ONLINE TESTING EMBEDDED SYSTEMS: ADAPTING AUTOMATIC CONTROL TECHNIQUES TO MICROELECTRONICS TESTING

Emmanuel SIMEU, Salvador MIR, Libor RUFER

TIMA Laboratory, 46 Av. Félix Viallet
38031 Grenoble FRANCE

Abstract: This paper is aimed at exploiting Fault Detection and Isolation (FDI) techniques widely known in automatic control for solving online test problem in embedded Integrated Circuits (ICs). Before reaching this aim, we will briefly review the field of microelectronics testing, introducing basic concepts and techniques. We will next introduce FDI model-based approaches and their application for online testing of embedded ICs considering linear systems with potential faults and disturbances. The parity relation-based residual is specially suitable for this type of application. As an example, we will apply it to concurrent fault detection in a digital embedded filter. The proposed scheme will then be illustrated for a linear digital pass-band elliptic filter. *Copyright © 2005 IFAC*

Keywords: Embedded systems, online testing, parity space, fault detection and isolation, microelectronic circuits.

1. INTRODUCTION

Embedded systems are equipment made up of hardware and software incorporated in consumer products or other devices to perform some application specific functions. Generally, the product user is not even aware of the existence of these systems. From toys to medical devices, from ovens to automobiles, the range of products incorporating microprocessor-based embedded controlled systems has expanded rapidly since the introduction of the microprocessor in 1971. Embedded systems (we will use the term '*systems*' most often used in this paper) promise previously impossible functions enhancing the performance of people or machines.

As these systems gain sophistication, manufacturers are using them in increasingly critical applications that can result in injury, economic loss, or unacceptable inconvenience when they do not perform as required. Embedded systems can contain a variety of computing devices, such as microcontrollers, application-specific integrated circuits, and digital and analogue signal processors. A key requirement is that these computing devices continuously respond to external events in real time. Makers of embedded systems take many measures to ensure safety and reliability throughout the lifetime of products incorporating the systems.

Modern microelectronic manufacturing technologies and software tools make it feasible to integrate tens of millions of transistors, or even a complex system, into a single chip (SoC) able to hold all the components and functions that historically required a hardware board. In addition, in order to reduce development time, designers increasingly embed pre-designed and pre-verified complex functional modules in their SoC designs. These reusable modules are known as *cores*. Moreover, each core can serve diverse scenarios, so are reusable in different designs. Designers can also integrate cores from different vendors. When a certain combination of cores becomes common, a system integrator or core provider can create a new core from that combination. Hence, today SoCs could become the cores of tomorrow in more complex SoCs. Cores can be very different in nature, including digital, analogue, and even radio-frequency cores on the same chip.

Test development for such large, core-based SoCs poses major challenges; test *access* is one of them. Typically, a core is deeply embedded in the SoC and due to its surrounding circuitry direct access from the SoC pins to the core terminals is not possible. In

order to be able to test an embedded core as a stand-alone unit, it should be isolated from its surrounding circuitry and electrical test access needs to be provided (Rajsuman, 1999). Traditional constrains for IC design cost (expressed as silicon area), latency (expressed as the time required to generate an output), etc. must of necessity be completed with evaluation of testability and cost of testing. Algorithm-specific architectures seldom inherently exhibit those characteristics (such as regularity, ease of controllability and observability, possibility of partitioning, etc.) that facilitate testing. Adding design-for-testability features to a complete logic level design may become excessively costly in terms of both area and performance. In recent years, a number of authors have advocated introduction of test-related techniques since the first synthesis steps. In the same line, introduction of Built-In-Self Test (BIST) features allows us to achieve autonomous testing. BIST techniques correspond to an off-line testing philosophy, suitable for end-of-production testing or for periodic life-time testing. Strong reliability constraints have contributed to the evolution of highly reliable digital components (Nicolaidis, 1996). The new design philosophy, based on the hierarchical reuse of cores, requires system-level test architectures. For demanding applications, requiring high reliability of the embedded system and correctness of the results, some form of techniques for identifying faults during normal operation of the product must be introduced. That is, online testing techniques. Such solutions often exploit characteristics of the application (as specified by the algorithm implemented by the module) to achieve the required performance while limiting redundancy.

In automatic control area, the problem of fault detection and isolation (FDI) in dynamical systems has attracted a good deal of attention (Gertler, 1988, Chen *et al*., 1996, De Persis and Isidori, 2001). FDI deals with the generation of diagnostic signals sensitive to the occurrence of faults. Regarding a fault as an input acting on the system, a diagnostic signal must be able to detect its occurrence as well as to isolate this particular input from all other inputs (disturbances, other faults) affecting the system behaviour. One specific diagnostic signal (also called *residual*) must be generated per each fault to be detected, each diagnostic signal being sensitive only to one particular fault.

It seems attractive to adapt some of the results that are abundantly available in automatic control research to deal with the problem of online fault detection in electronic embedded systems. However, only some techniques will be applicable to electronic systems because the design and implementation constrains are very different in both research fields. In this paper, we will illustrate efficient online test architecture for digital embedded systems with respect to electronic design constrains. Our solution is built on the model-based parity space approach which, compared to other known FDI architectures, allows not only for efficient fault coverage but also for efficient implementation facilities.

The sequel of this paper is organized as follows. The next section introduces typical FDI approaches and discusses their applicability to online testing to of embedded ICs. Section 3 studies in particular the parity space method applied to concurrent fault detection in a linear digital filter. Results are provided for a linear digital pass-band elliptic filter. Finally, conclusions are discussed in Section 4.

2. ONLINE TESTING OF EMBEDDED ICs WITH MODEL-BASED FDI TECHNIQUES

2.1. Online testing of ICs

In developing an online BIST methodology for embedded systems, we must consider four primary parameters related to those listed earlier for online-testing techniques:

i) error coverage: the fraction of model errors detected, usually expressed as a percentage. Critical and highly available systems require very good error coverage to minimize the probability of system failure.
ii) error latency: the difference between the first time an error becomes active and the first time it is detected. Error latency depends on the time taken to perform a test and how often tests are executed. A related parameter is fault latency, the difference between the onset of the fault and its detection. Clearly, fault latency is greater than or equal to error latency, so when error latency is difficult to determine, test designers often consider fault latency instead.
iii) hardware overhead: the extra hardware needed for BIST. In most embedded systems, high hardware overhead is not acceptable. In any case, a hardware overhead equivalent to a duplication nominal system size should be avoided.
iv) performance penalty: the impact of BIST hardware on normal circuit performance, such as worst-case (critical) path delays. This includes the extra time needed for online testing. Overhead of this type is sometimes more important than hardware overhead.

The ideal online-testing scheme would have 100% error coverage, error latency of 1 clock cycle, no space redundancy, and no time redundancy. It would require no redesign of the CUT and impose no functional or structural restrictions on it. Most BIST methods meet some of these constraints without addressing others. Considering all four parameters in the design of an online-testing scheme may create conflicting goals. High coverage requires high error latency, space redundancy, and/or time redundancy. Schemes with immediate detection (error latency equalling 1) minimize time redundancy but require more hardware.

In a typical automatic control FDI implementation, the residuals are usually computed using specific algorithms embedded in controller computers. In this case, only two of the four constraints required for online testing of ICs have to be taken into account in the FDI design techniques: error coverage and error latency. Hardware overhead and

performance penalty constrains become pointless. FDI implementation does not require any additional hardware and do not generate any disturbance on the performances of the nominal system. These additional design constrains should be taken into consideration when applying FDI techniques to embedded microelectronic devices.

2.2. Model-based FDI techniques

Previous fault detection techniques were restricted to check directly measurable variables for upward or downward transgression of fixed limits or trends (Abdelhay and Simeu, 2000). This technique could be automated by using a simple limit-value monitor. Various faults in the plant could then be recognised only when the controlled signal exceeds some predefined thresholds. For so called *voting techniques*, a fault occurrence is determined by a sensor when some redundant equipment allowing a comparison between two or more measures points out a possible mismatch. This technique (often referred as *hardware redundancy*) is a more powerful method, but it has the disadvantage of requiring a lot of expensive fault-detection devoted equipment.

In the last twenty years, the use of process computers has allowed the development of new methods based on advanced signal processing techniques (Frank, 1990, Chen *et al.,* 1996)). In addition to model-free (MF) methods several model-based (MB) techniques for fault diagnosis have been proposed in the literature (Patton and Chen, 1991, Patton *et al*., 1989). A large class of MB diagnosis schemes relies on analytical models such as differential and difference equations. For example, linear discrete-time digital systems (LDS) is described by

$$x(k+1) = Ax(k) + Bu(k) + E_d d(k) + E_f f(k) \quad (1)$$

$$y(k) = Cx(k) + Du(k) + F_d d(k) + F_f f(k) \quad (2)$$

where $x \in R^n$ denotes the state vector, $u \in R^p$ the input vector, $y \in R^m$ the output vector, $d \in R^k$ the unknown disturbance vector, and $f \in R^q$ the fault vector to be detected. A, B, C, D, E_d, E_f , F_d, and F_f are known constant matrices of appropriate dimensions. The first step to successful fault detection is residual generation. For this purpose, various model-based methods have been developed, including unknown input observers, parameter identification and parity relation approaches.

Unknown input observer

The unknown input observer approach is based on the design of a full-state observer given as follows:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + K(y(t) - c\hat{x}(t)) \quad (3)$$

where $\hat{x} \in R^n$ is the estimated state, K is a matrix to be designed such that $\hat{x}(t)$ asymptotically converges to x(t), when no fault and no disturbance are considered. The residual is designed using the output estimation error derived from the observer

$$r(t) = Q(y(t) - C\hat{x}(t)) = QC(x(t) - \hat{x}(t)) + QF_d d(t) + QF_f f(t) \quad (4)$$

where Q is a r x m matrix and r is the size of the residual.

In the diagnosis schemes, a bank of observers may be used instead of a single observer-based scheme. Each observer residual is designed to be sensitive to a single fault while remaining insensitive to the other faults and disturbances. This approach requires a total implementation of the system model and will need more hardware overhead required than duplication. For this reason, observer-based FDI is not appropriate for adaptation to online testing of embedded ICs.

Parameter estimation method

The parameter estimation method makes use of the fact that the faults of a dynamic system are reflected in physical parameters as, for example, resistance, capacitance, inductance or friction, mass, viscosity, etc. The idea of the parameter estimation approach is to detect the faults via identification of the parameters of the mathematical model. Deviations of the process are evaluated or tracked through a recursive estimation of its parameters. Therefore, the available input-output data must be processed nonlinearly. The input signal must satisfy several conditions concerning process excitation with respect the dynamics and the nonlinearity which have to be estimated. The test can then be performed using a residual measure of form:

$$r = (\hat{\theta}_a - \hat{\theta}_0)^T (\hat{\theta}_a - \hat{\theta}_0) \quad (5)$$

Where $\hat{\theta}_0$ and $\hat{\theta}_a$ denote the estimates of nominal and actual parameters respectively. Finally, in any case, the residual is compared with a fixed threshold r_{max} and fault free hypothesis is accepted if $r < r_{max}$.

The parameter estimation method requires online monitoring of the system parameters. Complex calculation is developed in the monitoring algorithm, including large matrix inversion or application of matrix inversion theorem in the case of recursive algorithm. The hardware overhead required is widely prohibitive for an application to online testing of ICs.

Parity relation-based method

The basic principle of this method is as follows: as input of concurrent fault detection scheme, only available measurable signals must be used (input vector u and output vector y). Since the state vector x is not supposed to be directly measurable from the system under test, it is assumed unknown and must not be used in the concurrent fault detection scheme. Hence, the vector of state variables $x(t)$ has to be eliminated from Equation (2). The goal is achieved by expressing the output vector y at the successive clock time (t+1, t+2, …, t+k) in terms of state vector $x(t)$ and the input sequence, ($u(t+1)$, $u(t+2)$, …, $u(t+k)$). In the fault free operation, the k+1 successive expressions of the output vector are given by the following equations:

$$y(t)=Cx(t)+Du(t)$$
$$y(t+1)=CAx(t)+CBu(t)+Du(t+1)$$
$$y(t+2)=CA^2x(t)+CABu(t)+CBu(t+1)+Du(t+2)$$
$$\dots$$
$$y(t+k)=CA^kx(t)+CA^{k-1}Bu(t)+CA^{k-2}Bu(t+1)+\dots+CBu(t+k-1)+Du(t+k).$$

This set of equations can be group and presented in the following condensed matrix form:

$$Y^{[k]}(t) = OB^{[k]}x(t) + H^{[k]}U(t) \qquad (6)$$

Where

$$Y^{[k]}(t) = \begin{bmatrix} y(t) \\ y(t+1) \\ y(t+2) \\ : \\ y(t+k) \end{bmatrix}; \quad OB^{[k]} = \begin{bmatrix} C \\ CA \\ CA^2 \\ : \\ CA^k \end{bmatrix}; \quad U^{[k]}(t) = \begin{bmatrix} u(t) \\ u(t+1) \\ u(t+2) \\ : \\ u(t+k) \end{bmatrix};$$

$$H^{[k]} = \begin{bmatrix} D & 0 & .. & 0 & 0 \\ CB & D & : & 0 & 0 \\ CAB & CB & : & : & : \\ : & : & .. & D & 0 \\ CA^{k-1}B & CA^{k-2}B & .. & CB & D \end{bmatrix}.$$

The integer k will define the order of the residual detection scheme, i.e. the number of unit delay that will be needed for each signal connected in the residual generation scheme. If n is the order of the system, then $OB^{[k]}$ is the observability matrix of the system when $k=n-1$. Elimination of the state vector $x(t)$ from Equation (6) gives a redundancy relation which is a linear combination of present and lagged values of input and output sequences. The residual associated with the redundancy relation equals zero if no failure occurs in the system. Elimination of unknown variables from Equation (6) can be achieved if there is a vector v orthogonal to $OB^{[k]}$ such that $v^T.OB^{[k]} = 0$. The subspace P_k of all the vectors orthogonal to $OB^{[k]}$ is defined by:

$$P_k = \{ v \mid v^{\mathrm{T}}.OB^{[k]} = 0 \} \qquad (7)$$

P_k is the *parity space* of order k, and any vector v in P_k is a *parity vector.* Equation (7) can be verified if the rows of matrix $OB^{[k]}$ are linearly dependent. In another way, P_k exists if the $OB^{[k]}$ matrix *rank* is lower than the number of its *rows* ($p\times(k+1)$). Every *parity vector* v of the subspace P_k can be associated at any time (t) with a redundancy relation to generate a residual or a *parity check* $r(t)$, like:

$$r(t) = v^T[Y^{[k]} - H^{[k]}U^{[k]}] \qquad (8)$$

Equation (8) defines a redundancy relation and in the fault free operation:

$$r(t) = v^T OB^{[k]} x(t) = 0 \qquad (9)$$

From the above discussion, it is clear that the problem of concurrent fault detection in LDSs using only available measurable variables can be reformulated as follows: find parity vectors $v=[v_1,\dots, v_n]$ belonging to P_k such that Equation (8) can be verified; then, redundancy relations can be determined and so concurrent fault detection circuits can be constructed.

The implementation of the parity residual given by Equation (8) seems relatively simple. Moreover, as opposed to the automatic control processes on which any measurement is obtained using expensive sensors on few and far connections, a lot of measurement possibilities generally exist in electronic ICs. This enables the design of efficient parity residual of order 1 or 2 enabling the simplification of the implementation scheme. These advantages make the parity state approach more suitable for online testing of embedded systems than observer-based and parameter identification techniques (Simeu *et al.*, 2001). The next section illustrates the application of the parity space method for fault detection in an embedded digital elliptic filter.

3. APPLICATION OF PARITY BASED TESCHNIQUE FOR CONCURRENT TESTING

To illustrate the parity space technique described above, the linear digital pass-band elliptic filter in Fig. 1 is used. This system has one external input u, one external output y_0, two connectable internal signals y_1, y_2 and four state variables $x_0, \dots, x_3$. The state-space matrices for this system are:

$$A = \begin{bmatrix} -0.3277 & -0.1331 & 0.7887 & -0.1561 \\ 0.1331 & -0.1847 & 0.1561 & 0.9565 \\ -0.7887 & 0.1561 & 0.0747 & 0.1831 \\ -0.1561 & -0.9565 & -0.1831 & -0.1221 \end{bmatrix},$$

$$C = \begin{bmatrix} 0.0177 & 0.3103 & 0.0208 & 0.3641 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.3578 \\ 0.0708 \\ -0.4198 \\ -0.0831 \end{bmatrix}, \quad D = \begin{bmatrix} 0.1094 \\ 0 \\ 0 \end{bmatrix}.$$

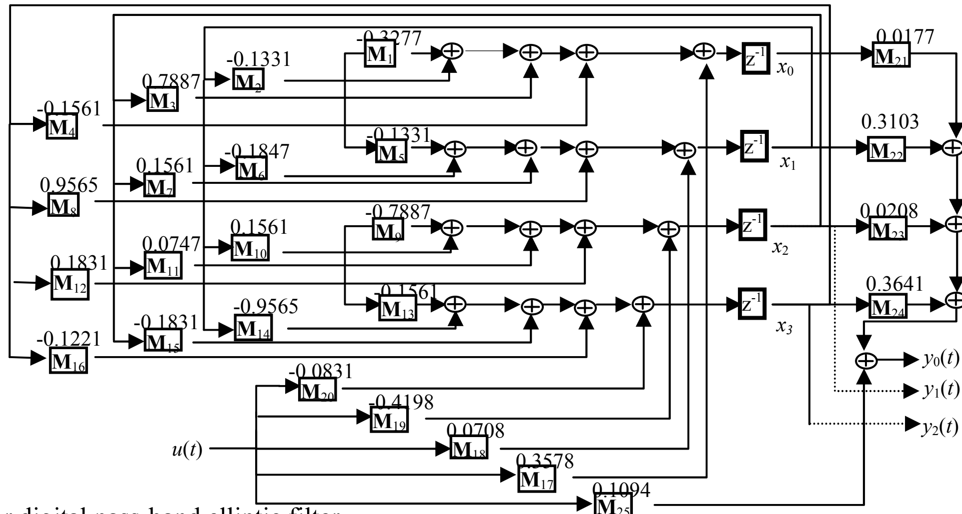


Fig. 1. Linear digital pass-band elliptic filter

3. 1. Concurrent residual design

The parameters of the fault detection circuit for this system have been computed using the procedures described above. These parameters are the following:
Redundancy relation order: $k = 1$
Optimal parity vector:

$$v^{T} = [-73.99 \quad -6.76 \quad 8.16 \quad 49.42 \quad 4.71 \quad -44.14].$$

$$v^{T}H^{[1]} = [-6.93 \quad 5.41]$$

The optimal robust redundancy relation is:

$$r(t) = v^{T} \cdot \begin{bmatrix} y_0(t-1) \\ y_1(t-1) \\ y_2(t-1) \\ y_0(t) \\ y_1(t) \\ y_2(t) \end{bmatrix} - v^{T}H^{[1]} \cdot \begin{bmatrix} u(t-1) \\ u(t) \end{bmatrix}$$

$r(t)$={-73.99$y_0(t$-1)-6.76$y_1(t$-1)+8.16$y_2(t$-1)
+49.42$y_0(t)$+4.71$y_1(t)$-44.14$y_2(t)$}
-{-6.93$u(t$-1)+5.41$u(t)$}.

The scheme of the circuit computing the normalised residual is given in Fig. 2.

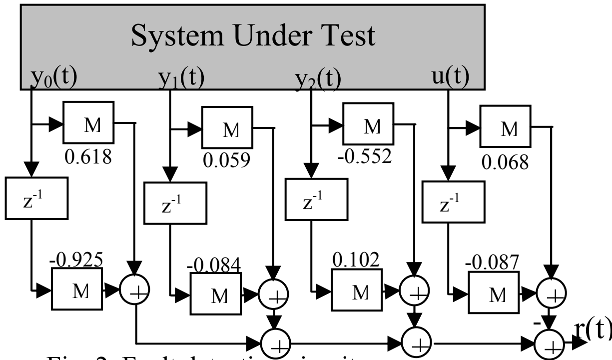


Fig. 2. Fault detection circuitry

3.2. Experimental results

The filter was simulated with the error (fault) detection circuit indicated in Fig. 2 and the error detection threshold was chosen (by simulation) to be the largest residual response value in the case of fault-free operation.
Generally, faults are simulated using fault models. Fault models are means of describing the effects of faults that may cause an error in the output of the circuit under test. For complex microelectronic circuits, circuit level fault models (such as at transistor and gate level fault model) result in very time consuming fault simulation. The fault models to be considered here and used to simulate faults are the behavioural, i.e. the faults are expressed by means of deviations from the nominal parameters included in the model describing the system under test. The filter to be simulated consists of constant-multipliers, adders and registers. The faults of a constant-multiplier can be modelled as deviations from its nominal constant value, so changing the nominal constant can simulate the presence of faults. Adder and register faults are assumed to be covered by faults on its input constant-multiplier. These faults can be then simulated using the constant-multiplier fault models.

Concurrent detector fault coverage

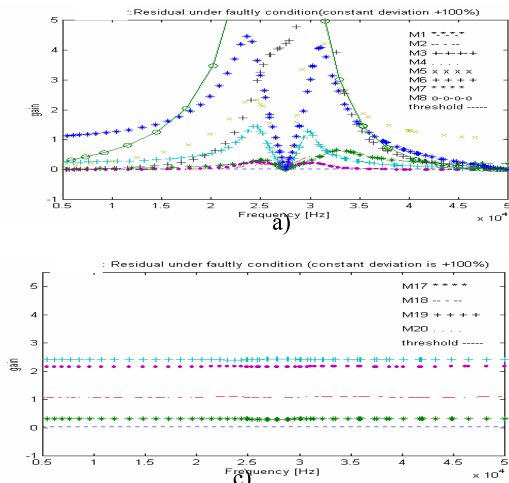
The filter was simulated for various frequencies of the input signal. The fault simulation corresponds to parametric deviations from their nominal value of the multiplier constant inputs. For each single fault, the corresponding residual response was evaluated. For each multiplier constant, Table 1 gives the maximal deviation tolerated by the system behaviour and the minimal deviation detected by the detector output.

Table 1. Robust detector fault coverage

Constant Value of multiplier	Deviation of the constant value	
	Fault-secure maximal deviation in %	Minimal detectable Deviation in %
M_1	0.5	0.8
M_2	2.5	2
M_3	1.5	0.5
M_4	1.8	1.5
M_5	0.6	0.2
M_6	0.3	0.1
M_7	0.5	0.2
M_8	0.1	0.02
M_9	0.1	0.06
M_{10}	0.7	0.3
M_{11}	0.9	0.8
M_{12}	0.6	0.3
M_{13}	0.15	0.08
M_{14}	0.06	0.01
M_{15}	0.7	0.08
M_{16}	0.3	0.1
M_{17}	1.3	2
M_{18}	0.9	0.5
M_{19}	0.5	0.2
M_{20}	0.2	0.2
M_{21}	0.3	0.2
M_{22}	0.01	0.01
M_{23}	0.55	0.2
M_{24}	0.04	0.01
M_{25}	0.09	0.04

The second column of this table represents (in percent) the minimal deviation of the corresponding constant-multiplier required to observe the error manifestation on the output of the system. Any deviation less than this value will not have a significantly observable effect on the system behaviour. So the system can be considered as fault-secure for any deviation less than this limit value. The third column represents the minimal parameter deviation detectable by the robust detector scheme implemented by the circuit of Fig. 2. The error detector output oversteps the threshold significantly when the corresponding value of deviation is simulated. The following results may be deduced from Table 1: 23 over 25 (92 %) of deviations are detected by the robust detector scheme before their effect is observable on the nominal system behaviour. This is not the case for M_1 and M_{17}.

Fig. 3 presents the complete spectral plot of the corresponding residual responses for deviation value of +100% for each constant-multiplier. These simulation results show that all faults are detectable in a large band of operational frequency.



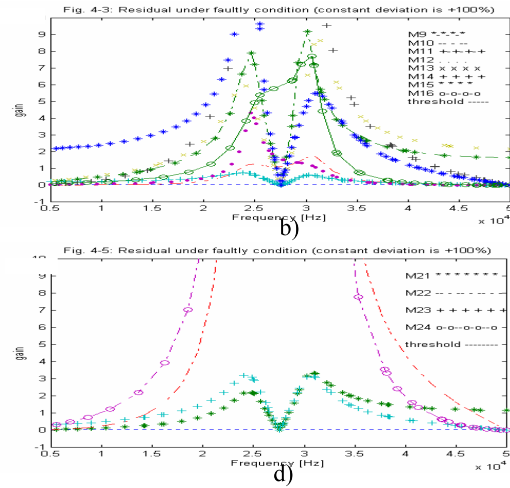


Fig 3. Spectral plot of the faulty detector response

Efficiency in robust concurrent detection

To illustrate the efficiency of the proposed scheme to detect faults concurrently, the fault detection circuit in Fig. 2 was simulated with the filter circuit indicated in Fig. 1. A sinusoidal signal of frequency 30 kHz and amplitude 1V was sampled and applied into the system input, u. During the simulation, a deviation of +50% from the nominal value of constant multiplier M_{24} was made. The calculation of the temporal residual response indicated in Fig. 4 shows how the residual changes immediately when the deviation (fault) occurs.

Insensitivity to noise of the robust scheme

Fig. 4 shows the efficiency of the robust fault detection scheme and its insensitivity to process noise. For simulation, a sinusoidal signal of frequency 30 kHz and amplitude 1V was sampled and applied into the system input. A digital noise signal with amplitude 0.002 was first injected into the filter circuit during a short period of time. Next, a fault inducing a small deviation of the constant of multiplier M_{25} was integrated. It is clear on the plot of the detector output signal that the robust residual scheme is sensitive to fault and insensitive to a noise.

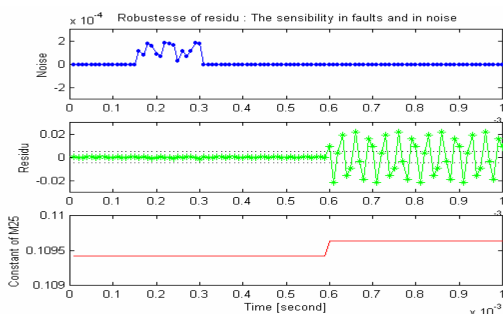


Fig. 4. Robustness of the detector response.

4. CONCLUSION

This paper prospects the possibilities for adapting model-based FDI techniques that are developed in the area of automatic control for online testing of embedded electronic systems. A new low-overhead online testing technique is deduced from FDI parity based residual.

This technique is illustrated for online concurrent testing of a digital embedded filter. The proposed scheme ensures a high sensitivity against faults while the robust fault detection circuit presents a negligible sensitivity towards system noise. The proposed technique is applicable to any type of linear digital system and the test circuitry required for the implementation of the concurrent fault detector is still very reasonable.

REFERENCES

Abdelhay, A. and E. Simeu (2000) ‘‘Analytical Redundancy Based Approach for Concurrent Fault Detection in Linear Digital Systems’’, *6th IEEE International OnLine Testing Workshop* (*IOLTW*), July 2000.

Chen, J., R. J. Patton, and H. Y. Zhang, (1996) “Design of unknown input observers and robust fault detection filters,” *Int. J. Control*, vol. 63, no. 1, pp. 85–105, 1996.

De Persis, C. and A. Isidori, (2001) A Geometric Approach to Nonlinear Fault Detection and Isolation, *IEEE Trans on automatic control* vol.. 46, n°. 6, JUNE 2001

Frank, P. M. (1990) Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy—A survey, *Automatica*, vol. 26, pp. 459–474, 1990

Frank, P. M. (1996) “Analytical and qualitative model-based fault diagnosis—A survey and some new results,” *Eur. J. Control*, vol. 2, pp. 6–28, 1996.

Gertler, J. J. (1988) “Survey of model-based failure detection and isolation in complex plants,” *IEEE Contr. Syst. Mag.*, vol. 3, pp. 3–11, 1988.

Nicolaidis, M. (1996) “Theory of Transparent BIST for RAMs,” *IEEE Trans. Computers*, Vol. 45, No. 10, Oct. 1996, pp. 1141-1156.

Patton, R. J., P. M. Frank, and R. N. Clark, Eds.(1989), *Fault Diagnosis in Dynamic Systems, Theory and Application.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

Rajsuman, R. (1999) “Testing a System-on-a-Chip with Embedded Microprocessor,” in *Proc. Int. Test Conf.*, 1999, pp. 499–508.

Simeu, E., A. Abdelhay and M. A. Naal (2001). Robust Concurrent Self Test of Linear Digital Systems’’ *Tenth Asian Test Symposium November* 19-21, 2001, Rihga Royal Hotel Kyoto, Kyoto, Japan.