Adaptive Separation of Mixed Broad-Band Sound Sources with Delays by a Beamforming Hérault–Jutten Network

Shaolin Li, *Member, IEEE*, and Terrence J. Sejnowski, *Senior Member, IEEE*

Abstract—**The Hérault–Jutten network has been used to separate independent sound sources that have been linearly mixed. The problem of separating a mixture of several independent signals in free-field conditions or a signal and echoes in confined spaces is compounded by propagation time delays between the source(s) and the microphones because the conventional Hérault–Jutten network cannot tolerate time delays. In this paper, we combine a symmetrically balanced beamforming array with the conventional Hérault–Jutten network. The resulting system can adaptively separate signals that include delays introduced by the propagation medium. The proposed algorithm has been simulated in digital communication multipath channels where intersymbol interference exists. The simulation results show two clear advantages of the proposed method over the conventional adaptive equalization: 1) there is no penalty for very long impulse responses caused by long delays, and 2) no training signals are needed for equalization. The design of a multibeamformer to handle the source separation of multiple broad-band signals is also presented.**

I. Introduction

THE problem of separating mixtures of independent signals or their delayed versions is encountered in many fields: in the "cocktail party" problem, a person wants to listen to a single sound source and filter out other interfering sources; in underwater acoustic digital communication, a receiver needs to filter out delayed versions of the transmitted signal in order to eliminate intersymbol interference (ISI). In both circumstances, the locations of sources and/or their echoes are unknown and may vary over the time.

The Hérault–Jutten (HJ) network can deal with these types of problems because of its adaptive separation ability [1]–[4]. However, in many common circumstances, such as in the "cocktail party" problem, it is not possible to obtain N distinct linear combinations of N signals without delays or phase shifts required by the HJ network. In order to generate N full rank linear combinations of inputs to the HJ network, N microphones have to be placed at different locations for N signal sources located at different places. The propagating medium between the sound sources and the microphone produces different weights on the arrivals of different sound source signals as required, and introduces significant signal delays which cannot be handled by the conventional HJ network. Platt and Faggin [7] have proposed an extension of the HJ network to estimate the matrix of delays, together with the mixing matrix $\boldsymbol{A}$.

We have investigated another method for coping with unknown delays that relies on beamforming. An array of microphones can be organized to produce N fixed beam lobes without phase shifts. This is equivalent to having N virtual directional microphones at a single spatial location each aimed at a different direction. If all of the sound sources have different directions, N full-rank linear combinations of N signals are formed, as required for the HJ network. A brief report of this method was given in [9].

II. An Overview of the Hérault–Jutten Network

The HJ network is an $N \times N$ network [1]–[4] which can be used to solve the following signal processing problem: given N observed data sequences $\boldsymbol{Y}(t) \triangleq [y_1(t), y_2(t), \cdots, y_N(t)]^T$ which are distinct linear combinations (full rank) of N physical independent signals.[1]

$\boldsymbol{X}(t) \triangleq [x_1(t), x_2(t), \cdots, x_N(t)]^T$ without time delays or phase shifts, the network can adaptively recover $\boldsymbol{X}(t)$ without *a priori* knowledge of the mixing matrix $\boldsymbol{A}$. In mathematical form,

$$\underbrace{\begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_N(t) \end{bmatrix}}_{\text{Observed Data } \boldsymbol{Y}(t)} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & & \cdots & a_{2N} \\ \vdots & & & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}}_{\text{Mixing Matrix } \boldsymbol{A}} \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_N(t) \end{bmatrix}}_{\text{Original Signal } \boldsymbol{X}(t)} \tag{2.1}$$

where $\boldsymbol{Y}(t)$ is the known observed data, $\boldsymbol{A}$ is an unknown full rank matrix, and $\boldsymbol{X}(t)$ is the signal to be recovered. It is

Manuscript received April 1, 1993.
The authors are with the Computational Neurobiology Laboratory, The Salk Institute, San Diego, CA 92186 USA
IEEE Log Number 9403705.

[1] For $\boldsymbol{X}(t)$ to be independent signals is a sufficient but not necessary condition. The HJ network can converge as long as $\int_0^T x_i(t)^{2k+1} x_j(t)^{2l+1}\, dt = 0$, $i \neq j$; see [1]–[4].

0364-9059/95$04.00 © 1995 IEEE

important to note that (2.1) is a memoryless system. Thus, at a given time $t = t_1$, the output $\boldsymbol{Y}(t_1)$ depends only on $\boldsymbol{X}(t_1)$ and not on any other past or future values of $\boldsymbol{X}(t)$.

Introduce a matrix $(\boldsymbol{I}+\boldsymbol{C})$, where $\boldsymbol{I}$ is an identity matrix and

$$\boldsymbol{C} \triangleq \begin{bmatrix} 0 & & c_{ij} \\ & \ddots & \\ c_{ji} & & 0 \end{bmatrix}. \tag{2.2}$$

The output of an HJ network is $\boldsymbol{S}(t)$:

$$\boldsymbol{S}(t) \triangleq \begin{bmatrix} s_1(t) \\ \vdots \\ s_N(t) \end{bmatrix} = (\boldsymbol{I}+\boldsymbol{C})^{-1} \begin{bmatrix} y_1(t) \\ \vdots \\ y_N(t) \end{bmatrix}. \tag{2.3}$$

The adaptive rule used in the HJ network is

$$dc_{ij}/dt = -\mu f(s_i) g(s_j) \tag{2.4}$$

where $f(s)$, $g(s)$ are different odd functions. In later simulations, we use $f(s) = s^3$ and $g(s) = s$. The quantity μ sets the rate of adaptation. When the network learning approaches an equilibrium point,

$$\begin{bmatrix} s_1(t) \\ \vdots \\ s_N(t) \end{bmatrix} \Rightarrow \begin{bmatrix} x_1(t) \\ \vdots \\ x_N(t) \end{bmatrix}. \tag{2.5}$$

At the equilibrium point, $\boldsymbol{I}+\boldsymbol{C}$ is equal to $\boldsymbol{PDA}$, where $\boldsymbol{A}$ is defined in (2.1), $\boldsymbol{D}$ is a diagonal matrix, and $\boldsymbol{P}$ is a permutation matrix (see [1]–[4]).

III. Separating Signals from Different Directions

In this section, a beamformer is constructed to form multibeams using inputs $x_i(t)$ $(i = 1, \cdots, N)$, which are independent signals or a signal with its delayed versions, and propagated from different directions. Each beam output of the beamformer is a distinct linear combination of $x_i(t)$ $(i = 1, \cdots, N)$ as required in (2.1).

Assume $2M + 1$ sensors are placed in a straight line a distance δ apart. Let the signal received by the sensor at the center be $z_0(t)$. A beam is formed by adding delays to the output of each sensor and summing them up:

$$y_n(t) = \sum_{m=-M}^{M} w_m z_m(t - m\tau_n) \tag{3.6}$$

where $y_n(t)$ is the output of the nth beam, $z_m(t)$ is the output of the mth sensor, and w_m is the weight to form a beam.

Assume that a plane wave $x_i(t)$ is arriving from an angle θ_i (see Fig. 1). The output of sensor m is

$$z_m(t) = x_i(t - m\tau_i) \tag{3.7}$$

where $\tau_i = \delta \sin(\theta_i)/c$ is the time delay between neighboring sensors and c is the wave propagation speed in the medium.[2]

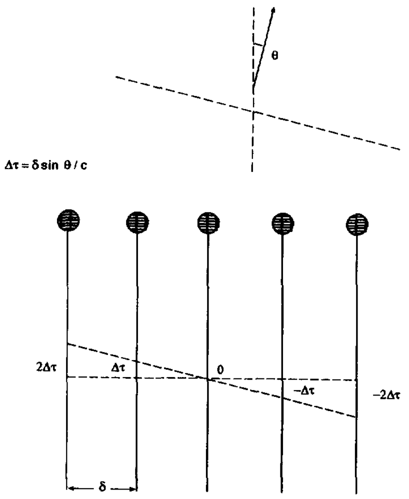


Fig. 1. The array and arriving plane wave geometry.

Substituting (3.7) into (3.6), one obtains

$$y_n(t) = \sum_{m=-M}^{M} w_m x_i[t - m(\tau_n + \tau_i)]. \tag{3.8}$$

From (3.8), the beam direction can be steered by changing τ_n. If the weights are symmetric around the center $w_m = w_{-m}$ and $w_0 = 1$, the Fourier transform of (3.8) is

$$\begin{aligned} Y_n(\omega) &= X_i(\omega) \sum_{m=-M}^{M} w_m \exp^{-im\omega(\tau_n+\tau_i)} \\ &= X_i(\omega) \underbrace{\left[1 + 2\sum_{m=1}^{M} w_m \cos\left[m\omega(\tau_n + \tau_i)\right]\right]}_{\triangleq\, a_{ni}(\omega)}. \end{aligned} \tag{3.9}$$

Since $a_{ni}(\omega)$ in (3.9) has a real value, there are no phase shifts or delays for any values of ω, τ_i, and τ_n. The symmetry of the array with respect to the central sensor eliminates the phase shifts between the beam outputs. Thus, when exact delay lines are used to form the beam, the phase shifts of the $-m$th and $+m$th sensor outputs exactly cancel. For narrow-band signals centered at ω_0, $a_{ni}(\omega_0)$ can be considered approximately constant over the bandwidth. For I incoming signals from different angles θ_i, one can form $N = I$ beams with different τ_n aimed at N different but fixed directions; the narrow-band sources will produce N linear combinations of signals without delays, which can then serve as inputs to an HJ network. This is equivalent to having N virtual directional microphones at a single spatial location (at the central sensor), each aimed at a different direction. If all of the sound sources have different directions, N full-rank linear combinations of N signals are formed, as required for the

[2] We use i as an index of incoming signals, n as an index of the beams formed, and m as an index of sensors.

HJ network. In the time domain, (3.9) has the same form as (2.1):

$$\underbrace{\begin{bmatrix} y_1(t)\\ y_2(t)\\ \vdots \\ y_N(t)\end{bmatrix}}_{\text{Outputs of } N \text{ Beams}} = \underbrace{\begin{bmatrix} a_{11}(\omega_0) & a_{12}(\omega_0) & \cdots & a_{1N}(\omega_0)\\ a_{21}(\omega_0) & & \cdots & a_{2N}(\omega_0)\\ \vdots & & & \vdots \\ a_{N1}(\omega_0) & a_{N2}(\omega_0) & \cdots & a_{NN}(\omega_0)\end{bmatrix}}_{\text{Mixing Matrix } \boldsymbol{A}} \cdot \underbrace{\begin{bmatrix} x_1(t)\\ x_2(t)\\ \vdots \\ x_N(t)\end{bmatrix}}_{\substack{\text{Original Signals}\\ \text{from Different Directions}}} . \qquad (3.10)$$

A conventional HJ network uses amplitude differences to form the different weight a_{ij} in $\boldsymbol{A}$; out method relies on differences in beam magnitudes for signal arrivals from different directions. Passing the N outputs of beams through an $N \times N$ HJ network, the output should recover the original signals $x_i(t)$ when the HJ network has reached equilibrium.

The separation of multiple broad-band signal sources is not feasible with the beamformer described above because the directional sensitivity of the sensor array decreases from high frequency to low frequency. In that case, $a_{ni}(\omega)$ in (3.9) will change with frequency, and (3.10) is no longer correct for the broad-band signals. However, this problem could be solved by replacing the weights w_m in (3.8) with filters on $x_i(t)$ before forming a beam. With this approach, it should be possible to construct a beamformer whose beam lobe width is frequency-invariant over a broad bandwidth. Therefore, (3.8) becomes

$$y_n(t) = \sum_{m=-M}^{M} h_m(t) * x_i[t - m(\tau_n + \tau_i)] \qquad (3.11)$$

where "*" stands for the time convolution. In the frequency domain, (3.11) has the form

$$Y_n(\omega) = X_i(\omega)\left[1 + 2\sum_{m=1}^{M} H_m(\omega)\cos\left[m\omega(\tau_n + \tau_i)\right]\right]. \qquad (3.12)$$

Here, we also assume the filters are symmetric to the center: $H_m(\omega) = H_{-m}(\omega)$.

We seek the filters with transfer functions $H_m(\omega)$ so that $\sum_{m=1}^{M} H_m(\omega)\cos(m\omega\tau)$ will be independent of ω over the signal bandwidth $\omega_a \le \omega \le \omega_b$. There are many ways to parameterize $H_m(\omega)$. The method we use to achieve this goal is as follows: First, choose a frequency ω_0 where $\omega_a \le \omega_0 \le \omega_b$. At the frequency ω_0, the outputs of sensors are weighted by a Gaussian window function to reduce the sidelobes, i.e.,

$$H_m(\omega_0) = e^{-m^2/\alpha_0}. \qquad (3.13)$$

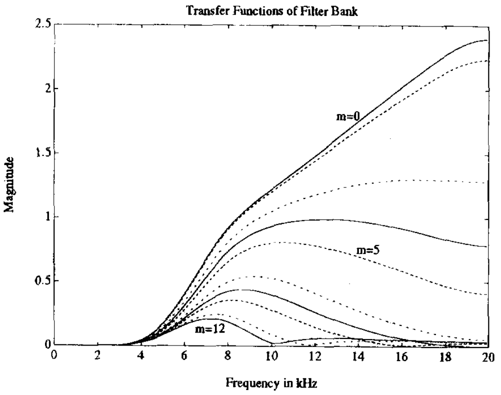


Fig. 2. The magnitude of $H_m(\omega)$ for 25 sensors over a frequency band of 5–20 kHz. Each curve represents $H_m(\omega)$ of a different sensor (because of symmetry, only 13 curves are shown here).

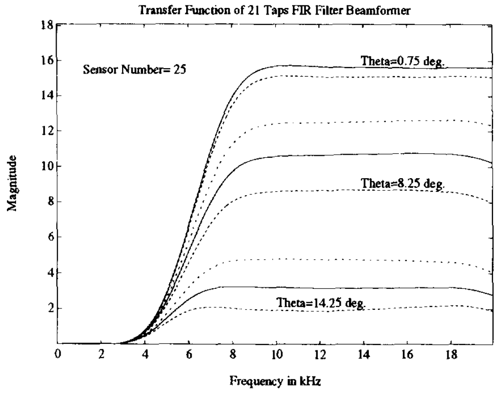


Fig. 3. The beam power pattern obtained from a 25-sensor array using the filters shown in Fig. 2. The filters are realized by 21-tap FIR filters with symmetric tap weights. The different curves correspond to different directions of arrivals from 0° to 15°. The beam power is approximately constant over the bandwidth 8–20 kHz for a given direction of arrival.

At the frequencies around ω_0, we minimize the following cost function at each frequency bin ω over a range of steering directions from $\tau = 0$ to $\tau = \tau_0$:

$$[\alpha(\omega), \beta(\omega), \gamma(\omega)] = \arg\min_{\alpha,\beta,\gamma} \int_0^{\tau_0} \left[\sum_{m=1}^{M} e^{-m^2/\alpha_0}\cos(m\omega_0\tau) - \left(\gamma e^{-m^2/\alpha} + \beta\right)\cos(m\omega\tau)\right]^2 d\tau. \qquad (3.14)$$

Then the transfer function of the filter for the mth sensor is given by

$$H_m(\omega) = \gamma(\omega)e^{-m^2/\alpha(\omega)} + \beta(\omega). \qquad (3.15)$$

Fig. 2 shows a set of calculated transfer functions of $H_m(\omega)$ over the frequency band 5–20 kHz for various sensors. At lower frequencies, $H_m(\omega)$ have Gaussian window weights (here, $\alpha_0 = 1.5$). As the frequency increases, the $H_m(\omega)$ of the two end sensors go to zero and the central $H_m(\omega)$ increases. The resulting beam power over the given frequency band is plotted in Fig. 3; the directional sensitivity is approximately uniform over the bandwidth, as required.

The relative bandwidth ratio is defined as

$$r \triangleq \frac{\omega_b - \omega_a}{\sqrt{\omega_a \omega_b}}. \tag{3.16}$$

In the example above, $r = 1.5$. In underwater communication, this ratio provides adequate bandwidth for practical purposes, as we will demonstrate. For the "cocktail party" problem, the frequency band usually is from 300 Hz to 5 kHz, which corresponds to $r = 3.84$, twice that in the case above. To obtain a uniform beam over that frequency band, two arrays could be used to form the beams: one operating from 300 Hz to 1.2 kHz, and the other one operating from 1.2 to 5 kHz. The two set of beams can be superposed together to form the desired beams.

IV. Simulations

A 2×2 HJ network with a two-beam array was simulated for a two-path channel. The array had $2M + 1 = 25$ sensors. One formed beam was aimed at 0°, and the other beam was aimed at 10°. The simulated direct path came from 2°, and the reflected path came from 8° with a delay of 12.75 ms. The signal used in the simulation was a biphase-shift keying (BPSK) signal and the carrier frequency was 14 kHz. The bit rate was 5 kb/s. With this carrier frequency and bit rate, the main signal bandwidth ranged from $\omega_a = 5$ kHz to $\omega_b = 20$ kHz. The filter bank consisted of 21-tap finite impulse response (FIR) digital filters to realize $H_m(\omega)$ in (3.15). Fig. 3 shows the output beam power of such a beamformer. One can see that for a given direction of arrival, which is represented by one of curves, the output beam power is approximately constant over the signal bandwidth from 5 to 20 kHz. The whole simulation was carried out in the time domain. The carrier synchronization was obtained using a matched filter which convolved the received signal with a time-reversed replica of the signal transmitted.

As the delay time between the different paths increases, unlike a conventional equalization filter which requires additional taps [12], there is no penalty in our algorithm because the signal and its delayed versions are treated as two uncorrelated signals. The information in the signal bandwidth and angle of arrival is enough to resolve the different paths. The estimation of time delays, which is difficult in underwater channels, can be performed after the signals from multichannels are detected. The signal and its delays can be considered as uncorrelated when the delay time is longer than a symbol length and there is no correlation between the adjacent bits transmitted. If the adjacent bits are correlated, decorrelation can also be achieved by modulating the transmitted bit train with a pseudorandom bit train which has the same bit rate as the data. When the delay time is smaller than the symbol length, which is 0.2 ms in the simulation, the HJ network will not converge in general. For short delay times, the spread spectrum technique can be used to reduce the symbol length. In practice, the main concern usually is for very long delay times rather than very short delay times. A 0.2 ms delay presents only a 6.8 cm path length difference in air and 30 cm in water, which are very short lengths indeed, and the length can be further decreased as the signal bit rate increases.

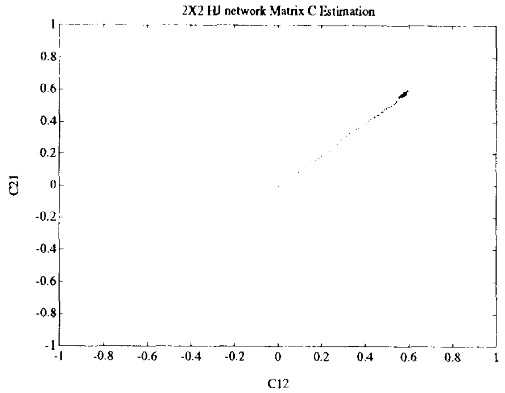


Fig. 4. The coefficients c_{12} and c_{21} converge to the correct values during the adaptation. The initial condition is $c_{12} = c_{21} = 0$.

Fig. 4 shows the adaptation process of the 2×2 HJ network which used the two beam outputs of the array mentioned above as two inputs. The two quantities c_{12} and c_{21} of the matrix $\boldsymbol{C}$ in (3.7), which are plotted as the x axis and y axis, respectively, were set to zero at the beginning of adaptation. We used $f(s) = s^3$ and $g(s) = s$ for c_{12} and c_{21} in the HJ network adaptation [see (2.4)]. The adaptation rate parameter was $\mu = 0.2$.

Two kinds of white noises were included in the simulation. One was the noise associated with the signal (if this directional noise is strong enough, it can be treated as an independent signal). The other is the nondirectional noise that includes the surrounding noise of the sensors and the electronic noise from the receiver circuit. With these noise sources, (3.6) becomes

$$\boldsymbol{Y}(t) = \boldsymbol{A}[\boldsymbol{X}(t) + \boldsymbol{W}_1(t)] + \boldsymbol{W}_2(t) \tag{4.17}$$

where $\boldsymbol{W}_1(t)$ and $\boldsymbol{W}_2(t)$ are independent white Gaussian random vectors with power spectral densities S_{w1} and S_{w2}, respectively. The signal-to-noise ratios, defined as $\text{SNR}_1 \triangleq 10\log_{10}(S/S_{w1})$ and $\text{SNR}_2 \triangleq 10\log_{10}(S/S_{w2})$, were $\text{SNR}_1 = 7$ dB and $\text{SNR}_2 = 27$ dB in the simulation.

The quantities c_{12} and c_{21} quickly converged to the correct values in simulation. Fig. 5 shows the detection of the bit train when only the output of the beam aimed at 0° was used. There were several bit errors as shown in Fig. 5. Fig. 6 gives the detection of the same bit train using one of the HJ network outputs. The first 30 data bits were received with about the same reliability as before, but after the HJ network converged, the transmission was error free. Fig. 7 shows the detection when only the output of the beam aimed at 10° is used. Fig. 8 gives the detection of the bit train using the other HJ network output. When μ increases, the adaptation speed increases, but the fluctuation of the final values of c_{12} and c_{21} around the correct values also increases. There is a tradeoff between the adaptation speed and the separation accuracy. When the directions of signal arrivals in the underwater acoustic channels change rapidly, the degree of separation will

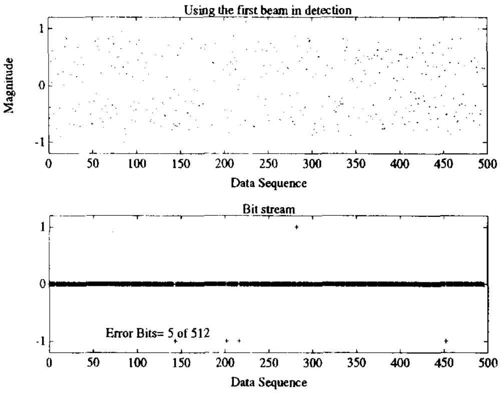


Fig. 5. The detection of a BPSK data stream of 512 b in 0.1 s using only the beam aimed at 0° without equalization. When a bit was transmitted correctly, there was + located at zero. There were five bit errors as shown in the lower panel at +1 and −1.

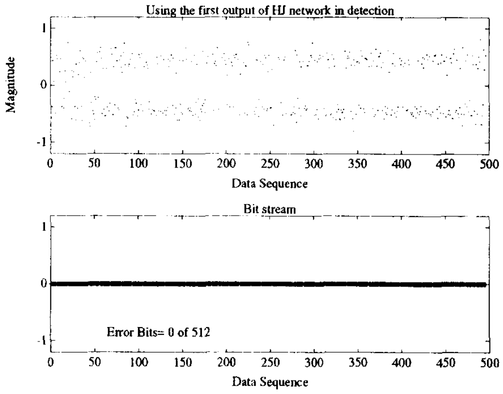


Fig. 6. The detection of the same data stream as in Fig. 6 of 512 b in 0.1 s using one of the HJ network outputs. There were no errors. See Fig. 8 for the performance of the second output channel.

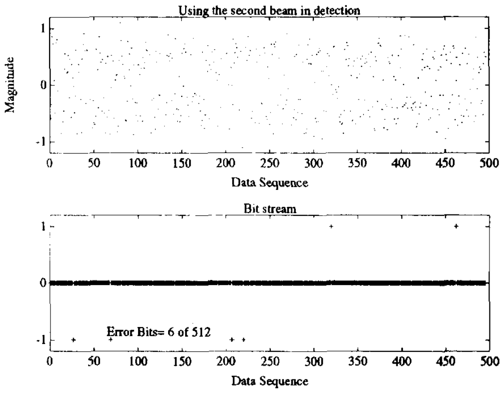


Fig. 7. The detection of the data stream of 512 b in 0.1 s using only the beam aimed at 10°. There were six bit errors, shown in the lower panel.

suffer. In our simulation, c_{12} and c_{21} adapted from the random initial condition to their correct values within 10–20 ms. Thus, the algorithm is fast enough to handle time-variant fading in underwater acoustic channels.

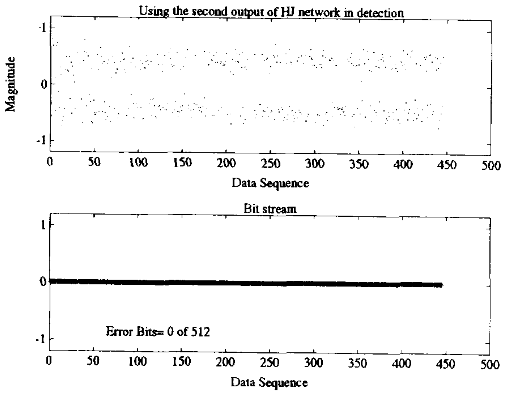


Fig. 8. The detection of the data stream of 512 b in 0.1 s using the second HJ network output. There were no bit errors. The gap at the end of the bit stream is due to the 12.75 ms delay in the pathway.

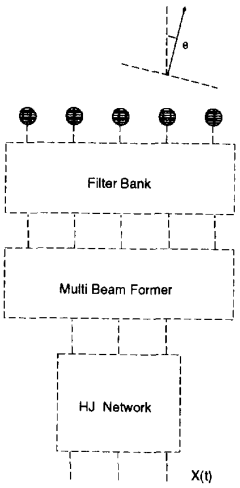


Fig. 9. The implementation diagram of the proposed system.

V. Implementation

Fig. 9 shows a diagram of a complete system for the proposed adaptive signal source separation. The filter bank, beamformer, and HJ network can be implemented for real-time processing in several different ways.

A. Digital

FIR digital filters can be used to implement $H_m(\omega)$ of (3.15) using DSP chips. The motivation to use an FIR filter is due to its linear phase property which is required by the beamformer at the next stage. Everything will be the same as in the simulation, except that the system runs in real time using DSP chips instead of a computer. A filter bank of transfer functions as $H_m(\omega)$ in Fig. 2 can be realized by

$$y(n\,\Delta T) = B_0 x(n\,\Delta T) + \sum_{l=1}^{L} B_l[x(n\,\Delta T - l\,\Delta T) + x(n\,\Delta + l\,\Delta T)] \quad (5.18)$$

where B_l is the first $L+1$ coefficients of the inverse discrete Fourier transform of $H_m(\omega)$.

B. Analog

The main computational burden of the digital implementation is to solve N unknowns from the N equations in (2.3) in real time at least at the Nyquist sampling rate. When the number of sources N is large and the sampling frequency is high, the analog implementation has a great advantage over the general-purpose DSP chip implementation. The analog implementation can preprocess an enormous amount of data from the sensor array, separate the signals among the paths with a moderate accuracy, and leave the detection to the digital processing part.

Analog VLSI HJ network chips already exist [5], [6], and it should be possible to implement the entire system including the filter bank, beamformer, and HJ network using CMOS analog VLSI technology on a single chip. The filter bank can be implemented by either the charge-coupled device (CCD) split-electrode transversal filter, the bucket brigade device (BBD) transversal filter, or a liner phase transconductance capacitor filter. We have successfully implemented a CMOS three-order Bessel transconductance capacitor filter bank and a BBD beamformer with 15 or more microphone inputs and six beams [10]. The next step is to integrate the whole system on one chip.

VI. Conclusion

We have presented a system for separating independent broad-band sound sources and multipath delays using a conventional HJ network and beamforming. Our system relies on a symmetric multibeamformer with frequency-equalized sensors to cancel time delays and frequency variations of its inputs to the HJ network.

Our technique can be used in underwater acoustic communication where, usually, several constantly changing paths exist and cause severe ISI in the data link. Unlike other methods [11], [12] that have been proposed for reducing ISI in multipath fading channels, which either treat the other multipath signals as noise or result in a high computational complexity for long delays, our system can recover information from each path without using training signals. It is therefore possible to achieve faster signal transmission rates than with other methods. The simulations show that our method can significantly reduce the error rate, and may be a practical approach to reliable high-speed acoustic telemetry. For application to the "cocktail party" problem, it could be used to separate various voice and music sources in video conferencing.

The analog implementation can be realized in a single analog VLSI chip, forming a very compact system. The digital implementation is suitable for the "cocktail party" problem where the sampling rate required for the audio frequency is relatively low. At present, our system is restricted to bandpass communication channels, but this limitation could be overcome with several arrays.

Our beamformer solution to the problem of delays could be used as well to augment other methods for blind separation [13].

Acknowledgment

We are grateful to A. G. Andreou, W. Liu, and M. H. Cohen for help and advice with our implementation of the BBD beamformer and transconductance–capacitor filter.

References

[1] J. Hérault and C. Jutten, "Space or time adaptive signal processing by neural network models," in *Neural Networks for Computing, AIP Conf. Proc.*, vol. 151, Snowbird, UT, 1986, pp. 207–211.

[2] C. Jutten and J. Hérault, "Blind separation of sources, Part I: An adaptive algorithm based on a neuromimetic architecture," *Signal Processing*, vol. 24, no. 1, pp. 1–10, 1991.

[3] P. Comon, C. Jutten, and J. Hérault, "Blind separation of sources, Part II: Problem statement," *Signal Processing*, vol. 24, no. 1, pp. 11–20, 1991.

[4] E. Sorouchyari, "Blind separation of sources, Part III: Stability analysis," *Signal Processing*, vol. 24, no. 1, pp. 21–29, 1991.

[5] M. H. Cohen, P. O. Pouliquen, and A. G. Andreou, "Silicon implementation of an auto-adaptive network for real time separation of independent signals," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1991, pp. 2971–2974.

[6] ——, "Analog VLSI implementation of an auto-adaptive network for real time separation of independent signals," in *Advances in Neural Information Processing Systems, Vol. 4.* San Mateo, CA: Morgan Kaufmann, pp. 805–812, 1992.

[7] J. C. Platt and F. Faggin, "Network for the separation of sources that are superimposed and delayed," in *Advances in Neural Information Processing Systems, Vol. 4.* San Mateo, CA: Morgan Kaufmann, pp. 730–737, 1992.

[8] J. Catipovic, "Performance limitations in underwater acoustic telemetry," *IEEE J. Oceanic Eng.*, vol. 15, pp. 205–216, July 1990.

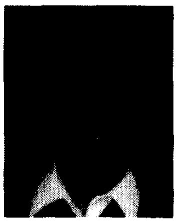
[9] S. Li and T. Sejnowski, "Automatic separation of mixed sound sources by a beamforming Hérault-Jutten network," in *Proc. Annu. Res. Symp. Inst. for Neural Computation*, Univ. California, San Diego, June 1992.

[10] S. Li, T. Sejnowski, and A. Andreou, "Implementation of FIR filter and beamformer using analog CMOS VLSI," in preparation.

[11] J. Fisher, K. Bennett, S. Reible, J. Cafarella, and I. Yao, "A high data rate, underwater data-communication transceiver," in *Proc. Oceans '92*, vol. 2, 1992, pp. 571–576.

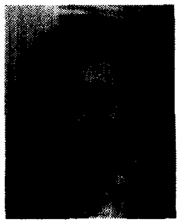
[12] J. Proakis, "Adaptive equalization techniques for acoustic telemetry channels," *IEEE J. Oceanic Eng.*, vol. 16, pp. 21–31, Jan. 1991.

[13] A. J. Bell and T. J. Sejnowski, "A non-linear information maximization algorithm that performs blind separation," in *Advances in Neural Information Processing Systems*, vol. 7, San Mateo, CA: Morgan Kaufmann, 1995.



Shaolin Li (S'86–M'91) received the B.S. degree from the Nanjing Institute of Meteorology, Nanjing, People's Republic of China, in 1982, the M.S. degree from Drexel University, Philadelphia, PA, in 1985, both in physics and fluid dynamics, and the M.S. and Ph.D. degrees in electrical engineering from Yale University, New Haven, CT, in 1988 and 1991, respectively. His dissertation topic dealt with the estimation of source locations in underwater acoustic channels and its Cramer–Rao bounds.

He has been a Research Associate at The Salk Institute, San Diego, CA, since 1991. His present research interests are in the areas of neural networks, analog VLSI, digital communications, and underwater acoustic signal processing.



Terrence J. Sejnowski (M'83–SM'92) received the B.S. degree in physics from the Case-Western Reserve University, Cleveland, OH, in 1968, the M.A. degree in physics from Princeton University, Princeton, NJ, in 1970, and the Ph.D. degree in physics from Princeton University in 1978.

From 1978 to 1979 he was a Postdoctoral Fellow in the Department of Biology at Princeton University, and from 1979 to 1982 a Postdoctoral Fellow in the Department of Neurobiology at Harvard Medical School. In 1982 he joined the faculty of the Department of Biophysics at the Johns Hopkins University, Baltimore, MD. He is currently an Investigator with the Howard Hughes Medical Institute and a Professor at The Salk Institute for Biological Studies, San Diego, CA, where he directs the Computational Neurobiology Laboratory. He is also a Professor of Biology and an Adjunct Professor in the Departments of Electrical and Computer Engineering and Computer Science and Engineering at the University of California, San Diego, where he is also Director of the Institute for Neural Computation and Director of the McDonnell–Pew Center for Cognitive Neuroscience. His primary interests are in the theory and applications of neural systems.

Dr. Sejnowski received a Presidential Young Investigator Award in 1984. In 1988 he founded the journal *Neural Computation* (M.I.T. Press). For 1993–1994 he was a Sherman Fairchild Distinguished Scholar at the California Institute of Technology.