Tutorial: Adaptive filter, acoustic echo canceller, and its low power implementation

Part I: Adaptive filter

An adaptive filter is a computational device that attempts to model the relationship between two signals in real time in an interactive manner. Adaptive filters are well accepted in communication systems, for echo cancellation and line equalization. An adaptive filter is also suitable for real time control systems for different kinds of applications related to real-time optimization. Adaptive signal processing is also expanding in other fields such as radar, sonar, seismology, and biomedical electronics.

An adaptive filter is defined by four aspects:

a. the signals being processed by the filter *x(n)*
b. the structure that defines how the filter output is computed from its input signal
c. the filter parameters can be interactively changed to alter the filter's in-out relation.
d. the adaptive algorithm that describes how parameters are adjusted from one time instant to the next.

An adaptive filter could be implemented as an open-loop filter or a closed-loop filter. We refer to a close-loop filter in this tutorial. The algorithm operates in an iterative manner and updates the adjustable parameters with the arrival of new data and current-signal performance feedback parameters. During each iteration, the system learns more about the characteristics of the input signal. The processor makes adjustments for the current set of parameters based on the latest system performance, i.e. the error signal *e (n).* The optimum set of values of the adjustable parameters is then approached sequentially.

Adaptive filters are often realized as a set of program instructions running on a digital signal processor. In general, any system with a finite number of parameters that affect how *y (n)* is computed from *x (n)* could be used for the adaptive filter in the following figure:

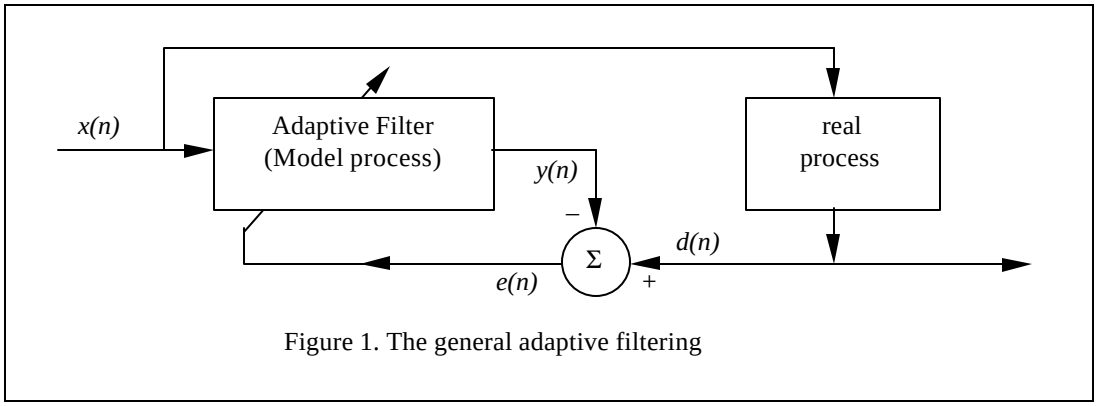

Figure 1. The general adaptive filtering

The coefficient vector of the filter is defined $T(n) = [t_0(n)\ t_1(n) \ldots t_{L-1}(n)]^{T}$. The input of the filter is *x (n).* The output of the filter is *y (n).* The desired response signal is *d (n).* The error signal *e (n)* is from the difference of the desired signal and the real output. The most frequently used structure of an adaptive filter is the *finite-impulse-response FIR* filter, figure 1.

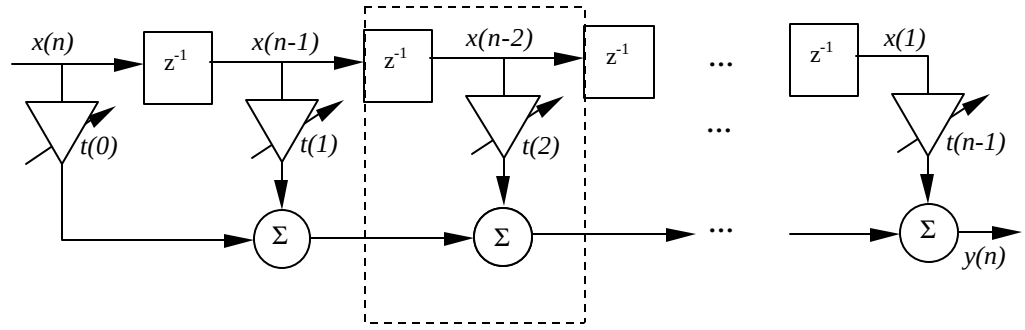

Figure 2. Adaptive FIR filter

In the figure above, the unit z^{-1} is called the delay unit. The filter itself is based on convolution. The computing step included in the dash-line block is called a tap. A filter tap includes both a step of convolution and a step of coefficient adaptation. The algorithm of the FIR filter is given in *eq. 1*.

$$Y(n) = \sum_{i=0}^{n-1} x(i)t(n-i) \qquad \text{eq. 1}$$

When the output is not close to *d(n),* the adaptation algorithm will be executed to correct the coefficient set so that the output *y(n)* will gradually approach *d(n).* The desired signal *d(n)* is unknown and changes all the time. Therefore, the adaptive filter has to be a real time close loop feedback system adapting all the time to follow the definition of *d(n).*

In a high quality adaptive filter, the coefficient set is adapted all the time and it costs a lot in terms of computing power. Therefore, it makes an adaptive filter expensive in computing power. The most popular adaptation algorithm is called the normal *least-mean-square algorithm,* with the abbreviation of *LMS* or *NLMS*. The LMS algorithm makes use of the steepest descent approach, and it derives the estimation of the gradient vector based on a limited number of data samples.

This adaptation algorithm includes the convergence control and the coefficient adaptation. The convergence control will not be performed in every tap so that the cost of the computing power is not high. The optimization of the convergence control is therefore not included in this tutorial. The coefficient adaptation, however, should be

performed for all taps during each sample for a high performance adaptive filter. Normally, most of the computing power is consumed when performing the coefficient adaptation. A tap of computing is given in the following equations:

$$y(i) = y(i-1) + x(i)t(n-i) \qquad \text{eq. 2}$$

$$t_{new}(i+1)=t_{old}(i+1)\pm(\text{convergence factor})*x(i) \qquad \text{eq. 3}$$

In the above equation, $t_{new}\ (i+1)$ is the new coefficient after adaptation, and $t_{old}\ (i+1)$ is the old coefficient going to be adapted.

Part II: Acoustic and electric echo canceller

The are mainly two kinds of echo existing in communication systems, the electric echo and the acoustic echo. The electric echo is also called hybrid echo or line echo. This echo can be found in the public-switched telephone network (PSTN), mobile, and IP phone systems. The electric echo is created at the hybrid connection at the two-wire / four-wire PSTN conversion points in the following figure:

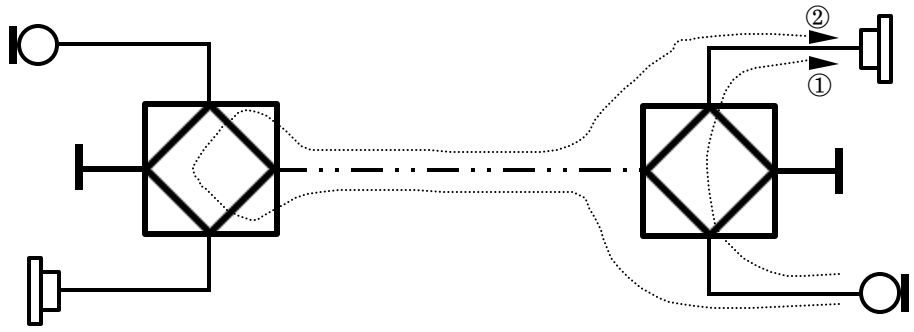

Figure 3. Hybrid and generated electric echo

The electric echo can be generated from both the near end electric devices ① and the far end electric devices ②. The near end echo ①, hybrid echo, has been around almost since the advent of the telephone itself. Due to the economic reason, we use two-wire system to perform full duplex function that actually require the performance of a four-wire system. The principle is to use different kinds of “hybrid” to balance and separate the sending and receiving signals. The real hybrid circuits can not be 100% ideal because of the leakage, and the parasitic or parameter deviations. Therefore, part of the signal takes the wrong path from both the near end hybrid and the far end hybrid and becomes echo. In the old telephone system (POT over PSTN) the echo is 28ms or less. In the modern telephone

system, the electric echo will be longer. The electric echo in GSM could be up to 80ms The electric echo in IP telephone could be up to 120ms or even more.

Acoustic echo is generated from either handfree telephone or a telephone with poor voice coupling between the earphone and the microphone. In this tutorial, we only discuss the acoustic echo induced by a handfree telephone. In this kind of application, we have to let remote voice goes through the loudspeaker and becomes part of the microphone signal. When subscriber B is calling subscriber A, and A uses a handfree telephone, the voice from subscriber B is sent to the loudspeaker. The microphone of subscriber A picks up both the voice from the subscriber A and the loudspeaker voice from subscriber B. Thus, the subscriber B receives an acoustic echo. There are two different components making up the acoustic echo. The first is the direct coupling between the loudspeaker and the microphone, and the second is the undesired remote speech reflected from roof, windows, and walls. The echo from the second component could be as long as 200 millisecond.

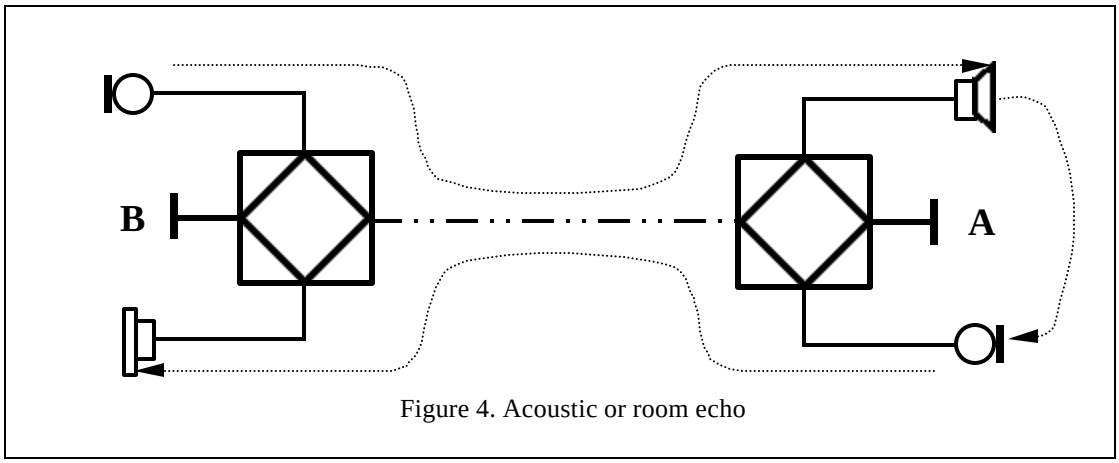

Figure 4. Acoustic or room echo

High quality telephone equipment will require a good acoustic echo canceller. It should fulfill at least:

1. Full duplex: Transfer complete voice on both sides, no voice cutting at all.
2. High echo compression: larger than 30dB when the echo canceller is stable.
3. Fast convergence: Fast following the change of the room environment.
4. Stable, not oversensitive to room environment changes.

The early echo cancellation implementations were based on analog circuit technique. Because the analog technique can not completely follow changes of room environments, it was taken over by digital technique. A digital echo canceller is an adaptive FIR with long tap size. For example, the number of taps is more than 3200 for a 200ms echo-canceller used in a 16kHz sampling-rate ISDN telephone system. It means that at least 7200 MAC operations are required in every sample including 3200MAC for the convolution and 3200MAC for the coefficient adaptation. It is equivalent to 102.4MIPs. Including other associated computing and control operations, the total number of MIP

could be more than 110 MIPs. One way to decrease the number of MIP is to skip part of the adaptation computing, which gives relatively low adaptation quality.

Part III: Low Power acoustic echo canceller

Digital signal processors are implemented using digital CMOS technology. The power consumption in a digital CMOS circuit consists of dynamic power consumption, static power consumption, and short-circuit power consumption. The dominant power consumption is normally from the dynamic power. The power is used in charging capacitance. The modeling of the dynamic power is:

$$P_{dyn} \sim V^2 S\, Cf \qquad \text{eq. 4}$$

Here P_{dyn} is the dynamic power consumption. V is the supply voltage and we suppose the change of the logic value is with full swing. C is the node capacitance and f is the toggling rate of the node. The dynamic power is proportional to the sum of all toggling on all nodes in the chip. It is related to the supply voltage, the chip size, and the circuit activities on chip. Note that when the supply voltage is very low, the static power consumption can be the dominant part.

In a normal application, the supply voltage is fixed from the system specification. The essential low-power technique is then to decrease the scale of the chip and decrease the redundant circuit activities. Low power circuit technique on basic logic cell level is not practical because it requires the design of a cell library or modification of an existing one.

The opportunity for low power is on the instruction design level and the architecture design level. A general-purpose DSP chip can not reach the low power because there is no way to change the architecture and the instruction set. It means that we can not cut off extra circuit activities by cutting off redundant instructions and hardware circuits. Using a state of the art DSP processor, about 100MIP DSP computing gives the power consumption about 100mW at year 2000. This is not acceptable for mobile-based applications.

The use of domain-specific digital signal processors is the best solution for a low power product. The first reason is that the instruction set is optimized for low power with enough flexibility. The second reason is that the hardware can be simplified compared to a general-purpose DSP processor. The third reason is that the acceleration hardware can be implemented for specific applications, so that kernel operations can be accelerated with higher performance and lower clock rate.

Terminology

Acoustic echo

Echo that occurs from sounds reflecting off surfaces in the surrounding environment.

CMOS Technology

Complementary **M**etal **O**xide **S**emiconductor, a family of integrated circuits whose output structure consists of an n-type metal oxide semiconductor field effect transistor (MOSFET) and a p-type MOSFET connected in series. The term complementary is used since the n-type and p-type MOSFETs are complement of each other.

Convergence time

The time required the echo canceller to fully learn the acoustic picture of a room.

Domain Specific Digital Signal Processor

A programmable digital signal processor dedicated for a group of applications.

Full duplex

Simultaneous, interactive two-way communications; transmission in two directions simultaneously; this means that if parties at either end of the conversation talk at the same time they will both be heard.

Half duplex

Transmit in one direction at one time. This means that if parties at either end of the conversation talk at the same time, only one party will be heard.

Heterogeneous DSP

The DSP system architecture, integrating more than one DSP processor for the complete digital signal processing in the system, each processor can have different architecture, different instruction set, and different embedded DSP programs.

Hybrid echo

Echo that occurs where two-wire to four-wire conversation takes place.

LMS

Least Mean Square based algorithm