# 6. EQUALIZATION AND ECHO CANCELLATION

## 6. 1 Channel Equalization

The most frequently used channel in for transmitting digital data is a telephone channel, which is severely band-limited as shown below. On the left, we show the insertion-loss of the channel for a typical toll connection, where the insertion loss is defined by:

$$Insertion\ Loss = 10 \log_{10}(P_0 / P_2) \quad dB \tag{6.1}$$

where $P_2$ is the power delivered to a load by the channel and $P_0$ is the power delivered to the same load when the channel is removed. From here we can easily conclude that this channel is dispersive and the leading+trailing tails of adjacent pulses will overlap and the *Intersymbol Interference (ISI)* is eminent. As a consequence of this dispersion, it is observed that the pulse shape in a telephone channel is more effected by the ISI than by AWGN.

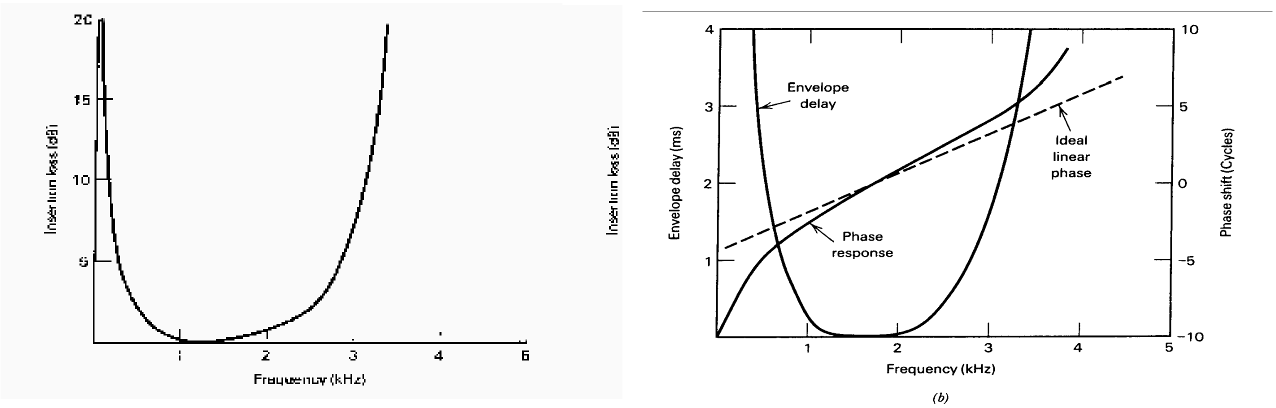
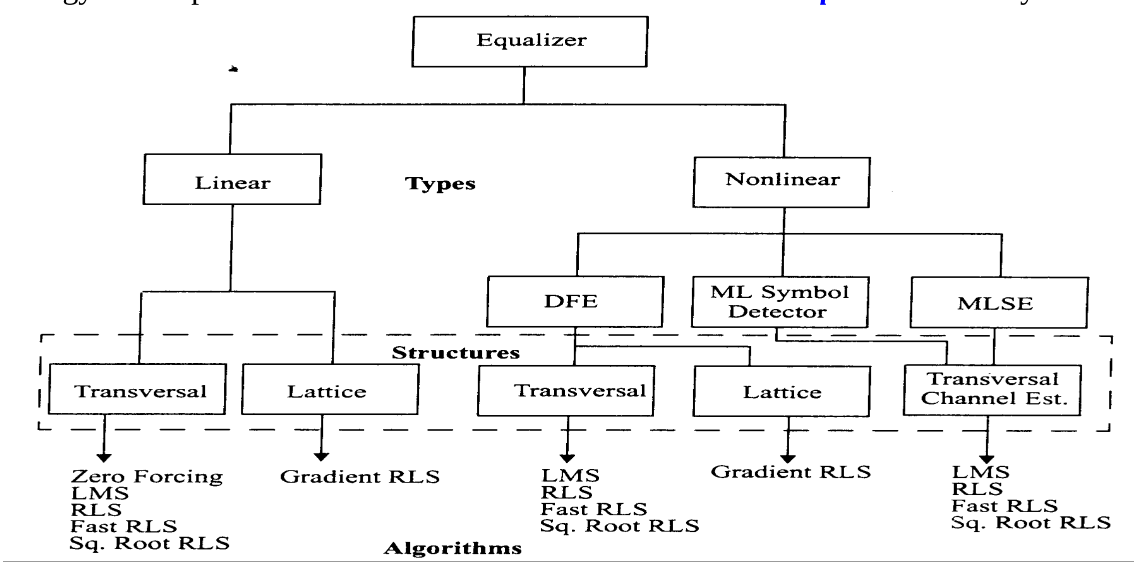
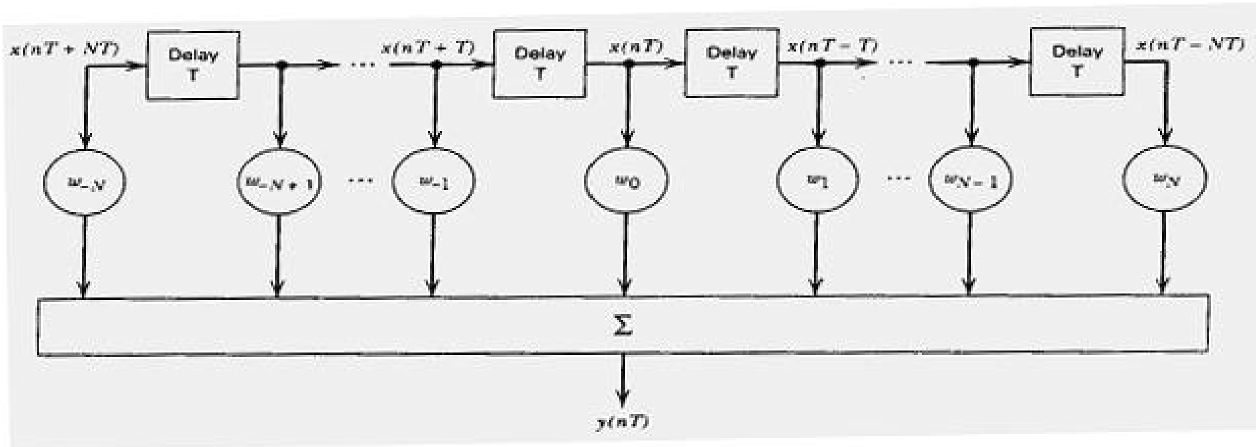



**Figure 7.21** (*a*) Amplitude response of typical toll connection. (*b*) Envelope delay and phase response of typical toll connection. (Bellamy, 1982.)

- If the channel is known precisely, it is theoretically possible to eliminate ISI at the sampling instants by using suitable transmit and receive to flatten the overall spectral behavior. In practice, we do not have the exact shape of the channel and also the finite-size limitations of the transmitter and receiver filters causes unavoidable ISI.
- Strategy to compensate this intrinsic distortion is to include ***an equalizer*** in the system.



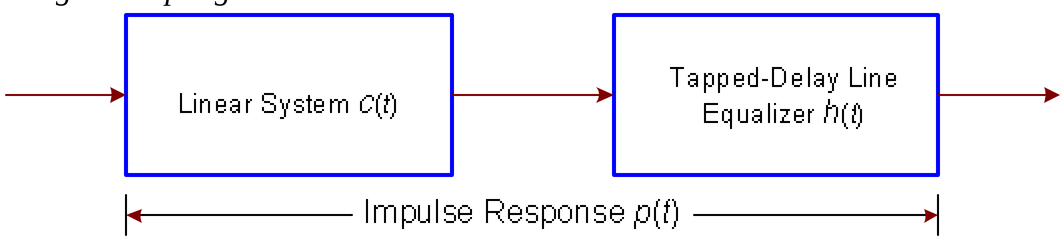

**Linear Equalizers (Tapped-Delay Line):** It is very frequently used in practice and it has (2N+1) taps, symmetrical wrt the current sample as shown below.




If the tap weights $\{w_{-N}, \cdots, w_{-1}, w_0, w_1, \cdots, w_N\}$ are not adjustable then the equalizer is non-adaptive and its impulse response is given by:

$$h(t) = \sum_{k=-N}^{N} w_k . d(t - kT) \tag{6.2}$$

where T is the symbol (sample) duration. In order to cancel ISI, suppose that we connect the tapped-delay-line equalizer in cascade with a linear system c(t), the combined structure will satisfy the *Nyquist Criterion of ISI free sampling, which states that the tails of adjacent pulses will have zero-crossing at sampling instant*.




Let the impulse response of the equalized system be:

$$p(t) = c(t) * h(t) = c(t) * \sum_{k=-N}^{N} w_k\, d(t - kT) = \sum_{k=-N}^{N} w_k . c(t) * d(t - kT) = \sum_{k=-N}^{N} w_k . c(t - kT) \tag{6.3}$$

At the sampling instants: $t = nT$ , the above summation results in a discrete convolution sum:

$$p(nT) = \sum_{k=-N}^{N} w_k . c((n-k)T) \tag{6.4}$$

For no ISI, the above criterion requires that:

$$p(nT) = \begin{cases} 1 & n = 0 \\ 0 & n = \pm 1, \pm 2, \cdots, \pm N \end{cases} \tag{6.5}$$

If we place these (2N+1) conditions on (6.4) we obtain a set of (2N+1) simultaneous solutions, the solution of which will yield the zero ISI equalizer tap values. In other words:

$$\sum_{k=-N}^{N} w_k . c((n-k)T) = \sum_{k=-N}^{N} w_k . c_{n-k} \tag{6.6}$$

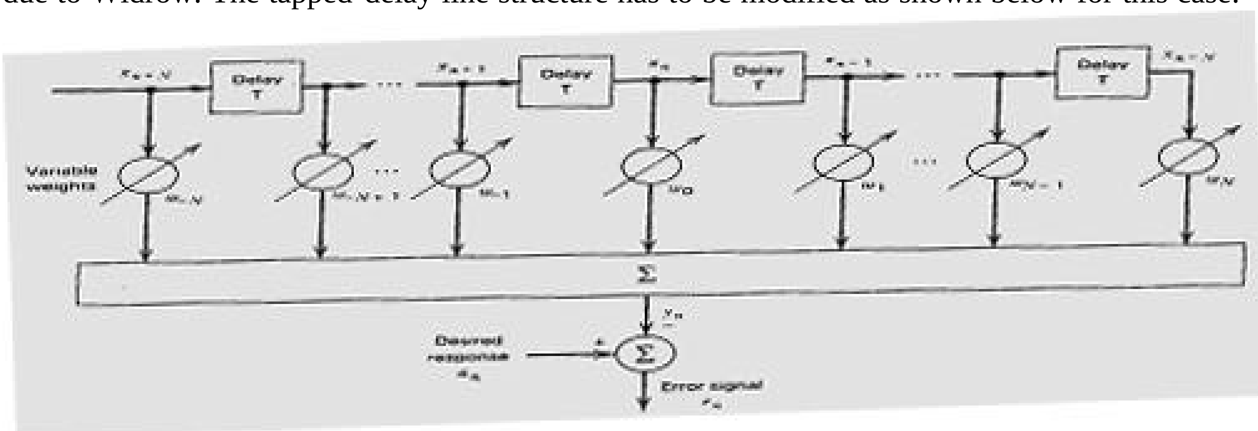
Equivalently we have the matrix form:

$$
\begin{bmatrix}
c_0 & \cdots & c_{-N+1} & c_{-N} & c_{-N-1} & \cdots & c_{-2N} \\
\vdots & & \vdots & & & & \\
c_{N-1} & \cdots & c_0 & c_{-1} & c_{-2} & \cdots & c_{-N-1} \\
c_N & \cdots & c_1 & c_0 & c_{-1} & \cdots & c_{-N} \\
c_{N+1} & \cdots & c_2 & c_1 & c_0 & \cdots & c_{-N+1} \\
\vdots & & \vdots & \vdots & \vdots & & \vdots \\
c_{2N} & \cdots & c_{N+1} & c_N & c_{N-1} & \cdots & c_0
\end{bmatrix}
\begin{bmatrix} w_{-N} \\ \vdots \\ w_{-1} \\ w_0 \\ w_1 \\ \vdots \\ w_N \end{bmatrix}
=
\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}
$$

The equalizer of (6.6) is referred as a *Zero-Forcing (ZF) Equalizer*, which requires the solution of a (2N+1)-by-(2N+1) linear system. Matrix theory based solutions are not difficult to implement the tap weights are constant. However, almost all the equalizers used in digital communication applications have adaptive tap weights and hence, the matrix inversion based techniques are impractical.

## 6.2 Adaptive Equalizers

We may adjust the tap weights either as the channel conditions change (asynchronous) or at each sample duration (synchronous). The last equalizer is easy to implement if the computations are manageable. The most common method in practice is the Least-mean-Square (LMS) algorithm due to Widrow. The tapped-delay line structure has to be modified as shown below for this case:




Let $a_n, y_n, and\ e_n$ denote the desired response, the actual response of the equalizer, and the error signal, respectively by:

$$e_n = a_n - y_n \tag{6.7}$$

The Mean-Squared Error (MSE) is the ubiquitous cost function: $\mathrm{E} = E[e_n^2]$ used in performance evaluations. The gradient of this MSE with respect to $k^{th}$-tap weight is expressed by:

$$\partial \mathrm{E}/\partial w_k = 2.E[e_n.\partial e_n/\partial w_k] = -2.E[e_n.\partial y_n/\partial w_k] = -2.E[e_n.x_{n-k}] = -2R_{ex}(k) \tag{6.8}$$

which is the cross-correlation between the error signal and the input signal for a lag of $k$ samples. Thus, the gradient of the MSE cost function is simply:

$$\partial \mathrm{E}/\partial w_k = -2R_{ex}(k)$$

The optimality condition for MSE is the standard derivative set to zero form for each tap:

$$\partial \mathrm{E}/\partial w_k = 0 \Rightarrow R_{ex}(k) = 0 \quad for\ k = 0, \pm 1, \cdots, \pm N \tag{6.9}$$

This can be interpreted as the MSE performance of the tap weights forms a multi-dimensional bowl shaped surface. If we can reach the bottom of the bowl successively then we have the minimum value for each tap. The basic technique for finding the bottom of a bowl expediently is to choose the steepest descents in each slope.

## 6.3 Steepest Descent and the LMS Algorithm

**Steepest descent algorithm:** The tap weights are recursively obtained by:

$$w_k(n+1) = w_k(n) - \frac{1}{2} m . \frac{\partial E}{\partial w_k} \quad for\ k = 0, \pm 1, \cdots, \pm N \tag{6.10}$$

where $m$ is a step-size parameter. Using the cross-correlation representation we can re-write this:

$$w_k(n+1) = w_k(n) + m.R_{ex}(k) \quad for\ k = 0, \pm 1, \cdots, \pm N \tag{6.11}$$

This formulation requires the knowledge of the exact cross-correlation function, which is not available when operating over an AWGN and time-varying channel. Instead, let us use an instantaneous estimate for the cross-correlation based on one computation instead of expectation:

$$\hat{w}_k(n+1) = \hat{w}_k(n) + m.e_n.x_{n-k} \quad for\ k = 0, \pm 1, \cdots, \pm N \tag{6.12}$$

This algorithm is known as the LMS algorithm due to Widrow. Below we will present the algorithmic form of LMS. However, you can find both an M-file versions for MATLAB implementation and C-Code if you refer to our home presented in the Appendices.

### LMS Algorithm:

Let the (2N+1)-by-(2N+1) vectors be the tap-inputs and the weights of the equalizer, respectively:

$$\underline{X}_n = [x_{n+N}, \cdots, x_{n+1}, x_n, x_{n-1}, \cdots, x_{n-N}] \tag{6.13a}$$

$$\underline{\hat{W}}_n = [\hat{w}_{-N}(n), \cdots, \hat{w}_{-1}(n), \hat{w}_0(n), \hat{w}_1(n), \cdots, \hat{w}_{+N}(n)]^T \tag{6.13b}$$

We may then use the matrix notation to recast the convolution sum in a compact form:

$$y_n = \underline{X}_n^T . \underline{\hat{W}}_n \tag{6.14a}$$

where the right-hand side is commonly known as *the inner product* of two vectors.

**Step 1: Initialization:** Set the tap weight At time $t=T$ to zero: $\underline{\hat{W}}_1 = \underline{0}$.

**Step 2: Computation:** For n=1,2,…., compute:

$$y_n = \underline{X}_n^T . \underline{\hat{W}}_n \tag{6.14a}$$

$$e_n = a_n - y_n \tag{6.14b}$$

$$\underline{\hat{W}}_{n+1} = \underline{\hat{W}}_n + m.e_n.\underline{X}_n \tag{6.14c}$$

**Step 3: Continuation Check:** Continue computation until the steady-state conditions are reached.

- There are two modes of operation for an adaptive equalizer: The training mode and the decision-directed mode.
- During *the training mode*, a known sequence is transmitted and a synchronized version of this sequence is generated at the receiver and the tap weights are adjusted in accordance with the LMS algorithm.
- After the training, the equalizer is switched *to the decision-directed* mode. Here the error signal is the difference between the final estimate (not necessarily) correct estimate of the transmitted symbol. Then the decisions made by the receiver are *correct with high probability.*

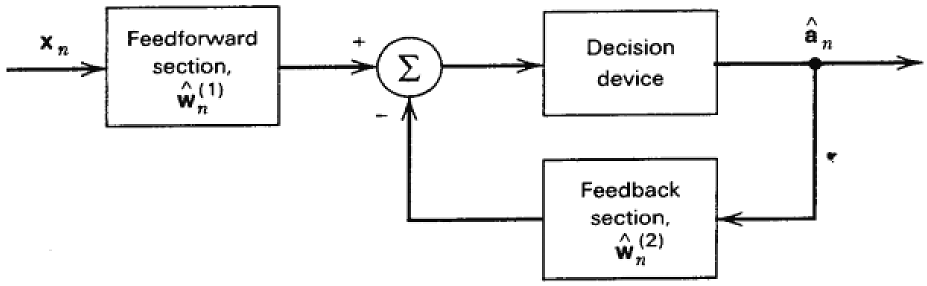
## 6.4 Decision-Feedback Equalization

The idea of decision-feedback equalizer is to use data decisions made on the basis of the precursors of the channel impulse response to take care of postcursors as described below.

- Consider a baseband channel with an impulse response samples $\{h_k\}$; *where* $h_k = h(kT)$. The response of this channel to an input sequence is simply the discrete convolution:

$$y = \sum_k h_k . x_{n-k} = h_0 . x_n + \sum_{k<0} h_k . x_{n-k} + \sum_{k>0} h_k . x_{n-k} \tag{6.15}$$

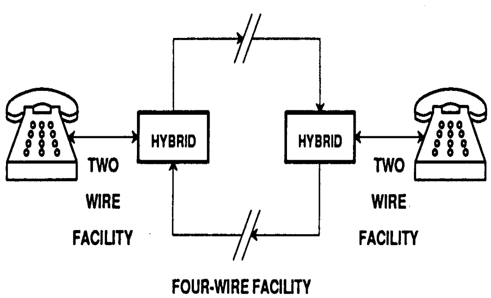
Here the first term is the desired data symbol. The second term is due to the *precursors* of the channel impulse response that occur before the main sample. The third term is due to *postcursors* of the channel impulse response that occur after the main sample. This logic works efficiently provided the decisions have to be correct. The block diagram of a decision-feedback equalizer is shown below:




- Here the feed-forward section is a tapped-delay line filter whose taps are spaced at the reciprocal of the signaling rate. The feedback section consists of an another tapped-delay line filter, similarly, whose taps are spaced at the same rate. The function of this feedback is to subtract out that portion of the ISI produced by the previously detected symbols from the estimate of the future samples.
- Note that the inclusion of the decision device in the feedback loop make the equalizer intrinsically non-linear and therefore, much more difficult to analyze. However:
- The LMS algorithm is used to jointly adapt both the feedforward and the feedback taps-weights. Therefore there will two sets of equations to replace (6.14) and two step-size parameters: $m_1, m_2$ in the computations.

## 6.5 Echoes in Channel

A simplified telephone channel setups for voice and modem communication and a circuit for an electronic hybrid are shown below.



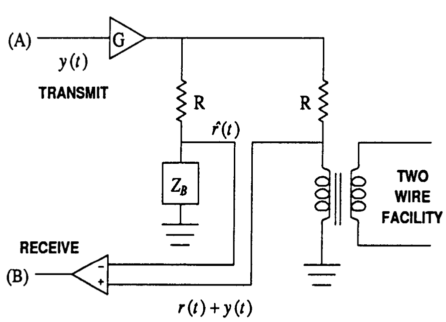
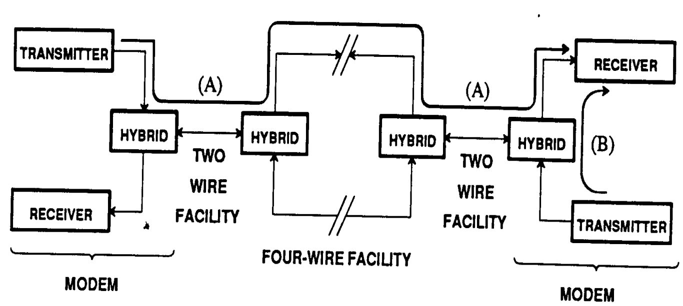



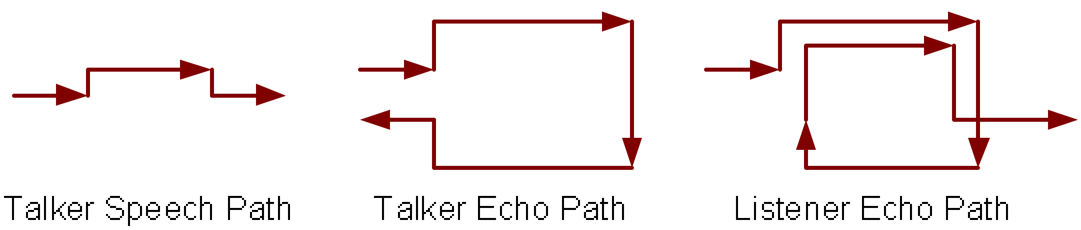

**Figure 5-37.** An electronic hybrid. To avoid leakage of the receive signal (A) into the transmit path (B) the impedance $Z_B$ should exactly match the impedance of the transformer and two-wire facility.




**5-39.** Two modems connected over a single simplified telephone channel. The re-
on the right must be able to distinguish the desired signal (A) from the signal leaked
wn transmitter (B).

The local loop, which is the twisted-wire pair connecting the central office with the customer premise, is used for full-duplex (transmission in both directions), whereas, the inter-office transmission over trunks are four-wire. A device called *"hybrid"* separates the two directions of the communication and an electronic circuit model is also shown. The signal from the other end of the 2-wire facility is fed through the receive port.

- The transmit signal appears at the transformer as a voltage divider with impedances $R, Z_0$, where the latter is the input impedance of the 2-wire facility. We normally cancel this undesired feedthrough by constructing another voltage divider with a balance impedance $Z_B$.
- When $Z_B = Z_0$, we have the complete balance from transmit to receive port.
- Due to varying distances involved from facility to facility, a fixed compromise impedance is used in practice and this results in leakage of a portion of the signal (A) to (B) with an attenuation as small as 6-10 dB.
- For the case of voice communication, the signal and two types of echo paths resulting from the above impedance mismatch are shown below.




- Talker echo is the signal which leaks in the far-end hybrid and returns to the sender (talker).
- The listener echo is the component of the talker echo that leaks through the near-end hybrid and returns again to the listener.
- The third type of echo is the "ambient chamber echo," typically seen in the "cock-pit" and "vehicular" chambers, which is very critical in cellular communications. (Interested readers should read our paper included in the appendix and also hear a real-life demonstration by visiting our homepage. The last example in the LMS algorithm examples in Appendix!!)
- The length of the telephone channel determines the round-trip echo delay. Echoes from the near end of the connection typically undergo 0-2 ms of delay; whereas far-end echoes can have round-trip delays of 10-60 ms for terrestrial facilities, or up to 600 ms on satellite connections.
- To mitigate the effects of echo on speech quality or digital data integrity, several strategies co-exist on the network:

1. For short delays, loss added in the talker speech path. This is advantageous because the echoes can experience this loss more than once.
2. The loss in (1), plus the loss of the subscriber loops at each end, is the source of the attenuation that must be accommodated by the communication equipment. It can be as high as 40 db at 1,004 Hz.
3. For longer delays, devices known as echo suppressors, or more appropriately, echo cancellers have to be used.
4. In the case of modems, there are two 2-to-4-wire conversions and hence, two sets of imperfect balance impedances in the hybrids. The hybrid echo loss can be as low as 6 dB, and the received signal may have experienced as much as 40 dB loss, so the desired far-end signal may be as much as 34 dB below the echo. This necessitates the inclusion of adaptive echo cancellation schemes at baud rates 2,400 bps and higher. Important standardized voiceband data modems are summarized below, where EC stands for systems with echo cancellers and FDM indicates the usage of frequency-division multiplexing. Symbols in the last column are associated with the underlying digital communication scheme.

| speed (b/s) | symbol rate | duplex (method) | CCITT std. | modulation |
|---|---|---|---|---|
| ≤ 300 | ≤ 300 | full(FDM) | V.21 | 2-FSK |
| 1200 | 1200 | half | V.23 | 2-FSK |
| 1200 | 600 | full(FDM) | V.22 | 4-PSK |
| 2400 | 1200 | half | V.26 | 4-PSK |
| 2400 | 600 | full(FDM) | V.22bis | 16-QAM |
| 2400 | 1200 | full(EC) | V.26ter | 4-PSK |
| 4800 | 1600 | half | V.27 | 8-PSK |
| 4800 | 2400 | full(EC) | V.32 | 4-QPSK |
| 9600 | 2400 | half | V.29 | 16-AM/PM |
| 9600 | 2400 | full(EC) | V.32 | 32-QAM+TC |
| 14,400 | 2400 | full(EC) | V.32bis | 128-QAM+TC |
| ≤ 28,800 | ≤ 3429 | full(EC) | V.fast(V.34) | 1024-QAM+TC |

## 6.6 Echo Cancellers

Consider the generic echo canceller, where the local transmitter signal $y(t)$ at port A generates an undesired echo signal $r(t)$.

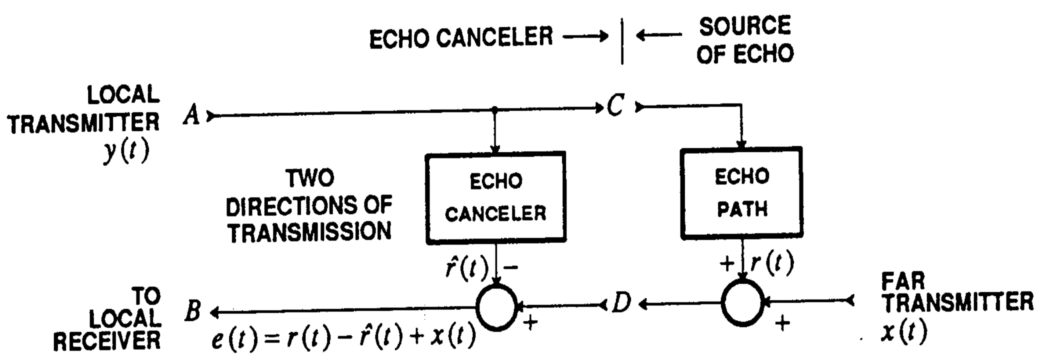



**Figure 19-2.** The principle and notation of an echo canceler.

1. At the output of the hybrid (port D), this signal is superimposed with the far transmitter signal $x(t)$.
2. From the knowledge of the local transmitter signal the canceller has to generate a replica of the echo $\hat{r}(t)$.
3. This replica is subtracted from the echo plus the far transmitter signal to yield $e(t)$, which ideally contains the far transmitter signal $x(t)$ alone.
4. Echo canceller is usually implemented using a finite transversal filter discussed above.
5. Essentially the algorithms used in equalizers, in particular LMS, can be applied to tackle this problem. (In Appendix, we present our paper on acoustic echo cancellers using LMS

**APPENDICES**

Reprinted from: Proceedings of the ASILOMAR-1993 Conference on Signals, Systems & Computers, Vol. I, pp. 776-780, IEEE Computer Society Press, 1993.

# IMPLEMENTATION OF A 5.0 KB/S. CODER FOR VEHICULAR APPLICATIONS
## Part II : Acoustic Echo and Noise Canceller

**Erhun Arkan, Hüseyin Abut, Simon Pelling, and frederick j. harris**
E.C.E. Department, San Diego State University, San Diego, CA 92182, and
Gonçalo C. Marques, Instituto de Engenharia e Sistemas de Computadores (INESC), Lisbon, Portugal

ABSTRACT [1]

*In this paper, we present the results of our research on the real-time implementation of a digital echo and vehicular noise canceller on a TMS320C50/51 platform to comply with the ongoing efforts on half-rate coder implementation at various industrial and academic research and development centers. The proposed echo and noise cancellers are expected to be interfaced with the speech coding system for use in an automobile environment subject to both road, engine and wind noises and acoustic echo generated inside a vehicle, which is presented in a companion paper. At the time of this writing the echo canceller based on adaptive filtering techniques for automobile acoustics has been fully implemented using a DSP board equipped with a Texas Instruments TMS320C50/ 51 processor and it is hosted on a PC. We have also completed the simulations for road, engine, and wind noise canceller based on beam forming and adaptive filtering techniques for a microphone array system.*

### INTRODUCTION

The increasing demand for digital cellular telephony and other new services including multi-media communications promp ted numerous studies on implementing not only the algorithms for half-rate speech coding using the available DSP processors on the market but also the need to enhance the speech quality subject to both degradations due to road, engine, and wind noise and the echo present in the near-end speaker side --sources effecting the car phone input. All of these must be achieved with a single DSP chip in order the system to be both cost-effective and power efficient.

The speech quality of the emerging totally digital cellular phones will greatly depend upon the speech quality available at the near-end transmitter end. Despite this, the literature and the implementation efforts indicate a major emphasis on the speech coding and channel transmission issues of cellular telephony, with little reported research on the effects of the ambient acoustical noise and the echo in the vehicular environment. Recently, a few sound works have appeared addressing to both of these problems [1-5]. First we will present our echo canceller and then the noise enhancement techniques.

### ACOUSTIC ECHO CANCELLER

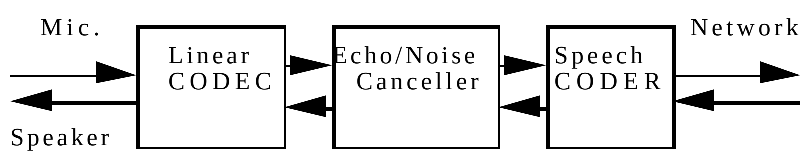



**Figure 1. The Cellular Speech Coder with Acoustic Echo& Noise Canceller.**

For echo cancellation task, we have extensively utilized architecture of a digital voice echo canceller implemented on a TMS32020 by Messerschmitt, et al. [2] and the software programming tools developed for the speech coder of Part I.

---

[1] This work is partially supported by the National Science Foundation and the Rockwell International Corporation

The interaction of the echo & noise cancellers and speech coder subsystems is shown in Figure 1. However, we show the principle of echo canceller portion based on an adaptive finite impulse response (FIR) structure in a vehicular environment in Figure 2.

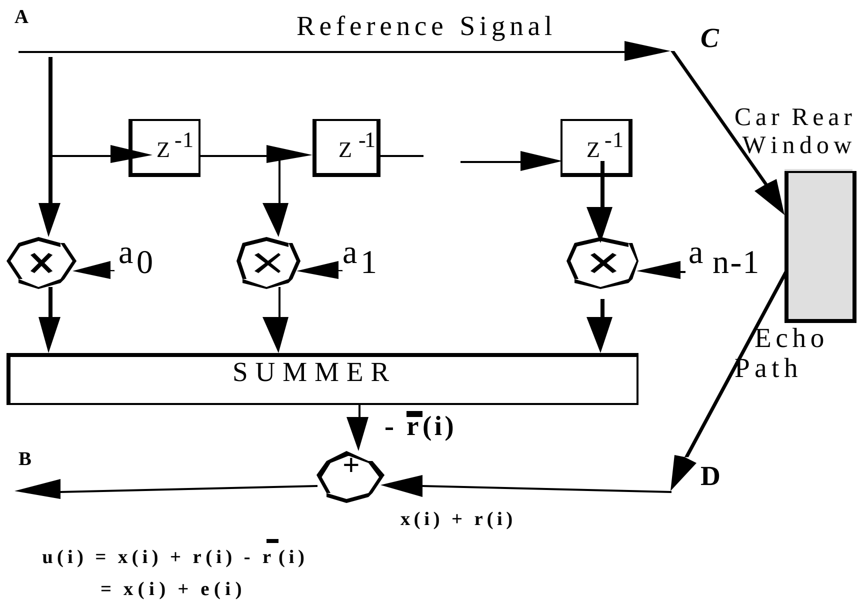



**Figure 2. Adaptive Echo Canceller Using an FIR Filter Structure.**

The longest undesired echo **r (i)** takes place in the path from the speaker output of the cellular phone **C** to the furthest point in the car, usually the rear window, and the back to the microphone input **D**. For most mid-size vehicles this distance is approximately 16 feet, which corresponds to 16 ms. delays at a sampling rate of 8,000 samples per second. In order to achieve the Industry Standard IS-54 performance requirements it became necessary to have a 128-tap adaptive FIR filter.

This echo of the far-end speaker leaks to the system to produce annoying effects at the far side. When the near-end speaker speaks though the echo is mixed with his speech and it is transmitted in the form of degradation. So, the task of the acoustical echo canceller is to adaptively cancel the echo during non-speech periods; but it must cancel the echo only when the near-end speaker speaks. In other words, no adaptation is to be performed during those instances. In order to achieve that near-end speaker activity detection is needed. Messerschmitt et. al have developed a coefficient adaptation algorithm based LMS algorithm and we have used their technique with a basic difference in how we implement the multipliers needed in updating. We have replaced costly multipliers with novel Shift-Accumulate operations to reduce the computational load.

## IMPLEMENTATION

We have implemented the acoustic echo canceller to cancel echo of 16 ms. or less on a DSP board by Spectrum Signal Processing, Inc., based on the Texas Instruments TMS320C50/51 processor and it is hosted on a 486-PC. This development system has been equipped with a user interface, two serial ports, and a parallel expansion system (DSP~LINK) to communicate with other systems. In addition, we have acquired the complete set of software tools from Texas Instruments including a C-compiler, an assembler and a linker. We have written the final code in assembly language to achieve the computational target of 3.0 MIPS or less for this echo canceller module. However, we had also written everything in C to simulate in the PC workstation.

The DSP hardware board modules and the software development flow diagram have been depicted in Figures 2 and 3, respectively, where the C-compiler accepts the C-source code and produces a TMS320C50/51 assembly language code. We compare this translation with our own version of the assembly implementation. The assembler translates this last code into object files. The linker combines object files into a single executable object module. The analog I/O tasks and waveform handling, display, and graphics are handled via the Dual Communication Module installed in the TMS320C30 Development System. The DAT is used for storing speech and noise samples from analog sources, including the database gathered from the field tests. The object module is loaded into the TMS320C50/51 application board at run-time

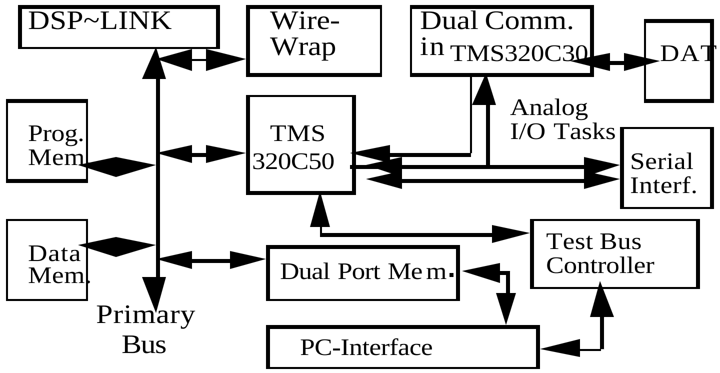



**Figure 2. DSP Development Board Modules.**

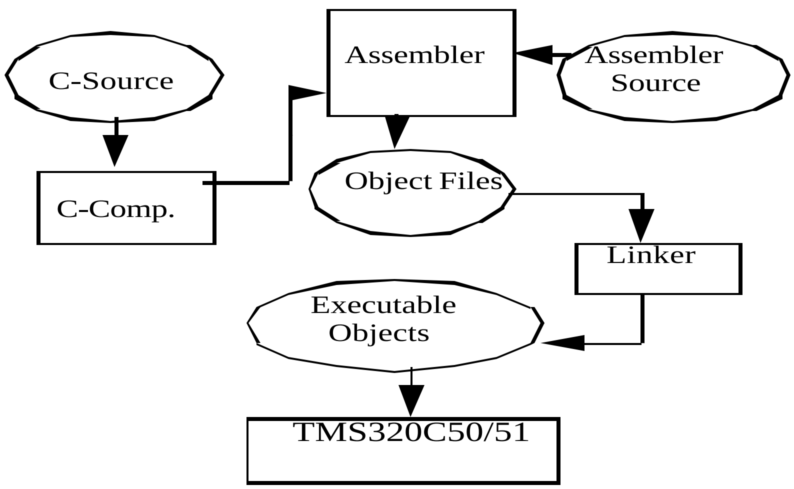



**Figure 3. TMS320C50/51 Software Development Flow Diagram.**

## ECHO CANCELLER RESULTS

In our implementation of this acoustic echo canceller we used one of the internal timers of the TMS320C50/51 processor to check the percentage of real-time at which each function operates. Depending upon the number segments of the 128-tap FIR filter is being adapted we have obtained 2.983 MIPS - 4.069 MIPS. In their original implementation, Messerschmitt et al. have achieved the task at a computational load of 5.06 MIPS for the same performance. To test that we have incorporated our measurements on their figures 13 and 14 [2, pp. 426]. The results indicate that our present implementation is approximately 20% faster than their version, which is very critical for the single chip implementation objective of the overall speech coder with enhancement packages. Next, we have modified our updating strategy to compute and adapt the portions of the coefficient set two cycles but do not update the next three cycles. This way we have reduced the overall computational cost to 2.983 MIPS at a minimal degradation in performance. But it was still under the IS-54 specifications. This final acoustic echo canceller offers savings of 40% from the baseline system of [2].

## VEHICULAR NOISE CANCELLER

In the vehicular hands-free cellular communication framework, we have observed that the degradation in the intelligibility and the general quality of the cellular speech due to the engine, road, and wind noise components is equally disturbing as the vehicular echo of the previous section. In order to have a feel for the issues involved we have collected road, engine, and wind noise samples through a single microphone array using a DAT. In addition, we have obtained a copy of the speech database using a microphone array developed at Texas Instruments [3][2].

Our observations from these limited tests and the literature [3-5] indicate that a microphone array and a noise cancellation using beam forming techniques followed by an adaptive filtering process will be required to combat all of the ills mentioned in this paragraph.

**Beamforming Algorithm:** Beamforming techniques have found practical use in radar, sonar, radio astronomy, geophysics, and biomedical signal processing applications. The most simple form of beamforming is called the delay and sum beamforming, which compensates the delay of the target signal and sums the signals in the beam so that the target signals have the same phase while the interfering signals exhibit different phase.

Here we will try to use the delay and sum beamforming technique to cancel the noises coming from the engine, the wind, especially critical when the windows of the vehicle are down, and the road noise coming from other vehicles and the road itself. There are a few studies in the literature on this for speech recognition in a hands-free telephone setup [3,4,5,6]. In our studies, we have placed between four to seven microphones in a linear array as shown in Figure 4. If we assume the speed of sound is 340 m/s then the microphones are needed to be placed at 5 cm. intervals in order to cover a spectrum up to 3,400 Hz [3,6]. Since the cellular phone speech has a bandwidth of 300-3,200 Hz our selection of 5 cm. for microphone spacing is a reasonable choice.

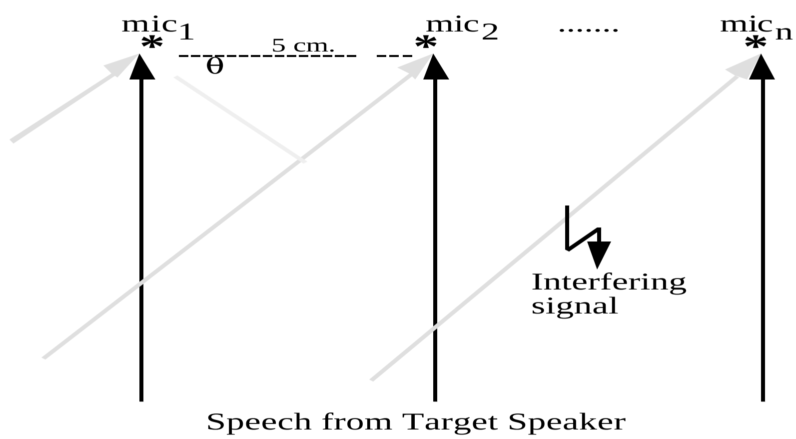



**Figure 4. Experimental Setup for Noise Cancelller**

Assuming the minimum distance between the target speaker and the microphones is greater than twice the total array aperture it is reasonable to assume the far-field condition. In our case, for seven microphones the array aperture is 30 cm or less then the speaker is to talk from a distance of 60 cm or more. Our second assumption was that the interfering signal is coming from the left of the driver with an angle of $\theta$ = 58.2 degrees. This choice corresponds to a delay of one sample between adjacent microphones. Our third assumption was that the speaker position is fixed during a test session. These assumptions lead to ideal far-field situation.

However, actual measurements in a compact or even a mid-size car are not completely satisfying these conditions due to "spherical spreading" as reported in [3]. In reality, the speaker, usually the driver moves when he drives and there could be interspeaker interference from passengers in the car. Because of all of these a number of powerful algorithms do not yield successful results in vehicular noise cancellation applications.

We have tested the above microphone array setup with road noise collected in a mid-size car along the Interstate-15 freeway during a rush hour traffic. Our database included noise segments recorded under the following two conditions: a) the vehicle is not moving and the windows are down; b) the vehicle is moving and the windows are again down. It is worth noting that the cancellation tasks will be less demanding if the windows were closed tightly. We have added these noise samples to our clean speech as shown in Figure 4 to create a microphone array setup.

---

[2] This is a copy of the database used by Oh, Viswanathan, and Papamichalis [3]. We would like to acknowledge Stephen Oh and Panos Papamichalis for their assistance.

We have measured the performance of the delay and sum beamformer quantitatively using the signal-to-noise ratio (SNR) and segmented SNR (SegSNR) improvements as our yardsticks, where the definition of the SNR improvement is given by:

$$\mathrm{SNR} = 10\mathrm{loq}\left(\frac{P_{in}}{N_{in}}\right) - 10\log\left(\frac{P_{out}}{N_{out}}\right)$$

Similarly, the SegSNR improvement values are computed using the above equation segment at-a-time, where each segment contains 1000 samples. These are shown in Table 1. As it can be seen from these values that the beamformer can cancel up to 5.9 dB when the vehicle is not moving. However, the SNR improvement is not more than 3.593 dB when the vehicle is moving at 55 mph. These results can be explained by the fact that there are a number of additional noise sources when the vehicle is moving and these noise components are time-varying at a different pace than the stationary case.

**<u>Beamforming followed by Adaptive Filtering Algorithm:</u>** The performance of the noise canceller can be significantly improved if the beamformer is followed by an adaptive filter bank, where there is one adaptive filter for each microphone in the array. This structure can be classified as a modified version of the Griffiths-Jim adaptive beamformer [7]. Initially, we have tried to use the Griffiths-Jim system for noise cancellation.

However, as it was noticed by Claesson, et al.[5], the algorithm did not yield good results since our assumption of far-field holds marginally in a vehicular cellular phone environment. This results from the fact that the speaker is moving and he is close to the microphone array. This yields almost a near-field situation and has a non-negligible physical spreading. To improve the performance of the adaptive beamformer we have decided to include speech detection in the system as in the echo canceller. We train our adaptive filters when the target speaker is silent. When the target speaker talks the detector instructs the system to stop updating the coefficients of the filter bank. However, it continues to cancel noise components as shown in Figure 5 with the filter parameters measured during a 2.0 s. silent period.

**Table 1. Results of Beamforming Experiments**

| No. of Microphones | Car not moving; Windows Down | | Car Moving at 55 mph; Windows Down | |
|---|---|---|---|---|
| | SNR (dB) | SegSnr(dB) | SNR (dB) | SegSNR (dB) |
| 4 | 3.111 | 2.920 | 1.598 | 1.578 |
| 5 | 4.091 | 3.772 | 2.158 | 2.088 |
| 6 | 4.940 | 4.500 | 2.847 | 2.730 |
| 7 | 5.910 | 5.101 | 3.593 | 3.445 |

**Table 2. Experimental Results for Beamformer Followed by Adaptive Filtering**

| No. of Microphones | Car not moving; Windows Down | | Car Moving at 55 mph; Windows Down | |
|---|---|---|---|---|
| | SNR (dB) | SegSnr(dB) | SNR (dB) | SegSNR (dB) |
| 4 | 14.888 | 14.986 | 13.336 | 13.356 |
| 5 | 16.740 | 16.840 | 15.329 | 15.368 |
| 6 | 17.980 | 18.109 | 16.938 | 16.950 |
| 7 | 19.172 | 19.276 | 18.147 | 18.163 |

# of taps: 128; Beta=$0.8\times10^{-9}$; Filter training duration= 16,000 samples

We have obtained the SNR and SegSNR improvements shown in Table 2. As it can be seen from this and the previous tables, the SNR improvements are drastically higher in this case. For instance, we have obtained 18.147 dB for a microphone array with 7-elements, whereas the corresponding SNR improvement was only 3.593 dB for the beamformer alone. The difference for the two different scenarios reduced to almost 1.1 dB from 2.3 dB.

## CONCLUSIONS

In this work, we present an echo canceller based on a DSP chip and effective techniques for canceling engine, road, and wind noise in a vehicular environment.

## REFERENCES


[1] D.J. Goodman, "Trends in Cellular and Cordless Communications," IEEE Communications Magazine, Vol. 29, No. 6, pp. 31-40, June 1991.

[2] D. Messerschmitt, D. Hedberg, C. Cole, A. Haoui, and P. Winship, "Digital Voice Echo Canceller with a TMS32020," in *Digital Signal Processing Applications, K*.-S. Lin, Ed., Prentice-Hall, Englewood, Cliffs, N.J. 1987.

[3] S. Oh, V. Viswanathan, and P. Papamichalis, "Hands-Free Voice Communication in an Automobile With a Microphone Array," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. I-281-284, March 1992, San Francisco, CA.

[4] M.M. Goulding and J.S. Bird, "Speech Enhancement for Mobile Telephony," IEEE Trans. on Vehicular Technology, Vol. 39, pp. 316-326, Nov. 1990.

[5] I. Claesson, S.E. Nordholm, B.A. Bengtsson, and P. Erickson, "A Multi-DSP Implementation of a Broad-Band Adaptive Beamformer for Use in a Hands-Free Mobile Radio Telephone," IEEE Trans. on Vehicular Technology, Vol. 40, pp. 194-201, Feb. 1991.

[6] Y. Grenier, "A Microphone Array for Car Environments," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. I-305-308, March 1992, San Francisco, CA.

[7] L.J. Griffiths and C.W. Jim, "An Alternative Approach to Linearly Constrained Adaptive Beamforming," IEEE Trans. on Antennas Propag., Vol. AP-30, pp. 27-34, January 1982.

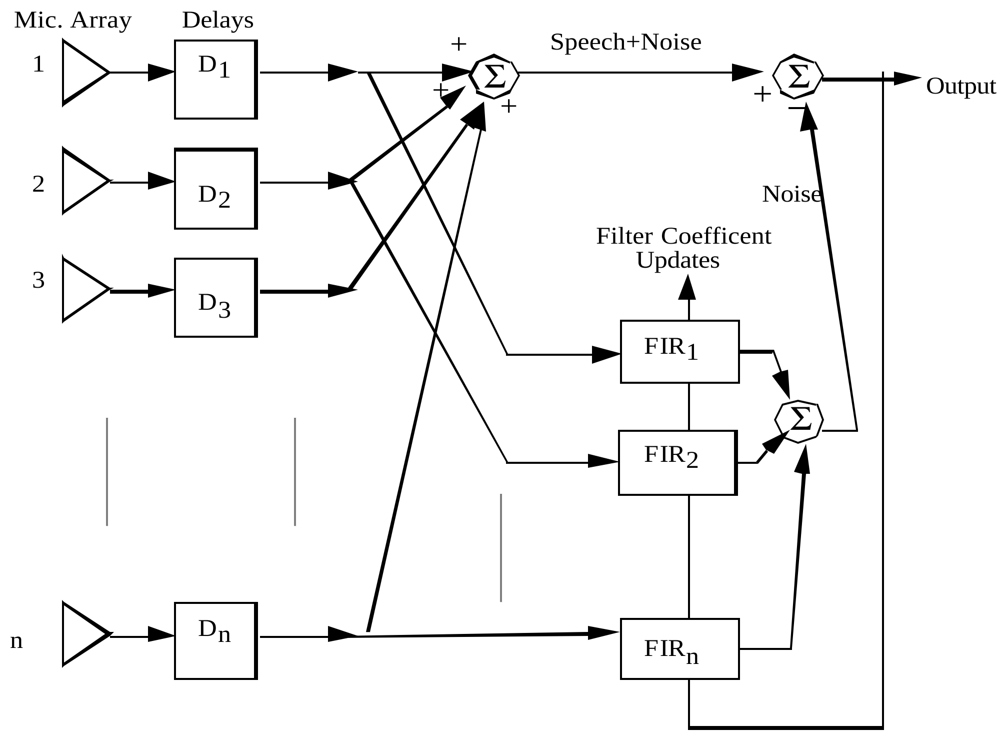




**Figure 5. Noise Canceller with Beamformer Followed by an Adaptive Filter Bank.**

**Note:** The TMS320C5X code for the Echo Canceller portion of this work is available in Mr. E. Arkan's Thesis.

**Appendix on Student Works on LMS Algorithm and Its Applications**

# *Interactive Classroom Applications*

**This page is located at: dspserv.sdsu.edu under "Interactive Classes: CE309"**

## Contents Page

**☐Applications Ready to Use**


| | |
|---|---|
| Introduction by H. Abut | Interactive Classroom Examples of Students |
| Sampling | The Sampling Theorem |
| Aliasing | Aliasing |
| Fourier Series | Fourier Series and Fourier Transform |
| Discrete and Fast Fourier Transform | DFT / FFT |
| Convolution and Correlation | Convolution and Correlation |
| Modulation | ASK, PSK and FSK |
| Comparison of FIR / IIR Filters | Finite Impulse Response (FIR) / Infinite Impulse Response (IIR) Filters |
| Filtering (Simulink Application) | Filtering Using Butterworth Filters |
| Comparison of Filters (Simulink Application) | Lowpass Filters |
| Adaptive Filtering and Automatic Noise Cancellation | AF and ANC using Least Mean Squares |
| Neural Networks | Noise Cancellation |


## Introduction to Interactive Classroom Courses

## And Examples from NTU Students

### Dr. Huseyin Abut

Currently two courses at SDSU and one at Nanyang Technological University in Singapore can be considered as good examples of the interactive instructional environment concept. Two other SDSU courses contain some of the features and we label them as "Courses in Progress."

CE 309 is a final year projects course in a Three-year Computer Engineering Program at Nanyang Technological University in Singapore. In this program, every student is required to complete a project during his or her third/final year (FYP) under the supervision of a faculty member. If they are in the four-year Honors Program, however, the students have to complete another project called "Honors Year Project (HYP)." Three FYP students were supervised by Prof. H. Abut while he was visiting the School of Applied Science during the academic year 1996-97.

The works by two students were exceptionally well prepared and suitable for this new instructional mode. That is, the page CE309 is primarily their contribution to the classroom. Prof. H. Abut was simply the embracing instructor. His role was to guide the students throughout the year to forge their knowledge on these subjects and assist them on various toolboxes of the Matlab package, clarify technical questions, and edit their material. In other words, the students have performed almost all the technical work.

It is worth noting that the material presented here is not completely interactive yet. Since we do not give the participant an option to change any of the designs on-line, this classroom can be better described as a collaborating classroom at this time. Instead, we give him/her two options: Either go over the limited number of cases we supply reference to or download the Matlab script and use it as you please. In the next phase of the project, we plan to have an option to make changes on-line.

## Description of Examples

In the first two examples, the notions of ***sampling theorem (Nyquist Theorem)*** and ***the aliasing distortion*** are presented. In each case, a short description of the problem is given. Key equations are provided and a few specific examples are included. The user can reach any of these by clicking the appropriate entry. Then the page for that particular application comes to the whiteboard. The user is also provided with return options. In each case, the Matlab script is a stand-alone program, which can be downloaded very easily.

In the next three applications, the concepts of Fourier analysis, convolution, and correlation are discussed. First, the Fourier series representations of periodic signals and Fourier transform of aperiodic waveforms are presented. Next, basic Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT) techniques are explained with examples. Fast convolution algorithm based on FFT is demonstrated in computing the response of an electromagnetic channel to a typical excitation signal. Similarly, the periodicity of a signal corrupted by a uniform noise is obtained from the correlation analysis.

In the next application, three basic digital communication schemes (ASK, FSK, and PSK) are presented. There are three applications on digital filters and their applications.

In the Adaptive Filtering application, we present the LMS algorithm and its four colorful examples. They are adaptive line enhancement, interference cancellation, adaptive design of digital filters, and acoustic echo cancellation. In particular, we have an audio demo in the last example. Here a 2.6 second segment of speech is first corrupted by its four echoes of varying scaling factors and delays and then an echo canceller based on the LMS algorithm is employed to reconstruct the original speech.

Finally, we have used neural networks to enhance signals corrupted by Gaussian noise. We have experimented with different learning rates for the system designed.