RTLinux Driven Hungarian Automated Telescope for All Sky Monitoring

Gáspár Á. Bakos

Konkoly Observatory, Budapest, H-1525, P.O. Box 67

Abstract. Massive variability searches initiated the design of HAT (Hungarian Automated Telescope), an autonomous observatory[1]. HAT consists of a horseshoe mount similar to the one used in the Automated All Sky Survey (ASAS, Pojmański 1997), a clamshell dome with various utilities, a telephoto lens and a CCD. Expensive hardware elements have been substituted by software running under Realtime Linux: a multitask operating system which can handle processes in real time. A virtual observer – modeled as a finite state machine – is responsible for managing the observatory. The modular structure allows of running virtually any kind of observing program for small telescopes. We describe technical aspects, as well as present test results carried out by HAT1 (Budapest).

1. Introduction

An effective way of survey mode observations is to use a completely autonomous observatory placed on a perfect site, thus minimizing manpower, usually a bottleneck in recent projects. This observing philosophy reduces the size of teams by substituting observers with software controlled operation and data reduction. Quality software also plays a central role because it can supersede unnecessary hardware elements and makes communication between telescopes well-controlled.

Perhaps 90% of bright variable stars are unknown, or their light curves have not been followed (Paczyński, 2000). The ASAS project has found thousands of new variables in the southern hemisphere with a small telephoto lens an amateur CCD camera attached to a horseshoe mount (Pojmański 2000).

HAT1 is intended to carry out the northern counterpart of Pojmański's study, but it is also suitable for any project which needs automatization of a small telescope. The system was tested in the fall of 2000 at Konkoly Observatory, Budapest, and has recently been relocated to Steward Observatory, Kitt Peak, Arizona.

---

[1]Members of the HAT construction team are József Lázár (software), István Papp (electric engineering), Pál Sári (structural design) and the author. Continuously updated information on this project can be found at the following website: http://www.konkoly.hu/staff/bakos/HAT.

2.   Hardware Overview

We use a horseshoe mount which is friction driven by five phase stepper motors (both for the right ascension and declination). The stepper motors are controlled by a simple electronic converter card, which receives signals from the PC's parallel port. We would like to emphasize that the card is simply a converter, and that the motordrive/clockdrive is the PC's central processing unit (CPU). Proximity sensors on both axes detect home and end positions in order to ensure fail-safe operation, so that the telescope can never loose orientation, even though we use an open-loop control system (no costly encoder employed). The RA and Dec resolutions are $1''$/step and $5''$/step, respectively. The system is backlash-free due to the use of sprocket gears and friction drive.

The telescope and electronics are enclosed in a weather-proof box, which has a special, double-axis opening structure minimizing resistance against wind in any position. Attached devices include lens-heating, rain-detector, domeflat light and power on/off, all controlled by the second parallel port of the PC, including dome opening/closing and status information.

The test CCD was an amateur-class Meade Pictor 416xt camera with 512×768, $9\mu m$ pixels. This camera yielded noisy images with bad interference pattern and showed erratic functioning, but still eligible for test-mode operation. The Kitt Peak setup will try to use an Apogee AP10 $2K \times 2K$, $14\mu m$ camera, which also shows some operational anomalies. A Bessel I filter is selected due to its high transmission at the CCDs sensitivity maximum and because the sky background in I band is low even at full moon.

We chose a Nikon 180mm f/2.8 ED lens, which yields a sharp and relatively non-varying point spread function (PSF) all over the 9 degree field of view with the AP10 camera, but has rather strong vignetting (60% intensity on the edge compared to the center). Although under normal conditions the PC controls the dome, it is possible to switch to manual mode for servicing.

3.   Software System

All the devices are controlled by one single PC running RTLinux, a multitask operating system (OS) which allows using realtime applications. Generally speaking, running tasks (eg. receiving an email) on a multitask OS trigger interrupts and might disturb the scheduling of other important tasks (such as emitting clock signals to drive a mount with sidereal rate). The RTLinux kernel (core of the OS) treats these realtime processes as independent threads with signal handlers with response times better than $15\mu s$. The Linux environment, the user's interface, runs as the lowest priority thread, but on relatively fast PCs and assuming only a few real-time processes, there is no noticeable difference to the user. The software environment consists of four main parts: **drivers** (low level programs), **mountserver**, **central database (DB)** and **virtual observer**.

While the mount is operated as a realtime application (via kernel driver), other devices are run by low-level C programs, behaving similarly to drivers. Communication with the devices is done via the mountserver, which receives complex commands from a TCP/IP port (ie. anywhere on the "internet"), parses them, and distributes them to the appropriate resort (dome, mount, ccd, etc.).
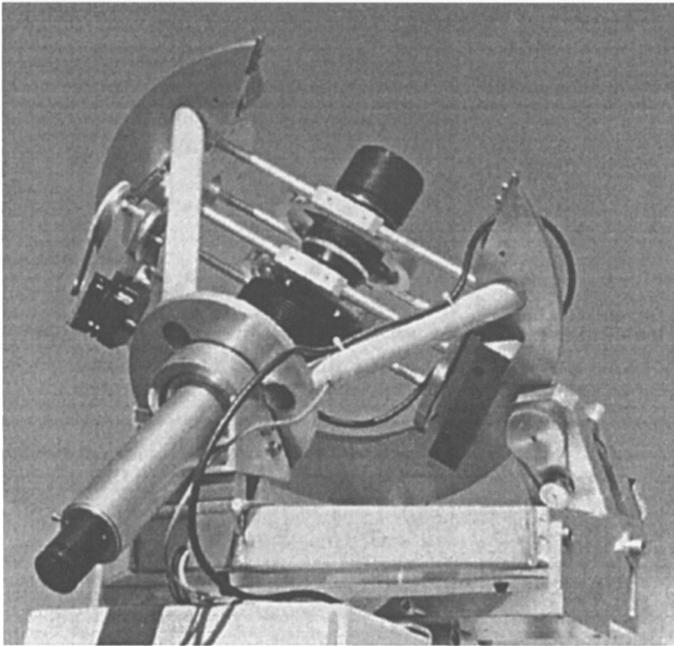
Figure 1. The HAT horseshoe at Budapest during the tests with attached devices: Nikon lens, Meade Pictor CCD and polar telescope.

The mountserver also has the ability to receive packages from the GRB Coordinates Network (GCN).

We use a fast and reliable relational SQL database to store associated parameters of the stations, such as information on the mount, telescope, CCD, scheduled tasks, archived images, and various other properties, all of them arranged into tables. One central DB can maintain several sites set up in any preferred topology, which greatly simplifies management. Calibration information of any station can be updated straight away by modifying the database, which involves automatic re-computation of all the derived parameters.

The virtual observer is responsible for managing the observatory by communicating with the mountserver, the central database and the running (observer) programs. The observer programs (such as taking bias, dark, flatfield frames, all-sky monitoring) are run as independent threads, so the observer is never stuck waiting for a time-consuming operation to finish. Tasks have starting times (relative to sunset) and priorities – both taken from the DB, which properties are used by a task-manager (part of the observer) to launch or stop them and to suspend or interrupt other tasks. Similar to LaTeX text processing language, there is a template, which makes writing new tasks very simple for the user without requiring a profound understanding of the underlying structure. In a simplified picture, the observer is modeled as a finite state machine with transitions between the 5 states: run, wthsleep, daysleep, closeexit and service. The observer resides in "run" state if there is any task to be executed and if the weather conditions are adequate. Otherwise, if bad weather hinders observa-
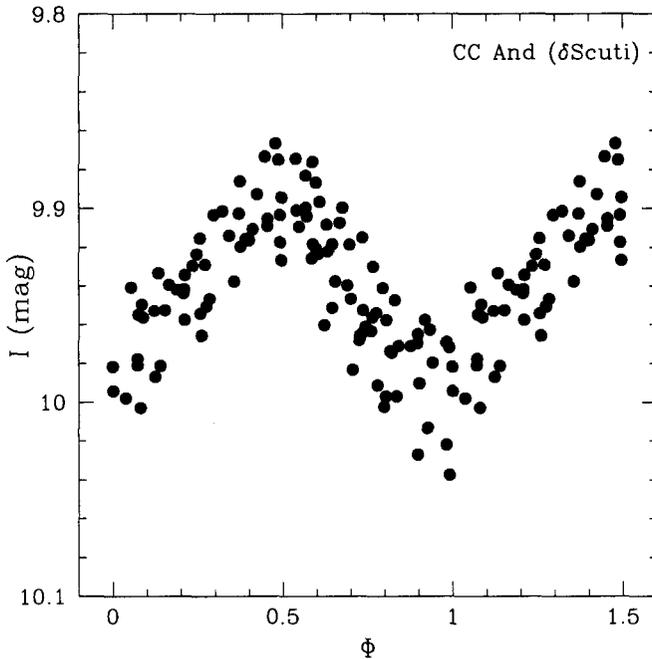
Figure 2.   Light curve of CC And. Note that the small amplitude, and the width of the curve are partly due to the multiperiodic behavior of the variable. The measurements were carried out at Budapest with a 6cm lens and a Meade Pictor Camera.

tions, the program enters "wthsleep" state, basically waiting for clearing, with devices turned off. If no tasks are specified within a given time interval, the observer completely shuts down the observatory (eg. daytime). The "closeexit" and "service" states are self-explanatory. Other stations can monitor the status of the observatory, and with supervisor/service entitlement there is also the possibility of interacting with the observer.

The observatory automatically starts operation after booting the PC, and it is properly shut down in case of rebooting (eg. due to a power outage). The status of the programs is regularly checked (Linux cron daemon), so as to detect and to try to recover from anomalies.

4.   Test Results from Budapest

During three months of test period 14 "clear" nights were used to monitor selected variables under the light-polluted sky of Budapest with the Meade CCD camera attached to HAT. One example is featured in this paper: CC And, a δ Scu variable with $< I > \approx 10^m.0$, $A \approx 0.2^m$, $P \approx 0.12^d$ (Fitch, 1960). Simple aperture photometry was used without removing any bad data point to yield the light curve shown in Fig. 2. The phase was derived from the the main frequency $f_0 = 8.00620$ cyc/day of the star after fitting a double sine.

As the figure indicates, the width of the light-curve (residual $\approx 0.02^m$) is not purely due photometric scattering. Fourier analysis (cf. Fig. 3) revealed the following frequencies (values in braces are from Fitch, 1960): $f_0 = 8.00620$ (8.00591), $f_1 = 7.81920$ (7.81480) and $f_2 = 6.59466[c/d]$ (new, with a lower significance). The final residual after whitening with the three Fourier components was $\sigma = 0.^m015$.

Previous observations were performed using 0.5-1m class telescopes with photometers. This is to demonstrate that a 0.06m instrument is capable of valuable results as well.

References

Fitch, W. S. 1960, ApJ, 132, 701

Paczyński, B. 2000, PASP, 112, 1281

Pojmański, G. 1997, Acta Astronomica, 47, 467

Pojmański, G. 2000, Acta Astronomica, 50, 177
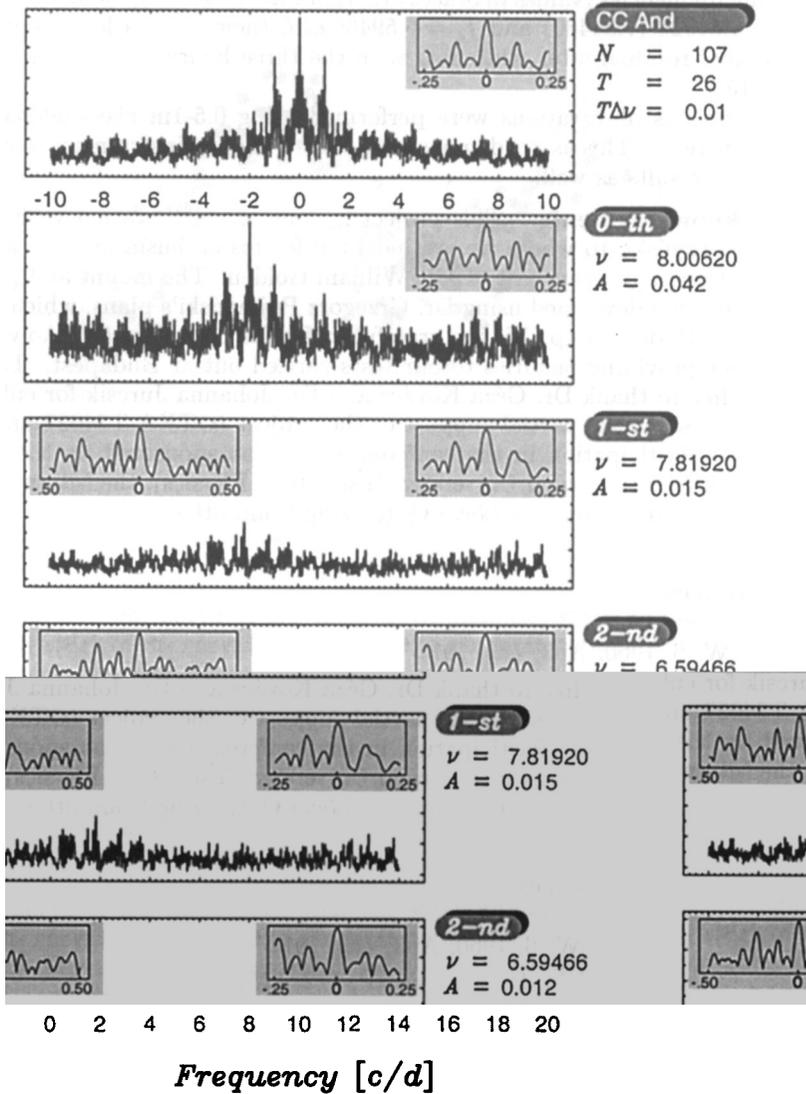
Figure 3. Amplitude spectrum of CC And. The upper panel shows the spectral window, the second panel the raw spectrum, consecutive panels the spectra prewhitened with all the previous main frequencies. The 40000 spectrum points are binned, and only the maxima of each bin are shown (this type of compression yields no loss of information at the resolution of the figure). Blow-ups of the maxima are shown as inserts in the corresponding panels.