International Journal of Computer Science and Informatics

Volume 3 | Issue 2 Article 8

October 2013

A VELOUR PLEBISCITE TRACING FOR HARDWARE AND SOFTWARE MISCUE REMEDIAL SYSTEMS

B.V.V.S. PRASAD
CSE Department, Malineni Lakshmaiah Women's Engineering College, vvsprasad.author@gmail.com

V. BALAJI
CSE Department, University College of Engineering, v12.balaji@yahoo.com

M.MADHAVA RAO
CSE Department, CR Reddy College of Engineering, madhavaraomaganti@gmail.com

A VELOUR PLEBISCITE TRACING FOR HARDWARE AND SOFTWARE MISCUE REMEDIAL SYSTEMS

B.V.V.S. PRASAD[1], V. BALAJI[2] & M.MADHAVA RAO[3]

[1] CSE Department, Malineni Lakshmaiah Women's Engineering College
[2] CSE Department, University College of Engineering
[3] CSE Department, CR Reddy College of Engineering
{[2] Guntur, [3]Eluru}, Andhra Pradesh, India
Email: Guntur, Andhra Pradesh, India, vvsprasad.author@gmail.com, v12.balaji@yahoo.com, madhavaraomaganti@gmail.com

Abstract— Referending algorithms are used to adjudicate among the consequences of surplus modules in miscue-remedial systems. Not exactly, popular allies entail a request-precise 'Disciple Point' value to be précised, whereas biased standard allies are incapable to generate a compassionate productivity while refusal concurrence exists between the allies inputs. The disciple is tentatively appraised from the summit of vision security and accessibility, and contrast with the imperfect popular disciple in a Triple Modular Redundancy prearranged outline. We show that the Velour disciple gives extra exact outputs (advanced accessibility) than the wrong popular disciple with petite and hefty slip-ups, fewer false outputs (superior security) than the imperfect popular disciple in the occurrence of tiny slip-ups, and a smaller amount kind outputs than the inaccurate popular choose. The proportion of the kind outputs of the bulk disciples that are fruitfully hold by the Velour fanatic (ensuing in accurate outputs) is more than the proportion of those that are disastrously determined by the Velour fanatic (ensuing in erroneous outputs).

Keywords- *Velour, Disciple Point, Disciple, allies, Triple Modular Redundancy*

I. INTRODUCTION

In many applications, dependency is going on increasing. These applications incorporate safety-critical computer control systems, recognition of a pattern, hugely dependable applications, and extremely accessible systems. Such applications use idleness to decrease the problems connected with depending upon any single component operating perfectly. Triple Modular Redundancy, TMR, and 3-Version Programming, 3VP, are generally used in miscue remedial systems to present reactive redundancy for masking dynamic liabilities at hardware and software levels, correspondingly (Fig. 1). The outputs from three equal elements working in similar with the same inputs are passed to a selection unit that decides connecting them to produce an overall output. The selected output will be accurate so long as a particular number of elements (depending on the

selection strategy) and the selection unit are functioning properly. The selection unit will be referred as *picker* in this paper. The outputs of unnecessary elements supply the *picker inputs.* Due to cost overheads, the amount of unnecessary elements in sensible cases rarely goes beyond 5. There are conditions, yet, where selection of a big quantity of inputs is necessary. Best example is presented in Image Processing filters where, at the time of each pass, pixel assessments can be substituted by assessments determined from selection on an already defined area of a close by position. In this paper, we deal with 3 input disciples regularly used in extremely reliable, very secure, and vastly accessible systems.

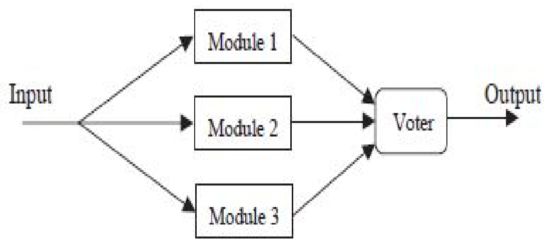


Fig 1: A Triple Modular Redundant System

Not exactly, many disciples generate an output from unnecessary inputs if there is concurrence between a many numbers of disciple inputs [20]. Biased standard disciples always generate an output despite of the concurrence, or else, between unnecessary inputs by merging the inputs. A key complexity with wrong majority disciples is the need to prefer a suitable point value [22], which has a straight contact on the disciple presentation [2]. The difficulty of all recognized biased regular disciples is their failure to generate a gentle output (e.g., no output or secured output) in cases of entire divergence between the disciple inputs. Mutually, categories of disciples are also incapable to survive with doubts linked with disciple inputs generated from untrue software, strident situation, or strident hardware elements (Fig 2). In this, we initiate a novel plebiscite tracing based on the fuzzy set theory which addresses both of these harms. It lessens the ruthless behavior of the inaccurate majority disciple in the neighborhood of the ‘Disciple Point’, and can be observing as a horizontal simplification of the ‘inaccurate majority’ disciple.

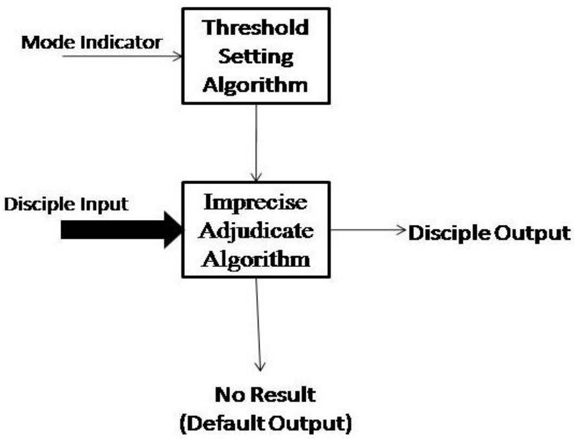


Fig 2: An imperfect disciple with a dynamic point

The narrative ‘Velour disciple’ is the first reported use of an entire Velour disciple in miscue remedial systems, and varies from the further types of fuzzy plebiscite tracings described in the literature [1], [3], [12], [18], [19] which are mostly used for model identification principle, and merging numerous organization systems.

1.1 DISCIPLE ACTIVITIES:

From the previous issues of social sciences, plebiscite is an admired system amalgamation process in various engineering disciplines, especially in safety-critical computer control systems, recognition of a pattern, hugely dependable applications, and extremely accessible systems. In hugely dependable applications, plebiscite will be functional at dissimilar planes; for model:

- at sensor stage to combination of information attained from simulated sensors [6];
- at actuator stage as used in x-by-wire systems and space shuttle [9];
- at control stage, where three hardware elements execute the similar manage task to generate a single output as used in FTMP [11], Tandem Integrity S2 Computing System [13], K-1 Active Dispenser [17], and safety-critical PLC;
- at software level, where three software program execute the similar control actions to generate a single output as used in SIFT;

II. RELATED WORKS

Selection on the outcomes of surplus components with distinct values is simple, and is called as accurate choice. The 3-input accurate majority disciple, for example, generates an output when 2-out-of-3 of its inputs are identical. However, exact selection on the outputs of unnecessary elements with real number outputs is not apt. The necessary variations might happen from dissimilarities in sensor calibration, data communication faults, quantization, instances, and/or rounding faults [8]. A number of explanations have been projected to tackle this trouble. In those explanations, the simple and easy method is dependent on the usage of median-sector algorithm [14], and it is very much useful for dealing with the output of unnecessary sensors. In this algorithm, it chooses the mid-value of the disciple’s input and then it utilizes that input value directly as the disciple output.

Another explanation for dealing estimated unnecessary values is the use of imperfect (point) disciples. In imperfect plebiscite, a little difference between the inputs is permitted; contract now means that the unnecessary outputs are not accurately the similar, but the dissimilarity among them is less than an exacting point. The value of this point is called the *compatibility point* and it is functioning precise. Deliberating the limits on the standard divergence among the outputs of unnecessary units for a system’s complete processing time gives an approximation for the value of compatibility point. There have been done numerous trials for formalizing, applying and choosing an absolute value from the approved input values, and selecting the point value of imperfect disciples on this basis [2], [15], [16], [22]. However, the use of imperfect disciples with a flat point value at the control and computing levels is difficult for some reasons: (i) the choice of the point is important and there is no logical loom for setting this value; (ii) a few suitable unit outputs may be denied when using a fixed point value; and (iii) disciples with fixed point values are incapable to differ their reaction in the appearance of dissimilar levels of deviation in safety-allied Activities such as phased-mission systems. Imperfect plebiscite with an dynamic point has been suggested as a way of solving problems (i) and (ii). In this disciple, the value of the point is determined on-line as a role of input curve and input data values in each plebiscite cycle. Experimental outputs reveal the advantage of a disciple with a dynamic point value over an disciple with a flat point in conditions of system safety and dependability. In recent times, the notion of a *flexible point* as the basis of a novel disciple was represented [7]. This disciple permits the user to set a range, as an alternative of a permanent value, as a disciple point. It levels the go/no-go performance of the imperfect disciple with an unchanging point value at the locality of the point, and determines all of the troubles declare above for permanent-point disciples. The point range is again activity-precise. Flexible disciples perform as a biased standard disciple within the point range, and as an imperfect majority disciple outside this range.

An imperfect disciple with an unchanging point value may cause troubles in many real time control systems. In multi-state security-critical systems some of the prepared modes are more serious than the others; in a flight control system, for example, take-off and landing modes are more miscue/fault-level than the rising, sliding, and cruising modes. Thus the miscue remedial methods used for lofty-serious outfitted modes must vary from that of the fewer-serious modes. While using a TMR miscue pretense approach, the previous modes need a plebiscite algorithm with a suspiciously chosen point value whereas the end modes are possible to work properly with a better point value. Assume that in the ready state A, the disciple is faced with data from the period [1 5], and in state B it is tackled with data from the period [100 150]. At this time, decision among unnecessary data from the two dissimilar periods with the same point value (e.g., 1.0) is doubtful. Clearly, judging among unnecessary small numbers needs a smaller point value than arbitrating linking the unnecessary large real numbers. To facilitate, for state A the disciple inputs *{1 2 3}* (with the divergence of 1.0 from each other) are more likely considered in disagreement whereas for state B, disciple inputs {120, 121, 122} with the same divergence are measured in concurrence. The disciple point is relative to the predictable numerical values of its inputs. The relative coefficient is a activity detailed restriction.

The Fig 2 structure is making clear by using a theoretical flight managing example shown in Table 1. In this example, it is expected that in elevated-significant modes (take-off and landing) the disciple is bumped with small input values (from the range [1 5]), therefore a small point value has been set for this mode ($\frac{1}{10} a_{emax} = 0.48$, where a_{emax} is the higher posse of the range). For less-significant modes (e.g., cruising mode) the disciple is encountered with large numbers (from the interval [10 20]), and therefore, a large point value is chosen ($\frac{1}{7} a_{emax} = 3.11$).

Table 1: An example for setting a Disciple Point

Ready Mode	Pointer of the Mode	Range of disciple output	Disciple Point
Flight Taking Off	R	$0 < a_e \leq 5$	$\frac{1}{10} a_{emax}$
Increasing	S	$5 < a_e \leq 10$	$\frac{1}{7} a_{emax}$
Decreasing	S	$5 < a_e \leq 10$	$\frac{1}{7} a_{emax}$
Swiftness	T	$10 < a_e \leq 20$	$\frac{1}{6} a_{emax}$
Flight Landing	R	$0 < a_e \leq 5$	$\frac{1}{10} a_{emax}$

In this paper, we introduce a Velour plebiscite tracing, an improvement of the imperfect mainstream disciple. Beneath this loom, the primary difficulty of selecting a permanent value or a flexible range for disciple point is largely curved, and the force of doubts are considered into account. In addition to this, the central fuzzy rules allocate a zero weight value for all disciple inputs in whole divergence Referending cycles.

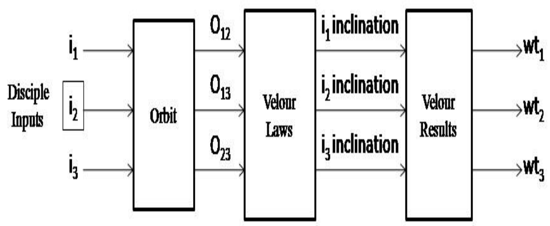


Fig 3: Configuration of a 3-input Velour Plebiscite unit

The later potential outputs in a kind output (e.g., a secure output) for the disciple, and increases its security level. The fuzzy set theory has previously been used for calculating the ultimate disciple output, among the arranged disciple inputs, of an imperfect disciple [16]. Fuzzy set theory has also been used to develop the consistency of categorization procedure in pattern recognition systems [5]. However, the use of fuzzy set theory for arbitrating between unnecessary values is narrative.

III. VELOUR DISCIPLE

With the preceding assistance, the Velour disciple utilizes fuzzy logic to create the weights essential for calculating a biased standard disciple output. Fig. 3 shows the basic Configuration of a 3-input Velour Plebiscite unit.

3.1 Calculating the Velour Results of Disciple inputs:

In this, the first pace in the loom needs the definition of a Velour variation inconsistent to illustrate each pair of inputs to the disciple. For each pair i_m and i_n with statistical orbit O_{mn}, based on the triangular relationship services shown in Fig. 4, we delineate a Velour variation inconsistent represented by a set of relationship grades μ_R (O_{mn}) where R: *{*low; average; high*}*. At the time of using balanced sets, this needs two bound values to be particular. On the basis of statistical variation between any two inputs, a non-zero relationship grade will be allotted to one or two of the fuzzy sets defined for the consequent Velour variation inconsistent. For expediency, triangular Velour relationship services are used. This definition was implemented as a normal development from the simpler *flexible disciple* described in [23] which an incline service in place of the fixed rigid point found in conventional imperfect common disciples.

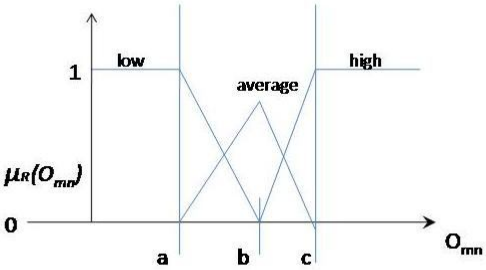


Fig 4: Definition of the variation inconsistent relationship services, $\mu_R(O_{mn})$ where R: {low; average; high}

We describe the inconsistent and Velour relationship services as follows:

Dissimilarity among two disciple-inputs:
$O_{mn} = |i_m - i_n|$, where $m \neq n$ (1)
Proportion:
$c - b = b - a$ (2)
Where a, b and c are real numbers, and a<b<c

$$\mu_{low} = \begin{cases} 1 : O_{mn} \le a, \\ \dfrac{b - O_{mn}}{(b-a)} : a < O_{mn} \le b, \\ 0 : b < O_{mn} \end{cases} \quad (3)$$

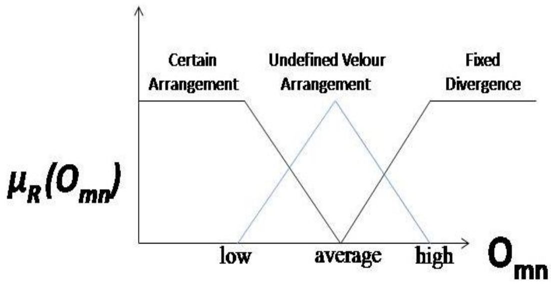
$$\mu_{average} = \begin{cases} 0 : O_{mn} \le a, \\ \dfrac{O_{mn} - a}{(b-a)} : a < O_{mn} \le b, \\ \dfrac{c - O_{mn}}{(c-b)} : b < O_{mn} \le c, \\ 0 : c < O_{mn} \end{cases}$$
(4)

$$\mu_{high} = \begin{cases} 0 : O_{mn} \le b, \\ \dfrac{O_{mn} - b}{(c-b)} : b < O_{mn} \le c, \\ 1 : c < O_{mn} \end{cases} \quad (5)$$

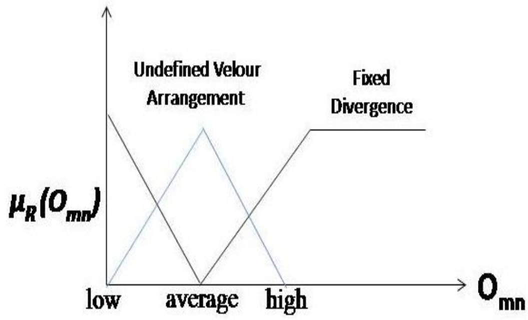
Commonly, in a k-way disciple, there are k (k-1)/2 Velour variance variables. For every change, the tracing will effect in a non-zero relationship assessment being dispersed to one or two of the fuzzy sets separate for that inconstant.

In the extreme circumstance of a=b=c (which we word 'inflexible differencing'), we identify that two inputs will be consigned fixed (unity) relationship of either the low or the high variance set. In this structure, the Velour disciple repeats a customary secure-point popular disciple.

The classification of the Velour variance sets consents the narration of the Velour disciple to be regulated. Fig. 5 illustrates two qualitatively distinct Velour variance in constants. In the first case, there is an important section in which two inputs which contrast by a non-zero extent are considered as being in certain arrangement; a midway section in which the variation is identified using dialectal inconstant that may be correct to a slighter of larger amount (for instance, the variation among two inputs may be such that a non-zero relationship is presented to the low and average Velour variation sets); and a third section which classifies inputs that are in fixed divergence. In the second case, Velour inconstant, there is no province of certain arrangement identified, while there is a section of fixed divergence.



(a)



(b)

Fig 5: Qualitatively dissimilar descriptions of Velour variances.

3.2 Outlining the Velour Inclination of Every Input:

Per every inputs of i_m, before we express a Velour inclination in constant, $\mu_S(wt_m)$ with reverence to the other inputs. The Velour inclination assessment is a quantity of the degree to which an input accepts with the other two inputs. Respect to this, the loom is parallel to the biased regular disciples which determine the weights as a purpose of the orbit among deviations.

For every Velour inclination inconstant, we describe five intersecting fuzzy sets S: (vsmall, small, average, large, vlarge). An example of the fuzzy set definition is shown in Fig. 5. The Velour inclination inconstant is essentially definite above the range [0 1]. The inclination measure (and inclination evades) is took out from the error model [2], where any disciple input is labeled as *out of range, improper, adequate,* and *accurate* value. Varying the constraints j, k, and l has straight affected on the output of the disciple which, in order, vagaries the security and obtain ability concert of the disciple. The Velour inclination value is used to identify the biased involvement of the consistent input to the disciple output.

3.3. Fuzzy rule set description: Qualitatively mapping Velour variations to Velour inclination:

From the below Table 2, it describes a rule matrix that go over one feasible set of fuzzy rules for joining and plotting Velour dissimilarity values onto a Velour inclination value in a 3-input system. In this, the matrix will be steady with the categorization of disciple input inconsistencies shown in [22]. Other developments of the rules are potential, but we have to bind our own to the subsequent in this paper, because it has the majority expected elucidation.

		O_{mn}		
		Low	**Average**	**High**
O_{mp}	**Low**	Vlarge	Large	Average
	Average	Large	Small	Vsmall
	High	Average	Vsmall	Vsmall

Table 2: Rule matrix used for Velour input inconstant

For instance, let us take a three input disciple with inputs i_1, i_2 and i_3. At this time, we can describe a set of fuzzy rules to explain the inclination of input i_1 (i.e., the measure of its concurrence with the other two inputs) in accordance with Table 2 as follows:

1) IF (O_{12} is low) AND (O_{13} is low) THEN i1-inclination is vlarge.
2a) IF (O_{12} is low) AND (O_{13} is average) THEN i1-inclination is large.
2b) IF (O_{12} is low) AND (O_{13} is low) THEN i1-inclination is large.
3a) IF (O_{12} is high) AND (O_{13} is low) THEN i1-inclination is average.
3b) IF (O_{12} is low) AND (O_{13} is high) THEN i1-inclination is average.
4) IF (O_{12} is average) AND (O_{13} is average) THEN i1-inclination is small.
5a) IF (O_{12} is high) AND (O_{13} is average) THEN i1-inclination is vsmall.
5b) IF (O_{12} is average) AND (O_{13} is high) THEN i1-inclination is vsmall.
5c) IF (O_{12} is high) AND (O_{13} is high) THEN i1-inclination is vsmall.

		O_{mn}		
		Low	**Average**	**High**
O_{mp}	**Low**	Vlarge	Average	Large
	Average	Average	Small	Vsmall
	High	Large	Vsmall	Vsmall

Table 3: Different, Unusable rule matrix

For this, we don't have any common explanation for logical operators. All together, the connection of two fuzzy set is executed by a t-norm operator, and the combination of two fuzzy set is anecdotal by an s-norm operator. In the following execution of the rules, t is represented by the min operator, and s is represented by the max operator:

(1′) $\mu_{vlarge}(wt_m) = \min\{\mu_{low}(O_{mn}), m \neq n\}$ (6)

(2′) $\mu_{large}(wt_m) = \max\{\min[\mu_{low}(O_{mn}), \mu_{average}(O_{mp}), \forall p: p \neq m, n], \forall n: n \neq m\}$ (7)

(3′) $\mu_{average}(wt_m) = \max\{\min[\mu_{low}(O_{mn}), \mu_{high}(O_{mp}), \forall p: p \neq m, n], \forall n: n \neq m\}$ (8)

(4′) $\mu_{small}(wt_m) = \min\{\mu_{average}(O_{mn}), m \neq n\}$ (9)

(5′) $\mu_{vsmall}(wt_m) = \max\{\{\min[\mu_{low}(O_{mn}), \mu_{high}(O_{mn}), \forall p: p \neq m, p \neq n], \forall n: n \neq m\}, \{\mu_{high}(O_{mn}); m \neq n\}\}$ (10)

Rather than a Takagi – Sunego loom, with its undeviating task in the rule output, that A Mamdani–Larsen differencing method was used which well discards a collective-logic verdict of which inputs was maximum identical, and does not need the assortment of output task limitations.

3.4 Results of Velour inclination to its input bias standards:

The easy and best way for getting the Velour results is MIRROR RULE [10] and Larsen's Product operation rule [21]. With the usage of Velour inclination sets defined earlier, the three central sets (small, average and large inclinations) are balanced and the two risky sets (vsmall and vlarge) are abstract to be disallowed at the margin. The tiniest and determined distinct values that we defined are 0 and 1, correspondingly. The described five sets are must and should have to be located in the same region. Now, it is essential to have a general awareness on cancroids, Q, for defuzzification of each of the inclination sets showed in Fig 5. The tracing continues as follows:

$Q_{vsmall} = 0$, $Q_{small} = 0.25$, $Q_{average} = 0.50$, $Q_{large} = 0.75$, $Q_{vlarge} = 1$ (11)

aggSet = {vsmall, small, average, large, vlarge},

$$\forall\, m: wt_m = \frac{\sum_{n \in aggSet} inclination_{m,n}\, Q_n}{\sum_{n \in aggSet} inclination_{m,n}} \quad (12)$$

An additional option, and further common loom, to defuzzification (using Larsen's product rule), is competent of getting the result of a Velour inconstant distinct more than any number of randomly sized sets by enchanting into description the areas of the fuzzy sets that lie down within the distinct variety of the set. In this case, the Velour value is given by the following equation where $Expanse_n$ is the region of the set within the distinct variety of the inconstant.

$$q_i = \frac{\sum_{n \in aggSet} inclination_{m,n} \, Expanse_n \, Q_n}{\sum_{n \in aggSet} inclination_{m,n} \, Expanse_n} \quad (13)$$

3.5 Biased disciple output computation:

The bias values wt_m are used in the usual biased standard disciple outline for calculating the disciple output j:

$$j = \frac{\sum_{m=1}^{k} i_m \, wt_m}{\sum_{m=1}^{k} wt_m} \quad (14)$$

IV. EXPERIMENTAL METHODOLOGY

The particulars of new assessment connection for software disciples used in this toil, and the technique of tests have been represented in [4], and are momentarily explicated beneath.

4.1. Assessment connection structure:

The traditional new assessment connection, shown in Fig. 6, suggests a TMR system. It includes an input assist initiator, three defilers (to insert mistakes to pretended input data), a disciple, and a comparator. The assist initiator constructs one imaginary accurate output in each assessment cycle. This series of figures replicates equal accurate outputs created by surplus units. Duplicate of the imaginary accurate output are sent to each defiler in each cycle. The results of all defilers are preferred to be scanned disciple, and disciple result is evaluated by the cycle imaginary accurate output by ways of the comparator.

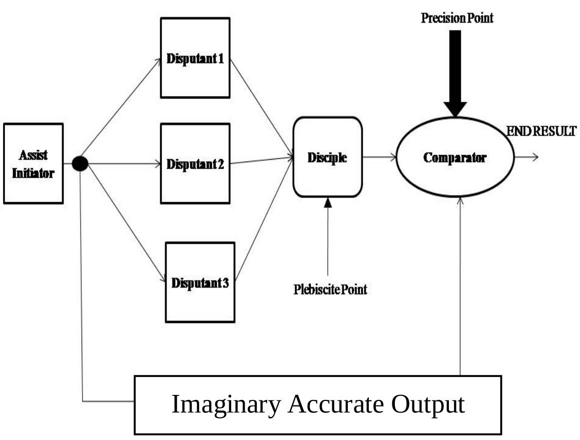


Fig 6: Assessment Connection Structure

V. EXPERIMENTAL RESULTS

This primary trial is traditional that the Velour disciple carries out properly; contrast its consequences with those of the imperfect greater part disciple in some preferred input situations. Table 4 demonstrates the output of the fuzzy-A and imperfect greater part disciples in 14 autonomous cases. In these cases the estimated accurate output is implicit to be 1. The primary channel of equally disciples is subjected to tiny slip-ups while the additional two channels are subjected to huge blunders.

CASE	DISCIPLE INPUTS	VELOUR – A RESULT	COMMON DISCIPLE
1	[1 1.1 1.2]	1.12	1
2	[1 1 1.5]	1.15	1
3	[1 1 12]	1.12	1
4	[1 1.1 1.7]	1.19	1.22
5	[1 1.4 2.1]	1.72	1.64
6	[1 1.5 1.9]	1.62	1.61
7	[1.3 1.8 2.1]	1.67	1.73
8	[1.2 1.7 2.5]	1.51	1.71
9	[1.1 1.6 2.3]	1.72	---
10	[1 1.6 2.3]	1.91	---
11	[1 1.6 3.1]	---	---
12	[1.2 1.6 2.8]	1.36	---
13	[1 1.5 0.2]	1.28	---
14	[1 1.7 22]	1.38	---

Table 4: Illustrated Disciple Outputs

In the above circumstances 1–7, all disciples have related act. Like to the common disciple, the Velour disciple is capable to effectively reject a remote input when creating the disciple result. In the circumstance 8, the common disciple creates an erroneous result but the Velour disciple gives an accurate output. Yet, in 9 and 10 circumstances, the common disciple produces no output but the Velour disciple generates erroneous results. In the circumstance 11 both of the disciples provide no output, as estimated. In 12–14 circumstances, which replicate an absolute failure in one of the inputs, the common disciple gives no result but the Velour disciple is capable to generate accurate outputs.

The outputs of probing the disciples with a lot of additional input cases direct us to the subsequent preliminary ending.

(A) When the common disciple does well in generating the result, either accurate or erroneous, the

Velour disciple also is successful. That is, n_a (vel) ≥ n_a (com).

(B) In a lot of circumstances in which the common disciple generates a gentle output, the Velour disciple can generate a result. To perceive that what proportion of such results is accurate or erroneous, we composed the outputs from 1000 selection cycles (Fig. 7). On this stature, n_d(com) shows the amount of gentle results of the common disciple (gentle cycles); n_d(vel), n_c(vel) and n_{ic}(vel) represent the amount of kind, accurate and erroneous results correspondingly of the Velour-A disciple during those gentle cycles of the common disciple. For this reason, n_d (com) $=n_d$ (vel) $+n_c$ (vel) + n_{ic} (vel). The stature represents that:

- The amount of gentle results of the Velour disciple is constantly less than that of the common disciple: n_d(vel) ≤ n_d(com). This means that Velour disciple is proficient of managing number of several fault cases than the common disciple.
- The common disciple produces some gentle results up to e_{max} =1 (i.e., tiny faults). This it cannot manage some of the numerous fault cases generated by tiny faults. The Velour disciple effectively manages all such numerous faults, and generates an accurate result in every case.

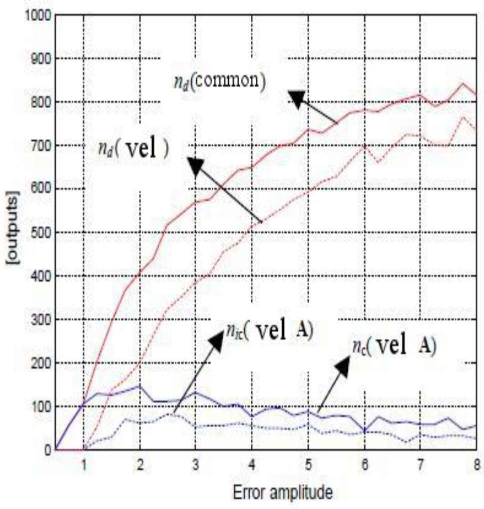
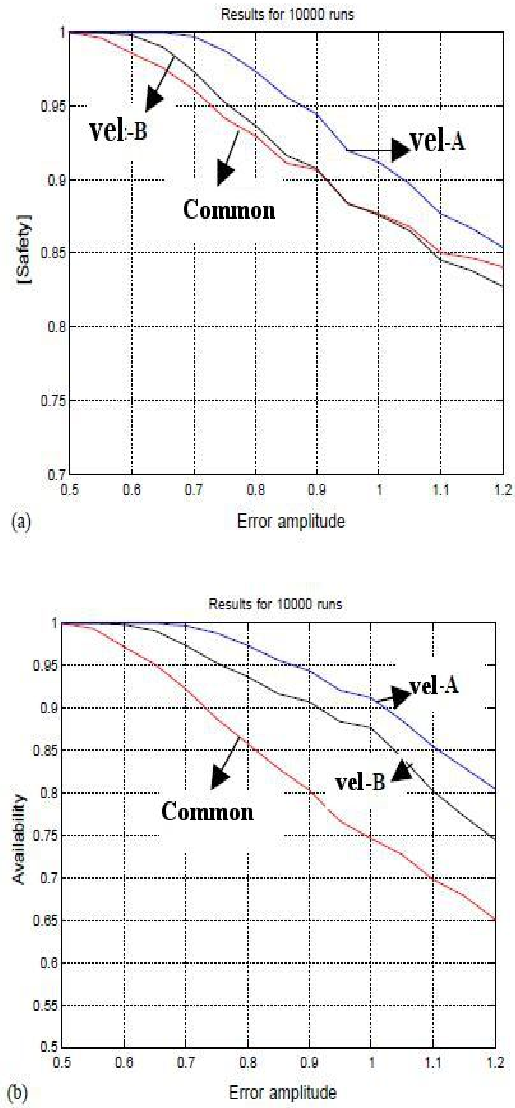


Fig 7: The amount of gentle cycles of the common disciple and all various (gentle, accurate, and erroneous) results of the Velour-A disciple product during the gentle cycles of the common disciple.

5.1 Security and Ease of Use Assessment with Tiny Faults:

Fig. 8 represents the security and ease of use assessment tracings of the common, vel-A and vel-B disciples for tiny faults. Vel-A disciple has higher security (3–5%) and ease of use (8–15%) than the other disciples in the occurrence of tiny faults. Such developments are considerable in many security-related and hugely accessible applications. Vel- B disciple gives more security than the common disciple only up to point e_{max} =0:9 but offers huge accessibility than that disciple for all tiny faults.

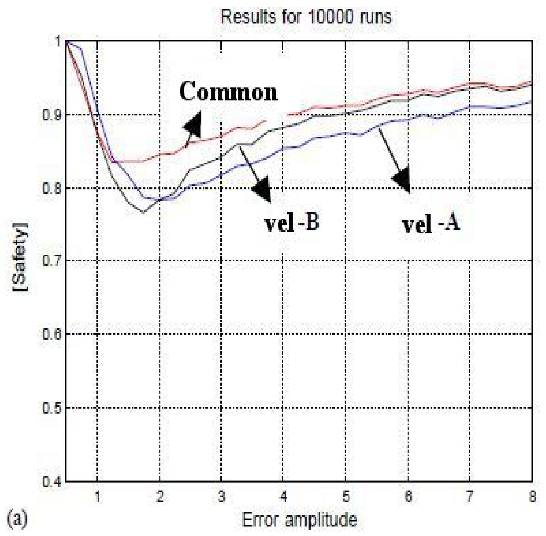
Fig 8: Disciple Presentation with tiny faults
a) Security b) Ease of use

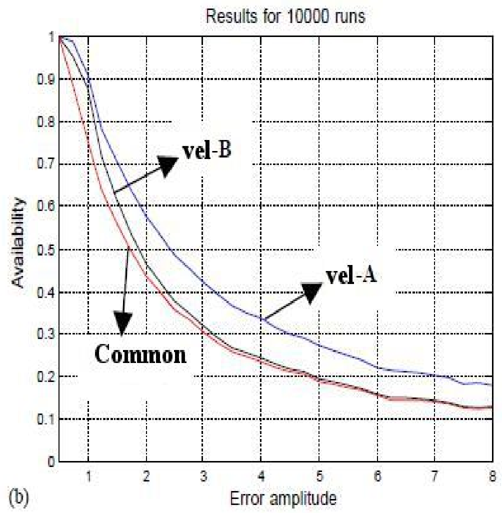


5.2 Security and Ease of Use Assessment with Huge Faults:

Fig. 9 represents the security and ease of use assessments of the compared disciples for an extensive range of faults. With huge faults (i.e., where e_{max} >~1.2) the ease of use of the Velour disciples is larger than that of the common disciple, significantly so in the circumstance of disciple vel-A. Though, for such huge faults and in contrast to the tiny fault case, the common disciple has a 3–8% improved security presentation than the Velour disciples, with disciple vel-B performing better than disciple vel-A in most circumstances.

Fig 9: Disciple Presentation with Huge Faults a) Security b) Ease of use





VI. CONCLUSIONS

A Velour plebiscite tracing has been introduced in this paper. It can be regarded as a curved imperfect common disciple. When the imperfect common disciple succeeds in generating a result, either accurate or erroneous, the Velour disciple does well. The Velour disciple softens the rough performance of the imperfect common disciple in the region of the pointed 'Disciple Point', and manages doubts and some of the numerous fault cases in the location distinct by the Velour input inconstant.

The disciple works by captivating the arithmetical orbit among the input pairs as input parameters, and connecting it with a Velour linguistic inconstant defining low, average, and high dissimilarities between each input pair. Five simple fuzzy rules are used to determine the Velour inclination of each input with respect to the others. The inclination inconstant describes the extent to which an input selectively agrees with both of the other inputs. The Velour inclination of each input is then defuzzified to give the crusty values wt_m that are used as the biasing factor of the disciple inputs when calculating its concluding result

Experimental results showed that the Velour disciple gives low gentle results than the imperfect common disciple. The proportions of the gentle results of the imperfect disciple are effectively managed by the Velour disciple is more than the proportion of those that are ineffectively cleared by the Velour disciple. In the experiments carried out, both transformation of the Velour disciple produces larger accessibility than the standard imperfect common disciple; so they are probably an appropriate contender for large obtainable systems. On the other hand, from the security viewpoint, Velour disciples are better to the common disciple only in the presence of tiny faults. Apparently, a suitable change of Velour inconstant perk ups their presentation.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the contributions of the reviewers. Their comments have considerably improved the quality of this article. We, authors express gratitude to all the anonymous reviewers for their affirmative annotations among our paper.

REFERENCES

[1] J.F. Baldwin, J. Lawry, T.P. Martin, A mass assignment theory of the probability of fuzzy events, Fuzzy Sets and Systems 83 (3) (1996) 353–367.

[2] J.M. Bass, P.R. Croll, P.J. Fleming, L.J.C. Woolliscroft, Three domain voting in real-time distributed control systems, Proc. 2nd Euromicro Workshop on Parallel and Distributed Processing, 1994, pp. 317–324.

[3] N. Belacel, M.R. Boulassel, Multicriteria fuzzy classi6cation procedure PROCFTN: methodology and medical application, Fuzzy Sets and Systems 141 (2) (2004) 203–217.

[4] S. Bennett, G. Latif-Shabgahi, and Evaluation the performance of voting algorithms used in fault tolerant control systems, Proc. 14th World Congress of Internat. Federation ofAutomatic Control, Vol. Q, Beijing, PR China, July 5–9, 1999, pp. 525–530.

[5] V. Chatzis, I. Pitas, Fuzzy cell hough transform for curve detection, Pattern Recognition 30 (12) (1997) 2031–2042.

[6] D. Chen, A. Albrecht Schmidt, H. Gellersen, 1999. Architecture for multi-sensor fusion in mobile environments, Proc. Internat. Conf. on Information Fusion, Sunnyvale, CA, USA, Vol. II, pp. 861–868.

[7] A.D. De Leon, Voting algorithms, M.Sc. Thesis, Department of Automatic Control and Systems Engineering Department, The University of Sheffield, Sheffield, UK, 2003.

[8] Z. Georgiev, M. Stojcev, VLSI common voting module for fault-tolerant TMR system in industrial system control applications, Internat. J. Electron. 76 (2) (1994) 163–205.

[9] B. Hendentz, R. Belschner, Brake-by-wire without mechanical backup by using TTP-communication protocol, URL:http://www.vmars.tuwien.ac.at/projects/xbywire/projects/new-BBW.html.

[10] A. Hopgood, Intelligent Systems for Engineers and Scientists, 2nd Edition, CRC Press, Boca Raton, FL, 2001.

[11] A.L. Hopkins, T.B. Smith, J.H. Lala, FTMP—a highly reliable fault-tolerant multiprocessor for aircraft, Proc. IEEE 66 (1978) 1221–1239.

[12] H. Ishibuchi, T. Nakashima, T. Morisawa, Voting in fuzzy rule-based for pattern classi6cation problems, Fuzzy Sets and Systems 103 (2) (1999) 223–238.

[13] D. Jewett, A fault-tolerant unix platform, Digest of IEEE 21st Ann. Internat. Symp. on Fault-Tolerant Computing Systems, June, Montreal, Canada, 1991, pp. 512–519.

[14] B.W. Johnson, Design and Analysis ofFault-Tolerant Digital Systems, Addison Wesley, Reading, MA, 1989.

[15] K. Kim, M.A. Vouk, D.E. McAllister, An empirical evaluation ofmaximum likelihood voting in failure correlation conditions, Proc. ISSRE'96 pp. 330–339.

[16] K. Kim, M.A. Vouk, D.F. McAllister, Fault tolerant software voters based on fuzzy equivalence relations, Proc. IEEE Aerospace Conf. 4 (1998) 5–19.

[17] D. Kohrs, S. Knowles, D. Cochran, R. Curtis, G. Lai, Beyond LEO: the K-1 active dispenser, Presented at the Space Technology and Applications Internat. Forum, Albuquerque, New Mexico, February 11–14, 2001, URL: http://www.kistleraerospace.com/publications/.

[18] L.I. Kuncheva, C.J. Whitaker, C.A. Shipp, Is independence good for combining classi6ers?, Proc. Internat. Conf. on Pattern Recognition, Barcelona, Spain, 2000.

[19] L. Lam, C.Y. Suen, Application of majority voting to pattern recognition: an analysis of its behaviour andperformance, IEEE Trans. Man Systems Cybernet. A: Systems Humans 27 (2) (1997) 553–568.

[20] G. Latif-Shabgahi, J.M. Bass, S. Bennett, A taxonomy for software voting algorithms used in safety-critical systems, IEEE Trans. Reliability (2004), in press.

[21] C.C. Lee, Fuzzy logic in control systems: fuzzy logic controller—part I, IEEE Trans. Systems Man Cybernet. 20

(1990) 404–418.

[22] P.R. Lorczak, A.K. Caglayan, D.E. Eckhardt, A theoretical investigation of generalised voters, Digest of IEEE 19th Ann. Intenat. Symp. on Fault-Tolerant Computing Systems, Chicago, IL, 1989, pp. 444–451.

[23] M.K. Stojcev, J.Lj. Djordgevic, M.D. Krstic, hardware mid-value select voter architecture, Microelectron. J. 32 (2001) 149–162.

Authors:

B. V. V. S. Prasad (PhD) is a Research Scholar under the Guidance of Dr. G. Manoj Someswar and completed M.Tech in Computer Science & Engineering. He is presently working as an Asst. Prof. in Malineni Lakshmaiah Women's Engineering College, Pulladigunta, and Guntur, India. He is having about 5years of teaching experience in different Engineering Colleges and an associate member of CSI and life member of ISTE, Member in IEEE. Undergone CIT program conducted by IIIT Hyderabad and published 5 international journals in the area of Data Mining and Image Processing.

V. Balaji completed his M.Tech in CSE in Aacharya Nagarjuna University. He is working as Asst. Professor in CSE Department, University College of Engineering and Technology, Aacharya Nagarjuna University, Nagarjuna Nagar, Guntur. He is Having 7 years of teaching Experience. His research Areas are Data mining.

M. Madhava Rao is a Research Scholar in CSE Department, ANU Campus. He completed his M.Tech in CSE in Aacharya Nagarjuna University. He is working as Asst. Professor in C. R REDDY College of Engineering, Eluru. He is having 7 years of Experience in Teaching.