4

Introduction to the Next Level

A few basic principles and some experimentation can get you a long way in mechatronics, but the time comes when they must be backed up by mathematical theory. That is not to deny that the empirical approach will get you much further than the theory on its own.

I have tried to feature theory that will actually be useful, rather than mental party tricks that are only really relevant to answering examination questions. We cannot avoid differential and difference equations. They are the essential substance of the things that we are trying to control. We must be able to spy out the state variables that define the behavior of the system and then derive state equations to describe them.

One vital use for the differential equations is to enable us to simulate the systems in software and try out the algorithms, whether based on theory or pragmatism, to obtain an early verdict on the likelihood of the success of the outcome. But unless the algorithm is tried out on a real system, simulation is a mere mental pastime.

We have to include methods of testing for stability, for despite the legion of thermostats happy in their limit cycles, there are many systems where true stability is essential. Transfer functions certainly have their uses, but they are just one piece of the jigsaw puzzle and not an end in themselves. When we use mathematical shorthand to put our state equations into matrix form, we find ourselves led into the world of transformations and eigenvalues and we need to brush up on a mathematical toolkit.

When we step back to the mechanical reality of moving parts, accelerations and forces, we again find that the coordinate geometry is leading us down the matrix path. Rotations become tensors, and the articulation of a robot arm involves a chain of affine transformations, performed very neatly by a few matrix software operations.

The mechatronic engineer should not be in a hurry to dismiss electronic design too lightly, thinking that purchasing ready-made circuitry is an easy answer. The ADC chip that costs under $10 will find its way into a board marketed for many hundreds of dollars, probably with elaborations that make it difficult to do the simple operations you require. The vendor of the board will offer you a FIFO buffer that can hold hundreds of samples. These are useless to you for control; you need just one value of the variable, measured as recently as possible. You may have to put your own board together!

The ability to throw a circuit together around an operational amplifier will be another essential skill. Once again, a chip costing a few cents will usually be packaged in a smart box with a huge price tag, and there is no guarantee that you can actually find the product that does what you need.

Your circuitry skills will be tested even further if you intend to embed your own microcomputer in your design, rather than exploiting a PC. Think long and hard before you select a particular processor. The wide variation does not just cover price, you must also consider the sizes of program and working memory, number and type of interfaces, and the availability of cross-support software. Many devices such as the PIC have a huge hobbyist following, with an abundance of freeware available on the Web. Others can include bus systems aimed at industrial applications such as the motor industry, but these too may have excellent free cross support.

4.1 THE www.EssMech.com WEBSITE

There are things that a book simply cannot achieve, such as real-time simulations, interactive examples, and movie illustrations. Those shortcomings are easily overcome by mounting a Website for readers of the book.

4.1.1 Examples and Simulations

There is nothing like trying out a simulation in real time to see the problems that a control system can really present. But what is the easiest way to present the simulation experience to the reader? Packages seldom come cheaply. They also present problems all of their own in the "learning curve" requiring time to use them to their full advantage.

There is a graphics computing environment already installed on your computer. It is hidden within the browser. It allows you to enter or modify code displayed in a text area of a Web page and execute it at the click of a button.

The language is JavaScript, and it is used here with a simple Java "applet" that puts all the graphic tools within reach. With the title *Javascript On-Line Learning Interactive Environment for Simulation*, examples have already been on show for some time at http://www.jollies.com/.

Although the code looks very much like C, there are some subtle differences. Nevertheless, it is easy to edit or create new code for the Jollies pages both for simulations and for graphic image processing.

Although the Jollies are probably the easiest route by which to approach the examples, many on the EssMech Website are duplicated in QBasic or QuickBasic (the syntax is the same) and in Visual Basic as well.

You will find other code there, too. There are examples of assembly code for embedded microcomputers and "solution" code for the laboratory exercises. That is in addition to some packages to help with vision examples and experiments and the occasional movie of mechatronics in action.

4.1.2 Finding the Code

If you simply enter the URL of the Website, you will find pages that tell you about the book, with links to the examples.

To find a direct path to each example, you must add the subsection number in a slightly cryptic way as follows; The example for this subsection, 4.1.2, would be linked at http://www.essmech.com/4/1/2.htm. Try it! You can also try http://www.essmech.com/4/1/ to see an index of examples in the whole of Section 4.1.