Adaptive Noise Cancellation

Aarti Singh
1/ECE/97
Dept. of Electronics & Communication
Netaji Subhas Institute of Technology

Certificate

This is to certify that Aarti Singh, student of VIIIth semester B.E. (Electronics and Communication) carried out the Project on “Adaptive Noise Cancellation” under my guidance during a period of four months –February to May 2001.

It is also stated that this Project was carried out by her independently and that it has not been submitted before.

Prof. M.P. Tripathi
Advisor

Prof. Raj Senani
(HOD-ECE)

Acknowledgement

I take this opportunity to express my regards and sincere thanks to my advisor and guide **Prof. M. P. Tripathi**, without whose support, this project would not have been possible. His constant encouragement and moral support gave me the motivation to carry out the project successfully.

I am also indebted to **Dr. Harish Parthasarthy** for his valuable and timely guidance. The discussions with him helped a lot in developing an in-depth understanding of the topics involved.

Also, I would like to thank Sh. Bhagat, the DSP Lab Incharge, who helped me with the lab facilities whenever I needed them.

Aarti Singh

Abstract

This Project involves the study of the principles of Adaptive Noise Cancellation (ANC) and its Applications. *Adaptive Noise Cancellation* is an alternative technique of estimating signals corrupted by additive noise or interference. Its advantage lies in that, with no apriori estimates of signal or noise, levels of noise rejection are attainable that would be difficult or impossible to achieve by other signal processing methods of removing noise. Its cost, inevitably, is that it needs two inputs - a *primary* input containing the corrupted signal and a *reference* input containing noise correlated in some unknown way with the primary noise. The reference input is adaptively filtered and subtracted from the primary input to obtain the signal estimate. Adaptive filtering before subtraction allows the treatment of inputs that are deterministic or stochastic, stationary or time-variable.

The effect of uncorrelated noises in primary and reference inputs, and presence of signal components in the reference input on the ANC performance is investigated. It is shown that in the absence of uncorrelated noises and when the reference is free of signal, noise in the primary input can be essentially eliminated without signal distortion. A configuration of the adaptive noise canceller that does not require a reference input and is very useful many applications is also presented.

Various applications of the ANC are studied including an in depth quantitative analysis of its use in canceling sinusoidal interferences as a notch filter, for bias or low-frequency drift removal and as Adaptive line enhancer. Other applications dealt qualitatively are use of ANC without a reference input for canceling periodic interference, adaptive self-tuning filter, antenna sidelobe interference canceling, cancellation of noise in speech signals, etc. Computer simulations for all cases are carried out using Matlab software and experimental results are presented that illustrate the usefulness of Adaptive Noise Canceling Technique.

I. Introduction

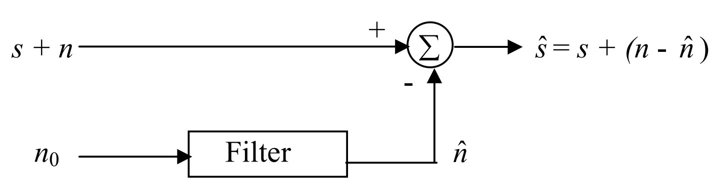
The usual method of estimating a signal corrupted by additive noise is to pass it through a filter that tends to suppress the noise while leaving the signal relatively unchanged i.e. *direct filtering*.

The design of such filters is the domain of optimal filtering, which originated with the pioneering work of Wiener and was extended and enhanced by Kalman, Bucy and others.

Filters used for direct filtering can be either *Fixed* or *Adaptive*.

1. *Fixed filters* - The design of fixed filters requires a priori knowledge of both the signal and the noise, i.e. if we know the signal and noise beforehand, we can design a filter that passes frequencies contained in the signal and rejects the frequency band occupied by the noise.
2. *Adaptive filters* - Adaptive filters, on the other hand, have the ability to adjust their impulse response to filter out the correlated signal in the input. They require little or no a priori knowledge of the signal and noise characteristics. (If the signal is narrowband and noise broadband, which is usually the case, or vice versa, no a priori information is needed; otherwise they require a signal (desired response) that is correlated in some sense to the signal to be estimated.) Moreover adaptive filters have the capability of adaptively tracking the signal under non-stationary conditions.

Noise Cancellation is a variation of optimal filtering that involves producing an estimate of the noise by filtering the reference input and then subtracting this noise estimate from the primary input containing both signal and noise.



It makes use of an auxiliary or reference input which contains a correlated estimate of the noise to be cancelled. The reference can be obtained by placing one or more sensors in the noise field where the signal is absent or its strength is weak enough.

Subtracting noise from a received signal involves the risk of distorting the signal and if done improperly, it may lead to an increase in the noise level. This requires that the noise estimate $\hat{n}$ should be an exact replica of n. If it were possible to know the relationship between n and $\hat{n}$, or the characteristics of the channels transmitting noise from the noise source to the primary and reference inputs are known, it would be possible to make $\hat{n}$ a close estimate of n by designing a fixed filter. However, since the characteristics of the transmission paths are not known and are unpredictable, filtering and subtraction are controlled by an adaptive process. Hence an adaptive filter is used that is capable of adjusting its impulse response to minimize an error signal, which is dependent on the filter output. The adjustment of the filter weights, and hence the impulse response, is governed by an adaptive algorithm. With adaptive control, noise reduction can be accomplished with little risk of distorting the signal. Infact, Adaptive Noise Canceling makes possible attainment of noise rejection levels that are difficult or impossible to achieve by direct filtering.

The error signal to be used depends on the application. The criteria to be used may be the minimization of the mean square error, the temporal average of the least squares error etc. Different algorithms are used for each of the minimization criteria e.g. the *Least Mean Squares* (LMS) *algorithm*, the *Recursive Least Squares* (RLS) algorithm etc. To understand the concept of adaptive noise cancellation, we use the minimum mean-square error criterion. The steady-state performance of adaptive filters based on the *mmse* criterion closely approximates that of fixed Wiener filters. Hence, Wiener filter theory (App.I) provides a convenient method of mathematically analyzing statistical noise canceling problems. From now on, throughout the discussion (unless otherwise stated), we study the adaptive filter performance after it has converged to the optimal solution in terms of unconstrained Wiener filters and use the LMS adaptive algorithm (App.IV) which is based on the Weiner approach.

II. Adaptive Noise Cancellation – Principles

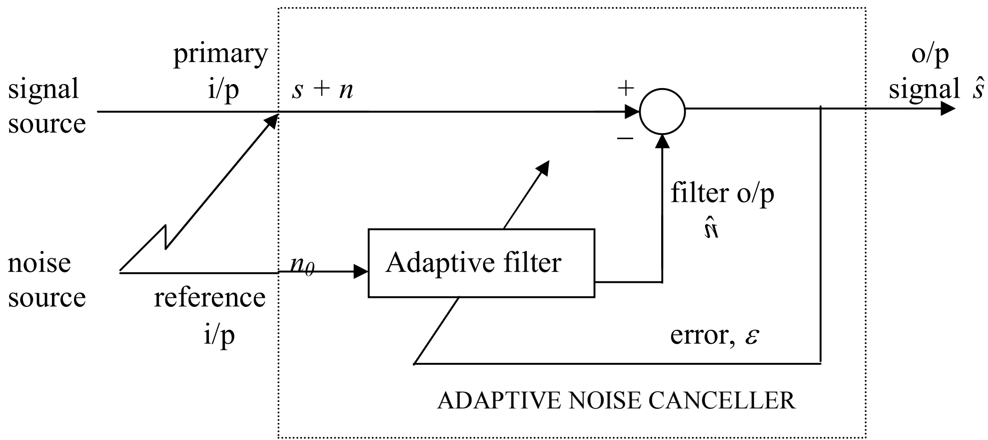


Fig. 1 Adaptive Noise Canceller

As shown in the figure, an Adaptive Noise Canceller (ANC) has two inputs – primary and reference. The primary input receives a signal *s* from the signal source that is corrupted by the presence of noise *n* uncorrelated with the signal. The reference input receives a noise n_0 uncorrelated with the signal but correlated in some way with the noise *n*. The noise n_o passes through a filter to produce an output $\hat{n}$ that is a close estimate of primary input noise. This noise estimate is subtracted from the corrupted signal to produce an estimate of the signal at $\hat{s}$, the ANC system output.

In noise canceling systems a practical objective is to produce a system output $\hat{s} = s + n - \hat{n}$ that is a best fit in the least squares sense to the signal s. This objective is accomplished by feeding the system output back to the adaptive filter and adjusting the filter through an LMS adaptive algorithm to minimize total system output power. In other words the system output serves as the error signal for the adaptive process.

Assume that s, n_0, n_1 and y are statistically stationary and have zero means. The signal s is uncorrelated with n_0 and n_1, and n_1 is correlated with n_0.

$$\hat{s} = s + n - \hat{n}$$

$$\Rightarrow \quad \hat{s}^2 = s^2 + (n - \hat{n})^2 + 2s(n - \hat{n})$$

Taking expectation of both sides and realizing that s is uncorrelated with n_0 and $\hat{n}$,

$$E[\hat{s}^2] = E[s^2] + E[(n - \hat{n})^2] + 2E[s(n - \hat{n})]$$

$$= E[s^2] + E[(n - \hat{n})^2]$$

The signal power $E[s^2]$ will be unaffected as the filter is adjusted to minimize $E[\hat{s}^2]$.

$$\Rightarrow \quad \min E[\hat{s}^2] = E[s^2] + \min E[(n - \hat{n})^2]$$

Thus, when the filter is adjusted to minimize the output noise power $E[\hat{s}^2]$, the output noise power $E[(n - \hat{n})^2]$ is also minimized. Since the signal in the output remains constant, therefore *minimizing the total output power maximizes the output signal-to-noise ratio*.

Since $\qquad (\hat{s} - s) = (n - \hat{n})$

This is equivalent to causing the output $\hat{s}$ to be a *best least squares estimate* of the signal s.

IIA. Effect of uncorrelated noise in primary and reference inputs

As seen in the previous section, the adaptive noise canceller works on the principle of correlation cancellation i.e., the ANC output contains the primary input signals with the component whose correlated estimate is available at the reference input, removed. Thus the ANC is capable of removing only that noise which is correlated with the reference input. Presence of uncorrelated noises in both primary and reference inputs degrades the performance of the ANC. Thus it is important to study the effect of these uncorrelated noises.

Uncorrelated noise in primary input

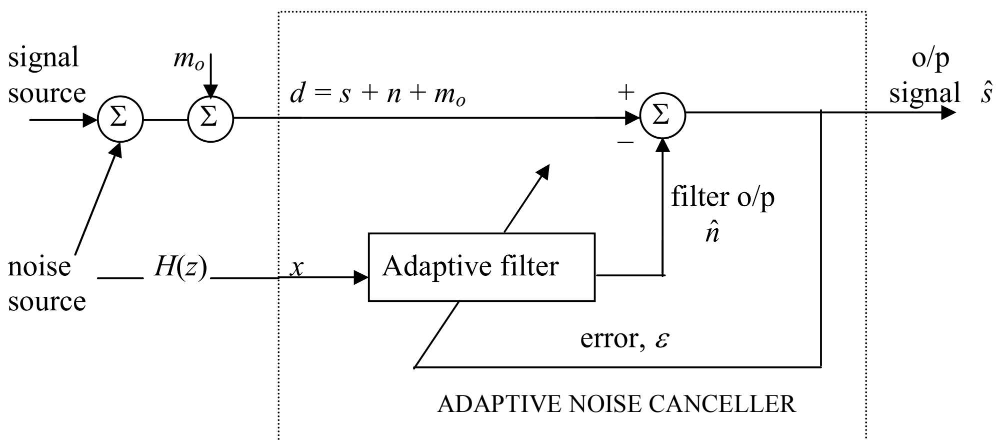


Fig. 2 ANC with uncorrelated noise m_0 in primary input

The figure shows a single channel adaptive noise canceller with an uncorrelated noise m_o present in the primary input. The primary input thus consists of a signal and two noises m_o and n. The reference input consists of $n * h(j)$, where h(j) is the impulse response of the channel whose transfer function is $H(z)$. The noises n and $n * h(j)$ have a common origin and hence are correlated with each other but are uncorrelated with s. The desired response d is thus $s + m_o + n$.

Assuming that the adaptive process has converged to the minimum mean square solution, the adaptive filter is now equivalent to a Wiener filter. The optimal unconstrained transfer function of the adaptive filter is given by (App.I)

$$W^*(z) = \frac{\delta_{xd}(z)}{\delta_{xx}(z)}$$

The spectrum of the filters input $\delta_{xx}(z)$ can be expressed as

$$\delta_{xx}(z) = \delta_{nn}(z)\,|H(z)|^2$$

where $\delta_{nn}(z)$ is the power spectrum of the noise n. The cross power spectrum between filter's input and the desired response depends only on the mutually correlated primary and reference components and is given as

$$\delta_{xd}(z) = \delta_{nn}(z)H(z^{-1})$$

The Wiener function is thus

$$W^{*}(z) = \frac{\delta_{nn}(z)\ H(z^{-1})}{\delta_{nn}(z)\,|\,H(z)\,|^{2}} = \frac{1}{H(z)}$$

Note that $W^{*}(z)$ is independent of the primary signal spectrum $\delta_{ss}(z)$ and the primary uncorrelated noise spectrum $\delta_{momo}(z)$.This result is intuitively satisfying since it equalizes the effect of the channel transfer function $H(z)$ producing an exact estimate of the noise n. Thus the correlated noise n is perfectly nulled at the noise canceller output. However the primary uncorrelated noise n_o remains uncancelled and propagates directly to the output.

Uncorrelated noise in the reference input

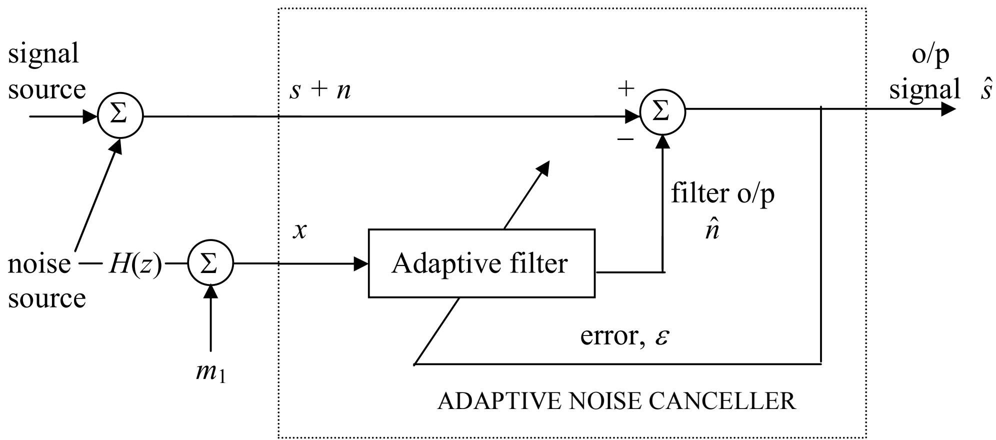


Fig. 3 ANC with uncorrelated noise in reference input

The figure shows an adaptive noise canceller with an uncorrelated noise m_1 in the reference input. The adaptive filter input x is now $m_1 + n * h(\,j\,)$. The filters input spectrum is thus

$$\delta_{xx}(z) = \delta_{m1m1}(z) + \delta_{nn}(z)\,|\,H(z)\,|^{2}$$

The Wiener transfer function now becomes

$$W^{*}(z) = \frac{\delta_{nn}(z)\ H(z^{-1})}{\delta_{m1m1}(z) + \delta_{nn}(z)\,|\,H(z)\,|^{2}}$$

We see that the filter transfer function now cannot equalize the effect of the channel and the filter output is only an approximate estimate of primary noise n.

Effect of primary and reference uncorrelated noises on ANC performance

The performance of the single channel noise canceller in the presence of uncorrelated noises-m_o in primary input and m_1 in reference input simultaneously, can be evaluated in terms of the ratio of the signal to noise density ratio at the output, $\rho_{out}(z)$ to the signal to noise density ratio at the primary input, $\rho_{pri}(z)$. Factoring out the signal power spectrum yields

$$\frac{\rho_{out}(z)}{\rho_{pri}(z)} = \frac{\text{primary noise spectrum}}{\text{output noise spectrum}} = \frac{\delta_{nn}(z) + \delta_{momo}(z)}{\delta_{nout}(z)}$$

The canceller's output noise power spectrum $\delta_{nout}(z)$ is a sum of three components:

1. Due to propagation of m_o directly to the output.
2. Due to propagation of m_1 to the output through the transfer function, $-W^*(z)$.
3. Due to propagation of n to the output through the transfer function, $1 - H(z)W^*(z)$.

The output noise spectrum is thus

$$\delta_{nout}(z) = \delta_{momo}(z) + \delta_{m1m1}(z)\,|W^*(z)|^2 + \delta_{nn}(z)\,|1 - H(z)\,W^*(z)|^2$$

We define the ratios of the spectra of the uncorrelated to the spectra of the correlated noises at the primary and reference as

$$R_{prin}(z) \stackrel{\Delta}{=} \frac{\delta_{momo}(z)}{\delta_{nn}(z)}$$

and

$$R_{refn}(z) \stackrel{\Delta}{=} \frac{\delta_{m1m1}(z)}{\delta_{nn}(z)\,|H(z)|^2}$$

respectively.

The output noise spectrum can be expressed accordingly as

$$\delta_{nout}(z) = \delta_{momo}(z) + \frac{\delta_{m1m1}(z)}{|H(z)|^2\,|R_{refn}(z)+1|^2} + \delta_{nn}(z)\left|1 - \frac{1}{R_{refn}(z)+1}\right|^2$$

$$= \delta_{nn}(z)\,R_{prin}(z) + \delta_{nn}(z)\frac{R_{refn}(z)}{R_{refn}(z)+1}$$

The ratio of output to the primary input noise power spectra can now be written as

$$\frac{\rho_{out}(z)}{\rho_{pri}(z)} = \frac{(R_{prin}(z)+1)(R_{refn}(z)+1)}{R_{prin}(z) + R_{prin}(z)\,R_{refn}(z) + R_{refn}(z)}$$

This expression is a general representation of the ideal noise canceller performance in the presence of correlated and uncorrelated noises. It allows one to estimate the level of noise reduction to be expected with an ideal noise canceling system.

It is apparent from the above equation that the ability of a noise canceling system to reduce noise is limited by the uncorrelated-to-correlated noise density ratios at the primary and reference inputs. The smaller are $R_{prin}(z)$ and $R_{refn}(z)$, the greater will be the ratio of signal-to-noise density ratios at the output and the primary input i.e. $\rho_{out}(z)/\rho_{pri}(z)$ and the more effective the action of the canceller. The desirability of low levels of uncorrelated noise in both primary and reference inputs is thus emphasized.

IIB. Effect of Signal Components in the reference input

Often low-level signal components may be present in the reference input. The adaptive noise canceller is a correlation canceller, as mentioned previously and hence presence of signal components in the reference input will cause some cancellation of the signal also. This also causes degradation of the ANC system performance. Since the reference input is usually obtained from points in the noise filed where the signal strength is small, it becomes essential to investigate whether the signal distortion due to reference signal components can outweigh the improvement in the signal-to noise ratio provided by the ANC.

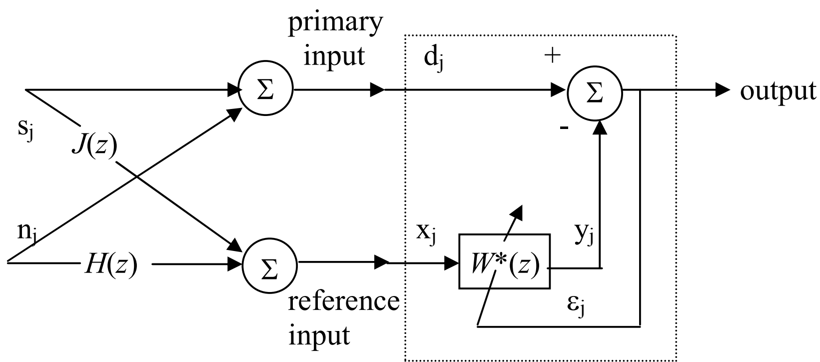


Fig. 4 ANC with signal components in reference input

The figure shows an adaptive noise canceller that contains signal components in the reference input, propagated through a channel with transfer function *J(z)*. If the spectra of the signal and noise are given by $\delta_{ss}(z)$ and $\delta_{nn}(z)$ respectively, the signal-to-noise density ratio at the primary input is

$$\rho_{\text{pri}}(z) \overset{\Delta}{=} \frac{\delta_{ss}(z)}{\delta_{nn}(z)}$$

The spectrum of the signal component in the reference input is

$$\delta_{ssref}(z) = \delta_{ss}(z) \mid J(z) \mid^2$$

and that of the noise component is

$$\delta_{nnref}(z) = \delta_{nn}(z) \mid H(z) \mid^2$$

Therefore, the signal-to-noise density ratio at the reference input is thus

$$\rho_{\text{ref}}(z) = \frac{\delta_{ss}(z) \mid J(z) \mid^2}{\delta_{nn}(z) \mid H(z) \mid^2}$$

The spectrum of the reference input *x* can be written as

$$\delta_{xx}(z) = \delta_{ss}(z) \mid J(z) \mid^2 + \delta_{nn}(z) \mid H(z) \mid^2$$

and the cross spectrum between the reference input *x* and the primary input *d* is given by

$$\delta_{xd}(z) = \delta_{ss}(z)\, J(z^{-1}) + \delta_{nn}(z) H(z^{-1})$$

When the adaptive process has converged, the unconstrained Weiner filter transfer function is thus given by

$$W^*(z) = \frac{\delta_{ss}(z)\, J(z^{-1}) + \delta_{nn}(z) H(z^{-1})}{\delta_{ss}(z) \mid J(z) \mid^2 + \delta_{nn}(z) \mid H(z) \mid^2}$$

We now evaluate expressions for the output signal-to-noise density ratio and the signal distortion and then compare them to see whether the effects of signal distortion are significant enough to render the improvement in *SNR* useless.

Signal distortion $\mathcal{D}(z)$:

When signal components are present in the reference input, some signal distortion will occur and the extent of signal distortion will depend on the amount of signal propagated through the adaptive filter. The transfer function of the propagation path through the filter is

$$- J(z)W^*(z) = - J(z)\frac{\delta_{ss}(z)\, J(z^{-1}) + \delta_{nn}(z)H(z^{-1})}{\delta_{ss}(z)\,|\, J(z)\,|^2 + \delta_{nn}(z)\,|\,H(z)\,|^2}$$

When $|J(z)|$ is small i.e. signal components coupled to the reference input are small, this function can be expressed as

$$- J(z)W^*(z) \cong - \frac{J(z)}{H(z)}$$

The spectrum of the signal component propagated to the noise canceller output through the adaptive filter is thus approximately

$$\delta_{ss}(z)\left|\frac{J(z)}{H(z)}\right|^2$$

Hence, defining the signal distortion $\mathcal{D}(z)$ as the ratio of the spectrum of the signal components in the output propagated through the adaptive filter to the spectrum of signal components in the primary input, we have

$$\mathcal{D}(z) \overset{\Delta}{=} \frac{\delta_{ss}(z)\,|\, J(z)\, W^*(z)\,|^2}{\delta_{ss}(z)}$$

$$= |\, J(z)W^*(z)\,|^2$$

When $J(z)$ is small, this reduces to

$$\mathcal{D}(z) \cong |\, J(z)/ H(z)|^2$$

From the expressions for *SNR* at the primary and reference inputs,

$$\mathcal{D}(z) \cong \frac{\rho_{\mathrm{ref}}(z)}{\rho_{\mathrm{pri}}(z)}$$

This result shows that the relative strengths of signal-to-noise density ratios at the primary and reference inputs govern the amount of signal distortion introduced. Higher the *SNR* at the reference input i.e. the larger the amount of signal components present in the reference, the higher is the distortion. A low distortion results form high signal-to-noise density ratio at the primary input and low signal-to-noise density ratio at the reference input.

Output signal-to-noise density ratio
:

For this case, the signal propagates to the noise canceller output via the transfer function $1-J(z)W^*(z)$, while the noise propagates through the transfer function $1-H(z)W^*(z)$. The spectrum of the signal component in the output is thus

$$\delta_{ssout}(z) = \delta_{ss}(z)\ |1\text{-}J(z)W^*(z)|$$

$$= \delta_{ss}(z)\left|\frac{[H(z) - J(z)]\delta_{nn}(z)H(z^{-1})}{\delta_{ss}(z)|\ J(z)|^2 + \delta_{nn}(z)|H(z)|^2}\right|^2$$

and that of noise component is similarly

$$\delta_{nnout}(z) = \delta_{nn}(z)\ |\ 1\text{-}H(z)W^*(z)|$$

$$= \delta_{nn}(z)\left|\frac{[J(z) - H(z)]\delta_{ss}(z)J(z^{-1})}{\delta_{ss}(z)|\ J(z)|^2 + \delta_{nn}(z)|H(z)|^2}\right|^2$$

The output signal-to-noise density ratio is thus

$$\rho_{\text{out}}(z) = \frac{\delta_{ss}(z)}{\delta_{nn}(z)}\left|\frac{\delta_{nn}(z)\ H(z^{-1})}{\delta_{ss}(z\ )\ J(z^{-1})}\right|^2$$

$$= \frac{\delta_{nn}(z)|H(z)|^2}{\delta_{ss}(z)|\ J(z)|^2}$$

From the expression for signal-to-noise density ratio at reference input, this can be written as

$$\rho_{\text{out}}(z) = \frac{1}{\rho_{\text{ref}}(z)}$$

This shows that the signal-to-noise density ratio at the noise canceller output is simply the reciprocal at all frequencies of the signal-to-noise density ratio at the reference input, i.e. the lower the signal components in the reference, the higher is the signal-to-noise density ratio in the output.

Output noise:

When $|J(z)|$ is small, the expression for output noise spectra reduces to

$$\delta_{nnout}(z) \cong \delta_{nn}(z)\left|\frac{\delta_{ss}(z)\ J(z^{-1})}{\delta_{nn}(z)H(z^{-1})}\right|^2$$

In terms of signal-to-noise density ratios at reference and primary inputs,

$$\delta_{nnout}(z) \cong \delta_{nn}(z)\ |\rho_{ref}(z)|\ |\ \rho_{pri}(z)|$$

The dependence of output noise on these three factors is explained as under:

1. First factor $\delta_{nn}(z)$ implies that the output noise spectrum depends on the input noise spectrum, which is obvious.
2. The second factor implies that, if the signal-to-noise density ratio at the reference input is low, the output noise will be low, i.e. the smaller the signal components in the reference input, the more perfectly the noise will be cancelled.
3. The third factor implies that if the signal-to-noise density ratio in the primary input is low, the filter will be trained most effectively to cancel the noise rather than the signal and consequently output noise will be low.

IIC. Use of ANC without a reference signal

An important and attractive use of ANC is using it without a reference signal. This is possible for the case when one of the signal and noise is narrowband and the other broadband. This is particularly useful for applications where it is difficult or impossible to obtain the reference signal.

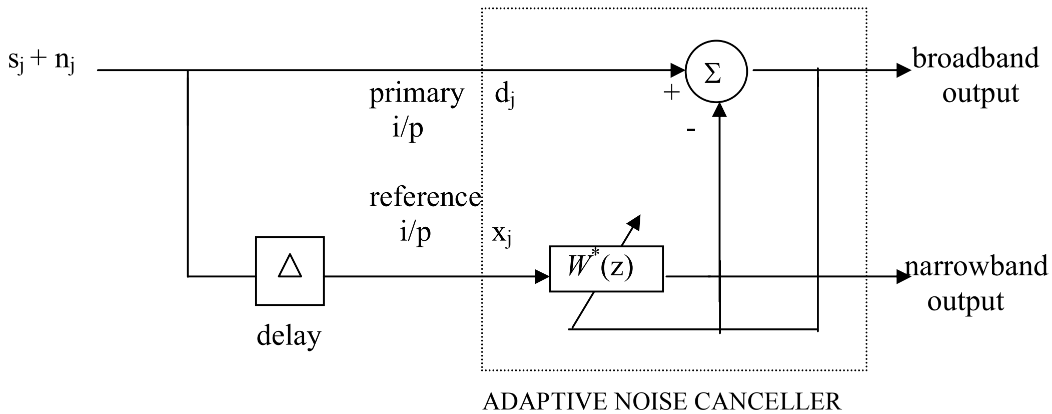


Fig. 5 ANC without reference input

In the case where signal is narrowband and noise is broadband, or signal is broadband and noise is narrowband, a delayed version of the input signal can be used as the reference input. This is because a broadband signal is not correlated to previous sample values unlike a narrowband signal. We only need to insure that the delay introduced should be greater than the decorrelation-time of the broadband signal and less than the decorrelation-time of the narrowband signal.

i.e. $\tau_d \text{ (BB)} < \text{delay} < \tau_d \text{ (NB)}$

This concept is applied to a number of problems -

1. Canceling periodic interference without an external reference source.
2. Adaptive self-tuning filter
3. Adaptive Line Enhancer

These applications are discussed later.

III. Applications

IIIA. Adaptive Noise Canceling applied to sinusoidal interferences

The elimination of a sinusoidal interference corrupting a signal is typically accomplished by explicitly measuring the frequency of the interference and implementing a fixed notch filter tuned to that frequency. A very narrow notch is usually desired in order to filter out the interference without distorting the signal. However, if the interference is not precisely known, and if the notch is very narrow, the center of the notch may not fall exactly over the interference. This may lead to cancellation of some other frequency components of the signal i.e. distorting the signal, while leaving the interference intact. Thus, it may infact lead to an increase in the noise level. Also, there are many applications where the interfering sinusoid drifts slowly in frequency. A fixed notch cannot work here at all unless it is designed wide enough to cover the range of the drift, with the consequent distortion of the signal. In situations such as these, it is often necessary to measure in some way the frequency of the interference, and then implement a notch filter at that frequency. However, estimating the frequency of several sinusoids embedded in the signal can require a great deal of calculation.

When an auxiliary reference input for the interference is available, an alternative technique of eliminating sinusoidal interferences is by an adaptive noise canceller. This reference is adaptively filtered to match the interfering sinusoids as closely as possible, allowing them to be filtered out. The advantages of this type of notch filter are-

1: It makes explicit measurement of the interfering frequency unnecessary.
2: The adaptive filter converges to a dynamic solution in which the time-varying weights of the filter offer a solution to implement a tunable notch filter that helps to track the exact frequency of interference under non-stationary conditions or drifts in frequency.
3: It offers easy control of bandwidth as is shown below.
4: An almost infinite null is achievable at the interfering frequency due to the close and adjustable spacing of the poles and zeros.
5: Elimination of multiple sinusoids is possible by formation of multiple notches with each adaptively tracking the corresponding frequency.

ANC as Single-frequency notch filter:

To understand the operation of an Adaptive Noise Canceller as a Notch filter, we consider the case of a single-frequency noise canceller with two adaptive weights.

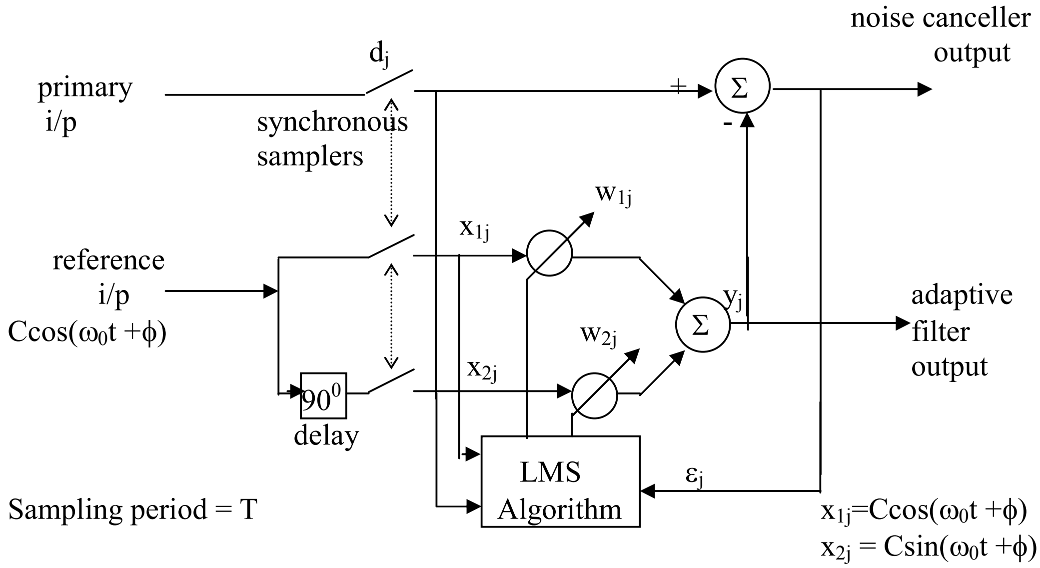


Fig. 6 Single-frequency adaptive noise canceller

The primary input consists of the signal corrupted by a sinusoidal interference of frequency ω_0. The reference input is assumed to be of the form $C\cos(\omega_0 t + \phi)$, where C and ϕ are arbitrary i.e. the reference input contains the same frequency as the interference while its magnitude and phase may be arbitrary. The primary and reference inputs are sampled at the frequency $\Omega = 2\pi/T$ rad/s. The two tap inputs are obtained by sampling the reference input directly and sampling a 90^o shifted version of the reference as shown in the figure above.

To observe the notching operation of the noise canceller, we derive an expression for the transfer function of the system from the primary input to the ANC output. For this purpose, a flow graph representation of the noise canceller system using the LMS algorithm is constructed.

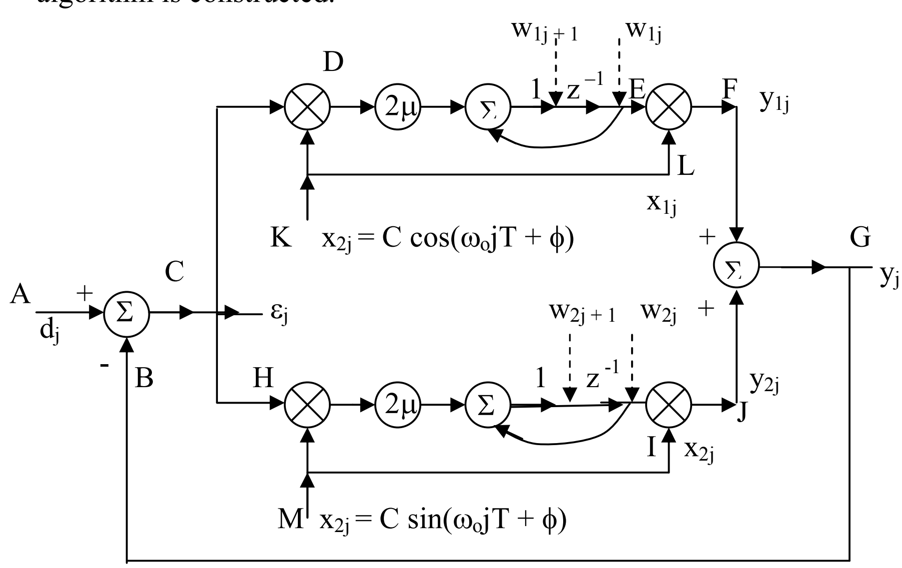


Fig. 7 Flow diagram representation

The LMS weight update equations are given by

$w_{1j+1} = w_{1j} + 2\mu\, \varepsilon_j\, x_{1j}$

$w_{2j+1} = w_{2j} + 2\mu\, \varepsilon_j\, x_{2j}$

The sampled tap-weight inputs are

$x_{1j} = C\, cos(\omega o T j + \phi)$

and

$x_{2j} = C sin(\omega o T j + \phi)$

The first step in the analysis is to obtain the isolated impulse response from the error ε_j, point C, to the filter output, point G, with the feedback loop from point G to point B broken. Let an impulse of amplitude unity be applied at point C at discrete time $j = k$; that is,

$$\varepsilon_j = \delta(j - k)$$

where $\delta(j - k)$ is a unit impulse.

The response at point D is then

$$\varepsilon_j x_{1j} = C\cos(\omega_o kT + \phi) \text{ for } j \neq k \text{ and zero for } j = k$$

which is the input impulse scaled in amplitude by the instantaneous value of x_{1j} at $j = k$. The signal flow path from point D to point E is that of a digital integrator with transfer function $2\mu/(z-1)$ and impulse response $2\mu u(j-1)$,where $u(j)$ is a unit-step function.

Convolving $2\mu u(j-1)$ with $\varepsilon_j x_{1j}$ yields the response at point E:

$$w_{1j} = 2\mu C\cos(\omega_o kT + \phi)$$

where $j \geq k + 1$. When the scaled and delayed step function is multiplied by $x_{1j,}$ the response at point F is obtained:

$$y_{1j} = 2\mu C^2\cos(\omega_o jT + \phi)\cos(\omega_o kT + \phi)$$

where $j \geq k + 1$. The corresponding response at J may be obtained in a similar manner

$$y_{2j} = 2\mu C^2\sin(\omega_o jT + \phi)\sin(\omega_o kT + \phi)$$

Combining these equations ,we obtain the response at filter output G:

$$y_j = 2\mu C^2 u(j - k - 1)\cos\omega_o T(j - k)$$

We now set to derive the linear transfer function for the noise canceller. When the time k is set equal to zero, the unit impulse response of the linear time-invariant signal-flow path from C to G is given as

$$y_j = 2\mu C^2 u(j - 1)\cos\omega_o jT$$

and the transfer funcxtion of this path is

$$G(z) = 2\mu C^2\left(\frac{z(z - \cos\omega_o T)}{z^2 - 2z\cos\omega_o T + 1} - 1\right)$$

This can be expressed in terms of a radian sampling frequency $\Omega = 2\pi/T$ as

$$G(z) = \frac{2\mu C^2\left(z\cos(2\pi\omega_o\Omega^{-1}) - 1\right)}{z^2 - 2z\cos(2\pi\omega_o\Omega^{-1}) + 1}$$

If the feedback loop from G to B is now closed, the transfer function $H(z)$ from the primary input A to the noise canceller output C can be obtained from the feedback formula:

$$H(z) = \frac{1}{(1 - G(z))}$$

$$= \frac{z^2 - 2z\cos(2\pi\omega_o\,\Omega^{-1}) + 1}{z^2 - 2(1 - \mu C^2)z\cos(2\pi\omega_o\Omega^{-1}) + 1 - 2\mu C^2}$$

The above equation shows that the single-frequency noise canceller has the properties of a notch filter at the reference frequency ω_o. The zeros of the transfer function are located in the plane at

$$z = \exp(\pm i2\pi\omega_o\,\Omega^{-1})$$

and are located on the unit circle at angles of $\pm 2\pi\omega_o\,\Omega^{-1}$ rad. The poles are located at

$$z = (1 - \mu C^2)\cos(2\pi\omega_o\,\Omega^{-1}) \pm i[(1 - 2\mu C^2) - (1 - \mu C^2)\cos^2(2\pi\omega_o\Omega^{-1})]^{1/2}$$

The poles are inside the unit circle at a radial distance $(1 - 2\mu C^2)^{1/2}$, approximately equal to $1 - \mu C^2$, from the origin and at angles of

$$\pm\arccos[(1 - \mu C^2)(1 - 2\mu C^2)^{-1/2}\cos(2\pi\omega_o\Omega^{-1})]$$

For slow adaptation, that is, small values of μC^2, these angles depend on the factor

$$\frac{1 - \mu C^2}{(1 - 2\mu C^2)^{1/2}} = \left(\frac{1 - 2\mu C^2 + \mu^2 C^4}{1 - 2\mu C^2}\right)^{1/2}$$

$$\cong 1 - \frac{1}{2}\mu^2 C^4 + \ldots$$

which differs only slightly from a value of one. The result is that, in practical instances, the angles of the poles are almost identical to those of the zeroes.

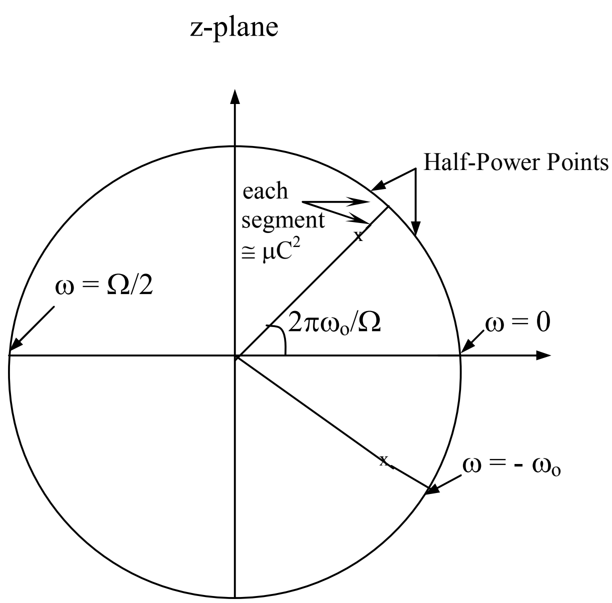


Fig. 8 Location of poles and zeros

Since the zeros lie on the unit circle, the depth of the notch is infinite at the frequency $\omega = \omega_0$. The closeness of the poles to the zeros determines the sharpness of the notch.

Corresponding poles and zeros are separated by a distance approximately equal to μC^2. The arc length along the unit circle (centered at the position of a zero) spanning the distance between half-power points is approximately $2\mu C^2$. This length corresponds to a notch bandwidth of

$$\mathrm{BW} = \mu C^2\Omega/\pi = 2\mu C^2/T$$

The Q of the notch is determined by the ratio of the center frequency to the bandwidth:

$$Q = \omega_0/\mathrm{BW} = \frac{\omega_0\pi}{\mu C^2\Omega}$$

The single-frequency noise canceller is, therefore, equivalent to a stable notch filter when the input is a pure cosine wave. The depth of the null achievable is generally superior to that of a fixed digital or analog filter because the adaptive process maintains the null exactly at the reference frequency.

Multiple-frequency notch filter:

This discussion can be readily extended to the case of a multiple-frequency noise canceller. The formation of multiple notches is achieved by using an adaptive filter with multiple weights. Two weights are required for each sinusoid to achieve the necessary filter gain and phase. Uncorrelated broadband noise superposed on the reference input creates a need for additional weights.

Suppose the reference is a sum of M sinusoids

$$x_j = \sum_{m=1}^{M} C_m \cos(\omega_m jT + \theta_m)$$

At the i^{th} tap-weight input of the transversal tapped delay-line filter of order N,

$$x_{ij} = \sum_{m=1}^{M} C_m \cos(\omega_m [j - i + 1]T + \theta_m) \qquad i=1\ldots N$$

$$= \sum_{m=1}^{M} C_m \cos(\omega_m jT + \theta_{im})$$

where $\theta_{im} = \theta_m - \omega_m [i-1]T$.

The filter output at the i^{th} tap-weight y_{ij} is given by

$$y_{ij} = w_{ij} x_{ij}$$

Proceeding as before, we get a similar equation for w_{ij} as,

$$w_{ij} = 2\mu \sum_{m=1}^{M} C_m \cos(\omega_m kT + \theta_{im}) \qquad \text{where } j \geq k+1$$

$$\therefore \qquad y_{ij} = 2\mu \sum_{m=1}^{M} C_m \cos(\omega_m kT + \theta_{im}) \sum_{n=1}^{M} C_n \cos(\omega_n kT + \theta_{in}) \qquad j \geq k+1$$

The overall filter output is given as

$$y_j = \sum_{i=1}^{N} y_{ij}$$

Substituting and taking the Z-transform of both sides, gives the transfer function $G(\mathrm{z})$ as

$$Y(z) = G(z) = 2\mu \sum_{i=1}^{N} \sum_{n=1}^{M} \frac{C_n z^{-1}[\cos(\omega_n T + \theta_{in}) - \cos\theta_{in} z^{-1}]}{1 - 2z^{-1}\cos\omega_n T + z^{-2}} \sum_{m=1}^{M} C_m \cos\theta_{im}$$

Since the input is a unit impulse and time of applying the pulse *k* is set to zero.

The denominator of *G*(z) is of the form

$$\prod_{n=1}^{M} (1 - 2z^{-1}\cos\omega_n T + z^{-2})$$

Therefore, poles of *G*(z) are located at

$$z = \exp(\pm i2\pi\omega_n T) \qquad n = 1 \ldots M$$

i.e. poles are located at each of the reference frequencies. Since poles of *G*(z) are the zeros of *H*(z), the overall system function has zeros at all reference frequencies i.e. a notch is formed at each of the reference sinusoidal frequencies.

IIIB. Bias or low-frequency drift Canceling using Adaptive Noise Canceller

The use of a bias weight in an adaptive filter to cancel low-frequency drift in the primary input is a special case of notch filtering with the notch at zero frequency. A bias weight is incorporated to cancel dc level or bias and hence is fed with a reference input set to a constant value of one. The value of the weight is updated to match the dc level to be cancelled. Because there is no need to match the phase of the signal, only one weight is needed.

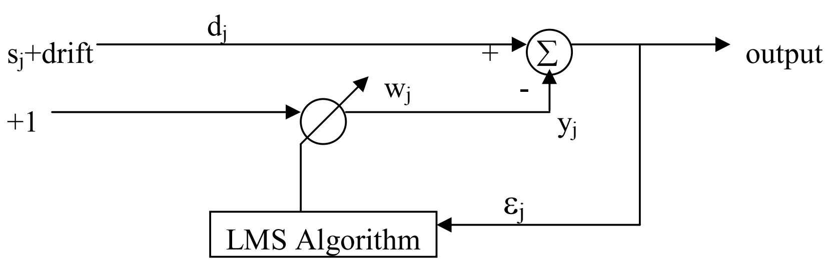


Fig. 9 ANC as bias/low-frequency drift canceller

The transfer function from the primary input to the noise canceller output is now derived. The expression of the output of the adaptive filter y_j is given by

$$y_j = w_j \cdot 1 = w_j$$

The bias weight w is updated according to the LMS update equation

$$w_{j+1} = w_j + 2\mu(\varepsilon_j \cdot 1)$$

$$\Rightarrow \quad y_{j+1} = y_j + 2\mu(d_j - y_j)$$

$$= (1-2\mu)y_j + 2\mu d_j$$

Taking the z-transform of both the sides yields the steady-state solution:

$$Y(z) = \frac{2\mu}{z - (1 - 2\mu)} D(z)$$

Z-transform of the error signal is

$$E(z) = D(z) - Y(z)$$

$$= \frac{z - 1}{z - (1 - 2\mu)} D(z)$$

The transfer function is now

$$H(z) = \frac{E(z)}{D(z)} = \frac{z - 1}{z - (1 - 2\mu)}$$

This shows that the bias-weight filter is a high pass filter with a zero on the unit circle at zero frequency and a pole on the real axis at a distance 2μ to the left of the zero. The smaller the μ, the closer is the location of the pole and the zero, and hence the notch is precisely at zero frequency i.e. only dc level is removed. The single-weight noise canceller acting as a high-pass filter is capable of removing not only a constant bias but also slowly varying drift in the primary input. If the bias level drifts and this drift is slow enough, the bias weight adjusts adaptively to track and cancel the drift. Using a bias weight alongwith the normal weights in an ANC can accomplish bias or drift removal simultaneously with cancellation of periodic or stochastic interference.

IIIC. Canceling Periodic Interference without an External Reference Source

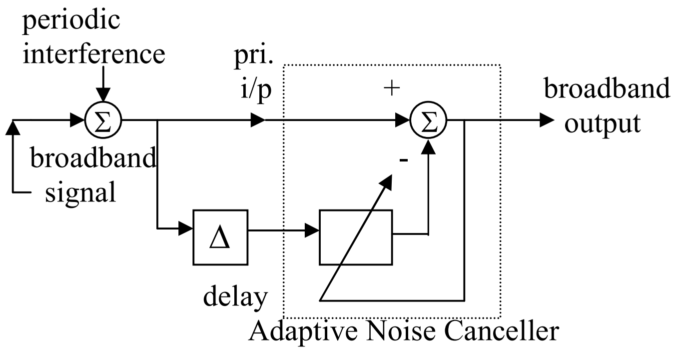


Fig.10 Cancellation of periodic interference

There are a number of circumstances where a broadband signal is corrupted by periodic interference and no external reference input free of the signal is available. This is the case for playback of speech or music in the presence of tape hum or turntable rumble. Adaptive Noise Canceling can be applied to reduce or eliminate such interference by introducing a fixed delay Δ in the reference input drawn directly from the primary input. The delay chosen must be of sufficient length to cause the broadband signal components in the reference input to become decorrelated from those in the primary input. The interference components, because of their periodic nature, will remain correlated with each other.

IIID. Adaptive Self-tuning filter

The noise canceller without a reference input can be used for another important application. In many instances where an input signal consisting of mixed periodic and broadband components is available, the periodic rather than the broadband components are of interest. If the system output is taken from the adaptive filter in an adaptive noise canceller, the result is an adaptive self-tuning filter capable of extracting a periodic signal from broadband noise. The configuration for the adaptive self-tuning filter is shown below:

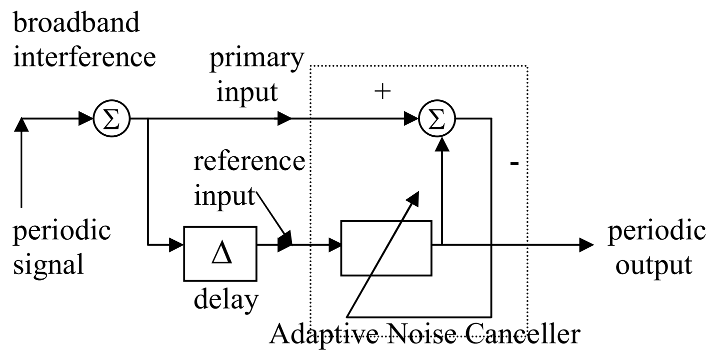


Fig. 11 ANC as self-tuning filter

With sum of sinusoidal signals in broadband stochastic interference, the adaptive filter developed sharp resonance peaks at the frequencies of all the spectral line components of the periodic portion of the primary input.

IIIE. ANC as Adaptive Line Enhancer

The use of ANC as a self-tuning filter suggests its application as an ALE *Adaptive Line Enhancer* for detection of extremely low-level sine waves in noise. The adaptive self-tuning filter, whose capability of separating periodic and stochastic components of a signal was illustrated above (where the components were of a comparable level), is able to serve as an adaptive line enhancer for enhancing the detectability of narrowband signals in the presence of broadband noise.

The configuration of ANC without a reference input, as discussed previously, is used here.

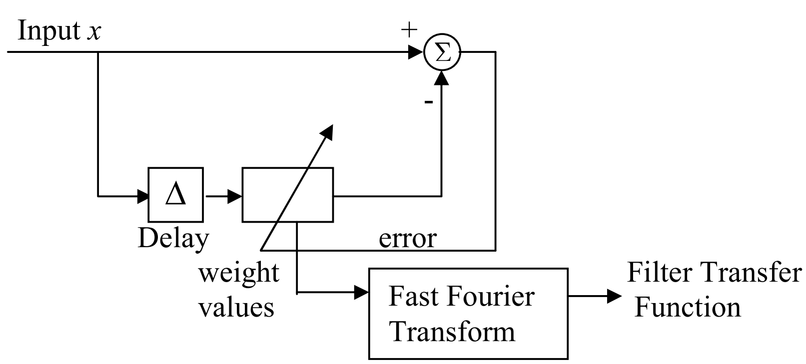


Fig. 12 ANC as Adaptive Line Enhancer

The input consists of signal plus noise. The output is the digital Fourier transform of the filter's impulse response. Detection is accomplished when a spectral peak is evident above the background noise.

Operation of the adaptive line enhancer can be understood intuitively as follows. The delay causes decorrelation between the noise components of the input data in the two channels while introducing a simple phase difference between the sinusoidal components. The adaptive filter responds by forming a transfer function equivalent to that of a narrow-band filter centered at the frequency of the sinusoidal components. The noise component of the delayed input is rejected, while the phase difference of the sinusoidal components is readjusted so that they cancel each other at the summing junction, producing a minimum error signal composed of the noise component of the instantaneous input data alone.

The advantages of adaptive line enhancing with respect to digital Fourier analysis include its effective application over a wide range of input signal and noise parameters with little apriori information. It is capable of estimating and tracking instantaneous frequencies and hence is especially advantageous in applications like where the sine wave is frequency modulated.

We now analyze its steady state behavior with a stationary input consisting of multiple sinusoids in uncorrelated noise. Using the method of undetermined coefficients, the *L*x*L* Weiner-Hopf matrix describing the steady-state impulse response of an *L*-weight ALE with arbitrary delay or “prediction distance”Δ may be transformed into a set of 2*N* coupled linear equations, where *N* is the number of sinusoids in the ALE input. This set of equations, which decouples as the adaptive filter becomes longer, provides a useful description of the interaction between the sinusoids introduced by the finite-length of the filter.

Using the Weiner-Hopf model for the ALE response, *L*x*L* matrix equation can be written in component form as:

$$\sum_{k=0}^{L-1} \phi_{xx}(l-k)w^*(k) = \phi_{xx}(l+\Delta) \qquad 0 \le l \le L\text{-}1$$

where ϕ_{xx} is the autocorrelation of the input
$w^*(k)$ are the optimal weights
When $x(j)$ consists of N sinusoids in white noise,

$$\phi_{xx}(k) = \sigma_0^2\delta(k) + \sum_{n=1}^{N}\sigma_n^2 \cos\omega_n k$$

where $\delta(k)$ is the Kronecker delta function.
σ_0^2 is the white noise power
σ_n^2 is the power in the n^{th} sinusoid
ω_n represents the frequencies of the sinusoids

To avoid the computational complexity involved in taking matrix inverse, we use the method of undetermined coefficients. Since the ALE adaptive filter is expected to respond by forming peaks at the input frequencies, we assume the following solution for $w^*(k)$

$$w^*(k) = \sum_{n=1}^{2N} A_n e^{j\omega_n k}$$

where for notational convenience, ω_{n+N} is defined as $-\omega_n$ (n=1…N); the ω_{n+N} are thus the negative frequency components of the input sinusoids. Substituting with $\phi_{xx}(l)$, and equating coefficients of $\exp(j\omega_r l)$, leads to the following 2N equations in the 2N coefficients $A_1, \ldots, A_{2N}$.

$$A_r + \sum_{\substack{n=1 \\ n \neq r}}^{2N} \gamma_{rn} A_n = \frac{e^{j\omega_r \Delta}}{L + 2\sigma_0^2 / \sigma_r^2} \qquad r = 1,2,\ldots 2N$$

where σ_{n+N}^2 is defined as σ_n^2 and γ_{rn} is given by

$$\gamma_{rn} = \frac{1}{L + 2\sigma_0^2 / \sigma_r^2} \frac{1 - e^{j(\omega_n - \omega_r)L}}{1 - e^{j(\omega_n - \omega_r)}}$$

The solution for the A_n completely determine w*(k).

When the input to the ALE consists of N sinusoids and additive white noise, the mean steady-state impulse response of the ALE can be expressed as a weighted sum of positive and negative frequency components of the input sinusoids. It is seen that the coefficients that couple A_n together are proportional to the *L*-point Fourier transform of $\exp(j\omega_n k)$ evaluated at ω_r. From the form of γ_{rn}, $A_{n+N} = \overline{A}_n$. This shows that w*(k) are real. Since the form of γ_{rn} is of sinc-type, when L becomes large or when ω_n-ω_r is some integral multiple of $2\pi/L$, γ_{rn} can be neglected. Further as L becomes large, the ratio of the main lobe (at $\omega_n - \omega_r = 0$) to the sidelobe peaks is given approximately by $1/(p+1/2)\pi$. Even if ω_n is within the first few peaks of ω_r, the associated γ_{rn} can be neglected.

As $\gamma_{rn} \rightarrow 0$ for all n and r(i.e. as L becomes large), the A_n uncouple and are given to a good approximation by

$$A_n = \frac{e^{j\omega_n \Delta}}{L + 2\sigma_0^2 / \sigma_r^2} \qquad n = 1\ldots 2N$$

Therefore as $\gamma_{rn} \rightarrow 0$, the ALE for N sinusoids will adapt to a linear superposition of N independent ALE's, each adapted to a single sinusoid in white noise.

The frequency response of the steady-state ALE can now simply be expressed as

$$\begin{aligned} H^*(\omega) &= \sum_{k=0}^{L-1} \omega^*(k) e^{-j\omega(k+1)} \\ &= \sum_{n=1}^{2N} A_n e^{-j\omega} \frac{1 - e^{j(\omega_n - \omega)L}}{1 - e^{j(\omega_n - \omega)}} \\ &\cong \sum_{n=1}^{N} \frac{e^{-j(\omega_n \Delta + \omega)}}{L + 2\frac{\sigma_o^2}{\sigma_n^2}} \frac{1 - e^{-j(\omega_n + \omega)L}}{1 - e^{-j(\omega_n + \omega)}} + \sum_{n=1}^{N} \frac{e^{j(\omega_n \Delta - \omega)}}{L + 2\frac{\sigma_o^2}{\sigma_n^2}} \frac{1 - e^{j(\omega_n - \omega)L}}{1 - e^{j(\omega_n - \omega)}} \end{aligned}$$

The above equation corresponds to sum of bandpass filters (centers at $\pm\omega_n$), each having a peak value given by

$$(L/2)SNR_n/((L/2)SNR_n + 1)$$

where $SNR_n = \sigma_n^2/\sigma_o^2$. As $L \rightarrow \infty$, all of the peak values and the ALE becomes a super position of perfectly resolved bandpass filters, each with unit gain at its center frequency. Caution must be exercised in choosing L because as L is increased, the weight-vector noise also increases.

The ALE thus provides an alternative to spectral analytic techniques and has the advantage of not requiring a priori information and also adaptively tracking the sinusoidal frequencies.

IIIF. Canceling Antenna sidelobe Interference

Strong unwanted signals incident on the sidelobes of an antenna array can severely interfere with the reception of weaker signals in the main beam. The conventional method of reducing such interference, adaptive beamforming, is often complicated and expensive to implement. When the number of spatially discrete interference sources is small, adaptive noise canceling can provide a simpler and less expensive method of dealing with this problem.

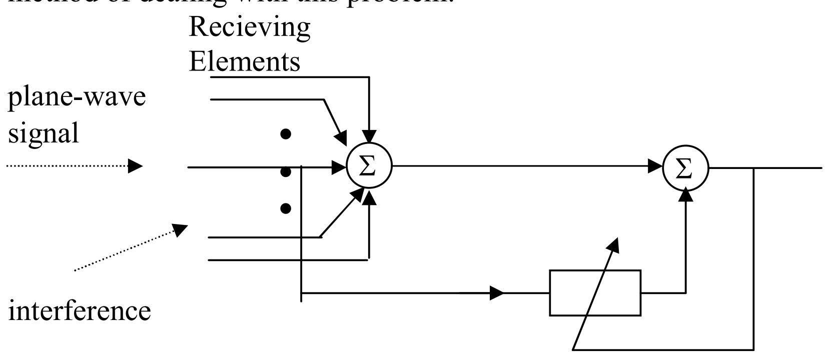


Fig. 13 ANC applied to a receiving array

The reference is obtained by steering the reference sensor in the direction of the interference.

Conclusion

Adaptive Noise Cancellation is an alternative way of canceling noise present in a corrupted signal. The principal advantage of the method are its adaptive capability, its low output noise, and its low signal distortion. The adaptive capability allows the processing of inputs whose properties are unknown and in some cases non-stationary. Output noise and signal distortion are generally lower than can be achieved with conventional optimal filter configurations.

This Project indicates the wide range of applications in which Adaptive Noise Canceling can be used. The simulation results verify the advantages of adaptive noise cancellation. In each instance canceling was accomplished with little signal distortion even though the frequencies of the signal and interference overlapped. Thus it establishes the usefulness of adaptive noise cancellation techniques and its diverse applications.

Scope for further work:

In this project, only the Least-Mean-Squares Algorithm has been used. Other adaptive algorithms can be studied and their suitability for application to Adaptive Noise Cancellation compared. Other algorithms that can be used include Recursive Least Squares, Normalised LMS, Variable Step-size algorithm etc.

Moreover, this project does not consider the effect of finite-length filters and the causal approximation. The effects due to these practical constraints can be studied.

Appendix

I. Weiner Filter Theory

Wiener proposed a solution to the continuous-time linear filtering problem and derived the Wiener-Hopf integral equation. The discrete-time equivalent of this integral equation is called the 'Normal equation'. Solution of these two equations defines the Wiener filter. We concentrate here on the discrete-time case only.

Formulation of Wiener filters in discrete-time case for the general case of complex-valued time-series:

This discussion is limited to- 1) Filter impulse response of finite duration.
2) A single input and single output filter.

Statement of the Optimum Filtering Problem:

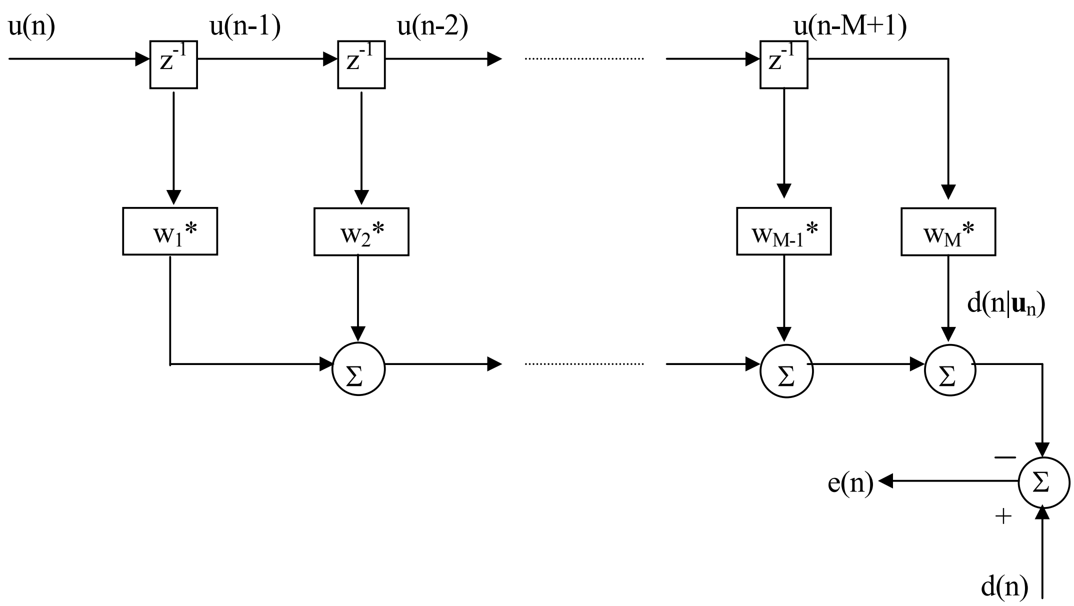


Fig.A Linear Transversal Filter

Order of filter = no. of delay elements in the filter = M-1

Impulse response of the transversal filter $\{h_k\}=\{w_k^*\}$ $k = 1,2\ldots M$

Filter output is related to filter input and impulse response of the filter by the *convolution sum*

$$d(n|\mathbf{u}_n) = \sum_{k=1}^{M} w_k^* \, u(n-k+1)$$

Signal output of the filter at time n = $d(n|\mathbf{u}_n)$, estimate of desired response d(n) assuming knowledge of the tap inputs.

The estimation problem is solved by designing filter so that difference between d(n) & $d(n|\mathbf{u}_n)$ is made as "small" as possible in a statistical sense.

estimation error = $e(n) = d(n) - d(n|\mathbf{u}_n)$

In Wiener theory *"minimum mean-squared error criterion"* is used to optimize the filter. Specifically, tap weights are chosen so as to minimize the "index of performance" $J(\mathbf{w})$, the mean-squared error MSE.

$$J(\mathbf{w}) = E[e(n)e^*(n)]$$

By minimizing $J(\mathbf{w})$, we obtain the best or *optimum* linear filter in the *minimum mean-square sense.*

<u>Error Performance Surface</u>

Let: Mx1 tap-weight vector $\mathbf{w}^T = [w_1, w_2, \ldots, w_M]$

Mx1 input vector $\mathbf{u}^T(n) = [u(n), u(n-1), \ldots, u(n-M+1)]$

Then, filter output $d(n|\mathbf{u}_n) = \mathbf{w}^H\mathbf{u}(n)$

where ^{H} denotes the Hermitian Transpose

or $d(n|\mathbf{u}_n) = \mathbf{u}^H(n)\mathbf{w}$

=> Estimation error between desired response d(n) and filter output $d(n|\mathbf{u}_n)$

$$e(n) = d(n) - \mathbf{w}^H\mathbf{u}(n)$$

$$\text{or} \quad e^*(n) = d^*(n) - \mathbf{u}^H(n)\mathbf{w}$$

Hence,

Mean squared error, $J(\mathbf{w}) = E[e(n)e^*(n)]$

$$= E[(d(n) - \mathbf{w}^H\mathbf{u}(n))\,(d^*(n) - \mathbf{u}^H(n)\mathbf{w})]$$

Expanding and recognizing tap-weight vector $\mathbf{w}$ is constant,

$$J(\mathbf{w}) = E[d(n)d^*(n) - \mathbf{w}^H E[\mathbf{u}(n)d^*(n)] - E[d(n)\mathbf{u}^H(n)]\mathbf{w} + \mathbf{w}^H E[\mathbf{u}(n)\mathbf{u}^H(n)]\mathbf{w}$$

We make the following assumptions

- discrete time stochastic process represented by tap inputs u(n), u(n-1)… is weakly stationary.
- Mean value of the process is zero.
- Tap-input vector $\mathbf{u}(n)$ and desired response d(n) are *jointly stationary.*

We can now identify,

1. $E[d(n)d^*(n)] = \sigma_d^2$ variance of desired response assuming d(n) has zero mean.
2. $E[\mathbf{u}(n)d^*(n)] = \mathbf{p}$ Mx1 cross-correlation vector between tap input vector & desired response.

   i.e. $\mathbf{p}^T = [p(0), p(-1), \ldots, p(1-M)]$

   where $p(1-k) = E[u(n-k+1)d^*(n)]$ $k=1,2,\ldots M$
3. $E[d(n)\mathbf{u}^H(n)] = \mathbf{p}^H$
4. $E[\mathbf{u}(n)\mathbf{u}^H(n)] = \mathbf{R}$ MxM correlation matrix of tap input vector

$$\Rightarrow \quad J(\mathbf{w}) = \sigma_d^2 - \mathbf{p}^H\mathbf{w} - \mathbf{w}^H\mathbf{p} + \mathbf{w}^H\mathbf{R}\mathbf{w}$$

When u(n) and d(n) are jointly stationary, mean-squared error $J(\mathbf{w})$ is precisely a second-order function of the tap-weight vector $\mathbf{w}$. Therefore, dependence of $J(\mathbf{w})$ on elements of $\mathbf{w}$ i.e. tap weights $w_1, w_2, \ldots w_M$ is equivalent to a bowl-shaped surface with a unique minimum. This is the *error-performance surface* of the transversal filter.

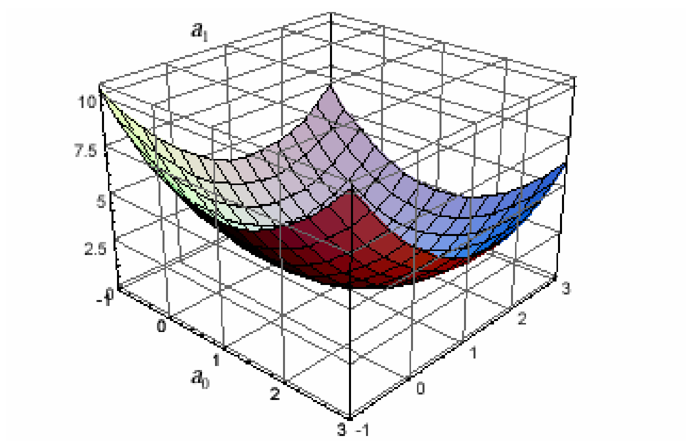


Fig. B Error surface for M=2 and filter weights $w_1=a_0$ and $w_2=a_1$

<u>Optimum Solution</u>:

Now, requirement is to design the filter so that it operates at this 'bottom' or 'minimum' point. At this point,

J(**w**) has a minimum value, J_{min}

w has an optimum value, $\mathbf{w_0}$

The resultant transversal filter is said to be 'optimum in the mean-squared sense'.

To get $\mathbf{w_0}$, we apply the condition for minimum J(**w**) i.e.

$$\frac{d\,J(\mathbf{w})}{d\mathbf{w}} = 0$$

We get:

$$\frac{d\,\sigma_d^2}{d\mathbf{w}} = 0 \qquad \text{(for a weakly stationary process, variance is constant)}$$

$$\frac{d}{d\mathbf{w}}(\mathbf{p}^H\mathbf{w}) = 0$$

$$\frac{d}{d\mathbf{w}}(\mathbf{w}^H\mathbf{p}) = 2\mathbf{p}$$

$$\frac{d}{d\mathbf{w}}(\mathbf{w}^H\mathbf{R}\mathbf{w}) = 2\mathbf{R}\mathbf{w}$$

Hence, *gradient vector* ∇ = derivative of mean-squared error J(**w**) wrt tap-weight **w**

$$\nabla = \frac{d}{d\mathbf{w}} J(\mathbf{w})$$

$$= -2\mathbf{p} + 2\mathbf{R}\mathbf{w} \qquad \ldots(1)$$

At the bottom of the error-performance surface i.e. for the optimal case, this gradient is equal to 0 or as earlier mentioned, J(**w**) is a minimum.

$$\Rightarrow \quad \mathbf{R}\mathbf{w_0} = \mathbf{p} \qquad \ldots(2)$$

This is the discrete-form of the Weiner-Hopf equation, also called *normal equation.*

Its solution gives:

$$\mathbf{w_0} = \mathbf{R}^{-1}\mathbf{p}$$

Computation of optimum tap-weight vector $\mathbf{w_0}$ requires knowledge of two quantities:

1. Correlation matrix **R** of the tap-input vector **u**(n).
2. Cross-correlation vector **p** between tap-input **u**(n) and desired response d(n).

The curve obtained by plotting the mean-squared error versus the number of iterations, n is called the *learning curve.*

II. Adaptive Filters

Adaptive filters are digital filters with an impulse response, or transfer-function, that can be adjusted or changed over time to match desired system characteristics.

Unlike fixed filters, which have a fixed impulse response, adaptive filters do not require complete a priori knowledge of the statistics of the signals to be filtered. Adaptive filters require little or no a priori knowledge and moreover, have the capability of adaptively tracking the signal under non-stationary circumstances.

For an adaptive filter operating in a stationary environment, the error-performance surface has a constant shape as well as orientation. When, however, the adaptive filter operates in a non-stationary environment, the bottom of the error-performance surface continually moves, while the orientation and curvature of the surface may be changing too. Therefore, when the inputs are non-stationary, the adaptive filter has the task of not only seeking the bottom of the error performance surface, but also continually *tracking* it.

III. Steepest Descent Algorithm

An adaptive filter is required to find a solution for its tap-weight vector that satisfies the normal equation. Solving this equation by analytical means presents serious computational difficulties, especially when the filter contains a large number of tap weights and when the data rate is high. An alternative procedure is to use the *method of steepest descent*, which is one of the oldest methods of optimization.

1. Initial values of **w**(0) are chosen arbitrarily i.e. initial guess as to where the minimum point of the error-performance surface may be located. Typically **w**(0) = null vector.
2. Using this, we compute the *gradient vector*, defined as the gradient of mean-squared error J(n) wrt **w**(n) at time n (n^{th} iteration).
3. We compute the next guess at the tap-weight vector by making a change in the initial or present guess in a direction opposite to that of the gradient vector.
4. Go back to step 2 and repeat the process.

$$\Rightarrow \quad \mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{2}\mu\,[-\nabla(n)] \qquad \mu = \text{positive real-valued constant}$$

From equation (1),

$$\nabla(n) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n)$$

For the application of the steepest-descent algorithm, we assume that the correlation matrix **R** and cross-correlation matrix **p** are known.

$$\Rightarrow \quad \mathbf{w}(n+1) = \mathbf{w}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)] \qquad n = 0, 1, 2,\ldots \qquad (3)$$

We observe that the parameter μ controls the size of the incremental correction applied to the tap-weight vector as we proceed from one iteration to the next. Therefore, μ is referred to as the *step-size parameter* or *weighting constant*.

The equation (3) describes the mathematical formulation of the steepest-descent algorithm or also referred to as the *deterministic gradient algorithm*.

Feedback-Model of the steepest-descent algorithm:

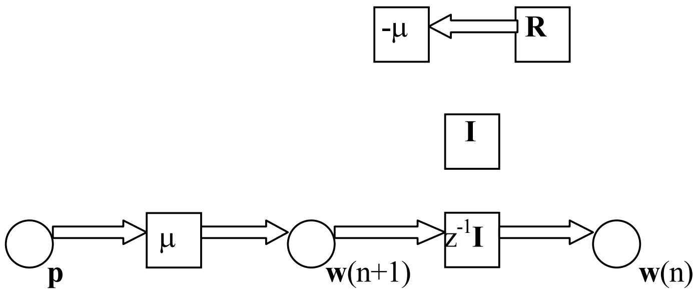


Fig. C Signal-flow graph representation of the steepest-descent algorithm.

Stability of the steepest-descent algorithm:

Since the steepest-descent algorithm involves the presence of *feedback*, the algorithm is subject to the possibility of its becoming *unstable*. From the feedback model, we observe that the *stability performance* of the algorithm is determined by two factors:

1. the step-size parameter μ
2. the correlation matrix $\mathbf{R}$ of the tap-input vector $\mathbf{u}(n)$

as these two parameters completely control the transfer function of the *feedback loop*.

Condition for stability:

weight-error vector, $\mathbf{c}(n) = \mathbf{w}(n) - \mathbf{w_0}$

where $\mathbf{w_0}$ is the optimum value of the tap-weight vector as defined by the normal equation. Therefore, eliminating cross-correlation vector $\mathbf{p}$ in equation (3) and rewriting the result in terms of the weight-error vector,

$$\mathbf{c}(n+1) = (\mathbf{I} - \mu\mathbf{R})\,\mathbf{c}(n)$$

Using unitary-similarity transformation, we may express the correlation matrix $\mathbf{R}$ as

$$\mathbf{R} = \mathbf{Q}\Lambda\mathbf{Q}^H$$

(refer Appendix B)

$$\Rightarrow \quad \mathbf{c}(n+1) = (\mathbf{I} - \mu\,\mathbf{Q}\Lambda\mathbf{Q}^H)\,\mathbf{c}(n)$$

Premultiplying both sides of the equation by $\mathbf{Q}^H$ and using the property of unitary matrix $\mathbf{Q}$ that $\mathbf{Q}^H = \mathbf{Q}^{-1}$, we get

$$\mathbf{Q}^H\,\mathbf{c}(n+1) = (\mathbf{I} - \mu\,\Lambda)\,\mathbf{Q}^H\,\mathbf{c}(n)$$

We now define a new set of coordinates as follows:

$$\mathbf{v}(n) = \mathbf{Q}^H\,\mathbf{c}(n)$$
$$= \mathbf{Q}^H\,[\mathbf{w}(n) - \mathbf{w_0}]$$

Accordingly, we may write

$$\mathbf{v}(n+1) = (\mathbf{I} - \mu\,\Lambda)\mathbf{v}(n)$$

The initial value of $\mathbf{v}(n)$ equals

$$\mathbf{v}(0) = \mathbf{Q}^H\,[\mathbf{w}(0) - \mathbf{w_0}]$$

Assuming that the initial tap-weight vector $\mathbf{w}(0)$ is zero, this reduces to

$$\mathbf{v}(0) = -\mathbf{Q}^H\,\mathbf{w_0}$$

For the k^{th} natural mode of the steepest-descent algorithm, we thus have

$$v_k(n+1) = (1 - \mu\lambda_k)\, v_k(n) \qquad k=1, 2, \ldots, M \qquad \ldots(4)$$

where λ_k is the k^{th} eigenvalue of the correlation matrix **R**. This equation is represented by the following scalar-valued feedback model:

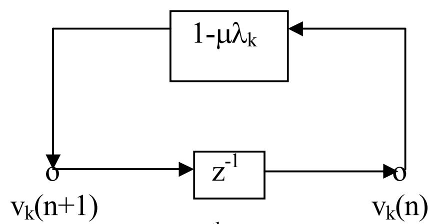


Fig. D Signal-flow graph of the k^{th} mode of the steepest-descent algorithm

Equation (4) is a homogeneous *difference equation of the first order*. Assuming that $v_k(n)$ has the initial value $v_k(0)$, we readily obtain the solution

$$v_k(n) = (1 - \mu\lambda_k)^n\, v_k(0), \qquad k=1, 2, \ldots, M \qquad \ldots(5)$$

Since all eigenvalues of the correlation matrix **R** are positive and real, the response $v_k(n)$ will exhibit no oscillations. For *stability* or *convergence* of the steepest-descent algorithm, the magnitude of the geometric ratio of the above *geometric series* must be less than 1 for all k.

$$\Rightarrow \qquad -1 < 1-\mu\lambda_k < 1 \qquad \forall k$$

Provided this condition is satisfied, as the number of iterations, n, approaches infinity, all natural modes of the steepest-descent algorithm die out, irrespective of the initial conditions. This is equivalent to saying that the tap-weight vector **w**(n) approaches the optimum solution $\mathbf{w_0}$ as n approaches infinity.

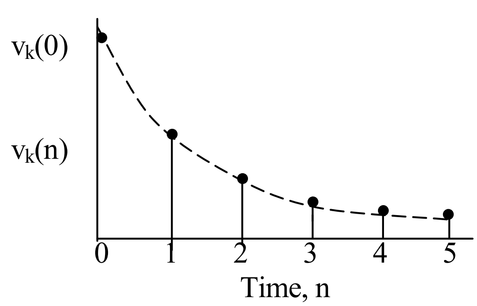
Therefore, the necessary and sufficient condition for the convergence or stability of the steepest-descent algorithm is that the step-size parameter μ satisfy the following condition:

$$0 < \mu < 2/\lambda_{max}$$

where λ_{max} is the largest eigenvalue of the correlation matrix **R**.

Convergence rate of the steepest-descent algorithm:

Assuming that the magnitude of $1-\mu\lambda k$ is less than 1 i.e. stability criterion is met, from equation (5) the variation of the k^{th} natural mode of the steepest-descent algorithm with time is as shown below:



We see that an exponential envelope of *time constant* τ_k can be fitted to the geometric series by assuming the unit of time to be the duration of one iteration cycle and by choosing the time constant τ_k such that

$$1 - \mu\lambda_k = \exp(-1/\tau_k)$$

$$\Rightarrow \qquad \tau_k = \frac{-1}{\ln(1 - \mu\lambda_k)} \qquad \ldots(6)$$

The time constant τ_k defines the time required for the amplitude of the k^{th} natural mode $v_k(n)$ to decay to 1/e of its initial value $v_k(0)$.

For slow-adaptation i.e. small μ,

$$\tau_k \approx \frac{1}{\mu\lambda_k} \qquad \mu << 1$$

We may now formulate the transient behavior of the original tap-weight vector $\mathbf{w}(n)$. We know,

$$\mathbf{v}(n) = \mathbf{Q}^H [\mathbf{w}(n) - \mathbf{w_0}]$$

Pre-multiplying both sides by $\mathbf{Q}$, and using the fact that $\mathbf{QQ}^H=\mathbf{I}$, we get:

$$\mathbf{w}(n) = \mathbf{w_0} + \mathbf{Qv}(n)$$

$$= \mathbf{w_0} + [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_M] \begin{bmatrix} v_1(n) \\ v_2(n) \\ \vdots \\ v_M(n) \end{bmatrix}$$

$$= \mathbf{w_0} + \sum_{k=1}^{M} \mathbf{q}_k v_k(n)$$

Substituting equation (5), we find that the transient behavior of the *i*th tap weight is described by

$$w_i(n) = w_{0i} + \sum_{k=1}^{M} q_{ki} v_k(0) (1-\mu\lambda_k)^n \qquad i=1, 2,\ldots, M \qquad \ldots(7)$$

where w_{0i} is the optimum value of the *i*th tap weight, and q_{ki} is the *i*th element of the k^{th} eigenvector $\mathbf{q}_k$.

Equation (7) shows that each tap weight in the steepest-descent algorithm converges as the weighted sum of exponentials of the form $(1 - \mu\lambda_k)^n$. The time τ_k required for each term to reach 1/e of its initial value is given by equation (6). However, the *overall time constant*, τ_a, defined as the time required for the summation term in (7) to decay to 1/e of its initial value, cannot be expressed in a similar simple form.

Nevertheless, the *rate of convergence* is bounded on the lower side by the term that decays slowest i.e. $(1-\mu\lambda_{min})$ and on the upper side by the fastest decaying term i.e. $(1-\mu\lambda_{max})$. Accordingly, the overall time-constant τ_a for any tap weight of the steepest-descent algorithm is bounded as follows:

$$\frac{-1}{\ln(1-\mu\lambda_{max})} \leq \tau_a \leq \frac{-1}{\ln(1-\mu\lambda_{min})}$$

If we choose μ to be half the upper bound i.e. $1/\lambda_{max}$, then rate of convergence is limited by the term

$$(1-\mu\lambda_{min}) = (1- \lambda_{min}/\lambda_{max})$$

$\Rightarrow$ for *fast* convergence, we want $\lambda_{min}/\lambda_{max}$ close to one, that is a small eigenvalue spread.

for *slow* convergence, $\lambda_{min}/\lambda_{max}$ will be small, thus the eigenvalue spread will be large.

A large eigenvalue spread indicates that input is highly correlated. When the eigenvalues of the correlation matrix **R** are widely spread, the settling time of the steepest-descent algorithm is limited by the smallest eigenvalue or the slowest mode.

IV. Least-Mean-Squares LMS Algorithm

If it were possible to make exact measurements of the gradient vector at each iteration, and if the step-size parameter μ is suitably chosen, then the tap-weight vector computed by using the method of steepest-descent would indeed converge to the optimum Wiener solution. In reality, however, exact measurements of the gradient vector are not possible, and it must be estimated from the available data. In other words, the tap-weight vector is updated in accordance with an algorithm that adapts to the incoming data.

One such algorithm is the *least mean square* (LMS) algorithm. A significant feature of LMS is its simplicity; it does not require measurements of the pertinent correlation functions, nor does it require matrix inversion.
We have earlier found that

$$\text{gradient vector, } \nabla(n) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n)$$

To estimate this, we estimate the correlation matrix **R** and cross-correlation matrix **p** by *instantaneous estimates* i.e.

$$\mathbf{R}'(n) = \mathbf{u}(n)\mathbf{u}^H(n)$$

$$\mathbf{p}'(n) = \mathbf{u}(n)\, d^*(n)$$

Correspondingly, the instantaneous estimate of the gradient-vector is

$$\nabla'(n) = -2\,\mathbf{u}(n)\, d^*(n) + 2\,\mathbf{u}(n)\mathbf{u}^H(n)\mathbf{w}(n)$$

The estimate is *unbiased* in that its expected value equals the true value of the gradient vector. Substituting this estimate in the steepest-descent algorithm, equation (3), we get a new recursive relation for updating the tap-weight vector:

$$\mathbf{w}'(n+1) = \mathbf{w}'(n) + \mu\mathbf{u}(n)[d^*(n) - \mathbf{u}^H(n)\mathbf{w}'(n)] \qquad \ldots(8)$$

Equivalently the LMS update equation can be written in the form of a pair of relations:

$$e(n) = d(n) - \mathbf{u}^H(n)\mathbf{w}'(n) \qquad \ldots(9)$$

$$\mathbf{w}'(n+1) = \mathbf{w}'(n) + \mu\mathbf{u}(n)e^*(n) \qquad \ldots(10)$$

The first equation defines the estimation error e(n), the computation of which is based on the *current estimate* of the tap-weight vector $\mathbf{w}'(n)$. The term $\mu\mathbf{u}(n)e^*(n)$ in the second equation represents the *correction* that is applied to the current estimate of the tap-weight vector. The iterative procedure is started with the initial guess $\mathbf{w}'(0)$, a convenient choice being the null vector; $\mathbf{w}'(0) = \mathbf{0}$.

The algorithm described by the equation (8) or equivalently by the equations (9) and (10), is the complex form of the adaptive *least mean square* (LMS) *algorithm*. It is also known as the *stochastic-gradient algorithm*.

The instantaneous estimates of **R** and **p** have relatively large variances. It may therefore seem that the LMS algorithm is incapable of good performance. However, the LMS algorithm, being recursive in nature, effectively averages these estimates, in some sense, during the course of adaptation.

Ideally, the minimum mean-squared error J_{min} is realized when the coefficient vector **w**(n) of the transversal filter approaches the optimum value $\mathbf{w_0}$. The steepest descent-algorithm does realize this idealized condition as the number of iterations, n approaches infinity, because it uses *exact* measurements of the gradient vector at each iteration. On the other hand, LMS relies on a *noisy* estimate of the gradient vector, with the result that the tap-weight vector only approaches the optimum value after a large number of iterations and then executes small fluctuations about $\mathbf{w_0}$. Consequently, use of LMS results in a mean-squared error $J(\infty)$ after a large no. of iterations.

Excess mean-squared error: is defined as the amount by which the actual value of $J(\infty)$ is greater than J_{min}.

Misadjustment: The misadjustment *M* is defined as the dimensionless ratio of the steady-state value of the average excess mean-squared error to the minimum mean-squared error. It can be shown that

$$M = \frac{\mu \sum_{\forall i} \lambda_i}{2 - \mu \sum_{\forall i} \lambda_i} \qquad i=1, 2,\dots, M$$

Now, for convergence

$$\mu < 2/\lambda_{max}$$

If μ is small enough so that

$$\mu < 2/\sum\lambda_i$$

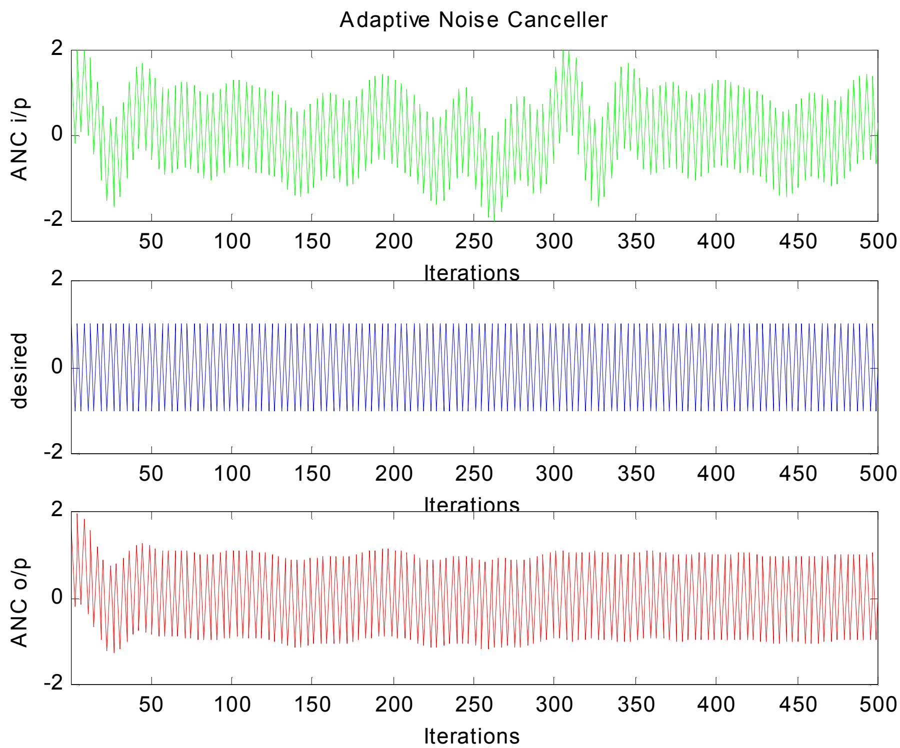
then the misadjustment *M* varies linearly with μ.

References

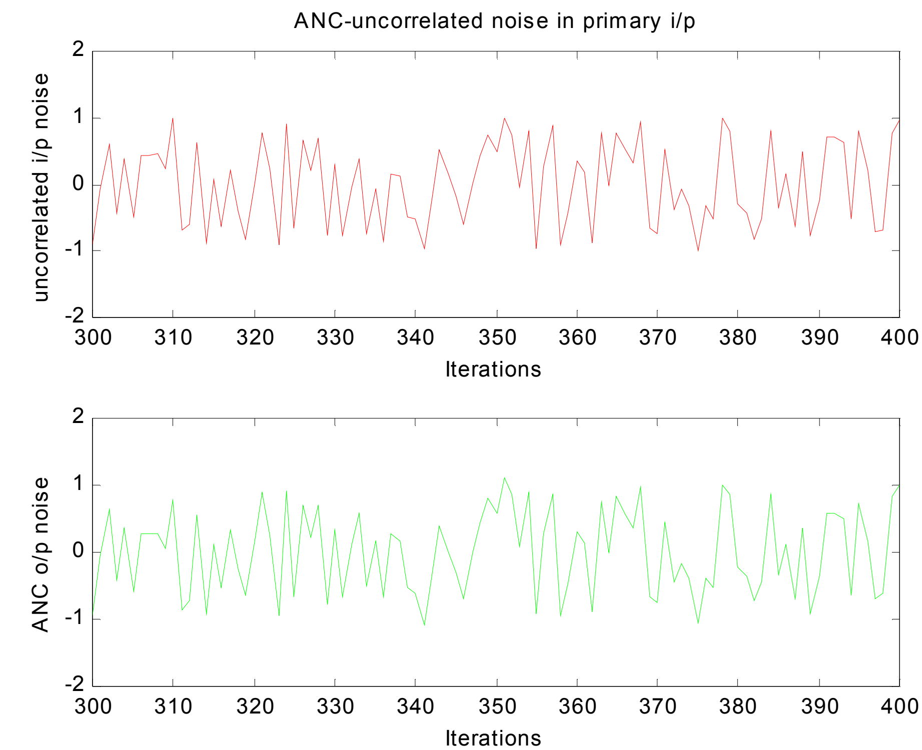
(1) B. Widrow, et al., "Adaptive Noise Cancelling: Principles and Applications", *Proc. IEEE*, vol. 63, pp.1692-1716, Dec. 1975.
(2) Simon Haykin, Adaptive Filter Theory, Prentice Hall, II. Edition
(3) John R. Glover, Jr., "Adaptive Noise Canceling Applied to Sinusoidal Interferences", *IEEE Trans. ASSP,* Vol. ASSP-25, No. 6, pp. 484-491, Dec. 1977.
(4) J.R. Zeidler et al., "Adaptive Enhancement of Multiple Sinusoids in Uncorrelated Noise", *IEEE Trans. ASSP,* Vol. ASSP-26, No. 3, pp. 240-254, June 1978.
(5) D. W. Tufts, "Adaptive Line Enhancement and Spectrum Analysis", *Proc. IEEE (Letts.)*, vol. 65, pp.169-170, Jan. 1977
(6) L. Griffiths, *Proc. IEEE (Letts.)*, vol. 65, pp.170-171, Jan. 1977
(7) B. Widrow et al., *Proc. IEEE (Letts.)*, vol. 65, pp.171-173, Jan. 1977

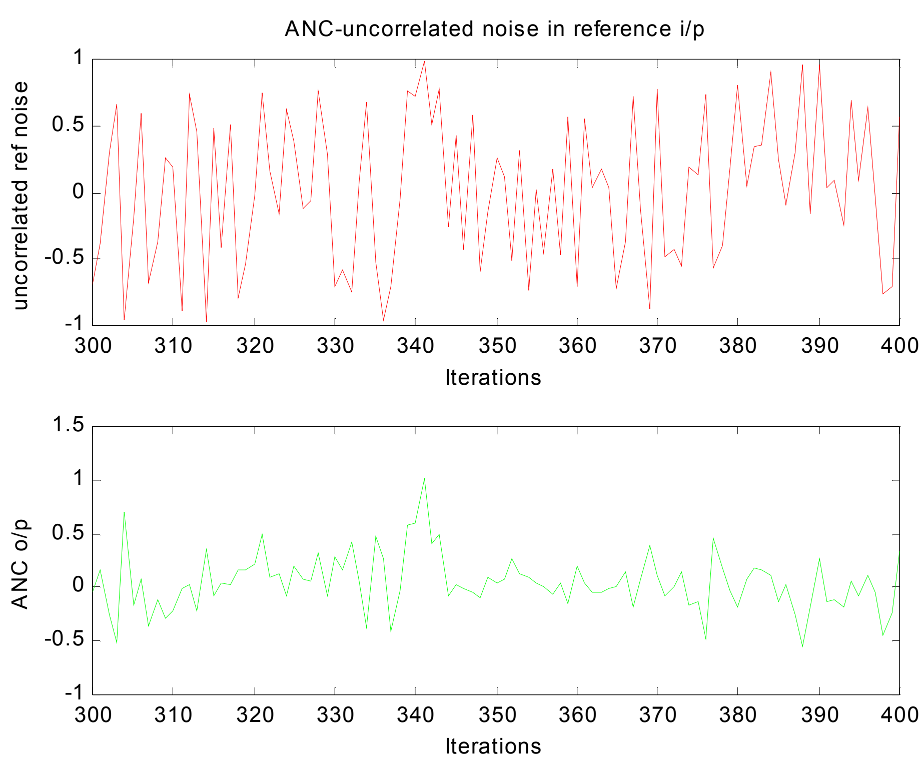
Simulation Results

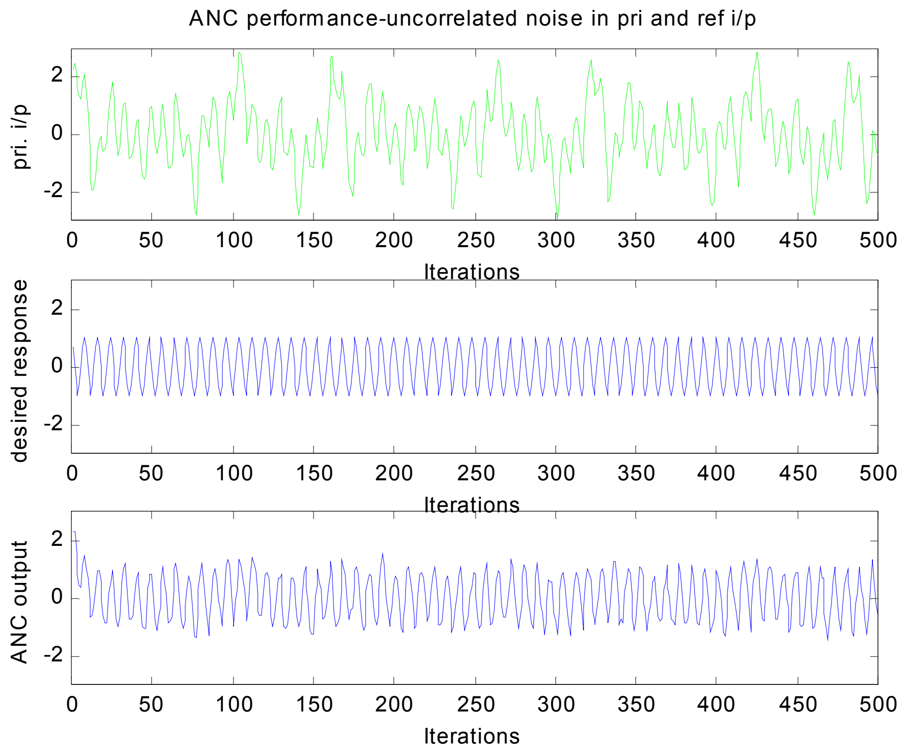
(1) Adaptive Noise Canceller



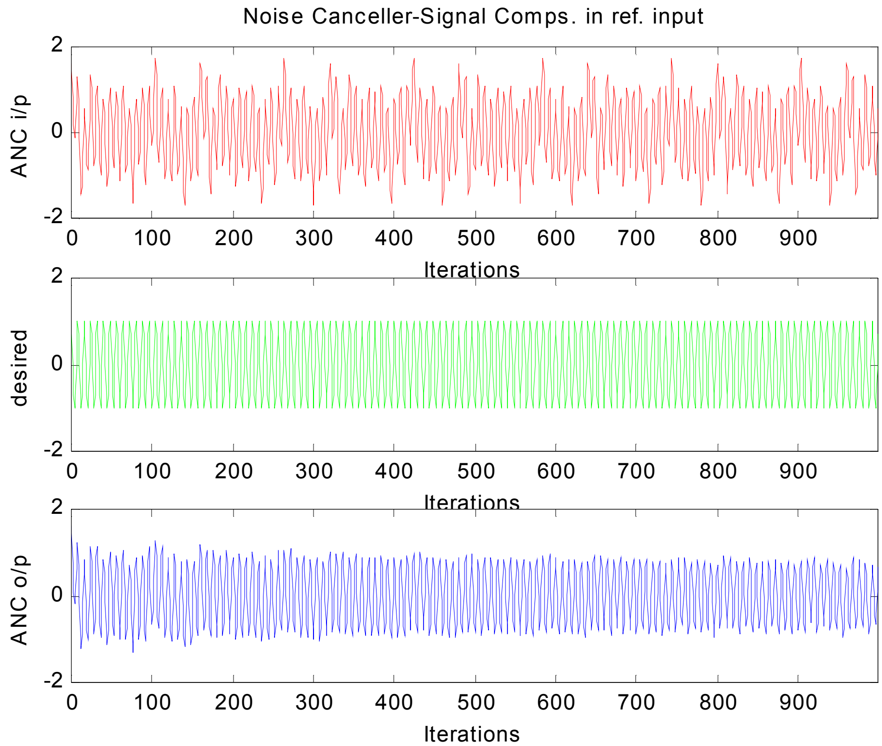
(2) ANC with uncorrelated noises in primary and reference inputs



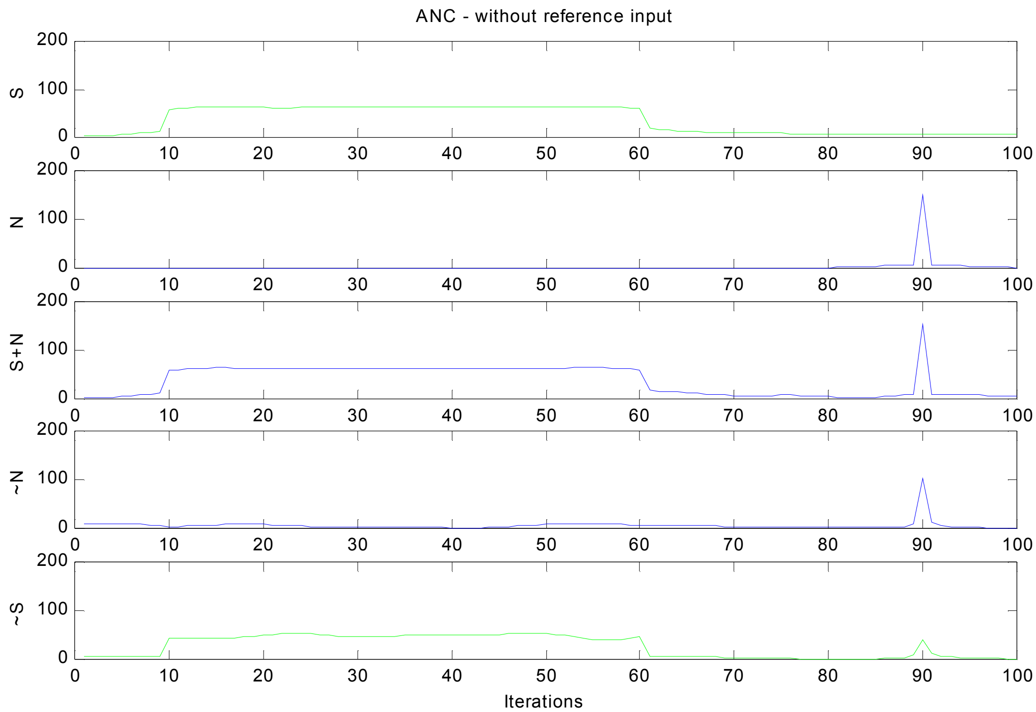



ANC performance-uncorrelated noise in pri and ref i/p
pri. i/p
desired response
ANC output
Iterations
Iterations
Iterations
0
50
100
150
200
250
300
350
400
450
500
2
0
-2

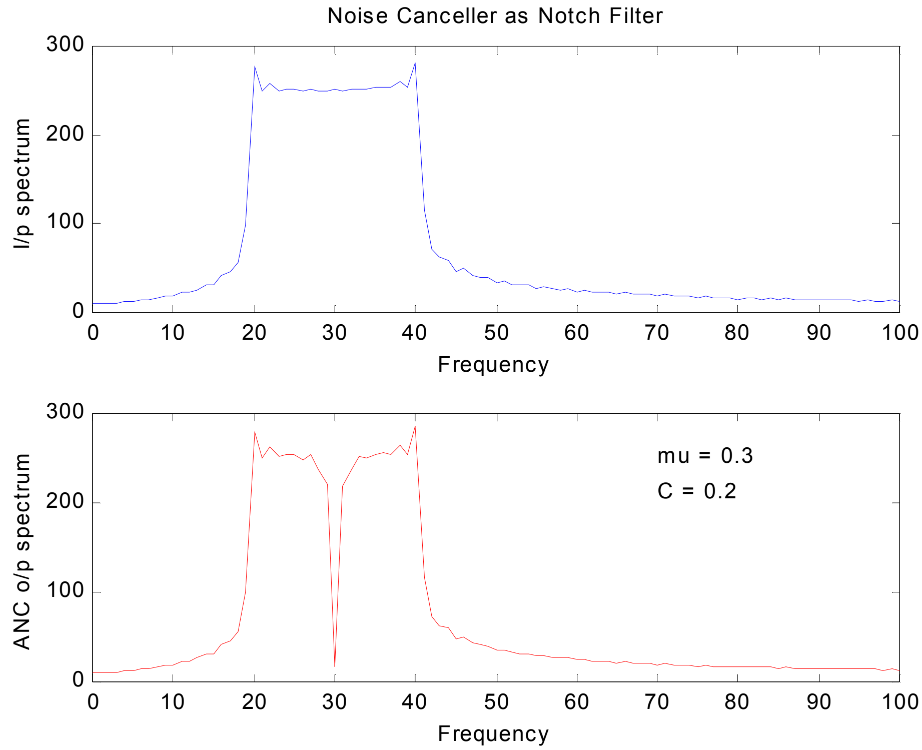
(3) ANC with signal components in reference input

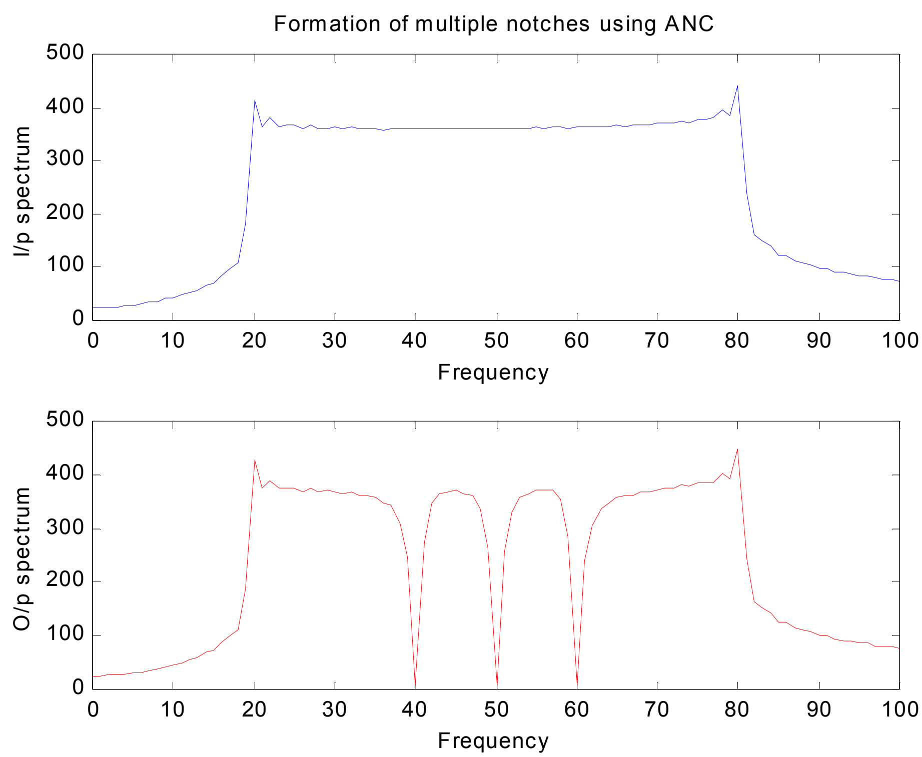


(4) ANC – without reference input

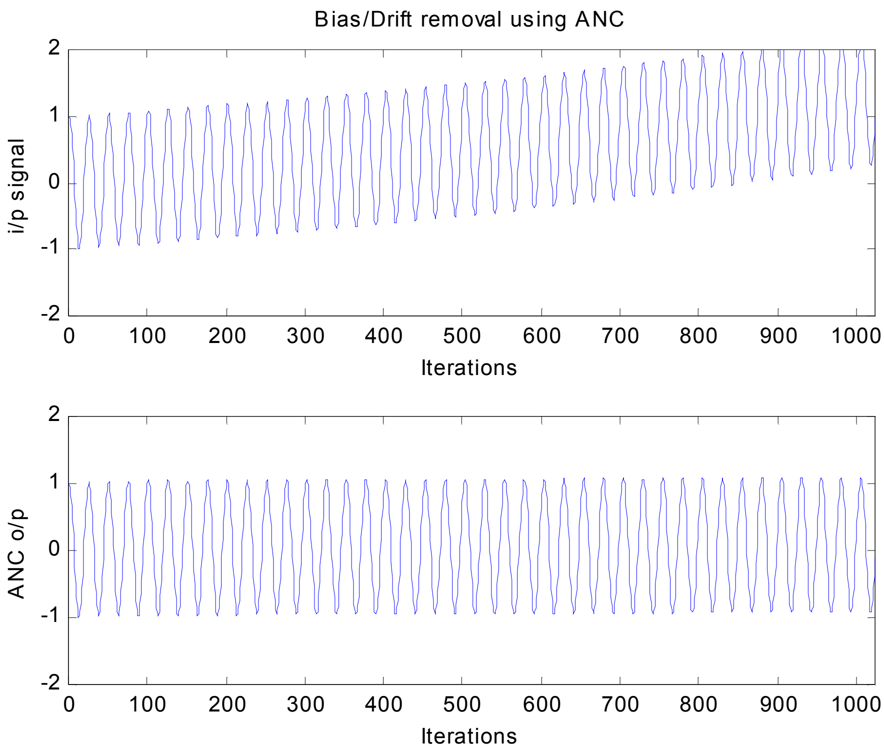


(5) ANC as notch filter – single and multiple frequency canceller

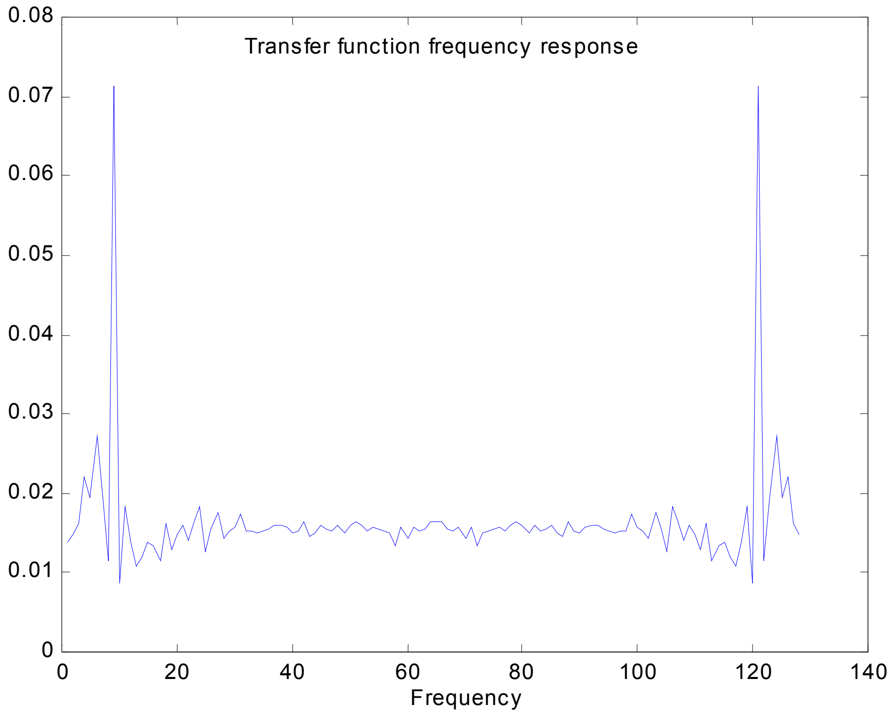




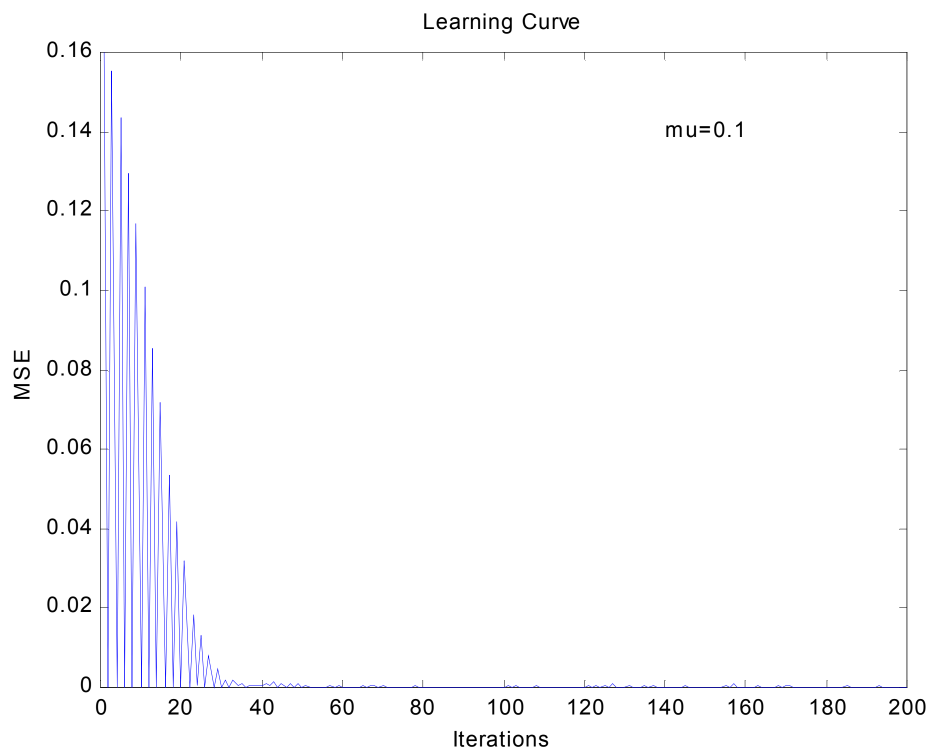
(6) Bias/Drift Removal using ANC

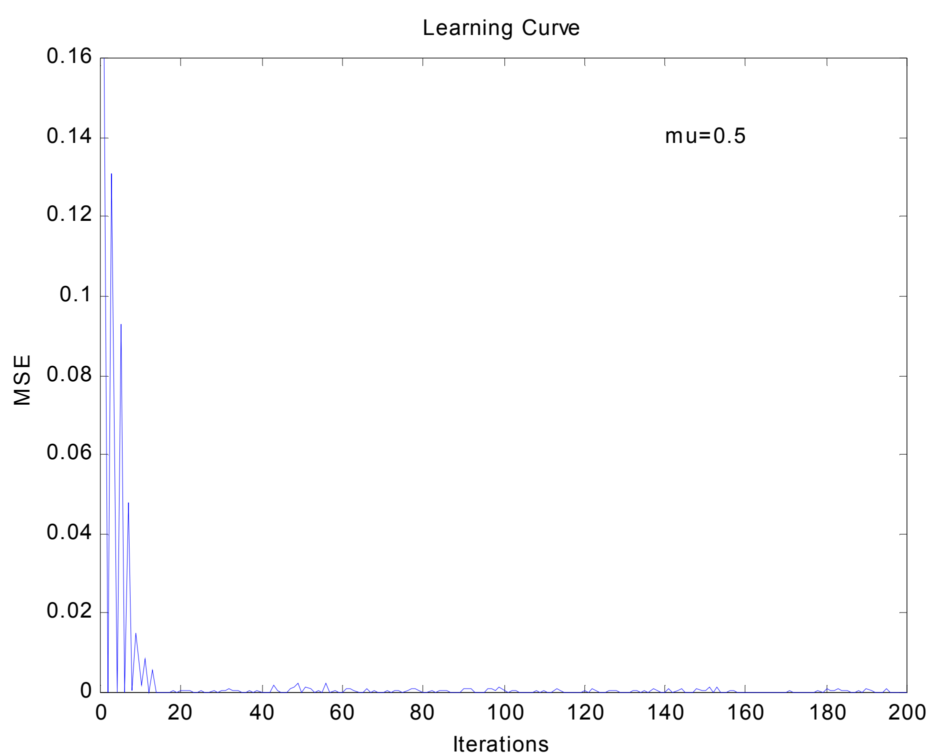


(7) ALE as Adaptive Line Enhancer

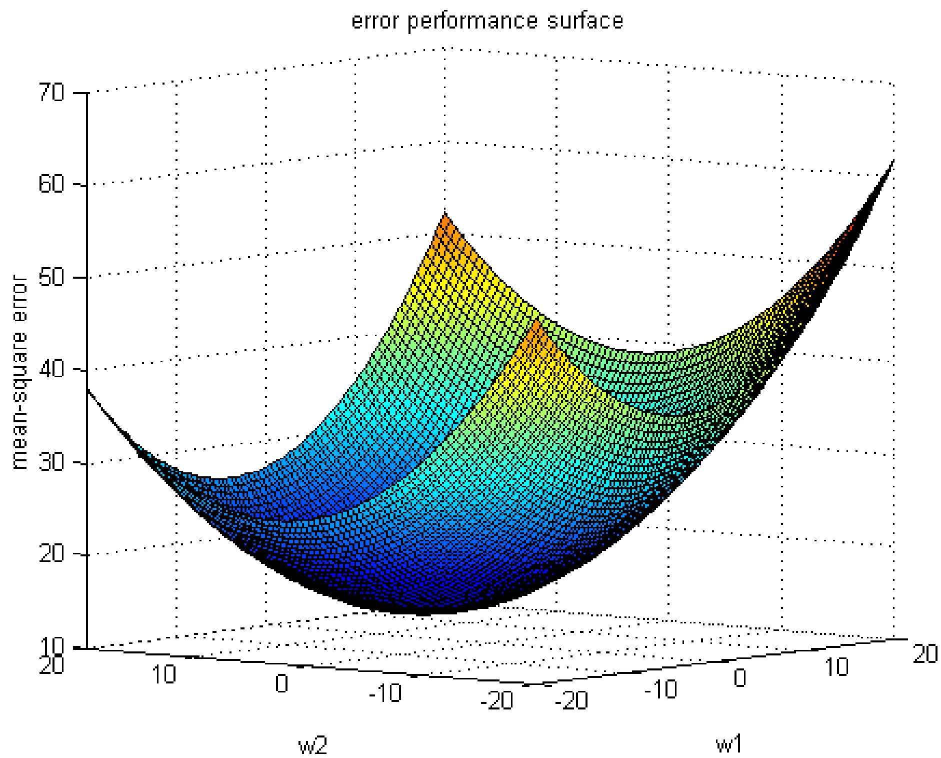


(8) Learning Curves for various step-size parameters





(9) Error performance surface



Matlab codes

(1)

```
                     %Adaptive Noise Canceller

clear;
M=16;                      %order of filter
mu=0.04;                   %step-size
N=200;                     %Iterations

f=750;
Ts=1/(4*f);                %fs=4 times the freq of the signal
noise=(rand(N,1)-0.5);
n=zeros(M,1);
w=zeros(M,1);

Wn=[0.1 0.5];           %see w/o filter
[B,A]=BUTTER(2,Wn);
x=filter(B,A,n);

for i=1:N
   t=(i-1)*Ts;
   for k=M:-1:2
      n(k)=n(k-1);
   end;
   s(i)=cos(2*pi*f*t);
   n(1)=0.2*(cos(2*pi*50*t)+sin(2*pi*100*t)+cos(2*pi*60*t)+
             sin(2*pi*80*t)+cos(2*pi*30*t)+ sin(2*pi*20*t)+
             sin(2*pi*10*t)+ sin(2*pi*90*t));             %noise(i);
   d(i)=s(i)+n(1);
   x=filter(B,A,n);
   d_out(i)=w'*x;
   e(i)=d(i)-d_out(i);
   w=w+mu*e(i)*x;
end;

i=1:N;
subplot(3,1,1);
plot(i,d,'g');
title('Adaptive Noise Canceller');
xlabel('Iterations');
ylabel('ANC i/p');
axis([1 N -2 2]);
subplot(3,1,2);
plot(i,s,'b');
xlabel('Iterations');
ylabel('desired');
axis([1 N -2 2]);
subplot(3,1,3);
plot(i,e,'r');
xlabel('Iterations');
ylabel('ANC o/p');
axis([1 N -2 2]);
```

(2)

```
    %ANC with uncorrelated noises in primary and reference inputs

sampling_time=1/(8*200);
mu=0.04;
M=16;
Iterations=500;
u=zeros(M,1);
x=zeros(M,1);
w=zeros(M,1);
e=zeros(Iterations,1);
Wn=[0.1 0.5];
[B,A]=BUTTER(4,Wn);
pri_n=0.5*(rand(Iterations,1)-0.5);
ref_n=0.5*(rand(Iterations,1)-0.5);

for n=0:Iterations-1
   t=n*sampling_time;
   for i=M:-1:2
      u(i)=u(i-1);
   end;
   u(1)=0.5*(cos(2*pi*50*t)+sin(2*pi*100*t)+cos(2*pi*60*t)
             +sin(2*pi*80*t)+cos(2*pi*30*t)+sin(2*pi*20*t));
        %rand(1);
   d(n+1)=cos(2*pi*200*n*sampling_time)+u(1)+pri_n(n+1);
   x=filter(B,A,u)+ref_n(n+1);
   d_out=conj(w')*x;
   e(n+1)=d(n+1)-d_out;
   w=w+mu*x*conj(e(n+1));
end;

n=1:Iterations;
subplot(3,1,1);
plot(n,d,'g');
title('ANC performance-uncorrelated noise in pri and ref i/p');
axis([0,500,-3,3]);
xlabel('Iterations');
ylabel('pri. i/p');
subplot(3,1,3);
plot(n,e','b');
axis([0,500,-3,3]);
xlabel('Iterations');
ylabel('ANC output');
subplot(3,1,2);
plot(n,cos(2*pi*200*n*sampling_time));
ylabel('desired response');
xlabel('Iterations');
axis([0,500,-3,3]);
```

(3)

```
                %ANC with signal components in reference inputs

sampling_time=1/(8*200);
mu=0.05;
M=16;
Iterations=1000;
u=zeros(M,1);
x=zeros(M,1);
w=zeros(M,1);
s=zeros(M,1);
e=zeros(Iterations,1);
Wn=[0.1 0.5];
[Bn,An]=BUTTER(2,Wn);
Ws=0.5;
[Bs,As]=BUTTER(2,Wn);

for n=0:Iterations-1
   t=n*sampling_time;
   for i=M:-1:2
        u(i)=u(i-1);
        s(i)=s(i-1);
   end;
   u(1)= 0.2*(cos(2*pi*50*t)+sin(2*pi*100*t)+cos(2*pi*60*t)
              +sin(2*pi*80*t) +cos(2*pi*30*t)+sin(2*pi*20*t));
      %rand(1)-0.5;
   s(1)=cos(2*pi*200*n*sampling_time);
   sig(n+1)=s(1);
   noi(n+1)=u(1);
   d(n+1)=s(1)+u(1);
   x=filter(Bn,An,u)+0.04*filter(Bs,As,s);
   d_out=conj(w')*x;
   e(n+1)=d(n+1)-d_out;
   w=w+mu*x*conj(e(n+1));
end;

n=0:Iterations-1;
subplot(3,1,1);
plot(n,d,'r');
title('Noise Canceller-Signal Comps. in ref. input');
axis([0 Iterations-1 -2 2]);
xlabel('Iterations');
ylabel('ANC i/p');
subplot(3,1,2);
plot(n,cos(2*pi*200*n*sampling_time),'g');
axis([0 Iterations-1 -2 2]);
xlabel('Iterations');
ylabel('desired');
subplot(3,1,3);
plot(n,e','b');
axis([0 Iterations-1 -2 2]);
xlabel('Iterations');
ylabel('ANC o/p');

ZOOM XON;
```

(4) ANC as notch

```
%ANC notch

clear;
mu=0.3;
M=2;
Iterations=512;

C=0.2;
w0=2*pi*30;
phi=pi/4;

phi1=phi;
phi2=phi+pi/2;

w=zeros(M,1);
e=zeros(Iterations,1);

fs=(20:1:40)';
ws=2*pi*fs;
A=rand(size(ws));
theta=2*pi*rand(size(ws));
Ts=1/(8*max(fs));

for n=1:Iterations
    t=(n-1)*Ts;
    s(1:size(ws),n)=(cos(ws*t));
    signal(n)=sum(s(1:size(ws),n));
    pri_noise(n)=n_a*cos(w0*t+2*pi*n_p);
    d(n)=signal(n)+pri_noise(n);
    x(1,1)=C*cos(w0*t+phi1);
    x(2,1)=C*cos(w0*t+phi2);
    y=conj(w')*x;
    e(n)=d(n)-y;
    w=w+2*mu*conj(e(n))*x;
end

f=0:100;
Se=zeros;
Sd=zeros;
Ssignal=zeros;
for n=1:Iterations
   Se=Se+e(n)*exp(-sqrt(-1)*2*pi*f*(n-1)*Ts);
   Sd=Sd+d(n)*exp(-sqrt(-1)*2*pi*f*(n-1)*Ts);
   Ssignal=Ssignal+signal(n)*exp(-sqrt(-1)*2*pi*f*(n-1)*Ts);
end;
subplot(2,1,1);
xlabel('Iterations');
plot(f,abs(Sd),'b');
title('Noise Canceller as Notch Filter');
ylabel('I/p spectrum');
xlabel('Frequency');
subplot(2,1,2);
plot(f,abs(Se),'r');
ylabel('ANC o/p spectrum');
xlabel('Frequency');
text(70,250,'mu = 0.3');
text(70,210,'C = 0.2');
```

(5) ANC for Bias/Drift removal

```
clear;
N=1024;
mu=0.01;            %change mu=0.001 and see the result-mmse vs. speed
of adaptation
ws=200;
Ts=1/(4*ws);
%drift=1.2;              %constant bias;
w(1)=0;
for n=1:N
   t=(n-1)*Ts;
   s(n)=cos(ws*t);
   drift(n)=(-1+exp(0.0008*n));
   d(n)=s(n)+drift(n);
   drift_=w(n);
   s_(n)=d(n)-drift_;
   e(n)=s_(n);
   w(n+1)=w(n)+2*mu*e(n);
end;


subplot(2,1,1);
plot(d);
title('Bias/Drift removal using ANC');
ylabel('i/p signal');
xlabel('Iterations');
axis([0 N -2 2]);
subplot(2,1,2);
plot(s_);
ylabel('ANC o/p');
xlabel('Iterations');
axis([0 N -2 2]);
```

(6) ANC as Adaptive Line Enhancer

```
clear;
f=200;
Ts=1/(16*f);

L=1024;
for l=1:L
   t=(l-1)*Ts;
   s(l,1)=0.1*cos(2*pi*f*t);
end;

N=8;
for l=1:L
    n(l,1:N)=normrnd(0,0.8,1,N);
end;

mu=0.01;
M=128;
for i=1:N
   w=zeros(M,N);
   d=s+n(1:L,i);
   ref=zeros(M,1);
   for l=1:L
      for k=M:-1:2
         ref(k)=ref(k-1);
      end;
      ref(1)=n(l,i);
      d_out(l)=w(1:M,i)'*ref;
      e(l)=d(l)-d_out(l);
      w(1:M,i)=w(1:M,i)+2*mu*ref*conj(e(l));
   end;
end;

w_avg=zeros(M,1);
for i=1:N
   w_avg=w_avg+w(1:M,i);
end;
w_ens=w_avg/N;

DFT=DFTMTX(length(w_ens))*w_ens;
Power_spec=DFT.*conj(DFT);

plot(Power_spec);
title('ALE - Adaptive Line Enhancer');
xlabel('frequency');
ylabel('Transfer function freq. response');
```