A Novel History based Weighted Voting Algorithm for Safety Critical Systems

PhaniKumar Singamsetty
Department of CSE, Bharath University, Chennai, India
Email: phanikumar.s@gmail.com

SeethaRamaiah Panchumarthy
Department of CS&SE, Andhra University College of Engineering, Visakhapatnam, India
Email: psrama@gmail.com

Abstract— **Safety critical systems are the systems which may lead to hazards, loss of lives and great damage to the property if they fail due to errors which may lead to faults. N-Modular Redundancy or N-Version Programming along with the voter is used in the safety critical systems to mask the faults. In this paper different existing weighted average voting algorithms are surveyed and their merits and demerits or limitations are discussed based upon which a novel History based weighted Voting algorithm with Soft Dynamic Threshold is proposed. Experimentation results of the novel voting algorithm for Triple Modular Redundant (TMR) system are compared with existing voting algorithms and the novel voter is giving almost 100% Safety if two of the three modules are error free and giving better results for one error free module. Novel voter is also giving better results for the multiple error conditions with all the modules having errors.**

Index Terms—**Triple Modular Redundancy, Result Amalgamation, Weighted Average Voters, History Records, Soft Dynamic Threshold, Safety Critical Systems**

I. INTRODUCTION

Safety Critical systems are the systems which may lead to hazards, loss of lives or great damage to the property if they fail. There are different domains in which safety critical control systems are used - Automotives – Drive-by-wire systems, Break by wire systems used in cars, Medicine - Infusion pumps, Caner Radiation Therapy machines etc., Military and Space applications - Rocket launchers, Satellite launchers etc., Industrial Process Control, Robotics and Consumer electronic appliances. There is a need to increase the reliability, availability and safety in all these applications. Faults that occur in these applications may lead to hazardous situations. 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[1][2].

There are different architectural patterns [10] in which redundant modules with a voter are used in the safety-critical systems. All the N-modules or N-versions [3] are designed by different teams to meet the same specifications. All these modules take the same input data, process it and generate the results which will be passed to the voter. The voter has to mask the fault by isolating or avoiding the faulty module and the correct value has to be picked by the voter.

There are different types of voting algorithms [7] mentioned in the literature. Some Voting algorithms like Majority, Plurality voters [4] generate the output if the majority or required numbers of inputs to the voter are matched; otherwise it will generate no output so that the system can be taken to the fail safe state. Adaptive Majority voting algorithm [9] gives better performance by using history records. But for some safety-critical systems, there may not be any fail safe state. In such systems, the voter has to generate some value as the output using some methods like amalgamating the outputs or results of all modules, which is called as *result amalgamation*. Median, average, weighted average voters are some examples for the voters which amalgamate the inputs of the voter and generate some value as the voter output. History based weighted average voters consider the history of the modules and for the highly reliable module high weight is given.

In this research work, Instead of harsh threshold, *Soft threshold* which can be changed dynamically is used to find the agreeability value of each module output with the other remaining module outputs. Harsh threshold results in agreeability value of either 0 or 1 but soft threshold method uses fuzzy Z function to generate agreeability or closeness value as shown in Figure 2.

This Research Paper is organized as follows:
Section II is the literature survey of the existing voting algorithms.
In Section III, Proposed Novel History based weighted Voting algorithm with soft dynamic threshold is given
In Section IV, Experimental method and Test Harness is described
In Section V, Experimental results are analyzed
In Section VI, Conclusions and Future works are given.

doi:10.4304/jait.2.3.139-145

II. RELATED WORKS

In this section, existing weighted average voting algorithms are described and the limitations are discussed.

A. Basic Lorkzok's Standard weighted average voting algorithm (Lorkzok WA):

In this voting algorithm [8] weights are calculated based on the distances between the module outputs as given below

$$w_i = \frac{1}{1 + \prod_{\substack{i=1, j=1 \\ i \neq j}}^{N} \frac{d^2(x_i, x_j)}{a^2}} \quad (1)$$

where $d(x_i,x_j)$ is the distance between the output values of module i and module j and a is a scaling factor.
After assigning the weights, output of the voter is calculated as follows:

$$x_o = \sum_{i=1}^{N} \left(\frac{w_i}{s}\right) . x_i \quad (2)$$

Where *s* is the sum of all the weights
In this algorithm, reliability of the modules in the previous voting cycles called history is not considered.

B. History based weighted average voting algorithm

Algorithm for building history records:

History records [6] are built based on the reliability of the modules. If a module has contribution for the majority consensus of the outputs of all the modules in a particular voting cycle, then a Boolean variable is set to 1 otherwise cleared to 0. The cumulative sum of this Boolean variable up to the current voting cycle is calculated which is the history record of a particular module. A module with high cumulative sum value is the highest reliable module and the one with low cumulative sum value is less reliable module.

This history value is normalized by dividing it by the cycle number and is called as the state indicator P_i of the module i. There are two versions of history based weighted average voters called state indicator based and module elimination based weighted average voting algorithms as described in the reference[6].
In the *state indicator based weighted average voting algorithm (HWA1)*, weights are assigned based on the state indicator Pi value

$$W_i = p_i^2 \quad (3)$$

In *module elimination based weight assignment (HWA2)* method if state indicator value of a module is less than the average state indicator value of all the modules then weight for that module is assigned as zero and eliminated from contributing to the voter output.

$$W_i = 0 \text{ if } P_i < Pavg \quad (4)$$

where $Pavg = (P_1+P_2+P_3+.....+P_N)/N$
Otherwise $W_i = P_i^2$

If we consider Triple Modular Redundancy (TMR), these two versions work well if the same two modules consistently reliable and the other module generates outputs with some error. But in the reality, any module may fail randomly and generate erroneous outputs. The existing history based weighted average algorithms failed to produce the correct results even though majority of the modules have generated the error free outputs. This problem occurred since values for weights are assigned only based upon history. The module which generates correct output in the present cycle may be neglected and zero or less weight may be assigned for that module if it has poor history record. Hence proper weight is not given for the degree of closeness or agreement of a module with other module outputs.

C. Weighted average voter with Soft Threshold (WA ST):

In this voting algorithm [5] Degree of Closeness is calculated. Degree of closeness of each module with other modules is calculated and average agreement value is calculated and assigned as a weight for that module. Threshold is made soft by using a roll-off constant which is tunable. But in this algorithm history is not used. This algorithm generated no output or benign output if all the weights of all the modules are assigned zero value.

In Reference [11], Modified History based weighted average voting with soft dynamic threshold is given. In this work, the threshold is calculated based upon the notional correct output of the voter. It is difficult to predict the voter output before only to decide the threshold. It is a major limitation in this voter.

In Reference [12], a neural network based voter is designed and the neural network is trained using feed forward error back propagation algorithm. It is time taking process to train the network.

III. NOVEL ALGORITHM

A novel history based weighted average voting algorithm with soft dynamic threshold is given below:

1. The distance between the output of module$_i$ x_i and output of module$_j$ x_j is calculated as

$$d_{ij} = |x_i - x_j| \quad (5)$$

2. For i =1 to N and j=1 to N and ($i \neq j$) Find Closeness index $\boldsymbol{S_{ij}}$ using following formula

$$S_{ij} = \begin{cases} 1 & \text{if } d_{ij} \leq vt \\ 1 - \frac{|d_{ij} - vt|}{|n*vt - vt|} & \text{if } d_{ij} \leq n*vt \\ 0 & \text{if } d_{ij} > n*vt \end{cases} \quad (6)$$

Where n is a variable that can be assigned a value >=2 to make the threshold soft.

And d_{ij} is the distance between i and j module outputs and *vt* is the voting threshold.

3. Calculate History values using the procedure given in the Reference [6] but use n*vt as the threshold for agreement while calculating history records. Find the Normalized history values for each module by dividing the history with cycle number.
4. Find History and Closeness Product (HCP) for each module as follows
   For i =1 to N
   For j= 1 to N and i ≠ j

   $$HCP_i = P_i*(\sum S_{ij}/N-1) \quad (7)$$

   Where N is the Total number of modules
   And Pi is a normalized history value of the module i and $P_i=Hist_i/cycleno$
5. Find Normalized History average P_{avg}

   $$P_{avg} = \sum_{i=1}^{N} \frac{Pi}{n} \quad (8)$$

6. Calculate the weights for all N modules as follows
   For i =1 to N
   if $HCP_i=0$ AND $P_i<P_{avg}$
   $W_i=0$ (9)
   Otherwise
   $W_i=2*HCP_i$

7. If all the weights are equal to zero in the worst case, Modify the weights as follows
   $W_i=p_i^2$ for i= 1 to N (10)
8. Calculate weighted average using the weights

   $$x_o = \sum_{i=1}^{N} \left(\frac{w_i}{s}\right).x_i \quad (11)$$

   Where W_i is the weight of i th module
   and x_i is the output of i th module
   and *s* is the sum of all the weights.

IV. EXPERIMENTAL METHOD

Test Harness: Test Harness for experimentation with voting algorithms is shown in the Figure 1. below.

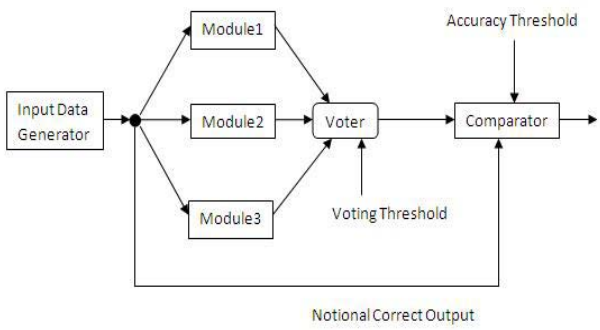

Figure 1. Experimental Setup to evaluate the Performance of a Voter

Cyclic data like Sin wave is generated using the equation given below
Input data = 100 + 100 * sin(t)
Sample rate t is taken as 0.1.

Generated input data is given to each of the modules and the random error of uniform distribution is injected into each of the required module in the required range [-e,+e]. Initially generated input data before injecting the error is considered as the notional correct output.
Fixed voting Threshold and Accuracy Threshold are considered as 0.5
For the Soft Dynamic Threshold methods, Voting Threshold can be varied dynamically.
For the Weighted Average voter with Soft Dynamic Threshold the tunable parameter is taken as 5.
For the Novel Algorithm n=5 taken which is same as tunable parameter of Weighted Average voter with Soft Dynamic Threshold so that results can be compared
Based on the n value threshold is changed. Closeness Index varies as shown in the Figure 2. below based upon the distance measure.

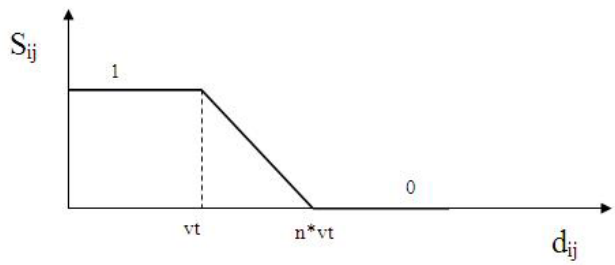

Figure 2. Closeness Index (S_{ij}) versus distance measure (d_{ij}) in the novel algorithm

The generated output by the voter is compared with the notional correct output and if the difference is less than the accuracy threshold value, it is considered as the correct result otherwise incorrect result.
Each set of Experiment is performed for 10000 runs and the number of correct results (nc) and number of incorrect results (nic) are counted.

Then the performance of the voter is evaluated by using the parameters Availability and Safety as given below:

Availability = nc /n
Safety = 1-(nic/n)

Where nc = Number of correct results given by a voter
nic = Number of Incorrect results given by a voter
and n= Total number of runs or voting cycles

V. EXPERIMENTAL RESULTS

Empirical evaluation of the safety performance of the voters is done by running each voter for 10000 voting cycles.

A. Safety with two error free modules:

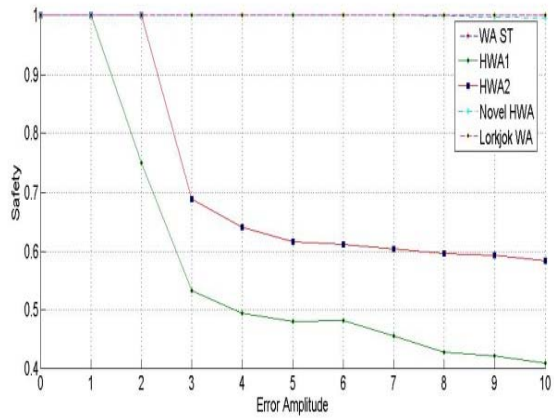

Figure 3. Safety Comparison with two modules error free for large errors for 10000 runs (WA ST, Novel HWA and Lorkjok WA are 100% safe and giving safety values almost equal to 1)

Safety of different voting algorithms with two error free modules for 10000 runs is compared in Figure 3.
In this scenario 1, upto 3000 cycles module1 and module2 are error free and module3 is perturbed with an error in the range [-e, +e], From 3001 to 7000 cycles module1 and 3 are error free and module2 is perturbed with an error in the range [-e, +e], From 7001 to 10000 cycles modules 2 and 3 are error free where as module1 is perturbed with an error in the range [-e,+e].

The two History based weighted average versions called State Indicator based version and Module elimination based version failed to give 100% safety even though two modules are error free. The reason is, much importance is given for previous reliability history but in the current voting cycle, things may be different. A module which has got good history so far may be perturbed with errors in the current voting cycle. But due to its past reliability history, It is given high weight and the erroneous module contributes much for the voter output. This is a major limitation in the two versions of history based weighted average voter which has been overcome in the Novel Algorithm by taking the History and Closeness Product(HCP) into consideration as given in the algorithm while assigning the

Weights assigned and outputs for the given input values are shown in the Table 1. and Table 2. for Module Elimination based weighted average voter and proposed novel History based weighted average voter with Soft Dynamic Threshold respectively.
In the Table 1. and Table 2., Column headings are given below

x – Notional Correct output
x1,x2,x3 are the outputs generated by module1, module2 and module3 respectively.
H1,H2,H3 are history values of the modules and w1,w2,w3 are the weights assigned for the modules and HWA O/P is the output produced by the module elimination version of History based weighted average voter.
NH1, NH2, NH3 are the history values and N_w1,N_w2,N_w3 are the weights assigned for the modules and N_O/P is the output produced by the proposed novel algorithm.

In the Table 1. and Table 2. Third module is perturbed with error upto 20 th cycle where as remaining two modules are error free and there onwards for the remaining voting cycles, Second module is perturbed with error where as remaining two modules are error free. Module Elimination version of History based weighted average voter results are compared with the Novel voting algorithm. If the same two modules are consistently error free, module elimination based version is producing the correct results. But practically this is not possible. Any module may be inconsistent and fail randomly at the runtime. Cycle no 21 onwards, second module is perturbed with errors. But module elimination version gives importance to the previous history and hence gives high weight to the erroneous module2 as shown in Table.1. Due to this high weight, it contributes much for the result. Module elimination version needs some recovery time. Whereas, the novel algorithm considers History and Closeness or Consensus for Majority of each module to assign the weights and is able to produce correct outputs as shown in Table.2., if any two modules are error free.

B. Two Modules have error with equal error Amplitude and One module is Error Free:

In this Scenario2, One module is Error Free. up to 3000 voting cycles, Module1 is error free whereas Remaining two modules are perturbed with equal error amplitude in the range [-e,+e]. From 3001 to 7000 cycles, Module3 is error free and the remaining modules are perturbed with equal error amplitude in the range [-e,+e]. From 7001 to 10000 cycles, Module2 is error free and the remaining modules are perturbed with equal error amplitude in the range [-e,+e].

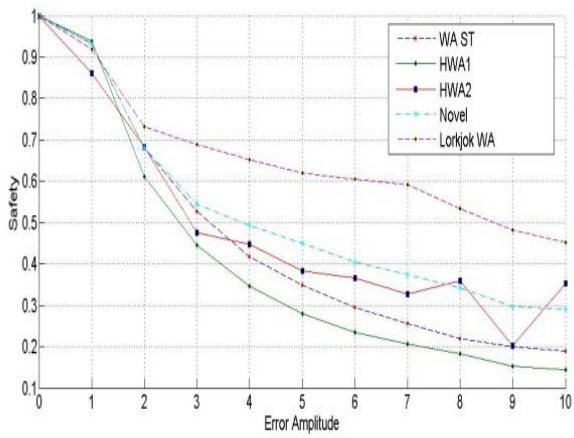

Figure 4. Safety Comparison with One Error free module for Large errors for 10000 runs

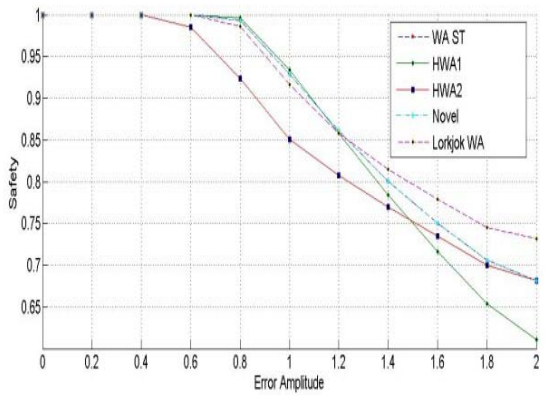

Figure 5. Safety Comparison with One Error free module for Small errors for 10000 runs

Figure 4. shows the safety performance with One Error free module for large errors and Figure 5. shows the safety performance for small errors. With one Error free module, The proposed novel algorithm has better safety performance than the two versions (state indicator version and module elimination version) of the History based weighted average voter and Weighted average voter with Soft dynamic threshold for the small and large errors. But Lorkzok's Weighted average voter has somewhat better performance than the proposed novel algorithm in this scenario since it is considering only the distances between the module outputs in the current cycle but not history.

C. *All Modules have error with equal error Amplitude:*

In this Scenario 3, All the modules are perturbed with errors of equal error amplitude in the range [-e,+e] for all 10000 voting cycles and safety performance is compared.

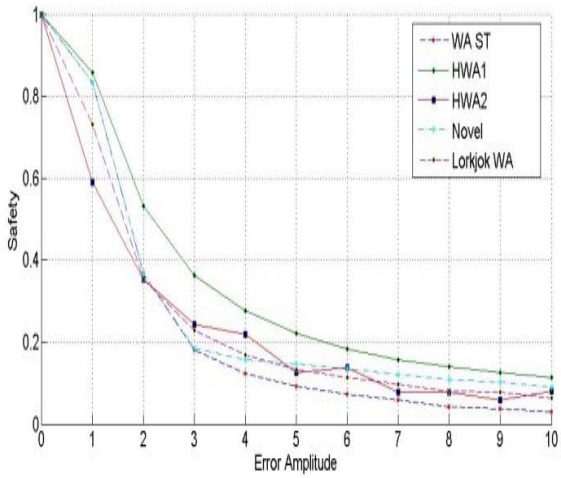

Figure 6. Safety Comparison with all modules have equal amplitude errors (Large Errors) for10000 runs

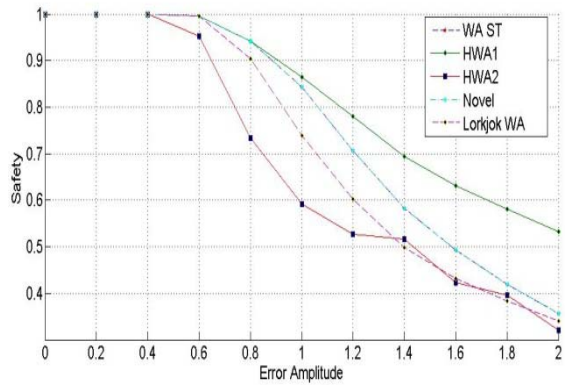

Figure 7. Safety Comparison with all modules having equal amplitude errors (Small Errors) for 10000 runs

Figure 6. shows Safety performance of the different voting algorithms with all modules having errors of equal error amplitude for large errors and Figure 7. shows safety performance for small errors. For the large and small errors, proposed novel algorithm has better safety performance than the Module elimination version of the History based weighted average voter, Lorkzok's weighted average voter and Weighted average voter with Soft dynamic Threshold. But in this case, State Indicator version of the History based weighted average voter is performing better than all other voting algorithms since it considers only history to assign the weights.

VI. CONCLUSIONS & FUTURE WORK

In this work, a Novel History based weighted average voter with Soft Dynamic Threshold is designed and safety performance is evaluated empirically for 10,000 voting cycles on a Triple Modular Redundant system (TMR). *Reliability history* of the modules and *closeness* or *agreeability* of a module output with other module outputs (majority consensus) in a voting cycle are used to assign the weights for the individual modules and final output is generated by calculating the weighted average of all the module outputs.

The Novel voting algorithm is performing better and giving almost 100% safety if majority of the modules are error free which is the much needed behavior for fault masking in the practical applications. Novel voting algorithm is also giving better safety performance for the multiple error scenarios compared to the other history based weighted average voters.

Majority consensus is established if the majority of the modules generate the same output values, which need not be correct. Majority of modules may coincidentally generate the erroneous same output and may cause for the majority consensus and contribute for the final output. This can be overcome using forecasting and prediction algorithms like double exponential smoothing and interpolation to predict the cyclic pattern data output for the current cycle based on the outputs of the past cycles and it remains the future work.

REFERENCES

[1] J.-C. Laprie, "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, June 1985,pp. 2–11.

[2] B. W. Johnson, *Design and Analysis of Fault-Tolerant Digital Systems*. New York: Addison-Wesley, 1989.

[3] L. Chen and A. Avizienis, "N-Version Programming: a fault-tolerance approach to reliability of software operation," in *Digest of Papers FTCS'8: IEEE 8th Annu. Int. Symp. Fault-Tolerant Computing Systems*, Toulouse, France, June 1978, pp. 3–9. York: Addison-Wesley, 1989.

[4] D. M. Blough and G. F. Sullivan, "A comparison of voting strategies for fault-tolerant distributed systems," in *Proc. IEEE 9th Symp. Reliable Distributed Systems*, Huntsville, Alabama, Oct. 1990, pp. 136–145.

[5] G. Latif-Shabgahi, "A Novel Algorithm for Weighted Average Voting Used in Fault-Tolerant Computing Systems," Microprocessors and Microsystems, vol. 28, pp. 357-361,2004.

[6] G. Latif-Shabgahi, J. M. Bass, and S. Bennett, "History-Based Weighted Average Voter: A Novel Software Voting Algorithm for Fault-Tolerant Computer Systems," Euromicro Conference on Parallel, Distributed, and Network-Based Processing, pp. 402-409, 2001

[7] G. Latif-Shabgahi, Julian M. Bass and Stuart Bennett, "A taxonomy for software voting algorithms used in safety-critical systems," IEEE Trans.Reliability, vol. 53, no. 3, pp 319-328, Sept. 2004.

[8] P. R. Lorczak , A. K. Caglayan, and D. E. Eckhardt, "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, 1989.

[9] G. Latif-Shabgahi, S. Bennett, "Adaptive majority voter: a novel voting algorithm for real-time fault-tolerant control systems," in 25th Euromicro Conf., vol. 2, 1999, pp. 113-120.

[10] S.Phani Kumar, P.S.Ramaiah,V.Khanaa, "Architectural Patterns to Design Software Safety based Safety-Critical Systems" Proceedings of ICCCS' 11 International Conference on Communication, Computing & Security,Feb 12-14, 2011, pp:620-623 ACM New York, NY, USA ©2011 ISBN: 978-1-4503-0464-1 doi>10.1145/1947940.1948069

[11] M.Das, S.Battacharya, "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, ISBN: 978-0-7695-4058-0 doi>10.1109/ACE.2010.45

[12] Zarafshan, F, Latif-Shabgahi, G.R., Karimi, "A Novel Weighted Voting algorithm based on Neural Networks for Fault-Tolerance Systems",ICCSIT, 2010, IEEE International Conference on Computer Science and Information Technology, pp.135-139 doi>10.1109/ICCSIT.2010.5565122

TABLE 1.

OUTPUTS GENERATED BY HISTORY BASED WEIGHTED AVERAGE VOTER

Cycle no	x	x1	x2	x3	H1	H2	H3	w1	w2	w3	HWA O/P
16	199.749	199.749	199.749	195.921	16	16	1	1	1	0	199.749
17	199.957	199.957	199.957	205.465	17	17	1	1	1	0	199.957
18	199.166	199.166	199.166	200.042	18	18	1	1	1	0	199.166
19	197.385	197.385	197.385	190.56	19	19	2	1	1	0	197.385
20	194.63	194.63	194.63	184.68	20	20	2	1	1	0	194.63
21	190.93	190.93	191.429	190.93	21	21	3	1	1	0	191.179
22	186.321	186.321	191.554	186.321	22	21	4	1	0.911	0	188.816
23	180.85	180.85	187.615	180.85	23	21	5	1	0.834	0	183.926
24	174.571	174.571	177.004	174.571	24	21	6	1	0.766	0	175.626
25	167.546	167.546	173.889	167.546	25	22	7	1	0.774	0	170.315
26	159.847	159.847	159.01	159.847	26	22	8	1	0.716	0	159.498
27	151.55	151.55	157.04	151.55	27	23	9	1	0.726	0	153.858
28	142.738	142.738	144.704	142.738	28	23	10	1	0.675	0	143.53
29	133.499	133.499	132.126	133.499	29	24	11	1	0.685	0	132.941
30	123.925	123.925	130.267	123.925	30	25	12	1	0.694	0	126.524
31	114.112	114.112	123.683	114.112	31	25	13	1	0.65	0	117.884
32	104.158	104.158	95.7403	104.158	32	25	14	1	0.61	0	100.968

TABLE 2.

OUTPUTS GENERATED BY PROPOSED NOVEL VOTER

Cycle no	x	x1	x2	x3	NH1	NH2	NH3	N_w1	N_w2	N_w3	N_O/P
16	199.749	199.749	199.749	195.921	16	16	1	1	1	0	199.749
17	199.957	199.957	199.957	205.465	17	17	1	1	1	0	199.957
18	199.166	199.166	199.166	200.042	18	18	2	1.8125	1.81249	0.1806	199.208
19	197.385	197.385	197.385	190.56	19	19	2	1	1	0	197.385
20	194.63	194.63	194.63	184.68	20	20	2	1	1	0	194.63
21	190.93	190.93	191.429	190.93	21	21	3	2	2	0.2857	191.163
22	186.321	186.321	191.554	186.321	22	21	4	1	0	0.1818	186.321
23	180.85	180.85	187.615	180.85	23	21	5	1	0	0.2174	180.85
24	174.571	174.571	177.004	174.571	24	22	6	1.0332	0.06082	0.2583	174.68
25	167.546	167.546	173.889	167.546	25	22	7	1	0	0.28	167.546
26	159.847	159.847	159.01	159.847	26	23	8	1.8314	1.47092	0.5635	159.529
27	151.55	151.55	157.04	151.55	27	23	9	1	0	0.3333	151.55
28	142.738	142.738	144.704	142.738	28	24	10	1.2671	0.45794	0.4525	143.151
29	133.499	133.499	132.126	133.499	29	25	11	1.5634	0.97132	0.593	133.072
30	123.925	123.925	130.267	123.925	30	25	12	1	0	0.4	123.925
31	114.112	114.112	123.683	114.112	31	25	13	1	0	0.4194	114.112
32	104.158	104.158	95.7403	104.158	32	25	14	1	0	0.4375	104.158

Mr. S. Phani kumar obtained his Bachelors degree in Computer Science and Engineering from Rural Engineering College, Bhalki, which is affiliated to Visvesvaraiah Technological University, Karnataka –INDIA in 2002 and his Masters degree in Software Engineering from Bharath University, Chennai, INDIA in 2006. He is pursuing the doctoral degree in Computer Science and Engineering from Bharath university, Chennai – INDIA.

He has 9 years of teaching experience and presently working as an Associate Professor in the Department of Information Technology of Lakireddy Balireddy College of Engineering, Mylavaram, Andhra Pradesh, India. He has published two International Journal papers and presented Five International Conference papers in addition to two papers at National Conferences in India His research interests are Safety Critical Computing Systems, Safety Software, Software Engineering, RealTime Systems and Fault Tolerant Systems

Mr. S. Phani Kumar is a member of various professional societies like IEEE and ISTE.

Dr.Panchumarthy Seetha Ramaiah obtained his Ph.D in Computer Science from Andhra university in 1990.

He is presently working as a professor of Computer Science in the department of Computer Science and Systems Engineering, Andhra University College of Engineering, Visakhapatnam-INDIA. His research interests are safety critical systems, VLSI design, Embedded Systems and Fault Tolerant Systems.

Dr.P. Seetha Ramaiah is the Principal Investigator for several Defence R&D projects and Department of Science and Technology projects of the Government of India in the areas of Embedded Systems and robotics. He has published twelve journal papers, and presented twenty International Conference papers in addition to twenty one papers at National Conferences in India.