Fault Detectability Analysis for Requirements Validation of Fault Tolerant Systems

Diego Del Gobbo*, Bojan Cukic†, Marcello R. Napolitano*

*Department of Mechanical and Aerospace Engineering
†Department of Computer Science and Electrical Engineering
West Virginia University
Morgantown, WV 26506-6106
delgobbo@ds5500.cemr.wvu.edu, cukic@csee.wvu.edu,
napolit@cemr.wvu.edu

S. Easterbrook

Department of Computer Science
University of Toronto
6, King's College Rd.,
Toronto, Ontario M5S 3H5
sme@cs.toronto.edu

Abstract

When high assurance applications are concerned, life cycle process control has witnessed steady improvement over the past two decades. As a consequence, the number of software defects introduced in the later phases of the life cycle, such as detailed design and coding, is decreasing. The majority of the remaining defects originate in the early phases of the life cycle. This is understandable, since the early phases deal with the translation from informal requirements into a formalism that will be used by developers. Since the step from informal to formal notation is inevitable, verification and validation of the requirements continue to be the research focus. Discovering potential problems as early as possible provides the potential for significant reduction in development time and cost.

In this paper, the focus is on a specific aspect of requirements validation for dynamic fault tolerant control systems: the feasibility assessment of the fault detection task. An analytical formulation of the fault detectability condition is presented. This formulation is applicable to any system whose dynamics can be approximated by a linear model. The fault detectability condition can be used for objective validation of fault detection requirements. In a case study, we analyze an inverted pendulum system and demonstrate that "reasonable" requirements for a fault detection system can be infeasible when validated against the fault detectability condition.

1. Introduction

Fault identification has been recognized as an important mechanism for improving system safety. For example, Failure Modes and Effects Analysis (FMEA) [17] considers the failure of any component within a system and tracks its effects to determine the ultimate consequences. Similarly, Event Tree Analysis (ETA) takes as its starting points the events that can affect the system and tracks them forward to determine their potential consequences. Fault Tree Analysis (FTA) differs from ETA in that it tackles the problem in the reverse direction. It starts with the identification of system hazards and works backwards to determine their possible causes.

FMEA, ETA, and FTA provide good frameworks for fault identification. Based on their results, system designers employ fault tolerant mechanisms to minimize the probability that, during system operation, identified potential faults propagate and cause the system to fail. However, fault identification techniques do not attempt to study *fault detectability*. In other words, these techniques may point to a potential problem and its consequences, but do not provide any assurance in the ability of the system to detect the problem if it appears during system operation.

In this paper, we focus on the specific requirements validation technique applicable in the analysis of dynamic fault tolerant control systems, such as nuclear and chemical plants, aircraft, and spacecraft. Fault tolerance is required in these systems for obvious safety, mission criticality and cost issues and it is often achieved by physical redundancy of the critical components. A Fault Detection (FD) scheme monitors these critical components and, as a fault is detected, recovers the system operation by switching from the faulty unit to one of the (non-faulty) backup units. Physical redundancy is a straightforward approach to fault tolerance. However, it presents some drawbacks, such as additional cost, power consumption, weight (important design parameters in aircraft and spacecraft) as well as the introduction of additional (but not essential) complexity.

0-7695-0418-3/99 $10.00 © 1999 IEEE

To overcome these limitations analytical redundancy ([5], [6], and [14]) has been considered as an alternative approach to physical redundancy in the technical literature. This approach is based on the fact that system inputs and outputs are functionally related variables that, when properly processed, allow the detection of faults in the system. High performance, reduced costs and reduced weight for these schemes make them desirable for practical applications. A variety of solutions has been considered in the technical literature and many issues related to their application have been addressed. Nevertheless, the practical implementation of analytical redundancy based fault tolerance schemes is still very limited. Fault tolerance algorithms are typically based on Kalman filtering, neural networks, and fuzzy logic techniques. The analytical complexity of these schemes can be problematic, especially when compared to the straightforward simplicity of physical redundancy. The skepticism with which analytical redundancy is sometimes received cannot be fully removed until a trustful verification procedure to certify a given fault detection scheme's performance becomes available.

Testing procedures for analytical redundancy schemes are often informal and inadequate. A typical testing environment involves the simulation of the system dynamics, along with the FD scheme. Faults are then injected into the simulated system and the FD scheme performance is assessed based on its promptness in detecting faults, its sensitivity to incipient faults (small size, slow varying faults), its completeness, as well as its false alarm rates. These analyses provide an overall measure of the scheme's capabilities, but this is still far from a formal verification of its functionality and performance. A formal procedure needs to involve system requirements specification, as well as verification and validation techniques spanning throughout the development life cycle. Requirements specification plays a fundamental role in the verification process since verification is meaningless without requirements.

In this paper we focus on the specific validation technique for the requirements of fault tolerant dynamic systems, with particular attention on the validation of feasibility of the requirements. Formal methods contribute significantly in validating consistency and non-ambiguity of requirements; however, there is no guarantee that a set of well-formed consistent requirements fully captures the desired behavior of the system and its interaction with the environment. The evaluation of the matching between "actual" functions needed within the environment and the specified requirements is often left to a subjective judgement. In this paper, we demonstrate that the validation of the requirements feasibility can be conducted in a more objective and formal framework.

For fault tolerant systems, the validation of the requirements needs to include the assessment of feasibility of the fault detection task. The degree of "visibility" and, consequently, the detectability of a fault in the system depends on the fault magnitude, on the signal-to-noise ratio related to the fault, as well as on the system dynamics.

The National Transportation Safety Board (NTSB) database lists several accidents where the fault detectability problem appears to be the critical issue. For example, [9] is a report related to an accident that occurred in 1983, involving a McDonnell DC-10-30 airplane. The aircraft experienced the separation of 50 inches of the right flap; however, "the incident was not noticed until a local resident called to report the fallen article". Detection of such a fault would be desirable to prevent a progressive deterioration of the surface with potentially catastrophic consequences. However, requiring to detect such a fault may be unrealistic if the detection task is unfeasible for circumstances related to the operational environment.

The purpose of the paper is to illustrate the scope of the fault detectability problem and to present a condition to assess fault detectability for analytical redundancy based FD schemes. The proposed condition is formulated in the frequency domain and is applicable to any system whose dynamics is well approximated by a linear model. Because of its analytical nature, it is a formal and objective tool for feasibility validation of system requirements for fault tolerant dynamic systems.

In Section 2, a brief overview of the validation process is provided to better define the feasibility validation task within the requirements specification process. Section 3 illustrates the fault detectability problem and provides the detectability condition. Then, in Section 4, an application of the detectability condition to the requirements of a simple system is presented to show how "reasonable" requirements can turn out to be unfeasible. We conclude the paper with a summary, in Section 5.

2. Validation of requirements

Specification of requirements involves three main steps: formulation of a high-level model of the system, definition of the requirements, and validation of the requirements. The first phase aims to capture the most relevant functions of the system to build and its interactions with the environment. Defining a conceptual

model of the system requires a thorough understanding of the environment in which the system will operate and of the functions that the system will provide [7]. Definition of the requirements involves the description of the services that the system should provide, the constraints under which it must operate and the desired performance attributes. Validation aims to assess if the requirements are complete, unambiguous, consistent, and realistic.

For high assurance systems, requirements validation is a particular concern [16]. The terms "Verification" and "Validation" are commonly used in software engineering to mean two different types of analysis. Validation is concerned with checking that the system will meet the customer's actual needs, while verification is concerned with whether the system is well-engineered. The distinction is clearer if one considers the role of a specification. Validation is the process of checking whether the specification captures the customer's needs, while verification is the process of checking that the software meets the specification.

Validation of requirements is problematic because it usually involves a subjective judgment of how well the system under consideration addresses a real-world need. In contrast, verification should be a relatively objective process, in that if the various products and documents are expressed precisely enough, no subjective judgments should be needed [4]. The use of formal methods for software specification is based on this observation. However, formal methods provide little help with requirements validation, because there is no guarantee that a set of well-formed, consistent requirements actually captures the desired behavior of the system and its interaction with the environment.

The current state of practice in requirements validation relies on a wide variety of validation techniques, ranging from informal and formal inspection processes, to detailed prototyping of user interfaces, and the use of simulations [8]. Ideally, the requirements described in the specification should be realistic, complete, consistent, unambiguous, and testable. Sommerville [15] lists four tasks in the requirements validation process:

1. Check if required functions satisfy user's needs,
2. Check if requirements include all functions and related constraints,
3. Check if requirements are unambiguous and consistent (non-conflicting),
4. Check if functions and constraints specified by the requirements are realistic.

It should be clear from this list of tasks that requirements validation is a systems engineering activity. A system level understanding of the purpose of the system, its environment, and the available technology is crucial. Prototyping and simulations can be used to assist with the first two tasks, while formal analysis can assist with the third. However, the last task, that of determining whether the requirements are feasible, has been given limited attention in the requirements engineering community.

An analysis of requirements feasibility is especially important for requirements that relate to safety and reliability. If it cannot be determined ahead of time whether the safety requirements for a system can feasibly be met, a number of problems may result. Firstly, time and effort can be wasted trying to meet the requirements. Worse, a significant change to safety requirements late in the lifecycle can be disastrous. Such a change will generally mean a different approach to safety and assurance, impacting a large number of design decisions that have already been made.

3. Validation of FD systems requirements: Fault detectability analysis

In this section an approach to feasibility validation of requirements for FD schemes is presented. The FD schemes considered are based on analytical redundancy. In order to formulate a detectability condition, we need to understand the analytical redundancy approach to fault tolerance and the dynamics of how the detectability problem arises.

Let us consider an actuator fault on an aircraft. Modern aircraft have several control surfaces, which are controlled either by the pilot or by the flight control computer. The "actuating chain" consists of an aerodynamic surface, an actuating unit, and linkages along which a control command is sent from the cockpit or the on-board computer. All of the components of the actuating chain can be duplicated except the aerodynamic surface. Therefore, if one of the actuating units fails, it can be replaced with its backup. But if a problem occurs with the aerodynamic surface, neither detection of the fault, nor its accommodation is possible. Aircraft accidents involving separation of aerodynamic surfaces during flight are not unusual. [10], [11], and [12] are the reports of three different aircraft that experienced partial separation of one of the trailing edge flaps. In all three cases the flight conditions were nominal until the descent for approach and landing. After deflection of the flaps the aircraft experienced rolling; the induced rolling moment had to be compensated through ailerons deflections by the pilot for the aircraft to land safely.

These examples not only illustrate the limitation of physical redundancy, but also show a possible different solution to detection and accommodation problems.

The separation of the flap caused a rolling moment which was not consistent with the aircraft inputs at that particular instant of time. A clear understanding of the aircraft dynamics and availability of roll angle and aileron deflection is sufficient to detect the fault. This is the basis of the analytical redundancy approach to the FD problem. Furthermore, the aircraft control was regained through aileron input by the pilot, implying that physical redundancy is not needed when the fault does not compromise the controllability property of the system.

To understand how the detectability problem may arise let us consider two interesting accident reports by the NTSB.

[12] states that on March 1997, 18 feet of the right outboard flap separated from a Boeing 767-232 during the airplane's approach to the Dallas/Fort Worth International Airport (DFW), Texas. The captain reported that "take-off and departures were routine, as were all aspects of the enroute phase of flight". According to the Digital Flight Data Recorder, after 1 minute and 54 seconds since deflection of the flaps the aircraft started rolling to the right. By using a significant amount of left aileron the first officer regained control and safely landed.

[9] reports that on September 1983, a McDonnell DC-10-30 experienced the separation of 50 inches of the right flap. The report follows saying that "the incident was not noticed until a local resident called to report the fallen article".

In the first case, the magnitude of the fault was such that the pilot clearly felt the separation of the surface. In the second accident the effects of the fault were negligible with respect to normal disturbances, and the problem passed undetected. Detectability of the fault clearly depends on the relationship between fault and disturbances effect on the output of the system. More specifically it depends on their ratio, so that what is actually relevant in the detectability problem is the signal-to-noise (S/N) ratio of the fault. Another key element that is brought to light by the first accident is the relationship between fault detectability and the state of the system (in this case the maneuver of the aircraft). It is unlikely that 18 feet of the flap surface suddenly detach from the wing without any warning. It is plausible that the detachment of the flap, or more specifically, the fracture of the bolts which fastened the surface to the carriage support beam (as reported in [12]) slowly propagated during flight and erupted during approach when the flaps are fully loaded. Unfortunately, the effects of an on-going fault on a flap are not visible in straight-and-level flight, while they become visible, and potentially dangerous, during approach, when the flaps are deflected and the missed surface induces a rolling moment on the aircraft.

So far we have considered the actual effects of the fault on the system. However, analytical redundancy based FD schemes use commanded inputs and measured outputs, which may not reflect the actual response of the system. If a fault occurs on one of the sensors, the actual output of the system will be different from the sensor output. This implies that the effects of the fault could be masked in the measured outputs even when they are evident in the actual output of the system. The key elements in this "game" of actual vs. measured outputs are the system and fault dynamics.

Following the previous discussion, the concepts summarized below are identified as the critical concepts in the fault detectability problem.

Detectability is strictly related to the fault dynamics: slow varying faults and low magnitude faults can be more difficult to detect. Disturbances also play a major role. Because of them, the functional relationships between system inputs and outputs suffer an inherent uncertainty. Analytical redundancy based FD algorithms depend on these functional relationships to accomplish their task; hence, disturbances are a key parameter in fault detectability assessment. The dynamics of the system plays an important role as well. Depending on the system dynamics, system outputs can mask or enhance information related to the fault. Hence, the elements involved in detectability assessment are the fault dynamics, the system dynamics, and the disturbance power spectrum.

The above examples and following argumentation provide an intuitive understanding of the detectability problem. The next step is to provide an objective definition of the problem and an analysis tool that can be used in the requirements validation process.

To specify meaningful requirements, detection feasibility of the fault set of interest needs to be assessed. In other words, the feasibility of the functions and performance constraints required from the FD scheme needs to be validated. Only a few papers in the technical literature have focused on fault detectability and a definition unanimously accepted by the technical community is still missing. In [3] and [13] different definitions of detectability are used, each referring to a specific FD approach. However, the definition of detectability is useful in the process of requirements

validation only if it is general, i.e., not specific to the particular FD technique employed. The FD scheme requirements, in fact, specify the desired functions of the scheme without describing the details about their implementation. The definition of detectability condition as a system property that does not depend on the adopted FD scheme has been recently proposed in [2]. This condition can be summarized as follows:

"A fault on the i^{th} component is detectable if a frequency band $\Delta\omega$ exists, such that the power in that band of the j^{th} output, due to the fault signal, is larger than the power due to disturbances".

In analytical terms, a fault on the i^{th} component is detectable if:

$$\exists\{j \in [0,m], \Delta\omega = (\omega_1, \omega_2)\}/ \quad \left\|Y_{ij}(j\omega)\right\|_{\Delta\omega} > \left\|Y_{dj}(j\omega)\right\|_{\Delta\omega} \tag{1}$$

where m is the number of outputs, $Y_{ij}(j\omega)$ is the Fourier transform of the output on the j^{th} sensor generated by a fault on the i^{th} system component, $Y_{dj}(j\omega)$ is the Fourier transform of the output on the j^{th} sensor generated by the disturbance, and $\|\cdot\|_{\Delta\omega}$ denotes the power of the signal in the frequency band $\Delta\omega$.

Eq. (1) is the condition needed to validate detection capabilities required from an FD system. Its application to any linear (or linearized) system is straightforward since the power spectral densities related to faults and disturbances can be easily computed separately and then compared. By comparing spectral densities over all frequencies and all outputs, it is possible to state whether a frequency band $\Delta\omega$ and an output $y_j(t)$ exist such that condition (1) is satisfied. Hence, it is possible to assess the detectability characteristics of the fault.

The system linearity constraint imposed to the application of Eq(1) is not as constrictive as it may appear. The dynamics of a system like an aircraft is described by a set of non-linear differential equations resulting form the application of basic physic laws. However, under certain flight conditions the aircraft dynamics can be approximated by a linear system. In fact, most of the design of flight control laws is based on this linear model. Use of linearized models is common practice in system engineering and it allows application of condition (1).

In the following section a practical example is provided where the linearized model of a non-linear system is used to validate the detection requirements for an analytical redundant FD scheme.

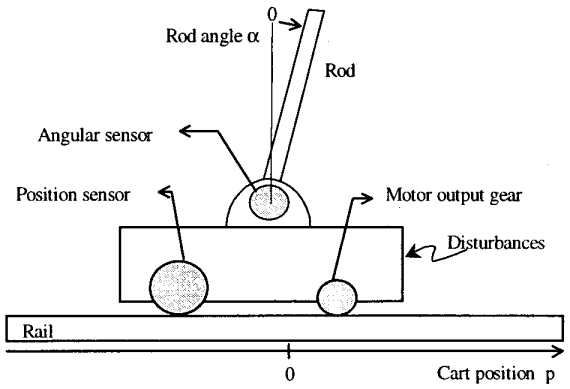

Figure 1. Inverted pendulum system.

4. Case study

Consider the inverted pendulum system in Figure 1. This system consists of a cart and a rod. The cart slides on a 1-meter length rail driven by a motor, while the rod is free to rotate around an axis perpendicular to the direction of motion of the cart. The system is equipped with sensors that measure cart position and rod angle. These variables are used within a feedback loop to stabilize the system around the unstable equilibrium condition with the rod balanced in the vertical position. The system is affected by disturbances whose statistical properties and power spectra are known.

A simplified block diagram of the system is shown in Figure 2 [1, 2]. $G_o(s)$ represents the non-linear dynamics of the inverted pendulum, K_r and K_x implement the control law, $H_d(s)$ describes the effects of disturbances within the system, r(t) is the reference signal for the position of the cart, d(t) is the disturbance input, f(t) is the fault input used to represent the occurrence of a fault on the system outputs, x(t) is the system output (actual output), y(t) is the sensor's output (measured output). Since y(t) represents the measurement of the system output, y(t) and x(t) will assume the same values (within the limit of the sensor accuracy) unless a fault occurs on one of the sensors. The reference input to the system is a square wave, so that the cart slides periodically from one end of the rail to the other, keeping the rod balanced within the capability of the control law and in spite of the disturbance action.

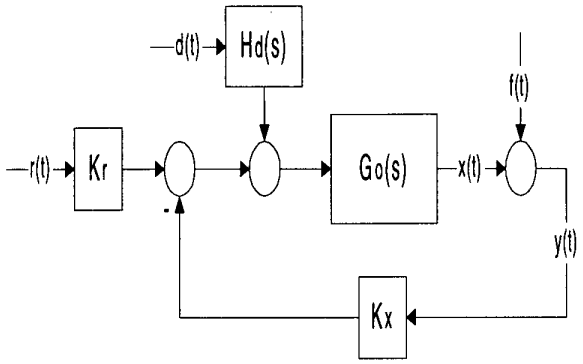

Figure 2. Block diagram of the inverted pendulum system.

A fault on the cart position sensor may cause the system to go beyond the rail ends, with possible damages to the system. Thus, an FD scheme is needed. Assume that the dynamics of the fault is exponential with a given size and time constant within a known interval as described by:

$$f(t) = A\left(1 - e^{-t/\tau}\right)$$
$$A = 0.1\ (m),\ \tau \in [0,2]\ (sec) \qquad (2)$$

The following constraint could be the core of the requirements specification for the FD system:

" the FD system shall detect any fault belonging to the class described by Eq. (2) "

From an engineering point of view the above constraint seems reasonable. The fault class described by Eq. (2) is realistic and a fault-size of *0.1 m* is considerable when compared to the rail length of *1 m*, so that the effects of the fault are visible on the system. The development of the requirements for such a system is beyond the scope of this paper; nevertheless, it is reasonable to assume that it is possible to specify a complete set of requirements including the above constraint. By using a formal language such requirements could be expressed without ambiguity and inconsistencies. What is left to be addressed is the feasibility validation of the requirements.

In Figure 3, the reference input, the actual position of the cart, and the output of the position sensor, are displayed against time for two different faults. The faults differ for the value of the time constant. In Figures 3a and 3b, the faults with time constants $\tau = 0$ and $\tau = 2$ are

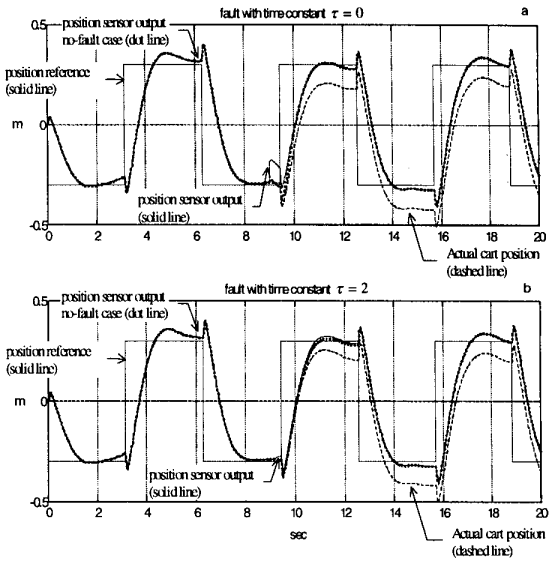

Figure 3. Fault effects on position sensor output.

considered, respectively. Both faults occur at $t = 9$ sec. The output of the position sensor in absence of fault (no-fault case) is also displayed on both graphs. Recall that, in the absence of a fault, actual cart position and position sensor output coincide.

By comparing the actual cart position with the output of the position sensor in the no-fault case, it is clear that the fault effects are visible for both faults. After the occurrence of the fault, the cart shifts downward, touching the end of the rail at $t = 15.7$ sec. Unfortunately, only measurements of system outputs are available for the FD scheme, while the actual position of the cart is not. In Figure 3a, the effects of the fault are visible on the position sensor output. However, after a short impulse at $t = 9$ sec, the effects of the fault rapidly disappear and, for $t > 10$ sec., the position sensor output in presence or absence of the fault almost coincide. Nevertheless, the fault is still present as the actual position of the cart demonstrates. Things deteriorate in Figure 3b, where the output of the position sensor does not show any effect of the fault.

From the above analysis it can be concluded that the system has a remarkable weakness in providing fault-related information for slow varying faults on the position sensor. To quantify this weakness, fault detectability analysis will be performed.

Figure 4 shows the spectral density of the sensors output for a fault on the position sensor, along with the power spectral density of the output component due to disturbances, for two different values of the time constant. All spectral densities have been computed using a linearized model of the non-linear dynamics $G_o(s)$. According to the detectability condition stated in Eq. (1), the fault is detectable if a frequency interval exists, where either the condition (3) or (4) is satisfied:

$$\left|Y_{pp}(j\omega)\right|^2 > \left|Y_{dp}(j\omega)\right|^2 \tag{3}$$

$$\left|Y_{pa}(j\omega)\right|^2 > \left|Y_{da}(j\omega)\right|^2 \tag{4}$$

From the power spectral density diagrams, it is clear that, for a value $\tau = 0$ of the time constant, the detectability condition is satisfied since:

$$\left|Y_{pp}(j\omega)\right|^2 > \left|Y_{dp}(j\omega)\right|^2 \quad \forall\omega > 7\,\mathrm{rad/sec} \tag{5}$$

On the other hand, a fault with time constant $\tau = 2$ reveals to be undetectable, since:

$$\left|Y_{pp}(j\omega)\right|^2 < \left|Y_{dp}(j\omega)\right|^2 \forall\omega \tag{6}$$

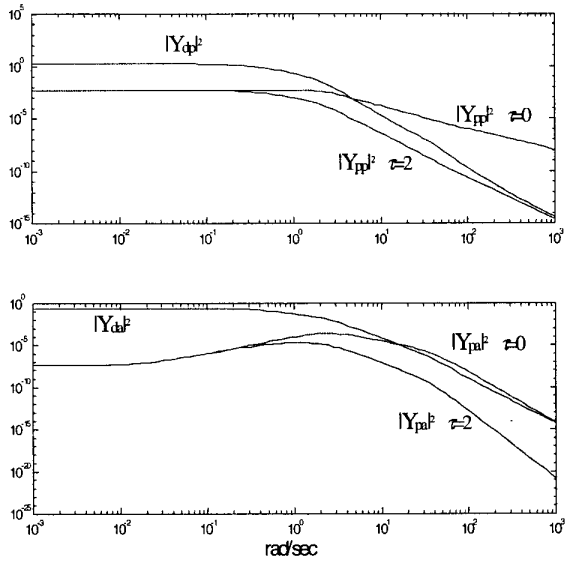

Figure 4. Power spectral density of sensor outputs for a fault on position sensor.

$$\left|Y_{pa}(j\omega)\right|^2 < \left|Y_{da}(j\omega)\right|^2 \forall\omega \tag{7}$$

Any fault with time constant $\tau \geq 2$ is masked by the effects of disturbances. Therefore, the detectability analysis has proved that the constraint stated at the beginning of this section is not realistic, and any set of requirements including that constraint is meaningless, regardless of its completeness and consistency properties.

5. Conclusion

System requirements specification and their validation play a key role in the development and the analysis of high insurance applications. Unfortunately, the extent to which system requirements meet the correctness, completeness and feasibility criteria is often left to the subjective judgement of the system analyst.

The assessment of requirements feasibility is a problem whose solution largely depends on the application field. We have focused on this problem within the field of analytical redundancy based FD schemes for dynamic systems. This study provides an analytical condition that allows assessing the feasibility of the fault detection task in systems whose dynamics can be modeled by a set of differential equations. The detectability condition captures the three key elements in assessment of fault detection feasibility: the dynamics of the fault, the impact of system disturbances on the functional relationship between system inputs and outputs, and the dynamics of the system.

The detectability condition does not directly provide a basis for the implementation of a FD system. However, it highlights the strengths and weaknesses of the system in propagating the effects of the fault throughout its output as a function of frequency. Hence, it provides information on which frequency regions are more suitable for the detection of a fault. Moreover, the detectability condition clearly states when a system is incapable of detecting fault propagation effects so that an approach other than analytical redundancy can be adopted for fault detection purposes.

Fault detectability analysis is an important step in requirements validation of FD systems. The suggested fault detectability condition provides an objective assessment tool within this framework. This has been demonstrated by applying the detectability condition in a case study involving the requirements for the FD scheme for an inverted pendulum system.

Acknowledgement

Partial support for research reported in the paper has been provided by DEPSCOR/AFOSR grant F49620-98-1-0136, NASA Ames grant NAG 2-1158, and NASA Dryden grant NAG 4-163 (administered by the Institute for Software Research).

References

[1] D. Del Gobbo, "Sensor Failure Detection and Identification using Extended Kalman Filtering", Master thesis, Department of Mechanical and Aerospace Engineering, West Virginia University, 1998.

[2] D. Del Gobbo and M. R. Napolitano, "Issues in Fault Detectability for Dynamic Systems", submitted for presentation.

[3] X. Ding, P.M. Frank and L. Guo, "An approach to residual generator and evaluator design and synthesis", IFAC 12th Triennial World Congress, pp. 383-86,1993.

[4] S. M. Easterbrook, "The Role of Independent V&V in Upstream Software Development Processes," Journal of Integrated Design and Process Science, vol. 2, pp. 37-46, 1998.

[5] P.M. Frank, "Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-based Redundancy-A Survey and Some New Results", Automatica, Vol. 26, No. 3, pp. 459-474, 1990.

[6] R. Isermann, "Process Fault Detection Based on Modeling and Estimation Methods-A Survey", Automatica, Vol. 20, No. 4, pp. 387-404, 1984.

[7] M. Jackson, "The Meaning of Requirements," Annals of Software Engineering, vol. 3, pp. 5-21, 1997.

[8] M. Lubars, C. Potts, and C. Richter, "A Review of the State of the Practice in Requirements Modeling," IEEE International Symposium on Requirements Engineering, San Diego, CA, IEEE Computer Society Press, pp. 2-14.

[9] NTSB final report, Accident no. MIA83IA218, September 1983.

[10] NTSB final report, Accident no. CHI93IA354, September 1993.

[11] NTSB final report, Accident no. NYC96IA169, August 1996.

[12] NTSB final report, Accident no. FTW97IA144, March 1997.

[13] M. Nyberg and L. Nielsen, "Parity functions as universal residual generators and tool for fault detectability analysis", 36th IEEE Conference on Decision and Control, vol. 5; pp.4483-9.

[14] R. Patton, P. Frank and R. Clark, "Fault Diagnosis in Dynamic Systems, Theory and Applications", Prentice Hall, 1989.

[15] Sommerville, "Software Engineering", 3rd Ed., Addison Wesley, 1989.

[16] D. R. Wallace and R. U. Fujii, "Software Verification and Validation: Its Role in Computer Assurance and Its Relationship with Software Project Management Standards", NIST Computer Systems Laboratory, Gaithersburg, MD, NIST Special Publication 500-165, 1989.

[17] N. G. Leveson, "Safeware: System Safety and Computers", Addison-Wesley, 1995