

Automated Methods for Converting a Non Real-time Cartesian Multi-body Vehicle Dynamics Model to a Real-time Recursive Model

Huron S. Perera
Richard Romano
Realtime Technologies, Inc.

Patrick Nunez
US Army RDECOM-TARDEC

Copyright © 2005 SAE International

ABSTRACT

The United States Army Research, Development, and Engineering Command's Tank Automotive Research, Development and Engineering Center (U.S. Army RDECOM-TARDEC) laboratories is seeking to advance modeling and simulation methods used for predicting the performance of ground vehicles. TARDEC typically generates non-real-time models of its vehicles using DADS [1]: a general purpose commercial, multi-body software package based on a Cartesian coordinate formulation. TARDEC also currently uses SimCreator [2], [3] to develop real-time multi-body vehicle models. SimCreator uses recursive techniques to perform the simulations in real-time. The goal of the study presented here was to develop rapid conversion methods for translating models of DADS and other commercial multi-body software packages into SimCreator models. A procedure that can be automated was developed to convert a DADS model of a High-Mobility Multipurpose Wheeled Vehicle (HMMVW) to a SimCreator model. The vehicle model consisted of the chassis base body and the 4 wheels connected to it with double A-Arm suspensions and steering constraints. This model with 2 closed kinematic loops at each suspension was preprocessed to determine the best joints to be used as constraints (cut joints). The Cartesian joint coordinates of the DADS models were converted to relative joint coordinates used by SimCreator. Bodies, joints, and force elements were then converted using the components from SimCreator's multi-body dynamics library. In place of the radial spring tire model used in the DADS simulation, a multi-disc tire model was developed to be used with SimCreator. The vehicle running over 8-inch, 6-inch, and 4-inch half-round obstacles at 10 mph was then simulated using the SimCreator model and the results were compared with field test results and similar simulations of the DADS model. Review of the results showed that the

SimCreator results closely resembled the test data and was typically more accurate than the DADS results. Since, the multi-body effects of SimCreator and DADS should be generating the same results, it is believed that the differences between the results are due to the difference in the tire models.

INTRODUCTION

The modeling and simulation methods used for predicting the performance of off-road and on-road vehicles at the United States Army Research, Development, and Engineering Command's Tank Automotive Research, Development and Engineering Center (U.S. Army RDECOM-TARDEC) laboratories can be divided into three categories based on the amount of CPU time usage. They are: offline in which the CPU usage is exceeding the limits required to be met for a real-time interactive simulation, real-time simulation, and faster than real-time simulation in which a large number of such vehicles need to be involved in a simulation experiment. Most of the non-realtime models of the vehicles at TARDEC are typically generated using DADS [1]: a general purpose commercial, multi-body software package based on a Cartesian coordinate formulation. For real-time multi-body vehicle models TARDEC currently uses SimCreator and SimCreator's multi-body dynamics component library [2], [3].

SIMCREATOR

SimCreator is a commercial product developed and marketed by Realtime Technologies, Inc. (RTI) that provides a graphical hierarchical control system simulation and modeling environment similar to EASY5 [4] and Simulink [5]. SimCreator has been used in the past to build real-time vehicle dynamics, and motion control algorithms for military, commercial and

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE 01 JUN 2006	2. REPORT TYPE N/A	3. DATES COVERED DEC 2004 - JUN 2005	
4. TITLE AND SUBTITLE Automated Methods for Converting a Non Real-time Cartesian Multi-body Vehicle Dynamics Model to a Real-time Recursive Model		5a. CONTRACT NUMBER W56 HZV-05-C-0107	5b. GRANT NUMBER
6. AUTHOR(S) Huron S. Perera Richard Romano Realtime Technologies, Inc. Patrick Nunez US Army RDECOM-TARDEC		5c. PROGRAM ELEMENT NUMBER	
		5d. PROJECT NUMBER	
		5e. TASK NUMBER	
6. AUTHOR(S) Huron S. Perera Richard Romano Realtime Technologies, Inc. Patrick Nunez US Army RDECOM-TARDEC		5f. WORK UNIT NUMBER	
		8. PERFORMING ORGANIZATION REPORT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Realtime Technologies, Inc 1517 N. Main Street Royal Oak, MI 48067		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) USA TACOM 6501 E. 11 Mile Road Warren, Michigan 48397-5000		10. SPONSOR/MONITOR'S ACRONYM(S) REDCOM TARDEC	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) USA TACOM 6501 E. 11 Mile Road Warren, Michigan 48397-5000		11. SPONSOR/MONITOR'S REPORT NUMBER(S) 15421	
		12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release, distribution unlimited	
13. SUPPLEMENTARY NOTES Joint work; US Government purpose license Submitted for Society of Automotive Engineers (SAE)World Congress, April 2006			
14. ABSTRACT Technical paper describing methods of converting a non-real-time vehicle dynamics model into a real-time model. The main topic of the paper is the conversion from a cartesian description of the model to a recursive description.			
15. SUBJECT TERMS Simulation, Vehicle dynamics, real-time			
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT SAR
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified	18. NUMBER OF PAGES 10
			19a. NAME OF RESPONSIBLE PERSON

entertainment applications as well as complete driving simulators. SimCreator's multi-body dynamics component library is based on Composite Rigid Body Methods (CRBM) developed by Walker and Orin [6], [7]. The CRBM method is used for open kinematic chains. To handle closed kinematic chains, constraint equations with corresponding Lagrange multipliers are introduced and are used to augment the mass matrix. For each constraint equation, a second-order dynamic system is also introduced that minimizes position and velocity errors during the simulation according to methods developed by Baumgarte [8]. SimCreator's set of multi-body vehicle dynamics components are shown in Figure 1. In this case dark and light red defines the sides that can be connected together.

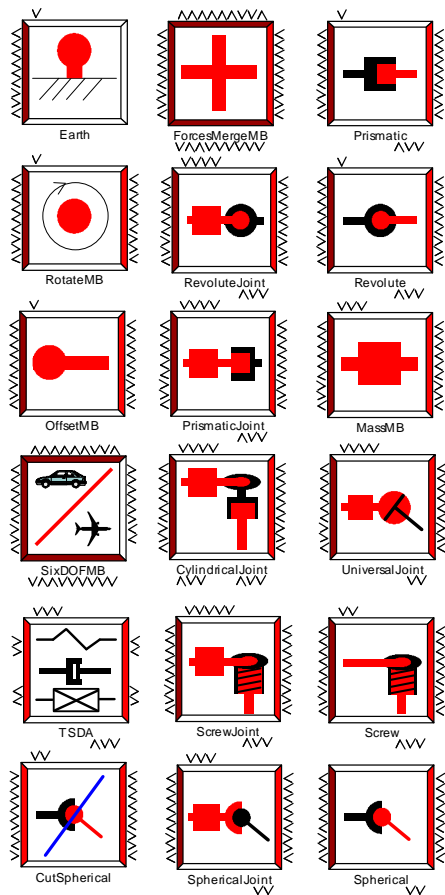


Figure 1 SimCreator Multi-Body Dynamics Components

CONVERSION OF A CARTESIAN MULTI-BODY DYNAMICS MODEL TO A RECURSIVE MODEL

DADS, a general purpose commercial, multi-body software package developed by LMS, is a Cartesian multi-body modeling approach. Several steps are required to translate a Cartesian model into a recursive model, needed for real-time simulation in SimCreator. These steps include:

1. Determining weighting factors for each joint
2. Determining the best joints to cut
3. Determining the optimal base body

Once the cut joints and the base body have been determined, the recursive spanning tree can be identified. User intervention during the translation process will allow modification of cut joints and base bodies selected by the automated system. Typically graph theory is used to represent the mechanical system and identify the topological structure. Each body is defined as a node and kinematic joints are defined as an edge. A graph is connected if all nodes can be reached from an arbitrary node along some path. A closed loop is a loop in which the beginning node and the ending node are the same. If there are no closed loops in the system graph, the system is said to have a tree structure. If a graph is not a tree, an edge (joint) is cut in each closed loop to form a tree structure. Weighting factors on joints and chains along the graph can be used to minimize the length of chains as well as the order of the mass matrix and associated constraint equations. Tsai and Haug [9] developed an automated method for translating Cartesian vehicle models into an open tree structure with cut joints.

During Phase I of the study, a HMMWV model was converted from DADS into SimCreator by hand. Notes were taken on the method and approach required to generate the model. The DADS .DEF file which stores the DADS elements, model hierarchy and transformation data for the HMMWV was obtained from the US Army Corp of Engineers. This model was selected because of the availability of test data to compare the results with. The model and test data are described in Jones [10] and given in Figure 2. The

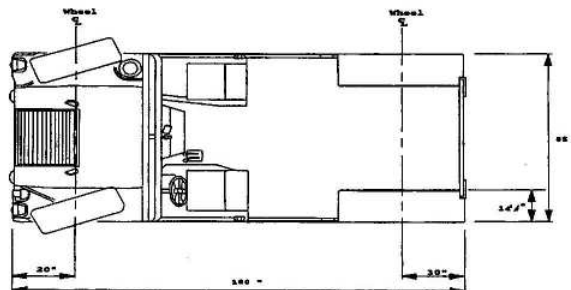
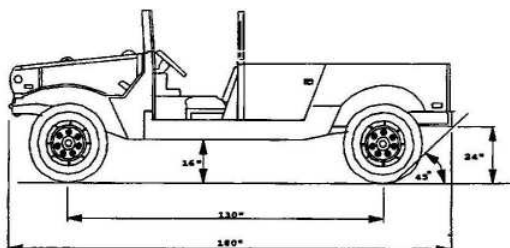


Figure 2 HMMWV Utility Truck

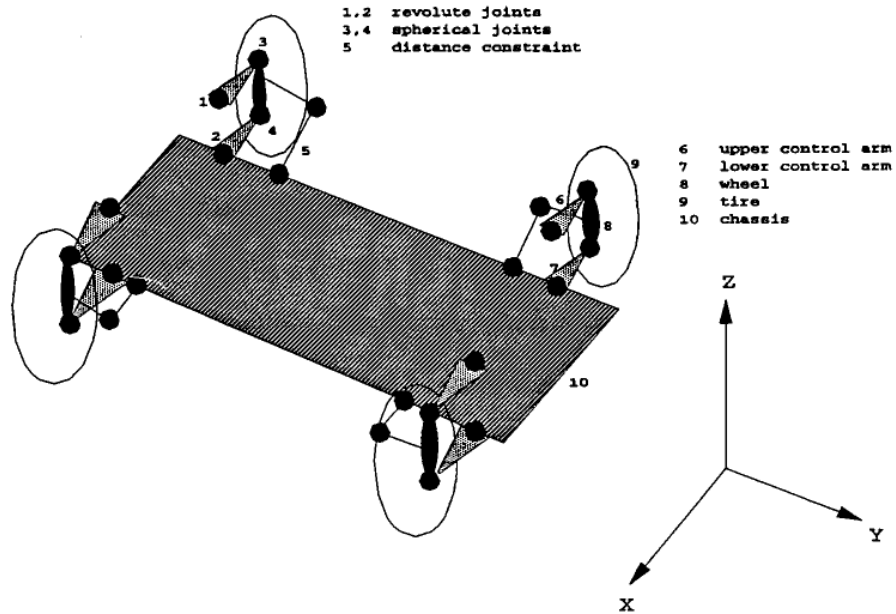


Figure 3 Jones [10] DADS HMMWV Model

equivalent DADS model is given in Figure 3.

The procedure proposed by Tsai and Haug [9] was used to preprocess the DADS model and convert it to a SimCreator model. For this study, the procedure was done without developing any code to automate it. The total number of joints to be cut is determined using the formula:

$$N_{cj} = N_j - (N_b - 1) \quad (1)$$

Where

- N_{cj} = number of cut joints
- N_j = number of joints in the system
- N_b = number of bodies in the system

Therefore for the HMMVW model,

- Number of Bodies = 13
- Number of Joints = 20
- Number of required Cut Joints = $20 - (13 - 1) = 8$

In order to find which joints to cut, all possible combinations of spanning trees that can be formed must be identified. In general the solution must calculate all spanning trees available. However, the process was simplified for the hand calculations by observing that 2 joints must be cut in each corner. Therefore a single corner will be presented here. The process presented can be extended to support the entire vehicle in an automated fashion. Which 2 joints to be cut are calculated using weighting factors. A graph representing the left front corner of the vehicle and its connection to the chassis is shown in Figure 4. There are 6 possible combinations of spanning trees that are generated by selecting different joints to cut. The graphs representing

them are shown in Figure 5. Since these spanning trees only include the non-cut joints, the configuration with minimum total weight is the most efficient.

Using the weighting factors for each joint provided by Tsai and Haug [9] and reproduced in Table 1, the weight for each configuration can be computed as follows.

Total Weights for Spanning Trees:

- Spanning Tree #1: $R + R + D = 1.1 + 1.1 + 6.0 = 8.2$
- Spanning Tree #2: $R + R + S = 1.1 + 1.1 + 3.0 = 5.2$
- Spanning Tree #3: $R + R + S = 1.1 + 1.1 + 3.0 = 5.2$
- Spanning Tree #4: $R + S + D = 1.1 + 3.0 + 6.0 = 10.1$
- Spanning Tree #5: $R + S + D = 1.1 + 3.0 + 6.0 = 10.1$
- Spanning Tree #6: $R + S + S = 1.1 + 3.0 + 3.0 = 7.1$

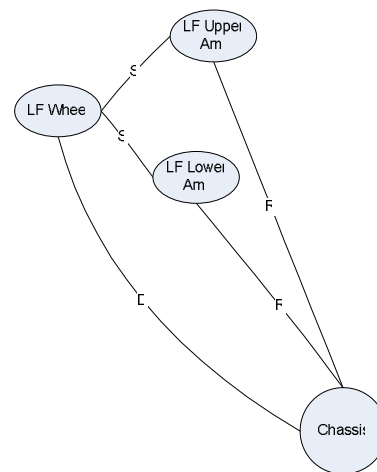


Figure 4 Left Front DADS Model

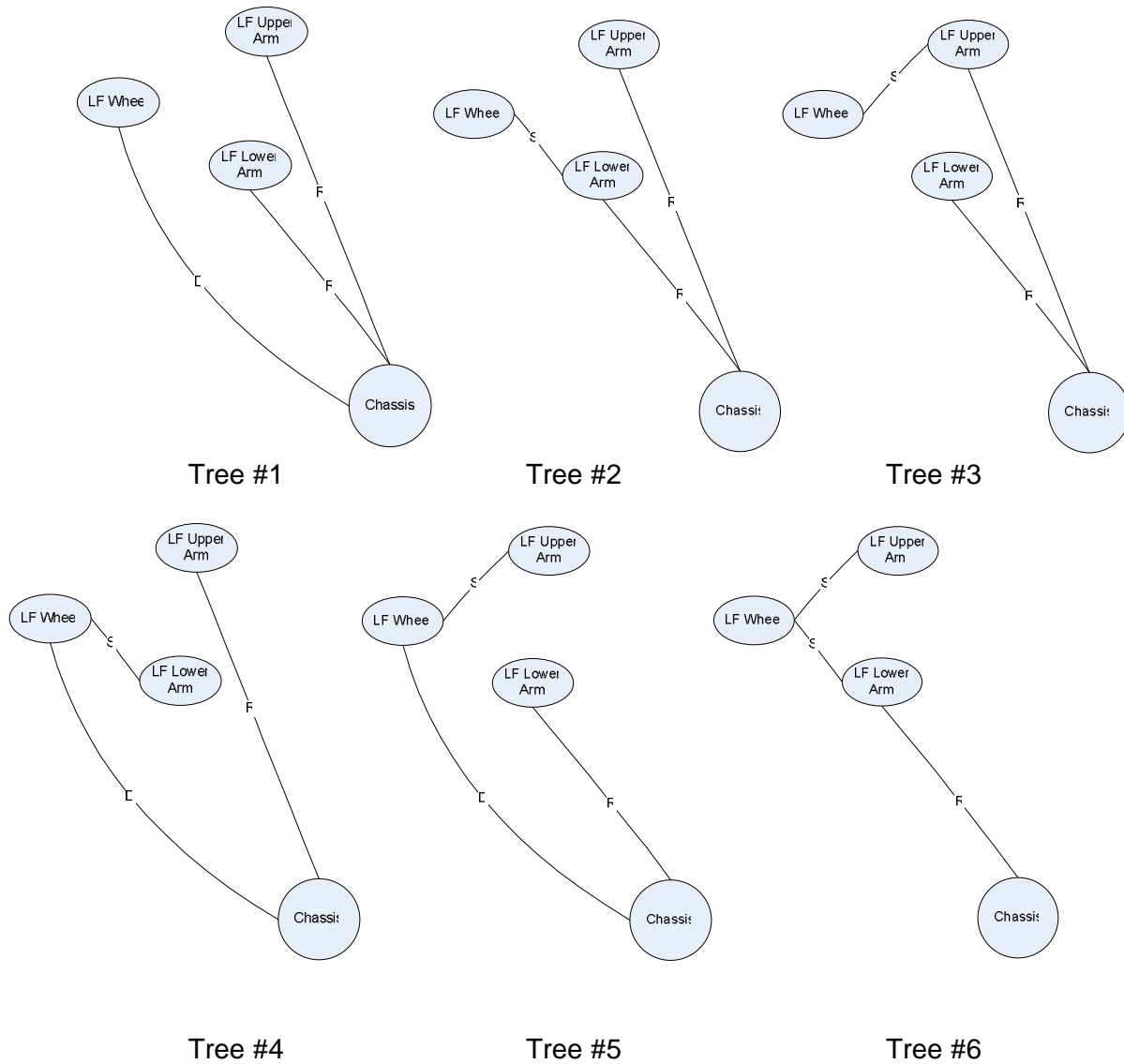


Figure 5 Possible Spanning Trees

Table 1 Weighting Factors for Joints

Joint type	Revolute Joint	Translational Joint	Spherical Joint	Distance Constraint
Relative coordinates reduced	1	1	3	6
Cut joint constraints introduced	5	5	3	1
Dimension of coeff. matrix increased	4	4	0	-5
Arithmetical operations	more	less	no comparison	no comparison
Weighting factors	1.1	1.0	3.0	6.0

Note: The decimal in Revolute Joint above account for the arithmetical operations. A cylindrical joint is modeled as a revolute joint combined with a translational joint and therefore has a weighting factor of 2.1. A universal joint is modeled as two revolute joints and therefore has a weighting factor of 2.2.

The weight computations above show that Tree #2 or #3 above is the optimal configuration (minimum weight) minimizing the computation time in simulations using the model. Hence, spanning tree #3 which cuts the spherical joint between the upper arm and the wheel and the distance constraint was used to create the SimCreator model.

COORDINATE SYSTEM CONVERSIONS

Initial review of the DADS model identified two immediate issues. The coordinate systems in DADS and SimCreator are different. The coordinates need to be changed from x-right, y-forward and z-up to the SAE convention of x-forward, y-right and z-down. Also DADS allows multiple units conventions to be used. Therefore the ANALYSIS CREATE SYSTEM.DATA in the DEF file where a DADS model indicates the units being used needs to be read and the appropriate conversion to Metric calculated if needed.

The joint coordinate system (CS) orientations of the DADS models are usually different from that of SimCreator models. Furthermore, the joint orientation in DADS is usually expressed relative to each body connected by the joint. Due to use of the recursive formulation, a hierarchal joint representation is used in SimCreator and therefore one body is the parent and the other the child. Hence, the joint transformation matrices need to be converted as follows. The known transformations are,

T_{DP} = DADS joint CS transformation matrix relative to parent body

T_{DC} = DADS joint CS transformation matrix relative to child body

T_{SP} = SimCreator joint CS transformation matrix relative to parent body

Given these transformation matrices the unknown transformation matrix can be found as,

$$T_{CS} = [T_{DC}]^{-1} [T_{DP}] [T_{SP}]^{-1} \quad (2)$$

where

T_{CS} = Child body transformation matrix relative to SimCreator joint CS

In the DADS model that was used in the current investigation the joint CS orientations were defined using 3 points P, Q and R where P is a point defining the origin, the point Q forms the PQ vector defining the positive z-axis, and R is a point not collinear with PQ which defines the PR vector in the x-z plane. Then the rows of transformation matrix can be expressed as,

$$\begin{aligned} T_{0j} &= [[(PQ \times PR) \times PQ] & 0] \\ T_{1j} &= [[PQ \times PR] & 0] \\ T_{2j} &= [[PQ] & 0] \\ T_{3j} &= [x_p \quad y_p \quad z_p \quad 1] \end{aligned} \quad (3)$$

TIRE MODEL

The DADS tire model described by Jones [10] was a radial spring tire model. However the tire model described in the DADS user manual is a thin disc with vertical slices. Therefore the tire model to implement was not clear. With the current DADS representation, a contact patch close to the sides of the tire will come out different from that close to the bottom of the tire. After discussions with TARDEC personnel, a better representation of the tire using a set of discs as described by Blundell and Harty [11] was selected. In this model intersection of the discs as a whole with the terrain geometry is used to find the contact patch and the forces at the contact patch. This multi-disc tire model was implemented in SimCreator for the purpose of the current study.

The algorithm developed carries out initial iterations to identify road elements that are subject to potential contact, at the current integration time step, before evaluating each tire disc or slice with each of the candidate road elements. For each of the discrete elements used to model the tire cross-section the intersection with the road surface elements produces a line projection of the intersection of the tire element. From this it is possible to compute the area and hence volume related to the penetration of tire cross-sectional element by the road. Hence, for a tire with n cross-sectional elements, where each component has m components of penetrated area, the effective penetrated volume, V_{eff} , the x coordinate of the effective contact point of the tire, X_{ecp} , the x component of the effective road normal, X_{ern} , and the effective coefficient of friction, μ_{eff} are given by the [11]:

$$V_{eff} = \sum_{j=1}^n \sum_{i=1}^m A_i w_j \quad (4)$$

$$X_{ecp} = \frac{1}{V_{eff}} \sum_{j=1}^n \sum_{i=1}^m A_i w_j X_i \quad (5)$$

$$X_{ern} = \frac{1}{V_{eff}} \sum_{j=1}^n \sum_{i=1}^m A_i w_j n_{xi} \quad (6)$$

$$\mu_{eff} = \frac{1}{V_{eff}} \sum_{j=1}^n \sum_{i=1}^m A_i w_j \mu_i \quad (7)$$

where

A_i = the penetrated area of the i th component of area within the cross sectional tire element,

w_j = the width of the j th cross-sectional element of the tire,

X_i = the coordinate of the contact point for the i th component of area within the cross-sectional tire element,

n_{xi} = the x component of the road normal for the i th component of area within the cross-sectional tire element, and

μ_i = coefficient of friction associated with the i th component of area within the cross-sectional tire element.

It should be noted that y and z coordinates of the effective contact point and the y and z components of the effective road normal can be found similar to X_{ecp} and X_{erm} in equations 5 and 6 above.

The computation of the normal force acting on the tire by the road involves an intermediate step where the effective volume of penetration, V_{eff} , is related to radial penetration of the tire. For this purpose, an interpolation is done on a look-up table held within the tire model that relates (for the defined tire profile) the tire penetration volume to radial penetration when the tire is compressed onto a flat surface. The normal force vs. radial penetration curve for the tire then gives the required normal force.

COMPARISON OF SIMULATION RESULTS AND CONCLUSIONS

The SimCreator model as implemented is shown in Figure 6. The left front corner is shown in Figure 7. The SimCreator model was run over a 4", 6", and 8" half round obstacle at 10 MPH. The results for vehicle pitch rate, and acceleration at the driver seat were compared with the DADS results as well as test data. The results are shown in Figure 8 through Figure 10. The test data is the dotted line, the DADS data is the solid line, while the SimCreator data is the thick grey line. The DADS results and test data is taken from Jones [10]. It should be noted that since that study was conducted, the DADS commercial package has gone through several improvements and therefore the version of DADS that was used in Jones [10] was not available to authors.

Reviewing the results, the SimCreator data closely resembles the test data. It is typically closer to the test data than the DADS results. Therefore for vertical ride evaluation the conversion process is considered a success. Since the multi-body effects of SimCreator and DADS should be generating the same results, it is expected that the differences between the results are due to the difference in tire models. The interaction between the half-round and radial spring tire model is of discrete nature compared to the continuous interaction

of the multi-disk tire and the half-round. This continuous ground interaction of the multi-disk tire model implemented in SimCreator appears to be generating better forces than the radial spring model implemented in DADS.

A timing analysis of the HMMWV model in SimCreator revealed that one model update took 0.6 msec on a 3.2 GHz Pentium IV using a Runge-Kutta 2nd order integrator. Therefore it is expected that the model can easily run in real time at up to 800 Hz. It should also be noted that 26% of each update was spent on the multi-disc tire model computations.

REFERENCES

1. Haug, E. J., "Computer Aided Kinematics and Dynamics of Mechanical Systems", Allyn and Beacon, Needham Heights, MA, 1989.
2. Romano, R., "Realtime Driving Simulation Using a Modular Modeling Methodology", SAE Technical Paper Series No. 2000-01-1297, March 2000.
3. Romano, R. "Real-Time Multi-Body Vehicle Dynamics Using a Modular Modeling Methodology", SAE Technical Paper Series No. 2003-01-1286, March, 2003.
4. Ciesla, C. R., Jennings, M. J., "A Modular Approach to Powertrain Modeling and Shift Quality Analysis," SAE Technical Paper Series No. 950419, February, 1995.
5. Zachary, J. R., Munns, S. A., and Moskwa, J. J., "The Development of Vehicular Powertrain System Modeling Methodologies: Philosophy and Implementation", SAE Technical Paper Series No. 971089, February, 1997.
6. Walker, M. W. and Orin, D. E., "Efficient Dynamic Computer Simulation of Robotic Mechanisms," ASME Journal of Dynamic Systems, Measurement and Control, Vol. 104, 1982, pp. 205-211.
7. Romano, R., "Real-time Multi-Body Vehicle Dynamics Using A Modular Modeling Methodology", SAE Technical Paper Series No. 2003-01-1286, March 2003.
8. Baumgarte, J., "A New Method of Stabilization for Holonomic Constraints," ASME Journal of Applied Mechanics, 50:869-870, 1983.
9. Tsai, F. F. and Haug, E. J., "Automated Methods for High Speed Simulation of Multibody Dynamic Systems", Technical Report R-47, Center for Simulation and Design Optimization of Mechanical Systems, University of Iowa, Iowa City, Iowa, 1989.
10. Jones, R. A., "Validation Study of Two Rigid Body Dynamic Computer Models", Department of the Army, Waterways Experiment Station, Corps of Engineers, Technical Report GL-92-17, September 1992.
11. Blundell, M. and Harty, D., "The Multibody Systems Approach to Vehicle Dynamics", SAE International, SAE, Warrendale, PA, 2004.

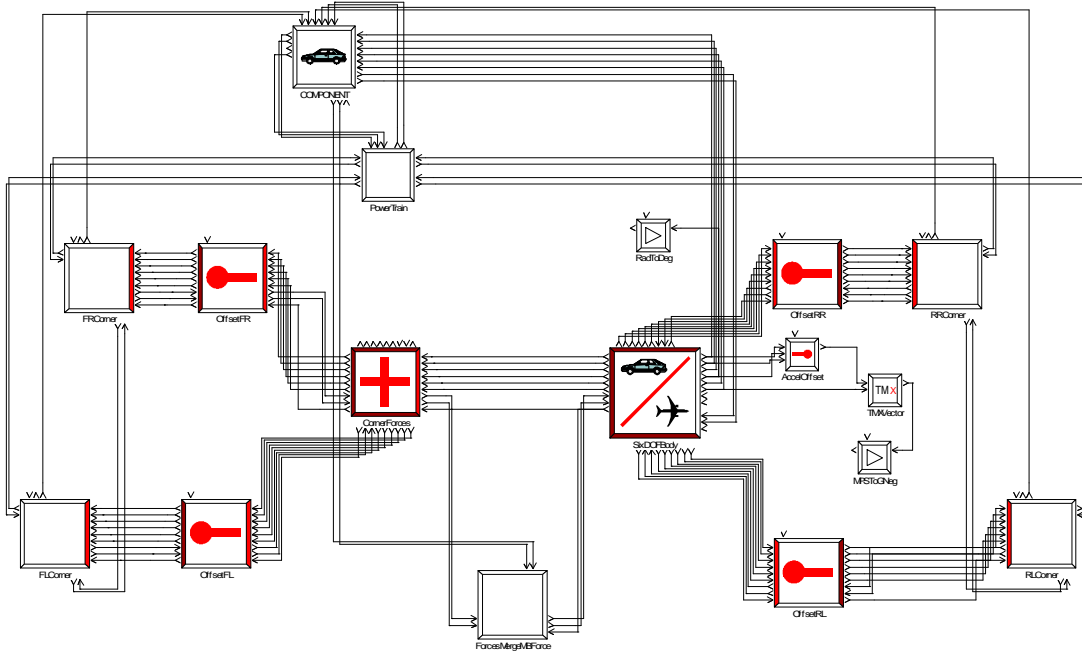


Figure 6 SimCreator HMMWV Model

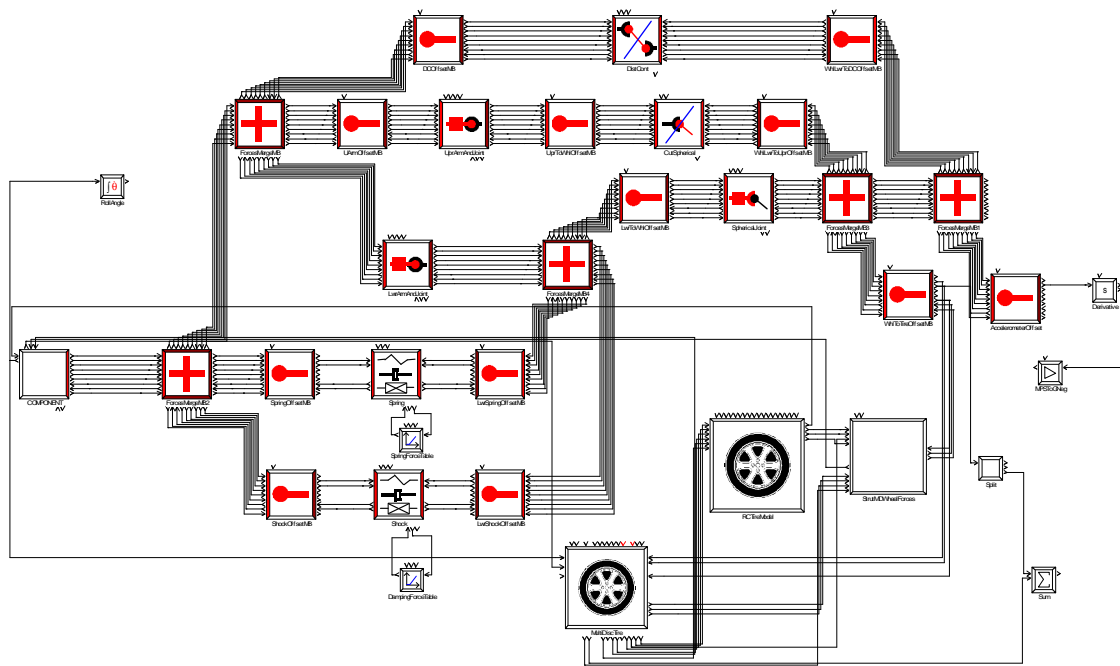


Figure 7 SimCreator Front Left Corner Model

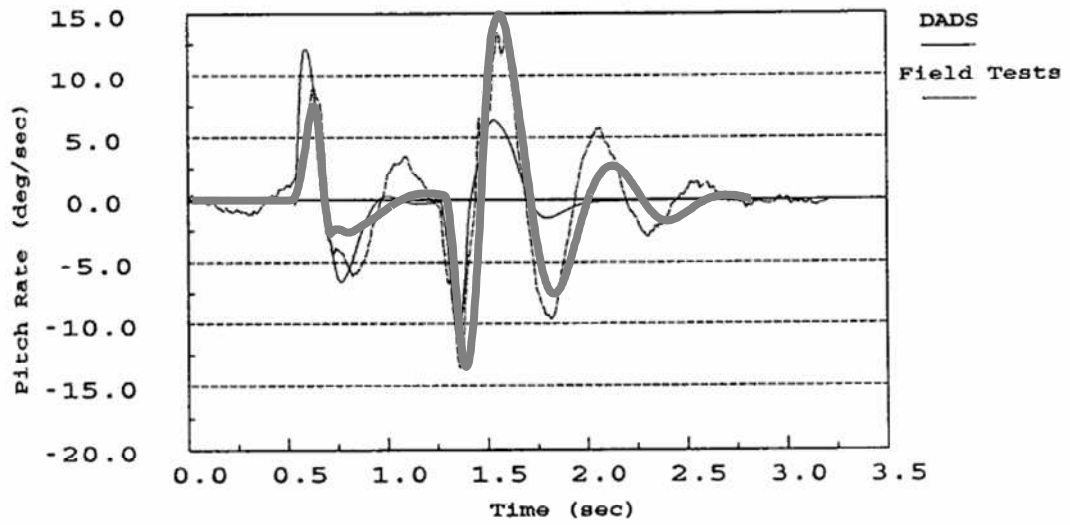
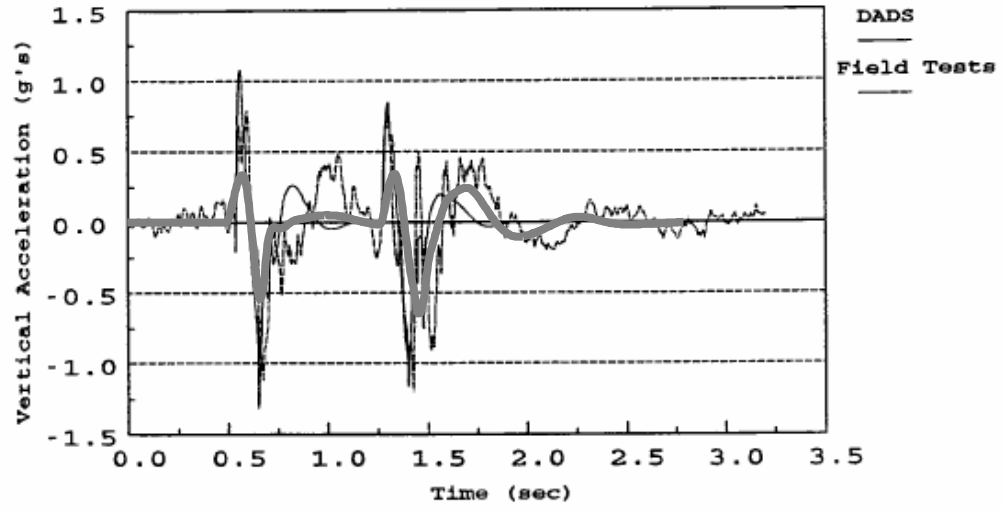


Figure 8 Vertical Accel and Pitch Rate as vehicle passed over a 4-inch half-round obstacle at 10 mph

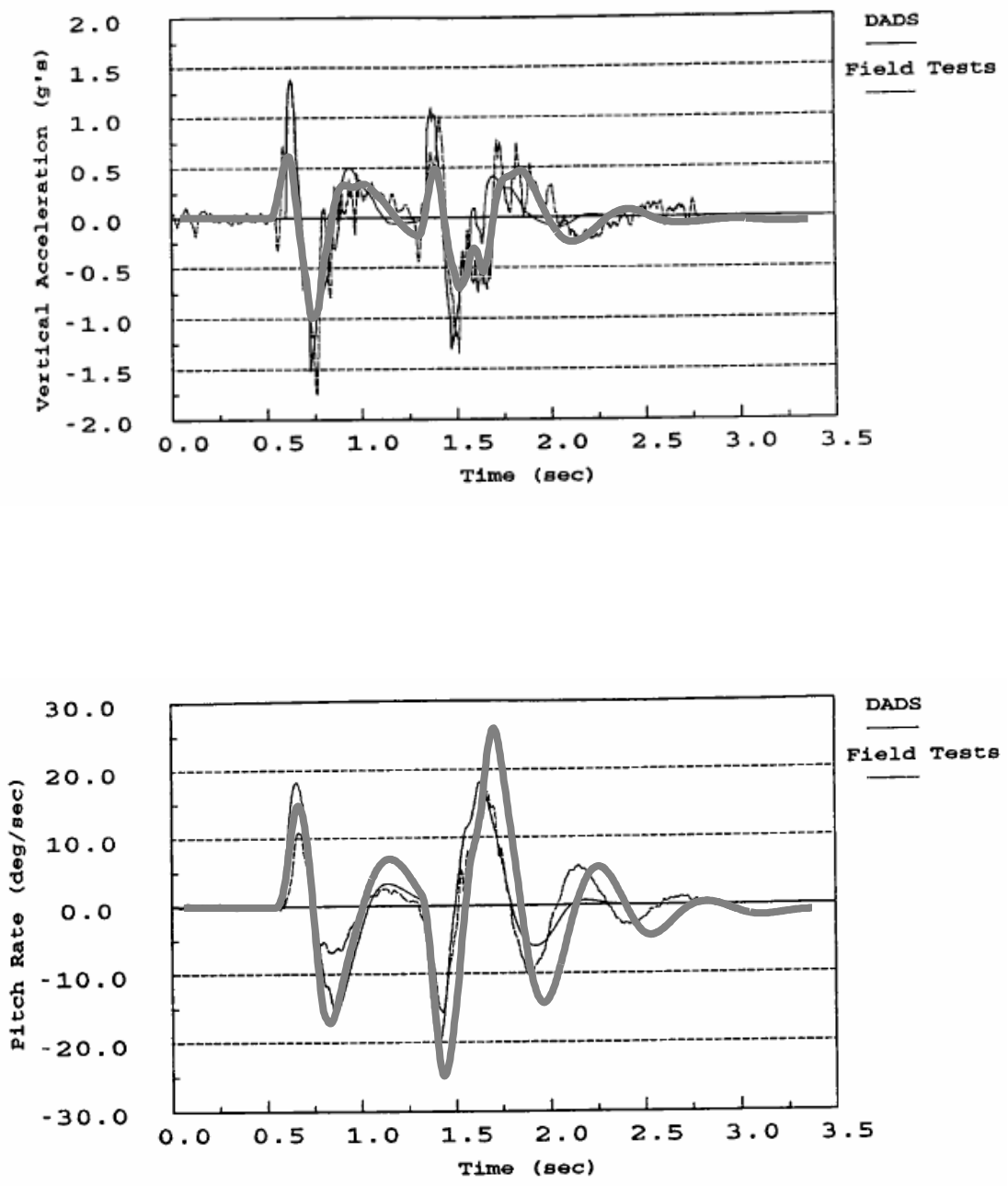


Figure 9 Vertical Accel and Pitch Rate as vehicle passed over a 6-inch half-round obstacle at 10 mph

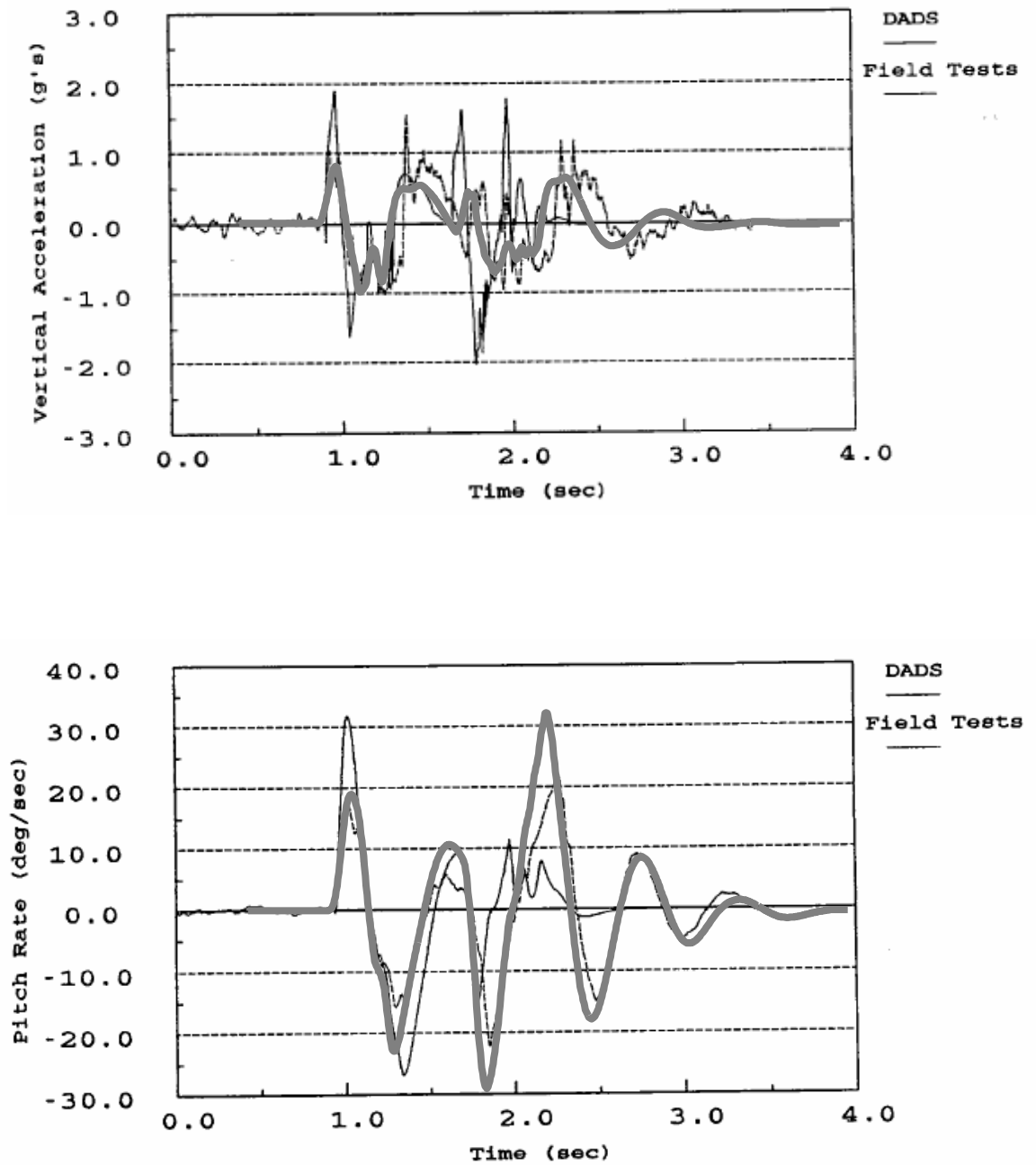


Figure 10 Vertical Accel and Pitch Rate as vehicle passed over a 8-inch half-round obstacle at 10 mph

