

An Incorporated Voting Strategy on Majority and Score-based Fuzzy Voting Algorithms for Safety-Critical Systems

B. Umamaheswararao
Dept. of CS & SE
Andhra University
Visakhapatnam, India

P. Seetharamaiah
Dept. of CS & SE
Andhra University
Visakhapatnam, India

S. Phanikumar
Dept. of Computer Sciences
GITAM University
Hyderabad, India

ABSTRACT

Triple Modular Redundancy (TMR) is usually used to increase safety and the reliability of safety-critical systems where three identical segments are used in identical and the ultimate outcome is reached using voting techniques. Fault masking is one of the main techniques to improve the normal actions of a range of safety-critical techniques. Some commercial areas which implement such techniques include process control, transport, and atomic power plant and army programs. Integrated Voter for majority and weighted-average used to provide for a fault masking capability in safety-critical systems [1]. The Majority voting gives a high level of safety and the weighted-average offers a good level of availability. If integrated these two gives a good level of safety in Majority voting not in integrated voting. Here propose a new voting algorithm for faulty masking taking the disadvantage of previous algorithm. In this Incorporating Majority voting and score based fuzzy voting schemes. Safety performance is evaluated by running proposed, Majority and score based fuzzy voting on a triple modular redundant (TMR) system for 10000 voting cycles in various error scenarios. Experimental results show that proposed fuzzy Voter is given a higher safety than other two voting algorithms.

Keywords

Fuzzy Voters, Safety-Critical Systems, Triple Modular Redundant (TMR), Fault Tolerance, Fault masking.

1. INTRODUCTION

A safety-critical system is one that has the prospective to cause injuries. Safety-critical systems are the systems which may cause to risks, loss of lives or great harm to the property if they do not succeed. In the past 30 years, there is substantial evidence that software flaws can contribute to accidents and failures involving safety-critical systems, e.g., Therac-25 [2] and Ariane5 [5], Boeing 777-200 (registered 9M-MRG) [6] and the Toyota Prius1 and recently Malaysia MH370 missing.

The development of a safety-critical system is generally expensive, higher the complexity and safety integrity, the higher the expected cost. There are different domains in which safety-critical control systems are used: (automotives) drive-by-wire systems, brake by wire systems used in cars; (medicine) infusion pumps, cancer radiation therapy machines, etc.; (military and space applications) rocket launchers, satellite launchers, etc.; and (industrial process control) robotics and consumer electronic appliances. The reliability, availability and safety can be increased in all these applications. Faults that occur in these applications may lead to hazardous situations [13]. If a single module or channel is

used and when it becomes faulty due to some noise the system may fail and hazard may occur. Hence N-modular redundancy or N-version programming along with voting technique is used to mask the faults in the faulty environments [3, 4]. Different voting strategies have been introduced to mask the faults and provide safety and availability. Some Voting techniques like Majority, Plurality Voters produce the outcome if most or required variety of information to the Voter are matched; otherwise it will produce no outcome so that the system can be taken to the don't succeed secure condition [7]. Adaptive Majority voting algorithm gives better efficiency by using recorded information. But for some safety-critical systems, there may not be any don't succeed secure condition [8]. Median, average, weighted-average Voters are some illustrations for the Voters, which amalgamate the information of the Voter and produce some value as the voter outcome [9]. Two conventional and commonly used voting methods are the majority and fuzzy weighted-average voting with score. In its general form, an inexact majority voter [10] produces a correct output if the majority of its inputs match each other. This fuzzy weighted-average voting with scores gives greater safety as opposed to weighted-average voting methods, but reduced safety principles in comparison to the inexact majority voting, at the existence of large errors. At the existence of small errors Fuzzy voting provide reduced protection principles than the other formerly stated weighted-average voting methods and greater safety as opposed to inexact majority voting criteria. The availability provided by fuzzy voting criteria is also less than the other calculated weighted-average voting methods. Incorporated voting algorithm gives higher safety than fuzzy voting and majority voting algorithms.

Safety = $(1 - n_{ic} / n)$ and Availability = n_c / n .

This Research Paper is organized as follows. Section 2 is the literature survey of Inexact Majority voting and fuzzy weighted-average voting with scores algorithms. In Section 3, Proposed Incorporated voting algorithm. In Section 4, Experimental method and Test Harness is described. In Section 5, Experimental results are analyzed. In Section 6, Conclusions and Future works are given.

2. RELATED WORKS

The standard majority and fuzzy weighted-average voter with scores cases of two exclusive groups of voting techniques. One with an innovative level of safety yet with a low level of availability and the other with a low level of safety compare to the majority voting yet an innovative level of availability.

2.1 Inexact Majority Voting

Inexact Majority Voting Using Distance Metric [11] is a commonly used voting technique in the safety-critical systems. If, among N repetitive component outcomes at least $(N+1)/2$ component outcomes are decided i.e., each couple of the component outcomes in the decided majority subgroup meets the situation, $|x_i - x_j| \leq a$ (where, x_i & x_j are the results of the component i & j respectively and 'a' is the predetermined voting threshold), then the chosen last outcome from most subgroup will be the component outcome which is in lowest range with all the other component outcomes of most subgroup. If, no decided majority subgroup exists, then benign error will be produced. For example, say, five repetitive segments are here; whose outcomes are respectively x_1, x_2, x_3, x_4 and x_5 . In between these outcomes, x_3, x_4 and x_5 are in the majority contract i.e. The difference of x_3 and x_4 is 'a1' which is less than 'a', x_3 and x_5 is 'a2' which is less than 'a' and x_4 and x_5 is 'a3' which is less than 'a' and

$a_2 > a_1 > a_3$. The component outcome x_3 is in the range with x_4 and x_5 as, (a_1+a_2) , the component outcome x_4 is in the distance with x_3 and x_5 as, (a_1+a_3) , and the component outcome x_5 is in the range with x_3 and x_4 as, (a_2+a_3) . As, (a_1+a_3) is the lowest range as opposed to ranges (a_1+a_2) & (a_2+a_3) . So, this voting technique selects the component outcome x_4 from the agreed majority subgroup as the ultimate outcome.

2.2 The Score-based dynamic threshold with dynamic bandwidth selection for fuzzy weighted-average voting

This score based fuzzy voting algorithm is useful for multi state safety-critical systems. It improves the both safety and availability. In all case this algorithm gives good results both small errors and large errors. These findings cause for the need of a voting criteria which can merge all the benefits of the interviewed voting methods, but can prevent the disadvantages of all of them. Inspiration of the perform here is to apply that desired voting criteria, which can provide excellent accessibility and safety performances in all kinds of mistake circumstances and enhances the overall reliability and stability of the program. Modules scores are computed to avoid the classical fuzzy rules used for inference. These scores can be used as weighted vectors directly instead of computing several rule outputs. Scores for the module i and j which has d_{ij} as the module agreement distance can be calculated as follows:

$$\text{Score}_i += \mu_{\text{high}}(d_{ij}) + \mu_{\text{med}}(d_{ij}) - \mu_{\text{low}}(d_{ij}) \text{ and}$$

$$\text{Score}_j += \mu_{\text{high}}(d_{ij}) + \mu_{\text{med}}(d_{ij}) - \mu_{\text{low}}(d_{ij})$$

For example, for each of the normalized distances d_{12}, d_{23} and d_{31} corresponding module scores are updated as given below:

For normalized distance d_{12} ,

$$\text{Score}_{1+} = \mu_{\text{high}}(d_{12}) + \mu_{\text{med}}(d_{12}) - \mu_{\text{low}}(d_{12}) \text{ and}$$

$$\text{Score}_{2+} = \mu_{\text{high}}(d_{12}) + \mu_{\text{med}}(d_{12}) - \mu_{\text{low}}(d_{12})$$

For normalized distance d_{23} ,

$$\text{Score}_{2+} = \mu_{\text{high}}(d_{23}) + \mu_{\text{med}}(d_{23}) - \mu_{\text{low}}(d_{23}) \text{ and}$$

$$\text{Score}_{3+} = \mu_{\text{high}}(d_{23}) + \mu_{\text{med}}(d_{23}) - \mu_{\text{low}}(d_{23})$$

For normalized distance d_{31} ,

$$\text{Score}_{3+} = \mu_{\text{high}}(d_{31}) + \mu_{\text{med}}(d_{31}) - \mu_{\text{low}}(d_{31}) \text{ and}$$

$$\text{Score}_{1+} = \mu_{\text{high}}(d_{31}) + \mu_{\text{med}}(d_{31}) - \mu_{\text{low}}(d_{31})$$

Calculate the voter output as follows. If $\sum \text{score}_i > 0$,

$$\text{Output} = \frac{\sum \text{score}_i \cdot X_i}{\sum \text{score}_i}$$

Otherwise Output = $(x_1+x_2+\dots+x_n)/n$.

3. INCORPORATED VOTER DESIGN

A properly performing majority voting with a given consensus-threshold chooses at unique, from the agreed voter information, where a majority exists. The outcome of inexact majority and fuzzy weighted-average voters for all contract voting periods are identical. This implication leads us to introduce a novel voter that is a combination of majority and weighted-average voters. Majority voting used in agreement cases and Score-based fuzzy weighted-average voting used in disagreement cases [13]. The voter is less complex and quicker than the weighted-average voter, since in the majority of the cases it does not perform the relatively time Consuming weighted averaging procedure. Moreover, the use of the majority algorithm in agreeing voting cycles of the novel voter improves its whole safety level compared to the standard weighted-average voter. This effect brings us to present a novel voter that is a combination of majority and fuzzy weighted-average voters.

3.1 Proposed voter algorithm

Step1: $A = \{x_1, x_2, x_3\}$ denote the set of n voter inputs.

Step2: By using a sorting technique, arrange the set 'A' in ascending order to construct the new set $AS = \{y_1, y_2, y_3\}$.

Step3: If, among N repetitive component outcomes at least $(N+1)/2$ component outcomes are decided i.e., each couple of the component outcomes in the decided majority subgroup meets the situation,

$|y_i - y_j| \leq a$ (where, x_i & x_j are the results of the component i & j respectively and 'a' is the predetermined voting threshold), then the majority for the original set A will be satisfied. Select the mid-located module result in the set 'AS' as Voter output = $x_{[(n+1)/2]}$.

Step4: If $d(y_i, y_j) > a$; then Normalize the distances as $nd_{12}, nd_{23}, nd_{31}$.

Step5: Fuzzy bandwidth can be dynamically changed for every voting cycle by setting the values of 'a', 'b', and 'c' used in fig.1 as given below

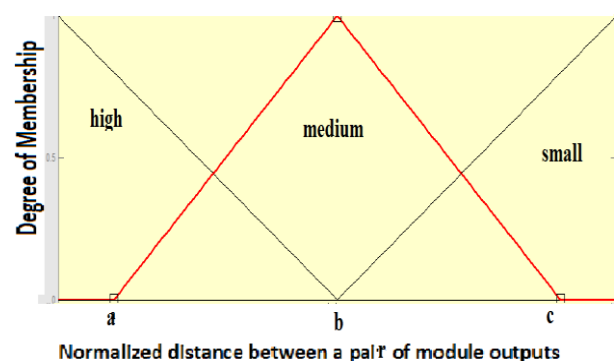


Fig 1: Fuzzy Membership Functions for Dynamic Fuzzy Voter

Calculate the mean of normalized distances formed in previous step 2

$X_i = 1/N \sum nd_i$ Where nd_i is the i th element and N is the total number of elements. $b = \text{mean}(ND)$, $a = b - \text{Standard Deviation}(ND)$ and

$c = b + \text{Standard Deviation}(ND)$.

Step6: Modules scores are computed to avoid the classical fuzzy rules used for inference. These scores can be used as weighted vectors directly instead of computing several rule outputs. Scores for the module i and j which has d_{ij} as the module agreement distance can be calculated as follows:

$$\text{Score}_i += \mu_{\text{high}}(d_{ij}) + \mu_{\text{med}}(d_{ij}) - \mu_{\text{low}}(d_{ij}) \text{ and}$$

$$\text{Score}_j += \mu_{\text{high}}(d_{ij}) + \mu_{\text{med}}(d_{ij}) - \mu_{\text{low}}(d_{ij})$$

For example, In TMR system, for each of the normalized distances d_{12} , d_{23} and d_{31} , corresponding module scores are updated as given below: For normalized distance d_{12} ,

$$\text{Score}_1 += \mu_{\text{high}}(d_{12}) + \mu_{\text{med}}(d_{12}) - \mu_{\text{low}}(d_{12}) \text{ and}$$

$$\text{Score}_2 += \mu_{\text{high}}(d_{12}) + \mu_{\text{med}}(d_{12}) - \mu_{\text{low}}(d_{12})$$

For normalized distance d_{23} ,

$$\text{Score}_2 += \mu_{\text{high}}(d_{23}) + \mu_{\text{med}}(d_{23}) - \mu_{\text{low}}(d_{23}) \text{ and}$$

$$\text{Score}_3 += \mu_{\text{high}}(d_{23}) + \mu_{\text{med}}(d_{23}) - \mu_{\text{low}}(d_{23})$$

For normalized distance d_{31} ,

$$\text{Score}_3 += \mu_{\text{high}}(d_{31}) + \mu_{\text{med}}(d_{31}) - \mu_{\text{low}}(d_{31}) \text{ and}$$

$$\text{Score}_1 += \mu_{\text{high}}(d_{31}) + \mu_{\text{med}}(d_{31}) - \mu_{\text{low}}(d_{31})$$

Step7: Calculate the voter output as a weighted-average of scores and module output values. If $\sum \text{score}_i > 0$ then

$$\text{Output} = \frac{\sum \text{score}_i * x_i}{\sum \text{score}_i}$$

$$\text{Otherwise Output} = (x_1 + x_2 + \dots + x_n) / n$$

3.2 Design of Incorporated voting algorithm

This voting algorithm uses the concept of the dynamic threshold. It gives all advantages of Majority voting and fuzzy weighted- average voting strategies in the case of safety and availability performances. This voting algorithm is useful for multi state safety-critical system. This voting technique is useful for both permanent and intermittent errors. This algorithm is used to provide an error masking capability in safety-critical systems and to hide the occurrence of errors from the system output. This voter can be used in any safety-critical system without having much information about the system, data and range of data. Proposed voting algorithm maximizes the advantages, but minimizes the disadvantages.

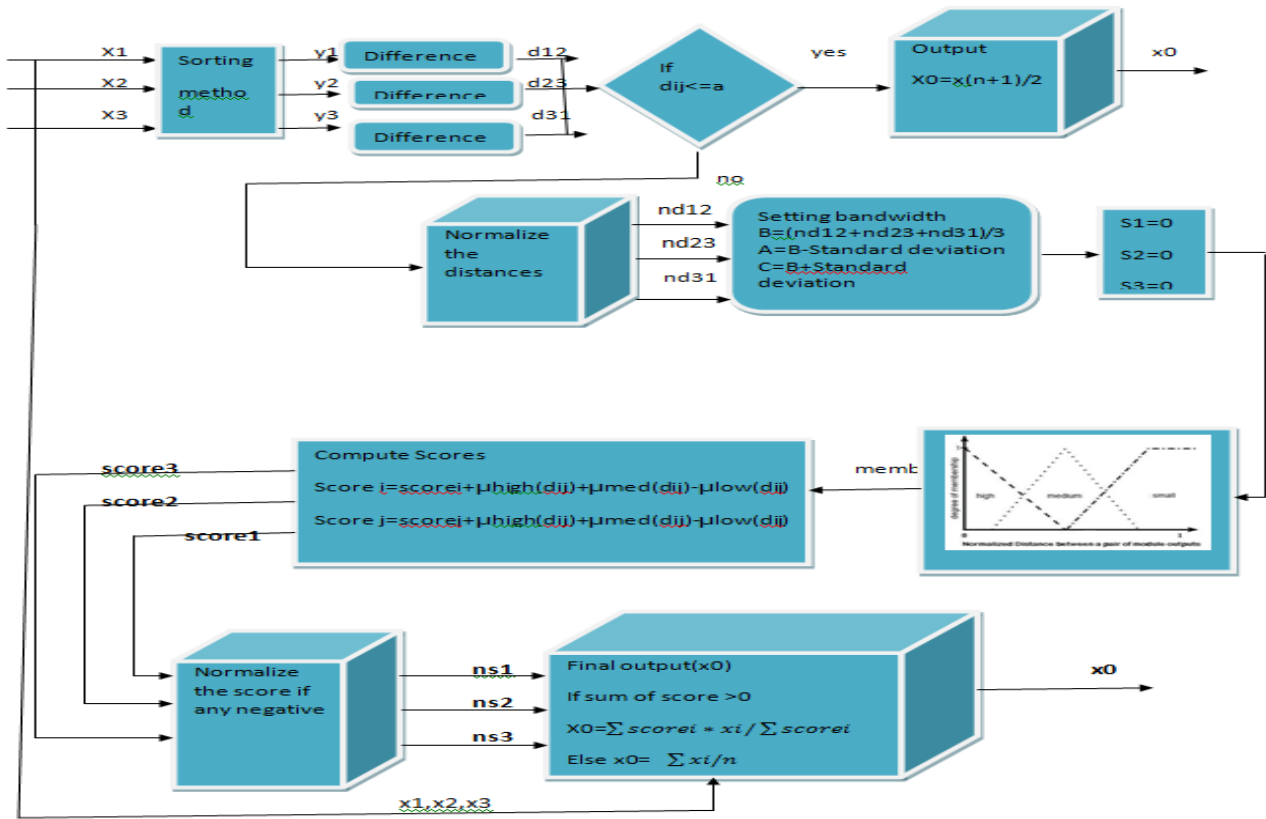


Fig 2 : Block diagram of proposed voting algorithm for a TMR system

4. EXPERIMENTAL METHOD

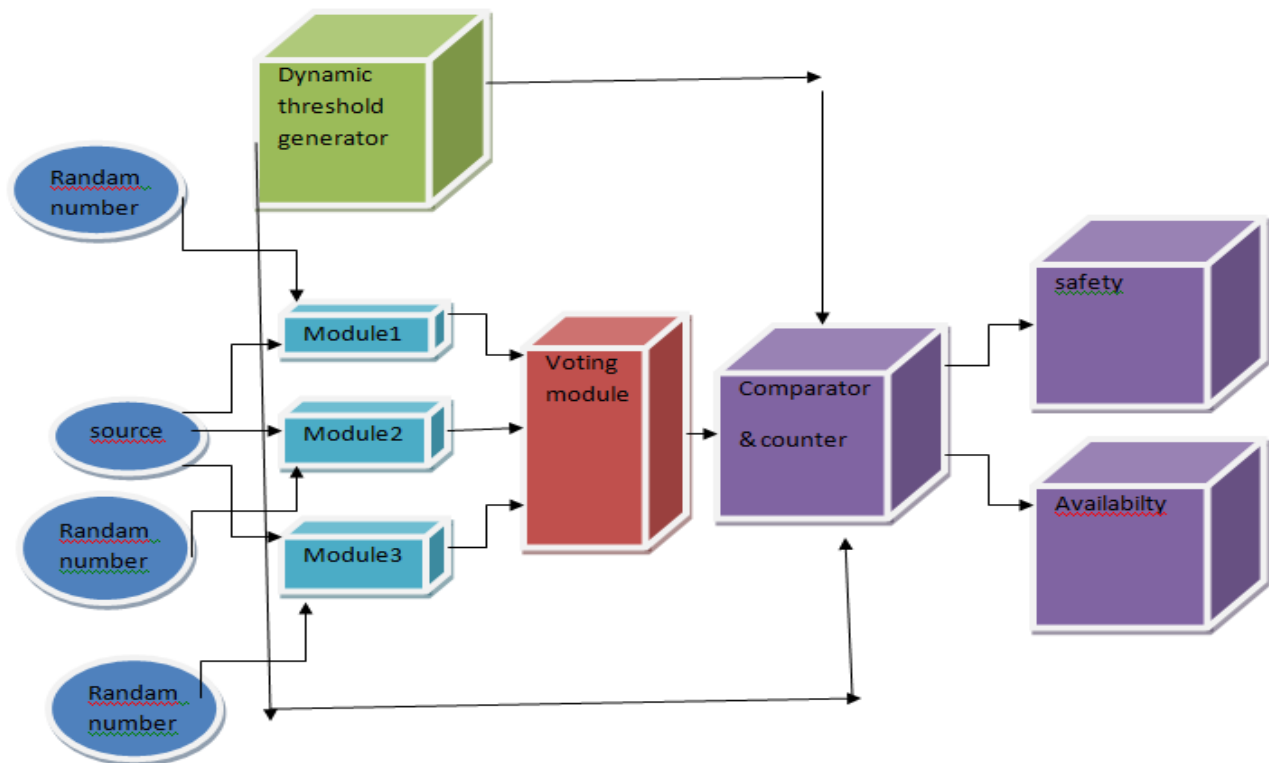


Fig 3: Experimental test harness

Test harness for analysis with voting methods is shown in Fig.3. Cyclic information like sin wave is produced using the formula given. Input information = $100 + 100 * \sin(t)$ and Sample amount 't' is taken as 0.1. Generated input information is given to each of the segments the random error of uniform distribution is treated in each of the needed components in the needed range [-e, +e]. Initially generated input information before treating the error is regarded as the notional appropriate outcome. Accuracy threshold is taken as 0.2 and 0.1 and safety efficiency is analyzed. The generated outcome by the voter is compared with the notional correct outcome and if the difference is less than the truth threshold value, it is regarded as the appropriate outcome otherwise wrong outcome. Each set of Research is conducted for 10000 runs and the number of correct results (nc) and number of incorrect results (nic) are mentioned. Then the efficiency of the voter is analyzed by using the parameters safety and availability as given below: $Safety = 1 - (nic/n)$ and $Availability = nc/n$

Where nc = Number of correct results given by a voter

nic = Number of Incorrect results given by a voter

n = Total number of runs or voting cycles

Within the test harness the following parameters can be adjusted. The value of accuracy threshold, Number of voter inputs, Input data trajectory and sample rate, the amplitude of injected errors, Number of injected errors, Error persistence time.

5. ACKNOWLEDGMENTS

Scientific assessment of the Proposed voters is done by operating each voter for 10000 voting period on a TMR program since TMR is commonly used in safety-critical applications[12] is given below.

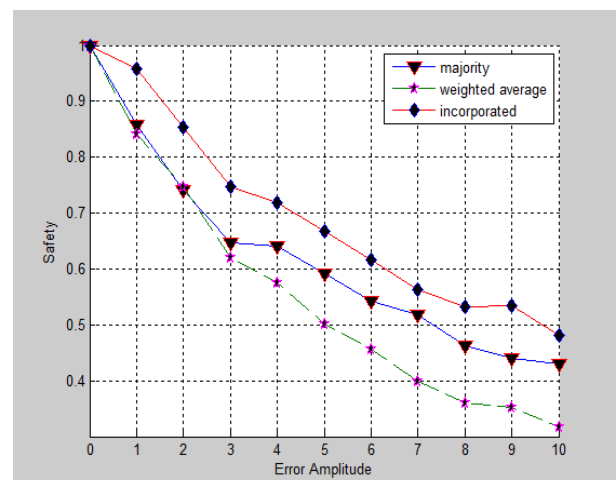


Fig 4: Safety of the voter vs. Error amplitude

Fig.4 shows the safety plot of Majority, fuzzy weighted-average and incorporated voters vs. error amplitude for 10000 runs of voters for a range of error injection tests. Two error 100 % free segments and the other component perturbed with large errors. In this situation two segments are mistake 100 % free and the other component is perturbed with the huge mistakes. For each mistake plentitude e=0 to e=10 and e value incremented by 1, voters are run for 10000 voting period and how many periods each voter is providing the appropriate outcome (safety performance) is documented. Module1 and module2 are mistake totally exempt from 0 to 3999 periods, whereas module3 is perturbed with a mistake in the variety [-e +e]. Module2 and module3 are mistake totally exempt from 4000 to 6999 periods, whereas module1 is perturbed with a

mistake in the variety [-e +e]. Module1 and module3 are mistake totally exempt from 7000 to 10000 periods, whereas module2 is perturbed with a mistake in the variety [-e +e]. Different approval or precision limits like 0.1, 0.2, 0.5 and 1.0 are taken while determining the safety efficiency. The plot indicates that in all cases the proposed voter shows more safety than remaining voting algorithms.

6. CONCLUSION

Fault masking programs usually need an advanced stage of safety and accessibility, which frequently conflict with each other. The Majority voter with high level of safety has usually a low stage of accessibility and the Fuzzy weighted-average gives advanced stage of availability in the cost of low safety and higher safety than the standard weighted-average voting algorithm. In this combine these properties, take the advantages of both voters. The experimental results revealed that the novel voter has higher safety performance than the Majority and fuzzy weighted-average voting algorithms. If, one component is mistake free, this voting criterion can provide almost 100 % (100%) safety and also a good range of accessibility. Time complexities are calculated for future work.

7. REFERENCES

- [1] Soureh Latif-Shabgahi” An Integrated Voting Algorithm for Fault Tolerant Systems” 2011 International Conference on Software and Computer Applications IPCSIT vol.9 (2011) © (2011) IACSIT Press, Singapore.
- [2] EUROCAE/RTCA, “ED-12B/DO-178B: Software Considerations in Airborne Systems and Equipment Certification”, EUROCAE/RTCA, 1994
- [3] J.-C. Laprie, (1985). Dependable computing and fault-tolerance: concepts and terminology, in Digest of Papers FTCS’15: IEEE 15th Annu. Int. Symp. Fault-Tolerant Computing Systems, Ann. Arbor, MI, pp. 2–11.
- [4] B. W. Johnson (1989), Design and Analysis of Fault-Tolerant Digital Systems. New York: Addison-Wesley.
- [5] J. L. Lions, “ARIANE 5: Flight 501 failure,” Ariane 5 Inquiry Board Report, Paris, Tech. Rep., 1996.
- [6] “Aviation safety investigation report - in-flight upset; 240km nwpert,wa;boeingco777-200,9m-mrg,”<http://www.atsb.gov.au/publications/investigationreports/2005/AAIR/air200503722.aspx>, 2005.
- [7] D. M. Blough & G. F. Sullivan (1990). A comparison of voting strategies for fault-tolerant distributed systems, in Proc. IEEE 9th Symp. Reliable Distributed Systems, Huntsville, Alabama, pp. 136–145.
- [8] G. Latif-Shabgahi & S. Bennett (1999), Adaptive majority voter: a novel voting algorithm for real-time fault-tolerant control systems, in 25th Euromicro Conf., vol. 2, pp. 113-120.
- [9] G. Latif-Shabgahi, Julian M. Bass & Stuart Bennett (2004), A taxonomy for software voting algorithms used in safety-critical systems, IEEE Trans. Reliability, vol. 53, no. 3, pp 319-328,
- [10] P. R. Lorzak, A. K. Caglayan and D. E. Eckhardt. “A Theoretical Investigation of Generalised Voters”, Proc. Of IEEE 19th Ann. Int. Symp. on Fault-Tolerant Computing Systems, 1989, Chicago, USA, June, pp. 444-451.
- [11] B. Baykant Alagöz, “Adaptive fault masking with incoherence scoring,” *Oncu Bilim Algorithm And System Lab.*, vol. 08, no. 01, Jul. 2008, pp. 1-14.
- [12] Latif-Shabgahi, G., & Hirst, A. J. (2005). “A fuzzy voting scheme for hardware and software fault tolerant systems. *Fuzzy Sets and Systems*”, 150(3), 579–598. doi:10.1016/j.fss.2004.02.014
- [13] B. Uma Maheswara Rao and Seetha Ramaiah Panchumarthy, “An Improvement in History Based Weighted-Voting Algorithm for Safety Critical Systems” *International Journal of Computer Science Engineering Research and Development (IJCSERD)*, ISSN 2248 – 9363(Print) ISSN 2248 – 9371(Online), Volume 3, Number 1 Jan- March (2013), pp: 41-59 © PRJ Publication, <http://www.prjpublication.com/IJCSERD.asp>.