UNCLASSIFIED: Dist A. Approved for public release
2010 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
MODELING & SIMULATION, TESTING AND VALIDATION (MSTV) MINI-SYMPOSIUM
AUGUST 17-19 DEARBORN, MICHIGAN

VEHICLE DYNAMIC MODELING & SIMULATION: COMPARING A FINITE-ELEMENT SOLUTION TO A MULTI-BODY DYNAMIC SOLUTION

Paramsothy Jayakumar, PhD
Dynamics & Structures Modeling & Simulation,
US Army RDECOM-TARDEC
Warren, MI

Tamer Wasfy, PhD
Advanced Science and Automation Corp.
Indianapolis, IN

ABSTRACT

The dynamic response of two multibody systems, a planar mechanism and a spatial robot, are generated using an explicit time integration finite element code and a multi-body dynamics code. Comparisons are made of the dynamic solution including body motion, joint constraint forces, conservation of energy, and CPU time. While finite-element simulation offers accurate modeling of structural flexibility, multibody dynamic simulation demonstrates the capability to produce accurate and efficient results.

INTRODUCTION

The Army routinely performs multibody dynamic simulations of its ground vehicles, both tracked and wheeled, in order to assess their mobility performance and improve durability. Traditionally these simulations are performed using rigid multibody dynamics software. However, to improve the accuracy of modeling flexible structures and the desire to have an integrated simulation environment, finite element software have lately been considered for performing the same simulations. There are advantages and disadvantages to such an approach. This paper addresses the solution accuracy of this approach by comparing solutions of two benchmark problems using an explicit finite element approach to the multi-body dynamic simulation.

EXPLICIT FINITE ELEMENT (FE) CODE FORMULATION

The translational equations of motion are written with respect to the global inertial reference frame and are obtained by assembling the element equations. The finite elements use only translational degrees-of-freedom (DOF) and no rotational DOF. The translational DOF includes the rigid-body translation and the finite element nodal positions. The equations of motion can be written as:

$$M_K \ddot{x}^t_{Ki} = F_{s}{}^t_{Ki} + F_{a}{}^t_{Ki} \qquad (1)$$

where t is the running time, K is the global node number (no summation over K; K=1→N where N is the total number of nodes), i is the coordinate number (i=1,2,3), a superposed dot indicates a time derivative, M_K is the lumped mass of node K, x is the vector of nodal Cartesian coordinates with respect to the global inertial reference frame, F_s is the vector of internal structural forces, and F_a is the vector of externally applied forces which include surface forces and body forces.

Each rigid body is represented by a body-fixed material frame whose origin is located at the body's center of mass. The mass of the body is concentrated at this center-of-mass node, and the inertia of the body, given by the inertia tensor I_{ij}, is defined with respect to the body frame. The orientation of the body-frame is given by $R_K^{t_o}$ which is the rotation matrix relative to the global inertial frame at time t_0. The rotational equations of motions are written for each rigid body with respect to its body-fixed material frame as:

$$I_{Kij}\ddot{\theta}^t_{Kj} = Ts_{Ki}^t + Ta_{Ki}^t - \left(\dot{\theta}^t_{Ki} \times (I_{Kij}\dot{\theta}^t_{Kj})\right)_{Ki} \qquad (2)$$

where I_K is the inertia tensor of rigid body K, $\ddot{\theta}_{Kj}$ and $\dot{\theta}_{Kj}$ are the angular acceleration and velocity vector components for rigid body K relative to it's material frame in direction j, Ts_{Ki} is the component of the vector of internal torque at node K in direction i, and Ta_{Ki} is the component of the vector of applied torque at node K in direction i. The Einstein

summation convention is used only for the lower case indices *i* and *j*.

Constraint equations are algebraic equations, which describe

- Prescribed motion constraints:
$$f(\{x\},t)=0 \quad (3)$$
- Joint constraints:
$$f(\{x\})=0 \quad (4)$$
- Contact constraints:
$$f(\{x\})\geq 0 \quad (5)$$

The penalty technique is used for imposing the joint and contact constraints. In that technique a normal reaction force is generated between two points $\vec{x}_{p1}$ and $\vec{x}_{p2}$ in order to satisfy the joint or contact constraint. The constraint penalty force is given by:

$$F_{c_j}=(k_p\left|\vec{d}\right|+c_p\,\hat{d}_i\dot{d}_i)\hat{d}_j \quad (6)$$

$$d_i=x_{p1_i}^t-x_{p2_i}^t \quad (7a)$$

$$\dot{d}_i=\dot{x}_{p1_i}^t-\dot{x}_{p2_i}^t \quad (7b)$$

$$\hat{d}_i=d_i/\left|\vec{d}\right| \quad (7c)$$

where k_p is the penalty stiffness and c_p is penalty damping. The magnitude of the penalty force is given by Equation (6) for joint constraints. For contact constraints the penalty force is given by Equation (6) if point 1 is inside the volume of the body that point 2 belongs to, otherwise it is zero. An asperity-spring friction model is used to represent joint and contact friction in which friction is modeled using a piece-wise linear velocity-dependent approximate Coulomb friction.

An explicit solution procedure such as the Newmark integration formula is used to solve the equations of motion (1, 2) along with the constraint equations. In explicit solution techniques the time step (Δt) used must be less than a critical time step in order to obtain a stable solution. The critical time step is less than half of the smallest characteristic wave propagation time in an element. This procedure is implemented in a commercial software [1] that is used in the present paper.

MULTIBODY DYNAMIC (MBD) CODE FORMULATION

A commercial software described in [2] is the multibody dynamics code used in this paper. In it, the equations of motion for a rigid multibody system are combined with the constraint acceleration equations and written in the following form:

$$\begin{bmatrix} M & \dfrac{\partial\Phi^T}{\partial q} \\ \dfrac{\partial\Phi}{\partial q} & 0 \end{bmatrix}\begin{Bmatrix}\ddot{q}\\ \lambda\end{Bmatrix}=\begin{Bmatrix}Q\\ \gamma\end{Bmatrix} \quad (8)$$

where q is the vector of generalized coordinates of the system, M is the generalized mass matrix, Q is the vector of applied forces, Φ are the constraint equations, λ is the vector of Lagrange multipliers of the constraints, and γ is the right hand side vector of the constraint acceleration equations.

In addition, the constraint equations and constraint velocity equations are,

$$\Phi(q,t)=0 \quad (9)$$

$$\frac{\partial\Phi}{\partial q}\dot{q}=\nu \quad (10)$$

respectively.

Equations of motion (8) along with the kinematic constraint equations (9) and the constraint velocity equations (10) yield a mixed system of differential-algebraic equations of motion for the multibody system. These equations can be solved using implicit integration methods such as the Backward Differentiation Formula (BDF). For stiff systems, the superior stability characteristics of the BDF methods allow them to take much larger stepsizes than would be possible with explicit methods.

BENCHMARK PROBLEMS

Two problems that were used as benchmark problems in *Multibody Systems Handbook* [3] are solved using the explicit finite element software and the multibody dynamics software. A comparison of the simulation results is presented below.

Benchmark Problem 1: 7-Link Planar Mechanism

The 2-D mechanism shown in Figure 1 was used as one of the multibody dynamics benchmarks in *Multibody Systems Handbook* [3]. A description of the mechanism and a list of parameters such as the mass, moments of inertia, link lengths, and initial link positions are also given in [3]. The mechanism consists of 7 rigid links connected by frictionless revolute joints. The mechanism has one linear spring that connects point *D* on link K_3 and point *C* on the ground. In the initial position, the spring is under compression. The mechanism is driven by a motor torque that is applied to link K_1 and given by:

$$T_{K1} = \begin{cases} 0.033 \text{ N.m} & 0 \le t \le 0.1 \\ 0 & t > 0.1 \end{cases}$$

The drive torque is removed at time 0.1 sec in order to assess the energy conservation capability of the simulation code. The simulation is run for a total time of 0.5 sec. The 2-D mechanism has 1 degree-of-freedom.

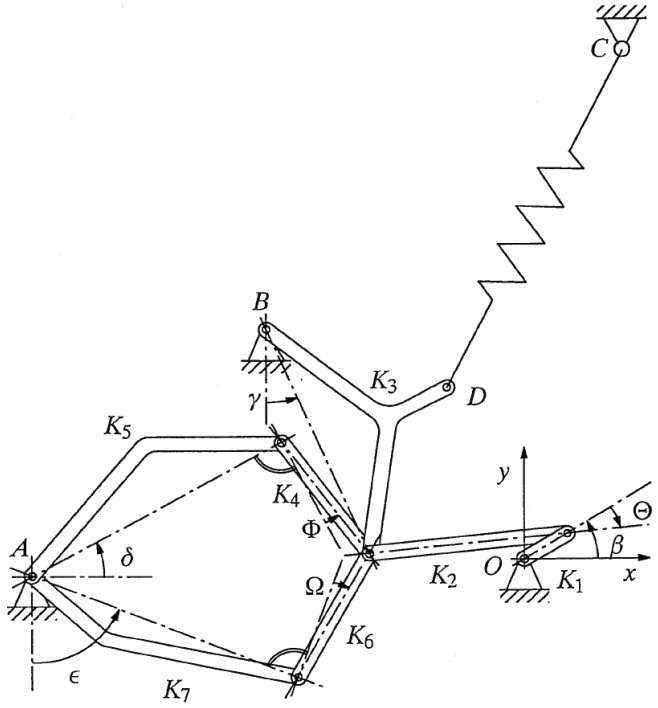


Figure 1: 7-link planar mechanism [3].

Table 1 shows the multibody dynamic simulation CPU time, finite element simulation CPU time, and the corresponding time steps used. Two runs with two different time steps (Δt = 0.4E-5, 0.1E-5) are used for FE, while the MBD time step chosen was 1.0E-5. It will be shown later (Figure 7) that for comparable accuracy between the two codes, FE requires the finer time step of 0.1E-5 s. Figure 2 shows snapshots of the first revolution of the mechanism simulated using FE. Table 2 shows a comparison of MBD and FE results against Handbook [3] results using the Kane's Method for the rotation angle of link K_1. Both MBD and FE results match well with Handbook results. Figures 3 and 4 show the time-histories of the rotation angles and angular velocities of links K_1, K_3, and K_5 generated using FE and MBD. Figure 5 shows the time-history of the spring force generated using FE and MBD. The spring force is a function of the link positions. Comparing these results, the rotation angles, angular velocities, and spring force, one can conclude that both FE and MBD predict the same link motion.

Table 1: Comparison of MBD and FE time steps and CPU times for the 7-link planar mechanism for a total simulation time of 0.5 sec. Runs were performed on a DELL OPTIPLEX 760 Intel Core 2 Duo 3.16GHz.

	MBD	**FE**	**FE**
Time step (s)	1.0E-5	0.4E-5	0.1E-5
CPU time (s)	20.875	4.955	19.820
Position/velocity error Tolerance	1E-3	-	-

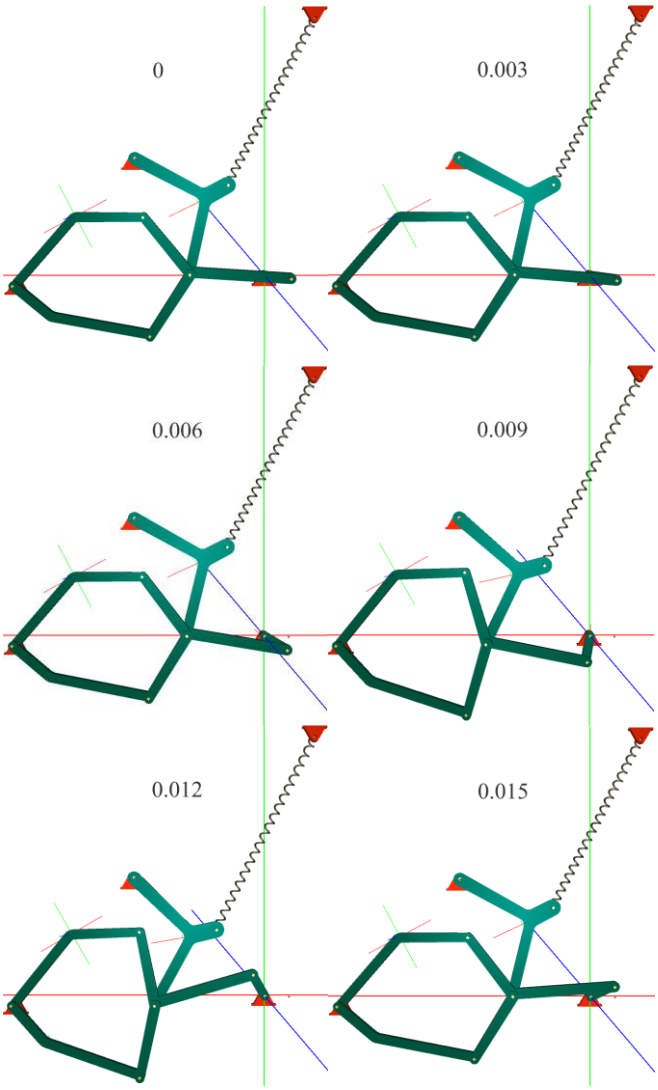


Figure 2: Snapshots of the motion of the 7-link planar mechanism simulated using FE (Δt = 0.4E-5).

Table 2: Comparison of MBD and FE (Δt = 0.4E-5) against Handbook [3] results for the rotation angle of link K1 of the 7-link planar mechanism.

Time (s)	Beta (rad)			% Difference between Handbook &	
	Handbook [3]	MBD	FE	MBD	FE
0.000E+00	-6.1994E-02	-6.2017E-02	-6.19953E-02	0.04	0.00
5.000E-03	2.10881E-01	2.06715E-01	2.10215E-01	-1.98	-0.32
1.000E-02	2.16040E+00	2.14057E+00	2.15981E+00	-0.92	-0.03
1.500E-02	5.65556E+00	5.64598E+00	5.65841E+00	-0.17	0.05

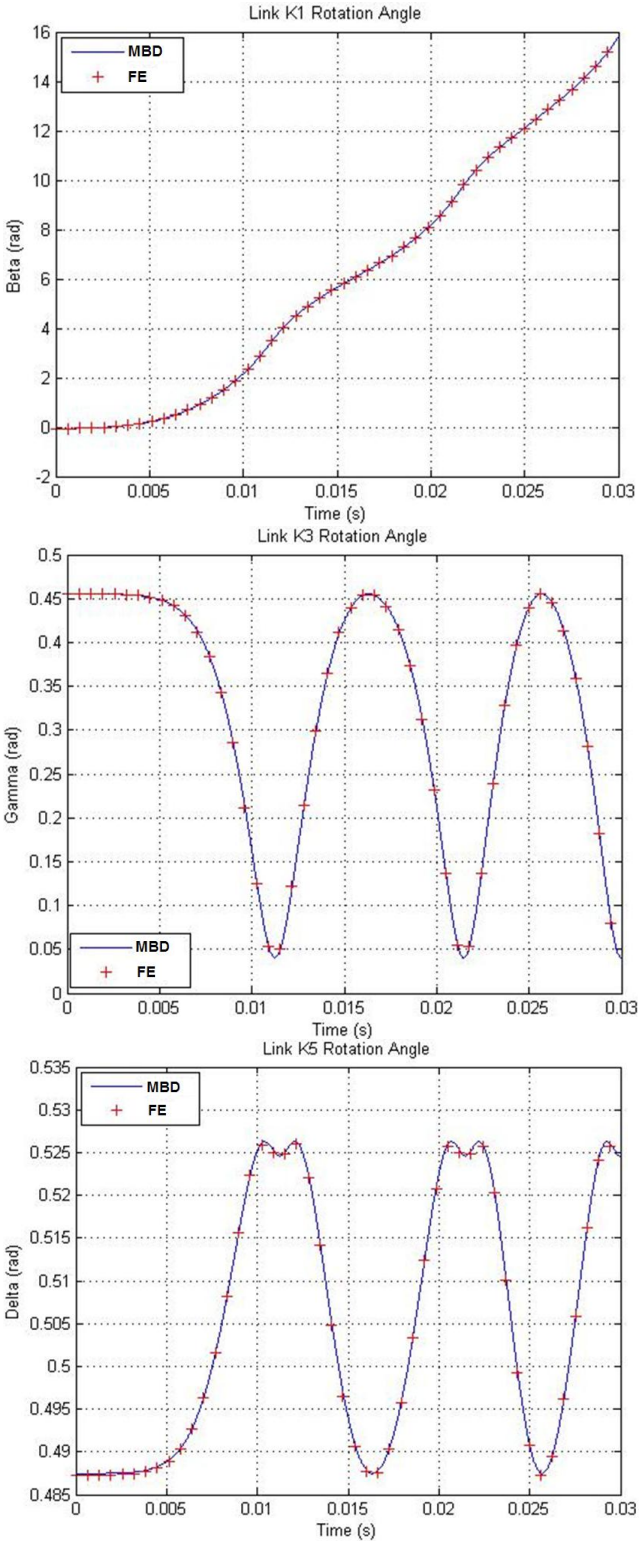


Figure 3: Time-histories of the rotation angles of links K_1, K_3, and K_5 generated using FE (Δt = 0.4E-5) and MBD.

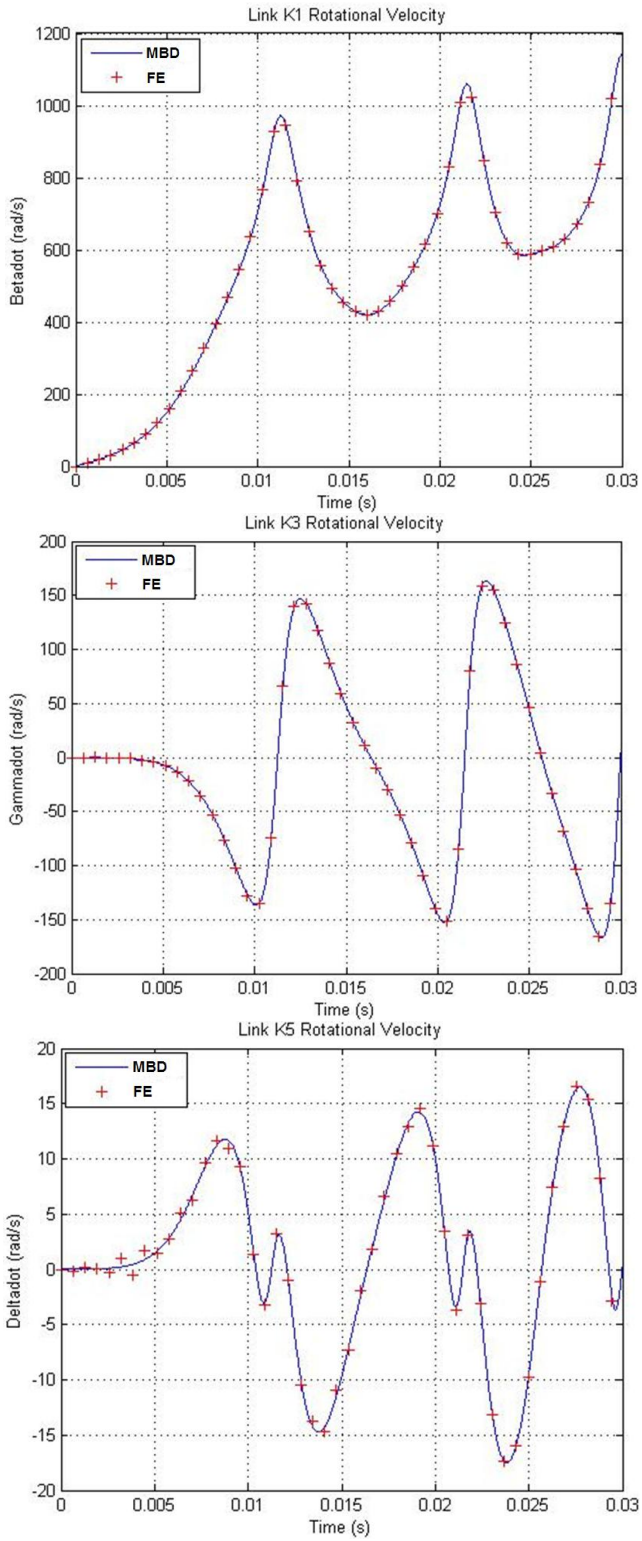


Figure 4: Time-histories of the angular velocities of links K_1, K_3, and K_5 generated using FE (Δt = 0.4E-5) and MBD.

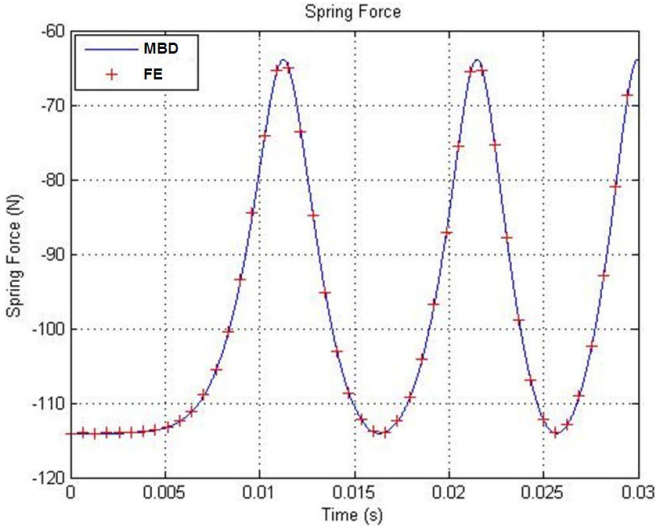


Figure 5: Time-history of the spring force generated using FE (Δt = 0.4E-5) and MBD.

Figure 6 shows the time-histories of the X and Y reaction force components at joint B (between link K_3 and ground) generated using FE (Δt = 0.4E-5) and MBD. Figure 6 shows that the reaction forces predicted using FE have high-frequency oscillations. However, the time average of the forces predicted using FE is very close to the forces predicted by MBD. Also, the amplitude of the FE oscillations decreases with time. Figure 7 adds to the plots in Figure 6 FE simulation results with the time-step reduced by a factor of 4: Δt = 0.1E-5. From Figure 7, we see that the amplitude of the oscillations predicted with a FE time-step of 0.1E-5 is less than the amplitude using a time step of 0.4E-5. In addition, the amplitude decreases more quickly with time but the frequency of the oscillations is higher at the smaller time step.

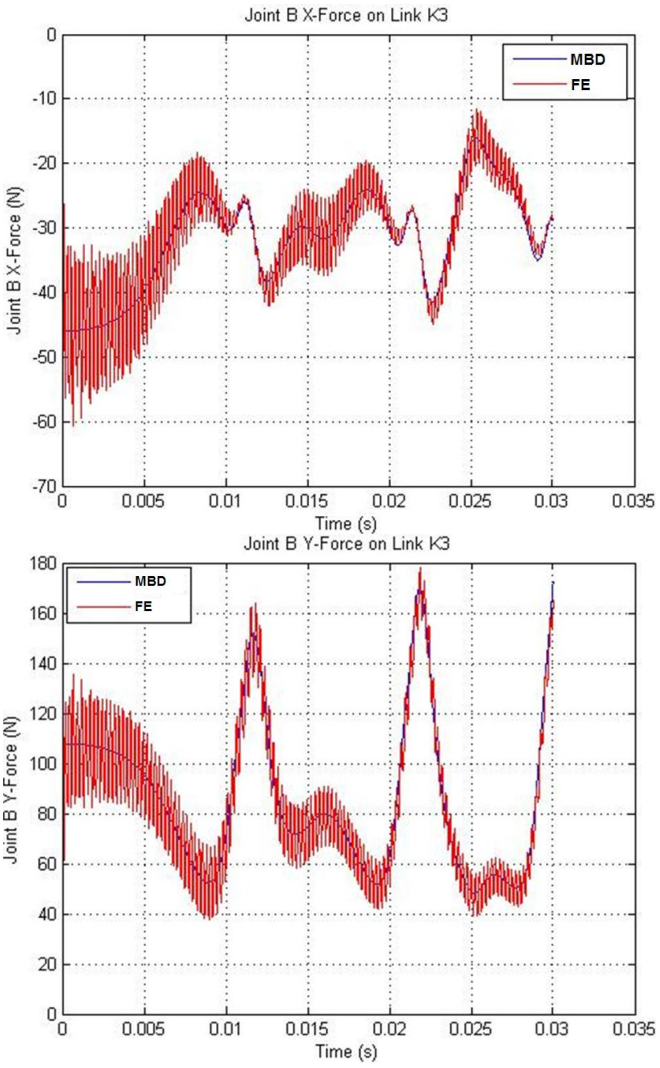


Figure 6: Time-histories of joint B (between link *K3* and ground) *X* and *Y* reaction force components generated using FE (Δt = 0.4E-5) and MBD.

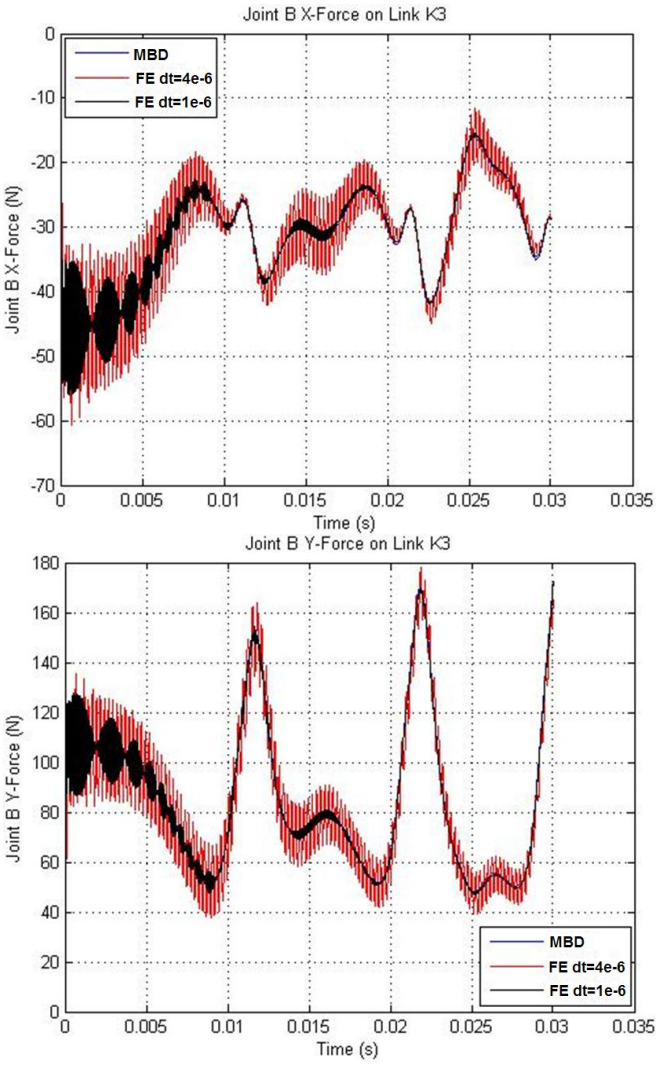


Figure 7: Same as Figure 6 but with the addition of FE results for a reduced time step of 1e-6 sec.

The reason why the forces predicted using finite-element software have high-frequency oscillations is because of the penalty technique used to model the joints. Each revolute joint in the mechanism is modeled using a penalty spring-damper. The stiffness and damping of the penalty spring-damper are set by FE such that they are the maximum allowable for a stable solution for the time-step used causing the oscillations. As the time-step is reduced, the stiffness and damping of the penalty spring-damper increase. So, the joint becomes stiffer increasing the oscillation frequency further. Though this approximates a perfect (infinitely stiff) joint better, it also causes the higher-frequency oscillations. The penalty stiffness in FE may be viewed as the physical joint

stiffness. Multibody dynamic software on the other hand models the joint using an algebraic constraint that enforces the joint constraint within tolerance without the use of stiff penalty springs. Thus, joints in MBD are near-perfect joints without the use of stiff springs.

Explicit finite-element software such as [1] cannot solve for static equilibrium as the way MBD solves using an implicit solver. Hence, to avoid the high-frequency oscillations in the FE solution, the initial drive torque and the spring compression can be gradually applied without being step inputs. To demonstrate this fact, another FE run is performed with the input torque linearly ramping from time 0 to time 0.0005 sec:

$$T_{K1} = \begin{cases} \dfrac{t}{0.0005}0.033 & 0 \le t \le 0.0005 \\ 0.033 & 0.0005 < t \le 0.1 \\ 0 & t > 0.1 \end{cases}$$

Also, the spring length is linearly ramped from the undeformed length to the initial compressed length in 0.0005 sec. Figure 8 shows the resulting reactions forces at joint *B*. These reaction forces exhibit no high-frequency oscillations and are close to the MBD reaction forces seen in Figure 7, except that the new results are shifted in time by 0.0005 sec.

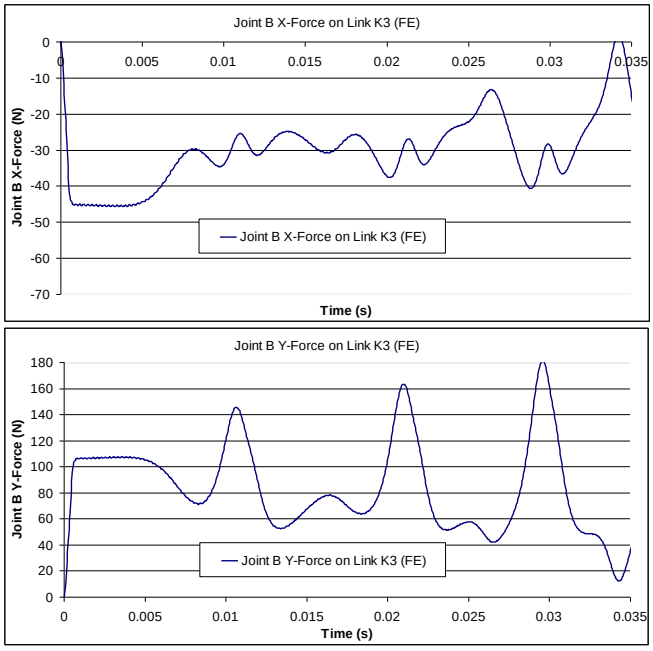


Figure 8: Time-histories of the X and Y reaction force components at joint B predicted using FE (Δt = 0.4E-5) by applying a ramped drive torque and a ramped initial spring deflection.

Figure 9 shows the time-history of the mechanism's potential energy predicted using FE and MBD. Figure 10 shows the time-history of the mechanism's total energy (sum of kinetic energy and potential energy). The drive torque is removed at time 0.1 sec, so the total energy must remain constant after that time. Both FE and MBD predict the correct mechanism total energy. The speed with which a solution looses energy with time that is supposed to have a constant total energy is a measure of the solution drift. Thus, the energy conservation capability of a code is a measure of the capability of the code to maintain an accurate solution with no solution drift over time.

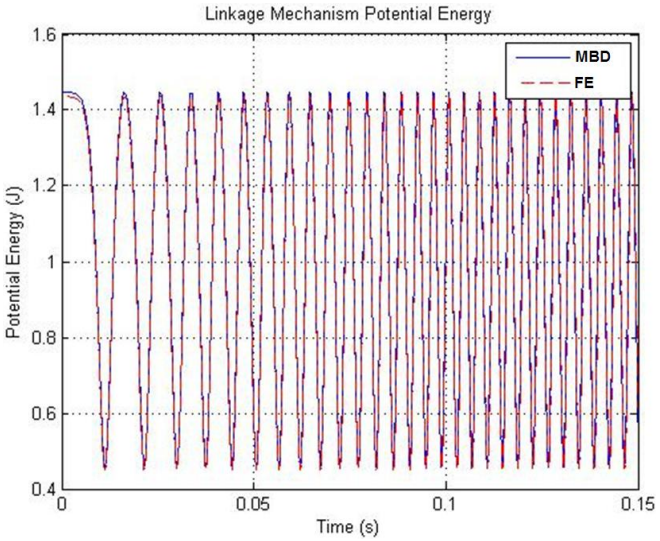


Figure 9: Time-history of the mechanism potential energy predicted using FE (Δt = 0.4E-5) and MBD.

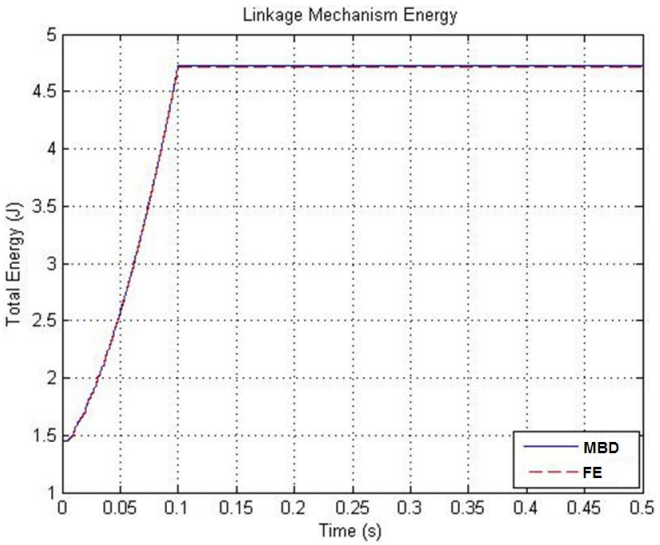


Figure 10: Time-history of the total system energy of the mechanism predicted using FE (Δt = 0.4E-5) and MBD.

Benchmark Problem 2: Spatial Robotic Manipulator

The 3-D robotic manipulator shown in Figure 11 was used as one of the benchmarks in *Multibody Systems Handbook* [3]. A description of the system and a list of system parameters are given in [3]. The manipulator consists of 3 rigid bodies connected by frictionless cylindrical and revolute joints. The mass, moments of inertia, and initial positions of the bodies are given in [3]. Each joint actuator of the manipulator is driven by a prescribed force and/or torque such that the end-effector traces a straight line with a trapezoidal velocity profile over a simulation time of 2 sec. The manipulator system has 5 degrees-of-freedom.

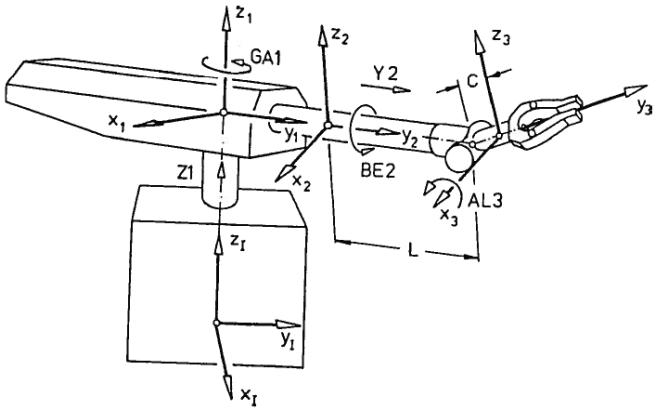


Figure 11: Spatial robotic manipulator [3].

Table 3 shows the MBD and FE simulation CPU time and corresponding time step used. For implicit codes the CPU time increases as the square of the number of coordinates whereas for explicit codes CPU time increases linearly with the number of coordinates. The motion of the robot is slow and does not involve high-speed rotation, thus a large time step can be used with an implicit code. However, explicit codes must use a time step that is smaller than the critical time step even for slow moving systems. This is because the time step depends primarily on the penalty stiffness of the joint constraints and not on the speed of the motion. For problems involving a small number of coordinates and a slow motion, implicit codes are always more computationally efficient than explicit codes. However, as the number of coordinates increases and/or as the speed of the motion increases explicit codes become more computationally efficient relative to implicit codes.

Table 3: Comparison of MBD and FE time steps and CPU times for the robotic manipulator. Runs were performed on a DELL OPTIPLEX 760 Intel Core 2 Duo 3.16GHz.

	MBD	**FE**
Time step (s)	0.01	3.75E-5
CPU time (s)	0.172	1.29

Figure 12 shows snapshots of the motion of the manipulator simulated using FE. Figures 13-18 show the time-histories of joint actuator motions generated using FE and MBD and compared with Handbook [3] results. These figures show that the motion predicted by FE and MBD are practically the same though small oscillations are noticeable in the FE result in Figure 16.

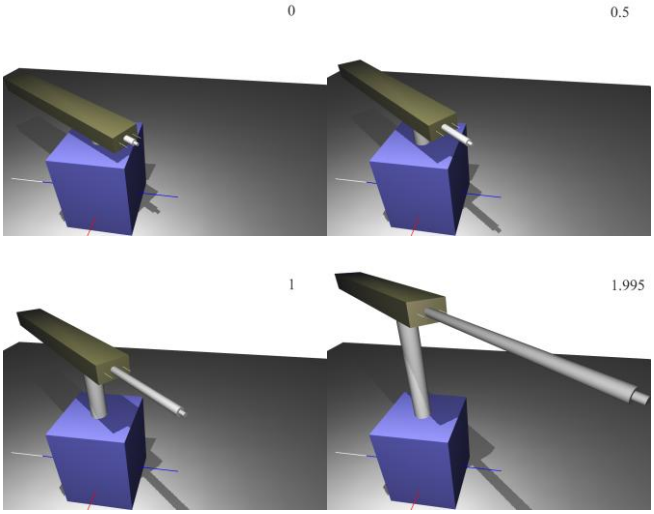


Figure 12: Snapshots of the motion of the manipulator simulated using FE.

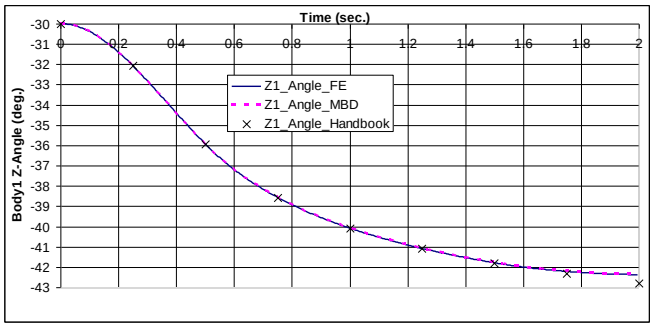


Figure 13: The rotation angle of body 1 around the z-axis (GA1).

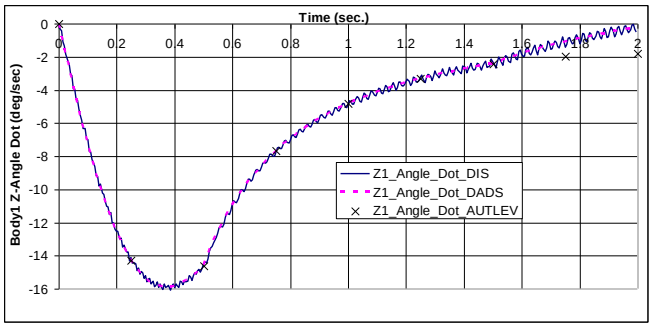


Figure 16: The angular velocity of body 1 about the z-axis (GA1_dot).

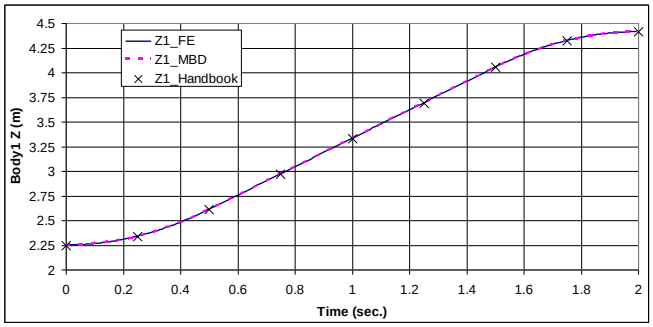


Figure 14: The z-coordinate of body 1 (Z1).

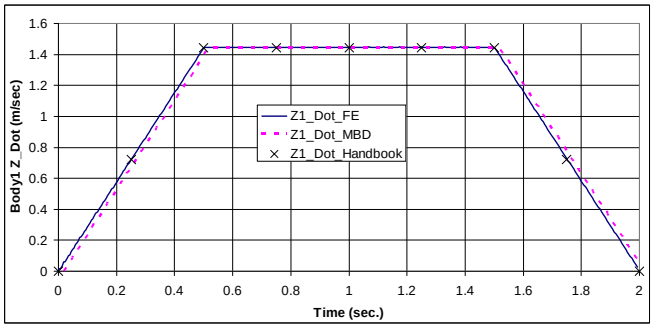


Figure 17: The velocity of body 1 along z-axis (Z1_dot).

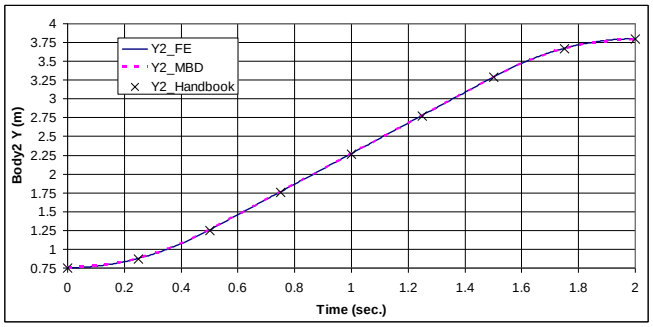


Figure 15: The y-coordinate of body 2 (Y2).

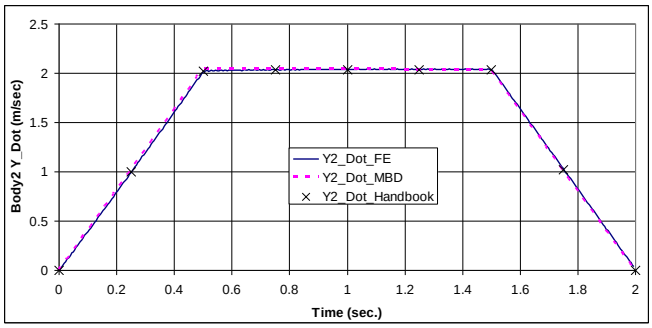
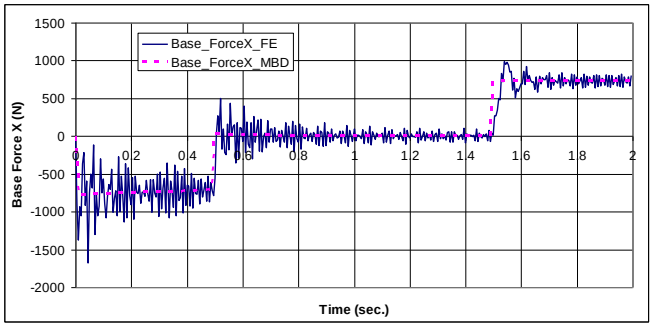
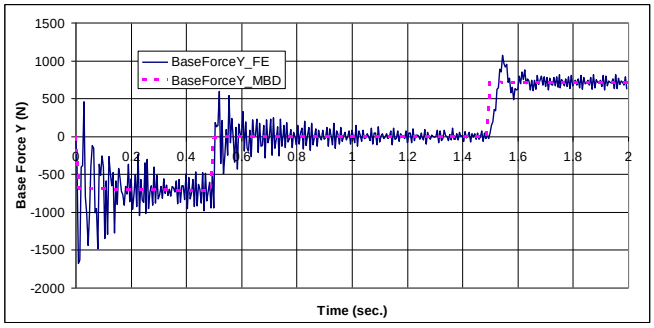


Figure 18: The velocity of body 2 along y-axis (Y2_dot).

Figure 19 shows the time-histories of the *X*, *Y*, and Z reaction force components at the base of the robot. Similar to the previous mechanism benchmark problem, Figure 19 shows that the reaction forces predicted using FE have high-frequency oscillations. However, the time average of the forces predicted using FE is very close to the forces predicted using MBD. The reason for the high-frequency force oscillations in FE is again because the joints are modeled using penalty spring-dampers and at the same time the joint actuator torques and forces have jump discontinuities at times 0, 0.5, and 1.5 sec. In order to eliminate the force oscillations in FE, the drive forces and torques need to be made continuous which is also physical.





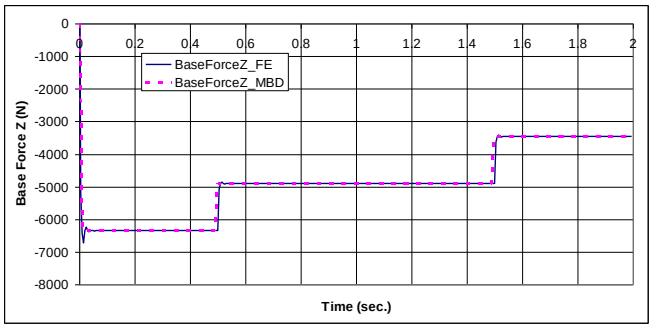


Figure 19: X, Y, and Z components of the reaction force on the base of the robotic manipulator.

CONCLUSIONS

Two multibody dynamics benchmark problems were solved using an explicit finite-element code and an implicit multibody dynamics code. The two codes predict practically the same system motion. However, the joint reaction forces predicted by FE have high-frequency oscillations. These oscillations are due to the penalty spring-dampers used for modeling the joints and the presence of discontinuities in applied force/moment. If applied forces/moments were continuous and the simulation were started from a static equilibrium configuration, then the FE solution would not exhibit the high-frequency force oscillations. An explicit finite-element code such as [1] offers the advantage of modeling non-linear structural flexibility, fluid-structure interaction, and thermal effects. However, multibody dynamics code such as [2] offers accurate and efficient solutions.

REFERENCES

[1] Wasfy, T. M., "Modeling Spatial Rigid Multibody Systems Using An Explicit-Time Integration Finite Element Solver And A Penalty Formulation," ASME Paper No. DETC2004-57352, Proceeding of the DETC: *28th Biennial Mechanisms and Robotics Conference*, DETC, Salt Lake, Utah, 2004.

[2] Haug, E. J., *Computer-Aided Kinematics and Dynamics of Mechanical Systems, Volume I: Basic Methods*, Boston: Allyn and Bacon, 1989.

[3] Schiehlen, W. (ed.), *Multibody Systems Handbook.* Berlin: Springer-Verlag, 1990.