IMAGE PROCESSING WORKSHOP

FEB 2018

SLIDE 1:

INTRODUCTION

About Presenter

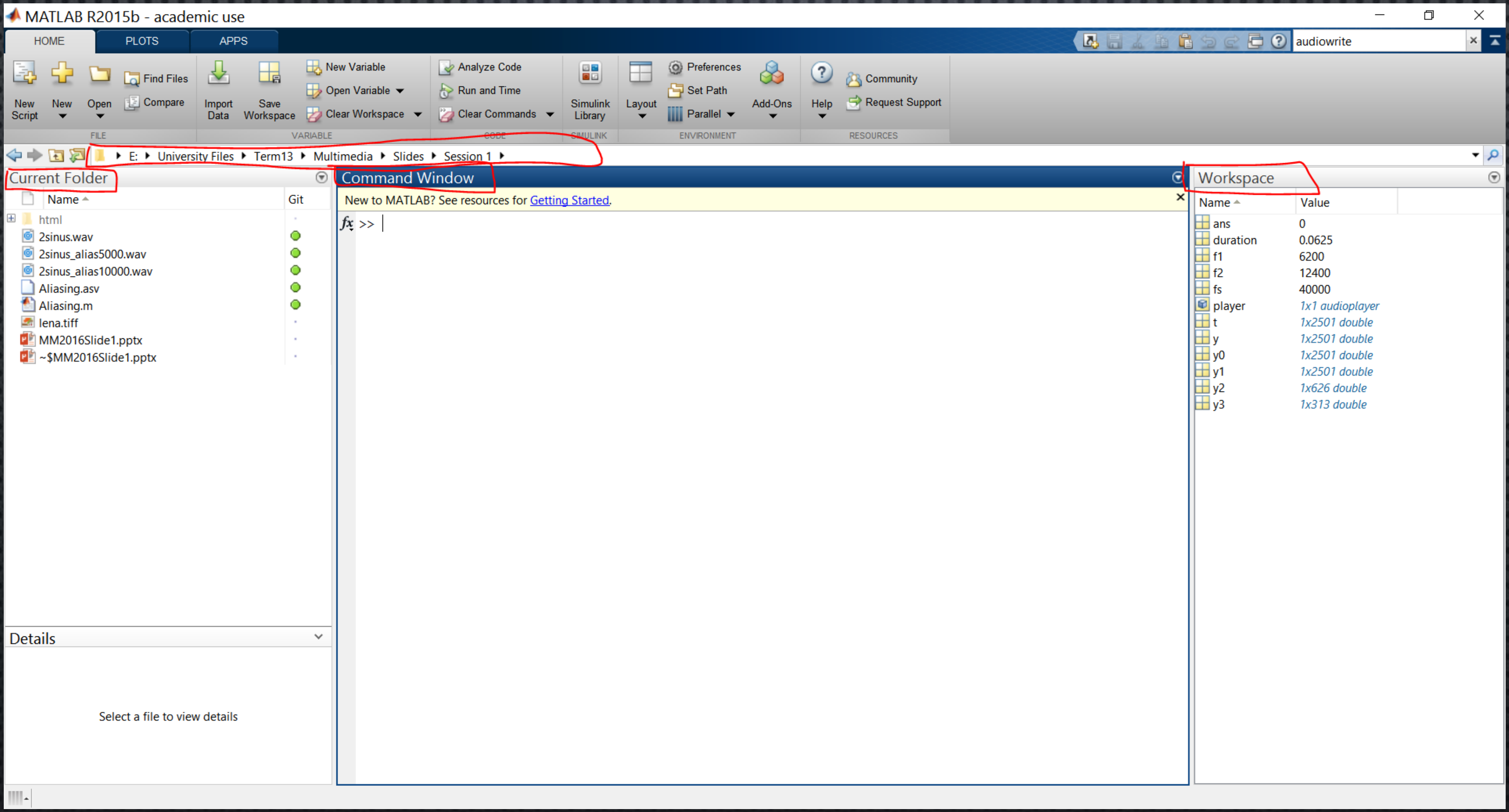
- Founder and CTO, LuxinTech (2014-now)
- Master of Digital Electronics Engineering (2014-2017)
- Bachelor of Electrical Engineering, Communications (2010-2016)
- Bachelor of Electrical Engineering, Electronics (2010-2014)
- TA of several courses: Multimedia Systems (3years), Pulse and Digital Circuits, Electronic Measurement
- Member of SPRL lab (2014-2017), under supervision of Dr. F. Abdollahi
- Member of CMVS lab (2012-2017), under supervision of Dr. S. M. Ahadi

OUTLINE

- Introduction to MATLAB
- Introduction to Image Representation in MATLAB
- Image Filtering in MATLAB
- Edge Detection on images using MATLAB
- Introduction to Cascade Detectors
- Hands-on Cascade Detector Training

WHY MATLAB?

- MATLAB = Matrix Laboratory
- MATLAB is a high-level language and interactive environment that enables you to perform computationally intensive tasks faster than with traditional programming languages such as C, C++ and Fortran.
- MATLAB is an interactive, interpreted language that is designed for fast numerical matrix calculations
- A huge amount of Toolboxes and Code samples (Check out your MATLAB Help)
- A de facto standard framework for academic advancements
  - Except for its competitor in computer vision: OpenCV



Current Folder

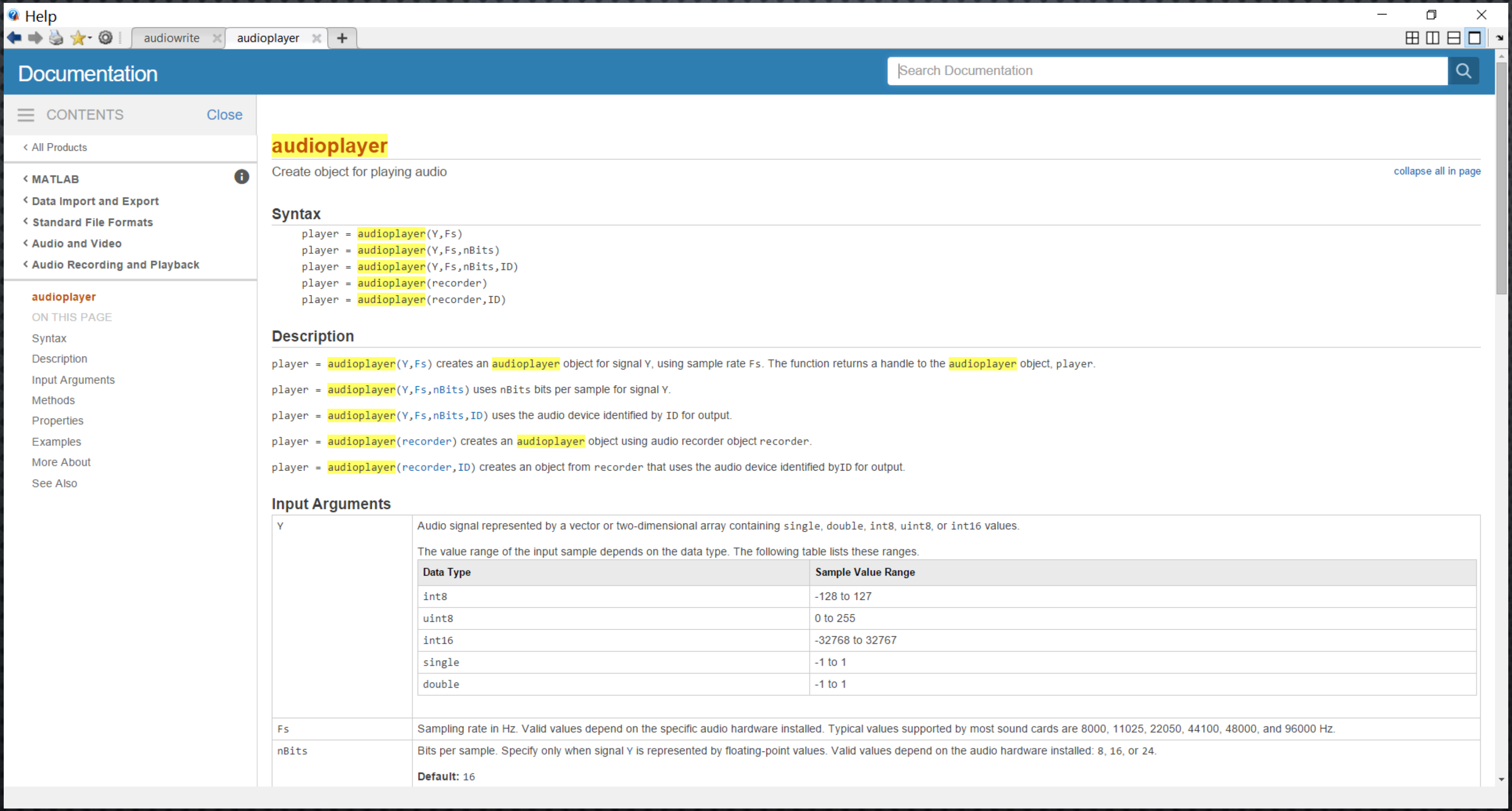
Name	Git
html	
2sinus.wav	●
2sinus_alias5000.wav	●
2sinus_alias10000.wav	●
Aliasing.asv	●
Aliasing.m	●
lena.tiff	
MM2016Slide1.pptx	
~$MM2016Slide1.pptx	

Command Window

```
New to MATLAB? See resources for Getting Started.
fx >>
```

Workspace

Name	Value
ans	0
duration	0.0625
f1	6200
f2	12400
fs	40000
player	1x1 audioplayer
t	1x2501 double
y	1x2501 double
y0	1x2501 double
y1	1x2501 double
y2	1x626 double
y3	1x313 double



audioplayer

Create object for playing audio

Syntax

```
player = audioplayer(Y,Fs)
player = audioplayer(Y,Fs,nBits)
player = audioplayer(Y,Fs,nBits,ID)
player = audioplayer(recorder)
player = audioplayer(recorder,ID)
```

Description

`player = audioplayer(Y,Fs)` creates an `audioplayer` object for signal `Y`, using sample rate `Fs`. The function returns a handle to the `audioplayer` object, `player`.

`player = audioplayer(Y,Fs,nBits)` uses `nBits` bits per sample for signal `Y`.

`player = audioplayer(Y,Fs,nBits,ID)` uses the audio device identified by `ID` for output.

`player = audioplayer(recorder)` creates an `audioplayer` object using audio recorder object `recorder`.

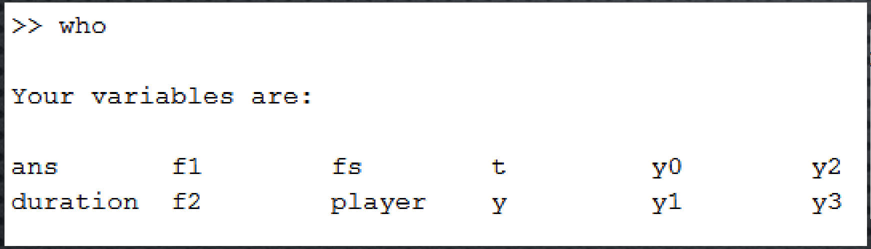
`player = audioplayer(recorder,ID)` creates an object from `recorder` that uses the audio device identified by`ID` for output.

Input Arguments

<table>
<tr><td><code>Y</code></td><td>Audio signal represented by a vector or two-dimensional array containing <code>single</code>, <code>double</code>, <code>int8</code>, <code>uint8</code>, or <code>int16</code> values.
The value range of the input sample depends on the data type. The following table lists these ranges.
<table>
<tr><th>Data Type</th><th>Sample Value Range</th></tr>
<tr><td><code>int8</code></td><td>-128 to 127</td></tr>
<tr><td><code>uint8</code></td><td>0 to 255</td></tr>
<tr><td><code>int16</code></td><td>-32768 to 32767</td></tr>
<tr><td><code>single</code></td><td>-1 to 1</td></tr>
<tr><td><code>double</code></td><td>-1 to 1</td></tr>
</table></td></tr>
<tr><td><code>Fs</code></td><td>Sampling rate in Hz. Valid values depend on the specific audio hardware installed. Typical values supported by most sound cards are 8000, 11025, 22050, 44100, 48000, and 96000 Hz.</td></tr>
<tr><td><code>nBits</code></td><td>Bits per sample. Specify only when signal <code>Y</code> is represented by floating-point values. Valid values depend on the audio hardware installed: <code>8</code>, <code>16</code>, or <code>24</code>.
<b>Default:</b> <code>16</code></td></tr>
</table>

MATLAB Basics

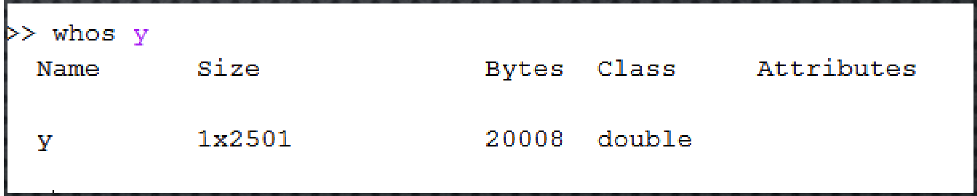
- To find out the type and value of a variable you are using: who, whos



```
>> who

Your variables are:

ans        f1         fs         t          y0         y2
duration   f2         player     y          y1         y3
```



```
>> whos y
  Name       Size                Bytes  Class      Attributes

  y          1x2501              20008  double
```

- Saving and loading variables to *.mat files: save, load
- A good practice to include in your files before doing anything:
  - clear all, clc, close all, fclose all;

MATLAB BASICS

- Main matlab variable type: Matrix
- Creating matrices:
  - A = [1 2 3; 4 5 6; 7 8 9]
  - Special Matrices: zeros, ones, rand, randn, eye
  - Finding size of matrix: size(A) = [3 3]
  - Sum(A), sum(A, 2), sum(sum(A))
  - CODE
- Finding something in a matix:
  - Find(A < 4)

CONTROL STATEMENTS

- If statement
- Switch statement
- For loop
- While loop
  - Continue
  - Break
  - End

- Whenever in doubt, use MATLAB Help

M-Files

- There are 2 kinds of *.m files
    - Functions
        - Well, you should know what functions are, if not, Google it ☺
        - We will get into more details later
    - Scripts
        - Scripts are simple MATLAB code that will be run line by line
        - It's like you would run them line by line in command window

Any Questions?

BTW, we have GIT!

Check my website for getting the GIT link

www.nmahmoudi.ir/teaching.php