CONTINUOUSLY PREDICTING CRASH SEVERITY

Dorel M. Sala
J. T. Wang
General Motors Corporation
USA
Paper No. 314

ABSTRACT

In this paper we describe a preliminary version of a frontal impact crash sensing algorithm capable of continuously predicting the severity of a crash in real time. This kind of algorithms could be used to control an airbag system with a variable output inflator, which supplies a variable amount of gas into the airbag on demand.

The algorithm consists of two parts linked in series. The first part categorizes the class of an event. The second part predicts the severity of the crash using a function of the occupant free flight displacement and time.

Linear regression and neural network analyses were performed separately to determine the coefficients for the severity function of each crash mode. The algorithm was implemented in Simulink and validated with test data. While both analyses achieved reasonably good correlation between the severity of each event and its corresponding severity function, the neural network analysis generally provided a better correlation.

INTRODUCTION

The newly revised United States Federal Motor Vehicle Safety Standard 208 for frontal impact protection requires new vehicle programs to meet performance requirements for both 50th %ile male and 5th %ile female occupants under multiple test conditions. Most automotive manufacturers are relying on a new generation airbag system, which is equipped with a dual-stage inflator, to meet the revised standard. The additional inflation stage allows engineers to design an airbag system such that a lower inflator output will be generated under a less severe crash event, and *vice versa.*

An algorithm that is capable of continuously predicting the crash severity of a collision event, and a variable inflator that can deliver the amount of gas as required based on the severity of a crash event could potentially offer additional benefit in some crash conditions.

In this paper we present a preliminary version of such an algorithm. The algorithm is also FEA-compatible [1], since in making the airbag deployment decision it uses only cumulative measures like velocity and displacement, which are identified measures that can be reliably predicted by FEA.

CRASH MODE AND DEPLOYMENT DECISION

The example vehicle described in this paper is equipped with two accelerometers: one mounted in front of the vehicle in the crush zone (called the Electronic Front Sensor, or simply the EFS) and another in the vehicle compartment (called the Sensing and Diagnostic Module sensor, or simply the SDM sensor). First, the algorithm determines the deployment decision and crash mode. The algorithm is enabled whenever a potential crash situation is determined. When the acceleration of the SDM sensor reaches a predetermined threshold value, a possible crash event is assessed and the algorithm is enabled. At this time, the crash event clock is initialized and the acceleration from both sensors is integrated to find the velocity. The velocity of the SDM is reported as calculated, while that for the front sensor is reported as the maximum attained. These velocity measures are then integrated to determine displacement measures.

Figure 1 shows the Simulink [2] model of the algorithm for determining the deployment time and the crash mode. The model consists of a series of modules and blocks.

Test data acceleration signals from the SDM and EFS sensors were provided with 16-bit resolution at 0.1 ms sampling interval, and duration of 300 ms from the onset of the crash. These acceleration signals were preprocessed in the SDM and EFS accelerometer models. The preprocessing consisted of filtering the signals at 120 Hz for SDM and 400 Hz for EFS, re-sampling with a 1 ms period, clipping at 50 g for SDM and 250 g for EFS, and A/D conversion with an 8 bit resolution. The signals thus obtained and denoted by As for the SDM and Af for the EFS were input to the Processing module of the algorithm.

The enabling of the algorithm and calculation of the measures is done in the Processing module. Vs is the SDM velocity, calculated by integrating As; Ss is SDM displacement, calculated by integrating Vs; Sf is the EFS max displacement, calculated by integrating the max velocity which in turn is calculated by integrating Af and taking the maximum; Saf is an EFS measure calculated by subtracting Sf from the double integration of absolute Af ; and t is the event clock calculated from the enable time.

The Preprocessing module also contains a reset logic that is used for resetting the algorithm in case the algorithm is enabled but the SDM velocity does not increase above a predetermined threshold. The reset threshold is determined mainly based on the rough road data.

Figure 2 shows the velocities at the SDM location for 11 crash events (all plots are represented vs. the crash event clock). Of these, three events, 16 km/h AZT, 30 km/h left 40% offset deformable barrier (ODB) and 20 km/h 0° barrier impacts are no-deployment events.

Information from SDM and EFS locations is combined to discriminate between non-deployment and deployment events in a timely manner.

The events are classified into three modes (namely Frontal mode, Angle/ODB mode and Pole mode) based on their similar crash signatures. The decision whether or not to deploy the airbag is made by three parallel modules that compare SDM and EFS displacement measures with a set of thresholds. The thresholds of each module are calibrated based on the data from events that belong to the same crash mode.

The module labeled 'Front' corresponds to the Frontal mode and will be triggered by the signals like the 0° frontal barrier impact events. The module labeled 'Angle/ODB' corresponds to the Angle/ODB mode and will be triggered by signals like the angle and ODB impact events. The module labeled 'Pole' corresponds to the Pole mode and will be triggered

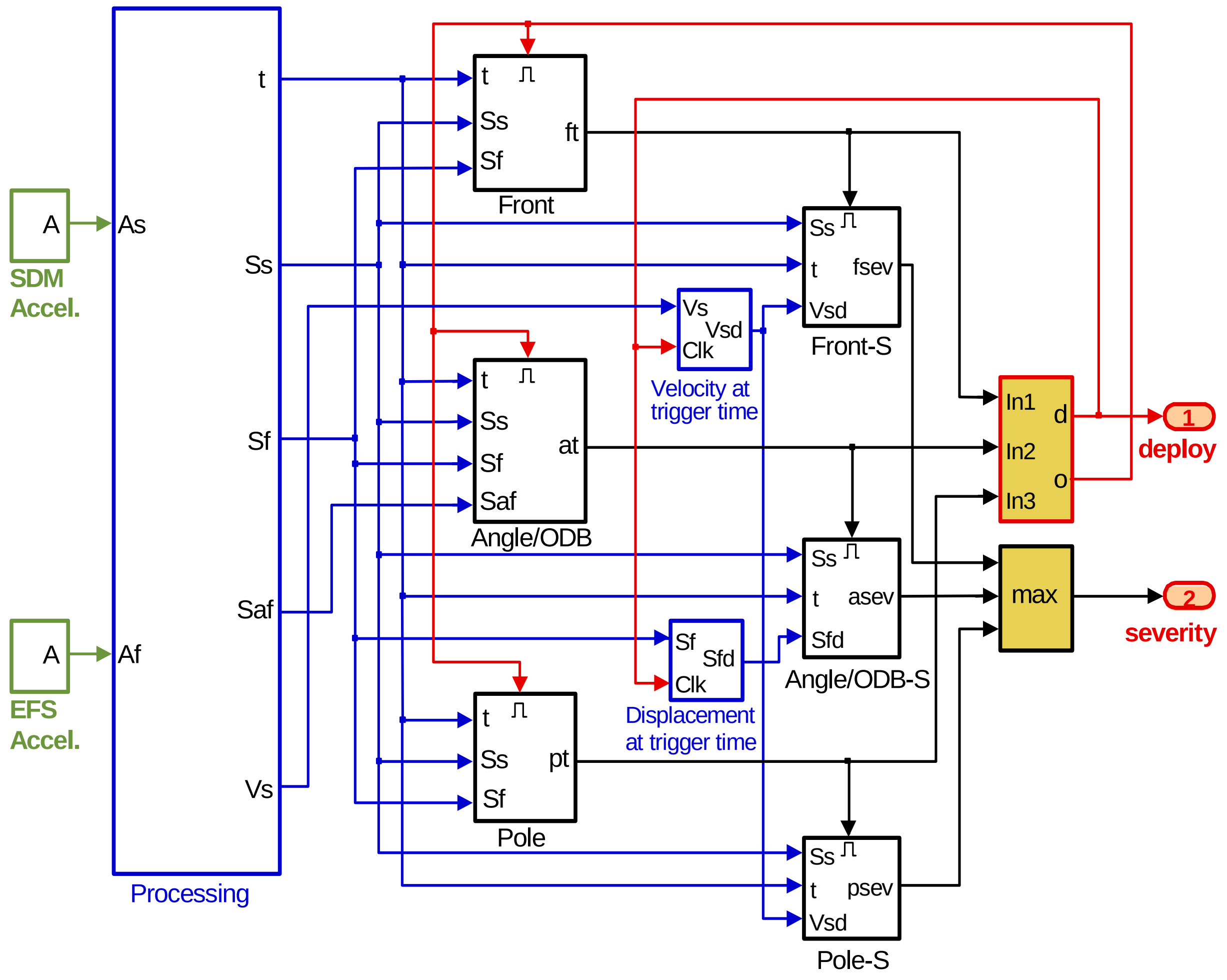

Figure 1. Simulink model of the algorithm.

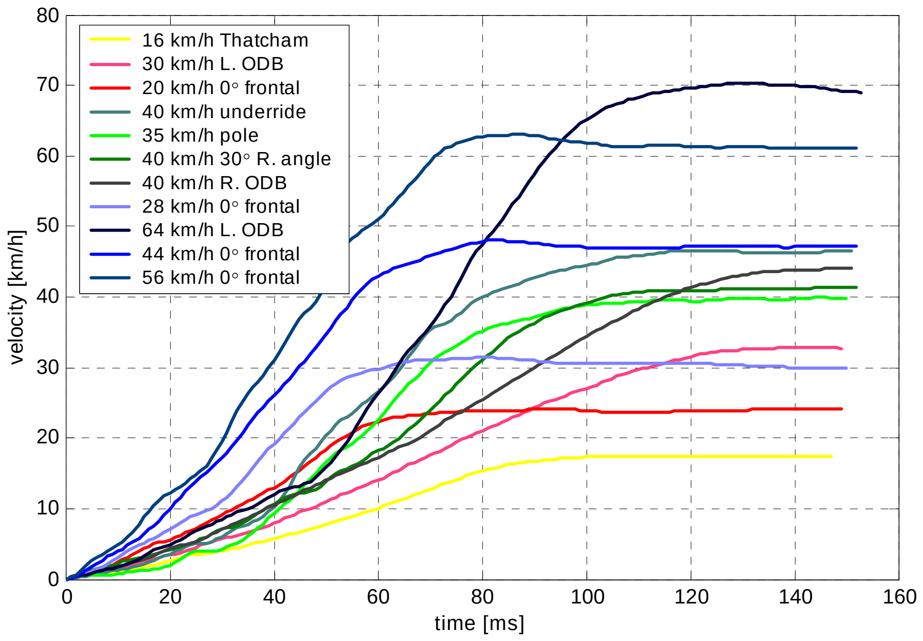

Figure 2. SDM velocities.

by the signals like the pole impact events. All three crash mode modules are working in parallel until a deployment decision is made. The module that triggers the deployment also initiates the severity prediction calculation according to the crash mode represented by that module.

CRASH SEVERITY

We chose to define the crash severity based on the impact velocity of the occupant with the steering wheel. The driver is assumed to be in free flight and at a distance of 350 mm from the steering wheel. Figure 3 depicts the impact velocities for all events ranked in ascending order. The first three events are no deployment events. Based on this ranking we simplified the severity for each event as shown in Figure 4. Note that the severity is generally in agreement with a broader classification used for dual stage airbags.

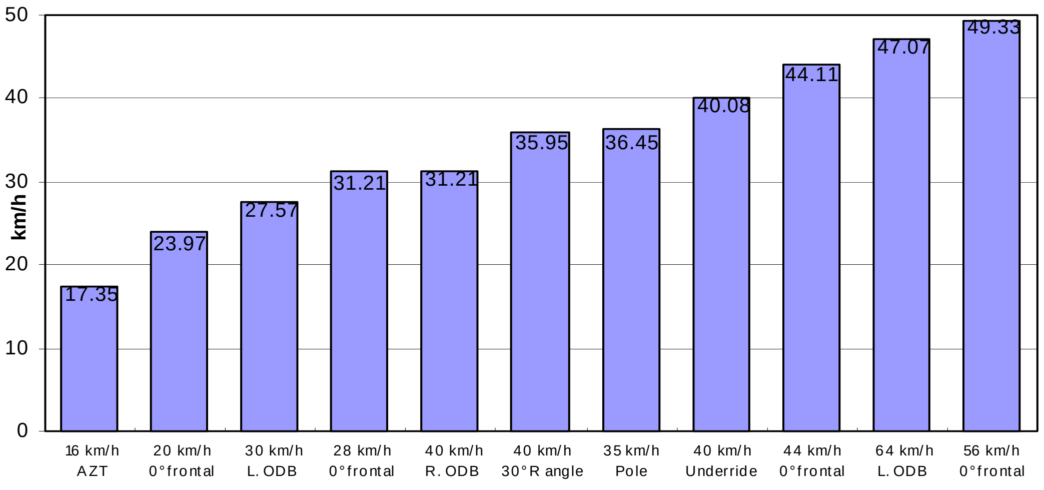

Figure 3. Free flight driver impact velocity at 350 mm displacement.

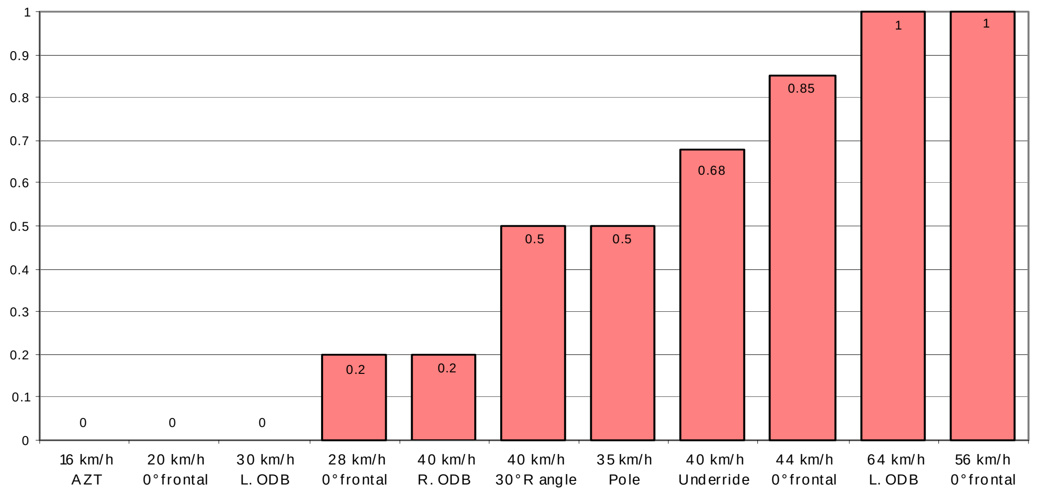

Figure 4. Normalized crash severity of various events.

Regression Approach

It is obvious that the deployment of the airbag has to take place a long time before the driver hits the steering wheel. That means we have to predict the driver impact velocity in real time. In the following we assume that the prediction of the severity of an event can be reasonably accurate for a period of 20 ms past deployment time. The information provided by the EFS accelerometer, past the deployment time, becomes unreliable and therefore, we did not use it in determining the severity. That left us to use only information from the SDM location.

The severity measure that we arrived at has the characteristics of an average acceleration and is defined by equation 1.

$$\text{severity}(t) = ks \cdot \frac{Ss(t)}{t^2} - kv \cdot \frac{Vsd}{t} \qquad (1.)$$

The coefficients ks and kv depend on the crash mode. Vsd is the SDM velocity at deployment time and is used here to compensate the larger values of the measures for events that fire later. Its influence in the formula is decreasing with time. As mentioned previously we had to determine two coefficients for each crash mode, six in total. The values have to provide not only the ranking within each category but also over all events. For each mode we started by calculating the values of the displacements up to 20 ms past trigger time and of the velocities at trigger time. Then we used a linear regression technique to determine the initial values for the coefficients. In general, the values thus obtained were not the best considering the possible variation of the crash signals. By trial and error we modified the initial values. Table 1 shows the values of the final coefficients.

The implementation of the severity calculation in Simulink is shown in Figure 5.

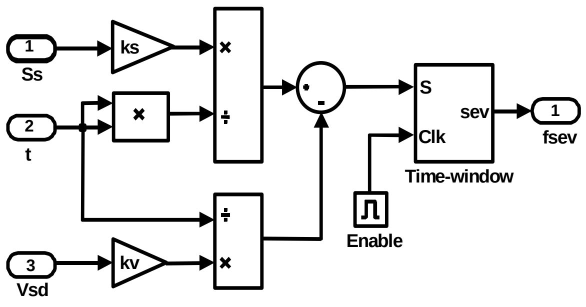

Figure 5. Severity calculation module.

Table 1.

Coefficients ks and kv

	ks	kv
Frontal	135	1
Angle/ODB	330	3.25
Pole	345	2

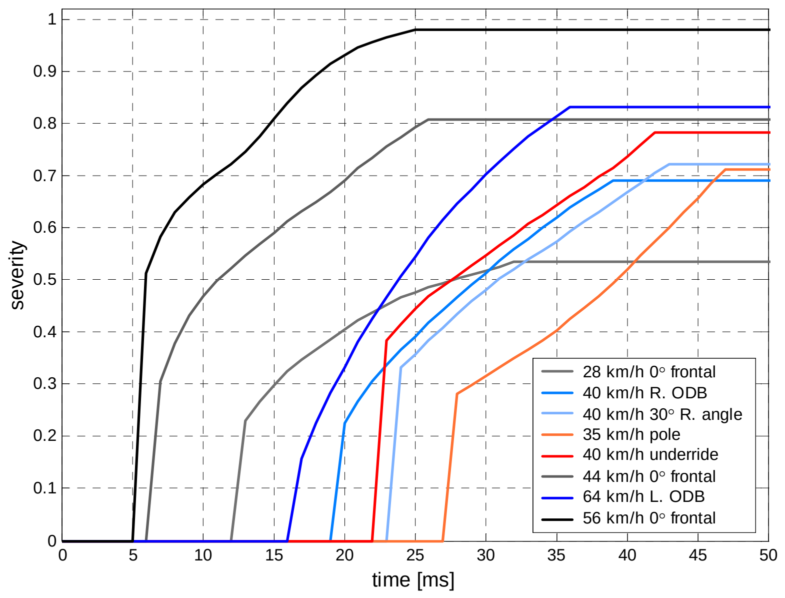

Figure 6. Severity output for all modes – nominal values.

In Figure 6 the output of the Simulink model is shown for all eight deployment events. The time is indicated in ms after event detection. For some events the severity at deploy time may be very small but within the time window of 20 ms they reach their rank. For example the 64 km/h left ODB starts with a small value, 0.16 and reaches 0.83 by the end of the 20 ms time window.

Neural Network Approach

The previous approach showed that defining the severity as a function of displacement and velocity could approximately match the severity ranking. Considering the variation effects resulted in relatively large variations of the severity. That is why we looked for an improved, more complex mapping function based on similar measures. We employed a neural network to determine the mapping function.

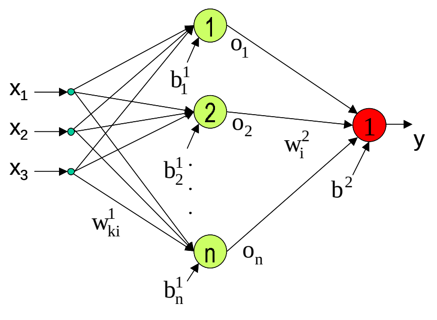

Figure 7. Neural network architecture.

The architecture of the neural network is presented in Figure 7. It is a typical multi-layer network with three inputs corresponding to the measures, one hidden layer and one output neuron [3]. Inputs to the neural network include t, the elapsed time from collision detection; Ss, the SDM displacement at time t; either Vsd, the SDM velocity at the time airbag deployment was triggered, or Sfd, the EFS displacement at the time airbag deployment was triggered. The severity is given by the neural network output. The neural network can be trained using data from collision tests or from simulation of various events in the three collision modes. The desired neural network outputs can be defined as linear functions starting from zero and ending at the value corresponding to the normalized ranking. The neural network utilized is a typical multi-layer network with three inputs corresponding to the measures, one hidden layer and one output neuron. The output, y, of the network can be represented by the equation 2.

$$y = \sum_i w_i^2 \bullet \text{tansig}\left(\sum_k w_{ki}^1 x_k + b_i^1\right) + b^2 \qquad (2.)$$

where x_k are the inputs, w^1_{ki} represents the weight from input k to neuron i, b^1_i is the bias of neuron i, w^2_i represents the weight from the hidden layer neuron i to the output neuron, and b^2 is the bias of the output neuron.

The resulting weights and biases that are developed during the neural network training are calculated once and then recorded in the system that controls the airbag deployment. Table 2 shows the inputs corresponding to each crash mode that were used in the present study.

Table 2.

Neural Network Inputs

	x_1	x_2	x_3
Frontal	Ss	Vsd	t
Angle/ODB	Ss	Sfd	t
Pole	Ss	Vsd	t

To better understand the procedure we illustrate it for the frontal mode. Inputs, $x_{k,}$ to the neural network for the frontal mode include Ss, Vsd and t; and three neurons are utilized in the hidden layer. We let x_1 = Ss, the SDM displacement values for a 20 ms period after the trigger time, x_2 = Vsd, SDM velocity at trigger time and a constant for the network, and x_3 = t, the event time. The test data available for this mode contained data from three crash tests, 28 km/h, 44 km/h and 56 km/h 0° frontal barrier. Using these test data and including possible variations we trained the neural network. As desired outputs we defined linear functions starting from zero and ending at the value corresponding to the ranking shown in Figure 4. The desired outputs used for training are shown in Figure 8.

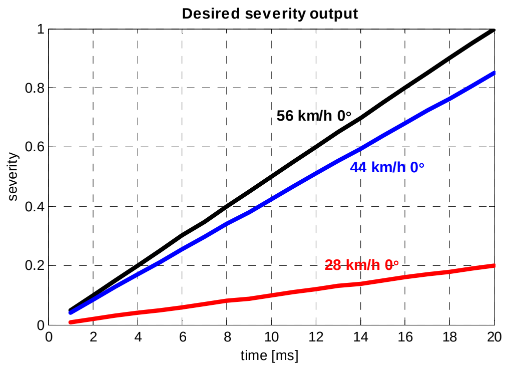

Figure 8. Desired severity output for frontal events used for training.

We chose 3 neurons in the hidden layer and we trained the network until the training error became smaller than 10^{-3}. Table 3 shows the resulting neural network's weights and biases after training. The network was implemented in a Simulink module named Front-S, as shown in Figure 9. The Saturation block limits the output to values between 0 and 1. Presumably, a higher speed collision would have a higher severity value than a lower speed collision. If the airbag is at maximum inflation level for the lower speed collision, it cannot be inflated more. Rather than limiting in the model the severity to one for all collisions above a certain speed we are only limiting the value used to control the inflator. The Time-window block enables the calculations for the 20 ms period after the deployment decision.

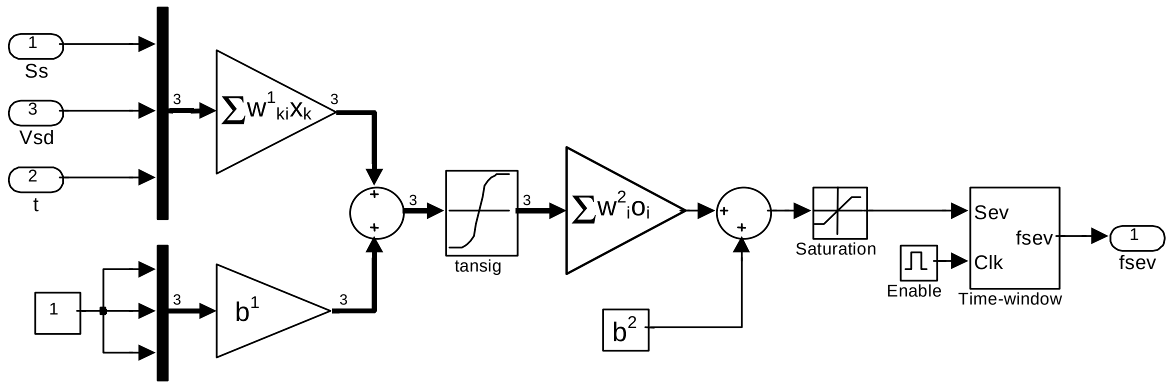

Figure 9. Simulink implementation of the frontal mode severity prediction module.

Table 3.

Frontal mode network weights and biases

Hidden Layer			
w^1_{ki}	-0.6459	0.2973	3.1349
	-0.1548	-0.3174	7.0245
	-0.0016	-0.0624	-1.6418
b^1_i	-0.6469	1.0961	-2.6380
Output Layer			
w^2_i			b^2
-5.1214	1.0593	0.0168	-4.1304

In a similar manner we developed the modules for angle/ODB mode and pole mode. It is worth noting that for angle/ODB mode the second input to the neural network is x_2 = Sfd, the EFS displacement value at trigger time and five neurons were used in the hidden layer.

Figure 10 plots the severity outputs of the algorithm's Simulink model (see Figure 1) for various crash events when the corresponding acceleration signals were input to the model. Only the deploy events are depicted versus event clock indicated in ms.

Using the neural network approach we could match very closely the desired severity ranking. In case one wants to use only a dual stage inflator one could use the same algorithm by choosing an appropriate threshold, for example 0.7 between the two stages.

CONCLUSION

We have developed a preliminary version of a frontal impact crash sensing algorithm to continuously predict the severity of a crash in real time.

The algorithm first determines the airbag deployment time using a crash mode discrimination method. The severity of each event is then predicted by a function of the driver free flight displacement and time. Linear regression and neural network analyses were performed separately to determine the coefficients for the severity function of each crash mode. While both analyses achieved reasonably good correlation between the severity of each event and its corresponding severity function, the neural network analysis generally provided a better correlation. Since the number and type of events available were somewhat limited in this analysis, a larger number of events, as well as a wider range of event types would be necessary for a more rigorous validation of the analyses presented in this paper.

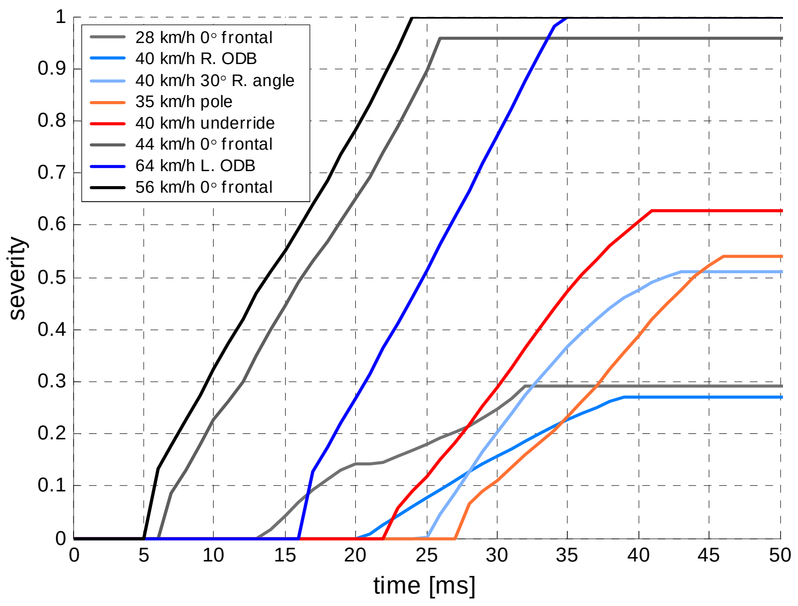

Figure 10. Severity output for all events – nominal values.

REFERENCES

[1] Chin-Hsu Lin, Mark Neal, and J. T. Wang, “Can Finite Element Models be Used to Aid in Calibrating a Crash Sensing System?”, Proceedings of the 2000 ASME Design Engineering Technical Conferences (on CD-ROM), September 10-13, Baltimore, Maryland, Paper No. DETC2000/DAC-1429.

[2] The MathWorks Inc., Using Simulink, Version 4, Nov. 2000.

[3] Laurene V. Fausett, Fundamentals of Neural Networks: Architectures, Algorithms and Applications, Prentice Hall PTR, 1993.