Project 2: Adaptive Noise Cancelation

In many environments, we wish to reduce or remove noise or interference from a signal of interest. In some cases, we have a reference for the noise signal which we can use to enhance the signal of interest. In other cases, we have no noise reference.

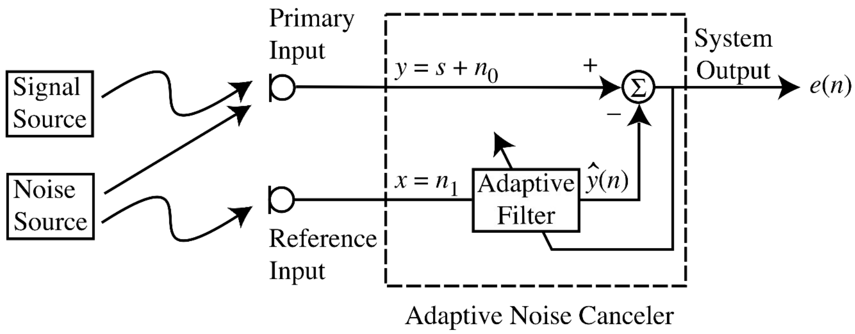

Figure 1: Hand-free telephony in an automobile scenario

Naturally, if the interference is well-understood (deterministic), one could design a frequency-selective filter to remove it. However, if the interference has a random component, then a fixed filter would not suffice. In this case we must design a filter which "tracks" the random nature of the interference.

In this project, we consider the adaptive cancellation of a narrow-band interference (tone) whose frequency and phase are random. The random tone is embedded in a signal of interest: the goal is to eliminate the interference while preserving (to the best extent) the signal of interest.

Demo
Sade random tones
Kravitz chirp

We'll explore two solutions known as the adaptive noise canceler (ANC) and the adaptive line enhancer (ALE). Note that we assume A_0, ω_0, and ϕ_0 are changing at random.

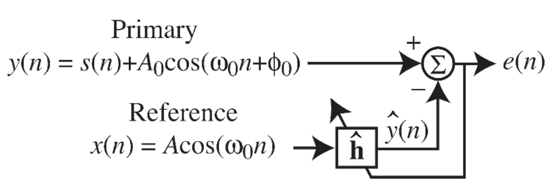

Figure 2: Basic block diagram of ANC (given both a primary and reference)

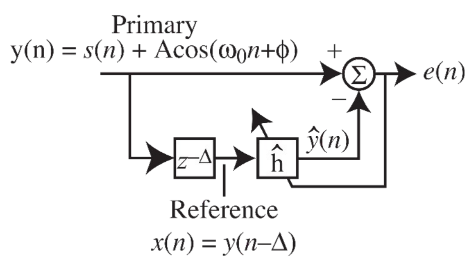

Figure 3: Basic block diagram of ALE (given only a primary)

Filter Classes

Deterministic Linear Filtering

If prescribed filtering specifications are given, such as "remove all frequencies above 400 Hz" or "remove all frequencies in the band 200 - 600 Hz", then the design of an LTI filter entails three basic steps:

- approximation of specifications by a rational transfer function, $H(z)$ (may be either FIR or IIR)
- the choice of appropriate structure or realization (Direct, Canonical, or Cascade Forms or Lattice)
- implementation of the filtering algorithm (MATLAB, C, Assembly, FPGA)

Statistical Linear filtering

Design Example: Given a noisy input signal (data), design a filter to minimize the effects of the noise. We may consider the noise to be broadband Gaussian noise, additive harmonic noise (sinusoids), or even echoes. In this case since the input signal or filter requirements are not deterministic. What can we do?

Optimal Filters: Certain statistics are available about the signal and noise and we can design filters which are statistically-optimal
1) Assuming a stationary (statistics are not time-varying) input, one solution is the *Wiener* filter (MMSE optimal)
2) Assuming a non-stationary input, one solution is the *Kalman* filter (MMSE optimal)

Adaptive Filters: Statistics are not available about the signal and noise
1) The solution is the *adaptive filter*, i.e. a filter whose coefficients are time-varying.

Definition (Adapt): To make suitable; to fit, or suit; to adjust; to alter so as to fit for a new use.

Adaptive signal processing in essence is developing algorithms to adjust or adapt filter coefficients to filter noisy signals. EE592 Adaptive Signal Processing.

Our focus in this project is on adaptive filters. In order to understand adaptive filters, we must first understand the classic, optimal *Wiener filter*.

Wiener Filters

One common filtering problem is to design an FIR filter $\hat{h}$, such that when driven by an input $x(n)$, yields an output, $\hat{y}(n)$ that is *close* in some sense to a desired output, $y(n)$.

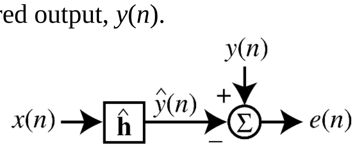

Figure: Linear model of a desired signal.

If we're already given $y(n)$ why do we need to estimate it? In many cases $y(n)$ has an interference which we want to eliminate and $\hat{y}(n)$ helps us model and remove the part.

Example: Acoustic echo cancelation

One measure of the quality of the statistical filter is how well it minimizes the mean-squared error $\xi \equiv E\left[e(n)^2\right]$ where E[·] is the statistical expectation operator. Our optimization problem is to find the set of filter coefficients $\hat{h} = \left[\hat{h}_0, \hat{h}_1, L, \hat{h}_{N-1}\right]^T$ that make ξ as small as possible. The optimal solution is called the *Wiener filter*.

The error signal at sample index *n* is given by

$$e(n) = y(n) - \hat{y}(n) \tag{3}$$

where $y(n)$ is the desired output signal and $\hat{y}(n)$ is the approximation to $y(n)$. The filter output $\hat{y}(n)$ is given by

$$\begin{aligned} \hat{y}(n) &= \sum_{k=0}^{N-1} \hat{h}_k x(n-k) \\ \text{¿} &\hat{h}^T x(n) \end{aligned} \tag{4}$$

where ^{T} denotes matrix transpose,

$$x(n) = \left[x(n), x(n-1), L, x(n-N+1)\right]^T \tag{5}$$

is the vector containing the current input sample as well as the previous $N - 1$ input and

$$\hat{h} = \left[\hat{h}_0, \hat{h}_1, L, \hat{h}_{N-1}\right]^T \tag{6}$$

is the vector containing the *N* filter coefficients. The mean-squared error (MSE) is defined as

$$\begin{aligned} \xi &\equiv E\left[e(n)^2\right] \\ \text{¿} &E\left\{\left[y(n) - \hat{y}(n)\right]^2\right\} \\ \text{¿} &E\left\{\left[y(n) - \hat{h}^T x(n)\right]^2\right\}. \end{aligned} \tag{7}$$

Expanding (7) we have,

$$\begin{aligned} x &= E\left[y(n)^2\right] - E\left[\hat{h}^T x(n) y(n)\right] - E\left[y(n) x^T(n) \hat{h}\right] + E\left[\hat{h}^T x(n) x^T(n) \hat{h}\right] \\ \text{¿} &E\left[y(n)^2\right] - \hat{h}^T E\left[x(n) y(n)\right] - E\left[y(n) x^T(n)\right]\hat{h} + \hat{h}^T E\left[x(n) x^T(n)\right]\hat{h} \\ \text{¿} &E\left[y(n)^2\right] - \hat{h}^T p - p^T \hat{h} + \hat{h}^T R \hat{h} + \left(p^T R^{-1} p - p^T R^{-1} p\right) \\ \text{¿} &E\left[y(n)^2\right] - p^T R^{-1} p + \hat{h}^T R \hat{h} - \hat{h}^T p - p^T \hat{h} + p^T R^{-1} p \\ \text{¿} &E\left[y(n)^2\right] - p^T R^{-1} p + \left(\hat{h} - R^{-1} p\right)^T R \left(\hat{h} - R^{-1} p\right) \\ \text{¿} &x_{\min} + \left(\hat{h} - R^{-1} p\right)^T R \left(\hat{h} - R^{-1} p\right) \end{aligned} \tag{8}$$

where

$$x_{\min} = E\left[y(n)^2\right] - p^T R^{-1} p \tag{9}$$

is the minimum mean-squared error (MMSE),

$$R^o E\left[x(n)\,x^T(n)\right] \tag{10}$$

is defined as the correlation matrix of *x*, and

$$p^o E\left[x(n)\,y(n)\right] \tag{11}$$

is defined as the cross-correlation vector between *x* and *y*.

Equation (8) is a quadratic equation in $\hat{h}=\left[\hat{h}_0,\hat{h}_1,L,\hat{h}_{N-1}\right]^T$ and forms an *N*-dimensional paraboloid

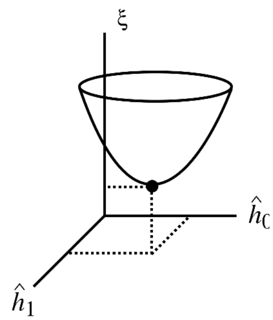

Figure: MSE paraboloid diagram

The MSE quadratic in (8) is minimized by choosing

$$\hat{h}_{\text{opt}}=R^{-1}p \tag{12}$$

assuming R^{-1} exists. The above solution is commonly referred to as the *Wiener filter* or Wiener solution or the Least-Squares (LS) filter.

When this filter is driven by *x*, its output $\hat{y}$, is the closest (in the LS sense) approximation to *y*. If the input or desired signal statistics change, i.e. **R** and/or **p**, the Wiener filter is no longer optimal and in this case must be recalculated.

If *x* or *y* have time-varying statistics, then the Wiener filter will not produce the MMSE at all points in time. In this case, $\hat{h}$ must be a time-varying filter and attempt a Wiener solution at all points in time. Solving the Wiener equation at every point in time is quite computationally expensive and so we turn to adaptive adjustment techniques. In addition, we may not have the true **R** and **p** in order to compute the Wiener filter.

