4 Measurement Data Analysis (Experiment FMT)

Contents

4.1 Hints for writing the report

A report must be prepared to pass the experiment. The experiment instructions contain some questions, each of which must be answered and documented as indicated. For this purpose, create a report in PDF format and upload it to StudOn in your respective group. Be sure to observe the submission deadlines, the time stamp of the StudOn server applies.

4.2 Notes on working on the laboratory course

During the laboratory course MATLAB, you are to familiarise yourself with the basics using the technical programming language MATLAB. The tasks are set in such a way that help for self-help is taught in particular. This means that you should independently search for solutions to certain problems and, in particular, independently consult the help functions of the programming environment used.
Task 1: Research a function to calculate the magnitude of a complex number and a vector!
Task 2: Research a function to save a plot as a (compressed) image file.
Task 3: Create a command (1 line) which creates a variable with 1000 random vectors (3D, so each with x, y and z component). The random numbers should be normally distributed.

4.3 Signal processing with the aid of the Discrete Fourier Transform (DFT) or Fast Fourier Transform (FFT)

The Discrete Fourier Transform (DFT) is a transformation from the field of Fourier analysis. It maps a discrete-time finite signal, which is continued periodically, onto a discrete, periodic frequency spectrum, which is also called the image or frequency domain. The DFT is very important in digital signal processing for signal analysis. The DFT is used in signal processing for many tasks, such as:

- To determine the frequencies occurring in a sampled signal.
- To determine the individual amplitudes and phases at these frequencies.

The basis of the Fourier transform is the fact that every periodic signal can be represented by a linear combination of sine and cosine waves (see lecture notes). For certain lengths of the data sets (length of the time vector is a power of 2 or a decomposition results in small prime factors), acceleration techniques exist that are summarised under the term FFT (fast Fourier transform). This is also where the Matlab function name for the DFT is derived from; it is `fft()` and automatically applies runtime optimisations if possible.
The starting point for the following tasks is a vibration measurement on the crane boom. A signal analysis is to be carried out to determine the natural frequencies of the occurring vibrations.

Task 1

a) First, you will learn some important properties of the Fourier transform.

   (1) Open the script `DemoFFTgrenzFrequenz.m` and reproduce it. It includes the definition of a sampling rate (samplingFrequency) and creates (discrete) cosine signals that are

sampled at the defined rate. The result of the samples is then output in a plot. Run the script and observe how the signal frequency influences the curves shown as a function of the sampling rate. The higher the frequency of the signals becomes, the more signal periods occur in a given time and the fewer sampling points are available per signal period.

(2) In a Fourier transformation, signals are transformed from the known time domain (amplitude over time) into the frequency domain (complex amplitude over frequency). The sampling rate at which the signals were "observed" plays an important role here. Set the switch "runFFT" to "true" in the script and observe the calculated frequency of the sampled signals. You will notice that there are 2 data points, i.e. apparently 2 frequencies in the sampled signals.

(3) Change the signal frequencies and observe the Fourier transformed signals. What is the difference between 2 signals with frequencies $f_1 = k$ and $f_2 = f_{\text{abtast}} - k$? What conditions must the examined frequencies meet in comparison to the sampling frequency? Compare the observed correlations with the Nyquist-Shannon sampling theorem $f_{\text{abtast}} > 2f_{\text{max}}$.

b) The results of a vibration measurement are stored in the file `Daten_Schwingungssensor.xlsx` gespeichert.

(1) Create a new script and save it with a meaningful name. In Windows Explorer, navigate to the location of the .xlsx file and place the complete file name incl. path in the form drive letter:\folder1\folder2\...\filename.extension
in a new variable in the script. In the following, also use the option of executing only parts of a script by selecting it and pressing F9 to process the subtasks.

(2) Open the file with Excel and familiarise yourself with the structure of the data at hand.

(3) import the measurement data without any incorrect data that may be stored in the table. Use the function `num = readtable(...)` for this. If necessary, refer to the documentation.

(4) Assign the variable `xValues` the x-values and the variables `yValues` the y-values of the measurement series.

(5) visualise the signal in a suitable way with a formatted plot (incl. title, axis labels). Use the colour cyan for the representation of the signal curve. Can you see which frequencies the signal is composed of?
Task 4: Described plot.

(6) Check the x values for equidistance between adjacent values. Ensure that the imported data has this property so that an FFT can be applied (this usually requires a constant sampling frequency over the entire data series). Note:Calculate the derivative of the vector (e.g. function `gradient()`) and determine the unique values (function `unique()`) in the resulting vector). Why does the function `unique()` determine multiple values even though the vector was created with the colon notation (`a:b:c`) and thus the distance between two neighbouring values should be constant? Determine the difference between any two neighbouring determined "`unique` values" and consider what this specifically says.

Task 5: a) number and mean value of the unique values in the x vector. b) Convert the x values into a vector with simple floating point accuracy and repeat the procedure from a). c) Suggest a solution for calculating a correct (unique) frequency value from the time values of the measurement series? (Hint 1:) The question has something to do with how (floating point) numbers are represented in the computer.) (Note 2: If you cannot answer the question, use the value $\frac{1}{x(2)-x(1)}$) for the sampling frequency for the following calculations.

c) Let's assume that the vibration sensors used are technically outdated and a new acquisition of high-quality components is out of the question for financial reasons. Therefore, a software solution for processing the signal is to be developed. In the following, it will be determined how it is possible to determine the natural frequencies even of very noisy data with the help of DFT/FFT. The aim is to determine the frequencies with which the crane vibrates from a bad signal.

(1) Perform the Fourier transformation of the imported signal from b). (see also example script `DemoFFTgrenzFrequenz.m`)
Divide the transformed signal for normalisation by half the number of data points and calculate the amounts from the complex results of the FFT.

(2) Visualise the result. Remember that you have to scale the frequency vector (x-axis) accordingly.

$$\texttt{f = (0: length(yData)-1)} \cdot \frac{\texttt{samplingFrequency}}{\texttt{length(yData)}}$$

Task 6: Create a plot that shows the discrete data points in red and also connects the data points with a blue line.

(3) Determine a command with which you can determine the indices of the dominant frequencies from the transformed signal by logical indexing.

(4) The superordinate goal of sub-task c) is to filter out the interfering frequency components from the signal, which are caused by noise, for example. To do this, create a new complex(!) frequency vector which, apart from zeros, consists only of the complex(!) frequency components determined in the previous sub-task. This ensures that the original signal can be reconstructed without the existing interfering signal components when transforming back into the time domain.

(5) Transform the cleaned complex(!) frequency vector back into the time domain. To do this, use the function `ifft()`.

(6) Visualise the result and compare it with the imported output signal in a plot. Use the previously imported time vector.
Task 7: Described plot with complete labelling and legend of the number series shown.

4.4 Solving differential equations with the Symbolic Math Toolbox using the example of a bending beam

The boom of the crane from chapter 3 is to be considered simplified as a clamped bending beam (Fig. 4.1). The basic equation for determining the bending line of a bending beam is known from engineering mechanics. It reads for small deformations (Fig. 4.1):

$$w'' = \frac{M_y(x)}{EI_{yy}(x)}$$

Here $M_y(x)$ is the bending moment around y, E is the modulus of elasticity and $I_{yy}(x)$ is the surface moment of inertia when bending around y. Using this simple differential equation as an example, symbolic calculations will be carried out below.

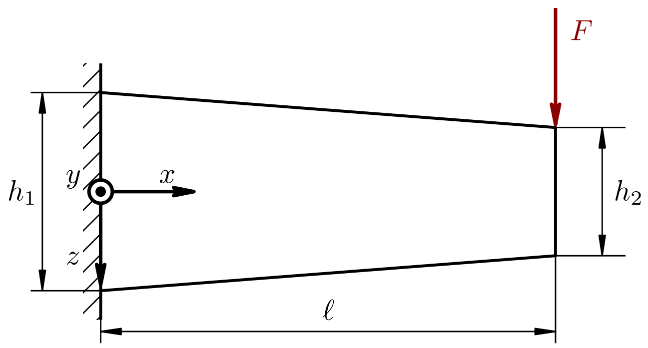

Figure 4.1: Bending beam schematic

Task 2

a) The first objective is to determine the analytical solution $w(x)$ using symbolic expressions for a bending beam with a given variable cross-section. While the bending line is comparatively easy to determine for a beam with a constant cross-section, an analytical solution is more difficult to determine in the case of a different cross-section. The surface moment of inertia of the beam for a moment about the y-axis is:
Moment of inertia per unit area $I_{yy} = \frac{b \cdot h^3}{12} \cdot \left(2 - \frac{x}{1000}\right)$ with:
Height of the bar h and width of the bar b
The bending beam is firmly clamped at one end and loaded at the other end with a force F perpendicular to the beam. This results in the bending moment $M_y(x) = F \cdot (l - x)$. By evaluating the boundary conditions (beam firmly clamped on one side), the relationships $w'(x = 0) = 0$ and $w(x = 0) = 0$ are obtained. The dead weight of the beam is neglected..

(1) Create the required symbolic variables `syms`, which are necessary to describe the problem.

(2) Determine $w'(x)$ by integration `int()` and calculate the constant of integration C1 with `solve()`.

(3) Replace C1 with your determined expression `subs()`.

(4) Repeat the procedure and determine $w(x)$.

b) Calculate the solutions of the bending line at the positions of the position vector x, with the following given quantities:
Length of the beam, location vector `x = 0:1000` with $[x]$ =mm.
Force $F = 1000$ N
Beam height $h = 25$ mm
Beam width $b = 15$ mm
E-Modul $E = 210$ GPa (Hint: 1 Pa = 1 N/m^2)
First create a structure `struct` in which the given quantities are stored. Pay attention to the correct units!
Note:

- Create another symbolic variable from $w(x)$ that depends only on x.
- From the printout received, create a *function handle* (Matlab: *matlabFunction*()). This allows efficient evaluation of the printout at many locations x.
- From here on, please continue to calculate the real part of the solution for all solutions and neglect the (very small) imaginary part.

c) For the following tasks, reshape $w(x)$ so that the function only depends on x, F, and h by replacing E and b with the given values. Save the new symbolic variable under a meaningful name (from here on called $w\#(x)$).

d) Now an uncertainty analysis according to GUM (Guide to the Expression of Uncertainty in Measurement) is to be carried out on the bending model. For this purpose, the input variables are assumed to be uncertain, with which the uncertainty of the output variable can be estimated. Real measurements of the input variables (e.g. force, displacement) are always subject to measurement uncertainty, so that the result of a subsequent calculation is also uncertain. The following shall apply:
Standard uncertainty of the force F: $u_F = 50$ N
Standard uncertainty of the beam height h: $u_h = 1,5$ mm
The force F and the beam height h represent independent measured quantities, so that the combined uncertainty u_c for b) can be determined in a simplified way according to the Gaussian error propagation law.

$$u_c(x) = \sqrt{\left(\frac{\partial w\#(x)}{\partial F}\right)^2 \cdot u_F^2 + \left(\frac{\partial w\#(x)}{\partial h}\right)^2 \cdot u_h^2}$$

(1) Using the symbolic derivative function `diff()`, calculate the symbolic expression $u_c(x)$.

(2) Calculate the course of $u_c(x)$ with the given quantities.

(3) Create a formatted plot (title, axis labels) showing for b) the combined uncertainty $u_c(x)$ together with the bending curve $w(x)$. Remember that the uncertainty itself is unsigned but affects the result in both positive and negative directions.
Task 8: Inscribed plot with full caption and legend.

e) Finally, the expected uncertainty is to be estimated additionally with a Monte Carlo simulation. In this procedure, random numbers are used to simulate the occurring uncertainty. In simple terms, many possible combinations of input variables are determined until a statistically determinable and reproducible scatter of the results of the output variable is available. It is assumed that the uncertain input variables F and h are normally distributed (Gaussian curve).

   (1) Research the required function around two arrays of normally distributed random numbers of length 1000 for representing the uncertain values of F and h.

   (2) Convert the symbolic expression $w\#(x)$ - as described above - into a function.

   (3) Now calculate the bending of the beam at the point of maximum deflection with the created uncertain input vectors F and h. To do this, call the Matlab function you just created with the value $x = 1000$ and the created vectors for F and h and save the result vector in a new variable. Remember to ignore the imaginary parts of the solution.

   (4) Prepare a plot comprising 6 individual plots (2 vertical, 3 horizontal, function *subplot*()). Visualise the input vectors F and h and the result with a histogram plot with `histogram()` in the top row of the prepared plot. What stands out? Why is the distribution of w slightly skewed?

   (5) Repeat the procedure just carried out with a length of the input vectors for F and h of 10^6 and visualise the results in the 2nd row of the prepared plot. What is the effect of increasing the number of random numbers?
   Task 9: Plot with the 6 sub-plots described).