

Pilot Signal Removal, Digital Signal Processing Algorithm and it's Practical Implementation

Tibor Wühl

Óbuda University, Kandó Kálmán Faculty of Electrical Engineering
Bécsi út 96/b, H-1034 Budapest, Hungary
wuhrl.tibor@kvk.uni-obuda.hu; ORCID: <https://orcid.org/0000-0002-7522-3511>

Abstract: In this paper, an algorithmic development, using an efficient wave, digital signal processing method, is presented. The developed algorithm can also be used in practice. The algorithm is obtained by transforming a passive reference network, the passive reference network is described by voltage waves. After the theoretical summary, the steps of design and design review are presented through a concrete practical example. The algorithm is guaranteed to be stable and limit-cycle free.

Keywords: UIC pilot signal; DSP algorithm; Digital Signal Processing; Wave Digital Filtering

1 Introduction

The International Union of Railways (UIC) defines the specifications for a number of railway communication systems and constituents.

The UIC codex 751-3, 4th edition 2005 [1] gives the "Technical specifications for analogue train radio systems in international service". The specifications of analog systems cannot be called new, but due to compatibility requirements, old system specifications must also be taken into account when designing new system components. The new system components are preferably made with digital solutions at the technical level of our time and thus also in the case of analog systems. Circuits implemented in digital circuits operating in analog environments take a sample from the analog signal then quantize and encode them, i.e., digitize them. Output samples are calculated from the digital samples according to the signal processing algorithm in real time [2]. The DSP algorithm defines the calculations and their order. The creation of DSP algorithms is a mathematical and electrical engineering task, during which several mathematical transformations are required.

In every step of the design of the signal flow diagram, we have to take care of the stability, i.e., that the developed algorithm does not excite and is free of limit cycles [3]. This can be guaranteed by maintaining passivity throughout [4]. In the case of wave-digital filter design, our reference circuit is composed of passive components (resistor, capacitor, inductance, and lossless interconnect components) [3], but the stability conditions can only be met if the implementation is not included any faults. We use a linear model throughout the design of the signal flow diagram, however, all of the components in the model are lossless. In the implementation of the algorithm (FPGA, or signal processing processor design), the operating environment is not linear due to the finite number of bits. I treat the granular nonlinearity resulting from quantization as well as the overflow separately. MATLAB® simulation is used to handle problems and detect individual faults [5].

1.1 Transformation System, Design Steps of DSP Algorithm

The DSP algorithm is built on different planes. Transformations give connections between each plane. The mathematical relationships between the planes are important elements of the design steps. All of this is summarized in the following figure:

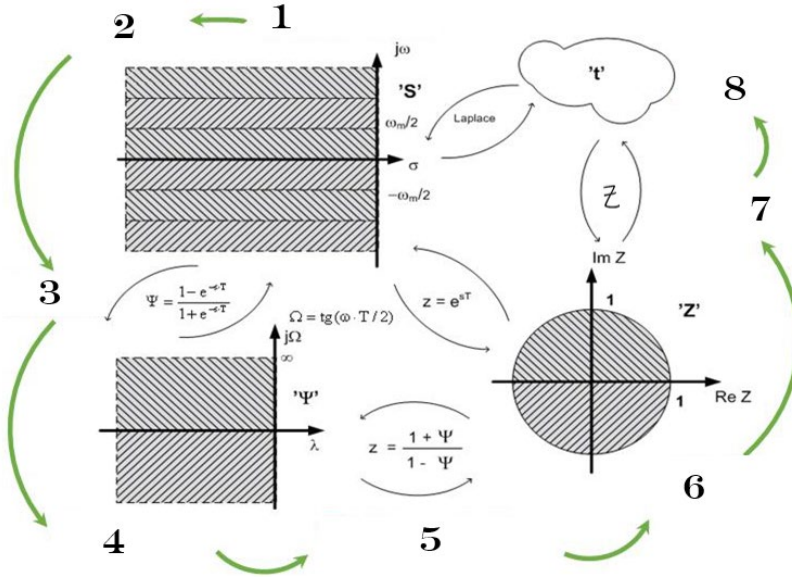


Figure 1
System of Domains and Transformations

The step marked '1' in the figure above is to specify the requirement on the complex frequency plane. In the quasi-stationary case, the real part of the complex frequency is zero, so the requirement is defined with respect to $j\omega$, as an independent variable.

Figure 2 shows the requirement for a low pass filter. The requirement for high-pass, band-stop, and band-pass filters can be converted to a low-pass requirement after so-called type transformation [6], that will not detailed in this article.

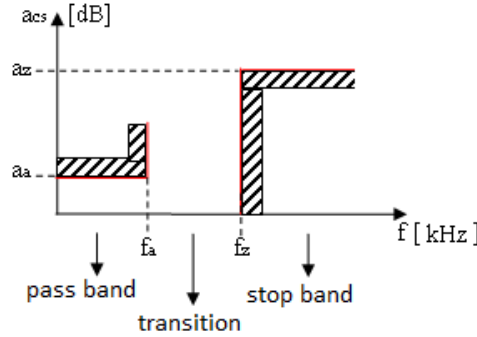


Figure 2

Low-pass filter criteria in frequency domain

In the next step, we select a sampling frequency for the digital signal processing system that matches the spectrum of the signal to be processed. The value of the sampling frequency is given in the Sannon's theorem:

$$fm \geq 2 \cdot f \max \quad (1)$$

As the result of sampling, the 's' complex frequency plane and its $j\omega$ axis become periodic. The alias components are displayed. This means that the requirement in Figure 2 also becomes periodic in the $+\infty$ and $-\infty$ range. The band periodicity makes it impossible to use well-proven mathematical methods in the design of circuits and algorithms. Step 3, of Figure 1, eliminates periodicity with the following relationship:

$$\Psi = \frac{1 - e^{-sT}}{1 + e^{-sT}} \quad (2)$$

Equation (2) makes the plane Ψ free of physical dimensions. Several DSP literatures [7] use a different transformation and define the plane Ψ denoted as a distorted but physically dimensional plane. Exemption from the physical dimension has a number of advantages that are an advantage in the design steps.

The Ψ plane is of complex frequency nature, the real part of that is denoted by λ , while the complex component is denoted by Ω , so the independent complex variable is as follows:

$$\Psi = \lambda + j\Omega \quad (3)$$

Important features of the transformation (2) are that it transforms the 's' plane origin into the ' Ψ ' plane origin, as well as an axis-bearing transformation, i.e., moves the $j\omega$ axis to the $j\Omega$. For $\sigma = 0$, transformation (2) will have the following form (4):

$$\Omega = tg(\omega \cdot \frac{T}{2}) \quad (4)$$

Of course, as we have established the physical lack of dimensions in the plane Ψ , this is now also true for the Ω axis. Transformation (4) eliminates the periodicity that became periodic after sampling (Figure 3) (3rd step of Figure 1):

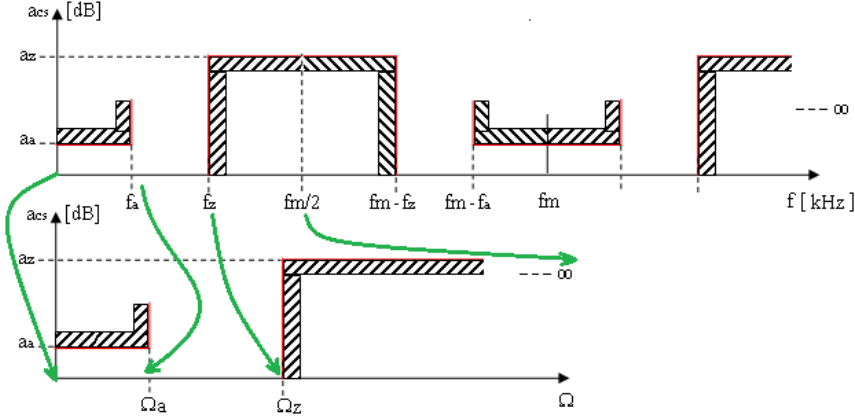


Figure 3
Elimination the periodicity of requirement

Figure 3 shows that half of the sampling frequency is transformed from the 's' plane, to the $+\infty$ point in plane Ψ .

The fourth step in Figure 1 is designing an analog reference filter in the Ψ domain. In this domain we can use all well-known analog passive filter design methods. These are not intended to be presented in this article.

The fifth step in Figure 1 is more complex.

Before beginning the transformation, it is advisable to redraw the analog reference circuit. The following figure shows an example of how to connect the individual components per gate.

Based on Figure 4, the individual building elements were highlighted as passive components, as well as the passive network components that connect them. Passive connections can be parallel or serial.

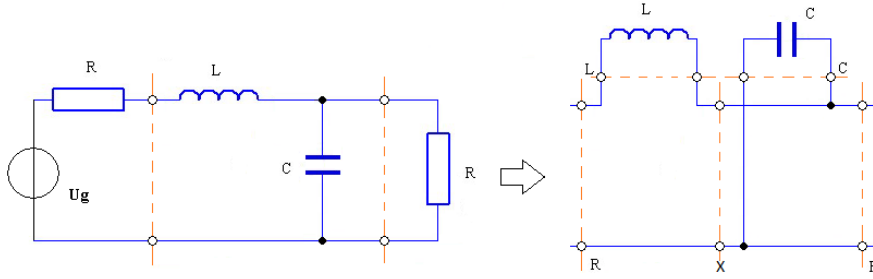


Figure 4

Redrawing the reference filter in “ Ψ ” domain (an example)

The transformation of the components related to the gate coupling [4], is discussed in the next subsection due to its significance.

1.2 Transformation of Components with Voltage Waves

As described in the previous section, the reference circuit is composed of passive components only. By separating these passive components from each other and coupling them as gates, we can obtain separately transformable elements. Our goal is to transfer these components to the “ z ” plane. During the transformation, care will be taken to maintain passivity. Each building block is described by voltage waves (5).

$$\begin{aligned} a(t) &= u(t) + i(t) \cdot R \\ b(t) &= u(t) - i(t) \cdot R \end{aligned} \quad (5)$$

In the context of (5) above, $a(t)$ is the voltage wave passing through the gate, while $b(t)$ is the reflected wave. “ R ” is the gate resistance parameter specific to the gate. The “ Ψ ” plane does not contain physical dimensions, so the gate resistance parameters are just mere numbers in Figure 4 marked “ R ”, “ L ”, “ C ” and “ X ”. The traveling and reflected waves can, of course, be interpreted in all the planes defined in Figure 1, for example in the “ Ψ ” plane (6):

$$\begin{aligned} A(\Psi) &= U(\Psi) + I(\Psi) \cdot R \\ B(\Psi) &= U(\Psi) - I(\Psi) \cdot R \end{aligned} \quad (6)$$

Thus, we interpreted propagating and reflected waves between the gates, and assigned a gate resistance to the gates.

In describing the transformations, I have shown that the plane Ψ is a plane free of physical dimensions. We will also take advantage of this, since the gate resistance R has only a numerical value. The same is true for capacitor “ C ” and inductance “ L ”.

Let us examine how the capacitor “C” defined in the plane Ψ can be transformed into the plane “z”:

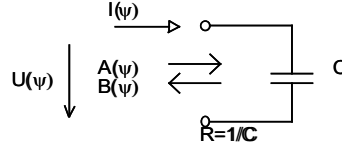


Figure 5
“C” components on “ Ψ ” domain

The relationship between current and voltage on the capacitor can be determined by the following equation:

$$U(\psi) = I(\psi) \cdot \frac{1}{\psi \cdot C} \quad (7)$$

The voltage and current can be described by the voltage propagation and reflected waves in the plane Ψ , which are derived from (6):

$$\begin{aligned} U(\psi) &= \frac{A(\psi) + B(\psi)}{2} \\ I(\psi) &= \frac{A(\psi) - B(\psi)}{2 \cdot R} \end{aligned} \quad (8)$$

Substituting the current and voltage values expressed by voltage waves (8) into the Equation (7), we get:

$$\frac{A(\psi) + B(\psi)}{2} = \frac{A(\psi) - B(\psi)}{2 \cdot R} \cdot \frac{1}{\psi \cdot C} \quad (9)$$

After choosing the gate resistance condition $R = 1 / C$, our relation (9) will have the following form:

$$B(\psi) = A(\psi) \cdot \frac{1 - \psi}{1 + \psi} \quad (10)$$

Applying the relation marked in step 5 at (Figure 1) of our transformation system we get the following:

$$B(z) = A(z) \cdot z^{-1} \quad (11)$$

Equation (11) gives the building element transformed to the “z” plane (Figure 6). Here we must not forget that this result was obtained in connection with the choice of the gate resistor $R = 1 / C$.

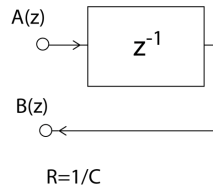


Figure 6

Transforming the “C” Components to “z” domain with voltage waves

Let us examine how the inductance “L” defined in the plane Ψ can be transformed into the plane “z”. With a similar line of reasoning and choosing the gate resistor $R = L$, the result will be:

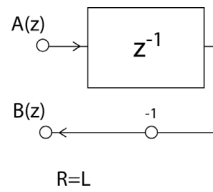


Figure 7

Transforming the “L” Components to “z” domain with voltage waves under condition $R=L$

The parallel connecting network part can be described using the Kirchhoff's laws. In the simplest case, two components are connected in parallel, but of course the connection can also be done by connecting a “n” gate. Connecting network components are called adaptors [4] [8].

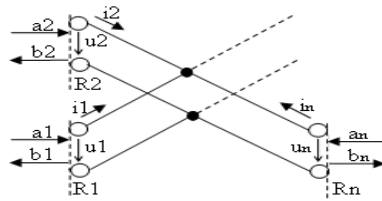


Figure 8

Parallel connection model

Excluding derivations, the general wave equation for the parallel connection is as follows [3]:

$$b_v = (\gamma_1 \cdot a_1 + \gamma_2 \cdot a_2 + \dots + \gamma_n \cdot a_n) - a_v \quad (12)$$

In the context of (12), G is the gate conduction parameter which is the reciprocal of the gate resistance R . In the context, “n” is the total number of gates, and the index “v” is the number of the gate to be characterized.

With the gate guidance parameter, the reflection factor of a given gate is given by the following relation [3]:

$$\gamma_v = \frac{2 \cdot G_v}{\sum_{k=1}^n G_k} \quad (13)$$

The initial figure for the serial connection is illustrated in the following figure:

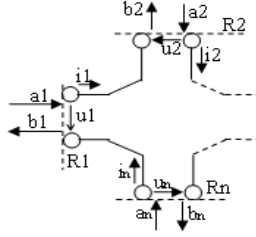


Figure 9

Serial connection model

Excluding derivations, the general reflected wave of the series connection at the v^{th} gate is as follows [3]:

$$b_v = a_v - \gamma_v \cdot (a_1 + a_2 + \dots + a_n) \quad (14)$$

The reflection factor of equation (14) with the gate resistance parameters R will be as follows [3]:

$$\gamma_v = \frac{2 \cdot R_v}{\sum_{k=1}^n R_k} \quad (15)$$

It can be seen from (12) (13) and (14) (15) that the adaptor components are memory - free, the interconnection of the components is loss - free, and can be considered passive under linear operating conditions.

1.3 Simulation Method before Realization of Algorithm

A linear model of the DSP algorithm was generated in the “z” plane. This is well suited for thinking in the frequency range.

The operation of the DSP algorithm, i.e., the input pattern at the sampling intervals (T) and the state variables storing the previous effects, determine the output pattern of the given phase. The signal flow diagram to be implemented will thus be a task scheduled at “ T ” intervals. This matches the DSP flowchart drawn

in the time range. For the above reasons, the simulation and the effect of nonlinearities are investigated in the time domain. Observation and model creation in the time domain are used in many fields also [15]. To do this, the DSP algorithm generated in the “z” plane should be transformed into the time domain (see Fig. 1, step 7).

The tests are performed in the time domain with MATLAB® simulation, the advantage of which is that it can give results close to the actual implementation. The purpose of the primary results is to support the fulfillment of the filter requirement. Then, within the limits of the number representation accuracy provided by MATLAB® [9], The simulation environment and the obtained result are considered as a study under linear conditions. The purpose of this simulation is to shed light on possible design errors. The Kronecker delta signal as the input, excitation signal was used. This signal assumes a value of “1” at the time of first sampling, so that the spectrum of the signal is “1” for all possible spectral components, although for a period of “dt”. Of course, the examined filter modifies the amplitude and phase position of the spectral components of the input signal according to its transmission function. The applied simulations are even used in the time domain analysis of mechanical systems too [17].

In the time domain, the response to the Kronecker delta signal will be the weight function, by whose spectral analysis we can determine the transfer function. For the simulation result, was expect that the transfer function fulfills the expectation for all parameters of the filter requirement given for the initial condition (see in Figure 2). If this was not the case, then we made a mistake in one of the points 1 to 7 of the design (see in Figure 1), so it is necessary to check and correct the individual steps.

After the successful simulation in the environment considered as linear conditions, was consider it necessary to investigate the nonlinear effects caused by the finite number representation. This also requires running the MATLAB® time domain simulation [5]. Two types of tests should be performed with a test signal that matches the typical conditions of use of the algorithm.

The first aspect of the study is the overflow analysis. Here, it must be determined whether there are points in the implementation (FPGA or microcontroller) that the algorithm contains overflow-prone nodes for the available number representation (case of given bit number and representation mode). Based on the results of the discovery of such nodes, the control of these points can be reduced by so-called scaling [10]. During scaling, the risk of overflow can be eliminated or reduced by replacing some network component with a linear equivalent [10]. This intervention ultimately improves the dynamic range of the algorithm.

The second nonlinearity test assumes a more accurate knowledge of the actual algorithm implementation. Now we need to know exactly the quantization points that are forced to be used in the algorithm during implementation. These typically occur at the output of multipliers, since in the case of a fixed-point representation,

the result swells to 2 times "n" bits by multiplying the "n-bits" represented numbers with each other. Bit count truncation (rounding, truncation) occurs as a noise generating generator. In addition to signal-to-noise degradation, excess signal energy appears in the algorithm, compromising the passivity criterion.

2 Pilot Signal Remover Filter

After the theoretical introduction, the design and feasibility are presented through a specific task. In formulating the requirement, the goal is to remove a pilot carrier signal mixed to an audio frequency signal [1]. A band-stop filter with a sufficient slope is suitable for this. The stop band of the band should be as narrow as possible in terms of the speech signal, but the unpleasant effect of the components resulting from the frequency inaccuracy of the pilot signal and possible switch-off phenomena is contrary to this requirement. It is important to maintain a "balance" in the requirement. The LWDF structure was chosen to solve this problem. [10].

2.1 Structure of Lattice Wave Digital Filter []

The LWDF filter consists of two branches, a lower branch and an upper branch. A Figure 10, in the illustrated structure, S' and S'' are called the "all-pass components".

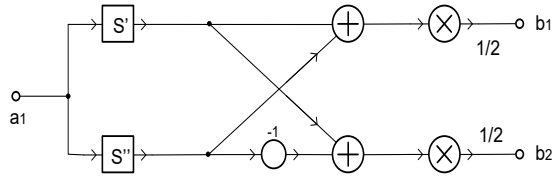


Figure 10
Applied LWDF structure

The absolute value of the transmission function of the all-pass component is a constant one regardless of the complex frequency. This means that these terms do not change their gain as a function of frequency, but do phase shift only. At the frequencies at which the phase shifts of S' and S'' are the same and the two signals are summed, a passband is obtained. At frequencies where the phase difference between the phase rotations of the two all-pass parts outputs is 180° , a closing band is obtained due to signal quenching. Output "b2" gives the dual pair of the result of "b1".

For example, a first-order all-pass function can be formed with a circulator and a capacitive element:

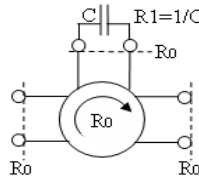


Figure 11

First order all-pass component at “Ψ” domain with circulator and capacitor

In practice, an all-pass component can be made by connecting a two-gate adaptor and a reactant component. These components were described in the previous chapter.

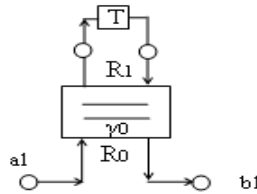


Figure 12

First order all-pass component at “time” domain with two port adaptors

The input of the first-order all-pass section is now the input of the forward wave (a1) of gate number 1, while the output is the reflected wave point (b1) of the same gate. The γ_0 multiplication factor of adaptor is now not calculated from the indicated gate resistances, but is defined by the following relation (12):

$$\gamma_0 = \frac{1 - B_0}{1 + B_0} \quad (12)$$

Figure 13 illustrates the n^{th} degree LWDF structure built from all-pass components.

2.2 Pilot Remover LWDF

According to UIC Code 751-3, 4th edition of 2005 [1], the audio frequency speech signal broadcast on the radio sits on a 2800 Hz harmonic pilot carrier. The pilot carrier is in the audible range during reception, so it is absolutely necessary to remove it, in order to understand the transmitted speech.

Samples are available from the analog signal at intervals of 125 μs , i.e., the sampling frequency is 8000 Hz.

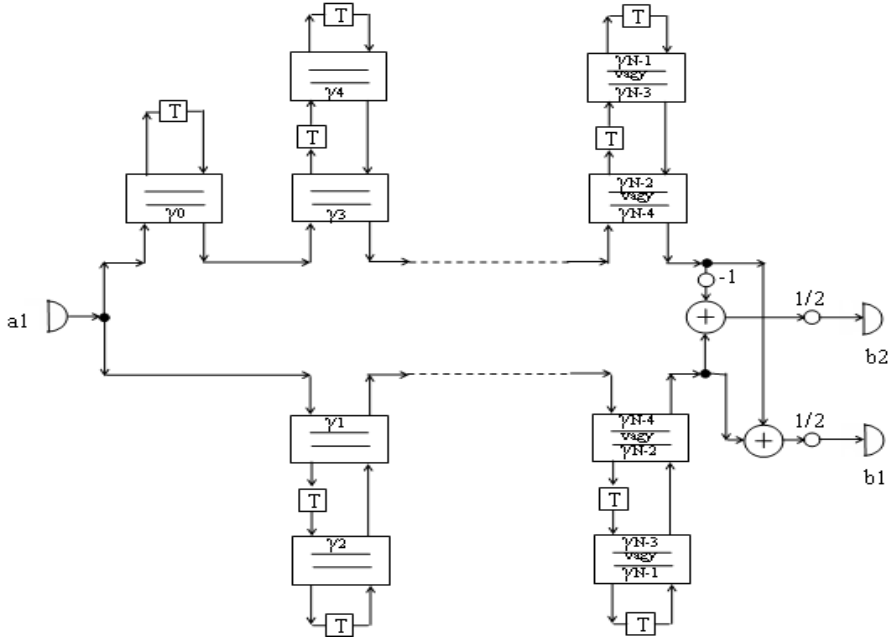


Figure 13
nth order LWDF structure

At the frequency of 2800 Hz, the minimum requirement is 30 dB attenuation, therefore a starting demand of 40 dB is assumed for the approximation (This means an attenuation margin of 10 dB.). The approximation tasks are solved with the MATLAB® program. Using the MATLAB toolbox [11], the specified parameters are as follows without explaining the execution steps:

- $f_s=8000$ Hz
- Center frequency = 0.35 (relative from f_s)
- Bandwidth = 0.09 (relative from f_s)
- Filter type: band stop

The approximation method is Butterworth.

The reflection factors of the individual adapters are given the following numerical values:

$$\gamma_1 = -0.88701$$

$$\gamma_2 = -0.58779 \quad (13)$$

The above filter coefficients can be directly applied to the all-pass components shown in Figure 12 and Figure 14.

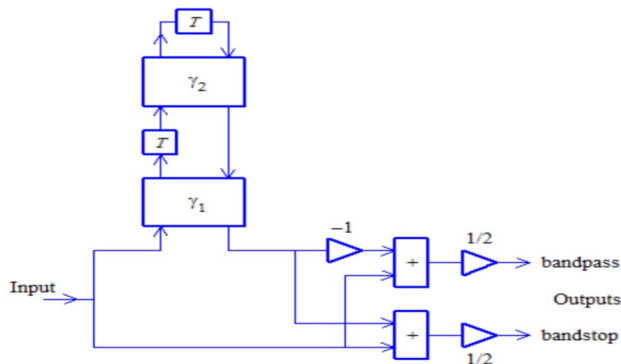


Figure 14

UIC pilot signal remover LWDF structure, designed with MATLAB® [11]

Under linear conditions, this would give a satisfactory result, but due to the quantization and overflow problems caused by the finite number of bits, the scaling of the structure [10] becomes necessary.

2.3 Scaling of the Pilot Remover LWDF

When scaling WDFs, we use the linear equivalents of each building block. In linear conditions, this has no effect. In a finite-bit implementation environment, the linear equivalents [8], on the other hand, behave differently.

High-frequency theoretical foundations [14] are often used in LWDF design.

After scaling, the filter coefficients were changed (14):

$$\begin{aligned} g1 &= 1 + \gamma_1 = 0.11299 \\ g2 &= 1 + \gamma_2 = 0.41221 \end{aligned} \quad (14)$$

After scaling the filter, we should check that it is working properly.

The detailed signal flow diagram of the scaled DSP algorithm is illustrated in Figure 15.

For verification, a time-domain simulation was performed on the MATLAB® platform.

A lot of literature deals with the description and simulation of time series [12], [13]. The planned algorithm is also now expediently simulated in the time domain.

In Figure 15, the nodes calculated in the time-domain simulation are marked with capital letters A - G.

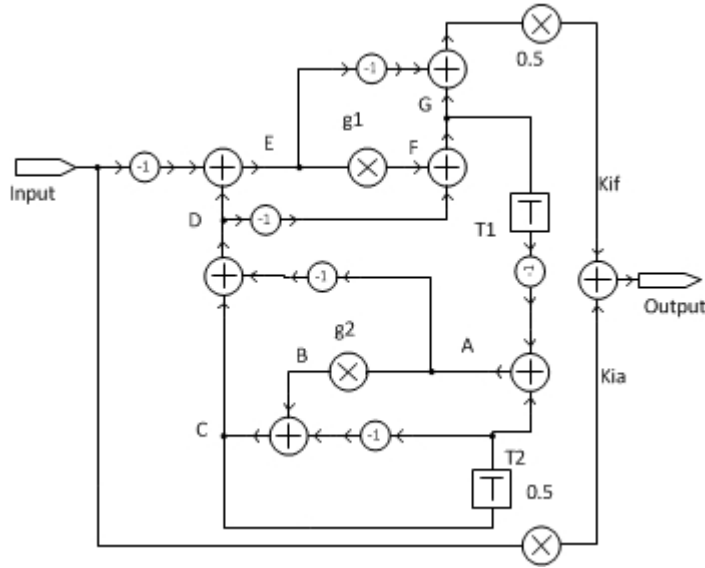


Figure 15

Detailed algorithm of UIC pilot signal remover - optimized

The order of the time-domain simulation fits well with the implementation with the signal processing processor; therefore the "m" source code has also been published. It can be downloaded from this link: https://bit.ly/UIC_filter1

After running the code, one of the results is shown in Figure 16.

Based on Figure 16, it can be seen that the response (weight function) to the Kronecker delta excitation signal has a stable decay. This result confirms the stability of the filter. Algorithm stability is also highly significant in the field of sound and 2D image processing [17]. The Fourier transform of the Kronecker delta is a constant "one". This means that during one sampling time unit (ideally "dt"), all spectral components are present in the excitation signal with unit amplitude. For this reason, it is also advisable to examine the spectral image of the response function, which in this case is the transfer function of the filter circuit. See Figure 17.

It can be seen from the transfer function that the filter has significant attenuation at a frequency of 2800 Hz. According to the result, the 2800 Hz harmonic pilot signal will be attenuated by approximately 80 dB. This value is much better than expected in the requirement.

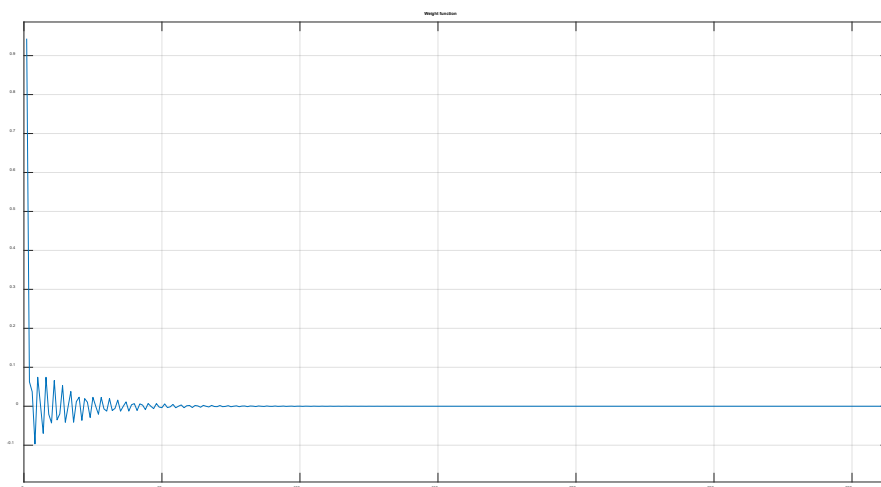


Figure 16

UIC pilot signal remover filter simulation result in time-domain

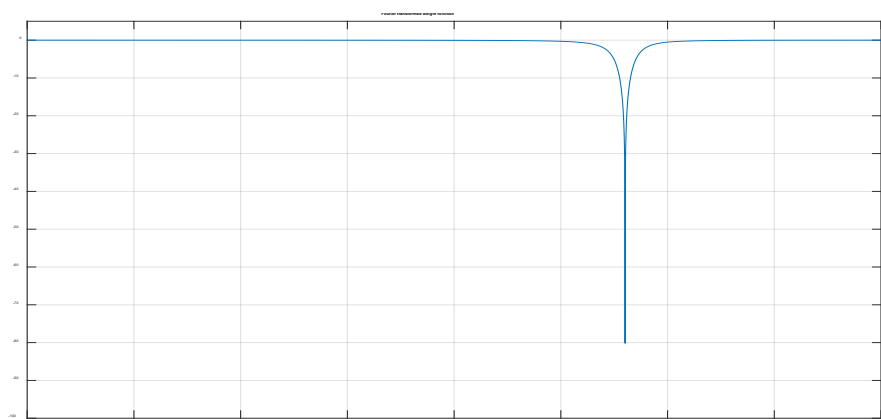


Figure 17

UIC pilot signal remover filter simulation result in frequency domain (Transfer function)

2.4 Questions of Realization

We kept the passivity in some steps of the design, thereby ensuring zero input signal stability of the DSP algorithm. The algorithm was also scaled. In this step, the absolute value of the constant multiplier value of the multiplier elements is kept below 0.5. The use of linear equivalents in the algorithm does not endanger its stability, but gives better dynamic range.

For better dynamics, the output 0.5 multiplier is also moved in front of the adding element (compare Figure 14 with Figure 15):

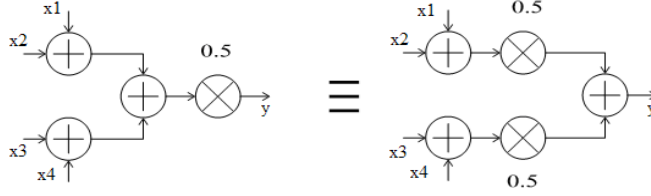


Figure 18

Output node conversion of UIC pilot signal remover

The linear equivalence in Figure 18 can also be seen mathematically.

When implementing DSP algorithms, numbers are represented using a finite number of bits. The effect of the finite number of bits becomes apparent when the algorithm is implemented cost-effectively, that is, by incorporating the actually necessary hardware components. For real-time DSP solutions, we often use fixed-point number representation, for example according to the IEEE1.15 or IEEE1.31 format. In the case of such solutions, the loss of even a few bits from the useful operation results in a low-quality solution. The finite number of bits means a perceptible quantization error and this error affects the operation of the algorithm in a non-linear manner. In addition to the appearing quantization noise, a bigger problem is that the value of the sample at certain nodes of the algorithm is either increased or decreased by the quantization effect. With this DSP algorithm, the pseudo energy of the signal represented by the patterns generated in each node can also increase. An increase in energy threatens passivity. In case of loss of passivity, we lose the guarantee of zero input stability.

An example of this is presented, in which mathematical rounding is applied to the output of the multiplier components of the scaled UIC filter based on the IEEE1.15 representation in the simulation. After multiplication, the number represented by IEEE1.31 is converted to IEEE1.15 format by mathematical rounding. Rounding in the simulation is done with the following MATLAB® code:

```
function [q]=q1(x,n)
q=(round(x*2^n))/2^n (15)
```

The "n" represents the number of bits between the radix point and the LSB, "x" is the number to be rounded, which must be entered in decimal form. The function returns the value "q". We can see the impulse response in Figure 19.

The weight function of the algorithm containing mathematical rounding goes into an infinite limit cycle from the 100th beat, marked with red line in the Fig 19. With the mathematical rounding, we violated the passivation conservation rule, so we lost the zero-input signal stability guarantee.

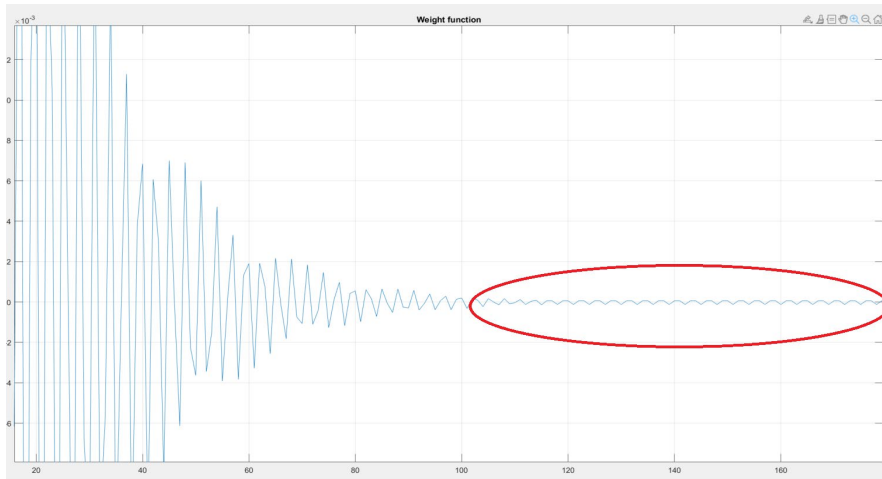


Figure 19

Infinite limit cycle in weight function

Of course, this phenomenon is also present in the case of floating-point number representation using a high number of bits, but the hardware, which is significantly oversized for the task, masks the errors. The existence of the error carries a source of danger, since nothing guarantees when a small-amplitude limit cycle will degenerate into an uncontrolled, high-amplitude excitation.

The passivity can also be maintained in the implementation of the DSP algorithm, if the quantization is performed with absolute value truncation. In this case, the samples representing the signal are pushed towards the zero point at each quantization point, that is, the signal energy is reduced in each case.

Conclusions

This paper illustrates the process efficiently designing an effective DSP algorithm and how to guarantee its' stable operation. The UIC pilot filter presented herein, can be successfully implemented, with minimal hardware requirements.

Acknowledgement

This work was supported by Óbuda University.

I thank the management of Óbuda University for supporting my work and I thank the Acta Polytechnica Hungarica journal for publishing my article.

References

- [1] International Union of Railways: UIC codex 751-3 2005./4
- [2] T. Wühl: DSP algorithms OE – KVK 2116, Budapest, 2014
- [3] T. Wühl: Wave Digital Filtering, OE-KVK 2073. Budapest, 2010

- [4] A. Fettweis: Wave Digital Filters: Theory and Practice, Proceedings of the IEEE, Vol. 74, No. 2, pp. 270-327, 1986
- [5] T. Wühl: P Introduction in to the MATLAB: Telecommunication and DSP, OE-KVK 2071, Budapest, 2010
- [6] K. Géher, J. Solymosi: Linear Circuit Design, Budapest, 1992
- [7] R. Lyons: Understanding Digital Signal Processing, Prentice Hall, Upper Saddle River, NJ, 2001
- [8] A. Fettweis, K. Meerkötter: On Adaptors for Wave Digital Filters, IEEE Transactions on Audio, Speech, and Signal Processing, Vol. ASSP-23, No. 6, pp. 516-525, 1975
- [9] MathWorks: Representation of Numbers in Matlab, <https://uk.mathworks.com/matlabcentral/fileexchange/33992-representation-of-numbers-in-matlab>
- [10] L. Gazsi: Explicit Formulas for Lattice Wave Digital Filters, IEEE Transactions on Circuits and Systems, Vol. 32, No. 1, pp. 68-88, 1985
- [11] M. Tockner, H. G. Brachtendorf, M. Steiger: Design of Wave Digital Filters with the TU Delft Toolbox, Proceedings of 2021 16th International Conference on Telecommunications (ConTEL), pp. 18-22, 2021
- [12] E.-L. Hedrea, R.-E. Precup, R.-C. Roman, E. M. Petriu: Tensor Product-based Model Transformation Approach to Tower Crane Systems Modeling, Asian Journal of Control, Vol. 23, No. 3, pp. 1313-1323, 2021
- [13] C. Andreeski, D. Mechkaroska: Modelling, Forecasting and Testing Decisions for Seasonal Time Series in Tourism, Acta Polytechnica Hungarica, Vol. 17, No. 10, pp. 149-171, 2020
- [14] D. Singh, A. Shukla: Manifold Optimization with MMSE Hybrid Precoder for Mm-Wave Massive MIMO Communication, Romanian Journal of Information Science and Technology, Vol. 25, No. 1, pp. 36-46, 2022
- [15] C. Pozna, R.-E. Precup: Aspects Concerning the Observation Process Modelling in the Framework of Cognition Processes, Acta Polytechnica Hungarica, Vol. 9, No. 1, pp. 203-223, 2012
- [16] N.-A. Serban, R. Hobincu: Computer Vision Algorithm for Detecting Resistor Color Codes, Romanian Journal of Information Science and Technology, Vol. 24, No. 3, pp. 321-333, 2021
- [17] A.-I. Szedlak-Stinean, R.-E. Precup, E. M. M. Petriu, R.-C. Roman, E.-L. Hedrea, C.-A. Bojan-Dragos: Extended Kalman filter and Takagi-Sugeno Fuzzy Observer for a Strip Winding System, Expert Systems with Applications, Vol. 208, paper 118215, 2022