Signal acquisition

Introduction

Variables representing physical world quantities are for the most part analog, such as temperature, pressure, positions, velocities, chemical concentrations, and many others. They are characterized by varying continuously with time, and we may be interested in the values at any instants in time, or only in the peak, average or r.m.s. values, or in their rates of change with time. Or we may be interested in their frequency content, that is, we would like to perform some type of frequency analysis of the signals. Or maybe we want to store their values at specific points in time, called data logging. Another possibility is that the analog signals represent variables in a manufacturing process and their values are to be used as inputs in the process control mechanism.

Whatever the application, the most powerful and versatile method of performing it is by first bringing the analog signals into the digital domain, into a digital machine, and then do all the signal processing digitally. We can easily do this with a PC equipped with a data acquisition card and the appropriate software. This software incorporates an interface with the card and, through this, the signal samples are transferred to the PC memory. Other parts of the program may allow the user to set such things as the gain of the several analog channels and to visualize the signals on the PC monitor screen, or do some processing of the values and show the results.

The data acquisition card contains a circuit known as analog-digital converter. This is the circuit where the analog signal amplitude is measured at equally spaced discrete time points, and the value of the amplitude is coded into digital words. We look next at this process because it contains several pitfalls.

Analog-digital conversion

The analog-digital converter (ADC) performs two basic functions. It samples the analog signal, and then it converts the amplitude of the sample to a digital value, a process known as quantization. We can think of these two processes as time discretization and amplitude discretization.

Sampling

Figure 1(a) represents an analog signal x(t), and 1(c) a sequence of samples $x_p(t)$, with a period T_s, which is the result of the sampling or time discretization of the analog signal. Mathematically this is equivalent to multiplying the analog signal by the train of impulses p(t), shown in figure 1(b). The multiplication in the time domain results in the convolution of the spectra of the signal X(ω) and the impulse train P(ω) in the frequency domain (fig. 2) [Oppe97]. In the signal spectrum, Fig. 2(a), ω_m is the frequency above which the signal has no energy content. In the spectrum of the impulse train, fig. 2(b), $\omega_s=2\pi/Ts$, is the sampling frequency.

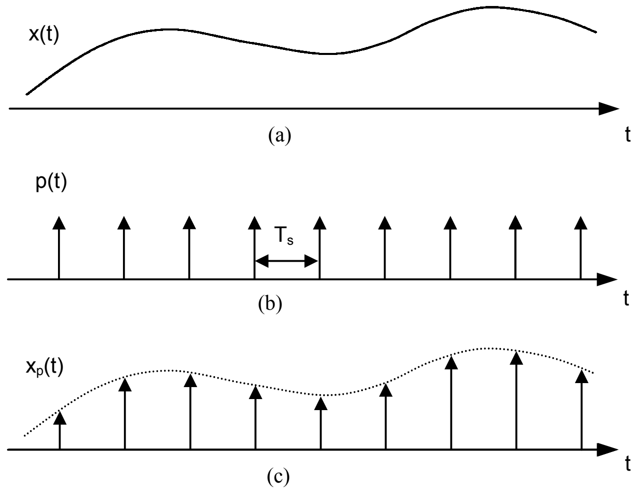


Figure 1: Sampling an analog signal with an impulse train

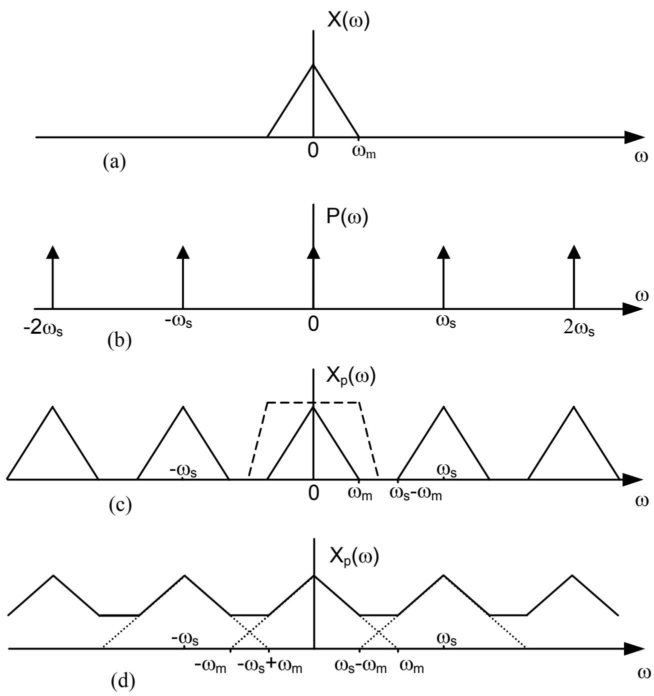


Figure 2: Spectra of the signals in figure 1

Aliasing

The convolved spectrum, that is the spectrum of the sampled signal, fig. 2(c), shows that if $\omega_m<(\omega_s/2)$ the original signal can be completely recovered from the sampled signal by a suitable low-pass filter, with a frequency response for example like the one shown as a dashed line. If, on the other hand, $\omega_m>(\omega_s/2)$, the images of the analog signal spectrum, centered at multiples of ω_s, overlap as shown in fig. 2(d), and no filtering will succeed in recovering the original spectrum intact. This problem is called aliasing, because the high frequency components of an adjacent signal spectrum image overlap the original spectrum masquerading as (with the alias of) lower frequency components. For example, the component at frequency ω_m is folded back as a component at frequency ω_s-ω_m, and if this frequency falls within the spectrum of the signal as in fig. 2(d), when we try to recover the original signal we no longer have the original spectrum to do it. This is then the first pitfall to avoid. In practice, this means that we have to filter the analog signal to limit its bandwidth to less than half the sampling frequency, prior to sampling. The usual and most effective place for this filter is the input to the data acquisition board. This may be needed even if we are dealing with a very low frequency signal (compared with the sampling frequency). Real world signals often appear corrupted by noise, usually picked up by connecting wires from ambient electric and magnetic fields, and the noise frequency content may be much higher than the signal's. This high frequency noise energy would then be folded back into the signal band increasing the overall noise.

Quantization

Every time the analog signal is sampled, its amplitude is converted into a N bit binary value. Since there are only 2^N possible binary values, this means that, at best, the analog signal amplitude can be measured with a resolution of $1/2^N$. For instance, in a 10 bit analog-digital converter, the resolution is 1/1024, or slightly better than 0,1%. The actual amplitude of this resolution interval depends on the maximum analog voltage that the converter is able to convert correctly, called the full-scale voltage (V_{fs}). For example, in a converter that has a 0V to 10V analog range (V_{fs}=10V), the resolution interval is $V_{fs}/2^N$ or about 9.77mV. This quantity is sometimes called the quantum *Q*, also the least significant bit or *LSB*, because it is the weight of the least significant bit of the binary value output by the converter. The quantum or LSB is then given by the following expression:

$$Q = LSB = \frac{V_{FS}}{2^N}$$

If the converter is a bipolar one, that is, the input voltage can assume positive or negative values, the expression above will still be valid if we consider the range given by $\left[-\frac{V_{fs}}{2};+\frac{V_{fs}}{2}\right]$.

One can think of the converter as a device that converts each analog sample into a binary code, which, if decoded, that is, converted back into an analog value, would produce the best approximation to the original analog value.

The input/output characteristic of an ideal analog-digital converter is shown in figure 3. For clarity, the number of bits is three, and the analog input range is 0V to 8V. The analog values corresponding to each binary code are shown between parentheses next to the codes.

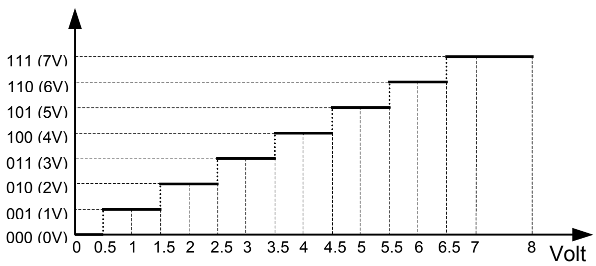
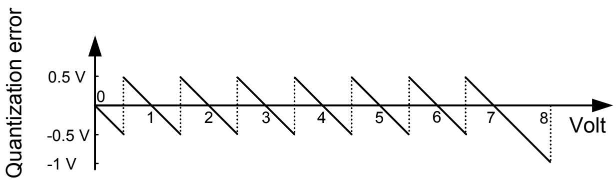


Figure 3: Ideal characteristic for a 3 bit unipolar ADC

For this converter, 1 *LSB* = 1V. It can be seen that input analog values between 0 and 0,5V, that is ½ LSB, correspond to the binary output 000, which corresponds to 0V. The values between 0,5V and 1,5V correspond to 001, which translates into 1V. The same happens in all the other 1V, or 1 LSB, intervals, except the last one. The last interval runs from 6,5 V to 8V, corresponding to the code 111, that is 7V. Note that except for this last interval, the maximum difference between the coded analog value and the true analog value never exceeds ½ LSB. This difference is called the quantization error, and its amplitude as a function of the analog input value is represented in figure 4.

It can be seen that, except for the last interval, corresponding to the code 111, this operation is equivalent to rounding to the nearest integer the input value measured in units of quantum. That is, the output code C is such that its binary value C_b is given by

$$C_b = \text{Round}\,(Vi/Q)$$



(b)

Figure 4: Quantization error for the ADC shown in figure 3

Real-world ADC Errors

Without even discussing the actual electronic circuits that are used to implement the analog-digital conversion process we can be sure of one thing: they are not going to behave ideally. Even if we use the right sampling rate and an adequate number of bits, the results of the conversion, that is, the actual digital values of each sample, are going to suffer from different types of errors. The one that really matters, because it is not easily compensated, is the linearity error.

We can understand what the linearity error is by imagining the following experiment: we apply a perfect sine wave to the input of the ADC; we then reconstitute the wave shape from the samples with a perfect DAC. If the output wave shape is not a perfect sinusoid, then the ADC is not linear. The term linear applied to a single input single output system indicates that the equation relating the output of the system to its input, is a linear function that is it only has first order terms. When we look at the output signal in the frequency domain, we can see that it has several spectral lines, that is, energy at other frequencies has appeared, and it can be shown that this only happens at frequencies that are integer multiples of the input sine wave frequency. They are called harmonics - second harmonic, third harmonic and so on, according to their frequencies, twice, three times, etc. of the original sinusoid which is of course still present in the output. This happens with any signal that we apply to the ADC, but we only see harmonics easily when the input is a pure sine wave.

References

[Oppe97] Alan Oppenheim, Alan Willsky, S. H. Nawab, “Signals and Systems”, 2nd edition, Prentice-Hall Signal Processing Series, 1997