Presented at the IEEE International Measurement Technology Conference, Baltimore, Maryland, May 1–4, 2000

Diagnosis of a Continuous Dynamic System from Distributed Measurements

Eric-J. Manders[1] and Lee A. Barford[2]
[1]Department of Electrical Engineering and Computer Science, Vanderbilt University,
P.O. Box 1824 Station B, Nashville, Tennessee 37235.
Phone: +1 615 322-2771, fax: +1 615 343-6702, email: manders@vuse.vanderbilt.edu
[2]Agilent Laboratories,
1501 Page Mill Road, MS 4AD, Palo Alto, California 94303-0889.
email: lee_barford@labs.agilent.com

Abstract

A diagnosis application has been built for a three-tank fluid system. The tank system is equiped with a distributed measurement and control system based on smart transducer nodes with embedded computing and networking capabilities that use the IEEE 1451.1 object model to provide high level sensing and actuation abstractions. The diagnosis system operates on-line on a workstation that appears on the network as another transducer node. The diagnosis methodology has several aspects that allow distribution of the monitoring and diagnosis functionality on a network of embedded processors. The current application represents the initial phase in building a truly distributed monitoring and diagnosis application.

1 Introduction

Smart transducers with computing and networking capabilities have started to become commercially available. The embedded processor on a transducer supports complex sensing and actuation tasks, and other high level applications. Combined with the networked communication, this facilitates the construction of distributed measurement and control systems. The advantages of such an approach include increased scalability and potentially also improved robustness.

This paper discusses monitoring and diagnosis of complex continuous dynamic systems in the context of exploiting smart transducer technology. In complex industrial processes, operational safety and reduced down-time are typically ensured by hardware redundancy and localized hardware safety mechanisms (e.g., check valves). To reduce cost, hardware redundancy is increasingly being replaced by functional redundancy techniques. The use of functional redundancy requires a *model* of the system under scrutiny and uses functional relations between system variables to infer discrepancies in measured variables. When such discrepancies occur a fault is detected. Fault detection followed by a fault isolation stage to accurately locate the failing physical component, establishes the fault detection and isolation (FDI) paradigm.

In addition to the benefits of distributed measurement and control systems mentioned earlier, an FDI application may exploit the ability to run application code on an embedded processor. Local signal processing and analysis enables data reduction on the sensor, and thus a decrease in the necessary network bandwidth. In addition, as distributed measurement and control systems grow in size and complexity it will become increasingly important to detect and isolate faults both in the system under observation as well as in the measurement and control system itself. This opens the possibility for diagnostic analysis on the transducer.

Transcend is a framework for model-based diagnosis of continuous dynamic systems based on functional redundancy techniques [8]. The fault isolation algorithms apply qualitative constraint analysis methods that effectively realize a parameter estimation scheme. Model parameters correspond directly to system components and when estimated parameter values deviate from their expected values, the associated components are implicated. The qualitative approach avoids difficulties in the convergence, precision, and computational complexity of established numerical parameter estimation methods, especially when system behavior is non-linear. 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 a secondary sodium cooling loop for a fast breeder reactor [8]. In recent work, the method was applied to the diagnosis of faults in the cooling system of an automotive internal combustion engine [6]. In that application, TRANSCEND has been used off-line on data collected with a personal computer based data acquisition system.

The current work demonstrates TRANSCEND in online operation for a multi-tank fluid system that is fitted with a distributed instrumentation and control system. As such, it is the first step in building a truly distributed FDI system. Several aspects of the TRANSCEND framework translate well to an implementation in a distributed architecture. First, the signal-to-symbol transformation is ideally suited to local processing on the transducer node. The measurement data will often be oversampled to obtain a robust local signal approximation and as a result the symbolic feature data can be decimated in time with respect to the original signal. Second, the TRANSCEND modeling methodology can be extended to include sensor failures.

The remainder of this paper is organized as follows. Section 2 describes the fluid system and the distributed measurement and control system, which is based on the proposed *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators*, IEEE P1451 [4]. Section 3 describes the complete diagnosis application and section 4 presents a summary and conclusions of this work.

2 A test bed for distributed measurement and control

To demonstrate the advantages of using the IEEE 1451 standard to build distributed measurement and control systems, a multi-tank fluid system test bed was designed and built at Agilent Laboratories in Palo Alto, California.

2.1 IEEE P1451

The IEEE P1451 standard consists of a set specifications, in various stages of the standardization process. The 1451.2 standard, first to be accepted, specifies a Smart Transducer Interface Module (STIM), the interface between a sensor or actuator and a microprocessor that provides plug-and-play capabilities at the transducer level [2]. The 1451.1 standard, currently an approved draft document, specifies a Network Capable Application Processor (NCAP) information model, an extensible common object model for sensors and actuators that provides interoperability at the application layer of a distributed measurement and control system [3]. The standard provides programming constructs that support network-neutral communication with both publish-subscribe and client-server mechanisms. A smart transducer node consists of an NCAP/STIM pair, and a physical NCAP contains an 1451.2 interface driver for the communication with the STIM.

The NCAP nodes used in this work are a prototype version of the *Embedded Ethernet Controller* product now available from Agilent Technologies [1]. The controller runs the Wind River Systems VxWorks real-time operating system on a 40 MHz Motorola 68K class microprocessor and has a standard Ethernet interface. The prototype has two limitations that are not present in the actual product. First, the implementation of the network layer imposes a limit on the sampling rate of 1 s, and second, sensor data is not time stamped by the transducer.

2.2 Multi-tank fluid system test bed

The fluid system, shown in Figure 1, consists of three tanks, connected in a series configuration. A closed loop fluid system is created with an additional drain tank, a variable speed electric pump, and additional connecting pipes. Sophisticated fluid control is made possible by the use of many solenoid operated valves in the system.

The fluid flow for each tank is controlled by an inflow, outflow, and bypass valve. Three additional valves in the system provide further control of the fluid flow and pressure in the system. The fluid level in a tank is measured with an ultrasonic level sensor. Additional sensors in the system provide pressure and flow measurements in the fluid line downstream from the pump, and the pressure in the fluid line just following the outlet of tank 3 that leads to the drain tank.

The measurement and control system, also shown in Figure 1, consists of six smart transducer nodes. Each tank has a dedicated node that manages all sensing and control operations specific to that tank, except the control of the bypass valve. A tank node publishes the level sensor measurement data and the valve state data for its tank. Tank control commands are sent from the main node to a tank node via a dedicated client-server connection. The main node subscribes to all the tank level data and valve state data and controls the additional valves in the system. The main node also acquires the sensor data from the flow sensor

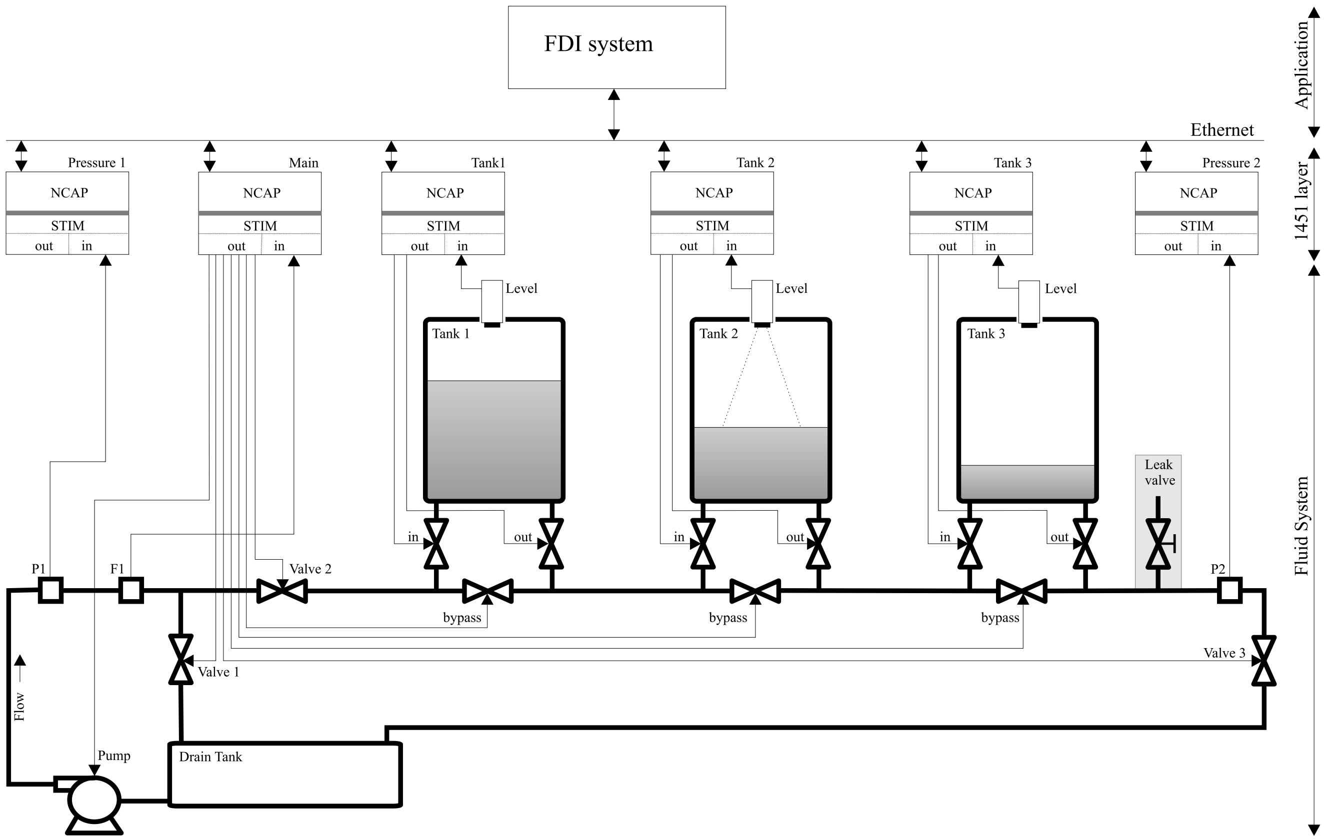

Figure 1. The three-tank fluid system test bed, with the embedded Ethernet controllers, and the FDI application

and controls the speed of the pump through its own STIM. Finally, the two pressure sensors are prototype version of Ethernet-based pressure transmitters from an industrial automation vendor. These transducers integrate an NCAP and a STIM with pressure sensor in a single housing. The pressure transmitters publish the pressure data on the network.

Control of a tank is abstracted into fill, hold, and drain commands that are issued by the main node. A finite state machine application object on the main node controls the sequence of successively filling and draining tanks in a fixed cycle. The tank node controls the actual inflow and outflow valves, using the 1451.2 interface, using feedback from the level sensor to stop and start the filling and draining operations.

3 The three-tank FDI system

The multi-tank fluid system has become a reference application in simulation studies and actual physical system testing for research in automatic control and fault detection and isolation. TRANSCEND simulation experiments for a two-tank fluid system are described in [8].

3.1 Diagnosis model

TRANSCEND diagnosis models capture dynamic characteristics of the dependency relations between component parameters and the measured variables in the form of a *temporal causal graph* (TCG). The diagnosis algorithms are implemented as graph propagation algorithms on the TCG. The models are created with the hybrid bond graph modeling and simulation tool HYBRSIM [9].

The functional relation for flow, f, through a pipe is given by

$$f = \frac{p}{R}, \tag{1}$$

where p is the pressure drop over the pipe and R is the pipe resistance. For a pipe connected to a tank (see

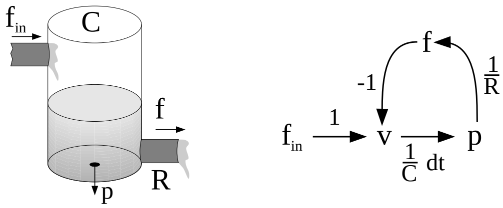

Figure 2. One-tank fluid system (l) and its TCG model (r).

Figure 2), the rate of change in the pressure, $\dot{p}$, at the bottom of the tank is given by

$$\dot{p} = \frac{1}{C}(f_{in} - f), \quad (2)$$

where f_{in} is the flow into the tank and C is the tank capacity. A TCG model for this system can then be constructed based on Equations 1 and 2 (see again Figure 2), where an additional variable v is introduced to represent the $f_{in} - f$ summation.

The pressure at the bottom of the tank is proportional to the fluid level measured by the level sensor. Because TRANSCEND uses symbolic feature values it is in fact not necessary to convert the level measurement values into pressure values. The only flow measurement in the system is not in the outflow pipe of a tank, and thus does not have a straightforward relation with tank pressure.

Because the three-tank system cycles through different modes of operation, the dynamic model changes when switching between modes. Several constraints influenced the construction of the diagnosis model for the complete system. A primary constraint was that the system should be diagnosed while operating in the existing tank control cycle. In each mode of this control cycle, only one tank can be draining, and only one tank can be filling. This means that the dynamics of the model in any mode is always of the first order. In addition, although extensions to the modeling methodology will support this, TRANSCEND currently diagnoses only faults in physical system components, not actuator or sensor failures. For this system that implies that the state of a valve is the state of the valve controller, and that the pump is also not a fault candidate. Therefore, for the current work all failures are those corresponding to parameters of the continuous dynamic system, such as the resistance of a pipe and the capacitance of a tank.

3.2 Design and implementation

The monitoring and diagnosis application is hosted on a LINUX workstation and communicates with the tank system via the 1451.1 model. Thus, the workstation appears on the network as another NCAP, albeit without sensing or actuation capabilities. The 1451.1 standard has C++ language bindings and a sample implementation was developed by Agilent Laboratories. The monitoring and diagnosis application is implemented in Python, an object oriented, byte-compiled/interpreted programming language with high level data types, dynamic semantics, and other features that support rapid application development (RAD) [5]. An extension API supports modules written in compiled languages, and modules for numerical computing, and data visualization are used.

3.2.1 System architecture

Figure 3 shows the architecture of the system. The reference implementation of the TRANSCEND fault detection and isolation algorithms, written in Python, is unchanged from previous work. The system further includes modules to provide 1451.1 capabilities, compute the residuals, and perform the signal-to-symbol transformation. These modules are described in detail in the following sections.

In addition, a graphical user interface (GUI) was created for data visualization and interpretation of the diagnosis results. In off-line use, TRANSCEND generates diagnostic output as a table of active fault hypotheses for each diagnosis step, and the hypothesis refinement process can be interpreted based on this. In the current application the GUI shows this information, while also allowing interactive review of the hypothesis refinement history. Multi channel strip charts display the measurement data and the residual data.

3.2.2 Communication with the tank system

The FDI application uses the fluid level measurement data and valve state data for each tank, the two pressure measurements, and the main node command state data. A subscriber object is created for each of these publications. Because the data is not time stamped by the publisher, it is done by the subscriber when the data message is read from the input queue. This results in a small but non-deterministic error in the time-stamp value. The application also communicates with the main node in a client-server protocol, but this is only used to start and stop the fill/drain system cycle.

The communication between the FDI application and the three tank system is encapsulated in a single

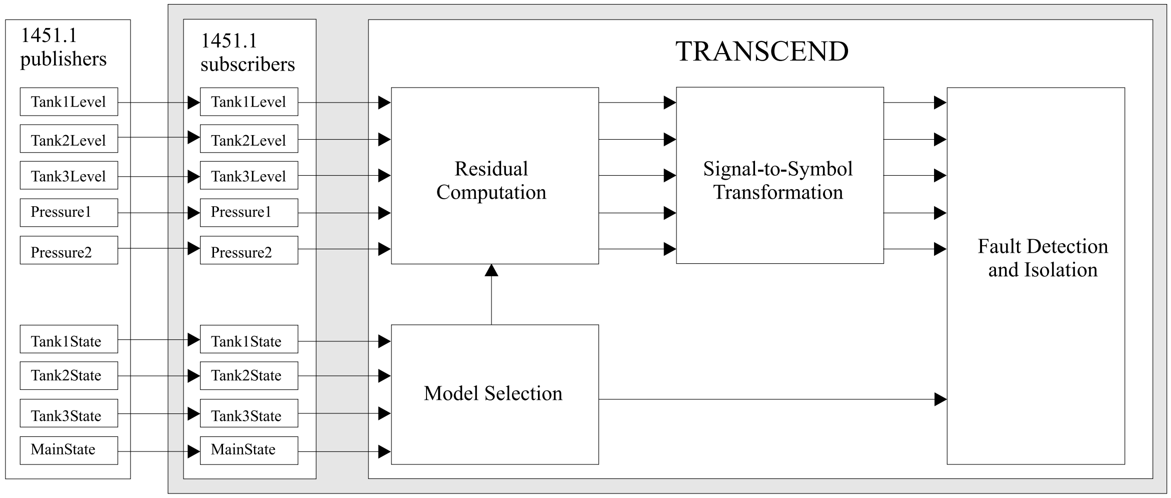

Figure 3. FDI application software architecture

C++ language object that manages all subscriber objects and opens a client-server connection to the tank system main node. A Python extension module is then automatically generated from the C++ class definition file of the interface object with the code generation tool SWIG (Simplified Wrapper and Interface Generator) [5]. The interface class and 1451.1 library are then dynamically loaded by the Python interpreter upon importing the extension module.

This facility provides direct access to the three-tank system, and 1451.1 objects that interact with the system can be interactively created, destroyed, and queried. This feature proved very useful during application development.

3.2.3 Computing residual data

Transcend detects faults as discrepancies between measured variables and their expected values that are determined by the nominal behavior of the system. In general the nominal behavior may be established with an observer scheme that tracks the system behavior and corrects for small deviations [8]. For the three-tank system, the nominal behavior is known in a more direct way because the normal operating cycle is known. The nominal behavior for all the measurements was determined from several cycles of normal system operation, and is stored for each mode of operation with the description of the system model for that mode. Therefore, the residual data is obtained in a straightforward way as the difference between the actual measurements and the nominal behavior.

The lack of precision in the time stamps of the data imposes a limit on the accuracy at which the nominal behavior can be reproduced. A threshold is required to determine when the residual data indicate a significant discrepancy. The time-stamp error effectively results in a low signal to noise ratio for fault detection. The problem is overcome by introducing sufficiently large faults in the system in the experiments.

3.2.4 Signal-to-symbol transformation

The signal-to-symbol transformation component operates on the residual data. Sophisticated techniques have been developed to compute robust estimates of signal derivative values and to detect discontinuous changes in signals [7]. However, the signal-to-symbol transformation methods required for the three-tank system are very basic. The low sampling rate does not provide enough data to use discontinuity detection or slope estimation methods that are based on noise models for the signal. A basic threshold detector is used for fault detection, based on instantaneous signal value, and a simple first order difference operator is used to compute the slope of the signal.

3.3 On-line experiments

The on-line FDI system was evaluated on two experiments that were set up to make the diagnostic problem as interesting as possible. It should be noted that Transcend has been applied to much more complex physical systems than the three-tank system.

In the first experiment, a leakage fault is introduced in the outflow pipe of tank 3 immediately before the pressure sensor. For this purpose a T-junction piece is inserted in the pipe and a ball valve that allows fluid to be drained from the system is connected to the junction (see Figure 1). The leakage fault introduces a change in the resistance parameter of the valve, from an infinitely high value when the valve is closed, to some finite value when the valve is opened. When the valve is opened, an immediate pressure decrease is observed by pressure sensor 2 and 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 is trivial, since the model is of order 0 because there is no dynamic behavior involving the tank parameters. The fault is therefore introduced when the tank is draining. Dropping an object in a tank leads to an immediate increase in fluid level and this leads again to the correct isolation of the fault.

4 Conclusions

We have built a diagnosis application that operates on-line on data from a distributed measurement system. Two different types of failures were introduced in the system, and successfully diagnosed using the Transcend framework. The diagnosis application is an example of a complex application that benefits from the 1451.1 information model. Two aspects of the 1451 standard play an important role in application development. First, the information model enables streaming measurements, where data is available as a stream of continuously sampled data. Second, the data can be accessed from the NCAP nodes through connections that can be dynamically established through the client-server or publish-subscribe methods. When this is combined with an interactive programming environment as is provided by the Python interpreter, a powerful tool to explore the operation of a distributed measurement and control system is created.

At present it is not possible for the system to diagnose a fault whose dynamic response spans a model switch, that is, currently the fault detection and isolation must complete within one operating mode of the system. This is an issue that is being addressed in ongoing research on modeling of hybrid systems.

The work described in this paper represents the first phase in the development of truly distributed fault detection and isolation systems. The next steps in this research include exploring what aspects of the Transcend functionality can be moved to the network of NCAP nodes. Although the diagnosis problem in general requires a global view of the system, several aspects allow for local focus. Transcend models are created using compositional modeling techniques, and models can be partitioned into model fragments. It then becomes possible to distribute the model fragments themselves on the transducer nodes.

Acknowledgments

The authors would like to thank Jeff Burch, Matt Johnson, Jerry Liu, and Stan Woods, all with the Systems and Solutions Laboratory at Agilent Laboratories in Palo Alto, for their helpful suggestions and technical support with various aspects of the three-tank system, and the 1451.1 sample implementation.

References

[1] Agilent Technologies, BFOOT–11501 Embedded Ethernet Controller – Technical Data Sheet (preliminary). http://www.hpie.com/info/BFOOT11501.pdf, Dec. 1999.

[2] 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 1451.2-1997. The IEEE, Inc., New York, NY USA, Sept. 1997.

[3] IEEE Standard for a Smart Transducer Interface for Sensors and Actuators – Network Capable Application Processor (NCAP) Information Model, Approved Draft 1451.1-1999. The IEEE, Inc., New York, NY USA, 1999.

[4] K. Lee and R. Schneeman. Internet-based distributed measurement and control applications. *IEEE Instrumentation & Measurement magazine*, 2(2):23–27, June 1999.

[5] M. Lutz. *Programming Python*. O'Reilly & Associates, Inc., Sebastopol, CA USA, 1996.

[6] E.-J. Manders, G. Biswas, P. J. Mosterman, L. A. Barford, and R. J. Barnett. Signal interpretation for monitoring and diagnosis, a cooling system testbed. *IEEE Trans. on Instrumentation and Measurement*, 2000. To Appear.

[7] E.-J. Manders, P. J. Mosterman, and G. Biswas. Signal to symbol transformation techniques for robust diagnosis in transcend. In *Tenth International Workshop on Principles of Diagnosis*, pages 155–165, Loch Awe, Scotland, 1999.

[8] P. J. Mosterman and G. Biswas. Diagnosis of continuous valued systems in transient operating regions. *IEEE Trans. on Systems, Man and Cybernetics*, 29(6):554–565, 1999.

[9] P. J. Mosterman and G. Biswas. A Java implementation of an environment for hybrid modeling and simulation of physical systems. In *ICBGM99*, pages 157–162, San Francisco, CA USA, Jan. 1999.