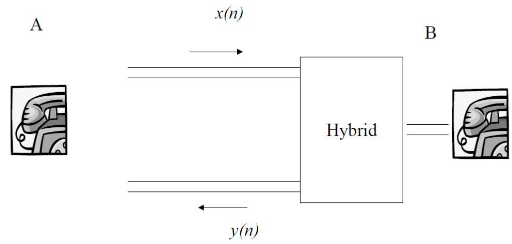
Applied signal processing

Lab 4, Feb-2017

In this lab, we will implement an echo cancellation filter.

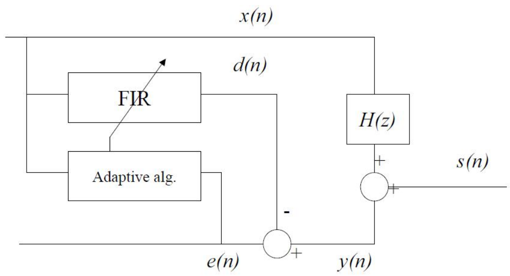
Back in time long-distance telephone systems employ four copper wires between two central offices. In each central office, a *hybrid* connects the four wire circuit to the two-wire circuit used between the central office and the subscriber (figure 1). Normally the hybrid should be balanced so that no echo goes back.



However, due to imperfect balancing of the hybrid (today mobile device), there is always a echo from the insignal $x(n)$ to the outsignal $y(n)$ back the talker. If the delay of this echo is more than 100 ms, it starts to disturb the communication. The delay is nowadays often increased due to digital techniques, due to increased buffering of data in the overall system. Hence, to minimize the echo in the outsignal $y(n)$ is important.

Today, we use digital communication systems, but the issue is the same, to avoid the echo generated in the end device.

One way of canceling the echo is to insert a adaptive FIR-filter at the hybrid end. This adaptive filter should estimate the transfer function $H(z)$ of the hybrid that causes the echo (figure 2). Using this filter, a estimate of the echo can be subtracted from the signal leaving the hybrid.



The crucial element in adaptive filters are the adaptive algorithm. Least Mean Square (LMS) algorithms are often used in this context. In this exercise we will use Stochastic Iteration Algorithm given by

$$H(n+1) = H(n) + \mu e(n) X(n) \tag{1}$$

where $H(n)$ is a vector describing the estimate of the filter coefficients at iteration n, $e(n)$ is the estimation error at time n, $X(n)$ is the vector of input signals, and μ is the adaptation step. This corresponds to what is given in the lectures.

The error $e(n)$ is given as the difference

$$e(n) = y(n) - d(n) \tag{2}$$

The estimate d(n) of the output signal y(n) is given by the FIR-filter

$$d(n) = X^T H(n) \tag{3}$$

At iteration n, this can be written as

$$d(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \tag{4}$$

where M is the filter length of the FIR filter used, $h(k)$ the filter coefficients and x is the vector of input samples.

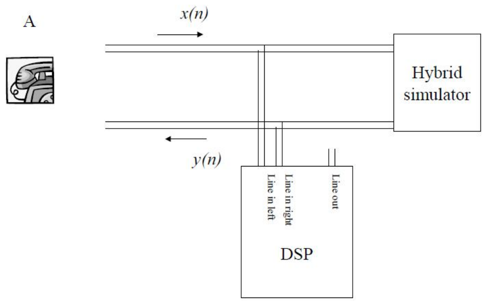
One problem with this algorithm is that it will not adapt well to input signals with large variation in signal power, as the adaptation step μ is fixed. One way is to make the adaptation step variable. The following can be used

$$H(n+1) = H(n) + \frac{\mu^*}{\rho + \sigma^2} e(n)X(n) \tag{5}$$

where u^* is a dimensionless step, σ^2 is the total input power, and ρ is a small constant to prevent small values of input power to result in large coefficient correction terms. As the total input power is unavailable s^2 is taken as the input signal windowed over a certain period. Here the squared norm of the input vector can be used, $|X(n)|^2$. The algorithm will be convergent if the following condition is satisfied:

$$0 \leq \mu^2 \leq 2 \tag{6}$$

In this exercise, we will use a system like in figure 3. The task is to implement the adaptive filter in the DSP implement the adaptive filter so that echos are removed.


A
x(n)
Hybrid
simulator
y(n)
Line in left
Line in right
Line out
DSP