfigurability and safety

The topology (structure) approach that is pursued in this paper considers a system as consisting of a set of components which each offer a service and performs this service through defined normal behaviours. A component can offer different versions of services and command to the component can define which version of a service is made available. Within a component, fault-tolerant techniques can use fault-diagnosis and fault-handling to switch between services or offer a service in a version with degraded performance if local malfunction should make this necessary.

2.2 Subsystem services

A system breakdown in Fig.1 shows three different topologies, by which we mean the arrangement of the system components and their interaction. Component k has input u_k , output y_k, parameters θ_k and a behaviour $c_k(y_k, u_k, \theta_k) = 0$. The behaviour may be constructed from a set of constraints $\{c_{k1}, c_{k2}, ..., c_{kn}\}$ associated with the subsystem and the exterior behaviour of the component is the union of internal behaviours $c_k = c_{k1} \cup c_{k2} \cup .... \cup c_{kn}$ or for brevity, $c_k = \{c_{k1}, c_{k1}, ..., c_{kn}\}$. Following the generic component definition in Blanke et al. (2003, 2nd ed. 2006), the service $S^{(k)}$ offered by component k is to deliver *produced variables* (output), based on *consumed variables* (input) and available *resources*, according to the specified behaviour $S^{(k)}_{(v)}$ where $v \in \{1, 2, 3, ..\}$ is the version of the service. Clearly, the exterior behaviour is associated with the service offered by the component, we denote this behaviour by $c^{(v)}_k$.

In this context we particularly wish to consider versions of the same service that follow from the condition of the component, from normal over degraded to none. If a component has an internal failure, fault-tolerant techniques may still provide a version of the service with degraded performance $\left(S^{(k)}_{(d)}\right)$ or the service may not be available at all $\left(S^{(k)}_{(o)}\right)$. Hence, we consider the set of versions $v \in \{n, d1, d2, .., o\}$ where $n : normal; d1 : degraded1; d2 : degraded2; o : none$.

2.3 Service at system level

The service obtained by the system as an entirety is a function of the component architecture $\mathcal{A}$ and the versions for the present condition k_i of components. With m components in a system, each component in one out of p conditions, $k_i \in \mathbb{N}^{p_i}$, we have a versions vector $\mathbf{v} = [v_1(k_1), v_2(k_2), ..., v_m(k_m)]$, and the set of available behaviours $C_v = \left\{c_1^{v(k_1)}, c_2^{v(k_2)}, ..., c_m^{v(k_m)}\right\}$.

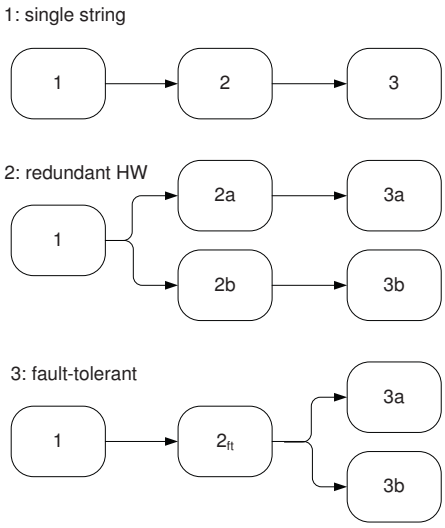

Fig. 1. Three architectures, single line with no redundancy (1), hardware redundancy (2) and combined fault-tolerance and redundancy (3)

Definition 1. The overall service available from a system is $S^{(s)}(c_i^{v(k_i)}) = \mathcal{A}(C_v|\mathbf{v}(\mathbf{k}))$, $i = 1, ..., m$.

With a single string architecture from Fig. 1, we obtain

$$S^{(s)} = S^{(1)} \cap S^{(2)} \cap S^{(3)} \tag{1}$$

With redundancy in the system, the hardware configuration with two parallel, totally redundant lines with only one component in common,

$$S^{(s)} = S^{(1)} \cap \left(\left(S^{(2a)} \cap S^{(3a)}\right) \cup \left(S^{(2b)} \cap S^{(3b)}\right)\right) \tag{2}$$

This solution is expensive as it requires two completely redundant subsystems. A cost effective solution would be to have some components intrinsically safe $S^{(1)}$, have others equipped with fault-tolerant properties so their service $S^{(2)}_{v(2)}$ will be available but in degraded version when local faults occur, and just have hardware redundancy for few essential components $(3a, 3b)$. The fault tolerant architecture shown in part C of Fig. 1 is based on this idea. The service at system level is

$$S^{(s)} = S^{(1)} \cap S^{(2)}_{v(2)} \cap \left(S^{(3a)} \cup S^{(3b)}\right) \tag{3}$$

The paradigm in this architecture is that component failures should be detectable and control be switched to obtain a fault-tolerant service or reconfigure the system bypassing faulty components. This should be achieved by controlling the signal flow in the software of the system.

2.4 Availability and safety

The plant at the system level is *available* as long as the predefined normal service is offered in some version, normal or degraded. A fault-tolerant version of the service is obtained when one or more of the component services are offered in a fault-tolerant version. A fail-operational version of the service is obtained when hardware reconfiguration has been made to bypass a failure in the redundant component.

When multiple local failures are present, the service at system level is

$$S^{(s)} = \mathcal{A}\left(C_v|\mathbf{v}\right)$$

Definition 2. Available. The system is available when $S^{(s)} \subseteq \mathcal{S}$ where $\mathcal{S} = \left\{S_1^{(s)}, S_2^{(s)}, ..., S_n^{(s)}\right\}$ is the set of admissible services that meet specified overall objectives for behaviour $\mathcal{O}$ of the system:$\forall S_i^{(s)}(c_i^{v(k_i)}) : C_v \subseteq \mathcal{O}$.

Definition 3. Fault. A fault is a deviation from normal behaviour, $\exists i : c_i \neq 0$.

Definition 4. Critical fault. A fault in c_i is critical $c_i \in C_{crit}$ if it will cause the system's behaviour to be outside the set of admissible behaviours, $c_i \in C_{crit} \;\; iff \;\; c_i \neq 0 \Rightarrow C_v \nsubseteq \mathcal{O}$.

Definition 5. Useability for reconfiguration. A faulty system is usable for reconfiguration, from the structure point of view, if all critical faults are structurally detectable, $\forall c_i \in C_{crit} : c_i \in C_{\det ectable}$

Assumption 6. It is a natural assumption that shut-down of the system is intrinsically safe and that the system can be shut down to a safe mode from any condition where $S^{(s)} \subseteq \mathcal{S}$.

Definition 7. Structural reconfigurability. A system is structurally reconfigurable if $c_i \neq 0 \Rightarrow \exists j \neq i, \mathbf{v}(j) \neq \mathbf{v}(i) : C_{v(j)} \subseteq \mathcal{O}$

The task of fault-tolerant control is to find an appropriate $\mathbf{v}(j)$ when the fault c_i is detected and isolated and bring the system from version $v(i)$ to $v(j)$.

Having defined the system properties in terms of behaviours, it is natural to employ structural analysis where behaviours are defined in terms of constraints between variables and graph theory methods offer rapid and rigourous analysis.

2.5 Structure graph

A structural model of a system can be represented as a bipartite graph that connects constraints and variables. The structure graph Staroswiecki and Declerck (1989) of a system (C, Z) is a bipartite graph $G = (C, Z, E)$ with two set of vertices whose set of edges $E \subseteq C \times Z$ is defined by $(c_i, z_i) \in E$ iff the variable z_i appears in constraint c_i.

The variables in Z are divided into known K and unknown variables X. Similarly, the constraints C are divided into constraints C_K that only apply to the known variables and C_X that involve at least one unknown variable. An incidence matrix S describes the structure graph where each row in the matrix represents a constraint and each column a variable. $S(i, j) = 1$ means that variable x_j appears in constraint c_i, $S(i, j) = x$ denotes a directed connection.

2.6 Constraints

Constraints represent the functional relations in the system, i.e. originating in a physical model using first principles. The constraints needed for structural analysis are

far more simple. Instead of using the explicit system equations, structural analysis need to know whether a certain constraint makes use of a particular variable. Parameters that are known from the physics of the plant or from properties of the automation system, e.g. a control gain, are treated as part of the constraint in which the particular parameter is used. A constraint can be directed. This implies that a variable on the left hand side of the constraint can not be calculated from the right hand side of the constraint.

2.7 Variables

There are three different kinds of variables: *Input variables* are known, externally defined; *Measured variables* are entities measured in the system; *Unknown variables* are internal physical variables. Input and measured variables both belong to the set K but are separated for calculation of controllability.

2.8 Matching and results

The central idea in the structure graph approach is to match all unknown variables using available constraints and known variables, if possible. If successful, the matching will identify over-determined subgraphs that can be used as analytical redundancy relations in the system.

Results of the structural analysis are

- List of parity relations that exist
- Auto-generated suggestion of residual generators
- List of detectable faults
- List of isolable faults

When a matching has been found, backtracking to known variables will give a suggestion for parity relations that could be used as residual generators. A system with m constraints and n parity relations will give a relation showing which residuals depend on which constraints.

One view on these relations is the boolean mapping,

$$\mathcal{F} : r \leftarrow M \otimes (c_i \neq 0) \tag{4}$$

from which structural detectability and isolability can be found.

Definition 8. A fault is structurally detectable iff it has a nonzero boolean signature in the residual, $c_i \in C_{\mathrm{det}\,ectable}$ iff $\exists j : c_i \neq 0 \Rightarrow r_j \neq 0$

Definition 9. A fault is structurally isolable iff it has a unique signature in the residual vector, i.e. column m_i of M is independent of all other columns in M, $c_i \in C_{isolable}$ iff $\forall j \neq i : m_i \neq m_j$

2.9 Active isolation

In some cases faults are group-wise isolable, i.e within the group individual faults are detectable but not isolable. This implies that with the given architecture of the system, these faults are group-wise not isolable. This does not necessarily imply that isolation can not be achieved in other ways. Indeed, although the same set of residuals will be "fired" when either one or the other of non-structurally isolable constraints is faulty, the time response of the residuals may be different under the different fault cases. Exciting the system with an input signal perturbation may therefore make it possible to discriminate different responses of the same residual set when different constraints within the group are faulty. The analytical idea of applying test signals to isolate faults is not new. Zhang (1989) designed test signals for diagnosis. A sophisticated setup for active diagnosis was made in Niemann (2006) and controller switching made active diagnosis in Poulsen and Niemann (2008). The structural analysis approach was first suggested in Blanke and Staroswiecki (2006) and extended to different use modes in Laursen et al. (2008).

Proposition 10. Active structural isolation is possible if and only if both a structural condition and a quantitative condition are true.
Structural condition : the known variables in the set of residuals associated with a group of non-structurally isolable constraints include at least one control input.
Quantitative condition : the transfer from control inputs to residuals is affected differently by faults on different constraints.

Definition 11. Input to output reachability. Let $p^{(i,j)} = \{c_f, c_g, ..., c_h\}$ be a path through the structure graph from input u_j output y_j and $\prod^{(i,j)}$ the union of valid paths from u_j output y_j. Let $C^{(i,j)}_{reach} = \left\{ c_g \mid c_g \in \prod^{(i,j)} \right\}$. A constraint c_h is input reachable from input u_j if a path exists from u_j to any output y_k and the path includes the constraint, $c_h \in C^{(i,k)}_{reach}$

Proposition 12. Active structural isolability from output signature. Two constraints c_g and c_h are actively isolable from output signatures if $\exists i, j, k, l : c_g \in C^{(i,j)}_{reach}, c_h \in C^{(k,l)}_{reach}$ and $\{c_g, c_h\} \notin C^{(i,j)}_{reach} \cap C^{(k,l)}_{reach}$

2.10 Scenarios with multiple faults

Scenarios of multiple faults are dealt with, in the structural analysis context, by removing one or more constraints that represent the faulty parts of the system. Should c_6 be subject to a local failure, the remaining system $\mathbf{S}_f = \mathbf{S} \setminus \{c_6\}$ need be re-analysed. The results can show which residual generators exist for the faulty system, and which further faults could be isolated or detected. An application to a marine control system was treated in Blanke (2005) where analysis of multiple faults was a part of a fault-tolerant design.

Following the formal introduction to methodologies, the use of graph-based techniques is illustrated by two real-life cases.

3. AEROPLANE DIAGNOSIS AND FAULT HANDLING

A case of diagnosis and fault handling of speed sensor faults on a small Unmanned Aerial Vehicle (UAV) was studied in Hansen et al. (2010). Focusing on faults in the pitot tube, that easily causes a crash if not diagnosed and handled in time, redundant information about airplane velocity, residuals are easily generated that allow both isolation and handling of a single sensor failure. Recorded

telemetry data of an actual event with a pitot tube defect illustrate the efficacy of the diagnostics.

3.1 Airspeed Sensor Problem

Defects on sensors can have catastrophic consequences for airplanes, specially smaller airplanes, which do not have the same sensor redundancy that is available on larger aircraft. It is therefore important to be able to detect whether a sensor defect has occurred. One of the most important sensors is the pitot tube that measures airspeed of the vehicle. This sensor is very exposed because of its position in the airstream and can easy be clogged by dust or water particles that freeze at higher altitudes.

The solution to these clogging problems usually employed on larger aircrafts is to install several pitot tubes with build in heating devices to have a redundant system that can accommodate icing. Because of weight and space limitations, adding more sensors is usually not an option on smaller unmanned aerial vehicles (UAV). Therefore, a different approach must be taken to diagnose and accommodate faults. One way could be to have artifact readings detected and replaced with estimated values. Detection of faults and fault-tolerance for UAVs has a lot of focus and, as described in Ducard (2009), many parts of the aircraft control and operation can benefit from using fault tolerant methods. A systematic approach to fault detection is described in Fravolini et al. (2009) and some of the applications of these methods are, detection of mechanical defects, like stuck control surfaces. These were studied in Bateman et al. (2009) and Park et al. (2009) where active methods were used to isolate faults. Observer based fault diagnosis was investigated in e.g. Heredia et al. (2008). Nonlinear models that describe the aircraft can also be used in fault diagnosis, this was demonstrated on small helicopters in Freddi et al. (2009).

3.2 Model for Diagnosis

An airplane is modelled by the dynamic and kinematic equations, which describe its motion through the air, and very detailed models are available. With nonlinear terms from gyroscopic couplings and others, and nonlinear parameters cause by aerodynamic effects, the equations of motion are inherently nonlinear. Several textbooks, including Stevens and Lewis (2003) describe the detailed models.

A formulation of the generic behaviours for an airplane, by Fravolini et al. (2009), showed that the structural analysis approach is also well suited for this problem. The behavioural formulation reduces the complexity to precisely what is required for diagnosis and the details of particular nonlinearities need be scrutinised only when needed for particular residuals. Focusing on the speed sensor fault diagnosis, a simplified behaviour description, related only to vehicle speed, is shown in Table 1 where $\mathbf{v}_n$ is speed over ground (navigation frame), v_t is air speed through propeller disc if the propeller was not present, $\mathbf{v}_a$ is air speed vector,$\mathbf{v}_w$, the wind velocity vector, v_p pitot tube measured air speed, $\mathbf{v}_g$ velocity estimated by the GPS.

Table 1. Velocity Related Behaviours

constraint				behaviour
c_1	:	$\mathbf{v}_n$	=	$\mathbf{v}_a + \mathbf{v}_w$
c_2	:	$\hat{v}_t$	=	$g_1(n, \lvert\mathbf{v}_a\rvert, \theta)$
c_3	:	$\hat{v}_t$	=	$\lvert\mathbf{v}_a\rvert$
c_4	:	$\hat{\mathbf{v}}_w$	=	$g_2(\mathbf{v}_n, \mathbf{v}_a)$
c_5	:	$\hat{\mathbf{v}}_w$	=	$\mathbf{v}_w$
m_1	:	$\mathbf{v}_g$	=	$\mathbf{v}_n$
m_2	:	v_p	=	$\lvert\mathbf{v}_a\rvert$

Table 2. Graph for Aeroplane Velocity Case.

	known				unknown				
	n	$\mathbf{v}_g$	v_p	θ	$\hat{v}_t$	$\hat{v}_w$	$\mathbf{v}_n$	$\mathbf{v}_a$	$\mathbf{v}_w$
c_1							1	1	1
c_2	-1			-1	1			-1	
c_3					1			-1	
c_4						1	-1	-1	
c_5						1			1
m_1		1					1		
m_2			1					-1	

In Table 1, the function g_1 is an estimator of air speed $\mathbf{v}_a$ using the propeller thrust relations, where n is known rotational speed, g is gravity acceleration constant, m i mass of the airplane, T_{nn} and T_{nv} are parameters determined from propeller characteristics, and F_A is aerodynamic drag,

$$g_1 = (T_{nv} n)^{-1} \left(T_{nn} n^2 - F_A(v_a, \theta) - mg \sin(\theta)\right) \tag{5}$$

The function g_2 is an estimator that provides a fairly uncertain estimate of the wind velocity vector.

The sets $\mathcal{K}$ and $\mathcal{X}$ of known and unknown variables in this problem are,

$$\begin{aligned} \mathcal{K} &= \{n, \mathbf{v}_g, v_p, \theta\} \\ \mathcal{X} &= \{\hat{v}_t, \hat{\mathbf{v}}_w, \mathbf{v}_n, \mathbf{v}_a, \mathbf{v}_w\} \end{aligned} \tag{6}$$

With 5 unknowns and 7 constraints, there are up to 2 unmatched constraints, that can be used as parity relations. When solving the set of matched constraints in Table 1, calculability need be accounted for since a vector length can be determined from the components of the vector but not reversely. A representation of the structure graph associated with the constraints are shown in form of the incidence matrix in Table 2. A 1 in the matrix shows that the associated variable can be calculated from the constraint, a -1 that it can not.

A complete matching on the unknown variables is achieved as follows:

$$c_1 \rightarrow \mathbf{v}_a; c_4 \rightarrow \hat{\mathbf{v}}_w; c_5 \rightarrow \mathbf{v}_w; m_1 \rightarrow \mathbf{v}_n; c_3 \rightarrow \hat{\mathbf{v}}_t; \tag{7}$$

The set of unmatched constraints are $\{c_2, m_2\}$. Backtracking to known variables along the matching leads to a symbolic form of the residuals and insertion of the analytical constraints from Table 1 gives the residuals,

$$R_1 = \lvert\mathbf{v}_g - \hat{\mathbf{v}}_w\rvert - v_p = \lvert\mathbf{v}_g - g_2(\mathbf{v}_n, \mathbf{v}_a)\rvert - v_p \tag{8}$$

$$R_2 = \hat{v}_t - v_p = g_1(n, \lvert v_p\rvert, \theta) - v_p \tag{9}$$

This result is quite intuitive but the formal procedure assures that all possible redundancies have been explored.

While the simple estimate g_1 was found to suffice for the purpose of fault diagnosis by Hansen et al. (2010),

more sophisticated and precise nonlinear estimators have been suggested in the literature. Zhou and Blanke (1989) described a way to estimate state and parameters in nonlinear systems with a structure similar to the thrust equation here, Blanke et al. (1998) applied an adaptive observer scheme and Pivano et al. (2009) showed nonlinear observer designs for thrust estimation.

3.3 Signal analysis

Figure 2 shows a time history and a histogram for R_1 and R_2 from UAV flight data in the fault free case.

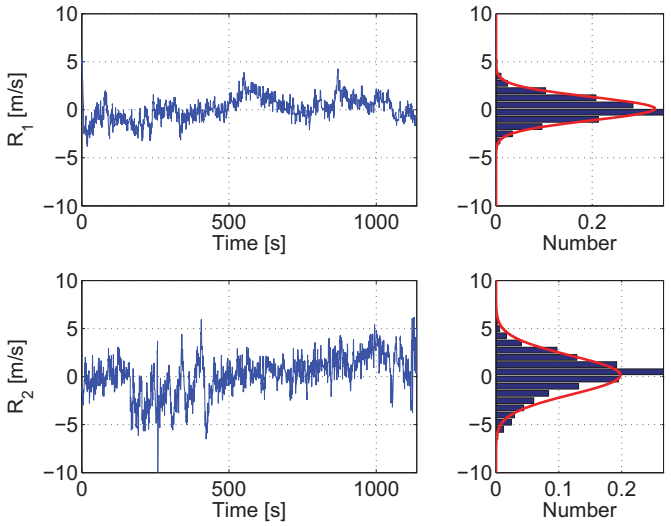

Fig. 2. Time development and histogram for residual R_1 and R_2 in the fault free case.

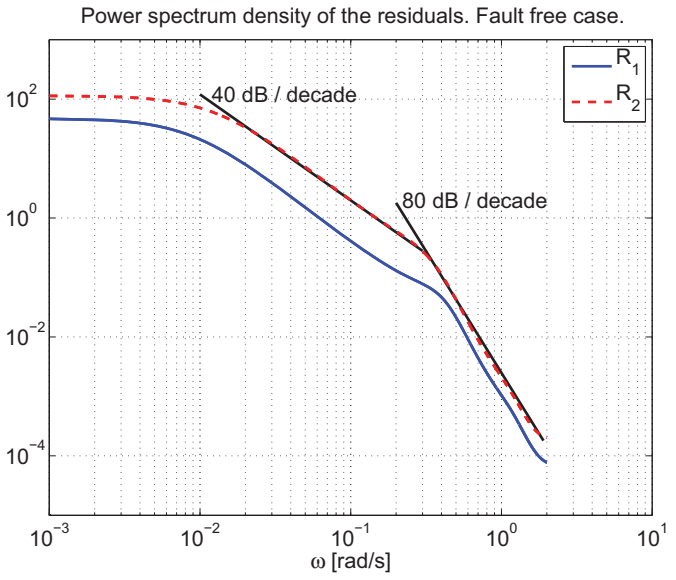

Fig. 3. Power spectrum densities for residual R_1 and R_2.

The power spectral densities of the residual signals are not white, however. Since white noise is one of the requirements for most statistical change detectors to perform optimal, the low-pass filtered nature of the noise should be removed. One solution of dealing with coloured noise is to filter white noise through a suitable filter function to take into account the correlations present in the coloured noise. This can be created from a large record of data where all the signals properties are present. As indicated on figure 3 the power spectrum density of the two residuals consists of a part which decreases with 40 dB/decade and a part which decreases with 80 dB/decade. A whitening filter can be implemented as any stable filter, including as a Kalman filter.

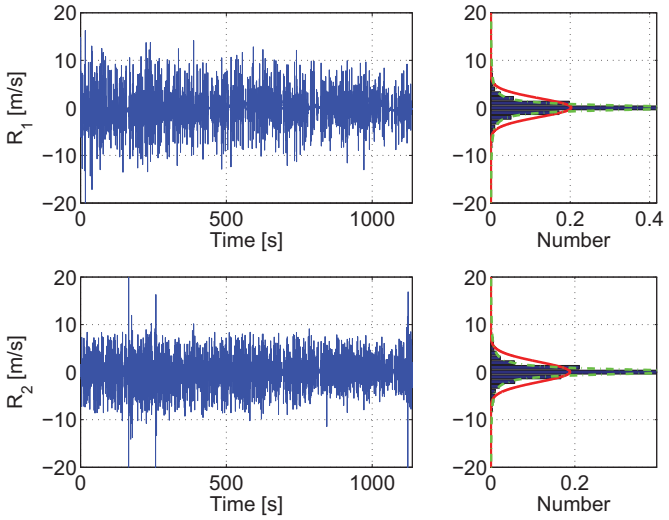

Fig. 4. Time development and histogram for the whitened residual R_1 and R_2 in the fault free case.

Figure 4 shows time series of the whitened residuals and their histograms. Neither of the residuals are now Gaussian distributed. Instead they follow the Cauchy distribution (equation 11) very well as indicated by the green dotted line in the histogram. The Gaussian nature that the residuals had before whitening apparently came from the effects of low-pass filtering. When removed during the whitening process, the Gaussianity was lost.

3.4 Change Detection

Detection of and unknown change of mean value A of a signal x should distinguishing between two hypotheses,

$$\begin{aligned} \mathcal{H}_0 &: x[n] = w[n] \\ \mathcal{H}_1 &: x[n] = A + w[n] \end{aligned} \tag{10}$$

The variance of the noise w is σ_w^2. The GLRT (Generalised Likelihood Ratio Test) is a standard way to solve such a problem. The standard solution assumed Gaussian noise. When the residuals are distributed according to a Cauchy distribution (as seen from figure 4) the following probability distribution function must be used

$$p(x; x_o, \beta) = \frac{\beta}{\pi \left(x - x_o\right)^2 + \beta^2} \tag{11}$$

where the two parameters are the half-width half-maximum scaling, β, and the offset x_o. The GLRT test statistic becomes

$$L_G(\mathbf{x}) = \frac{\prod_{i=1}^{N} p\left(x_i; \hat{x_o}, \hat{\beta}\right)}{\prod_{i=1}^{N} p\left(x_i; 0, \hat{\beta}\right)} > \gamma_c \tag{12}$$

The maximum likelihood estimate (MLE) of $\hat{\beta}$ and $\hat{x_o}$ is found by fitting the data to equation 11.

Detection of change according to Eq. 12 requires that the threshold γ is determined. If the distribution of $L_G(\mathbf{x}|\mathcal{H}_o)$ is known, the probability of false alarm is:

$$P_{FA}(\mathbf{x}|\mathcal{H}_0, \gamma) = \int_{\gamma}^{\infty} p_{L_G}(x|\mathcal{H}_0)dx \qquad (13)$$

The probability density (pdf) and the cumulative density (cdf) for $L_G(\mathcal{H}_0)$ are shown using flight data in Fig. 5.

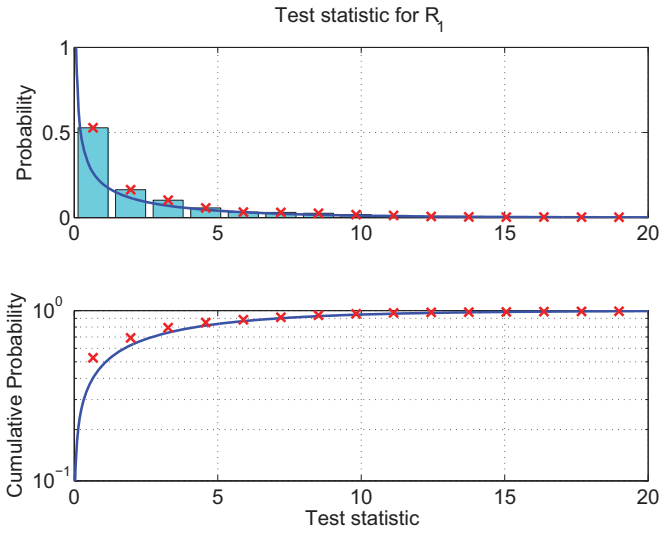

Fig. 5. Histogram and cumulative density for the $L_G(\mathcal{H}_0)$ test statistics for the GLRT Cauchy detector. Data and approximating function.

The approximating function to the CDF in the Figure is a Gamma distribution.

$$G(x; a, b) = \frac{1}{b^a \Gamma(a)} x^{a-1} \exp\left(-\frac{x}{b}\right) \qquad x > 0 \qquad (14)$$

where Γ denotes the Gamma function. The distribution is fitted with a MLE and the following parameters is found

	a	b
R_1	0.46	5.58
R_2	0.43	4.91

Selecting a threshold of $\gamma = 50$ gives a probability of false alarm of $1 - G(50; 0.46, 5.58) \cong 2.0\ 10^{-5}$ for R_1.

The theoretical performance could also be calculated using formulas from Hansen et al. (2010) but since theory and practice sometimes differ, it is advisable to check one's threshold value using an observed CDF from the test statistics of real data. This is particularly the case when the assumption of independent and identically distributed (*iid*) samples is not valid. The threshold required to achieve a certain false alarm probability can be very much different from its theoretical value Kay (1998) when the *iid* assumption is not valid.

3.5 Results

To test the detection performance, data from a real event are used where a UAV crashed, caused by a pitot tube defect and propagated effects on the flight control. The detection results are shown together with the residuals in Figure 6.

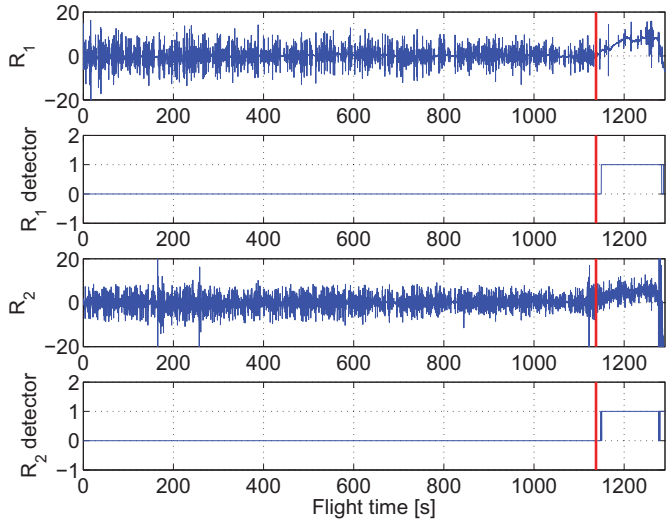

Fig. 6. Residuals and detector output. At approximately $t = 1140$ s the clogging of the pitot tube occurs (marked with the red vertical line).

As seen an alarm is raised, indicated by a value of 1, about 14 seconds after assumed instant when the incident started. Timely detection is hence obtained and with yet another 120 seconds elapsing until the crash happened, there would have been ample time to execute remedial actions had this detection system been available.

The simple detection methods in this example are sufficient to avoid simple accidents with simple equipped UAV's. Pitot tube defects have happened also on larger aircraft despite redundancy to a fail-operational level and the fault-diagnosis techniques could be part of fault-tolerant solutions and have a wider area of application.

4. FAULT-TOLERANT GUIDANCE USING VISION FOR AGRICULTURE

This second case case from agriculture brings diagnosis and fault-tolerant control techniques in operation within advanced computer vision based sensors. In agriculture, typical tasks are to follow structures in the field to plow, seed, spray or harvest. The specific harvesting task of baling involves to follow rows of cut straw or grass (swath) in order to pick it up and process it into bales. This is a labour intensive and repetitive task, which is of interest to automate. The difficulties pertaining to automating this task are similar to the difficulties in automating a large range of agricultural tasks. The ability to track this structure using 3D shape information from a stereo camera and/or GPS information was demonstrated in Blas et al. (2009) and a detailed presentation of the baling problem was presented in Blas and Blanke (2011) where a classifier was employed based on online learning of texture information about the swath and the surroundings. This was then coupled with shape information to extract the swath position and a mapping kept track of measured swath positions. The map was used to guide the vehicle

along the swath by steering the tractor's front wheels while a driver controlled the throttle and brakes. The novelty with respect to diagnosis in this application is the use of diagnostic techniques to obtain fault tolerance in the stereo vision sensor itself, avoiding typical reasons for artifacts that occur in the stereo vision process when distance to objects are calculated. Highlights from this real-life case are given below.

This case is dealing with an agricultural vehicle with the task to follow cut grass (swath) in order to pick it up with a baler (see fig. 7). The system to be analysed is equipped with stereo-vision, a single antenna GPS and an IMU, in one configuration. GPS positions of the vehicle that formed the swath are known. The combination of stereo-vision and GPS allows the system both to "see" the swath but also navigate based on the given map. This creates system redundancy that is essential for achieving fault-tolerance. A visual odometry algorithm on the stereo-camera allows for the relative position of the vehicle without GPS or IMU. The GPS receiver used was ground-station compensated and the IMU was a tactical grade (low accuracy) MEMS based unit.

The two main ideas presented is first a behavioural model for representing the natural environment, namely the swath. Secondly, it is shown how parts of this model (the swath location) can be used in conjunction with sensor inputs to create a fault tolerant sensor fusion system. The fault diagnosis is illustrated using real data.

Fig. 7. Swath in the field to be collected.

4.1 Modelling the Natural Environment

A model of the swath requires extracting the salient features of the environment required for the field operation and storing them in the model representation. The salient features are the location of the swath and the distribution of the swath material across the swath. Swath location, width, and height are modelled as a map using splines to follow the centre of the swath in a metric map. The swath height is represented by a grid map of resolution 2 cm for each grid point. The model is illustrated in fig. 8.

The swath location is defined as being in a 2D coordinate system on the ground plane. A function f represents the lines down the middle of the swaths. Given coordinate pairs (x, y) then f is:

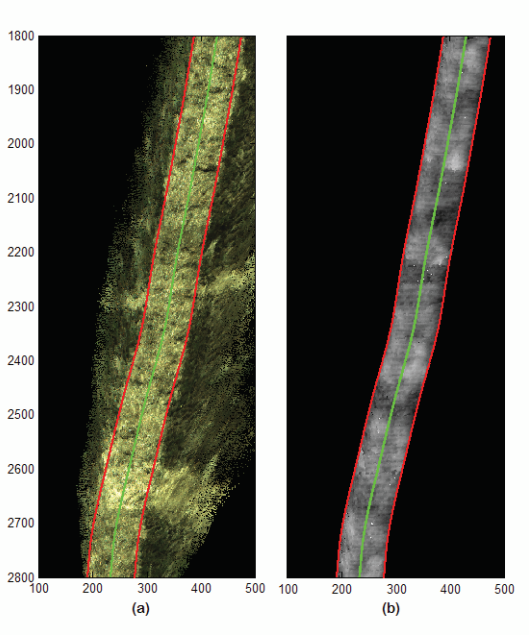

Fig. 8. (a) RGB topdown-view of the swath with swath location and estimated swath width superimposed. (b) The swath model showing both swath location, width and height by image feature extraction.

$$y = f(x) \tag{15}$$

The model of the swath location is then $s(x)$ with $s \in \mathcal{S}_3(\mathbf{k}_{0:n})$, where $\mathbf{k}_{0:n}$ are the spline knots and the spline coefficients and $\mathcal{S}_3$ is the cubic spline domain. Then the model is equal to the swath location plus the approximation error ϵ_a of fitting a spline to f:

$$s(x) = f(x) + \epsilon_a \tag{16}$$

Based upon the concept of having a controller that allows the vehicle to follow the swath location, the x-track error ε_x (signed shortest distance from the control point to the spline) can be found as a function of the tractor position and the spline. Defining the spline s_b in body coordinates a function $\mathcal{E}$ can be set to find the x-track error:

$$\varepsilon_x = \mathcal{E}(s_b) \tag{17}$$

4.2 Stereo-Camera

A stereo algorithm is used to find the correspondence between features in the left and right image sensor (i_l, i_r). The position of the features relative to the stereo-camera can then be inferred in 3D. Modern vision algorithms then exist to register 3D models with the 3D point cloud provided by the stereo-camera: Goshtasby (2005). An algorithm has been constructed that allows such registration between the swath model and the 3D points. The stereo-algorithm and registration will be denoted by the function $\mathcal{V}_{reg}$. Thus, given the two images a measurement of the swath location s_c, width g_c, and height h_c can be computed for the part of the swath in the image.

$$[s_c, g_c, h_c]^T = \mathcal{V}_{reg}(i_l, i_r) \tag{18}$$

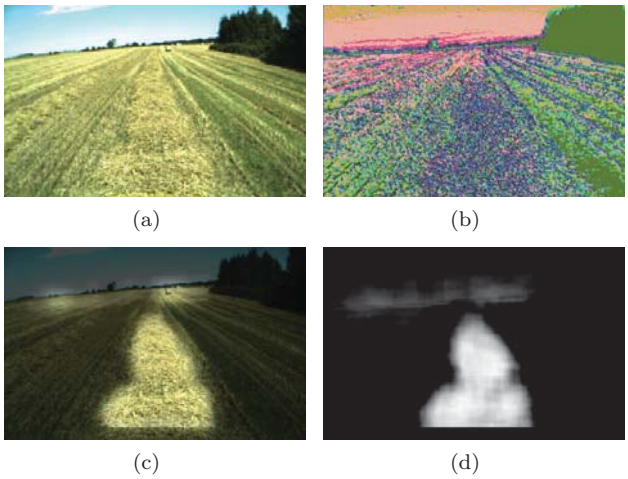

Fig. 9. (a) Left image from the stereo camera. (b) Texton classification. Pixel colour shows class of texton. (c) The stereo camera image with a transparency mask based on the swath classification. (d) Swath classification based on texture with intensity representing the strength of classification.

These measurements are stored in a map representation for an individual swath and s_m is the spline formed by combining N previous measurements,

$$\mathbf{s}_m(k) = \min_{spline} (s_c(j), j = k - N, k) \tag{19}$$

4.3 Robust Stereo Enhancement by Texture

Stereo vision uses correlation of textures to determine distance to objects in view of both cameras. When misclassification of texture happens, gross errors may occur in the stereo calculated distance to objects or artifact objects can occur or objects can disappear. To robustify the stereo against such artifacts, Blas and Blanke (2011) introduced statistical validation of texture using so called *textons*. Texture properties can include colour distribution, intensity, shape and patterns. Textons are basis vectors extracted from the local descriptors of texture. Following Varma and Zisserman (2003), who showed that small local texture neighbourhoods may be better than using large filter banks, Blas et al. (2010) and Blas and Blanke (2011) employed statistical classification to the problem and used the texture-based classification of swath as a supplement to the geometrical mapping described above.

4.4 Texton Labelling

Given a colour image as input, pixel neighbourhoods in the image are grouped into belonging to a texton type, which belongs to a set of basis textons υ_i obtained by prior learning from a training image. This is done by first extracting a descriptor in the form of a vector from each pixel location in the image. For each location the p_i vector is:

$$p_i = \begin{bmatrix} w_1 * l_c \\ w_2 * a_c \\ w_2 * b_c \\ w_3 * (l_1 - l_c) \\ \vdots \\ w_3 * (l_8 - l_c), \end{bmatrix} \tag{20}$$

where $[l_c, a_c, b_c]$ is the colour of the pixel at this location in CIE*LAB colour-space. $(l_1 - l_c), ..., (l_8 - l_c)$ are the intensity differences between the pixel at this location and the 8 surrounding pixels in a 3×3 neighbourhood. The vector elements are then weighted using $\{w_1 = 0.5, w_2 = 1, w_3 = 0.5\}$. A K-means algorithm is then run on all these descriptors to extract cluster centres which we refer to as textons. The fast K-means algorithm (Jain and Dubes (1988)) was used in Blas and Blanke (2011) to find the set of textons υ_j that partitions the descriptors into κ sets $\mathcal{T} = \mathcal{T}_0, \mathcal{T}_1, ..., \mathcal{T}_\kappa$ by trying to minimise:

$$\mathcal{T}_{reg} = \underset{\mathcal{T}}{\operatorname{argmin}} \sum_{j=1}^{\kappa} \sum_{p_i \in \mathcal{T}_j} \|p_i - \upsilon_j\|^2 \tag{21}$$

As a final step, processing of all labelled pixels in an image at time k gives an estimate of swath density using the textrons that are classified as belonging to the swath Υ_{swath},

$$s_t(k) = \max_{density} \mathcal{T}_{reg}(p_i \in \Upsilon_{swath}, i = k - N \dots k) \tag{22}$$

Each pixel location in the image is then labelled by finding the nearest texton in Euclidean space. An example of the result of such a classification is shown in Fig.9.

One way of abstraction of the texture processing could be a spline function describing the texture-classified swath,

$$\mathbf{s}_t(k) = \min_{spline} (s_t(j), j = k - N, k) \tag{23}$$

Another use of the texton classification could be to robustify the stereo estimate 18 by,

$$[s_{ct}, g_{ct}, h_{ct}]^T = \mathcal{V}_{reg}(i_l, i_r, \mathcal{T}_{reg}). \tag{24}$$

Using the *intelligent sensor* capability of Eq. 24 represents a fault-tolerant processing within the stereo algorithms that could be used after learning of the set Υ_{swath} has been obtained. This is an alternative to using Eqs. 23 and 19 separately, but the latter could have benefits during unsupervised learning.

4.5 Structural Model

The constraints in this case describe the structured natural environment of the field with swath, the baler and the available sensors. Let the constraints be composed of those from measurements (m), differential (d), and the "system" constraints (c). Using $\mathbf{v}^b$ for visual odometry measured velocity vector over ground seen in body coordinates; $\mathbf{a}^b$ for IMU measured acceleration vector; $\mathbf{p}^n$ the position in (North, East) coordinates with $\mathbf{p}_1^n$ being the position measurement from the GPS; $\mathbf{R}_b^n$ is the rotation matrix from body to navigation frame, which is a function of Θ, the attitude vector (Euler angles roll, pitch and yaw) and of λ, the latitude;. s^b is the swath position spline seen

in body coordinates. In this analysis, the $\mathbf{R}_b^n$ matrix is assumed to be known.

With variables defined above, the sets of known and unknown variables are,

$$\begin{aligned}\mathcal{K} &= \{\mathbf{v}^b, \mathbf{a}^b, \mathbf{p}_1^n, \mathbf{s}_g, \mathbf{s}_c, , \mathbf{s}_m, \mathbf{s}_t, \mathbf{R}_b^n(\Theta,\lambda)\} \\ \mathcal{X} &= \{\mathbf{p}^n, \dot{\mathbf{p}}^n, s^b, s, \varepsilon_x\} \end{aligned} \quad (25)$$

The constraints are summarised in Table 3.

Table 3. Behaviours for Baler Control Example

c_1	:	s^b	=	$\mathbf{R}_b^n(\Theta,\lambda)s + \mathbf{p}^n$
c_2	:	ε_x	=	$\mathcal{E}(s^b)$
d_1	:	$\dot{\mathbf{p}}^n$	=	$\frac{d}{dt}\mathbf{p}^n$
m_1	:	$\mathbf{v}^b$	=	$\mathbf{R}_b^n(\Theta,\lambda)\dot{\mathbf{p}}^n$
m_2	:	$\mathbf{a}^b$	=	$\frac{d}{dt}\mathbf{R}_b^n(\Theta,\lambda)\dot{\mathbf{p}}^n$
m_3	:	$\mathbf{p}_1^n$	=	$\mathbf{p}^n$
m_4	:	s_g	=	s
m_5	:	s_c	=	s^b
m_6	:	s_m	=	s^b
m_7	:	s_t	=	s^b

4.6 Structural Results

The number of constraints available depends on: the *use mode*, eg. m_7 is only available when the tractor is in the field with a swath and texture learning has been completed; the *configuration*, eg. whether an inertial measurement unit (IMU) is mounted; on *faults*, eg. failure of an instrument. Structural analysis under different use modes was presented in Laursen et al. (2008) and aspects of analysis under simultaneous faults was discussed in Blanke and Staroswiecki (2006). A structural analysis of the baler example, without the texture element, was presented in Blas and Blanke (2008). Analysing the constraints listed in Eq. 3, the following residuals are obtained,

$$\begin{aligned} r_1 &: \mathbf{v}^b - \mathbf{R}_b^n(\Theta,\lambda)\frac{d}{dt}\mathbf{p}_1^n = 0 \\ r_2 &: \frac{d}{dt}\mathbf{v}^b - \mathbf{a}^b = 0 \\ r_3 &: \mathcal{E}(s_c) - \mathcal{E}(\mathbf{R}_b^n(\Theta,\lambda)s_g + \mathbf{p}_1^n) = 0 \\ r_4 &: \mathcal{E}(s_c) - \mathcal{E}(s_m) = 0 \\ r_5 &: \mathcal{E}(s_m) - \mathcal{E}(s_t) = 0 \end{aligned} \quad (26)$$

4.7 Field Tests

The properties of residuals were investigated based on recorded data. The data stems from the test field run illustrated in fig. **??**. The position of the swath was first logged by following the middle of the swath manually - emulating the vehicle forming the swath. This was repeated for a second pass emulating the vehicle that should pick up the swath. This provides some form of limited ground truth. The position error of the driver is bounded between the runs as he constantly steers relative to the swath. Experience with driving with balers puts the error associated with not driving exactly over the center of the swath to under ± 0.2 m as this is required to pick up the swath successfully. In the data examined the GPS has a false offset in the second pass relative to the first pass of approximately 0.6 m for the first approx. 70 s before it corrects its position estimate to bring it to about 0.15 m of the swath location. This offset is acceptable for normal operation. Field tests enabled calculation of residuals r_1, r_3 and r_4 as an instrumentation issue prevented data reception from the IMU. The field test is hence representing a case of one permanent failure and an additional fault occurring.

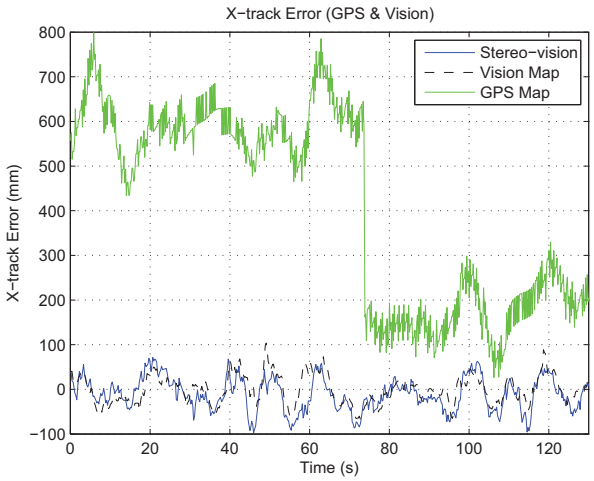

Fig. 10. The vehicle was driven manually over a swath. The driver centered the vehicle over the middle of the swath and drove for 2 min while maintaining this centered position. The x-track errors from the subsystems were recorded.

The driver interfaces to the control system through a terminal to change settings and engages/disengages the automatic steering system through a switch. The baler has integrated pressure sensors which are used to measure the bale diameter. This information is used in the controller to assure an even filling of the bale chamber. A wheel angle sensor provides feedback about the angle of the front wheels relative to the tractor. Hydraulics allow actuation of the front wheels.

4.8 Control

Tests were conducted with a tracking control system that was made to collect the swath if the match score was above a predefined threshold. The control system remains active as long as map information is available ahead of the vehicle. A main difficulty in baling is that the bale chamber must be filled evenly. Pressure sensors inside the baler provide a measure of how evenly it is filled. If the bale chamber is unevenly filled then the bale becomes cone shaped. The bale must have a certain size before the pressure sensors give usable feedback. To compensate for this lack of feedback the controller has two states. In the initial state where pressure has not yet built up an open-loop steering pattern is followed where the vehicle changes between being the left and right edges of the swath. The steering system changes mode when the bale size reaches a threshold and sensor feedback is used. The sensor feedback spans an interval from -1 to 1 indicating how cone shaped the bale is. This information and the measured lateral

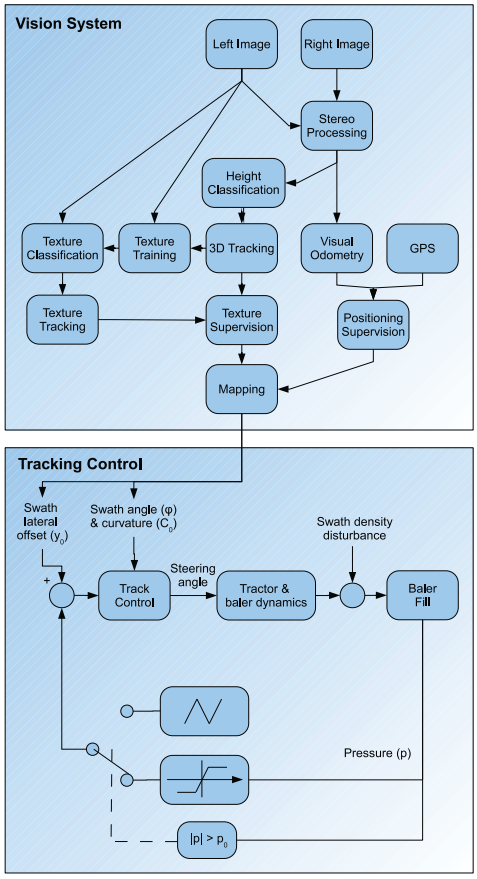

Fig. 11. Block diagram of vision system and tracking control for baling. Supervised classification and positioning provides mapping of field structures, which are fed to the steering controller.

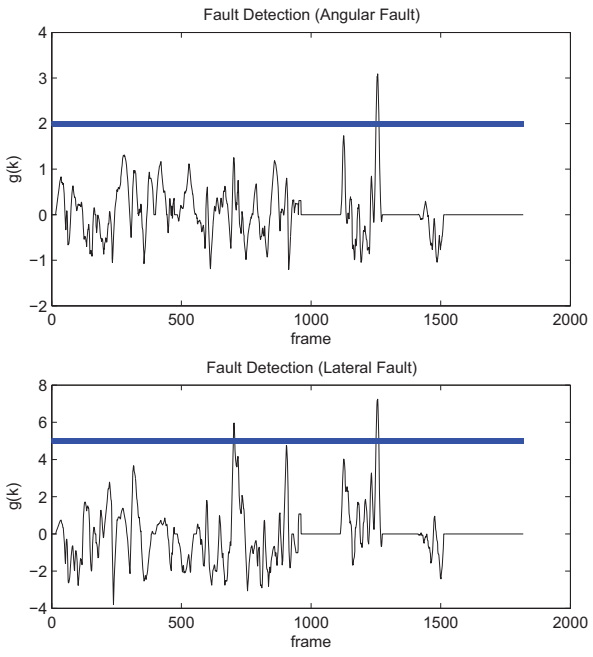

Fig. 12. Detection of deviation in angle or across the swath track using a CUSUM test. The thick line defines the threshold.

offset are used in the control. The swath parameters y_0, ϕ, C_0 are then fed to a generic curve tracking controller. Fault-tolerant control described in this example focused on the complex parts, ie. the natural environment. Extension with the baling control part would be simple.

Figure 12 shows detection of deviations in steering angle and cross-track based on a CUSUM detection. The statistical change detection is much more robust and has less nuisance from noise than any of the single measurements.

5. CONCLUSIONS

This paper has introduced to generic principles of diagnosis and fault-tolerant control based on the formal description of *behaviours* and *services*. The steps of system analysis, residual generation and change detection were discussed and exemplified through two large case studies. One dealt with a UAV sensor fault diagnosis and fault-handling, another with vision-based baling for agriculture. It was shown the the generic theoretic methods are indeed applicable to the complexity met in a natural environment, and the paper showed how such generalisation was obtained in both theory and in practice.

ACKNOWLEDGEMENTS

Research support from CLAAS Agrosystems, from the Joint Danish Forces Drone Team and the Danish Ministry for Science, Development and Innovation are gratefully acknowledged.

REFERENCES

Åstr0̈m, K. J., Albertos, P., Blanke, M., Isidori, A., Schaufelberger, W., Sanz, R. (Eds.), 2000. Control of Complex Systems. Springer.

Basseville, M., 1988. Detecting changes in signals and systems - a survey. Automatica 24 (3), 309–326.

Basseville, M., Nikiforov, I. V., 1993. Detection of Abrupt Changes: Theory and Application. Information and System Science. Prentice Hall, New York.

Bateman, F., Noura, H., Ouladsine, M., 2009. Fault tolerant control strategy based on the doa: Application to uav. In: 7th IFAC Symposium on Fault Detection Supervision and Safety of Technical Processes.

Blanke, M., 1996. Consistent design of dependable control systems. Control Engineering Practice 4 (9), 1305–1312.

Blanke, M., June 2005. Fault-tolerant sensor fusion with an application to ship navigation. In: Proc. IEEE Mediterranean Control Conference. pp. 1385–1390.

Blanke, M., Izadi-Zamanabadi, R., Bøgh, S. A., Lunau, C. P., 1997. Fault-tolerant control systems - a holistic view. Control Engineering Practice 5 (5), 693–702.

Blanke, M., Izadi-Zamanabadi, R., Lootsma, T., 1998. Fault monitoring and re-configurable control for a ship propulsion plant. Journal of Adaptive Control and Signal Processing 12, 671–688.

Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., 2003, 2nd ed. 2006. Diagnosis and Fault-tolerant Control, 2nd Edition. Springer-Verlag.

Blanke, M., Lorentzen, T., 2006. Satool - a software tool for structural analysis of complex automation systems. In: 6th IFAC Symposium on Fault Detection, Supervision

and Safety of Technical Processes SAFEPROCESS. Beijing.

Blanke, M., Staroswiecki, M., September 2006. Structural design of systems with safe behavior under single and multiple faults. In: 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes SAFEPROCESS. Beijing, P. R. China, pp. 511–516.

Blas, M. R., Blanke, M., 2008. Natural environment modeling and fault-diagnosis for automated agricultural vehicle. In: Proceedings 17th IFAC World Congress, Seoul, Korea. pp. 1590–1595.

Blas, M. R., Blanke, M., 2011. Automatic baling using stereo vision and texture learning. Computers and Electronics in Agriculture 75, 159–168.

Blas, M. R., Blanke, M., Madsen, T. E., March 2010. A method of detecting a structure in a field, a method of steering an agricultural vehicle and an agricultural vehicle. Patent Application, eP10157313.7.

Blas, M. R., Blanke, M., Rusu, R. B., Beetz, M., 2009. Fault-tolerant 3d mapping with application to an orchard robot. In: Proc. 7th IFAC International Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS'09). pp. 893–898.

Chen, J., Patton, R. J., 1996. Optimal filtering and robust fault-diagnosis of stochastic-systems with unknown disturbance. IEE Proceedings Control Theory and Applications 143 (1), 31–36.

Chen, J., Patton, R. J., 1999. Robust Model-based Fault Diagnosis for Dynamic Systems. Kluwer Academic Publishers.

Chow, E. Y., Willsky, A. S., 1984. Analytical redundancy and the design of robust failure detection systems. IEEE Transactions on Automatic Control AC-29 (7), 603–614.

de Persis, C., Isidori, A., 2001. A geometric approach to non-linear fault detection and isolation. IEEE Transactions on Automatic Control 45 (6), 853–865.

Ducard, G. J. J., 2009. Fault-tolerant Flight Control and Guidance Systems. Springer Verlag.

Düştegör, D., Cocquempot, V., Staroswiecki, M., 8-10 december 2004. Structural analysis for fault detection and identification: an algorithmic study. In: Proceedings of the 2^{nd} Symposium on System Structure and Control 2004 (SSSC'04). Oaxaca, Mexico.

Düştegör, D., Frisk, E., Cocquempot, V., Krysander, M., Staroswiecki, M., 2006. Structural analysis of fault isolability in the damadics benchmark. Control Engineering Practice 14 (6), 597–608.

Dulmage, A. L., Mendelsohn, N. S., 1959. A structure theory of bi-partite graphs. Trans. Royal Society of Canada. Sec. 3. 53, 1–13.

Dulmage, A. L., Mendelsohn, N. S., 1963. Two algorithms for bipartite graphs. Journal of the Society for Industrial and Applied Mathematics. (1), 183–194.

Dunia, R., Qin, S. J., Edgar, T. F., McAvoy, T. J., 1996. Identification of faulty sensors using principal component analysis. AIChE J 42, 2797–2812.

Emami-Naeini, A., Akhter, M., Rock, S. M., 1988. Effect on model uncertainty on failure detection ; the threshold selector. IEEE Transactions on Automatic Control, Vol AC-33 no.12, 1106–1115.

Frank, P. M., 1990. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy. Automatica 26 (3), 459–474.

Frank, P. M., 1994. On-line fault detection in uncertain nonlinear systems using diagnosticobservers; a survey. Int. Journal of Systems Science 25 (12), 2129–2154.

Frank, P. M., Ding, X., 1994. Frequency domain approach to optimally robust residual generation and evaluation for model-based fault diagnosis. Automatica 30 (5), 789–804.

Fravolini, M. L., Brunori, V., Campa, G., Napolitano, M. R., La Cava, M., 2009. Structured analysis approach for the generation of structured residuals for aircraft fdi. IEEE Transactions on Aerospace and Electronic Systems 45 (4), 1466–1482.

Freddi, A., Longhi, S., Monteriu, A., 2009. A model-based fault diagnosis system for unmanned aerial vehicles. In: 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes.

Gertler, J., 1998. Fault detection and diagnosis in engineering systems. Marcel Dekker.

Gertler, J. J., 1988. Survey of model-based failure detection and isolation in complex plants. IEEE Control Systems Magazine 8 (6), 3–11.

Gertler, J. J., 1997. Fault detection and isolation using parity relations. Control Engineering Practice 5 (5), 653–661.

Goshtasby, A. A., 2005. 2-D and 3-D Image Registration: for Medical, Remote Sensing, and Industrial Applications. Wiley.

Hansen, S., Blanke, M., jens Adrian, 2010. Diagnosis of uav pitot tube defects using statistical change detection. In: 7th IFAC Symposium on Intelligent Autonomous Vehicles. Vol. 7(1). IFAC PapersOnLine.

Heredia, G., Ollero, A., Bejar, M., Mahtani, R., 2008. Sensor and actuator fault detection in small autonomous helicopters. Mechatronics 18, 90–99.

Hopcroft, J. E., Karp, R. M., 1973. An n/sup 5/2 algorithm for maximal matchings in bipartite graphs. SIAM Journal on Computing, 225–231.

Horak, D. T., 1988. Failure detection in dynamic systems with modeling errors. Journal of Guidance, Control and Dynamics 11 (6), 508–516.

Isermann, R., May. 1997. Supervision, fault-detection and fault-diagnosis methods - an introduction. Control Eengineering Practice 5 (5), 639–652.

Isermann, R., 1998. On fuzzy logic applications for automatic control, supervision, and faultdiagnosis. IEEE Trans. On Sys. Man and Cyber. Part A-Sys. & Humans 28 (2), 221–235.

Isermann, R., 2005. Fault-diagnosis systems:An introduction from fault detection to fault tolerance. Springer.

Jain, A., Dubes, R., 1988. Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs.

Kay, S. M., 1998. Fundamentals of Statistical Signal Processing: Detection Theory. Prentice-Hall PTR.

Krysander, M., June 2006. Design and analysis of diagnosis systems using structural methods. Ph.D. thesis, Linköping University.

Laursen, M., Blanke, M., Düştegör, D., 2008. Fault diagnosis in a water for injection system using enhanced structural isolation. International Journal of Applied Mathematics and Computer Science 18 (4), 593–603.

Leitold, A., Hangos, K. M., 2001. Structural solvability analysis of dynamic process models. Computers and Chemical Engineering 25, 1633–1646.

Mangoubi, R. S., 1998. Robust Estimation and Failure Detection. Springer-Verlag.

Niemann, H., September 2006. A setup for active fault diagnosis. IEEE Trans. on Automatic Control 51 (9), 1572–1578.

Noura, H., Theilliol, D., Ponsart, J.-C., Cham, A., 2009. Fault-tolerant Control Systems - design and Practical Applications. Springer.

Park, W., Lee, S. H., Song, J., 2009. Fault detection and isolation of durumi-ii using similarity measure. Journal of Mechanical Science and Technology 23, 302–310.

Patton, R. J., 1991. Fault detection and diagnosis in aerospace systems using analytical redundancy. IEE Computing & Control Eng. J. 2 (3), 127–136.

Patton, R. J., Frank, P. M., Clark, R. N. (Eds.), 1989. Fault Diagnosis in Dynamic Systems, Theory and Application. Control Engineering Series. Prentice Hall, New York.

Pivano, L., Johansen, T. A., Smogeli, O. N., 2009. A four-quadrant thrust estimation scheme for marine propellers: Theory and experiments. IEEE Transactions on Control Systems Technology 17 (1), 215–226.

Poulsen, N. K., Niemann, H., 2008. Active fault diagnosis based on stochastic tests. International Journal of Applied Mathematics and Computer Science 18 (4), 487–496.

Romano, D., Kinnaert, M., 2006. Robust design of fault detection and isolation systems. Quality and Reliability Engineering International 22 (5), 527–538.

Staroswiecki, M., Attouche, S., Assas, M. L., June 1999. A graphic approach for reconfigurability analysis. In: Proc. DX'99.

Staroswiecki, M., Cassar, J. P., Cocquempot, V., Jul. 1993. Generation of optimal structured residuals in the parity space. In: 12th IFAC World Congress. Vol. 8. pp. 299–305.

Staroswiecki, M., Declerck, P., 1989. Analytical redundancy in nonlinear interconnected systems by means of structural analysis. In: Proc. IFAC AIPAC'89 Symposium. Vol. 2. Elsevier - IFAC, pp. 23–27.

Staroswiecki, M., Gehin, A. L., 2000. Control, fault tolerant control and supervision problems. In: IFAC Safeprocess' 2000. Budapest, Hungary.

Stevens, B. L., Lewis, F. L., 2003. Aircraft Control and Simulation, 2nd Edition. John Wiley & Sons.

Travé-Massuyès, L., Escobet, T., Olive, X., 2006. Diagnosability analysis based on component supported analytical redundancy relations. IEEE Trans. on Systems, Man and Cybernetics, Part A : Systems and Humans 36 (6), 1146–1160.

Unger, J., Kröner, A., Marquardt, W., 1995. Structural analysis of differential-algebraic equation systems - theoryand applications. Computers and Chemical Engineering (8), 867–882.

Varma, M., Zisserman, A., 2003. Texture classification: Are filter banks necessary? Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2, 691–696.

Wu, N. E., Klir, G. J., 2000. Optimal redundancy management in reconfugurable control systems based on normalised nonspecificity. Int. Journal of Systems Science 31, 797–808.

Zhang, X., 1989. Auxiliary signal design in fault detection and diagnosis. Springer Verlag.

Zhou, W.-w., Blanke, M., 1989. Identification of a class of non-linear state space models using rpe techniques. IEEE Transactions of Automatic Control 34 (3), 312–316.