Microprocessors and Microsystems 35 (2011) 453–461

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

A fuzzy predictive redundancy system for fault-tolerance of *x*-by-wire systems

Man Ho Kim [a], Suk Lee [a], Kyung Chang Lee [b,*]

[a] School of Mechanical Engineering, Pusan National University, Busan, South Korea
[b] Department of Control and Instrumentation Engineering, Pukyong National University, Busan, South Korea

ARTICLE INFO

Article history:
Available online 30 April 2011

Keywords:
Safety critical systems
Fault-tolerant system
x-by-wire systems
Intelligent vehicle
Fuzzy predictive redundancy system
Fault-detection algorithm
Threshold prediction method
Threshold level calculation method
Embedded microcontroller unit

ABSTRACT

Safety issues and the dependence of numerous systems on electronics are rapidly increasing the concern over fault-tolerance. As an example, an intelligent vehicle with electronically controlled *x*-by-wire systems composed of dynamically configurable electronic elements instead of rigid mechanical components must be fault tolerant because a devastating failure could occur without warning. In particular, a safety-related malfunction of the brakes, throttle, or steering system could lead to serious injury or death and damage the manufacturer's reputation. If there is a warning it may not be as devastating as one could prevent it or mitigate it. Therefore, fault-tolerance is the primary focus of *x*-by-wire systems development. To address this concern, this paper presents a fuzzy predictive redundancy system that can remove most erroneous faults with a fault-detection algorithm. This paper also introduces a prototype of the system using an embedded microcontroller unit to show that it outperforms well-known average and median voters. The experimental results show that fuzzy predictive redundancy can be an appropriate choice for fault-tolerance in the *x*-by-wire systems such as steer-by-wire system or brake-by-wire system of intelligent vehicle.

1. Introduction

Recent interest has focused on intelligent vehicles that offer the potential of significantly enhanced safety and convenience for both drivers and passengers [1,2]. As a component of an intelligent vehicle, in-vehicle network (IVN) systems, in which electronic components such as window motors and switches are connected to an electronic control unit (ECU) through a shared network cable [3,4], are widely used in automobiles, trucks, public transportation, and industrial vehicles. Especially, *x*-by-wire systems, which are replacing rigid mechanical components with dynamically configurable electronic elements and digital communication networks, are being developed to expand the application of IVN systems to real-time components such as brakes, throttle, and steering systems [5–7]. These drive-by-wire or *x*-by-wire systems expand the intelligent functions of safety algorithms, including adaptive cruise control systems and lane-keeping assist systems.

However, *x*-by-wire systems require a higher level of fault-tolerance than traditional systems based on mechanical links, because mechanical systems can provide drivers with some warning feedback of their status while electronic systems tend to fail abruptly without any warning. Because a safety-related malfunction of the brakes, throttle, or steering wheel could lead to injuries or deaths of the vehicle occupants and damage the manufacturer's reputation, fault-tolerance is the principal focus of *x*-by-wire systems [7–9].

The design of fault-tolerant functions generally includes redundant systems that duplicate several modules such as actuators, microcontrollers, and sensors with the same function. In general, a redundant system is classified into hardware redundancy, analytical redundancy (so called software redundancy), and information redundancy [10,11]. The hardware redundancy is to add an extra embedded hardware with the same functions implemented in the original hardware [12–20]. The analytical redundancy consists of a hardware module and multiple model-based analytical models that execute a given set of functions of the original hardware [21–23]. The information redundancy is to add extra information such as a parity bit to detect any fault. Among these approaches, due to downward trend of microcontroller costs, the hardware redundancy has been the center of research for many fault-tolerant systems such as intelligent vehicles.

Hardware redundancy systems can be classified according to the architecture and function: static hardware redundancy, dynamic hardware redundancy, and hybrid hardware redundancy [11]. A static redundancy system requires a voter that determines the final output of the system using the majority [12,13], median [14], mid-value [15], average, or a weighted rule [16,17] as its fault-masking algorithm to isolate any faulty inputs. However, static redundancy tends to cost more because it requires at least three parallel modules and detecting faults is difficult when two or more modules are faulty. A dynamic redundancy system achieves

* Corresponding author. Tel.: +82 51 629 6332; fax: +82 51 629 6309.
E-mail address: gclee@pknu.ac.kr (K.C. Lee).

0141-9331/$ - see front matter
doi:10.1016/j.micpro.2011.04.003

fault-tolerance by having fault detection and reconfiguration functions instead of a voter. In general, a dynamic redundancy system can be classified as hot or cold depending on whether all modules are always operating. A hot standby dynamic redundancy system uses two modules to determine outputs, and the fault detector determines which module is correct and the reconfigurator selects either of the two modules using an output switch. A cold standby dynamic redundancy system uses only one module at a time, and the reconfiguration module controls two switches to block the signal from the faulty module. By combining the two approaches, a hybrid redundancy system using techniques such as self-purging redundancy [18] or a smoothing voter [19] can mask a fault just like the static approach, but it can also detect a fault and reconfigure the system just like the dynamic approach using a switch and a fault-detection algorithm [17,20]. A hybrid redundancy system is simpler than a dynamic redundancy system because it requires no reconfiguration algorithm, and it is more cost effective than a static redundancy system because it requires fewer extra modules.

In order to implement more effective and cost efficient hardware type of the hybrid redundancy system, this paper presents a fuzzy predictive redundancy system. Also, this paper suggests a fault-detection algorithm to detect and isolate faulty elements from a signal by forecasting the change from the last value of the input signal using the threshold prediction and threshold level calculation method. Because many sensor signals or the control output of a controller may follow a trend such as a sine wave or step response in real industrial applications such as *x*-by-wire systems, and will change abruptly when it fails [23], fault-detection algorithms based on the threshold prediction and threshold level calculation method may be appropriate for detecting and isolating faulty signals. Finally, to verify the feasibility of fuzzy predictive redundancy system, we developed an experimental fuzzy predictive redundancy system using an embedded microcontroller unit with an experimental setup to simulate a redundant brake pedal signal, and compared the performance of the fuzzy predictive redundancy.

The remainder of this paper is organized as follows. Section 2 describes the structure of the fuzzy predictive redundancy system along with the fault-detection algorithm using the threshold prediction and threshold level calculation method. Section 3 describes the implementation details and experimental results of the fuzzy predictive redundancy system using an embedded microcontroller unit. Section 4 presents the conclusions.

2. Structure of the fuzzy predictive redundancy system

2.1. Schematic diagram of the fuzzy predictive redundancy system

We propose the fuzzy predictive redundancy system shown in Fig. 1 to enhance the redundancy of an *x*-by-wire system of intelligent vehicle. The rationale for having this type of structure is that we cannot afford many redundant sensors on a system such as a passenger car. Therefore, the cost and complexity of static or *n*-modular redundancy with spares are difficult to justify. However, we can consider the hybrid of hot standby dynamic redundancy and static redundancy along with some capability to detect faulty elements. The fuzzy predictive redundancy system makes use of a powerful microcontroller to detect fault from the last value of the system output using the threshold prediction and threshold level calculation method. The threshold prediction method is used to determine if a fault exists in the current value of the sensor signals by checking whether the new inputs lie within the predicted interval. The threshold level calculation method is used to determine the appropriate threshold levels of the fault-detection algorithm dynamically. The fuzzy predictive redundancy system is based on the assumption that the variables being measured change relatively gradually without large fluctuations.

Fig. 1 shows the schematic diagram of the fuzzy predictive redundancy system with five modules: a threshold predictor, a threshold level calculator, a fault detector, an exception handler, and a voter. First, the threshold predictor forecasts a threshold $FT(k)$, which is essentially the expected change in the output signal. This threshold is calculated using the exponential smoothing method, which is explained later. Second, the threshold level calculator computes a threshold levels β_i and β_j, which is essentially the expected threshold range in the output signal. This threshold level is computed using the fuzzy logic, which is explained later. Third, the fault detector decides whether a fault exists in the two input values $a_i(k)$ and $a_j(k)$ using the fault-detection algorithm. Here, if an input value lies within an interval centered on the system output in the previous step $a(k-1)$, it decides that the input is error-free. Fourth, the exception handler, determines an output value $a(k)$ when the fault detector decides that both input values are unreliable. When some unexpected external disturbance affects the system, the fault detector decides that both inputs are erroneous and that a valid output is unavailable. When that happens, the exception handler synthesizes a plausible output value by incrementing the last valid output value to prevent abnormal operation due to the absence of an actual output value. Finally, the voter calculates the output value using an averaging method.

2.2. Functions of sub-modules for the fuzzy predictive redundancy system

The threshold predictor must first forecast a threshold for determining whether a fault exists in the values supplied to the fault detector. The double exponential smoothing method, which is a representative method of time-series forecasting methodology, was chosen to forecast such a threshold in the fuzzy predictive redundancy system. The double exponential smoothing method

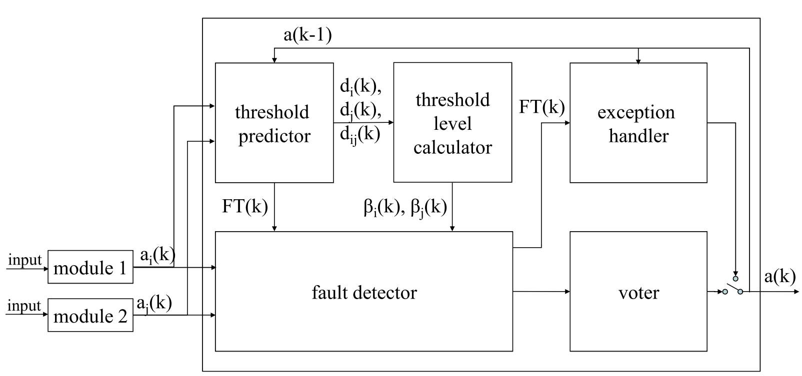

Fig. 1. Schematic diagram of the fuzzy predictive redundancy system.