An Approach for Fault Detection and Isolation in Dynamic Systems From Distributed Measurements

Eric-J. Manders, *Student Member, IEEE*, Lee A. Barford, *Member, IEEE*, and Gautam Biswas, *Senior Member, IEEE*

***Abstract*—An application is presented for online model-based fault detection and isolation (FDI) in a multitank fluid system. The tank system is equipped with a distributed measurement and control system that implements components of the IEEE standard for smart transducers, IEEE 1451 [1]. This standard includes an information model that provides programming constructs to support high level application functionality on a distributed network of smart transducers. The model-based FDI methodology in this work has several aspects that may be realized on such a distributed network. In the current work, the FDI application operates on a workstation that appears on the network as another (virtual) transducer node. The concurrent tasks in the application may be associated with actual transducer nodes. It represents a first effort toward constructing capabilities for distributed FDI in complex dynamic systems.**

***Index Terms*—Fault diagnosis, IEEE 1451, instrumentation, monitoring.**

I. Introduction

MANY complex engineering systems are deployed in environments where reliability and safety are of primary importance. Such systems are subject to stringent requirements for robust supervision and control. Faults in system components are likely to adversely affect both safety and functionality, and consequently, the need for capabilities to detect faults and to identify the faulty component(s) is becoming an integral part of those requirements. This is the fault detection and isolation (FDI) problem, and the primary objective in fault diagnosis [2].

Conventional techniques to ensure operational safety and reliability have included hardware redundancy and localized hardware safety mechanisms (e.g., check valves). These mechanisms are also the basis for many current approaches to FDI. For complex systems, the design of comprehensive FDI capabilities using this approach becomes unfeasible, both from a theoretical and practical viewpoint. An alternative approach is to infer faults from discrepancies in the observed behavior of the system through analytical techniques. Ideally, this analysis should take the process dynamics into account. When these dynamics can be described with a model, FDI techniques may exploit the functional redundancy in the model. This is referred to as the model-based approach to FDI.

Manuscript received May 4, 2000; revised January 27, 2002.

E.-J. Manders and G. Biswas are with the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37235-1592 USA (e-mail: manders@vuse.vanderbilt.edu; biswas@vuse.vanderbilt.edu).

L. A. Barford is with Agilent Laboratories, Palo Alto, CA 94303-0889 USA (e-mail: lee_barford@labs.agilent.com).

Publisher Item Identifier S 0018-9456(02)04320-6.

The operational reliability of a complex system can potentially be enhanced also through the use of a distributed measurement and control (DMC) system, particularly in large scale, sensor rich environments. The emerging technology of networked smart transducers facilitates the construction of DMC systems. Smart transducers contain an embedded processor providing computational resources to support complex sensing and actuating tasks and high level applications in a distributed setting. FDI is one such application.

This paper discusses an application for online model-based FDI that exploits the resources of a network of smart transducers. The application is designed around a multitank fluid system test bed, that is equipped with a DMC system using smart transducers. The smart transducers are implemented using components from IEEE 1451, the IEEE Standard for a Smart Transducer Interface for Sensors and Actuators [1].

TRANSCEND is a framework for model-based FDI of dynamic systems [3]. It combines robust transient analysis methods with a model-based qualitative fault isolation strategy that apply a qualitative constraint analysis. Because qualitative methods process input in symbolic form, a signal-to-symbol transformation step is required to compute symbolic feature values from continuously sampled measurement data. The methodology has been evaluated with simulation studies for various systems, including multitank fluid systems and a secondary sodium cooling loop system for a fast breeder reactor [3], [4]. In recent work, the method was applied offline in FDI of the cooling system of an automotive engine test bed [5].

Several aspects of this FDI scheme are suitable for a distributed architecture. Fault isolation based on symbolic descriptions of transient data is explicitly separated from the signal-to-symbol transformation methods that generate those symbolic descriptions. Realizing the symbol generation on the transducer node itself is one of the goals in building the distributed application. This can potentially reduce the network load of the monitoring and supervision tasks considerably.

The remainder of this paper is organized as follows. Section II describes the DMC system for the tank system. Section III describes the modeling of the tank system for FDI, the FDI application itself, and some experimental results. Section IV presents a summary and conclusions.

II. Test Bed for Distributed Measurement and Control

To demonstrate the benefits of IEEE 1451 for a complex DMC system, a multitank fluid system test bed, incorporating

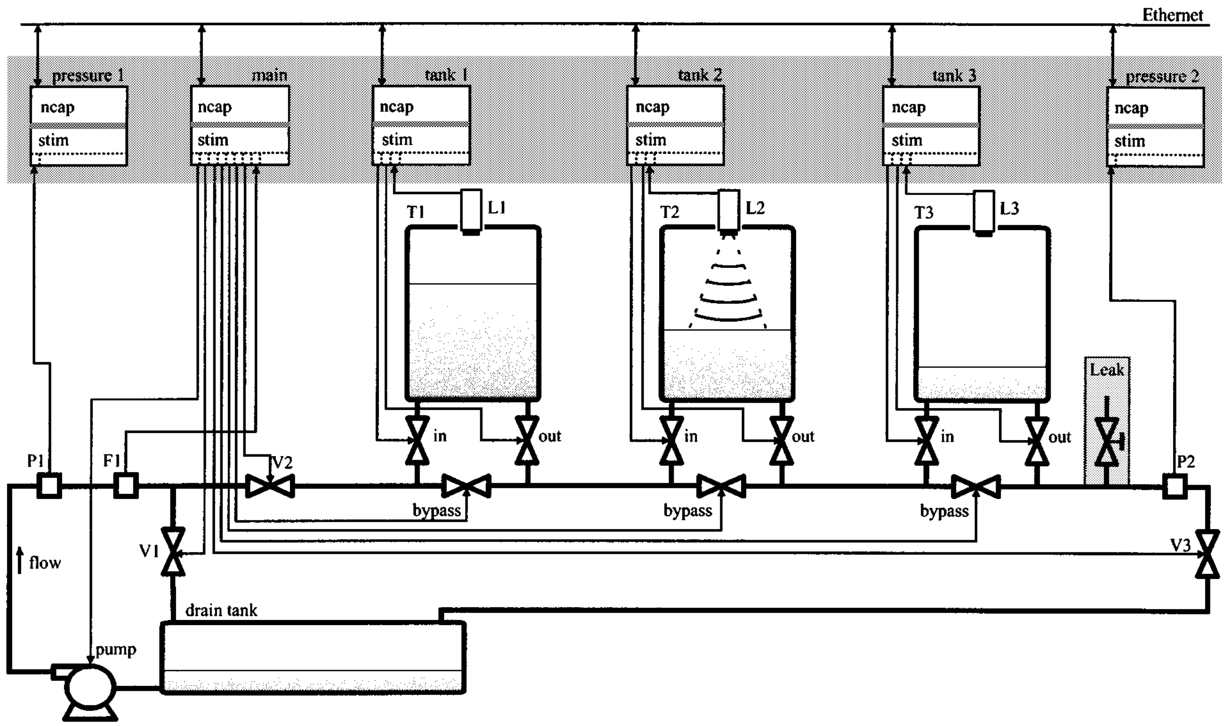


Fig. 1. Three-tank fluid system test bed with a distributed measurement and control system (shaded) based on the IEEE 1451.1 and 1451.2 standards.

standards compliant hard- and software, was designed and built at Agilent Laboratories, Palo Alto, California. The system has subsequently become a test bed for research in distributed model-based FDI.

A. A Standard for Smart Transducers

The IEEE 1451 family of standards provides an open platform for the development of networked smart transducers [1]. The standardization process is ongoing but several component standards have been ratified.

The 1451.2 standard specifies a smart transducer interface module (STIM), the interface between a sensor or actuator and a microprocessor [6]. The STIM provides plug-and-play capabilities at the transducer level and may handle sensing and actuating functions over multiple channels. The 1451.1 standard specifies a Network Capable Application Processor (NCAP), an extensible object-oriented information model that represents the interface of an (abstract) transducer to a network [7]. The NCAP thus facilitates interoperability at the application layer of a DMC system. The information model includes function objects and transducer objects that encapsulate application functionality and sensor/actuator channels, respectively. Distributed programming constructs support network-neutral communication through publish-subscribe and client-server mechanisms. A complete smart transducer node is composed of an NCAP/STIM pair, where the physical device operating as the NCAP includes a 1451.2 interface driver.

The transducer node hardware in this work is designed and built by Agilent Laboratories. The STIM is built around a standard micro-controller [9] and can be configured for multi-channel mixed analog and digital I/O. A prototype of an embedded ethernet controller acts as NCAP.[1] This device includes an Ethernet network interface and implements the IEEE 1451.2 interface through custom hardware. The operating system is an off-the-shelf real-time embedded OS (VxWorks [8]), with the publish-subscribe mechanism implemented over IP/multicast. The prototype transducer nodes have some limitations that affect the design of the FDI application. This will be discussed further in Section III.

B. Three-Tank Fluid System

The fluid system, shown in Fig. 1, consists of three tanks, connected in a series configuration. A closed-loop fluid path is created with an additional drain tank and a variable speed electric pump. The fluid flow for each tank is controlled by an inlet, outlet, and bypass valve. Three additional valves in the system provide further control of the fluid flow and pressure in the system. Each tank has an ultrasonic level sensor (L1, L2, L3), to measure the fluid level, which is proportional to the pressure at the bottom of the tank. The main fluid line includes additional pressure sensors (P1, P2) and a flow sensor (F1).

A network of six smart transducer nodes makes up the DMC system. A dedicated node for each tank manages the level sensing functions and actuating functions of the inlet and outlet valves for that tank. A tank node publishes its level sensor data and the state of the valves. A tank with its transducer node functions as a "smart tank," capable of adjusting its own fluid level in a semi-autonomous way using feedback from the level sensor to stop and start the fill and drain operations. Supervisory control for a smart tank can therefore be abstracted into "fill, " "hold," and "drain" commands.

[1]The production version of this device is no longer commercially available.

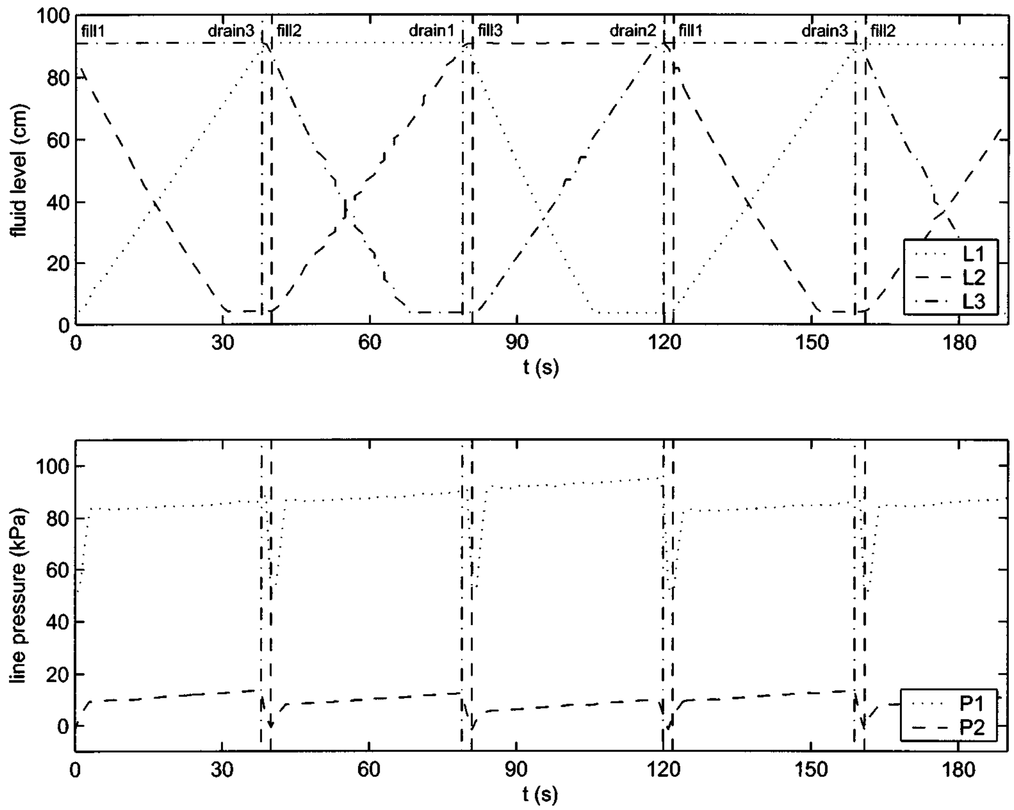


Fig. 2. Complete operating cycle of the system, showing tank level data (t) and line pressure data (b). Commands that initiate "fill tank" and "drain tank" modes are indicated also. The sequence starts with tank 1 in "fill" mode, tank 2 in "drain" mode, and tank 3 full and in "hold" mode.

The main node subscribes to the published measurement and state data from each tank. The main node also has its own STIM through which it acquires flow sensor data, controls the pump speed, and controls the additional valves in the system. The pressure transducers, prototypes from a process automation industry manufacturer, integrate the pressure sensor with the transducer node in a single housing. These devices publish pressure data.

Supervisory control for the system is also carried out on the main node. Tank control commands are issued to each tank node through a dedicated client-server connection. The controller is a finite state machine, implemented as an IEEE 1451.1 function object. The standard autonomous operating sequence fills and drains the tanks in a fixed cyclical pattern, shown in Fig. 2. The tank system thus has multiple operating modes, with different continuous time dynamics in each mode, making it a hybrid system. In this work, we address the FDI problem within an operating mode.

III. FDI Application for the Three-Tank System

The multitank fluid system represents a simplified version of a large class of physical plants, and is often used as reference system for research in automatic control and fault detection and isolation. Simulation experiments with TRANSCEND for faults in a two-tank fluid system are described in [3].

A. Modeling the System for FDI

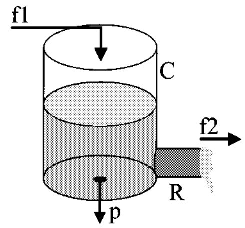
The continuous dynamics of a fluid tank are illustrated in Fig. 3. For a pipe connected to a tank the rate of change in the pressure, $\dot{p}$, at the bottom of the tank is proportional to the net fluid flow, $f_1 - f_2$, and inversely proportional to the tank capacitance, C. The functional relation for flow, f, through a pipe is proportional to the pressure drop over the pipe, p, and inversely proportional to the pipe resistance, R. Because only qualitative values and qualitative relations are used for fault analysis, it is in fact not necessary to convert the level measurement values from the system into actual pressure values.

The model used in the qualitative FDI algorithms captures dynamic characteristics of the dependency relations between component parameters and the measured variables in the form of a temporal causal graph (TCG). The model is constructed as a bond graph (Fig. 4), a graphical, component based, compositional modeling language from which the TCG can be generated automatically. In this work, the pump is modeled as an ideal source of flow, although it has been modeled as a system component (gyrator) in other work [3].

When a mode switch occurs during the tank system cycle, the continuous dynamics, and consequently the model, will change. In each mode, only one tank can be draining, and only one tank can be filling. This implies that, when inertial effects of the fluid flow can be ignored, the dynamics of the model in any mode never exceeds first order behavior.

B. Fault Isolation

The core algorithms of the qualitative FDI mechanism are reported in detail in [3]. Fig. 5 shows a high level diagram of the method. A numerical residual, r_n, is computed as the difference between the observed and nominal system behavior.



$$f_2 = \frac{p}{R}$$

$$p = \frac{1}{C}(f_1 - f_2)$$

Fig. 3. One-tank fluid system and model equations.

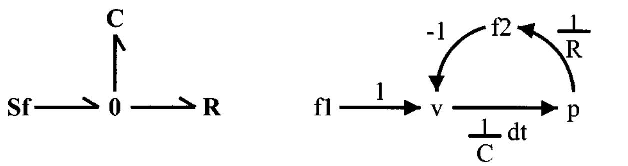


Fig. 4. Bond graph model for the fluid tank (l) and the TCG derived from it (r).

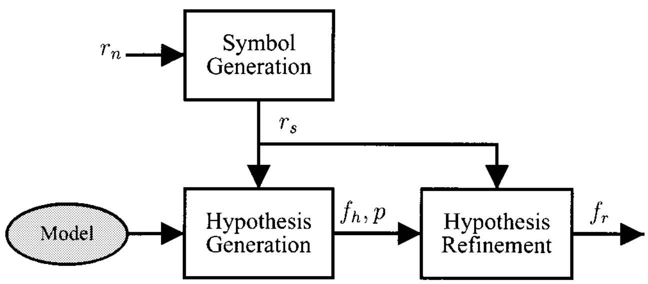


Fig. 5. Qualitative fault isolation core.

This residual is mapped into a symbolic form, r_s, that captures the dynamics of the fault transients. The hypothesis generation algorithm, using the TCG, computes a set of possible fault candidates, f_h, and predicted behavior, p, for each candidate. During hypothesis refinement, features computed from the evolving fault transient are matched against the predicted behavior. Spurious candidates are eliminated, resulting in a final set, f_r. The method is essentially a qualitative parameter estimation method, and applies to those system components that are modeled as parameters. For the tank system, the parameters are the resistance of the pipes and capacity of the tanks.

C. Design and Implementation

The FDI application is implemented on a workstation that appears on the network as another NCAP, albeit without sensing or actuating capabilities. We have taken a rapid application development (RAD) approach using the Python language. Python is an object-oriented, byte-compiled/interpreted language with high-level data types and dynamic semantics. An extension API supports dynamically loadable modules written in compiled languages.

Fig. 6 shows the architecture of the system. In addition to the fault isolation core described above, the application includes modules to communicate with the tank system, compute the residuals, and perform the signal-to-symbol transformation. These modules are described next.

1) Communication With the Tank System: An interface to the tank system was created as a Python extension module. This module was constructed using a sample implementation of the C++ language bindings for IEEE 1451.1, and an automatic code generation tool to construct the interface code for the extension API. The module allows 1451.1 objects that interact with the DMC application to be dynamically created, queried, modified, and destroyed. When the module is loaded in an interactive Python interpreter, a user can explore all 1451.1 function objects accessible on the network.

The client-server and publish-subscribe mechanisms facilitate event driven processing of streaming measurement data. Event handlers in the application create the subscriber objects, and forward the data to the rest of the application in a data-flow computational structure. The fault isolation process uses the fluid level measurement data and the two pressure measurements in the main line. Tracking the mode of the system requires the tank state data and the main node command state data.

As mentioned above, the prototype transducer hardware has some limitations. First, due to various limitations on the network code, the data publishing rate, effectively the sampling rate of the system, is limited to 1 (s). Second, The NCAP does not generate time stamp information for the sensor data. A time stamp is generated by the subscribers in the FDI application, but as a result of the properties of Ethernet, this leads to a small nondeterministic error in the time-stamp value. These limitations affect the residual generation and signal-to-symbol generation steps described below.

2) Residual Generation: The residual vector is computed as the difference between the actual measurement data and the nominal behavior. For the three-tank system, the nominal behavior is known in a direct way because the normal operating cycle is known. For each mode of operation, the nominal behavior for all measurements was determined from data of several cycles of normal system operation, and then mapped onto a parametric estimate using regression analysis. The imprecision in the time stamp of the measurement data increases the variance of the estimates. The parameter values are stored, together with the description of the system model for that mode.

3) Signal-to-Symbol Transformation: The signal-to-symbol transformation component operates on the numerical residual data. Sophisticated techniques have been developed for robust fault detection and robust analysis of the transient dynamics, including derivative estimation and discontinuity detection [10]. However, the low sampling rate does not provide enough data to exploit these techniques effectively. This implies that the signal-to-noise ratio in the experiments must be high enough so that measurement noise can be ignored and the use of naive feature extraction methods is permissible.

A fault is detected when the residual deviates significantly from zero. Fault detection therefore requires a threshold that indicates a significant discrepancy. In this application, a fault is detected based on instantaneous signal value, with the threshold value chosen based on the variance of the signal. An estimate for the derivative of the signal is computed with a simple first-order difference operator. Note that fault detection is adversely affected by the time stamp error in the data. The variance in

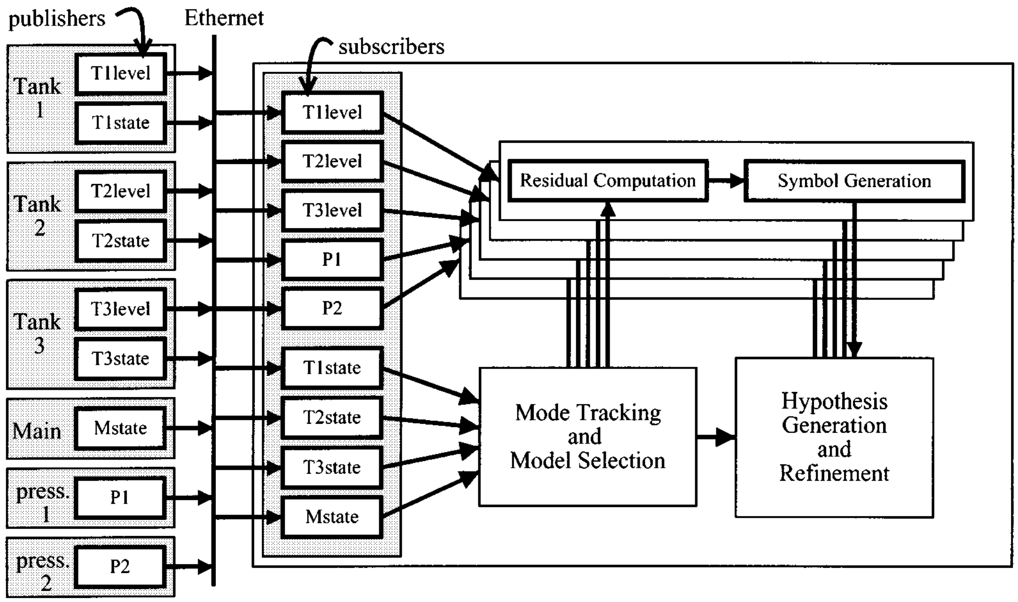


Fig. 6. FDI application architecture, the IEEE 1451.1 publishers are grouped by their corresponding embedded Ethernet controller.

the estimated nominal behavior requires a higher threshold, and thereby decreases the sensitivity of the detection scheme.

The preceding indicates that the transducer limitations result in lower sensitivity for fault detection and a lack of robustness in the computed symbol values. These problems are overcome in the experiments by introducing sufficiently large faults in the system.

D. Online Experiments

The online FDI system was evaluated on two experiments that were set up to make the diagnostic problem as interesting as possible. In the first experiment, a leak is introduced in the main line pipe. Fluid can be drained from the system through an additional valve, fitted between the outlet valve of tank 3 and pressure sensor 2, that is independent from the DMC system (see Fig. 1). The leak is modeled as a change in the resistance of the fluid path through this drain valve, from an infinitely high value when the valve is closed, to a finite, but unknown, value when the valve is opened. Introducing the fault, by opening the valve, results in an immediate pressure decrease, that is observed by pressure sensor 2. The resistance parameter is correctly implicated as the fault.

In the second experiment, an object is dropped in tank 1, which reduces the capacitance parameter of the tank. The diagnosis of an object dropped in a tank that is in a "holding" mode or a into a tank that is filling would be trivial because there is no dynamic behavior involving the tank parameters, and the resulting model would be of zero-order. Therefore, the fault is introduced when the tank is draining. Dropping an object in a tank leads to an immediate increase in fluid level which is detected in the residual, and the subsequent fault isolation correctly implicates the tank capacitance.

IV. CONCLUSION

We have built an online FDI application for a tank-system that is equipped with an advanced DMC system. Two different types of failures were introduced in the system, and successfully detected and isolated using the TRANSCEND framework.

The provisions for both hardware abstraction and interoperability at the application level in IEEE 1451 can reduce the need for specialized knowledge in instrumentation. A DMC application with IEEE 1451 enabled smart transducers in effect becomes a distributed object-oriented system. An application developer can apply modern software engineering practices to manage the complexity of a design, including patterns for distributed applications, and exploit advances in real-time middleware for embedded systems. In this work, we found that the standard also facilitates rapid application development. This is due to the possibility of streaming measurement data and the dynamic configuration of distributed object communication. When these features are combined with an interactive environment such as provided by the Python interpreter, a powerful tool to explore the operation of a DMC system is created.

The work described in this paper represents a first phase in the development of distributed FDI systems. The next step includes exploring other aspects of the FDI functionality that can be moved to a transducer node. Although the diagnosis problem in general requires a global view of system behavior, several aspects allow for local focus. FDI models constructed using compositional modeling techniques will likely play an important role. If global system models can be partitioned into model fragments, it becomes possible to distribute those model fragments on the transducer nodes. The problem of fault isolation across mode switches is being addressed in ongoing research on modeling of hybrid systems. An approach to hybrid model-based FDI for a multitank system is discussed in [11].

Following the work described in this paper, Agilent Technologies has donated the tank system to the Modeling and Analysis of Complex Systems (MACS) Laboratory at Vanderbilt University where its continued development focuses on model-based FDI and fault adaptive control technology (FACT) for complex dynamic systems.

ACKNOWLEDGMENT

The authors would like to thank J. Burch, M. Johnson, J. Liu, and S. Woods, all with the Systems and Solutions Laboratory at Agilent Laboratories, Palo Alto, California, for their helpful suggestions and technical support with various aspects of the three-tank system, and the 1451.1 sample implementation. Vanderbilt University Department of Electrical Engineering and Computer Science gratefully acknowledges Agilent Technologies for the donation of the fluid system as well as the continued technical assistance provided by the Systems and Solutions Laboratory.

REFERENCES

[1] K. Lee, "IEEE 1451: A standard in support of smart transducer networking," in *Proc. 17th IEEE Instrum. Meas. Technol. Conf.*, Baltimore, MD, 2000, pp. 525–528.
[2] R. Isermann, "Process fault detection based on modeling and estimation methods: A survey," *Automatica*, vol. 20, pp. 387–404, 1984.
[3] P. J. Mosterman and G. Biswas, "Diagnosis of continuous valued systems in transient operating regions," *IEEE Trans. Syst., Man, Cybern. A*, vol. 29, pp. 554–565, Nov. 1999.
[4] E.-J. Manders, S. Narasimhan, G. Biswas, and P. J. Mosterman, "A combined qualitative/quantitative approach for fault isolation in continuous dynamic systems," in *Proc. 4th IFAC Symp. Fault Detection Supervision and Safety for Technical Processes*, vol. 1, Budapest, Hungary, June 2000, pp. 1074–1079.
[5] E.-J. Manders, G. Biswas, P. J. Mosterman, L. A. Barford, and R. J. Barnett, "Signal interpretation for monitoring and diagnosis, a cooling system test bed," *IEEE Trans. Instrum. Meas.*, vol. 49, pp. 503–509, June 2000.
[6] *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheets (TEDS) Formats*, IEEE Std 145.2-1997, 1997.
[7] *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Network Capable Application Processor (NCAP) Information Model*, IEEE Std 145.1-1999, 1999.
[8] Wind River Systems, Alameda, CA, http://windriversystems.com
[9] R. L. Fischer and J. Burch, "The PICmicro MCU as an IEEE 1451.2 Compatible Smart Transducer Interface Module (STIM), Application Note AN214a," Microchip Technology Inc., http://www.microchip.com, 2000.
[10] E.-J Manders, P. J. Mosterman, and G. Biswas, "Signal to symbol transformation techniques for robust diagnosis in TRANSCEND," in *Working Papers Tenth Int. Workshop Principles Diagnosis (DX'99)*, Loch Awe, Scotland, June 1999, pp. 155–165.
[11] S. Narasimhan and G. Biswas, "Efficient diagnosis of hybrid systems using models of the supervisory controller," in *Working Papers Twelfth Int. Workshop Principles Diagnosis—(DX'01)*, Via Lattea, Italy, Mar. 2001.

Eric-J. Manders (S'91) received the Ingenieur (Ir.) degree, equivalent to the M.S., in electrical engineering from Delft University of Technology, The Netherlands, in 1991. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN.

His research interests include knowledge based signal processing, model based fault detection and isolation, instrumentation, and embedded systems. He is conducting his graduate research in the area of the coupling of numerical and symbolic methods for signal analysis and interpretation with applications to diagnosis problem solving.

Lee A. Barford (S'82–M'86) received the B.A. degree in mathematics and computer science from Temple University, Philadelphia, PA, in 1982 and the M.Sc. and Ph.D. in computer science from Cornell University, Ithaca, NY, in 1985 and 1987, respectively.

He is a Senior Scientist and Project Manager with Agilent Laboratories, Palo Alto, CA, where he leads research in image processing, dynamical systems theory, statistics, and distributed systems as applied to current industrial problems in industrial inspection, monitoring and diagnosis, and data-driven identification of complex nonlinear systems ranging from manufacturing processes to microwave circuits.

Gautam Biswas (S'72–M'82–SM'91) received the M.S. and Ph.D. degrees in computer science from Michigan State University, East Lansing, in 1979 and 1983, respectively.

He is an Associate Professor in the Department of Electrical Engineering and Computer Science , Vanderbilt University, Nashville, TN. He conducts research in artificial intelligence with primary interests in modeling and analysis of complex systems and their applications to diagnosis, design, and control. He has published in a number of journals and contributed book chapters.

Dr. Biswas is a senior member of the IEEE Computer Society, ACM, AAAI, and the Sigma Xi Research Society.