1

Introduction

1.1 A PERSONAL VIEW

Although many writers are happy to put a date on the day a Japanese (or was it a Finn?) coined this rather ungainly word, *mechatronics* has been around in spirit for many decades.

My first brush with industry involved designing autopilots. The computers on which they were based used analog magnetic amplifiers—and later transistors—rather than the digital microcomputer we would expect today. Nevertheless, how can we describe as anything but a robot a machine that trundles through the sky, obeying commands computed from a multitude of sensor signals that enable it to make a perfect automatic landing?

By the mid-1960s, some computers had started to shrink. While the Atlas was fed a succession of jobs by an army of operators, an IBM1130, built into a desklike console, allowed real time interaction by the user. Soon we were able to buy "budget" single-board computers for a thousand British pounds. Although these had a mere 16 kilobytes (kbytes) of memory, their potential for mechatronics was immense.

One of my Cambridge researchers took on the task of revolutionizing the phototypesetter. The current state of the art was to spin a disk of letter images, triggering a flash to expose each letter onto photographic film. This was certainly "mechatronic" to an extent, requiring the precision positioning and timing under electronic control, but the new approach distilled the essence of mechatronics.

Essentials of Mechatronics, by John Billingsley

The method is now commonly found in the laser printer. A spinning mirror scans a laser beam across the photosensitive film, building up the image by rapid switching of the beam. Letter shapes are held in computer memory, and the entire mechanical design is simplified.

I consider this tradeoff between mechanics, electronics, and computing power to be the guiding principle of mechatronics.

The research team were soon knitting similar computers into a variety of real-time applications, including an "acoustic telescope" to build the signals from 14 microphones into an image of the source. Hydrofoils were simulated, violins were analyzed for their "Stradivarius-like qualities," and music was synthesized. A display for a color television, novel in those days, depended on a minimum of electronics and a wealth of software.

But computing power soon came in increasingly small packages. Texas Instruments had produced a single chip that could function as a pocket calculator. By the time I had moved from Cambridge to Portsmouth, Intel and Motorola were head-to-head with competing microprocessors.

In Britain, the Microprocessor Awareness Project (MAP) triggered a deluge of applications—but only a small proportion of them deserve truly to be considered as mechatronics.

Industrial firms were offered 2000 pounds'-worth of consultancy to consider how microprocessors could be added to their products. Some sharp operators made a killing, providing virtually identical reports to a diversity of clients. Others "brokered" projects to earnest academics. Printing machines sprouted boxes with twinkling LEDs (light-emitting diodes), wiring and relays patched on top of the "standard model." In many cases it made the machines virtually unusable and impossible to maintain.

Gradually, however, the concept percolated through that the computing aspect could be made fundamental to the operation of a machine. The mechanical precision and complexity could be traded off against electronics and computing power, just as in the case of the typesetter.

One MAP project concerned the design of a clock for a domestic cooker. Not very romantic, perhaps, but the client's choice of the primordial chip as used in the earliest pocket calculators made it a conundrum with attitude. It took several years and many generations of the product to persuade the company to adopt something simpler to program. The manufacturers of the original chip kept halving their price.

The chips were supplied, mask-programmed, in batches of 10,000. That concentrated the mind wonderfully on making sure that the code was correct. But once we had weaned the company off the TMS1000, there was room in the chip's memory not only for the job at hand but also for the next version we had in mind.

One focus of our research was the *Craftsman Robot*. An *energy regulator* is the switching element behind the knob that allows the power of a cooking ring to be varied. During its manufacture, several adjustments have to be made. We used a Unimation Puma 560 robot to pick each regulator from a tray and offer it to a test rig. Instead of acting as a simple "mover," however, the Puma was equipped with a screwdriver to adjust the regulator when it was still held in its gripper. Of course, we could not resist taking the robot apart and analyzing its software and drive circuitry.

Other industrial projects included marine autopilots and a flux-gate compass. But another interest would soon seize my attention.

In 1979, planning started for holding the Euromicro conference in London. Lionel Thompson, the chairman, wanted an added showpiece, and his mind was on "The Amazing Micromouse Maze Contest" that had just been announced by *IEEE Spectrum.* I put my hand up to organize the contest.

I then started to follow the news from the United States. Blows were nearly exchanged when the "dumb wall followers" sprinted through the maze from the entrance at one corner to the exit at the other, much faster than their brainier rivals. How could the rules be massaged to give brains the edge?

Donald Michie, a guru of technical conundrums, was all for making the objectives more abstract, perhaps adding a cat to the fray. The solution lay in the opposite direction, to give the mouse builders more specific information that could be designed into the logic of their machines. Our maze was specified as 16 × 16 squares, with the target at the center, not on the edge. In that way, paths could circle the center to form "moats" that no mere wall-follower could cross.

A preliminary run was held in Portsmouth in July, with results that literally gave me nightmares. Of the six mice that competed, only one could make any attempt to follow a passageway, let alone find the center. Japanese observers were there in force, cameras snapping away, and I was amazed that everyone seemed to enjoy the show.

At the conference in September, 15 mice competed. A sleek machine from Lausanne should perhaps have won—but it expected more precision of the maze than the carpenters had provided and became lodged on a join in the boards of the base.

The winner was a clanking contraption, cobbled together around a brilliant maze-solving algorithm that has remained relevant to this day.

The contest went from strength to strength, held in Paris, Tampere, Madrid, and Copenhagen, but for these first few years something struck me as strange. Not one of the winners was trained as an engineer. Great machines came from mathematicians, computer maintenance staff, and programmers for manufacturing industry, but engineers were notable by their absence.

In 1985 I was invited to Tsukuba, to see what the Japanese had made of the contest. There were 200 contestants, but the champion, Idani, was not an engineer in the formal sense. Later that year we took the contest back to the United States—the Japanese funded the trip to put some life back into an old adversary. A future champion was unearthed in MIT—but he was not then an academic; he was part of the laboratory staff.

So, what is it that defines a mechatronic engineer? What is the special aptitude that singled out these champions? What had they learned from their endeavors that was not to be found in a formal engineering course?

They were able to put together a concept in which strategy, computing hardware, sensors, electronics, and motors were blended together in harmony, not as a cobbled assembly of diverse technologies. Therefore we must distill the "good bits" from the diverse range of specializations that make up engineering as a whole.

Mobile robots are a fascinating application of mechatronics. A spinoff of the cooker clock project was the addition to our team of a seasoned researcher—a director of the company—who joined our Portsmouth research group to indulge his obsession with legged robots. Robug I rather ominously looked like a coffin on somewhat wobbly legs. Robug II shed all unnecessary weight and climbed walls. Together with Zig-Zag, it impressed the nuclear industry enough that they started placing orders for the design of robots for specific applications.

While we had been keen to give our robots intelligence, the last thing the clients wanted was for a robot, clambering on a nuclear pressure vessel with an angle grinder in its claw, to start showing initiative!

The market for these robots set a whole new direction for the company, newly emerged from the Tube Investments Group via a management buyout. Portsmouth Technology Consultants was born. I remained a director of the new company, even though by then I had moved to Queensland, Australia.

Ten years later, despite some major European funding for walking robot development, the company failed. The cloud had a silver lining. For scrap-metal prices, we were able to buy for the University of Southern Queensland the latest eight-legged walker, the result of a million dollars or more of development.

Although we had already developed an Australian ceiling walker all of our own, seen worldwide on BBC television, the research interest turned to agricultural applications, in particular to the vision guidance of tractors. With a videocamera, a computer, and a submodule for operating the hydraulic steering system, we were able to steer to an accuracy of better than an inch. The project was a technical tour de force, but a commercial failure. In hindsight, it is clear that the reason for the lack of sales was that we had set the price too low. Yes, too low.

We aimed to sell the system to dealers for $5000, for them to sell on at $10,000. That might appear to be a generous margin, but it was not enough. A purchaser might work a property many hundreds of miles from the dealer. A simple fault might render a quarter million dollar tractor unusable, and the dealer would be called out. After a lengthy journey, the dealer was still likely to be baffled.

A phoenix rose from the ashes of the project. An Australian company started to market a GPS (Global Positioning System) guidance system, one that displayed steering instructions to a human driver, at a price of many tens of thousands of dollars. A demand was swiftly seen for an interface between the GPS system and the actual steering of the tractor. The steering submodule that was a small part of the vision guidance system was just what was wanted. This time the price was set at several times the price of the entire original vision system, and sales were very good.

With a new commercial partner, we will soon combine vision with a low-cost precision GPS technique that we have developed. The project will be rolling again.

Another project with journalist appeal was Robocow—a nimble mobile robot for training horses for cutting contests.

In some ways, as technology advances the task of exploiting it becomes harder. The traditional approach to embedding some computing power was to take a microprocessor chip, add some supporting memory and interfaces, and then write the software "from the ground up." The concept of an "operating system" would be as alien as adding antilock braking to a rollerskate.

But when Webcams can be bought with drivers to interface them via DirectShow to Windows-based applications, how far up the evolutionary tree do you have to go to find your computing power? The price of a fully equipped PC card is today little more than that of an evaluation board for a Motorola HC12. Are we locked into complicated but popular technology "because it's there"? That is certainly the line we have been taking with a deluge of agricultural application opportunities. The data capture is quick and dirty, and we can concentrate on innovating ways to analyze it.

A project that appears strange—but actually makes good sense—is based on the ability to discriminate between animal species. When a sheep approaches a watering place, it is recognized and allowed to pass through a gate. When a feral pig comes the same way, it is also recognized and allowed to pass through an adjacent gateway, to another water source.

The difference is that the sheep will be allowed to go on its way after drinking, while the pig is confined until the farmer comes to pay it some serious attention. The economics of damage by feral pigs and the trade in feral pork are convincing reasons for funding the project.

The dynamic behavior of small marsupials is another area of interest. There is a breeding program for an endangered species of *sminthopsis*. The problem is that if the lady is not "in the mood," the animals are apt to kill each other. By tracking the movement of separated partners in adjoining cages, we hope to detect in real time when true love can take its course.

Texture analysis is usually a lengthy business, requiring substantial computing effort for correlations. Two applications require a speedy solution. The first is for the grading of oranges, where the extent of "goose bumps" on the surface is an indicator of quality.

The second is for the game of football. A speedy analysis of the status of the grass cover must be made, at least to avoid a lawsuit when an overvalued player slips on a bare patch and falls on his fundament. But is this really mechatronics?

So, what of the next generation of mechatronic engineers? How do we give them skill and ability with the essentials, without deluging them with the entire contents of the textbooks of at least three diverse disciplines? The Micromouse experience suggests that hands-on experimentation is an essential ingredient. While learning, software must be "crafted" by the student, rather than being ladled into the project as a bought-in commodity. The student must be prepared to deal with hydraulics or electromechanics, treating them as two sides of the same coin.

After the "bare essentials" whistle-stop tour of mechatronics, some experiments are presented that could whet the appetites of students to study the more detailed material that follows. "Seat of the pants" engineering will certainly get you started, but will go only so far.

Mechatronics is special. It is no more a mere mixture of electronics, mechanics, and computing than a *Chateau Latour* (or *Grange Hermitage*) vintage wine is a mixture of yeast and grape juice.

1.2 WHAT IS AND IS NOT MECHATRONICS?

Long ago, Caryl Capek wrote a book, *Rossum's Universal Robots*. It was as little about robotics as *Animal Farm* was about agriculture, but the term had been coined. Science fiction writers grew fat on the theme, and the idea of mechanical slave workers was lodged in the mind of the public.

When Devol designed a mechanical manipulator for Engelberger's firm, Unimation, it was endowed with the term "a robot arm." As a research topic, robotics ceased to be about tin men and turned to the articulation of mechanical joints to move a gripper or workpiece to a precise set of coordinates. The new "three laws of robotics" concerned the Denavit–Hartenberg transformation matrices, discrete-time control algorithms, and precision sensors.

Robotics is just a narrow subset of mechatronics. It is true that it has all the ingredients of sensing, actuation, and a quantity of computer-assisted strategy in between, but with every day the list of mechatronic products increases. In videorecorders, DVD players, jet airliners, fuel injection motor engines, advanced sewing machines, and Mars rovers, not to mention all the gadgetry that surrounds a computer, the jigsaw pieces of mechatronics are slotted together.

In something as simple as a thermostat, sensing and actuation of the heater are linked. But the element of computation is missing. It is not mechatronic. In automatic sliding doors, however, the criterion is not as cut and dried. A few simple logic circuits are enough to link the passive infrared sensor to the door motor, but the designer might have found that the alternative of embedding a microprocessor was in fact simpler to design and cheaper to construct.

Before 1960, autopilots were capable of automatic landing. Their computational processes were based on *magnetic amplifiers*, circuits using the saturation of a mumetal core with no semiconductor more complicated than a diode. As the aircraft approached its target, the mode switching from height-lock to ILS (instrument landing system) radiobeam to flareout controlled by a radar altimeter was performed by a clunking Ledex switch, a rotary solenoid driving something similar to an old radio waveband changer.

This must come close to qualifying as robotics, but lacking any trace of digital computation, it must fall short of mechatronics. For today's aircraft, however, with digital autopilots that can not only guide the aircraft across the world and land it, but also taxi it to the selected air bridge at the terminal, there can be no question that it is a mobile robot.

Machines that can roll, walk, climb, and fly under their own automatic control have come to share the title of robots, mobile robots. One example of such a robot is the Micromouse, which will be mentioned several more times in this book. *IEEE Spectrum Magazine* and David Christiansen must take the credit for devising a contest in which small trolleys explore a maze. I would like to claim personal credit for redefining the maze design and rules to give victory to the "intelligent" mouse, rather than the "dumb wall followers."

Many early Mice used stepper motors to move and steer them, controlled by microprocessors of one sort or another. The maze walls were sensed by a variety of photoelectric devices, although in at least two cases mechanical "feelers" were used with great success. To navigate through the maze, a map had to be built up in the microcomputer's memory. To solve the maze, a strategy was required. A further aspect of the software was the need to apply control to keep the mouse straight as it ran through the passageways. So, in one not-so-simple contest, all the ingredients of mechatronics were brought together.

The contest runs regularly to this day. Many of the early champions are still at the forefront, while simplified versions of the contest have been developed to encourage young entrants. While the experts hone their expertise, however, the bar has to be set lower and lower for the newcomers. Simply running through a twisted path with no junctions is a testing problem for most schools' entrants.

So, what is the "mechatronic approach"? How would a mechatronics engineer design a set of digital bathroom scales? Would they be based on a strain-gauge sensor, on the "twang" frequency of a wire tensioned by the user's weight, or on some more subtle piece of ingenuity?

When I opened up the machine on my bathroom floor, I was disappointed to discover that the pointer of a conventional mechanical scale had simply been replaced with a disk with a notched edge. As it rotated under the weight of the user, an *incremental optical encoder* counted the notches of the disk as they went by and displayed the count on a luminous display.

For a manufacturing company with an established market in mechanical scales, the "pasted on" digital feature makes sense. However a "truly mechatronic" solution would find a tradeoff between digits and mechanical precision that would simplify the product.

A hairdryer marketed some years ago featured a “bonnet,” coupled by a hose to the hot-air unit. A plastic knob could be rotated to give continuously variable temperature control. So, how would you go about designing it? When the question is put to university classes, it always brings answers featuring potentiometers, thyristor power controllers, and often a microcomputer.

The product was actually much simpler. The airflow was divided into two paths after the fan. In one path was a heating element, regulated by a simple thermostat just “downstream,” while the other simply blew cold air. The ornate knob moved a shutter that closed off one or other flow, or allowed a variable mixture of the two.

Good design can often demand an awareness of how to avoid excessive technology.