A SMART AIRBAG SYSTEM

David S. Breed
Automotive Technologies International, Inc.
United States
Paper Number: 98-S5-O-13

ABSTRACT

Pattern recognition techniques, such as neural networks, have been applied to identify objects within the passenger compartment of the vehicle, such as a rear facing child seat or an out-of-position occupant, and to suppress the airbag when an occupant is more likely to be injured by the airbag than by the accident. Neural networks have also been applied to sense automobile crashes. The use of neural networks is extended here to tailoring the airbag inflation to the severity of the crash, the size, position and relative velocity of the occupant and other factors such as seatbelt usage, seat and seat back positions, vehicle velocity, and any other relevant information.

It is well known that a neural network based crash sensor can forecast, based on the first part of the crash pulse, that the crash will be of a severity which requires that an airbag be deployed. This is extended here to enhance the capabilities of this sensor to forecast the velocity change of the crash over the entire crash period. Then a pattern recognition occupant position and velocity determination sensor is added. Finally, an occupant weight sensor is included to permit a measure of the occupant's momentum or kinetic energy. The combination of these systems in various forms will be used to optimize inflation and/or deflation of the airbag to create a "smart airbag" system.

Crash sensors can predict that a crash is of a severity which requires the deployment of an airbag for the majority of real world crashes. A more difficult problem is to predict the crash velocity versus time function and then to adjust the airbag inflation/deflation over time so that just the proper amount of gas is in the airbag at all times even without considering the influence of the occupant. To also simultaneously consider the occupant size, weight, position and velocity renders this problem unsolvable by conventional methods.

BACKGROUND

Pattern recognition techniques, such as artificial neural networks, are finding increased application in solving a variety of problems such as optical character recognition, voice recognition, and military target identification. In the automotive industry, neural networks have now been applied to engine control and to identify various objects within the passenger compartment of the vehicle, such as a rear facing child seat. They have also been proposed for use with anticipatory sensing systems to identify threatening objects, such as an approaching vehicle about to impact the side of the vehicle. Neural networks have also been applied to sense automobile crashes for the purpose of determining whether or not to deploy an airbag or other passive restraint, or to tighten the seatbelts, cutoff the fuel system, or unlock the doors after the crash. Heretofore, neural networks have not been applied to forecast the severity of automobile crashes for the purpose of controlling the flow of gas into or out of an airbag in order to tailor the airbag inflation characteristics to the crash severity. Neural networks have also not been used to tailor the airbag inflation characteristics to the size, position or relative velocity of the occupant or other factors such as seatbelt usage, seat and seat back positions, headrest position, vehicle velocity, etc.

"Pattern recognition" as used herein means any system which processes a signal that is generated by an object, or is modified by interacting with an object, in order to determine which one of a set of classes the object belongs to. In this case, the object can be a vehicle with an accelerometer which generates a signal based on the deceleration of the vehicle. Such a system might determine only that the object is or is not a member of one specified class (e.g., airbag required crashes), or it might attempt to assign the object to one of a larger set of specified classes, or find that it is not a member of any of the classes in the set. One such class might consist of vehicles undergoing a crash of a certain severity into a pole. The signals processed are generally electrical signals coming from transducers which are sensitive to either acceleration, or acoustic or electromagnetic radiation and, if electromagnetic, they can be either visible light, infrared, ultraviolet or radar. The particular pattern recognition techniques used here are neural networks.

To "identify" as used herein means to determine that the object belongs to a particular set or class. The class may be one containing all frontal impact airbag-desired crashes into a pole at 20 mph, one containing all events where the airbag is not required, or one containing all events requiring a triggering of both stages of a dual stage gas generator with a 15 millisecond delay between the triggering of the first and second stages.

SINGLE POINT CRASH SENSORS

All electronic crash sensors currently used in sensing frontal impacts include accelerometers that detect and measure the vehicle accelerations during the crash. The accelerometer produces an analog signal proportional to the acceleration experienced by the accelerometer, and hence the vehicle on which it is mounted. An analog to digital converter (ADC) transforms this analog signal into a digital time series. Crash sensor designers study this digital acceleration data and derive therefrom computer algorithms that determine whether the acceleration data from a particular crash event warrants deployment of the airbag. This is usually a trial and error process wherein the crash sensor designer observes data from crashes where the airbag is desired and when it is not needed, and other events where the airbag is not needed. Finally, the crash sensor designer settles on the "rules" for controlling deployment of the airbag which are programmed into an algorithm which appears to satisfy the requirements of the crash library. The resulting algorithm is not universal and most such crash sensor designers will answer in the negative when asked whether their algorithm will work for all vehicles. Such an algorithm also merely determines that the airbag should or should not be triggered. Heretofore, no attempt has been made to ascertain or forecast the eventual severity of the crash or, more specifically, the velocity change versus time of the passenger compartment during the crash from the acceleration data obtained from the accelerometer.

Several papers have been published pointing out some of the problems and limitations of electronic crash sensors that are mounted out of the crush zone, usually in a protected location in the passenger compartment of the vehicle (1-6). The crush zone is defined, for the purposes herein, as that portion of the vehicle that has crushed at the time that the crash sensor must trigger deployment of the restraint system. These sensors are frequently called single point crash sensors.

These papers demonstrate, among other things, that there is no known theory which allows an engineer to develop an algorithm for sensing crashes and selectively deploying the airbag except when the sensor is located in the crush zone of the vehicle. These papers show that, in general, there is insufficient information within the acceleration signal measured in the passenger compartment to sense all crashes. Another conclusion suggested by these technical papers is that if an algorithm can be found which works for one vehicle, it will also work for all vehicles since it is possible to create any crash pulse measured in one vehicle, in any other vehicle. Note in particular SAE paper 920124 (3).

In spite of the problems associated with finding the optimum crash sensor algorithm, many vehicles on the road today have electronic single point crash sensors. Some of the problems associated with single point sensors have the result that an out-of-position occupant who is sufficiently close to the airbag at the time of deployment will be injured or killed by the deployment itself. Fortunately, systems are now being developed which monitor the location of occupants within the vehicle and can suppress deployment of the airbag if the occupant is more likely to be injured by the deployment than by the accident. At Present, these systems do not provide the information necessary for the control of airbag systems that have the capability of varying the flow of gas into or out of the airbag, and thus to tailor the airbag to the position, size and weight of the occupant. More particularly, no such system exists which uses pattern recognition techniques to match the airbag deployment or gas discharge from the airbag to the severity of the crash or the size, weight, position, velocity and seatbelt use of an occupant.

Since there is insufficient information in the acceleration data, as measured in the passenger compartment, to sense all crashes and since some of the failure modes of published single point sensor algorithms can be easily demonstrated using the techniques of crash and velocity scaling described in the referenced technical papers, and, moreover, since the process by which engineers develop crash sensor algorithms is based on trial and error, pattern recognition techniques such as neural networks, should be able to create an algorithm based on training the system on a large number of crash and non-crash events which, although not perfect, will be superior to all others. Such a crash sensor has been demonstrated which is based on the ability of neural networks to forecast, based on the first part of the crash pulse, that the crash will be of a severity requiring airbag deployment.

SMART AIRBAG CATEGORIES

An improvement to this neural network based crash sensor carries this process further by using the neural network to forecast the velocity change of the crash over time so that the inflation and/or deflation of the airbag can be optimized. Then by the addition of a neural network occupant position and velocity determination system as disclosed in (10,11) the occupancy category (forward facing human, rear facing child seat, box etc.), position and velocity can be obtained. Finally, the addition of the weight of the occupant provides a measure of the occupants kinetic energy as a further input to the system. The combination of these sub-systems in various forms can be called "smart airbags" or "smart restraints". In a preferred implementation, the crash severity is not explicitly forecasted. Rather, the value of a control parameter used to control the flows of inflator gas into or out of the airbag is instead forecasted.

Smart airbags can take several forms which can be roughly categorized into four evolutionary stages, which will hereinafter be referred to as Phase 1 (2,3,4) Smart Airbags, as follows:

1) Occupant sensors use various technologies to turn off the airbag where there is a rear facing child seat present or if either the driver or passenger is out-of-position to where he/she is more likely to be injured by the airbag than from the accident.
2) Occupant sensors are used along with variable inflation or deflation rate airbags to adjust the inflation/deflation rate to match the occupant, first as to his/her position and then to his/her morphology.

The neural network occupant sensors discussed in (10,11) will also handle this with the addition of an occupant weighing system. One particular weight measuring system, for example, makes use of strain gages mounted onto the seat supporting structure. At the end of this phase, little more can be done with occupant measurement or characterization systems.

3) The next improvement is to use a neural network as the basis of a crash sensor not only to determine if the airbag should be deployed, but also to predict the crash severity from the pattern of the initial portion of the crash pulse. Additionally, the crash pulse can continue to be monitored even after the decision has been made to deploy the airbag to see if the initial assumption of the crash type, based on the pattern up to the deployment decision, was correct. If the pattern changes indicating a different crash type, the flow rate to the airbag can be altered instantaneously.
4) Finally, anticipatory sensing using neural networks can be used to identify the crash before it takes place and select the deployment characteristics of the airbag to match the anticipated crash with the occupant size, position, velocity etc..

Any of these phases can also be combined with various methods of controlling the pretensioning, retraction or energy dissipation characteristics of the seatbelt. Although the main focus here is the control of the flows of gas into and out of the airbag, the control of the seatbelt can also be accomplished and the condition of the seatbelt can be valuable input information into the neural network system.

The smart airbag problem is complex and difficult to solve by ordinary mathematical methods. Looking first at the influence of the crash pulse, the variation of crash pulses in the real world is vast and quite different from the typical crashes run by the automobile industry as reported in the referenced technical papers. It is one problem to predict that a crash is of a severity level to require the deployment of an airbag. It is quite a different problem to predict exactly what the velocity versus time function will be and then to adjust the airbag inflation/deflation control system to make sure that just the proper amount of gas is in the airbag at all times, even without considering the influence of the occupant. To also simultaneously consider the influence of occupant size, weight, position and velocity renders this problem, for all practical purposes, unsolvable by conventional methods.

On the other hand, if a neural network is used and trained on a large variety of crash acceleration segments, and a setting for the inflation/deflation control system is specified for each segment, then the problem can be solved. Furthermore, inputs from the occupant position and occupant weight sensors can also be included. The result will be a training set for the neural network involving many millions, and perhaps tens of millions, of data sets or vectors as every combination of occupancy characteristics and acceleration segment is considered. Fortunately, the occupancy data can be acquired independently and is currently being done for solving the occupant position sensing problem of Phase 1 smart airbags. The crash data is available in abundance and more can be analytically created using the crash and velocity scaling techniques described in the referenced papers. The training using combinations of the two data sets, which must also take into account occupant motion that is not adequately represented in the occupancy data, can then be done by computer.

CRASH SEVERITY FORSCASTING

When a crash commences, the vehicle starts decelerating and an accelerometer located in the passenger compartment begins sensing this deceleration and produces an electronic signal that varies over time in proportion to the magnitude of the deceleration. This signal contains information as to the type of the crash that can be used to identify the crash. A crash into a pole gives a different signal than a crash into a rigid barrier, for example, even during the early portion of the crash before the airbag triggering decision has been made. A neural network pattern recognition system can be trained to recognize and identify the crash type from this early signal, and other available information such as vehicle speed, and further to forecast ahead the velocity change versus time of the crash. Once this forecast is made, the severity and timing of the crash can be predicted. Thus, for a rigid barrier impact, for example, an estimate of the eventual velocity change of the crash can be made and the amount of gas needed in the airbag to cushion an occupant as well as the time available to inject that amount of gas into the airbag can be determined and used to control the airbag inflation.

Alternately, consider a crash into a highway energy absorbing crash cushion. In this case, the neural network based sensor determines that this is a very slow crash and causes the airbag to inflate more slowly thereby reducing the incidence of collateral injuries such as broken arms and eye lacerations.

In both of these cases, the entire decision making process takes place before the airbag deployment is initiated. In another situation where a soft crash is preceded by a hard crash, such as might happen if a pole were in front of a barrier, the neutral network system would first identify the soft pole crash and begin slowly inflating the airbag. However, once the barrier impact began, the system recognizes that the crash type has changed and recalculates the amount and timing of the introduction of gas into the airbag and sends appropriate commands to the inflation control system of the airbag to increase the introduction of gas into the airbag.

VARIABLE INFLATORS

There are many ways of controlling the inflation of the airbag and several are now under development by the inflator companies. One way is to divide the airbag into different charges and to initiate these charges independently as a function of time to control the airbag inflation. An alternative is to always generate the maximum amount of gas but to control the amount going into the airbag, dumping the rest into the atmosphere. A third way is to put all of the gas into the airbag but control the outflow of the gas from the airbag through a variable vent valve. For the purposes herein, all controllable apparatus for varying the gas flow into or out of the airbag over time will be considered as a gas control module whether the decision is made at the time of initial airbag deployment, at one or more discrete times later or continuously during the crash event.

INTEGRATION

The use of neural networks in crash sensors has another significant advantage in that it can share the same hardware and software with other systems in the vehicle. Neural networks have proven to be effective in solving other problems related to airbag passive restraints. In particular, the identification of a rear-facing child seat located on the front passenger seat, so that the deployment of the airbag can be suppressed, has been demonstrated. Also, the use of neural networks for the classification of vehicles or objects about to impact the side of the subject vehicle for use in anticipatory side impact crash sensing shows great promise. Both of these neural network systems, as well as others under development, can use the same computer system as the crash sensor and prediction system. Moreover, both of these systems will need to interact with, and should be part of, the diagnostic module used for frontal impacts. It would be desirable for cost and reliability considerations, therefore, for all such systems to use the same computer system. This is particularly desirable since computers designed specially for solving neural network problems, such as neural-computers, are now available.

THE NEURAL NETWORK SYSTEM

The neural network crash sensor described is capable of using information from three accelerometers, each measuring acceleration from an orthogonal direction. As will be described in more detail below, other information can also be considered by the neural network algorithm such as the position of the occupants, noise, data from anticipatory acoustic, radar, infrared or other electromagnetic sensors, seat position sensors, seatbelt sensors, speed sensors, or any other information present in the vehicle which is relevant. Since the algorithm is trained on data from real crashes and non-crash events, it can handle data from many different information sources and sort out what patterns correspond to airbag-required events in a way which is nearly impossible for an engineer to do. For this reason, a crash sensor based on neural networks, for example, will always perform better than one devised by engineers. The theory of neural networks including many examples can be found in several books on the subject including (7-9).

The process can be programmed to begin when an event occurs which indicates an abnormal situation such as the acceleration in the longitudinal direction, for example, exceeding the acceleration of gravity, or it can take place continuously depending on the demands on the computer system. The digital acceleration values from the ADC may be pre-processed, as for example by filtering, and then entered successively into the neural network algorithm which compares the pattern of values on nodes 1 through N with patterns for which it has been trained. Each of the input nodes is connected to each of the second layer nodes h-1,...,h-n, called the hidden layer, either electrically as in the case of a neural computer, or through mathematical functions containing multiplying coefficients called weights.

The weights are determined during the training phase while creating the neural network as described in detail in the text references. At each hidden layer node, a summation occurs of the values from each of the input layer nodes, which have been operated on by functions containing the weights, to create a node value. Similarly, the hidden layer nodes are connected to the output layer nodes, which in this example is only a single node representing the control parameter to be sent to the gas control module. If this value exceeds a certain threshold, the gas control module initiates deployment of the airbag.

During the training phase, an output node value is assigned for every setting of the gas control module corresponding to the desired gas flow for that particular crash as it has occurred at a particular point in time. As the crash progresses and more acceleration values appear on the input nodes, the value of the output node may change. In this way, as long as the crash is approximately represented in the training set, the gas flow can be varied at each one or two milliseconds depending on the system design to optimally match the quantity of gas in the airbag to the crash as it is occurring. Similarly, if an occupant sensor and a weight sensor are present, that information can additionally be fed into a set of input nodes so that the gas module can optimize the quantity of gas in the airbag taking into account both the crash deceleration and also the position, velocity, size and weight of the occupant to optimally deploy the airbag to minimize airbag induced injuries and maximize the protection to the occupant. The details of the neural network process and how it is trained are described in referenced texts and will not be presented in detail here.

A time step such as two milliseconds is selected as the period in which the ADC pre-processes the output from the accelerometers and feeds data to input node 1. Thus, using this time step, at time equal to 2 milliseconds from the start of the process, node 1 contains a value obtained from the ADC and the remaining input nodes have a random value or a value of 0. At time equal 4 milliseconds, the value which was on node 1 is transferred to node 2 and a new value from the ADC is fed into node 1. In a similar manner, data continues to be fed from the ADC to node 1 and the data on node 1 is transferred to node 2 whose previous value was transferred to node 3 etc.. Naturally, the actual transfer of data to different memory locations need not take place but only a redefinition of the location which the neural network should find the data for node 1. For one case, a total of one hundred input nodes were used representing two hundred milliseconds of acceleration data. At each step, the neural network is evaluated and if the value at the output node exceeds some value such as .5 then the airbags are deployed by the remainder of the electronic circuit. In this manner, the system does not need to know when the crash begins, that is, there is no need for a separate sensor to determine the start of the crash or of a particular algorithm operating on the acceleration data to make that determination.

In the example above, one hundred input nodes were used, twelve hidden layer nodes and one output layer node. In this example, accelerations from only the longitudinal direction were considered. If other data such as accelerations from the vertical or lateral directions were also used, then the number of input layer nodes would increase. If the neural network is to be used for sensing rear impacts, or side impacts, 2 or 3 output nodes might be used, one for each gas control module. The theory for determining the complexity of a neural network for a particular application has been the subject of many technical papers and will not be presented in detail here. Determining the requisite complexity for the example presented here can be accomplished by those skilled in the art of neural network design and is discussed briefly below. In another implementation, the integral of the acceleration data is used and it has been found that the number of input nodes can be significantly reduced in this manner.

The particular neural network described and illustrated above contains a single series of hidden layer nodes. In some network designs, more than one hidden layer is used although only rarely will more than two such layers appear. There are of course many other variations of the neural network architecture illustrated above which appear in the literature.

OCCUPANT MONITORING SYSTEM

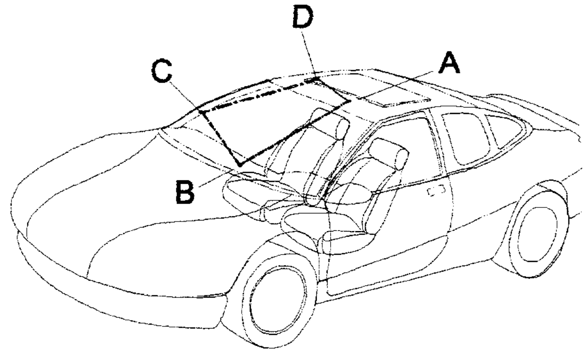

Figure 1. Occupant monitoring system

Figure 1 illustrates an occupant monitoring system that is capable of identifying the occupancy of a vehicle and measuring the location and velocity of human occupants. This system is now being developed for implementation on a production vehicle. In one implementation, four ultrasonic transducers are used to provide accurate identification and position monitoring of the passenger of the vehicle. Naturally, a similar system can be implemented on the driver side. The system is capable of determining the pre-crash location of the critical parts of the occupant, such as his/her head and chest, and then to track their motion toward the airbag with readings as fast as once every 10 milliseconds. This is sufficient to determine the position and velocity of the occupant during a crash event. The implementation described can therefore determine at what point the occupant will get sufficiently out-of-position so that deployment of the airbag should be suppressed as in solving the standard occupant sensing problem. Alternately, the information is used to determine how fast to deploy the airbag. If the weight of the occupant is also known, the amount of gas which should be injected into the airbag and perhaps the out flow resistance can be controlled to optimize the airbag system not only based on the crash pulse but also the occupant properties. This provides the design for Phase 3 Smart Airbags.

Although the system illustrated uses ultrasonic transducers, other systems use a variety of other technologies including electromagnetic (optical, passive or active infrared, radar), capacitive, seatbelt switch, seat and seatback location transducers, weight sensors and in fact any sensing system which can provide relevant information. The neural network is the ultimate "sensor fusion" technology and can use any type of sensors and will provide the system designer with a quantitative measure of the importance of any of the sensors. The optimum combination of four sensors, for example, might be one active infrared sensor, two ultrasonic sensors and a single strain gage weight sensor. The initial investigation might have included, four ultrasonic sensors, two active infrared sensors, four weight sensors, a seat position sensor, a seatback position sensor, and a seatbelt buckle sensor. A cost benefit analysis can easily be performed to determine the effect of adding any particular additional sensor to the system.

ANTICIPATORY SENSING

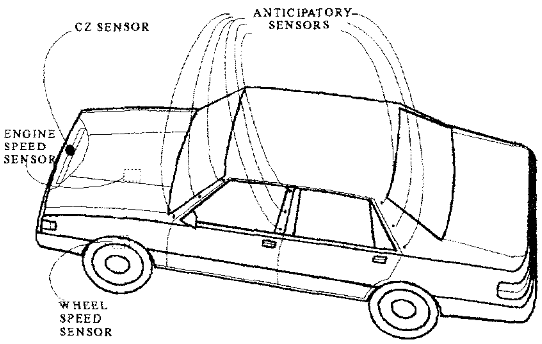

Figure 2. Side impact anticipatory sensor system.

Figure 2 illustrates a side impact anticipatory sensor system using transducers that are situated in different locations on one side of the vehicle, using the same computer system as discussed above. These sensors can provide the data to permit the identification of an object that is about to impact the vehicle at that side as well as its velocity. An estimate can then be made of the object's weight and therefore the severity of the pending accident. This provides the information for the initial inflation of the side airbag before the accident begins. If additional information is provided from the occupant sensors, the deployment of the side airbag can be tailored to the occupant and the crash in a similar manner as described above. Figure 2 also illustrates additional inputs that, in some applications, provide useful information in determining whether an airbag should be deployed. These include inputs from a front crash sensor mounted on the vehicle radiator, an engine speed sensor, and a wheel speed sensor, as used in the antilock braking system sensor.

This anticipatory sensor can act in concert with or in place of the accelerometer-based neural network crash sensor described above. In the preferred case, both sensors are used with the anticipatory sensor forecasting the crash severity before the collision occurs and the accelerometer based sensor confirming that forecast.

Collision avoidance systems currently under development use radar or laser radar to locate objects such as other vehicles that are in a potential path of the subject vehicle. In some systems, a symbol is projected onto the windshield in a heads-up display signifying that some object is within a possible collision space with the subject vehicle. No attempt at present is made to determine what that object is and to display an image of the object. Neural network pattern recognition systems have that capability and future collision avoidance systems may need this capability. Naturally, as above, the same neural network computer system which is proposed herein for sensing crashes can also the used for collision avoidance neural network as well as anticipatory sensing.

OPERATION OF THE SYSTEM

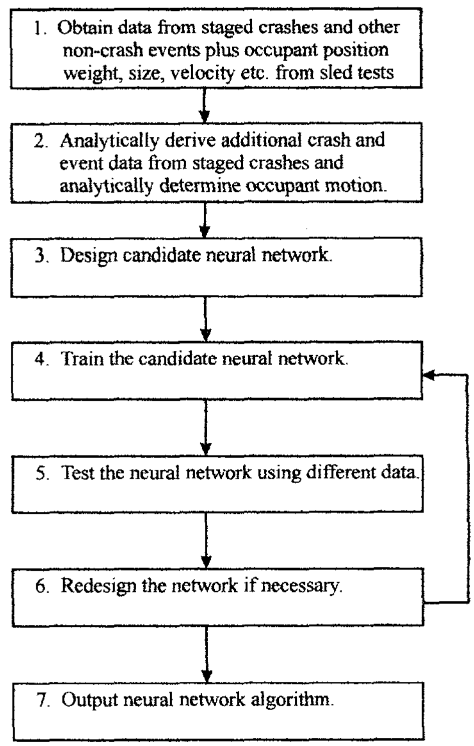

Figure 3. Smart airbag system development block diagram

The neural network algorithm which forms an integral part of the crash sensor described herein can be implemented either as an algorithm using a conventional microprocessor or through a neural computer which is now available. In the former case, the training is accomplished using a neural pattern recognition program and the result is a computer algorithm frequently written in the C computer language. In the latter case, the same neural computer can be used for the training as used on the vehicle. Neural network software for use on a conventional microcomputer is available from several commercial sources.

A block diagram of the neural network computer method of obtaining a smart airbag algorithm is illustrated in Figure 3. In the first step, one or more vehicle models are crashed under controlled conditions where the vehicle and crash dummies are fully instrumented so that the severity of the crash, and thus the need for an airbag, can be determined. An occupant sensor is also present and in use so that occupant motion data can be obtained. The acceleration during the crash is measured at all potential locations for mounting the crash sensor. Normally, any position which is rigidly attached to the main structural members of the vehicle is an adequate mounting location for the sensor.

The following crash event types, at various velocities, are representative of those which should be considered in establishing crash sensor designs and calibrations for frontal impacts:

Frontal Barrier Impact
Right Angle Barrier Impact
Left Angle Barrier Impact
Frontal Offset Barrier Impact
Frontal Far Offset (Outside of Rails) Barrier Impact
High Pole on Center Impact
High Pole off Center Impact
Low Pole (below bumper)Impact
Frontal Car-to-Car Impact
Partial Frontal Car-to-Car Impact
Angle car-to-car Impact
Front to Rear car-to-car Impact
Front to Side Car-to-Car Impact, Both Cars Moving
Bumper Underride Impact
Animal Impact - Simulated Deer
Undercarriage Impact (hangup on railroad track type of object)
Impact Into Highway Energy Absorbing Device (Yellow Barrels, etc.)
Impact Into Guardrail
Curb Impacts

The following non-crash event types are representative of those considered in establishing crash sensor designs and calibrations:

Hammer Abuse (shop abuse)
Rough Road (rough driving conditions)

Normally, a vehicle manufacturer will only be concerned with a particular vehicle model and instruct the crash sensor designer to design a sensor for that particular vehicle model. This is in general not necessary when using the techniques described herein and vehicle crash data from a variety of different vehicle models can and should be included in the training data.

Since the system is being designed for a particular vehicle model, static occupant data needs to be obtained for that particular model. Although crash data from one vehicle can be used for the training purposes for other vehicles, occupant data cannot in general be interchanged from one vehicle model to another. Dynamic position data for an occupant will be in general be analytically derived based on his/her initial position and rules as to how the body translates and rotates which will be determined from sled and crash tests. This is not as complicated as might first appear since an unbelted occupant will usually just translate forward as a free mass and thus the initial position plus the acceleration of the vehicle allows a reasonably accurate determination of his/her position over time. The problem is more complicated for the belted occupant and the rules governing occupant motion must be learned from modeling and verified by sled and crash tests. Fortunately, belted occupants are unlikely to move significantly during the critical part of the crash and thus the initial position at least for the chest is a good approximation.

The vehicle manufacturer will be loath to conduct all of the crashes listed above at several different velocities for a particular vehicle since crash tests are expensive. If, on the other hand, a particular crash type that occurs in the real world is omitted from the library, there is a chance that the system will not perform optimally when the event occurs later resulting in death or injury. One way to partially solve this dilemma is to use crash data from other vehicles as discussed above. Another method is to create data using the data obtained from the staged crash tests and operating on the data using various mathematical techniques which permits the creation of data which is representative of crashes not run. One method of accomplishing this is to use velocity and crash scaling as described in detail in the referenced papers and particularly in reference (1) page 8 and reference (2) pages 37-49 . This is the second step in the process illustrated in Figure 3. Also included in this step is the analytical determination of the occupant motion discussed above.

The third step is to assume a candidate neural network architecture. A choice which is moderately complex is suggested. If the network is too simple, there will be cases for which the system cannot be trained and, if these are important crashes, the network will have to be revised by adding more nodes. If the initial choice is too complex, this will usually show up after the training with one or more of the weights having a near zero value. In any event, the network can be tested later by removing one node at a time to see if the accuracy of the network degrades. Alternately, genetic algorithms are used to search for the optimum network architecture.

The training data must now be organized in a fashion similar to the way it will be seen on a vehicle during a crash. Although data from a previously staged crash is available for the full time period of the crash, the vehicle mounted system will only see the data one value at a time. Thus, the training data must be fed to the neural network computer, or computer program, in that manner. This can be accomplished by taking each crash data file and creating 100 cases from it, assuming that the time period chosen for a crash is 200 milliseconds and that each data point is the pre-processed acceleration over two milliseconds. This data must also be combined with the occupant data derived as discussed above. The first training case contains the first crash data point and the remaining 99 points are zero, or random small values for the crash data nodes, and the segmented occupant position data for the occupant nodes.

The second crash data case contains the first two data points with the remaining 98 points set to zero or random low values etc. For the tenth data file, data point one will contain the average acceleration at twenty milliseconds into the crash, data point two the average acceleration at eighteen milliseconds into the crash, and data point ten will contain the data from the first two milliseconds of the crash. This process is continued until the one hundred data cases are created for the crash. Each case is represented as a line of data in the training file. This same process must be done for each of the crashes and non-crash events for which there is data. A typical training set will finally contain on the order of 50,000 crash data cases and 500,000 occupant static data cases.

In the pure neural network crash sensor case, it was possible to substantially trim the data set to exclude all those cases for which there is no definite requirement to deploy the restraint, and the same is true here. For a particular 30 mph frontal barrier crash, for example, analysis of the crash has determined that the sensor must trigger the deployment of the airbag by 20 milliseconds. It is therefore not necessary to use data from that crash at less than 20 milliseconds since we are indifferent as to whether the sensor should trigger or not. Although data greater than 20 milliseconds is of little value from the point of view of a neural network crash sensor which only needs to determine whether to deploy the airbag since that would represent a late deployment, such is not the case here since, for some gas control modules, the inflation/deflation rate can be controlled after the decision to deploy. Also, the 20 millisecond triggering requirement is no longer applicable since it depends on the initial seating position of the occupant. For cases where the airbag should not trigger, on the other hand, the entire data set of 200 data files must be used. Finally, the training set must be balanced so that there are about as many no-trigger cases as trigger cases so that the output will not be biased toward one or the other decision. This then is the fourth step in the process as depicted in Figure 3.

In the fifth step, the neural network program is run with the training set. The program uses a variety of techniques, such as "back propagation", to assign weights to the connections from the input layer nodes to the hidden layer nodes and from the hidden layer nodes to the output layer nodes to try to minimize the error at the output nodes between the value calculated and the value desired. For example, for a particular crash such as a 30 mph frontal barrier impact, an analysis of the crash and the particular occupant has yielded the fact that the sensor must trigger in 20 milliseconds and the data file representing the first 20 milliseconds of the crash would have a desired output node value which would instruct the gas module to inject a particular amount of gas into the airbag. For another crash such as an 8 mph barrier crash where airbag deployment is not desired, the desired output value for all of the data lines which are used to represent this crash (100 lines) would have associated with them a desired output node value of 0 which corresponds to a command to the gas control module not to inject gas into the airbag. The network program then assigns different weights to the nodes until all of the airbag-deployment-not-desired cases have an output node value nearly equal to 0 and similarly all of the airbag-deployment-desired cases have an output value close to that which is required for the gas control module to inject the proper amount of gas into the airbag. The program finds those weights that minimize the error between the desired output values and the calculated output values.

The term weight is a general term in the art used to describe the mathematical operation which is performed on each datum at each node at one layer before it is inputted into a node at a higher layer. The data at input layer node 1, for example, will be operated on by a function that contains at least one factor which is determined by the training process. In general this factor, or weight, is different for each combination of an input node and hidden layer node. Thus, in the example above where there were 100 input nodes, 12 hidden layer nodes and 1 output node, there will in general be 1,212 weights which are determined by the neural network program during the training period. An example of a function used to operate on the data from one node before it is input to a higher level node is the sigmoid function:

In the usual back propagation trained network, let

O_{ij} be the output of node j in layer i,

then the input to node k in layer i+1 is

$$I_{i+1,k} = \sum_j W_{kj}^{(i)} O_{ij}$$

where $W_{kj}^{(i)}$ is the weight applied to the connection between node j in layer i and node k in layer i+1.

Then the output of node k in layer i+1 is found by transforming its input, for example, with the sigmoid function:

$$O_{i+1,k} = 1/(1+e^{-I_{i+1,k}})$$

and this is used in the input to the next, i+2, layer.

If the neural network is sufficiently complex, that is if it has many hidden layer nodes, and if the training set is small, the network may "memorize" the training set with the result that it can fail to respond properly on a slightly different case from those presented. This is one of the problems associated with neural networks which is now being solved by more advanced pattern recognition systems including genetic algorithms which permits the determination of the minimum complexity network to solve a particular problem. Memorizing generally occurs only when the number of vectors in the training set is not sufficiently large or varied compared to the number of weights. The goal is to have a network which generalizes from the data presented and therefore which will respond properly to a new case which is similar to but only slightly different from one of the cases presented. The network can also effectively memorize the input data if many cases are nearly the same. It is sometimes difficult to determine this by looking at the network so it is important that the network not be trained on all available data but that some significant representative sample of the data be held out of the training set to be used to test the network. It is also important to have a training set that is very large and varied (one hundred to one thousand times the number of weights or more is desirable). This is the function of step five, to test the network using data that it has not seen before, i.e., which did not constitute part of the training data.

Step six involves redesigning the network and then repeating steps three through five until the results are satisfactory. This step is automatically accomplished by some of the neural network software products available on the market.

The final step is to output the computer code for the algorithm and to program a microprocessor, or a neural computer, with this code. One important feature of this system is that the neural network system chosen is very simple and yet, because of the way that the data is fed to the network, all relevant calculations are made with a single network. There is no need, for example, to use an additional network to translate a prediction of a vehicle velocity change, and thus the crash severity, into a setting for the gas controller. In fact, to do this would be difficult since the entire time history would need to be considered. The output from the network is the setting of the gas controller in the preferred system design.

OPERATION OF THE NEURAL NETWORK CRASH SENSOR - AN EXAMPLE

In Figure 4, the results of a neural network pattern recognition algorithm for use as a single point crash sensor are presented for a matrix of crashes created according to the velocity and crash scaling techniques presented in the referenced papers. The table contains the results for different impact velocities (vertical column) and different crash durations (horizontal row). The results presented for each combination of impact velocity and crash duration consist of the displacement of an unrestrained occupant at the time that airbag deployment is initiated and 30 milliseconds later. This is presented here as an example of the results obtained from the use of a neural network crash sensor that forms the basis of the smart airbag system. In Figure 4, the success of the sensor in predicting that the velocity change of the accident will exceed a threshold value is demonstrated. Here this capability is extended to where the particular severity of the accident is indirectly determined and then used to set the flow of gas into or out of the airbag to optimize the airbag system for the occupant and the crash severity.

Airbags have traditionally been designed based on the assumption that 30 milliseconds of deployment time is available before the occupant, as represented by a dummy corresponding to the average male, has moved five inches. An occupant can be seriously injured or even killed by the deployment of the airbag if he or she is too close to the airbag when it deploys and in fact many people, particularly children and small adults, have now been so killed. It is known that this is particularly serious when the occupant is against the airbag when it deploys which corresponds to about 12 inches of motion for the average male occupant, and it is also known that he will be uninjured by the deploying airbag when he has moved less than 5 inches when the airbag is completely deployed. These dimensions are based on the dummy that represents the average male, the so-called 50% male dummy, sitting in the mid-seating position. The threshold for significant injury is thus somewhere in between these two points and thus for the purposes of this table, two benchmarks have been selected as being approximations to the threshold of significant injury. These benchmarks are, based on the motion of an unrestrained occupant, (i) if the occupant has already moved 5 inches at the time that deployment in initiated, and (ii) if the occupant has moved 12 inches by the time that the airbag is fully deployed. Both benchmarks really mean that the occupant will be significantly interacting with the airbag as it is deploying. Other benchmarks could of course be used; however, it is believed that these two benchmarks are reasonable lacking a significant number of test results to demonstrate otherwise, at least for the 50% male dummy.

The tables shown in Figures 4 and 5, therefore, provide data as to the displacement of the occupant relative to the airbag at the time that deployment is initiated and 30 milliseconds later. If the first number is greater than 5 inches or the second number greater than 12 inches, it is assumed that there is a risk of significant injury and thus the sensor has failed to trigger the airbag in time. For these cases, the cell in the table has been outlined. As can be seen in Figure 4, which represents the neural network crash sensor, none of the cells are outlined so the performance of the sensor is considered good.

The table shown in Figure 5 represents a model of a single point crash sensor used on several production vehicle models in use today. In fact, it was designed to be optimized for the crashes shown in the table. As shown in Fig. 5, the sensor fails to provide timely airbag deployment in a significant percentage of the crashes represented in the table. Since that sensor was developed, several manufacturers have developed crash sensor algorithms by trial and error which probably perform better than that of Figure 5. It is not possible to ascertain the success of these improved sensors since the algorithms are considered proprietary. Some algorithms have recently been published in the patent literature and can now be analyzed using the above methods.

SCALED VELOCITY	BARRIER SCALING FACTOR					
	1	1.2	1.4	1.6	1.8	2
8 MPH	NT	NT	NT	NT	NT	NT
10 MPH	NT	0.7/2.9	0.9/3.1	1.0/3.0	NT	NT
12 MPH	0.0/1.1	0.8/3.5	0.9/3.5	1.0/3.4	1.4/3.9	2.0/4.7
14 MPH	0.0/1.2	0.9/4.1	1.0/3.8	1.2/4.0	1.3/4.0	1.7/4.5
16 MPH	0.0/1.4	0.9/4.4	1.0/4.0	1.1/4.0	1.4/4.3	1.7/4.6
18 MPH	0.0/1.6	0.8/4.2	0.7/3.6	1.2/4.5	1.6/4.8	1.8/4.9
20 MPH	0.0/1.8	0.7/4.3	0.7/4.0	1.1/4.3	1.3/4.4	1.0/3.8
22 MPH	0.0/1.9	0.5/3.9	0.7/4.0	0.9/4.1	1.2/4.6	1.1/4.2
24 MPH	0.0/2.1	0.1/2.3	0.8/4.4	0.8/4.2	1.3/5.0	1.4/4.8
26 MPH	0.0/2.3	0.1/2.5	0.5/4.0	0.9/4.5	1.0/4.4	1.2/4.6
28 MPH	0.0/2.5	0.0/2.1	0.1/2.4	0.7/4.2	0.8/4.1	0.5/3.2
30 MPH	0.0/2.7	0.0/2.3	0.1/2.6	0.1/2.3	0.8/4.4	1.2/5.0
32 MPH	0.0/2.8	0.0/2.4	0.1/2.8	0.1/2.5	0.9/4.7	1.1/4.9
34 MPH	0.0/3.0	0.0/2.3	0.0/2.0	0.0/1.8	0.6/4.2	1.2/5.3

Figure 4. Neural network single point sensor performance.

SCALED VELOCITY	BARRIER SCALING FACTOR					
	1	1.2	1.4	1.6	1.8	2
8 MPH	NT	NT	NT	NT	NT	NT
10 MPH	4.7/10.3*	NT	NT	NT	NT	NT
12 MPH	2.2/6.7	5.8/12.1	NT	NT	NT	NT
14 MPH	2.2/7.2	2.7/7.5	3.9/8.9	NT	NT	NT
16 MPH	2.2/7.6	2.7/7.9	3.4/8.5	4.2/9.3	NT	NT
18 MPH	2.2/8.0	2.8/8.7	3.6/9.2	4.2/9.7	5.0/10.5	17.8/27.5
20 MPH	2.0/7.9	3.1/9.3	3.7/9.7	4.3/11.2	5.0/10.9	5.9/11.7
22 MPH	1.0/5.3	2.7/8.9	3.9/10.4	4.5/10.9	5.2/11.5	5.9/12.2
24 MPH	.5/4.2	1.6/6.5	3.9/10.8	4.8/11.6	5.4/12.0	6.1/12.8
26 MPH	.4/4	1.2/5.7	2.0/6.8	4.5/11.5	5.8/13	6.4/13.5
28 MPH	.4/4.1	.6/4.0	1.8/6.6	2.7/7.8	5.9/13.5	6.8/14.4
30 MPH	.4/4.2	.5/4.0	.8/4.2	2.2/6.9	6.4/14.5	7.1/15.1
32 MPH	.3/4.2	.5/4.1	.7/7.2	2.1/7.0	2.6/7.4	3.4/8.4
34 MPH	.3/4.0	.5/4.2	.7/4.3	.9/4.5	2.6/7.5	4.0/9.6

Figure 5. Optimized standard single point sensor performance.

GAS FLOW CONTROLLER

One issue that remains to be discussed is the derivation of the relationship between the gas controller setting and the desired volume or quantity of gas in the airbag. Generally, for a low velocity, long duration threshold crash, for a small light weight out-of-position occupant, the airbag should be inflated slowly with a relatively small amount of gas and the out flow of gas from the airbag controlled so a minimum value, constant pressure is maintained unitl the occupant just contacts the vehicle interior at the end of the crash. Similarly, for a high velocity crash with large heavy occupant, positioned far from the airbag before deployment is initiated, but with a significant forward relative velocity due to pre-crash braking, the airbag should be deployed rapidly with a high internal pressure and an out flow control which maintains a high pressure in the airbag as the occupant exhausts the airbag to the point where he almost contacts the interior vehicle surfaces at the end of the crash. These situations are quite different and require significantly different flow rates into and out of the airbag. As crash variability is introduced such as where a vehicle impacts a pole in front of a barrier, the gas flow decisions will be changed during the crash.

In theory the neural network crash sensor has the entire history of the crash at each point in time and therefore knows what instructions it gave to the gas controller during previous portions of the crash. It therefore knows what new instructions to give the controller to account for new information. The problem is to determine the controller function when the occupant parameters and the crash forecasted severity are known. This requires the use of an occupant crash simulation program such as Madymo™ from TNO in Delft, The Netherlands, along with a model of the gas control module. A series of simulations are run with various settings of the controllable parameters such as the gas generation rate, gas inflow and gas outflow restriction until acceptable results are obtained and the results stored for that particular crash and occupant situation. In each case, the goal may be to maintain a constant pressure within the airbag during the crash once the initial deployment has occurred. Those results for each point in time are converted to a number and that number is the desired output of the neural network used during the training. A more automated approach is to couple the simulation model with the neural network training program so that the desired results for the training are generated automatically. Thus, as a particular case is being prepared as a training vector, the Madymo™ program is run which automatically determines the settings for the particular gas control module, through a trial and error process, and these settings are converted to a number and normalized which then become the desired output value of the output node of the neural network. Naturally, the above discussion is for illustration purposes only and there are many ways that the interface between the neural network system and the gas controller can be designed.

The gas flow controller can also make use of additional inputs including in particular the pressure within the airbag. All such information inputs can be handled within the neural network or, in the case of the airbag pressure input, within the control mechanism itself. In this case the output from the neural network would be the desired airbag pressure.

The descriptions above have concentrated on the control of the gas flows into and out of an airbag. Naturally, other parts of the occupant restraint system can also be controlled in a similar manner as the gas flows. In particular, various systems are now in use and others are being developed for controlling the force applied to the occupant by the seatbelt. Such systems use retractors or pretensioners, others use methods of limiting maximum the force exert by the seatbelt, while still others apply damping or energy absorbing devices to provide a velocity sensitive force to the occupant. Also, the crash accelerometer and occupant sensors have been the main inputs to the neural network system as described above. Although not described in detail, the neural network can make optimum use of other sources on information such as seatbelt use, seat position, seat back position, vehicle velocity etc. as additional inputs into the neural network system for particular applications depending on the availability of such information.

CONCLUSION

The system described herein uses a neural network, or neural-network-derived algorithm, to analyze the digitized accelerometer data created during a crash and, in some cases, occupant size, position, seatbelt use, weight and velocity data, and, in other cases, data from an anticipatory crash sensor, to determine not only if and when a passive restraint such as an airbag should be deployed but also to control the flow of gas into or out of the airbag.

Generally, the present device provides a smart airbag system that optimizes the deployment of an occupant protection apparatus in a motor vehicle, such as an airbag, to protect an occupant of the vehicle in a crash. The system includes an accelerometer mounted to the vehicle for sensing accelerations of the vehicle and producing an analog signal representative thereof; an electronic converter for receiving the analog signal from the sensor and for converting the analog signal into a digital signal, and a processor which receives the digital signal. The processor includes a neural network and produces a deployment signal when the pattern recognition system determines that the digital signal contains a pattern characteristic of a vehicle crash requiring occupant protection and further produces a signal which controls the flow of inflator gas into or out of the airbag. In some cases, the system also includes an occupant position and velocity sensor which outputs a signal that is also used by the processor in producing the signal which controls the flow of gas into or out of the airbag.

REFERENCES

1. Breed, D.S. and Castelli, V. "Problems in Design and Engineering of Air Bag Systems", Society of Automotive Engineers Paper SAE 880724, 1988
2. Breed, D.S., Castelli, V. "Trends in Sensing Frontal Impact", Society of Automotive Engineers Paper SAE 890750, 1989.
3. Breed, D.S., Sanders, W.T. and Castelli, V. "A Critique of Single Point Crash Sensing", Society of Automotive Engineers Paper SAE 920124, 1992.
4. Breed, D.S., Sanders, W.T. and Castelli, V. "A Complete Frontal Crash Sensor System - I", Society of Automotive Engineers Paper SAE 930650, 1993.
5. Breed, D.S. and Sanders, W.T. "Using Vehicle Deformation to Sense Crashes", Presented at the International Body and Engineering Conference, Detroit MI, 1993.
6. Breed, D.S., Sanders, W.T. and Castelli, V., "A Complete Frontal Crash Sensor System - II", Proceedings Enhanced Safety of Vehicles Conference, Munich, 1994, Published by the US Department of Transportation, National Highway Traffic Safety Administration, Washington, D.C.

7) Techniques And Application Of Neural Networks, edited by Taylor, M. and Lisboa, P., Ellis Horwood, West Sussex, England, 1993.
8) Naturally Intelligent Systems, by Caudill, M. and Butler, C., MIT Press, Cambridge Massachusetts, 1990.
9) Digital Neural Networks, by Kung, S. Y., PTR Prentice Hall, Englewood Cliffs, New Jersey, 1993.

10. Breed, et al., U.S. Patent 5,563,462 Vehicle Occupant Position And Velocity Sensor.
11. Breed, et al., U.S. Patent (allowed) Vehicle Interior Identification and Monitoring System.