1974, *Astron. Astrophys. Suppl.* **15**, 367-369.

A SENSITIVE METHOD FOR DETECTING DISPERSED RADIO EMISSION

J.H. TAYLOR
Department of Physics and Astronomy,
University of Massachusetts, USA

This paper describes an on-line computer program recently developed to improve the sensitivity of searches for pulsars and possibly other types of rapidly-varying radio sources. The program runs in a small on-line computer which is interfaced to a 32 channel filter-type receiver. During normal operation, the computer (a) samples the detected output of each receiver channel once every τ sec; (b) introduces progressive delays into the paths of the higher frequency channels; and (c) combines the delayed signals to produce 32 optimally "de-dispersed" output signals at τ–sec intervals. Additional sets of 32 output signals can also be produced, covering twice or four times the dispersion range, at time intervals of 2τ or 4τ. The system has been tested briefly at the 92 m telescope of the National Radio Astronomy Observatory (NRAO). The receiver had a total bandwidth of 40 MHz, centered at 410 MHz; the system provided a time resolution of $\tau=40$ msec for signals with dispersions in the range 0 to 260 cm^{-3} pc, and a resolution of $2\tau=80$ msec for dispersions from 0 to 520 cm^{-3} pc. Two new pulsars were detected in approximately 48 h of tests (Taylor *et al.* 1972).

The computer program uses a highly efficient algorithm which avoids redundant additions in much the same way as the "fast folding algorithm" (Staelin 1969). A block diagram of a 4 channel version is shown in figure 1. In this example, the detected receiver outputs at four adjacent frequencies are digitized and loaded into the array represented by the squares to the left of figure 1, at intervals τ. The remaining operations indicated in the figure are performed from right to left, and must also be carried out once per sample interval. Rectangles represent summations, and numbers within the rectangles indicate receiver channel numbers in order of increasing frequency. Subscripts on the channel numbers give the accumulated delay, in units of τ, of a particular datum. Circles in figure 1 represent holding locations within the program; they have the effect of introducing one unit of delay.

With these conventions in mind, one can see that the bottom rectangle on the output side (for example) represents the sum $V_1(t)+V_2(t-\tau)+V_3(t-2\tau)+V_4(t-3\tau)$, where $V_i(t)$ is the signal sampled at time t in receiver channel i. In this example the software network provides dispersion filters "tuned" to four different dispersions.

This algorithm may be implemented to process data from any N channel receiver, where N is an integral power of 2, and it will yield N de-dispersed output signals. The time delay introduced across N receiver channels varies progressively from 0 to $(N-1)\tau$. Consequently, the range of dispersion measures covered extends (in N equal intervals) from zero to

$$DM_{\max}=0.000241(N-1)(\tau)(1/f_1^2-1/f_2^2)^{-1}, \tag{1}$$

where $DM_{\max}$ is the maximum dispersion measure in cm^{-3} pc, τ is the sample interval in seconds, and f_1 and f_2 are the frequencies of the lowest and highest receiver channels in MHz.

It is evident from figure 1 that a $2N$ channel algorithm can be made by combining two N channel versions and adding a final set of delays and summations. Therefore, the number of additions required increases as $Nn=N\log_2 N$, rather than N^2 as would be the case if redundant additions were not avoided. The number of computer memory locations required for data storage is $(3N+5Nn+N^2)/4$. The number of locations required for program instructions depends on the computer hardware and on the extent to which program loops are used, but typically amounts to not more than about 4nN locations. Thus, for $N=32$, we require 480 data locations and $\lesssim 640$ instructions. If a single instruction is executed by the machine in time t, the full algorithm can be executed in a time approximately $4nNt$. Thus, for $t=3$ μsec, which is typical of many small computers, the 32 channel algorithm can be executed in about 2 msec. This time clearly is a lower limit on the sampling interval τ.

It is desirable to keep the sample interval τ rather small; ideally, τ should not be greater than the widths of pulses being sought, or the pulses will be diluted in strength. This restriction places a limit on the dispersion range which can be covered with a given receiver bandwidth, as shown in equation (1). However, a larger dispersion range can be covered simultaneously (with reduced time resolution) by maintaining two or more copies of the de-dispersing algorithm in the computer memory and executing them at different rates, such as τ, 2τ, and 4τ.

Once the desired dispersion selectivity has been achieved, what is the best method of detecting the presence of a fluctuating signal in one of the dispersion channels? For a sporadic signal with occasional large peaks, such as the Crab Nebula pulsar, a simple peak detector is probably best. For example, one might record as "events" any output sample exceeding the mean level by 3σ or more. On the other hand, for periodic sources of relatively constant strength, or for very "deep" surveys, some kind of period analysis is best. This analysis would have to be done independently for each dispersion channel, and preliminary work suggests that it would be difficult, but not necessarily impossible, to do this in real-time with existing small computers. It can certainly be done by recording the de-dispersed data on magnetic tape for later analysis.

A third possibility for the detection process also exists. If the dispersed signal being sought has no large peaks exceeding 3σ, but has frequent 1σ or 2σ excursions, then it may be detected by examining the *variance* of each dispersion channel, averaged over some suitable interval large compared to the fluctuation time scale. When the variance of a particular dispersion channel exceeds the average variance by 3 or more standard deviations, a possible detection would be recorded. This method would work well for any signal which varies rapidly, even though it be neither impulsive nor periodic! It would also work well for weak pulsars similar to the Vela pulsar, which show only slight pulse-to-pulse intensity variations. One can show that the integration time required for a 3σ detection by this method is given by the relation

$$T = 9\tau D^{-2} S^{-4}, \tag{2}$$

where τ is the sample interval, D is the effective duty cycle of the signal (approximately unity for a sine wave, and 0.03 for a typical pulsar), and S is the peak strength of the signal being sought, in units of the rms fluctuations at the output of the software dispersion filters. For $\tau = 10$ msec, $D = 0.03$, and $S = 1$, we find $T = 100$ sec. Thus, this method is an attractive possibility in cases where S is not much less than unity.

I thank T. Cram of the National Radio Astronomy Observatory for coding portions of the computer program used at NRAO. This work was performed under grant GP–32414X from the U.S. National Science Foundation.

REFERENCES

Staelin, D.H.: 1969, *Proc. Inst. Elec. Electron. Engrs.* **57**, 724.
Taylor, J.H., Huguenin, C.R. and Manchester, R.N.: 1972, *IAU Circ.* no. 2435.

J.H. Taylor

Department of Physics and Astronomy
University of Massachusetts
Amherst, Massachusetts 01002, USA

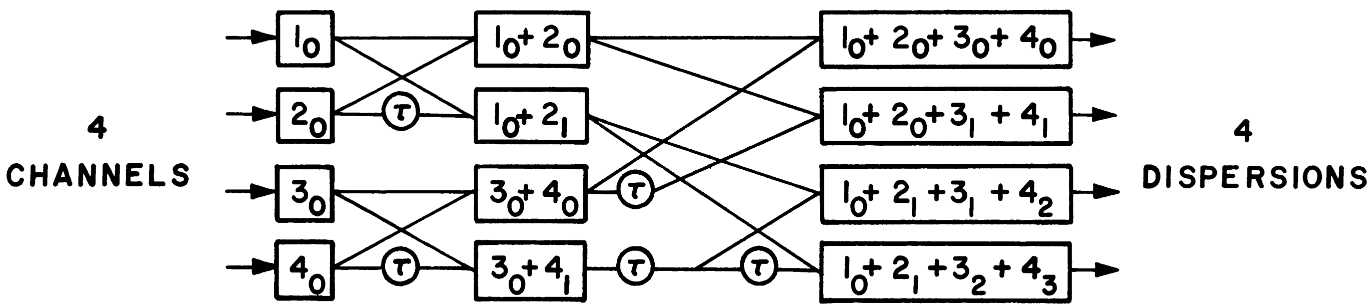

Figure 1 Block diagram of a 4 channel digital dispersion filter. Detected signals from a 4 channel receiver are input at the left; de-dispersed output signals are taken from the right. Rectangles represent summations, and circles represent unit delays. The indicated operations are performed from right to left.