EDC Technical Session
Engineering Dynamics Corporation
February 2003

EDC Library Reference No. 1080

Preface

Since January 1986, Engineering Dynamics Corporation has been publishing its Technical Newsletter for the benefit of it customers: *EDVAP*, *HVE*-2D, and *HVE* users. The purpose of the newsletters is to pass along ideas and innovations that might improve a customer's use of EDC software. These newsletters contain product information, lists of upcoming events, important announcements, and technical articles including the Technical Session, User Hints and Tips, and Frequently Asked Questions. Beginning with the June 2000 Newsletter, a Contributor's Corner section is also included. This document represents the compilation of all technical articles published in the EDC Technical Newsletter from Jan/Feb 1986 to present.

Each Technical Session explores a feature or application of EDC software that goes beyond what is covered in the Physics and Operations Manuals to draw from the 20 + years of safety research experience of the EDC staff. Many columns are specific to one of the EDC physics models, whereas others can be generalized to any simulation or reconstruction.

User Hints and Tips columns give specific advice for using the software more efficiently and effectively. These are often, but not always, specific to either *HVE* or *HVE*-2D.

Frequently Asked Questions sections are geared primarily toward computer troubleshooting and problem solving. They also often address questions that arise as the result of software upgrades, improvements, and added features.

The Contributor's Corner, inspired by the success of the White Paper Session at the 2000 *HVE* Forum, is reserved for *HVE* and *HVE-2D* users to share innovative techniques with each other.

Readers will note that many of the older technical sessions were written with *EDVAP* users in mind. EDC, however, has not been updating *EDVAP* software since January 1994. Even if you have never used *EDVAP*, most of the ideas and concepts contained in these articles are generally applicable to both *HVE*-2D and *HVE* simulation environments as well.

We hope that you find these technical articles both informative and stimulating.

Engineering Dynamics Corporation

Table of Contents for EDC Technical Sessions

Technical Sessions Table of Contents

Technical Sessions Table of Contents

Other Related Sessions (continued)

Legal Sessions

User Hints And Tips

Technical Sessions Table of Contents

User Hints And Tips (continued)

Technical Sessions Table of Contents

User Hints And Tips (continued)

FAQs

Technical Sessions Table of Contents

FAQs (continued)

Technical Sessions Table of Contents

FAQs (continued)

Technical Sessions Table of Contents

FAQs (continued)

Contributor's Corner

Technical Session

This Technical Session describes a powerful feature that is under-utilized: the Event Set-up, Wheels dialog. This dialog has three tabbed pages:

- Tire Blow-out page
- Wheel Damage page
- Wheel Brake page

Each of these pages contains important parameters, especially for vehicles involved in a collision. Their use is described below.

Tire Blow-out

The *HVE* Tire Blow-out Model™ is used for modeling the effects of a tire blow-out (loss of air) during a simulation. The Tire Blow-out Model is a powerful feature, and it's easy to use. The parameters used by the Tire Blow-out Model are:

T_{start} – Time when the blow-out begins (using the AutoStart checkbox causes the blow-out to triggered automatically by some event – such as a collision)

$T_{duration}$ – The time interval over which the deflation occurs. This is normally 0.10 second, although the user may make it as short or as long as desired.

Stiffness Factor – The fraction of the tire's original cornering stiffness and radial stiffness remaining after the blow-out is complete. This is normally 10 percent of the original stiffness.

Rolling Resistance Factor – The increase in the tire's original rolling resistance after the blow-out is complete. This is normally an increase of 10 times the original value.

The *HVE* Tire Blow-out Model page is shown in Figure 1.

The method employed by the Tire Blow-out Model has been described previously and validated [1]. Briefly, the model uses linear interpolation to vary the tire's radial stiffness, cornering stiffness and rolling resistance from their original values to their modified values over the time interval defined by T_{start} and $T_{duration}$. The Tire Blow-out Model may be applied to any or all tires on a vehicle, including individual (inner and/or outer) dual tires. The model provides exceptionally detailed transient information (tire forces, sprung mass roll, pitch and yaw, etc) occurring during and following a tire air loss.

[1] Day, T.D., Grimes, W.D., Blythe, W., "3-Dimensional Simulation of Vehicle Response to Blow-outs," SAE Paper No. 980221.

The *HVE* Tire Blow-out Model should be used for all events involving a reduction in tire air pressure. Examples include handling simulations involving blow-outs on straight or curved paths, as well as events involving collision-related tire debeading. The *HVE* Tire Blow-out Model has also been used successfully for studying tread separation.

The *HVE* Tire Blow-out Model is implemented for *SIMON* and *EDVSM*. A limited, 2-dimensional version is implemented for *EDSMAC4*.

Wheel Damage

The Wheel Damage option improves the modeling of vehicles with wheels damaged by collision. The parameters used by the Wheel Damage option are:

T_{start} – Time when the wheel damage begins (using the AutoStart checkbox causes the damage to triggered automatically by some event – such as a collision).

$T_{duration}$ – The time interval over which the damage occurs. This is normally 0.10 second, although the user may make it as short or as long as desired.

Peak Lock-up Torque – The percentage of braking torque required to lock the wheel on flat terrain under static load. The default value is 100 percent; this may be decreased as required to simulate partial lock-up, or increased as required to ensure lock-up under higher friction or loading conditions.

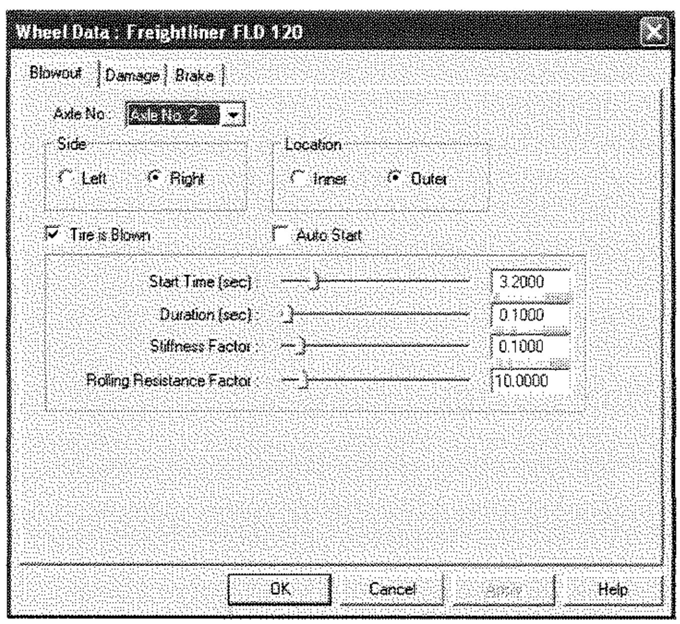


Figure 1 - Event Set-up, Wheels dialog, Tire Blow-out page

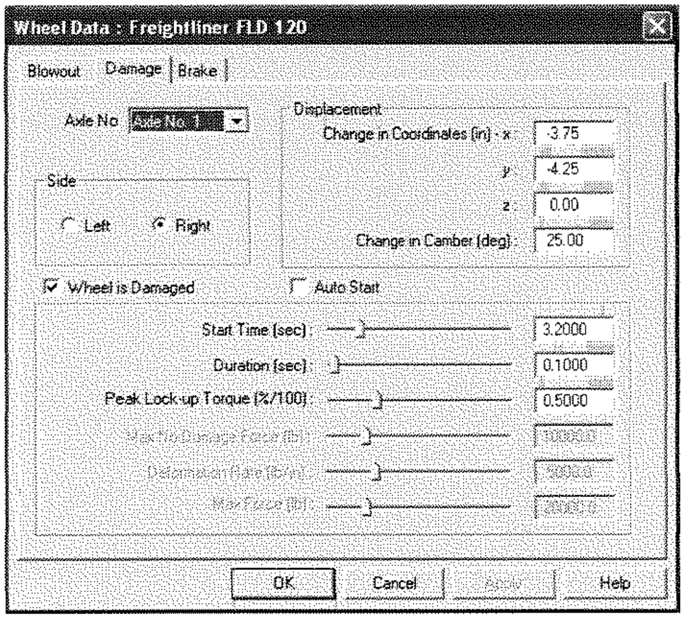


Figure 2 - Event Set-up, Wheels dialog, Damage page

Displacement – The change in vehicle-fixed wheel x,y,z coordinates and camber angle.

The Wheel Damage page is shown in Figure 2.

The Wheel Damage option uses linear interpolation to displace the wheel and/or increase the current level of wheel torque to the peak value over the time interval

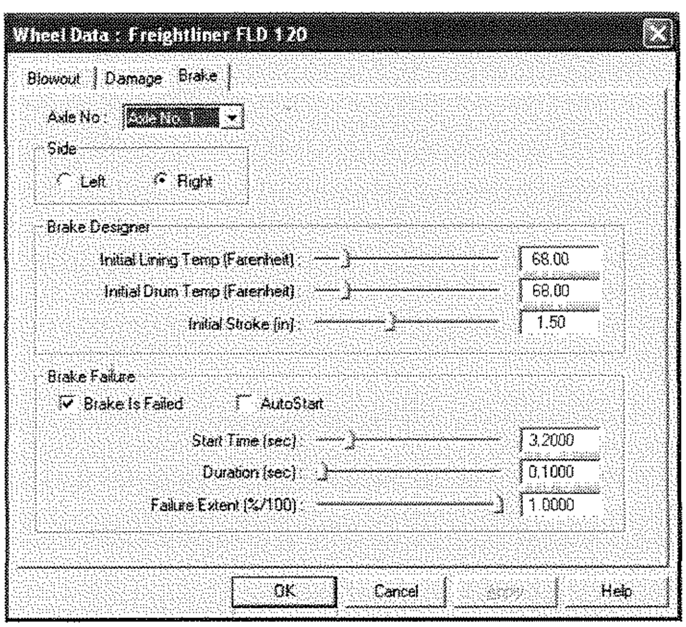


Figure 3 - Event Set-up, Wheels dialog, Brake page

defined by T_{start} and $T_{duration}$. The Wheel Damage option may be applied to any or all wheel locations on a vehicle. When used with the *SIMON* vehicle dynamics model, the Wheel Damage option allows the user to study locked wheels while maintaining the use of the wheel spin equations of motion (use of the Percent Available Friction brake table option clobbers these equations of motion; thus it is not recommended for *SIMON*). *SIMON* uses the current wheel displacements to correctly distribute the vertical tire force, $F_{z'}$, for wheels that have been displaced during a collision. Because the distance from the CG to the tire contact patch affects the yaw moment acting on a spinning vehicle, the post-impact path may be significantly influenced by displaced wheels.

The Wheel Damage option should be used for all events involving a collision-induced wheel lock-up or displacement. The Wheel Damage option is supported by *SIMON*. A 2-dimensional version is implemented for *EDSMAC4*.

Wheel Brake

The Wheel Brake option has two components:

- Brake Designer
- Brake Failure

The parameters included in the Brake Designer component are:

Initial Pad/Lining Temperature – The temperature of the friction material (pad or lining) at the start of the simulation.

Initial Rotor/Drum Temperature – The temperature of the brake rotor or drum at the start of the simulation.

Initial Brake Stroke – For S-cam air brake systems, the stroke at the start of the brake application required to produce brake torque.

The Wheel Brake page is shown in Figure 3.

The Brake Designer option uses a thermodynamic model to compute the internal temperature of the brake lining and drum at 13 locations. These temperatures increase or decrease as braking heat energy is added by brake torque and removed by conduction and convection. These changes in temperature affect the current coefficient of friction of the lining material, as well as the stroke required in an S-cam brake to apply brake torque. The Brake Designer option may be applied to any or all wheel locations on a vehicle.

The Brake Designer option should be used for all events involving prolonged brake application where brake failure at one or more brakes is suspected due to excessive temperature and/or poorly adjusted slack adjusters. The Wheel Brake option is supported by *SIMON* and *EDVDS*. *EDVSM* also uses the Brake Temperature model, but does not include the Brake Stroke portion of the model.

The Brake Failure option is new for *HVE* Version 5. The parameters included in the Brake Failure component are:

T_{start} – Time when the brake failure begins (using the AutoStart checkbox causes the damage to triggered automatically by some event – such as a collision).

$T_{duration}$ – The time interval over which the failure occurs. This is normally 0.10 second, although the user may make it as short or as long as desired.

Failure Extent – The fraction of normal brake torque resulting from failure. The default value is 100 percent (i.e., total brake failure); the percentage may be decreased as required to model partial brake failure.

The Brake Failure option uses linear interpolation to reduce the current level of wheel torque to the specified fraction of normal brake torque over the time interval defined by T_{start} and $T_{duration}$. The Brake Failure option allows the user to study the effects of sudden or gradual brake failure at an individual wheel. The Brake Failure option may be applied to any or all wheel locations on a vehicle. Note that this option is different from the temperature-related brake failure described in the previous section.

The Brake Failure option should be used for all events involving a sudden brake failure at one or more wheels. The Brake Failure option is supported by *SIMON*, *EDVSM* and *EDVDS*.

We opened the Technical Session by saying these options are under-utilized. This is true. Reflecting on each of these features, one realizes how often issues, such as blow-outs, wheel displacements, partial or total wheel lock-up and brake failure occur during real-world crashes. The Event Set-up, Wheel options accurately model these conditions - and they are easy to incorporate into your simulations. We encourage users to use them.

Importing Vehicle Geometries into HVE

If a specific make and model of vehicle is not available from the *EDVDB* vehicle database, a user can order a custom vehicle from EDC or choose to build the vehicle on their own. Users who build their own vehicles may follow procedures outlined in White Paper WP-2006, "Building Vehicles for *HVE*", or in the "Building Vehicles For *HVE* & *HVE-2D*" workshop at the *HVE* Forum.

These procedures include tips on using a digitizer to create the vehicle geometry file. In some cases however, a user may choose not to create their own geometry for the vehicle, but rather purchase one from 3D model suppliers or download one from modeling community websites. Experienced CAD users manipulate the vehicle geometry files obtained from websites or suppliers to match the scale and orientation requirements of *HVE*. An important step however, is importing the geometry into the Vehicle Editor. If the file is of type VRML (.wrl), the file may be brought directly into the Vehicle Editor. If the file is of type DXF, then a series of steps must be followed to translate and then import the file.

Here is a summary of the process using the Environment Editor DXF translator to convert the file from DXF to an Inventor (.iv) format model than can be directly imported into the Vehicle Editor:

1. Go to the Environment Editor and click on the Add New Object button to add an environment. In the Environment Information dialog, choose to open a geometry file. In the file browser for the geometries, change it to display *Files of type* DXF Files and then locate and select the DXF file of the vehicle model.
2. When the vehicle geometry appears in the Environment Editor, the geometry has been converted from DXF to a *HVE*-compatible format. Do not try to save this geometry as an environment model. The temporary file created by the translator is used instead.
3. Using a file browser outside of *HVE*, locate the file named temp.iv in the *HVE* folder. This is the temporary file created by the translator. Rename this file to be appropriate for the vehicle. Move this file to the supportFiles/images/vehicles folder.
4. Start a new *HVE* case or open the case that has the vehicle requiring the geometry. Once the vehicle is displayed in the Vehicle Editor, click on the CG, select Geometry File, change to Files of type .iv and then select the file. The vehicle geometry file will now be displayed on the vehicle.

Matching the Connections Between Vehicles

When a user creates an *EDSMAC4* or *EDVTS* event, the compatibility of the connections (see Table 1) and their relative heights is checked by *EDSMAC4*. If either of these conditions is not satisfied, an error message is displayed informing the user of the problem. The most frequently encountered error message is about the connection heights. The vertical height of the connections between the tow and towed vehicles must be within a distance of 1 inch (2.54 cm) of each other relative to the ground.

Here is a series of steps to help users properly adjust the connection heights of the vehicles before trying to create an event:

1. Display the tow vehicle in the Vehicle Editor. Click on the CG and select Move CG on the pop-up menu. Note the value for CG height displayed at the bottom of the dialog. Close the dialog.
2. Click on the tow vehicle's CG again and select Connections on the pop-up menu. Note the value for the z coordinate for the rear connection. Remember that a negative z coordinate for the connection physically places the connection above the CG of the vehicle because of the SAE coordinate system.
3. Subtract the connection z coordinate from the CG height of the vehicle. (BE CAREFUL! Subtracting a negative number is the same as adding a positive number.) The resulting value is the connection height above the ground for the tow vehicle.
4. Repeat the above steps for the trailer (or towed vehicle) to determine the connection height relative to the ground for the front connection of the trailer.
5. Edit the connection z coordinate of the vehicle(s) to bring them into within 1 inch (2.54 cm) of each other.

Repeat this process for each set of connected vehicles involved in the event. A table of connection compatibility between the vehicles is provided below.

Table 1 - Compatible connections between the tow vehicle and towed vehicle

Tow Vehicle Rear Connection	Towed Vehicle Front Connection
Ball	Hitch
Fifth Wheel	King Pin
Pintle Hook	Fixed Drawbar
Pintle Hook	Hinged Drawbar

HVE and HVE-2D F.A.Q.

This section contains answers to frequently asked questions submitted to our Technical Support staff by *HVE* and *HVE-2D* users.

Q: I have set up an EDSMAC4 simulation involving a truck with unloaded generic class 4 trailer combination with hard braking. I have set the % available friction value to 1.0 for all axles of the truck and the trailer. I am only seeing skidmarks from the truck and not the trailer. Why?

A: The vertical force (load) on the tires of the trailer is less than the value assigned for the Min FZ for skidmark setting in the tire dialog in the Vehicle Editor. In *HVE-2D* this is value is set in the Tire Information dialog. In *HVE*, this value is set in the Tire Physical Properties dialog.

To determine if this is the cause, add the value for Fz' for each tire on the axles of the trailer to your Key Results Windows and see if the values are less than the value set for Min Fz. If so, then you can adjust the value for Min Fz to be less than the vertical tire forces experienced during the simulation.

Q: I am reviewing the work of another reconstructionist who is using HVE or HVE-2D. What information do I need to request from that expert in order to evaluate his or her results?

A: For both *HVE* and *HVE-2D*, you should request the case file (with the events executed).

The case file will contain all environment (including the 3-D geometry) and vehicle data. The case file does not contain any 3-D vehicle geometry files. These files are required for reproducing *DyMESH* simulations involving collisions and *EDVSM* simulations involving rollover. These 3-D geometry files are licensed property and cannot be transferred without permission of the vendor (EDC, Digimation or whomever). These files may however, be purchased from the vendor.

Q: What is the source for the tire data found in the Vehicle Editor?

A: All default tire data in the *HVE* Vehicle Database is based on results from experimental studies performed by Calspan, UMTRI or EDC. The majority of data for passenger vehicle tires is derived from tests performed by Calspan Corporation in 1983 (Extended Tire Testing: Tapia, G.A.; Report number 6871-V-1, Nov. 1983 - available as EDC Library Reference 1092). These tires were tested on Calspan's flat-bed tire test machine. The majority of data for heavy truck tires is derived from tests performed at the University of Michigan in 1981 [unpublished data; University of Michigan Transportation Research Institute, 1981]. These tires were tested on UMTRI's flat bed tire test machine. Tire data for tires that were not tested by either Calspan or UMTRI are found by interpolating between actual test data.

There is very little tire data available from 1983 to 1999. However, in 1999 EDC conducted a comprehensive test program of 21 common tires at the Calspan tire test facilities. This data is available for purchase by contacting EDC Customer Service.

F.A.Q.'s on EDC Website

The EDC website provides a special section dedicated to Technical Support, including answers to F.A.Q.'s from this and previous Newsletters. We encourage you to visit this page to search for answers to your questions before contacting Technical Support for assistance. Go to www.edccorp.com and follow the link to Support.

User Hints and Tips

This section contains useful hints and tips to help *HVE* and *HVE-2D* users make the most of their software.

- When positioning vehicles in *EDCRASH*, remember that heading angles of 200° and -160° are different. *EDCRASH* uses these angles to determine direction of rotation by subtracting the final/rest heading from the impact heading. If the result is positive, the vehicle rotated clockwise. Likewise, if negative, the vehicle rotated counterclockwise.

- When importing DXF format environment models, use the Options button on the Environment Information dialog to "Flip about the X-axis" or "Scale" the model during the import. This will allow you to import the original AutoCAD DXF file and rotate and resize it directly in *HVE*, rather than having to mirror and rotate and flip and scale in AutoCAD before exporting as a DXF. This saves considerable time and effort!

- *HVE* and *HVE-2D* offer many capabilities that users may overlook or not take complete advantage of for their cases. If you want to discover these capabilities and how they can best be applied to your regular routine, then sign up to attend the 2004 *HVE* Forum today!

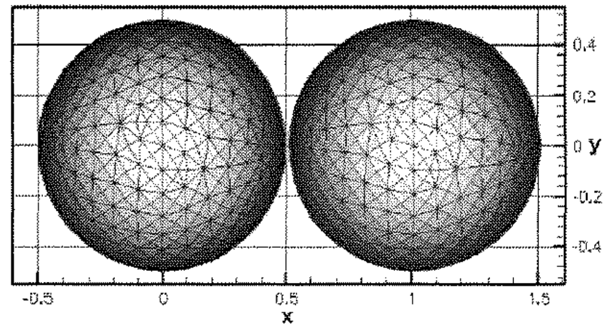
Technical Session

This Technical Session is a high-level introduction to HVE's new DyMESH collision model. Although an overview is provided here, EDC will be holding a DyMESH Applications Workshop during the first week of May at the HVE Forum (see Page 8). The purpose of this seminar is to accelerate the learning curve. Users of DyMESH are strongly encouraged to attend the Forum workshops.

How DyMESH Works

First, a basic observation: DyMESH is not a program, it is a collision algorithm. As such, DyMESH calculates only collision forces and moments between two (or more) colliding vehicles. SIMON is the program (vehicle simulation model) that uses the collision forces and moments calculated by DyMESH. As a result of adding DyMESH, SIMON can be now be used for simulating the entire crash sequence (pre-crash, crash and post-crash). As such, SIMON is much like EDSMAC4, except that SIMON is 3-dimensional (a *huge* difference!).

Conceptually, DyMESH works on a very simple principle: When two meshes come into contact with each other, they become deformed (that is, the meshes' vertices are displaced; see Figures 1 and 2). Each vertex has force-deflection properties; it behaves like a spring. Therefore, pushing on a vertex produces a force. That force may be calculated as the product of its force-deflection properties and its displacement. This calculation is performed for all displaced vertices. Then, DyMESH sums the forces for each vertex, sums the moments about the CG, and *Voila!* DyMESH is done for the current timestep. SIMON's vehicle dynamics engine will add these collision forces and moments to those computed for the tires, connections and aerodynamic surfaces to calculate the vehicle's current acceleration, velocity and position.



(a)

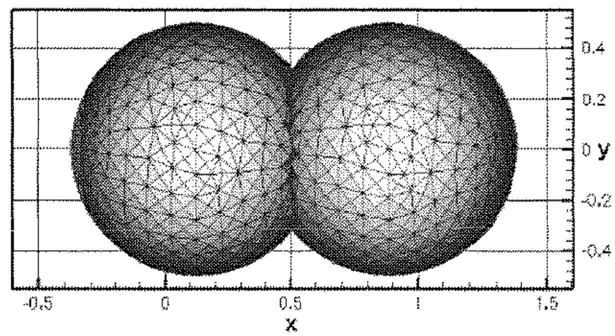


Figure 1 - Two meshes contacting each other and deforming

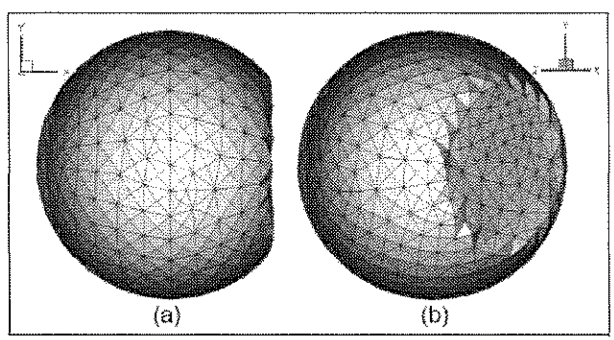


Figure 2 - Meshes after colliding with each other, showing vertex displacement (deformation)

Damage Simulation

Deformation is visualized inherently, simply by viewing the mesh with its displaced vertices. HVE always displays the vehicle mesh (whether the vertices are displaced or not; it makes no difference). The resulting damage profiles are 3-dimensional (see Figure 3)

Data Requirements

DyMESH has no additional data requirements, per se. The data required by DyMESH already exist in an HVE simulation. These are:

- a 3-D vehicle mesh
- stiffness coefficients
- inter-vehicle friction coefficient

If a custom 3-D vehicle mesh is not supplied, HVE automatically provides a generic mesh that can be tessellated (see Tessellation option, below) to any degree of coarseness; typically a 10 to 20 inch mesh size works well. The stiffness coefficients are derived from the traditional A and B coefficients used by EDSMAC4 and EDCRASH. A *Height Factor* is used to provide the 3[rd] dimension. The default Height Factor is 30 inches, because a typical barrier crash test results in about 30 inches of crush height. An inter-vehicle friction coefficient is provided for each pair of colliding vehicles, just as in EDSMAC4. HVE now includes an additional

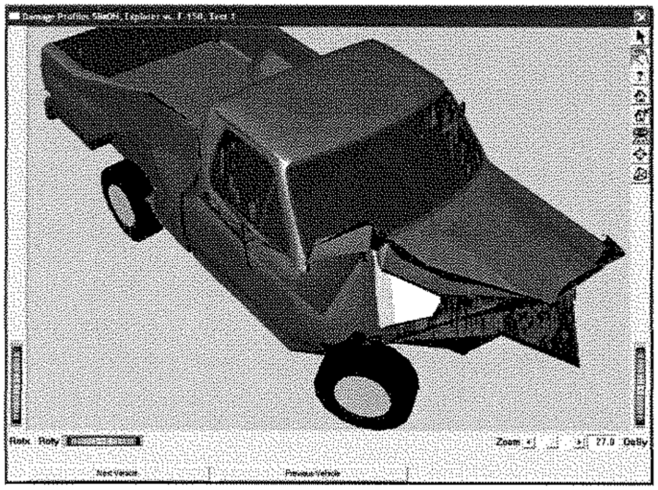


Figure 3 - 3-dimensional damage profile resulting from a DyMESH collision simulation that included secondary impact

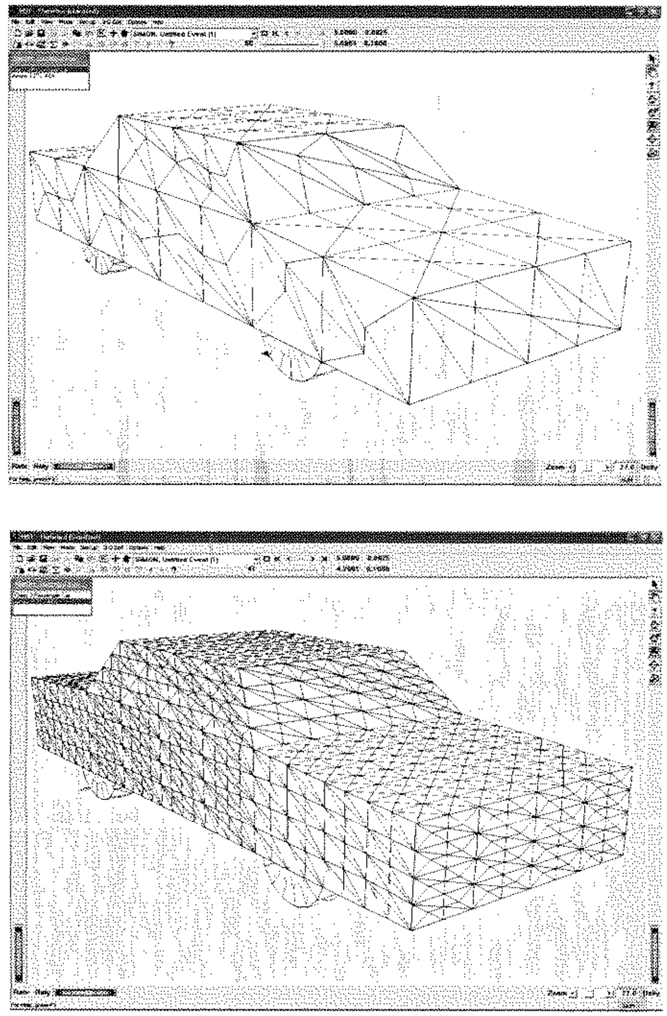


Figure 4 - Generic Vehicle shape with 30 inch tessellation (above) and 10 inch tessellation (below)

feature: the inter-vehicle friction coefficient can be varied for each pair of colliding vehicles (both SIMON and EDSMAC4 now use this feature).

User Interface

HVE now includes a DyMESH Options dialog. This dialog provides access to various options that help DyMESH under certain conditions. These conditions generally arise when the mesh has extremely complex geometry (for example, the inside of a front grill) and the mesh folds back on itself as it crushes. In this and similar cases, the direction of vertex displacement needs close scrutiny; these options provide that scrutiny. There is a *Force to x-y Plane* option that is useful when the simulation predicts excessive over-ride (this phenomenon sometimes occurs when a large vehicle and small vehicle collide at a large closing speed). Several other options are also available, many of which were used during the development and debugging of the DyMESH algorithm (many or all of these options may be eliminated as we gain experience using DyMESH).

HVE also includes a new Vehicle Mesh dialog. This dialog provides two new capabilities: *Tessellation* and *Collision* options. The *Tessellation* parameter establishes a maximum length for the side of any polygon in the mesh. Setting this value has two important effects: First, it establishes the maximum size of the triangles in the mesh. Obviously a mesh with 10 inch polygons will provide more detail than a mesh with 30 inch polygons; see Figure 4. Second, it prevents a mesh from having triangles with large aspect ratios (that is, triangles that are long and skinny: these triangles can be problematic for DyMESH); see Figure 5.

The Vehicle Mesh dialog's Collision Options include Inter-vehicle *Friction* and *Restitution* parameters for each pair of colliding vehicles. EDSMAC4 also uses the *Friction* option. Note that this is a new capability; previous versions used the same inter-vehicle friction for all collisions in the event.

Applications

Generally speaking, DyMESH may be applied to all collisions of all types of vehicles (unit and articulated) and barriers. DyMESH is particularly useful for collisions that are beyond the scope of EDSMAC4. Examples include collisions resulting in 3-dimensional vehicle behavior, such as over-ride and collisions that

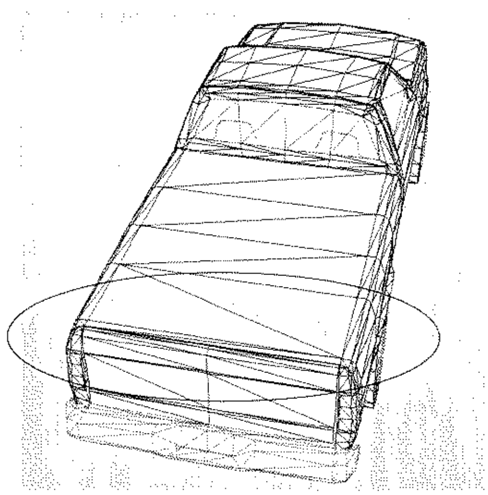


Figure 5 - Region of vehicle reveals large aspect ratio polygons (see inside oval)

produce significant roll and pitch. DyMESH inherently includes the capability to model a collision between the vehicle's body and the environment (after all, the environment is also a mesh!). However, this capability has not yet been integrated into HVE. When integrated (in the next release), DyMESH will allow SIMON users to simulate complete rollovers and collisions with roadside barriers and other obstructions. For the first time, complete rollover simulation of tractor-trailers will be possible.

Limitations

Although very powerful, DyMESH does have limitations, including:

No buckling of surfaces – It follows from the way DyMESH works (pushing on vertices) that surface buckling cannot be modeled. For example, the middle of a hood will not buckle upwards during a severe head-on collision. To model this type of behavior requires a finite element method. Also, vehicle decapitation (such as occurs when a car drives under a trailer, cutting off the top of the roof) and catastrophic collisions that result in the vehicle breaking in two are beyond the scope of DyMESH.

Wheel force not included – Interaction between the wheels and the body mesh is not modeled by DyMESH. Therefore, collision forces and moments produced when the wheel(s) of one vehicle contact the body of another vehicle are not simulated.

Unknown stiffness – HVE provides stiffness coefficients for all six surface of a vehicle (front, sides, back, top and bottom). However, the stiffness coefficients for the top and bottom are essentially unknown and the user must estimate them if these surfaces are deformed. Note that the default A and B stiffness coefficients for the top and bottom are set to 1000; this will result in an obviously unrealistic vehicle response. An example of this condition is the under-ride of a heavy truck or trailer. For these collisions, the A and B coefficients must be reduced.

One collision per vehicle – The initial release of DyMESH is limited to one collision per vehicle. For example, if Vehicle 1 collides with Vehicle 2, then Vehicle 1 continues on to collide with Vehicle 3, DyMESH will fail. This limitation will be eliminated in the next release of DyMESH.

Hardware Requirements

If you are thinking about upgrading your computer, now might be the time! EDC suggests a fast processor (2+ GHz) and a minimum of 512 MB RAM. The hard disk, Open GL-compatible graphics card and other hardware components found on today's computers are adequate.

Run Times

Run times for DyMESH collision simulations using a Pentium 3 GHz vary from about 2 minutes to 15 minutes. Run times are strongly correlated to the number of vertices in the vehicle mesh (vehicles in the HVE Vehicle Database typically have between 2000 and 5000 vertices) and the duration of the collision phase. In one instance, a highly tessellated mesh (approximately 18,000 vertices) undergoing a 3-second sustained contact collision required about 50 minutes to run to completion.

Validation

EDC is presenting a technical paper at the 2004 SAE International Congress on Thursday, March 11th. The paper title is "Validation of the SIMON Model for Vehicle Handling and Collision Simulation – Comparison of Results with Experiments and Other Models," SAE Paper No. 2004-01-1207. This paper includes the results from five staged collisions that were simulated using DyMESH. Please contact EDC if you would like to receive a copy of the paper. The runs in this paper represent a subset of validation runs for approximately ten handling tests and fourteen collision tests.

An important fact is that the validations were run with no tweaking of the DyMESH parameters. This was a major requirement before EDC deemed that DyMESH was ready for release.

HVE and HVE-2D F.A.Q.

This section contains answers to frequently asked questions submitted to our Technical Support staff by *HVE* and *HVE-2D* users.

Q: I have just replaced my old computer with one of the newest models available. When I run HVE on the new computer, I find that my simulation runs seem to go for a few timesteps and then stop. The program doesn't crash and the event controller still responds, but the run just stops. No matter how many times I reset and execute the event, I can't seem to get a complete run. HVE ran fine on my old computer, so what is happening on my new computer?

A: This behavior is caused by the Hyper-Threading (HT) Technology available in the latest Pentium 4 processors (available for 2.4 GHz and faster). Simply stated, the HT Technology is like a multi-processor system, where processes can be routed to separate processors to speed up the performance. With this technology enabled, your computer can mis-route the communication between the HVE Event Viewer and the physics program used for your simulation. This results in your simulation appearing to run for several timesteps and then just stop.

To correct your problem, all you have to do is disable the Hyper-Threading Technology setting in your BIOS. If you are not sure what you need to adjust in your system BIOS, the following Intel web page contains a detailed description of Hyper-Threading Technology and how to possibly enable and disable it on your computer.

www.intel.com/support/processors/pentium4/pentium4_ht.htm#2

If you require further assistance in accessing your system BIOS, you need to contact your computer manufacturer.

Q: I am using a 3-D vehicle geometry (VRML format) that is saved in the MyVehicles folder on my computer. I edited a generic vehicle in the Vehicle Editor to match the vehicle's performance parameters and dimensions, and I even imported the geometry onto the vehicle. Everything looks great in the Vehicle Editor. However, when I create an event involving this vehicle and I try to initially position the vehicle, I receive a error message dialog saying "Inventor Read Error: Can't open file "C:\ProgramFiles\hve\supportFiles\images\vehicles\mycar.wrl". I then close the dialog and I see that my actual geometry has been replaced by the generic image. Why has this happened?

A: This error message is telling you that your vehicle geometry file is not in the expected place where HVE looks for the vehicle geometry while in the Event Mode. The default path location for vehicle geometries is /hve/supportFiles/images/vehicles. If you will place your geometry file in this location, you will find that your vehicle geometry will be displayed in the Vehicle Editor and the event as expected.

Q: I am working on my environment model in the 3D Editor. I have downloaded a tree from a 3D modeling site on the Internet in VRML format and I want to add it directly in the environment. When I select the Library and then my tree file and try to bring it in, HVE crashes. What do I need to do to get this to work properly?

A: You need to bring in the tree first as an environment object and then save it into the library in the 3D Editor. That will then allow you to pull that object out of the library later and add it to any environment model.

Q: I have created a 3D terrain model in CAD and have imported it into HVE. I have a simulation that the vehicle travels along the road and at some point it jumps up or falls through the environment. Why?

A: The most likely answer is that the surface normal for the polygon that you are trying to drive over is facing the wrong direction. *GetSurfaceInfo()* must see the surface normal. You can either use the 3D Editor to reverse the direction of the surface normal for that polygon, or you can return to your CAD program and unify the normals so that they all point in the same direction and then import it once again into HVE.

F.A.Q.'s on EDC Website

The EDC website provides a special section dedicated to Technical Support, including answers to F.A.Q.'s from this and previous Newsletters. We encourage you to visit this page to search for answers to your questions before contacting Technical Support for assistance. Go to www.edccorp.com and follow the link to Support.

User Hints and Tips

This section contains useful hints and tips to help *HVE* and *HVE-2D* Users make the most of their software.

- Want to have one of the ¼ screen viewers in the 3D Editor temporarily full screen for a better view of your model? Try placing the cursor over the viewer and then click the right mouse button. On the pop-up menu, select *Full Screen*. To return to normal view, reverse the steps.

Technical Session

This Technical Session explains how the brake system on a 3-D simulation model (e.g., *SIMON, EDVSM*) works. 3-D simulation users will note that the driver brake table includes the *Pedal Force* option (2-D simulations, not having a brake system model, include only the *Wheel Force* and *% Available Friction* options; although these options may be available in some 3-D simulation models, they reduce the quality of the results and EDC does not recommend their use).

The 3-D brake system model in the simulation works just like it does in a real vehicle. It starts with the driver pressing the brake pedal (in the simulation, this is presented as a table of brake pedal force vs. time). The master cylinder converts this force to brake system pressure (in the simulation, this is called the *Brake Pedal Ratio*). The system pressure may then be reduced through a proportioning valve (modeled by a *Threshold Pressure* and *Proportion Ratio*). Next, the system pressure may be modulated at each wheel by an ABS system. This modulation is controlled by a computer software algorithm in the simulation, just as it is on the actual vehicle. Finally, the system pressure is converted to brake torque by the wheel brake assembly. The simulation calls this the *Brake Torque Ratio*. The entire process is shown schematically in Figure 1.

So, how does the resulting brake torque actually result in braking? In addition to equations of motion for linear motion (X,Y,Z) and rotation (roll, pitch, yaw), 3-D simulations also have equations of motion for wheel spin. Application of first principles to wheel spin, Ω_{Wheel}, results in:

$$\Sigma M = Brake\ Torque - F_{X_{Tire}} = I_{Spin}\ddot{\Omega}_{Wheel}$$

From this, wheel spin acceleration, $\ddot{\Omega}_{Wheel}$, is calculated (I_{spin} is contstant and *Brake Torque* and $F_{X_{Tire}}$ for the current timestep are known):

$$\ddot{\Omega}_{Wheel} = \frac{BrakeTorque - F_{X_{Tire}}}{I_{Spin}}$$

Numerical integration is used to update the spin velocity of the wheel for the start of the next timestep. Using this spin velocity and the current tire radius, R_{Tire}, (also known), the linear velocity, $V_{X_{Tire}}$, at the tire contact patch is calculated:

$$V_{X_{Tire}} = \dot{\Omega}_{Wheel} \cdot R_{Tire}$$

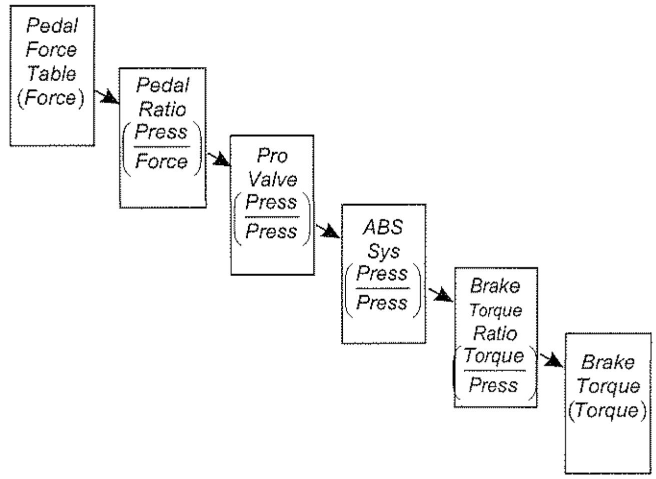


Figure 1 - Schematic flow of 3-D brake system simulation model

The tire longitudinal slip is then:

$$Slip = \frac{V_{Wheel\ Center} - V_{X_{Tire}}}{V_{Wheel\ Center}}$$

The tire's friction (mu vs slip) data are then used to determine directly for the start of the next timestep the longitudinal braking force, $F_{X_{Tire}}$.

Earlier we said that the *Wheel Force* and *% Available Friction* brake table options were not recommended. It is now clear why: Use of either of these methods bypasses completely the wheel spin equations of motion. Any details related to tire spin dynamics are completely lost. Obviously, ABS simulation is also not possible when either of these options is used (refer to the December 2001 Technical Newsletter for more details).

A question sometimes asked by new 3-D users is, "How can I tell how much pedal force to use?" There are a couple of ways to answer this question. First, you have your common sense and years of actual driving experience. Think about it: You probably apply around 10 lb during a normal brake application, and maybe 50 lb (or more) during heavy braking. If you're the analytical type, here's a method for you: First, create a brake table that varies from zero force at time zero, and increases to 100 lb at 5 seconds. Start the vehicle at a high speed, say 80 mph, and execute the run. Then produce a graph of Forward Acceleration vs Brake Pedal Force. *Voila!* You now know precisely how hard to press the brake to achieve a desired deceleration.

HVE Brake Designer

As shown earlier, *Brake Torque Ratio* is one of the key players in the calculation of brake torque. By default, *Brake Torque Ratio* is a single lumped parameter assigned for each wheel and editable by the user. However, we can go a step further: *Brake Torque Ratio* can also be calculated using the mechanical and geometrical properties in the individual brake assembly. The HVE Brake Designer does exactly that.

Different types of brakes (disc, various types of drums, S-cams, wedge, etc) are mechanically different from each other. Free-body diagrams of the forces acting between the pads and rotor (or shoes and drum) are used to compute the brake torque ratio for each of eight-different user-selectable brake types. The brake torque ratio will also vary as changes in brake temperature dynamically affect lining friction. Thus the model allows for a very thorough analysis of brake failure mechanisms. A good example of this is the study of marginal brake adjustment on the ability of a truck driver to maintain control while traveling over a long down-hill grade. See the following three articles for examples: SAE Paper No. 200-01-1294, September 1999 EDC Technical Newsletterr, WP#2003-5. Contact EDC if you would like to receive copies of these articles.

ABS

An ABS (Anti-lock Braking System) system is an optional component that, when present, plays a major role in the calculation of brake torque (see Figure 1). ABS works by reducing brake system pressure, and therefore brake torque, so as to prevent wheel lock-up. This task is accomplished by monitoring the current longitudinal tire slip (actually, it is the wheel spin velocity and acceleration that are being monitored by the ABS computer; longitudinal tire slip may then be inferred). When tire slip reaches a specified level, system pressure is quickly dumped from the brake, thus allowing the wheel to again spin freely. Pressure is then allowed to rebuild. The main goal of this process is to keep the tires rolling so the driver can steer the vehicle. Reduced braking distance is a secondary benefit.

For more information about calculating brake torque, refer to the HVE User's Manual, Section Eleven - HVE Brake Designer. Or contact EDC Technical Support - we're always glad to help!

Using Aerial Photos as Scene Drawings

Many users are working with aerial photographs as the background for their reconstructions and simulations. These photographs may be from a company hired to provide a specific digital image or downloaded directly from image libraries available on the internet as shown in Figure 2.

Figure 2 - Example aerial photograph of a crowded intersection obtained from a company specializing in providing aerial images over the Internet

There are different techniques for using these images, but the basic concepts are similar. You may need to manipulate the image in a photo/drawing editor program. Then, you can use the capabilities of the 3D Editor to apply a "material texture" to a surface and create your scaled site. Here are two approaches that you might consider:

Approach 1 - You may find it easiest to match a square image to a square surface in the 3D Editor. You will need to add background area or crop your image to make its dimensions a perfect square.

Approach 2 - You may choose to work with a rectangular image and use the translation and repeat sliders on the Material Texture dialog to match your image to the surface.

For either approach, follow these general steps:

1. Determine the size of the image in inches or pixels and also the scale of the image in feet per inch or pixel.

2. Make sure your image is saved in the proper location of /supportFiles/images/environments/EnvTextures. It must be in this location to be available as a texture in the 3D Editor.

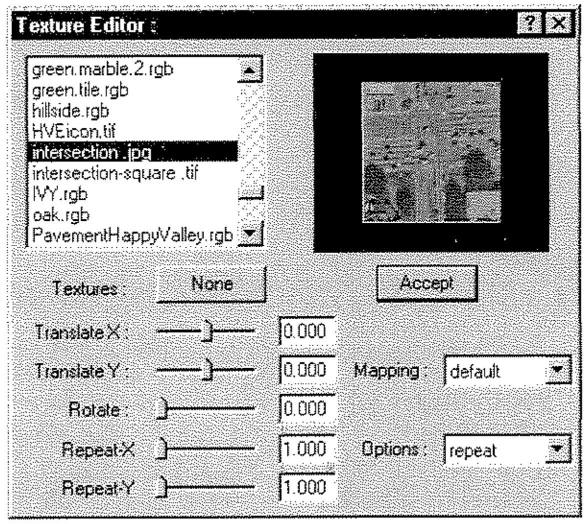


Figure 3 - Material Texture Editor dialog displaying the selected image to be applied to the surface.

3. Build a surface in the 3D Editor that corresponds to the size and shape of the image that you have created. If 1 inch equals 100 feet in your image and your image is 5 inches on edge, make the surface 500 feet.

4. After completing your surface and preparing it for editing, choose 3D Edit, Material Texture from the menu. This will display the Material Texture Editor dialog. Choose your image from the list of textures on the left hand side of the Material Texture dialog as shown in Figure 3.

5. Check the appearance of the image on the sample and adjust the rotation if required. Click the Accept button to attach the texture to your surface.

At this point, if you are using a rectangular image, you will need to adjust the Repeat and also the Translate settings for the image. The value used for Repeat will be the longest side of your image divided by the shortest side. For example 600 pixels divided by 300 pixels is a Repeat value of 2. This will shrink (scale) the texture which was stretched to a square in the Material Texture dialog back to the rectangular shape you started with. You need to click the Accept button to update your image displayed on the surface.

6. That's it! Now create an event and place a vehicle on your suface to test the scale of your scene.

As a side note, you could use the advanced capabilities of a 3D modeling program to create a 3D terrain model and then map an aerial photograph to the surface. This provides the scaled image appearance and also accounts for terrain irregularities and changes when vehicles drive on the surface.

User Hints and Tips

This section contains useful hints and tips to help *HVE* and *HVE-2D* Users make the most of their software.

- Not satisfied with the image quality of your movies, but don't want to change the video codec you are using? Try increasing the settings for Anti-Aliasing from 1 to at least 3. (It is best to do this just after you set your Destination to AVI, as it increases the time to refresh the screen.) You will see immediate improvements. Some users report adjusting the setting up to 10 to achieve a very clear, crisp image. The only drawback to increasing your anti-aliasing setting this high is the length of time your graphics card will take to refresh the screen. You'll have to try and decide for yourself. Remember, if you do turn up your anti-aliasing settings, be sure to set them back down to 1 after making the movie, otherwise your program will be very slow to respond.

HVE and *HVE-2D* F.A.Q.

This section contains answers to frequently asked questions submitted to our Technical Support staff by *HVE* and *HVE-2D* users.

Q: I try to run HVE and I immediately get an error message from Windows saying that the program failed to initialize. Why can I not run HVE?

A: The error message you received probably read something like "The application failed to initialize properly (0xc000142). Click OK to terminate the application."

This error occurs when a Windows program (often running as a service) restricts access to OpenGL. When HVE starts it has to access OpenGL, so the restriction results in HVE failing to initialize. To correct the problem, you must stop the program/service that has exclusive control over OpenGL.

Two programs that we are aware of that cause this problem are MatLab and PCAnywhere. If one of these programs is running as a service, change the Startup Type of the service to Manual. Or you can "disable" the MatLab or PCAnywere program temporarily by right-mouse clicking on their icon in the Taskbar Notification Area and then choosing "disable" or "exit".

If you find other programs that cause this problem, please contact EDC Technical Support.

Q: Can I make a QuickTime format movie directly from the Playback Window?

A: No. The Playback Window produces an AVI format movie. If you need to produce a different format of movie, such as QuickTime or mpeg, you need to convert the movie using a movie editing program such as Adobe Premier.

Q: I've been working on the main collisions of my multiple vehicle crash simulation, but now I want to add an additional vehicle to my EDSMAC4 event? Is this possible?

A: No. When you first created the event, you selected all of the objects that would be allowed in the event. You cannot add or remove any vehicles from the event. However, you have two possible choices to consider. One is to recreate your entire event and add the additional vehicle initially. The other is to add an additional event with only the new vehicle and then combine the Traj Sim sequences together in the Playback Window to complete your sequence.

Q: I have a 16 second simulation run, but I only want to make a movie of the action from t = 5 seconds to t = 13.6 seconds. I've tried moving the slider to select the correct starting frame, but when I change the Destination to AVI and press the Play button to begin recording, the sequence starts from the beginning t = 0. Do I have to use an external movie editing program, or can I somehow do this within the Playback Window?

A: You can do this within the Playback Window. What you need to do is to ***play*** the sequence forward from the beginning to the point where you want it to start recording. (Don't try to use the slider to set it to a starting frame). Press the Pause button to stop the sequence and then change your render settings (if desired). Change the Destination to AVI and press the Play button. Your movie will begin recording from that frame forward. When you reach the point where you want to stop the recording, press the Pause button and change your Destination back to the Playback Window. When you view your AVI, you will see that you have recorded the exact sequence you wanted to.

F.A.Q.'s on EDC Website

The EDC website provides a special section dedicated to Technical Support, including answers to F.A.Q.'s from this and previous Newsletters. We encourage you to visit this page to search for answers to your questions before contacting Technical Support for assistance. Go to www.edccorp.comand follow the links to Support.

Contributors Corner

Interested in learning how to improve the appearance of the textures applied to your environment model? Here is a method submitted by user Eric Deyerl for smoothing out texture maps to reduce the "tiling" effect which often occurs when using a texture map available straight out of the box.

As an example, let's look at a texture map intended to represent a concrete surface in HVE. This texture map is stored as a file named "cement.tif" which is located in the /supportFiles/images/environments/EnvTextures directory within the HVE program directory.

If you used HVE's 3D Editor and created a rectangular roadway surface with dimensions 50 feet by 200 feet and applied the standard "cement" texture to the surface at a repetition of 2.5 cycles in X and 10 cycles in Y, you'd get the environment model shown in Figure 4.

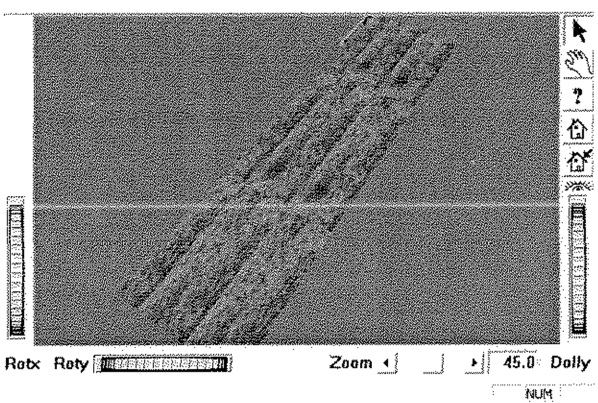


Figure 4 - Roadway with unsmoothed concrete texture map

A viewer of the scene gets the general idea that the surface is comprised of concrete, but the effect is not entirely realistic because of the pronounced tiling effect created by the repeated square boundaries of the applied texture.

This effect can be reduced by modifying the edges of the texture map so that they match better when the pattern is repeatedly applied to a surface. Here's how.

First, open the texture map within an image editing application. This article will describe how to modify the texture using Adobe Photoshop 7.0, but other software packages should have similar features. Before proceeding further, be sure to save the file with a different name so you can experiment freely without fear of losing the original file.

Next, use the Image/Image Size command to determine the dimensions of the texture map in pixels. The cement.tif texture supplied in HVE is 256 pixels by 256 pixels.

Next, engage the Filter/Other/Offset command. In the window which appears, enter the values of "128" in the "Horizontal" and "Vertical" boxes, and put a check in the box labeled "Wrap Around". This will cause the texture to be shifted exactly half its dimension in both the vertical and horizontal directions. By doing this, you have just created a texture tile whose pattern will match exactly with the edges of the tiles adjacent to it on all sides when the texture is repeated. Figure 5 depicts the results of the application of this filter in Photoshop compared to the original texture.

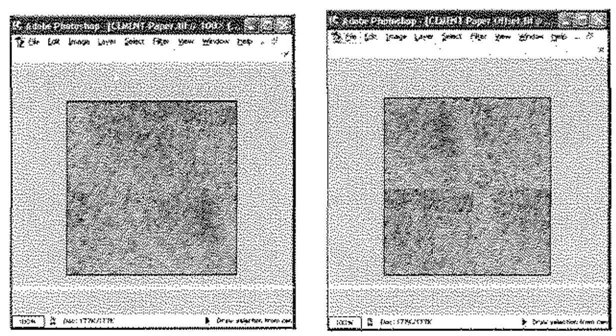


Figure 5 - Original texture map (left) and texture map with offset filter applied (right)

We're not quite done yet, though. As you can see, the offset filter has eliminated the differences in patterns at the edges of adjacent texture maps, but it has created a pair of vertical and horizontal discontinuities which run across the middle of the texture. To minimize this effect, click on the "Clone Stamp" tool in Photoshop. Use this tool to copy smooth areas of the texture onto the hard discontinuities across the middle of the texture.

To make this effect realistic, use a medium-sized brush with a feathered edge from the Brushes palette (a 45 pixel brush works well). You may have to make multiple attempts to get the effect just right. Also, take care not to modify the texture near the edges of the map, as the edges by definition already match perfectly from the application of the offset filter, and any changes to the texture near the edges will destroy this effect. Figure 6 depicts the texture after manipulation with the Clone Stamp tool.

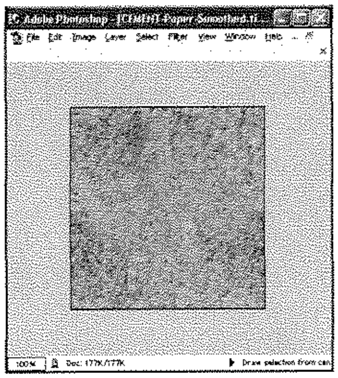


Figure 6 – Texture map after offset filter and smoothing via the Clone Stamp tool

You've now completed the modification of the texture map. Try applying the new texture to the roadway surface you created in HVE. Compare it with the surface mapped with the unmodified texture, and you'll likely find that it appears smoother and more realistic as shown in Figure 7. You'll notice that there is still repetition in the surface – the texture map has been smoothed but is, after all, still being repeated. However, the effect is now less noticeable. Now, your vehicles will appear less like they are traveling across a dance floor. Good luck!

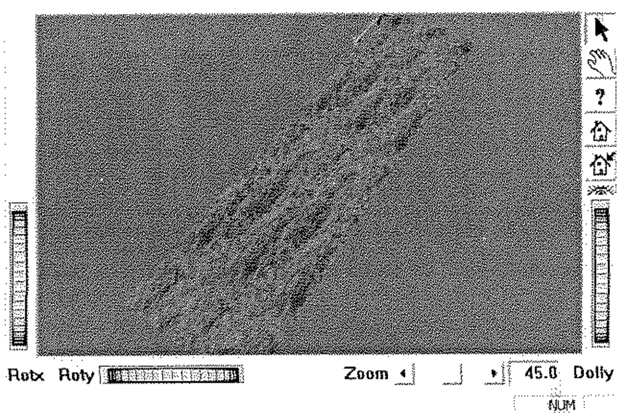


Figure 7 - Roadway with smoothed concrete texture map

If you have any questions or would like to discuss the techniques described in this Contributor's Corner article, please contact the author:

Eric Deyerl, P.E.
214 Main Street, Suite 303
El Segundo, CA 90245
(310) 615-2600
edeyerl@sbcglobal.net

Technical Session

Our Technical Session this time deals with a common task: simulating barrier collisions with *EDSMAC4*. Although there are several approaches, we'll illustrate the best approach – in terms of both the mechanics of the collision and the visualization.

The basic steps are as follows:

- Add the vehicle(s) and barrier(s) to the case
- Add the environment
- Set up and execute the *EDSMAC4* event

We see right away that the basic steps are the same for barrier collision simulation as for all other *HVE* simulations. We chose *EDSMAC4* over *EDSMAC* for three reasons: Our crash involves three vehicles and two barriers (*EDSMAC* is limited to one vehicle and one barrier; *EDSMAC4* has no such limitation). Second, *EDSMAC4* simulates objects with no wheels (e.g., *barriers!*). Third, the *EDSMAC4* collision algorithm has some features that better facilitate barrier collisions. The procedure for setting up and executing this simulation follows.

Description of Crash

In our sample crash, two vehicles (a 1999 Volkswagen Jetta and a 2002 Chevrolet S-10 Extended Cab Pickup) sideswipe each other on a freeway, sending the Jetta spinning into the median barrier. After rebounding back into the traffic lanes, the Jetta is struck by a third vehicle, a 2001 Ford Expedition. The crash is illustrated in Figure 1. Let's build the *EDSMAC4* simulation.

Adding Vehicles and Barriers

From the EDC Custom Vehicle Database, add the 1999 VW Jetta, the 2002 Chevrolet S-10 Extended Cab Pickup and the 2001 Ford Expedition. Then add two Generic SAE J850 Fixed Barriers (because of the length of vehicle-barrier contact, we'll use two barriers).

Next, edit the exterior dimensions of both barriers: Set CG to Front to 500 in; CG to Rear –500 in; CG to Right 200 in; CG to Left –200 in. These exterior dimensions make the barrier long enough to be struck by the spinning vehicle and wide enough to maintain its aspect ratio (see the March 2000 Newsletter Technical Session for further information regarding aspect ratio).

Adding the Environment

Next, go to the Environment Editor and add an environment. Open the 3-D Geometry file browser and choose the geometry file we created for our environment. This 6-lane divided highway was created from a total station survey. It depicts a construction zone in which the traffic flow has been rerouted.

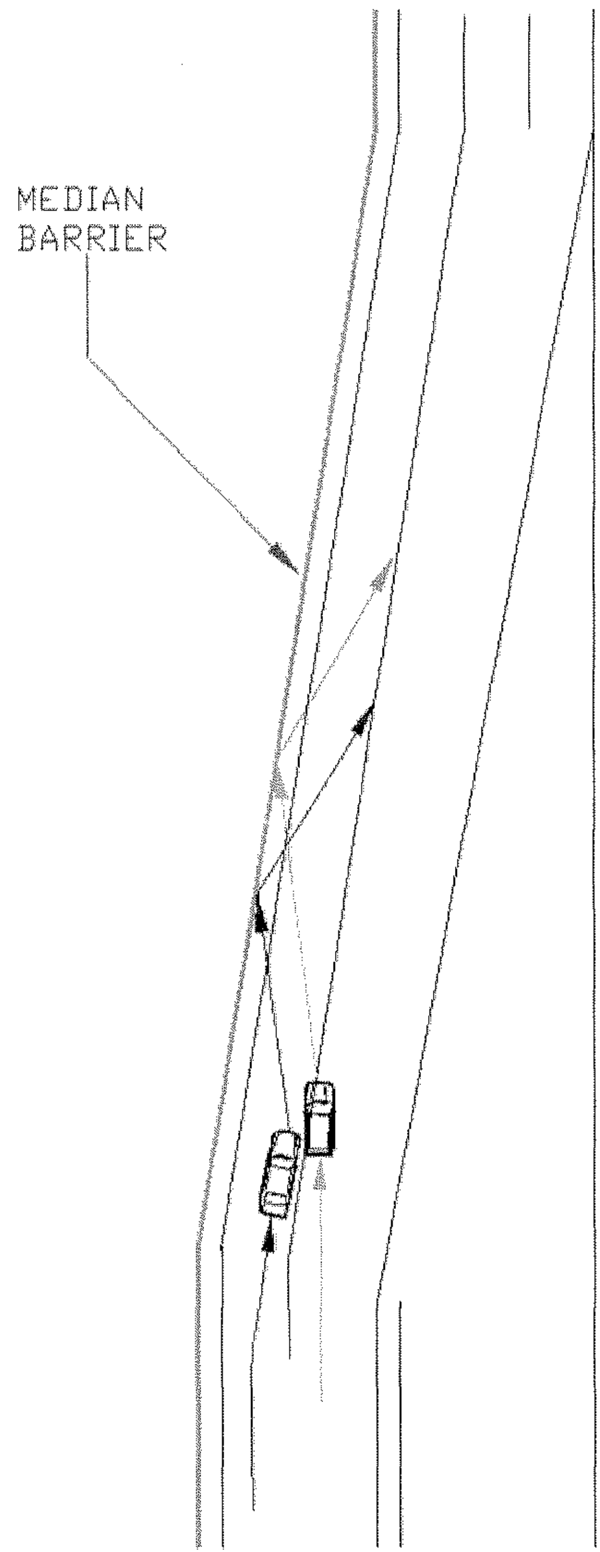


Figure 1 - Schematic of Crash Sequence

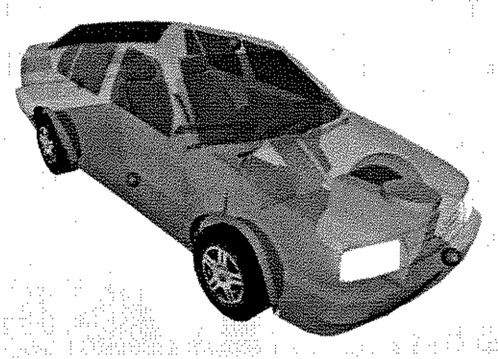


Figure 2 - 1999 Volkswagen Jetta 4-Dr

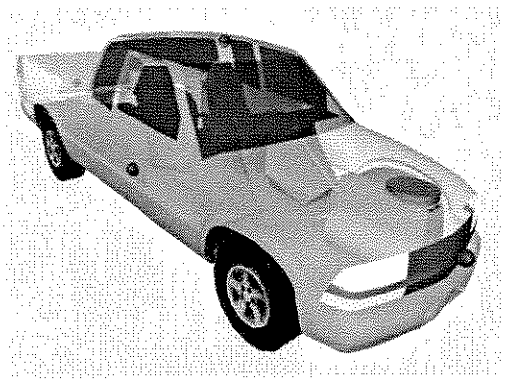


Figure 3 - Chevrolet S-10 Extended Cab Pickup

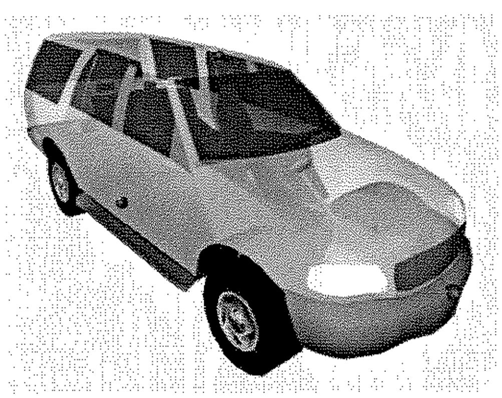


Figure 4 - Ford Expedition 4-Dr Sport-Utility

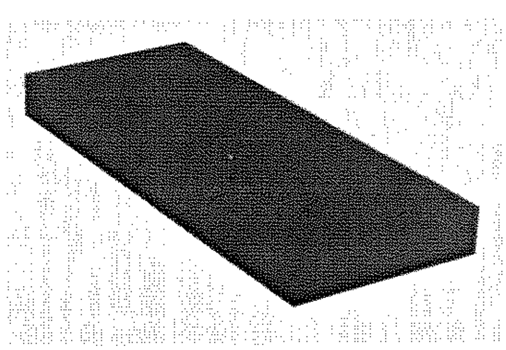


Figure 5 - SAE J850 Fixed Barrier (2)

Setting Up and Executing

Now go to the Event Editor, select *EDSMAC4*, and add the vehicles and barriers. Position each of the vehicles and barriers, and assign velocities, as shown in the following table:

Vehicle Name	X (ft)	Y (ft)	Ψ (deg)	V_{tot} (mph)
Volkswagen Jetta 4-Dr	480.2	20.4	180.0	55.0
Chevrolet S-10 Extended	512.0	9.4	180.0	65.0
Ford Expedition 4-Dr	570.0	20.0	180.0	60.0
SAE J850 Fixed Barrier	150.0	31.7	5.25	0.0
SAE J850 Fixed Barrier 2	66.5	24.1	5.25	0.0

Note the position of the barriers. Each was carefully positioned so its right side was coincident with the struck face of the median barrier (see Figure 6).

Next, assign driver controls for each vehicle. The steering and braking for the Volkswagen Jetta are as follows:

Time (sec)	Steering Wheel Angle (deg)	Braking (% Available Friction)			
		R/F	L/F	R/R	L/R
1.50	8.0	-	-	-	-
2.50	8.0	-	-	-	-
3.00	0.0	-	-	-	-
3.70	0.0	0.00	0.00	0.00	0.00
3.80	200.0	0.20	0.75	0.00	0.00

The steering for the Chevrolet S-10 is as follows:

Time (sec)	Steer Angle At Axle (deg)	
	R/F	L/F
4.0	0.0	0.0
4.1	0.0	25.0

Finally, the steering for the Ford Expedition is as follows:

Time (sec)	Steering Wheel Angle (deg)
1.5	0.0
2.5	8.0
3.5	8.0
4.0	0.0

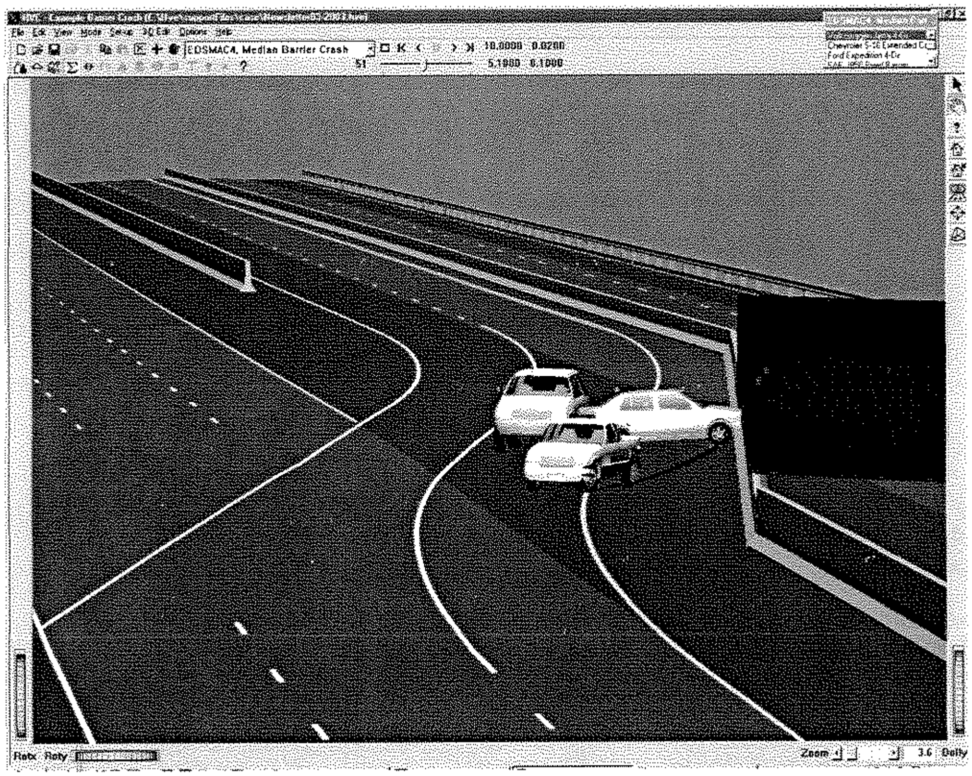


Figure 6 - Snapshot of barrier 3-vehicle barrier crash simulation. The fixed barrier is visible in this view

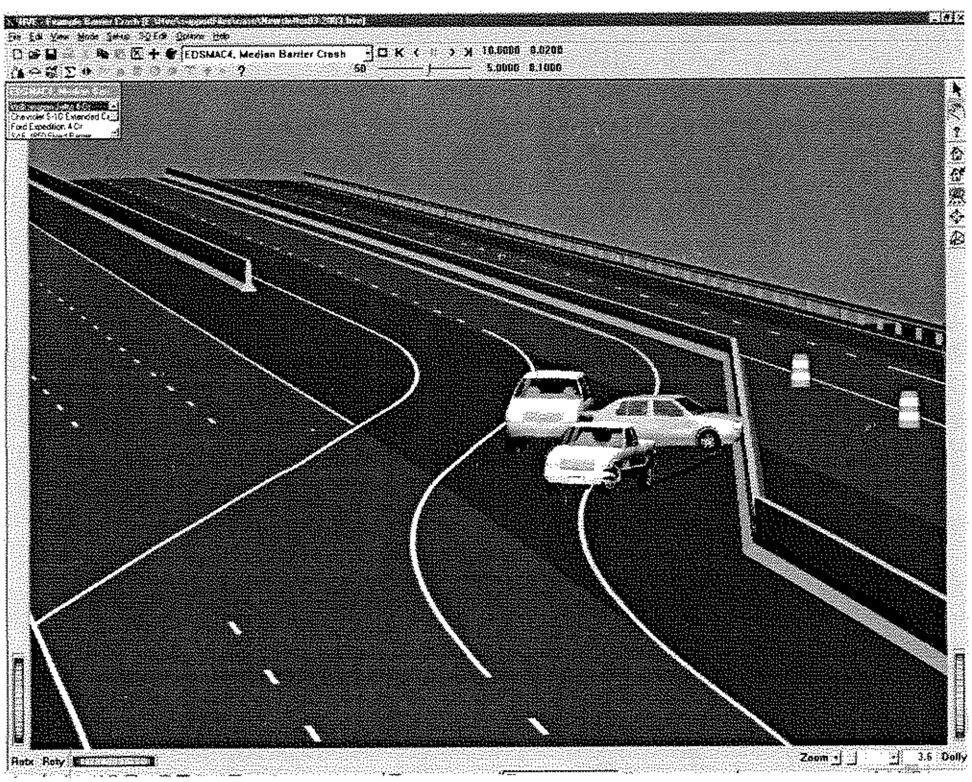


Figure 7 - Snapshot of barrier 3-vehicle barrier crash simulation. The fixed barrier's geometry has been removed in this view

Now execute the *EDSMAC4* event. A snapshot is shown in Figure 6.

But what if we don't want the Generic Barrier objects showing? After all, it's really the median barrier that was struck by the vehicles during the actual crash. There's a simple solution: Go to the Vehicle Editor. Add a Geometry File to each Generic J850 SAE Barrier, selecting NoBody.h3d (the name speaks for itself). Now, return to the Event Editor and reset the event. You'll see the Generic Barrier disappear (actually, the generic geometry has been replaced by NoBody.h3d). Execute the event. The resulting vehicle motion will be the same as before (see Figure 7). This is expected because the *EDSMAC4* collision algorithm is not using the geometry file in its calculations; the geometry is being shown for visualization purposes only.

Would you like to visualize this crash sequence? Go to www.edccorp.com/support/examples.html and use the link to download MedianBarrierCrashSimulation.mov.

If you would like to download the *HVE* case file and view and edit this crash sequence (i.e., play around – we won't tell anyone), use the links on the same web page as the movie file to download the *HVE* case file MedianBarrierCrashSimulation.hve.

Finally, if you would like a detailed tutorial describing the above process, use the link to download the file MedianBarrierCrashSimulationTutorial.pdf.

(Thanks to Baker-Sneddon Consulting, in Chicago, IL, for providing the 3-D environment geometry model.)

User Hints and Tips

This section contain useful hints and tips to help *HVE* and *HVE-2D* Users make the most of their software.

- Want to add a dimensional reference to Site Drawings you print out from the Playback Editor? Two new environment library objects have been added to Version 4.40 to provide you with a Scale (one is in feet, the other in meters). Using the 3D Editor, you can insert these objects from the Environment Objects Library (they are named Scale-Feet.h3d and Scale-Meters.h3d) onto your environment drawing or model and position them appropriately.
- When creating an event involving an articulated vehicle such as a tractor-trailer, you must add vehicles to the Event Humans and Vehicles section of the Event Information dialog in a specific order. The proper order is to select them from the Vehicle List, proceeding from the front to the rear of the combination. If you don't select the vehicles in the proper order for their connections to match up, *HVE* will issue an error message and prevent you from creating the event.
- Version 4.40 allows users to copy selected rows and columns of data directly from the Variable Output report into spreadsheet programs like Excel. If you have been using the procedure of printing the Variable Output report to a text file and then importing the text file into Excel in order to produce detailed graphs previously, you will want to try selecting the data in the Variable Output report and then using the Ctrl + C (copy) and then Ctrl + V (paste) shortcut keys to quickly achieve the same goal.

HVE and *HVE-2D* F.A.Q.

This section contains answers to frequently asked questions submitted to our Technical Support staff by *HVE* and *HVE-2D* users.

Q: I made a simulation movie of a 15 second event. I then transferred the AVI file to a CD and gave it to my client to play on his computer. When he tried to play the movie directly off the CD, the cars motions were very jerky and it did not play in real time. What happened?

A: Your movie file from a 15 second simulation is probably quite large. If you try to play it directly from the CD, the limitation of the data transfer rate from the CD to the computer can cause the movie to play slower than real time and appear to stutter. To overcome this, just have your client copy the movie to their hard drive and play the movie directly from there. The data transfer rate will be much faster and the movie will play as expected.

Q: I have made a simulation movie and I want to display it full screen on my computer. When I do play it full screen, it appears fuzzy and "blocky". I have checked the properties of the AVI file and it is only 640 x 480 resolution. Can I increase the size to better match my screen resolution, which is set to 1280 x 1024?

A: At this time, the answer is no. The size of the movie file produced from the Playback Window has been set to 640 x 480, which is formatted for NTSC Video standards. Since most presentations of simulation movies were made using VCR's, this format was suitable for a time. However, with the increase in popularity of DVD and computer based media presentations, the request for sizes other than 640 x 480 has grown. In the short term, you could try to convert the 640 x 480 AVI movie file to other sizes and formats using editing programs like Adobe Premier. However, you will most likely lose some image quality depending upon the changes you make.

Q: I have just installed the new Version 4.40 update and now when I try to run it, I get a message that reads something like "Application is Node Locked". My 4.30 software worked just fine and I've never seen this message before. What happened?

A: You must use a license file that matches the version of the software you are trying to run. If you didn't install the new license file for the update, then you are trying to run the new version with an old license. You will get the error message that the "Application is Node Locked", meaning that you don't have a proper license file and the program will only operate in Demo Mode.

Did you remember to install the new license file that was included with your Version 4.40 update? (It is on the black 3.5" diskette included with your Version 4.40 CD.)

F.A.Q.'s on EDC Website

The EDC website provides a special section dedicated to Technical Support, including answers to F.A.Q.'s from this and previous Newsletters. We encourage you to visit this page to search for answers to your questions before contacting Technical Support for assistance. Go to www.edccorp.com/support/faq.html, or just follow the links for Support from any page of the website.

Technical Session

This Technical Session describes how EDCRASH produces the diagnostic message *"You are grossly violating the conservation of energy." This message alerts the user that a difference exists between the various methods used to calculate the energy lost during the collision phase.*

In general, EDCRASH has two available methods of computing the damage energy. The first method uses the results of the damage analysis, while the second uses the loss in kinetic energy that occurs between impact and separation.

Damage Analysis

The Damage Analysis calculates the damage energy from the user-entered damage profile, stiffness coefficients and PDOF. This is done for both vehicles (unless the vehicle strikes a rigid barrier; rigid barriers are not damaged, and therefore, absorb no energy). The total damage energy, then, is simply the sum of the individual damage energies:

$$E_{Damage} = E_1 + E_2 \qquad \text{(Eq. 1)}$$

where

E_1 = Damage energy for vehicle 1
E_2 = Damage energy for vehicle 2

Loss in Kinetic Energy

If scene data are entered, EDCRASH calculates the impact and separation velocities for both vehicles. In this case, the damage energy is simply the loss in kinetic energy that occurs between impact and separation:

$$E_{Damage} = \Delta KE = KE_{Impact} - KE_{Separation} \qquad \text{(Eq. 2)}$$

where

KE_{Impact} = Total system kinetic energy before impact
$KE_{Separation}$ = Total system kinetic energy after impact (i.e., separation)

Kinetic Energy at Separation

The kinetic energy at separation is calculated from vehicle inertial properties and separation velocities:

$$KE_{Separation} = \tfrac{1}{2} m_1 V_{1,Separation}^2 + \tfrac{1}{2} m_2 V_{2,Separation}^2 + \tfrac{1}{2} I_1 \dot{\psi}_{1,Separation}^2 + \tfrac{1}{2} I_2 \dot{\psi}_{2,Separation}^2 \qquad \text{(Eq. 3)}$$

where

m_1 = Mass of vehicle 1
m_2 = Mass of vehicle 2
$V_{1,Separation}$ = Linear velocity of vehicle 1 at separation
$V_{2,Separation}$ = Linear velocity of vehicle 2 at separation
I_1 = Yaw moment of inertia of vehicle 1
I_2 = Yaw moment of inertia of vehicle 2
$\dot{\psi}_{1,Separation}$ = Angular velocity of vehicle 1 at separation
$\dot{\psi}_{2,Separation}$ = Angular velocity of vehicle 2 at separation

Kinetic Energy at Impact

The kinetic energy at impact is calculated in a similar fashion:

$$KE_{Impact} = \tfrac{1}{2} m_1 V_{1,Impact}^2 + \tfrac{1}{2} m_2 V_{2,Impact}^2 \qquad \text{(Eq. 4)}$$

where

m_1 = Mass of vehicle 1
m_2 = Mass of vehicle 2
$V_{1,Impact}$ = Linear velocity of vehicle 1 at impact
$V_{2,Impact}$ = Linear velocity of vehicle 2 at impact

Note that for an oblique collision (i.e., a collision in which the pre-impact velocity vectors are more that 10 degrees from parallel), there are available two estimates of the impact velocity.

Using Linear Momentum:

$$V_{Impact} = V_{Separation} - \Delta V_{LinearMomentum} \qquad \text{(Eq. 5)}$$

Using Damage:

$$V_{Impact} = V_{Separation} - \Delta V_{Damage} \qquad \text{(Eq. 6)}$$

Note that the only difference in these two estimates is the source of the delta-V: It may come from either the *momentum-based* delta-V or the *damage-based* delta-V. The momentum-based delta-V is calculated strictly from vehicle inertial properties and scene data (path lengths, drag factors and departure angles), while the damage-based delta-V is calculated from inertial properties, damage profile, stiffness coefficients and PDOF. In either case, the separation velocity is calculated from scene data.

Diagnostics

The EDCRASH Messages report includes a diagnostic that compares the damage energies as follows:

Damage Data:	$E_1 + E_2$
Damage and Scene Data:	$KE_{Impact_{Damage}} - KE_{Separation}$
Linear Momentum:	$KE_{Impact_{Momentum}} - KE_{Separation}$

In the above message, $E_1 + E_2$ comes directly from the damage energies, as shown in eq 1. $KE_{Impact_{Damage}}$ comes from eq. 2 and uses the damage-based delta-V's in the calculation of impact velocities (see eq. 6). $KE_{Impact_{Momentum}}$ also comes from eq. 2, but uses the momentum-based delta-V's in the calculation of impact velocities (see eq. 5). Note that the separation velocities in both $KE_{Impact_{Damage}}$ and $KE_{Impact_{Momentum}}$ use the scene data, as shown in eq. 3, to calculate $KE_{Separation}$.

In the message, ΔKE_{Damage} and $\Delta KE_{Momentum}$ are each compared with $E_1 + E_2$. If either ΔKE_{Damage} or $\Delta KE_{Momentum}$ differs from $E_1 + E_2$ by more than 50%, the message is issued (50% is the default value and may be modified by the user). Note that this value (and all similar defaults) are displayed in EDCRASH's Program Data report.

Finding the cause(s) of this message can be quite difficult. Almost every piece of user-entered data affects this message in some way. However, some things are predictable: The *Damage Data* value ($E_1 + E_2$) is determined only from damage profiles, stiffness coefficients and PDOFs of both vehicles, so focus on those areas if you suspect your Damage Data is the problem. Similarly, the *Linear Momentum* value ($KE_{Impact_{Momentum}} - KE_{Separation}$) is largely dependent upon scene data (post-impact path lengths, drag factors and departure angles), assuming your vehicle weights are correct, so focus on those areas if you suspect your scene data. The *Damage and Scene Data* value is dependent on both damage and scene information, so solving a problem in this area can be quite difficult.

For more information about solving the Conservation of Energy diagnostic message, refer to your EDCRASH User's Manual, Chapter 6, Messages.

User Hints and Tips

This section contain useful hints and tips to help HVE and HVE-2D Users make the most of their software.

- Trying to interpret the EDSMAC(4) Damage Data report? - To interpret the Damage Data report you need to know what all of the numbers are referring to. For each vehicle, the data is broken into four columns: RHOB, PSIB, X, and Y. RHOB and PSIB are distance and angle measurements in polar coordinates, meaning that they represent a length (in) and an angle (deg) from the vehicles CG to the exterior of the damaged vehicle. X and Y are measurements in Cartesian coordinates, meaning they represent distances (in) in the x and y direction from the vehicles' CG. Mapping out and overlaying the RHOB and PSIB values over the x and y values will show they represent the same coordinates. It is just another way of displaying the data. These coordinates are provided so that if it you want to plot out the damage profile, you have all the data points. To show the damage profile of the vehicle, plot the coordinates to the body of the vehicle and then connect them using lines between the coordinate points.

- Want a quick overview or refresher on how to use a particular physics program or feature of HVE (HVE-2D)? Try referring to the Tutorial sections of the User's and Physics manuals. Every physics program has a tutorial available in Chapter 5 of it's manual. (e.g. EDCRASH tutorial is found in Chapter 5 of the EDCRASH physics manual.). Tutorials for navigating the interface and using the various editors can be found in Chapter 32 of the User's Manual for your HVE or HVE-2D software. A tutorial for using the 3D Editor can be found in Chapter 23 of the User's Manual.

- Want a quick way to evaluate driver steering inputs required to negotiate a path? Try using the Path Follower. (NOTE: The Path Follower is only available as an option in Driver Controls when using EDVSM, EDVDS or SIMON.) Basically what you do is define a path of travel by positioning targets (or Path Locations) indicating the position and heading of the vehicle that you would like to try to be at that point. An obvious application is to combine the HVE Path Follower with the HVE Tire Blow-out Model to assess the driver steering inputs required to maintain control during a rapid air loss situation.

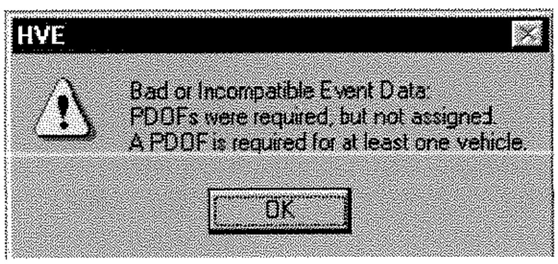
HVE and HVE-2D F.A.Q.

This section contains answers to frequently asked questions submitted to our Technical Support staff by HVE and HVE-2D users.

Q: Why is it that when I try to do a rollover in EDVSM, the body of the vehicle falls right through the environment geometry as if it wasn't there?

A: This problem is a common occurrence when running your first rollover event. To enable vehicle body to environment interaction, EDVSM requires that you activate the feature "Vehicle Body vs. Environment Contact". This feature can be activated by going to Options, Calculations Options, and then checking the box next to Vehicle Body vs. Environment Contact.

Q: Why do I get this error message asking for PDOF's in EDCRASH, when I've already entered them in the Damage Profiles dialog?



A: Chances are that when you entered in the damage profiles for the vehicles in this event, you activated the Newton's 3rd Law feature on both vehicles. Clicking on this check box causes EDCRASH to calculate the PDOF for the selected vehicle based on the vehicles' impact heading angles and the PDOF of the other vehicle. This is a convenient way to assign the PDOF for one vehicle. YOU CANNOT USE THIS CHECK BOX FOR BOTH VEHICLES!

Q: How do you position a non-moving vehicle in an EDSMAC/EDSMAC4 event without getting an "Event Termination: No Vehicle Velocities Assigned," message when you try to execute the event?

A: The key to answering this question is that you much do exactly what the message is asking you to do: You must assign a velocity to the non-moving vehicle. Since we want this to be a non-moving vehicle, all we have to do is assign the vehicles velocity to be zero (0). Then when you execute the event, the vehicle will stay put just as you assigned.

Q: When I first position my vehicle in an event, the vehicle is displayed on it's side and only shows two wheels. What's happening?

A: Somewhere in the process of editing your wheel locations, you have mixed up the right and left side tires. This means you've got the left side tires on the right side of the vehicle or vice versa. To correct this, you need to go back to the Vehicle Editor. First check the tires on the vehicle by opening the wheel location dialog for a tire, place the cursor in blue title bar area, and wait until the full name of the wheel location appears. This will tell you if you have the right tire on the right side of the car, etc.. Then enter the correct coordinates for that wheel location. Another method of correcting this problem would be to view the wheel locations in the Vehicle Data report and then go back to the Vehicle Editor and make the necessary corrections. Last but not least you can always delete the vehicle and start fresh.

Q: I used to run my HVE (HVE-2D) software using an EDKEY on a Windows 98 computer. I just upgraded my operating system to Windows XP on the same computer and now I get a licensing error message that limits me to only run in Demo Mode. It used to work before I upgraded to Windows XP, what happened?

A: HVE (HVE-2D) is probably not the only program that you will have problems running when you upgrade from Windows 98 to XP. However, we do have a solution that fixes this problem. The utility program that allows HVE to communicate with the EDKEY needs to be uninstalled and reinstalled. Use the Windows Add/Remove Programs function to uninstall the *Sentinel System Driver*. When that is complete, you can simply reinstall your HVE(HVE-2D) software, which includes the Sentinel System Driver in the installation. (NOTE: Version 4.40 includes a newer version of the Sentinel System Driver which is more compatible with Windows XP installations and may complete eliminate this "upgrade to XP" problem.

F.A.Q.'s on EDC Website

In response to suggestions made on Technical Support Feedback Surveys, we have redesigned our website to provide a special section dedicated to Technical Support, including answers to F.A.Q.'s from this and previous Newsletters. You can find this page directly at www.edccorp.com/support/faq.html, or by simply following the links for Support from any page of the website.

We encourage you to visit this page to search for answers to your questions before contacting Technical Support for assistance.

Technical Session

This Technical Session explains how the "Steer Degree of Freedom" works. First a definition: The term "degree of freedom" means that a particular motion is determined by its equations of motion using Newton's 2nd law. In the case of the steer degree of freedom, there is a single equation of motion. That equations defines the angular acceleration of a single wheel assembly about its steer axis. In Newton's 2nd law form,

$$\Sigma M = I_S \ddot{\delta}$$

where

- ΣM = summation of external moments acting about the steer axis to resist wheel steer
- I_S = rotational inertia of the wheel assembly about its steer axis
- $\ddot{\delta}$ = angular acceleration of the wheel assembly about its steer axis

Rotational Inertia

Note that we've used the term "wheel assembly". That's because we're concerned with the total rotational (angular) inertia of the wheel. That would also include the brake rotor and caliper and the steering knuckle. The rotational inertia of these objects is normally small compared to the wheel and may often be ignored.

External Moments

There are three basic sources of external moments acting on the wheel assembly. These are

- Steer torque produced at the tire-road interface
- Frictional torque in the steering system
- Steering stops

Steer torque occurs because the point of application of tire force lies a distance away from the point where the steer axis intersects the ground plane (see Figure 1). This produces a moment about the steer axis. The distance from the point of force application to the intersection of the steer axis with the ground plane is the moment arm.

Frictional torque exists in the spindle (solid axle suspension), upper and lower ball joints (upper/lower control arm-type independent suspension) or shock tower and lower ball joint (strut-type independent suspension). Frictional torque also exists in the steering gear and steering column.

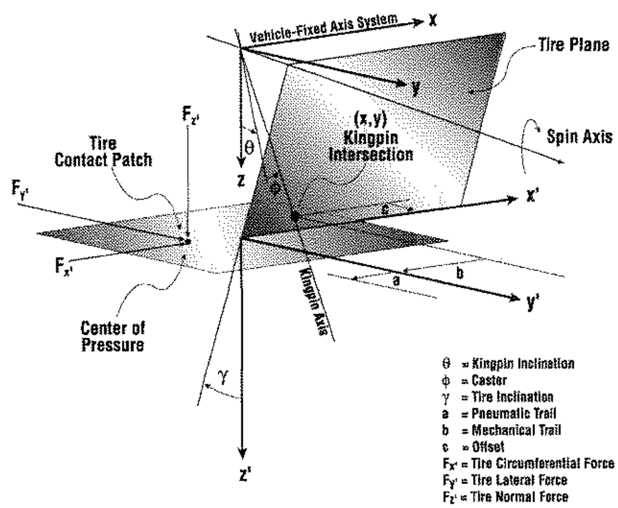


Figure 1 - Steer axis intersection with ground plane produces a moment arm about steering axis

The steering stops ultimately produce a moment that limits the left and right steer angles. While the steer torque and frictional torque exist throughout the full range of steering (lock-to-lock), moments from the steering stops exist only at the limits of the steering range, (i.e., full left or full right steering).

A mechanical model of the steering system is shown in Figure 2.

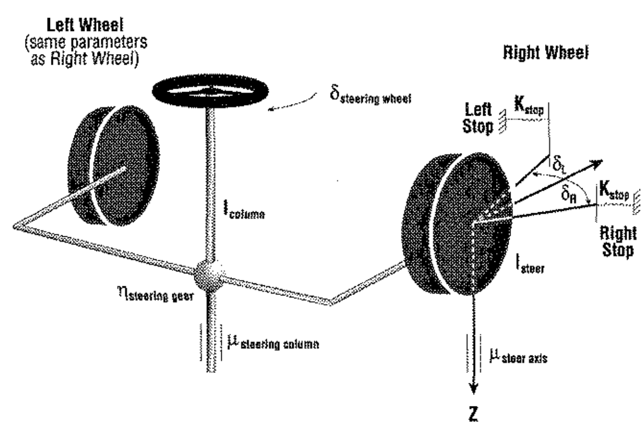


Figure 2 - Steeering system model

Angular Acceleration

Because the external moments and rotational inertia are known, we have a single equation with one unknown: the angular acceleration of the wheel about its steer axis. Thus, the angular acceleration is

$$\ddot{\delta} = \frac{\Sigma M}{I_s}$$

Application

By far the most useful application of the steer degree of freedom model is the simulation of post-impact trajectory.

Referring to Figure 1, while a vehicle is traveling straight ahead, F_x is small (existing because of rolling resistance) and F_y is approximately zero. Also, the moment arms for F_x at the left and right wheels are such that their resulting moments cancel each other. Thus, there is no significant total steering moment produced during straight ahead driving (this, of course, is a good thing!).

On the other hand, consider a car sliding sideways (as often occurs during a post-impact spinout). In this case, F_y is quite large and the moments *do not* cancel – in fact they are additive. Thus, a significant steering moment is produced while a car is sliding sideways after a collision. This is well modeled by the steer degree of freedom and can easily be visualized in either SIMON or EDVSM. To illustrate this point, set up a simulation with an initial sideslip angle of, say, 45 degrees and turn on the steer degree of freedom (Calculation Options dialog). Execute the simulation. You will see how the resulting trajectory is affected by the steering induced at the tire-road interface. You can track the actual steer angle at each wheel in the Key Results window; see Wheel Data Group, Delta). To better visualized the actual steering, you can attach the camera to the vehicle. Position the camera at a distance of about 25 to 50 feet ahead of the vehicle and looking back at the vehicle. You will see the wheels steer as the vehicle progresses towards its rest position.

The steer degree of freedom is a relatively new tool and is not yet all that popular. That should change!

User Hints and Tips

This section contain useful hints and tips to help HVE and HVE-2D Users make the most of their software.

- Need to produce a different graph of your Variable Output than just a selected variable vs time? Print the results of your Variable Output report to a text file. This will allow you to import the results into your spreadsheet program as a fixed width format text file. Refer to page 2-62 of the HVE User's Manual, or page 2-51 of the HVE-2D User's Manual for specific details.

- Want to print out a sharper, crisper image of your Traj Sim than what you are able to produce directly from using *File*, *Print* on the Main Menu? Try capturing the active window and pasting it into Microsoft Paint or directly into your Word document. This wil produce a higher resolution image that what you will produce via your printer. Plus you get the added benefit of being able to have it where you want it in your report, rather than having to just attach it at the end. You can also capture other views, such as the Event Viewer with the current simulation time and Key Results displayed, which many users like to have to present as a series of stills of their simulations. To capture the active window, press *Alt* + *Prt Sc* on your keyboard. Then simply use the paste command in your other program. To capture the whole desktop, just press *Prt Sc* and then paste it into your other program.

- Have you added a friction zone surface to your environment model and would like to have it blend in better? Try placing the surface on it's own overlay and then turn the overlay off. This will hide the surface from your view, but not from the physics calculations. The vehicles tires will still "see" the friction factor assigned to that surface, and the results will be the same whether the overlay is turned on or off.

- Having trouble matching post-crash vehicle trajectories between your EDSMAC4 simulation and the actual measurements from the crash site? It may be because of a deflated tire or a dislocated wheel during the crash. Try using the tire blowout or wheel displacement setup capabilities within EDSMAC4 to account for actual damage to the vehicles tires. This may help account for the change of tire/terrain contact conditions that frequently occur during a crash.

HVE and HVE-2D F.A.Q.

This section contains answers to frequently asked questions submitted to our Technical Support staff by HVE and HVE-2D users.

Q: Every time I open my case files, the main title bar displays HVE-2D - Untitled (filename.hve) or HVE - Untitled (filename.hve). Why does it say Untitled?

A: When you save a case, the Save As dialog appears and in this dialog there is a field for Filename and a field for Case Title. The default case title name is Untitled, and if this field is not edited the title listed above will be displayed. If you change the default case title to something like EDSMAC4 Barrier Collision, the main title bar will display:

HVE - EDSMAC4 Barrier Collision (filename.hve)

HVE-2D - EDSMAC4 Barrier Collision (filename.hve)

Q: When I start my HVE or HVE-2D program, I see a message that indcates a license file was not found and I am running in Demo Mode. I' am sure I have a license. Why am I getting this message?

A: When the software starts, the license manager checks for a valid license file. If a license file is not found or if the license file is not validated properly, then you will receive this message. If you believe you have a license properly installed on your computer, try the following steps to troubleshoot the installation:

1. The HVE (and HVE-2D) software version numbers and the license file version number must match exactly. If you just installed an update, did you use the new license diskette with the same version number as the update?

2. When you purchased your software, did you choose to lock it to your computer or did you purchase an EDKEY? If your software is locked to an EDKEY, is the EDKEY plugged into your computer? (Believe it or not, this happens quite often!)

3. If you are using a parallel port EDKEY and you have other hardware locks installed (such as AutoCAD, 3D Studio), try plugging only your EDKEY into the computer.

4. Click on Start, Programs, HVE-2D (or HVE). In the list of programs, run the program *Show Computer ID*. The first line displayed in the dialog is "Locking Code 1" and is based upon the computer hard drive serial number. The second line is the "Locking Code 2" and is based upon the EDKEY plugged into your computer.

If you have an EDKEY, you should see Locking Code 2 of 80-xxxxx. If you cannot run HVE-2D, but see 80-xxxxx displayed, then jump down to Step 6.

5. If you see Locking Code 2 of 0 - 0, then that means that the EDKEY is not being read. Follow these steps to see if the driver for reading the key is working:

A. Go to C:\Program Files\Rainbow Technologies\Sentinel System Driver and run the program SetupSysDriver.exe.

B. In the dialog that is displayed, you should see an Installed Driver Version and it should be Version 5.39 for Windows. If it does, then proceed to Step 6. If it indicates that the driver is not available, then this is your problem. You will need to run the Add/Remove Programs utility in Windows and remove the Sentinel System Driver program. Now, you will need to reinstall the HVE-2D software, as it includes the Sentinel System Driver at the end. Restart your computer after the installation. Now run the SetupSysDriver.exe program again and see if it is running now.

6. If you read theLocking code 2 as 80-xxxxx, check that the license file matches that code. Using a file browser, go to your HVE-2D (or HVE) folder and locate the file named lservrc. Open this file using Wordpad or Notepad. You should see lines at the top for your User-ID number, your company name, and also the code for your key. The 0x-xxxxx code in the license should match your 80-xxxxx code.

If you have not resolved your license problem by following these steps, please contact EDC Technical Support for assistance.

Q: How do I change the background color of my line drawing from blue to some color that more closely resembles a road?

A: If you are using a line drawing, what you see as the "background" color of the drawing is actually the Sky of the environment. If you don't want to deal with creating a surface and assigning colors in the 3D Editor, you can simply import your line drawing and set the Sky color to be the color of the "road". This can be done in the Environment Information dialog when adding a new environment. Press the button for *Sky Attributes*, then *Set Sky Color*, and then set your color using the color wheel. To get a nice Grey road click on the center of the color wheel and then drag the color intensity slider down to about 0.5.

Technical Session

One of the important results of simulating a vehicle maneuver is the presence or absence of tire skidmarks. This Technical Session describes how tire skidmarks are produced and displayed by HVE.

HVE displays the skidmarks for each tire. However, it is the physics program (e.g., EDSMAC4, SIMON) that is responsible for telling HVE when, where and how to display the skidmarks.

HVE's Role

HVE can display a skidmark for each tire on the vehicle. The width of the skidmark is equal to the nominal width of the tire, as defined by the tire size string. For example, the skidmark for a P225/60R15 tire will be 225 mm wide. The skidmark will be drawn in segments from the tire's earth-fixed X,Y,Z contact patch coordinates during the previous simulation output timestep to the X,Y,Z coordinates at the current simulation timestep. As a result of using the simulation output time interval, the skidmarks may have greater detail if the simulation output interval is reduced.

Each skidmark segment is actually a rectangle drawn slightly above the road surface (the exact elevation is user-editable; see the User Preferences dialog) in order to ensure that the skidmark is visible and not blended into the road surface geometry. (Note that the same can be said of lane striping. If your skidmark disappears beneath a lane stripe, it means that the lane stripe's elevation above the terrain is greater than the skidmark's elevation above the terrain. In this case, you should either reduce the elevation of the lane stripe or increase the elevation of the skidmarks.)

Physics Program's Role

The physics program is responsible for creating the skidmark information and passing it along to HVE. At each simulation output timestep, the physics program sends the following data (along with a lot of other data) via the output tracks to HVE:

X,Y,Z Coordinates – These are earth-fixed coordinates of the tire contact patch. These coordinates provide the start and end points for each segment of the skidmark.

Skid Flag – This value determines the visibility of the skidmark. A value of 0.0 means no skidmark is visible; a value of 1.0 means the skidmark is an opaque black line.

The current values for the tire's X,Y,Z contact patch coordinates are computed directly from the vehicle's current earth-fixed position and orientation and the vehicle's dimensions. The values are actually returned by the simulation's tire model using `GetSurfaceInfo()` (a lot has been written about `GetSurfaceInfo()`, e.g.,HVE User's Manual, SAE Paper No. 970958).

Setting the current value of the skid flag is a complex modeling issue. For example, the skid flag needs to be set when a brake is locked. It also needs to be set when the brake is off and the vehicle is cornering at an excessive rate. If the brakes are partially applied while the vehicle is cornering, the skid flag needs to be set at a different (and lower) cornering rate. Such issues consume the entire careers of engineering specialists at Goodyear and Michelin. As you might expect, each simulation program (more specifically, the simulation program's tire model) is responsible for determining the criteria used for setting the skid flag. The various methods employed by EDC's simulations are described below.

Straight-line Braking

In general, a non-steered tire begins to skid when the attempted longitudinal tire force from braking exceeds the available friction force. The resulting unstable condition quickly leads to wheel lock-up and associated tire skidding (at least on a non-ABS-equipped vehicle). For 2-D simulations (e.g., EDSVS, EDVTS, EDSMAC, EDSMAC4), this condition is as follows:

$$F_{X\,\text{Attempted}} > \mu F_Z \qquad \text{(eq. 1)}$$

Figure 1 (see Page 4) shows skidmarks from a straight-line braking simulation using EDSMAC4.

3-D simulations that employ a spin degree of freedom at each wheel can compare the current longitudinal tire slip with the characteristics of the tire's mu-slip curve (refer to Figure 1 in the December 2001 Technical Newsletter for a description of a mu-slip curve). A tire begins to skid when the current tire slip exceeds the slip at μ_p.

Steering Without Braking

It is not so simple to determine the point at which a steered tire begins to skid (technically, when the tire has a slip angle the term is *scuff*, not *skid*). Whereas skidding from heavy braking tends to occur quickly and predictably (see above), lateral tire scuffing tends to occur gradually; there exists no simple mechanism that leads to tire scuffing. The most common criteria used to determine the presence of a tire scuff is non-dimensional sideslip, $\bar{\alpha}$. This parameter is fundamental to the Fiala tire model used by EDSVS, EDVTS, EDSMAC, EDSMAC4 and EDVSM (EDVSM, a 3-D simulation, uses a somewhat extended version of the Fiala tire model). In general,

$$\bar{\alpha} = \frac{C_\alpha \alpha}{\mu F_Z} \qquad \text{(eq. 2)}$$

where

C_α = Cornering Stiffness
α = Slip Angle
μ = Tire-Road Friction Coefficient
F_z = Vertical Tire Load

According to the Fiala tire model, when $\bar{\alpha}$ equals 3.0 the tire is saturated (i.e., it cannot generate any greater side force). Thus, if $\bar{\alpha}$ exceeds 3.0, the tire model sets the skid flag to 1.0 and a skidmark is drawn by HVE (note that HVE does not distinguish between skidmarks and scuffmarks).

Figure 2 shows scuffmarks from a hi-G steering maneuver using EDSVS.

Close inspection of equation 2 reveals an anomaly: Since F_z is in the denominator, using the Fiala model predicts that a lightly loaded tire is more likely to leave a skidmark than a heavily loaded tire (given the same values for C_α, α and μ). Thus, for a vehicle negotiating a high-speed turn, the Fiala model predicts an inside tire will leave a tire mark *before* an outside (more heavily loaded) tire. This condition is visible in the EDSVS simulation in Figure 2. This is contrary to experience; outside tires usually leave marks first. This simplification means that most tire models using $\bar{\alpha}$ to set the skid flag will tend to over-predict the presence of a tire mark. Therefore EDC recommends against using the presence or absence of skidmarks produced by these models as the basis for drawing important conclusions. The skidmarks are useful, however, for revealing tires that have lost traction (i.e., the attempted steering or braking force has exceeded the available force at the tire-road interface).

The problem with using Fiala's $\bar{\alpha}$ parameter directly to set the skid flag is that, even though the tire may be saturated, it may not have enough vertical load on it to leave a black mark on the road. EDC's 3-D models (EDVSM, EDVDS and SIMON) provide an additional parameter to improve the fidelity of skidmark simulation. These models include the minimum vertical tire force required to cause a skidmark. Thus, the more lightly loaded (inside) tire will not leave a skidmark unless it also has sufficient vertical load on it.

Combined Braking and Steering

All of the above tire models employ either the friction circle (or ellipse) or a slip vs. roll-off table in the presence of combined braking and steering to further reduce the level of sideforce required to cause a skidmark.

EDC Semi-empirical Tire Model

This tire model, an extended version of the HSRI tire model developed at the University of Michigan, employs a completely different scheme to determine the presence of tire saturation. This model computes the percentage of the tire contact patch that is no longer adhering to the road surface. When the percentage falls below a threshold value (typically 25 percent) and the vertical tire load is sufficient, the skid flag is turned on. The EDVDS model uses this method to set the skid flag.

ABS Tiremarks

With the introduction of the new ABS model, EDC has developed a new skidmark model as well. A key feature of the new model is that the skidmark is no longer simply *OFF* (i.e., 0.0) or *ON* (i.e., 1.0). Instead, the skid flag can vary continuously between 0.0 and 1.0 – and the skidmark's opacity can vary continuously between 0.0 (completely transparent) and 1.0 (completely opaque). By varying the opacity, the characteristics of the tire marks become very realistic. Figures 3 and 4 show SIMON simulations of straight-line braking and a hi-G steering maneuver, respectively.

The opacity of the tire mark is determined by a 3rd-order polynomial:

$$x = \sqrt{\frac{C_4 LongSkid^2 + C_5 LatSkid^2}{2.0}} \qquad \text{(eq. 3)}$$

$$y = \frac{\min(Fz, FzMin)}{FzMin}\left(C_1 + C_2 x + C_3 x^2\right) \qquad \text{(eq. 4)}$$

where

LongSkid = Longitudinal skid criterion (%)
LatSkid = Lateral skid criterion (%)
Fz = Current vertical tire load
FzMin = Minimum Fz required for 100% opacity
C = Coefficient matrix
C_1 = Polynomial constant
C_2 = Polynomial linear coefficient
C_3 = Polynomial quadratic coefficient
C_4 = Longitudinal skid weighting factor
C_5 = Lateral skid weighting factor
x = Weighted tire mark criterion (independent variable)
y = Tire mark opacity (dependent variable)

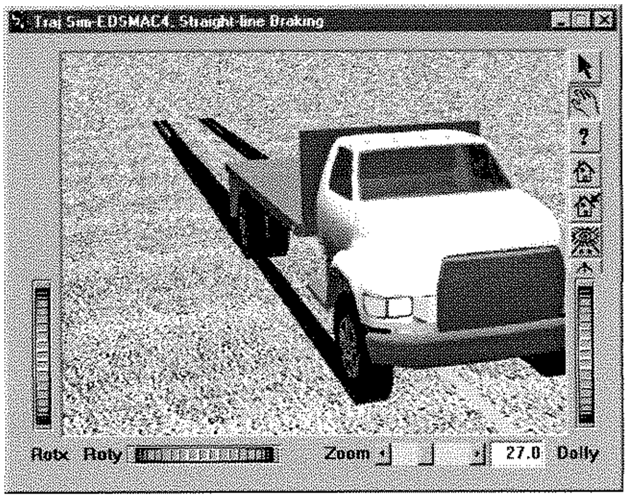


Figure 1 - Skidmarks from straight-line braking using EDSMAC4.

Figure 2 - Skidmarks from a hi-G turn using EDSVS.

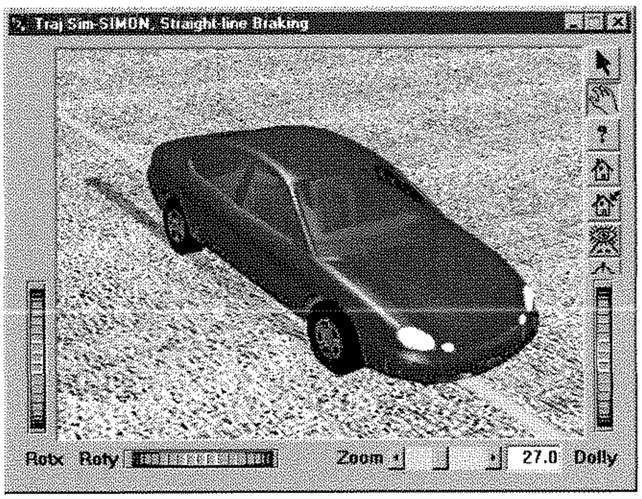


Figure 3 - Skidmarks from straight-line braking using SIMON (ABS is turned off).

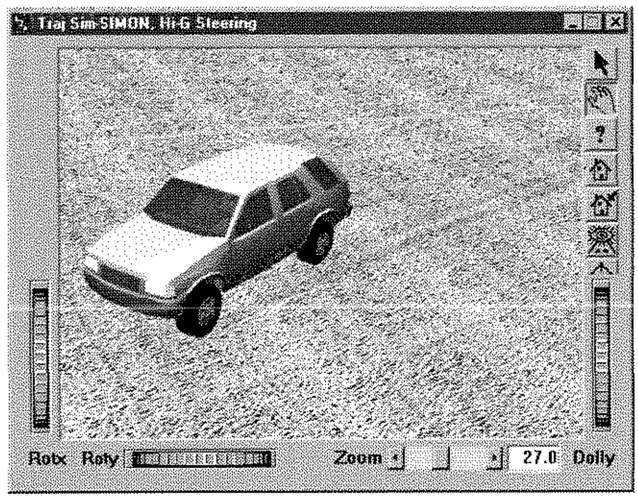


Figure 4 - Skidmarks from a hi-G turn using SIMON.

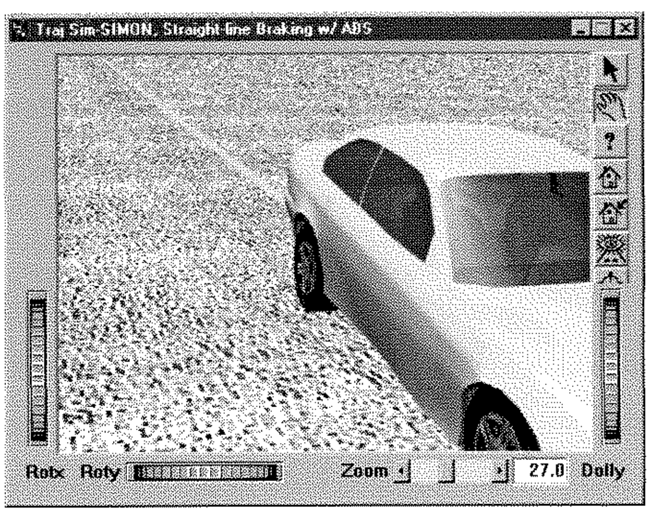


Figure 5 - Straight-line braking of an ABS-equipped vehicle using SIMON. Do you see any tire marks? They're there!

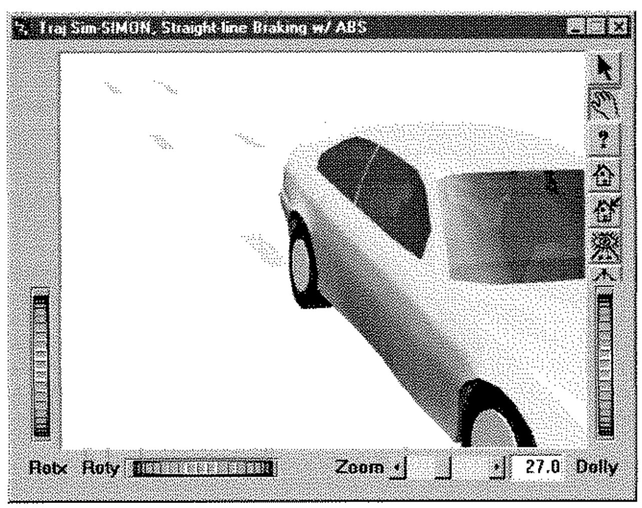


Figure 6 - Same as Figure 5, except the pavement texture has been removed, revealing the faint tire marks left by the ABS-equipped vehicle

The coefficients, C, are selected such that

$$C_1 + C_2 + C_3 = 1.0$$

and

$$C_4 + C_5 = 2.0$$

It is the responsibility of the simulation program to determine the values for LongSkid and LatSkid. These are included in the simulation's tire model. In addition, Fz is also computed by the simulation program. FzMin is assigned using the Vehicle Editor, Tire Physical Data dialog.

Simulations that use the HVE ABS model may also incorporate this new skidmark model. At this time, SIMON is the only simulation that uses the ABS model. Figures 5 and 6 show a SIMON simulation involving heavy braking of an ABS-equipped vehicle.

User Hints and Tips

This section contain useful hints and tips to help HVE and HVE-2D Users make the most of their software.

- Reversing surface normals in the 3D-Editor can be very difficult when you can't see the individual polygons making up the surface. To solve the problem ***Right click*** in the viewer, *Go to Draw Style*, and *Click on Wireframe.* This allows you to view your drawing in wireframe mode thus making it easier to reverse individual polygon surface normals.
- The latest release of HVE and HVE-2D, Version 4.30, now includes functionality for "flipping" and scaling imported DXF files (scene drawings) from CAD programs. Many users report this saves them several extra steps from the usual procedure for preparing their drawings for import into HVE. Refer to the Environment Editor section of your User's Manual for more information.
- If you need to study additional variations of your event set-up in order to conduct a sensitivity analysis, make a copy of your event and work with the copy rather than the original. It's simple to do; simply have the event you want to copy displayed in the Event Editor, then select Edit, Copy on the menu bar. You will be prompted to name the copy of the event and then you're ready to start working with the copy.

HVE and HVE-2D F.A.Q.

This section contains answers to frequently asked questions submitted to our Technical Support staff by HVE and HVE-2D users.

Q: I am setting up an EDSMAC4 crash simulation that involves a tractor-trailer. I can input braking values for the 3 axles of the tractor, but how do I input braking values for the 3 axles of the trailer?

A: The trailer is considered a "child" of the tractor "parent" vehicle. To enter braking values for the trailer axles in EDSMAC4, you will need to place the mouse cursor in Pick Mode and click on the trailer. You will see the manipulator appear on the trailer and when you choose Set-up, Driver Controls, you will see the Brake Table for the trailer displayed. When you are finished entering your values of braking for the trailer, you can add accelerometers, wheel displacements and other event-related parameters for the trailer.

Q: I notice that when I enter in values in the brake table for a 3 axle vehicle, that the values for Axle 2 are automatically applied to Axle 3. I would like to be able to simulate a situation involving only the 3rd axle braking, such as an emergency or parking brake being applied to only the 3rd axle. Can I do this?

A: The answer is *yes*, but what you have just noticed about the brake table is a temporary problem. The value you are entering for Axle 2 should not be applied automatically to, or overwrite your inputs for Axle 3. This is a bug and will be fixed in the next update.

Q: I am editing the exterior dimensions of a vehicle from the EDVDB vehicle database and I do not see the geometry file scaling to match the new positions of the red spheres indicating the exterior dimensions of the vehicle. Why?

A: The exterior geometry of a vehicle from the EDVDB database is an exact 3D model of an actual vehicle made by physically recording the vehicle using a digitizer. This geometry is not resizable and cannot be edited within the Vehicle Editor, except for changing the color of the vehicle. When you edit the exterior dimensions of the vehicle by clicking on the red spheres and changing the values, you are changing the values of the dataset for the vehicle, but you are not editing the geometry mesh. If you edit the exterior dimensions of a Generic vehicle, you will see that the generic geometry does scale to match your new dimensions. This is because the generic vehicle body is drawn by HVE.

Q: I am trying to create an EDSMAC4 event involving a vehicle towing a trailer. I am getting an error message indicating that the connect heights are not compatible between the vehicles. How do I overcome this error message?

A: With the release of HVE v4.30, EDSMAC4 now requires that the connection elevations relative to the ground to be within 1 inch of each other for connected vehicles. Here are some helpful steps to quickly check and edit inter-vehicle connection heights:

1. Display the tow vehicle in the Vehicle Editor. Click on its CG and select *Move CG* on the pop-up menu. Note the CG Height displayed at the bottom of the Move CG dialog. Close the dialog.
2. Click on the tow vehicle's CG again and select *Connections*. Note the z coordinate for the rear connection. Remember, a negative z coordinate for the connection actually places the connection above the CG of the vehicle because of the SAE coordinate system.
3. Subtract the connection z coordinate from the CG height of the vehicle. (BE CAREFUL! Subtracting a negative number is the same as adding a positive number.) The resulting value is the connection height relative to the ground.
4. Repeat the above steps for the trailer to determine the connection height relative to the ground for the front connection of the trailer.
5. Edit the connection z coordinate of the vehicle(s) to bring them into within 1 inch of each other.
6. Repeat this proceed for each set of connected vehicles.

Now you will be able to create and run your EDSMAC4 event with vehicles that are properly connected!

Q: I remember back in the EDVAP days that the manual said to select the smaller vehicle first for an EDSMAC event. Does this still hold true for the latest versions of EDSMAC and EDSMAC4?

A: The answer is *yes*. When the collision algorithm is initialized, a temporary coordinate system is established from the first vehicle viewing the second vehicle. This "view" establishes the total possible damage width on the first vehicle, and therefore, the total number of collision ("RHO") vectors. More RHO vectors means better resolution is possible in the collision algorithm, and the need for increasing the number of RHO vectors later in the collision is lessened. These RHO vectors are ultimately used to calculate the forces and moments on the vehicle, as well as to define the damage profile.

Technical Session

This Technical Session is a continuation of our last Technical Session dealing with anti-lock braking systems (ABS). Our last Technical Session provided a general overview of ABS. This time we describe the new ABS model released with *HVE* Version 4.30.

General Description

The *HVE* ABS model works like the ABS system on an actual vehicle: It uses an algorithm to modulate the brake pressure at each wheel. The algorithm is designed to maximize brake force and prevent excessive wheel slip. The ABS model has been implemented in the SIMON vehicle simulation model (see SAE Paper No. 2002-01-0559).

ABS User Interface

The *HVE* ABS user interface allows the user to select an ABS algorithm and to enter and edit the independent parameters required by the selected ABS algorithm. The interface includes numerous options, thus, various algorithms may be supported. The interface is divided into two sections:

System Variables – Variables that are applicable to the entire vehicle

Wheel Variables – Variables that are applicable to (and may be specified independently for) each wheel

The interface, dialogs and associated variables are described below.

System Variables

The ABS System Data variables included in the *HVE* ABS model are presented in the ABS System Data dialog. The variables and a brief description are shown in Table 1. The ABS System Data dialog is shown in Figure 1. (A new chapter in the *HVE* User's Manual describes each of these parameters in detail.)

Table 1 – ABS System Variables

Variable	Description
Algorithm	ABS algorithm selected from a list of available algorithms
Control Method	ABS control method selected from a list of available control methods
Cycle Rate	Sets the time required for a complete ABS cycle
Threshold ABS Pressure	Minimum pressure for ABS activation
Threshold ABS Velocity	Minimum vehicle velocity for ABS activation
Friction Threshold	Tire-terrain surface friction threshold
Delay Method	Delay method selected from a list of available delay methods
Apply Delay	Time delay for controlled output pressure increase
Release Delay	Time delay for controlled output pressure release

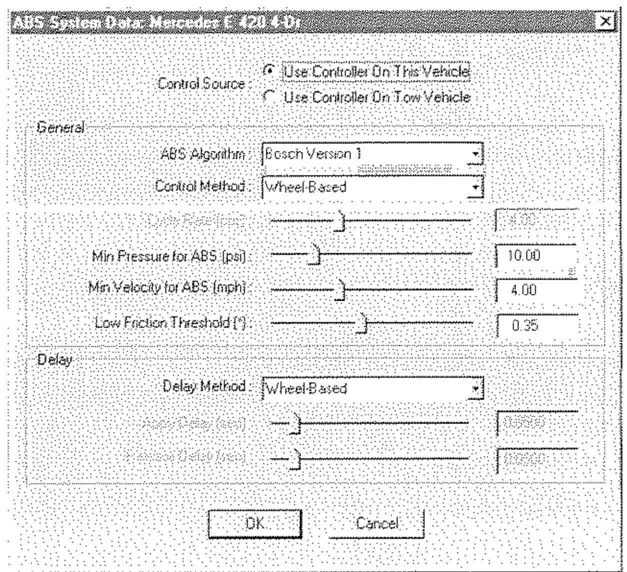


Figure 1 – ABS System Data dialog

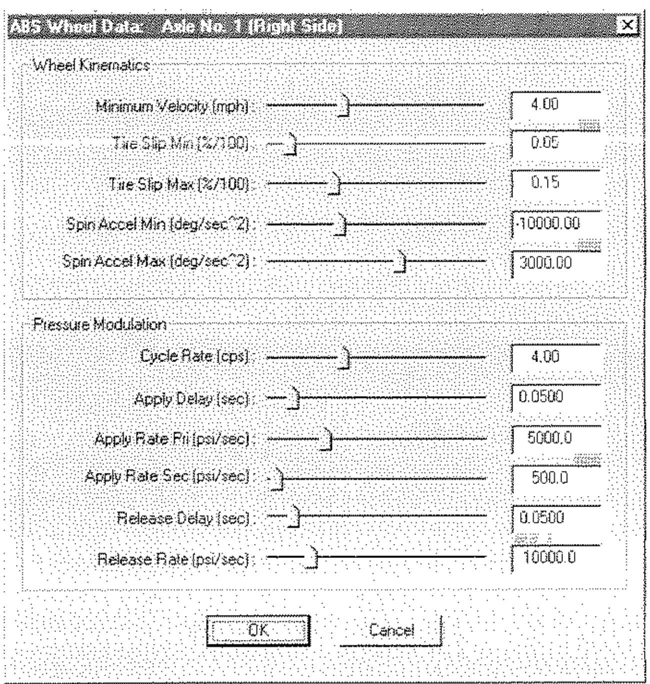
Wheel Variables

Wheel variables are those ABS parameters that are assigned independently for each wheel. A brief description of the ABS Wheel Data variables included in the *HVE* ABS model is provided in Table 2. The ABS Wheel Data dialog is shown in Figure 2. (Again, the manual provides a detailed description.)

Table 2 – ABS Wheel Variables

Variable	Description
Cycle Rate	Sets the time required for a complete ABS cycle
Threshold Wheel Vel	Minimum wheel forward velocity for ABS control
Tire Min Slip	Minimum tire longitudinal slip for ABS control
Tire Max Slip	Maximum tire longitudinal slip
Wheel Min Spin Accel	Minimum wheel angular acceleration
Wheel Max Spin Accel	Maximum wheel angular acceleration
Apply Delay	Time delay for controlled output pressure increase
Primary Application Rate	Initial rate of controlled output pressure increase
Secondary Application Rate	Initial rate of controlled output pressure increase
Release Delay	Time delay for controlled output pressure decrease
Release Rate	Rate of controlled output pressure decrease

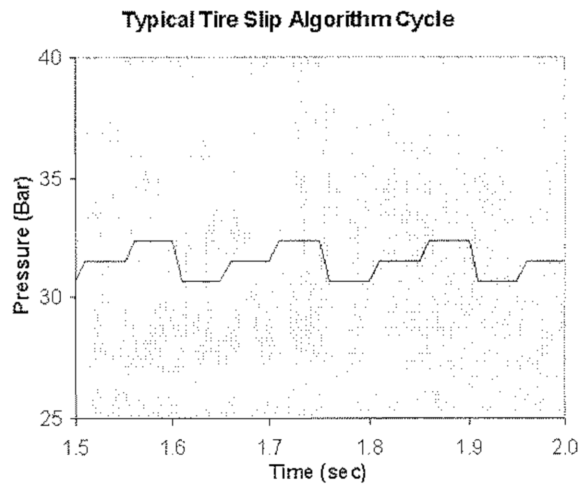
Figure 2 – ABS Wheel Data dialog



Current Algorithms

Two ABS algorithms are currently implemented in the *HVE* simulation environment. These are the *Tire Slip* algorithm and the *HVE Bosch Version 1* algorithm. These algorithms are described below.

Figure 3 – Typical Pressure vs. Time History



Tire Slip Algorithm

This is a simple and straight-forward ABS algorithm. Its design is based on the fundamental goal of an ABS system, that is, to maintain tire slip in the vicinity of peak friction coefficient, μ_p (refer to Figure 1 in the December 2001 Technical Session). It is generally applicable to any type of vehicle (passenger car, truck, etc). Figure 3 shows a typical pressure vs. time history for a few cycles of a hard brake pedal application (i.e., enough system pressure to lock the tires).

HVE Bosch Version 1 Algorithm

The *HVE* Bosch Version 1 ABS algorithm is based on the information provided in Bosch technical literature[1]. The Bosch ABS system is used on many US and foreign passenger cars. The algorithm is based on wheel spin acceleration and a critical tire slip threshold. Figure 4 shows a typical pressure vs. time history for a single cycle of a hard brake pedal application (i.e., enough system pressure to lock the tires).

Other ABS Algorithms

The ABS model implemented in *HVE* is not restrictive in terms of the algorithms it can support, other than its need to provide the parameters required by the algorithm. Endless tweaking of an algorithm is possible, resulting in different ABS system characteristics, each with its advantages and disadvantages. Thus, it is certain that new ABS algorithms will be developed and implemented in *HVE* over time, both to develop and to model new ABS systems.

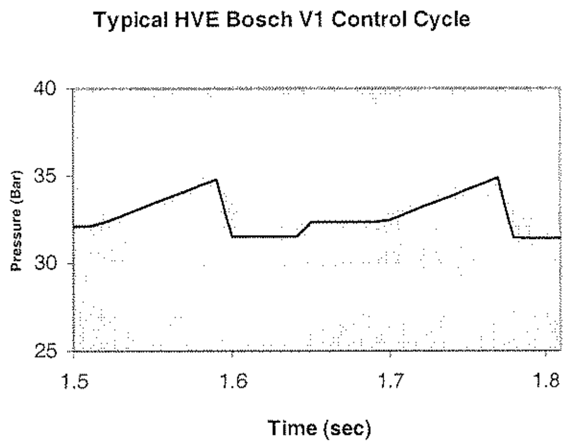


Figure 4 – Typical Pressure vs. Time History

Using the ABS Model

To set up and execute simulations using the new ABS model is quite easy. The Brake System dialog has been extended to include an "ABS Installed" check box. Simply click on this box and ABS simulation will be performed when a SIMON simulation is executed.

The effects of the ABS simulation are best observed in the *Wheel Brake Pressure* and *Tire Longitudinal Slip* results in the Key Results window or Variable Output table.

Tire Marks

To further enhance the results from an ABS simulation, *HVE* now includes a new method for displaying tire marks. In the new method, the opacity of a tire mark is varied between 0 and 1 (0 being transparent and 1 being opaque) according to the current vertical tire load (heavier tire loads produce darker marks), the percentage of longitudinal tire slip and the percentage of lateral tire slip (increased slip produces darker marks). In addition, a set of weighting coefficients determines the opacity of tire marks during combined braking and steering. The figure on the front page of this Newsletter shows an example of a vehicle pumping its brakes in a high-speed turn.

Since the vast majority of vehicles are now fitted with ABS, the new *HVE* ABS model is an important new feature, especially for simulating pre-impact braking and loss of control. Enjoy!

HINTS AND TIPS

- **Want to be more adept at navigating physics tutorials?** Following the tutorials for the different physics models can be frustrating if you are not already familiar with the *HVE* interface. To become familiar with the *HVE* interface refer to the Tutorial section of your Operations manual. This tutorial will give you the tools to navigate *HVE* with ease and efficiency.

- **Review Your Data!** Because *HVE* is a general purpose simulation environment, it supports numerous features not included in some simulation models. For example, the *HVE* Tire Data dialogs include load- and speed-dependent values for friction, cornering stiffness and camber stiffness. EDSMAC and EDSMAC4 do not have load-dependent tire models, so much of this data is not used in an EDSMAC or EDSMAC4 simulation. Here's the point: Always review the Vehicle Data output report to confirm that EDSMAC4 (or whatever model you are using) is using the expected data. Do this early in your analysis; it would be a shame to work for several days fine-tuning your simulation, then discover you were using the wrong cornering stiffness!

- **Driver Controls for 3-D Simulations.** 3-D simulations (EDVDS, EDVSM, and SIMON) all include a complete brake system that models brake torque according to the pedal force, proportioning values, lag and rise times, temperature, wheel brake assembly components (if the Brake Designer is used) and, possibly, ABS. In order to include the effects of the brake system on the simulation results, the default method for driver brake table is *Pedal Force.* Although the *Percent Available Friction* and/or *Wheel Force* driver brake tables are enabled, their use is discouraged because these methods bypass the effects of wheel spin inertia on the current level of braking force. The same is true for Throttle Tables. 3-D simulations include a Drivetrain model that calculates drive torque from the engine power curve and current gear ratios. The use of *Percent Available Friction* and/or *Tractive Effort* tables bypasses the effects of drivetrain inertia on the current level of tractive effort at each drive wheel.

- **Use Antialiasing!** Antialiasing is used to reduce the "jagged lines" resulting from the discrete pixel size on your computer monitor. It is especially useful when creating videos or printing graphical reports (e.g., Trajectory Simulations, Damage Profiles). To use Antialiasing, click on *HVE*'s Options menu and choose *Render.* Set the Antialiasing level to 10. Do this just prior to printing or creating your AVI movie file. When finished, then reset Antialiasing back to 1 (Antialiasing increases rendering time significantly. You can work more quickly – and antialiasing is unnecessary – during the normal course of your analysis.)

- **Dual Processors.** *HVE* is not supported on computers using dual processors. If you wish to run *HVE* on such a computer, here's how:

  1. Start *HVE.*
  2. Launch Task Manager (press ctrl-alt-delete on your keyboard)
  3. Select the Processes list.
  4. Locate and select the *HVE*.exe process using the left mouse button.
  5. Click the right mouse button. On the pop-up menu, select *Set Affinity.* On the Set Affinity dialog, you will see a list of all the processors that the process is assigned to, typically CPU 0 and CPU 1.
  6. Deselect one of the processors, most likely CPU 1.

  Now you can run *HVE* on your dual processor computer! You will need to do this each time you start *HVE* (or *HVE-2D*)

  Thanks to Chuck Rogers of Friedman Research for providing this feedback to EDC!

- **Layers and Objects.** If you are creating an environment model or scene drawing in AutoCAD and want to retain the ability to edit individual objects in the 3-D Editor, you must place each object on a separate layer. For example, every line or surface you create that is a piece of the road will be assigned to the road layer. Every piece of the sidewalk should be assigned to the sidewalk layer. This will ensure that the road and sidewalk will be separate objects in *HVE* and *HVE-2D*, thereby making it easier to edit the drawing later.

HVE and *HVE-2D* FAQ

This section contains answers to frequently asked questions submitted to our Technical Support staff by *HVE* and *HVE-2D* users.

Q: I've been able to make real-time simulation movies using the Playback Window, but how do I make a slow motion movie?

A: At this time, the movies created using *HVE* or *HVE-2D* will only play in real-time. If you change the output time interval in the Playback Editor from the default of 0.0333 (1/30 of a second) to 0.0167 (1/60 of a second), twice as many "frames" of the movie will be recorded which provides greater resolution of the motion, however the movie will still play back in real-time. The movie player will play movies recorded with 30 frames per second at 30 frames per second and movies recorded at 60 frames per second at 60 frames per second.

The solution is to use a non-linear editing program such as Adobe Premier to modify the Clip Speed Rate of the movie file. You would simply import the avi file, adjust the Clip Speed to 50% and the movie would then display the motion of the 60 frames per second movie at 30 frames per second. You have just made a slow-motion movie of your simulation!

Q: How can I get my HVE or HVE-2D simulation movies onto videotape, if my computer doesn't have a video out capability already?

A: One method is to hire a local video production company to make the videotape for you. Typically you would provide them with the avi files produced using *HVE/HVE-2D*, the wording for any title slides and a "story board" of the sequence of movie files and transitions. They would use a combination of off-line and on-line editing to produce the completed videotape. This is exactly how the *HVE* Demo Video was created. If you would like a copy of the *HVE* Demo Video as an example, please contact EDC Customer Service.

Another method is to purchase a "video-out" device for your computer. Several users have recommended the Pinnacle DV500 Plus, as it includes the video out card for your computer as well as Adobe Premier 6 for editing your movie files into a complete sequence. For more information, contact your local computer store, or visit the Pinnacle Systems website at www.pinnaclesys.com

Another option is to not produce a videotape of your work, but rather a CD or DVD. The avi files of your simulation movies can be integrated directly into Powerpoint presentations or just "burned" directly onto a CD.

The hardware for producing a DVD containing all of your simulation movie files, title slides and other information is continuously improving. We know of one user who is presently providing his clients with DVD's and finds it very professional to be able to quickly access any simulation movie, play it forward or backward, and even present a crystal clear still frame image from the movie.

Q: What are the recommended steps to follow to prepare my environment model/scene drawing built in AutoCAD for importing as an environment geometry in HVE and HVE-2D?

A: Here are the recommended steps to prepare your model/drawing in AutoCAD:

1. Make sure the scale of the model/drawing is in the units of Decimal Inches.

2. *HVE* recognizes the origin (0,0,0) of the World Coordinate System (WCS) in AutoCAD as the origin of the model/drawing when it is imported. If (0,0,0) is not located on your model/scene in a convenient reference location for your simulation work, you should move your model/drawing in AutoCAD so that it is.

3. Rotate your model/drawing in AutoCAD 180 degrees about the X axis.

   (For a 2D line drawing, this can be done by using the Mirror command in a Top view. Be sure to draw your mirror line along the X axis.)

4. Finally, export your model/drawing as a R12, R13, R14 or 2000 dxf format file.

To better understand the behaviors of common linetypes and orientations of drawings imported into *HVE* from AutoCAD, view the Sample.dxf file located in the supportFiles/images/environments folder. You can open this file in *HVE* as an environment and in AutoCAD as a regular dxf file.

Learn more about preparing models/scenes by attending the workshops on Environment Modeling at the 2002 *HVE* Forum!

Technical Session

This Technical Session deals with anti-lock braking systems (ABS). The vast majority of vehicles are now fitted with ABS. The purpose of ABS is to allow the driver to maintain directional control of the vehicle during maneuvers that involve heavy braking. Thus, accurate simulation modeling of these maneuvers requires the effects of ABS to be included in the analysis. In this Technical Session we provide an overview of ABS. Our next Newsletter will address the issue of simulating an ABS system.

Overview of ABS

The basic concept behind ABS is quite simple and can be demonstrated by the graph of normalized braking force vs. longitudinal tire slip shown in Figure 1. This graph is traditionally called a *mu-slip curve*. It defines the relationship between longitudinal tire slip and the available longitudinal (braking) force. A key observation is that the maximum braking force occurs at μ_p (peak friction coefficient) in the vicinity of 10 to 15 percent longitudinal tire slip (this varies somewhat from tire to tire). Also, as the tire slip continues to increase to 100 percent, the available braking force falls off. The region of tire slip between μ_p and μ_s (slide friction coefficient, or 100 percent longitudinal slip, associated with locked-wheel braking) is a region of dynamic instability. As slip begins to increase beyond μ_p it quickly increases to 100 percent (i.e., the tire locks) with a commensurate loss in available braking force.

The goal of an ABS system is simply to prevent the tire slip from increasing significantly beyond μ_p – regardless of how much brake pressure is applied. By limiting longitudinal slip, the tire continues to roll and, therefore, maintains directional control capability (i.e., the driver can steer the vehicle). In addition, as shown in Figure 1, the available braking force is larger than for a locked tire and, therefore, braking distance can be reduced.

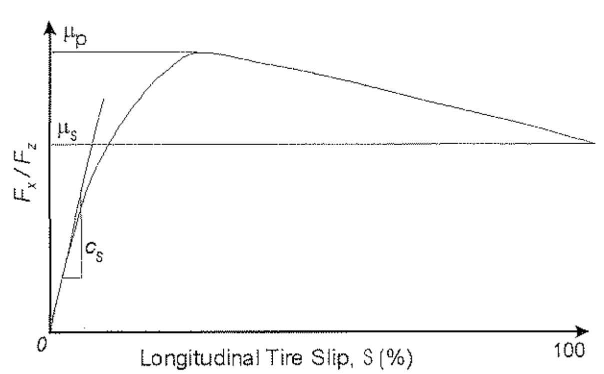


Figure 1 – Mu-Slip Curve

EDC introduced its first ABS model in 1984. The EDSVS and EDVTS programs allowed the user to specify the effectiveness of the ABS system by assigning a single value (for each wheel) that defined the percentage of increased braking force associated with the difference between μ_p and μ_s. This approach works for simulations that simply require a user-entered braking force at the tire-road interface. However, ABS simulation benefits greatly when used in conjunction with 3-dimensional models (e.g., EDVSM, EDVDS and SIMON) in which the wheel spin accelerations are actually calculated from equations of motion, rather than simply specified as a braking force at the tire-road interface. To truly *simulate* ABS, the ABS simulation algorithm must modulate the brake pressure at each wheel cylinder or brake caliper (just as on a real vehicle). This modulated pressure is then used to calculate the wheel brake torque that, in turn, is used to calculate the wheel spin acceleration. The vehicle simulation model then integrates the wheel spin acceleration to calculate wheel spin velocity. This spin velocity is used to calculate the longitudinal tire slip (see Equation 1, below) that determines the braking force at the tire-road interface (whew!). The major improvement using this approach is that the resulting steering and braking forces are algorithm-driven (just as they are on a real vehicle), and are dependent upon the current conditions existing at the tire-road interface.

Before going into the details of ABS simulation, it will be helpful to describe the basic methodologies currently used on vehicles fitted with ABS.

ABS Methodologies

All ABS methodologies work by controlling longitudinal tire slip. This is accomplished through the use of wheel sensors that compare the tire circumferential velocity to the current reference velocity, V_r, normally calculated using the current spin velocities of two or more wheels (see Reference 1 for a detailed discussion of the calculation of reference velocity). On the vehicle, tire slip cannot be measured directly. Instead, it is calculated:

$$Slip = \frac{V_r - \dot{\Omega}_w \times R_{tire}}{V_r} \qquad \text{(Eq. 1)}$$

where

V_r = Reference velocity

$\dot{\Omega}_w$ = Wheel spin velocity

R_{tire} = Tire rolling radius

State Variables

To accomplish the required control of longitudinal slip, the following state variables are monitored or calculated by the vehicle's ABS control module:

Vehicle Velocity - Linear velocity of the vehicle sprung mass

Wheel Spin Velocity – Angular velocity of each wheel

Tire Longitudinal Slip - Relative velocity between the tire and road, expressed as a fraction of vehicle velocity

Wheel Spin Acceleration - Angular acceleration of each wheel

Tire-Road Surface Friction – Ratio of the maximum braking force to the normal tire force

Brake System Pressure – Pressure produced as a result of brake pedal application (input variable)

Wheel Brake Pressure – Pressure supplied to the wheel brake assembly (output variable)

Typical Hardware

To monitor or calculate the above state variables, the typical vehicle ABS system includes the following hardware components:

Electronic Control Unit (ECU) – This is the vehicle's microcomputer. It is programmed with the algorithm that reads the current state variables, determines the required pressure at each wheel and sends the appropriate signals to the brake pressure modulator (see below).

Wheel Speed Sensors – These components directly measure the wheel spin velocity of each wheel using a wheel-mounted pulse rotor (a notched metal ring) and a fixed, magnetic sensor that measures the rotation of the pulse rotor.

Brake Pressure Modulator – This component (or components, depending on the system) controls the wheel brake pressure according to the control conditions specified by the ECU.

Brake Master Cylinder/Air Compressor – This component provides the fluid pressure source.

Wheel Brake Caliper/Cylinder/Chambers – These components apply the braking force at each wheel according to the wheel brake pressure.

The basic hardware requirements are the same for all vehicle types, ranging from passenger cars to on-highway trucks. Reference 1 provides a detailed description of these required components.

In our next Technical Session we will address the issue of simulating the effects of ABS on the wheel spin equations of motion, and thus, on the handling behavior of vehicles equipped with ABS.

USER HINTS AND TIPS

- **Need to change the background color behind a line drawing you have imported as your environment geometry?** If you're working with a basic line drawing in HVE-2D and don't want to deal with creating polygon surfaces and assigning colors in a CAD program, you can simply import your line drawing into the Environment Editor and set the Sky color to white, black or even the color of the road. This can be done in the Environment Information dialog. Click on *Sky Attributes*, then on *Set Sky Color*. Now, set your background color using the color dialog. To get a nice gray road, click on the center of the color wheel and then drag the color intensity slider down to about 0.5. This will give you a gray background for your line drawing.

- **Want your simulation to run faster while you are working to refine your event?** Redrawing the simulation results at each timestep is one controlling factor that can be adjusted. To do so, go to the *Options* menu and choose *Render* and set *Render Method* to *Wireframe*. (Or, click the right mouse button and choose *Draw Style – wireframe.*) Your humans, vehicles, and environment will be displayed as wireframe models and the simulation will execute faster because of the decreased rendering time.

- **Trying to simulate a vehicle crashing into a building in your environment model, but finding that the vehicle crashes right through the wall of the building and keeps on going?** It may be because collisions only occur between vehicles in HVE (and HVE-2D), and you have not provided any type of a "vehicle" to represent the wall of the building. EDSMAC4 allows the user to have vehicles with wheels (e.g. Passenger Car, Truck types) and also vehicles without wheels (i.e. Fixed Barrier type) in the event. When building your EDSMAC4 case, be sure to add the vehicles involved, plus a fixed barrier such as the SAE J850 barrier. When setting up the EDSMAC4 event, position the fixed barrier to correspond with the wall of the building that the vehicle crashes into. Now when you run your EDSMAC4 event, you will see the vehicle crash into the wall and have simulated damage as expected.

HVE and *HVE-2D* FAQ

This section contains answers to frequently asked questions submitted to our Technical Support staff by HVE and HVE-2D users.

Q: *I am trying to simulate a vehicle accelerating from a complete stop. When I set the initial velocity of the vehicle to 0 mph and run the simulation, I am seeing unexpected results. Why?*

A: Simulating a vehicle with zero velocity is difficult using any simulation. The problem stems from the way the tire model calculates forces, especially lateral forces. Small rounding errors occur, resulting in a non-zero sideslip angle. This, in turn, causes a large lateral force to be simulated. To avoid this problem, simulation programs use a set of non-zero velocities (usually about 2 mph linear and 5 deg/sec angular) as a threshold below which the simulation terminates. If you need to simulate a vehicle accelerating from a stop, use a small initial velocity of about 0.5 mph.

Q: *Is EDCRASH suitable for reconstruction of low speed impacts?*

A: In the EDC Reconstruction course, we teach that the damage analysis is best suited to collisions with delta-V's between about 10 and 40 mph. The reasons are as follows:

1. Below 10 mph, there is normally very little damage (sometimes none), so it is not possible to determine the damage width (a required input).
2. Restitution is ignored by the Damage Analysis. Below 10 mph, restitution may be significant.
3. The Damage Analysis assumes a linear force vs crush relationship. This assumption is not good for high-speed impacts wherein structural disintegration results in lowering of vehicle stiffness. There is no magic point at which this begins to occur, but we feel the computed delta-V begins to over-estimate the actual delta-V for delta-Vs above about 40 to 50 mph.

For more information, make plans to attend the EDC Reconstruction course. Course dates are listed on the back page of this newsletter.

Q: *I have created a scaled drawing of my crash site and saved it as a .dxf format model in Autocad. How do I get them into HVE (or HVE-2D)?*

A: There is a translator to convert your .dxf file drawings to a HVE (and HVE-2D) compatible format available within Version 4.20. First, be sure that you have set the scale to be decimal inches and that it is oriented properly (remember the SAE coordinate system?). Then, when selecting a geometry file to open in the Environment Information dialog, set the *Files of type* option list to be *DXF Files.* You can now select files with the extension .dxf and open them in the Environment Editor. The translator does it's conversion in the background and provides you with an HVE format model. You can then work with the model as is, or edit this model using the 3-D Editor. There is an example DXF file included in the Version 4.20 release named *sample.dxf* containing a variety of objects and colors

Q: *I am installing HVE-2D Version 4.20 on my computer. Will it overwrite my old HVE-2D Version 1.33?*

A: No. HVE-2D Version 4.20 installs as a new program on your computer. (Remember that HVE-2D Version 4.20 is more than just an update; it is truly a completely redesigned program!) Your old HVE-2D is not removed or overwritten by the installation of Version 4.20 because you may need to refer to your old HVE-2D for existing case files. These old case files can not be imported directly into the new HVE-2D. If you want to reproduce an old case in the new HVE-2D, simply print out all of the output reports from the old case and use the data as inputs for the new case.

Q: *I am trying to use HVE-2D Version 4.20. I have my old EDKEY plugged into the parallel port, but when HVE-2D starts, it says it is running in Demo Mode. Why?*

A: HVE-2D Version 4.20 uses the latest technology in software licensing. As such, it requires the use of a new EDKEY, rather than the one used for your old HVE-2D. These new EDKEYS are more robust, and they do not have conflicts with the latest computer hardware and operating system configurations. If you need a new EDKEY for running HVE-2D Version 4.20, please contact EDC Customer Service and order one.

Q: I can run HVE (or HVE-2D), but when I try to create an Event, I receive an error message indicating a "Socket Error" has occurred. What is the cause of this error?

A: The socket error message is displayed whenever HVE and the selected calculation method (e.g. EDSMAC4) are not able to pass data between each other. We have discovered four potential causes of "socket errors"

1. Your computer does not have enough memory available to support the processes. Either shut down other open programs or boost your computer memory to a minimum of 128 MB of RAM.
2. You do not have TCP/IP protocols setup on your computer because you do not use a dial-up modem or network connection. If you can browse the internet, you probably have the TCP/IP protocol available. To see if you have TCP/IP protocols setup, check the network settings using the Windows Control Panel on your computer.
3. You have a program called Webcelerator installed on your computer. You probably downloaded this program thinking that it would make your internet work faster. However, it does things behind the scene that totally disrupts the HVE communication process. Uninstall Webcelerator and your HVE (or HVE-2D) will work without issuing a socket error message.
4. You are using McAfee Personal Firewall (MPF) and it is not allowing the HVE communication process to proceed. You need to adjust your MPF configuration settings to allow the HVE process, as well as several other IP processes, such as fragmented packets. If you have MPF, you may want to contact their Technical Support for more information about adjusting your configuration settings.

Technical Session

HVE allows tremendous flexibility for the user to control both the HVE-compatible physics simulation or reconstruction and the presentation of results from these analyses. Many of these options are found in the Simulation Controls or Calculation Options dialogs. It is important to understand the functions of these variables. In this article we take a close look at an often overlooked, yet vital, parameter: Output Time Interval.

In order to demonstrate the importance of Output Time Interval, a simple EDSMAC4 event was created. A passenger car and a pickup truck collide in an oblique collision. We are particularly interested in the collision pulse, so let's take a close look at the output.

Critical data are located in the Accident History, Damage Data, and Variable Output output reports. The Accident History output report reports T_{impact} = 0.03 sec and $T_{separation}$ = 0.13 sec, resulting in a total collision pulse duration of 0.10 seconds (Table 1). In the Damage Data output report, Peak Acceleration for the passenger car is 40.7 g at time = 0.084 sec. These values for peak acceleration and pulse duration seem quite reasonable, but, when we graph Total Acceleration from the Variable Output Table, it looks like the pulse duration is 0.23 sec and peak acceleration is just over 10 g at time = 0.1 seconds (Fig. 2). How do we reconcile this apparent inconsistency?

Table 1: Comparison of alpha-numeric output to Variable Output graph

Variable	Alpha-numeric	Graph
T_{impact} (sec)	0.03	0.00
$T_{separation}$ (sec)	0.13	0.23
Duration (sec)	0.10	0.23
A_{max} (g)	40.7	10

The key is in choosing an appropriate Output Time Interval. The Output Time Interval is set in the Simulation Controls dialog, under the Options menu. This parameter controls the time interval at which physics, in this case EDSMAC4, reports results back to HVE. Unlike the other time intervals that are set in this dialog, the Output Time Interval does not affect the calculations within the simulation itself.

If we accept all of the default values in the Simulation Controls dialog, the Collision Time Interval = 0.001 sec and the Output Time Interval = 0.10 sec. Therefore, even though EDSMAC4 is doing calculations every 0.001 seconds during the

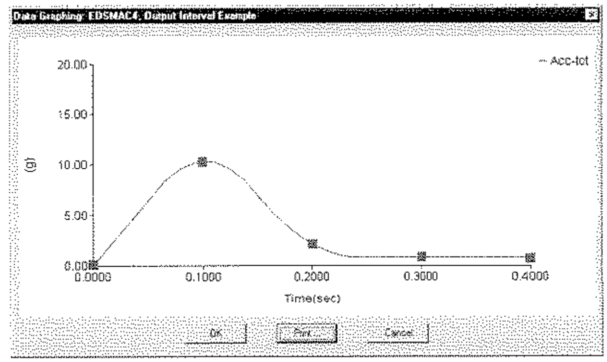


Figure 2: Total acceleration with Output Time Interval = 0.10. Solid squares are overlaid at timesteps where EDSMAC4 reported results to HVE. In the Variable Output table, HVE draws the graph based on these values.

collision phase, it is only outputting its results to HVE every 0.10 seconds. In other words, for every 100 calculations EDSMAC4 does, HVE receives only 1 value. You can see this when you watch an event execute. Keep your eye on the timer; it increments every 0.1 seconds, and your vehicles move every 0.1 seconds.

So, why does the graph look different from what we expect based on the Accident History and Damage Data output reports?

Alpha-numeric output reports (e.g., Accident History, Damage Data) are created by physics and critical values are stored as the simulation is executing. On the other hand,

> The Variable Output table is created by HVE using the values reported by physics at the Output Time Interval (Fig 3).

HVE uses a smoothing algorithm to create a graph based on the data sent to HVE at the Output Time Interval. If things are changing very rapidly, like reaching the peak of a collision pulse, and the Output Time Interval is not sufficiently small, it is likely that key data will be missed. This is exactly what happened in Figure 2 of our example.

Figure 3: Process used to pass numeric data to the Variable Output Table.

The solution is to decrease the Output Time Interval. In this case, decreasing the Output Time Interval from the default, 0.1, to 0.001 sec produced graphical results that matched the alpha-numeric results (Fig. 4). Comparing Figures 2 and 4, it is clear that the beginning, peak, and end of the collision pulse were all missed by using an Output Time Interval of 0.1 sec.

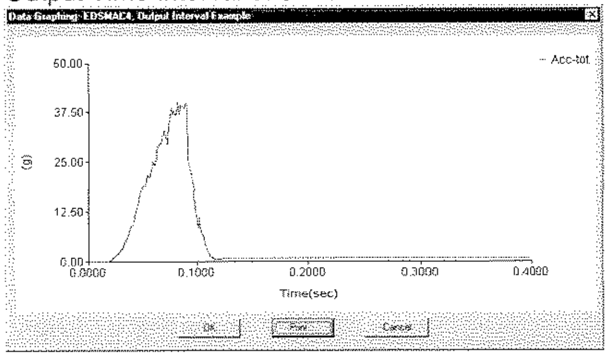


Figure 4: Total acceleration with Output Time Interval = 0.001.

Before you start decreasing the Output Time Interval in *all* of your events, however, consider the questions that you are trying to answer with your simulation. (Remember, decreasing the Output Time Interval will increase the time required to execute a simulation event because the scene is redrawn that many more times.) Are you looking for delta-V, damage, and rest positions? If so, it is probably not necessary to decrease the value. But if you are looking at the details of the collision pulse, you will have better results by decreasing the Output Time Interval. Users of occupant or pedestrian simulation models should be aware that the default Output Time Interval in these events is 0.001 sec for this very reason.

User Hints and Tips

- Want your simulations to run even faster while you are working to refine your events? Click the right mouse button and choose Draw Style – wireframe. (Or, under the Options menu choose Render and set Render Method to Wireframe.) Your humans, vehicles, and environment will be displayed as wireframes and the simulation will execute faster because of the decreased rendering time. To get it back to normal, choose Draw Style – as is. (Or, under the Options menu choose Render and set Render Method to Phong.)
- If you have humans or vehicles in a scene whose movements are just "for looks", the simplest approach is often to use EDGEN. EDGEN is an object kinematics program that uses user entered positions and velocities to determine the path and speed of the object. Don't try to use the steering, brake, throttle, and gear tables to back that tractor-trailer up. Instead, place the tractor-trailer at up to eight positions and let EDGEN do the rest. Best of all, EDGEN is free to all HVE users and will soon be free for all users of HVE-2D.
- Broke a Rim? Occasionally a 3-D simulation, such as EDVDS, EDVSM or SIMON, will terminate with the message "Excessive Tire Deflection! (Broke a Rim?)." This typically occurs when a vehicle's tire strikes a curb. To understand why the simulation terminated (and this message is issued) requires a basic understanding of how 3-D simulations calculate radial tire force. These simulation models begin by calculating the earth-fixed coordinates of the center of the wheel, then the earth-fixed coordinates of the "bottom" of the undeflected tire ("bottom" is defined in the direction of the radial tire vector normal to the terrain). The distance between these two earth-fixed coordinate positions is the tire deflection. Multiply the deflection by the tire's radial stiffness and you have the radial force acting on the tire normal to the surface. Of course, the tire's radial stiffness is not a simple linear constant. In fact, it is a 2-stage linear spring with a knee at about 80 percent of the tire's section height (Figure 5).

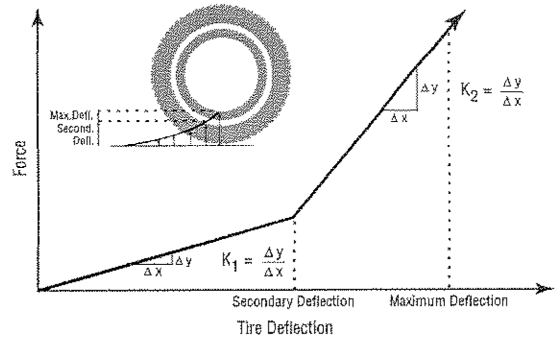


Figure 5: Force vs Tire Deflection curve.

Obviously, the forces change drastically when the deflection reaches the tire's section height. At that point, the tire deflection has reached a maximum. Any further deformation is not deforming the tire, it is deforming the rim. Of course, the rim does not deform in a linear manner, like the tire does. Depending on the rim's material and structural characteristics, it might either bend in a non-uniform manner, or it might simply fracture. These behaviors are quite different from the elastic behavior of the tire, and would require a finite element simulation to predict the resulting forces and moments. This is, of course, completely beyond the scope of vehicle dynamics simulation. Therefore, the simulation terminates with the message "Excessive Tire Deflection! (Broke a Rim?)."

It is rather easy to predict that a curb impact by a tire might terminate a simulation: Simply compare the curb height with the tire's section height. If the curb height is greater, chances are quite good that the simulation will terminate.

If the curb height is only slightly greater than the section height, the termination can be avoided by editing the tire's section height. This is done in the Vehicle Editor. Simply click on the desired wheel, choose Tire, select Physical Data, and increase the Maximum Deflection. This should only be done when necessary, and the increase should be nominal, say an inch or two. Obviously, if the actual event you are simulating really did result in a broken rim, you must recognize that no vehicle dynamics simulation can rigorously account for it. However, your simulation might provide some insight.

Another interesting condition can also terminate a simulation with the same message. That condition occurs when the terrain elevation is below the origin (i.e., its Z coordinates are positive), and the tire encounters a hole in the terrain. The hole is assigned a default elevation of 0.0, thus there is an elevation change. If the elevation change is greater than the tire's section height, the result is the same – for exactly the same reasons as outlined above. In this case, the solution is to find the hole in the terrain and repair it.

HVE and *HVE-2D* FAQ

This section contains answers to frequently asked questions submitted to our Technical Support staff by HVE and HVE-2D users.

Q: I am locking up my tires in a 3D physics program, but I don't see skidmarks. How could this be?

A: This is a fairly common question, especially for unloaded or lightly-loaded tractors (for the reason described below).

The programming logic that determines whether or not a tire leaves a skidmark is complex and varies between physics programs (e.g. EDSMAC4 is different from EDVSM). Refer to your Physics Manual, Calculation Method chapters for the details.

In general, however, the most common reason that a skidmark is not left when the user believes that it should be is that the vertical tire force, $F_{z'}$, is not large enough. All 3D tire models require a minimum vertical tire force to leave a skidmark. This minimum load value, "Minimum F_z for Skidmark", is a user-editable parameter found in the Tire, Physical Data dialog.

Q: When I use the HVE Path Follower, my vehicle does not go toward the target vehicle. Why not?

A: The HVE Path Follower allows the user to specify a desired vehicle path and attempts to force the vehicle, subject to the constraints of physics, to drive through those targets. Sometimes the vehicle cannot match the path because the maneuver is simply impossible for the vehicle or the driver or both. Other times, however, the vehicle may not follow the path because it does not "see" where it is intended to go.

The HVE Path Follower uses the user-editable value of Driver Preview Time (Driver Controls, Path Follower, Driver Data) to specify how far ahead, in time, the simulated driver should look for the path. The default value is 1 second.

Imagine that you want to drive in a circle at 55 mph. How far will you travel in 1 second? 80 feet. That means that at any timestep the model will ignore anything happening for the next 80 feet. It is therefore recommended that, instead of beginning the simulation right on the circle, you begin the simulation on a tangent to the circle about 80 feet away. Then the vehicle and driver will have plenty of time to ease into the turn.

Q: When using EDCRASH, what is the difference between the Δ-V displayed in the Event Editor Damage Profiles dialog and the value in the Accident History output report?

A: The Δ-V value displayed in the Damage Profiles dialog is the Δ-V that would have caused the entered damage had the vehicle hit a rigid barrier. Because the barrier would completely stop the vehicle, this is also the barrier equivalent velocity (or energy equivalent speed).

The Δ-V values reported in the Accident History report are the values computed by EDCRASH for the vehicle-to-vehicle collision. This calculation is based on either damage data or linear momentum, depending on the data that you entered and the details of the collision. Because the vehicle that you hit is neither rigid nor stationary (barriers are not allowed in EDCRASH), this value of Δ-V will always be lower than the barrier-based Δ-V.

Technical Session

This Technical Session addresses the use of *HVE* for motorcycle crash reconstruction. As we made quite clear in the June 2000 Technical Newsletter (see *Newton on Motorcycles*), it is unwise to attempt a mathematical reconstruction of the collision phase of a vehicle vs. motorcycle collision – if your goal is to calculate the speed of the motorcycle. However, there is much that can be done in motorcycle crash reconstruction, and this Technical Newsletter tells you how to use *HVE* to perform such a reconstruction.

HVE includes a number of tools that can be used for studying the pre-impact and post-impact phases of a motorcycle vs. vehicle collision. In our example, we'll use *EDSMAC4*.

To perform the *EDSMAC4* analysis of the pre-impact phase, the general procedures are described below:

Start by adding the vehicles to the case. Use a movable barrier to model the motorcycle. Attach the *MCYamaha650.h3d* motorcycle geometry file to the movable barrier.

Change the default exterior dimensions to create a box the approximate size of the motorcycle. Change the default wheel locations such that the wheels are positioned correctly, using the wheels displayed in the geometry file as a guide. It is important to make the wheel y coordinates as small as possible, say +/- 0.5 inches for the right and left sides, respectively. This approximates a cycle-type vehicle. This approximation does not affect what we are attempting to show in our simulation of the pre-impact phase.

Next, add the *Freightliner* tractor and *Generic Class 4* trailer. Attach the *TLVdsTutorTrlr45Van.h3d* geometry file to the trailer.

Next, add the environment. In our example, we added one of the pre-defined intersection geometry files, *4T4_Intersection.h3d*. Or, you can add your own, if desired.

Next, create an *EDSMAC4* event including the *Yamaha, Freightliner* and *Class 4 Generic* trailer.

Set up the event as you would normally, supplying the initial positions and velocities for the *Yamaha* and *Freightliner* (the trailer position and velocity are automatically assigned by *HVE* because it is attached to the tractor). Provide driver controls for the *Yamaha* (we added pre-impact braking to illustrate the motorcycle driver's response to the tractor/trailer entering the intersection) and the *Freightliner* (we added steering and throttle as required to cause the tractor/trailer to pull out into the intersection and turn left).

Now execute the event. The figure below shows the sequence just before impact.

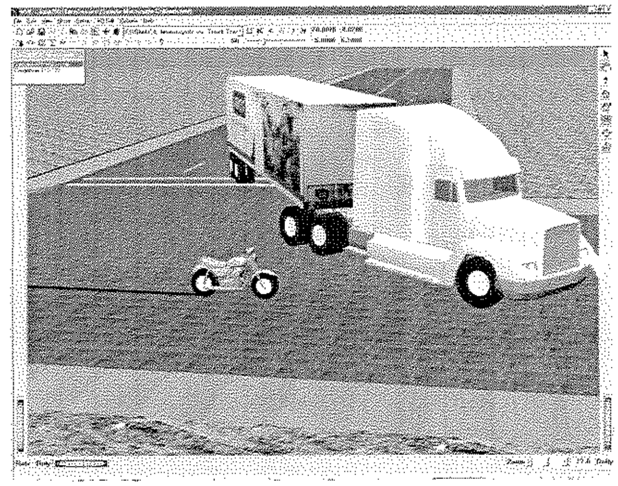


Figure 2: Screenshot of visualization of the pre-impact vehicle trajectories.

Because we used *EDSMAC4*, the motorcycle rider is not included in the event This is easily remedied using an *EDGEN* event to model the rider (if we had used *SIMON*, we could have included the motorcycle rider directly in the same event!).

That's all there is to it. The above simulation is a simple, straight-forward application of *EDSMAC4* to simulate the pre-collision phase of the motorcycle collision. The key step was to create a motorcycle, starting with a Generic Movable Barrier.

Several additional things can be done to extend and/or enhance the simulation. For example, *EDGEN* can be used to simulate the post-impact phase. *EDGEN* can also be used to illustrate various trajectories for the motorcycle driver after impact. Of course, you can also produce several views of the simulation, first attaching the camera to the motorcycle and then attaching the camera to the truck tractor, to illustrate the view available to each driver.

If you would like to see our final result, check out the movies on our website at www.edccorp.com/products/movies.html.

Creating a New Vehicle

Let's say we need to reconstruct a crash involving a 1999 Mercedes E420. We find that *HVE* does not include the E420 in its custom vehicle database, so we decide to build one.

There are two ways to create a vehicle: One way is to find a similar vehicle and modify it. The other way is to start with a generic vehicle and modify it. Frankly, there is seldom enough similarity between unrelated vehicle models to justify starting with a "similar" vehicle. Thus, we normally suggest starting with a generic vehicle.

It is worthwhile to remember that *HVE* generic vehicles were developed from statistical analysis of similar vehicles within a vehicle type and size category. The procedure for developing the Generic vehicle database was described in SAE Paper No. 960897, "Updating the Vehicle Class Categories." By starting with a generic vehicle, whether it is a 2-D vehicle or a 3-D vehicle, you are assured that the entire set of vehicle properties will include sound estimates for the vehicle you are simulating.

Creating a new vehicle is easy. Just perform the following steps:

1. Choose the Vehicle Editor. The Vehicle Editor is displayed.

2. Choose *Add Vehicle, New.* The Vehicle Information dialog is displayed. This dialog provides access to the various vehicle databases (Generic.DB, EDC.DB, MyVehicles.DB, etc.).

3. Select the vehicle type. The options are *Passenger Car, Pickup, Sport-Utility, Van, Truck, Trailer, Dolly, Fixed Barrier, Movable Barrier.* Our Mercedes is a passenger car, and that's the default selection, so we don't need to do anything further.

4. Select the vehicle make. The first passenger car make in the vehicle database is displayed. Since we want a generic vehicle, click on the *Make* option list and choose *Generic.*

5. Select the *Model* and *Year.* Because a *Generic* make was selected, both the *Model* and *Year* fields already display *Generic.*

6. Select the *Body Style.* For Generic vehicles, this is where the Vehicle Class Categories reside. Choose the correct category according to wheelbase. The Mercedes E420 has a 111.5 inch wheelbase. From SAE 960897, Table 4-1, the Mercedes E420 is a *Class 4* Passenger Car, so select Class 4 from the option list.

7. The remaining vehicle descriptors (*Number of Axles, Driver Location, Engine Location* and *Drive Axle*) are correct, so press *OK.* The Generic Class 4 Passenger Car is displayed in the Vehicle Editor.

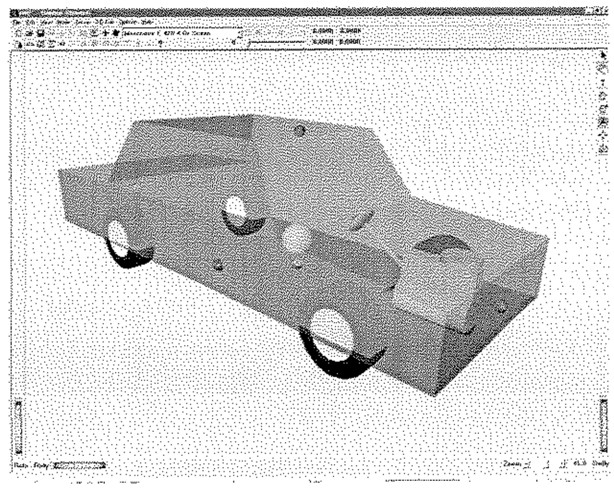


Figure 3: Generic body on Mercedes E420 4-Dr Sedan.

That's all there is to it. We now have a reasonable approximation of the 1999 Mercedes E420.

Of course, it is possible to edit the vehicle's generic parameters if they are known. The most common parameters are the vehicle total weight, exterior dimensions, wheelbase, trackwidth and stiffness coefficients. Most of these parameters are easily found from a variety of data sources – many you can measure yourself.

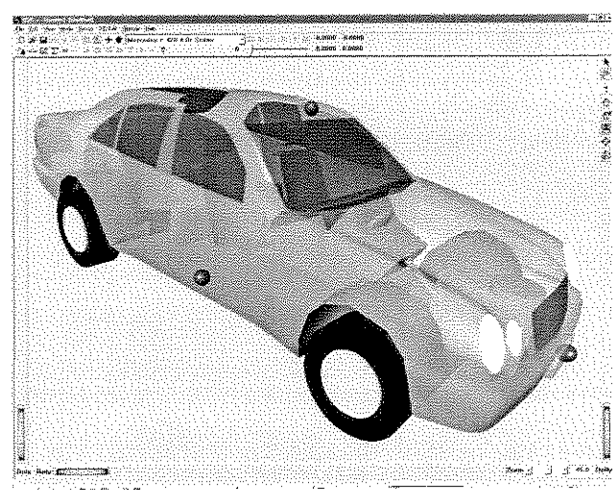


Figure 4: 3-D Model on Mercedes E420 4-Dr Sedan.

By default, the vehicle is displayed as a boxy approximation of a passenger car. If desired, *HVE* can display a 3-D model of the actual vehicle, as shown in Figure 4. This is done by obtaining a 3-D wireframe model (they are available from a variety of sources, including EDC) and attaching it to your vehicle.

The model of the 1999 Mercedes E 420 is now available for your current *HVE* case. But what about future cases? We can easily make it available to all future cases by adding it to our User database:

1. Click on the *Object Info* icon on the *HVE* toolbar. The Vehicle Information dialog is displayed, showing the attributes for the Generic Vehicle.

2. Choose *Save As.* The Vehicle Save As dialog is displayed.

3. Type in the *Make* (Mercedes), *Model* (E 420), *Year* (1996 – 2001; from Sisters and Clones List*) and *Body Style* (4-Door Sedan).

4. Press *OK* to accept the vehicle database descriptors. The Vehicle Information dialog now displays the vehicle descriptors for the Mercedes E 420. Note also the Source Database is displayed as *user.DB*.

5. Press *OK* again to remove the Vehicle Information dialog. The Mercedes is now displayed in the Vehicle Editor.

The Mercedes is now selectable in the Vehicle Information dialog as well. Let's confirm this:

1. Choose *Add Vehicle, New.* The Vehicle Information dialog is displayed.

2. Select *Mercedes* as the vehicle Make. Note the *Model* (E 420), *Year* (1996–2001) and *Body Style* (4-Door Sedan) for the newly created vehicle are displayed. Note also the Source Database is *user.DB* – this is a database containing vehicles that you created.

3. Press *OK*. Your new Mercedes is displayed in the Vehicle Editor.

---

* EDC Library Reference No. 1057

Tips on Making Simulation Movies with *HVE* Version 4 and *HVE-2D* Version 2

One of the benefits of *HVE* (and also *HVE-2D* version 2.0) is the ability to create an AVI (Video for Windows) simulation movie file of your results directly from the Playback Window. This feature allows the user to create a real-time movie that can be played back on a computer, or captured and recorded on video. Here are some important tips to remember when creating movie files.

1. *Select the best video codec (**co**mpressor/ **dec**ompressor) suited to your requirements.* There are 3 possible choices at this time, that are typically available within the Windows operating system. (NOTE: Other codecs you may see listed may not be suitable for the 640 x 480 pixel format and the typical 16 or 32-bit color settings used in the *HVE* Playback Window.) These are: Full Frames (uncompressed), Microsoft Video 1, and Cinepak Codec. There are also suitable "aftermarket" codecs available from software vendors, such as TechSmith Screen Capture (TSSC) available from TechSmith at www.techsmith.com. Each codec has tradeoffs for quality, compression speed and file size, and some codecs have additional configuration or setting adjustments. Full Frames (uncompressed) is going to provide loss-less image quality, but recent testing with Microsoft Video 1 and TSSC have also proven to provide excellent image quality and with a smaller resultant file size. Below is a table highlighting some of the differences between movie files produced from HVE.

Table 1- Five second simulation movie with the camera view attached to the moving vehicle.

Video Compression (codec) option selected within HVE	File Size
Full Frames (uncompressed)	90.6 MB
Microsoft Video 1	25.8 MB
TechSmith Screen Capture	33.6 MB
Cinepak Codec	12.4 MB

Typically, the smaller the file size, the "smoother" the playback will appear. However, smaller may not always be better for image quality. You should experiment to determine which code to use.

2. *Consider using a "post-production" program to further compress the movie file size.* Typically, users are working with programs such as Adobe Premier or Camtasia Producer (TechSmith) to combine individual AVI files into one complete movie of their case work. These programs allow a user to use a codec to further reduce the file size generated during the production of the final movie file. Below is a table highlighting results from using Camtasia Producer to further compress a single AVI file.

Table 2 – Post-production file size results of high-quality 5 second simulation movie with a fixed camera.

Codec option selected within *HVE*	File Size	Codec used in Post-production	File Size
Full Frames	90.6 MB	MS Video 1	51.2 MB
Full Frames	90.6 MB	TSSC	1.0 MB
MS Video 1	25.8 MB	MS Video 1	2.0 MB
MS Video 1	25.8 MB	TSSC	0.9 MB
TSSC	33.6 MB	TSSC	1.1 MB

Table 3 – Post-production file size results of 5 second simulation movie file with moving camera.

Codec option selected within *HVE*	File Size	Codec used in Post-production	File Size
Full Frames	90.6 MB	MS Video 1	51.1 MB
Full Frames	90.6 MB	TSSC	21.4
MS Video 1	25.8 MB	MS Video 1	15.5 MB
MS Video 1	25.8 MB	TSSC	18.4 MB
TSSC	33.6 MB	TSSC	21.4 MB

As can be seen clearly from the tables, use of a post-production program can significantly reduce the file size, especially if your movie file is from a simulation using a fixed camera view. These smaller file sizes may help you produce your final presentation materials for your clients much easier. You will need to view the resultant movie file to be sure that you still have the image quality and playback smoothness that you require.

NOTE: One important discovery of note that is evident from our testing is that when a simulation movie is recorded using *HVE* v4.0, the same file size will be produced no matter if the camera is fixed or moving, if rendering is set from 1 to 10, or if anti-aliasing is set from 1 to 10. SGI version users may recall that the video file size was affected by static or moving camera view, rendering and anti-alias settings.

There are also several programs available that allow you to record the "action" on a selected window or area of your desktop directly to a movie file. If you try to use one of these programs to capture a simulation movie being played either in the *HVE* Playback Window or in Windows Media Player, you may not be satisfied with the results. These screen capture programs may not be able to record as fast as the movie is played, thereby missing several frames which will cause the resulting movie to display very broken motion during playback.

A user should take a few moments and experiment with the different codecs available within *HVE* and determine which one is best suited for their requirements. If you have any questions, please contact technical support for more information.

Special thanks to Toby Frerich and Brian David Charles of Accident Reconstruction and Analysis in Corpus Christi, TX., for bringing the products available from TechSmith to our attention.

Importing Environment Files Into *HVE* Version 4

HVE Version 4 supports the use of VRML, Inventor or *HVE* format models directly, but requires the use of a translation program to prepare .3ds or other file types for use. The following guidelines are provided to help users import 3D terrain models built using AutoCAD or 3D Studio.

- When creating the model, use different layers and objects to keep groups separated. (E.g., roads, lane delineation, grass, sidewalks)
- Rotate the final model 180 degrees about the X axis.
- Scale the drawing to be in units of decimal inches.
- Export or save the model as a .3ds file format model. (This applies to both AutoCAD and 3D Studio users.)

There are advantages to saving as a .3ds file. Exporting the model as a .3ds file unifies the surface normals of faces to point in the positive Z direction (up in AutoCAD). Before the model is brought into *HVE*, these normals will need to be reversed to point properly in the negative Z direction (up in *HVE*). You can "invert" the faces using a program such as 3D Exploration available from www.righthemisphere.com.

- Translate the model from .3ds into .wrl (VRML 1.0) format using a translation program such as 3D Exploration. If using 3D Exploration, you must ensure that all default option boxes in the Save As dialog are unchecked before saving the model as a VRML 1.0 file. (I.e., Add indent, Lights, Cameras, Export Materials, Normals, Texture Coordinates and Polylines should be unchecked.)
- Copy the final .wrl format model into the HVE/supportFiles/images/environments directory.
- Open the model in *HVE*'s Environment Editor.

During the translation process from .3ds to VRML, the colors you used in 3D Studio or AutoCAD will be lost. You can use *HVE*'s 3-D Editor to re-assign colors and re-apply texture maps to objects in the model. Your model is now ready to drive on!

For more detailed instructions on creating and importing 3-D terrain models, please contact Technical Support.

Technical Support

HVE Version 4 includes some very nice features to help our technical support technicians assist you. The *Help* option includes your User ID and the email address for the support desk at EDC. Every technical support call goes through our support desk and all calls are documented. You are also given a Case ID number for future reference (in case there is a need for a call-back). These steps have been implemented to assist us in tracking repeated problems. For this reason, it is important that you always request EDC Technical Support, rather than ask for a support person directly. You may request to work with a particular support person. However, that person may not be available (out of town, on vacation, etc.). For that reason you should not leave email or voice mail pertaining to a new case.

EDC support technicians also meet each week to discuss *every* call logged during the previous week. Documenting each call allows our technicians seek advice from each other as well as to share their solutions.

Our technicians sometimes may ask you to email a case file to us for evaluation. This is quite easy to do. Simply send an email to the technician and include the case file as an attachment. In your email, describe the problem in as much detail as possible. Then click on your email program's *Attach* option and select the case file. By default, the file is found in the \Hve\supportFiles\case subdirectory, although the user can change this. If the case file is particularly large (while many case files a about 1 MB, it is possible for a case file that includes numerous lengthy events to exceed 50 MB), you can greatly reduce the file size by resetting each of the events prior to saving the case.

All requests for technical support are responded to within 24 hours. Our goal is to respond within 2 hours (our statistics show that 78 percent of all calls are responded to at the time of the call). Occasionally, information from a specific engineer may take somewhat longer. The time required to come to a solution varies greatly. Most requests for technical support are closed at the conclusion of the first call. However, some issues result in an ongoing dialogue. In addition, research may sometimes be required for complete closure of a technical support issue.

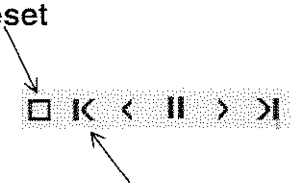
One more thing that's worth mentioning: All calls are strictly confidential, and you will *never* see us opposing you in court. We serve only our customers – not lawyers!

Technical Support can be reached by telephone between 8:00 AM and 5:00 PM PST at 503.644.4500 and by email at support@edccorp.com. Please be ready to provide your User ID#, your name, your company name and contact phone number when you call or include it directly in your email.

Tech Support Hints and Tips

- In order to simulate a failed brake system in *HVE*, click on the failed vehicle wheel(s), choose *Brake*, and set the Brake Torque Ratio equal to 0.0
- The Event Controller contains both *Reset* and *Rewind to Beginning* buttons, that may, at first, appear the same, but are, in fact, quite different.

Reset



Rewind to Beginning

After executing an event, pressing the *Rewind to Beginning* button causes the humans and vehicles to return to their initial positions. Notice that the *Reset*, *Execute*, and *Advance to End* buttons are all selectable. By contrast, while pressing the *Reset* button causes the humans and vehicles to return to their initial positions, it also causes the output tracks (results) to be erased. After pushing *Reset*, only the *Execute* button is selectable and no output is available in the Playback Editor.

Technical Support FAQ

We are frequently asked about the meaning of the A, B, and K_v stiffness coefficients used in the *HVE*-compatible programs *EDCRASH*, *EDSMAC*, and *EDSMAC4*. Both the CRASH and SMAC collision algorithms use vehicle crush stiffness coefficients that are derived from vehicle-to-barrier crash data. Following are descriptions of the methods for calculating the two sets of coefficients from fixed, rigid barrier crash test data. Crush stiffness coefficients can also be calculated from moving, rigid barrier crash test data, but the computations are more involved (see Vehicle Crush Stiffness Coefficients, EDC Library Reference No. 1042).

As you will see, you cannot convert directly from A and B to K_v without using details from the crash test data.

Fixed, Rigid Barrier Crash Test

W = test Weight (lb)

V = test Speed (in/sec)

C_{avg} = average Crush depth (in)

L = crush Length (in)

b_0 = threshold "no damage" test speed (in/sec)

b_1 = speed to cause 1 inch of crush $\left(\text{in/sec}/\text{in}\right)$

g = acceleration due to gravity (in/sec^2)

First, calculate C_{avg} as in Figure 5.

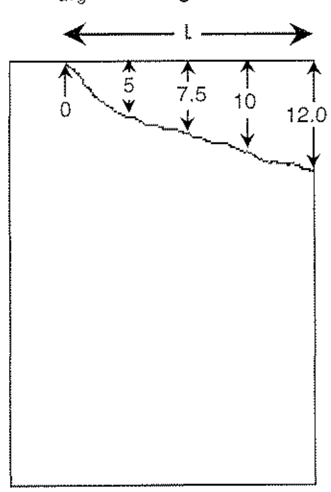


Figure 5: Vehicle with measured crush depths (in).

$$C_{avg} = \frac{0+5+7.5+10+12.5}{5}$$

$$C_{avg} = 7.0 \text{ inches}$$

CRASH and *EDSMAC4* Collision Algorithms

The *EDCRASH* and *EDSMAC4* collision algorithms use A and B stiffness coefficients, defined below and in Figure 6. Using these two coefficients, Force and Crush are linearly related.

A = force required to initiate crush (per unit width) (lb/in)

B = spring rate (per unit width) (lb/in^2)

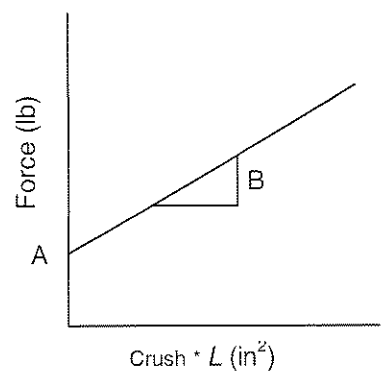


Figure 6: A and B stiffness coefficient definitions.

The process for calculating A and B is described below.

(1) Assume b_0 = 5 mph = 88 in/sec (for a 5 mph bumper)

(2) $b_1 = \dfrac{V - b_0}{C_{avg}}$ (assumes rebound velocity = 0)

(3) $A = \dfrac{W \;\; b_0 \;\; b_1}{g \;\; L}$

(4) $B = \dfrac{W \;\; b_1^2}{g \;\; L}$

SMAC Collision Algorithm

The *EDSMAC* collision algorithm uses a single stiffness coefficient, K_v, defined below and in Figure 7. Using this one coefficient, Force and Crush are linearly related, as in the *EDCRASH* and *EDSMAC4* collision algorithms, but now the curve of Force vs. Crush passes through (0,0). This means that *any* force will result in crush.

K_v = spring rate (per unit width) (lb/in2)

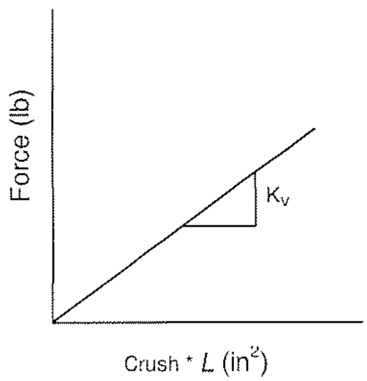


Figure 7: K_v stiffness coefficient definition.

The process for calculating K_v is described below.

(1) Assume b_0 = 0 mph = 0 in/sec

(essentially a 0 mph bumper)

(2) $b_1 = \dfrac{V}{C_{avg}}$

(3) $K_v = \dfrac{W \quad b_1^2}{g \ L}$

For simulations in which the closing velocity is less than the test speed, SMAC over predicts crush, but

➢ GETS THE FORCE CORRECT! (Figure 8).

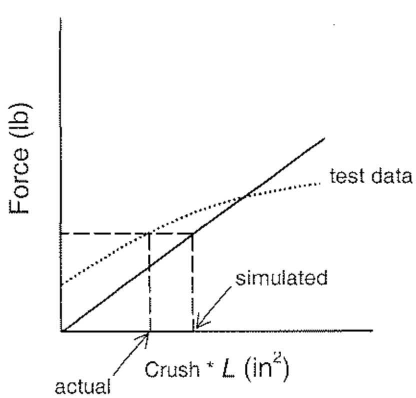


Figure 8: Comparison of actual to modeled vehicle crushes and crush stiffness using the K_v stiffness coefficient.

Technical Session

Rollover Simulation

HVE allows the study of complete vehicle rollover using kinetic simulation (that is, the vehicle motion is determined by forces acting against the vehicle body while in contact with the ground). The computational method was described in SAE Paper No. 2000-01-0852, "Applications and Limitations of 3-Dimensional Vehicle Rollover Simulation." An additional article appeared in the June 2000, Technical Newsletter.

With the introduction of *HVE* Version 4, we expect (and hope for) a significant increase in the number of rollover events being simulated. Simulation provides an increased understanding of the events leading up to the crash, as well as a better understanding of the various mechanisms involved in rollover.

A word of caution is in order. Just because it can now be done does not imply it can be done easily! This point was made in both the SAE paper and the Technical Session. There are two reasons for making this point. First is a technical problem: The tire models in all 3-D simulations begin by calculating the tire's radial deflection due to contact with the ground. When, as the vehicle rolls, if the tire plane becomes nearly parallel with the ground plane, the radial deflection can increase suddenly to an abnormally large value. In the extreme case, the tire plane and ground plane become parallel and any tire penetration into the ground results in an infinite tire deflection. This is a technical problem for which there are often workarounds (refer to the SAE paper). Modeling changes are also possible (in fact, likely) that will eliminate this issue. But, this is the easy part.

A vehicle rollover is nearly chaotic. Witness the fact that it is not possible to duplicate the same results during repeated rollover testing under well-controlled test conditions. While the rollover model is very robust and sophisticated, it is just a model. It is interesting to make small changes to the inputs and observe the changes in the results. Sometimes the differences are minute, while other times, they are significant. Duplication of a real-world event takes time, skill and (perhaps most of all) luck.

So why do a rollover simulation? Many reasons. One reason is to estimate the speed at the start of a loss-of-control maneuver. A second important reason is to gain an understanding of the driver inputs required to initiate the rollover. You will find that, regardless of the exact details of the rollover process, the total distance traveled and the number of complete rolls are quite speed-dependent. Thus, rollover simulation is quite useful to help pin down a possible range of initial speeds and driver inputs.

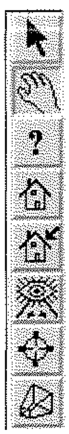
Rollover simulation is also useful for studying how various vehicle factors, such as CG height, track width, suspension design and tire selection, affect the propensity for rollover.

Another important reason is to study the effects of various environment tripping mechanisms, such as curb tripping and soil furrowing. Although neither of these mechanisms is handled perfectly by our current *point contact* tire models, significant insight is still gained through approximate models of these environments. Curbs can be modeled using steeply sloped surfaces and a significantly reduced integration timestep. Furrowing can be approximated using terrain friction multipliers.

With sufficient time and the proper level of expectation, rollover simulation can be a valuable new tool in your arsenal. We encourage you to discover how rollover simulation can assist you in your reconstructions.

User Hints and Tips

- Your experience while using *HVE* Version 4.00 will be greatly enhanced by using a 3-button mouse. You may choose one with or without a center scroll feature. The key is to configure the mouse so that the middle button, whether traditional button or scroll disk, is the "middle mouse button". (An optical mouse is really nice because you do not need a mouse pad or even a flat surface to use it; e.g. a pants leg works great.)
- Version 4.00 includes a new way to toggle between *Pick* and *Manipulate* modes: press the right mouse button and select *Viewing.* Alternatively, you can either press the <*Escape*> key or click the hand and pointer icons on the right-side tool bar.
- *HVE* Version 4.00 makes use of a wide array of viewer functions with buttons on the right-side toolbar (see below for descriptions). In addition to the *Pick* and *Manipulate* buttons with which you are already familiar, the *Go to Home view, Assign Home view,* and *Zoom on selected object* functions are particularly useful. Experiment!



Change to Pick mode
Change to Manipulate mode
Access viewer Help system
Go to Home view
Assign Home view
View all
Zoom on selected object
Toggle to orthographic view

The environment and vehicle geometry files that you built to use on the SGI can be used immediately in Version 4.00. Simply copy your environment and vehicle geometry files directly from your SGI to your PC. They are ready to go in .iv format. Similarly, copy your .rgb format textures files.

- You can change the font size used in alpha-numeric output reports by editing the Preferences under the Options menu. You may need to decrease the font size from the default (10 pt) to 8 pt in order for some reports to fit on a printed page. Any changes that you make will be saved in your configuration file, as are other user preferences such as units and background color.
- Exporting time-based vehicle simulation data from *HVE* to a spreadsheet or graphing software package, such as Microsoft Excel, allows you to make great looking graphs and to graph one variable against another (i.e., without time as the independent variable). It is simple to do this in Version 4.00. With the Variable Output window selected in the Playback Editor (the title bar will be highlighted), select *Print* from the File menu. Choose a Generic/Text printer. Click in the *Print to File* check box and press the *Print* button. Assign a file name with the .txt extension, and change the file type to *All Files (*.*)*. Press *OK*. Now, open Excel and select *Open* from the File menu. Select the text file that you just created. Use the formatting options to make sure that your data stays in the correct columns. That's all there is to it.
- It is also very simple to take graphic screen shots of your human, vehicle, environment, or event. Press either <*Print Scrn*> to make a screen shot of everything appearing on your monitor or simultaneously press <*Alt*> and <*Print Scrn*> to make a screen shot of only the active window. Next, open Paint by going to your Start menu and selecting *Programs, Accessories, Paint*. With Paint open, *Paste* the screen shot into the window. Paint can be used to crop, add text, add shapes, and perform other useful functions. When you are happy with your image, *Save* it. This image can be imported into word processing, presentation, or publication software packages.
- Use the arrow keys on the keyboard to move within the Driver Controls tables.
- Use the <*Tab*> key to move between fields in the Contact Surfaces dialog.
- *HVE* Version 4.00 has features that make it easier for you to contact us for technical support. Click on *Help* on the main menu bar and choose *Tech Support*. We have listed telephone, fax, and email contact information for EDC Technical Support. Also, we have listed your name, company name, and user identification number. Click on *Help* on the main menu bar and choose *About*. Your *HVE* version number is listed. Have this information available when you contact EDC, and you will be put through to a technical support engineer even faster.

HVE Version 4.00 FAQ

This section contains answers to questions submitted to Technical Support by *HVE* users.

Q: When will *HVE* Version 4.00 be available?

A: RIGHT NOW!! Enjoy.

Q: Can I install *HVE* Version 4.00 on more than one computer?

A: Yes. You must have the EDKEY security dongle attached to the printer port in order to run the program. This arrangement allows you to use *HVE* Version 4.00 both on your desktop at work and on you laptop at home or on the road. Or, share the EDKEY with a colleague in your office so that you both have access to *HVE* at your own desks.

Q: Can I use my environments (and textures and vehicle geometries) that were made on the SGI?

A: Yes, your environments can be used right away in *HVE* Version 4.00. Just copy them onto your PC and into your /hve/supportFiles/images/environments/ folder. Similarly, copy over texture and vehicle geometry files into the appropriate folders and they are ready for use.

Q: Which texture formats are accepted by *HVE*?

A: The best file formats for textures are .tif, .gif, .rgb, .bmp, and .jpg. These formats basically cover the spectrum of standard formats.

Q: Can I use my cases from the SGI?

A: A case translator is not yet available for converting files between SGI and PC formats, but we are working on it.

Q: Is there a SceneViewer-type utility program that I can use to change the transparency of windows, etc.?

A: Yes, it is called SceneViewer. SceneViewer is shipped as an *HVE* utility and has all of the functionality that you have used on the SGI.

Q: What if I have only a 2-button mouse?

A: Use a combination of your 2-button mouse and the keyboard to produce the same rotate, pan, and dolly functions.

Function	Mouse Button	Keyboard Key
Rotate	Left	
Pan	Left	*Shift*
Dolly	Left	*Ctrl-Shift*

However, your experience with *HVE* will be more enjoyable if you do use a 3-button mouse.

Q: What formats are recommended for new vehicle and environment geometry files?

A: Both .iv (Inventor) and .wrl (VRML) format geometry files can be imported by *HVE* v.4.00. For 3-D environments, version 1.0 .wrl is the preferred format. If you are using 3-D Studio for building vehicle or environment geometry files, simply save the file as version 1.0 .wrl.

Q: Can I print with a non-Postscript printer?

A: Yes, just like any other Windows-based program, *HVE* will print to any Windows compatible printer. As a general tip, printer drivers are updated often, so check the website of your printer manufacturer to ensure that you have the latest version for your printer.

Q: How do I make a video?

A: First, please read the *HVE* User's Manual. There are changes from *HVE* Version 3.01. *HVE* has the built in capability of creating an AVI file. This can be played back within *HVE* or using an external application such as Movie Player. It can also be put into various presentation software packages, such as Microsoft PowerPoint. The process of routing it to videotape will depend on your video card or peripheral. Refer to the owner's manual for your specific video hardware.

Q: Are there ways to store/record a movie file other than on videotape?

A: Many computers now come with read/write CD ROM drives. If you have this type of drive, it is easy to copy your AVI movie file on to a CD for transportation or distribution. Among other benefits, the quality of this file will remain higher than achieved when routing the movie file to videotape.

Technical Session

4-Spring and Walking Beam Suspensions

The new versions of *SIMON* and *EDVDS* now model tandem axles with 4-spring and walking beam suspension types. These suspensions are different from solid axle suspensions in that the axle displacement of one axle affects the axle displacement of the other axle. The axle displacements, in turn, affect suspension forces. The result is a load-leveling effect between the leading and trailing axles. The modeling approaches for 4-spring suspensions and walking beam suspensions are different.

4-Spring Suspensions

In a 4-spring suspension, the front and rear leaf springs are connected by a pinned link (see Figure A). This link tends to transfer the load between front and rear suspensions. The result is that suspension force for the front and rear springs is equalized (or nearly so). The equalizing effect is modeled by calculating the average axle displacement from equilibrium and adding/subtracting this value to/from each spring deflection:

$$\overline{\Delta}_{Leading} = \dot{\overline{\Delta}}dt$$

$$\overline{\Delta}_{Trailing} = -\dot{\overline{\Delta}}dt$$

$$\delta_{Spring,Leading} = \delta_{0_{Spring,Leading}} - \overline{\Delta}_{Leading}$$

$$\delta_{Spring,Trailing} = \delta_{0_{Spring,Trailing}} - \overline{\Delta}_{Trailing}$$

$$\dot{\delta}_{Spring,Leading} = \dot{\delta}_{0_{Spring,Leading}} - \dot{\overline{\Delta}}$$

$$\dot{\delta}_{Spring,Trailing} = \dot{\delta}_{0_{Spring,Trailing}} + \dot{\overline{\Delta}}$$

$$F_{Spring} = f\left(K, C, C_f, \delta_{Spring}, \dot{\delta}_{Spring}\right) + \Delta F_{X-fer}$$

where

$\overline{\Delta}$ = Average spring deflection

$\dot{\overline{\Delta}}$ = Average spring velocity

dt = Integration timestep

δ = Spring deflection

$\dot{\delta}$ = Spring deflection rate

δ_0 = Axle deflection

$\dot{\delta}_0$ = Axle deflection rate

F_{Spring} = Suspension spring force

K = Spring rate

C = Spring damping coefficient

C_f = Spring coulomb friction

ΔF_{X-fer} = Inter-axle load transfer due to wheel torque

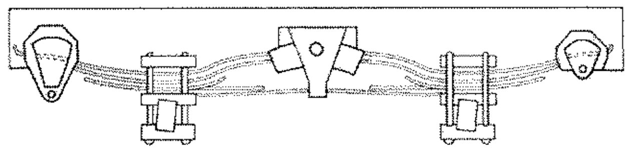


Figure A

Walking Beam Suspensions

In a walking beam suspension, both axles are supported by a single spring (see Figure B). Therefore, only the front axle suspension parameters are used when a walking beam suspension is selected. The approach for walking beam suspensions is quite similar to 4-spring suspensions:

$$\overline{\Delta}_{Leading} = 0.5\left(\delta_{0_{Leading}} - \delta_{0_{Trailing}}\right)$$

$$\overline{\Delta}_{Trailing} = -\overline{\Delta}_{Leading}$$

$$\dot{\overline{\Delta}} = 0.5\left(\delta_{0_{Leading}} + \delta_{0_{Trailing}}\right)$$

$$\overline{\Delta} = \delta_{0_{Leading}} - \overline{\Delta}_{Leading}$$

$$F_{Spring} = f\left(K, C, C_f, \overline{\Delta}, \dot{\overline{\Delta}}\right) + \Delta F_{X-fer}$$

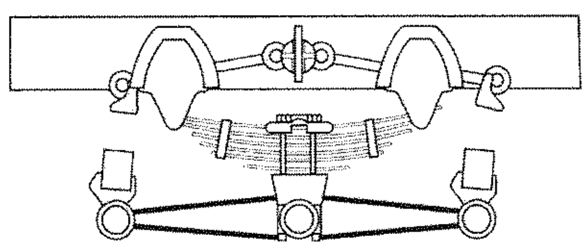


Figure B

Load equalization between axles is desirable as it prevents tire/suspension overloading while traveling over irregular terrain. Both the 4-spring and walking beam suspensions do a reasonable job in this regard.

Inter-Axle Load Transfer

Driving and braking torques result in an inter-axle load transfer for both suspension types. The reaction at the spring ends increases or decreases due to the applied torque (see Figure C). This reaction force increases or decreases the total spring force. The amount of load transfer is calculated from the applied axle torque and the user-supplied inter-axle load transfer coefficient,

$$\Delta F_{X-fer} = \gamma \frac{T_{Leading} + T_{Trailing}}{X_{Leading} - X_{Trailing}}$$

where

γ = Inter-axle load transfer coef (+ for load transfer from the rear axle to the front axle)
T =Wheel torque (+ for drive torque; - for brake torque)
X =Wheel vehicle-fixed x-coordinate

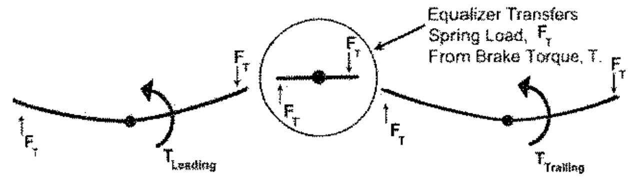


Figure C

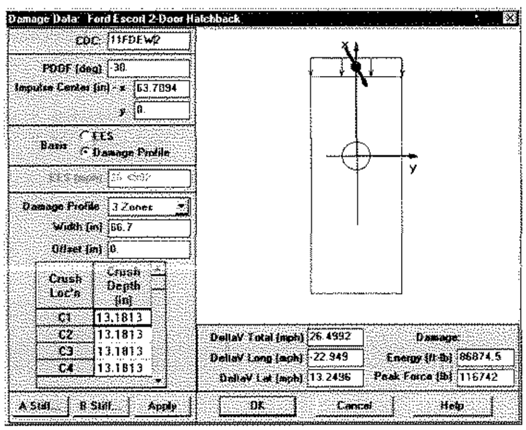
The inter-axle load transfer coefficient represents the ratio of the vertical load transferred from the trailing axle to the leading axle, to the total applied wheel torque. Fancher* provides test results for inter-axle load transfer coefficients for several suspension systems. In general, inter-axle load transfer is negative for 4-spring suspensions (meaning that the spring force is reduced on the front axle and increased on the rear axle). –0.10 to –0.20 is a reasonable range for most 4-spring suspensions. Inter-axle load transfer is approximately zero for walking-beam suspensions.

* Fancher, P.S., et. al., "A Factbook of the Mechanical Properties of the Components for Single-Unit and Articulated Heavy Trucks," University of Michigan Transportation Research Institute, Ann Arbor, 1986, EDC Lib. Ref. No. 1053.

User Hints and Tips

General

- *EDCRASH* uses the 7-character Collision Deformation Classification (CDC) code that describes damage profile characteristics including PDOF, Damage Location, Damage Width, and Average Crush Depth. Consider the CDC 11FDEW2 (shown below). "11" is the clock direction of the PDOF. "F" identifies the vehicle front as the general damage region, and "D" identifies the entire front as the specific damage region. "E" specifies that the damage occurred below the beltline. "W" means that the damage width was wide (> 16 inches). Finally, "2" pertains to the maximum extent of penetration.



The vehicle's damage profile default information can be edited once a CDC has been entered. An in-depth discussion of CDC can be found in SAE Information Report (SAE J224).

HVE-2D

- *HVE-2D* version 1.33, shipped in December, contains an important change in the physics behind the Edit - Driver Controls – Wheel Data dialog. The dialog prompts the user to enter pre-impact braking as Percent of Total Wheel Lock-up, rather than Drag Factor. This accurately reflects the new equation for computing begin braking velocity (below). The interface has not changed, just the physics model. Your new documentation reflects this change. It is advised, therefore, that you reexecute any events in which a begin braking position was entered.

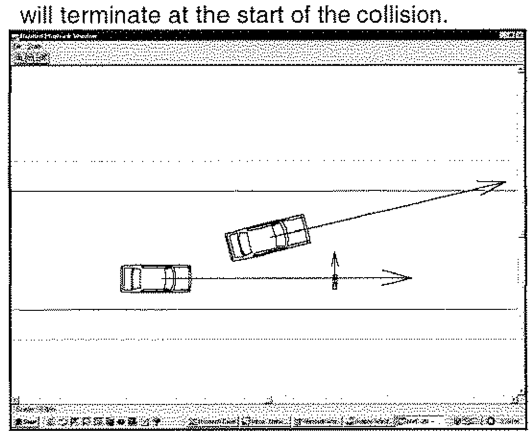
$$V_{bb} = \sqrt{V_{imp}^2 + 2g\theta\mu S}$$

where:

V_{bb} = velocity at begin braking
V_{imp} = velocity at impact
g = acceleration of gravity
θ = total wheel lock-up
μ = coefficient of friction
S = path length

- Although *HVE-2D* does not allow for vehicle to human collisions, moving barriers may be used to represent visually a human pedestrian or other moving object in time-distance studies. The dimensions of the barrier can be changed to the approximate size of a human, a velocity can be assigned, and driver controls can be used to dictate the speed and direction of the barrier. In the example below, two *EDSMAC* events are combined in a Playback Window to simulate a vehicle maneuvering around a pedestrian who is crossing the street.

  Caution: Moving barriers must not be used to simulate collisions with vehicles. The simulation will terminate at the start of the collision.



- Sometimes it is desired to have *HVE-2D* up and running without seeing your desktop icons in the background. In order to have *HVE-2D* occupy the entire screen, follow these simple instructions.

  (1) Right click on the *HVE-2D* icon, and then choose Properties.
  (2) In the Properties dialog, select the Shortcut tab. In the Run command line, choose Maximized from the drop down list.
  (3) Press OK, and you are finished.
  (4) Start *HVE-2D*.

  Note: If you need to view the event information dialog in the Event Editor, you must first hide the key results, under the Options menu.

HVE and HVE-2D FAQ

This section contains answers to questions submitted to Technical Support by users of *HVE* and *HVE-2D*.

Q: I am reviewing the work of another reconstructionist who is using *HVE* or *HVE-2D*. What information do I need to request from that expert in order to evaluate his or her results?

A: For *HVE-2D*, you should request the case file (with the events executed) and the environment geometry file. While the environment is not used by the physics, it may be important for visualization studies.

If you are reviewing *HVE* results and are a user of *HVE*, you should request the case file (with the events executed). The case file will contain all environment (including 3-D geometry) and vehicle data. The case file does not contain any vehicle 3-D geometry files. These files are required for reproducing *DyMESH* simulations involving collisions and *EDVSM* simulations involving rollover. These 3-D geometry files are licensed property and cannot be transferred without permission of the vendor (EDC, Viewpoint, or whomever). These files may, however, also be purchased from the vendor.

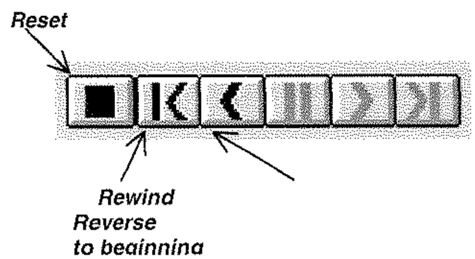
If you are reviewing *HVE* results and are not a user of *HVE*, you should request all of the numeric output reports, including Messages, and a video of any Trajectory Simulations or event Playbacks.

Q: When using *EDSVS*, why is the yaw moment of inertia that I enter in the Vehicle-Inertias dialog different from the one reported in the Vehicle Data output report?

A: The value for yaw moment of inertia that you enter in the Vehicle-Inertias dialog is the value about the vehicle's center of gravity, as expected. *EDSVS* is based on a model developed at the University of Michigan, TBST. In this model, the yaw moment of inertia is resolved about the geometric center of the vehicle. *EDSVS* uses the same logic. As for all variables, the value given in the output report is the value actually used by the program.

Q: Are the data in the output reports updated when I reexecute an event?

A: Yes, the data in the output reports are updated when you reexecute an event. In order to reexecute an event, you must first press the Reset button on the Event Controller (see below). This will effectively erase all of your previous results but will not effect input values. If you did not reset the event, you will just be replaying the results from before you made your changes, and the numeric reports will be identical.



Q: Why do I get Security Error 2 when I try to start *HVE-2D* on my Gateway G6 computer running Windows 98?

A: There is a known conflict between some soundcard drivers on Gateway G6 computers and the EDKEY driver. The soundcard on these computers overwrites the memory used by the EDKEY System Driver. The solution is to go to the Gateway Computer website, http://www.gateway.com/support/product/drivers/index.shtml, and download the latest driver for your particular soundcard.

Q. When I open the Damage Profile Preview Window for an *HVE-2D EDSMAC* simulation, the vehicle does not look damaged. Why?

A. You must open both a Trajectory Simulation and a Damage Profile Preview Window for your event in the Playback Editor to view properly vehicle damage. Similar to playing a Trajectory Simulation, when you press the Play button in the Playback Editor you will see the damage displayed as a function of time. More detailed instructions on viewing damage are included in the file HVE2d/damage.DOC.

Technical Session

This installment of the Technical Session discusses time-distance studies. A time-distance study is used to define the relative positions of two or more moving objects as a function of time. (The objects are typically vehicles and/or humans – although they may conceivably be anything.) The study may be tabular (i.e., X-Y coordinates vs. time) or visual (i.e., a trajectory simulation). If visual, it may be 2-dimensional or 3-dimensional. In any case, the goal of a time-distance study is to define what is often referred to as the *last clear chance* for each driver and/or pedestrian to avoid a possible collision. By comparing the time at which the last clear chance occurs for each participant with typical human perception-reaction times, the crash recon-structionist can draw conclusions regarding causation and avoidability.

Our Technical Session will use *EDSMAC(4)* to demonstrate the procedures involved. We will assume the crash has already been reconstructed and simulated. Thus, pre-impact paths and initial velocities are known. A little planning helps: start the simulation well before impact so the simulation begins before the point of last clear chance. The methods for *HVE-2D* and *HVE* are different, so they will be discussed separately.

HVE-2D Procedure

A scaled diagram of the crash sequence is shown in Figure A. The simulation is executed at 0.1 second intervals. As the vehicles progress along their pre-impact trajectories, identify the time when visibility begins. Referring to Figure A, Point A, represents the obstruction. Think of point A as a "fulcrum". Rotate a straight edge (e.g., a ruler or similar object) about point A until the straight edge intersects the driver's eye and the first visible part of the other vehicle *at the same simulation time.* Do this for each vehicle's driver, identifying this time (let's call it T_n, where n is the driver's vehicle number). You may find that the driver of one vehicle can see the other vehicle slightly before or after the other driver can see his/her vehicle.

Next, note the simulation time for various important events: the time at which each vehicle begins steering ($T_{Steer,n}$), the time at which each vehicle begins braking ($T_{Brake,n}$), and the time of collision (T_{Coll}). We now have all the necessary data to draw some very important conclusions:

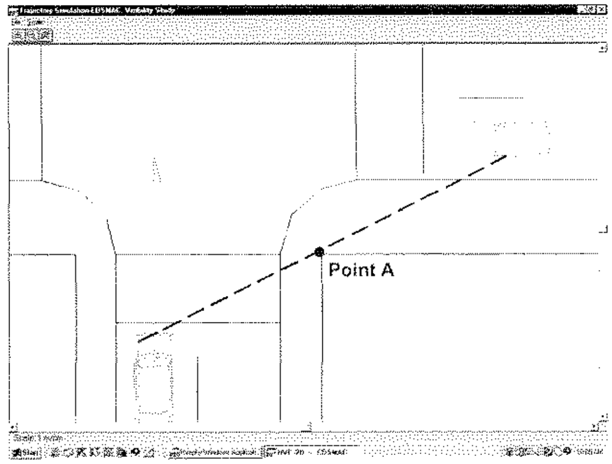


Figure A-HVE-2D Scaled Diagram of Crash Sequence

The total time available (T_{Avail}) for driver n to avoid impact is

$$T_{Avail,n} = T_{Coll} - T_n \qquad \text{(Eq. 1)}$$

The actual perception/reaction time ($T_{P/R,Steer}$) driver n begins steering is

$$T_{P/R,Steer,n} = T_{Steer,n} - T_n \qquad \text{(Eq. 2)}$$

The actual perception/reaction time ($T_{P/R,Brake}$) driver n begins braking is

$$T_{P/R,Brake,n} = T_{Brake,n} - T_n \qquad \text{(Eq. 3)}$$

By comparing $T_{P/R}$ with accepted values (see footnote), one can draw important conclusions regarding driver attentiveness.

The above procedure shows why a scaled trajectory simulation has a major advantage over a tabular approach: The table does not include the effect of the obstruction. Thus, when a table is used, the reconstructionist still needs to plot the distances on a scaled diagram that includes the obstruction.

Footnote
Olsen (SAE Paper No. 870600) provides a good reference for driver perception/reaction times. For designing highways, the AASHTO Manual (A Policy on the Geometric Design of Highways and Streets, American Association of State Highway and Transportation Officials, Washington, DC, 1990) uses 2.5 seconds as a visibility requirement.

HVE Procedure

A 3-D model of the crash site is shown in Figure B. The procedure is quite similar to the procedure for HVE-2D. One huge difference is that *HVE* allows the user to attach the camera to the vehicle at the driver's vehicle-fixed eye coordinates so you can *see* when the other vehicle becomes visible. Proceed to display the simulation, stopping when the other vehicle first becomes visible. This has been done in Figure B (note the other vehicle is just becoming visible beyond the corner of the building). Again, repeat this process for the other driver; the times may be slightly different.

Eqs. 1 through 3 (previous page) are used for the *HVE* procedure.

Note there is a significant benefit to using *HVE* for a time-distance study: In addition to learning directly about visibility through the use of a vehicle-fixed camera, the user inherently learns about *conspicuity*. For example, a simple tabular study or *HVE-2D* simulation tells the user when visibility first exists, but it does not address the qualitative aspect, *How Visible?* Straight-line visibility may exist between a driver's eye and a child's head, but that does not say anything about *how visible* the child's head might be from a distance of, say, 300 feet (can you pick out a cantaloupe a football field away?)

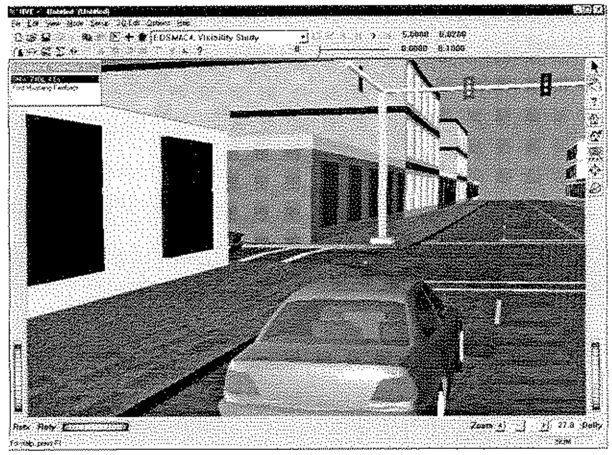


Figure B-HVE Event View of Crash Sequence

User Hints and Tips

General

- Cases of secondary vehicle impact (where two vehicles impact each other twice) can be reconstructed easily using *EDCRASH*. The technique is to perform a momentum only analysis. Obviously, this will not work for collinear impact, but collinear impacts rarely result in secondary collisions. If you do put in damage for an event in which there was a secondary collision, you are likely to receive an error message because the damage analysis ignores the second collision.

HVE

- You can make "photographs" of whatever is on your screen using a program called snapshot. Simply open a winterm on your SGI, and type "snapshot". A small gray box will appear with the word "snapshot" written in white letters across it. Move your cursor on top of the box and it will change from the shape of an arrowhead to the shape of a camera. Press and hold down the left mouse button while in camera mode and you can stretch a rubber band to define the border of your picture. After placing the first corner, you can go back to the opposite corner and place that one. If your rubber band is not exactly how you want it, you must start again by going back to the snapshot box. Once your rubber band is positioned properly, click the right mouse button to access the menu for saving and exiting. First, assign a new file name ending in the .rgb extension. Then save, or exit and save. You will now have the .rgb file, entitled yourfile.rgb, in the directory from which you typed "snapshot".

- If you are going to transfer this figure to a PC and use it in software such as Microsoft Word or Adobe Illustrator, however, you now need to convert the .rgb file to .tif. You can do this, as well as any editing, in Image Works. Simply type "imageworks" in the winterm, click on File-Open, and select your file. When ready to save, select File-Save As. Under File Format, select TIFF. Use the Set Directory buttons to

- Identify the location in which you want your new file to reside. Enter the new file name as yourfile.tif Click Accept. You are now ready to use the image for making reports, slides, posters, or whatever you need.

- Take advantage of *EDGEN* to make humans, vehicles, or just about anything move in a prescribed path where simulation is not important. An important example of *EDGEN's* utility is when there are vehicles in the scene that are not involved in the event that you are simulating. Create one or more *EDGEN* events with these vehicles and combine them with the simulation in the Playback Editor.

HVE-2D

You can make "photographs" of whatever is on your screen for use in reports, slides, posters, or wherever you need them. With the image open (Trajectory Simulation, Damage Profile window, etc.), select the window that you wish to capture by clicking on the title bar. Press the Alt and Print Screen buttons simultaneously. Open the Paint Application. On most PCs you can find this by clicking on the Start button in the lower left-hand corner of the screen and selecting Programs-Accessories-Paint. With Paint open, go to File-Paste (ctrl-v). This will place an image of the selected window into the Paint window. Without clicking on the image, go to Edit-Copy To. This allows you to save the image to a file. Give the file a name and select the desired file format. You are now ready to use the image for making reports, slides, posters, or whatever you need.

If you are a new user of *HVE-2D*, make sure to work through the tutorials found at the end of each chapter in the Physics Manual. By doing the tutorials, you will learn how to navigate efficiently through the interface and gain a greater understanding of the physics programs.

In HVE-2D there are three important user assigned names: Case Title, Case File Name, and Event Name. All three are saved with your case and will help you keep track of your work.

Each file will have exactly one Case Title and one Case File Name. These are entered in the Save Case File dialog, accessed by choosing Save As from the File menu or when choosing Save for the first time (Figure 1). The Case File Name is the name by which the file is stored on your computer and the name that you will select when opening a previously saved case. There is an eight character limit on the name length, and it will always end in the extension *.cs2.

The Case Title is entered at the bottom of the Save Case File dialog (Figure 1). The Case Title field is a place for you to enter a descriptive title for the case. For instance, in the case of the EDCRASH tutorial, the case title is "EDCRASH Tutorial, Visibility Study". It is not mandatory that you enter a unique Case Title, but it can be handy when keeping track of your work.

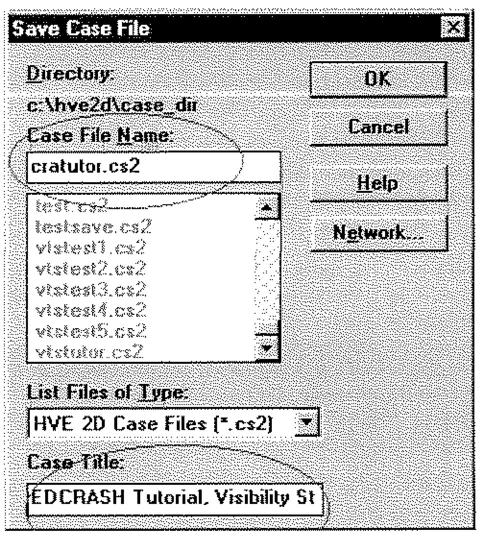
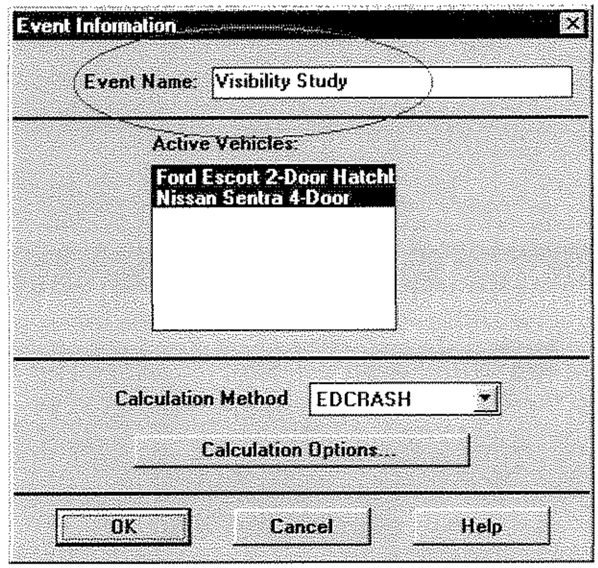


Figure 1: Save Case File dialog.

Each event within an HVE-2D case has its own individual Event Name. This name is entered when creating the event in the Event Information dialog (Figure 2). There is a 30 character limit on the length of the Event Name. Like the Case Title, this name is best used when it is descriptive enough to allow you to distinguish easily between events using the same Calculation Method. For instance, you might name events "Smith Vel=30", "Smith Vel=35", and "Smith Vel=40" in order to distinguish between three events in which the initial velocity of Smith was varied. HVE-2D will automatically attach the Calculation Method to the beginning of the Event Name that you enter in this dialog.



Figrue 2: Event Information dialog.

The Case Title and Event Name that you enter in these two dialogs will show up elsewhere in HVE-2D. The Case Title will be displayed on the main menu bar (Figure 3, next page) and the Event Name will be displayed in the header of the Event Window and the header of any Playback or Output Report Window for that event (Figure 3, next page).

Further, both the Case Title and the Event Name will appear on any alpha-numeric output reports that you print. For the EDCRASH tutorial, the left-side header will appear something like:

EDCRASH Tutorial, Visibility Study
Accident History-EDCRASH, Visibility Study
Licensed User: Engineering Dynamics Corporation

Above, the first line gives the Case Title. The second line gives the report type followed by the Event Name.

The easiest way to perform parametric studies without erasing your previous results is to use the Save As dialog (under the File menu) to save the file with a different name. You can then make changes in the new file, execute the event(s), and review the results without disturbing your original case. For example, lets say you are performing a reconstruction with EDCRASH and you are not terribly confident about your damage data. Therefore, you want to do three analyses: one with your best estimate, one with a realistic maximum, and one with a realistic minimum. You might first set up the case with your best estimate, complete with damage and scene data. Because setting up the event took a while and you only want to change a few variables, you do not want to re-enter all of that data to make the other two events. Go to the File menu and choose Save As. Enter a new Case File Name and new Case Title. You are limited to eight characters in the Case File Name, but your Case Title can be descriptive enough to differentiate easily between this and the other case. For instance, call the second case "Smith Intersection, Maximum Intrusion". Now you can go into the Damage Profiles dialog and increase the damage depths without affecting the original case. In this example, you would end up with three different case files.

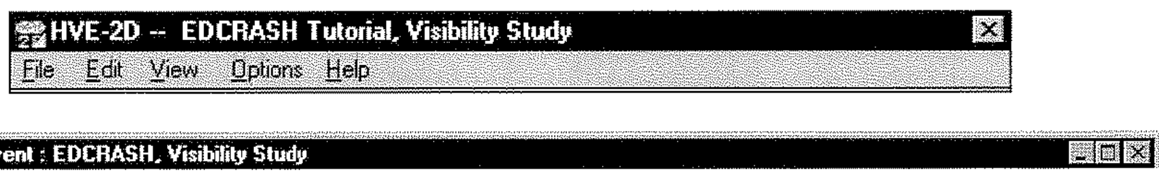


Figure 3: Main menu bar, above, and event window header, below.

HVE-2D FAQ's

This section contains answers to questions submitted to Technical Support by users of the new HVE-2D Version 1.32.

Q. What is the source for the tire data found in the Vehicle Editor?

A. All default tire data in the *HVE-2D* Vehicle Database is based on results from experimental studies performed by Calspan, UMTRI, or EDC. The majority of data for passenger vehicle tires is derived from tests performed by Calspan Corporation in 1983 [*Extended Tire Testing*; Tapia, G.A.; Report No. 6871-V-1, Nov. 1983]. These tires were tested on Calspan's flat-bed tire test machine. The majority of data for heavy truck tires is derived from tests performed at the University of Michigan in 1981 [unpublished data; University of Michigan Transportation Research Institute, 1981]. These tires were tested on UMTRI's flat bed tire test machine. Tire data for tires that were not tested by either Calspan or UMTRI are found by interpolating between actual test data. There is very little tire data available past the early 1980's. However, in 1999 EDC conducted a comprehensive test program of 21 unique tires at the Calspan tire test facilities. This data is available for purchase from EDC's Library section at www.edccorp.com.

Q. When I position vehicles during event set-up by using the manipulator I can only set the yaw angle to between –180 deg and 180 deg. What if I want the angle to be outside that range?

A. You have realized the importance of differentiating between –140 deg and 220 deg. In HVE-2D, the direction of rotation is inferred by the difference in heading angles between user-entered positions. In order to enter angles outside the range of –180 to 180, you must enter the yaw angle numerically in the Position/Velocity dialog window. Remember to press the Enter key after typing in your value.

Q. When I open the Damage Profile Preview Window for an *EDSMAC* simulation, the vehicle does not look damaged. Why?

A. You must open both a Trajectory Simulation and a Damage Profile Preview Window for your event in the Playback Editor to view vehicle damage. Similar to playing a Trajectory Simulation, when you press the Play button in the Playback Editor you will see the damage displayed as a function of time. More detailed instructions on viewing damage are included in the file HVE2d/damage.DOC.

Q. I notice that the size of case files saved in version 1.32 is much larger than version 1.2 cases. Why? Also, how can I reduce the size of case files?

A. If you save a case after executing the event, the output tracks (time-based data) now include the damage profiles, which is a large amount of additional data. If you want to minimize the size of your files to save space on your hard drive, press the reset button in the Event Editor before saving the file for the final time. This will erase your output tracks, but will not affect your event set-up. The next time you open the file again, simply run the event to recreate the output tracks.

Q. I am using Windows NT as the operating system for my PC, and sometimes I can't enter values into the driver control tables (e.g. throttle, brakes, and steering) or the *EDCRASH* damage profile crush depth table. Why?

A. When using the NT operating system, these tables may sometimes stop responding to user inputs. This problem is related to *HVE-2D's* ability to run in Windows 3.1 (a 16-bit operating system). The workaround for this problem is to save your file and exit *HVE-2D*, then shut down and restart your computer. You can then restart *HVE-2D* and pick up where you left off.

Technical Session

The *EDVSM* model has been recently extended to include force and moments created by interaction between the vehicle body and environment terrain. Thus, the kinetic simulation of a complete vehicle rollover is now possible. Our Technical Session describes this model.

The general modeling approach used by the *EDVSM* body vs. terrain model is similar to the "hard point" method used by HVOSM-RD2. The vehicle body is comprised of nodes that may interact with the environment terrain. The nodes have material attributes and, thus, generate forces as a result of their interaction with the terrain. The forces are resolved into vehicle-fixed components, which are then summed to calculate forces and moments acting at the center of gravity of the sprung mass. These forces and moments are included in the equations of motion, along with suspension forces and aerodynamic forces, to calculate the current sprung mass linear and angular acceleration vectors.

Node Definition

The nodes required by the model are the vertices supplied by the vehicle 3-D geometry file. Since every *HVE* vehicle has a geometry file, every vehicle has the nodes required to perform the simulation. A vehicle may have an unlimited number of nodes. The vehicle-fixed coordinates for each node are inherently included in the geometry file, as is the number of nodes.

Force and deformation can occur only at node coordinate locations, thus, the number of nodes determines the resolution of the model. For example, a Generic Class 1 Passenger Car is comprised of 20 vertices. Thus, forces can be produced only at 20 locations on the vehicle (see Figure 1). On the other hand, a 1998 Ford Taurus 4-Dr Sedan in the EDC Vehicle Database is comprised of 2792 vertices (not including tires and wheels) (see Figure 2).. Thus, forces can be produced at 2792 locations on the Ford Taurus. The resulting analysis can include significantly more detail

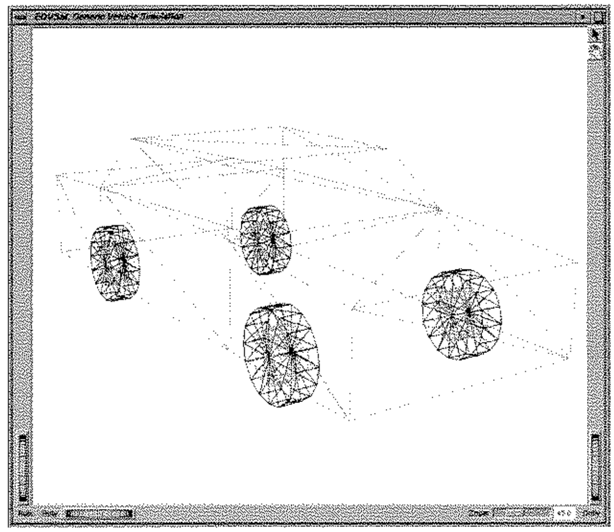


Figure 1 – Generic Vehicle (Passenger Car) in Wireframe. Vehicle has 20 nodes (vertices).

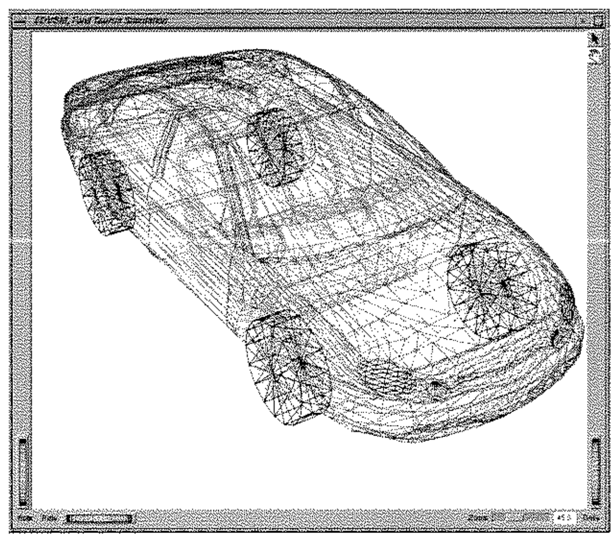


Figure 2 – 1998 Ford Taurus in Wireframe. Vehicle has 2792 nodes (vertices).

Node Material Attributes

The *HVE* simulation environment supplies each node with a complete set of material attributes; the attributes are user-editable. However, the *EDVSM* body model uses only the exterior stiffness, K_v, and vertex friction, μ_v. The material attributes are omni-directional (i.e., force-displacement characteristics of the node are independent of the direction of deformation).

The material attributes are not assigned according to the area surrounding each vertex. Routines of significantly greater complexity are required for such an approach (e.g., DyMESH). Thus, closely spaced vertices will create a harder region than widely spaced vertices. To some degree, this correlates with one's intuition because complex regions comprised of many vertices, such as door pillars, tend to he harder than flat areas comprised of fewer vertices, such as door panels. However, the user should be aware of this characteristic when assigning stiffness values.

Because node force is not assigned according to the area surrounding the vertex, the proper selection of K_v depends on the number of vertices included in the vehicle geometry. Experience has shown that a value of 15-25 lb/in^2 (actually interpreted as lb/vertex) is reasonable for vehicles having 1000 to 3000 vertices. Because the forces are impulsive, the effect of selecting a lower value of K_v is to decrease the peak force and increase the deformation and duration of impact. Selecting a higher stiffness generally increases the peak force and decreases the deformation and duration of impact. In any case, the impulse (area under the force vs time curve) is approximately the same over a rather wide range of stiffness values. The basic approach is to select a K_v value that results in a crush depth that approximates the actual crush.

K_v is assigned using the *HVE* Vehicle Editor by clicking on the desired surface icon (*front, right, back, left, top* or *bottom*), choosing the Stiffness dialog and entering the desired K_v stiffness value (*A* and *B* stiffnesses are also available in the dialog, but are not used by the *EDVSM* body model).

The default body-terrain friction coefficient is 0.50. The body-terrain friction may be modified using environment friction zones.

Node Force

The force at each node (or vertex) is calculated from the penetration of the node into the surface terrain. An *HVE* function, called `GetSurfaceInfo()`, is used to perform this calculation (see Figure 3), the same as for calculating tire radial force. These calculations require that each vertex be transformed from the

vehicle-fixed coordinate system to the earth-fixed coordinate system. The node deformation is then calculated in the earth-fixed coordinate system (see Figure 4). The earth-fixed node velocity is then calculated to determine the direction of the friction force. The earth-fixed normal component of the node deformation is calculated and applied to the friction coefficient to determine the frictional force component. Ten percent restitution is applied during unloading. Finally, the total force is calculated in the earth-fixed coordinate system and resolved in the vehicle-fixed coordinate system to be applied to the vehicle sprung mass (Figure 5). This procedure is performed for each vertex in the vehicle geometry file.

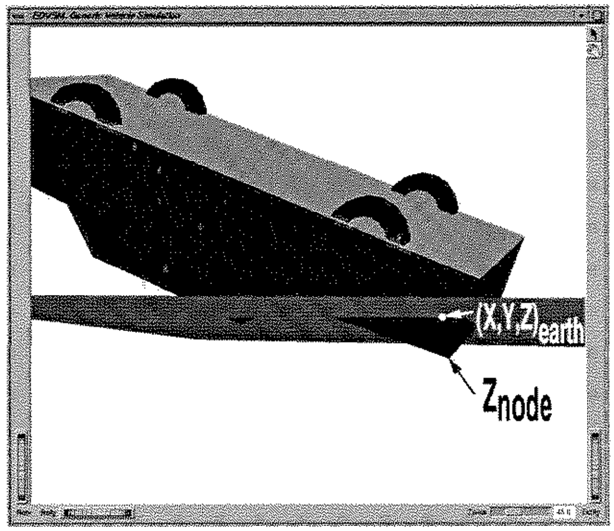


Figure 3 – Body vs ground interaction using `GetSurfaceInfo()`

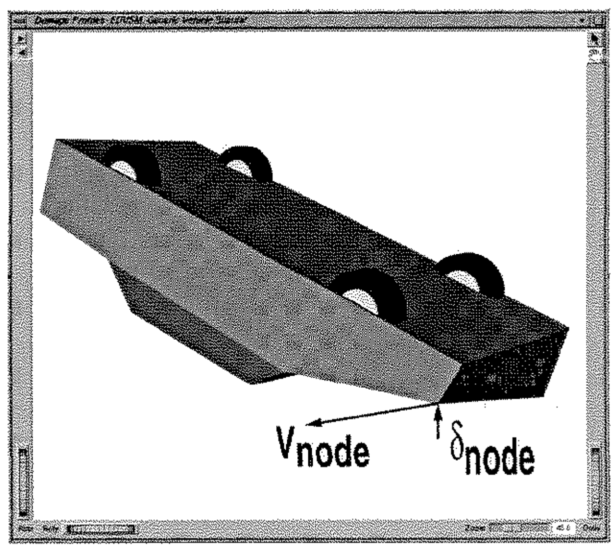


Figure 4 – Node deformation (displacement) and velocity

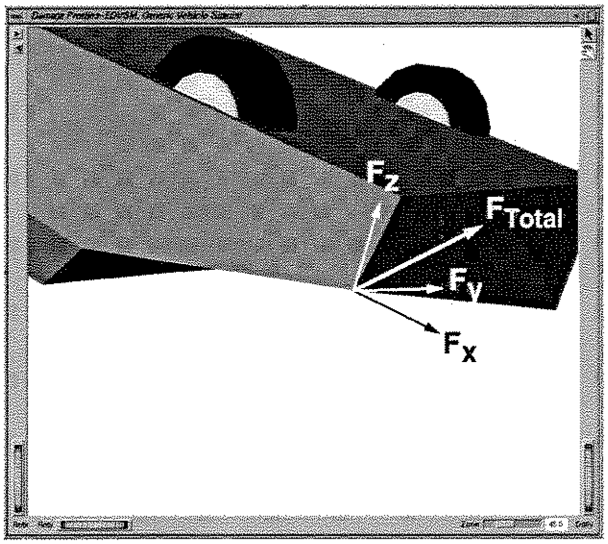


Figure 5 – Vehicle-fixed node force components

Forces and Moments on Sprung Mass

Once the 3-dimensional force components are known for each vertex location, it is a simple matter to calculate the total forces and moments on the sprung mass:

$$F_{x_{Vehicle}} = \sum_{i=l}^{NumNodes} F_{x_{Node}}$$

$$F_{y_{Vehicle}} = \sum_{i=l}^{NumNodes} F_{y_{Node}}$$

$$F_{z_{Vehicle}} = \sum_{i=l}^{NumNodes} F_{z_{Node}}$$

$$F_{x_{Vehicle}} = \sum_{i=l}^{NumNodes} (F_{z_{Node}} y_{Node} - F_{y_{Node}} z_{Node})$$

$$F_{y_{Vehicle}} = \sum_{i=l}^{NumNodes} (F_{x_{Node}} z_{Node} - F_{z_{Node}} x_{Node})$$

$$F_{z_{Vehicle}} = \sum_{i=l}^{NumNodes} (F_{y_{Node}} x_{Node} - F_{x_{Node}} y_{Node})$$

Note that the vertex *x,y,z* coordinates used in the above calculations are based on the current node deformation. This is important for two reasons:

First, current node coordinates define the moment arm for a node's moment calculations, thus, current deformation must be considered, and second, if the vehicle rolls twice (or more) on the same portion of the vehicle, the calculation procedure must remember the deformation associated with previous contact.

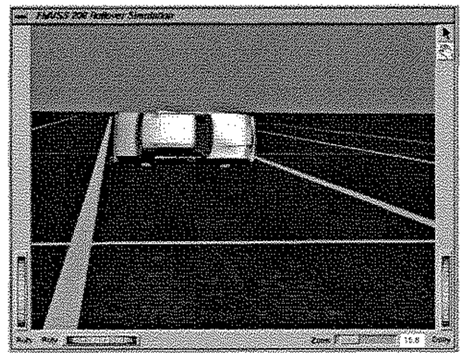


Figure 6

6=0.9 sec Following Launch

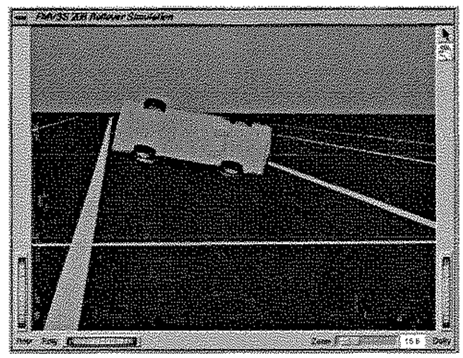


Figure 7

t=1.4 sec

Figure 8

t–3.0sec

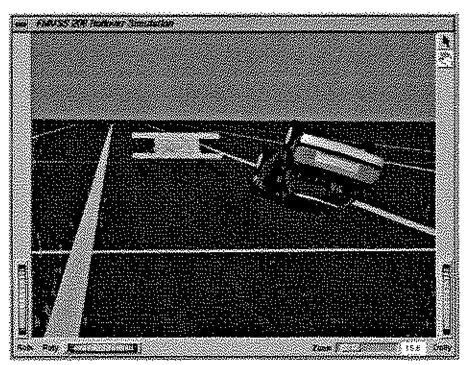


Figure 9

6=5.1 sec (~rest)

Frames from simulation of FMVSS 208 Rollover Test.

A full description of the model was presented in March at the SAE International Congress and Exposition (SAE Paper No. 2000-01-0852; please contact EDC Customer Service if you would like to receive a free copy of the paper). The paper includes several examples of the use of 3-dimensional rollover simulation. One of the examples is an FMVSS rollover test (see Figures 6 though 10). Other examples include handling studies and real-world rollover crashes.

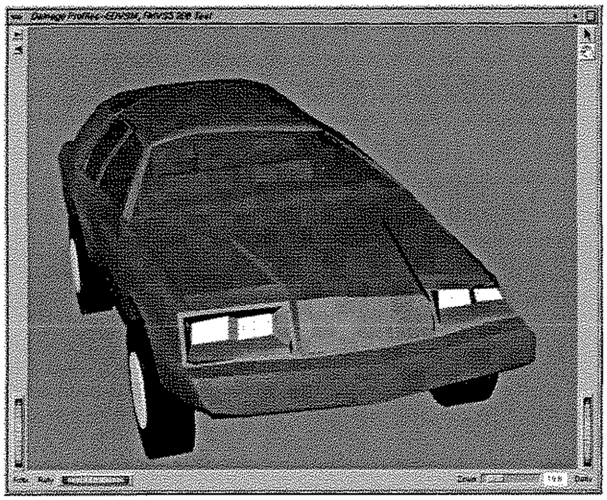


Figure 10 – Simulated vehicle damage resulting from FMVSS Rollover Test.

One final word of caution: Just because it can now be done does not mean it is easy! Although a typical rollover simulation performed for illustrative purposes may take little time (say, about an hour), it can take a significant amount of time to duplicate the exact number of rolls and vehicle damage from an actual case study or experiment.

User Hints and Tips

HVE

- In the 3-D Editor, move the cross-hair selector into any of the planar views, click the right mouse button, and choose Decoration in order to get the translate and zoom dials and the tool buttons for that window (same as default in the perspective view).
- In all editors, push the Esc key in order to toggle between arrow-cursor mode for selecting objects (e.g. vehicle wheel) and hand-cursor mode for changing views. When the cursor is in hand-mode, press the left mouse button to rotate an object, the center mouse button to translate an object, and both the left and center mouse buttons simultaneously to dolly in or out.
- In order to export data from the Variable Output report and import it into a spreadsheet or graphing program, first, choose File – Print from the main menu bar. In the Print dialog, select Print To File. If your spreadsheet will be more than 1 page long, do not Print Page Headers. Leave the font as Courier. Click OK. In the Printer File Selection dialog, choose the File Format to be Text. Enter a File Name, and click OK. This file can be imported into a spreadsheet program as either fixed-width or tab and space delimited text.

HVE-2D

- In order to export data from the Variable Output report and import it into a spreadsheet or graphing program, first click the Print button in the Variable Output window. In the Print dialog, click Setup. In the Print Setup dialog, under Specific printer, choose Generic/Text Only on FILE:, and click OK. Click OK again in the Print dialog. Enter a file name and location in the Print To File dialog and click OK. This file can be imported into a spreadsheet program as either fixed-width or tab and space delimited text.
- When importing AutoCAD drawing files into *HVE*-2D, it is important that you understand how *HVE*-2D determines the origin. There are two coordinate systems in AutoCAD: a fixed coordinate system called the World Coordinate System (WCS), and a movable coordinate system called the user coordinate system (UCS). *HVE*-2D recognizes only the World Coordinate System origin.

  The WCS origin is located in the lower left corner of the drawing where the X and Y axes intersect (0,0). The UCS is used to set up a new origin within the WCS. Moving the UCS can make it easier to work on particular sections of a drawing.

  In order to return to the WCS, follow these instructions in AutoCAD.

  1. From the Tools menu, choose UCS, and then Named UCS.
  2. In the UCS Control dialog box, select *WORLD*.
  3. Select Current
  4. Click OK.

  Once your drawing is in the WCS, you can select all objects in the drawing and move them so that your intended origin in *HVE*-2D corresponds with the AutoCAD WCS origin.
- The scale at which you must import AutoCAD drawings into *HVE*-2D is decimal inches. *HVE* expects all information to be entered in what are called Program units. These are the units that are used for computations within *HVE*. Program units consist of the English units pounds, seconds, inches, and derivatives thereof. Vehicles and environments drawn to a metric scale must be converted to inches before importing (1 inch = 0.0254 meter). If you import a drawing and it appears very small with respect to an *HVE* vehicle, it is likely that your drawing is in the units of feet. If this is the case, scale the drawing by 12 before importing it into *HVE*-2D.

Newton on Motorcycle Collisions

A frequent question at the EDC customer technical support desk is "Can I use HVE to study motorcycle collisions?" During the pre-impact phase, HVE is perfect for producing a time-distance study to determine avoidability. You get a *free* visibility study to boot! But, what about the collision? To get the best possible answer, we decided to ask the man himself, Sir Isaac Newton.

EDC: Thank you for coming back for this interview. So tell us, can we study the dynamics of a vehicle vs. motorcycle collision?

SIN: In a word, *NO*. I mean, you can do the calculations, but the result is worthless!

EDC: Why is that?

SIN: If you believe in my laws of motion, the answer lies in my second and third laws.

EDC: Please explain.

SIN: OK, according to my third law, the collision force on the vehicle is the same as the force on the motorcycle, but in opposite directions. Now, according to my second law, the force on each object is equal to the product of mass times acceleration. In equation form,

$$F_1 = -F_2 \quad (3^{rd} \text{ law})$$

and

$$F_1 = m_1 a_1 \quad and \quad F_2 = m_2 a_2 \quad (2^{nd} \text{ law})$$

Substituting for F_1 and F_2,

$$m_1 a_1 = -m_2 a_2$$

Now, since $a = dV/dt$ (thank goodness I also invented the calculus!), or $a = \Delta V/\Delta t$ for the entire collision interval, we can again substitute

$$m_1 \frac{\Delta V_1}{\Delta t_1} = -m_2 \frac{\Delta V_2}{\Delta t_2}$$

Since Δt is the same for both the motorcycle and the vehicle,

$$m_1 \Delta V_1 = -m_2 \Delta V_2$$

In the above equation lies the reason that the calculation of speed change for a motorcycle collision is worthless. A simple rearrangement results in

$$\frac{m_1}{m_2} = -\frac{\Delta V_2}{\Delta V_1}$$

Or, stated in words, the ratio of the speed changes during the collision is inversely proportional to the ratio of the masses. Note that I didn't make any assumptions here, other than assuming my laws are valid (you can ignore Dr. Einstein, unless your vehicles are traveling at the speed of light). So the above result is true for every collision throughout the universe – whether you're talking about a car hitting a pickup or a motorcycle hitting a car. And it doesn't matter if you're using a hand-held calculator, a fancy computer program or the back of an envelope.

EDC: Sounds good. So, what's the problem?

SIN: It's the *mass ratio*! When analyzing anything, one should always study the sensitivity of the results to the inputs. If you perform such a study by varying the ranges of the estimated input parameters (such as stiffness, crush, approach and departure directions – not to mention the fact that the motorcycle's mass changes during the collision when its rider flies off) you'll find that the range of possible speed change for the car (heavier object) is quite reasonable (say 2 to 5 mph), but the range for the motorcycle (lighter object) is huge! Say, 20 to 50 mph. So, if you use such an approach, the findings in your report would be something like, "The speed of the Cadillac at impact was 35 to 40 mph and the speed of the motorcycle at impact was 25 to 75 mph." Now, if that's helpful…

EDC: Now I get it. The calculation is valid, but it's too sensitive to be useful. But what about studying a collision between a car and a fully loaded semi-truck? Don't you have the same problem?

SIN: You sure do! Now, I have a question for you.

EDC: Fire away.

SIN: What's a motorcycle?

Contributor's Corner

By James Sneddon
Senior Consultant and Lecturer at the
Northwestern University Traffic Institute

CREATING VIDEOS WITH HVE-2D

Videos of an *HVE*-2D simulation can be created with your PC and videocassette recorder. The videotape is an effective exhibit for trials, depositions, hearings and settlement proceedings.

To create a video, you will need to convert the computer's VGA signal to composite video or S-Video (Super Video). This will require special hardware. Peripheral devices such as TV View Gold and Iomega Buzz connect to your PC's parallel port or monitor output. The video output from these peripherals is connected to your videocassette recorder.

The introduction of DVD-ROM drives has spawned a new generation of PC video cards that include video-out. This new technology comes standard on many new PC's, and permits connecting a television or VCR directly to your PC. Due to the high resolution of DVD, S-Video is the preferred format. S-Video contains 400 lines of resolution, compared to 240 lines for standard video. To record this signal, a SVHS videocassette recorder is required. These models are less common than standard VHS VCR's, but models are available for around $300.

Consider purchasing a VCR with a flying erase head and editing capabilities. This will prevent the "snow" effect when stopping or pausing the tape during recording.

Most SVHS videocassette recorders can record on both SVHS and standard VHS tapes. However, the SVHS tape can only be played in a SVHS videocassette recorder. Therefore, it may be necessary to dub the SVHS recording to standard VHS for distribution to attorneys and use in court.

Routing the video signal to your VCR can be hardware or software controlled, and may require changing your display properties. Since these procedures vary by device, familiarize yourself with your hardware.

Your video should display only the simulation view of the Playback Window. However, most video output hardware will display the entire desktop. Therefore, the taskbar, wallpaper, and all windows and icons will appear in your video unless you maximize the Playback Window. Many users have noticed "graffiti" when either maximizing or resizing the Playback Window. A black border appears instead of the "graffiti" when operating under Windows NT. This can be corrected by resetting the monitor resolution.

First, ensure that your monitor display resolution is set to 1024 by 768 pixels or greater. Check the display properties by double-clicking the display icon in the Windows Control Panel, or move the mouse to a blank area of your desktop and click the right mouse button. Select *Properties* from the pop-up menu.

The display properties window will appear. Click on the *Settings* tab, and check the screen area setting. Move the slider to the right to increase the resolution to 1024 by 768 pixels if necessary. Not all graphics cards support this resolution. If that is the case with your PC, you will not be able to move the slider past 800 by 600 pixels (or 640 by 480 pixels). If this occurs, check the color setting immediately to the left of the screen area settings. Reset the colors setting to *256 Colors* if not already set. Reducing the color setting may permit a higher screen resolution. Try again to set the screen resolution to 1024 by 768 pixels.

Next, open the Playback Window in *HVE*-2D. The size of the Playback Window will be proportional to the screen area at the time is it opened. With the Playback Window open, reset the screen area to 640 by 480 pixels by using the procedure previously described. The Playback Window will now be larger than the desktop, but can be maximized without the "graffiti" appearing. The maximize button near the right side of the title bar may now be off the desktop. You can still maximize the window by double-clicking the title bar, or click the icon in the upper left corner of the window, and select *maximize* from the pop-up menu. Scale and center the view in the Playback Window as desired for your video.

The Windows taskbar will also appear in your video unless you hide it from view. To hide the taskbar, move the mouse pointer over the edge of the taskbar. As it passes over the edge, the pointer will change from a normal select (single arrow) to a vertical resize (double arrow). With the vertical resize pointer visible, click and drag the top of the taskbar downward. The taskbar will change to a thin gray line. To restore the taskbar, move the pointer over the taskbar until the vertical resize pointer appears. Click and drag the taskbar upward until the taskbar returns.

To play the simulation in the *HVE*-2D Playback Window, the play button must be clicked in the Playback Editor. However, with the Playback Window maximized, the playback editor is not visible. This requires maximizing the Playback Window after clicking the play button in the Playback Editor. To prevent the simulation from starting immediately, delay the start of the simulation. An initial time delay of five seconds usually provides sufficient time.

Enter an initial time delay by inputting a value for *dt* (T_{delay}) in the Variable Output Window. This will have to be done for each active trajectory simulation in the Playback Window. If a trajectory simulation is already delayed, add the initial time delay to the current value for *dt.*

With an output signal routed to the VCR, click the play button in the Playback Editor; maximize the Playback Window; move the mouse pointer off the desktop (else it will appear in your video); and press record button on the VCR. These actions must be completed, and the VCR must be recording before the simulation begins. Use the pause function on the VCR to ensure that it begins recording as soon as the record button is depressed, and to provide seamless transition between views.

Your selection of colors for the vehicles and environment will affect the quality of your video. Pure colors such as red, blue and green tend to "bleed" into the adjacent pixels when viewed on a television. The vehicles appear to glow on the TV screen. To minimize this effect, select darker colors for the vehicles. In the vehicle viewer, click on the center of mass, and select Color from the pop-up menu. A pallet of forty-eight basic colors will appear. This pallet includes darker colors that will work better when creating videos. Darker colors should also be used when importing an environment file.

Use presentation software such as PowerPoint to add title and narrative slides to your video. You can display slides in "full screen," and record them to videotape. Then pause the video, and switch to *HVE*-2D to record your simulation. Your final product will be a professional looking presentation.

Technical Session

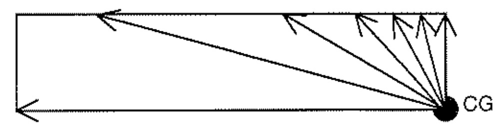
One of the most common questions we receive regarding *EDSMAC* is how to simulate a collision with a rigid barrier (e.g., tree or bridge abutment).

First, it is important to understand that the original *SMAC* collision algorithm was not designed to accommodate barrier collisions (barrier being defined as an infinitely rigid object that absorbs no energy during impact). The reason is clear upon close examination of the collision algorithm: it requires that both colliding objects be deformable. The algorithm adjusts the deformation between the two colliding objects until the forces balance between them. Clearly, any adjustment of the deformation of an infinitely rigid object will result in an infinitely large change in force; thus, a force balance cannot be achieved. (For a detailed development of the collision algorithm, see SAE Papers No. 880069 and 1999-01-0102.) Therefore, if you set up a collision between a vehicle and an infinitely rigid movable barrier, *EDSMAC* will terminate as soon as the vehicles collide, displaying an error message at the beginning of the collision phase (Event Termination: Too Many Collision Vector Adjustments).

Furthermore, you cannot select a fixed barrier as one of your vehicles in an *EDSMAC* simulation. Because a fixed barrier has neither wheels nor axles you may not select a fixed barrier as one of the vehicles. If you try, *EDSMAC* will display an error message (Incompatible Vehicle(s) – Number of Axles Not Supported). You cannot even create the event.

This being said, *EDSMAC* may often be used successfully for barrier collisions and pole impacts by making certain adjustments. These adjustments are as follows:

- To create the barrier, start by choosing any vehicle or moveable barrier from the vehicle database.
- Change the vehicle's dimensions as required. Because the *EDSMAC* algorithm creates collision damage vectors emanating from the vehicle CG at equally spaced increments (defined in Calculation Options – Vector Spacing (deg)), barrier aspect ratio is important to consider. For a long and skinny barrier (e.g. a guardrail), distance between collision force vector tips along the vehicle exterior increases dramatically as vector length increases (see illustration below). In *HVE-2D*, it is advisable to use a barrier with an aspect ratio closer to 1.0 even though it may not always "look right". *HVE*, however, allows the user to define vehicle/barrier dimensions independently of the vehicle geometry file. Therefore, you may create a realistic looking, yet mathematically square barrier.



- Assign a frontal stiffness, K_v, of 1000 lb/in^2.
- If the barrier that you are modeling is, in reality, a fixed barrier, increase the simulated barrier's weight to 1,000,000 lb and its yaw inertia to 10,000,000 lb-sec^2-in in order to help prevent its movement during impact.
- If the barrier is small (e.g., a pole or tree), select the simulated barrier as the first vehicle when creating the *EDSMAC* event. This creates better resolution in the collision model.
- In *EDSMAC's* Calculation Options dialog, reduce the Vector Adjustment Increment to 0.02 in (from the default value, 0.2 in), and increase the Vector Force Tolerance to 25 lb/in (from its default value, 15 lb/in).
- For fixed barriers, select the Driver Controls - Brakes dialog and assign a large braking value (i.e., Percent Available Friction=1.0) to further help prevent movement.
- Finally, set up the simulation so the first timestep involving a collision force occurs with a small overlap between the vehicle and barrier. This may be accomplished by reducing the trajectory and separation integration timesteps or by adjusting the initial positions as required for a small overlap at the start of the collision. This is particularly important for small barriers, such as trees and poles. Otherwise, the pole may be completely engulfed by the vehicle before the collision is detected, resulting in a fatal error during the collision phase.

Limitations

The above steps help *EDSMAC* work within the basic assumptions of the collision algorithm. However, this does not always work. If a fatal collision phase error occurs, try reducing the stiffness by 100 lb/in^2. Note that the barrier will deform slightly; as explained, this is necessary in order to equalize the force between the vehicle and barrier.

Also, pole impacts (actually, any small barrier) are more difficult to simulate than larger barriers because of the barrier size. Reviewing the technical background for the collision algorithm reveals that the problem is due to the small number of collision vectors interacting with the barrier. In exceptional cases the pole may actually lie *between* collision vectors and no interaction will occur. Consider decreasing Vector Spacing in the Calculation Options dialog for such cases.

The best suggestion is to carefully set up and execute the *EDSMAC* barrier simulation. If a problem occurs, first confirm that all the above suggestions have been followed. Sometimes a small adjustment in the inputs solves the problem.

A note to *EDSMAC4* users: Fixed barriers are allowed. Also, barrier stiffnesses are automatically assigned (A = B = 1000.0); the Vector Adjustment Increment is reduced to 0.01 inches; and the number of iterations in the force balance loop is increased from 200 to 3000. These and other changes in the *EDSMAC4* algorithm are described in SAE Paper No. 1999-01-0102.

User Hints and Tips

HVE

- If you would like to make a video/JPEG of an event in slow motion, simply reduce the playback interval (found under Options – Playback on your main menu bar) before recording a JPEG movie. For instance, if you want the scene to play at one-half real time, decrease the playback interval from 0.0333 seconds (30 frames for 1 second of time) to 0.0167 seconds (60 frames for 1 second of time). The JPEG will still play at 30 frames per second, so your desired result is achieved.

- When adding vehicles to a case, it is preferable to add vehicles from the vehicle database as new vehicles, rather than directly from previous case files. This is because vehicles used in previous cases may not have the same data structure that is used in *HVE* version 2.2x cases. Also, you may not clearly remember the changes you made to those vehicles. If you have created vehicles that you would like to use in the future, simply save these vehicles into your user vehicle database while working in the specific case. You will then be able to pull these vehicles from the vehicle database.

HVE-2D

- If you accidentally create the wrong position for a vehicle when setting up an *EDCRASH* event, you can now delete this position in *HVE-2D* version 1.3. Simply reselect the position from the "Edit - Positive/Velocity" choices on your main menu toolbar and then "Edit – Delete". The vehicle image corresponding to that position will disappear from your work.

- It can be very useful to position target vehicle 'ghost images' to indicate impact positions, rest positions, or other important points when trying to match *EDSMAC* simulations to scene data.

- The origin of an environment drawing imported from AutoCAD into *HVE-2D* will be the origin of the World Coordinate System, not the User Coordinate System. It is best to make this origin near where the center of "the action" is in the reconstruction or simulation. When you click on the Center button in an Event, Preview, or Playback Window, the drawing origin will be moved to the center of the window.

HVE and HVE-2D Tips

- Using uniquely colored vehicles makes it easier to identify which vehicle is which, especially in complicated *EDCRASH* events where you may have 6 or more total vehicle positions on the site drawing.

- You can save your case files in unexecuted form in order to minimize the file size. To do this, press the reset button in the Event Editor prior to saving your case for the final time. This will remove all output tracks from your case file. NOTE: With the size of hard drives on today's computers, you may not need to even worry about the size of your case files.

- When positioning vehicles in *EDCRASH*, remember that 200° and –160 ° do not have the same meaning. *EDCRASH* uses these angles to determine direction of rotation by subtracting the final/rest heading from the impact heading. If this angle is positive, the vehicle rotated clockwise. Likewise, if negative, the vehicle rotated counterclockwise.

Bring you own tips to share, and learn from others, at the HVE Forum, May 8-12 in San Diego.

HVE-2D FAQ's

This section contains answers to questions submitted to Technical Support by users of the new HVE-2D version 1.3.

Q. If I try to run an *EDCRASH* event without specifying damage profiles for one or both vehicles, I get an error message about a floating point error. Why?

A. This problem is resolved by downloading a *HVE-2D* physics software patch from the EDC ftp site. Detailed instructions on how to download this file have been mailed to all *HVE-2D* users who have received *HVE-2D* version 1.3. Please contact Technical Support if you need further assistance.

Q. When I open the Damage Profile Preview Window for an *EDSMAC* simulation, the vehicle does not look damaged. Why?

A. You must open both a Trajectory Simulation and a Damage Profile Preview Window for your event in the Playback Editor to view properly vehicle damage. Similar to playing a Trajectory Simulation, when you press the Play button in the Playback Editor you will see the damage displayed as a function of time. More detailed instructions on viewing damage are included in the file hve2d/damage.DOC.

Q. When I am in the Vehicle Editor and I click on the CG of the vehicle, I notice that "Import Vehicle Drawing" is now grayed out (compared to *HVE-2D* version 1.2). Why?

A. With the implementation of vehicle damage profile viewing for *EDSMAC* events, the vehicle images used in the Event and Playback Editors had to be changed to specific discretized vehicle meshes. These vehicle images are scaled based upon vehicle dimensions entered by the user in the Vehicle Editor. It is no longer possible to import and use your own vehicle images in *HVE-2D* version 1.3.

Technical Session

HVE and HVE-2D are designed to allow the user maximum flexibility in data input and output. An example of this flexibility is the ability to manipulate user units, number of decimal places, and displayed variable names in the language.rsc and units.si/units.us files. Any changes will show up in both the input dialogs and the output reports and graphs. Another example is in the definition of warning and minimum/maximum variable limits. These features are intended to serve as a check to make sure that values entered by the user are reasonable. While reasonable default values are given, you may want to make changes to the units, variable names, limits, etc. for exceptional cases.

You may, for instance, wish to increase the maximum simulation time (termination condition) beyond 20 seconds and change the units from seconds to milliseconds. We will make those changes for our example.

First, you must edit the language.rsc file in order to increase the maximum simulation time. In *HVE*, this file is located in the supportFiles/sys/ directory. In *HVE-2D*, this file is located in the Extensio/ directory. It is advisable to make a copy of the language file before editing it, so that you can always go back to the original. You may use any text editor to do the editing, but you must make sure to save the file as Text. Once you open the file, find the line corresponding to the variable for which you would like to make the changes. It is easiest to search for the user interface display name, in this case "Maximum Time", in order to locate quickly the correct lines in the file.

```
Numeric OpSimMaxTime SimulationControls
   UtTime sec
   1.0   20.0  1.0   20.0  5.2f
   "Maximum Time"
```

The first section describes the variable:

Variable Type = Numeric
Variable Name = OpSimMaxTime
Dialog Name = SimulationControls
Unit Name = UtTime
Program Unit = sec

The second section sets the minimum and maximum variable values for the warnings and the limits (Figure 1). It also sets the number of significant digits in the numeric output.

Warning minimum = 1.0
Warning maximum = 20.0
Limit minimum = 1.0
Limit maximum = 20.0
Total characters = 5
Number of decimal places = 2
Field type = f (floating point)

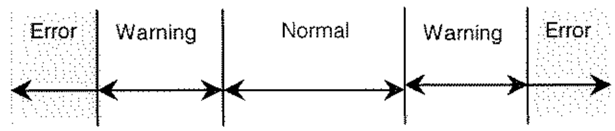


Figure 1: Normal, Warning, and Error Ranges allow HVE and HVE-2D to give the user feedback as to how reasonable a variable's value is under most conditions.

Finally, the display string of the variable is listed.

Display String = "Maximum Time"

After you make the changes to increase the maximum simulation time from 20 to 150 seconds, the section will look like:

```
Numeric OpSimMaxTime SimulationControls
   UtTime sec
   1.0   150.0  1.0   150.0  9.2f
   "Maximum Time"
```

Both the limit and warning time maximum values are now set to 150.0 seconds. In this case it is necessary to make the warning limits equal to the error limits because the user interface uses a "slider" (Figure 2). If the user interface used just a box, the values would not have to be equal. Also, the total characters for numerical output was increased to 9 to accommodate both the additional time and the change to milliseconds.

Figure 2: The graphical user interfaces of HVE and HVE-2D use a slider and a box for inputting maximum simulation time.

Next, change the units from seconds to milliseconds by editing both the units and the language files. First, look at the bottom of the language file. Useful conversion factors are given. The user may add any conversion that is necessary. To convert from seconds to milliseconds, the necessary line is:

```
Convert   sec   msec    1000.0
```

where

Operation = Convert
Program Unit = sec
User Unit = msec
Conversion Factor = 1000.0

Note that the program unit for time is always seconds. Therefore, whether you want to use milliseconds, decades, scores, or millennia in the HVE or HVE-2D interface, you must always convert the user units to the program units, seconds.

The last step is editing the units files in order to specify the new user unit. You will notice that there are three units files in the sys/ (HVE) or Extensio/ (HVE-2D) directory: *units.hh*, *units.si*, and *units.us*. Do not edit the *units.hh* directory; HVE will no longer run. *units.si* specifies the SI user units, and *units.us* specifies the US units for program input and output. To be thorough, edit the line for the time unit (as specified in the language file) in both units files so that milliseconds will be used for both SI and US units.

```
UtTime,     "msec"
```

All time units will now be in milliseconds and the simulation will terminate after 150,000.00 milliseconds (150 seconds).

Remember that these files are accessed every time you use HVE or HVE-2D. Therefore, any changes that you make will remain next time you use the software unless you change the files back to their original form.

Because of the openness of this programming structure, there is no limit on the units that the user may introduce. All that is necessary is the conversion factor. The conversion factor is given in the *language.rsc* file and the instructions to use that unit are given in the *units.us* and *units.si* files.

Technical Session

Our Technical Session this time provides a complete example of using the HVE Brake Designer to evaluate the temperature rise and stroke increase in a highway truck fitted with air drum S-cam actuated brakes.

In our example, a loaded tractor-trailer is negotiating a 6 % downgrade. All brakes are adjusted to within allowable tolerance. The initial speed is 65 mph and the brakes are applied hard enough to approximately maintain that speed (+/- 5 mph).

The vehicle in our study is a 1995 Freightliner FLD-120 with a sleeper. It is towing a 45-ft Class 4 semi-trailer with a 17,600 lb payload. The tractor is fitted with Type 20 15x3.5 wedge brakes on the front axle. All other brakes on the tractor and trailer are Type 30 16.5x7 S-cam drum brakes. Slack adjuster lengths are 6.5 inches. The linings are SAE FF rating, with nominal friction coefficients of 0.35 at both low temperature and high temperature. The HVE Brake Designer dialog for this brake assembly is shown in Figure 1.

EDVDS is used to perform the simulation. First, increase maximum simulation time and maximum time for brake pedal force input to 150 seconds. Using the HVE Event Editor, use the Position/Velocity dialog to place the vehicle at the top of a 6 percent downgrade, and enter an initial velocity of 65 mph. Supply a trailer payload of 17,600 lb, and rotational inertias of 1,000,000 lb-sec^2-in. Precise values for rotational inertia are not important because our vehicle isn't turning.

Now, fill in the brake pedal force table so that the vehicle stays within 5 mph of the initial speed. This will take a bit of iterating. For air brake systems, the pedal force is the same as the treadle pressure.

For each wheel on the tractor and trailer, use the Brake dialog to change the initial lining and drum temperatures to 100 degrees from the default value, 68 degrees, and change the initial stroke to 2.00 inches from the default value, 1.5 inches (Figure 2). Input a convection coefficient of 37 $\text{in}-\text{lb}/\left(\text{in}^2\ \text{hr degF}\right)$.

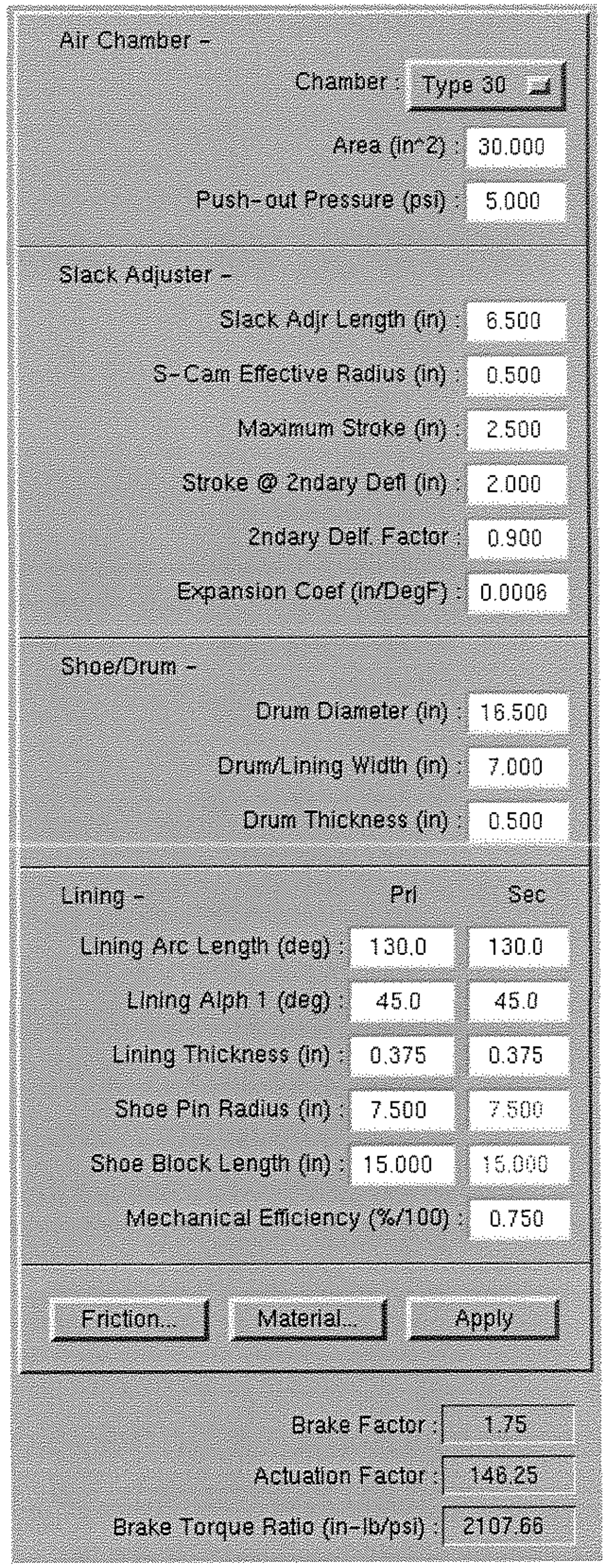


Figure 1: S-Cam Brake Designer Dialog

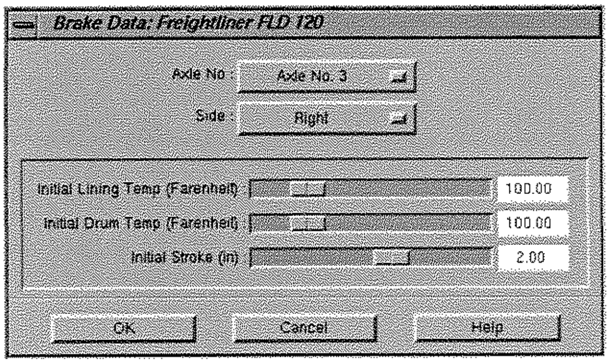


Figure 2: Initial Conditions

Now execute the run, and select total velocity, brake stroke, brake torque, and temperatures (lining, drum and interface) from the Key Results Variable Selection dialog.

The brake component temperature increase is steady throughout the simulation (Figure 3) and has a dramatic effect on braking performance. Brake drum and drum-lining interface temperatures increase substantially faster than lining temperature because brake lining is a relatively poor heat conductor. Notice how long it takes for the temperatures to increase substantially.

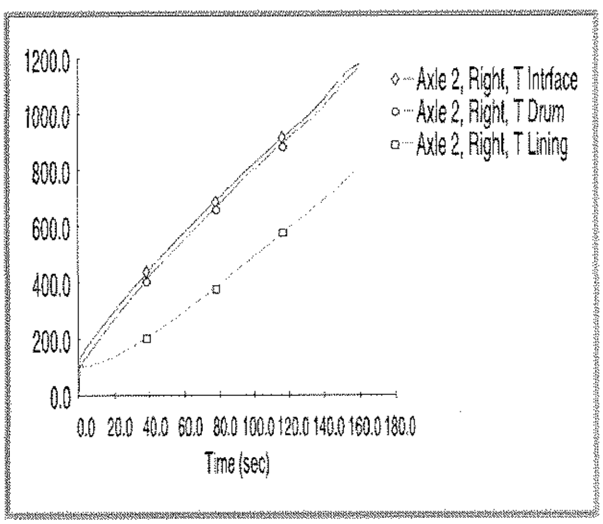


Figure 3: Brake Component Temperatures

Because the initial stroke is so large, brake performance is compromised. After the pedal force is applied, brake stroke immediately increases to 2.1 inches and increases linearly for the rest of the simulation to the maximum of 2.5 inches. There is a concomitant decrease in brake torque (Figure 4).

Because of the steady decrease in brake torque, brake application must be increased substantially, from less than 10 psi (t = 0 s) to over 100 psi (t = 150 s) in order to maintain a constant speed. Ultimately, the required application pressure exceeds the available system pressure and it's no longer possible to maintain a constant speed.

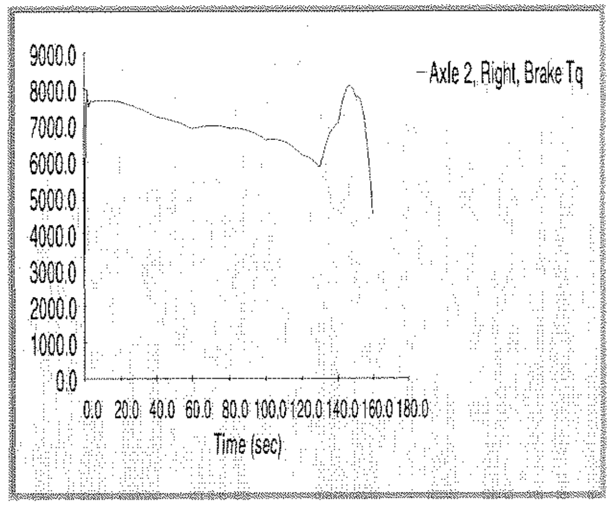


Figure 4: Brake Torque

As an exercise, what brake inputs would be required to maintain a constant speed if the initial brake stroke had been 1.75 in?

HVE-2D FAQ's

This is a new section of the newsletter dedicated to answering frequently asked questions (FAQ's) regarding the use of *HVE-2D*. These questions are compiled from our Technical Support Case Logs.

Q. When I try to run HVE-2D, I get a Security Error (1 or 2) message. What does this mean?

A. *HVE-2D* cannot communicate with the EDKEY.

1. *Is the EDKEY installed?* If not, then install your EDKEY and run *HVE-2D.*
2. *Can you print through the parallel port?* If not, then you have a configuration problem with your parallel port.
3. *Is the EDKEY driver running?* Run C:*HVE-2D*\WIN\DDINST32.EXE

Check the status of the security drivers under the System heading*. Is the Express button highlighted*? If it is then the EDKEY driver is NOT running. Click on the Express button to install with the default settings. Restart the computer, and run *HVE-2D*.

If the Express Button is grayed out, then the EDKEY driver is running, however you will need to set custom parameters to fit your system. Click on the Custom Settings button. In the Custom Settings window, click on the Parameters button. In the Parameters window, click the check-box next to the SSI_ACT setting, and set the three SSI_ACT fields to 200 200 500. Click OK in the Parameters window, and OK in the Custom window. Click close in the DDINSTALL program window. Now open your c:\autoexec.bat file in a text editor. Locate three "SET *HVE*...". Statements, and then insert the following line above these:

SET SSI_ACT=200,200,500.

Save and exit the autoexec.bat file, and reboot your computer. Run *HVE-2D* with the new EDKEY driver settings.

Q. When I open a drawing that I have created from AUTOCAD in HVE-2D, I get an error "Sorry Your Bits Won't BLT". What does this mean?

A. Like many other high performance graphics programs, *HVE-2D* may not be compatible with the highest graphics acceleration settings in Windows and your particular display hardware. To edit the current settings, close the *HVE-2D* and the *HVE*PHY programs. Click on Start-Settings-Control Panel. Double-click on Display. Click on settings. Click on the advanced button. Decrease the graphics Acceleration setting to the second setting. Click on Apply.

Windows will prompt you to restart the system. Run *HVE-2D* with the new display acceleration settings. (You may have to experiment with these settings to find the optimal setting for your system.)

Q. I am seeing "colored snow" in the borders of my AUTOCAD drawings in HVE-2D.

A. Let's make this a short "how to" for AUTOCAD drawings in *HVE-2D*.

First make sure the CAD program you are using is capable of creating either DWG or DXF files in the AUTOCAD release 12 or 13 file formats. (Release 13 DWG preferred). Setup your origin near the center of your drawing, or at your reference points in your scene measurements. Draw your site in any fashion you wish, however avoid the use of solids or fills in your drawing. When you are finished, ensure that your drawing units are in inches, if you have drawn in feet just scale the entire drawing by a factor of 12.

Next explode all of the objects in your drawing, this will convert complex objects into basic drawing entities, and remove external references. Now purge all of the objects in your drawing to remove any empty references in your drawing. It is best to purge your drawing twice, as the first purge can extract from a hierarchy an additional empty reference into your drawing. Save your drawing in the DWG or DXF format into the C:*HVE*2D\ENV\ directory. Open your environment in the *HVE-2D* Environment Editor.

Q. Do I need to install or uninstall version 1.1 before I install Version 1.2?

A. No, *HVE-2D* version 1.2 is a stand-alone, full installation that will over write the necessary files for the new version and will not affect your previous work.

Q. I see portions of my Desktop showing through the right and bottom margins of my viewer win-dow when working on my case. What's wrong?

A. *HVE-2D* can be incompatible with the Windows standard large display fonts. You will need to change to the Windows standard, small display fonts. Close *HVE-2D* and the *HVE*PHY programs. Click on Start-Settings-Control Panel. Double-click on Display. Click on Settings. (*In Windows 95*, change the Font Size to Small Fonts. *In Windows 98*, click on Advanced. Click on General settings. Change the Font Size to Small Fonts). Click on Apply. Windows will prompt you to restart the system. Run *HVE-2D* with the new display font settings.

Technical Session

This Technical Session provides an overview of the inner workings of the *HVE* Path Follower (Driver Model). The *HVE* Path Follower, a new tool in *HVE* Version 2, allows the user to define a 3-D path (using target positions). A vehicle simulation model then employs the *HVE* Path Follower to determine the driver steering inputs required to make the vehicle follow the defined path. The *HVE* Path Follower is not a statistical optimizer! It is a mathematical/physical model of a vehicle driver.

The *HVE* Path Follower has four components:

- Path Generator
- General Parameters
- Driver Descriptors
- Driver Neuro-Muscular Filter

The path generator uses up to eight 3-D positions and orientations to define the attempted path. The path is constructed from a 3-D spline curve passing through each user-specified location and tangent to the roll, pitch and yaw orientations for each location. As a spline curve, it is constructed of piecewise linear segments 12 inches in length.

The general parameters provide control over the path follower algorithm. Control parameters include driver starting time (T_{start}), driver sample interval (ΔT_{sample}), path error null distance (E_N), and initial steering wheel angle (δ_o).

The driver descriptors describe how the operator attempts to control the vehicle. These parameters include the driver preview time ($T_{preview}$), driver comfort level ($A_{\max}$), maximum steering wheel velocity ($d\delta_{\max}$), steer correction rate, (δ_c) and steer correction damping ($\dot{\delta}_c$).

The driver neuro-muscular filter represents a mathematical model of the human operator in man-machine performance. The model used in *HVE* was derived from the model described by McRuer, et al (1965) as published in the NASA Bioastronautics Databook (NASA SP-3006). The model includes an effective time delay (T_{delay}) representing the time required to read, interpret and decide upon the appropriate control motion, the lead time (T_{lead}) representing a ratio of the weight the driver attributes to the displayed velocity compared to the displayed position, and the time lag (T_{lag}) representing the amount of data smoothing the driver applies to his external stimuli.

How the Model Works

While executing, the simulation model calculates the vehicle's current position and velocity. Then, at each driver sample interval, the driver preview distance, S_p is calculated

$$S_p = T_{preview} \times U$$

where $T_{preview}$ is the driver's preview time (i.e., *How many seconds ahead is the driver looking?*) and U is the vehicle's forward velocity component. The earth-fixed coordinates of the point X_p, Y_p where the driver is looking are then calculated. From this information, we can see the point X_p, Y_p might not lie on the specified path (see Figure 1). The distance, E, from the desired path to X_p, Y_p is then calculated (note that E is normal to the desired path).

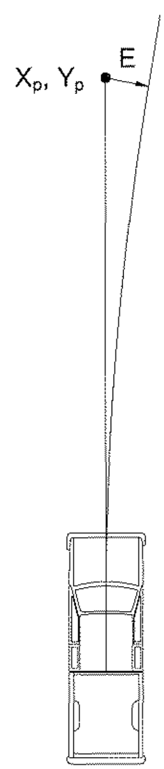


Figure 1

If E is greater than E_N, path correction is required. First, the path error rate, $\dot{E}$, is calculated:

$$\dot{E} = \frac{(E - E_p)}{\Delta t_{sample}}$$

where E_p is the path error for the previous sample interval and Δt_{sample} is the sample interval. The steer correction is then computed:

$$d\delta = \delta_c E + \dot{\delta}_c \dot{E}$$

where δ_c and $\dot{\delta}_c$ are the user-entered steer correction gain and steer correction damping, described earlier.

The rate of steer correction is:

$$d\dot{\delta} = \frac{d\delta - d\delta_p}{\Delta t_{sample}}$$

where $d\delta_p$ is the steer correction for the previous sample interval. $d\dot{\delta}$ is not allowed to exceed the user-defined maximum steer velocity, $\dot{\delta}$ max.

Finally, the required steer angle is calculated:

$$\delta = \delta_p + d\delta$$

where δ_p is the steer angle for the previous sample interval.

In practice, the allowable steer angle at the axle is limited by the steering stops. Thus, if the resulting steer angle is greater than the steering stop angle, δ is set equal to the steering stop angle.

Most drivers have a maximum comfort level related to lateral acceleration. This level, typically about 0.4 g, represents an acceleration the driver will not willingly exceed. The *HVE* Path Follower compares the current lateral acceleration with the user-assigned value, $A_{\max}$. If exceeded, the *HVE* Path Follower terminates the simulation and displays a diagnostic message.

The effect of the Neuro-muscular filter will be described in a technical paper planned for the 2000 SAE International Congress.

The *HVE* Path Follower has several interesting and useful applications. For example, while the (open loop) driver steer tables allow the user to enter an arbitrary table of steering wheel angle vs time, the *HVE* Path Follower will assess whether a *driver is capable* of performing the maneuver. An obvious application is to combine the *HVE* Path Follower with the *HVE* Tire Blow-out Model to assess the driver steering inputs required to maintain control during a rapid air loss situation

Technical Session

Before HVE, EDC's tire models used well-established techniques, such as the Fiala tire model and friction vs longitudinal slip curves for calculating forces from free-rolling and braking tires, respectively. The friction circle and slip vs rolloff methods were used for combined braking and steering maneuvers. These techniques were described in the Technical Sessions found in the previous two Newsletters. With the introduction of 3-dimensional simulation in HVE, significantly more robust models are required. For example, as a vehicle rolls about its longitudinal axis, its tires develop an inclination angle with respect to the terrain, thus producing a lateral tire force due to camber thrust. Tire moments were not considered in our earlier models, but in HVE, tire self-aligning torque may be included.

As part of the development of the EDVDS tractor-trailer simulator, EDC developed a new tire model, called the *EDC Semi-empirical Tire Model*. This new tire model is very similar to the HSRI semi-empirical tire model, developed at the University of Michigan; in fact, EDC's model was based on the Michigan model, but was extended to allow a full range of tire slip angles and tractive wheel torque (i.e., torque that accelerates the vehicle). The purpose of this Technical Session is to provide an overview of the EDC Semi-empirical Tire Model.

The EDC Semi-empirical Tire Model describes empirically what is happening at the tire-road shear interface (sometimes called the *contact patch*). The model calculates shear forces, $F_{x'}$,$F_{y'}$, and aligning torque, $M_{z'}$. The model assumes the contact patch can be divided into two regions: an adhesion region and a sliding region. The shear force generated in the adhesion region depends on the elastic properties of the tire and the shear force generated in the sliding region depends on the frictional properties at the contact patch.

The inputs to the model are:

μ_p,μ_s,S_p	Peak and slide tire-road friction and longitudinal slip at peak friction at up to 3 loads and 3 speeds (these data generate a Mu vs Slip curve at the current vertical tire load and speed)
C_α	Cornering stiffness at up to 3 loads and 3 speeds
C_γ	Camber stiffness at up to 3 loads and 3 speeds
X_p	Aligning torque stiffness
F_z, V_u	Tire current vertical load and speed (tire axis system)

The model first determines the effective levels of friction and cornering stiffness properties according to the current tire load and speed. From these properties, the fractions of the adhesion and sliding regions in the contact patch are calculated. Finally, the components of $F_{x'}$,$F_{y'}$ in the adhesion region and sliding region are calculated. The total $F_{x'}$,$F_{y'}$ tire force components are the sum of the components in the adhesion and sliding regions. The camber thrust, is added to $F_{y'}$ to account for tire inclination angle.

The above discussion provides a high-level background on the EDC Semi-empirical Tire Model. The model is described in detail, including equations, in SAE Paper No. 1999-01-0103, "Differences Between EDVDS and Phase 4." Please contact EDC Customer Service (503-644-4500) if you would like to receive a free copy of this paper.

It should be noted that 3-dimensional vehicle simulation also requires additional tire parameters. To simulate vertical tire force, F_z, on 3-dimensional terrain (F_z is crucial, since all other tire forces and moments are a function of F_z), tire radial stiffness and mass properties are required for the equations of motion. To simulate transient brake torque requires wheel spin inertia. To simulate steering from road inputs requires wheel steer inertia. All HVE tire models inherently include these parameters.

Technical Session

Our Technical Session this time is a continuation of our last Technical Session. The subject is the modeling of tire forces. In our last Session, we described various methods for calculating braking forces for a non-steered tire and steering forces for a non-braked (free-rolling) tire. Now, we're going to address the complex subject of calculating tire forces during combined braking and steering conditions.

Under the conditions of combined braking and steering, a tire produces longitudinal and lateral forces simultaneously. Experiments have shown (and most drivers' experiences have confirmed) that conditions of lateral slip reduce longitudinal force and vice versa. Several methods have been employed in various vehicle dynamics tire models for modeling this phenomenon. Two popular methods are:

- Friction Circle
- Slip vs Roll-off Function

In the ground plane, tire forces Fx and Fy are produced in the x and y directions (tire-fixed coordinate system), respectively. These forces are limited by the available friction force, μFz, where μ is the tire-ground friction and Fz is the vertical tire load. This observation leads to the following general relationship between Fx, Fy and the total available force, μFz:

$$\sqrt{F_x^{\ 2} + F_y^{\ 2}} \le \mu F_z$$

This is simply the equation of a circle which has its origin at (0,0) and radius μF_z. This relationship is called the *friction circle*. The friction circle, shown in Figure 1, is a useful way of visualizing the possible range of tire forces during combined braking and steering. Note that the total tire force is limited to the available force regardless of the magnitude of the attempted force. If the available force is exceeded, the maximum longitudinal force, F_x, is $\mu F_z\cos(\alpha)$ and F_y is $\mu F_z\sin(\alpha)$. Otherwise, the longitudinal force is the attempted F_x. The lateral force, F_y, for an unsaturated tire is that calculated using the Fiala tire model (refer to our last Technical Session); if the tire is saturated, F_y is $\mu F_z\sin(\alpha)$.

The *Slip vs Rolloff* method replaces the friction circle with measured tire parameters. This method requires testing on a flat-bed tire tester under conditions of combined braking and steering. The test results determine the loss in steering force due to the presence of braking. As shown in Figure 2, as the longitudinal tire slip increases, the percentage of F_y is reduced when compared to a free-rolling tire. The Slip vs Rolloff tire model uses the Fiala model to calculate a baseline value for F_y, then reduces F_y according to the current level of longitudinal tire slip. If the current level of longitudinal tire exceeds the slip at μ_p (again, refer to our last Technical Session), the tire forces are calculated just like the friction circle:

$F_x = \mu F_z\cos(\alpha)$ and $F_y = \mu F_z\sin(\alpha)$.

With the introduction of EDC's new 3-D models, *EDVSM* and *EDVDS*, significantly more sophisticated tire models are required. In our next Technical Session, we will discuss EDC's versions of the UMTRI Semi-empirical Tire Model and Table Look-up method.

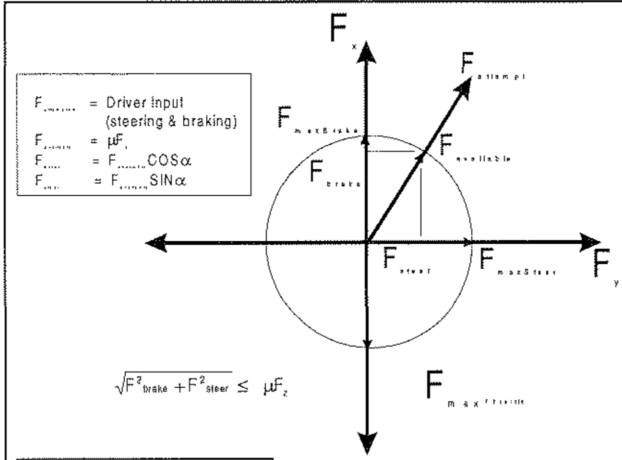


Figure 1: Friction Circle

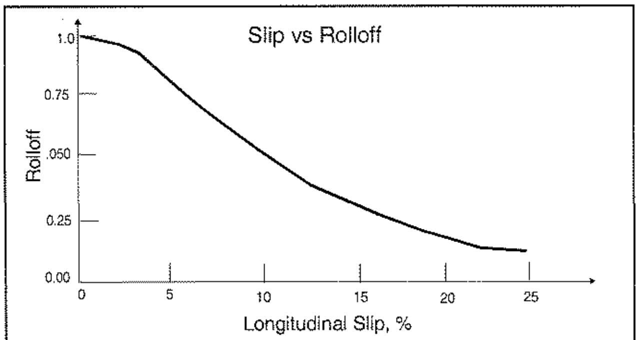


Figure 2: Slip – Rolloff Table

Technical Session

The subject of our Technical Session is *Tire Modeling*, that is, how a simulation program calculates tire forces. Consider for a moment that virtually ALL forces acting on a vehicle during a maneuver at normal highway speed come from the vehicle's tires (aerodynamic forces become significant at higher speeds). Obviously, the subject of tire modeling is quite important.

Essentially, three force-producing mechanisms act at the tire-road interface. These are forces from longitudinal slip (braking and accelerating), lateral slip (steering), and camber thrust (roll angle). We begin by addressing each of these mechanisms individually.

Longitudinal Slip

Tire longitudinal slip occurs whenever there is relative velocity between the tire contact patch and the ground. Mathematically, tire slip, S, is defined by the ratio of the linear velocity at the center of the wheel to the tangential velocity at the contact patch, or

$$S = 1 - \frac{\omega r}{V}$$

where ω is tire spin velocity, r is tire loaded radius and V is wheel center velocity. Slip is produced by braking (note that for a locked wheel, ω is 0 and Slip equals 1.0, or 100 percent), and by accelerating (note that Bullitt's '68 Mustang can create situations where $\omega r > V$, thus, we have negative slip associated with "burning rubber").

We can produce a graph of longitudinal force, F_x, vs slip. Better yet, we can divide F_x by the vertical tire load, F_z, and graph F_x/F_z vs slip. This is exactly what we've done in Figure 1. This graph is called a μ-Slip curve, and sheds a lot of insight about the behavior of tires during braking and accelerating maneuvers.

During normal driving, almost no longitudinal slip exists, therefore F_x/F_z is very low (see region *a* in **Figure 1**). Normally, there is just enough F_x to offset rolling resistance and maintain a constant speed. As the brakes are applied, longitudinal slip increases; therefore, F_x/F_z increases (see region *b*). The slope of the graph in this region is rather linear, and defines a tire property called *Longitudinal Stiffness*, C_s. As the brakes are applied more heavily, the tire slip begins to increase more rapidly. Ultimately, F_x/F_z reaches

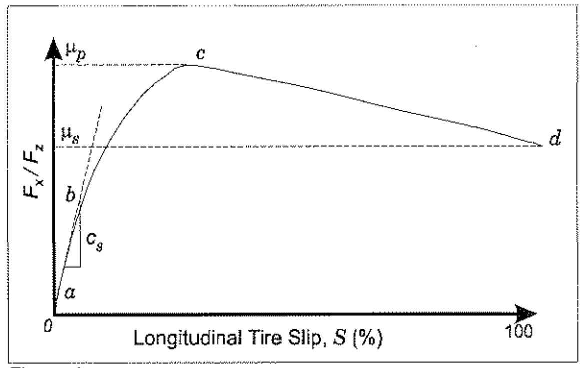


Figure 1

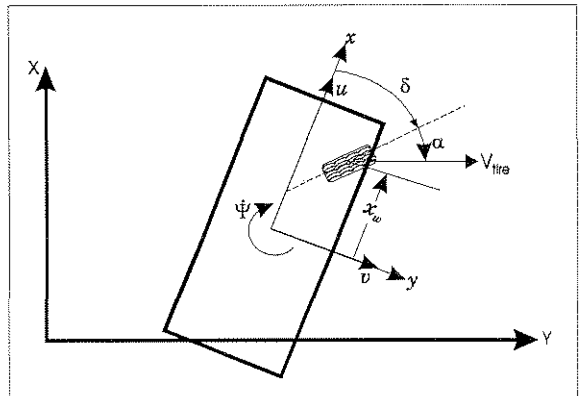


Figure 2

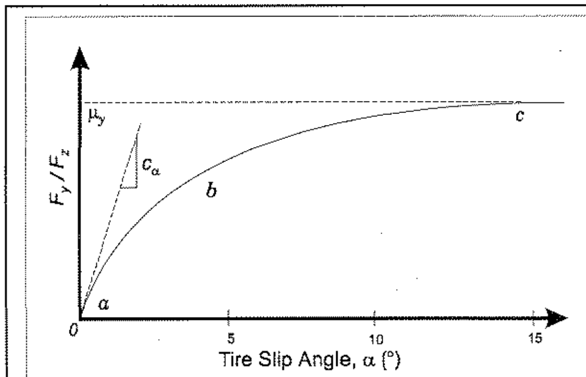


Figure 3

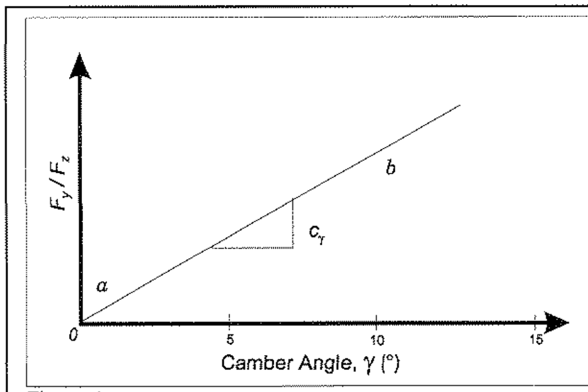


Figure 4

the maximum value available at the tire-road interface. F_x/F_z at this point (see region *c*) is the tire's peak friction coefficient, μ_p. (NOTE: By choosing to graph F_x/F_z rather than simply F_x, we are able to directly relate our tire forces to available tire friction!) Any additional braking force exceeds the available tire-road friction force, and the tire slip rapidly increases to 100 percent (see region *d*), i.e., the wheel locks up and the wheel spin velocity becomes zero. At this point, F_x/F_z is equal to the tire's slide friction coefficient, μ_s. (This is the value most people are trying to measure when they perform a locked-wheel skid test in a vehicle.) The region between *c* and *d* is dynamically unstable; a tire never operates between region *c* and *d*. Note that the basic goal of an ABS brake system is to maintain a wheel spin velocity near region *c*, regardless of how hard the brakes are applied.

Lateral Slip

Tire lateral slip occurs whenever there is an angle between the tire's longitudinal axis and its direction of motion. This angle is called the tire slip angle (see **Figure 2**). Mathematically, the slip angle, α, is defined according to the vehicle orientation and tire steer angle, or

$$\alpha = \arctan\left(\frac{v + x_w \dot{\Psi}}{u}\right) + \delta$$

where u and v are forward and lateral vehicle velocity, x_w is the longitudinal distance from the CG to the tire, Ψ-dot is the vehicle yaw velocity and δ is the tire steer angle.

Just as we did for tire longitudinal slip, we can produce a graph of tire lateral force, F_y, vs slip angle, α. Again, dividing our lateral force by F_z, we get the graph of F_y/F_z vs α shown in Figure 3.

During normal driving maneuvers, almost no lateral slip exists, therefore, F_y/F_z is very low (see region *a* in **Figure 3**). The slope of the graph in this region defines a tire property called *Cornering Stiffness,* C_α (technically, C_α is the slope at zero slip angle). As slip angle increases (as from heavy steering), F_y/F_z increases. Unlike the μ-slip curve in Figure 1, this *entire area* is non-linear (see region *b* in Figure 3; compare with Figure 1). This is one reason that tire forces are not easily modeled. As the slip angle continues to increase, F_y/F_z levels out (see region *c*). This value of F_y/F_z corresponds to the tire's lateral friction coefficient, μ_y. (Again, this point is better illustrated by graphing F_y/F_z vs α, rather than simply F_y vs α.)

Many people have heard of the Fiala tire model. The basic premise of that tire model is that F_y varies as a 3rd order polynomial function based on α, that the derivative of F_y vs α at α=0 is known (C_α, the tire's cornering stiffness), that F_y/F_z is known (μ_y the tire-road lateral friction) coefficient, and that the derivative of F_y vs α becomes zero at $F_y/F_z = \mu$ (i.e., region *c*).

Of course, there are many other tire models (Linear, Table Look-up and Magic Formula to name a few).

Camber Thrust

Tire camber thrust produces a lateral force whenever the tire's vertical axis is not normal to the road surface. The angle between the vertical axis and the surface normal is called the camber angle, γ.

We can produce a graph of F_y/F_z vs γ, as shown in **Figure 4**. During normal driving, the camber angle is quite small, so F_y from camber thrust is quite small (see region *a*). As the vehicle leans during cornering, γ increases so F_y from camber thrust increases (region *b*). Note that the slope of F_y/F_z vs γ is virtually linear up to camber angles of about 15°.

This graph illustrates that camber thrust becomes significant when a vehicle rolls. Thus, when simulating a rollover event, modeling the camber thrust is critical. In fact, camber thrust is often the only lateral force acting at the tire-road interface. Another example of camber thrust is the cornering force acting on a motorcycle (virtually 100 percent of the cornering force is produced by camber thrust due to the motorcycle's leaning).

In the above developments, we described how individual tire forces are produced from braking, steering and roll. What about when these forces act *simultaneously*? Stay tuned! The modeling of tire forces produced during combined braking and steering maneuvers is the subject of the Technical Session in our next EDC Newsletter.

Technical Session

Our Technical Session in this issue describes *EDSMAC4*, a significant reformulation of *EDSMAC*. Three portions of the code were revised: the main control algorithm, the collision algorithm and the trajectory algorithm. Our Technical Session focuses on the reformulation of the collision algorithm. The following assumes the reader is generally knowledgeable about the techniques used by the original *EDSMAC* collision algorithm. For a detailed description, the reader is referred to SAE Paper No. 880069 ("An Overview of the Way *EDSMAC* Computes Delta-V").

Reformulation of the collision algorithm involved the following tasks:

- Complete Reorganization
- A, B Stiffness Model
- Separate Stiffness for Each Side
- Support for Barriers

Re-organization of the code was a major undertaking. Although the original FORTRAN code had earlier been ported to the C language, it still contained lots of 'spaghetti'. Reorganization involved removing all GOTO statements and using 'while loops' to work through the table of Rho vectors and to iterate the force balance on each set of opposing Rho vectors. The resulting code is elegant, easy to follow, and most of all, extendable! The first extension was to revise the code to accept an arbitrary set of any two vehicles from a list. The new code is not dependent on which vehicle in the list is causing damage to the current vehicle; it simply knows the current vehicle has damaged Rho vectors (by the way, the new *EDSMAC4* control routine keeps track of which vehicles are in contact with each other).

The original K_v stiffness model was extended to use an A, B stiffness model (these are the same A and B values used in *EDCRASH*). The purpose for this change was to allow a more realistic modeling of actual vehicle structural behavior. As shown in Figure 1, the structural stiffness of a vehicle typically requires a threshold force to be applied before deformation begins. This force corresponds to the A stiffness. Note the K_v model essentially assumes this value is 0.0. As a result, the K_v model predicts an overly soft vehicle for minor to moderate amounts of crush (refer to Figure 1). This issue was addressed in our *EDSMAC* Validation Study, "Further Validation of

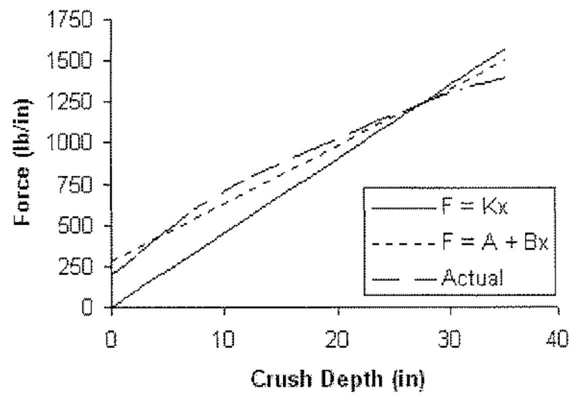


Figure 1: Experimental and modeled stiffness relationships.

EDSMAC Using the RICSAC Staged Collisions," SAE Paper No. 900102. While this issue does not greatly affect the overall vehicle trajectory, it can greatly affect the shape of the acceleration vs. time curve by increasing the peak acceleration and shortening the impact duration. Thus, this new algorithm may be important when a collision pulse is used in an occupant simulation.

Implementing the A, B stiffness model required changing the force vs. deflection model for each Rho vector from $F = K_v x$ to $F = A + Bx$. The new model also required an initial deflection constant for near zero deflection (typically, 0.5 inches is assumed). Also, when the deflection is below 0.5 inches, the Rho vector adjustment, delRho, is reduced by a factor of 20. This reduction is required because of a potentially large A stiffness value. The more complex A, B stiffness model also requires many more iteration attempts while balancing force vectors. Therefore, the number of allowable iterations per Rho vector has been increased from 200 to 3200. During normal penetration, this large value is irrelevant because the force normally equalizes within 10 to 20 attempts. However, when the deflection is less than 0.5 inches, the number of iterations can increase to over 1000 owing to the reduced value of delRho.

The original *EDSMAC* algorithm assumes the vehicle exterior has a homogeneous stiffness about its perimeter. *EDSMAC4* uses different A, B stiffness pairs for the front, sides and back. Implementing this change simply required that the radial sweep angle of each Rho vector be compared with the vehicle's x,y corner coordinates to determine if the Rho vector passes through the front, side or back. The appropriate A, B stiffness pair is then selected for the Rho vector's force calculations.

The original *EDSMAC* algorithm allows 100 Rho vectors to be included in the table of damaged vectors. Because *EDSMAC4* allows a vehicle to be struck on all four sides by any number of vehicles at the same time, the number of allowable Rho vectors in the damage table has been increased to 360. This allows every Rho vector to be included in the damage table.

The implementation of an A, B stiffness model also allows for the direct support of barrier collisions (a barrier is defined as a nondeformable object that does not absorb energy). If the collision routine is told that one of the vehicles is a barrier, the A and B stiffnesses are assigned an extremely high value (e.g., A = 1000 lb/in, B = 1000 lb/in^2), and delRho is further reduced by a factor of 10.

These changes have resulted in a stable damage algorithm that more accurately duplicates real vehicle behavior. Validation has shown what was expected: for most collisions, the peak acceleration is slightly greater and the duration of impact is slightly less. This behavior is more noticeable at lower speeds. A full validation study will be published.

Technical Session

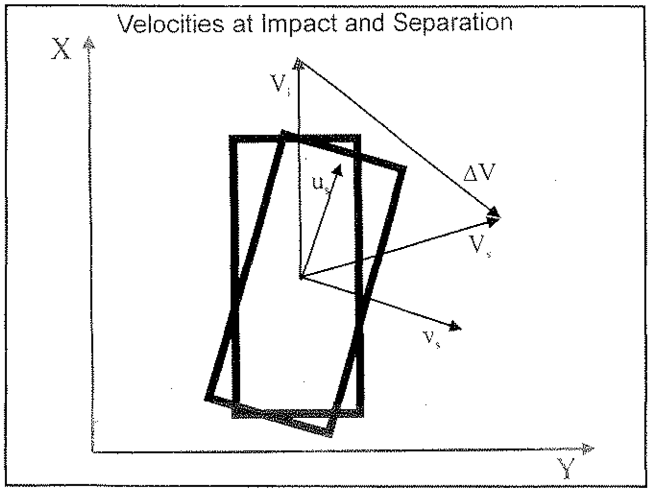
Why does the ΔV reported in EDSMAC's Damage Summary often differ from the difference between vector velocities at impact and separation? To answer this question requires a bit of background.

During a collision (defined in EDSMAC as the interval during which the total acceleration exceeds 1G), EDSMAC accumulates and stores the current vehicle-fixed acceleration, both magnitude and direction. From this information, EDSMAC calculates and stores the ΔV (area under the acceleration vs. time curve) for each clock direction (note the clock direction comes from the acceleration). Finally, EDSMAC stores the peak acceleration for each clock direction. Again, the preceding information is accumulated and stored during the entire collision phase.

At the end of the simulation , EDSMAC uses the stored information to assign a CDC, PDOF and delta-V to each vehicle damage region. If a vehicle has exactly one damage region, all delta-Vs within a one hour clock direction are summed and included as part of the reported delta-V. Those delta-Vs that lie outside this range are ignored. For this reason, the reported Delta-V may be slightly less than the difference between vector velocities at impact and separation.

EDSMAC also allows a vehicle to have more than one damage region. For example, during an intersection collision, a vehicle quite often has secondary damage at the (rear) quarterpanel resulting from rapid rotation and subsequent slapping together of the vehicles' rear ends. In this case, EDSMAC determines a CDC, PDOF and ΔV for each damage region. The direction for each acceleration peak is compared to the location of damage to determine the damage region to which PDOF and ΔV are assigned. If the direction of an acceleration peak is more than 60 degrees from the angle to the midpoint of the damage region, the CDC, PDOF and ΔV are assumed to belong to a different damage region.

Velocities at Impact and Separation



The above logic involving multiple damage regions is very complex, and sometimes fails to match a CDC, PDOF and ΔV to a damage region. In that case, there may be a significant difference between the ΔV reported in the Damage Summary and the difference in vector velocities between impact and separation, especially if the unmatched ΔV is large.

Users often perform a simple calculation using the vector difference in impact and separation velocities displayed in EDSMAC's Accident History report. This normally provides a reasonable estimate for the ΔV. However, if the vehicle undergoes significant rotation between impact and separation, this rotation must be accounted for. The following example illustrates the effect of rotation on the calculation:

	X	Y	ψ	V-total	u	v
	ft	ft	deg	ft/sec	ft/ sec	ft/ sec
Impact	0	0	0	45.4	45.4	0.0
Separation	1.4	0.7	11	21.5	12.7	17.3

Ignoring the rotation, the ΔV is calculated as follows:

$$\Delta V = V_s - V_i = \sqrt{(u_s - u_i)^2 + (v_s - v_i)^2}$$
$$= \sqrt{(12.7-45.4)^2 + (17.3-0)^2}$$
$$= 37.0 \text{ ft/sec}$$

When rotation during impact is considered, the actual ΔV is calculated as follows:

ν = ψ + β = ψ + ATAN2(v/u)
= 0 + ATAN2(0/45.4) =0° (at impact)
= 11 + ATAN2(17.3/12.7) = 64.7° (at separation)

Calculating V_x and V_y relative to the earth-fixed coordinate system,

V_x = $V_t$cos (ν)
= 45.4cos(0) = 45.4 ft/sec (at impact)
= 21.5cos(64.7) = 9.2 ft/sec (at separation)

V_y =$V_t$sin(ν)
= 45.4sin(0) = 0.0 ft/sec (at Impact)
=21.5sin(64.7) = 19.4 ft/sec (at separation)

$$\Delta V = V_s - V_i = \sqrt{(V_{x,s} - V_{x,i})^2 + (V_{y,s} - V_{y,i})^2}$$
$$= \sqrt{(9.2-45.4)^2 + (19.4-0)^2}$$
$$= 41.1 \text{ ft/sec}$$

As you can see, the ΔV is different when rotation is considered.

For normal amounts of rotation during impact, the difference is not great. However, the difference can become sizable when rotations are large. Note that the ΔV reported in the Damage Summary Report addresses this problem by reporting a ΔV for each CDC having a different clock direction.

Technical Session

Our Technical Session deals with a relatively common occurrence for those studying collisions: How does one use EDSMAC to simulate a missing or damaged wheel? First, it must be understood that it is difficult to simulate a missing wheel. To understand why, it is helpful to understand how EDSMAC simulates a vehicle's motion.

Remember that the vehicle's motion is determined by the collision forces and tire forces. After the collision is over, the only forces on the vehicle are applied at the tires. EDSMAC's "Tire Model" simulates these forces by assuming "normal" behavior exists between the tire and the road. "Normal" means that steering, braking, friction, tire cornering stiffness and vertical tire load are the parameters that affect tire force (in HVE-3D, surface slope also affects tire force). To simulate a missing tire, we first need to determine how the missing tire affects forces on the vehicle.

One possibility is that the missing tire might simply result in no force applied at that wheel location. Using EDVAP, we have a problem because EDSMAC does not allow us to assign friction values for individual wheels. We can handle this situation in HVE by using the Vehicle Editor to assign a very small friction and cornering stiffness value for that tire (using the In-use Factor is a convenient way to do this). When EDSMAC executes, the tire model will calculate negligible force at this tire. Remember, this is still an approximation.

The other possibility is that the tire is missing or damaged, and the rim, brake disk or lower suspension arm is dragging on the ground. This is a problem because the force produced at this wheel location may be different from the forces produced by a tire (this is especially true for gouging by suspension components). The best approach is to drop the cornering stiffness to a very small value, then use a friction value that approximates what you believe occurred between the metal and the road. Because we are talking about one wheel out of a total of four, it is not likely the error produced by the above approximation will totally invalidate your run, but you should perform a sensitivity analysis (varying the values of friction and cornering stiffness) to confirm the effect of your assumption.

Note that we did not attempt to solve the problem by supplying a 0 force in the Driver Wheel Force table. Changing the Throttle or Braking table only affects the attempted tire force in the longitudinal direction. Even with a force entry at a particular wheel, the tire will still produce a potentially large force when it has a large sideslip angle!!!

HVE Technical Tips

EDC recently introduced EDHIS, a 3-D human impact simulator used for studying occupant and pedestrian collision. Because the subject of human simulation is new to many, we thought an overview of the EDHIS model would be helpful.

EDHIS is an 3-D simulation of the human body. It models the human as three masses (Head, Torso and Legs) connected by two joints (Neck and Hip). The human contact surfaces are modeled by 3-D ellipsoids attached to each mass segment (up to 25 ellipsoids may be attached). Note that EDHIS uses only a portion of the HVE Human Model, which has 15 masses and 14 joints.

EDHIS also includes a 3-D model of the vehicle. The vehicle includes a number user-defined 3-D planes, called contact surfaces.

Like all simulations, EDHIS solves the equations of motion at discrete timesteps. The basic procedure involves sensing the interaction between human ellipsoids and vehicle contact surfaces at each timestep. For each human ellipsoid interacting with a vehicle contact surface, the interaction force is computed. Given the sum of all these forces acting on the human, the human's acceleration is calculated. Numerical integration is then used to calculate the change in velocity and position for the start of the next timestep. This process is repeated until the run terminates, resulting in a time history for position, velocity and acceleration for each mass segment, as well as the interaction force.

The process of using EDHIS involves the following basic steps:

Step 1—Use the HVE Human Editor to select a 3-D human (according to sex, age, weight %-tile and height %-tile). The human may be edited.

Step 2—Use the HVE Vehicle Editor to select a 3-D vehicle (according to type, make, model, year and body style). Add the necessary vehicle contact surfaces (e.g., seat bottom, seat back, floor board, dashboard) and restraint systems.

Step 3—Use the HVE Event Editor to a) supply an initial position and velocity for the vehicle, b) supply an initial position and velocity for the human, c) supply a collision pulse, d) supply a restraint system, and e) select or deselect any contact.

Step 4—Use the HVE Event Editor to execute the event. You can visualize the motion in HVE's 3-D viewers while the simulation is running. In addition, you can display the current simulation results in numeric Key Results window.

The primary use of EDHIS is to study an occupant's response to the force of impact. EDHIS predicts and shows the resulting motion. This is useful for showing how and where an occupant strikes the vehicle interior during a crash. You can also execute the event with and without seat belts. These studies are useful to help answer such questions as *Who was driving?* and *Were seat belts in use?*

Another important use for EDHIS is the study of a pedestrian's response to impact. This is useful for estimating the speed of a striking vehicle and the effect of vehicle exterior design. Both of these issues may be examined by observing how and where the pedestrian contacts the vehicle exterior.

Limitations – Although EDHIS is a very powerful simulation tool, the simplified, 3-mass model used by EDHIS restricts its use. For example, EDHIS should not be used to predict occupant motion during a rollover, or for estimating vehicle speed based on pedestrian throw distance (EDHIS does not model the human's interaction with the ground). 90 degree pitch angles for any segment also pose a problem due to a computational singularity. As is always the case, it is important to understand the model's assumptions and limitations in order to use it correctly. EDC has planned two technical papers, one to be delivered at the AIRIL conference (Oct. 20-23, 1997) in Brisbane, Australia, and the other at the SAE Congress (February 23-26, 1998 SAE paper #980221) in Detroit. Please let us know if you would like preliminary copied of these reports.

Technical Session

One of the most common misconceptions about vehicle damage energy is the (thoroughly incorrect) belief that damage energy increases as impact speed increases. It can be shown through derivation from first principles that this is not true. However, the best illustration comes from a simple example. First, let's assume a collinear collision wherein the bullet vehicle traveling at 20 mph strikes the rear of a target vehicle traveling 10 mph. Given:

$$W_1 = 3500 \text{ lb.}, \quad W_2 = 4000 \text{ lb}$$
$$V_1 = 20 \text{ mph}, \quad V_2 = 10 \text{ mph}$$

To determine the post-impact speed V'_{12}, use linear momentum and assume plastic impact:

$$\begin{aligned} V'_{12} &= (m_1V_1 + m_2V_2) / (m_1 - m_2) \\ &= 14.7 \text{ mph} \end{aligned}$$

Note that the ΔVs are:

$$\Delta V_1 = 14.7 - 20 = -5.3 \text{ mph}$$
$$\Delta V_2 = 14.7 - 10 = 4.7 \text{ mph}$$

Now, set up an energy balance to determine the loss in kinetic energy:

$$\begin{aligned} E_{crush} &= \tfrac{1}{2}[(m_1V_1^2 + m_2V_2^2) - (m_1 + m_2)V'^2_{12}] \\ &= 74{,}820 \text{ in-lb} \\ &= 6{,}235 \text{ ft-lb} \end{aligned}$$

Now, let's increase the speeds, $V_1 = 80$ mph and $V_2 = 70$ mph:

$$\begin{aligned} V'_{12} &= (m_1V_1 + m_2V_2) / (m_1 + m_2) \\ &= 74.7 \text{ mph} \end{aligned}$$

Again, note the delta-Vs are

$$\Delta V_1 = 74.7 - 80 = -5.3 \text{ mph}$$
$$\Delta V_2 = 74.7 - 70 = 4.7 \text{ mph}$$

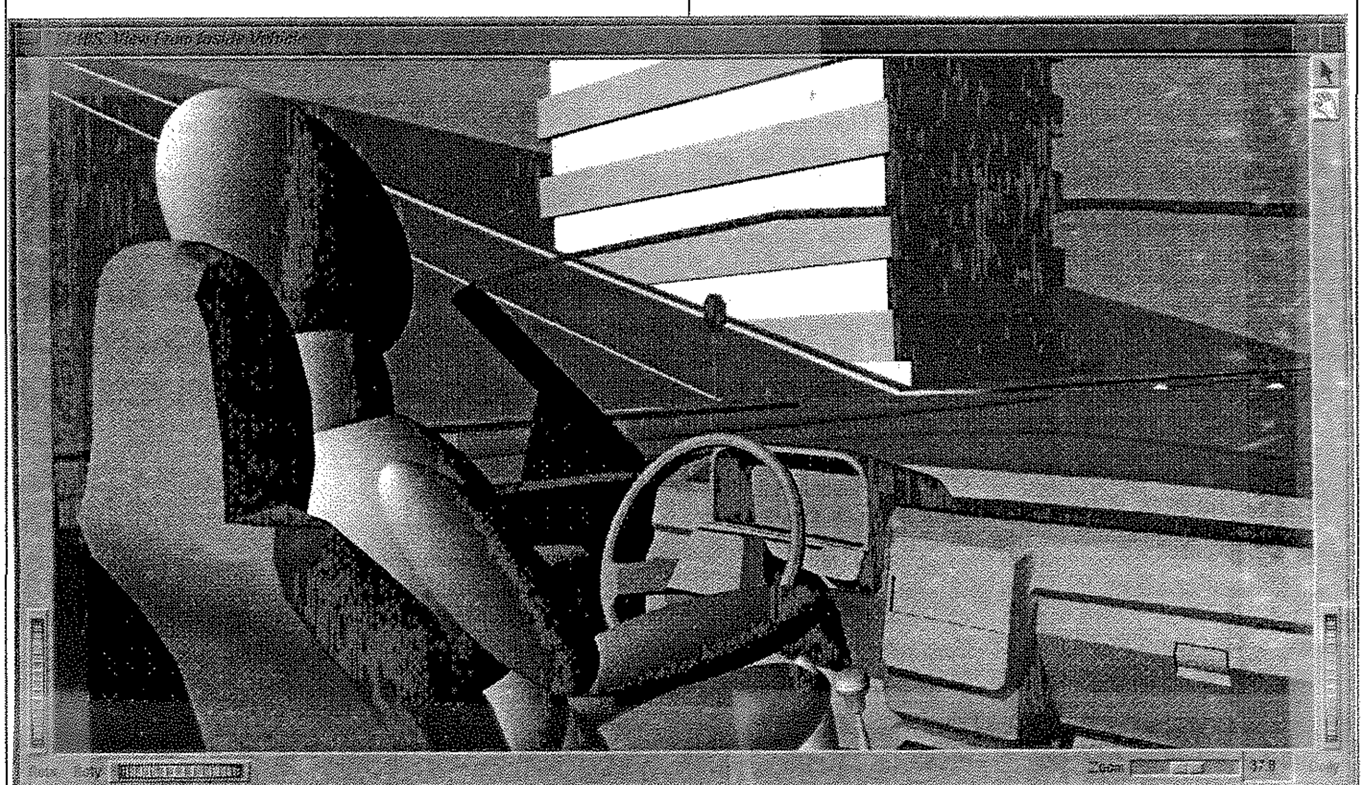
Note the delta-Vs are the same!

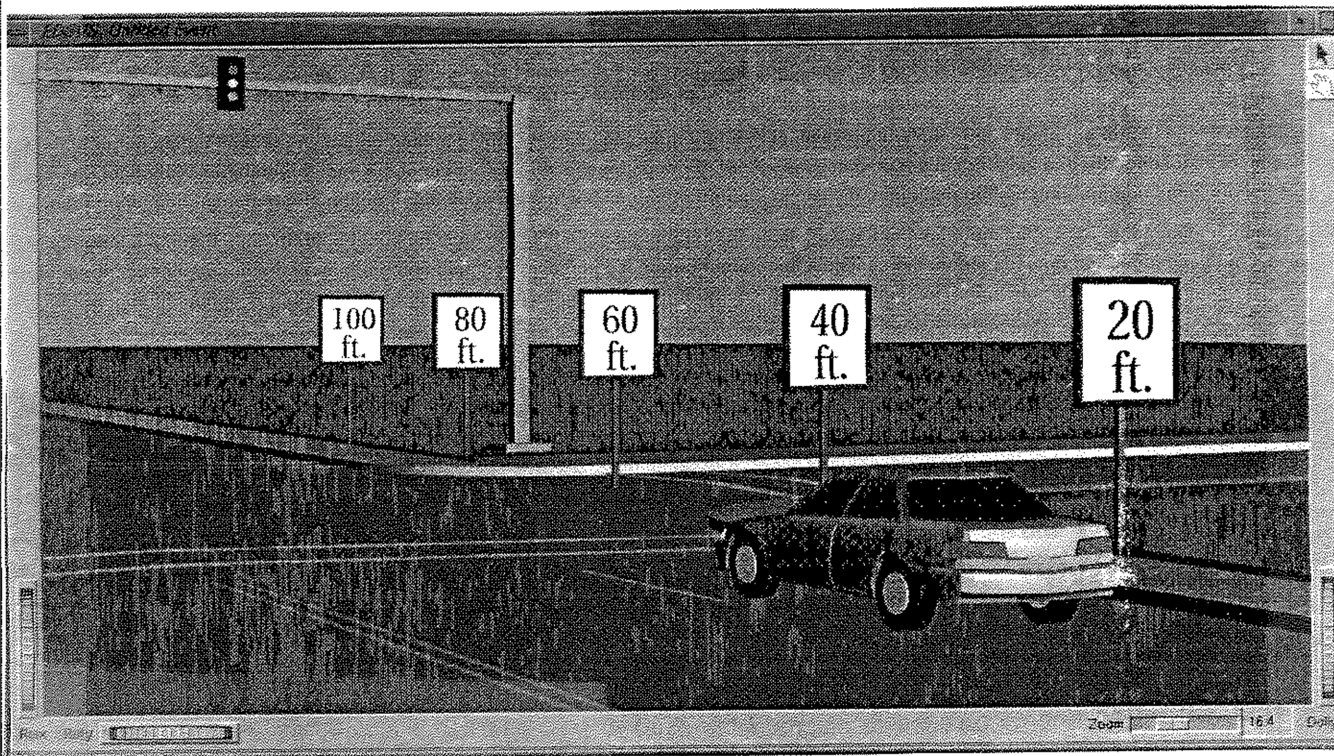
Now for the crush energy, the energy balance yields

$$\begin{aligned} E_{crush} &= \tfrac{1}{2}[(m_1V_1^2 + m_2V_2^2) - (m_1 + m_2)V'^2_{12}] \\ &= 74{,}820 \text{ in-lb.} \\ &= 6{,}235 \text{ ft-lb.} \end{aligned}$$

Clearly, the damage energy is independent from the impact speeds. If you're scratching your head, thinking "How can that be? There is so much more kinetic energy in the second scenario." The key is to note that *system* kinetic energy is conserved, and in addition to having much greater kinetic energy at impact, the second scenario also has much greater kinetic energy at separation. It is a fact that damage energy is a function of closing velocity, not either vehicle's absolute velocity!

Thanks to Wes Grimes, Collision Engineering Associates, for this simple, yet convincing, illustration.





HVE Technical Tips

Viewing From Inside a Vehicle

The 3-D geometry models used for HVE vehicles normally include tinted windows. This is great, because you normally don't want to visualize the wheels and brake lights inside the vehicle (they are visible because the interior is not completely rendered; only the seats, dash and door panels are included).

If the visibility from the driver's perspective is important, you'll want to attach the camera to the driver's location inside the vehicle. However, if the windows are tinted, you won't be able to see through the window. This problem is easily solved by editing the 3-D geometry file. To do this, perform the following steps:

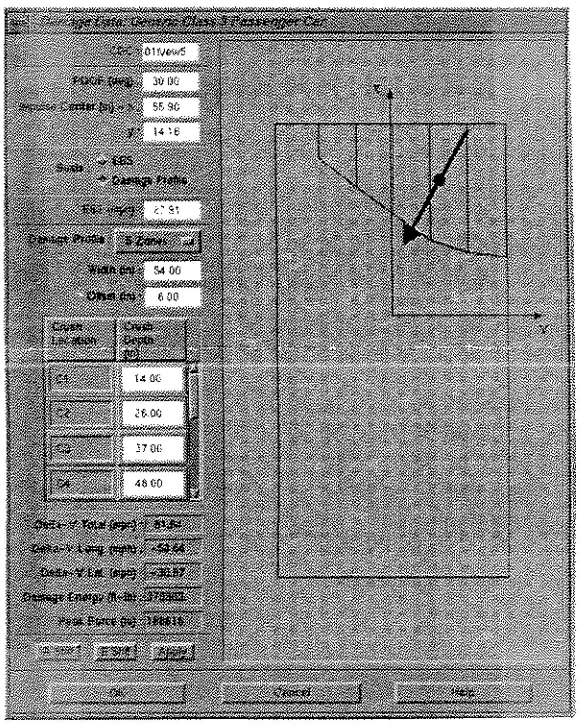
1. In the Vehicle Editor, choose the desired vehicle. Click on the CG and choose Geometry File, Open, from the option list. The listbox will highlight the current 3-D Geometry filename.
2. In a separate shell, use your favorite text editor to edit this file. You'll find it in the /hve/supportFiles/images/vehicles directory. Search for the string "DEF GLASS Material". The default transparence is -.20. Change it to the desired value (increasing the value makes the windows more translucent). To make it transparent, change the value to 1.0. Save the file.
3. Choose OK in the 3-D Geometry file listbox to load and display the modified geometry file. Alternatively, you can save the file as a new file (see Step 2, above) and choose the modified file in the listbox. This latter approach has the advantage that it does not affect you existing files.

Validating You Scene

An important question you might need to answer is, "how do you know it's 100 ft., for instance, from point A to point B". An easy and convincing way to answer that question is to place road signs at user-defined intervals in your environment, then actually drive a vehicle through the environment and note the time required. You can name this your Validation Event and show it on request.

Technical Session

This Technical Session discusses a question we are sometimes asked: "Using EDCRASH, I've run a vehicle into a fixed-barrier. Why do my EDCRASH Damage-based results differ from a simple energy balance?" In other words, why isn't the damage energy equal to $1/2mV^2$? The answer is that the above relation holds only for a central collision, that is , one where the PDOF acts directly through the CG. If the collision is non-central, the vehicle will rotate. As a result, the delta-V will always be lower than predicted by the energy balance because some of the damage energy is spent causing a change in angular velocity.



Let's use an example. Suppose a 2700 lb. vehicle strikes a fixed barier as shown above. Note the reported damage energy is about 371,000 ft-lb. and the total delta-V is 61.9 mph.

According to the energy balance, the damage energy would be

$$\begin{aligned} E &= 1/2mV^2 \\ &= 1/2(2700/386.4)\ (61.9\text{x}17.6)^2 \\ &= 4{,}147{,}000 \text{ in-lb} \\ &= 346{,}000 \text{ ft-lb} \end{aligned}$$

The reported damage energy was higher because it included the rotational kinetic energy. Let's illustrate the point by reducing the moment arm of the PDOF, such a change would affect the damage energy because of the Energy Magnification Factor (perhaps the topic of another Technical Session). Instead, let's simply increase the damage offset until the moment arm is zero. By trial and error, the value that results in a zero moment arm is 24.1 inches. The resulting delta-V is 64.1 mph and the damage energy is still 371,000 ft-lb. Now, let's compare the results with an energy balance equation.

$$\begin{aligned} E &= 1/2(2700/386.4)\ (64.1\text{x}17.6)^2 \\ &= 4{,}447{,}000 \text{ in-lb} \\ &= 371{,}000 \text{ ft-lb} \end{aligned}$$

As we can see, the energy *does* balance.

Another related question we get a lot has to do with the loss of kinetic energy during a two-car collision. That's the topic of our next Newsletter's Technical Session. Stay tuned!

HVE Technical Tips

Driving in 3-D

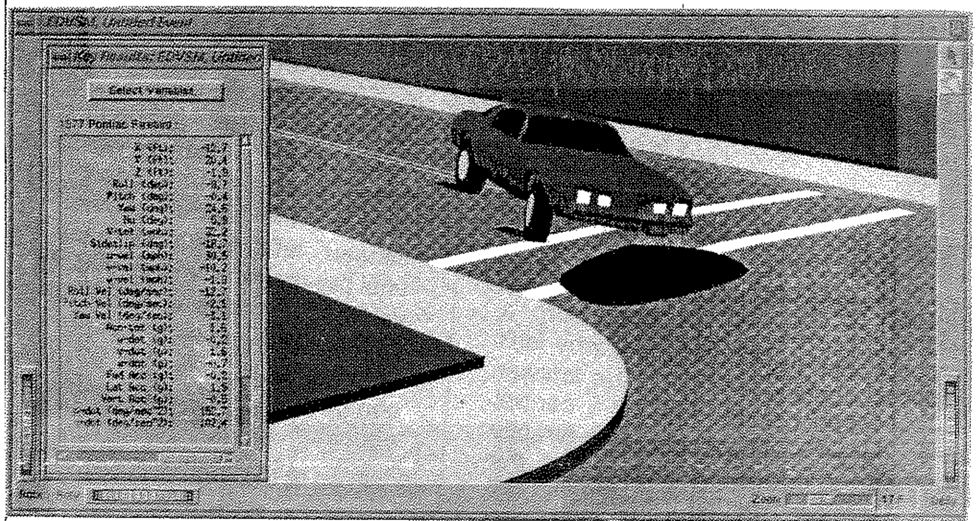
This Technical Session is for HVE users driving in a 3-D world. The surfaces on which the vehicle is driving are actually the same surfaces used to view the 3-D environment. You can use HVE's 3-D editor to create these surfaces, or you can use your favorite 3-D modeling program.

In either case, some interesting situations may arise:

> Suppose you are simulating a vehicle on an overpass. What prevents the simulator from using the surface of the road beneath the overpass? How about the under side of the overpass?
>
> Alternatively, suppose the vehicle is on an underpass. What prevents the simulator from using the overpass?
>
> Suppose you created an oil slick and placed it on the road surface. What prevents the simulator from using the road, instead of the oil slick?

To solve these problems, HVE's 3-D Editor allows you to define three surface types for environments: *Road, Friction Zone* and *Other*. In addition, the following important rules are used by the simulation:

1. The first surface found beneath the tire is used. This rule saves time, for it means the simulator does not need to investigate every polygon, then decide which one to use; it simply uses the first polygon found beneath the tire. If two polygons exist, one just above the other, the



first one found in the database will be used, regardless of its elevation.

2. Friction Zones are used first. This is an important rule. It implies a hierarchy.

Thus, to ensure the overpass or oil slick is used by the simulator, assign it's type as Friction Zone. Note that if you use your favorite modeling program to create your environment, you'll need to edit the environment in HVE's 3-D Editor and assign friction zones where required.

Technical Session

This is the third in a series of Technical Sessions about Driver Tables in EDSMAC, EDSVS and EDVTS. In the first of the series, we started with an example using the Steer Table. In the second, we provided on example using the Wheel Force Table. In this Technical Session, we provide an example of combined braking and steering.

Our goal in this example is to start from an initial speed of 35 mph, enter a left turn lane and stop at the stop line. The total distance traveled is 225 feet, and the lanes are 12 feet wide.

The first step is to determine the require level of deceleration.

$$f = \frac{v^2}{30S} = \frac{35^2}{30 * 225} = 0.18g$$

Assuming a coefficient of friction of 0.07, you would enter:

Time sec	Wheel force			
	r/f	l/f	r/r	l/r
0.00	0.26	0.26	0.26	0.26

Notice we entered 0.26, instead of 0.18. As we mentioned in out last Technical Session, you must divide by the coefficient of friction before entering the values:

$$value = \frac{f}{\mu} = \frac{0.18}{0.70} = 0.26$$

The next step is to determine the level of steering. While there are several approaches, one approach is by far the best: Simply let the simulator guide you! Proceed as follows.

First, think about how you would perform this task if you were actually driving the car. You might estimate that you would turn the steering wheel 45 degrees to the left. Then you would turn back 45 degrees to the right. Finally, you would steer back to zero to straighten out. Notice this requires three steering maneuvers (left, right, straight). Assuming your vehicle's steering gear ratio was 20:1, 45 degrees at the steering wheel equals 2.25 degrees (45/20) at the tires. Your table should look similar to this:

Time (sec)	Steer Angle (deg)
0.00	0.00
0.50	-2.25
1.23	-2.25
1.73	2.25
2.70	2.25
3.25	0.00

The above table causes the vehicle to travel the desired path. Notice we used ±2.25 degrees for every steering input. The only thing we varied were the times. Using RTG, let the simulator (we used EDSMAC) tell you when to begin the steering maneuvers. A little trial and error is required. We were able to arrive at the above table in less than 10 minutes.

Again, we stress two things: First, your table should be simple, and should reflect real life (notice our table has only three maneuvers). Second, when it comes to steering tables (especially when combined with braking), use the current simulation time displayed in RTG to tell you when to change the steer angle.

Technical Session

This is a continuation of our last Technical Session about Driver Tables in EDSMAC, EDSVS and EDVTS. When properly understood, these tables are easy and convenient. If not properly understood, you will probably become frustrated. As we said last time, the magic words are *Linear Interpolation*. Again, we stress the importance of understanding these tables.

In our last Technical Session, we gave an example using the Steer Table. (We even gave you a Pop Quiz!) In this Technical Session, we'll take a look at the Wheel Force Table. And, we'll make it interesting by trying to accelerate a vehicle at a desired rate of acceleration.

Before turning to our example, let's review the basic concept of linear interpolation. Briefly, when you enter a table of wheel forces at consecutive time intervals, the simulation interpolates the table values to determine the wheel forces at times between the table entries. Refer to the February, 1995 Technical Session for the details.

In our example, we wish to accelerate a vehicle at 0.2 g. Assume the coefficient of friction is 0.70. In the table, you would enter:

Time		*Wheel Force*		
(sec)	*r/f*	*l/f*	*r/r*	*l/r*
0.000	0.29	0.29	0.29	0.29

Notice that we entered 0.29, instead of 0.20, because we must first divide the desired acceleration rate by the friction coefficient:

$$\text{Value} = a/\mu = 0.20/0.7$$
$$= 0.286$$

Notice also that our table inherently assumes a 4-wheel drive table (we entered a force at each wheel). If you have a 2-wheel drive vehicle you would need to double the values at the drive wheels and enter zero for the non-drive wheels. Assuming a front drive vehicle, you would enter:

$$\text{Value} = 2(a/\mu) = 2(0.20/0.70)$$
$$=0.57$$

Your table should look like this:

Time		*Wheel Force*		
(sec)	*r/f*	*l/f*	*r/r*	*l/r*
0.000	0.57	0.57	0.0	0.0

EDSMAC users sometimes ask why, after going through the above process, they do not get the expected rate of acceleration – in fact, the simulated vehicle's rate of acceleration changes during the run! Here's the reason: During EDSMAC's Input Session, Question 14 allows the user to enter a *velocity dependence factor* which decrements the current friction level as a function of the current speed. As a result, the friction keeps changing during the run. The default velocity dependence factor is 0.0003. To achieve the desired acceleration rate, change this factor to 0.0 before executing your simulation.

Technical Session

Our Technical Session deals with the Driver Tables in EDVAP simulations EDSVS, EDVTS and EDSMAC. Understanding these tables will probably provide you with the single greatest time savings while using EDVAP simulations. Obviously, then, this is an important Technical Session.

The key to success lies in two simple words: *Linear Interpolation*! The Wheel Force and Steer Tables both use linear interpolation to determine the current level of driver inputs, according to the following formula:

$$x = \left(\frac{x_2 - x_1}{t_2 - t_1}\right)(t - t_1)$$

where: x is the level of the interpolated variable (wheel force or steer angle) at time, t, x_1 and x_2 are the user-entered table values for steer force or steering; and t_1 and t_2 are the user-entered times corresponding to x_1 and x_2, respectively (see Figure 1).

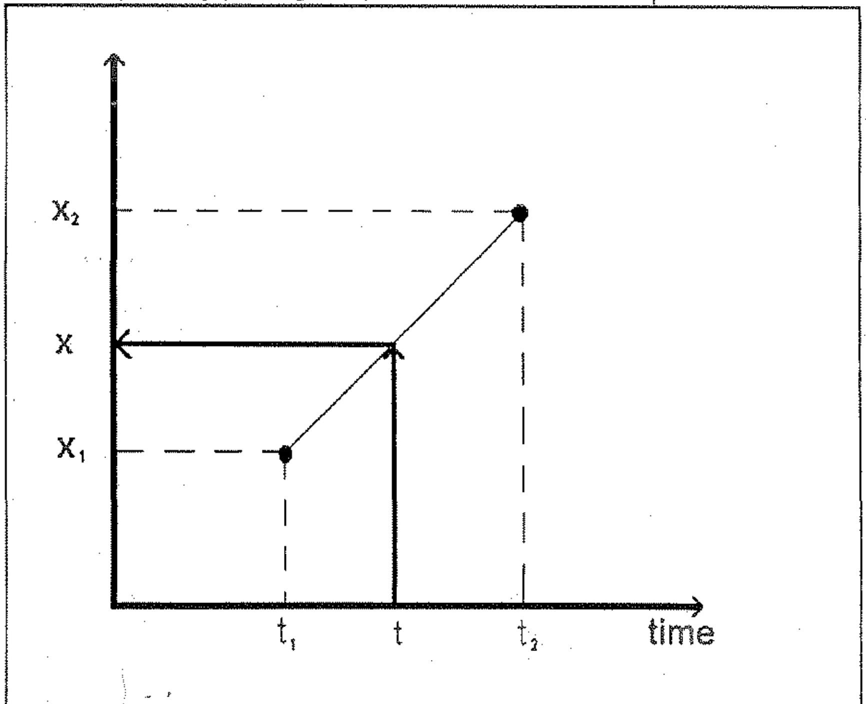


Figure 1

The process is best illustrated by example. Assume you want 5 degrees of steering to begin at 1 second. Assuming it takes the driver ½ second to produce this steer angle (remember: because of a steering gear ratio of approximately 20:1, the driver must steer about 100 degrees at the steering wheel). You should enter the following steering table:

t (sec)	steer angle (deg)
0.000	0.0
1.000	0.0
1.500	5.0

This will result in no steering until 1 second into the run; the steering will then build up to 5 degrees at 1.5 seconds.

Assume you entered any of the following tables:

t (sec)	steer angle (deg)
0.000	0.0
1.000	5.0

or

t (sec)	steer angle (deg)
0.000	0.0
1.500	5.0

or

t (sec)	steer angle (deg)
1.000	5.0

In all these tables, steering would begin immediately and the vehicle would begin its turn before it was supposed to!

The same is true with the wheel force table. Unless you enter the table correctly, your simulated vehicle will begin accelerating or braking before you want it to.

Once you become familiar with the use of interpolation for the driver tables, you will realize how well-suited the method is for driver input tables: The interpolation allows you to easily model the build-up time for steering and braking. In fact, you should consciously think about the amount of time it takes for brake torque and steering build-up, and let the table's linear interpolation assist you!

Another suggestion regarding your driver tables: these tables, especially the steering table, should have as few lines of data as possible. Long tables indicate unnecessary complexity. If you think about a typical steering maneuver (e.g., negotiating a curve), you tend to steer only twice: once to enter the curve and then again when you exit. Often, your tables should only be 2 or 3 lines in length.

Technical Session

Our Technical Session this month deals with loading EDCRASH results directly into EDSMAC. As you may know, the original purpose of CRASH was to serve as a preprocessor for SMAC in order to improve the initial conditions and, therefore reduce a number SMAC runs.

First, it is important to know that the structure of an EDCRASH output file is exactly the same as an EDSMAC input file. Also, every time you run EDCRASH, the output file it produces contains all the necessary data to run EDSMAC.

To load an EDCRASH output file into EDSMAC, you can simply choose Option 5 (EDSMAC) from the EDCRASH Program Menu. You can also directly execute EDSMAC and select Option 2 (Rerun) from the Program Menu. Then, enter the name of the EDCRASH output file. In either case, EDSMAC will load the data. If you scan through the EDSMAC inputs, you'll see the EDCRASH-generated impact velocities entered for the initial velocities. The initial positions, friction and wheel lock-ups will be those you entered in EDCRASH. The vehicle data will be the same as in your EDCRASH run. All the remaining data are EDSMAC-specific and are assigned automatically, every time you run EDCRASH—just in case you might want to run EDSMAC.

The main purpose of this Technical Session is to properly set your level of expectation. The calculations in EDCRASH are not connected to the calculations in EDSMAC. In fact, the two calculation methods are completely different. Therefore, merging EDCRASH results directly into EDSMAC does not ensure a successful simulation—that is, the simulated rest positions may not match the actual rest positions, and the simulated damage may not match the actual damage. The reason for this are discussed below.

During the impact phase, EDCRASH determines the speed change using an impulse equation (either linear momentum or damage). In each case, the procedure is performed only once. The departure angle is not determined by the procedure, but by your scene data (impact and rest positions, etc.). Rotation is not considered because angular momentum is not used. On the other hand, EDSMAC determines the speed change by balancing the inter-vehicle forces at discrete time intervals. The departure angle is determined by the cumulative effect of the inter-vehicle forces and moments. Because even a slight repositioning of the vehicles at impact may change the moment produced by the impact force, EDSMAC may predict significant changes in both departure angle and separation angular velocity. EDCRASH, on the other hand, is relatively insensitive to these changes in most cases.

During the post-impact phase, EDCRASH determines the separation velocity based on the total tire force and distance traveled, much like a traditional energy analysis. Steer angles are not required and the individual wheel lock-ups are averaged into a single value. The path direction is determined by the user's input. On the other hand, EDSMAC determines the path calculated at each individual tire. Changing an individual tire force changes the force balance and redirects the vehicle. Likewise, steering (which had no effect on EDCRASH) will also redirect the vehicle.

Thus, you can see that trial and error is still required, even if your EDSMAC inputs come directly from EDCRASH. The main benefit of starting with EDCRASH is that you're likely to be much closer on your initial attempts, thus, reducing the amount of trial and error.

After achieving an acceptable match in EDSMAC, you should return to EDCRASH and modify your input if you changed your impact positions or drag factors (friction and wheel lockups).

Technical Session

Our Technical Session this time discusses the coefficient of restitution, *r*, of two colliding vehicles. Restitution is found in both EDCRASH and EDSMAC, but, as we'll see, it is used quite differently.

According to classical physics, restitution of two colliding "particles" is defined as the negative ratio of the rebound velocity to the approach (impact) velocity, or:

$$-\frac{(v_2 - v_1)_{sep}}{(v_2 - v_1)_{imp}}$$

For most collisions, *r* is normally between 0.05 and 0.3 and is somewhat speed-dependent (very low speed impacts may result in even greater restitution). With the possible exception of Flubber™, the laws of physics dictate that *r* can never exceed 1.0. For practical reasons, *r* can never be less than 0.0 (see following comments regarding EDCRASH).

In EDCRASH, *r* is calculated at the end of the run, *after* the separation and impact velocities have been determined. Because vehicles are much larger than particles, it becomes necessary to consider the orientation of the vehicle at impact by computing *r* from the velocity components along a line between the vehicle CGs (these components are displayed during EDCRASH output, see "Relative Velocity Data"). In addition, the collision actually occurs over a finite time period, typically about 100 to 150 ms. During the collision, the vehicles may rotate a significant amount if the collision is sufficiently non-central. During this time, the orientation of the line between the CGs changes! As a result, the classical value of *r* may become negative. EDCRASH reports this as a warning message. Users should pay attention to this message for central and near-central collisions, because it implies the striking vehicle is going through the struck vehicle like a ghost! However, for non-central collisions reporting a relatively small negative restitution (say, 0.0 to –0.2), the diagnostic may usually be ignored.

In EDSMAC, restitution is an input parameter and is used quite differently. Restitution cannot be simulated directly because the separation velocity is unknown until *after* the simulation has been executed. Therefore, EDSMAC models restitution in the collision algorithm by allowing the collision vectors to "grow back" a small amount. This dynamic crush is modeled by a 2nd order polynomial curve fit to experimental data. The "coefficients" entered into EDSMAC are the coefficients of the second order polynomial – NOT coefficients of restitution! This approach does not result in a predictable restitution for each accident, but for most accidents and vehicle sizes, the default values result in restitution of about 15 to 25 percent.

McHenry published values for "plastic" and "elastic" impact, intended to result in restitution values of 0.0 and 1.0, respectively. However, experience has shown that it is difficult to achieve restitution much less than 0.15 regardless of the chosen coefficients; experience has also shown that the elastic coefficients often cause collision phase errors in EDSMAC.

For details regarding the calculation of restitution in SMAC, refer to EDC Library Reference No. 1052.

Technical Session

This month's Technical Session describes a very useful (and quick!) use of EDCRASH, called the "Damage-only" analysis. We will also discuss some potential limitations.

Simply stated, a damage-only analysis is an analysis without using scene data. Before going any further, it must be stressed that it is physically impossible to calculate impact speed without an estimate of separation velocity (which of course, requires scene data). However, there are many cases in which you can infer that the separation velocity is small enough to be neglected. Examples are impacts with immovable objects, such as large trees and bridge abutments. Under these circumstances, the impact speed is equal to the delta-V (assuming negligible restitution).

Using EDCRASH, a damage-only analysis is performed using the following procedure:

1) Enter a valid class category for vehicle 1, and select class 11 (fixed barrier) for vehicle 2. EDCRASH will ask for a CDC for vehicle 1 only (a barrier has no damage profile, and thus, no CDC). Next, enter a valid stiffness category for vehicle 1 and select category 11 (fixed barrier) for vehicle 2. By making these selections, EDCRASH will assign "infinite" mass and stiffness for vehicle 2. (NOTE: The barrier may be entered as vehicle 1 or vehicle 2; it makes no difference.)

2) Enter ***no*** to EDCRASH's request for scene data (question 6). EDCRASH will skip forward to the Damage Data question. Enter the damage width, depths and offset for vehicle 1 (again, vehicle 2 has no damage profile). Be sure to include induced damage in your measurements!

3) Finally, replace any default values (dimensions, inertias, crush stiffnesses) for vehicle 1 (again, the values are N/A for vehicle 2).

EDCRASH will report the results for delta-V, as well as your entered values for the damage profile and vehicle properties. Notice that impact speed is not reported. However, as stated above, the impact speed is approximately equal to the reported delta-V.

Factors which influence the accuracy of the results include, as always, the integrity of your data, especially the damage data and stiffness coefficients. Because no scene data are entered, EDCRASH can perform no comparative diagnostics between damage-based and scene-based results. Also, because vehicle 2 is a barrier and has no damage profile, its damage energy is set equal to zero and the force is set equal to the force on vehicle 1. If the barrier was a small tree that actually moved during the collision, such movement represents an energy absorption which is not considered, so your EDCRASH results will be conservative.

If the tree broke off, two problems are encountered which make the damage-only analysis not useful. Firstly, significant energy was probably absorbed by the tree (this energy is ignored), and secondly, the vehicle probably had residual velocity, thus, the impact speed will not be equal to the delta-V.

Technical Session

This Technical Session discusses the method used by EDCRASH to calculate linear and angular separation velocities. In many cases, EDCRASH uses the same methods you would use for hand calculations. In all cases, EDCRASH calculates the path length from the user-entered path positions (impact, rest, and optional point on curve and end of rotation). The path length, *L*, is calculated using simple plane geometry:

$$L = \sqrt{(x_r - x_i)^2 + (y_r - y_i)^2}$$

where the subscript r is rest and i is impact. In the case of a curved path, a rather lengthy linear algebra process is used to calculate the path length. (It is worth noting the process assumes a CONSTANT radius!)

During input, the user enters the tire-ground friction coefficients, μ, for each vehicle and the rolling resistance at each wheel (actually, the percent of available friction). From this data, EDCRASH calculates the average lock-up:

$$\Theta = \frac{(\% r/f + \% l/f + \% r/r + \% l/r)}{4}$$

The formula used by EDCRASH to calculate velocity depends on your impact-to-rest path data. If you specify no rotating or lateral skidding (by answering NO to question 12 (veh. #1) of 19 (veh. #2)), EDCRASH uses the traditional energy formula:

$$V_{sep} = \sqrt{2g\mu\,\Theta L}$$

If you specify no rotation (by answering NONE to question 17 (veh #1) or 24 (veh #2)), or if the change in heading angle is less than about 5 degrees, or if the ratio of linear to angular displacement between impact and rest is more than 500 (i.e., the rotation is small compared to the path length), EDCRASH calculates the post-impact sideslip and modifies the average lock-up accordingly. Then EDCRASH calculates V_{sep} using the same energy formula:

$$V_{sep} - \sqrt{2g\Theta_{mod} L}$$

Note that EDCRASH accounts for post-impact sideslip for you! If you increase the wheel lock-ups to account for sideslip, EDCRASH will increase it again resulting in an over-estimate of V_{sep}.

If none of the above conditions exists, the vehicle has a significant post-impact rotation. This fact changes things in a significant way. Because of vehicle rotation, the forces are no longer constant, thus the deceleration is not constant. Therefore, the simple energy formula no longer applies. This really complicates things, because there is no *closed form* solution for calculating V_{sep}. In 1966, Marquard provided a solution that included some very restrictive assumptions. When McHenry developed CRASH in 1976, he started with Marquard's solution then added several empirical coefficients derived from simulations. The resulting method removed the restrictions imposed by Marquard. The method is described in several references (see EDCRASH references 1, 2, 3 and 16).

An energy-based hand calculation for vehicles which rotate is subject to error because it requires an estimate of the average deceleration rate. This estimate is extremely difficult. Some texts suggest a multiplication factor of 0.85. However, this only applies to one specific ratio of linear to angular displacements and wheel lock-ups, and will be errant for any other ratio.

Once the separation velocity has been estimated, simulation provides a useful check. If the estimate is reasonable, a trajectory simulation should result in a reasonable match between the simulated path and the actual path. That check is the purpose of the trajectory simulation in EDCRASH. We will save a discussion of the Trajectory Simulation for a future technical session.

Technical Session

This month's Technical Session describes how to deal with warning messages produced by EDCRASH. First, it is important to remember that EDCRASH is a reconstruction. As such , it applies the laws of physics to your data, and works backwards to determine the initial conditions (impact speed, speed change). Warning messages are caused by two possible problems: The crash you are studying is not modeled well by the laws of physics assumed by EDCRASH (this is a topic in itself), and/or your data are errant. Unfortunately, there is nothing you can do about the first problem, which falls under the "limitations" category. Therefore, we will focus on possible data problems in the Technical Session.

EDCRASH produces three basic levels of messages: Fatal (impact speeds will not be reported), Warning (impact speeds are reported, but results are suspicious) and Diagnostic (impact speeds are reported, but there are inconsistencies between damage-based and momentum-based results). Fatal errors (Common Velocity Error and Spinout Error) and Warning message (vehicles rotating wrong direction, unequal impact forces, point-on-curve discarded, negative restitution, impact sideslip angle adjusted) must be resolved before proceeding to resolve diagnostic errors (the subject of this session).

The diagnostic errors are related only to differences between damage-based and momentum-based results. These messages are: difference between PDOFs is more than 10 degrees, difference between delta-Vs is more than 10 percent, and difference between damage energies is more than 30 percent.

The PDOF diagnostic is easy to remedy if you have confidence in your scene data: simply modify your PDOFs to be within 10 degrees of those reported by the momentum-based results (in the Summary of Results screen).

The Delta-V diagnostic is normally due to errant stiffness coefficients (again, assuming our scene data are correct). Simply increase or decrease the coefficients until the damage-based delta-Vs are within 10 percent of the momentum-based delta-Vs (again, refer to the Summary of Results screen) An important word of caution here: You must have a sound basis for these adjustments. Refer to the literature for good stiffness data and adjustment techniques (call EDC Technical Support if you require assistance).

The Damage Energy diagnostic is not as easy to remedy. In fact, it can be downright stubborn! Unfortunately, there is no straight forward way to instruct users on how to solve this problem, but we will give you a start in the right direction.

First, understand what the message says. Reported are three different estimates (only two for collinear collisions) of the energy lost during the impact phase. The first estimate comes from your damage data; therefore, it is changed in a very predictable way by changing your stiffness coefficients and/or damage profile. The second estimate is the difference in kinetic energy at impact and separation ($KE_{damage} = KE_{impact} - KE_{separation}$) where KE_{impact} is calculated using the damage-based impact speed. $KE_{separation}$ is always calculated from scene data). Therefore, increasing (decreasing) the damage parameters (stiffnesses and crush depths) will increase (decrease) the reported value. Increasing (decreasing) scene parameters (i.e., drag factors and path lengths) will increase (decrease) *both* KE_{impact} and $KE_{separation}$. You'll have to use trial and error to discover the net effect. The third estimate is the difference in kinetic energy at impact and separation ($KE_{damage} = KE_{impact} - KE_{separation}$) where KE_{impact} is calculated using the momentum-based impact speed. It is, therefore, sensitive to changes only in scene data. Because the impact speeds are a function of direction, we know of no way to predict accurately how a given change will affect the results. And often, changes which improve the momentum-base result adversely affect the damage + scene data result. EDC's recommendation is as follows: Using a pencil and paper, write down the three initial damage energy values. Then, based on your knowledge of the accident and the available data, make a logical change and compare the new energy values with the original values. This comparison will tell you if your changes helped or hurt. Respond accordingly—again and again—until the three reported values converge.

Two important facts to remember are: (1) Any data changes must ultimately be supportable, and (2) If your scene data are good, the changes you are making (and the time you are spending) are purely academic, since the effect on impact speeds will be negligible (oblique collisions use the momentum-based delta-V; the damage-base results are only provided as a confirming check).

Technical Session

This month's Technical Session answers one of the most frequently asked questions: How do I present my results in court? The primary answer can be stated in one sentence: Understand how the software works! Of course, there are many secondary issues—including legal issues. Let's assume you are preparing to go to court. You understand how the software works, but your attorney/client has never introduced computer-generated evidence during a legal proceeding. It's these legal issues we will address in this Technical Session.

Briefly stated, four steps are required: Qualify the expert, then qualify the software, the input and the output. In any case (whether or not a computer is used), the expert, as well as the input and output must be qualified; thus, the only additional step is to qualify the software.

First, you must realize your exact challenge: You must be prepared to "educate" not only your client, but the judge as well. The most common cause for rejection (of computer-generated evidence) by judges is that they do not have, or have not been given, the necessary background to understand what they are actually dealing with. The most often-cited requirement for admissibility is called the *Frye Test* (see *Frye v. United States*, 293 F. 1013 (D.C. Cir. 1923). This landmark 1923 case states that for a scientific method to be acceptable "it must have found general acceptance in the appropriate scientific community." Use of the Frye Test has been cited in numerous cases. EDVAP easily passes the Frye Test for two reasons: First, EDVAP software is used by over 700 accident research firms worldwide, including nearly all well-known engineering/scientific research institutes. Second, EDCRASH and EDSMAC have been thoroughly validated scientifically (see SAE 890740, 900102), and the results are available in the literature. Public validation is quite important, and many reconstruction programs have no scientific validation.

Using the above background information, software qualification testimony should be performed in the following manner: First, when your client asks what method you used in your analysis, respond with all methods (not just the computer methods), and describe why each is used. When describing the EDVAP method, take time to describe the history of development and use in highway safety. Tell the court about the numbers of researchers using the method, and the number of cases for which you have used the method. Also, describe how the program was validated.

Next, describe how the program works, using a three-step explanation: Describe the input requirements, provide an overview of the calculation processes (flow diagrams found in the *EDVAP Training Manuals* provide great outlines), and finally, describe the output parameters of importance to your case. Just like a doctor is a *user* of many sophisticated instruments in medicine, you are also a *user* of the computer program; you need not understand or explain how every single calculation is performed. But, you must understand the basic calculation process. However, if you really want the detailed calculation (or, if the court request it), the source code is available for every program.

It is important that you address the assumptions and limitations of the program and how they affect your particular reconstruction. You should have performed sensitivity studies to gain confidence in your results, and *always* present you results as a range based on the available data!

The above steps will usually be sufficient for the judge to allow your testimony. Occasionally, a judge will still balk at the testimony, usually because he or she requires more education (back to the Frye Test!).

If you are preparing to go to court using EDVAP and would like more background information, EDC has additional information. Please give us a call. We're glad to help!

Technical Session

This issue's Technical Session deals with the integration timesteps used in EDSMAC. (NOTE: This session assumes the reader has a basic understanding of how the EDSMAC collision model works. For details, refer to SAE Paper No. 880069 and The EDC Simulations Training Manual, EDC Lib. Ref. No 1055.)

During the input session (question 3), the user assigns individual timesteps used during the collision phase (DTCOLL; normally 0.001 sec), during the first 100 timesteps after the collision phase (DTSEP: normally 0.010 sec) and during the remainder of the trajectory (DTTRAJ; normally 0.050 sec). In addition, DTSEP is used at the start of the run until either (a) a collision occurs, and the timestep changes to DTCOLL , or (b) 100 timesteps occur without a collision, and the timestep changes to DTTRAJ. A firm understanding of the above rules helps immensely when it comes to solving EDSMAC collision phase errors which may occur during high-speed collisions and "barrier collisions" (quotes are used here because EDSMAC really cannot analyze collision with infinitely stiff objects, such as barriers; however, it is often possible to use EDSMAC despite this apparent limitation).

Consider the following scenario: You are studying a head-on collision between two vehicles travelling 55 mph (= 80.7 ft/sec = 968 in/sec). The initial positions of the vehicles are 404 ft (4848 in) apart. At 0.050 second timesteps, the vehicles will collide after the 50^{th} timestep (4848 in / 968 $^{in}/_{sec}$ / 0.050 $^{sec}/_{timestep}$ = 100 timesteps; 50 timesteps when both vehicles are traveling towards each other). Now, *HERE'S THE PROBLEM*: After exactly 50 timesteps, the vehicles are still 8 inches apart. (The math is left to the reader. Call EDC Technical Support if you need help!) Since no collision has yet been detected, the integration timestep is still 0.050 sec, during which time the vehicles each move another 96.8 inches towards each other. During the next timestep, EDSMAC will detect the collision, but the overlap has now become 44 inches per vehicle! The collision model is usually not able to balance the collision force when the damage depth goes from zero to 44 inches in one timestep.

For barrier collisions, especially pole impacts, the problem is worse. Imagine how confused the collision routine might become if the vehicle moves about 50 inches each timestep and the pole is only 12 inches deep. The vehicle could actually pass completely over the pole during a single timestep.

Fortunately, there is a relatively simple solution and it does not require you to get out a calculator. Simply use Run-time Graphics (RTG) during the run and watch the vehicles (or vehicle and pole) as they come together. Your task is to make sure that, during initial contact, the vehicle-to-vehicle overlap is relatively small (say, about 10 inches or so; there is no exact rule and even 10 inches may be too much for a pole impact). This can be accomplished by moving the vehicles' initial positions slightly or reducing the integration timestep

One parting thought: Imagine how the above problem is exacerbated if the vehicles move more than 100 timesteps before they collide. (Ponder that one…!)

Technical Session

The new version of EDCRASH (ver. 4.5) contains a new type of graphic display, called *a momentum diagram*. A momentum diagram is a very useful tool, especially when debugging your results. EDCRASH will display two momentum diagrams, one based on delta-Vs calculated from damage and the other based on delta-Vs calculated from the scene data.

The momentum diagram is an illustration of the conservation of linear momentum, which states that the pre-impact momentum (product of mass and velocity) is equal to the post-impact momentum.

The diagram displays four basic vectors: the pre-impact momentum vector and post-impact momentum vector of each vehicle.

To assist in clarity, the momentum diagram is constructed one vector at a time. The scale is automatically selected so the diagram will be centered and fit on the screen. When you view each vector, you should consciously realize it represents a vehicle's quantity of motion in both magnitude and direction. The first vector drawn represents the magnitude and direction of the pre-impact momentum of vehicle 1. The second vector represents the magnitude and direction of the pre-impact momentum of vehicle 2. Since it is laid "head-to-tail" with the pre-impact momentum vector of vehicle 1, the total distance and direction from the start of vehicle 1's vector to the end of vehicle 2's vector represents the total momentum of both vehicles just prior to impact.

Next, the above process is repeated for the post-impact momentum (the vectors are dotted to help distinguish them from the pre-impact vectors). [Editor's Note: In HVE-2D, the pre-impact vectors are represented by short dashes, and the post-impact vectors are represented by long dashes.] The first dotted vector drawn represents the magnitude and direction of the post-impact momentum of vehicle 1. The second dotted vector represents the post-impact momentum of vehicle 2. Naturally, by conservation of momentum, the head of the second vector ends at the same place its pre-impact momentum vector ended

Two additional vectors are drawn: The first is a line representing the total momentum (pre- and post-impact). The second is a fat line showing the difference between vehicle 1's pre-impact and post-impact momentum. By definition, this difference in momentum is called the principal direction of force, of PDOF. There's a lot of information packed into this single diagram!

The diagram produced by damage should (theoretically) look identical to the one produced by scene data. Both diagrams use exactly the same vectors for post-impact momentum (you can compare any two diagrams to convince yourself). Therefore, any differences between the diagrams are an illustration of differences between the damage-based method and scene-based method of computing delta-V. Thus, this diagram is useful as a visual comparison of the two methods.

For both diagrams, the directions of the post-impact vectors are the same; they are determined by the departure angles. For pre-impact velocity vectors, the directions for the scene-based diagram will always be determined by the sum of the user's input for pre-impact heading and sideslip angles. However, the direction for the damage-based pre-impact vectors is determined by the components calculated for the user-entered PDOFs and pre-impact sideslip angles, and the calculated departure angles. Any difference between the direction displayed on the screen and the direction entered by the user is the result of an inconsistency between your PDOFs, pre-impact sideslip angles is suggested.

Technical Session

The technical session this issue deals with how EDC simulation programs simulate tiremarks (tire skids and scuffs). Tiremark simulation is included as part of the tire model. Two conditions occur which can cause tiremarks—locked-wheel braking and exceeding the tire's sideforce limit.

Anytime you enter a braking force (using the Driver Input Tables) which is greater than the available force (i.e., friction times vertical tire load), the tire model turns on the skid flag (a non-zero number which indicates the tire is skidding) for the current integration time step.

The second condition which causes simulated tiremarks occurs if the tire's sideforce limit is exceeded (a condition called "saturation"), whether by steering or any other condition which induces yaw. (Technically, this produces a tire "scuff," rather than a skid. However, the tiremarks displayed by EDC simulation programs do not distinguish between skids and scuffs.)

According to the tire model, saturation and associated tire scuffing occur when a parameter called non-dimensional sideslip, alf, exceeds 3.0. Mathematically:

$$\text{alf} = \frac{C_\alpha \operatorname{Sin}\alpha}{\mu \, Fz}$$

where C_α = tire cornering stiffness (lb/radian)

α = tire slip angle (radians)

μ = tire friction coefficient

Fz = Vertical tire load (lb)

If alf exceeds 3.0, the skid flag is turned on.

Graphically, EDC simulations simply look for contiguous time increments during which the skid flags are turned on. When found, a solid line is drawn between the associated X, Y tire coordinates.

Inspection of the above formula gives some insight regarding when scuffing occurs. The value of alf increases with stiffer tires, higher slip angles, lower friction and lower vertical tire force.

The above is a brief description of some features included in the Fiala tire model. Additional information on this model is contained in the EDSVS/EDVTS Program Manuals and Lateral Forces at the Rolling Pneumatic Tire (EDC Library Reference 1011).

Technical Session

This month, we deal with the computation of EDCRASH and EDSMAC stiffness coefficients from test data. First, let's lay some basic groundwork.

When you select a stiffness category, EDCRASH uses a built-in table which automatically assigns the vehicle's *A* and *B* stiffness coefficients. These coefficients describe the crush resistance of the vehicle, and play an important role in calculating the damage-based delta-V. *A* defines the force per inch of damage width required to start crushing the vehicle. *B* defines the force required for each additional inch of crush per inch of damage width.

The *A* and *B* coefficients used by EDCRASH are the same as those used by CRASH3. Frankly, there are shortcomings in some of this data (refer to SAE Paper No. 900101 and upcoming SAE papers). To improve the results, EDCRASH allows you to replace the table data with coefficients derived from actual crash tests. EDC publishes *A* and *B* coefficients for over 500 vehicles manufactured between 1970 and 1984.

Coefficients for newer cars or cars not included in EDC's publication can be calculated from crash test data using the following formulas and example:

Test Data:

test weight, W = 3200 lb
test delta-V, ΔV = 35 mph
= 616 in/sec
average crush, C_{avg} = 25.3 in
length of damage, L = 62 in

Assume:

b_0 = no-damage speed = 5 mph
= 88 in/sec
g = accel of grav = 386.4 in/sec^2

Calculate:

b_1 = crush rate = $(\Delta V - b_0)/C_{avg}$
= (616-88)/25.3 = 20.9 in/sec/in

A = Wb_0b_1/gL
= (3200)(88)(20.9)/(386.4)(62)
= 246 lb/in

B = Wb_1^2/gL
= $(3200)(20.9^2)/(386.4)(62)$
= 58.3 lb/in/in

The above *A* and *B* coefficients could then be used in EDCRASH to improve the results. If you are an EDSMAC user, you can use the same formulas with an important change: Set b_0 equal to 0. This has the effect of causing the force vs deflection curve to go through the origin, consistent with the SMAC collision model. Using these formulas, the *B* coefficient for EDSMAC equals 79 (the calculations are left as an exercise). Interestingly, the trajectories simulated by EDSMAC (or SMAC) are not very sensitive to the selected stiffness coefficients. However, the simulated crush depth and peak accelerations are affected. For a discussion of the subject, refer to our SAE Technical Paper No. 900102 (call us and we'll send you a free copy).

Crash test results are frequently published in the Accident Reconstruction Journal (for subscription information, call 301-843-1371). The results may also be obtained from NHTSA, which conducts the tests as part of the New Car Assessment Program (NCAP), and from Neptune Engineering (www.neputune-engineering.com).

Technical Session

This month, the subject is a new diagnostic which is available from EDCRASH, Version 4.4. The new version calculates and displays the coefficient of restitution. Classical physics tells us restitution is the ratio of the separation (rebound) velocity to the impact velocity. Mathematically,

$$\text{Restitution} = \frac{Vsep_2 - Vsep_1}{Vimp_1 - Vimp_2}$$

The coefficient of restitution should always lie between 0 and 1. A value of 0.00 corresponds to a *plastic* impact. An example of a plastic impact is the collision of two balls of clay. After their collision, they are stuck together—they have identical separation velocities. A value of 1.0 corresponds to an *elastic* impact. Because energy is always absorbed during a collision, no collision is perfectly elastic. A "nearly elastic" collision occurs when you drop a superball on a hard concrete surface. Its rebound velocity is nearly identical to its impact velocity, so it bounces nearly as high as the elevation from which it is dropped.

Typical automobile crashes result in a coefficient of restitution between 0.05 and 0.30. Slower closing velocities typically result in higher restitution. Intuition tells us this is true. Consider a 5 mph frontal collision with a concrete wall; the car is not damaged and bounces back (high restitution). Next, consider a 35 mph head-on collision between two cars of equal weight; the cars are severely damaged and there is virtually no rebound (low restitution).

A negative value for restitution gives us some important feedback. It says that the striking vehicle has a higher separation velocity than the struck vehicle! This is impossible unless these are "ghost vehicles" and the striking vehicle passes through the struck vehicle. EDCRASH uses this logic to help you arrive at consistent scene data. If you get this message, your restitution is negative. Eliminating this message requires evaluating the data which caused it; only scene data can cause this massage. The following are potential problems: striking vehicle's drag factor is too high (or struck vehicle's is too low), striking vehicle's path to rest is too long (or struck vehicle's is too short). Because these velocities are computed along a line connecting the vehicle CGs, the orientations at impact and departure angles can also play a role.

When working to eliminate this message, like other messages, the best suggestion is to make a change and see how it affects the results. Proper corrections will cause the value to become positive.

Technical Session

This month's Technical Session uses a new feature of the recent RTG release *–pre-impact positioning*. When transferring from EDCRASH to EDSMAC, the vehicle positions begin at impact. But for final reports and to represent pre-impact conditions, the EDSMAC simulation can be improved by "backing up" the vehicles.

The method for doing this has been greatly simplified with RTG. Here's how you do it:

1. After transferring from EDCRASH to EDSMAC, enter "N" for "Any changes!".
2. Activate RTG mode with F2.
3. Pause the simulation by pressing the *spacebar*.
4. Note the vehicle positions and angles at impact.
5. Press *ESCape* to return to "Any changes?".
6. Enter 1 to change General Data.
7. Change the initial vehicle headings for both vehicles by adding 180°
8. Change the Wheel Force and Steering Tables to allow for any braking, accelerating, or steering. (Remember to reverse any angles when "backing up" the vehicle.)
9. Run the simulation. Let it run until the vehicles are in the desired "backed-up" positions and pause the simulation as in Step 3 above. Again note the vehicle positions and angles.
10. Press *ESCape* again. This time, change the vehicle positions to those in Step 7 above. Now, subtract 180° from the heading angles to return the vehicles back to their original angles.
11. Now *run* the simulation and the vehicles should start in their desired pre-impact positions.

There are some limitations to this process. For one, the rear overhang and front overhang should be about the same. Also be aware of braking or steering conditions. But the end result is a *complete* presentation—both before and after the collision.

Technical Session

This month's Technical Session describes an unpublished feature in EDCAD—rotating, scaling and positioning an accident site.

Have you ever spent hours fine-tuning an EDCAD drawing, only to find out that the client wants it rotated 30°? Or copied to another portion of the accident site?

If the accident site does not contain any shapes, this is an easy task. If is has shapes, remove the shapes using INEDIT. Then follow these procedures:

1. Exit to DOS, making sure you save your new drawing with an .SHP filename extension. For instance, F8906CAD.SHP.

2. Copy the file you created above into the SHAPES subdirectory. For the directory containing EDCAD, enter:

COPY filename SHAPES [return]

where, filename is the name of the file saved in step 1.

3. Start EDCAD by entering:

EDCAD [return]

4. Start a new drawing by entering:

1

5. Enter the SHAPES command and the Select option. The file you created in step 3 will appear as one of the possible shapes. Select the file.

6. Now you can position, scale, or even define the angle of your entire drawing via the SHAPE command.

Some additional comments. If you want to maintain the same scale as the data received for a merged EDVAP analysis program, use a scale factor of 1 in step 6.

Likewise, remember that EDCAD does not allow "shapes within a shape file". For this reason, the drawing you start out with in step 1 must NOT contain any shapes.

Technical Session

This month's Technical Session deals with a slightly different subject: Courtroom admissibility. Since EDVAP software is used extensively in courts, this subject can be important to many users.

Your complete reconstruction is wasted if, after performing a thorough analysis, the court refuses to admit your testimony. Investigators who understand the calculations performed by the computer program seldom have difficulty having their results admitted in court. Four steps are generally involved in the qualification process:

- Qualify the expert
- Qualify the software
- Qualify the input
- Qualify the results and opinions.

The expert must always be qualified before allowing his/her opinions to be expressed in court. However, beyond the normal requirements the expert must demonstrate an understanding of the use of computers.

Qualifying the software is quite simple: It must be shown to produce reliable results when supplied with good input data. The best way to do this is to rely on a validation study. A history of use by our peers is also beneficial. Because the expert will be questioned about the software, an understanding of how the program calculates is essential. This usually involves addressing its weaknesses, not just its strengths, and showing that the software is applicable to the accident.

When good accident site and vehicle data are available from first-hand inspections, the task of qualifying the input data is relatively easy. However, the quality of the input data is often beyond the investigator's control, due to a limited on-scene investigation or the age of the case before the investigator becomes involved. Under these circumstances, qualifying the input can be a big hurdle. Using reasonable ranges for known or estimated parameters usually solves this problem, but sometimes yields a useless range of results (i.e., the speed was between 15 and 65 mph!). Still, it is important to know that, given the available data, a useful reconstruction is sometimes not possible.

Finally, the investigator presents his/her results and opinions. This step is the same regardless of the methods used: Simply apply the information derived from your reconstruction to explain what happened (and maybe, what could not happen).

Some additional thoughts to remember: The computer is just one tool. You must never base your reconstruction on just one tool. Remember that your experience is also a tool, as are your evaluations of witness statements and general understanding of the laws of physics.

Stay within the scope of the program's intended purpose. Stretching the program's basic assumptions and limitations opens you up for scrutiny. Be ready to defend your assumptions.

Finally, do not exaggerate the accuracy of the results, even though your computer output reads 54.5 mph you would not report the speed as such. By using a range of reasonable input, you will find (and should report) the speed as a range of values.

Technical Session

One of the most difficult pieces of data to find is a vehicle's moment of inertia, I_{zz}. This month's Technical Session provides a pretty good solution for passenger cars.

I_{zz} describes a vehicle's resistance to an increase or decrease in angular velocity about the z-axis. A vehicle with a large moment of inertia is more difficult to start or stop spinning. I_{zz} is related to two factors: The vehicle's size and mass.

Another property called radius of gyration, R, is related to I_{zz} by the formula $I_{zz} = mR^2$. Close inspection reveals that R has the same physical meaning as I_{zz}. However, two vehicles of the same size can have vastly different I_{zz}, depending on their masses. But, both vehicles will have nearly the same radius of gyration.

Based on this fact, we developed a graph of *radius of gyration vs. wheelbase* for each of the NHTSA class categories (see Table 3 in the EDCRASH Program Manual, attached). With the interesting exception of Class 1, the graph formed a straight line.

For some unknown reason, Class 1 produced an R value way below the graph's line. This suggests that the R values estimated for vehicles outside the wheelbase range shown in the graph may be suspect.

The computed formula for the line is R = 0.37 x Wheelbase + 18.1. Statistical analysis revealed the correlation coefficient equal to 0.99, meaning the straight line is a very good predictor for R, given a vehicle's wheelbase. (A value of 1.0 indicated perfect correlation.)

The following is an example: A vehicle has a wheelbase of 107 inches and weighs 3100 lb. Interpreted from the graph or computed from the formula, the radius of gyration is 57.7 in.:

$$
\begin{aligned}
R &= 0.37 \times 107 + 18.1 \\
&= 57.7 \text{ inches}
\end{aligned}
$$

Its mass is 3100 / 386.4 = 8.02 $\text{lb-sec}^2/\text{in}$. Then the moment of inertia is:

$$
\begin{aligned}
I_{zz} &= mR^2 \\
&= 8.02 \times 57.7^2 \\
&= 26{,}710 \text{ lb-sec}^2\text{-in.}
\end{aligned}
$$

Table 3 - Vehicle Class Categories [2]

PARAMETER	CLASS CATEGORIES →								
	1	2	3	4	5	6	7 (Vans)	8/9	10 (Movable Bar.)
Wheelbase (in)	80.9-94.8	94.8-101.6	101.6-110.4	110.4-117.5	117.5-123.2	123.2-150	109-130	No Data for Category 8 (Pickups) or 9 (Front wheel drive) Select Category 1 - 6 According to Wheelbase	120.0
Track (in)	51.1	54.6	58.9	61.8	63.7	63.7	67.6		60.0
Length (in)	159.8	174.9	196.2	212.8	223.7	229.4	183.6		180.0
Width (in)	60.8	67.2	72.6	77.0	79.8	79.8	79.0		78.0
A (in)	45.1	46.3	51.3	54.7	58.1	60.1	48.5		54.0
B (in)	48.1	50.1	55.5	59.2	63.0	65.1	68.5		66.0
X_f (in)	76.0	83.3	89.8	98.8	101.8	104.2	75.6		84.0
X_r (in)	83.8	91.6	106.4	114.0	121.9	125.2	107.0		96.0
Rsq (in^2)	2006	2951	3324	3741	4040	4229	3713		4024
I_{zz} (lb-sec^2-in)	11434	23313	30514	41114	50864	58106	41586		41906
Weight (lb)*	2202	3053	3547	4247	4865	5309	4300		4000
$C_{alfa,f}$ (lb/deg)	94	131	152	182	209	228	209		1000
$C_{alfa,r}$ (lb/deg)	88	121	141	168	193	210	193		1000

LEGEND:
A = Distance from CG to front axle
B = Distance from CG to rear axle
X_f = Distance from CG to front of vehicle
X_r = Distance from CG to rear of vehicle
Rsq = Radius of gyration squared
I_{zz} = Yaw moment of inertia
C_{alfa} = Tire cornering stiffness
*Weight includes 300 lb occupant loading

NOTE: Fixed barriers (category 11) are assigned inertial properties (mass, yaw inertia) of 10^6. These values are not reassignable. The barrier's dimensions (100" by 100") can be reassigned, however.

Technical Session

This month's Session describes two major files used by your computer whenever you turn the computer on. The two files are:

- AUTOEXEC.BAT
- CONFIG.SYS

Every time you start your computer, DOS searches for these files in the root directory on the disk from which DOS was started. Both files are optional; if either does not exist, DOS uses built-in defaults.

The AUTOEXEC.BAT file contains a series of DOS commands that tell your computer what to do every time you start DOS. We recommend having at least the following commands in your AUTOEXEC.BAT file:

PROMPT PG
PATH=[path1];...;[pathN]

The PROMPT command instructs DOS to display the current subdirectory as part of your DOS prompt. With the current subdirectory displayed on your screen, you will always know your subdirectory position in a hard disk.

The PATH command instructs DOS on the search sequence to locate a requested program. For example, if all DOS commands are in your \DOS subdirectory, a PATH=C:\DOS command will instruct DOS to first look in the current subdirectory for a given command, and then check the DOS subdirectory.

The CONFIG.SYS file contains a list of commands that sets up your computer. The commands define the peripheral equipment attached to your computer (sometimes called **DRIVERS**) and how they interface with your computer (for instance the country format of your keyboard). As part of the INSTALL process, EDVAP will properly update or create these files.

To organize your hard disk, we recommend your root directory contain only three files and subdirectories. The three files are:

- AUTOEXEC.BAT
- CONFIG.SYS
- COMMAND.COM

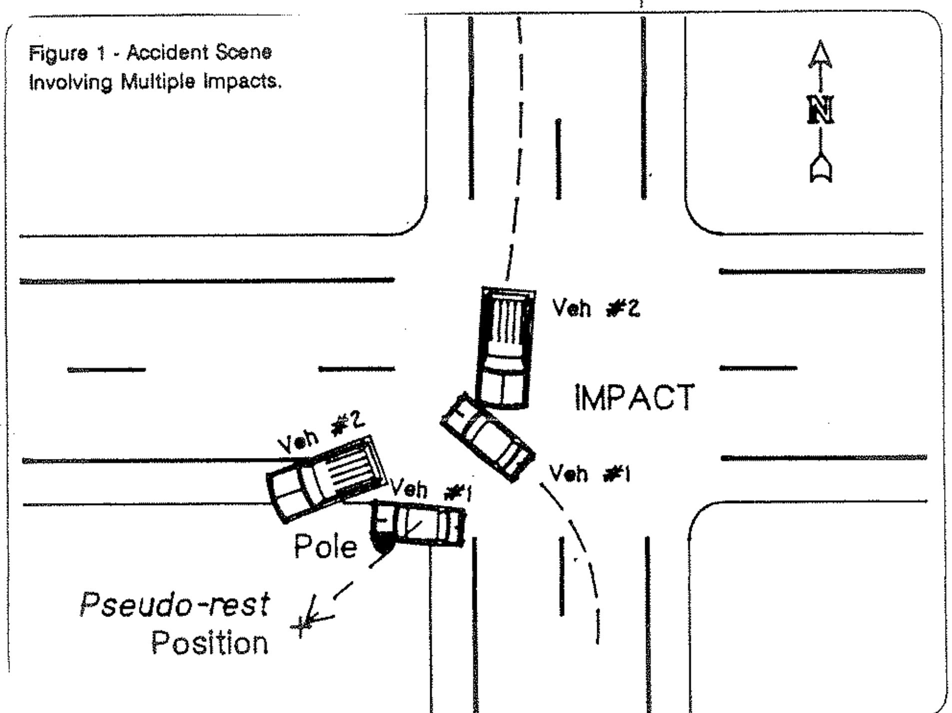
Each subdirectory contains a specific application along with related files and programs or additional subdirectories. This layered approach will help you locate programs or files and minimize the frequency of duplicate files. Obviously, one of the subdirectories would be \EDVAP.

Technical Session

This month's session describes how to use EDCRASH to analyze multiple impacts. A multiple impact occurs when two vehicles collide and then one or both vehicles subsequently collide with a tree or third vehicle. As we'll see, the procedure is really quite simple.

Let's say vehicle #1 turns left in front of vehicle #2 and is struck in the side (see Figure 1). As a result of being struck, vehicle #1 slides into the utility pole located in the southwest corner of the intersection. The collision is analyzed as follows:

Figure 1 - Accident Scene Involving Multiple Impacts.



Analyze the last collision first—Perform a damage-only analysis between veh #1 and the utility pole (call the pole veh #2 and assign it class and stiffness category 11, immovable barrier). This simple analysis will give you the delta-V for veh #1. Since the pole stopped the car, $V_{s,car}$ (separation velocity of the car) equals zero and its impact speed is equal to the delta-V. Next, the critical step: use this impact speed to determine where veh #1 would have come to rest if it had not been stopped by the pole. This distance is estimated by using the skid-to-stop formula: $s=V^2/30f$. In our example, s=15 feet, so extend the path between the original impact and the pole impact by 15 feet (see Figure 1, pseudo-rest).

Now you can analyze the first collision in the usual fashion—Document the impact and rest positions for both vehicles, using the pseudo-rest position for veh #1. An important note: You can theoretically continue this process for a third or even a fourth collision. However, the process of making estimates based on previous estimates is likely to broaden the range in your final results for impact speeds in the first collision. A sensitivity analysis should be performed to determine this range.

Another important limitation of this analysis is the need to distinguish the two damage profiles for vehicle #1 (otherwise, we wouldn't be able to separate the energy absorbed by the car-to-car impact).

Technical Session

This month's technical session is a continuation of our last Newsletter on the subject of Common Velocity Error. Previously, we discussed *What* the message is and *Why* it is necessary. Now we will address the issues of *How* Common Velocity is determined, *When* the check is performed, and most important, *How to eliminate* the error condition.

How – The COMMON VELOCITY CHECK is performed by computing the velocity at each vehicle's damage centroid, a location computed from damage dimensions and the point where the PDOF is assumed to act. The centroid velocity is computed from the linear separation velocities (U-vel and V-vel) and the tangential velocity (a product of the angular velocity and the distance from the vehicle CG to the damage centroid). These separation velocities are displayed in the Complete EDCRASH output. The component of this centroid velocity in the direction of the PDOF is then computed. The COMMON VELOCITY CHECK compares this velocity for vehicles 1 and 2. The actual procedure is rather involved, but boils down to this: If V_{com1} differs for V_{com2} by less than 10 percent, the check is satisfied and the remaining calculation procedures are performed. If the check is not satisfied, the lower velocity is increased by 10%, the higher velocity is decreased by 10%, and the check is performed again. If the check is now satisfied, a warning message is issued and the remaining calculations are performed. If the check is still not met, a fatal COMMON VELOCITY ERROR in issued and further calculations are aborted.

When - The COMMON VELOCITY CHECK is performed when scene data are entered to insure the data for the two vehicles are compatible. If scene data are not entered, the check is not performed. Therefore, no Common Velocity warnings will be issued for a damage-only analysis and it is up to the user to decide if the vehicles reached a common velocity during impact.

How to eliminate – This subject is described in detail in the EDCRASH Program manual (see attached). The manual lists 5 steps to go through to help eliminate the error. Go through these steps in order. The keys are the SCENE, PATH , and TIRE/ROAD FRICTION DATA (these are used to compute the velocities) and the PDOFs (these decide the components for the velocity). The PDOFs tend to be significant for intersection collisions. By changing the PDOFs, the error can usually be eliminated. For some head-on collisions, it is very difficult to eliminate the error. REMEMBER: GRAPHICS displays the centroid velocities in both numeric and picture form.

No response by the user is necessary. However, refer to the information below *(COMMON VELOCITY ERROR)* for data which affects this warning message.

***** COMMON VELOCITY ERROR *** The vehicle contact interface failed to reach a common velocity, based on user-supplied data for impact-to-rest path length and rolling resistance.**

(FATAL)

The earth-fixed velocity at the damage centroid is computed for each vehicle separately, based on user-entered scene data and the PDOF. If this velocity differs between vehicles by more than 10 percent, an adjustment of the separation conditions is performed (see Common Velocity Warning, above). For collinear collisions, if, after this adjustment, the velocities at the damage centroid still differ by more than 10 percent, the common velocity assumption cannot be satisfied and EDCRASH aborts.

What To Do:

1. Go directly to Graphics and check first to see that all accident site Path Data (impact, rest, EOR and POC) were entered correctly. Then see if the damage data (damage profiles and PDOFs) were entered correctly. Use the Graphics.

   The velocity vector at the damage centroid of each vehicle is displayed in the Site Drawing if a common velocity error occurred. Check the relative magnitude (length) of each vector. Also, check the direction of each vector. The length and direction of both vectors should be approximately equal. Compare the present run with the next run to see if your adjustments had a positive effect.

2. Scrutinize the Path Data. Is it possible the impact positions and headings or the rest positions and headings were *not* what you thought? Were there pre-impact sideslip angles? Was there an EOR that you did not enter? Was the POC valid? Are you sure about the rotation direction? Scrutinize your Path Data.

3. Scrutinize the Tire/Road Data. Check the tire/road friction. Were the wheels *really* locked? Free rolling? In between? (If so, how did you arrive at your entries?) Scrutinize your Tire/Road Data.

4. Scrutinize the PDOFs. Is it possible they were greater (or less) than the values you entered? Try entering a range of possible values, checking the effect on your results. Scrutinize your PDOFs.

5. Finally, make sure the collision was not a sideswipe collision which violates the common velocity assumption (see **Assumptions**).

WARNING: Damage-based estimates for Magnitude of Principal Force grossly violate Newton's Third Law of motion. Review the output to determine required corrections to Damage Data and adjust as necessary. The Magnitudes of Principal Force should be approximately equal.

(Informative)

An analysis of the vehicle damage data is always performed, whether by using CDC-generated data or user-entered damage dimensions. This analysis results in the damage-based delta-V, the energy absorbed by damage and the *magnitude* of the principal force. This fact serves as an excellent check on the validity of the damage-based results. If the results are reasonable, then the independently-calculated forces should be approximately equal as well. If these forces differ by more than 100 percent, the above message is displayed.

What To Do:

1. Go directly to Graphics and check to see if the damage data (damage profiles and PDOFs) were entered correctly. Use the Graphics.

2. Scrutinize the Damage Data. Check the damage dimensions (width, crush depths, and damage offset) to be sure they were accurately measured. Remember to include induced damage.

Technical Session

This month's technical session deals with the subject of the Common Velocity Error found in all CRASH programs: What is it? Why is it necessary? How is it determined? When does it occur? And most important, how can I get rid of it?

What – All accident reconstruction algorithms that use a DAMAGE analysis procedure for determining delta-V (i.e. EDCRASH, CRASH3) are based on a fundamental differential equation. The equation defines the collision of two objects with deformable outer layers and is based on Newton's Laws of Motion. Its derivation is explained in detail in the EDCRASH Training Manual and the equation is summarized below:

$$\frac{d^2\delta}{dt^2} + \left(\frac{K_1K_2}{K_1+K_2}\right)\left(\frac{M_1+M_2}{M_1M_2}\right)\delta = 0$$

where K = crush stiffness
M = vehicle mass
δ = crush deflection

Mathematically, this is called a "boundary value problem" because certain conditions must be known before the differential equation has a solution. There are two such conditions:

(1) the initial rate of crush defection is equal to the closing velocity , or

$$d\delta/dt_{t=0} = V_{10} - V_{20}$$

where dδ/dt = the rate of crush between the vehicles and

$$V_{10}–V_{20} = \text{closing velocity}$$

(2) the final velocities at the damage region are equal, or

$$V_{1f} = V_{2f} = V_{common}$$

If these two conditions are met, the solution for the impulse used by the DAMAGE procedure (any CRASH program) is valid. If either condition is <u>not</u> met, the solution for the impulse is invalid, because the basic differential equation does not apply. The process for checking that the second condition is met is called the "common velocity check." If the process reveals the condition for a common velocity is not met, a COMMON VELOCITY ERROR is issued.

Why – Inspection of the above criteria reveals that the first condition is always met: The rate of crush defection between two vehicles is always equal to the closing velocity vectors. The second condition, however, is dependent upon data not even used by the DAMAGE analysis procedure: the vehicles' separation velocities. To insure the second condition is met, the common velocity check compares the vehicles' separation velocities <u>in the damage region</u> to confirm the velocities are about the same. If they are, the damage-based delta-V is valid. Otherwise, the damage-based delta-V is invalid because an underlying requirement has been violated.

We will complete this important discussion by discussing the HOW, WHEN and GETTING RID in our next Technical Session.

Technical Session

For this month's technical session, we will investigate a very simple equation with very important implications:

$$V_{impact} = V_{separation} - \text{delta-V}$$

This equation, in English, states "The velocity at impact is equal to the velocity at separation minus the delta-V" (the sign convention requires subtracting the delta-V). Since this is a vector equation, both the forward and lateral velocities must be considered. Fortunately, EDCRASH displays all this information.

Suppose you know the precise impact velocity of one of the vehicles. A typical example occurs when a stopped vehicle is struck from the rear (its impact velocity is zero mph).

You can use EDCRASH to estimate the speed of the striking vehicle even if no scene data is available. How? Let's look at the above equation. For the struck vehicle, you know its impact speed (zero). From vehicle damage alone, EDCRASH computes its delta-V. (This is intuitively obvious when we consider that during the collision phase, the vehicle's speed must increase until the vehicles separate.) If we neglect restitution (an important, but typical assumption), the striking vehicle's separation velocity is equal to the struck vehicle's separation velocity, which in turn we just discovered is equal to the struck vehicle's delta-V. Since EDCRASH also computed the delta-V for the striking vehicle, we have both the separation velocity and delta-V for the striking vehicle. Therefore, its velocity at impact is equal to

$$\begin{aligned} V_{impact} &= V_{separation} - \text{delta-V} \\ &= \text{delta-}V_{struck} - \text{delta-}V_{striking}. \end{aligned}$$

And all this without any scene data! (Here's the key: You actually had excellent scene data—You knew the impact velocity of one of the vehicles!)

Technical Session

This month's technical session deals with the subject of calculating the delta-V for a barrier collision. In the past, EDCRASH handled barrier collision by either the momentum solution or the damage solution, depending on the angle of the pre-impact velocity vectors (if these vectors are more that +/- 10 degrees from collinear, the momentum solution was used; otherwise, the damage solution was used). Sometime between 1982 and 1986, the CRASH3 code was changed by NHTSA. When EDCRASH Version 4 was released, this change was incorporated. Now, when a barrier collision is analyzed, two things happen: the damage solution is used and the common velocity check is not performed.

Refer to the equations for barrier collisions. Note for barrier collisions, only the energy for vehicle #1 is used. This reflects the definition of a barrier: A barrier is an object which absorbs no energy. Since it absorbs no energy, it cannot have a damage profile either. This is why no CDC/PDOF and damage profile are requested for a barrier. However, a barrier <u>can</u> have a delta-V (refer to the equations).

A couple of things are worth noting. First, note the vehicles *share* the impulse. Therefore, the ratio of the delta-Vs is inversely proportional to the vehicle weights (or masses). For example, if a 2000 lb car hits a 3000 lb car the delta-V for the 2000 lb car will be 1 ½ times higher than the delta-V for the 3000 lb car. This is a basic physics principle.

Second, note that an immovable barrier is made "immovable" simply by assigning it a default mass of 1,000,000 lb-sec^2/in. If you re-assign the weight, an immovable barrier could become movable.

Third, and perhaps most important, note that the delta-V of a particular vehicle is not proportional to its damage. Rather, the delta-V is inversely proportional to the weights of the respective vehicles as shown above. In fact, the sum of the damage energy is used to compute the impulse which is shared between the vehicles.

Technical Session

The topic for this session is how the collision dynamics are computed by EDSMAC. The input parameters which affect this process are found in questions 18 & 19 (refer to your EDSMAC manual). They are the exterior stiffness, AKV; the angle between force vectors, DELPSI; the incremental adjustment between the force vectors of vehicle #1 and vehicle #2, DELRHO; and the maximum allowable difference in the computed forces for two opposing force vectors, ALAMB. (The effect of intervehicle friction and restitution [question 20 & 21] will not be discussed here.) Below, we will describe how each of these values is used.

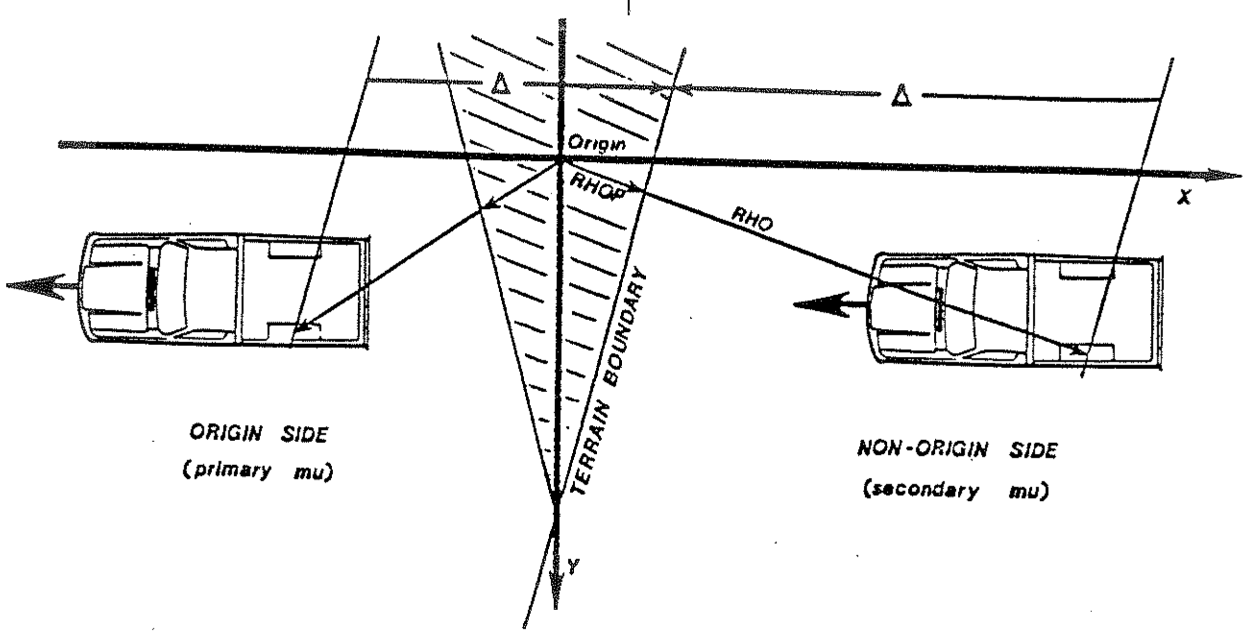
The basic model assumes the vehicle exterior is uniformly stiff (i.e., the wheel regions have the same resistance to crunch as the quarterpanels). For each collision time step (DTCOLL, see question no. 3), vectors called RHO vectors are drawn from the vehicle CG to the exterior. These "force vectors" begin along the vehicle-fixed x-axis (towards the front of the vehicle, psi=0) and are spaced DELPSI degrees apart. Each force vector acts like a rather stiff spring, having a stiffness, or spring rate, equal to AKV. The goal is simply to seek when interaction occurs between the springs and, once interaction is found, equalize the force between each set of springs (by Newton's 3rd law, they should be equal). Interaction is first sought by locating the corners of vehicle #2 from vehicle #1's CG (or point of reference). By doing this, the maximum potential "damage range" for vehicle #1 is found. Then, this damage range is scanned in a clockwise direction, from beginning to end, for each RHO vector seeking interference (contact) between the vehicle outlines. The first angle where contact is found is reported in the output as the "Beginning of Range"; the last angle were contact is found is called the "End of Range" (see Output, Vehicle Damage Ranges). The process is then reversed, locating the corners of vehicle #1 from vehicle #2's point of reference and the damage range for vehicle #2 is determined. The next step is to adjust the length of each of the RHO vectors found to be in contact with the other vehicle until the force between them equalizes. This is an iterative process which proceeds in the following manner for each RHO vector: First, the existing force is computed by multiplying the stiffness (AKV) by the deflection from the original surface. The resulting value for vehicle #1 is compared with vehicle #2. If the difference is more than allowed (ALAMB, usually about 20 lb), the length of the RHO vectors will be adjusted by increasing RHO for vehicle #1 and decreasing RHO for vehicle #2 (or vice versa). The amount of the RHO adjustment is DELRHO, usually about 0.2 inches. The adjustment will continue until the difference is less than ALAMB. The resulting length of all the RHO vectors for each vehicle defines its damage profile (see RHOBI, Output, Vehicle Damage Ranges).

Ever wonder why it takes EDSMAC so long to run? Consider that the above process takes place for every RHO vector for each vehicle at each collision time step! Execution time can be reduced by increasing DTCOLL from 0.001 to 0.01 seconds for your preliminary runs. Errors may occur during the collision phase, and it's a good idea to do a final run at 0.001 seconds.

Technical Session

In our last newsletter, we told you there was a problem with the terrain boundary option in EDSMAC. We traced the problem all the way back to the original CALSPAN formulation for CRASH and SMAC. We will describe the error and then its solution.

In the original code, the method used to determine if a wheel was on one side of the boundary or the other is as follows: First, the X and Y coordinates of the wheel were determined for the current time step (simple trigonometry). Next, the length of a vector caller RHO, running from the origin to the wheel, was determined (square root of the sum of the squares). Then, the distance, RHOP, along the RHO vector to the terrain boundary was determined. Finally, if RHO is greater than RHOP, then the secondary coefficient was used; otherwise, the primary coefficient was used. This concept is shown below. In the illustrated case, this method only works in the shaded area and the wrong coefficient is used. Outside the shaded area, the method doesn't work at all, and the secondary coefficient is used, even if none was specified (the default was set equal to 0.6).



The new method is based on simple algebra. First, if there is no terrain boundary, the primary coefficient is assigned and the test is by-passed. If there is a terrain boundary, the sign of its X-intercept is determined. Next, the X-intercept of a line parallel to the terrain boundary which goes through the wheel of interest is determined. If the sign of the difference between the two X-intercepts is the same as the sign of the terrain boundary X-intercept, the secondary coefficient is used; otherwise the primary coefficient is used.

This error has some effect on all SMAC runs, unless the secondary coefficient was set equal to the primary coefficient. The effect on delta-V is rather minimal because the collision force is so much greater than the tire force. The effect on the trajectory calculations varies according to the difference between the primary and secondary friction coefficient.

This error also affects CRASH runs (including EDCRASH) which used the Trajectory Simulation Option, unless the secondary coefficient was set equal to the primary coefficient. EDCRASH 4.1 and ESMAC 2.0 have the error corrected.

We suggest you verify this effect on your individual analyses.

Technical Session

We have been learning a lot from our EDCRASH Training Seminars at Northwestern. An example is the users' potential misinterpretation of rolling resistance input data (questions no. 27 & 28). Four entries are required, one for each wheel. Each entry may be thought of as either (a) the rolling resistance, or (b) fraction of wheel lock-up. The rule is simple: If you're thinking in terms of rolling resistance, divide by the coefficient of friction before entering the value. To illustrate the concept, let's look at the equation for straight line skidding. (Although the equations are different, the concept is identical.)

$$\Theta = \sum RR / 4$$

$$V = \sqrt{30\Theta\mu S}$$

Where:
- V = velocity (mph)
- Θ = the average rolling resistance (option 1)
- μ = coefficient of friction
- S = path length (ft)

If you select option 1 (highly recommended, since it allows you to account for the motion-resisting force at each wheel individually), the entered value will be summed and divided by four. The result is then multiplied by the coefficient of friction! (which you previously entered). If it is your intention to supply values corresponding to the rolling resistance at each wheel (which has nothing to do with the coefficient of friction), you must select the appropriate rolling resistance (approximately 0.01 lb/lb for a free-rolling wheel) and <u>divide</u> it by the coefficient of friction before entering the value into EDCRASH [i.e., if you wanted EDCRASH to assign a rolling resistance force for the right front wheel of 0.013 and you previously entered a friction coefficient of 0.7, you should enter 0.0186 (0.013/0.7)]. Later, during the calculations, EDCRASH will multiply the value you entered by the coefficient of friction, and the result is 0.0186*0.7 = 0.013 ... just what you wanted in the first place! The limiting value you may enter is 1.0 ... the equivalent of locked wheel braking, since EDCRASH multiplies it by the coefficient of friction (1.0*0.7 – 0.7). If you intend to supply EDCRASH with the fraction of wheel lock-up, you may enter the number directly. For example, if you believe a wheel is restrained approximately 50% by sheet metal contact, then you specify rotational/lateral skidding (questions no. 12 and 19). For more information about the inner working of EDCRASH, send for our EDCRASH Training Manual ($40).

Technical Session

Many EDVAP users have asked for information on improving the speed of EDSMAC simulations. In response to this, the user must first understand that the main processing of EDSMAC (not the input or output session) contains numerous calculations and uses the numerical processing power of the PC extensively. Presently, EDSMAC will take about 90 minutes on a PC of PC/EX (or other 4.77 MHz 8088 processor and about 30 minutes on an IBM PC AT (or other 6 MHz 80286 processor) for a normal simulation with a 4 second duration. These times may vary depending on the velocities of the vehicles and the severity of the impact (during the impact phase, the processing speed slows to about ¼ the processing speed of the pre-impact and post-impact phases).

Technologically, there have been some recent improvements in both hardware and software to reduce these times substantially. (You cannot simply replace your processor with a faster processor, since the speed is tied to the clock which in turn controls other functions in your computer). From a software standpoint, we are presently testing a new version of EDSMAC and EDCRASH that utilize the 8087 or 80287 numeric coprocessor. The current release of EDSMAC will not recognize the coprocessor and we anticipate a 100-200% increase in speed with the new software version. Once released, all you will need to do is install an 8087 (retail $90-180) for an IBM PC or PC/EX or an 80287 (retail $150-350) if you own an IBM AT and run the new software. Engineering Dynamics will be selling the 8087 upgrade version to present users for $75. We will formally announce its release after all testing is complete – around May 15.

A variety of add-in boards that boost the standard PC performance to the PC AT level have been announced. Some of these boards require removing the PC's central processor (8088 Chip), while others simply use an expansion slot. Also called Turbo-boards, they replace the central processing unit with a faster member of the Intel 8088 family–either an 8086, 80168 or 80286. Some also utilize a numeric coprocessor. The internal clock speed of the 8086 and 80286 is 10 MHz: and 80186 is 8 MHz. When compared to the 8088's 4.77 MHz clock speed, you can see the ultimate effect on processing speed. A coprocessor further reduces the average processing time by 100-300%

Although we have not had the opportunity to test these boards with EDVAP software, we have successfully tested EDVAP on IBM compatibles using a faster Intel chip. If you decide to buy one of these boards, we highly recommend you test EDVAP at the dealer <u>before</u> your purchase. We will post the test results for these boards in future newsletters as we receive them. Call us if you have any questions about a particular board. Two articles that review the various Turbo-boards, and 8087 chips are "Speed-up Boards", E. Grevstad, <u>Business Computer Systems</u>, Jan., 1986, P. 95-102 and the 8087/80287 Performance Curve", S. Fried, <u>Byte</u>, Fall, 1985-Inside the IMB PC, P 67-88.

An improvement in file input/output processing speed can be made if you use DOS 3.0 or greater and use a hard disk. During initialization, DOS 3.0 + executes a file called CONFIG.SYS to determine the parameters of devices attached to your PC (to "configure" your system). Since disk access is usually a slow process, you can minimize the number of times the disk is accessed by allocating some of your excess memory to buffer storage. This is done by using the command BUFFERS-20 in the CONFIG.SYS file. (See the IBM DOS 3.0 manual pages 4.8 thru 4.11). The exact buffer number bay be smaller if you have less than 512 K memory of use EDVAP in conjunction with other applications