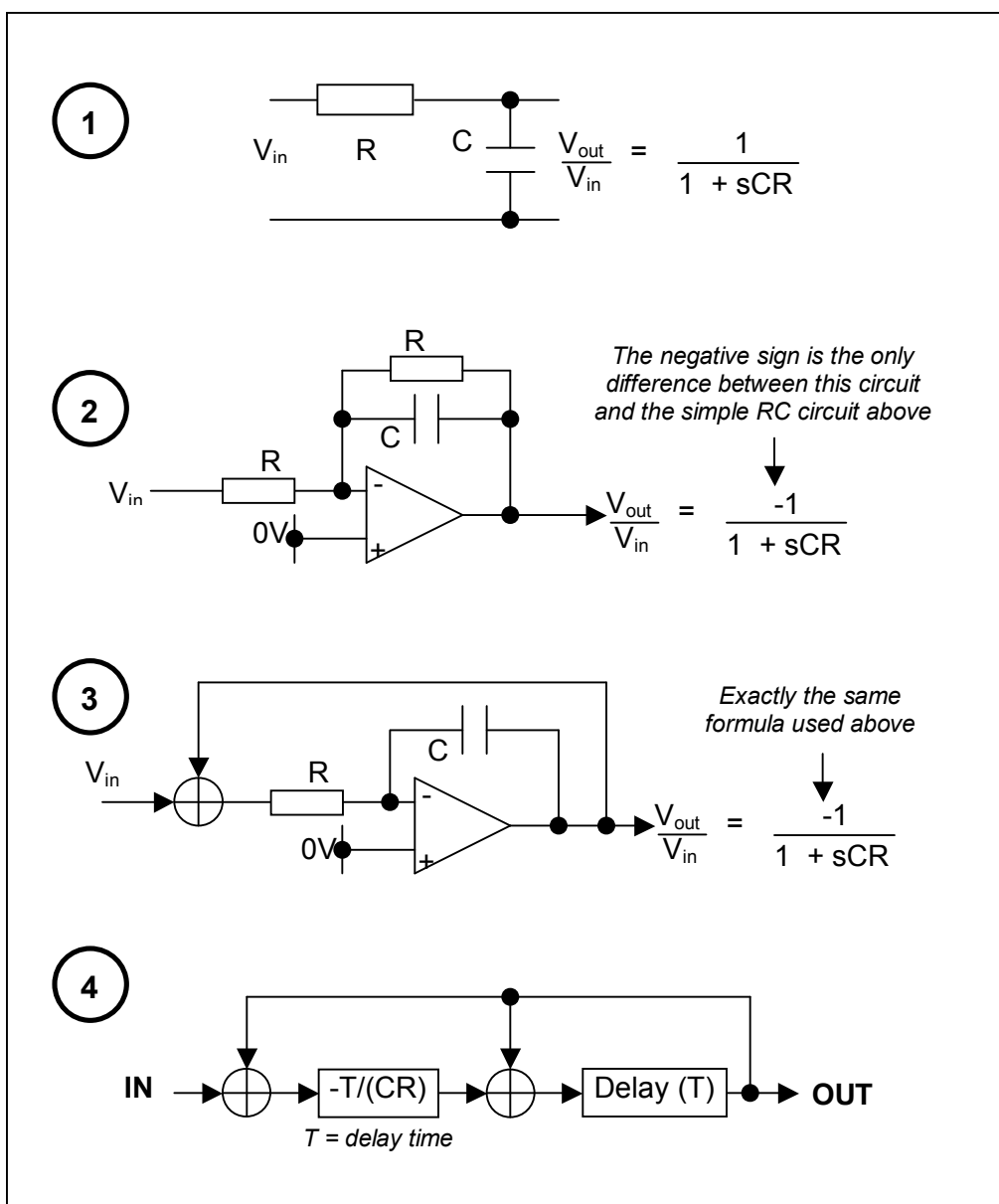


SIMPLE DIGITAL FILTER DESIGN FOR ANALOGUE ENGINEERS



Please note that this document is evolving and some chapters may be rewritten or added from time to time – *June 2010, corrections and amendments to June 2008 version.*

Introduction

If I'd read this document years ago it would have saved me hours of agony and confusion understanding what should be conceptually very simple. My biggest discontent with topical literature is that it never cleanly relates digital filters to "real world" analogue filters.

This document sets out to: -

- Describe the function of a very basic RC low pass analogue filter
- Convert this filter to an analogue circuit that is digitally realizable
- Develop a basic 1st order digital low-pass filter
- Develop a 2nd order filter with independent Q and frequency control parameters
- Demonstrate digital filters using a spreadsheet such as excel
- Explain the limitations of use of digital filters i.e. instability and frequency warping

Before the end of this article you should hopefully find something useful to guide you on future designs and allow you to successfully implement a digital filter.

This document is aimed at analogue engineers and uses analogue terminology. If you are not familiar with electronic design this article may not work for you. Concepts that should ideally already be appreciated by the reader are: -

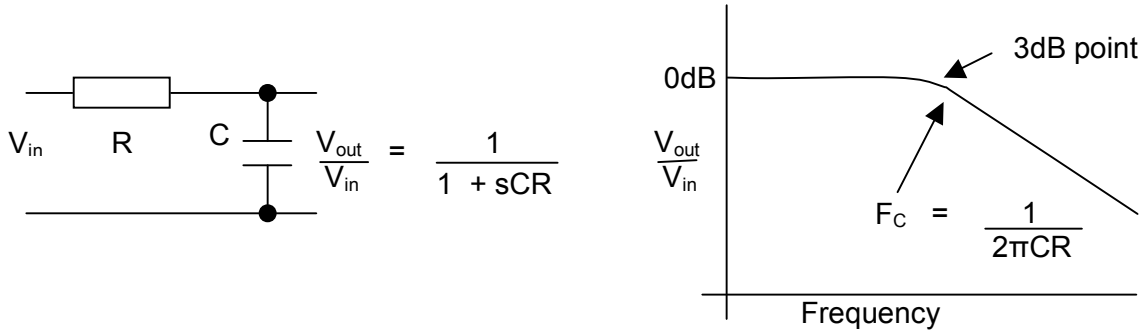
- Resistor-capacitor and op-amp filters
- Analogue integrators
- 2nd order high-pass and low-pass filter design
- Laplace transforms (not too much!!)
- A little bit of control theory
- A little bit about microprocessors and sampling analogue signals
- A little bit about Excel (if you want to design a digital filter)

In the 2010 release a couple of errors were corrected on page 5. These errors relate to formulas that were incorrectly written. A vertical line in the left paragraph specifically indicates the section. Also, I've tried to make things a little clearer – having just re-read it (May 2010) over two years after the original, some parts were not all that clear so I've tried to improve the text.

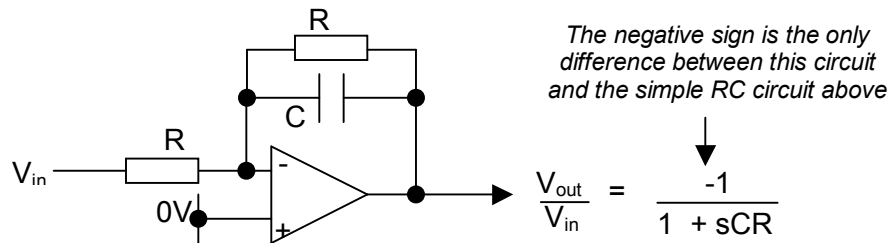
andy@jamesoakwood.co.uk

Simple low-pass filter

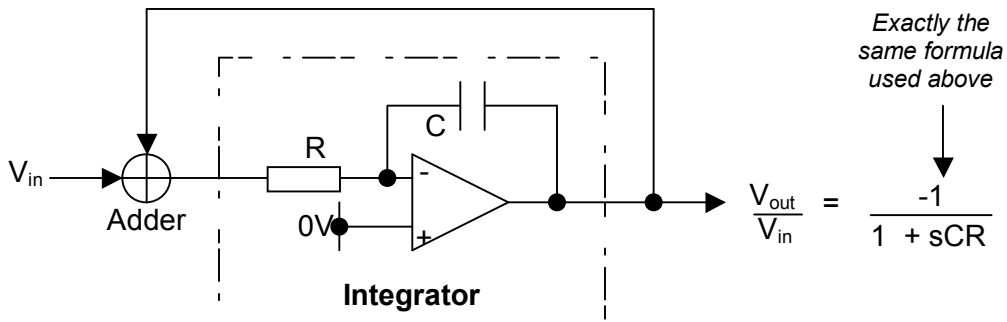
Take the simple low-pass RC filter: -



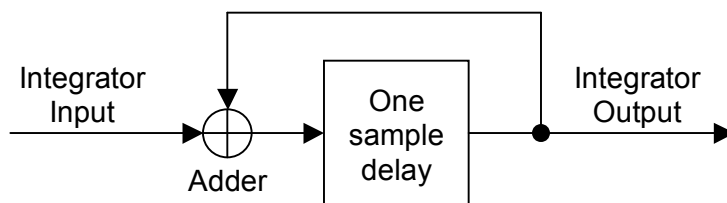
It is equivalent to the op-amp filter below: -



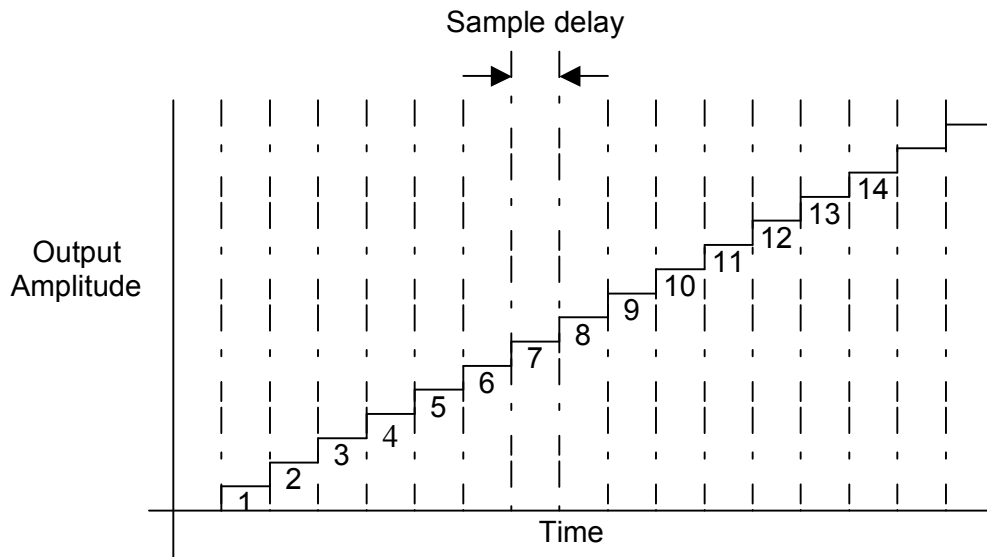
Now segregate R and C feedback paths: -



Implementing an *integrator* in a microprocessor is simple: -

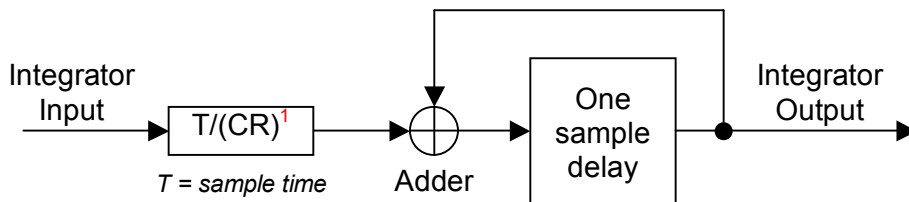


If input signal were constant the output would ramp like this: -

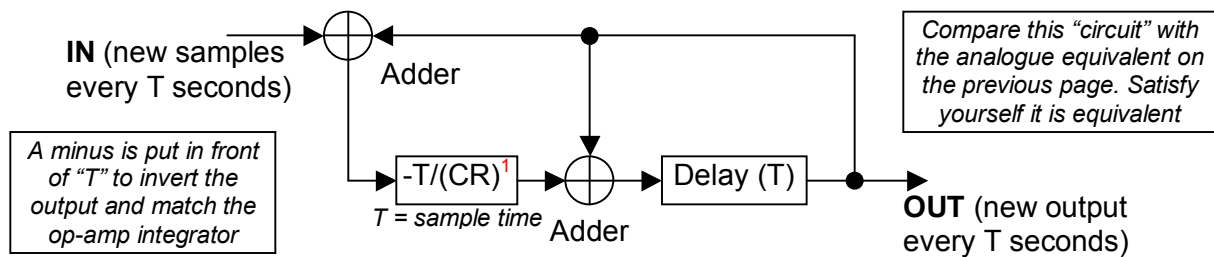


Output slope is dictated by sample delay time (CR time for an analogue integrator). If CR time were 1ms, an equivalent digital-integrator of this type would use a sample time of 1ms.

If the digital-integrator input were attenuated by 1000 it would ramp at one-thousandth the rate. If the sample rate were 1kHz it would be equivalent to a CR time of 1.00. Conclusion: -

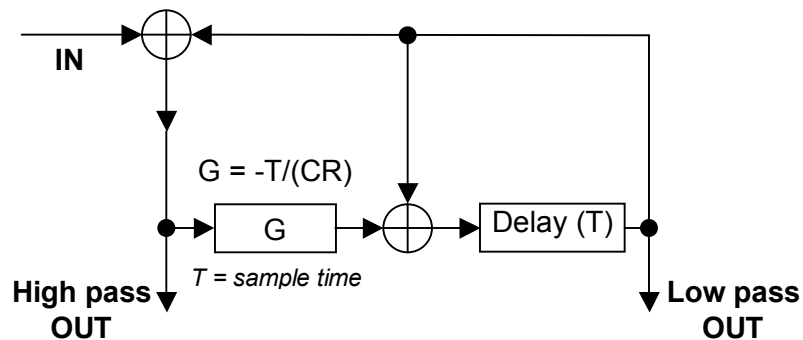


If you want a digital-integrator to perform like an analogue RC integrator, attenuate the input by $T/(CR)^1$ where T is the sample time in seconds. This leads us to a digital representation of the simple analogue low-pass circuit: -



¹ When digital integrators are used in filters an effect called frequency warping takes place that affects the accuracy of the formula particularly when the desired cut-off frequency approaches that of the sampling frequency. This is discussed later but for now assume that T/CR holds true for most integrator and filter designs.

Bonus – there are two useful outputs from this filter: -



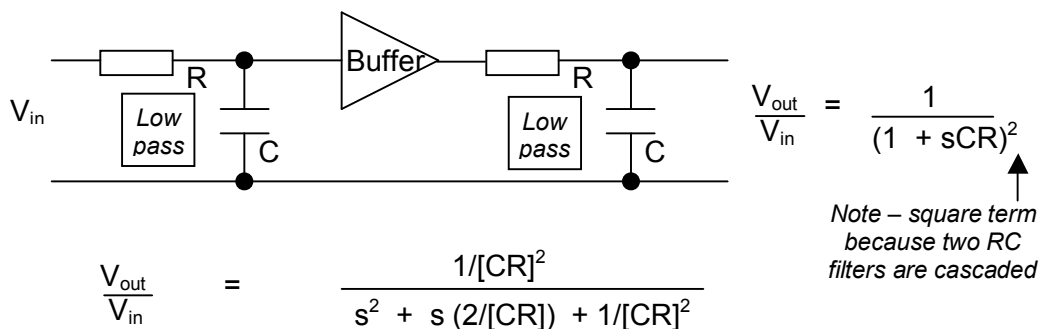
If a filter was required to have a cut-off at 10Hz calculate CR as if it were an analogue filter: -

$$CR = \frac{1}{2\pi F} \quad (CR = 0.0159 \text{ for a } 10\text{Hz cut off})$$

For 1ms sampling, $G = -0.063$. This applies to both high pass and low pass outputs.

2nd order Low-pass filter

Cascading two low-pass RC filters produces a 2nd order filter as shown below²: -



“Theory” informs us the above formula is of the form: -

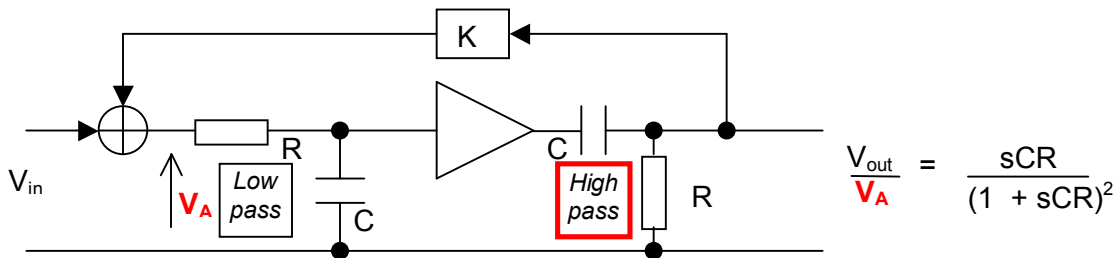
$$\frac{V_{out}}{V_{in}} = \frac{[\omega_n]^2}{s^2 + s(2\zeta\omega_n) + [\omega_n]^2}$$

ω_n – undamped resonant frequency
 ζ – damping ratio (1/2Q)

Clearly there is no Zeta (ζ) term when two RC stages are cascaded. This means Q is fixed *but* we need a design where Q can be varied and done so independently of frequency...

² Oops, a couple of formula errors occurred in the section – corrected for June 2010

A 2nd approach applies feedback (via K) to the input but this time the 2nd RC filter is made into a high pass type: -



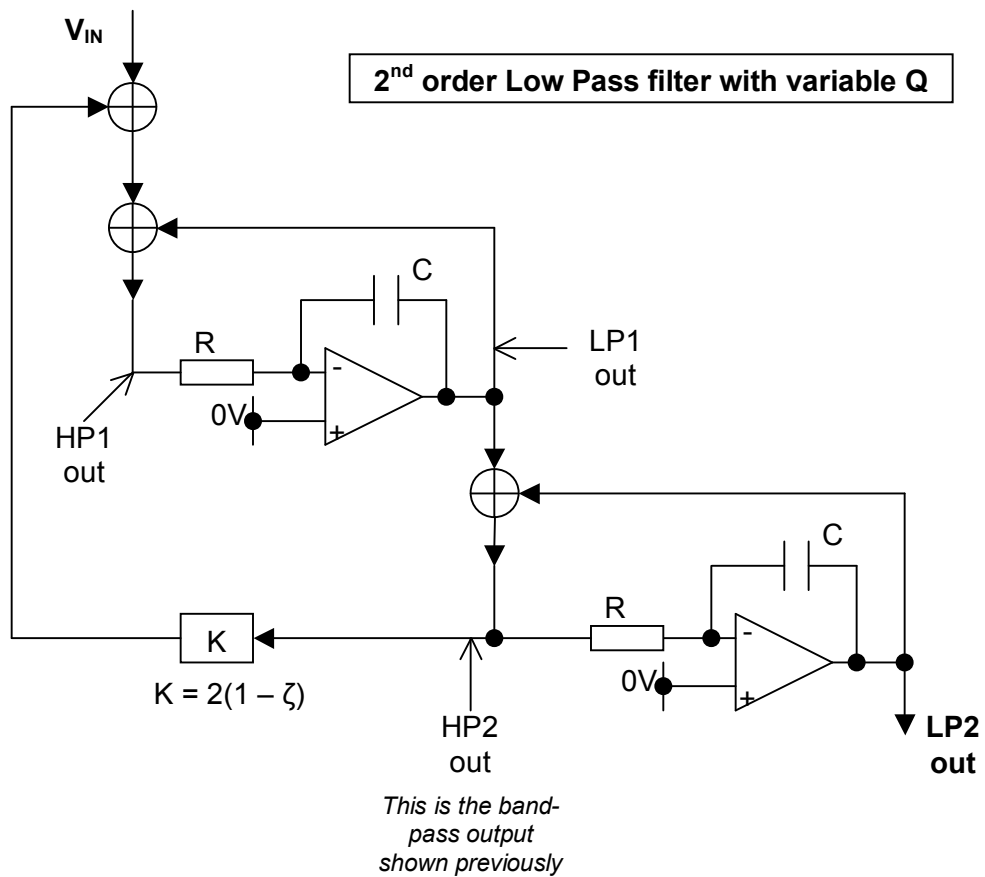
Because $V_A = V_{in} + K V_{out}$ it can be shown that: -

$$\frac{V_{out}}{V_{in}} = \frac{s/(CR)}{s^2 + s(2-K)/(CR) + 1/(CR)^2}$$

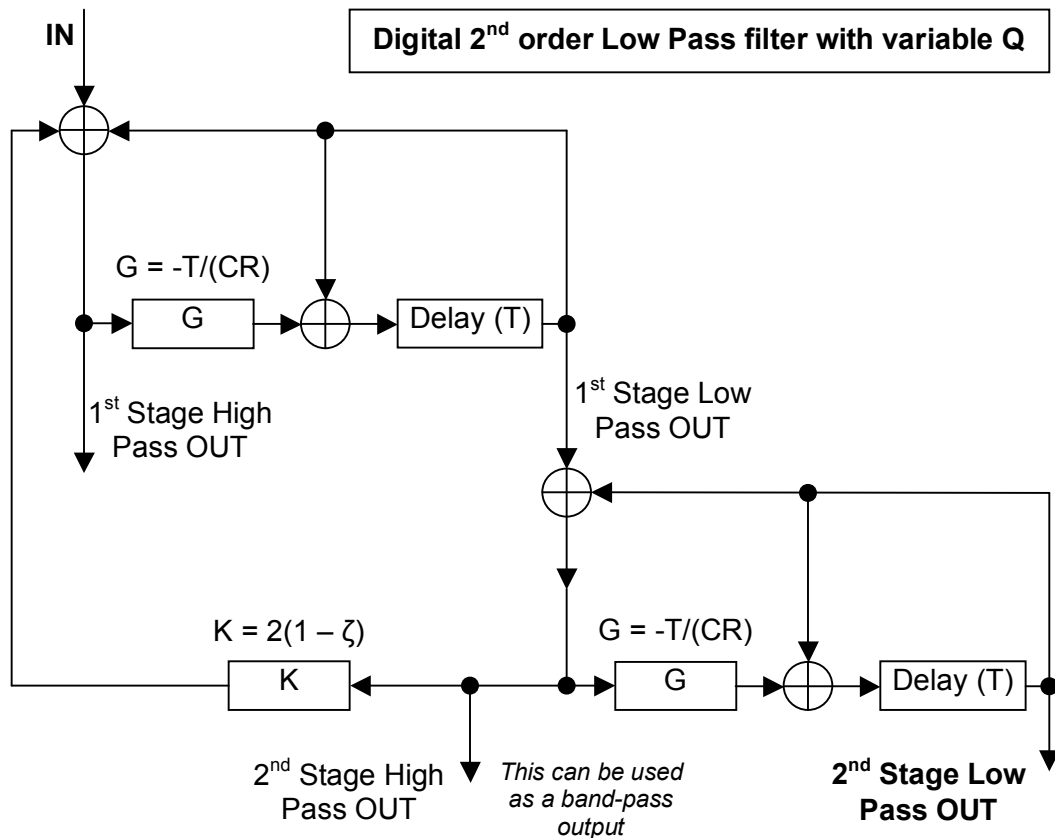
NB This is a band-pass output

This time, ζ (zeta) = 1 - K/2 and importantly ζ is independent of frequency.

Here's the analogue implementation using integrators: -



Here's the digital implementation: -



The usual Digital Filter limitations

If the demanded cut-off frequency (F_c) is too high compared to the sample rate, instability occurs. Below are example figures where instability occurs at a 1kHz sample rate: -

- If $\zeta = 1.00$ ($K = 0.0$), filter becomes unstable when F_c is 318Hz or $G=2$
- If $\zeta = 0.50$ ($K = 1.0$), filter becomes unstable when F_c is 159Hz or $G=1$
- If $\zeta = 0.25$ ($K = 1.5$), filter becomes unstable when F_c is 80Hz or $G=0.5$
- If $\zeta = 0.125$ ($K = 1.75$), filter becomes unstable when F_c is 40Hz or $G=0.25$

A similar picture arises when zeta is higher than 1: -

- If $\zeta = 1.5$ ($K = -1$), filter becomes unstable when F_c is 121Hz or $G=0.764$
- If $\zeta = 2.0$ ($K = -2$), filter becomes unstable when F_c is 85Hz or $G=0.536$
- If $\zeta = 3.0$ ($K = -4$), filter becomes unstable when F_c is 54Hz or $G=0.343$
- If $\zeta = 4.0$ ($K = -6$), filter becomes unstable when F_c is 40Hz or $G=0.254$

However there is a technique that can mitigate this drawback: -

Virtual sampling

If the physical sample rate cannot be raised then raise the virtual sampling rate; create new samples halfway between real samples. This doubles the cut-off frequency attainable.

All samples need to be processed and this of course increases overhead. On the example above, the 40Hz limitation for a damping ratio of 0.125 can be raised to 80Hz. If this is not sufficient, double-up again. If you can't be bothered to calculate mid-point samples then reuse the current sample!

Frequency Warping

On page 4 the digital integrator was first discussed and it was explained that $G = -T/CR$ with the proviso that if the desired cut-off frequency was significantly close to the sample frequency an effect called Frequency Warping meant the formula no longer held its accuracy. Frequency warping has the effect of reshaping the formula for G as follows: -

G (originally) was shown to be T/CR , which equals $2 \pi T F_c$

The frequency warped (and correct version) is $G = 2 \text{Sin} (\pi T F_c)$

So how much does the “correction” affect the design? See below: -

F_c (sample frequency = 1k)	G = 2πT F_c	G = 2 Sin (π F_c T)	Error
1 Hz	0.0031415927	0.0031415914	-
2 Hz	0.0062831853	0.0062831750	-
5 Hz	0.0157079633	0.0157078018	-
10 Hz	0.0314159265	0.0314146346	-
20 Hz	0.0628215182	0.0628318531	0.02%
50 Hz	0.1570796327	0.1569181915	0.1%
100 Hz	0.3141592654	0.3128689301	0.4%
200 Hz	0.6283185307	0.6180339887	1.7%

The table above is for a 1kHz sample rate with the desired cut-off frequency rising from 1Hz to 200Hz. Clearly, at cut-off frequencies below 50Hz the error is small.

Sin calculations are not trivial: -

$\text{Sin } x = x - x^3/3! + x^5/5! - x^7/7! + \dots$ and when x is small $\text{Sin } x$ is easily approximated to x .

If $x = 0.1$, $\text{Sin } x = 0.09983$ i.e. 0.17% error.

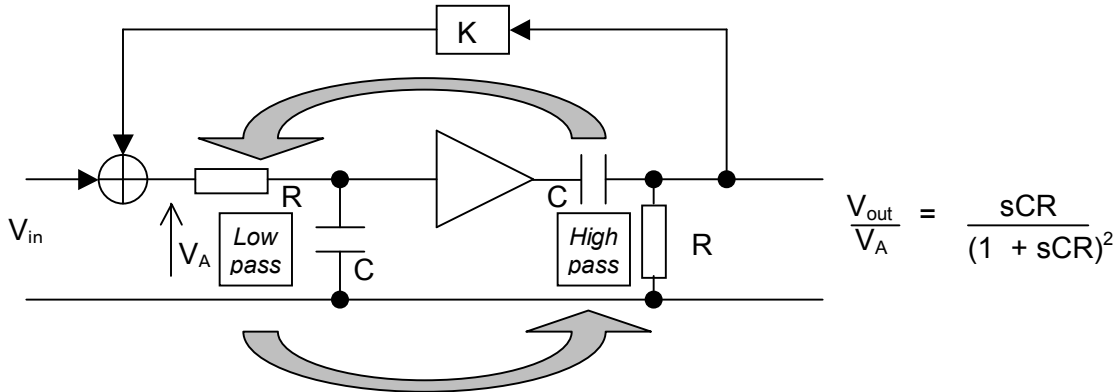
The choice is yours. If you require pinpoint filter accuracy calculate G using several terms of the sine equation. If not, use the simpler equation (especially if the desired cut-off frequency is low compared to the sample rate).

So, what is frequency warping all about – what causes it? Without going into the mathematics, a digital integrator is an approximation to a true analogue integrator and this approximation becomes flaky as the desired cut-off gets closer to the sample rate. Indeed, the instability of the filter (previous page) is due to this flakiness.

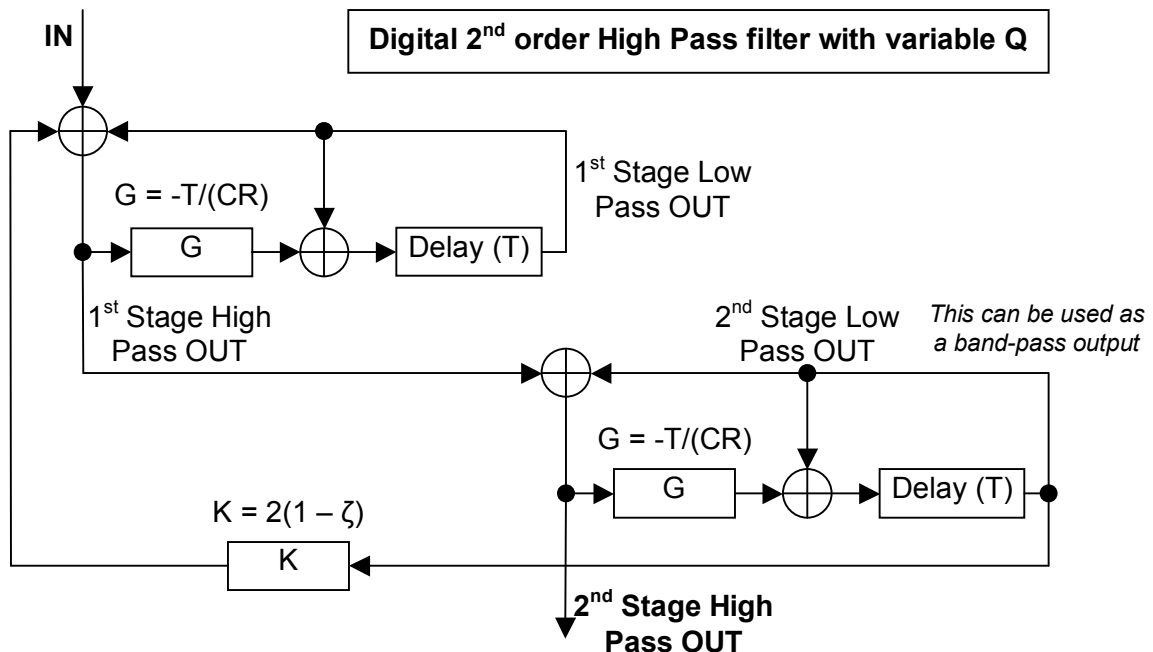
Virtual sampling reduces flakiness and improves the accuracy of the simple equation for G .

2nd order High-Pass filter

Swap the low pass and high pass sections of the 2nd order low pass filter previously described: -



Swapping the sections does not change to the formula (previously derived on page 6). It just means that “K” feedback is from the 2nd stage LP output of the digital implementation: -



Summary

- A simple low-pass RC filter was shown equivalent to a circuit using an integrator.
- An analogue integrator is easily realizable in a digital environment
- A Digital LP filter with 10Hz cut-off was designed
- A 2nd order LP filter was designed by cascading two 1st order filters
- A modified version was derived allowing Q to be independently controlled
- This design was extended to 2nd order HP filters
- Digital filter limitations (instability and frequency warping) were covered

Realizing these filters using excel

I used excel to verify the performance of 2nd order LP filter. This is what I did: -

- In column A I created a sequence of integers going from 0 to over 1000
- In column B I converted the sequence of integers to a Sinewave
- Column B understood the frequency of Sinewave I desired and the sampling rate

Column_B = SIN (B\$2*2*PI (*)A\$2*A6/1000) – this column becomes the input to the filter

Where B\$2 is the desired frequency of the Sinewave

And, A\$2 is the sample delay time in milli-seconds (Note the “divide by 1000” in the formula)

A6 is the relevant integer in Column_A used to calculate the Sinewave sample value

Next I created a series of columns called: -

LPOUT1 HPOUT1 LPOUT2 HPOUT2 MODK

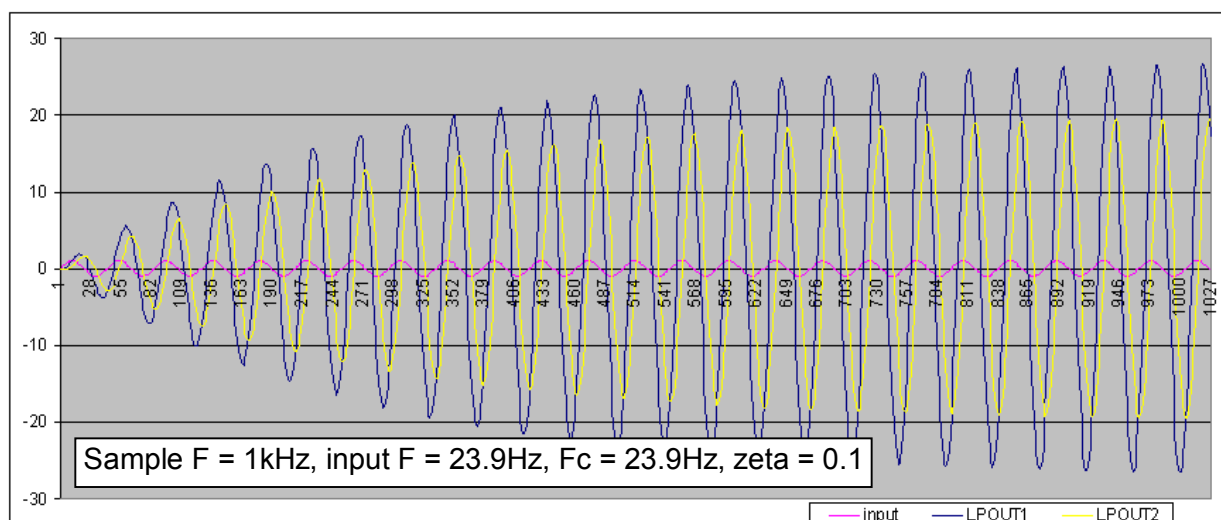
Here’s the formula for each column: -

Column	Top row formulae	All other row formulae
LPOUT1	0	LPOUT1 _{OLD} – G * HPOUT1 _{OLD}
HPOUT1	INPUT – LPOUT1 + MODK	INPUT – LPOUT1 + MODK
LPOUT2	0	LPOUT2 _{OLD} – G * HPOUT2 _{OLD}
HPOUT2	LPOUT1 – LPOUT2	LPOUT1 – LPOUT2
MODK	HPOUT2 * K	HPOUT2 * K

G and K in the above table are as previously described i.e. :-

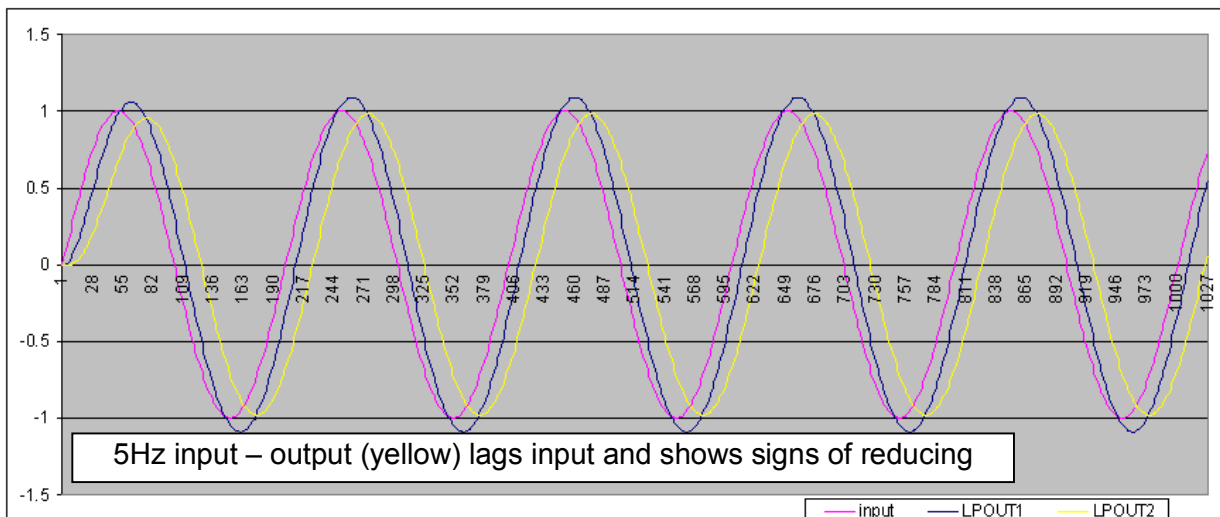
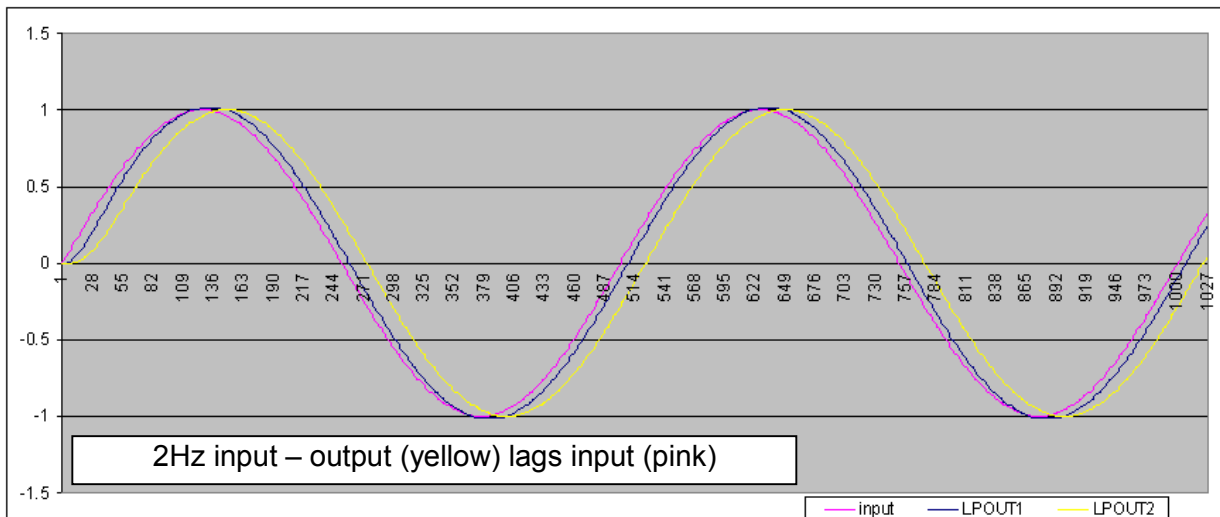
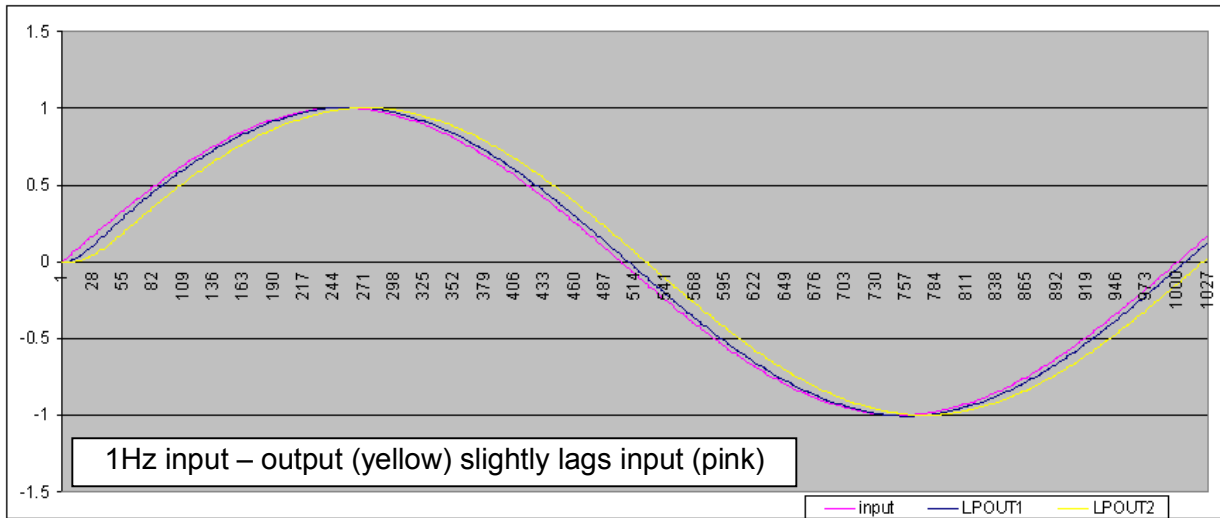
$G = T/(CR)$ or $T \cdot 2 \cdot \pi \cdot F_c$ and $K = 2(1 - \zeta)$ or $2 - 1/Q$

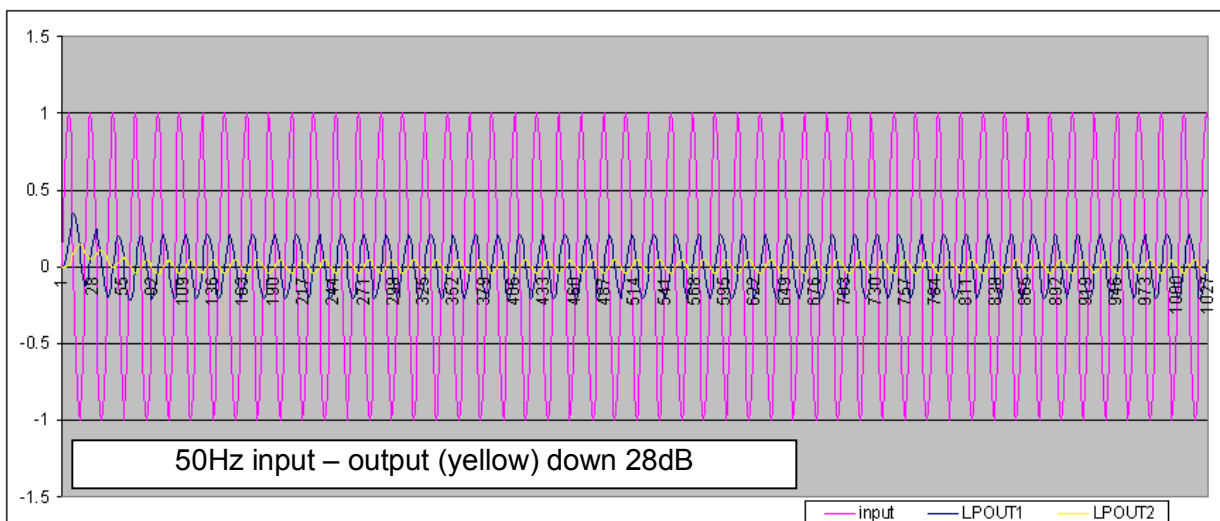
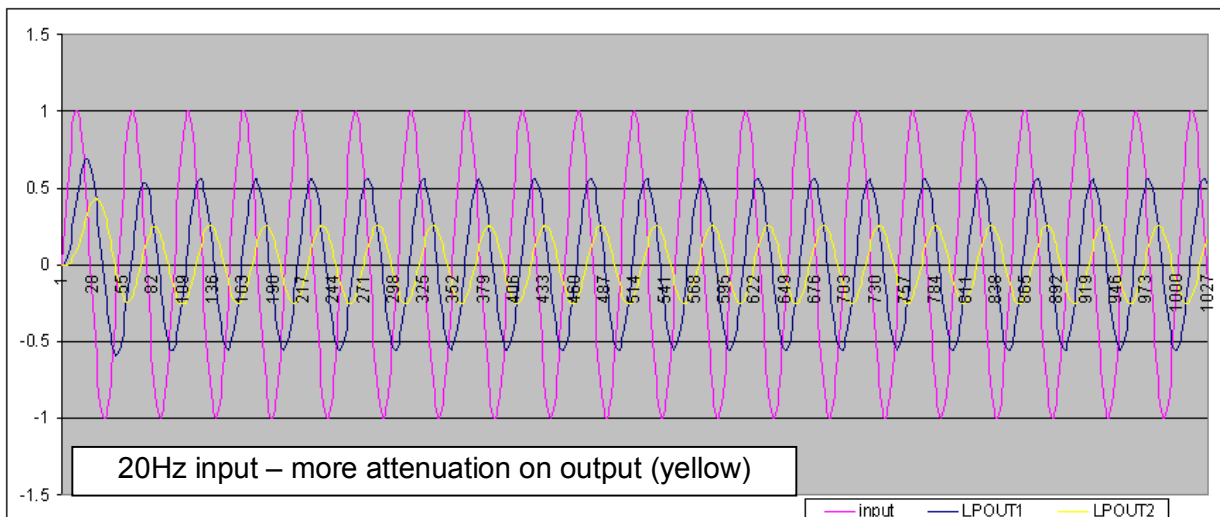
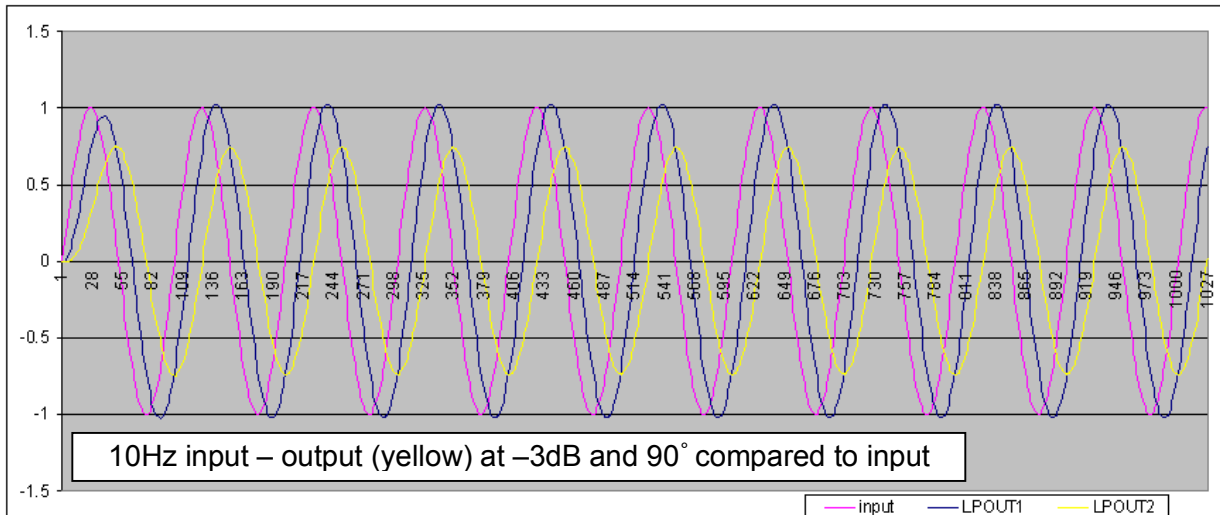
Here’s an example using 1kHz sampling: -

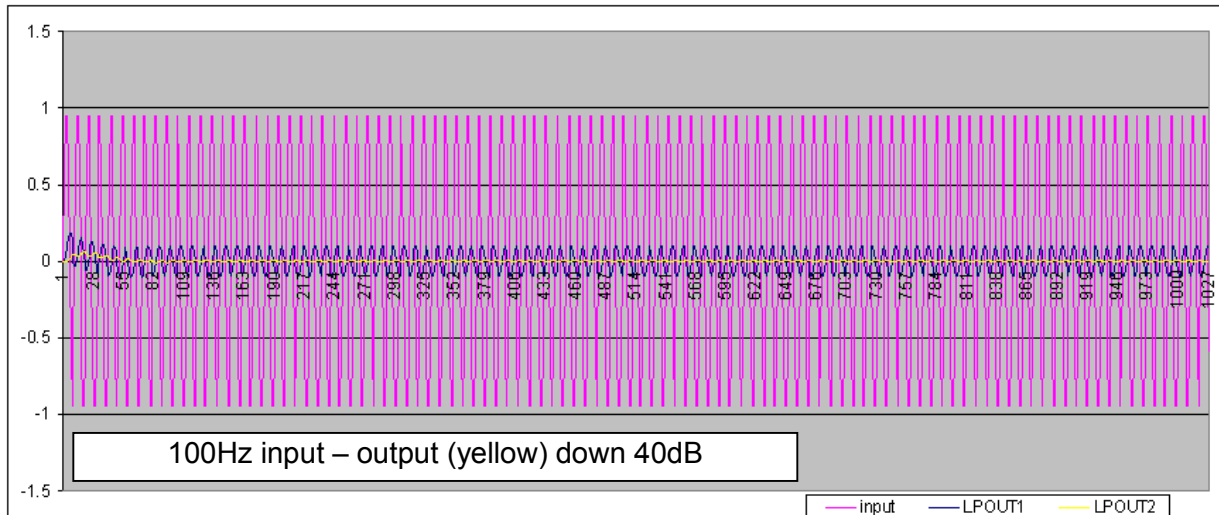


Zeta is set to be 0.1 and as would be expected LPOUT2 (yellow trace) attains Q times the input (pink trace) i.e. 20 times higher. Cut-off frequency and input frequency was arbitrarily chosen to be 23.9Hz. The blue trace is LPOUT1.

Next are a series of traces for a 10Hz filter (zeta = 0.71) with input frequency rising: -

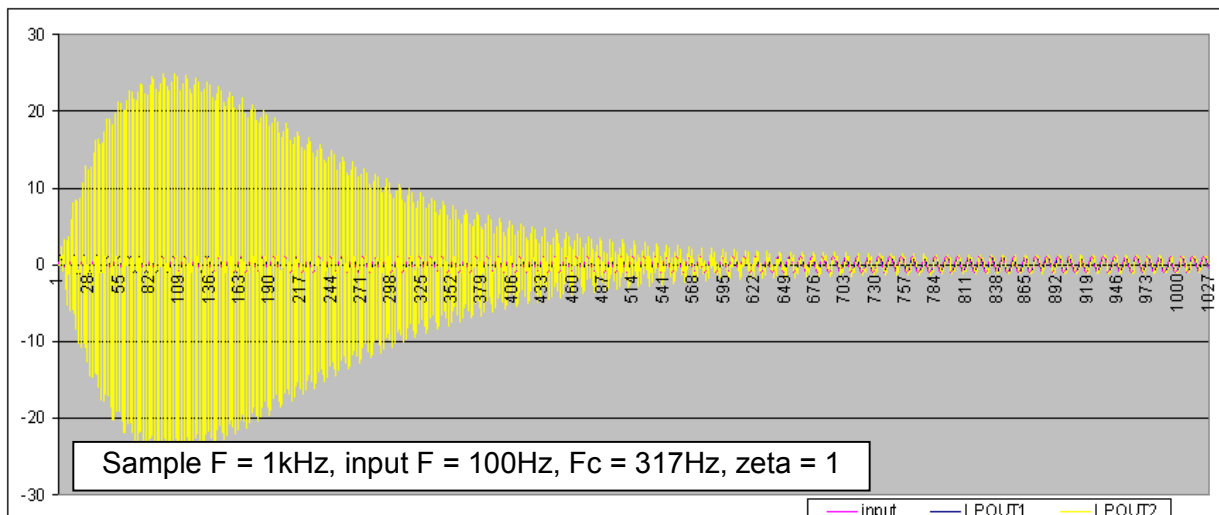






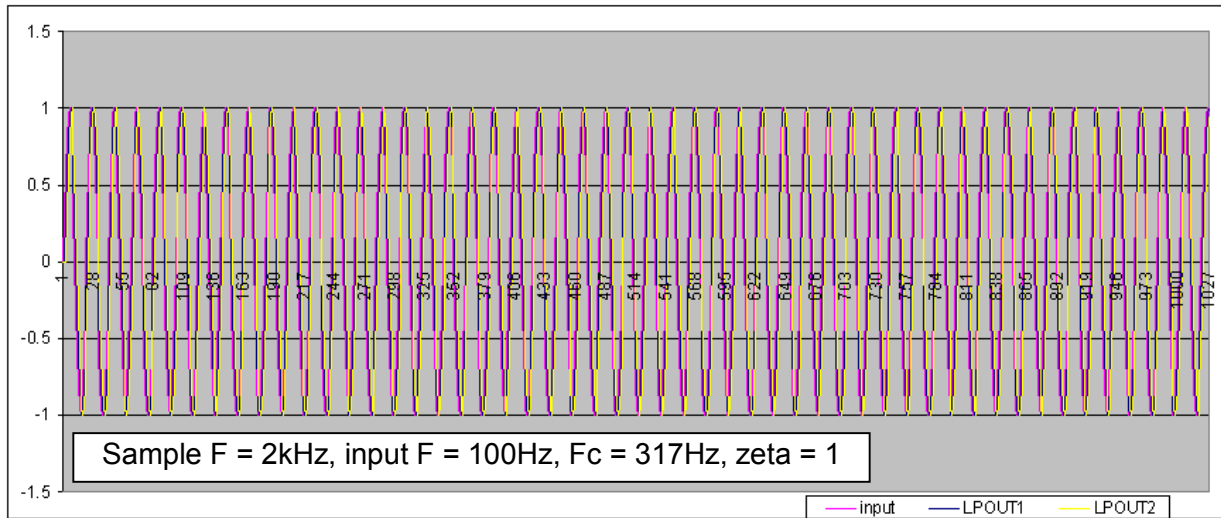
The previous 7 graphs showed a 2nd order 10Hz LP filter with $\zeta = 0.71$. At low frequencies, there is little output attenuation (yellow trace) but as the signal approaches 10Hz there is an attenuation of about 3dB and this becomes 40dB at 100Hz i.e. 40dB per decade as expected. At 50Hz the output is down 28dB i.e. 12dB higher than 100Hz.

Below is an example of the instability problem mentioned earlier. Sample frequency is 1kHz, input signal is 100Hz, cut-off frequency is 317Hz and $\zeta = 1$ ($K=0$).

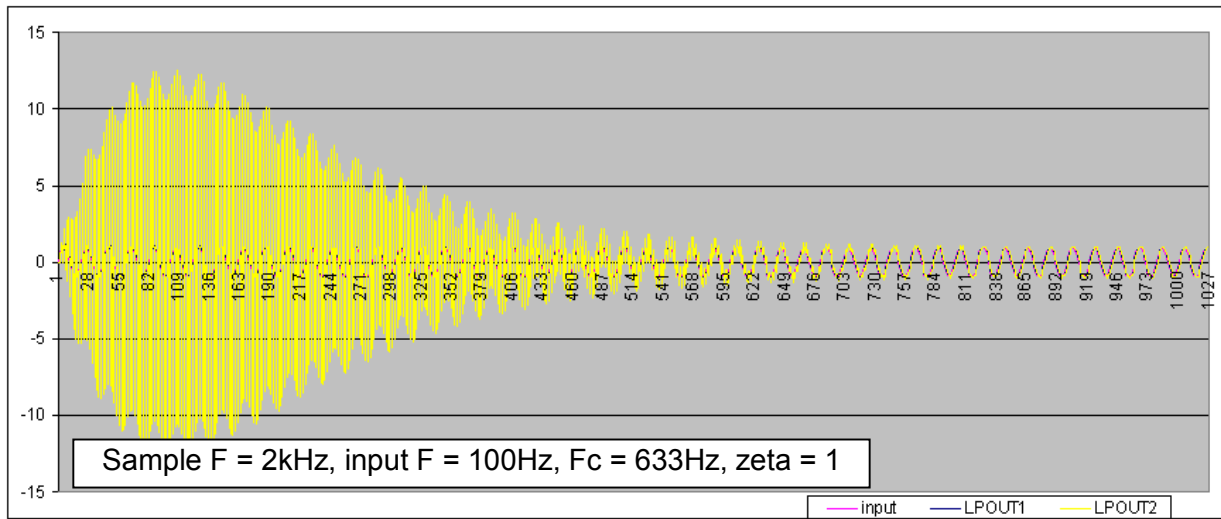


The initial 800 samples show distinct signs of instability but the filter eventually settle down. If cut-off frequency were raised fractionally higher continuous instability would ensue.

Changing the sample rate to 2kHz results in this: -



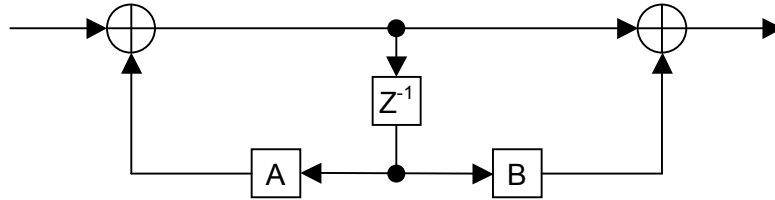
The filter is fully stable proving the point that with a limited sample rate, virtual sampling allows cut-off frequencies to double. Here's the result at 633Hz cut-off frequency i.e. borderline instability almost identical to the result when the sample rate was 1kHz and cut-off at 317Hz.



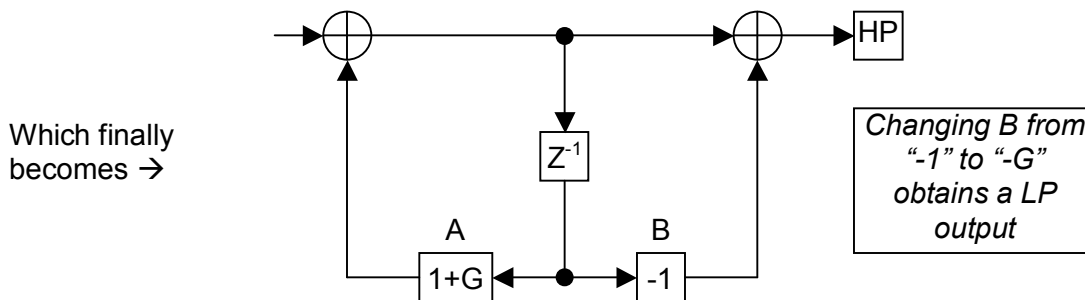
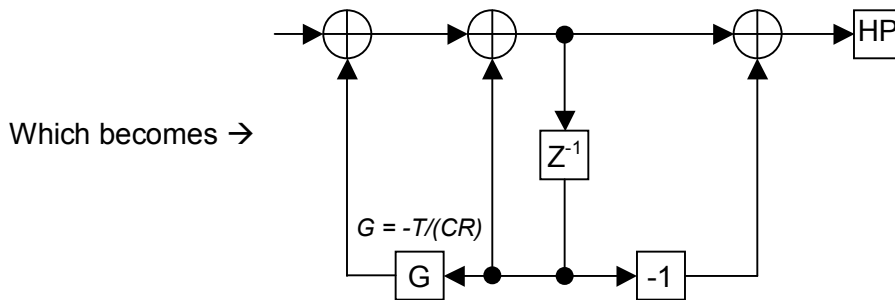
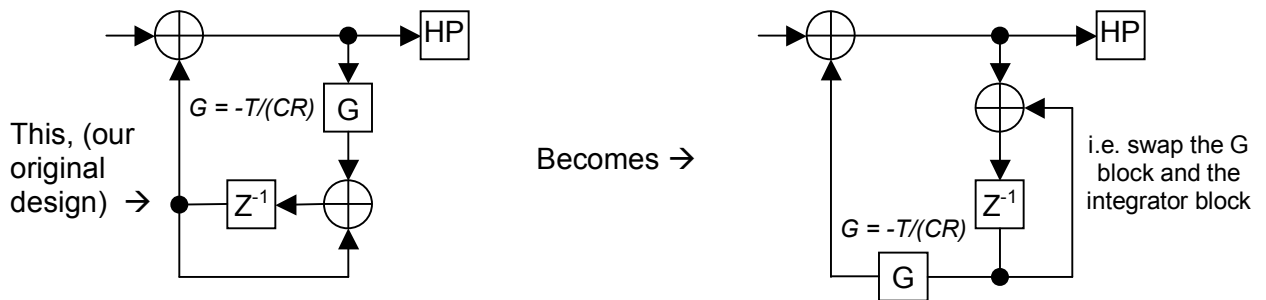
Now go and do your own spreadsheet and play-around!!

What do the text books say...

Many textbooks on the subject show digital filters looking like this (Z^{-1} is a sample delay):



How do we rationalise the above design with the integrator-based design? See below: -



It is evident that the limitations of the integrator method equally apply to the textbook filter. In fact the textbook filter produces digital signals that are greater than 1 so maybe the textbook version has to suffer the indignity of pre-scalers!!