

Interfacing Abaqus with Dymola: A High Fidelity Anti-Lock Brake System Simulation

Brad Schofield Harish Surendranath Magnus Gäfvert Victor Oancea
Modelon AB, Ideon Science Park, SE-22370 Lund, Sweden
{brad.schofield, magnus.gafvert}@modelon.se
Dassault Systèmes Simulia Corp., 166 Valley Street, Providence, RI 02909
{harish.surendranath, victor.oancea}@3ds.com

Abstract

Accurate simulation of anti-lock braking systems (ABS) requires detailed models of several subsystems in different physical domains. The most important subsystems are the hydraulic brake system, the tire and the control algorithm. The creation of detailed models of each subsystem in a single modeling tool may be difficult if not impossible. To overcome this, co-simulation may be used to combine the strengths of different tools. In this article, co-simulation between Dymola and Abaqus is used to investigate the performance of an ABS algorithm with a highly detailed finite-element tire model. The brake system hydraulics along with the control algorithm are simulated in Dymola while the tire model, the wheel, the braking caliper and the contact with the road are simulated in Abaqus.

While computationally more expensive than a traditional modeling approach when a semi-analytical tire model (such as the Magic Formula model) may be used to model the tire and tire road interaction, the approach described in this paper includes a fair amount of details when modeling of the tread, the tire plies, the wire reinforcements in the tire and the contact with the road. The necessary data is exchanged between the two applications using the co-simulation capabilities available in Abaqus and the .DLL option in Dymola. Sensors in Abaqus provide information about the mechanical state of the system such as forward translational velocity, angular velocity/acceleration and the free rolling effective radius. This information is communicated to Dymola at frequent simulation time intervals at runtime. Dymola uses this information as inputs and computes the brake caliper clamp force. This force is communicated in turn to Abaqus which determines the force on the brake rotor.

Keywords: Anti-lock Brake Systems, Dymola,

Abaqus, Hydraulics, Finite elements, Automotive Control, Co-simulation

1 Introduction

Modeling automotive systems very often necessitates modeling in multiple physical domains. The Modelica language is a natural choice for such multi-domain modeling, but for situations in which very high resolution is required, or where physically-derived models are not available, other tools such as finite-element modeling may be more appropriate. In such situations, co-simulation may be used to combine the strengths of different modeling tools to achieve the desired level of accuracy. In this paper, co-simulation between Dymola and Abaqus is used to investigate the effects of ABS braking with a highly detailed finite-element tire model.

In the investigation, a single wheel setup was used. For simplicity, suspension components are not modeled. Dymola was used for the implementation of the brake system hydraulics model as well as the control algorithm, and Abaqus was used for the tire and brake rotor models. The signals necessary for ABS control are passed from Abaqus to Dymola, and the Dymola model provides a brake caliper clamp force output.

2 Brake System Modeling in Dymola

For the co-simulation, the hydraulic braking system as well as the control algorithm were implemented in Dymola. Hydraulic components from the HyLib library are used. The braking system consists of a single brake caliper cylinder, connected to a master cylinder via a three port valve. The three port valve is set up to have three modes of operation: a pressure increase mode in which the master cylinder is connected to the slave,

a hold mode in which all ports are disconnected, and a pressure decrease mode in which the slave cylinder is connected to a tank. A return circuit is not modeled. This system represents the simplest form of production ABS implementation in which brake fluid is not returned to the master cylinder during braking, but rather after the braking event has taken place.

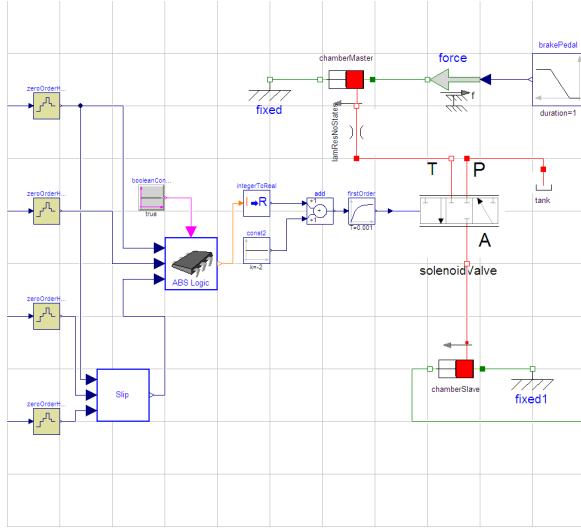


Figure 1: Diagram view of the brake system hydraulics and control model in Dymola

2.1 Control Algorithm

Control is performed by modulating the position of the three port valve depending on wheel acceleration and slip. The algorithm is based on that described in the Bosch Automotive Handbook [1], and is modeled as a state machine. The required inputs to the controller are wheel angular velocity ω , angular acceleration $\dot{\omega}$, rolling radius r and hub longitudinal velocity v_x . The input signals to the controller are sampled with period $T_s = 1ms$. The longitudinal slip is calculated as

$$\lambda = \frac{v_x - r\omega}{v_x}$$

The ABS is triggered when the wheel deceleration falls below the prescribed threshold $-a$. Pressure is then held until the slip exceeds a threshold λ_T , at which point pressure is dropped for a given time. Pressure is held until a positive acceleration A is reached, at which point pressure is increased until the acceleration drops to a . At this point pressure is increased slowly via alternate hold and increase commands. This allows the ‘peak’ of the friction characteristic to be traversed slowly, before the unstable side is

reached. The cycle begins again once the $-a$ acceleration threshold is crossed. The controller is deactivated for longitudinal velocities under a prescribed level. In addition, a timeout parameter is used to reset the ABS algorithm if it remains in any one state for extended periods. This is necessary to prevent the controller becoming ‘locked’ in any state when ABS action is no longer required.

The aim of the algorithm is essentially to move quickly away from the region of the slip characteristic corresponding to unstable slip dynamics, and to maximize the time spent near the the friction peak (to improve braking performance).

Rule-based control algorithms such as the one described here often suffer from problems such as large numbers of tuning parameters, lack of robustness to process variations and difficult stability and performance analysis. Model-based approaches often lead to more manageable systems with fewer tuning parameters and greater reusability. Nevertheless, rule-based algorithms are widespread in production systems. A model-based approach to ABS control design is described in [3].

3 Tire, road and brake model in Abaqus

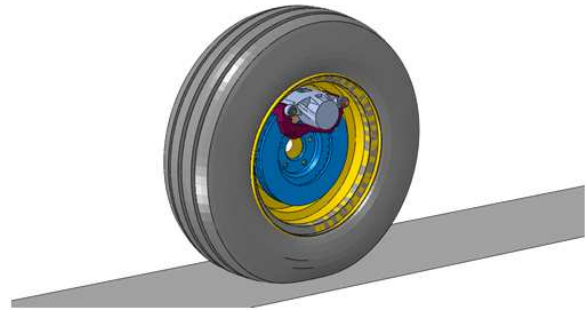


Figure 2: Animation of tire, road and brake model in Abaqus

The wheel, the tire, and the braking caliper-rotor subassembly are modeled in the finite element software Abaqus. The most important ingredient in the Abaqus model is the detailed modeling of the tire. The Abaqus software is used by a large number of tire manufacturers to study tire responses spanning steady state, braking or abuse type loading conditions such as curb hitting or driving over a pothole. The finite element tire models the tread, plies and wire reinforcements in the actual tire. The tire is first pressurized

and placed in contact with the road under the weight corresponding to that wheel (quarter car). While a flat road was depicted above, Abaqus modeling of more complex road geometries such as uneven cobblestone roads is readily available. A steady state transport analysis is performed next in Abaqus/Standard (implicit integration) to compute the state of stress and deformation in the tire corresponding to a given forward velocity before any braking is applied. Optionally, a cornering radius can be specified as well. The contact pressures between the tire and the road corresponding to this configuration are depicted below.



Figure 3: Brake disc model in Abaqus

The deformations and stresses in the tire are then imported in Abaqus/Explicit (explicit integration) which is used for the actual braking analysis via co-simulation with Dymola. Abaqus/Explicit solves for most of the mechanics in the analysis to update the current state of stress and deformation in the tire. Sensor information such as wheel angular velocity/accelerations are communicated at runtime at frequent simulation intervals to Dymola which computes the brake caliper cylinder pressure for effective braking. The braking pressure is then communicated back to Abaqus/Explicit which applies this pressure to the brake caliper cylinder depicted below. The brake pads are pressed against the brake rotor to produce a braking torque that decelerates the single wheel-brake sub-assembly.

The total number of degrees of freedom in the Abaqus model is about 125000 which are basically nodal translations and rotations.

4 Simulation of Single-Wheel Braking in Dymola

In order to test the ABS system in Dymola, a single-wheel test rig was constructed using components from the VehicleDynamics library, see Figure 4. Only lon-

gitudinal dynamics were modeled. A ‘Magic Formula’ tire model was used.

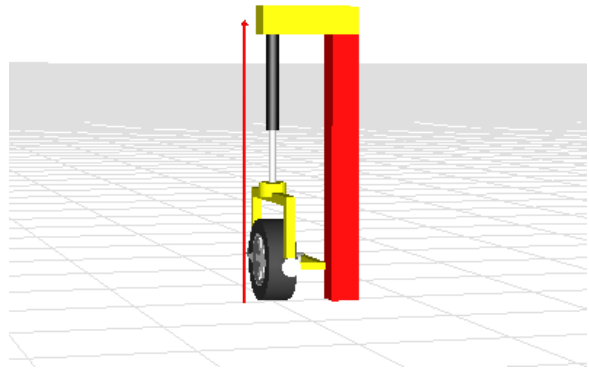


Figure 4: Animation of single-wheel test rig used for simulation in Dymola

The rig was simulated with an initial velocity of $10m/s$, and a brake pedal force ramp was applied. Figure 5 shows the state of the controller during the simulation. The rapid switching represents the slow build-up mode, achieved by alternating between hold and increase positions of the three port valve. The brake caliper clamp force is illustrated in Figure 6. In Figure 7 the hub fore-aft velocity and wheel circumferential velocity are shown. The wheel circumferential velocity is computed by multiplying the wheel angular velocity by the effective rolling radius.

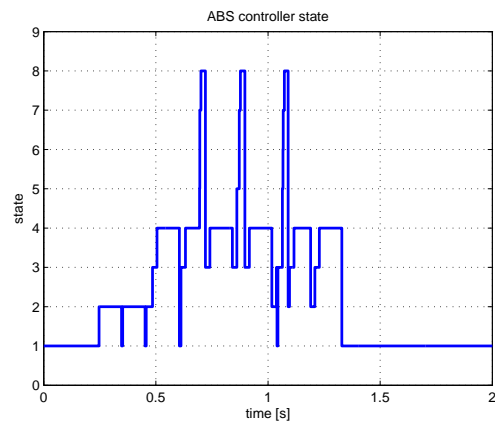


Figure 5: The ABS controller state during simulation of single-wheel braking in Dymola.

5 Co-simulation Setup

The physical system modeled with Abaqus is furnished (in Abaqus) with sensors and actuators to create what is commonly referred to as the Plant. The

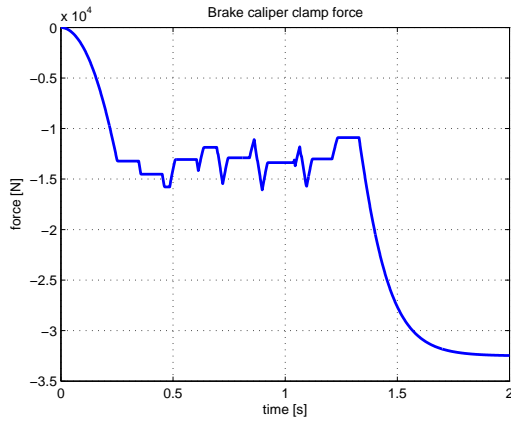


Figure 6: The brake caliper clamping force during simulation in Dymola.

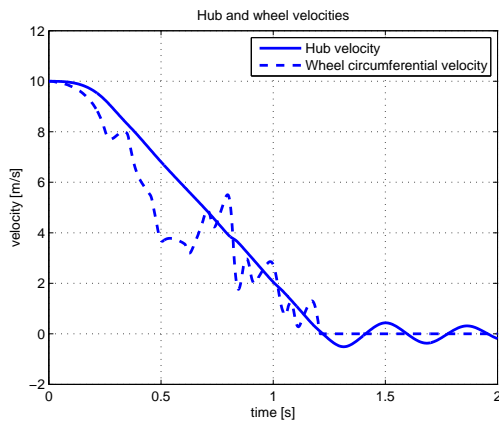


Figure 7: Hub velocity and wheel angular velocity during simulation in Dymola.

sensor data computed by Abaqus is passed to Dymola (the controller) which computes the needed actuation to drive the physical system in the desired state. The translational velocity, angular velocity, angular acceleration and the free rolling radius of the tire are defined as the sensors in Abaqus. The actuator (computed in Dymola during co-simulation) is the clamp force to push the brake pads against the brake disc. The solution in Abaqus/Explicit uses a central difference method to compute the equilibrium condition of the physical system at time $t + \delta t$ based on the equilibrium conditions at time t . One of the fundamental requirements of the central difference scheme is that the time step δt be smaller than a critical value, known as the stable time increment. In this particular application, the value of stable time increment is close to 3 microseconds. Note that this is significantly below the required time increment in Dymola which is on the order of milliseconds. The co-simulation procedure

analysis requires significantly more computational resources than a standalone Dymola run. This is primarily due to the fact that finite element computations in this application are computationally very intensive given the required small time increment size and the number of solution variables involved. Consequently, given that the number of unknowns is roughly 125000 in the Abaqus model and roughly 585000 increments are required to complete the 2.0 seconds of simulation time, the computational cost of the co-simulation is dominated by far by the cost of the finite element analysis. In this particular case, the analysis was run using 8 CPUs on a HP BL460 Intel Xeon Dual Core 3Ghz Linux 64 bit computer and required 237 minutes to complete. By contrast, the Dymola job required less than a minute of actual computational time to complete on a 32 bit Windows computer using only one CPU. Dymola with the .dll option is used. A C++ driver uses the API for the .dll to advance the computations in the ABS model.

The Abaqus co-simulation capabilities include various co-simulation rendezvousing schemes. It is not the purpose of this paper to review the algorithmic nature of these schemes as they are described in [2]. Given the time incrementation in Abaqus/Explicit requires a far smaller time increment than Dymola and that the required 3 microseconds time increment is far smaller than the sampling frequency of sensors used in ABS systems, a simple non-iterative co-simulation scheme is used:

- Each Abaqus increment sensor information is computed (Hub velocity, free rolling radius, angular velocity and angular acceleration) and then communicated to Dymola via a socket-based interface.
- Dymola (with the .dll option) is run via a driver that uses the API to the .dll to advance the simulation time in Dymola. The sensor information from Abaqus is read in as inputs to Dymola. Dymola integrates in time with a time step size equal to the Abaqus/Explicit time increment (roughly 3 microseconds) which is far smaller than what Dymola would require for an accurate integration. Actuators computed in Dymola (brake caliper pressure) are communicated back to Abaqus which applies this freshly computed load as the load for the next increment.
- The process is repeated until the simulation time is exhausted.

The co-simulation process was also run with an even smaller time increment (2 microseconds) to check whether the results change. The results were virtually identical.

6 Co-simulation Results

In this section the results of the co-simulation using are presented, and compared to the results obtained using the simple tire model in Dymola. It should be noted that no re-tuning of the ABS control algorithm was performed for the co-simulation. It is highly likely that performance could be improved considerably by re-tuning the controller with the Abaqus wheel model.

6.1 Parameter selection

The primary aim of the investigation was to test the co-simulation interface between Dymola and Abaqus on a relevant industrial example. In particular, due to the extremely large differences in complexity of the tire models in Dymola and Abaqus, it was not expected that the co-simulation results would be quantitatively very similar. To this end only a rudimentary synchronization of parameter values between the Dymola and Abaqus brake and wheel models was performed. Wheel and brake rotor diameters are similar but not exactly equal. Tire force-slip characteristics are close, but the stiffnesses differ slightly.

In the Dymola brake model, the braking moment is obtained using the clamping force, an effective radius and a friction model. The Abaqus brake model uses a nonuniform pressure distribution on the brake pads, making a comparison with the effective radius of the Dymola model difficult.

Figure 8 shows the hub and circumferential velocities during the co-simulation. The slip is initially large, but the controller is able to prevent wheel lock. Figure 9 shows the brake caliper clamping force during co-simulation. The rapid force build-up and release phases are clearly visible.

6.2 Comparison between co-simulation and Dymola

Figure 10 shows the hub and circumferential velocities in the co-simulation and Dymola-only simulation respectively. The qualitative behavior is clearly similar. The controller is more effective in the Dymola-only simulation, maintaining lower slip. This is to be expected as the controller was tuned using the Dymola tire model.

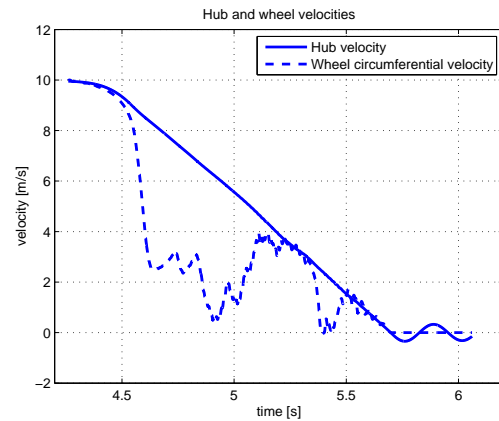


Figure 8: The hub velocity and wheel circumferential velocity during co-simulation.

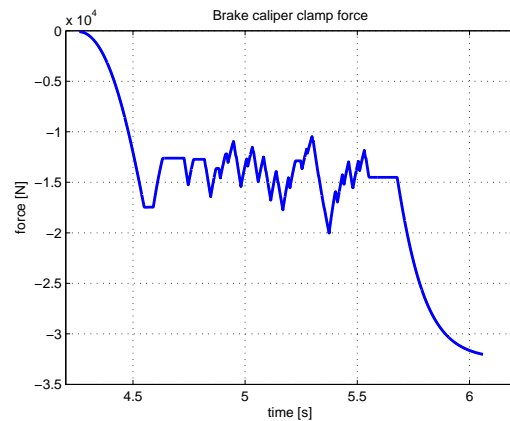


Figure 9: The brake caliper clamping force during co-simulation.

Figure 11 shows a comparison of the brake caliper clamping forces in the co-simulation and Dymola-only simulation. The forces are very similar. The resultant longitudinal tire forces are compared in Figure 12. The tire force results from the co-simulation had considerable high-frequency content, as may be expected from a high-fidelity model. For the comparison the results were low-pass filtered with a bandwidth of 100Hz.

During the initial braking the tire forces are very similar. A larger slip is developed in the co-simulation leading to lower tire forces. The oscillations seen after roughly 1.3 seconds are due to the test rig coming to a standstill. This behavior is observed in both the co-simulation and the Dymola-only simulation.

7 Conclusions

In this article, co-simulation is used to study anti-lock braking control using a high-fidelity tire model. Dy-

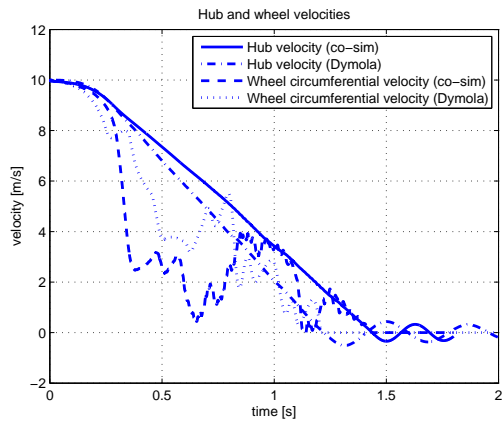


Figure 10: The hub velocities and wheel circumferential velocities during co-simulation and Dymola-only simulation.

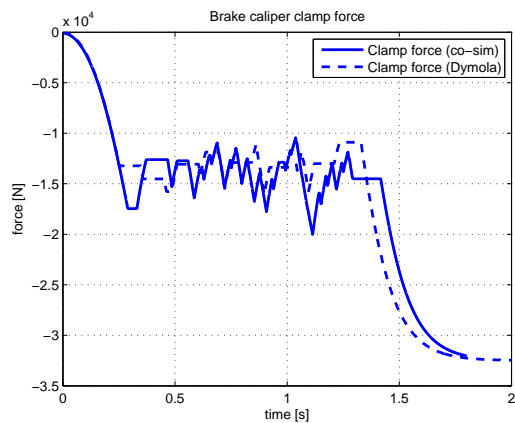


Figure 11: The brake caliper clamping forces during co-simulation and Dymola-only simulation.

mola is used to implement a realistic hydraulic braking circuit using existing components from the HyLib library. The control algorithm is also implemented in Dymola. Abaqus is used to implement very high fidelity tire, road and brake rotor models.

The co-simulation results presented here are preliminary in the sense that controller tuning was performed using a much simpler tire model in Dymola. No re-tuning was performed for the co-simulation. As previously mentioned, the tire characteristics and brake model parameters were not exactly the same for the Dymola and Abaqus models. Nevertheless, the ABS controller was capable of preventing wheel lock in the co-simulation with the high-fidelity wheel model.

This example illustrates how co-simulation between different packages may extend the realism of system-level simulations. The multi-domain modeling strengths of Dymola are combined with the compu-

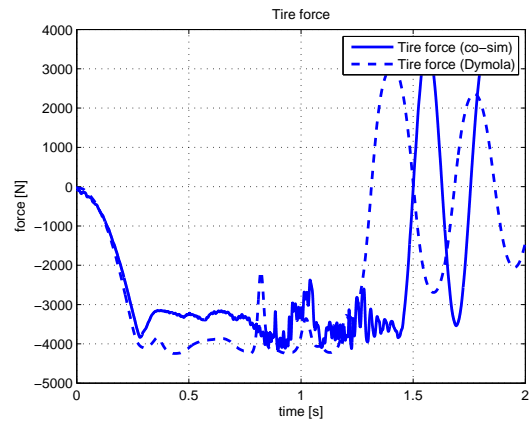


Figure 12: The tire longitudinal forces during co-simulation and Dymola-only simulation. The result for the cosimulation was low-pass filtered with 100Hz bandwidth.

tational power of Abaqus to provide a level of detail which would be extremely time-consuming to achieve with a single tool.

As this was a preliminary investigation, there is considerable scope for future work. In particular, tuning of the ABS controller using the Abaqus is expected to yield large performance increases. In addition, simulation with a full vehicle model in Abaqus is also planned. Successful co-simulation of an ABS system with a full vehicle model would then allow high-fidelity simulations of vehicle dynamics controllers to be performed.

References

- [1] Horst Bauer, editor. *Bosch Automotive Handbook*. Robert Bosch GmbH, 4th edition, October 1996.
- [2] Dassault Systèmes Simulia Corp. *Abaqus Analysis User's Manual*, 2009. Version 6.9.
- [3] Stefan Solyom, Anders Rantzer, and Jens Luedemann. Synthesis of a model-based tire slip controller. *Vehicle System Dynamics*, 41(6):477–511, June 2004.