**EEE 264 ADVANCED TOPICS IN WIRELESS COMMUNICATIONS FALL 2023**

Instructor : Preetham B. Kumar

Website : <https://www.csus.edu/faculty/k/preetham.kumar/>

Class Hours : Tues. 1.30 – 2.45 p.m. Classroom TBA

 Thur. 1.30 – 2.45 p.m. via Zoom:  <https://csus.zoom.us/j/83829879263>

Office hours: Mon/Wed: 1 -2 pm

 Tues: noon- 1 pm

 <https://csus.zoom.us/j/867091331>

Telephone : 916-278-7949

E-mail : kumarp@ecs.csus.edu

Prescribed Text: **Instructor notes (available on website)**

Reference Texts: **‘Wireless Communications: Principles and Practice’ by Theodore S. Rappaport, Prentice Hall, 2nd Edition, 2001.**

**MATLAB/SIMULINK Guide, The Mathworks.**

 <http://www.mathworks.com/>

Course Grading

#  Midterm I : 20%

 Midterm II: 20 %

Final : 30 % (comprehensive)

 Project: 20%

 Homework: 10%

**2 double-sided pages** of notes are allowed for midterms

**4 double-sided pages** of notes are allowed for final.

**Top Hat**

Students are required to obtain a Top Hat account from the online link or our university bookstore We will be using the Top Hat ([www.tophat.com](http://www.tophat.com/)) classroom response system in class. You will be able to submit answers to in-class questions using Apple or Android smartphones and tablets, laptops, or through text message.

You can visit the Top Hat Overview (<https://success.tophat.com/s/article/Student-Top-Hat-Overview-and-Getting-Started-Guide>) within the Top Hat Success Center which outlines how you will register for a Top Hat account, as well as providing a brief overview to get you up and running on the system.

An email invitation will be sent to you by email, but if don’t receive this email, you can register by simply visiting our course website: <https://app.tophat.com/e/579134>

Note: Course Join Code is **579134**

**Class Schedule for Fall 2023**

 WEEK BEGINNING TOPICS NOTES FILE \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 1 08/29/23 Overview of modern EEE264\_1

 Wireless Communications 2 09/05/23 Capacity increase techniques EEE264\_2

 Handoff Protocols

3,4 09/12/23 Practical Radio Propagation EEE264\_3

 Models

 **5 09/26/23 Theory Review**

 **09/28/23 Problems Review**

 **6 10/03/23 Midterm 1**

 2.5 G Wireless Networks EEE264\_4

 and architecture

 7,8 10/10/23 3G Wireless Networks EEE264\_5

 **9 10/24/23 Theory Review**

 **10/26/23 Problems Review**

 **10 10/31/23 Midterm 2**

 Higher Generation EEE264\_6

 wireless networks

 11,12 11/07/23 Short range wireless networks EEE264\_7

 13,14 11/21/23 Wireless Measurement techniques EEE264\_8

 15 12/05/23 Review

 **16 12/14/23 Final exam: 12.45-2.45 pm**

**MATLAB COMMANDS AND TOOLBOXES**

**System operating commands**

PC based MATLAB can be opened by clicking on the MATLAB icon. The MATLAB prompt is >>, which indicates that commands can be started, either line by line, or by running a stored program. A complete program, consisting of a set of commands, can be stored in a MATLAB file for repeated use as follows:

(a) Open a file in any text editor ( either in MATLAB or Word/Notepad/Wordpad), and write the program.

(b) After writing the program, save it as a *filename.m* file.

(c) To run the program, click on the Run arrow, or type the filename after the prompt:

>> filename

The program will run, and the results/error messages, if any, will be displayed on the screen. Plots will appear on a new screen.

**I. NUMBERS**

 **Generation of numbers**

Example: Generate the real numbers z1 = 3, z2 = 4.

 >> z1 = 3

 >> z2 = 4

 Example: Generate the complex numbers z1 = 3+j4, z2 = 4+j 5

 >> z1 = 3+4j

 >> z2 = 4+5j

Note: The symbol i can be used instead of j to represent √-1.

Example: Find the magnitude and phase of the complex number 3+j\*4

>> z = 3+j\*4

>> zm = abs(z) ; gives the magnitude of z

>> zp = angle(z) ; gives the phase of z in radians

**Addition or Subtraction of Numbers (real or complex)**

>> z = z1 + z2 ; addition

>> z = z1 - z2 ; subtraction

**Multiplication or Division of Numbers (real or complex)**

>> z = z1\*z2 ; multiplication

>> z = z1/z2 ; division

**II. VECTORS**

 **Generation of vectors**

Example: Generate the vectors x = [1 3 5] and y = [ 2 0 4 5 6]

 >> x = [1 3 5] ; generates the vector of length 3

 >> y = [2 0 4 5 6] ; generates the vector of length 5

**Addition or Subtraction of Vectors x and y of same length**

>> z = x+ y ; addition

>> z = x - y ; subtraction

**Multiplication or Division of Vectors x and y of same length**

>> z = x**.** \* y ; multiplication

>> z = x**.** / y ; division

Note: The dot after x is necessary since x is a vector and not a number.

## MATLAB TOOLBOXES

## MATLAB commands are divided into different toolboxes depending on the applications. Various toolboxes developed by MATLAB include:

**Communication Toolbox**

**Image Processing Toolbox**

## Signal Processing Toolbox

## Controls Toolbox

## Spline Toolbox

## Neural Network Toolbox

## Nonlinear Control Design Toolbox

## Statistics Toolbox

## Optimization Toolbox

## Symbolic Math Toolbox

## Partial Differential Equation Toolbox

## PROGRAMMING WITH VECTORS

Programs involving vectors can be written using either FOR LOOPS or VECTOR commands. Since MATLAB is basically a vector based program, it is often more efficient to write programs using VECTOR commands. However, FOR LOOPS give a clearer understanding of the program, especially for the beginner:

**Example: Sum the following series:**

 **S = 1 + 3 + 5 . . . . . . .99.**

### FOR LOOP approach

>> S = 0.0; ; initializes the sum to zero

>> for i = 1 : 2 : 99;

 S = S + i;

 end;

>> S ; gives the value of the sum

### VECTOR approach

>> i =1: 2 : 99; ; creates the vector i

>> S = sum ( i ); ; obtains the sum S

**Example: Generate the discrete-time signal y(n) = n sin(n/2) in the interval 0 ≤ n ≤ 10.**

### FOR LOOP approach

>> for n = 1:1: 11;

 n1 = n – 1;

 y(n) = n1 \* sin(pi\*n1/2);

 end;

>> y ; gives the vector y

>> n = 0:1:10; ; generates the vector n

>> stem(n,y); ; plots the signal y vs. n with impulses

### VECTOR approach

>> n = 0 : 10; ; creates the vector n

>> y = n**.**\*sin(pi\*n/2); ; obtains the vector y

>> stem(n,y) ; plots the signal y vs. n with impulses

**SIMULINK COMMANDS AND EXAMPLES**

After logging into MATLAB, you will receive the prompt >>. In order to open up SIMULINK, either click on the SIMULINK icon or type in the following:

>> simulink

**GENERAL SIMULINK OPERATIONS**

**Two** windows will open up: the **model** **window** and the **library** **window**. The **model window** is the space utilized for creating your simulation model. In order to create the model of the system, components will have to be taken from the **library** using the computer mouse, and inserted into the model window.

If you browse the library window, the following sections will be seen. Each section can be accessed by clicking on it.

1. **Sources -** This section consists of different signal sources such as sinusoidal, triangular, pulse, random or files containing audio or video signals.
2. **Sinks -** This section consists of measuring instruments such as scopes and displays
3. **Linear** - This section consists linear components performing operations like summing,

 integration, product.

1. **Nonlinear** - Nonlinear operations
2. **Connections** - Multiplexers, Demultiplexers
3. **Toolboxes** - These specify different areas of SIMULINK
4. Communications
5. DSP
6. Neural Nets
7. Simulation Extras

**EDITING, RUNNING AND SAVING SIMULINK FILES**

The complete system is created in the model window by utilizing components from the various available libraries. Once a complete model is created, **save** the model into a file. Click on **Simulation** and select **Run.** The simulation will run, and the **output plots** can be displayed by clicking on the appropriate **sinks**. Save the output plots also into files. The model and output files can be printed out from the files.

**DEMO FILES**

Try out the demo files, both in the **main library** window, and in the **Toolboxes** window. There are several illustrative demonstration files in the areas if **signal processing**, **image processing** and **communications**.

**Simulation and graphical display of continuous-time signals and systems**

**Continuous-time system**

 x(t) = A sin(t)

n(t)

**Analog signal**

**Pseudo-random noise**

**Time scope**

 +

**Time scope**

**Time scope**

Run the simulation for sinusoidal signal, x (t), amplitude of 5 Volts and frequency  = 10 rad./s. The signal n(t) is a pseudo-random noise with maximum amplitude of 0.5 volts.

Observe the combined signal on the time scope, and familiarize yourself with the settings.

(b) Try changing the sinusoidal signal amplitude (2V, 10V), and frequency (20 rad./s, 50 rad./s), and observe the output on the time scope.

**Discrete-time system**

 **+**

 **0.4**

 **z-1**

 **x(n)**

 **y(n)**

 **z/(z2+z-0.3)**

(a) Observe the output signal on the time scope, for an input periodic pulse generator having the following parameters: Pulse amplitude 1 V, Pulse period 2 seconds and pulse width of 1 second.

(b) Try changing the input signal amplitude (2 V, 3V) and pulse width (0.5, 1.5 sec.), and observe on the time scope.