

12/9/22

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

combine ✓

inspect ✓ voltage robotics

smooth ✓ (check libml)

detect ✓

cluster ✓

analyse ✓ z c cluster cleaners

inspect - zc ✓

cluster - intvers ✓

analyse - intvers ✓

inspect intvers ✓

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 200mb of network bandwidth.

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 address.
cloud-drk ✓

14/9/22

Adding more explanation/concise on the graphs.

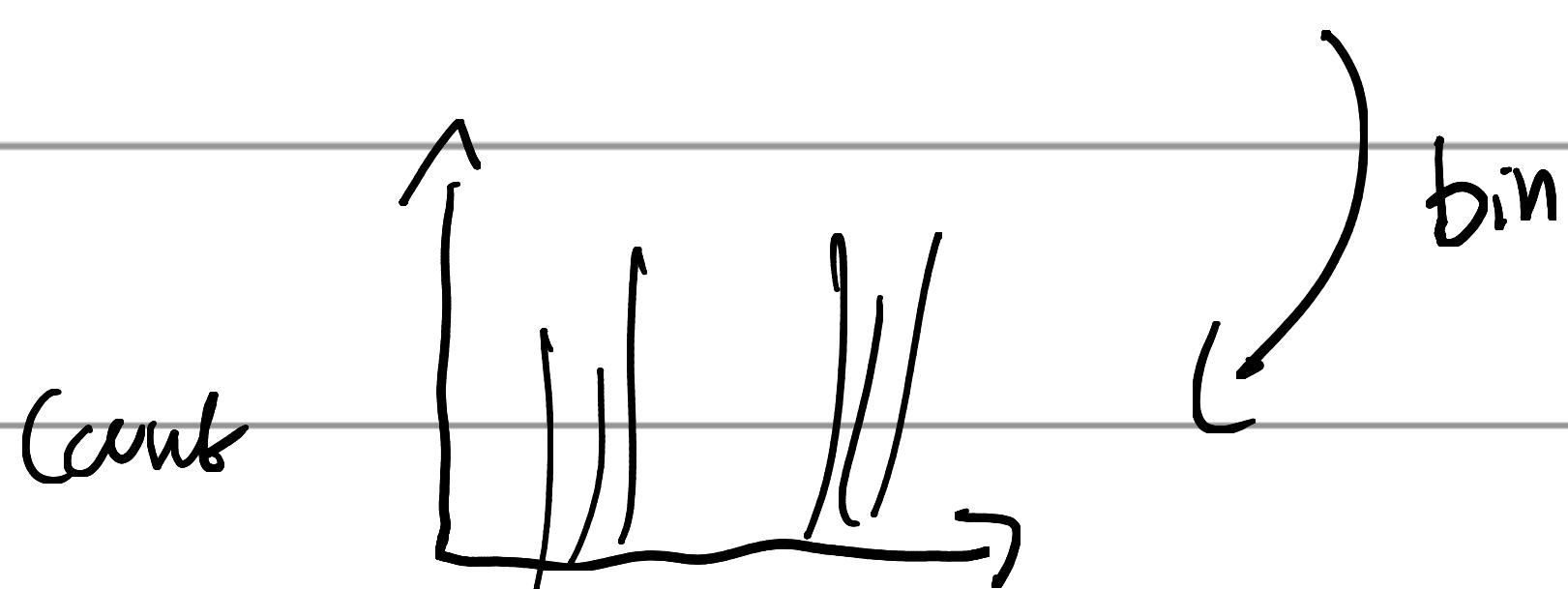
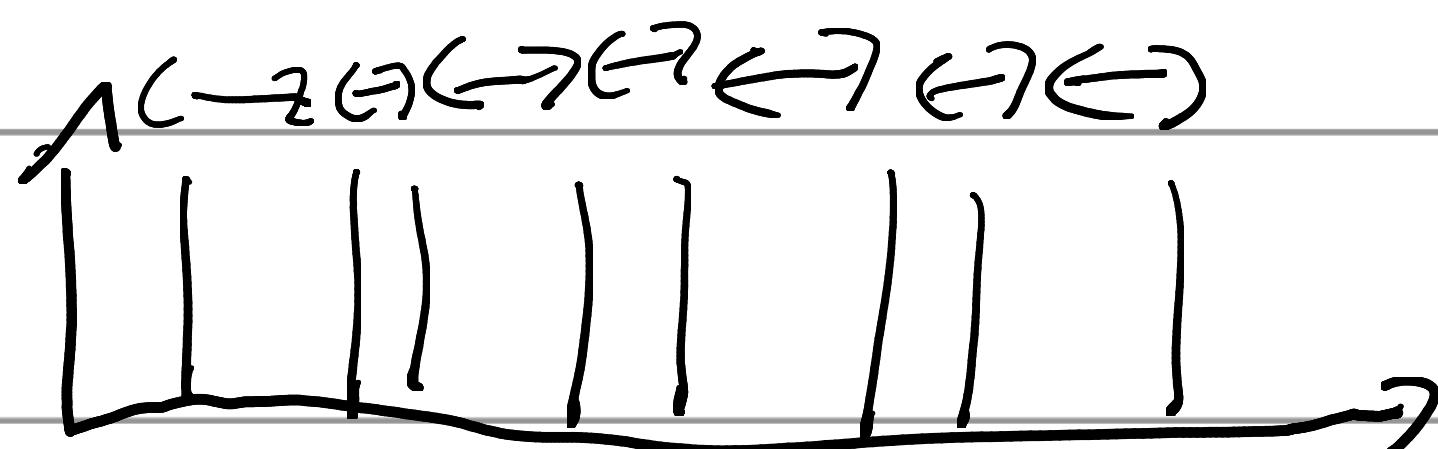
Binning binary pulse spike train in order to quantify periodicity.

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)



Consecutive pulse key time.

Need a lib folder, subdivide graph report into classes.

15/9/22

Thinking about

do chores for rising vs falling. phase A, B, C

rising vs falling.

Adding more chunks to smooth

Made a class for report generation.

Need to fix smooth plots \sim vN plot on to -50 on q axis

Add text description to the original cluster

plot.

On the density plot +1 to each identifier.

Need to service Nebraska w/br capture to check for

older existences for a run. \sim Need to test?

copy report card

Need to rename temporal to special

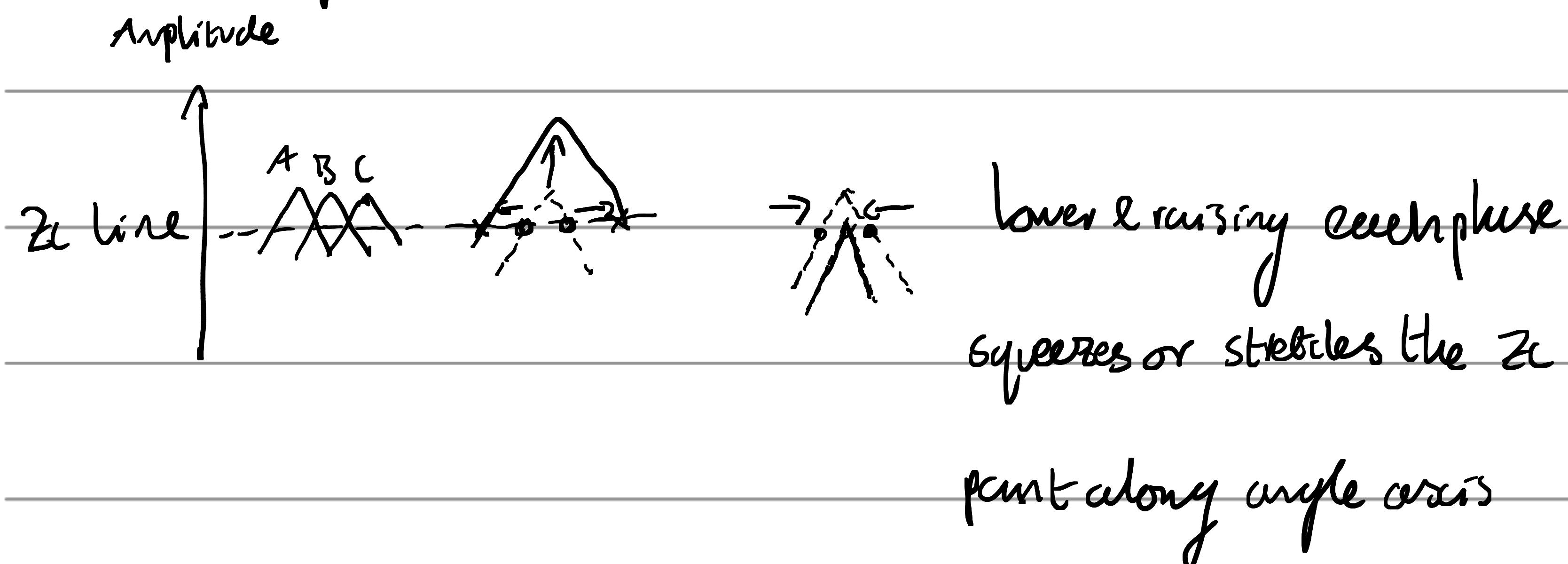
Code is now producing nice reports, looking at the Sept 2 dataset,

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 (consecutive) is shorter than falling to rising edges which appear on average longer.

1619122

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



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 pattern check, distures past the first 2c

99.8% match after angle step zero:

~3 sizes of spacing S, L, S, L

Attempting cw run:

Had some issues with bad runs low match rate, need to employ 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 pattern check, distances past the first 2c

99.89% match after angle step zero:

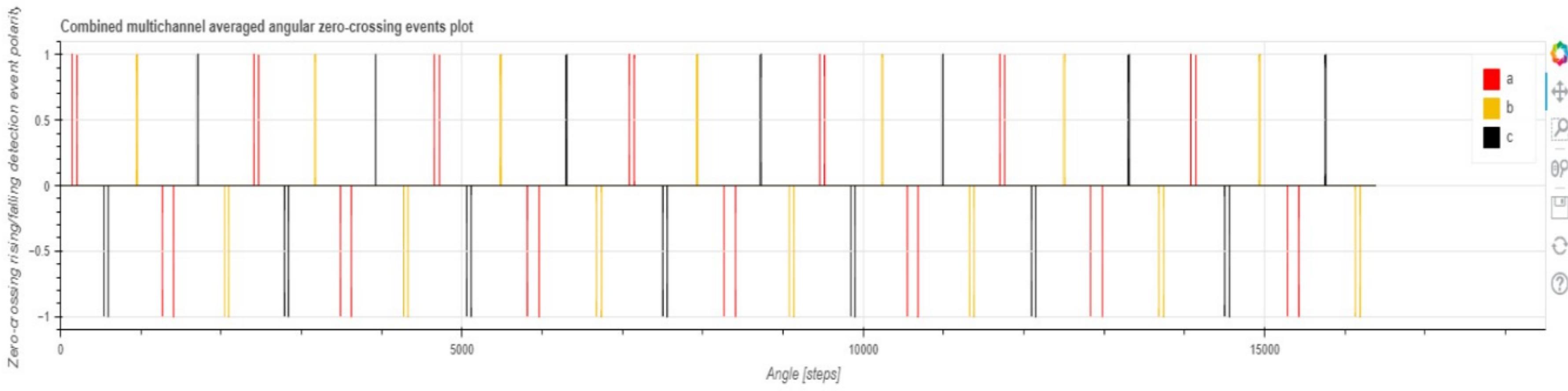
~ 3 clusters of spacing L, S, L, S, L, S

Definitely seeing some reversal of the pattern, now have 2 runs with groups of great match rate. There certainly appear to be different "pulse trainings", 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

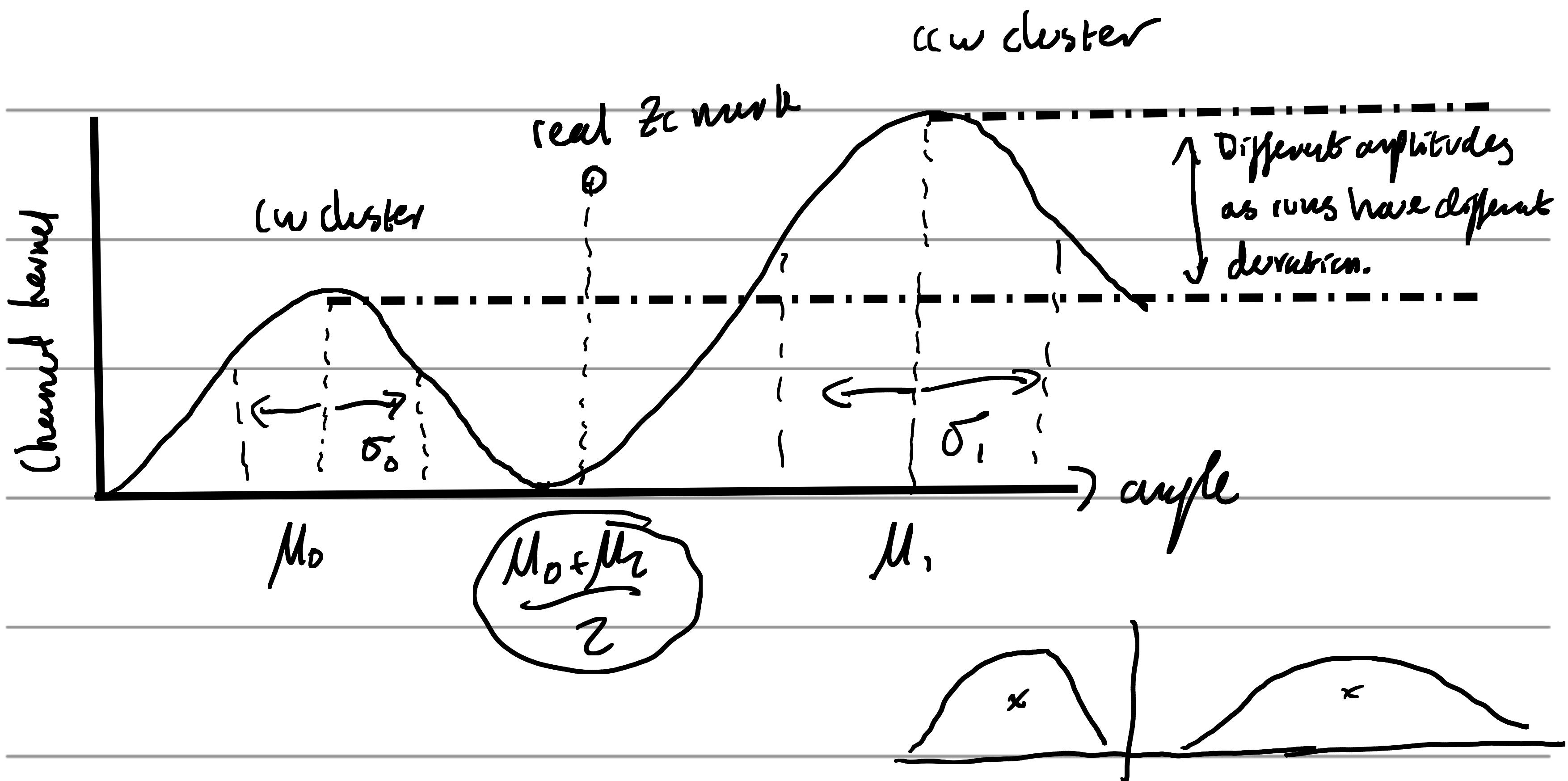
Post clustering, the mean of each channels zero-crossing channel detections is known. Therefore we can reduce the data so that there is only a single definitive pulse (rising or falling) for every cluster within each zero-crossing channel. The combined zero-crossing events are combined into a single 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 combine these results to reveal the true centers.

Consider merging the Z_C 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 CW & CCW clusters separately 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 M_0 & M_1 as given with their σ_0 & σ_1 errors & combine them.

\sim mean

Perhaps the best way to find the mid point between θ_1 & θ_2 would be the circular mean, as it respects midpoints when passing the $180^\circ - 20^\circ$ 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 & this gives a good geometric mean.

e.g. hour of day = $[0 \dots 24]$

hours sample = Random ($[0 \dots 24]$, 4)

convert hour samples into radians.

$$\arctan 2(\bar{y}, \bar{x}) \cdot \frac{2^4}{2\pi} \% 2^4$$

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

$$\bar{x} \left(\underbrace{\frac{2\pi}{2^4} w_i \sin(y_i)}_{\text{mean}} \right)$$

$$\bar{y} \left(\underbrace{\frac{2\pi}{2^4} w_i \cos(y_i)}_{\text{mean}} \right)$$

$$\text{hours-sample-}x = \sin(n) \text{ for } n \text{ in hours-sample}$$

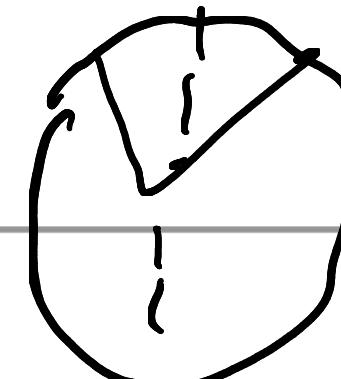
$$\text{hours-sample-}y = \cos(n) \text{ for } n \text{ in hours-sample}$$

weighted

average \bar{x}, \bar{y}

Version?

11 12,



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

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

$$\text{hours-avg} = \left(\arctan 2(\bar{y}, \bar{x}) \cdot \frac{2^4}{2\pi} \right) \% 2^4$$

We can have $M_0 \rightarrow M_1$, know $M_0 \approx 16,394$

So the errors for M_0 & M_1 are σ_0 & σ_1 , respectively.

Circular mean does deviate from arithmetic the under

the distribution of cycles but is approximately $\frac{1}{N} \sum_{i=1}^N x_i$,

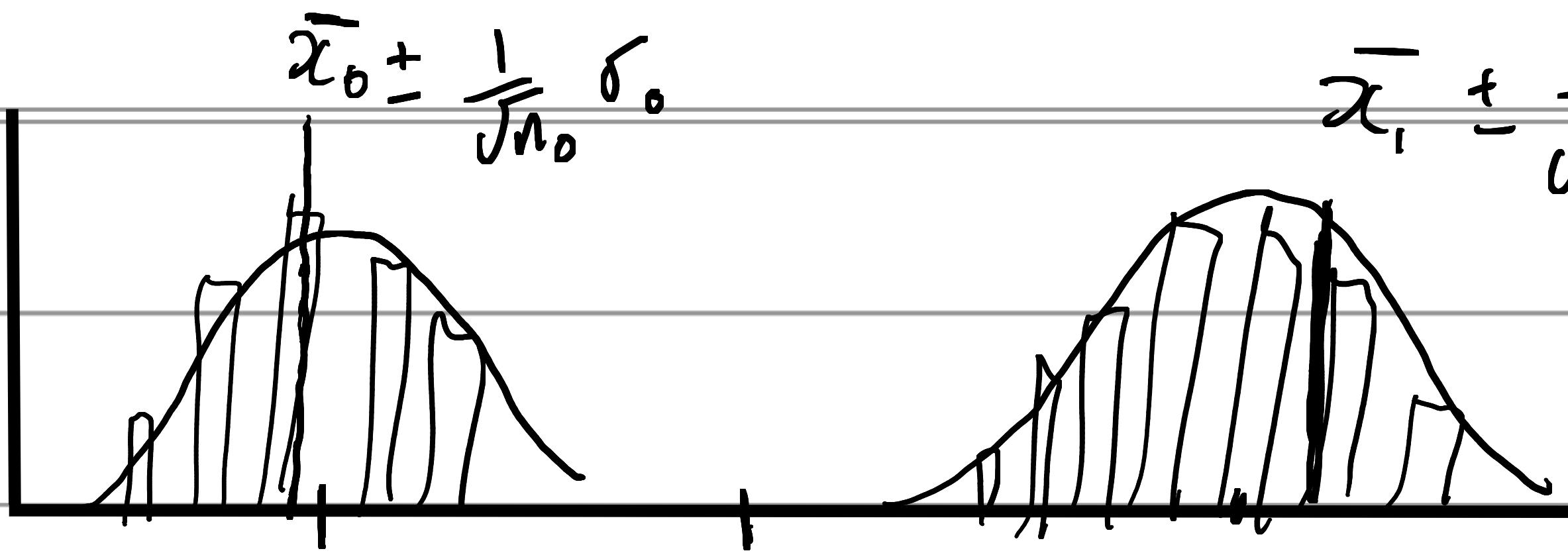
this is the form of simple addition of each cycle count, thus

$$M_{\text{combined}} \sim \frac{1}{2} (M_0 + M_1)$$

when adding variables with errors you would sum the
absolute errors

$$\sigma_{\text{combined}} \sim \sqrt{\sigma_0^2 + \sigma_1^2}$$

With a try, the circular mean & combined error.



$$\mu_0 \pm \sigma_0$$

$$\mu_1 \pm \sigma_1$$

Variance of combined sample mean $\frac{1}{2} (\bar{x}_1 + \bar{x}_2) = \left(\frac{\sigma_0^2}{n_0} + \frac{\sigma_1^2}{n_1} \right) \frac{1}{4}$

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

2nd analysis for each cluster we have $\mu_i, \sigma_i, \underbrace{N_i}_{\text{population}}$

$$N_i$$

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

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

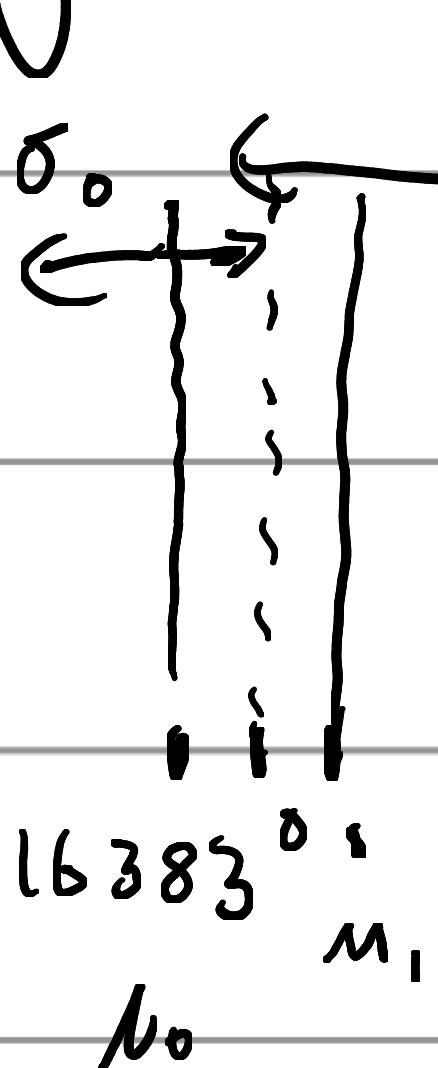
$$S_n = \sum_{i=1}^N n_i \quad S_x = \sum_{i=1}^N \sum_{j=1}^{n_i} x_{ij}$$

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

Combined results of $(\mu_0, \sigma_0, \ell, n_0) + (\mu_1, \sigma_1, \ell, n_1)$

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

lets try an example



Now population N_0 & N_1 , will be different due to the run time of each data collector we can see dependent thus could bias the data towards the cluster with the highest population. To avoid this try setting N_0 & N_1 to 1 (unity).

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

$$S_x = \sum_{i=1}^N M_i \cdot n_i = \sum_{i=1}^N \mu_i \quad \sum_{j=1}^{n_i} x_{ji} = \mu_i \cdot n_i$$

$$S_x^2 = \sigma_i^2 \underbrace{(n_i - 1)}_0 + \underbrace{\frac{1}{n_i} \sum_{j=1}^{n_i} (\mu_i \cdot n_i)^2}_1 = \sum_{i=1}^N \mu_i^2$$

$$C_n = S_n \quad C_\mu = \frac{S_n}{S_x} \quad C_\sigma = \sqrt{\frac{S_{xx} - (S_x)^2}{S_n - S_{n-1}}} \\ C_n = 2 \quad C_\mu = \frac{2}{\sum_{i=1}^N M_i} \quad C_\sigma = \sqrt{\frac{\sum_{i=1}^N M_i^2 - \frac{N}{i=1} \mu_i^2}{z-1}}$$

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

$$\sigma_{\bar{x}} = \frac{1}{\sqrt{n}} \sigma \quad \begin{matrix} \leftarrow \\ \text{population standard deviation} \end{matrix}$$

\uparrow
sample size

std dev of sample mean.

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

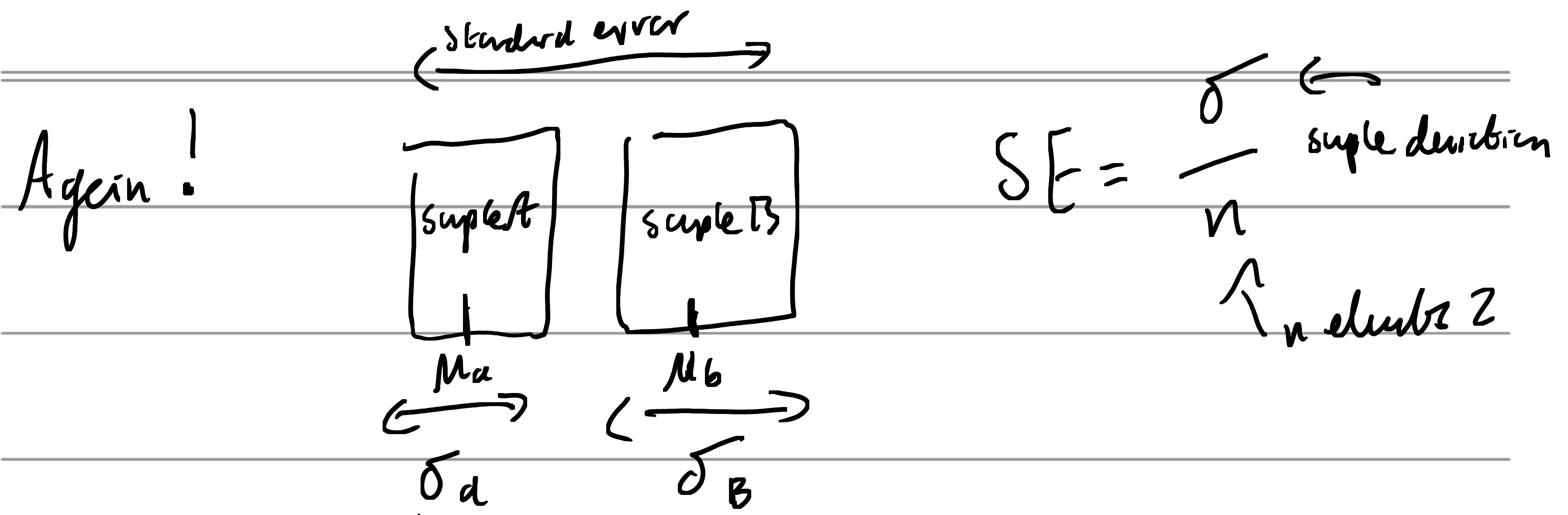
$$n = \frac{\sigma^2}{\sigma_{\bar{x}}^2}$$

Again: $\mathcal{X} = (x_0, x_1, \dots, x_n) \quad n=m=1$

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

$$z = (x_{-0}, x_1, \dots, x_n, y_0, y_1, \dots, y_m)$$

$$Z =$$



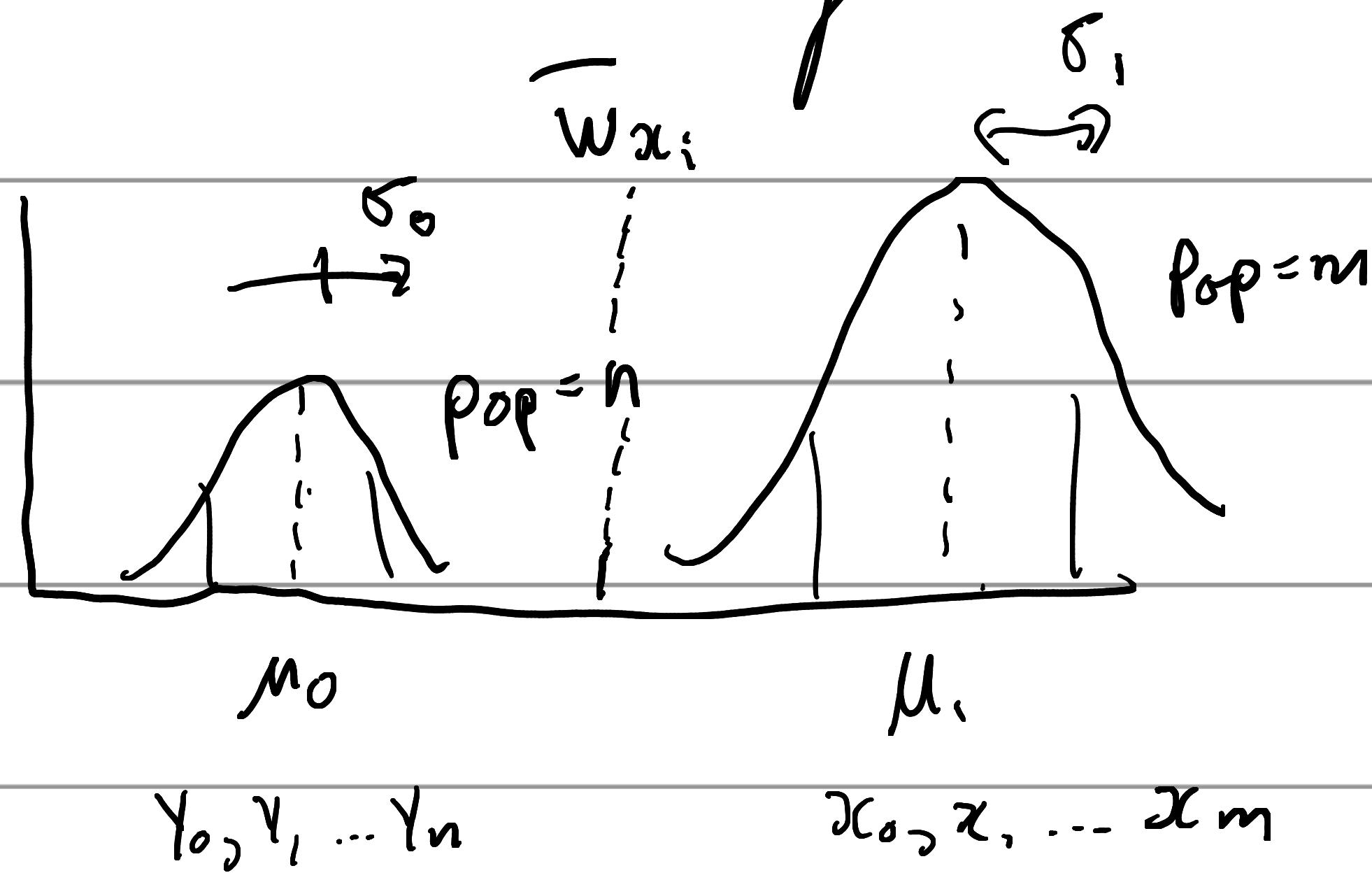
Simpler idea: use geometric cluster mean to get μ of the combined sample, then calculate the error from pairwise of cubing histogram of clusters.

Normalise
using

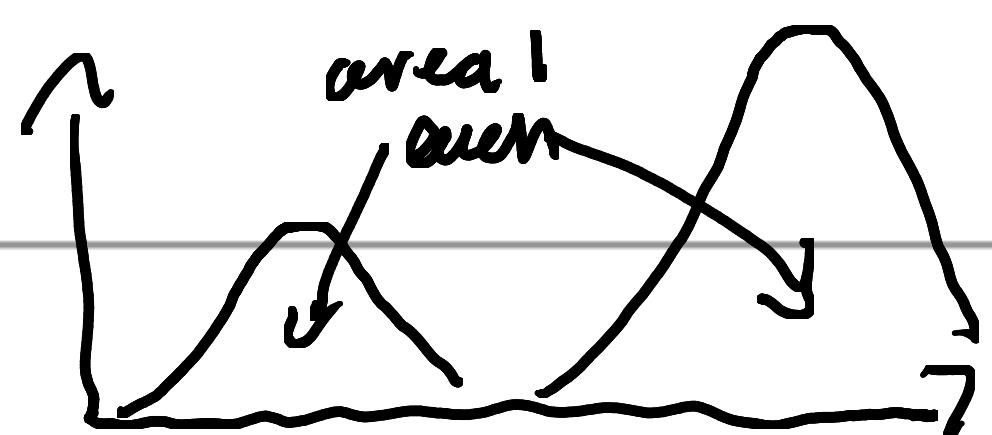
Steps will be from run $cw\backslash l\backslash cw$ (cubing channel) data, re-cluster using Kmedoids. Ignore the cerebrodes, calculate geometric cluster mean for each cubed channel, from this geometric mean use pairwise distance. Perhaps normalising each cube to a percentage the total cluster n will sum to 1, but as it will consider both cluster equally weight.

$$\text{angles} = [0 \dots 16,383]$$

We have a combined histogram like



Need to normalize this



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

so that $\sum Y'_i = 1 \quad \text{and} \quad \sum X'_i = 1$ so that the

population of each i : $1/l$, clusters elements are treated as equals.

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

$$\bar{W}_{X_i} = \frac{\sum_i w_i x_i}{\sum_i w_i} \rightarrow X' \quad \text{where no } \frac{1}{N} \text{ is it is included in } X'$$

weighted mean of X_i

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

$$Z = \{X, Y\} \quad \text{but } \sum_i^{n+m} z_i = \sum_i^n x_i + \sum_i^m y_i = 2$$

Let's go more general p sets of different samples which we

wish to weight such that the mean will not

not bias/prefer any particular set with a higher population than the

others.

p sets within Z

$$z_j = \left(\frac{y_0}{p \sum_i^n y_{0i}}, \frac{y_1}{p \sum_i^n y_{1i}}, \dots, \frac{y_p}{p \sum_i^n y_{pi}} \right)$$

should sum

$$\sum_j^p z_j = \frac{1}{p} \sum_j^p \sum_i^n \frac{n_j}{n_j} \frac{y_{ij}}{\sum_i^n y_{ij}} = 1$$

normalized

hopefully $\sum_j^p z_j = 1$

really? w_j should do this

$$\bar{x} = \frac{2\pi}{2^{14}} \sum_{j=1}^N w_j \sin(z_j)$$

mean $\left(\frac{2\pi}{2^{14}} w_i \sin(z_{ij}) \right)$

mean $\left(\frac{2\pi}{2^{14}} w_i \cos(z_{ij}) \right)$

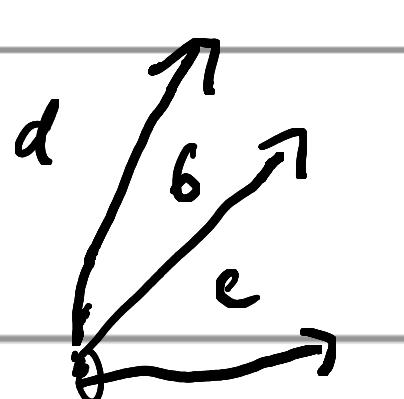
weighted mean

why this?

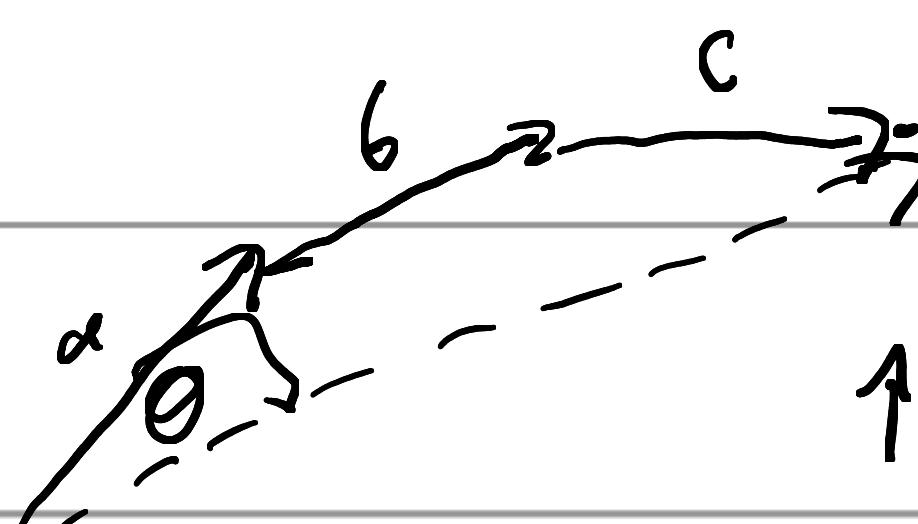
single combined z series

arc tan $2(\bar{y}, \bar{x}) \cdot \frac{2^{14}}{2\pi} \cdot 2^{14}$ = z_{ij}

copied of P



resultant vector



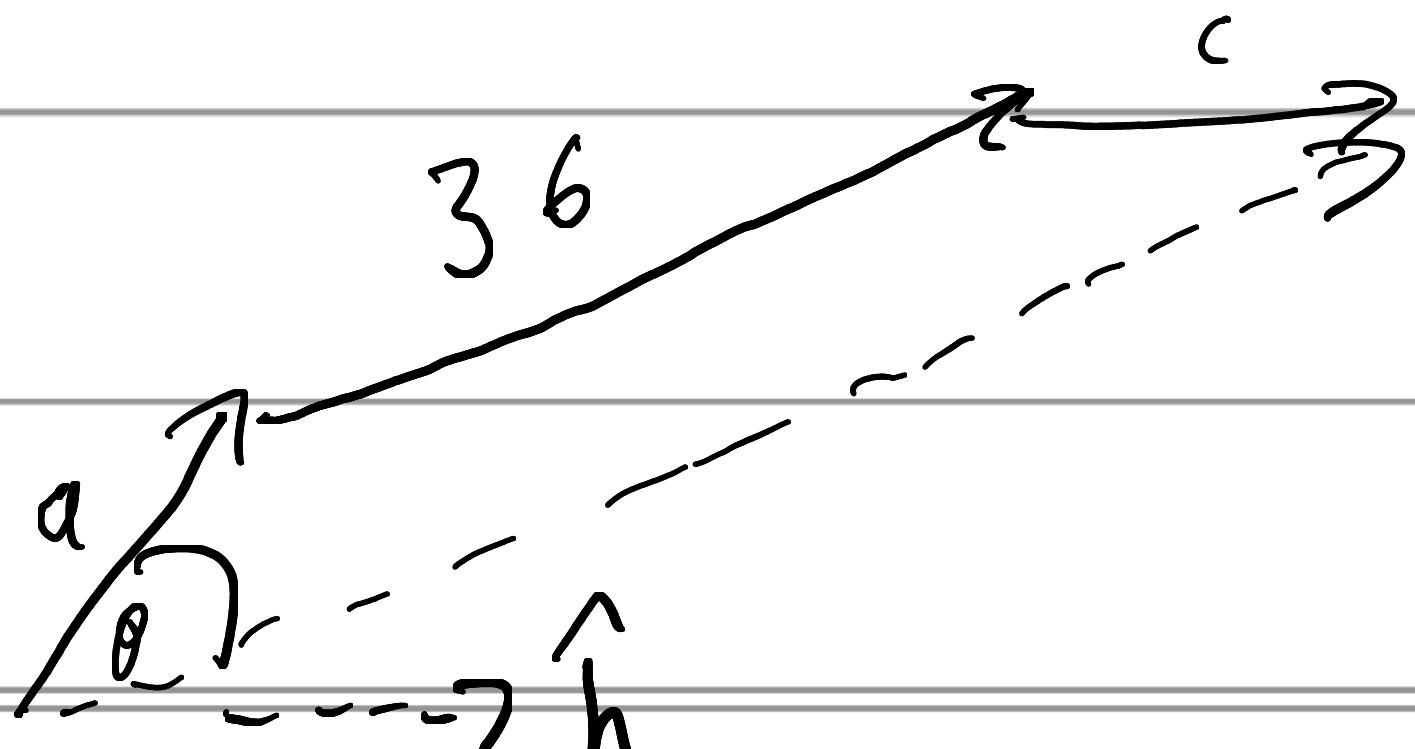
data set runs.

↑ N total data points.

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

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

then stretching b will skew the mean towards its angle.



shallow

here is flutter wrt \hat{h}

And for the error: observations

$$\sigma = \sqrt{\sum_j^N w_j (z_j - \bar{z})^2}$$

weighted mean.

$$\left(\frac{M-1}{M}\right) \sum_j^N w_j$$

\uparrow

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

number of not zero weights M

SE dev (weighted)

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

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

$$\sum_j^N w_j = 1$$

simplification if we normalize.

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 is called zero-crossing-detections.channels.inliers.json
channel names like "ZC-channel-of-detects"

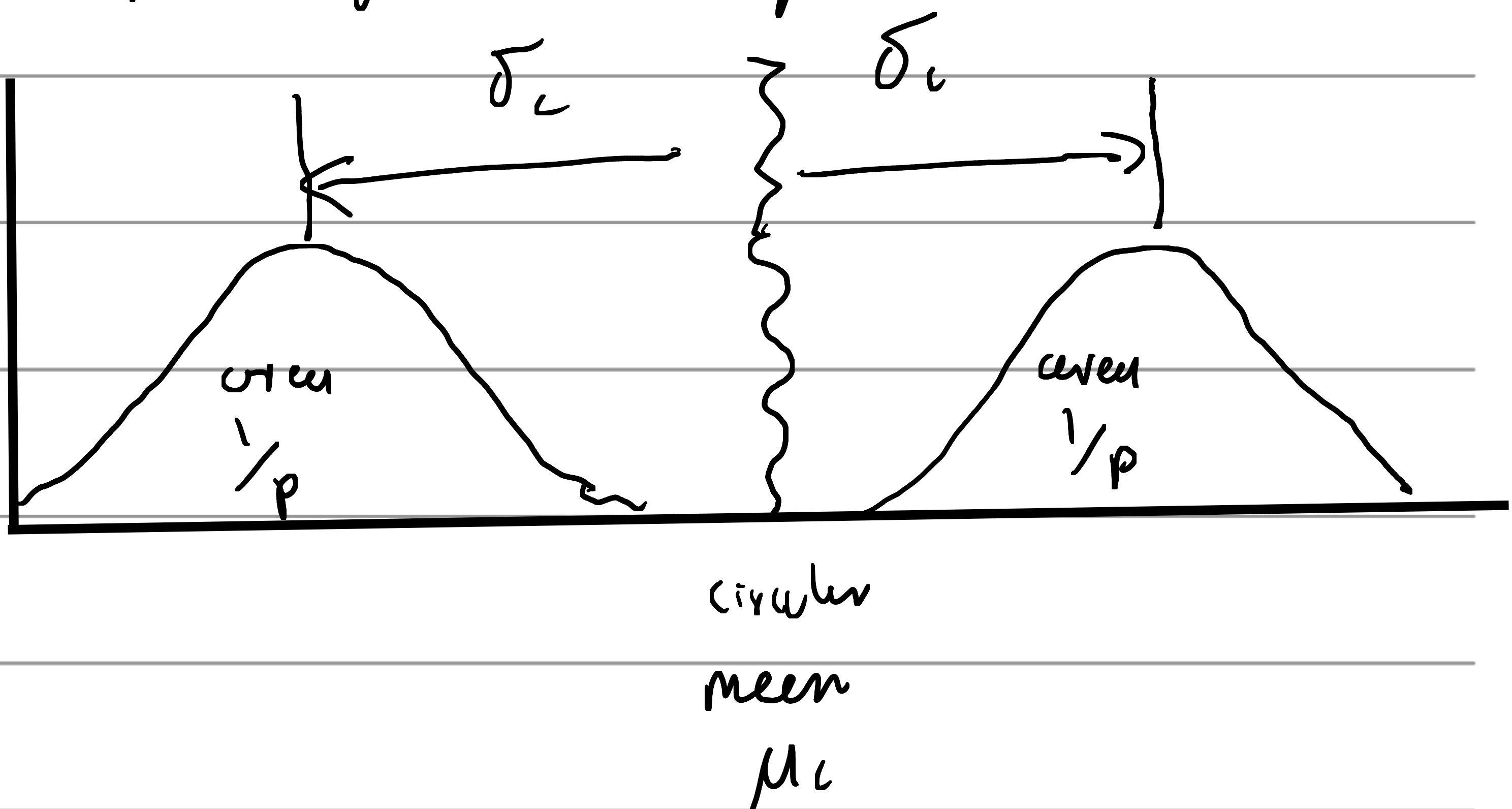
so for each run need to map channel detect, then need
to reconstruct the histogram of term kernel-x-rising/falling
per angle.

Next merge the runs as np arrays.

Then cluster with K-means.

Then for each channel cluster normalise the histogram so that it sums 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 $\{ X_{\text{cluster}_i} = 1 \}$

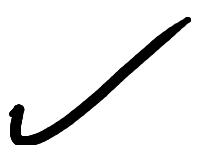
20/9/22

Working on merged clustering program.

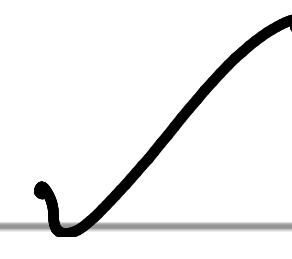
- make size histograms ✓
- make merged histogram ~
- reconstruct angles per cluster ✓
- cluster merged channels ✓
- create channel angle cluster id bipper. ✓
- iterate histograms & find channel cluster population. ✓
- Next need to normalise each histogram of counts such that

^{merged}
a^v channel clusters counts sum to 1, thus represents a probability density. The sum of a channel would be $S_c = \frac{\text{num poles}}{z}$

all
the sum of channels would be $S_q = \frac{\text{num poles}}{z} \times 6 = 3 \times \text{num poles}$



→ Next display histogram channels



→ Display old stubs mean + st-dev

sorted

→ calculate circular mean (weighted) $N \rightarrow$ got a dogg cluster

→ calculate pairwise st-dev

black rising 2

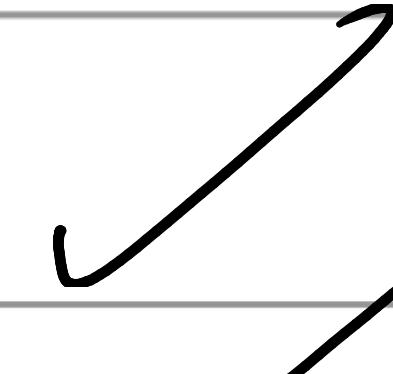
→ Display density (cluster)

??

→ Display new error + mean



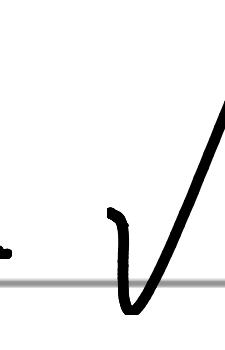
→ Recreate polarized spike brain



→ Generate binary spike brain



→ perform spectral periodicity analysis l plot.



21/9/22

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

-0.92

0.41

finished analysis for today

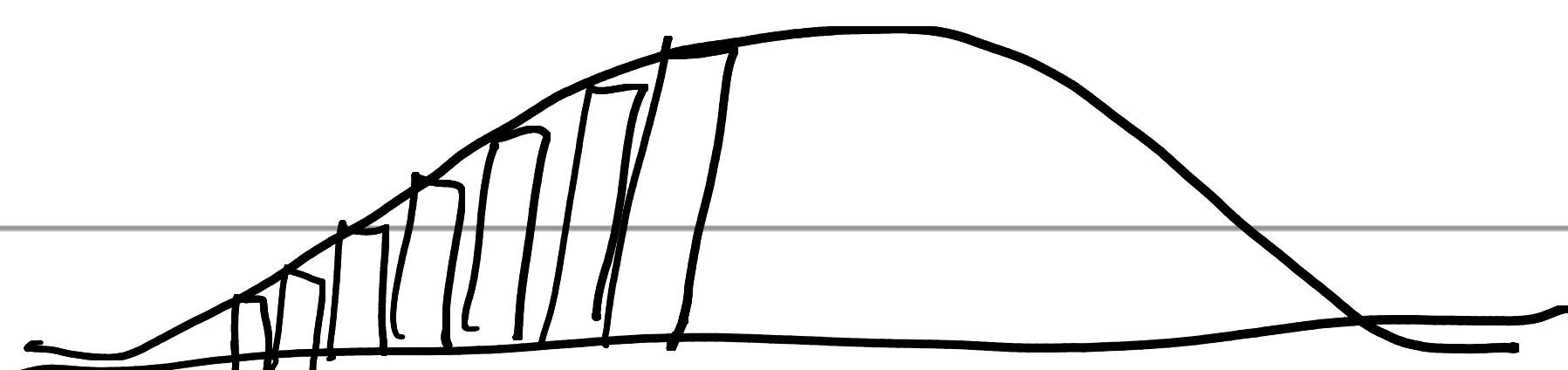
draws:

re-centralizing ggt plots

cluster density start from 1 & not 0 for consistency.

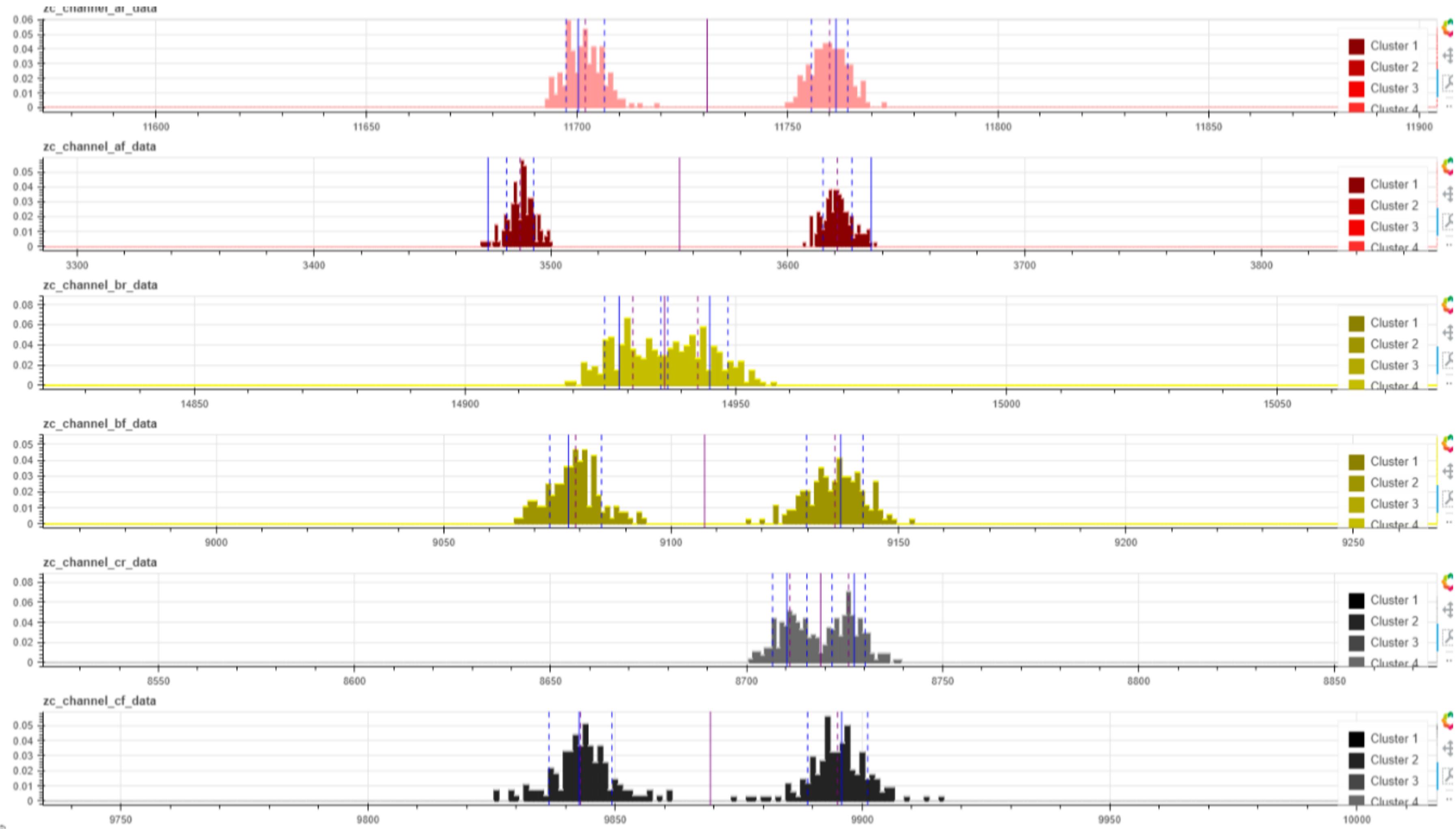
New cerebroids look good except from black rising odd one (2^{nd})
out... will re-run analysis & check.

Should take more data, check for node memory flag.



Results analysis

22/9/22



Above is a chart of the merging multiple runs to Sept. cw &

16 sept-4 - cw, both runs of about 3-4 seconds. Notice the circles

mean gets a good midpoint between the normalized databases

because of the applied weighting giving equal votes to each database

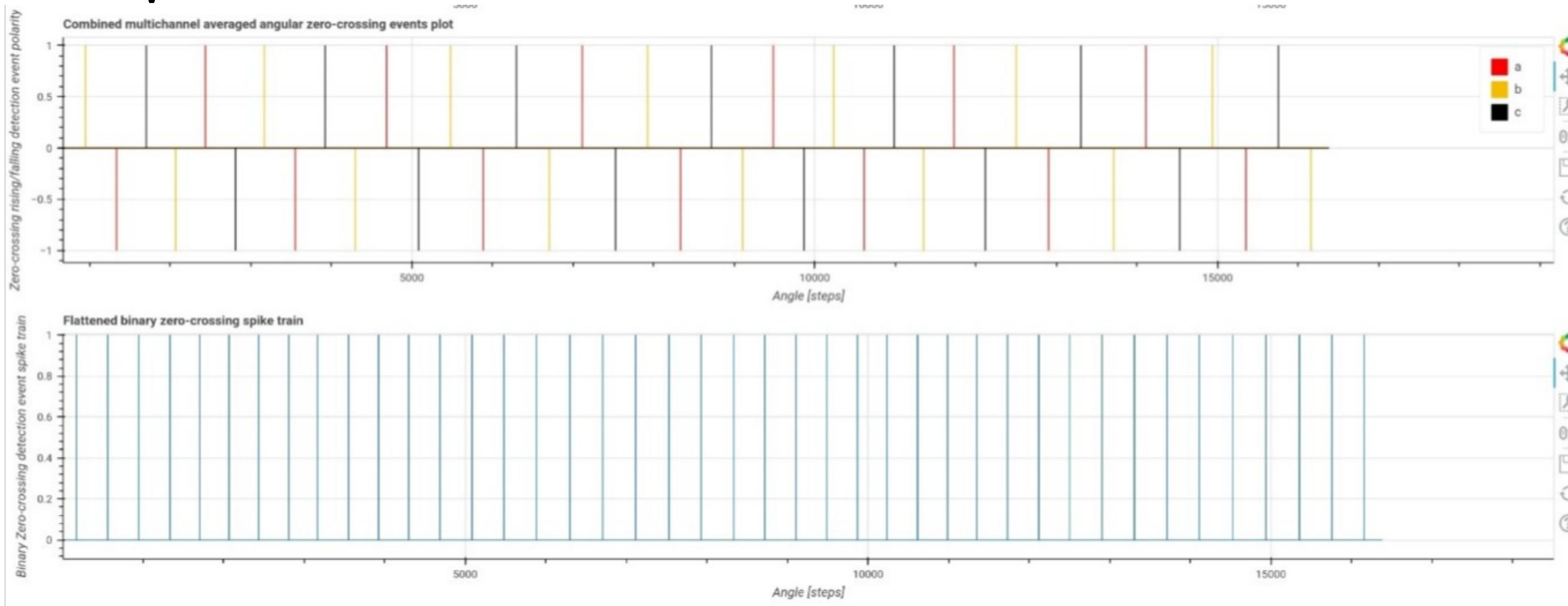
regardless of the run time. Note the clusters approximate gaussians.

We can see the systemic error here. Channel B rising (black)

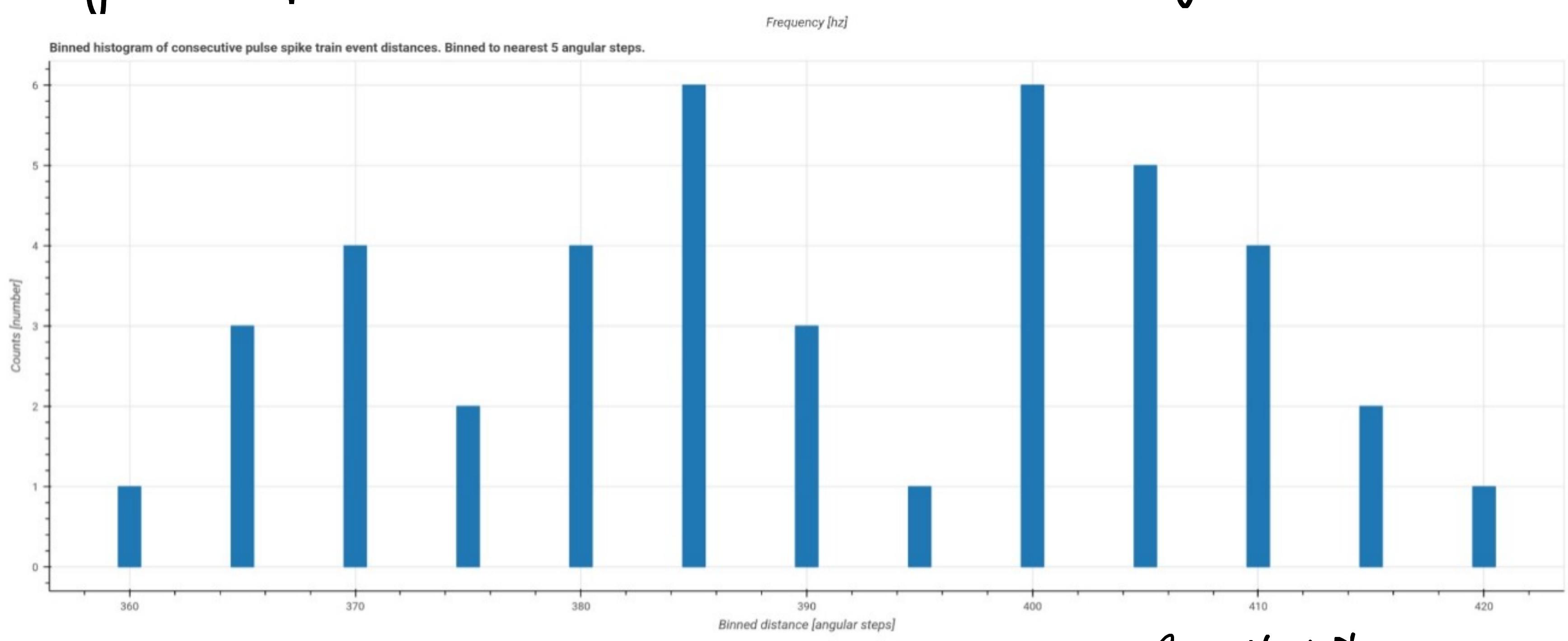
The gaussians overlap as there is low noise, (rising is similar. All others

have some separation, with cw with long short & ccw with short long distances
here they both overlap.

Analysys continued



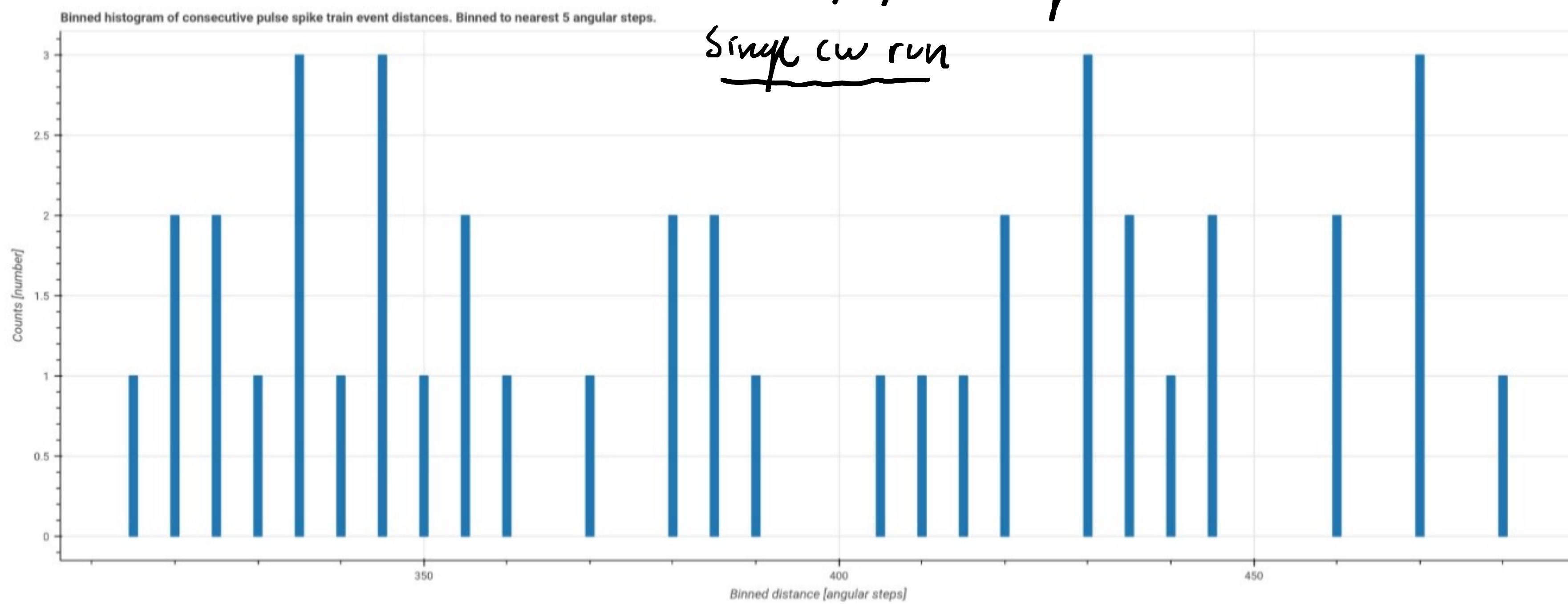
Here we see the result of the spike train by taking the circular mean between the two databases self-identified as a cluster as per kmedoids clustering. Note there is no discernable short, long nor long, short pattern... the results seem much more evenly spaced.



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

The histograms for the spike train distances for a single cw run

combined was much less convincingly a single distance.



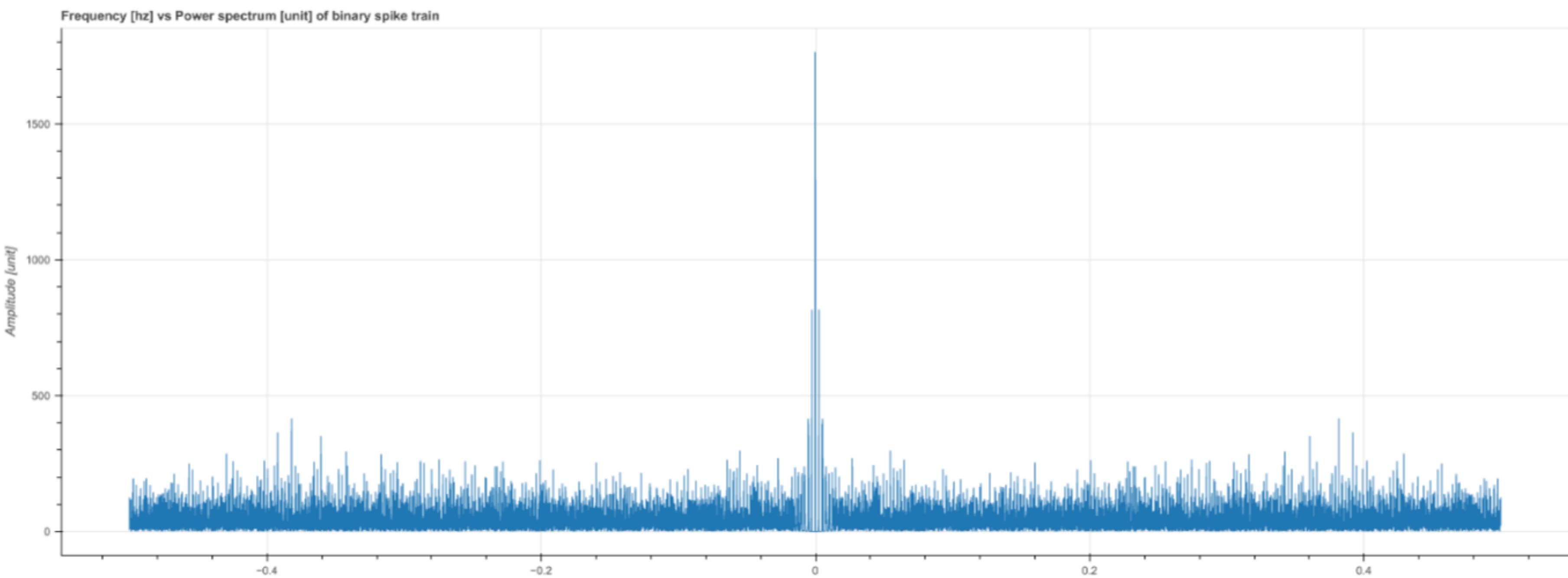
Instead for the uncubined run there are more like 3
distances, long, short & somewhere in between.

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

FFT of combined dataset.

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

In order to determine the periodicity 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 train. The distances can be rounded and binned into a histogram showing us the number occurrence 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.

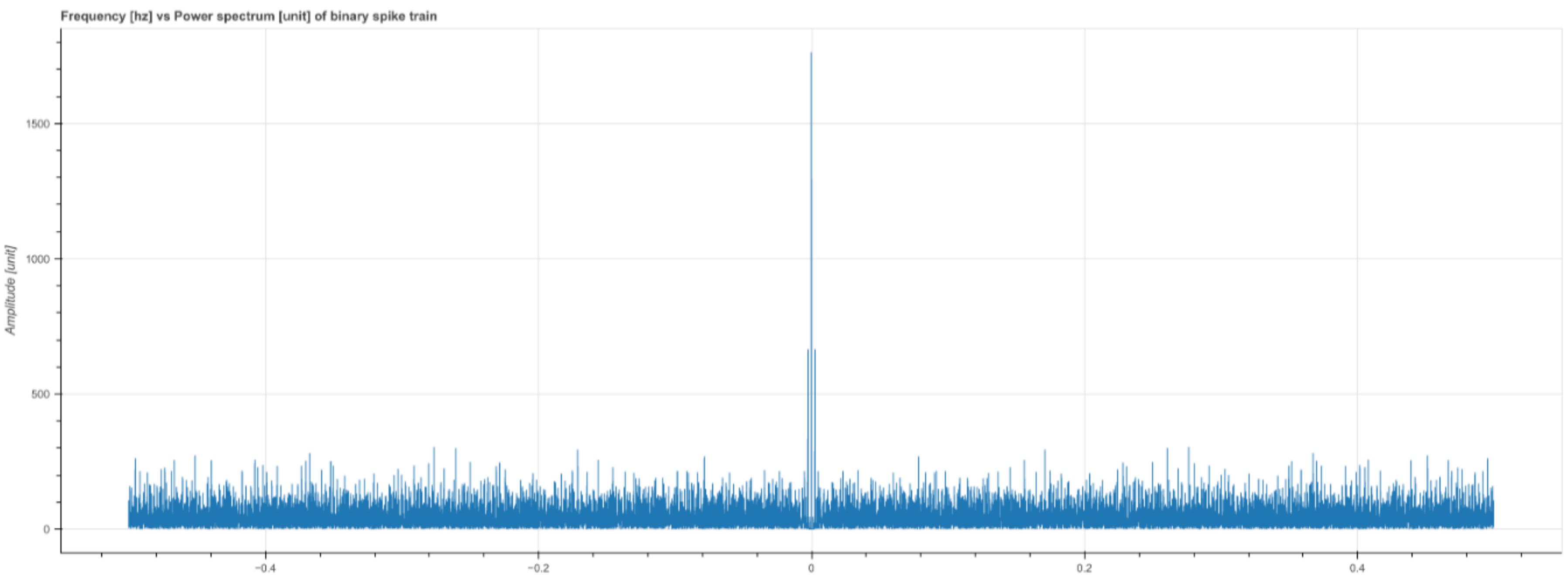


Although admittedly hard to see in the image above, the central region has some structure. The FFT of a uniform spike train is a spike train

so we are seeing some signal here. Compare that to the FFT of a single run below:

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

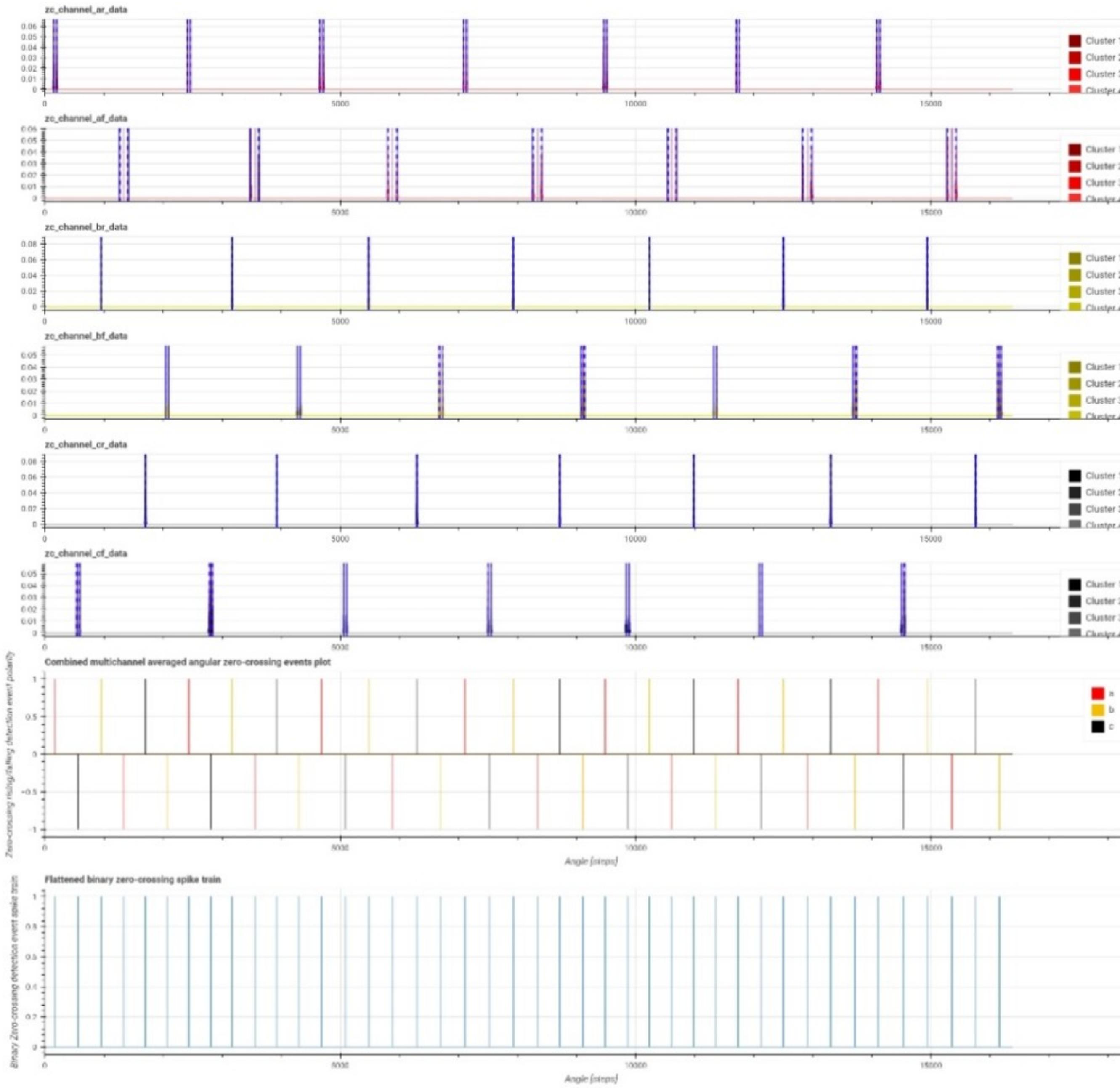
In order to determine the periodicity of the spike train there are two methods, one generate an fft on the spike train and look for peaks in frequency which dominate, the second strategy is to measure the distance between each zero-crossing spike with the next spike in the train. The distances can be rounded and binned into a histogram showing us the number occurrence 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

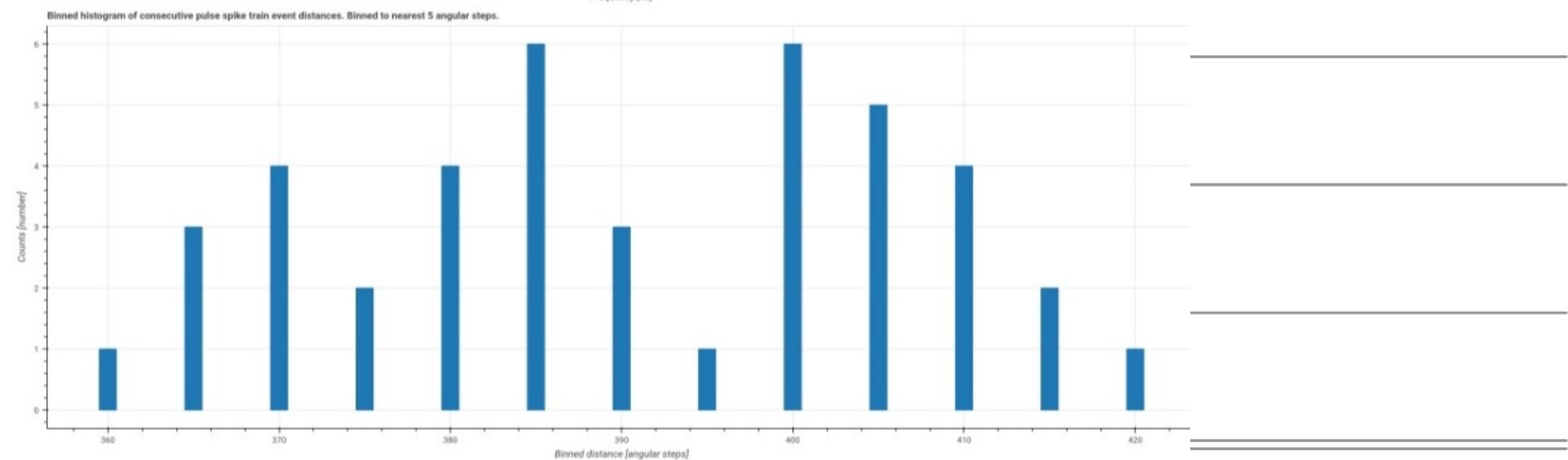
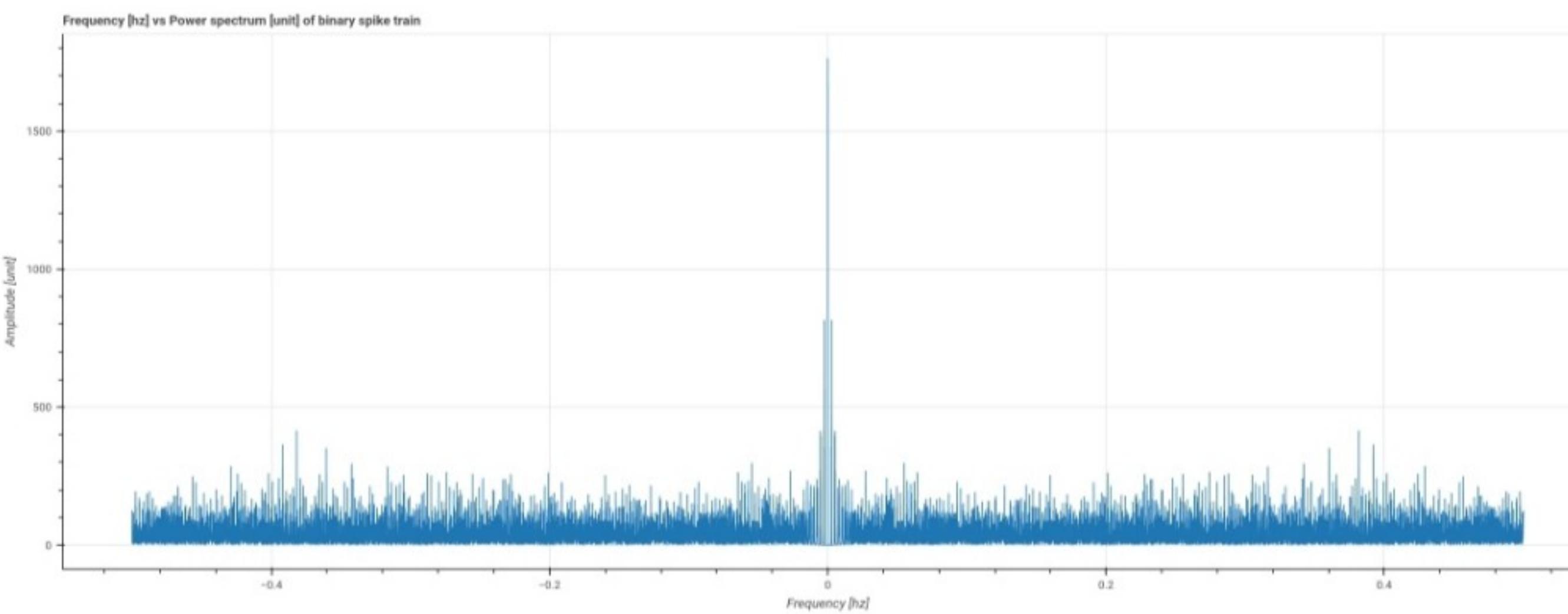
is not enough evidence of a spike train, this result is less periodic

& clean than the combined.



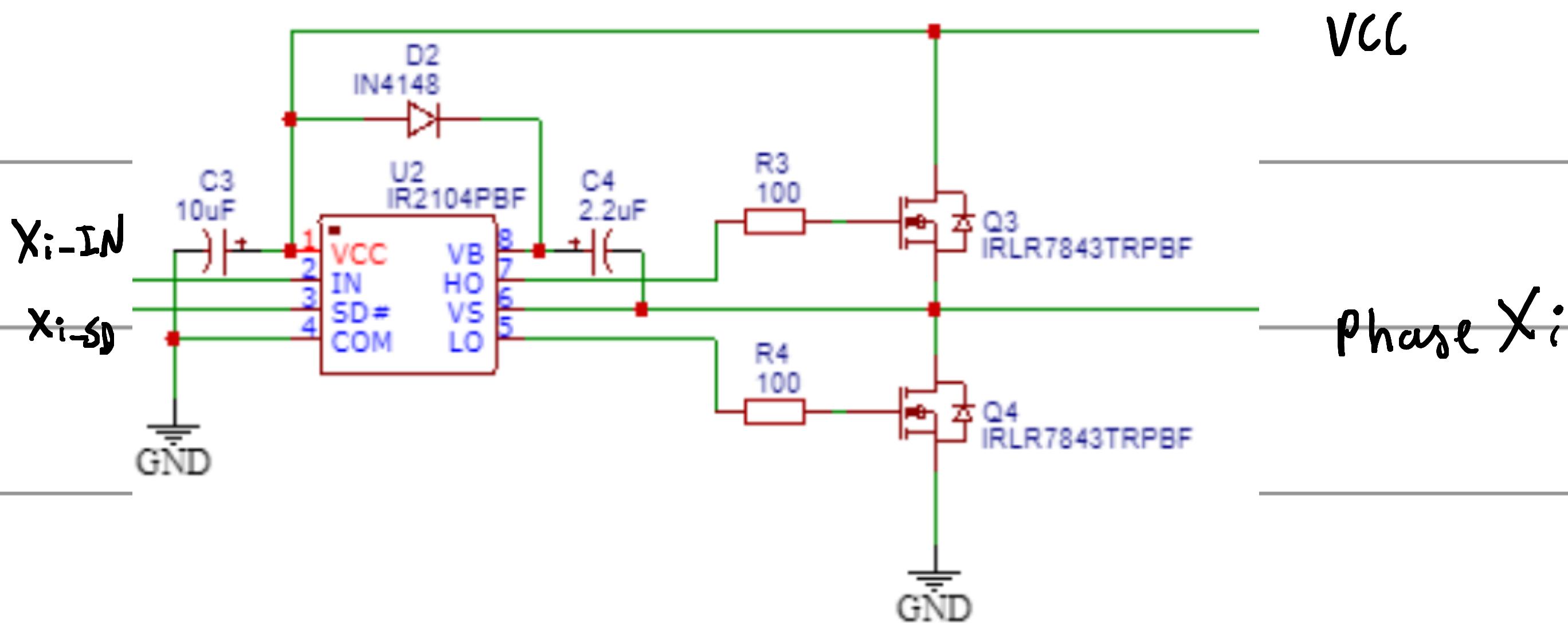
Temporal / spectral analysis of combined binary zero-crossing event spike train

In order to determine the periodicity 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 distances can be rounded and binned into a histogram showing us the number occurrence 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.



23/9/22

Thinking about the ESC:



VCC

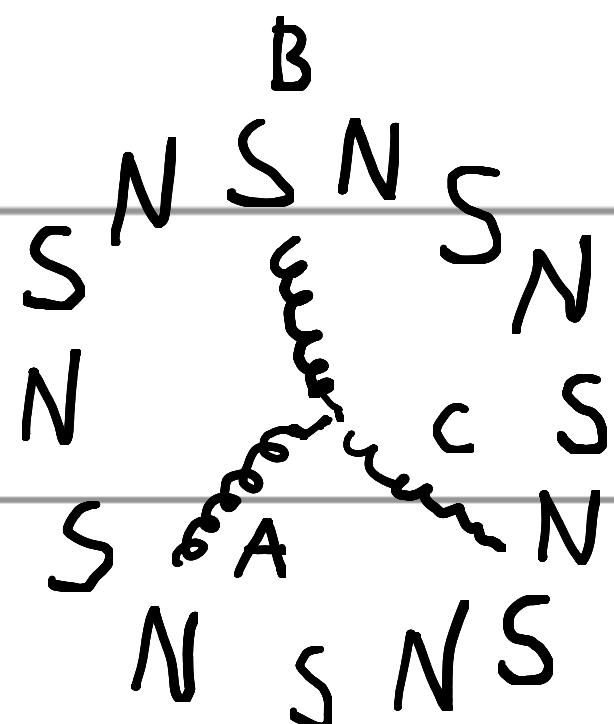
Phase X_i :

Counter-clockwise rotation (-ve)

ESC	0	1	2	3	4	5
IN	C	C	B	A	A	
SD	B	A	A	C	C	B
RISING	A	X	C	X	B	X
FALLING	X	B	X	A	X	C

clockwise rotation (+ve)

ESC	0	1	2	3	4	5
IN	A	A	B	B	C	C
SD	B	C	C	A	A	B
RISING	C	X	A	X	B	X
FALLING	X	B	X	C	X	A



ESC Electrical Step Counter

RISING / FALLING row indicates when phase to sample & whether we would expect a rising/falling edge of the next ~~zero~~ zero-crossing

IN identifies which 'IN' phase pin should be high during this commutation step. The other two phases should be low.

SD indicates which 'SD' phase pin should be written to with a PWM signal with a duty cycle equal to the THROTTLE variable. The other 'SD' phase pins should be set to low.

	0	1	2	3	4	5
0	A	A	B	B	C	C
1	B	C	C	A	A	B
2	C	A	A	B	C	A

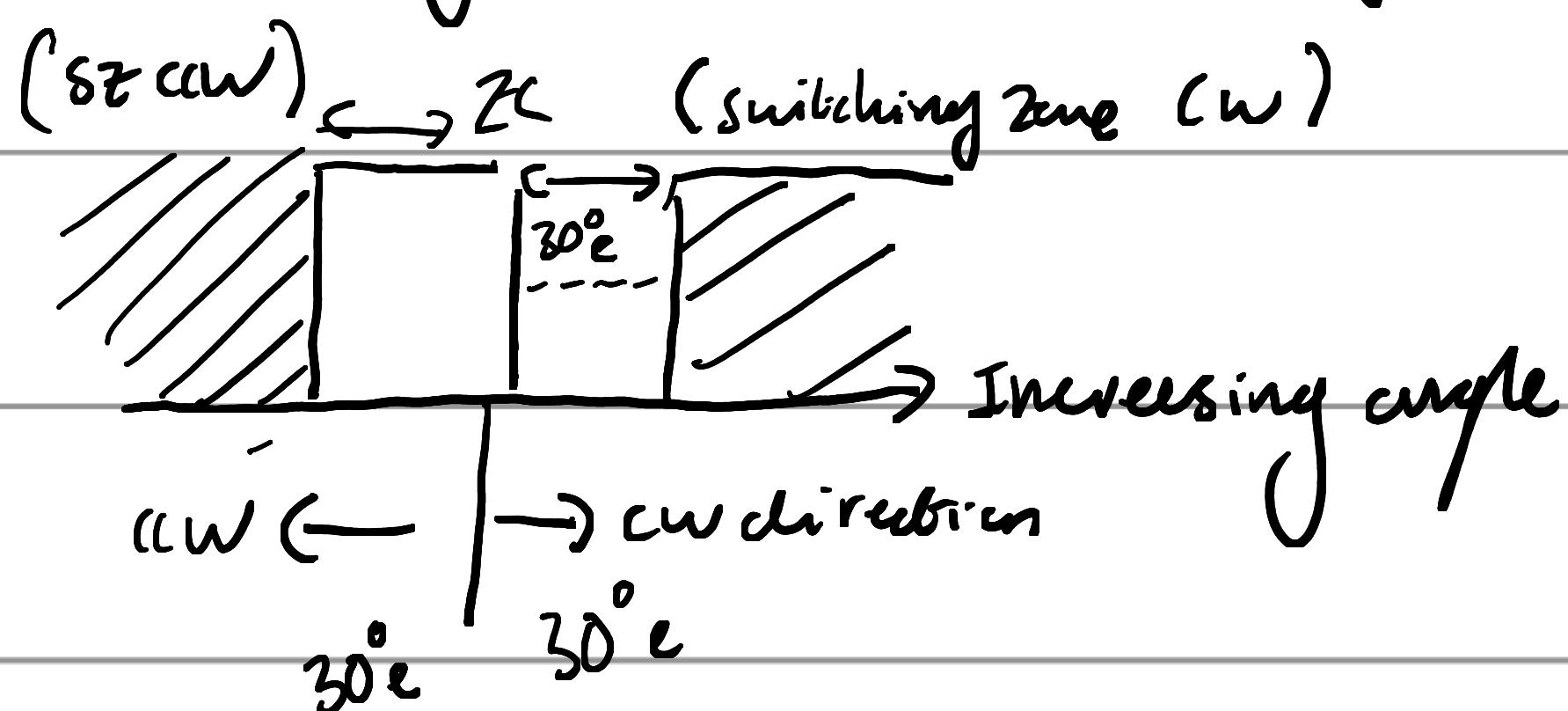
For each direction there are 6 total steps

Step rise/fall	5 4 3 2 1 0				
	↑	↓	↑	↓	↑
SD CCW sequence	A	B	C	A	B

Step rise/fall	0	1	2	3	4	5
Phase delay	↑	↓	↑	↓	↑	↓

Step rise/fall	0	1	2	3	4	5
Phase delay	C	B	A	C	B	A

Zero crossing mark can help identify the ideal switching point.



6 electrical steps per cycle, $\frac{\text{poles}}{2}$ electrical cycles per mechanical cycle. 16,384 angular steps (full mechanical cycle).

30 electrical degrees displacement is what?

What is the electrical steps per mechanical cycle? $\frac{\text{poles}}{2}$

$$\frac{360 \cdot 2}{\text{poles} \quad 360} = \frac{2}{\text{poles}} \text{ mechanical deg per electrical deg.}$$

Electrical cycle (360°) is ~ 51.42 degrees mechanical

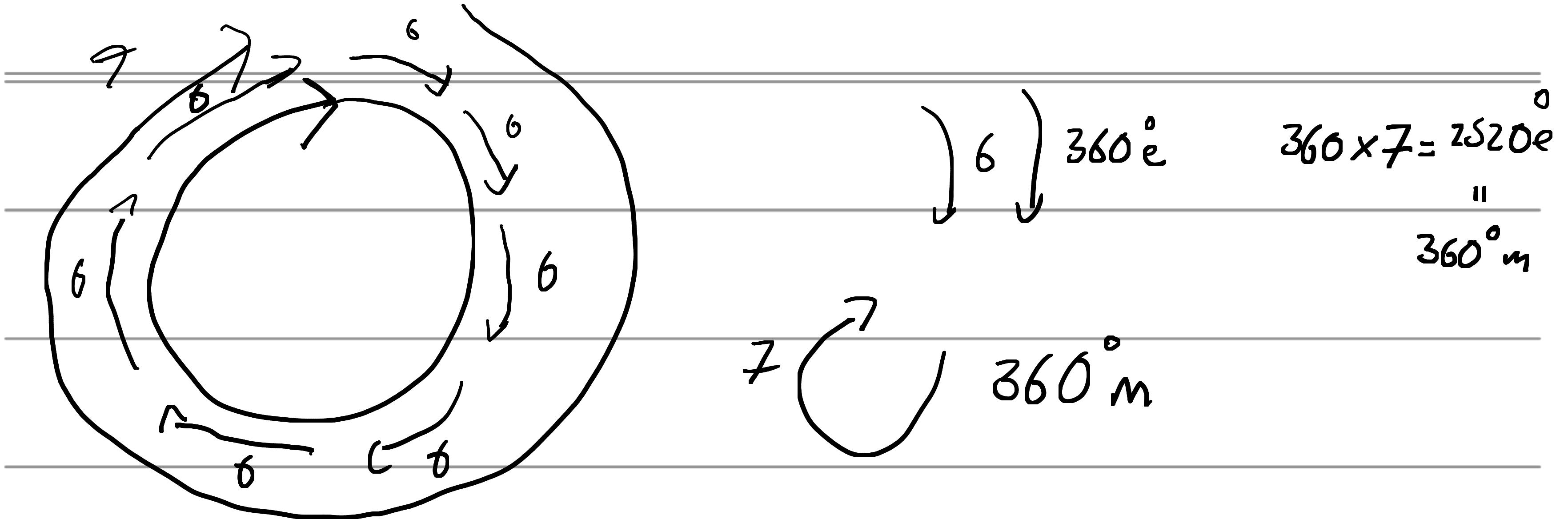
3 poles =

$$\frac{14}{2} \times 6 = 3 \times 14 = 42 \text{ switching events.}$$

$$\frac{360^\circ}{7} = \frac{51.42}{7} \text{ electrical cycles per mechanical cycle}$$

\downarrow
 $\sim 0.014^\circ \text{ deg}$

42 steps is 360° mechanical deg



$$\frac{360^\circ_m}{2520^\circ_e} = \frac{360^\circ}{2520^\circ_e} = \frac{360^\circ m \cdot 2}{360^\circ e \times \text{pades}}$$

(1) What sample rate is needed to get at least 1 sample per angular step?

Step 5 for a motor spinning $\omega = 20\text{ k rpm}$

(2) what sample rate is needed to be with X angular steps of the ideal switching point; for a motor spinning $\omega \text{ rpm}$

of the ideal switching point; for a motor spinning $\omega \text{ rpm}$

Solution to ①: convert rpm into Hz, $\omega = 20,000 \text{ rpm}$ in 60 s

$$\frac{20,000 \text{ rps}}{60} = 333.\dot{3} \text{ rps}$$

$$\text{period} = \frac{1}{f} \text{ s} \quad t = \frac{1}{f} = \frac{60}{\omega [\text{rpm}]} \quad f = 333.\dot{3} \text{ Hz}$$

$$t[s] = \frac{60}{\omega[\text{rpm}]} \quad t_{20,000 \text{ rpm}} = 0.003 [\text{s}] *$$

There are 2^{14} or 16,384 values round the circle.

$$l[\text{hz}] = 16,384 \text{ [angular step per second]} \text{ [dsps]}$$

$$f[\text{hz}] = f \cdot 16,384 \text{ [dsps]}$$

$$f[\text{hz}] = \frac{\omega[\text{rpm}]}{60}$$

* \Rightarrow 0.003 [s] to complete a cycle, need at least

16,384 samples in this time to get 1 sample per angular step (S_a)

$$S_a \cdot 2^{14} \cdot \omega[\text{rpm}] = \text{sample rate [s]}$$

$$\frac{16,384 \text{ [samples]}}{0.003 \text{ [s]}} = \frac{1 \text{ [angular sample]}}{1 \text{ angular step}} \frac{16,384 \cdot S_a}{0.003 \text{ [s]}}$$

$$= 5.461 \text{ MHz!}$$

As S147 SPI slave up to

10MHz (need to check driver)

Need to create a new simple Beesyg program to extract as many angles as possible I use GPT 1 second timer to get the angle increments / second. Then log angle vs sample rate. Improve (decrease) delay time whilst still stable. Aim for 8MHz which would allow for 28krpm. $7.646 \sim 8\text{MHz}$
 $\downarrow 7\text{s cpu steps} @ 600\text{MHz!}$ \hookrightarrow overload? 800MHz?

Not a problem for the slow speed $33\text{ rpm} \sim 9\text{ kHz}$

70 rpm 11.1 kHz $\uparrow 20\%$ 23 kHz $26\text{k CPU steps} @ 600\text{MHz}$ $66\text{k CPU steps} @ 600\text{MHz}$

Kalman filter would probably not be possible @ high rpm; only simple switching.

$8 - 10\text{ MHz } 1.2 \times 34$

$5.508\text{ MHz } 20\%$

AZ212 1000kv @ 16.8V (4s) = 16,800 rpm 4.59 MHz
 $130\text{ CPU instructions.}$

THROTTLE 4006 620kv @ 16.8V (4s) = 10,416 rpm 2.84 MHz
 $3.408\text{ MHz } 20\%$ $211\text{ CPU instructions}$

$$S_a \cdot \frac{2^{14} \circ (W_{\text{crpm}})}{60} = \text{sample rate (s)}$$

if using slave (beesyg generates)
 $3.5 \text{ poles } \circ \left(\frac{W_{\text{crpm}}}{60} \right)$

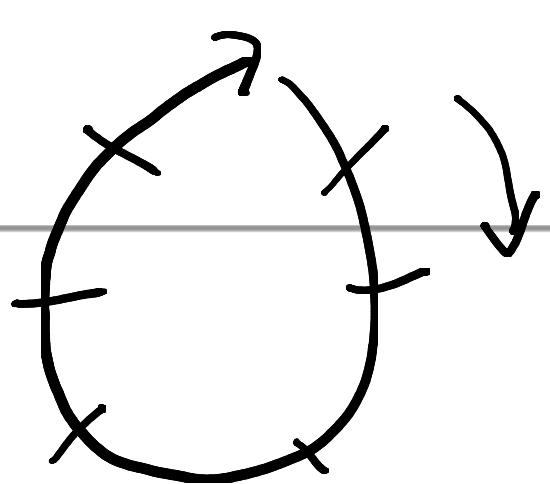
$3.408\text{ MHz } 20\%$ $211\text{ CPU instructions}$

11.2 kHz $4006 \sim 5.73\text{ MHz}$

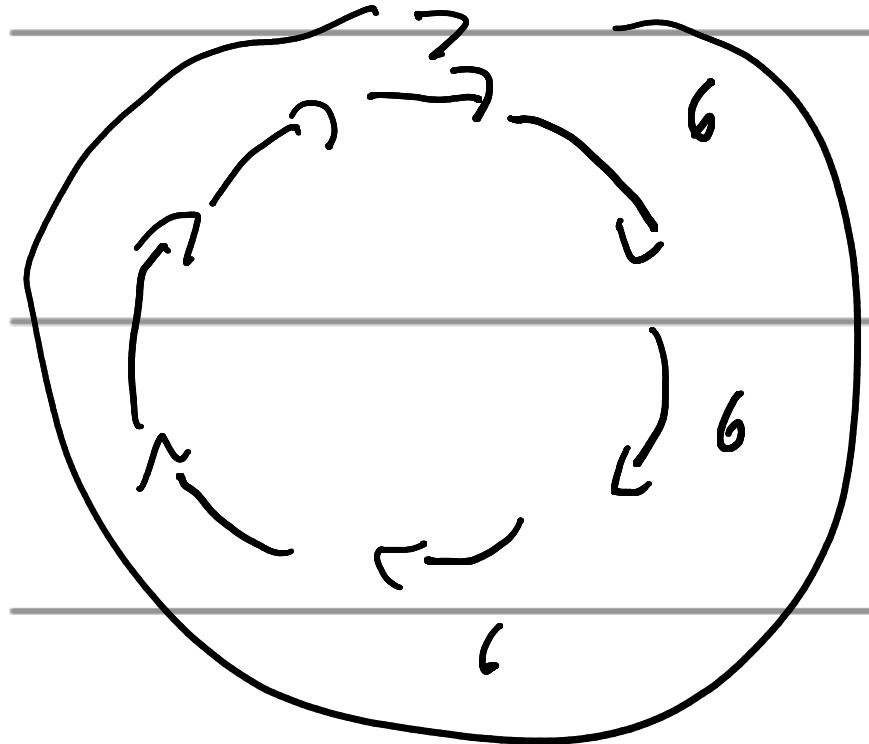
\uparrow Much lower than 96kHz

Sanity check

$$360^\circ_e$$



$$\frac{360^\circ_e}{6} = 60^\circ_e \text{ per electrical step.}$$



7 times ($\frac{\text{Poles}}{2}$) of 6 steps or 42 (poles.3)

steps around

$$360^\circ_M$$

s

$$\frac{360^\circ_M}{7} = 60^\circ_e$$

42 [steps mechanical]

$$\begin{bmatrix} 0 \\ M \end{bmatrix}$$

$$\frac{360^\circ}{3 \cdot \text{poles}} = 60^\circ_e \quad \frac{360^\circ_M}{3 \cdot \text{poles} \cdot 60^\circ_e} = 0.143 = \frac{3^\circ}{\text{poles}_M} = 1^\circ_e$$

$$\frac{2\pi}{3 \cdot \text{poles}} = \frac{\pi}{3} \text{ radians}$$

$$\frac{6\pi}{3 \cdot \text{poles} \cdot \pi} = \frac{2}{3} \text{ poles}$$

$$1 e = \frac{2}{\text{poles}} \frac{1 \text{ radian}}{M}$$

$$|e = \frac{2}{\text{poles}} \cdot |^{\circ} \text{m}$$

$$|e = \frac{2}{\text{poles}} \cdot |^{\circ} \text{m}$$

$$30^{\circ} \text{e} = \frac{2}{\text{poles}} 30^{\circ} \text{m}$$

$\frac{360^{\circ} \text{e}}{360^{\circ} \text{m}}$
 $\frac{30^{\circ}}{360^{\circ}} \times 16,384$

$$\frac{16,384 \text{ [steps]}}{42 \text{ [zc-events]}} = \frac{2^{14}}{3 \cdot \text{poles}}$$

$$2^{14} \text{ poles} \cdot |^{\circ} \text{m}$$

Mechanical 11

Steps per event. ~ 390.01

Agrees with experiment.

Just divide
this by 2¹⁰.

$$30^{\circ} \text{e} = \frac{2}{\text{poles}} 30^{\circ} \text{m} =$$

$$4.28^{\circ} \text{m} \quad \frac{4.28^{\circ}}{360^{\circ}} \times 16,384 = 195. \text{ steps}$$

see note 5

Encoder reads 16 bits with 3ns delay plus cpl time.

Minimum of 48 ns delay + cpl time per encoder read.

$$0.0048 \mu\text{s} \quad \left\{ \begin{array}{l} 8 \text{ MHz} \\ 29 \text{ clocks} \end{array} \right. = 1.25 \times 10^{-7} = 0.125 \mu\text{s}$$

7s clocks

$$600 \text{ MHz} = 0.0016 \mu\text{s}$$

1 clock

Thinking about repositories: record-deck,

record deck

sub repo

→ digital ESC / low speed - Kalman - speed control - PID / SM

| high speed - speed - control

| high speed - modulated - speed-profile

↳ configurable.

other blocks

analog - A/DMP - Arduino UNO

- teasy 40

analog - ADC - teasy 40

40 steps min

7.36 max

Estimated cluster error 16 sept - CCW 4,5,6,7 steps.
analysis, outliers

worse cases

$$\frac{420}{390} = 1.077 = 7.7\%$$

from binned

$$1 - \frac{360}{390} = 0.077 = 7.7\%$$

multirun
combined.

± 8 steps

switching ZONE ~ 390 steps

$$8/390 = 2\%$$

error.

Quite a big max error, last option merge carlo of whole sequence, allow to shift +/- avg pos with probability calculated from cluster probability density, assume ideally symmetric, energy function minimise error in spanning. Pick random ZC, pick random shift, weight shift probability by cluster density, calculate probability $P^{-\frac{4.3}{T} \Delta E}$ monte carlo, accept or reject, measure adjusted spanning metric. AKA assume symmetric & do a global optimisation.

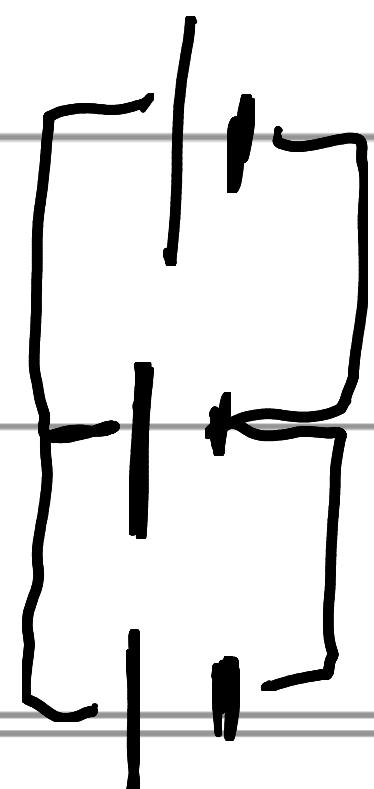
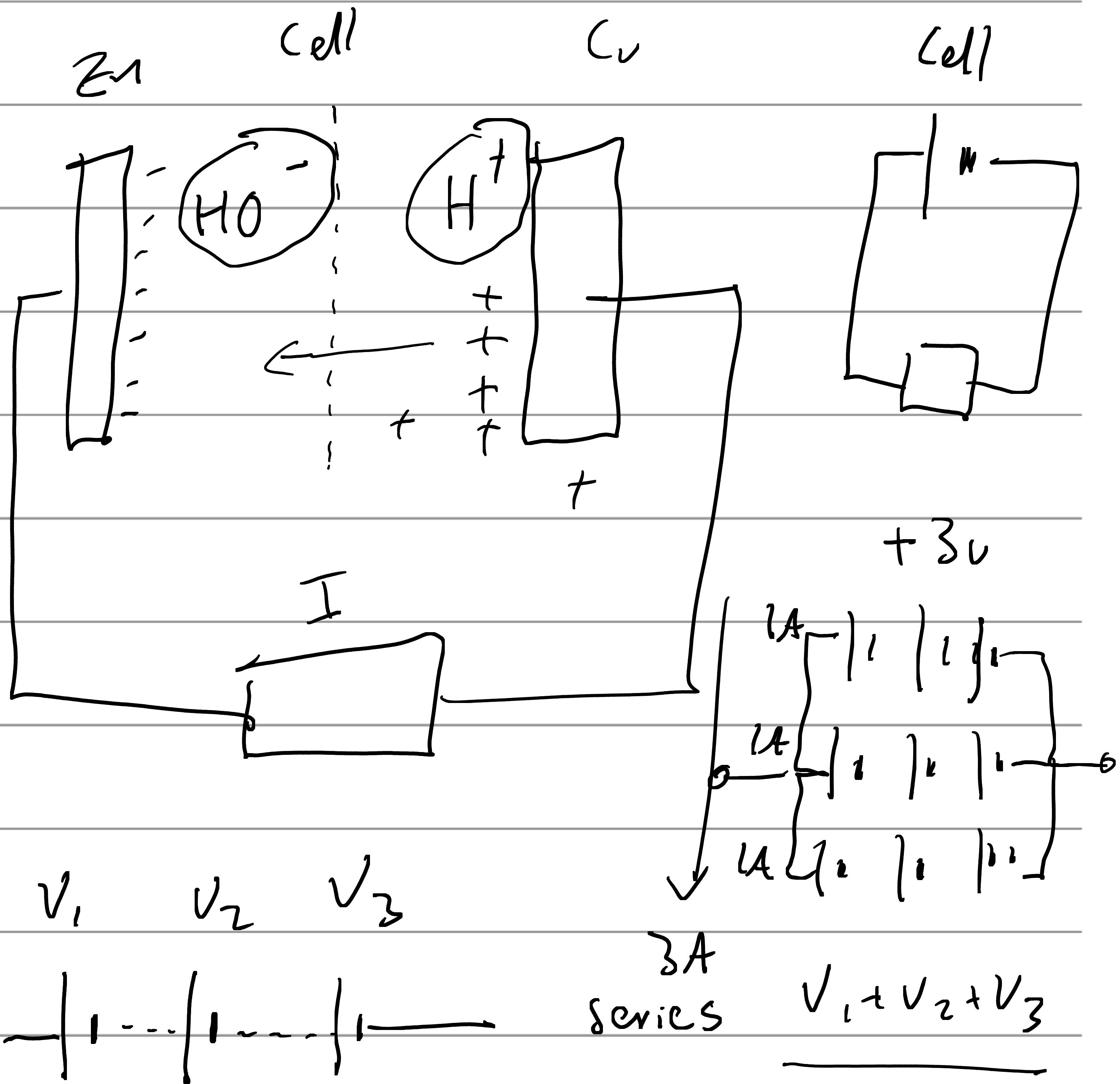
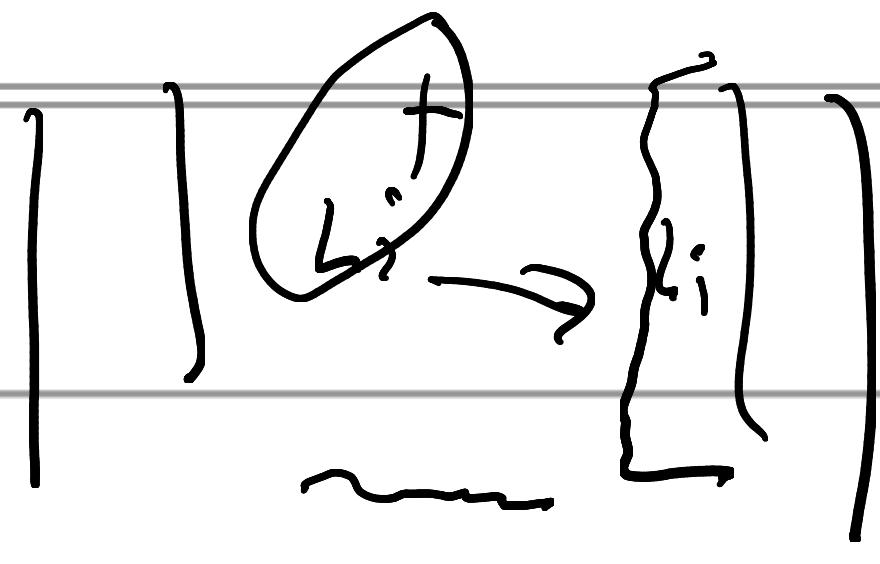
Centroids fixed within 3σ , perturbed randomly, measure global error in distance of consecutive points.

28? 95% rather than 30 99.7%

Simpler perhaps brighter outlier elimination, check outlier charts!

Need to display quantitative error stuff on reports, use a class! Relative error w.r.t switching zone, switching zone length annotation.

min/max/avg cluster error, errors from single clusters vs merged etc.
much more data!



parallel

$$A_1 + A_2 + A_3$$

A_h

math

$$\frac{1}{100^\circ}$$