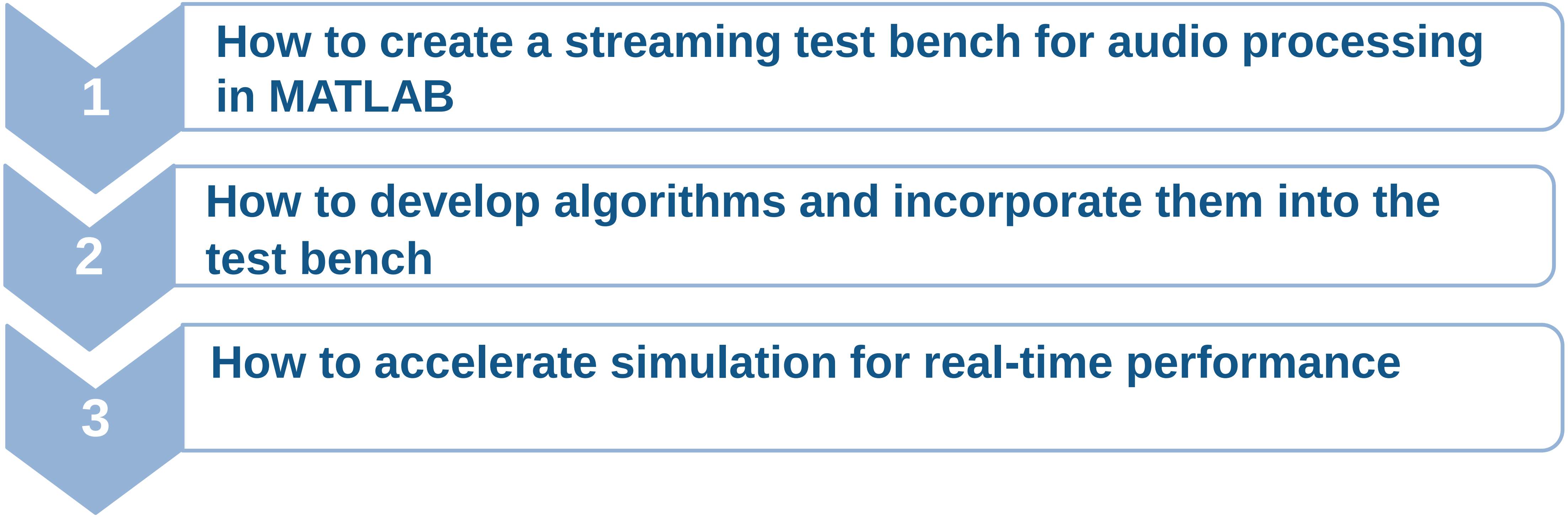
Audio Signal Processing in MATLAB

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!

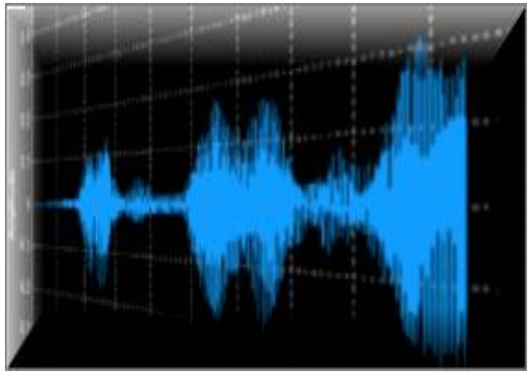
Tablet/MP3 Player & Smart Phone

Gaming System

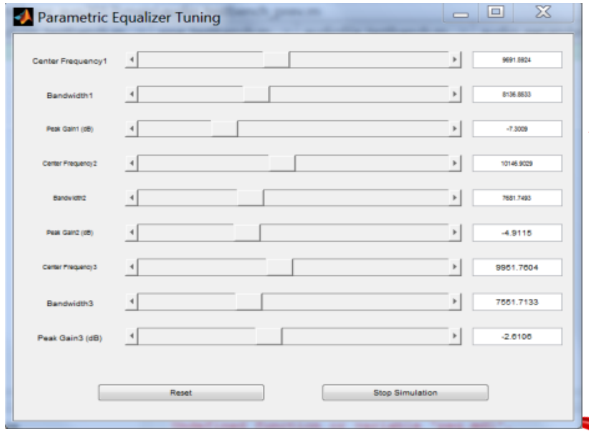
Automotive Infotainment

Professional Audio & Music

Medical Devices-Hearing Aids



Tunable parameter equalizer example



Parametric Equalizer Tuning

Center Frequency1 9691.8824

Bandwidth1 8136.8633

Peak Gain1 (dB) -7.3009

Center Frequency2 10146.9029

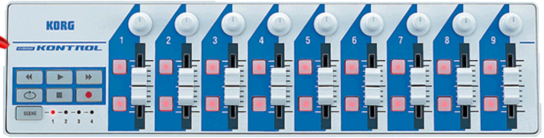
Bandwidth2 7681.7493

Peak Gain2 (dB) -4.9115

Center Frequency3 9951.7604

Bandwidth3 7551.7133

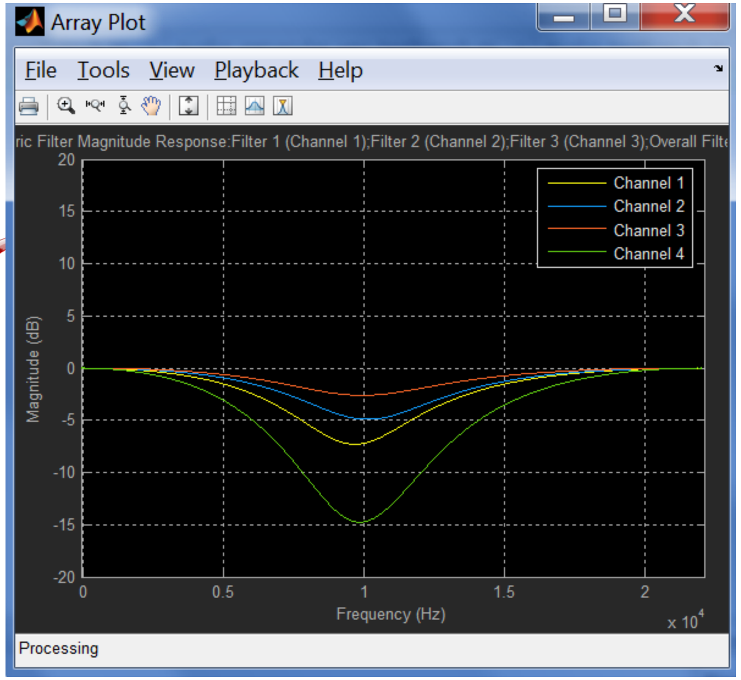
Peak Gain3 (dB) -2.6106



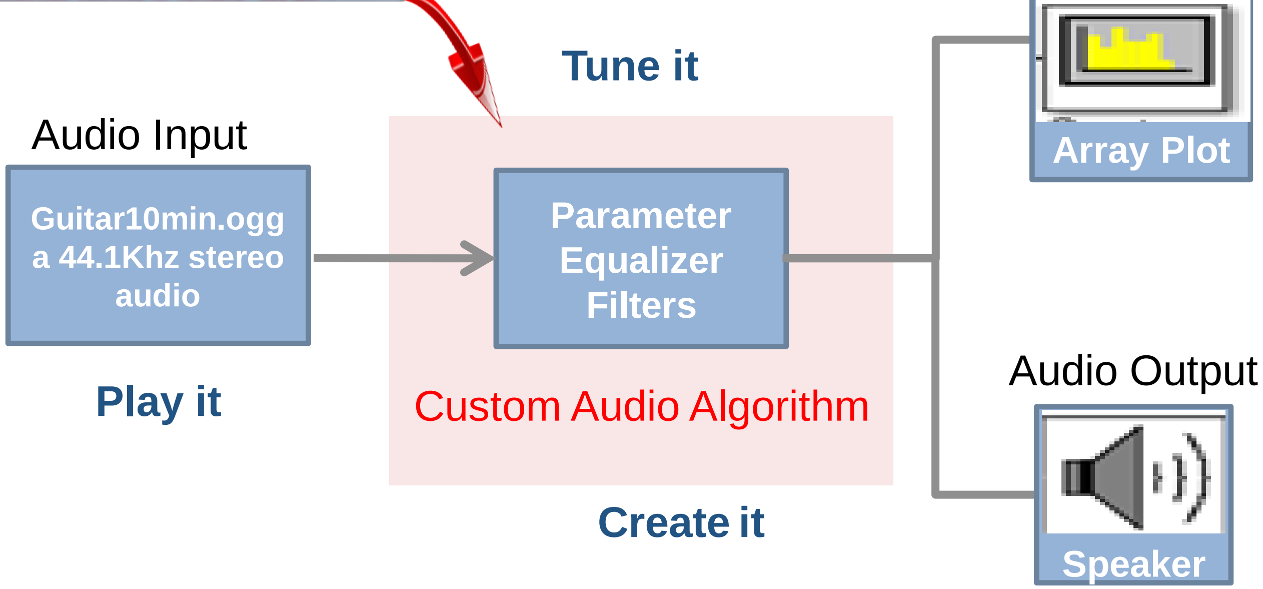
MIDI Control

Tune parameters in real-time

See it



Visualize audio waveforms in real-time


Tune it
Audio Input
Guitar10min.ogg a 44.1Khz stereo audio
Parameter Equalizer Filters
Array Plot
Audio Output
Speaker

Play it

Custom Audio Algorithm

Create it

Hear it

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

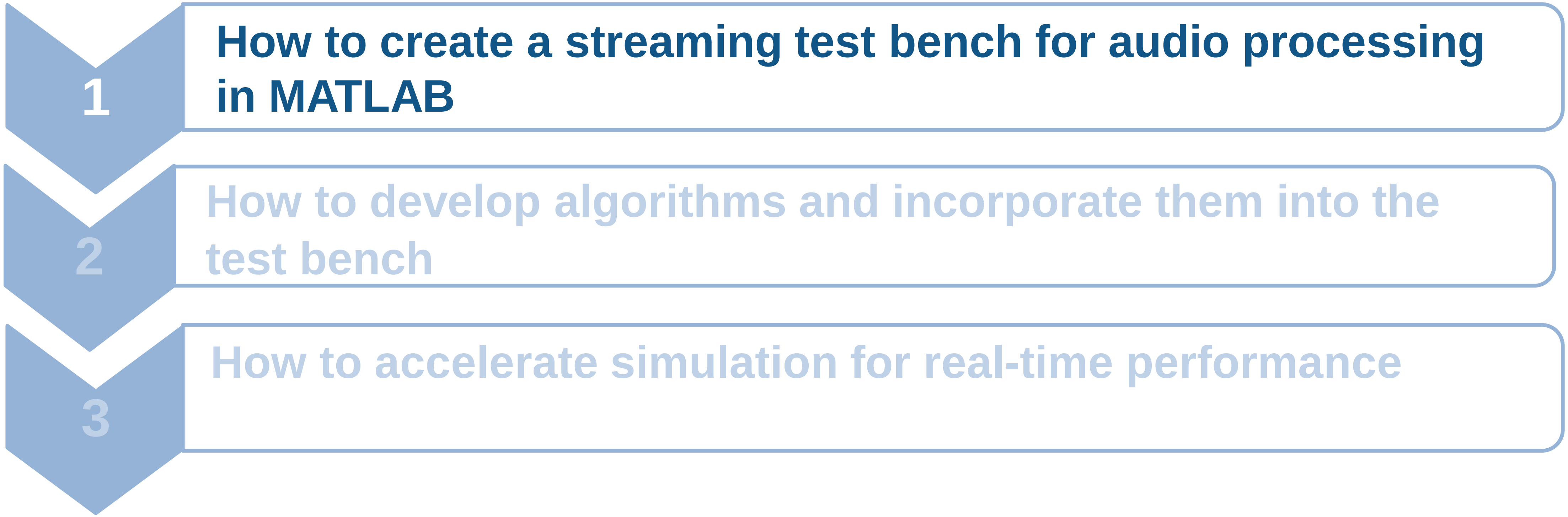
Quick Innovation & modeling

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

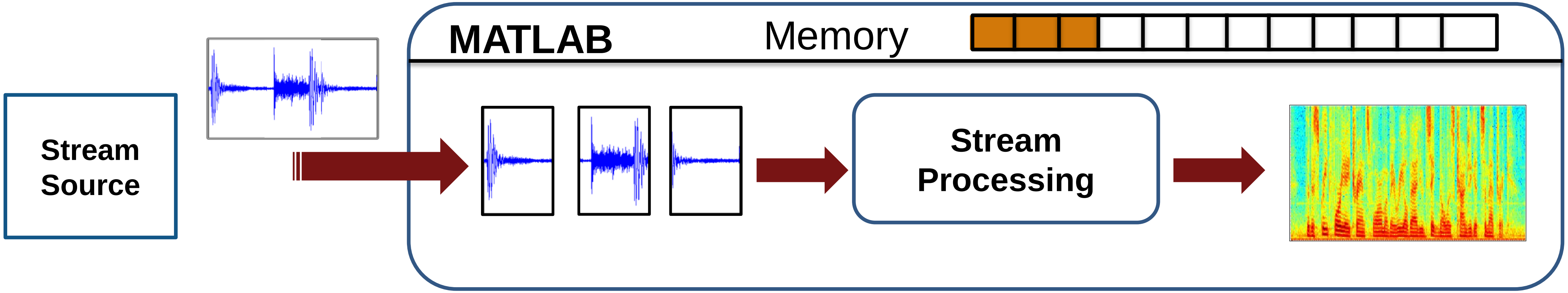
Part 1: Test bench and peripheral access



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



Stream Source

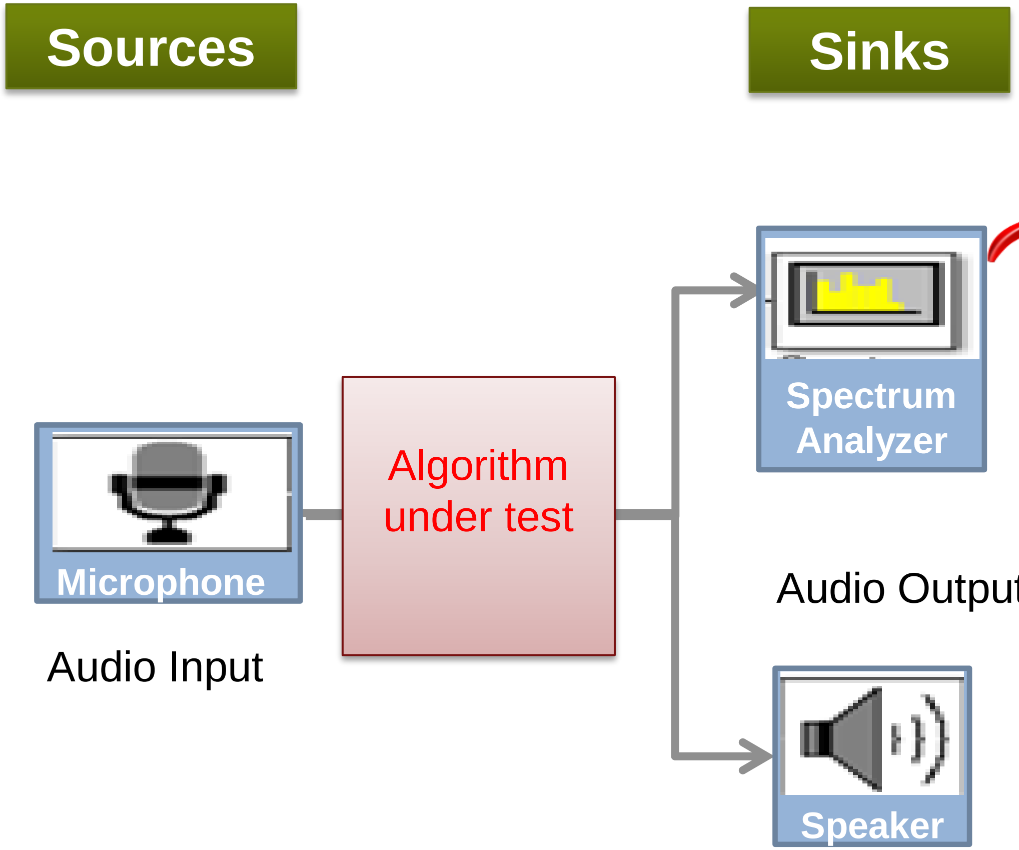
MATLAB

Memory

Stream Processing

*http://www.mathworks.com/discovery/stream-processing.html

How to create a streaming test bench



Sources

Sinks

Microphone

Audio Input

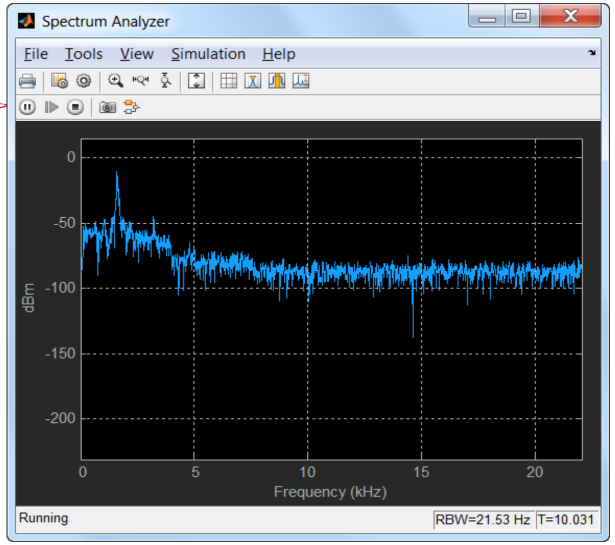
Algorithm under test

Spectrum Analyzer

Audio Output

Speaker

Spectrum Analyzer



Running

RBW=21.53 Hz T=10.031

Visualize sound in real-time

How to create test bench in MATLAB

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

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

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

Initialize

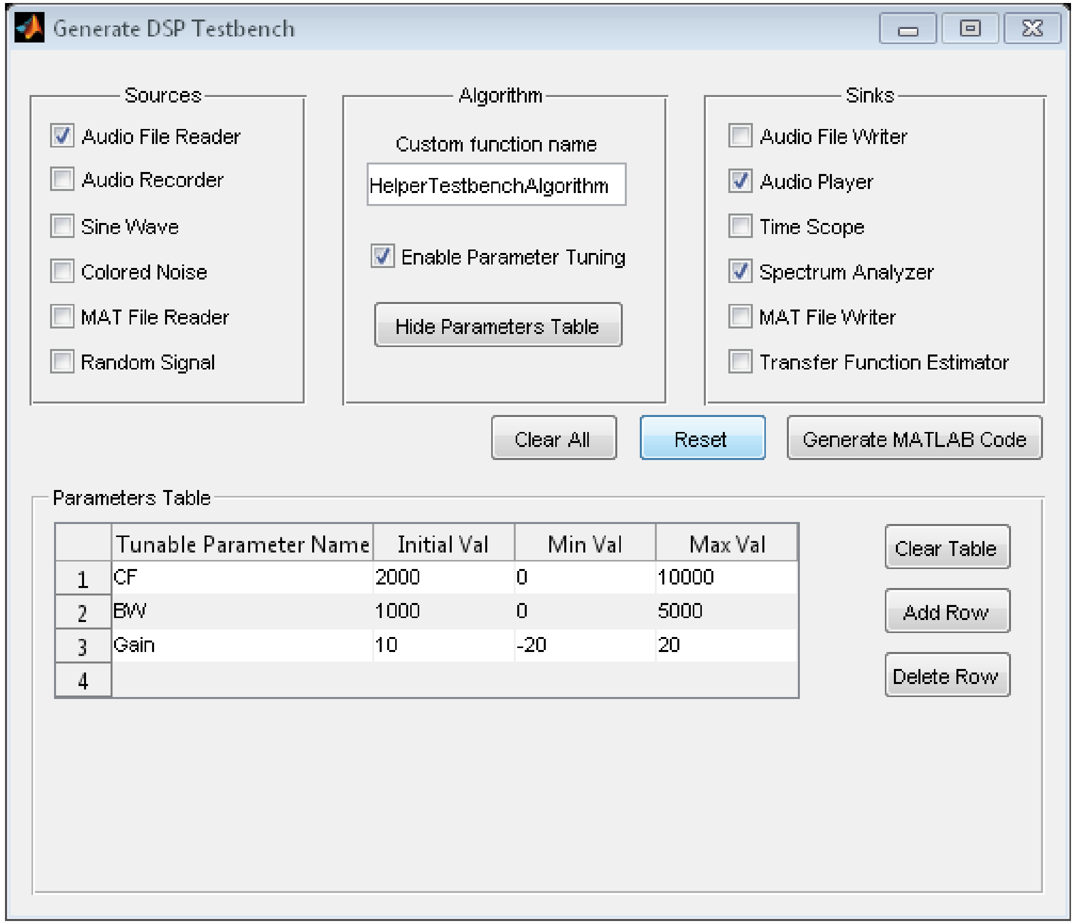
```
%% Stream processing loop
tic;

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

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

Process
in-the-loop

```
%% Terminate
release(Microphone)
release(Spectra)
```

Terminate

How to automatically create test benches from “Generate DSP Testbench” App



Generate DSP Testbench

Sources

- ☑ Audio File Reader
- □ Audio Recorder
- □ Sine Wave
- □ Colored Noise
- □ MAT File Reader
- □ Random Signal

Algorithm

Custom function name: HelperTestbenchAlgorithm

☑ Enable Parameter Tuning

Sinks

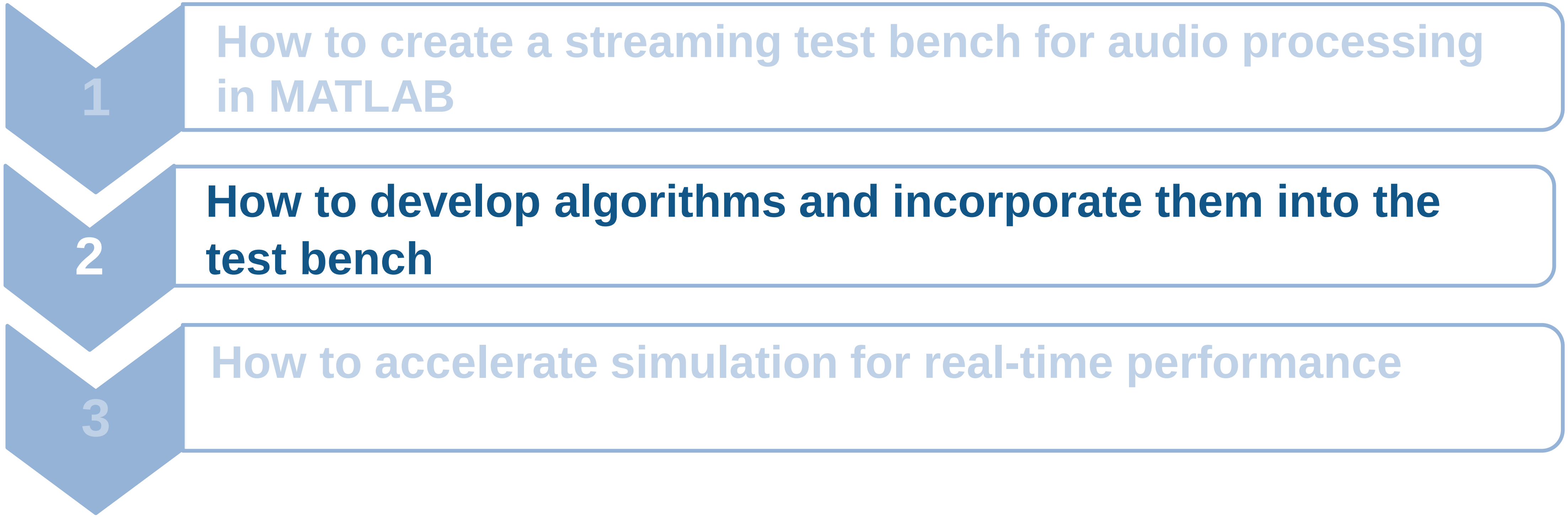
- □ Audio File Writer
- ☑ Audio Player
- □ Time Scope
- ☑ Spectrum Analyzer
- □ MAT File Writer
- □ Transfer Function Estimator

Parameters Table

	Tunable Parameter Name	Initial Val	Min Val	Max Val
1	CF	2000	0	10000
2	BW	1000	0	5000
3	Gain	10	-20	20
4				

```
>>HelperGenDSPTestbenchUI
```

Part 2: Algorithms

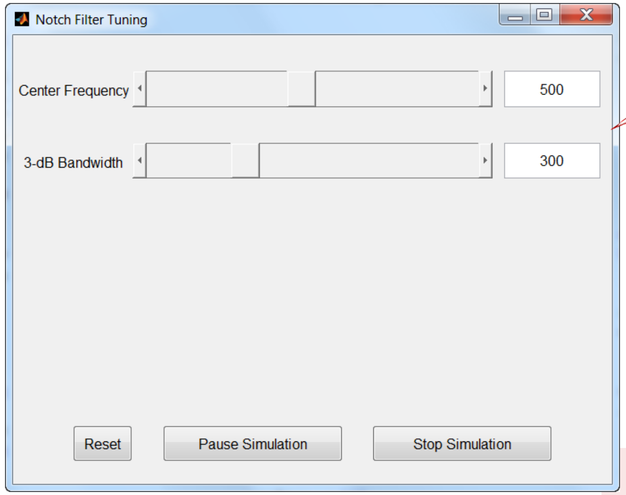


1 How to create a streaming test bench for audio processing in MATLAB

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

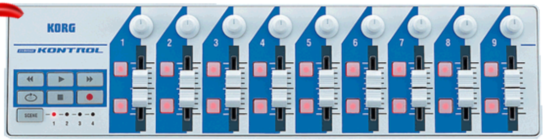


Notch Filter Tuning

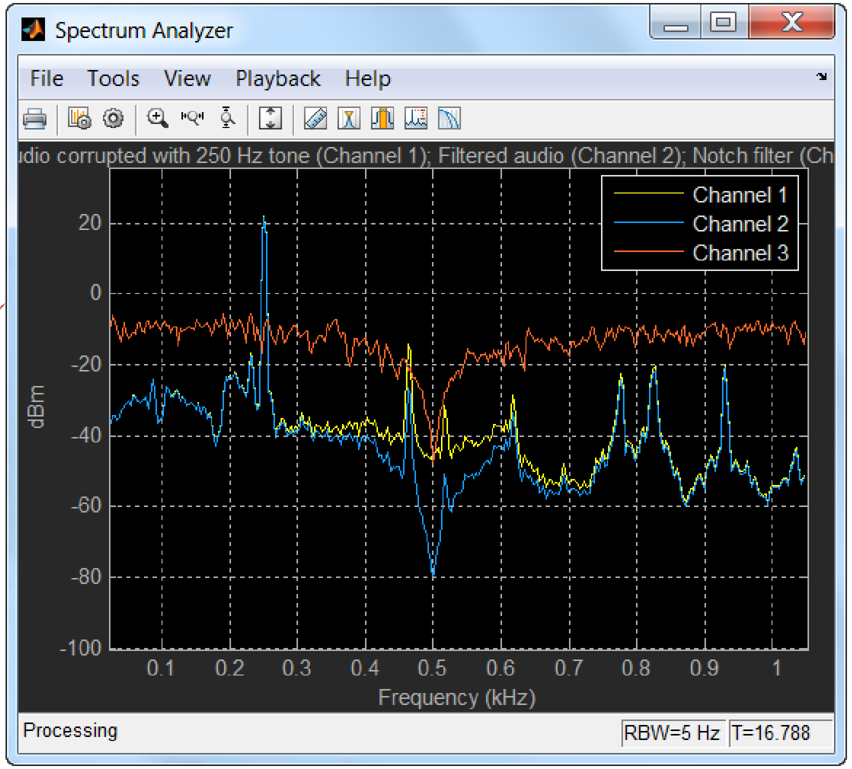
Center Frequency 500

3-dB Bandwidth 300

MIDI Control



Tune parameter equalizer in real-time

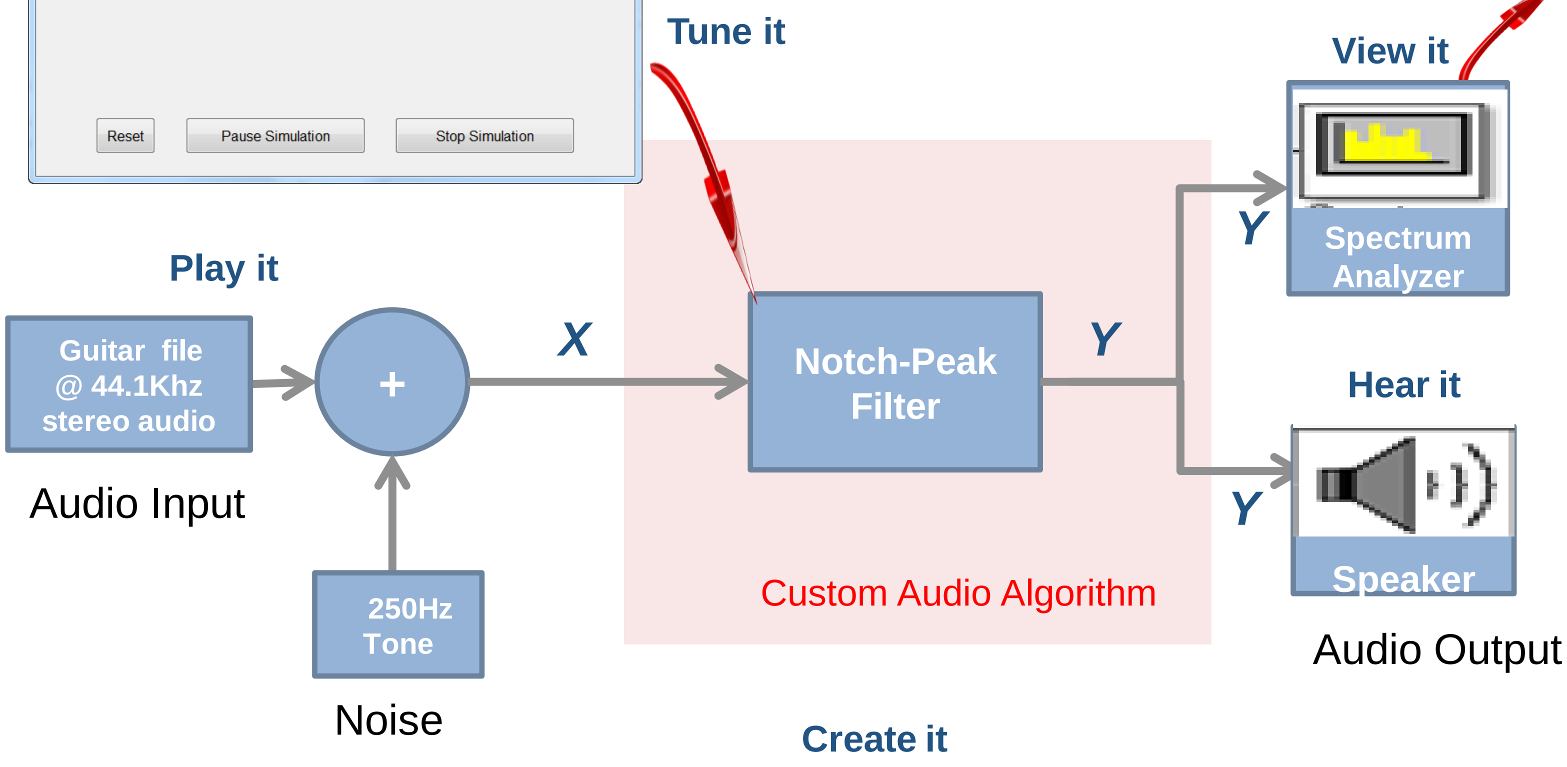


Spectrum Analyzer

udio corrupted with 250 Hz tone (Channel 1); Filtered audio (Channel 2); Notch filter (Ch

Processing

RBW=5 Hz T=16.788



Audio Input

Noise

Create it

Audio Output

Visualize audio waveform spectra in real-time

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],'SamplesPerFrame',SamplesPerFrame);
hs = dsp.SpectrumAnalyzer('SampleRate',Fs,'SpectralAverages',5,'ShowLegend',true,’Title',titlestr);
```

initialize

```
%% 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
```

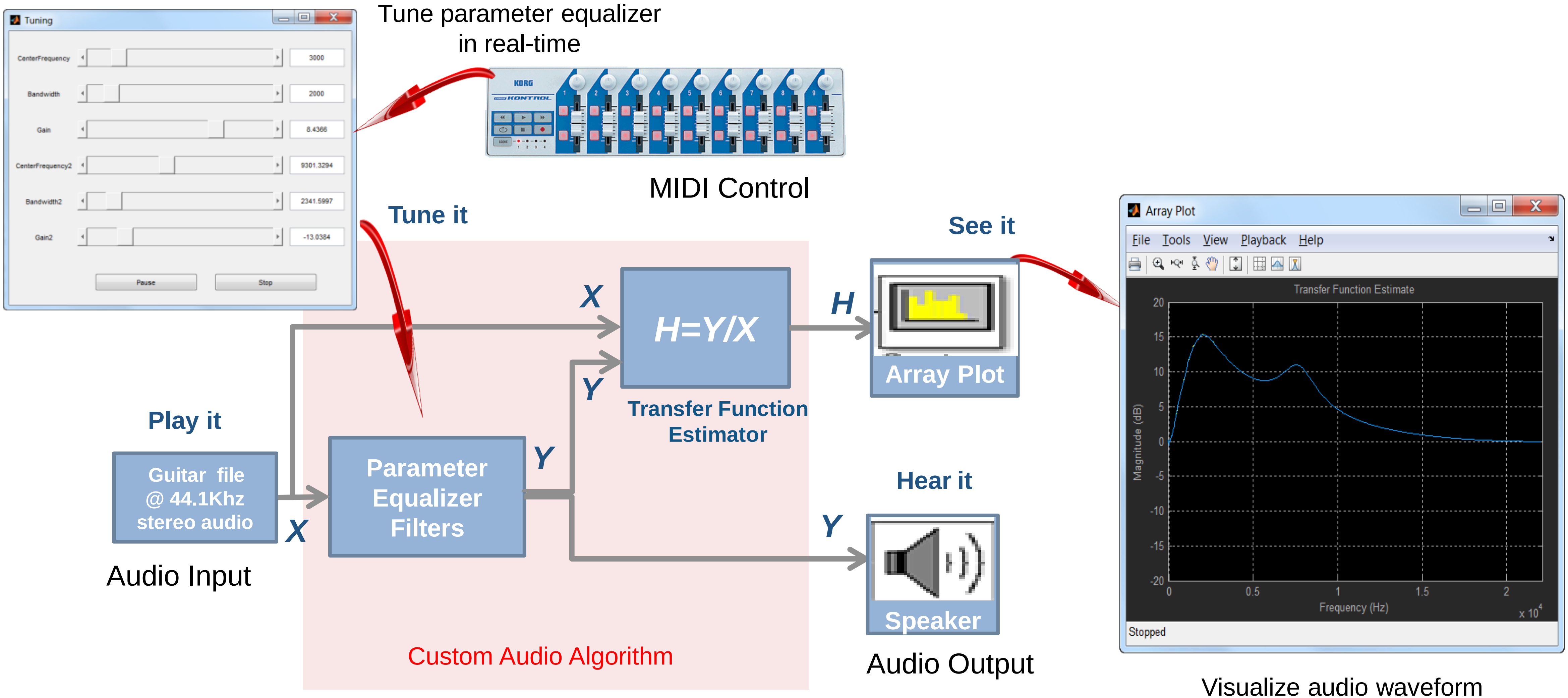
Algorithm

Process in-the-loop

```
%% Terminate
release(h)
release(hs)
release(hp)
```

Terminate

Example 2: Tunable audio parametric equalizer



Tuning

CenterFrequency 3000

Bandwidth 2000

Gain 8.4366

CenterFrequency2 9301.3294

Bandwidth2 2341.5997

Gain2 -13.0384

Pause Stop

Tune parameter equalizer in real-time

MIDI Control

Tune it

See it

Play it

Hear it

Audio Input

Audio Output

Custom Audio Algorithm

KORG
KONTROL
Guitar file @ 44.1Khz stereo audio
X
Parameter Equalizer Filters
Y
X
Y
H=Y/X
Transfer Function Estimator
H
Array Plot
Y
Speaker
Array Plot
File Tools View Playback Help
Transfer Function Estimate
Magnitude (dB)
20
15
10
5
0
-5
-10
-15
-20
0
0.5
1
1.5
2
Frequency (Hz)
x 10^4
Stopped

Visualize audio waveform in real-time

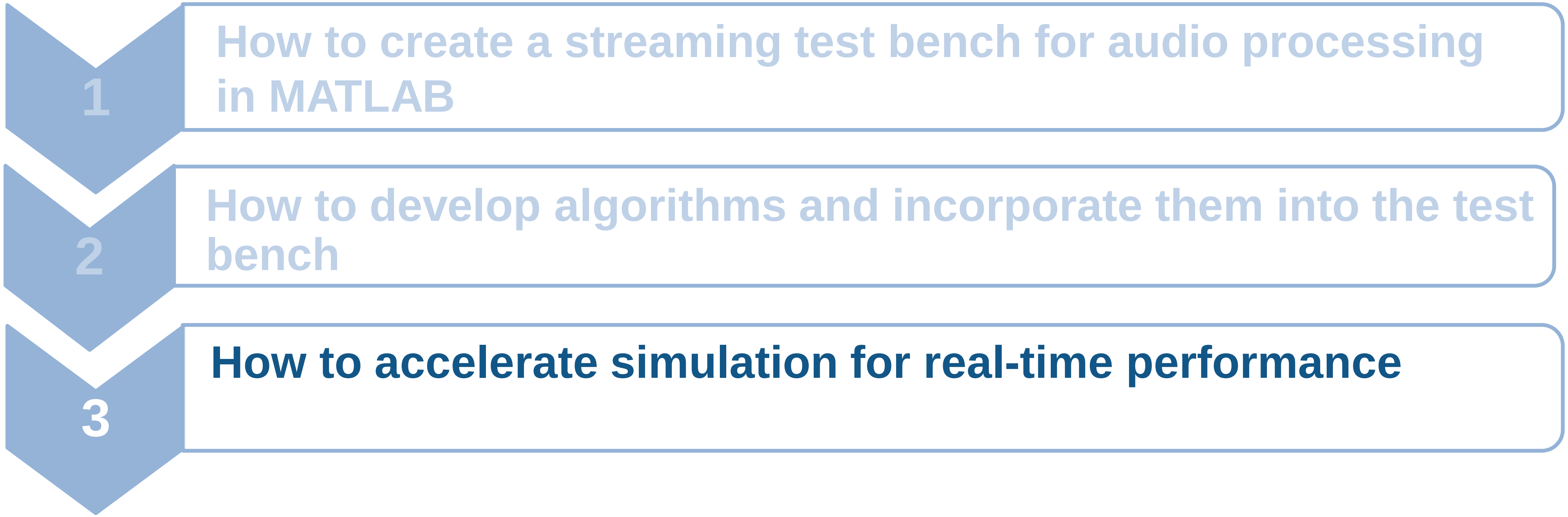
Create it

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

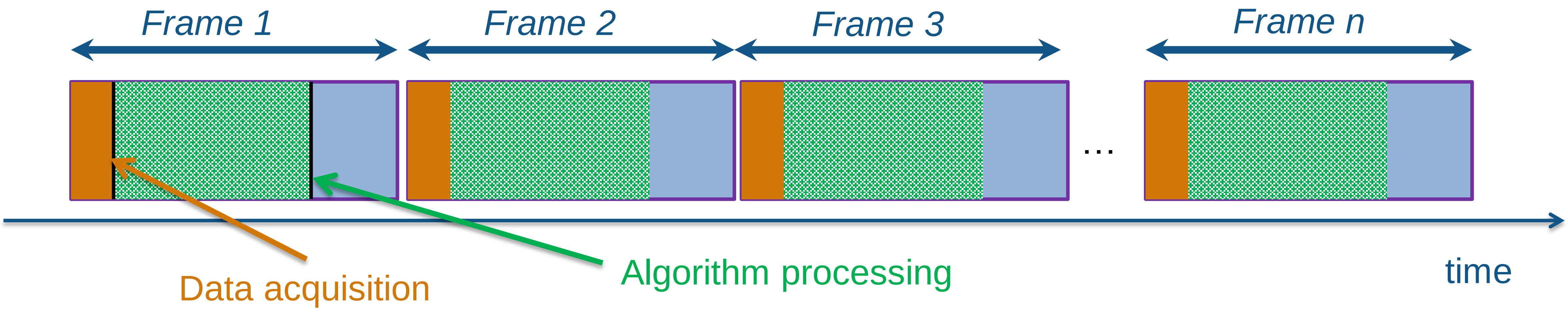


1 How to create a streaming test bench for audio processing in MATLAB

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



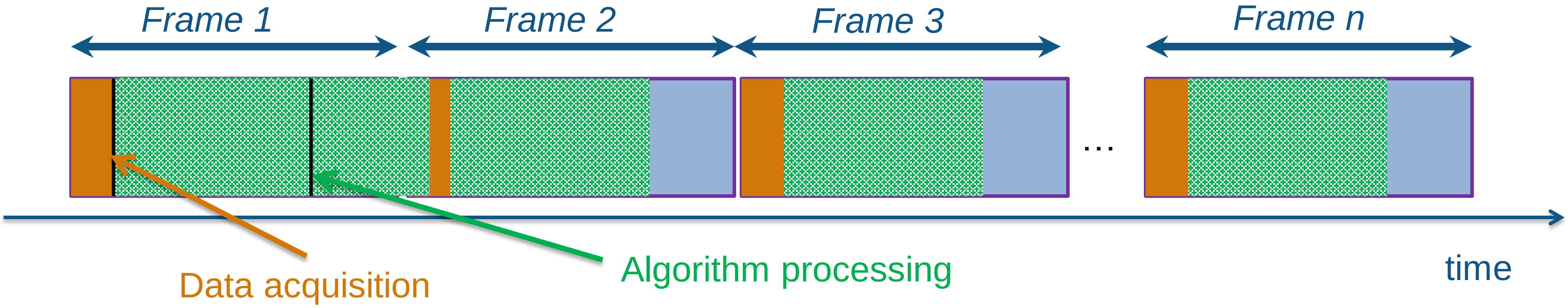
As long as

Data acquisition **+** Algorithm processing **<=** Frame time

We have

Real-time signal processing

Stream processing in real-time
Data acquisition & algorithm times

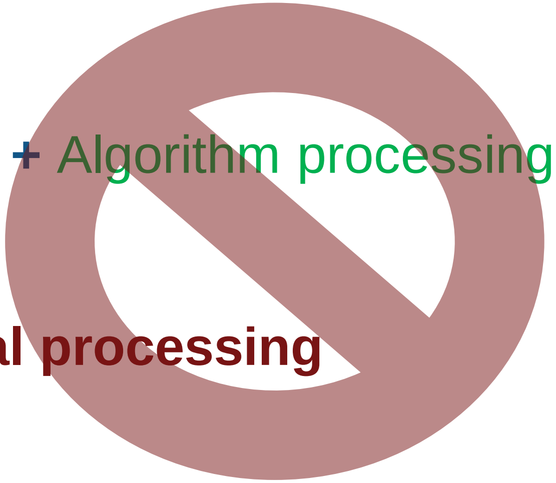


As long as

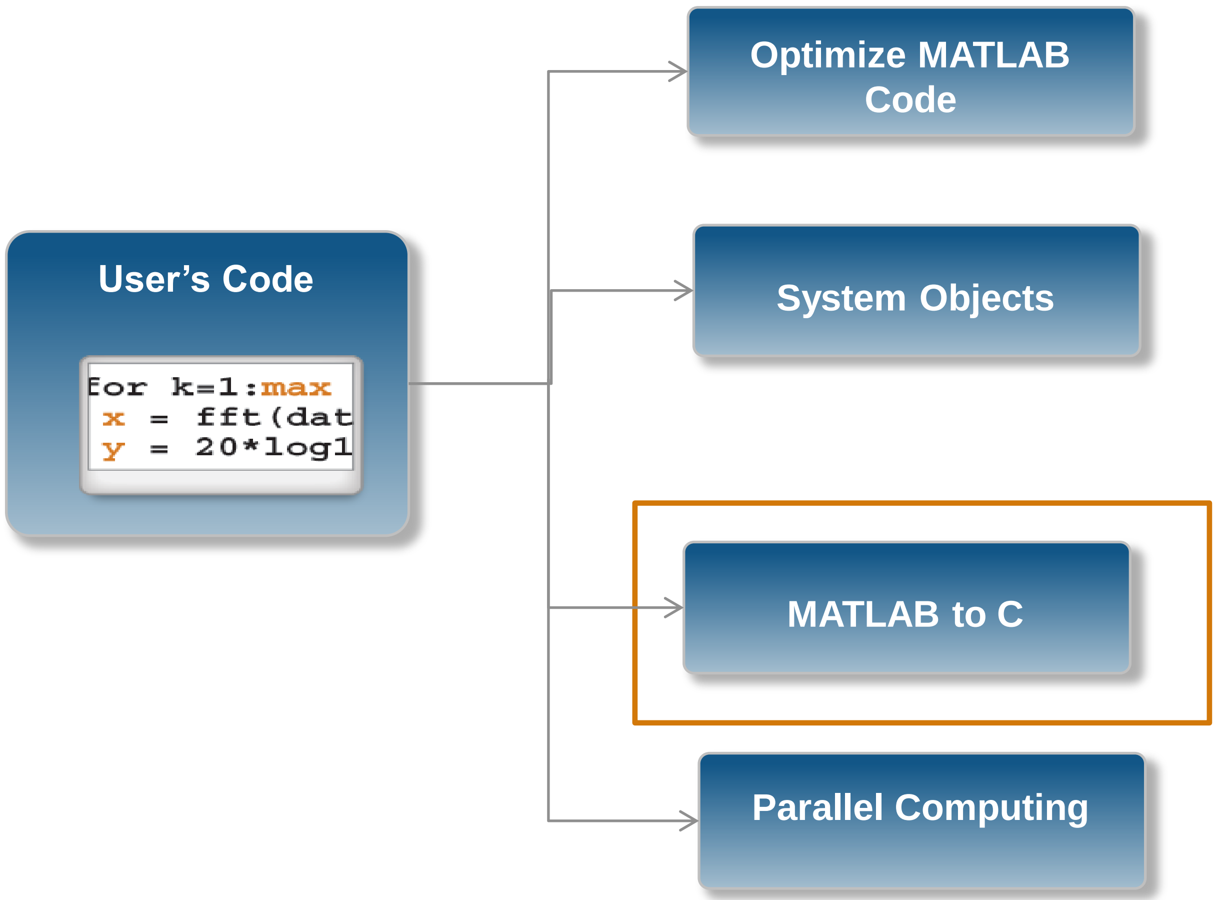
Data acquisition **+** Algorithm processing **<=** Frame time

We have

Real-time signal processing



Accelerating algorithm execution*



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

Pre-allocation and vectorization

Pre-defined efficient implementations of algorithms

Generate MEX files automatically with MATLAB Coder

Parallel computations on multicore computers, GPUs, and clusters

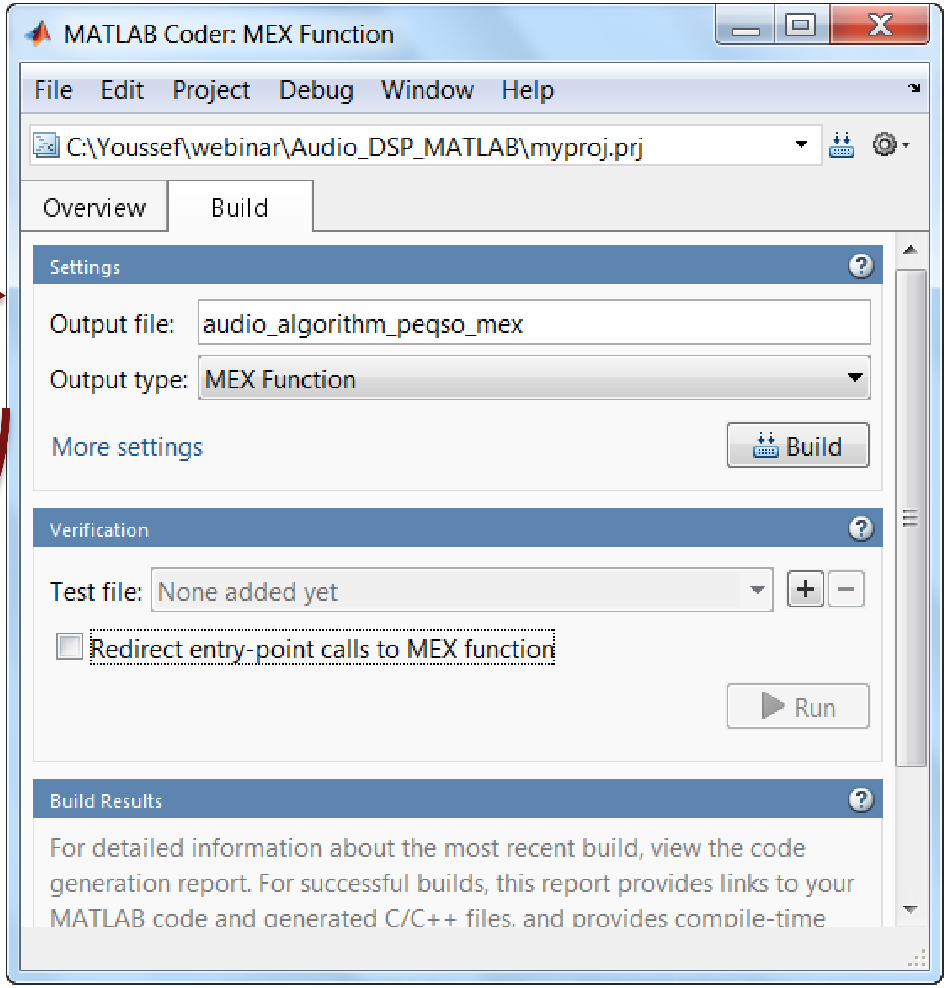
*Webinar: Accelerating Simulation for Communications Applications
https://www.mathworks.com/company/events/webinars/wbnr78473.html

MATLAB to C code generation*

```
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] = processtunedparams(tunedparams,PE1,PE2);
v = step(PE1,u);
y = step(PE2,v);
%-------------------------------------
function [PE1,PE2] = processtunedparams(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
```

Algorithm.m

MATLAB Coder



C:\Youssef\webinar\Audio_DSP_MATLAB\myproj.prj

Output file: audio_algorithm_peqso_mex

Output type: MEX Function

Test file: None added yet

□ Redirect entry-point calls to MEX function

For detailed information about the most recent build, view the code generation report. For successful builds, this report provides links to your MATLAB code and generated C/C++ files, and provides compile-time

Algorithm.mex

[(*) Design and Prototype Real-Time DSP Systems with MATLAB (Conference Presentation):](http://www.mathworks.com/company/events/conferences/matlab-virtual-conference/2013/proceedings/design-and-prototype-real-time-dsp-systems-with-matlab.html)

http://www.mathworks.com/company/events/conferences/matlab-virtual-conference/2013/proceedings/design-and-prototype-real-time-dsp-systems-with-matlab.html

Simulation acceleration benchmarks

2-band parametric equalizer algorithm	Processing time
MATLAB code	**23.37** seconds
MEX code	**2.84** seconds

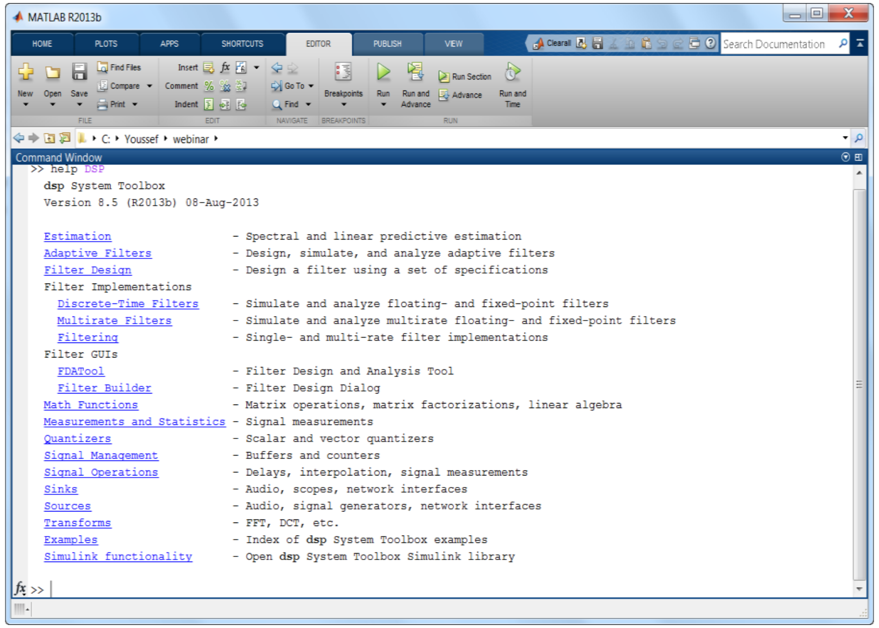
Speed up of ~ 8 x

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

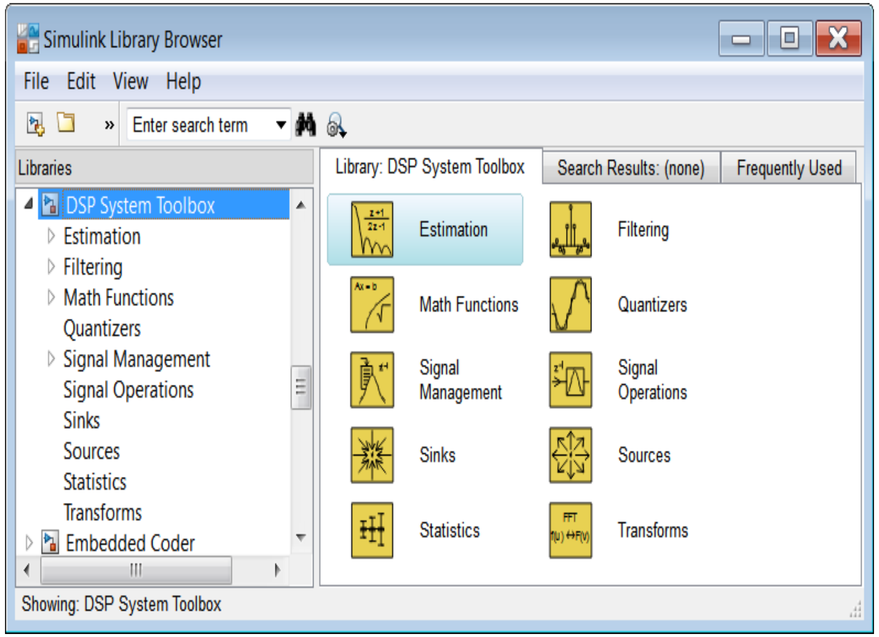
Algorithm libraries in MATLAB



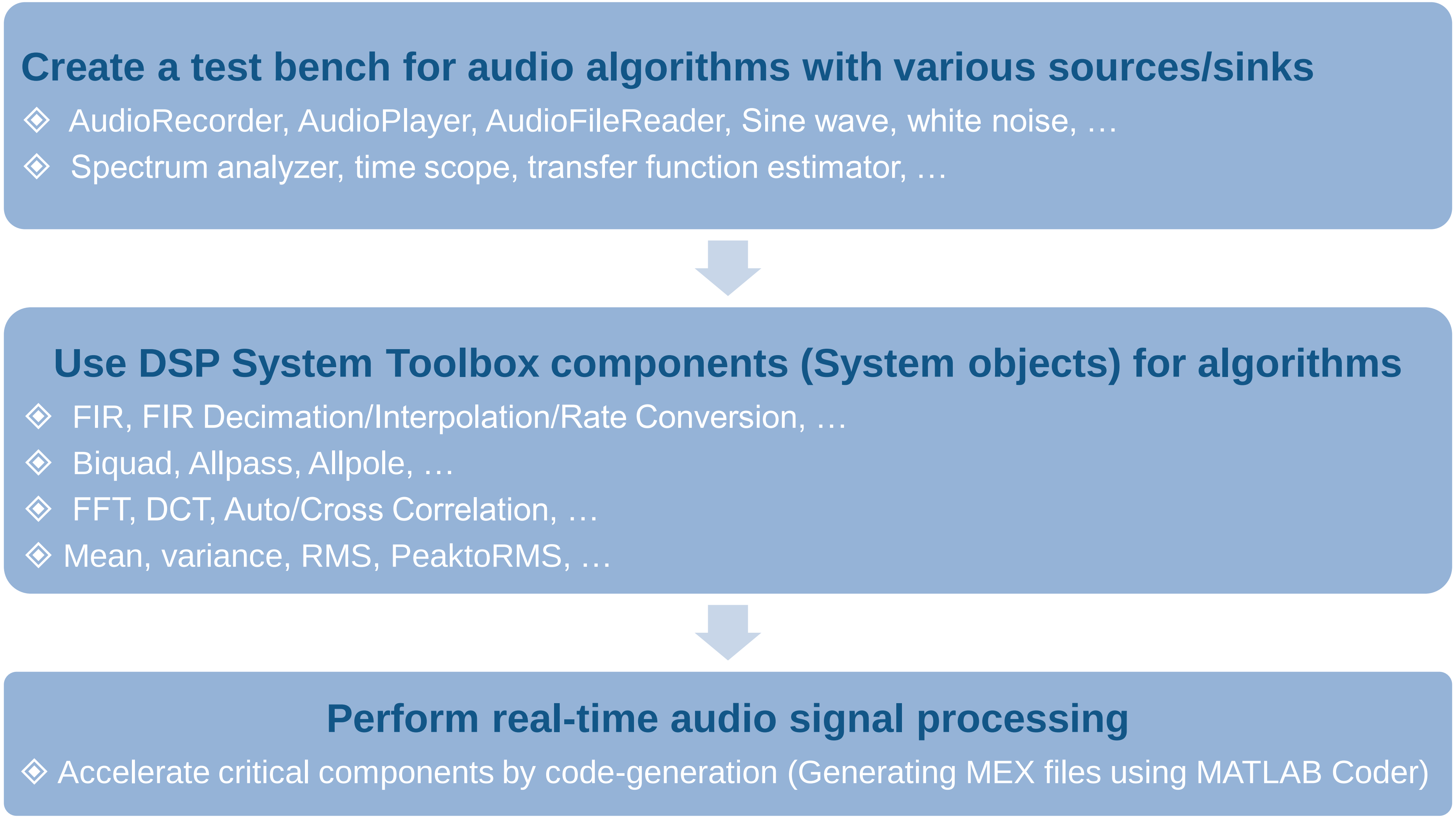
```
>> help DSP
  dsp System Toolbox
  Version 8.5 (R2013b) 08-Aug-2013

  Estimation                   - Spectral and linear predictive estimation
  Adaptive Filters             - Design, simulate, and analyze adaptive filters
  Filter Design                - Design a filter using a set of specifications
  Filter Implementations
    Discrete-Time Filters      - Simulate and analyze floating- and fixed-point filters
    Multirate Filters          - Simulate and analyze multirate floating- and fixed-point filters
    Filtering                  - Single- and multi-rate filter implementations
  Filter GUIs
    FDATool                    - Filter Design and Analysis Tool
    Filter Builder             - Filter Design Dialog
  Math Functions               - Matrix operations, matrix factorizations, linear algebra
  Measurements and Statistics - Signal measurements
  Quantizers                   - Scalar and vector quantizers
  Signal Management            - Buffers and counters
  Signal Operations            - Delays, interpolation, signal measurements
  Sinks                        - Audio, scopes, network interfaces
  Sources                      - Audio, signal generators, network interfaces
  Transforms                   - FFT, DCT, etc.
  Examples                     - Index of dsp System Toolbox examples
  Simulink functionality       - Open dsp System Toolbox Simulink library

fx >>
```

*http://www.mathworks.com/products/dsp-system/index.html

Algorithm libraries in Simulink



Summary



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!