Data Analysis Lab: Assignment

Emma Vandrey
10019590

August 23, 2022

Contents

1 Introduction

The search for gravitational waves and many other research areas in physics require the detection of periodic signals in large sets of noisy data. This project demonstrates one example for such a detection: searching for monochromatic signals injected in Gaussian noise. The signals' frequencies, phases and amplitudes are known to be within the intervals $f \in [10\,\mathrm{Hz}, 1000\,\mathrm{Hz}]$, $\phi \in [0, 2\pi]$ and $A \in [0.5, 2.0]$, making this a so-called ”blind search” with only little prior knowledge about the signals' parameters.
The data sets to be analysed here have the following properties: an observation time of $T_{obs} = 180\,\mathrm{s}$, time steps between the measurements of $dt = \frac{1}{4096}\,\mathrm{s} \approx 2.44\,\mathrm{ms}$, a number of measurement points of $N = \frac{T_{obs}}{dt} = 180 \cdot 4096 = 737\,280$ and Gaussian noise with a standard deviation of $\sigma = 25.0$.

The detection algorithm is written in a C program, which is in the following sections referred to as ”prober”. A second C program is used to generate data sets with known properties like frequency, phase and amplitude, this is used to generate sources for testing different search setups.

2 Statistical methods

2.1 Methods for analysing the data

When analysing noisy data there are two possible outcomes, either the data consists of pure noise or of noise and a signal. Mathematically, this can be described by two hypotheses [1]:

$$H_0 : \quad d(t) = \text{pure noise} = n(t) \tag{1}$$

$$H_1 : \quad d(t) = \text{noise} + \text{signal} = n(t) + s(t) \quad \text{where} \quad s(t) = A \cdot \sin(2\pi f t + \phi) \tag{2}$$

For the analysis of the data, three different methods are being used in this project. The first method calculates a scalar product of the data and the signal which is tested for [1]:

$$S_1 = \frac{1}{N} \sum_{k=0}^{N-1} d(t_k) \cdot s_{norm}(t_k, f, \phi) \quad \text{where} \quad s_{norm}(t) = \sin(2\pi f t + \phi) \tag{3}$$

If the data contains a signal, the value of S_1 reaches a maximum at the frequency and phase of this signal. With this method, it is possible to determine the frequency, phase and amplitude of a signal. While the values of the frequency and phase are returned by prober directly, together with the result for S_1, the amplitude is twice the value of

S_1: $A = 2\,S_1$. This relation can be shown by replacing the general data function d(t) in expression (3) with the data function (2):

$$
\begin{aligned}
S_1 &= \frac{1}{N}\sum_{k=0}^{N-1}(n(t)+s(t))\cdot s_{norm}(t_k,f,\phi)\\
&= \frac{1}{N}\sum_{k=0}^{N-1} n(t)\cdot s_{norm}(t_k,f,\phi) + \frac{1}{N}\sum_{k=0}^{N-1} s(t)\cdot s_{norm}(t_k,f,\phi)\\
&= \frac{1}{N}\sum_{k=0}^{N-1} n(t)\cdot s_{norm}(t_k,f,\phi) + \frac{1}{N}\sum_{k=0}^{N-1} A\cdot s^2_{norm}(t_k,f,\phi)\\
&= \langle n(t)\cdot s_{norm}(t_k,f,\phi)\rangle + \left\langle A\cdot s^2_{norm}(t_k,f,\phi)\right\rangle\\
&= \langle n(t)\rangle\cdot\langle s_{norm}(t_k,f,\phi)\rangle + \left\langle A\cdot s^2_{norm}(t_k,f,\phi)\right\rangle\\
&= \underbrace{\langle n(t)\rangle}_{=0}\cdot\underbrace{\langle \sin(2\pi f t+\phi)\rangle}_{=0} + A\underbrace{\left\langle \sin(2\pi f t+\phi)^2\right\rangle}_{=1/2} = \frac{A}{2}\\
&\Rightarrow A = 2\cdot S_1 \qquad (4)
\end{aligned}
$$

Here, $\langle x\rangle$ denotes the expectation value of x. The expectation value of the product $n(t)\cdot s_{norm}(t_k,f,\phi)$ can be split into two separate expectation values because n(t) and $s_{norm}(t_k,f,\phi)$ are independent. Moreover, the expectation values of both terms vanish since n(t) is a Gaussian random variable with a mean of zero and $s_{norm}(t_k,f,\phi)$ is a sine.

The second method is a so-called "χ^2-test", for which the test values are given by the following expression [1]:

$$S_2 = \frac{1}{N}\sum_{k=0}^{N-1}(d(t_k) - s(t_k,f,\phi))^2 \qquad (5)$$

At the correct frequency and phase of the data's signal (if there is one), the S_2-values reach a minimum. By using this method one can find the frequency and phase of a signal, these values are returned by the program "prober" together with S_2.

The third method computes the Fourier spectrum of the input data [1]:

$$S_3 = \frac{1}{N}\left|\sum_{k=0}^{N-1} d(t_k)\exp(i2\pi f\Delta t+\phi)\right|^2 \qquad (6)$$

For this method, the output of prober is a file with an array of frequencies and the corresponding result for S_3. By plotting S_3 as a function of frequency, one can see sinusoids of which frequencies occur in the data and what is their strength compared to the other constituents of the data. In this way, it is possible to precisely search for signal frequencies.

However, the phase can only be recovered if the amplitude is known and vice versa.

To decide whether or not a local maximum or minimum of S_i is assumed to be a signal, suitable thresholds for each method need to be established. For every value of S_i, there is a certain probability that this value corresponds to pure noise or noise and a signal, respectively. Plotting these probabilities as a function of S_i yields the two probability distributions depicted in figure 1.

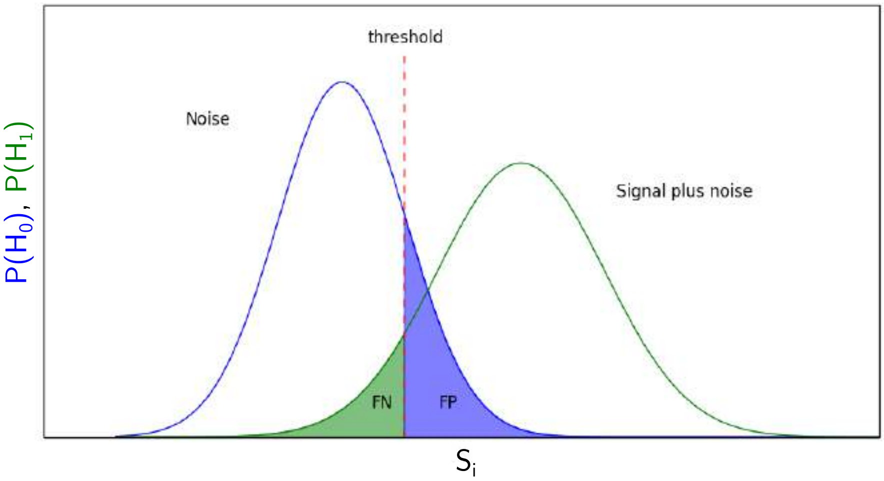

Figure 1: Probability distributions of the two hypotheses (”no signal” or ”signal”), adapted from [1]. The green area equals the probability of a false negative result, the blue area the probability of a false positive result.

If the probability distributions in figure 1 overlap in regions where they are noticeably larger than zero, there is a range of S_i-values for which one cannot decide on one of the hypotheses for sure. Depending on the magnitude of the chosen threshold, there is the possibility of a false negative result, i.e. a signal that is mistaken for noise. On the other hand, there is also the possibility of a false positive, meaning that a supposed signal was only noise in reality. Consequently, the choice of the threshold used for the analysis is an essential way of balancing the probability of false negative and false positive results.

2.2 The ROC-Curve

The receiver operating characteristic curve (ROC-curve) is a way to graphically present the quality of a detector’s results. It is the plot obtained by plotting the true positive rate of the detector versus the false positive rate while varying the threshold used to decide whether a result corresponds to a signal or not.

A perfect detector, never returning a false negative or positive result, has a ROC curve that looks like curve A in figure 2. A detector choosing "signal" or "no signal" randomly would have a ROC curve along the 45° line. A real detector's ROC curve lies somewhere between the 45° line and curve A, like curve B in figure 2, for instance.

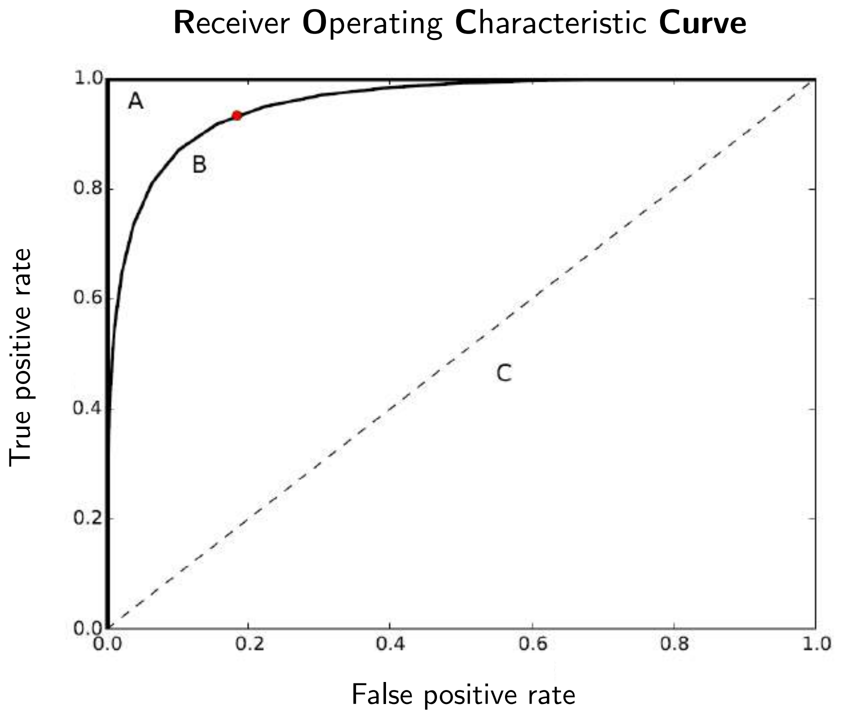

Figure 2: ROC curve of a perfect detector (curve A) and of a real detector (curve B), adapted from [1].

Plotting the ROC curve can be useful when choosing a threshold for S_i: To obtain results that have a high probability of being actual signals and not false positives, it makes sense to choose a threshold for which the ROC curve still has a high true positive rate, but simultaneously a low false positive rate. For curve B in figure 2 this would be somewhere close to the red dot.

2.3 Plotting the ROC-Curves

To be able to make statements about the true and false positive rates of the chosen detection setup, the ROC curves are computed. The general procedure used to arrive at a ROC curve is the following:
First, multiple sources with and without signals (here: 200 each for ”signal” and ”no

signal”) are generated and analysed using methods 1 and 2. Since the parameters of the injected signals are known, ”prober” is applied at these exact parameters. The returned value of S_i is saved in one of two lists, one with the results for pure noise and one with the results for data sets containing a signal.
Then, a list of possible thresholds to distinguish between ”signal” and ”no signal” is generated, ranging from the minimum S_i of both of the result lists to their maximum. For each of those threshold values, the true positive rate is obtained by counting the number of hits among the results in the ”signal”-list that would occur at this threshold and dividing the result by the number of data sets analysed. Similarly, the false positive rate is determined by computing the relative frequency of hits in the ”pure noise”-list. A scatter plot with the true positive rate as the x-coordinate and the false positive rate as the y-coordinate now yields the ROC curve.
Since the magnitude of S_i is influenced by the signal’s amplitude, the ROC curves are plotted separately for different amplitudes $A \in [0.5, 2.0]$.

2.3.1 ROC-Curve of method 1

For method 1 of prober, the distributions of S_1 for data with pure noise and data with signals of the amplitudes 0.5 and 2.0 are shown in figure 3.

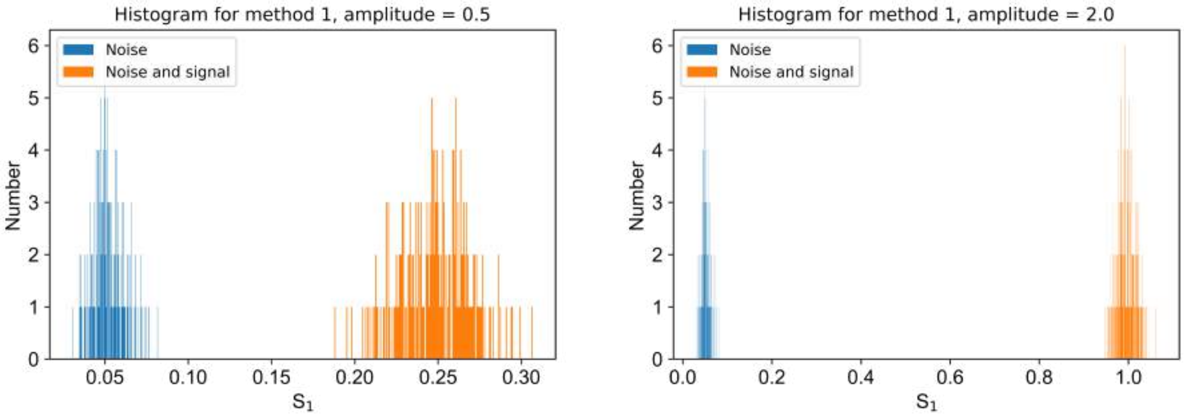

Figure 3: Histograms of S_1 for pure noise and noise and signal, plotted for different amplitudes of the signal.

The distributions of S_1 for data with and without signals in the histograms in image 3 do not overlap. This explains the approximately perfect ROC curve for this method, depicted in figure 4. By choosing a threshold for S_1 that lies at some point where both distributions are zero, one could get a true positive rate of approximately one and a false positive rate of about zero.

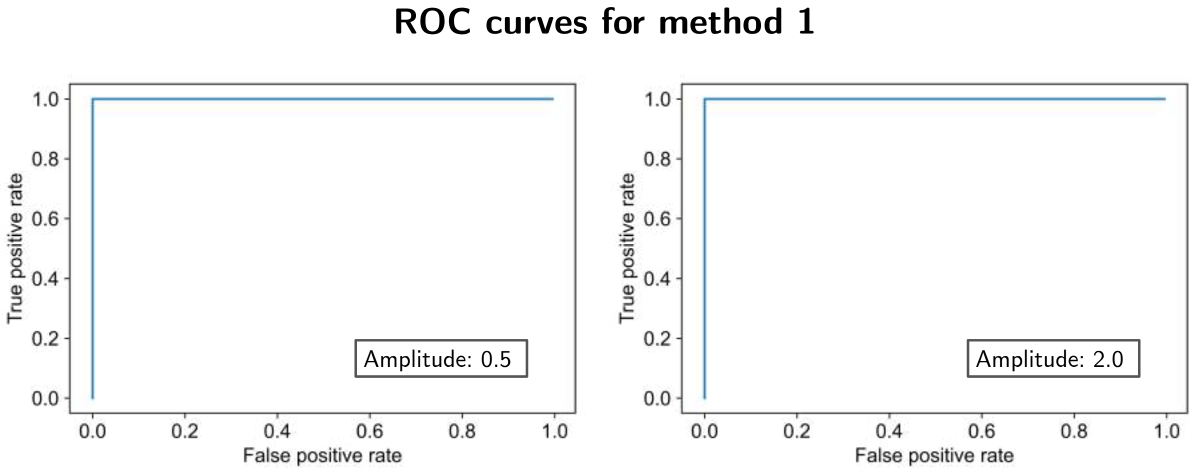

Figure 4: ROC curves for method 1 of prober (used with a template bank with df = 0.005 Hz and dp = 0.5) when analysing signals with an amplitude of 0.5 (left) and 2.0 (right).

2.3.2 ROC-Curve of method 2

Unlike for the first method, the distributions of S_2 for ”no signal” and ”signal” overlap for method 2 of prober (see figure 5). A possible explanation for that could be that the data’s noise is amplified by method 2 more strongly than by method 1. Therefore, choosing a threshold with a higher true positive rate here comes at the cost of a higher false positive rate as well. This is also depicted in the ROC curves in figure 6, which deviate significantly from the ROC curve of a perfect detector and are very close to the 45° line.

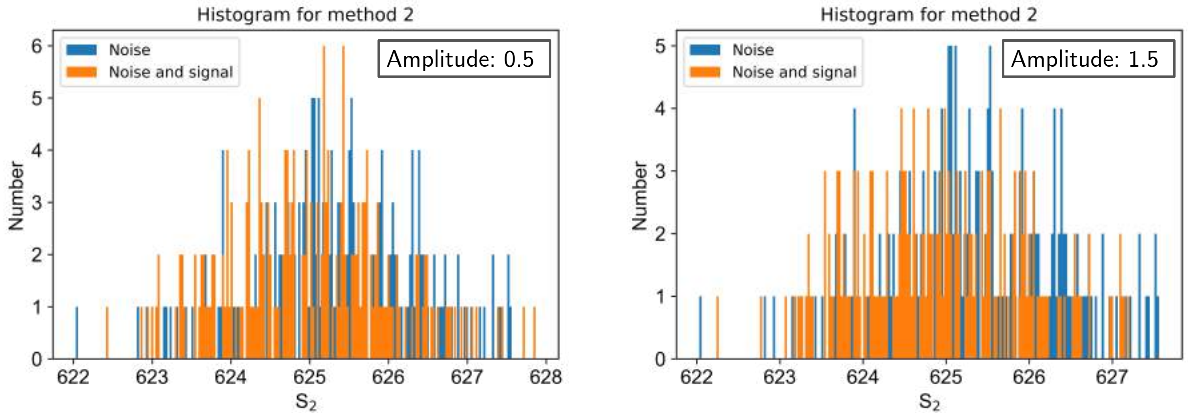

Figure 5: Histograms of S_2 for pure noise and noise and signal, plotted for different amplitudes of the signal.

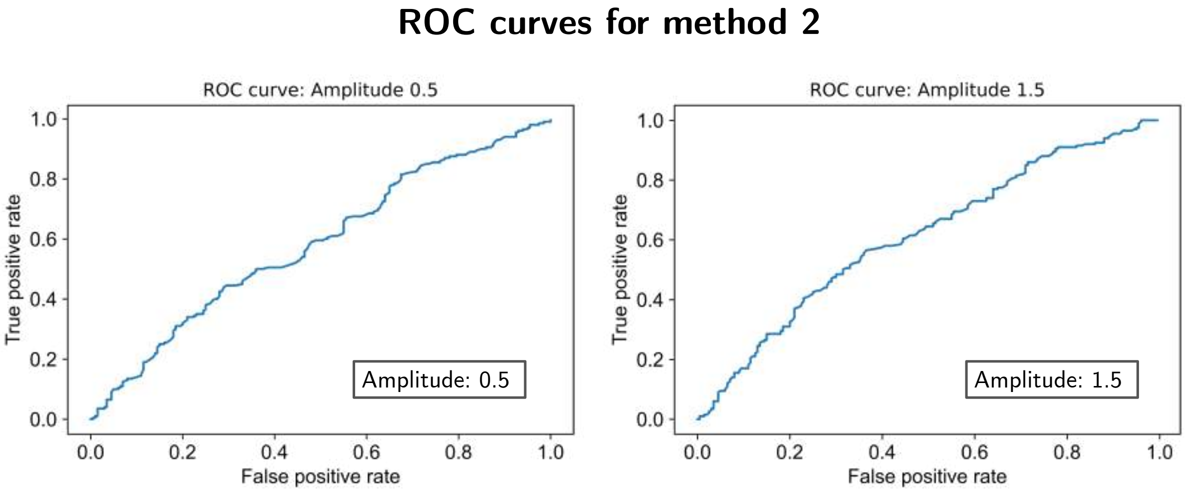

Figure 6: ROC curves for method 2 of prober (used with a template bank with df = 0.005 Hz and dp = 0.5) when analysing signals with an amplitude of 0.5 (left) and 2.0 (right).

Since the ROC curves of method 2 are significantly worse than the ones for method 1, the analysis of the data sets in section 4 will be mainly based on method 1.

3 Finding a suitable setup

3.1 Methods for finding a setup

Before the actual data can be analysed, one needs to know how close the templates of the template bank used for the search need to be in frequency and phase in order to not miss a signal but still come with a reasonable computational cost. The suitability of a template bank for the detection of a particular signal can be estimated by looking at the average of the so-called mismatch, which is given by:

$$M_i = 1 - \frac{S_i(\text{nearest template})}{S_i(\text{correct parameters of the signal})} \tag{7}$$

Here, the values of S_i represent the results of prober for the nearest point of the template bank or the optimal on-target recovery, respectively.

To examine template banks with different degrees of accuracy and the resulting differences in frequency (df) and phase (dp) as well as the mismatch, multiple sources with signals within these three intervals are generated:

1. $f \in [11.0\,\mathrm{Hz}, 11.1\,\mathrm{Hz}]$, $\phi \in [1.0, 2.0]$
2. $f \in [471.0\,\mathrm{Hz}, 471.1\,\mathrm{Hz}]$, $\phi \in [5.0, 6.0]$
3. $f \in [996.0\,\mathrm{Hz}, 996.1\,\mathrm{Hz}]$, $\phi \in [3.0, 4.0]$

Those intervals are chosen to span only a small range in frequencies to be able to test template banks with very small frequency spacing on many data sets without needing runtimes of more than a few hours. In order to still test frequencies of different orders of magnitude within the search space ($f \in [10\,\mathrm{Hz}, 1000\,\mathrm{Hz}]$), there is one interval each for the low, middle and high end of the frequency range.
A number of 50 sources per interval is generated, the frequencies and phases of the signals are selected randomly from the intervals.

3.2 Testing coarse template banks on the whole parameter space

First, the goal is to find out whether we can analyse the whole search space given by $f \in [10\,\mathrm{Hz}, 1000\,\mathrm{Hz}]$, $\phi \in [0, 2\pi]$ with a template bank that is fine enough to not miss a signal but still has a sufficiently small number of templates to be used with prober in a reasonable runtime. The first template bank tested has the following properties:

Name in jupyter notebook	f_{min} [Hz]	f_{max} [Hz]	df [Hz]	ϕ_{min}	ϕ_{max}	$d\phi$
coarse_template_bank	10	1000	1	0	2π	$2\pi/3$

One data set per interval is analysed using method 1 and 2 of prober with the coarse template bank. The result file contains an array with one row per template and three columns, of which the first element is the template's frequency, the second is the template's phase and the third is the output value of prober for those parameters. These results are plotted in a two-dimensional contour plot where the two axes represent the frequency and phase and the colour shows the value of prober's result. Figure 7 shows the contour plots of the results of method 1 and 2 for a dataset with an injected signal of frequency $f = 471.031\,\mathrm{Hz}$ and phase $\phi = 5.883$.

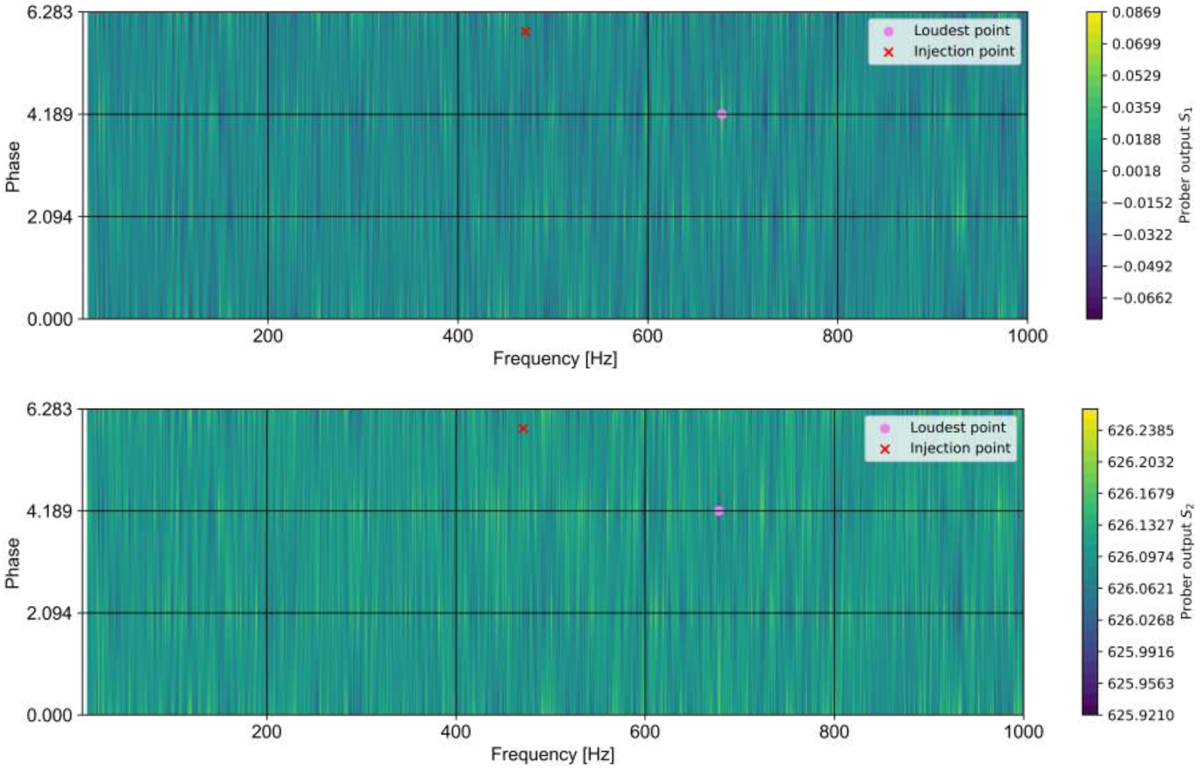

Figure 7: Frequency-phase contour plot of the results of prober used with a template bank with df = 1 Hz.

The images above clearly show that the template bank used here is too coarse and the injected signal cannot be recovered, since the parameters of the injection and the ones of the loudest point differ significantly.

Therefore, the second template bank has a smaller spacing in frequency:

Name in jupyter notebook	f_{min} [Hz]	f_{max} [Hz]	df [Hz]	ϕ_{min}	ϕ_{max}	dϕ
medium_template_bank	10	1000	0.1	0	2π	$2\pi/3$

Analysing the same files as before with this template bank yields the following results:

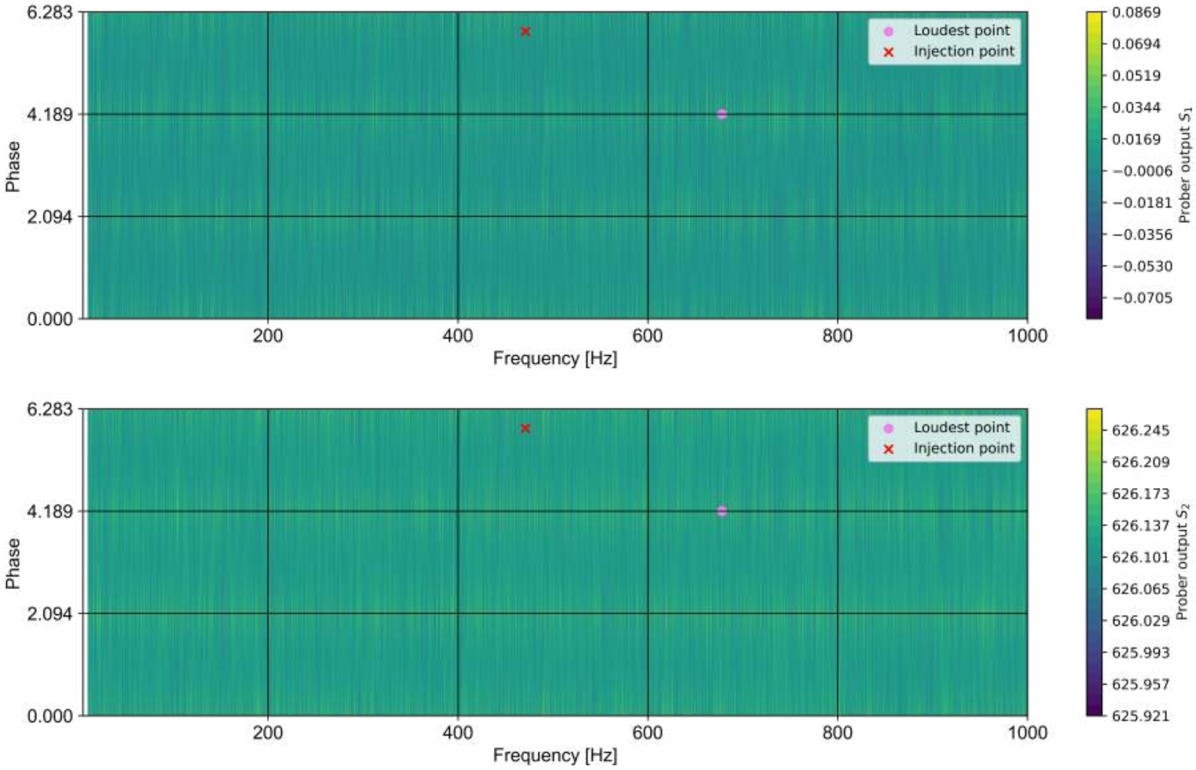

Figure 8: Frequency-phase contour plot of the results of prober used with a template bank with df = 0.1 Hz.

Again, the correct parameters of the signal could not be recovered, so a spacing in frequency of df = 0.01 Hz is not sufficient to detect signals under the conditions given here.

3.3 Testing finer template banks on small subspaces

3.3.1 Differences in frequency, phase and the mismatch

The template banks used in section 3.2 were not suitable to correctly recover the signals because the templates' distances were too large. In this section, finer template banks are being tested that only span the three intervals in which the signals lie:

Name in jupyter notebook	f_{min} [Hz]	f_{max} [Hz]	df [Hz]	ϕ_{min}	ϕ_{max}	dϕ
fine_template_bank_f{f_0}_20	f_0	$f_0 + 1$	0.05	ϕ_0	$\phi_0 + 1$	0.25
fine_template_bank_f{f_0}_50	f_0	$f_0 + 1$	0.02	ϕ_0	$\phi_0 + 1$	0.25
template_bank_f{f_0}_10	f_0	$f_0 + 0.1$	0.01	ϕ_0	$\phi_0 + 1$	0.25
template_bank_f{f_0}_20	f_0	$f_0 + 0.1$	0.005	ϕ_0	$\phi_0 + 1$	0.25
template_bank_f{f_0}_25	f_0	$f_0 + 0.1$	0.004	ϕ_0	$\phi_0 + 1$	0.25
template_bank_f{f_0}_50	f_0	$f_0 + 0.1$	0.002	ϕ_0	$\phi_0 + 1$	0.25
template_bank_f{f_0}_100	f_0	$f_0 + 0.1$	0.001	ϕ_0	$\phi_0 + 1$	0.25

Here, the values of f_0 and ϕ_0 for the three intervals are:

Interval	f_0 [Hz]	ϕ_0
Interval 1: $f \in [11.0\,\text{Hz}, 11.1\,\text{Hz}]$, $\phi \in [1.0, 2.0]$	11	1.0
Interval 2: $f \in [471.0\,\text{Hz}, 471.1\,\text{Hz}]$, $\phi \in [5.0, 6.0]$	471	5.0
Interval 3: $f \in [996.0\,\text{Hz}, 996.1\,\text{Hz}]$, $\phi \in [3.0, 4.0]$	996	3.0

For the frequency spacings in this section, testing the whole search space would take too long, that is why only small intervals are used.
All of the 50 sources per interval are analysed by prober, using the various template banks. For each template bank, the average and maximum of the difference in frequency between the closest template and the actual parameters (df), the difference in phase (dϕ) and the corresponding mismatch are determined:

df of template bank [mHz]	⟨df⟩ [mHz]	Max. df [Hz]	⟨dp⟩	Max. dp	⟨mismatch⟩	Max. mismatch
0.010	2.5	14.0	0.457	0.990	0.3807	0.9475
0.005	1.6	3.0	0.523	0.990	0.2488	0.5959
0.004	0.9	2.0	0.388	0.990	0.1032	0.3126
0.002	0.3	1.0	0.204	0.613	0.0243	0.0796
0.001	0.0	0.0	0.100	0.177	0.0069	0.0279

Table 1: Average and maximum values of df, dp and the mismatch for template banks of varying accuracy.

3.3.2 ROC curves and frequency-phase plots

To gain a better insight into the properties of the fine template banks, the ROC curves for using method 1 of prober with those template banks are plotted. In order to do so, 100 sources each with pure noise and with noise and a signal are generated and analysed

using method 1 and the template bank to be tested. To obtain the true positive rate for a certain threshold of S_1, the maximum value of S_1 is determined for every dataset containing a signal. If this value of S_1 is above the threshold and the frequency of the corresponding template is within a certain range of the signal's actual frequency, the count of true positive results is increased by one. The final number of true positives is divided by the number of signal files that were analysed. To obtain the false positive rate, the maximum values of all results of the noise files are extracted and counted as a false positive, if the value of S_1 is higher than the threshold. The result is again divided by the number of noise files analysed.
The range of frequencies that is accepted as a true positive is varied depending on the accuracy of the template bank. It was chosen in such a way that a frequency can be off by at most three templates (in both directions) until it is no longer considered to be correct.

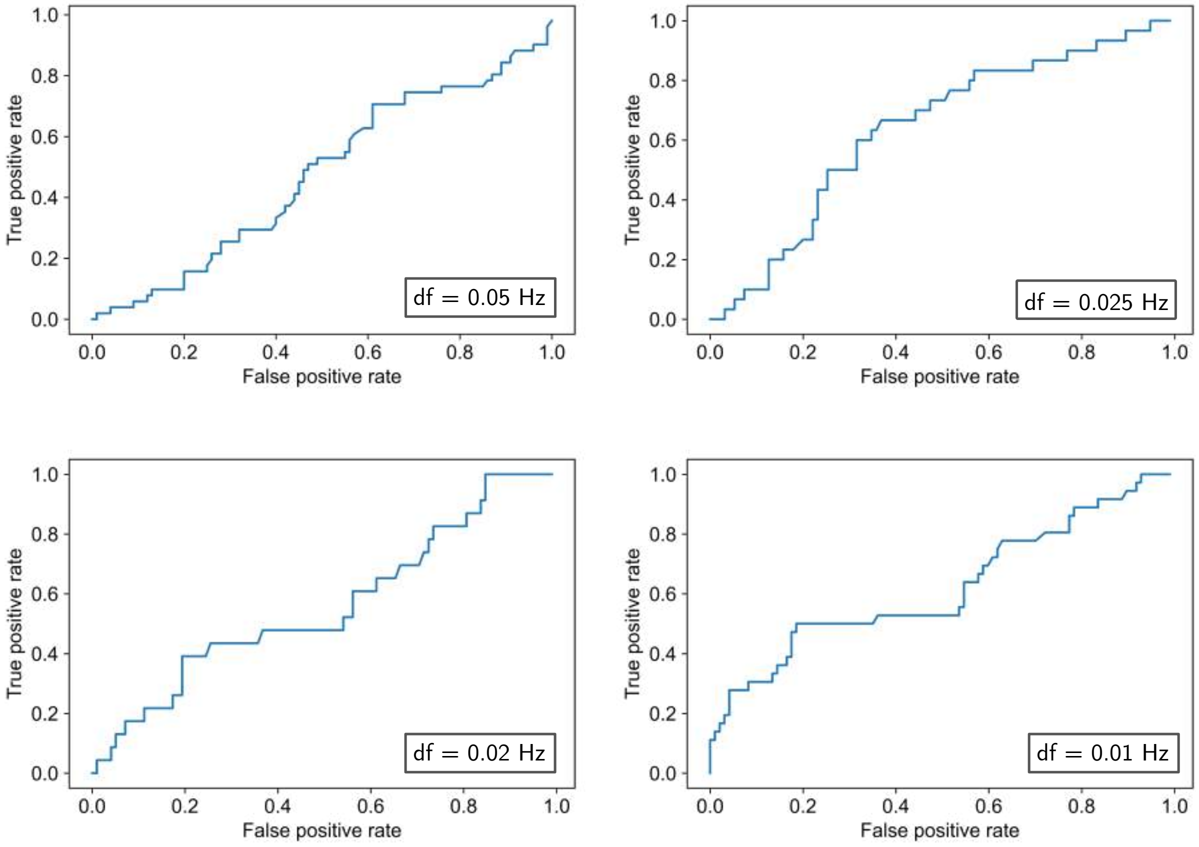

Figure 9: ROC curves for template banks with df = 0.05 Hz, df = 0.025 Hz, df = 0.02 Hz and df = 0.01 Hz.

Since the ROC curves in figure 9 of the template banks with frequency steps of df = 0.05 Hz, df = 0.025 Hz, df = 0.02 Hz and df = 0.01 Hz are all very close to the 45° line, these template banks cannot be used to detect the signals at a sufficiently high true positive rate and a low false positive rate.

The ROC curve for method 1 used with a template bank of accuracy df = 0.005 Hz is shown in figure 10. Unlike the ROC curves of the coarser template banks, this ROC curve is almost equal to the one of a perfect detector, so a template bank with df = 0.005 Hz can be used to analyse the data.

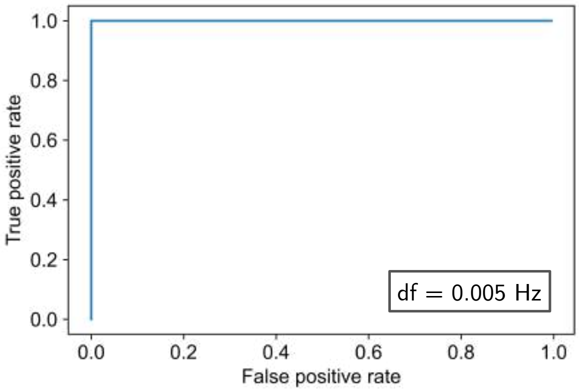

Figure 10: ROC curves for template bank with df = 0.005 Hz.

Additionally, the frequency-phase contour plots are plotted for three data sets, one for each interval of parameters. For the template banks with df = 0.005 Hz and $d\phi = 0.25$, the frequency-phase contour plots are shown in figure 11, where the three loudest points are at templates with the closest possible frequency to the injection point. The grid nodes in the plot depict the templates of the template bank.

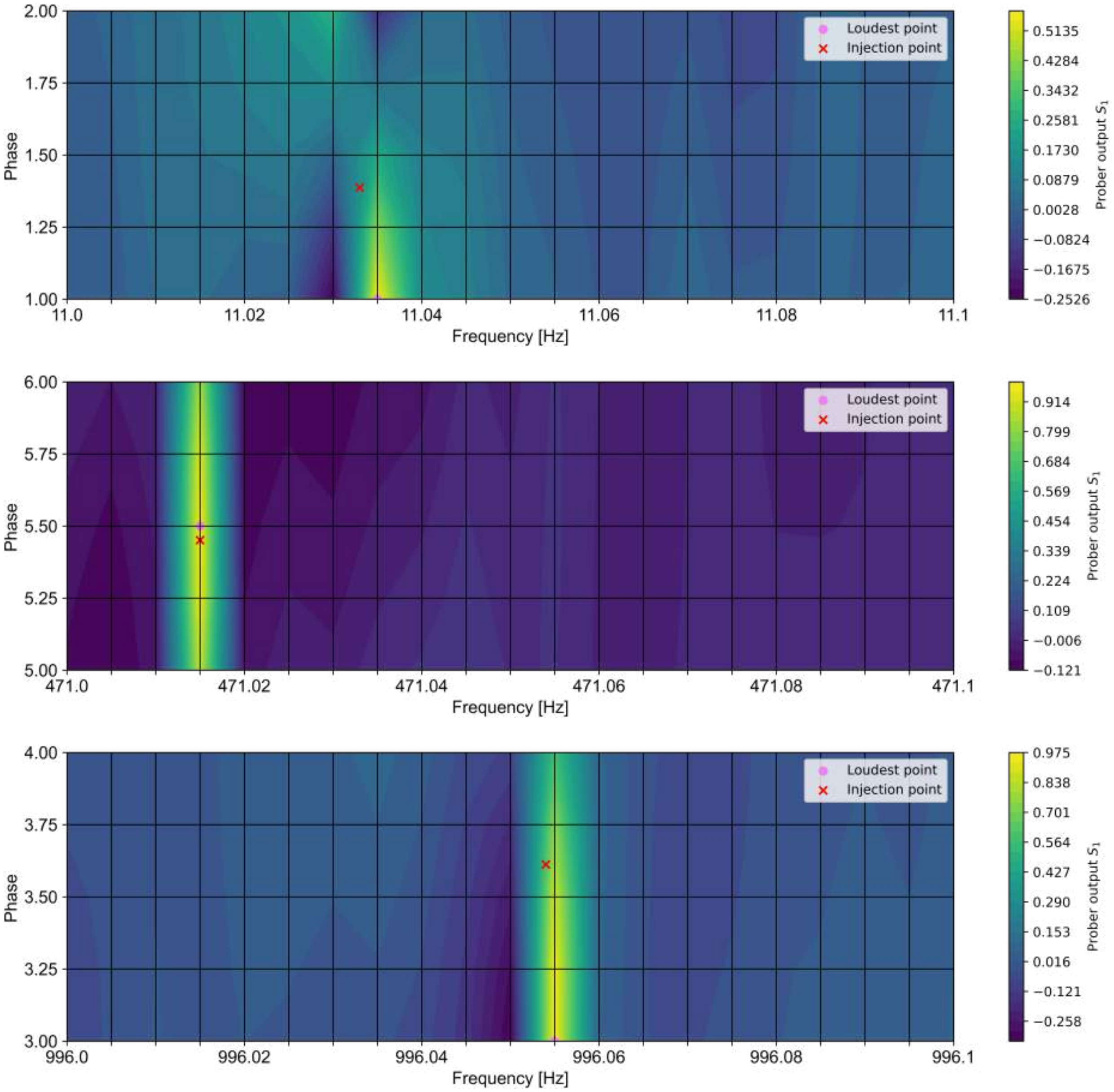

Figure 11: 2D contour plots for template banks with df = 0.005 Hz.

In figure 12 and 13, the results of prober for template banks with df = 0.002 Hz and df = 0.001 Hz, respectively, are plotted. As for df = 0.005 Hz, the loudest points are very close to the injection points, i.e. the risk of missing a signal would be very low when using a template bank with those frequency distances. By looking at the values of S_1 assigned to the colours of the contour plot, one can also see that the closer the templates are, the higher are the results for S_1 (on average). So for a finer template bank, the results corresponding to a signal can be distinguished more clearly from the results corresponding to the surrounding noise.

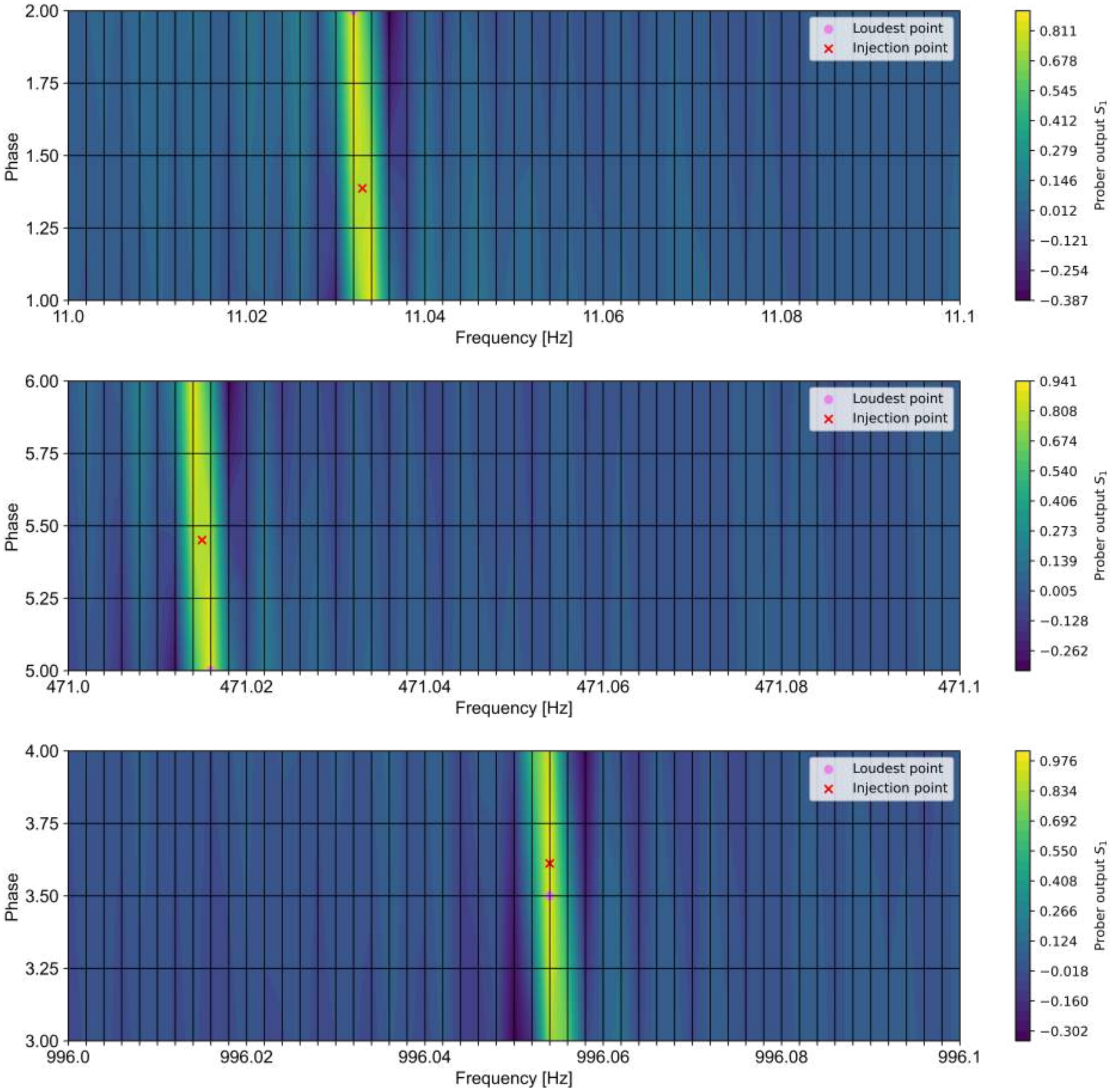

Figure 12: 2D contour plots for template banks with df = 0.002 Hz.

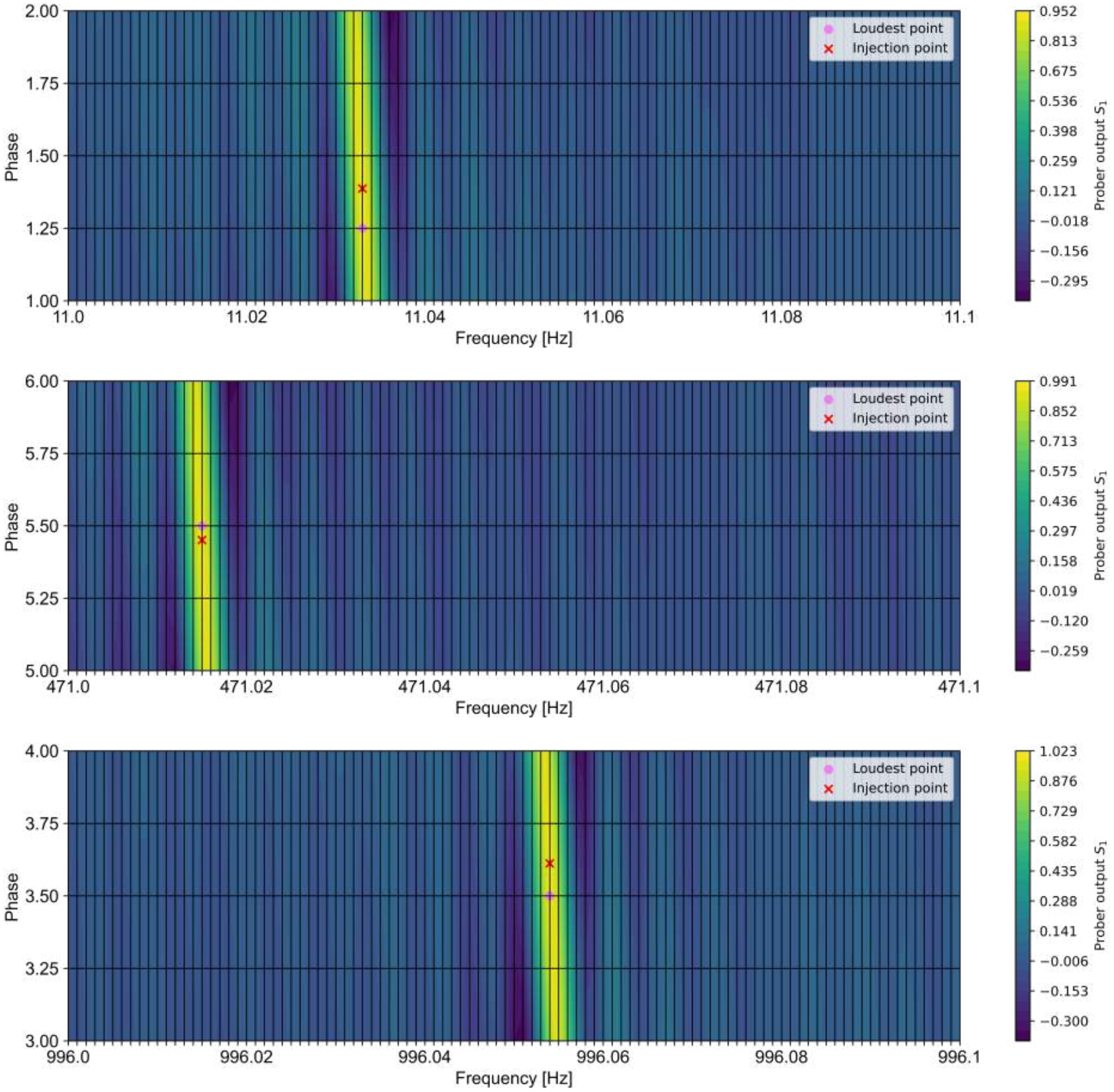

Figure 13: 2D contour plots for template banks with df = 0.001 Hz.

3.4 Runtime analysis

To predict the runtime prober needs to search the whole parameter space when used with a particular template bank, smaller areas of the parameter space are analysed using the template banks examined in section 3.3. The runtime is measured and then scaled up to the number of templates necessary to span the total parameter space. In that way, it

is possible to determine which template banks would have a reasonable computational cost and which would exceed a certain time limit.
The scaling-up of the runtime is done by inserting the measured time and the number of templates into the following expression:

$$T_{whole\,space} = T_{subspace} \cdot \frac{N_{templates,whole\,space}}{N_{templates,subspace}} \tag{8}$$

Here, $T_{whole\,space}$ is the time needed to analyse the whole parameter space using that particular template bank, $T_{subspace}$ is the measured runtime for a smaller subspace, $N_{templates,whole\,space}$ is the number of templates one would need for the total space in this case and $N_{templates,subspace}$ is the number of templates analysed for the subspace. The results are listed in the following table 2:

df of template bank [Hz]	dp of template bank	Number of templates needed for whole parameter space	Runtime for ... templates Number of templates: runtime [s]	Total runtime [h]
0.005	0.25	5,174,309	273: 14.6	76.88
0.004	0.25	6,467,880	338: 17.5	93.02
0.002	0.25	12,935,733	663: 32.3	175.06
0.001	0.25	25,871,440	1313: 60.0	328.40

Table 2: Runtime needed for method 1 used with template banks of varying accuracy.

Since all of the total runtimes in table 2 are too long to use those template banks in this project and the coarser template banks, which would require less time, would have a high probability of missing signals, the first search will be done using method 3. The follow-up searches can then be done using methods 1 and 2 with fine template banks, analysing only certain areas of interest and therefore coming at a smaller computational cost.

4 Analysis of the Data

4.1 Steps of the blind search

Since the use of coarse template banks like the ones tested in section 3.2 would lead to a high risk of missing signals, methods 1 and 2 of prober need to be used with finer template banks that require long run times. Therefore, methods 1 and 2 will not be used to search the whole parameter space but only small intervals of it after pre-filtering for areas of interest with method 3.
To further examine these results, small intervals around each of the filtered frequencies

are searched using method 1 with fine template banks like the ones in section 3.3. All candidates with values of $S_{1,2}$ larger than certain thresholds are clustered in frequency and a follow-up search is performed, using maximally refined template banks with df = 0.001 Hz, $d\phi = 0.01$.

To sum up, the steps of the blind search are:

1. Pre-filtering for frequencies of interest using method 3.
2. Analysis of 5 Hz-intervals with methods 1 and 2, using template banks with df = 0.005 Hz and $d\phi = 0.25$.
3. Clustering the candidates.
4. Follow-up search using template banks with df = 0.001 Hz and $d\phi = 0.04$.

4.2 Pre-filtering with method 3

The pre-filtering is done by extracting all frequencies for which the result of method 3 is larger than $S_{3,threshold} = 5.6 \cdot 10^9$. This threshold was chosen to be only slightly above most of the peaks corresponding to noise in the Fourier spectrum in order to not miss any signals during the first stage of the search.

Pre-filtering dataset 1

The Fourier spectrum of the first data set is shown in figure 14. There are two peaks above the chosen threshold of $S_{3,threshold} = 5.6 \cdot 10^9$:

Frequencies [Hz] with $S_3 \geq S_{3,threshold} = 5.6 \cdot 10^9$	S_3 [10^9]
20.0033	153.801
160.099	5.648

Table 3: Candidates above threshold for dataset 1, analysed with method 3.

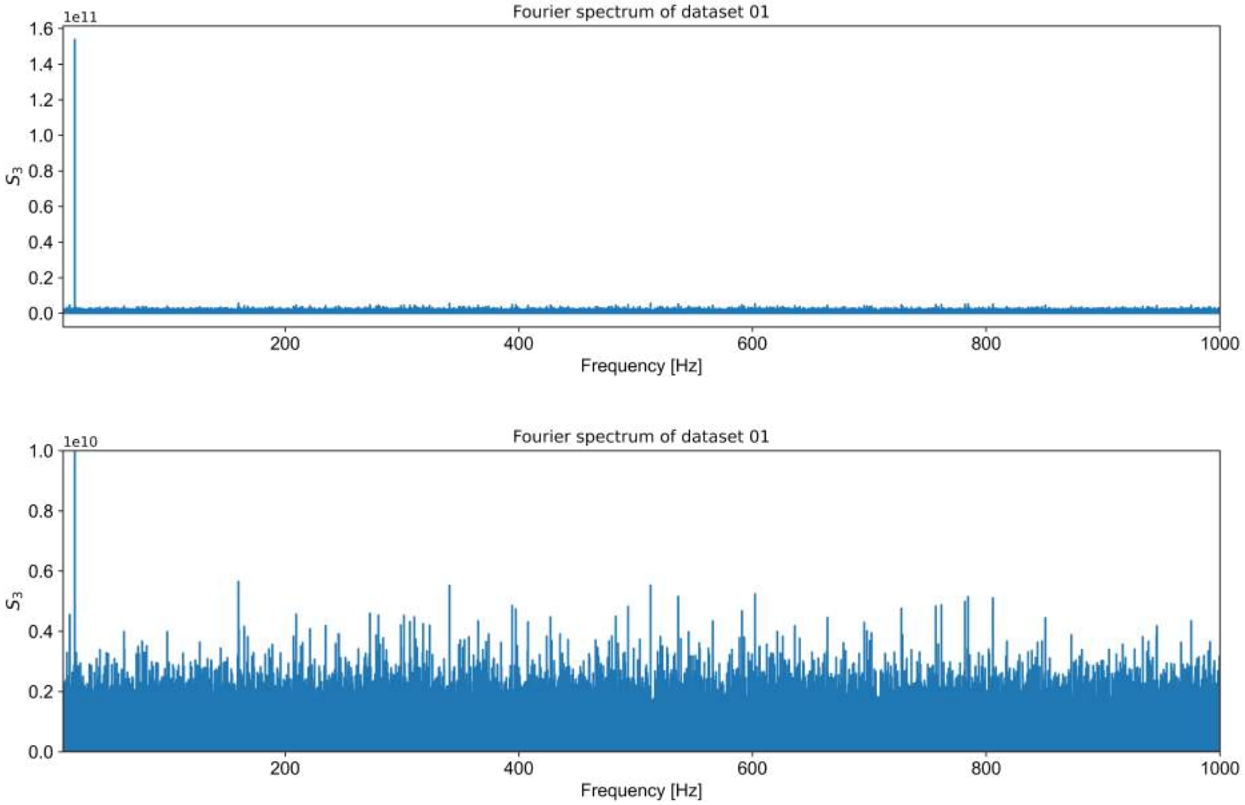

Figure 14: Fourier spectrum of dataset 1.

Pre-filtering dataset 2

The second dataset's Fourier spectrum is plotted in figure 15. Peaks above the threshold occur at the following frequencies:

Frequencies [Hz] with $S_3 \geq S_{3,threshold} = 5.6 \cdot 10^9$	S_3 [10^9]
113.308	5.848
163.599	5.915
385.364	506.506
692.737	5.662

Table 4: Candidates above threshold for dataset 2, analysed with method 3.

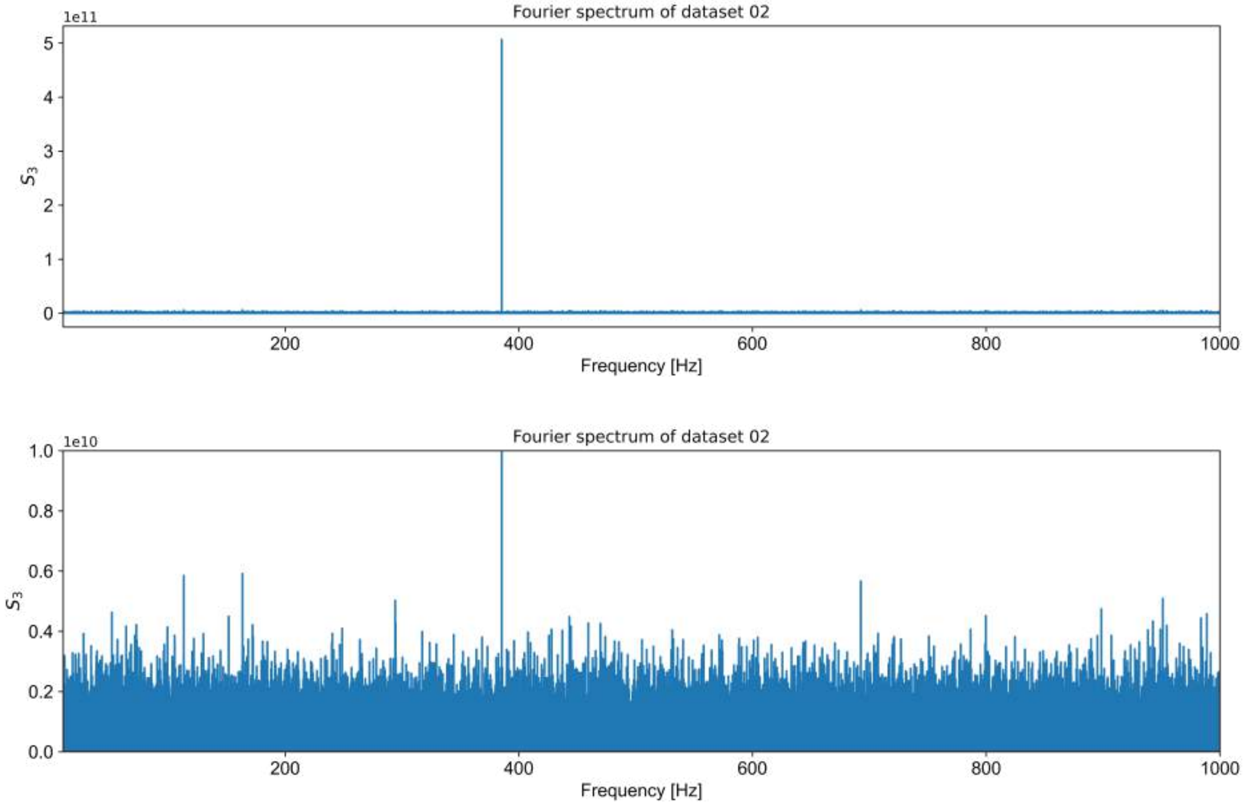

Figure 15: Fourier spectrum of dataset 2.

Pre-filtering dataset 3

The Fourier spectrum of dataset 3 has several peaks above the threshold of $S_{3,\,threshold} = 5.6 \cdot 10^9$:

Frequencies [Hz] with $S_3 \geq S_{3,\,threshold} = 5.6 \cdot 10^9$	S_3 [10^9]
57.9708	17.150
57.9763	164.313
57.9819	89.873
57.9874	9.969
57.9930	7.845
57.9985	6.868
467.7440	6.141
560.4650	8.686

Table 5: Candidates above threshold for dataset 3, analysed with method 3.

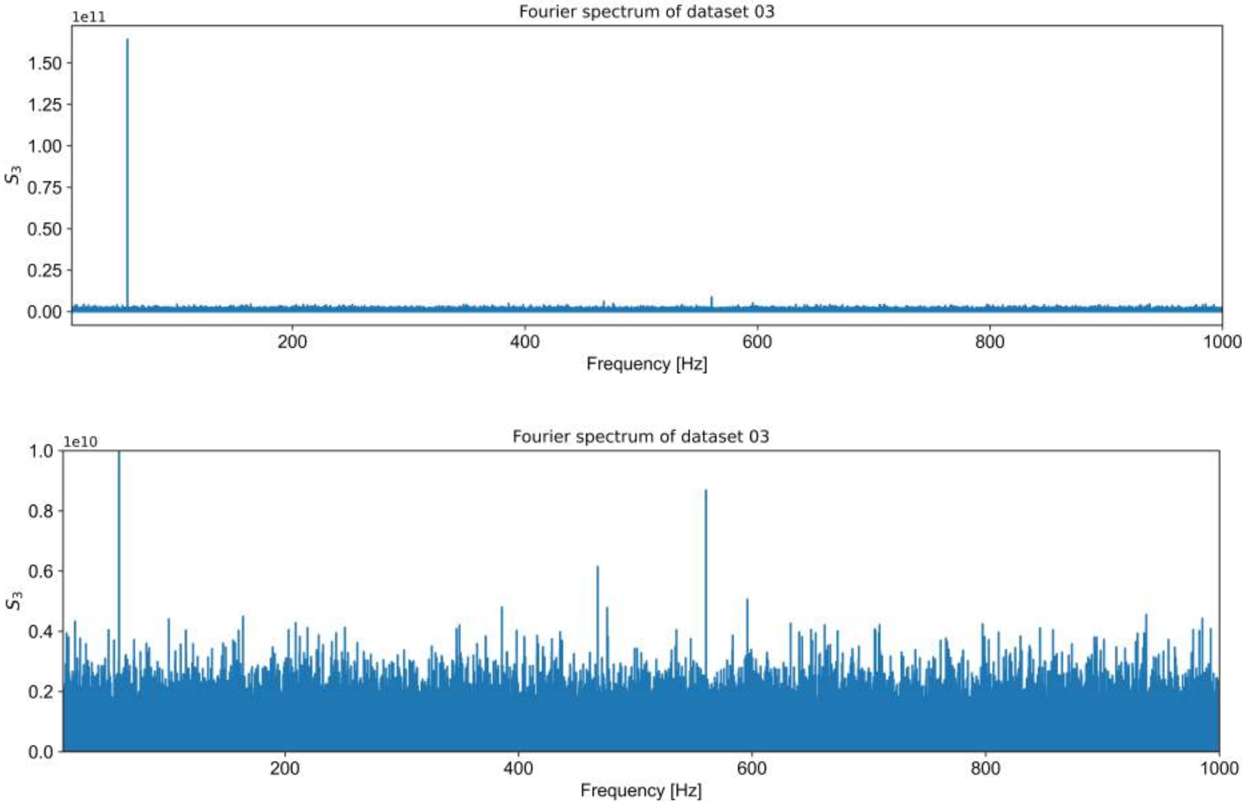

Figure 16: Fourier spectrum of dataset 3.

Pre-filtering dataset 4

For the fourth dataset, the peaks in the Fourier spectrum above the threshold are:

Frequencies [Hz] with $S_3 \geq S_{3,threshold} = 5.6 \cdot 10^9$	S_3 [10^9]
13.9579	5.631
137.3400	5.689
887.5030	7.103
887.5090	15.490
887.5140	409.924
887.5200	56.750
887.5250	8.732
945.7910	6.157

Table 6: Candidates above threshold for dataset 4, analysed with method 3.

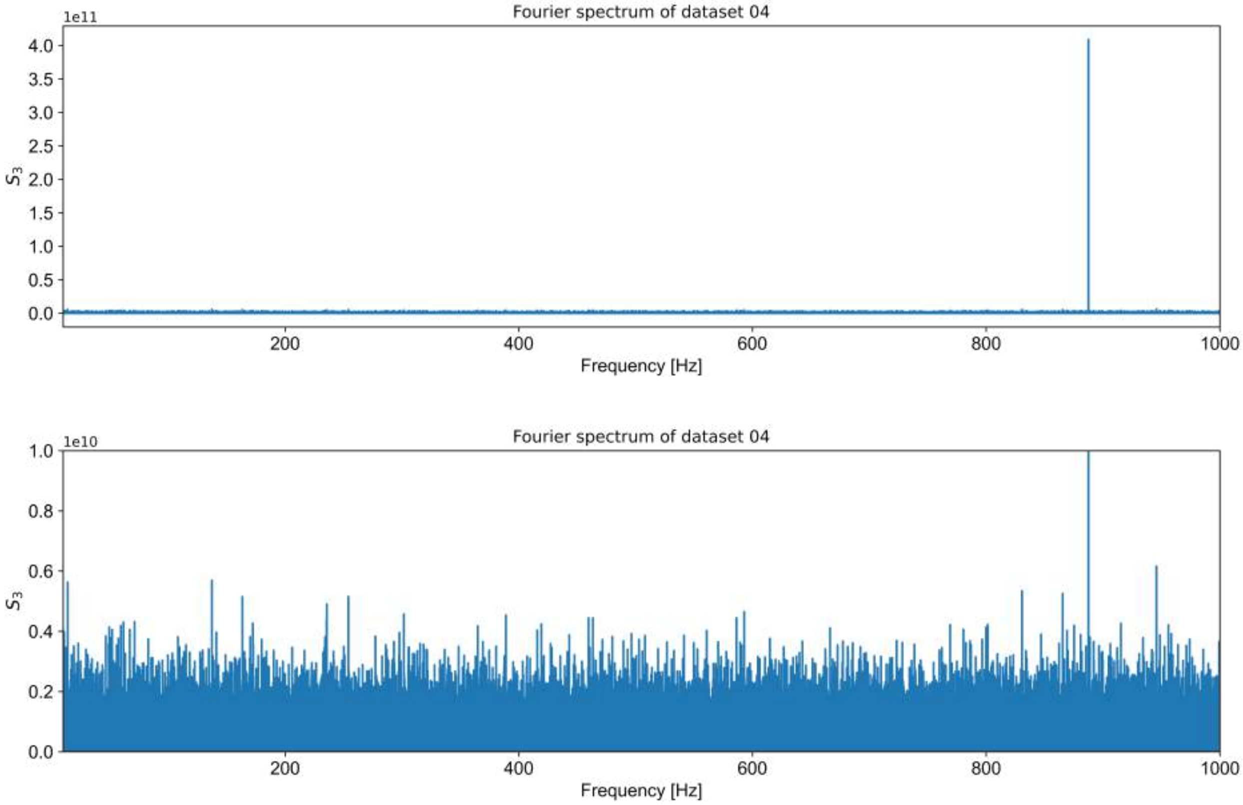

Figure 17: Fourier spectrum of dataset 4.

Pre-filtering dataset 5

The candidates of dataset 5, filtered from the data's Fourier spectrum, are:

Frequencies [Hz] with $S_3 \geq S_{3,\,threshold} = 5.6 \cdot 10^9$	S_3 [10^9]
211.207	6.342
211.213	40.824
211.218	472.570
211.224	14.651
211.230	14.073
519.575	5.661

Table 7: Candidates above threshold for dataset 5, analysed with method 3.

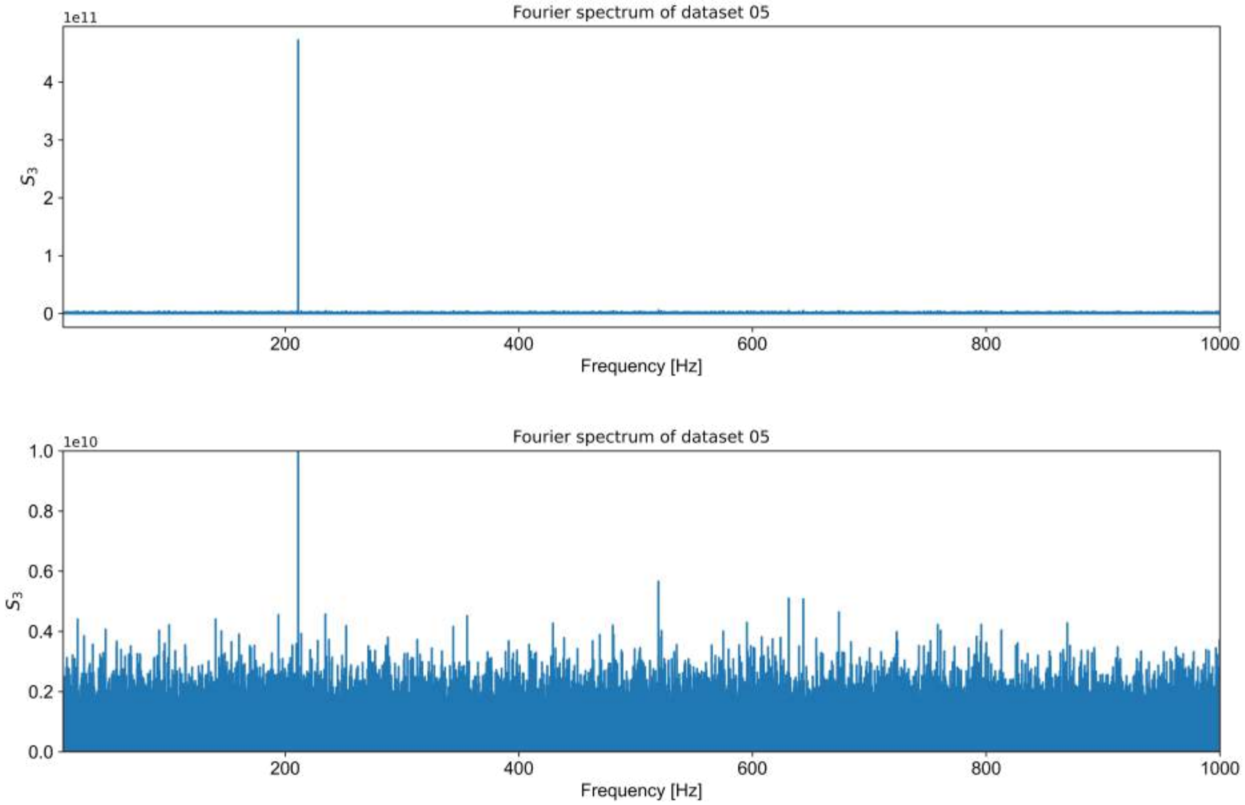

Figure 18: Fourier spectrum of dataset 5.

4.3 Search with method 1

To further analyse the areas of interest found in section 4.2, template banks spanning intervals of 5 Hz around each frequency from the tables 3 to 7 with frequency steps of df = 0.005 Hz and phase steps of dp = 1.05 are generated. Using these template banks, the data sets 1 to 5 are analysed with method 1 of prober and all results with values of S_1 larger than 0.075 are extracted. Figure 19 shows the frequency-phase contour plot of the results for the first area of interest of dataset 1.

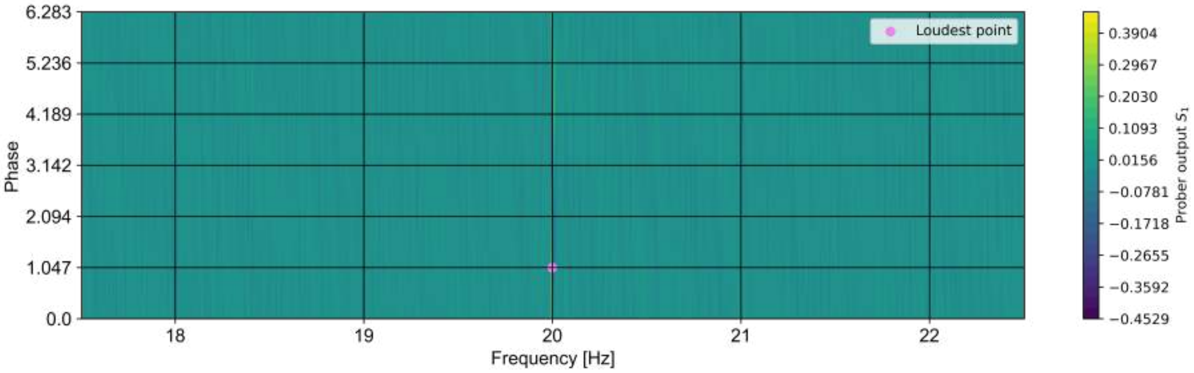

Figure 19: 2D contour plot for the first area of interest of dataset 1, analysed with method 1 using a template bank with df = 0.005 Hz.

4.4 Clustering and follow-up search

Before the final search with a maximally refined template bank is carried out, the candidates of the earlier search stage are clustered in frequency. The resulting clusters for each of the five data sets are:

Data set	Cluster	Start frequency $f_{c,s}$ [Hz]	End frequency $f_{c,e}$ [Hz]
Data 01	1	19.981	20.011
	2	160.059	160.089
Data 02	1	113.273	113.303
	2	163.559	163.589
	3	385.291	385.321
	4	692.607	692.637
Data 03	1	57.940	57.970
	2	467.654	467.684
Data 04	1	137.300	137.330
	2	887.350	887.380
Data 05	1	211.170	211.200
	2	519.475	519.505

Table 8: Clusters in frequency to be analysed in follow-up search.

All of the clusters in table 8 are analysed using methods 1 and 2 with a template bank with df = 0.001 Hz and $d\phi = 0.04$. As the total range of possible phases is searched, the parameter space analysed in this last step is given by $f \in \bigcup_c \left[f_{c,s}, f_{c,e}\right]$ and $\phi \in [0, 2\pi]$.

For each cluster, the candidate with the best result for S_i is extracted and the corresponding probabilities of a true positive and a false negative are determined using the ROC curves. The results are listed in table 9:

Data set	Loudest candidate (f, ϕ)	S_1	Amplitude	P(TP) [%]	P(FP) [%]
Data 01	20.000, 6.24	0.5319	1.0638	100.0	0.0
	160.073, 5.36	0.1044	0.2088	100.0	0.0
Data 02	113.290, 2.24	0.1081	0.2162	100.0	3.5
	163.572, 1.24	0.1036	0.2078	100.0	1.0
	385.299, 1.60	0.9654	1.9308	100.0	1.5
	692.623, 2.72	0.1017	0.2034	100.0	18.5
Data 03	57.969, 0.40	0.7587	1.5174	100.0	0.0
	467.667, 4.96	0.1090	0.2180	100.0	22.0
Data 04	137.316, 1.00	0.1039	0.2078	100.0	30.0
	887.368, 5.64	0.969	1.938	100.0	1.0
Data 05	211.182, 1.36	1.015	2.030	100.0	0.0
	519.489, 1.68	0.1026	0.2052	100.0	25.7

Table 9: Loudest candidates of each cluster and corresponding probabilities of true positive (TP), false positive (FP) result.

The following figures 20 to 24 show the frequency-phase plots for each cluster. They can be used to estimate the error in frequency and phase for all of the loudest candidates, the results together with these errors are summed up in table 10 in the result section 5.

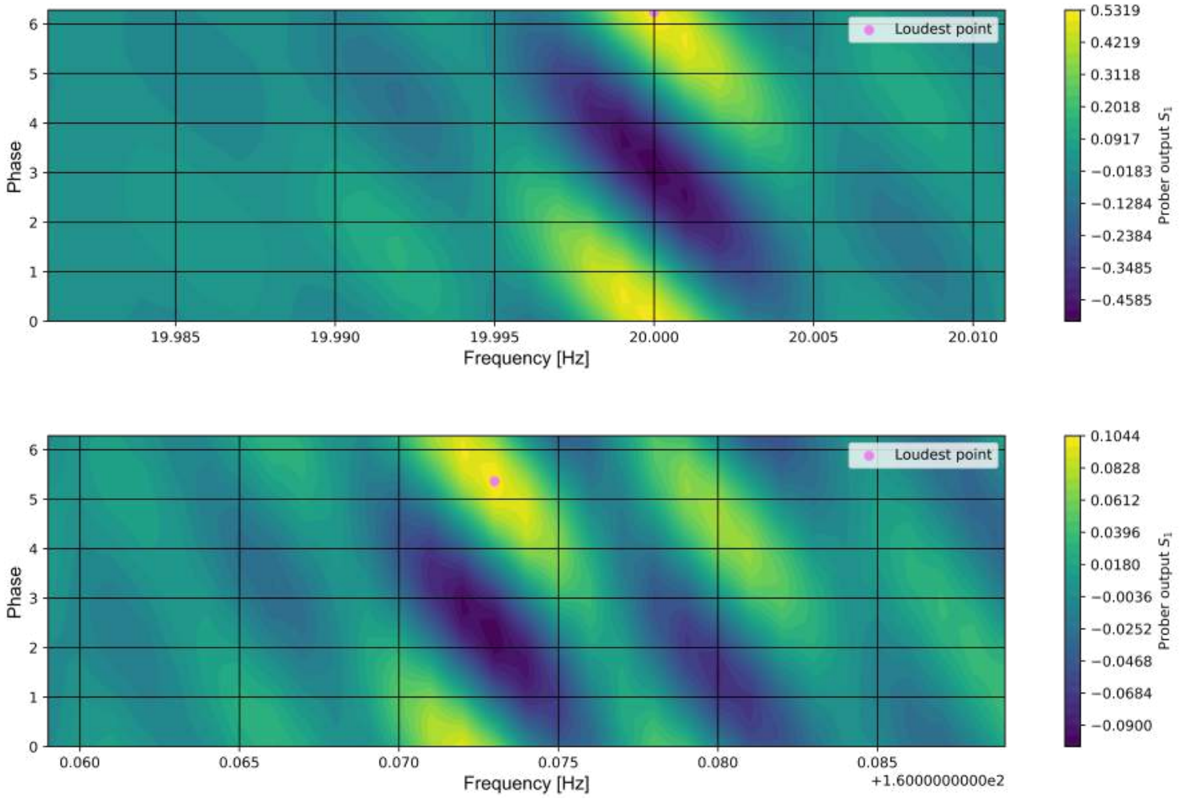

Figure 20: 2D contour plots for clusters of dataset 1.

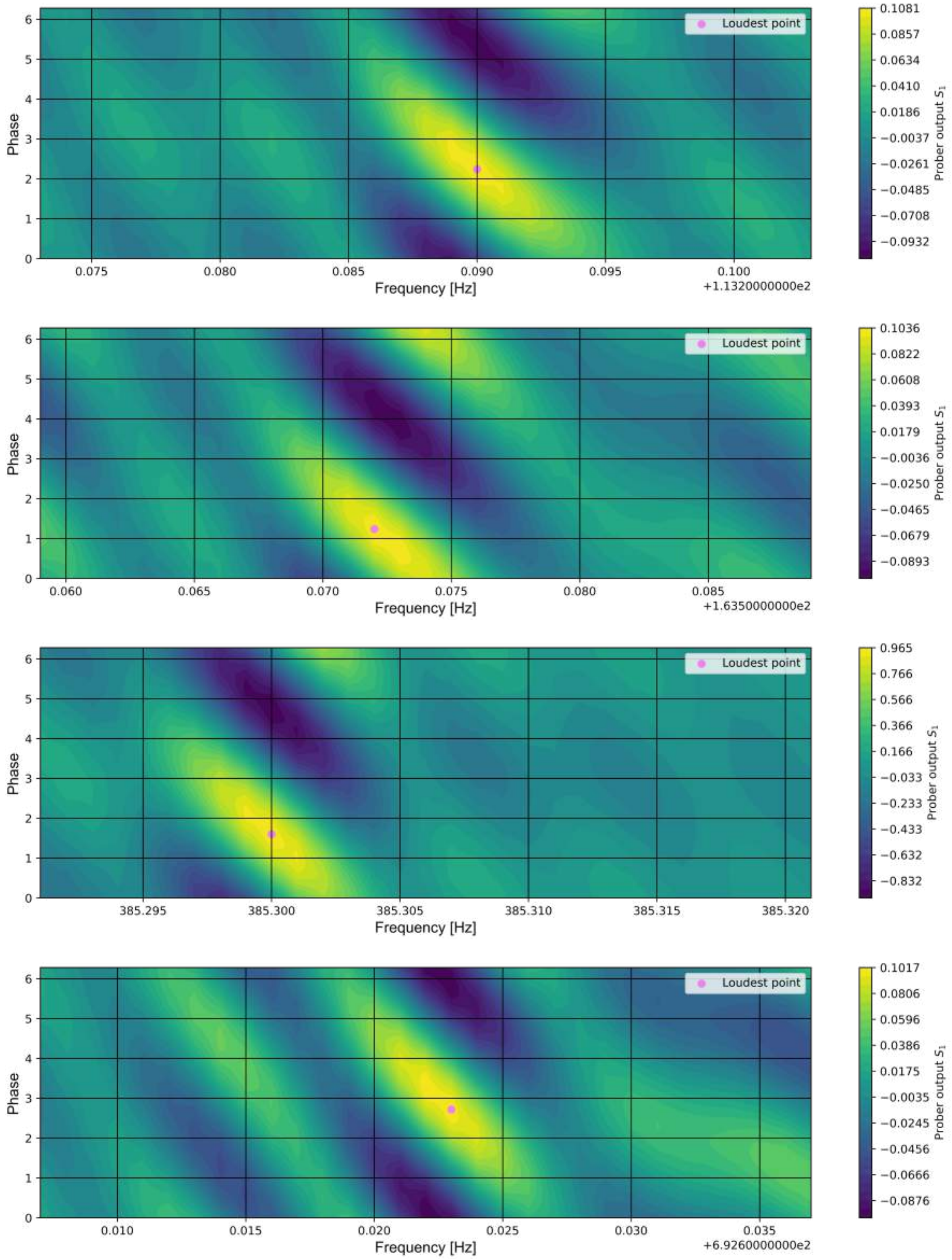
Loudest point
Phase
6
5
4
3
2
1
0
0.075
0.080
0.085
0.090
0.095
0.100
+1.1320000000e2
Frequency [Hz]
0.1081
0.0857
0.0634
0.0410
0.0186
−0.0037
−0.0261
−0.0485
−0.0708
−0.0932
Prober output S_1
Loudest point
Phase
0.060
0.065
0.070
0.075
0.080
0.085
+1.6350000000e2
Frequency [Hz]
0.1036
0.0822
0.0608
0.0393
0.0179
−0.0036
−0.0250
−0.0465
−0.0679
−0.0893
Prober output S_1
Loudest point
Phase
385.295
385.300
385.305
385.310
385.315
385.320
Frequency [Hz]
0.965
0.766
0.566
0.366
0.166
−0.033
−0.233
−0.433
−0.632
−0.832
Prober output S_1
Loudest point
Phase
0.010
0.015
0.020
0.025
0.030
0.035
+6.9260000000e2
Frequency [Hz]
0.1017
0.0806
0.0596
0.0386
0.0175
−0.0035
−0.0245
−0.0456
−0.0666
−0.0876
Prober output S_1

Figure 21: 2D contour plots for clusters of dataset 2.

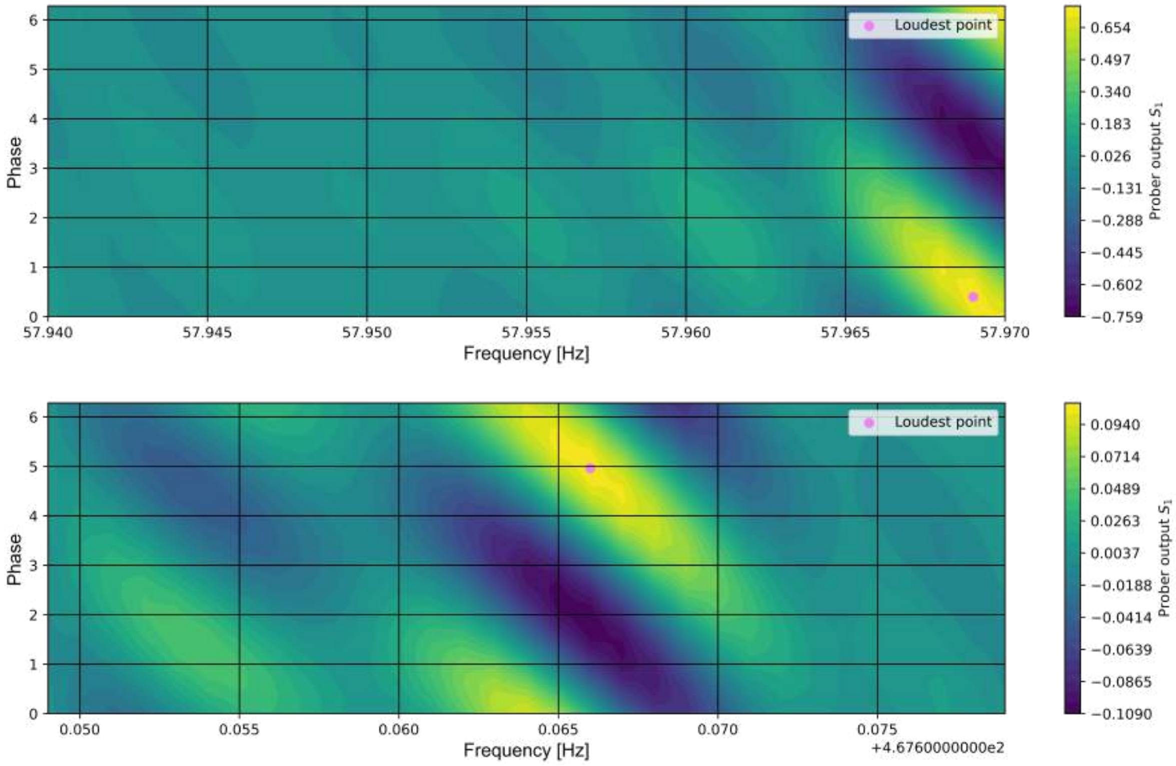

Figure 22: 2D contour plots for clusters of dataset 3.

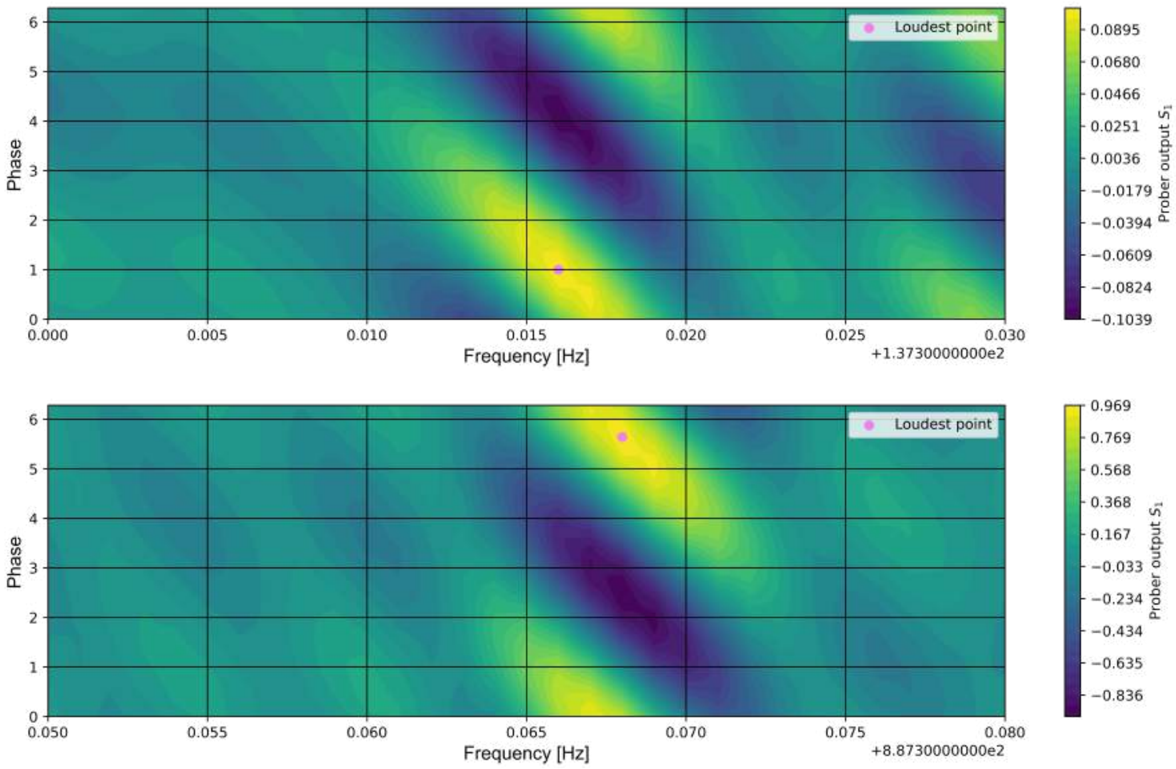

Figure 23: 2D contour plots for clusters of dataset 4.

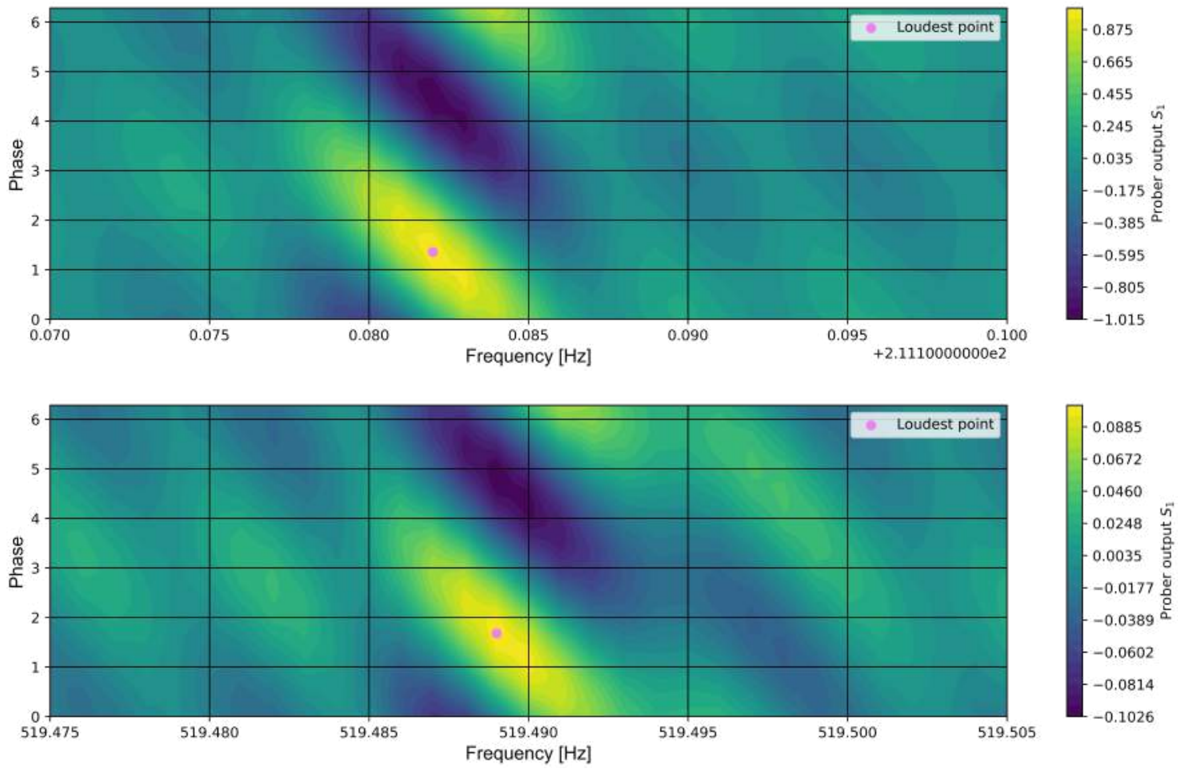

Figure 24: 2D contour plots for clusters of dataset 5.

5 Results

The signals detected in the five data sets are summarized in table 10, where the frequencies, phases and corresponding errors are listed. It should be noted that for some of those signals, the amplitude obtained by multiplying S_1 by two is not within the expected range of $A \in [0.5, 2.0]$.

Moreover, the first step of the search was done with method 3 and the threshold used to filter for areas of interest was only estimated by examining the average noise of similar data files. Therefore, any signal that was not within the selected areas of interest would be missing in this table. To achieve a higher probability of not missing any signals in the first step, method 1 could be used together with some of the fine template banks shown in section 3.3 if one can provide higher computational power.

Data set	Loudest candidate (f, ϕ)	S_1	Frequency error [Hz]	Phase error
Data 01	20.000, 6.24	0.5319	± 0.0015	± 1.5
	160.073, 5.36	0.1044	± 0.0025	± 1.5
Data 02	113.290, 2.24	0.1081	± 0.003	± 1.5
	163.572, 1.24	0.1036	± 0.0025	± 1.2
	385.299, 1.60	0.9654	± 0.002	± 1.1
	692.623, 2.72	0.1017	± 0.0015	± 1.2
Data 03	57.969, 0.40	0.7587	± 0.003	± 1.3
	467.667, 4.96	0.1090	± 0.0025	± 1.2
Data 04	137.316, 1.00	0.1039	± 0.002	± 1.1
	887.368, 5.64	0.969	± 0.003	± 1.2
Data 05	211.182, 1.36	1.015	± 0.002	± 1.2
	519.489, 1.68	0.1026	± 0.002	± 1.0

Table 10: Loudest candidates of each cluster and corresponding errors in frequency and phase.

6 Bibliography

References

[1] *Data Analysis Lab, Exercise 1.* Gottfried Wilhelm Leibniz Universität Hannover, 2022