Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **500,** 2969–2978 (2021) doi:10.1093/mnras/staa2551

Radio frequency interference mitigation based on the asymmetrically reweighted penalized least squares and SumThreshold method

Qingguo Zeng,[1] Xue Chen,[2,3] Xiangru Li,[4★] J. L. Han,[2,3,5] Chen Wang,[2,3,5] D. J. Zhou[2,3] and Tao Wang[2,3]

[1]*School of Mathematical Sciences, South China Normal University, No. 55 West of Yat-sen Avenue, Guangzhou 510631, China*
[2]*National Astronomical Observatories, Chinese Academy of Sciences, Jia20 Datun Road, Beijing 100012, China*
[3]*Astronomy School, University of Chinese Academy of Sciences, No.19(A) Yuquan Road, Shijingshan District, Beijing 100049, China*
[4]*School of Computer Science, South China Normal University, No. 55 West of Yat-sen Avenue, Guangzhou 510631, China*
[5]*The FAST Key Laboratory, Chinese Academy of Sciences, Jia20 Datun Road, Beijing 100012, China*

Accepted 2020 August 17. Received 2020 August 12; in original form 2020 March 18

ABSTRACT

As radio telescopes become more sensitive, radio frequency interference (RFI) is becoming more important for interesting signals of radio astronomy. There is a demand for developing an automatic, accurate and efficient RFI mitigation method. Therefore, we have investigated an RFI detection algorithm. First, we introduce an asymmetrically reweighted penalized least squares (ArPLS) method to estimate the baseline more accurately. After removing the estimated baseline, several novel strategies were proposed based on the SumThreshold algorithm for detecting different types of RFI. The threshold parameter in SumThreshold can be determined automatically and adaptively. The adaptiveness is essential for reducing human intervention and for the online RFI processing pipeline. Applications to data from the Five-hundred-meter Aperture Spherical Telescope (FAST) show that the proposed scheme based on ArPLS and SumThreshold is superior to some typically available methods for RFI detection with respect to efficiency and performance.

Key words: methods: data analysis – pulsars: general.

1 INTRODUCTION

In radio astronomy, radio frequency interference (RFI) is becoming more important for radio observational facilities. RFI has always influenced the search for and analysis of interesting astronomical objects. Mitigating RFI has become an essential procedure in pulsar survey data processing. The Five-hundred-meter Aperture Spherical Telescope (FAST) is an extremely sensitive radio telescope formally in operation in 2020 January. It is necessary to find an effective and precise RFI mitigation method for FAST data processing.

Available RFI mitigation methods can be divided into three categories based on their principles (Akeret et al. 2017b). The first category consists of linear methods, such as singular vector decomposition (SVD; Offringa et al. 2010), principle component analysis (PCA) and their variants (e.g. Zhao, Zou & Weng 2013). In practice, these methods are not suitable for dealing with frequency-varying RFI (Offringa et al. 2010). The widespread use of radio sources in everyday life causes a diversity of RFI. The diverse contamination of RFI makes it difficult to model RFI using these linear methods. The second category consists of machine learning algorithms that can automatically learn the discriminating features between RFI and non-RFI (Akeret et al. 2017b; Mosiane et al. 2017; Kerrigan et al. 2019). One typical limitation of this type of method is that these methods need a set of observations with labels, which are time-consuming to obtain. The last category consists of thresholding methods, which are widely used in the available RFI mitigation pipelines because of their simplicity and effectiveness. One typical thresholding method is simple thresholding (Schoemaker 2015), which flags a pixel as RFI when its intensity is larger than a preset parameter (called the threshold). The advantages of this method are its simplicity and high efficiency. However, this method is sensitive to noise because it is dependent on the comparison of single pixels. To overcome this limitation, Offringa et al. (2010) introduced an RFI detection algorithm, SumThreshold, based on computing the combined effects of some adjacent pixels. The SumThreshold method has been wrapped in the RFI detection pipeline for the Low Frequency Array (LOFAR), e-MERLIN (Peck & Fenech 2013), the Bleien Radio Observatory (Akeret et al. 2017a), etc.

In the thresholding methods for RFI detection, a fundamental assumption is that the intensities of the data should be constant in the absence of interference (Winkel, Kerp & Stanko 2007). However, almost all astronomical data do not fit this assumption because of the presence of the inconsistency of receiver response and background information. This type of inconsistency has some negative effects on RFI detection and can be approximately described using a smooth surface (referred to as the baseline; Winkel et al. 2007). The baseline should be accurately estimated and removed from data. To do this, Winkel et al. (2007) propose a scheme to describe the baseline using a two-dimensional (2D), low-order polynomial, whereas Offringa et al. (2010) propose a baseline fitting scheme based on a sliding window and some weighted Gaussian filters. However, it is shown that the accuracy of these baseline estimations can be affected by broad-band RFI.

★ E-mail: xiangru.li@gmail.com

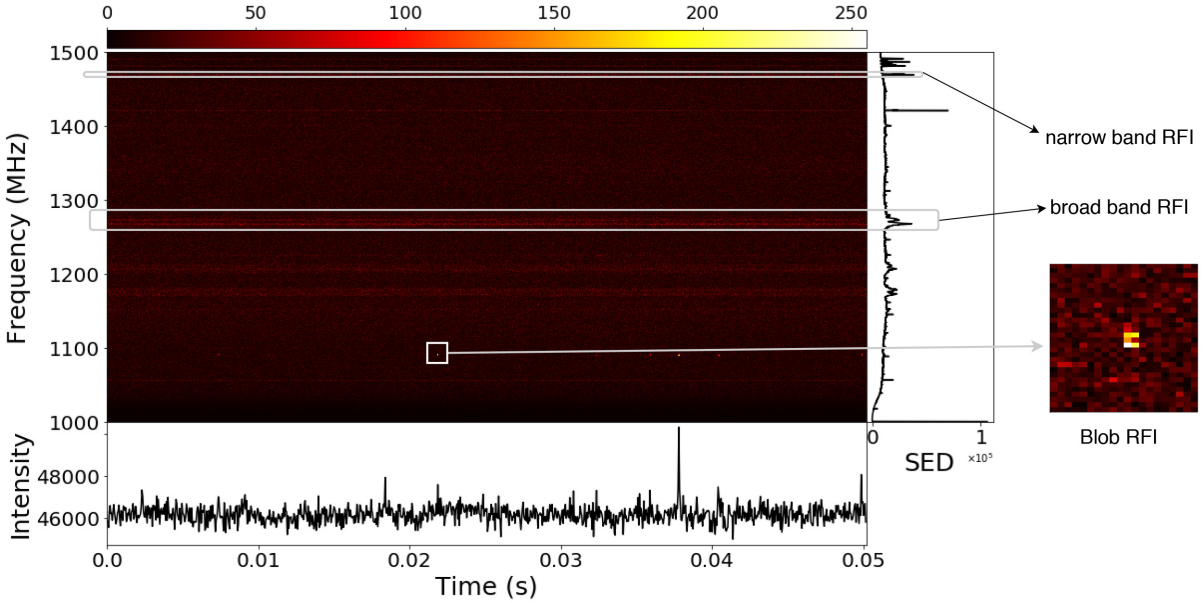


Figure 1. A time–frequency image of FAST data for 0.05 s in the frequency range from 1000 to 1500 MHz with an SED curve on the right, a frequency-integral curve in the lower panel, and a zoomed view of the RFI region as the main panel. A narrow-band RFI and a broad-band RFI can be identified from the peaks on the SED curve. The blob RFI contaminates only a small ratio of pixels and cannot be identified based on the SED curve. The colour bar of the image is shown at the top of the time–frequency image.

Therefore, we proposed a baseline fitting method based on an asymmetrically reweighted penalized least squares algorithm (ArPLS; Baek et al. 2015). The penalized constraint in this method makes the baseline fitting more robust and accurate than traditional methods, by mitigating the negative influences from instrumental response. The baseline is estimated from a time-integral curve/spectral energy distribution (SED) curve (a one-dimensional vector), while the traditional method is carried out using a time–frequency image (Offringa 2012). Therefore, this ArPLS-based method is more efficient.

For flagging the RFI, we propose several strategies based on the SumThreshold algorithm. Not only can these strategies detect the traditional band RFI more efficiently, but they can also more accurately detect blob RFI, a short and small-bandwidth interference typically covering nearly 100 μs and a bandwidth of less than 1 Hz.

This paper is organized as follows. The experimental FAST data and their characteristics are described in Section 2. In Section 3, we present the proposed baseline fitting method and the strategies to detect the RFI. We present the application of the proposed scheme to FAST data and discuss the results in Section 4.

2 EXPERIMENTAL DATA AND THEIR CHARACTERISTICS

The proposed RFI mitigation scheme is tested on FAST observations. These data are sampled at a time resolution of 4.9152×10^{-5} s on 4096 frequency channels. The size of each time–frequency image is 4096×1024 pixels, where 1024 is the number of sampling points per frequency channel within one subintegration.

The data set consists of 100 time–frequency images (subintegrations), which can be taken for any beam in different areas of the sky and at different observation times by the 19-beam receiver. The diversity of the RFI and baseline guarantee the objectiveness of the performance evaluation on the proposed scheme. To design the RFI mitigation scheme to be as efficient and accurate as possible, it is necessary to investigate the characteristics of RFI on the FAST data.

The pulsar search observations of FAST take a wavelength range from 1000 to 1500 MHz and a frequency resolution of 122.07 KHz (Jiang et al. 2020). In FAST observations, there are mainly two types of RFI: band RFI and blob RFI (Fig. 1). The band RFI is likely to be generated by television broadcasts, mobile communication and radar. The blob RFI is a short, small-bandwidth signal from unknown sources. Suppose $s(t, f)$ represents the input 'Data' in Fig. 2, where t represents time and f is the frequency. The SED is computed by aggregating the energies along the time axis $\mathrm{SED}(f) = \sum_t s(t, f)/n_t$ (Fig. 1), where n_t is the number of pixels per frequency channel in one subintegration. The band RFI occupies one or several frequency channels with a time duration of almost the whole subintegration, whereas the blob RFI just contaminates several pixels. Fig. 1 shows one typical FAST observation and the corresponding SED curve. In practice, there may be more than one peak on one SED segment contaminated by one frequency-varying band of RFI (Jiang et al. 2020).

3 THE PROPOSED SCHEME

Based on the characteristics of the RFI in the FAST data, we propose a novel RFI mitigation scheme. This RFI mitigation scheme is designed based on the two main parts: ArPLS and the SumThreshold algorithm (for convenience, this scheme is referred to as ArPLS-ST). A flowchart of the ArPLS-ST is presented in Fig. 2. The core procedures are 'Baseline fitting and removal on SED', 'SumThreshold for band RFI detection', 'Baseline removal on image' and 'Blob RFI detection'. For fitting the baseline, we introduce the ArPLS method. In the procedures 'SumThreshold for band RFI detection' and 'Blob RFI detection', several novel strategies based on the ST algorithm are applied to detect different types of RFI. Besides, the threshold in ArPLS-ST can be automatically determined by a generalized PauTa criterion (Shen et al. 2017). This automatic parameter setting reduces

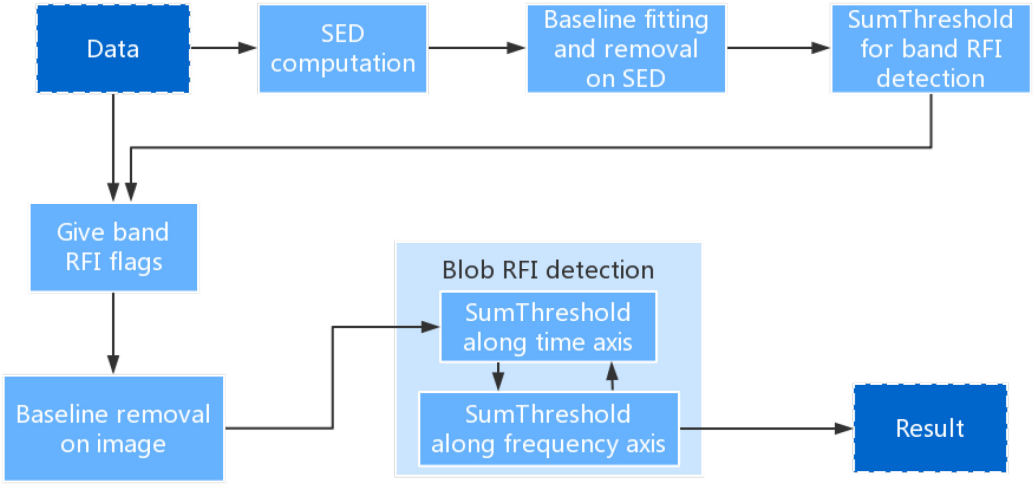


Figure 2. A flowchart of the ArPLS-ST scheme. The 'Data' is an observation of a time–frequency image $s(t,f)$. The baseline fitting and removal are designed to reduce the negative effects on RFI detections from the inconsistency of receiver response and background information.

manual intervention and makes the scheme suitable for an automatic processing pipeline.

3.1 Baseline fitting and removal

Instead of estimating the baseline in the time–frequency image, we propose an estimation from the SED curve using the ArPLS.

3.1.1 The ArPLS for fitting the baseline on the SED curve

A suitable baseline estimation should satisfy two requirements: fitness and smoothness. Let $\boldsymbol{y} \in R^D$ denote the data being processed and let $\boldsymbol{z} \in R^D$ be the estimated baseline of $\boldsymbol{y}$, where D is a positive integer and denotes the number of sampling points along the frequency axis. In this work, $\boldsymbol{y}$ represents the SED curve of an observation (a subintegration from FAST). The constraint 'fitness' ensures that the estimated baseline $\boldsymbol{z}$ precisely describes the information of the original signal $\boldsymbol{y}$ within interference-free regions, while 'smoothness' ensures the estimated baseline is not influenced by the RFI. Consequently, the optimal estimation of $\boldsymbol{z}$ can be obtained by minimizing the following weighted penalized least squares function (Eilers 2003; Cobas et al. 2006; Zhang et al. 2010; Baek et al. 2015)

$$S(\boldsymbol{z}) = (\boldsymbol{y} - \boldsymbol{z})^{\top}\mathbf{W}(\boldsymbol{y} - \boldsymbol{z}) + \lambda \boldsymbol{z}^{\top}\mathbf{M}^{\top}\mathbf{M}\boldsymbol{z}, \quad (1)$$

where $\mathbf{W}$ is a diagonal matrix with its diagonal element $\boldsymbol{w}_i \geq 0$ representing the weight corresponding to the square difference $(\boldsymbol{y}_i - \boldsymbol{z}_i)^2$, $i = 1, \ldots, D$; $\mathbf{M}$ is a $D \times D$ matrix. Actually, $\mathbf{M}$ is a second-order difference matrix, which is considered to be a natural way to express the roughness in mathematics (Ramsay & Silverman 2007). Besides, λ is a preset coefficient that controls the balance between fitness and smoothness. Ideally, $\boldsymbol{w}_i$ should be set to a value of almost 0 for the pixels in the peak regions contaminated by RFI and nearly 1 for the pixels outside these regions. Unfortunately, these peak regions remain unknown for a given observation and it is difficult and time-consuming to locate them in application (Andreev et al. 2003; Jirasek et al. 2004). Baek et al. (2015) proposed an iterative weighting procedure to obtain the optimal estimation of $\boldsymbol{z}$ and $\mathbf{W}$ without peak searching. This iterative weighting procedure is referred to as the ArPLS algorithm.

3.1.2 Baseline fitting and removal on SED curve

The SED curve can be divided into three parts according to their RFI-contamination characteristics: interference-free regions, narrow-band RFI regions and broad-band RFI regions. In the interference-free regions, the SED curve is smooth, although the band RFI causes some dramatic fluctuations (Fig. 1). There are sometimes multiple peaks within one protuberance in the regions contaminated with some broad-band RFI, which inevitably cause some difficulties in baseline fitting.

It is shown that ArPLS can quickly converge in the interference-free regions and narrow-band RFI regions (Fig. 3a). In the broad-band RFI regions, although the ArPLS converges relatively slowly, experiments show that it is still capable of giving a reasonable estimation for the baseline after several more iterations (Fig. 3a). To our knowledge, the typical baseline fitting methods used in pulsar data processing are the tile-based polynomial fitting (TPF; Winkel et al. 2007) and the weighted Gaussian filter (GF; Offringa et al. 2010). It is shown that both the TPF and GF work well in the interference-free regions and narrow-band RFI regions (Figs 3b and c). However, the baselines fitted by them tend to be raised up by the peaks within broad RFI regions. Furthermore, the TPF method performs poorly near the edges of each tile because of the boundary effects of the polynomial fitting, especially when the edge is in the peak regions.

The most significant difference between these two methods and the proposed ArPLS is that the ArPLS can fit the baseline directly by tolerating the RFI in the data, while the TPF and GF couple the RFI removal and baseline fitting because of their sensitivity to RFI. When the sharp peaks are marked as candidate RFI regions by the TPF and GF, the pixels within these regions are discarded. However, this discarding makes it difficult for these methods to accurately estimate a smooth baseline and to judge whether the regions between the marked regions are contaminated with any relatively weak RFI or not. Therefore, the TPF and GF often fail to detect some relatively weak RFI in the regions between two strong peaks in the broad-band RFI regions (Fig. 3b and c).

Actually, the smoothness of the baseline estimated by the proposed scheme is controlled by the second term of equation (1). This constraint is implemented using a second-order difference and adapted to the radio observations in the iterative estimation procedures. In

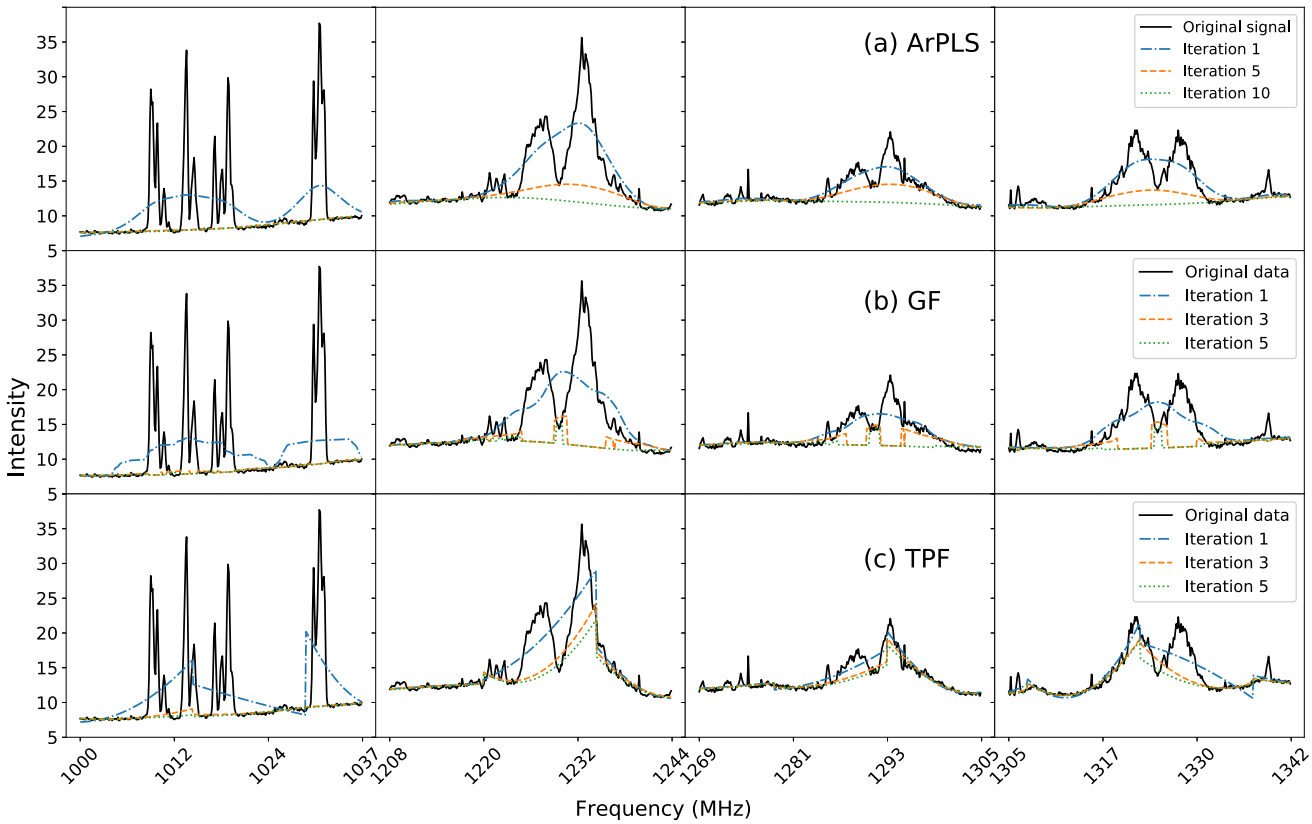


Figure 3. Examples of baseline fitting, using three baseline fitting methods (ArPLS, TPE and GF), to the observation data in the frequency ranges 1000–1037, 1208–1244, 1269–1305 and 1305–1342 MHz. In these examples, there exist several narrow-band and broad-band RFI regions. The convergence characteristics of the iterative process are shown for several iteration steps.

Table 1. The average execution times of the baseline fitting methods for one FAST time–frequency image. These are computed by running every method 10 times.

Method	Execution time
ArPLS	27.4 ± 0.4 ms
Weighted Gaussian filter (GF)	32.7 ± 0.4 ms
Tile-based polynomial fitting (TPF)	38.3 ± 0.6 ms

the TPF and GF, however, the smoothness is constrained by the order of the polynomial and the scale parameter, respectively, which are preset based on human experience. Therefore, the proposed ArPLS method is more robust than the TPF and GF.

Although the ArPLS is an iterative algorithm and there are some equations to be solved in each iteration, experiments show that it is still fast enough because the system is sparse and almost all calculations can be implemented in a vectorized style. The running time for the execution of the three baseline fitting methods is presented in Table 1. The ArPLS is the fastest of these three methods, and the other two methods need to be executed several times for every time–frequency image in the FAST application.

3.1.3 Baseline removal on the time–frequency image

To detect blob RFI, the baseline removal should be performed on the time–frequency image. However, it is time-consuming to estimate the baseline directly in a time–frequency joint space because of the large number of observation pixels. Fortunately, it is shown that the spectrum in a subintegration from FAST observation generally tends to be stable on time (Fig. 5). Therefore, the baseline of each spectrum can be approximated by a shared curve theoretically, and this work used the baseline estimated from the SED curve as the shared curve.

Fig. 6 presents the result of baseline removal on a time–frequency image. Compared with the original time–frequency image, the background inconsistency of the processed image is removed excellently and the area with low contrast in the original image becomes easier to distinguish. Therefore, the blob RFI can be identified more accurately using the thresholding algorithms after baseline removal. Meanwhile, this scheme saves computing resources and time by avoiding estimation of the baseline for a time–frequency image from scratch.

3.2 RFI detection based on SumThreshold

After baseline removal, the pixel intensity should be almost constant in the interference-free regions while peaks caused by the RFI still remain and are even more prominent (Figs 4b and 6b). Therefore, we can accurately detect RFI using the SumThreshold method.

The input to SumThreshold is a one-dimensional vector, which is the SED curve in the band RFI detection. For blob RFI detection, the input to SumThreshold is a row or a column of a matrix representing a time–frequency observation after baseline removal. SumThreshold is an iterative algorithm. In each iteration, four

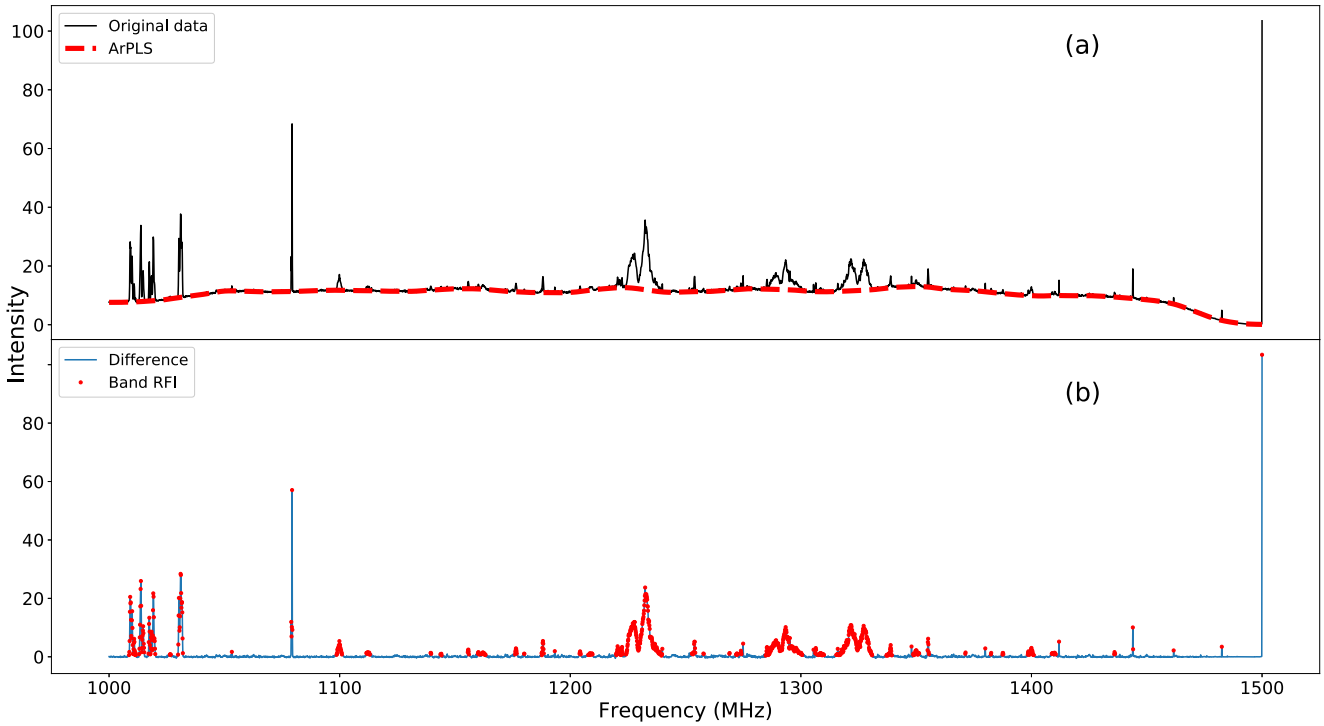


Figure 4. Results of baseline fitting and removal using the ArPLS for the time–frequency image in Fig. 1. (a) An original SED curve (solid line) and the baseline (dashed line) fitted by the ArPLS. (b) The SED curve after removing the estimated baseline (solid line), and the frequency channels corresponding with the detected band RFI (marked as points).

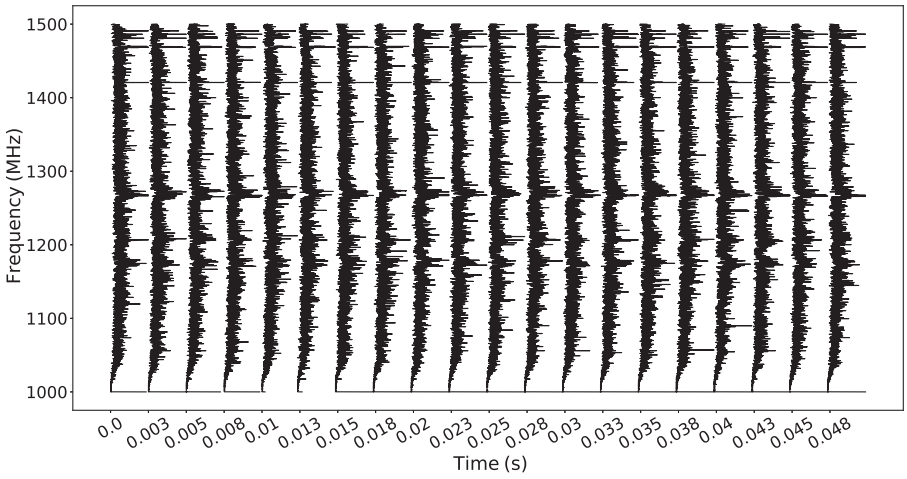
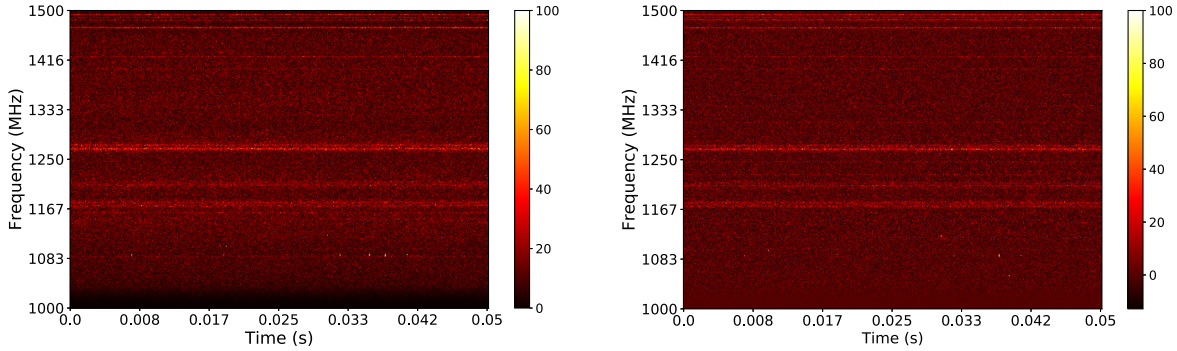


Figure 5. The stability of the SED with time for a set of subintegration from FAST observations. This stability indicates that the baseline of a time–frequency observation from FAST can be estimated using the SED curve.

computational steps are carried out: calculation of the threshold, value replacement, summation, and RFI detection and flagging. The fundamentals of SumThreshold can be found in Offringa et al. (2010). The $K\sigma$ criterion (a variant of the PauTa criterion) is applied to adaptively determine the value of the threshold. Specifically, RFI is detected by checking whether a pixel deviates from the mean more than K times the standard deviation. The $K\sigma$ criterion determines the threshold based on the pixel-value distribution of the input to SumThreshold (Fig. 7). Some excellent investigations on the estimation of standard deviation can be found in Fridman (2008).

3.2.1 Band RFI detection

SumThreshold is subsequently applied to the SED curve for detecting the band RFI and to a time–frequency image for detecting the blob RFI. Actually, the order in which SumThreshold is used for detecting the band RFI or blob RFI does not have any influence on the detection results. However, we can detect RFI more efficiently by using the SumThreshold-based scheme in the order of band RFI first, especially when there is too much band RFI. After that, in detecting blob RFI, SumThreshold does not need to be performed on the regions where band RFI is detected.



(a) An original time-frequency image.

(b) The result after baseline removal.

Figure 6. Comparison of a time–frequency image and the result after baseline removal.

Table 2. Results of six RFI flagging methods evaluated on 100 time–frequency images (see Section 2). Each of the methods – TPF-ST, GF-ST, ArPLS-ST, rfifind, SumThreshold (with SIR) and SumThreshold (without SIR), where SIR is the scale-invariant rank operator – refers to a full pipeline of detecting both band RFI and blob RFI. 'Execution time' consists of the computation time of baseline fitting and RFI detection on one 4096 × 1024 image randomly selected from the 100 time–frequency images (Section 2), and the accuracy, false positive rate (FPR), false negative rate (FNR) and F1 score are computed from all of the 100 images.

Method	Accuracy	FPR	FNR	F1	Execution time
Rfifind	88.54	3.14	55.60	58.22	Not comparable
SumThreshold (with SIR)	82.98	7.98	64.95	48.96	16 900 ± 66 ms
SumThreshold (without SIR)	83.80	2.05	91.25	16.08	16 500 ± 17 ms
TPF-ST	93.51	3.20	23.89	82.63	705 ± 9.35ms
GF-ST	96.60	**1.08**	16.08	87.39	752 ± 17.5ms
ArPLS-ST	**97.95**	1.53	**4.78**	**93.65**	**534 ± 4.5ms**

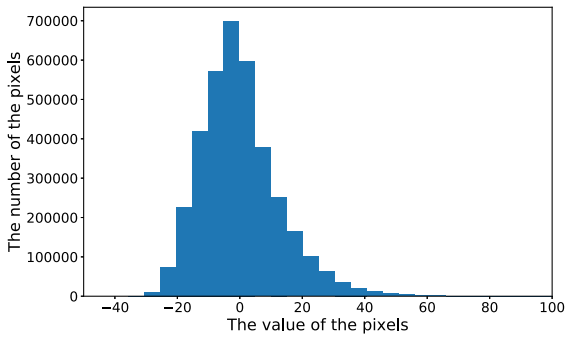


Figure 7. A histogram of pixel values after baseline removal. This histogram can be approximated using a normal distribution. Note that the figure is truncated on value 100 in the *x*-axis to show the most pixels.

To detect the band RFI, SumThreshold is performed on the SED curve after removing the estimated baseline. Experiments show that the protrusions above the horizontal line are detected excellently by the proposed scheme (Fig. 4). These protrusions result from the energy differences between the RFI-contaminated pixels and the interference-free pixels. Therefore, the pixels on the frequency band corresponding with the detected protrusions on the SED curve will be flagged as band RFI .

Actually, there may exist some significant different intensities among the pixels contaminated by band RFI. The differences result from the variation in received power even though the telescope continuously receives interference. This variation may come from several effects, such as intrinsic variation of the interference, change of propagation environment, and instrumental effects (Offringa 2012), etc. These differences in energy can result in leak detection for the RFI mitigation methods directly using the time–frequency image. Therefore, we remove the pixels corresponding to the flagged frequency channels on SED curves. However, this band RFI mitigation method has the potential to bring about some false positives, which can occur between two strong band RFI in case of the uncontaminated pixels covering a small ratio of the area in the subintegration being processed. This ratio should be so small that the corresponding frequency band can trigger the threshold in the SED curve. The probability of occurrence depends on the duration length of a subintegration, and a short duration helps reduce this possibility. Therefore, our experiments show that, for FAST data, this kind of negative possibility is insignificant and acceptable (Table 2).

3.2.2 Blob RFI detection

After removing the estimated baseline from a time–frequency image (Sections 3.1.3 and 3.2.1), we can obtain some results similar to Fig. 6(b). After the band RFI is removed, the results are fed to SumThreshold for blob RFI detection. Blob RFI bursts often exist with a certain duration, in both the time and frequency direction. Therefore, SumThreshold is executed iteratively and alternately along the time and frequency axes in a two2Dage with the detection window increasing from 1 to a preset maximum width. The flagging procedure naturally starts from a large threshold for strong RFI, and then the threshold decreases exponentially. Finally, it outputs the mask indicating the position of the RFI (Fig. 8).

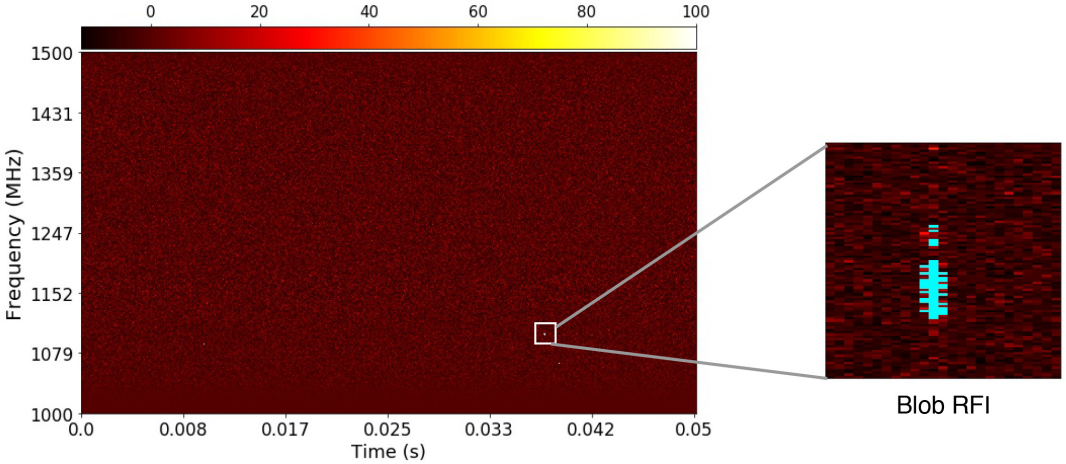


Figure 8. The result of blob RFI detection.

4 APPLICATION TO THE FAST DATA AND DISCUSSION

To investigate the effectiveness of the proposed scheme, some quantitative evaluations and comparisons with several representative methods are conducted on real radio astronomy data (Section 2) for RFI detection. In this section, we first introduce experimental setting, and then present the experimental results.

4.1 Experiment setting

In this experiment, the proposed scheme is compared with five other methods: rfifind from PRESTO;[1] the SEEK[2] Sumthreshold implementation with or without a morphological scale-invariant rank (SIR) operator;[3] a one-dimensional polynomial fitting-SumThreshold (TPF-ST) and one-dimensional Gaussian filter-SumThreshold (GF-ST). Each of the six RFI fagging methods was evaluated on 100 time–frequency images (Section 2). To make the evaluation results fair and not favour any automatic method, the ground-truth labels are generated by marking the RFI manually on the time–frequency images. It is worth noting that the methods are only tested for visually present RFI. The SIR operator is meant to detect RFI samples that are under the noise and invisible. Such samples would be counted as false positives. The performances of these methods are evaluated by accuracy, false positive rate (FPR), false negative rate (FNR) and F1 score. Besides, the implementations of the last four methods are also based on SEEK (a Python library), which makes the execution time of these methods comparable, except for rfifind.

Among the five RFI detection methods, rfifind is a unique method that is not a thresholding method. The operations for RFI detection in TPF-ST and GF-ST are the same as those in ArPLS-ST except for the baseline fitting methods. In the meantime, TPF and GF need to be executed alternatively with thresholding algorithms, because of their sensitivity to the RFI. However, the ArPLS only needs to be executed once before detecting the RFI. These three methods all detect the band RFI on the SED curve and identify the blob RFI on the time–frequency image utilizing the SumThreshold methods. As for the traditional SumThreshold, detection of all types of the RFI is performed on the time–frequency image.

The parameters (e.g. thresholds in SumThreshold, smoothness parameter λ in ArPLS, etc.) that need to be determined are optimized by maximizing the F1 score. This is because the F1 score is capable of measuring the overall performance of the methods when faced with the classification on the imbalanced data. The imbalance refers to the situation when RFI-free pixels are much more than RFI-contaminated pixels. The optimization of the parameters is implemented through a grid search.

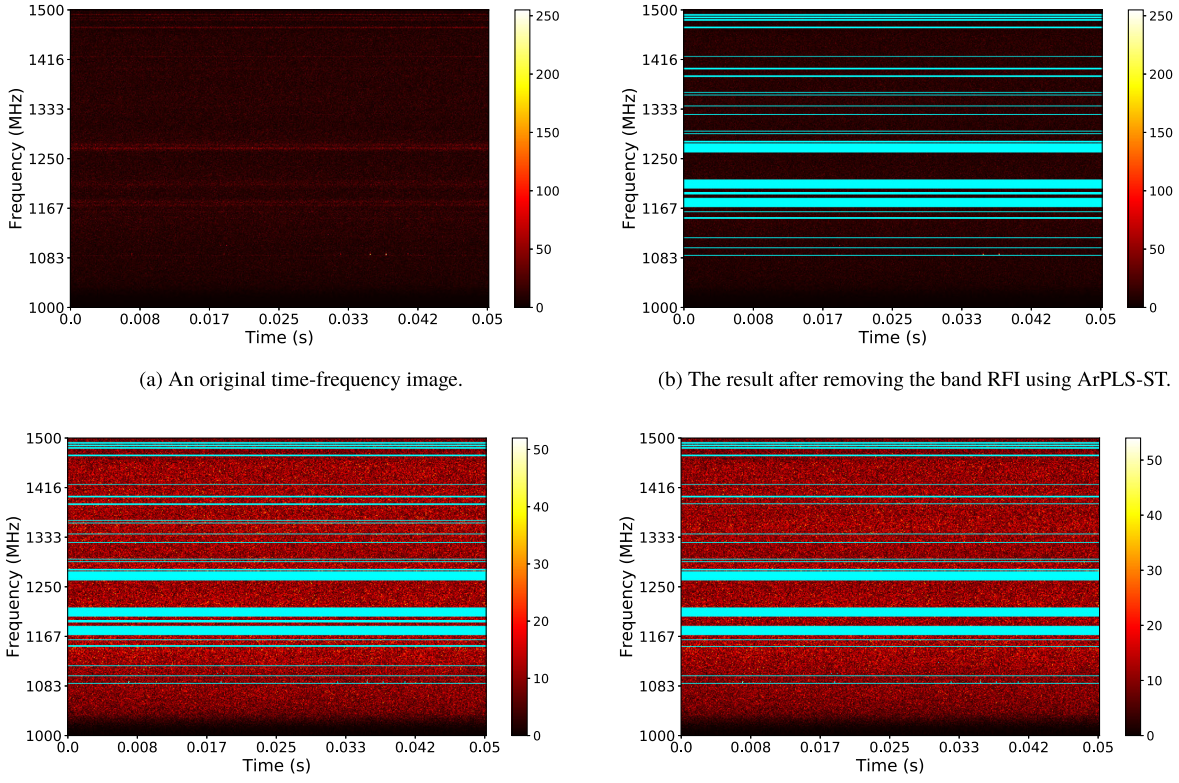
4.2 Experimental results and discussion

The performance metrics of the RFI detection on the FAST data are presented in Table 2. On the whole, ArPLS-ST outperforms the other methods, especially on accuracy, FNR, F1 score and efficiency. The performances of the last three methods (TPF-ST, GF-ST and ArPLS-ST) are better than the traditional SumThreshold.[4] The main difference between the implementations of these three methods, and the traditional SumThreshold is that the traditional SumThreshold fits the baseline and detects the band RFI in a 2D time–frequency image, while the other methods do these on the integration curve SED. The experimental results in Table 2 show that the methods TPF-ST, GF-ST and ArPLS-ST achieve much better performance than the traditional SumThreshold, and indicate the superiority of baseline estimation on the SED curve.

However, the intensities of band RFI are much weaker than those of the blob RFI. Therefore, the thresholds for band RFI performed on SED curves are set smaller than those for blob RFI on the time–frequency images in TPF-ST, GF-ST and ArPLS-ST. In the traditional SumThreshold, there is just one threshold for all types of RFI. To detect the band RFI as much as possible, a small threshold should be chosen. However, this small threshold is likely to result in too many non-contaminated pixels with slightly high intensity that are mistakenly detected as blob RFI. In contrast, a large threshold can bring about leak detections around a detected band RFI, and a high FNR. Therefore, the last three rows in Table 2 show that

[1] https://github.com/scottransom/presto
[2] https://github.com/cosmo-ethz/seek
[3] For convenience, this work uses SumThreshold (with SIR) and SumThreshold (without SIR) as the abbreviations for the cases of the traditional SumThreshold with an SIR operator and the traditional SumThreshold without an SIR operator, respectively.

[4] Traditional SumThreshold refers to the SumThreshold (with SIR) and the SumThreshold (without SIR) introduced by Offringa et al. (2010).



(a) An original time-frequency image.

(b) The result after removing the band RFI using ArPLS-ST.

(c) The result after removing the band RFI and blob RFI using ArPLS-ST.

(d) The mask manually flagged.

Figure 9. An example of FAST data for RFI detection and its labelled mask. (b) The computational results based on the procedures in Section 3.2.2 for band RFI. (c) The result after flagging both band RFI and blob RFI using the ArPLS-ST method. The result in (b) is blacker than in (c) because the blob RFI with higher intensity has not been removed in the former.

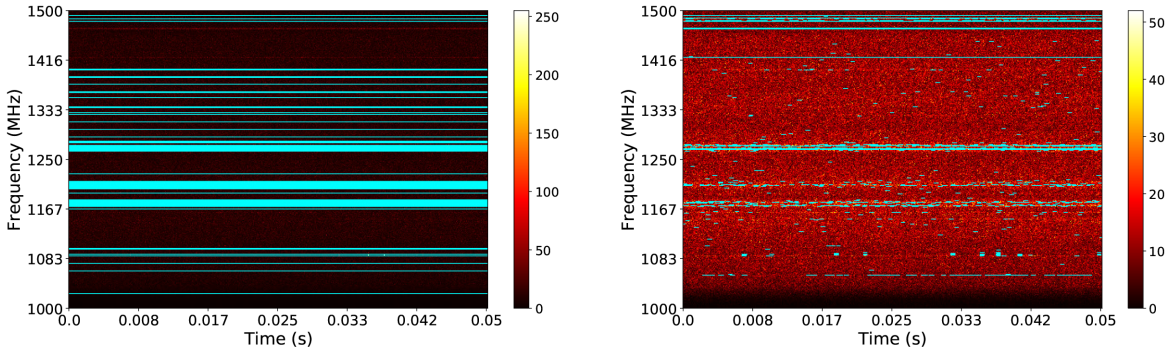
designing different thresholds for different types of RFI is essential to substantially improve the performance of RFI detection.

In addition, the traditional SumThreshold algorithm applies the SIR operator (Offringa, Van De Gronde & Roerdink 2012; Van de Gronde, Offringa & Roerdink 2016) to enlarge the flag mask and avoid a failed detection for RFI with weak intensities in the presence of the variation in received power (Offringa et al. 2012). After applying the SIR operator, the performance, based on the FNR and F1 score, of the traditional SumThreshold is improved. However, it raises the FPR from 2.05 to 7.98 per cent. Therefore, this work utilizes the thresholding method on a SED curve and removes all the pixels corresponding to the flagged frequency channels to handle the variation of the received power. This optimization dramatically improves the performance of the RFI mitigation method without bringing about too many false positives (the last row in Table 2). This work also tried to use the SIR operator to detect blob RFI but it did not give the result we expected. Although the main parts of the blob RFI are stronger than other signals (such as pulsar signals, band RFI, background information, etc.), its wings are weak and presumably continue under the noise (Fig. 8). The detection of these weak RFI pixels can be improved by using the SIR operator. However, some of the similar weak pixels could also be non-RFI pixels, which result in false positives with the SIR operator. At the same time, the SIR operator needs more computation and decreases efficiency. Therefore, the proposed scheme utilizes the SumThreshold without SIR to detect blob RFI.

As for the TPF-ST and GF-ST, the difference between these two methods and the proposed ArPLS-ST is in the baseline fitting procedure. The experiments in the last three rows of Table 2 show that the proposed ArPLS method can obtain a more appropriate estimation for the baseline than the tile-based polynomial fitting and Gaussian filter method. Although the proposed scheme has a slightly higher FPR than the GF-ST, the false positives in the ArPLS-ST are always the 'small burr' in the integration. However, the GF-ST always makes some mistakes in the multiple-peak regions, which is likely to have some more severely adverse effects on subsequent analysis and application. As shown in Fig. 10, the TPF-ST method also suffers from the multiple-peak problem in frequency ranges 1220–1244 and 1281–1305 MHz, etc.

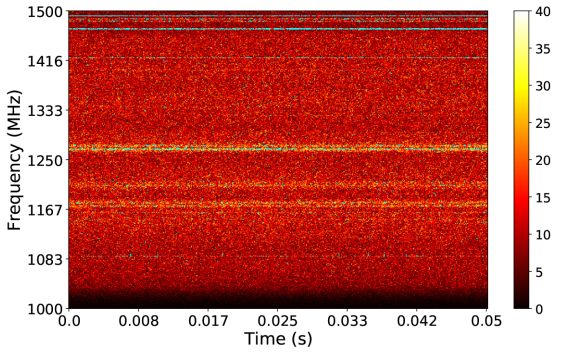
The first method, rfifind, is totally different from the other methods described above. It mainly detects the broad-band RFI with a short duration and strong narrow-band RFI (Ransom 2001). The broad-band RFI with a short duration is detected by performing a time-domain clipping of the curve integration by channels and the strong narrow-band RFI is detected based on the computational result of the fast Fourier transform algorithm. However, rfifind is not able to detect relatively weak RFI and blob RFI. Therefore, the results in Table 2 show that rfifind does not perform as well as the ArPLS-ST overall.

Fig. 10 shows the results of the six RFI flagging methods on one time–frequency image. It is found that the results of the last three methods (TPF-ST, GF-ST and ArPLS-ST) are similar, in general. Besides, rfifind is unable to flag the blob RFI. Therefore, the area outside the band RFI regions is dark because of the high intensity of blob RFI (Fig. 10a). As for the traditional SumThreshold, it cannot flag the band RFI completely, especially in the absence of the SIR operator.

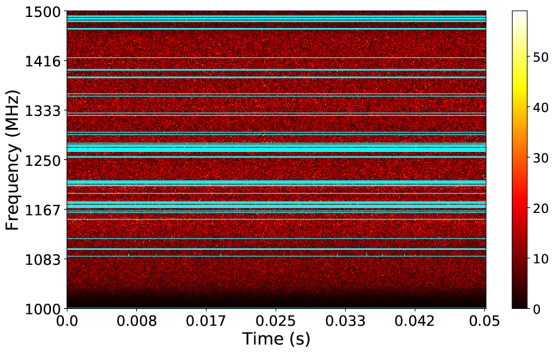


(a) The detection result of rfifind.

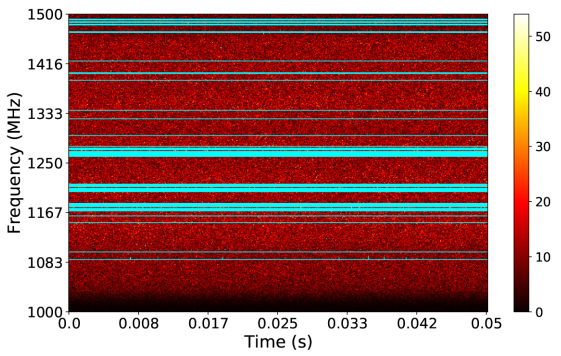
(b) The detection result of the SumThreshold (with SIR).



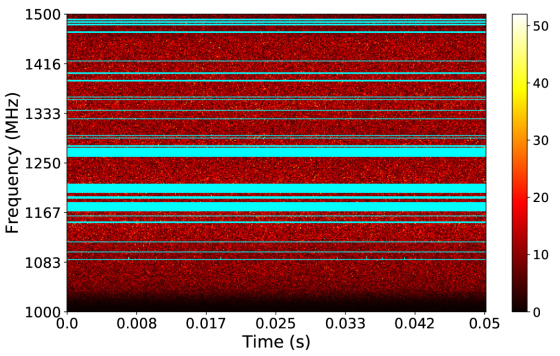
(c) The detection result of the SumThreshold (without SIR)



(d) The detection result of PF-ST.



(e) The detection result of GF-ST.



(f) The detection result of ArPLS-ST.

Figure 10. The detection results for the six methods. The top-left panel looks black outside the RFI regions because the strong blob RFI has not been removed by rfifind and the intensity of the remaining part is relatively weak.

Logically, every method in rfifind, SumThreshold (with SIR), SumThreshold (without SIR), ArPLS-ST, TPF-ST and GF-ST consists of two procedures: baseline fitting and RFI detection (band RFI and blob RFI). To fit the baseline, the traditional SumThreshold utilizes a Gaussian filter (Offringa et al. 2010) in a 2D time–frequency image; the last three methods (TPF-ST, GF-ST and ArPLS-ST) conduct computations on a one-dimensional SED curve, which result in a more efficient implementation than the traditional SumThreshold. In the RFI detection, the five methods have similar running-time costs. Therefore, TPF-ST, GF-ST and ArPLS-ST are much more efficient than SumThreshold (with SIR) and SumThreshold (without SIR) (Table 2).

It is worth noting that all the listed methods except for rfifind are implemented in Python without any optimization. However, the execution speed of Pthe ython code is significantly lower than that of Fortran, C,or C++ because Python is an interpreted language, not compiled, and its efficiency is affected by the Global Interpreter Lock (GIL). Therefore, although the comparison on execution time between the methods is relatively fair in Table 2, the efficiency of TPF-ST, GF-ST, SumThreshold (with or without SIR) and ArPLS-ST can be increased if they are implemented using C, C++ or Fortran, and optimized (e.g. parallel computing, GPU acceleration).

In summary, the scheme ArPLS-ST is proposed for radio data processing. Experiments on the FAST data show that this scheme can effectively detect RFI. It provides a fast and accurate baseline estimation method based on the SED curve to reduce the potentially negative influences from the inconsistency of the receiver response, and can accurately locate the RFI regions. Several identification strategies are designed for detecting RFI.

In future, some potential improvements and extensions still can be made.

(i) **Parameter set-up**. There are two types of parameters that need to be set in the ArPLS-ST. One is the smoothness parameter λ in the ArPLS algorithm. This work experientially set it to 10 000, a constant, which could obtain satisfactory results for all of the available FAST data. However, it may not be good enough to handle the complex radio environment in other situations. In practice, the smoothness parameter can be automatically determined by some statistics that quantify the characteristics of the original integration curve. Another is the threshold in the SumThreshold algorithm. As mentioned in Section 3, the threshold is determined by the Kσ criterion, which concentrates on the aggregation of the pixel intensity distribution. In fact, it may be a more natural and robust way to set this kind of parameter through pixels far away from the cluster. Some outlier detection techniques may be taken into account to solve this problem in future.

(ii) **More accurate flagging strategy for band RFI**. The band RFI flagging strategy in the ArPLS-ST will remove all of the pixels within the marked channel in one subintegration. This trigger-remove-all scheme may potentially result in some false positives to some extent. An accurate band RFI flagging strategy that has the ability to identify the band RFI with different durations may be a better choice.

(iii) **Distinction between the signal of interest and RFI**. The key to threshold-based RFI flagging methods is that the energy of the RFI bursts is much stronger than that of non-RFI data and the signal of interest. Traditional thresholding algorithms will identify the strong signal as the RFI. This kind of false positive causes huge losses for research and is not allowed to happen in practice. Therefore, distinguishing them according to their characteristics is the most important and urgent task for the thresholding-based RFI flagging methods. We designed a novel method to distinguish between the signals of interest and the detected candidate RFI.

(iv) **Software availability**. The Python software package of this work will be updated whenever possible at http://zmtt.bao.ac.cn/GPPS/RFI for open usage, given the proper citation to this paper.

ACKNOWLEDGEMENTS

XL and QZ were supported by the National Natural Science Foundation of China (Grant No. 61075033), the Natural Science Foundation of Guangdong Province (No. 2020A1515010710). CW was supported by the National Natural Science Foundation of China (U1731120).

DATA AVAILABILITY

A Python software package of this work and sample data are available at http://zmtt.bao.ac.cn/GPPS/RFI.

REFERENCES

Akeret J., Seehars S., Chang C., Monstein C., Amara A., Refregier A., 2017a, Astronomy and Computing, 18, 8
Akeret J., Chang C., Lucchi A., Refregier A., 2017b, Astronomy and Computing, 18, 35
Andreev V. P., Rejtar T., Chen H-S., Moskovets E. V., Ivanov A. R., Karger B. L., 2003, Analytical Chemistry, 75, 6314
Baek S-J., Park A., Ahn Y-J., Choo J., 2015, Analyst, 140, 250
Cobas J. C., Bernstein M. A., Martín-Pastor M., Tahoces P. G., 2006, Journal of Magnetic Resonance, 183, 145
Eilers P. H. C., 2003, Analytical Chemistry, 75, 3631
Fridman P., 2008, AJ, 135, 1810
Jiang P. et al., 2020, Res. Astron. Astrophys., 20, 28
Jirasek A., Schulze G., Yu M., Blades M., Turner R., 2004, Applied Spectroscopy, 58, 1488
Kerrigan J. et al., 2019, MNRAS, 488, 2605
Mosiane O., Oozeer N., Aniyan A., Bassett B. A., 2017, in Monebhurrun V., ed., IEEE Radio and Antenna Days of the Indian Ocean, Materials Science and Engineering Conference Series, Vol. 198. IOP Publishing, Bristol, p. 012012
Offringa A., 2012, PhD thesis, University of Groningen
Offringa A., De Bruyn A., Biehl M., Zaroubi S., Bernardi G., Pandey V., 2010, MNRAS, 405, 155
Offringa A., Van De Gronde J., Roerdink J., 2012, A&A, 539, A95
Peck L. W., Fenech D. M., 2013, Astronomy and Computing, 2, 54
Ramsay J. O., Silverman B. W., 2007, Applied Functional Data Analysis: Methods and Case Studies. Springer, Berlin
Ransom S. M., 2001, PhD thesis, Harvard University
Schoemaker L., 2015, Master's thesis, Vrije Universiteit Amsterdam
Shen C., Bao X., Tan J., Liu S., Liu Z., 2017, Optics Express, 25, 16235
van de Gronde J. J., Offringa A. R., Roerdink J. B., 2016, Journal of Mathematical Imaging and Vision, 56, 455
Winkel B., Kerp J., Stanko S., 2007, Astronomische Nachrichten: Astronomical Notes, 328, 68
Zhang Z., Chen S., Liang Y., Liu Z., Zhang Q., Ding L., Ye F., Zhou H., 2010, Journal of Raman Spectroscopy, 41, 659
Zhao J., Zou X., Weng F., 2013, IEEE Transactions on Geoscience and Remote Sensing, 51, 4830

This paper has been typeset from a TEX/LATEX file prepared by the author.