Lab 3
Binary Baseband Digital Communication System

Christian Chiarulli
Tug19562@temple.edu

Summary
In this lab we will be modeling a binary baseband digital communication system. Two signals will be sent representing a binary zero and one. The zero will be represented by a half sine wave and the one will be represented as a ramp function. Both signals will share the same amplitude and period. The signals will be sent randomly every bit time. The the signals will be then be received by a single correlator receiver which will check if valid information is being sent.

Introduction
The parameters for the experiment can be found in table 1. The bit time will be 400 microseconds which will represent the period for each binary signal. The amplitude should reach amplitude of 2 for each signal.

Table 1

Amplitude	2V
Bit time T_b	$400*10^{-6}$ s

To transmit the signal a random integer will be sent as either a zero or a one. This will have to pass through a sum block to add one. This is because the multiport switch chooses between a one and a two to delineate between the two waveforms. The one will represent the half sine wave and the 2 will represent the ramp. The output of the multiport switch and a reference source developed from the two signals will be fed into a single correlator receiver. The output of the multiport switch will then be multiplied by the reference signal $\widehat{\phi_2}$. The signal will then be sent through an integrate and dump block with the output intermediate values box unchecked. This will produce pulses that will hold their value until the integrate and dump block resets. This is in direct contrast to how the ramp was developed by outputting the intermediate values to get each iterative step for the ramp function. This block also produces a delay which will be compensated for in the error calculation block later. The threshold T will be computed for equal probability events which will drop the second term from the typical threshold equation. This will be added after the integrate and dump block. Next the signal will pass through a sign block which will convert all positive values to ones and all negative values to negative ones which effectively normalizes the signal. Then the signal will be passed into the lookup table to pull the negative values up to zero. When passed through a receiver without additive white gaussian noise (AWGN) and after correcting for the delay the signal should match the originally transmitted binary sequence.

From MATLAB's documentation on the integrate and dump block:
"If the input is a scalar value, then the output is delayed by one sample after any transient effect is over. That is, after removing transients from the input and output, you can see the result of the *m*th integration period in the output sample indexed by m+1."

Discussions

Task 1

Running the transmitter for one bit time resulted in the wave form found in figure 1. The amplitude is equal to 2 for all transmissions and the the period is 400 microseconds. The ramps represent binary 1's and the half sine waves represent binary 0's.

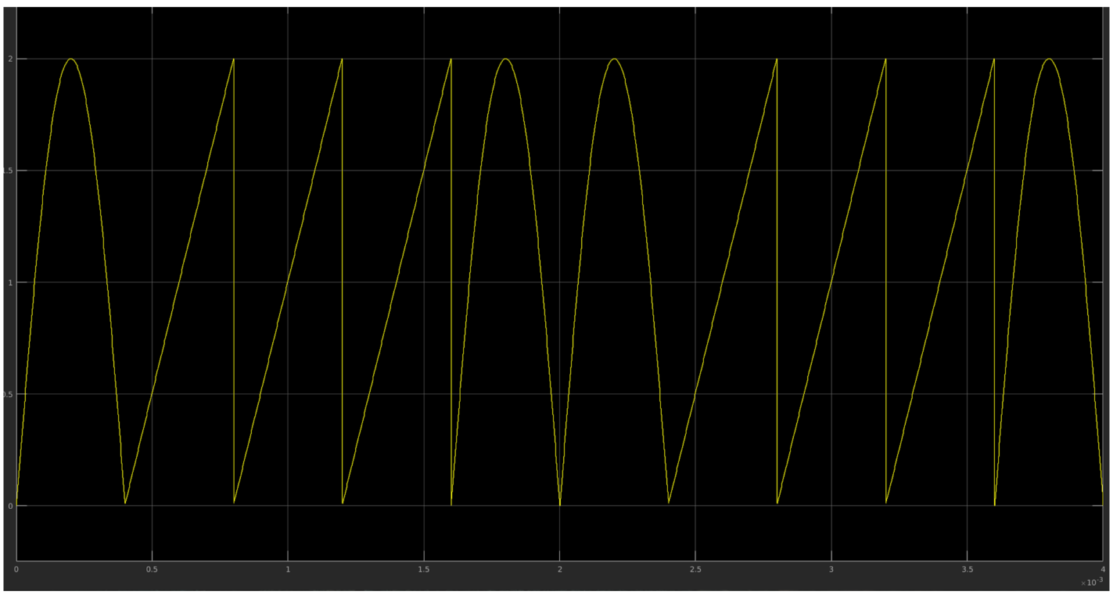

Figure 1

Task 2

The following calculations must take place before obtaining $\widehat{\phi_2}$:

$$E_1 = \int_0^{400\cdot10^{-6}} (2sin(\pi2500t))^2 dt \;=\; 8\cdot10^{-4}$$

$$E_1 = \int_0^{400\cdot10^{-6}} \left(\frac{2t}{400\cdot10^{-6}}\right)^2 dt \;=\; 5.333\cdot10^{-4}$$

$$\rho = \frac{1}{\sqrt{8\cdot10^{-4}\cdot5.333\cdot10^{-4}}}\cdot\int_0^{400\cdot10^{-6}} 2sin(\pi2500t)\cdot\frac{2t}{400\cdot10^{-6}}dt = 0.7797$$

$$\widehat{\phi_2} = \frac{s_2(t) - s_1(t)}{\sqrt{5.333 \cdot 10^{-4} - 2(0.7797)\sqrt{8 \cdot 10^{-4} \cdot 5.333 \cdot 10^{-4}} + 8 \cdot 10^{-4}}}$$

Where: $s_2(t) = \frac{2t}{400 \cdot 10^{-6}}$ and $s_1(t) = 2sin(\pi 2500t)$

And the coefficient of $\widehat{\phi_2}$ will be 56.367. This value will be put into a gain block to be configured after subtracting the values in simulink. The simulink model for this can be found in figure 2 and the plot for $\widehat{\phi_2}$ can be found in figure 3.

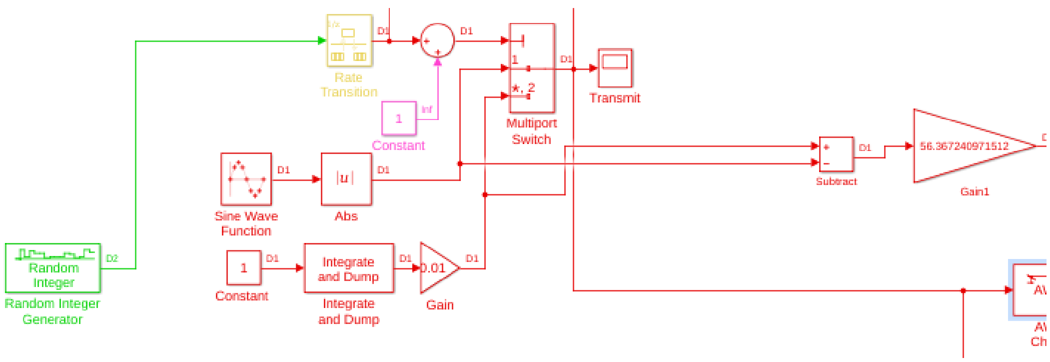

Figure 2

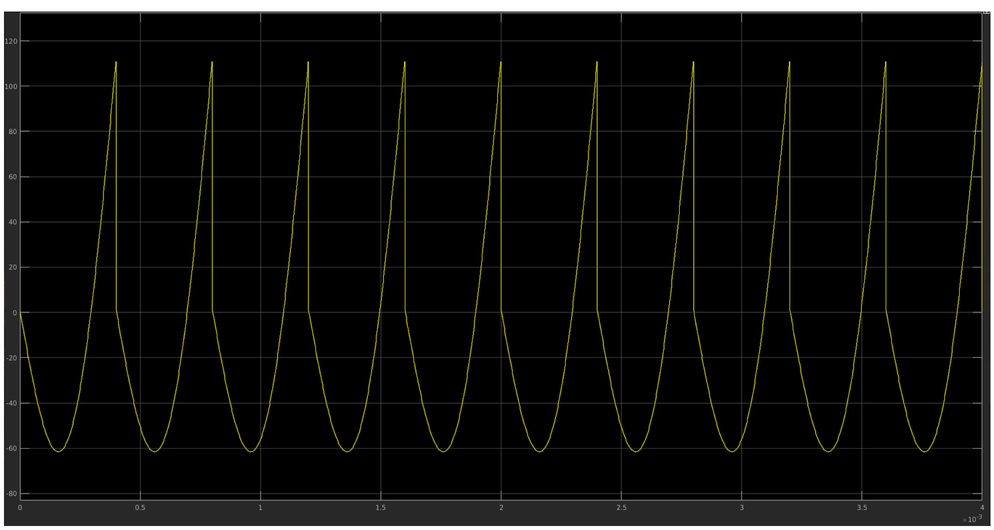

Figure 3

This waveform represents $\widehat{\phi_2}$ and can be confirmed to be correct by taking the integral of $\widehat{\phi_2}$ which should be equal to 1.

$$\int_{0}^{400 \cdot 10^{-6}} (56.367 \cdot (\frac{2t}{400 \cdot 10^{-6}} - 2sin(\pi 2500t)))^2 dt = 1$$

Task 3

Threshold T is to be computed with $P_1 = P_2$ so the second term in the calculation is dropped leaving the following equation:

$$T \equiv \frac{\hat{s}_{22} + \hat{s}_{12}}{2} + \cancel{\left(\frac{N_0/2}{\hat{s}_{22} - \hat{s}_{12}}\right) \ln\left(\frac{P_1}{P_2}\right)}$$

Before calculating and adding T the integrate and dump block must first be scaled, this entails dividing by N number of samples as per MATLAB's documentation:

"If the input is a scalar value, then the output sample time is *N* times the input sample time and the block experiences a delay whose duration is one output sample period."

in the integrate and dump block and multiplying it by the bit time 400 microseconds. This is because the integration is replaced by a summation and must still be multiplied by dt which in this case is the bit time. The threshold is then calculated as:

$$\hat{s}_{12} = \int_0^{400\cdot 10^{-6}} \left(2sin(\pi 2500t) \cdot 56.367 \cdot \left(\frac{2t}{400\cdot 10^{-6}} - 2sin(\pi 2500t)\right)\right) dt = -0.01639$$

$$\hat{s}_{22} = \int_0^{400\cdot 10^{-6}} \left(\frac{2t}{400\cdot 10^{-6}} \cdot 56.367 \cdot \left(\frac{2t}{400\cdot 10^{-6}} - 2sin(\pi 2500t)\right)\right) dt = 0.001355$$

$$T = \frac{-0.01639 + 0.001355}{2} = -0.0075156$$

To confirm the calculations a scope can be added directly after the integrate and dump block, the peaks should match the calculated $\hat{s}_{12}$ and $\hat{s}_{22}$, the first will represent a zero and the second will represent a 1. The simulated values can be found in figure 4.

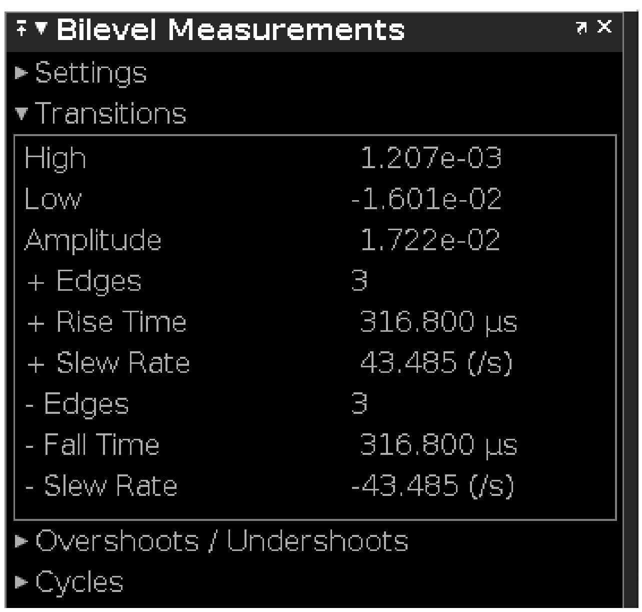

Figure 4

Taking the average of the high and low values: (1.207e-3 + -1.601e-2)/2 = -0.007403 which is very similar to the calculated threshold confirming further the calculation with error being attributed to rounding difference between simulink and my hand calculations.

Task 4

Configuring the single correlator receiver with the reference source $\hat{\phi}_2$ with the the receiver is shown in figure 5 the AWGN variance is set to 0 when testing the communication system in a vacuum without the introduction of error.

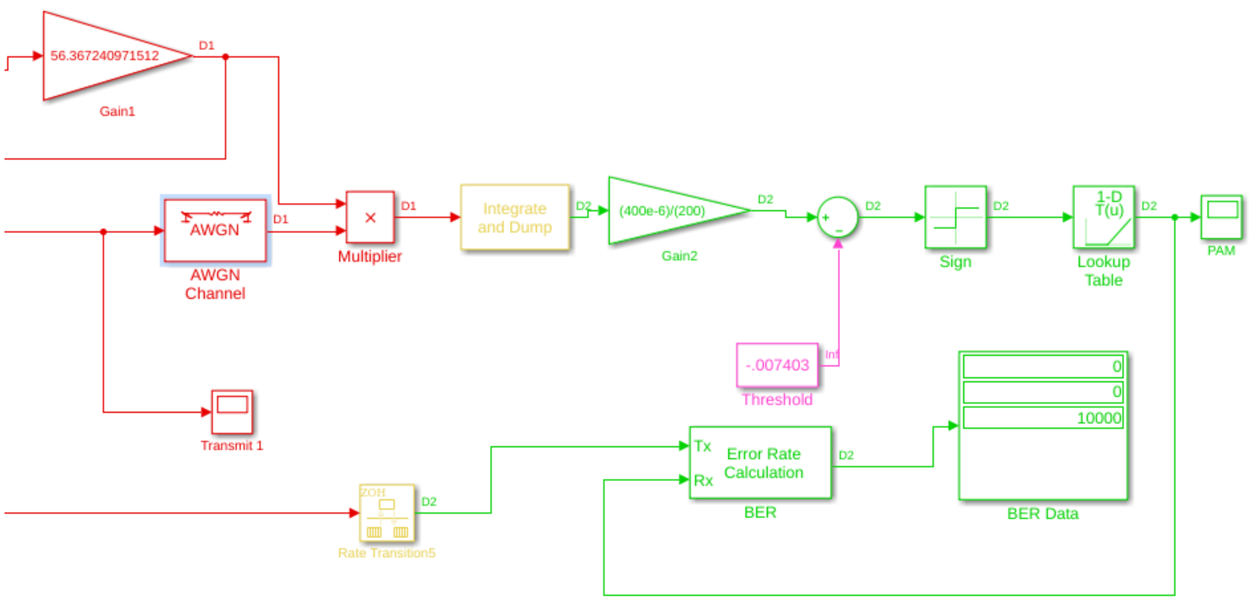

Figure 5

Task 5

The diagram is configured with the constant T found in simulation being added after the integrate and dump. This is then fed into the sign block to convert all positive values to 1's and negative values to -1's this can then be passed into the lookup table to pull the -1 values up to 0 which should match the input signal while taking the shift from the integrate and dump block into account.

Task 6

Figure 5 also shows the Error rate calculation block which must be configured to have a receive delay of 1 due to the integrate and dump. The output from the 1D lookup table can then be fed in and compared to the original signal which must be downsampled to from the 2 microsecond simulation time to 400 microseconds (the 1D lookup output was already downsampled effectively due to the integrate and dump block down sampling to the bit rate).

Task 7

Table 2 shows the introduction of the AWGN for different values of σ^2 and figure 6 shows the received and transmitted signals for 0 AWGN to verify best case scenario performance where no bits should be received in error. The received signal will be shown to be shifted by 1 bit which will be corrected by the shift in the BER block.

Table 2

σ^2	BER (single correlator receiver)
0	0
2.5	0
5	0.0029
10	0.0239
20	0.078
100	0.2669
500	0.3972

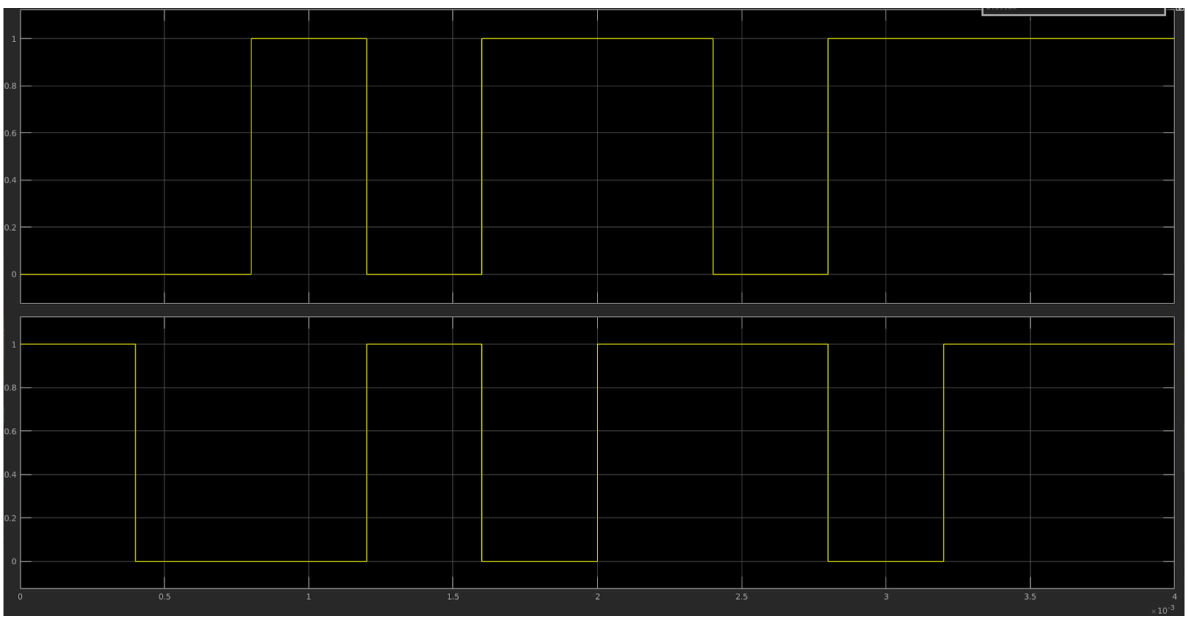

Figure 6 (Top transmitted, Bottom received)

Conclusion

The lab seemed to be a success as the values seemed to be confirmed in both hand calculation and simulation. From table 1 errors seemed to be introduced around when σ^2 = 5. It may be a good exercise to check these same conditions in a LPF receiver to see how they differ for different values of σ^2 to get an idea of how well this receiver compares to other methods. Also you may have noticed the color coding in the simulink models the red represents sampling at the fixed sampling rate 2e-6 and the green at the bit time 400e-6, the yellow represents transitions from one rate to another and the pink blocks are constants sampled at infinity this was very helpful for debugging the models.