in such a way that it could provide both range and velocity data continuously at distances up to 400 ft. These could be sent by radio telemetry to the ship, where they could be displayed on dials mounted in a portable receiver which is carried by the pilot. An audible warning device could be fitted in the receiver which would operate whenever the velocity of approach exceeds the maximum safe value.

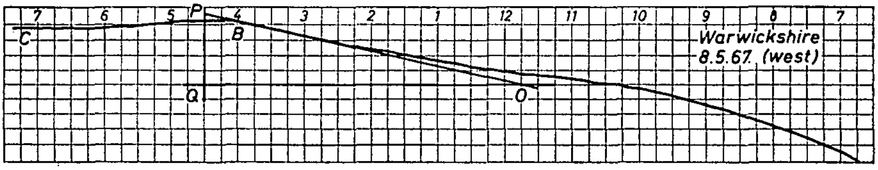

FIG. 2. Typical tracing during the berthing operation

A brief summary of the characteristics of the 'Hecta' is as follows:

Ranging accuracy: 3 per cent of range or 6 in., whichever is the greater.
Power consumption: 100 mW. from internal mercury cells.
Display: Meter of 2 mA. sensitivity for F.S.D., together with recorder. The two instruments are series-connected.
Weight of instrument case: 4·5 lb.
Ultrasonic carrier frequency: 177 kHz.
Transducer beam width between the −6 dB. points: 25°.

The type of recorder in most general use in the berthing application uses pressure-sensitive paper and has a chart width of 2¼ in. The chart runs at a speed of 3 in. per minute.

Rhumb-line Failings

J. S. McKenzie

(Marine & General Computer Consultancy (I.O.M.) Ltd.)

TURNER has chosen in a recent paper[1] to criticize Moss[2] for inaccurate methods and observes that the inherent inconsistencies are 'educationally objectionable'. However, he who criticizes must be prepared to be criticized in turn, and when a computer program of the standard evidenced in the printout of PROGRAM 2 is put forward as a general solution to the rhumb-line problem, and one which will enable the navigator to compute his tracks rapidly and accurately, we must needs examine it closely. Such an examination shows that, at the very least, the programmed solution offered is 'computerizationally objectionable'.

Throughout his paper Turner seems to infer that the computer is a panacea and non-technical readers are presumably expected to express awe when told

https://doi.org/10.1017/S0373463300047871 Published online by Cambridge University Press

that, in the first short program (PROGRAM 1), the computer evaluates $L(\phi)$ 5400 times. It could be observed that if the loop parameters in statements 28 (+1) and 28 (+3) were changed to:

```
DO 20 J=1, 1000
```

and

```
DO 10 I=1, 1000
```

$L(\phi)$ would be evaluated one million times.

However, it is when we turn to examine PROGRAM 2 with a critical eye that we find several baffling oddities.

(a) No comments at all are included in the listing. Surely these are necessary to inform the users of the program what sign convention is to be adopted, or whether N.S.E.W. can be used, as the author omits these details from his description. Furthermore he does not tell us how data is to be punched on the cards. Most programs which are to be used by others have some comments to enable a new user to follow the program's organization.

(b) Study of the program shows that latitude and longitude are entered in whole numbers of degrees and minutes—no seconds or decimals of a minute allowed! Yet $L(\phi)$ is worked out with considerable accuracy, quoted figures being to nearest 6 ft. (the third place of decimals is unrounded, viz. statements 10 and 20 in PROGRAM 1). Does the author imagine that the vessel is always going to be conveniently situated at the intersection of a whole minute of lat. and long.?

(c) The statements where the constants B, C, D, F, and RADC are evaluated would make most programmers shudder. These could be calculated once and for all and fed into the program as Real constants—which is what they are. If the compiler being used has a DATA statement then this should be used as it is more efficient.

(d) Statements 3 (−1) to 6 inclusive are somewhat peculiar. First we test to see if the current value of latitude is negative, and if it is we negate the corresponding value of $L(\phi)$ for use in a later test. Incidentally, floating-point multiplication by −1 is considerably more time-consuming than simple negation, i.e. $A = -A$ should be used rather than $A = A*(-1.)$. So far so good. However, we then test the latitude again to see if it is still negative, and if it is we negate the corresponding mer parts. Perhaps the programmer doesn't trust the computer and suspects that somewhere around statement 4 it will slip in a minus sign!

One would have thought the program should look something like:

```
  IF (RLAT) 3,6,6
3 ARCL (I)  =  -ARCL (I)
  XMERP (I) =  -XMERP (I)
6 CONTINUE
```

One could go on line by line but perhaps we could leave this *pia fraus* by observing that Format statements 21 and 22 appear to be identical although, to be charitable, the printout is a little blurred so they may not be the same. If they are, one wonders if the compiler had hiccups.

Looking at the broader aspect it could be observed that, when a G.P. computer becomes one of the navigator's everyday tools the size of core available, and the time-sharing demands likely to be made upon it, will necessitate every program being tailored and optimized to cater for space and time constraints. One obvious use of a course calculation program would be to make it continuous, using data derived from the navigation sub-system, with the output controlling the auto-helm. Such a program must be constructed *from the start* to allow for upward compatibility. It is also unlikely that a full U.S.A.S.I. Fortran compiler would be available, so Mr. Turner's logical IF statements would be igNORed. It may well be that the program would have to be written in Assembly language.

The message is clear.

(a) That a rhumb-line calculation may use Sadler's formulae[3] can be accepted.

(b) That the navigator will 'compute his tracks rapidly and accurately with a program *such as this*', is to be doubted.

To be constructive, it is suggested that any contributor to the *Journal* should polish his programs as he polishes his prose because many more people are in a position today to judge the result. Very few contributors would submit an article which contained a mathematical proof twice as long as it need be, so there seems to be little reason for including an inefficient program—it can always be omitted.

One last observation. If a G.P. computer becomes available, given the exception of avoiding a storm centre, what navigator in his right mind would use anything other than a great, or composite great, circle sailing?

REFERENCES

[1] Turner, R. J. (1970). Rhumb-line sailing with a computer. This *Journal*, **23**, 233.

[2] Moss, B. J. (1969). Exploration with a computer on rhumb-line sailing. This *Journal*, **22**, 242.

[3] Sadler, D. H. (1956). Spheroidal sailing and middle latitude. This *Journal*, **9**, 371.

Application of Omega to Aircraft Navigation and Traffic Control

E. R. Swanson

(*Naval Electronics Laboratory Center, San Diego, California 92152*)

THE many references to Omega in your excellent discussions on Navigation and traffic control published in the January issue were particularly gratifying to one who has worked on Omega for the past decade. Especially memorable was Mr. Stringer's statement that '[Omega] satisfies the three R's—reliability, redundancy and range' and Mr. Jones' observation '. . . [Omega] is already the most widely deployed ground-based navigation aid in the world by a very substantial

https://doi.org/10.1017/S0373463300047871 Published online by Cambridge University Press