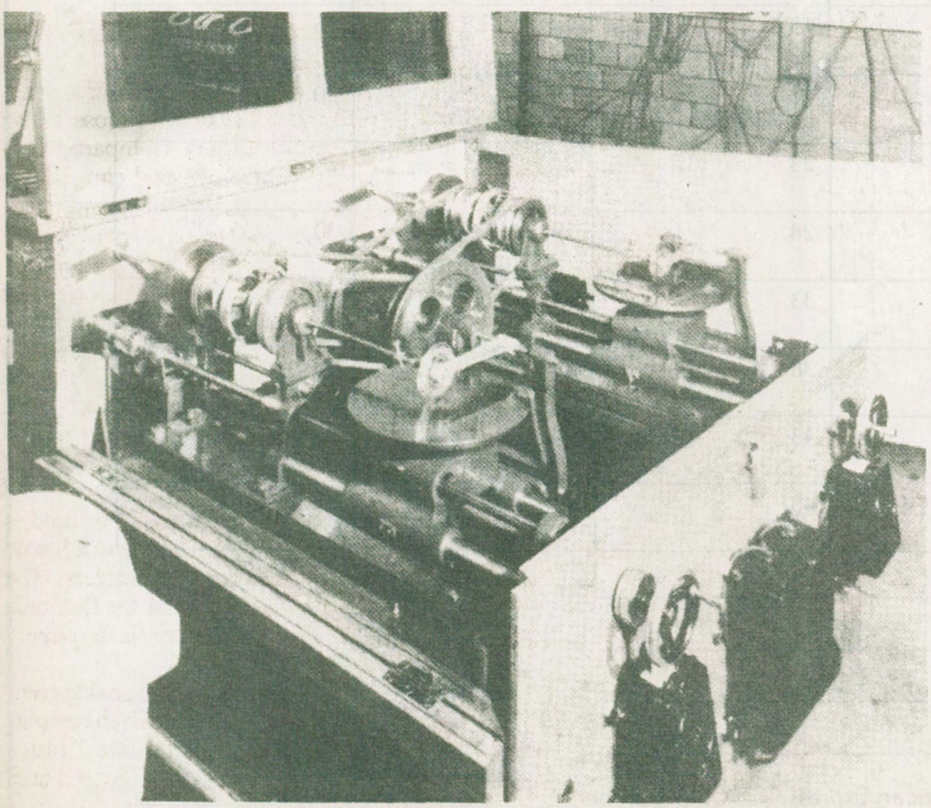
FEATURE

The Analog Computer

The analog computer is not dead, but alive and well and living in industrial applications.

PAUL CUTHBERTSON



The analog computer has been with us in one form or another for some considerable time. Despite this, it could be called the "forgotten computer." Today the public imagination is swamped by notions of word processing, high resolution graphics and digital communications – all (rightly) the domain of the digital computer.

However, there are many analog computers around. They lack the glamour and fascination of their digital counterparts and can be found incorporated into industrial controllers, dedicated to keeping steel at such a thickness and ketchup at such a consistency. Otherwise, they are mostly found languishing in dusty closets.

Analog computers deserve a better fate than this. They are a valuable tool for the scientist, engineer and mathematician, providing a direct means of modelling systems as diverse as control mechanisms, vehicle suspension units and animal populations.

The history of the analog computer is as varied and interesting as that of the digital computer. There were a few mechanical versions around in the 19th century (the slide rule is really a mechanical analog computer and you could argue certain ancient navigational instruments are too) but the first really successful mechanical design arose about 1930 or so in such places as MIT and Cambridge. Electronic versions appeared in the 1940s. RCA built the first accurate design in 1950, since then the advent of integrated circuits has made the design of analog computers easier, in just the same way as digital computers.

Many of the pre-war analog computers had military purposes such as bomb or gun aiming and were very successful. Connected directly to the airspeed, height and heading instruments in the aircraft, even the primitive versions of automatic bombsights were vastly superior to eye alone.

Further improvements used a gyroscope to allow for the aircraft banking and allowed the operator to input a drift rate to compensate for the effects of the wind. Anti-aircraft guns used a "computer predictor" which computed a trajectory for a shell, assuming that the target was holding a steady course, or that any change was at a constant rate.

Mechanical Matters

Mechanical analog computers use the amount of rotation of a shaft or the length of a piston as the variable. Multiplication by a constant is achieved simply by meshing two gears of a certain ratio. Summation can be done by levers.

Integration was performed in an intriguingly elegant manner by a "spinning disc integrator." A roller bears on the surface of a disc which spins at constant speed. This roller is free to move along its axle towards the periphery or the centre of the spinning disc. The shaft of the roller will accumulate a rotation depending on how near the roller is to the periphery of the disc. If the roller is at the centre of the disc, then no rotation occurs. If the roller is moved right over the centre and onto the other side, then the direction of accumulation reverses. Figure 1-4 show some examples of mechanical computer functions.

One of my friends who works in a fisheries research establishment tells me that there used to be a mechanical model of fish populations standing in one corner of his lab. Nowadays electronics has taken over and they use a big VAX computer system for such things.

A digital computer deals with data in the form of discrete numbers and processes these in turn according to a sequence of instructions. The bit-length of the word dictates the resolution. The electronic analog computer represents quantities as voltages. These voltages are analogs to the quantities we wish to represent and vary in a manner analogous to the manner in which the quantities vary.

To make an example of the differences in operation, suppose we fire a shell from an artillery piece and this shell will attain an altitude of 10km before its vertical motion stops and it starts back to earth. In the digital computer we might calculate the altitude of the shell at discrete intervals. If we calculate to the nearest metre, the number 2000 would represent 2km, 1000, 1km, and so forth. A binary word of 16-bits would easily accommodate the maximum altitude of 10km.

However, in the analog computer the altitude of the shell would be represented by a continuously varying voltage — 1V might represent 1km. This is a far more direct method than the digital but each has its own advantages and disadvantages:

- Noise and drift (due to temperature and ageing) and tolerances in the circuitry all contribute errors in the analog computer. There are no such errors in the digital computer, excepting gross fault conditions which cause a bit to change state.
- The digital computer suffers from rounding errors. In fact a small number added to a much larger one can vanish entirely under certain conditions. The resolution of the analog computer is infinite (in any practical sense) and there are no rounding errors. We can minimize rounding errors in a digital system by increasing word length, but then we suffer the cost of extra hardware or increased processing time.

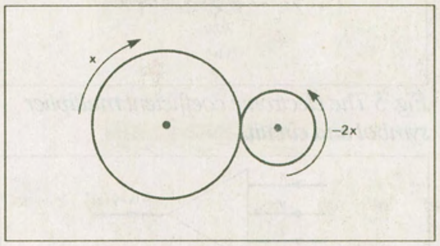


Fig. 1 A mechanical coefficient multiplier using two gears at 2:1 ratio (rotary motion)

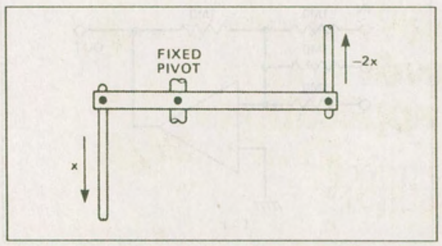


Fig. 2 A mechanical coefficient multiplier using levers (linear motion)

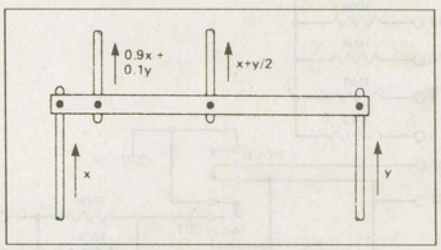


Fig. 3 Mechanical summation.

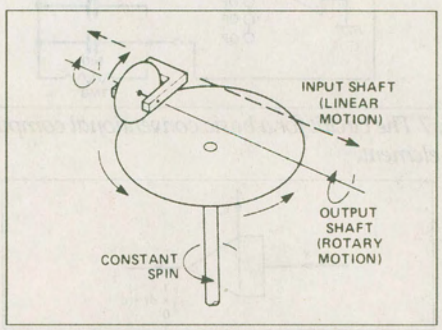


Fig. 4. The principal of the spinning disc integrator.

- The digital computer is an essentially serial device performing primitive operations on fragments of numbers in sequence. This makes for slow arithmetic. An analog computer is inherently parallel. A single summer could take an unlimited number of inputs, multiply each by a coefficient and add them all in a few microseconds. There may be tens or even hundreds of these "computing elements" working simultaneously.
- Results are available continuously from an analog computer. In the digital computer the results will progress by discrete jumps an intervals. A number which may be precise at the instant of its calculation will usually be progressively less accurate until replaced by its successor.
- There is a certain minimum hardware requirement for a digital computer. We have to have a processor, RAM, ROM and IO (even if these are all on the same chip). A useful analog computer which might be used to solve a second order differential equation can be built from a few op amps. The total cost of the components for such would be less than a few dollars. In fact an analog computer model of a filter — a state variable filter — needs three or four op amps, a few resistors and two capacitors. The display for an analog computer can be a meter, an oscilloscope or a DVM.
- The method of interconnection of the analog computer elements is a very direct way of numerically solving systems of equations, even those which might defy analysis. Compared with these methods the digital computer is an abstraction, requiring massive underpinning of languages, operating system and such.
- A sensor such as a potentiometer can be wired straight into the analog computer inputs. The outputs can drive an audio amplifier, or servo amplifiers.
- The operator can interact directly with the analog computer in an experimental fashion — to try things out. This is less easy on a digital computer.
- An analog computer cannot be used as a word processor or the like as it has no way of representing characters. The digital computer is ideal for that task. An analog computer is a purely numeric machine.
- The parallel nature of the analog computer makes testing easy. Each computing element can be tested independently and if needs be ignored until a service is done.
- The digital machine can store information indefinitely. This is not possible on an analog computer.

I would identify inability of the analog computer to store information or to handle text as the two major reasons for the ascendancy of the digital computer. Hybrid machines do exist – where numeric computation is performed by the analog computer and the digital section is responsible for generating functions, for storage of output or for performing any long term integration or summation where speed is not important. Connection between the two parts is via DAC and ADC converters.

Attempting to patch the analog computer connections from the digital computer is a complex business. Interestingly enough, the arrival of a new generation of crosspoint switch chips on the scene a short while ago may herald a more compact and effective hybrid computer.

If you were to see an analog computer and one of the more usual desktop digital computers side by side, the superficial differences would be glaringly obvious. In fact you might not recognize the analog computer as being a computer at all, as all the more usual keyboard, video monitor, printers and disk drives are entirely absent. Instead we might have a large panel on which is an array of sockets, a set of knobs, one or two switches and an analog meter movement (or possibly a simple scope of DVM).

The array of sockets is known as the patch panel and the analog computer is programmed by linking (patching) various of the computing element sockets together, rather in the manner of the old time telephone exchange. The analog computer software is easy to see – it is the wiring on the patch panel. There is no confusion about where the software is on an analog computer.

Let's examine the individual computing elements before discussing how they might be interlinked. The three most commonly used are the coefficient multiplier, the summer and the integrator. Useful work can be done on systems of linear equations with no more than these three types of elements. We built our own analog computer at Aberdeen University recently. It incorporates all these three. Our approach has been slightly unconventional and where there are differences between the Aberdeen unit and the usual case, I'll mention them.

The coefficient multiplier multiplies an incoming voltage by a constant. The coefficient must be between zero and one. Physically the multiplier is usually a potentiometer, with one end connected to 0V (Fig. 5).

When the output of this arrangement is patched to the input of the next element, the set coefficient will tend to droop, due to the next element's non-infinite input impedance. Our own analog computer

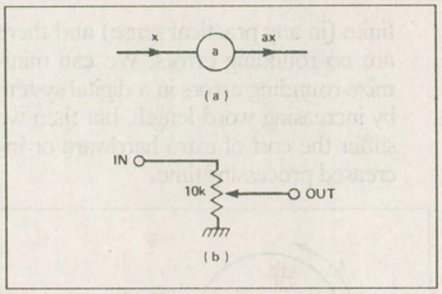


Fig. 5 The electronic coefficient multiplier symbol and circuit.

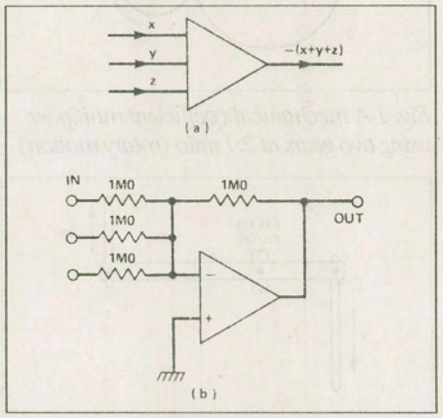


Fig. 6. The summer symbol and circuit.

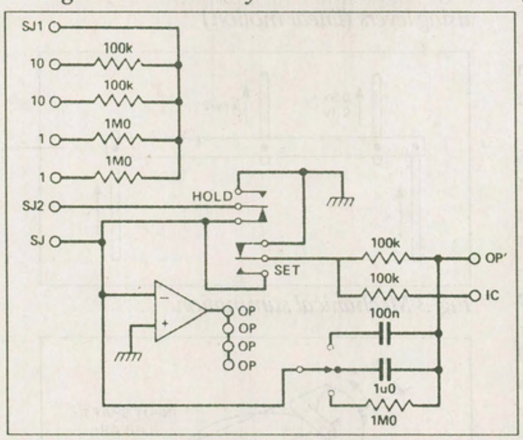


Fig. 7 The circuit for a basic conventional computing element.

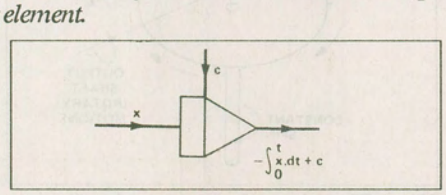


Fig. 8 The integrator symbol.

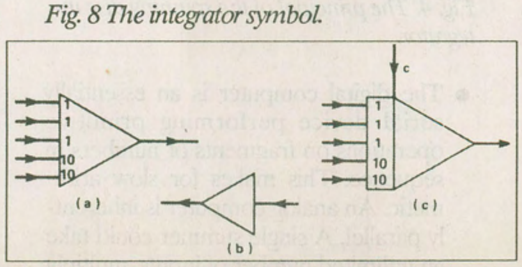


Fig. 9 Symbols for the Aberdeen unit computing elements (a) Summer (b) Inverter (c) Integrator

has the slightly unconventional addition of an op amp buffer after the potentiometer, which does away with this problem. Some of our potentiometers are also double ended – neither end is taken to 0V. This is occasionally useful and again unconventional.

The summer (Fig. 6) takes a number of voltages as inputs and adds them together inverting in the process. The actual circuit consists of a single op amp and a number of resistors. In our version the input resistors are trimmable through a limited range to eliminate initial tolerances. Any practical circuit must also include a nulling potentiometer. The inputs on our version are each tied to 0V via a 10k resistor. This means the input may easily be left open without disturbing the impedance balance of the circuit too much, thus minimizing offsets.

The input resistors are connected direct to the op amp circuit, the general trend being to keep these separate. In a conventional computer this gives access to the virtual earth point and allows the operator to introduce feedback networks other than the one supplied. Figure 7 shows a typical analog computer summer element which illustrates this.

The conventional circuit also doubles as an integrator if you should switch in either of the capacitors, and another element's input resistors could be hijacked if necessary. In our circuit the elements are fixed and trimmed for accuracy, which does not allow this flexibility.

The integrator element integrates the sum of the input voltages with respect to time. If we suppose that the input x is a constant, the output voltage will change by xV in 1 second. Note that there is an inherent sign reversal as in the summer.

The Aberdeen unit is unconventional in that the initial conditions input is not sign reversed. The relays are to do with setting the element to its initial conditions or holding the computation at any point. We chose IC analog switches instead, principally because they do not bounce. Figure 8 shows the symbol for an integrator.

Figure 9 shows the elements of the Aberdeen unit as they might appear on a problem flow chart. The triangle is an inverter. Normally one would press a summer into service as an inverter because of the way our circuit is built, there is a spare inverter with each summer, which is brought out to the front panel. The numbers refer to gains – 10 is a x10 input. Use of stackable hermaphrodite connectors remove the need for the usual multiple outputs on elements.

There are numerous other circuits

which can be used on analog computers. Among the most important we could mention are four quadrant multipliers and the various diode circuits for modelling nonlinearity, discontinuities and hysteresis. In fact, any circuit which behaves in a fashion analogs to a physical system can be pressed into service. None of these non-linear elements are incorporated on the Aberdeen unit ... yet.

So how do we patch those together to produce something useful? We can appreciate what is happening better if we devise a model of a system and set out to solve it. I have chosen the classic mass spring damper model of a car suspension, beloved of generations of long suffering fifth formers ever since Newton. It is not too complex to imagine what is happening in the mind's eye but at the same time it is not a trivial example. Figure 10 shows the arrangement.

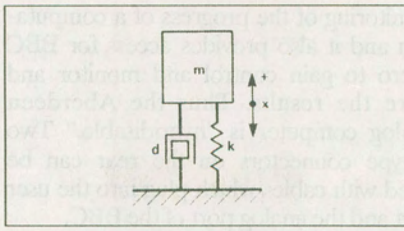


Fig. 10 The mass-spring-damper problem.

The deviation of the spring from its natural (unstretched) length I have called x. This is a distance of course. The rate of change of distance with time is called velocity. The rate of change of velocity with respect to time is acceleration. I have called the velocity x (x-dot') and the acceleration x ("x-double-dot") which is mathematicians' parlance for the derivative and the double derivative of x. Now, you needn't worry about all this calculus. The only important point to remember for this purpose is that integration is the opposite of differentiation.

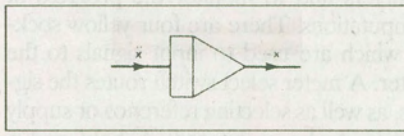


Fig. 11 First steps

As the spring is stretched or compressed, it will exert a force equal to the stiffness times the distance we have stretched it. If we call the stiffness k, the force is kx. So far so good. There is also a force exerted by the damper. The damper only exerts force when we try to move it. If we call the damping factor d, then the force exerted by the damper will be d times the velocity which is dx.

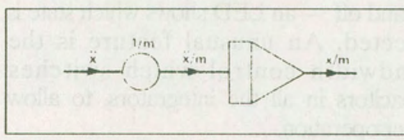


Fig. 12 Accounting for mass

These are the only forces on the mass, so we can add them together to get the total force:

F = -dx-kx.

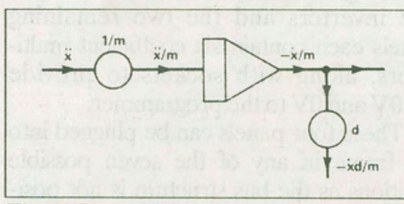


Fig. 13 Damping

There are two important points to note here. We have ignored such complications as air resistance and mass of spring (and a good thing too, I hear someone saying). We also have to decide which direction is positive and I have decided that up is positive. When distance is negative, the spring is compressed and the force it exerts is upward hence the negative sign in front of the spring's force. Similarly when the motion of the mass is downward (negative) then the damper exerts an upward force.

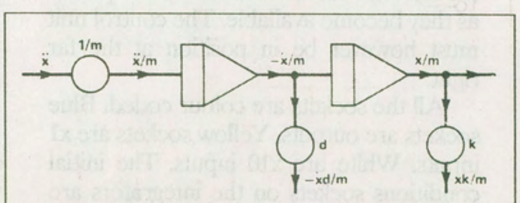


Fig. 14 Adding the spring.

These forces make the mass accelerate. Newton (bless him) said that force is mass times acceleration, so:

mx = -dx-kx

All right then, that's our model of how the system behaves. How to get it into the computer? Let's indulge in some algebra and get m (the mass) out of the way to leave x on its own

x = -dx/m-kx/m

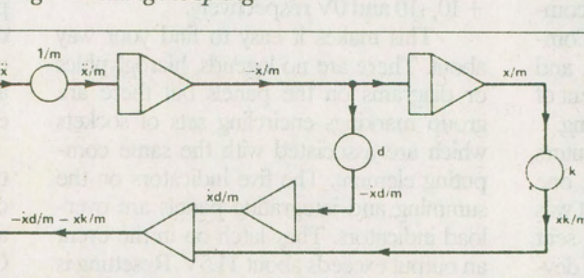


Fig. 15 The complete patch

I mentioned earlier that integrating is the opposite of differentiating so if we fix up an integrator as in Fig. 11, it's a good start. We get -x out (remember the sign inversion).

If I integrate a constant times x I will get the same constant time -x. So, if I put in a coefficient multiplier set to 1/m as in Fig. 12, we can see the result. Then we can add in a coefficient multiplier for d (Fig. 13) and then another integrator and coefficient multiplier for k (Fig. 14). Finally we can add these two in a summer (Fig. 15). It's fairly easy to see how the patching is built up. Figure 15 shows the "open loop" flow diagram for the problem.

But there's still one last thing. Where do we get x from in the first place? Lo and behold, we have what seems to be the right thing coming out of the summer. We can make the left hand and right hand sides of the equation equal if we connect the input and the summer output together as shown by the loop in Fig. 16. This is the closed loop flow diagram and is the patch that we need to solve the problem.

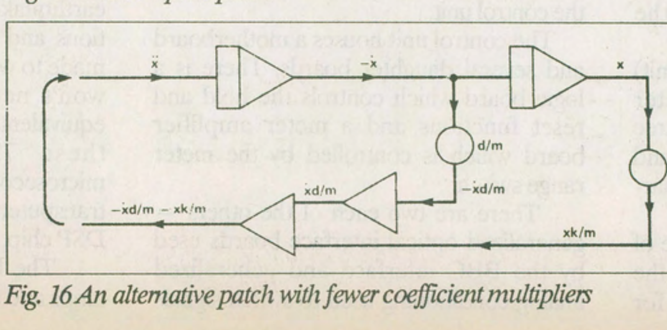


Fig. 16 An alternative patch with fewer coefficient multipliers

Provided we've got the plusses and minuses right, the solution is a decaying sine wave. Mathematically it's possible to have a "draft damper" which assists motion instead of retarding it. That gives an increasing sine wave. It's also possible to have a "silly spring" which pushes in the wrong direction as we stretch it. The solution in this case would probably be an exponential (depending on the ratio of k and d).

This problem is quite easy to solve analytically. The analog computer really comes into its own where we encounter sets of differential equations which are difficult to analyse. These are no more difficult in principle to solve on an analog computer. For example air resistance, double acting dampers, spring masses and the like can all be built in. All we have to do is derive a set of equations which describe the system. We can build several separate models and interconnect to feed the results of one into the next.

The model we have just used does not account for gravity or a "bumpy road". We can add in any acceleration we like at the summer, including that of gravity. We can connect an oscillator to the same place, to inject "bumps." (This oscillation is known as a forcing function). This illustrates the direct nature of working with an analog computer.

So far we have not attempted to quantify the settings of the pots. To get a useful quantitative result we must scale the problem. Ideally the model will use full dynamic range of the machine (+10V in our case) without going appreciably outside those limits (which may cause clipping and invalid computation). It's a similar problem to that encountered by anyone confined to integer arithmetic or the user of a slide rule. The slide rule user has to keep track of all the zeroes or he will end up a factor often-to-the-something out. Similarly, the integer user may run out of bits.

I don't propose to go into scaling in any detail, except to say that there are well defined procedures for doing it which consist basically of writing out an equation for each computing element, estimating the maximum value a variable can be expected to take and dividing through, calculating the pot settings as we go. Some operators get by using try-it-and-see methods.

Anyone who is particularly interested in the rigorous scaling of problems is recommended to read "Systematic Analog Computer Programing' by Charlesworth and Fletcher which gives a detailed treatment of this and other facets of analog computing.

My own interests in analog computers started when I was asked to look at one which appeared faulty. Unfortunately it was an extremely poor design and was sent packing. Subsequently we decided to develop our own system. The photographs shows views inside and outside the machine.

On the right is a panel (the control unit) which contains a large analog meter movement, three rotary switches, three push buttons and a variety of lamps and 4mm sockets. On the left are four narrower panels.

The control unit is the nerve centre of the computer. As well as controlling the hold and reset functions, it allows for monitoring of the progress of a computation and it also provides access for BBC micro to gain control and monitor and store the results. Thus the Aberdeen analog computer is "hybridisable." Two D-type connectors on the rear can be fitted with cables which plug into the user port and the analog port of the BBC.

The meter is used to set up the potentiometers and to monitor the progress of computations. There are four yellow sockets which are used to input signals to the meter. A meter select switch routes the signals, as well as selecting reference or supply voltages to be monitored. A hold and a reset button toggle the hold and reset states on and off — an LED shows which state is selected. An unusual feature is the bandwidth control which switches capacitors in all the integrators, to allow faster operation.

Of the four smaller panels, one contains five integrators, one has five summers and five inverters and the two remaining panels each contain six coefficient multipliers, along with sockets to provide +10V and 0V to the programmer.

These four panels can be plugged into the frame in any of the seven possible positions as the bus structure is not position sensitive. The three spare slots allow the introduction of similar or other panels as they become available. The control unit must however be in position at the far right.

All the sockets are colour coded. Blue sockets are outputs. Yellow sockets are x1 inputs. White are x10 inputs. The initial conditions sockets on the integrators are brown. The red, black and green are for +10, -10 and 0V respectively.

This makes it easy to find your way about. There are no legends, hieroglyphics or diagrams on the panels but there are group markings encircling sets of sockets which are associated with the same computing element. The five indicators on the summing and integrating panels are overload indicators. They latch on in the event an output exceeds about 11.5V. Resetting is by a common pushbutton marked OVV on the control unit.

The control unit houses a motherboard and several daughter boards. There is a logic board which controls the hold and reset functions and a meter amplifier board which is controlled by the meter range switch.

There are two each of the others — generalized optical interface boards used by the BBC interface and generalized analog conditioning used to switch signals or to attenuate and shift the normal +10V range of the analog computer to suit the BBC ADC inputs.

The power supply board is on the rear panel of the frame, along with the transformer, filter, rectifier and reservoir capacitors which are all off board. This power supply performs well. No voltage deviation registers 4-1/2 digit DMM when full load (500mA) is applied. I couldn't believe it at first. No current limit is necessary as the supplies are not available externally. The supplies are +15V for the analog circuitry and +7V for the digital circuitry, which is all CMOS. The 0V line is not a supply, and does not carry supply currents. It is purely a reference. This also helps lessen noise. An interesting feature of the supply is its sequencing. The 15V rails cannot come right up until the 7V rails are established. This prevents damage to the CMOS analog switches.

The master references are on this board too. These are trimmed to within 1 millivolt. We can claim 10.00V in fact, or 0.01% accuracy. The supplies are trimmed to within a few millivolts too. The primary reference is a band gap diode and the setup is remarkably stable in the long term. Each reference socket is individually buffered to prevent loading of the master reference. Incidentally the integrators and summers on the Aberdeen unit are trimmed to within 0.01% as well. It's quite possible to set up the zero point, with the aid of a decent DVM, to within a few tins of microvolts. However, having said that, the time constant on the integrators is the very devil to set up accurately.

This, like most analog computes, has proven to be a valuable tool. Even in these days of fast digital arithmetic, the analog computer should not be despised or cast aside. It offers direct, easily interpreted evaluation of problems.

One of my little projects for the near future will be to make up a dedicated analog computer which multiplies six variables by a coefficient matrix, giving six outputs. One of my colleagues will use it to investigate the motion of buildings during earthquakes. It has to perform 36 additions and 36 multiplications. It could be made to work at up to 100kHz (although it won't need to in this instance). An equivalent digital system would need to do these 72 operations every five microseconds to keep pace — a good few transputers worth may be, or a very fast DSP chip, plus converters, etc, etc.

The hardware I'm using? A few op amps and a few dozen resistors. ■