

Multi-Domain Modeling and Simulation of an Electro-Hydraulic Implement System

Sameer M. Prabhu, Jeff Wendlandt, John Glass, and Tom Egel
The MathWorks, Inc.

ABSTRACT

The need to meet new regulatory requirements as well as customer expectations in terms of machine productivity, safety, maintenance and uptime, is driving a significant transformation from conventional hydraulic and mechanical systems to electro-hydraulic systems in the earth-moving and agricultural equipment industry. The ability to model and simulate such systems plays a key role in this transformation by allowing manufacturers to test whether the system meets requirements using virtual prototypes rather than physical prototypes. Modeling the electrical, electronic, mechanical, and hydraulic domains in the same modeling environment can significantly improve the product development process of such machines. This paper illustrates those benefits using the example of an electro-hydraulic implement system.

INTRODUCTION

The earthmoving and agricultural equipment industry is undergoing a significant transformation from conventional hydraulic and mechanical systems to electro-hydraulic systems. The market drivers for this transformation can be broadly classified into two categories. The first category deals with regulatory requirements on engine emissions, machine noise levels, etc. As an example, engine emissions regulations are getting more and more restrictive which in turn necessitates the adoption of electronics to precisely control engine combustion and after-treatment. The second category deals with the need to meet customer demands in terms of improved machine productivity, safety, maintenance and uptime. As an example, in an electro-hydraulic implement system the lift and tilt modulation characteristics can be adapted based on the work tool connected to the implement linkage (e.g., bucket vs. forks), thereby lessening the burden on the operator and improving productivity.

THE CHALLENGE OF ELECTRO-HYDRAULICS

However, it is not enough to just take individual systems such as implements and transmission and transform them to their electro-hydraulic equivalents. In order to

fully realize the benefits of electronics, both from a customer and a machine manufacturer point of view, it is imperative that these systems “talk” to each other and thereby can be fully integrated with each other into an overall machine system. As an example, in a typical off-highway machine such as a wheel loader, if the powertrain and the electro-hydraulic implement controller can talk to each other then the transmission “knows” the position of the implements. This means that if the bucket is high up in the air then the transmission shifts can be made smoother so as to increase load retention in the bucket. On the other hand, when the bucket is near the ground level, the transmission shifts can be more aggressive and thereby help in reducing the overall cycle time. This customer benefit in terms of reduced cycle times can only be achieved when the individual systems talk to each other and we can implement features that we couldn’t when only individual systems were electronically controlled. The key then is to integrate the various systems such that the overall machine performance is optimized. We are, in effect, designing a “system of systems”

The design of these complex machines is already a difficult challenge due to the complex interaction of various individual systems. We complicate it further by introducing electronics and the accompanying system behavior adaptability. To better understand why this is the case, let’s consider the example of a wheel loader. One of the most common applications for a wheel loader is the V-cycle where the loader penetrates into a pile, picks up gravel, backs up from the pile, reverses direction, approaches a truck, while raising the implement linkage, such that the linkage is in a position to dump when it approaches the truck. Once the loader has dumped the gravel in the truck, the loader backs up from the truck, lowers the linkage, reverses direction again, and approaches the pile, at which point the cycle repeats. This operation involves the complex interaction of the loader powertrain, hydraulics, implement linkage, and steering system. The net performance of the machine is a function of these systems and their interactions. Furthermore, the problem is essentially a multi-domain problem since there are different physical phenomena such as hydraulics and mechanics that interact with each other during this operation. These

interactions are highly nonlinear and dynamic, making them difficult to understand completely. When we introduce electro-hydraulics into this machine in the form of electronic control units that can essentially alter the behavior of each of these systems “on the fly”, we complicate this system even further. This significantly stresses the traditional development process used for designing these machines.

THE TRADITIONAL DESIGN PROCESS

The traditional development process for such machines focuses on individual systems and tries to optimize each system based on design parameters that are specific to that system. For example, one of the key metrics in the design of the powertrain system is the traction force delivered to the wheels. Sequential design processes are used to account for the fact that the machine function is a combination of individual systems. In this case, lift force is a combination of the hydraulic system and mechanical implements. This is handled as a series of analyses to achieve a balance between powertrain, hydraulics, and the implement linkage. These analyses are typically static, are backed with heuristics based on experience, and do not take into account the dynamic nature of the interactions between these systems. During the traditional design process, the machine level performance targets are broken down into respective targets for systems and components and typically it is difficult to determine how a change in an individual system can affect the overall machine performance. As we discussed before, these are highly nonlinear systems belonging to multiple domains that interact with each other, making prediction and optimization of machine performance a difficult task. This also makes it difficult to ensure that the machine will meet the customer needs.

Using the traditional process, a typical way to ensure that the design will work well in the dynamic environment that the machine usually operates under, is to build physical prototypes of these machines. The physical prototype is typically the first time that the entire machine is tested as a whole. If the machine does not work as intended, a significant amount of rework is required for the components and systems designed earlier. This process has several disadvantages. First, this is extremely costly since different variants of the hardware have to be manufactured and sequentially improved in order to make sure that the machine meets the performance requirements. Second, the different design alternatives are very much constrained by available hardware. The result is optimum design of the individual components which may or may not result in the optimal machine performance. Third, this hardware intensive, iterative process is extremely time-consuming since the engineers must wait for the redesigned hardware before they can re-test the overall machine.

MODEL-BASED DESIGN

In order to address these issues there is an increased focus on employing modeling and simulation during the

design process. Using a simulation model of the various machine systems allows the dynamic performance of these systems to be verified in the absence of physical hardware [1,2]. Typically this is done today using domain specific modeling and simulation tools. For instance, if the goal is to design a hydraulic system, then commercial or in-house hydraulic simulation tool is used. Similarly, if the goal is to optimize the linkage geometry a commercial or in-house mechanical simulation tool is used. Since the design and development tools used in such a process tend to be domain specific, the ability to reuse work performed in one domain for analysis in another domain is limited. For example, if a hydraulic system design is to be evaluated at the machine level, it is difficult, if not impossible, to reuse the model used during the mechanical design of the implement linkage. Instead the hydraulic designer has to recreate an equivalent linkage in the hydraulic design environment to test the hydraulic system. This can lead to transcription errors when data and results from one domain are used for work in a different domain. This approach also has the drawback that the interactions between the various systems cannot be assessed. This significantly limits the optimization of the overall machine performance and the amount of machine performance verification we can achieve in absence of hardware. Co-simulation between the various domain specific tools can be used to work around this, but that has its own challenges in terms of trying to make two different simulation solvers work together, simulation speed, and other factors.

To achieve the real benefit of Model-Based Design, what is needed is an integrated environment in which the machine design can be verified and validated. Such an environment will allow multiple domains such as hydraulics, mechanics, electronics, etc. to be modeled and simulated in a single environment. This means that hydraulic designers can reuse the models used by the mechanical designers to test their own design, and thereby save time and reduce errors. Having multiple domains modeled in the same environment allows engineers to easily evaluate interactions between the domains enabling rapid design iterations. The iterative process provides a method for optimizing the entire system simply by changing the various system design parameters and re-running the simulation. This not only allows us to identify design errors earlier and address their effect on machine performance, but also leads to better traceability in terms of how the machine performance requirements relate to different design decisions. Using integrated model-based methods, we only need to build machine hardware once we have verified machine performance through simulations, which saves the cost and time involved in building multiple prototypes.

This paper describes the use of the MATLAB® and Simulink® environment [3,4] to address some of the issues described above by using the design of an electro-hydraulic implement system, similar to the one shown in Figure 1, as an example. The MATLAB and Simulink environment is used throughout the design

process since it provides high-level formalisms such as SimMechanics [5] and SimHydraulics [6] to support system level modeling of the implement system. The ability to model multiple domains such as hydraulics and mechanics in the same environment has many benefits. It brings the design process much closer to the realization before committing to an implementation, and uncovers incompatibilities and interactions between systems while the system is still in its conceptual form and can be easily modified. This also allows experimenting with different design alternatives during the conceptual design stages, and detailed implementation effects can be added as the need arises. The paper is organized as follows: Section 2 introduces the overall electro-hydraulic implement system and then focuses on the behavioral modeling and analysis of the mechanical linkage. The hydraulic system is a key component of the overall system and it is introduced in Section 3 along with some simulation results. Section 4 discusses the reuse of the models developed during mechanical and hydraulic design stages to create an overall electro-hydraulic implement system model and presents simulation results of the overall system performance. The key conclusions are then outlined in Section 5.

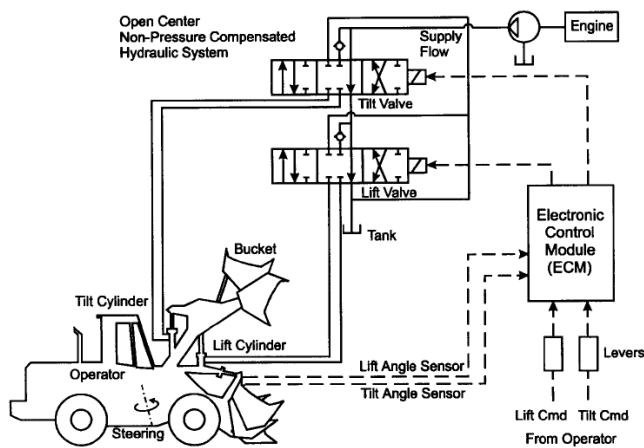


Figure 1 Example of a Wheel Loader Electro-Hydraulic System [7]

ELECTRO-HYDRAULIC IMPLEMENT SYSTEM AND MODELING AND ANALYSIS OF THE IMPLEMENT LINKAGE

OVERVIEW OF THE ELECTRO-HYDRAULIC IMPLEMENT SYSTEM

In this paper we will focus our attention on a wheel loader electro-hydraulic implement system similar to the one described in [7] and as shown in the schematic diagram in Figure 1. The overall system consists of the implement linkage which is attached to the wheel loader non-engine end-frame. The linkage has two degrees of freedom which are controlled by the lift and tilt cylinders and actuated by hydraulic power. The engine drives a pump, the output of which is regulated by the tilt and lift valves such that the desired motion of the linkage is obtained. The wheel loader operator indicates his desired motion by actuating the lift and tilt levers from

the wheel loader cab, the motion of which is read by the electronic control module. Software control algorithms are used to calculate the appropriate commands to the hydraulic pump and valves such that the motion desired by the operator is obtained. For the purposes of this paper we will focus on the lift motion only, although the concepts presented are generic enough and can be extended to the modeling and analysis of tilt cylinder motion. In the next section we will discuss the implement linkage in more detail.

MODELING THE IMPLEMENT LINKAGE

There are a number of implement linkages used on a wheel loader depending on the application needs, but arguably none is as wide spread in its use as the so-called Z-Bar linkage illustrated in Figure 2. This linkage configuration is simple and at the same time yields excellent breakout force which are some of the reasons for its popularity. The main components of a Z-Bar linkage are the lift arm, the lift cylinder, the tilt cylinder, the lever, the link, and the bucket. These are interconnected through 9 pin joints such that the tilt cylinder, lever, and link resemble the letter Z. The linkage attaches to the wheel loader frame through pins A and Y respectively. As can be seen from Figure 2, even though there are nine pin joints, the linkage only has two degrees of freedom, viz., lift motion and tilt motion.

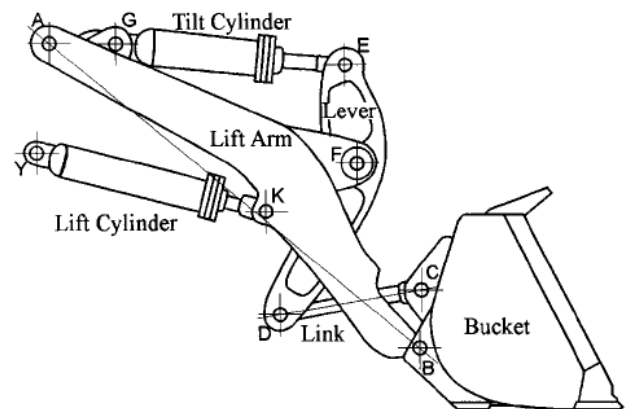


Figure 2 Wheel Loader Z-Bar Linkage [7]

As discussed before, because of the need for an integrated environment for design and analysis of the electro-hydraulic linkage we will use the MATLAB and Simulink environment [3,4] and the family of products in the Physical Modeling area. These products allow users to create models that reflect the physical nature of the system using a graphical language with physical connections that closely mirrors the language of the engineering domain. The products in this family are SimDrivelink [8], SimHydraulics [6], SimMechanics [5], and SimPowerSystems [9]. Each product interfaces with the rest of the Simulink family of products through signals, sensors, actuators, and sources.

We will use SimMechanics to model the dynamics of the Z-Bar. SimMechanics models and simulates 3D mechanical systems, also known as multibody systems.

Users can create mechanical systems by connecting Body and Joint - blocks. The body blocks are used to specify the mass, inertia matrix, and coordinate systems. The joints provide the relative motion between the body blocks and a variety of different joints are provided in the form of pre-built libraries. In order to actuate the motion of these joints sophisticated force laws can be created using sensors and actuators and Simulink blocks. Key linkage information such as pin locations, mass and inertia properties of the linkage can be obtained from a CAD program and used to populate the SimMechanics model parameters. The model of the Z-Bar is shown in Figure 3.

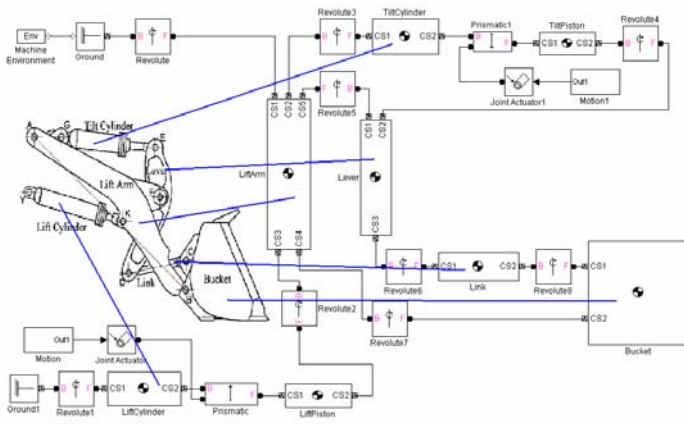


Figure 3 SimMechanics Model of the Z-Bar Linkage

The model contains an image of the Z-bar linkage and the remainder of the model contains SimMechanics blocks to create the simulation model. The parts in the image correspond to blocks in the diagram. The correspondence of some of the parts is indicated with lines from the image to the block diagram model. The model consists of Joint blocks (prismatic and revolute in this case) and Body blocks. The rigid body blocks contain the mass and inertia of the parts as well as the connection points represented with coordinate systems. In the upper left and lower left are ground blocks that represent fixed attachment points. Between the upper left ground block and the LiftArm is a revolute joint (Revolute). The LiftArm is then connected to the LiftCylinder through another revolute joint (Revolute2). The LiftCylinder consists of a piston, a prismatic joint, and a cylinder. The LiftArm is also attached to the TiltCylinder through another revolute joint (Revolute3). The TiltCylinder consists of a cylinder, prismatic joint, and piston. The LiftArm is attached to the Lever through Revolute5. The Lever is attached to the TiltPiston through Revolute4. The Lever is attached to the Link through Revolute6. The Link is then attached to the Bucket through Revolute8. The Bucket is attached to the LiftArm through Revolute7. Through these Body, Joint, and Ground blocks, we are able to create the model that we can use for simulation, design, and analysis.

SimMechanics also offers a built-in visualization for the linkage so that the user can visualize the 3-D motion of this linkage. Figure 4 shows a screen capture of the Z-Bar linkage visualization. The lift arm pivot pin is

assumed to be the origin of the coordinate system and the various components of the lift arm assembly are represented by equivalent ellipsoids based on the inertia tensor specified in the lift arm component data. The lift and tilt cylinders are represented by the prismatic joints in the model and can be seen as translational motion in the visualization. If a more realistic visualization is desired, then a virtual reality modeling language (VRML) file can be interfaced to the model as well.

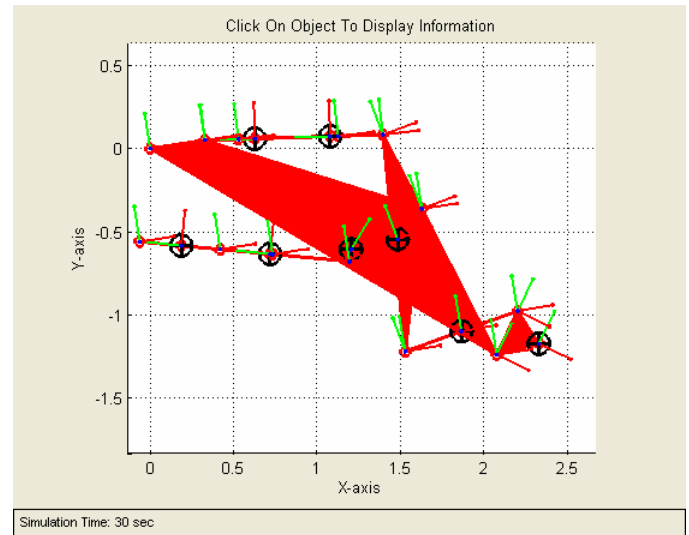


Figure 4 3-D Visualization of the Z-Bar Linkage

ANALYSIS OF THE IMPLEMENT LINKAGE

Once a model of the Z-Bar linkage is built we can analyze the linkage design by pacing the model through various usage scenarios. The mechanical designer who designs the linkage cares about various kinematics aspect of the linkage such as lift height, etc. which are easily obtained from this model. Similarly by positioning the lift cylinder at various discrete positions from its minimum to maximum displacement we can obtain the position of the bucket tip to establish the working envelope of the Z-Bar linkage. In SimMechanics this involves prescribing a motion profile to the prismatic joint that represents the lift cylinder and then running the simulation to determine the position of the bucket.

For the purpose of this paper we will focus on developing a load-capacity chart of the Z-Bar linkage wherein the linkage is positioned at discrete points, a force equivalent to full system pressure is applied at the lift cylinder, and the force required in the bucket to balance this applied cylinder force is calculated. This is essentially a statics problem where the objective is to determine the load in the bucket that can be carried when the lift cylinder is positioned at different extensions and full system pressure is applied at the cylinder. Simulink offers a sophisticated suite of control design tools [10,11] which can be used to perform this analysis.

Simulink Control Design [11] is a product that is used to design and analyze control systems built in Simulink.

One of the available core features is the ability to search for steady state operating points based on user specified set points. This process is known as “trimming” a model in the aerospace industry. These operating points are typically used to linearize a Simulink model to design a compensator. In the case where a Simulink model is continuous (hybrid operating point searches are supported in Simulink Control Design) the operating point search routine works on the differential equation below:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= g(x(t), u(t)) \end{aligned}$$

where $x(t)$ are the states of the model, $u(t)$ are the model inputs, and $y(t)$ are the model outputs. The goal of the operating point search is to find a set of $x(t)$ and $u(t)$ that meet the following conditions:

$$\begin{aligned} \dot{x}(t) &= 0 \\ y(t) &= y_d \end{aligned}$$

where the system is at rest and the output of the model is a desired set point.

In this application, the goal is to find the load in the bucket, $u(t)$, that keeps the system at rest at a specified lift cylinder position, $y(t)$. To reduce the complexity of the search process the problem is broken up into two problems. One, search for the condition $y(t) = g(x(t), u(t)) = y_d$ while not enforcing the condition that the system is at steady state. This will give a set of states x_0 that moves the machine to the desired lift cylinder position. Two, search for the steady state condition $\dot{x}(t) = f(x(t), u(t)) = 0$. Since the states x_0 are known they can be fixed reducing the search for the load $u(t)$ on the bucket to ensure the steady state condition.

To facilitate this process Simulink Control Design provides a GUI that allows the enforced conditions to be easily specified. In addition, these operations can be automated programmatically using MATLAB scripts. When scripted the following curve (Figure 5) can be generated that relates a desired steady state lift cylinder extension to steady state load in the bucket.

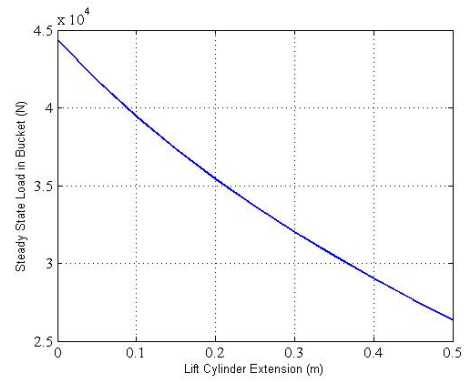


Figure 5 Load Capacity Chart for the Linkage

As a result of this analysis the mechanical designer can ascertain whether the design meets the machine requirements. In a traditional process the key mechanical data such as pin locations, mass and inertia properties of linkage, etc. would be passed on to the hydraulic system designer so that the hydraulic system can be designed per requirements.

MODELING THE IMPLEMENT HYDRAULIC SYSTEM

The implement hydraulic system is an electro-hydraulic system and consists of a pump driven by an engine which provides the fluid power, electro-hydraulic valves for the lift and tilt circuits that divert the pump flow to the circuit that needs it, sensors to sense the position of the lift and tilt cylinders in case closed loop control of the implement function is desired, and electronic joysticks which provide an indication of operator desire for lift and tilt movement. As can be seen, this system involves a combination of hydraulic, electrical, and electronic phenomena and in order to understand the entire system it is essential that we use one environment. As discussed before, we will use SimHydraulics to model the implement hydraulic system. SimHydraulics models and simulates hydraulic systems inside the Simulink environment. It provides blocks for typical valves, pipes, accumulators, pumps, and cylinders. The blocks have hydraulic connections that are connected together analogous to the physical hydraulic system. Further, the blocks use schematic symbols commonly used in the fluid power industry and thus the model visually resembles the hydraulic circuit.

A simple model of the implement hydraulic system is shown in Figure 6. As can be seen, the basic components of SimHydraulics such as pump, valve, etc. resemble the schematic symbols used in hydraulic circuits and thus make it easy for a user to interpret the model and understand it. The pump in the hydraulic system produces fluid flow which is then diverted by the 4-way valve to the appropriate cylinder port depending on whether the linkage is being raised or lowered. Once we have a hydraulic system model we can apply boundary conditions to this model to represent the points of interaction with the rest of the electrical and mechanical components of an implement system. In this

case, the pump is driven by a mechanical speed reference input which represents the engine driving the pump. To represent the mechanical interactions from the linkage and the rest of the machine we can apply simplified loads in the form of mass, spring, and damper components, as is shown in Figure 6. The response of the hydraulic circuit is as shown in Figure 7. This is similar to the traditional method of analyzing these systems. While this certainly can be done using SimHydraulics, it does not fully capture the dynamic interactions that occur between the two systems when the linkage moves up and down and the amount of load in the bucket varies. In order to capture the true implement system behavior we need to connect the SimMechanics implement linkage model to the SimHydraulics implement hydraulics model, which is discussed in the next section.

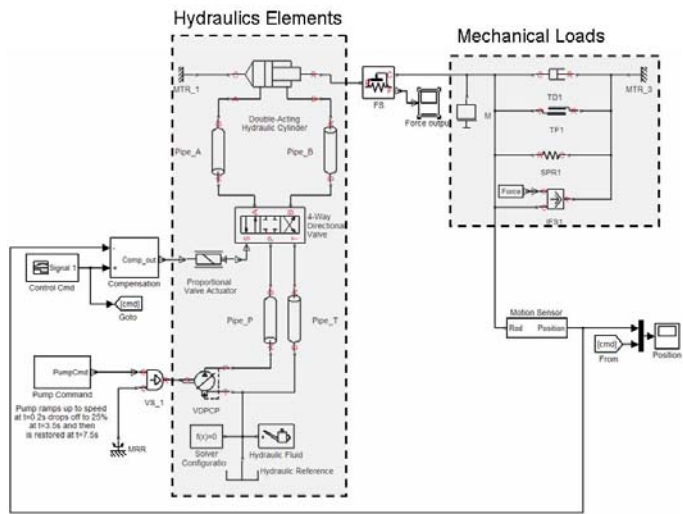


Figure 6 Implement Hydraulic System Modeled Using SimHydraulics

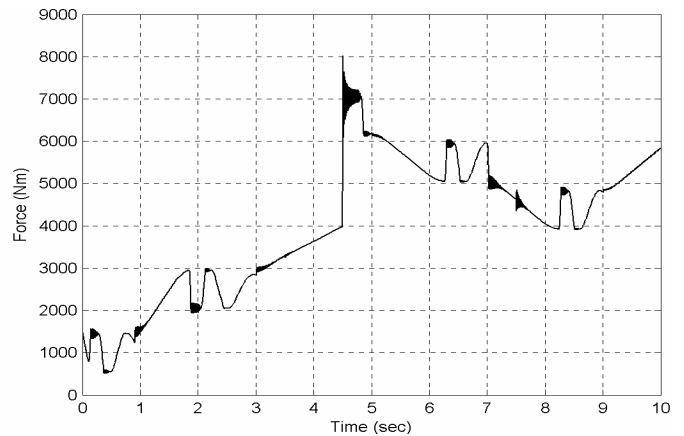
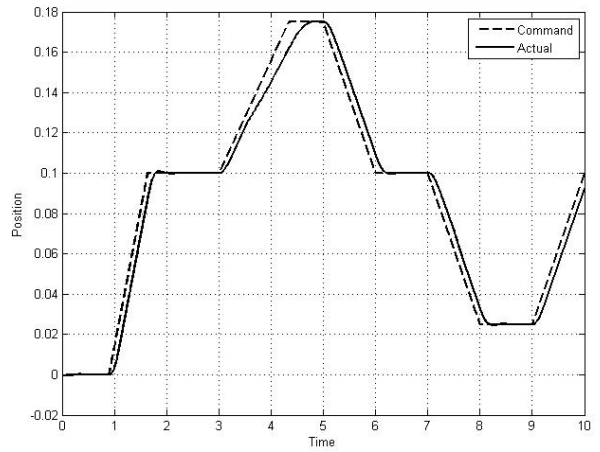


Figure 7 Hydraulic Circuit Responses: Cylinder Position and Output Force

IMPLEMENT HYDRAULIC SYSTEM SIMULATION AND VERIFICATION

The integrated MATLAB and Simulink environment facilitates multi-domain simulation through the use of Simulink as the common link between different domains. Thus, the models developed by the mechanical and hydraulic designers can be reused and this in itself saves time and avoids transcription errors. Further, having both the mechanical and hydraulic domains in the same environment facilitates design iterations where parameters belonging to both domains can be varied in order to optimize the overall system and not just the individual mechanical or hydraulic systems.

The integrated electro-hydraulic implement system model is shown in Figure 8. As can be seen, the hydraulic system output results in a cylinder force that gets applied to the prismatic joint that represents the cylinder and thus provides the interconnectivity desired for multi-domain simulations. In this model the pump displacement is controlled by a simple behavioral model of the implement control unit which also controls the lift valve displacement and thereby the lift cylinder movement. This simple behavioral model can be augmented in the later design stages with a more detailed controller model using Simulink's control design capabilities. For the purposes of this paper, the

controller does not account for any closed loop performance requirements.

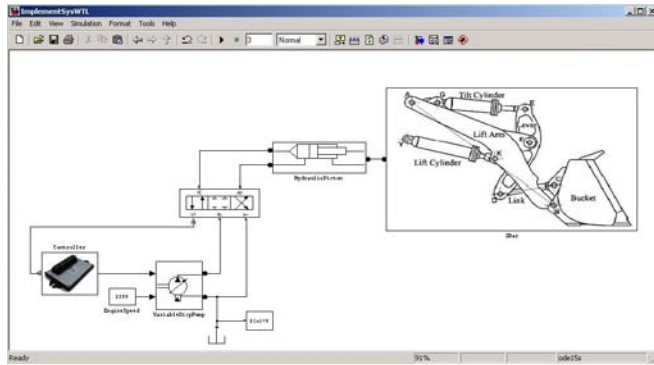


Figure 8 Electro-Hydraulic Implement System Modeled in the Simulink® Environment

Once we have the hydraulic, mechanical, and controller domains in the same model we can simulate the entire system to understand how the electro-hydraulic system would behave. The simulation result from one such run is shown in Figure 9, where the operator issues a lift up command and expects the linkage to accelerate rapidly to its maximum velocity. The lift cylinder position and velocity can be seen in Figure 9, and similarly we can look at other system variables such as lift force generated, etc. We can then compare these simulation results with the implement system requirements to determine whether our system design meets the requirements or not. Thus, we can achieve some early machine level testing in the absence of hardware and address some of the issues we discussed in section 1.

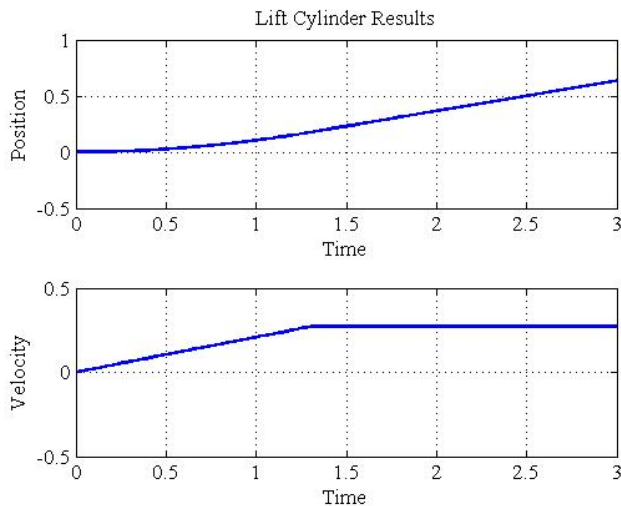


Figure 9 Electro-Hydraulic Implement System Simulation Results

Once we determine that the system concept meets our requirements we can proceed with detailed design, whereby we can add detailed effects such as pump swash plate actuator dynamics, friction at pin joints and cylinders, closed loop controller algorithm, etc. and ensure that the design still meet the system

requirements. The MATLAB and Simulink environment facilitates this transition to a detailed model very well and this will be discussed in more detail in a future paper.

CONCLUSIONS

This paper discussed some of the issues facing manufacturers of earthmoving and agricultural equipment today and how an integrated environment for modeling and simulating these machines can provide a potential solution for these issues. The integrated modeling environment allows for the reuse of design information through various stages of the machine development process, time savings and reducing transcription errors. By simulating multiple domains in the same environment, the entire machine behavior and the interactions between the various systems can be understood allowing the systems to be optimized to meet the machine level requirements. This is achieved in the absence of hardware to evaluate design options, resulting in a significant amount of time and cost savings. The same model can then be used and refined for a variety of tasks carried out in a typical development process as different levels of design details become available. The net benefit of using an integrated environment to model and simulate the machine behavior prior to building hardware is that we can build the right machine, on time and within budget.

REFERENCES

1. A.J. Koivo, M. Thoma, E. Kocaoglan, and J. Andrade-Cetto. Modeling and Control of Excavator Dynamics During Digging Operation. *J. of Aerospace Eng.*, pp. 10-18, 1996.
2. P. Grant, J. Freeman, R. Vail, and F. Huck, "Preparation of a Virtual Proving Ground for Construction Equipment Simulation", *Proceedings of DETC*, 1998.
3. The MathWorks Inc., "Using MATLAB," Version 7.2, The MathWorks Inc., Natick, MA, March, 2006.
4. The MathWorks Inc., "Using Simulink," Version 6.4, The MathWorks Inc., Natick, MA, March, 2006.
5. The MathWorks Inc., "SimMechanics User's Guide," Version 2.6, The MathWorks Inc., Natick, MA, March, 2006.
6. The MathWorks Inc., "SimHydraulics User's Guide," Version 1.0, The MathWorks Inc., Natick, MA, March, 2006.
7. U Pinsopon, T Hwang, S Cetinkunt, R Ingram, Q Zhang, M Cobo, D Koehler and R Ottman, "Hydraulic actuator control with open-centre electrohydraulic valve using a cerebellar model articulation controller neural network algorithm", *Proc Instn Mech Engrs.*, Vol 213, Part I, pp. 33-48, 1999.
8. The MathWorks Inc., "SimDriveline User's Guide," Version 1.3, The MathWorks Inc., Natick, MA, March, 2006.

9. The MathWorks Inc., "SimPowerSystems User's Guide," Version 4.2, The MathWorks Inc., Natick, MA, March, 2006.
10. The MathWorks Inc., "Control Systems Toolbox User's Guide," Version 7.0, The MathWorks Inc., Natick, MA, March, 2006.
11. The MathWorks Inc., "Simulink Control Design User's Guide," Version 2.0, The MathWorks Inc., Natick, MA, March, 2006.

CONTACT

SAMEER M. PRABHU

Sameer has over ten years of experience applying The MathWorks' products in various application areas. As a Principal Applications Engineer in the Detroit, Michigan office, Sameer manages a team of applications engineers focused on working with customers in the automotive and commercial vehicles industry to address the systems integration challenges posed by increased adoption of electronics in these industries. Prior to joining The MathWorks, Sameer engaged in the R&D of complex control systems through his work at Visteon, Caterpillar, and Tata Motors. The results of his work have been documented through publications and United States patents. Sameer graduated from University of Bombay with Bachelors in Mechanical Engineering and received his Ph.D. in Mechanical Engineering from Duke University in the area of robotic controls and artificial intelligence. He also holds an MBA from the University of Michigan. He can be reached at Sameer.Prabhu@mathworks.com

JEFF WENDLANDT

Jeff Wendlandt is the team lead and development manager for the Physical Modeling group. He has an expertise in robotics, control, and simulation algorithms and has worked in a number of industries. He received his BSME (1991) from the University of Illinois and his

MSME (1993), MA Math (1995), and Ph.D. (1997) from the University of California at Berkeley. His graduate work concentrated on multibody control and simulation, medical robotics, and integration algorithms for Lagrangian and Hamiltonian systems. He performed some of his research at MIT and Caltech during his graduate studies. He worked at Fermi National Accelerator Laboratory, Argonne National Laboratory, Mitsubishi Electric Research, and Boston Scientific. He can be reached at Jeff.Wendlandt@mathworks.com

JOHN GLASS

John joined The MathWorks in 2001. He is the development lead of the team focused on building tools for linear analysis and control design for systems built in Simulink. John received his Ph. D., Master, and Bachelor of Science in Mechanical Engineering from Purdue University. His research at Purdue University focused on the nonlinear system identification and control of internal combustion engines. He can be reached at John.Glass@mathworks.com

TOM EGEL

Tom has over 15 years of experience working with modeling and simulation tools for mechatronic system applications. As a Principal Applications Engineer in Detroit, Michigan, Tom supports customers as they use the MathWorks products for model-based design of controlled physical systems. Prior to joining The Mathworks, Tom 12+ years developing/supporting/selling physical modeling tools in the automotive/transportation industry. Specific work includes electrical design engineer at Texas Instruments, applications engineer with Analogy, Inc. (Saber), and technical marketing engineer at Mentor Graphics specializing in VHDL-AMS. Tom holds a B.S. in Electrical Engineering from Washington University in St. Louis and a B.A. in Physics from St. Louis University. He can be reached at Tom.Egel@mathworks.com

MATLAB, Simulink, Stateflow, Handle Graphics, Real-Time Workshop, and xPC TargetBox are registered trademarks and SimBiology, SimEvents, and SimHydraulics are trademarks of The MathWorks, Inc. Other product or brand names are trademarks or registered trademarks of their respective holders.