Radio Science

RESEARCH ARTICLE
10.1029/2019RS006821

Key Points:
- A method of designing true time delay filter is presented
- All-in-one and cascaded arrangement is formulated
- Cascaded filtering saves DSP resources when compared with all-in-one filtering operation

Correspondence to:
B. Damtie,
bayliedamtie@yahoo.com

Citation:
Lehtinen, M., Damtie, B., & Orispää, M. (2019). Optimal true time delay filter with application to FPGA firmware-based phased array radar signal processing. *Radio Science*, *54*, 810–821. https://doi.org/10.1029/2019RS006821

Received 20 FEB 2019
Accepted 13 AUG 2019
Accepted article online 18 AUG 2019
Published online 5 SEP 2019

Optimal True Time Delay Filter With Application to FPGA Firmware-Based Phased Array Radar Signal Processing

Markku Lehtinen[1], Baylie Damtie[2], and Mikko Orispää[1]

[1]Sodankylä Geophysical Observatory, University of Oulu, Sodankylä, Finland, [2]Baylie Damtie, Washera Geospace and Radar Science Laboratory, Bahir Dar University, Bahir Dar, Ethiopia

Abstract The European Incoherent Scatter-3D phased array radar system will be largely based on field-programmable gate array (FPGA) firmware electronics that carry out the signal processing by using different digital filters. In this paper we have presented a method of designing an optimal true time delay finite impulse response filter with applications to an FPGA firmware-based multichannel signal processing system. The method provides an optimal true time delay finite impulse response filter with the desired responses at both band pass and stopband. This is possible by finding a mathematical minimization solution for the total power of all filter coefficients longer than a prespecified half-length. The analysis is based on freely choosing the responses in the transition band until user-specified desired responses are achieved. We have investigated the performance of these optimal digital filters in terms of the required digital signal processing (DSP) resources in GMACs (giga multiply accumulates per second) by considering both all-in-one stage filtering and cascaded solutions for ion and plasma line incoherent scatter radar measurements. We have shown that the cascaded solution provides more efficient utilization of DSP resources and hence represents the optimal choice for processing the proposed European Incoherent Scatter-3D phased array radar signals. An example is demonstrated in which 906.88 GMACs are required to process 208 ion line beams with 2×4-bit resolution in all-in-one stage processing, as compared to the 79.16 GMACs needed for a similar task in a cascaded solution.

1. Introduction

Scattering of electromagnetic waves from accelerating electrons in the upper atmosphere has been developed successfully as a powerful ionospheric plasma diagnostic tool (Hargreaves, 1991). This is carried out by transmitting high-power electromagnetic waves radiated from a transmitting antenna into the ionosphere (the partially ionized part of the upper atmosphere) and receiving the backscattered radio signals. The incoherent scatter radar measurement technique is presented in detail in Nygren (1996). The received signal, which is a narrow passband signal, undergoes various signal processing stages to extract the baseband information-bearing signal. The signal processing in an incoherent scatter radar usually includes amplification, frequency mixing, filtering, correlation, averaging, and estimation of plasma parameters. In the early years of incoherent scatter radar measurements, most of the signal processing operations were carried out by using dedicated analog and digital hardware (see, e.g., Wannberg et al., 1997, and references therein).

With rapidly evolving digital technology, scientific instruments are increasingly becoming software dependent with very limited hardware components. Accordingly, several researchers around the world have presented various methods of replacing the analog and digital hardware components of the incoherent scatter radar system by more flexible software. For example, the first experiment that replaced the standard European Incoherent Scatter (EISCAT) Svalbard radar receiver at 70 ± 2.5 MHz intermediate frequency (IF) stage by a spectrum analyzer, which acted as a combination of a down-converter, an analog to digital converter (ADC), and a quadrature detector giving complex digital samples at a rate of 4 MHz, was carried out by Lehtinen et al. (2002) in 1999. In this experiment it was possible for the first time to store a total multichannel signal instead of separate lagged products for each frequency channel and to carry out channel separation, clutter suppression, correlation, lag profile inversion, and estimation of plasma parameters using software. Similar work has also been reported using the Millstone Hill Incoherent Scatter radar system (Holt et al., 2000). This was done by band-pass sampling of 2.25 MHz intermediate frequency at 1-s intervals and followed by filtering and decimation of the complex baseband samples and computation of the lag profile matrix of the decimated samples using software.

The EISCAT-3D phased array radar, which will be the first of its kind and the most advanced incoherent scatter radar system in the world, will rely on advanced digital signal processing technique to measure ionospheric plasma parameters in three dimensions. The science case has been well established (McCrea et al., 2015), and the new radar will capture for the first time the three-dimensional spatiotemporal dynamics of the ionosphere. The system is expected to have five phased-array antenna fields situated in the northern parts of Finland, Norway, and Sweden, and each field will contain 9,919 crossed dipole antenna elements. The transmitter, which will be established at Skibotn in Norway, will have a 233-MHz transmitting frequency. The receivers at each site will measure the scattered radio signals from the accelerating electrons in the ionosphere. The acceleration is induced by the transmitted electromagnetic waves. All five sites will have sensitive receivers to measure the returned radio signals. The first-stage receiver unit is proposed to have front end (low noise amplifier, analog filter, and ADC), true time delay (TTD) filters, and a beam former. The efficiency of the digital processing system after the ADC will be critical in order to handle several thousand beams simultaneously and transfer the data with a feasible transfer rate.

In this paper we present a method of finding optimal digital finite impulse response (FIR) filters, including TTD filtering, mirror image suppression, and passage of the suitable band-pass signal for further processing, with application to the EISCAT-3D phased array radar system and other similar FPGA firmware-based digital signal processing needs. The filtering operation is carried out in real time in the FPGA firmware of the receiver electronics. In this kind of systems one usually first uses a low noise amplifier to amplify the very weak received signal and then passes it to the analog filter that suppresses the unwanted noise power, whose frequencies are outside the desired band width. The analog to digital conversion can be carried out by employing an undersampling method, which produces multiple copies of the spectrum of the desired signal at an integer multiple of the sampling frequency. This means that in the next stage of the signal processing, we need to select one suitable copy and suppress the other extra images and also carry out TTD operations in order to compensate for the particular delay associated with each receiving antenna element. We have carried out a detailed investigation of how to carry out these filtering and decimation operations in an efficient manner in the FPGA firmware.

Several methods of designing optimal digital FIR filters are available in various publications including windowing (Oppenheim, 1999) and the Parks and McClellan algorithm (Parks & McClellan, 1972), and comprehensive presentations are available in Proakis and Manolakis (1996). FIR design methods and specifications depend on three basic parameters: transition width, ripples in the peak passband and stopband, and number of taps (filter order). Most often any two of these parameters are fixed, and one tries to find the optimal values for the remaining parameter using suitable design algorithms. In the case of applications with more stringent requirements, where it is necessary to obtain more flexible and accurate control over the frequency response of the designed filter, the optimal design of FIR filters is still an active research topic; for example, see Revuelta and Arribas (2018) and the references therein. The algorithm presented here is based on fixing the exact desired frequency response at the passband and stopband and then finding the optimal filter by varying the ramp of the transition band in order to obtain the fixed desired responses. We then investigate the performance of the optimal filter in terms of the required DSP resources in the FPGA firmware, evaluated in GMACs, by implementing the processing in both an all-in-one stage filtering and a multistage (cascaded) solution. The study shows the performance as a function of the data resolution in bits and the number of beams by assuming a realistic data transfer rate for the case of EISCAT-3D phased array radar system.

2. Mathematical Formulation

In this work our mathematical formulation is based on the definition and concept of set theory and related rules. This allowed us to use some features of the programming language R, in which we plan to publish the design routines as a free package.

Let us consider a discrete Fourier transform sequence pair of a filter $x(n)$ and its frequency response $X(i)$. We can consider that the set of frequency indices $\{i | i = 1, \ldots, N\}$ is a union of three disjoint sets A_0, A_1, and B such that

- A_0 is a set containing stopband indices where we want the frequency response $X(A_0)$ to be 0.
- A_1 is a set containing passband indices where we want the frequency response $X(A_1)$ to be 1 or any other desired form (e.g., it becomes $e^{-j\omega\tau_d}$ when we have TTD τ_d).

- B is a set containing transition band indices where we want to choose the frequency response $X(B)$ by the design routine in order to obtain a FIR filter with minimum number of taps in time domain $x(n)$.

We can also define a set A as a union of passband and stopband indices by $A = A_0 \cup A_1$. We can now consider the set of filter indices $\{n | n = 1, \ldots, N\}$ in time domain as a union of two disjoint sets: a set C that contains user-specified coefficients of $x(n)$ and its complement C^c (called the long coefficients). We then find the values $X(i), i \in B$ so that the relative power of the long coefficients denoted here by R_L such that

$$R_L = \frac{\sum_{n\in C^c}^{\#C^c} |x(n)|^2}{\sum_{n=1}^{N} |x(n)|^2}, \tag{1}$$

is as small as possible. Here $\#C^c$ denotes the number of elements of the set C^c (cardinality of C^c). This is a quadratic problem, and the solution can be found by the set of linear equations.

The stopband and passband are set to the required fixed values, and we only change coefficients of the transition band. When the set of short filter coefficients is big enough, the minimization typically finds very low values for the total power of long coefficients. In the algorithm we increase the set of short coefficients (repeating the minimization for each new set C) until the total power of the long coefficients goes below a user-defined goal. The long coefficients can then be truncated away with minimal harm to the frequency response in the stopband and passband.

Let us investigate a discrete filter $x_q = \sum_{q=0}^{N-1} p_r e^{i2\pi rq}$ given as a discrete Fourier transform (DFT) of a power transfer passband function p_r. We assume that the frequency index is divided into two sets, A and B, such that the values of p_r are real and have given values p_r when $r \in A$. When $r \in B$, p_r must be chosen to minimize the power of coefficients over a set C using the relation

$$\sum_{q\in C} |x_q|^2 = \sum_{q\in C} \left| \sum_{r\in A} p_r e^{i2\pi rq} + \sum_{r\in B} p_r e^{i2\pi rq} \right|^2. \tag{2}$$

Let us consider an arbitrary $\tilde{p}_r$ defined on the set $r \in B$. At the minimum, we can get

$$\left.\frac{\mathrm{d}}{\mathrm{d}\lambda}\right|_{\lambda=0} \sum_{q\in C} \left| \sum_{r\in A} p_r e^{i2\pi rq} + \sum_{r\in B} (p_r + \lambda\tilde{p}_r) e^{i2\pi rq} \right|^2 = 0, \forall \tilde{p}_r, \tag{3}$$

which is the same as

$$\sum_{q\in C} \left(\sum_{r\in A\cup B} p_r e^{i2\pi rq} \sum_{r\in B} \tilde{p}_r e^{-i2\pi rq} + \right. \tag{4}$$

$$\left. \sum_{r\in A\cup B} p_r e^{-i2\pi rq} \sum_{r\in B} \tilde{p}_r e^{i2\pi rq} \right) = 0, \forall \tilde{p}_r, \tag{5}$$

or

$$\mathrm{Re} \sum_{q\in C} \left(\sum_{r\in A\cup B} p_r e^{i2\pi rq} \sum_{r\in B} \tilde{p}_r e^{-i2\pi rq} \right) = 0, \forall \tilde{p}_r. \tag{6}$$

Further, by using the Kronecker delta notation and choosing $\tilde{p}_r = \delta_{r-r'}$, $r' \in B$, we see that this is equivalent to

$$\mathrm{Re} \sum_{q\in C} \sum_{r\in A\cup B} p_r e^{i2\pi(r-r')q} = 0, \forall r' \in B. \tag{7}$$

This is a set of $\#B$ (the notation meaning the number of elements in the set B) linear equations for the $\#B$ unknowns $p_r, r \in B$, and it has a mathematical solution. This can be easily seen when the constant terms are written on the right-hand side of the equation as

$$\mathrm{Re} \sum_{r\in B} \sum_{q\in C} p_r e^{i2\pi(r-r')q} = -\mathrm{Re} \sum_{q\in \mathbf{C}} \sum_{r\in A} p_r e^{i2\pi(r-r')q}, \quad \forall r' \in B. \tag{8}$$

In the case of complex spectral response, where p_r is complex, the derivation is just a little bit more difficult. Above, we proceeded from (6) to (7) by applying the condition $\tilde{p}_r = \delta_{r-r'}$ for all $r' \in B$. In the complex case, we can insert first $\tilde{p}_r = \delta_{r-r'}$ and then $\tilde{p}_r = i\delta_{r-r'}$ for all $r' \in B$ directly in (4) and get the two sets of equations

$$\sum_{q\in\mathbf{C}} \left(\sum_{r\in A\cup B} p_r e^{i2\pi(r-r')q} \pm \sum_{r\in A\cup\mathbf{B}} \bar{p}_r e^{-i2\pi(r-r')q} \right) = 0, \qquad \forall r' \in B. \tag{9}$$

The set corresponding to the + sign is the same as (7), while the set corresponding to the − sign is also otherwise the same as (7), but starting with Im instead of Re. Together, they are the same as a single set of complex equations same as (7) but without the starting Re. Again, by rearranging the terms with the unknowns p_r with $r \in B$ to the left side and the constant terms depending on p_r with $r \in A$ on the right side, we have a set of $\#B$ complex equations for $\#B$ complex unknowns p_r, $r \in B$

$$\sum_{r\in B}\sum_{q\in C} p_r e^{i2\pi(r-r')q} = -\sum_{q\in C}\sum_{r\in A} p_r e^{i2\pi(r-r')q}, \forall r' \in B. \tag{10}$$

instead of the similar real set of equations (8).

3. Numerical Implementation

The mathematical notation above is the suitable way of expressing the equations in a manner directly applicable to modern vector and matrix-based calculation systems like R, Python, and MATLAB. In this work we have used R, which is capable of vectored indexing through index masks using the sets A_0, A_1, A, B, and C. Using these sets and the number of elements in them, we can write the formulation in a form suitable for numerical calculation

$$\begin{aligned} &\sum_{i_r=1}^{\#B} p_{B_{i_r}} \sum_{i_q=1}^{\#C} e^{i2\pi(B_{i_r}-B_{i_{r'}})C_{i_q}} = \\ &-\sum_{i_q=1}^{\#C}\sum_{i_r=1}^{\#A} p_{A_{i_r}} e^{i2\pi(A_{i_r}-A_{i_{r'}})C_{i_q}}, \forall i_{r'} = 1 \dots \#B. \end{aligned} \tag{11}$$

Here we have assumed the set elements to be the actual index values. In our R code the corresponding sets are defined through logical vectors of values TRUE or FALSE defined over the Fourier index set from 1 to N so that B in our mathematical terminology is actually expressed by $(1 : N)[B]$ in R. With this convention $\#B$ corresponds to the number of elements in B.

We can also notice that in the sum $r \in A$ in the right-hand side of equation (10) only values in $r \in A_1$ contribute because $q_r = 0$ for all $r \in A_0$ by definition. This means that instead of A we should actually have A_1 in equation (11) to avoid wasting computational power in simply adding up zeroes. In order to simplify the formulation and avoid too deep nesting of indices, we denote A_1 by $A1$ to obtain

$$\begin{aligned} &\sum_{i_r=1}^{\#B} p_{B_{i_r}} \sum_{i_q=1}^{\#C} e^{i2\pi(B_{i_r}-B_{i_{r'}})C_{i_q}} = \\ &-\sum_{i_q=1}^{\#C}\sum_{i_r=1}^{\#A1} p_{A1_{i_r}} e^{i2\pi(A1_{i_r}-A1_{i_{r'}})C_{i_q}}, \forall i_{r'} = 1 \dots \#B. \end{aligned} \tag{12}$$

The optimal filter coefficients can then be found by implementing the numerical calculation using equation (11). This is done by using the specified TTD and the corresponding frequency responses at the band pass and stopband. We can then find a filter with a suitable number of taps, whose performance can easily be investigated.

Figure 1 demonstrates how we can find the optimal filter with a half passband width of 2 MHz when 233-MHz nominal frequency is sampled at a rate of 84 MHz and the transition band width is set to 1 MHz. We can see that the required frequency response of −90 dB at stopband, including at the location of the mirror image at 19 MHz, has been achieved (bottom curve). This demonstration is based on an all-in-one stage filtering arrangement, and this means that the filter removes the extra image of the signal, passes the desired passband, and also carries out TTD filtering simultaneously.

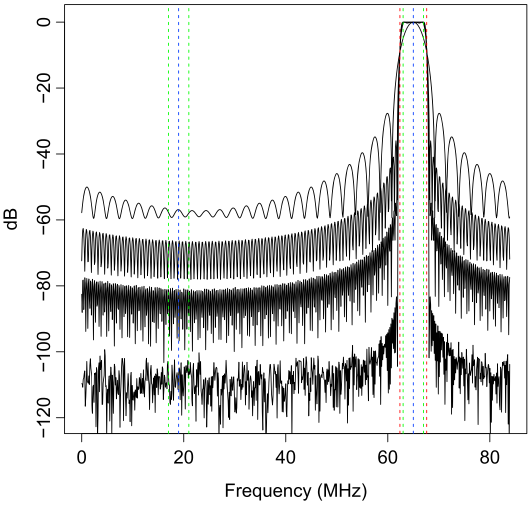

Figure 1. Method of finding optimal finite impulse response filter taps for FPGA firmware with application to an incoherent scatter phased array radar system (with center frequency of 233 MHz and sampling frequency of 84 MHz). The frequency response at the start of the computation (the curve above −60 dB), frequency responses between the desired response and beginning of computation (curves between −100 and −60 dB), and the desired frequency response (the curve below −100 dB).

The filter designed by this method is implemented in the FPGA firmware. It is important to measure the required DSP resources for all-in-one and cascaded filtering arrangements. In this work the DSP resources required for carrying out the filtering is measured in terms of GMACs (giga multiply-accumulates per second). We define GMACs using the number of taps in the FIR filter and the sampling frequency by

$$\mathrm{GMACS} = N \times f_s, \tag{13}$$

where N denotes the length of the FIR filter (total number of taps) and f_s is the sampling frequency in gigahertz, and the value obtained by equation (13) is multiplied by 2 when the impulse response is complex.

4. Filtering of the Ion Line Channels

In an incoherent scatter radar measurement the ion line contains valuable information about basic ionospheric parameters. The signal is formed by the backscattering of a transmitted radar signal by the ion acoustic waves, which basically consist of propagating compressions and rarefactions of plasma, in some ways like ordinary sound waves. The time-varying transmitted electromagnetic wave accelerates electrons, which in turn act like antennas and emit electromagnetic waves. These accelerating electrons are also under the influence of much heavier background ions. This means that the width of the backscattered (ion line) electromagnetic spectrum is a function of the movement of the ions, and thus it is a much narrower spectrum than one might expect by simply considering the motion of freely accelerating electrons.

Our interest is to design an optimal filter with application to the electronics of a phased array radar system. Thus, in this simulation we shall use the 233-MHz transmission frequency, which has been proposed for the new EISCAT-3D phased array radar system, and an 84-MHz sampling frequency (we assume undersampling) in the analysis of optimal FIR filters with application to both ion and plasma line spectra. We

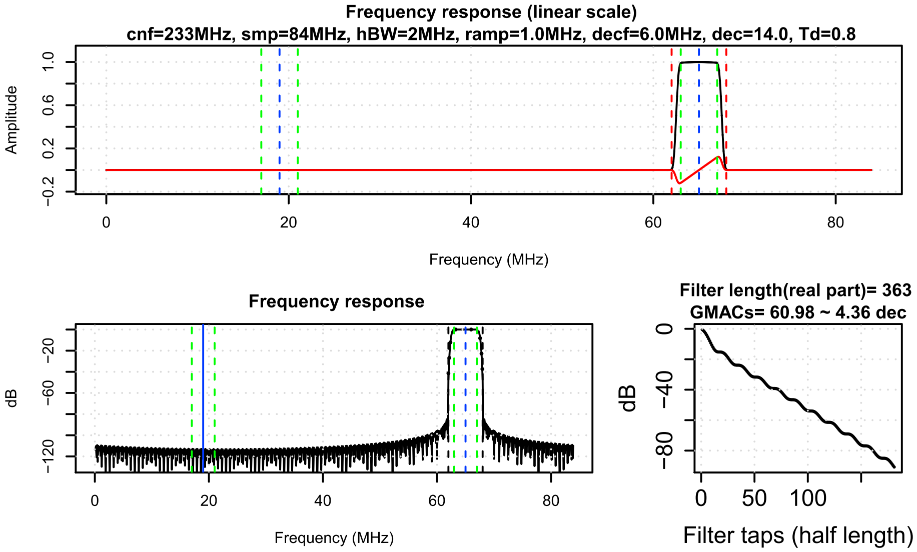

Figure 2. Ion line filtering (signal image and noise suppression and true time delay filtering carried out in a single-stage operation) with transmitter frequency of 233 MHz, sampling frequency of 84 MHz, and decimated frequency of 6 MHz. Frequency response of the filter in linear scale (top panel), frequency response in decibels (bottom left panel), and performance of the filter as a function of the number of taps in its half-length (bottom right panel). GMACs = giga multiply accumulates per second.

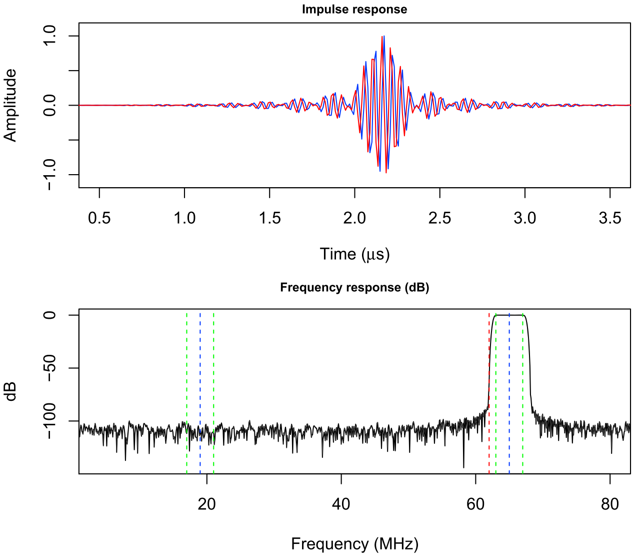

Figure 3. Impulse response of a single-stage filter for ion line filtering (top panel, real [blue] and imaginary [red]) and the corresponding frequency response in decibels (bottom panel) after truncation.

consider that a decimated IQ rate of 6 MHz with 2 MHz half band width is sufficient for measuring the ion line under all possible circumstances. The filtering operation, consisting of the removal of unwanted mirror image of the received signal, allowing the passage of the desired spectrum, TTD filtering and decimation, can be carried out either by one single stage of processing or by using cascaded multistage processing.

Figure 2 shows the main features of an optimal FIR filter for the case of a single-stage filtering (all-in-one) of the ion line, which incorporates removal of image of the signal, band pass, TTD filtering, and decimation in one filter, producing IQ samples at a rate of 6 MHz . We have displayed the frequency response in linear scale (top panel), frequency response in decibels (bottom left panel), and the performance of the filter as a function of its length (bottom right panel). In this figure the blue, green, and red vertical broken lines indicate the center frequency, edge of passband, and transition band of filter, respectively. It is clear that this filter passes the ion line band pass centered at $f_c - 2f_s = 65$ MHz with half band width of 2 MHz, where $f_c = 233$ MHz is the center frequency (transmission frequency) and $f_s = 84$ MHz is the sampling frequency. The mirror image and unwanted noise are sufficiently suppressed as we can see from the frequency responses in decibels. including at the location of the mirror image centered at $-f_c + 3f_s = 19$ MHz as shown by the solid blue vertical line (bottom right panel).

We found the desired filter by letting our algorithm optimize the responses of the transition band (ramp) and increasing the half-length of our filter until our goal for response accuracy was obtained. This is demonstrated in Figure 1. The bottom right panel of Figure 2 shows the accuracy of the filters at different stages of the iteration (for different filter lengths), and we see that one can decrease the DSP resources needed in the FPGA when the suppression requirements are not very high. This single-stage filtering of the ion line demands a relatively long filter, but it can be truncated with minimal effects on the performance. The top panel of Figure 3 shows the truncated complex impulse response of the filter, and the bottom panel displays the corresponding frequency response (displayed here to show that the required performance is maintained after suitable truncation). We see that the real and imaginary parts of the filter have each 363 filter taps

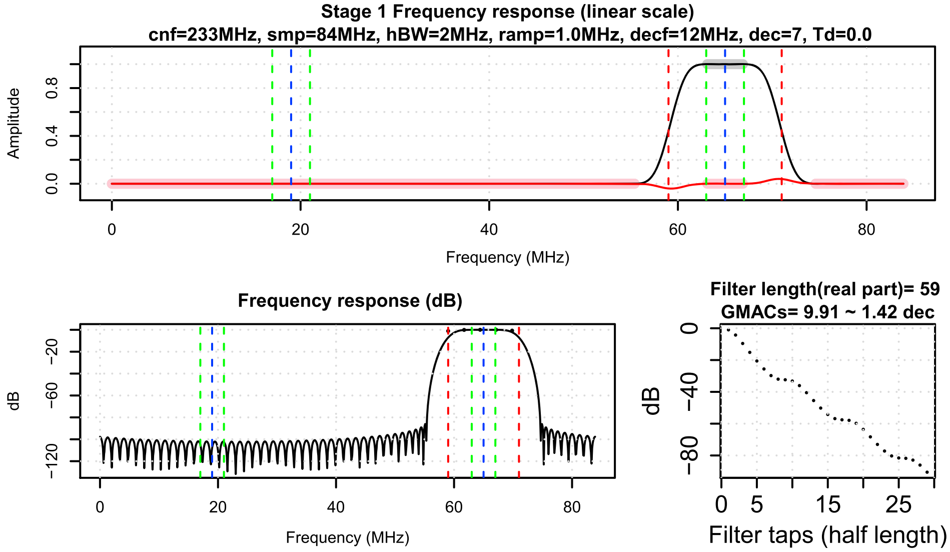

Figure 4. Frequency response in linear scale (top panel), frequency response in decibels (bottom left panel), and performance of different length filter (bottom right panel) of the first-stage filter in a multistage filtering of the ion line. GMACs = giga multiply accumulates per second.

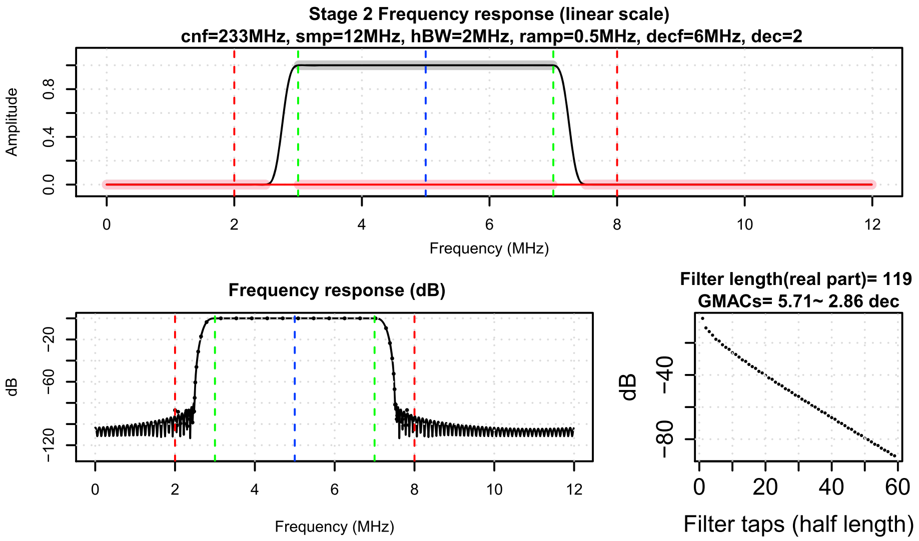

Figure 5. Frequency response in linear scale (top panel), frequency response in decibels (bottom left panel), and performance of different length filter (bottom right panel) of the second-stage filter in a multistage filtering of the ion line. GMACs = giga multiply accumulates per second.

and thus the required DSP resource in GMACs is 60.98. This is rather demanding in terms of DSP resource on FPGA firmware, though it can be handled with a modern sampler when we have limited beams. However, for processing many channels as in the case of EISCAT-3D, where one expects several thousand, the alternative is to use cascaded (multistage) filtering.

Figure 4 displays the frequency response of the first-stage filter in cascaded filtering of the ion line. The filter in this stage removes the image of the signal spectrum, passes the desired signal, and decimates the signal to an IQ rate of 12 MHz. The frequency response of the filter shows that we have sufficiently removed the image of the signal spectrum (see the response centered at 19 MHz) and also that the required performance is achieved in the stopband and passband. We have shown that we need DSP resources of 1.42 GMACs for this task. The characteristics of frequency response of the second-stage filter in cascaded filtering of the ion line are shown in Figure 5. This filter further decimates the signal to a rate of 6 MHz, and the frequency response has been maintained to have the desired shapes at the stopband and passband. It has 238 filter taps and requires 2.86 GMACs DSP resources.

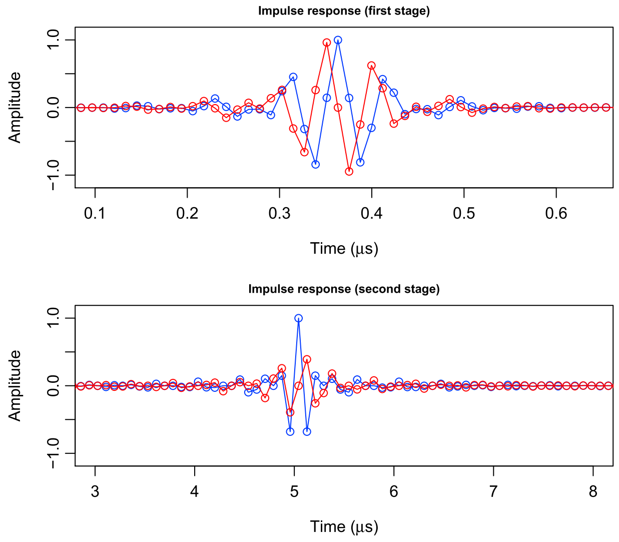

Figure 6. Impulse response of the first-stage filter (top panel, real [blue] and imaginary [red]) and second-stage filter (bottom panel, real [blue] and imaginary [red]) in multistage ion line filtering after truncation.

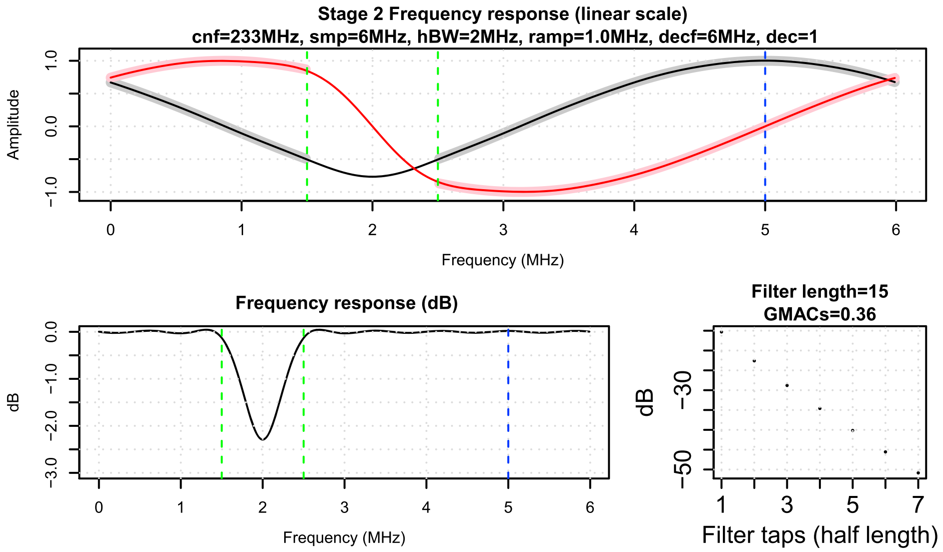

Figure 7. True time delay filter frequency response in linear scale (top panel), frequency response in decibels (bottom left panel), and performance of different length true time delay filter (bottom right panel) in cascaded filtering of the ion line. GMACs = giga multiply accumulates per second.

Figure 6 shows the complex filter taps, which are used in the FPGA firmware, for the first-stage filter (top panel) and second-stage filter (bottom panel). We see that the number of taps in these cascaded filters can easily be handled by the FPGA firmware. The truncation is carried out with minimal effects on the desired responses of the filters. The final stage of filtering of the ion line in the cascading processing is time delay filtering and the frequency response of the TTD filter is displayed in Figure 7. In the case of TTD filtering −50-dB response for stopband is sufficient, and as a result we need 15 filter taps and 0.36 GMACs DSP resources for this processing. The frequency response of a time delaying system is obtained by setting $z = e^{-i\omega}$ and that is

$$H_T(e^{i\omega}) = e^{-i\omega T_D}, \tag{14}$$

where T_D is the time delay, $\omega = 2\pi f t$ is the normalized frequency, and t is the sampling interval. This equation needs to be valid for the values of ω corresponding to the set A_1 defining the desired frequency range. Detailed information on TTD filter beam forming simulation results is available in our report to EISCAT (Orispää et al., 2014).

We now need to compare the performance of these two kinds of filtering of the ion line in terms of the DSP resources required in GMACs by considering different in-phase and quadrature component (IQ) resolutions (expressed in bits) and realistic data transfer rates. We shall use a complex data transfer rate of 10 gigabits per second (10 Gigabit Ethernet), which has been proposed for the EISCAT-3D phased array radar system. The number of maximum complex data streams, N, we can handle per second is then given by

$$N = \frac{R}{b \times f_{\mathrm{dec}}}, \tag{15}$$

where R is the data transfer rate, f_{dec} is the decimated frequency (the IQ rate) and b is the number of bits (resolution). This means that we can fit more data streams in our data transfer rate by decreasing the resolution.

Table 1
Comparisons of All-in-One and Cascaded Ion Line Filtering

IQ resolution (bit)	up to 2×18	2 × 16	2 × 8	2 × 4
Number of beams	10	52	104	208
GMACs, all-in-one filtering	43.6	226.72	453.44	906.88
GMACs, cascaded filtering	7.88	23	41.72	79.16

Note. GMACs = giga multiply accumulates per second.

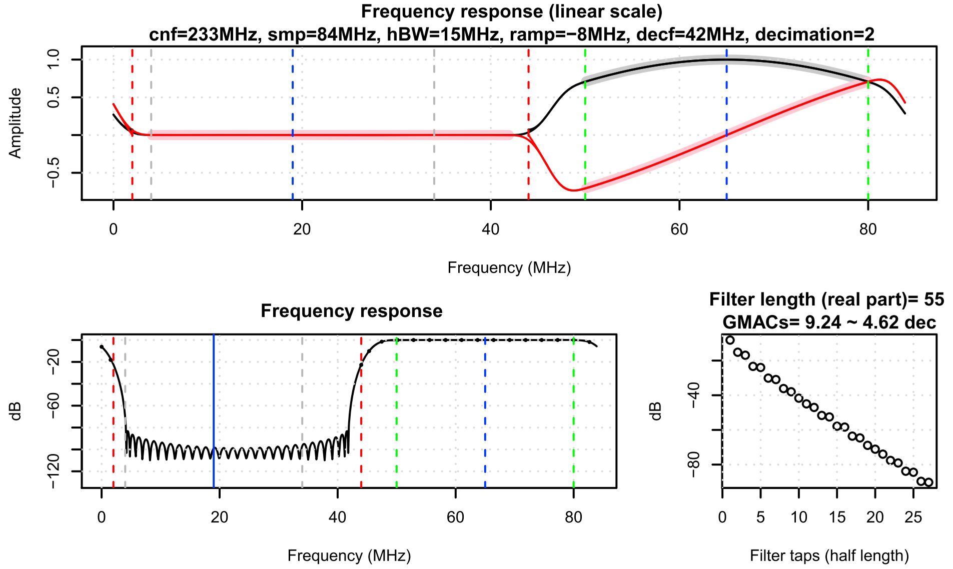

Figure 8. Plasma line filtering (signal image and noise suppression and true time delay filtering carried out in a single-stage operation) with transmitter frequency of 233 MHz, sampling frequency of 84 MHz, and decimated frequency of 42 MHz. Frequency response of the filter in linear scale (top panel), frequency response in decibels (bottom left panel), and performance of the filter as a function of the number of taps in its half-length (bottom right panel). GMACs = giga multiply accumulates per second.

The required GMACs will increase as we increase the number of data streams. As we can see from Table 1, the cascaded solution has a clear advantage in terms of the required DPS resources in GMACs especially when we wish to handle many beams.

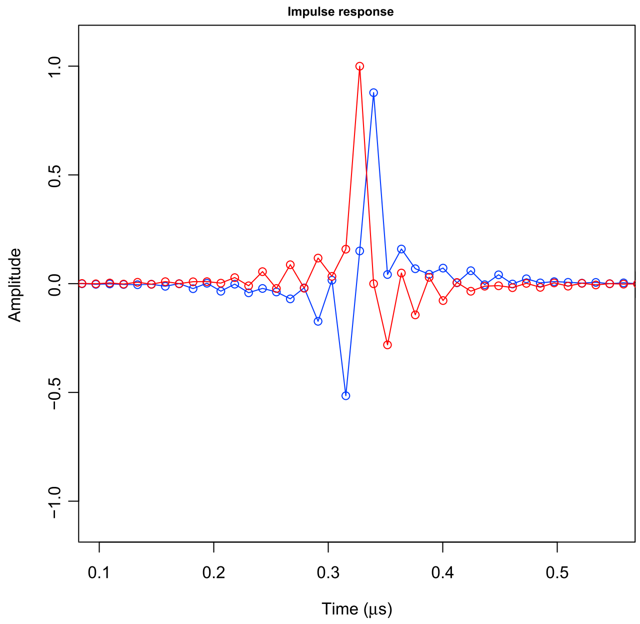

Figure 9. Impulse response of image signal suppression, true time delay, and decimation filter (all in one stage) with center frequency of 233 MHz, sampling frequency of 84 MHz, and decimated frequency of 42 MHz (real [blue] and imaginary [red]) .

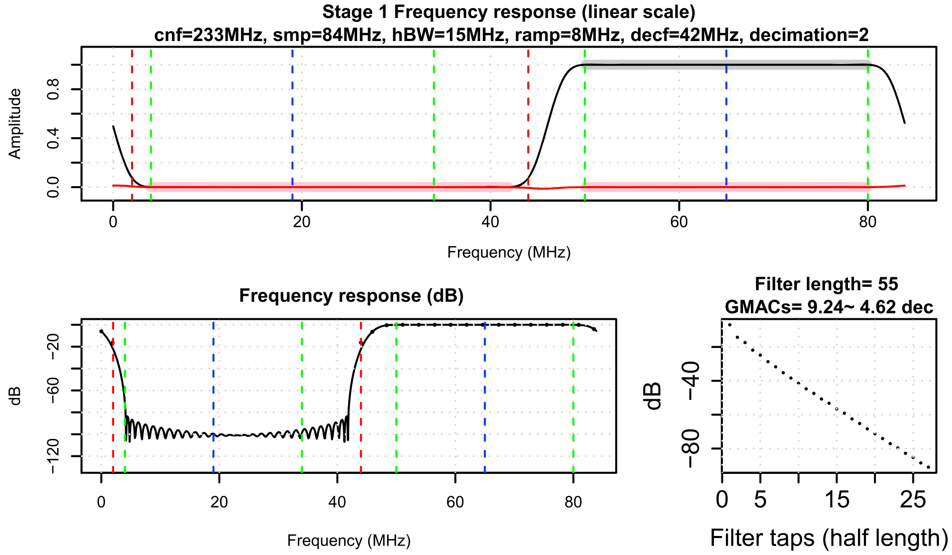

Figure 10. Frequency response of the first-stage filter in linear scale (top panel), frequency response in decibels (bottom left panel), and performance of the filter as a function of the number of taps in its half-length (bottom right panel) in cascaded filtering of the plasma line. The image signal removal and the decimation to 42 MHz are carried out by this filter. GMACs = giga multiply accumulates per second.

5. Filtering of the Plasma Line Channels

The plasma line is formed by the incoherent backscattering of the transmitted radar signal from electron plasma waves, which are formed by the energetic electrons. The ions cannot respond to these very high frequency plasma waves, and as a result the backscattered signal reflects the characteristics of the freely accelerating electrons. This means that it has a much wider spectrum than the ion line. In this demonstration we consider 15 MHz frequency as the half band width of the plasma line with an IQ rate of 42 MHz, wide enough to cover all the possible scenarios. The transmission frequency of 233 MHz and sampling frequency of 84 MHz are used here as in the preceding study of the ion line filtering.

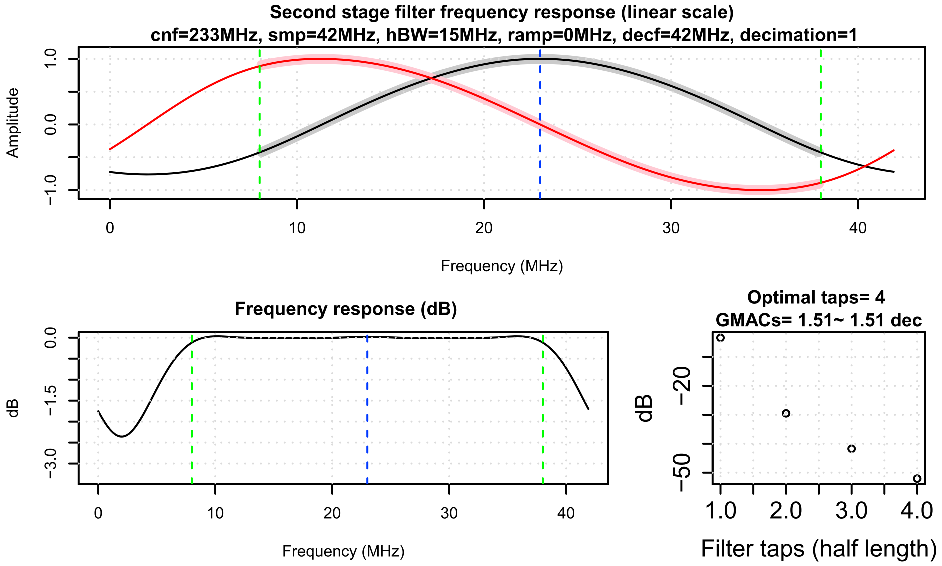

Figure 11. Frequency response of the second-stage time delay filter in linear scale (top panel), frequency response in decibels (bottom left panel), and performance of the filter as a function of the number of taps in its half-length (bottom right panel) in cascaded filtering of the plasma line. Time delay filtering is carried out by this filter. GMACs = giga multiply accumulates per second.

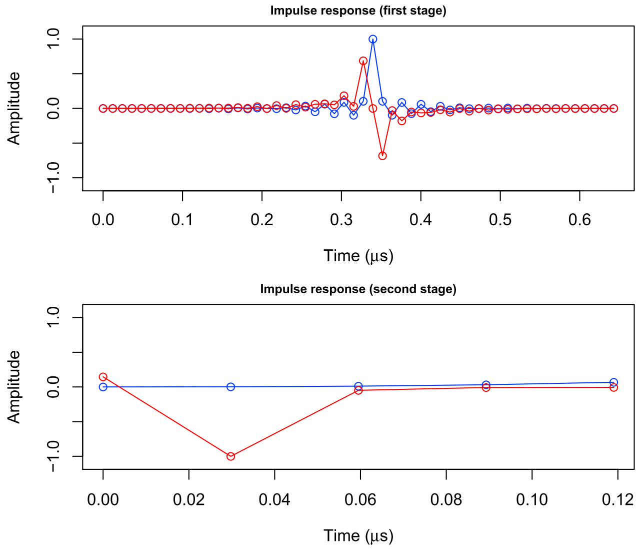

Figure 12. Impulse response of first-stage filter (top panel, real [blue] and imaginary [red]) and second-stage true time delay filter (bottom panel, real [blue] and imaginary [red]) in multistage plasma line filtering after truncation.

Figure 8 shows the frequency response of the all-in-one filter, which carries out mirror image signal removal and band pass and time delay filtering all in a single-stage processing scheme. We can see that the DSP resources required for this arrangement are 4.46 GMACs, which can be done using modern FPGA firmware for limited beams in phased array radar signal processing. When we consider a complex data stream using a data transfer rate of 10 gigabits per second (10 Gigabit Ethernet) and an IQ rate of 42 MHz with 2 × 16-bit resolution, from equation (15) we see that we can handle only a maximum of seven complex data streams. When we decrease the resolution of the samples to 2 × 4 bit, we can process 29 complex data streams and the DSP resources we need increase to 129.34GMACs. The actual filtering of the plasma line is carried out inside the FPGA firmware using a limited number of filter taps. Figure 9 depicts the complex filter taps of this all-in-one filter after truncation, and this length can easily be handled by the firmware.

Figure 10 depicts the frequency response of the first-stage filter in the cascaded filtering of the plasma line. In this stage the filtering includes mirror signal removal, band-pass filtering of the desired signal, and decimation to 42-MHz IQ rate. We can see that the filter has the desired response at the passband and stopband. The next stage of the cascaded solution involves time delay filtering, and the frequency response requirement in the stopband in decibels can be relaxed in this case. Here we have set it to be −50 dB as shown in the bottom right panel of Figure 11. We have also displayed the frequency response of the time delay filter in linear scale (top panel) and frequency response (bottom left panel) of Figure 11. We can see that we need only four filter taps and 1.51 GMACs DSP resources. The actual filtering is carried out by taking truncated filter taps with minimum effect on the desired response, and Figure 12 shows the corresponding complex filter taps for the first-stage filter (top panel) and second-stage filter (bottom panel). We can see that the required band-pass and stop band characteristics have been achieved with limited filter taps.

The performance comparison of the all-in-one stage filtering and the cascaded solution for the case of plasma line filtering can be made by considering a data transfer rate of 10 gigabits per second (10 Gigabit Ethernet) and an IQ rate of 42 MHz using equation (15). The result of this comparison study is shown in Table 2.

Table 2
Comparisons of All-in-One and Cascaded Plasma Line Filtering

IQ resolution (bit)	up to 2×18	2 × 16	2 × 8	2 × 4
Number of beams	10	7	14	29
GMACs, all-in-one filtering	46.2	32.34	68.75	133.98
GMACs, cascaded filtering	19.72	15.19	25.76	48.41

Note. GMACs = giga multiply accumulates per second.

We see that when the number of beams is increased, the cascaded filtering provides the means of handling the computational tasks with relatively small GMACs DSP resources. For example, to filter 29 beams with 2 × 4-bit resolution, we see that all-in-one filtering requires 133.98 GMACs, while the cascaded filtering demands only 48.41 GMACs.

6. Conclusions

We have presented a method of finding TTD filters with application to FPGA firmware-based signal processing, applicable to realistic incoherent scatter radar systems including EISCAT-3D. The mathematical foundation of the algorithm, which has been used to search a filter with specified frequency responses at passband and stopband, has been presented in a manner suitable for numerical implementation in R. The analysis relies on achieving the desired responses by freely choosing the number of filter taps until a user-specified goal is achieved. The basic solution is very fast since it is noniterative and vectorized in R. We have demonstrated the characteristics of responses of different TTD filters with application to ion and plasma lines incoherent scatter measurement. The illustration was carried out by considering the proposed EISCAT-3D transmitting radar frequency of 233 MHz with the assumption of undersampling.

We have investigated the performance of a TTD filter in terms of the required DSP resources in GMACs during both all-in-one stage processing and multistage (cascaded) filtering. When one faces the problem of processing several thousand beams at the same time, as in the case of EISCAT-3D, we have shown that the cascaded solution gives optimal performance in terms of the required DSP resources in GMACs. For example, we have found out that 906.88 GMACs is needed for all-in-one stage processing of the ion line IQ with 6-MHz flow rate and 2 × 4-bit resolution, as compared to 79.16 GMACs needed to carry out the same task in a cascaded arrangement.

Acknowledgments
B. Damties' work has been partially supported by Air Force Office of Scientific Research, Air Force Material Command USAF under Award 41 FA9550-16-1-0070. The programs we have developed to produce the figures can be obtained online (http://urn.fi/urn:isbn:9789526205878).

References

Hargreaves, J. K. (1991). *The solar-terrestrial environment*. Cambridge, UK: Cambridge University Press.

Holt, J. M., Erickson, P. J., Gorczyca, A. M., & Grydeland, T. (2000). MIDAS-W: A workstation-based incoherent scatter radar data acquisition system. *Annales Geophysicae*, *18*, 1231–1241.

Lehtinen, M, Markkanen, J., Väänänen, A., Huuskonen, A., Damtie, B., Nygren, T., & Rahkola, J. (2002). A new incoherent scatter technique in the EISCAT Svalbard Radar. *Radio Science*, *37*(4), 1050. https://doi.org/10.1029/2001RS002518

McCrea, I., Aikio, A., Alfonsi, L., Belova, E., Buchert, S., Clilverd, M., et al. (2015). The science case for the EISCAT_3D radar. https://doi.org/10.1186/s40645-015-0051-8, Progress in Earth and Planetary Science.

Nygren, T. (1996). *Introduction to incoherent scatter measurements*. Sodankylä, Finland: Invers Oy.

Oppenheim, A. V. (1999). *Discrete-time signal processing*. Noida, Uttar Pradesh, India: Pearson Education India.

Orispää, M., Virtanen, I., & Lehtinen, M. (2014). EISCAT-3D final report of Work Package 11, Deliverable D11.2 of WP11: Software theory and implementation (Report No 67): Sodankylä Geophysical Observatory.

Parks, T., & McClellan, J. (1972). Chebyshev approximation for non-recursive digital filters with linear phase. *IEEE Transactions on Circuit Theory*, *19*(2), 189–194.

Proakis, J. G., & Manolakis, D. G. (1996). *Digital signal processing: Principles, algorithms, and applications*. Upper Saddle River, NJ: Prentice-Hall, Inc.

Revuelta, S. J., & Arribas, J. I. (2018). A new approach for the design of digital frequency selective FIR filters using an FPA-based algorithm. *Expert Systems With Applications*, *106*, 92106.

Wannberg, G., Wolf, I., Vanhainen, L. G., Koskenniemi, K., Röttger, J., Postila, M., et al. (1997). The ElSCAT Svalbard radar: A case study in modern incoherent scatter radar system design. *Radio Science*, *32*, 2283–2307.