Computational homework

August 30, 2023

Simulink-based implementation of an M-QAM modem

Digital communication systems

Telecommunication Engineering

André Lima Ferrer de Almeida

Problem 1 Develop a Matlab function that calculates a bit sequence in Gray encoding. It must input the number of bits (an integer value) and output the bit sequence. For example, for a sequence of 3 bits, your function must return

```
    >> get_gray_codes(3)

ans =

     0     1     1     0     0     1     1     0
     0     0     1     1     1     1     0     0
     0     0     0     0     1     1     1     1
```

where each column corresponds to a sequence (from the left to the right) of 3 bits in Gray encoding. The first row is the least significant bit (LSB) and the last row is the most significant bit (MSB).

Tips:

1. Before generating the Gray encoding, first generate the bit sequence without any encoding, i.e.,

```
>> original

original =

     0     1     0     1     0     1     0     1
     0     0     1     1     0     0     1     1
     0     0     0     0     1     1     1     1
```

2. Transforming the uncoded bit sequence into Gray encoding is easier if you use a recursive function. See here.

Note: For the following problems, use the Simulink template provided by the professor.

1 Baseband modulator

Problem 2 Generate a bit stream with the following parameters:

- Bit rate: $R_b = 10\,\text{Mbit/s}$.
- Equiprobable bits.

Problem 3 Use a block called "Matlab function" to create a function that maps a stream of $N_b = \log_2(M)$ bits to a complex symbol from an alphabet of $M = 256$ symbols. The real and imaginary part of each constellation point must be spaced as Proakis suggests, i.e., $\left\{\pm 1, \pm 3, \ldots, \pm\sqrt{M}\right\}$. Moreover, the mapping of the real/imaginary part of the complex symbol must be in the Gray encoding sequence (see [3] for more info).

(a) What is the operating rate at the output of "Matlab function"? What is its relationship with N_b and R_b?

(b) Use a block called "Gain" to apply a gain of $\sqrt{E_g}$ to the complex symbol outputs, where $E_g = 5$.

(c) By using a block called "Constellation Diagram", show that each point in the constellation is now in the form $\sqrt{E_g}A_m$, where $A_m = A_{mi} + jA_{mq}$ is the constellation for the mth symbol and that $A_{mi}, A_{mq} \in \left\{\pm 1, \pm 3, \ldots, \pm\sqrt{M}\right\}$. Note that the kth transmitted symbol can be represented by the vector

$$\mathbf{s}_m[k] = \left[\sqrt{E_g}A_{mi} \quad \sqrt{E_g}A_{mq}\right]^\top \in \mathbb{R}^2 \tag{1}$$

Tips:

1. Use the "Buffer" block to vectorize the bit stream into an N_b-bits frame and use it as input of the "Matlab function".
2. You can pass a Gray-encoded bit sequence to the "Matlab function" block. You can use your function developed in Problem 1 to generate this bit sequence.

Problem 4 Pass your symbol stream through the "Raised Cosine Filter" block with the following parameters

- Roll-off factor: 0.2.
- Filter span (in symbols): 4.
- Number of samples per symbol: $N = 100$ samples.

The filter gain must be adjusted so that the energy of the impulse response is one. In the "Raised Cosine Filter" block, you can obtain the impulse response by exporting the filter coefficients to the workspace. You can use the `stem()` function to plot the impulse response samples. Use the function `trapz()` to perform a numerical integration via the 1th-order Newton-Cotes formula (i.e., the trapezoidal method). By integrating the square of the impulse response, you obtain its energy. If the result is I, you can normalize it by adjusting the filter gain to $1/\sqrt{I}$.

Note: In fact, the energy of the baseband (BB) pulse shaping has already been included separately in the "Gain" block (its energy is E_g).

(a) What is the new operating rate of the system? Why has the operating rate changed to this value?

(b) Check if the raised cosine filter obeys the Nyquist criterion for non-intersymbol interference (ISI). Use the "scope" block to prove your answer graphically.

2 Upconverter

Problem 5 Use a block called "sine wave" (from the DSP system toolbox) with the following parameters

- Amplitude: $\sqrt{2}$;
- Frequency: $f_c = \frac{1}{4T_{samp}}$, where T_{samp} is the sampling interval of the baseband signal;

- Sample mode: Discrete;
- Output complexity: Complex;

The sampling rate must be consistent with the sampling rate of your system. Let us denote the BB pulse shaping (i.e., the raised cosine filter impulse response) of your system as $g(t) \in \mathbb{R}$, whose energy is denoted as E_g. Then the pre-envelope (or analytic signal) of the mth symbol is

$$s_{m+}(t) = s_{ml}(t)\, e^{j2\pi f_c t} \in \mathbb{C}, \tag{2}$$

where $s_{ml}(t) = A_m g(t) \in \mathbb{C}$ is the complex envelope (or lowpass equivalent) of the real-valued transmitted signal for the mth symbol,

$$s_m(t) = \sqrt{2}\,\mathrm{Re}\left\{s_{ml}(t)\, e^{j2\pi f_c t}\right\}. \tag{3}$$

The pre-envelope has important properties concerning the spectrum of the transmitted signal. The output of the block named "sine wave" should exactly be the discrete-time version of the pre-envelope discussed in the literature.

(a) Make a mathematical analysis of this signal and use the "Spectrum Analyzer" block to compare the spectrum obtained in Simulink with the theoretical spectrum. Good references can be found in [1–3]

(b) Repeat the item (a) for the spectrum of the complex envelope.

(c) Use a block called "Complex to real-imag" to obtain the real part of the analytic signal and repeat the item (a) for the transmitted signal.

(d) Based on [3, Equation (3.2-42)], the average energy of the transmitted signal is[1]

$$E_s = \frac{2(M-1)}{3} E_g \tag{4}$$

For $M = 256$ and $E_g = 5$, we have that $E_s = 850$.

In reality, the transmitted signal is a stochastic process and therefore it is seen as a power signal since its energy is infinity. However, the concept of "average energy" expressed in (4) does not refer to the energy in its original sense, i.e.,

$$\lim_{T\to\infty} \int_{-\frac{T}{2}}^{\frac{T}{2}} |s_m(t)|^2 \,\mathrm{d}t\,. \tag{5}$$

[1] Proakis does not normalize the carrier energy by the $\sqrt{2}$ factor. Therefore, the average energy in his book is half of what we are using here.

Rather, the average energy is obtained from the product between the average power,

$$P_s = \lim_{T \to \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} |s_m(t)|^2 \, \mathrm{d}t \,, \tag{6}$$

and the symbol interval, T_s. Therefore,

$$P_s = \frac{E_s}{T_s}. \tag{7}$$

The equation (6) converges and can be estimated in Simulink by using the blocks:

1. "Square" to take the square absolute value of the transmitted signal.
2. "Discrete-time integrator" to numerically integrate $|s_m(t)|^2$.
3. "Clock" to obtain an input of the running simulation time.
4. "Divide" to divide the integration result by the running simulation time.

Finally, use the block named "gain" to multiply the average power to T_s and show that, indeed, the average energy of the transmitted signal is approximately 850.

References

[1] Umberto Mengali. *Synchronization Techniques for Digital Receivers.* Springer Science & Business Media, 1997.

[2] Heinrich Meyr, Marc Moeneclaey, and Stefan A. Fechtel. *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing.* Vol. 444. Wiley Online Library, 1998.

[3] John Proakis and Masoud Salehi. *Digital Communications.* 5th ed. edição. Boston: Mc Graw Hill, Jan. 1, 2007. ISBN: 978-0-07-295716-7.