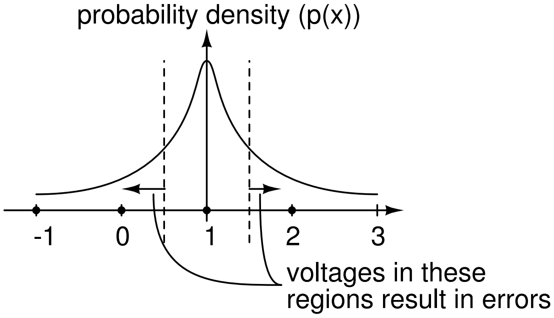
Solutions to Assignment 2

Question 1

(a) If a +1 V signal level is transmitted with Gaussian noise that has a noise voltage of $V_{\text{rms}} = 150$ mV then the mean of the signal is $\mu = 1$ V and the standard deviation is $\sigma = 0.150$ V.

For symmetric noise (e.g. Gaussian) and equally probable signal levels, the optimum decision thresholds will be half-way between the voltage levels. For the +1 V level the thresholds will be at 0.5 V and 1.5 V. An error will be made if the received signal (transmitted signal plus noise) is greater than 1.5 V *or* less than 0.5 V:


probability density (p(x))
-1
0
1
2
3
voltages in these regions result in errors

We can find the normalized thresholds for these two voltages as:

$$t_{1\text{V}} = \frac{v-\mu}{\sigma} = \frac{1.5-1}{0.15} \approx 3.33$$

and

$$t_{0.5\text{V}} = \frac{v-\mu}{\sigma} = \frac{0.5-1}{0.15} \approx -3.33$$

.

From the graph (or using a calculator) we can find the probability that the normalized signal is less than the upper threshold is $P(x > 1.5\text{ V}) = P(3.33) \approx 0.9996$ and so the probability that it's greater than the threshold is $P(x > 1.5\text{ V}) = 1 - P(x < 1.5\text{ V}) \approx 1 - 0.9996 = 4 \times 10^{-4}$.

Similarly, the probability that the signal plus noise is less than the lower normalized threshold is $P(x < 0.5\text{ V}) \approx P(-3.33) = 4 \times 10^{-4}$.

The probability of either of these two events is the sum of the probabilities, 8×10^{-4}.

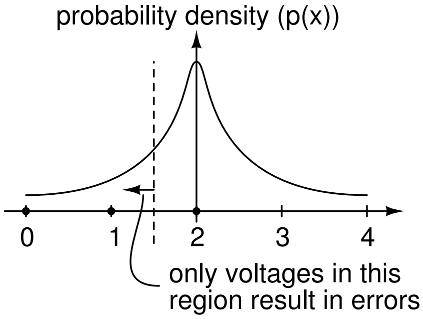
(b) When a +2 V signal level is transmitted the mean of the signal is now 2 V.

The only decision threshold will be at 1.5 V, half-way between the +2 V level and the +1 V level. An error will only be made if the voltage (signal plus noise) is less than 1.5 V.

As before ee can find the normalized thresholds for this voltage as:

$$t_{1\text{V}} = \frac{v-\mu}{\sigma} = \frac{1.5-2}{0.15} \approx -3.33$$

and the probability, as before, that the normalized signal is less than the lower threshold is $P(x < 1.5\text{ V}) = 4 \times 10^{-4}$.


probability density (p(x))
0
1
2
3
4
only voltages in this region result in errors

(c) The answers are different because when a 2V signal level is transmitted, the only threshold is the one below the expected signal level and no errors are caused by positive noise voltages.

In practice, the fact that the error rate for certain levels is $\frac{1}{2}$ that of others has a relatively small effect on the overall error rate compared to other effects and is often ignored.

Question 2

The following table shows the bits that would be transmitted (in binary notation), including the start and end flag sequences, for an HDLC frame with an address field of `0xff`, a control field of `0x01`, one byte of data with a value of `0xfe`, and a CRC of `0x20`, `0xB0`. The start and end flags and the underlined zero bits are added for HDLC framing.

Note that HDLC transmits bits LS-bit first (as with asynchronous serial interfaces).

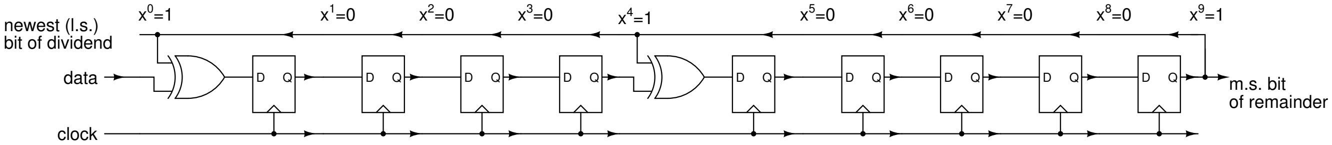
byte	hex	binary	transmitted
flag	`7E`	`0111 1110`	`0111 1110`
address	`FF`	`1111 1111`	`1111 10111`
control	`01`	`0000 0001`	`1000 0000`
data	`FE`	`1111 1110`	`0111 11011`
CRC	`20`	`0010 0000`	`0000 0100`
CRC	`B0`	`1011 0000`	`0000 1101`
flag	`7E`	`0111 1110`	`0111 1110`

Question 3

The schematic of a circuit that can divide a polynomial with coefficients from GF(2) by the generator polynomial:

$$G(x) = x^9 + x^4 + 1$$

is shown below:



The circuit consists of a sequence of flip-flops. The input to each flip-flop is either the output of the previous flip-flop or the output of the previous flip-flop `xor`'ed with to the output of the last flip-flop.

The bits of the dividend polynomial are input, one bit per clock, into the stage at the left and the quotient appears at the output of the stage at the right.

The flip-flop bits represent the partial remainder. The `xor` gates add* the generator polynomial to the partial remainder when the most-significant bit of the partial remainder is '1'.

On each clock edge the flip-flops load the new partial remainder.

A practical circuit requires additional circuitry to initialize the flip-flops, to feed in $n-k$ (9 in this case) bits to compute the CRC and to extract the CRC.

*Equivalent to subtracting in GF(2).