Audio Signal Processing in MATLAB

Youssef Abdelilah
Senior Product Manager
Agenda

- Tunable parametric equalizer example
- Audio tone removal example

1. How to create a streaming test bench for audio processing in MATLAB
2. How to develop algorithms and incorporate them into the test bench
3. How to accelerate simulation for real-time performance
Stream processing of audio is everywhere!

- Tablet/MP3 Player & Smart Phone
- Gaming System
- Automotive Infotainment
- Professional Audio & Music
- Medical Devices-Hearing Aids
Tunable parameter equalizer example

Input

Guitar10min.ogg

a 44.1Khz stereo audio

Custom Audio Algorithm

Parameter Equalizer Filters

Array Plot

Audio Input

Guitar10min.ogg a 44.1Khz stereo audio

Play it

Tune it

Tune parameters in real-time

See it

Create it

Array Plot

Audio Output

Speaker

Hear it

Visualize audio waveforms in real-time

MIDI Control
Challenges in audio system design

Framework for real-time simulations

“I have to process large data and test my simulations with streaming signals. I need a simulation testbench that can keep up with real-time data.”

Quick Innovation & modeling

“I need to find innovative algorithms and create and model a working system very quickly.”

Rapid prototyping & simulation acceleration

“I need to optimize my high-level MATLAB algorithm for speed. I then need to verify that the optimized code works the same way as the original MATLAB code.”
What DSP System Toolbox offers in MATLAB

Framework for real-time simulations
- Stream processing techniques and hardware peripheral access that speed up simulation and reduce memory footprint

Quick Innovation & modeling
- Pre-defined algorithms as functions and System objects for quick prototyping

Rapid prototyping & simulation acceleration
- Support for C/C++ code generation that enables design continuity and faster simulation
Part 1: Test bench and peripheral access

1. How to create a streaming test bench for audio processing in MATLAB
2. How to develop algorithms and incorporate them into the test bench
3. How to accelerate simulation for real-time performance
Stream processing in MATLAB

- **Streaming techniques** process continuous data from a captured signal or large file by dividing it into “frames” and fully processes each frame before the next one arrives
  - Memory efficient
- Streaming algorithms in DSP System Toolbox provide
  - Implicit data buffering, state management and indexing
  - Simulation speed-up by reducing overhead

How to create a streaming test bench

- **Sources**
  - Microphone
  - Audio Input

- **Algorithm under test**

- **Sinks**
  - Spectrum Analyzer
  - Audio Output
  - Speaker
  - Visualize sound in real-time

- **Spectrum Analyzer**
How to create test bench in MATLAB

%% Create and Initialize
SamplesPerFrame = 1024;
Fs = 44100;

Microphone = dsp.AudioRecorder('SamplesPerFrame',SamplesPerFrame);

Spectra = dsp.SpectrumAnalyzer('SampleRate',Fs);

%% Stream processing loop
tic;

while toc < 20
    % Read frame from microphone
    audioIn = step(Microphone);

    % View audio spectrum
    step(Spectra,audioIn);
End

%% Terminate
release(Microphone)
release(Spectra)
How to automatically create test benches from “Generate DSP Testbench” App

>>HelperGenDSPTestbenchUI
Part 2: Algorithms

1. How to create a streaming test bench for audio processing in MATLAB
2. How to develop algorithms and incorporate them into the test bench
3. How to accelerate simulation for real-time performance
Example 1: Audio Tone Removal

Audio Input

Guitar file @ 44.1Khz stereo audio

250Hz Tone

Noise

Custom Audio Algorithm

Notch/Peak Filter

Tune parameter equalizer in real-time

MIDI Control

Play it

Create it

Hear it

View it

Spectrum Analyzer

Speaker

Audio Output

Visualize audio waveform spectra in real-time

Audio Tone Removal

Tune it

X

Y

Input

Example 1: Audio Tone Removal

Audio Input

Guitar file @ 44.1Khz stereo audio

250Hz Tone

Noise

Custom Audio Algorithm

Notch/Peak Filter

Tune parameter equalizer in real-time

MIDI Control

Play it

Create it

Hear it

View it

Spectrum Analyzer

Speaker

Audio Output

Visualize audio waveform spectra in real-time

Input
How to incorporate algorithm into test bench

%% Create & Initialize
Fs = 44.1e3;
SamplesPerFrame = 1024;
h = dsp.AudioFileReader('guitar10min.ogg');
hp = dsp.AudioPlayer;
% Interfering tone
ftone = 250;
hw = dsp.SineWave('Amplitude',0.8,'SampleRate',Fs,'Frequency',[ftone ftone],'SampleRate',Fs,'SampleRate',Fs);hs = dsp.SpectrumAnalyzer('SampleRate',Fs,'SpectralAverages',5,'ShowLegend',true,'Title',titlestr);

%% Stream processing loop
while ~isDone(h)
  % Read one frame from audio file and add a tone to input audio
  x = step(h) + step(hw);
  % Audio Tone Removal algorithm
  [y,pauseSim,stopSim]HelperAudioToneRemovalProcessing(x,args,param);
  % View audio waveform
  step(hs,[x(:,1),y(:,1),y(:,3)]);
  % Start playing audio
  step(hp, y(:,1:2));
end

%% Terminate
release(h)
release(hs)
release(hp)
Example 2: Tunable audio parametric equalizer

Tune parameter equalizer in real-time

MIDI Control

Tune it

H = Y / X

Transfer Function Estimator

Parameter Equalizer Filters

Audio Input

Create it

Play it

Guitar file @ 44.1Khz stereo audio

See it

Array Plot

Visualize audio waveform in real-time

Hear it

Speaker

Audio Output

Create it
DSP System Toolbox audio related components (supported on Apple/Windows/Linux)

- **Multichannel audio I/O** (Number of channels depends on hardware)
  - **Audio Player/Recorder**
    (Supports multiple devices, one sound driver per MATLAB session)
  - **Audio File Reader/Writer**
  - ASIO low latency driver support on Windows (R)
  - Custom channel mapping

- **Audio signal analysis**
  - Scopes: time, spectrum analyzer, Array plot
  - Transfer function estimator
  - Measurements: SNR, THD, Average power, PeaktoRMS ratio, mean, variance, ...

- **Signal processing algorithms**
  - **FIR, Biquad**, Multirate FIR, FFT, LMS, ...

- **Connectivity**
  - **UDP, MIDI** (simultaneous support for multiple controls on multiple devices)
Part 3: Acceleration of simulation

1. How to create a streaming test bench for audio processing in MATLAB
2. How to develop algorithms and incorporate them into the test bench
3. How to accelerate simulation for real-time performance
Stream processing in real-time
Data acquisition & algorithm times

As long as
Data acquisition + Algorithm processing \(\leq\) Frame time

We have
Real-time signal processing
Stream processing in real-time
Data acquisition & algorithm times

As long as
Data acquisition + Algorithm processing \(\leq\) Frame time
We have
Real-time signal processing
Accelerating algorithm execution*

User’s Code

```
for k=1:max
  x = fft(dat)
  y = 20*log1
```

- **Optimize MATLAB Code**
  - Pre-allocation and vectorization

- **System Objects**
  - Pre-defined efficient implementations of algorithms

- **MATLAB to C**
  - Generate MEX files automatically with MATLAB Coder

- **Parallel Computing**
  - Parallel computations on multicore computers, GPUs, and clusters

*Webinar: Accelerating Simulation for Communications Applications
https://www.mathworks.com/company/events/webinars/wbnr78473.html
function y = audio_algorithm_peqso(u,tunedparams)
% Copyright 2014 The MathWorks, Inc.
persistent PE1 PE2
if isempty(PE1)
    PE1 = parametricEQFilter('Bandwidth',2000,'CenterFrequency',3000,'PeakGaindB',6.02);
    PE2 = ParametricEQFilter('Bandwidth',2000,'CenterFrequency',1000,'PeakGaindB',-6.02);
end
[PE1,PE2] = processstunedparams(tunedparams,PE1,PE2);
v = step(PE1,u);
y = step(PE2,v);

%-------------------------------------
function [PE1,PE2] = processstunedparams(tunedparams,PE1,PE2)
if ~isnan(tunedparams.CenterFrequency)
    PE1.CenterFrequency = tunedparams.CenterFrequency;
end
if ~isnan(tunedparams.Bandwidth)
    PE1.Bandwidth = tunedparams.Bandwidth;
end
if ~isnan(tunedparams.Gain)
    PE1.PeakGaindB = tunedparams.Gain;
end
if ~isnan(tunedparams.CenterFrequency2)
    PE2.CenterFrequency = tunedparams.CenterFrequency2;
end
if ~isnan(tunedparams.Bandwidth2)
    PE2.Bandwidth = tunedparams.Bandwidth2;
end
if ~isnan(tunedparams.Gain2)
    PE2.PeakGaindB = tunedparams.Gain2;
end

(*) Design and Prototype Real-Time DSP Systems with MATLAB (Conference Presentation):
## Simulation acceleration benchmarks

<table>
<thead>
<tr>
<th>2-band parametric equalizer algorithm</th>
<th>Processing time</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLAB code</td>
<td>23.37 seconds</td>
</tr>
<tr>
<td>MEX code</td>
<td>2.84 seconds</td>
</tr>
</tbody>
</table>

**Speed up of \( \sim 8 \times \)**
DSP System Toolbox *

Over 300 algorithms for modeling, designing, implementing and deploying dynamic system applications

- Advanced Filter Design, Adaptive, Multistage and Multi-rate Filters
- FFT, DCT & other Transforms
- Signal processing blocks for Simulink
- Support for Fixed-Point, C/C++ code generation and HDL
- Visualization in Time and Frequency-domain
- System objects and functions in MATLAB
- Stream signal Processing
- ARM Cortex-M support for hardware prototype

Algorithm libraries in MATLAB

Algorithm libraries in Simulink

Create a test bench for audio algorithms with various sources/sinks
- AudioRecorder, AudioPlayer, AudioFileReader, Sine wave, white noise, …
- Spectrum analyzer, time scope, transfer function estimator, …

Use DSP System Toolbox components (System objects) for algorithms
- FIR, FIR Decimation/Interpolation/Rate Conversion, …
- Biquad, Allpass, Allpole, …
- FFT, DCT, Auto/Cross Correlation, …
- Mean, variance, RMS, PeaktoRMS, …

Perform real-time audio signal processing
- Accelerate critical components by code-generation (Generating MEX files using MATLAB Coder)
THANK YOU!