Integrated Model-Based Machine Design with Motion

Robert Steele
Chief Technical Officer
Kollmorgen
805-696-1212
support@kollmorgen.com
Kollmorgen.com

Tony Lennon
Industry Manager
The MathWorks
508-647-7696
mathworks.com

Motion control applications are becoming increasingly challenging with higher performance, lower weight requirements, and increased functionality. Meeting these challenges often requires complex control algorithms, which call for more expertise and longer development time. In addition, the software is usually not tested until hardware prototypes are available which leads to many errors that require extensive re-coding. This design challenge is increased by the fact that manpower reductions are frequently forcing mechanical engineers to take over control system software development and testing functions that were previously handled by specialists.

As a result, a new approach has organically evolved that combines machine design with motion control under a model-based design umbrella to substantially reduce design and development time required for sophisticated control schemes. In the new approach, the mechanical engineer designs the mechanism in a computer aided design (CAD) system, brings the mechanism into the model-based design environment and develops complex algorithms without writing code by dragging and dropping function blocks from a toolkit. The engineer simulates the mechanism in the virtual environment. When the engineer is satisfied with the performance of the model, he or she downloads code to the motion controller and tests, measures and runs diagnostics in a real-time operating system (RTOS) environment for continual retesting and adjustments.

This approach enables substantial improvements in machine design in the virtual environment to optimize machine performance in motion. A small team, or even an individual, can use it to design complex motion control applications in a fraction of the time and cost required by the traditional build and test methods.

Control System Design Challenges

Higher performance and functionality requirements and cost constraints have substantially increased the challenges of designing a wide range of products, processes and systems. Hardware performance improvements have made it possible for mechanisms to operate at higher speeds, but system resonance and vibration control have become a much greater concern. These changes have dramatically increased the challenges and costs of designing control systems.

Managing this complexity can prove challenging to the hardware and software engineering teams responsible for designing control loop strategies to address electro-mechanical interactions and external disturbances, providing the required level of performance, ensuring regulatory compliance, and so on. These requirements are often difficult or impossible to meet with standard algorithms that use simple mathematical computations and are generally limited to one or possibly two inputs. Machine builders are increasingly turning to complex algorithms that account for multiple inputs and incorporate complex mathematical computations and models. Yet, more and more, the control algorithm, coding and testing specialists on these teams are going away with mechanical engineers expected to pick up the slack.

The traditional approach to developing complex control systems involves first writing text-based specifications that define the requirements for the control systems in as much detail as possible. These specifications are passed to control engineers who design a control strategy that is then implemented in C/C++ code. The control systems development process may be hindered by the difficulty of finding people who are experts in control systems algorithms and also proficient programmers. It's also very easy to make, and very difficult to detect, mistakes in hand-written code, especially during the early phases of the development effort. Chances are that the handwritten code may only make sense to the person who wrote it, which may create communications difficulties in larger projects.

Simulating the operation of the complete system at this stage by writing the equations of motion to model the interaction of subsystems can be prohibitively complex. Software validation normally cannot be addressed until late in the development cycle so errors are usually not detected until the code is run on the actual hardware. After each iteration, the design must be re-coded and the source code re-compiled, re-run, and debugged, which greatly increases development time and expense. And in addition, when the code is finally ready for production, it must normally be re-coded to run on the target processor.

<u>Advantages of Model-based Design</u>

Model-based design offers a new approach to the design of mechatronics control systems by providing an environment that makes it possible to develop an executable specification that incorporates the key elements of the control logic and mechanical and electrical design. A system level model is defined using blocks that represent mathematical operations between input and output signals. For example, the model may contain a block representing a motor. Initially, the model may simply take a voltage input and convert it to an output torque. As the design process continues, additional detail can be added to increase the fidelity of the model, such as by adding noise in the voltage, temperature and magnetic saturation effects, and so on. Engineers can further increase model fidelity by replacing approximate mathematical representations of the mechanical system with blocks that represent mechanical bodies and linkages translated automatically from the computer aided design (CAD) file.

Engineers can use the model to simulate the operation of the control system in software. This approach makes it possible to investigate a wide range of alternative design concepts without investing time and money in writing code or building hardware. Until recently, a limitation of model-based design has been that the pre-configured blocks used to construct the model were available only for standard control algorithms such as PIV and PID. This meant the technology could only be used on less sophisticated control systems. In addition, code could not be automatically generated for leading-edge hardware.

<u>Extending Model-based Design into Mechatronics</u>

More recently, graphical building blocks have been introduced that make it possible to model far more sophisticated control systems and download them to the RTOS environments used on the most advanced motion control systems. These standard function blocks reduce the need for specialized knowledge of software programming expertise, enabling mechanical engineers who understand the underlying application to take control of the design process. The function blocks have been optimized to provide efficient and predictable motion control performance on the latest generation of motion controllers.

Blocks are now available that support complex gearing and following methods; gain switching; vibration control; notch, resonator, low-pass and bi-quad filters; sophisticated multiple input multiple output (MIMO) plant models; velocity, acceleration and friction feed forward; and observers as well as the ability to monitor response at any point in the control loop. The control block set also allows multi-axis line-shafting, following and gearing to be performed quickly with the control block set. Equipped with comprehensive data capture, logging and visualization tools, the function blocks allow mechanical, I/O and software data integration into one measurement environment. This approach enables machine designers to take advantage of emerging accelerometer feedback devices. This in turn enables mechanisms to be designed with reduced weight and stiffness, resulting in lighter, faster, less expensive, and more precise machines.

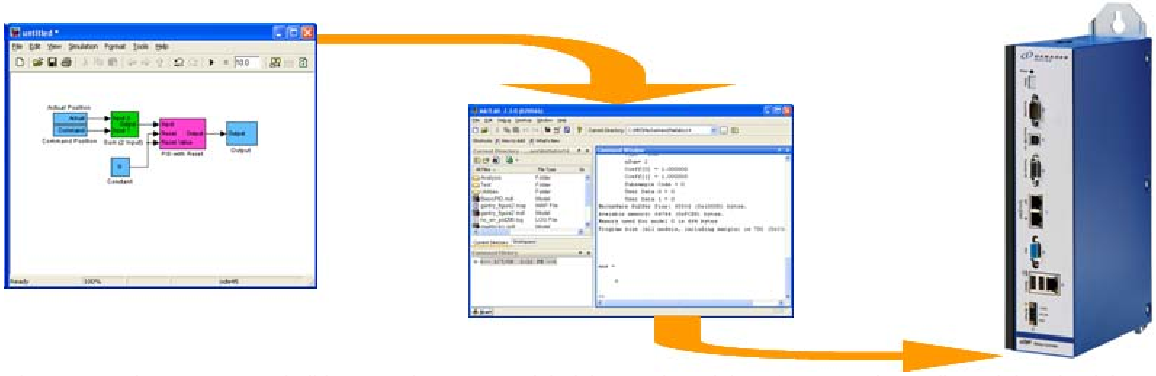

Figure 1: Engineers can now build a control system model, debug and test it in software, and compile and download the model to the motion controller.

Plant model output parameters, such as motor drive and positional feedback, can be used to evaluate control performance. Physical parameters such as mass, length and capacitance that cause instability can be easily identified and adjusted. Engineers can predict the move performance, ringing and settling time, motor and drive sizing of alternative control systems concepts prior to the prototyping phase. Test cases can be embedded in the model to check each design iteration against requirements and identify mistakes that can be easily and inexpensively corrected.

Once the control strategy has been developed and tested in simulation, engineers convert the code algorithms from continuous (analog) to discrete (digital). Code is automatically generated for the motion controller. This eliminates the need for systems engineers to be code-writing experts, prevents the introduction of coding errors and saves time. The result is that sophisticated custom motion algorithms can be implemented in a fraction of the time and cost of conventional methods.

The approach protects intellectual property by eliminating the need to disclose control strategies to vendors, while making it nearly impossible to reverse-engineer the control strategy. The control system can be tested using hardware in the loop testing with control algorithms expressed in code on a target microprocessor, and the plant model expressed in code on a real-time system. The next step is usually prototyping with the control algorithms expressed in code on a real-time system, and the plant model consisting of real hardware.

Solving a Challenging Control System Design Problem

Let's look at how model-based design with a mechatronics toolkit was used to solve a challenging control system design problem. A company builds machines used to assemble printed circuit boards. The PCBs being assembled are clamped to the machine base prior to placement. The placement head rides on along the beams of an overhead gantry that is also mounted to the base to pick up components from the feeder, move to the appropriate position on the board and place the components.

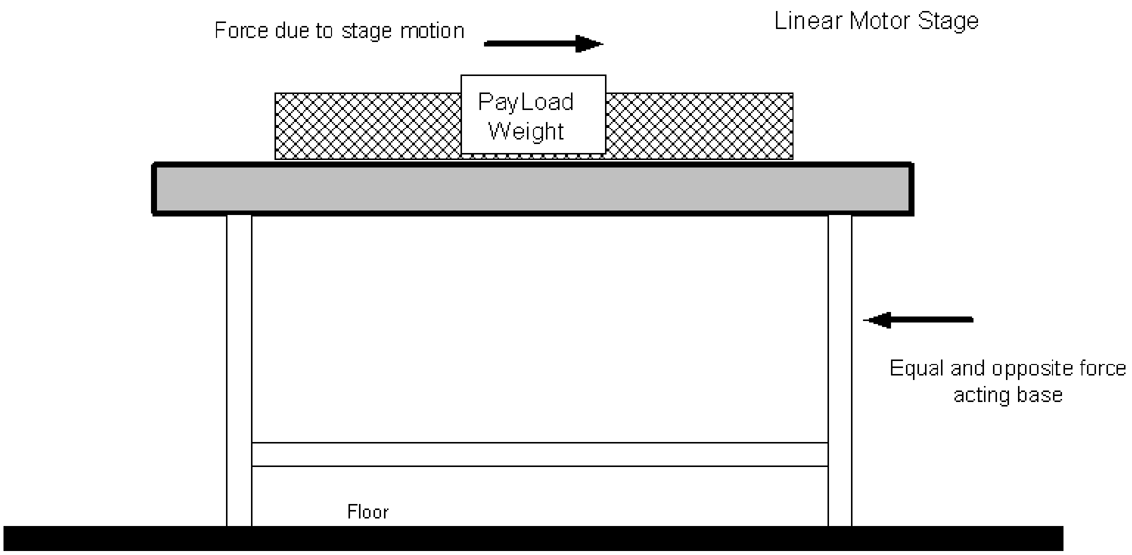

Figure 2: Reaction forces cause vibration.

In early versions of the product, a relatively simple PIV control algorithm was used. But as the performance of the gantry system was increased to improve the productivity of the machine, the reaction forces generated by the gantry as moved over the board increased. These reaction forces generated vibrations strong enough that the components could no longer be placed within the specified tolerance limits. Initially, the machine builder addressed this problem by increasing the size of the base, reducing the deflection generated by the reaction forces. However, this approach increased the size and cost of the machine and limited future performance improvements.

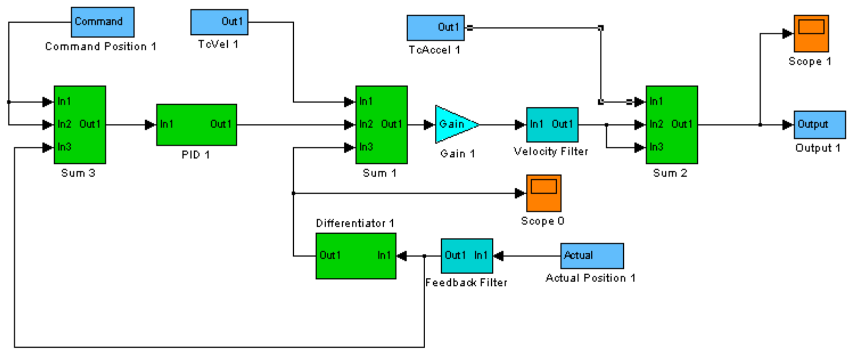

Figure 3: Original PIV-based control system

Recently, the machine builder validated model-based design with a mechatronics toolkit as the solution to this problem. The first step was to model the existing PIV controller using functional blocks as shown in Figure 3. The command position input comes from the program that indicates the position of the components on the PCB while the actual position comes from the sensors on the gantry axes. Control engineers simulated the performance of the original control system and predicted levels of vibration that matched what was seen in the real world.

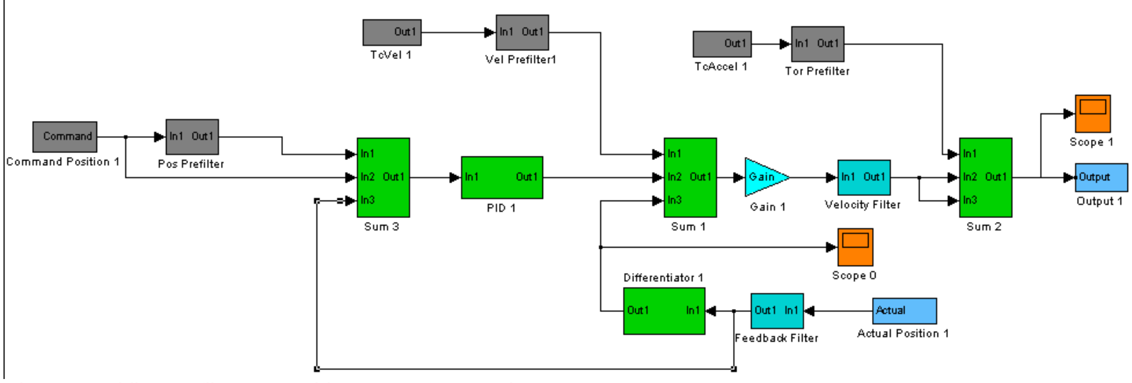

Figure 4: Adding prefilters to avoid resonant frequencies.

The next step was to upgrade the control system by adding position, velocity and torque prefilters as shown in Figure 4. These prefilters prevent the control system from generating a motion profile that would cause the structure to resonate. Control engineers simulated the performance of this control system and determined that it provided a significant but not sufficient improvement in performance.

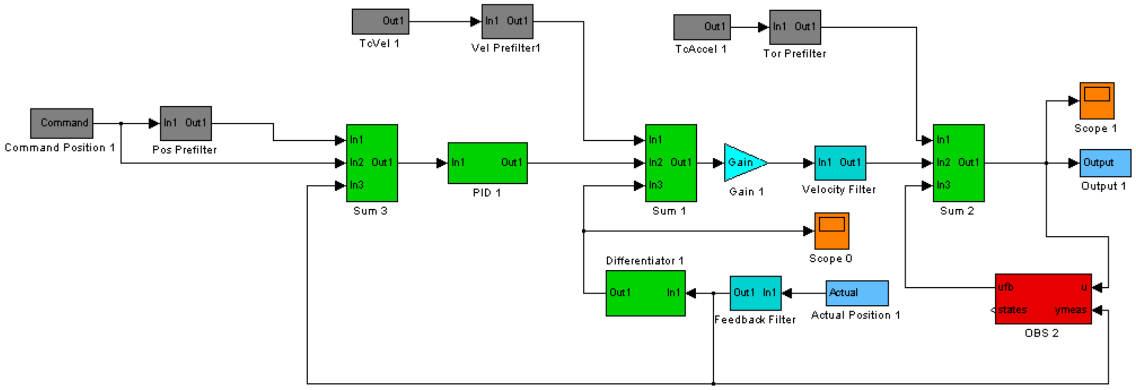

Figure 5: Adding a sensor to counteract vibration.

Next, an observer block (OBS 2) was connected to an accelerometer mounted on the base as shown in Figure 5. The accelerometer measures the actual vibration experienced by the base and provides an additional input to the control system so that the control system can generate a motion profile to counteract the vibration. The simulation showed that this approach reduced vibration to levels that made it possible to substantially reduce the size of the base. Engineers substituted different base masses into the plant model in order to determine the minimum base mass that would make it possible to meet component placement specifications.

It would have been very expensive and taken months to code each of these possible solutions, implement them in hardware, and run experiments to determine their performance. Instead, the machine builder control engineers were able to evaluate these alternate approaches in software in weeks. After deciding to use the approach shown in Figure 5, code was automatically generated for the machine controller. A prototype was built to test the new approach and it worked exactly as predicted by the model. By greatly reducing vibration, the new approach made it possible to increase production rates from 50,000 parts per hour to 100,000 parts per hour while greatly reducing the weight of the machine.

New Ways to Work

As the above example illustrates, model-based design, combined with integrated, advanced mechatronics on a RTOS platform, enables substantial improvements in control system performance while reducing development time. This translates directly to the bottom line through reduced head-count and the ability of a lean team to bring machines to market faster.

As original equipment manufacturers (OEMs) continue to reduce resources and increase demands for higher performance, new approaches to efficient, harmonious and economical mechanical and software design development becomes necessary. By using industry-leading solutions available on the market today in this creative manner, OEMs will find their mechanical engineers quickly transforming into Renaissance men capable of bringing high performing machines to market with much less support from algorithm development and testing specialists.

ABOUT KOLLMORGEN

Kollmorgen is a leading provider of motion systems and components for machine builders around the globe, with over 70 years of motion control design and application expertise.

Through world-class knowledge in motion, industry-leading quality and deep expertise in linking and integrating standard and custom products, Kollmorgen delivers breakthrough solutions unmatched in performance, reliability and ease-of-use, giving machine builders an irrefutable marketplace advantage.

For more information visit www.kollmorgen.com, email support@kollmorgen.com or call 1-540-633-3545.

.