

### 3.2.3 KSP Bandwidth Synthesizing Software

Tetsuro KONDO, Mamoru SEKIDO, and Hitoshi KIUCHI

Communications Research Laboratory

#### Abstract

The bandwidth synthesizing-software KOMB, initially developed for K3 VLBI data reduction on a minicomputer, was ported to a UNIX workstation to process KSP VLBI correlator output data. The KOMB software obtains directly observed values — such as delay time, delay rate, and correlated amplitude — from cross-correlated data. We compared the results processed by the KSP data-reduction system (correlator + KOMB) with those by the K3 system. The difference in the observed delays used for baseline analysis was usually smaller than the theoretical errors calculated from the signal-to-noise ratio. Thus, there is no fatal programming error either in K3 or KSP KOMB.

**Keywords** : bandwidth synthesis, coarse-delay search, fine-delay search, fringe

#### 1. Introduction

The accuracy of the delay time observed in very long baseline interferometry (VLBI) is inversely proportional to the bandwidth of the receiving frequency. A wider frequency band gives more accurate results. However, it has been difficult to record a wide band signal due to limited recording capacities. This problem has been overcome in conventional VLBI by combining multi-channel signals to increase the measurement accuracy. This technique is called “bandwidth synthesis”<sup>(1)</sup>.

For example, the accuracy of delay determination is about 500 ns in the case of one-channel receiving at a frequency bandwidth of 2 MHz. However, if the signals received over two channels separated by 100 MHz are combined, the determination accuracy improves to about 10 ns (=1/100 MHz). By increasing the number of receiving channels and extending the frequency separation, we can further improve the delay measurement accuracy. The bandwidth synthesizing technique improves accuracy.

In our VLBI system, the software performing the bandwidth synthesis is called “KOMB”, named after the comb-like shape of the correlation function after the multi-channel data are combined. The KOMB

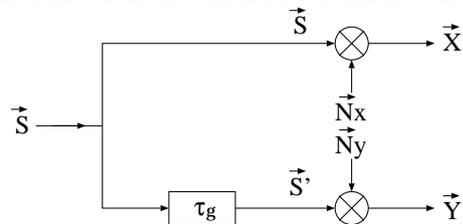
software is located in the final stage of the data-reduction system of the VLBI system. The data output by the correlator are processed by KOMB to obtain the observed delays and delay rates. Further analysis, such as baseline estimation, is carried out using these observed values.

The KOMB software was initially developed on a minicomputer when the K3 VLBI system was developed<sup>(2)</sup>. It was modified for KSP to handle the KSP correlator output<sup>(3)</sup>. It was also modified to run on a UNIX workstation. The principle method used by KOMB to obtain the delay time and delay rate is described in Sec. 2. In Sec. 3, the KOMB processing is explained in more detail. The results of KSP and K3 correlation-processing systems are compared in Sec. 4.

#### 2. Cross-correlation function

Cross-correlation processing is carried out to obtain the delay time from VLBI observation data. However, as delay time changes continually due to the earth’s rotation, the shape of the correlation function also changes due to this delay change.

First we explain a cross-correlation function for direct RF signals without considering the earth’s rotation. Then we explain the cross-correlation function after the frequency is converted from RF signals to video signals. Next we explain the cross-correlation function when the earth’s rotation is considered.



**Figure 1.** *Equivalent signal model for VLBI.*

An equivalent signal model for VLBI is shown in Fig. 1. Vector  $\vec{S}$  denotes the signal from the radio source,  $\tau_g$  is the delay between passes,  $\vec{N}_x$  and  $\vec{N}_y$  are the noises imposed at the receivers, and  $\vec{X}$  and  $\vec{Y}$  are the signals received at the end of the baseline. In this figure, each signal is represented as a vector. We express them as a function of angular frequency  $\omega$  in the following discussion for convenience.

The signals received at stations X and Y are given by

$$X(\omega) = S(\omega) + N_x(\omega) \quad (1)$$

$$Y(\omega) = S'(\omega) + N_y(\omega). \quad (2)$$

The cross spectrum of these signals is given by

$$S_{xy}(\omega) = X(\omega)Y^*(\omega)$$

$$= S(\omega)S'^*(\omega) + S(\omega)N_y^*(\omega) + N_x(\omega)S'^*(\omega) + N_x(\omega)N_y^*(\omega). \quad (3)$$

Assuming  $S$ ,  $N_x$ , and  $N_y$  are independent Gaussian noises, the 2nd and following terms on the right side of Eq. (3) become zero after integration. Because the signal from the radio star reaching station Y is delayed by  $\tau_g$ , it is expressed as

$$S'(\omega) = S(\omega)e^{-i\omega\tau_g}. \quad (4)$$

Substituting this relation into Eq. (3), we get

$$S_{xy}(\omega) = |S(\omega)|^2 e^{i\omega\tau_g}. \quad (5)$$

The cross-correlation function is obtained by performing an inverse Fourier transform of Eq. (5). Thus, a normalized cross-correlation function is given by

$$r_{xy}(\tau) = \frac{\int S_{xy}(\omega)e^{i\omega\tau}d\omega}{\int |X(\omega)|d\omega \cdot \int |Y(\omega)|d\omega}. \quad (6)$$

The normalized cross-correlation function when limited frequency band signals ( $\omega_0$  to  $\omega_0 + \omega_B$ ) are received is obtained by integrating Eq. (6) over this frequency range:

$$r_{xy}(\tau) = \rho_0 \cos\{(\omega_0 + \pi B)(\tau + \tau_g)\} \frac{\sin \pi B(\tau + \tau_g)}{\pi B(\tau + \tau_g)}, \quad (7)$$

where

$$\rho_0 = \sqrt{\frac{T_{ax}T_{ay}}{(T_{ax} + T_{nx})(T_{ay} + T_{ny})}}, B = \frac{\omega_B}{2\pi}$$

and  $T_{ax}$ ,  $T_{nx}$ ,  $T_{ay}$ , and  $T_{ny}$  are the antenna temperature and system-noise temperature of station X, and those for station Y, respectively.

In actual VLBI observations, the received signals are converted into video-band signals, then correlated. Next, we derive a cross-correlation function for this case.

Suppose that frequency conversion is performed by the local oscillator at angle frequencies  $\omega_x$  and  $\omega_y$  at stations X and Y. The spectrums after conversion are

$$X_v(\omega) = X(\omega + \omega_x)e^{-i\phi_x} + X(\omega - \omega_x)e^{i\phi_x} \quad (8)$$

$$Y_v(\omega) = Y(\omega + \omega_y)e^{-i\phi_y} + Y(\omega - \omega_y)e^{i\phi_y}, \quad (9)$$

where  $\phi_x$  and  $\phi_y$  are additional phases at frequency conversion at each station. The first term on the right side of these equations represents the upper side band (USB), and the second term represents the lower side band (LSB). In the following we consider only the USB for simplicity. The same arguments hold when the LSB or both side bands are considered.

Let the cross spectrum when receiving the USB be  $S_{xy}^v(\omega)$ :

$$\begin{aligned} S_{xy}^v(\omega) &= [X(\omega + \omega_x)e^{-i\phi_x}][Y(\omega + \omega_y)e^{-i\phi_y}]^* \\ &= S(\omega + \omega_x)S'^*(\omega + \omega_y)e^{i(\phi_y - \phi_x)} \\ &\approx |S(\omega + \omega_x)|^2 e^{i\omega\tau_g} \cdot e^{i(\phi_y - \phi_x)} \cdot e^{i(\omega_y - \omega_x)t}, \end{aligned} \quad (10)$$

where the following approximation is applied under the condition  $|\omega_y - \omega_x| \ll 1$ .

$$S^*(\omega + \omega_y) \approx S^*(\omega + \omega_x e^{i(\omega_y - \omega_x)t})$$

Furthermore, by defining  $\theta$ ,  $\omega'$ , and  $\omega_0$  as

$$\begin{aligned} \theta &= \phi_y - \phi_x + (\omega_y - \omega_x)t, \\ \omega' &= \omega + \omega_x, \\ \omega_0 &= \omega_x, \end{aligned}$$

we obtain the following cross spectrum:

$$S_{xy}^v(\omega) = |S(\omega')|^2 e^{i\{\theta + (\omega_0 + \omega)\tau_g\}}. \quad (11)$$

After inverse Fourier transformation of Eq. (11) and normalization, we obtain a normalized cross-correlation function:

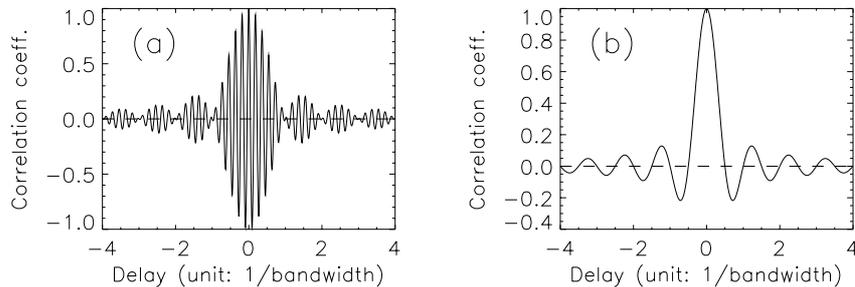
$$r_{xy}^v(\tau) = \frac{\int S_{xy}^v(\omega)e^{i\omega\tau}d\omega}{\int |X_v(\omega)|d\omega \cdot \int |Y_v(\omega)|d\omega}. \quad (12)$$

When receiving white noise at an angle frequency ranging from 0 to  $\omega_B$ , the cross-correlation function for an actual VLBI observation is obtained by integrating over this frequency range:

$$r_{xy}^v(\tau) = \rho_0 \cos\{\theta + \omega_0\tau_g + \pi B(\tau + \tau_g)\} \frac{\sin \pi B(\tau + \tau_g)}{\pi B(\tau + \tau_g)}. \quad (13)$$

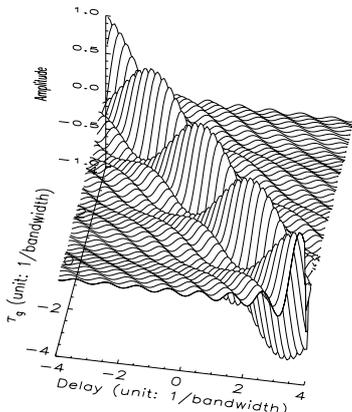
Both Eqs. (7) and (13) are cosine functions of the envelope represented by  $\sin x/x$ . However, their responses to a delay change differ. For Eq. (7), change  $\tau_g$  appears as a vibration with a frequency of  $\omega_0\tau$ , i.e., the changes corresponds to the RF frequency itself, while for Eq. (13) it appears as  $\pi B\tau (= 1/2 \cdot \omega_B\tau)$ , i.e., it corresponds to the frequency of the video-frequency bandwidth. These correlation functions are illustrated in Fig. 2.

We have so far considered  $\tau_g$  to be a constant to simplify the derivation of equations. However, in actual VLBI observations, it changes continuously due to the earth's rotation. The position of the peak of the correlation function thus moves in accordance with the change in the shape of the function. Therefore, a simple integration will result in a decrease in the correlation amplitude. To avoid this problem, the delay change is continuously compensated for to retain the shape of the correlation function. When  $\tau_g$



**Figure 2.** (a) Correlation between RF signals. (The RF frequency was set to five times the bandwidth in order to make the figure legible.) (b) Correlation between video signals. (Both abscissas were normalized by the reciprocal of the receiving bandwidth.)

changes to  $\tau_g + \Delta\tau_g$  in Eq. (13), we can retain the shape of the correlation function by changing  $\tau$  to  $\tau - \Delta\tau_g$  and simultaneously adding  $-\omega_0\Delta\tau_g$  to the phase term of a cosine function. The first compensation corresponds to the shift of the time-axis, and is called “delay tracking”. The second one compensates for the fringe phase, and is called “fringe stopping”.



**Figure 3.** Effect of delay time ( $\tau_g$ ) change in correlation function. The RF is set very low to clearly show the change of the correlation function. Note that the delay time change changes both the shape of the correlation function and its peak position.

Figure 3 shows how the correlation function expressed by Eq. (13) actually changes when  $\tau_g$  is changed. Note that the change in  $\tau_g$  not only shifts the correlation function, but also changes its shape.

A correlator integrates the correlation function by changing  $\tau_g$  according to a value calculated using the station positions, source position, and earth-orientation parameters. However, a small difference still remains between this predicted value and the actual change in delay due to unmodeled effects, such as a change in the station clock. As a result, the inte-

gration period can not be extended to as long as it is in conventional correlation processing. We therefore used 1 to 4 seconds for the unit integration period in our KSP correlator. Correlated data integrated per unit integration period are output from correlator. The bandwidth-synthesizing software calculates the residual delay and residual-delay rate from these data. The correlator also detects the phase of the phase-calibration (PCAL) signals imposed on each channel; this phase is necessary for the bandwidth synthesis.

### 3. Precise determination of delay time

*A priori* values to a certain degree of accuracy are required for the correlation processing. The degree of accuracy is determined from the error in the fringe-phase rotation in a unit integration period. It should be less than about 100 degrees to avoid significant correlation loss. The error in the delay time itself is not severe because a certain delay range is covered by the correlator’s lag window.

In the correlation processing, the *a priori* values of delay time ( $\tau$ ), delay rate ( $\dot{\tau}$ ), and delay acceleration ( $\ddot{\tau}$ ) are used to calculate the expected delay and delay rate at any moment. The KOMB synthesizing software searches the residuals of the delay and delay rate ( $\Delta\tau$  and  $\Delta\dot{\tau}$ ). We will next explain the procedure used in KOMB and the processing flow based on the output from the correlator we developed. This software can easily be adapted to a general time-domain correlator by simply adjusting for the differences in the sampling rates and number of lags.

#### 3.1 Normalization of correlator output

Correlator output consists of three kinds of counters: the counter used for cross correlation (cross-correlation counter), the counter that counts the number of bits used in the correlation (total bit-

number counter), and the counter used for the phase-calibration-signal phase detection (PCAL counter). These counters are output every integration unit time (PP: parameter period). The number of lags in the cross-correlation counter is 8 bits for the K-3 correlator and 32 bits for the KSP correlator. Both the cosine and sine components of the correlated values for these lags are output as correlated count values. The total bit-number counter is used to normalize the correlation amplitude. The PCAL counter is used to synthesize the bandwidth.

Let the cosine and sine components of the cross-correlation counter and those of the total-bit counter be  $a_r(n, k, d)$  and  $a_i(n, k, d)$  and  $A_r(n, k)$  and  $A_i(n, k)$ , respectively. The complex cross-correlation function after normalization can then be expressed as

$$\begin{aligned} \text{Re}\{R(n, k, d)\} &= \frac{2a_r(n, k, d) - A_r(n, k)}{A_r(n, k)} \\ \text{Im}\{R(n, k, d)\} &= -\frac{2a_i(n, k, d) - A_i(n, k)}{A_i(n, k)}, \end{aligned}$$

where  $n$ ,  $k$ , and  $d$  are the indices for the frequency channel, the PP number, and the lag (in bit units), respectively.

The PCAL counter output consists of the cosine and sine components of the station X and station Y data. Let these be  $P_{xr}(n, k)$ ,  $P_{xi}(n, k)$ ,  $P_{yr}(n, k)$ , and  $P_{yi}(n, k)$ , respectively. Hence, the normalized PCAL count value is given for station X by

$$\begin{aligned} Q_{xr}(n, k) &= \frac{2P_{xr}(n, k) - N_{pp}}{N_{pp}} \\ Q_{xi}(n, k) &= -\frac{2P_{xi}(n, k) - N_{pp}}{N_{pp}}, \end{aligned}$$

where  $N_{pp}$  is the total number of bits included in a PP. From these components the phase of the PCAL signals can be calculated using

$$\phi_{nx}(k) = \tan^{-1}\left\{\frac{Q_{xi}(n, k)}{Q_{xr}(n, k)}\right\}. \quad (14)$$

The normalized PCAL count value and phase are similarly obtained for station Y by using

$$\begin{aligned} Q_{yr}(n, k) &= \frac{2P_{yr}(n, k) - N_{pp}}{N_{pp}} \\ Q_{yi}(n, k) &= -\frac{2P_{yi}(n, k) - N_{pp}}{N_{pp}} \\ \phi_{ny}(k) &= \tan^{-1}\left\{\frac{Q_{yi}(n, k)}{Q_{yr}(n, k)}\right\}. \end{aligned}$$

To obtain the true correlation amplitude, three corrections are adopted; a one-bit sampling (A/D conversion) correction, a bias correction due to 3-level approximation of the fringe rotation, and a fractional-bit correction.

The loss due to the one-bit sampling is given by <sup>(4)</sup>

$$\rho_0 = \sin\left\{\frac{\pi}{2}\rho_c\right\}, \quad (15)$$

where  $\rho_0$  is the true correlation coefficient and  $\rho_c$  is that after one-bit sampling.

The function used for fringe stopping in the correlator is a sinusoidal function approximated by three levels ( $-1$ ,  $0$ , and  $+1$ ). The fundamental sinusoidal component of this function has an amplitude of  $4/\pi \cdot \cos(\pi/8)$  ( $\approx 1.176$ ). It thus affects the correlation amplitude, resulting in a biased amplitude. When a value of zero is realized in the fringe-stopping function by compression of the correlation, a quarter of the total integration period, it should be compensated for afterwards. These corrections are for the case of using a 3-level approximation of the fringe-stopping function.

The fractional-bit correction compensates for the loss of correlation caused by the step-wise delay tracking. Delay tracking in a correlator is performed in bit units even though the actual delay change is continuous. In other words, the KSP correlator cannot perform continuous delay tracking. The difference between a continuous delay change and a step-wise delay change reduces the correlation amplitude. To reduce this loss, fringe stopping is done at the RF corresponding to the center frequency in the video frequency band, and the fringe phase is jumped by  $90^\circ$  when a one-bit shift occurs in the delay tracking. Although a loss still remains, it is considerably smaller ( $\pi^2/288 \approx 3.4\%$ ) than when fringe stopping is done at the base of each video frequency band and there is no  $90^\circ$  jumping (the loss remaining is  $\pi^2/72 \approx 13.7\%$ ).

These compensations can be summarized as follows,

$$r(n, k, d) = \sin\{C \cdot R(n, k, d)\}, \quad (16)$$

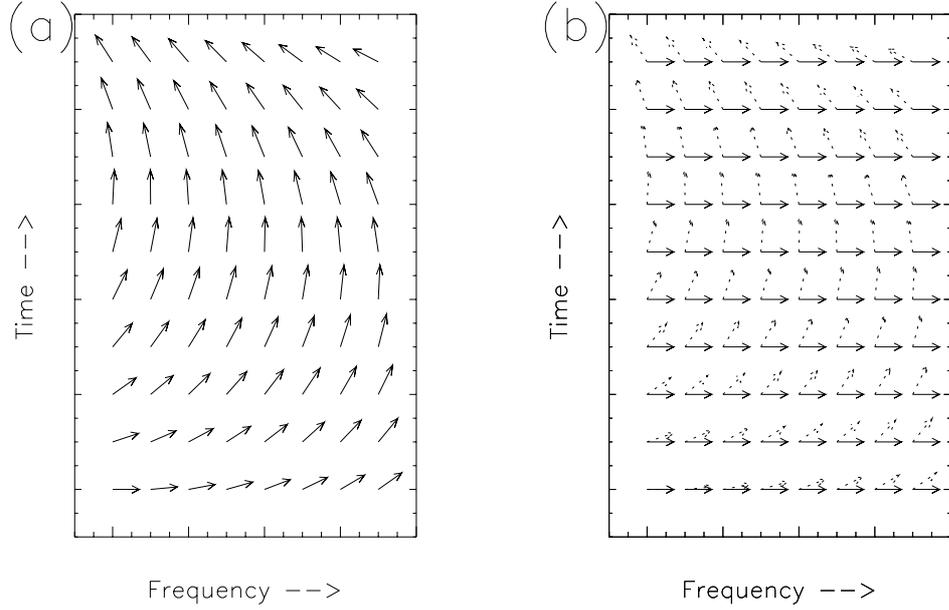
where  $r(n, k, d)$  is the correlation coefficient after compensation,  $R(n, k, d)$  is the raw correlation coefficient, and  $C$  is given by

$$C = \frac{\pi}{2} \cdot \frac{3}{4} \cdot \frac{\pi}{4 \cos \frac{\pi}{8}} \cdot \frac{1}{1 - \frac{\pi^2}{288}}. \quad (17)$$

When the correlation amplitude is very small ( $\ll 1$ ), as is the case in conventional VLBI observation,  $r(n, k, d)$  can be approximated as

$$r(n, k, d) = C \cdot R(n, k, d). \quad (18)$$

A Fourier transform of this correlation function is a video cross spectrum. In the Fourier transform we add an extra  $J$  points of zero data to a correlation function consisting of  $J$  lag points, then we perform a  $2 * J$  point Fourier transform, thereby obtaining a



**Figure 4.** Principle of finding delay residual  $\Delta\tau$  and delay-rate residual  $\Delta\dot{\tau}$ . (a) A vector display of the video cross spectrum on a frequency-time plane.  $\Delta\tau$  causes phase rotation in the frequency direction.  $\Delta\dot{\tau}$  causes phase rotation in the time direction. (b) By applying a reverse-phase rotation to each vector to cancel out the phase errors, all vectors can be aligned in the same direction.

cross video spectrum value at  $J+1$  points dividing the video frequency band into  $J$  frequency sub-bands. Let the cross video spectrum be  $S_v(j, k, n)$ , where  $j$  ( $j = 0 \sim J$ ) is the video frequency index. A coarse search of the delay and delay rate is performed using the video cross spectrum.

### 3.2 Coarse-delay search

If the cross spectrum, which is the Fourier transform of a-cross correlation function, is used, it is easy to understand the principle of searching for the residuals of delay time  $\Delta\tau$  and delay rate  $\Delta\dot{\tau}$ . A vectorial representation of the cross video spectrum on a frequency-time plane is shown in Fig. 4(a). Generally, the residuals are not zero, so a vector (phase) rotates in the time and frequency directions. Because the gradient of the phase in the frequency direction is the delay time,  $\Delta\tau (\neq 0)$  causes the phase to rotate in the frequency direction. Similarly,  $\Delta\dot{\tau} (\neq 0)$  causes the phase to rotate in the time direction. If such residuals can be compensated for perfectly, a vector can be wholly turned in the same direction, as shown in Fig. 4(b). Thus, the intensity integrated over the frequency and time domains becomes maximum.

In a coarse-delay search, the compensation for  $\Delta\tau$  is limited only in the video frequency band. In this case, the operation shown in Fig. 4(b) can be ex-

pressed for channel  $n$  as

$$F(n, \Delta\tau, \Delta\dot{\tau}) = \frac{1}{K} \sum_{k=1}^K \left\{ \frac{1}{J-1} \sum_{j=1}^{J-1} S_v(j, k, n) e^{-i\omega_j^n \Delta\tau} \right. \\ \left. \cdot e^{-i\omega_0^n \Delta\dot{\tau} \Delta tk} \right\}, \quad (19)$$

where  $\omega_j^n$  is the angle frequency for index  $j$  in the video frequency band,  $\omega_0^n$  is the RF angle frequency corresponding to the base band frequency of channel  $n$ ,  $\Delta\tau$  and  $\Delta\dot{\tau}$  are the trial delay time and delay rate, and  $\Delta t$  is the time interval of index  $k$ , i.e., the PP period. Equation (19) thus performs integration by compensating for the phase rotation in the video frequency band by  $\Delta\tau$  and for the fringe-phase rotation in the time direction by  $\Delta\dot{\tau}$ . The range of summation inside the  $\{ \}$  is  $j = 1 \sim J-1$  instead of  $j = 0 \sim J$  so as to remove the DC offset and to avoid the effect of the band edges of the video frequency band;  $j = 0$  corresponds to the DC component and  $j = 0, J$  the band edges. We define the coarse-delay search function by integrating the absolute value of  $F(n, \Delta\tau, \Delta\dot{\tau})$  over the frequency channels:

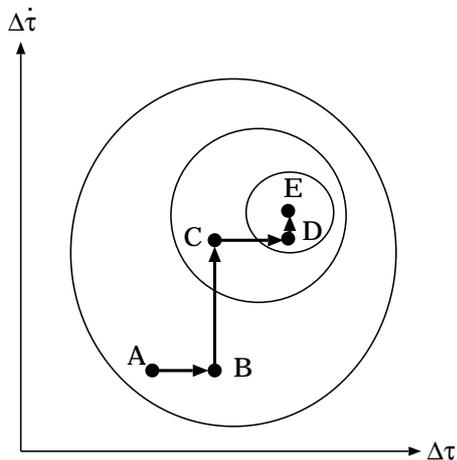
$$F(\Delta\tau, \Delta\dot{\tau}) = \frac{1}{N} \sum_{n=1}^N |F(n, \Delta\tau, \Delta\dot{\tau})|. \quad (20)$$

We can say that a coarse-delay search is the process for finding the  $\Delta\tau$  and  $\Delta\dot{\tau}$  that maximize  $F(\Delta\tau, \Delta\dot{\tau})$ .

Equation (19) is considered to be a two-dimensional Fourier transform with respect to parameters  $\Delta\tau$  and  $\omega_0^n \Delta\dot{\tau}$ . Therefore, we can apply the fast Fourier transformation (FFT) to compute the function. Note that it is necessary to set  $S_v(j, k, n)$  to 0 both at  $j = 0$  and  $J$  and to set the lower side band values to 0 before performing the FFT.

There are several mathematical methods for finding the maximum value of Eq. (20). One is the steepest descent method; another is the Newton method. KOMB uses a simple mountain-climbing method.

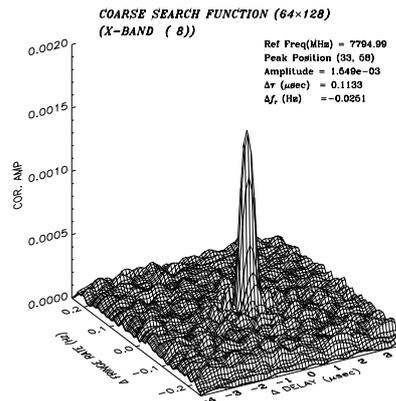
We can compute  $F(n, \Delta\tau, \Delta\dot{\tau})$  for the discrete values of  $\Delta\tau$  and  $\omega_0^n \Delta\dot{\tau}$  by using two-dimensional FFT. By converting  $\omega_0^n \Delta\dot{\tau}$  into a  $\Delta\dot{\tau}$  domain and integrating over the channels, we obtain  $F(\Delta\tau, \Delta\dot{\tau})$  for the discrete values of  $\Delta\tau$  and  $\Delta\dot{\tau}$ . Thus, by searching for the maximum value of  $F(\Delta\tau, \Delta\dot{\tau})$ , we can obtain  $\Delta\tau$  and  $\Delta\dot{\tau}$ . The  $\Delta\tau$  and  $\Delta\dot{\tau}$  obtained this way become the starting values for finding a solution that gives a larger value of  $F(\Delta\tau, \Delta\dot{\tau})$ . This process is analogous to mountain climbing, as shown in Fig. 5. We start from point “A” in Fig. 5; this point is a discrete set of points given by the FFT. We calculate  $F(\Delta\tau, \Delta\dot{\tau})$  using Eq. (19), changing  $\Delta\tau$  in small steps. If we find a maximum (point “B” in the figure), we fix  $\Delta\tau$  there. Next we compute  $F(\Delta\tau, \Delta\dot{\tau})$ , changing  $\Delta\dot{\tau}$  in small steps to reach point “C”. This search process is then repeated, but for the  $\Delta\tau$  direction. The search alternates between directions  $\Delta\tau$  and  $\Delta\dot{\tau}$  until arriving at the mountain summit (point “E”).



**Figure 5.** Search for function maximum by “mountain climbing” method. Point “A” is the start point;  $\Delta\tau$  and  $\Delta\dot{\tau}$  are alternatively searched for; we finally arrive at the maximum (the summit, point “E”).

To compute the maximum value and position we use a parabolic approximation given by the values of three different points. To calculate a function value for parameters not given by the FFT, a formula must be calculated directly. Therefore, the fewer the itera-

tions in the mountain climbing, the shorter the calculation time. When the software was first developed, this was an important factor given the limited power of the computer. Now that a much more powerful computer is being used, the number of iterations has become less important. In the actual processing, the required accuracy is being achieved after three iterations in each direction. Figure 6 shows an example of a coarse-delay search function  $F(\Delta\tau, \Delta\dot{\tau})$ .



**Figure 6.** Example of coarse-search function for the X band (8 ch) for 3C345 observed on the Kashima-Koganei baseline at 06:12 UT on December 3, 1995. The video bandwidth was 2 MHz. Later, the video bandwidth was set to 8 MHz for routine KSP observations. The width of the peak structure in the direction of delay time was thus narrowed (sharpened) to about 1/4 that shown in the figure.

### 3.3 Fine-delay search

A fine-delay search is the process of finding the delay residual by extending the phase correction to different channels. The phase difference between channels, which is caused by a different local oscillator and by a different electrical path, should thus be corrected. The PCAL signal phase is therefore used to compensate for the difference between channels.

Let the delay and delay residuals obtained by a coarse-delay search be  $\Delta\tau_s$ , and  $\Delta\dot{\tau}_s$ . Before proceeding with a fine-delay search, we define

$$D_s(n, k) = \frac{1}{J-1} \sum_{j=1}^{J-1} S_v(j, k, n) e^{-i(\omega_j^v \Delta\tau_s + \omega_0^n \Delta\dot{\tau}_s \Delta tk)}. \quad (21)$$

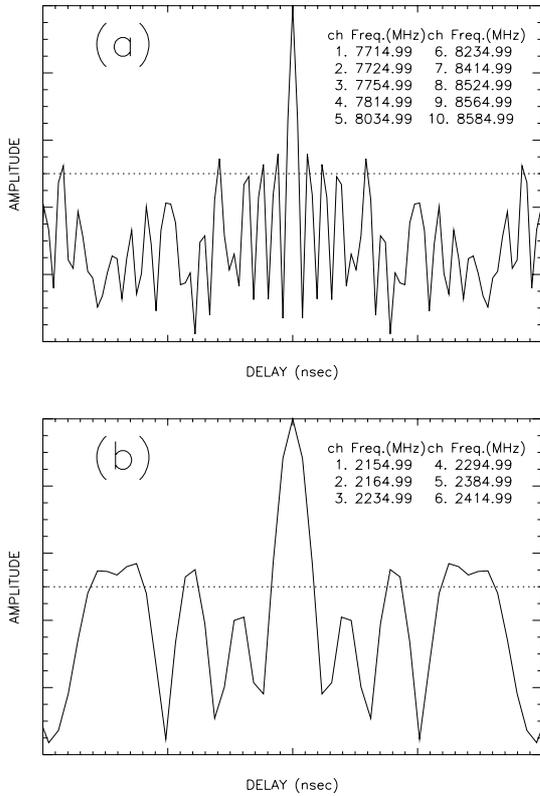
This is the correlation function after the coarse-search residuals are corrected for. Using  $D_s(n, k)$ , we define the fine-delay search function:

$$D(\Delta\tau, \Delta\dot{\tau}) = \frac{1}{K} \sum_{k=1}^K \left[ \frac{1}{N} \sum_{n=1}^N D_s(n, k) e^{-i(\omega_0^n \Delta\tau + \Delta\phi_n)} \right] \cdot e^{-i\omega_0^n \Delta\dot{\tau} \Delta tk}, \quad (22)$$

where

$$\Delta\phi_n = \phi_{nx} - \phi_{ny} \quad (23)$$

and  $\phi_{nx}$  and  $\phi_{ny}$  are the average values of the PCAL phase of channel  $n$  for stations X and Y, respectively. Inside the [ ] of Eq. (22) is the delay residual compensation over different channels. Hence, the determination accuracy (in general, the order of the reciprocal of the bandwidth) of the delay time can be improved more than that of the video bandwidth. This is called bandwidth synthesis. An example of an actual bandwidth-synthesizing function is shown in Fig. 7.



**Figure 7.** Example of bandwidth-synthesizing function for (a) X band (10 ch) and (b) S band (6 ch).

An ideal bandwidth-synthesizing function is symmetrical on both sides of the peak. In actual processing, the symmetry can be used to judge whether the bandwidth synthesis has been done properly. Allocation of frequency channels is selected so as to minimize the side-lobe level of the bandwidth-synthesizing function. We can use minimum redundancy array theory to search for such a frequency allocation. When the number of channels is few, it is also possible to search for the combination by directly computing the bandwidth-synthesis function for all possible combinations of frequency channels under appropriate limitations such as a limited number of channels and a

fixed minimum space between channels. This method works effectively for searching for the combination of frequency channels that avoids the interference signals that exists at certain frequencies.

Because Eq. (22) forms a two-dimensional Fourier transform, like that in the coarse-search function, we can use FFT to compute the function for the discrete values of  $\Delta\tau$  and  $\Delta\dot{\tau}$ . To apply FFT to bandwidth synthesis, Eq. (22) should be modified a little. If the lowest frequency among channels is  $\omega_0^1$ , we can rewrite the inside of [ ] in Eq. (22) to

$$\begin{aligned} & \frac{1}{N} \sum_{n=1}^N D_s(n, k) e^{-i(\omega_0^n \Delta\tau + \Delta\phi_n)} \\ & = e^{-i\omega_0^1 \Delta\tau} \left[ \frac{1}{N} \sum_{n=1}^N D_s(n, k) e^{-i\phi_n} e^{-i(\omega_0^n - \omega_0^1) \Delta\tau} \right]. \end{aligned} \quad (24)$$

The inside of [ ] in Eq. (24) can be computed by FFT. The phase term  $i\omega_0^1 \Delta\tau$  is compensated for later. The search process for finding the maximum fine-delay search function is similar to that for the coarse-delay search. First,  $D(\Delta\tau, \Delta\dot{\tau})$  is computed using FFT for the discrete set of  $\Delta\tau$  and  $\Delta\dot{\tau}$ . Then, the initial values of  $\Delta\tau$  and  $\Delta\dot{\tau}$  where  $D(\Delta\tau, \Delta\dot{\tau})$  takes a maximum are found. Next comes a mountain-climbing process. Let the delay and delay residuals obtained by this fine-delay search be  $\Delta\tau_m$  and  $\Delta\dot{\tau}_m$ ; the observed values of the final delay and delay-rate residuals are given by

$$\Delta\tau_{obs} = \Delta\tau_m + l \cdot t_{am} \quad (25)$$

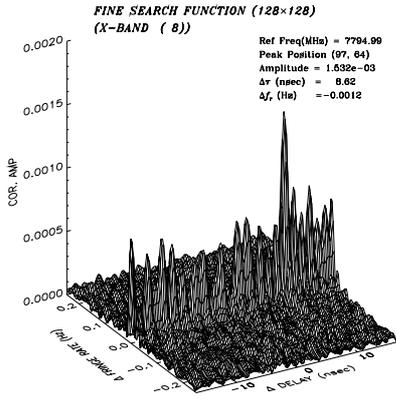
$$(l = 0, \pm 1, \pm 2, \dots)$$

$$\Delta\dot{\tau}_{obs} = \Delta\dot{\tau}_m + \dot{\psi}_x - \dot{\psi}_y, \quad (26)$$

where  $t_{am}$  is the reciprocal of the greatest common divisor of the span of the interval frequency of the received frequency channels; this is the repetition cycle of the pattern in the  $\Delta\tau$  direction caused by the discrete coverage on the frequency domain (called an ambiguity). This ambiguity is always included in observed delays. However, it can be removed in the baseline analysis. The  $\dot{\psi}_x$  and  $\dot{\psi}_y$  values are the PCAL rate, which is the changing rate of the PCAL phase imposed on the received signals to compensate for the phase difference between channels. The observed correlation amplitude is given by

$$\rho_{obs} = |D(\Delta\tau_m, \Delta\dot{\tau}_m)|. \quad (27)$$

The observed fringe phase is the phase of function  $D(\Delta\tau_m, \Delta\dot{\tau}_m)$ . Figure 8 shows an example of the fine-delay search function.



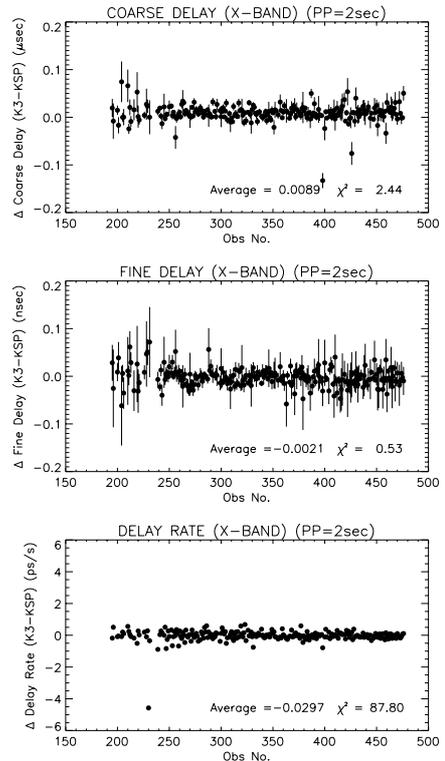
**Figure 8.** Example of fine-delay search function for the X band (8 ch) for 3C345 observed on the Kashima-Koganei baseline at 06:12 UT on December 3, 1995. If the characteristics of the system are ideal, the function should be symmetrical with respect to the direction of delay time on both sides of the peak. Here, a little asymmetry can be seen, but it has a negligible effect on further processes.

#### 4. Comparison of K-3 and KSP correlation-processing results

The bandwidth synthesizing software for KSP (KSP-KOMB) was ported from the KOMB developed for the K3-VLBI system (K3-KOMB) so as to process the KSP correlator output data. Although the K3-KOMB is a Fortran program that operates on a mini-computer (HP 1000/45F or A900), a special function that can be used only on these computers was used to reduce the processing time. In porting the program to the KSP computer (HP-9000/700 series UNIX workstation), we used standard Fortran syntax as much as possible. The increase in the number of lags (from 8 to 32) of the correlation data output from the correlator and the increase in the number of channels (from 14 to 16) were considered when revising the program. The first increase affected the method used to store the correlation data and the subroutine used to calculate the video spectrum. The second increase resulted in a change in the KOMB output-file format. The software for the interface to the baseline analysis was rewritten to handle the new output format.

The KSP-KOMB was tested and the results were compared to those of the K3 processing. The same observation tapes were processed by the K3 correlator and the KSP correlator, independently. The correlator output data were then processed by each KOMB, and the results were compared. The data used were for 254 scans observed on the Kashima-Miura baseline (about 123 km in length) on July 25, 1995. The width of the video frequency band was 2 MHz. The

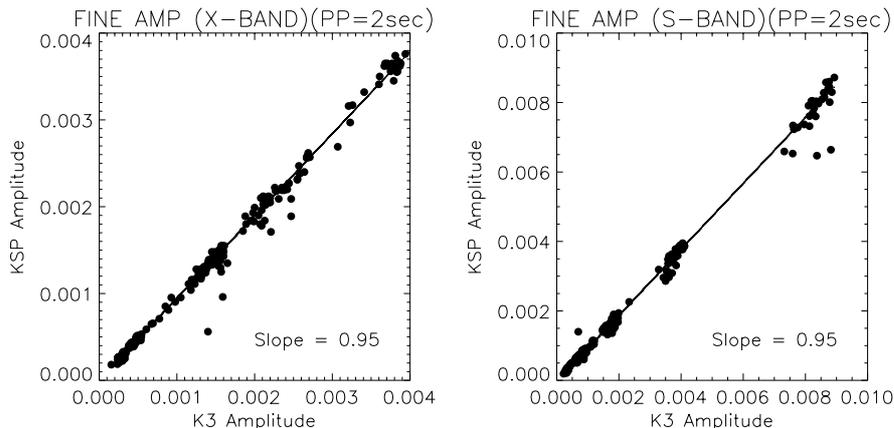
comparison results are shown in Figs. 9 and 10.



**Figure 9.** Comparison of K3 and KSP correlation-processing (including correlator and KOMB processing) results. Each error bar shows a one- $\sigma$  error. The  $\chi^2$  in each figure is the value normalized by the degree of freedom (=the number of data points -1). The data used were observed on the Kashima-Miura baseline on July 25, 1995. The frequency band was the X band and the video frequency bandwidth was 2 MHz.

Figure 9 compares the coarse-delay search, fine-delay search, and delay rate for the X band. The  $\chi^2$  normalized by the degree of freedom (=the number of data points -1) was 2.43, 0.53, and 87.8, respectively. When calculating  $\chi^2$ , the observation errors obtained by KSP-KOMB were multiplied by  $\sqrt{2}$  because K3-KOMB and KSP-KOMB give almost the same observation errors. Because the observation error output by KOMB is the theoretical value of the thermal noise error calculated from the signal-to-noise ratio (i.e., the correlation amplitude and integration time), the error produced by the correlator is not included. The difference between the K3 and KSP results are thus considered to reflect the error produced by the correlation processing because the same data were independently processed.

The  $\chi^2$  denotes the coincidence of the actual data scatter and the theoretical observation error. The  $\chi^2$  for the coarse-search delay of 2.43 means that an error a little larger than the theoretical error occurred in



**Figure 10.** Comparison of correlation amplitudes of K3 and KSP for fine-search correlation amplitude of X band (left panel) and S band (right panel). The data used were the same as for Fig. 9.

the correlation processing. The  $\chi^2$  for the fine search delay of 0.53 means that the error generated during the processing was smaller than the theoretical error. For the baseline analysis, an additional error,  $\sigma_{add}$ , is introduced to take any unexplainable errors into account<sup>(5)</sup>. The error produced during correlation processing may explain part of  $\sigma_{add}$ . The  $\chi^2$  for the delay rate of 87.8 means that the error generated during processing was larger than the theoretical one. The standard deviation in the difference between the delay rates for K3 and KSP processings was 0.24 ps/s, which is about ten times larger than the theoretical  $1\sigma$  error (average is 0.026 ps/s). It has been pointed out that the formal error for the delay rate might be underestimated. This is demonstrated by our comparison results. Fortunately, no bias error can be seen, and the weight of the delay-rate data is low in the baseline analysis. Therefore, it does not affect the analysis very much. The cause of the large error in the delay rate, however, should be resolved.

Figure 10 compares the correlation amplitudes of the K3 and KSP processings. The amplitude with KSP processing was smaller by about 5% for both the S- and X-band data. If this was caused by imperfect *a priori* values used during correlation processing, the drop in the correlation amplitude should differ between the two bands. However, the amplitude fall was the same. This suggests that the cause is not due to an error in the *a priori* values. The design of the KSP correlator differs from that of the K3 one in the number of lags (8 for K3, 32 for KSP). This might explain the difference in the correlation amplitude.

## 5. Conclusions

The KOMB bandwidth-synthesizing software was initially developed for the K3 correlator. It was modified to handle KSP correlator output data. Although the KOMB program is large, the development was carried out smoothly by using the old software effectively. To keep the effect on the baseline analysis software as low as possible, we tried to keep the format of the KOMB output the same as that of K3's. However, part of the format was changed due to the differences in the number of channels and the number of lags. This format change was absorbed by the interface program that creates the database for the baseline analysis software.

The KSP correlation-processing system is the second system (the first is the K3 one) in Japan dedicated to geodetic VLBI observation. It enables objective evaluations of the errors generated by the correlation processing. Error at the time of processing can be evaluated simply by processing the same observation data on an independent processing system. The processing error was found to be smaller than the theoretical error for delay time, while the delay-rate error was about ten times larger than the theoretical one. However, no bias can be seen in these processing errors, so they should not affect the results of baseline analysis very much.

### Acknowledgments.

We would like to express our thanks to Jun Amagai for his helpful discussions regarding the comparison of the K3 and KSP processing results.

## References

- (1) Rogers, A. E. E.; “Very-long-baseline interferometry with large effective bandwidth for phase-delay measurements”, *Radio Sci.*, Vol.5, pp.1239, 1970.
  - (2) Kondo, T. and H. Kunimori; “III-5 bandwidth synthesizing software (KOMB) ”, *Rev. Radio Res. Lab.*, Vol.30, No.1, pp.199-216, 1984 (in Japanese).
  - (3) Kiuchi, H., T. Kondo, and M. Sekido; “3.2.1 Correlation processor”, *J. Commun. Res. Lab.*,——, 1999 (this issue).
  - (4) Van Vleck, J. H. and D. Middleton; “The Spectrum of Clipped Noise”, *Proc. IEEE*, Vol. 54, No. 1, pp. 2-19, 1966.
  - (5) Takahashi, Y. and T. Yoshino; “IV-6 parameter presumption software (KLEAR)”, *Rev. Radio Res. Lab.*, Vol.30, No.1, pp.295-309, 1984 (in Japanese).
-