Digital Communications Project 1
Introduction to MATLAB

1 Preface

The purpose of this project is to familiarize you with the basic commands in MATLAB for the signal and system representation.Since MATLAB works with data stored as vectors and matrices, the aim of this project is also to learn how to program in MATLAB environment and how to use MATLAB scripts and functions with respect to this. The communications lab is closely-supervised hands-on course, therefor you are strongly encouraged to experiment with the different MATLAB features.

Help for each MATLAB function can be displayed by entering `help <function name>`. For example `help plot` describes the `plot` function. The complete online manual can be started by entering `helpdesk` at the MATLAB command line.

2 Exercises

2.1 Indispensible Important Fundamentals

1. Generate the vectors `x1=[1 2 3 4]` and `x2=[4 3 2 1];` . What is the purpose of the semicolon at the end of the line?

2. Multiply these vectors element by element (`.*`). Form the scalar (inner or dot) product $\mathbf{x}_1\,\mathbf{x}_2{}^T$. The special character (`'`) denotes the transpose of the matrix. For the scalar product MATLAB uses the operator (`*`). What is the difference between the operators (`.*`) and (`*`), (`./`) and (`/`), respectively? Now, calculate the sum and the difference of the vectors `x1` and `x2`.

3. Generate the sequences: `x=0:1:15;` and `y=0:15;`. Is there a difference between those vectors? Now use the double increment and display the first element of the vector `x` (`x(1)`). How are the vectors in MATLAB subscribed?

4. Enter the following expression: `A=[2 -1 5; 6 8 -3];`. What did you generate? Show the dimension of the matrix `A`. Use the function `size(A)`. What is the third element of the second row: `A(?,?)`. Generate two row vectors, the first formed from the first column of the matrix `A` and second formed from the second row of `A`.

5. Now form an arbitrary 3-by-4 matrix `B` and compute the matrix product `C=A*B`. Use MATLAB function to generate a vector that contains sums of the elements in columns of the matrix `C`.

6. Create the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & \dots & 10 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \tag{1}$$

(use MATLAB functions, e.g. `diag`, `ones`).
Compute the column–sum norm (use Maltlab functions, e.g. `max, abs, sum`)

$$||\mathbf{A}|| := \max_{j=1\dots n} \sum_{i=1}^{n} |a_{ij}|,\ \mathbf{A} = [a_{ij}] \tag{2}$$

and normalize the matrix $\mathbf{A}$ to this value (compute $\mathbf{A}/||\mathbf{A}||$).

2.2 For-Loops and Functions in MATLAB

The multiplication of matrices $\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$ is defined by

$$c_{ij} = \sum_{k=1}^{Q} a_{ik} b_{kj},\ i = 1 \dots N,\ j = 1 \dots M, \tag{3}$$

where $\mathbf{C} = [c_{ij}]$ is a N-by-M matrix, $\mathbf{A} = [a_{ik}]$ is a N-by-Q matrix and $\mathbf{B} = [b_{kj}]$ is a Q-by-M matrix.

Write a MATLAB function “`matmul`” to compute this algorithm. The function should also verify the correct matrix dimensions of the input matrices $\mathbf{A}$ and $\mathbf{B}$ (use MATLAB statements `for`, `if`, `disp`, `error`).

The name of the file that contains a MATLAB function consists of the function name and extension “.m”. (For this example `matmul.m`). The structure of `matmul.m` is as follows:

```
function[C]=matmul(A,B);
% online help for your function
%
... your operations
C = ... result assignment
```

2.3 Advanced Vector and Matrix Creation and Access

1. Write a MATLAB function “`three`” to calculate the function

$$f(t) = \begin{cases} \dfrac{t + t_0}{t_0} & t \in [-t_0, 0) \\ \dfrac{t_0 - t}{t_0} & t \in [0, t_0] \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where the input arguments t and t_0 and the output argument f are scalar parameters.

```
function[f]=three(t,t_0)
% online help for your function
... your operations
f = ... result assignment
```

2. Modify the program from (a) so that the input t and output $f(t)$ are now vectors (t_0 is still a scalar). Do not use `for`–loops (use for example: `find`, `ones`, `zeros`).

Remarks:

MATLAB is optimized to work with vectors and matrices so try to avoid for loop and to use vector operations. For example `a = 1:10;` is extremely faster than

```
a=[];
for i=1:10
   a = [a i];
end
```

2.4 Matlab-Function Exercises and Plotting

The Fourier series of the 2π periodically repeated signal $y(t) = t$ for $-\pi < t < \pi$ is given by

$$\tilde{y}(t, N) = 2 \cdot \sum_{k=1}^{N} (-1)^{k-1} \frac{\sin(kt)}{k}. \tag{5}$$

Keep in mind that $y(t)$ is just $y(t) = t$ for $-\pi < t < \pi$. This part of the signal is then periodically repeated with period 2π and then called $y_p(t)$.

1. Write a MATLAB–function "`series`" that computes $\tilde{y}(t, N)$, where N is the number of Fourier-coefficients. Write a function "`original`" that computes the original signal $y_p(t)$ for all t. In both cases the input argument t must be given as a vector. This function should not use `for`–loops (use `mod`).

2. Plot the function $y(t)$ in the interval of $[-2.5\pi, 2.5\pi]$ using 500 points (`help plot`, `plot`), label your graphics with `title('...')`, `xlabel('...')` and `ylabel('...')` then print a copy of the graphics (`print` button in the figure). Plot $\tilde{y}(t, N)$, $N = 3, 7, 20$ in the same window, use different colors (`hold`, `hold off`). Enlarge the interesting regions of the graphic (use `zoom`).

Remarks:

- The default notation in MATLAB is matrix notation. Basically, it is not possible to represent the continuous-time signal in MATLAB. However, an approximation can be achieved through the use of sufficient number of samples.

- MATLAB statement `figure(n)` with `n=1,2...` makes `n` the current figure or creates a new figure with handle `n` if figure `n` does not exist.

- The statement `subplot(mnp)` divides the Figure window into an `m`-by-`n` matrix of small axes and uses the `p`-th axes for the current plot.

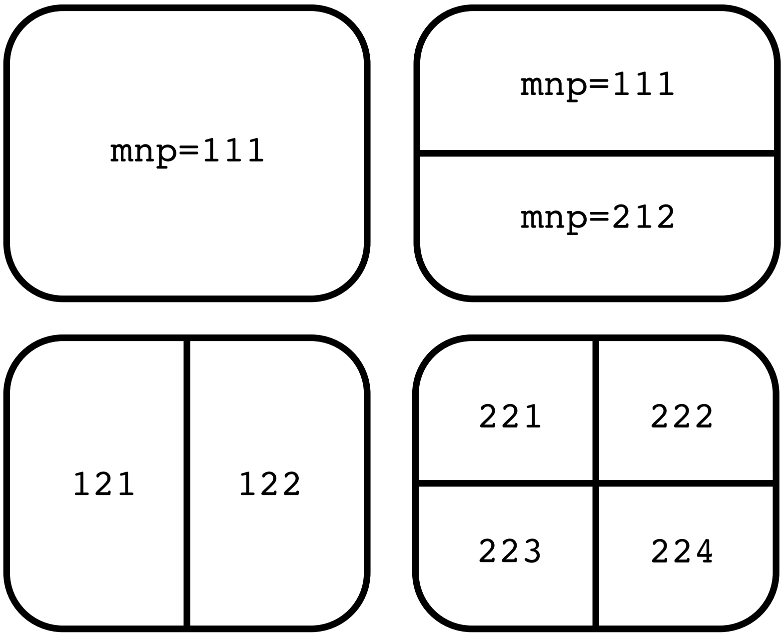

Fig. 1: Subdivision of a graphic window with `subplot(mnp)`.

- The command `zoom` allows an enlargement of interesting parts of the graphic. Clicking on the left mouse button and dragging defines the area of enlargement. A single click on the left/right mouse button enlarges/downsizes the graphic window by a factor of two.