

Summary:

* We have taken FIR filter to model P(z), C(z), S(z), and Sh(z).
* P(z) is the medium from source to sensor
* S(z) is the medium from sensor to hearing point.
* Sh(z) is the estimate of Secondary Path.
* Noise x(k) is propagated from the source to the sensor through the fluid medium P(z). The sensor measures the arriving noise as yp(k).
* To reduce noise, we generate another 'noise' or weights yw(k) using the controller coefficients C(z) - a model of the propagation medium P(z).
* There is also fluid medium S(z) that stay between the actuator and sensor (secondary propagation path(Sh(z).)
* We also compensate the adjustment process using Sh(z) which is an estimate of S(z).
* Least mean square algorithm is applied to adjust the controller coefficient/weight.

We have generated 5000 samples for simulation. We do not know P(z) and S(z) in reality. So we have to make dummy paths. So we have assigned some set of values to Pw. And taken Sw as Sw=Pw\*0.45.

Our first task is to estimate S(z) . So, for that we generate white noise signal of 5000 samples i.e. x\_iden=randn(1,T)

And send it to the actuator to measure it at the sensor position(secondary propagation path.

So the output of inbuilt filter function is y\_iden=filter(Sw, 1, x\_iden)

For the identification process of secondary path, we compute the state of Sh(z) and the weights of Sh(z) for 16 coefficients.

Shx=zeros(1,16)

Shw=zeros(1,16)

And compute the identification error for 5000 samples.

e\_iden=zeros(1,T)

We then apply least mean square algorithm for discrete time k by adjusting the weights and calculating the error recursively to reduce it to get the desired output.

Shy=sum(Shx.\*Shw); % output of Sh(z)

e\_iden(k)=y\_iden(k)-Shy; % error

Shw=Shw+mu\*e\_iden(k)\*Shx; % adjust the weight

We then plot the results for LMS algorithm. We plot the error signal, coefficients of medium S(z) and Sh(z).

Second Task includes implementing FxLMS algorithm to remove the noise(i.e. Active control itself)

Again, we need to simulate the actual condition. In practice, it should be an iterative process of ‘'measure', 'control', and 'adjust'; sample by sample. We generated the random noise of 5000 samples X=randn(1,T);

and measure the arriving noise at the sensor position i.e. Yd=filter(Pw, 1, X);

We compute the state and weights of the controller C(z), dummy state for secondary path, compute control error and the state of filtered noise signal Xhx and initiate the system

% Initiate the system,

Cx=zeros(1,16); % the state of C(z)

Cw=zeros(1,16); % the weight of C(z)

Sx=zeros(size(Sw)); % the dummy state for the secondary path

e\_cont=zeros(1,T); % data buffer for the control error

Xhx=zeros(1,16); % the state of the filtered x(k)

We then apply FxLMS algorithm to compute the secondary path, controller output and filtered x(k) by an iterative process of ‘measure’ , ‘control’ and ‘adjust’ sample by sample.

mu=0.1; % learning rate

for k=1:T, % discrete time k

Cx=[X(k) Cx(1:15)]; % update the controller state

Cy=sum(Cx.\*Cw); % calculate the controller output

Sx=[Cy Sx(1:length(Sx)-1)]; % propagate to secondary path

e\_cont(k)=Yd(k)-sum(Sx.\*Sw); % measure the residue

Shx=[X(k) Shx(1:15)]; % update the state of Sh(z)

Xhx=[sum(Shx.\*Shw) Xhx(1:15)]; % calculate the filtered x(k)

Cw=Cw+mu\*e\_cont(k)\*Xhx; % adjust the controller weight

end