Design World

November 2009

www.designworldonline.com

ALSO INSIDE:

SENSOR NETWORKS:
Are wireless sensor networks really ready for industry?
Page 54

FLUID POWER:
Diaphragm pumps by design
Page 62

SENSORS:
Acoustic resonance: the heartbeat of touch sensors
Page 70

Motion Control:

Integrate model-based machine design with motion control

Page 50

Integrate model-based machine design with motion control

Model-based design methods cut the time and cost of building and testing motion control systems.

Robert Steele
Chief Technical Officer
Kollmorgen
Radford, Va.

Tony Lennon
Industry Manager
Industry Marketing,
Industrial Automation
and Machinery
The MathWorks, Inc.
Natick, Mass.

Two factors have made the design of motion control systems in machines more challenging. One is that mechanical improvements have made it possible for machines to operate at much higher speeds than ever before. But the resulting mechanical problems such as resonance and vibration control have become a much greater concern. Solving these complex problems often requires in-depth knowledge of control systems as well as mechanical and electrical systems. The other factor is the loss of engineers with extensive control system design expertise due either to layoffs or company restructuring. The problem is then how to solve these increasingly complex problems with too few people and limited time and funds.

A promising new technique called model-based design integrates the machine design function with the motion control system. Model-based design makes it easier for engineers with limited expertise in control systems to design machines and solve the problems that arise. It generates executable code based on the control logic and the mechanical and electrical specifications of the system in a model-based environment, overcoming the limitations of more traditional approaches to control system design.

Traditional approach

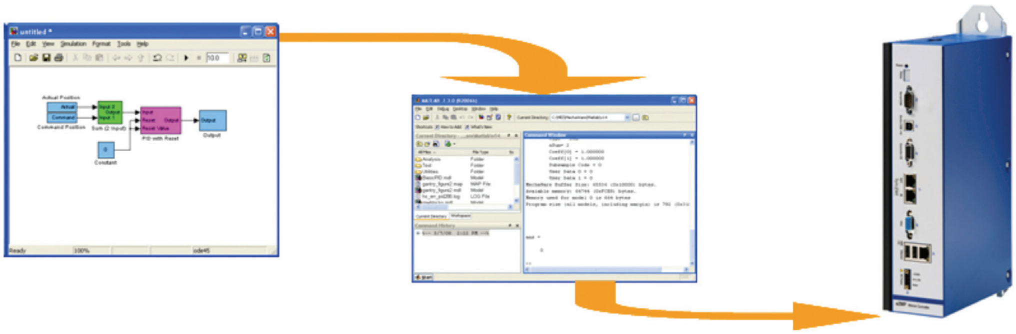
The traditional approach to designing complex control systems involves first writing text-based specifications that define the requirements in as much detail as possible. These specifications are then used to design a control strategy implemented in computer code. This process can be hindered by the difficulty of finding experts in control system algorithms. It's also very easy to make, and very difficult to detect, mistakes in hand-written code, especially during the early phases of development effort. Chances are the handwritten code may only make sense to the person who wrote it, which may create communication difficulties in larger projects.

The next step is to simulate the operation of the system. However, writing the equations of motion to model all of the interactions of subsystems can be complex. Furthermore, software validation normally can't 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 recoded and the source code recompiled, rerun, and debugged, which increases development time and expense. In addition, when the code is finally ready for production, it must normally be recoded to run on the target processor.

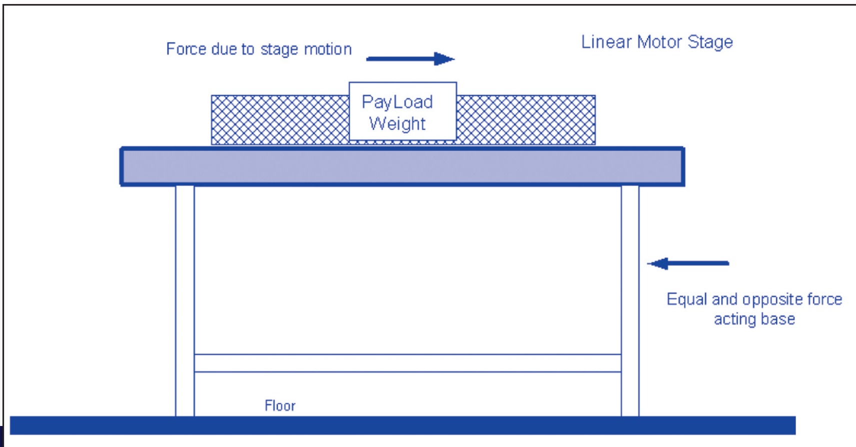
Advantages of model-based design

Model-based design offers an alternative to the traditional approach to control system design. It provides an environment to develop an executable specification in the form of software code that incorporates all of the key elements of the control logic and mechanical and electrical design.

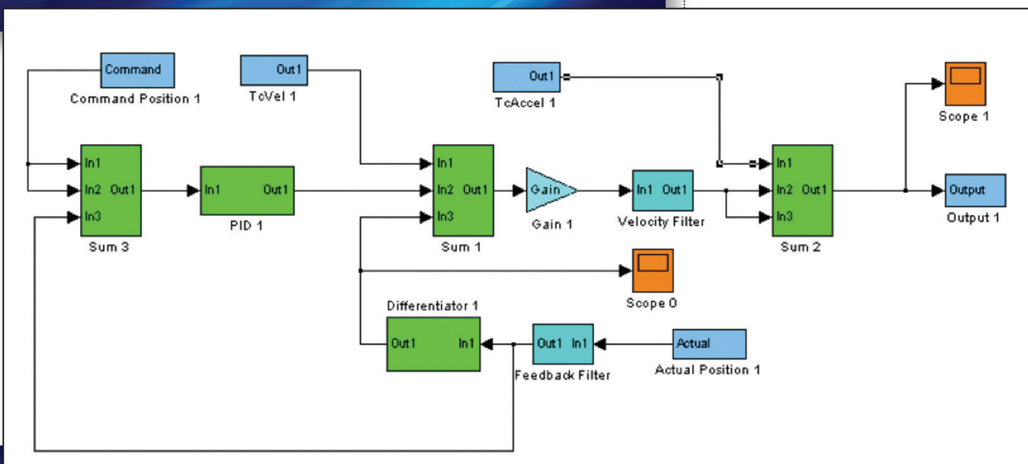
A typical system-level model uses blocks that represent mathematical operations between input and output



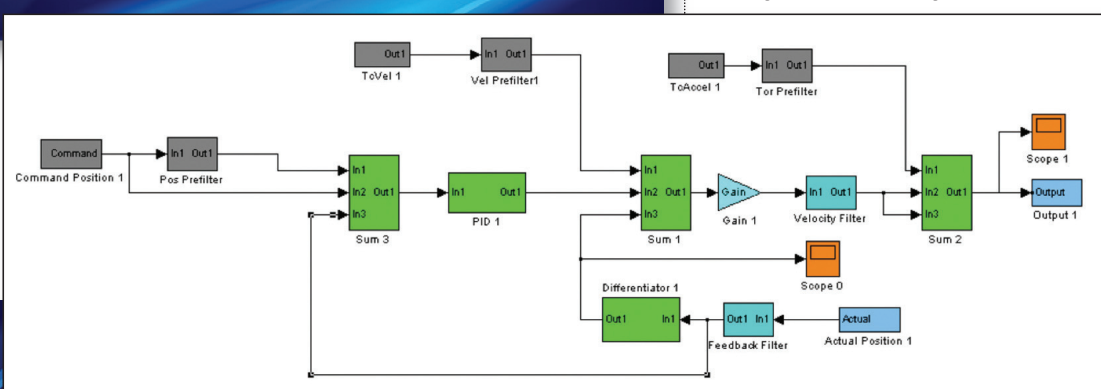
Engineers can now build a control system model, debug and test it in software, and compile and download the model to a motion controller, saving valuable time and cost.



In this pick-and-place machine, increased gantry speeds led to an increase in reaction forces, which caused excessive vibration.



The original control system for the pick-and-place machine was a simple PIV controller.



The updated controller uses position, velocity, and torque pre-filters to prevent resonance.

signals. For example, a model can 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 adding noise in the voltage or temperature and magnetic saturation effects. You 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 a CAD file. You can also use the model to simulate operation of the control system in software. This approach makes it possible to test 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 models were available only for standard control algorithms such as PIV (Proportional-Integral-Velocity) and PID (Proportional-Integral-Derivative). 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. Newer graphical building blocks make it possible to model more complex control systems and download them to RTOS (Real-Time Operating System) environments used on the most advanced motion control systems. These standard function blocks reduce the need for specialized software programming expertise, letting mechanical engineers take control of the design process.

Newer function blocks can also support complex gearing and following methods such as gain switching, vibration control, and multiple input multiple output (MIMO) plant models. Combined with data capture, logging and visualization tools, these function blocks integrate mechanical, I/O and software data into one measurement environment.

Plant model output parameters, such as motor drive and positional feedback, can be used to evaluate the controller, making it simple to identify and adjust physical parameters such as mass, length, and capacitance which can cause instability. You can predict the move performance, ringing and settling time, and motor and drive sizing of alternative control system

A real-world example

Here's how a model-based design approach with a mechatronics toolkit solved one machine builder's control system problem. The company builds machines used to assemble printed circuit boards (PCBs). The PCBs are clamped to the machine base prior to placement. The placement head rides along the beams of an overhead gantry mounted to the base to pick up components from the feeder, move to the appropriate position on the board, and place the components.

Early versions of the machine used a relatively simple PIV control algorithm. However, as the speed of the gantry system was increased, the reaction forces also increased. These forces generated vibrations strong enough that the components could no longer be placed within the specified tolerance limits. Initially, the machine builder addressed the 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 modifications.

Recently, the machine builder switched to a model-based design approach to solve the problem. The first step was to model the existing PIV controller using functional blocks. 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. The original control system model was then simulated with predicted levels of vibration matching what was seen in the real world.

The control system was then upgraded by adding position, velocity and torque prefilters. These prefilters prevent the control system from generating a motion profile that would cause the structure to resonate. The simulation found that it provided a significant but not sufficient improvement in machine performance.

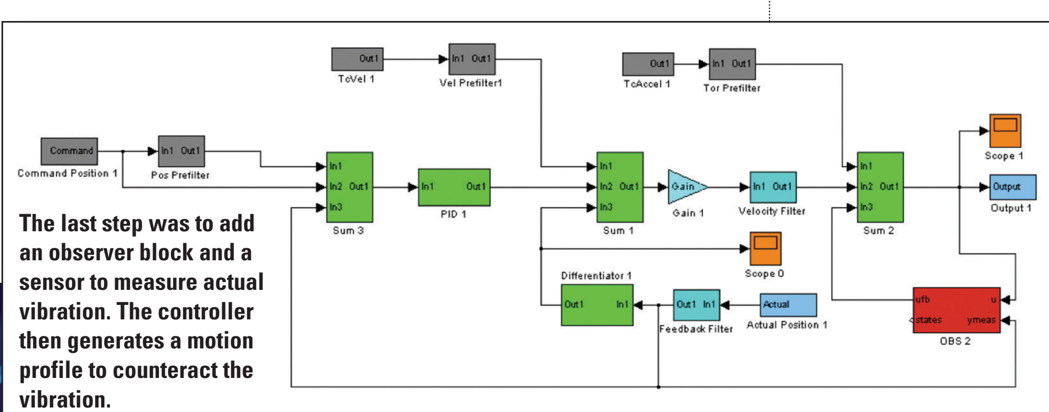
Next, an observer block was connected to an accelerometer mounted on the base. The accelerometer measures the actual vibration of the base and provides an additional input to the control system so that it can generate a motion profile to counteract the vibration. The simulation showed that this approach reduced vibration to levels that made it possible to substantially reduce the size of the base.Different base masses were substituted 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 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 on the final approach, the 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 reducing the weight of the machine.

concepts before prototyping the design. Test cases can be embedded in the model to check each design iteration against requirements and find and correct any mistakes.

Once the control strategy is developed and tested in simulation, the code algorithms are converted from continuous (analog) to discrete (digital) and code is automatically generated for the motion controller. This eliminates the need for code-writing experts, prevents the introduction of coding errors, and saves time.

The model-based design approach also protects intellectual property by eliminating the need to disclose control strategies to vendors, while making them nearly impossible to reverse-engineer. 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 on a real-time system using real hardware. **DW**



The last step was to add an observer block and a sensor to measure actual vibration. The controller then generates a motion profile to counteract the vibration.

Kollmorgen
1.540.633.3545
contactus@kollmorgen.com
www.kollmorgen.com

MathWorks
www.mathworks.com