Gunthard Kraus, DG8GB

Introduction to Digital Signal Processing (DSP)

1.0

Principles of Digital Signal Processing

Fig 1 shows an overview of the complete DSP process:

The analog signal is first passed through a low pass filter to limit the frequency range. This feeds a sampling circuit that constantly measures the value of the signal with extremely short sampling pulses. This result of these samples is held in a buffer to feed an analogue to digital converter. This generates samples with a constant frequency (sampling rate) that is processed as a serial 8-bit or 16-bit data stream by a computer, Digital Signal Processor or a micro controller.

The processing in the computer produces a resulting signal as if it had been sent as an analog signal via a filter circuit. The program defines whether this is a low pass filter, high pass filter, band pass filter or a band stop filter.

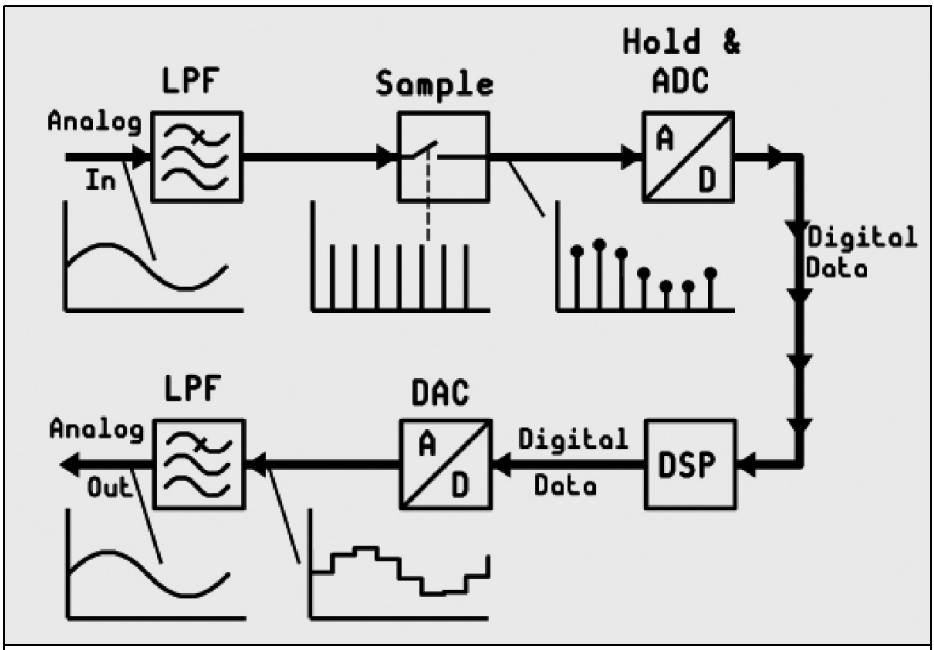

Fig 1: Block diagram of a DSP system: the information goes in as an analogue signal and comes out after digital processing as an analogue signal again.

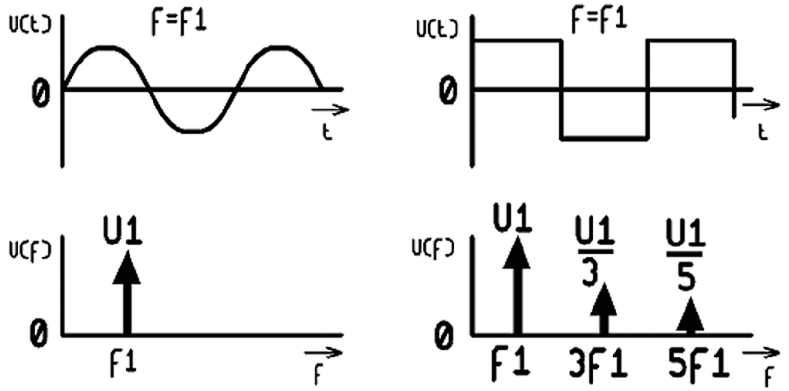

Fig 2: Representation of a signal in the time Domain (oscillogram) or in the frequency Domain (spectrum) are two sides of the same coin.

The digital signal is converted back to an analogue signal using a digital to analogue converter then passed through a low pass filter to suppress the unwanted signals (sampling rate and its harmonics) to produce the required signal.

1.1 Advantages of Digital Signal Processing

The same performance as using arithmetic calculation could only be realised traditionally by great technical expense or may not be achievable at all. In addition the filter values (cut off frequency, type of filter, stop band attenuation etc.) can be switched very easily by changed parameters in the computer program. Adaptive filters can be programmed also coding or decoding etc. can be performed.

2.0

Signal frequencies and sampling rate

The following basic rule must be followed rigidly if everything is to work properly. It is known as the Nyquist criteria:

- For a purely sinusoidal signal of a given frequency the sampling rate must be at least twice the frequency of the incoming signal. It is better if the sampling rate is more than twice the signal frequency. The analogue signal can only be reconstructed again without additional errors after the digital to analogue conversion and the filtering if this is true.

Caution:

As soon as the waveform deviates from a sine wave, such a signal contains a whole range of frequencies other than the basic frequency. These can be harmonics as produced by distortion in an audio signal. They are always an integer multiple of the fundamental.

Fig. 2 shows a comparison between a sine and square wave signal:

- The top left shows a pure sine wave signal as it would be displayed on an oscilloscope and below a single line as it would be seen on a spectrum analyser.
- The top right shows a square wave signal as it would be displayed on an oscilloscope and below the odd harmonics that would be seen on a spectrum analyser.

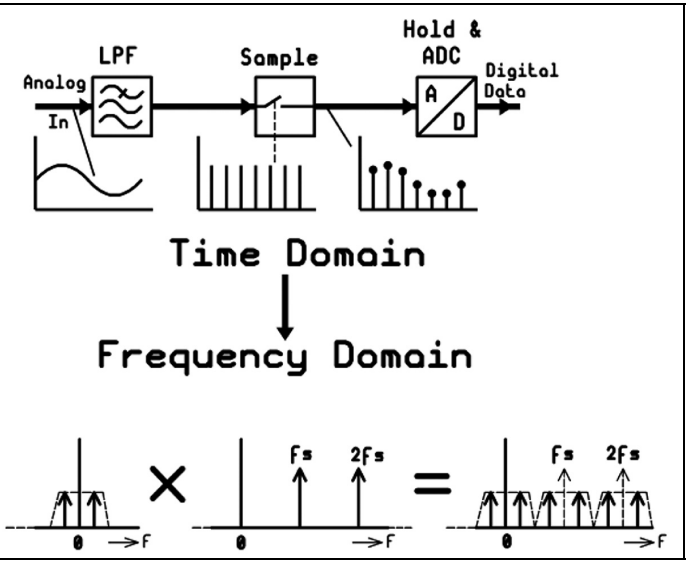

Fig 3: The sampling process shown in the Time Domain or in the Frequency Domain (see text).

Note: the negative frequency portions are not shown in the spectrum because they do not display separately on a spectrum analyser.

As expected a pure sine wave signal only contains one frequency but a symmetrical square wave signal contains the fundamental and odd number harmonics.

Therefore the following condition can defined more precisely:

- The sampling rate must be at least twice as high as the highest single frequency or harmonic in the input signal. Those familiar with harmonics will ask:
- The harmonics of a non-sinusoidal signal theoretically go to infinity with ever decreasing amplitudes. Which scanning rate should be select?

There is only one solution; that is to cut off at some frequency and accept that this will affect the final analogue signal. A low pass filter, or anti aliasing low pass filter with the chosen cut off frequency is used in front of the A/D converter. It should have a very sharp cut off. Select a sampling rate that is at least twice the cut off frequency of this low pass filter.

Example: The cut off frequency for an audio CD Player is 20kHz. The signal is sampled at 44.1kHz.

There are more decisions to be made:

- If the cut off frequency of the low pass filter is very low a relatively low sampling rate can be used. That means less calculation power is required but the danger is that now harmonics of the input signal are missing that are important for the correct waveform shape of the analogue signal.
- If the cut off frequency and the sampling rate are higher than necessary, the waveform shape of the analogue signal is correct, but the following drawbacks result:
- The higher range of the input signal increases the self-noise level. That can be quite a problem if the information signals are in the order of a microvolt.
- The higher scanning rate requires such a high calculation performance that even the most modern PCs and processors are pushed to their limits. Only genuine Digital Signal Processor (DSP) will help, these have a

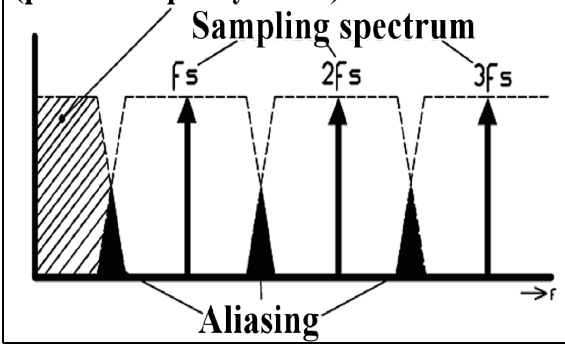

Fig 4: Aliasing happens if the sampling rate is not high enough.

different hardware structure to perform more than a billion 32-bit floating point calculations per second.

For information: because the power of computers and DSPs seems to increase by the hour, the calculation performance is no longer the problem. Therefore the second option, to sample at a higher frequency than necessary, is often chosen these days. The data stream is filtered with a "digital decimation filter" after the A/D converter that artificially reduces the cut off frequency giving consistency with a lower sampling rate, but this must still be at least twice as high as the data bandwidth according to Shannon.

The question to be clarified now is: What happens if the sampling rate is not more than twice the maximum input frequency?

Fig 3 shows a comparison of all the signals involved both in the time domain and the frequency domain:

- In the time Domain the sampling can be regarded as multiplication of the analogue input signal by sampling pulses. Therefore the pulses at the output of the sampler follow the amplitude of the analogue input signal.
- In the frequency domain a multiplication of two sine signals always results in two new signals: the sum and difference frequencies of both.

Consider this:

The scanning pulse produces a spectrum containing the odd harmonics with the same amplitude. Each harmonic is multiplied by the analogue input signal. Therefore the final result is a spectrum containing sum and difference frequencies as multiplication results of every harmonic and the input signal. Developing this further; with an input signal containing more than one frequency you get lower and upper sidebands as multiplication results.

Fig 4 shows what happens with a sampling rate that is too low:

- The spectra begin to overlap that leads to extremely unpleasant distortions. This is called ALIASING and the distortion is called "Aliasing distortions". Unfortunately this not only sounds very unpleasant but if this happens there is no way to eliminate it. The low pass filter before the A/D converter is to prevent this effect. Therefore it has the appropriate name of "Anti Aliasing" Filter.

Remember this:

EVERYTHING in the input signal must be suppressed above half the sampling rate at all costs otherwise there is terrible trouble.

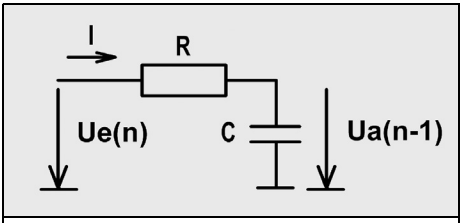

Fig 5: This well-known simple circuit can show the principle the digital filter.

3.0

Practical project: A simple digital filter

3.1. Introduction

The serial output of the sampler is fed into a digital filter. The different sampled values are now multiplied by special factors and the multiplication results added together. So the input analogue signal is affected in the same a way as if it had passed through a discrete filter circuit.

In order to understand this, Fig 5 shows a circuit that everyone knows; the simple RC low pass filter. At a certain time the output voltage $U_{a(n-1)}$ is developed across the capacitor. If the sampling circuit produces the next sampled value, the input voltage changes suddenly. The new voltage $U_{e(n)}$ is at the input while the capacitor voltage still holds the old value $U_{a(n-1)}$.What will the new output voltage $U_{a(n)}$ be until the next scanning value arrives, for the time interval $\Delta t = t_{sample}$

The value at the capacitor will be:

$$U_{a(n)} = U_{a(n-1)} + \Delta U_c = U_{a(n-1)} + \frac{\Delta Q}{C}$$

$$= U_{a(n-1)} + \frac{I \cdot \Delta t}{C} = U_{a(n-1)} + I \cdot \left(\frac{t_{sample}}{C} \right)$$

$$U_{a(n)} = U_{a(n-1)} + \left(\frac{U_{e(n)} - U_{a(n-1)}}{R} \right) \cdot \left(\frac{t_{sample}}{C} \right)$$

$$= U_{a(n-1)} + \left[U_{e(n)} - U_{a(n-1)} \right] \left(\frac{t_{sample}}{RC} \right)$$

Expressing t_{Sample} in terms of the sampling frequency gives $t_{Sample} = 1/f_{Sample}$

There is a formula for the cut off frequency of an RC filter used in communications theory:

$$f_{cut-off} = \frac{1}{2\pi RC} \Rightarrow \frac{1}{RC} = 2\pi f_{cut-off}$$

Substituting this in the formula for $U_{a(n)}$ gives:

$$U_{a(n)} = U_{a(n-1)} + \left[U_{e(n)} - U_{a(n-1)} \right] \left(\frac{2\pi f_{cut-off}}{f_{sample}} \right)$$

Using the term k for the term in the right hand bracket gives a simplification. The result after multiplying out gives:

$$U_{a(n)} = U_{a(n-1)} + k \cdot \left(U_{e(n)} - U_{a(n-1)} \right)$$

$$= U_{a(n-1)} - k \cdot U_{e(n)} - k \cdot U_{a(n-1)}$$

$$U_{a(n)} = k \cdot U_{e(n)} + (1-k) \cdot U_{a(n-1)}$$

The new value is calculated by adding together the new input value $U_{e(n)}$ multiplied by k and the preceding output value $U_{a(n-1)}$ multiplied by 1-k. This is a usual procedure during digital signal processing; it is abbreviated to MAC (Multiply And Accumulate).

3.2. Training example: Investigation of the RC circuit by hand

For this example use a low pass filter with a cut off frequency of 100Hz and a sampling rate of 2kHz. The input frequency is comfortably smaller than half the sampling frequency.

Step 1:

First the factors k and 1-k are calculated:

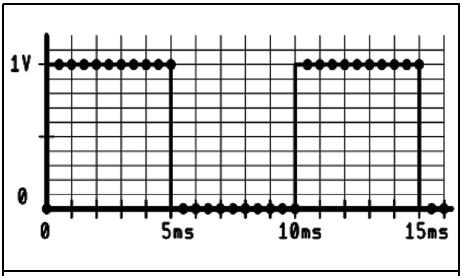

Fig 6: The input signal with the sampling times.

$$k = \frac{2\pi \cdot 100Hz}{2000Hz} = 0.314$$

$$(1-k) = 1 - 0.314 = 0.686$$

Step 2:

The initial value of the output voltage with t = zero is zero. Starting from the time t = zero as the input square wave signal with a frequency f = 100Hz is applied the amplitude goes from 0V to 1V. The sampled input signal from t = zero to t = 15mS is shown in Fig 6.

Step 3:

Table 1 shows the output voltage values across the capacitor. The fastest way to do this is to use Excel, but naturally it can be done with a pocket calculator. The results are also shown in Fig 7. The upper half shows the input signal with the sampling times. The lower half shows the output signal that would be produced after Digital Signal Processing and D/A conversion given by Table 1. It is similar to the known output of a low pass filter, it just need filtering to remove the remains of the sampling.

Table 1: Calculation of the output voltage for different samples.

Sample Number n	New Input $U_{e(n)}$	k * Ue(n) $0.314 * U_{e(n)}$	Old Output $U_{a(n-1)}$	(1-k) * $U_{a(n-1)}$ $= 0.686 * U_{a(n-1)}$	New output $U_{a(n)} = 0.314 * U_{e(n)} + 0.686 * U_{a(n-1)}$
0	0	0	0	0	0
1	1	0.314	0	0	0.314
2	1	0.314	0.314	0.215	0.529
3	1	0.314	0.529	0.363	0.677
4	1	0.314	0.677	0.464	0.778
5	1	0.314	0.778	0.533	0.848
6	1	0.314	0.848	0.581	0.895
7	1	0.314	0.895	0.614	0.928
8	1	0.314	0.928	0.637	0.95
9	1	0.314	0.95	0.652	0.966
10	1	0.314	0.966	0.662	0.977
11	0	0	0.977	0.670	0.670
12	0	0	0.67	0.459	0.459
13	0	0	0.459	0.315	0.315
14	0	0	0.315	0.216	0.216
15	0	0	0.216	0.148	0.148
16	0	0	0.148	0.1	0.1
17	0	0	0.1	0.069	0.069
18	0	0	0.069	0.0479	0.0479
19	0	0	0.0479	0.0328	0.0328
20	0	0	0.0328	0.0225	0.0225
21	1	0.314	0.0225	0.015	0.329
22	1	0.314	0.329	0.226	0.54

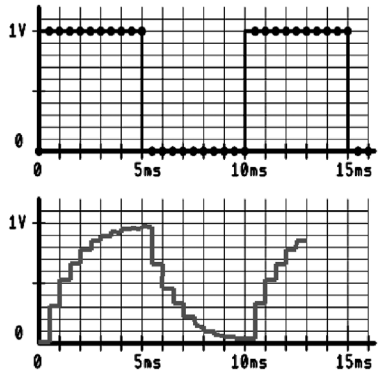

Fig 7: What the digital filter makes from the analogue signal.

4.0

FIR – Filter (Finite Impulse Response filter)

In practice a substantially more complex filter than a simple low pass filter is required. Knowledge of higher mathematics (complex calculation and integral calculus) is required to design these filters because they use Z transforms. Their design and function, in a simplified form, becomes clear with the following fact:

- In a block diagram each element has a value z^{-1} that represents the storage of the previous sampled value. Fig 8 shows how a draft filter diagram could look and with this knowledge we can understand the structure the FIR filter.

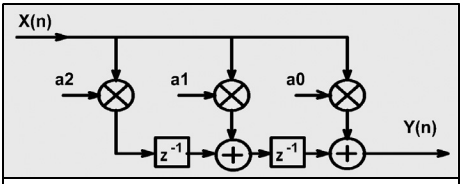

Fig 9: Block diagram of the transfer function of the FIR filter (see text).

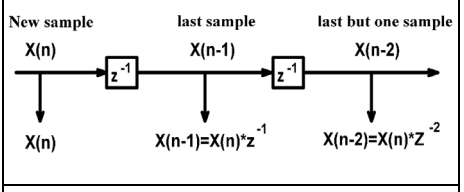

Fig 8: The Z transform working on the sampled value.

Explanation:

FIR means Finite Impulses Response. These filters respond to a needle pulse input signal with an output signal that decays to zero. These filters never ring and are stable. They can be recognised because they have a one-way structure from input to output, there is NEVER feedback from the output to the input. An alternative designation is a non-recursive filter.

A practical example is shown in Fig 9.

The output signal is given by:

$$Y_{(n)} = a_0 \cdot X_{(n)} + a_1 \cdot X_{(n)} + a_2 \cdot X_{(n)} \cdot z^{-2}$$

that means:

$$Y_{(n)} = a_0 \cdot X_{(n)} + a_1 \cdot X_{(n-1)} + a_2 \cdot X_{(n-2)}$$

Extracting the factor $X_{(n)}$ gives the transfer function:

$$Y_{(n)} = X_{(n)} \cdot \left[a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}\right]$$

$$\Rightarrow \frac{Y_{(n)}}{X_{(n)}} = H(z) = a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}$$

(A small reference: a filter is only then absolutely stable if in the denominator of the transfer function no z^{-1} can be found)

Advantages of the FIR filter:

- Absolute stability, thus no ringing
- If the filter is symmetrically designed ($a_0 = a_2$), it has a linear phase re-

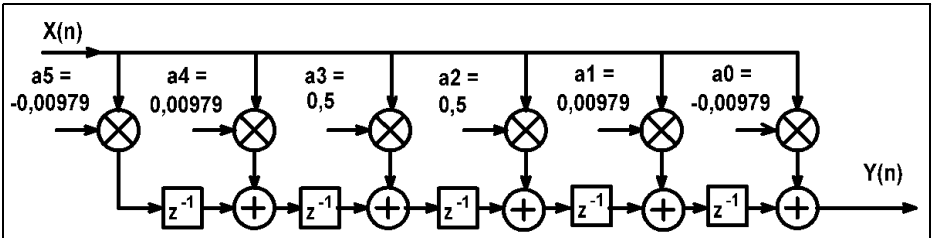

Fig 10: This example shows that the programming work for the DSP machine is not an easy walk.

sponse and thus a constant Group Delay.

- Design not critical even with rounding or simplifications in the program

Disadvantages of the FIR filter:

- Filter slope not as steep as IIR filters of the same degree.

Some references to the design of such FIR filters.

The design needs good knowledge of mathematics but now on the Internet, there are many programs (including online calculators) that can do the job perfectly. Nevertheless some fundamental things should be known:

- The sum of the coefficients ($a_0/a_1/a_2$…) gives the DC voltage gain of the filter.
- The standardised cut off frequency F_g is used for design rather than the actual cut off frequency. This makes it easier to understand the relationship between the actual cut off frequency and the sampling rate that must always be below 0.5 (the Nyquist criteria).
- The filter degree should be always odd because then there is always a zero in the frequency response at half the sampling rate (the Shannon condition).

Example:

- A filter for $F_g = 0.25$ and filter degree N = 5 with the filter coefficients
- $a_0 = a_5 = -0.00979$
- $a_1 = a_4 = +0.00979$
- $a_2 = a_3 = +0.5$

Calculating the actual cut off frequency if the sampling frequency is 8kHz.

Solution:

$$f_{cut-off} = F_g \cdot f_{sample} = 0.25 \cdot 8kHz = 2kHz$$

The associated circuit is shown in Fig 10.

5.0
IIR filter (Recursive filter)

These always have feedback from the output to the input and therefore need substantially more care to design - they are inclined to oscillate with the smallest sloppiness (known from analogue feedback amplifiers). Rounding or truncating large numbers too early in the design can cause this.

The same filter degree gives a substantially steeper filter slope therefore improving the filter effects compared to the FIR filters discussed.

IIR means Infinitely long Impulse Response. The response to a needle pulse at the input decays away but never completely to zero but hopefully becomes ever smaller. In addition these strange

Continued on page 35

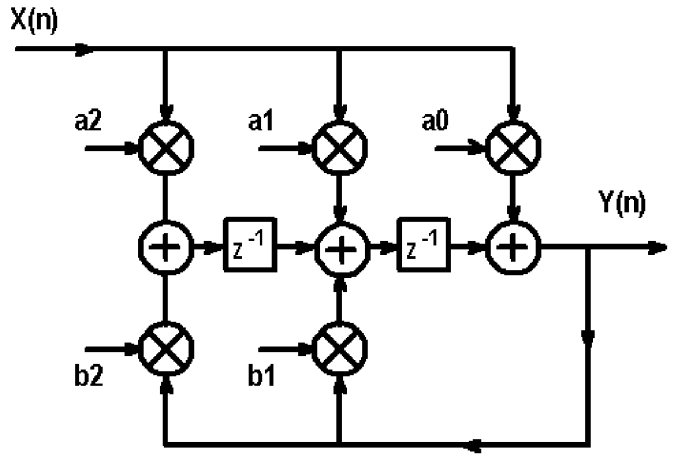

Fig 11: IIR filters are more complicated. They make greater demands on the processor, the program and the programmer.

filters already supply a result at the output before the impulse appears at the input, thus they are a special kind of filter that must be treated carefully.

An example of such a filter arrangement can be seen in Fig 11:

$$Y_{(n)} = a_0 \cdot X_{(n)} + a_1 \cdot X_{(n)} \cdot z^{-1} + a_2 \cdot X_{(n)} \cdot z^{-2}$$
$$+ b_1 \cdot Y_{(n)} \cdot z^{-1} + b_2 \cdot Y_{(n)} \cdot z^{-2}$$
$$\Rightarrow Y_{(n)} \cdot \left(1 - b_1 \cdot z^{-1} - b_2 \cdot z^{-2}\right)$$
$$= X_{(n)} \cdot \left(a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}\right)$$

From this the transfer function follows:

$$\frac{Y_{(n)}}{X_{(n)}} = \frac{\left(a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}\right)}{\left(1 - b_1 \cdot z^{-1} - b_2 \cdot z^{-2}\right)}$$

The denominator of the transfer function can possibly become zero - particularly if the coefficients b_1 and b_2 have positive values. Then the filter oscillates.

This calculation also uses the stored preceding values. With such a complex structure a whole assortment of initial values and buffers will be required.

Such a filter was introduced in simple form with the RC low pass filter in chapter 3.2. The following applied:

$$U_{a(n)} = k \cdot U_{e(n)} + (1 - k) \cdot U_{a(n-1)}$$

The actual cut off frequency was 100Hz with a sampling rate of 2kHz.

The coefficients were:

k = 0.314 and 1-k = 0.686

Example:

a) The transfer function of this filter with these two coefficients is derived

b) The circuit of this filter drawn with registers of the correct values.

Solution to a):

$$U_{a(n)} = k \cdot U_{e(n)} + (1 - k) \cdot U_{a(n-1)}$$
$$U_{a(n)} = k \cdot U_{e(n)} + (1 - k) \cdot U_{a(n-1)} \cdot \left(z^{-1}\right)$$
$$U_{a(n)} - (1 - k) \cdot U_{a(n)} \cdot \left(z^{-1}\right) = k \cdot U_{e(n)}$$
$$U_{a(n)} \cdot \left[1 - (1 - k) \cdot z^{-1}\right] = k \cdot U_{e(n)}$$
$$\frac{U_{a(n)}}{U_{e(n)}} = H(z) = \frac{k}{1 - (1 - k) \cdot z^{-1}}$$

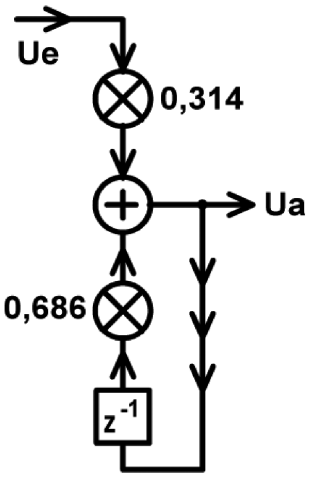

Fig 12:Even a simple RC low-pass filter more complex in digital form.

Thus:

$$H(z) = \frac{0.314}{1 - 0.686 z^{-1}}$$

The block diagram of this RC low pass filter in digital form is shown in Fig 12.

6.0

Organisation of the sampled values in the memory

There are always a constant number of samples being processed in the DSP. The number is nearly always multiples of two e.g. 512, 1024 or 2048. When the next sampling value arrives from the A/D converter there is congestion:

Not only is the current value stored but X_n values are stored so the oldest value is deleted and the newest value is set to the front of the list. Additionally all remaining values in the list must be renumbered. This is fastest for a small number of stored values. All the values are relocated to put them in the correct position. The process of relocating 4 sampling values is shown in Fig 13. This would take a long time for a large number of samples therefore the following method is adopted:

- The number of sampled values is always multiples of two e.g. 2, 4, 8, 16, 32, 64, 128….1024, 2048, 4096...

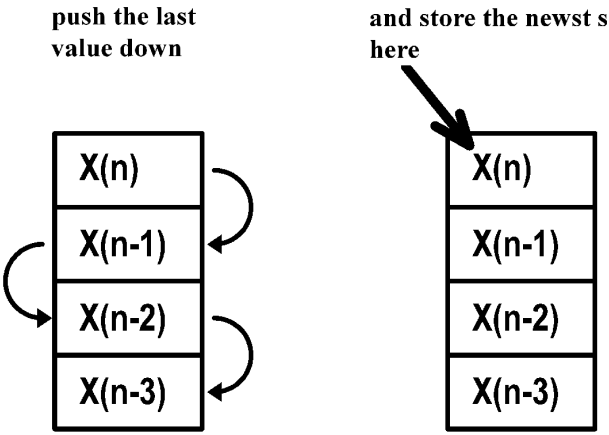

Fig 13: This is one method of updating the stored values when a new sampled value arrives. This method is only recommended for a few stored values.

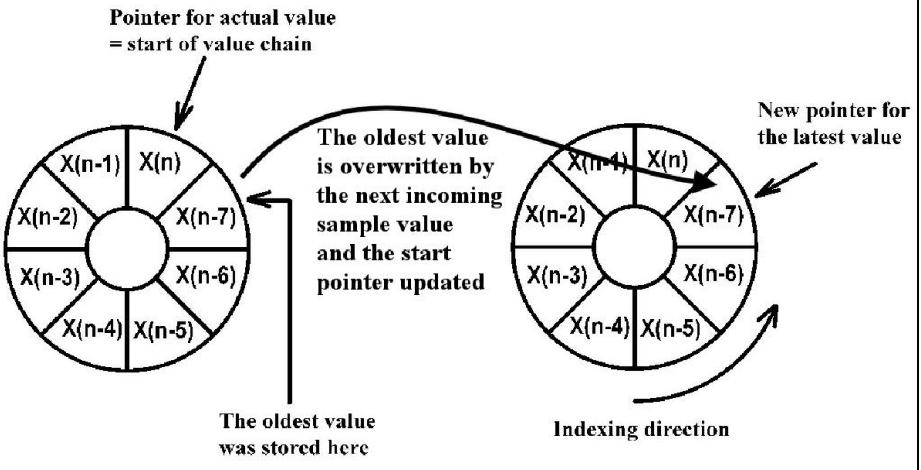

Fig 14: Using a pointer is much better for a large number of sampled values.

The samples are organised in memory like a ring and the start of the chain is stored as a pointer in a variable. This pointer always points to the newest value. The next sample is stored in the last place of the ring overwriting the oldest value. The pointer is changed to the new start address of the ring and calculation can begin. This principle is represented in Fig 14.

Let's hope that now everything is clear!