Wavelet analysis for real-time determination of the sawtooth behavior in non-stationary fusion plasmas

M. van Berkel
CST 2010.040

Master’s thesis

Committee:  ir. G. Witvoet¹ (coach)
             dr. ir. P.W.J.M. Nuij¹ (supervisor)
             prof. dr. ir. M. Steinbuch¹ (supervisor)
             prof. dr. M.R. de Baar¹
             dr. H.G. ter Morsche²
             prof. dr. N.J. Lopes Cardozo³

¹Eindhoven University of Technology
   Department of Mechanical Engineering
   Control Systems Technology Group

²Eindhoven University of Technology
   Department of Mathematics and Computing Science

³Eindhoven University of Technology
   Department of Applied Physics
   Science and Technology of Nuclear Fusion

Eindhoven, June, 2010
Contents

Wavelet analysis for real-time determination of the sawtooth behavior in non-stationary fusion plasmas 1
  Introduction ................................................................. 1
  Principles of Wavelet theory ..................................... 2
  Application sawtooth signal ...................................... 3
  Results ................................................................. 7
  Summary and Conclusions ....................................... 11

Future Research 13

Appendices 15

A Fast Wavelet Transform in the frequency domain 17
  A.1 Properties of the Fourier Transform ....................... 17
  A.2 Wavelet and scaling functions time domain ............... 17
  A.3 Wavelet and scaling functions in frequency domain .... 18
  A.4 Fast Wavelet Transform in frequency domain .......... 18
  A.5 Samples as scaling coefficients ............................. 19

B Inversion radius 21
  B.1 Theoretical background ...................................... 21
  B.2 Temperature estimation ...................................... 22
  B.3 Estimate temperature without Thomson scattering .......... 22
  B.4 Example calculation Sensitivity (Shot 106478) ........... 23
  B.5 Amplitude versus wavelet coefficients .................... 24
  B.6 Polynomial fit .................................................. 25
  B.7 Results ............................................................. 25
  B.8 Summary ............................................................. 25

C Tuning and practical considerations 27
Wavelet analysis for real-time determination of the sawtooth behavior in non-stationary fusion plasmas

M. van Berkel\textsuperscript{1,2}, G. Witvoet\textsuperscript{1,2}, M.R. de Baar\textsuperscript{1,2}, P.W.J.M. Nuij\textsuperscript{1} and M. Steinbuch\textsuperscript{1}

Abstract—Nuclear fusion plasmas are subject to many different non-stationary phenomena, which need to be controlled. A suitable technique to detect and distinguish these phenomena is Wavelet Analysis, currently mainly presented in the form of a Morlet Wavelet through a CWT, which uses averaging and is time intensive to calculate. Other wavelets allow a more local signal analysis and if implemented in a Fast Wavelet Transform (filter bank) can be used real-time. In this paper a proof of principle is given of such a time-scale analysis through an example: the determination of the period of the sawtooth instability through ECE-measurements using a B-spline wavelet based on Canny Edge-Detection.

I. INTRODUCTION

HIGH performance nuclear fusion plasmas are operated in the vicinity of a number of operational limits. Quasi-periodic reconnection events occur that could push the plasma over an operational limit, and hence deteriorate the plasma performance or lead to plasma disruption. Examples of such behavior are the sawtooth crash that may trigger a Neo-classical Tearing Mode or the Edge-Localized Modes that are associated with the overall confinement and heat losses to the wall. Such events may require real-time control of the plasma, or controlled plasma termination. The period between the reconnection events is an important qualifier for assessing the regime in which the plasma is operated, and the impact of the event on the ambient plasma. This implies that for real-time control of the high performance plasmas, real-time analysis of the local plasma behavior is required.

The traditional tool for such a signal analysis is the Fourier transform, which can be windowed resulting in the Short Time Fourier Transform (STFT), to study non-stationary plasma events. The time-frequency resolution created by the STFT is currently the most popular method for non-stationary plasma analysis of which a number of examples are discussed in [1]. Nevertheless, the time-frequency resolution calculated with the STFT is not optimal for every point in the time-frequency plane. Often the resolution is inadequate for detection.

The natural extension to further improve resolution is using a Gaussian window that is optimal for the product of time and frequency, such that amplitudes in time-frequency can be better distinguished. This is known as the Continuous Wavelet Transform (CWT) using a Morlet wavelet, which is already applied to a variety of fusion plasma problems. Important applications of the Morlet wavelet are the period determination of sawtoothing plasmas and that of the precursors based on measurements of for instance: a heavy ion beam probe in TEXT-U [2]; soft X-ray (SXR) diagnostics in ASDEX [3]; and magnetic signals from pickup coil signals in JET [4]. Despite the improvement in time-frequency resolution of the Morlet wavelet compared to the STFT, the window still encompasses multiple sawteeth (effective support of 5.5 periods), where the period with the most weight in the averaging process is delayed by about 3 periods. Moreover, due to the averaging it is not possible to distinguish individual periods. Furthermore, the CWT (and also STFT) takes a great computational effort to calculate and is therefore difficult to implement in real-time applications with a high sample frequency, even when the number of calculations is reduced by choosing a dyadic grid at the cost of resolution. Another application of the Morlet wavelet is the analysis of Edge Localized Modes (ELMs), on which wavelet analysis is applied to analyze ELM precursors, postcursors [5], [6] and ELM type characterization [7]. Other applications of the Morlet wavelet are the analysis of Neoclassical Tearing Modes [8], nonlinear phenomena and intermittency in plasma turbulence analysis by means of wavelet bi-coherence [9], [10], however with the same disadvantages as explained before.

The Wigner-Ville Distribution (WVD) and Choi-Williams Distribution (CWD) give a better time-frequency resolution compared to the Morlet wavelet because both the WVD and CWD analyze the density of energy by correlating the analyzed signal with a time and frequency translation of itself in the time-frequency plane resulting in no loss of resolution. The WVD and CWD have been used for off-line applications [11] e.g. analysis of magnetic signals from pickup coils in JET [4]. However, the product between time and frequency of the WVD results in a quadratic form, which leads to interference, creating artifacts. Therefore, in practice, the WVD is difficult to use. In contrast, the CWD reduces these artifacts by smoothing the WVD but at the cost of resolution. Despite this smoothing, the artifacts cannot be removed entirely, and therefore they remain problematic for automated algorithms. The WVD and CWD take a lot of computational effort because it is necessary to calculate the Fast Fourier Transform of the convolution between half shifted versions of the signal [12]. The authors are currently unaware of real-time implementation with high sample frequencies of the WVD or CWD, which seem unsuitable for real-time algorithms.

The STFT, Morlet wavelet and CWD have in common that they study the frequency content of signals and are time-
Further improvement of the algorithm by combining channels of wavelet and accompanying choice of threshold is described. In practice, i.e., the Fast Wavelet Transform (FWT), it is also able to handle applications sampled with a high frequency. However, the variation of the feature should be within certain boundaries, which are determined by the size of the wavelet such that the feature can be detected properly.

In this paper the benefits of wavelet theory are fully exploited. The features that describe the reconnection events can be much better localized in time using a coupling of scales; this allows a much better resolution than simple filters or alternative linear methods. Another important aspect is that robustness can be introduced without the cost of extra delay. In addition these wavelets allow a real-time implementation with a very local resolution. The great advantages of local feature detection (time-scale analysis) using wavelets becomes even more clear in the application discussed in this paper: the determination of the period of the sawtooth instability through Electron Cyclotron Emission (ECE) measurements using a B-spline wavelet based on Edge-Detection.

The sawtooth instability has a clear crash feature, however, due to the noise sensitivity of the ECE-measurements [17] and the dynamical behavior of plasmas [18], a simple edge detector algorithm does not suffice to robustly and accurately determine the sawtooth period, which can be problematic for real-time control of the sawtooth period. Therefore, multiresolution analysis is used to determine the sawtooth fast, accurately and robustly. In the TEXTOR in-line ECE system six measurement channels are available, which by combining can further improve the wavelet detection algorithm.

This paper is organized as follows. In Section II the basics of wavelet theory are explained. This includes the CWT, the idea of multiresolution, and the efficient implementation of wavelets in practice, i.e. the FWT. In Section III, the choice of wavelet and accompanying choice of threshold is described. Further improvement of the algorithm by combining channels is given. In Section IV the results based on ECE-measurements from TEXTOR are presented and the applicability for the compound regime is indicated. Finally, in Section V a summary and conclusion is presented.

II. PRINCIPLES OF WAVELET THEORY

The Morlet wavelet transform is the most popular wavelet for the analysis of non-stationary fusion plasmas [15]. Its popularity is based on the fact that the CWT of a Morlet Wavelet gives time-frequency analysis, which is a familiar concept from Fourier analysis. However, the Morlet wavelet is only a particular example of a much broader family of wavelet functions. The Fourier transform expresses the signal in terms of harmonic functions. Similarly a wavelet transform expresses the signal in terms of wavelets, which by summation can reconstruct the signal. However, wavelets are localized in time, which allows them to study non-stationary signal behavior. All wavelets have finite energy and fulfill a weak admissibility condition [19] such that there is energy conservation; in practice this means that a (mother) wavelet \( \psi(t) \) is defined as a function that has zero average:

\[
\int_{-\infty}^{+\infty} \psi(t) \, dt = 0. \tag{1}
\]

This wavelet \( \psi(t) \) forms the bases of the wavelet transform.

A. Continuous Wavelet Transform

A Continuous Wavelet Transform (CWT) analyzes a signal \( f(t) \) by convolving it with different dilations \( s \in \mathbb{R}^+ \) and translations \( \tau \in \mathbb{R} \) of the (mother) wavelet \( \psi(t) \) [12]:

\[
W \{ f(\tau, s) \} = \langle f, \psi_{\tau,s} \rangle = \int_{-\infty}^{+\infty} f(t) \frac{1}{s} \hat{\psi} \left( \frac{t - \tau}{s} \right) dt,
\]

where the factor \( s^{-1/2} \) is introduced such that the transform is norm preserving, i.e., \( \| \psi_{\tau,s} \|_2 = 1 \). This transform is known as the CWT because it results in a continuous two dimensional function. The convolution expresses the amount of overlap as the wavelets are dilated with \( s \) and are shifted with \( \tau \) over the signal, in other words if it matches well, the resulting wavelet coefficients are big and vice versa. Wavelet coefficients of scales \( s \sim 1 \) describe mainly the high-frequent behavior (bands in the high-frequent region), which are the fine and local properties of signals. By moving to \( s \gg 1 \) the pass-band moves to the low-frequent spectrum of the signal and therefore the wavelet coefficients describe the more coarse and low-frequent behavior of the signal. Such an analysis is called multiresolution.

Multiresolution is an important property because it allows the study of phenomena on different scales (‘frequency’), which often gives more insight, and allows indiscernible phenomena to become visible or better distinguishable. How well these phenomena are represented in wavelet coefficients depends on the choice of wavelet. In essence the wavelet transform can identify energy in both time and scale, however, its localization in time and frequency is bounded by the
Heisenberg uncertainty principle [12], which describes the optimal resolution of the product between time and scale.

The CWT gives the highly redundant representation of the signal in wavelet coefficients, i.e., the resulting wavelet coefficients give the most complete image of the wavelet transform. The disadvantage is that every wavelet coefficient needs to be calculated separately for all different dilations and translations, which costs a lot of calculation power.

B. Fast Wavelet Transform

The most popular wavelet transform is the Discrete Wavelet Transform (DWT), which uses continuous wavelets but on a discrete grid of exponential ($s_0 > 1$) dilations $m \in \mathbb{Z}$ and translations $n \in \mathbb{Z}$ such that discrete wavelet coefficients are calculated [20]:

$$d_m[n] = \langle f, \psi_{m,n} \rangle = \int_{-\infty}^{+\infty} f(t) s_0^{-m/2} \psi(s_0^{-m}t-n) \, dt.$$  \hspace{1cm} (3)

The wavelet transform can be implemented very efficiently by introducing an additional function called the scaling function, which is dilated and translated similarly to $\psi_{m,n}$:

$$a_m[n] = \langle f, \phi_{m,n} \rangle = \int_{-\infty}^{+\infty} f(t) s_0^{-m/2} \phi(s_0^{-m}t-n) \, dt.$$  \hspace{1cm} (4)

However, note that $\phi(t)$ is not a wavelet because $\int_{-\infty}^{+\infty} \phi(t) \, dt = 1$, see (1). The resulting coefficients $a_m[n]$ are called scaling coefficients and are a result of the convolution between $f(t)$ and the newly introduced scaling function $\phi(t)$. The DWT can be implemented in a Fast Wavelet Transform (FWT) if the scaling function has the following two properties, which often only hold for special choices of $s_0$: $\phi(\frac{t}{s_0})$ can be written as a summation of different translations of $\phi(t)$, and $\psi(\frac{t}{s_0})$ can be written as a summation of different translations of $\phi(t)$ [12]:

$$\phi(\frac{t}{s_0}) = \sqrt{s_0} \sum_{n=-\infty}^{\infty} l[-n] \phi(t-n) \quad \text{and}, \quad \psi(\frac{t}{s_0}) = \sqrt{s_0} \sum_{n=-\infty}^{\infty} h[-n] \phi(t-n),$$  \hspace{1cm} (5)

where $l[-n]$ and $h[-n]$ can also be expressed as filters in the $Z$-domain by $X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$.

The twoscale relation (5) can be transformed, such that the scaling coefficients of a coarse scale can be expressed in terms of fine scales, whereas (6) expresses the transformation from scaling coefficients to wavelet coefficients. The initial scaling coefficients $a_0$ are approximated by discretizing (4), which acts as a pre-filter operation:

$$a_0[n] = \langle f, \phi_{0,n} \rangle = \sum_{k=-\infty}^{\infty} f[k] \phi_0[k-n].$$  \hspace{1cm} (7)

The size of $\phi_0$ is used to determine the initial size of the first wavelet $\psi_0$. The initial approximation error depends on the number of samples encompassed by $\phi_0$. Together (5), (6) and (7) form the basis for the FWT, which in practice is applied in the form of a filter bank shown in Fig. 1. The great advantage of the FWT is that wavelet coefficients are calculated by a cascade of filters, instead of separately as is the case for the CWT, hence the calculation effort is reduced significantly. The disadvantages are that the grid in s-direction is often limited to certain $s_0$, and that not every wavelet is suitable to be written in terms of (5) and (6) e.g., the Morlet wavelet.

There are different variants of filter banks, in this paper is chosen for a FWT without downsampling, the so-called à trous algorithm or Stationary Wavelet Transform (SWT). It has the advantage of being time-invariant and retaining maximum resolution in time, which is not the case when downsampling is used. In addition it is less computational efficient as the FWT with downsampling. In practice a FWT transform can be calculated using three filters: an initialization filter or prefilter $\Phi_0(z^{-1})$ from (7), low-pass filters $L(z^j)$ and high-pass filters $H(z^j)$ from (5) and (6), which are represented in Fig. 1.

Remark 1: The filter bank with downsampling is the most popular variant because it is very efficient and gives a sparse signal representation. Its filters do not change with $m$ because the downsampling removes samples compensating for $j$, however, it comes with significant losses to the resolution especially for $m \gg 1$. Therefore, downsampling can reduce the accuracy of any detection algorithm and leads to delays because the sample rate has been reduced substantially.

Remark 2: It is also possible to choose the samples of the signal $f[k]$ as the scaling coefficients $a_0[n]$ because the product of the scaling filters converges to the scaling functions i.e. $\Phi(e^{j\omega}) = \prod_{j=0}^{\infty} L(z^{-j})$. However, the error is considerable for the first scales and a mechanism to control the initial size of the wavelet is lost.

III. APPLICATION SAWTOOTH SIGNAL

In this section the FWT is applied to signals with sawteeth. These sawteeth are a consequence of a periodically occurring instability, which has many important effects on the plasma behavior inside tokamaks. Therefore the real-time behavior of the sawtooth instability and particularly the sawtooth period gives insight into the plasma behavior. In addition, control of the period is necessary, to optimize between the

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{The Dyadic Filter Bank for the SWT used to calculate the wavelet coefficients, where $\Phi_0(z^{-1})$ is the initialization filter, $L(z^j)$ the low-pass filters and $H(z^j)$ the wavelet filters, with $j = s_0^n$ (normalization factors have been omitted).}
\end{figure}
The deterministic signal describes the unscaled temperature inside the tokamak and an insight signal can be modeled by a deterministic signal, which what type of noise is present in the ECE-signals. The line-of-sight detection algorithm. Therefore, it is possible to establish the sawtooth period. In addition, if the sawtooth period needs to be controlled in real-time, the sawtooth period detection algorithm should also have a small delay. In the TEXTOR in-line ECE system six measurement channels are available, which can be combined to further improve the wavelet detection algorithm.

A. Noise analysis of the ECE-signals

The properties of the signal play an important role in the selection of the wavelet. Therefore it is important to establish what type of noise is present in the ECE-signals. The line-of-sight signal can be modeled by a deterministic signal, which describes the unscaled temperature inside the tokamak and an additive noise part:

\[ x[n] = u[n] + N[n]. \]  

(8)

The deterministic signal \( u[n] \) consists of the sawtoothing plasma signal. The deterministic sawtooth signal is dominant in the lower frequency regions, while the noise is dominant in the high-frequency region. Therefore, it is possible to separate noise from the ramp. The distribution is clearly of positive and negative effects. Consequently, measurement of the period is essential.

In this paper the sawtooth instability is studied by means of in-line ECE-measurements in TEXTOR, which indirectly measures plasma temperatures [17]. This system is chosen because it is combined with Electron Cyclotron Resonance Heating (ECRH), such that the sawtooth period can be altered using ECRH in the same line of sight if necessary. In Fig. 2 such ECE-measurements in TEXTOR are shown. The sawtooth period can be determined based on the crash [13], [14], however, due to the noise sensitivity of ECE-measurements and the dynamical behavior of plasmas, a simple edge detector algorithm often does not suffice to robustly and accurately determine the sawtooth period. In addition, if the sawtooth period needs to be controlled in real-time, the sawtooth period detection algorithm should also have a small delay. In the TEXTOR in-line ECE system six measurement channels are available, which can be combined to further improve the wavelet detection algorithm.

Figure 2: Line-of-sight receiver data sampled at 100 kHz, TEXTOR discharge No. 106482. It shows a detailed view around 1.6 s, such that the periodic sawtooth behavior is visible. The inversion radius, which is situated near the \( q = 1 \) surface, just above channel 2 (135.5 GHz).

Figure 3: Real distribution of the noise on ECE 135.5 (1.5-1.8 s ramp only) compared to the probability density function (pdf) with same variance and zero mean. Auto-correlation of the noise ECE 132.5 GHz and the Cross-correlation of the noise between ECE 132.5 and ECE 135.5 GHz.

B. Choice of wavelet

The wavelet is selected such that it can robustly and accurately detect sawtooth crashes in Gaussian noise environments. From classic edge detection it is well known that the optimal edge detector for step-edges in signals flooded by Gaussian noise is the first derivative of a Gaussian function, which is often referred to as the Canny edge detector [23]. The local maximum of the resulting convolution between the signal and detector indicates the position of the edge.

The first derivative of a Gaussian is a continuous wavelet. If it is applied to one sample shift, \( \sigma \), the wavelet is optimal for every dilation, the criteria SNR and localization cannot be optimized simultaneously when the single-response criterion applies; they are subject to an uncertainty principle. Small dilations of the wavelet react on every edge and therefore give many responses that are well localized, but the responses are relatively small (small SNR) and therefore difficult to distinguish. On the other hand a big dilation (\( s \gg 1 \)) only rec-
Figure 4: Line-of-sight receiver data of noisy channel 2 (135.5 GHz) and channel 3 (138.5 GHz) of TEXTOR discharge No. 106482 measured around 1.6 s and their Continuous Wavelet Transforms (contour plots) using the first derivative of the wavelet, reveal the sawtooth crashes. However, these lines do not end on a big dilation and become narrower moving to small dilations. CWT are shown, where vertical dark lines, which are wide at a big dilation, where it is clearly distinguishable, its location of the edge. The key result is that when an edge is detected in a big dilation, where it is clearly distinguishable, its position of the edge can be well determined.

 recognizes big edges, however, the maximum is also influenced by events around the edge, making it difficult to determine the location of the edge. The key result is that when an edge is detected in a big dilation, it is clearly distinguishable, its maxima line is followed. This line, called a maxima chain, is followed until the well localized maxima at the position of the edge is found, i.e. sawtooth crashes can be well distinguished and at the same time can be well localized. It has also been shown to work well for points of sharp variations such as ridges and other edges [24]. In Fig. 4, the contour plots of a CWT are shown, where vertical dark lines, which are wide at a big dilation and become narrower moving to small dilations of the wavelet, reveal the sawtooth crashes. However, these lines do not end on \( s = 1 \) but somewhat above, which is a consequence of noise and ridges being analyzed by a step edge detector. This makes it impossible to find the middle of the edge in one point. Nevertheless, the position of the edge can be well determined.

Figure 5: Cubic B-spline and its first time derivative, where the grey lines present the half-size B-splines used in (5) and (6).

C. Discrete implementation

The first derivative of a Gaussian is a continuous wavelet, which is not suitable to use in a FWT. A common method to approximate a Gaussian function for this type of application is by using compactly supported B-splines, which fulfill (5) and (6) [24]. The B-spline \( \beta^p(t) \), which serves as the scaling function \( \phi(t) \), is constructed by repeated convolutions of the box-spline. The scaling function converges with increasing order \( p \) to a Gaussian (for odd \( p \)) [25]:

\[
\phi(t) = \beta^p(t) = \frac{1}{p!} \sum_{q} \left( \frac{p+1}{q} \right) (-1)^q \left( t - q + \frac{p+1}{2} \right)_+^p.
\]

(9)

The B-spline fulfills (5) for any \( s_0 \in \mathbb{N} \) [26]:

\[
\frac{1}{s_0} \beta^p \left( \frac{t}{s_0} \right) = \sum_{n=-\infty}^{+\infty} b_{s_0}^p [n] \beta^p(t-n),
\]

(10)

where \( b_{s_0}^p(z) \) respectively \( L(z) \) in terms of (5) is calculated by:

\[
L(z) = B_{s_0}^p (z) = \frac{1}{s_0^p} \left( \sum_{n=0}^{s_0-1} z^{-n} \right)^{p+1}.
\]

(11)

The approximation order \( p \) and the dilation \( s_0 \) can be chosen differently for the application and consequently give extra design freedom.

In this paper is chosen to use a cubic B-spline \( (p = 3) \) (Fig. 5) because its approximation error is less than 3% compared to a Gaussian and it is within 2% of the limit of the Heisenberg uncertainty [27]. The cubic B-spline is applied on a dyadic grid \( (s_0 = 2) \), which is a common choice. It has the advantage that the calculated scales are relatively close together where a maxima chain is difficult to follow and scales are far apart, where detection is important. The B-spline (9) needs to be interpolated as described in (7) to determine \( \Phi(z) \) such that the first scale gives generally useful maxima (symmetric function i.e. \( \Phi(z) = \Phi(z^{-1}) \)). Another important property of B-splines is that the difference of two shifted B-splines gives an even better approximation \((p+1)\) of the first derivative of a Gaussian [25]:
Figure 6: Real-time response of the wavelet coefficients on a sawtooth measured on 132.5 GHz of discharge No. 106482, where a very robust threshold is chosen. The right pointing triangle (red) is the point of detection, the other triangles present the selected maxima, where the last triangle (black) illustrates the chosen maxima. The vertical dash-dot line shows the real-time instant of detection (rt det.). The wavelet coefficients for the different scales are the parabola shaped curves from $m=1$ (yellow) to $m=7$ (black). The sawtooth has been scaled for illustrational reasons.

It can be rewritten similarly to (6) such that is is possible to define $h[n]$:

$$
\psi(t) = \sum_{n=-\infty}^{\infty} h[n] \beta^p(t-n), \text{ with } h[n] = [1, -1].
$$

The resulting filters, which completely define the wavelet transform: $L(z)$, $H(z)$ and $\Phi(z)$ can also be found in Table I in the Appendix.

**Remark 3:** The second derivative of a Gaussian (a.k.a. the Mexican hat wavelet) can be found using $h[n] = [1, -2, 1]$ [26]. Edges are detected by means of zero-crossings (Marr-Mildreth [28]) and is generally seen as equivalent to the first derivative [23], [24].

**D. Real-time sawtooth crash detection**

Sawtooth crashes are detected when the wavelet coefficients exceed a critical threshold value, which are calculated real-time resulting in delayed responses. This is shown in Fig. 6. Big dilations of the wavelet are delayed more compared to that of the small dilations. In contrast, wavelet coefficients calculated with small dilations often do not exceed a chosen threshold, whereas for big dilations the coefficients exceed the threshold (Section III-B). Therefore, a tight threshold is sensitive to false detections, especially in scenarios with small partial reconnection events, but on the other hand the edge detector has a small delay.

It is difficult to calculate the optimal threshold, however, it is possible to determine a lower-bound for the threshold. The minimum threshold should always be taken such that it will not detect wavelet coefficients that are the result of noise. Therefore, an adaptive threshold is used such that a near optimal trade-off between robustness and delay can be achieved. As explained in Section III-A the filter bank entrance noise is of Gaussian nature, which means that the wavelet coefficients resulting from this noise are also Gaussian distributed [29]. The finiteness of the signal means that it is possible to define a risk, in terms of the variance ($\sigma^2$), of detecting noise. A well-known threshold from wavelet theory of which the maximum noise amplitude has a very high probability of being just below is [12]:

$$
T_{\min} = \sigma \sqrt{2 \log_2 N}.
$$

The lower bound threshold $T_{\min}$ depends on the number of sample points $N$, which is a logical consequence of the fact that if the number of points increase, the chance of taking points from the tail of the Gaussian distribution increases.

The variance of the noise in (13) on which the threshold is based, needs to be calculated for every scale. This would make the algorithm very inefficient. However, it is possible to make an estimate of the variance for every scale based on the 2-norm of one scale. Every branch of the filter bank can be represented by one equivalent filter $W[n]$, which is normalized to one. This makes it possible to estimate an upperbound on the noise in terms of the variance, which acts as the lower bound on the threshold (Cauchy-Schwarz inequality):

$$
|\langle W, N \rangle| \leq \|W[n]\|_2 \|N[n]\|_2.
$$

The variance is estimated by the output of the first scale ($d_1$), which mainly consists of noise. This has the advantage that no additional calculation steps need to be taken to separate the noise and allows for a bit more robust estimation because the small responses on sawteeth are included. Other possibilities are to estimate the signals noise by use of automatic variance estimators, which are present in many signal processing software, which allows for a threshold purely based on noise but lacks robustness on dynamical behavior of the plasma.

The real threshold is always chosen above $T_{\min}$. However, this threshold is not robust due to the dynamical plasma behavior. Therefore it needs to be chosen above $T_{\min}$ (as a multiple of $T_{\min}$). This is a good alternative because Fig. 6 clearly shows there is still a significant margin above the real noise level ($\approx 0.2$) to detect the sawtooth crash. It is even possible to add additional scales until a wavelet that has the size of twice the period such that the response is even more significant and the region in which the threshold can be chosen is extended but at the cost of delay and calculation time.

The threshold is preferably chosen one-sided due to the fact that bigger wavelets also start responding on the ramp-part of the signal but in negative direction. For a two sided threshold this would lead to an over robustly threshold, which eventually would lead to misses. In addition only maxima of the different wavelet scales on one side need to be determined.
and stored. However, this makes a real-time estimate of the inversion radius necessary.

Summarizing, it is possible to estimate an absolute lower-bound on the threshold, however, it remains difficult to identify an optimal threshold due to many different scenarios of sawtothing plasmas. Consequently, to embed some extra robustness, the wavelet coefficients of the first scale are used as noise estimate, however, it is still necessary to take a higher threshold.

E. Period determination

Sawtooth crashes can be detected robustly, however, it is still necessary to accurately determine the location of the crash and the resulting period. This is done by means of a maxima chain (Section III-B) whereby a number of maxima, which indicate the middle of the edge, are followed from the maxima in a scale \( m > 1 \) to lower scales preferably to \( m = 1 \). The maxima are detected in every scale by a simple maxima detector \( (d[1] < d_{\text{max}}[2] > d[3]) \) and are stored in a buffer, which has the size of the wavelet with the biggest dilation. Otherwise, due to the different delays for the different scales, important maxima can be lost.

Crashes are detected in scales \( m > 1 \). The wavelet coefficients \( d_{m-1} \) are less delayed than \( d_m \), consequently it is generally possible to detect a sawtooth crash at scale \( m \) and select the maximum at scale \( m - 1 \), which belongs to the same crash, this can be followed to the smallest crash (see Fig. 6). Great advantage is that at the time instant the sawtooth crash is detected, an accurate estimate of the position is calculated.

The non-causal responses (without delay) need to be calculated to be able to follow the maxima. This calculation is straightforward because the product of all filters in one branch of the filter bank, which describe the wavelet, are anti-symmetric FIR-filters (linear-phase). Consequently the delay for every scale is the length of the wavelet filter divided by two. The non-causal responses are shown in Fig. 7.

It is not always possible to follow the maxima chain to the smallest scale \( m = 1 \) because sometimes there are more than one or no maxima in the vicinity of the previous maxima. Consequently it is difficult to determine which maxima belongs to the maxima of the detected edge. Therefore the maxima in the smallest scale that can be traced back to the edge is selected as the location of the crash, which could also be scale \( m - 1 \). The period is then calculated by taking the difference between two consecutive crashes. In summary, the period determination takes place immediately after the detection and its resolution is in principle independent of the detection scale. The maxima chain allows a very accurate localization of the crash, hence also of the period.

F. Combining channels

The line-of-sight ECE-system consists of six channels. Therefore it is possible to improve the wavelet algorithm by combining the different channels. Not all channels are suitable: especially those near to the inversion radius often contain more noise than signal, which can lead to loss of detection.

The responses of the wavelet coefficients on a crash for the different channels on a scale \( m \gg 1 \) give significant responses, which are related to the amplitude and the sign of the sawtooth. This also holds for a channel near the inversion radius where the wavelet coefficients are too low to trigger a threshold, but still give a distinguishable response compared to the noise level. The sign and amplitude of the wavelet coefficients on all channels at a crash indicate the location of the inversion radius, which can be estimated by finding the zero-crossing of a function fitted through these wavelet coefficients (channels need to be calibrated).

The wavelet transform is performed parallel on channels 1-5. Channel 6 (147.5 GHz) is prone to non-linearity and cross-talk and has been gained extremely high [30]. Therefore, it cannot be trusted; this channel is not used in the algorithm. It is possible by comparing the channels 1-5 to suppress these false crashes, because often a false crash only occurs on one channel. Therefore, an evaluation block has been designed, which observes real-time the crash time instants, if enough crashes (here 2) from the different channels have been detected near the same location, the mean is taken. Together with the previous crash, the difference is calculated resulting in the period. The next detection events are blocked for some period after the detection. If during a certain observation time only one (false) crash is detected, the specific channel is reset.

IV. Results

ASIC sawteeth, as presented in Fig. 7, do not pose any problem for this type of algorithm. However, there is a variety of sawteeth that are much more difficult to detect, whereof some are discussed here. The sawteeth are all extracted from Channel 1 (132.5 GHz) of TEXTOR discharges.
No. 106482 and No. 107915. The responses are calculated by a detection algorithm, which is implemented in a real-time testing environment (Simulink®). The chosen parameters are found in Table of the Appendix II. In addition to the sawteeth, also the period is calculated in a real-time environment, for the individual channels and for the combination of channels.

A. Bump

In Fig. 8 the response on a crash, which is succeeded by a bump is shown. Initially, two falling edges are detected by $m = 4, 5$ because these wavelets only encompass one edge. However if the wavelets become bigger, the edges slowly start to grow together, which is the result of the wavelet $m = 6$ (size 1.66 ms) encompassing parts of both edges. At a certain point the wavelet $m = 7$ (size 3.26 ms) overlaps both edges entirely and therefore it is treated as one edge. Nevertheless, the hole between the real crash and the bump reduces the magnitude of the wavelet coefficients, which makes it more difficult to detect this edge using a robust threshold. The crash is detected accurately because delay is taken into account, resulting in the left maxima of $m = 4$.

B. Crash with precursor

Fig. 9 shows a classic sawtooth [18], where the crash is preceded by a precursor. Small wavelets respond on the precursor and are positive for a downwards edge and negative for an upwards edge. Consequently the small sized wavelets at scale $m = 3$ (blue), which is twice the size of the wavelet at $m = 2$ (green), start to oscillate with the precursor. The wavelet at scale $m = 4$ (cyan) encompasses part of both edge directions and therefore they compensate each-other resulting in small coefficients. At $m = 6$ (red) and also $m = 7$ (black), the wavelet captures the more global behavior whereby it is recognized that at the beginning of the precursor the overall temperature slowly drops and then rises a bit, where-after the crash takes place. This crash is detected in $m = 6$ and the maximum chain is followed accurately to the middle of the real crash. At the moment of detection in scale $m = 6$, the wavelet coefficients of $m = 7$ are still below the threshold because they are delayed compared to $m = 6$ (see Fig. 6). Therefore scale $m = 7$ plays no role anymore in the detection of this crash.

C. Dominating precursor

Sometimes the sawtooth behavior is not recognizable at all but only the ‘precursor’, which is represented by strong oscillating behavior in ECE-measurements (Fig. 10). Nevertheless the periodic behavior is still detectable, but it is more difficult. The oscillation is amplified by the multiresolution analysis. Although hard to recognize here, the crash is upwards, which means that the response and threshold are negative. The
wavelets at $m = 3$ (size 0.23 ms) and $m = 4$ (size 0.46 ms) are both in the range of the precursor period, which is around 0.30 ms and therefore start to oscillate. However, when comparing the wavelet of $m = 4$ to the signal, it matches somewhat better resulting in higher amplitudes. The scales $m = 6, 7$ only capture the general movement of the crash, which is upwards in the middle of the cursor but without a real crash, hence these scales do not detect much.

**D. Near detection failure**

Sometimes sawteeth crashes are difficult to detect in a regime with well conditioned sawteeth due to a very different crash behavior (Fig. 11). In this case the middle crash is nearly missed because of three factors: 1) the threshold is very robust (compared to the noise level). This does not pose a problem for the surrounding crashes where detection already takes place in $m = 6$; 2) the crash is not as deep as the surrounding crashes and therefore also the response is less high; 3) a precursor is present, which reduces the resulting magnitude of the wavelet coefficients. Together this results in a miss. However, here the number of scales has been extended such that this crash can be detected ($m = 9$) at the cost of extra calculation time.

These extra scales also have more delay, however, this delay only applies to the crash at the nearly missed crash and not to the surrounding crashes, which are detected at $m = 6$. The accuracy is not affected at all because Fig. 11 clearly shows that it is still possible to form a maxima chain such that the crash can be accurately detected from a starting point on $m = 9$. However, the time that detection is suppressed to prevent multiple detections on the same edge needs to become longer. Furthermore, it is important to track the direction of the sawteeth (inversion radius) because a double threshold will give here false responses.

**E. Period of discharges No. 106482 and No. 107915**

From the detected crashes it is possible to determine the sawtooth period. The results are presented for two discharges No. 106482 and No. 107915, respectively Fig. 12 and Fig. 13, where the first has a nearly constant sawtooth period and the second a fast changing period. Although a lot of bumps and other spikes are present in discharge No. 106482, it is possible to detect the period on four of the five channels. The noisy channel near the inversion radius shows many misses due to a high threshold resulting in unrealistic periods. This threshold is relatively high compared to the small depth of the sawteeth because the risk of detecting noise in combination with a choice of a higher threshold (above $T_{\text{min}}$) to prevent detection on dynamical behavior leads to high threshold. Nevertheless it poses no problem here because more channels are available and it is still possible to choose a lower threshold, however, increasing the risk of false detection.
The period of discharge No. 107915 is more difficult to measure due to a lot of dynamical behavior, which is not fully understood and an inversion radius that moves through channels, which is represented in Fig. 14. False detections rarely occur due to a robust threshold, however resulting in errors, which are predominately made by misses. This can be taken care of by combining channels to determine robustly the sawtooth period. From this discharge the great advantage of the wavelet based algorithm becomes clear because it is possible to identify the individual periods, whereas the frequency based methods will give an averaged period.

F. Compound regime

The real-time wavelet detection algorithm due to multiresolution can distinguish small intermediate crashes from bigger real crashes (see Fig. 15). The smaller wavelets up to scale $m = 7$ react strongly on the intermediate crashes at 2.48 s and 2.518 s as if it are normal crashes. However, if the scales become bigger, the magnitude of the wavelet coefficients become smaller again, where $m = 8$ still detects the intermediate crashes but its magnitude is smaller than the previous scale $m = 7$. At scale $m = 9$ the intermediate crashes starts to disappear, whereas the magnitude for the real crashes at 2.47 s, 2.49 s and 2.51 s have become bigger (but delayed). Although in $m = 9$ the wavelet coefficients are negative at the intermediate crashes, it is still recognizable by the upward movement that an intermediate crash is present. Consequently (in principle), multiresolution analysis can differentiate between intermediate crashes and real crashes, which allows identification of compound regimes.

G. Accuracy

It is generally not possible to give a real estimate of the error because the real period is unknown. However, it is possible for a regime with well defined sawtooth crashes to compare the calculated period to the individual estimates of the period. This is presented in Fig. 16. The uncertainty of the period fluctuates much more than the difference between the individual channels, which are generally close to each other. This also follows from the standard deviations, measured from $1 \rightarrow 4$ s, where the standard deviation of the period is $\sigma_{\text{period}} = 0.51$ ms and respectively of the differences: $\sigma_{132.5-138.5} = 0.13$ ms, $\sigma_{132.5-141.5} = 0.22$ ms, and $\sigma_{138.5-141.5} = 0.20$ ms. Therefore it can be concluded that the error on the period estimates of the individual channels is much smaller than that of the uncertainty of the period itself. The two signals with the best SNR (ECE 132.5 and 138.5 (see Fig. 2)) give the best estimates, the difference between the best estimates compared to the uncertainty is almost 4 times higher, consequently it is not useful for general sawtooths to further improve the algorithm with respect to the accuracy.
V. SUMMARY AND CONCLUSIONS

A
dlalgorithm for accurate real-time detection of sawtooth crashes has been developed. The algorithm is based on time-scale wavelet theory and edge-detection. The performance of the algorithm is tested and compared with other methods, such as the single band-pass filter. Detection of ‘standard’ sawtooth crashes was demonstrated to have considerably less delay than a single band-pass filter, which needs to detect all crashes accurately. It is also shown that more challenging sawtooth crashes can be handled, as a consequence of the robust scales. The choice of a general edge detector instead of an optimized sawtooth detector is justified by the existence of many different types of sawtooth and the realized accuracy of the detection algorithm, which is well below the uncertainty of the sawtooth period itself for most crashes. Moreover, multiresolution enables distinction between different sizes of sawtooth crashes due to the different sizes of wavelets.

Tests show that a flawed period estimation algorithm based on edge detection due to the dynamical behavior of the plasma is not possible. This complicates the selection of an optimal threshold. Nevertheless, the algorithm simplifies the selection of a proper threshold as a consequence of the better SNR of wavelet scales ($m \geq 1$) (band-pass filters) compared to the single band-pass filter based algorithms presented in [14], [13]. Further improvements of the algorithm are possible by combining ECE-channels.

Therefore, the current algorithm is the intermediate step for period estimation between a very robust detection algorithm based on the Morlet Wavelet, which is difficult to implement in real-time and the current real-time implemented simple one band-pass filter methods. Due to the calculation of at least two scales, the algorithm allows an immediate and accurate estimate of the crash position in time. Numerically, the proposed algorithm has a high flexibility e.g. the step size $s_0$ in scales $s$ can be varied due to the B-spline implementation, and the initial scale can be chosen by varying the pre-filter. In addition, the wavelets are calculated with a filter bank, which filters are (anti-)symmetric and contain only a few non-zero coefficients, which makes the implementation very efficient and consequently real-time applicable as demonstrated in a Simulink® implementation.

APPENDIX

The filter coefficients used are given in Table I, the other parameters are given in Table II.

Table I: Filter coefficients that generate the cubic B-spline scaling functions and its derivative.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$L(z)$</th>
<th>$L'(z)$</th>
<th>$\Phi_0(z)$</th>
<th>$H(z)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0625</td>
<td>0.0625</td>
<td>0.0029</td>
<td>-0.5</td>
</tr>
<tr>
<td>1</td>
<td>0.2500</td>
<td>0</td>
<td>0.0234</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>0.3750</td>
<td>0.2500</td>
<td>0.0786</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.2500</td>
<td>0</td>
<td>0.1637</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.0625</td>
<td>0.3750</td>
<td>0.2314</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0.2314</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.2500</td>
<td>0</td>
<td>0.1637</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0.0786</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.0625</td>
<td>0</td>
<td>0.0234</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td>0.0029</td>
<td></td>
</tr>
</tbody>
</table>

Table II: Parameters used to generate figures

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increase threshold based on noise $m = 1$ by factor</td>
<td>2</td>
</tr>
<tr>
<td>Channels</td>
<td>2</td>
</tr>
<tr>
<td>Number of channels in which crash is detected such that it is forwarded</td>
<td></td>
</tr>
<tr>
<td>Block Time</td>
<td>2.43 ms</td>
</tr>
<tr>
<td>Time detection is blocked after crash: maximum delay 1.63 ms + 0.2 ms</td>
<td></td>
</tr>
<tr>
<td>Distance between maxima</td>
<td>[2 2 2 2 5 5 5]</td>
</tr>
<tr>
<td>Maximum distance between maxima if bigger, chain is broken</td>
<td></td>
</tr>
<tr>
<td>Buffer size</td>
<td>2.43 ms</td>
</tr>
<tr>
<td>Storage of maxima in a matrix [Boolean]</td>
<td></td>
</tr>
<tr>
<td>Delay</td>
<td>0.08, 0.13, 0.23, 0.43, 0.83, 1.63 ms</td>
</tr>
<tr>
<td>Delay of detection compared to the crash time ($m = 8, 93.29, 6.43 ms$)</td>
<td></td>
</tr>
</tbody>
</table>

REFERENCES


Future Research

The edge detector (wavelet) can be further optimized to increase the magnitude of the resulting wavelet coefficients, making it somewhat easier to detect (and in an earlier stadium) the sawtooth crashes. Although optimization of the wavelets to improve accuracy of the algorithm is not necessary.

The threshold remains the Achilles’ heel of the algorithm, therefore it is important to further investigate a suitable threshold for individual situations. Another option is to make the threshold choice less sensitive by for instance taking the square of the wavelet coefficients.

The algorithm is preferably implemented on a FPGA when used real-time. Especially the number of scales should be chosen with care. In the case that still extra calculation power is necessary it is also possible to add downsampling into the current algorithm but at the cost of delay.

The combination of channels needs to be further improved. An alternative is to observe all crashes from the different channels and decide on basis of an intelligent algorithm comparing height and time instant of the crash, if it belongs to the period of the sawtooth signal. Furthermore noise needs to be further reduced by cross-correlating the different signals due to the uncorrelation of the noise between different channels but at the cost of some delay.

Currently the direction of the sawteeth is based on the maxima and minima (observed in a window) of the response on a crash, this generally works well because the crash gives a clear directional difference. However, the threshold is preferably based on the sign of the sawteeth due to the negative response of the wavelet coefficients from big wavelets. Therefore, especially in regimes where the inversion radius changes position between channels within a few periods, it is difficult to track. Consequently, making it difficult to track the sign of the sawtooth and find a threshold in the right direction. Although for small wavelets the threshold can be taken two-sided. Therefore it is important to develop an algorithm that tracks the inversion radius real-time independently from the crashes.
Appendices
Appendix A

Fast Wavelet Transform in the frequency domain

A.1 Properties of the Fourier Transform

The following properties of the Fourier Transform are used:

\[ f(at) = \frac{1}{|a|} \hat{f} \left( \frac{\omega}{a} \right) \quad (A.1) \]
\[ f(x - a) = e^{-i\omega a} \hat{f}(\omega) \quad (A.2) \]
\[ \frac{d^n f(x)}{dx^n} = (i\omega)^n \hat{f}(\omega) \quad (A.3) \]
\[ (f \ast g)(x) = \hat{f}(\omega)\hat{g}(\omega) \quad (A.4) \]

A.2 Wavelet and scaling functions time domain

The wavelet transform and scaling function with \( s_0 = 2 \) are defined as:

\[ d_m[n] = \langle f, \psi_{m,n} \rangle = \int_{-\infty}^{+\infty} f(t) s_0^{-m/2} \psi_s(t - n) dt. \quad (A.5) \]
\[ a_m[n] = \langle f, \phi_{m,n} \rangle = \int_{-\infty}^{+\infty} f(t) s_0^{-m/2} \phi_s(t - n) dt. \quad (A.6) \]

The DWT can be implemented in a Fast Wavelet Transform (FWT) if the scaling function has the following two properties: \( \phi(t) \) can be written as a summation of different translations of \( \phi_m \left( \frac{t}{2} \right) \), and \( \psi \left( \frac{t}{2} \right) \) can be written as a summation of different translations of \( \phi(t) \):

\[ \phi_m \left( \frac{t}{2} \right) = \sqrt{2} \sum_{n=-\infty}^{\infty} l[-n] \phi(t - n) \quad \text{and,} \]
\[ \psi \left( \frac{t}{2} \right) = \sqrt{2} \sum_{n=-\infty}^{\infty} h[-n] \phi(t - n), \quad (A.7) \]
A.3 Wavelet and scaling functions in frequency domain

The Fourier transform of (A.5) and (A.6) are calculated with (B.1) and (A.4):
\[ \hat{d}_m = 2^{m/2} \hat{f}(\omega) \hat{\psi}(2^m \omega), \text{ and} \]
\[ \hat{a}_m = 2^{m/2} \hat{f}(\omega) \hat{\phi}(2^m \omega) \]
(A.9)

The Fourier transforms of (A.7) and (A.8) are given by (see also Appendix C [1]):
\[ \hat{\phi}(2\omega) = \frac{1}{\sqrt{2}} L(\omega) \hat{\phi}(\omega) \]
(A.11)
\[ \hat{\psi}(2\omega) = \frac{1}{\sqrt{2}} H(\omega) \hat{\phi}(\omega) \]
(A.12)

It becomes clear that \( L(\omega) \) describes the transfer from \( \hat{\phi}(\omega) \) to \( \hat{\phi}(2\omega) \).

A.4 Fast Wavelet Transform in frequency domain

Now, all the tools to construct a fast wavelet transform in the frequency domain are available. For instance we wish to calculate the wavelet coefficients on scale \( m = 3 \), therefore (A.9) needs to be changed:
\[ \hat{d}_3 = 2\sqrt{2} \hat{\psi}(8\omega) \hat{f}(\omega) \]
(A.13)

which, can be rewritten using (A.10) and (A.9) in a different form:
\[ \hat{\phi}(4\omega) = 2^{-0.5} L(2\omega) \hat{\phi}(2\omega), \]
(A.14)
\[ \hat{\psi}(8\omega) = 2^{-0.5} H(4\omega) \hat{\phi}(4\omega). \]
(A.15)

Filling this into (A.13) we find:
\[ \hat{d}_3 = 2 \cdot H(4\omega) \hat{\phi}(4\omega) \hat{f}(\omega) = \sqrt{2} \cdot H(4\omega) L(2\omega) \hat{\phi}(2\omega) \hat{f}(\omega), \]
(A.16)
together with (A.14):
\[ \hat{d}_3 = H(4\omega) L(2\omega) \hat{\phi}(\omega) \hat{f}(\omega). \]
(A.17)

Let’s compare this to that of \( m = 2 \):
\[ \hat{d}_2 = H(2\omega) \hat{\phi}(\omega) \hat{f}(\omega). \]
(A.18)

In addition the scaling coefficients \( \hat{a}_0, \hat{a}_1 \) and \( \hat{a}_2 \) are defined as:
\[ \hat{a}_0 = \hat{f}(\omega) \hat{\phi}(\omega), \ \hat{a}_1 = \sqrt{2} \cdot \hat{f}(\omega) \hat{\phi}(2\omega), \text{ and} \]
\[ \hat{a}_2 = 2 \cdot \hat{f}(\omega) \hat{\phi}(4\omega), \]
(A.19)

which is also equal to:
\[ \hat{a}_1 = L(\omega) \hat{\phi}(\omega) \hat{f}(\omega) = L(\omega) \hat{a}_0 \]
(A.20)

Now, in combination with \( H(\omega) \) it is possible to calculate from the scaling coefficients the wavelet coefficients.
\[ \hat{d}_1 = H(\omega) L(\omega) \hat{a}_0. \]
(A.21)
Consequently every wavelet coefficient can be calculated by a cascade of filters $L(2^m \omega)$ and $H(\omega)$ and starting scaling coefficients $\hat{a}_0 = \hat{f}(\omega) \hat{\phi}(\omega)$, which is better known as the Filter Bank. However, these are often expressed in terms of $z = e^{j\omega}$ and therefore need to be rewritten. Important to note is that $l[-n]$ and $h[-n]$ in (A.7) and (A.8) are already exact in $z$, as a consequence all calculated wavelet coefficients are exact, if we are dealing with continuous $f(t)$. However, the signal is always discrete and thus the product $\hat{a}_0 = \hat{f}(\omega) \hat{\phi}(\omega)$ needs to be approximated, which has following form in the time domain:

$$a_0[n] = \langle f, \phi_0, n \rangle \approx \sum_{k=-\infty}^{\infty} f[k] \phi_0[k-n],$$

thus introducing an error, dependent on the product. The increasing order has than following form:

$$d_3(z) = H(z^4) L(z^2) L(z) \hat{\phi}(z^{-1}) F(z).$$

### A.5 Samples as scaling coefficients

In the field of wavelet theory it is common practice to fill into the filter bank directly the signal instead of the scaling coefficients, which is purely theoretical wrong as we have seen in the previous section and leads to errors. However, this is only half-true because there is a second mechanism, which partly justifies this choice: this is the convergence of the cascade $L(2^m \omega)$ to the function $\hat{\phi}(a\omega)$. This is explained in more detail in Section 6.5 and appendix C of [1], the convolution is defined in the frequency domain as (no normalization factors considered anymore):

$$\hat{\phi}(\omega) = \prod_{p=1}^{\infty} L \left( \frac{\omega}{2^p} \right).$$

Let’s for instance consider $\hat{a}_4$, which can be calculated by:

$$\hat{a}_4 = \hat{f}(\omega) \hat{\phi}(16\omega) = L(8\omega) L(4\omega) L(2\omega) L(\omega) \hat{\phi}(\omega) \hat{f}(\omega).$$

This means that $\hat{\phi}(16\omega)$ is approximately:

$$\hat{\phi}(16\omega) = \prod_{p=1}^{\infty} L \left( \frac{16\omega}{2^p} \right) \approx L(8\omega) L(4\omega) L(2\omega) L(\omega).$$

Consequently the term $\hat{\phi}(\omega)$ becomes of less importance and justifies the choice of filling in directly the signal samples. However, in the scales where $m$ is still small the approximation is bad and consequently a big error is introduced, which becomes smaller with increasing $m$. Therefore the convergence rate of the cascade to $\phi$ is another important field of study.
Appendix B

Inversion radius

B.1 Theoretical background

The inversion radius $r_{\text{inv}}$ is a measure of the location of the sawtooth instability and describes where $q$ is approximately 1 i.e. it determines where the temperature before a crash and after a crash is the same inside the mixing area. The sawtooth instability results in two extreme temperature profiles, which melt together. Their relationship is given in Fig. B.1.

Fig. B.1(a) presents the half-sided temperature profile before a crash and after a crash. The inversion radius is denoted by $r_{\text{inv}}$, the mixing radius by $r_{\text{mix}}$ and the electron temperature as $T_e$. In Fig. B.1(b) a zoom of Fig. B.1(a) is shown, such that also different measured channels can be represented schematically. The different ECE-channels are related to a certain position inside of the tokamak, depending on the position of the launcher and the magnetic field. Therefore it is possible to determine the temperature on different positions inside the tokamak. The resulting measured signals are the sawtooth temperature signals, which slowly move towards the desired temperature profile but collapses because of the mixing in the center. It becomes clear that the position inside the mixing area where the temperature remains constant is the inversion radius. It can be found by considering the sawtooth signal, which amplitude represents the difference between a mixed temperature profile and the unmixed temperature profile. As a consequence the radius at which the crash amplitude is zero is the location of the inversion radius (Fig. B.1(c)).

An alternative method to determine the inversion radius is based on the phase of the sawtooth signal. The temperature on the left side of the inversion radius decreases when the center is mixed and on the right side it increases leading to a phase difference of 180 degrees when compared or as a difference in sign for a crash detector. Nevertheless this is only between two channels. Consequently if we want to do better it is necessary to calculate the zero

![Figure B.1: Schematic representation of the sawtooth instability [hastie1997]](image-url)
B.2 Temperature estimation

The real-time in-line Electron-Cyclotron-Emission-measurements on TEXTOR are all unscaled. This means that the output voltage of the different channels needs to be related to the temperature. For every channel this relationship needs to be determined individually. Therefore it is necessary that we know the DC Gain, which determines what the gain relationship is between temperature and voltage. This is called the sensitivity \( \frac{\text{eV}}{\text{V}} \). In addition we need to know an initial value. Generally the temperature is determined using Thomson scattering, which needs to be performed prior to the experiment to be able to use it for real-time inversion radius estimation. However under the same conditions the relationship between voltage and temperature remains similar, making it possible to perform this measurement in one of the shots prior to the experiment. This is explained in [2], which also presents an example of Thomson scattering (Fig. B.2)

Using Thomson scattering it is possible to calculate the real temperature output. However, we are only interested in the amplitude of the sawtooth signal. Therefore it suffices to only determine the sensitivity. This can be done as follows: First determine the initial DC gain \( (t = 0 \text{ s}) \), which corresponds to a very low temperature of about 0.025 eV and thus can be neglected. The voltage at 1.2 seconds is determined by the Thomson scattering, which enables us to calculate the sensitivity as follows:

\[
\text{Sens} = \frac{T_{\text{thomson}} - T_0}{V_{\text{thomson, time}} - V_0},
\]

Consequently because we have the full shot including the heating phase, we only need the temperature at one point inside the tokamak.

B.3 Estimate temperature without Thomson scattering

Sometimes we do not have a Thomson scattering measurement available. In that case we need to make an estimate of the temperature based on the known variables. We will need the following variables: the toroidal magnetic field strength \( B_T \); the launcher angle and the Shafronov shift \( S_{\text{shift}} \). The magnetic field determines the relationship between frequency and the radius. This is also dependent on the angle of the launcher. The inverse big radius depends on the \( ECE_R = ECE_{\text{frequency}} \left( \frac{\text{MHz}}{\text{GHz}} \right) \). If the relationship between the channels and the radius is determined, it is possible by estimating a temperature profile, to calculate the temperatures of every channel. However, to estimate the temperature exactly it is necessary to compensate for the Shafronov shift, which determines how much the temperature profile is shifted from
B.4 Example calculation Sensitivity (Shot 106478)

Table B.1: Required parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beta toroidal</td>
<td>$B_T$</td>
<td>2.4 T</td>
</tr>
<tr>
<td>Launcher angle</td>
<td>$\text{Angle}$</td>
<td>0°</td>
</tr>
<tr>
<td>Shafronov shift</td>
<td>$S_{\text{shift}}$</td>
<td>1.81 cm</td>
</tr>
<tr>
<td>Temperature center (estimate)</td>
<td>$T_0$</td>
<td>2200 eV</td>
</tr>
<tr>
<td>Temperature wall (estimate)</td>
<td>$T_a$</td>
<td>100 eV</td>
</tr>
<tr>
<td>Radius</td>
<td>$a$</td>
<td>0.46 m</td>
</tr>
</tbody>
</table>

Figure B.3: Unscaled temperature overview and start of Shot 106478

The center of the tokamak. This Shafronov shift is difficult to determine when the radius is different from zero. Also the radius of the tokamak needs to be known (textor $a = 0.46$), the center and edge temperature needs to be valued ($T_0, T_a$). The temperature profile can now be estimated by considering following approximation:

$$T = T_a + (T_0 - T_a) \left( 1 - \left( \frac{S_{\text{shift}}(r) - r}{a} \right)^2 \right)^2$$  \hspace{1cm} (B.1)

Now we have an estimate of the temperature, which belongs to the temperature profile just before the collapse. This makes it possible to make a rough estimate of the relationship between the different amplitudes of our saw teeth by choosing a point at the top of the temperature. Although this estimate is rough due to the big difference between the temperature at the start of the experiment and the desired temperature, which results in a sensitivity near enough to its real value.

B.4 Example calculation Sensitivity (Shot 106478)

In the last two sections we discussed how we can calculate the temperature using a rough estimate. Logically we wish to apply this on a real measurement. Therefore the following parameters must be known or need to be estimated:

In combination with the start of the real measurement it is possible to find the resulting sensitivity.

In Fig. B.3 we see the entire shot and its start. We know that at the start the temperature is negligible compared to the temperature in operation, hence the input voltages correspond to $T = 0$. These voltages are called the offset. When the temperature reaches its "steady-state" temperature, we can link the voltage to the temperature calculated with (B.1). This leads to a channel dependent sensitivity. For shot 106478 all the intermediate values are presented in Table B.2.

Now we found the sensitivity which tells us the relationship between a voltage difference and temperature difference.
Table B.2: Intermediate values and resulting sensitivity

<table>
<thead>
<tr>
<th>Channel [GHz]</th>
<th>$T_{est}$ [eV]</th>
<th>Offset ($t = 0$) [V]</th>
<th>$V_{ECE}$ ($t = 3.3$ s) [V]</th>
<th>$\Delta V$ [V]</th>
<th>Sens ($T_{est}/\Delta V$) [eV/V]</th>
</tr>
</thead>
<tbody>
<tr>
<td>132.5</td>
<td>2181</td>
<td>0.63</td>
<td>6.00</td>
<td>5.37</td>
<td>406.15</td>
</tr>
<tr>
<td>135.5</td>
<td>2104</td>
<td>0.57</td>
<td>5.60</td>
<td>5.03</td>
<td>418.29</td>
</tr>
<tr>
<td>138.5</td>
<td>1977</td>
<td>0.75</td>
<td>7.10</td>
<td>6.36</td>
<td>310.85</td>
</tr>
<tr>
<td>141.5</td>
<td>1811</td>
<td>0.77</td>
<td>7.35</td>
<td>6.59</td>
<td>274.81</td>
</tr>
<tr>
<td>144.5</td>
<td>1617</td>
<td>1.85</td>
<td>3.64</td>
<td>1.79</td>
<td>903.35</td>
</tr>
</tbody>
</table>

B.5 Amplitude versus wavelet coefficients

Next step to calculate the inversion radius is determining the sawtooth’s signal amplitude, which still is unknown. The best way is to determine the minimum and maximum of the sawtooth signal (in eV) and calculate the resulting amplitude. However, this is not an easy process to do real-time. Another possibility is to use frequency methods such as the Complex Morlet Wavelet to determine an amplitude and phase but which is averaged. Both these methods expand the currently used algorithm for period detection considerably. Therefore it is chosen to use existing data-streams to make an estimate of the different amplitudes. From the real-time wavelet analysis we already have calculated the wavelet coefficients, which already describe the direction of the crash. Their response is biggest in the highest scale $m$ and we know also that the magnitude of the wavelet coefficient is a measure for the depth of the crash i.e. the amplitude. Although it could be argued that they are not really linear dependent. Nevertheless we assume the magnitude of the wavelet coefficient is a good measure for the amplitude. It should be remembered that the wavelet coefficients only describe the amplitude near a crash. We know that the highest scale wavelet coefficients are delayed more than the detected value (threshold is lower). Therefore we just need to calculate the moment of a real crash and add the delay and pick those values as our measure of amplitude. This is presented in Fig. B.4.

Figure B.4: Wavelet coefficients different channels biggest scale $m$
B.6 Polynomial fit

Now we know in principle everything to determine the inversion radius. However, because the limited number of channels we only have an estimate on 5 points inside the tokamak. Ideally we would have also a description of the difference of temperature profiles presented in Fig. B.1. Consequently we are left with fitting the values through a polynomial. The values near the inversion radius have very low amplitude and have importance because they show best where the inversion radius is located. Unfortunately these channels are also very noisy and small SNR making it difficult to estimate the amplitude well. On the other hand the channels far from the inversion radius describe the location of the inversion radius badly but have a much better estimate of the amplitude. Therefore is chosen to fit a polynomial, which should incorporate all points. This results in a fourth order polynomial, although it is probably not the ideal solution. The lack comparable information on the inversion radius makes it difficult to make a good choice. The last step that needs to be taken is to find the zero of this polynomial. This needs to be done efficient and accurately. Therefore the zeros of this polyfit are calculated using roots and the zero is chosen between the two channels where the wavelet coefficients switch signs. Now it is possible to determine the inversion radius.

B.7 Results

In the described shot the inversion radius remains almost constant over the entire region of interest. Only at the end a clear change can be noted looking at the direction of the crash. Although not optimal it is possible to have an higher resolution than only between channels. Consequently it is also possible to track the drift of the inversion radius. This is shown in Fig. B.5. The fourth order fit does perform quite acceptable. To further reduce this wobbling of the inversion radius estimate we take also in account the 4 previous estimates. Then we dispense the biggest and smallest value and calculate the mean over 3 samples. This results in an acceptable measure for the inversion radius but is delayed.

Figure B.5: Averaged inversion radius calculated using 4th order fit

B.8 Summary

The three main steps necessary to estimate the inversion radius are: first to find the relationship between voltage and temperature. Then to compensate the signals with the sensitivity, such that the wavelet coefficients describe the real amplitudes at least compared to each-other near a crash. Finally to find the zero of the function, which describes these amplitudes in relationship to the radius.
Appendix C

Tuning and practical considerations

In this appendix the different tuning parameters are discussed and some practical implementation suggestions are made.

Block-time

The block-time is the time that the wavelet coefficients are suppressed after detection has taken place. This is done to prevent multiple detection on the same crash. This block-time is related to the size of the wavelet. Therefore it is chosen as the delay of the biggest wavelet added by some extra time. This extra time should be chosen such that the wavelet coefficients are below the threshold again for the responses on the biggest wavelet. It also needs to be taken into account that when wavelets are near the size of twice the period, the next crash could be blocked accidentally when the same block-time applies to all scales (currently implemented as such). This can be partially solved by making the block-time scale dependent. Then the small scales can detect new crashes in an earlier stadium again.

Buffer size

The buffer stores the maxima of the responses. This in the form of logicals (0-1) such that it is implemented efficiently. The buffer-size can be chosen as big as one wants the minimum size should be about the maximum delay of the biggest wavelet (preferably somewhat bigger).

Prefilter

The prefilter should be chosen such that crashes can generally be recognized but not such that they become very big. This is because the first scale is used to find the position of the crash and should not be used for detection. In addition if the threshold is coupled to the first scale, high responses will lead to a too high threshold. Although tuning in the frequency domain is a possibility, it is more difficult because the crash stands in relationship to the noise. Consequently, the Fourier transform should be calculated carefully such that it does not averages out the noise.

Threshold

In the simulink algorithm two possible choices of the threshold are implemented. This is the threshold purely based on the noise and the one based on the first scale of the wavelet coefficients (used in this thesis). The first, purely based on the noise works well in clean regimes with not to much dynamical behavior. However, the underbound $T_{\text{min}}$ is somewhat to low because not only noise is present. A factor 3-4 generally works well, however, in channels near the inversion radius, where the noise is much more significant a somewhat lower threshold is necessary.

The threshold based on the wavelet coefficients of the first scales is more robust for regimes with significant dynamical behavior because the wavelet coefficients react on small crashes such
that they are taken into account in the threshold. This also means that the threshold becomes
dependent on the choice of the prefilter, which determines the size of the wavelet at the first
scale (factor 1.5-2.5). Both the thresholds are currently based on a clock basis (few periods).
This has the advantage compared to a crash based reset if crashes are missed, the threshold is
automatically re-adjusted. It is important to note that to prevent instant loss of the threshold
by activation and shutting down the NBI system due to the huge edges. Therefore during these
events threshold updates need to be suppressed temporarily. The threshold can be improved
by implementing a continuous changing threshold (using FIR-filter).

**Distance between maxima**

The distance between maxima is used to prevent the maxima chain to wrong maxima. It
should be chosen tight when dealing with strange crashes. In normal regimes it can be chosen
somewhat less tight. Although generally the maxima are situated within a few samples from
each other anyway. At the bigger scales, maxima are far apart, therefore the maximum distance
between maxima should be somewhat bigger for big wavelets.

**Channels**

This part of the algorithm needs further improvement. However, as implemented now, the
choice of the number of channels in which a crash is detected depends on the threshold. If
chosen for a robust threshold the channels in which detection takes place can be 2-3. However,
if a less robust threshold (less-delay) is chosen this should be about 4. It is important to also
consider the reset time of falsely detected crashes.
Bibliography
