**ESE-3014 Lab1**

**Review Octave**

# **Theory**

GNU Octave is a high-level language, primarily intended for numerical com- putations. It provides a convenient command line interface for solving linear and nonlinear problems numerically, and for performing other numerical exper- iments using a language that is mostly compatible with Matlab. It may also be used as a batch-oriented language.

Octave has extensive tools for solving common numerical linear algebra prob- lems, finding the roots of nonlinear equations, integrating ordinary functions, manipulating polynomials, and integrating ordinary differential and differential- algebraic equations. It is easily extensible and customizable via user-defined functions written in Octave’s own language, or using dynamically loaded mod- ules written in C++, C, Fortran, or other languages.

**analog signal processing**

# **Task**

1. Create a 5x1 vector of zeros. Create a 2x5 matrix of random numbers.
2. Multiply a column of a matrix with an element of this same matrix.
3. Create a plot of the sin function between 0 and 6*π*.
4. Simulate an amplitude modulation (AM) system with all input, carrier and output signals. Say the input signal is a cosine wave with amplitude as 2V and frequency as 1000Hz. The carrier signal is also a cosine wave with amplitude as 5V and frequency as 10KHz. The modulation degree is 0.5, and the initial phases of all cosine wave are 0. (Recall Nyquist sampling theorem to avoid distortion i.e. under sampling)
5. Use the signals above, consider a actual vivid simulation mode, and add random noise in output signal. In this simulation, we divide time domain into several duration, and call each duration as frame. The scanning cycle of an oscilloscope is equal to frame period, that means each time we simulate a frame of signal, and the display will be refreshed once. Therefore, we can get a constantly sliding input signal, a carrier signal with phase jitter, and output signal with noise.

**CDMA**

# **Task**

1. Simulate Code Division multiplexing to achieve multiple access in a shared medium.

2. Follow the process on today's course, you can use our Walsh matrix. Create three data sequences for three users, encode and decode with the code you chose for all of them from matrix. You don't have to use pn sequence, m sequence, and gold sequence.

**The transmitter of a PCM system**

# **Task**

Simulate all operations performed in the transmitter of a PCM system include steps below. You can choose the techniques we introduced in the course, and please show your code in text form and screenshots of each steps.

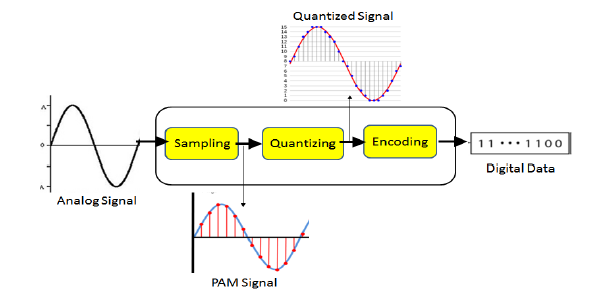


Figure 1: Pulse-code modulation

1. Analog signal: it could be a sin or cos wave.

2. low-pass \_lter (optional)

3. sampling: Use the result from last step, and sample it with a proper sampling frequency.

4. quantizing: You can apply Uniform quantization or Non-uniform quantization, and you should indicate which type you used.

5. encoding: Please refer the encoding from our slides, choose and indicate the line code can be used for electrical representation of binary symbols "1" and "0". (e.g. on-o\_ signaling, NBZ, RZ, and so on)

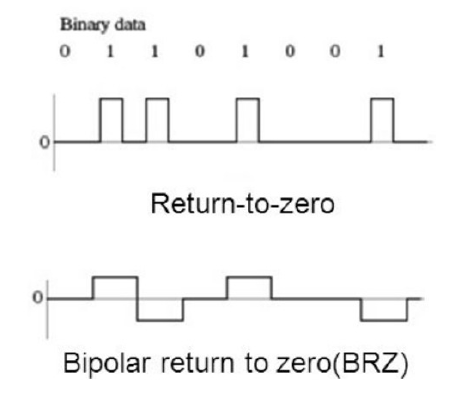
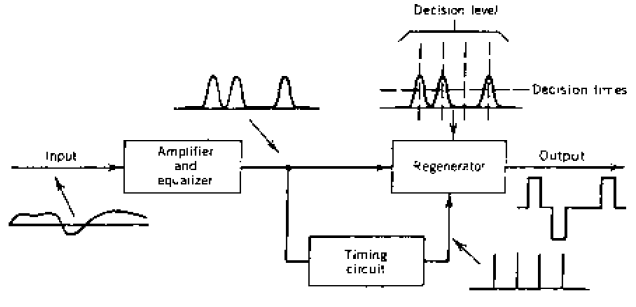


Figure 2: Line code for encoding

**The receiver of a PCM system**

# **Task**

Simulate the operations performed in the regeneration circuit of the PCM system receiver(like the figure below) include input, amplifier equalizer, timing circuit, decision making device, and output. You should add appropriate noise in distorted signal (in your input), and calculate the bit error rate at the end of receiver.

 Figure 3: Pulse-code modulation regeneration circuit (with using BRZ in Output coding)