Abstract

For a given filter specification, infinite impulse response (IIR) filters offer faster computation and less memory requirement than finite impulse response (FIR) filters. These advantages come with the complexity of nonlinear phase, coefficient round-off sensitivity, and possible instability. In this laboratory exercise, students use pseudo-code from the prelab exercise to develop a real-time IIR filter as a cascade of second-order sections.

I. Filter design

In the IIR Filtering Pre-lab, Matlab was used to design a sixth-order IIR band-stop filter. The edges of the pass bands are 9600 Hz and 12000 Hz, with stop-band attenuation of 50 dB and no more than 1 dB pass-band ripple. The sampling rate is 48 kHz. For example,

```
Fs = 48000;       % Sampling Frequency
N = 6;            % Filter Order
Fpass1 = 9600;    % First Pass-band Edge Frequency
Fpass2 = 12000;   % Second Pass-band Edge Frequency
Ap = 1;           % Pass-band Ripple (dB)
As = 50;          % Stop-band Attenuation (dB)
% Calculate the transfer function using the ELLIP command.
[B,A] = ellip(N/2, Ap, As, [Fpass1 Fpass2]/(Fs/2), 'stop');
freqz(B,A);%display magnitude (dB) and phase
```

II. Real-time implementation

From the pseudo-code prepared for the pre-lab assignment, complete an assembly language implementation of a generic IIR filter using cascaded second-order sections.

A speech signal has been recorded in the presence of a loud, annoying, high-frequency hum. Implement your IIR notch filter on the TMS320C55x and verify its operation using the oscilloscope and speakers.

1. What is the 3dB cut-off frequency for the filter implemented in assembly code? Make a sketch of the experimentally observed magnitude response. Compare qualitatively with the predicted $H(e^{j\omega})$ plotted in the pre-lab assignment.

2. Describe the perceptual effect of your notch filter on the recorded speech.