Brandon Wright
ECE 6590
December 14, 2023

Software Radio Final Project – QPSK Receiver

Abstract—For this project, we were tasked with creating a QPSK receiver to demodulate a series of progressively more degraded input signals. Part I implements demodulation, timing acquisition and data extraction. Part II adds a symbol-spaced equalizer to Part I. Part III adds a fractionally-spaced equalizer to Part I. Part IV implements carrier acquisition and tracking to correct for phase and carrier offset. Part V adds symbol rate tracking. By following the textbook and applying topics taught in class, the end result was a receiver that is able to successfully demodulate and extract the data from all samples, despite how degraded or noisy the input signal was.

The software radio receiver in this project was implemented in five parts using textbook Chapter 12 as a guide. The data packet of the received signal consisted of a 128-symbol cyclic preamble and payload. The preamble consisted of four repetitions of a 32-symbol pilot sequence **cp**, which was followed by a data payload of variable length. The data symbols in the packet were mapped using QPSK with Gray mapping. The sequence **cp** was used to identify the beginning of the payload starting in **Part I**. **Parts I, II** and **IV** decimate the received baseband signal **xBB** by a factor of 100 by finding the correct symbol timing by finding the maximum values of the periodic signal power function **rhot**.

Parts II and **IV** use **cp** to implement a symbol-spaced equalizer to correct for noise and channel distortion. For these parts, the absolute value of the signal's autocorrelation is measured and a peak is found in the beginning of one repetition of the sequence **cp**. The location of this peak is then used to estimate tap weights for the equalizer using the NLMS adaptation algorithm. The tap weights are then used on the rest of the packet to adjust for any noise or other error. **Part III** skips the timing phase estimation and instead decimates **xBB** at twice the symbol rate (i.e. L=50) and implements a half symbol-spaced equalizer using **cp**.

Part IV introduces carrier acquisition and tracking to estimate and correct for carrier frequency offset (**Dfc)** as well as carrier phase offset (**phic)**. **Dfc** is estimated by finding complex number **J** over one period in the preamble using pilot aided carrier recovery. The angle of **J** is then used to estimate and adjust for carrier offset **Dfc_est. Phic** is then estimated using a decision-directed phase recovery loop and used to adjust the baseband signal accordingly. **Part V** implements a method to track the signal's symbol rate in the event that symbol rate experiences an unknown frequency offset. This is done using a decision-directed timing phase recovery method.

After adjusting for the various sources of error in **Parts I** through **V**, the payload is converted from QPSK to an array of bits which are saved to the file **transmitted_file.txt** to store the final, decoded message. To test the receiver, the class CD contained several .mat files that contained different data messages of various lengths that were affected by noise, error, and different types of offsets. Test files xRF1 through xRF9 were all successfully demodulated and decoded and read with very little error in the final text.

Part I – Demodulation, timing acquisition, and data extraction

For the first part of the project, we were tasked with demodulating a signal with very little noise and extracting its data. First, input signal **xRF** was converted to the complex-valued baseband QPSK signal **xBB**. Because $f_c = 10^5$ Hz, the out-of-spectrum components were filtered out of the baseband by convolving **xBB** with **pT**, which acted as a bandpass filter. The spectrum of **xBB** and filtered baseband signal **xBBf** are shown in **Figure 1(a) and (b)**. As shown in **Figure 1(b)**, the out-of-spectrum components were successfully removed as seen by the single spike in the frequency domain. The eye pattern of **xBBf** is shown in **Figure 1(c)**.

Next, the timing phase of **xBBf** was recovered using a non-data aided timing recovery method. In the absence of timing offset, the timing phase that maximizes the signal power **rhot** should be used to get the correct set of symbols. Because of the lack of timing offset for the first set of samples, **rhot** should be periodic, and the maximum value should repeat every L samples, where L is the decimation factor. Because **rhot** is periodic, the first maximum only needs to be located and the rest of the baseband can be decimated at intervals of L starting at the first maximum, yielding decimated baseband signal **xBBd**. L = 100 for this project.

Finally, after the timing phase was determined, the preamble was identified in **xBBd** by looking for four repeats of the 32-sybmol pilot sequence **cp**. The beginning of the data payload was found by looking for the end of the preamble, and the payload symbols were converted from QPSK to binary using the Gray mapping found in **Table 1**. The eye pattern of the payload is shown in **Figure 1(d)**, and it is apparent that its contents are indeed a QPSK signal considering the values all land at values near to ±1 ± j. The data bits of the payload were saved to a text file, which contained a pun about two math books.

Part II – Symbol-spaced equalizer

For the second part of the project, a symbol-spaced equalizer was implemented to adjust for channel distortion and noise in the signal. Firstly, the absolute value of the autocorrelation $\mathbf{r}_{yy}$ of **xBBd** was calculated to find the beginning of the preamble, as shown in **Figure 2**. As shown in **Figure 2**, there are peaks in $|\mathbf{r}_{yy}|$ that correspond to the beginning of the end of one iteration of **cp**. The location of one of the peaks was used as the starting point to get a series of 32 symbols from **xBBd** that were then used to estimate the 32-symbol long tap weights **w** using the NLMS adaptation algorithm with 100,000 iterations and **mu** = 0.0025. Tap weights were found by repeatedly iterating through the 32-symbol sequence (**y**) extracted from the prelude and slowly adjusting the values for **w** to gradually lower the mean square error between **w**·**y**. After completion of the algorithm, **w** was then shifted to have the highest value in the middle to ensure the weights were time aligned, and **w** was used to obtain an equalized baseband signal **xBBe** by multiplying **xBBd** by the final values of **w** that looped front to back using the modulus operator.

Finally, the preamble was detected in **xBBe** and the payload data was extracted. The receiver was run using sample data **xRF1**-**xRF5** and their corresponding answers again yielded text that was a series of puns and answers. The eye patterns of **xRF2**-**xRF5** and their corresponding payloads are found in **Figure 3.** As seen in these figures, the payload eye patterns become rougher with more variation around the QPSK values, but the equalization worked well enough that none of the data was lost. To better deal with the timing phase, the calculation of **rhot** was changed to a method that required taking the FFT of xBBf to estimate the first two terms in a Fourier series (**rho0 and rho1**) and using them to calculate **rhot** instead. This yielded better results for the samples with higher noise. As the signal samples deviated from the timing phase found through power maximalization, the cyclic equalizer still worked, but the end results were not as accurate, as seen in the eye patterns. However, because of the Gray mapping, the symbols were still read correctly from the input signal.

Part III – Fractionally-spaced equalizer

For **Part III**, the symbol-spaced equalizer of **Part II** was replaced with a half-symbol spaced equalizer and the difference in performance between the two was evaluated. Firstly, because a fractionally-spaced equalizer is insensitive to the timing phase, the timing recovery functionality of **Part I** was removed. Instead, **xBBd** was obtained by decimating the baseband to twice the symbol rate, or by a decimation factor of L = 50.

Next, just as in **Part II**, $|\mathbf{r}_{yy}|$ was calculated and one of the peaks in the preamble was picked as a starting point to estimate the tap weights **w**, which were estimated using the NLMS adaptation algorithm. Because **xBBd** was decimated at twice the sample rate, 2**N** (64) values were sent to the tap weight estimator. The length of **w** was also twice the length of the pilot **cp**, i.e. 64 symbols. Tap estimation was similar to **Part II** except the adaptation algorithm is iterated every two clock cycles, as seen by the

modulus operator on line 144 of partIII.m. **xBBd** and **w** were then sent to a fractionally-spaced equalizer, which functioned similarly to the equalizer of **Part II**. The highest magnitude of **w** was also centered like **Part II**.

Finally, as in the other parts, the payload was located on **xBBe** and the data extracted into a stream of bits. The half-symbol spaced equalizer was tested with test files **xRF1-xRF5**, which yielded the text of the same puns as before. A few characters were changed around in the final text of some of the test files with higher noise, but this proved to only be a small problem and most of the data was perfectly intact. The eye patterns for the payloads in **xRF2-xRF5** using **Part III** are found in **Figure 4**. As seen in the figures, eye patterns show symbols fairly close to the ideal QPSK Gray mapping. The eye patterns in **Figure 7** are a little rougher than those in the previous two parts, but it seems like the fractionally-space equalizer is still doing a good job of retrieving the data from the signal, even if the timing phase is not being tracked.

To test if **Part III** could really work independent of the timing phase, the eye pattern of xRF5 was examined for different values for the variable **timing_phase_start**, which was used to choose the first value for decimation. The eye patterns for the payloads were virtually identical no matter what starting point was used for the timing phase.

The mean square error (MSE) of the equalizers in **Part II** and **Part III** for **xRF5** are compared in **Figure 5**. As seen in the **Figure 5(a)**, the MSE in the symbol-spaced equalizer drops as the algorithm runs through its iterations and settles at a value near 0. However, as seen in **Figure 5(b)**, the half symbol-spaced equalizer only drops to a value around 0.75 before it begins decreasing roughly linearly. It appears tracking the symbol rate in **Part II** greatly reduced the minimum error achieved by the adaptation algorithm.

Part IV – Carrier acquisition and tracking

For **Part IV**, carrier acquisition and tracking were added to the receiver of **Part II** to estimate and adjust for offsets in the carrier frequency (**Dfc**) and phase (**phic**). Firstly, **Dfc** is estimated by using pilot aided carrier recovery. To estimate the value for **Dfc**, a complex-valued sum **J** must be found from interval **N1** to **N2**. **N1** is found by looking at the preamble to find the end of initial transience in the beginning of the signal **xBBd**. The end in transience was measured by looking for the point at which the deviation between y(i) and y(i+N) drops below a certain threshold. A value of 5% seemed to be a satisfactory threshold for the test files in this part. Because **J** must be found in a periodic symbol, **N2** = **N1** + **N** (length of the pilot **cp** or 32), **Dfc_est** is found by using the angle of **J** along with **N** and $\mathbf{T_b}$. The offset is corrected for by multiplying **xBBd** by exp(-j·2π·Dfc_est·t1) to get **xBBo**.

Next, **xBBo** is sent though the cyclic equalizer as in **Part II** to yield the equalized baseband signal **xBBe**. The phase offset is estimated from **xBBe** using a decision-directed phase recovery loop. The recovery loop tracks the phase error between the measured input **s1** and an expected value **s2**. The phase error between **s1** and **s2** is continually adjusted until the end of **xBBe** is reached. After **phic** has been estimated, the phase offset is adjusted by multiplying **xBBe** by exp(-j·phic) to get **xBBp**.

Finally, the data payload of **xBBp** is detected and the QPSK symbols are converted into a stream of bits using Gray mapping. The test data for **xRF6-xRF8** includes frequency and phase offset errors. By running the test data through the receiver in **Part IV**, the offsets were successfully adjusted for and the data of the text files was recovered, which all included a series of puns or other jokes. The contents of all the test files were successfully recovered without any loss of information, even in cases such as **xRF8** where there was a large amount of frequency and phase offset. **Figure 6(a)-(c)** show the eye patterns of the payload portion of **xBBp** for test files **xRF6-xRF8**. Although the eye patterns in these figures are messier than before, the payload still just contains QPSK symbols and the data is still able to be successfully retrieved despite the large carrier and phase offsets.

Part V – Symbol rate tracking

For **Part V**, the receiver of **Part I** was adapted to be able to track and recover an unknown frequency offset in the symbol rate introduced in the transmitter using a decision directed timing phase recovery method. Because the timing phase is not constant, it is no longer possible to recover the timing phase by looking for the maximum value in **rhot**. Instead, the symbol phase of **xBBf** was found by running a timing recovery algorithm that attempts to minimize the mean square error of transmitted symbols. The output of the symbol rate tracker is saved as **xBBd**, and the payload is detected as before and its data symbols converted into a stream of bits.

First, the receiver of **Part V** was tested with **xRF1**, where no timing phase offset is present. The text of the data was recreated without any errors. Next, test data **xRF9** was used, which contains a signal with considerable variance in the symbol rate. The receiver in **Part V** was also able to correctly recreate the data, which this time was a very lengthy collection of jokes. The eye patterns of **xBBf** and the payload of **xBBd** is found in **Figure 7**. As shown in the figure, the payload is comprised of a series QPSK symbols with only a little variation. It appears as if the symbol rate tracking is working as desired.

To compare these results with a receiver without timing phase recovery, **xRF9** was run through the receiver of **Part I,** which was able to successfully generate a QPSK symbol payload. However, although the first few sets of characters were correct in the final text file, the text soon became unreadable as the file went on. The offset must have gone back to near zero near the middle of the payload, because the text became recognizable again. However, the text after this point also became increasingly more corrupted again and the rest of the file was illegible. The results of **Part V** show that this symbol rate variation was successfully tracked and adjusted for by the timing recovery algorithm.

Conclusion

The QPSK receiver of **Parts I** to **V** was a success. All the test files were able to be successfully demodulated and decoded and all the jokes in the test files were able to be read. There were only some very slight errors in **Part III**, but this resulted in only a changed character or two. The rest of the test files were able to be read without the presence of any errors at all, even when faced with test files with a large amount of noise, distortion, or offsets of various types and sizes.

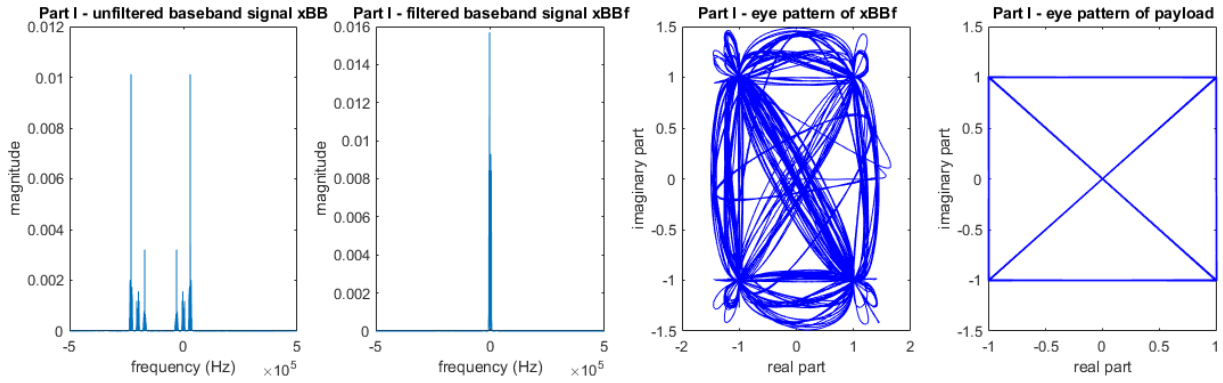

Figure 1. Part I - Frequency bands and eye patterns. (a) spectrum of unfiltered baseband signal xBB. (b) spectrum of baseband signal xBBf with out of spectrum frequencies filtered out. (c) eye pattern of filtered baseband signal xBBf for xRF1. (d) eye pattern of payload on xRF1.

Bits	Symbol
0 0	-1 – j
0 1	-1 + j
1 1	+1 + j
1 0	+1 – j

Table 1. QPSK Gray mapping.

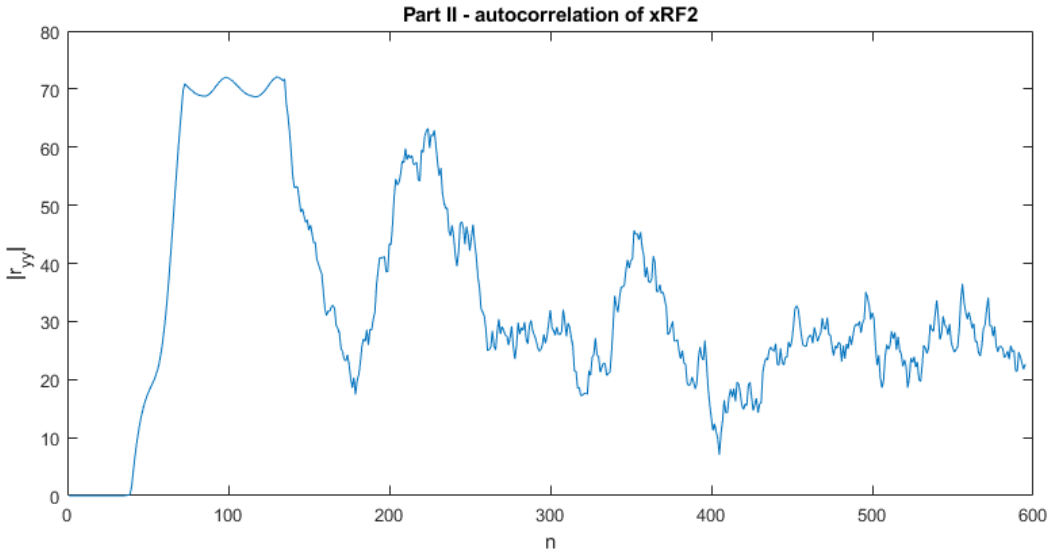

Figure 2. Part II - Absolute value of autocorrelation of xRF2 used to find beginning of preamble sequence for cyclic equalizer.

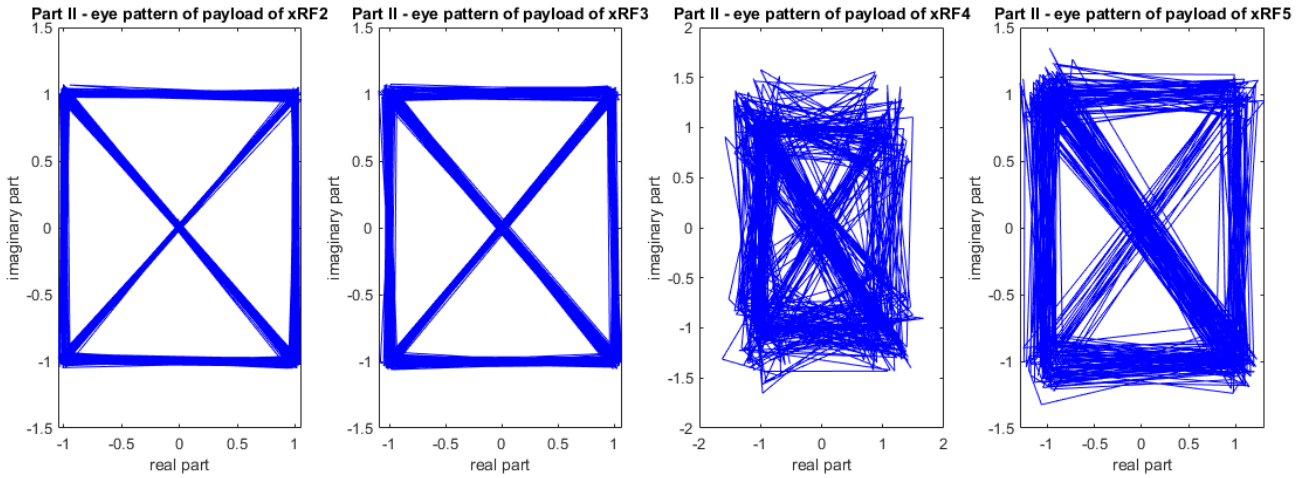

Figure 3. Part II - Eye patterns of payloads (a) xRF2. (b) xRF3. (c) xRF4. (d) xRF5. Despite the presence of distortion and noise, the symbols fall around the values expected for QPSK.

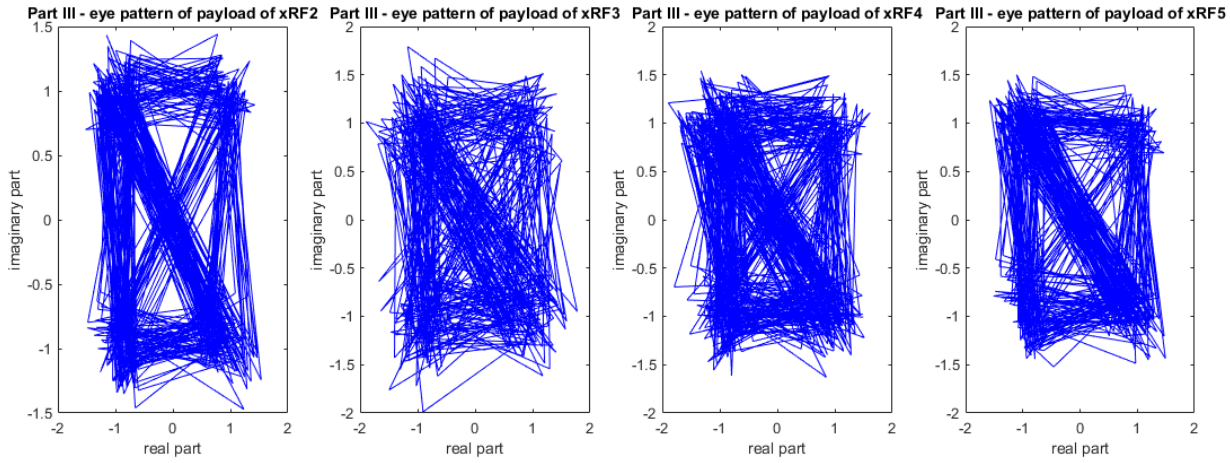

Figure 4. Part III - Eye patterns of payloads (a) xRF2. (b) xRF3. (c) xRF4. (d) xRF5. Despite the presence of distortion and noise, the symbols fall around the values expected for QPSK. There is more noise in the eye patterns of Part III than Part II.

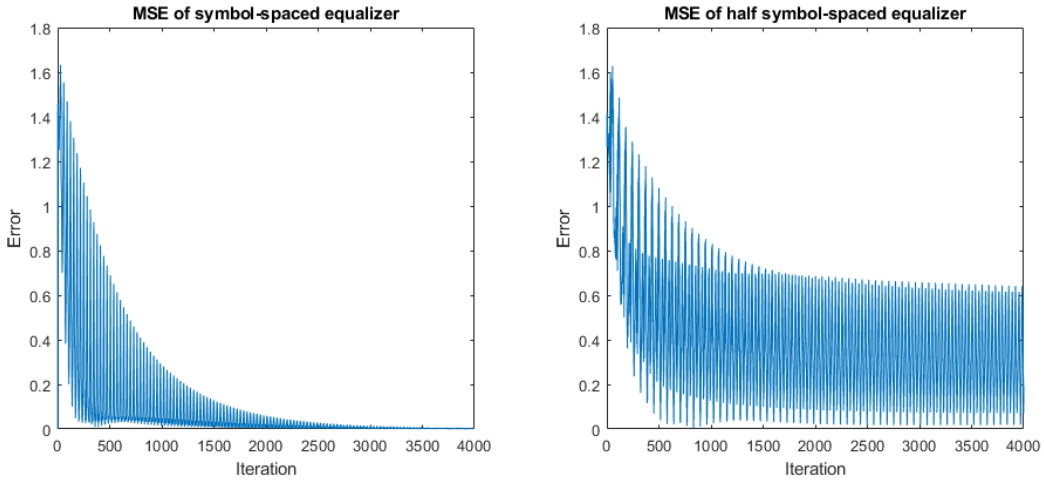

Figure 5. Mean square error of (a) symbol-spaced equalizer and (b) half symbol-spaced equalizer. MSE drops to a lower final value in symbol-spaced equalizer.

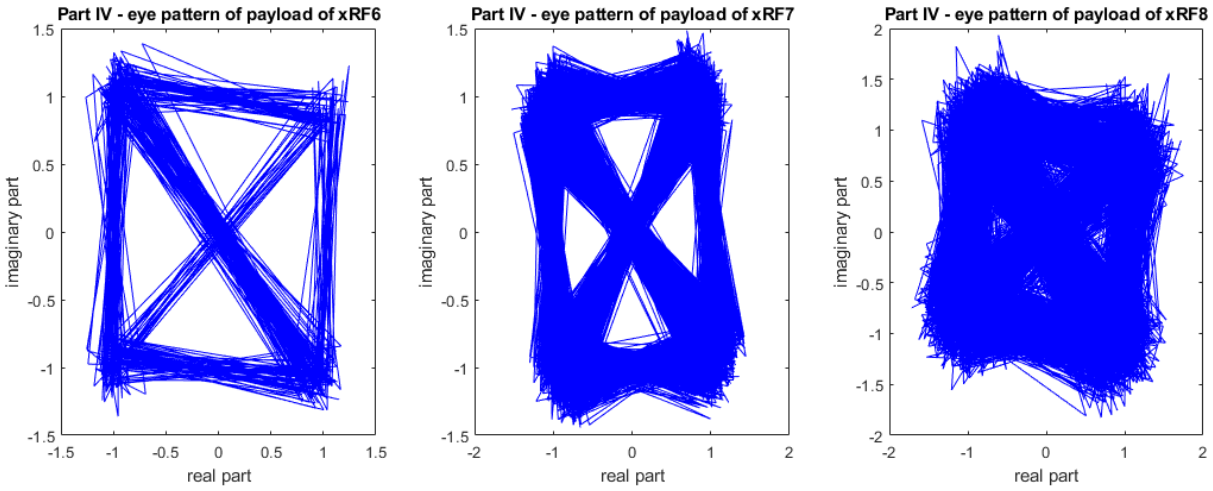

Figure 6. Part IV – Eye patterns for payloads in (a) xRF6. (b) xRF7. (c) xRF8. The carrier frequency and phase offsets create substantial noise in the payload even after equalization. Still, the QPSK symbols were still maintained and the data was read correctly.

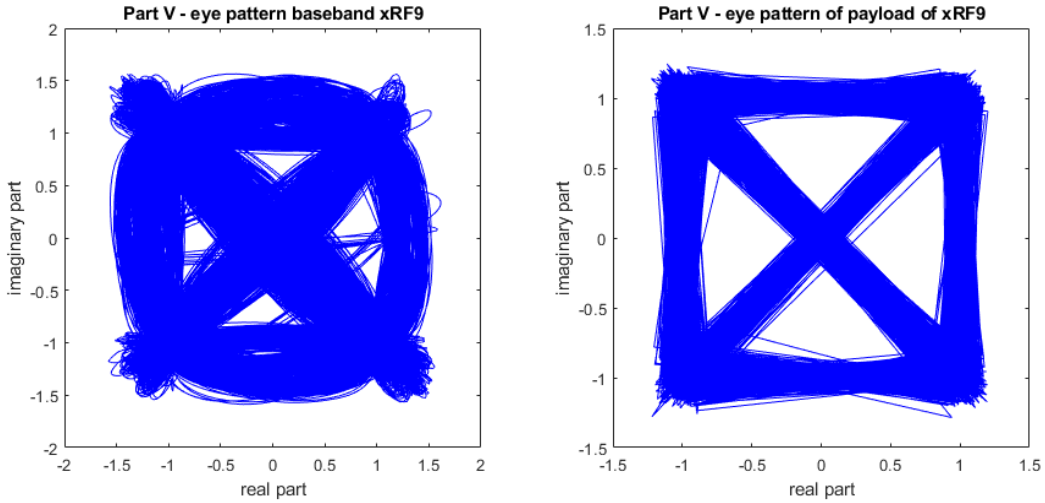

Figure 7. Part V – Eye patterns for (a) filtered baseband signal xRF9. (b) payload of timing-corrected payload for xRF9. The timing phase was correctly tracked and accounted for and the data was recreated without any errors.