Maplesoft Engineering Solutions

Unlocking Commercial Knowledge: How System-Level Modeling forms the Basis for High-Performance, Low-Risk Designs

After long iterations and expensive prototypes, it seems you've got all the kinks out of your design. Weeks later, your system fails in the field for some unknown reason, despite your existing design tools showing no errors. A costly repair is in order, but the underlying problem hasn't been addressed. How do you track down exactly what went wrong? Recently, a leading industrial automation company found themselves in this exact situation, searching for a solution that would offer a cost-effective fix and prevent similar issues in the future.

The solution they found helped them see the interactions between every mechanism in their robot, and to accurately predict its behavior in the field. While their previous tools modeled the static loading of the robots, the complex motion of the robot arm was giving rise to brief moments of damaging torque for the motors – something their old tools couldn't model. What could have been a costly and daunting recall and repair became a software update to tweak the motion of the robot, eliminating the moments where the motors experienced the highest torque. This solution was made possible by modern design tools from Maplesoft and an innovative design approach called system-level modeling.

This is a true story of how system-level modeling can effectively address many of the problems faced by today's engineers. The approach uses software to model entire system interactions in one environment, long before expensive prototypes are built. With a good model in place, engineers can add mechanisms, test new scenarios, and get this done in a fraction of the time it might have taken before. Reducing downtime between iterations is crucial to keeping the design process energized and full of momentum, all while reducing the cost overruns that are primarily due to late-stage design changes.

What do we mean by system-level modeling? Generally, we are referring to a modeling process that uses physics-based components, connected together in a system spanning multiple physical domains. These models are a mix of discrete and continuous systems that aim to replicate the product's performance in real life – a virtual prototype upon which testing occurs. Like a physical prototype, a system-level model can be designed at different levels of fidelity, depending on the requirements that are being tested and how quickly the model needs to be constructed. For example, you might design a hydraulic lift so you can quickly answer some basic questions about the lift's movement and loading capacities. Later, you could add detailed controllers to the lift in order to fit the requirements of the project.

System-level modeling is concerned with the aggregate properties of the materials in a model. This is different

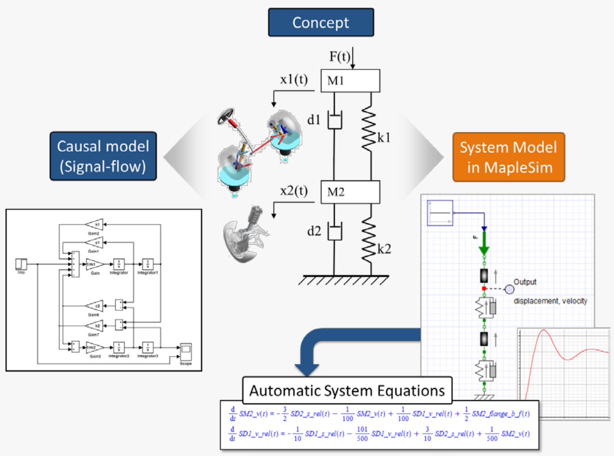


Figure 1: Earlier attempts at system-level modeling were done exclusively in a signal-flow environment. Models in MapleSim actually resemble the physical design, and still allow for signal flow blocks where necessary.

than other types of engineering models, such as finite element analysis (FEA) models, which dive into greater detail at the cost of much greater time and resource investment. Both kinds of models can play an important part in the design process, each having limitations and strengths. System-level modeling is useful to quickly test designs across an entire system, typically including domains that FEA isn't well suited for.

When a system-level model is ready for simulation, the underlying partial differential equations (PDEs) are automatically simplified to sets of differential algebraic equations (DAEs). These simplifications rest on various assumptions that can be made by both the software and the engineer. Simulating a model can be thought of as asking a particular question of the design, and assumptions determine the kinds of questions that a simulation could provide meaningful answers to. In many ways, this is similar to traditional design processes that use assumptions to build physical prototypes. In the world of system-level modeling, making (and tracking) the right assumptions are critical, as they will determine whether your model is giving you results that are relevant to your project. With a properly managed system model, the process of tracking underlying design choices is much more transparent than a typical design process, where an engineer's rationale might be hidden away in big spreadsheets, scratchpads, or even in their own head.

The move to system models is a significant step forward in taking disorganized design knowledge from across multiple domains to create a structured, transparent system of knowledge, unlocked for use across teams and projects to come.

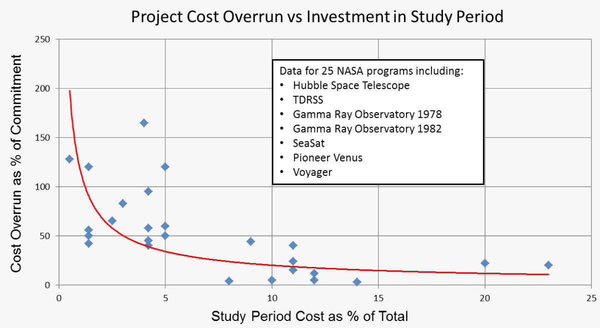
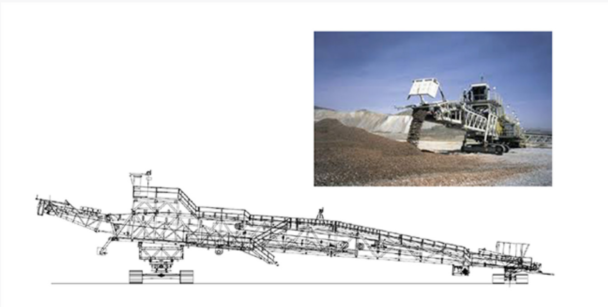


Figure 2: In a study of 25 NASA programs, it was shown that cost overruns were greatly minimized by spending even 5% more of a budget on the study period of a design process - a small price to pay for what can be many times that amount. Adapted from: Forsberg, Kevin, Hal Mooz, and Howard Cotterman, Visualizing Project Management. John Wiley & Sons. Hoboken, NJ. 2005.

Discovering Design Flaws Early

A leading manufacturer of industrial machinery asked the Maplesoft Engineering Solutions group to help test a concept design for a new radial stacker. With their current tools, they were looking at two weeks of design work with minimal flexibility to make large design changes. By using a system-level modeling tool like MapleSim, the radial stacker was designed in two days, demonstrating that the design was unstable due to sway and vibrations. The manufacturer was able to rule out this design early on, and quickly move to iterating on other designs without slowing down the design team's momentum.



Making the most of your Company Know-How

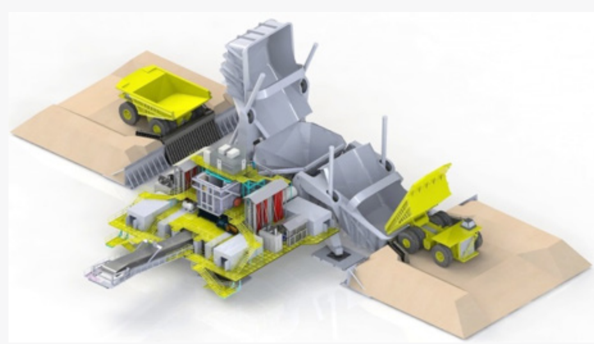
Adopting any new design technique within a company can be a significant challenge. The system-level modeling approach stands to offer a significant leap forward for the design process, but it must be implemented properly. Successful system-level modeling, like all design processes, is fundamentally reliant upon the skillful use of design **knowledge**. The proper handling of a team's knowledge and expertise will dictate iteration speed, design accuracy, and the success of a changing engineering team. MapleSim, advanced system-level modeling software from Maplesoft, was designed as a system-level modeling tool that aims to capture, preserve, and leverage company knowledge at every step of the modeling process.

Much of product design begins in the same place – an engineer's scratch pad. This is where ideas take shape as they move down the design chain into more structured approaches. This scratch pad is also where many engineers start to derive the mathematics in their concept designs, building in reasonable assumptions along the way. Regardless of the engineer's design toolchain, these assumptions and derivations are the foundation for understanding the future modeling process. Without the proper documentation, a retiring engineer stands to leave their replacement in a world of confusion, often leading new engineers to reinvent lost knowledge. This sort of lost knowledge is preventable with modern tools for derivation and assumption capturing. The tools provided with MapleSim offer fully interactive, live mathematical worksheets that allow for the freedom of a scratchpad in a documented environment. The next time an engineer wants to use another's design, they will have full access to that design's derivation in order to take what they need, and modify what they don't.

Now that an engineer has their derivations and assumptions in place, they're ready to create a system-level model with both standard components and their own custom components – all within MapleSim. Different than older, signal-flow modeling techniques, MapleSim offers an intuitive drag-and-drop interface, giving the engineer freedom to quickly create their system from a familiar set of industry-standard components that

Iterating Fast at the Concept Level

A mining company was trying to size a reservoir for a hydraulic lift system, and turned to system-level modeling to explore their design possibilities. Not only did they learn that their reservoir wasn't initially sized correctly, but they were able to run many iterations and change the details of several other parts of their model. The speed of studying test cases with a tool like MapleSim allowed them to ask many questions of their initial concept design in a short period of time. After this phase, they continued to develop their model into a high-fidelity, comprehensive system that could ask even deeper questions of their product.



actually resemble the physical system under design. Multiple domains can be easily connected together, and MapleSim monitors the system to ensure components are connected in physically meaningful ways, which is only one way in which system-level modeling helps automatically prevent design errors. When an engineer is satisfied with their initial model, it is ready to simulate and answer questions about the design's behavior.

The process of working with models has often been limited to a small group of engineers and experts within the company. MapleSim was designed to leverage the knowledge in system models by developing easy-to-use interfaces for usage across the company. By compiling all of the engineer's knowledge into intuitive and flexible deployment interfaces, managers and those without expertise are still able to run simulations and learn about the design performance. For example, a junior engineer could run various "what if" scenarios to optimize a design feature, freeing up resources for a lead engineer to continue with more important model design. The knowledge that stems from a system-level model in MapleSim can also be leveraged downstream, designed to seamlessly connect to other tools using FMI code, C-Code, and S-functions – all royalty free.

System level models can answer all kinds of questions during the design process. The real value in a system-level modeling tool like MapleSim is how well it can capture knowledge in a design process, and the way this knowledge is available across the organization and across the toolchain. This is a difficult process, and many system-modeling tools are limited in the utility they can have for those outside of a few select experts. With MapleSim, not only is the model creation process highly efficient and intuitive, but the basis of effective system-level modeling – design derivations and assumptions – are recorded and fully transparent through the models, avoiding any confusion as models are used and re-used in an organization. Further, MapleSim allows for easily created, web-deployed interfaces so that these highly valuable models can be explored and used across an organization.

System-level modeling is a powerful tool that seeks to help engineers answer questions about their design long before significant resource investment has been made. For some companies, this means performing hundreds of simulations to optimize the size of critical components, while for others, it means finding creative answers they didn't know existed for their problem. By evaluating a

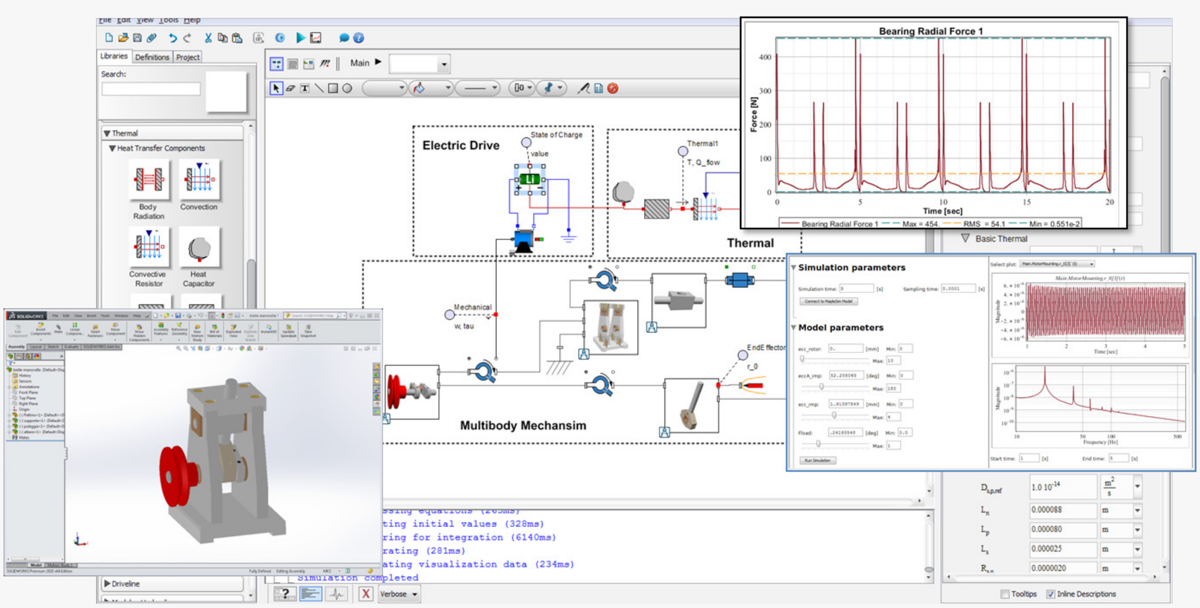


Figure 3: The MapleSim workspace offers an intuitive, drag-and-drop interface, while providing a virtually unlimited amount of customization for components and model analysis. This allows an engineer to start with a basic design concept, and continually add model fidelity along the way.

design in full view, and across all domains, the insight available to a company stands to be significant. Of course, success is highly dependent on fully utilizing all of the knowledge within the organization, and leveraging the knowledge generated with the modeling tool. By choosing the right system-level modeling simulation tool, the addition of this modern design process can be invaluable for not only the engineers, but for anyone who interacts with product design. By choosing MapleSim, powerful system-level models can be used with ease across an organization, providing insight that was previously locked away for only a select few to access. This insight can have significant impacts in energizing any design process, whether it's reinventing a design from the ground up, or fixing significant hardware failures with a simple software update.