Applications of a Length-Limited Parametric Array: Benignly Excluding Birds

Matthew Groves
Senior Physics Thesis
May 2016
advised by Dr. Mark Hinders

Abstract

Birds present major problems in both the developed and the developing world, and the approaches to exclude them are broadly ineffective or inefficient. Noticing that birds are highly social creatures, we aim to fill the frequency range that birds use to communicate with static, so they cannot communicate with each other and subsequently leave. The static is within the audible spectrum, so we are applying two acoustic techniques – the parametric array and length-limiting – to create highly directional sound and minimize noise pollution. Specifically, we collected data of a length-limited effect at various parameters, which we used to benchmark simulations and maximize the effect; Measurement 12 provides the strongest example.

Table of Contents

List of Figures

List of Tables

1. Introduction and Motivation

Our research focuses on an application of two specific acoustic wave techniques. The first, a parametric array [1], creates a highly directional sound column with a relatively narrow width and has been well-evidenced for decades [2]. The second effect, length-limiting, sets a distance at which the sound column will rapidly die off, but is less well-established in the literature [17].

These acoustic techniques allow us to tackle a specific application with much more scientific precision than ever before: bird exclusion. In the developed world, birds range from being pests – eating crops, dropping feces on storefronts or sidewalks, and disturbing tourism sites – to being legitimate dangers if sucked into a plane engine. On a financial basis alone, bird interference cost the United States agricultural industry over $800 million per year [3] and global civil aviation industry over $1 billion per year [4], with nearly 100,000 civil aircraft bird strikes from 1990-2009 [5]. In the developing world, agricultural damage can cause “ruinous crop losses and endanger a small-scale farmer's source of sustenance” [6].

Many birds are highly social, relying on acoustic communication for many facets of their ecology; when human-generated noise from traffic or machinery overlaps in frequency with their communication range, the species departs the area [7]. Therein, if we can blanket the relatively narrow frequency range that birds use to communicate, we can fool the birds into leaving the area. In principle, regular speakers perform this function equally well. In practice, the frequency range we blanket is in the audible spectrum and grates human ears. The directionality and length-limiting features will make the concept much more applicable in locations where noise pollution is a major concern.

Additionally, the other techniques for excluding birds from an area are ineffective, impractical, or both. “Birds quickly habituate to startle regimes [8],” such as traditional scarecrows or audible scarecrows, which emit cries from predators through speakers in hopes of frightening away target birds.

Early results show that our technique does not have a similar decrease in effectiveness over time [9].

2. Theory and Simulations

In 1963, Peter Westervelt's research [1] described a new technique to produce a highly directional sound beam by manipulating the nonlinear properties of air. His “Parametric Array” emits two ultrasound frequencies which propagate through the air and interact with each other. Similar to any wave interaction, this overlapping region creates two additional frequencies, with frequencies equal to the summation and difference of the original two ultrasound waves. The ultrasound and summation frequencies attenuate rapidly, but the difference frequency propagates much further. By virtue of the ultrasound frequencies chosen, the difference frequency lies in the audible spectrum and propagates much farther and with much greater directionality than sound waves emitted from traditional speakers. The chart below (Figure 1) demonstrates this effect propagating over distance, with an audible wave emerging from the two ultrasound waves and attenuating much less quickly.

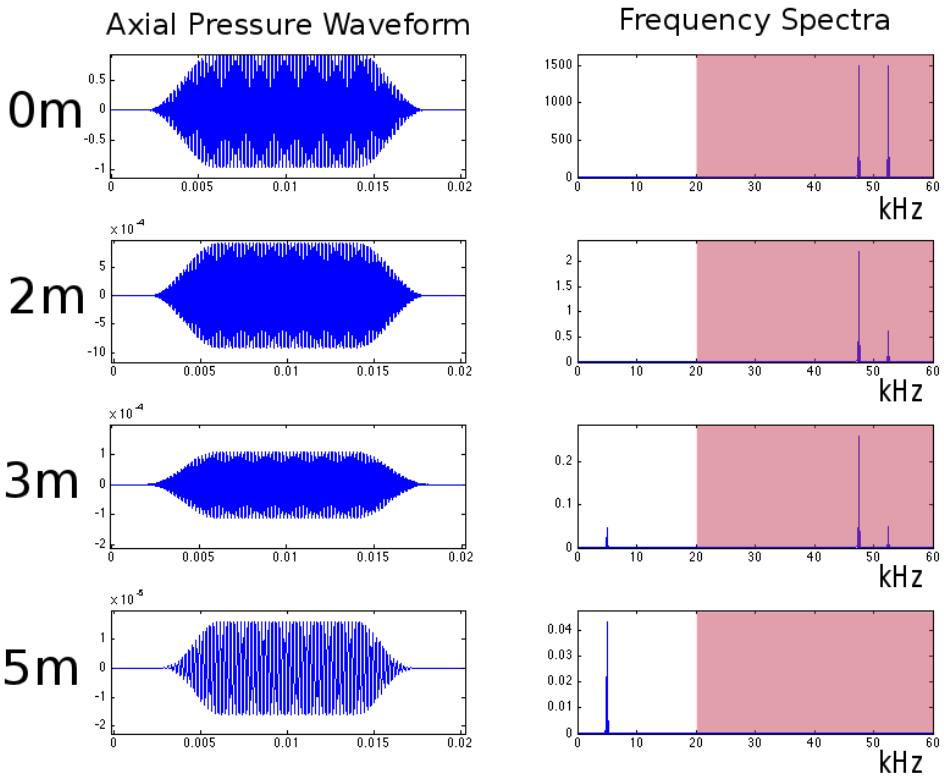


Figure 1: Two peaks chosen in the ultrasound range (here indicated in pink) attenuate relatively quickly over distance. Their difference frequency, however, remains undiminished for much longer.

Progressing from Westervelt's original equation describing a parametric array,

$$\nabla^2 p - \frac{1}{c_0^2}\frac{\partial^2 p}{\partial t^2} + \frac{\delta}{c_0^4}\frac{\partial^3 p}{\partial t^3} = -\frac{\beta}{\rho_0 c_0^4}\frac{\partial^2 p^2}{\partial t^2} \quad (1)$$

Khoklov, Zabolotskaya, and Kuznetsov made augmentations to account for the "combined effects of diffraction, absorption, and nonlinearity in directional sound beams [10]." Their resulting equation [11],

$$\frac{\partial^2 p}{\partial z \partial \tau} = \frac{c_0}{2}\nabla_\perp^2 p + \frac{\delta}{2c_0^3}\frac{\partial^3 p}{\partial \tau^3} + \frac{\beta}{2\rho_0 c_0^3}\frac{\partial^2 p^2}{\partial \tau^2} \quad (2)$$

commonly known as the KZK Equation, has been solved using a finite difference model [12], allowing for simulations of a parametric array in two dimensions. Although there are no analytical solutions, the Non-Destructive Evaluation Lab has used numerical solutions to model several applications, including robot echolocation [13], concealed weapon identification [14, 15], and simulations of bird exclusion [6].

The other primary acoustic technique analyzed in this paper, a length-limited effect, also addresses sound attenuation based on spatial dimension. While a parametric array attenuates sound outside a narrow beam, a length-limiting affect attenuates sound after the beam travels a certain distance. To generate this effect, we utilize two parametric arrays playing the same audible difference frequency; in this case, 2kHz. We then alter the phase of one of the speakers to create a parametric array with more control of the propagation distance. Since traditional instincts relating to linear acoustics are not applicable in this situation, we must rely on simulations.

In 2012, Nomura, Hedberg, and Kamakura published research [16] describing simulations of a length-limited parametric effect. The NDE Lab attempted to simulate the effect using the traditional

KZK approach, but found that the KZK simulations were not appropriate for simulating a length-limited effect (see Figure 2 below).

.

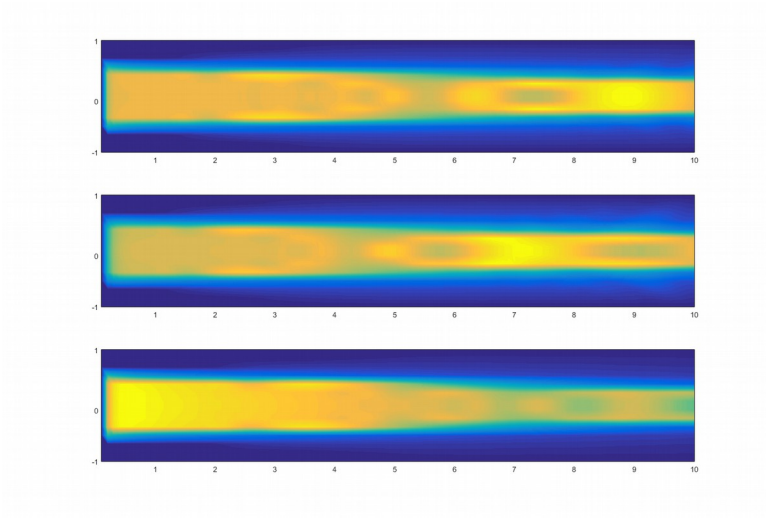


Figure 2: With no effect applied (top), beams out-of-phase by .7*π (middle), and beams out-of-phase by .9*π. Note the lack of a true length-limited effect. As the beam propagates rightward along the x-axis, the beam intensity does not attenuate completely, but rather devolves into pockets of attenuation. Image courtesy Elizabeth Skinner, Non-Destructive Evaluation Lab, College of William & Mary.

We then began developing three-dimensional, finite difference time-domain (FDTD) simulations, relying on the above-mentioned apparatus containing two parametric arrays propagating beside each other to achieve a length-limited effect. The two beams contain the same difference frequency, 2kHz in this case. However, one beam is phase-shifted, allowing the two difference frequencies to interfere constructively and destructively in a traditional linear pattern. The destructive interference causes a node of damped sound, and the beams will naturally attenuate before the beams regain constructive interference. Figure 3 below shows an example of the input beams, and Figure 4 shows the simulation's expected signal at different locations throughout a testing location. This allows us to experimentally record signals at specific locations and compare them with the expected simulation signals. These simulation programs require experimental data to benchmark accuracy. That data's collection and analysis is the focus of this project.

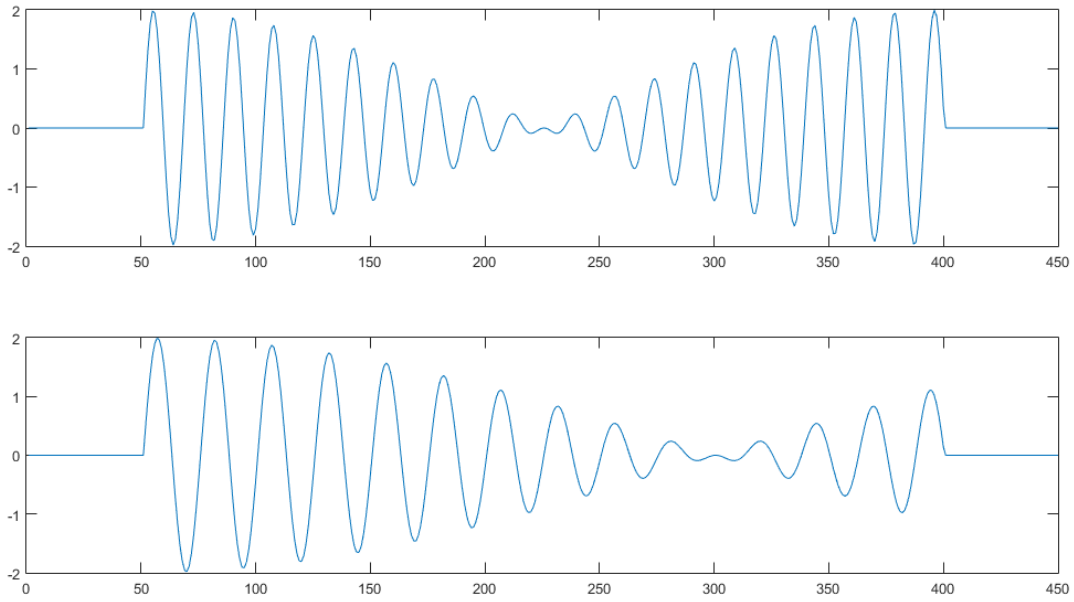


Figure 3: Top plot shows a summed wave of 39kHz and 41kHz, giving a 2kHz difference frequency; bottom plot shows the same wave with a .7*π phase offset. These each represent signal from one of the two speakers propagating simultaneously, with the end goal creating a 2kHz difference frequency parametric array that achieves a length-limited effect. The beams will interfere constructively for a certain distance, then destructively. This destructively-interfering node dampens the sound and causes a length-limiting effect because the beams will naturally attenuate before the node returns to constructive interference.

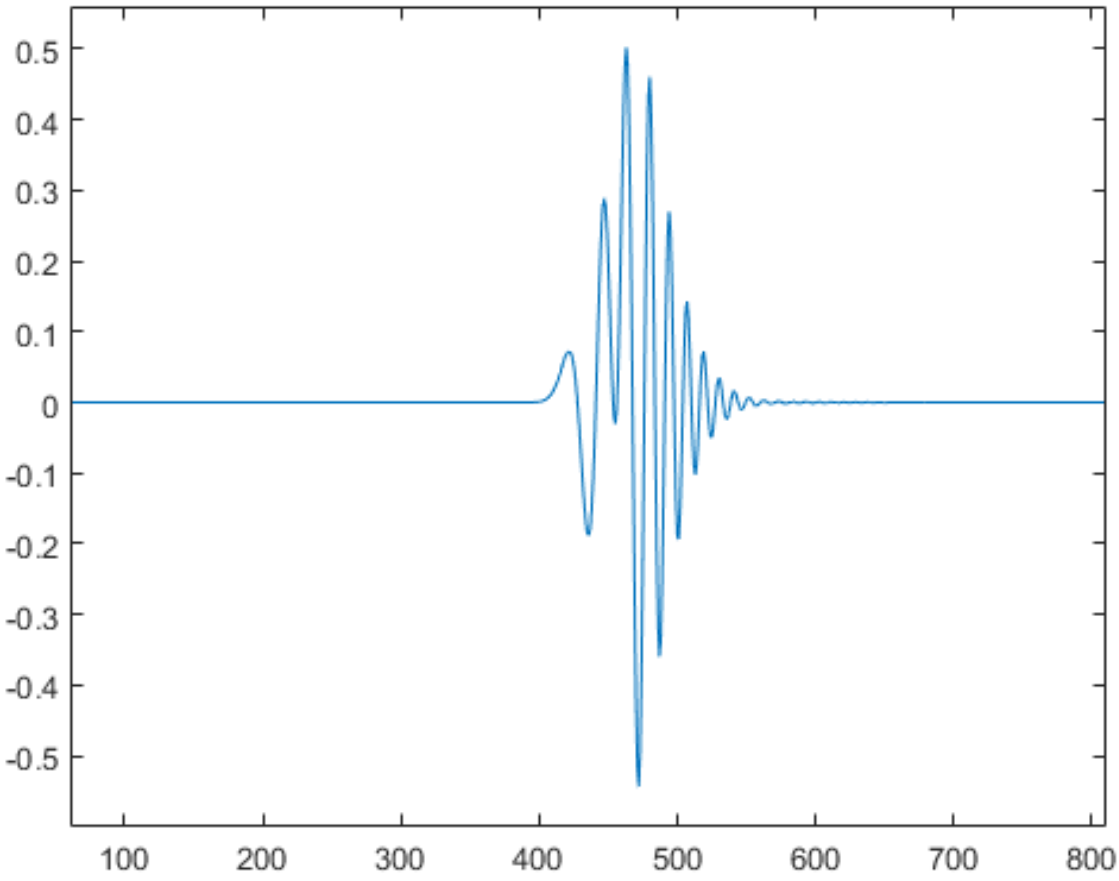


Figure 4: The expected signal recorded after the beams in Figure 3 propagate for .3 meters.

3. Method and Data Collection

We acquired data to support or improve our simulations, by recording audio over a grid to chart the sound pressure level of the length-limited effect. After examining various rooms across campus, we decided to use Small Hall 110, an amphitheater-style lecture hall. Outdoor locations proved to be too volatile with respect to background noise, and no other indoor location matched Small 110's size, relative lack of background noise, and accessibility. The room is also equipped with sound-absorbing panels on the wall help reduce reflections of the sound waves. Using H2 Zoom microphones and tripods borrowed from Swem Library's Media Center, we made a grid six rows deep and three columns wide, placing tripods on top of desks, as seen in Figures 5, 6, and 7. Note: Distance 6 is abbreviated because of a wheelchair row in the room, and we tilted the speakers to account for the room's slanted floor.

We used commercially available parametric arrays. We mounted two parametric array speakers side-by-side on a movable stand so that their beams would slightly interfere with each other, as our simulations assume.

For each of the three columns and each of the six distance points, we input the same data: the Right speaker plays a base 2kHz wave, and the Left speaker plays 2kHz wave that is phase shifted. Our phase shift is measured in 19 increments of Pi, by tenths, so that the 20^{th} shift is merely a 2*Pi shift back to the base case.

Although the frequency range we use to exclude birds is actually a “pink noise” static that amplifies specific frequency bands in the 2kHz to 10kHz range, we chose a sample tone of 2kHz for several reasons: first, simpler tones are less complicated to research, by virtue of having less “moving parts.” Second, previous simulations in the literature [26] used 2kHz, so we would like to compare results as exactly as possible.

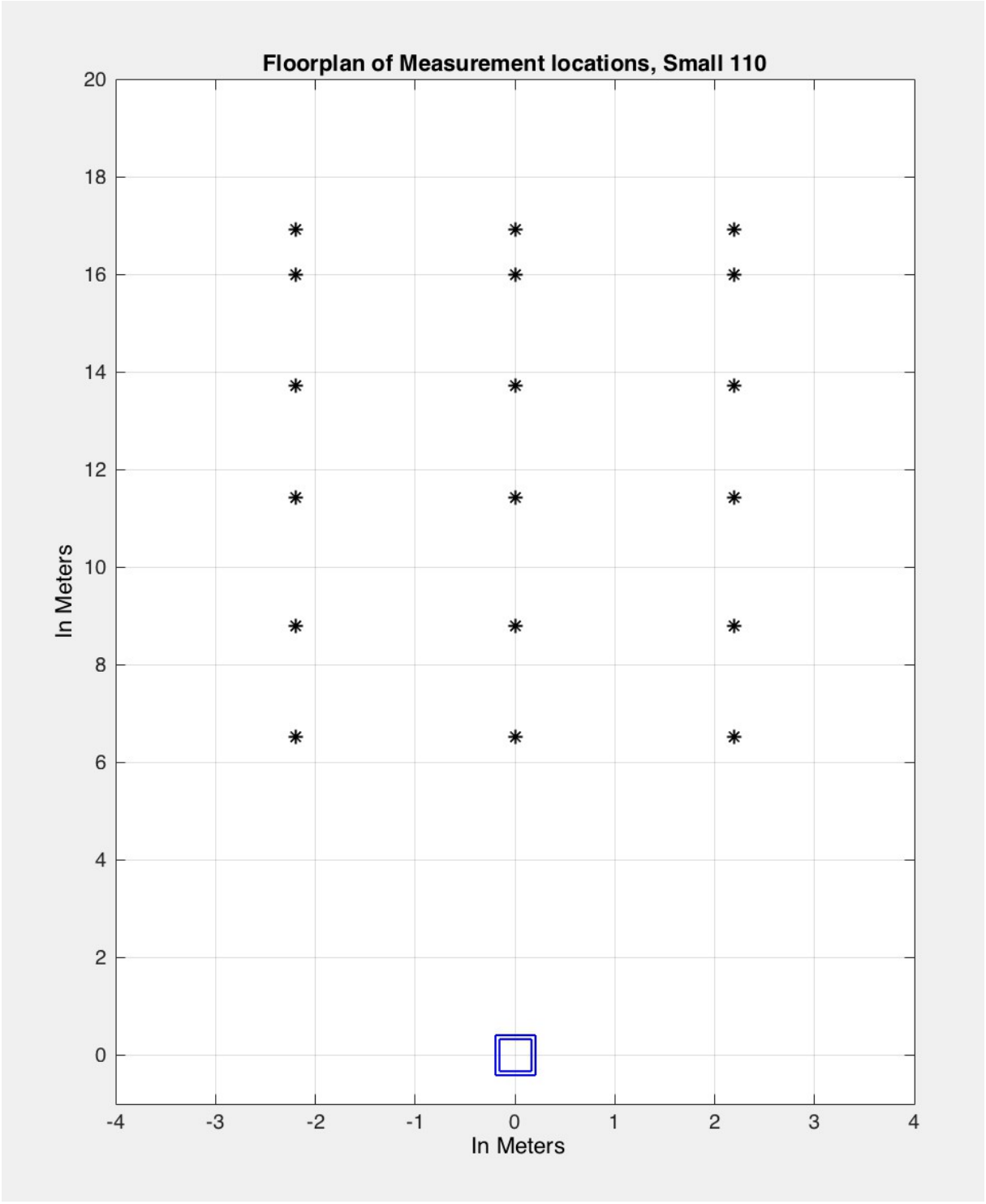


Figure 5: The blue square marks the speaker location, and the black asterisks are locations of microphones.

Figure 6: Microphones at distance 1 (or distance “a”).

Figure 7: Microphones at distance 1 (or distance “a”), Audio Spotlight speakers.

Within that basic framework, we have measured several variations to help improve our simulations. These mostly center around varying the relative audio output levels of the Right speaker's 2kHz base output and the Left speaker's 2kHz, phase-offset output. Our other chief parameter was varying the angle of overlap between the Right and Left speakers' beams. Additionally, we determined that a thorough testing of the approach should include data from two speaker models, so we performed the same audio measurements with the Audio Spotlight 24i Speakers [18] (Figure 8) and the HyperSound Virtual Reality Audio Systems [19] (Figure 9). The Audio Spotlight 24i's data and Measurement parameters are recorded in Tables 1, 2, and 3, and the HyperSound Virtual Reality Audio Systems are recorded in Tables 4 and 5. We did not repeat Measurements 7-11 with the HyperSound speakers because we deemed the 10 degree overlap to be too severe an angle. We instead adjusted the angle to 4 degrees and repeated the parameters for both speakers in Measurements 12-16 of the Audio Spotlight speakers and Measurements 25-29 of the HyperSound speakers. Table 6 shows which Measurements shared angle and audio parameters across both speakers.

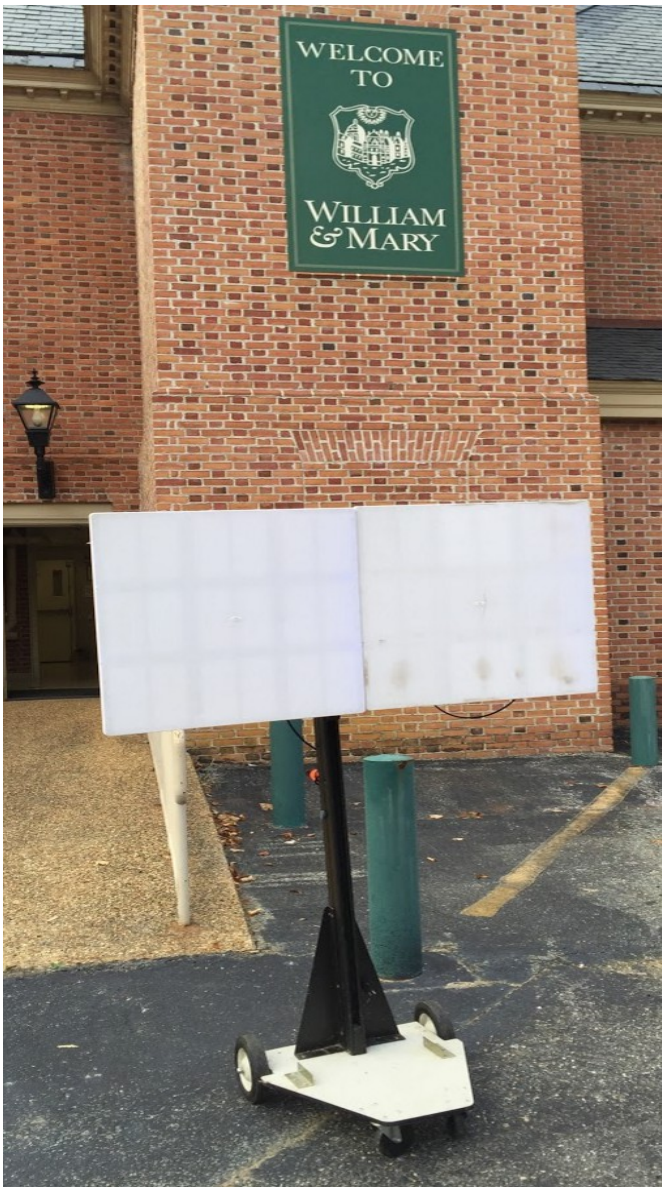


Figure 8: Audio Spotlight 24i Speakers

Figure 9: HyperSound Virtual Reality Audio Systems Speakers

Table 1: Audio Spotlight 24i Speakers, angle = 0 degrees

Measurement	Audacity Volume	2kHz-0 Track Gain	2kHz-X Track Gain
1	.4	0dB	0dB
2	.56	0dB	0dB
3	.6	-10dB	0dB
4	.7	-20dB	0dB
5	.6	0dB	-10dB
6	.7	0dB	-20dB

Table 2: Audio Spotlight 24i Speakers, angle = 10 degrees

Measurement	Audacity Volume	2kHz-0 Track Gain	2kHz-X Track Gain
7	.56	0dB	0dB
8	.6	-10dB	0dB
9	.7	-20dB	0dB
10	.6	0db	-10dB
11	.7	0dB	-20dB

Table 3: Audio Spotlight 24i Speakers, angle = 4 degrees

Measurement	Audacity Volume	2kHz-0 Track Gain	2kHz-X Track Gain
12	.56	0dB	0dB
13	.6	-10dB	0dB
14	.7	-20dB	0dB
15	.6	0dB	-10dB
16	.7	0dB	-20dB

Table 4: HyperSound Virtual Reality Audio Systems Speakers, angle = 0 degrees

Measurement	Audacity Volume	2kHz-0 Track Gain	2kHz-X Track Gain
17 (**failed)**	.05	0dB	0dB
18 (**failed)**	.09	-10dB	0dB
19 (**failed)**	.19	-20dB	0dB
20	.15	-12dB	-12dB
21	.16	-22dB	-12dB
22	.19	-32dB	-12dB
23	.16	-12dB	-22dB
24	.19	-12dB	-32dB

Table 5: HyperSound Virtual Reality Audio Systems Speakers, angle = 4 degrees

Measurement	Audacity Volume	2kHz-0 Track Gain	2kHz-X Track Gain
25	.15	-12dB	-12dB
26	.16	-22dB	-12dB
27	.19	-32dB	-12dB
28	.16	-12dB	-22dB
29	.19	-12dB	-32dB

Table 6: Measurements with comparable parameters

Angle	Measurement	
	Audio Spotlight 24i	HyperSound Virtual Reality Audio Systems
0 degrees	1	**-**
	2	**20**
	3	**21**
	4	**22**
	5	**23**
	6	**24**
10 degrees	7	-
	8	-
	9	-
	10	-
	11	-
4 degrees	**12**	**25**
	13	**26**
	14	**27**
	15	**28**
	16	**29**

Note: **Bolded** measurements were taken with the second technique to allow use of an algorithm in their processing. The algorithm required a manual entry of the timestamp for the file's first recording, which are listed in Appendix I below.

We faced several complications concerning parameters that shifted relative audio levels. Because the base speaker (Right) and the changing speaker (Left) have been defined for the duration of the testings, we expected that damping the Right speaker by 10dB would have the same effect as damping the Left speaker by 10dB. This was not the case, so we repeated all such measurements featuring different audio levels with the damped speaker switched (compare Measurements 3 and 5 in Table 1).

Similarly, our Audio Spotlight speakers were identical models, but one had been used in an outdoor experiment for several weeks. Although we initially believed this would not make a significant impact, the slightly damaged speaker (see Figure 8, right speaker) did produce different sound levels than its new counterpart. To mitigate this, we repeated Measurements with speaker-specific parameters in new Measurements with the speakers switched. (See Table 1 for the parameters in Measurements 3 and 4 being repeated on the other speaker in Measurements 5 and 6.)

Additionally, we initially applied the exact same parameters to the HyperSound speakers used in Measurements 17-29 as we had for the Audio Spotlight speakers in Measurements 1-17. This proved to be a mistake, as the HyperSound speakers contained an internal volume control that was slightly shifted from the ingrained volume of the Audio Spotlight speakers. Accordingly, we adjusted the Audacity volume and track gain parameters to create the maximum similarity between comparable Measurements (see Table 9 for comparable Measurements, and Tables 1-5 for the Measurements' individual parameters.) Measurements 17-19 were performed before this adjustment and deemed unusable.

4. Data Analysis

After recordings, we download the data from Swem's microphones and store the files in multiple places for safekeeping. With three widths, six distances, and twenty total phases, a full Measurement includes 360 files. Early in the recording process, we recorded all distances of a given Measurement within one sound file. Because each of the 20 phases recorded for each Measurement at each distance are one minute long, this total file amounted to just over two hours. Over time, we transitioned to recording multiple Measurements at a single distance, which decreased the amount of time spent adjusting microphones and increased the efficiency of the data collection process. This also meant that our recordings were just over twenty minutes instead of two hours, so our files became more manageable and less content was in jeopardy if one recording did not save.

After recording the large files, we parceled them into the individual one-minute samples for analysis. We originally clipped the files in Audacity, but the selection is based on highlighting a segment of the audio wave by hand and requires sound clips to end exactly on second integers. We transitioned to Matlab, which allows precision to the millisecond level.

Performing shorter-length recordings also allowed a significant improvement in data parceling. Before Measurement 12, we announced a verbal ID before each individual one-minute sample. Beginning with Measurement 12, we announced a verbal ID before each distance per Measurement, and played all twenty phases back-to-back of a single Measurement and distance. This allowed for a constant gap between phases for all Measurements and distances. We took advantage of this to write an algorithm to begin at a specific point and clip the rest of the twenty-minute file. The algorithm, exampled by Export14.m in Appendix II, required start points for each Measurement and distance, collected manually and listed in Appendix I. This new approach removed a potential source of human error in file clipping, drastically reduced the amount of time required to process files, and allowed the

researcher breaks of twenty minutes between verbal IDs, instead of one minute.

We are now analyzing the data to optimize our parameters. Of the 29 Measurements, we are most focused on Measurements 2-6, 12-16, and 20-29. We deemed Measurement 1 to be too quiet, and Measurements 7-11 to be too severely angled towards each other. The twenty remaining measurements are split into 7,200 one-minute files, which we will use to determine which Measurement, and which phase offset within that Measurement, most strongly displays a length-limiting effect.

Initial filters in Audacity confirmed expectations for the nature of our recordings. Unfiltered files (Figures 10 and 12) show the background noise of the room in Small 110, mostly attributed to the air circulation system. A rudimentary High and Low pass filter, cutting out 24dB per octave below 1800Hz and above 2200Hz, clearly shows the remaining 2kHz peak we are focused on, as seen in Figures 11 and 13.

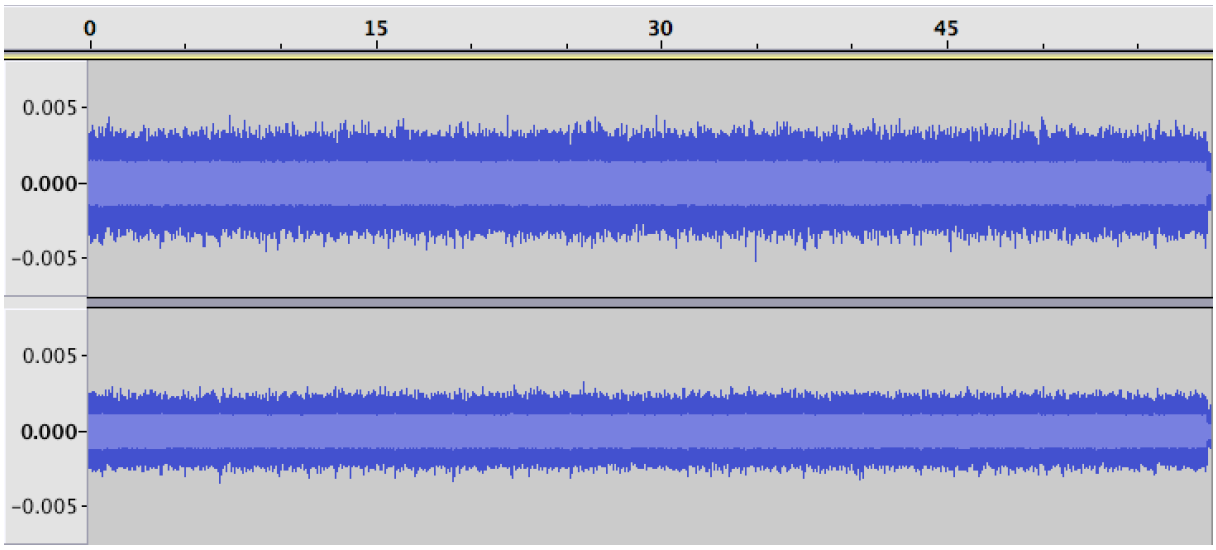


Figure 10: Unfiltered file, Measurement 2, Middle Column, distance 3, phase-offset 7

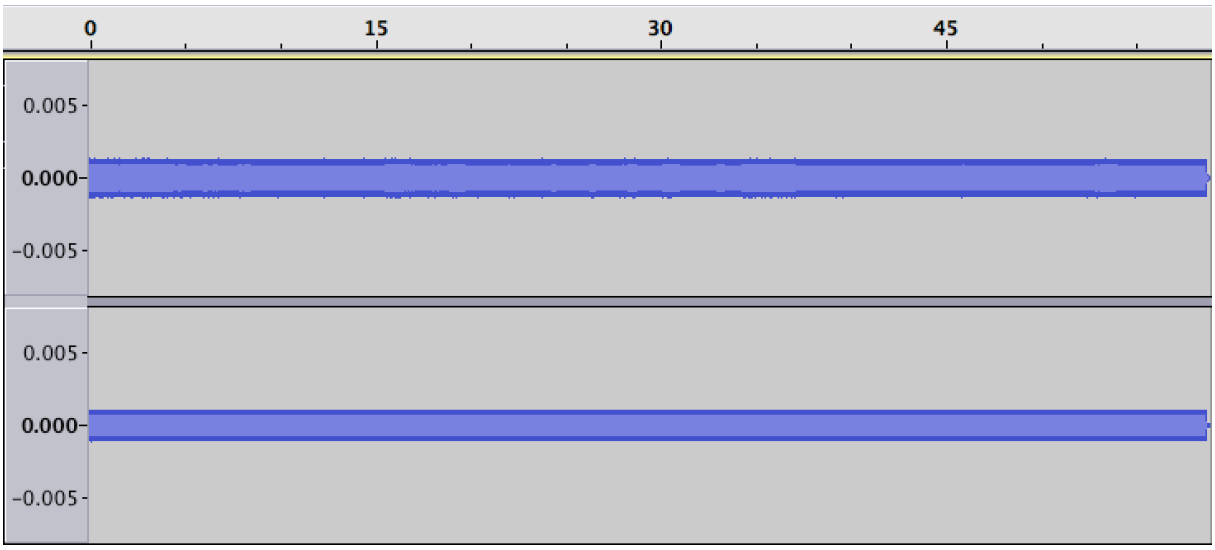


Figure 11: Filtered file, Measurement 2, Middle Column, distance 3, phase-offset 7. The zoom between Figures 4 and 5 is equal, notice the greater signal uniformity from the filter in Figure 5.

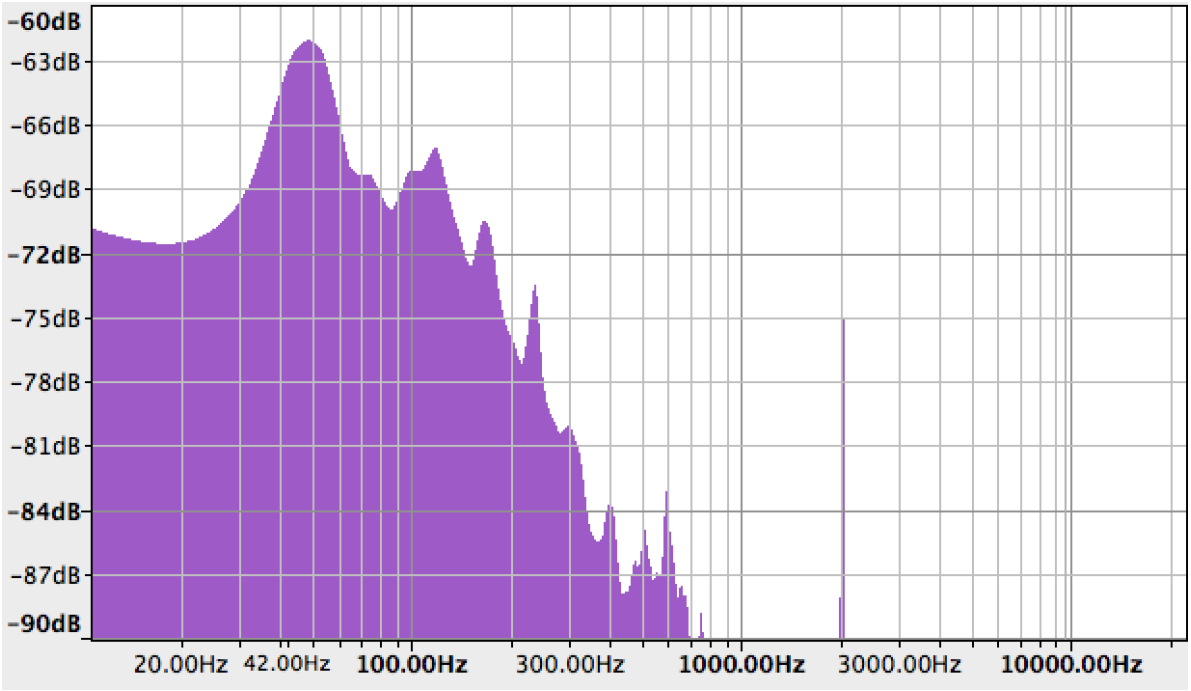


Figure 12: Fourier transform of the unfiltered file in Figure 4. All of the signal to the left of the 2kHz spike is background noise in the room, mostly the air-conditioning unit.

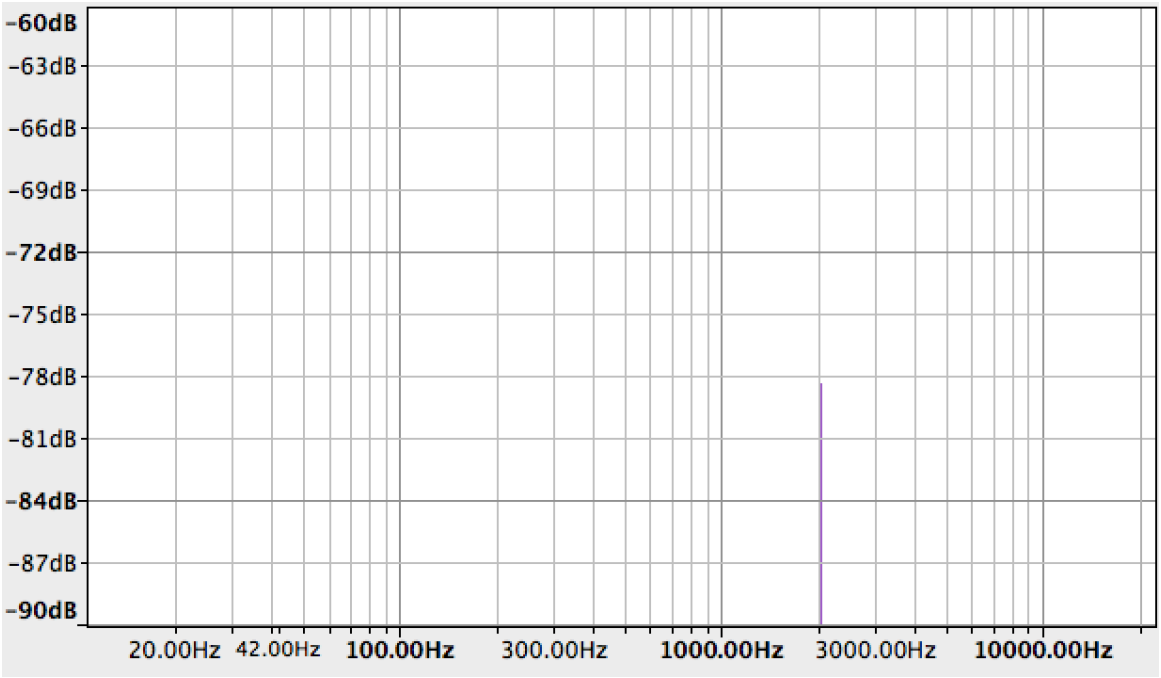


Figure 13: Fourier transform of the filtered file in Figure 5.

By direct observation, it is clear that our length-limiting attempts are successful on some level. We know from visual observation of file amplitudes that phase offsets between 6 and 11 display a stronger effect than phase offsets of 0 or 19 (see Figures 14), regardless of Measurement parameters. Additionally, the human ear can discern the differences in the effect between phase offset 7 and phase offset 19 at Distance 6. For a more precise analysis, we turn to Matlab's fast Fourier transform (FFT) function, which breaks individual files into their frequency components and weights each frequency with a coefficient. The 2kHz coefficient value provides an internally consistent variable for an empirical comparison of the strength of signal for each Measurement and phase. The process is somewhat illuminated in Appendix II.

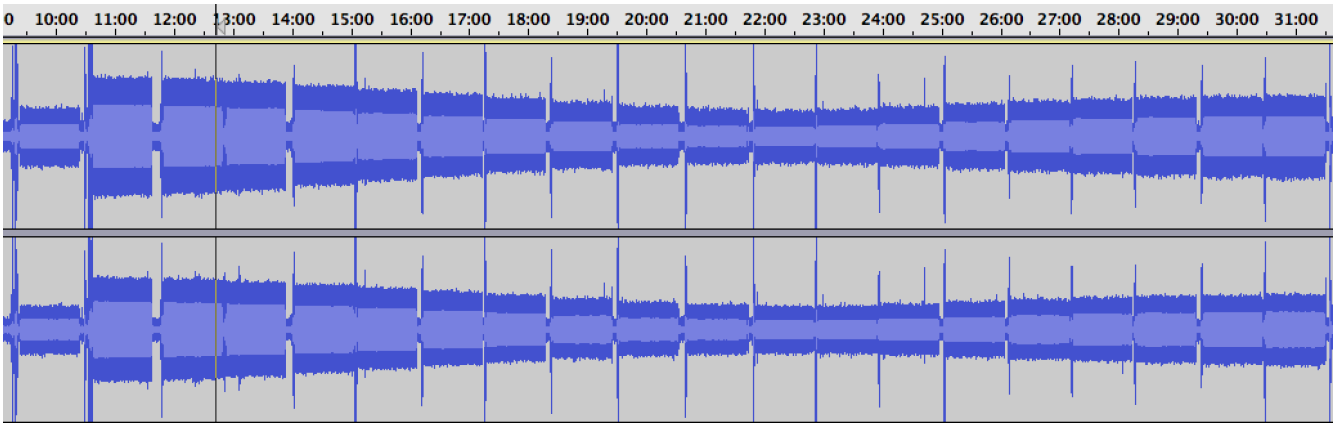


Figure 14: Measurement 4, Middle Column, distance A. Notice the constructive interference on the left side of the plot, near the 11-minute mark, when the phase offset between the two input waves is 0*π. As the effect maximizes at the 22-minute mark when the phase offset nears 1*π, the destructive interference becomes more apparent. As the phase offset approaches 2*π, the interference becomes constructive again.

Processing for Measurements 12-16 and 20-29 functioned smoothly. For much of the signal in Measurements 2-6 and 12-16, a lower, 40Hz noise overshadowed the 2kHz spike, presumably from the air conditioning system. We processed those files using a bandpass filter to isolate the 2kHz peak and continued with identical analysis techniques.

Appendix III contains final plots for all of the Measurements. In most cases, the Right and Left column data lacks structure. We believe these are a result of the beam scattering off of walls and re-entering its original path. The Middle column shows a much stronger pattern of attenuation as the effect is applied; as highlighted below, Middle column plots for Measurements 20 (Figure 15) and 25 (Figure 16) demonstrate sharp attenuation at phases 9 and 10, especially when compared with confusing (Figure 17) or noisy (Figure 18) plots from the Right or Left columns.

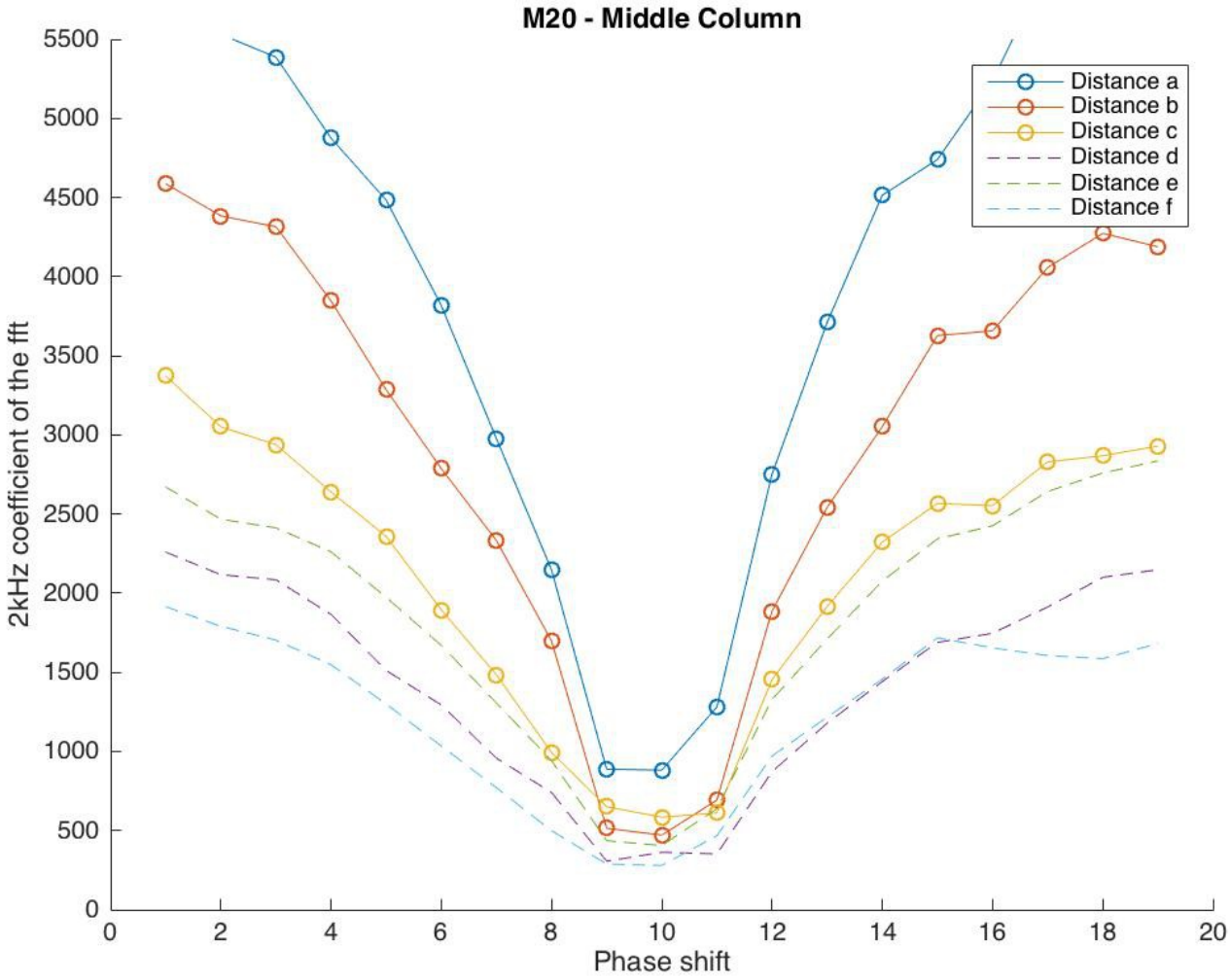


Figure 15: As the destructive interference reaches a maximum at the phase shift 1*π, coefficients for 2kHz drop significantly for each distance.

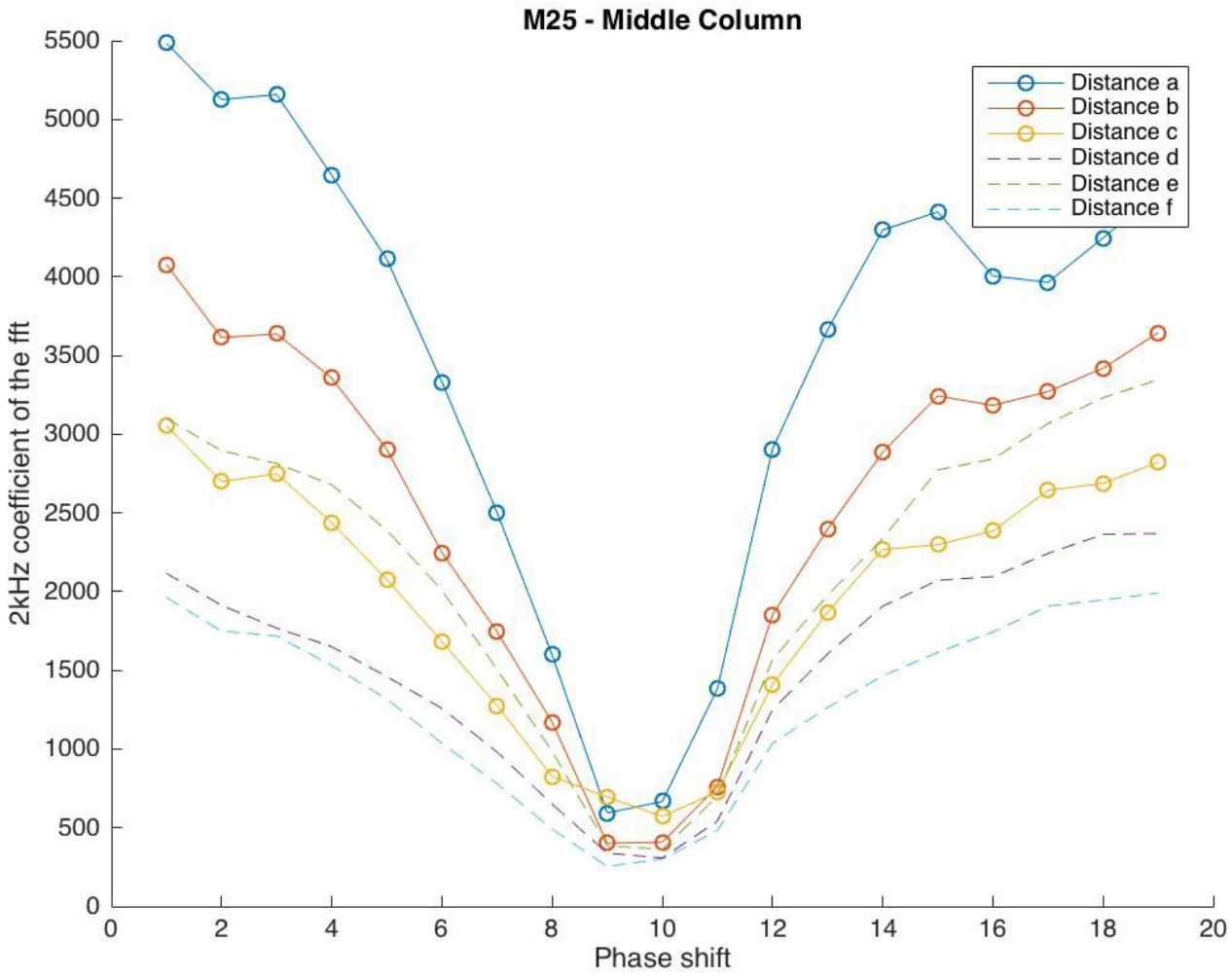


Figure 16: As the destructive interference reaches a maximum at the phase shift 1*π, coefficients for 2kHz drop significantly for each distance.

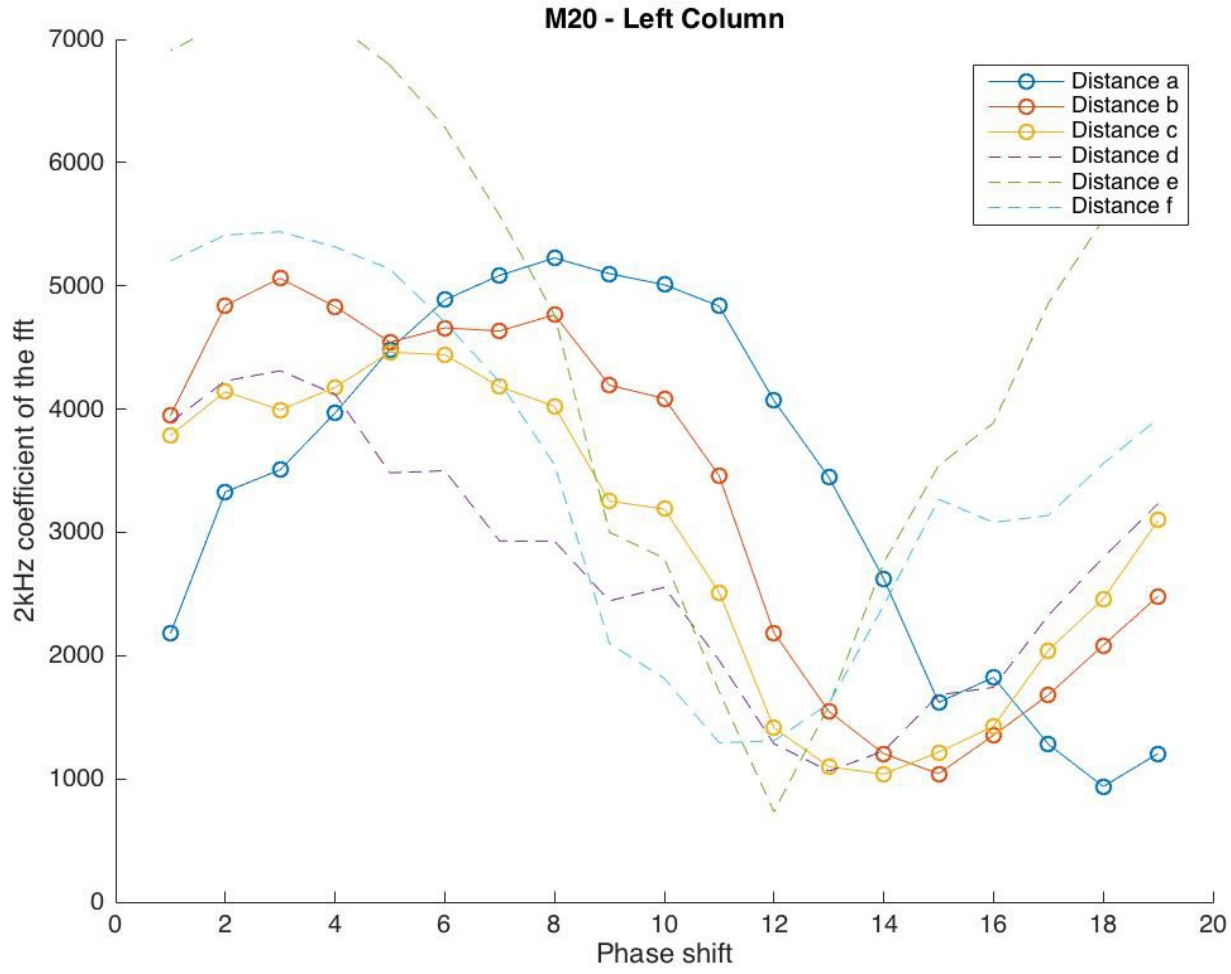


Figure 17: Although this plot demonstrates a trough that is somewhat similar to those seen in Figures 15 and 16, the rises in lower phase shifts is unexpected. We presume this is the result of reflections.

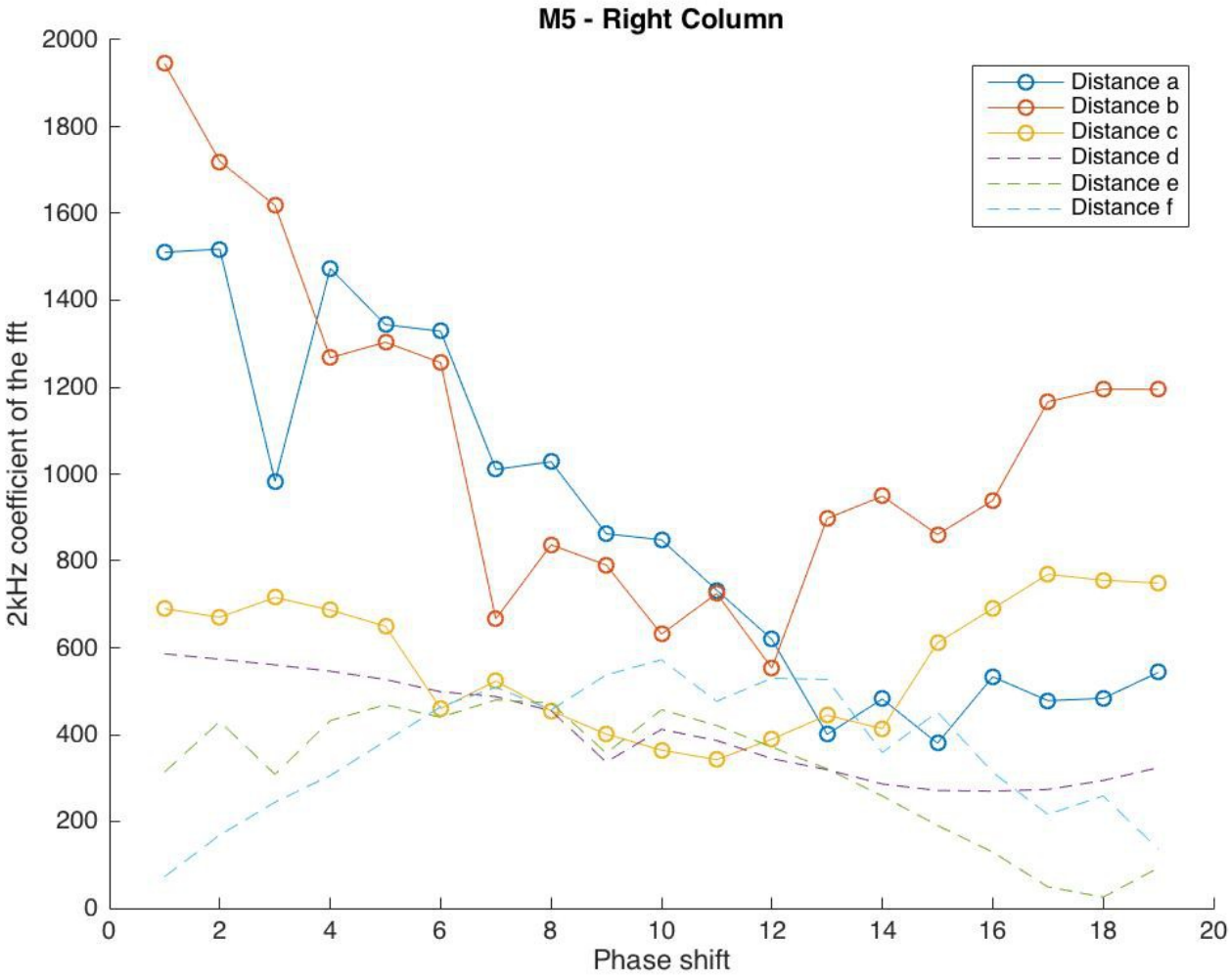


Figure 18: This plot only vaguely displays a trough, and its various distances show no cohesive pattern. This is typical of plots from the Right and Left columns, indicating reflections or other noise.

The most dramatic example occurs when comparing Middle column measurements for the pairs of Measurement 20 (Figure 19) and 25 (Figure 20), and especially Measurements 2 (Figure 21) and 12 (Figure 22); these Measurements share comparable parameters, as shown in Table 6, and did not have relative volume differences between speakers. Our simulations predicted that the Audio Spotlight 24i (Meaurements 2-6 and 12-16) would demonstrate a greater length-limited effect than the HyperSound system (Measurements 20-29) because of its superior parametric array effect. Additionally, our simulations suggested that angling the speakers slightly towards each other to increase the beams' overlapping area would increase the effect (Measurements 12-16 and 25-29). Both of these hypotheses are supported by our data. Measurement 20's effect appears to be much stronger than Measurement 2; this may be due to the lack of angled speakers. In this situation, the Audio Spotlight speakers' superior parametric array ay be a hindering effect, because the beams do not sufficiently. The HyperSound speakers, with a less narrow parametric array, shows a strong pattern.

The comparison in Measurements 12 and 25 is even more illuminative. Measurement 25's plot appears to be very similar to that of Measurement 20 as expected, because the two Measurements' only parameter difference is the angled speakers. However, Measurement 12's plot shows the clearest yet example of a length-limited beam. For distances a, b, and c, applying the length-limiting effect does not dramatically change the strength of the 2kHz signal. However, at distance d, and particularly distances e and f, the signal attenuates drastically. This is clear evidence of a length-limited effect, with the attenuating distance lying between distances c and d (roughly 12 meters distance from the speakers).

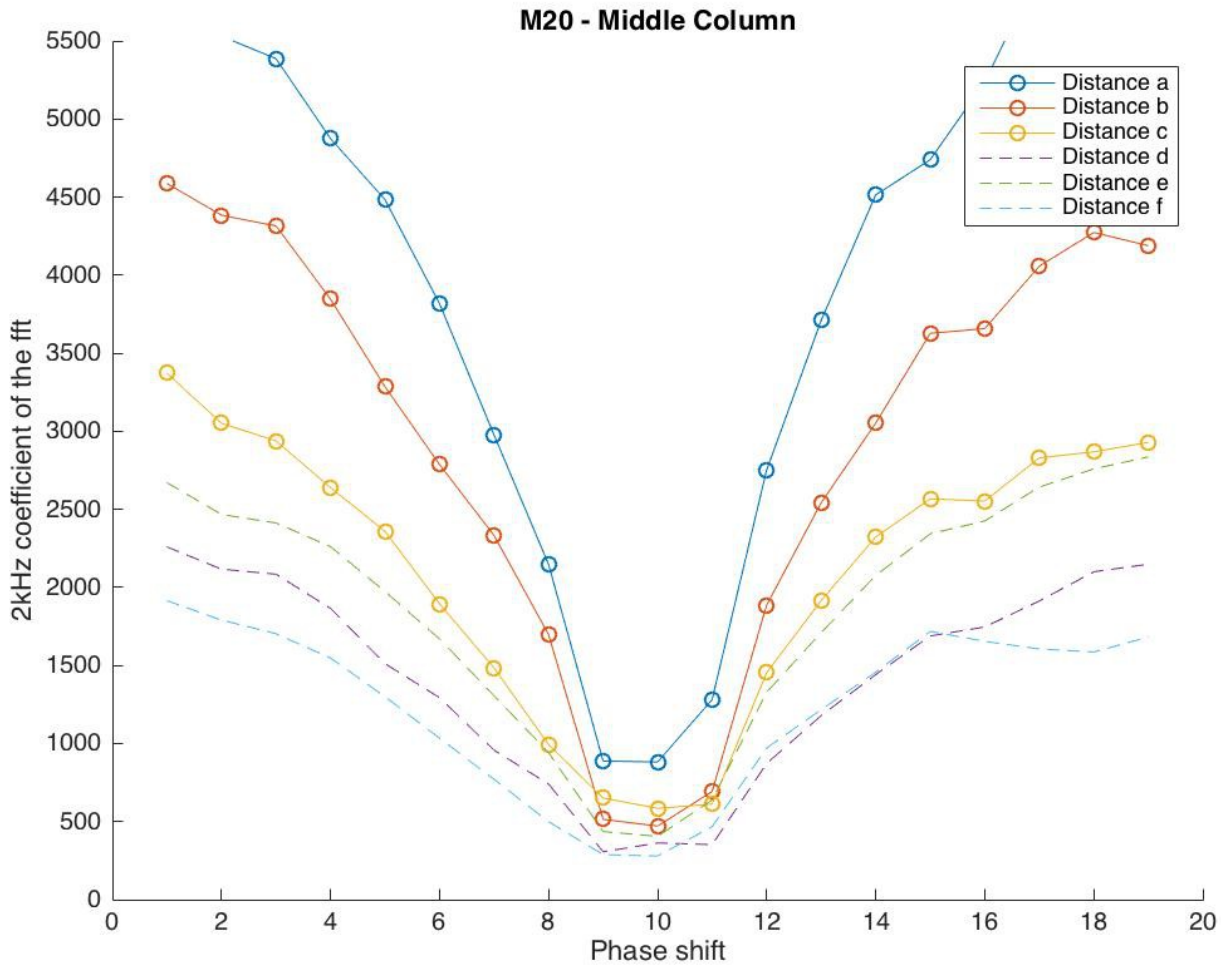


Figure 19: This plot of Measurement 20's Middle column results shows the HyperSound speakers at a 0° angle to each other. Because these speakers have a relatively inferior parametric array effect, we presume their beams did not require angling to intersect; therefore, they show strong destructive interference patterns.

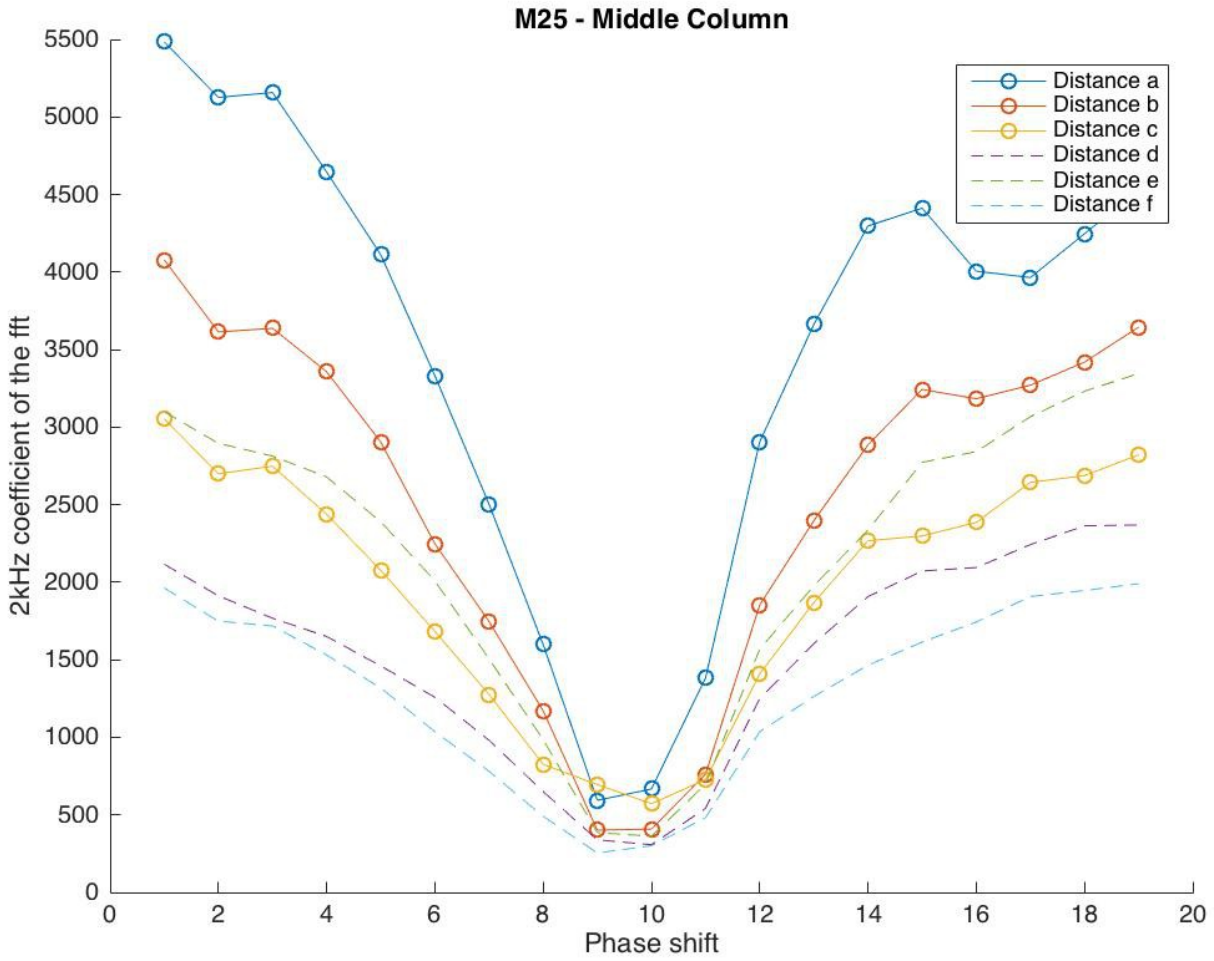


Figure 20: This plot's parameters are the same as those of Figure 19, with the sole difference of a 4° angle between the two speakers. As might be expected, the plots are almost identical because the HyperSound speakers have a less-significant parametric array effect; a 0° and 4° angle difference makes a comparatively small impact because the beams are relatively wide.

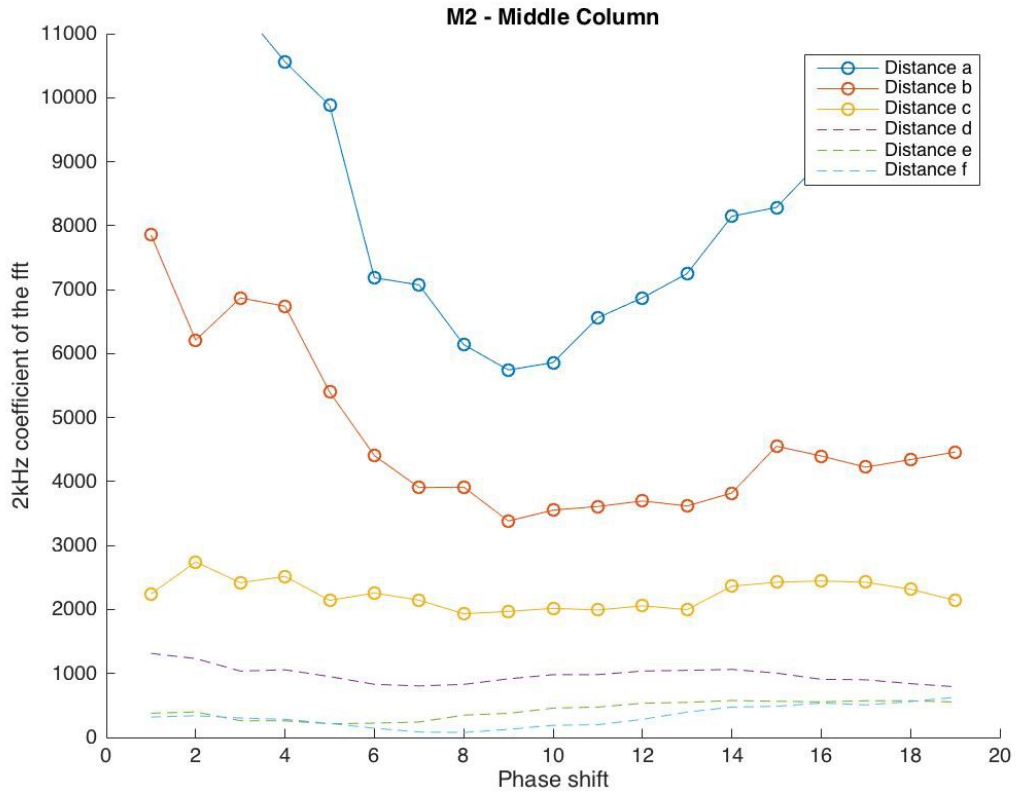


Figure 21: This Measurement 2, Middle column result is most helpful in contrast with the following figure. Although the closer distances show somewhat of a trough pattern, the farther distances show no change in 2kHz coefficient relative to phase shift; this indicates the two beams are not interfering with each other. This is not surprising, given the greater parametric array effect of the Audio Spotlight speakers and the 0° angle between the speakers. The beams are essentially parallel to each other. Notice the shift in the following plot, when the angle between speakers is increased to 4°.

Note: point 16 for distance b registered as an unreasonably low coefficient (86.2) and was deemed an outlier and error. It was replaced by an average of points 15 and 17.

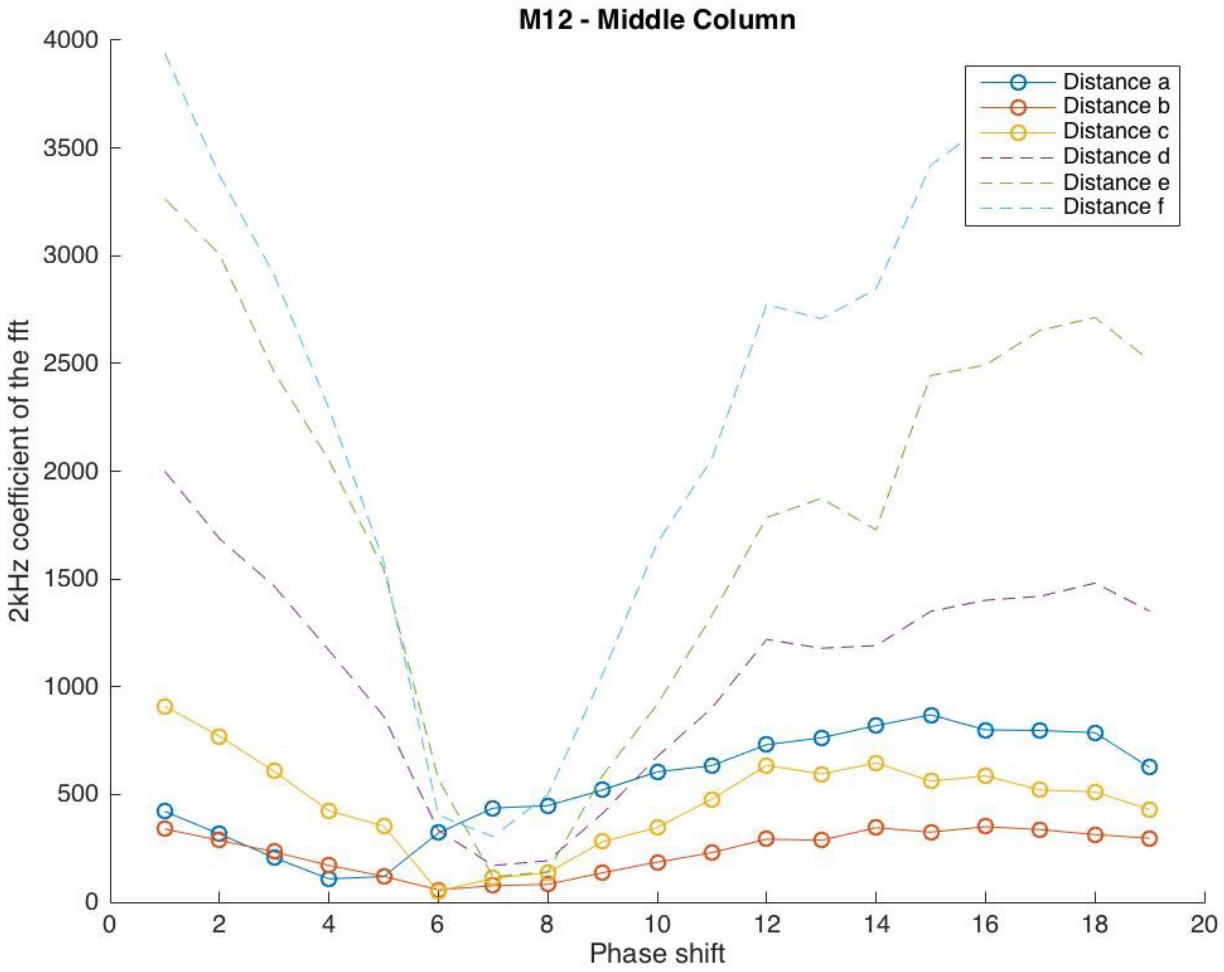


Figure 22: This plot of Measurement 12's Middle column, with a 4° angle between speakers, is an excellent example of what a viewer might expect to see in a length-limited sound beam. The distances closest to the speaker are relatively unaffected by the application of the effect, as the phase offsets travel to 1*π and then 2*π. However, beginning with distance d (approximately 10m from the speakers), the sound level is strongly affected by the phase shift. Besides indicating a length-limited effect, this confirms our opinion that the Audio Spotlight speakers are superior to the HyperSound speakers. A 0° or 4° angle made no difference to the HyperSound speakers because their beam is wide enough to allow interference without an angle. The Audio Spotlight speakers emit a narrow beam that must be angled to ensure overlap and interference.

5. Conclusions

In summary, our simulation's predictions that a greater parametric array capability in the Audio Spotlight speakers would grant them a greater length-limited effect than the HyperSound system was supported by the experimental data, as was the approach of angling the speakers slightly inward to maximize beam interference. Of our varied parameters, those which played the Right channel and Left channel speakers at equal volume (Measurements 2, 12, 20, and 25) proved more successful than those with varied volume intensities (Measurements 3-6, 13-16, 21-24, and 26-29). Of all our parameters tested, Measurement 12 displays the most promise for both the future of our application to bird deterrence and the broader field of length-limited parametric arrays. Besides the parameters concerning relative audio levels between speakers, these results are completely expected. The destructive interference relies on beam overlap, and the Audio Spotlight beams display a greater parametric array effect and cannot overlap unless angled towards each other. This also explains why the HyperSound results do not change significantly when recorded at 0° or 4°; their beams are wide enough to interfere with each other regardless of a 4° angle difference.

The lack of a definitive Figure for the HyperSound speakers should not be interpreted as a sign that they are incapable of producing a length-limited effect. Rather, they may indicate that the attenuation distance is less than 7 meters, below which we did not collect data. Indeed, distances d-f in Figure 22 closely resemble all distances in figures 19 and 20. This could imply that the distances at which the phase shift line is relatively horizontal are merely closer than we measured. This assumption is also reasonable considering the relatively lower size of the HyperSound speakers.

6. References

[1] Westervelt, Peter J. “The Parametric Acoustic Array.” *The Journal of the Acoustical Society of America*. Volume 35, number 4. April 1963.

[2] Bennett, Mary Beth, and David T. Blackstock. “Parametric Array in Air.” *The Journal of the Acoustical Society of America*. Volume 57, number 3. March 1975.

[3] D. Pimentel, et. al. “Environmental and economic costs of nonindigenous species in the United States.” *Bioscience.* Volume 50, number 1. 2000. pp. 53-65.

[4] J. Allan. *The costs of bird strikes and bird strike prevention*. in *Human Conflicts with Wildlife: Economic Considerations.* Proceedings of the Third NWRC Special Symposium, National Wildlife Research Center, Fort Collins, CO 2002

[5] R.A. Dolbeer, S.E. Wright, J. Weller, and M.J. Begier. *Wildlife strikes to civil aircraft in the United States (1990-2011)*, F.A.A. U.S. Department of Transportation, ed., Office of Airport Safety and Standards Serial Report No. 18 2012

[6] Eric. A. Dieckman, Elizabeth Skinner, Ghazi Mahjoub, John Swaddle, and Mark *Hinders. Benign exclusion of birds using acoustic parametric* arrays. Proceedings of Meetings on Acoustics, Acoustical Society of America, Volume 19. 2013.

[7] Swaddle, John P., Dana L. Moseley, Mark K. Hinders, and Elizabeth P. Smith. “A sonic net excludes birds from an airfield: implications for reducing bird strike and crop losses.” *Ecological Applications.* Volume 26, number 2. 2016. p340.

[8] Mahjoub, Ghazi, Mark K. Hinders, and John P. Swaddle. “Using a “Sonic Net” to Deter Pest Bird Species: Excluding European Starlings from Food Sources by Disrupting Their Acoustic Communication.” *Wildlife Society Bulletin.* Volume 39, number 2. April 2015. pp.326-333.

[9] Swaddle, John P., Dana L. Moseley, Mark K. Hinders, and Elizabeth P. Smith. “A sonic net excludes birds from an airfield: implications for reducing bird strike and crop losses.” *Ecological Applications.* Volume 26, number 2, 2016. p342.

[10] Hamilton, Mark F. and David T. Blackstock. *Nonlinear Acoustics.* Academic Press (Elsevier), Boston. 1997. pp. 55-62.

[11] Zabolotskaya, E. A., and R. V. Khokhlov. "Quasi-plane waves in the nonlinear acoustics of confined beams." *Sov. Phys. Acoust* Volume 15, number 1. 1969. pp. 35-40.

[12] Lee, Yang-Sub and Mark F. Hamilton. “Time-domain modeling of pulsed finite-amplitude sound beams.” *Journal of the Acoustical Society of America.* Volume 97, number 2. February 1995.

[13] Dieckman, Eric A. *Use of Pattern Classification Algorithms to Interpret Passive and Active Data*

Streams from a Walking-Speed Robotic Sensor Program. Diss. The College of William & Mary, May 2014.

[14] Rudd, Kevin and Mark Hinders. “Simulation of Incident Nonlinear Sound Beam and 3D Scattering from Complex Targets.” *Journal of Computational Acoustics.* Volume 16, number 3. 2008. p427–445,

[15] Rudd, Kevin and Mark Hinders. “Acoustic Parametric Array for Identifying Standoff targets.” *Review of Nondestructive Evaluation.* Volume 29. 2010.

[16] Nomura, Hideyuki, Claes M. Hedberg, and Tomoo Kamakura. “Numerical simulation of parametric sound generation and its application to length-limited sound beam.” *Applied Acoustics*. Volume 73, issue 12. December 2012.

[17] Bhan Lam, Wong-Seng Gan, and Chuang Shi. *Feasibility of a Length-Limited Parametric Source for Active Noise Control Applications.* Proceedings of the 21st International Congress on Sound and Vibration, Beijing, China, 13-17 July 2014

[18] “Audio Spotlight 24i,” sold by Holosonics. https://holosonics.com/home/11-audio-spotlight-24i.html

[19] “Virtual Reality Audio System,” sold by HyperSound. http://hypersound.com/products.php

Appendix I

Starting points (in seconds) for the second batch of measurements (M12-M16 and M20-M29), utilizing an algorithm as discussed in the Data Analysis section
Columns: Right, Middle, Left
Distances: A, B, C, D, E, F

* indicates noise, in which the maximum peak was not 2kHz. More **** indicates more noise, Matlab bandpass filtering applied to isolate the 2kHz peak.

Measurement 12*****

	A	B	C	D	E	F
Right	30.7	36.54	26.4	46.15	68.32	1074
Middle	25.27	31	33.24	38.6	62.99	1086.2
Left	19.82	25.75	39.53	30.98	51.65	1092

Measurement 13*****

	A	B	C	D	E	F
Right	50.7	83.25	75.25	61.71	76.38	110
Middle	61.18	89.65	68.86	55.62	68.7	116.6
Left	67.49	94.53	63.26	49.23	60.86	122.9

Measurement 14*****

	A	B	C	D	E	F
Right	33.31	59.14	67.62	75.49	90.54	74.94
Middle	39.27	50.57	61.94	65.33	80.92	65.6
Left	46.11	41.18	50.71	56.17	72.21	56.44

Measurement 15*****

	A	B	C	D	E	F
Right	139.2	66.09	128.7	98.07	81.41	76.41
Middle	34.13	56.75	121.9	86.91	70.63	66.12
Left	77.98	48.21	111.9	79.52	60.67	58.12

Measurement 16*****

	A	B	C	D	E	F
Right	72.25	61.53	68.02	80.45	77.39	64.4
Middle	48.24	53.62	61.24	71.14	66.83	55.27
Left	39.77	46.15	54.71	62.66	57.21	47.9

Measurement 20

	A	B	C	D	E	F
Right	205.3	77.56	39.94	109.7	85.29	65.37
Middle	199.2	72.09	56.1*	102.6	79.59	72.7
Left	188.3	67.19	48.78**	96.84	72.9	79.6*

Measurement 21

	A	B	C	D	E	F
Right	87.29**	90.2	70.63	83.44	104.2	87.49
Middle	79.54	82.83	62.73	75.65	98.04	78.87
Left	69.21	69.91	55.22	66.44	92.77	70.13

Measurement 22

	A	B	C	D	E	F
Right	63.48	67.15	87.19	108.4	80.45	113.5
Middle	70.08	68.69	97	102.8	72.64	105.4
Left	75.6)	52.28	103.7	92.47)	64.93	97.12

Measurement 23

	A	B	C	D	E	F
Right	87.47	68.95	76.71	89.42	111.4	82.34
Middle	80.55	60.87	68.39	81.69	103	88.58
Left	72.76	55.05	61.48	73.83	95.57	95.84

Measurement 24

	A	B	C	D	E	F
Right	61.56	62.49	72.18	123	83.17	72.9
Middle	52.58	54.48	63.28	115	74.63	64.01
Left	45.13	46.56	54.78	107.1	67.04	56.27

Measurement 25

	A	B	C	D	E	F
Right	34.57**	62.03	51.29	57.9	84.69	46.59
Middle	40.38	67.29	56.94	52.15*	91.16*	51.81*
Left	46.66**	72.43**	62.96*	47.54*	99.81	57.13

Measurement 26

	A	B	C	D	E	F
Right	55.58***	62.51**	65.03**	84.8	78.98	47.39
Middle	47.96	55.27	58.25	76.73	71.56	61.65
Left	40.4***	48.07	50.83	69.63	64.21	53.7

Measurement 27

	A	B	C	D	E	F
Right	63.4***	71.51	82.26	62.85	84.25	71.77
Middle	56.29	63.83	73.95	58.25	76.86	62.34
Left	49.91	57.56	67.78	49.09	69.93	55.24

Measurement 28

	A	B	C	D	E	F
Right	68.04	69.82	72.6	70.21	79.4	88.67
Middle	60.52	62.56	65.57	61.45	71.57	79.04
Left	52.83**	55.25**	58.19	55.12	63.99	71.15

Measurement 29

	A	B	C	D	E	F
Right	104.3	74.67	62.04	61.43	76.72	89.64
Middle	96.77	67.1	55.03	53.91	65.69	79.46
Left	89.2	59.16	47.74	47.18	57.78	72.05

Appendix II

relevant MATLAB Code

Import14.m - reading audio files into Matlab

(example: Measurement 14, Right column.)

```
%Read file into Matlab
[Right_a_14,fs] = audioread('Right_a_14.wav');
[Right_b_14,fs] = audioread('Right_b_14.wav');
[Right_c_14,fs] = audioread('Right_c_14.wav');
[Right_d_14,fs] = audioread('Right_d_14.wav');
[Right_e_14,fs] = audioread('Right_e_14.wav');
[Right_f_14,fs] = audioread('Right_f_14.wav');

%Divide into left and right channels
Right_a_14_l = Right_a_14(:,1);
Right_b_14_l = Right_b_14(:,1);
Right_c_14_l = Right_c_14(:,1);
Right_d_14_l = Right_d_14(:,1);
Right_e_14_l = Right_e_14(:,1);
Right_f_14_l = Right_f_14(:,1);
Right_a_14_r = Right_a_14(:,2);
Right_b_14_r = Right_b_14(:,2);
Right_c_14_r = Right_c_14(:,2);
Right_d_14_r = Right_d_14(:,2);
Right_e_14_r = Right_e_14(:,2);
Right_f_14_r = Right_f_14(:,2);

display('Measurement 14, Right column upload complete')
```

Incode14.m – plot file to identify timestamp for starting point of relevant audio

```
function[] = incode14(fs,File)
%Filename=([sprintf('%s',File) '.wav']);
figure('units','normalized','outerposition',[0 0 1 1])
time = (1/fs)*length(File);
t = linspace(0,time,length(File));
plot(t,File(1:length(File)))
ylim([-.02,.02])
```

Spikefinder.m – identify the largest frequency via a FFT, export its value and coefficient

```
function[r,max1] = spikefinder(n,fs)
L = length(n);
L2 = round(L/2);
time = (1/fs)*L;
```

```
t = linspace(0,time,L);
fn = fft(n);
fa = abs(fn(1:L2)); %only the first half is essential, these are the frequency
amplitudes
fmax = fs/2; % the maximum frequency
fq = ((0:L2-1)/L2)*fmax; %frequencies
plot(fq,fa);
[max1,index1]=max(fa(:)); %finding the spike
[p,q] = ind2sub(size(fa), index1);
r = p*fs/L; %printing the actual frequency
display(sprintf('The strongest frequency of the file is %s Hz with a relative
amplitude of %d .',r,max1))
if r < 1900
    display(sprintf('There may be noise in the above file'))
end
```

Filer.m – Manually insert timestamps for each Column and distance to the Export.m function.

Example: Measurement 15, Left Column

```
[Left_a_15l_amp,Left_a_15r_amp]=export15('Left_a_15', 39.77);
save('Left_a_15r_amp.mat','Left_a_15r_amp')
save('Left_a_15l_amp.mat','Left_a_15l_amp')
display('End of a')

[Left_b_15l_amp,Left_b_15r_amp]=export15('Left_b_15', 46.15);
save('Left_b_15r_amp.mat','Left_b_15r_amp')
save('Left_b_15l_amp.mat','Left_b_15l_amp')
display('End of b')

[Left_c_15l_amp,Left_c_15r_amp]=export15('Left_c_15', 54.71);
save('Left_c_15r_amp.mat','Left_c_15r_amp')
save('Left_c_15l_amp.mat','Left_c_15l_amp')
display('End of c')

[Left_d_15l_amp,Left_d_15r_amp]=export15('Left_d_15', 62.66);
save('Left_d_15r_amp.mat','Left_d_15r_amp')
save('Left_d_15l_amp.mat','Left_d_15l_amp')
display('End of d')

[Left_e_15l_amp,Left_e_15r_amp]=export15('Left_e_15', 57.21);
save('Left_e_15r_amp.mat','Left_e_15r_amp')
save('Left_e_15l_amp.mat','Left_e_15l_amp')
display('End of e')

[Left_f_15l_amp,Left_f_15r_amp]=export15('Left_f_15', 47.9);
save('Left_f_15r_amp.mat','Left_f_15r_amp')
save('Left_f_15l_amp.mat','Left_f_15l_amp')
display('End of f')
```

Export14.m – algorithm to clip file into its twenty, minute-long phases.

Example: Measurement 14

```
function [amp_l,amp_r]=export14(File, a)
```

```
%------------Variables you change-----------------------
%"a" is the starting point (in seconds), slightly different for each recording.
AFTER verbal ID, ~.5 seconds before first audio starts

Filename=([sprintf('%s',File) '.wav']);
filename_r=([sprintf('%s',File) '_r%d.wav']);
filename_l=([sprintf('%s',File) '_l%d.wav']);

%-----------Read in file-------------------------------
[Audio,fs]=audioread(Filename);

%-----------Assign Channels----------------------------
Right_Channel=Audio(:,2);
Left_Channel=Audio(:,1);

for temp=0:19
    file_r=sprintf(filename_r,temp);
    file_l=sprintf(filename_l,temp);

    if temp==0
        clipped_r=Right_Channel(int64(a)*fs:int64(a+63.1)*fs);
        clipped_l=Right_Channel(int64(a)*fs:int64(a+63.1)*fs);
    elseif temp==1
        clipped_r=Right_Channel(int64(a+63.2)*fs:int64(a+128)*fs);
        clipped_l=Left_Channel(int64(a+63.2)*fs:int64(a+128)*fs);
    elseif temp==2
        clipped_r=Right_Channel(int64(a+128.1)*fs:int64(a+192.9)*fs);
        clipped_l=Left_Channel(int64(a+128.1)*fs:int64(a+192.9)*fs);
    elseif temp==3
        clipped_r=Right_Channel(int64(a+193)*fs:int64(a+257.3)*fs);
        clipped_l=Left_Channel(int64(a+193)*fs:int64(a+257.3)*fs);
    elseif temp==4
        clipped_r=Right_Channel(int64(a+257.3)*fs:int64(a+322)*fs);
        clipped_l=Left_Channel(int64(a+257.3)*fs:int64(a+322)*fs);
    elseif temp==5
        clipped_r=Right_Channel(int64(a+322.1)*fs:int64(a+387.1)*fs);
        clipped_l=Left_Channel(int64(a+322.1)*fs:int64(a+387.1)*fs);
    elseif temp==6
        clipped_r=Right_Channel(int64(a+387.2)*fs:int64(a+452.5)*fs);
        clipped_l=Left_Channel(int64(a+387.2)*fs:int64(a+452.5)*fs);
    elseif temp==7
        clipped_r=Right_Channel(int64(a+452.6)*fs:int64(a+517.4)*fs);
        clipped_l=Left_Channel(int64(a+452.6)*fs:int64(a+517.4)*fs);
    elseif temp==8
        clipped_r=Right_Channel(int64(a+522.5)*fs:int64(a+587.4)*fs);
        clipped_l=Left_Channel(int64(a+522.5)*fs:int64(a+587.4)*fs);
    elseif temp==9
        clipped_r=Right_Channel(int64(a+593.5)*fs:int64(a+657.9)*fs);
        clipped_l=Left_Channel(int64(a+593.5)*fs:int64(a+657.9)*fs);
    elseif temp==10
        clipped_r=Right_Channel(int64(a+665.1)*fs:int64(a+729.5)*fs);
        clipped_l=Left_Channel(int64(a+665.1)*fs:int64(a+729.5)*fs);
    elseif temp==11
        clipped_r=Right_Channel(int64(a+738.5)*fs:int64(a+803)*fs);
        clipped_l=Left_Channel(int64(a+738.5)*fs:int64(a+803)*fs);
    elseif temp==12
        clipped_r=Right_Channel(int64(a+806.5)*fs:int64(a+870.5)*fs);
        clipped_l=Left_Channel(int64(a+806.5)*fs:int64(a+870.5)*fs);
    elseif temp==13
        clipped_r=Right_Channel(int64(a+875.4)*fs:int64(a+939.6)*fs);
        clipped_l=Left_Channel(int64(a+875.4)*fs:int64(a+939.6)*fs);
    elseif temp==14
        clipped_r=Right_Channel(int64(a+944.7)*fs:int64(a+1008.4)*fs);
```

```
        clipped_l=Left_Channel(int64(a+944.7)*fs:int64(a+1008.4)*fs);
    elseif temp==15
        clipped_r=Right_Channel(int64(a+1008.5)*fs:int64(a+1074.6)*fs);
        clipped_l=Left_Channel(int64(a+1008.5)*fs:int64(a+1074.6)*fs);
    elseif temp==16
        clipped_r=Right_Channel(int64(a+1074.7)*fs:int64(a+1140.6)*fs);
        clipped_l=Left_Channel(int64(a+1074.7)*fs:int64(a+1140.6)*fs);
    elseif temp==17
        clipped_r=Right_Channel(int64(a+1140.7)*fs:int64(a+1207.6)*fs);
        clipped_l=Left_Channel(int64(a+1140.7)*fs:int64(a+1207.6)*fs);
    elseif temp==18
        clipped_r=Right_Channel(int64(a+1207.7)*fs:int64(a+1275.6)*fs);
        clipped_l=Left_Channel(int64(a+1207.7)*fs:int64(a+1275.6)*fs);
    elseif temp==19
        clipped_r=Right_Channel(int64(a+1275.7)*fs:int64(a+1343.6)*fs);
        clipped_l=Left_Channel(int64(a+1275.7)*fs:int64(a+1343.6)*fs);

        figure
        time = (1/fs)*length(clipped_r);
        t = linspace(0,time,length(clipped_r));
        plot(t,clipped_r)

        figure
        time = (1/fs)*length(clipped_l);
        t = linspace(0,time,length(clipped_l));
        plot(t,clipped_l)

    end

   audiowrite(file_r,clipped_r,fs);
   audiowrite(file_l,clipped_l,fs);

    [r,max1] = spikefinder(clipped_r,fs);
    amp_r(temp+1)=max1;

    [r,max1] = spikefinder(clipped_l,fs);
    amp_l(temp+1)=max1;

end
figure
plot(amp_r(2:20),'ko')

figure
plot(amp_l(2:20),'ko')
```

Averages.m – average the right and left channel Spikefinder.m outputs into one variable

```
Left_a_15avg_amp=(Left_a_15l_amp + Left_a_15r_amp)/2;
Left_b_15avg_amp=(Left_b_15l_amp + Left_b_15r_amp)/2;
Left_c_15avg_amp=(Left_c_15l_amp + Left_c_15r_amp)/2;
Left_d_15avg_amp=(Left_d_15l_amp + Left_d_15r_amp)/2;
Left_e_15avg_amp=(Left_e_15l_amp + Left_e_15r_amp)/2;
Left_f_15avg_amp=(Left_f_15l_amp + Left_f_15r_amp)/2;
Right_a_15avg_amp=(Right_a_15l_amp + Right_a_15r_amp)/2;
Right_b_15avg_amp=(Right_b_15l_amp + Right_b_15r_amp)/2;
```

```
Right_c_15avg_amp=(Right_c_15l_amp + Right_c_15r_amp)/2;
Right_d_15avg_amp=(Right_d_15l_amp + Right_d_15r_amp)/2;
Right_e_15avg_amp=(Right_e_15l_amp + Right_e_15r_amp)/2;
Right_f_15avg_amp=(Right_f_15l_amp + Right_f_15r_amp)/2;
Middle_a_15avg_amp=(Middle_a_15l_amp + Middle_a_15r_amp)/2;
Middle_b_15avg_amp=(Middle_b_15l_amp + Middle_b_15r_amp)/2;
Middle_c_15avg_amp=(Middle_c_15l_amp + Middle_c_15r_amp)/2;
Middle_d_15avg_amp=(Middle_d_15l_amp + Middle_d_15r_amp)/2;
Middle_e_15avg_amp=(Middle_e_15l_amp + Middle_e_15r_amp)/2;
Middle_f_15avg_amp=(Middle_f_15l_amp + Middle_f_15r_amp)/2;
save('Left_a_15avg_amp.mat','Left_a_15avg_amp')
save('Left_b_15avg_amp.mat','Left_b_15avg_amp')
save('Left_c_15avg_amp.mat','Left_c_15avg_amp')
save('Left_d_15avg_amp.mat','Left_d_15avg_amp')
save('Left_e_15avg_amp.mat','Left_e_15avg_amp')
save('Left_f_15avg_amp.mat','Left_f_15avg_amp')
save('Right_a_15avg_amp.mat','Right_a_15avg_amp')
save('Right_b_15avg_amp.mat','Right_b_15avg_amp')
save('Right_c_15avg_amp.mat','Right_c_15avg_amp')
save('Right_d_15avg_amp.mat','Right_d_15avg_amp')
save('Right_e_15avg_amp.mat','Right_e_15avg_amp')
save('Right_f_15avg_amp.mat','Right_f_15avg_amp')
save('Middle_a_15avg_amp.mat','Middle_a_15avg_amp')
save('Middle_b_15avg_amp.mat','Middle_b_15avg_amp')
save('Middle_c_15avg_amp.mat','Middle_c_15avg_amp')
save('Middle_d_15avg_amp.mat','Middle_d_15avg_amp')
save('Middle_e_15avg_amp.mat','Middle_e_15avg_amp')
save('Middle_f_15avg_amp.mat','Middle_f_15avg_amp')
```

Plots.m – plot the coefficients per Column and distance for the entire Measurement (Output recorded in Appendix III)

Example: Measurement 15

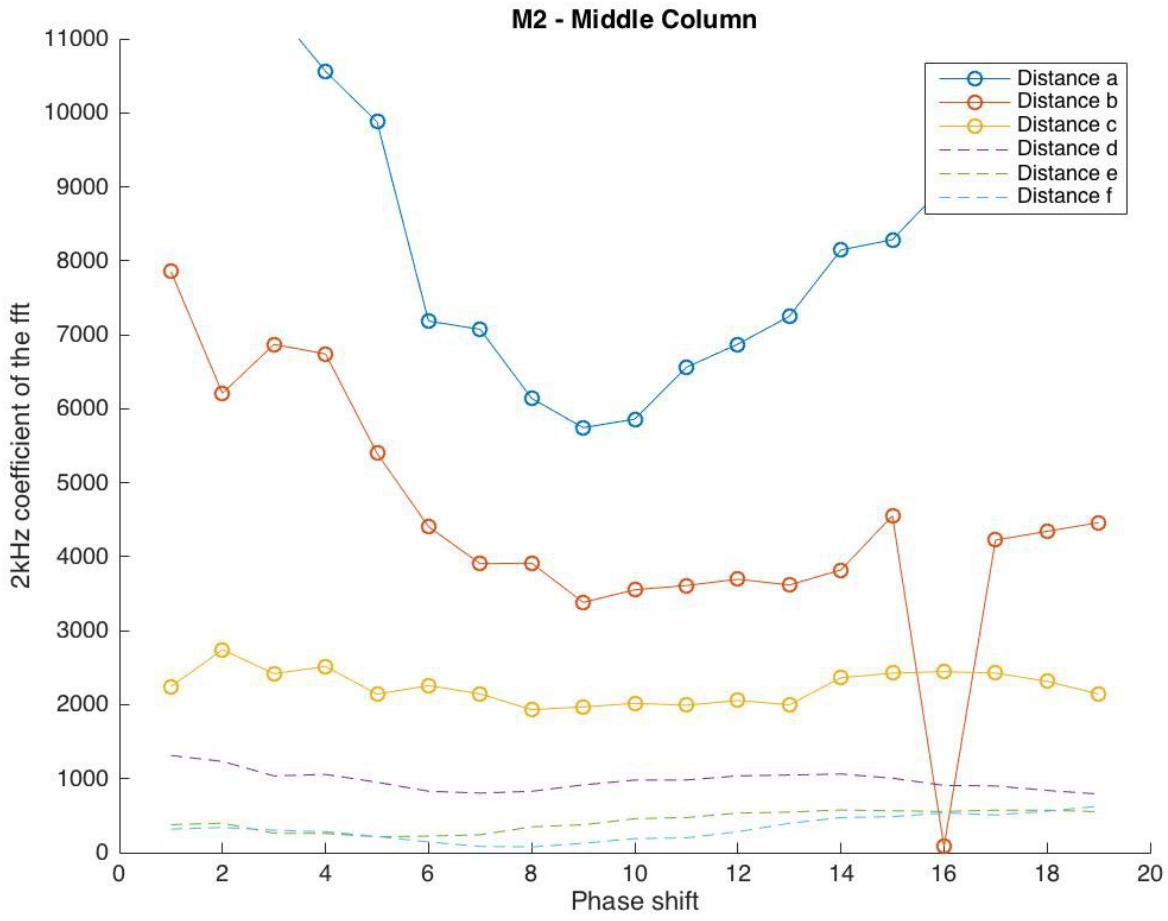
```
figure
hold on
plot(Left_a_15avg_amp(2:20), '-o')
plot(Left_b_15avg_amp(2:20), '-o')
plot(Left_c_15avg_amp(2:20), '-o')
plot(Left_d_15avg_amp(2:20), '--')
plot(Left_e_15avg_amp(2:20), '--')
plot(Left_f_15avg_amp(2:20), '--')
legend('a','b','c','d','e','f')
title('Measurement 20, Left column distances a-f')
xlabel('Phase shift')
ylabel('2kHz coefficient of the fft')

figure
hold on
plot(Middle_a_15avg_amp(2:20), '-o')
plot(Middle_b_15avg_amp(2:20), '-o')
plot(Middle_c_15avg_amp(2:20), '-o')
plot(Middle_d_15avg_amp(2:20), '--')
plot(Middle_e_15avg_amp(2:20), '--')
plot(Middle_f_15avg_amp(2:20), '--')
legend('a','b','c','d','e','f')
title('Measurement 20, Middle column distances a-f')
xlabel('Phase shift')
ylabel('2kHz coefficient of the fft')

figure
hold on
```

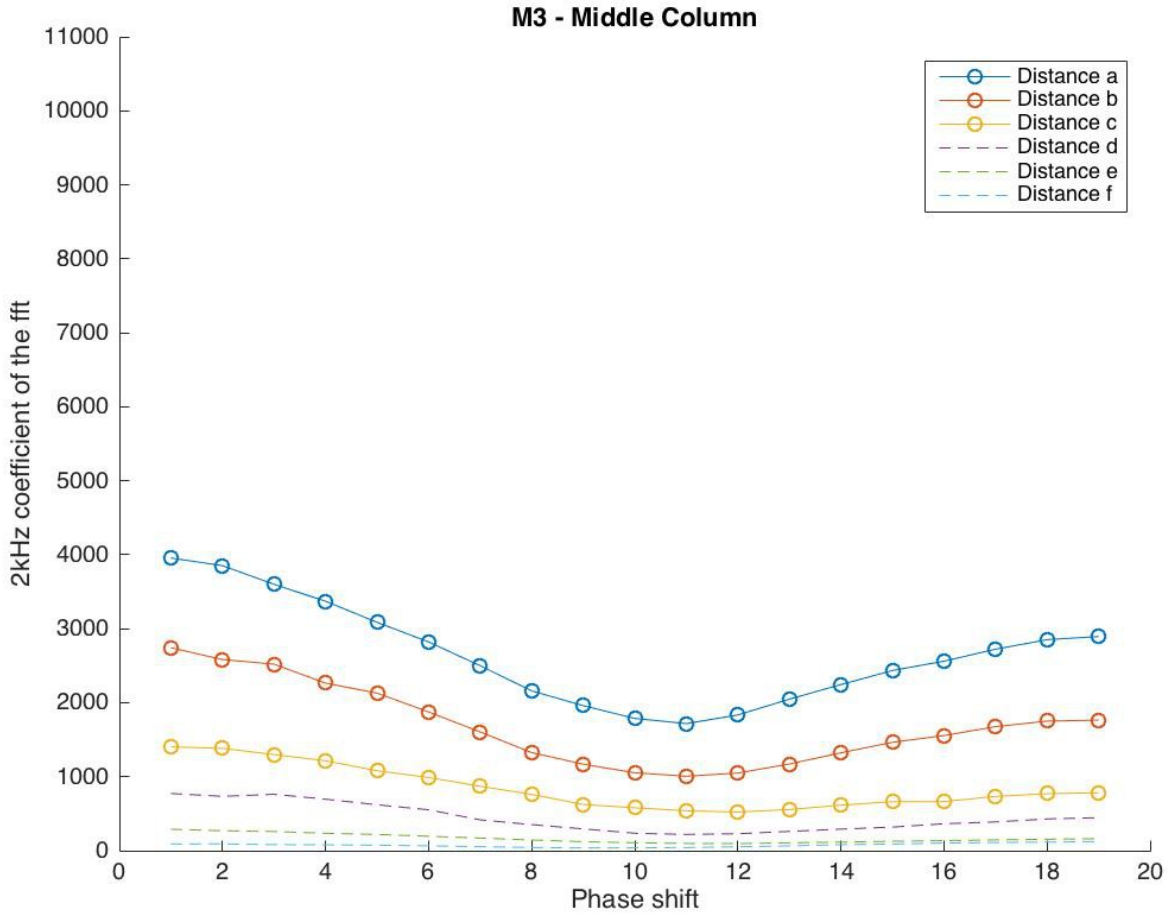
```
plot(Right_a_15avg_amp(2:20), '-o')
plot(Right_b_15avg_amp(2:20), '-o')
plot(Right_c_15avg_amp(2:20), '-o')
plot(Right_d_15avg_amp(2:20), '--')
plot(Right_e_15avg_amp(2:20), '--')
plot(Right_f_15avg_amp(2:20), '--')
legend('a','b','c','d','e','f')
title('Measurement 20, Right column distances a-f')
xlabel('Phase shift')
ylabel('2kHz coefficient of the fft')
```

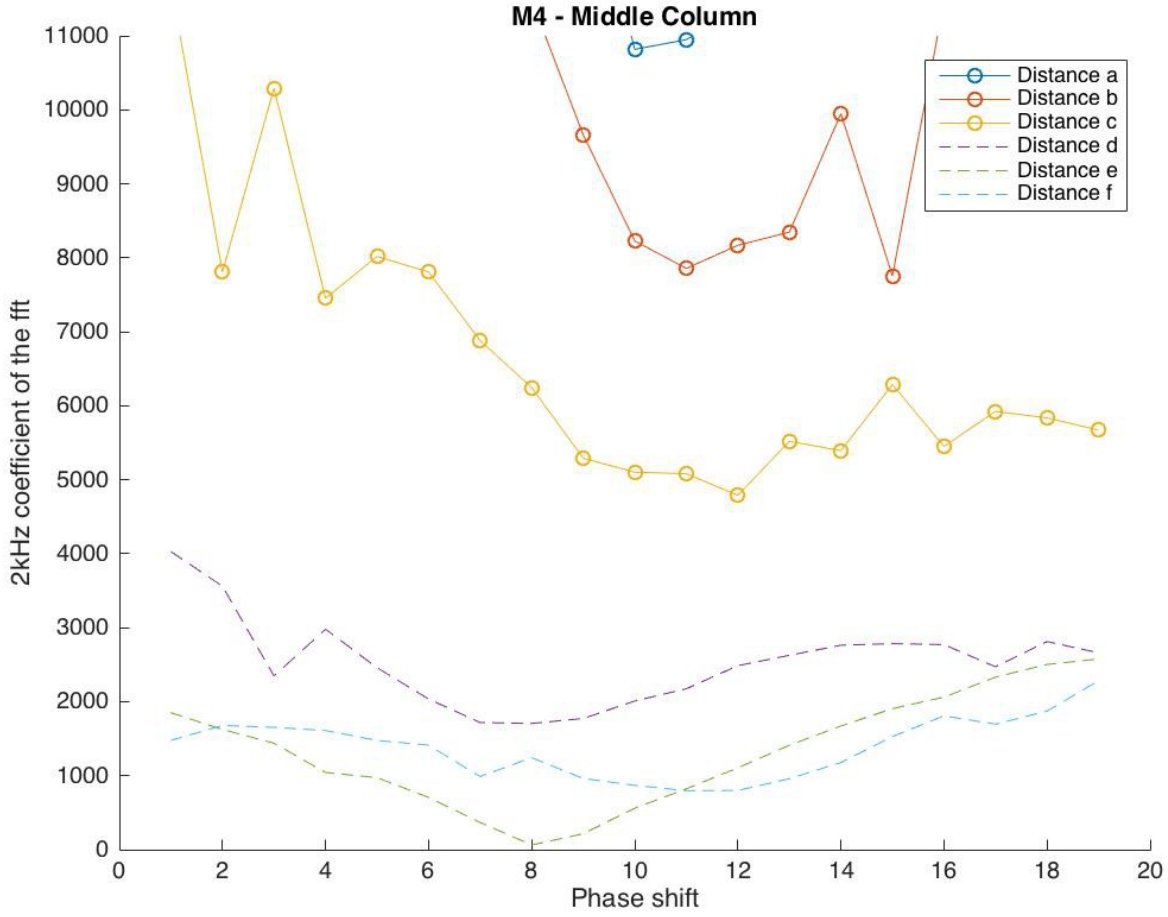
Appendix III

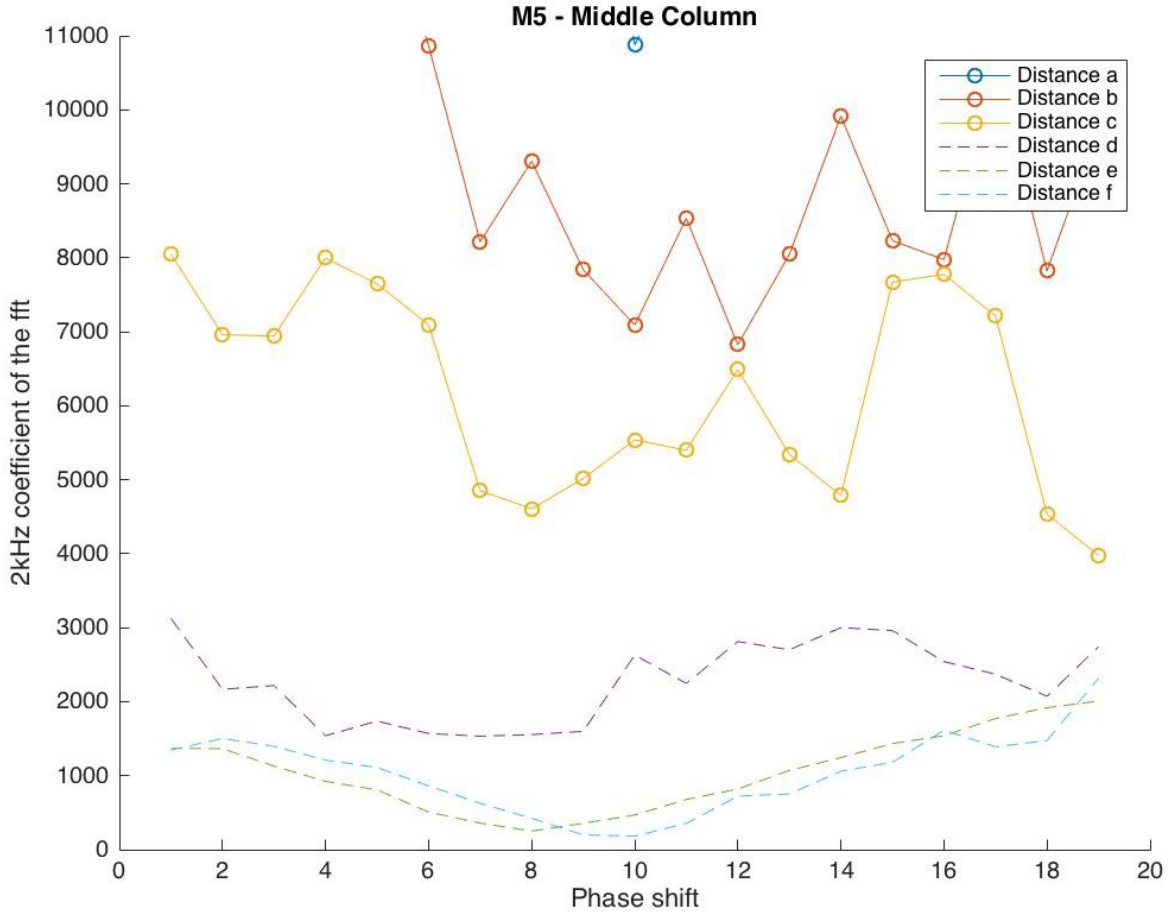
Concluding plots

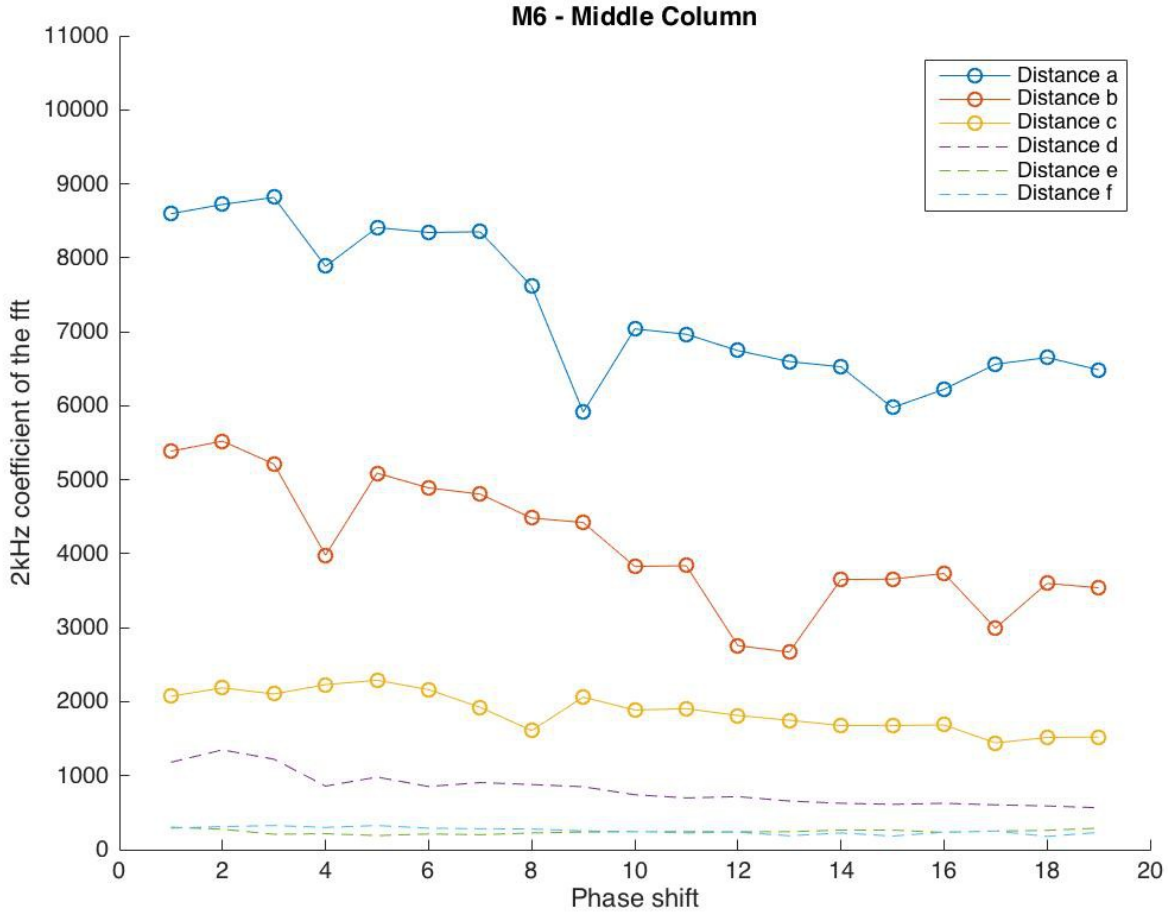
(As discussed in the Data Analysis section, Right and Left columns contain mostly unhelpful data. Therefore, Middle column data is displayed first. Relevant plot clusters shared a fixed y-axis limit)

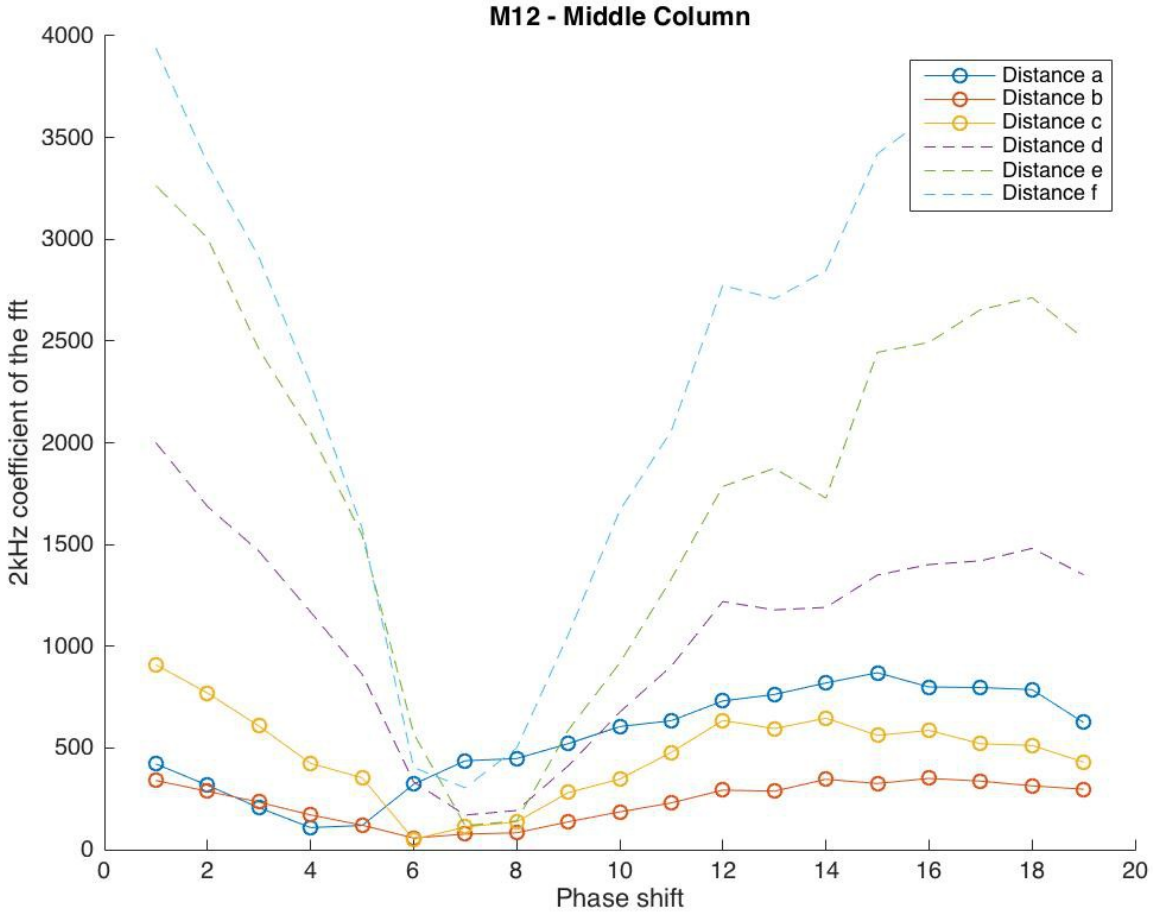


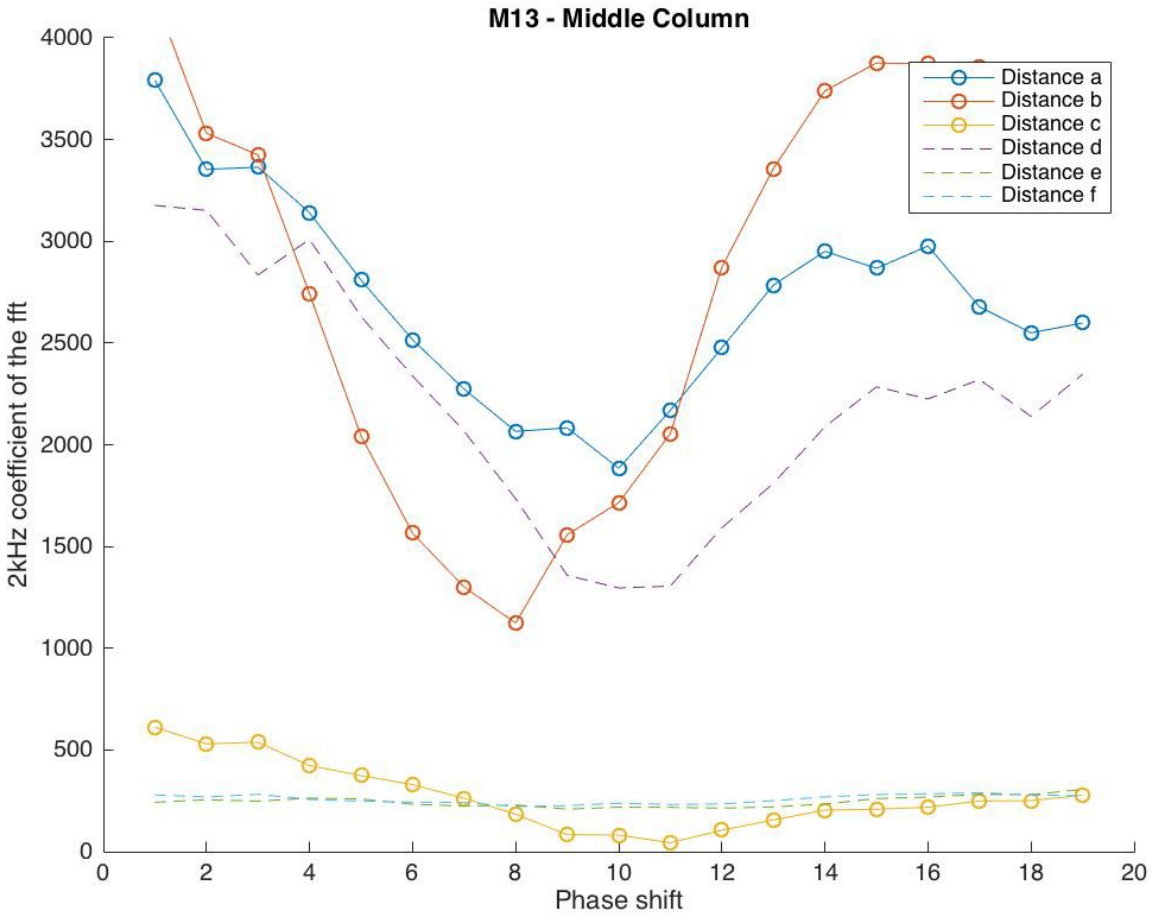

M3 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
2
4
6
8
10
12
14
16
18
20

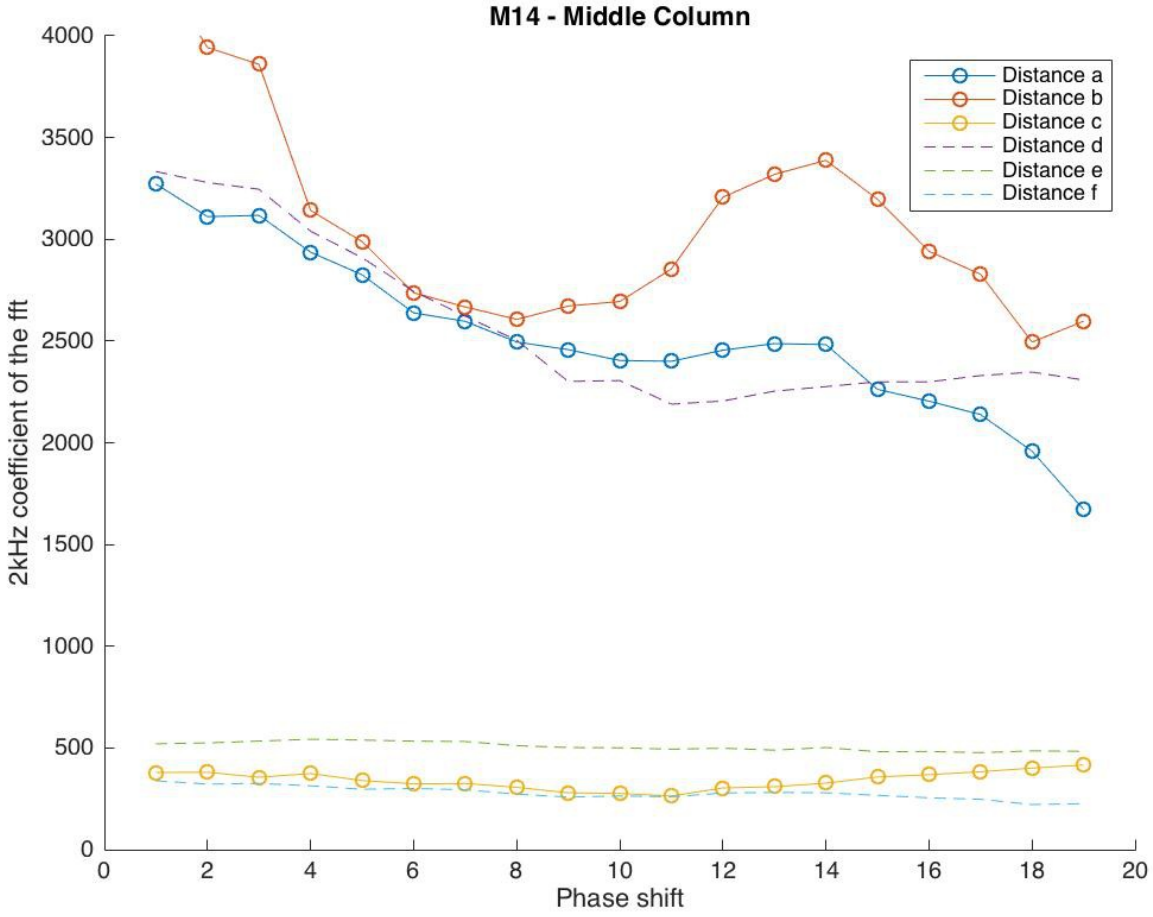

M4 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
2
4
6
8
10
12
14
16
18
20

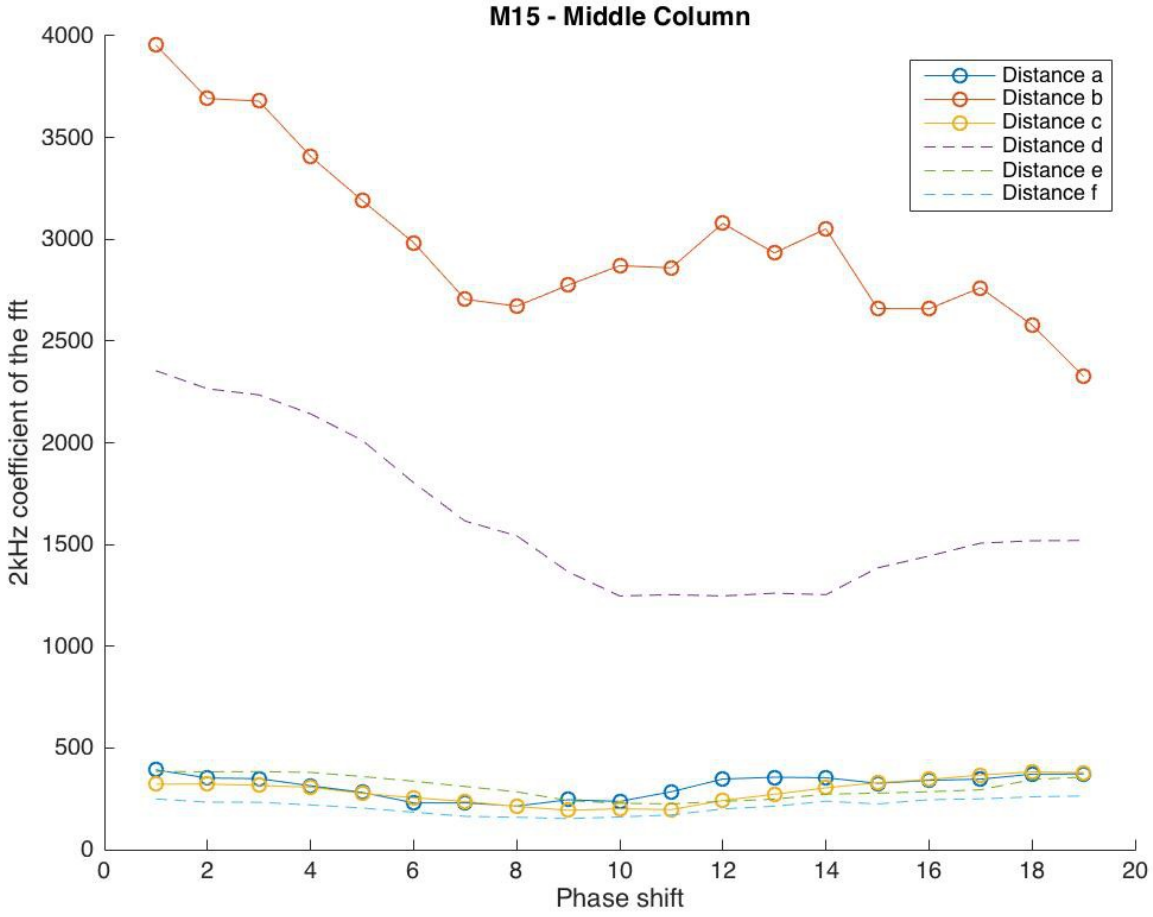

M5 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
0
2
4
6
8
10
12
14
16
18
20

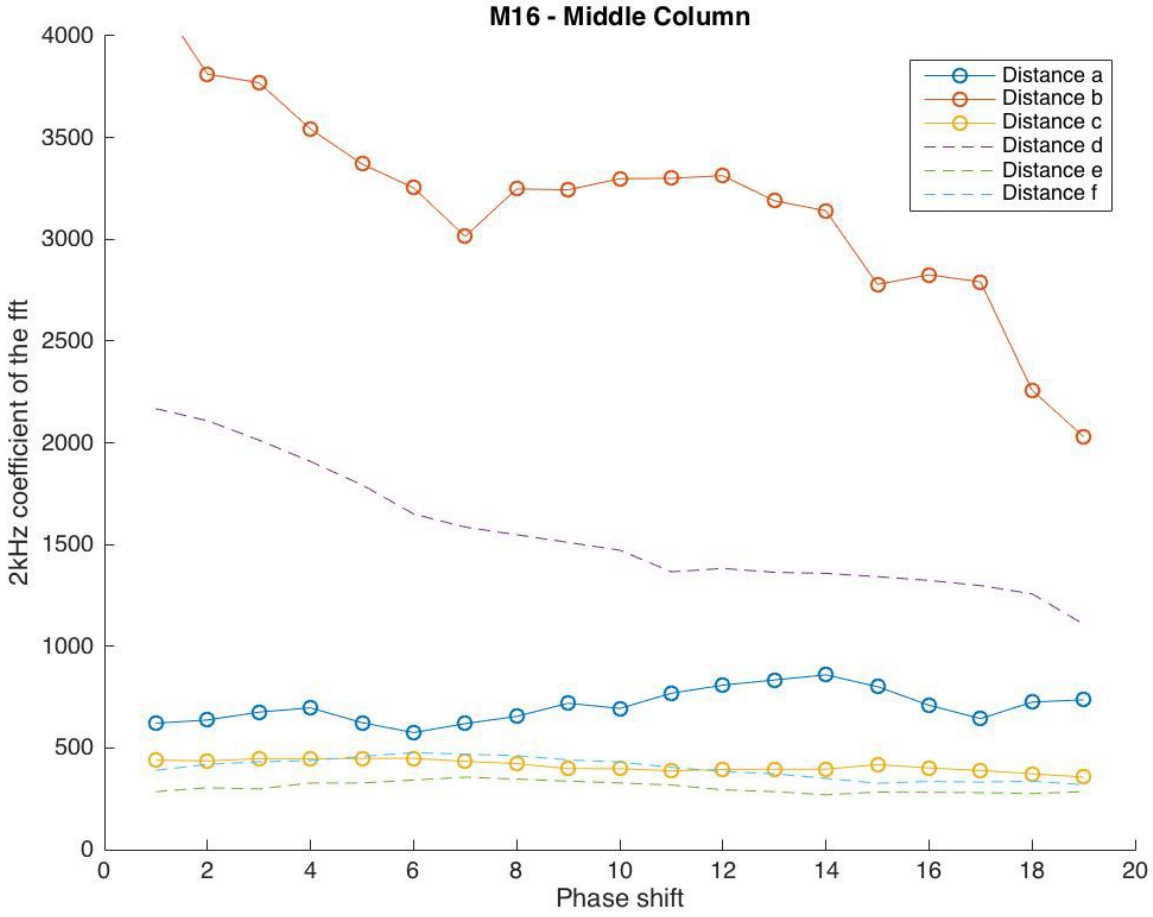

M6 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
2
4
6
8
10
12
14
16
18
20

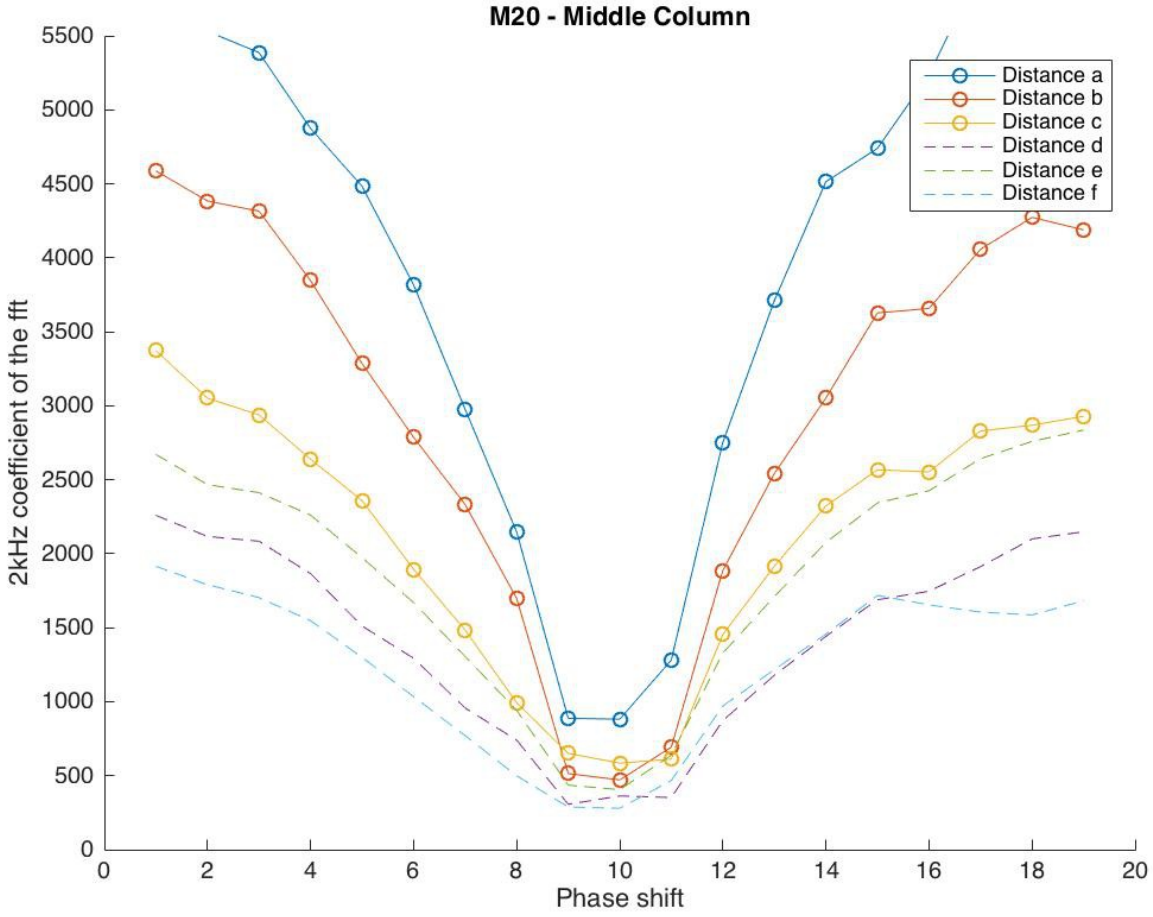

M12 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
3000
3500
4000
0
2
4
6
8
10
12
14
16
18
20

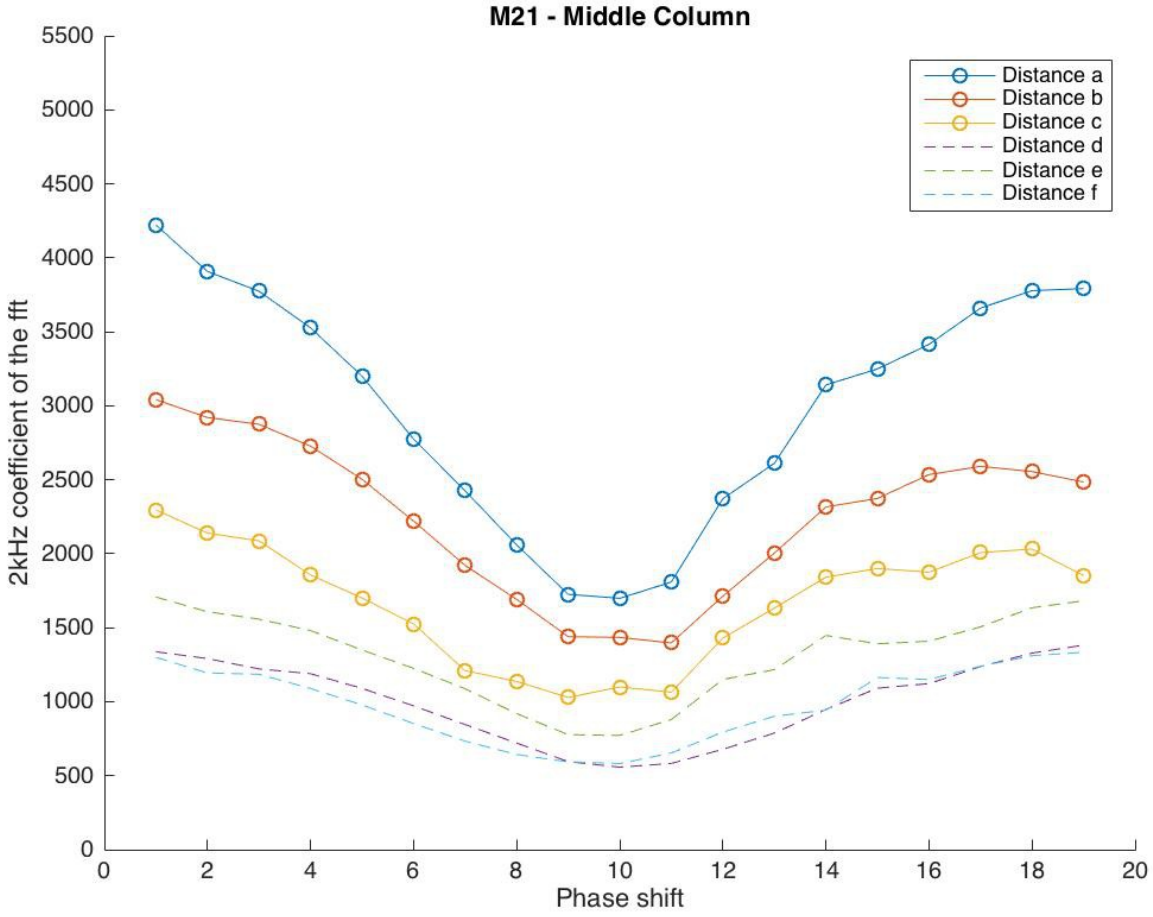

M13 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
3000
3500
4000
0
2
4
6
8
10
12
14
16
18
20

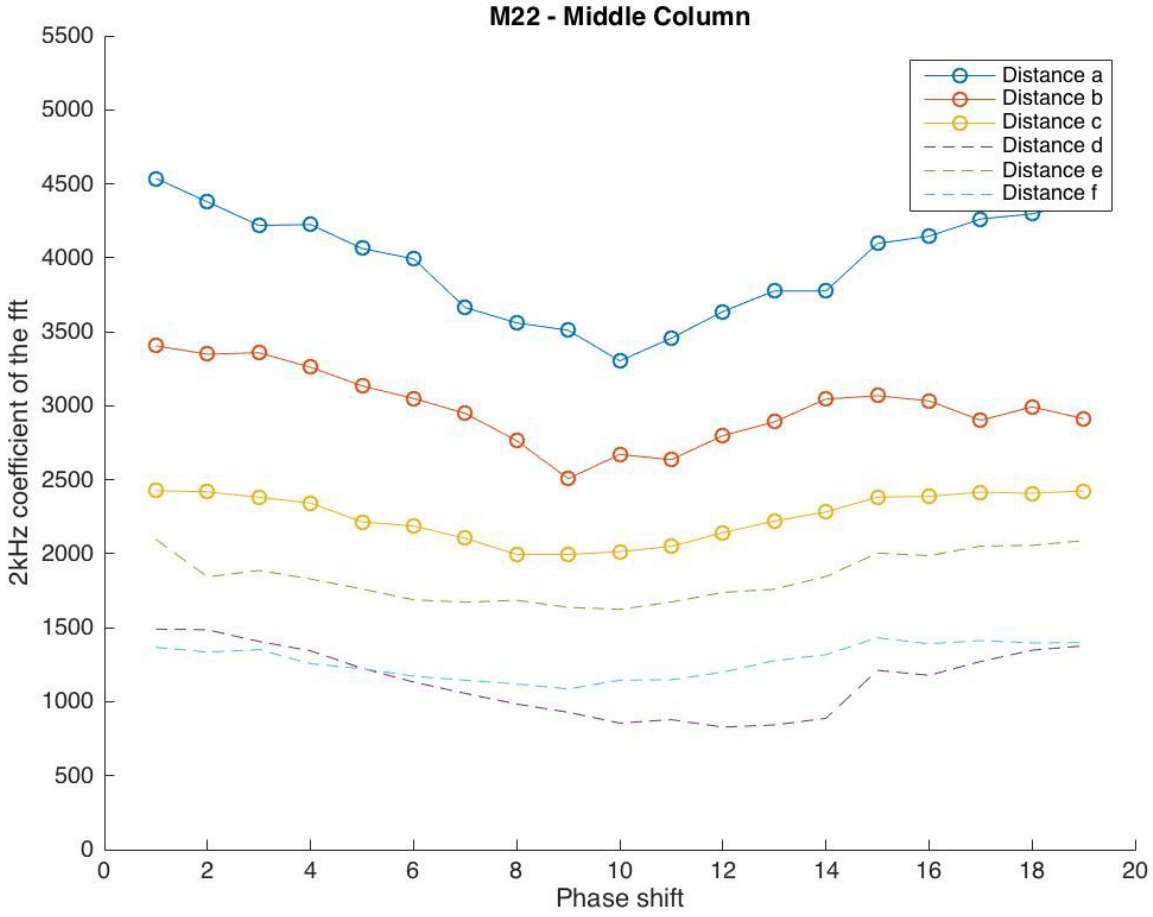

M14 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
3000
3500
4000
0
2
4
6
8
10
12
14
16
18
20

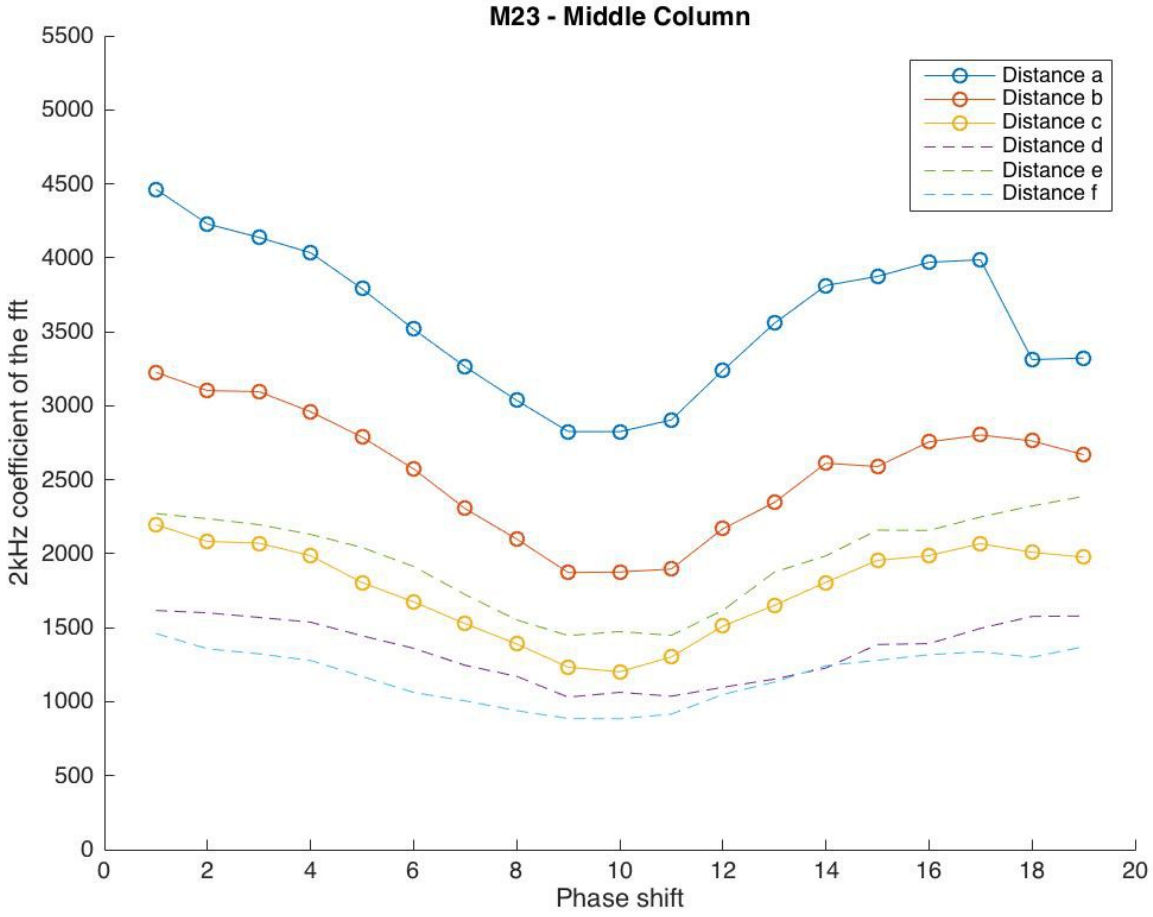

M15 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
3000
3500
4000
0
2
4
6
8
10
12
14
16
18
20

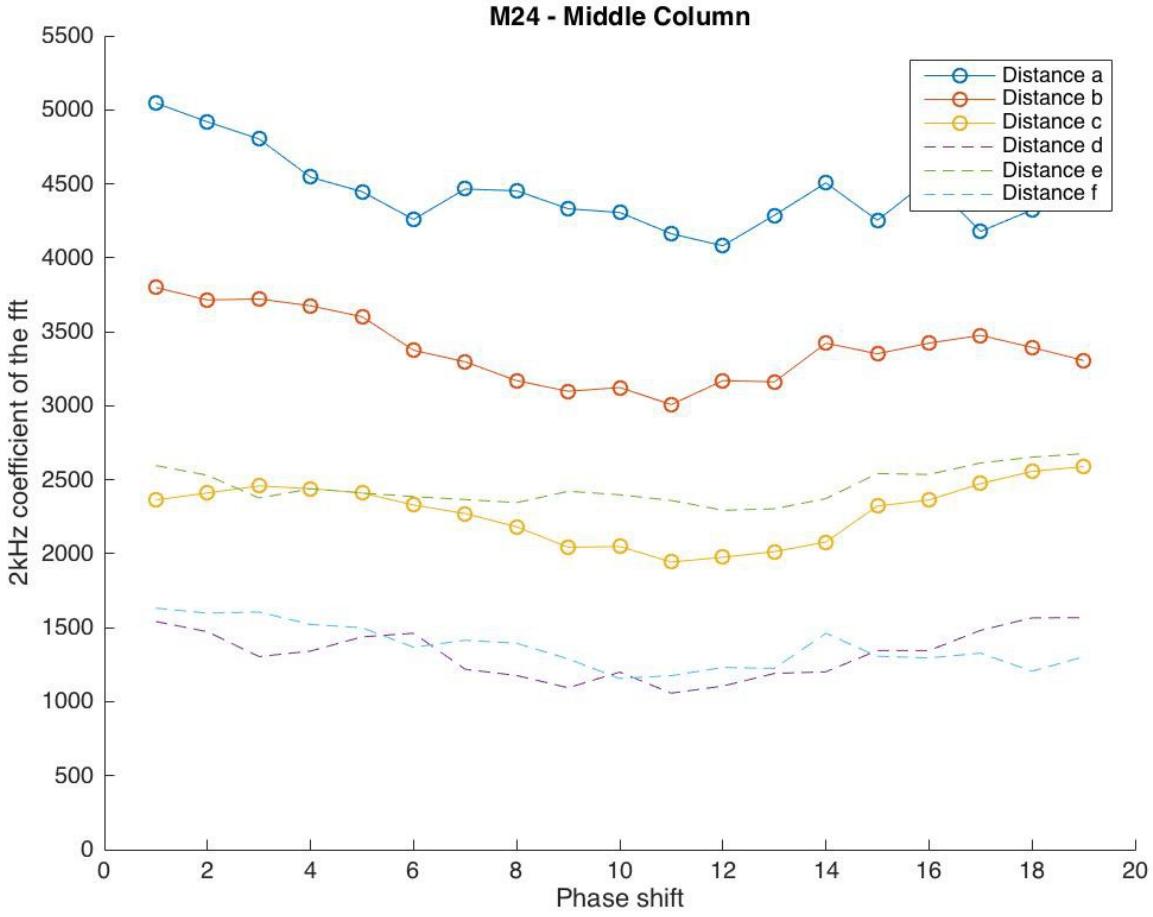

M16 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
3000
3500
4000
0
2
4
6
8
10
12
14
16
18
20

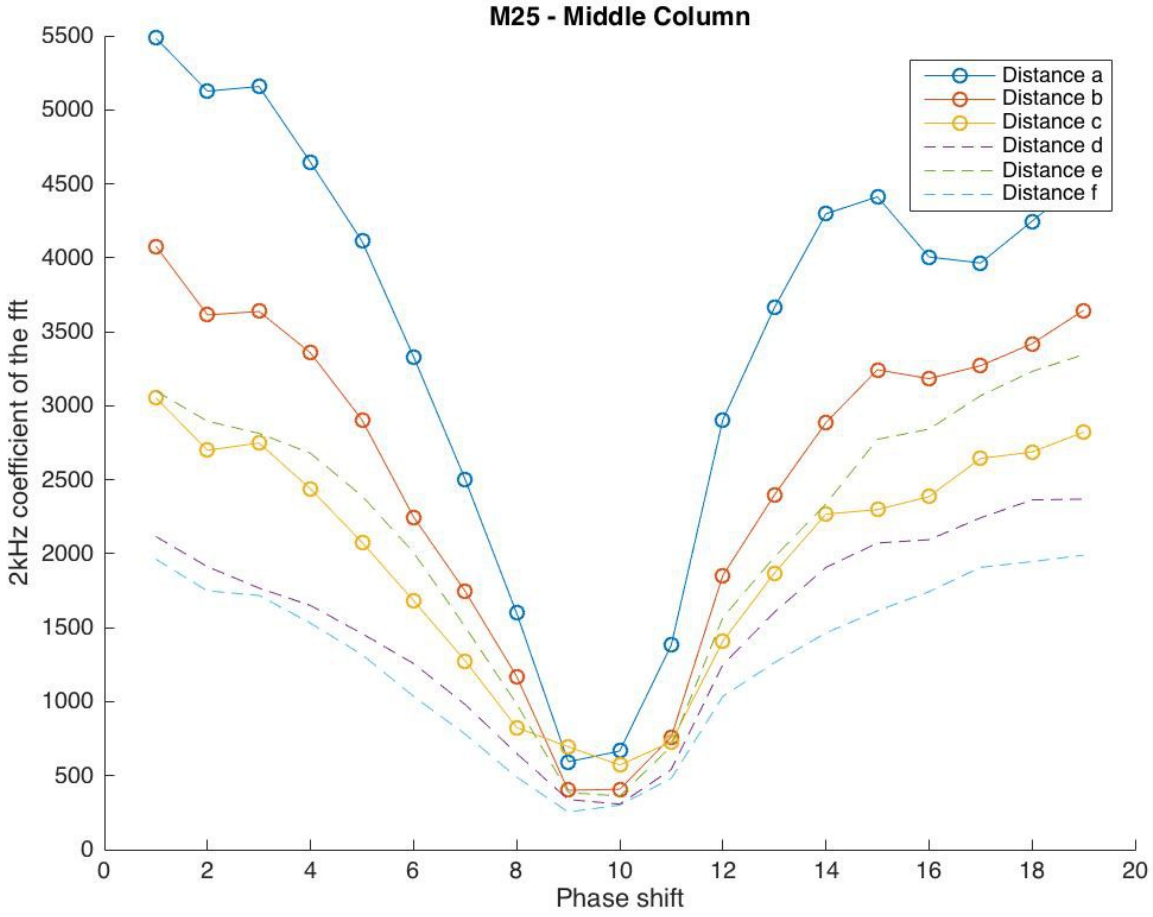

M20 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
5500
0
2
4
6
8
10
12
14
16
18
20

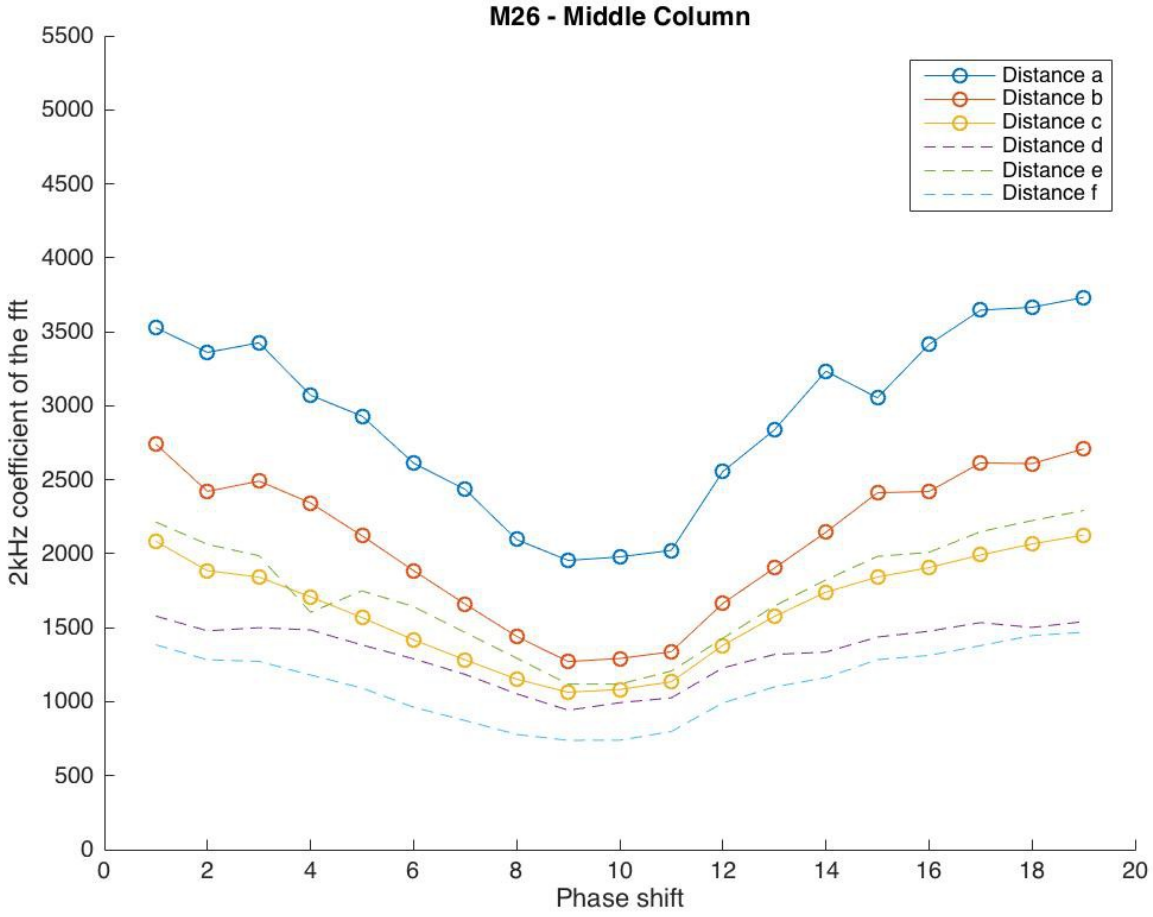

M21 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
5500
0
2
4
6
8
10
12
14
16
18
20

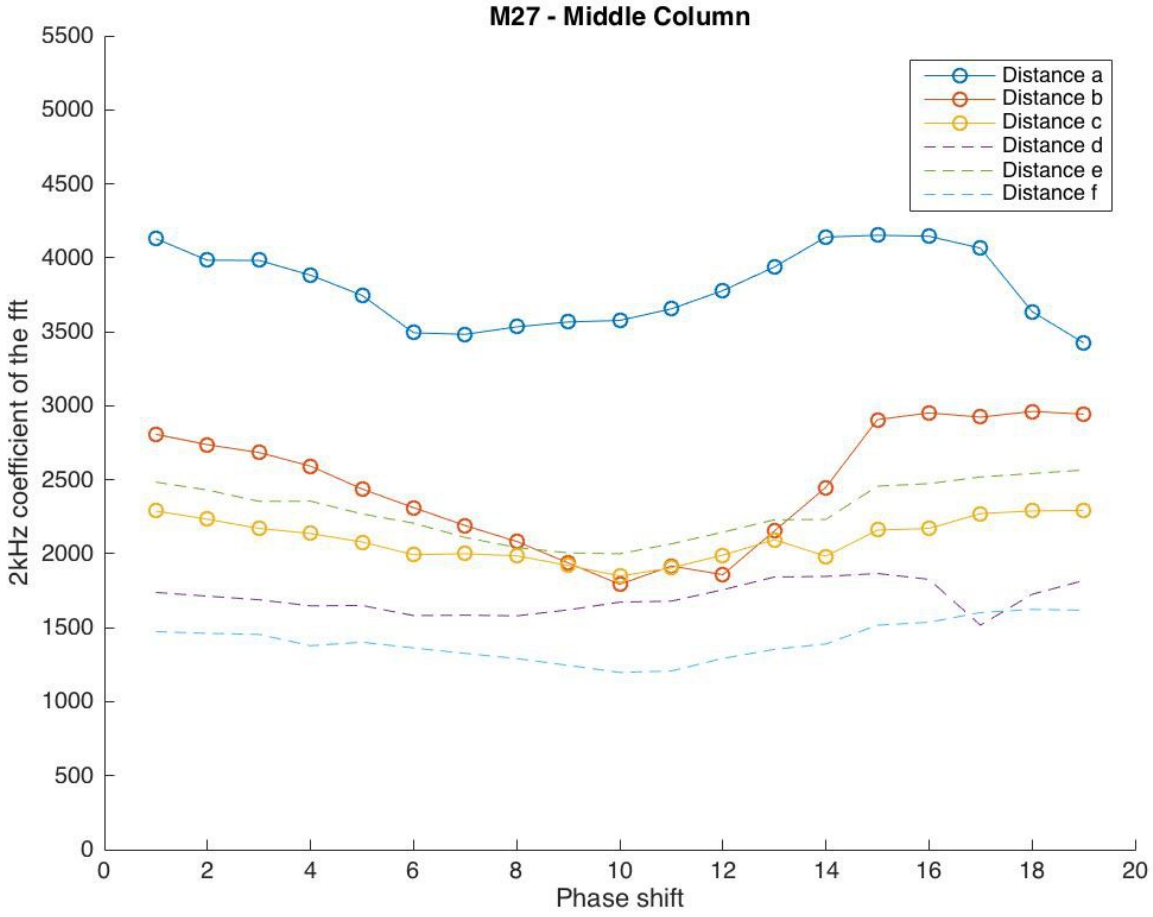

M22 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
12
14
16
18
20

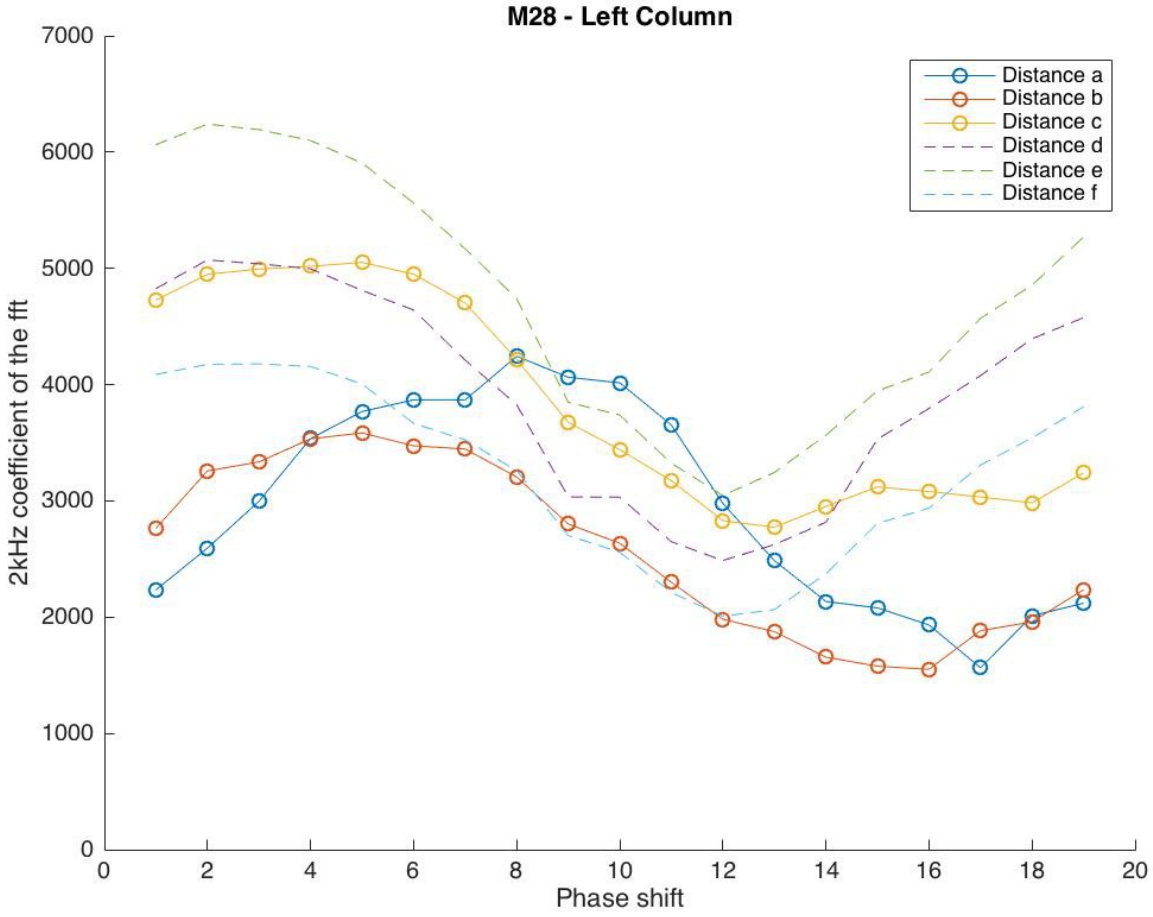

M23 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
12
14
16
18
20

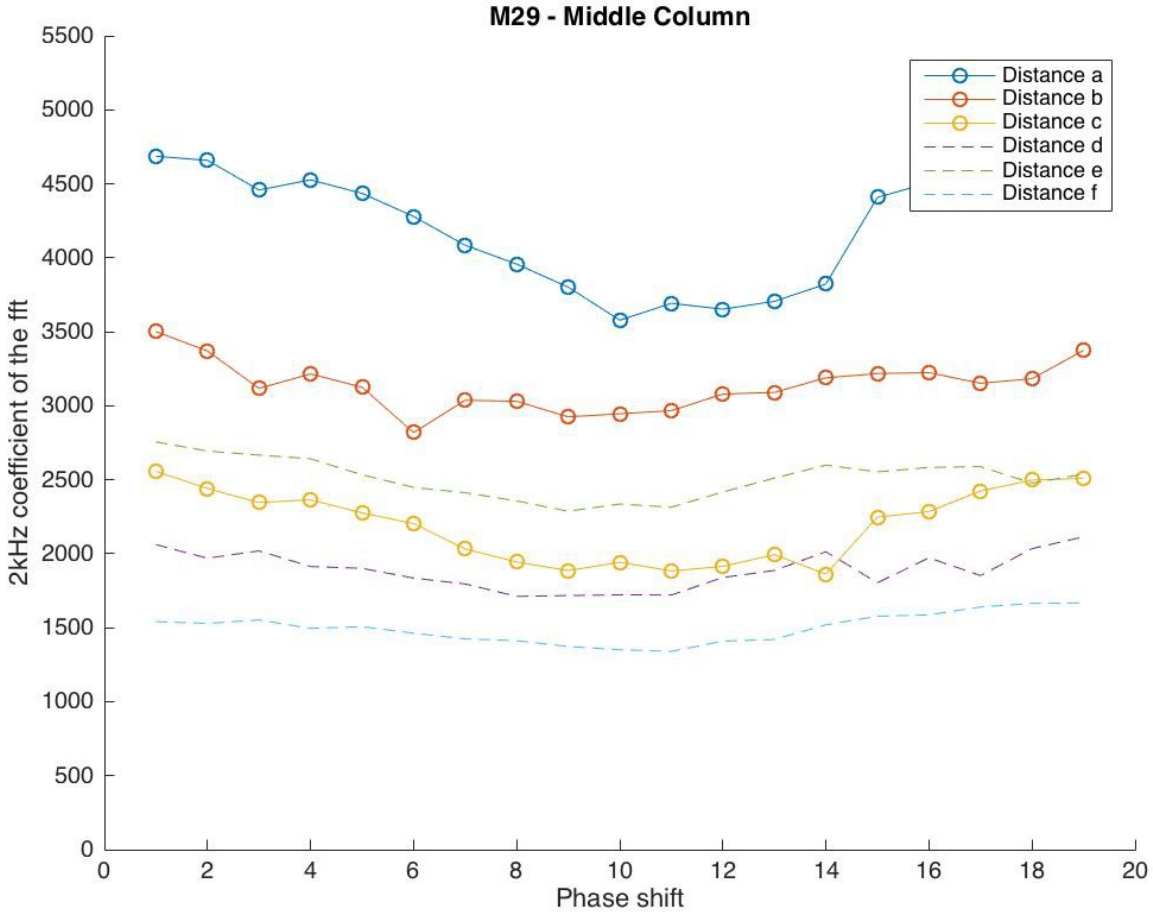

M24 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
12
14
16
18
20

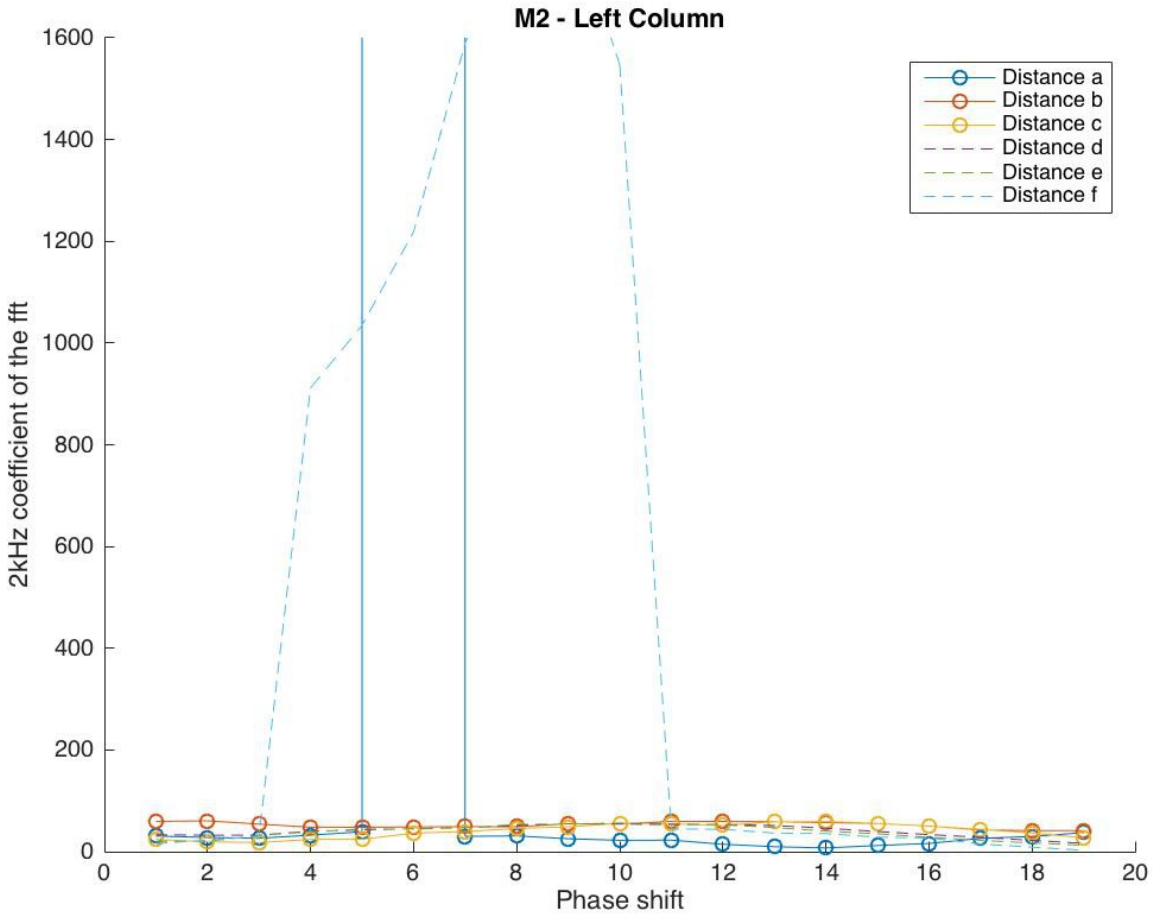

M25 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
5500
0
2
4
6
8
10
12
14
16
18
20

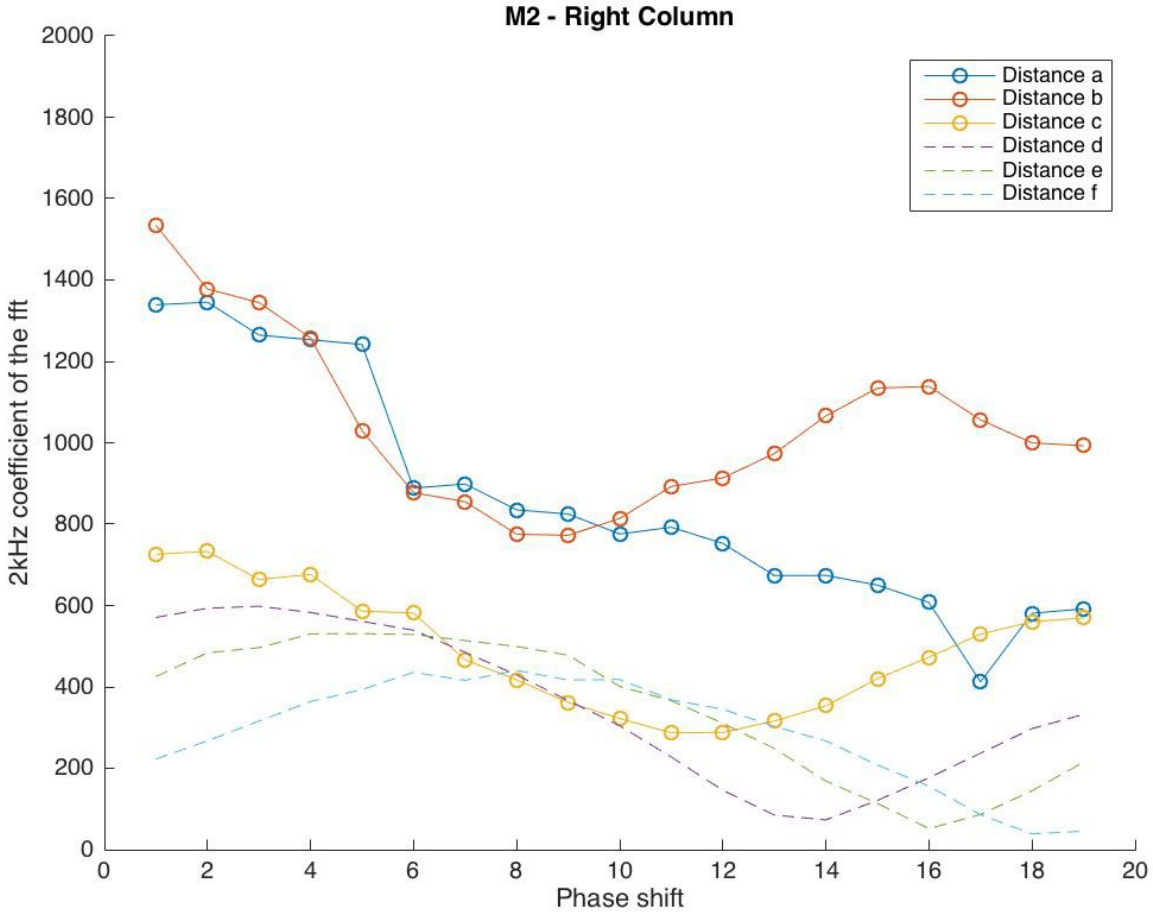

M26 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
12
14
16
18
20

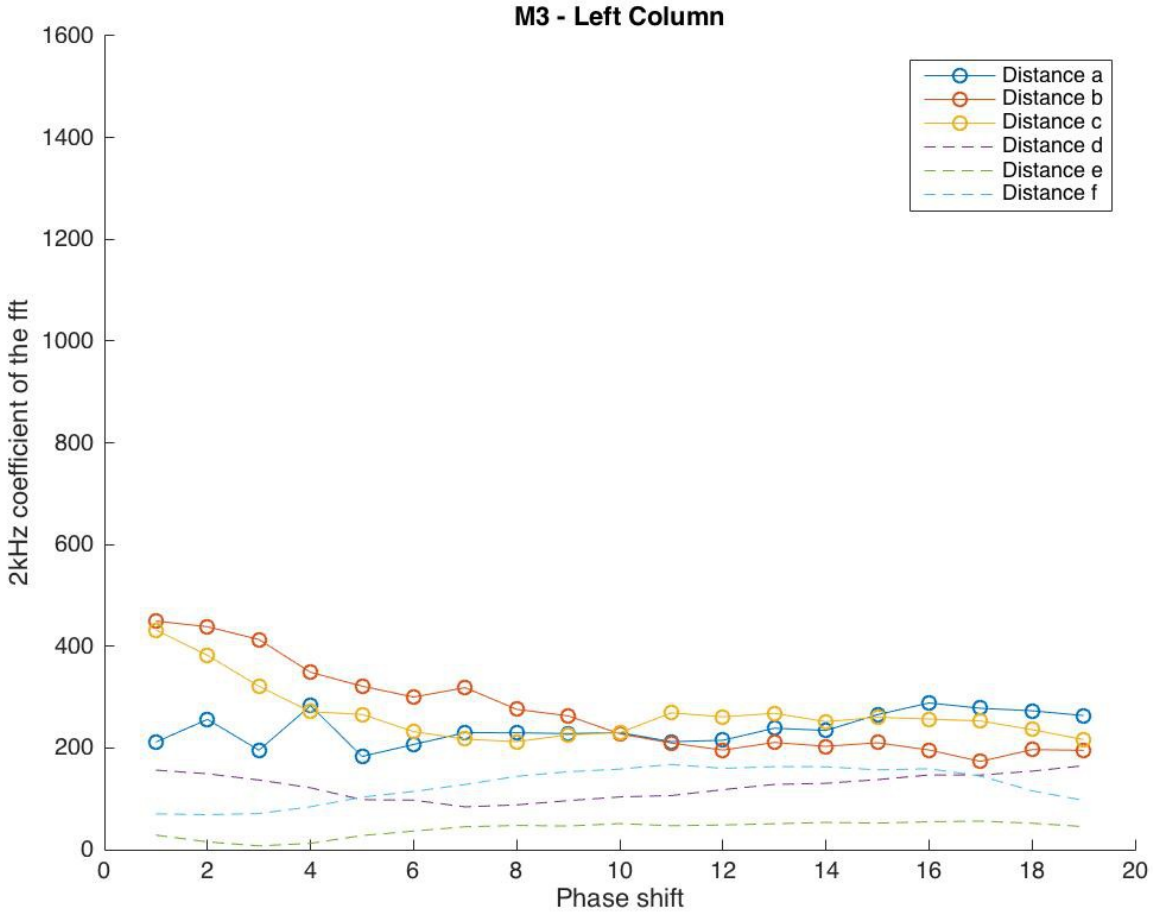

M27 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
12
14
16
18
20

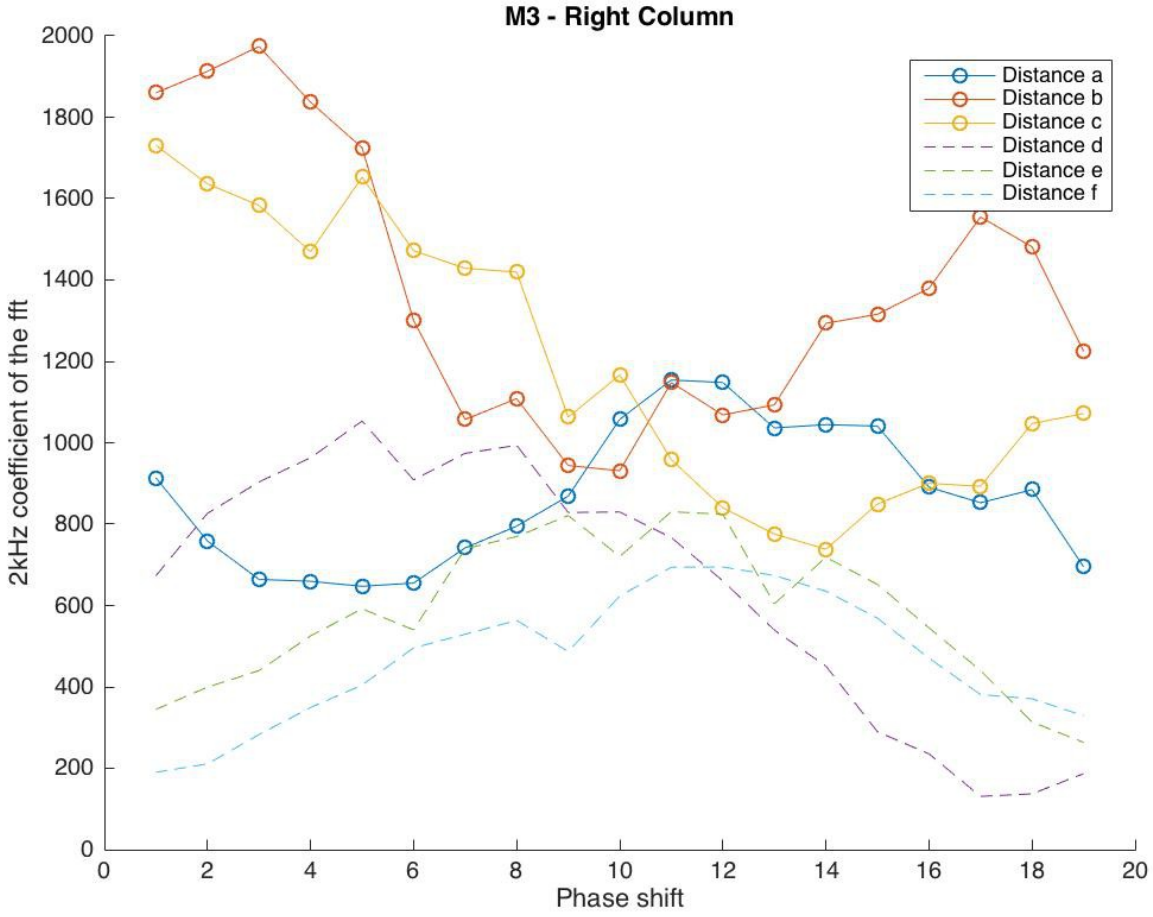

M28 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
1000
2000
3000
4000
5000
6000
7000
0
2
4
6
8
10
12
14
16
18
20

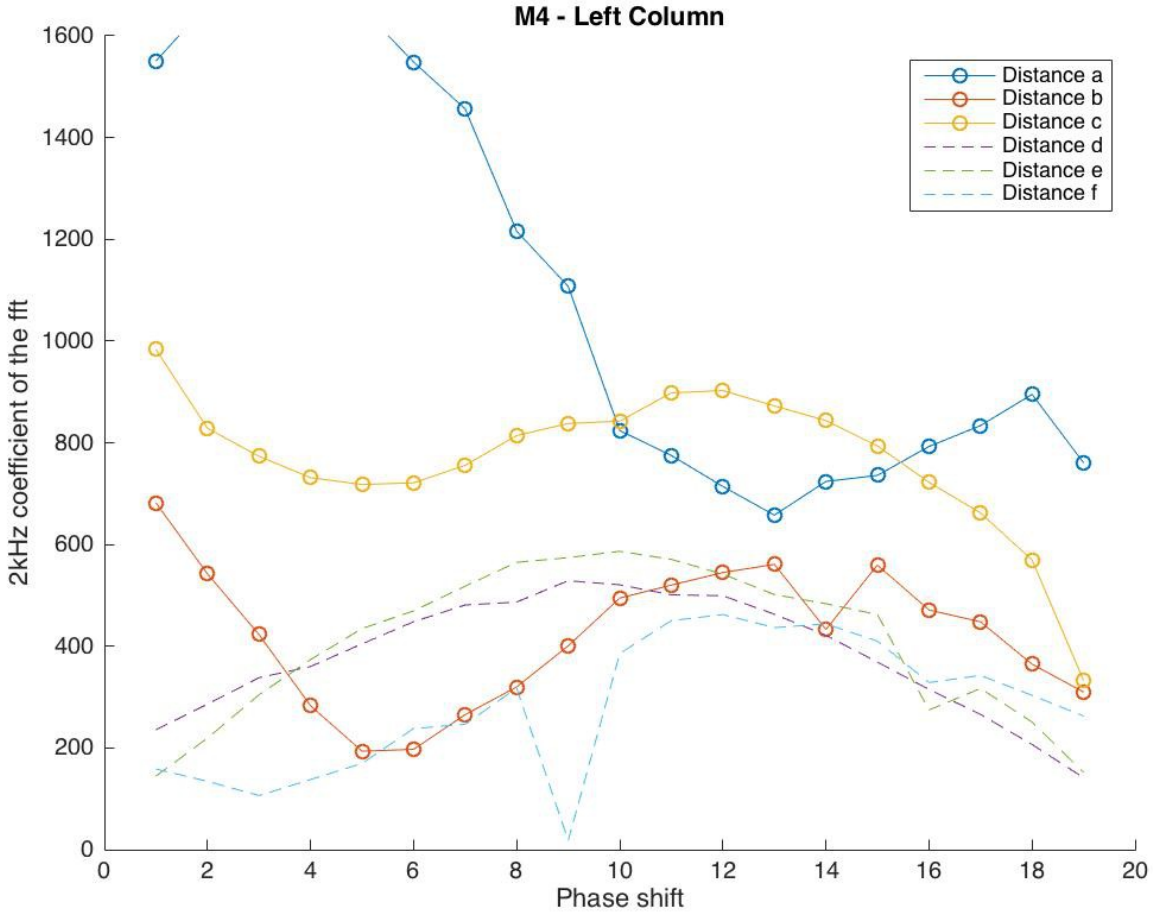

M29 - Middle Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
12
14
16
18
20

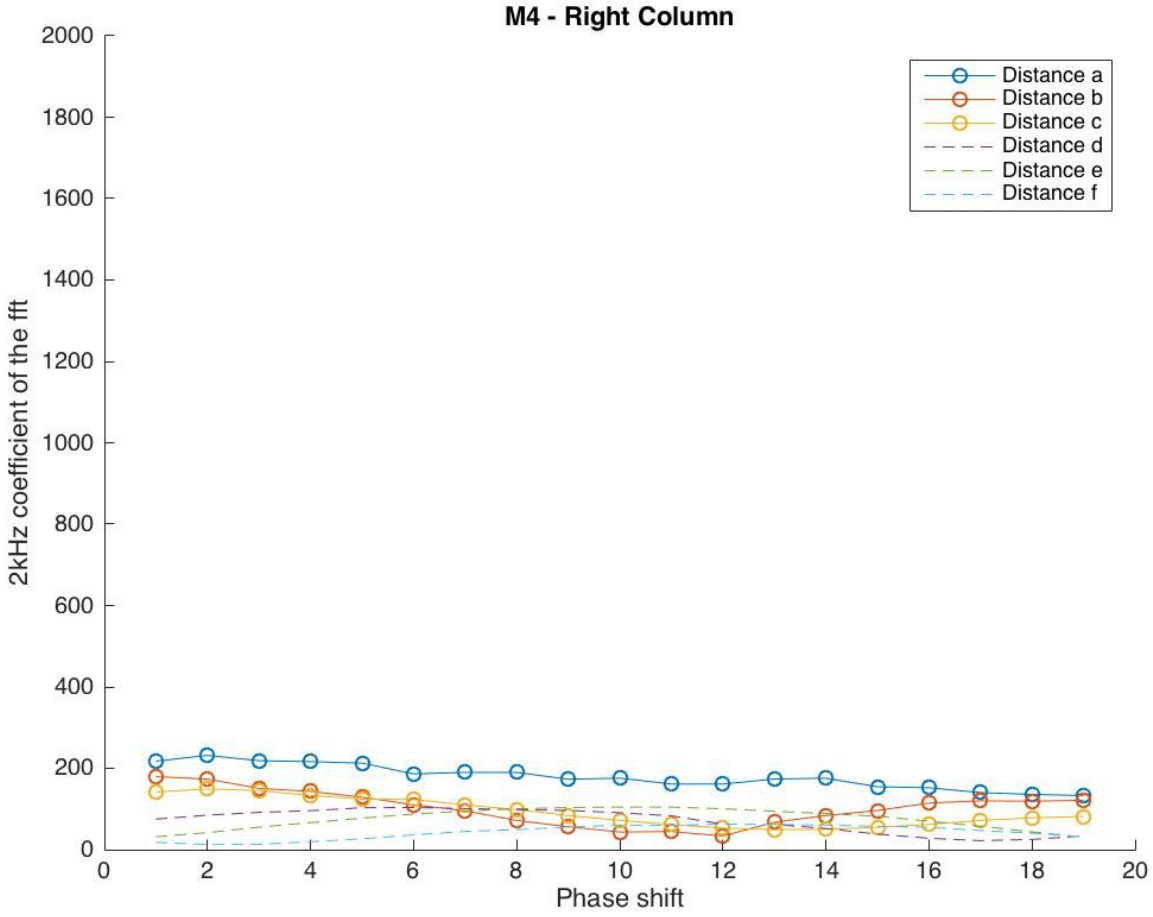

M2 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
200
400
600
800
1000
1200
1400
1600
0
2
4
6
8
10
12
14
16
18
20

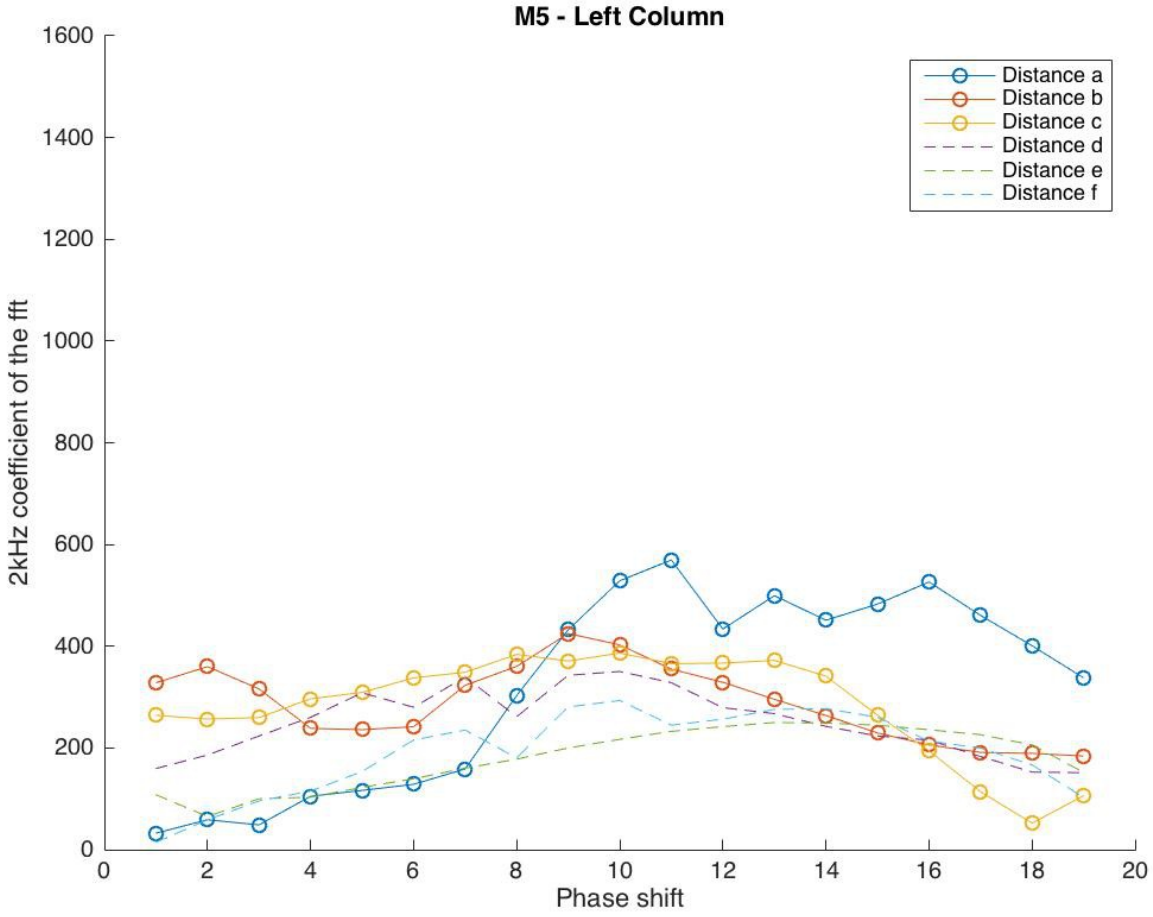

M2 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
2
4
6
8
10
12
14
16
18
20

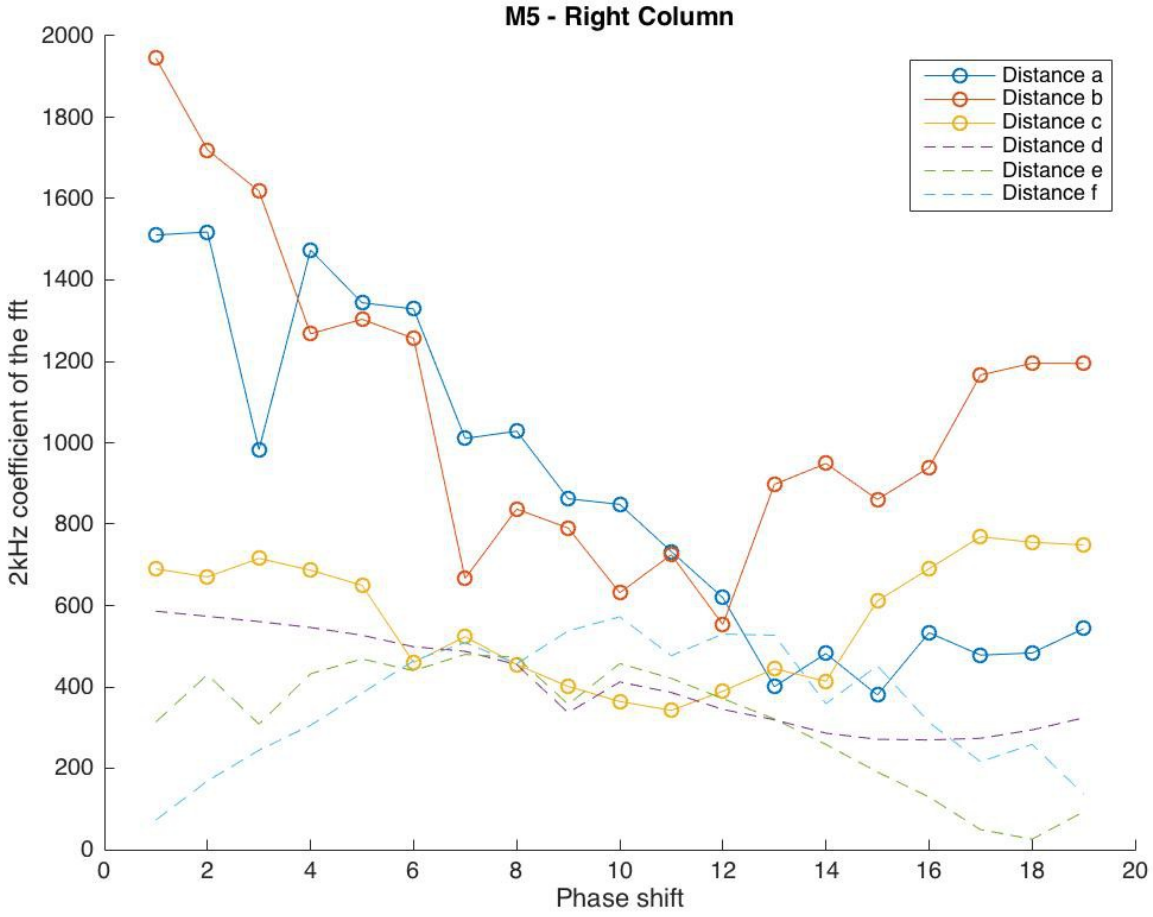

M3 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
200
400
600
800
1000
1200
1400
1600
0
2
4
6
8
10
12
14
16
18
20

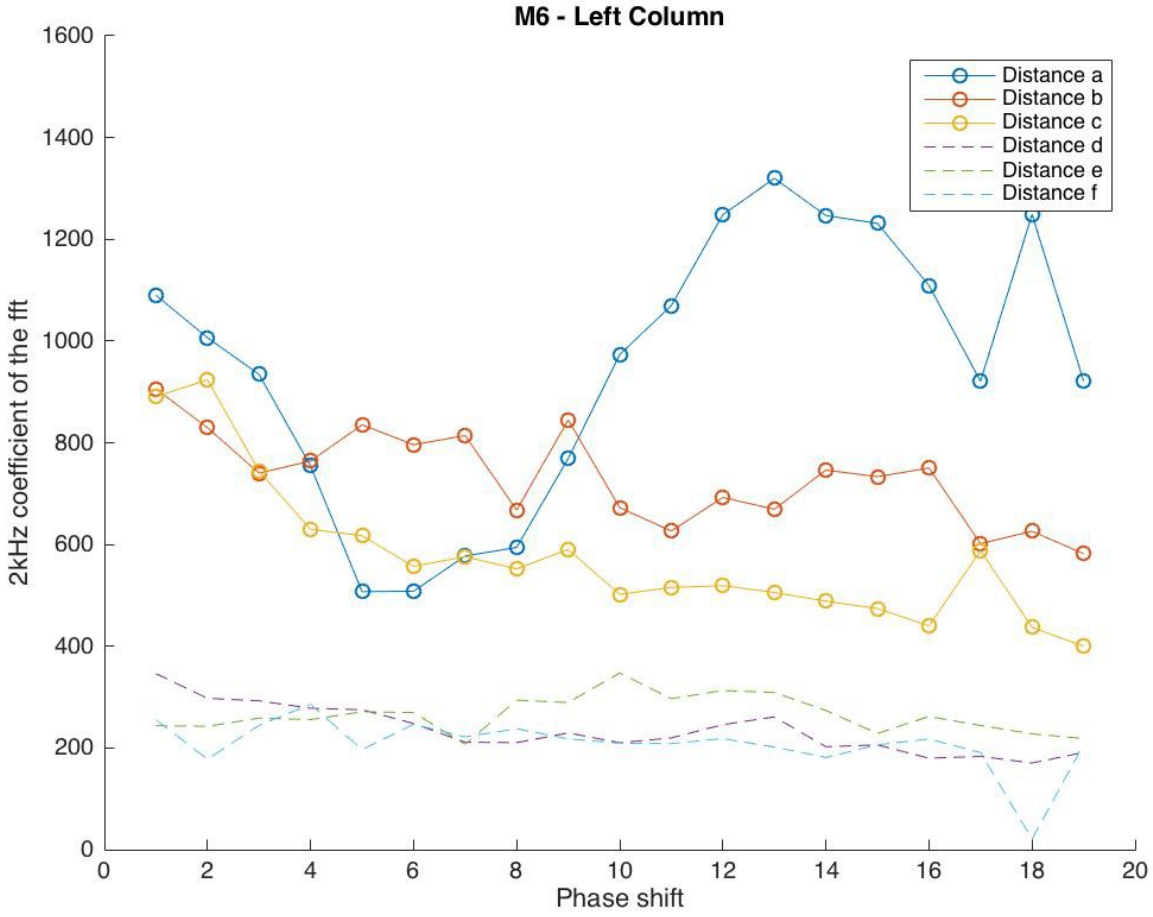

M3 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
2
4
6
8
10
12
14
16
18
20

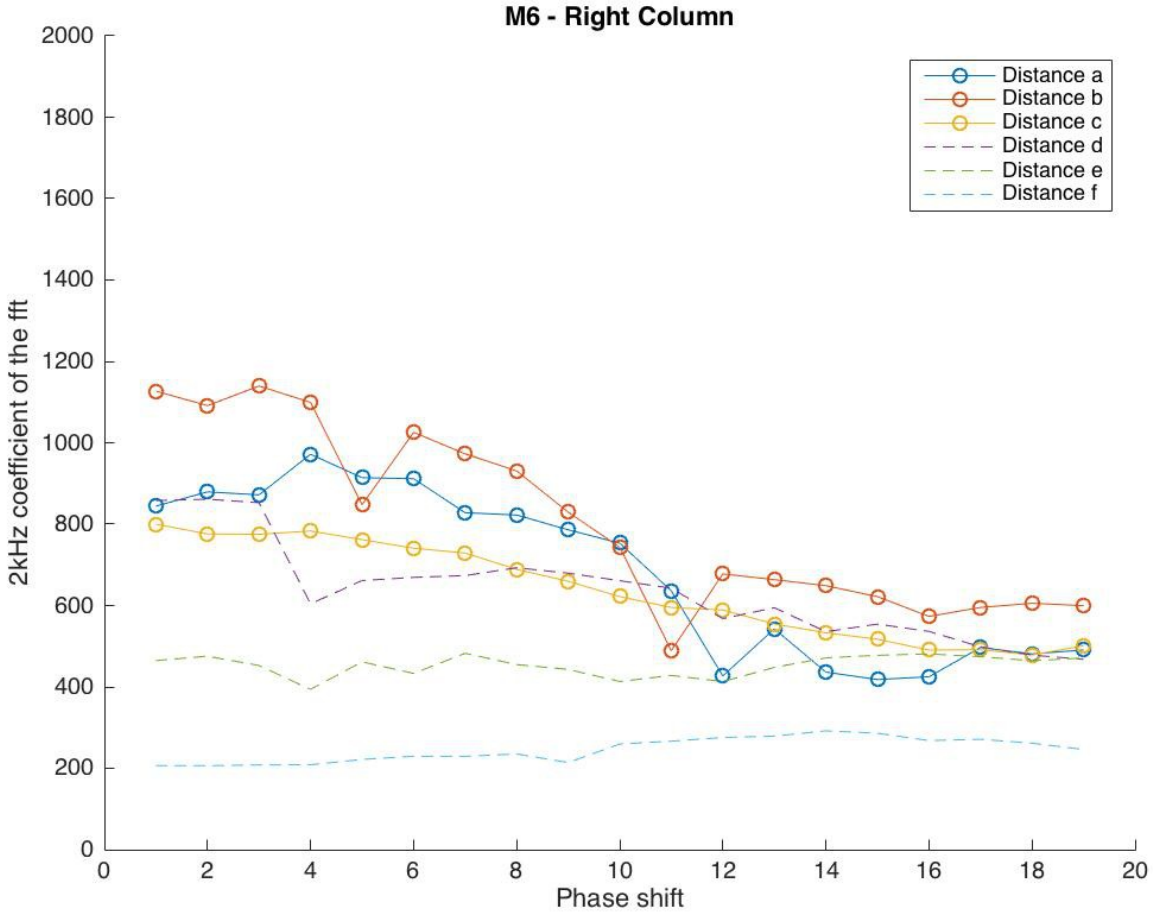

M4 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
200
400
600
800
1000
1200
1400
1600
0
2
4
6
8
10
12
14
16
18
20

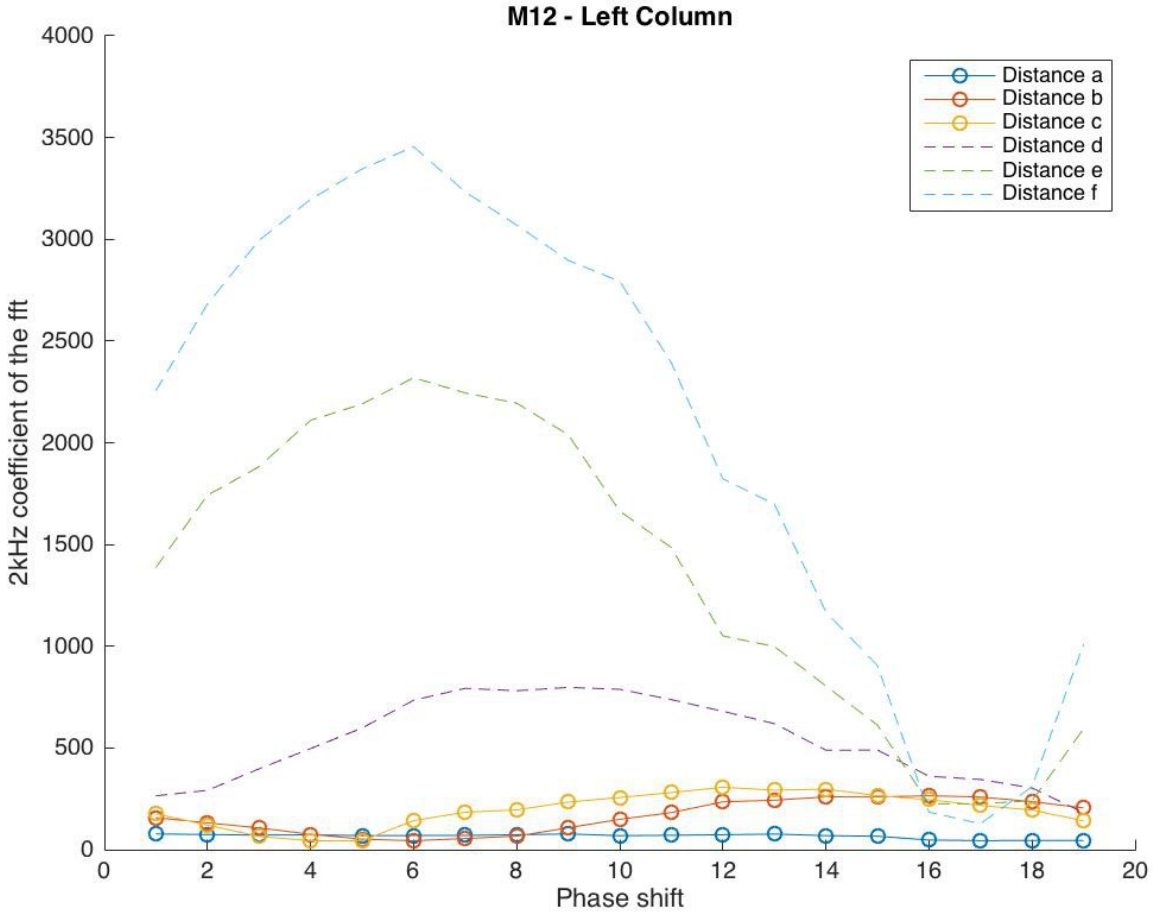

M4 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
2
4
6
8
10
12
14
16
18
20

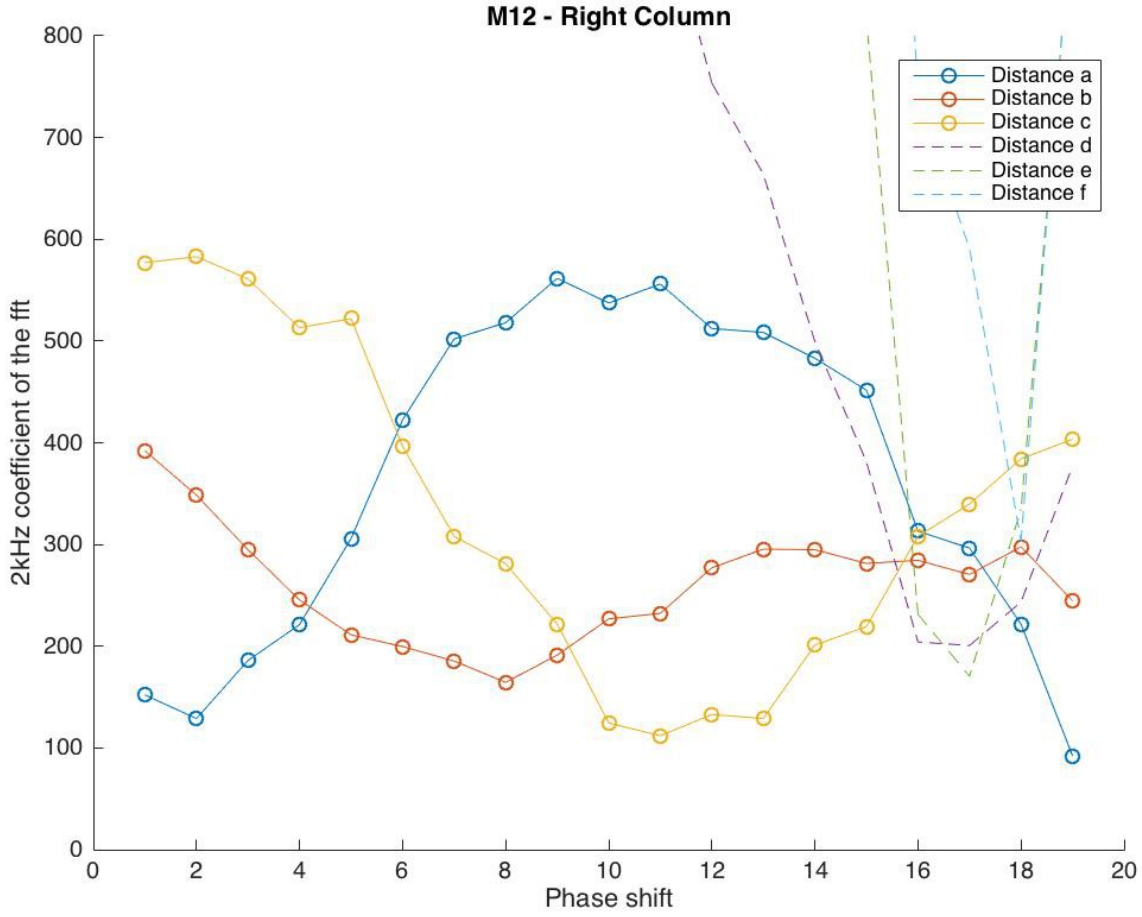

M5 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
200
400
600
800
1000
1200
1400
1600
0
2
4
6
8
10
12
14
16
18
20

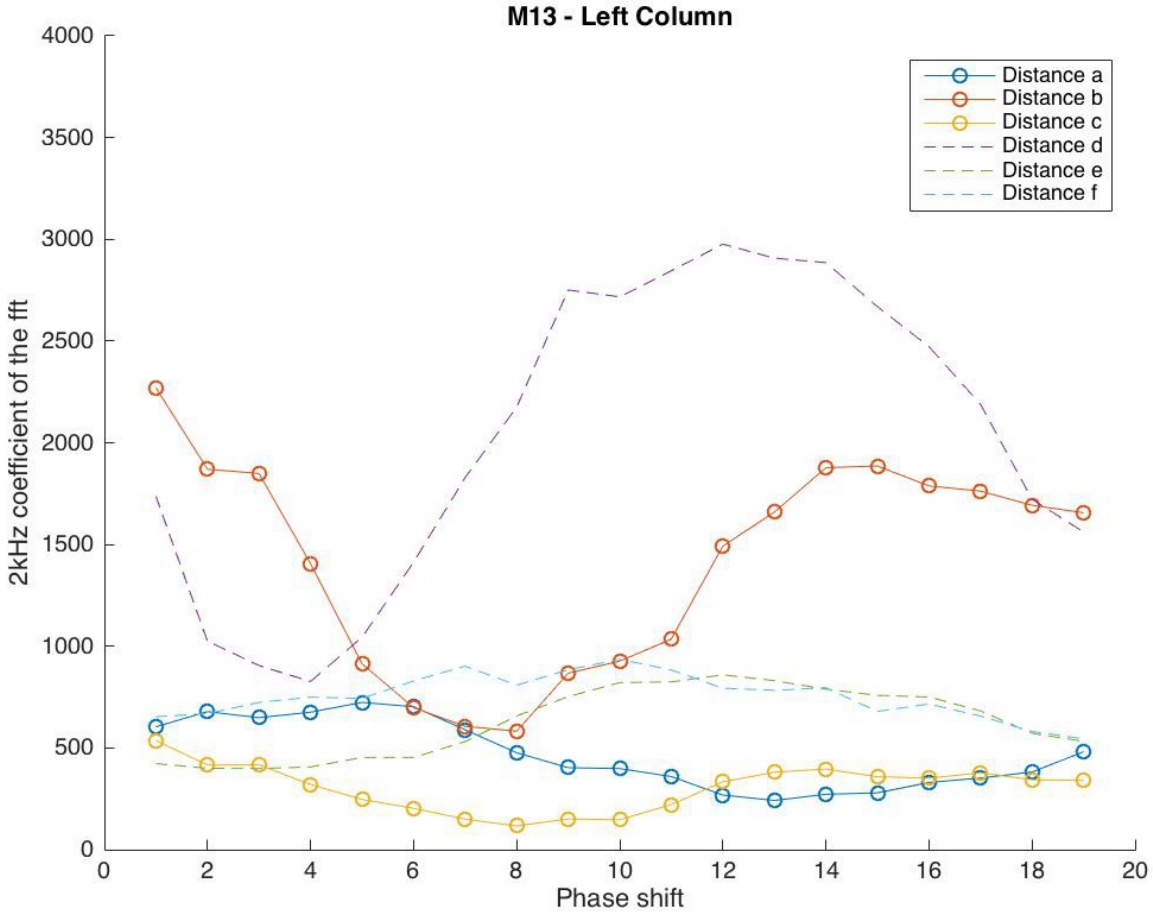

M5 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
2
4
6
8
10
12
14
16
18
20

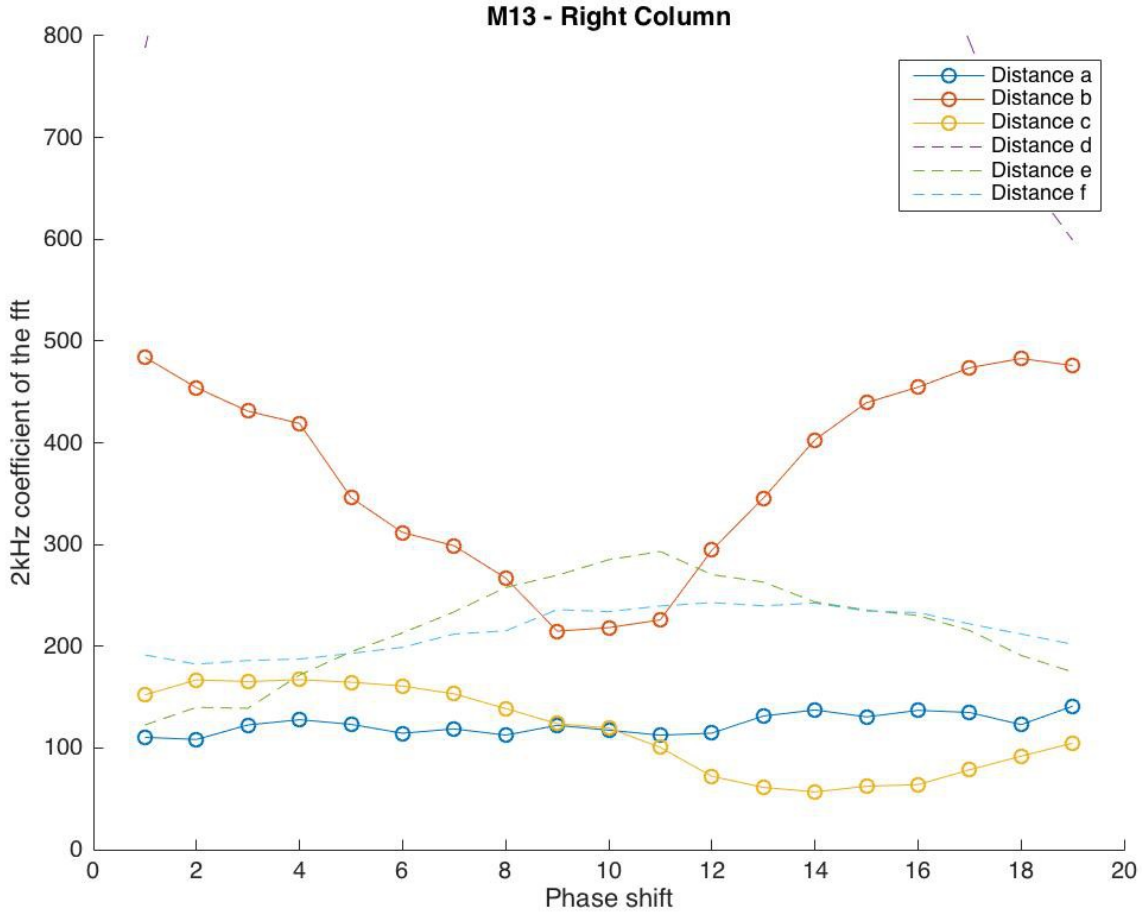

M6 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
200
400
600
800
1000
1200
1400
1600
0
2
4
6
8
10
12
14
16
18
20

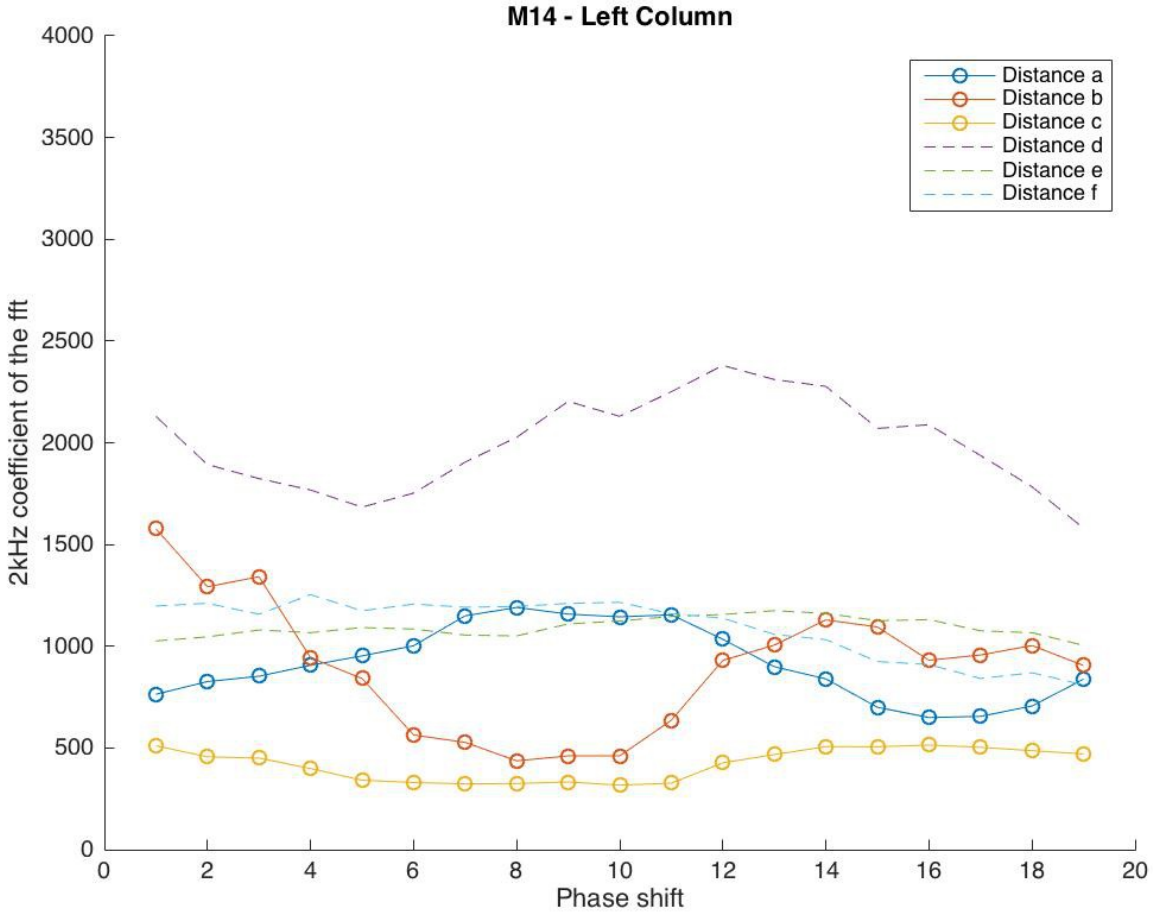

M6 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
2
4
6
8
10
12
14
16
18
20

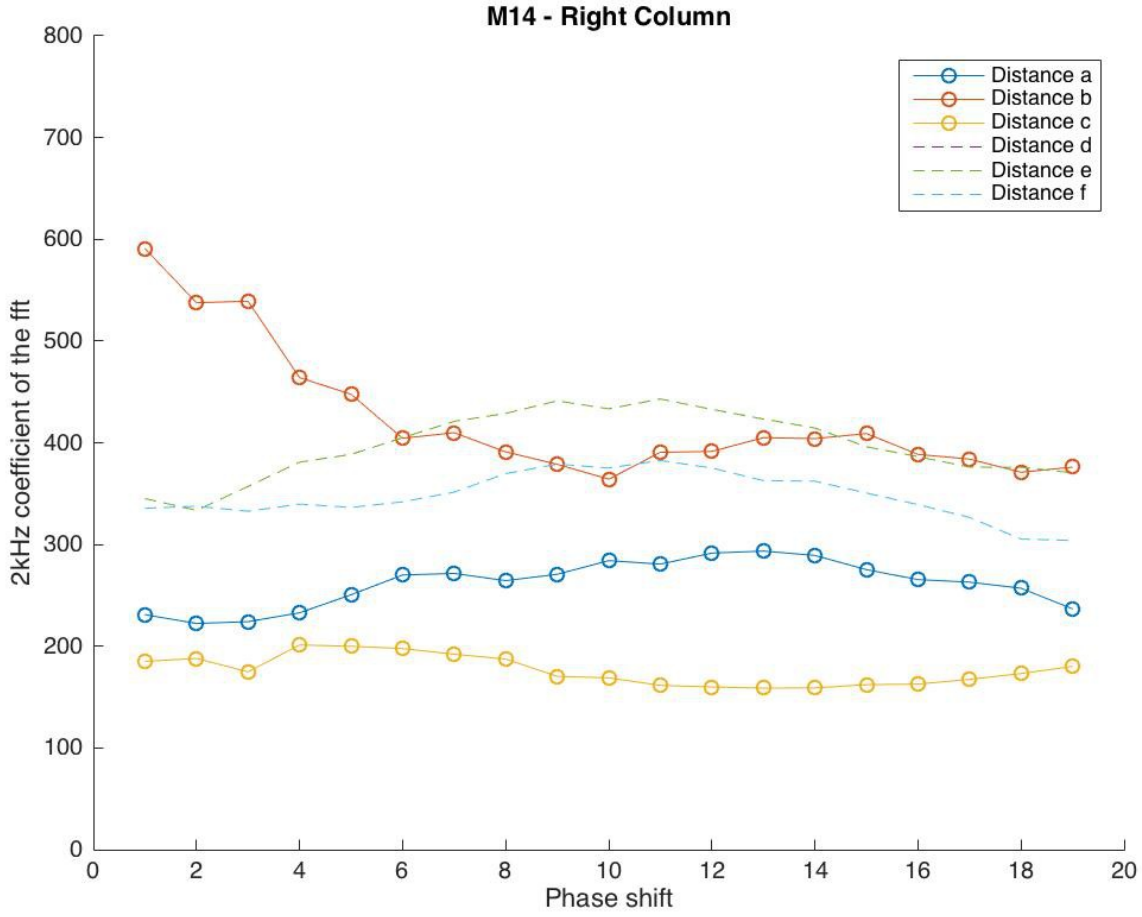

M12 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
3000
3500
4000
0
2
4
6
8
10
12
14
16
18
20

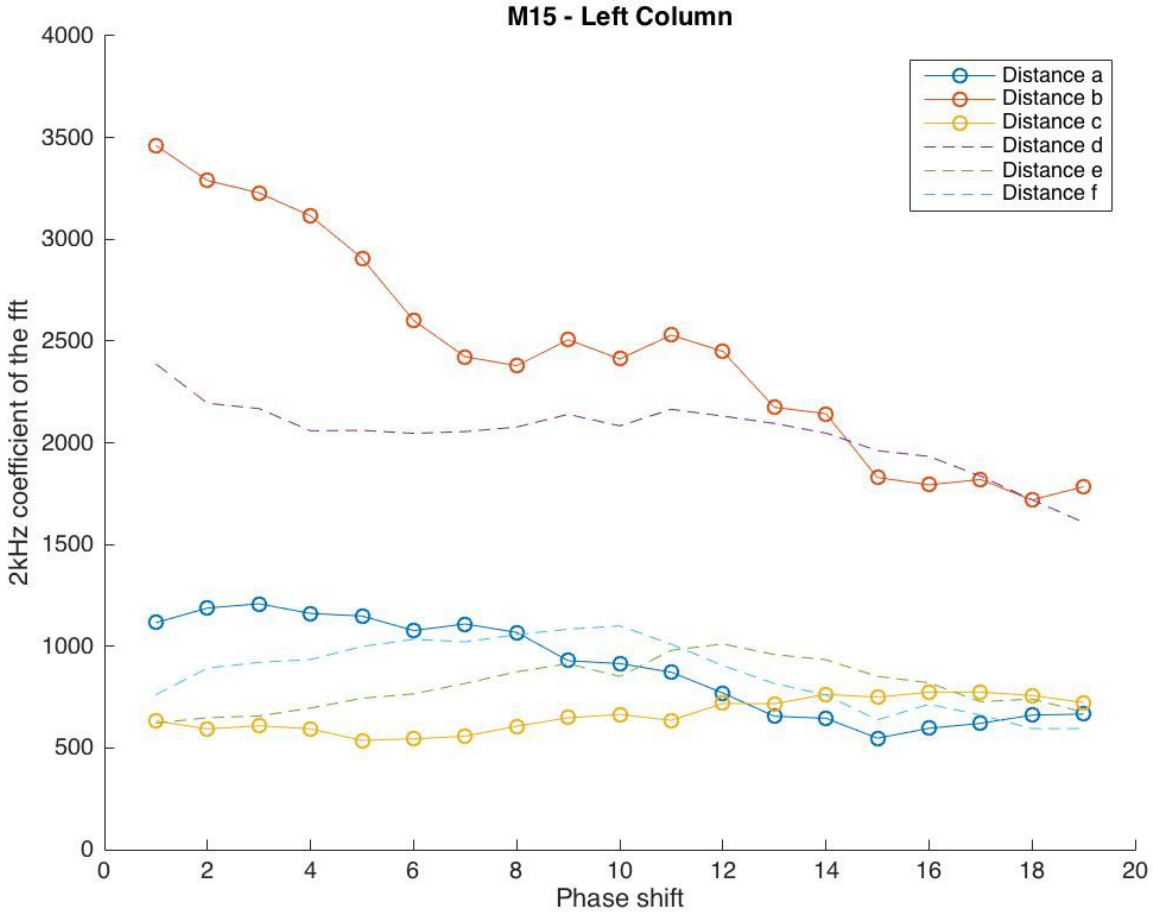

M12 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
100
200
300
400
500
600
700
800
0
2
4
6
8
10
12
14
16
18
20

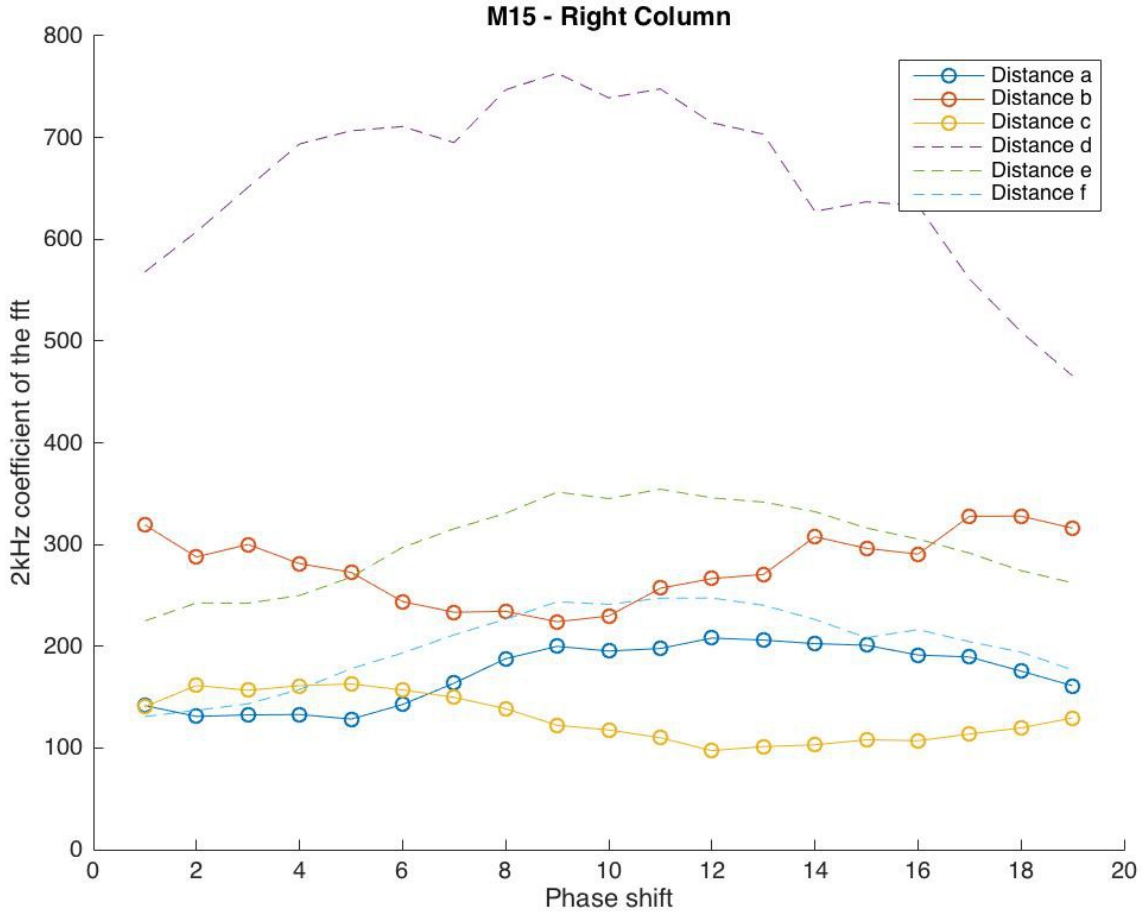

M13 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
3000
3500
4000
0
2
4
6
8
10
12
14
16
18
20

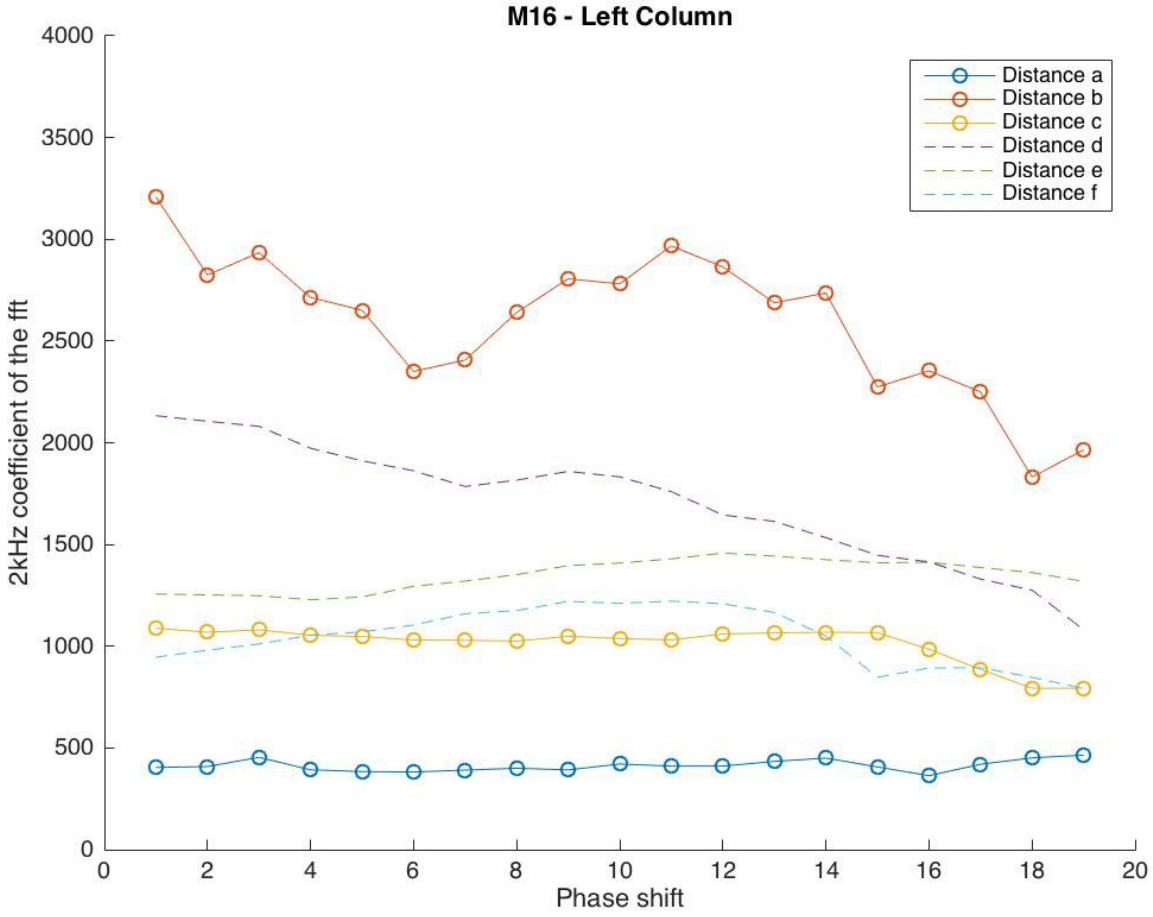

M13 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
100
200
300
400
500
600
700
800
0
2
4
6
8
10
12
14
16
18
20

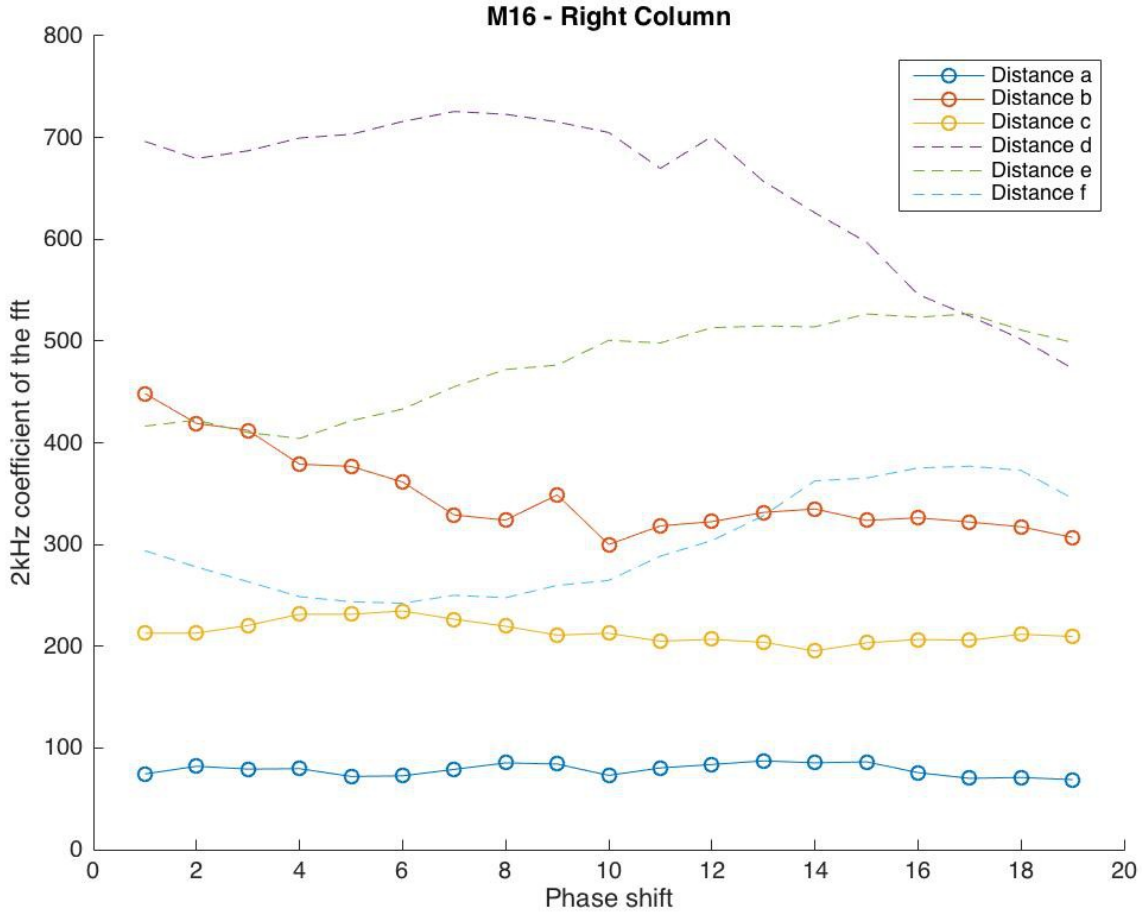

M14 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
3000
3500
4000
0
2
4
6
8
10
12
14
16
18
20

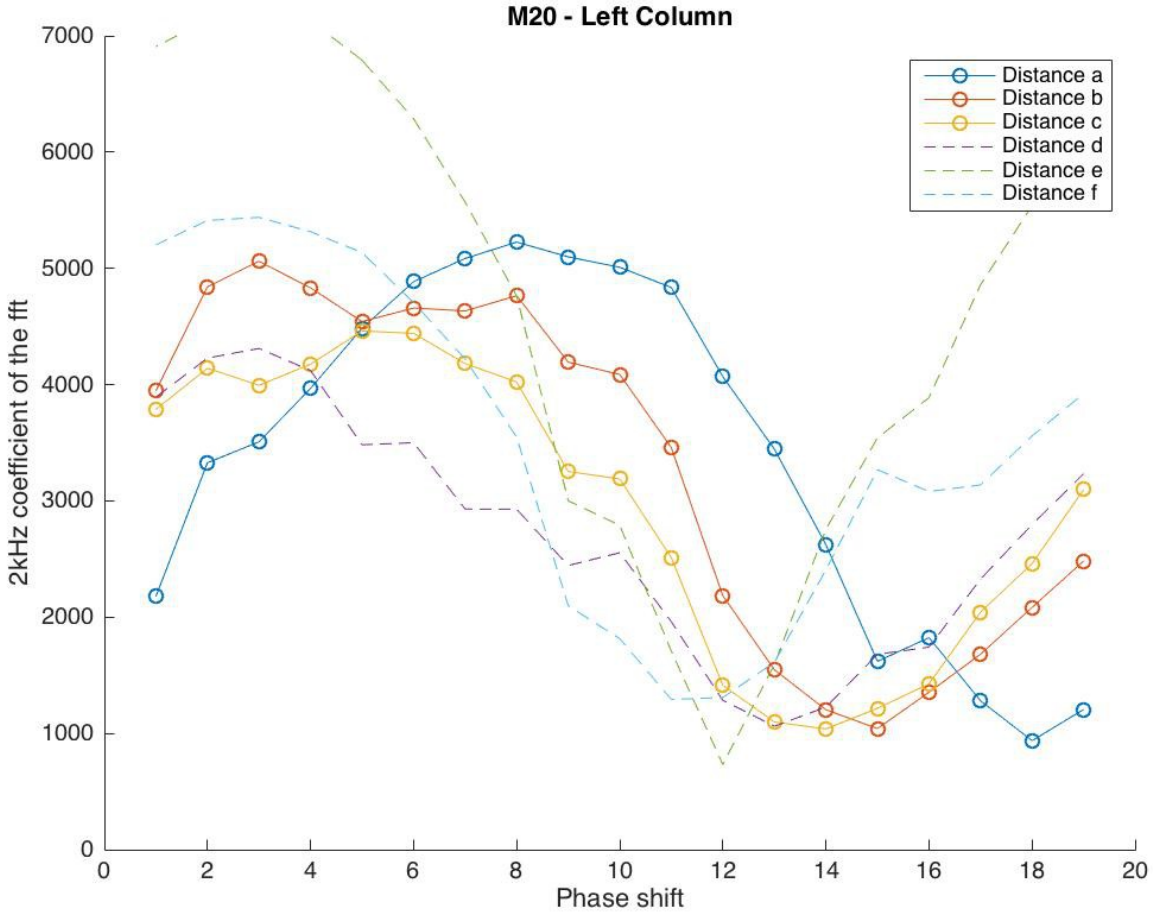

M14 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
100
200
300
400
500
600
700
800
0
2
4
6
8
10
12
14
16
18
20

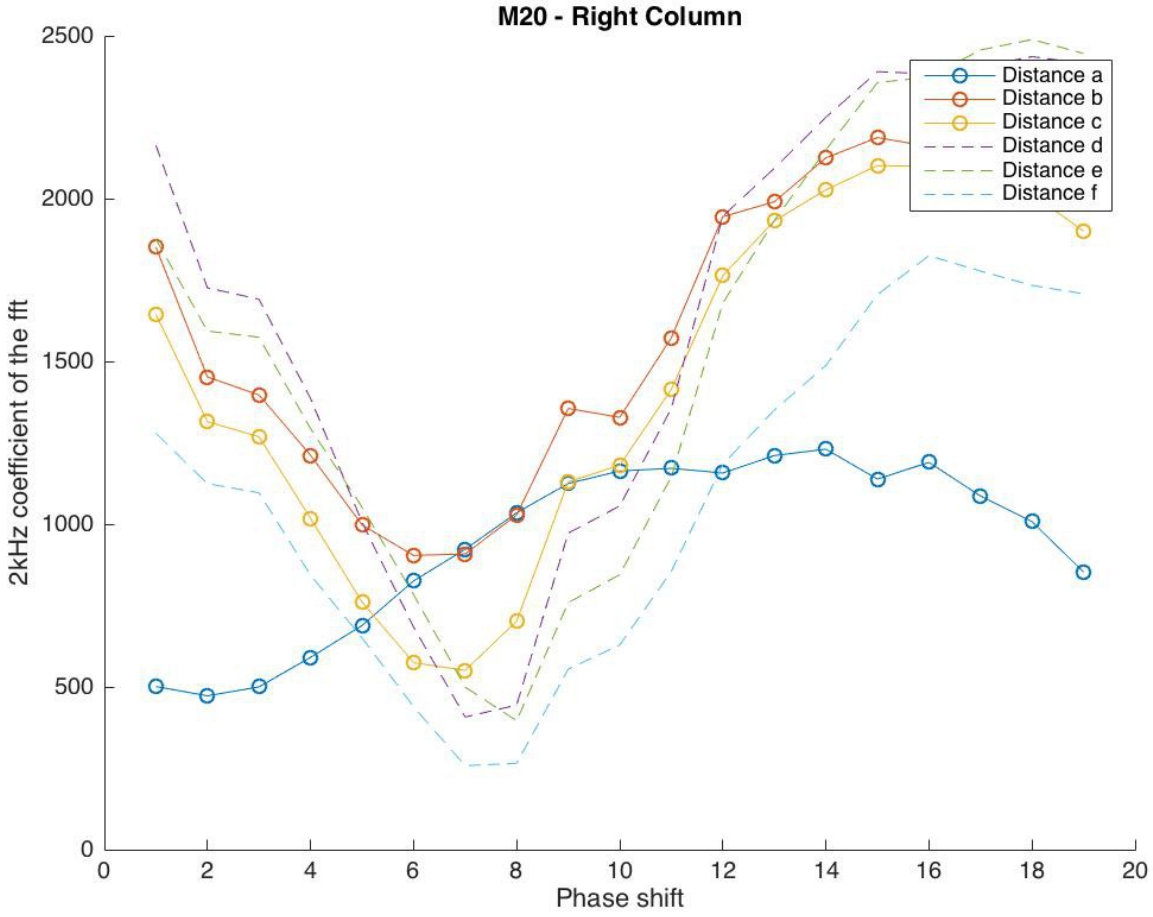

M15 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
3000
3500
4000
0
2
4
6
8
10
12
14
16
18
20

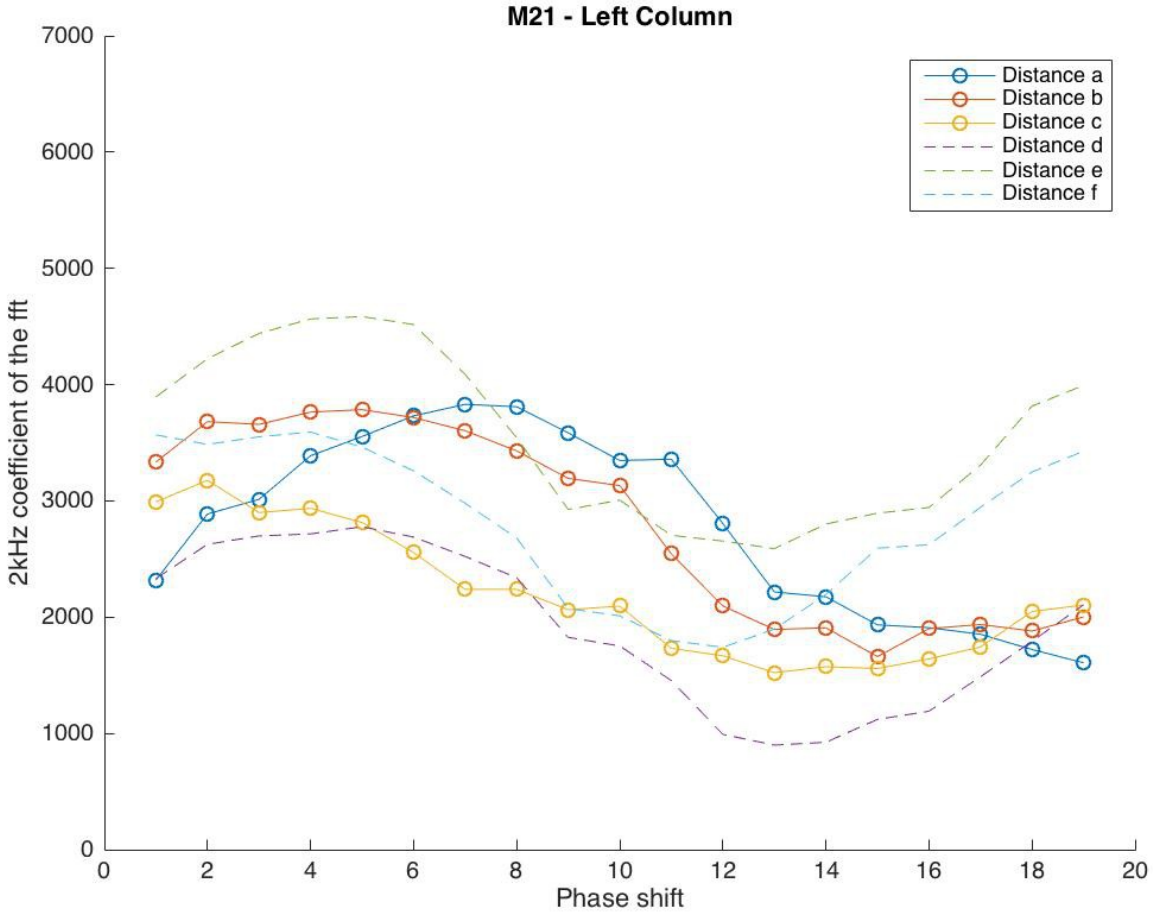

M15 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
100
200
300
400
500
600
700
800
0
2
4
6
8
10
12
14
16
18
20

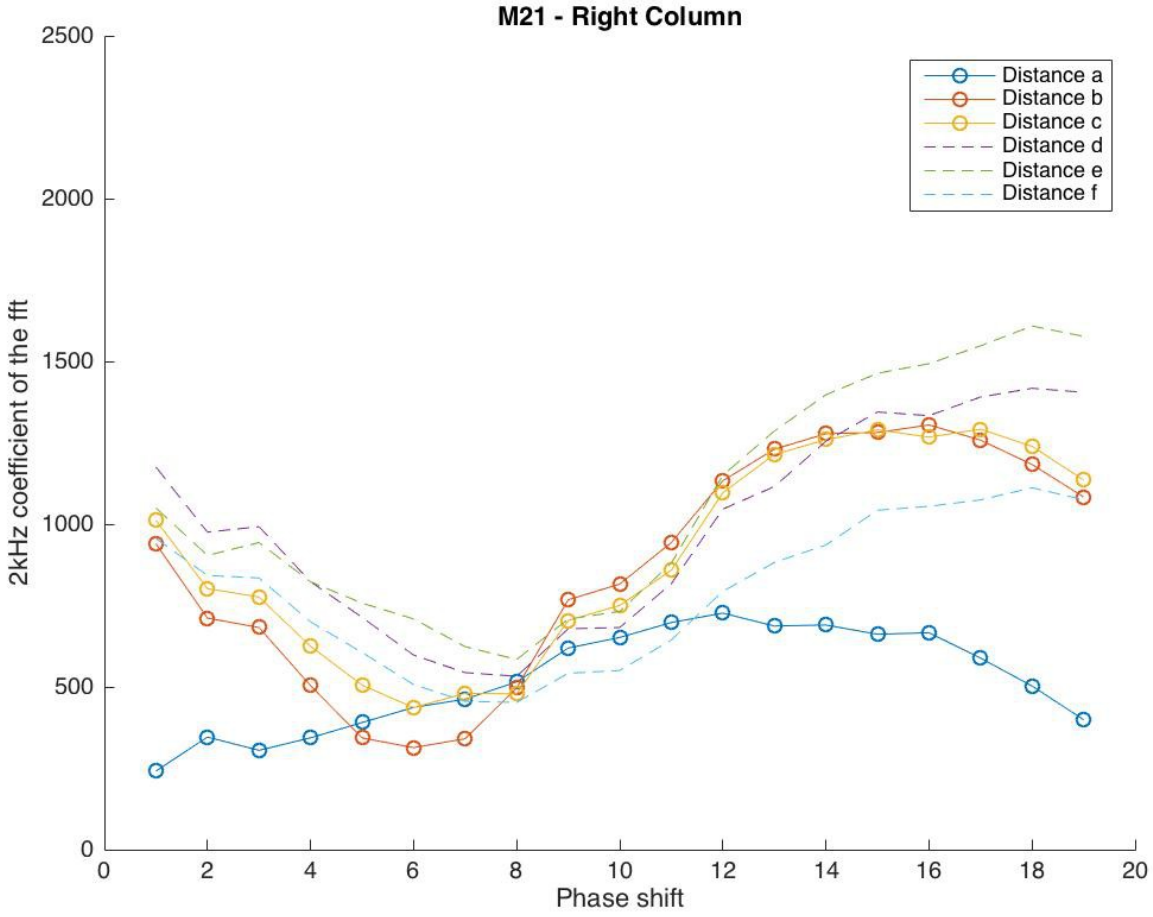

M16 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
3000
3500
4000
0
2
4
6
8
10
12
14
16
18
20

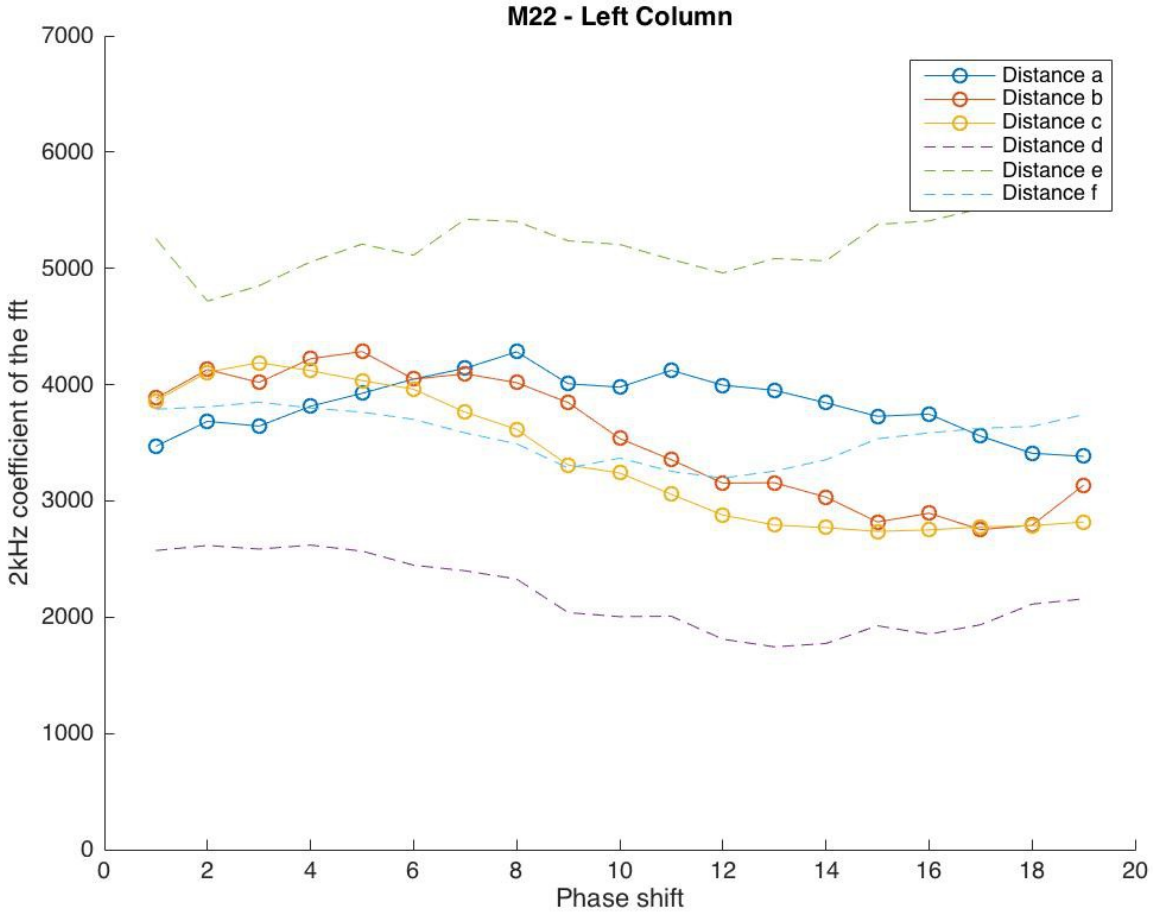

M16 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
100
200
300
400
500
600
700
800
0
2
4
6
8
10
12
14
16
18
20

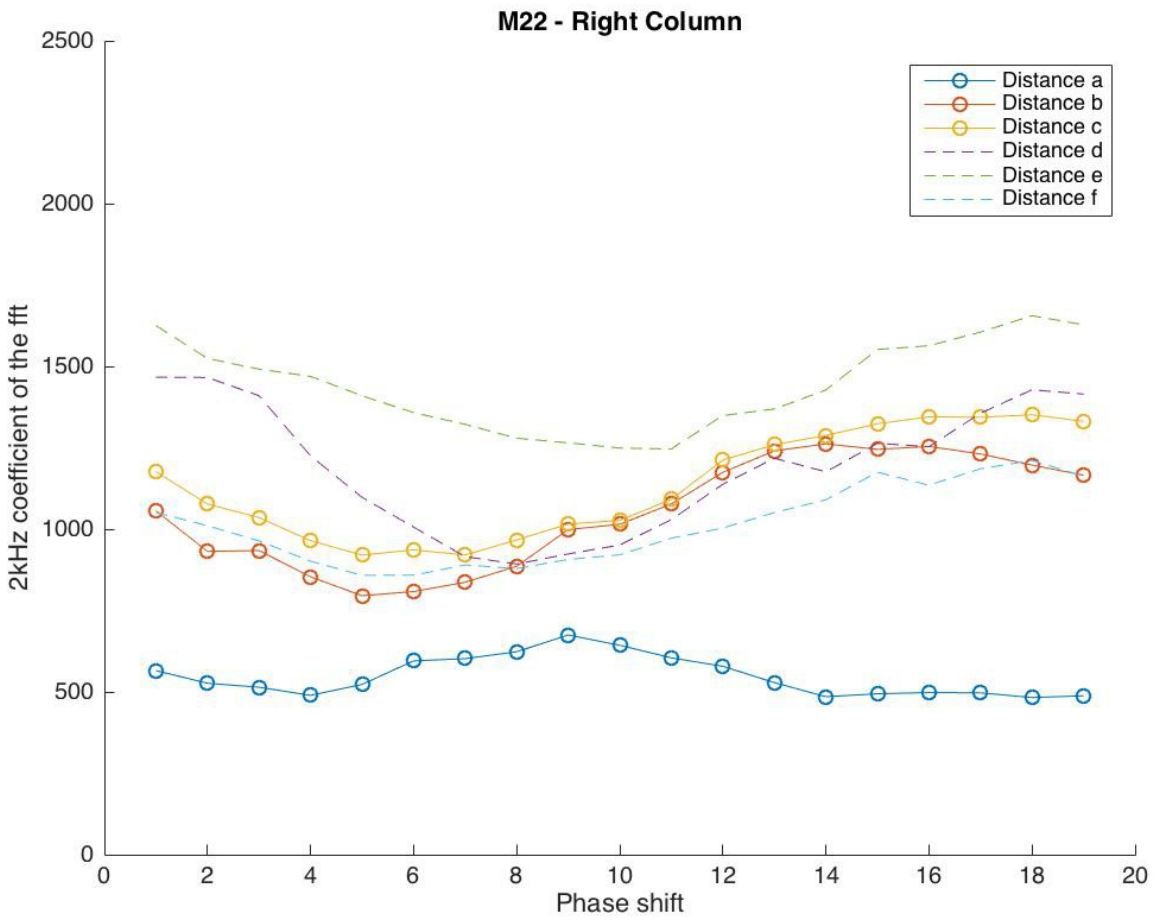

M20 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
1000
2000
3000
4000
5000
6000
7000
0
2
4
6
8
10
12
14
16
18
20

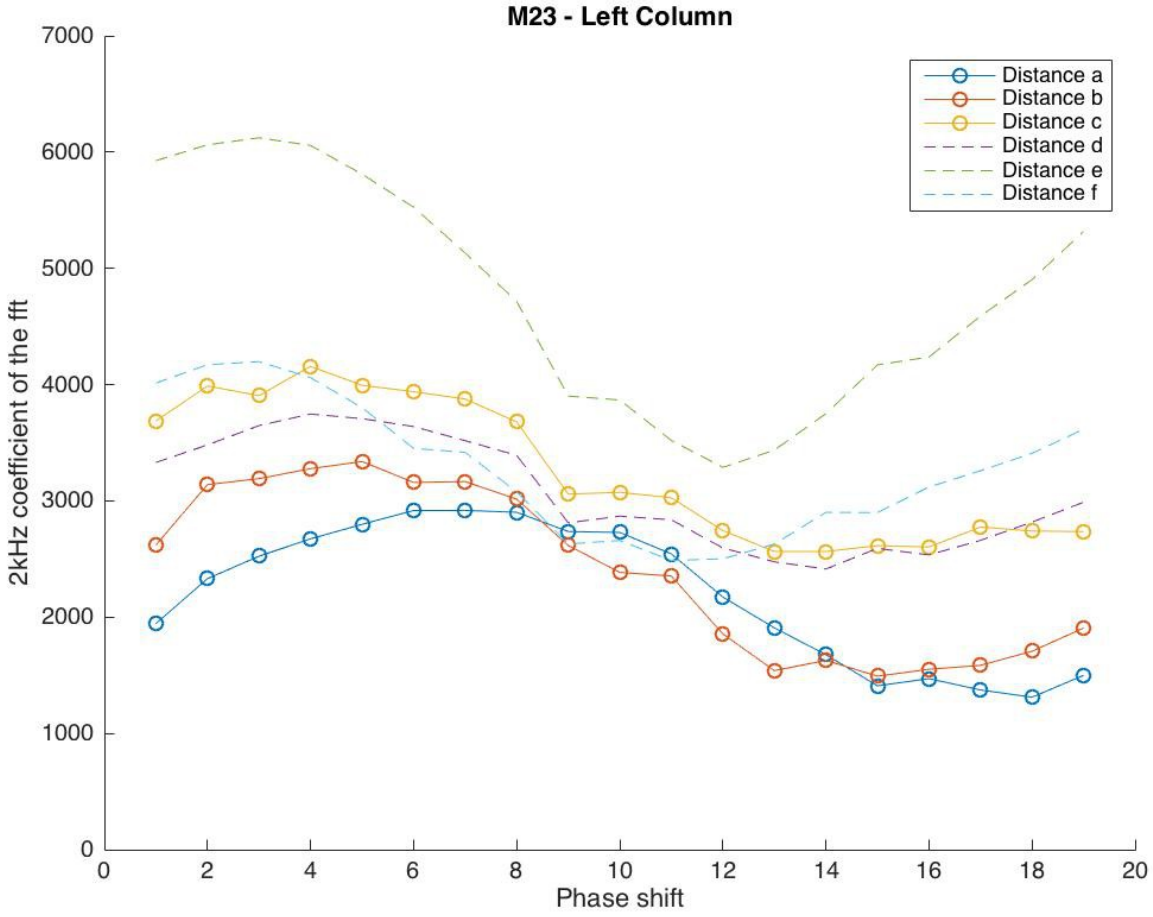

M20 - Right Column
2kHz coefficient of the fft
Phase shift
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
0
500
1000
1500
2000
2500
0
2
4
6
8
10
12
14
16
18
20

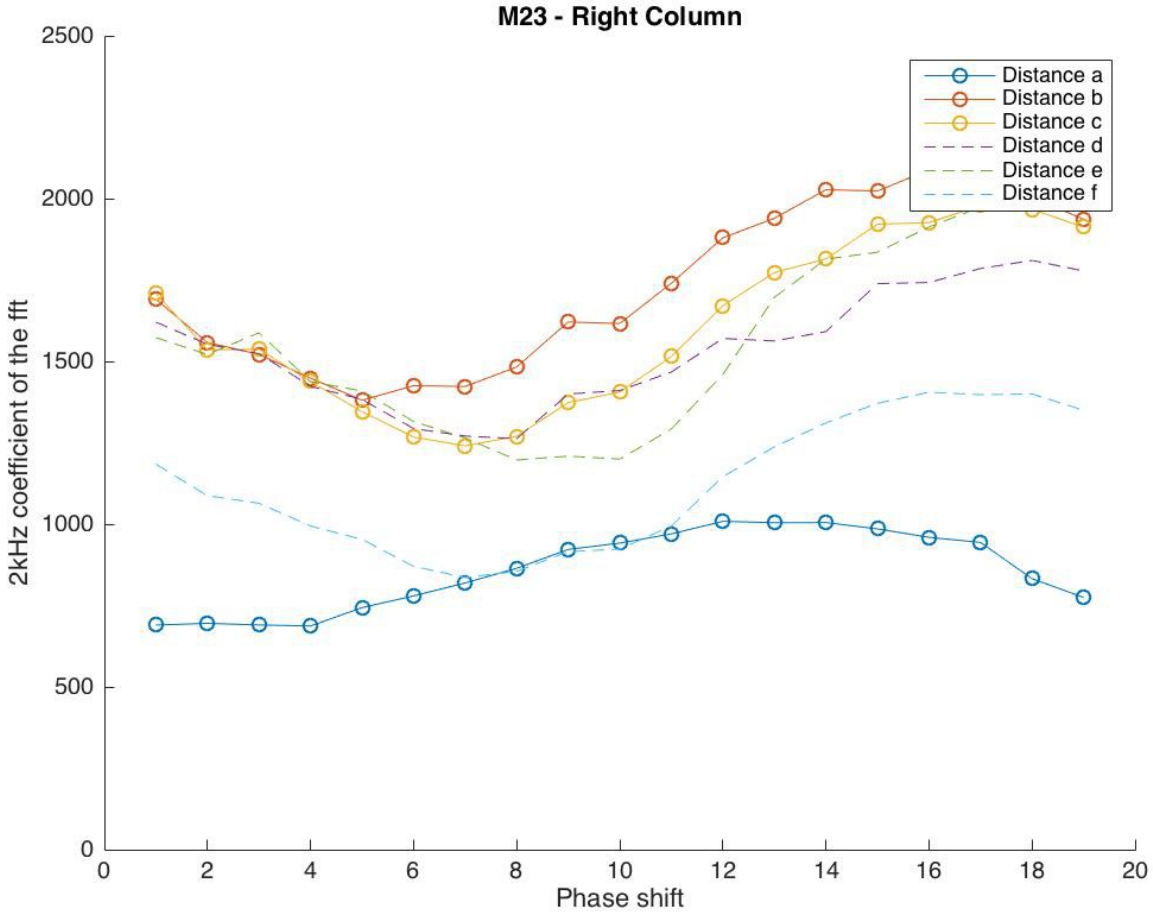

M21 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
1000
2000
3000
4000
5000
6000
7000
0
2
4
6
8
10
12
14
16
18
20

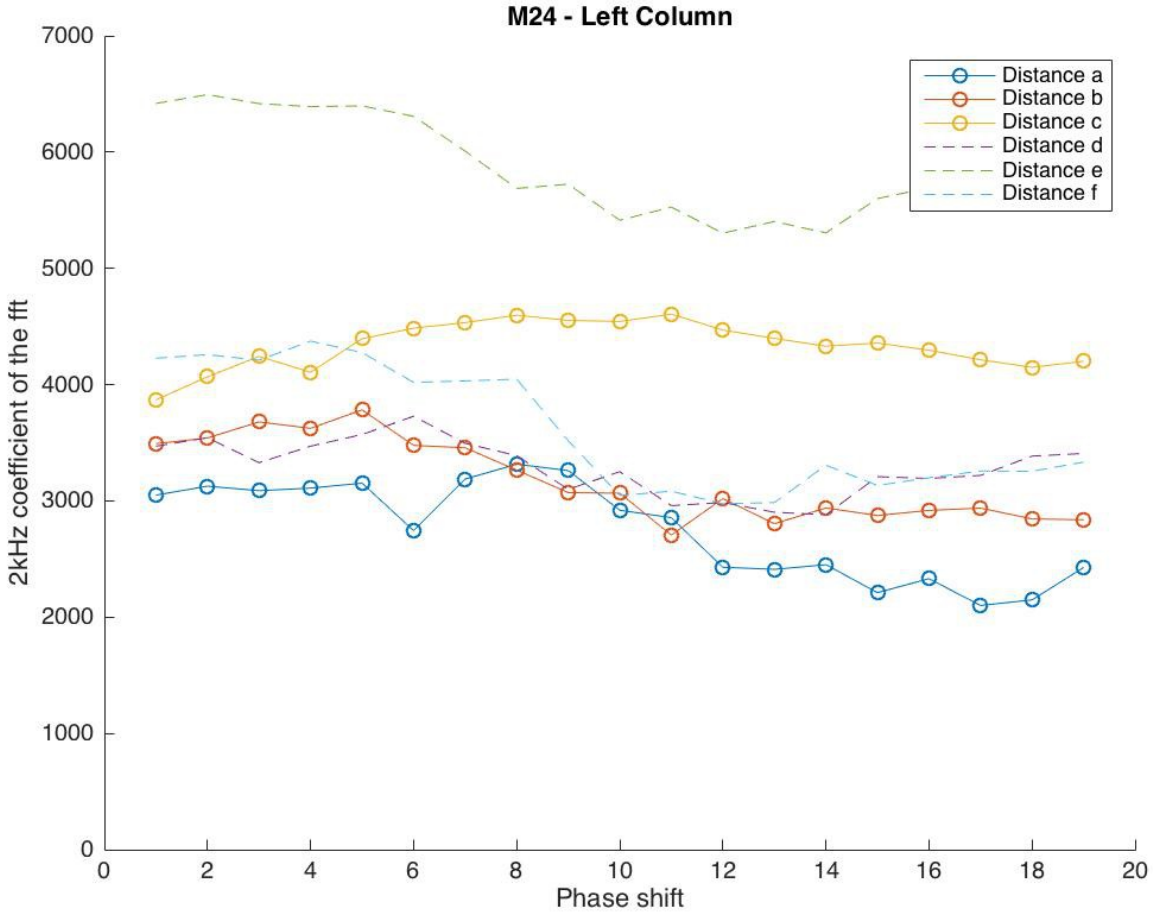

M21 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
0
2
4
6
8
10
12
14
16
18
20

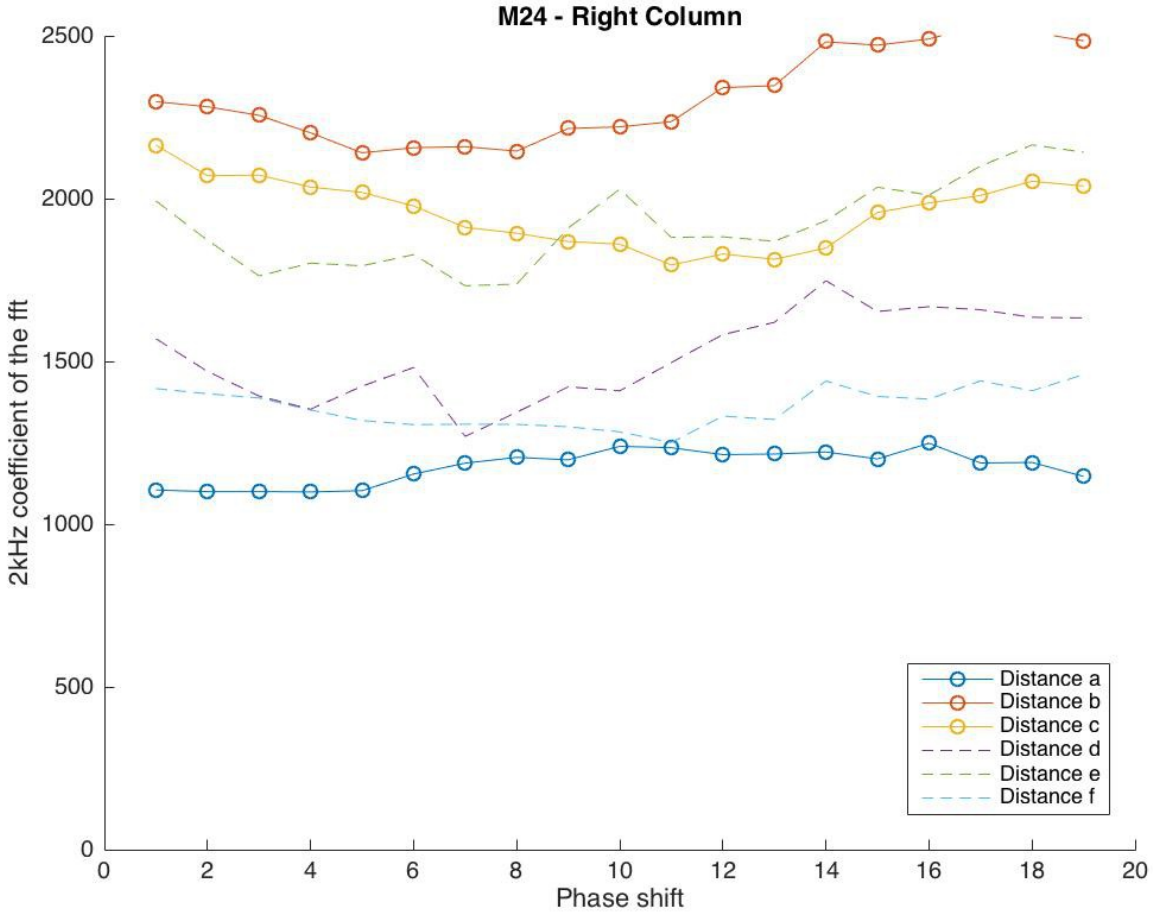

M22 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
1000
2000
3000
4000
5000
6000
7000
0
2
4
6
8
10
12
14
16
18
20

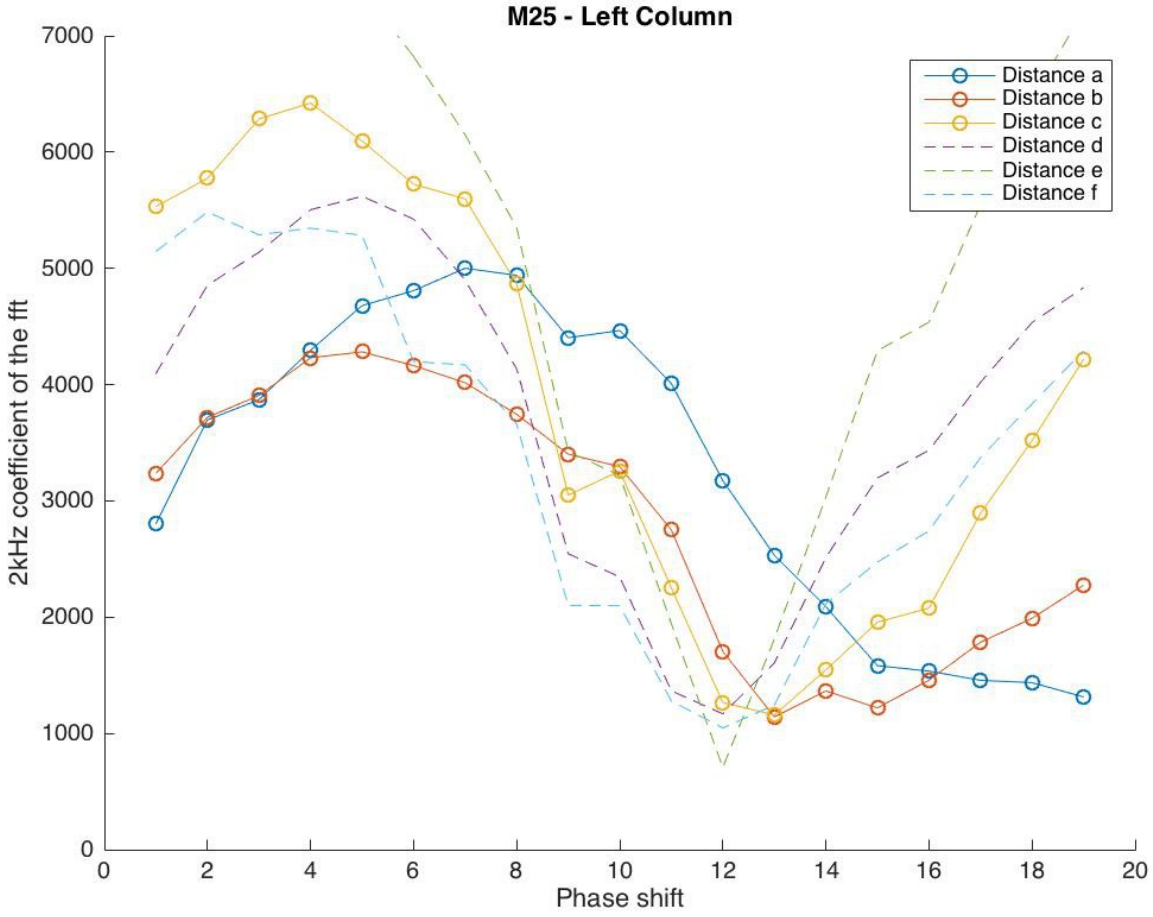

M22 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
2500
2000
1500
1000
500
0
0
2
4
6
8
10
12
14
16
18
20

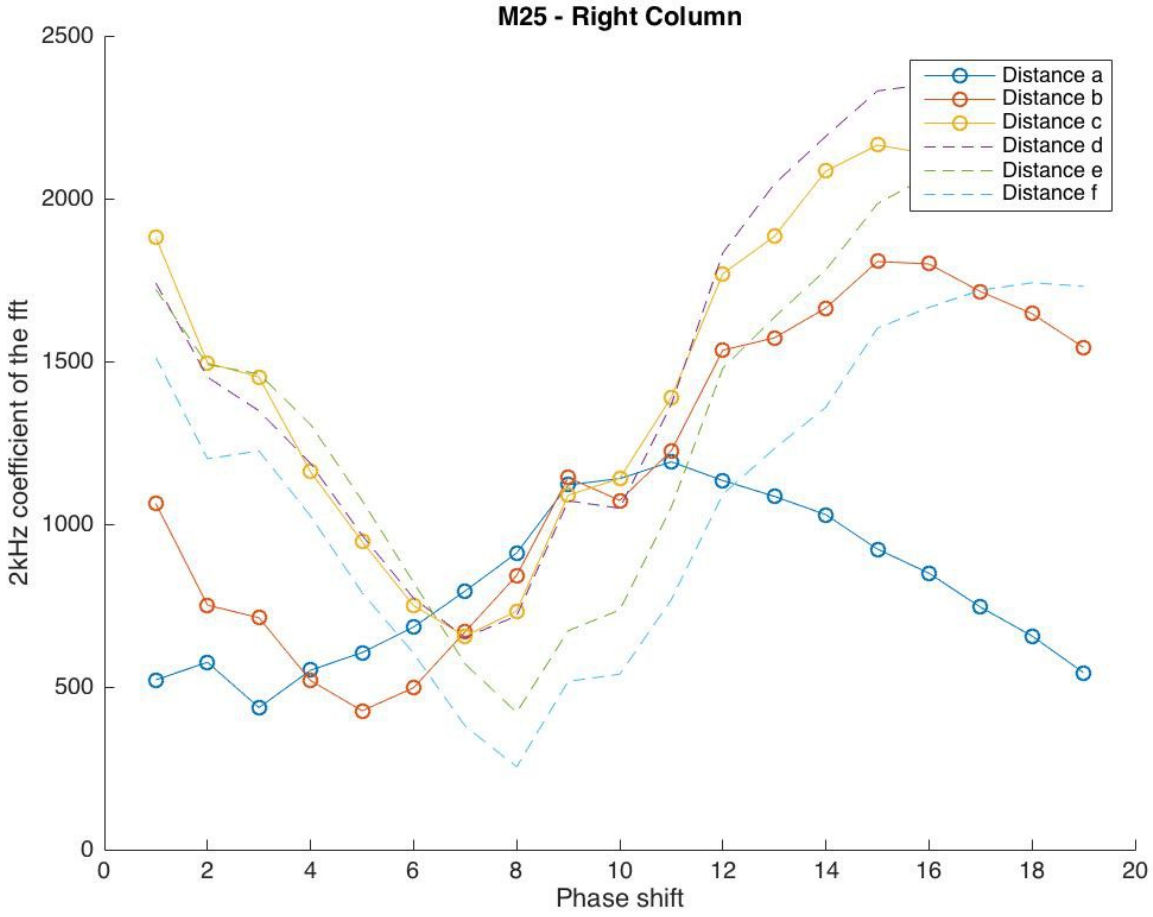

M23 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
1000
2000
3000
4000
5000
6000
7000
0
2
4
6
8
10
12
14
16
18
20

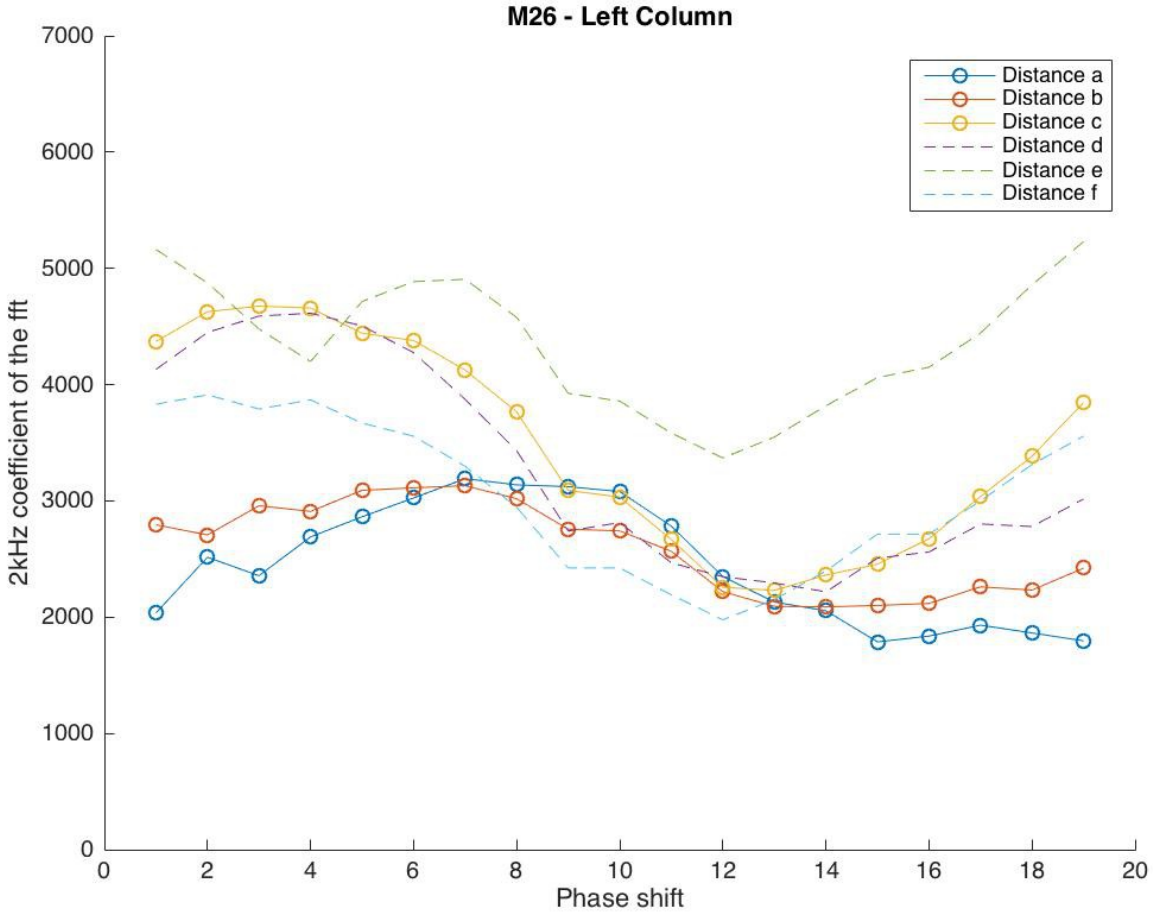

M23 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
0
2
4
6
8
10
12
14
16
18
20

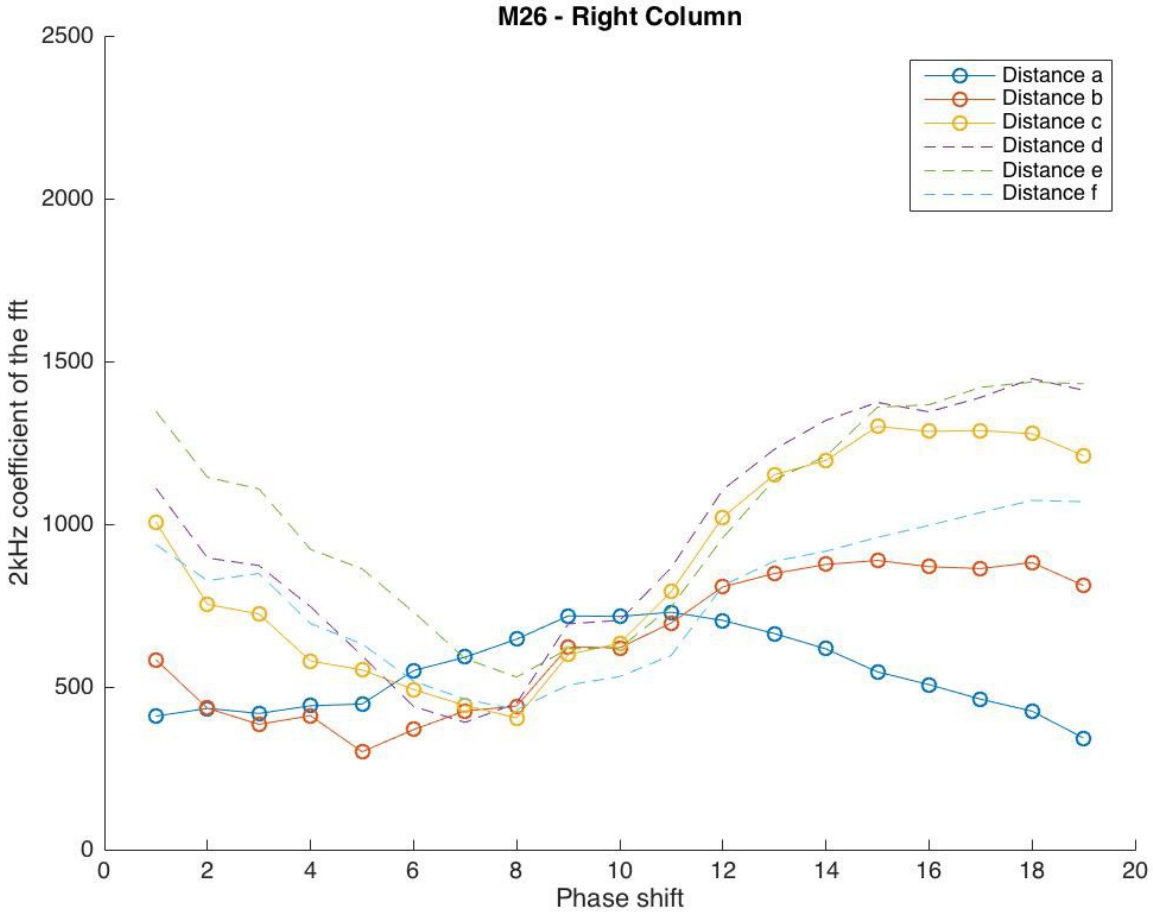

M24 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
1000
2000
3000
4000
5000
6000
7000
0
2
4
6
8
10
12
14
16
18
20

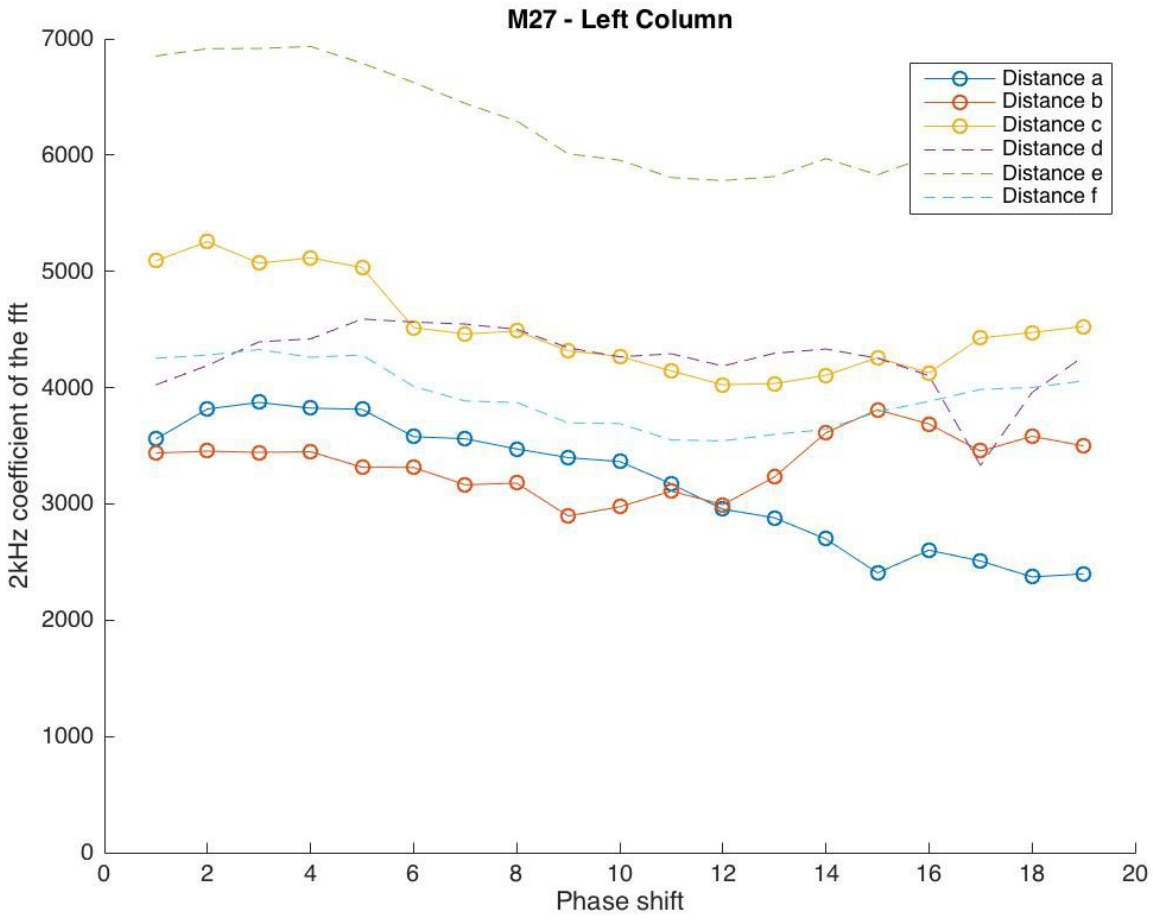

M24 - Right Column
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
0
2
4
6
8
10
12
14
16
18
20
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f

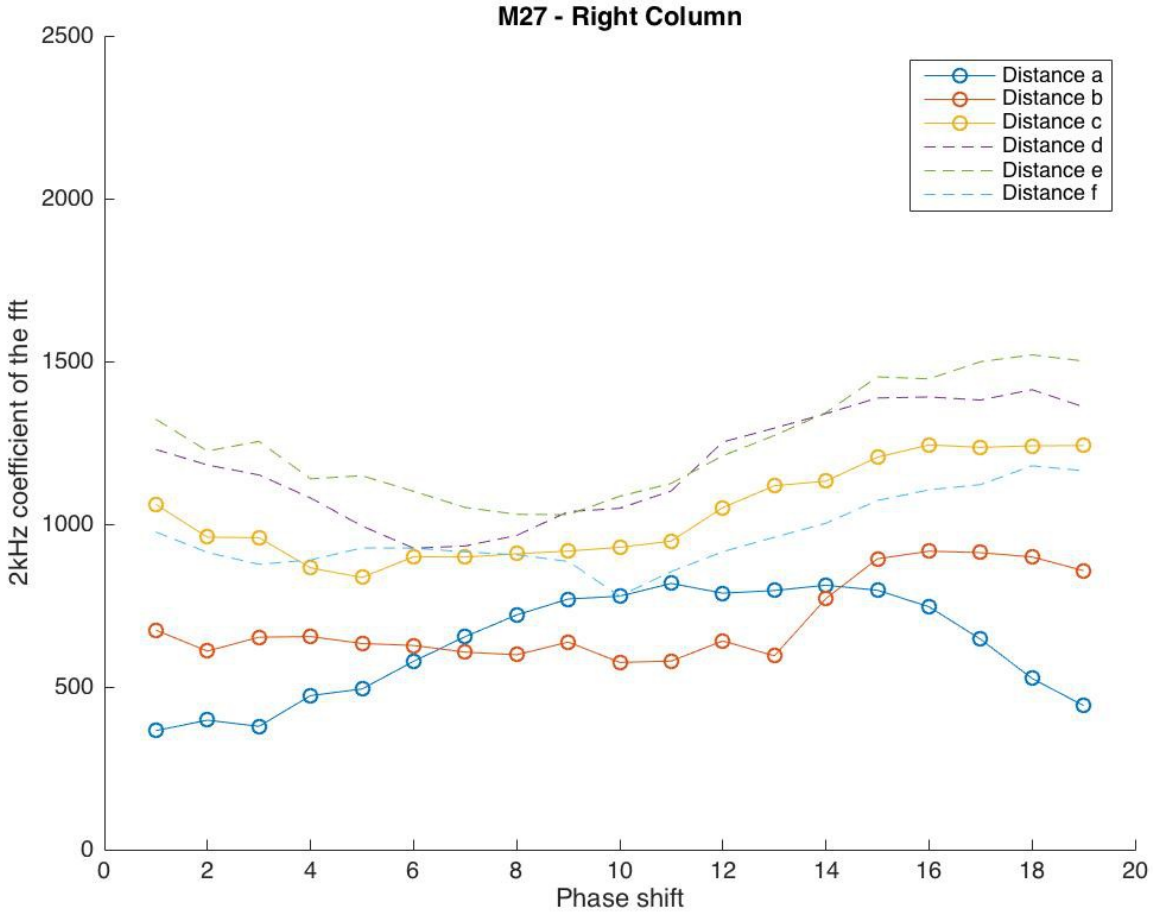

M25 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
1000
2000
3000
4000
5000
6000
7000
0
2
4
6
8
10
12
14
16
18
20

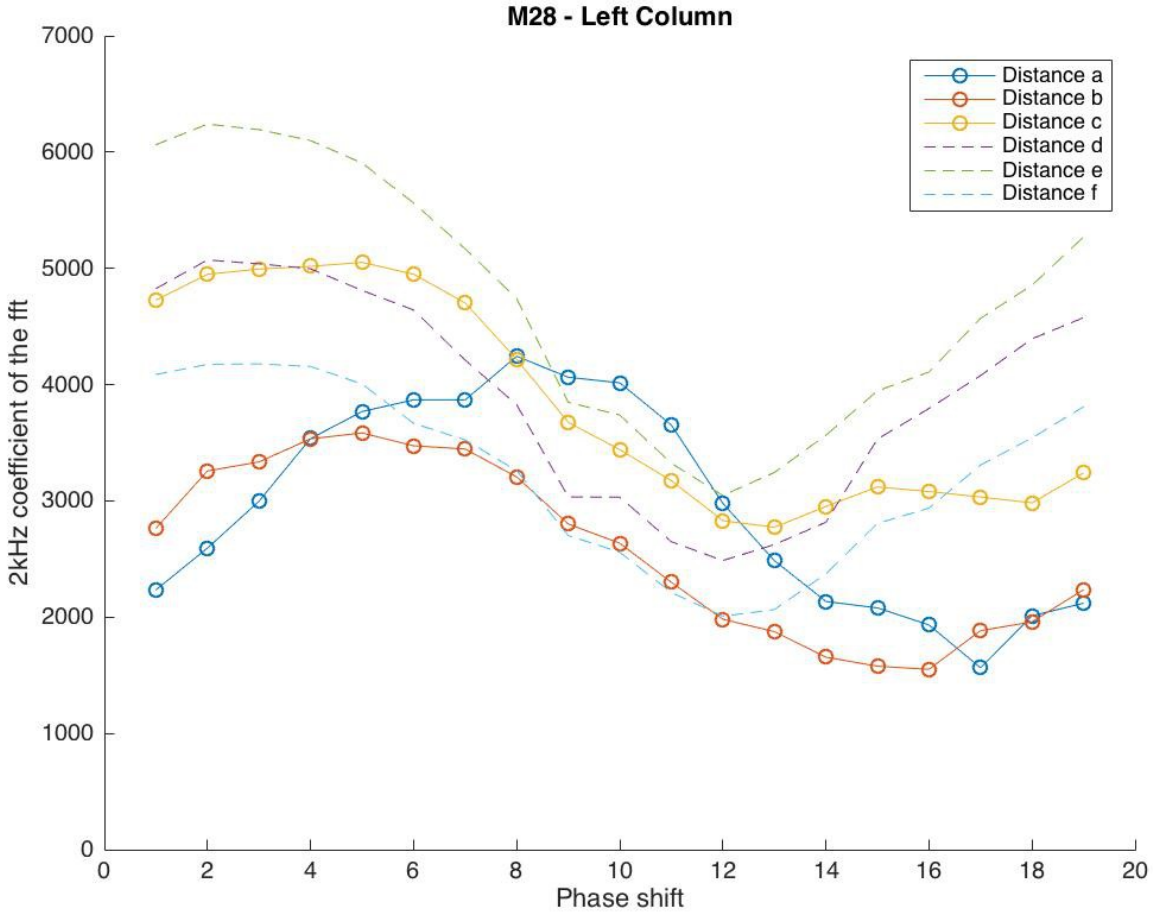

M25 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
0
2
4
6
8
10
12
14
16
18
20

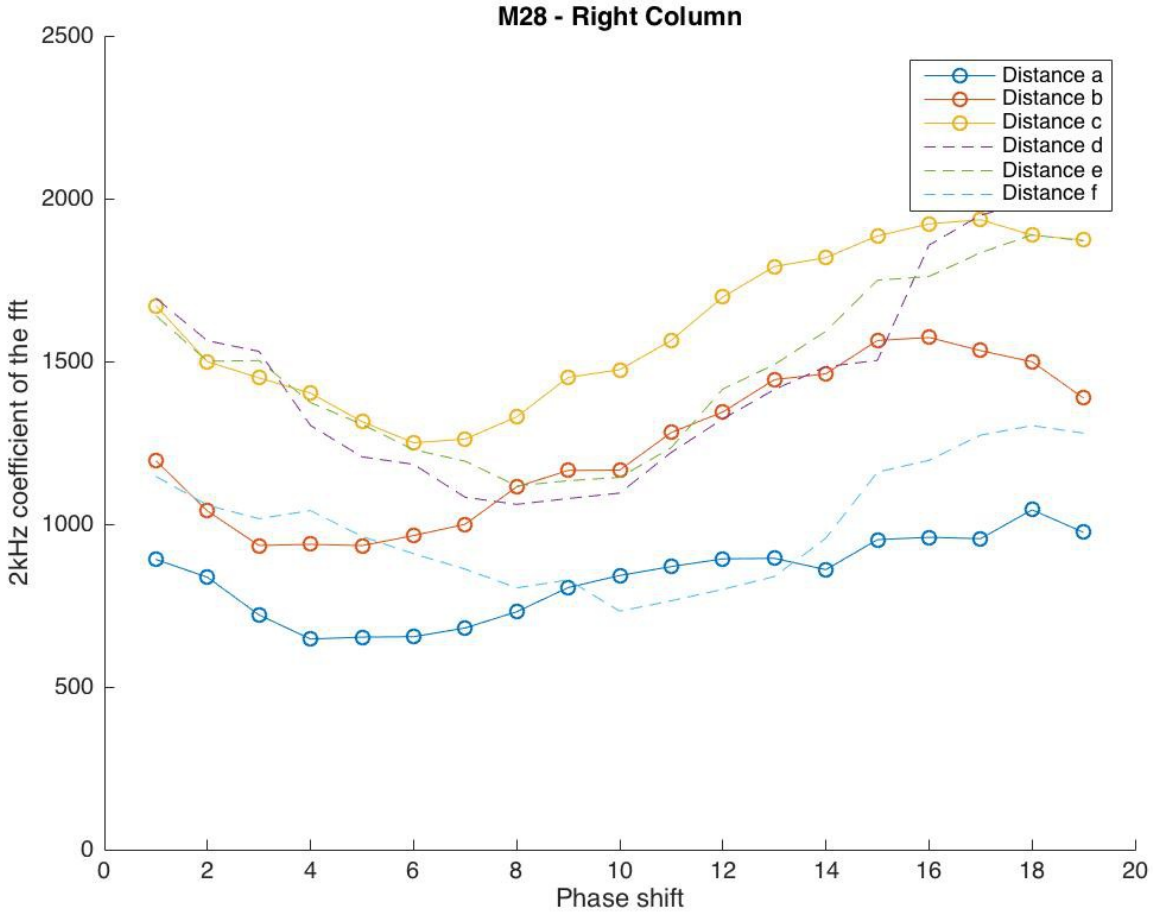

M26 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
1000
2000
3000
4000
5000
6000
7000
0
2
4
6
8
10
12
14
16
18
20

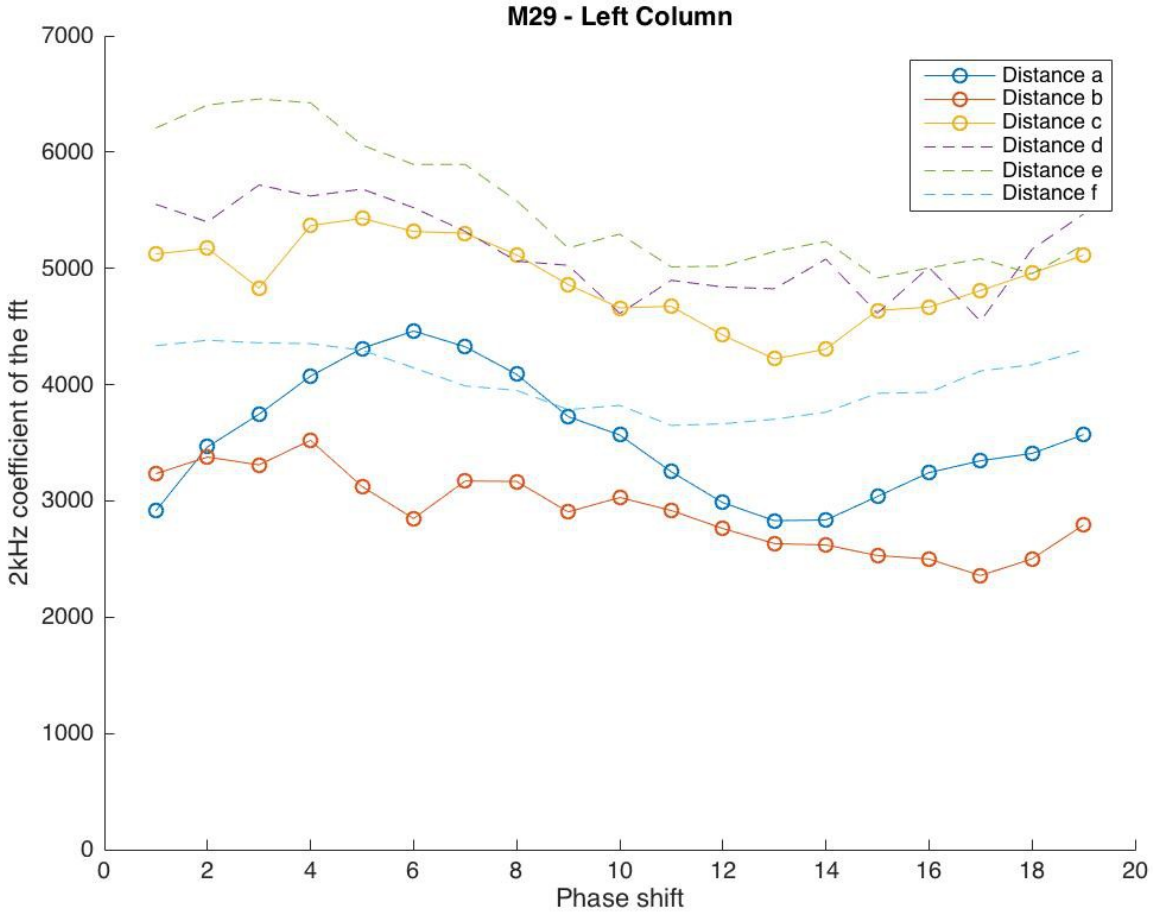

M26 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
0
2
4
6
8
10
12
14
16
18
20

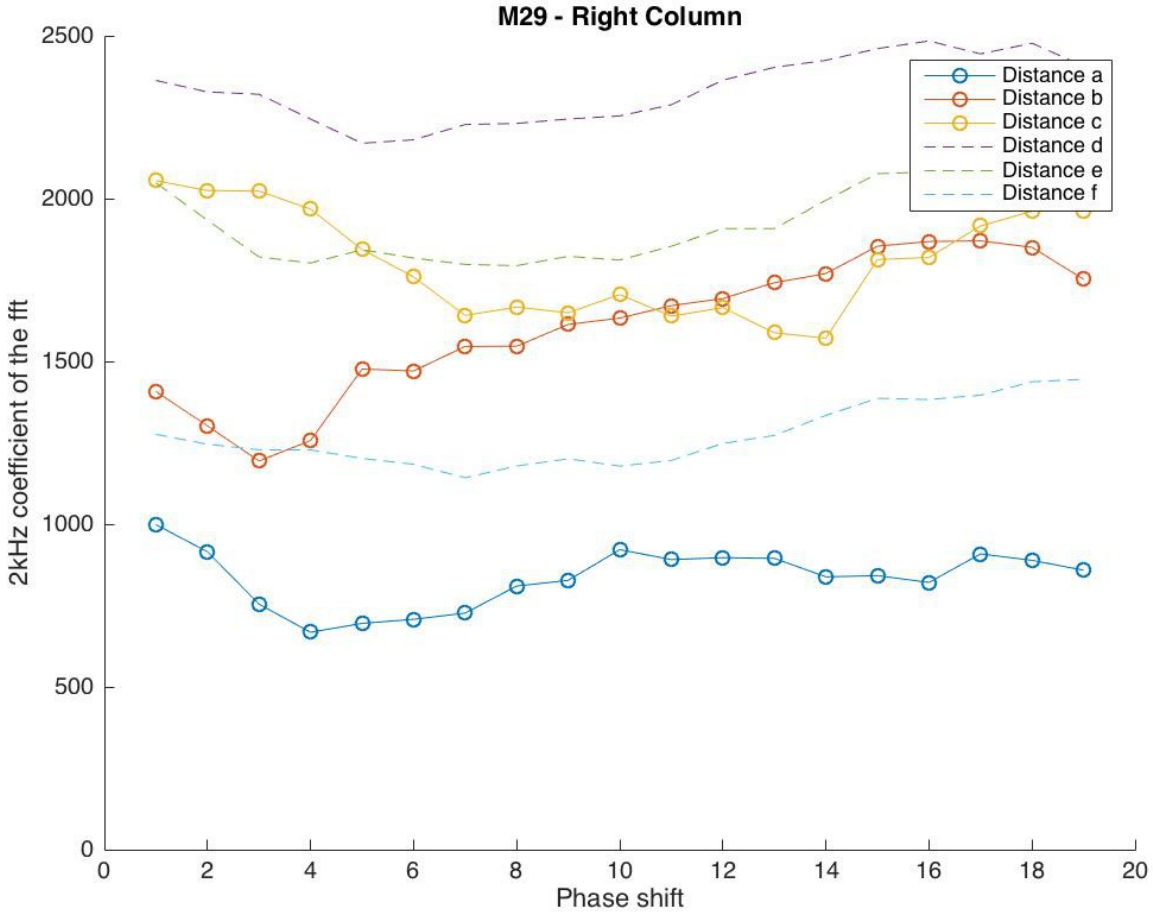

M27 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
1000
2000
3000
4000
5000
6000
7000
0
2
4
6
8
10
12
14
16
18
20


M27 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
0
2
4
6
8
10
12
14
16
18
20


M28 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
1000
2000
3000
4000
5000
6000
7000
0
2
4
6
8
10
12
14
16
18
20


M28 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
0
2
4
6
8
10
12
14
16
18
20


M29 - Left Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
1000
2000
3000
4000
5000
6000
7000
0
2
4
6
8
10
12
14
16
18
20


M29 - Right Column
Distance a
Distance b
Distance c
Distance d
Distance e
Distance f
2kHz coefficient of the fft
Phase shift
0
500
1000
1500
2000
2500
0
2
4
6
8
10
12
14
16
18
20