

III. VLBI SYSTEM

III. 6 CROSS-CORRELATION PROCESSING IN A COMPUTER FOR VLBI FRINGE TESTS

By

Tetsuro KONDO, Jun AMAGAI, Hitoshi KIUCHI, and Munetoshi TOKUMARU

(Received on March 18, 1991)

ABSTRACT

Data obtained by a very long baseline interferometer (VLBI) usually correlate with each other by a designated processor composed of many wired logic components for measuring correlation. Consequently, only a center (or station) equipped with a correlator can process the data. Even though data used to check system coherence are observed in advance of full (>24 hours) observations (this kind of test observation is called a "fringe test"), tapes containing sampled data must be gathered to the correlation center for processing. This is a disadvantage in getting quick results, especially for stations located far from the center. To obtain results of a fringe test quickly and to demonstrate feasibility, a computer software performing correlation processing of VLBI data completely by a computer (i.e., without a correlator) is developed. This software is actually used for the fringe test data observed on the baseline between Kashima and Miyazaki (about 1000 km distance) in Japan, where data are transmitted via telecommunications networks, and successfully detected fringes.

1. Introduction

Data acquired by a 1-bit digitizing method using a very long baseline interferometer (VLBI) used to be correlated completely by a computing program on the basis of every sample (bit) in the data stream⁽¹⁾. This computing program was, however, unable to process a huge amount of data, such as a geodetic VLBI lasting for more than 24 hours, in a realistic processing period (e.g., several days for one day experiment data) because of a limited speed of calculation. The total amount of data sometimes reaches up to 2×10^{12} bits/station. Even though it takes 1 μ sec for one-bit correlation, the total time to finish the entire data correlation is estimated to be about 23 days (this is probably quite an underestimation). In accordance with the progress in VLBI systems, a designated correlator has been used for correlation processing. Consequently, only the center (or station) equipped with a correlator can process the data.

On the other hand, in the case of the so-called "fringe test" that is a preliminary observation for checking system coherence in advance of full observations, it is convenient to correlate the data in a computer using data transmitted via telecommunications networks. This is usually faster than the direct transportation of tapes, especially in the case of stations located far from the correlation center.

For the above-described purpose, a software package that calculates cross correlation and obtains the observed delay and delay rate was developed in an HP-1000 45F computer (we named the software "CCC": Cross Correlation in a Computer). CCC has since been transplanted to an HP-1000 A900 computer.

In this paper, we describe the principle behind CCC and demonstrate some results.

2. Principle of CCC

CCC performs correlation processing in the same way as a correlator. First, we briefly describe the correlation processing carried out in the K-3 correlator.

Figure 1 shows a schematic block diagram of the K-3 correlator. Fringe stopping used to cancel out the Doppler shifts due to the earth's rotation is applied to the X (reference) data stream. It is performed by multiplying the cosine and sine functions of the a priori fringe phase to the data stream independently, where a 3-level approximation (-1, 0, 1) of functions is adopted. Also, delay tracking used to calibrate the continuous change of delay due to the earth's rotation is applied to the Y (remote) data stream. To minimize the coherence loss due to step-like delay tracking, the phases of the fringe stopping functions are changed by 90° degrees at the time when a bit shift takes place⁽²⁾. Then both X and Y data streams correlate with each other to form 8-lag complex cross-correlation function.

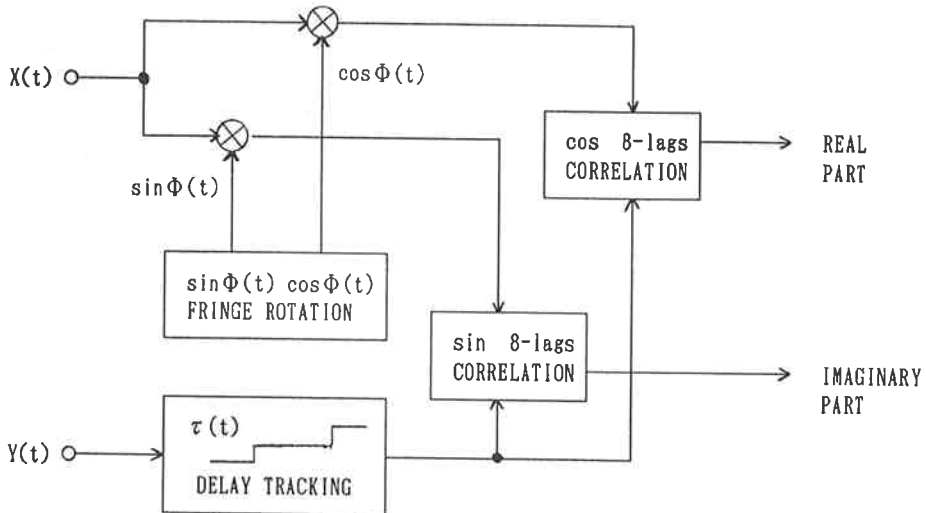


Fig. 1 Schematic block diagram of the K-3 correlator.

Data processing by CCC is essentially the same as that by the K-3 correlator, except for the lag size in correlation and representation of the fringe stopping function. The length of lags is changeable from 8 lags up to 512 lags, and no level approximation is made for cosine and sine functions.

2.1 CCC Data

The data necessary for CCC are prepared by the K-3 decoder⁽³⁾. The K-3 decoder has a 1-Mbit

(1,048,576 bit) buffer memory in order to correspond to a data stream period of 0.262 sec when data are sampled at a frequency of 4 MHz, which is common for a geodetic VLBI. It takes approximately 1 minute for data to transfer from the K-3 decoder to the host computer. Both the K-3 recorder⁽⁴⁾ and K-3 formatter⁽³⁾ can be data inputs for the decoder. However, to use the data available for a fringe search covering a 0.25 Hz wide search range that is typical for a geodetic VLBI, data streams at 4 sec intervals at the longest are required. This is realized only by playing back the data repeatedly, because the K-3 formatter produces a data stream in real time. Thus, only data at 1 minute intervals are available when the K-3 formatter is used. These intervals are too long. Therefore only the data taken from the data recorder are described below.

First, data on tapes are reproduced and stored in the buffer memory in the K-3 decoder. Then the data are transferred into a computer via the IEEE-488 bus. A tape must be reproduced and rewind repeatedly with data read out point forward by at least a 0.262 sec step in order to obtain the data stream necessary for an effective fringe search (Fig. 2). For other station data, the same procedure should be done. Especially, the beginning of the data stream should be the same among stations in terms of the time label. In the CCC software, 1 Mbit data are recognized as a block of 16 records of data, where one record consists of 8192 byte data (see Fig. 2). As one frame of a K-3 data format consists of 2500 byte data, of which the first 20 byte data are header data including time information, at least one header is included in a record. We can, therefore, process the data on a record by record basis because the beginning and end time of one record data can easily be obtained by decoding the header data included in a record.

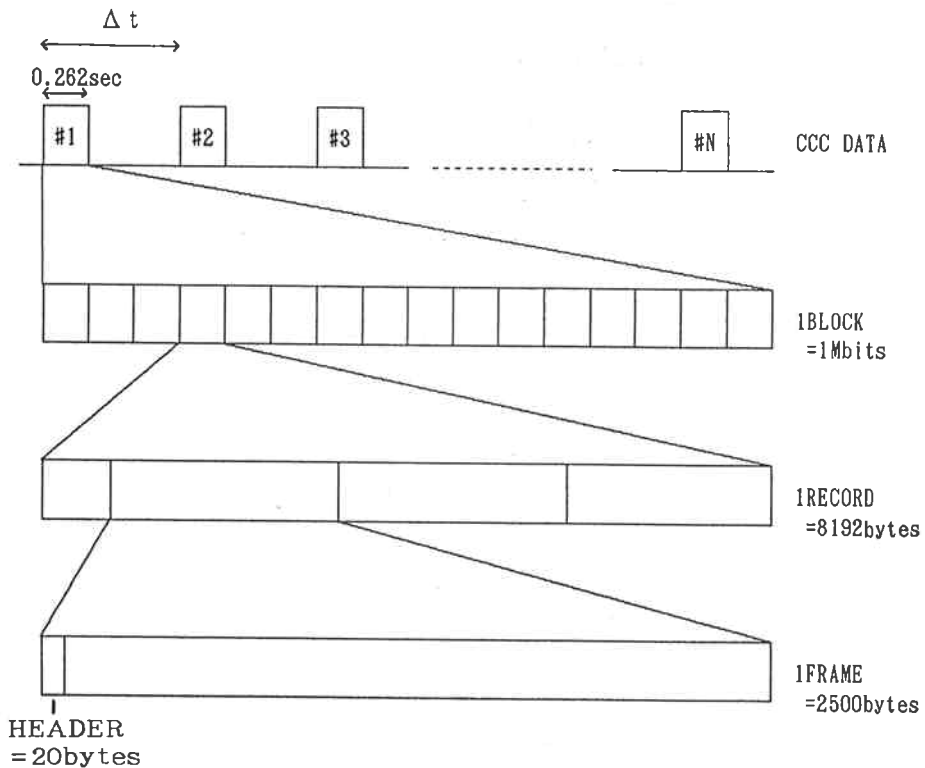


Fig. 2 Data format for CCC.

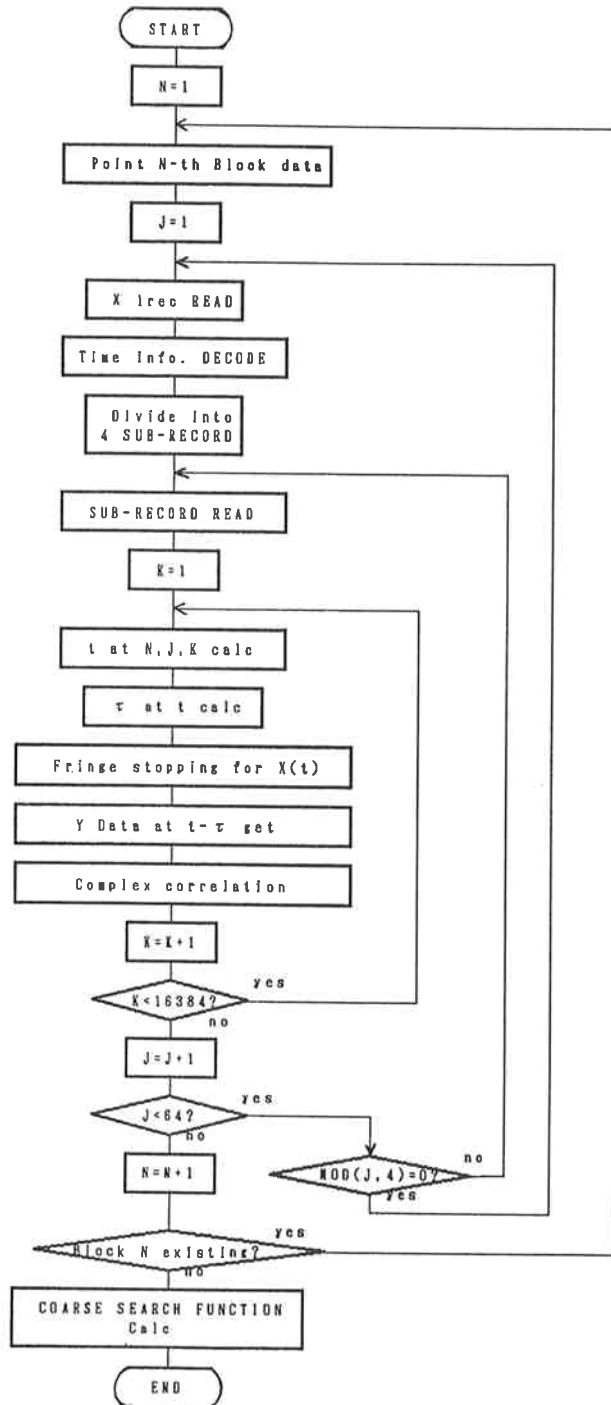


Fig. 3 Flowchart of CCC.

A detailed flow of data processing is shown in Fig. 3. A record of data is further divided into four 2048-byte subrecords, so as to speed up computation using a linear approximation of delay during the sub-record period, as described later. The total number of sub-records in a block is, therefore, 64.

2.2 Cross Correlation

The actual cross correlation is computed for every subrecord and accumulated up to one block. Finally, we obtain N correlated data, where N is the number of blocks. These data are used to calculate the “coarse search function” described later. In the explanation below, n is the block number, j is the subrecord number in a block, and k is the bit position in a subrecord, e.g., $F_{nj k}$ expresses the value F at the k -th bit in the j -th sub-record of the n -th block. Also, the X and Y data streams are expressed as $x_{nj}(k)$ and $y_{nj}(k)$.

The parameters necessary for actual correlation processing in a sub-record, such as a delay and delay rate, are calculated for the time at the first bit, t_{nj1} , of every sub-record as

$$\tau_{nj1} = \tau_0 + \dot{\tau}_0(t_{nj1} - t_0) + \frac{1}{2}\ddot{\tau}_0(t_{nj1} - t_0)^2 + \frac{1}{6}\ddot{\tau}_0(t_{nj1} - t_0)^3 - Clk_{x-y}, \dots \dots \dots (1)$$

$$\dot{\tau}_{nj1} = \dot{\tau}_0 + \ddot{\tau}_0(t_{nj1} - t_0) + \frac{1}{2}\ddot{\tau}_0(t_{nj1} - t_0)^2, \dots \dots \dots (2)$$

where, τ_0 , $\dot{\tau}_0$, $\ddot{\tau}_0$, and $\ddot{\tau}_0$ denote the a priori values of delay, delay rate, delay acceleration, and rate acceleration, respectively, at the epoch t_0 taken at the center of the whole data stream, and Clk_{x-y} denotes the a priori clock offset between the X and Y station clocks (the positive sign of the clock offset here is taken when a clock of Y station is delayed to that of X station). These a priori values are calculated by a different software (it is not described here, because it is beyond the scope of this paper). The length of a sub-record is about 4 msec for a 4 MHz sampling. This is short enough to neglect a delay change caused by delay acceleration in the period; the effect converted into a fringe phase at 8 GHz is estimated to be merely 10 arc seconds at most during 4 msec. Therefore, the expected delay at the k -th bit in the j -th sub-record of the n -th block is approximated well by τ_{nj1} and $\dot{\tau}_{nj1}$ as

$$\tau_{nj k} = \tau_{nj1} + \dot{\tau}_{nj1}(k-1)T_b, \dots \dots \dots (3)$$

where T_b is the sampling period (0.25 μ sec for 4 MHz sampling). Then we can calculate the fringe phases necessary for fringe stopping as

$$\phi_{nj k} = \omega_0 \tau_{nj k} \text{ (rad)}, \dots \dots \dots (4)$$

where ω_0 is the observation angular frequency in the sky. Because phase increment (or decrement) during T_b is at most 2° for a baseline on the earth at 8 GHz, we can decrease the calculation frequency of $\phi_{nj k}$ from every bit to every 8 bits (byte) without introducing any serious error in the correlation. This results in a speedup of processing. Quantization of $\tau_{nj k}$ into the T_b step is required to get the pick-up bit position in the Y data stream. It is given by

$$q_{nj k} = \text{nint}\left(\tau_{nj k} / T_b\right), \dots \dots \dots (5)$$

where a function $\text{nint}(x)$ calculates the nearest integer value of x . Hence, the cross correlations at the l -th lag after fringe stopping are calculated as follows,

$$C_n(l) = \sum_{j=1}^J \left\{ \sum_{k=1}^K y_{nj}(k - q_{nj k} - l) x_{nj}(k) \cos(\phi_{nj k} + \Delta\phi) \right\}, \dots \dots \dots (6)$$

$$S_n(l) = \sum_{j=1}^J \left\{ \sum_{k=1}^K y_{nj}(k - q_{nj k} - l) x_{nj}(k) \sin(\phi_{nj k} + \Delta\phi) \right\}, \dots \dots \dots (7)$$

where $J = 64$, $K = 16384$, and

$$\Delta\phi = \begin{cases} 0 & \text{for } q_{nj k} = q_{nj}(k-1) \\ \frac{\pi}{2} & \text{for } q_{nj k} = q_{nj}(k-1) + 1 \\ -\frac{\pi}{2} & \text{for } q_{nj k} = q_{nj}(k-1) - 1, \end{cases}$$

where $q_{nj(k-1)}$ means the quantized delay at the time 1 bit earlier. By combining $C_n(l)$ and $S_n(l)$, we obtain a complex correlation function, i.e.,

$$R_n(l) = C_n(l) - iS_n(l). \dots \dots \dots (8)$$

2.3 Delay and Delay Rate Search

The Fourier transform of $R_n(l)$ gives the video cross-spectrum function $V_n(s)$, where s is the index of video frequency. We define the coarse search function $D(\Delta\tau, \Delta\dot{\tau})$ as

$$D(\Delta\tau, \Delta\dot{\tau}) = \sum_{n=1}^N \left\{ \sum_{s=1}^S V_n(s) \exp(-i\omega_s \Delta\tau) \right\} \exp(-i\omega_0 \Delta\dot{\tau} \Delta t n), \dots \dots \dots (9)$$

where ω_s is the s -th angular video frequency, and Δt is the time interval between blocks. The residual delay and delay rates are given as values that maximize the absolute value of $D(\Delta\tau, \Delta\dot{\tau})$.

3. Examples of Processing Results

We have developed CCC by checking results using real VLBI data obtained on a Kashima-Gilcreek (Alaska) baseline as a part of regular geodetic experiments conducted in May 1985. Figure

4 shows the results of the CCC test. In the figure, the coarse search function integrated for the period of 4.2 seconds is displayed. The 64-bit lag correlation function was computed in this example. It took about 8 hours to process 16 blocks of data in the HP-1000 45F; now, the processing speed is improved by a factor of 3 by transplanting CCC into the HP-1000 A900. The time required for processing is of course dependent upon search ranges. A clear peak appeared at the a priori position, i.e., just center of both the delay and rate search ranges. We used the a priori values that were obtained by the K-3 correlator. This demonstrates that CCC estimations are an unbiased delay and delay rate with respect to those obtained by the correlator hardware.

After developing CCC, it was used in the actual fringe test of a domestic VLBI measurement conducted on the baseline between Kashima and Miyazaki (about 1000 km west southwest from Kashima), which was operated by the Geographical Survey Institute (GSI) of Japan, on October 6, 1986. Figure 5 shows the system used for CCC. First, the sampled data were recorded on tapes as in usual VLBI observations both in Kashima and Miyazaki. Next, data were reproduced and acquired by the K-3 decoder. The data in the 1-Mbit buffer memory were written on a computer magnetic tape via a control computer at Kashima. In Miyazaki, the data were transmitted to Kashima over telephone lines by a modem with a baud rate of 1200 bit/sec. It took 1 hour and 21 minutes to transfer 4-Mbit data to Kashima. This time is much longer than that expected from the baud rate, probably due to some overhead included in the transmitting protocol (the equivalent baud rate calculated from above values was about 800 bit/sec). The data received at Kashima were also written on a computer magnetic tape. These tapes were read by the host computer HP-1000 45F, then CCC was executed for these data. Figure 6 shows three coarse search functions for different search ranges of delay, which are 3.88 μ sec,

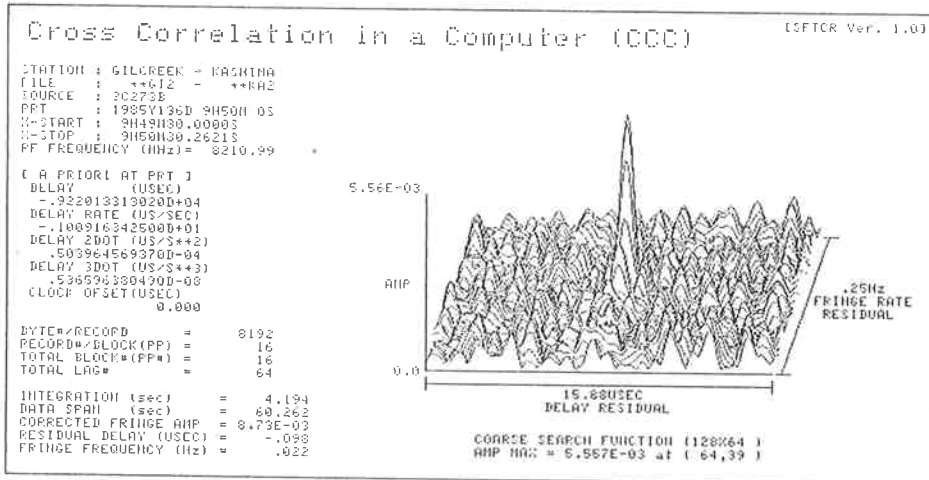


Fig. 4 An example of the CCC output display. VLBI data observed on Kashima-Gilcreek (Alaska) baseline were used in the development of the CCC software package. Coarse search function calculated for 3C273B is displayed in the figure. Abscissa denotes residual delay. Residual fringe rate (in Hz) and correlation (or fringe) amplitude are taken in the depth and height directions. CCC is executed for 16 blocks of data with a search width of 15.88 μ sec in delay and 0.25-Hz fringe rate. An eminent peak appears at the center on delay and rate surface.

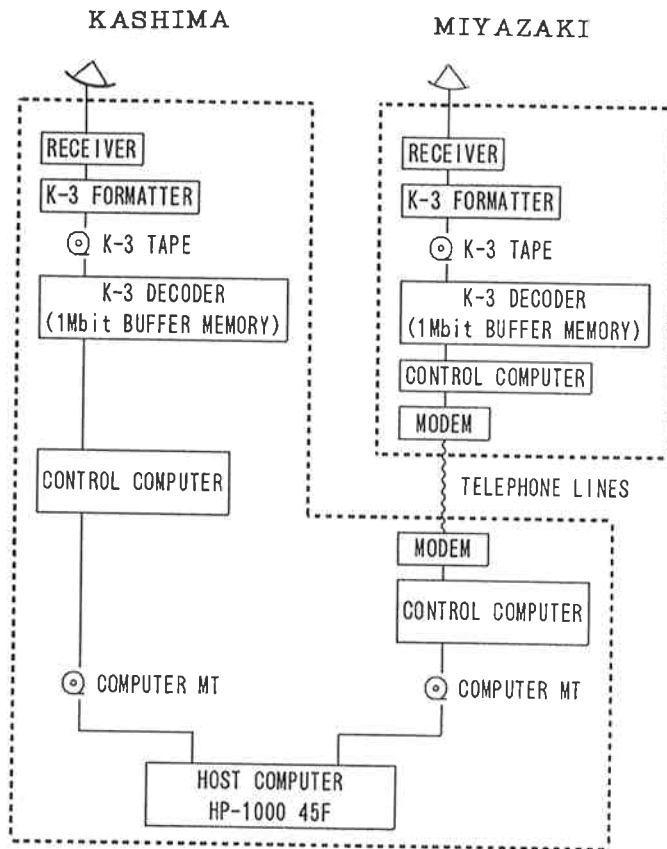


Fig. 5 Total system used for the actual fringe test using CCC. See text for details.

15.88 μsec , and 31.88 μsec for the top, middle, and bottom panels, respectively. The integration period was 3.1 seconds. As clearly shown in the figure, we successfully detected a single peak. The time required for the whole process to detect fringes was approximately 10 hours, and it was of course faster than the fringe check by means of tape transportation. Afterwards, the peak obtained by CCC was confirmed to be the real correlation maximum by comparing it with the results obtained by the K-3 correlator using raw observed data from Miyazaki.

Fig. 6 Fringe check results obtained by CCC using the system shown in Fig. 5. The fringe test was performed on the Kashima-Miyazaki baseline (about 1000 km distance) on October 6, 1986. Three panels correspond to three different search widths of delay, i.e., 3.88 μsec , 15.88 μsec , and 31.88 μsec for the upper, middle, and lower panels, respectively. Clear single peak represents a successful detection of fringes.

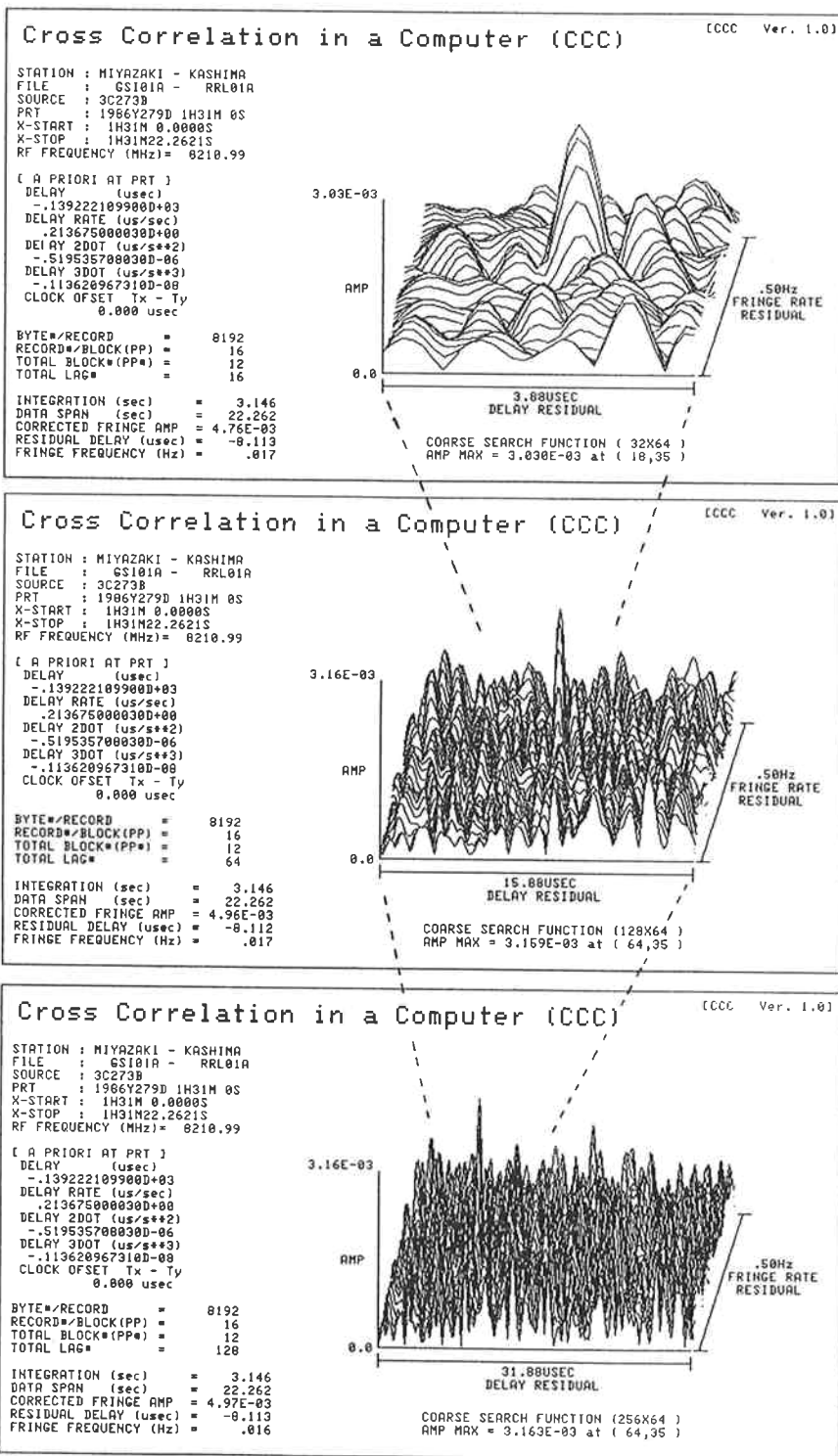


Fig. 6

4. Conclusion

We have developed a software named "CCC" that performs cross correlation of VLBI data completely by computer. CCC was actually used in the fringe test of a domestic VLBI experiment conducted on October 6, 1986 and successfully obtained fringes. The processing speed of CCC has been improving because of transplantation to a computer with a much faster operational speed. Increased speed in operation will be accomplished by improving the computation algorithm. Also, it is now possible to use a much faster modem than it was 5 years ago. Both of these improvements will dramatically decrease the time required for fringe checks to a more realistic time. Moreover, the development of a real-time VLBI technique will continue.

Acknowledgement

The authors would like to express their sincere appreciation to the staff members of the GSI VLBI group for their cooperation.

References

- (1) Moran, J. M., Very Long Baseline Interferometer Systems, Methods of Experimental Physics, Vol. 12 Astrophysics, Part C: Radio Observations (edited by L. Marton), pp. 174-197, Academic Press, 1976.
- (2) Sugimoto, Y. and S. Hama, "III-2 Correlation Processor", Rev. Radio Res. Lab., Vol. 30, No. 1, pp. 163-175, 1984 (in Japanese).
- (3) Kiuchi, H. and H. Kuroiwa, "II-5 Backend", Rev. Radio Res. Lab., Vol. 30, No. 1, pp. 69-85, 1984 (in Japanese).
- (4) Hama, S. and J. Amagai, "II-6 Data recorder", Rev. Radio Res. Lab., Vol. 30, No. 1, pp. 87-98, 1984 (in Japanese).