

# FEEDBACK

## Newton's Principia

I am astonished that T. Theocharis and M. Psimopoulos (August letters) should have interpreted my piece on Newton's *Principia* (February issue) as some kind of belittlement of that great work. The item was written to draw attention to and celebrate this scientific masterpiece, as any balanced reading of it will show. Einstein's relativity was mentioned once, and in parenthesis, purely as a matter of historical perspective.

In quoting Bertrand Russell's remark that force is "a mathematical fiction, not a physical entity" (from his *Principles of Mathematics*) I was certainly not trying to make any point about relativity, but rather a point about reality.

Before Newton defined force in the way he did, I imagine most people used the term, and understood its meaning, in much the same way as we do today in common parlance: the force of an explosion, the force of a blow, the force in a mechanical press, etc. These common-or-garden phrases, and the ideas behind them, represent our direct experience of energy as it manifests itself in the real world.

The world is as it is, and no amount of human analysis, definition, or verbal or mathematical description will make any difference to it. As soon as we invent a precise scientific definition like that for force we are producing an abstraction, for force cannot *exist* in isolation from the real phenomena of the world – just as voltage cannot *exist* in isolation from real electrical energy, temperature in isolation from real heat, or speed in isolation from something or other real moving about. These physical quantities are all abstractions, products of the mind, or *a priori* concepts. In that very abstraction lies their usefulness.

If, as your correspondents suggest, a champion boxer hit somebody (a relativist) in the face with considerable force (say 100 newtons), but a 'pure' force independent of any specific physical phenomenon, as in its scientific definition, if that were possible, the victim would not feel anything because there would be no energy delivered, just a "mathematical fiction".

I was under the impression that the inevitable chasm between the wonderful, brute reality of the world and the tidy mental framework that humans seek to impose on it was well understood by all scientists, artists, poets, philosophers and others who strive to express, describe or explain. Forgive me, then, if your correspondents make it necessary to labour the obvious fact that the map is not the territory.

Tom Ivall  
Staines  
Middlesex

## Oscillator amplitude stabilization

In the July issue, R. Shankar invites comments from readers to his proposed method of obtaining a ripple-free, rectified output from a sinewave oscillator, for use in the amplitude control circuit. A quadrature signal of the same amplitude is derived, the two signals are squared and then added. Since  $\sin^2 x + \cos^2 x = 1$ , the output will be d.c. proportional to the square of the peak amplitude.

This technique is not new; it is described in Burr-Brown's "Applications of Operational Amplifiers" from 1972. I tried the idea a few years ago and found that in spite of its apparent elegance, it is not a very practical method. As pointed out by Mr Shankar, squaring circuits are expensive and, for a low distortion, they also require careful offset and gain trimming, in order to completely balance out all a.c. components.

There is another principle, however, which may be put into practice without too many problems. What is required is a voltage which represents the waveform amplitude, which has no a.c. component in the steady state, yet still tracks the sinewave envelope during transient states with a minimum of delay. These conflicting requirements rule out the use of a straightforward rectifier. A sample-and-hold circuit, however, with a sinewave supplied to its input, will store the peak value of the waveform if its control terminal is driven by narrow pulses derived from the zero-crossing of a cosine (quadrature) waveform.

A fast-acting and accurate control loop is established as follows. The staircase envelope supplied by the sample-and-hold circuit is subtracted from a d.c. reference. The difference (the error voltage) is fed through a circuit which provides both the integral of the error voltage and the error voltage itself, the sum of these controlling the loop-gain in the feedback oscillator. By properly choosing integration time and gain constants, the amplitude can be made to settle within a few cycles of oscillation. With a well-designed amplifier, the distortion obtainable depends solely on the linearity of the gain control element, and figures below -90 dB (0.003%) are not unrealistic.

The principle described here was first put into commercial use in the Radford Low Distortion Oscillator (type LDO3) which I designed in 1972. This instrument was based on the parallel-T resonator while subsequent versions have employed the more modern state-variable filter structure, still with a sample-and-hold control loop.

Jens Langvad  
Solrod Strand  
Denmark.

## Flow diagrams

The article by David Sweeney on flow diagrams in your August issue rather underlines the drawbacks of using flowcharts instead of promoting their use. Flowcharts do not lend themselves to structured programming and, although his examples may be more flexible than usual, they are just as poorly structured.

Mr Sweeney tries to show that his method clarifies decision nesting (Fig.1.). In doing so, he compares his example to a Pascal fragment which he has listed without indentation. There is no way of working out which BEGIN goes with which END just by looking at it, so the nesting is indeed hidden. However, this fragment may be written out with normal indenting and, (surprise, surprise) decision nesting becomes quite clear again.

Mr Sweeney mentions escape routes and gives a really poor example as an illustration (Fig.4.). I have rewritten this example in pseudocode, which shows up two points: (a) there is a logical error in his example – if

Z is 20 then control passes to a test for  $Z > 50$ ; and (b), written out in this form, there is no need for an escape route at all. The use of an escape route as indicated is suspiciously like the Basic habit of sprinkling GOTOs around instead of trying to structure the code.

I would contend that pseudocode techniques may be of more benefit to well structured programs than Mr Sweeney's altered flow-chart method. As a final comparison, I give a pseudocode version of the CIRCGEN algorithm used in Fig.5 of the article, which was itself shown against the original type of flow chart. I wonder which readers would prefer.

I too started off learning computing with flow charts – the pseudocode came later. These days, I use pseudocode for all my programming projects in both high and low-level languages. Coupled with top-down design techniques, readable programs that work can be produced. How many people could adopt Mr Sweeney's method and say the same?

**Example PASCAL fragment, rewritten with normal indentation:**

```
IF B > 27 THEN
  BEGIN
    IF I < 50 THEN
      CALCVAL(Q)
    ELSE
      BEGIN
        B:=9
        I:=5
      END;
  END;
ELSE
  BEGIN
    B:=B+1
    C:=I+B/5
    X:=B*7
    CALCZFACT(X)
  END
```

**Fig.4, rewritten in pseudocode:**

```
BEGIN
  set A to B*2
  IF Z=20 THEN
    IF Z≤50 THEN
      set C to Z/7
    ENDIF
  ELSE
    set A to A/9
  ENDIF
  set I to I+1
END
```

The above is exactly equivalent to the flow diagram, logical error included.

Stewart Ross  
Inverness