ISSN 1453 – 7303 **"HIDRAULICA" (No. 4/2015)**
Magazine *of Hydraulics, Pneumatics, Tribology, Ecology, Sensorics, Mechatronics*

Multiplexed Delay Compensation and Circular Buffer Method for moving Average Filtering of Signal Acquired from Tactile Sensors in a Mechatronics System for Walking Analysis

PhDc. Eng. **Anghel CONSTANTIN[1],** Prof. Dr. Ing. **Constantine DAVID[2]**

[1] Doctoral School of Valahia University Targoviste, 130105, Romania, e-mail: anghel.constantin@incdmtm.ro

[2]Technological Education Institute of Central Macedonia -Serres, Greece, e-mail: david@teiser.gr

Abstract: *Traditional filtering performs very well when the frequency content of signal and noise do not overlap. When the noise bandwidth is completely separated from the signal bandwidth, the noise can be decreased easily by means of a linear filter. On the other hand, when the signal and noise bandwidth overlap and the noise amplitude is enough to seriously corrupt the signal, a traditional filter, designed to cancel the noise, besides might introduce signal cancellation or, at least, would result in signal distortion. The paper presents a multiplexed delay compensation and circular buffer method for moving average filtering of signals. The application refers to the measurement of the ground reaction force during walking using 20 tactile-sensors in order to calculate the caloric energy consumption of the human body. The proposed method is an improved version of a previous one, where a low-speed microprocessor was used and there was not enough time to complete the required calculations and corrections. The current results are very promising, as the error correction due to delay of the samples between sensors by applying the moving average filtering method and using as well circular buffer is minimized.*

Keywords*: DSP, ring buffer, circular buffer, tactile sensors*

1. Introduction

Advanced technology and new generation industrial applications, both matured and under development, contribute substantially to the design, development, validation and integration of Intelligent Mechatronics Systems [1]. However in many mechatronics applications there is a major problem how to prepare and process sufficiently the signals acquired from the sensors. This research work aims at the calculation of the caloric energy that is consumed by a human body during walking. To achieve that the ground reaction forces (GRF) acting on the foot sole at walking are experimentally assessed. The signals we are looking for are acquired from appropriate touch sensors applied on the sole. The force data are used for further analysis, whereby interesting technical aspects and restrictions have to be considered as following:

- The large number of sensors (10 on each foot) demands sampling and multiplexing digital analog conversion.
- The high impedance sensors cause noise and there is need to signal filtering, etc.

Figure 1 illustrates graphically the signal acquisition set-up. The signals recorded from the sensors after proper conditioning are acquired by a microcontroller supported device.
The analog multiplexer allows the use of a single analog digital converter with a 10 bits resolution, which is included in the microcontroller, for the 20 analog input channels. The closing and opening command for the analog switchers is performed by the means of 5 selection inputs commanded by the microcontroller. This sampling process within the time domain creates a delay of the observation moments for each channel, which is proportional to the number of that channel multiplied by the time needed for the conversion of each channel. This phenomenon introduces errors within the calculation needed by the processing in order to find out the powers developed by different compounds of the reaction force and the total caloric energy which is consumed.
The maximum error shows up at the maximum frequency and at the sensor No. 20 as shown hereinafter.

$$20F_{Nyquist} = 20 \cdot 200Hz = 4Khz \qquad (1)$$

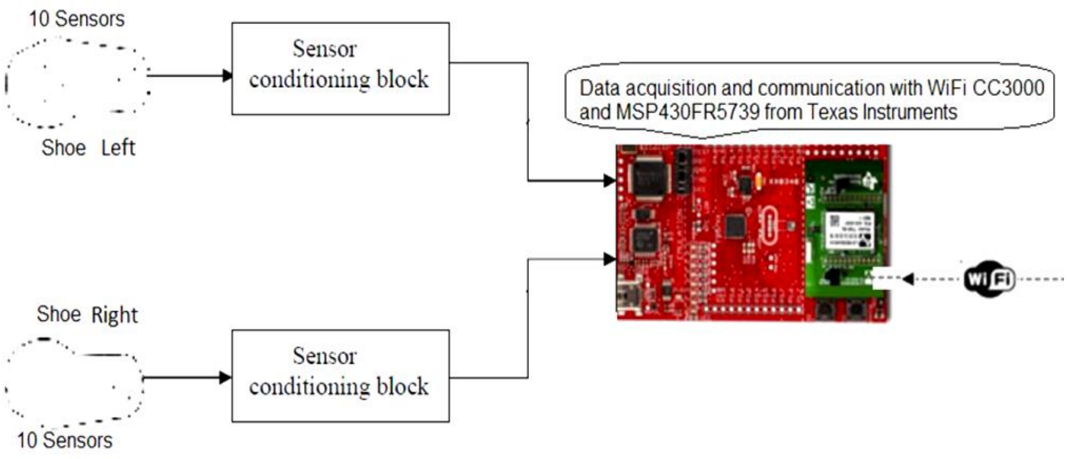
10 Sensors
Shoe Left
Sensor conditioning block
Data acquisition and communication with WiFi CC3000 and MSP430FR5739 from Texas Instruments
Shoe Right
Sensor conditioning block
10 Sensors
WiFi

Fig. 1. Block diagram of signal acquisition

In order to eliminate this type of error it was chosen the determination of an interpolation formula which should accomplish the samples temporal alignment through the determination of their estimated value in accordance with the number k of the selected sensor. During simplified conditions around a, t point, the Taylor formula is available.
In digital signal processing (DSP), the continuous time variable, t, is replaced by the discrete integer variable, n, with sampling units. More precisely, the time variable, in seconds, has been normalized (divided) by the sampling interval, T (seconds/sample), which causes time to have convenient integer values at the moments of sampling.
In such conditions, by considering T=5ms, the sampling period for a sensor, which corresponds to performing the necessary calculations, the following formula results [2]:

$$\widehat{S_k}[n] = S_k[n-1] + \left(1 + \frac{(k-1)\Delta\tau}{T}\right)(S_k[n] - S_k[n-1]);\;\; k = \overline{1..20} \qquad (2)$$

Using the following MATLAB code (Table 1) the error due to the mentioned delay was simulated and the correctness of the estimated value (without delay) using the formula (2) is checked:

TABLE 1: MatLab code

```
clc
f=100;
omega = 2*pi*f;
T=0.005;
tq =0.000016;
tau=0.000008;
factor =1-19*tq/T;
t=0:tau:0.020;
subplot(4,1,1), plot(t*1000,sin(omega*t),'LineWidth',2,'Color',[0 0 1]),grid, xlabel('Time [ms]'),title('Sensor1')
subplot(4,1,2), plot(t*1000,sin(omega*(t+20*tq)),'LineWidth',2,'Color',[0 0.5 0]),grid, xlabel('Time [ms]'),title('Sensor20')
text(0,sin(omega*20*tq),' \leftarrow error','FontSize',12,'Color',[1 0 0])
subplot(4,1,3), plot(t*1000,sin(omega*(t-T+20*tq))+factor*(sin(omega*t)-sin(omega*(t-T+20*tq))),'LineWidth',2,'Color',[0 0.5 0]),grid, xlabel('Time (ms)'),title('Sensor20 after correction')
f=0:1:100;
omega = 2*pi*f;
subplot(4,1,4), plot(f,sin(omega*20*tq)-sin(omega),'LineWidth',1,'Color',[1 0 0]),grid, xlabel('f [Hz]'),title('ERROR')
stem(t,sin(omega*t));
Fs = 100; % Sampling rate
y = [0:2*Fs+1]'/Fs;
```

In Figure 2 the ground reaction force measurement from the sensor No1 and sensor No20 is shown, as well as the dependency of the frequency error.

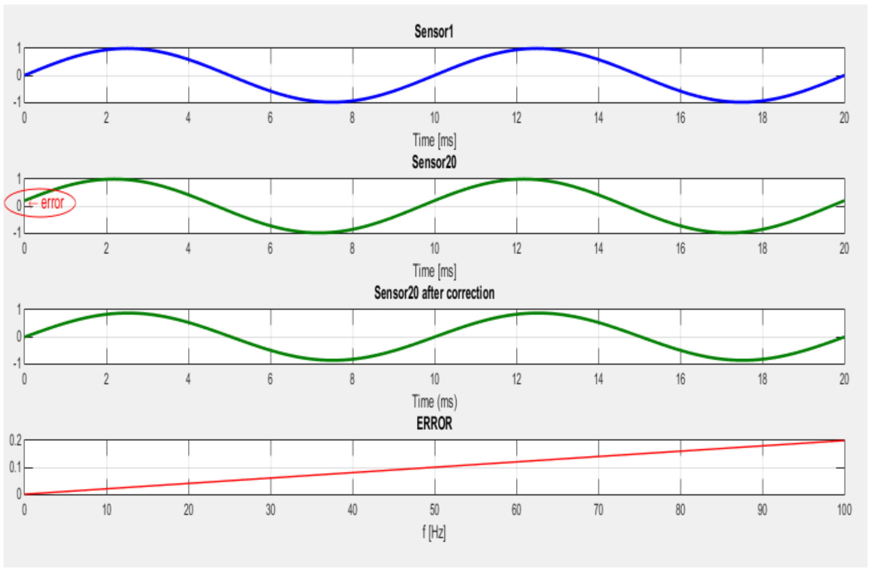

Fig. 2. Signal from two sensors and the resulted frequency error

2. Filter implementation and methods

In practice the difference between simulation and real signal to be processed is the action of noise. Because the input amplifier of a signal conditioning system has high impedance the main component of the noise has 50Hz frequency due to the power line capacitive common mode coupled with other power lines belong to the environment. In order to minimize this influence a conditioning amplifier was designed in differential topology with antialiasing filter and a moving average digital filter using a circular buffer on the basis of a signal processing software was developed.
Signal averaging is a signal processing technique applied in the time domain, intends to increase the strength of a signal relative to noise [3]. By averaging a set of replicate measurements, the signal-to-noise ratio, S/N, will be increased, ideally in proportion to the square root of the number of measurements [4].
If, x[n] is the input signal, y[n] is the output samples of the signal, M is the number of points used in the moving average window then the equation using only points on one side of the output sample is calculated as following:

$$y[n] = \frac{1}{M}\sum_{i=0}^{M-1} x[i+n] \tag{3}$$

For example, in an eight points moving average filter, ten points in the output signal is given by:

$$y[10] = \frac{x[10]+x[11]+x[12]+x[13]+x[14]+x[15]+x[16]+x[17]}{8} \tag{4}$$

2.1 Recursive implementation

Based on equation (3) two adjacent output points [n] and y[n+1], can be calculated using the recursive fast calculus. After the first point is calculated in y[1], all other points can be found with only a single addition and subtraction per point. This can be expressed as the equation (5) declares:

$$y[i] = y[i-1] + x[i+p] - x[i-q] \tag{5}$$

Where:

$$p = \frac{M-1}{2} \text{ and } q = p + 1 \quad (6)$$

Before using the equation (3), the first point in the signal must be calculated using a standard summation.

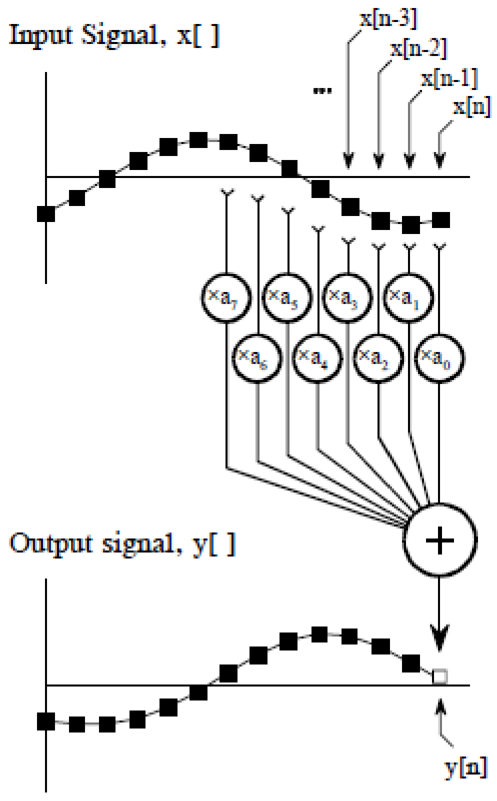

Fig. 3. Standard summation scheme of the signal

In Figure 3, the average within a sampling region [4] is illustrated, rather than just being equal to the signal value at the sampling instant. A coherent averaging procedure can be perceived as a digital filtering process, and the frequency characteristics can be also investigated. In expression (7) through the z- transform, the transfer function of the filtering operation results in:

$$H(z) = \frac{1+z^{-n}+z^{-2n}+\cdots+z^{-(N-1)n}}{N} \quad (7)$$

Where N is the number of elements in the average, and n is the number of samples in each response. This is a moving average low-pass filter as discussed earlier, where the output is a function of the preceding value with N samples. In practice, the filter does not operate on the time sequence but in the sweep sequence on corresponding samples. Basically the moving average filter is a *convolution* of the input signal with a *rectangular pulse* having an area of *one*.
By the aid of a MatLab application the frequency response of the filter as shown in Fig. 4 can be calculated.

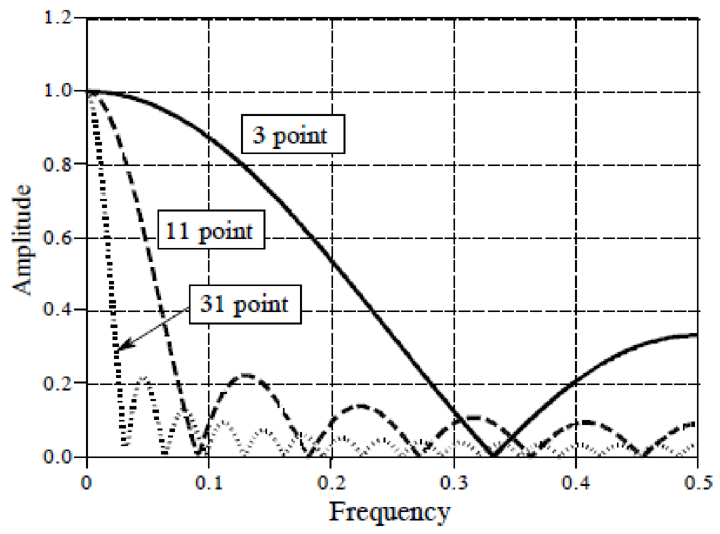

Fig. 4. Frequency response of the filter

The moving average filtering method proved to be a very good solution for many applications, it is *optimal* for reducing the noise while keeping the sharpest step response. The roll-off is very slow and the stop band attenuation is very bluff but the moving average filtering cannot separate one band of frequencies from another band [5].
In conclusion the moving average filtering has good performance in the time domain and poor performance in the frequency domain, and vice versa. In other words the moving average is an exceptionally good *smoothing filter* (the action in the time domain), but a bad *low-pass filter* (the action in the frequency domain).

2.2 Circular buffer implementation

In off-line processing, the *entire* input signal resides in the memory at the same time. This means for example, a doctor has the possibility to record the ground reaction force during human walking and analyze that afterward. Another example of off-line processing is medical imaging, such as computed tomography and MRI. This method supposes the existence of huge data recording and big memory support.
In real-time processing, the result of output signal is produced at the same time that the input signal is being acquired. For example, this is needed in voice communication, radar or other applications where there is an indispensable demand to have the information immediately available, although it can be delayed by a short amount. For instance, a 10 millisecond delay in a telephone call cannot be detected by the speaker or listener. Real-time applications input a sample, perform the algorithm of calculus, and output a sample, over-and-over [5].
To calculate the output sample, we must have access to a certain number of the most recent samples from the input. For example, we suppose eight coefficients in this filter.
This means, it is required to know the value a_1,..,a_7 of the eight most recent samples from the input signal, $x[n]$, $x[n+1]$,…,$x[n+7]$. These eight samples must be stored in memory and continually updated as new samples are acquired. The most satisfactory solution for memory managing is to use a circular buffering presented as principle in the Figure 5.

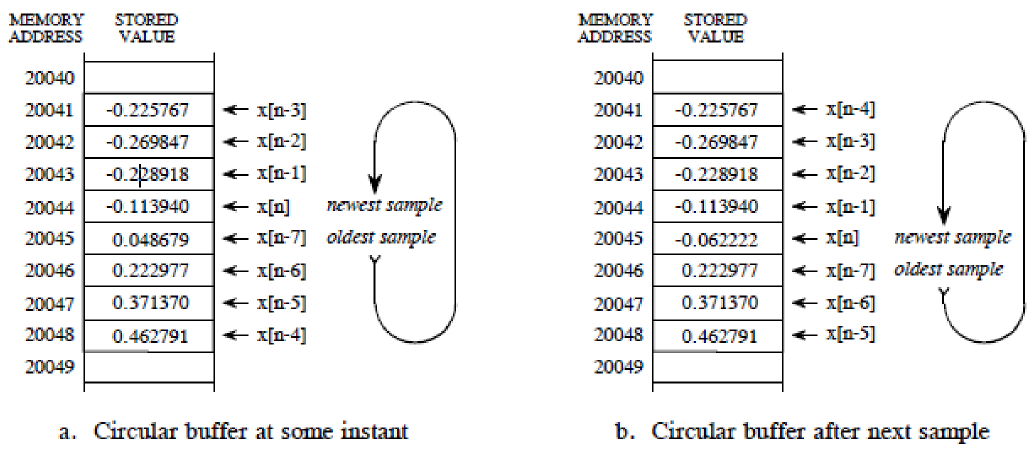

Fig. 5. Circular buffering principle scheme

In table 2 below a short listing of the C code used in the application is demonstrated. The code aims at the real-time filtering of the digital data acquired from the ground reaction force, which is recorded by means of 20 tactile sensors during walking.

TABLE 2: C code listing

```
#define BUFFER_SIZE 16

typedefstructcircular_buffer
{
int buffer[BUFFER_SIZE];
volatileunsignedint head;
volatileunsignedintrear;
}ring_buffer;

ring_buffermy_buf = { {0}, 0, 0 };
/*********************************************************************************/
voidstore_in_buffer(int data)
{
unsignedint next = (unsignedint)(my_buf.head + 1) % BUFFER_SIZE;
if (next != my_buf.rear)
  {
my_buf.buffer[my_buf.head] = data;
my_buf.head = next;
  }
/*********************************************************************************/

for (index_filtru=0;index_filtru<BUFFER_SIZE;index_filtru++)
{
          if (my_buf.head == my_buf.rear) {
            }
          else {
          data = my_buf.buffer[my_buf.rear];
          my_buf.tail = (unsignedint)(my_buf.rear + 1) % BUFFER_SIZE;
            }
   dADC1 =(double)data, dSum=dSum+dADC1;
}
```

4. Conclusions

The moving average method is the most common filtering technique in DSP, mainly because it is the simplest digital filter to understand and to use and it is optimum for a common task because of reducing random noise, while at the same time retaining a sharp step response [6]. This makes it the premier filter for time domain encoded signals. However, the moving average is the worst filter for frequency domain encoded signals because it does not have the ability to separate one band of frequencies from another.
Simultaneous Sampling eliminates time skew between channels, simplifies however both time and frequency based analysis techniques. Multiplexed Sampling channels are sampled sequentially requiring also software correction for detecting certain patterns.

Acknowledgements

This paper has been prepared with the financial support of the project “Quality European Doctorate - EURODOC”, Contract no. POSDRU/187/1.5/S/155450, project co-financed by the European Social Fund through the Sectoral Operational Programme “Human Resources Development” 2007-2013.

References

[1] Gh. I. Gheorghe, “Adaptronica Sistemelor Inteligente”, AGIR Publishing House, Bucharest, Romania, 2014, ISBN: 978-973-720-509-4, p.15, 2014;
[2] S. Pasca, R. Strungaru, A. Constantin, G. Capris, “New Data Acquisition and Processing Methods for Resistive Sensors in Pedometric Measurement Systems”, Nonlinear Optics, Quantum Optics: Concepts in Modern Optics; 2009, NLOQO 39.2-3, pp. 117-136;
[3] Ed. Joseph D. Bronzino, “The Biomedical Engineering Handbook, Second Edition”, CRC Press LLC, ISBN 0-8493-0461-X, pp.238, 988 2000;
[4] S. D. Stearns and D. R. Hush, “Digital Signal Analysis”, 2nd Ed., Englewood Cliffs, NJ: Prentice Hall, 1990;
[5] Texas Instruments, Inc., TMS320C55x Chip Support Library API Reference Guide, Literature no. SPRU433J, Sep. 2004;
[6] R. Rădoi, I. Duțu, “Virtual instrument for plotting servo-valves characteristics after significant maintenance operations”, In: Hidraulica no. 1/ 2013, pp. 86-89.