Lab 12 – IIR filter design

12.1 Notch filter

A notch filter passes most frequencies virtually unchanged but has zero frequency response at $\omega = \omega_0$. An easy way to do this is to place two zeros at $e^{\pm j\omega_0}$, and then place two poles at $re^{\pm j\omega_0}$, where r is close to 1. Write a function `notch()` that takes as inputs

- w0, the desired notch frequency w0 (in radians), and
- r, a value close to 1

and returns

- b, the numerator (a 1 x 3 vector) for the digital notch filter
- a, the denominator (a 1 x 3 vector) for the digital notch filter.

The filter should be normalized to have magnitude 1 at frequency $\omega = 0$.

```
function [b,a] = notch(w0,r)
 % Compute denominator polynomial from r and w0
 a = ...;
 % Compute numerator polynomial from w0
 b = ...;
 % Determine gain so that frequency response has magnitude 1 at w0
 b = b*...;
 % Diagnostic display
 freqz(b,a);
end
```

(a) Write the Z-Transform expression for this filter and identify vectors b and a above.
(b) How does the filter response change with radius r?
(c) Use `fvtool()` to visualize various aspects of the filter.
(d) Generate a signal consisting of sinusoids of frequency ω_0 and $2\omega_0$ and check whether your notch filter is working. Use the inbuilt matlab command `filter()` for this purpose.

12.2 Fibonacci sequence

The Each element of the Fibonacci sequence is the sum of the previous two elements, i.e., 1, 1, 2, 3, 5, 8, ... This relationship can be captured by the simple difference equation $y[n] = y[n-1] + y[n-2]$, where the input to the system is the unit impulse. That is, the Fibonacci sequence is the (infinite) impulse response of this digital filter. Write a

function `fibonacci_iir()` that takes as input a number N, an integer and returns the outputs

- fib, a vector containing the first N terms of the Fibonacci sequence
- b, the numerator of the transfer function (in terms of descending powers of *z^-1*)
- a, the denominator of the transfer function (in terms of descending powers of *z^-1*)

Use the given code template and answer the following.

(a) Modify the given difference equation to include the input signal $x[n]$ such that when the input is an impulse, the output is the Fibonacci sequence.
(b) For this system difference equation, identify its Z-transform and hence the vectors 'b' and 'a'.
(c) The function should produce fib using either the `filter()` or `impz()` built-in commands, the computed b and a, and an appropriate-length impulse.
(d) What can say about the stability of this filter? Verify your answer using `zplane()`.

```
function [fib,b,a] = fibonacci_iir(n)
  % Determine IIR filter that produces Fibonacci sequence
  b = ...
  a = ...
  % Create length-n impulse to drive filter
  imp = ...
  % Determine impulse response
  fib = filter(...);
end
```

12.3 Filter design using filterDesigner

This exercise is meant to familiarize you with the filter design tool available in matlab. Type `filterDesigner` in matlab command prompt and press enter to open its graphical interface. Explore the various filter design parameters available there and relate these to the parameters discussed in the class.

(a) Design a low-pass FIR *Equiripple* filter with the following specifications: Passband attenuation 1 dB, Stopband attenuation 80 dB, A passband frequency 0:2 [Normalized (0 to 1)], stopband frequency 0:5 [Normalized (0 to 1)].
(b) Use the tool to observe the magnitude response, phase response, impulse response, and pole-zero plot of the designed filter.
(c) Change the design method to *least-squares* and compare the obtained filter with the one obtained using Equiripple method.
(d) Similarly design a low-pass IIR filter with the above specifications. Compare qualitatively the filters obtained using the filter design methods of Butterworth, Chebyshev, and Elliptic. Observe the magnitude response, phase response, impulse response, and pole-zero plot of the designed filters.
(e) Import the 'a' and 'b' parameters of any one of the IIR filters obtained by this design tool and use it to perform filtering of the signal generated in part 12.1 above.