# Fusion of Redundant Information in Brake-By-Wire Systems Using a Fuzzy Voter

**REZA HOSEINNEZHAD**

**ALIREZA BAB-HADIASHAR**

Swinburne University of Technology

**In safety critical systems such as brake-by-wire, fault tolerance is usually provided by virtue of having redundant sensors and processing hardware. The redundant information provided by such components should be properly fused to achieve a reliable estimate of the safety critical variable that is sensed or processed by the redundant sensors or hardware. Voting methods are well-known solutions for this category of fusion problems. In this paper, a new voting method, using a fuzzy system for decision-making, is presented. The voted output of the proposed scheme is a weighted average of the sensors signals where the weights are calculated based on the antecedents and consequences of some fuzzy rules in a rule-base. In a case study, we have tested the fuzzy voter along with the well-known majority voting method for a by-wire brake pedal that is equipped with a displacement sensor and two force sensors. Our experimental results show that the performance of the proposed voting method is desirable in the presence of short circuits to ground or supply, excessive noise and sensor drifts. Voting error (in terms of mean square error) is reduced by 82% by the proposed fuzzy voting method, compared to majority voting.**

Manuscript received December 28, 2004; revised September 30, 2005 and April 25, 2006.

Refereeing of this contribution was handled by Professor David Hall.

This research work was supported by Research Centre for Advanced By-Wire Technologies (RABiT) and Pacifica Group Technologies Pty. Ltd.

Authors' address: Faculty of Engineering and Industrial Sciences, Swinburne University of Technology, Hawthorn, Victoria 3122, Australia, E-mail: (rhoseinnezhad@swin.edu.au, abab-hadiashar@swin.edu.au).



## 1. INTRODUCTION

Brake-by-wire is a frontier technology that will allow many braking functions to switch to electronic actuation. Its deployment will lead to more effective and safe braking systems, elimination of hydraulic technology, release of space and reduction of maintenance. Design and implementation of brake-by-wire systems has recently attracted interest from researchers in automotive and control engineering [9–12, 17]. The general architecture of a brake-by-wire system is shown (in schematic form) in Fig. 1. The figure shows that a large variety of sensors are utilised in a brake-by-wire system and therefore their consistent operation is vital for the functionality of such a system. To achieve a high level of coherency amongst such a large collection of sensors (mandated by the safety requirement of a brake system), the use of sophisticated data fusion techniques is unavoidable.

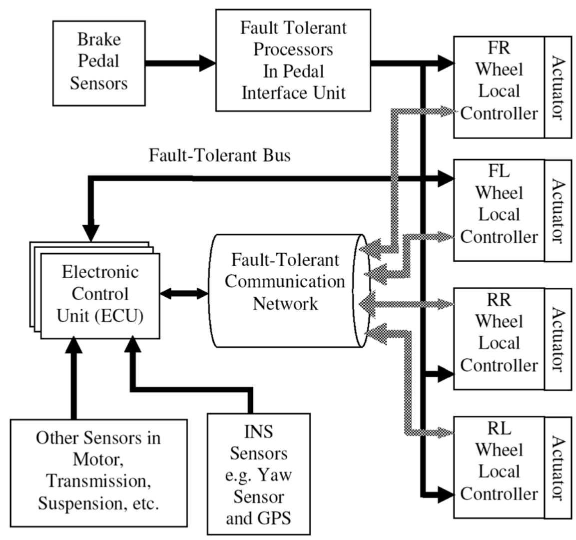


Fig. 1. A schematic architecture of a brake-by-wire system.

A brake-by-wire system, by nature, is a safety critical system and therefore fault tolerance is a vitally important characteristic of this system. As a result, a brake-by-wire system is designed in such way that many of its essential information would be derived from a variety of sources (sensors) and be handled by more than the bare necessity hardware. Three main types of redundancy usually exist in a brake-by-wire system:

1) Redundant sensors in safety critical components such as the brake pedal in Fig. 1.
2) Redundant copies of some signals that are of particular safety importance such as displacement and force measurements of the brake pedal copied by multiple processors in the pedal interface unit in Fig. 1.
3) Redundant hardware to perform important processing tasks such as multiple processors for the electronic controller unit (ECU) in Fig. 1.

Reliability, fault tolerance and accuracy are the main targeted outcomes of the fusion techniques that should be developed especially for redundancy resolution inside a brake-by-wire system. In order to utilise the existing redundancy, voting algorithms need to be evaluated, modified and adopted to meet the stringent requirements of a brake-by-wire system.

Several well known voting algorithms have been widely used in fault tolerant systems such as avionics and railway systems [6–8, 13] and fault tolerant VLSI circuits [4–6, 8, 13]. The *n*-input majority voter [1] produces a correct result if at least $[(n+1)/2]$ voter inputs match each other. In cases of no majority, the voter generates an exception flag, which can be detected by the system supervisor to move the system toward a safe state. As an extended version of majority voter, plurality voter [2] implements "*m* out of *n*" voting, where *m* is less than a strict majority. Median voter is a mid-value selection algorithm. Assuming an odd number of redundant inputs, this algorithm successively eliminates pairs of outlying values until a single result remains. The weighted average voter [21], on the other hand, calculates the weighted mean of its redundant input values. Parhami [16] examined the performance of different voting techniques, in terms of their execution time, and proposed efficient implementations of a variety of algorithms.

There is no agreement checking in weighted average and median voters [15]. Hence, they are not appropriate for safety critical applications such as braking. In the case of lack of majority agreement, majority voters give no result in the output and instead a flag is set. In a brake-by-wire system, however, "no result" is not acceptable as the output of fusion. Instead, a status bit is generated for each sensor.[1] If the sensors do not agree, invalidity of the voter output will be deduced from the status bits. Another problem with a majority voter is its considerable output discontinuity in the event of long-time disagreements [14, 18]. Latif-Shabgahi and his colleagues tried to solve this problem by introducing a smoothing voter in which an agreement-checking threshold is adaptively set when the voter produces no result. While their proposed method results in a lower number of no result events in the output of the voter, such events are not completely eliminated.

As an alternative solution for the problem, we propose to use the mean of agreeing sensors as the output of a majority voter and use their median value if there is no agreement. In this method that we call hard voting, a status bit is set if the sensors agree, and reset if they don't. The main issue in this voting method is how to set the geometric distance threshold [18] value by which sensor agreement is checked. Due to sensor conversion errors, there is almost always a distance between two agreeing sensors of different types. Therefore, distance threshold should be large enough to prevent incorrect decisions about sensor agreements in the presence of sensor conversion errors. A large value for the distance threshold in the hard voting method will, however, give rise to late fault detection if the fault causes a gradual change in the sensory signal. Such faulty gradual changes in sensory signals usually happen because of drifts, short circuits,[2] and sensor noise that gradually increases with temperature.

Genetic algorithms have also been applied for voting [19]. This approach, however, is only efficient when used with off-line calculations and in particular, for cases when the population of redundant components is large.

In this paper we propose a new voting method, called soft voting (in contrast to its alternative, hard voting), using a fuzzy logic paradigm. By using fuzzy logic rule-base inference, a faulty sensor is gradually removed from the output of our proposed soft voter. Instead of status bits, a faultiness measure is defined for each sensor that gradually increases in the event of faults. Although fuzzy inference and fuzzy systems have been utilised for sensor fusion in drive-by-wire applications, they have been employed merely to generate control commands or signal estimates for control and estimation applications in drive-by-wire technology [3, 20].

The fuzzy voter introduced in this paper is novel in the sense that it actually realises an adaptive weighted averaging mechanism for voting in which the weights are intelligently determined by the fuzzy inference engine. This inference engine is designed in such a way that faulty sensors are automatically detected based on the geometric distance between their outputs and other sensory measurements. As such distances grow, the weights corresponding to faulty sensors gradually decrease toward zero. To our knowledge, fuzzy systems have not been applied for voting in such a scheme.

For voting applications in systems with redundant sensors (or information sources), our proposed soft voter has the following advantages compared to other existing methods: Firstly, it does not output "no result." Secondly, it is capable of early detection and rejection of faulty sensors. Thirdly, its noise tolerance is higher than existing methods (due to the automatic fault detection and noise rejection phenomenon realised by the fuzzy inference machine). In addition, the output of our proposed voter does not suddenly jump or fall in case of signal short-circuits, and finally its computational complexity is comparable with simple voting methods like majority voters (particularly for a small number of sensors). These advantages all together make the proposed voter significantly efficient for real-time voting in redundant multi-sensor systems. We emphasize that most of the many voting techniques in the current lit-

[1] Henceforward, by sensor, a source of information is intended. It can be a redundant sensor, a redundant signal or a redundant processor.

[2] The RC filters that are connected to the inputs of ADCs (analog to digital converters) cause a gradual change in sensory signals when a short circuit happens.

erature have been designed for voting on multiple decisions (equivalent to fusion in decision or symbol level) while the method proposed in this paper and the methods reviewed in this section are applicable to voting on redundant signals i.e., the cases involving signal-level fusion.

We introduce our soft voting method in Section 2. Implementation of a soft voter for fusion of the redundant information provided by three sensors of a brake pedal is presented in Section 3. Then comparative experimental results of hard and soft voting methods on real sensory data will also be given in this section. Among the voting methods reviewed in this paper, hard voting is the closest to the proposed fuzzy voter in a sense that it is also a weighted-averaging voter but the weights have binary values and jump to zero in the case of a faulty sensor. Our soft voter is capable of early detection of faulty sensors and makes the weights gradually decrease toward zero in case of such faults. Due to their similarity and their meaningful difference, the fuzzy soft voter and the hard voter have been compared in Section 3 as a fair comparison. Section 4 concludes this paper. Although our method has been implemented and experimented for fusion of redundant safety critical components in a brake-by-wire system, the general scheme of our proposed fuzzy voter, explained in Section 2, can be applied to fuse redundant information in any application with safety critical issues and fault tolerance requirements.

## 2. PROPOSED SOFT VOTING METHOD

The block diagram of the proposed fuzzy voter for fusion of redundant information is shown in Fig. 2. In this diagram, $n$ sources of information (redundant sensors, signals or hardware) are called $S_1, S_2, \ldots, S_n$. Initially, low-pass filtering (to reduce the noise) and missing data handling (by using a multi-step ahead predictive filter [10, 11]) are performed on the raw sensory data. Then the signals are converted to an internal representation, which is a common format for the multi-source information. This conversion is required because different types of information (e.g. position data in millimetres and force information in Newton) should be converted to their equivalent values in a common format (internal representation) so that they have the same physical dimension before being compared and fused by a voter. The converted signals denoted by $x_1, x_2, \ldots, x_n$ are processed by an agreement evaluation block, resulting in $n(n-1)/2$ metrics denoted by $\{\alpha_{i,j} \mid i = 1, \ldots, n-1;\ j = i+1, \ldots, n\}$. In this block, the agreement of each pair of signals is quantified by an Euclidian distance measure. For example the agreement of the two sensors $S_i$ and $S_j$ is evaluated by the following equation:

$$\alpha_{i,j} = |x_i - x_j| \tag{1}$$

where $x_i$ and $x_j$ are the converted signals corresponding with the sensors $S_i$ and $S_j$. In the final step, the sensory

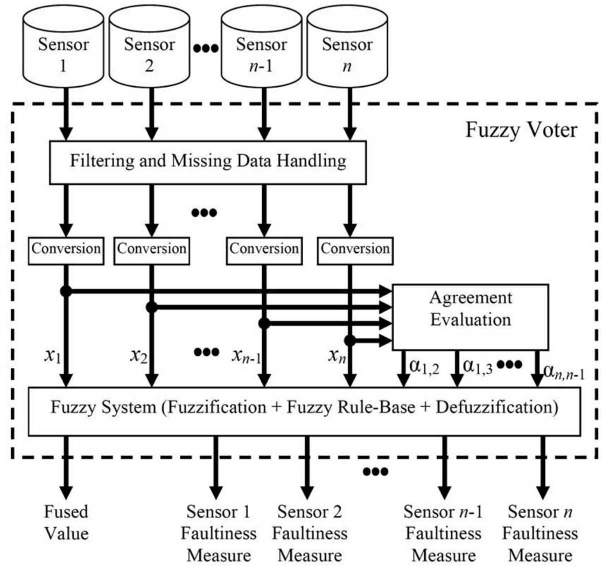


Fig. 2. Block diagram of the proposed fuzzy voter for fusion of redundant sensory information.

data $x_1, x_2, \ldots, x_n$ and their agreement evaluations $\{\alpha_{i,j}\}$ are passed on as inputs to a box that is responsible for fusion by voting. This box is a fuzzy system, comprising the common three subsystems i.e., fuzzification, a fuzzy rule-base and defuzzification. The fuzzy system has two outputs: a voted value as the main fusion output, and $n$ "faultiness measures" (instead of status bits) for the sensors. Each faultiness measure is a quantitative evaluation of voter's belief in the faultiness of a sensor in $[0, 1]$, with a value of 1 for total belief.

A hard voter outputs a fused value and $n$ status bits, showing the occurrence of faults in the sensors. More precisely, the hard voter does not need a fuzzy rule-base. Instead, its outputs are determined based on the results of comparing $\alpha_{i,j}$ values with an agreement threshold. For instance, in the case of $n = 3$ if $\alpha_{1,2}$ and $\alpha_{1,3}$ are higher than the threshold (i.e., $S_1$ and $S_2$ do not agree with each other; so do the pair of $S_1$ and $S_3$) and $\alpha_{2,3}$ is lower than the threshold (i.e., $S_2$ and $S_3$ agree with each other), then the hard voter will deduce that $S_1$ is faulty. In this case, the fused output will be the average of $S_2$ and $S_3$ and the faultiness status bits will be 100 for $S_1$, $S_2$ and $S_3$, respectively.

The agreement threshold is important in the voting process. It is tuned based on the $\alpha_{i,j}$ values in a normal working condition, when no sensor is faulty. They should be greater than the maximum $\alpha_{i,j}$ values in normal conditions, in such a way that conversion errors don't cause the voter to incorrectly assume that two sensors disagree. However, if a sensor gradually deviates from its true values because of sensor drifts or noise or short circuits, then the large thresholds cause a long delay in detection of the fault by a majority voter.

Our proposed soft voting method is mainly intended to solve the problem of late fault detection, and to prevent large discontinuities in the fusion output. Like any

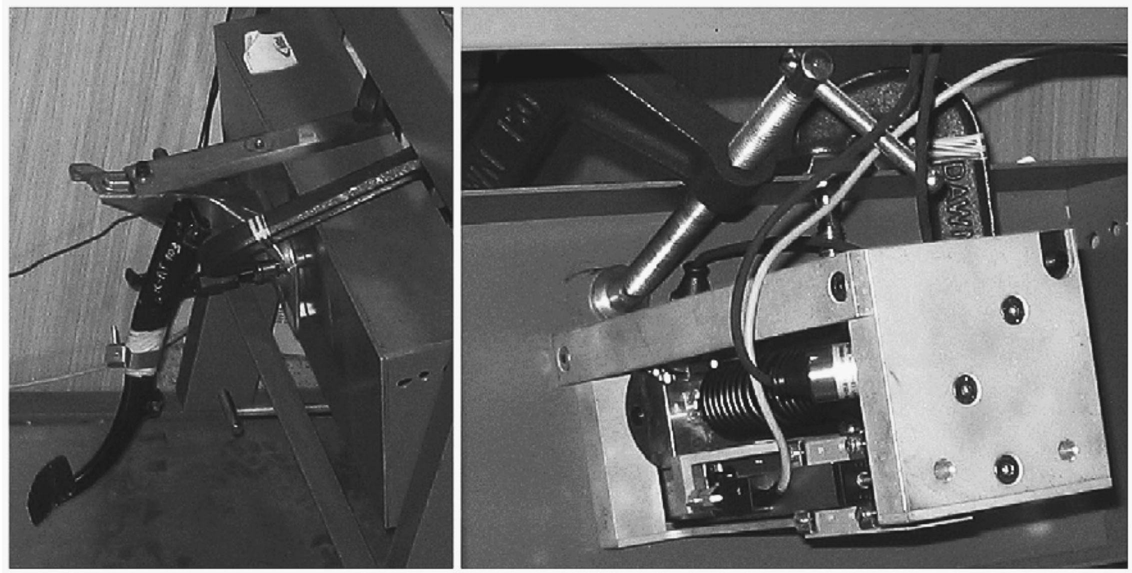

Fig. 3. Brake pedal and its sensors in our case study.

fuzzy system, $\alpha_{i,j}$ inputs are fuzzified first. We define three fuzzy sets of Large, Medium and Small agreements by their membership functions. These definitions are based on empirical maximum values of $\alpha_{i,j}$, derived from measurements and conversions. In practice, we collect some measurements from fine sensors and calculate the $\alpha_{i,j}$ values for each multi-sensory measurement. In case of triangular membership functions, if the maximum of $\alpha_{i,j}$ values is $\alpha_{\max}$, then breaking points of the Small fuzzy set are 0–$1.7\alpha_{\max}$, the breaking points of the Medium fuzzy set are $\alpha_{\max}$–$1.7\alpha_{\max}$–$2.3\alpha_{\max}$, and the breaking points of the Large fuzzy set are $1.7\alpha_{\max}$–$2.3\alpha_{\max}$. Generally, the application experts can determine the proper levels of $\alpha_{i,j}$ set as breaking points for Small, Medium and Large fuzzy sets. Based on the logic of majority voting, each fuzzy rule in the rule-base determines a voted output and $n$ faultiness measures. For example to vote three sensors, a typical fuzzy rule is expressed as follows:

IF
$S_1$ and $S_2$ agreement is Small
AND $S_2$ and $S_3$ agreement is Large
AND $S_3$ and $S_1$ agreement is Small
THEN
The fused output is the average of $S_2$ and $S_3$
AND $S_1$ faultiness is Large
AND $S_2$ faultiness is Small
AND $S_3$ faultiness is Small.

This rule explains what is logically expected as a voting result if $S_1$ does not agree with the other two sensors. The final defuzzified fusion output is calculated as a weighted average of all possible expected outputs by the following equation:

$$\text{Fused Output} = \sum_{i=1}^{M}(w_i O_i) \Bigg/ \sum_{i=1}^{M} w_i \tag{2}$$

where $M$ is the number of rules in the rule-base, $O_i$ is the fused output as it appears in the consequence of the $i$th fuzzy rule and the weight $w_i$ is the product of membership values of the conjoined parts of the antecedent of the rule. If the exemplar rule given above is the $k$th fuzzy rule in the rule-base, then $O_k = (x_2 + x_3)/2$ where $x_2$ and $x_3$ are the filtered sensory signals of $S_2$ and $S_3$ after conversion to the internal representation, as shown in Fig. 2. These weights smoothly change from 0 to 1 or reverse, and the fused output is smoothly switched from one vote to the other, hence the name soft voter. Sensor faultiness measures are defuzzified into crisp outputs by a fuzzy centroid method. In this method, a fuzzy number is transformed to crisp by taking the centre of gravity of its membership function. More precisely, if $Y$ is a fuzzy number with its membership functions determined as $\mu_Y(y)$, then the centroid crisp of $Y$ is given as below:

$$y = \int_{-\infty}^{+\infty} \alpha \mu_Y(\alpha) d\alpha.$$

## 3. EXPERIMENTAL RESULTS

We implemented our fuzzy voter to fuse the redundant information provided by three sensors mounted on a brake-by-wire pedal. Two sensors measure the force and the third sensor measures the pedal displacement. Although the sensors are different, they are redundant sources of information in the sense that they provide measurements for the same quantity: driver's brake demand. A photograph of the brake pedal and its sensors are displayed in Fig. 3.

As we have shown in the brake-by-wire diagram in Fig. 1, the displacement and force signals are preprocessed (low-pass filtering and missing data handling) by fault tolerant processors in the pedal interface unit and then transferred to four wheels via a fault tolerant communication bus (e.g. a LIN-bus). The processed sensory data are also sent to an electronic control unit (ECU) that includes a number of redundant processors generating the high level braking commands, such as anti-skid braking system (ABS), vehicle stability control (VSC) or traction control (TC).

In order to provide a reliable estimate for the driver's brake demand, pedal sensor data are voted in the ECU, where the resulting brake demand is then fused with the other vehicle sensor data (e.g. wheel speed or INS—Inertial Navigation System—sensors like accelerometers and gyros) to generate four final brake commands. To activate the brake actuators, these commands are sent to the local controllers in the four brake callipers via a fault tolerant time-triggered communication network. If for any reason the ECU is faulty then pedal sensory data will be voted in the local controller of each wheel unit, leading to generation of a brake response on each wheel. The main purpose of voting is to detect sensor faults (such as excessive noise, short circuits or sensor drifts) and to remove the effects of faulty measurements from the brake demand. In the presence of a fault or a substantial level of noise in sensor signals, they will not agree with each other. A voter should detect these disagreements and use them to identify faulty sensors. A hard voter simply discards faulty sensor data and outputs the average of agreeing sensors.

Fig. 4 shows a block diagram of the pedal sensor fusion scheme which is the revised version of the diagram shown in Fig. 2, for our experiments. $S_1$ and $S_2$ are the two force sensors giving $f_1$ and $f_2$, and $S_3$ is the displacement sensor with its signal denoted by $x$. Force is the quantity selected as the internal representation for fusion of the three sensors. In other words, the pedal displacement signal is converted to equivalent force signals $\hat{f}_1$ and $\hat{f}_2$ to be compared with the signals provided by the other two sensors. In order to perform this conversion, a model is required to mathematically relate the three signals $x$, $f_1$ and $f_2$. The passive push-return mechanism of the pedal can be modelled with an ideal spring in parallel with a damper, as shown in Fig. 5. The two force sensors are located at the two ends of the paralleled spring and damper model. Since the acceleration of pedal movements is too small to be considered in the model, the effect of the pedal mass is neglected. Thus, the two force sensor measurements are very close and have been simply labelled with $f$ in Fig. 5 and the following equations. Based on the simplified damper-spring model, the following equation expresses the measured force signals in terms of the measured displacement signal:

$$f = kx + b\dot{x} \tag{3}$$

where $k$ and $b$ are the spring and damping factors, respectively.

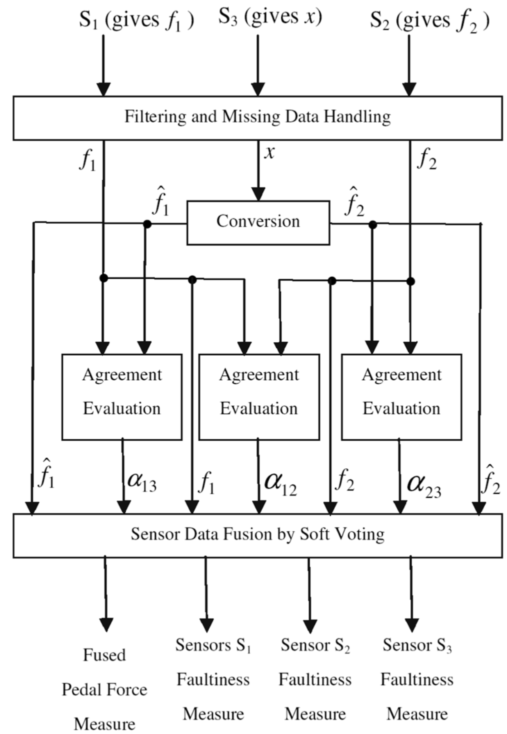


Fig. 4. Block diagram of pedal sensor fusion.

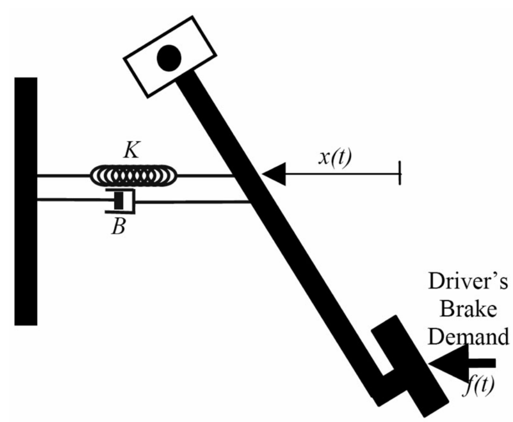


Fig. 5. A simplified model of the pedal and its sensors.

In order to validate the model and estimate its parameters, we ran a number of experiments and collected the three sensors measurements. In these experiments, the pedal set was installed in a car and a driver used it for different braking scenarios such as continuous soft brakes, frequent push-release and panic brakes. Using the collected sensory data, we examined the linearity between force, displacement and velocity using a least squares (LS) technique. More precisely, we utilised the recorded signals $f$, $x$ and $dx/dt$ and obtained a LS estimate of the parameters $k$ and $b$ in (3). This resulted in a low correlation coefficient and large difference between the measured forces $f$ and the force values $\hat{f} = kx + b\dot{x}$. These results showed a poor linear relationship between those quantities and a single linear model that would describe the repeated experiments could not be found.

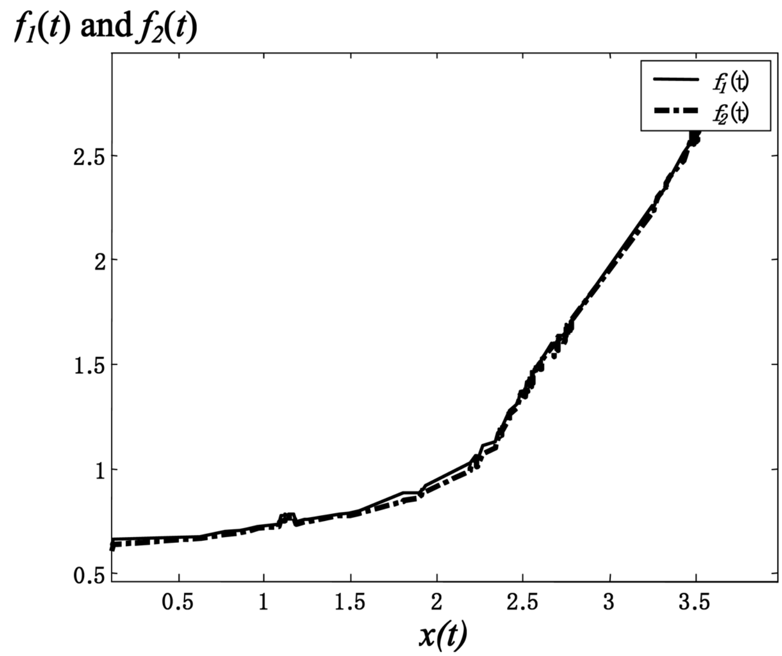


Fig. 6. Force signals versus displacement sensors at the time instants when the pedal is stationary.

Thus, a linear model for our spring and damper is not sufficient and their nonlinearity should also be taken into account. We examined a generalised version of the above linear model (3):

$$f = g_1(x) + g_2(\dot{x}). \tag{4}$$

In order to find the proper mathematical form of $g_1$, we examined the recorded force and displacement data for the stationary pedal, i.e., the data samples with almost zero velocity. Fig. 6 shows the force versus displacement plotted at the time instants when the pedal is stationary. The very close distance between the two static force signals confirms our assumption on negligibility of the spring and damper masses. Fig. 6 also shows that $g_1(x)$ can be properly modelled by a quadratic polynomial:

$$\hat{f}|_{dx/dt=0} = Ax^2 + Bx + C. \tag{5}$$

This model complies with the fact that the spring force substantially increases when it is compressed beyond a linear region. Using the recorded static data, we achieved a LS estimate for the parameters $A$, $B$ and $C$ in (5).

For the function $g_2$ in (4), another quadratic model was chosen and its parameters were also estimated by the LS technique. The viscous friction substantially increases when the pedal speed rises beyond the linear damper model, and this phenomenon is actually realised by the quadratic model for $g_2$. The models used for conversion of displacement measurements to equivalent force values are presented as follows:

$$\hat{f}_1 = A_1x^2 + B_1x + C_1 + D_1\dot{x}^2 + E_1\dot{x} \tag{6}$$

$$\hat{f}_2 = A_2x^2 + B_2x + C_2 + D_2\dot{x}^2 + E_2\dot{x}. \tag{7}$$

The LS estimates of $A_1$ and $A_2$ are very close to each other, and so are $B_1$ and $B_2$, $C_1$ and $C_2$, $D_1$ and $D_2$, and $E_1$ and $E_2$. This validates our assumption on negligibility of the effect of pedal mass and the sufficiency of a first order dynamic model. As shown in Fig. 4, after using the quadratic models, shown in (6)–(7), with their estimated parameters to convert the displacement sensor output to their equivalent force signals, the four signals $f_1$, $f_2$, $\hat{f}_1$ and $\hat{f}_2$ can now be utilised to evaluate the sensors agreement by calculating $\alpha_{1,2}$, $\alpha_{1,3}$ and $\alpha_{2,3}$ values. More precisely, the internal representation of signals in Fig. 4 is the "force" quantity and $f_1$ and $f_2$ are same as $x_1$ and $x_2$ in Fig. 2. Since the displacement measurement $x$ is converted to two estimates $\hat{f}_1$ and $\hat{f}_2$ (to be compared with $f_1$ and $f_2$), $x_3$ in Fig. 2 has two corresponding signals in Fig. 4: $\hat{f}_1$ and $\hat{f}_2$.

These values along with the forces and converted signals are then given to a fuzzy system where the agreement values are fuzzified. Fig. 7 shows the definitions of the fuzzy sets for fuzzification of agreement evaluations. Because of the conversion errors, $\alpha$-coordinates of the break-points of the piece-wise linear membership functions for $\alpha_{2,3}$ and $\alpha_{1,3}$ are higher than the $\alpha$-coordinates of the break-points for $\alpha_{1,2}$. Since a lower $\alpha_{i,j}$ value means stronger agreement between $S_i$ and $S_j$, the Large and Small fuzzy sets are associated with lower and higher $\alpha_{i,j}$ values, respectively. The resulting membership values are then used by a fuzzy rule-base for fuzzy inference. In our case study, the rule-base contains seven fuzzy rules as shown in Table I. The third fuzzy rule is the same rule stated before in Section 2. Based on the details given in Table I, the final fused value for the driver's brake demand is computed by (2)

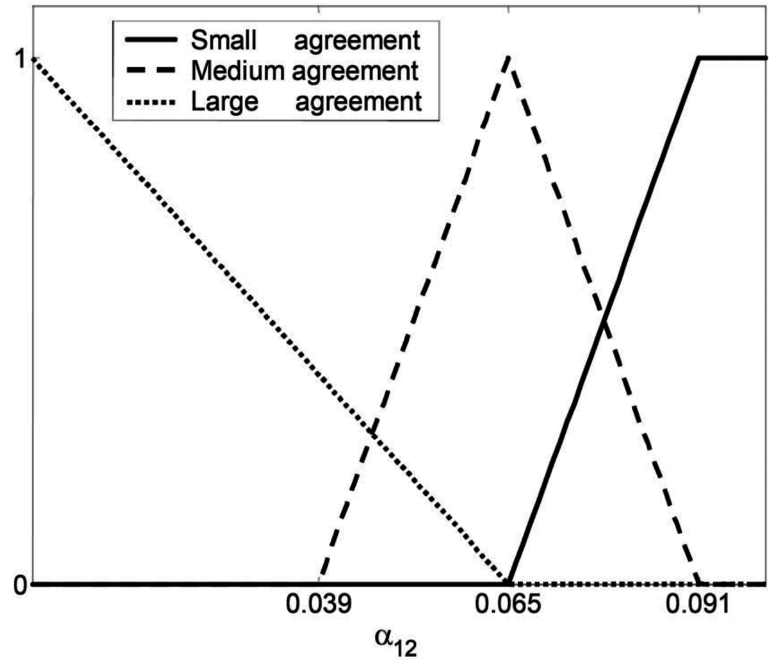


Fig. 7. Definition of three fuzzy sets for fuzzification of $S_1 - S_2$ agreement evaluation: Similar definitions apply to fuzzification of agreement evaluations of $S_1 - S_3$ and $S_2 - S_3$, however due to conversion errors the $\alpha$-coordinates of the break-points $\{0.039, 0.065, 0.091\}$ change to higher values of $\{0.52, 0.78, 1.04\}$.

TABLE I
The Fuzzy Rule-Base Utilised in for Sensor Fusion in our Experiments with the Brake-by-Wire Pedal
(L = Large, M = Medium, S = Small)

| $i$ | $\alpha_{12}$ | $\alpha_{23}$ | $\alpha_{13}$ | $O_i$ | $S_1$ Faultiness | $S_2$ Faultiness | $S_3$ Faultiness |
|---|---|---|---|---|---|---|---|
| 1 | L | L | L | $\frac{f_1 + \hat{f}_1 + f_2 + \hat{f}_2}{4}$ | Small | Small | Small |
| 2 | L | S | S | $\frac{f_1 + f_2}{2}$ | Small | Small | Large |
| 3 | S | L | S | $\frac{f_2 + \hat{f}_2}{2}$ | Large | Small | Small |
| 4 | S | S | L | $\frac{f_1 + \hat{f}_1}{2}$ | Small | Large | Small |
| 5 | L | M | M | $\frac{f_1 + f_2}{2}$ | Small | Small | Medium |
| 6 | M | L | M | $\frac{f_2 + \hat{f}_2}{2}$ | Medium | Small | Small |
| 7 | M | M | L | $\frac{f_1 + \hat{f}_1}{2}$ | Small | Medium | Small |

with $O_i$ and $w_i$ given as below:

$$
\begin{aligned}
w_1 &= \mu_L(\alpha_{12})\mu_L(\alpha_{23})\mu_L(\alpha_{13}), & O_1 &= (f_1 + \hat{f}_1 + f_2 + \hat{f}_2)/4 \\
w_2 &= \mu_L(\alpha_{12})\mu_S(\alpha_{23})\mu_S(\alpha_{13}), & O_2 &= (f_1 + f_2)/2 \\
w_3 &= \mu_S(\alpha_{12})\mu_L(\alpha_{23})\mu_S(\alpha_{13}), & O_3 &= (f_2 + \hat{f}_2)/2 \\
w_4 &= \mu_S(\alpha_{12})\mu_S(\alpha_{23})\mu_L(\alpha_{13}), & O_4 &= (f_1 + \hat{f}_1)/2 \\
w_5 &= \mu_L(\alpha_{12})\mu_M(\alpha_{23})\mu_M(\alpha_{13}), & O_5 &= (f_1 + f_2)/2 \\
w_6 &= \mu_M(\alpha_{12})\mu_L(\alpha_{23})\mu_M(\alpha_{13}), & O_6 &= (f_2 + \hat{f}_2)/2 \\
w_7 &= \mu_M(\alpha_{12})\mu_M(\alpha_{23})\mu_L(\alpha_{13}), & O_7 &= (f_1 + \hat{f}_1)/2.
\end{aligned}
\tag{8}
$$

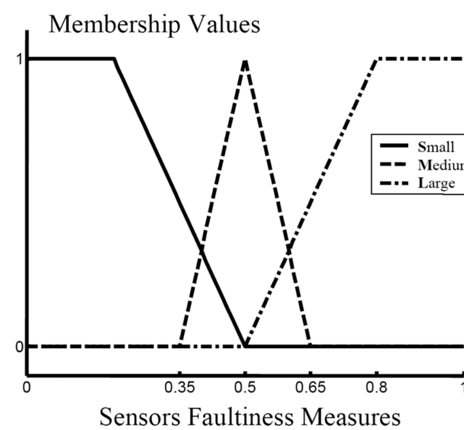


Fig. 8. Fuzzy sets definition for defuzzification of sensor faultiness measures.

In the consequences of the rules, the faultiness measures belong to one of the Small, Medium or Large fuzzy sets with piece-wise linear membership functions as shown in Fig. 8. The resulting faultiness measures are defuzzified by the fuzzy centroid method.

In our validation experiments, we applied different types of brake commands in various conditions such as a continuous panic brake, short-time panic brakes, short-time soft brakes, a continuous soft brake and so on. Total length of each experiment was 110 s. Fig. 9 shows the signals of the three sensors recorded during the validation experiments. $S_1$ and $S_2$ signals (pedal force measurements) are very close to each other and one of them is shown in Fig. 9. In this figure and the next signal plots, the vertical coordinate units are "volt," as the filtered "electrical" measurement signals and their fused measures have been plotted and all

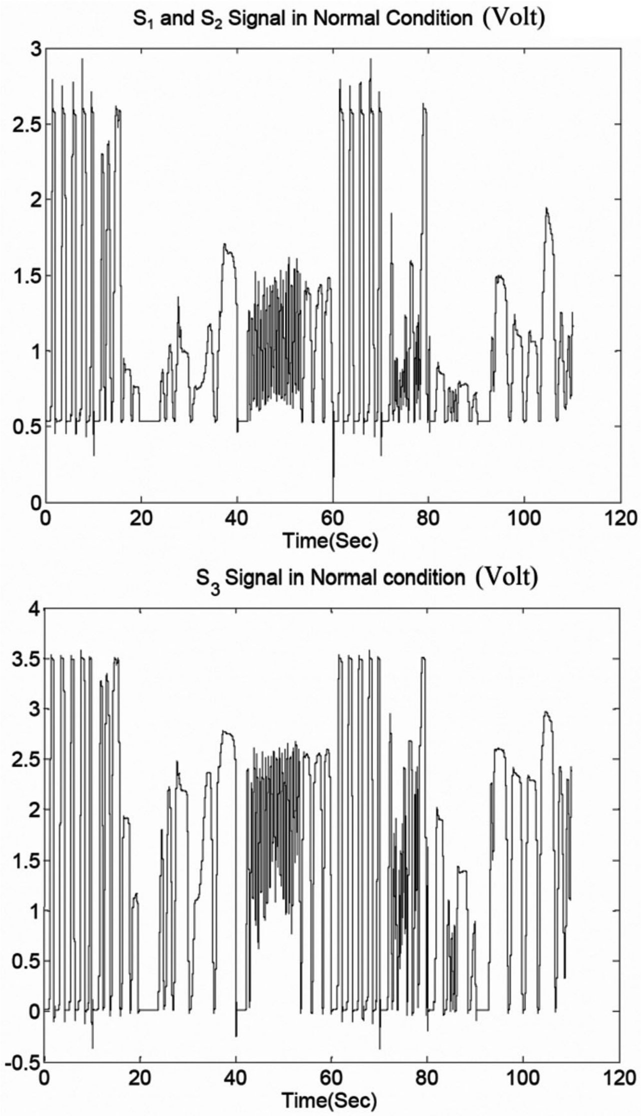


Fig. 9. Recorded sensory signals in normal (no fault) condition.

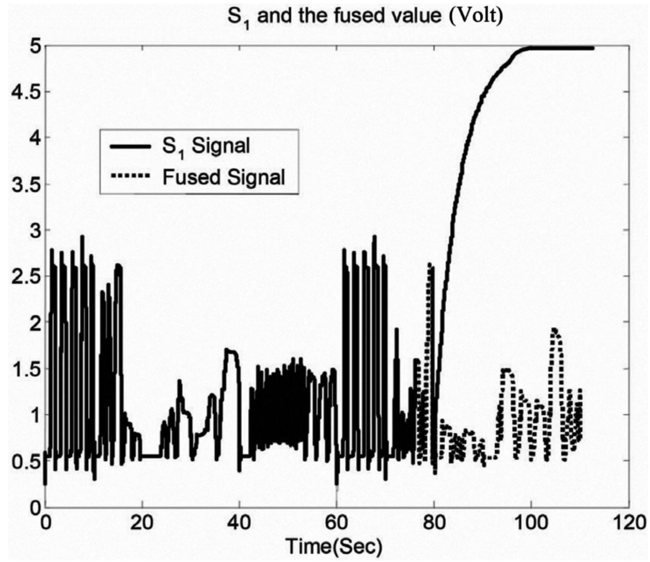


Fig. 10. Soft voting result when $S_1$ is short circuit and gradually rises toward supply voltage.

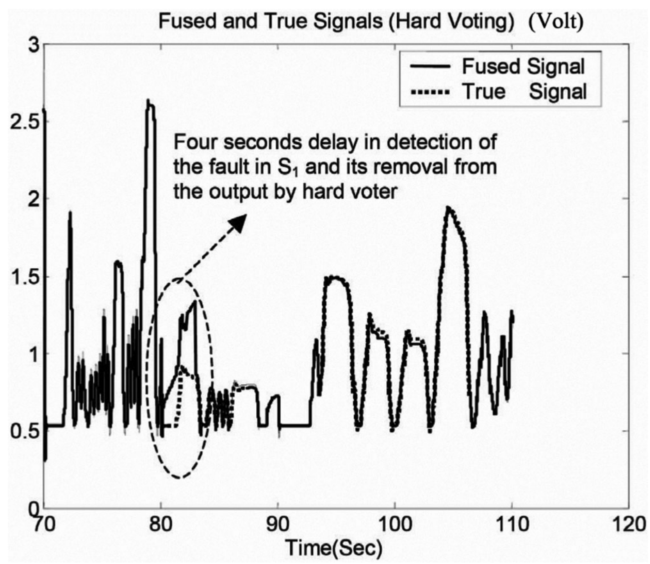


Fig. 11. Hard voting result when $S_1$ is short circuit and gradually moves toward supply voltage.

of them are proportional to the internal representation quantity (force) with a constant factor. We then injected several types of synthetic faults into $S_1$ during the time interval [80, 110] and used both the hard and the soft (fuzzy) voting methods to fuse the sensor data. Fig. 10 shows the results when the $S_1$ signal is short-circuited to supply. Because of the RC circuitry connected to the input of analogue to digital converters (ADCs) the $S_1$ signal does not suddenly jump to the supply voltage, but rises gradually. Soft voting detects the fault and removes the $S_1$ signal from voting process in a timely manner. We also applied hard voting to detect the same fault. Fig. 11 shows the fused signal and its expected true values in the time interval, starting 10 s before the short circuit event. It is observed that the short circuit is detected by hard voting after four seconds as the short circuit starts at $t = 80$ s but the deviation of the fused signal from the true signal returns to almost zero at $t = 84$ s. During these four seconds the hard voter provides a wrong fused measurement. This is fairly dangerous and unacceptable in a brake-by-wire application.

Pedal sensors data may also drift due to temperature variations during motor warm-up or cool-down periods. Fig. 12 shows a linear drift of 1000 mV injected into $S_1$ and the result of soft voting by which the drift is detected and removed. On the other hand, the hard voting method does not detect the drift, because the threshold of agreement evaluation is larger than the 1000 mV drift. Hard voting result is presented in Fig. 13. Faultiness measures resulted from soft voting in the presence of the linear drift in $S_1$ are also shown in Fig. 14. It is observed that faultiness for $S_1$ is always large and faultiness for $S_2$ and $S_3$ are initially large but decrease while the drift in $S_1$ grows. To examine the performance of the proposed technique for a noisy signal, excessive noise

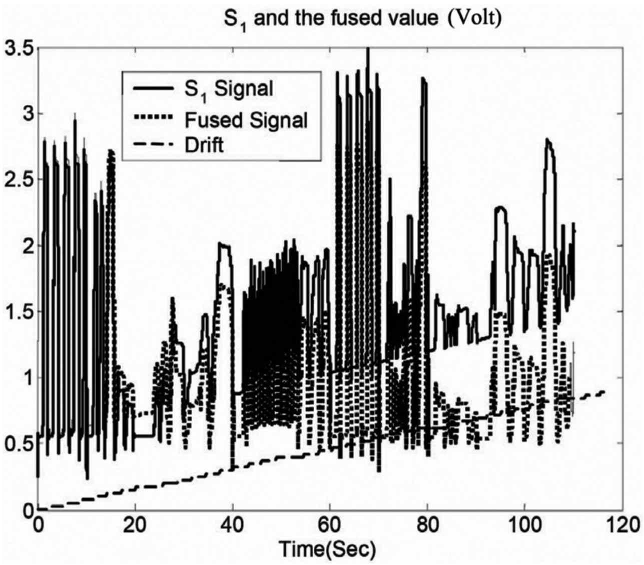


Fig. 12. Soft voting result when there is a linear drift in $S_1$.

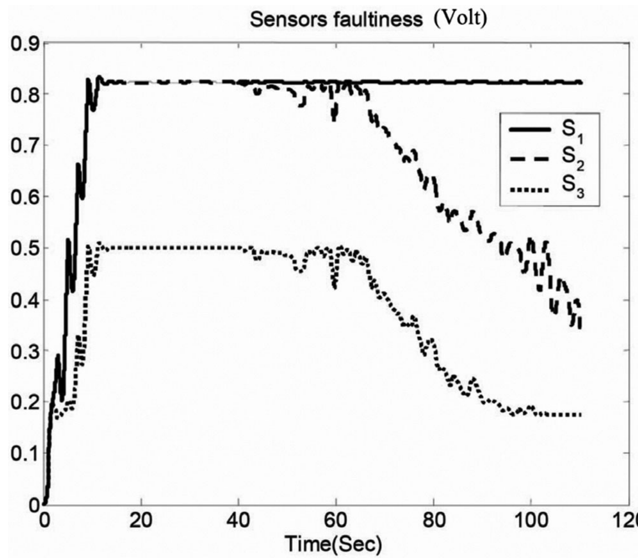


Fig. 14. Faultiness measures resulted by soft voting result in presence of a linear drift in $S_1$.

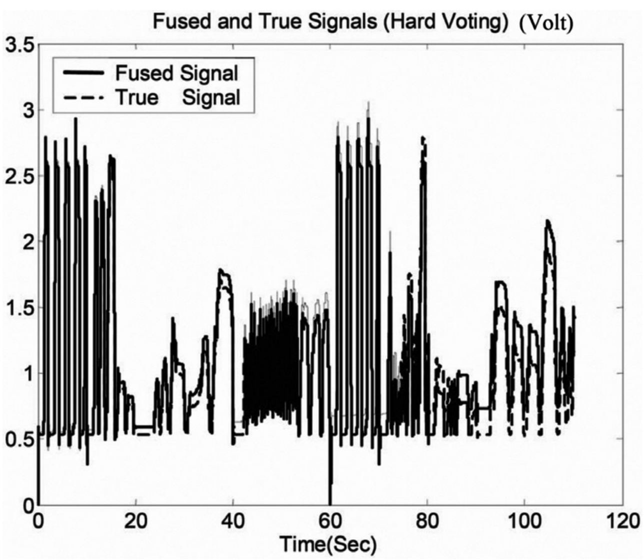


Fig. 13. Hard voting result when there is a linear drift in $S_1$.

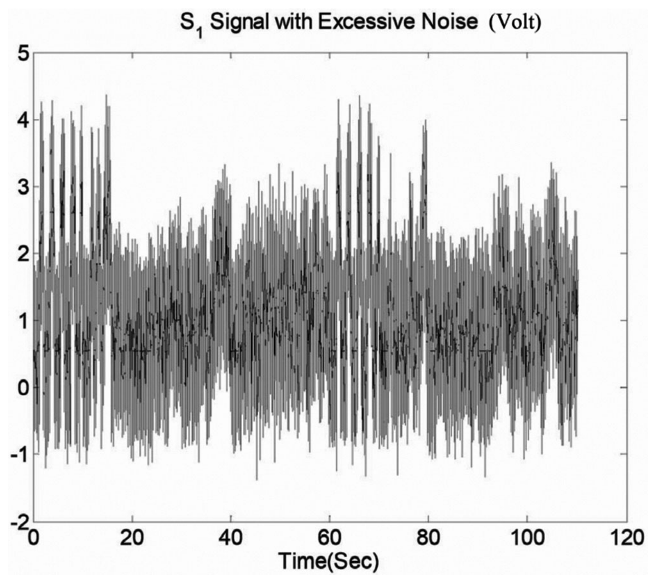


Fig. 15. $S_1$ signal in presence of excessive noise.

was injected into the $S_1$ signal as depicted in Fig. 15. As shown in Fig. 16, soft voting has been able to effectively detect and remove the noise from sensor fusion output, and Fig. 17 shows that hard voting can not substantially reduce the noise.

In order to compare the performance of the majority (hard) voting method with our proposed soft voting method quantitatively, we computed the mean square error (MSE) for soft and hard voting methods in the presence of various faults. Table II shows the result of our error computation. Overall, the MSE was reduced by 82% in soft voting compared to hard voting. That is because of the early fault detection and removal capability of the soft voter. Finally, it should be noted that our proposed method is a voting method, i.e., we do not expect it to detect a fault if it exists in the majority of sensors (two or more sensors in our case study). For example if a short circuit happens for both $S_1$ and $S_2$, then both the hard and the soft voter will incorrectly deduce that $S_3$ is faulty because it does not agree with the other two sensors.

## 4. CONCLUSIONS

In this paper, we introduced a new method for fusion of redundant sensory information in fault tolerant systems with focus on a by-wire braking system. We applied our method to fuse the redundant data provided by two force sensors and one displacement sensor in a by-wire brake pedal. Because of the sensor conversion errors, sensor agreement thresholds in a majority voter are so large that an unacceptable delay in fault detection occurs. Our proposed soft voting method applies a fuzzy

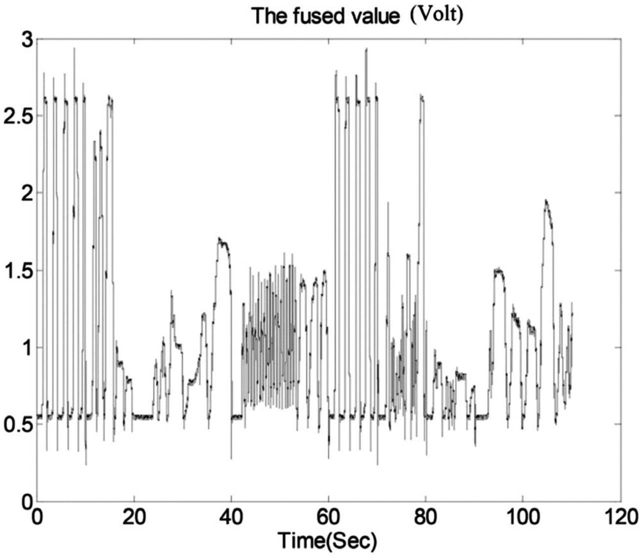


Fig. 16. Soft voting result when in presence of excessive noise in $S_1$.

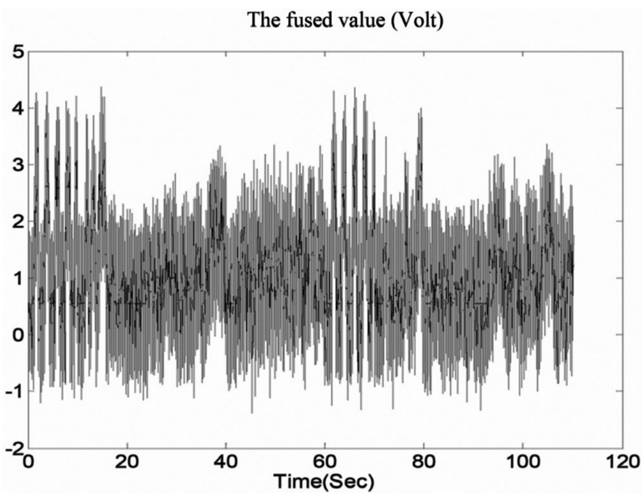


Fig. 17. Hard voting result when in presence of excessive noise in $S_1$.

rule-base to perform voting. The fuzzy rules here are designed in such a way that the voter output is smoothly switched from one majority voted value to another in case of a sensor fault. The proposed soft voter also gives faultiness measures for all sensors.

The novel idea in our approach is that we calculate the averaging weights as a normalised sum of products of membership values. The implementation of the proposed technique is straightforward and its execution is time efficient. As such, it is an appropriate solution for real-time and safety critical applications such as brake-by-wire, where computational load and memory requirements as well as convergence and stability are important issues. Experimental results show that our proposed method is successful in fault detection for cases where a majority voting approach either results in late detection or fails completely. Experiments also show that the soft voting total error (in terms of MSE) is reduced by around 82% compared to a hard voting technique.

TABLE II
MSE Error for Pedal Sensor Fusion by Soft and Hard Voting in Presence of Various Faults

| Injected Fault | Hard Voter | Soft Voter |
|---|---|---|
| Gradually Short to Ground | 0.1932 | 0.0367 |
| Gradually Short to Supply | 0.1033 | 0.0272 |
| Suddenly Short to Ground | 0.2123 | 0.0298 |
| Suddenly Short to Supply | 0.2099 | 0.0245 |
| Noise (Substantial SNR) | 0.1277 | 0.0434 |
| Drift | 0.2108 | 0.0269 |
| Total MSE | 1.0572 | 0.1885 |

REFERENCES


[1] A. Avizienis
The N-version approach to fault-tolerant software.
*IEEE Transactions on Software Engineering*, **1** (1985), 1491–1501.

[2] J. M. Bass, G. R. Latif-Shabgahi and S. Bennett
Experimental comparison of voting algorithms in cases of disagreement.
In *Proceedings of 23rd Euromicro Conference*, Budapest, Hungary, 1997, 516–523.

[3] D. Baum, C. D. Hamann and E. Schubert
High performance ACC system based on sensor fusion with distance sensor, image processing unit, and navigation system.
*Vehicle System Dynamics*, **28**, 6 (1997), 327–338.

[4] N. E. Belabbes, A. J. Guterman, Y. Savaria and M. Dagenais
Ratioed voter circuit for testing and fault-tolerance in VLSI processing arrays.
*IEEE Transactions on Circuits and Systems I—Fundamental Theory and Applications*, **43**, 2 (1996), 143–152.

[5] A. Bogliolo, M. Favalli and M. Damiani
Enabling testability of fault-tolerant circuits by means of I-DDQ-checkable voters.
*IEEE Transactions on VLSI Systems*, **8**, 4 (2000), 415–419.

[6] S. Dajani-Brown, D. Cofer and A. Bouali
Formal verification of an avionics sensor voter using SCADE.
In *Proceedings of Lecture Notes in Computer Science*, 3253 (2004), 5–20.

[7] S. Dajani-Brown, D. Cofer, G. Hartmann and S. Pratt
Formal modeling and analysis of an avionics triplex sensor voter.
In *Proceedings of Lecture Notes in Computer Science*, 2648 (2003), 34–48.

[8] M. Favalli and C. Metra
TMR voting in the presence of crosstalk faults at the voter inputs.
*IEEE Transactions on Reliability*, **53**, 3 (2004), 342–348.

[9] T. A. Johansen, I. Petersen, J. Kalkkuhl and J. Ludemann
Gain-scheduled wheel slip control in automotive brake systems.
*IEEE Transactions on Control System Technology*, **11**, 6 (Nov. 2003), 799–811.

[10] R. Hoseinnezhad and A. Bab-Hadiashar
Missing data compensation for safety-critical components in a drive-by-wire system.
*IEEE Transactions on Vehicular Technology*, **54**, 4 (July 2005), 1304–1311.

[11] R. Hoseinnezhad, A. Bab-Hadiashar and P. Harding
Missing data handling by a multi-step ahead predictive filter.
In *Proceedings of International Conference on Computational Intelligence for Modelling Control and Automation* CIMCA'2004, Gold Coast, Australia, 2004, 991–999.

[12] R. Hoseinnezhad
Position sensing in brake-by-wire callipers using resolvers.
*IEEE Transactions on Vehicular Technology*, **55**, 3 (May 2006), 924–932.

[13] H. Kim, H. J. Jeon, H. Lee and H. Lee
The design and evaluation of all voting triple modular redundancy system.
In *Proceedings of Annual Reliability and Maintainability Symposium*, Seattle, WA, 2002, 439–444.

[14] G. R. Latif-Shabgahi
A novel algorithm for weighted average voting used in fault tolerant computing systems.
*Microprocessors and Microsystems*, **28**, 7 (2004), 357–361.

[15] G. R. Latif-Shabgahi, S. Bennett and J. M. Bass
Smoothing voter: A novel voting algorithm for handling multiple errors in fault-tolerant control systems.
*Microprocessors and Microsystems*, 27 (2003), 303–313.

[16] B. Parhami
Voting algorithms.
*IEEE Transaction on Reliability*, **43**, 4 (1994), 617–629.

[17] K. Park and S. J. Heo
A study on the brake-by-wire system using hardware-in-the-loop simulation.
*International Journal of Vehicle Design*, **36**, 1 (2004), 38–49.

[18] J. M. Quintana, M. J. Avedillo and J. L. Huertas
Efficient realization of a threshold voter for self-purging redundancy.
*Journal of Electronic Testing—Theory and Applications*, **17**, 1 (2001), 69–73.

[19] F. Rothlauf
Population sizing for the redundant trivial voting mapping.
In *Proceedings of Lecture Notes in Computer Science*, 2724 (2003), 1307–1319.

[20] F. Tahami, R. Kazemi and S. Farhangi
A novel driver assist stability system for all-wheel-drive electric vehicles.
*IEEE Transactions on Vehicular Technology*, **52**, 3 (May 2003), 683–692.

[21] Z. Tong and R. Kain
Vote assignments in weighted voting mechanisms.
In *Proceedings of the Seventh Symposium on Reliable Distributed Systems*, Columbus, OH, 1988, 138–143.

**Reza Hoseinnezhad** was born in Tehran, Iran, in 1973. He received his B.E., M.E., and Ph.D. degrees from the University of Tehran, Iran, in 1994, 1996, and 2002, all in electrical engineering. From 2002 to 2003, he was an assistant professor at the University of Tehran. From July 2003 to October 2005, he was a postdoctoral research fellow, and since October 2005 he has been a senior research fellow, at Swinburne University of Technology, Victoria, Australia.

Dr. Hoseinnezhad's research is focused on new methodologies for signal processing and sensor data fusion techniques applied in drive-by-wire systems, and on the analysis of robust estimation techniques in computer vision. He has two international patents on brake-by-wire systems.

**Alireza Bab-Hadiashar** was born in Iran, in 1964. He received his B.E., M.E., and Ph.D. degrees in 1988 from the University of Tehran, Iran, in 1994 from the University of Sydney, Australia, and in 1997 from the Monash University, Australia, respectively. Since 1997, he has held various academic positions at both Monash University and the Swinburne University of Technology (Australia) where he is currently an associate professor and program manager of robotics and mechatronics. His research interest is the development of robust data analysis techniques for engineering applications in general and computer vision in particular.