

Algorithm for Crash Judgment by Frequency Analysis

Kazuo MATSUDA* Tsutomu FUKUI* Yukio HIRUTA*

ABSTRACT

A crash judgment algorithm that uses frequency analysis has been developed. This algorithm judges whether or not a crash has occurred by calculating both energy elements and frequency elements. Frequency analysis is implemented in hardware at the current microprocessor level using a Hadamard transform that does not require multiplication or division in the calculation of frequency elements. This approach improves the accuracy of crash judgments and reduces development time.

1. Introduction

Heightened awareness in recent years of the importance of passive safety has led to the inclusion of crash tests in domestic safety standards. Overseas, no longer are crash tests limited to the conventional full frontal barrier tests, but offset crashes, as typified by Offset Deformable Barrier (ODB) tests are also being implemented. Fig.1 gives examples of the most common crash tests and the corresponding

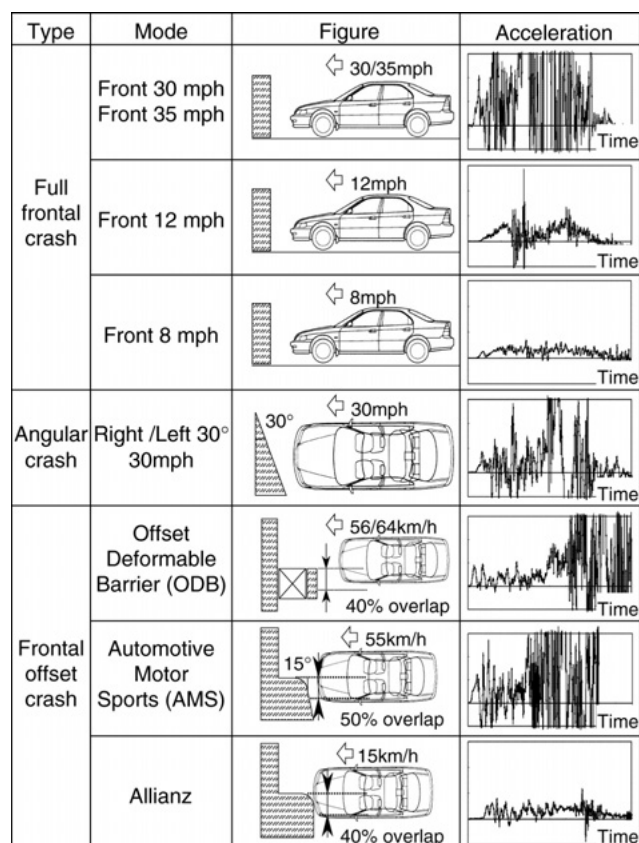


Fig.1 Common crash modes and acceleration wave forms

acceleration waveforms.

There are two important requirements for protecting occupants in the event of a crash including the following:

- The body crash performance needs to absorb the energy of the crash;
- The body crash performance needs to maintain survival space for occupants; and
- The occupant restraint system performance needs to minimize injury by helping prevent secondary impacts between occupants and interior components.

Crash judgment algorithms control the operation timing of certain components of such occupant restraint systems, namely, airbags and seatbelt pretensioners, including the turning on and off of these components.

The authors have developed a new crash judgment algorithm that incorporates a new frequency analysis technique. This paper discusses this algorithm and how it makes crash judgment easier than before.

2. Crash Judgment Algorithm

2.1. Overview of Electronic SRS

Early SRS placed a mechanical sensor in the vehicle front and a separate mechanical sensor in a unit located in the

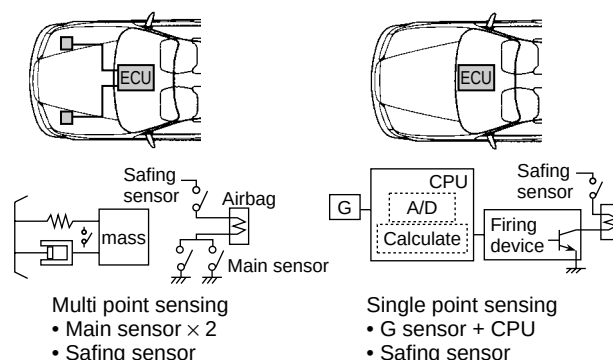


Fig.2 Mechanical/Electrical SRS

* Tochigi R&D Center

vehicle center (Fig.2, left side). With these sensors serving as electrical contacts, the system was designed so that when both sensors turned on simultaneously, electricity would be delivered to the airbag inflators, inflating the airbags with gas. This type of system is called a mechanical multipoint sensing system.

In the summer of 1993, a different type of system was released: electronic single point sensing (Fig.2, right side). In this system, the output from an acceleration sensor inside the unit (again located in the vehicle center) is processed by a CPU. When the results of CPU calculation indicate that a crash has occurred, the unit turns on the firing circuit.

It is the crash judgment algorithm that processes acceleration sensor output and determines whether a crash has occurred. Because acceleration differs from one car model to another, the threshold value used to judge whether a crash has occurred also differs from one car model to another. "Setting" is the process of determining the right threshold value for a given car model.

2.2. The Difficulty of Crash Judgment

There are two major difficulties in crash judgment:

- There are ON requirements and OFF requirements, each of which demands high reliability.
- ON requirements include a time requirement called TTF, discussed below.

An airbag must not deploy during normal driving (even when driving on rough roads) or during light crashes at speeds of approximately 8 mph BEV in which occupant injury would be minimal even without airbag deployment. Here BEV means Barrier Equivalent Velocity. On the other hand, an airbag can not fail to deploy in a crash in which deployment is imperative. The performance targets of deployment ("ON") and non-deployment ("OFF") require a

high level of reliability.

The performance target of deployment uses a predetermined time duration from crash start until the deployment signal is sent to the airbags; this is called Time to Fire (TTF). Ordinarily, TTF is set so that an airbag deploys at the point where the occupant has been displaced 5 inches. Since an airbag usually requires 30 ms to deploy, TTF is equal to [time required for the occupant to be displaced 5 inches] - 30 ms. In the case of a full frontal high-speed crash, this value is between 8 and 15 ms, which is the time before acceleration rises to the high levels that characterize high-speed crashes. Fig.3 gives the acceleration, velocity, and displacement of a high-speed crash. As this example illustrates, in judging a high-speed crash, it is necessary to predict whether this will be a major crash requiring airbag deployment, i.e., whether high-speed acceleration will occur, before sufficient crash acceleration has been generated. This is the difficulty of crash judgment.

2.3. Overview of Conventional Crash Judgment Algorithms

In Fig.4, an example of a conventional crash judgment algorithm, calculations are based on the acceleration differential, i.e., the rate of change in acceleration. Normally, other data are also used in the calculations, such as the acceleration integral, which indicates change in velocity, and the double integral of acceleration, which indicates displacement. According to the technique employed in this example, table processing is used during acceleration waveform computations. This allows for calculations to assess only a predetermined range of data. In cases of such processing, information beyond this range is sometimes absent from the body of data, the result being that sufficient extraction of acceleration waveform features is not possible.

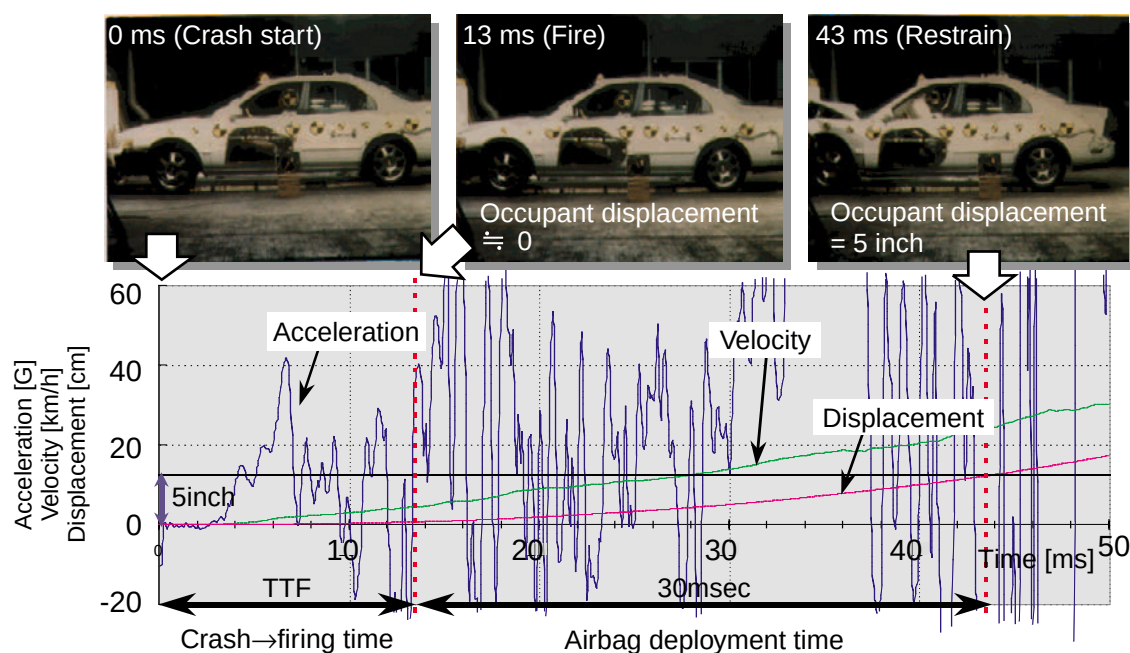


Fig.3 High speed crash

3. Overview of the New Crash Judgment Algorithm

3.1. The Validity of Crash Judgment by Frequency Analysis

The authors have employed frequency analysis to extract acceleration waveform characteristics. This method is already widely used for feature extraction for vibration, as in voice recognition and mode analysis. Fig.5 gives an example of the result of a type of analysis termed Short Time Fourier Transform (STFT)^{(1),(2)}, which yields a time variation showing fluctuation in the frequency element of the

acceleration waveform.

The results of this analysis show that in order to make a crash judgment based on frequency analysis, a mere frequency element is insufficient. Rather, as indicated in Fig.6, it is necessary to calculate the product of the frequency element data and the energy at that point. This can also be surmised from the mechanism in which, during a crash, energy accumulation (whereby stress first accumulates in a member) is followed by generation of high frequency elements, whereby stress is discharged through collapse. A simplified model of this process is shown in Fig.7.

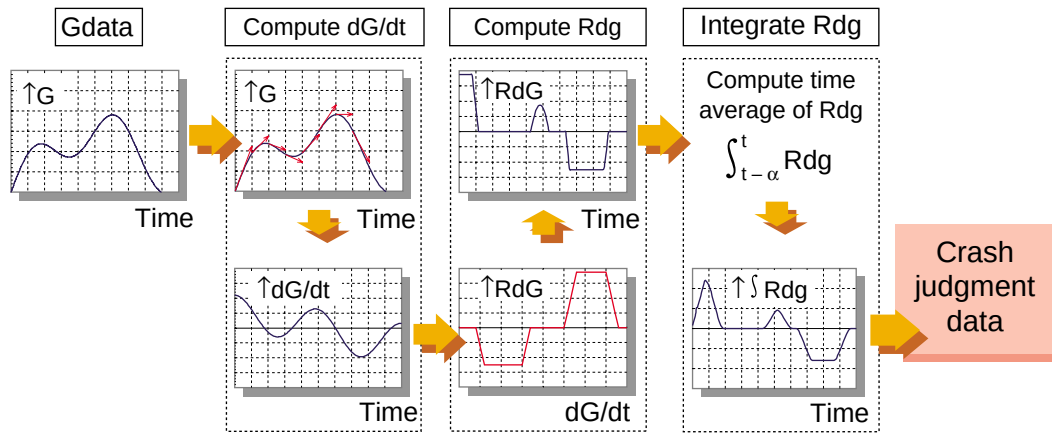


Fig.4 One of previous crash algorithm

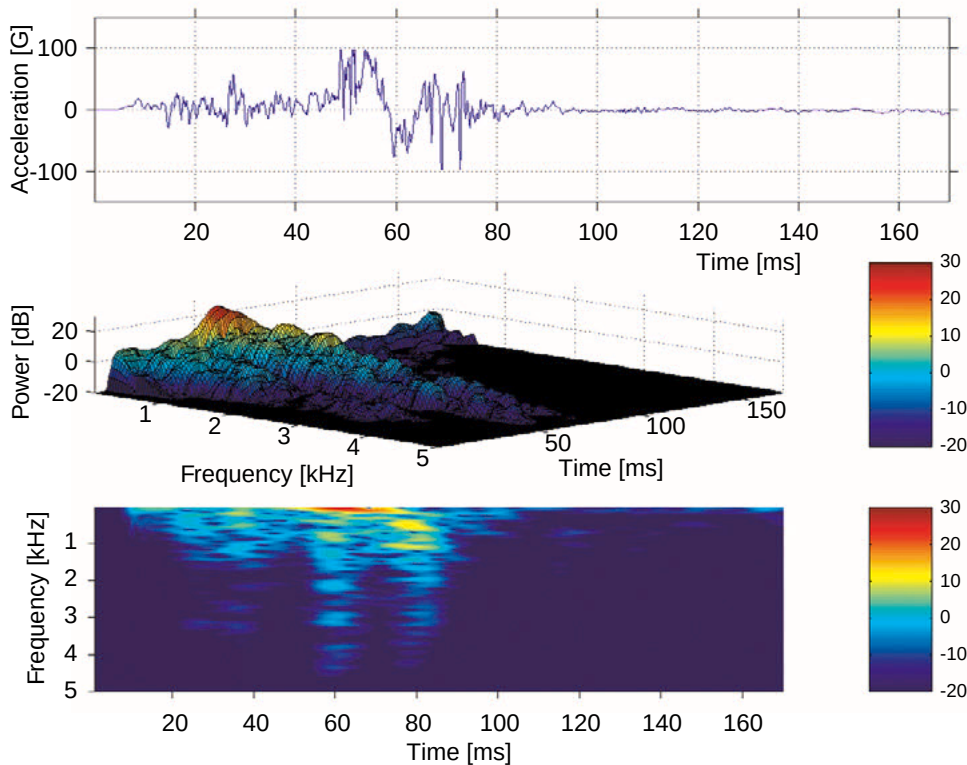


Fig.5 STFT analysis

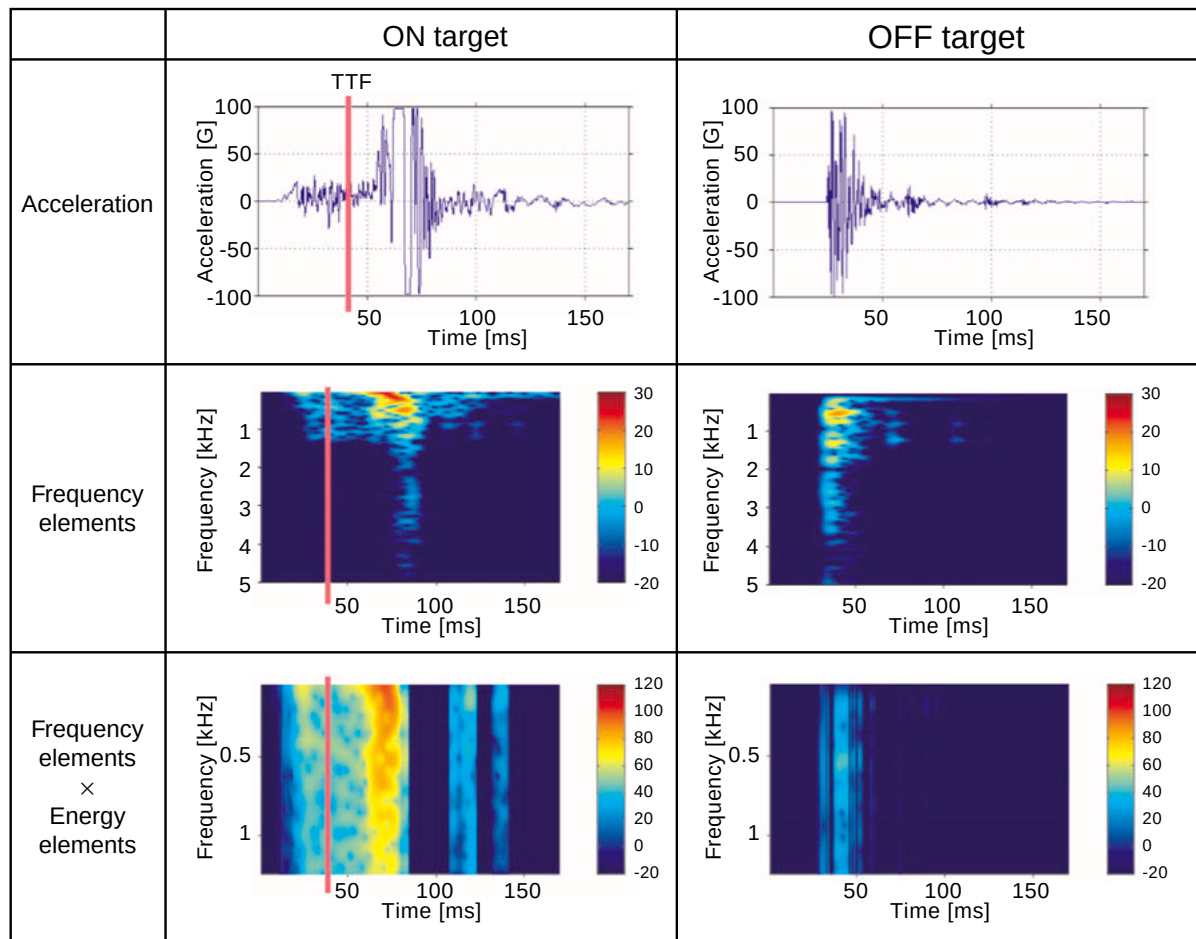


Fig.6 Frequency elements and Energy elements

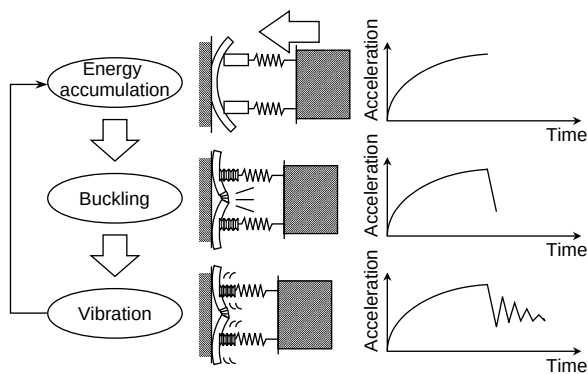


Fig.7 Model of generating a crash acceleration

3.2. Realization in ECU: Hadamard Transform

Fourier transform, used in the analysis thus far, provides a high accuracy of frequency decomposition, but is difficult to apply to current sensor ECU hardware because of the large volume of real-number multiplication and division operations. The authors have therefore selected Hadamard transform⁽³⁾, which although having a lower accuracy of

frequency decomposition, makes it possible to perform frequency decomposition with addition and subtraction alone. This reduces the volume of calculations to 1.9% in terms of time ratio.

Hadamard transform is a method of frequency analysis that is characterized by the following.

- Use of a matrix comprising only the (1, -1) transformation matrix (the only computations are addition and subtraction)
- The original waveform is expressed as an orthogonal, square wavegroup (Fourier transform in the broad sense)

This linear first order transformation can be expressed with the following equation.

$$\begin{pmatrix} Y_0 \\ Y_1 \\ \vdots \\ Y_{n-1} \end{pmatrix} = \begin{pmatrix} H \\ n \times n \end{pmatrix} \begin{pmatrix} X_0 \\ X_1 \\ \vdots \\ X_{n-1} \end{pmatrix}$$

Transformed data Transformation matrix Data before transformation

As shown below, a second-order basic matrix is used to generate the other transformation matrices recursively, thus: 2nd order $\rightarrow$ 4th order, 4th order $\rightarrow$ 8th order, ..., nth order $\rightarrow$ 2nth order.

$$\text{2nd order transformation matrix: } H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\text{4th order transformation matrix: } H_4 = \begin{pmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

$$\text{8th order transformation matrix: } H_8 = \begin{pmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}$$

$$\text{nth order transformation matrix: } H_n = \begin{pmatrix} H_{n/2} & H_{n/2} \\ H_{n/2} & -H_{n/2} \end{pmatrix}$$

Fig.8 gives the fundamental function for Hadamard transform and the resultant frequency characteristics for 16th order, 1 ms sampling. Although the effects of error due to square wave decomposition do exist, the basic frequency spectrum has been decomposed in keeping with theory. Here, SQ refers to sequency, an index in which the order of

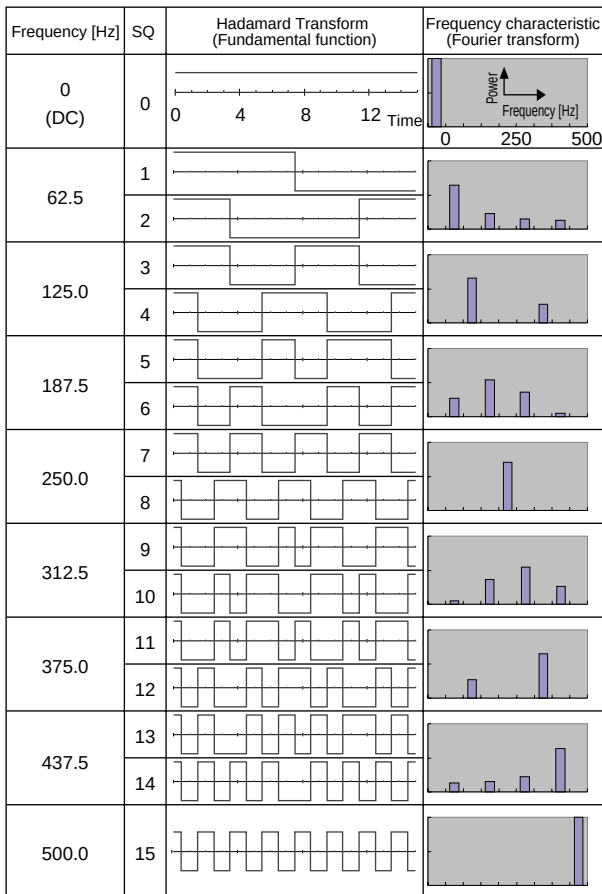


Fig.8 Frequency characteristic of Hadamard transform

Hadamard transform corresponds to frequency.

In the example application in an actual crash waveform in Fig.9, 16th-order Hadamard transform is used, generating 16 types of data for each frequency from one piece of acceleration data. The fact that the peak is at SQ4 indicates that that is where the correlation between the original acceleration waveform and the shape of the SQ4 fundamental function is the closest. In short, each of the 16 types of data that have been extracted from the original acceleration data for each frequency element represents the degree of correlation between that frequency and the fundamental function.

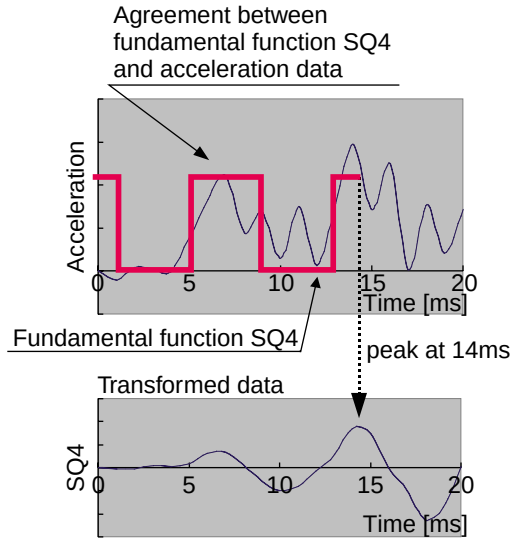
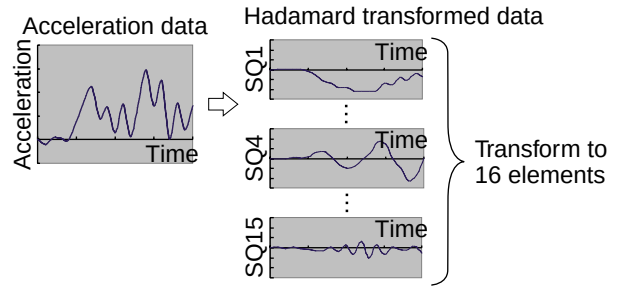


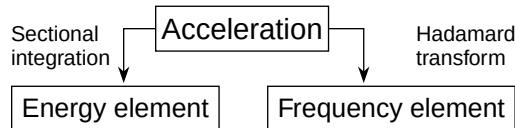
Fig.9 General idea of frequency analysis

3.3. Overview of the Algorithm

Using the results of analysis up to this point, the new algorithm decomposes the acceleration signal into energy elements and frequency elements. If specific energy elements and frequency elements (i.e., the Hadamard transform data) exceed the predetermined threshold value, it then judges that a crash has occurred.

The simplified block diagram and judgment flowchart in Fig.10 demonstrate how the threshold value for the frequency element is variable, changing in steps according to energy element output.

Block diagram



Judgment flow chart

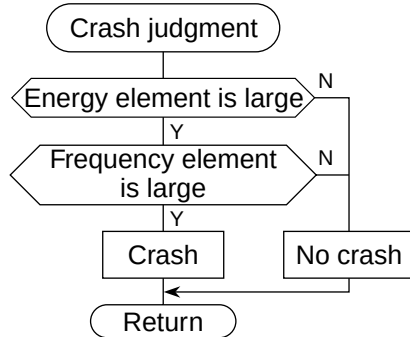


Fig.10 Concept of the crash algorithm

3.4. The Verification Process

Because this algorithm uses frequency elements as the basis of crash judgment, it is crucial to verify toughness (i.e., robustness) with respect to variation in frequency elements. This is because, in a real-world crash outside of the crash mode for which the algorithm was set, the frequency elements generated during the crash could vary and prevent the predetermined requirements from being met. Conventional algorithms, which base their calculations essentially on velocity change, have been empirically shown to maintain a relatively proportional relationship even in such situations. For this reason, the authors tested the suitability of this algorithm with respect to the following parameters.

- The effects of resonance
- Crash velocity and mode
- Frequency variation

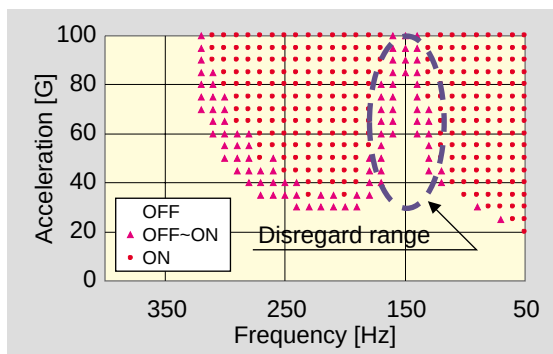
As for validation of robustness against resonance, the authors used crash waveforms from past measurements to verify that this algorithm permits easy setting even for crash waveforms where the resonance of the ECU mounting bracket was sufficiently large to warrant investigation. Fig.11 shows the ON characteristics by frequency for a conventional algorithm and the new algorithm. This shows how the new algorithm makes it possible to avoid bracket resonance by using a setting that disregards a specific frequency element, which would be difficult to distinguish with a conventional algorithm.

For validation of robustness with respect to crash velocity and mode, barrier testing simulating real-world crashes at non-regulated velocities was implemented. This was done with two crash modes, in which the suitability of airbag deployment was verified. The results are summarized in Table 1.

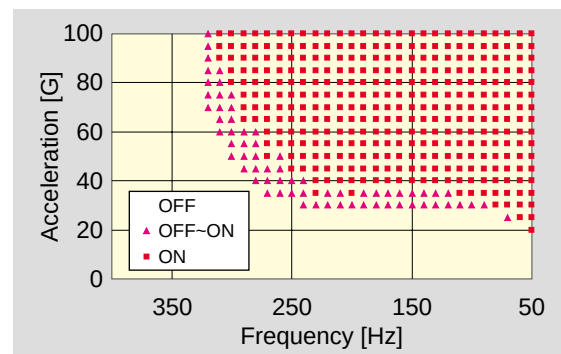
To verify the algorithm's robustness against the effects of frequency variation, the frequency elements of actual crash waveforms were varied using a random number generator program. Fig.12 shows the method of generation: The original acceleration waveform is decomposed into frequency elements, the amplitude of which is then changed with random numbers to produce variations that are inversely transformed to determine the waveform's usability under certain restraint conditions.

Table 1 Non-regulation crash modes

Mode	Figure	Target	Result	Judgement
Full frontal crash 24mph		15.8	14	○
Frontal offset pole crash 24mph		43.3	34	○



New algorithm



Previous algorithm

Fig.11 Disregarding a frequency range

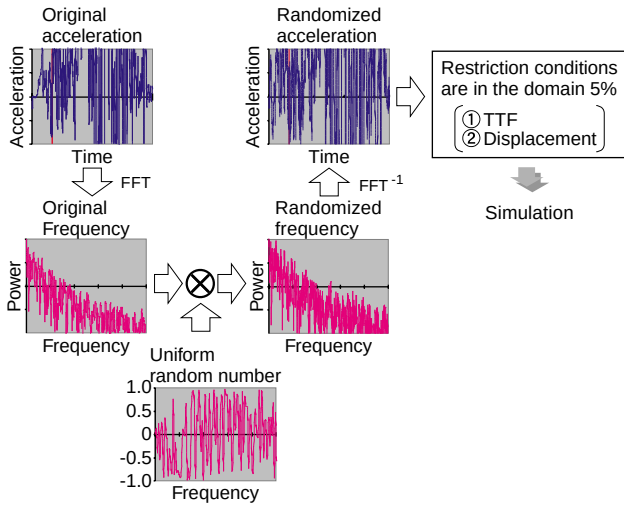


Fig.12 Changing frequency element

4. Validation of Effectiveness

Let us consider an example of the improved effectiveness of this algorithm in comparison with that of conventional algorithms.

In Fig.13, the feature with which the conventional algorithm meets only the OFF target are shown as “property 1.” Here, the issue is maintaining the ON target. Conventional algorithm property 2 is the opposite of 1 in that it meets only the ON target, the issue being the maintenance of the OFF target. Each line represents the velocity change of each crash waveform.

In Fig.14, the new algorithm was used for these waveforms. Note how the new algorithm easily meets both the ON and OFF targets.

In addition, the algorithm’s simplified structure has made it possible to create a tool for automatically searching for

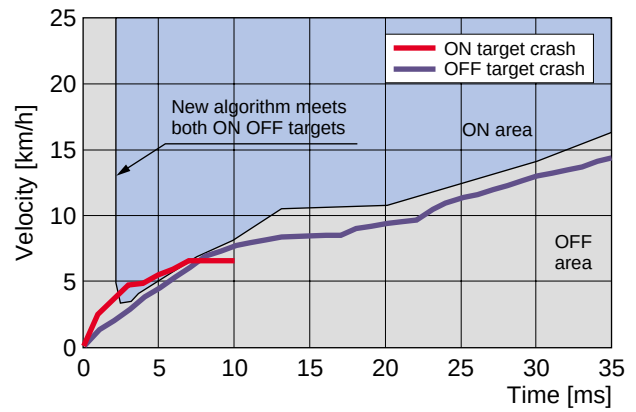


Fig.14 Characteristic of new algorithm

correlation between frequency elements and energy elements. This tool has greatly reduced the duration of setting by making it possible to approximate the setting semi-automatically.

5. Conclusion

By applying the Hadamard transform, the authors have created a crash judgment algorithm that employs frequency analysis at the level of microprocessors that are mountable in current sensor ECU hardware. Moreover, the validity of this algorithm was demonstrated.

This algorithm improves the accuracy of crash judgment and greatly reduces the amount of time required for setting.

The authors plan to make additional improvements to further enhance the performance and setting efficiency of the algorithm, thereby expanding the range of applicable vehicle models.

Acknowledgement

The authors wish to thank the staff of Keihin for their invaluable assistance with the development of this algorithm.

References

- (1) Keiichiro Minami, Satoshi Kawata: *Kagaku keisoku ni okeru data shori gihou*, Interface, Vol.20, No.1, p.110-119(1994) (Written in Japanese)
- (2) Charles K. Chui, Akira Sakurada, Tsutomu Arai: *Tanjikan fourier koukan to fukakuteiseikankei*, An Introduction to Wavelets, Tokyo Denki University Publishing, p.164-72(1993) (Written in Japanese)
- (3) Hiroshi Harashima: *Chokkouhenkan no tokuchou to sono hikaku*, Gazou Jyohou Assyuku, Ohmusha, Ltd. P.98-104(1991) (Written in Japanese)

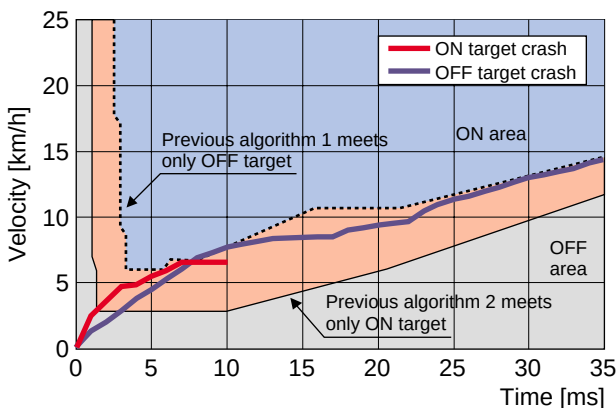


Fig.13 Characteristic of previous algorithm

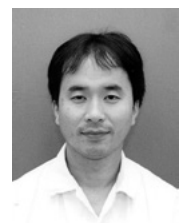
■ author ■



Kazuo MATSUDA



Tsutomu FUKUI



Yukio HIRUTA