Kalman Filter for Active Noise Control

Zhiqiang Xie
School of Information Science and Technology
ShanghaiTech University
Shanghai, China
xiezhq@shanghaitech.edu.cn

Abstract

This is the final report of the project on EE160 Introduction to Control course. Through this project we tried to implement an active noise control (ANC) headphone based on the Filtered-X structure, using both classicl LMS algorithm and simplified Kalman filter method. The LMS version has been deployed in hardware (a real headphone with two external welded microphones) via the interface of Simulink, the Kalman filter version is designed for theoretical analysis and some properties are discussed based on the simulation. In this report, the problem will be formulated from a control theory view, more details omitted in the presentation will be covered.

1. Introduction

1.1. Overview

Noise pollution is an increasing problem in modern society, which can be reflected from the prevailing of products about noise reduction. The most direct method of controlling the noise level we heard is to block or absorb it. For example using a muffler or specially designed chamber which reflects the noise back. These passive silencers are valued for their high attenuation over a broad frequency range; however, they are relatively large, costly, and ineffective at low frequencies.[2]

Therefore, the active noise control methods generate an anti-noise of equal amplitude and opposite phase, which is combined with the primary noise resulting in the cancellation of both noises based on the superposition principle. Since the world is not ideal, the active noise control problem is generating the proper anti-noise according to the prior knowledge and information received.

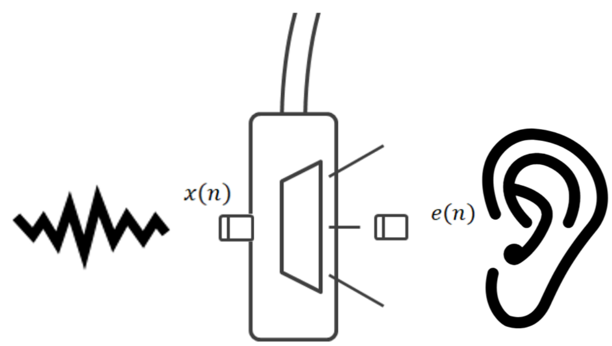

Figure 1. Diagram of the ANC in practice

1.2. System States

There're several ways to construct an ANC system, and the most popular one in practice is the feedforward control. The system we constructed is depicted in figure 1. Where the reference microphone is the one facing noise, error microphone is the one facing speaker.

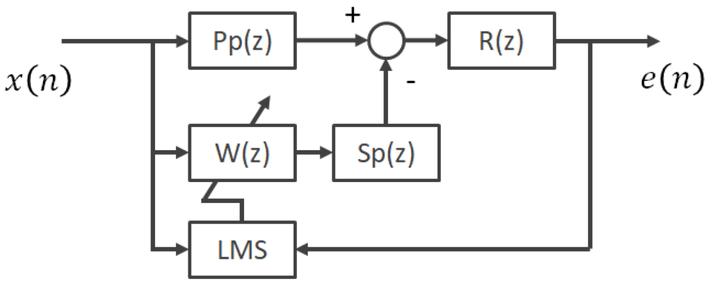

Figure 2. System representation of ANC

In fact, the feedforward here means we receive the primary noise signal (from the reference microphone) and emit the control signal (anti-noise) which is obtained by a ”pre-defined way”. Other than the general PID feedback control we use (obtain the control signal from the error signal by a ”pre-defined way”), we collect the error signal via the microphone to modify the parameters of the ”pre-defined way”. In some sense, the ANC system should be a hybrid control system if we treat the ”pre-defined way” to be a system state we care about. However, the processing of parameterization is usually not considered to be a control process, that's why I didn't specify the ”system states” clear in the presentation. Therefore, in this report, we'll **treat both the noise and the parameters of the ”pre-defined way” to be the states of the control system.**

2. Problem Formulation

2.1. Preliminaries

The diagram shown in figure 1 can be abstracted to a system representation figure 2. Here is a list of notations:

1. **x(n)**, primary noise, measured by the reference microphone
2. **P**, unknown primary path of noise from the reference microphone to the error microphone
3. **R**, parameterizing the error introduced by the error microphone and superposition of noises
4. **W**, our system state which is the core of the controller, calculating the anti-noise control signal
5. **S**, unknown (can be modelled offline) secondary path of anti-noise from the speaker to the error microphone
6. **LMS**, not only the LMS algorithm but a set of adaptive algorithm
7. **e(n)**, residual noise, measured by the error microphone

Almost any other unlisted components of the entire system can be treated as part of above.

Besides, for a more concise state-space representation of control system, we'll make following simplifications or assumptions:

1. **use matrix multiplication to replace the convolution operation** The impulse response of a system block could be very complicated, and this simplification can be generalized in more sophisticated digital filters, which means the analysis based on this simplification is also meaningful.
2. **assume the secondary path to be stationary** Many offline modeling methods research have verified that it's reasonable[3].
3. **the block R only introduce a white Gaussian noise**

2.2. Formulation

$$
\begin{aligned}
u[n] &= W[n]x[n] \\
e[n] &= P[n]x[n] - S \cdot u[n] + R[n] \\
W[n+1] &= W[n] + g[n]e[n] \\
g[n] &= f(x[n])
\end{aligned}
$$

Here $u[n]$ is the control signal and the input of the system, $e[n]$ is the output of the system, $W[n]$ and

$x[n]$ are the two states, $R[n] \in \mathcal{N}(0, V)$. Specifically, the state $x[n+1]$ usually doesn't depend on $x[n]$. Thus, it's also feasible to treat it as a parameter and I'll show that in the next section.

2.3. Filtered-X structure

From the formulas above, if we want to find a steady solution for state W, we must introduce a term S^{-1} inside it. However, this inverse may not exist. We'll introduce a more effective approach: filtered-X structure (FX)[7] to cancel the effects of the secondary path. The transformed system is depicted in figure 3, which is nearly equivalent to the previous one if the coefficients of W don't change too fast. Here the idea can be formulated as follows:

$$
\begin{aligned}
e[n] &= P[n]x[n] - S \cdot (W[n]x[n]) + R[n] \\
&\approx P[n]x[n] - W[n](S \cdot x[n]) + R[n]
\end{aligned}
$$

Generally, this assumption is allowable. And now the formulas are:

$$
\begin{aligned}
x'[n] &= \hat{S} \cdot x[n] \\
u[n] &= W[n]x[n] \\
e[n] &= P[n]x[n] - S \cdot u[n] + R[n] \\
W[n+1] &= W[n] + g[n]e[n] \\
g[n] &= f(x'[n])
\end{aligned}
$$

The target of our control system is minimizing the residual noise ($e[n]$) level, which consist of getting proper $W[n]$ and then emitting proper anti-noise $u[u]$.

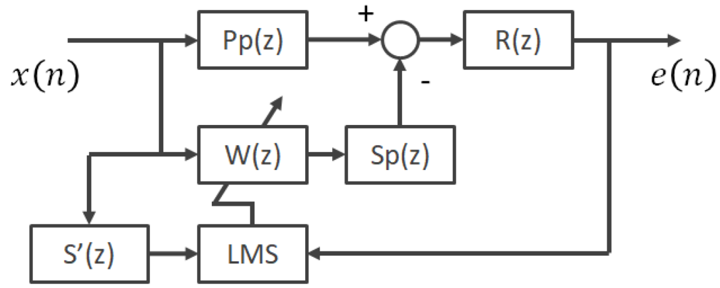

Figure 3. Filtered-X structure

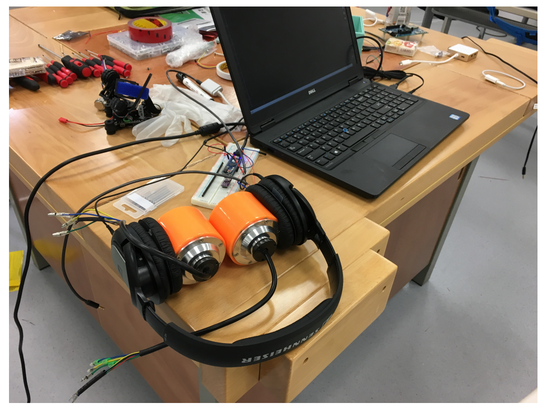

Figure 4. Offline Modeling Process

3. Solution Method

3.1. Offline Modeling

Some required prior knowledge should be modeled offline, such as secondary S. The detailed experimental setup and procedure for offline modeling related system blocks is summarized in Kuo's work[1]. The figure 4 shows our own modeling process.

3.2. Kalman Filter

This Kalman filter approach is basically based on part of Paulo's work[5]. The Kalman filter equations are:

$$
\begin{aligned}
x'[n] &= \hat{S} \cdot x[n] && (1) \\
W[n+1] &= W[n] && (2) \\
d[n] &= P[n]x[n] + R[n] && (3) \\
&= e[n] + W[n]x'[n] && (4)
\end{aligned}
$$

Specifically, equation (2) doesn't indicate the $W[n]$ being stationary. From a Bayesian estimation view, it means the previous prediction tell nothing about next prediction. In equation (3), $d[n]$ is a measurement of the unknown primary path $P[x]$: as we talked about $x[n]$, here we treat it as a parameter. Actually, $e[n]$ is our direct measurement but it's clear that $d[n]$ can be viewed as an indirect measurement.

Therefore, it now turns to be a general Kalman

Algorithm 1 The Kalman filter for ANC

At each time instant $t > 0$, based on measurements $\{d[n], x[n]\}$

1) Compute the Kalman gain:

$$g[n] = A[n-1]x'[n](x'[n]^T A[n-1]x'[n] + V)^{-1}$$

2) Update the coefficient estimate:

$$W[n] = W[n-1] + g[n]e[n]$$

3) Update the innovation covariance matrix (predict error):

$$A[n] = A[n-1] - g[n]x'[n]^T A[n-1]$$

filter form. More auxiliaries in the procedures:

$$\tilde{W}[n] = P[n] - W[n] \cdot \hat{S} \tag{5}$$

$$A[n] = E\{\tilde{W}[n]\tilde{W}[n]^T\} \tag{6}$$

where $\tilde{W}[n]$ is the innovation and $A[n]$ is innovation covariance matrix.

The overall procedures of Kalman filter method is described in Algorithm 1.

3.3. LMS Algorithm

This method is much simpler than the Kalman filter, and less prior knowledge is required. We can update the $W[n]$ based on $e[n]$ directly:

$$W[n+1] = W[n] + \mu[n]e[n]x'[n] \tag{7}$$

where $\mu[n]$ controls the step size to update $W[n]$. That's also the connection between the two methods: the Kalman filter can be interpreted as an LMS algorithm with optimal variable step[6].

3.4. Simulation

As for the LMS version, I constructed the control system in Simulink and deployed it in hardware (a real headphone with two external welded microphones). The overall system is depicted in figure 5.

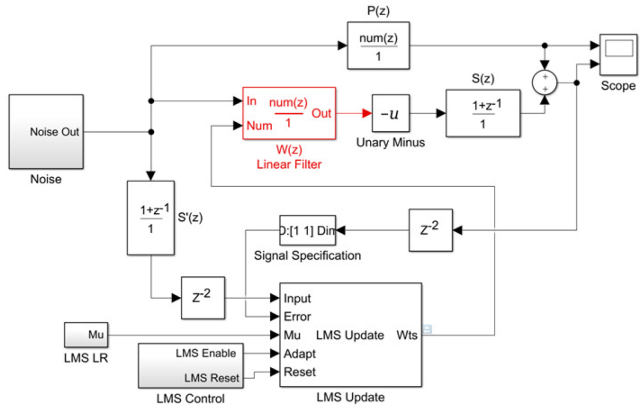

Figure 5. LMS-based ANC system

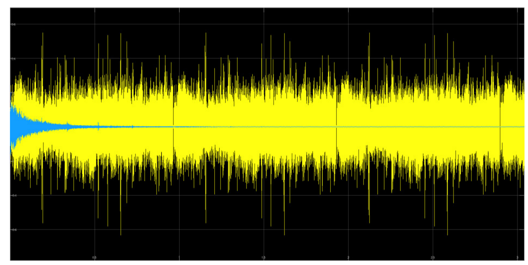

Figure 6. The Simulation Result in Simulink

As for the Kalman filter version, I wrote MATLAB scripts to simply verify the properties. Since the hardware doesn't support accurate offline modeling (no professional DSP development board, but only laptop), no practicable data are available as prior knowledge. Thus, only toy examples are tested.

4. Numerical Results

4.1. LMS Version

The simulation result in Simulink is shown in figure 6. You can see the residual noise (blue part) faded away soon compared to the original noise (yellow part). However, the testing result in hardware is not positive, and here're many practical issues involved, especially the IO latency of the laptop. We'll discuss it in the next section.

4.2. Kalman Filter Version

As we mentioned, only toy examples are tested. Here I set $A[0] = I$, $V = I$, randomly initialize

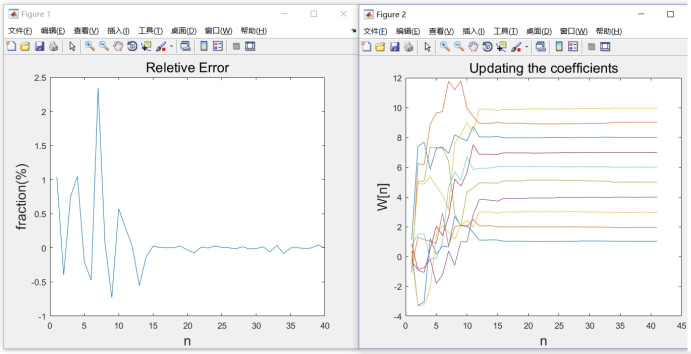

Figure 7. The Simulation Result of Kalman Filter

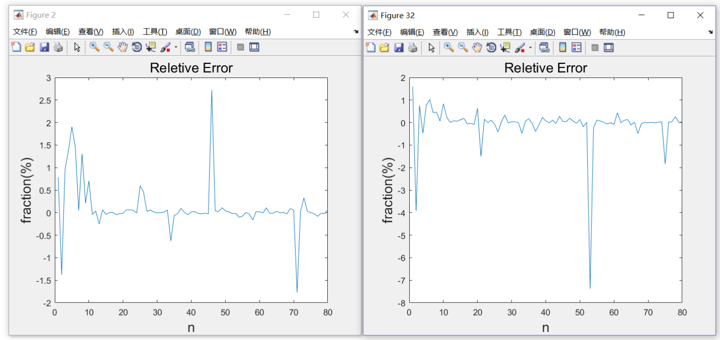

Figure 9. 3 times (left) and 6 times (right) Deviation

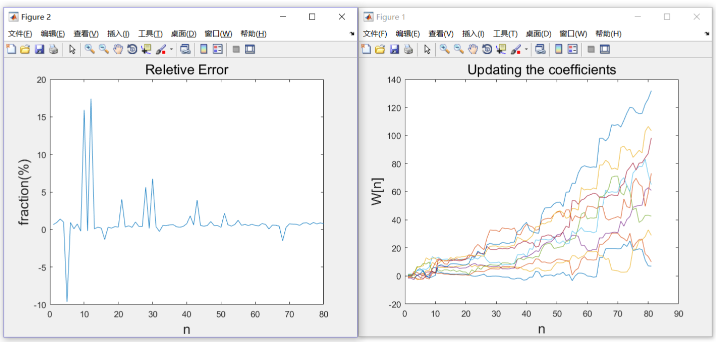

Figure 8. The Simulation Result of Time Varying Case

$W[0] \in \mathcal{N}(0, I)$. For given $P[n]$ and S, we have the nice result shown in figure 7. Here we can see that the system state $W[n]$ converging very fast, which means we obtain the steady states in approximately 15 samples (about $0.5ms$).

Aside from the LTI system case, the Kalman filter can actually handle the time varying case either. As we mentioned:

$$W[n+1] = W[n] \tag{8}$$

Here the previous prediction tell nothing about next one, it's a deficiency of prior knowledge for time varying case. For instance, as for $P[n+1] = 1.05 * P[n]$, the information of changing rate 1.05 is unknown to our control system. However, our controller can also respond quickly for this case, the simulation result is shown in figure 8.

Here the average level of residual noise is about 64% percentage of original noise. However, deficiency of prior knowledge is not worse than the deviation of the known prior knowledge. We introduce 3 times and 6 times deviation of prior knowledge V respectively, the simulation result is shown in figure 9. Note the difference of scale of y-axis, it's clear that Kalman filter is not reliable without good prior knowledge.

5. Additional Discussion

You may notice that the key word "time delay" is not mentioned in this report yet. However, it has been taken into account.The time delay of IO and communication in electronic device can be modeled in S. And the time cost of calculation is very small in modern DSP boards.

Moreover, the causality of the system correlated to time delay is a more critical issue. When the causality condition is met, the ANC system is capable of canceling broad-band random noise: nearly ideal, we only need to emit an anti-noise of equal amplitude and opposite phase once we receive the noise signal. If the latency of IO is large, the causality condition is not satisfiable, the system can effectively control only narrow-band or periodic noise. In general conditions, we always face the second case.

A benchmark for IO latency of my laptop is shown in figure 10. Optimized by ASIO driver, the latency keeps in the level about $0.15ms$. It's a very short time, but still far from the speed of voice.

Since the causality can't be satisfied, we focus on cancelling the narrow-band or periodic noise. Here the requirement towards calculating speed emerges. The speed consists of two parts: time of

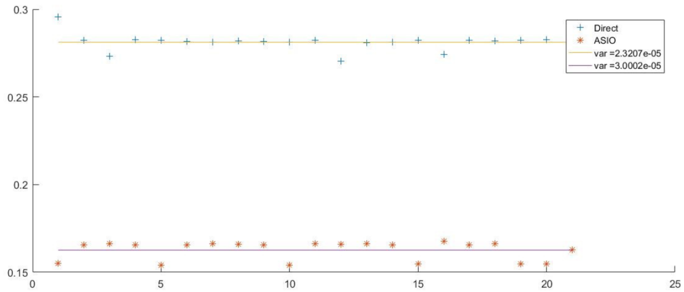

Figure 10. Latency Test Between ASIO and DirectSound

responding to inputs and iteration times to converge. The former directly affects the performance, and the latter decides how well the control system adapts varying environment.

6. Conclusion and Future Work

Nowadays, with the development of DSP technique, the calculation speed is not the bottleneck of the performance of ANC system. More and more commercial products care about adaptability, which encourage many methods with fast convergence were proposed. However, since Kalman filter has been proved to be the optimal (fast) deterministic algorithm, some researchers are trying to overcome its drawbacks.

Recently, a purely data-driven Kalman-like (no prior knowledge but estimating online) algorithm is proposed[4], which achieved a convincing performance. Thus, the direction of future works about ANC should be clear: faster convergence and less prior knowledge relied for robustness.

References

[1] S. M. Kuo and D. Morgan. *Active noise control systems: algorithms and DSP implementations*. John Wiley & Sons, Inc., 1995.

[2] S. M. Kuo and D. R. Morgan. Active noise control: a tutorial review. *Proceedings of the IEEE*, 87(6):943–973, 1999.

[3] S. M. Kuo and D. Vijayan. A secondary path modeling technique for active noise control systems. *IEEE Transactions on Speech and Audio Processing*, 5(4):374–377, 1997.

[4] S. Liebich, J. Fabry, P. Jax, and P. Vary. Time-domain kalman filter for active noise cancellation headphones. In *Signal Processing Conference (EUSIPCO), 2017 25th European*, pages 593–597. IEEE, 2017.

[5] P. A. Lopes and M. S. Piedade. A kalman filter approach to active noise control. In *Signal Processing Conference, 2000 10th European*, pages 1–4. IEEE, 2000.

[6] D. P. Mandic, S. Kanna, and A. G. Constantinides. On the intrinsic relationship between the least mean square and kalman filters [lecture notes]. *IEEE Signal Processing Magazine*, 32(6):117–122, 2015.

[7] B. Widrow. D. shur, and s. In *Shaffer,On adaptive inverse controls, in Proc. 15th Asilomar Conf Circuits, Syst., Comput*, pages 185–189, 1981.