4-2011

Three-Dimensional Security Barrier Impact Response Modeling

Ratul D. Sarmah

Christopher Y. Tuan
University of Nebraska Omaha, ctuan@unomaha.edu

ISSN 1749-3889 (print), 1749-3897 (online)
International Journal of Nonlinear Science
Vol.11(2011) No.2,pp.236-245

Three-Dimensional Security Barrier Impact Response Modeling

Ratul D. Sarmah[1] *, Christopher Y. Tuan[2]
[1] Senior Engineer, Schneider Structural Engineering, Inc., 5634 S. 85th Circle, Omaha, NE-68127, USA, ratul@cjseng.com, 1-402-592-1160
[2] Professor, Department of Civil Engineering, University of Nebraska-Lincoln, 1110 S. 67th Street, Omaha, NE-68182, USA, ctuan@mail.unomaha.edu, 1-402-554-3867
(Received 21 October 2010, accepted 20 March 2011)

Abstract: The occurrence of terrorist vehicular bomb attacks has been frequently reported. Perimeter security barriers are designed and built to defeat vehicle penetration. Due to the complexity of designing a security barrier, the current practice is to physically crash test the barrier head-on with a threat vehicle, at a certain velocity as a validation of the design. As full-scale vehicular crash tests are costly, facility designers are often forced to use existing barriers without the benefit of designing new, innovative and cost-effective systems for a specific threat. With the present-day computing capability, conducting three-dimensional simulations of vehicular crash dynamics on a personal computer has become very feasible. Vehicular crash simulations are traditionally conducted using finite element (FE) programs, which require extensive computer runtime. A first-principle approach based on the physics of crash dynamics has been developed to reduce computer runtime. Nonlinear structural responses due to the inelastic material effect and large deformation during the vehicular impact are taken into account. An explicit, step-by-step solution scheme is used for solving the transient dynamic problem. No inversions of large matrices are necessary at each time step in this algorithm. Validation examples using crash test data and data from the literature are presented to demonstrate the accuracy and efficiency of the developed algorithm.

Keywords: crash simulation; security barrier; nonlinear; colliding; crushing; separation

1 Introduction

This research involves the development of a three-dimensional barrier impact response model (BIRM3D) to address the physical security requirements of using perimeter security barriers to defeat vehicle penetration. Each government facility is subjected to unique threat due to its structural configuration, political environment and site conditions. Because of the high costs of conducting full-scale vehicular crash tests, facility designers are forced to use existing barriers without the benefit of designing new, innovative, and cost-effective systems for a specific threat. All the security barriers currently in service were proof-tested with a 5,000-pound pick-up truck for a head-on impact scenario at about 50 mph [4]. The effectiveness of these perimeter barriers to stop vehicle penetration under other impact conditions (i.e., different vehicles, impact angles and velocities) is uncertain. This limitation makes subsequent damage assessment difficult and unreliable.

With the latest advances in the CPU speed and computer animation, people expect a high degree of realism in computer simulation results. Existing full-scale vehicular crash tests conducted by federal and state agencies have been used to validate the models developed in this research. With a validated impact response model, a designer can quickly evaluate the vulnerability of a barrier and the associated vehicle trajectory for a variety of impact scenarios without conducting costly vehicular crash tests.

2 Description of BIRM3D

BIRM3D, written in C++ language, is a program for vehicular crash dynamics simulations. Five vehicle models: a 4000-lb. passenger car, a 5000-lb. minivan, a 7500-lb. pickup truck, a 15000-lb. cargo truck, and 72000-lb. 18-wheel

*Corresponding author. *E-mail address*: ctuan@mail.unomha.edu

IJNS.2011.04.15/469

semi-truck are resident in the database. Three types of barriers, rigid wall, bollards, and Jersey barrier, are available for simulation. The input and output parameters of the program are listed in Table 1. Vehicular crash simulations may be conducted for various impact velocities and impact angles, as defined graphically in Figure 1. User may select one vehicle model and one barrier model in each simulation. A user may edit the gross vehicle weight (GVW) of a vehicle, and the mass properties of the vehicle are automatically updated. User may also modify barrier sizes. Visual display of the crash dynamics simulation is accomplished by using OpenGL libraries. The runtime of a simulation on a PC with 1 GHz processor is typically under 10 minutes, as opposed to hours taken by conventional FE simulations. Simulation results are also stored in the form of time-histories for display. Validation examples using crash test data and data from the literature are presented to demonstrate the accuracy and efficiency of the program.

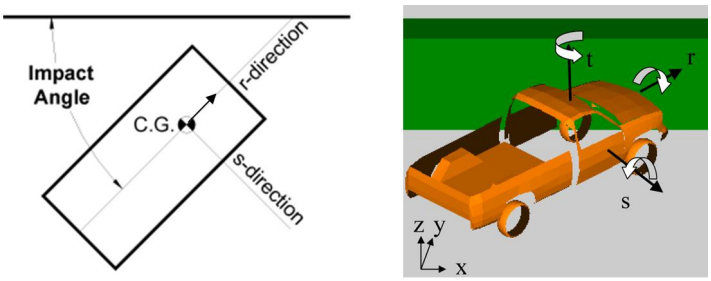

Figure 1: Impact velocity, impact angle, and r, s and t direction attached to the body

Table 1: Input and output of BIRM3D

Input	Output
Impact velocity	Trajectory of the center of mass of the vehicle
Impact angle	Roll, pitch and yaw angles about the r, s, and t axes of the vehicle CG.
Vehicle type	Time history of vehicle speed
GVW of vehicle	Angular velocity about r, s, and t axes of the vehicle
Barrier type	Normal impact force on the barrier
Pavement Friction Coefficient	Tangential impact force on the barrier
	Deceleration of the center of mass along r, s, and t axes of the vehicle
	Reactions under front and rear tires
	Force-deflection curves of truss members
	Crush of the vehicle body

3 Rigid Body Dynamics and Non-Penetration Constraints

The equations for unconstrained motion of a rigid body have been derived by Baraff [1] in terms of the kinematics and kinetics of the body. The kinematic parameters include position and orientation of the body, its linear velocity and angular velocity, mass of the body, its mass center, and the velocity of each particle of the body. The kinetic parameters include forces and torques acting on the body, linear momentum, angular momentum, and the inertia tensors. These parameters are used to describe the state of a rigid body.

A *colliding contact* is defined as where two bodies come in contact having a velocity v_{rel}^{-} towards each other. Colliding contact causes an instantaneous change in the velocity of a rigid body. When a collision occurs, the state of a body undergoes a discontinuity in the velocity. Numerical routines that solve ordinary differential equations commonly assume that the state of the body varies continuously and gradually. The resulting impulse j caused by colliding contact that prevents inter-penetration at time has been derived by Baraff [2] as

$$j = \frac{-(1+\xi)v_{rel}^{-}}{\frac{1}{M_a} + \frac{1}{M_b} + \vec{n}(t_0)\bullet[I_a^{-1}(t_0)(r_a \times \vec{n}(t_0)) \times r_a] + \vec{n}(t_0)\bullet[I_b^{-1}(t_0)(r_b \times \vec{n}(t_0)) \times r_b]} \tag{1}$$

where the quantity is the coefficient of restitution $(0 \leq \xi \leq 1)$, r_a and r_b are the displacements, M_a and M_b are the masses, $I_a(t_0)$ and $I_b(t_0)$ are the inertia tensors of bodies A and B, respectively, and $\vec{n}(t_o)$ is the normal vector of the colliding plane.

4 Vehicle Model Formulation

4.1 Stiffness Characteristics

Finite element models of common automobiles are available for download from the website of FHWA/NHTSA National Crash Analysis Center (NCAC) [7]. These models generally are for use in detailed crash dynamic simulations, and contain hundreds of thousands of nodes and elements. The downloaded models have been significantly modified to reduce the total number of nodes by at least an order of magnitude.

A vehicle is modeled as a 3D rigid body surrounded by deformable parts. The chassis, engine, gear-box, axles and wheel-hubs of the vehicle form the rigid body, whereas the steel frame and the sheet metal of the exterior surface form the deformable part of the model. The rigid and deformable vertices are inter-connected by inelastic truss members. From a modified NCAC model, 8-node solid elements with respective mass densities are used to model the rigid body of the vehicle. Mass properties are calculated from these solid elements. The mass densities of individual solid elements are adjusted to arrive at the GVW of the vehicle. Four-node shell elements, assumed to have no mass, are used to model the exterior surface. The stiffnesses of solid and shell elements are ignored in the vehicle model, which is different from a conventional FE simulation. Instead, each node of a shell element is connected to a node on the rigid body with a 2-node truss element. The selection of these nodes must be judiciously made such that the truss members during a collision would produce realistic forces and torques to yield the correct kinematic effects. Therefore, the selection of the truss nodes, the number of truss members, and the stiffness properties of the truss members are conducted through trial and error to achieve the most realistic results. This process is termed calibrating the model. Figure 2 shows a pickup truck model with solid and shell elements, and the inserted truss members between the solid and shell elements. When properly calibrated, the model will yield crush deformations compared favorably with those obtained from detailed FE crash simulations.

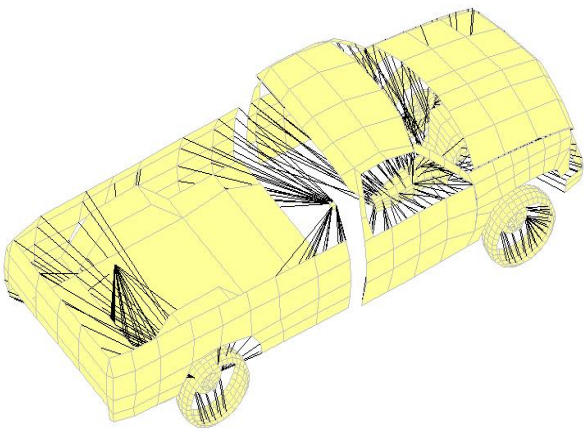

Figure 2: A Pick-up Truck model

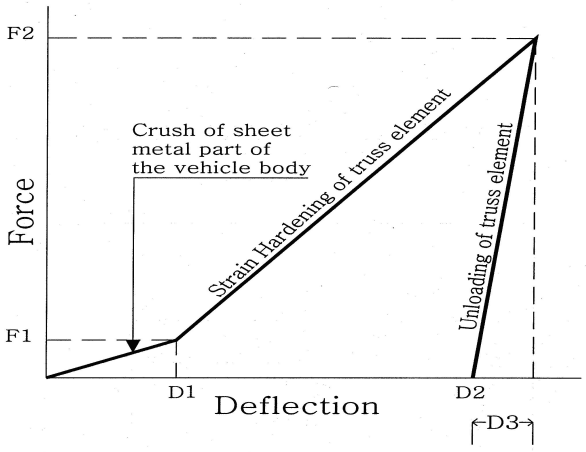

Figure 3: Force-deflection curve of a truss element

As shown in Figure 3, the force-deflection curve of a truss member at a contact node has an initial slope corresponding to initial crush stiffness of the vehicle body. This portion is followed by a steeper slope, which represents the strain hardening of the steel frame until the sheet metal is completely crushed onto the rigid body. The force in the member reaches its maximum value at this point. As the body begins to separate from the barrier, the force-deflection characteristic is governed by a third line, which corresponds to the unloading of the truss member. During this phase, the member will recover the elastic deformation, and the member force is calculated from the unloading curve. This process is repeated until a particular value of member length is reached to produce a zero member force.

Using these three force-deflection lines for each of the truss members in a vehicle model, the crash dynamics of a vehicle can be accurately simulated provided that:

a) Proper values of stiffness are used;

b) Proper nodes for the truss member with respect to the shell and solid nodes are chosen;
c) Nodes in the deformable and the rigid bodies are evenly distributed; and
d) Relatively small time-steps are used during contact with a barrier.

4.2 Mass Properties

Let a rigid body be made up of a large number of small particles indexed from 1 to N. The mass of the ith particle is denoted by mi, and each particle has a constant location ri in the body space. For a rigid body composed of 8-node solid elements with independent mass densities, the calculation of the total mass and the inertia effects of the body is not straightforward. The procedures of calculating the mass properties are described as follows. For an 8-node solid element consisting of 6 faces, the 4 nodes of a face must be on the same plane. This fact is used as the basis for calculation of mass properties. Most 3D objects are modeled with 8-node elements, and each of which may be broken into 6 tetrahedral elements as shown in Figure 4. The properties of a tetrahedral element can be calculated by first defining the four vertices of the tetrahedron [9] as $\vec{r_1} = [x_1, y_1, z_1]^T$, $\vec{r_2} = [x_2, y_2, z_2]^T$, $\vec{r_3} = [x_3, y_3, z_3]^T$ and $\vec{r_4} = [x_4, y_4, z_4]^T$. The center of gravity of the element can then be computed as

$$\vec{R_g} = \frac{1}{4}[\vec{r_1} + \vec{r_2} + \vec{r_3} + \vec{r_4}]^T \tag{2}$$

Its volume V, can be calculated as the determinant of the matrix as

$$V = \int dV = \frac{1}{6}\begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{bmatrix} \tag{3}$$

If a uniform mass density, ρ , is used, the mass of the tetrahedron is $m = \rho V$. Defining the position vectors with respect to the center of mass as $\vec{k_i} = (\vec{r_i} - \vec{R_g}), i = 1, 2, 3..and\ 4$ the elements inertia tensor about its center of gravity is

$$[I_g] = \frac{m}{20}\sum_{i=1}^{4}(\vec{k_i^T}k_i\vec{[1]} - \vec{k_i}\vec{k_i^T}) \tag{4}$$

$$[I_g] = \frac{m}{20}\sum_{i=1}^{4}\begin{bmatrix} k_{i2}^2 + k_{i3}^2 & -k_{i1}k_{i2} & -k_{i1}k_{i3} \\ -k_{i2}k_{i1} & k_{i1}^2 + k_{i3}^2 & -k_{i2}k_{i3} \\ -k_{i3}k_{i1} & -k_{i3}k_{i2} & k_{i1}^2 + k_{i2}^2 \end{bmatrix} \tag{5}$$

By parallel axis theorem, inertia tensor matrix, $[I_0]$, can be calculated as

$$[I_0] = [I_G] + [I_T] \tag{6}$$

where the translation matrix is expressed as

$$[I_g] = m(\vec{R_g^T}\vec{R_g[1]} - \vec{R_g}\vec{R_g^T}) \tag{7}$$

Since each of the 8-node elements in the rigid body may have different mass densities, the masses of the individual elements are integrated first to determine the total mass of the body, and subsequently used to calculate the center of mass of the rigid body. The inertia tensor of the rigid body is then calculated about its center of mass using Equation 5.

5 Simulation Algorithm

5.1 Collision Detection

The researchers at the University of North Carolina, Chapel Hills, has developed a C++ library, SWIFT++ [3], which can be used to effectively solve multiple collisions among a number of rigid bodies. However, it is only applicable to polyhedral models having non-convex body surfaces. Since vehicle models have convex surfaces, a more robust solution algorithm has been developed as is described as follows.

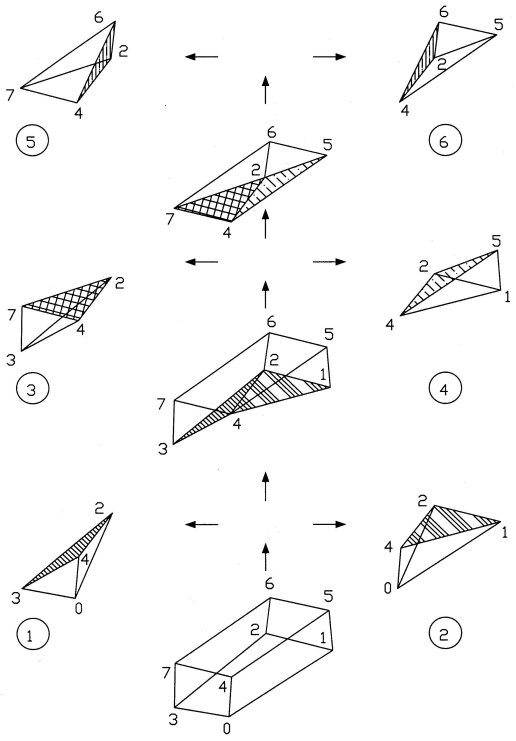

Figure 4: Break-up of an 8-node solid element into six tetrahedrons

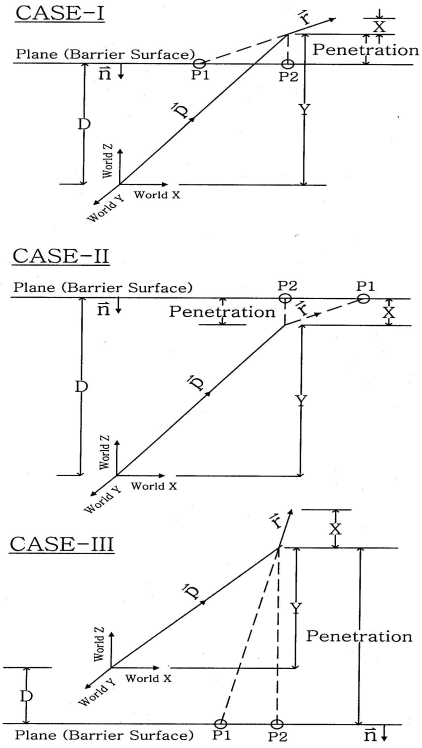

Figure 5: Collision between a vertex and a surface

There are four possible contact scenarios under which two bodies may collide with each other: *vertex to face*, *edge to edge*, *vertex to edge* and *vertex to vertex*. Physically, the vertex to edge and vertex to vertex contact modes are indeterminate. In the algorithm developed, only the vertex to face scenario with multiple point collisions is considered. Mathematically, a plane is represented by:

$$A.x + B.y + C.z - D = 0 \tag{8}$$

where *(A, B, C)* defines the normal vector of the plane, and *(x, y, z)* represents a point on the plane. D is a scalar, called the *plane-shift constant*, which specifies the distance of the plane from the origin of the world coordinate system. A shell element on a vehicle or a barrier surface consists of 4 nodes, and for calculation purpose, each four-node shell element is divided into two three-node shell elements. Two conditions must be met for a collision between a vertex and a surface to take place, as illustrated in Figure 5.

- The potential vehicle nodes velocity direction intersects with the barrier surface plane, and the point of intersection, P1, lies within that surface.
- The vector from the potential vehicle node along the barrier face normal intersects with the plane, and the point of intersection, P2, lies within that surface.

It can be shown that

$$D = \vec{a} \bullet (-\vec{n}) \tag{9}$$

$$Y = \vec{p} \bullet (-\vec{n}) \tag{10}$$

$$X = \vec{r} \bullet (-\vec{n}) \tag{11}$$

where $\vec{p}$ is a position vector of a vehicle node, $\vec{r}$ is a unit vector along a vehicle nodes velocity direction, $\vec{n}$ is a normal vector of the plane, $\vec{a}$ is a position vector of any coordinate of the barrier surface, D is the distance of the plane from the origin, and *penetration* is the perpendicular distance from the plane to the vehicle node. It can also be shown [5]that

$$penetration = D \pm Y \tag{12}$$

$$P1 = \vec{p} + (\frac{penetration}{X})\vec{r} \tag{13}$$

and

$$P1 = \vec{p} + penetration(\vec{n}) \tag{14}$$

The area of a triangle having vertices v_0, v_1 and v_2 is calculated [9] as:

$$A(\Delta) = \frac{1}{2}|(v_1 - v_0)\times(v_2 - v_1)| \tag{15}$$

When the point of intersection is connected to the three vertices of a 3-node shell element, the element is divided into three triangles. Therefore, if the sum of the areas of these three triangles is equal to the area of the 3-node element on the barrier surface, the point of intersection is verified to be on the barrier surface.

5.2 Contact Forces between Tires and Pavement

At the beginning of each time step, all forces acting on the vehicle body are initialized to zero. The acceleration due to gravity remains constant throughout the mass of the body without any torque effect. A resultant force equal to the vehicle weight is thus applied at the center of mass of the vehicle.

The ground or pavement is assumed to be a rigid surface. To simulate a vehicle moving on the pavement, impulses are calculated from the *colliding* vertices of the tires with the pavement and applied to the vehicles mass center to keep the vehicle on the pavement. The two front tires carry more weight than the rear tires. A value of ξ = 0.1 is used as the coefficient of restitution in the simulation. There are two types of frictional forces that act on each vehicle tire when a vehicle is in motion: rolling friction and skidding friction. Rolling friction is commonly ignored in the simulation due to its negligible magnitude. To determine skidding friction, the direction of the velocity vector of the vehicle mass center in the body co-ordinate system is first obtained. If the unit vector in the velocity direction is not parallel to the r-direction of the vehicle, skidding friction is mobilized and the direction of skidding frictional force is opposite to the unit vector. The impulse calculated at each of the colliding tire vertices is divided by the time-step size to get the reaction force at that vertex. At each time step, the reactions at all the colliding vertices are added to get the total reaction, which would be equal to the GVW of the vehicle. This normal force multiplied by the coefficient of friction between the tire and the pavement gives the total skidding frictional force. The frictional force vectors are applied to the vehicles center of mass. The corresponding torque vector from each individual force vector is calculated and added to the resultant torque vector at the center of mass of the vehicle.

5.3 Contact Forces between the Vehicle and the Barrier Surfaces

Once the vehicle surface collides with a barrier, calculation of contact forces based on rigid body dynamics is no longer valid. Since the vehicle body consists of many closely-spaced vertices, more than one vertex may collide with the barrier at an instant of time. Therefore, to realistically simulate a vehicular crash, it is necessary to identify multiple colliding vehicle vertices with barrier surface and to determine the contact forces at these vertices.

Each deformable vertex on the vehicle surface is connected to a truss member, which is characterized by a force-deflection curve as discussed earlier. Within a time step, potential colliding vehicle vertices on a barrier surface are determined, and contact forces are calculated based on the respective truss stiffness connected to the deformable vertices. These resulting force and torque vectors acting on the center of mass are used to determine the vehicles kinetic and kinematic parameters. These parameters are then used as the initial state for the next time step.

5.3.1 Crushing

If the relative velocity, v_{rel}, between two bodies is negative, then the bodies are moving towards each other. Penetration of a vehicle node through the barrier surface is considered to be negative. To determine deformation, the *crushing* status becomes true for all penetrating nodes having both negative v_{rel} and negative *penetration* values. Based on the penetration value, deflection along the corresponding truss element is calculated and a force along the truss element is determined using the force-deflection curve of that truss element. This force is applied to the center of mass of the vehicle. The resulting torque vector for each of such forces about the center of mass is calculated and added to the total torque acting at the center of mass. Since inter-penetration between bodies is not allowed, *penetrating* vertices are pushed back to the barrier surface along the surface normal. New locations of the vertices with respect to the vehicles center of mass are calculated. Contact forces are thus calculated based on the deformations as well as the force-deflection curves.

Referring to Figure 6, the variables involved in the calculation of deformation and contact forces are defined as, $\vec{n}$ = Outward normal of the barrier surface;

$\vec{D}$ = A vertex on the deformable surface;

$\vec{R}$ = A vertex of the rigid body connected to by a truss element;

$\vec{B} = \vec{D} - \vec{R}$ is the vector along the truss element;

$\vec{B_1}$ = Crushed position of a deformable vertex with respect to world x, y and z; $\vec{B_1} = \vec{B} + \vec{n}(Penetration)$;

$\vec{U}$ = Vector to a deformable vertex from the center of mass;

$\vec{U_1}$ = New position vector of deformable vertex from the center of mass $\vec{U_1} = \vec{U} + \vec{n}(Penetration)$

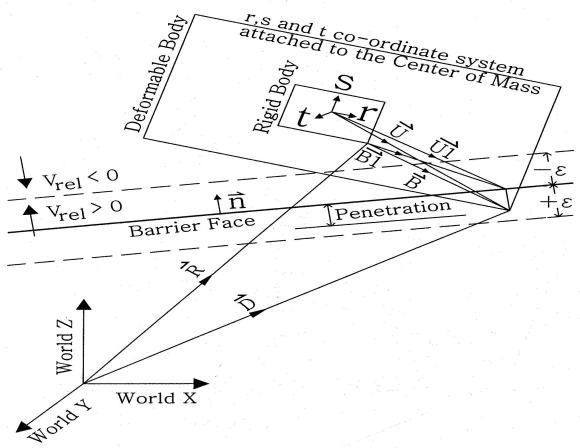

Figure 6: Calculation of deformation

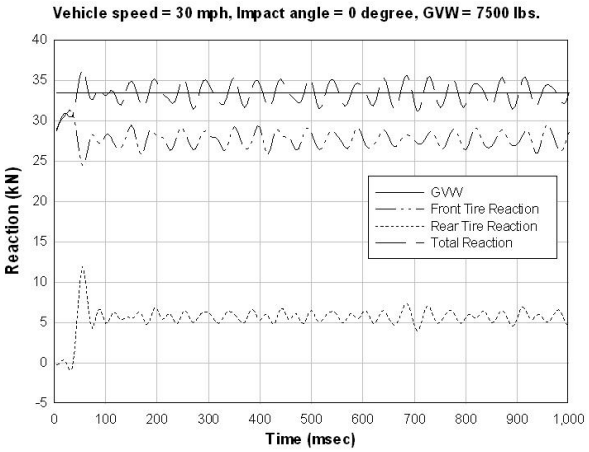

Figure 7: Comparison of reaction under tires and GVW

The new co-ordinates for a deformable vertex in the body space co-ordinate system are updated by $[R(t)]^{-1} \otimes U1$ at the end of each time step, and $R(t)$ is the orientation matrix which describes the orientation of the body in the world-space given by the equation:

$$[R(t)] = \begin{bmatrix} r_{xx} & r_{yx} & r_{zx} \\ r_{xy} & r_{yy} & r_{zy} \\ r_{xz} & r_{yz} & r_{zz} \end{bmatrix} \tag{16}$$

The first, second and the third column of $R(t)$ gives the directions of the x, y and z-axis of the vehicle in world space at time t, respectively.

5.3.2 Separation

The parameter ε (refer to Figure 6) was set at 5 mm in the collision detection algorithm. When the relative velocity of a vehicle vertex with respect to the barrier surface becomes positive, the vehicle vertex will begin to separate from the barrier surface. The *separation* status becomes true when separation is impending. For each time step thereafter, deformable vertices within the range of $+\varepsilon$ and 0 are identified, and the associated elements are pushed back to the barrier surface along the surface normal direction. Based on this deformation, forces are calculated corresponding to the third segment of the force-deflection curve, which signifies the *unloading* of the truss element. This process is repeated until a particular value of member length is reached to produce zero contact force.

5.4 Friction Forces between Vehicle and Barrier

Normal and tangential forces acting on a barrier during a vehicular crash is calculated from the amount of crushing along the truss elements and their respect stiffness properties as previously described. To incorporate the surface friction between the vehicle and the barrier, first the direction of the frictional force on the barrier surface is determined, which is opposite to the tangential force. The forces generated from each truss member during a time step are assumed to be equal to forces acting on the crushed vehicle surface. The component of each of these forces along the barrier surface normal is then calculated. Each of these normal forces is then multiplied by the frictional coefficient between steel and concrete surfaces, generally in the range of 0.15 to 0.20. The corresponding torque vectors are calculated from the frictional force vectors.

These frictional forces and torques are added to the total force and torque acting on the center of mass to determine the vehicle trajectory at the beginning of the next time step.

6 Validation Examples

6.1 Case 1: Total Tire Reaction vs. GVW

Test cases based on fundamentals of physics have been conducted to validate the algorithm developed. One such case is that the total reaction under the tires should be equal to the GVW of the vehicle.

Figure 7 shows a comparison of the total tire reaction against the GVW of the vehicle, for a vehicle traveling at 30 mph. These reactions were obtained by using Equation 1. Reactions under the tires were calculated by adding the impulses divided by the time step size for all the tire-contact points during each time step.

6.2 Case 2: Vehicle Deceleration vs. Normal Force on Barrier

The normal force exerted on a rigid barrier surface by a vehicle crashing onto it should equal to the mass of the vehicle times the deceleration at the mass center along the direction of normal force. Simulations of head-on collisions with impact velocities ranging from 20 mph to 60 mph were conducted, using the pickup truck model with GVW 7500 lbs against a rigid wall. The decelerations along r-direction of the vehicles center of mass are presented in Figure 8. The normal forces on the barrier are calculated from the amount of crush of the vehicle body, based on the stiffness properties of the truss elements connected to the deformable surface of the vehicle. The peak normal forces are compared against the theoretical impact forces in Table 2. It is seen that, for various impact velocities, the maximum normal force on the rigid wall compared closely with the mass times the maximum r-deceleration of the vehicle.

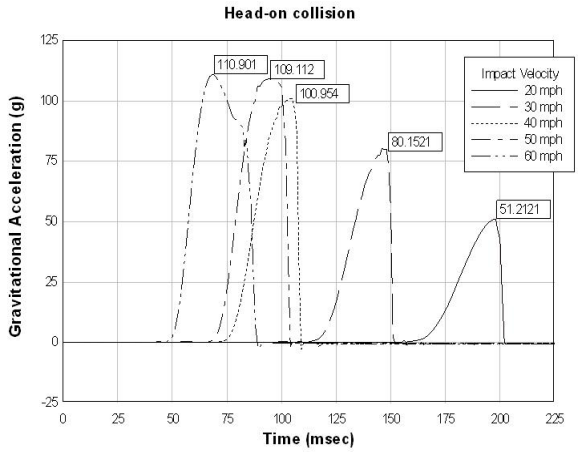

Figure 8: Deceleration of the center of mass along r-axis

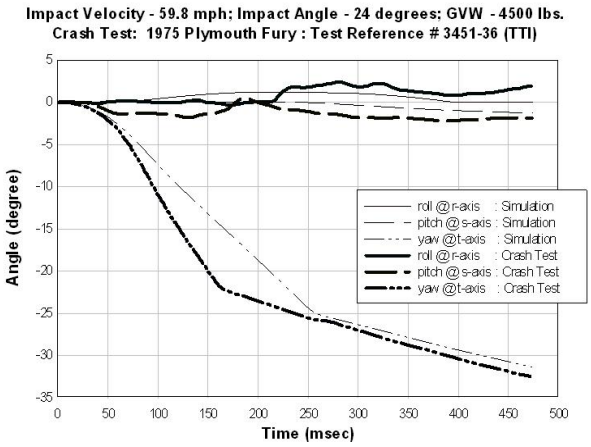

Figure 9: Roll, pitch, and yaw angles

Table 2: Comparison of theoretical and simulation values of normal force

Impact Velocity (mph)	Deceleration in-r-derection (g)	Mass of the Vehicle (kg)	Theoretical Impact Force (kN)	Simulation Normal Force (kN)
20	51.2121	3407.72	1712.01	1650.00
30	80.1521	3407.72	2679.46	2560.00
40	100.9540	3407.72	3374.87	3170.00
50	109.1120	3407.72	3647.59	3360.00
60	110.9010	3407.72	3707.39	3430.00

Table 3: Comparison of Simulation runtime

	BIRM3D	LS-DYNA	PAMCRASH
Platform	Desktop Pentium IV PC	Hewlett-Packard V2500 system	Silicon Graphics IBM SP2
Total number of finite element	6,000	336,000	60,000
CPU Speed	1 GHz	90 MHz	195 MHz
No. of Processor	1	8	126
Runtime	10 Minutes	31 hours [10]	27 minutes [11]

6.3 Case 3: Validation with Crash Test Data Passenger Car

In this validation, simulation results of roll, pitch, and yaw of a 4500-lb. passenger car colliding with a rigid wall at 59.8 mph and an impact angle of 24 degrees are compared to crash test data [6]. The passenger car used in crash test was a 1975 Plymouth Fury. The comparison results are presented in Figure 9.

Even though the passenger car model used in the simulation was different from the vehicle used in the actual crash test, Figure 10 shows that BIRM3D captured the salient characteristics of the roll, pitch and yaw angles obtained in the crash test.

6.4 Case 4: Validation with Crash Test Data Cargo Truck

The data two crash tests conducted by Texas Transportation Institute [8] were used to validate BIRM3D simulation results. A 15000-lb. diesel truck was used as the test vehicle. The barriers tested were a rigid wall and an array of bollards. In either case the test vehicle collided on the barrier head-on at 50 mph. The maximum pitch angles sustained by the truck obtained from the high speed film against the rigid wall and the array of bollards were 27 and 23 degrees, respectively. The pitch angles obtained from simulation were 25 and 22 degrees, as shown in Figure 10.

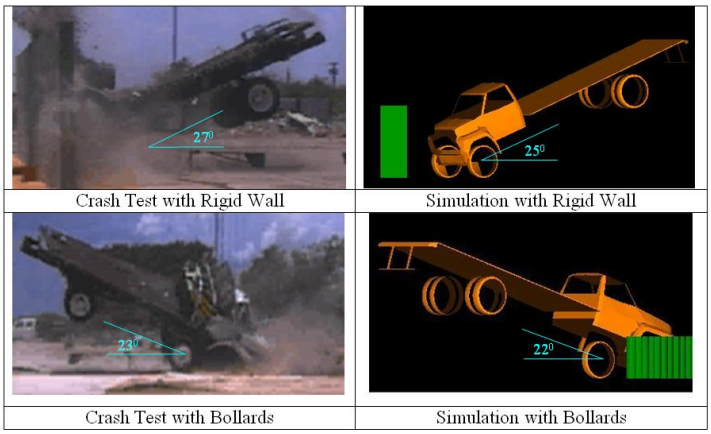

Figure 10: Comparison of Crash Test and Simulation of 15000 lbs. Truck

By proper calibration of the stiffness parameters of the truss elements in the vehicle model, accurate simulation results can be achieved. The above test cases proved that an algorithm based on simple physics coupled with calibrated stiffness characteristics of a vehicle body can allow accurate simulations without the disadvantage of long run-time, as in the case of a detailed FE simulation.

7 Conclusion

With the modern day computer technology, conducting three-dimensional simulations of vehicular crash dynamics on a notebook PC has become very feasible. A very efficient simulation algorithm has been developed in this research. Models

for simulation have been validated using fundamentals of physics and against full-scale crash test data. The simulation runtime is typically under 10 minutes as opposed to hours required by conventional FE simulations. A comparison of computing efficiency is given in Table 3. The test cases presented in this paper have proved that an algorithm based on the simple physics coupled with calibrated model parameters can allow accurate simulations without the disadvantage of long runtime.

Acknowledgments

The financial support provided by the Technical Support Working Group of the Combating Terrorism Technical Support Office, through contract No. N41756-02-C-4681, for the development of BIRM3D is gratefully acknowledged.

References

[1] D. Baraff. Physically Based Modeling Rigid Body Simulation I, Unconstrained Rigid Body Dynamics. SIGGRAPH Course Notes, G1-G31. 2001.

[2] D. Baraff. Physically Based Modeling Rigid Body Simulation II, Non-penetration Constraints. SIGGRAPH Course Notes, 2001, G32-G35, G40-G49, G55-G61. 2001.

[3] S. A. Ehmann. Speedy Walking via Improved Feature Testing for Non-convex Object (SWIFT++). Version 1.0, SWIFT++ Library and application Manual, University of North Carolina, Chapel Hills. 2002.

[4] K. L. Hancock and J. D. Michie. Development of a Vehicular Barrier Testing Program. Final Report, SWRI Project No.06-8443, Southwest Research Institute, San Antonio, TX. 1985.

[5] K. Hawkins and D. Astle. OpenGL Game Programming. Textbook. Prima Publishing. (2001):658-654. .

[6] H. S. Perera. Development of an Improved Highway-Vehicle-Object-Simulation Model for Multi-Faced Rigid Barrier. Test reference 3451-36. Transportation Research Record 1233, Texas Transportation Institute. (1989):105-116.

[7] Finite Element Model Archive. FHWA/NHTSA National Crash Analysis Center. George Washington University.

[8] Crash Test Report. Crash Testing and Evaluation of Rigid wall and Bollard. Test 400091-ARB3 (10 Rigid Wall) and Test 400091-ARB4 (Bollard), Texas Transportation Institute.

[9] M. d. Berg, M. v. Krefeld, M. Overmars and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications, Second Edition.* New York. 2000.

[10] C. Kan and Y. Lin. Evaluation of Performance, Reliability, and Consistency of MPP Version of LS-DYNA. 6th International LS-DYNA Conference. Detroit, 2000.

[11] M. Bubak, H. Afsarmanesh, R. Williams and B. Hertzberger. High Performance Computing and Networking. 8th International Conference, HPCN 200. Amsterdam, The Netherlands. May 2000.