Implementation of extensometer calibration and decimation filtering on Campbell Scientific CR10X datalogger

M. Bednárik, L. Brimich
Geophysical Institute of the Slovak Academy of Sciences[1]

Abstract: The quartz tube extensometer and associated electronics at the Vyhne tidal station operate in very unfriendly conditions of 100% humidity, which causes corrosion of metal parts (suspensions, wirings etc.), often leading to disturbances or complete dropouts of measurements.

Response of the instrument to the regular calibration pulses serves not only proper scaling of the output and easy orientation in the data stream but also the diagnostics of health of the system and the discrimination of disturbed (thus not suitable for geophysical interpretation) portions of the signal.

On the other hand, the calibration pulses themselves *disturb in some extent* the standard measurements, too. To minimize the extent of influence of the calibration switched on on standard data (and calibration switched off on calibration data) is, when one has to confine himself to the meager instruction set of the Campbell Scientific CR10X datalogger, an interesting engineering task.

Key words: quartz tube extensometer, calibration, CR10X datalogger, time multiplex, decimation, finite impulse response filter design

1. Introduction

The quartz tube extensometer at the Vyhne tidal station (Fig. 1) is calibrated by means of a magnetostrictive coil, which displaces the whole tube when the calibration current flows through it. Details on the configuration of the station can be found in *Dudášová (1998)*, the process of calibration is explained e.g. in *Mentes (1993)*. The response of the instrument to the rectangular calibration current pulse shows transients (Fig. 2), which have

[1] Dúbravská cesta 9, 845 28 Bratislava, Slovak Republic
e-mail: geofmabe@savba.sk; geofbrim@savba.sk

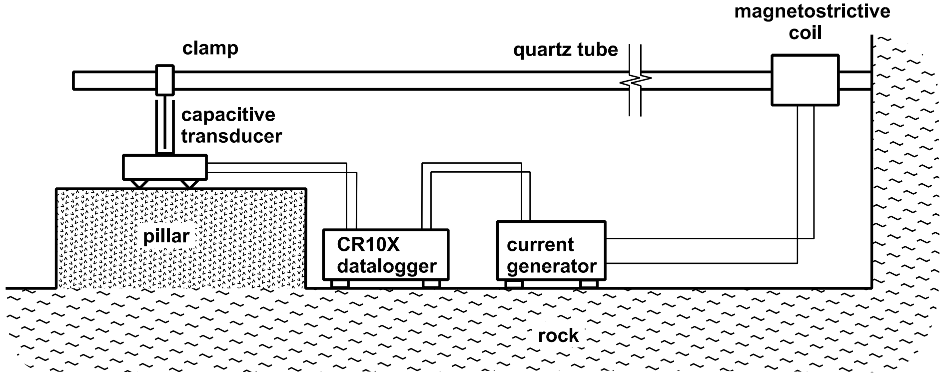

Fig. 1. Scheme of the extensometric equipment in Vyhne.

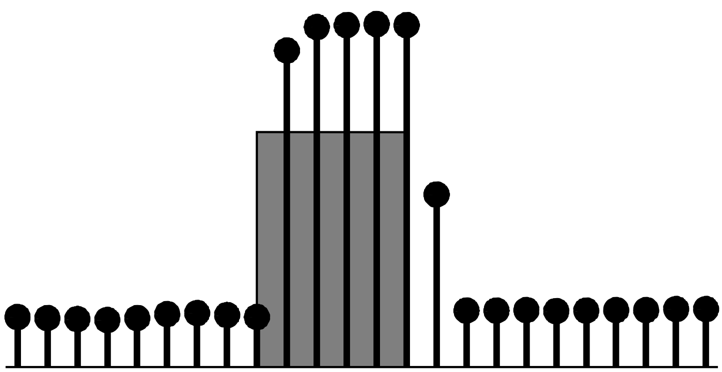

Fig. 2. Transients in extensometric signal (sampling period 1s) in response to 5s long rectangular calibration current pulse (gray rectangle). The record was made in Vyhne on November 17, 2005 just for the purpose of measuring the duration of transients. The standard duration of current pulse is 1 minute.

to be taken into account: the signal – either calibration response or standard extensometric signal – should not be sampled before the transients are effectively over. The usual sampling period of extensometric data for tidal analysis ranges from 10 minutes to 1 hour. In Vyhne, we use 10-minute period.

A typical sampling period of calibration data is 12 or 24 hours – the calibration period then corresponds to the dominant tidal waveform periods and the pulses provide nice frame ticks for easier overview – completely indispensable in the times of analogue recordings, but still keeping a lot of importance in the age of digital data. An example of an unfiltered 24 hours long record of extensometric data from Vyhne with sampling period 10 minutes, including the calibration pulses is given in Fig. 3.

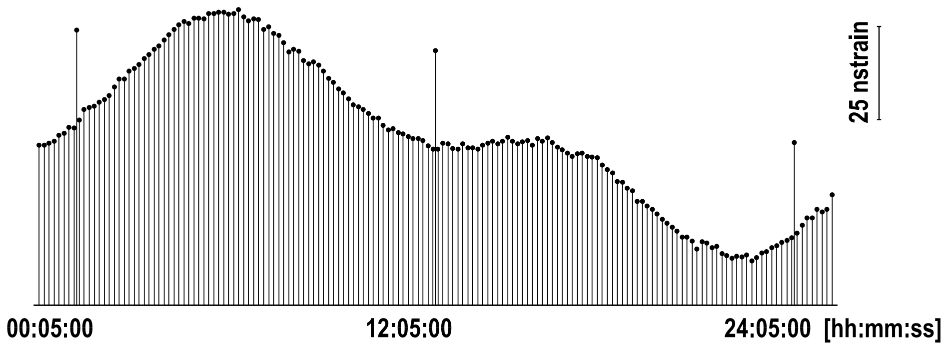

Fig. 3. Extensometric record (sampling period 10 minutes) made in Vyhne on November 29, 2005.

2. Time multiplex

Time multiplex is in our context a bit lofty terminus technicus for quite a trivial, nevertheless very advisable for those who do not use it yet, a “trick”:

If there are – in time domain – gaps in your data, then you can get other data within these gaps.

For instance, the calibration period 720 minutes is an integer multiple of the standard data period 10 minutes. One approach to it could be just to sample every 10 minutes and at each 72-nd sample to turn on and off the calibration current to get the calibration sample. Like this, however, we would, strictly speaking, loose every 72-nd sample of standard data stream (Fig. 4). Of course, we could “reconstruct” it by interpolation of the neighbouring samples, but why to do it when there is a cleaner way to go?

If we shift the sampling of the calibration data by the half of the standard data sampling period, we can have both data streams undisturbed (Fig. 5).

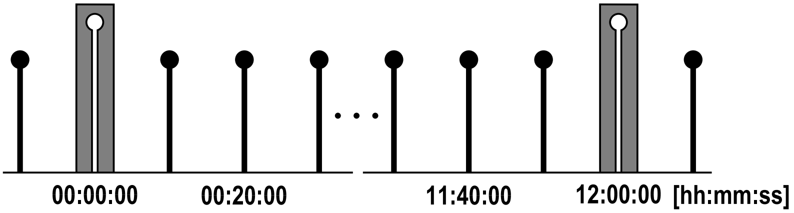

Fig. 4. Bad solution: Calibration samples (white bars) without time multiplex.

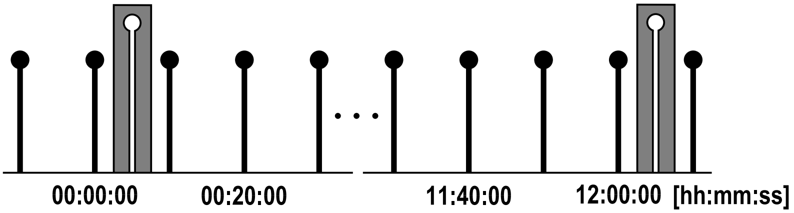

Fig. 5. Good solution: Calibration in time multiplex with standard samples.

3. Decimation

It gets a little bit more complicated if we want to improve the quality of our data by means of decimation i.e. first taking input samples with high rate and then downsampling it - at once or in more stages – to the low output sampling rate. Let us assume that by sampling with period T_1 we do not violate the Nyquist criterion, i.e. the spectrum of the analogue signal to be sampled has zero content above Nyquist frequency $\omega_{N1} = \omega_1/2 = \pi/T_1$. If we want to decimate this - once already sampled - signal down to a longer sampling period T_2, we have to filter the digital data with a digital lowpass filter whose corner frequency should be lower or equal to ω_{N2}, $\omega_{N2} = \omega_2/2 = \pi/T_2$. For the digital data sampled with the period T_1, we use the amplitude frequency plot with normalized frequency $\theta = \omega T_1$. The highest corner frequency $\omega_{N2} = \pi/T_2$ of the filter assuring non-violation of the Nyquist criterion is thus projected to $\theta_{N2} = \pi\, T_1/T_2$. The amplitude frequency response of an ideal lowpass decimation filter *(Ondráček, 2002*, p. 123) is then shown in Fig. 6. In general, the higher is the decimation ratio T_2/T_1, the harder it is to design the filter. For high decimation ratios, the decimation is usually performed in a cascade of stages.

The main problem with (whether time-multiplexed or not) calibration pulses is that they limit the length of time interval where the input samples for the decimation can be collected: the filtered sample *must not* be computed of a *mixture* of standard and calibration input samples, but only of standard data input samples for each standard data sample or calibration data input samples for each calibration data sample (Fig. 7). It is true that in our case, the calibration sample occurs 72 times less frequently than standard sample - we could handle the standard samples neighbouring to the calibration sample specially and differently to other standard samples. We think, however, that in favour of the homogeneity of the data it is much wiser to compute all the standard samples always the same way. The duration of the calibration current switched on should not exceed a few minutes in order not to overheat the magnetostrictive coil. To have at least tens of input samples available for filtering, the period T_1 should be in order of *seconds*. Then, strictly and formally, with the output sampling period T_2 of 12 *hours*, we have a really big decimation ratio for which it is impossible to find a proper anti-aliasing decimation filter with tens of coefficients length.[2]

In standard data alone, sampled with output rate 10 minutes, in order to leave some time for the calibration pulse, the interval available for the collection of input samples for standard sample is ca. 8 minutes long. Should we take input sample every second, we would have ca. 480 samples available, requiring the decimation filter of the same length for still very high 600:1 decimation ratio. Even with this filter length, the highest ratio for which we are able to find one-stage truly anti-alias decimation filter is about 60:1.

Shortly, the need to take calibration samples forces us to abandon com-

[2] The calibration data is, in fact – with regard to its interpretation – not an independent data stream: the height of the calibration pulse can only be determined when the neighbouring standard data samples are taken into account, as well. Merging together standard data sampled each 10 minutes and calibration data in time multiplex with it as described in section 2, we get each 12 hours a 3-sample rag of data with "instantaneous" sampling period 5 minutes (Fig. 3 and 4) – which is a lot anyway.
We could ask for homogeneity – the same way of computation of both standard a calibration output data – within these rags of merged data, too. This would lead, however, to the same length of intervals where the input samples of the two types of data are colected (ca. 4 minutes), which is technically unbearable (overheating of coil by calibration current) and does not correspond to the higher importance of standard data.

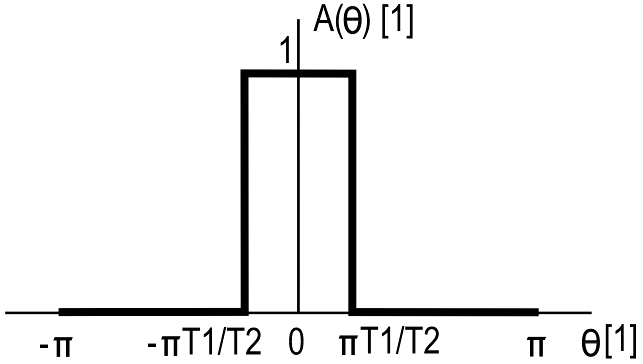

Fig. 6. Amplitude frequency response of an ideal lowpass filter.

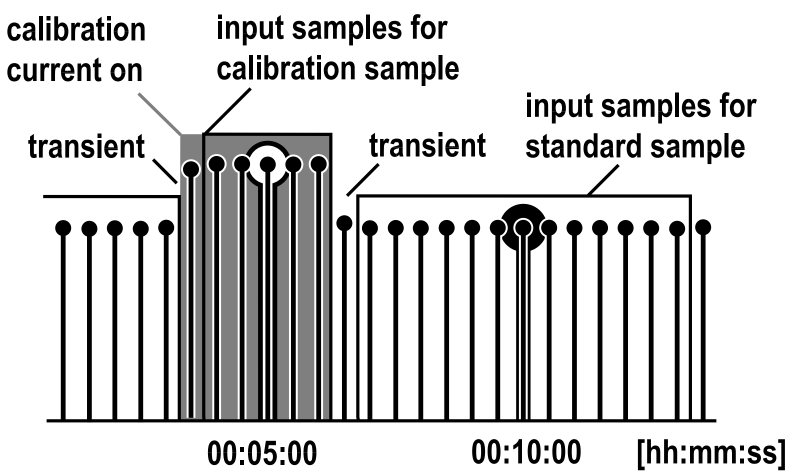

Fig. 7. Input samples for computing the calibration and standard samples (sampling period $T_1 = 30$ s).

pletely the aspirations to a downsampling in accordance with the Nyquist criterion. Fortunately, at as a quiet site as Vyhne and with the analogue input properly conditioned, the high frequencies have quite low amplitudes, except for occasional seisms. And, even if a decimation filter is too far from being anti-alias one, it is still true that the narrower (in frequency domain) the filter, the better. And surely better than no filter at all.

4. Suggestions for decimation filter design

The input signal $x(t_m)$ is sampled at times $t_m = t[m] = mT_1$, where m is integer. For brevity, $x(t_m)$ is denoted as $x[m]$. The filtered output signal $y[m]$ is obtained by convolution of the input signal $x[m]$ with the impulse response $h[n]$ of the filter – in other words, with the sequence of filter coefficients:

$$y[m] = \sum_n h[n]\, x[m-n]. \tag{1}$$

By filtering, we would like to improve the signal-to-noise ratio of the signal without any undesired side effects (e.g. phase distortion). Therefore, we restrict our choice to the cathegory of *symmetric* ($h[n] = h[-n]$) *finite* ($h[n] = 0$ for $n \notin \{-(L-1)/2, ..., (L-1)/2\}$, $L \in N$ is the filter length) impulse response (FIR) filters[3], whose frequency response

$$H\left(e^{j\theta}\right) = \sum_{n=-\infty}^{\infty} h[n] e^{-jn\theta} \tag{2}$$

is real, i.e. its continuous phase frequency response *(Ondráček, 2002*, pp. 115, 160) is zero.
It is obvious that the value of the filtered output sample

$$y[m] = \sum_{n=-(L-1)/2}^{(L-1)/2} h[n]\, x[m-n] \tag{3}$$

is influenced by input samples with $n < 0$, i.e. input samples "from the future" (if we identify "present" with the time m) – therefore, such a filter is called *non-causal.* Closely related problem is the *time shift* of the output sample: to compute $y[m]$, one has to wait until $x[m+(L-1)/2]$ occurs.

With the non-causality, there is not too much to do but simply to bear it. As far as the signal to be filtered is smooth, without abrupt steps, as it is the case with tidal extensometric signal, it is not a serious practical problem. The time shift, however, has to be properly managed: although $y[m]$ can not be output *before* the time $m+(L-1)/2$, it can and must be output

[3] Another advantages of FIR filters are stability and low sensitivity to the inaccuracies in entering the coefficient values.

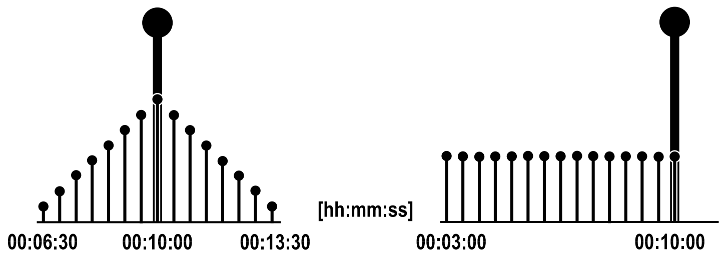

Fig. 8. Filter coefficients $h[n]$ of input samples in the vicinity of the filtered output sample. Left: good example, right: bad example (moving average computed only of previous input samples and output at the time of last input sample).

with the right time m. This may seem trivial, but bad timing of the filtered (averaged) output (Fig. 8 right) still surprisingly often occurs in practice. At each time we want to get the filtered sample, we are allowed to reset the "sample counter" m to zero, and, taking into account the symmetry of the desired filter, we can write:

$$y[0] = \sum_{n=-(L-1)/2}^{(L-1)/2} h[n]\, x[-n] = \sum_{n=-(L-1)/2}^{(L-1)/2} h[-n]\, x[-n] = \\ = \sum_{n=-(L-1)/2}^{(L-1)/2} h[n]\, x[n]. \tag{4}$$

The rule "the narrower the filter, the better" in the previous section is a bit simplified. The plots of the amplitude frequency response reveal that with any filter design method, narrowing the passband of the filter has got its cost in the increase of ripples in the stopband. A more adequately formulated task would be then to find the filter with the narrowest possible passband, whose stopband ripples are lower than an acceptable level.

An overview of the basic filter design methods can be found in *Ondráček (2002)* and *Vích and Smékal (2000)*. Those interested in more rigorous filter design can find an overview of relationships between width and shape parameters and minimum stopband attenuation for various windows in *Sharma et al. (2004)*. We will give here an example of FIR filter design procedure that leads to satisfactory results. It is based on weighted least-squares polynomial fitting, i.e. finding the values of polynomial coefficients a_k minimizing the sum E of weighted squares of residuals:

$$E = \sum_{n=-(L-1)/2}^{(L-1)/2} w[n] \left(\sum_{k=0}^{K} a_k t[n]^k - x[n] \right)^2, \tag{5}$$

where L is the length of the filter and K is the order of the approximating polynomial

$$P_K[n] = \sum_{k=0}^{K} a_k \, t[n]^k.$$

A weighting formula very suitable for our purpose (for narrow pass- and transition band, the side lobes have to be relatively small) is the one giving the coefficients of the Kaiser window (*Ondráček, 2002*, p. 188, gives the formula for another range of n):

$$w[n] = \frac{I_0\left(\beta\sqrt{1-\left(\frac{2n}{L-1}\right)^2}\right)}{I_0(\beta)}, \quad n \in \{-(L-1)/2, ..., (L-1)/2\}, \tag{6}$$

where

$$I_0(x) = \sum_{k=0}^{\infty} \left(\frac{1}{k!} \left(\frac{x}{2} \right)^k \right)^2 \tag{7}$$

is the modified Bessel function of order 0.

As for $t[0] = 0$ (time we want to take the filtered sample in), $P_K[0] = a_0$, we are interested only in finding the value of this one coefficient a_0, which is equal to the sought filtered output sample $y[0]$. In the case of $K = 0$,

$$a_0 = \sum_{n=-(L-1)/2}^{(L-1)/2} w[n]\, x[n] \Bigg/ \sum_{n=-(L-1)/2}^{(L-1)/2} w[n], \tag{8}$$

thus the filter coefficients $h[n]$ are equal to the normalized Kaiser window weights:

$$h[n] = w[n] \Bigg/ \sum_{n=-(L-1)/2}^{(L-1)/2} w[n]. \tag{9}$$

The classical windowing method – multiplication of the ideal filter impulse response by the window sequence (in time domain) - gives in this case of very narrow (given by the high decimation ratio) band lowpass filter almost the same result. The amplitude frequency response of normalized Kaiser window for β =8 and filter length L =23 is shown in Fig. 9.

The formulae for computing filter coefficients for the 2-nd and 4-th order polynomial fits are not too hard to find.[4] For $K = 2$:

$$h[n] = \frac{w[n]\left(S_4 - S_2 t[n]^2\right)}{S_0 S_4 - S_2^2}. \tag{10}$$

For $K = 4$:

$$h[n] = \frac{w[n]\left(S_4 S_8 - S_6^2 + \left(S_4 S_6 - S_2 S_8\right) t[n]^2 + \left(S_2 S_6 - S_4^2\right) t[n]^4\right)}{S_4^3 - 2 S_2 S_4 S_6 + S_0 S_6^2 - S_0 S_4 S_8 + S_2^2 S_8}, \tag{11}$$

where

$$S_l = \sum_{n=-(L-1)/2}^{(L-1)/2} w[n]\, t[n]^l, \quad t[n] = nT_1, \quad n \in \{-\left(L-1\right)/2, ..., \left(L-1\right)/2\}.$$

The respective amplitude frequency response for the same Kaiser weighting and the same filter length are shown in Figs 10 and 11. With increasing K, the filter gets wider and flatter in the passband, only slightly steeper in the transition, but more ripply in the stopband. Even without using the "frequency domain vocabulary", we can say that it is not desirable to go to very high orders, as with increasing order, the fitting polynomial is "more able" to reproduce the raw signal $x[n]$, which is not the aim of our filtering. For comparison, we include the plot of amplitude frequency response of the moving average filter of the same length as in previous examples to discourage of its use in spite of its very easy implementation (Fig. 12).

5. Implementation

We will split this section into more parts, each one covering one particular feature implemented in the *.csi program code as well as some related "hardware solutions". With regard to the compatibility, we use the instruction

[4] The sofware Mathematica can provide a lot of help both in derivation of the formulae and in computing the actual values of impulse response sequences.

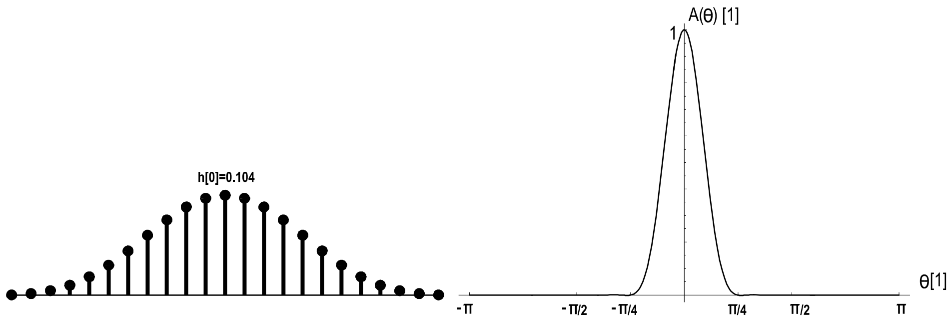

Fig. 9. Impulse response and amplitude frequency response of 0-th order polynomial fitting filter with Kaiser weighting ($\beta = 8$, filter length $L = 23$).

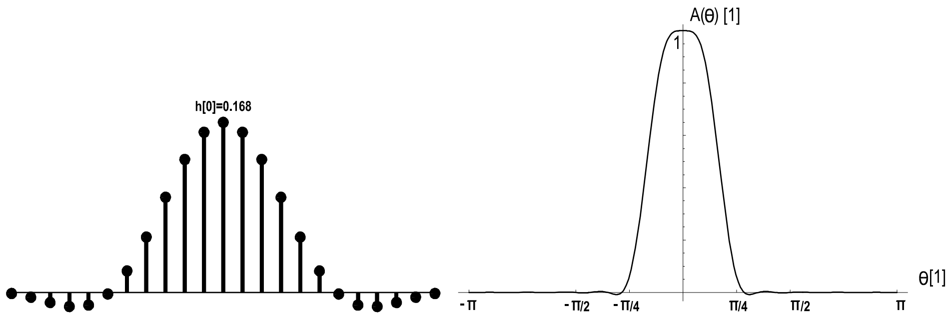

Fig. 10. Impulse response and amplitude frequency response of 2-nd order polynomial fitting filter with Kaiser weighting ($\beta = 8$, filter length $L = 23$)

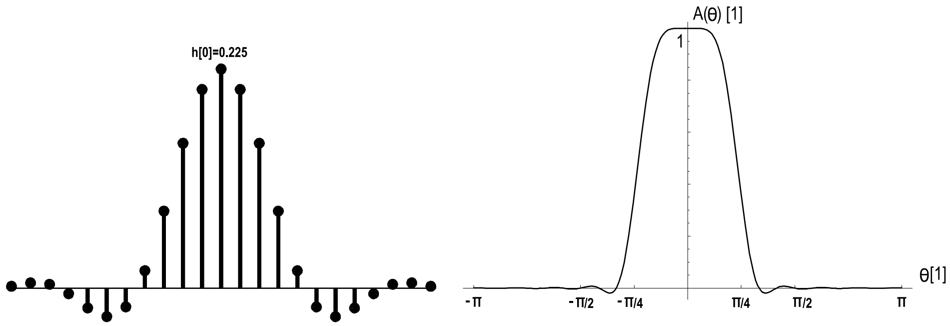

Fig. 11. Impulse response and amplitude frequency response of 4-th order polynomial fitting filter with Kaiser weighting ($\beta = 8$, filter length $L = 23$)

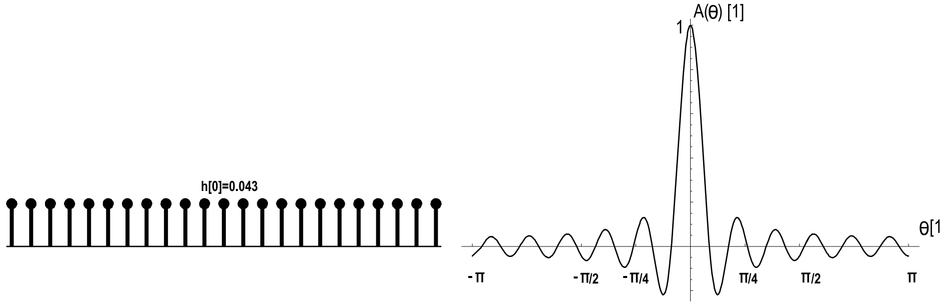

Fig. 12. Impulse response and amplitude frequency response of the moving average filter (filter length $L = 23$).

set of older datalogger type CR10, which is completely understood (unlike the opposite way!) by CR10X datalogger. We will refer to the instructions by their numbers, the interested reader can use the Edlog program editor provided by Campbell Scientific to see what they mean.

Switching on and off the calibration current

For this purpose, the digital I/O ports C1 to C8 of the CR10X can be used. They can be configured as outputs and set high (5V with no load) or low (0V) by some program instructions and commands to be mentioned later. These ports, however, have a limited drive capability (1.5 mA at 3.5V) *(CR10X Measurement and Control Module, 2000*, p. 14-6), whereas the current to the magnetostrictive coil during the calibration shall be ca. 60 mA and strictly constant. Therefore, additional circuitry is needed to generate and switch on and off the calibration current. In our case, some components of previous timing and calibration system were used. Of the eight control ports, we only use the port C1 (Fig. 13). If one wants to set only one port high or low, then it is more elegant to do by means of instructions 83, 86, 88, 89, 91, 92 rather than by the "full form" instruction 20. The following piece of program switches the calibration on in due time:

```
17: Do (P86)
1: 4    Call Subroutine 4 ; GIVES MINUTES MODULO CALIBRATION PERIOD

18: If (X<=>F) (P89)
1: 13    X Loc [ modmin ]
2: 1 =
```

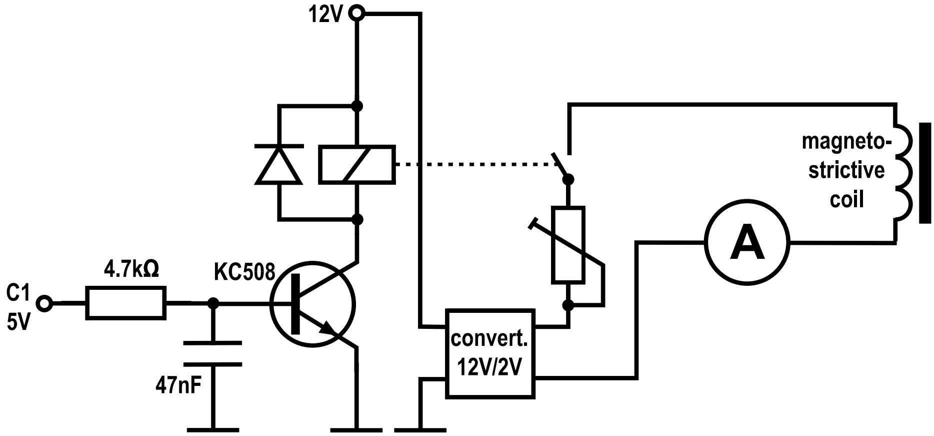

Fig. 13. Simple circuitry for switching on and off the calibration current.

```
3: 4    F ; IN WHICH MINUTE TO SWITCH ON CALIBRATION
4: 30   Then Do

19: If (X<=>F) (P89)
1: 12   X Loc [sec ]
2: 1 =
3: 0    F ; IN WHICH SECOND TO SWITCH ON CALIBRATION
4: 41   Set Port 1 High ; SWITCHES CALIBRATION ON

20: End (P95)
```

After the last calibration input sample was taken, the program calls the output subroutine, which is (in favour of brevity of program) common for both standard and calibration output. Within this subroutine, however, it has to be found out, whether standard or calibration sample is to be output. This can be done easily – we can read the state of the control port as any other flag: During the calibration, it is high, during standard sampling, low. To terminate the calibration pulse, the port has to be set low.

```
41: Do (P86)
1: 10   Set Output Flag High (Flag 0)

42: If Flag/Port (P91)
1: 51   Do if Port 1 is Low
2: 30   Then Do

43: Set Active Storage Area (P80)
```

```
1: 1    Final Storage Area 1
2: 1    Array ID ; "1" LABELS THE STANDARD SAMPLE

44: Else (P94)

45: Set Active Storage Area (P80)
1: 1    Final Storage Area 1
2: 2    Array ID ; "2" LABELS THE CALIBRATION SAMPLE

46: Do (P86)
1: 51    Set Port 1 Low

47: End (P95)
```

Taking input samples

The CR10(X) datalogger programming language is not very suitable for writing very complex programs implementing many stages of decimation. We decided to implement only one-stage decimation subroutine which is, again, common for standard and calibration data. In both cases, the number of input samples needed to compute one output sample is the same: 59. The input samples for a standard output sample are collected each 8 s, for a calibration output sample each second. The length of the standard input sampling interval is then 8 minutes, length of calibration input sampling interval is 1 minute. Both the input sampling intervals are centered around the respective output samples and the calibration output sample is exactly in the middle between two adjacent standard output samples (Fig. 5), therefore, the two kinds of input sampling intervals do not collide. If the calibration current is switched on immediately after the last standard input sample of the current interval is taken (at 4-th minute 0 s of the standard calibration period, as in the first program fragment above), then there is also enough time (30 s) for transients to vanish.

With the available instruction set, the regular way to find out whether the current minute is within the standard input sampling interval is to use *two* If instructions:

```
4: Do (P86)
1: 1    Call Subroutine 1 ; GIVES MINUTES MODULO SAMPLING PERIOD

5: If (X<=>F) (P89)
1: 13    X Loc [ modmin ]
2: 3 >=
3: 6    F ; BEGIN OF 8' INPUT SAMPLING INTERVAL
```

```
4: 30    Then Do

6: Z=X (P31)
1: 13    X Loc [ modmin ]
2: 17    Z Loc [ testmin ]

7: Do (P86)
1: 2    Call Subroutine 2 ; ADD NEW INPUT SAMPLE
8: Else (P94)

9: If (X<=>F) (P89)
1: 13    X Loc [ modmin ]
2: 4    <
3: 4    F ; END OF 8' INPUT SAMPLING INTERVAL
4: 30    Then Do

10: Z=X (P31)
1: 13    X Loc [ modmin ]
2: 17    Z Loc [ testmin ]

11: Do (P86)
1: 2    Call Subroutine 2 ; ADD NEW INPUT SAMPLE

12: End (P95)

13: End (P95)
```

We could, however, easily construct a *polynomial* which would give zero value only if the current minute is within the input sampling interval. Then only *one* If instruction 89 would be needed to make the decision. The problem is the maximum polynomial order of 5 allowed by instruction 55 Polynomial. This solution is then elegant only for short input sampling intervals and in any case, applicable only where the longer execution time of instruction 55 is not critical.

The subsequent calls of subroutine 2:

```
7: Beginning of Subroutine (P85)

1: 2    Subroutine 2

8: If Flag/Port (P91)
1: 41    Do if Port 1 is High
2: 30    Then Do

9: Z=X MOD F (P46)
1: 12    X Loc [ sec ] ; SECONDS
2: 1    F ; CALIBRATION INPUT SAMPLING PERIOD IN SECONDS
3: 14    Z Loc [ modsec ] ; TIME MODULO INPUT SAMPLING PERIOD
```

```
10: Else (P94)

11: Z=X MOD F (P46)
1: 12    X Loc [ sec ] ; SECONDS
2: 8    F ; NORMAL INPUT SAMPLING PERIOD IN SECONDS
3: 14    Z Loc [ modsec ] ; TIME MODULO INPUT SAMPLING PERIOD

12: End (P95)
```

. . .

generate altogether 60 samples, of which the first one has to be discarded. This can be done easily: the length of the "shift register" defined by the instruction 54 Block Move (contained in the subroutine 2) shall be 59 and the number of times this instruction is executed is 60.

Filtering

The filtering is implemented in two steps: first multiplication of the shift register members by filter coefficients with the help of instruction 53 Scaling Array and computation of Spatial Average (instruction 51). Attention, if the normalized coefficients (whose sum is equal to one) are entered, then the resulting spatial average shall be multiplied by the number of samples (swath) coming into it. Yet better solution is to enter the filter coefficients into instructions 53 already pre-multiplied by the swath, because this way, we increase the accuracy, which is very welcome in the outermost coefficients with very small values. In long filters, entering all the filter coefficients values into the instructions 53 is very tedious work demanding maximal concentration. This is one of the reasons why the same filter coefficients are used both for standard and calibration input samples.[5]

Output of sample

The output sample must be output with the right time, on the other hand, it can not be output before the last input sample needed for its computation arrives (Fig. 8 left) . Therefore, one can not take full advantage of the instruction 77 Real Time, which outputs the time at the moment it

[5] We computed the values of filter coefficients according to the equation (8) (0-th order fit with Kaiser weighting for $\beta = 8$). The shape of the impulse response and of amplitude frequency response for $L = 59$ are very similar to the ones for $L = 23$ in Fig. 9.

is executed. One must not forget to store the time corresponding to the output sample in a separate output location(s) and, at the moment of the output of sample, combine it with instruction 77:

```
48: Real Time (P77)
1: 100    Day (midnight = 0000)

49: Real Time (P77)
1: 1000    Year

50: Sample (P70)
1: 1    Reps
2: 10    Loc [ hour ] ; SAMPLES HOURS

51: Sample (P70)
1: 1    Reps
2: 18    Loc [ savemin ] ; SAMPLES MINUTES

52: Resolution (P78)
1: 1    High Resolution

53: Sample (P70)
1: 1    Reps
2: 19    Loc [ avextense ]

55: Serial Out (P96)
1: 71    SM192/SM716/CSM1

56: End (P95); OF SUBROUTINE 3
```

6. Conclusions

The main outcome of the work presented in this paper is the program controlling the data acquisition of standard and calibration extensometric data. We are aware that in the present form, it is best suited to meet the needs of those who visit their dataloggers to collect data manually only once upon a few weeks. We ourselves hope to leave this category of users soon. Nevertheless we felt that our experience is worth to be shared – to bring some benefit to the beginners in Campbell Scientific datalogger programming and possibly, to incite some feedback from the advanced users, of which we too can profit.

The copy of our *.csi program is freely available on demand.

Acknowledgments. We express our gratitude to the colleagues Dr. Gyula Mentes, Mrs. Ildikó Eperné Pápai, Mr. Tibor Molnár at the Geodetic and Geophysical Research Institute of the Hungarian Academy of Sciences for very inspirative environment during our visits of their institute in Sopron and for making us available their Campbell Scientific CR10 datalogger as well as other resources for development of datalogger programs.

We owe our thanks to Mr. Nigel Wills of Campbell Scientific, Ltd., UK, for his advices, which pointed us to the right direction.

We highly appreciate the care and technical skills of our volunteer co-worker Mr. Pavel Hudec who keeps our tidal station alive and kicking.

This work was supported by Science and Technology Assistance Agency under the contract No. APVT-51-002804 and by VEGA Grant Agency under Projects No. 2/3057/23 and 2/3004/23.

References

CR10X Measurement and Control Module – Instruction Manual. Campbell Scientific, Inc., 2000.

Dudášová V., 1998: Description of the renovated extensometer at the Vyhne tidal station. Contr. Geophys. Geod., **28**, 197–203.

Mentes Gy., 1993: Short remarks concerning the calibration of quartz tube extensometers. Marées Terrestres Bulletin d'Information, **115**, 8467–8471.

Ondráček O., 2002: Discrete signals and systems. Slovak Technical University, Bratislava, 250 p. (in Slovak).

Sharma S. R., Saxena R., Saxena S. C., 2004: Design of FIR Filters Using Variable Window Families: A Comparative Study, J. Indian Inst. Sci., **84**, (Sept.-Oct. 2004), 155–161.

Vích R., Smékal Z., 2000: Digital filters. Academia, Prague, 218 p. (in Czech).