2007-01-4248

Model-Based Design for Off-Highway Machine Systems Development

Sameer M. Prabhu
The MathWorks, Inc.

ABSTRACT

The increased adoption of electronic controls in off-highway machines increases the complexity of typical machine systems and stresses the traditional process used to develop these machines. To address this issue design engineers are turning from the traditional design methods to Model-Based Design. By using models in the early design stages, engineers can create executable specifications that enable them to immediately validate and verify specifications against the requirements. These models also allow the machine designer to evaluate the complex interactions between mechanics, hydraulics, electronics and other physical phenomena and thereby detect design errors earlier when the cost to fix them is less. This paper presents a model-based approach for developing off-highway equipment machine systems. A dynamic model of the machine and the electro-hydraulic implement and propulsion system is developed and used to verify the overall machine behavior. The models are linked to the machine requirements and instrumented to check the simulation results to achieve verification of machine behavior against requirements in a formal way.

1. INTRODUCTION

MACHINE SYSTEMS DEVELOPMENT CHALLENGE

Electronic controls in off-highway equipment are growing rapidly as a result of regulatory requirements such as emissions restrictions and customer demands for increased machine productivity, uptime, and safety. The design of a typical off-highway machine is already a difficult challenge due to the complex interaction of various individual systems and it is complicated further by introducing electronics and the accompanying system behavior adaptability. To better understand why this is the case, let us consider the example of a wheel loader. One of the most common applications for a wheel loader is what is referred to as hopper charging wherein the loader acquires material from a stockpile, backs up from the stockpile, reverses direction, traverses the distance towards a ramp, and then moves up the ramp while raising the implement linkage, such that the linkage is in a position to dump at the end of the ramp. This operation involves the complex interaction of the loader powertrain, hydraulics, implement linkage, and steering system and the net performance of the machine is a function of these systems and their interactions. Further, this operation involves different physical phenomena such as hydraulics, mechanics, etc. that interact with each other, where the interactions are highly nonlinear and dynamic. When we introduce electro-hydraulics into this machine in the form of electronic control units that can alter the behavior of each of these systems "on the fly", we complicate this system even further. Developing such a machine on-time and under budget poses a significant challenge to the traditional machine development process.

THE TRADITIONAL DESIGN PROCESS

The systematic design and realization process for an off-highway machine is typically represented by a V diagram as shown in Figure 1 [1]. The left branch captures the decomposition of machine requirements into systems and subsystems that are specified and implemented at a detailed level. The right branch represents the realization of these systems and subsystems and their integration and test in the final machine.

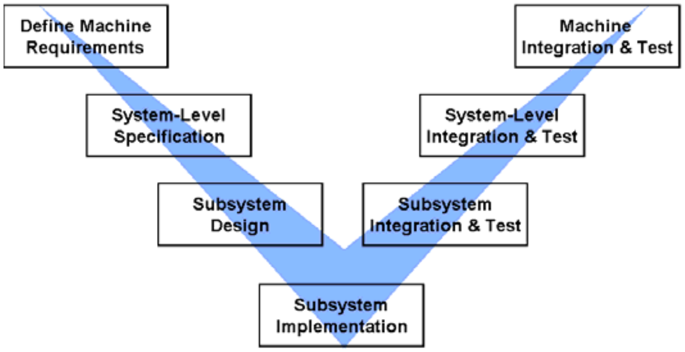


Figure 1 V Diagram of Product Development Process

In the traditional approach, engineering teams observe strict boundaries between their design activities and communicate by passing design documents back and forth between various stages and between different disciplines (e.g., hydraulics designer vs. mechanical designer, etc.). As a result, the traditional development process for such machines tends to take a sequential approach wherein each system is optimized based on design parameters that are specific to that system before moving on to the next system. For example, the powertrain system design is optimized for metrics such as traction force delivered to the wheels, gradeability, etc. whereas the mechanical linkage system is optimized for reach, breakout force, etc. The fact that the machine function is a combination of these individual systems is often addressed by static analyses, which 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. Thus when it comes time to develop the next generation of the machines it is difficult to rapidly iterate through system concepts to determine which concept best meets the machine requirements. This significantly affects the ability of off-highway machine manufacturers to innovate rapidly. Also, the lack of traceability from requirements to implementation makes it difficult to design the individual systems such that the overall machine performance is optimized while ensuring that the machine will meet the customer needs specified in the requirements document.

A typical way to ensure that a proposed machine concept will work 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 that the individual system performance is optimized which may or may not result in the optimal performance of the machine. 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 [2,3]. 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 or system for analysis in another domain or system is limited. For example, for a hopper charging cycle it is necessary that the hydraulics and the powertrain system work effectively in conjunction to meet machine requirements. In such a case it is difficult to reuse the model of the hydraulic system design along with the model of the powertrain system design. The machine engineer has to create a simplified machine level model using the behavioral characteristics of the individual powertrain and hydraulic system models in order to predict machine performance and to determine which of the various system concepts meet requirements. This can lead to transcription errors when data and results from one domain are used for work in a different domain. 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 allows the machine engineer to easily iterate through various system concepts and determine which of them meet the machine level requirements. Once the candidate system concept has been identified through this process, the hydraulic and mechanical designers can use the same models and add details to them pertaining to their own systems and continue to ensure that the overall machine still meets requirements. Having multiple domains modeled in the same environment allows engineers to easily evaluate interactions between the domains thereby 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 methods for Model-Based Design, 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 [4,5] to address some of the issues described above by using the example of a wheel loader performing a hopper charging cycle and the need to verify if a proposed system concept will meet the wheel loader requirements. The MATLAB and Simulink environment is used throughout the design process since it provides high-level formalisms such as SimMechanics [6] and SimHydraulics™ [7] to support system level modeling of the entire system. The paper is organized as follows: Section 2 discuses some typical wheel loader applications and then focuses on the hopper charging application in order to derive some requirements that drive the machine development process. Section 3 then presents the behavioral modeling and analysis of the implement hydraulic system and powertrain system which are combined into an overall vehicle model. Section 4 discuses how the wheel loader machine model is formally linked to machine requirements, such that we can verify if the wheel loader system concept meets program requirements. The key conclusions of the paper are then outlined in Section 5.

2. WHEEL LOADER APPLICATIONS AND PERFORMANCE REQUIREMENTS

OVERVIEW OF WHEEL LOADER APPLICATIONS

In this paper we will focus our attention on a wheel loader and will take a look at how the wheel loader applications drive machine requirements which in turn drives the machine systems development process. A wheel loader is a multi-function machine which can perform a variety of different application tasks depending on the tool connected to its implement linkage, which can be either a bucket, a set of pallet forks, etc. Typical applications for a wheel loader include, but are not limited to, V-cycle loading, hopper charging, backdragging, snow removal, hard bank excavation, etc. some of which are described below.

In V-cycle loading, 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 reaches 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. In this case, the ability of the loader to acquire material in the pile through a balanced combination of tractive force and lift force, and the ability of the loader to simultaneously steer, lift, and move in a coordinated fashion are some of the key drivers of performance characteristics.

In backdragging, the wheel loader bucket blade is used to push loose material where the loader travels in reverse gear such that the ground becomes relatively level due to the pressure applied by the bucket blade. In this application, the ability to apply downward pressure through the bucket while traveling in reverse is one of the key performance attributes. This involves small movements in the implement linkage system while the powertrain system operates in reverse gear. In snow removal, the loader is used as a dozer to collect and remove snow from a road, paved lot, etc. In this application the ability of the loader to run at sustained high speed conditions and the consequent demands on the cooling system is one of the key performance requirements. In hard bank excavation, the loader is used to break into an existing bank of solidified clay, sand, and rock mixture and move it to a different location or into a truck. In this case, the ability of the loader to dig into the hardened material and acquire it through a balanced combination of tractive force and lift force is one of the key drivers of performance attributes. In pallet movement, the forks at the end of the implement linkage are used to move pallets from one location to another location. In this case, preciseness of positioning the forks into the pallet tines and then placing the pallet at a certain location are some of the key drivers of performance attributes.

As can be seen from the above descriptions, each application drives a unique set of requirements for the wheel loader system design. Since the machine has to perform equally well in all of these applications it is necessary that we understand how a design change in the loader affects the performance of the loader in each of the applications. Since it would be impossible for us to cover each and every such application in detail in this paper we narrow our focus on one application, hopper charging, understand some of its key segments, identify some of the key performance attributes, and use the requirements based on these attributes to see how an integrated modeling environment can drive the development of the wheel loader machine systems and allow us to evaluate the machine performance in the absence of hardware.

HOPPER CHARGING

One of the most common applications for a wheel loader is what is referred to as hopper charging wherein the loader acquires material from a stockpile, backs up from the stockpile, reverses direction, traverses the distance towards a ramp, and then moves up the ramp while raising the implement linkage, such that the linkage is in a position to dump at the end of the ramp. Once the loader has dumped the material into a truck or a conveyor at the end of the ramp, the loader backs up, lowers the linkage, reverses direction again, and approaches the stockpile, 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. In a traditional design process, the development of these systems happens independently and the only time the system interaction issues are addressed is when the machine gets built and tested. In order to understand and analyze the performance of the machine it is essential that we take the different wheel loader applications such as the hopper charging cycle and break it down into individual elements so that we can quantify the performance of the various machine systems.

The hopper charging cycle can be broken down into the following segments:

a. approach pile (steer, move, and position linkage to penetrate pile)

b. dig and acquire material (keep machine pushed against pile, and lift and tilt movements to acquire load)

c. reverse from pile (steer and propulsion)

d. move forward and travel on level ground to approach ramp (propulsion)

e. travel up ramp (propulsion and lift)

f. dump (tilt)

g. reverse down ramp (propulsion and lift and tilt)

h. move forward and travel towards pile (propulsion).

This is by no means a unique breakdown of the hopper charging cycle, but just one of many ways in which the cycle can be decomposed into its elements. We can then take each of these segments and use them to quantify the performance of today's machine and also to develop the requirements for the next generation of machines. As an example, we can take the hopper charging cycle segment (travel up the ramp) and use it to develop requirements for the two systems involved in this element: propulsion and lift. A typical machine development process that is focused on individual systems will have a requirement that specifies the time taken for the propulsion system to reach its maximum speed along with a requirement for the time taken for the lift system to reach its maximum height. However, moving up the ramp in a hopper charging scenario involves both of these systems working in parallel. Since the engine can only provide a limited amount of power, the response times of the systems would be different when they work together as opposed to when they work in isolation. In this case we can develop a requirement that states the response time for the propulsion system when the lift system is active as well. Thus, by decomposing the overall cycle into its segments we can quantify overall machine performance. Further, during the machine development process for the next generation of machines we can set targets for these elemental performance metrics based on current and competitive machine performance under the same conditions, and other program requirements. Thus, these elemental performance metrics become traceable back to the machine application requirements. This allows us to start with requirements based on machine level applications, break it down into targets for individual and combined system operation, and then drive the machine design and development process to meet these targets.

For the purposes of this paper we will narrow our focus to the interactions that happen between the implement hydraulic and powertrain systems during the course of going up the grade only and not deal with the digging aspects of the hopper charging cycle. The performance requirements that arise out of the lift and propulsion system interactions during a hopper charging cycle are discussed next.

PERFORMANCE REQUIREMENTS

Given the above discussion, we can develop a test specification that creates scenarios to capture the wheel loader performance for these various segments. These test specifications can be used to test the current machine or competitor's machine to obtain the performance requirements for the wheel loader, which are stated as follows for both the individual system response and the coordinated system response:

Requirement 1: Lift System Response

With the linkage being at the lower most position where the lift cylinder is at minimum stroke with bucket loaded with rated load (5000kg), when the loader operator issues a step command to lift, the lift system should move through its full range of motion (the lift cylinder being at its maximum stroke at the end of the motion) in 6.5 seconds or less.

Requirement 2: Propulsion System Response

With the linkage being at the lower most position with the bucket loaded with rated load (5000kg), when the loader operator issues a step command to move, the loader should reach its first gear maximum speed of 4.35 mph (7kph) in less than 1.5 seconds.

Requirement 3: Simultaneous Lift and Propulsion on Level Ground:

With the linkage being at the lower most position with the bucket loaded with rated load (5000kg), when the loader operator issues a step command to move and lift at the same time on level ground (from rest for both the propulsion and lift system), the loader should reach its maximum speed of 4.35 mph (7kph) in less than 1.5 seconds.

Requirement 4: Simultaneous Lift and Propulsion on 12% Grade:

With the linkage being at the lower most position with the bucket loaded with rated load (5000kg), when the loader operator issues a step command to move and lift at the same time while the machine is at the bottom of a 12% grade (from rest for both the propulsion and lift system), the loader should reach its first gear maximum speed of 4.35 mph (7kph) in less than 2.0 seconds.

We will use these requirements to drive the machine systems development process for the wheel loader. For the purpose of this paper, we will consider the scenario where we are developing the next generation of the wheel loader and are evaluating a new transmission system concept which involves using a hydrostatic transmission coupled with a planetary gear train to form an Infinitely Variable Transmission (IVT). The IVT is combined with the existing implement hydraulic system and the goal of the machine development program is to meet or exceed the requirements stated previously.

In this scenario then, we are starting off with the top left corner of the V-process shown in Figure 1. We would like to verify whether the IVT based machine concept will meet our requirements. As discussed earlier, the traditional paper based development process makes it difficult to maintain this traceability between requirements and design, and to evaluate the effect of design changes on overall machine performance. In this paper we will discuss how models can be used throughout this process and requirements traceability and compliance can be ensured through the use of models without relying on hardware prototypes. In the next section we will take a look at the integrated environment that allows us to simulate the wheel loader to iterate through system concepts and determine if they meet the machine performance requirements or not.

3. WHEEL LOADER MACHINE SYSTEMS MODELING

As described earlier, the traditional design processes and tools are very good at optimizing individual system performance metrics, but the machine level performance is difficult to ascertain. In this case, the implement hydraulic system is designed to meet performance specifications such as time to move through the full range of lift motion, etc. whereas the powertrain system is designed to meet performance requirements such as time to get to maximum speed in first gear on level ground and when going up a grade, etc. Again, in the case of hopper charging what matters is the lift time when the machine is moving up the ramp, which requires the combined interactions of the implement hydraulic and propulsion systems. The lack of an integrated environment to verify machine performance prohibits rapid iterations through system concepts to determine the system concept that meets machine requirements.

For this paper we will use MATLAB and Simulink [4,5] and the family of products in the physical modeling area as the integrated environment for design and analysis of the wheel loader. 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 SimMechanics [6], SimHydraulics [7], SimDriveline [8], and SimPowerSystems [9]. This 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, while detailed implementation effects can be added as the need arises. We will discuss next how the different machine systems are developed using this integrated environment.

IMPLEMENT HYDRAULIC SYSTEM

The implement hydraulic system consists of a Z-Bar linkage coupled to electro-hydraulic implement system [10] which in turn 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 flow to the circuit that needs it, sensors to sense the position of the lift and tilt cylinders, and electronic joysticks which provide an indication of the loader operator's desire for lift and tilt movement. SimMechanics is used to model the dynamics of the Z-Bar as shown in Figure 2.

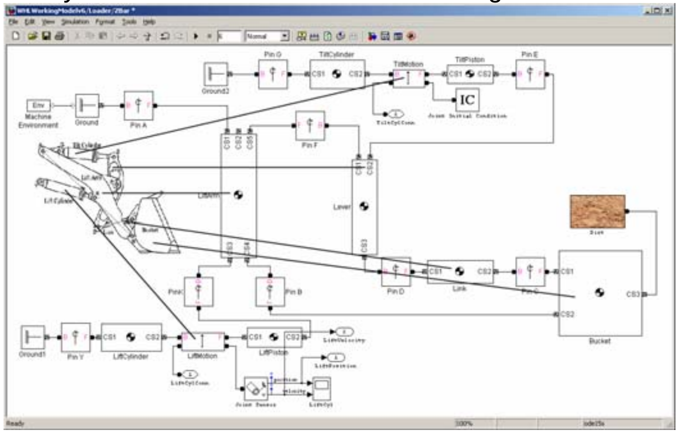


Figure 2 SimMechanics Model of the Z-Bar Linkage

The hydraulic system is modeled using SimHydraulics which provides blocks for typical cylinders, valves, pumps, etc. where the blocks have hydraulic connections that are connected together analogous to the physical hydraulic system. The blocks use schematic symbols commonly used in the fluid power industry and thus the model visually resembles the hydraulic circuit. The electro-hydraulic implement system model is shown in Figure 3. The hydraulic system output results in a cylinder force that is applied to the prismatic joint that represents the cylinder and thus provides the interconnectivity 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 the control design capabilities in Simulink.

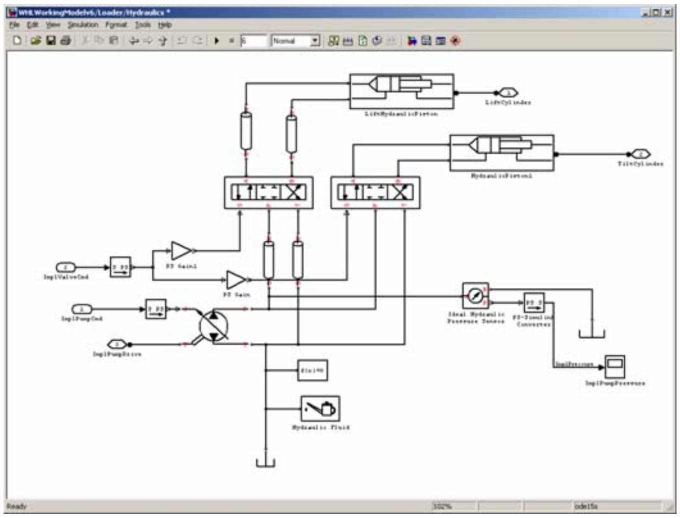


Figure 3 Electro-Hydraulic Implement System Modeled in the Simulink® Environment

Once we have the hydraulic, mechanical, and controller domains in the same model we can simulate the implement hydraulic system to understand how it would behave. The simulation result from one such run is shown in Figure 4, 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 is shown in Figure 4, 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 as stated in requirement 1 in section 2, and verify that the proposed implement system concept meets the machine requirements. This allows us to achieve early machine level testing in the absence of hardware and ensure that the implement system concept meets the machine requirements specified. The details of the modeling and simulation aspects are not included here since they are documented in another SAE paper [11].

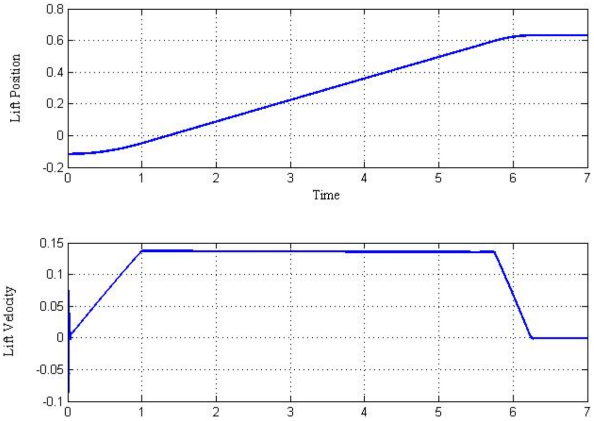


Figure 4 Electro-Hydraulic Implement System Simulation Results

POWERTRAIN SYSTEM

As mentioned previously, an infinitely variable transmission (IVT) is being considered as a potential solution for the loader propulsion system. An IVT consists of a hydraulic pump and motor (the hydrostatic part) connected with a planetary gear train. The machine speed is electronically controlled by controlling the pump and/or motor displacements. SimHydraulics is used to model the pump, motor, and associated hydraulic circuitry, as shown in Figure 5.

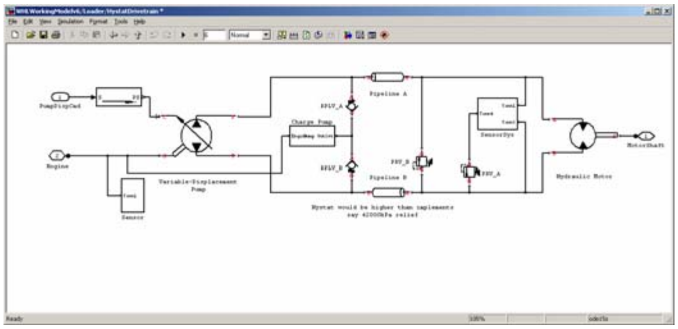


Figure 5 Hydrostatic Transmission Modeled in the Simulink® Environment

The hydrostatic transmission is then coupled to a planetary gear train to complete the IVT model. The IVT is coupled to the machine driveline and the tires which propel the machine forwards and backwards. SimDriveline can be used to model the planetary gear train, the tires, and the longitudinal machine dynamics as shown in Figure 6.

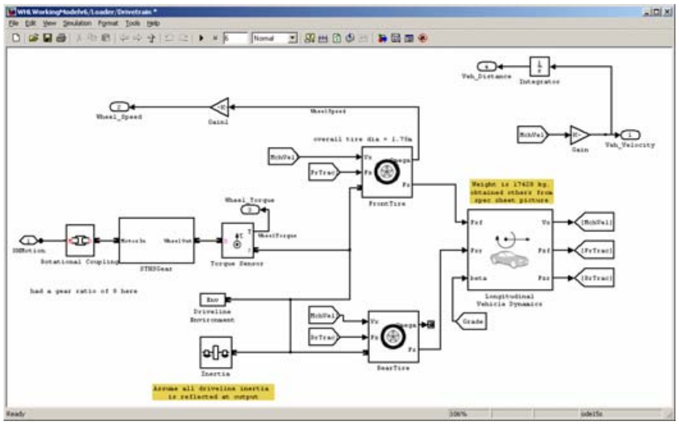


Figure 6 Wheel Loader Driveline, Tires, and Machine Dynamics Modeled in the Simulink® Environment

Once we have a model of the wheel loader propulsion system we can simulate the condition where the operator issues a step command to the propulsion system to determine if the system meets the requirement number 2, specified in section 2.

MACHINE MODEL

Now that we have the powertrain and the implement hydraulic system model we can combine these into the wheel loader machine systems model. The machine systems model is as shown in Figure 7.

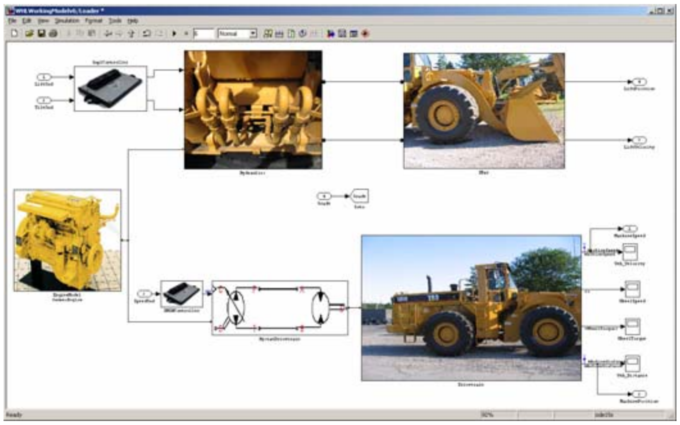


Figure 7 Wheel Loader Model

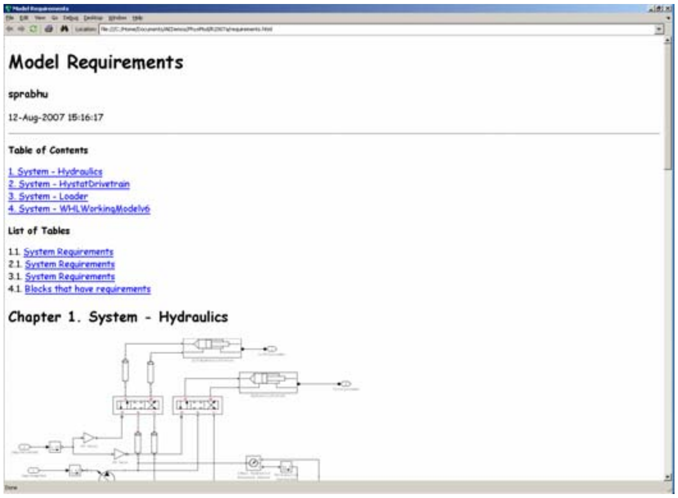
Using the machine level model of the wheel loader we can start simulating the scenarios corresponding to the requirements 3 and 4 stated in section 2. However as the number of machine requirements increase it is inconvenient to manually compare the results of each simulation to the requirements to verify compliance with requirements. This process becomes even more time consuming if we want to verify compliance with the entire set of requirements every time a design or concept change is made. This in turn inhibits rapid design iterations and causes the machine systems engineer to settle on a suboptimal design and also lengthens the machine development process. It would be nice if a mechanism could be created in the model which reported on which requirements were met and which were not met after a simulation run. This would enable rapid iteration of concepts and design changes to determine if all of the requirements are being met. We will show one way to address this issue in the next section.

4. VERIFICATION AND VALIDATION OF WHEEL LOADER MACHINE SYSTEM CONCEPTS

Our objective is to verify that the IVT based wheel loader concept meets the requirements specified in section 2. If we can test the system concepts against those requirements we can identify errors early in the development process when they are less expensive to fix and optimize the overall machine performance. This paper accomplishes this objective in a two step approach. First, the wheel loader model used so far and its various subsystems are associated with the respective requirements that drive the design of the loader and its subsystems. Second, test cases are derived from the high level requirements (and linked to the respective requirements) that are then executed on the simulation model to verify that the requirements are being met. Establishing such a formalized approach to testing the design is important to demonstrate that the system concept meets the loader requirements and allows us to ensure that there are no design errors early in the concept stage. Further, these links to requirements and test cases can be reused throughout the design and development process.

The loader model developed so far is actually an executable specification for the system to be designed. Running (or executing) the model tells us how the system will perform and if the system will meet the desired requirements. It is essential that all the elements in this specification are associated with requirements so that there are no redundant elements in the specification and that all the requirements are addressed in the specification so that there are no requirements that are left uncovered.

Simulink Verification and Validation [12] allows the establishment of a two way link between each element of the model (subsystem, blocks, state transitions, etc.) and a requirement document. The requirement document can be in a textual format, or commonly used file formats such as HTML. The two way link provides traceability from the model to the requirement document and vice-versa and is critical in ensuring that each element of the model specification has an associated requirement and that each requirement is realized somewhere in the model specification. To further formalize this traceability, we can automatically generate a HTML report which documents which of the subsystems in the model are associated with requirements for documentation purposes as shown in Figure 8. The subsystems in the model that are associated with requirements can be highlighted automatically as well, which helps in visually identifying which subsystems are not associated with requirements and thus can be tagged for further investigation.



Model Requirements

sprabhu

12-Aug-2007 15:16:17

Table of Contents

1. System - Hydraulics
2. System - HystatDrivetrain
3. System - Loader
4. System - WHLWorkingModelv6

List of Tables

1.1. System Requirements
2.1. System Requirements
3.1. System Requirements
4.1. Blocks that have requirements

Chapter 1. System - Hydraulics

Figure 8 Requirement Report for Model

Once we have associated requirements with the model and established traceability between the two it is necessary to ensure that the model specification complies with the requirements. To test the model specification against the requirements a test harness is established in Simulink consisting of test cases and verification blocks as shown in Figure 9. Each test case is linked to its associated requirement. Further, each test case is also associated with specific verification blocks so as to check for the expected output for that specific test case.

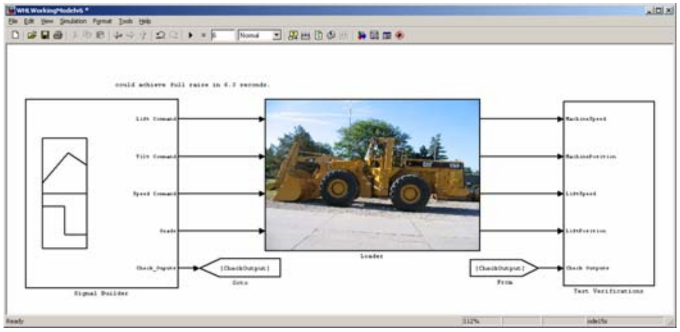


Figure 9 Wheel Loader Model with Test Cases and Verification Checks

Figure 10 shows the test case where the loader is asked to move on level ground while raising the linkage at the same time. Since requirement number 3 in section 2, deals with the expected behavior of the machine in this condition we associate this test case with requirement number 3 in the requirements document and also to the appropriate assertion blocks in the "Test Verifications" subsystem shown in Figure 9. On the bottom right hand side of Figure 10 there is a pane that shows the link to the requirements document and in the top right pane is the link to the assertion blocks.

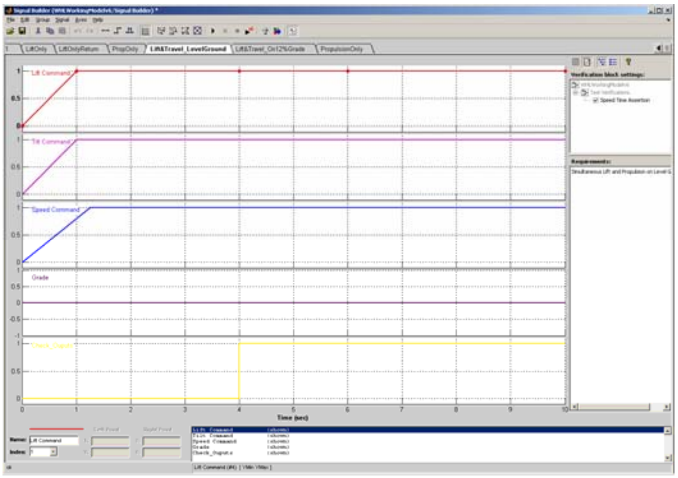


Figure 10 Defining Test Cases and Associating with Requirements using Signal Builder

The requirement states that the wheel loader should achieve its maximum speed of 4.35mph in less than 1.5 seconds. The "Test Verifications" subsystem logs the time it takes for the wheel loader to get to 4.35 mph during the simulation and then compares this to the requirement. When we execute the test case, we find that the loader takes 1.3 seconds to get to 4.35mph. Since this satisfies the requirement, the simulation proceeds to completion, as shown in Figure 11.

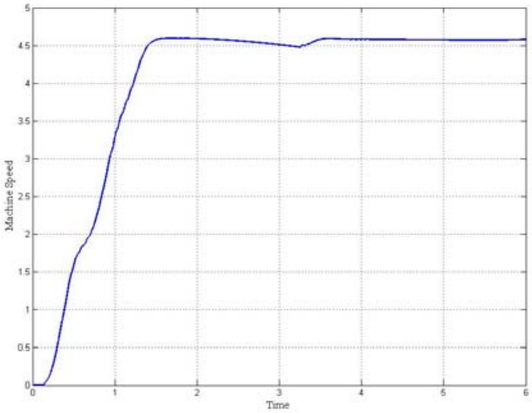
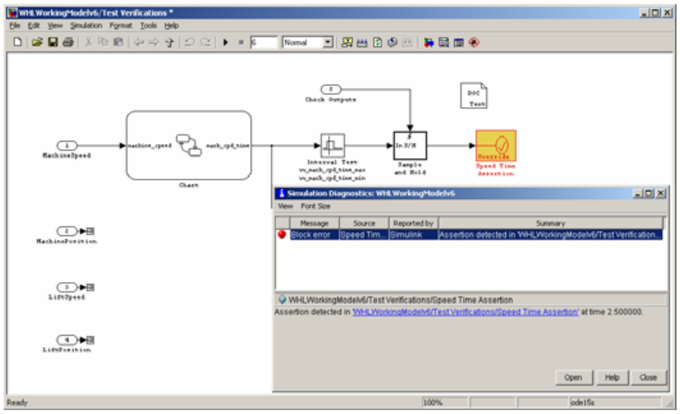


Figure 11 Wheel Loader Simulation Result

Since the wheel loader system concept meets requirements when the loader is commanded to move and lift on level ground, we can then take the next test case which involves the loader simultaneously moving and lifting on a 12% grade. The requirement states that the wheel loader should achieve its maximum speed of 4.35mph in less than 2.0 seconds. If this requirement is met then simulation will proceed as normal. However, if the time obtained from the simulation is greater than 2.0 seconds the assertion block throws off an error and stops simulation, as shown in Figure 12. At this point, the systems engineer can investigate what changes need to be made to the system concept so that this requirement can be met. Examples of such changes include, but are not limited to, changing the torque rise characteristic of the engine, changing the dynamic response of the IVT pump, etc. Having a model specification of the various systems allows us to easily investigate these changes and identify the best one that meets the machine level requirements for further design and development without the need for prototype hardware and the associated time and cost impact.



Message	Source	Reported by	Summary
Block error	Speed Tim...	Simulink	Assertion detected in 'WHLWorkingModelv6/Test Verification...

WHLWorkingModelv6/Test Verifications/Speed Time Assertion

Assertion detected in 'WHLWorkingModelv6/Test Verifications/Speed Time Assertion' at time 2.500000.

Figure 12 Assertion Error when Simulation Results Don’t Meet Requirements

Having this capability to create bi-directional links between the model and the requirements document, define the test cases corresponding to the requirements, capture the expected behavior in this case through the assertion blocks, and report if the simulation results meet the requirements or not, is a key enabler in getting early verification of whether the system concepts meet requirements and thereby rapidly iterate through the various proposed concepts to identify the best concept that needs to be pursued further. As we move further down the V development process in terms of designing the subsystems and components we can add the details relevant to subsystem and component design to the model. Since the requirements are linked to the model, we can simply run the simulation again with the test cases verify continued compliance with requirements. Reusing of the test and verification infrastructure in this way becomes a key enabler for early and continuous verification. Each time a change is made or a level of detail to the design is added, we can immediately verify if these changes or design details are still in compliance with the overall machine requirements.

5. SUMMARY AND CONCLUSIONS

This paper focused on the top left corner of the V process shown in Figure 1 and investigated how Model-Based Design can facilitate the exploration of system concepts for a wheel loader application. Instead of using prototype hardware for iterating through the system concepts, models of the wheel loader powertrain and implement hydraulic systems were built in an integrated environment to evaluate the interactions between these systems. This model then represented the specification of the loader concept which was executed through simulation. Example requirements for the wheel loader were derived based on a specific application, these requirements were then associated with the model, and test cases associated with the requirements were created to determine if the system concept meets requirements. This ability to evaluate multi-domain off-highway machine system concepts through models and link the models to the machine requirements along with the ability to instrument the models to check the results of the simulation to verify that the machine behavior meets requirements in a formal manner, allows the entire machine behavior and the interactions between the various systems to be well understood. Further, this ability allows the systems to be optimized to meet the machine level requirements in the absence of hardware to evaluate design options, resulting in a significant amount of time and cost savings.

Once the initial system concept study phase of the development process is over and the detailed design of the selected system concept proceeds further, the same model can be elaborated with the design details such as the control strategy, detailed component models, etc. The links to requirements and test cases can then be reused to ensure compliance with requirements for the detailed machine system concept. Once the control strategy is designed, software can be automatically generated from the same model and the software can be tested using the same requirements driven approach proposed in this paper. These aspects will be dealt with in a future paper.

Model-Based Design in an integrated modeling environment allows for the reuse of design information through various stages of the machine development process, thereby leading to time savings and reducing transcription errors. 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. K. Forsberg and H. Mooz, “The Relationship of Systems Engineering to the Project Life Cycle,” Proceedings of the Symposium of the NCOSE, NCOSE, 1991.
2. 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.
3. P. Grant, J. Freeman, R. Vail, and F. Huck, “Preparation of a Virtual Proving Ground for Construction Equipment Simulation”, Proceedings of DETC, 1998.
4. The MathWorks Inc., “Using MATLAB®,” Version 7.4, The MathWorks Inc., Natick, MA, March, 2007.
5. The MathWorks Inc., “Using Simulink®,” Version 6.6, The MathWorks Inc., Natick, MA, March, 2007.
6. The MathWorks Inc., “SimMechanics User’s Guide," Version 2.6, The MathWorks Inc., Natick, MA, March, 2007.

7. The MathWorks Inc., “SimHydraulics User’s Guide," Version 1.2, The MathWorks Inc., Natick, MA, March, 2007.
8. The MathWorks Inc., “SimDriveline User’s Guide," Version 1.3, The MathWorks Inc., Natick, MA, March, 2007.
9. The MathWorks Inc., “SimPowerSystems User’s Guide," Version 4.4, The MathWorks Inc., Natick, MA, March, 2007.
10. 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.
11. S. M. Prabhu, J. Wendlandt, J. Glass, T. Egel, Multi-Domain Modeling and Simulation of an Electro-Hydraulic Implement System, SAE Paper 06CV-145.
12. The MathWorks Inc., “Simulink® Verification & Validation User’s Guide," Version 2.1, The MathWorks Inc., Natick, MA, March, 2007.

CONTACT

SAMEER M. PRABHU

Sameer has over ten years of experience applying The MathWorks products in various application areas. As an Applications Engineering Manager 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