

DFiltMPFIR

This program designs minimum-phase filters. It calls DFiltFIR (available as a separate package) to design a double-length filter. The resulting linear phase filter is then factored into a minimum-phase and a maximum-phase part.

Features

- This routine calls DFiltFIR to design a double length filter. DFiltFIR accepts limits on the response. The limits are set to force the response to be positive. This results in double-order stopband zeros. Being a double-length linear-phase filter with double-order stopband zeros allows it to be factored into the product of a minimum-phase and a maximum-phase filter. The component filters have the same magnitude response.
- DFiltMPFIR factors the filter into the component parts. It uses the locations of the extrema in the stopband (outputs from DFiltFIR) to identify the (double-order) stopband zeros.
- The roots of the double-length are found using the polynomial root finder routine roots. The roots corresponding to the stopband roots are removed – the root information from DFiltFIR is more accurate than the information from the root finder.
- The remaining roots are factored into inside/outside (relative to the unit circle) roots. This segregates the zeros into those for the filter factors. Details appear in [1]
- Given the roots for the minimum-phase factor as determined above, a new procedure is used to find the filter coefficients. The frequency response corresponding the roots is carefully calculated to ensure good precision. The inverse Discrete Fourier Transform is used to find the filter coefficients. This procedure is well-suited to filter design – an accurate frequency response can be inverted accurately using the inverse DFT. Tests show that this procedure gives as good or better results than a reconstruction procedure based on the function `leja` (in the `filterdesign` toolbox). Details of the procedure used in DFiltMPFIR are given in [1].
- The frequencies, desired values, and weights in the design are specified for the double-length filter. The relationship between the parameters of the double-length filter and those of the minimum-phase factor are somewhat complicated and since they depend on the stopband attenuation of the double-length filter, cannot be specified *a priori*. Full details of the parameter relationships appear in an appendix in [1].

Example Design

Three lowpass filters were designed (sampling frequency 16 000 Hz) with a passband edge at 3850 Hz and a stopband starting at 4140 Hz. The first is a 101 coefficient linear-phase FIR filter. The next is a 101 coefficient minimum-phase filter. This filter has a better amplitude response than the linear-phase filter. This can be attributed to the fact that the linear phase filter has only 51 degrees of freedom due to the symmetry imposed by the linear-phase requirement. After some experimentation, it was determined that an 85 coefficient minimum-phase filter has about the same magnitude response as the 101 coefficient linear-phase filter. This is the third filter. The Matlab commands to design this filter are shown below.

```

NCof = 85;
SFreq = 16000;
fcP = 3850;
fcS = 4150;
[B(1:2).Freq] = deal ([0 fcP], [fcS SFreq/2]);
[B(1:2).Value] = deal (1, 0);
[B(1:2).Weight] = deal (1, 900*[1 10].^2);
[B(1:2).WeightInt] = deal ('linear', 'sqrt');
h3 = DFiltMPFIR (NCof, B, SFreq);

```

Figure 1 shows the group delay for the three filters. We see that the minimum-designs have a much smaller group delay in the passband than the linear-phase design.

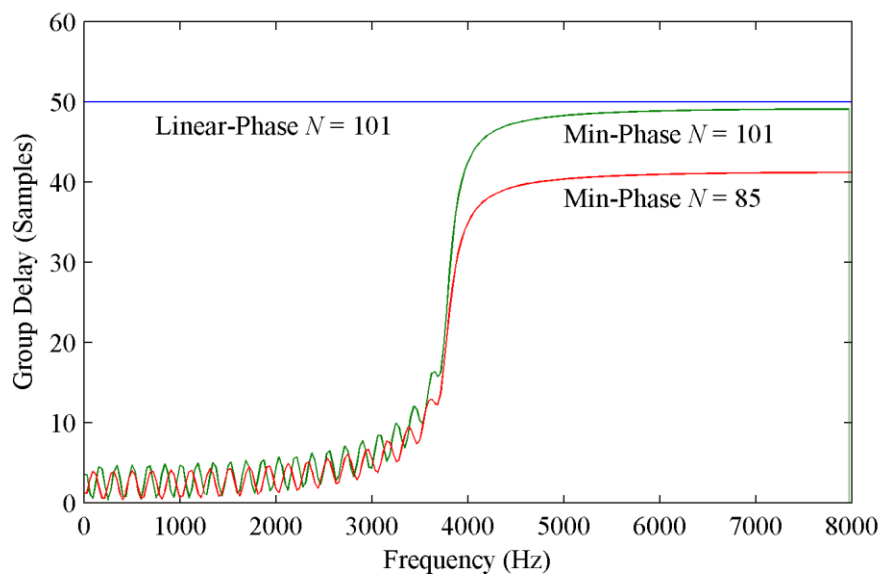


Figure 1 Group delay for three filter designs

Reference

1. P. Kabal, "FIR Filters: Frequency-Weighted and Minimum-Phase Designs", Technical Report, Electrical & Computer Engineering, McGill University, Nov. 2007. (available as part of the DFiltMPFIR package or on-line at www-mmsp.ece.mcgill.ca/MMSP/Documents)