**EE 460L**

**COMMUNICATION SYSTEMS**

**LABORATORY FINAL PROJECT: AMPLITUDE SHIFT KEYING**

**DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**

**UNIVERSITY OF NEVADA, LAS VEGAS**

**BY**

**MARIO VALLES**

**JASON SILIC**

**LAB REPORT:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Goals**: Is to be able to transmit more than one signal through a single channel using TIMS. This would be done by multiple amplitude signaling. Also, to check the errors that might happen by with noise and/or signal distortion.

**Roles:**

Jason Silic gave the idea and implemented the project. The M-ary and the error detection.

Mario Valles helped construct the M-ary and wrote the report and presentation.

**Theory of operation:**

When a digital signal is send normally is in a binary format of 0-1. However, this does not have to be the always the case, it can be M-ary format being M^2 the signal format. Here is an example:

Having figure 1(a) as a regular binary sequence Figure 1

and figure 1(b) as an M-ary with M = 4 = 2^2.

Being the number describes 00, 01,10, 11. This is

Very beneficial since we can send a 8-bit by a single

pulse.

However, as M increases the power increases and

also noise increases. This will give also an increment

in signal errors.

**Schematic/Block Diagram:**



**Circuit Operation:**

First two sequence generators are need to produce the signals.

The clock of both sequence generators were connected to a single clock, which in this case was a function generator.

After that the output of each of them were added together by an adder as the following figure shows:





 output

The gain of the adder was adjusted to have three values -3V, -1V, 1V and 3V. This way the M-ary was constructed.



Ch1-Ch2 Ch3

00 3 V

01 1 V

10 -1 V

11 -3 V

Then the output of the adder is connected to a low pass filter to simulate better an actual signal. First we configure the LPF to have a gain of 1



 To -1 multiplier

We got the following results for different filter frequencies. This would simulate a real signal







After that we compare both channels with reference. In the case of channel 1 ref. was 0V, but channel 2 ref. was -2V.

This figure is signal 1 after the -1 multiplication.







 GROUND

 OUTPUT

 -1





 -2V

 GAIN 0.8 OUTPUT

 COMPARATOR 1

 GAIN 1

 -1

Having all outputs together we reconstruct the signal.

****

Then we wanted to know how many errors there are for different frequencies. So we did the following configuration



 Input of Ch. 1 or 2

 XOR output

 Output of comparator

 1 or 2

 To TTL counter

 Clock

For 1.5 kHz



We got the following result



However this is because the adder and the -1 multiplier add delay on the signal



So, we can say that in reality that there is no error in the output signal with 1.5kHz clock signal.

Next the clock frequency was increased to 2.5 kHz. For these experiments the XOR gate counted errors for 10^4 clock periods.

At the first filter cutoff frequency we tried there were 625 errors in sequence 2, and 1563 errors in sequence 1. The reason for this is that sequence 1 has less "margin."



 Here is an example

 of an error counting

At a lower filter cutoff frequency there were 2180 errors for sequence 2 and 5936 errors for sequence 1. The low pass filter gets rid off the high frequencies which make the signal become more sinusoidal. This causes a lot more errors.



**Conclusion and Discussion of Results:**

The M-ary transmission is great because it saves channels and can use the same bandwidth. The problem is of course it has a more complicated receiver and transmitter. Also, M-ary is beneficial because it can send a word with a voltage value let say a 8-bit would be 8 volts, which is great. The problem is that more power is use and more errors are created. This is because the probability of get an error increases since now there will be no only zero and one, but M numbers.

*Lab report presented by:*

Mario Valles\_\_\_\_\_\_\_\_\_\_

Jason Silic\_\_\_\_\_\_\_\_\_\_\_\_