Technolution

/technology

High-speed control

>>electronics with software properties

Control engineers like to use software to make their controllers. Such controllers are flexible and often cheap, but not always fast enough. Electronics are many times faster and more accurate but inflexible. Technolution has designed a high-speed electronic controller with the flexibility of software.

When we talk about speed in *Objective*, we are usually talking about electronics. But mechanical systems are becoming ever faster too. For example, pick & place machines that can place components on a PCB at lightning speed, or wafer-steppers that can move the wafer into the right position under the lens in a fraction of a second. In an electron microscope, speed is less important, but the movements need to be made with extreme precision and as little vibration as possible. All of these are mechanical systems that demand fast and accurate control. A software controller cannot achieve the required speeds: the processing power required would result in a very expensive controller. An electronic controller with programmable logic is fast enough and cost-effective but not flexible. Once the design is fixed in the FPGA, there is nothing left to adjust. Or is there? Technolution has devised an electronics-based method with the flexibility of software. All the settings you want to be able to change in software are now also available in electronics. Moreover, the control frequencies are much higher than in software controllers and they are 100% predictable.

From transfer to controller

Control technology is the field of the mechatronic engineer. The first stage in his work is to describe the process that needs to be controlled. For example, a living room with a heater and a thermostat. If the thermostat goes up by a degree, how long does it take before the whole room is a degree warmer? It takes a while. Everything in the room has to heat up, and there is leakage from poor insulation, windows and doors. All of which results in the transfer: intervention versus result. For a simple process, this can be a mathematical formula. However, practical situations tend to be more complex and difficult to describe in a single formula. So control technology offers general formulas (PID) to describe the process. You test things out by introducing a reference signal at the input and then measuring how the system or process behaves at as many points as possible. For example, using a hammer to strike the axle of an engine and a sensor to pick up the resulting vibrations. This is how you determine the transfer of the system experimentally, which can result in a formula with a number of parameters. Using the experimental data, the mechatronic engineer can design a controller to govern the system. In his design, he wants to be able to measure what is happening and adjust it at as many points as possible, because in complex systems and machines, the controller settings will never be exactly right straight away. So you want to be able to adjust those settings.

Fast and predictable

A software package like Simulink offers that flexibility. The user can join all kinds of functions together as blocks. As long as the controller remains on the software side, there is enormous flexibility to measure and adjust all kinds of things. However, the great limitation of software is its speed and predictability. Software often suffers from jitter. The timing of software is not accurate enough for fast control processes. At low frequencies, that is not really a problem, but at around 10 kHz, the jitter in software is a serious obstacle. Electronics, by contrast, are predictable and accurate to the nearest clock pulse. In theory, electronics are also affected by jitter, but it is many orders lower than in software and therefore negligible. The clock frequency can easily be a factor of a thousand higher than the control frequency: control at 300 kHz, clock at 200 MHz. The accuracy of the overall control system is more likely to be limited by the sensors.

So if you want to design a fast controller, you will need to switch over to electronics (programmable logic in field-programmable gate array FPGAs). Simulink offers the option of downloading the software controller to an FPGA. The programme then converts the software into one large chunk of Visual Hardware Descriptive Language (VHDL) code which is synthesised and downloaded to the FPGA. Generating a new controller for the FPGA can take an hour. And once on the chip, the result is fixed: an electronics controller which cannot be altered and in which there is nothing left to measure between the separate controller blocks. A controller without adjustments and measurement points, in other words. Changes are only possible in Simulink, after which new

VHDL code needs to be generated to overwrite the FPGA. This results in time-consuming repetition.

Modified library

In order to retain the flexibility of a software controller, Technolution has developed its own library of building blocks for Simulink. These blocks are already present, part-configured, on the FPGA but have been modified in such a way that they can be used in Simulink. Each has specific functionality and can be joined together on the FPGA in the appropriate way. Now the controller can also be transferred to an FPGA using Simulink. Not as one big chunk of VHDL code but simply with some configuration data. These configuration data are determined using a tool developed by Technolution. The tool reads and interprets the Simulink files and generates the configuration data for the controller. You could compare it to a navigation system: instead of downloading a complete high resolution map to the FPGA, you merely download a route onto a map that is already there.

FPGA controller as prototype and end product

The controller thus created in the FPGA has the same measurement and control points as in Simulink. It is an electronic controller which the user can use for real-time measurement and control at all the same points as in the simulation – which is very useful for the mechatronic engineer. This system offers him a handy additional tool to define his processes with. He devises a controller and tests theoretically using Matlab and Simulink. When he is satisfied with the simulation, he transfers it to electronics in a matter of seconds. He can then link these electronics to the real process in order to see if they really do offer the control he is looking for. He can now make any adjustments to fine-tune the controller ‘on-the-fly’ and sees the results immediately. So he does not have to go back to the computer to change the parameters in Simulink and wait another hour for a new configuration of the FPGA to be generated.

This gives the user a handy tool which can achieve operating frequencies that are not possible in software, both for R&D and production. Even when the controller is mass produced, it retains its flexibility, and at a much lower price than comparable software controllers.