Digital Instrumentation: Pt100

Daniel Vergara - T7173177

December, 2017

Abstract

A digital instrument was designed using the software and programming language LabView in order to process the temperature information obtained from a temperature sensor Pt_{100}. A conditioning circuit, consisting on a quarter deflection bridge and a differential amplifier was build and a connection with the RTD sensor was tested. A National Instruments data acquisition card was used to digitize the data. However, due to the few measurements a numerical signal was implemented, by direct implementation of the circuit equations, to fully test the digital signal conditioning. The signal was then corrupted by a frequency controlled noise, a noise only affecting a particular frequency and a digital low pass filter, created by bilinear transformation of a second order Butterworth filter was implemented in order to remove this noise. The results shown the the major features of the DSP and the importance of the selection of a correct sampling rate.

Keywords: DSP, Digital Signal, Pt_{100}, Cascade Linearization, Digital Filters.

1 Introduction

Digital signal processing (DSP) consists on applying numerical tools to a modify a digitized signal in order increase its manageability and to extract from it information required for some practical use. For the DSP all signals are considered as information to be processed.

The task to complete was the design of a digital filter, implementing DSP, to extract some Temperature information from a RTD temperature sensor Pt_{100}, using NI LabView knowing the correlation between the temperature and electrical resistance change at least in the range 0 to 100°C. Due to the multiple non-linearities presented in the different stages of the measurement a cascade linearizator was implemented to obtain an overall linear system. Furthermore a frequency controlled noise was introduced into the signal to then design a digital filter to remove it without affecting the signal in outside that frequency ranges.

2 Overview

A conditioning circuit for a temperature measurement using a Pt100 sensor was given. The task was to digitize its analog signal output using a NI data acquisition card then analyze it through digital signal processing (DSP). Figure 1 shows a block diagram the whole sytem structure.

The overall DSP starts with by introducing an **Analog Signal**, in this case temperature (**T**), obtained through a resistive temperature sensor Pt_{100} into a conditioning circuit (figure 4).

The Pt_{100} is a platinum resistor with resistance equal to 100Ω at $0°C$ and 138.5Ω at $100°C$ with a linearity error of $0.5\%/°C$. The conditioning circuit consists of electronic components used to transform the resistance change in the sensor to an analog signal, in this case a voltage, that carries the information of the measured signal.

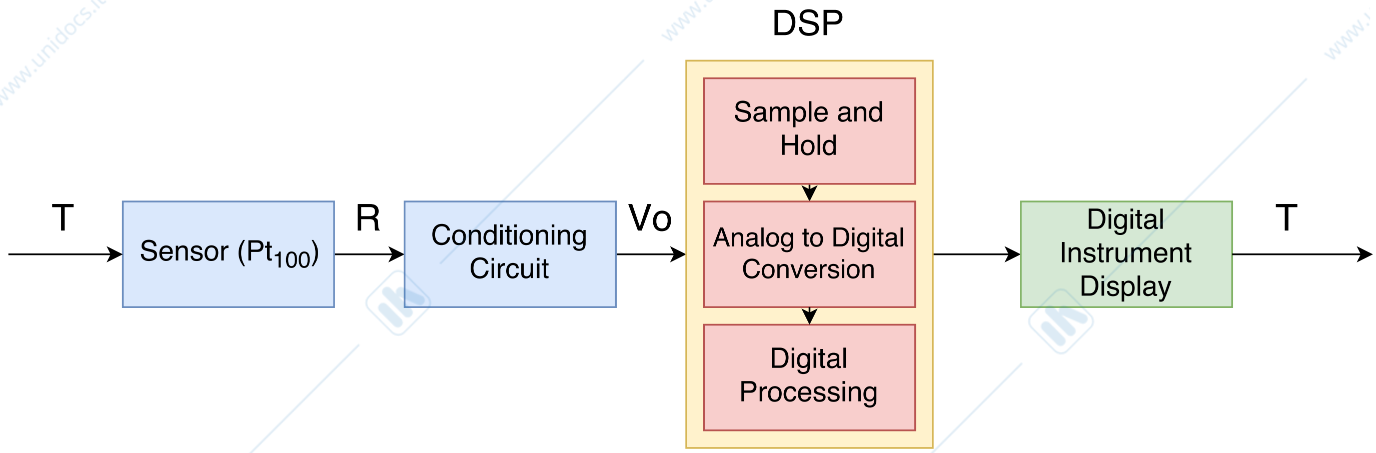

Figure 1: System Structure

Within the conditioning circuit lies an **Anti-Aliasing Filter**. The aliasing problem occurs when a signal is sampled at a frequency under the Nyquist-Shannon frequency threshold, that is, below twice the maximum frequency present in the system. The aliasing problem is shown in figure 2. It consist on a lose of information due to an incorrect sampling process.

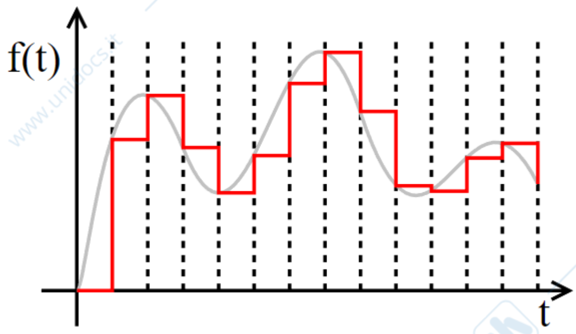

Figure 3: Sample and Hold

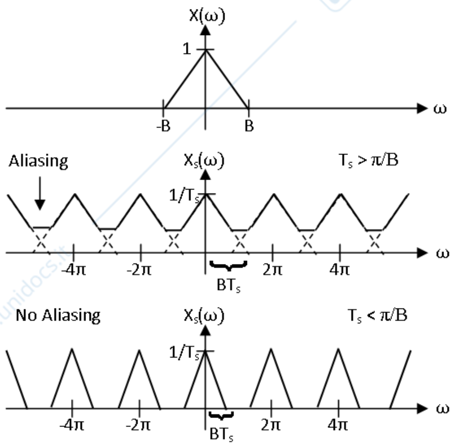

Figure 2: Aliasing problem.

Using the **Anti-Aliasing Filter** the signal frequency spectrum is able to pass through the **Data Acquisition Card (DAQ)** without suffering from the aliasing problem.

Inside the DAQ the **Sample and Hold** occurs. It consists on sampling signal values at a constant sampling frequency f_s and holding its value for a time T_s (inverse of the sampling frequency) and storing the value.

The sampled signal now have constant values in $\mathbf{n} \cdot \mathbf{T_s}$ intervals. It is then digitized using an **Analog to Digital Conversion (ADC)** circuit. The three most used are 'Successive Approximations', 'Integrating Dual-Slope' and 'Flash'.

In this step the signal is now digital and ready to be digitally processed. The DAQ gives as an output digital information to be analyzed. This digital signal can be filtered to remove almost all the noise it has through a digital filter. Furthermore it can be linearized through cascade algorithms, look-in tables or feedback algorithms, if necessary, in this case a cascade algorithm is implemented (see **Output Linearization**).

The implemented software for the required a virtual instrument was **NI-LabView**, software in which the data was processed.

3 Signal Conditioning

The signal conditioning for the Pt_{100} is given by the circuit in figure 4. It is a deflection bridge in series with a differential amplifier.

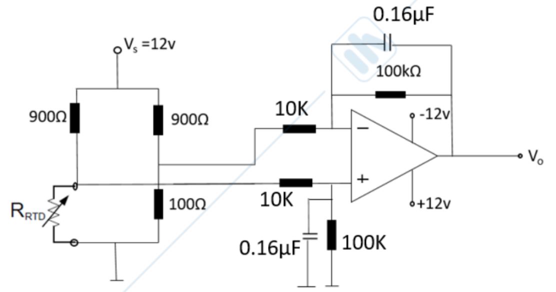

Figure 4: Conditioning Circuit Pt_{100}.

From the deflection bridge voltage dividers, the tension input of each terminal of the differential amplifier are given by:

$$\begin{aligned} V_1 &= \frac{R_{Pt_{100}}}{900\Omega + R_{Pt_{100}}} \cdot 12 \\ V_2 &= 1.2 \\ G &= 100k/10k = 10 \end{aligned} \quad (1)$$

The DC gain from the differential amplifier is $G_{amp} = 10$. Therefore the output tension going in the data acquisition card (DAQ) is:

$$V_o = G \cdot (V_1 - V_2) = \frac{R_{Pt_{100}}}{900\Omega + R_{Pt_{100}}} \cdot 120 - 12 \quad (2)$$

The deflection bridge is calibrated for the Pt_{100} at $T = 0°C$ therefore $R_0 = 100\Omega$. The differential amplifier is supplied by $\pm 12V$. This limits the V_o to a theoretical maximum value of $+12V$. Equation 2 is the first type of non-linearity in the system.

4 Temperature/Resistance relationship

The correlation between the $R_{Pt_{100}}$ and the temperature is given by the quadratic expression:

$$R_{Pt_{100}}(T) = R_0 \cdot (1 + BT + AT^2) \quad (3)$$

With $A = 3.9086 \cdot 10^{-3}\Omega/°C$, $B = -5.775 \cdot 10^{-7}\Omega/°C^2$ and $R_0 = 100\Omega$.

Therefore the output from the deflection bridge will be an implicit function of T.

Due to the second order Polynomial, equation 3 adds an extra non-linearity to solve in the system. Replacing 3 in 2 gives the relationship between the output voltage V_o from the signal conditioning circuit and the measured temperature:

$$V_o = \frac{R_0 \cdot (1 + BT + AT^2)}{900\Omega + R_0 \cdot (1 + BT + AT^2)} \cdot 120 - 12 \quad (4)$$

Figure 5 shows the output from the deflection bridge under two different approaches (see **Section 5**), it is injective and therefore invertible. As $Vo_{max} = 12V$ the maximum theoretically measurable temperature is $336.5°C$, although the requested range was [0-100°C], is useful to study the performance of the mathematical model at higher values of T.

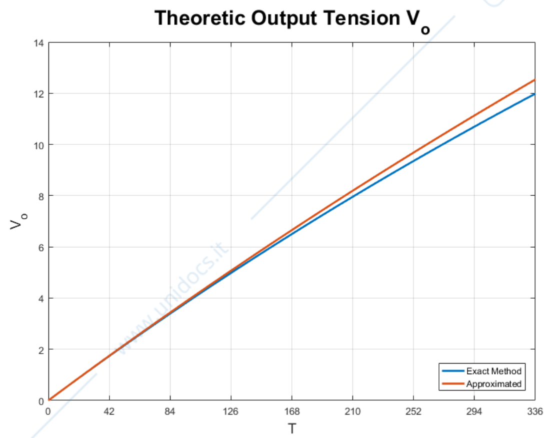

Figure 5: Output from the Conditioning Circuit

5 Output linearization

As seen before $V_o(R_{Pt_{100}}(T))$ is injective. Hence, applying some basic algebra manipulations to equation 2 to find its inverse function $R(V_o)$ gives:

$$R_{Pt_{100}}(T) = \frac{900V_o + 10800}{108 - V_o} \quad (5)$$

Equating equations 3 and 5, the general equation for T is a second order polynomial given by:

$$AT^2 + BT + 1 - \frac{900V_o + 10800}{(R_0 \cdot (108 - V_o)} = 0 \quad (6)$$

Defining:

$$C = 1 - \frac{(900V_o + 10800)}{(R_0 \cdot (108 - V_o))}$$
$$C = 1 - \frac{9V_o + 108}{108 - V_o} \quad (7)$$

Hence the cascade linearization for the output tension $T(V_o)$ is given by:

$$T(V_o) = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad (8)$$

Alternatively, in the range $[0, 100^oC]$ the term AT^2 has a maximum order of magnitude $OM(10^{-3})$, therefore it may be neglected and equation 6 reduces to:

$$BT + 1 - \frac{900V_o + 10800}{R_0 \cdot (108 - V_o)} = 0 \quad (9)$$

From which:

$$T(V_o)_{aprx} = \frac{(900 + R_0)V_o + 10800 - 108R_0}{(108 - V_o)B \cdot R_0}$$
$$T(V_o)_{aprx} = \frac{10 \cdot V_o}{(108 - V_o)B} \quad (10)$$

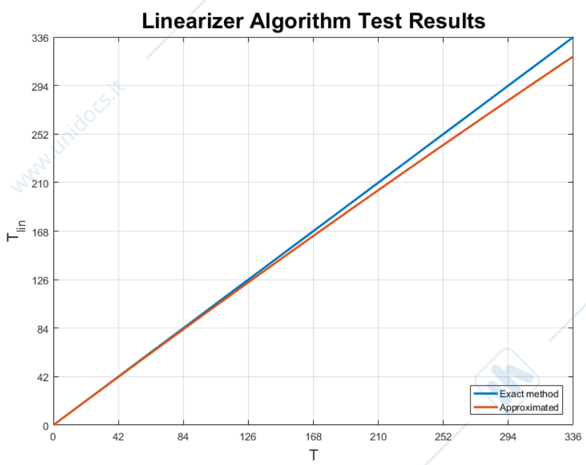

Figure 6: Exact method (blue) vs. Approximated method (Orange)

The error of doing such approximation is given by:

$$\%error(T) = \left| \frac{(T(V_o)_{aprx} - T(V_o))}{T(V_o)} \right| \cdot 100\% \quad (11)$$

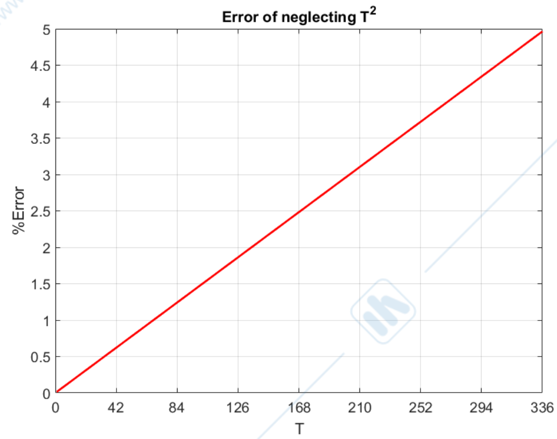

Figure 7: Error of using equation 9

Figure 7 shows how the approximation error grows with the temperature. For the study range, the maximum error is 1.4775% at $T = 100^oC$, this value is not neglectable, for example, if the range of measurements increases to from 100°C to a theoretical maximum of 336°C this error will be increased up to 4.96%. Hence, the "Exact" implementation of the inverse function was chosen as the cascade linearizator.

6 Resolution

The data acquisition card (DAQ) has 16bits precision, the measurement range is $[0 - 100^oC]$ and $[0 - 4V]$. The resolution for any variable X is given by:

$$Res(X) = \frac{Max(X) - Min(X)}{2^n - 1}$$
$$X = T \rightarrow Res(T) = 1.5259 \cdot 10^{-3}[^oC]$$
$$X = V_0 \rightarrow Res(V_0) = 61.2[\mu V] \quad (12)$$

With $n = 16$ the number of bits of the DAQ. However, as the DAQ takes as input the V_0 from the conditioning circuit, therefore the resolution of the DAQ is $61.2[\mu V]$.

7 Digital Filter Design

A digital filter is created by transforming the transfer function of an analog filter from

Laplace domain (S domain) to discrete domain (Z domain) and then again to discrete time domain (n domain).

Starting from the Second Order Butterworth Low Pass filter transfer function:

$$H(S) = \frac{\omega_0^2}{S^2 + \sqrt{2}\omega_0 S + 1} \tag{13}$$

and the bilinear transformation[1]:

$$\begin{aligned} S &\approx \frac{2}{T}\frac{1-Z^{-1}}{1+Z^{-1}} \\ \omega_a &= \frac{2}{T}tan\left(\frac{\omega_0 T}{2}\right) \end{aligned} \tag{14}$$

Replacing 14 in 13 and defining:

$$\begin{aligned} A &= 1 - \sqrt{2}tan\left(\frac{\omega_0 T}{2}\right) + tan^2\left(\frac{\omega_0 T}{2}\right) \\ B &= 2tan^2\left(\frac{\omega_0 T}{2}\right) - 2 \\ C &= 1 + \sqrt{2}tan\left(\frac{\omega_0 T}{2}\right) + tan^2\left(\frac{\omega_0 T}{2}\right) \\ D &= tan^2\left(\frac{\omega_0 T}{2}\right) \\ d &= \frac{D}{C}, a = -\frac{A}{C}, b = -\frac{B}{C} \end{aligned} \tag{15}$$

The filter equation reduces to:

$$H(Z) = \frac{Y(Z)}{X(Z)} = \frac{d(1 + 2Z^{-1} + Z^{-2})}{-aZ^{-2} - bZ^{-1} + 1} \tag{16}$$

With T the sampling period and ω_0 the filter cut-off frequency. Now choosing the right values for T and ω_0 and replacing them in 15 the Z-domain filter is fully designed.

To return to the n-domain, the shifting theorem of the Z-transform is applied many times to 16 to recover the realizations of Y(n) in terms of previous values of both Y(n) and X(n). In consequence:

$$\begin{aligned} Y(n) =& aY(n-2) + bY(n-1) + \\ & dX(n-2) + 2dX(n-2) + dX(n) \end{aligned} \tag{17}$$

Temperature is a slow changing phenomena. Taking a sampling rate of T=0.005s and a cut-off frequency of 15Hz the filter reduces to:

$$\begin{aligned} Y(n) =& -0.51398Y(n-2) + 1.348968Y(n-1) \\ & + 0.041254[X(n-2) \\ & + 2X(n-1) + X(n)] \end{aligned} \tag{18}$$

With $a = -0.51398$, $b = 1.348968$ and $d = 0.041254$.

Taking too many samples would make the digital filter to collapse due to the frequency transformation. As see from 14 the frequency transformation is proportional to $Tan(T)/T$, therefore the smaller the sampling period the closer ω_a to ω_0. The problem is that as T decreases the required numerical precision increases, adding rounding errors to the filter output and thus corrupting the final output.

8 Virtual Instrument

A virtual instrument was designed using NI-LabView.

Figure 8 shows the front panel of the designed instrument with the controls for its operation. It includes the different outputs from the raw data and the digital processed data, in both time domain and frequency domain in order to prove the how effective is the DSP. The instrument includes as well as save and stop buttons.

The temperature input is controlled by a scroll bar that modifies the signal's off-set, it can be also altered by modifying the amplitude of the signal fluctuations creating a sinusoidal wave. The noise has an amplitude of 1^oC and it is added to the signal as a sinusoidal wave controlled by its frequency.

The instrument includes user controlled limits for both high and low temperature, and the 'raw temperature' value and the processed value in two displays. The output from the instrument is a file in 'csv' format.

[1] See appendix for its derivation

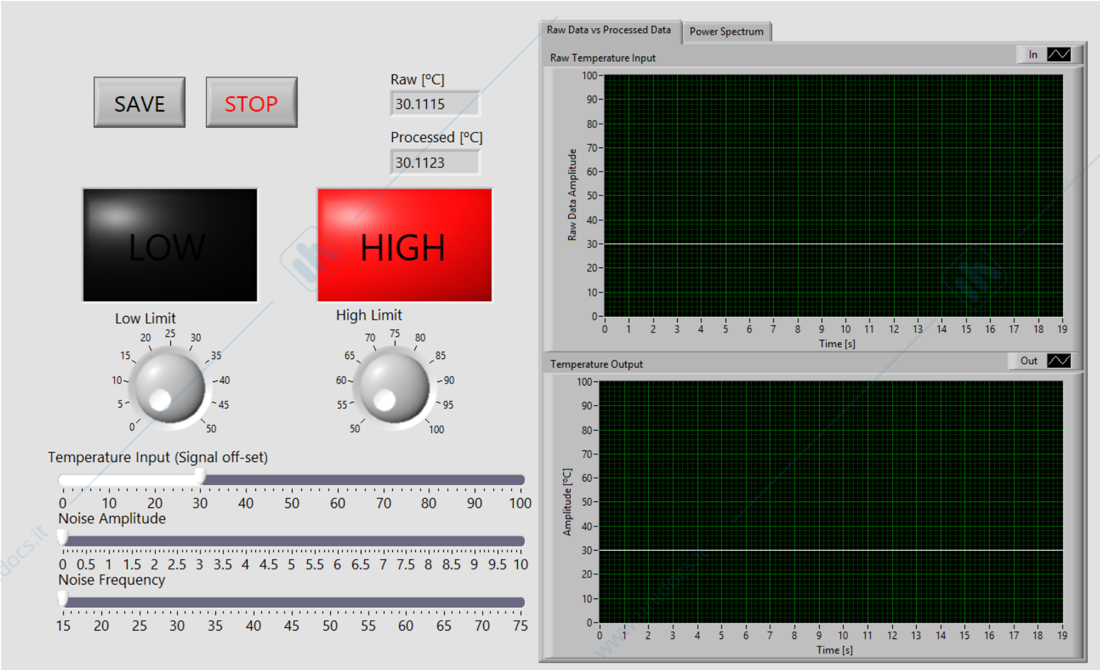

Figure 8: Virtual Instrument

It consists in three parts:

8.1 Signal Generation

A temperature carrying signal was generated using a sinusoidal wave and modifying its offset, amplitude and frequency. This signal goes through the conditioning circuit equations (in digital form) to generate a voltage signal.

8.2 Filtering

A frequency controlled noise was added to the signal to further implement the DSP. The voltage signal goes through a digital filter to remove almost all of this noise.
Figure 9 shows the Block Diagram code for the digital filter implementation.

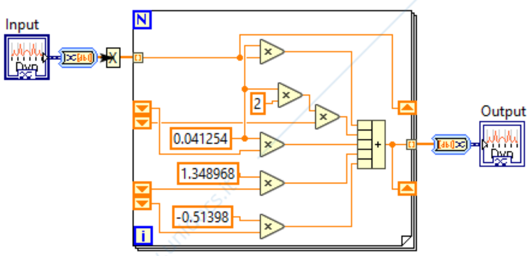

Figure 9: Filter Block Diagram (LabView).

The filter efficacy is then proved by comparing the results from the spectral measurements (Power Spectrum) of the signal before and after the filter application, as well as comparing the numerical value of the output temperature after linearization with the expected temperature value.

8.3 Linearization

The algorithm previously derived in section 5 is implement in digital form to reconstruct the temperature signal. The block diagram of its LabView structure is shown in Figure 10.

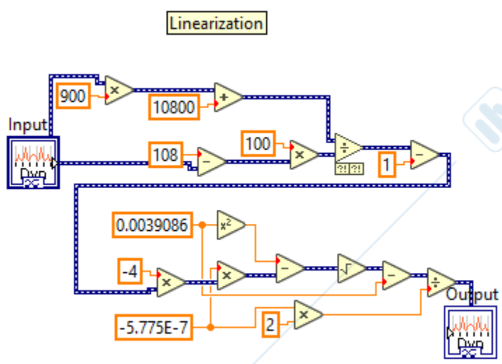

Figure 10: Filter Block Diagram (LabView).

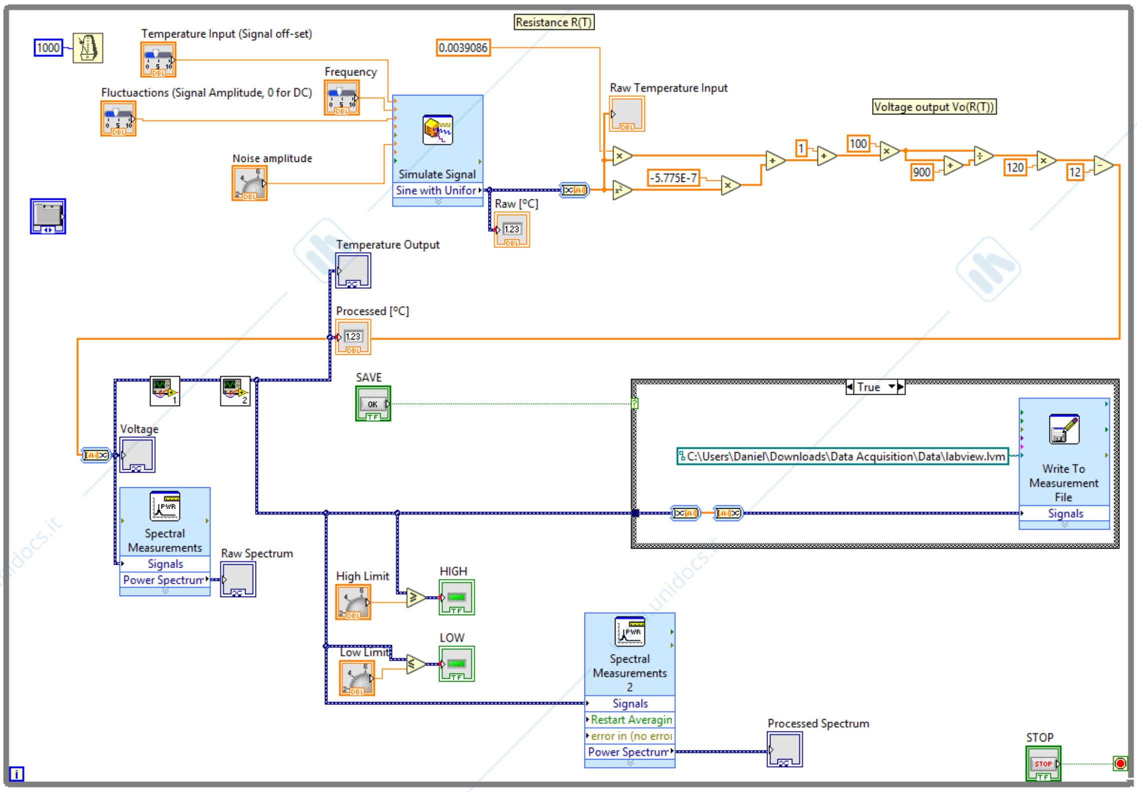

Figure 11: Block Diagram of the whole LabView instrument.

Figure 11 shows the whole block diagram code structure for the digital instrument as it was programmed in LabView. In the other hand figures 9 and 10 show the different internal implementations of the numerical models in LabView.

9 Spectral Analysis

It is the analysis of the signal from the frequency point of view, that is, applying a spatial transformation that drives the signal from the time domain to the frequency domain through the means of the Fourier Transform. The Fourier Transform is defined as:

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-j\omega t}dt \qquad (19)$$

For discrete applications the Fourier Transform is instead replaced by the Discrete Fourier Transform (DFT).

$$F(k) = \sum_{n=0}^{N-1} F(n)e^{-j(2\pi/N)kn} \qquad (20)$$

And it is computed through the algorithm of the Fast Fourier Transform.

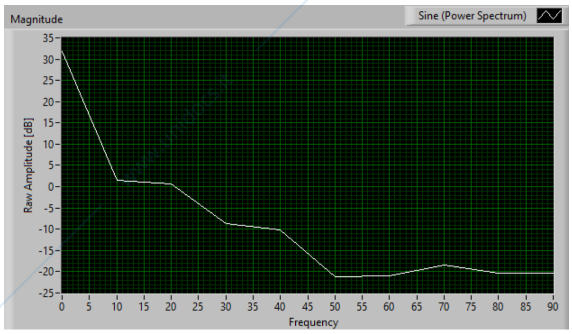

Figure 12: Power Spectrum of a Discrete Signal (LabView)

As the coefficients of the DFT are complex exponential, they are both symmetric and periodic. Therefore, by taking advantages of those properties it is sufficient to compute half of the DFT terms (N/2) twice to achieve the transformation, these properties are the basis of the so called Fast Fourier Transform. Only half of the frequency spectrum can be recovered this way, but due to symmetry it is sufficient to realize the whole system. One of the most widely used algorithms to compute a FFT is called the Butterfly Implementation.

The advantage of mapping a discrete signal in the frequency domain lies in the possibility to analyze its power and phase spectrum, that is, the amount of energy and phase angle of each energetic component of the signal for each of the different frequencies within it.

Each of the different 'k' from equation 20 are amplitude components of F(n) for different values of frequency.

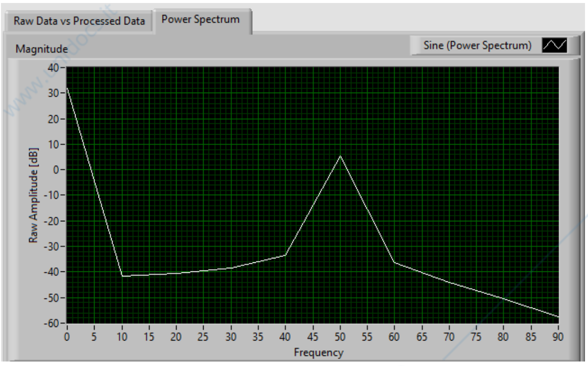

Figure 13: Power Spectrum of a Discrete Signal with 50Hz high energy spectra (LabView)

Once the power spectrum is fully realized a digital filter can be designed to properly remove energy from specific frequency ranges.

10 Digital Filter Outputs

The digital filter from equation 13 was implemented as shown in figure 9. The virtual implementation in LabView of the filter required the use of a 'For' loop structure and the use of the 'Shift-register' to access values from previous iterations.

Some test were be applied to the filter a priory to prove the filter's efficacy to remove frequency components.

Recalling equation 13 and replacing the coefficient values:

$$H(Z) = \frac{0.041254(1+2Z^{-1}+Z^{-2})}{0.51398Z^{-2}-1.348968Z^{-1}+1} \quad (21)$$

This is the filter in Z domain. Now replacing Z with:

$$Z = e^{j\omega T_s} = e^{j2\pi f/f_s} \quad (22)$$

With $f_c = 15Hz$ and $f_s = 200Hz$.

The following points are easy to evaluate: $f = 0 \rightarrow Z = 1$, $f = 0.5f_s \rightarrow Z = -1$. The most important point to evaluate is $f = f_c$ because for a Butterworth filter, for $f = f_c$, H(f)=0.707.
The first two points have immediate solution:

$$H(Z=1) = \frac{0.041254(4)}{0.51398-1.348968+1} = 1$$
$$H(Z=-1) = \frac{0.041254(0)}{0.51398+1.348968+1} = 0 \quad (23)$$

The third point is not straightforward as $exp(j2\pi 15/200)$ is a complex number with both real and imaginary parts. Therefore:

$$H(e^{j0.15\pi}) = \frac{0.041254(1+2e^{-j0.15\pi}+e^{-j0.3\pi})}{0.51398e^{-j0.3\pi}-1.348968e^{-j0.15\pi}+1}$$
$$H(f_c) = -j0.707043$$
$$||H(f_c)|| = 0.707043 \quad (24)$$

Therefore in theory the filter should work perfectly removing energy from frequencies $f >> f_c$. This hypothesis was tested implementing the filter in the virtual instrument (figure 9).

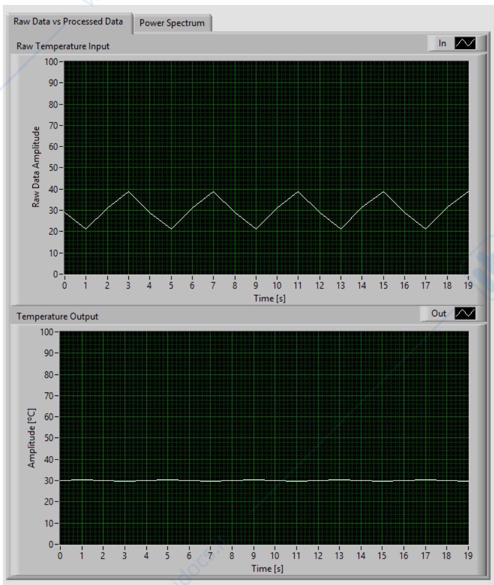

Figure 14: No-filtered/Filtered comparison in time domain.

Figure 14 show the comparison between the non filtered output and the post filtering operation for a DC temperature signal of 30°C and a 50Hz, triangle wave, 10°C Amplitude, noise. Notice how the filter reduces the spikes to an almost smooth DC output that shows clearly the measured temperature.

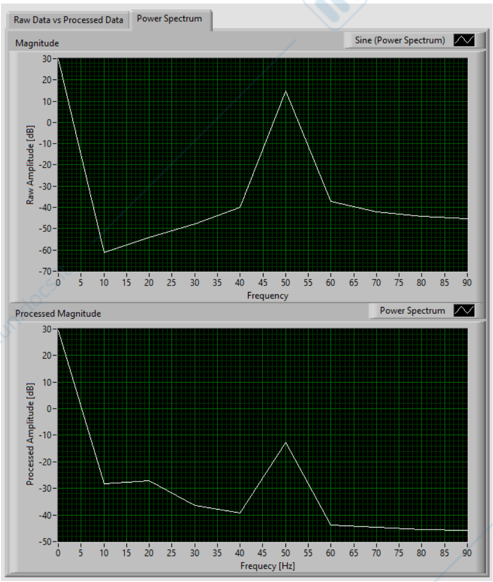

Figure 15: No-filtered/Filtered frequency power spectrum.

Figure 15 instead show the frequency spectrum for the same measurement in figure 14. The power spectrum is expressed in decibels (dB). Notice how the frequency components near to 50Hz are attenuated from $+20dB$ to below $-10dB$, justifying the results from figure 14.

11 Physical Circuit

Although it was not used, a the physical circuit of the conditioning circuit from 4 was built and tested.

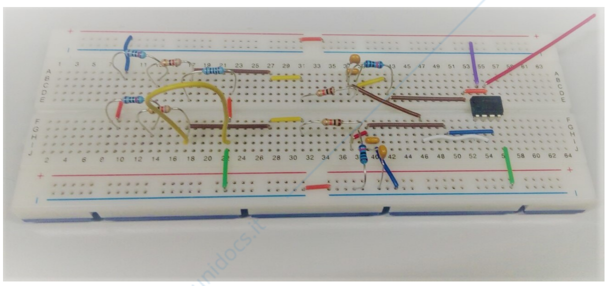

Figure 16: Conditioning Circuit for the temperature sensor

12 Conclusions

The steps of the digital signal processing and its major features were exposed. Showing clear advantages respect to the Analogue Signal Processing as it does not require any physical components, and therefore it does not suffer from aging. It also show some disadvantages, as some information is lost due to the signal discretization.

The circuit equations were derived and its non-linearities were both exposed and corrected though a cascade implementation of an inverse function linearizator. The overall system can be seen as a linear single input single output.

A digital filter based on a second order Butterworth low pass filter was built by the means of the bilinear transformation. All the filter coefficients were determined and it was simplified for a clear representation in 'n' domain.

The filter was fully implemented and tested and the results were in according with theory, obtaining, unit DC gain, and $1/\sqrt{2} \approx 0.707$ gain for its cutoff frequency. The filter coefficients were chosen as $f_c = 15Hz$ and $f_s = 200Hz$ due to the fact that temperature is a slow changing phenomena and therefore it requires time to develop, and as well that the highest the sampling frequency and the lowest the cut-off frequency the filter coefficients require more numerical precision to be implemented, introducing numerical errors in the calculations.

A frequency controlled noise was injected to the signal to prove the filter efficacy. The filter showed the capacity to remove great part of the noise as shown in by the pre/post filtered amplitude spectra.

The LabView's Virtual Instrument was successful tested for simulated signals with excellent results. The front panel is fully controllable for an operator for temperature related simulations. With some minor changes it can be fully operative for real measurements. The outputs of the instrument are fully saved in '.csv' format.

13 Bibliography

- Shenoi, B.A. (2006), *'Introduction to Digital Processing and Filter Design'*, United States, Wiley-Interscience

14 Appendix

14.1 Derivation of the bilinear transformation

Given $S = j\omega$ and $Z = exp(ST)$ the procedure to derive the bilinear transformation from the S domain to the Z domain is as follows:

$$Z = e^{ST} = \frac{e^{S\frac{T}{2}}}{e^{-S\frac{T}{2}}} \tag{25}$$

A power series expansion of the exponential function drives to:

$$Z = e^{ST} = \frac{1 + S\frac{T}{2} + \frac{(S\frac{T}{2})^2}{2!} + ...}{1 - S\frac{T}{2} + \frac{(S\frac{T}{2})^2}{2!} + ...} \tag{26}$$

Neglecting terms with order $>= 2$ under the assumption of $|ST/2| << 1$, that is, $(ST/2)^n \rightarrow 0$ for $n >= 2$. Equation 27 is reduced to:

$$\begin{aligned} Z &= \frac{1 + S\frac{T}{2}}{1 - S\frac{T}{2}} \\ S &= \frac{2}{T}\frac{1 - Z^{-1}}{1 + Z^{-1}} \end{aligned} \tag{27}$$

That is the bilinear transformation of the 'S' domain to the 'Z' domain.

Now replacing $S = j\omega_a$ and $Z = exp(j\omega_c T)$.

$$\begin{aligned} j\omega_a &= \frac{2}{T}\frac{1 - e^{-j\omega_c T}}{1 + e^{-j\omega_c T}} \\ j\omega_a &= \frac{2}{T}\left(\frac{e^{-j\omega_c T/2}}{e^{-j\omega_c T/2}}\right)\frac{e^{j\omega_c T/2} - e^{-j\omega_c T/2}}{e^{j\omega_c T/2} + e^{-j\omega_c T/2}} \\ j\omega_a &= \frac{2}{T}\frac{2}{2}\frac{sinh(j\omega_c T/2)}{cosh(j\omega_c T/2)} = j\frac{2}{T}\frac{sin(\omega_c T/2}{cos(\omega_c T/2)} \end{aligned} \tag{28}$$

Finally, from equation 29:

$$\omega_a = \frac{2}{T}tan\left(\frac{\omega_c T}{2}\right) \tag{29}$$

Therefore, the results from equations 28 and 30 prove the validity of the bilinear transformation. Q.E.D.

14.2 Output from the Virtual Instrument

When pressing the SAVE button in the front panel from figure 8 an output file in '.csv' format is saved.

22.21227
21.93422
21.92056
22.14672
22.23669
22.06268
21.91607
22.06462
22.35012
22.3454
22.10465
22.01888
22.19184
22.30992

Figure 17: Example of the output file.

Figure 17 shows how does the 'Data.csv' file look like when open using MS Excel. Every row represent a processed temperature measurement value.