Filter Design Results

Generated by: http://www-users.cs.york.ac.uk/~fisher/mkfilter

Summary

You specified the following parameters:

filtertype	= Butterworth
passtype	= Bandpass
ripple	=
order	= 1
samplerate	= 8000
corner1	= 2000
corner2	= 2500
adzero	=
logmin	=

Results

```
Command line: /www/usr/fisher/helpers/mkfilter -Bu -Bp -o 1 -a 2.5000000000e-01 3.1250000000e-01
raw alpha1    =   0.2500000000
raw alpha2    =   0.3125000000
warped alpha1 =   0.3183098862
warped alpha2 =   0.4763844100
gain at dc    :   mag = 0.000000000e+00
gain at centre:   mag = 6.026183138e+00   phase =   0.0062352617 pi
gain at hf    :   mag = 0.000000000e+00

S-plane zeros:
          0.0000000000 + j   0.0000000000

S-plane poles:
         -0.4966057627 + j   2.3957891742
         -0.4966057627 + j  -2.3957891742

Z-plane zeros:
          1.0000000000 + j   0.0000000000
         -1.0000000000 + j   0.0000000000

Z-plane poles:
         -0.1659106810 + j   0.8004075736
         -0.1659106810 + j  -0.8004075736

Recurrence relation:
y[n] = ( -1 * x[n- 2])
     + (  0 * x[n- 1])
     + (  1 * x[n- 0])

     + ( -0.6681786379 * y[n- 2])
     + ( -0.3318213621 * y[n- 1])
```

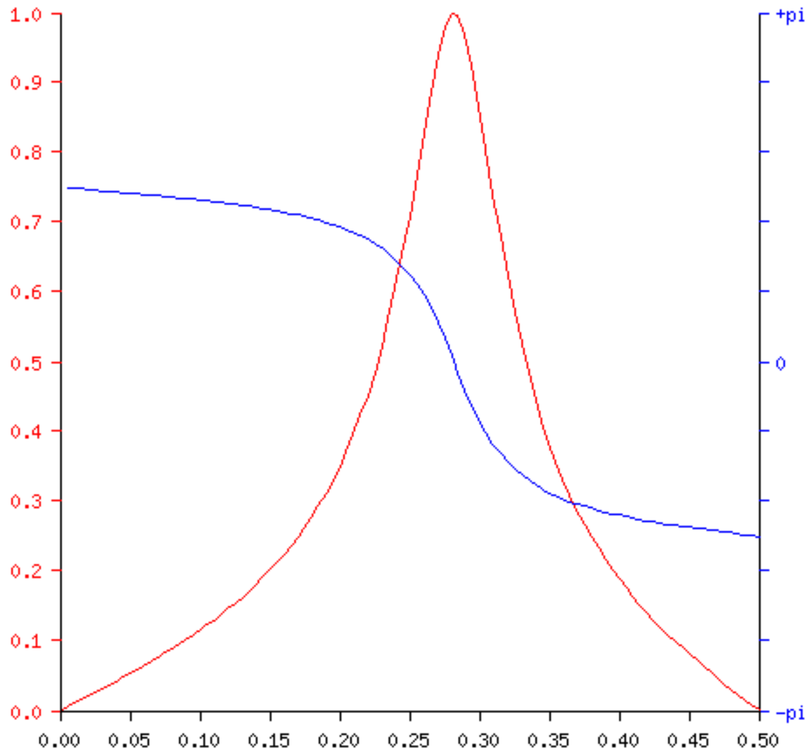
Ansi ``C'' Code

```
/* Digital filter designed by mkfilter/mkshape/gencode   A.J. Fisher
   Command line: /www/usr/fisher/helpers/mkfilter -Bu -Bp -o 1 -a 2.5000000000e-01 3.1250000000e-01 -l */

#define NZEROS 2
#define NPOLES 2
#define GAIN   6.026183138e+00

static float xv[NZEROS+1], yv[NPOLES+1];

static void filterloop()
  { for (;;)
      { xv[0] = xv[1]; xv[1] = xv[2];
        xv[2] = next input value / GAIN;
        yv[0] = yv[1]; yv[1] = yv[2];
        yv[2] =   (xv[2] - xv[0])
                     + ( -0.6681786379 * yv[0]) + ( -0.3318213621 * yv[1]);
        next output value = yv[2];
      }
  }
```

Download code and/or coefficients: TERSE VERBOSE

Magnitude (red) and phase (blue) vs. frequency

- *x* axis: frequency, as a fraction of the sampling rate (i.e. 0.5 represents the Nyquist frequency, which is 4000 Hz)
- *y* axis (red): magnitude (linear, normalized)
- *y* axis (blue): phase

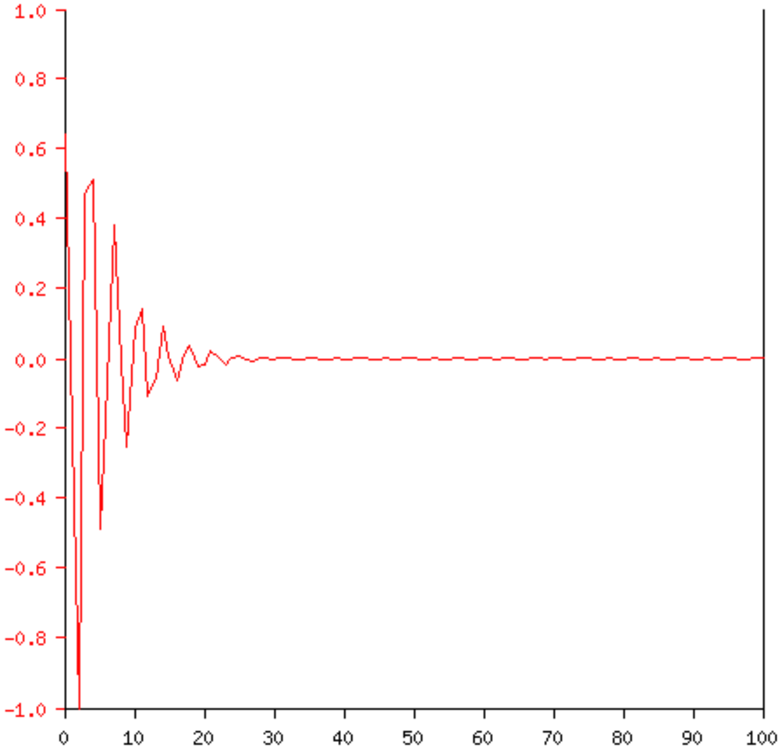


For an expanded view, enter frequency limits (as a fraction of the sampling rate) here:

Lower limit: Upper limit: zoom

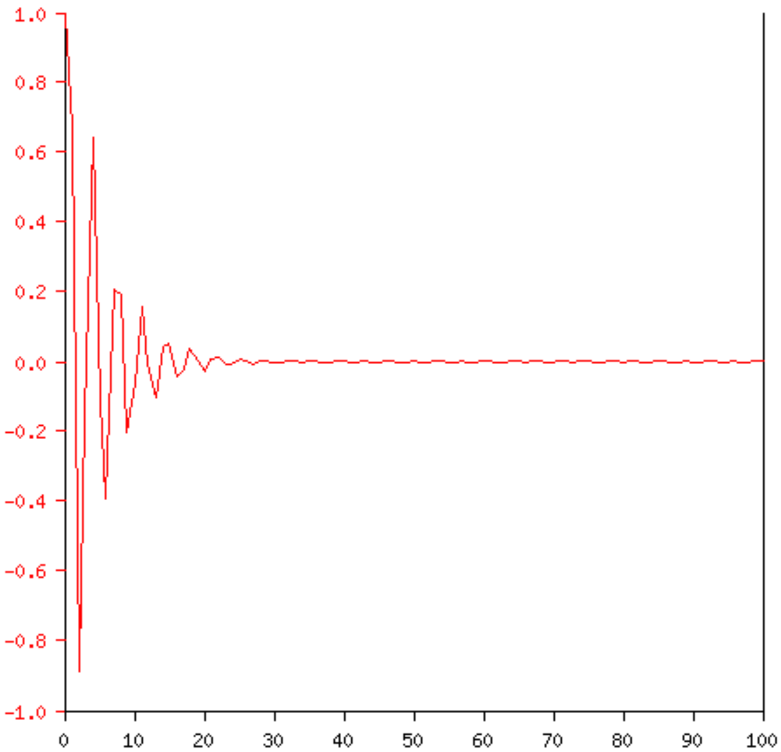
Impulse response

- *x* axis: time, in samples (i.e. 8000 represents 1 second)
- *y* axis (red): filter response (linear, normalized)



Step response

- *x* axis: time, in samples (i.e. 8000 represents 1 second)
- *y* axis (red): filter response (linear, normalized)



For a view on a different scale, enter upper time limit (integer number of samples) here:

Upper limit: zoom

Tony Fisher *fisher@minster.york.ac.uk*