

Solutions to Assignment 1

Corrected solution to Question 4 to generate random values in range (-1,1) instead of (0,1): 2018/2/19.

Question 1

- (a) Table 24-1 in the datasheet gives the following equation for the UBRRn setting in Asynchronous Normal mode as a function of the baud rate (BAUD):

$$\text{UBRRn} = \frac{f_{osc}}{16 \cdot \text{BAUD}} - 1.$$

For an oscillator frequency of $f_{osc} = 16$ MHz and a baud rate of 40 kbps:

$$\text{UBRRn} = \frac{16 \times 10^6}{16 \cdot 40 \times 10^3} - 1 = 24.$$

- (b) Section 24.12.2 of the datasheet shows that Bit 4, “FE0,” of USART Control and Status Register 0 A can be read to determine if a framing error has been detected.
- (c) As explained in section 24.5, the start bit is output as low (0V), the stop bit is output as high (3.3V) and data values of 1 and 0 are high and low respectively (3.3 and 0V). The differences from the RS-232 signal voltages are thus that the micro-controller’s voltages:
- are inverted in polarity,
 - are unipolar,
 - use voltages levels 0 and 3.3V instead of more than ± 5 V.

Question 2

The total power, P , can be found by integrating over a large enough frequency range. The function is even, so we can do the computation over positive frequencies and double the result.

We can do the integral numerically by adding up small rectangles that are the height of the function and whose width is the spacing between samples. Since the power drops as $1/f^2$, integrating over a

range of ± 1000 should result in a small error. The following C program does this using a spacing of 0.001 over the range 0 to 1000 to find the total power over positive frequencies:

```
#include <math.h>
#include <stdio.h>
float s(double x) {
    return x ? sin(x)/x : 1.0 ;
}
void main(void) {
    double sum=0.0 ;
    for ( double x=0.0 ; x<=1000.0 ; x+= 0.001 )
        sum += pow(s(x),2) ;
    printf("total power is %f\n", sum*0.001 ) ;
}
```

which prints out: total power is 1.570796. We can then modify the program to find the frequency where the sum is 80% of this value:

```
...
double sum=0.0 ;
for ( double x=0.0 ; x<=1000.0 ; x+= 0.001 ) {
    sum += pow(s(x),2) ;
    if ( sum*0.001 > 1.2566 ) {
        printf("80%% bandwidth is %f\n", 2.0*x ) ;
        break ;
    }
}
...
```

which prints: 80% bandwidth is 3.358000

We can do this more easily in Matlab (Octave, actually):

```
x=[1e-9:0.001:1000] ;
p=(sin(x)./x).^2;
sum(p)*0.001
```

which prints out ans = 1.5708. Rather than using a loop, we can find the cumulative sums and find the first value of x where the sum exceeds 80% of this:

```
2*x(find(cumsum(p)*0.001>(0.4*pi))(1))
```

which prints `ans = 3.3580` which is the 80% power bandwidth.

We can also find a solution in terms of the sine integral (for example, using [Wolfram Alpha web site](#)) as:

$$\int_0^B \left(\frac{\sin(f)}{f} \right)^2 df = \text{Si}(2B) - \frac{\sin^2(B)}{B}$$

The definite integral over 0 to ∞ is $\pi/2$ so the 80% power bandwidth is the bandwidth over which the above integral equals 0.4π . This is at approximately $B = 1.68$ which is the “one-sided” bandwidth. The two-sided bandwidth is twice this, or approximately 3.36.

Question 3

A sine wave applied to the given function would produce a square wave output with values ± 1 .

The definition of THD is:

$$\text{THD} = \sqrt{\frac{P_1 + P_2 + P_3 + \dots}{P_0}}$$

where P_0 is the power of the fundamental and P_n are the powers of the harmonics.

For a square wave the frequency components have amplitudes $\frac{1}{n}$ for odd n (1,3,5,...). The power of each component is proportional to the square of the amplitude so the THD will be:

$$\text{THD} = \sqrt{\frac{(\frac{1}{3})^2 + (\frac{1}{5})^2 + (\frac{1}{7})^2 + \dots}{(\frac{1}{1})^2}}$$

Using the result:

$$\sum_{n=0}^{\infty} \frac{1}{(2n+1)^2} = \frac{\pi^2}{8}$$

the THD would be $\sqrt{\frac{\pi^2}{8} - 1} \approx 0.4834$ or about -3 dB which we can also obtain with the following Matlab (Octave) calculation:

```
> sqrt(sum(1./[3:2:99999].^2))
ans = 0.48342
```

Some definitions of THD include the harmonic power in the denominator in which case the ratio would be:

$$\text{THD} = \sqrt{\frac{\pi^2/8 - 1}{\pi^2/8}} \approx 0.435 = -3.6 \text{ dB}$$

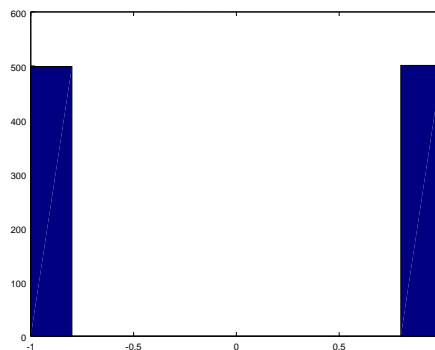
Question 4

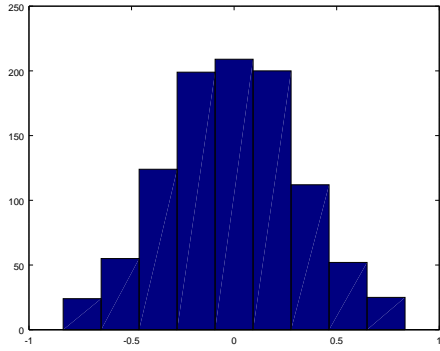
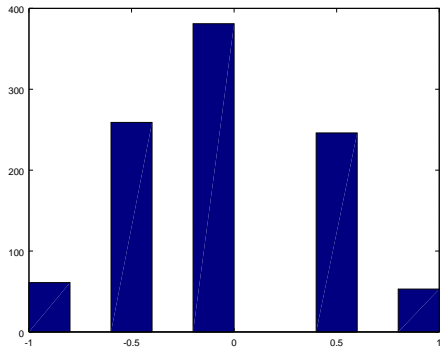
The Matlab commands:

```
for N=[1,4,12]
    x = sum(2*(rand(N,1000)>0.5)-1,1)/N ;
    hist(x,9) ;
    fn = sprintf("sol1-4-%02d.pdf",N) ;
    print(fn) ;
end
```

do the following:

- generates an N (rows) \times 1000 (columns) array of uniformly-distributed random numbers using the `rand` function
- converts them to uniformly distributed values in $(-1,1)$ by comparing to 0.5 (each value becomes 0 or 1), multiplying by 2 (each value becomes either 0 or 2) and subtracting 1 (each value becomes -1 or 1).
- adds up the N values in each column using the `sum` function with the dimension to sum along (1)
- computes the histogram of the row vector result using the `hist` function with 9 bins
- saves each figure to a file





The results demonstrate the Central Limit Theorem: the distribution of a sum of random variables tends towards a Normal distribution regardless of the distribution of the random variables being summed.