15th International Conference on
AEROSPACE SCIENCES & AVIATION TECHNOLOGY,
ASAT - 15 – May 28 - 30, 2013, Email: asat@mtc.edu.eg ,
Military Technical College, Kobry Elkobbah, Cairo, Egypt,
Tel: +(202) 24025292 –24036138, Fax: +(202) 22621908

A Novel Voting Algorithm Based on Weighted Average Voting with a Classification Technique

M. A. Kassab*, A. Hashad†, H. S. Taha‡, and S. A. Shedied‡

Abstract: Weighted average voting has been widely used in control and computing systems. A novel voting algorithm based on combining the weighted average voter (WAV) with the minimum distance classification (MDC) technique. This voting algorithm uses WAV to indicate the minimum distance that will be used in the classification technique to select one measurement as the voter output. It is applicable for handling the output of an array of skewed sensors in safety related applications. The performance of the proposed voter is evaluated through a series of fault injection experiments and the results are compared with those of the exact majority voter [1], the WAV [2] and the enhanced weighted average voter (EWAV) [3]. The experimental results showed that the novel voter gives better performance in terms of reliability and complexity than the three other voters.

Keywords: Sensor redundancy, voting algorithms, reliability enhancement

1. Introduction

Many real-time fault tolerant systems such as safety-critical computer control systems (e.g. flight control systems and patient monitoring equipment), highly reliable applications (e.g. railway-interlocking systems), and highly available systems (e.g. file server applications, and call processing applications) use sensor redundancy to increase the system reliability (system availability which means that the system ability to produce the required outputs within real time at certain moment with no errors). Thus voting algorithms should be used to manage these redundant sensors, and then the objective is to choose the better voting algorithm (selects or computes the more accurate measurement) to eliminate any perturbation before using in the control loop when selecting this measurement [4]. Triple Modular Redundancy (TMR) and 3-Version Programming (3VP) are commonly used techniques for masking runtime faults at hardware and software levels, respectively [5, 6]. In both techniques, the output from three independently developed but functionally identical modules operating in parallel with the same inputs are supplied to a voting unit that arbitrates between them to produce an overall output, and we are interested here in TMR (See Figure 1).

The proposed voting algorithm differs from the WAV and the EWAV as it selects only one input to be the voter output, this selection depends on the distance between each input and the output from the WAV.

* MSc student, mohamedkassab99@gmail.com
† Prof. AAST, Egypt.
‡ Egyptian Armed Forces, Egypt.

In safety critical systems, it is necessary to consider the behavior of voters in worst case conditions (the majority of sensors are not valid simultaneously). Such events may be improbable but can have catastrophic consequences [1]. Thus multiple error scenarios would be considered in this analysis.

Background on the famous voting algorithms and experimental evaluation of fault-tolerant mechanisms are described in Section 2. The proposed voter implementation is presented in Section 3. The fault injection approach and experimental results are presented in Section 4. Some conclusions are presented in Section 5.

2. Background

2.1 Exact Voting

An input is selected by the voter as an output if and only if it is in exact agreement with a majority (at least $(n+1)/2$) of the other inputs [1] where n is the number of voter inputs. Different approaches on implementation of such voters are described in [7].

There are two main problems with the exact voting approach, the first is that the voter has an output if and only if there is a majority and if there is no majority then there is no output, the second is the need of additional algorithm to overcome the first problem.

2.2 Inexact Voting

Several approaches to voting on correct inputs with similar but not identical, have been published. Formalized definitions of the Majority, Plurality, Weighted Average and Median voting algorithms are presented in [2].

The formalized majority voter constructs sets of those variant inputs each set contains the inputs that differ from each other by less than threshold value (ε, the error value that can be accepted from the voter by the system). If such a set is found to contain a majority ((n+1)/2) of variant inputs, one of the inputs of this set is chosen at random to form the voter output.

The formalized Plurality voter is a generalization of formalized majority voter, in which the need for a strict majority is relaxed. For example, in a five modular redundant system, 2-out of 5 agreements may be considered acceptable. It is claimed that plurality voting is equivalent to formalized majority voting in triple modular redundant configurations[1].

The weighted average voter in which the mean value of variant inputs is calculated, this mean value may be distinct from all the variant results. The calculation uses weights associated with each variant input. The weights can be calculated in a number of ways; the approach adopted here is that they are equals.

The generalized median voter that assumes the output space is a metric space. It repeatedly selects and discards those pairs of variant inputs with maximum reciprocal distance. This process leaves a single variant input (assuming the number of variants is odd) which is considered as the Median result and is selected as the voter output.

3. The Proposed Voting Algorithm

The proposed voting algorithm uses the WAV (with equal weights) to calculate the value that will be the reference value to the classification technique that will be used here, which based on choosing the input value that has the minimum distance from the reference value (calculated by the WAV) as the voter output.

As shown in the WAV (with equal weights) output (av) will be the usual mean of the three inputs due to the equal weights as follows:

$$av = \frac{x_1 + x_2 + x_3}{3} \tag{1}$$

Then the distances between each input and the average is calculated as follows:

$$\begin{aligned} d_{1av} &= |x_1 - av| \\ d_{2av} &= |x_2 - av| \\ d_{3av} &= |x_3 - av| \end{aligned} \tag{2}$$

Then the voter output will be as follows:

$$\begin{aligned} &if\ d_{1av} \le d_{2av}\ and\ d_{1av} \le d_{3av}, then\ y = x_1 \\ &else\ if\ d_{2av} \le d_{1av}\ and\ d_{2av} \le d_{3av}, then\ y = x_2 \\ &else\ y = x_3 \end{aligned} \tag{3}$$

4. Experimental Results

For TMR-based highly-safe systems, incorrect outputs can result in catastrophic consequences. The main objective in designing of such systems is therefore minimizing the number of incorrect outputs obtained from the system voter during the operation time [8, 9]. The ratio of incorrect voter outputs to the number of voting actions (within a test window) can be defined as a system safety measure [3]. This criterion is used in this paper to compare the performance of the voters in terms of reliability.

Also complexity will be calculated for each voter to indicate the execution time that will be needed by each voter.

Simulation time of 64.01 sec with $\varepsilon = 0.1$ and voter inputs would be updated every 10 m sec are assumed. Thus we will have 6401 voting actions in one process.

Two phases (one phase for small error amplitude ($|E| < 0.45$), the other for large error amplitude ($|E| < 2.7$)) are presented, each phase contains 11 processes each process with different error amplitude.

Four error configurations would be applied to the sensor measurements as shown in Figure 3:

The first configuration in this figure presents the probability that the failed sensors produce erroneous measurements that deviates from the notional measurement with positive increasing;

The second one in this figure presents the probability that the failed sensors produce erroneous measurements that deviates from the notional measurement with negative increasing;

The third one in this figure presents the probability that one of the failed sensors produces erroneous measurements that deviates from the notional measurement with positive increasing while the other produces erroneous measurements that deviates from the notional measurement with negative increasing (the amplitude of positive increasing is more than the amplitude of negative increasing).;

The fourth one in this figure presents the same behavior as the third one but with exchanging the error amplitude of the two failed sensors (the amplitude of positive increasing is less than the amplitude of negative increasing).

From experiments it is cleared that the first and the second configurations have the same outputs. And the third and fourth configurations also have the same output.

For each configuration there are three stages (See Figure 3) as follows:

- The first is the three sensors are valid (there are errors but less than ε).
- The second is only two sensors are valid.
- The third is only one sensor is valid.

All the above cases would be investigated to compare the behavior of the voters in terms of reliability.

4.1 Reliability Analysis

4.1.1 First phase

In this phase comparing voters for small errors will be investigated. The ratio of incorrect voters outputs are shown in Figure 4 versus the amplitude of injected errors$|E|$.

It is cleared that the proposed voter improves the system reliability in the four error configurations by decreasing the number of incorrect voter output in comparing with other voters (See Figure 4). The left hand side in Figure 4 represents the ratio of incorrect voter outputs versus the amplitude of injected errors in case of the first and the second configurations (See Figure 3). While the right hand side in Figure 4 represents the ratio of incorrect voter outputs versus the amplitude of injected errors in case of the third and the fourth configurations (See Figure 3).

The best voter in the system safety measure is the proposed voter, the EWAV, the WAV, then the majority voter.

4.1.2 Second phase

In this phase comparing voters for large errors will be investigated. The ratio of incorrect voters outputs are shown in Figure 5 versus the amplitude of injected errors$|E|$.

It is also cleared that the proposed voter improves the system reliability in the four error configurations by decreasing the number of incorrect voter output in comparing with other voters (See Figure 5). The left hand side in Figure 5 represents the ratio of incorrect voter outputs versus the amplitude of injected errors in case of the first and the second configurations (See Figure 3). While the right hand side in Figure 5 represents the ratio of incorrect voter outputs versus the amplitude of injected errors in case of the third and the fourth configurations (See Figure 3).

Like the first phase the best voter in the system safety measure is the proposed voter, the EWAV, the WAV, then the majority voter.

4.2 Complexity Analysis

Complexity can be measured in terms of the number of comparisons required for each voting action, CPU time, and the required memory resources [3].

As shown in Table 1 the best voter in the number of comparisons is the WAV, the proposed voter, the EWAV, then the majority voter.

The best one in CPU time is the WAV, the proposed voter, the majority voter, then the EWAV.

The best one in required memory locations is the WAV, the proposed and the majority voter, then the EWAV.

5. Conclusions

This paper introduces a new voting algorithm that has some benefits over the three mentioned voters. Firstly, it overcomes the problem of having no majority that exists in the majority voter. Secondly, it overcomes the problem that exists in the WAV and the EWAV which is taking in consideration all inputs values even the severe outliers (the inputs that deviates from the notional value with very large distance) even with small weights but still included in the calculations of the voter output. Thirdly, increasing the reliability of the system by improving the system safety measure (the ratio of the number of incorrect voter outputs to the total voting actions). Finally, offering the less complexity voting algorithm than the majority and the EWAV.

6. References

[1] Bass, J., G. Latif-Shabgahi, and S. Bennett; Experimental comparison of voting algorithms in cases of disagreement, in EUROMICRO 97. New Frontiers of Information Technology., Proceedings of the 23rd EUROMICRO Conference, 1997, IEEE.
[2] Lorczak, P. R., A. K. Caglayan, and D. E. Eckhardt; A theoretical investigation of generalized voters for redundant systems, in Fault-Tolerant Computing, 1989, FTCS-19 Digest of Papers., Nineteenth International Symposium, 1989. IEEE.
[3] Latif-Shabgahi, G., A novel algorithm for weighted average voting used in fault tolerant computing systems, Microprocessors and Microsystems, 2004, 28(7): pp. 357-361.
[4] Berdjag, D. ; Zolghadri, Ali ; Cieslak, J. ; Goupil, P., Fault detection and isolation for redundant aircraft sensors. in Control and Fault-Tolerant Systems (SysTol), 2010 Conference, 2010. IEEE.
[5] Johnson, B.W., Design and analysis of fault-tolerant digital systems, 1989, Addison-Wesley, New York: pp. 439-451.
[6] Pullum, L. L., Software fault tolerance techniques and implementation, 2001: Artech House Publishers.
[7] Parhami, B., Voting networks, Reliability, IEEE Transactions, 1991, 40(3): pp. 380-394.
[8] Bennett, S. and G. Latif-Shabgahi. Evaluation the performance of voting algorithms used in fault tolerant control systems, Proc. 14th World Congress of International Federation of Automatic Control, vol. Q, Beijing, China. 1999.

[9] Latif-Shabgahi, G., S. Bennett, and J. Bass, Smoothing voter: a novel voting algorithm for handling multiple errors in fault-tolerant control systems, Microprocessors and Microsystems, 2003, 27(7): pp. 303-313.

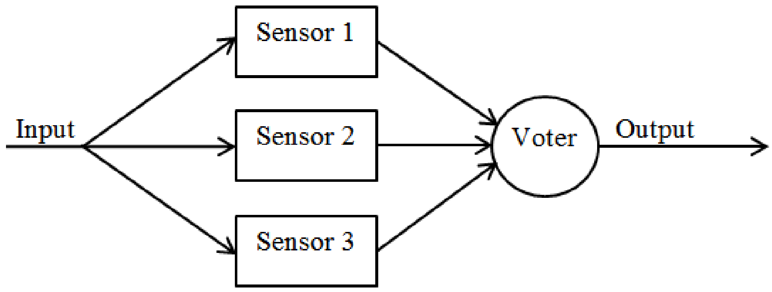

Figure 1. Triple modular redundancy

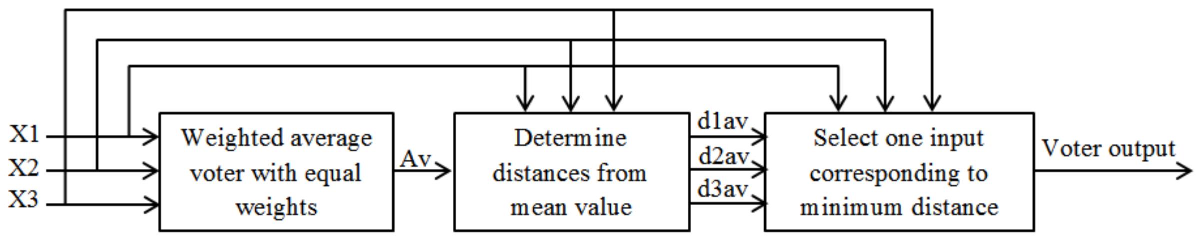

Figure 2. The proposed voter

Table 1 Complexity comparison

	Number of comp. for each voting action	CPU time	Required memory
majority voter	Min. 3 Max. 9	2.2272 µ Sec	8 memory locations
WAV (equal weights)	0	1.6325 µ Sec	4 memory locations
EWAV	Min. 3 Max. 6	6.6183 µ Sec	15 memory locations
proposed voter	Min. 2 Max. 4	2.2017 µ Sec	8 memory locations

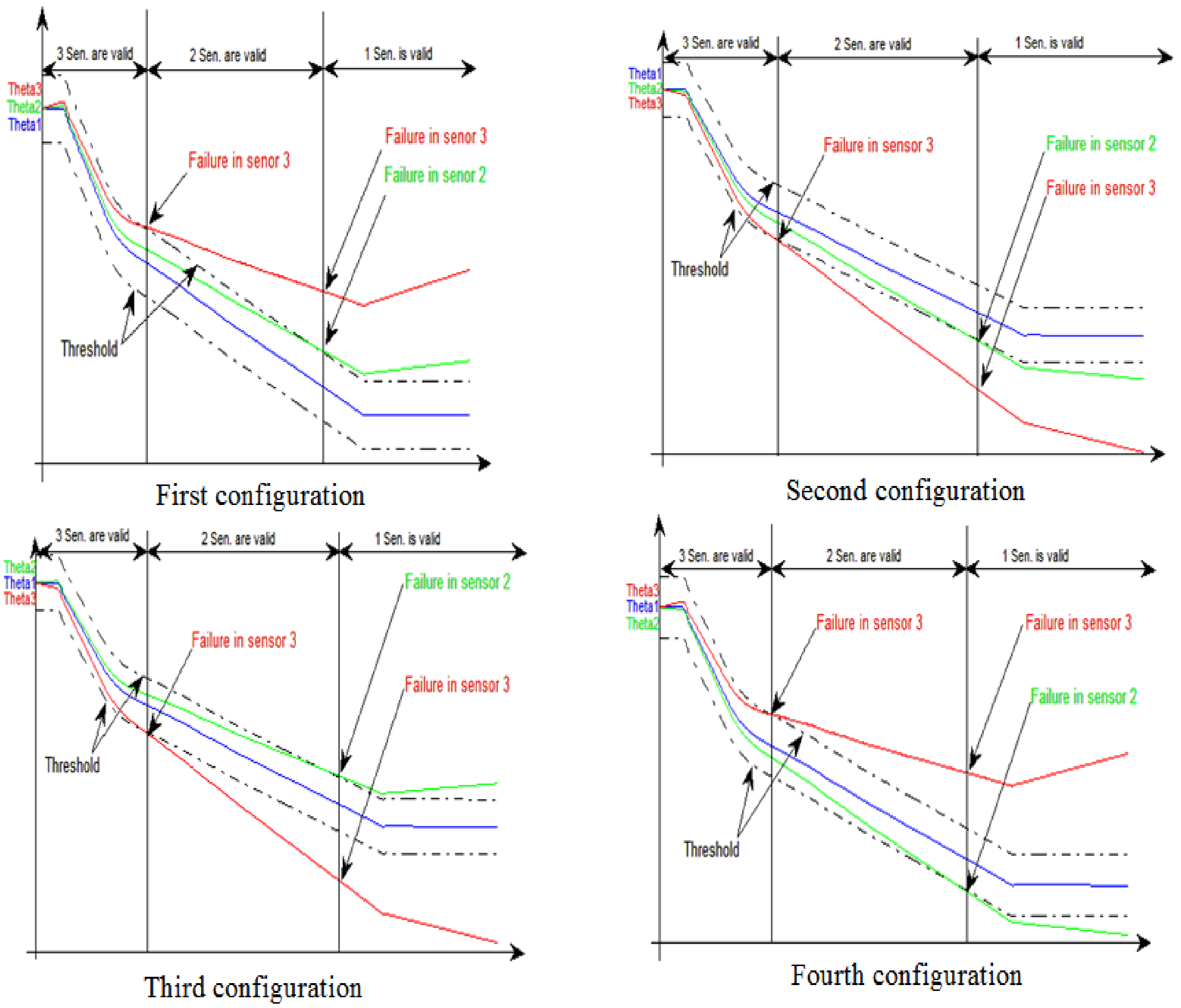

Figure 3 Error configurations

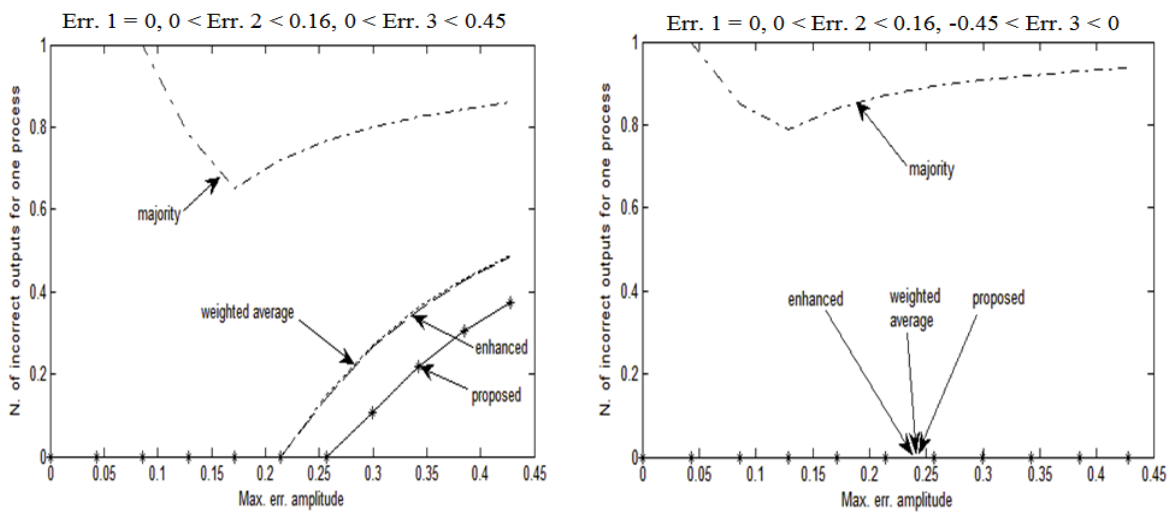

Figure 4. Phase 1

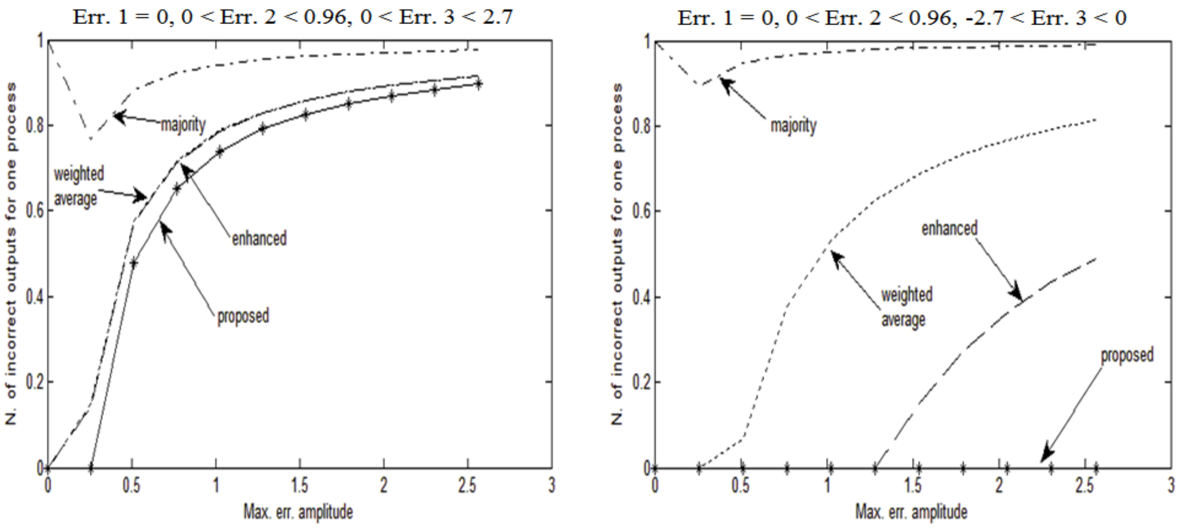

Figure 5. Phase 2