the electronics by John Watson and Richard Young. A paper by Walker describing this unusual photometer has previously appeared in the Journal,[5] and an outline of how this photometer works is also given in Part I of this paper.[1]

The APT's prototype JEAP differs from the production version in two particulars. First, the prototype APT JEAP has a parallel interface which is connected to the 1-MHz Bus of the BBC microcomputer, whereas the production version of the JEAP has a serial (RS232) interface, which would be connected to the BBC microcomputer's RS423 port. Second, we do not have the timer board which is now supplied as standard with the JEAP, and instead we make use of the BBC microcomputer's internal timers, as already described.

Walker, who supplies the JEAP, has details about how to interface it to home microcomputers, including the BBC and the IBM PC.

References

1 Ells, J. W. and Ells, P. E., *J. Br. Astron. Assoc.*, **99**, (6), 282 (1989).
2 Knight, B., *J. Br. Astron. Assoc.*, **90** (1), 45 (1979).
3 Ells, P. E., *J. Br. Astron. Assoc.*, **96** (4), 204 (1986).
4 Bray, A. C., Dickens, A. C. and Holmes, M. A., *The Advanced User Guide*, Cambridge Microcomputer Centre, 1983.
5 Walker, E. N., *J. Br. Astron. Assoc.*, **97** (1), 30 (1986).

Received 1989 January 25; accepted in revised form 1989 March 29

Part III: Software

The software, written to control the simple APT,[1,2] is discussed in non machine-specific terms. It is shown how the design requirements of flexibility and the ability to cope with poor weather conditions have been met.

Introduction

Designing software to control an APT is a large undertaking and will continue to occupy us for some time to come. The system is now functioning reliably and producing good results. Future work will build upon that already done, rather than involve substantial changes, so it is appropriate to describe the software in its present state.

Writing such software involves some special difficulties. From about the time when the procedure to centre stars is written, bench testing is no longer possible, and one must test the system *in situ*. This means that program errors have to be diagnosed and corrected under onerous conditions. The software must not only control the telescope and collect data for the observer, but must also display plentiful information on the computer screen for the benefit of the programmer. For instance, should the system not find a star, the programmer would want to know which procedure failed and also where it had failed. Furthermore, setting up the telescope depends on a complex interaction between computer, observer and telescope, and so the best method of doing this cannot be predicted in advance but must be found by experience.

Certain topics below are treated in depth, particularly if they are original (such as the interrupt driven tracking), or contain refinements needed to cope with the uncertain English weather. Other topics are omitted; for instance the data analysis routines, which involve only standard programming techniques, and which are similar to programs written for our previous system.[5] The programs discussed (with the exception of AcornSoft's Database), will be made freely available.

Combining tracking and slewing

With only a single stepper motor on each axis of the telescope, there arose the problem of how to combine tracking and slewing. An unsuccessful technique we initially tried was to have an interrupt service routine,[4] activated ten times per second (the sidereal rate), which sent a tracking pulse to the right ascension stepper motor. Simulation testing, with the telescope indoors, we found that the telescope would track accurately when pointing at a single star but would drift badly in right ascension when slewing between stars. This was because slewing pulses could be sent arbitrarily close to tracking pulses, resulting in pulses being lost by the stepper motor.

Soon we found an excellent software solution, in which tracking is done transparently. By this we mean that, having written the tracking routines, we could thereafter ignore the rotation of the Earth in writing the Main Program. This solution requires three simultaneous processes going on inside the microcomputer, and two variables, which are shared between them.

The two shared variables are called ra–steps, and dec–steps, and represent the pointing error of the telescope in right ascension and declination respectively; the telescope is pointing in the correct direction only when ra–steps and dec–steps are both zero. Because these variables are shared, any process accessing either has to lock the other processes out by temporarily disabling interrupts.

The three processes are:

(1) The Sidereal Process. This is an interrupt service routine, activated at the sidereal rate of 10 Hz, which decrements the ra–steps variable.

(2) The Slewing Process. This interrupt service routine is activated at the slewing rate (for us, typically 400 Hz). It accesses ra–steps and dec–steps to see whether or not they are zero. If they are not, then they are moved one step closer to zero, and an appropriate pulse is sent to each motor. Three examples should make this clear:

(i) Both ra–steps and dec–steps are equal to zero. In this case the slewing process has no effect.

(ii) ra–steps = 0, and dec–steps = 24. In this case dec–steps is set equal to 23, and the declination motor is made to go one step anticlockwise.

(iii) ra–steps = 17, and dec–steps = −10. In this case ra–steps is set equal to 16, and dec–steps is set equal to −9; the right ascension motor is made to go one step anticlockwise, and simultaneously the declination motor is made to go one step clockwise.

(3) The Main Program. This interracts with the other two processes via three procedure calls:

(i) PROCrequest–ra (*x*). This requests the telescope to slew in right ascension by *x* steps. The body of this procedure disables interrupts while it increments ra–steps by *x* (*x* may be positive or negative), before returning. The Slewing Process will then immediately begin to carry out this request.

(ii) PROCrequest–dec (*y*). Similarly, this requests the telescope to slew in declination by *y* steps.

(iii) PROCawait–arrival. When this procedure is called, the Main Program pauses until both ra–steps and dec–steps are equal to zero. It consists of a loop which repeatedly examines these variables, temporarily disabling interrupts at each iteration.

A typical slew request might be to go 450 steps in right ascension, and −67 steps in declination. This could be achieved by the program sequence:

```
1000 PROCrequest–ra (450)
1010 PROCrequest–dec (−67)
1020 PROCawait–arrival
```

The solution has many good features. There are no complicated timing loops in the Main Program. The Slewing Process 'evens-out' pulse requests, ensuring that steps are not missed. The telescope moves by the shortest route to the desired position. (For instance, in the above program segment, the telescope will slew diagonally for the first 67 steps.) Finally, when slewing eastwards, the telescope does not reverse direction for a step each time it receives a tracking pulse.

No allowance has been made for ramping the stepper motors to a higher speed, because this is not appropriate to our application. For suitable applications, it would be straightforward to modify the method to allow for ramping. (A separate slewing process would be needed for each axis, each process working at a rate which depended on the value of its shared variable.)

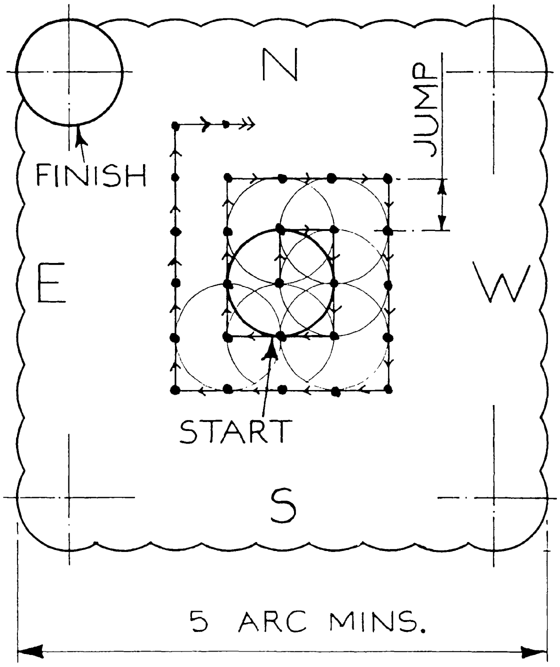


Figure 1. The spiral search used to locate a star, should it not be found in the pinhole after slewing to its nominal position.

Centering

The centering algorithm which we decided upon was originally intended to be a rough method, to enable us to test the Main Program. It turned out, however, to be so fast (taking only ten seconds), accurate (to about five arcseconds as determined by repeatedly centering on the same star) and robust (enabling stars to be centered even through thin cloud) that we have not needed to modify it.

It is similar to those of previous APTs,[5] but differs in some important details, in particular, in the self-adjusting thresholds which make it robust. A prerequisite for the algorithm to work is that, when the telescope has moved to the nominal star position, the star is somewhere in the pinhole.

The centering method is as follows:

(1) The telescope 'creeps' (as described below) eastwards, until the star is no longer in the pinhole. The position of the edge of the pinhole is noted, and the telescope returns to the nominal star position.

(2) The telescope creeps westwards until the opposite edge of the pinhole is found.

(3) The telescope moves to the point mid-way between the two edges of the pinhole. The star is now centered in right ascension.

(4) A similar procedure is carried out in declination. The star is now centered.

The 'creeping' consists of a succession of alternating hops and peeks; where a hop is a movement of 0.1 pinhole diameters across the sky, and a peek is a photometer reading of half a second duration. The peek reading is compared to a threshold brightness, to determine whether or not the star is still in the pinhole. (All brightness measures are reduced to photons per second, so that comparisons are valid, despite possibly differing durations.)

1990JBAA..100...30E

For our APT, the pinhole used is 60 arcseconds in diameter, and so each hop must move the telescope across the sky by 6 arcseconds; also, each stepper motor step turns the telescope through an angle of about 1.5 arcseconds about each axis. This means that, in declination, 1 hop = 4 steps of the declination stepper motor but in right ascension an extra factor arises because hour circles converge towards the celestial poles. Here 1 hop = INT (4/COS(DEC)) steps of the right ascension stepper motor, where DEC is the declination of the star.

The centering routine must continue working in poor sky conditions, so the threshold must reflect recent sky behaviour. We define threshold = (STAR+SKY)/2, where STAR reflects the recent behaviour of the star brightness, as viewed through the pinhole, and SKY similarly reflects the recent behaviour of the nearby sky. To obtain such values for STAR and SKY, a moving average of brightness readings for each target is kept. This moving average is updated after every full 40 second reading, using the expression

MAT = (oldMAT + BRT)/2

where MAT is the new moving average brightness for the given target, oldMAT is the previous value, and BRT is the brightness reading just obtained for the target – all in photons per second. With this expression, the influence of any given reading, on the moving average, decays by one half as each subsequent reading of the same target is made.

Spiral search

For the first six months of operation, the system worked without a spiral search. During the observer set-up it was sometimes found that, for various reasons, the observer had not pointed the telescope sufficiently accurately at all stars. On several frustrating occasions, set-up had to be repeated four or more times. Furthermore, during this period, stars were occasionally lost after about half an hour of automatic observing. On examination of the data, these latter failures could also be attributed to inaccurate star positions, caused by faulty set-up.

It was therefore decided to include a spiral search, mainly for the purpose of simplifying set-up. Having done this, it was an easy matter also to include this search in the automatic observing sequence.

Once automatic observing begins, a spiral search is a rare occurrence, not being required at all on many nights. When it is required (say if the telescope should be displaced by a strong gust of wind), then it is invaluable in preventing many hours of data being lost.

The spiral search is illustrated in Figure 1. The size of each jump is equal to the radius of the pinhole, as this allows sufficient overlap to ensure that the star is not missed. As with centering, the number of steps in each jump in right ascension depends upon the declination of the star (1 jump = 5 hops), and each jump is followed by a peek of half a second.

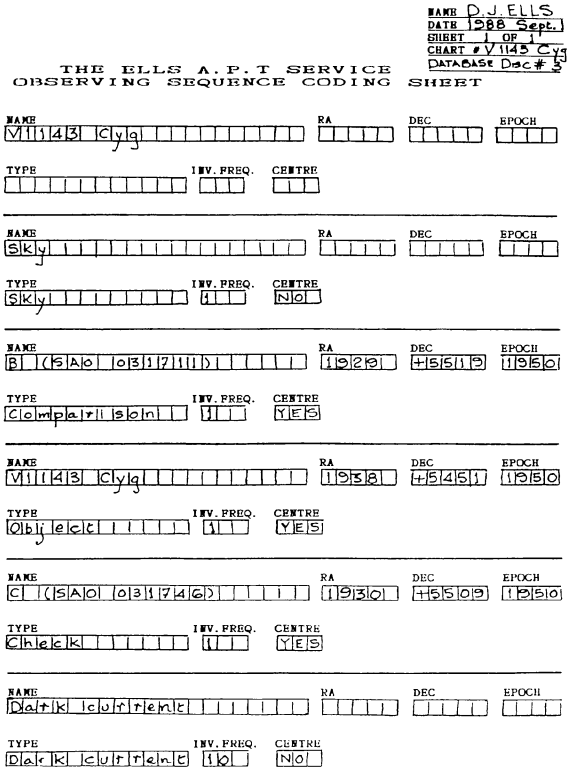
A spiral search is regarded as being successful if and only if (i) the star is found to be within the pinhole before four spirals have been completed and (ii) this star is successfully centered. The second condition is very important in preventing the telescope being misdirected to a bright patch of cloud. If a candidate star is found, but the computer fails to centre it, then those computer variables that represent telescope and star positions must be restored to the values they had before the search began.

The complete spiral search takes a minute to complete, but very often a star will be found in the first spiral, which takes seven seconds.

Preparing observing sequences

In PEP, the observing sequence may be broken down into a succession of cycles,[6] almost identical, but having slight variations. An observing cycle might be, say: sky, comparison, object, check (every other cycle), object, dark current (every tenth cycle). In this example, the cycle length is six; but there are only five distinct targets, because the object has repeated readings taken, at cycle positions three and five.

With an APT, observing sequences must be decided in advance. We wanted the ability to choose observing sequences at will, and our solution involved entering observing cycles on coding sheets (Figure 2). There is



NAME D.J. ELLS
DATE 1988 Sept.
SHEET 1 OF 1
CHART # V 1143 Cyg
DATABASE Disc # 3

THE ELLS A.P.T SERVICE
OBSERVING SEQUENCE CODING SHEET

NAME V1143 Cyg RA DEC EPOCH
TYPE INV. FREQ. CENTRE

NAME Sky RA DEC EPOCH
TYPE Sky INV. FREQ. 1 CENTRE NO

NAME B (SAO 031711) RA 1929 DEC +5519 EPOCH 1950
TYPE Comparison INV. FREQ. 1 CENTRE YES

NAME V1143 Cyg RA 1938 DEC +5451 EPOCH 1950
TYPE Object INV. FREQ. 1 CENTRE YES

NAME C (SAO 031746) RA 1930 DEC +5509 EPOCH 1950
TYPE Check INV. FREQ. 1 CENTRE YES

NAME Dark current RA DEC EPOCH
TYPE Dark current INV. FREQ. 10 CENTRE NO

Figure 2. Observing sequence coding sheet. These sheets allow observing sequences to be described briefly and unambiguously.

space for up to six records on each coding sheet; each record consists of seven fields, NAME...CENTRE. The first record on the first coding sheet is a dummy, containing only the name of the object (usually, for us, an eclipsing binary star). Subsequent records each represent a cycle position within the observing cycle, and long observing cycles may be entered by continuing over several coding sheets.

The field TYPE tells the computer the type of the target: one of 'object', 'sky', 'comparison', 'check', or 'dark current'. The fields NAME and TYPE, are together key fields identifying distinct targets, e.g. there might be two or more different check stars in the observing cycle, distinguished by having different names. On the other hand, the comparison star might be repeated in an observing cycle, by having two records with the same NAME and TYPE (= 'comparison') fields.

The field INV-FREQ holds a positive integer telling the computer how often to observe the target at that cycle position. An INV-FREQ of 1 means that the target is to be observed every cycle and an INV-FREQ of 10 means that the target is to be observed once every ten cycles.

The field CENTRE is now redundant, as we now centre on all localised targets.

The fields RA, DEC and EPOCH may be omitted for most targets, allowing targets to be chosen at the telescope. The only target for which positional information must be given is the object. (The computer cannot work without being given some positional information.)

The data from the coding sheet are saved to a file called 'D. <Object Name>', using a commercial package (AcornSoft's Database), which offers simple facilities for editing, viewing and amending data. Another program 'TCHECK', written by us, is then run, which takes this file and checks it thoroughly for errors, producing a report in which 29 different error or warning messages may appear. Only when no errors are found does TCHECK produce a file of checked data called 'C.<Object Name>'. This file will also contain an ephemeris for the object (assumed to be an eclipsing binary star), and because only these checked files can be used at the telescope, there is great confidence that observing sequences are correct. Observing sequences are created indoors, and may be used thereafter for all observing runs of the object concerned.

The Main Program

At the start of the observing session, the trolley containing the computing and other electronic equipment is wheeled out to the telescope. In disk drive 0 is the disk containing the Main Program; in disk drive 1 is the data disk, containing the checked observing sequence file C.<Object Name>. The observing shed is rolled back, and the trolley placed inside it. The stepper motor cables are attached to the sockets on the trolley. The plug supplying power to the trolley is connected. The various boxes on the trolley are switched on. The telescope is still in its stowed position, with the tube horizontal, and with the stepper motors disengaged. The Main Program is now run.

Step 1: preliminaries. The machine-code routines are loaded. The observer is asked to enter the slewing and tracking rates, and also the filter which is to be used. From this point on, tracking is activated.

Step 2: login. The observer's name is entered, followed by the date and Universal Time. A clock of UT is displayed in the upper right-hand corner of the computer screen. The observer confirms that these details are correct before continuing to step 3.

Step 3: load observing sequence. The observer enters the name of the object to be observed. The file C. <Object Name> is loaded from drive 1, and the computer reads the information about the observing sequence and the targets (stars or sky targets).

Step 4: preliminary pointing of telescope. The computer displays the declination and the LHA of the object – the latter continuously updated – and the observer is asked to point the telescope in the approximate direction of the object, to engage the stepper motor drive units and then to press the space bar. The observer now makes the final connection by attaching the fluid light guide to the photometer head.

Step 5: observer target set-up. The observer is now asked to centre each target in turn, on the graticule in the photometer head, using the button-box. During this step, the computer slews the telescope to the nominal target position, provided it has the information allowing it to do so. The object is always the first target to be centred by the observer. The observer has to press the YES button twice to confirm that each target is centered. (Pressing NO cancels the first YES and resumes centering the same target. The YES and the NO buttons have distinctive sounds associated with them)

Step 6: warnings. The computer gives three warnings: 'Is the mirror in the photometer head in the correct position to take readings?'; 'Is the light which illuminates the graticule switched off?'; 'Is the filter wheel set to the correct position?'. The computer waits for the space bar to be pressed before continuing.

Step 7: initial sky reading. The computer uses the information supplied by the observer in step 5 to calculate which sky is nearest to each star. (There might be more than one sky position in the observing cycle.) The computer moves the telescope to the position of the sky nearest the object and takes a reading. This reading is then assigned as a preliminary estimate of the sky brightness for all sky targets.

Step 8: computer star set-up. The computer now seeks to improve the observer-given positions as follows. For each star in turn:

(i) The telescope is slewed to the nominal star position.
(ii) An initial threshold is taken to be 1.3 times the sky reading given by step 7.
(iii) A half-second peek is taken to determine whether the star is in the pinhole.

1990JBAA..100...30E

(iv) If the star is not in the pinhole, then a spiral search is made for it.
(v) The star is centered in the pinhole.
(vi) A three second reading is taken, which is used to initialise the moving average brightness for the star, and to give an improved value for the threshold, mentioned in (ii) above.
(vii) The star is again centered in the pinhole, using the improved threshold.

Should either a spiral search or centering fail during this step, then the computer immediately returns to the beginning of step 5 above: this is a rare event if the observer takes reasonable care when first setting up in step 5.

Step 9: tour. The computer displays the readings obtained during step 8, and then takes the observer on a 'tour', slewing to each target position in turn, and waiting at each for three seconds. This is a final chance for the observer to check that the set-up has been performed correctly: the observer has the option of rejecting the set-up and returning to step 5, by pressing NO. We have never had to do this, and have always pressed YES to begin automatic observing.

Step 10: automatic sequence preliminaries. The warnings of step 6 are repeated. A file called 'P.<Julian Day>' is created on the data disk, which contains information concerning the set-up. A file called 'R.<Julian Day>' is opened, which will contain the data obtained during automatic observing.

Step 11: automatic observing sequence. The automatic observing sequence now begins. The observer closes the doors of the roll-off shed around the trolley, and retires indoors. The time the observer needs to spend outdoors is, after some experience, about 25 minutes.

There is a variable seq–no which the computer increments every time it attempts to take a reading. Using seq–no the computer can calculate which cycle it is on, and the position within that cycle. From the cycle position may be obtained information about the target. If the target is a dark current, then no reading is presently attempted (because we do not have a motor for the filter wheel). If, after dividing the cycle number by INV-FREQ, the remainder is not zero, then no reading is attempted (so that readings are only attempted once every INV-FREQ cycles).

Readings are attempted for all other targets. The telescope slews to the nominal target position, performs a spiral search if necessary, centres if necessary, and then takes a reading. As with our previous system,[3] each forty-second reading is divided into twenty sub-readings; and the percent coefficient of variation of these gives a good indication of reading quality.

After each reading, the data stored to disk are seq–no, total photon count, percent coefficient of variation, filter used (the same for all readings until we get a filter-wheel motor), and the UT of mid-reading.

Should a spiral search fail, it is assumed that cloud has intervened, and the telescope tracks for ten minutes before resuming the observing sequence with the next target.

There is no shutdown procedure within the Main Program; instead power is cut off by the shed's time-switch. To prevent the files from being left in a bad state by this, each time information is written to disk, a system call is made which forces the disk itself to be updated from the disk buffer.[7]

Theory of errors

The computer holds the positions of targets as integer (x,y) coordinates: the variables ra–step (i) and dec–step (i) being the coordinates of the ith. target, measured in stepper motor steps from some origin. (This origin is fixed at step 4 of the Main Program.) The computer also holds the telescope's direction, in variables t–ra, and t–dec.

When the telescope slews to the nominal position of a target (say the ith.), then, by definition, t–ra must be equal to ra–step (i), and t–dec must be equal to dec–step (i). In practice, the telescope will not be pointing exactly at the target. There is thus a discrepancy between the state of the telescope and the variables representing this state. The computer can measure the size of this discrepancy, in both x and y, by centering (after a spiral search if necessary).

The problem arises, how should the computer correct the discrepancy? Should it adjust t–ra and t–dec, or adjust ra–step (i) and dec–step (i), or a combination of both? These ways of making the adjustment are not equivalent, as only in the first case do the internal representations of the distances between the targets remain unaltered. If all targets are known catalogue stars, then the distances between them are known, and so only t–ra and t–dec need adjusting.

We, however, have made the requirement that stars may be chosen at the telescope, and, during our set-up, the reason for centering a star in the pinhole (step 8) is to improve the observer given position. The assumption has to be made that any discrepancy is due to the observer not pointing the telescope accurately at the requested target: during our set-up, therefore, the discrepancy is corrected by adjusting ra–step (i) and dec–step (i).

Once set-up is complete, there will be errors in the values of ra–step (i), and dec–step (i). How big are these errors? During our first outings, by setting up on the same group of stars on successive nights, we have found that the distances between stars, as determined by the computer at the end of each set-up, differed by as much as fifteen arcseconds. It should be recalled that set-up is necessarily an involved process, with the telescope slewing many times. These errors were nonetheless disturbing, and caused problems during early observing runs before we had a spiral search.

One way round this difficulty is to ignore it and rely upon a spiral search, but there is a better solution, which we now describe. The errors in ra–star (i) and dec–star (i) result from errors during set-up. We shall call such errors 'star position errors', although they are, more properly, errors in the computer's representation of star positions. They cannot be much bigger than about 15 arcseconds, otherwise the set-up would have failed. Errors arising during the automatic observing sequence are mostly due to slippage, and will usually be small, but might, very occasionally, be large.

During automatic observing, the size of the discrepancy is measured each time a star is centered. The computer is made to correct this discrepancy by assuming that at most three steps-worth is due to star position error, and that the remainder is due to slippage. The computer's star positions will then rapidly converge, during the first few observing cycles, to within about 5 arcseconds of the correct positions, and this amount of error will thereafter remain. Positional errors of this magnitude are of no consequence.

An example may make the method clear. In centering a star, a discrepancy of 17 steps in RA was found, together with a discrepancy of -2 steps in declination. The computer adjusts its variables by assuming 14 steps of slippage and 3 steps of star position error in RA. The computer also adjusts its variables by assuming -2 steps of star position error in declination. The state of the system afterwards is that (1) the star is centered in the pinhole, (2) ra–step (*i*) and t–ra have been incremented by 3 from the values they had before centering began, (3) dec–step (*i*) and t–dec have similarly been decremented by 2.

This method of centering is harder to program, but has great advantages. First, the APT can automatically follow moving targets, such as asteroids, provided that they are not moving at speeds in excess of a few arcminutes per hour. This has been successfully tested by observations of (7) Iris, the longest run being of 7 hours, during which the asteroid moved 4.5 arc minutes across the sky. Second, the computer can easily correct for any misalignment of the telescope. This is not a problem for us, but is for those with portable APTs – of which a few exist.[8] Third, this method copes with differential refraction, which becomes important when targets are unusually widely separated, and move between high and low altitudes during the course of the observing run.

Summary

A telescope has been built, which will, after a 25-minute set-up, work unattended for many hours performing differential photoelectric photometry. The telescope makes observations at twice the rate of a fast manual system and may be left to work until dawn. The stringent mechanical requirements of the telescope, and the way these have been met, have been discussed in detail. The software has been written in such a way that observing sequences may be chosen at will; intervals of bad weather cause only temporary disruption and moving targets can be followed.

Addresses: P. E. Ells, 2 Sunnyside, Cowley, Oxford OX4 2NW.
J. W. Ells, Crayford Manor House Astronomical Society.

References

1 Ells, J. W., and Ells, P. E., *J. Br. Astron. Assoc.*, **99** (6), 282 (1989).
2 Ells, J. W., and Ells, P. E., *J. Br. Astron. Assoc.*, **100** (1), 000 (1990).
3 Ells, P. E., *J. Br. Astron. Assoc.*, **96** (4), 204 (1986).
4 Bray, A. C., Dickens, A. C. and Holmes, M. A., *The Advanced User Guide*, Cambridge Microcomputer Centre, 1983, Ch. 13.
5 Hall, S., Genet, R. M. and Thurston, B. L. (editors), *Automatic Photoelectric Telescopes*, Fairborn Press, Ohio, 1986.
6 Pickard, R. D., *J. Br. Astron. Assoc.*, **97** (1), 14 (1986).
7 Bray, A. C., Dickens, A. C. and Holmes, M. A., *The Advanced User Guide*, Cambridge Microcomputer Centre, 1983, p. 337.
8 Trueblood, M. and Genet, R. M., *Microcomputer Control of Telescopes*, Willmann-Bell, Richmond, 1985.

Received 1989 January 25; accepted in revised form 1989 March 29