Comparison of delayless digital filtering algorithms and their application to multi-sensor signal processing

Anna Swider*, Eilif Pedersen

Abstract

In the phase of industry digitalisation, data are collected from many sensors and signal processing techniques play a crucial role. Data preprocessing is a fundamental step in the analysis of measurements, and a first step before applying machine learning. To reduce the influence of distortions from signals, selective digital filtering is applied to minimise or remove unwanted components. Standard software and hardware digital filtering algorithms introduce a delay, which has to be compensated for in order to avoid destroying signal associations. The delay from filtering becomes more crucial while analysis measurement from multiple sensors, therefore in this paper we provide an overview and comparison of existing digital filtering methods with application based on real-life marine examples. Additionally, design of special purpose filters is a complex process and for preprocessing data from many sources, application of digital filtering in the time domain can have a high numerical cost. For this reason we describe Discrete Fourier Transformation digital filtering as a tool for efficient sensor data preprocessing, which does not introduce a time delay and has low numerical cost. The Discrete Fourier Transformation digital filtering has a simpler implementation and does not require expert-level filter design knowledge, what is beneficial for practitioners from various disciplines. Finally, we exemplify and show the application of the methods on real signals from marine systems.

Index Terms

digital filtering, DFT, preprocessing, delay, delayless preprocessing, synchronisation, big data, IoT, sensors.

I. Introduction

DATA analysis has a vital role in many different industries, e.g.: medicine (gens analysis), economics (stock exchange) and in marine engineering (the ship industry enters the Shipping 4.0 phase Rødseth et al. (2016)). There is a need for collecting and processing huge quantities of measurements as time series data (signals) which comes from many sources, often referred to as big data DNVGL (2017).

The marine industry is now entering the challenging phase of smarter shipping, including on-board monitoring systems, and advisory tools. Modern vessels will be equipped with various on-line data collection and advanced monitoring systems. The on-board measurements from sensors of many installations play a crucial role, and their availability expands the functionality of marine products. The aim of data analysis in the marine application is developing on-shore and on-board advisory tools using prediction of propulsion power or ship performance monitoring, as well as enhancing knowledge about specific systems and components, and the relationship between systems Swider & Pedersen (2017).

Machine learning algorithms and statistical modelling become widely used tools in equipment monitoring and advisory systems. However they are sensitive to data quality, and in particular to relationships between subsystems retained as correlations in the data. A fundamental step in the analysis of measurements, and before applying machine learning is data preprocessing García et al. (2016), Taleb et al. (2015). Unfortunately, in the literature from different industries e.g. Kuhn & Johnson (2013), the importance of the quality of the time series is limited. Because measurements play an important role in marine applications, proper data preprocessing and improvement of their quality is critical to ensure correct interpretation on board the vessel or during off-line analysis. A major source of the disturbances and distortions in measurements is the Data Acquisition System (DAS). The role of the DAS is the collection of measurements of the desired variables, transmission and conversion of the recorded signals to digitized form Bendat & Piersol (2010). Among the most common disturbances and distortions are Vaseghi (2009), Bendat & Piersol (2010): white noise, poor calibration and digitization effects (ADC quantization or aliasing) Randall (2011). Data cleaning is a necessary stage, where distortions and disturbances are eliminated as far as possible. It plays an important role during data analysis Frnay & Verleysen (2014). A typical data analysis scheme is depicted in Figure 1. It contains some major steps like data collection, data cleaning and feature extraction before the machine learning or statistical modelling can be applied. In this paper we focus on the data cleaning stage.

Processing of data from several marine systems creates many challenges. One of them is lack of data synchronisation, which can be introduced by specific systems setups, different time intervals or preprocessing. In the literature from data analytics, data

Anna Swider*, correspondence author: anna.swider@rolls-royce.com, is with Rolls-Royce Marine AS and Norwegian University of Science and Technology (NTNU), Department of Marine Technology

Eilif Pedersen is with Norwegian University of Science and Technology (NTNU), Department of Marine Technology

Manuscript received February 14, 2018. Manuscript revised June 29, 2018.

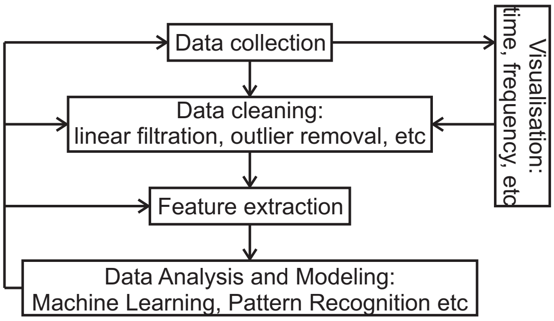

Data collection
Data cleaning:
linear filtration, outlier removal, etc
Feature extraction
Data Analysis and Modeling:
Machine Learning, Pattern Recognition etc
Visualisation:
time, frequency, etc

Fig. 1: Data analysis block diagram

cleaning is mainly based on PCA (Principal Component Analysis), outlier detection and NaN removal Perera (2017), Qiu et al. (2016), Slavakis et al. (2014), Kuhn & Johnson (2013), Duda et al. (2000), Trevor Hastie (2009). In DSP, the most common technique to clean time series is linear digital filtering. A very important aspect of digital filtering is the delay introduced to the signal. Ignoring this delay can lead to incorrect conclusions being drown in case of, for example, on board monitoring systems. A delay can also affect the reliability of on-board monitoring systems. The consequences of the delay which comes from DAS or from digital filtering on the conclusions and predictive models were described and presented in Swider & Pedersen (2017). Nowadays this topic is even more important while IoT era and multi-sensor signal analysis. In that case time delay from network, processors, calculations or preprocessing can have serious consequences on the signals relations form various subsystems. It is very important to secure the synchronisation of the subsystems and provide delayless preprocessing which will not obscure the associations between signals. Additionally, signals can have different noise levels, which motivate use of different filters. Knowing that, it is necessary to look for filtering methods, which do not introduce a delay, or compensate for it.

In this paper we present a comparison and application of existing digital filtering methods which have practical applications. We describe Discrete Fourier Transformation (DFT) filtering as a tool for efficient big data preprocessing, which does not introduce a time delay, and which has low numerical cost. We show that the use of DFT filtering does not require specialist knowledge of filter design, what is beneficial for practitioners from various disciplines. We emphasise limitations of classical digital filtering methods and show that the delay introduced by classical digital filtering has consequences for big data analysis - this is added value for industrial data scientists. Additionally, we compare the performance of the DFT approach with the standard time domain filtering to motivate the usage of the DFT method. We document and exemplify our results based on real signals from marine on-board systems from the offshore vessel.

II. Delays introduced by digital filtering

In this subsection we present properties of digital filters. We describe the properties of the filter amplitude and phase responses, which are very often neglected by non-experts applying digital filtering. The phase response has properties which determine the proper filter choice and should not be ignored. The phase response of digital filters is significant especially for multi-sensor signal processing because it is a cause of preprocessing delay. For further discussion we assume analysis of big data in a from of digital signals, which are sampled version of analogue signals i.e.:

$$x[n] = x(t)|_{t=n/F_s} \quad n = 0, 1, \ldots \tag{1}$$

where n is the number of samples, F_s is the sampling rate which fulfils the Nyquist theorem Mitra (2010).

As described in the previous section most disturbances considered in the literature, are additive like Additive White Gaussian Noise (AWGN) and can be efficiently removed by linear filters. This is why we focus in this paper on this type of filter. The frequency response of a digital filter is very often expressed by the formula:

$$H(f) = |H(f)|e^{j\theta(f)} \tag{2}$$

where $|H(f)|$ is the amplitude response and $\theta(f)$ is the phase response. The amplitude response of filters is very well known and it is used for classification of filters in frequency domain, e.g. low-pass, high-pass filter.

Unfortunately, the phase response and its effects are often ignored. Phase response can be interpreted as follows:

1) If $\theta(f) = 0$, then the digital filter does not delay the input signal, ($n_0 = 0$)
2) If $\theta(f)$ is a linear function of f, then the digital filter delays the input signal, but without disturbances, and the delay is constant, ($n_0 =$const)
3) If $\theta(f)$ is a nonlinear function, the digital filter introduce phase distortion - it distorts the time relation between single frequency components from the input signal, ($n_0 = n_0(f)$) Oppenheim & Schafer (1975).

One straightforward way to analyse the properties of the filter phase response $\theta(f)$, is the phase delay, defined as:

$$\tau_p(f) = -\frac{\theta(f)}{2\pi f} \tag{3}$$

and group delay

$$\tau_g(f) = -\frac{d\theta(f)}{2\pi df} \tag{4}$$

The phase delay can be interpreted as the time delay of each single complex sinusoidal component of the input signal $x[n]$. The group delay highlights the deviations from linearity, which comes from the basic property of the differentiation. The physical interpretation of the group delay is difficult, however it can be interpreted as the time delay of the signal envelope for the signal with amplitude modulation Mitra (2010), Oppenheim & Schafer (1975). The interpretation of the phase and group delay is presented in Figure 2.

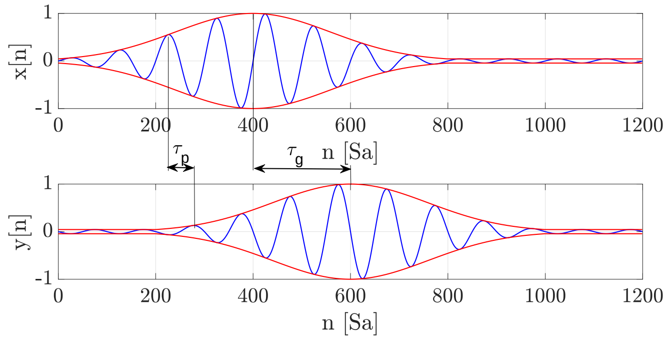


Fig. 2: Interpretation of the group delay Mitra (2010), where the $x[n]$ - the input signal and $y[n]$ - the output signal, red - signal envelope, blue - waveform of the signal, τ_p - phase delay, τ_g - group delay.

Based on the signal with amplitude modulation (variation of the amplitude), as can be seen from Figure 2, the carrier (blue) at the output $y[n]$ is delayed compared to the input $x[n]$ by the phase delay. The envelope (red) of the output signal is delayed compared to the envelope of the input signal by the group delay. These properties of the filter phase response are crucial in data analysis algorithms, where the time relation between signals is an important aspect.

Unfortunately, very often the problem of synchronisation and processing delay is omitted in the literature from data analysis e.g. Trevor Hastie (2009), Kuhn & Johnson (2013). Unsynchronised data can result in incorrect conclusions being drown at the end of the analysis and distorted correlations between signals. The delay effect can have a negative influence on machine learning algorithms or statistical modelling Swider & Pedersen (2017).

In this subsection we have shown that classical digital filtering introduces a delay to the output signal. Very often in practical solutions the phase response of the filters, which is causing the delay, is neglected. However, in big data preprocessing it will have a crucial impact on the time relation between signals. The description and importance of the delay from digital filtering on the data-driven models will be presented in the next section.

III. Influence of the delay from digital filtering on the quality of data-driven models

In this section we present the influence of the delay from digital filtering on the quality of data-driven models by analysing a simple example. The example of linear prediction is studied to show the negative influence of the delay from filtering. To analyse our model we assume the data collection system gathers p different signals $X_1, X_2, ..., X_p$, see Figure 3. In Machine Learning these are called predictors (independent variables, features or variables) and the signal Y is the output of the prediction model (the response or dependent variable).

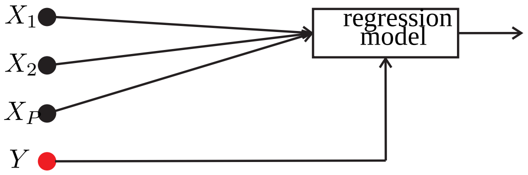


Fig. 3: Data analysis block diagram

The aim of the prediction (regression) model is to estimate the output signal Y based on the formula Kay (2005):

$$\hat{Y} = \sum_{i=1}^{P} a_i X_i \tag{5}$$

where $\hat{Y}$ is the estimate of response Y, a_i are linear predictor coefficients which are to be determined. The optimal coefficients are chosen to minimise the mean square error (MSE), given by:

$$\text{MSE} := E[(Y - \hat{Y})^2] \tag{6}$$

It can be shown that the solution of this equation is given by Kay (2005), Theodoridis & Koutroumbas (2008):

$$\begin{bmatrix} E[X_1X_1] & \dots & E[X_1X_p] \\ E[X_2X_1] & \dots & E[X_2Xp] \\ \vdots & \vdots & \vdots \\ E[X_pX_1] & \dots & E[X_pX_p] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} E[X_1Y] \\ E[X_2Y] \\ \vdots \\ E[X_pY] \end{bmatrix} \tag{7}$$

and the MSE error is as follows:

$$\text{MSE} = E[YY] - \begin{bmatrix} a_1 & a_2 & \dots & a_p \end{bmatrix} \begin{bmatrix} E[X_1Y] \\ E[X_2Y] \\ \dots \\ E[X_pY] \end{bmatrix} \tag{8}$$

The matrix given by (7) (the correlation matrix) contains information about the relation between measured signals. Based on the matrix we can conclude how the delay of single signal influence the prediction model. For example, if the first signal X_1 is delayed then the delay will influence the first row and column of the correlation matrix - the relation between X_1 and $X_2, ..., X_P$ and Y will be distorted. In case of the delay introduced by digital filtering in many signals, the final output of the prediction model will not be consistent. The statistical inferences based on such a model will be biased by high error Trevor Hastie (2009).

To visualise the achieved results we analyse the model with one variable X_1. An example from Swider & Pedersen (2017), shows the signal of voltage $Y = \sin(2\pi fn)$ and current $X_1 = \sin(2\pi fn)$ registered on the resistor with the resistance $R = 1\Omega$, see Figure 4.

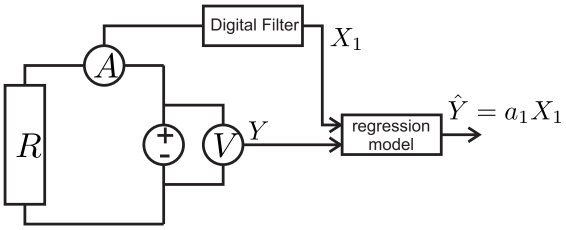


Fig. 4: Conceptual example of voltage (Y) and current (X1) registration.

From Ohm's law, the relation between current and voltage is given as:

$$Y = RX_1|_{R=1} = X_1 \tag{9}$$

If we assume that the current signal only is filtered by a digital filter with a phase delay of n_0, it can be expressed by the formula:

$$X_1 = \sin(2\pi f(n - n_0)). \tag{10}$$

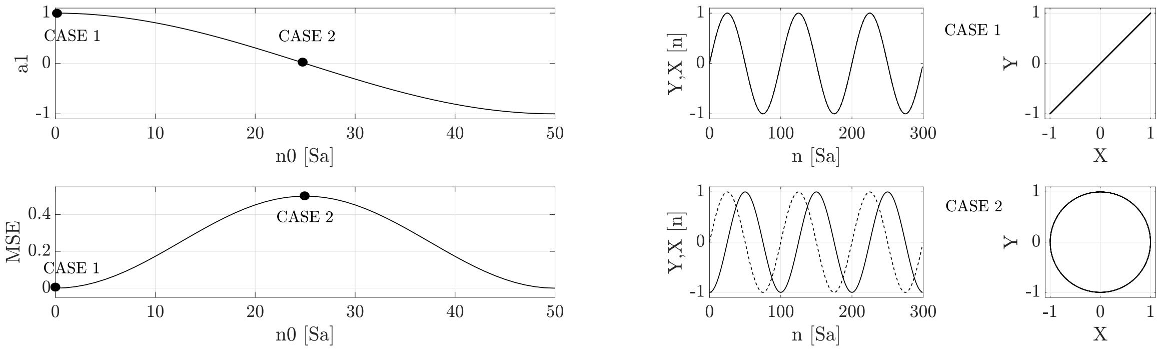
If we would like to calculate the regression coefficient and the MSE values we apply Equations (7) and (8) to achieve:

$$\begin{cases} a_1 = \cos(2\pi fn_0) \\ \hat{Y} = a_1X_1 = \cos(2\pi fn_0)\sin(2\pi f(n - n_0)) \\ \text{MSE} = 0.5(1 - \cos^2(2\pi fn_0)) \end{cases} \tag{11}$$

In Figure 5a, we show the regression coefficient and the MSE as a function of n_0, which is the introduced delay. We can see that if the delay is $n_0 = 0$, CASE 1, then $a_1 = 1, Y = X_1$ and the result is according to the physical law, see upper plots in Figure 5b. For CASE 2 when the delay is present in the signal $n_0 = 1/(4f)$, the regression model shows nonlinear relation between the voltage and current, $a_1 = 0, Y = 0$, which is contradictory to Ohm's law, see the lower plots in Figure 5b.

This simple example shows the importance of the delay introduced by filters and how the delay can completely change the result. The delay influences the quality of the prediction and statistical inference so the delay from filtering should be compensated. Unfortunately in the literature from big data and Machine Learning this aspect is not well described, e.g. Kuhn & Johnson (2013), Bishop (2011).

In order to illustrate the influence of the delay from filtering on the relationship between signals in data analysis from many sensors, we show the scatter plots of two signals from marine on-board monitoring systems, see Figure 6. One signal



(a)

(b)

Fig. 5: a) Waveforms of a coefficient a_1, given by ($a_1 = \cos(2\pi f n_0)$) and MSE given by (11) as a function of the time delay n_0. Black points depict CASE 1 and CASE 2. The frequency of the signal $f = 0.01$Hz. b) Waveforms and scatter plots of current X (dotted line) and voltage Y (dashed line) for two cases: upper subplots CASE 1: $n_0 = 0$, lower subplots CASE 2: $n_0 = 1/(4f)$.

(y-axis) is the main propulsion power in MW measured on the shaft, the second signal (x-axis) is the main propulsion power in MW (Megawatts), calculated in the drive. We would like to verify the linear relationship between the power calculated and measured, which is useful for confirmation of the calculation methodology in the drive system. In Figure 6.a we see the raw measurements, with high variance. The noise present in the measurements obscures the linear relationship between power calculated and measured. In this case, and also based on the time series and the spectrum analysis, digital filtering is recommended to remove the noise and emphasize the real relation between signals. More details is given in section VI. Figure 6.b shows the filtered measurements with distorted time relation (the delay between signals is equal to 150s). Despite the filtering, the linear relation is distorted, which is the result of the delay introduced by different length of the filter. The different filter length was applied to depict the influence of disparate filters in case of signals form various sources. Figure 6.c shows the filtered signals with compensated delay, which is desirable to confirm the linear relationship. Based on this example we can see that only delayless filtering (or delay compensation) can provide the proper results necessary for further analysis. Based on Figure 6.b we are not able to determine whether the measured and calculated signals are consistent or not. This is proven based on the results presented in the Table I, which show the MSE values and the regression equation. The linear relation between calculated and measured power can only be confirmed with the smallest MSE value, which corresponds to filtering without destroying the time relation between analysed signals.

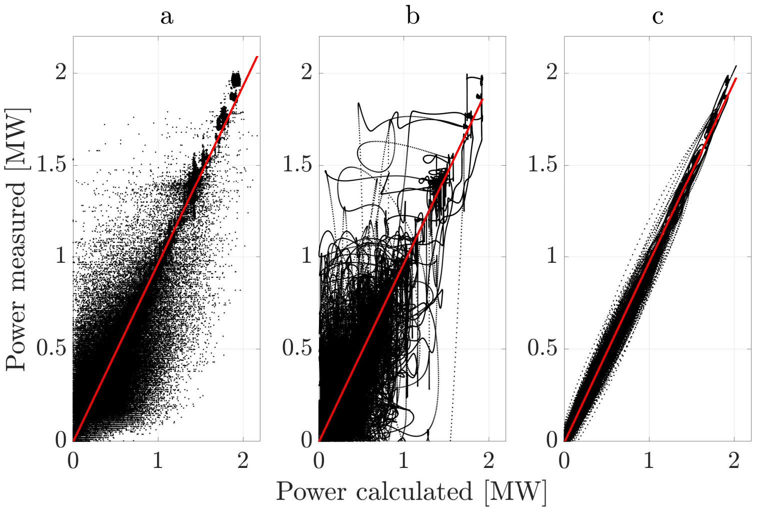


Fig. 6: Example of the relation between two signals: the power calculated in the drive and the power measured on the shaft. a - the raw measurements, b - filtered measurements with different delays, c - filtered measurements without delay

In this section we have presented two simple examples (synthetic and real) where we tried to find the relation for a single input-output model. We have to remember that during big data analysis, when we have multiple input models, the delay introduced by digital filtering will destroy the time relation between many signals, and can be difficult to detect. Additionally,

TABLE I: The MSE and regression equation for analysed cases a, b and c

Method	MSE	Regression
Raw measurements	3294	$y = 0.97x - 1.59$
Filtered measurements	5505	$y = 0.97x - 2.01$
Filtered measurements without delay	400	$y = 0.97x - 6.38$

the delay has the impact on the quality of the prediction model and statistical inferences. This conclusion holds for all data-driven models.

By presenting the importance of the delay introduced by filtering we would like to focus in this paper on the review of filtering methods and show a method which is straightforward to implement, is delayless and additionally has low computational cost (which makes it a good choice for preprocessing time series - big data e.g. from marine systems).

In the next section we will describe more advanced methods of digital filtering which do not introduce a delay and can be applied for preprocessing of archival sequence data.

IV. Digital filtering without delay

In the previous section, it was shown that classical filtering, which can implemented on-line and off-line, always introduces a delay. If the filtering of sequence data is required in off-line analysis (e.g. noise filtering) then filtering algorithms which do not introduce the delay should be used and more sophisticated methods are needed for filtering of archival sequence data.

Generally, digital filtering can be applied in the time domain or in the frequency domain (spectrum domain, DFT domain, - Discrete Fourier Transform domain). In the following sections we show the basics of digital filtering in the time domain Lyons (2010) and then we focus on the DFT domain Rao & Yip (2000), Proakis & Nikias (1992) because more delayless methods are available in this domain.

A. Digital filtering in time domain

One solution, which can be applied in the time domain is zero-phase filtering Lyons (2010). This type of filtering in the time domain can be performed as shown in Figure 7:

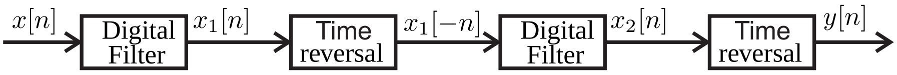


Fig. 7: Zero phase filtering Lyons (2010), $x[n]$ - the input signal, $x_1[n]$ - signal at the output of the filter, $x_1[-n]$ - flipped signal, $x_2[n]$ - signal at the output of the filter, $y[n]$ - final output signal

The same filter is applied twice with a time reversal between the two filters. Time reversal step is a left-right flipping of a time domain sequence. The output signal $y[n]$ is a filtered and delayless version of the signal $x[n]$. Such an approach can be applied while preprocessing archival sequence data, since the algorithm requires flipping the signal in the time domain. It is straightforward to show that the relation between the spectrum of the input and output signal is given by the relation:

$$Y(f) = |H(f)|^2 X(f) \tag{12}$$

where $X(f)$ is Discrete Time Fourier Transform (DTFT) of input signal $x[n]$, $H(f)$ is the frequency response of digital filter and $Y(f)$ is DTFT of output signal $y[n]$, defined as follows:

$$Y(f) = \sum_{n=-\infty}^{n=+\infty} y[n]e^{-j2\pi fn}; \qquad n = +\infty, \ldots, -1, 0, 1, \ldots, +\infty \tag{13}$$

Therefore the algorithm shown in Figure 7 implements a zero-phase filter with a frequency response $|H(f)|^2$.

The main disadvantages of such an approach are:

- in the most common implementation it has high numerical costs, however this can be reduced (for the FIR filters) by fast convolution Mitra (2010);
- based on relation (12) the algorithm works properly only if the ideal frequency response is $H(f) \in R$ and $H(f) = H(-f)$. This means that the implementation of some filters is impossible e.g differential filter, integral filter etc.
- the complex design of the digital filter, which requires specialist knowledge. Lack of expertise can result in some important details (e.g convolution properties or ripples in passband) being omitted. These details can influence the result of the filtering.

The above disadvantages are not present in filtering in the spectrum domain DFT, which will be described in the next section. The main advantage of the DFT domain is that there are no constraints about the frequency response of the filter, so the differential filter, integral filter can be implemented. DFT is very often referred as the FFT, however it is worth to remember that FFT is a fast algorithm for computing Discrete Fourier Transform (DFT). The idea behind the FFT is to decompose the N-point DFT computation into computation of smaller size DFT and to take advantage of the periodicity and symmetry properties of DFT. In literature DFT and FFT are used interchangeably Mitra (2010).

B. Filtering in the frequency domain (via DFT)

The spectrum of the output signal is equal to the product of the complex frequency response of the filter and the spectrum of the input signal. This can be described by the following relation:

$$Y(f) = H(f)X(f) \tag{14}$$

Then relation (14) can be written in the following way:

$$Y[k] = H[k]X[k] \tag{15}$$

where $H[k]$ is the N-points frequency response of the filter, $X[k]$ is the N-point DFT of the signal on the output of the filter and $Y[k]$ is the N-point DFT of the signal $y[n]$, given by formula:

$$Y[k] = Y(f)\Big|_{f=\frac{k}{N}} = \sum_{n=0}^{N-1} y[n]e^{-j\frac{2\pi k}{N}}; \tag{16}$$

where k for the N-points DFT transformations, are given by:

$$k = \begin{cases} -\frac{N}{2}, -\frac{N}{2}+1, ..., -1, 0, 1, ..., \frac{N}{2}-1 & \text{for even } N; \\ -\frac{N-1}{2}, ..., -1, 0, 1, ..., \frac{N-1}{2} & \text{for odd } N. \end{cases} \tag{17}$$

The main advantage of this approach is that the digital filter is directly implemented in the frequency domain, i.e.:

$$H[k] = H(f)\Big|_{f=\frac{k}{N}} \tag{18}$$

which is a simplification of the filtering algorithm.

The filtering algorithm in the DFT domain can be described by the following relation:

$$y[n] = \text{IDFT}\{\text{DFT}\{x[n]\}H[k]\} \tag{19}$$

where the Inverse Discrete Fourier Transforms (IDFT) is calculated by:

$$y[n] = \frac{1}{N}\sum_{\forall k} Y[k]e^{j\frac{2\pi nk}{N}} \tag{20}$$

which is depicted in Figure 8 and is called basic DFT filtering.

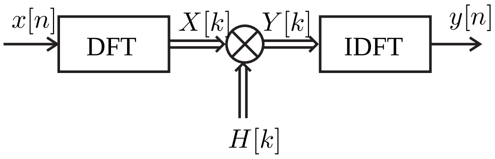


Fig. 8: Basic DFT filtering, $x[n]$ - the input signal, $X[k]$ - the DFT of the $x[n]$, $H[k]$ - the frequency response of the filter, $Y[k]$= the product of the $X[k]$ and $H[k]$, $y[n]$-the output signal

The quality of the filtering based on relation (19) is very often poor. It results from usage of the circular convolution instead of the linear convolution Mitra (2010). In practice we use two approaches which improve the quality of the filtering, which will be described below.

1) Filtering with zeropadding: In this algorithm, the preprocessing of the signal $x[n]$ is done based on extending the signal with Z values equal to 0, i.e.

$$x_1[n] = \begin{cases} x[n] & \text{for } n = 0, \ldots, N-1; \\ 0 & \text{for } n = N, \ldots, N+Z. \end{cases} \tag{21}$$

The typical size of the zeropadding is equal to N. This means that the length of the signal $x_1[n]$ is equal to $2N$. The filtering with zeropadding algorithm is represented by the block diagram in Figure 9.

For such a prepared sequence $x_1[n]$ we apply the relation (19), which ensures a $y_1[n]$ sequence, as follows:

$$y_1[n] = \text{IDFT}\{\text{DFT}\{x_1[n]\}H[k]\} \tag{22}$$

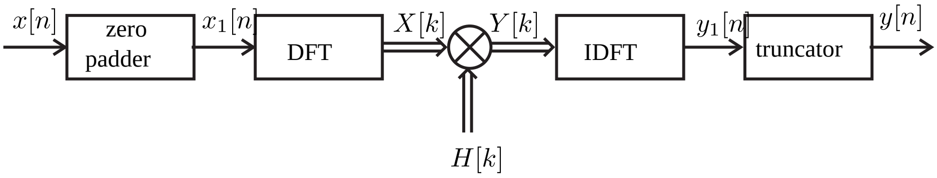


Fig. 9: Filtering with zeropadding, $x[n]$- the input signal, $x_1[n]$- the input signal after zeropadding, $X[k]$- the DFT of the $x_1[n]$, $H[k]$- the frequency response of the filter, $Y[k]$- the product of the $X[k]$ and $H[k]$, $y_1[n]$- IDFT from the $Y[k]$, $y[n]$- the output signal

The sequence $y_1[n]$ has the length $N + Z$, so it is necessary to remove the last value of Z:

$$y[n] = y_1[n] \qquad 0 \le n \le N - 1 \tag{23}$$

Modification of the sequence, which is given by (21), allows us to reduce the aliasing in the time domain, which is the main reason for significant distortions in DFT algorithms. Extension of the input signal $x[n]$ by zeropadding reduces the negative effects of the circular convolution.

2) Filtering with the even symmetric extension: In this algorithm data flipping is applied Smith & Eddins (1987), Kiya et al. (1994), i.e. signal $x_2[n]$ consists of the signal $x[n]$ and its mirror reflection, which can be described by the formula:

$$x_2[n] = \begin{cases} x[n] & \text{for } n = 0, ..., N-1; \\ x[2N-1-n] & \text{for } n = N, ..., 2N-1. \end{cases} \tag{24}$$

Filtering with the symmetric extension is represented by the block diagram in Figure 10:

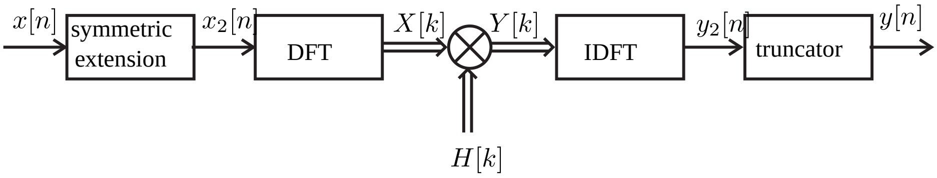


Fig. 10: The filtering with the symmetric extension, $x[n]$- the input signal, $x_2[n]$- the input signal after symmetric extension, $X[k]$- the DFT of the $x[n]$, $H[k]$- the frequency response of the filter, $Y[k]$- the product of the $X[k]$ and $H[k]$, $y_2[n]$- IDFT from the $Y[k]$, $y[n]$- the output signal

For a preprocessed signal we apply the algorithm (19), based on which we achieve the sequence $y_2[n]$ as following:

$$y_2[n] = \text{IDFT}\{\text{DFT}\{x_2[n]\}H[k]\} \tag{25}$$

In the last stage, we choose the N-initial values, i.e.

$$y[n] = y_2[n] \qquad n = 0, ..., N-1 \tag{26}$$

Very often the filtering algorithm with the symmetric extension is called the Cheh-Pan modification Pan (2001), Pan (1996). It should be noted that publications on this topic were available before the Cheh-Pan articles Smith & Eddins (1987), Smith & Eddins (1990).

V. Simulation experiments

In this section we describe the properties and performance of digital filtering algorithms without the delay based on synthetic signals. The benefits of such an approach include the possibility to evaluate the output of the algorithms based on a priori knowledge. Additionally, we can check the performance and the quality of solutions applying the filtering error, given by formula (27):

$$e[n] := y[n] - y_{ideal}[n]; \qquad n = 0, ..., N-1 \tag{27}$$

where $e[n]$ is the filtering error, $y[n]$ is the output signal, $y_{ideal}[n]$ is the ideal output signal.

The performance of the filtering can be described also by the Mean Square Error (MSE), given by the relation:

$$\text{MSE} := \frac{1}{N}\sum_{n=0}^{N-1}(y[n] - y_{ideal}[n])^2 = \frac{1}{N}\sum_{n=0}^{N-1} e^2[n] \tag{28}$$

Analysis of filtering based on simulations and synthetic examples allow to measure the quality of the filtering, because in the analytical way we can calculate the formula for the output signal of an ideal filter. Additionally, such experiment allow to simulate the properties of the input signal which are significant for quality of the filtering.

Comparison of delayless filtering algorithms is based on low-pass filtering- in this example we assume that the signal contains two frequency-separated components. The aim of the filter is to remove one of the components. This problem often appears in practice where the useful part of the signal energy spectrum is located in low frequencies (we would like to extract this part). The high frequency components are noise.

Low-pass filtering appears often in practice. Preprocessing of sequence data for Machine Learning algorithms to remove the noise from measurements, improves their performance. Using the non-filtered (raw) signal on the input of the machine learning algorithm results in poor quality output. High noise levels reduce the amount of information from measurements Duda et al. (2000). In the experiment we assumed that the signal is the sum of two components, given by the formula:

$$x[n] = \underbrace{\sin(2\pi 0.02n) + 2 + \frac{2n}{N}}_{Desired\ Signal} + 0.2\sin(2\pi 0.1\sqrt{2}n); \quad n = 0, ..., N-1 \tag{29}$$

where desired signal contains sine waveform, trend ($2n/N$) and DC value. Noise component is a sine waveform with frequency $0.1\sqrt{2}$. We add trend to signal $x[n]$ because it has a significant meaning for assessment of the quality of filtering. This is described in detail below.

The aim of the filtering is to remove the component above frequency $f = 0.07$. We can define low-pas filtering based on the frequency response of the ideal filter and the following relation:

$$H_{LP}(f) = \begin{cases} 1 & \text{for } -f_c \leq f \leq f_c; \\ 0 & \text{otherwise.} \end{cases} \tag{30}$$

This filter can be implemented as FIR, with an impulse response given by:

$$h_{LP}[n] = \begin{cases} 2f_c \text{sinc}(2f_c(n - \frac{M-1}{2})) & \text{for } n = 0, \ldots, M-1; \\ 0 & \text{otherwise.} \end{cases} \tag{31}$$

where M is the length of impulse response. In order to design the filter we should find the sufficient value M and the window function in such a way that the frequency response will fulfil the design requirements. More about digital filter design can be find in Lyons (2010), Mitra (2010).

For FIR filter the output signal $y[n]$ is given by formula:

$$y[n] = h_{LP}[0]x[n] + h_{LP}[1]x[n-1] + \ldots + h_{LP}[M-1]x[n-(M-1)] = \sum_{k=0}^{M-1} h_{LP}[k]x[n-k] \tag{32}$$

It means that this is a linear convolution between input signal $x[n]$ and impulse response.

Another way to implement the filtering is to take advantage of filtering in DFT domain. To achieve this we need to design only the frequency response of the filter, which can be done by the following formula:

$$H_{LP}[k] = H_{LP}(f)\Big|_{f=\frac{k}{N}} = \begin{cases} 1 & \text{for } -f_c N \leq k \leq f_c N; \\ 0 & \text{otherwise.} \end{cases} \tag{33}$$

From (14) and (33), we see that the implementation of the filter in the frequency domain DFT is simpler than in the time domain (classical approach). This property will be used later in the paper to motivate usage of this solution.

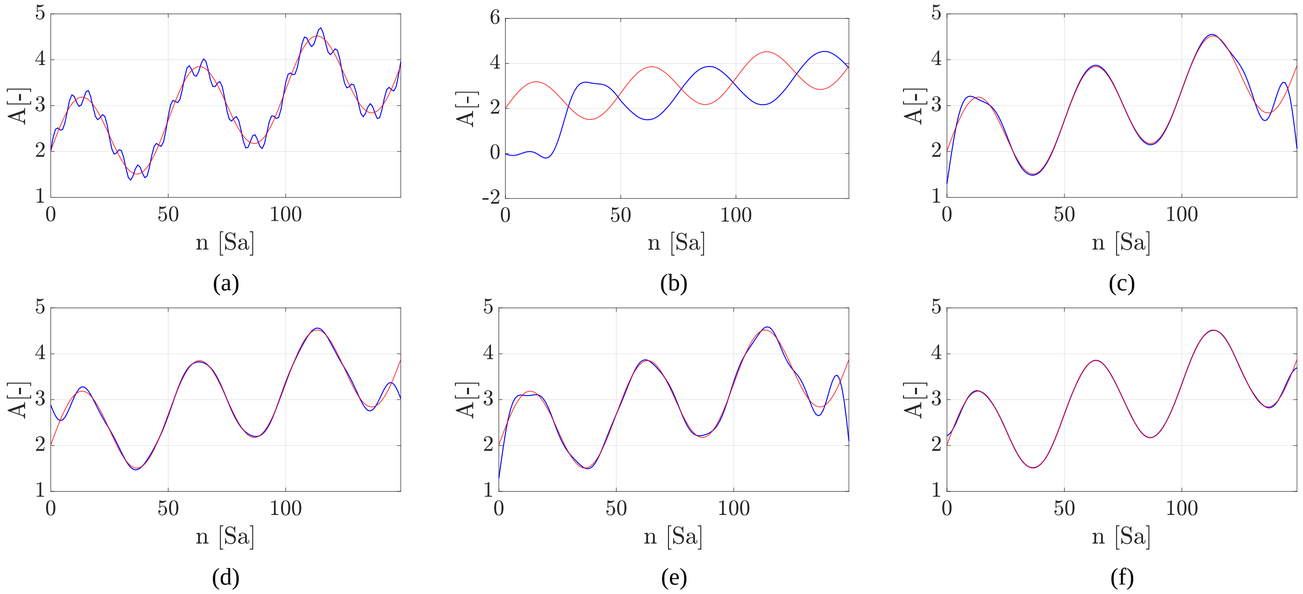
In Figure 11 we show results from the following experiment. Subplots in Figure 11 show (a) input signal $x[n]$ (blue) and the desired ideal filtered signal (red), (b-f) the output of the filtering (blue) and the desired signal (red). As we can see the output of the applied filter is delayed to the desired output signal for classical filtering. The reason for the delay is the implementation of the step given by the relation (32). The delay is equal to $(M-1)/2 = 25$. Additionally, we can observe high distortion at the beginning of the signal for almost all algorithms. The smallest distortion has the filtering with symmetrical extension, what we explain below.

Transients which appear at the beginning and at the end of the filtered signals by DFT algorithm in Figure 11 are highlighted in Figure 12 and are primarily influenced by properties of the DFT. The main assumption in DFT is signal periodicity, i.e.:

$$x[n] = x[n+N] \tag{34}$$

where N is the primary period. If in the signal $x[n]$ appear a significant difference between the last value of the first period of signal, $x[N-1]$, and the first value of the next period, $x[N] = x[0]$ (according the formula (34)), then transients will appear at the beginning and at the end of the filtered signal $y[n]$ as a result of filtering Pan (2001). The common solution to eliminate transients is removal of linear trend from the filtered signal $x[n]$,i.e.:

$$x'[n] = x[n] - \underbrace{\frac{x[N-1] - x[0]}{N} n}_{x_t[n]}; \quad n = 0, \ldots, N-1 \tag{35}$$



(a) (b) (c) (d) (e) (f)

Fig. 11: Experiment with synthetic signals: a) input signal (blue) (29) and desired signal (red), b) classical filtering: output signal (blue), desired signal (red) c) zero-phase filtering: output signal (blue), desired signal (red) d) basic DFT filtering: output signal (blue), desired signal (red) e) the filtering with zeropadding: output signal (blue), desired signal (red) f) the filtering with symmetrical extension: output signal (blue), desired signal (red).

where $x'[n]$ is a signal without trend and $x_t[n]$ is the linear trend. As a result of detrending we achieve:

$$x'[0] = x'[N-1] \tag{36}$$

This operation taper the samples towards same values at the end points, and so there is no discontinuity with a hypothetical next period. It is worth to notice that filtering with symmetrical extension causes exactly that $x_2[0] = x_2[2N-1]$, i.e. the first and the last sample of the signal $x_2[n]$ are the same, Pan (2001) and results in damped transients. In the analysed above example, we aimed to show the influence of the trend in the signal on the quality of filtering.

Transients which appear from classical and zero-phase filtering were discussed in Gustafsson (1996). The cause of transients are non-zero level of first samples of the signal and DC value P. Sadovsky (2000). The solution of this problem is calculation of initial buffer values in applied filters, which are dependent from filtered signal $x[n]$ and filter impulse response $h[n]$, more can be find in Gustafsson (1996), P. Sadovsky (2000). Based on this solution the commercial solutions (e.g. in MATLAB) of the zero-phase filtering is improved.

For the performance evaluation of the applied algorithms (the classical filtering is omitted, due to its poor properties), we use the formula (27) and results are shown in Figure 12.

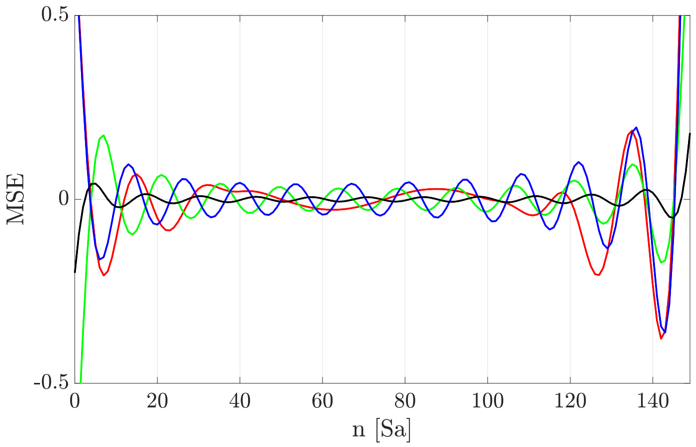


Fig. 12: Errors’ waveforms of filtering with zero-phase (red), basic DFT (green), zero-padding (blue), symmetric extension (black)

TABLE II: The MSE from the first experiment for different types of filtering

Method	MSE
Zero-phase	0.0494
Basic DFT	0.0190
Zero-padding	0.0452
Symmetric extension	0.00072

The highest error is at the beginning and end of the signal for all algorithms as the result of transients. The algorithm with the smallest error is filtering with the symmetric extension. Filtering with symmetric extension has the best properties and zero-phase filtering the worst. Conclusions are confirmed by the MSE values presented in the Table II, where the highest MSE is for the zero-phase filtering and the lowest for the symmetric extension. To improve the quality of the filtering in the DFT domain detrending (35) can be applied. This approach will improve the quality of the DFT spectrum, more details can be found in Oppenheim & Schafer (1975).

VI. Experiment based on real measurements from marine systems

In this section we present real examples where digital filtering is needed to remove noise from two low frequency signals. We consider real signals: the vessel roll and the main propulsion power signal, both sampled with sampling frequency $F_s = 1Hz$. There are many examples like this with low frequency measurements and much more with high frequency measurements ($F_s >> 1Hz$), where selective filtering is a common way to extract specific frequency components. The first example depicts the digital filtering of the vessel roll signal. The vessel roll signal is registered by the MRU (Motion Reference Unit) on board the vessel. The vessel roll describes the motions of the vessel around the longitudinal axis. This signal contains information about the permanent vessel heel (slow variable component or DC component), however the signal also contains movement of the vessel on the sea waves (fast variable component, AC component). The vessel heel is needed to describe the loading conditions and safe ship operations. To be able to extract vessel heel from the roll signal we have to remove the higher frequencies which come from the waves, this can be achieved by low-pass digital filtering. Figure 13 shows the vessel roll signal which has the length 180[Sa] ($1Sa \equiv 1s$). To extract the slow variable component we applied low-pass filtering with the cut-off frequency $f_c = 0.05$ [1/Sa]($1/Sa \equiv 1Hz$). The previously described algorithms were applied to this problem. The length of the classical FIR filter is 101[Sa]. As shown in Figure 13 we clearly see the delay from classical filtering. The algorithms which perform without delay are the filtering in the DFT domain and the zero-phase filtering. Beside the boundaries, all delayless algorithms perform similarly, which was expected based on the results from the experiment with synthetic signals. The behaviour of the algorithms at the edges show small discrepancies, and in this case they can be neglected. The small error from the DFT filtering is due to the first and last values in the signal which are similar ($x[0] \approx x[N-1] \approx 0$), which we can interpret as a single period of the periodic signal. Beside that, there is no trend present in this signal.

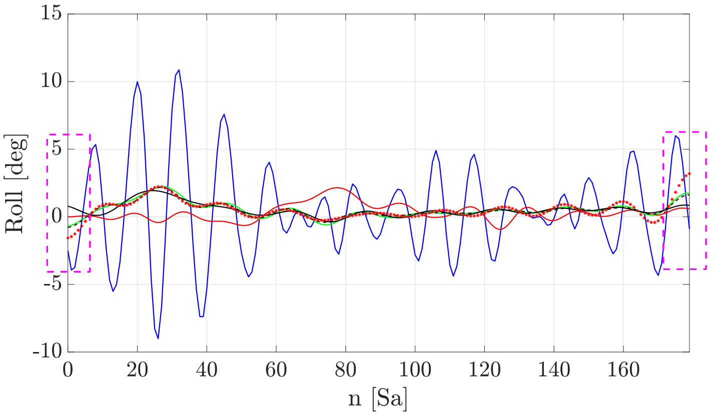


Fig. 13: Waveforms of the raw vessel roll signal (blue) and filtered signal: by the classical filtering (red), by the zero-phase (green), DFT (black), zeropadding (black dashed), symmetric extension (red star)

For comparison of quality of filtering we use SNR, which is defined in following way:

$$SNR = 10\log_{10}\left(\frac{\text{power of signal}}{\text{power of noise}}\right) \tag{37}$$

TABLE III: Comparison of different filtering methods based on SNR

Method	SNR of Roll	SNR of Power Calculated	SNR of Power Measured
Before filtering	-16.8	3.6	11.2
DFT	14.3	11.9	12.7
Zero-padding	16.9	11.9	12.6
Symmetric extension	19.5	12	12.7
Classical filtering	15.5	13	13.2
Zero-phase	20	13.5	13.6

For calculation of the power of the signal and power of the noise we use properties of power spectral density (PSD). PSD quantifies the distribution of power with frequency and it is defined in the following way Kay (2013):

$$P_y(f) = \lim_{M\to\infty} \frac{1}{2M+1} E\left[\left|\sum_{n=-M}^{n=M} y[n]e^{-j2\pi fn}\right|^2\right] \tag{38}$$

where E is the mean value, M is the number of samples, f denotes discrete-time frequency, which is assumed to be in the range $-0.5 \leq f < 0.5$.

In our case, power of the signal is the power of the signal in the frequency band $[0, f_c]$ (desired DC component is in this interval), however power of the noise is the power of the signal in frequency band $[f_c, 0.5]$. We apply following properties of the PSD, what allow to estimate power of signal in selected band, i.e.:

$$\text{Average power in}[f_1, f_2] = 2\int_{f_1}^{f_2} P_y(f)df \tag{39}$$

Power of the signal and noise can be calculated in following way:

$$\text{power of signal} = \int_0^{f_c} P_y(f)df \qquad \text{power of noise} = \int_{f_c}^{0.5} P_y(f)df \tag{40}$$

When we substitute formula (40) to (37), we achieve:

$$SNR = 10\log_{10}\left(\frac{\int_0^{f_c} P_y(f)df}{\int_{f_c}^{0.5} P_y(f)df}\right) \tag{41}$$

In practice the PSD have to be estimated and we decided to use the Welch estimator as it is recommended for this purpose Kay (1999) and Hayes (1996).

In Table III we show the SNR calculated by (41) for the roll signal and power measured and calculated described further in the paper.

In Table III we show the SNR for the roll signal before filtering ($SNR = -16.8dB$) and after filtering to evaluate different filtering method's performance. A negative value of the SNR of signal $x[n]$ can be explained by slow variable component (DC) which have a small amplitude comparing with the AC component, see Figure 13. As a result of LP filtering the AC component is filtered and according to Table III, the highest SNR we receive for zero-phase and symmetric extension filtering.

We can conclude that the DFT and zero-phase filtering are better algorithms than the classical filtering. To see better the differences between all delayless methods we present the second example.

The second example uses two signals: power calculated and measured. The objective is to verify the consistency of the calculated and measured power, see Figure 6. The measured power is the product of the torque and rpm measured by sensor installed on the shaft between the electric motor and the main propulsion thruster. The calculated power is the power calculated in the drive based on the internal motor model. The model use the motor nameplate values and internal measured current, modulated voltage and frequency to estimate the shaft power. There are many uncertainties in this calculation based on tolerances and other changing parameters. However, based on experience from tests, where the actual power was delivered on shaft and compared to drive calculation, we can rely on this signal with accuracy of few percentage of the maximal power. As we saw from the Figure 6.a, for small values of the power calculated and measured (in range 1-1000kW), there is a high variance. Therefore, in Figure 14 we have shown two cases: one when the power measured and calculated have low values (left subplot) and second when power signals have high values (right subplot). To investigate fluctuations and high variance of power signals we depicted their spectrum. In Figure 14.a and Figure 14.c we can see that for the power calculated in the drive (red) there are a lot of fluctuations form waves, what confirms the peak in the spectrum at the 0.18Hz. The peak from wave is not present for the measured power on the shaft (Figure 14.c - blue). We see also that the time series of power measured has lower variance than power calculated, so it is hard to compare both signals. Additionally, there are less variance for high power values (Figure 14.b). The difference between power calculated and measured is in range of 4-5$\%$ of the maximal power and can be the result of accuracy of calculations or the non-linear characteristic of the torque sensor. We can see that for high power values the noise looks like AWGN (Figure 14.d red). Based on this we concluded that to enable comparison of signals digital filtering is required to remove the noise and the wave component from measurements. To be more specific, to confirm the linear relation between the measured signal and the calculation procedure which is not obvious from the raw data, as depicted in Figure 6.a.

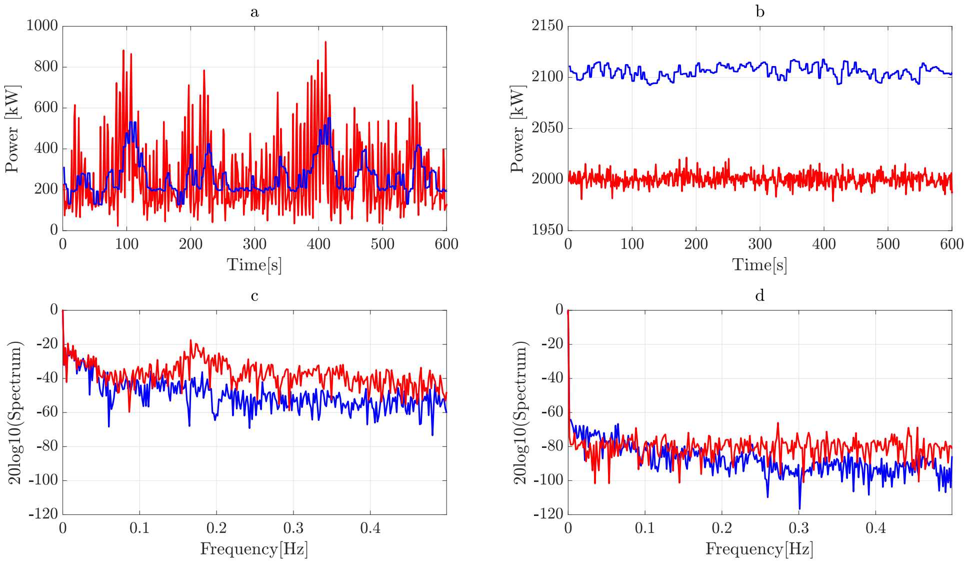


Fig. 14: Analysis of properties of the power signals in time domain (a,b) and in frequency domain (c,d) for two different power range. Power calculated in drive - red, power measured - blue.

In Figure 15, we show the results from filtering the measured and the calculated power by the methods described above. To extract the slow variable component we applied low-pass filtering with the cut-off frequency $f_c = 0.01$ [1/Sa]. The length of the classical filter is 101[Sa] and the delay from this type filtering is obvious. The delay is not present during the DFT and zero-phase filtering, however there are distortions at the signal boundaries. The worst results were obtained for the zero-padding and zero-phase filtering, which results from the constant component present in the signal (mean≈ 200). Despite the long length of the signal (600[Sa]∼ 10min) the transition in the zero-padding and zero-phase filtering at the boundaries is significant. The length of transitions states is equal approx 10% of the signal length and they should be omitted in further analysis, which is a limitation of those methods. In case of processing data blocks of short time intervals transition states are significant. The best results we received for the DFT and DFT with symmetric extension filtering, which comes from the properties of the DFT.

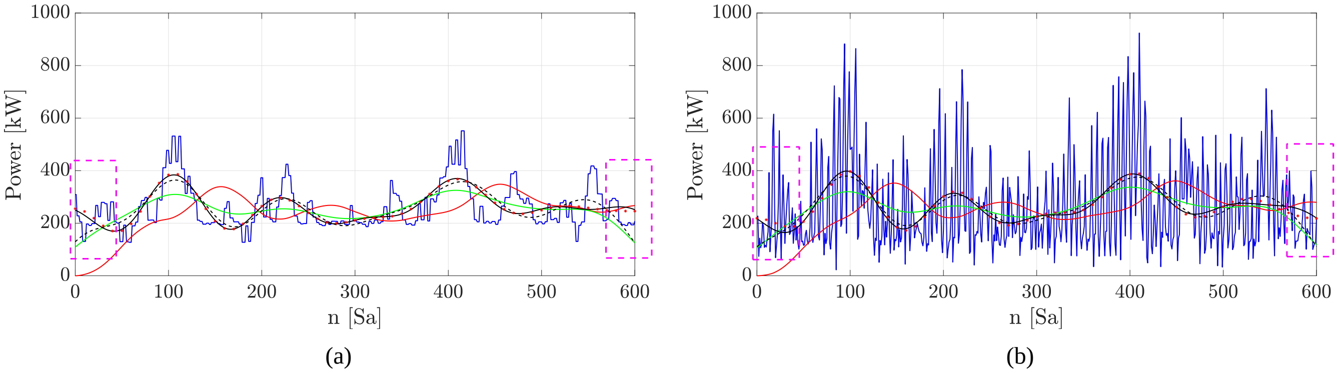


Fig. 15: Waveforms of the raw: a)power measured (blue) and b) power calculated(blue) and filtered signal: by the classical filtering (red), by the zero-phase(green), by the DFT (black), by the zero-padding (black dashed), by the symmetric extension (red star).

In Table III we showed the SNR for power measured and calculated. For power measured, the $SNR = 11.2dB$ before filtering and from analysis of the Figure 15a we see that the cause of distortions is low quality of the AC/DC converter and the AC component from ocean waves Fossen (2002). For power calculated the dominant distortion are ocean waves, which have larger amplitude than the power measured, what causes lower $SNR = 3.6dB$ than for power measured. We can see that the LP filtering improved the SNR significantly and the SNR for power calculated increased about 8dB however for the power measured the difference before and after filtering is not significant (the level of distortions is low). Based on the results shown

TABLE IV: Comparison of different filtering methods based on filtering time

Method	Filtering time of Roll	Filtering time of Power Calculated	Filtering time of Power Measured
DFT	$5{\cdot}10^{-5}$	$4{\cdot}10^{-5}$	$4{\cdot}10^{-5}$
Zero-padding	$7{\cdot}10^{-5}$	$6{\cdot}10^{-5}$	$6{\cdot}10^{-5}$
Symmetric extension	$6{\cdot}10^{-5}$	$5{\cdot}10^{-5}$	$5{\cdot}10^{-5}$
Classical filtering	0.001	$8.8{\cdot}10^{-4}$	$8.6{\cdot}10^{-4}$
Zero-phase	0.002	0.002	0.002

in Table III we see that the zero-phase filtering gave the highest SNR what is the result of (12) and Figure 7 where we see that the filtering is applied twice. In case of power measured and calculated the difference between filtering methods is not as significant as for the roll signal. In this case to evaluate the performance of filtering it would be beneficial to study the time of calculation for all these methods, what is done in the next section.

We can conclude that the DFT methods are delayless, so in the case of processing data from many sources, as in the case of big data, these methods are desirable for filtering. Additionally the DFT with symmetric extension has the best quality, which was shown both in the theoretical and real examples. This results from the fact that symmetrical extension enforces the periodicity without discontinuities. The good properties of the DFT will depend on the signal periodicity, so in some situations the results will be similar to the DFT with symmetric extension and in other situations they will be worse. In case of time series without a trend, all delayless algorithms will perform similarly. We would like to recommend the DFT as a tool for digital filtering for big data due to the robustness of this algorithm. Beside that, delayless differentiation, integration etc. can be applied only in the DFT domain (e.g. Hilbert Filter in MATLAB), which makes the DFT more appropriate than zero-phase filtering. Additionally, in the next section we will present the numerical complexity of the DFT method what makes it suitable for big data preprocessing.

VII. Numerical complexity

In the previous section we showed that filtering in DFT domain does not introduce delay, and its quality is therefore better compared to classical filtering solutions. An additional property of the DFT filtering is low computational cost in comparison to classical filtering. It results, from the numerical cost of the N-point DFT being of order N^2. If we apply the Fast Fourier Transformation, calculations will be the fastest for length of the signal equal $N = 2^m$, where $m \in \mathbb{N}$ - then the numerical cost is approximately equal to $0.5N \log_2 N$ Mitra (2010), Oppenheim & Schafer (1975). In Figure 16 we show the results from the experiment to compare the numerical costs for all described solutions. The experiment was carried out for the synthetic signal of length from $100Sa$ to $10^6 Sa$, which is relevant for a signal sampled with 1Hz during half a month. As we see from the Figure 16, the numerical cost is highest for filtering in direct time domain (implemented by (32)) and also increases with the length of the signal. We obtained lower computational costs for the other types of the filtering in the DFT domain, which was expected according to the DFT/FFT properties. The use of the recursive filter in the zero-phase filtering results in lower computational cost than non-recursive filters, however the design of the non-recursive filters is more complex and it is not possible to implement some types of filters.

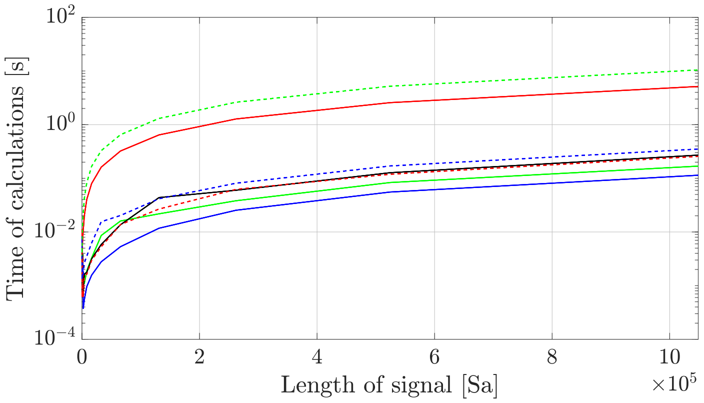


Fig. 16: Comparison of different filtering methods: the zero-phase filtering for direct time domain (dashed green) and fast convolution (dashed blue), the classical filtering for direct time domain (continues red) and fast convolution (continues green), the basic DFT (continuous blue), the symmetric extension (dashed red), zeropadding (continuous black)

Additionally, we evaluated the performance of filtering based on real examples presented in the previous section. Results are shown in Table IV. We see that the shortest computation time we achieved for the algorithm implemented in the DFT/FFT, i.e. for raw DFT and symmetric extension comparing with the classical filtering and the zero-phase filtering. To conclude, the recommendation is to use the DFT filtering for big data preprocessing.

VIII. Conclusions

Today, there is a focus on data analysis from many sensors, often referred to as big data in IoT era. In the case of sequence data, preprocessing is essential. One of the main algorithms for data preprocessing is digital filtering. In this paper we presented and compared existing algorithms of digital filtering from the perspective of the delay which is introduced by filtering. This is important during analysis data from many sources. The reason for the delay from digital filtering is the phase response of filters, which is often neglected by engineers. To avoid introducing distortions through digital filtering it is desirable to ensure a linear phase response, and this is possible only for non-recursive filters. Based on the prediction model we showed that the delay of some of the input signals influences the output of the prediction model, which was shown also based on the correlation matrix. This motivates the use of delayless digital filtering. In this paper we presented a comparison and application of existing digital filtering methods in the DFT domain based on real-life marine examples. The performance of the delayless algorithms was documented based on synthetic and real data and compared to classical filtering. Using synthetic signals we were able to measure the MSE of algorithms and we showed the limitations of zero-phase filtering. The best performance was achieved for the DFT filtering with symmetric extension. The solution gave almost ideal results without distortion at the beginning or end of the signal based on the simulation and real examples. This property is important for practical applications while processing data split into blocks, when distortions can be significant. The achieved results from real signals verified that the DFT with symmetric extension is the most robust and appropriate method for digital filtering. Additionally, we made a comparison of the numerical costs for different types of filtering, what is essential while big data preprocessing. Beside that, delayless differentiation, integration etc. can be applied only in the DFT domain, which makes the DFT more appropriate than zero phase filtering. The achieved results lead us to recommend the DFT algorithms due to the following benefits: the algorithms are delayless, have straightforward implementation, and are numerically efficient. The design of the filter in the DFT domain is simpler than the classical approach - it does not require filter coefficients and expert knowledge of filter design, what is beneficial for practitioners from various disciplines. We can apply sophisticated frequency response of filters in the DFT domain to solve complex filtering (e.g. extracting a multiple single frequency components). All of these aspects motivates the use of this method for big data preprocessing.

Acknowledgment

The author would like to thank Golden Energy Offshore who contributed by sharing data from the offshore vessel. The author gratefully acknowledges advice and support from Professor H. Langseth from the Norwegian University of Science and Technology and Dr M. Rojewski from the Gdansk University of Technology. Additionally, the author would like to thank colleagues from Rolls-Royce Marine AS especially Dr R.I. Stephens, Dr B. Vik, Dr A.Folkestad and others not mentioned here for many comments and valuable discussions.

Funding

This work was supported by Rolls-Royce Marine AS funded by The Research Council of Norway under Project 251823.

Declaration of conflicting interest

The authors declare that there is no conflict of interest.

References

Bendat, J. S. & Piersol, A. G. (2010), *Random Data: Analysis and Measurement Procedures*, 4 edn, Wiley.

Bishop, C. M. (2011), *Pattern Recognition and Machine Learning*, Springer.

DNVGL (2017), Creating value from data in shipping - practical guide, Technical report.

Duda, R. O., Hart, P. E. & Stork, D. G. (2000), *Pattern Classification*, 2 edn, Wiley-Interscience.

Fossen, T. I. (2002), *Marine Control Systems. Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles.*, Marine Cybernetics AS.

Frnay, B. & Verleysen, M. (2014), 'Classification in the presence of label noise: A survey', *IEEE Transactions on Neural Networks and Learning Systems* **25**(5), 845–869.

García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M. & Herrera, F. (2016), 'Big data preprocessing: methods and prospects', *Big Data Analytics* **1**(1), 9.
URL: *https://doi.org/10.1186/s41044-016-0014-0*

Gustafsson, F. (1996), 'Determining the initial states in forward-backward filtering', *IEEE Transactions on Signal Processing* **44**(4), 988–992.

Hayes, M. (1996), *Statistical digital signal processing and modeling*, New York, John Wiley & Sons, Inc.

Kay, S. (2005), *Intuitive Probability and Random Processes using MATLAB*, Springer.

Kay, S. M. (1999), *Modern Spectral Estimation: Theory and Application*, 1st edn, Prentice Hall.

Kay, S. M. (2013), *Fundamentals of Statistical Signal Processing, Volume III: Practical Algorithm Development*, Prentice Hall.

Kiya, H., Nishikawa, K. & Iwahashi, M. (1994), 'A development of symmetric extension method for subband image coding', *IEEE Transactions on Image Processing* **3**(1), 78–81.

Kuhn, M. & Johnson, K. (2013), *Applied Predictive Modeling*, Springer.

Lyons, R. G. (2010), *Understanding Digital Signal Processing*, 3 edition edn, Prentice Hall.

Mitra, S. K. (2010), *Digital Signal Processing*, 4 edn, McGraw-Hill.

Oppenheim, A. V. & Schafer, R. W. (1975), *Digital Signal Processing*, 1 edition edn, Pearson.

P. Sadovsky, K. B. (2000), Optimisation of the transient response of a digital filter, *in* 'Radioengineering', pp. 14–17.
URL: *http://hdl.handle.net/11012/58229*

Pan, C. (1996), 'Fourier transform processing for digital filters or other spectral resolution devices'. US Patent 5,574,674.
URL: *http://www.google.sr/patents/US5574674*

Pan, C. (2001), 'Gibbs phenomenon removal and digital filtering directly through the fast fourier transform', *IEEE Transactions on Signal Processing* **49**(2), 444–448.

Perera, L. P. (2017), Handling big data in ship performance and navigation monitoring, *in* 'Smart Ship Technology, The Royal Institution of Naval Architects, At London, UK'.

Proakis, J.G., R. C. L. F. & Nikias, C. (1992), *Advanced Digital Signal Processing*, Macmillan.

Qiu, J., Wu, Q., Ding, G., Xu, Y. & Feng, S. (2016), 'A survey of machine learning for big data processing', *EURASIP Journal on Advances in Signal Processing* **2016**(1), 67.
URL: *https://doi.org/10.1186/s13634-016-0355-x*

Randall, R. B. (2011), *Vibration-based Condition Monitoring: Industrial, Aerospace and Automotive Applications*, 1 edn, Wiley.

Rao, K. R. & Yip, P., eds (2000), *The Transform and Data Compression Handbook*, CRC Press, Inc., Boca Raton, FL, USA.

Rødseth, O. J., Perera, L. P. & Mø, B. (2016), Big data in shipping - challenges and opportunities, *in* '15th International Conference on Computer and IT Applications in the Maritime Industries (COMPIT)'.

Slavakis, K., Giannakis, G. B. & Mateos, G. (2014), 'Modeling and optimization for big data analytics: (statistical) learning tools for our era of data deluge', *IEEE Signal Processing Magazine* **31**(5), 18–31.

Smith, M. & Eddins, S. (1987), Subband coding of images with octave band tree structures, *in* 'ICASSP '87. IEEE International Conference on Acoustics, Speech, and Signal Processing', Vol. 12, pp. 1382–1385.

Smith, M. J. T. & Eddins, S. L. (1990), 'Analysis/synthesis techniques for subband image coding', *IEEE Transactions on Acoustics, Speech, and Signal Processing* **38**(8), 1446–1456.

Swider, A. & Pedersen, E. (2017), Influence of loss of synchronisation between signals from various marine systems on robustness of predictive model algorithms, *in* 'Smart Ship Technology, The Royal Institution of Naval Architects, At London, UK'.

Taleb, I., Dssouli, R. & Serhani, M. A. (2015), Big data pre-processing: A quality framework, *in* '2015 IEEE International Congress on Big Data', pp. 191–198.

Theodoridis, S. & Koutroumbas, K. (2008), *Pattern Recognition*, Academic Press;.

Trevor Hastie, Robert Tibshirani, J. F. (2009), *The Elements of Statistical Learning*, Springer-Verlag New York.

Vaseghi, S. V. (2009), *Advanced Digital Signal Processing and Noise Reduction*, 4 edn, Wiley.

Anna Swider holds the position of Data Scientist/Industrial Ph.D. Candidate at Rolls-Royce Marine AS, Norway. She is responsible for research within data analysis for vessel power system design support. Her Industrial Ph.D. Fellowship is within the Faculty of Engineering Science and Technology, Department of Marine Technology (IMT), Norwegian University of Science and Technology (NTNU). She graduated in 2009 - MSc.Eng. from the field of Electronics and Telecommunications, specialisation Real-Time Systems from Gdansk University of Technology and in 2011- Eng. Ocean Engineering and Ship Technology from the same university. Her previous experience includes an engineering job and research for data analysis and preprocessing of measurements in various applications. Her current research include data preprocessing, analysis and statistical modelling.

Eilif Pedersen received the M.Sc. degree in marine engineering from the Norwegian Institute of Technology, Norway, in 1983. He has been with the Norwegian Marine Technology Research Institute, as a Senior Research Engineer until 1999 when he joined The Norwegian University of Science and Technology, as an Associate Professor. His areas of expertise are in the field of modeling methodology and simulation of dynamic multidisciplinary and mechatronic systems focusing on machinery system dynamics, internal combustion engines, vibrations, thermal- and hydraulic machines, fuel-cell system dynamics, and hybrid power plants for marine applications. He has held multiple positions within the university such as Vice Dean of Education at the Faculty of Engineering Science and Technology, Head of Master Programs in Marine Technology, Leader of the Research Group of Marine Systems and Head of Machinery Laboratory at the Department of Marine Technology. Currently he is the Leader of the Power Systems and Fuels work package at the Smart Maritime Center for Research-Based Innovation.