

Comfortable Voter Design in Highly Effective Bandwidth Usage and Dynamic Threshold Selection for Safety Critical System

Umamaheswararao Batta*, Seetharamaiah Panchumarthy**,
Phanikumar Singamsetty***

Abstract

Safety and availability of safety critical systems are increased by using masking method. The main aim of this research paper is designing dynamic bandwidth selection for fuzzy voter for Railroad Crossing Control System (RCCS) and masking the faults. The failure of RCCS leads to damage of property and at-times loss of human life. Safety and availability of the safety critical system is calculated. The major limitation of existing fuzzy voters is uncertainties associated with the inputs and three values (a, b and c) are required as threshold. Only fixed factors are used for a particular set of information and acquainted information varies. In this paper, a dynamic fuzzy voter with three automatic values (a, b and c) are proposed in each voting cycle and existing standard weighted average voting algorithm, flexible weighted average voting algorithm, weighted average voting with soft threshold algorithm and Fuzzy weighted average voting are surveyed and their merits and demerits are discussed. Safety overall performance is empirically analyzed by using the fixed fluffy heavy average voter, powerful fluffy voters, fixed fluffy voter with ratings and fluffy powerful fluffy voter with ratings on a TMR system for 10000 voting periods and RMSE is measured for fixed and powerful fluffy voter. The approach uses some form of voting to avoid the classical fuzzy rules used for inference and compute scores that can be used as weighted factors directly instead of computing several rule outputs. We also briefly compare Static bandwidth fuzzy Weighted Average Voter, Dynamic bandwidth fuzzy Weighted Average Voter, Static fuzzy voter without weighted Average Voter and Dynamic fuzzy voter without Weighted Average Voter.

Keywords: Safety, Fuzzy Voter, Safety Critical Systems

1. Introduction

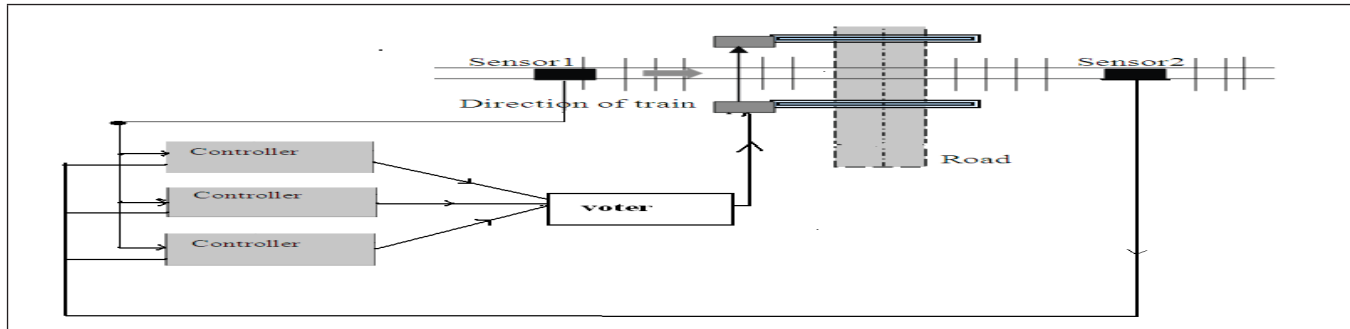
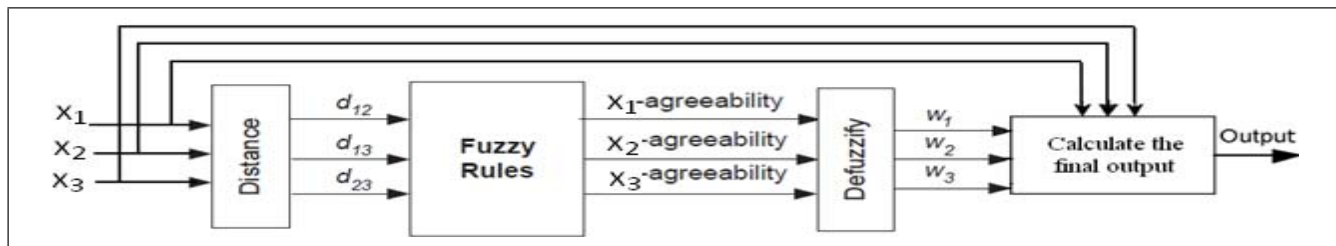
A safety critical system is a system which is dependent on correct operation. If operation fail this gives us hazards. Safety must always be calculated based on the complete system, such as application, components, other electronic and electrical components, technical components, and providers or users, not just the application element. Many safety-critical systems rely on software to achieve their purposes. There are different domains in which safety critical control systems are used in Automotives, Medicine, Military and Space applications, Industrial Process Control, Robotics and Consumer electronic appliances, transportation (such as railway, avionics and increasingly, automotive systems)[15,16]. There is a need to increase the reliability, availability and safety in all these applications. Risks happen if any mistake happens to these programs. If a single component is used and when it becomes defective due to some disturbance the program may don't succeed and threat may happen[20][22].

There are different types of voting algorithms used for mistake covering up[1][2]. They are Majority voting, plurality voting, average voting methods. In majority voting [3], if at least $n+1/2$ outcomes are decided then that decided outcomes are chosen as outcome out of n outcomes. Plurality[4] one form of majority voting and if

* Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, India.
E-mail: umamaheshbatta@gmail.com

** Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, India.

*** Department of Computer Sciences, GITAM University, Hyderabad, Andhra Pradesh, India.

Figure 1. RCCS with Dynamic Fuzzy Voter**Figure 2. Architecture of the Fuzzy Voter**

any principles seems to be large amount of time then that principle is taken as outcome. Median voter is not used for safety crucial techniques. Weighted regular voting has different types. They are Lorzak's heavy regular voting criteria, History based heavy regular voting criteria[9], Versatile heavy regular voting criteria, Weighted regular voting with soft limit Value voting criteria, Fuzzy heavy regular voting, maximum likelihood Voter [15], predictor voters [16], stepwise negotiation voter [17] and word voters [18]. Fuzzy heavy regular voting powerful data transfer usage.

This Research Paper is organized as follows:

Section 2 describes related work about RCCS and some fuzzy voting design.

In Section 3, Comfortable Voter Design and flowchart.

In Section 4, Experimental Framework.

In Section 5, Conclusion.

2. Related Works

2.1 RCCS with Dynamic Fuzzy Voter

Here controllers are replicated three times and voter is software or hardware that has three inputs and one output. If two or three of the Inputs are the same, the output is

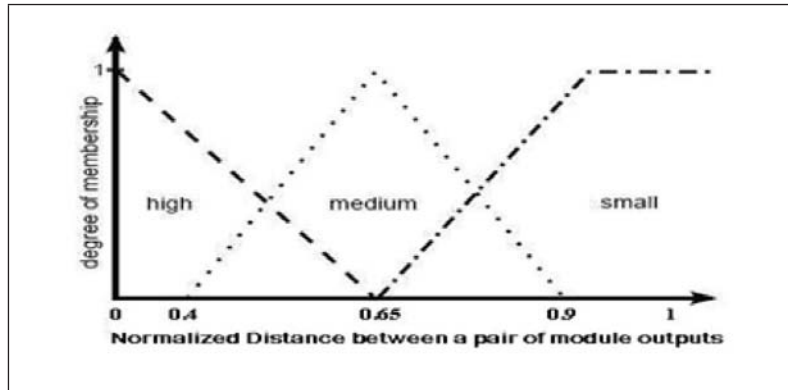
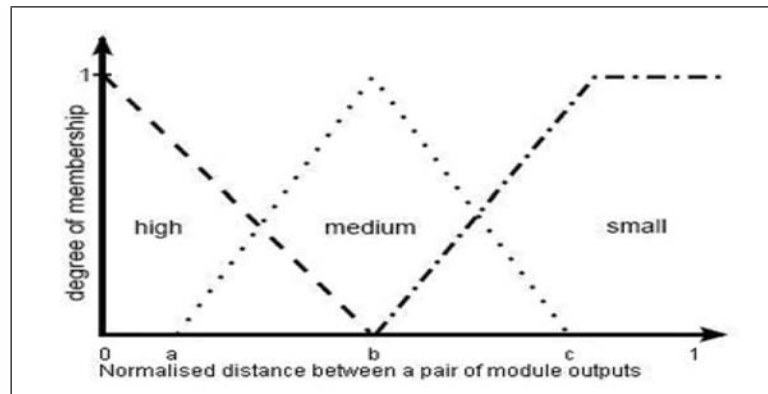
equal to that input. If all three inputs are different, the output is undefined. A sensor sends a signal to controllers and controllers give the output to voter. Here voter is used as dynamic inexact fuzzy voter for Railroad Crossing control system.

2.2. Fuzzy Voter

Fuzzy voter is used to cover up the mistakes in safety critical crucial systems. It calculate the differences between modules those are d_{12} , d_{13} , and d_{23} . Fuzzy rules are applied to find the agreeability of each module. This agreeability of each module is defuzzify and get the weight values of each module. Fuzzyfication is done with Mamdani-Larsen interfacing

Trial and error method is used for bandwidth selection. For example a, b, c values as 0.5, 0.75, 1. Without knowing system information, we cannot use this method. It is a significant disadvantage with the fixed technique of determining the fluffy information transfer usage and there are more possibilities of misclassification of the information.

Proposed dynamic fuzzy voter is dynamic selection of bandwidth for the fuzzy voter is configurable. It is easily implemented with some system information. This voter can be used anywhere in the N-Modular Redundant

Figure 3. Fuzzy Membership Functions for Static Fuzzy Voter**Figure 4. Fuzzy Membership Functions for Dynamic Fuzzy Voter**

Systems or Sensor Fusion applications without calibrating it for any threshold values and without knowing anything about those systems.

3. Proposed Dynamic Threshold with Dynamic Bandwidth Selection

Adaptive majority voting always generating the solutions for permanent errors. But it fails for intermittent errors. In The History Based Weighted Average Voting threshold selection is main problem. The Soft-threshold Weighted Average Voting provides a soft behavior, but provides poor safety and availability. The fuzzy weighted average voting provides higher safety and gives us lower availability for large errors. For small errors it provides lower safety than other voting algorithms. The dynamic accuracy threshold with dynamic bandwidth selection gives us better safety and availability in case of small and large errors. The Proposed methodology is as follows.

Find the distances between module output I, j. Normalize the module agreement distances(ND).Substitute upper

triangular of the distance matrix in to ND. The fixed threshold value is 0.5. Accuracy threshold value is taken same is as the voting threshold value. The Dynamic threshold is set as: $V_t = X / 100$. Setting the dynamic fuzzy bandwidth as follows $B = \text{mean}(ND)$, $A = B - \text{standard deviation}(ND)$, $C = B + \text{standard deviation}(ND)$. The fuzzy membership function can be computed for the agreement set. Therefore, it is possible to compute the membership values $\mu_{\text{high}}(d_{ij})$, $\mu_{\text{med}}(d_{ij})$, $\mu_{\text{low}}(d_{ij})$ very efficiently. In existing system finding the fuzzy rule set used for inference is overhead. Therefore estimated component ratings used instead of using fuzzy concept set for inference phase.

Scores for module i and j which have d_{ij} as the module agreement distance can be calculated as follows:

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

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

If any module score is negative, normalize all the module scores to make all of them positive.

Determine the voter outcome as a weighted average of ratings and component outcome values.

If $\sum_{i=1}^n score_i > 0$

$$Output = \sum_{i=1}^n score_i * xi \div \sum_{i=1}^n score_i$$

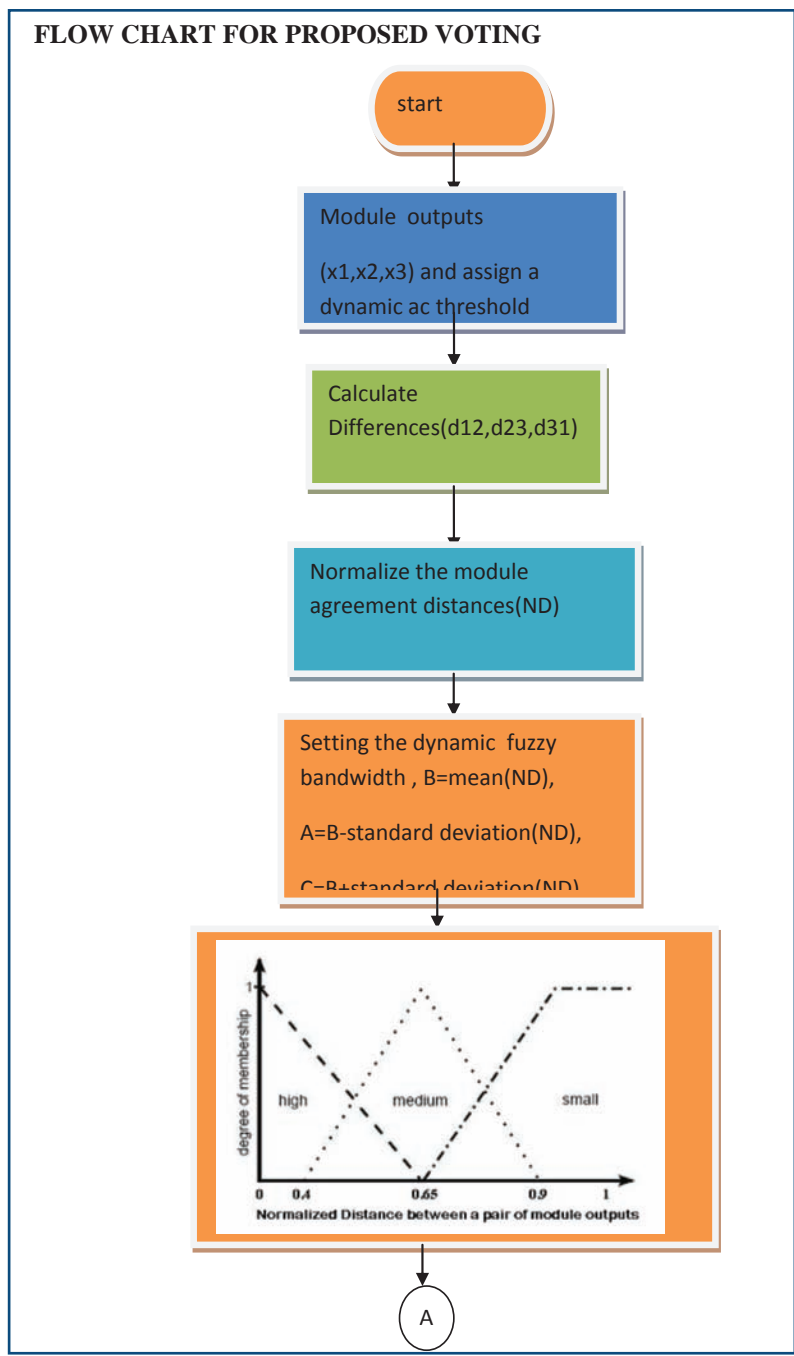
Otherwise

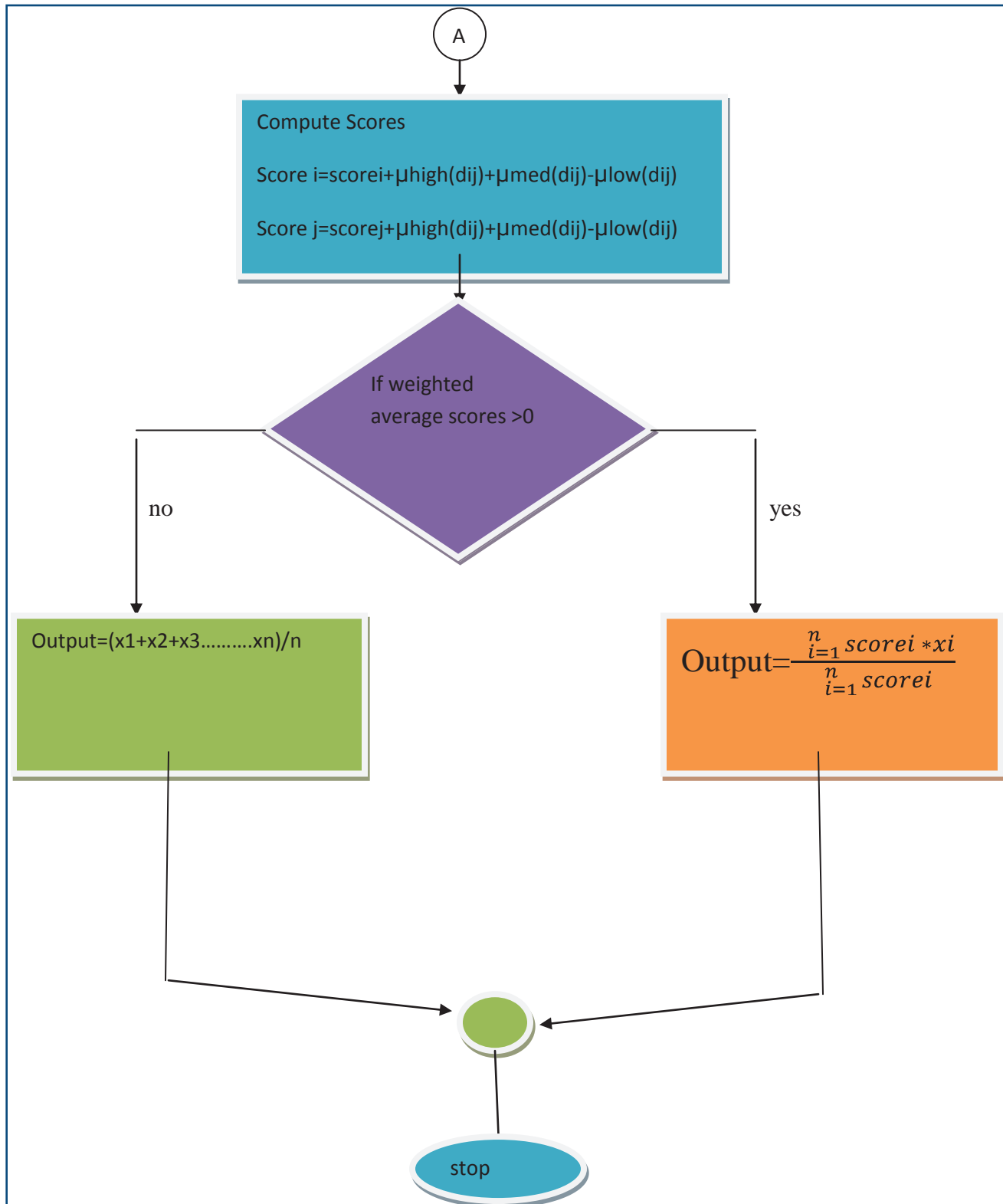
$$Output = (x_1 + x_2 + \dots + x_n) / n$$

In this mean of the stabilized ranges vector for a particular voting pattern is measured and allocated to parameter 'b'.

Conventional difference of the stabilized ranges deducted from mean is allocated to 'a' and the common difference included to mean is allocated to parameter 'c'.

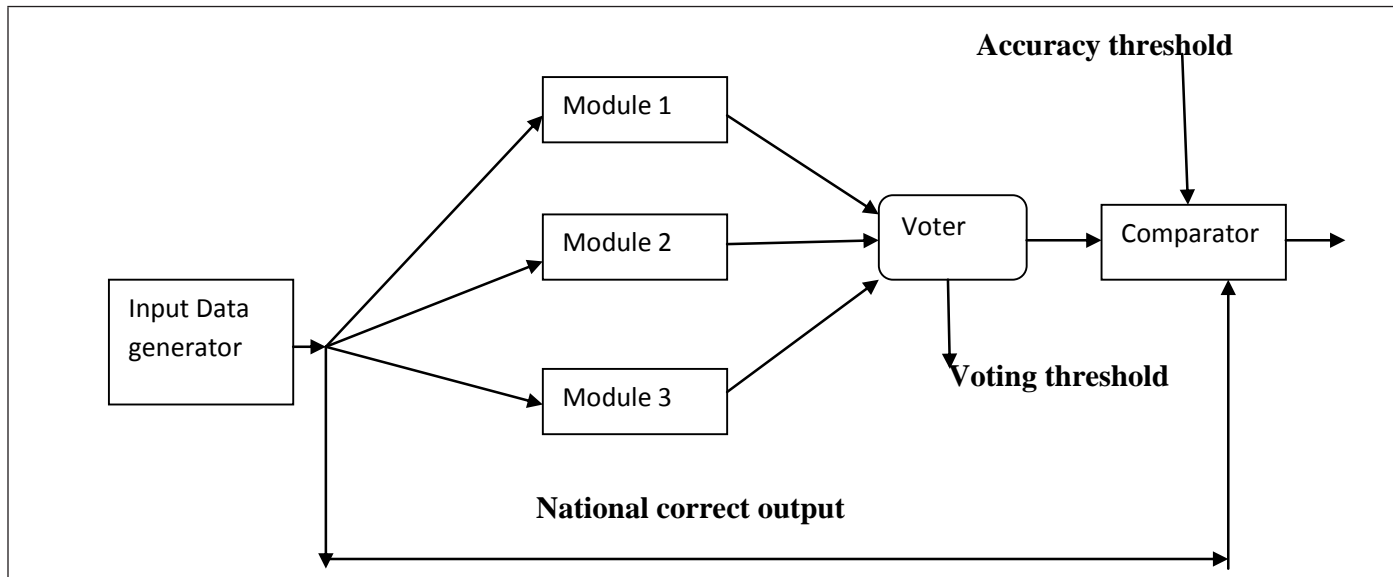
The Dynamic Inexact fuzzy voter in automatic bandwidth selection for Safety Critical System algorithm combines all concepts of the studied voting algorithms to maximize the advantages and minimize the disadvantages of the voting algorithms. Here at first all pair wise distances are calculated. Next normalize agreed distances for eliminating negative values and setting the bandwidth distances. First values takes as mean of normalized vector. Second values





as difference of mean and standard deviation and third value as sum of mean and standard deviation. Based on distances calculate the membership function as shown in flowchart. Determine scores depending on the membership

functions. Weighted average scores as the output for this algorithm. This algorithm gives 100% safety and higher range of availability than other algorithms if at least one module is fault free.

Figure 5. Experimental Setup to Evaluate the Performance of A Voter

4. Experimental Framework

4.1. Test Harness Structure Accuracy Threshold

Here cyclic data like sin wave is generated using the equation given.

$$\text{Input data} = 50 + 100 * \sin(t)$$

Sample rate t is taken as 0.1.

Produced feedback information is given to each of the segments and the unique mistake of uniform distribution is treated into each of the required module in the needed variety $[-e, +e]$. Initially generated feedback information before treating the error is regarded as the notional appropriate outcome. Accuracy or Approval Limit is taken as 0.2 and 0.1 and protection performance is analyzed. The generated outcome by the voter is compared with the notional appropriate outcome and if the difference is less than the precision threshold value, it is regarded as the appropriate result otherwise wrong result. Count the number of correct results and number of incorrect results for each cycle. Safety parameter is used to evaluate the performance of the voter is given below:

$$\text{Safety} = 1 - (\text{number of incorrect results} / \text{Total number of runs or voting cycles})$$

$$\text{Availability} = \text{number of correct results} / \text{Total number of runs or voting cycles}$$

4.2. Overall Performance of Powerful data Bandwidth Usage Fuzzy Voter with Dynamic Threshold.

It is seen from Figure.6, In between the error range 0 to 10 proposed gives higher safety compared to the all other voting algorithms.

4.3. Overall Performance of Powerful data Bandwidth Usage Fuzzy Voter with Static Threshold

It is seen from Figure.7, the safety performance of the suggested voting is better compared to the existing fuzzy weighted average voting method. In between the mistake range 0 to 1 all four voting methods protection performance is same and 1 to 10 suggested gives higher protection as opposed to all other voting methods.

Figure 8 shows RMSE (Root Mean Square error) which is used to measure differences between national correct a results and the values actually observed. It is used to aggregate the magnitudes of errors in predictions for various times into a single measure of predictive power. RMSE is a good measure of accuracy, but it is only to compare forecasting of different models for a particular variable. RMSE is most useful when large errors are particularly undesirable.

Figure 6. Safety Performance of Dynamic Threshold with Dynamic Bandwidth

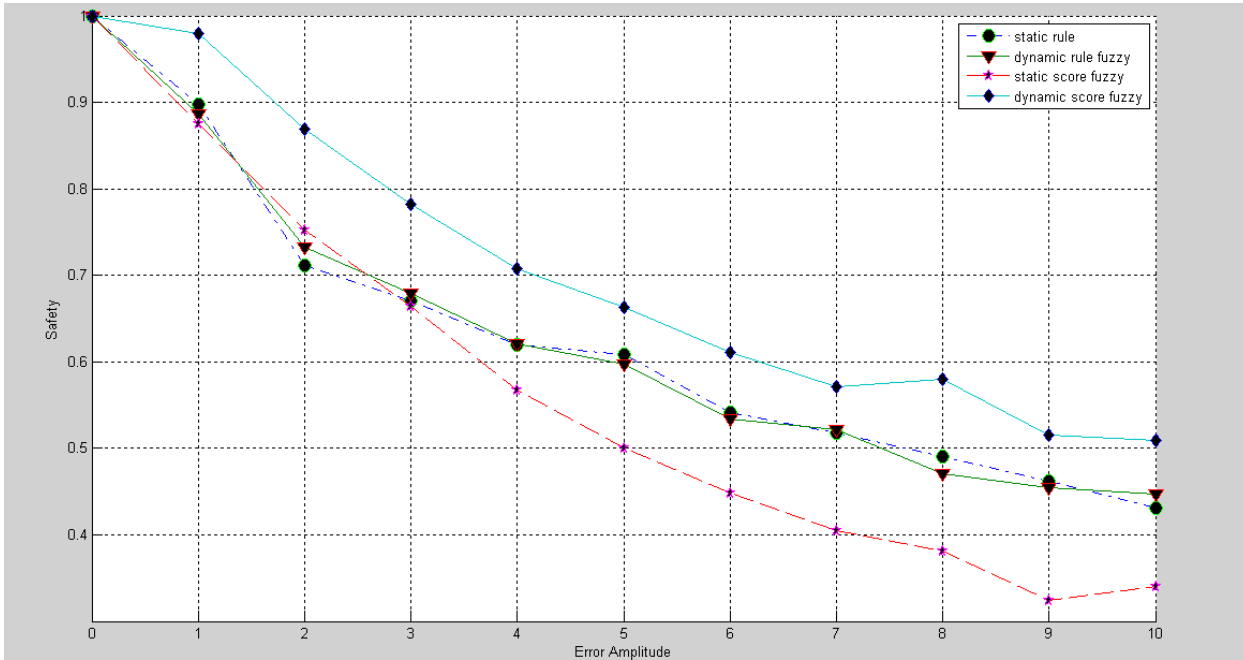
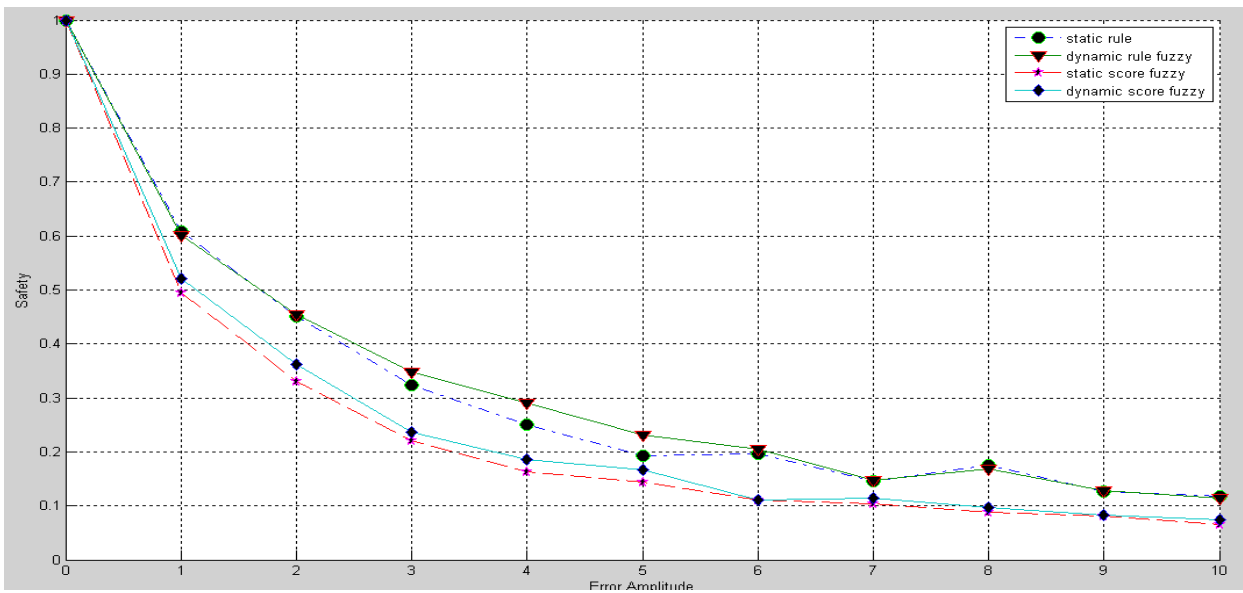


Figure 7. The Safety Performance of Static and Dynamic Fuzzy, Rule Fuzzy, Dynamic Rule Fuzzy Voters for 10000 Cycles for Small Errors with one Error Free Module (Accuracy Threshold = 0.5)



As given in the table 4.1, if two modules are error free then dynamic fuzzy voter gives 100% correct outputs. Cycle no 62 module1 and module2 are error free and module 3 giving error. Cycle numbers 44, 45, 46 module1 and module 2 are getting error and module 3 is error free. Cycle numbers 621-626 module 1, module 2, module 3 are getting error and it give nearest results. Powerful fuzzy voter determine the error segments and prevented

these modules from leading to the voter outcome.

As given in the table 4.2, if two modules are error free then static fuzzy voter gives 100% correct outputs. Cycle no. 1 to 9, module1 and module 2 are error free and module 3 giving error. Static fuzzy voter determine the error modules and avoided these modules from contributing to the voter output. But this voter is not useful if two modules getting errors and it cannot give us correct results.

Figure 8: RMSE for Static and Dynamic Voter

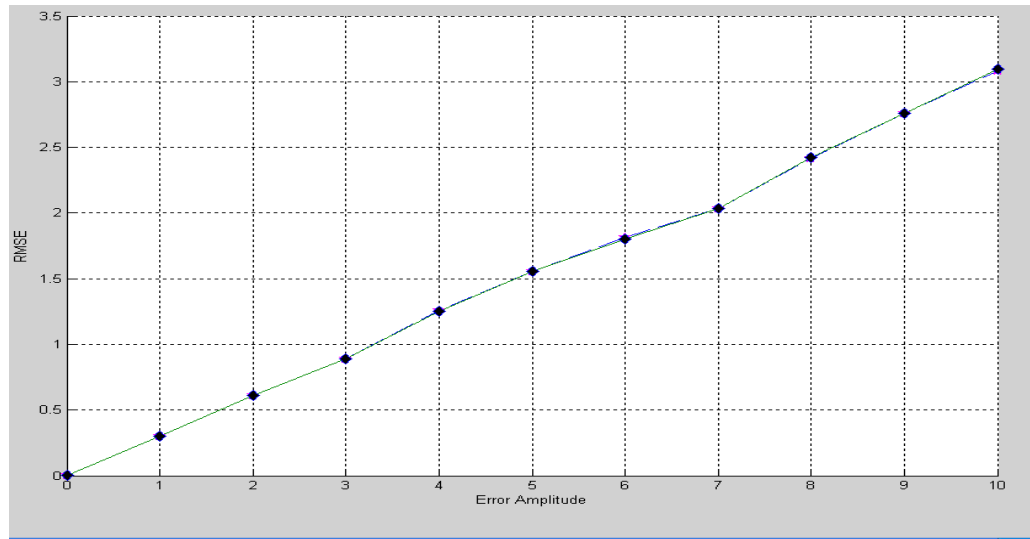


Table 4.1. Dynamic Selection of Fuzzy Parameters

Cycle no	Input	Module1	Module2	Module3	a	b	c	output
62	81.783750	81.783750	81.78375	81.87	0.40	0.65	0.90	81.78375
44	8.383406	5.46875	10.73513	8.38340	0	0.39	0.83	8.86451
45	4.8397	1.8732	6.1150	4.8397	0.11	0.52	0.93	4.9704
46	2.2469	4.5161	0.7580	2.2469	0.03	0.44	0.86	2.08097
621	26.0819	30.088	17.85744	20	0.099	0.507	0.915	26.221
622	33.1750	27.4695	34.9041	21	0.14	0.56	0.98	33.5878
623	40.9358	47.1876	47.4046	22	0.191	0.655	1	47.2904
624	49.2868	53.5256	55.8319	23	0.076	0.48	0.89	53.6057
625	58.1445	56.0428	50.827	24	0.121	0.53	0.94	56.328
626	67.420	64.916	75.055	25	0.141	0.55	0.97	66.8795

Table 4.2. Static Fuzzy Voter for two Error free Modules and Other Module Having Error

Cycle no	Input	Module1	Module2	Module3	a	b	c	output
1	100.00	100.00	100.00	100	0.4	0.65	0.9	100.00
2	109.9833	109.9833	109.9833	109	0.4	0.65	0.9	109.9833
3	119.8669	119.8669	119.8669	119	0.4	0.65	0.9	119.8669
4	129.5520	129.5520	129.5520	129	0.4	0.65	0.9	129.5520
5	138.9418	138.9418	138.9418	138	0.4	0.65	0.9	138.9418
6	147.9425	147.9425	147.9425	147	0.4	0.65	0.9	147.9425
7	156.4642	156.4642	156.4642	156	0.4	0.65	0.9	156.4642
8	164.4217	164.4217	164.4217	164	0.4	0.65	0.9	164.4217
9	171.7356	171.7356	171.7356	171	0.4	0.65	0.9	171.7356

5. Conclusions

This proposed voting gives correct outputs compared to remaining algorithms at the presence of intermittent errors. This work gives us better safety and accessibility as opposed to current heavy regular voting methods. If, any one of the module is free, this proposed voting can provides 100% safety and also higher range of availability. In the existing static fuzzy voter, choosing maximum fuzzy factors are set throughout the voting cycles, which is not a better idea. This method is useful only if information variations are known based upon which maximum fluffy factors are chosen to decide the fluffy information transfer usage. This proposed voter can be used in any type of safety critical systems. This work can be extended to design 5 input fuzzy voter and the time domain complexities of the voting algorithms are left as the future work. Time complexities are calculated for future work. We may test the performances of the proposed algorithm on any hardware system.

References

- [1] Laprie, J. C. (1985). *Dependable computing and fault-tolerance: concepts and terminology*. Digest of Papers FTCS'15: IEEE 15th Annual International Symposium on Fault-Tolerant Computing Systems (pp. 2-11). Michigan, Ann Arbor.
- [2] Johnson, B. W. (1989). *Design and Analysis of Fault-Tolerant Digital Systems*. New York: Addison-Wesley.
- [3] Chen, L. & Avizienis, A. (1989). *N-Version Programming: A fault-tolerance approach to reliability of software operation*. Digest of Papers FTCS'8: IEEE 8th Annual International Symposium on Fault-Tolerant Computing Systems. (pp. 3-9). New York: Addison-Wesley
- [4] Blough, D. M. & Sullivan, G. F. (1990). *A comparison of voting strategies for fault-tolerant distributed systems*. In Proceedings of IEEE 9th Symposium Reliable Distributed Systems (pp. 136-145) Huntsville, Alabama.
- [5] Latif-Shabgahi, G. (2004). *A novel algorithm for weighted average voting used in fault-tolerant computing systems*. Microprocessors and Microsystems, 28, 357-361.
- [6] Latif-Shabgahi, G., Bass, J. M. & Bennett, S. (2001). *History-based weighted average voter: A novel software voting algorithm for fault-tolerant computer systems*. *Euro-micro Conference on Parallel, Distributed, and Network-Based Processing* (pp. 402-409).
- [7] Latif-Shabgahi, G., Bass, J. M. & Bennett, S. (2004). *A taxonomy for software voting algorithms used in safety critical systems*. IEEE Transactions on Reliability, September, 53(3), 319-328.
- [8] Lorzak, P. R., Caglayan, A. K. & Eckhardt, D. E. (1989). *A theoretical investigation of generalized voters for redundant systems presented at FTCS-19*. Digest of Papers, Nineteenth International Symposium on Fault-Tolerant Computing. Chicago, USA.
- [9] Latif-Shabgahi, G. & Bennett, S. (1999). *Adaptive majority voter: A novel voting algorithm for real-time fault-tolerant control systems*. In 25th Euro-micro Conference, 2, 113- 120.
- [10] Phani Kumar, S., Ramaiah, P. S. & Khanaa, V. (2011). *Architectural patterns to design software safety based safety-critical systems*. Proceedings of ICCCS' 11th International Conference on Communication, Computing & Security (pp. 620-623). ACM New York.
- [11] Das, M. & Battacharya, S. (2010). *A modified history based weighted average voting with soft-dynamic threshold*. In ACE '10 Proceedings of the 2010 International Conference on Advances in Computer Engineering.
- [12] Zarafshan, F., Latif-Shabgahi, G. & Karimi, R. (2010). *A novel weighted voting algorithm based on neural networks for fault-tolerance systems*. IEEE International Conference on Computer Science and Information Technology (pp.135-139).
- [13] Lee, S., Jung, J. I. & Lee, I. (2007). *Voting structures for cascaded triple modular redundant modules*. *IEICE Electronics Express*, 4(21), 657-664.
- [14] Baykant, A. B. (2008). *Hierarchical triple-modular redundancy (H-TMR) network for digital systems*. *OncuBilim Algorithm and Systems Labs.*, 8(5), 1-15. Retrieved from <http://arxiv.org/ftp/arxiv/papers/0902/0902.0241.pdf>
- [15] Leung, Y. W. (1995). *Maximum likelihood voting for fault-tolerant software with finite output space*. IEEE Transactions on Reliability, 44(3), 419-427.
- [16] Bass, J. M., Croll, P. R., Fleming, P. J. & Woolliscroft, L. J. C. (1994). *Three domain voting in real-time distributed control systems*. Proceedings of 2nd IEEE

- Euro-micro Workshop on Parallel & Distributed Processing (pp. 317-324).
- [17] Kanekawa, K., Maejima, H., Kato, H. & Ihara, H. (1989). *Dependable on-board computer systems with a new method: Stepwise negotiated voting*. Proceedings of FTCS'19: IEEE 19th Annual International Symposium on Fault-Tolerant Computing Systems (pp. 13-19). Chicago, USA.
- [18] Mitra, S. & McCluskey, E. J. (2000). *Design of redundant systems protected against common-mode failures*. Technical Report, CRC-TR-00-2, Stanford University. Retrieved from <http://crc.stanford.edu>.
- [19] Lorzak, P. R., Caglayan, A. K. & Eckhardt, D. E. (1989). *A theoretical investigation of generalized voters*. Proceedings of IEEE 19th Ann. International Symposium on Fault-Tolerant Computing Systems, (pp. 444-451). Chicago, USA.
- [20] Yu, H. & Xu, A. (2010). *Design of a fault-tolerant voter for safety related analog inputs*. Proceedings of 3rd International Conference on Advanced Computer Theory and Engineering. Chengdu, China.
- [21] Kwiat, K., Taylor, A., Zwicker, W., Hill, D., Wetzonis, S. & Ren, S. (2010). *Analysis of Binary Voting Algorithms for Use in Fault Tolerant and Secure Computing*. Retrieved from <http://lcs.syr.edu/faculty/tang/Teaching/CSE791-Spring11/Papers/Voting.pdf>
- [22] Askari, S., Dwivedi, B., Saeed, A. & Nourani, M. (2009). *Scalable mean voting mechanism for fault tolerant analog circuits*. Proceedings of 4th International Workshop on Design and Test (pp. 1-6). Riyadh.