# Title

Sorting Overlapped Spikes using Superimposed Template Matching

By Srdjan Sumarac

# Significant Contributions

Designed and evaluated an algorithm to improve classification of non-isolated neural spikes.

# Poster

To add later

# Executive summary

A problem that is still present in modern day spike sorting is the presence of overlapping spikes. Resolving these waveforms at a higher accuracy would significantly improve the decoding of neural populations in the brain. In this paper, an add-on module approach that is capable of classifying and resolving overlapping spikes automatically at a higher accuracy is proposed. After constructing template waveforms from initial clustering of well-isolated spikes, a superimposed template generated from pair-wise combinations of those templates at all possible time shifts from each other is then implemented. Subsequently, overlapping waveforms were identified using a weighted absolute difference template matching method, and each spike waveform is assigned to their respective overlapping template label, and then automatically translated to the equivalent single-unit combination. This method was examined using simulated data from the SYNTH MONOTRODE dataset and the performance was evaluated using the Spike Forester accuracy metrics such as precision, recall and overall accuracy. It was found that the add-on module was able to resolve overlapping waveform at an average of 54% across the whole study, with various signal-to-noise ratios, an improvement of 4% from standard methods without the module. TALK ABOUT PRECISION AND RECALL TOO… As a result, this module is easily added as an extra step to improve standard low-dimensional clustering techniques such as k-means, dbscan, etc, in order to boost performance.

# Introduction

Neurons are the fundamental units of the brain that transmit and receive information through action potentials. These brief voltage spikes can be recorded with devices such as microelectrode arrays, silicone electrodes and tetrodes (REFERENCES) the enable the detection of large number of action potentials simultaneously. Being able to sort these spikes into their corresponding neuron is crucial for (ADD response variability, correlation, latencies, and information rates, REFERENCE). However, spike sorting has been a challenge for many years, and there is an increasing need to develop reliable spike sorting algorithms that are able to accurately distinguish neurons from background electrical noise and other spikes. It is common for neurons to fire simultaneously and alter the detected voltage waveforms. This results in spikes that significantly deviate in shape and amplitude from single-unit waveforms. These overlapped waveforms pose a significant problem for fully automated spike sorting algorithms due to the limiting constraints of available methods. Typical methods that have been proposed in literature to deal with overlapping spikes in extracellular recordings involve creating combinations of previously identified single-unit waveforms. These template waveforms are then used to measure the similarity between the detected spike and the template combination. There are many proposed methods that are used to find the best match, such as (amplitude values, Fourier coefficients, ML techniques such as SVM, decoupled parameter estimation algorithm, subtraction of templates from overlaps, Neural Networks). In each of these methods, a certain threshold needs to be determined to find the best matching template. These methods also tend to be slow, since comparisons of all potential overlapping waveforms are needed. (TALK ABOUT DIFFERENT METHODS). Methods that do not involve template matching focus on extracting robust features other than amplitude to cluster overlapping waveforms with single-unit waveforms. Due to the advancements in high density multielectrode arrays with much smaller spacing between electrodes have taken advantage of the fact that the activity of one neuron appears on several different electrodes. In order to improve some of the above methods, Superimposed Template Matching (STM) was proposed as an automated method to resolve overlapping waveforms. Single-unit templates are generated using any popular spike sorting technique, such as K-means, and then using the superposition of pair-wise combinations of single unit templates at all possible time shifts from each other, it is able to sort overlapping spikes. This method has been designed to be added as an in-between step for any standard spike sorting methods, with the advantage that it can improve overlapped spike accuracy. The method is fully automated and parameter less and unsupervised and can successfully improve the performance of spike sorting in simulated data.

# Literature Review

## Review of literature surrounding the research and how my work is positioned within the field of research

# Overview

The traditional approach to spike sorting is illustrated in Figure X.

A screenshot of a cell phone

Description automatically generated

Raw data is initially bandpass filtered and spike detection is usually performed using an amplitude threshold. The spikes are then extracted from the filtered signal and relevant features of the spike shapes are extracted, which consequently become the input to a clustering algorithm that usually performs classification of spikes into their single unit equivalent. The proposed STM model is an add-on step to any traditional spike sorting algorithm after initial classification. It can be broken down into the following main steps

1. Construction of single-unit templates
2. Generation of overlapping spike templates
3. Matching an identified overlapping waveform to the closest template match based on weighted absolute difference.

These methods will be described in the following sections in detail.

# Detailed Discussions

## Spike Detection

Initially, the raw data is bandpass filtered from 300 to 3000 Hz and used as an input the spike sorting algorithm. This allows for much easier spike visualisation and detection in the presence of noisy background activity. Spike peaks were then detected using an automatic amplitude threshold that was set as a multiple of an estimate of the median absolute deviation (MAD) of the filtered signal. This robust method proposed by (REF Quiroga) assumes noise is gaussian in nature and since the spike components do not affect the median deviation, i.e., median(|X|) = median(|N|). Even if the noise distribution deviates from Gaussian, studies have shown that MAD yields better results than using standard deviation estimates (REF). Therefore, the estimate used to compute the threshold is



Four times the standard deviation of the background noise is commonly used to set the threshold (REF), and this value was also used in this study. The value of this multiple was carefully considered. If the multiple was set too small, noise fluctuation would lead to more false positive events whereas if the multiple was set too large, low-amplitude spikes will be missed. As a result, the peaks and time stamps of any spike that exceeded this threshold was stored as a vector. A picture containing antenna, object

Description automatically generated

After peak detection, the next step in the algorithm was to extract all data points around each peak for a specified window size. The window size was initially set at 3 ms and later cut to 1.5 ms. The sampling rate set for the datasets was 24 kHz, so this would be corresponding to a spike waveform with 36 samples in total. Since each extracted waveform was aligned at the peak, further alignment algorithms were not required. Spike extraction could have further been improved using energy operators, continuous wavelet transform, cubic spline simplification or probability theories (REF). However, since this study focuses on overlapped waveform classification, these further improvements were omitted for simplicity.

A picture containing drawing

Description automatically generated

## Initial Clustering

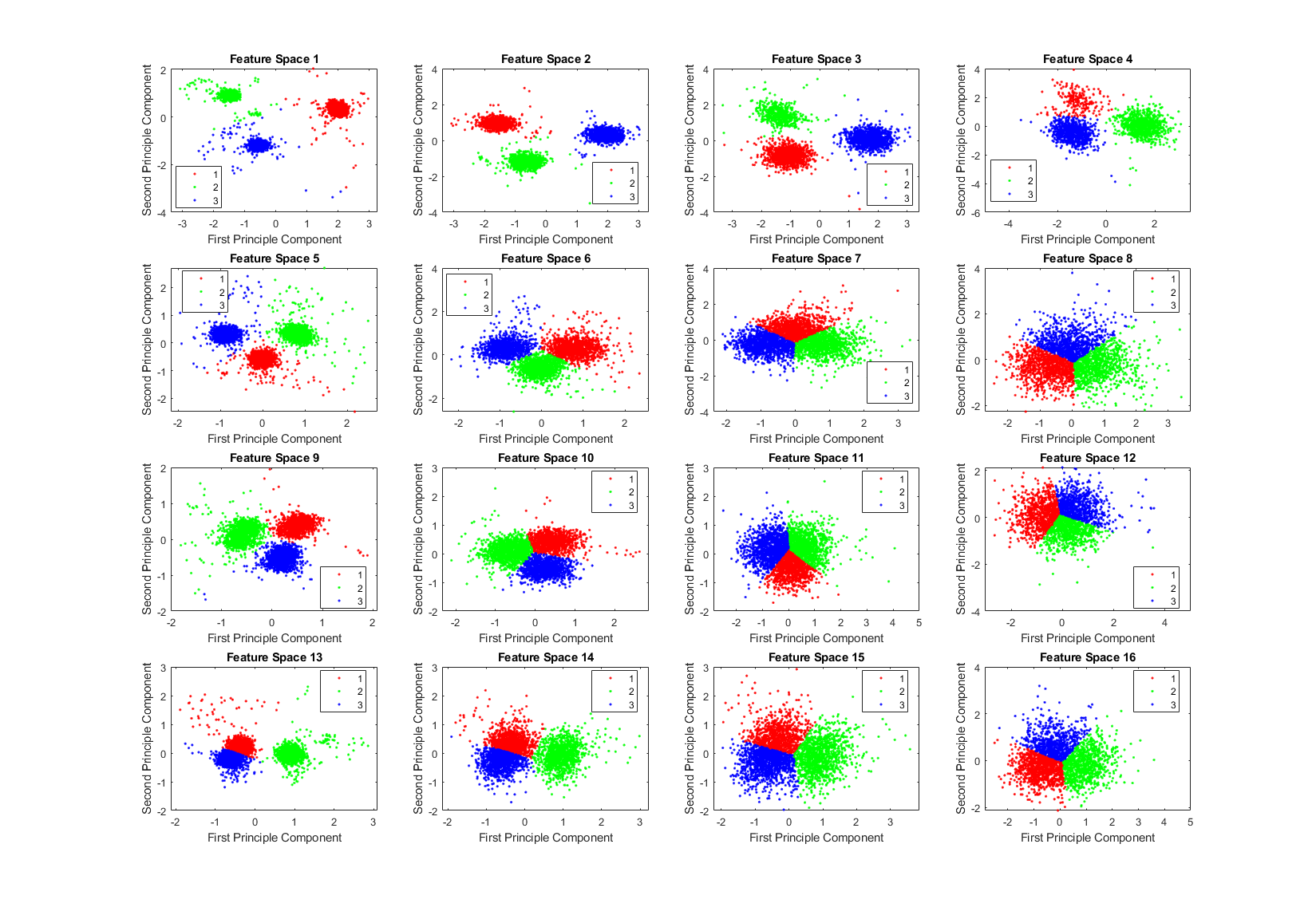
An effective way to extracting features from spike waveforms is known as principle component analysis (PCA). The logic behind PCA is that it gives an ordered set of orthogonal basis vectors that capture the directions in the data of largest variation. In order to represent a single spike, the principle components are scaled and added together and the scale factor for each component is known as the score. Since majority of the variance can be explained by the ﬁrst two principle components, only the first two principle components were used.

For both identifying the single unit templates and overlapping waveforms, the K-means algorithm was used, a clustering algorithm that attempts to split a given dataset into a fixed number of clusters. K-means is easily used to cluster high dimensional data collected from single and multichannel recording electrodes. K-means only needs the number of clusters as its parameter, and for the dataset used in this study, the number of clusters were set to three. As the focus of this paper is on improving overlapped spike classification accuracy, the choice is clustering method is not critical. It should be noted other clustering techniques such as density based clustering and fuzzy clustering were also tested during evaluation, and both methods saw performance improvements with STM.

Figure X shows how the clustered spikes from the low dimensional feature space match to those spikes at high dimensional feature space. As we can see, clusters that occur close to the centroid represent well isolated spikes. On the contrary, spikes that have a large deviation from the centroids represent spikes that deviate significantly in shape. Many of these large deviated spikes represent overlapping spike waveforms, in which traditional clustering methods fail to classify at a high accuracy. Figure X shows that as the noise increases, the clusters move closer together, and the ability to classify accurately becomes significantly more difficult.

A close up of a map

Description automatically generated



## Single Unit Template Construction

The first step that was considered in the design of STM was to identify the single unit templates that are required to build the overlapping waveforms. To find these single units, the spikes that were initially clustered using K-means and dense clusters were carefully inspected. Using the output labels from K-means classification, the median waveform for each cluster waveform was computed and used as a single unit template. Any spikes that deviated significantly from the cluster centroid were therefore removed as outliers, and any clusters with very small amplitudes were also not considered as they most likely were the result of multi-unit activity or noise. Figure X shows three clusters where each cluster has a corresponding single unit template waveform.

A close up of a map

Description automatically generated

## Generation of Overlapped Spike Templates

The overlapped templates were then generated using the single unit templates described in the section above. To do this, for each single unit template, a second unit template was shifted relative to the first and linearly superposed together at each sample in the window size. Since each single template was initially extracted at 3ms (72 samples), each superposition iteration started at the 18th sample and ended at the 54th sample, with the rest of values set at zero. The superposed template was then cut back to 36 samples to eliminate the superimposed zeros. This is done so that it was possible to directly match a 36-sample overlapped waveform with a 36-sample overlapped template. As a result, these new waveforms that have been generated can be referred to as synthetic waveforms. Figure X shows the single unit templates and the corresponding phase shift between the two single unit waveforms. Using the superposition principle, these two phase-shifted waveforms are added together to produce a synthetic overlapping waveform.

A close up of a map

Description automatically generated

As we can see in Figure X, two different single unit templates were linearly added together, with each waveform being phase shifted at every sample. It was assumed that it was unlikely that more than two neurons would fire simultaneously, and as a result the error associated with not sorting these overlaps would be small. Thus, in the current implementation of STM only considered pair-wise combinations of template waveforms. Although it may be possible to extend STM to more than two combinations, this becomes much more complicated due to the significant increase in the number of combinations, and the increase in likelihood that different combinations can create similar overlapping waveforms.

## Template Matching

The final stage in the STM module is to compare the overlapping waveforms with all the synthetic waveforms generated from the superimposed single unit templates. To do this, each detected spike waveform was weighted based on the features that were most significant in differentiating spike shapes. Upon experimentation, it was found that the samples between 9 and 27 from the 36-sample window were the most prominent features to consider. Therefore, those samples were multiplied by a large factor of 1000 to significantly bias the matching procedure. Then, the weighted absolute difference was computed for all possible pairs of overlapping waveforms. The synthetic waveform with the lowest minimum absolute difference was selected as the best match and assigned to a particular single unit cluster. Other similarity measures were attempted such as Pearson correlation coefficient, and sum-squared error was considered, but it was found that the weighted absolute difference has the quickest run time and returned the best overall results. Also, iterative subtraction of the single-unit templates was also considered in the template matching pipeline, however, this approach leads to higher matching errors due to the noise generated by the inaccuracies of the templates. The results from template matching are shown in Figure X, showing example overlapping waveforms (coloured) and the best matching synthetic waveforms (black).

A close up of a map

Description automatically generated

# Evaluation

The recordings used in the evaluation of STM consisted of four recordings; Easy1, Easy2, Difficult1, and Difficult2. Each recording was 60 seconds long in duration, sampled at 24 kHz and consisted of waveforms from three distinct neurons. Easy1 and Easy2 made up of clearly distinct waveforms whereas Difficult1 and Difficult2 consisted of waveforms that were more similar in shape. Within each of these datasets, the algorithm will be evaluated at increasing levels of noise, ranging from standard deviations of 0.05 to 0.2 times the spike amplitude. The details on how these simulated data were generated can be found in Quiroga et al. (2004).

The performance of STM was tested against the against the accuracy metrics from SpikeForest, an open-source software that benchmarks the performance of automated spike sorting algorithms across an extensive range of ground truth data, where the solution of each cluster is known. Therefore, the spike sorter output label and timestamps were compared to the ground truth timings and labels, and the number of matches to counted. An acceptable firing time error was set at 1 ms, which was assumed to be shorter than half the refractory period of any true neuron. The assumption was that at most one spike can be matched within this firing time error range. Consequently, the number of missed events and the number of false positives can therefore be counted. Depending on the experimental demands, false negatives (missed events) and false positives may have different relative importance. Therefore, the metrics proposed in SpikeForest allows for switching between three evaluation metrics for the comparison with ground truth:

1. Precision (which penalises only false positives)

A picture containing knife

Description automatically generated

1. Recall (which penalises only false negatives),

A picture containing knife

Description automatically generated

1. An overall accuracy metric (which balances the two).

A screenshot of a cell phone

Description automatically generated

For each ground truth unit, the best matching unit k to the ground truth firing time is defined as the one with the highest accuracy. Moreover, the signal-to-noise ratio (SNR) is computed from the filtered records and calculated by comparing the peak of each spike to the median absolute deviation of the signal noise. This metric was used to determine how well the algorithm performs for different SNR values.

Furthermore, it was in the interests to determine the accuracy of overlapped waveforms. So, the inter-spike interval (ISI) histogram was inspected to isolate spikes with large numbers of refractory violations where the ISI is smaller than a specified refractory period of 1ms. These spikes are then considered as a combination of multiple waveforms from different neurons and are thus considered overlapping spikes. Single units that fire relatively far apart in time from each other result in well isolated single unit spikes. On the other hand, when two single units fired very close to each other in time, this resulted in overlapping waveforms deviating significantly from either of the single spikes. Figure X shows the ISI for the combined datasets, and the fitted distribution. From this we can deduce that ISI is exponentially distributed with a mean of 1/19.0585 = 0.525, and so Pr(ISI < 1ms) = 0.0511 = 5.11%

A screenshot of a social media post

Description automatically generated

# Results

The sorting results from the simulated datasets from the ‘SYNTH MONOTRODE’ were split up into two sections. The first evaluation was performed without the incorporation of the STM module. Then, holding all parameters constant, the STM module was added to the pipeline and the combined results were recorded. In order to find out how well STM improves performance, the algorithm was tested across various datasets, which can be seen in figure X.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Without STM** | | **With STM** | |
| **Dataset** | **SNR** | **No. Spikes** | **No. Matches** | **False Positives** | **No. Matches** | **False Positives** |
| C\_Easy1\_noise005 | 16 | 3514 | 3438 | 33 | 3447 | 24 |
| C\_Easy1\_noise01 | 9 | 3522 | 3426 | 17 | 3430 | 13 |
| C\_Easy1\_noise015 | 6 | 3477 | 2796 | 24 | 2793 | 27 |
| C\_Easy1\_noise02 | 6 | 3474 | 1866 | 55 | 1891 | 30 |
| C\_Easy2\_noise005 | 19 | 3410 | 3340 | 64 | 3369 | 35 |
| C\_Easy2\_noise01 | 10 | 3520 | 3364 | 156 | 3383 | 137 |
| C\_Easy2\_noise015 | 7 | 3411 | 2885 | 525 | 2998 | 412 |
| C\_Easy2\_noise02 | 6 | 3526 | 1956 | 830 | 2300 | 486 |
| C\_Difficult1\_noise005 | 18 | 3383 | 3298 | 71 | 3337 | 32 |
| C\_Difficult1\_noise01 | 9 | 3448 | 3145 | 281 | 3184 | 242 |
| C\_Difficult1\_noise015 | 7 | 3472 | 2378 | 993 | 2809 | 562 |
| C\_Difficult1\_noise02 | 6 | 3414 | 1302 | 1162 | 1662 | 802 |
| C\_Difficult2\_noise005 | 18 | 3364 | 3191 | 180 | 3295 | 76 |
| C\_Difficult2\_noise01 | 10 | 3462 | 2706 | 763 | 2847 | 622 |
| C\_Difficult2\_noise015 | 7 | 3440 | 2402 | 988 | 2483 | 907 |
| C\_Difficult2\_noise02 | 6 | 3493 | 1747 | 940 | 1936 | 751 |
| **Average** |  |  | **2702.50** | **442.63** | **2822.75** | **322.38** |

It can be seen that the STM module improves spike sorting performance across various signal-to-noise ratios. On average, the STM module improve sorting accuracy by 4.6%, with the average accuracy increasing from 71.4% to 76.0%. Both recall and precision are also improved by a similar margin. It is also worth noting that this performance increase is more significant for datasets with lower SNR values, suggesting that the STM module is more robust towards higher noises. Furthermore, the performance can be compared with other spike sorting algorithms proposed on the SpikeForest website.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Without STM** | | **With STM** | |
| **Dataset** | **SNR** | **Precision** | **False Positive Rate** | **Precision** | **False Positive Rate** |
| C\_Easy1\_noise005 | 16 | 99.0% | 1.0% | 99.3% | 0.7% |
| C\_Easy1\_noise01 | 9 | 99.5% | 0.5% | 99.6% | 0.4% |
| C\_Easy1\_noise015 | 6 | 99.1% | 0.9% | 99.0% | 1.0% |
| C\_Easy1\_noise02 | 6 | 97.1% | 2.9% | 98.4% | 1.6% |
| C\_Easy2\_noise005 | 19 | 98.1% | 1.9% | 99.0% | 1.0% |
| C\_Easy2\_noise01 | 10 | 95.6% | 4.4% | 96.1% | 3.9% |
| C\_Easy2\_noise015 | 7 | 84.6% | 15.4% | 87.9% | 12.1% |
| C\_Easy2\_noise02 | 6 | 70.2% | 29.8% | 82.6% | 17.4% |
| C\_Difficult1\_noise005 | 18 | 97.9% | 2.1% | 99.1% | 0.9% |
| C\_Difficult1\_noise01 | 9 | 91.8% | 8.2% | 92.9% | 7.1% |
| C\_Difficult1\_noise015 | 7 | 70.5% | 29.5% | 83.3% | 16.7% |
| C\_Difficult1\_noise02 | 6 | 52.8% | 47.2% | 67.5% | 32.5% |
| C\_Difficult2\_noise005 | 18 | 94.7% | 5.3% | 97.7% | 2.3% |
| C\_Difficult2\_noise01 | 10 | 78.0% | 22.0% | 82.1% | 17.9% |
| C\_Difficult2\_noise015 | 7 | 70.9% | 29.1% | 73.2% | 26.8% |
| C\_Difficult2\_noise02 | 6 | 65.0% | 35.0% | 72.1% | 27.9% |
| **Average** |  | **85.3%** | **14.7%** | **89.4%** | **10.6%** |

A screenshot of a cell phone

Description automatically generated

The STM module was able to resolve single-unit and overlapping waveforms with a high degree of accuracy.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Without STM** | | **With STM** | |
| **Dataset** | **SNR** | **No. Spikes** | **No. Matches** | **False Positives** | **No. Matches** | **False Positives** |
| C\_Easy1\_noise005 | 16 | 259 | 125 | 25 | 134 | 16 |
| C\_Easy1\_noise01 | 9 | 241 | 112 | 22 | 116 | 18 |
| C\_Easy1\_noise015 | 6 | 262 | 72 | 20 | 79 | 13 |
| C\_Easy1\_noise02 | 6 | 264 | 22 | 8 | 29 | 1 |
| C\_Easy2\_noise005 | 19 | 256 | 148 | 79 | 176 | 51 |
| C\_Easy2\_noise01 | 10 | 285 | 150 | 95 | 183 | 62 |
| C\_Easy2\_noise015 | 7 | 242 | 109 | 135 | 130 | 114 |
| C\_Easy2\_noise02 | 6 | 224 | 67 | 112 | 87 | 92 |
| C\_Difficult1\_noise005 | 18 | 249 | 172 | 42 | 206 | 8 |
| C\_Difficult1\_noise01 | 9 | 243 | 147 | 38 | 167 | 18 |
| C\_Difficult1\_noise015 | 7 | 262 | 92 | 39 | 110 | 21 |
| C\_Difficult1\_noise02 | 6 | 243 | 19 | 13 | 20 | 12 |
| C\_Difficult2\_noise005 | 18 | 265 | 140 | 121 | 179 | 82 |
| C\_Difficult2\_noise01 | 10 | 204 | 116 | 104 | 137 | 83 |
| C\_Difficult2\_noise015 | 7 | 264 | 113 | 62 | 120 | 55 |
| C\_Difficult2\_noise02 | 6 | 259 | 49 | 45 | 55 | 39 |
| **Average** |  |  | **103.31** | **60.00** | **120.50** | **42.81** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Without STM** | | **With STM** | |
| **Dataset** | **SNR** | **Precision** | **False Positive Rate** | **Precision** | **False Positive Rate** |
| C\_Easy1\_noise005 | 16 | 83.3% | 16.7% | 89.3% | 10.7% |
| C\_Easy1\_noise01 | 9 | 83.6% | 16.4% | 86.6% | 13.4% |
| C\_Easy1\_noise015 | 6 | 78.3% | 21.7% | 85.9% | 14.1% |
| C\_Easy1\_noise02 | 6 | 73.3% | 26.7% | 96.7% | 3.3% |
| C\_Easy2\_noise005 | 19 | 65.2% | 34.8% | 77.5% | 22.5% |
| C\_Easy2\_noise01 | 10 | 61.2% | 38.8% | 74.7% | 25.3% |
| C\_Easy2\_noise015 | 7 | 44.7% | 55.3% | 53.3% | 46.7% |
| C\_Easy2\_noise02 | 6 | 37.4% | 62.6% | 48.6% | 51.4% |
| C\_Difficult1\_noise005 | 18 | 80.4% | 19.6% | 96.3% | 3.7% |
| C\_Difficult1\_noise01 | 9 | 79.5% | 20.5% | 90.3% | 9.7% |
| C\_Difficult1\_noise015 | 7 | 70.2% | 29.8% | 84.0% | 16.0% |
| C\_Difficult1\_noise02 | 6 | 59.4% | 40.6% | 62.5% | 37.5% |
| C\_Difficult2\_noise005 | 18 | 53.6% | 46.4% | 68.6% | 31.4% |
| C\_Difficult2\_noise01 | 10 | 52.7% | 47.3% | 62.3% | 37.7% |
| C\_Difficult2\_noise015 | 7 | 64.6% | 35.4% | 68.6% | 31.4% |
| C\_Difficult2\_noise02 | 6 | 52.1% | 47.9% | 58.5% | 41.5% |
| **Average** |  | **65.0%** | **35.0%** | **75.2%** | **24.8%** |

Future Work

Try to implement overlapping techniques to multi-channel recordings due to the improvement of dense electrode arrays.

It can be seen from some overlapped synthetic templates that the phase shift is within a few samples, resulting in waveforms that only contain one peak. This remains a problem with overlapped spike detection, where another spike must be added to a single unit cluster, as the two spikes happened almost simultaneously at this time point.

# Conclusion

The source code used in this method (written in MATLAB) is available for download from GITHUB.

# References

Review of methods

<https://www-tandfonline-com.ezproxy.lib.monash.edu.au/doi/pdf/10.1088/0954-898X_9_4_001?needAccess=true>

Easy data

<https://www-sciencedirect-com.ezproxy.lib.monash.edu.au/science/article/pii/S0165027009004506>

Spike Forest

<https://www.biorxiv.org/content/10.1101/2020.01.14.900688v1.full.pdf>

# Appendices