

Vibration and Tilt Measurements

1 Objectives

The objective of this laboratory exercise is to explore various applications of the accelerometer sensor. In particular, vibration measurements are performed and a frequency response function of a metallic rectangular plate is calculated. The same accelerometer is used in the tilt measurements allowing for comparison of the analog and digital data.

2 Theoretical background

2.1 Vibration measurements

A need for vibration analysis arises in many engineering applications. Two vibration parameters dominate practical considerations: the amplitude and frequency of vibration. The amplitude of vibrations is presented in terms of displacement, velocity, or acceleration, and the frequency is associated with the rate of change of these quantities. In a typical time-domain vibration record, the amplitude is obtained as a difference between the maximum and the minimum values of the signal, and the vibration frequency can be calculated from the period determined as a distance between two neighboring amplitude peaks. In reality, a vibration record most likely will feature not just one, but multiple frequencies and this simple analysis may not yield adequate results. Therefore, a detailed Fourier analysis of a waveform is needed. This analysis can be accomplished either analytically using a Fourier integration scheme, or numerically by employing the Fast Fourier transform (FFT). We will use the later approach.

To determine the natural frequencies of a structural system one needs to excite it mechanically and measure a response to the applied excitation. By exciting the system in a range of frequencies, it is possible to determine a frequency dependent response or *frequency response function* – one of the main dynamic characteristics of a structural system. To calculate a frequency response, it is necessary to know spectral density functions of the input S_{yy} and output S_{xx} signals and their cross-spectral densities S_{xy} , S_{yx} . A Fast Fourier Transform is used to obtain the spectral densities.

```
F1=fft(x);  
F2=fft(y);  
Sxy=F2.*conj(F1);  
Sxx=F1.*conj(F1);  
Syy=F2.*conj(F2);  
Syx=F1.*conj(F2);
```

Where conj means a complex conjugate of a function.

After the spectral densities are determined, calculation of the frequency response function is straightforward.

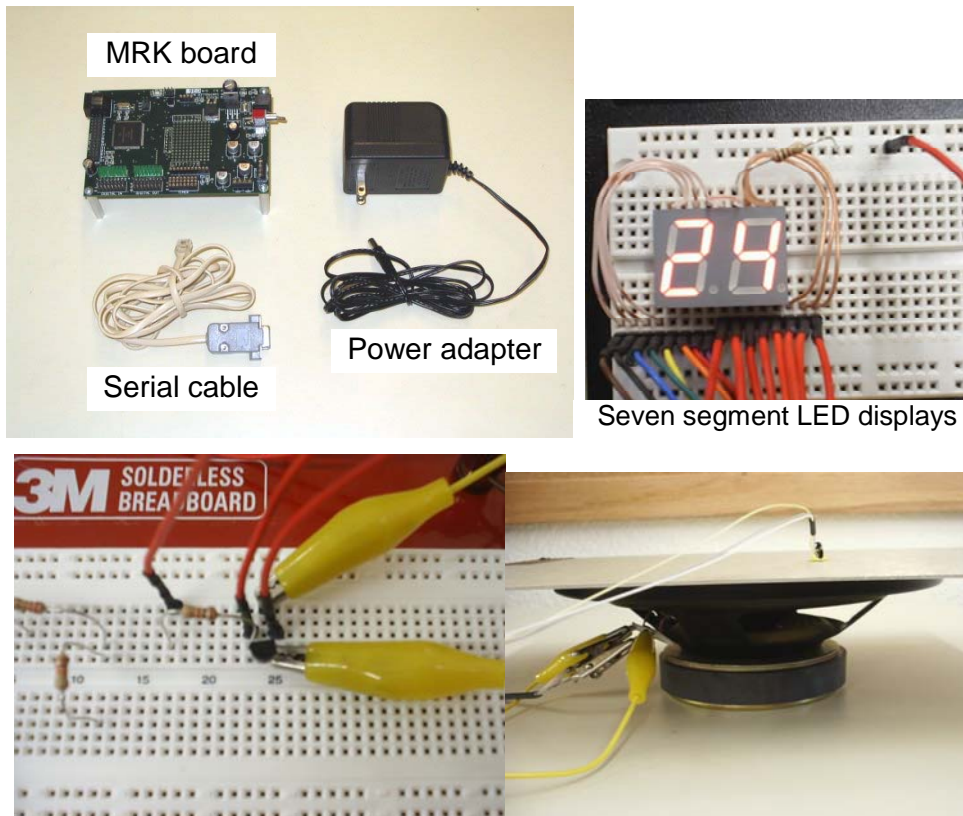
$$H(i\omega) = \frac{S_{yx}(\omega)}{S_{yy}(\omega)} = \frac{S_{xx}(\omega)}{S_{xy}(\omega)}, \quad (1)$$

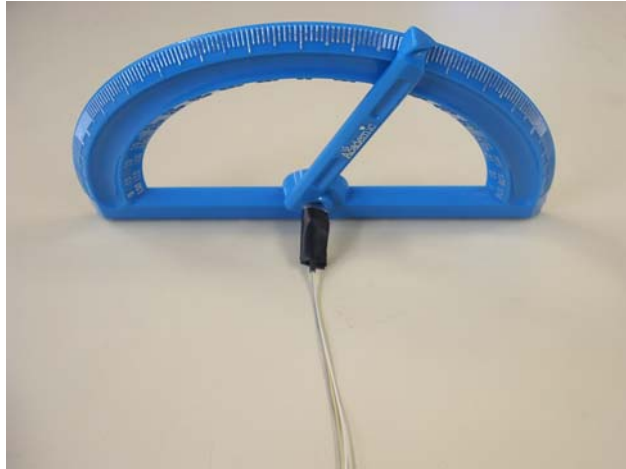
It should be noted that according to expression (1), two different paths can be followed to obtain $H(i\omega)$. This fact allows us to verify the adequacy of the measurement and calculation procedures. A coherence function can be defined to quantify the noise contamination of the signal.

$$Coh = \frac{|S_{xy}(\omega)|^2}{S_{yy}(\omega) \cdot S_{xx}(\omega)}, \quad (2)$$

For the signal containing a random noise, $Coh = 0$, the consistent measurements unaffected by noise will yield the Coh value close to unity, i.e. 1.

3 Equipment





- MRK Controller Board
- Serial communication cable
- Power adapter
- LED Display
- Protractor
- ADXL203 MEMS Accelerometer
- Transistor circuit
- Speaker

4 Procedures

4.1 Vibration measurements

Exercise #1: In this laboratory exercise you will be measuring a dynamic response of a rectangular plate to the applied impulse excitation. The excitation is realized using a speaker positioned underneath the metallic plate.

Power on the MRK controller board and enable the serial communication. Output of the audio amplifier is connected to the speaker. You will need to use an input signal to the speaker to both mechanically excite the metallic plate and to trigger the data acquisition. The triggering circuit involves a simple transistor (shown in the equipment section) and a timer block of the MRK controller board.

ADXL203 accelerometer is powered by the 5 V on board voltage, and utilizes one of the pins of the ANALOG_IN port for acquisition of the vibration data. The frequency bandwidth of the ADXL203 is from 0.5 to 2500 Hz.

To provide an excitation signal which will be exerted on a plate by a speaker, we will use the Pulse excitation option of the Realtime Analyzer software. In the software menu select the Pulse tab, select the shortest impulse length and maximum loudness. Turn the computer speaker volume control knob to 50% volume. If needed, de-select the Repeat pulse block. When you click on Start, you should hear a short audio pulse.

Begin a new Eclipse project; choose to store the program in EEPROM (starting address 0x4000). You must change the memory.x file to store your program in EEPROM and the data in SRAM. The example of the memory.x file is given below.

```

MEMORY {
page0 (rwx) : ORIGIN = 0x0, LENGTH = 256
text  (rx)  : ORIGIN = 0x4000, LENGTH = 0x1000
data   : ORIGIN = 0x1000, LENGTH = 0x3000
}
PROVIDE (_stack = 0x3C00);

```

The modifications to the memory.x file should be introduced before compiling your C program. Please consider that in Eclipse unsaved files have a * after the name. To run your program, you will have to type GO 4000 instead of GO 1000.

The C code that enables the synchronized data acquisition is presented below

```

#include "MRK.h"
#define ARRAY_SIZE 4096

int data[ARRAY_SIZE], i, rate, get_data;
volatile int trig;

void do_a2d(void) {
    TIME1 = TIMER + rate;
    get_data = 1;
}

void trigger(void) {
    trig = 1;
}

int main() {
    trig = 0;
    analog_in(RESOLUTION, 10);
    setup_timer(ENABLE, 0);
    setup_timer(INPUT_CAPTURE, 0, FALLING, &trigger);
    setup_timer(OUTPUT_COMPARE, 1, NOTHING, &do_a2d);

    /*===== sample rate =====*/
    // rate = 24000; // 1kHz
    // rate = 2400; // 10kHz
    // rate = 1200; // 20kHz
    // rate = 960; // 25kHz

    fprintf(SCI0, "Waiting for trigger...\n\r");
    while (trig == 0); // wait for trigger

    i = 0; get_data = 1;
    TIME1 = TIMER + rate;

```

```

while (i < ARRAY_SIZE) {
    if (get_data) {
        data[i] = analog_in(IN,3); // channel 3
        i++;
        get_data = 0;
    }
}
fprintf(SCI0, "\nData stored. To receive data, go to\n\r");
fprintf(SCI0, "'Transfers->Receive Text File...'\n\r");
fprintf(SCI0, "When ready, press any key. When done,
hit\n\r");
fprintf(SCI0, "stop button and then press any key.\n\r");
fgetchar(SCI0);
for (i=0;i<ARRAY_SIZE;i++) {
    fprintf(SCI0, "%u\n", data[i]);
}
fgetchar(SCI0);
}

```

Make sure that the components of the circuitry are connected correctly. Ask a TA to verify your setup. Download a *.s19 file into the microcontroller and run your program. When you press the Start button of the software signal generator, the signal is sent to an audio amplifier and eventually reaches a speaker. At this point the electrical signal is converted to an acoustic pressure pulse that travels through air and strikes a surface of a metallic plate setting up vibrations. You will have to put an accelerometer on a plate to measure these vibrations. At the same time, an electric pulse is sent to a TIMER block through the amplifying transistor circuit. This pulse tells the microcontroller when to start the data acquisition process.

To determine a frequency response of the metallic plate, it is necessary to use both input and output signals.

You will need to measure the excitation (input) signal first. This can be achieved by placing an accelerometer on top of the speaker diaphragm. Make sure that an accelerometer sits firmly on a surface of the diaphragm. Run your program. Note that the sample rate is set to 20kHz – you will need to know the sample rate to calculate a frequency response. Store your data in the IN.txt file.

Place the accelerometer on a surface of the metallic plate to measure a vibration response (output) signal. It is up to you to select a location. Repeat previous experiment. Store your data in the OUT1.txt file.

Place the accelerometer at the center of a plate and repeat the experiment. Store your data in the OUT2.txt file.

1. Load your data in Matlab. You will have 3 vectors: IN, OUT1, and OUT2 respectively. Plot the time-domain representation of these signals.
2. Calculate and plot the spectrums of these signals.

3. Calculate spectral densities according to an example given in the theoretical section.
4. Using expression (1) calculate the frequency response functions (FRF). Plot your FRFs.
5. Using expression (2) calculate the coherence function. What are the maximum and minimum values of your Coherence function? What do they tell you about consistency of your measurements?

4.2 Accelerometer as a tilt sensor

MEMS accelerometers can be used to determine an angle of tilt. This measurement is possible because the sensing axis of an accelerometer resting on a tilted surface is not aligned with the vector of gravity, thus generating a voltage output nonlinearly proportional to the angular difference between these two vectors. In this laboratory exercise we will only measure the tilt with respect to the y-axis. The Analog Devices ADXL203 accelerometer has a sensitivity of 1000 mV/g, which is sufficient for tilt applications.

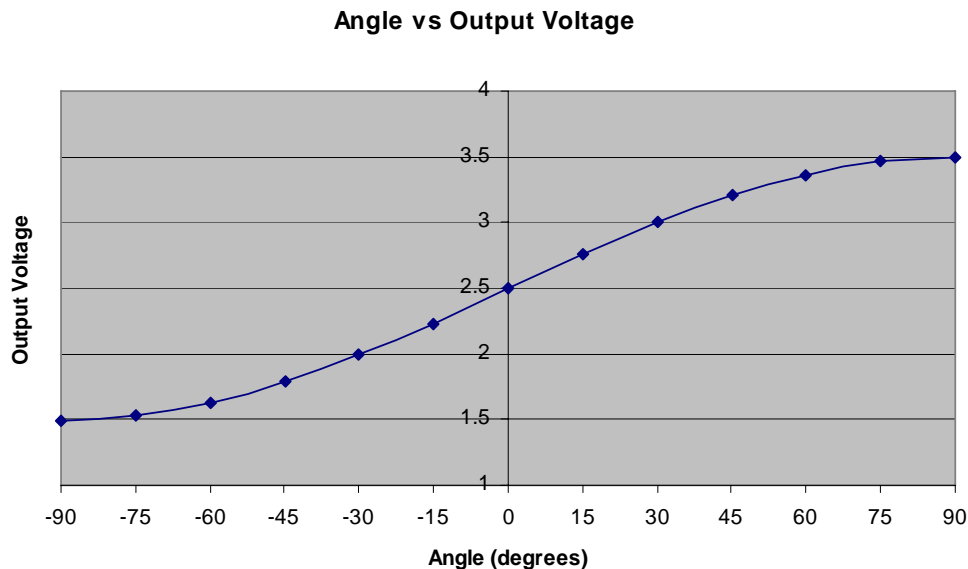


Figure 1 Nonlinearity of the accelerometer output in the tilt application.

Nonlinearity of the accelerometer:

As it can be seen in the figure, the typical output of a MEMS ADXL203 accelerometer resembles a sine function. The figure shows the analog output voltage of the accelerometer covering the tilt angles from -90° to $+90^\circ$. The change of the tilt directly corresponds to a change in the acceleration due to gravity acting on the accelerometer. The slope of the curve is the sensitivity of the device.

Exercise #2:

Before this exercise ask a TA to remove the accelerometer from the plate and connect it to the protractor. After the connection is completed, adjust the protractor knob to each angle and use a

voltmeter to measure the y-axis output voltage for each angle. Record the measured values in Table 1 and plot the nonlinear output characteristic pertaining to your measurements.

Table 1 Measurement of accelerometer tilt response.

Angle	-90°	-75°	-60°	-45°	-30°	-15°	0°	15°	30°	45°	60°	75°	90°
Output Voltage													

Exercise #3:

The program below is used to display the angle value which you measured. You will have to include your own digital values for the desired angles, because the digital values change from one accelerometer to another. Complete your project in Eclipse and load the program to the microcontroller. Adjusting the protractor knob to each of the desired angles, observe and record the digital value on the computer screen (ignore the angle value shown on the screen while recording). In your program, replace each digital value in the table with the new value you have recorded.

```
#include "MRK.h"

#define TABLE_SIZE 13

int x_table[TABLE_SIZE] = {258, 267, 291, 332, 387, 446, 515, 580, 642, 694,
735, 761, 769}; // digital values
int y_table[TABLE_SIZE] = {-90, -75, -60, -45, -30, -15, 0, 15, 30, 45,
60, 75, 90}; // angle values

/* ----- *
 * Function that interpolates between table values. *
 * ----- */
int interpolate(int x) {
    int y,n;
    // find position in table
    for (n=0; (x_table[n] < x) && (n < TABLE_SIZE); n++);
    // interpolate
    y = (y_table[n]-y_table[n-1])*(x-x_table[n-1])/(x_table[n]-x_table[n-1])+y_table[n-1];
    return y;
}

int main() {
    digital_in(OUTPUT,ALL);
    digital_out(OUTPUT,ALL);
    analog_in(RESOLUTION, 10);
    const char segments[] = {0x3f, 0x06, 0x5b, 0x4f, 0x66,
                             0x6d, 0x7d, 0x07, 0x7f, 0x67};

    int i, a, input, angle;
    i = 0;
```

```

while (1) {
    // average 8 values
    input = 0;
    for (i = 0; i < 8; i++) {
        input = input + analog_in(IN,2);
    }
    input = input / 8;
    // interpolate angle from table
    if (input <= x_table[0]) angle = -90;
    else if (input >= x_table[TABLE_SIZE-1]) angle = 90;
    else angle = interpolate(input);
    fprintf(SCI0,"Digital %R4d = %R3d degrees\r",input,angle);
    a = abs(angle);
    if (angle < 0) i = 0x80; // use DP to indicate negative
    else i = 0x00;
    if (a > 99) { // too big, can't display
        digital_in(OUT,ALL,0x40);
        digital_out(OUT,ALL,0x40);
    }
    else {
        digital_in(OUT,ALL,segments[a/10]);
        digital_out(OUT,ALL,segments[a%10] | i);
    }
    delay(1000);
}
}

```

Re-compile and download the program with the new digital values. Move the protractor knob throughout its range, noting the accuracy. You will observe that as the device is tilted from 0° to 90° or -90° the display will become less accurate and bounce more as you approach -90° and 90° . Explain what can cause this situation. Which improvements would you suggest to mitigate this problem?

5 Report and analysis requirements

5.1 Theory

What are the frequency response and coherence functions? What do they represent?

Where and for what purposes we may use them?

Draw and discuss schematics of the tilt experiment. Use three angles (0, 30, and 90 degrees) to show the alignment of the accelerometer's axis and the gravity vector.

5.2 Results and analysis

Answer all questions in previous sections of these laboratory instructions.

Present and discuss you results.

Why do you need a triggering circuit to start the acquisition of the vibration data?

How did you select the sample rate for the vibration measurements?

Determine first three resonance frequencies of the rectangular plate.

Did you obtain the same frequency values for the accelerometer placed in the center of a plate? Why?

What was the biggest problem you encountered during the tilt measurements? How would you mitigate it?

Give at least one example where the accelerometer can not be used for measuring a tilt angle.

*Last updated
October 26, 2006.*