Low-Cost Embedded Fault Detection for Safety-Critical Systems

Gary Balas, Mats Heimdahl, Pete Seiler,
Jaideep Srivastava, Mike Whalen, Antonia Zhai
University of Minnesota

1 Introduction

Fault tolerance is vital to ensuring the integrity and availability of safety critical control systems. Fault tolerance encompasses all of the architecture design and algorithms required to guarantee proper system operation during both normal and failure conditions. The architecture design includes the number and placement of sensors, actuators, and processors as well as the data buses required to interconnect all of the system components. A fault tolerant system must also include the logic and algorithms for fault detection, fault diagnosis, fault containment, and reconfiguration to continue operation in face of failures.

Some industries, notably the aircraft industry, have many years of experience designing systems driven by extremely stringent safety requirements. The system availability and integrity requirements for commercial flight control electronics are typically on the order of no more than 10^{-9} catastrophic failures per flight hour. They have converged to a design solution that is based almost exclusively on physical redundancy at all levels of the design. For example, the Boeing 777 has numerous redundant flight surfaces driven by redundant actuators connected to redundant hydraulic systems. Moreover, the control law software is implemented on three primary flight computing modules. Each computing module contains three dissimilar processors with control law software compiled using dissimilar compilers.

Although the use of physical redundancy helps achieve high levels of availability and integrity, this practice dramatically increases system size, complexity, weight, and power consumption. Moreover, such systems are extremely expensive in terms of both the design and development, as well as the unit production costs.

To bring high levels of reliability and integrity to domains that have more stringent resource and cost constraints than the commercials aircraft industry, for example, the automotive industry, there is a significant opportunity to use *model-based* and *data-driven* monitoring techniques to reduce the reliance on physical redundancy. The standard approach is to detect faults using a voting scheme on physically redundant measurements of the same signal. We are investigating an alternative three-pronged approach to fault detection. First, we estimate key signals using sensor measurements, actuator commands, and *models of the physical system being controlled*. Faults in this architecture can be detected by comparing the estimated quantities to the corresponding measured signals thus detecting faults in the physical parts of the system. Second, to detect faults in the cyber (software and hardware) parts of the system, we use monitors derived from model-based software requirements as

well as known limits on the on the physical system (e.g., actuator rate limits). Note here that the two approaches above are suitable for detecting known classes of failure modes. Third, we use data-driven (data-mining and machine-learning) anomaly detection methods to detect unexpected failure modes in the physical (sensors, actuators, and environment) as well a cyber (software and hardware) domains.

Implementing this three-prong approach to fault detection requires significant communication between the various software components; for example, the software controlling the system may have to share all sensor inputs, actuator commands, and substantial internal state information with the model-based estimators, software monitors, and the data-driven monitor. Naturally, this data sharing must not unduly (or at all) affect the behavior of the application being monitored, and must be high speed and low latency. This would be difficult, if not impossible, with existing software and computing architectures. Thus, we are also investigating the use of the rapidly emerging multi-core processor architectures as a basis for our approach to low-cost embedded fault detection. We plan to use cores for the execution of the control application as well as the various monitoring applications.

In summary, the path to low-cost embedded fault detection is to develop algorithms and computing architectures for fault detection without relying on costly physical redundancy of components.

2 Key Challenges

Model-based Fault Detection: The problem of model based signal estimation is fundamental to control theory and signal processing. The purpose of the estimator is to estimate the states of the plant and to estimate the current best model of the plant from a set of models representing nominal and faulted behaviors. This is a challenging problem since there are several sources of model uncertainty and the models typically have nonlinearities. We are expanding the algorithms for robust filtering and thresholding to nonlinear systems by exploiting the parameter-varying modeling framework.

Software Fault Detection: We are adopting advances in model-based software development for fault detection of software and hardware components of the system. We are developing techniques to model and deploy safety monitors and interlocks, and run-time assertions. In particular, we are using the artifacts from a model-based development process as well as formalized software and safety requirements. Assuming the existence of such artifacts is reasonable given the increased use of model-based software development across many industries—including automotive.

Data-driven Anomaly Detection: Model-based techniques can be designed to detect known failure modes but their performance degrades when unexpected failure modes are encountered. As a complement, we are developing data driven anomaly detection methods. There are two approaches within the data driven anomaly detection module: (1) anomaly detection from individual sensor or component data and (2) anomaly detection methods for complex systems. Both approaches work by looking for deviations from normal operating behavior. The conceptual difference is that the first approach looks for anomalous behavior of a single component signal while the second approach looks for anomalous behavior within a collection of signals which are related by some functional behavior.

Multi-Core Architectures For Fault Detection: With the emergence of multi-core processors, we are developing hardware and compiler architectures for executing the control law application and fault detection monitors on the same chip, but separate cores. The architecture must support non-intrusive, predictable, fine-grained, and highly flexible exchange of data between the application and its monitors. To extract events of interests from an application program and communicate the relevant portions of the state to the monitor(s), we are developing *monitoring-aware* compilers coupled with *novel architectural enhancements* to the multi-core architectures.

3 Summary

To achieve low-cost fault detection, we are developing a novel approach that blends monitoring algorithms with complementary characteristics, and we are leveraging advances in multi-core processors to enable implementation of these novel fault detection approaches.

Our results will enable fault detection schemes that are not reliant on physical redundancy; thus reducing the size, weight, power consumption, and cost associated with highly reliable systems.

4 Short Author Bios

Mats Heimdahl: Dr. Mats Heimdahl earned an M.S. in Computer Science and Engineering from the Royal Institute of Technology in Stockholm, Sweden and a Ph.D. in Information and Computer Science from the University of California at Irvine. He is currently a Full Professor of Computer Science and Engineering at the University of Minnesota, the Director of the University of Minnesota Software Engineering Center (UMSEC), and the Director of Graduate Studies for the Master of Science in Software Engineering program. His research interests include system and software safety, software verification and validation, software certification, and model based software development.

Michael Whalen: Dr. Michael Whalen is the Program Director at the University of Minnesota Software Engineering Center. Dr. Whalen is interested in formal analysis, testing, and requirements engineering. He has developed simulation, translation, testing, and formal analysis tools for Model-Based Development languages including Simulink, Stateflow, SCADE, and RSML^{-e}, and has published more than 30 papers on these topics.