

## Audio Files

Matlab can read WAV, AU, or SND files.

**Note: wavread will be removed in a future release. Use audioread instead (reads mp3).**

**Waveform Audio File Format (WAV)** is a Microsoft and IBM audio file format standard for storing an audio bitstream on PCs

Uncompressed WAV files are large, so file sharing of WAV files over the Internet has declined in popularity. However, it is still a commonly used file type, suitable for retaining "first generation" archived files of high quality.

```
[y, Fs, nbits] = wavread('LR.wav');
```

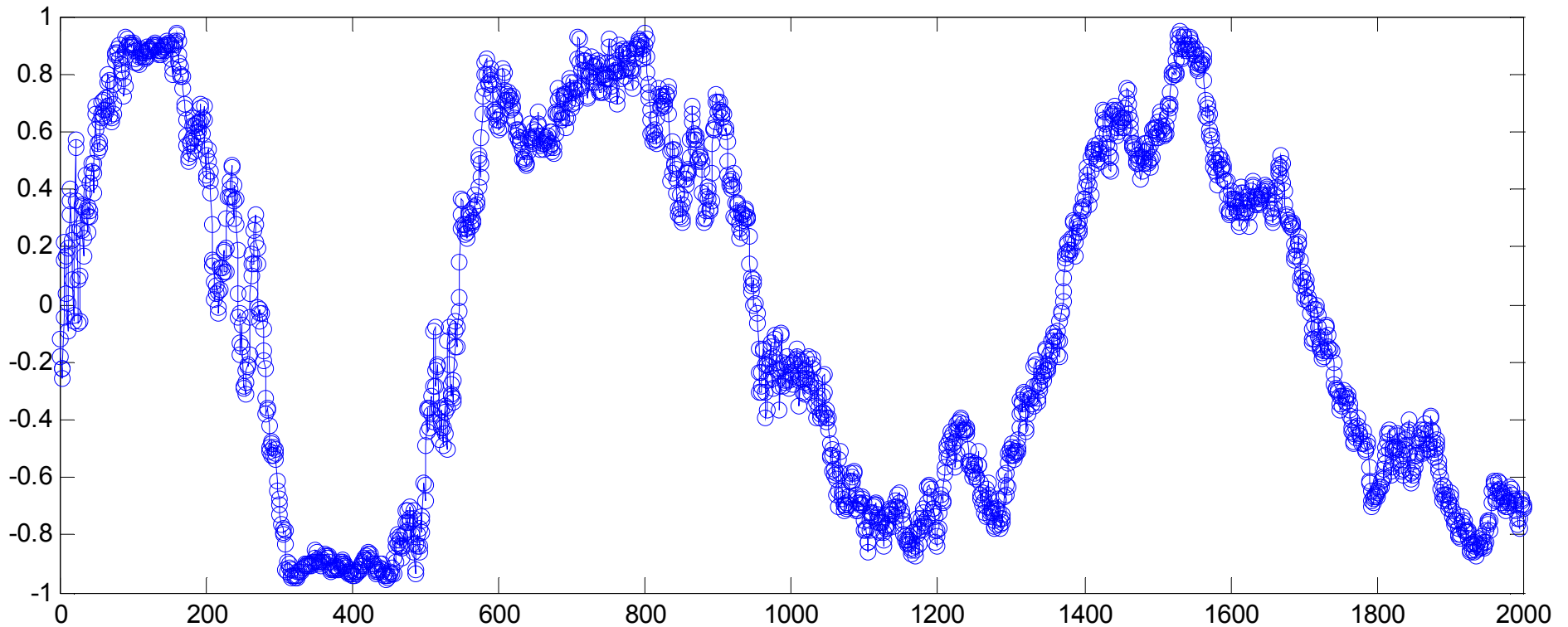
loads a WAV file specified returning the ***sampled data in y*** the ***sample rate (Fs)*** in Hertz used to encode the data in the file the ***number of bits per sample (nbits)***.

# Audio Files

The audio signal in a file represents a series of *samples* that capture the amplitude of the sound over time.

**figure;**

```
plot(y(12000:13000,2),'o','markersize',5,'markerfacecolor','c')
```



## Audio Files

The ***sample rate*** is the number of discrete samples taken per second and given in hertz. Typical values supported by most sound cards are 8000, 11025, 22050, and 44100 Hz.

The precision of the samples, measured by the ***bit depth*** (number of bits per sample), depends on the available audio hardware.

MATLAB audio functions read and store single-channel (***mono***) audio data in an ***m-by-1 column vector***, and ***stereo*** data in an ***m-by-2 matrix***. In either case,  $m$  is the number of samples. For stereo data, the first column contains the left channel, and the second column contains the right channel.

## Playing Audio

### `sound(y, Fs, nbits)`

sends audio signal  $y$  to the speaker at sample rate  $F_s$ . If you do not specify a sample rate, sound plays at 8192 Hz and specifies the bit depth. If you do not specify bits, the sound function plays at an 8-bit depth.

For more flexibility during playback, including the ability to pause, resume, or define callbacks, use the audioplayer function

```
player = audioplayer(y, Fs, nbits);
```

Then you can:

```
play(player) or stop(player) or pause(player) or  
resume(player)
```

# Playing Audio

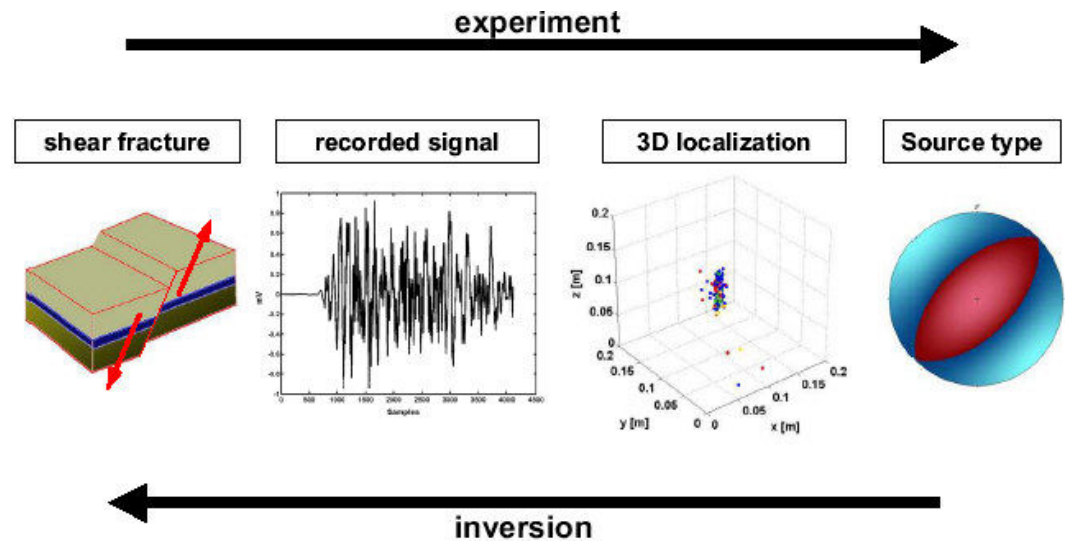
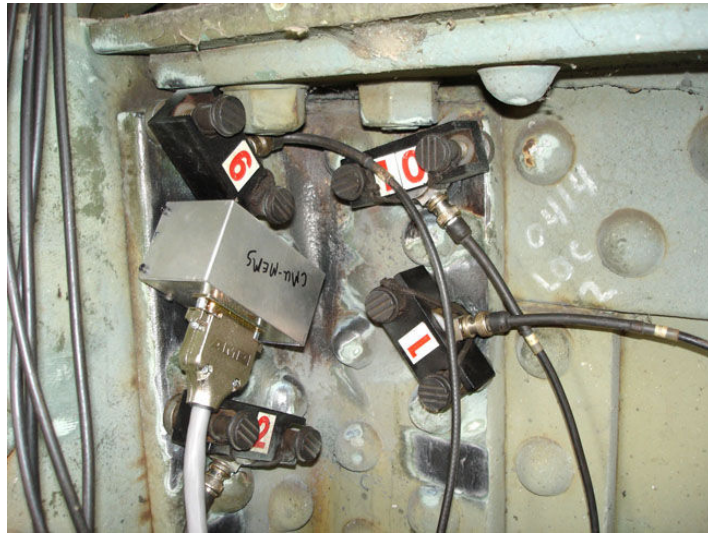
The effect of the sampling frequency:

```
player = audioplayer(y, 1.5*Fs, nbits); play(player)  
player = audioplayer(y, 0.8*Fs, nbits); play(player)
```

The effect of "bit depth":

```
y2 = round(10*y)/10;  
player2 = audioplayer(y2, Fs, nbits); play(player2)
```

# Related Engineering Applications



**Health monitoring applications, acoustic emission, ultrasound testing, Electrocardiogram (ECG – heart sounds)... once you have the digitized data, the possibilities are endless.**

MATLAB®

