International Conference
1ST International Conference on Innovations, Recent Trends and Challenges
in Mechatronics, Mechanical Engineering and New High-Tech Products
Development
MECAHITECH'09

Bucharest, 8-9 October 2009

Techniques to Incorporate Multibody Solvers Into Multiphysics Applications

William Prescott[1], Daniel Tohoneanu[2]
[1]LMS, 2425 Oakdale Blvd., Coralville, IA, 52246, USA, e-mail:bill.prescott@lmsintl.com
[2]LMS-Romania, 15A I. Slavici, Brasov, BV, RO, e-mail:daniel.tohoneanu@lmsintl.com

ABSTRACT

The traditional role of multibody dynamics has been in the analysis of existing designs to address problems or failures. The use of multibody tools to do virtual prototyping is now changing the role of multibody codes forcing them to become part of the design phase. The need for digital prototypes has produced the need for complete digital mock-ups to replace physical testing. In a virtual prototype environment the multibody dynamics solver is now is one part of a larger simulation environment, which includes other physics solvers. Multibody solvers may now have to simulate in concert with software used to model control systems or hydraulics. This increased use of multibody dynamics solvers in a multiphysics or multidomain application will only continue to grow. This is particularly true in the automotive industry where vehicles are being continually updated with intelligent systems. Intelligent systems usually take the form of an active control system used to control some aspect of the mechanisms motion. Commonly, the control system is not simulated with the same solver as the multibody system. Multibody solvers are focused on the solution of 3D mechanisms that are formulated with a set of differential-algebraic equations of motion while other physic solvers might only represent 1D equations of motion, which may or may not be formulated as a set of differential-algebraic equations.

This paper will review current techniques to incorporate multibody solvers into a multiphysics application. A model of industrial scope will be used to demonstrate the process.

INTRODUCTION

The use of so-called intelligent systems is becoming one of the largest drivers in not only the automotive industry, but in all mechanical systems. The use of intelligent systems requires a multibody dynamics solver to interact with different types of physical systems such as electronics, controls, hydraulics, etc. This mixing of systems is sometimes referred to as a multiphysics environment. The coupling of the systems will require mixing different forms of the governing equations such as differential-algebraic, discrete time equations or pure algebraic equations. In the multibody dynamics field the equations of motion are usually formed as a set of differential-algebraic equations, while a digital controller will formulate its equations in the discrete time domain and a finite element solver solving a static solution forms a purely algebraic set of equations. One of the complicating factors is that the different physical systems can often be seen as a black-box to the other systems and they only communicate with each other at a few connection points. Further, the different physical systems are often modeled in different software packages.

The solution of such multiphysics systems requires special solution techniques the two most common techniques can be generically termed coupled and co-simulation. A coupled simulation is one in which one solver is used to solve the complete system of equations while cosimulation uses separate solvers and synchronizes the solvers time stepping and communication between other blocks. In both types of systems a master-slave relationship is set up between the different solvers. The master solver controls the overall progression in time of the total simulation in both simulation modes.

Both coupled simulation and co-simulation have their advantages and disadvantages. A cosimulation environment allows a wide variety of solvers to be used together and for each system's solver to be used to solve its own set of equations. This can be a big advantage in that it takes advantage of a solver that has been specifically designed to solve a given type of system. The disadvantage of co-simulation is the communication interval between the systems is discrete and introduces a delay between the systems, which can cause instability. A coupled simulation will not introduce this delay and therefore is not as prone to instability problems. However, it is not always possible for each separate simulator to expose its underlying equations of motion to be solved by another solver and coupled simulations can result in longer simulation times than cosimulations.

This paper will explore the use of both coupled and co-simulation for multibody dynamics simulation. Further, an industrial model combining flexible bodies, multibody dynamics and hydraulics will be presented.

MULTIPHYSICS

In a multiphysics environment the system to be modeled is split into a number of distinct blocks or subsystems each subsystem may or may not have its own solver. The term multiphysics is used because each subsystem may represent a different physical domain such a hydraulics, electronics, fluid dynamics or thermal systems. Typically, these blocks are in different software packages. In Figure 1 a simple schematic of a system is shown where a multibody dynamics software must communicate with both a hydraulics model and a digital controller.

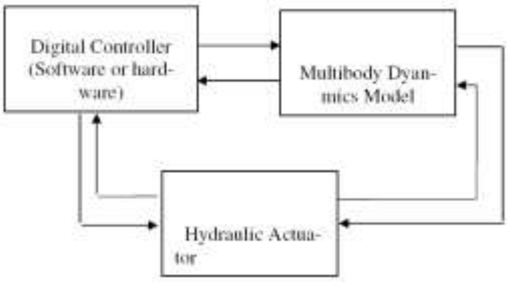


Figure 1: Multiphysics Environment

In this set up the digital controller can also be hardware meaning the multibody system can become part of either a hardware-in-the-loop environment or a software-in-the-loop environment. This system is becoming more prevalent in the industrial setting as the concept of digital mock-ups is being expanded to include all the subsystems not just the multibody dynamics. Even though different level of fidelity may be used in the different stages of the design steps by using variants of the same multibody dynamics model the same parameters can be adjusted in both the high fidelity and real-time models.

Master/Slave Relationship

In the environment such as the one shown in Figure 1 there has to be a master solver that controls the overall progression of the simulation. This solver may or may not be the multibody dynamics solver whether it is or not depends on the situation. The two basic methods for solving this set up are building one solver that solves all the system equations known as a coupled simulation and in the other situation a co-simulation is performed where each subsystem solves its own state equations. The master solver coordinates the progression of the simulation in time whether this is by co-simulation or coupled simulation. In either solution method the master solver determines the time step and the other "slave" solvers also move forward at this time step in the case of a coupled solution the slave solver becomes part of the master solver.

COUPLED SIMULATION

A coupled simulation is one in which all the different physical system formulations are solved by a single solver. This master solver may or may not be the multibody dynamics solver. In the case of a multibody dynamics solver we have a set of differential-algebraic equations the physics of the other solver may be a set of ordinary differential equations or another set of DAEs. In Eq. (1) an example of a coupled set of equations is given. The multibody equations are contained in Eq. (1a) and Eq. (1b) in Newton-Euler form. The generalized coordinates are the vector *q*, the generalized coordinate velocities are *v* and the vector λ is the LaGrange multipliers used to append the algebraic constraint. The matrix *M* is the mass matrix and Q_a is a vector of applied forces.

$$M\dot{v} + \Phi_q^T \lambda - Q_a(q, \lambda, \chi) = 0 \tag{1a}$$

$$\Phi(q, t) = 0 \tag{1b}$$

$$\dot{\chi} = g(q, v, \dot{v}, \lambda, \chi) = 0 \tag{1c}$$

A through derivation of this formulation can be found in [1]. The Eq. (1c) is a state equation from the other physics software and can represent a pressure, current, etc. The variable χ, in this equation represents the state vector.

In a coupled the system that formulates the largest number of equations is typically the master solver. It is often the case this is the multibody dynamics solver.

Differential Algebraic Equations

Differential-algebraic equations (DAE) are a common formulation in many types of simulations it is therefore necessary that the master solver in use must be able to handle these type of equations. It will also be necessary that the master solver be equipped with an implicit integration scheme this is a necessity because it is hard to predict whether or not the different blocks will be stiff. Certain physical systems such as hydraulics usually result in numerically stiff systems.

Discontinuity Handling

The master solver must be able to handle the other systems discontinuities. A discontinuity might be a discrete event such as the contact between bodies or a valve opening or closing in a hydraulic system. Usually, a discrete event causes a change in the formulation structure resulting in a new set of equations to be solved. This requires the master solver be able to handle root finding to isolate the discontinuities and restart the other solvers on the other side of the discontinuity.

Consistent Initial Conditions

One of the challenges of a coupled simulation is setting the initial conditions to be consistent between the different subsystems. One of the advantages of a coupled simulation over a co-simulation is a static solver can be used to settle the complete system into steady-state before launching the simulation. The discrete communication interval in a co-simulation environment can prohibit the use of a static solution.

CO-SIMULATION

A co-simulation system, similar to the coupled simulation, requires a master solver, however, in this case the master solver is not solving all the equations of motion of every subsystem, but synchronizing the communication and updating of all the other solvers in the co-simulation environment. One of the strengths of the co-simulation environment is that any one of the systems can be the master solver or a separate executable can be designed to control all the systems.

In a co-simulation environment each multiphysics block is running its own solver the advantage is that each subsystem has a solver tuned for its particular domain. Each block is moving forward in time on its own. The master simulation controller will tell each subsystem to integrate its own system of state equations forward in time.

Input/Output Communication

Communication between the blocks only happens at discrete times during the simulation. If one of the subsystems in the block is a controller and the other block is a multibody system solver then the input/output between the blocks will probably be such that the control system sends control forces or torques to the multibody system while the multibody system returns positions and velocities to the control system.

If the control system is the master solver then if the control system solver moves forward first in time it will only have the positions and velocities of the multibody system from the last time step. The same happens in the other direction when the control system has finished integrating forward in time the multibody system will next move forward in time. In this scenario the multibody system has the advantage that it has at least two sets of inputs from the control system one at the current time and one at the past time step. However, it has no intermediate data between the two points.

If the control system is actually a discrete time controller sampling at the communication rate between the control system and the multibody system then this delay between the systems should not cause any stability problems because the physical controller must be designed and tuned with this delay present. However, if this is not the case then some sort of interpolation/extrapolation method must be used between the two codes.

Different techniques have been developed to compensate for the instability caused by the delay. Transmission line modeling has been used to co-simulate multibody models into a cosimulation environment [3, 4]. A sliding-mode control algorithm has been developed to connect proprietary subsystems with algebraic connections [3]. The simplest method is the use of simple polynomial interpolation and extrapolation. Use of interpolation and extrapolation requires a number of past values of the input and output to be saved and then a polynomial is fit through these points this is the method used in this paper.

Consistent Initial Conditions in Co-simulation

One shortcoming of using co-simulation versus coupled simulation is the setting of initial conditions. A typical scenario in multibody dynamics is to settle the mechanism to an equilibrium position and to then start the solution from this configuration this will eliminate any transients in the system. Usually, a static solver is used to settle the system. A static solution requires the solution of the algebraic equations by a Newton's method. The difficulty in a cosimulation environment is that a standard static solution can not be easily performed. Each subsystem block will try and perform its own steady-state or static solution with its own Newton's method. Since a static solution happens at a fixed instant in time the communication interval between the systems can cause the Jacobian of each subsystem to be singular.

Assume a simple mass-spring-damper system is split between two subsystems such that the mass is in one subsystem and the spring damper is in another subsystem block as in Figure 2. While this system may seem overly simple it is representative of a vehicle model where the chassis body and tires are all modeled in a multibody system, but an active suspension is modeled in a different software package. This system can be solved in a dynamics solution, but it will not solve in a static solver this can best be determined by looking at the dynamic equations for the multibody system.

$$m\ddot{x} = Force\left(t^{i}, t^{i-1}\right) \tag{2}$$

In Eq. (2) the force is not a function of the position or velocity, but is a time dependent force only. In statics the Jacobian of Eq. (2) is singular. This means that the system must be solved by a dynamic settling solution where the system is solved using numerical integration until it reaches equilibrium and the new solution is restarted from the settled position. A dynamic settling run is a valid method to achieve a static equilibrium position, however, it is often computationally expensive and the final solution for the co-simulation might not be the same as for the coupled simulation. The difference in solutions may result from the time delay between the co-simulation blocks.

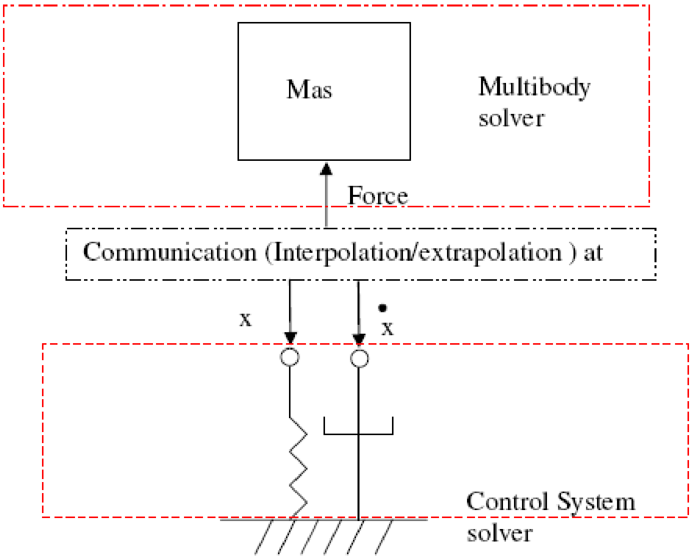


Figure 2: Partitioned Mass-Spring Damper

LANDING GEAR EXAMPLE

As an example model the drop test of an aircraft landing gear is used. The actual model used is shown in Figure 3. The post body of the landing gear is modeled both as a flexible body and as a rigid body. The drop test simulates the landing of the aircraft; therefore the mechanism is dropped from a given height above the runway. The model includes both a multibody model for the mechanism and a hydraulic model for the oleo.

Multibody Model

The multibody model is modeled in LMS Virtual.Lab Motion. The model includes 7 bodies and dual tires. The model will be run with the post body being both flexible and rigid. The Craig-Bampton or modal synthesis approach is used to model the flexibility of the post body. To capture the structural modal response of the post body the modal content will include 29 modes with frequencies ranging from 426Hz to 12059 Hz. The mechanism parts of the oleo the plunger and the cylinder are modeled inside of the multibody program while the hydraulics is modeled in a separate software system.

Hydraulics Model

The oleo or damper is modeled as a hydraulic cylinder inside of Imagine.Lab/AMESim software. The hydraulic schematic is shown if Figure 4. The multibody model feeds into this hydraulic circuit the displacement and velocity of the oleo bodies. In turn the hydraulic software calculates the force to contract or expand the oleo and feeds this back to the multibody software.

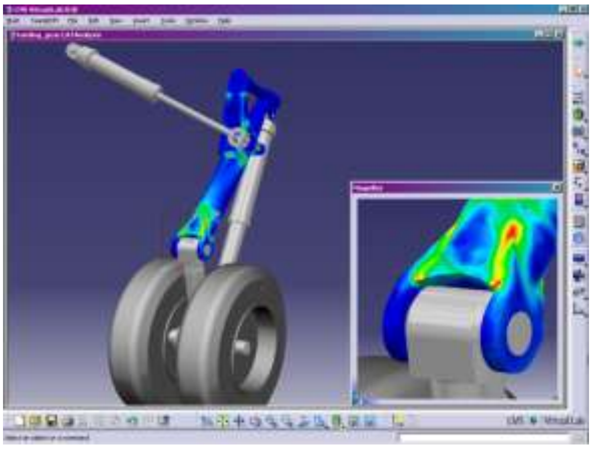


Figure 3: Flexible Landing Gear Model

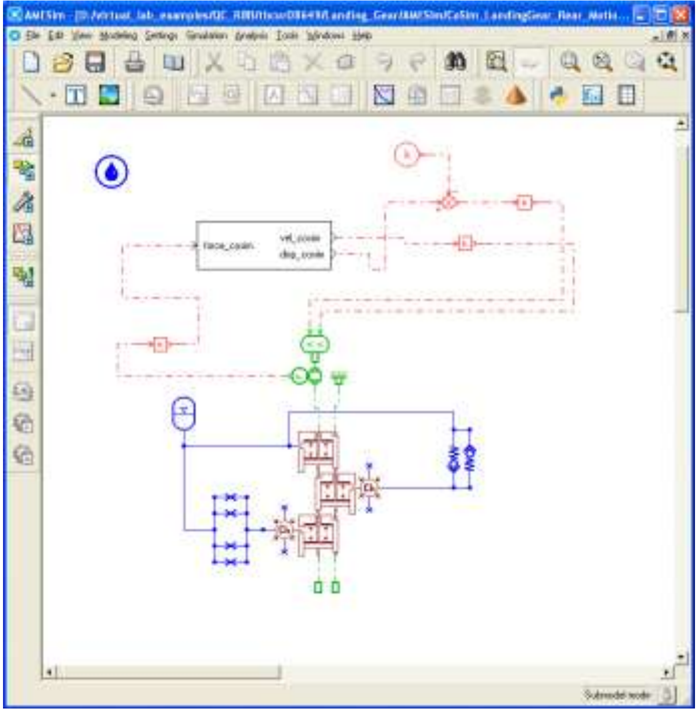


Figure 4: Landing Gear Hydraulic Circuit

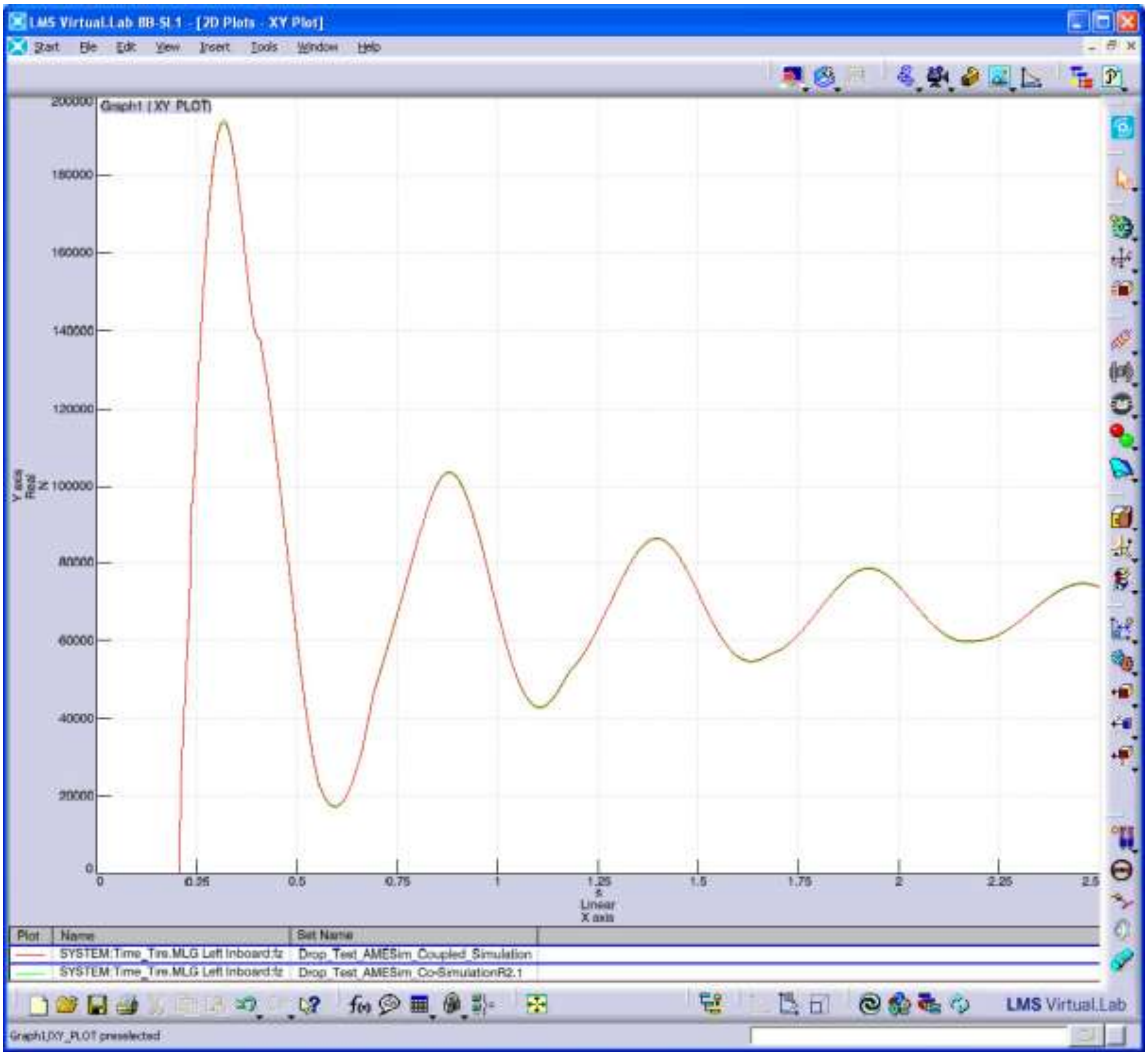
Test Cases

The Virtual.Lab motion has both a coupled and co-simulation module available with Imagine.Lab, therefore this drop test will be performed in both a coupled and co-simulation mode. In both cases the tires are spun up to simulate landing and the landing gear dropped on the runway to simulate the impact of landing.

Rigid Post Case

The model will be first run with the post body being rigid. The simulation is done in both the coupled mode and the co-simulation mode. In the co-simulation mode a communication interval between the multibody model and the hydraulics model of 1 millisecond is used and a zero-order hold is used to communicate the inputs and outputs between the multibody and hydraulics model.

In Figure 5 the tire normal force of both solutions is shown and it can be seen there is almost no difference between the results. For this solution the co-simulation with a communication interval of 1 millisecond is stable and gives almost identical results as the coupled simulation. A second solution at 2 milliseconds was also done and completed stably this communication interval was found to be the upper limit of stability for this model.



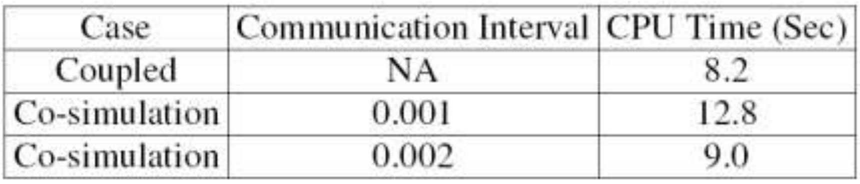
Plot	Name	Set Name
🟥	SYSTEM:Time_Tire.MLG Left Inboard:fz	Drop_Test_AMESim_Coupled_Simulation
🟩	SYSTEM:Time_Tire.MLG Left Inboard:fz	Drop_Test_AMESim_Co-SimulationR2.1

Graph1.XY_PLOT preselected

Figure 5: Tire Vertical Force for Rigid Post Model

The simulation times for the coupled versus co-simulation solutions are given in Table 1. The results show that for this case the coupled simulation is actually slightly faster.

Table 1: Solution Times for Rigid Post Solutions.



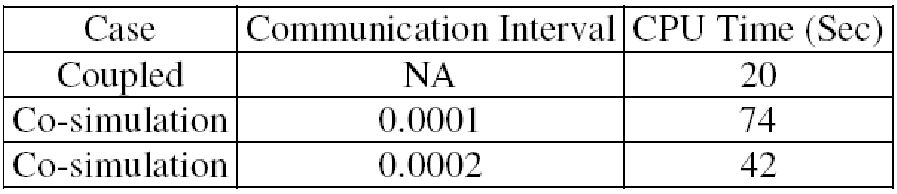
Case	Communication Interval	CPU Time (Sec)
Coupled	NA	8.2
Co-simulation	0.001	12.8
Co-simulation	0.002	9.0

Flexible Post Case

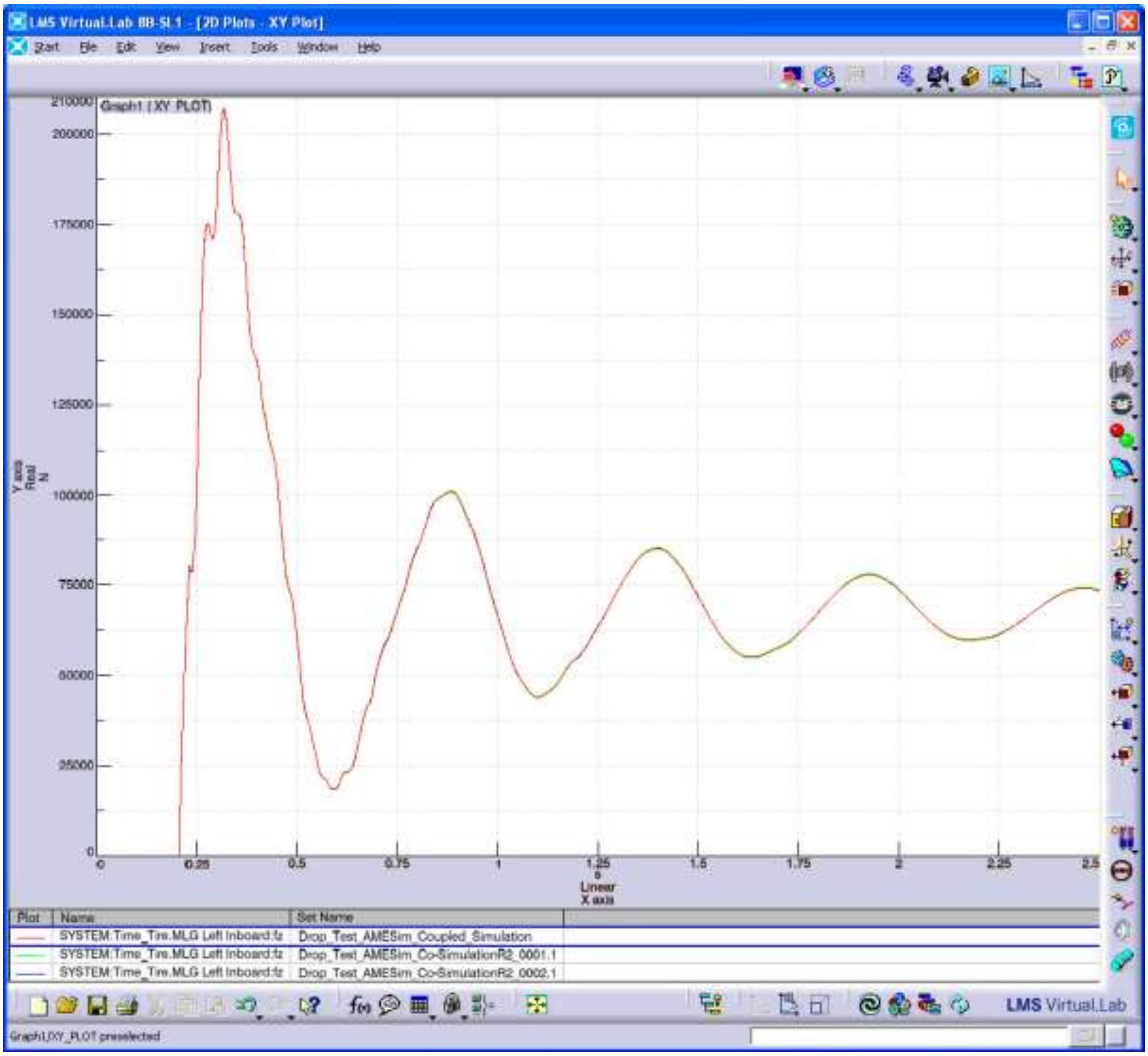
Next the post is made a flexible body and the same series of simulations rerun. The results between flexible and rigid solutions are not expected to be the same. The co-simulation run in this flexible body case was also executed at a rate of 1 millisecond. However, in this case the simulation could not complete. For the flexible body case the frequency content is different because of the flexible modes. This large of a communication interval will cause the simulation to abort. If the communication interval is reduced by an order of magnitude to 0.1 milliseconds then simulation will complete and give good quality results. Experimentation found the communication interval could be increased to 0.2 milliseconds and the simulation completes, however, a larger communication will cause failure. The tire vertical force is shown in Figure 6 for the coupled simulation, 0.1 millisecond and 0.2 millisecond simulation. The results for all three simulations are virtually identical.

The solution times for all simulations are given in Table 2. Examination of Table 2 shows that even for the flexible post case the coupled solver is significantly faster.

Table 2: Solution Times for Flexible Post Solutions.



Case	Communication Interval	CPU Time (Sec)
Coupled	NA	20
Co-simulation	0.0001	74
Co-simulation	0.0002	42



Plot	Name	Set Name
—	SYSTEM:Time_Tire.MLG Left Inboard:fz	Drop_Test_AMESim_Coupled_Simulation
—	SYSTEM:Time_Tire.MLG Left Inboard:fz	Drop_Test_AMESim_Co-SimulationR2_0001.1
—	SYSTEM:Time_Tire.MLG Left Inboard:fz	Drop_Test_AMESim_Co-SimulationR2_0002.1

Graph1.XY_PLOT preselected

Figure 6: Tire Vertical Force for Flexible Post Model

CONCLUSIONS

The results of the landing gear drop test show that a multibody dynamics solver and another physics solver can be combined into one model. In this particular example the results showed the coupled simulation solver to perform better, however, this is not always expected. For tightly coupled systems such as this where a hydraulic actuator model is split between two packages this is to be expected in such a case the communication interval is so small to maintain stability that there is no real performance advantage to co-simulation. For loosely coupled systems this not expected to be the case here the co-simulation should show a real advantage.

It was shown that with the co-simulation that it can be performed stably depending on the communication interval. This is an important result since the use of physical prototyping is being reduced in favor of simulation, which is leading to ever large more complex simulation models with subsystems from a wide variety of software packages. Since not all software packages can make its state equations available for a coupled simulation then it becomes necessary to use co-simulation.

Despite the need for co-simulation a coupled simulation can have some advantages. First, it is possible to do static or steady-state solution which might not be possible in co-simulation due to the communication interval. It was also found that choosing communication interval for a co-simulation can be tricky. For this example the communication interval that worked for the rigid body case did not work for the co-simulation case it had to be reduced significantly. This may present a problem in design of experiments or optimization studies where parametric changes may make the selected communication interval unstable for the new model variation.

REFERENCES

[1] E. Haug. “Computer-Aided Kinematics and Dynamics of Mechanical Systems Volume I: Basic Methods”., Allyn and Bacon, Needham Heights, MA., 1989.

[2] P. Krus, “Modeling of Mechanical Systems using Rigid Bodies and Transmission Line Joints”. *Transactions of ASME Journal of Dynamic Systems Measurement and Control*, Dec 1999.

[3] B. Gu, H. Asada, “Co-Simulation of Algebraically Coupled Dynamic Subsystems Without Disclosure of Proprietary Subsystem Models”, *Journal of Dynamic Systems, Measurement and Control*, Vol. 126, Issue 1, March 2004.

[4] D. Fritzson, J. Stahl, I. Nakhimovski, “Transmission Line co-simulation of rolling bearing applications”. In Proceedings of the 48th Conference of Simulation and Modeling.