Back to Square One

It's hard to imagine what it would be like to do control research or control applications without a computer. When I read early papers in *IEEE Transactions on Automatic Control* on "modern" control methods, I find myself amazed at how much was done with almost no computational resources. I imagine graduate students in the 1960s painstakingly assembling decks of punched cards and then waiting for the results of crude FORTRAN programs to elicit sparse but valuable numerical examples.

As a student in the 1970s, I was given a few "dollars" in a computing account to complete homework assignments. To save money, debugging was an arduous pencil-and-paper process. Infrequent computer runs were separated by long periods of mental testing. A lot of thinking was needed.

Now that computation is cheap and fast, we can use it in ways that were all but impossible 35 years ago. One use is sensitivity studies. You can learn a lot from a few good simulations, but you can learn a lot more from a few thousand. By varying plant properties, control laws, commands, disturbances, time delays, and anything else you can think of, it's easy to determine how a control system will perform under a range of real-world conditions—well beyond the predictions of analytical control methods. These studies may also provide new insights and valuable discoveries.

Another advantage of fast and cheap computing is debugging. Rather than

Dennis Bernstein in Ludington, Michigan.

spending hours laboriously looking for errors, a program can be run numerous times to isolate errors. Although this approach can save countless hours of tedious line-by-line checking, it can also promote a certain amount of laziness and sloppiness—the easier it is to debug the code, I find myself investing less time and effort designing and writing it in the first place.

The problem with this scenario—fast and cheap computing that facilitates thousands of runs and easy debugging—is that it's becoming increasingly untrue. I don't mean to suggest that computers are becoming slower or more expensive. Rather, the problem is that, as experts in systems and control, the applications we're facing are becoming increasingly demanding. I'm referring to very large-scale problems, such as data assimilation for weather forecasting, which involves thousands or millions of states. Running these models on a PC or even a MAC with 8 GB of RAM is out of the question.

IEEE Control Systems Magazine (*CSM*) editorial board members attending the *CSM* board meeting at the American Control Conference in St. Louis. From left: Zongli Lin, Luigi Del Re, Mike Polis, Alex Leonessa, Jan Swevers, Panagiotis Tsiotras, Maryam Khanbaghi, Dennis Bernstein, Katie Johnson, and Jonathan How.

Digital Object Identifier 10.1109/MCS.2009.934092

Harish Palanthandalam-Madapusi, M. Vidyasagar, and Dennis Bernstein at the American Control Conference in St. Louis.

Very large-scale applications can only be run on clusters of computing nodes, using parallelized codes. As in the early days of computing, jobs are submitted to a queue, and the results are obtained based on priority and run times. Modest runs can take anywhere between a day and a week or even longer to complete, depending on both the code and the number of users sharing the facility. The number of runs that can be made in a week, month, or year is extremely limited, and the computer can no longer be used to test programs. In other words, a lot of thinking is needed.

Dennis S. Bernstein

Skynet Redux

Even while he held out a hopeful vision of a future in which intelligent machines could resolve problems of great human concern and consequence, he was not without his fears of what the actual results might be (Ashby 1948). An intelligent machine by his definition was, after all, a machine that succeeded in achieving its *own purposes*, regardless of the resistance it encountered:

But perhaps the most serious danger in such a machine will be its selfishness. Whatever the problem, it will judge the appropriateness of an action by how the feedback affects itself: not by the way the action benefits us. It is easy to deal with this when the machine's behavior is simple enough for us to be able to understand it. The slave-brain will give no trouble. But what of the homeostat-type, which is to develop beyond us? In the early stages of its training we shall doubtless condition it heavily to act so as to benefit ourselves as much as possible. But if the machine really develops its own powers, it is bound sooner or later to recover from this. If now such a machine is used for large-scale social planning and coordination, we must not be surprised if we find after a time that the streams of orders, plans and directives issuing from it begin to pay increased attention to securing its own welfare. Matters like the supplies of power and the prices of valves affect it directly and it cannot, if it is a sensible machine, ignore them. Later, when our world-community is entirely dependent on the machine for advanced social and economic planning, we would accept only as reasonable its suggestion that it should be buried deeply for safety. We would be persuaded of the desirability of locking the switches for its power supplies permanently in the "on" position. We could hardly object if we find that more and more of the national budget (planned by the machine) is being devoted to ever-increasing developments of the planning-machine. In the spate of plans and directives issuing from it we might hardly notice that the automatic valve-making factories are to be moved so as to deliver directly into its own automatic valve-replacing gear; we might hardly notice that its new power supplies are to come directly from its own automatic atomic piles; we might not realize that it had already decided that its human attendants were no longer necessary. How will it end? I suggest that the simplest way to find out is to make the thing and see.

—Peter M. Asaro, *Mechanisms of Adaptation to Intelligence Amplifiers: The Philosophy of W. Ross Ashby*, excerpted from *The Mechanical Mind in History*, edited by Philip Husbands, Owen Holland, and Michael Wheeler, MIT Press, Cambridge, MA, 2008, pp. 179–180.