Washington University in St. Louis
School of Engineering
Electrical and Systems Engineering Department

ESE 498

Anti-Fraud System Design: Authenticating Magnetic Fingerprint Signal

By

Amanda Spencer and **Raja Timihiri**

Supervisors

Dr. Morley

Prof. Richter

Submitted in Partial Fulfillment of the Requirement for the BSEE Degree,
Electrical and Systems Engineering Department,
School of Engineering and Applied Science,
Washington University in St. Louis

May 2013

Student Statement

We, the authors of this paper affirm that we have applied ethics to the design process and in the selection of the final proposed design, and that we have complied with the Washington University Saint Louis.

Project Abstract

The objective was to design an anti-fraud system to discriminate between magnetic prints of cards swiped in real time, or saved as data files. The signals obtained from the card swipes will be resampled and cross correlated with all other cards, allowing the generation of two distributions: an accept and a reject distribution. These distributions allow for deciding on a threshold that would, with a small error, reject fraudulent cards and be forgiving of sloppy swipes. The resulting design successfully identifies Magneprints™ and identifies whether the swipes are authentic.

Acknowledgement

We would like to thank Dr. Morley for all his assistance in this project. We also acknowledge Professor Ed Richter support and help in various stages of the design process.

Table of Contents

List of Figures and Tables

Problem Formulation

Problem Statement

Magnetic cards are widely used in various applications to encode information. One of the main uses for magnetic cards is to store account information on credit cards, so counterfeiting is a serious issue. With current technology, it is relatively easy to duplicate a person's credit card by encoding the appropriate bit sequence on a blank card. Thus, being able to reliably authenticate cards is crucial to minimizing fraud.

A typical magnetic card is produced by encoding "0" and "1" bits on an uncoded magnetic strip, but uncoded doesn't mean completely blank. This anti-fraud system is possible because strips have small magnetic fluctuations, insignificant in comparison to the bits encoded over top, but unique to each individual card. This noise-like characteristic is known as the magnetic fingerprint. The aim of this project was to extract the magnetic fingerprint and use it to determine whether any given swipe came from the authentic card or a different one. To achieve this, it had to be isolated from the encoded information and sampled at consistent locations along the physical card regardless of swipe speed or direction. The resulting Magneprint™, which is also subject to quantizing to fit within the size requirements, must be consistent enough to produce a tight acceptance distribution when cross correlated with swipes from the same card with a significantly higher mean coefficient value than that of the reject distribution.

Problem Formulation

A magnetic card authenticator would go a long way towards preventing fraud which, besides being illegal, is also very costly to banks, customers, and enforcement agencies. We know the challenge to be realistic because there have already been successful patents. The design can be verified at the end of the project by comparing a figure of merit; in this case, separation.

Project Specifications

Background

Each card used had two tracks; Track 1 was blank and contained only the magnetic finger print while Track 2 had bits encoded over top. The cards' account numbers are encoded as binary data where "1" has two 2 peaks per bit and "0" has one. This phenomena is illustrated below in Figure 1. Each bit has the same physical length on the card, but varying swipe speed will cause the distance between bit peaks to vary. Start and end sentinels bracket the card account number all of which are encoded in groups of five bits. "0" bits pad the space before the start sentinel and after the end sentinel.

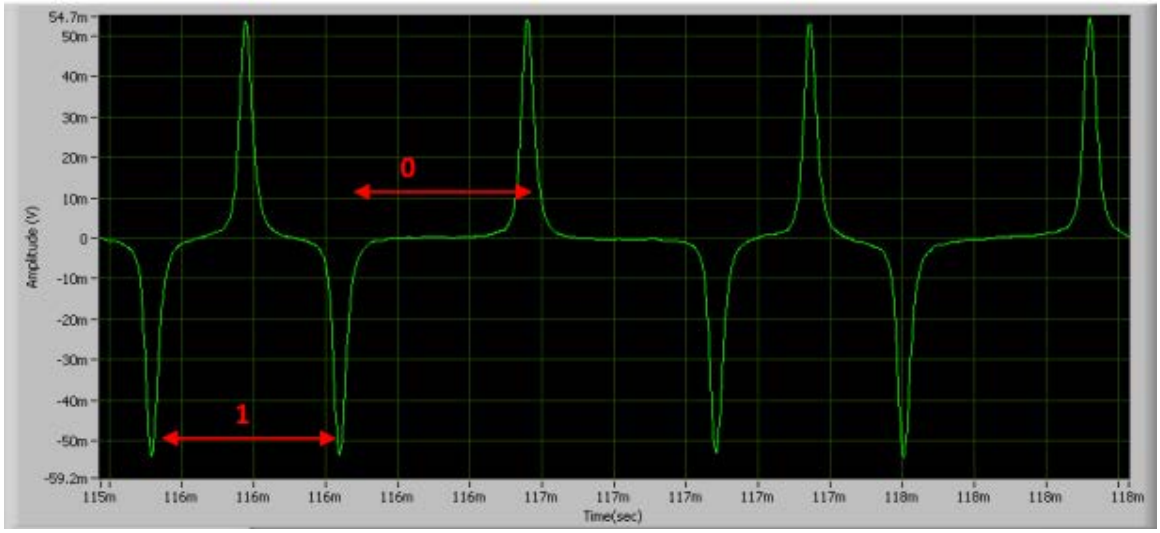


Figure 1: Card reader signal (Richter, 2013)

Magneprint™ is limited in size to one of four values: 384, 768, 1024, or 2048 bits. This value is equivalent to the product of the word size (B) in bits times the number of samples (M) used to create it.

Hardware and Software

- Grey Magtek Readers: 2 track read head
- Test cards have magstripes with 75 BPI
- Swipes generated with Collect 16 bit
- Software implemented in LabView

Design Requirements

Swipes from different cards should have low correlations even when the same data is encoded in order to differentiate between the authentic card and a fraudulent one. The mean for the reject distribution of coefficients of the cross correlations should be at or around 0. Different swipes of the same card should have nearly identical Magneprints™ so the card is identifiable as authentic regardless of speed or direction. This means the accept distribution should have a mean above 0.8. Also, the accept distribution should not be bimodal; cross correlating the Magneprints™ from forward swipes of one card to the Magneprints™ from reverse swipes of the same card should generate a positive coefficient, ideally averaging to the same value for similar swipe speeds. Success will be measured by separation which is calculated

$$S = \mu_A - \mu_R - 3\sigma_A - 3\sigma_R$$

where S is separation, μ is mean, and σ is standard deviation. Subscript A indicates the accept distribution and subscript R indicates the reject distribution. An optimal system maximizes S which can reach up to 0.7 under ideal circumstances and design choices.

To calculate a representative S, many swipes must be compared. To that end, our program was required to have a live mode which could process swipes in real time and optionally save the swipes to a file and a file mode which could load swipes from a file and process them that way.

Concept Synthesis

Preparation

This project concept was introduced by Dr. Morley, who is one of the researchers who discovered that “magnetic media, like that of a hard drive, has — if you look very, very closely — what amounts to a fingerprint” (Gustin, 2013). Though the problem had already been solved, and we were essentially led through a good portion of the rationalizations, our journey as a class and later as a group mirrored the original design process.

We began the semester by generating fake cards, represented by 1D arrays filled with 1’s and 0’s. We simulated a card reader which had a probability, P_{RE}, of flipping each bit. By comparing the output of the reader for the same cards and different cards, we generated our first accept and reject distribtions. By setting a threshold, we were able to calculate the true positive rate vs. false positive rate as P_{RE} changed and generate an ROC curve.

The next step was to create fake magnetic fingerprints. We represented the fingerprints with differentiated white Gaussian noise (σ=1) passed through a bandpass filter. We generated 50 cards of N samples each. To simulate the Magneprints™ , we then took every delta-th sample until we had M=1024 samples for each card. To simulate the reject distribution, we cross correlatated the sample of all the possible combinations of the cards. We then plotted the variance of the distribution against delta from delta=1 to delta =32. The resulting graph is shown in Figure 2 below.

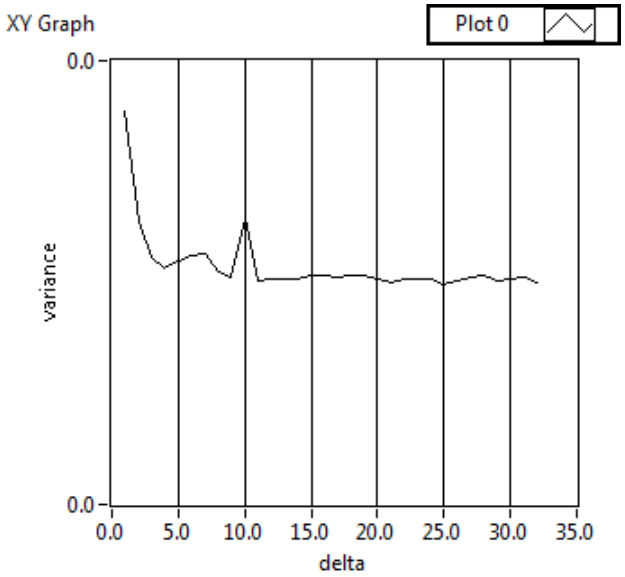


Figure 2: Variance vs. Delta

We found that variance decreases exponentially with increasing delta until it eventually plateaus. In this case, variance plateaus at around delta = 11. At fixed delta=11, we plotted variance of the reject distribution vs. M as it was varied from 2 to 2048. The resulting graph is shown in Figure 3 below.

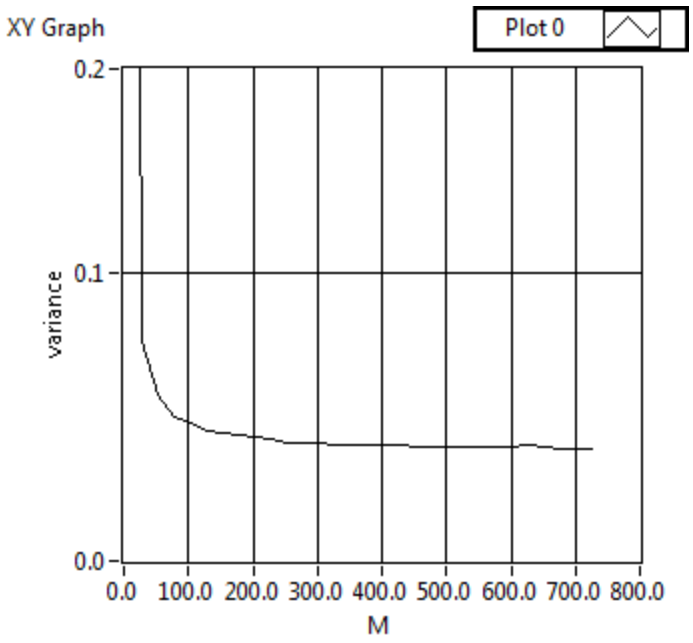


Figure 3: Variance vs. M

This would seem to indicate that any value of M over 300 gives almost equivalent results with negligible change in variance.

Final Design

After these simulations, we finally proceeded to real cards. We started out comparing Track 1, which saved us the trouble of extracting the magnetic fingerprint from the encoded data, and gradually worked our way up to Track 2. During the process, we ran into several design choices. Most design choices were debugging issues, errors that prevented the program from working at all and needed to be fixed. When creating the Magneprint^{TM}, we had to reverse and negate the backwards swipes to have any chance of getting the same values. The histograms were recounting previous swipes and skewing the data until we put the coefficient value being appended below the array of previous coefficients in the build array block and set the reset on each of them to true. The speed was way off before we realized that the array had already converted from time to samples so we had to multiply by the number of samples per second. Some design choices were already provided for us, such as the quantizer and the swipe capture though we altered the swipe capture method upon Dr. Morley's suggestion to fix a clipping problem. The resampler was waffling between 266 and 267 samples per bit until we switched the block to open interval.

Other design choices were common sense. We quickly observed that the output generated by the ELVIS board for the card swipes varied greatly with swipe speed. Since finding the same location on the card would be virtually impossible without a constant bit time, we decided to resample the signal. Threshold values were also determined based on looking at the input signal and choosing logical values.

The main design choice we faced was how to extract the magnetic fingerprint from the encoded information. The resampled waveform is shown below in Figure 4 with the area of interest highlighted.

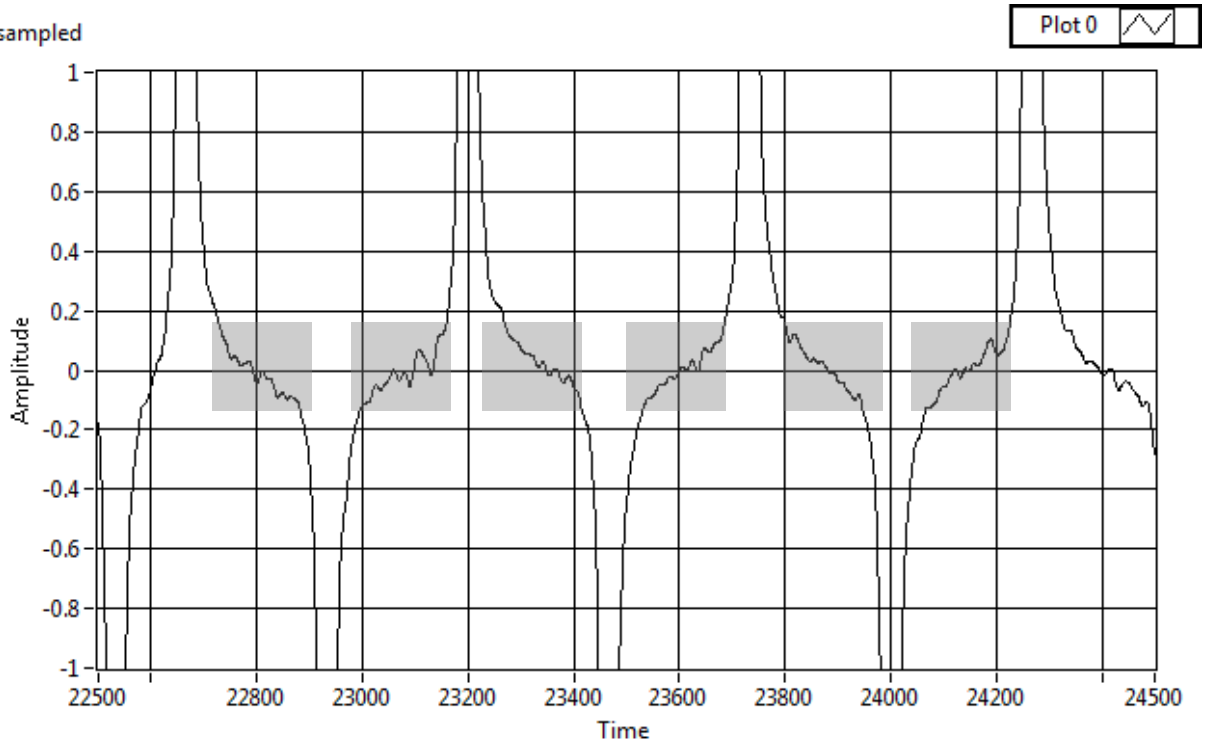


Figure 4: Resampled Waveform

We initially tried simple clipping to remove the peaks, but that still left the slope in the region of interest. We tried passing the clipped waveform through a highpass filter to remove the drift and alternatively passed the original resampled waveform through a bandpass filter in an attempt to cut out the low frequency drift and the high frequncy spikes simultaneously. We tried different cutoff values, but both the spikes and the drift remained. We even put the waveform through an FFT block before and after the filter to try to isolate the problem frequencies , but the filter wasn't having the effect we wanted.

We then turned to the literature (Morley J. e., 2009). The article said that the drift could be removed by subtracting out the running average of $1/8^{th}$ of the bit. Since our resampled waveform was normalized to 267samples, the running average would be taken over 33 samples; the current value and the sixteen samples on either side. The time domain transform is a rectrangle with width 33 and height 1/33 which translates to a sinc function in the frequency domain which acts as a lowpass filter. By subtracting the lowpass-filtered signal from the original signal, the frequency transform becomes a highpass filter, thus eliminating low frequency drift. This seemed to work relatively well, but since the flat portions are restricted to ±0.03 V, we decided to coerce the signal before filtering it to minimize any potential fluctuations, such as those made by scratches. The output of this process (shown over the same time range as the resampled waveform) is shown below in Figure 5.

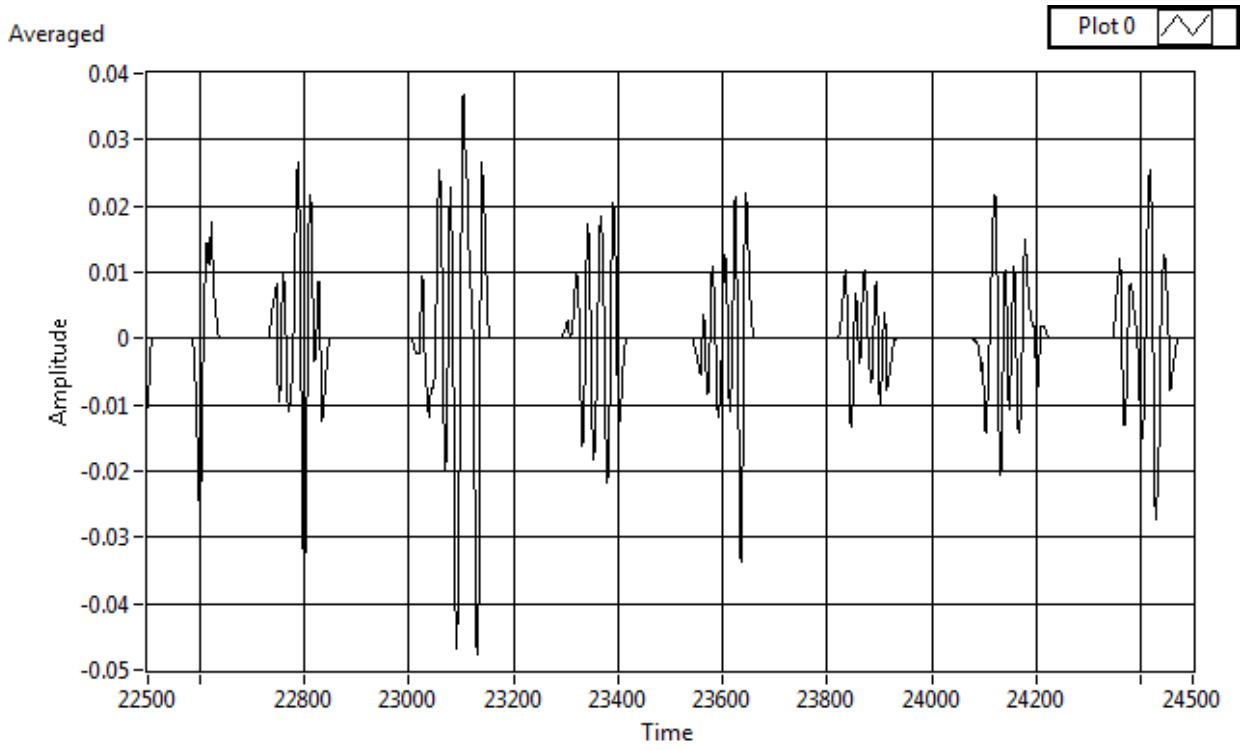


Figure 5: Averaged Waveform

The next step was to choose which bits to use. We couldn't use "1"s because their shorter period didn't allow for a flat region. The "0"s towards the beginning and the end were also out since they could wear off. The easiest way was to look for the first "0"s following the first "1". Orienting within the card is important, but so is orienting within the bit. Looking at several "0" bits, it appeared that the middle 100 samples were relatively flat before the filter, so we extracted

that region from each bit of the filtered signal and concatenated it back together. The output looks something like the graph in Figure 6 below.

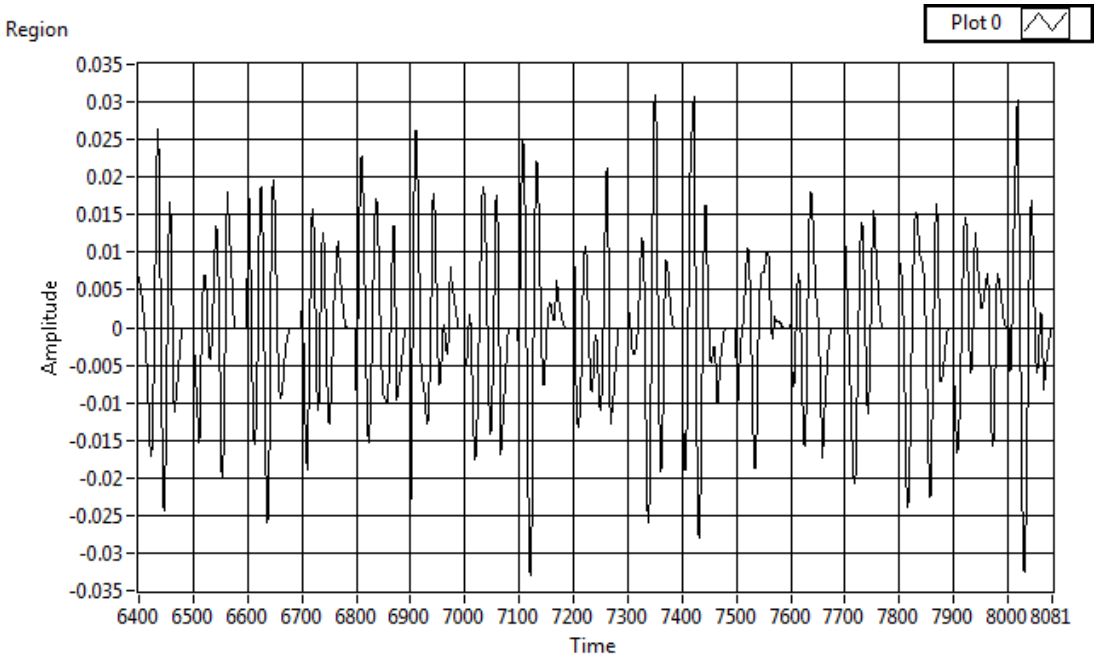


Figure 6: Extracted Region

After quantizing and indexing, the Magneprint™ should look similar to Figure 7 below.

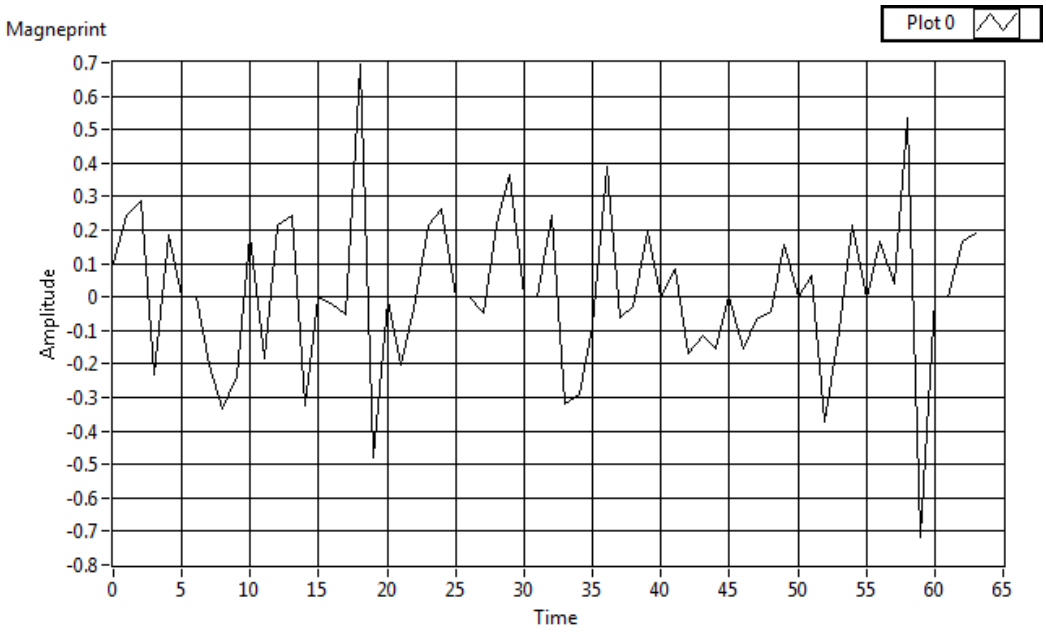


Figure 7: Magneprint™ Selection

Analysis and Design Presentation

Subsystems

The flow of information through the various modes and options is summarized in Figure 8 below.

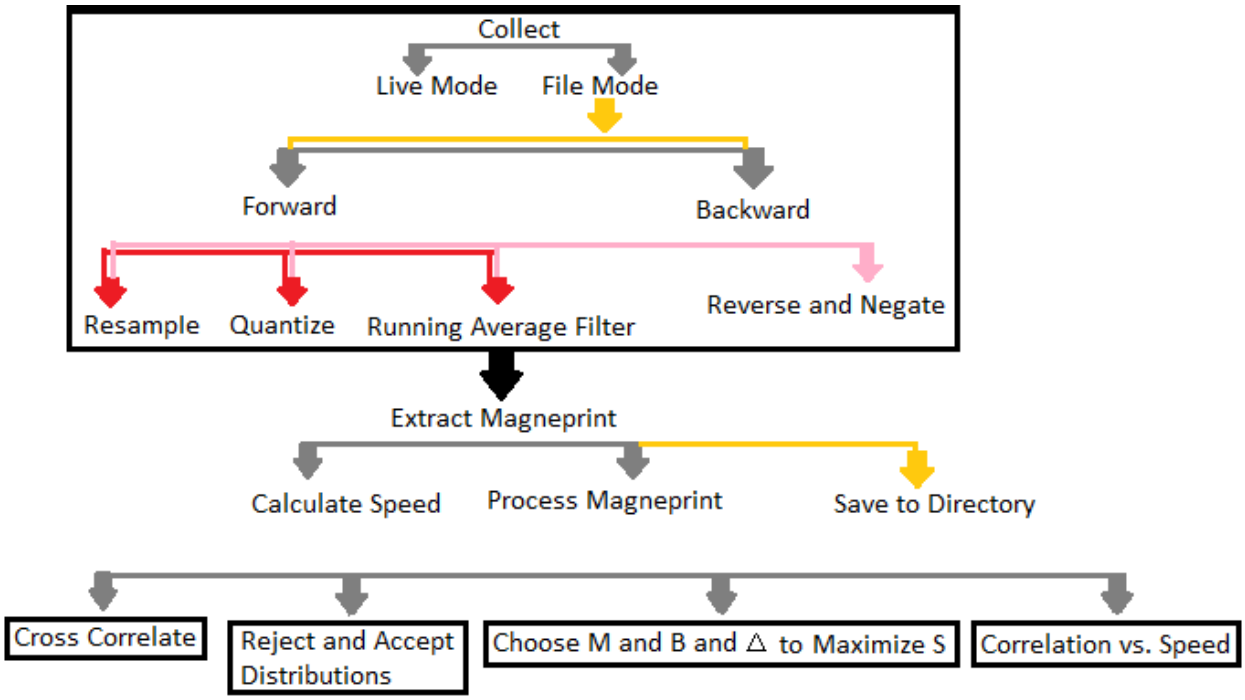


Figure 8: Software Flow Diagram

Live Mode

The Live Mode processes the swipes and cross correlates them as they are manually collected in real time. It extracts the Magneprint™ from the first card and cross correlates all subsequent ones against the original. A button on the front panel allows you to indicate whether the card being swiped is the same or not, whether to add it to the accept distribution or the reject distribution on the histogram. The front panel also allows you to choose whether to save the swipes to a file or not. Regardless, the processed swipes along with their speeds and their same card indicators are saved for the duration of live mode in a cluster shift register.

Resampling

First, the index of the peaks above a certain threshold are found. The number of samples between the first two peaks (the first bit will be 0 regardless of the direction of swipe) is measured to set a base bit length. If the length of the next peak to peak region is more than 75% of the last preceding “0” bit, the region is concluded to be another “0” bit; a “0” is added to the bit array, the region is resampled to 267 equally spaced samples, and the speed (in inches/second) is

calculated by multiplying length (equivalent to samples per bit) by 75 bits/inch, dividing by 500,000 samples/second, and taking the inverse . If the length is less, the region is concluded to be half of a “1” bit; “1” is added to the bit array, the region is resampled to 133 equally spaced samples, and the speed is calculated using 150 half bits/inch . The resampled bits are concatenated back together to generate the resampled waveform, the duplicate “1”s are removed from the bit array, and their corresponding speeds are removed from the speed array. This process is illustrated in Figure 9 below.

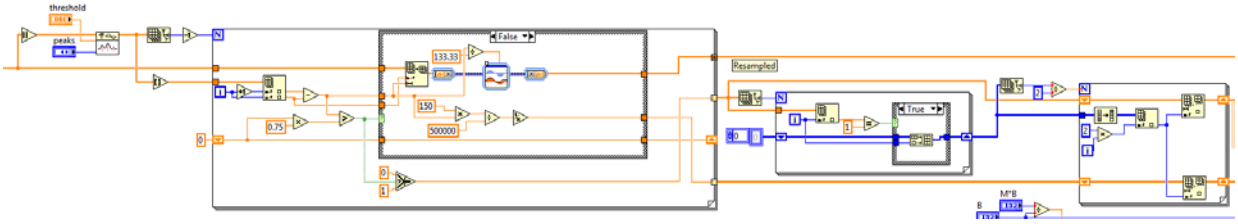


Figure 9: Resampling Block Diagram

Binary Reader

The binary reader takes in the bit array and scans from start to end and end to start for the start sentinel preceded by 10 “0”s to ensure the pattern is not also found within the sentinels. (Within the sentinels, the parity bit would prohibit that many contiguous “0”s). If the start sentinel is found scanning from start to end, the swipe was forward and if the start sentinel is found scanning from end to start, the swipe was reverse.

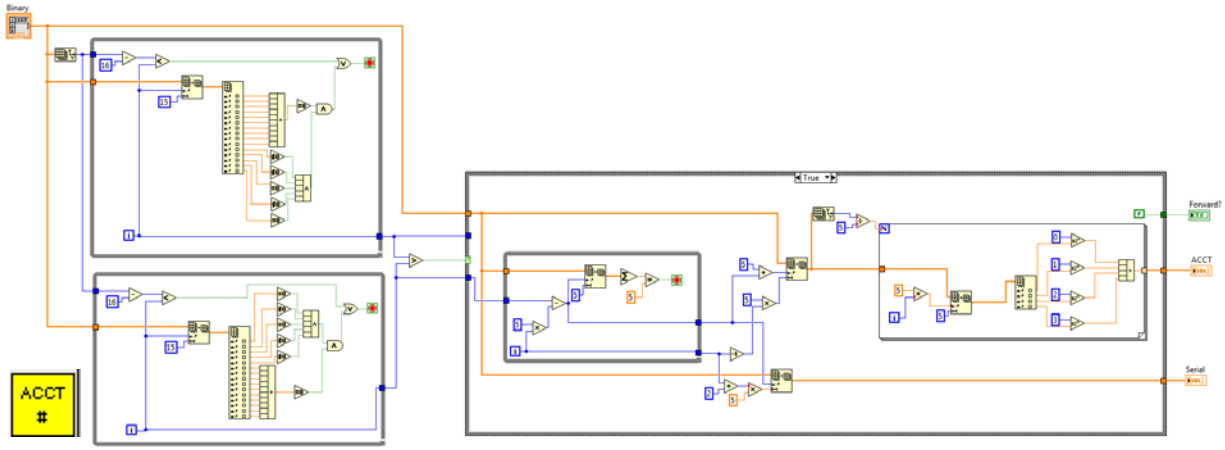


Figure 10: Binary Reader Sub VI

Extractor

The extractor takes the resampled waveform, removes the peaks by coercing it into the range of -0.06 to +0.06, then subtracts out the running average. When the swipe is forward, the VI searches the bit array for all of the "0"s following the first "1". It converts those indices to indices compatible with the resampled waveform, finds the corresponding bits, then takes the 100 middle samples which should contain the majority of the flat portion. The flat portions of each bit are concatenated at the end to form the extracted region. If the swipe was reverse, the bit array is reversed before searching for the first "1" as is the resampled waveform for consistency's sake. The resampled waveform is also negated to compensate for the north to south transitions becoming south to north transitions. The output should be the same regardless of swipe direction. The Sub VI is shown below in Figure 11 with the reverse case.

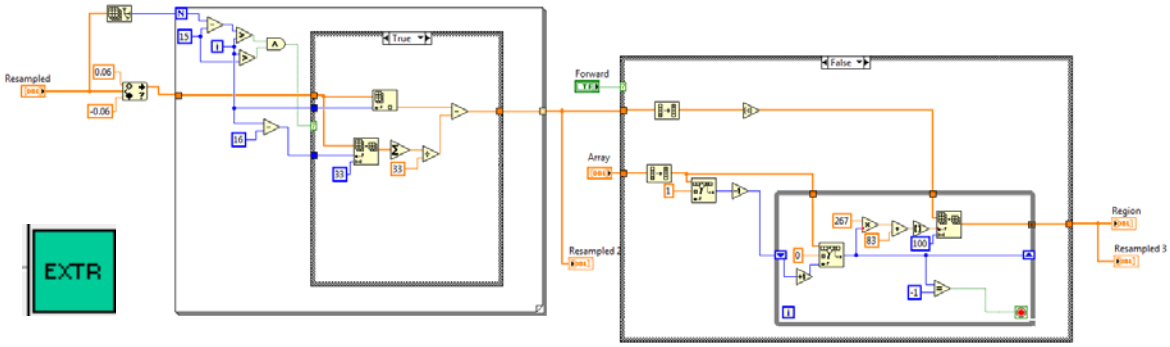


Figure 11: Region Extractor

Save to File

The case statement shown below in Figure 12 engages when the user selects the Save to File option. The File Saver VI saves the extracted region of the swipe to the folder of the user's choice. The other inputs include speed indicator, Card #, Swipe #, and direction. Speed indicator and direction were determined in earlier processes and Card # is a control on the front panel. Swipe # starts at 0 and increments with each successive swipe until Card # is changed, at which point it returns to 0 again.

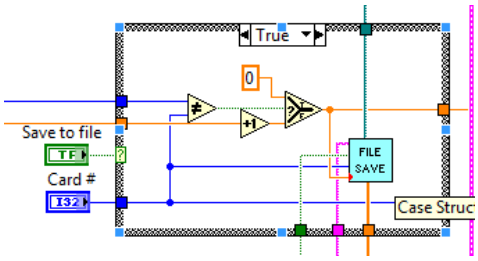


Figure 12: Save to File Block Diagram

The inside of the File Save Sub VI is shown below in Figure 13.

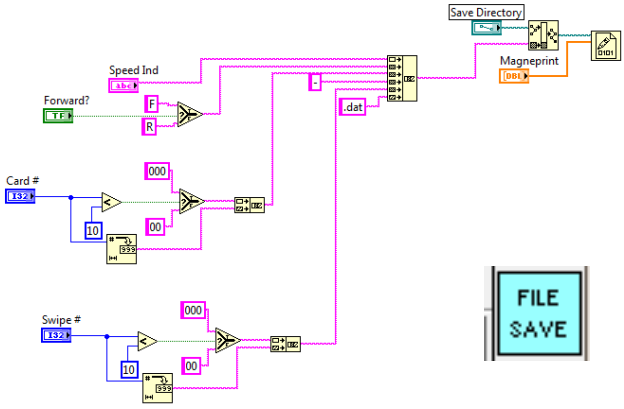
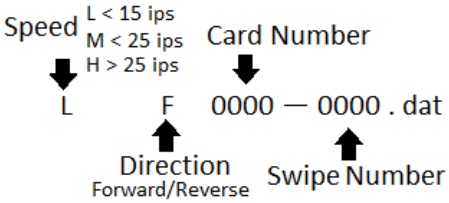


Figure 13: File Save Sub VI

The speed indicator, direction, card #, and swipe # are appended into a string to form the file name with the following format:



Indexer

The indexer takes in the extracted region, quantizes it to B bits per word, then takes every delta-th sample until it has M samples. These M samples are the output Magneprint™. The Sub VI is illustrated below in Figure 14.

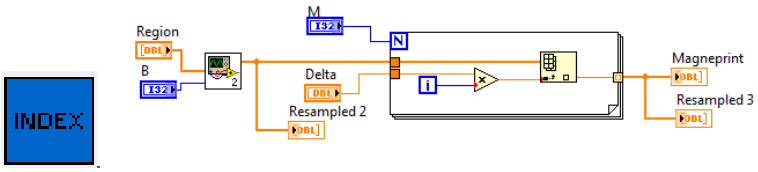


Figure 14: Indexer Sub VI

The indexer is placed within the for loop for varying delta and/or B when generating those graphs.

File Mode

File Mode pulls the saved extracted regions from the file, creates the Magneprint™ for each swipe, correlates every possible combination, and adds the correlation coefficients to the proper histogram. The program first generates a list of card numbers by converting the appropriate characters of each file name string to a number, then deleting the duplicates. This part of the program is illustrated below in Figure 15.

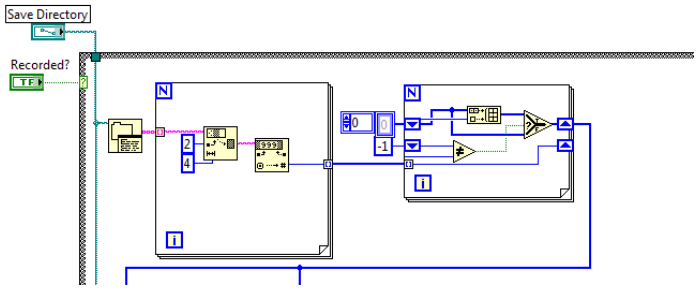


Figure 15: File to List of Card Numbers

The list is then passed to the File Selector Sub VI which calls the appropriate swipes

File Selector

The File Selector Sub VI obtains all the files from a specified directory with the desired card number. The inside of the Sub VI is illustrated below in Figure 16.

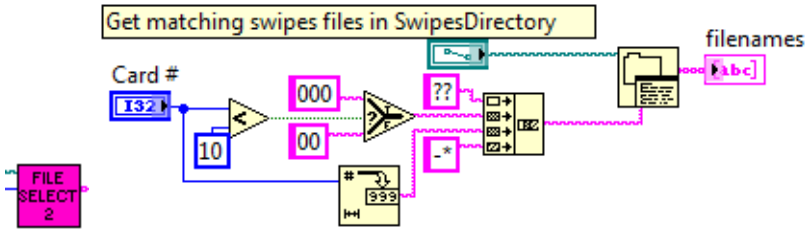


Figure 16: File Selector Sub VI

Cost Analysis

The design is completely implemented in software, so cost isn't an issue in the traditional sense. The main restraint is the size. The Magneprint™ sizes are restricted to 384, 768, 1024, and 2048 bits. This forces us to use quantization, so parameters must be chosen accordingly.

Hazards and Failure Analysis

Because the product is software, there are no significant safety hazards. There are, however, many potential failure spots, most of which sported probes or graph indicators at some point during debugging. The LRC Check VI was written to make sure that the bits were read properly for each swipe and that we were distinguishing between forward and backward swipes by checking the parity bits. The later reincarnation of that program is the Binary Reader VI which outputs the account number for each card and a serial number. For same swipes the account number remained the same, as well as for forward and reverse swipes of the same cards.

Parameters and Results

Once we had a functional program, we had to optimize S for each Magneprint™ size by choosing values for B, M, and delta. To be consistent, we used the 55 Card Reject Set for all optimization runs (55 cards of approximately 4 swipes each).

In the simulation, we found delta had to be at least 11, so we chose 20 just to be safe. We fixed the delta value and varied B (word length in bits) for each of the Magneprint™ sizes. Since M (number of samples in the Magneprint™) is inversely proportional to B (M*B = Magneprint™ size), changing B also affected M. Plots of S vs. B for each of the four sizes are shown below.

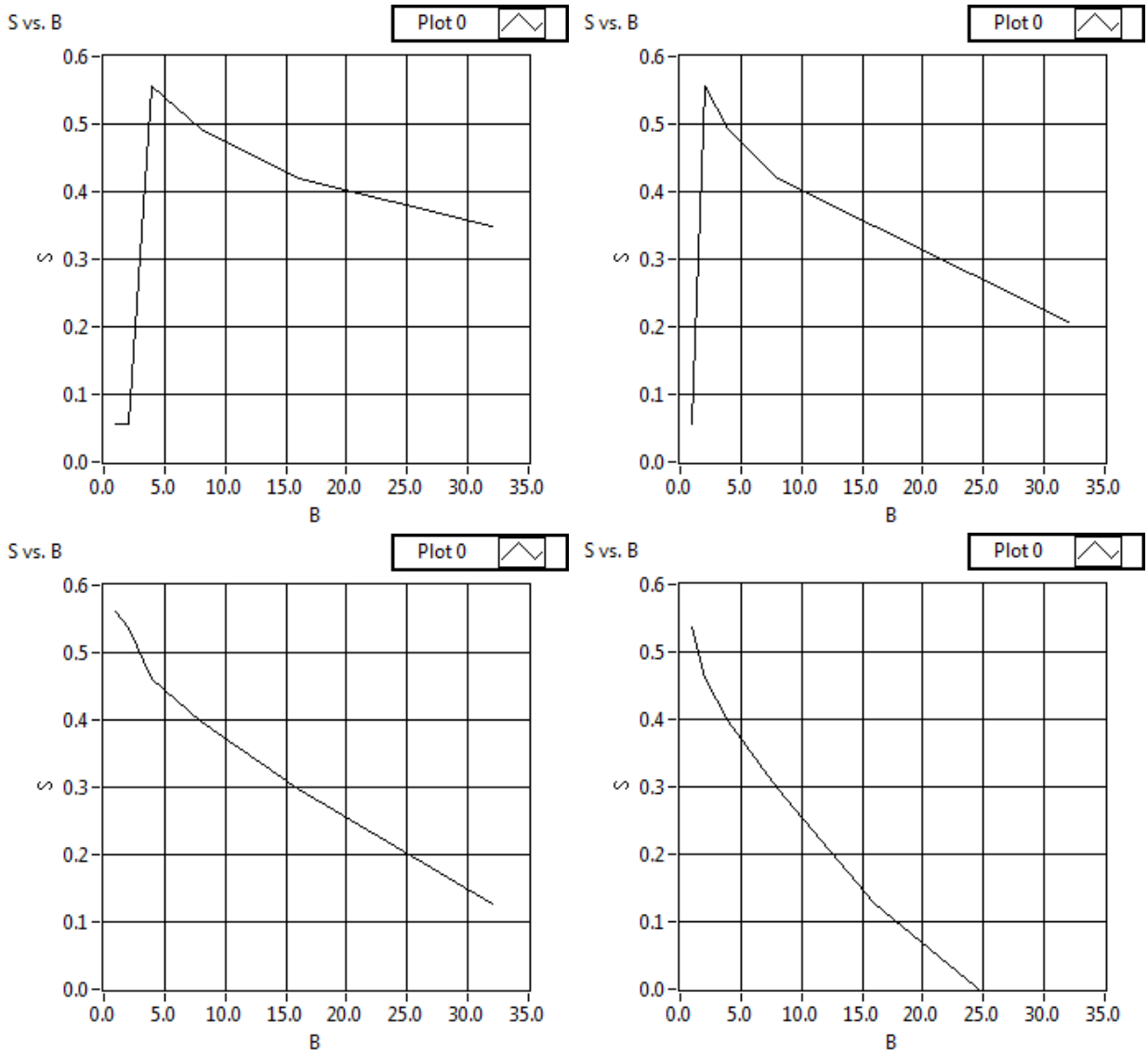


Figure 17: S vs. B

Figure 10a shows the graph for the 2048 bit size, Figure 10b represents 1024 bits, Figure 10c represents 768 bits, and Figure 10d represents 384 bits. B was varied exponentially, ranging from 2^0 to 2^5. For 2048 bits, it appears that the best B is 4. 1024 bits seems to peak at B=2. 768 and 384 bits achieve the highest S at B=1 and steadily decline from there.

Fixing B at their respective maximal values and using the corresponding M values based on their size, we varied delta to find maximal S. The four S vs. delta graphs are shown below.

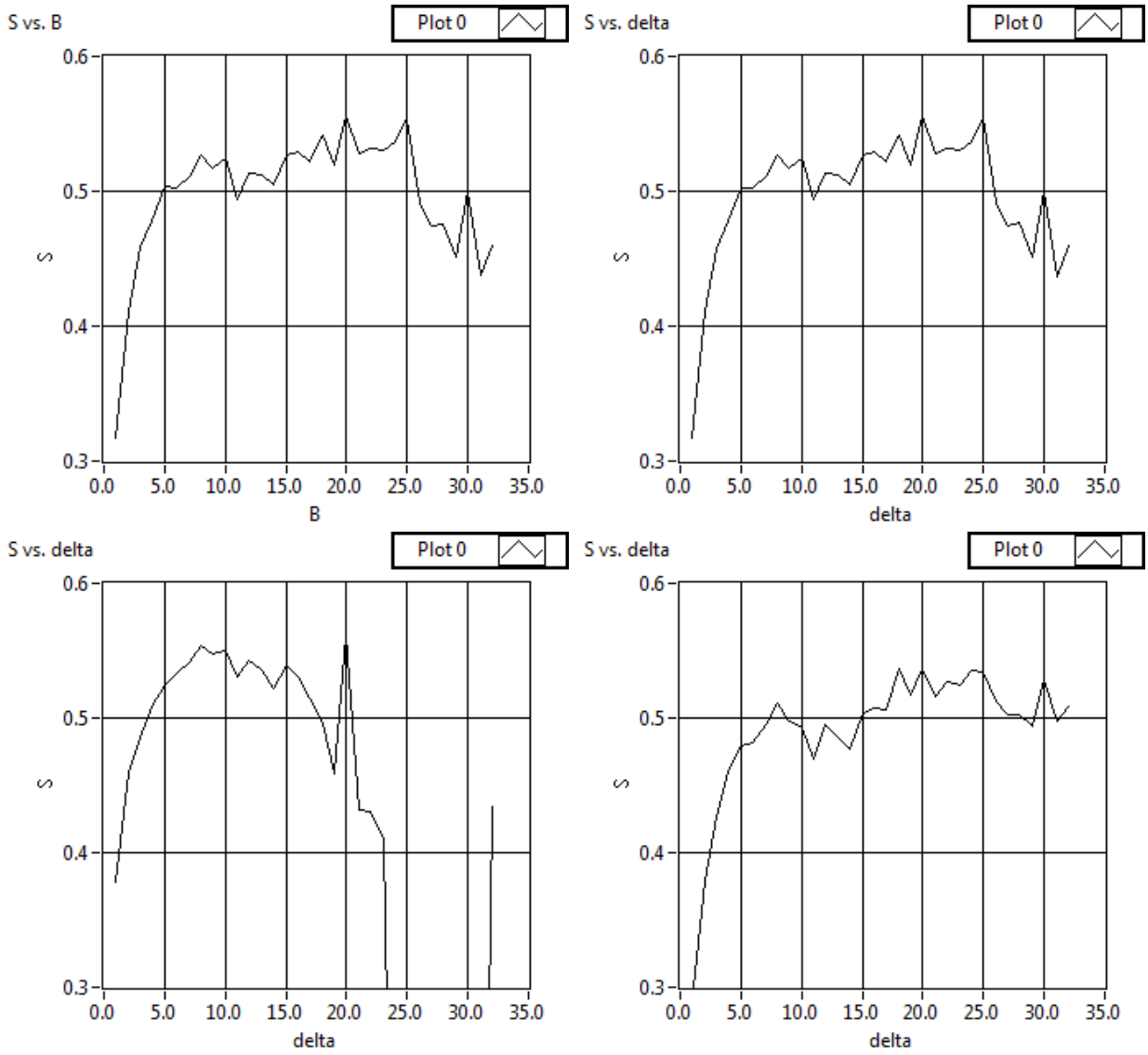


Figure 18: S vs. Delta

As in Figure 10, 11a shows 2048 bits, 11b shows 1024 bits, 11c shows 768 bits, and 11d shows 384 bits. All four graphs show an initial, sharp increase in S as delta increases but eventually peaks and begins to decline. For 2048, 1024, and 384 bits, S seems to be maximal around delta=20-25. For 756 bits, the maximum is around delta = 8, though there is another spike at delta=20. At any rate, delta = 20 is a pretty safe choice for all of the sizes. Optimization parameters for each of the sizes are summarized in the table below:

Table 1: Optimization Parameters			
Size (bits)	B(bits)	M(samples)	delta(samples)
384 bits	1	384	8
768 bits	1	768	20-25
1024 bits	2	512	20-25
2048 bits	4	512	20-25

When optimized, the distributions should look something like they do in the figure below. The graph shows a histogram of the correlation coefficients when the parameters are size=1024, B=2, and delta=20. The accept distribution is shown in black, and the accept distribution is shown in red. The resulting S was .56, and would probably have been higher if it weren't for the two bad swipes in the accept distribution inflating the standard deviation.

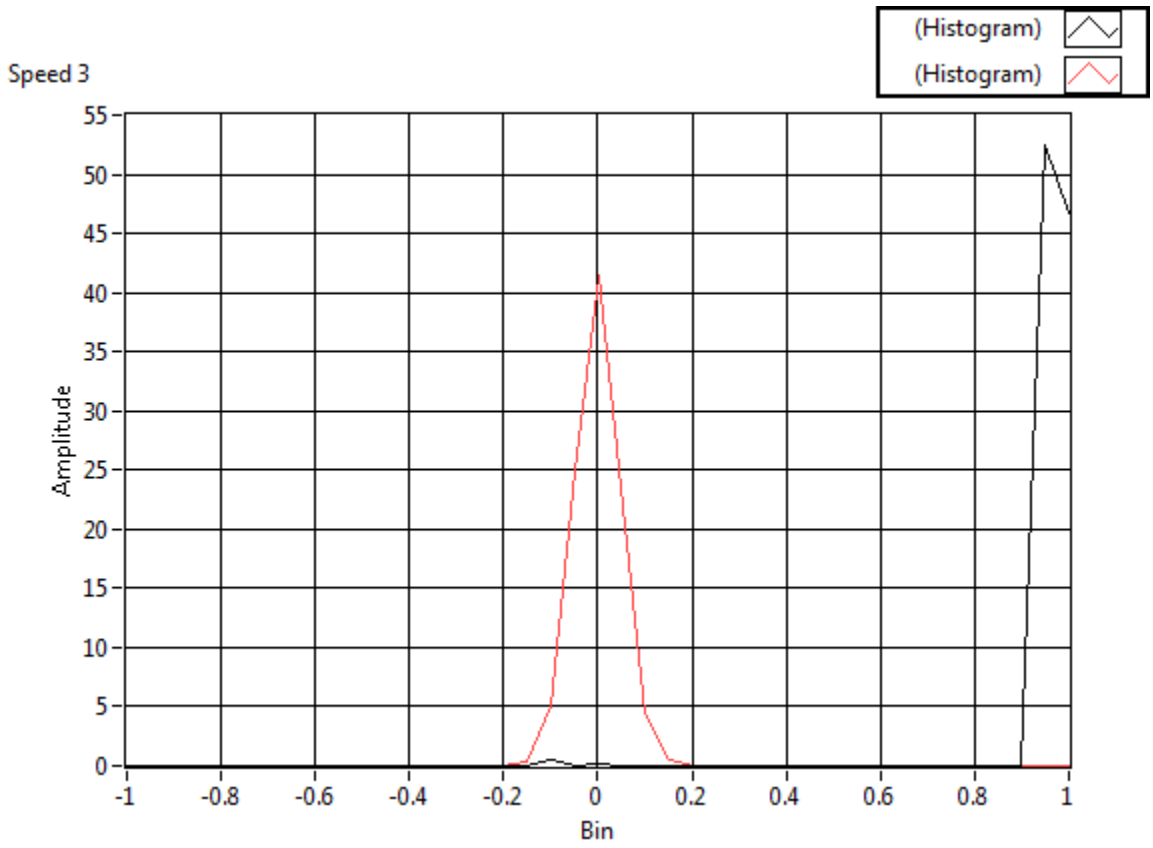


Figure 19: Histogram of Accept and Reject Distributions

Another factor that affects the separation is the speed of the swipe. For the sake of simplification, we took the average speed over the area used for the Magneprint™ and relegated the swipes to one of three categories; medium speed was defined as 15-25 inches/second (IPS), low speed was below that range, and high speed was above. To investigate the effects of speed on correlation coefficient, we plotted a histogram of the accept distribution with the swipes of different speed ranges depicted in different colors (original swipe is around 25 IPS). The resulting graph is shown below.

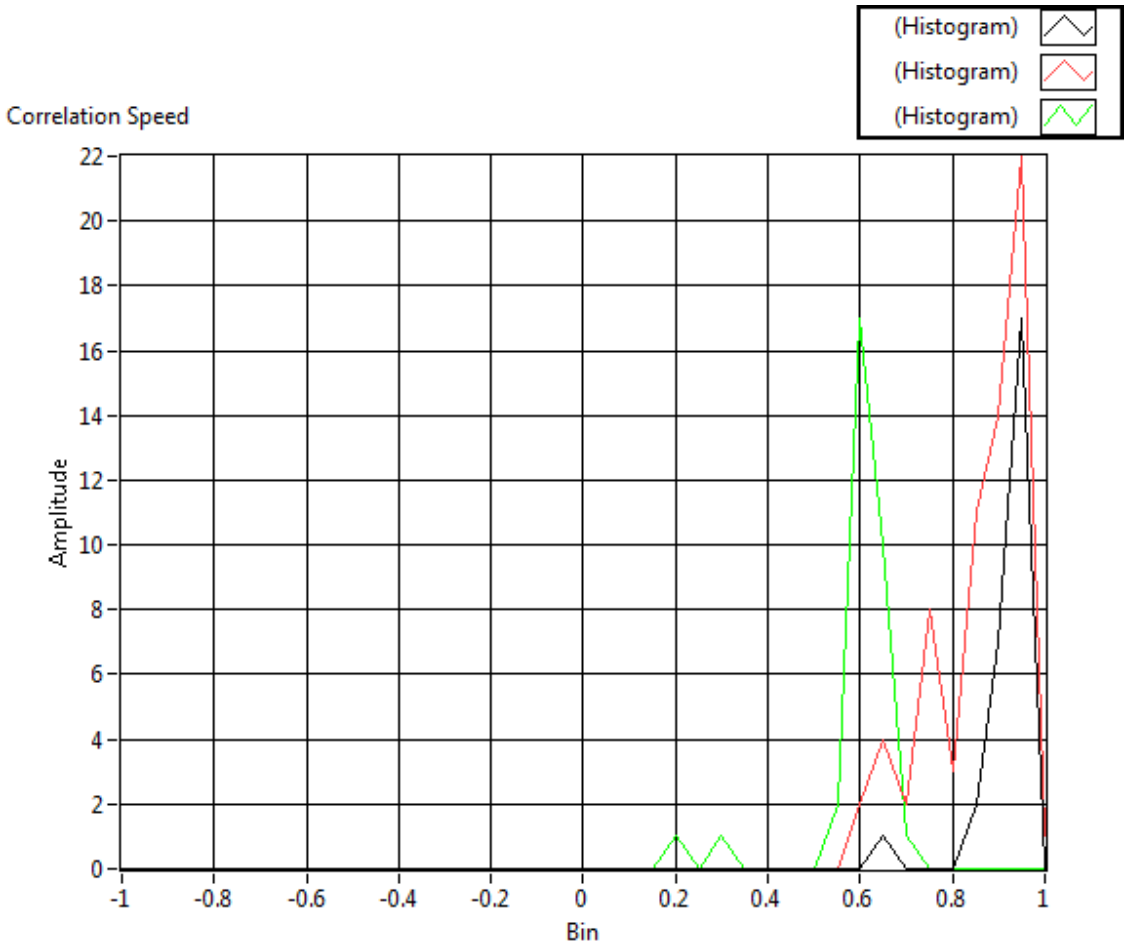


Figure 20: Accept Distribution for Various Speeds

In this graph, black shows high speed, red is medium speed, and green is low speed. The bins represent the correlation coefficients and the amplitude is the number of swipes. Low speed clearly tends towards lower correlation coefficients. Both medium and high speeds have mean correlation coefficients around .9, but high speed has a slightly higher mean and a smaller standard deviation. Consequently, faster swipes improve separation and the liklihood of correctly identifying the card.

Conclusion

The running average filter was the key to extracting the magnetic fingerprint from the encoded data. By carefully choosing the bits and the indices within each bit, we were able to create a consistent Magneprint^{TM} which correlated highly with swipes from the same card and didn't correlate with swipes of a different card. With optimum parameters, each size was able to obtain S>.5, though slower swipe speeds could lower that value. The reject distribution is centered at 0 and has minimal if any overlap with the accept distribution, which should make the placement of a threshold very straightforward. The system appears functional and ready to ship. If there were more time to make improvements, however, we would like to look into the quantizer and figure out why the region of the bit being used seems to shift to the right towards the end of the swipe. It could be that the

Bibliography

Gustin, G. (2013, April 06). *Washington University discovery could lead to more secure payments*. Retrieved April 30, 2013, from St. Louis Post-Dispatch: http://www.stltoday.com/business/local/washington-university-discovery-could-lead-to-more-secure-payments/article_a9b8497f-d1da-5ca0-8ab1-3f12ab7c4c13.html

Morley, D. (2013, January 14). *ESE498 Senior Design*. Retrieved 03 01, 2013, from ESE498: www.ese.wustl.edu/~rem/ese498/

Morley, J. e. (2009, January 20). Methods and Apparatus for Authenticating a magnetic Fingerprint Signal Using a Filter Capable of Isolating a Remanent Noise Related Signal Component. *Unites States Patent, Patent NO.: US 7,478,751 B2* , p. 28.

Richter, E. (2013, January 14). *Introduction to Electrical Engineering*. Retrieved from ESE103: www.classes.engineering.wustl.edu/ese103/images/3/38/magnetic_card_reader_instructions.pdf