12/9/22

Working on setup.py in root & pip install -e .
so tracking module can be imported within calibration
module.

combine ✓
inspect ✓ voltage rotations
smooth ✓ (check label)
detect ✓
cluster ✓
analyse ✓
inspect-zc ✓ z c cloud clusters
cluster-inliers ✓
analyse-inliers ✓
inspect inliers ✓

Working on setting SPARK_LOCAL_IP dynamically based upon whatever "Local-interface" exists with the spark config within package.json.

retesting all ✓ Full run used about 200 mb of network data.

Note ipv6 / v4 tinc issue, if on mobile network with only an ipv6 addr, created 2nd network for ipv6 (tinc.conf) & set to server ipv6 addr.
cloud-ok ✓

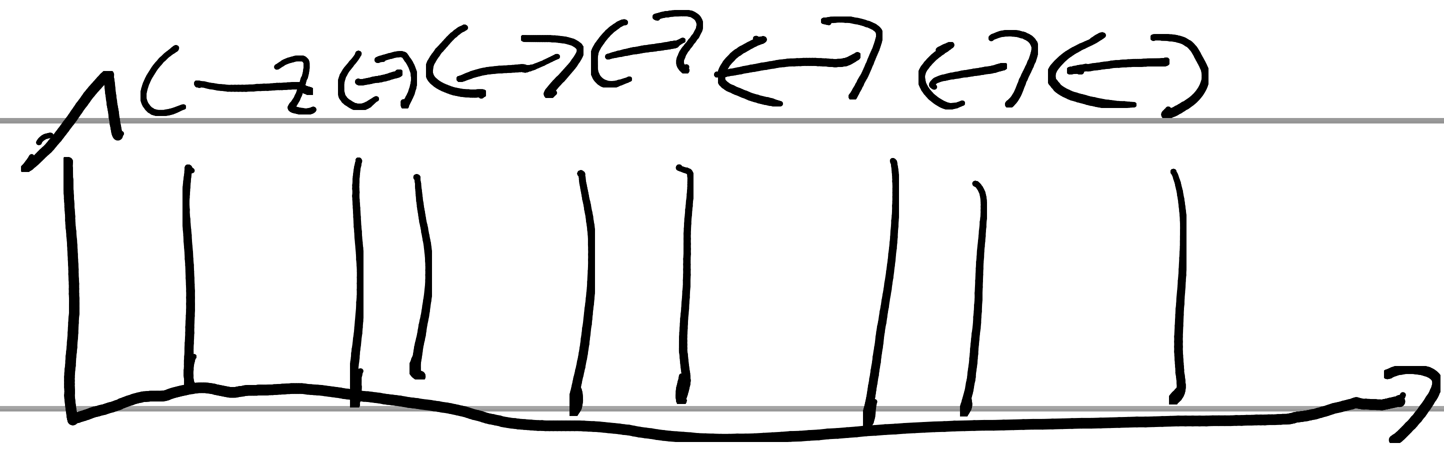
14/9/22

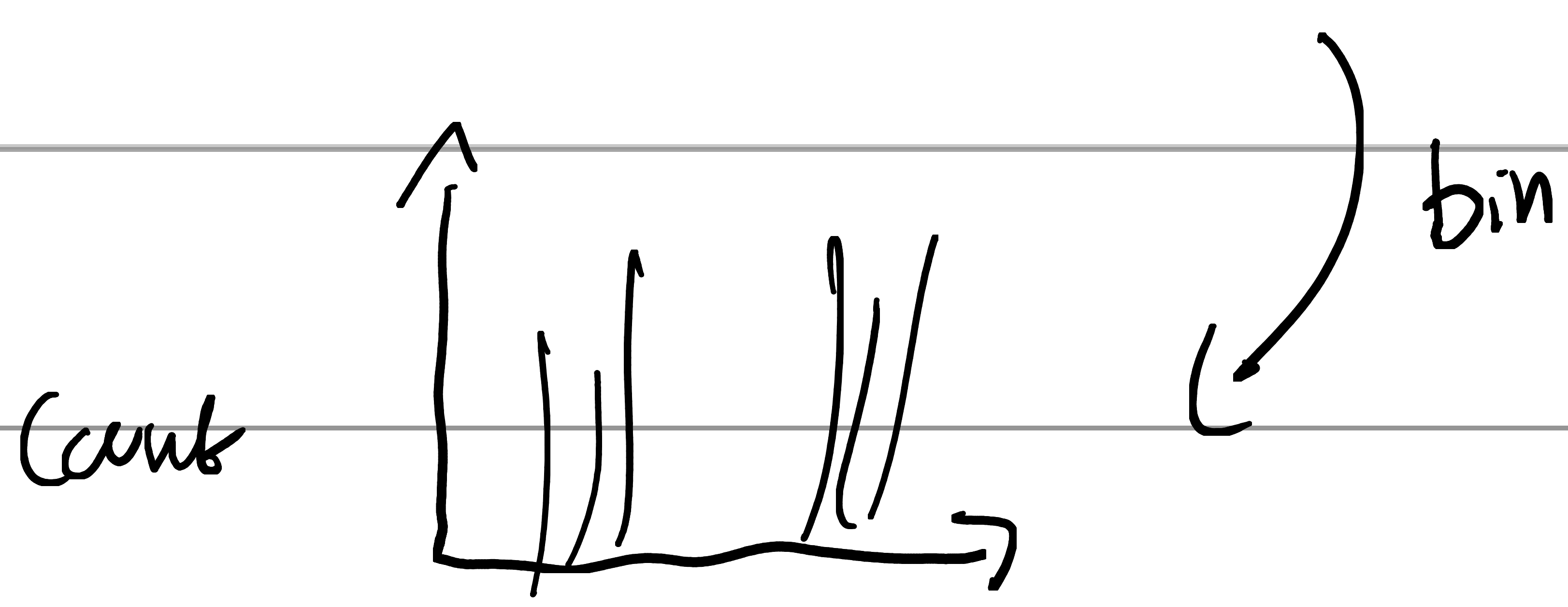
Adding more explanation comment on the graphs.

Binning binary pulse spike train in order to quantify periodicity.
FFT of a spike train is a noisy spike train.

Adding histograms for cluster density.

Need to determine each phases periodicity independently of the combined pulses (zero crossing events)





Consequitive pulse lag time.

Need a lib folder, factorise graph report into classes.

15/9/22

thinking about
do checks for rising vs falling. phase A, B, C
& rising vs falling.

Adding more checks to smooth

Made a class for report generation.

Need to fix smooth plots ~ un plot on to -50 on y axis

Add text description to the original cluster plot.

on the density plot +1 to each identifier.

Need to force network wilt capture to check for folder existences for a run. ~ need to test?

copy report card

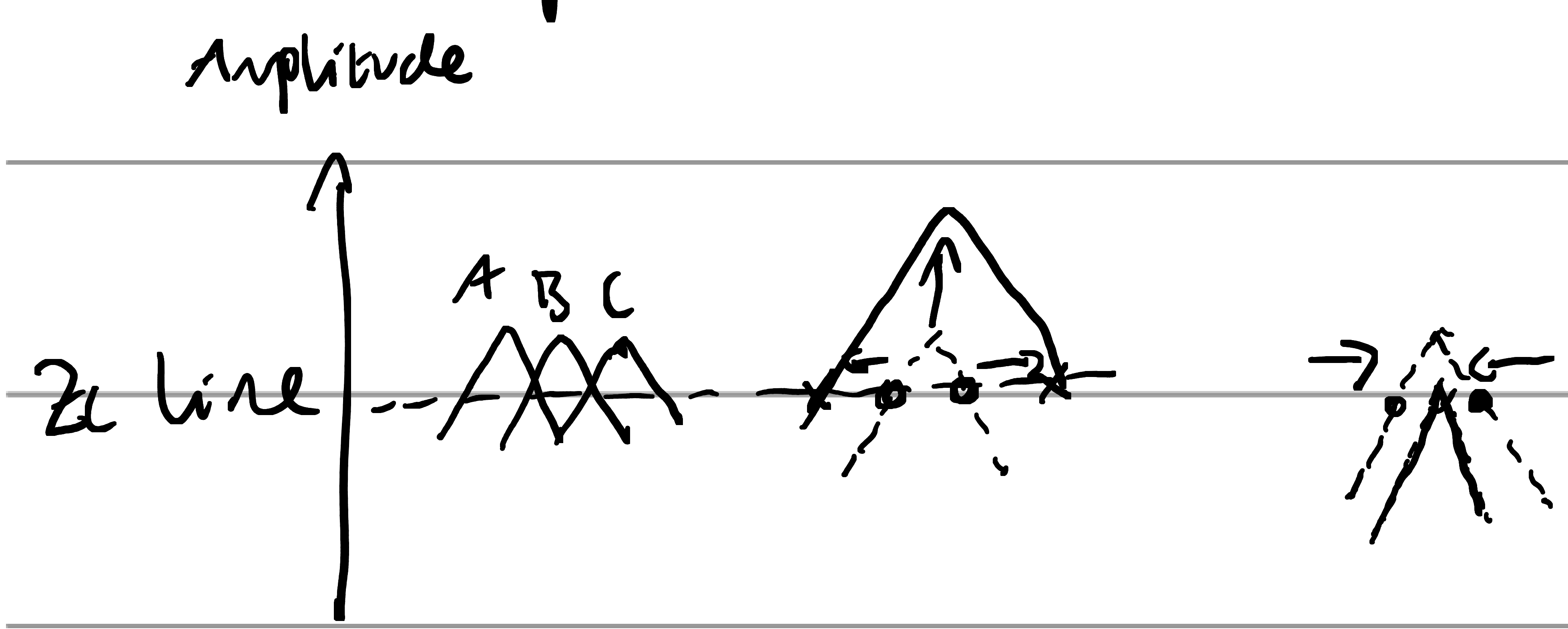
Need to rename temporal to spacial

Code is now producing nice reports, looking at the sept 2 dataset,

red phase is higher, periodicity of zero crossing has sort of 2 clusters of long, short, long, short. The distance from rising to falling edges (consequtive) is shorter then falling to rising edges which appear on average longer.

16/9/22

Plan today is to test both a forward (cw) & a backwards (ccw) calibration data capture & compare. With the periodicity pattern of cw (long, short, long, short), it appears there is some systematic error due to phase voltages not always having similar average value.



lower & raising each phase squeezes or stretches the ZC point along angle axis

Whatever is causing this drift away from ideal values, I hope with by taking experimental data in the reverse direction (ccw) that we can perhaps account for or even quantise the drift.

→ Fixed dir path for sync program.

Starting with a ccw run. "16 sept - ccw"

~3.4 seconds of data

99.86% match

~3 sizes of spacing

pattern check, distances past the first zc after angle step zero:

S, L, S, L .....

Attempting cw run:

Had some issues with bad runs low match rate, need to unplug the usb cleanly (should have a software defined method to terminate data collection) need to fix this later.

CW run "16-sept_4_cw":

~ 4.2 seconds of data

99.89% match

~ 3 clusters of spacing

pattern check, distances past the first ZC after angle step zero:

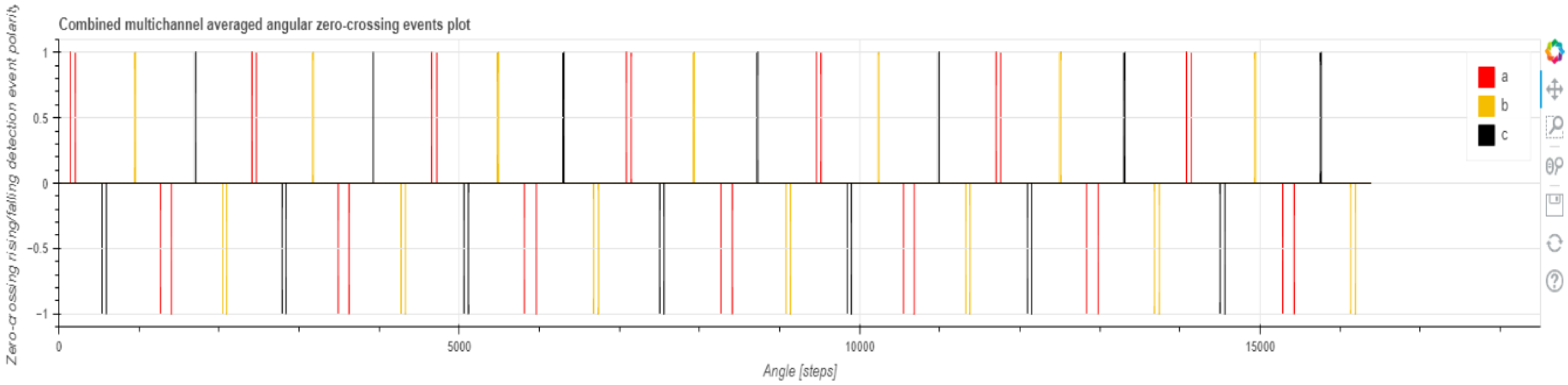
L, S, L, S, L, S

Definitely seeing some reversal of the pattern, now have 2 runs with great match rate. There certainly appear to be different groups of pulse timings, looking at the report spike train distances seem to be roughly 3.

Will superimpose the combined 3 phase zero-crossing spike plot.

Combined phase zero-crossing plot

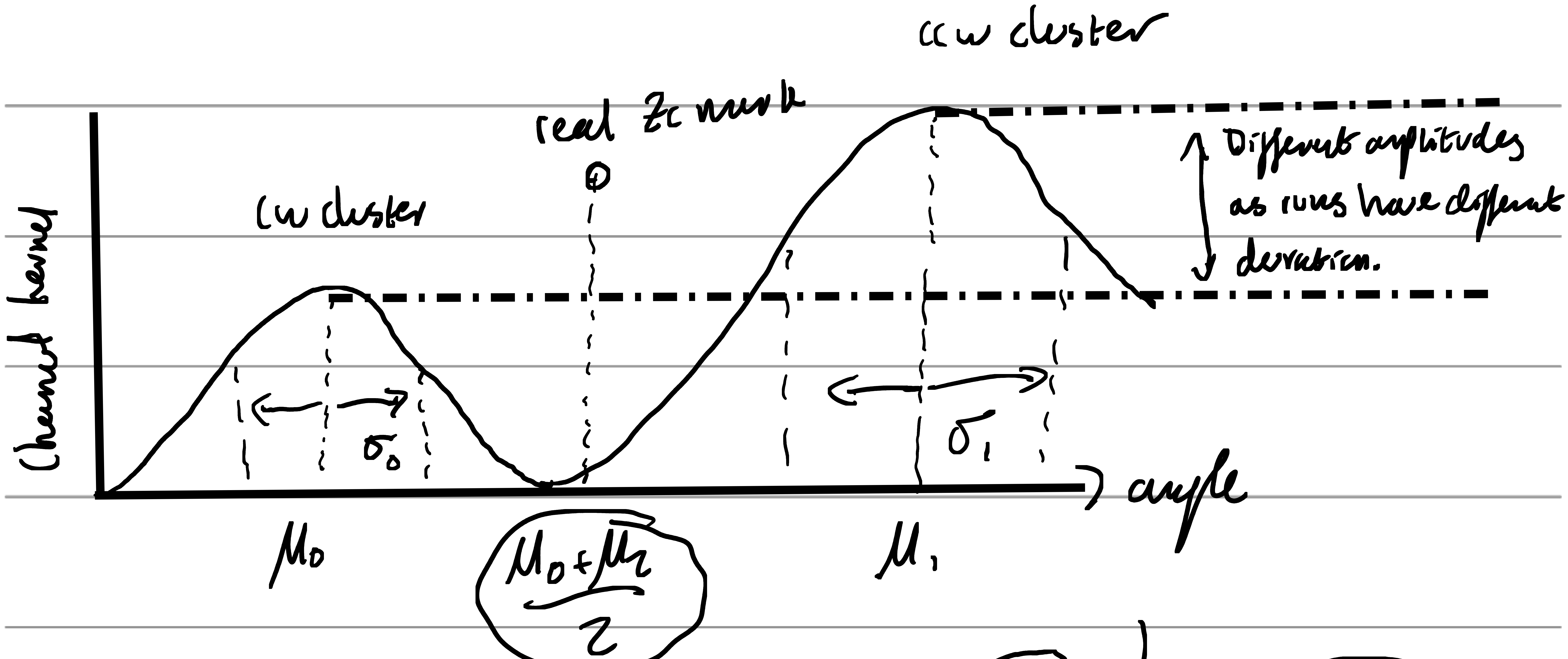
Post clustering, the mean of each channels zero-crossing channel detections is known. Therefore we can reduce the data so that there is only a single definitive pulse (rising or falling) for every cluster within each zero-crossing channel. The combined zero-crossing events are combined into a single plot. Note that whether or not a zero-crossing event is rising or falling is depicted by the polarity of the spike +1 indicates rising and -1 indicates falling for that phase.



Interesting combined plots, does appear like the mean value center of the combined cw/ccw plot is a better center, & we can now see the systematic drift indeed seems worse for phase A channel. Biggest discrepancy phase A falling then, A rising, then phase B (yellow) falling, phase C (black) falling, B (yellow) & C (black) rising seem pretty consistent in either direction.

Now the question is how to combined these results to reveal the true centers.

Consider merging the Zc histograms for the 6 channels for cw & ccw runs. Consider a single channel it might look like this:



It seems like calculating the means for both the ccw & cw clusters seperately is the best idea as the population of samples for each run is different & thus would artificially bias the center to the side which collected more samples. So we should take μ_0 & μ_1 as given with their σ_0 & σ_1 errors & combine them.

~ mean

Perhaps the best way to find the midpoint between μ_0 & μ_1 would be the circular mean, as it respects midpoint when passing the 16393 → 0 mark whereas the arithmetic mean will find wrong values if cluster angles span the zero mark.

The circular mean is calculated by thinking about angles of the unit circle, one can add these vectors end to end to find a resultant vector one where we do not care about the magnitude but the final resultant vectors angle should maximise the likelihood of the mean parameter & thus gives a good geometric mean.

e.g. hour of day = [0 .... 24]

hours sample = Random ([0 ... 24], 4)

convert hour samples into radians. $\arctan2(\bar{y}, \bar{x}) \cdot \frac{2^{4}}{2\pi} \% 2^{4}$

$$\bar{x} \quad \text{mean}\left(\frac{2\pi}{2^{14}} W_i \sin(Y_i)\right)$$

hours-sample = $2\pi \, \frac{\text{hours\_sample}}{24}$

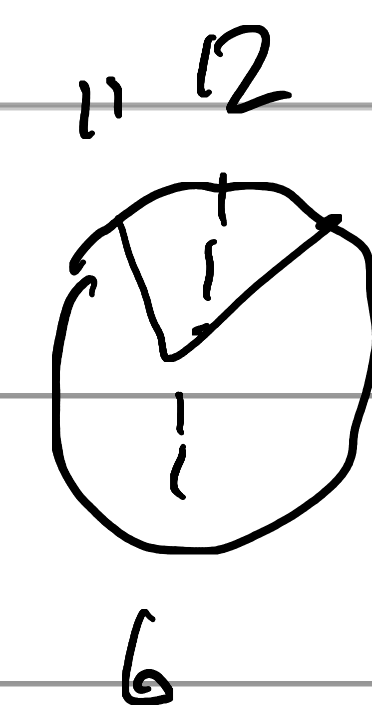
$$\bar{y} \quad \text{mean}\left(\frac{2\pi}{2^{14}} W_i \cos(Y_i)\right)$$

hours-sample-x = sin(n) for n in hours_sample

hours-sample-y = cos(n) for n in hours_sample

weighted version?

average x, y



$\bar{x}$ = mean(hours-sample-x)

$\bar{y}$ = mean(hours-sample-y)

$$\text{hours\_avg} = \left(\arctan2(\bar{y}, \bar{x}) \cdot \frac{24}{2\pi}\right) \% \ 24$$

2^{14}

we can have μ_0, μ_1 & now $\mu_{\frac{0+1}{2}} \% \ 16384$

So the errors for μ_0 & μ_1 are σ_0 & σ_1 respectively.

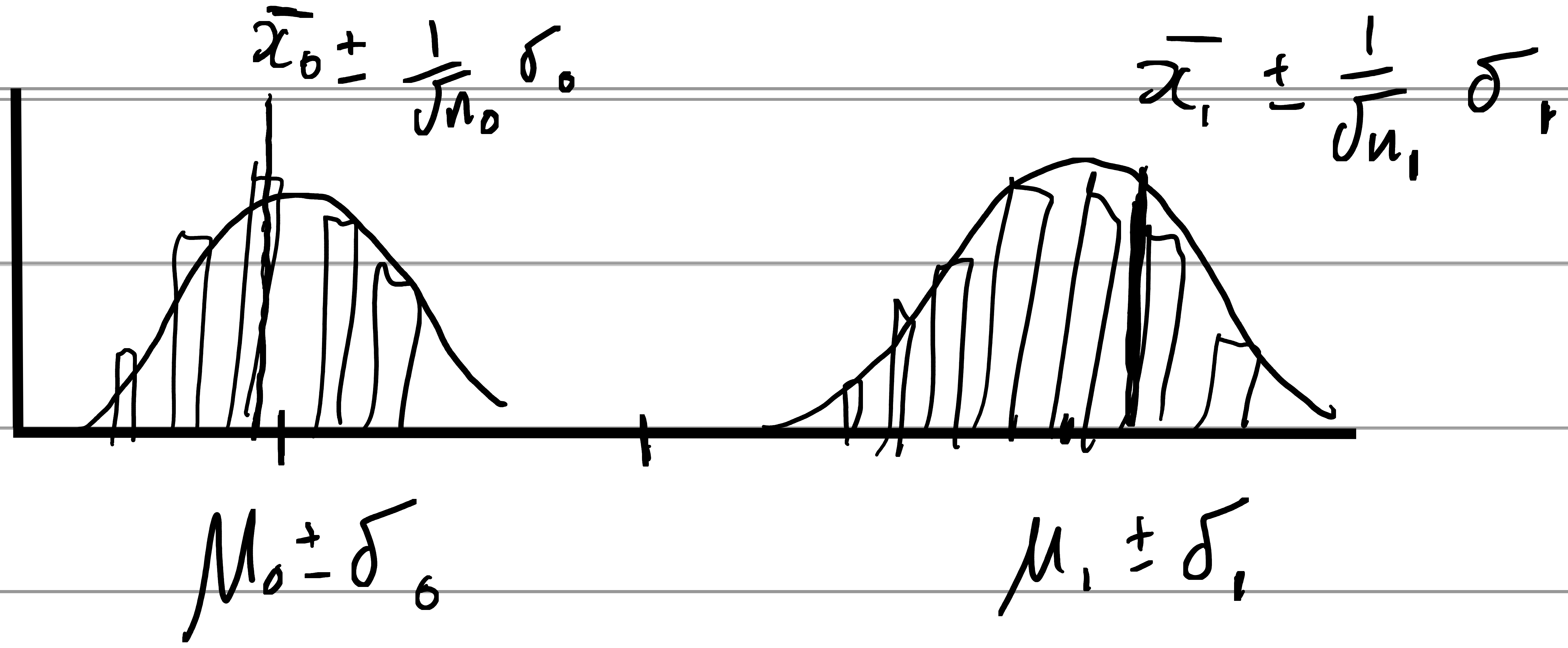
Circular mean does deviate from arithmetic the wider the distribution of angles but is approximately $\frac{1}{N}\sum_{1}^{N} x_i$, this is the form of simple addition of each angle count, thus

$$\mu_{combined} \sim \frac{1}{2}(\mu_0 + \mu_1)$$

when adding variables with errors you would sum the absolute errors

$$\sigma_{combined} \sim \sigma_0 + \sigma_1$$

Worth a try, the circular mean & combined error.



Variance of combined sample mean $\frac{1}{2}(\bar{x}_1 + x_2) = \left(\frac{\sigma_0^2}{n_0} + \frac{\sigma_1^2}{n_1}\right)\frac{1}{4}$

Standard deviation $\frac{1}{2}\sqrt{\frac{\sigma_0^2}{n_0} + \frac{\sigma_1^2}{n_1}}$

2nd analysis for each cluster we have μ_i, σ_i & n_i

population

For each group i $\quad \sum_{j=1}^{n_i} x_{ji} = \mu_i \cdot n_i \quad (1)$

N lots of i clusters

$$\sum_{j=1}^{n_i} x_{ji}^2 = \sigma_i^2 (n_i - 1) + \frac{1}{n_i} \sum_{j=1}^{n_i} x_{ji}^2 \quad (2)$$

sub (1) + (2)

$$\sum_{j=1}^{n_i} x_{ji}^2 = \sigma_i^2 (n_i - 1) + \frac{1}{n_i} \sum_{j=1}^{n_i} (\mu_i n_i)^2$$

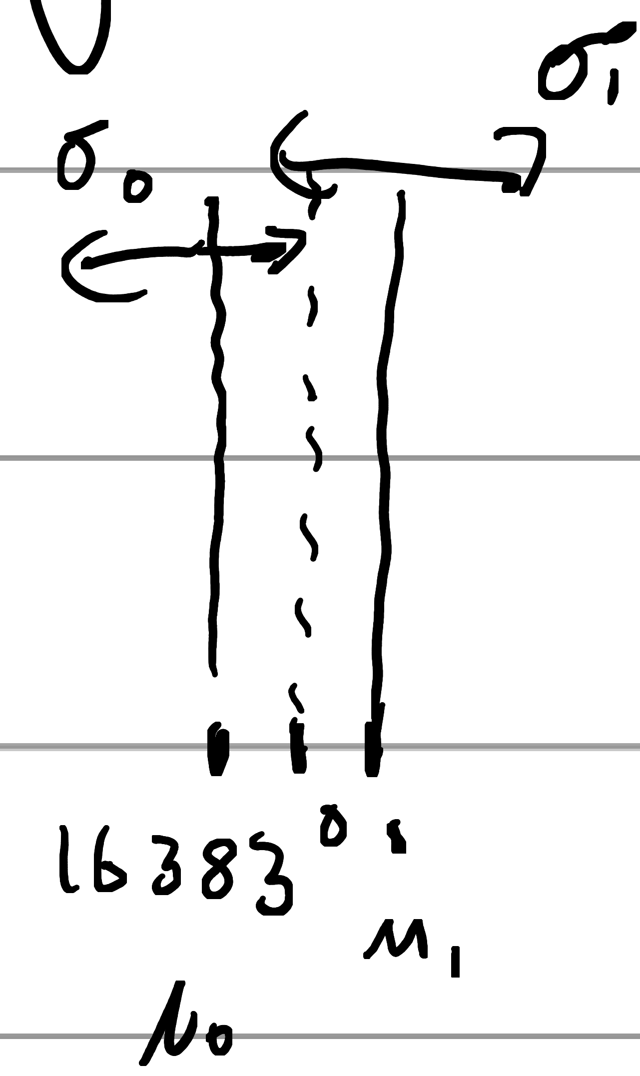
$$S_n = \sum_{i=1}^{N} n_i \qquad S_x = \sum_{i=1}^{N} \sum_{j=1}^{n_i} x_{ij}$$

$$S_{x^2} = \sum_{i=1}^{N} \sum_{j=1}^{n_i} x_{ij}^2$$

combined results of $(\mu_0, \sigma_0 \ \& \ n_0) + (\mu_1, \sigma_1 \ \& \ n_1)$

$$C_n = S_n \qquad C_\mu = \frac{S_n}{S_x} ? \qquad C_\sigma = \sqrt{\frac{S_{x^2} - (S_x)^2}{S_n} \frac{1}{S_n - 1}}$$

Lets try an example



Now population n_0 & n_1 will be different due to the run time of each data collect and a caw being different this could bias the data towards the cluster with the highest population. To avoid this try setting n_0 & n_1 to 1 (unity).

$$S_n = \sum_{i=1}^{N} n_i = 2 \qquad S_x = \sum_{i=1}^{N} \sum_{j=1}^{n_i} x_{ij} = \sum_{i=1}^{N} \sum_{j=1}^{n_i} x_{ij}$$

$$S_x = \sum_{i=1}^{N} \mu_i \underbrace{n_i}_{1} = \sum_{i=1}^{N} \mu_i \qquad \sum_{j=1}^{n_i} x_{ji} = \mu_i \cdot n_i$$

$$S_{x^2} = \sigma_i^2 \underbrace{(n_i - 1)}_{0} + \frac{1}{\underbrace{n_i}_{1}} \sum_{j=1}^{n_i} (\mu_i \underbrace{n_i}_{1})^2 = \sum_{j=1}^{N} \mu_i$$

$$C_n = S_n \qquad C_\mu = \frac{S_n}{S_x} \qquad C_\sigma = \sqrt{\frac{S_{x^2} - (S_x)^2}{S_n} \frac{1}{S_n - 1}}$$

$$C_n = 2 \qquad C_\mu = \frac{2}{\sum_{i=1}^{N} \mu_i} \qquad C_\sigma = \sqrt{\frac{\sum_{j=1}^{N} \mu_i - \sum_{j=1}^{N} \mu_i^2}{2} \frac{1}{2-1}}$$

this is not super useful as does not depend on σ_i with $n_i = 1$

$$\sigma_{\bar{x}} = \frac{1}{\sqrt{n}} \sigma$$ ← population standard deviation

↑ sample size

↑ st dev of sample mean.

$$\sigma = \sqrt{n}\, \sigma_{\bar{x}} \quad \therefore \sigma^2 = n\, \sigma_{\bar{x}}^2$$

$$n = \frac{\sigma^2}{\sigma_{\bar{x}}^2}$$

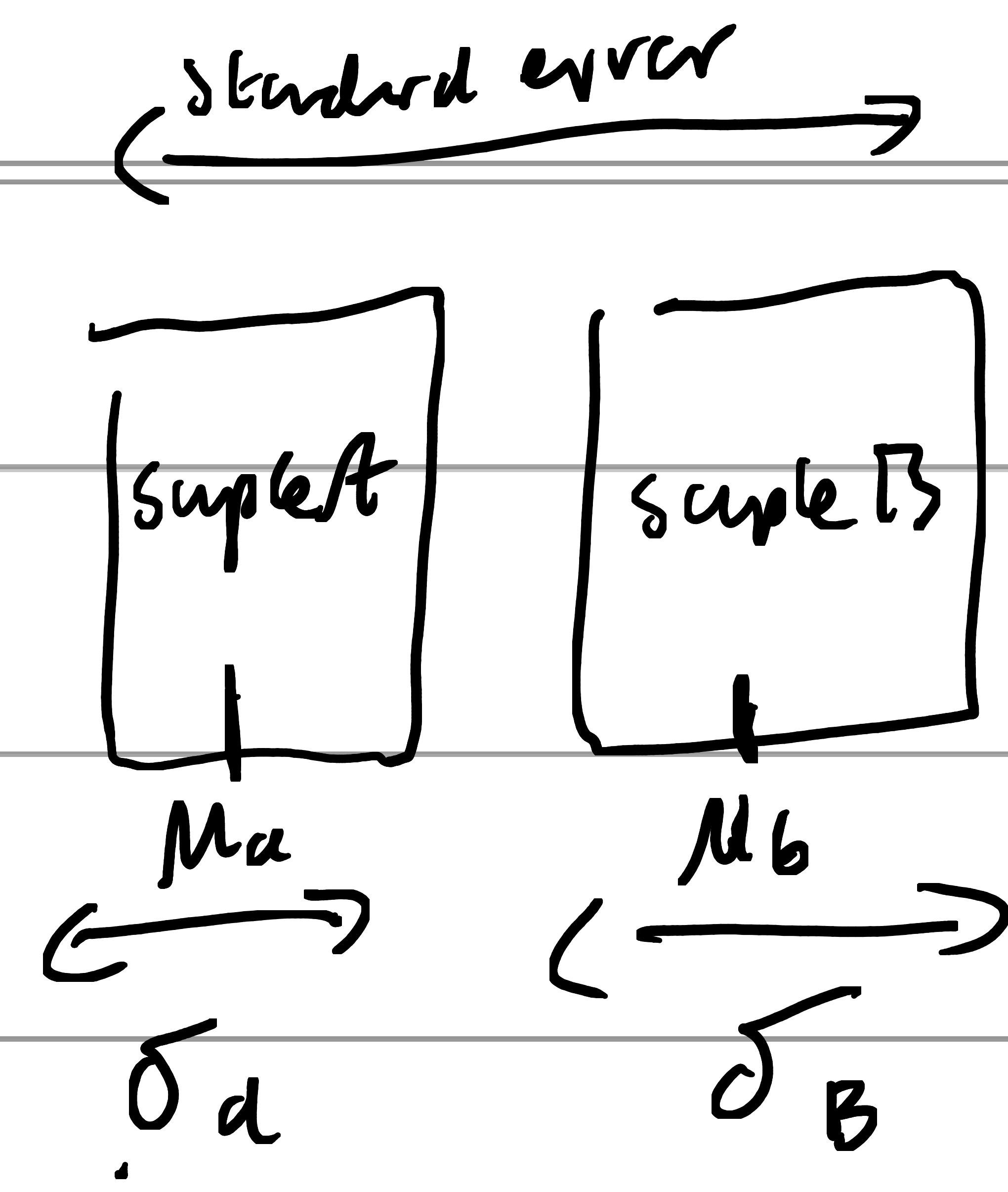
Again: $x = (x_0, x_1 \ldots x_n)$ $\quad n = m = 1$

$y = (y_0, y_1 \ldots y_m)$

$z = (x_0, x_1 \cdots x_n, y_0, y_1 \ldots y_m)$

$z =$

Again!



$$SE = \frac{\sigma}{n}$$

sample deviation

n cluster 2

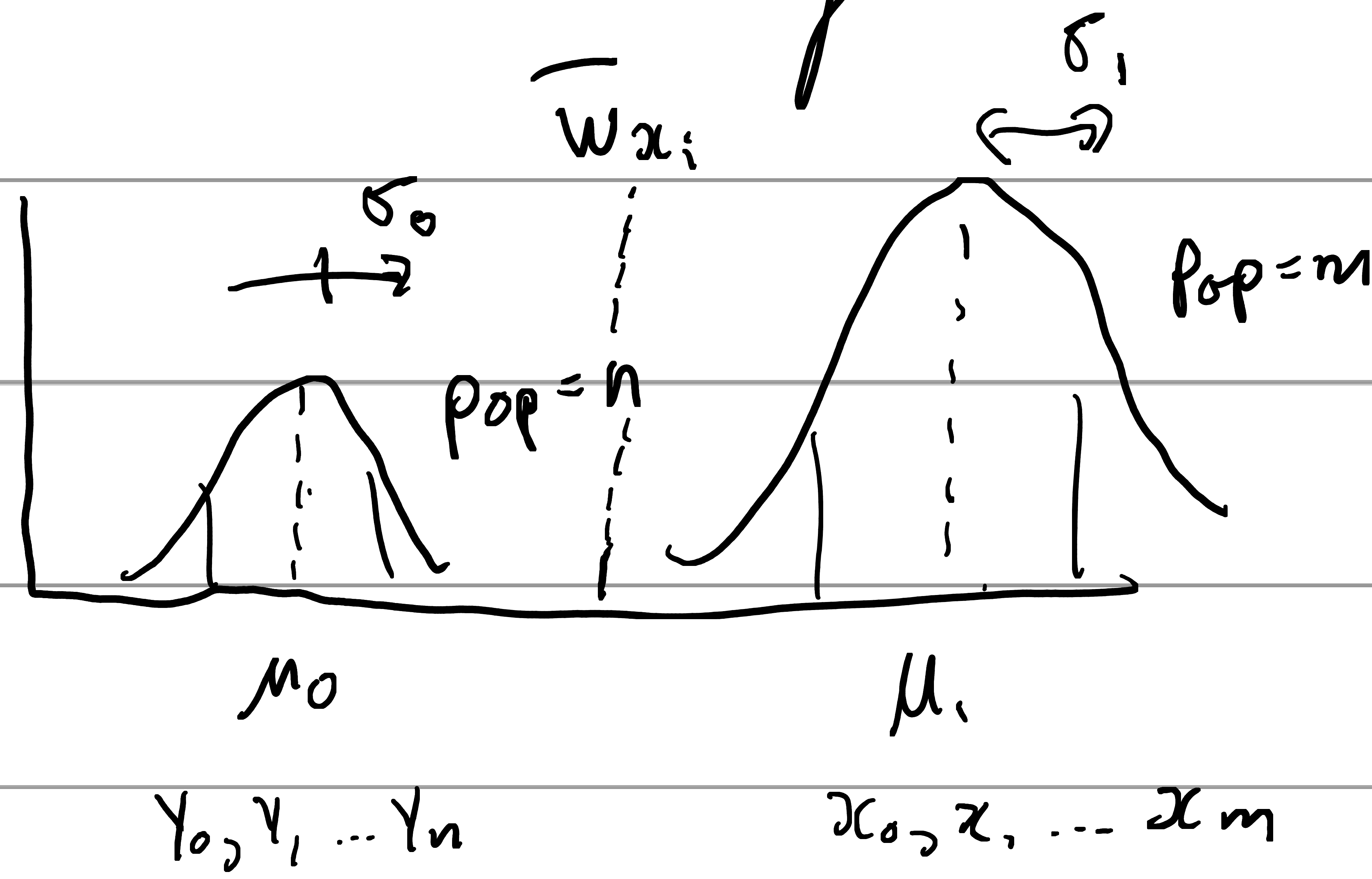
Simpler idea: use geometric circular mean to get μ of the combined sample, then calculate the error from pairwise of combine histogram of clusters.

Normalise hist

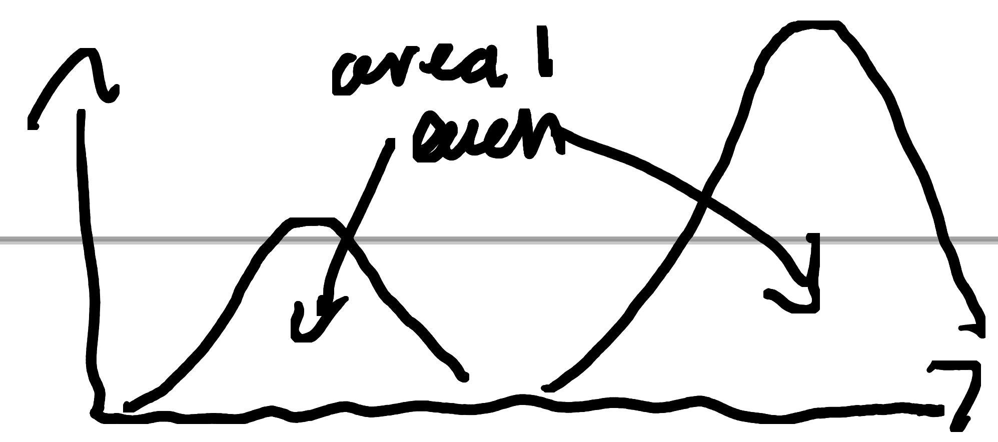
Steps will be from run cw & ccw combine channel data, re-cluster using kmedoids. Ignore the centroids, calculate geometric circular mean for each combined channel, from this geometric mean use pairwise distance. Perhaps normalising each count to a percentage the total cluster n will sum to 1, that way it will consider both cluster equally weight.

angles = [0 .... 16,383]

we have a combined histogram like



Need to normalise this



$$Y' = \frac{Y}{\sum_{i=1}^{n} Y_i} \quad \& \quad X' = \frac{x}{\sum_{i=1}^{m} X_i}$$

so that $\sum Y'_i = 1$ & $\sum X'_i = 1$ so that the population of each i 1 & clusters elements are treated as equals.

so to get the unbias mean we need to use a weighted mean:

$$\overline{W}_{x_i} = \frac{\sum_i^n w_i X_i}{\sum_i^n w_i}$$

$w_i X_i \rightarrow X'$ note no $\frac{1}{N}$ as it is included in X'

$\sum_i^n w_i \rightarrow 1$ due to normalisation

weighted mean of X_i

We now want to get statistics for the combined data for 2 sample sets

$$Z = \{X, Y\} \quad \text{but} \sum_i^{n+m} z_i = \sum_i^n x_i + \sum_i^m y_i = Z$$

lets go more general p sets of different samples which we wish to weight such that the mean will not not bias/preference any particular set with a higher population than the others.

p sets within Z

$$Z_j = \left\{ \frac{Y_0}{p\sum_i^n Y_{0i}}, \frac{Y_1}{p\sum_i^n Y_{1i}} \cdots \frac{Y_p}{p\sum_i^n Y_{pi}} \right\}$$

should sum to one

$$\sum_j^p Z_j = \frac{1}{p} \sum_j^p \sum_i^{n_j} \frac{Y_{ij}}{\sum_i^{n_j} Y_{ij}} = 1$$

normalised

hopefully p

$$\therefore \sum_j^p Z_j = 1$$

really? w_j should do this

m N weight from normalised hist

$$\bar{x}\ \mathrm{mean}\left(\frac{2\pi}{2^{14}} W_i \sin\left(Z_{ij}\right)\right) \qquad \bar{y}\ \mathrm{mean}\left(\frac{2\pi}{2^{14}} W_i \cos\left(Z_{ij}\right)\right)$$

$$\bar{x} = \frac{2\pi}{2^{14}} \frac{1}{n_j} \sum_j \frac{w_j \sin(Z_j)}{\sum_i^{n_i} w_j}$$

weighted mean

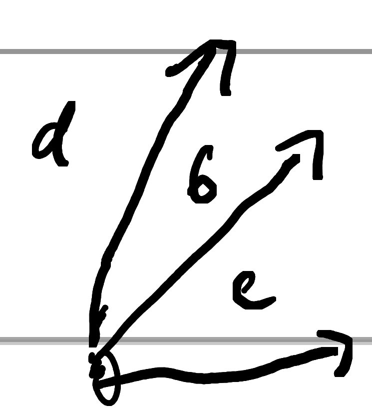
why this?

single combined Z series

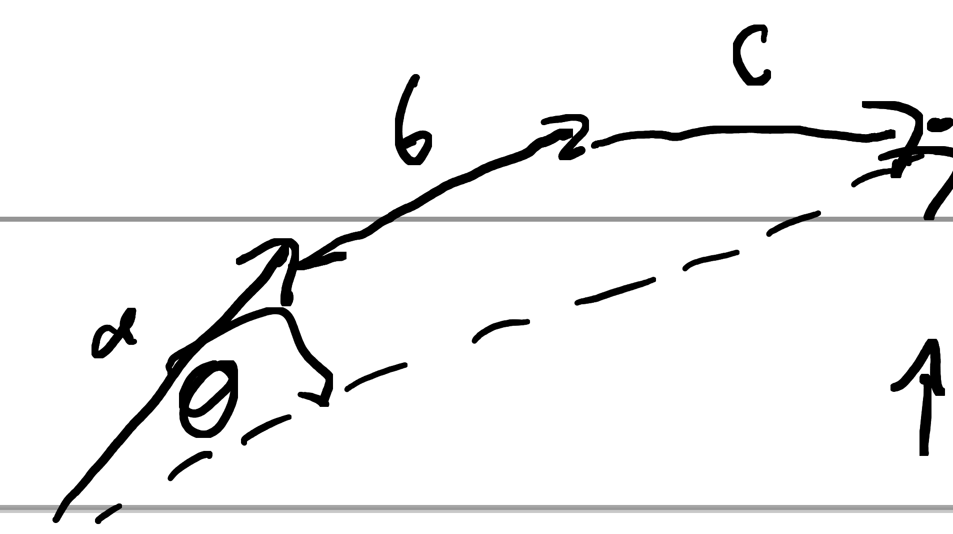
$$\arctan2(\bar{y}, \bar{x}) \cdot \frac{2^{14}}{2\pi} \% 2^{14} = \bar{Z}_{ij}$$

composed of P data set runs.

$\uparrow$ N total data points.



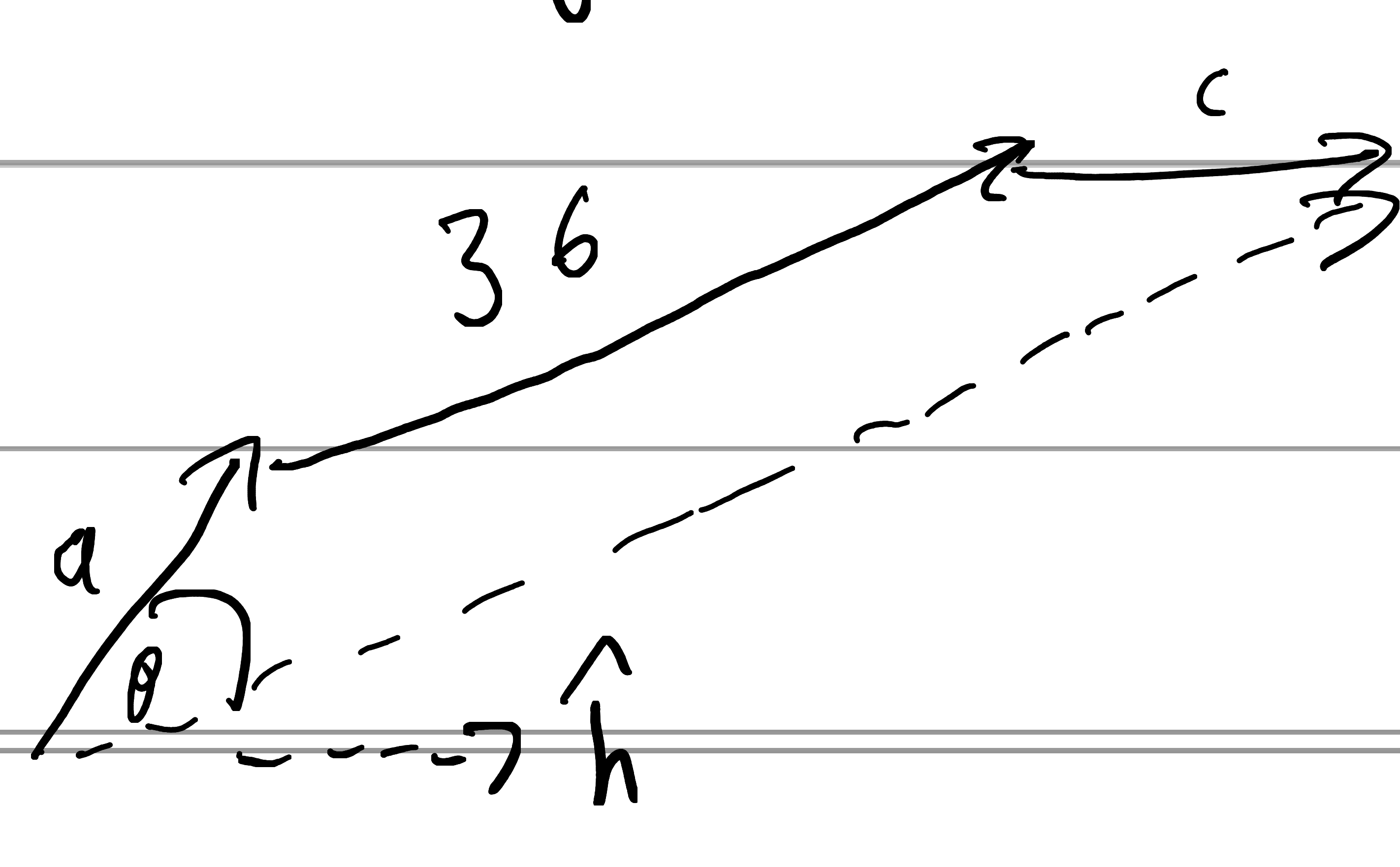
... resultant vector



a, b & c can scale the contribution to the circular mean in this direction

say that a & c have 1 count & b has 3 counts (un-normalised)

then stretching b will skew the mean towards its angle.



shallower

θ here is flatter wrt $\hat{h}$

And for the error:

observations

$$\sigma = \sqrt{\frac{\sum_j^N w_j (z_j - \bar{z})^2}{\left(\frac{M-1}{M}\right) \sum_j^N w_j}}$$

weighted mean.

number of not zero weights M

$$\bar{w} = \frac{\sum w_i x_i}{\sum w_i}$$

St dev (weighted)

$\frac{2\pi}{2^{14}}$ coeff to z_j

$$\bar{z}_j = \left(\arctan 2 \left(\frac{\sum_j^N w_j \cos(z_j)}{\sum_j^N w_j}, \frac{\sum_j^N w_j \sin(z_j)}{\sum_j^N w_j} \right) \cdot \frac{2^{14}}{2\pi} \right) \% \, 2^{14}$$

$$\sum_j^N w_j = 1$$

simplification if we normalise.

19/9/22

Thinking about steps need to find the cluster middle points.

Take multiple runs ZC- histogram (post outlier elimination.) prior to 2nd round of clustering.

File i need is called zero-crossing-detections.channels.inliers.json channel names like "zc_channel_og_data"

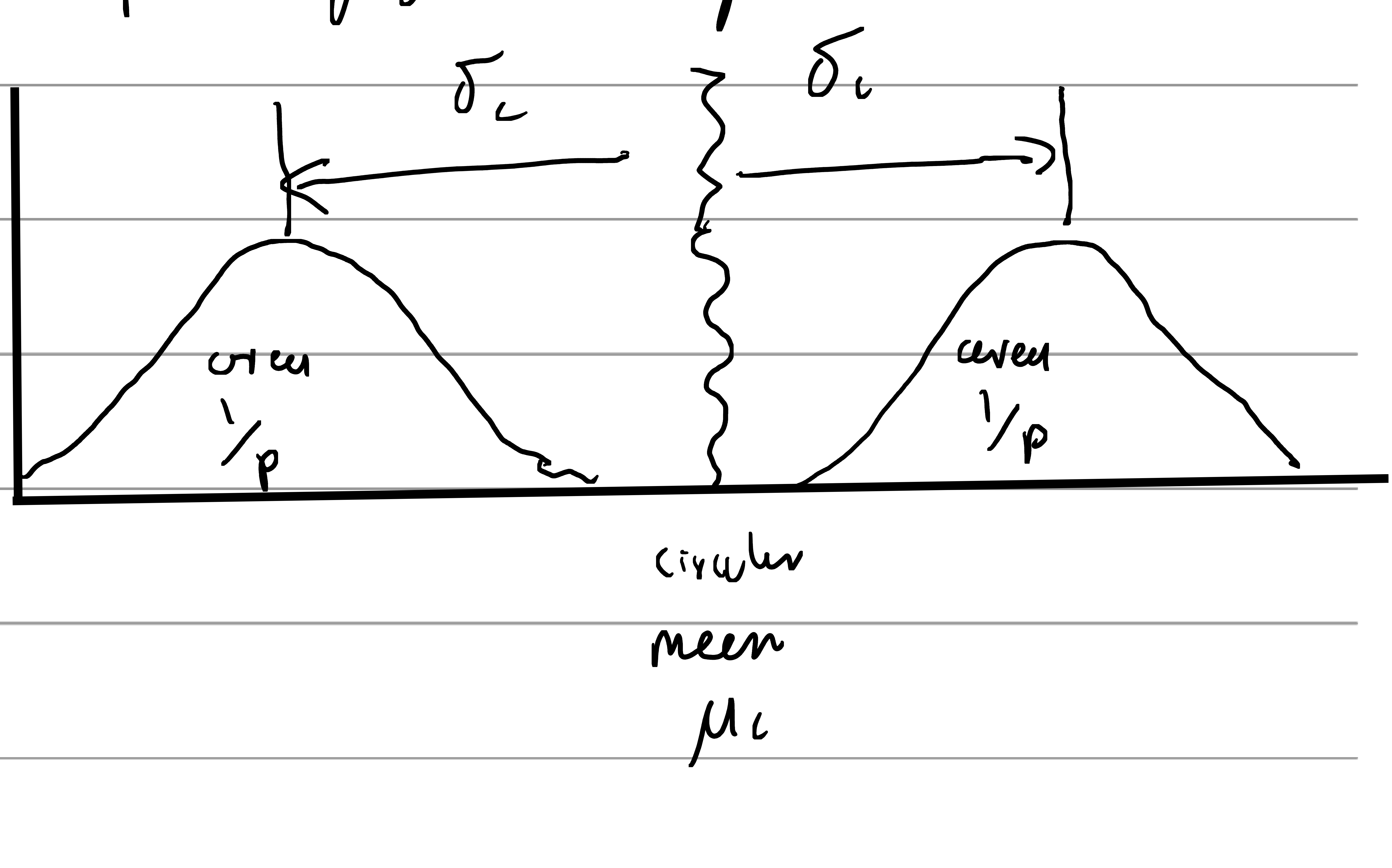
so for each run need to map channel data, then need to reconstruct the histogram of form kernel_x_rising/falling. per angle.

Next merge the runs as np arrays.

Then cluster with Kmedoids.

Then for each channel cluster normalise the histogram so that it sum to $\frac{1}{p}$, then combine, find circular mean & pairwise std-dev.

As each channel cluster should have p subset contributions and the sum of each subset should be $\frac{1}{p}$ then for each cluster we have a probability density, & therefore each histogram count per angle will estimate the probability for that angle.



maybe should assert $\sum X_{cluster_i} = 1$

20/9/22

working on merged clustering program.

→ make single histograms ✓
→ make merged histogram ~
→ reconstruct angles per cluster ✓
→ cluster merged channels ✓
→ create channel angle cluster identifier. ✓

→ iterate histograms & find channel cluster population. ✓

✓

→ Next need to normalise each histogram of counts such that a merged channel clusters counts sum to 1, thus represents a probability density. The sum of a channel would be $S_c = \frac{\text{num poles}}{2}$, the sum of all channels would be $S_a = \frac{\text{num poles}}{2} \times 6 = 3 \times \text{num poles}$ ✓

- → Next display histogram channels ✓
- → Display old stats mean + st dev ✓
- → calculate circular mean (weighted) sorted ✓ → got a dogy cluster black rising 2 ??
- → calculate pairwise st-dev ✓
- → Display density (cluster) ✓
- → Display new error + mean ✓
- → Recreate polarised spike train ✓
- → Generate binary spike train ✓
- → perform spacial periodicity analysis & plot. ✓

2/9/22

$\frac{2\pi}{360} \times \text{deg} = \text{rad}.$

-0.92

0.41

finished analysis for today

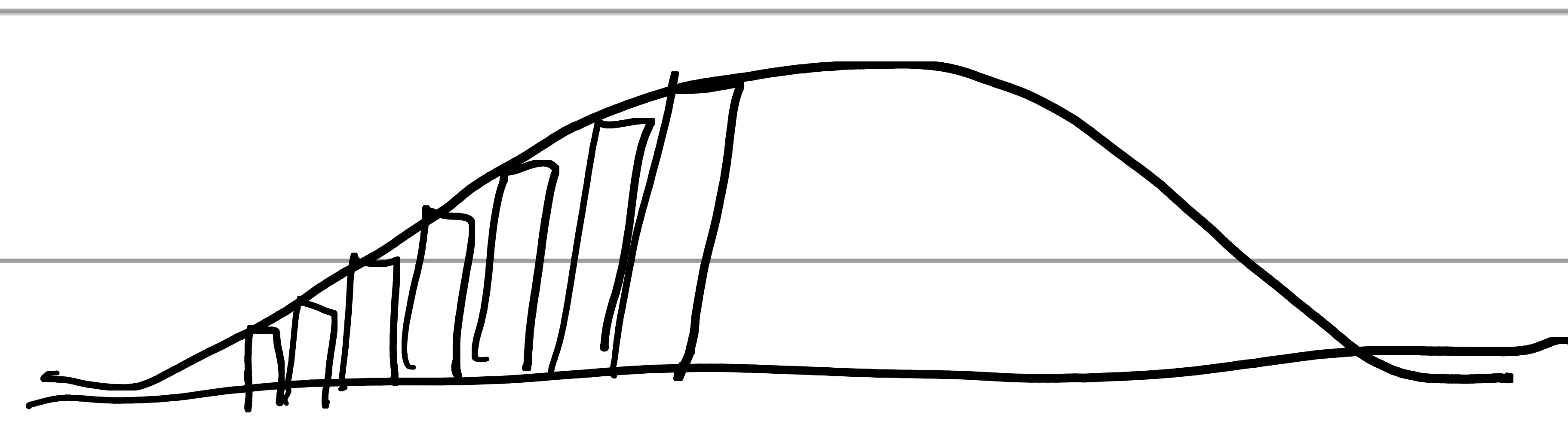
changes:

re-centralising fft plots

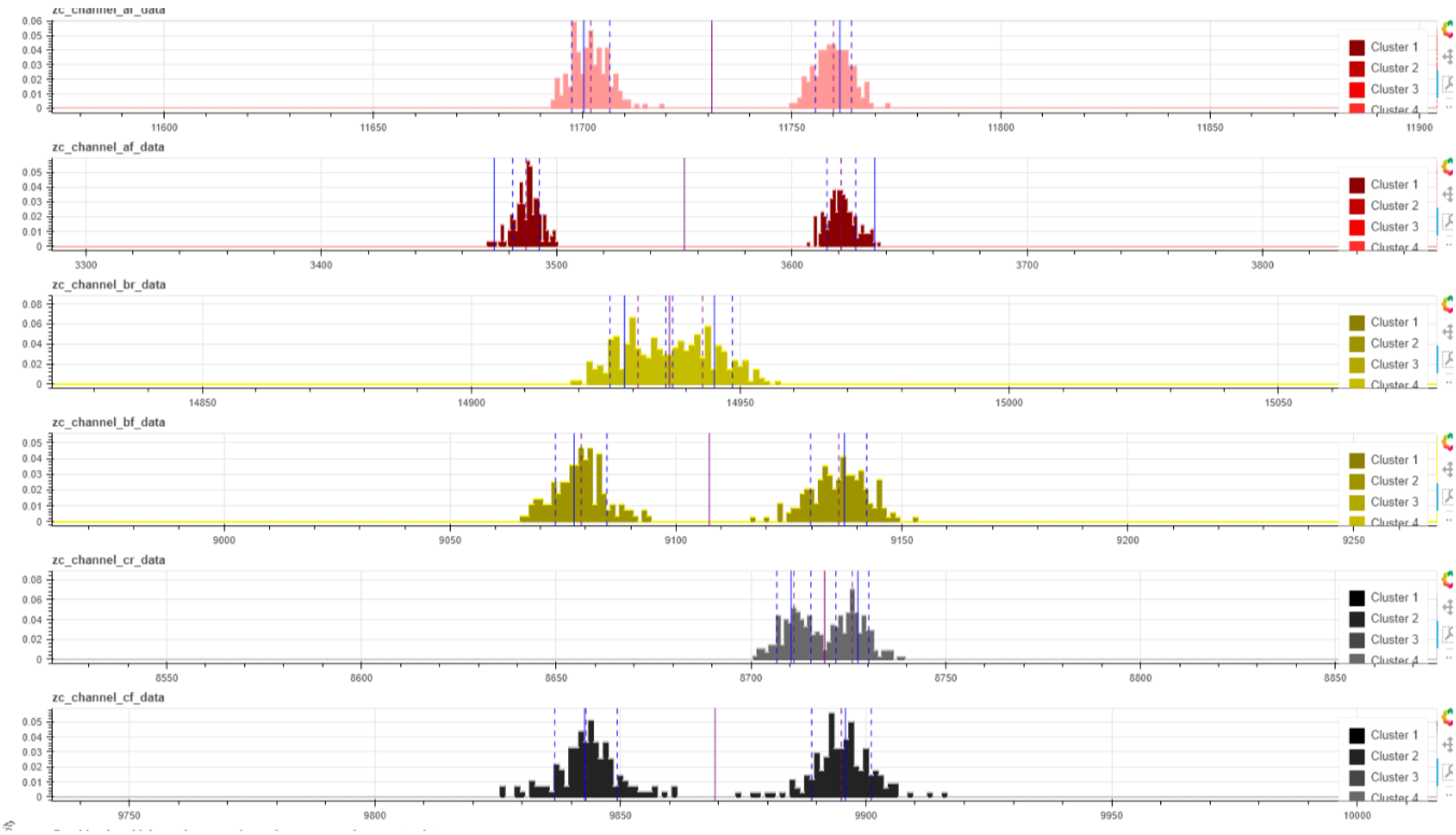
cluster density start from 1 & not 0 for consistency.

new centroids look good apart from black rising odd one (2nd) out... will re-run analysis & check.

Should take more data, check for node memory flag.

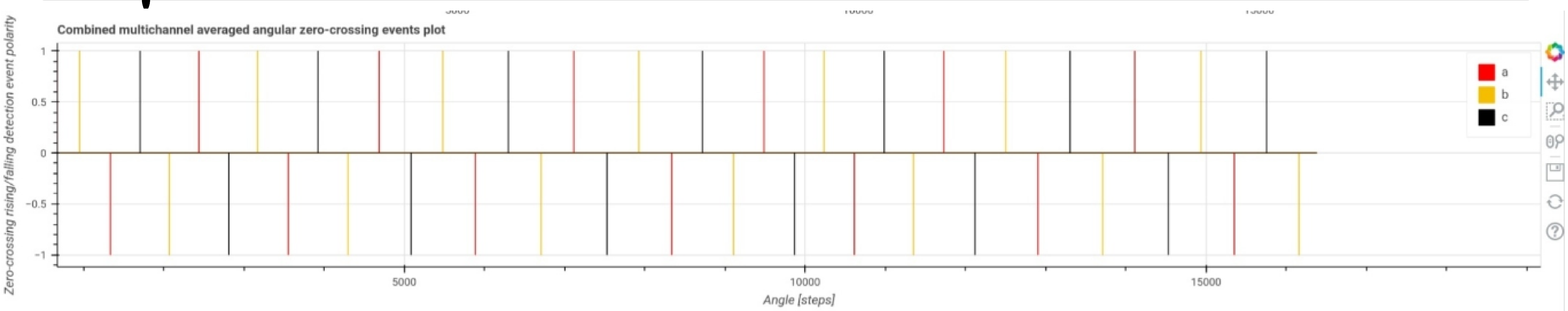


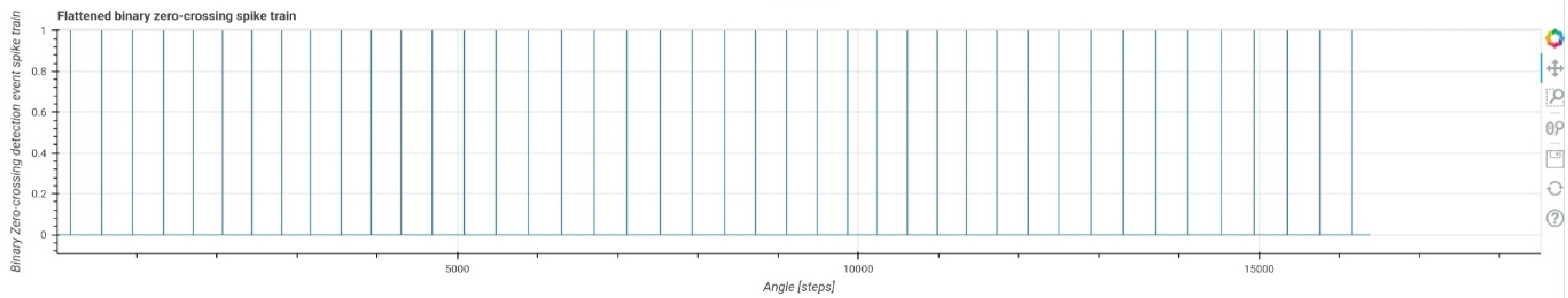
Results analysis 22/9/22



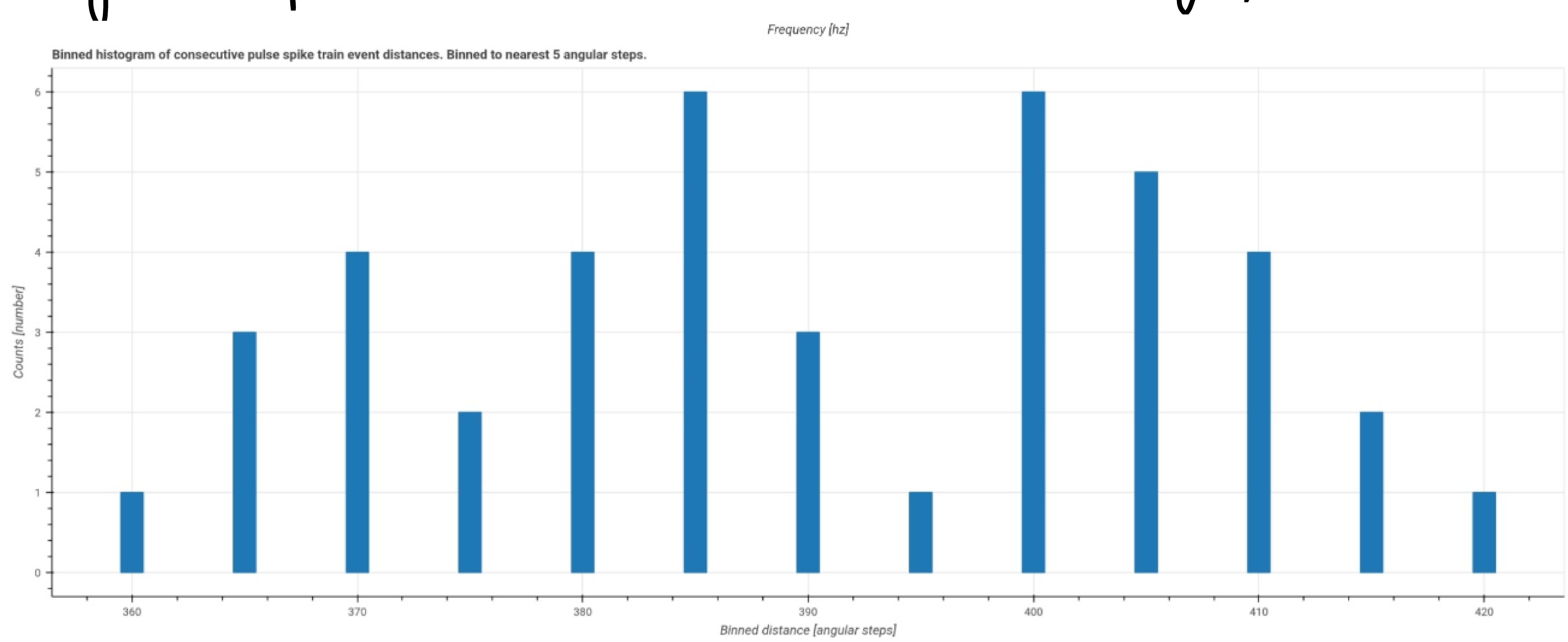
Above is a chart of the merging multiple runs 16 sept. ccw & 16 sept-4-cw, both runs of about 3->4 seconds. Notice the circular mean gets a good midpoint between the normalised datasets because of the applied weighting giving equal votes to each dataset regardless of the run time. Note the clusters approximate gaussians. We can see the systemic error here. Channel B rising (black) the gaussians overlap as there is low here, C rising is similar. All others have some seperation, with cw with long short & ccw with short long distances here they both overlap.

Analysis continued



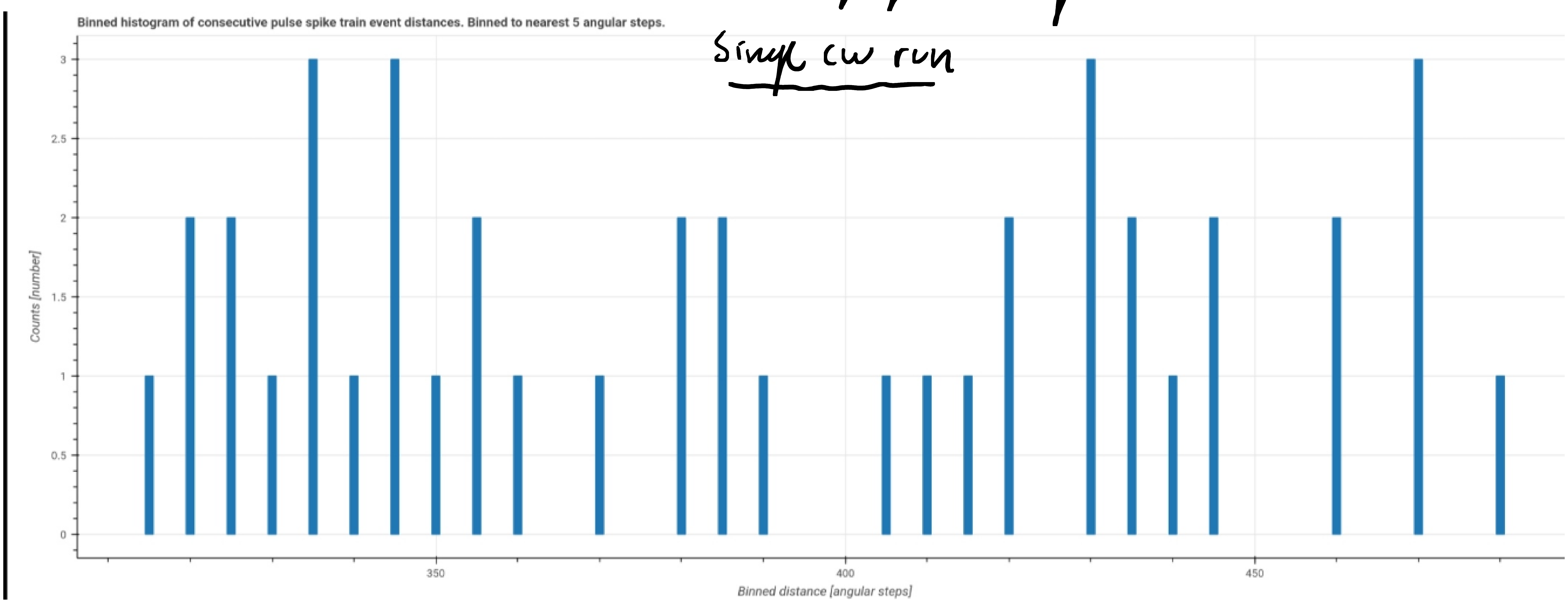


Here we see the result of the spike train by taking the circular mean between the two datasets self identified as a cluster as per Kmedoids clustering. Note there is no disernable short, long nor long, short pattern... the results seem much more evenly spaced.



We can confirm this by looking at the histogram of binned spike train distances. The distribution approximates a single gaussian, spacing is more uniform as compared to a single run... Next page

The histograms for the spike train distances for a single cw run uncombined was much less convincingly a single distance.



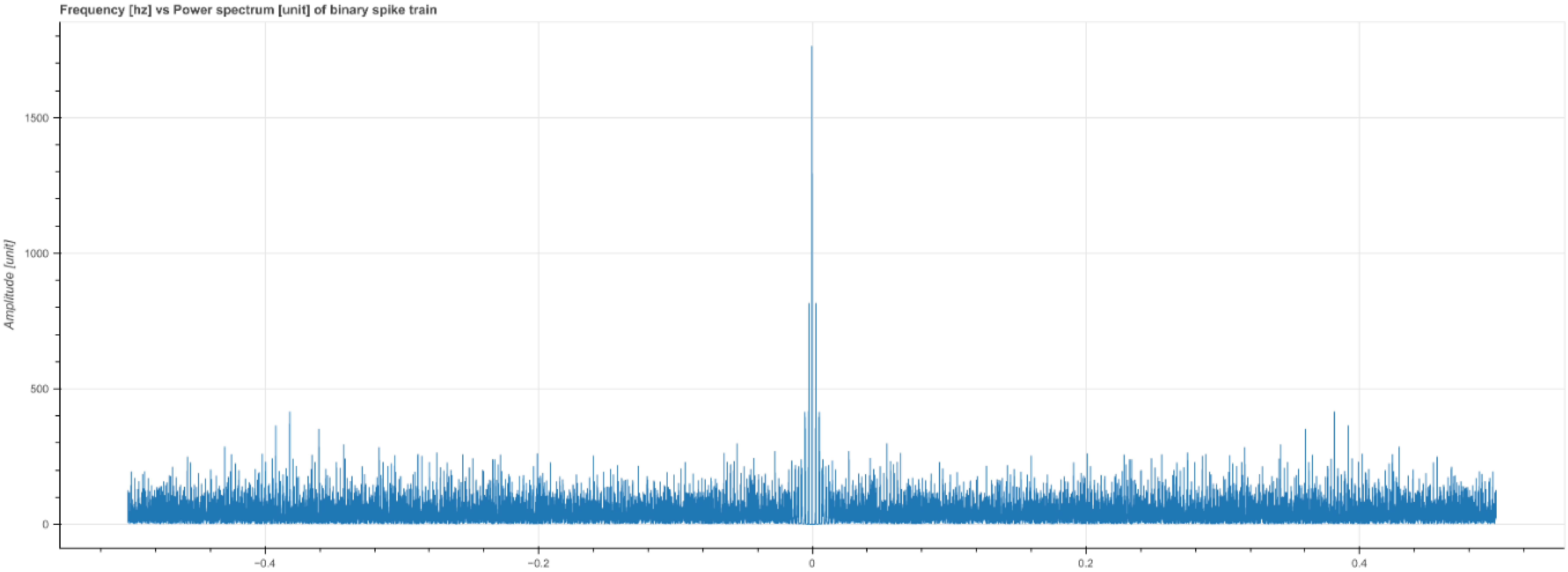
instead for the uncombined run there are more like 3 distances, long, short & somewhere in between.

Similarly we can compare the FFT's of the spike train data for a single direction run & the combined cw & ccw datasets.

FFT of combined dataset.

Temporal / spectral analysis of combined binary zero-crossing event spike train

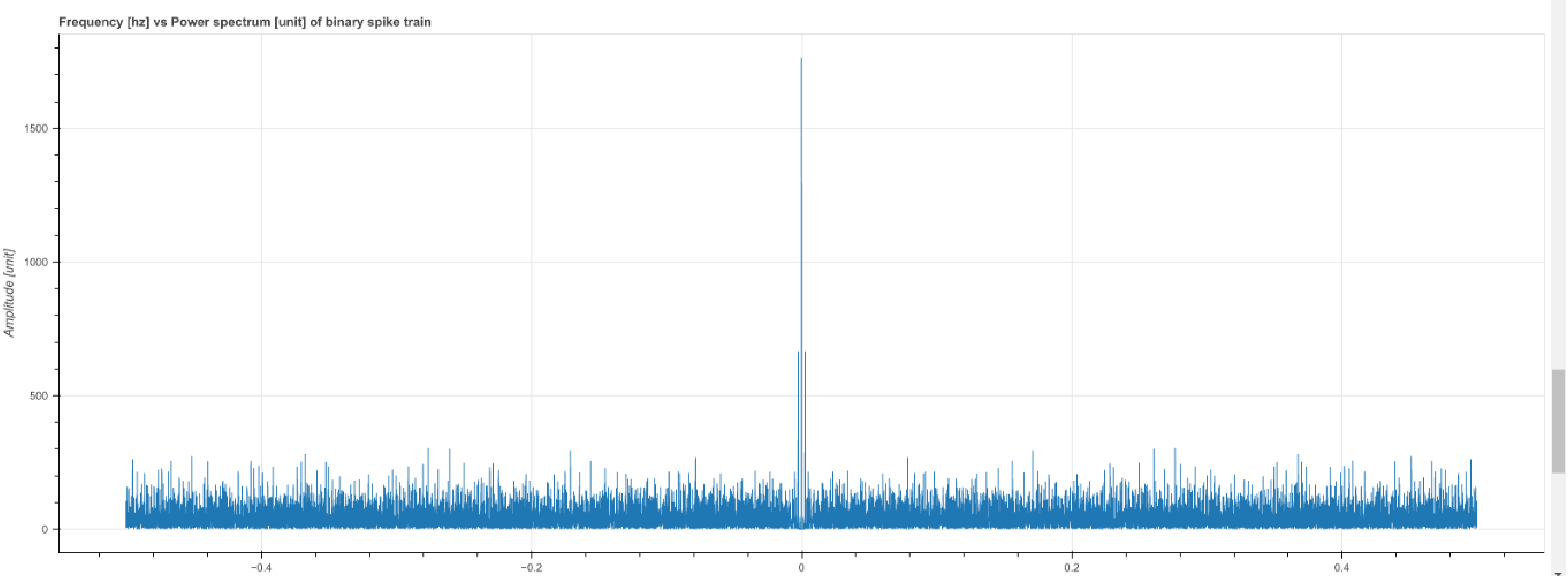
In order to determine the peroidicity of the spike train there are two methods, one generate an fft on the spike train and look for peaks in frequency which dominate, the second strategy is to measure the distance between each zero-crossing spike with the next spike in the tra
the distances can be rounded and binned into a historgram showing us the number occurance of spike distance of a certain binned value, if the motor is perfectly symmetrical it would be expected to see a dominate pulse delay time.



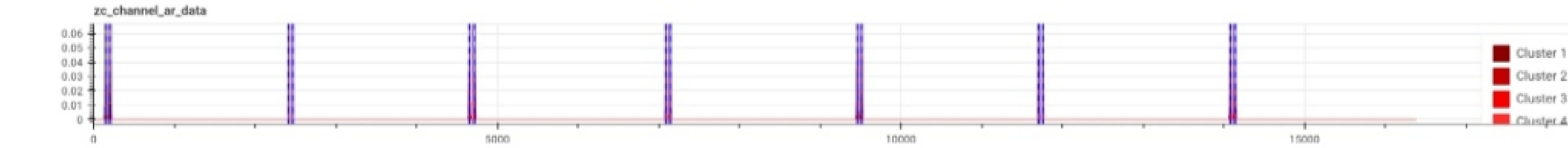
Although admittedly hard to see in the image above, the central region has some structure. The FFT of a uniform spike train is a spike train so we are seeing some signal here. Compare that the the FFT of a single run below:

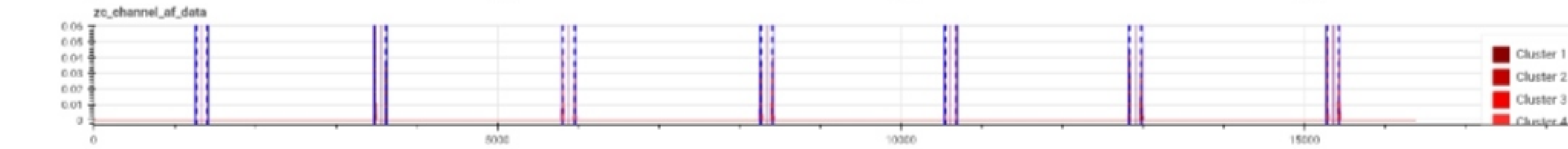
Temporal / spectral analysis of combined binary zero-crossing event spike train

In order to determine the peroidicity of the spike train there are two methods, one generate an fft on the spike train and look for peaks in frequency which dominate, the second strategy is to measure the distance between each zero-crossing spike with the next spike in the tra
the distances can be rounded and binned into a historgram showing us the number occurance of spike distance of a certain binned value, if the motor is perfectly symmetrical it would be expected to see a dominate pulse delay time.

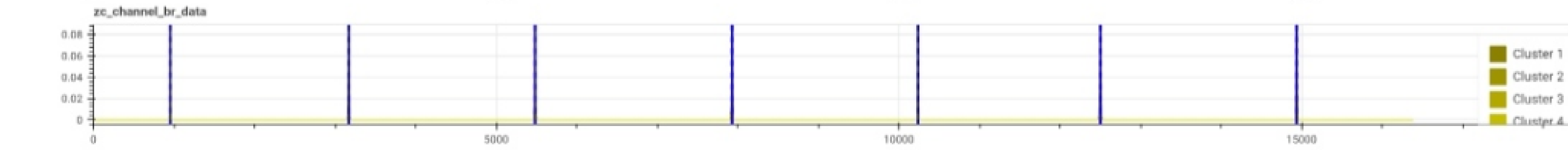


here there is only a little structure to the central region & there is not enough evidence of a spike train, this result is less periodic & clean than the combined.

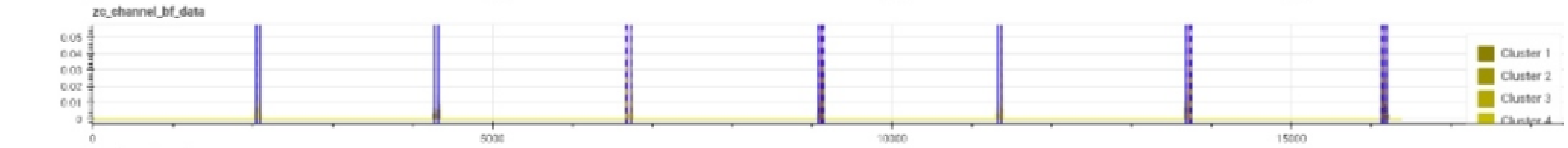


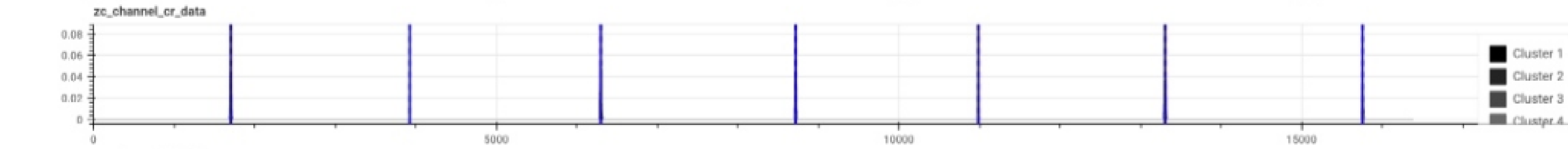


worse chanel

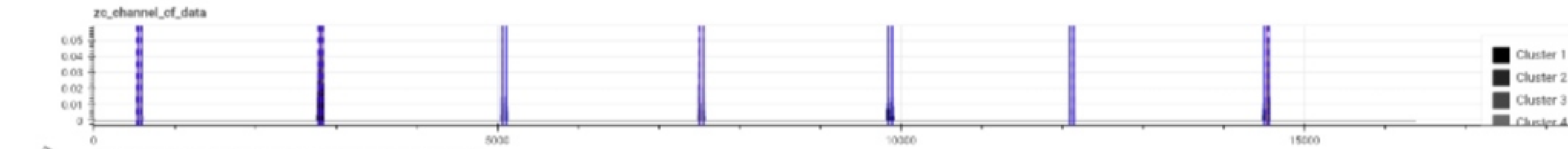


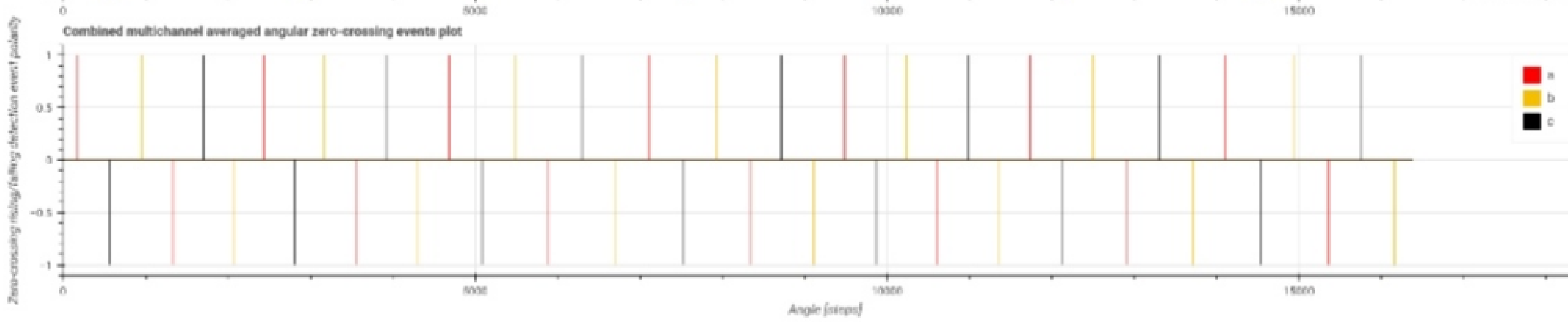
good chennel

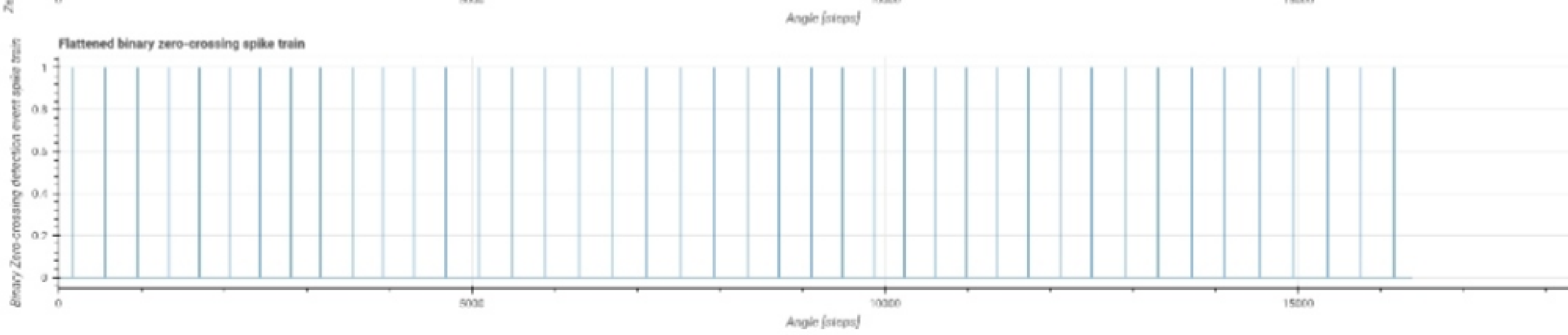




good chennel

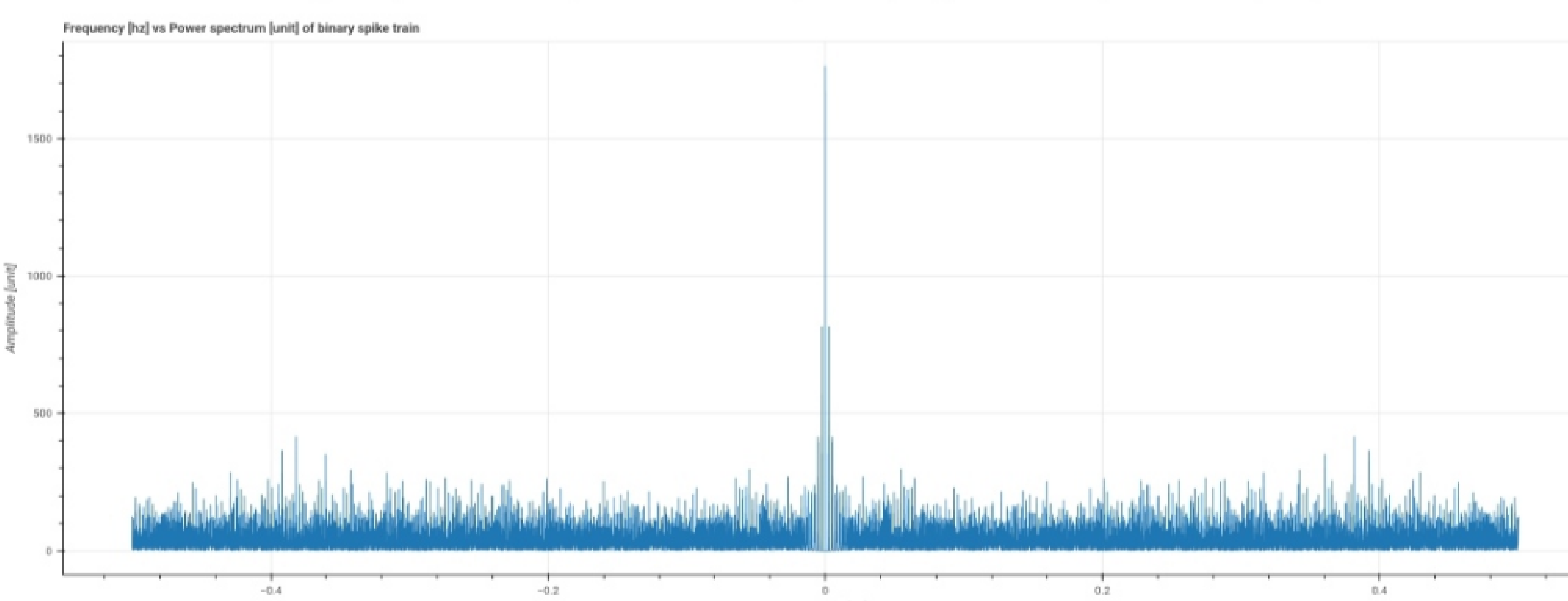


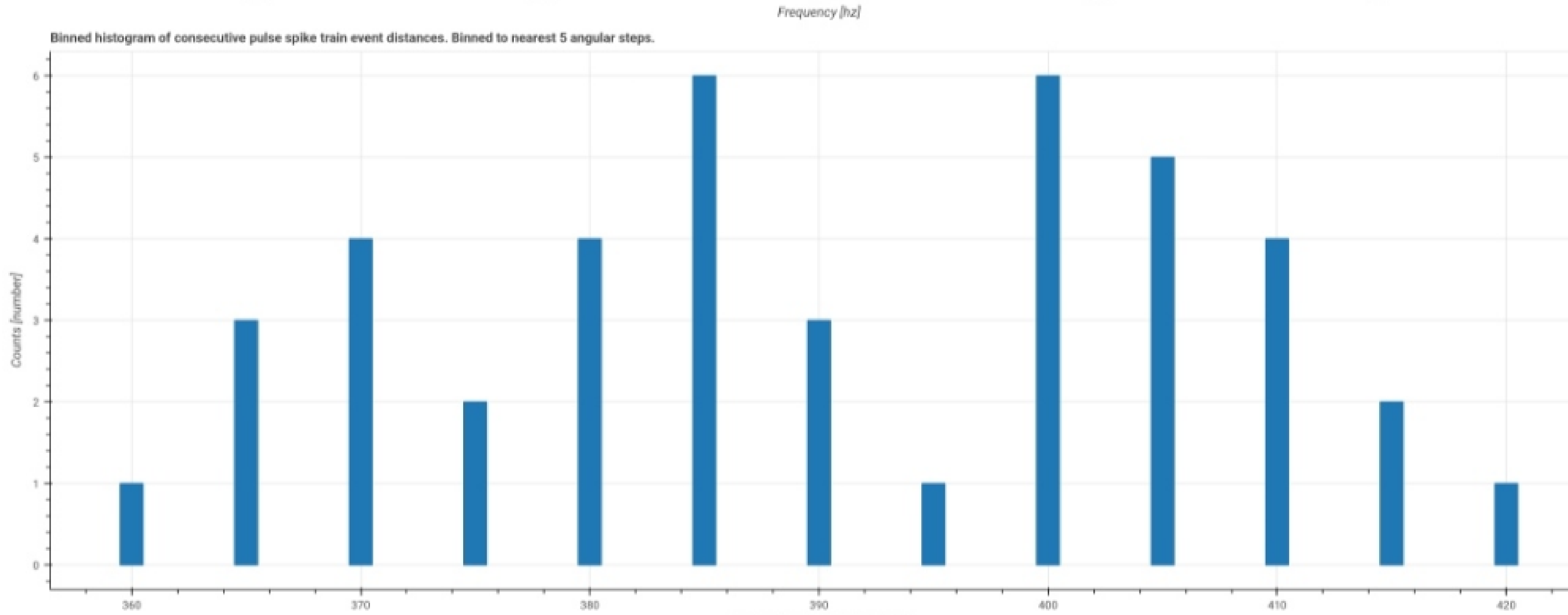


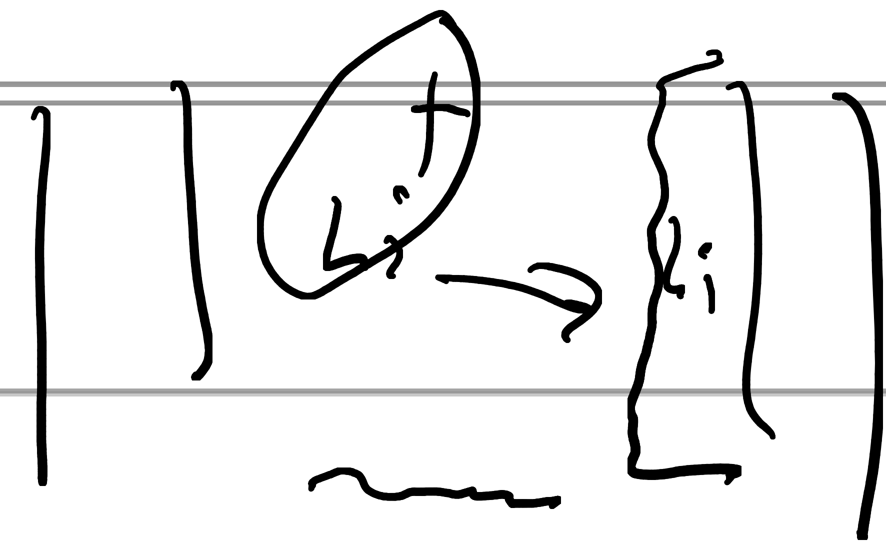


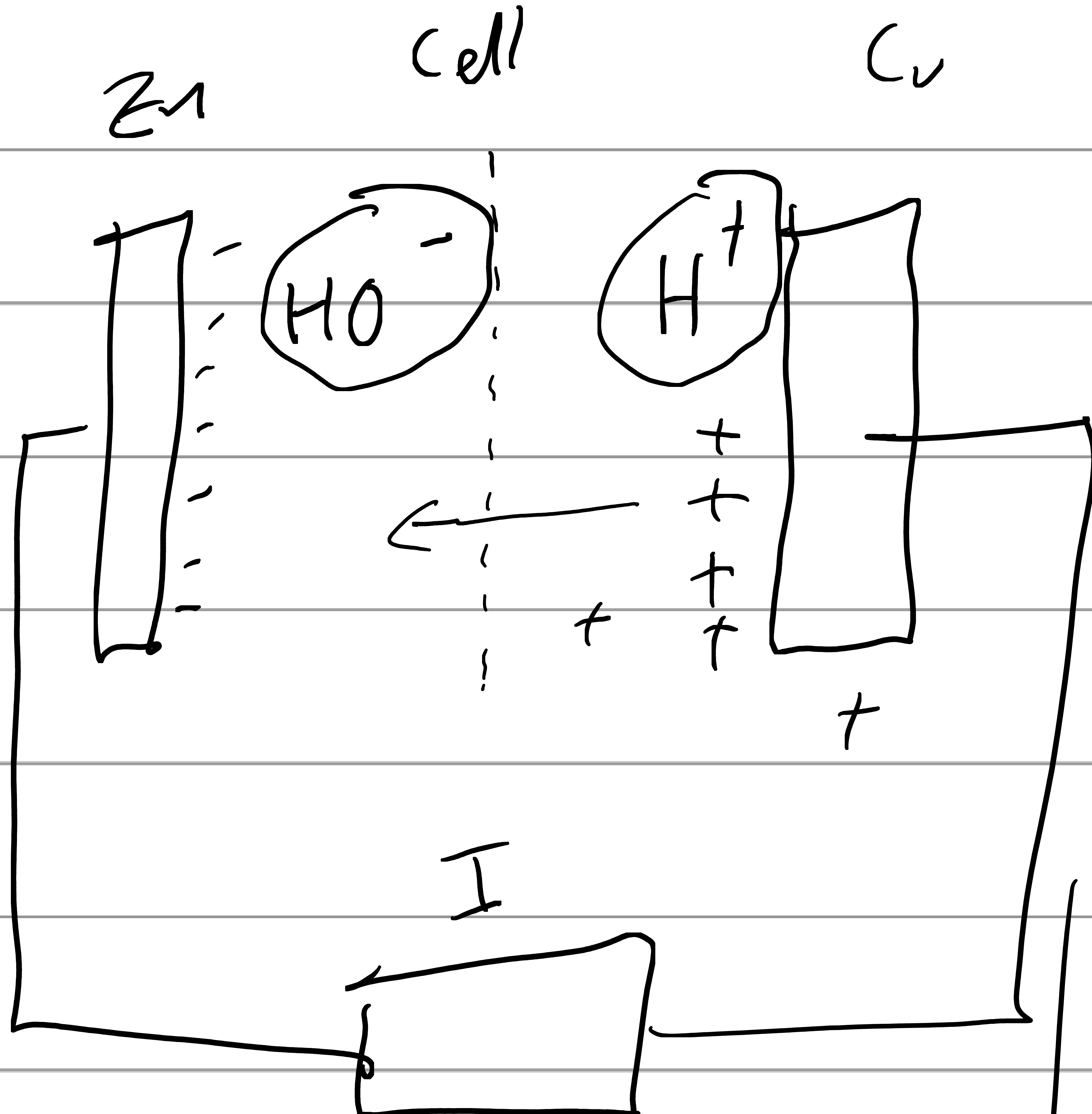
Temporal / spectral analysis of combined binary zero-crossing event spike train

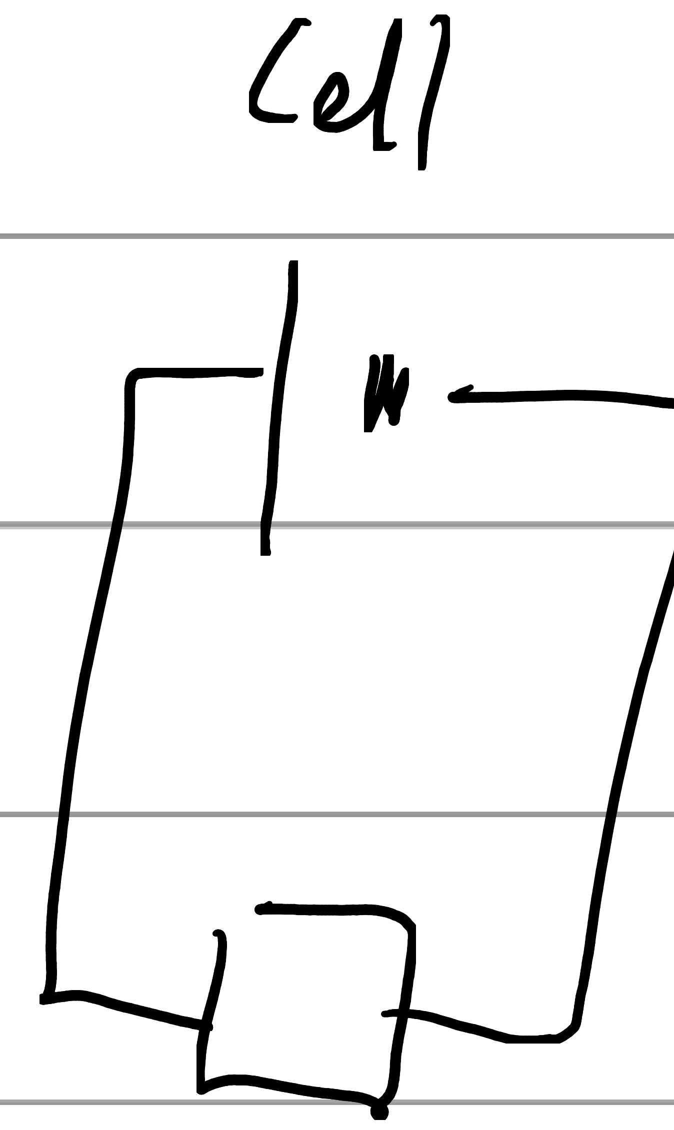
In order to determine the peroidicity of the spike train there are two methods, one generate an fft on the spike train and look for peaks in frequency which dominate, the second strategy is to measure the distance between each zero-crossing spike with the next spike in the the distances can be rounded and binned into a historgram showing us the number occurance of spike distance of a certain binned value, if the motor is perfectly symmetrical it would be expected to see a dominate pulse delay time.



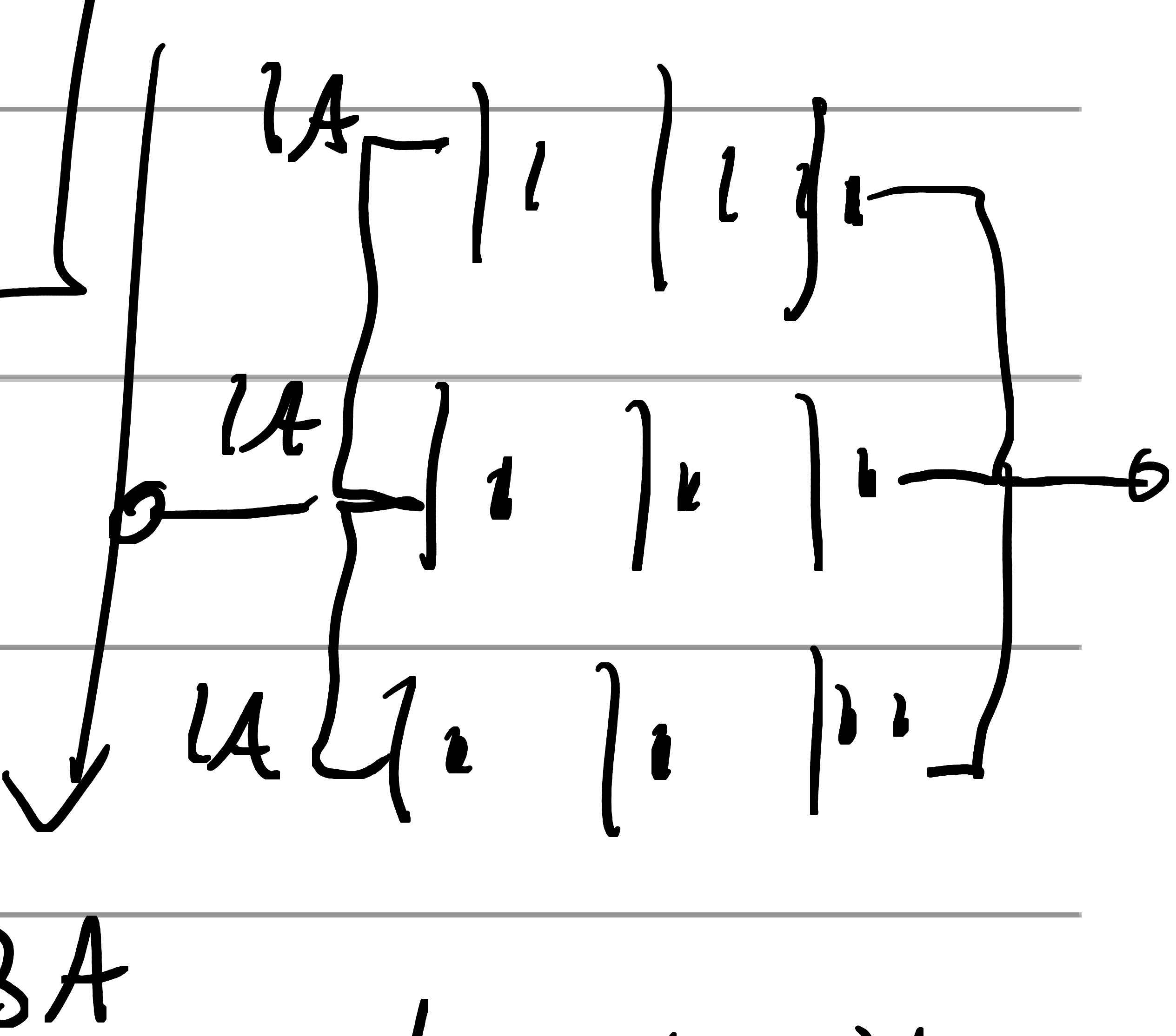


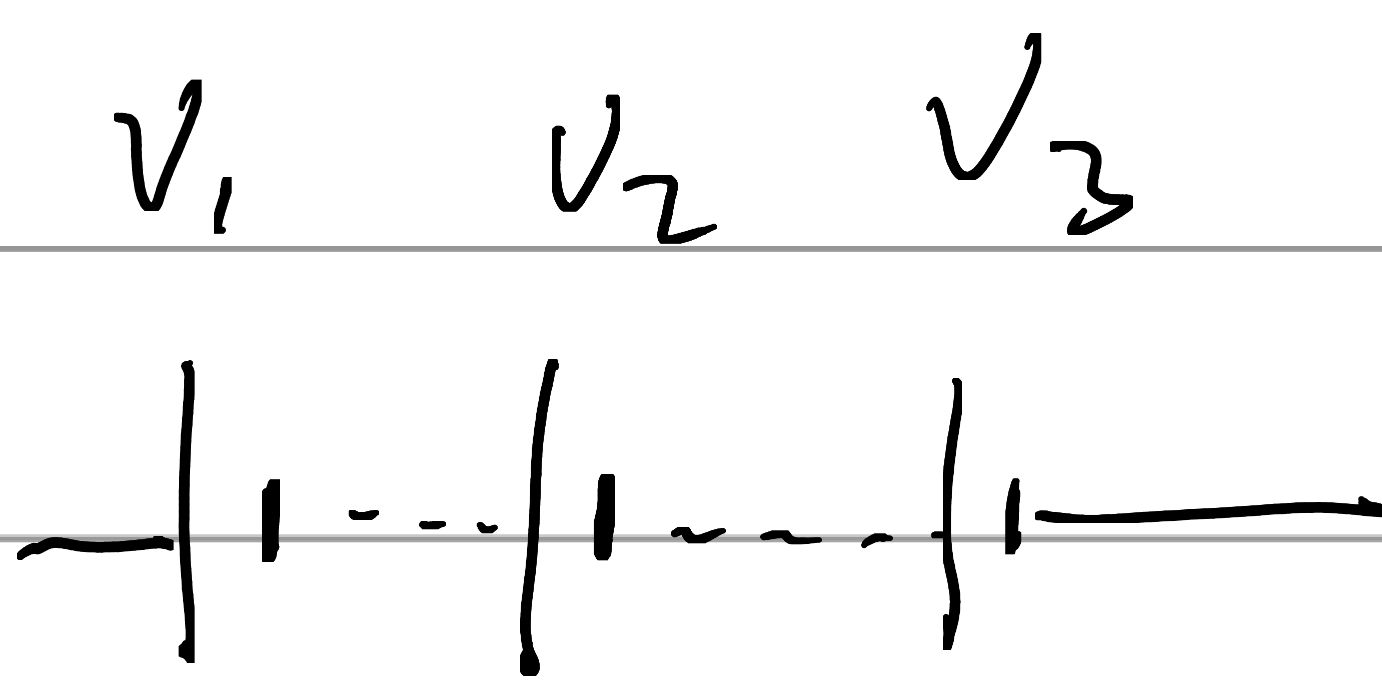




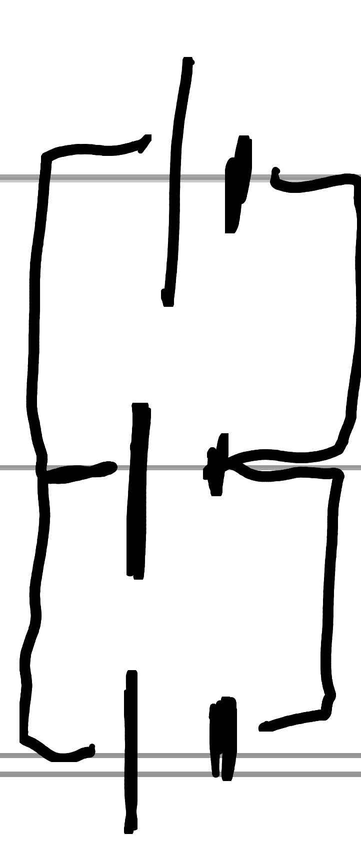


+3v





series $V_1 + V_2 + V_3$



parall $A_1 + A_2 + A_3$

Ah mAh $\frac{1}{1000}$