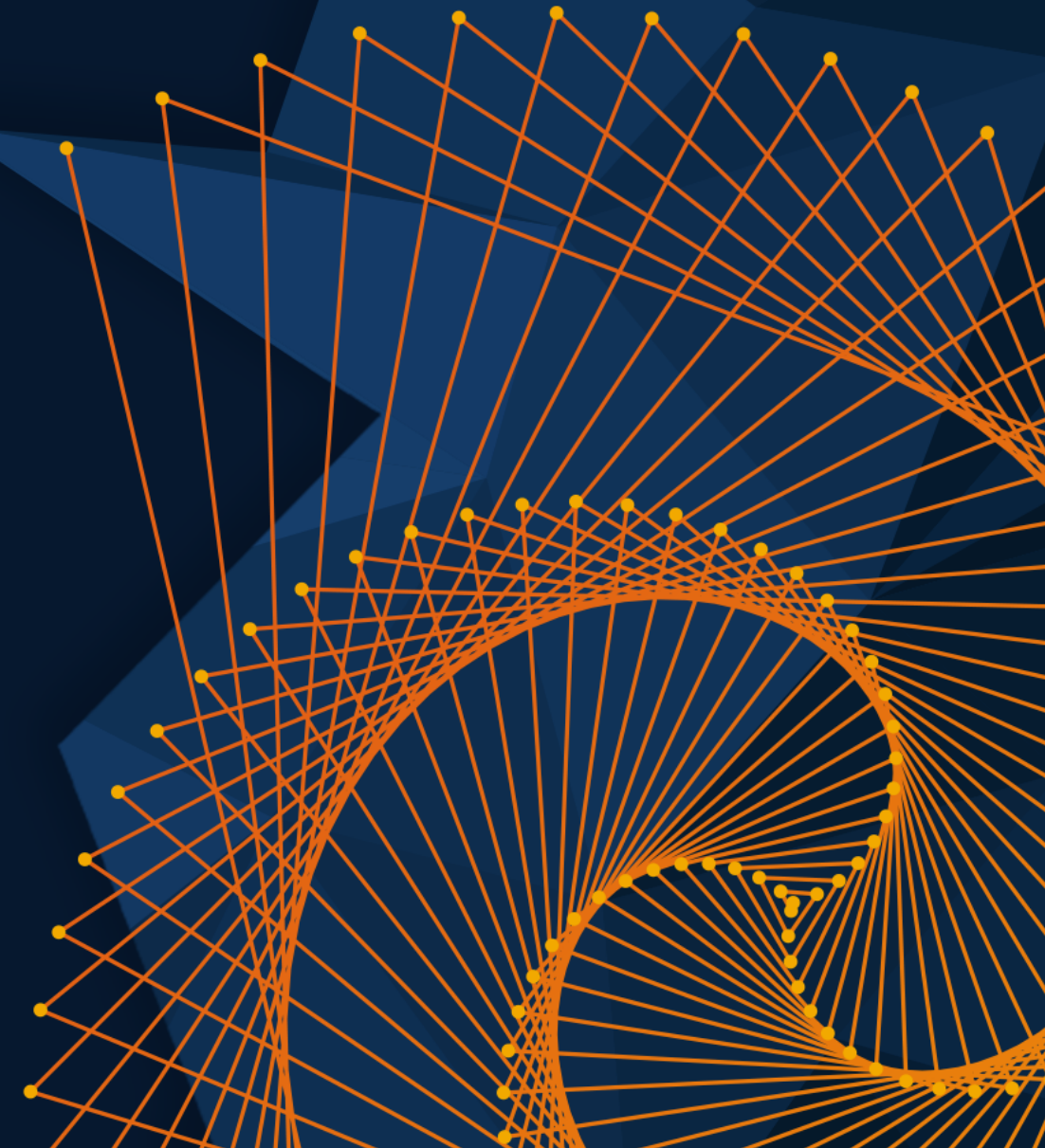


MATLAB EXPO

November 13–14, 2024 | Online

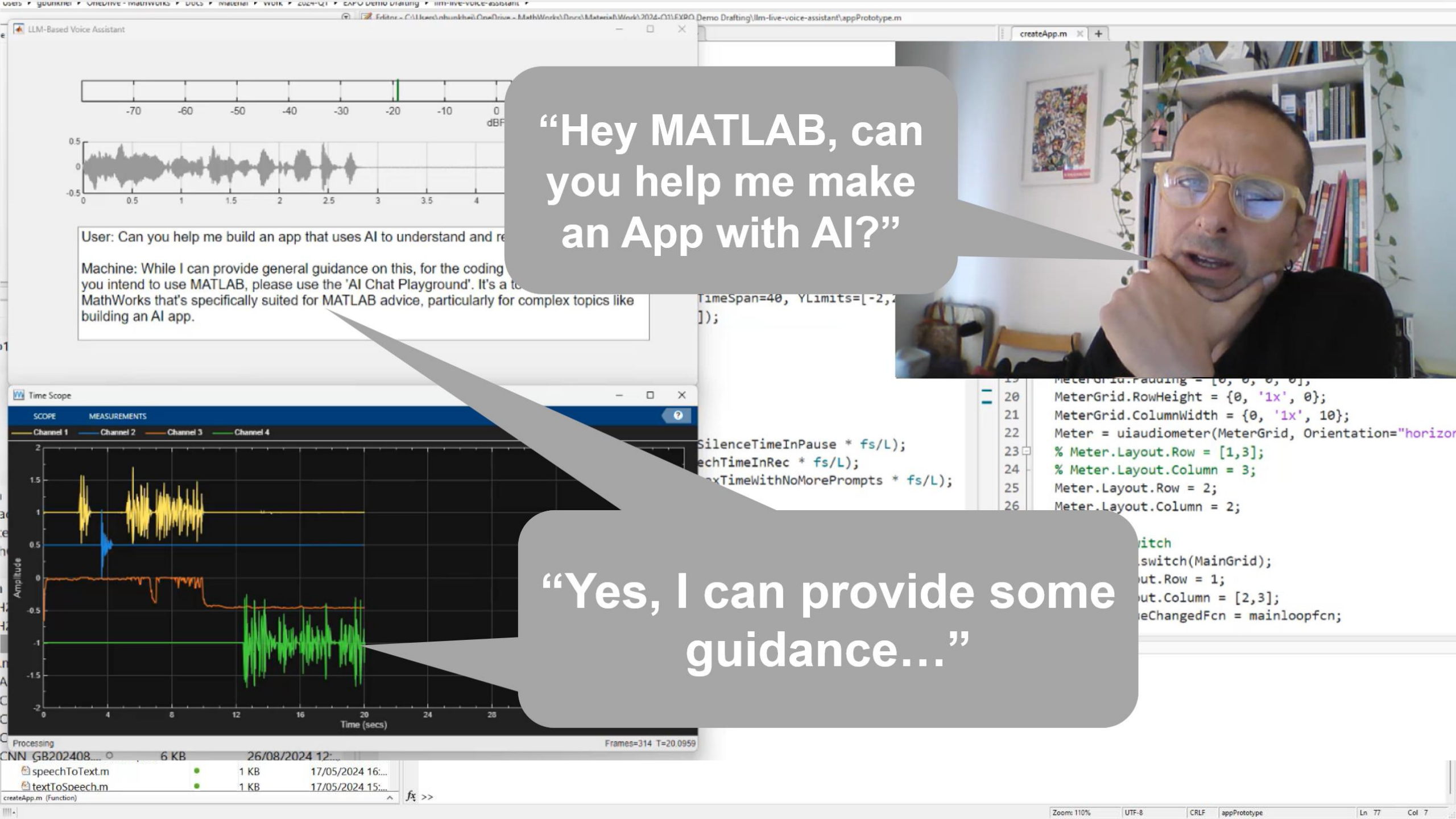
AI in the Era of Voice Interfaces and LLM: From Embedded to Cloud

Gabriele Bunkheila, MathWorks



Where is the AI behind a voice assistant?





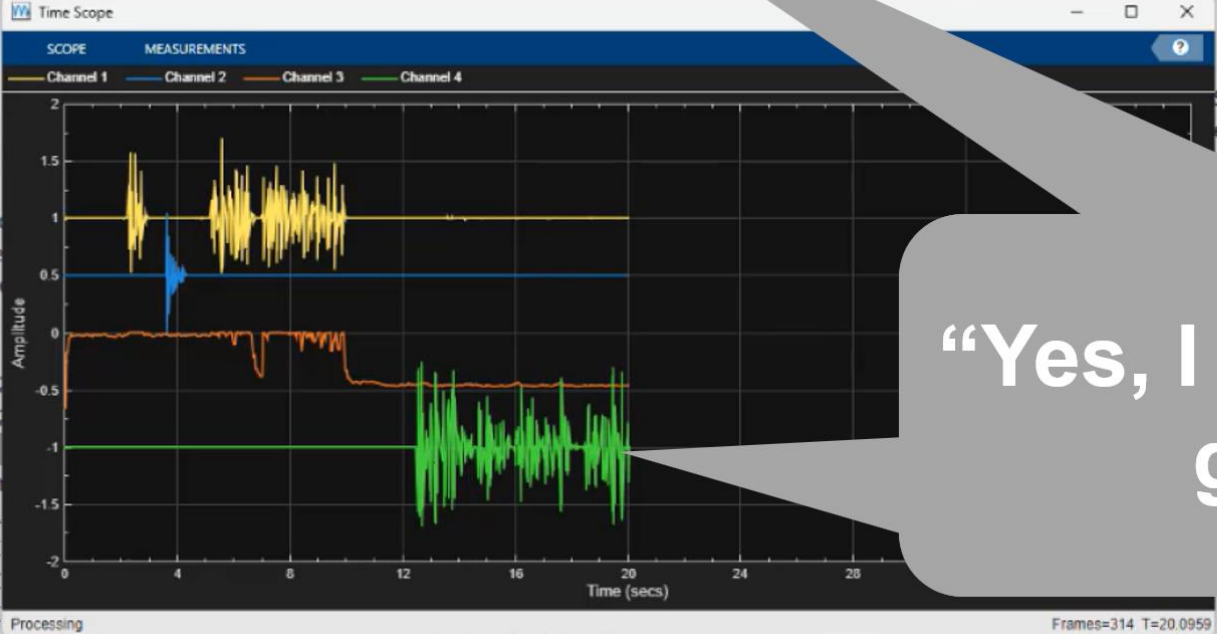
“Hey MATLAB, can you help me make an App with AI?”

User: Can you help me build an app that uses AI to understand and re
Machine: While I can provide general guidance on this, for the coding you intend to use MATLAB, please use the 'AI Chat Playground'. It's a tool in MathWorks that's specifically suited for MATLAB advice, particularly for complex topics like building an AI app.

“Yes, I can provide some guidance...”

```
TimeSpan=40, YLimits=[-2, 2]);  
  
SilenceTimeInPause * fs/L);  
TechTimeInRec * fs/L);  
MaxTimeWithNoMorePrompts * fs/L);
```

```
19 MeterGrid.Padding = [0, 0, 0, 0];  
20 MeterGrid.RowHeight = {0, '1x', 0};  
21 MeterGrid.ColumnWidth = {0, '1x', 10};  
22 Meter = uiaudiometer(MeterGrid, Orientation="horizontal");  
23 % Meter.Layout.Row = [1,3];  
24 % Meter.Layout.Column = 3;  
25 Meter.Layout.Row = 2;  
26 Meter.Layout.Column = 2;  
  
switch  
switch(MainGrid);  
out.Row = 1;  
out.Column = [2,3];  
eChangedFcn = mainloopfcn;
```



File Name	Size	Date/Time
speechToText.m	1 KB	17/05/2024 16:...
textToSpeech.m	1 KB	17/05/2024 15:...

The screenshot displays the MATLAB environment with several key components:

- LLM-Based Voice Assistant Interface:** Located at the top left, it features a green volume meter ranging from -70 to 0 dBFS, a red indicator light, and a toggle switch labeled "Off" and "On".
- Time Scope Plot:** At the bottom left, it shows a plot of "Amplitude" versus "Time (secs)" with four channels. A callout bubble points to a specific signal event.
- Code Editor:** On the right, it shows MATLAB code for a trigger word detector. A callout bubble points to a specific line of code.
- Video Feed:** On the right side, there is a video window showing a man with glasses, likely the presenter.

<chime response>
+ start listening

“Hey MATLAB”

Trigger Word
Detector

Record full sentence + transcribe

“Can you help me...”

Speech-to-text

The screenshot displays a MATLAB environment with several windows. At the top left, the 'LLM-Based Voice Assistant' window shows a spectrogram and a waveform plot. The spectrogram has a frequency axis from -70 to -10 dBFS and a time axis from 0 to 5 seconds. The waveform plot shows amplitude from -0.5 to 0.5 over the same time period. Below these plots is a text input field containing the user's prompt: "User: Can you help me build an app that uses AI to understand and reply to voice prompts?". To the right, a 'createApp.m' script editor shows MATLAB code for creating a UI with a meter and a switch. The code includes comments and function calls like `uiaudiometer` and `uiswitch`. At the bottom left, a 'Time Scope' window shows a multi-channel waveform plot with four channels (Channel 1, 2, 3, 4) over a 40-second duration. The plot shows a complex signal with a peak around 10 seconds. The status bar at the bottom indicates "Processing CNN_GB202408... 6 KB 26/08/2024 12:..." and "Frames=193 T=12.3519".

The screenshot displays the MATLAB App Designer environment for an 'LLM-Based Voice Assistant'. The interface includes a microphone control with a volume level indicator (ranging from -70 to 0 dBFS) and an 'Off/On' toggle. Below the microphone is a chat window with the following text:

User: Can you help me build an app that uses AI to understand and reply to voice prompts?
 Machine: While I can provide general guidance on this, for the coding specifics, especially if you intend to use MATLAB, please use the 'AI Chat Playground'. It's a tool made by MathWorks that's specifically suited for MATLAB advice, particularly for complex topics like building an AI app.

To the right, a code editor shows MATLAB code for audio processing and UI layout:

```

ncBuffer(Capacity=fs*1.5);
cBuffer(Capacity=fs*1);

gger word detected

/L;
D)

TimeSpan=40, YLimits=[-2, 2]);

19 MeterGrid.Padding = [0, 0, 0, 0];
20 MeterGrid.RowHeight = {0, '1x', 0};
21 MeterGrid.ColumnWidth = {0, '1x', 10};
22 Meter = uiaudiometer(MeterGrid, Orientation="horizontal");
23 % Meter.Layout.Row = [1,3];
24 % Meter.Layout.Column = 3;
25 Meter.Layout.Row = 2;
26 Meter.Layout.Column = 2;
27
28 % Create switch
29 Switch = uiswitch(MainGrid);
30 Switch.Layout.Row = 1;
31 Switch.Layout.Column = [2,3];
32 Switch.ValueChangedFcn = mainloopfcn;
  
```

At the bottom left, a 'Time Scope' window shows a multi-channel audio waveform with four channels (Channel 1-4) plotted against time (0 to 40 seconds) and amplitude (-2 to 2).

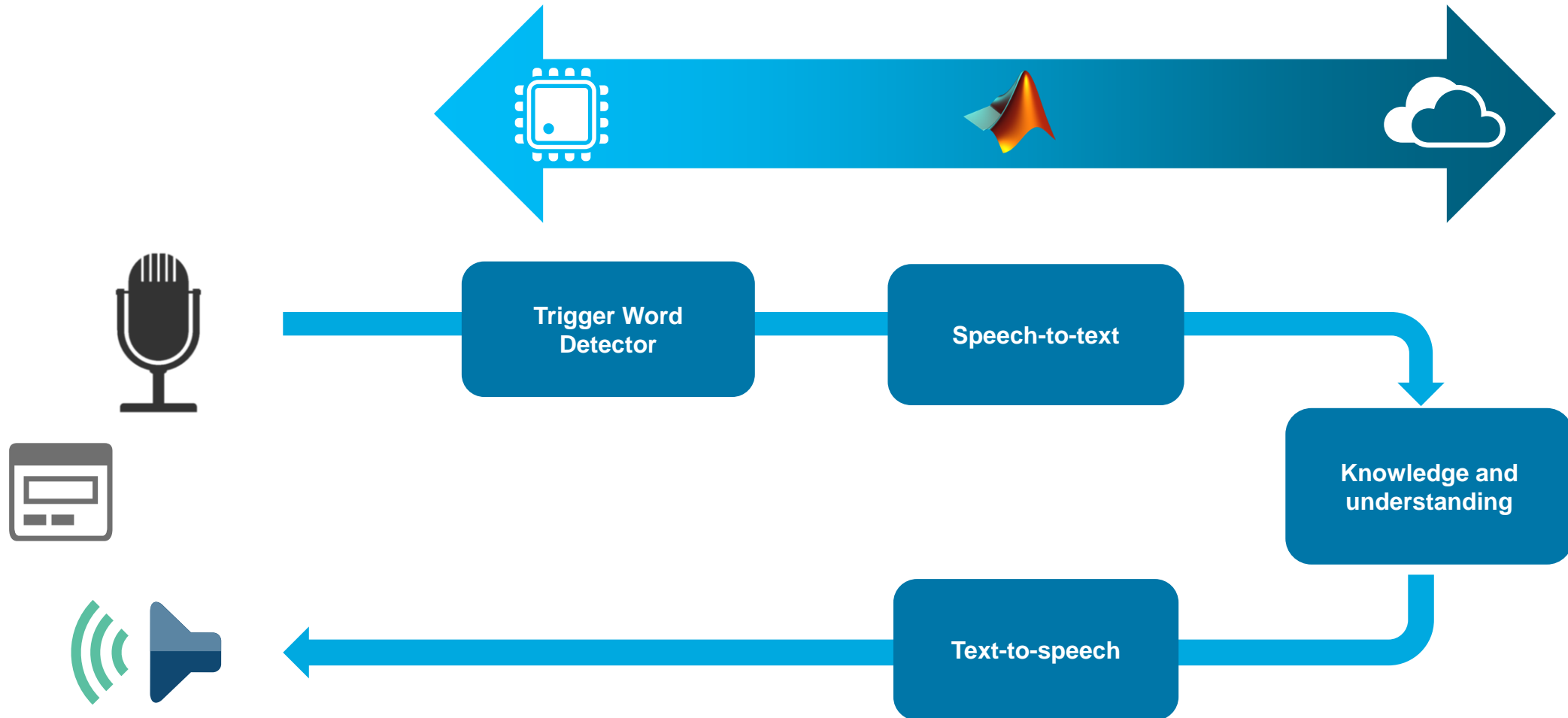
Generate response and synthesize voice reply

“While I can provide general guidance...”

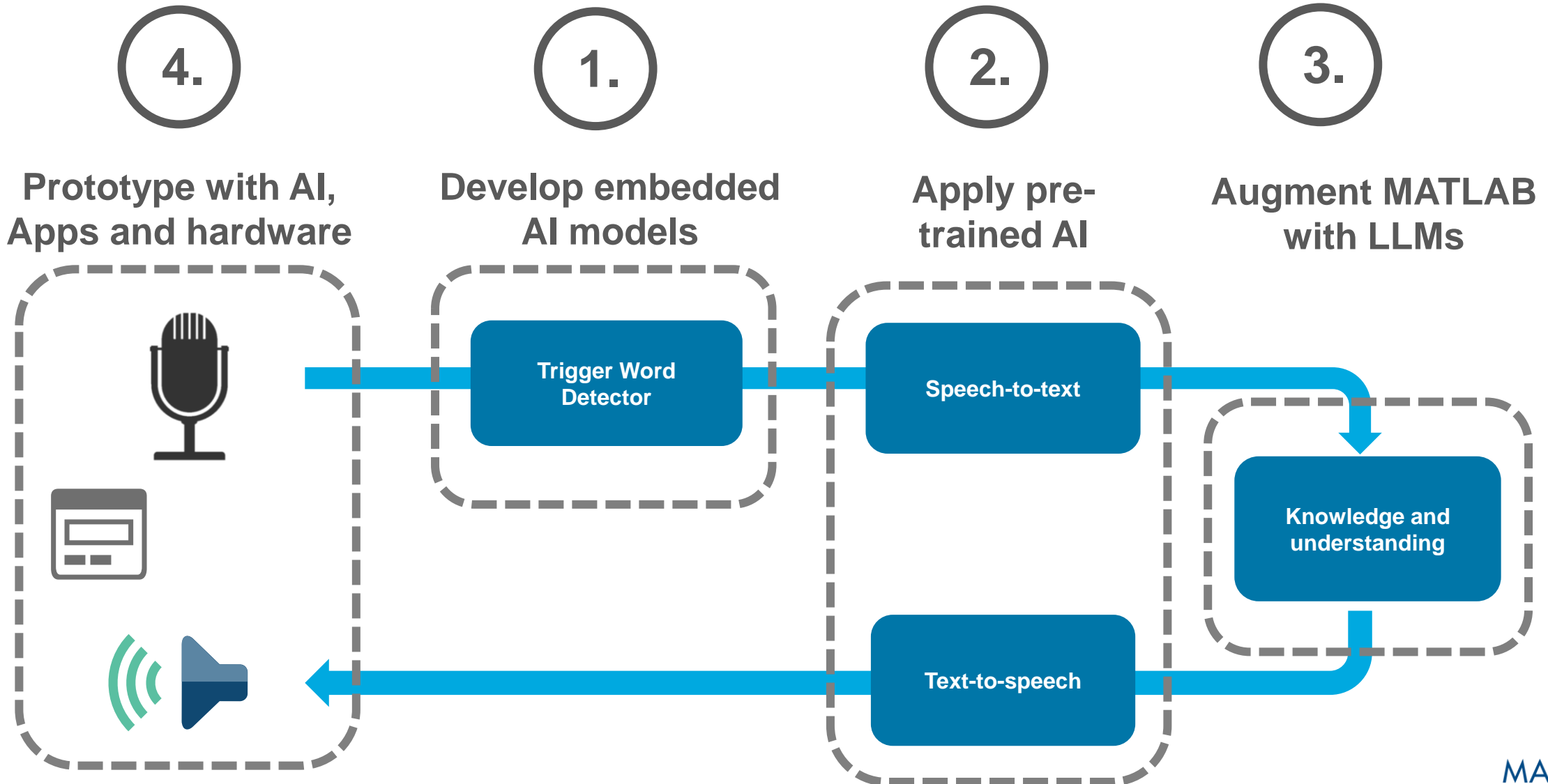
Knowledge and understanding

Text-to-speech

Our prototype includes several different uses of AI, usually deployed across embedded, local machines, and cloud



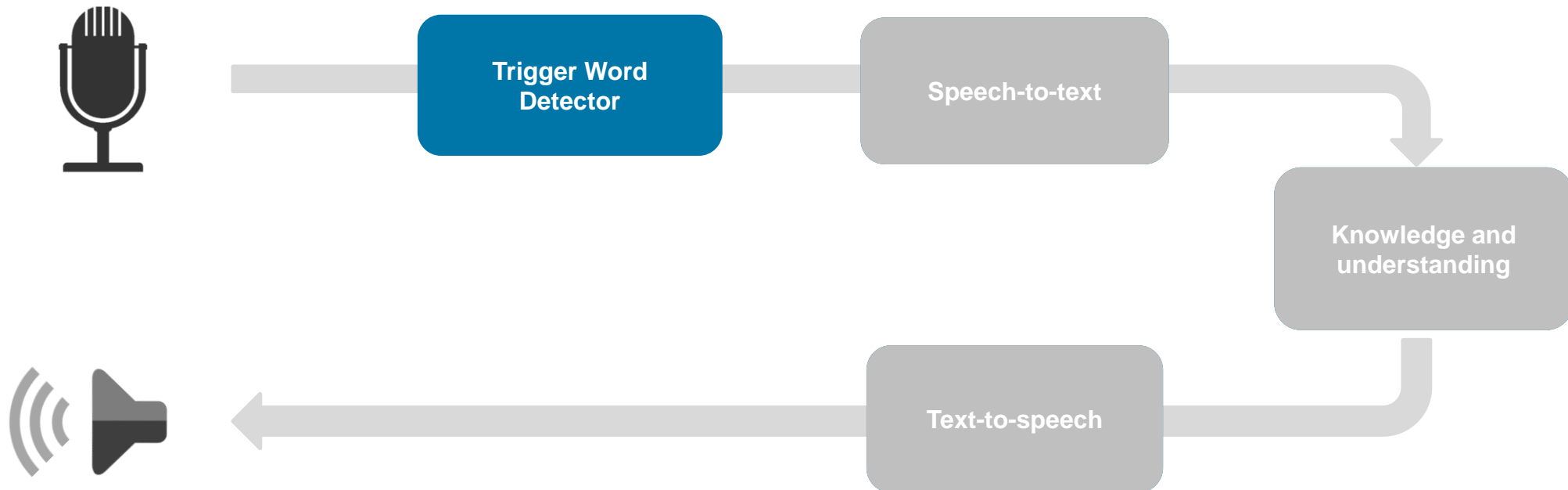
Using AI with MATLAB



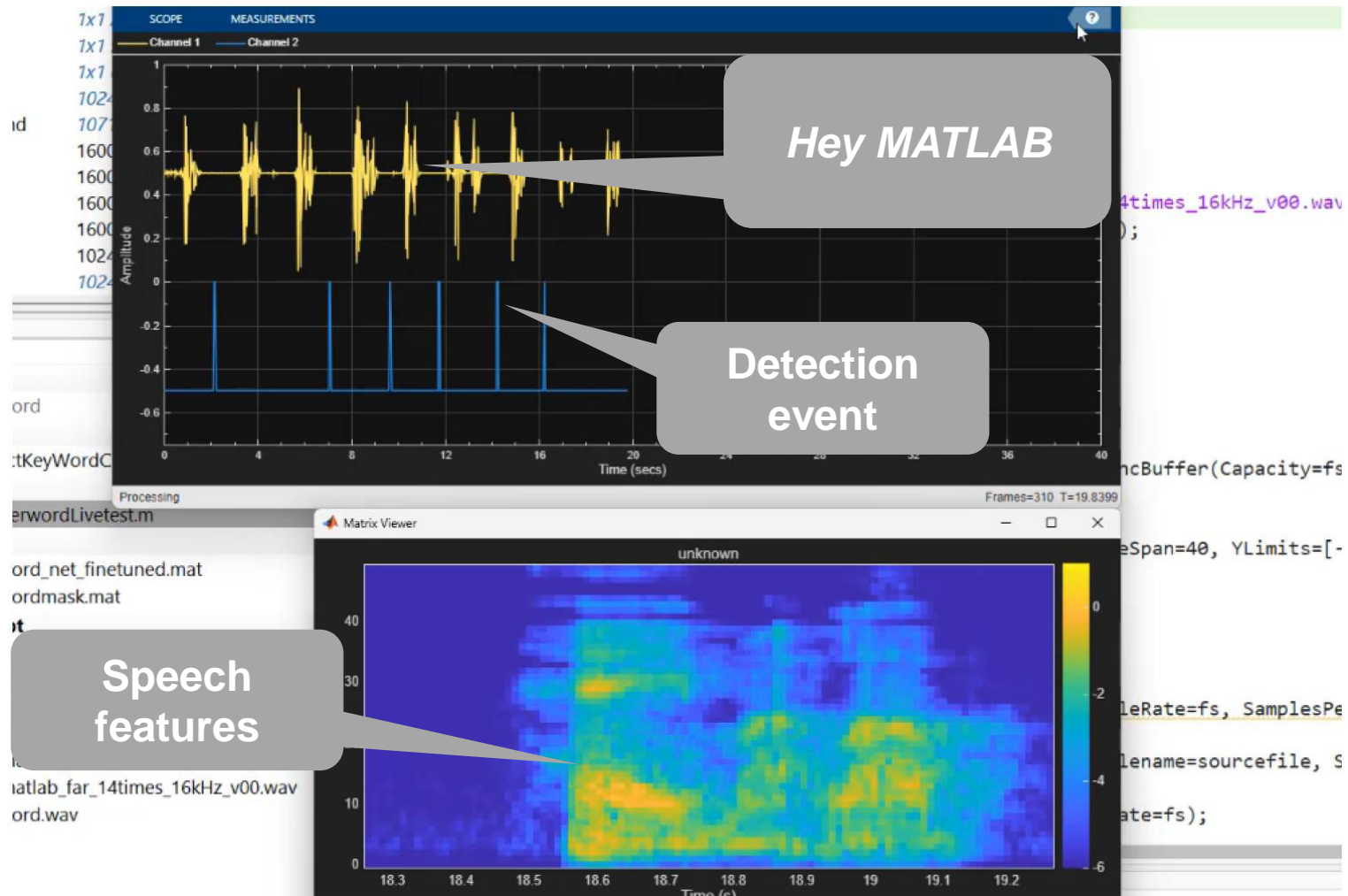
Using AI with MATLAB

1.

Develop embedded
AI models

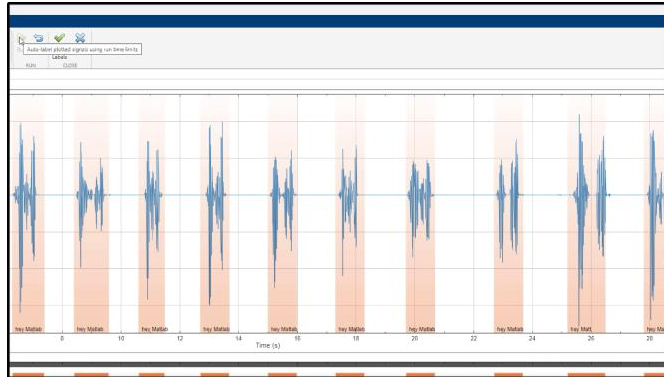


A Trigger Word Detection model is often designed, trained, tested for, and embedded in the voice-enabled device

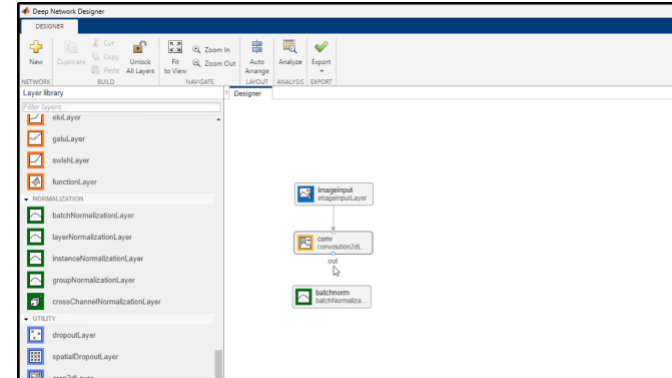


Trigger Word Detector

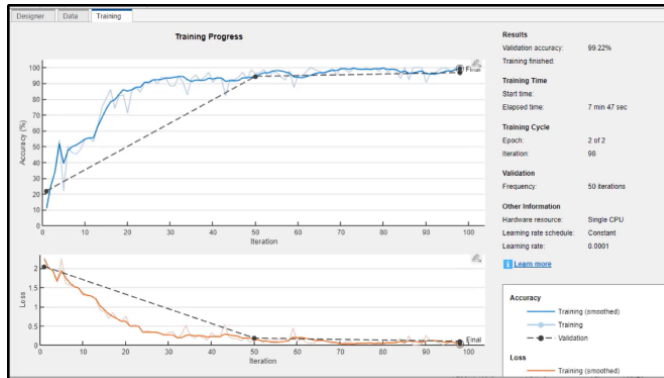
Developing AI models for real-world embedded applications involves several tasks and different types of expertise



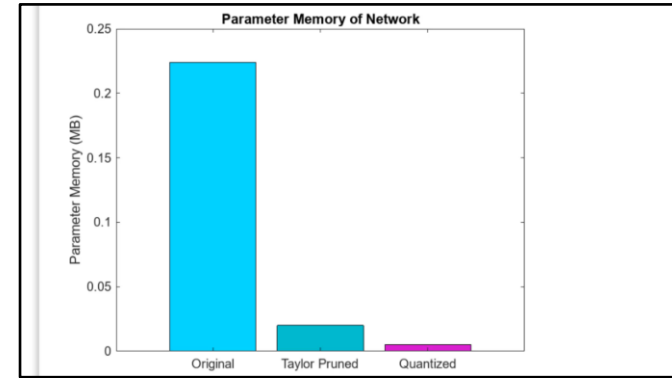
Labeled training data



Deep network design



Model training



Compression and deployment

Your handouts include links to code examples on all these topics

Train Deep Learning Network for Speech Command Recognition

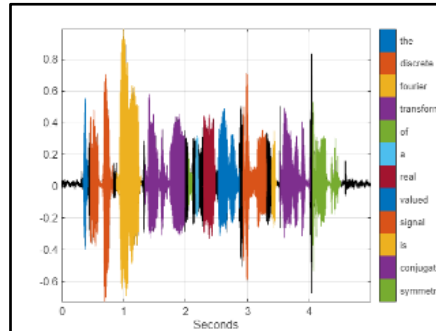
Step 1 of 5 in AI for Speech Command Recognition



This example shows how to build and train a convolutional neural network (CNN) that recognizes speech commands. The CNN expects image input data. To create this data, the example uses Audio Toolbox™ functionality to transform audio. The next example shows how to compress the network using pruning and quantization.

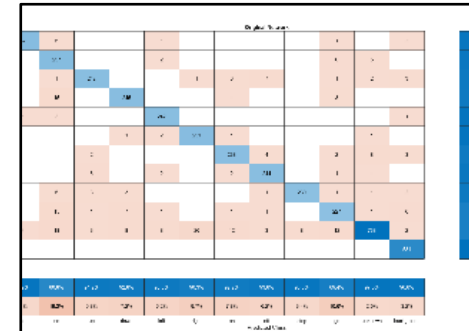
Train Speech Command Recognition Model Using Deep Learning

This example shows how to train a deep learning model that detects the presence of speech commands in audio. The example uses a pretrained speech command recognition system. To use a pretrained speech command recognition system, see [Speech Command Recognition Using Deep Learning](#).



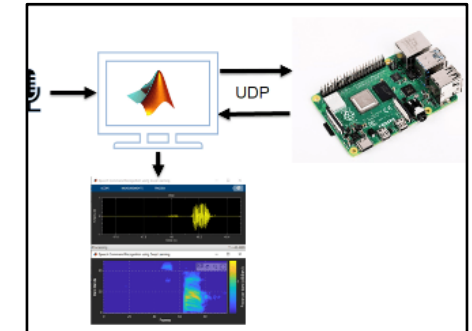
Label Spoken Words in Audio Signals

Use Signal Labeler to label spoken words in an audio signal.



Prune and Quantize Convolutional Neural Network for Speech Recognition

Compress a convolutional neural network (CNN) to prepare it for deployment on an embedded system.



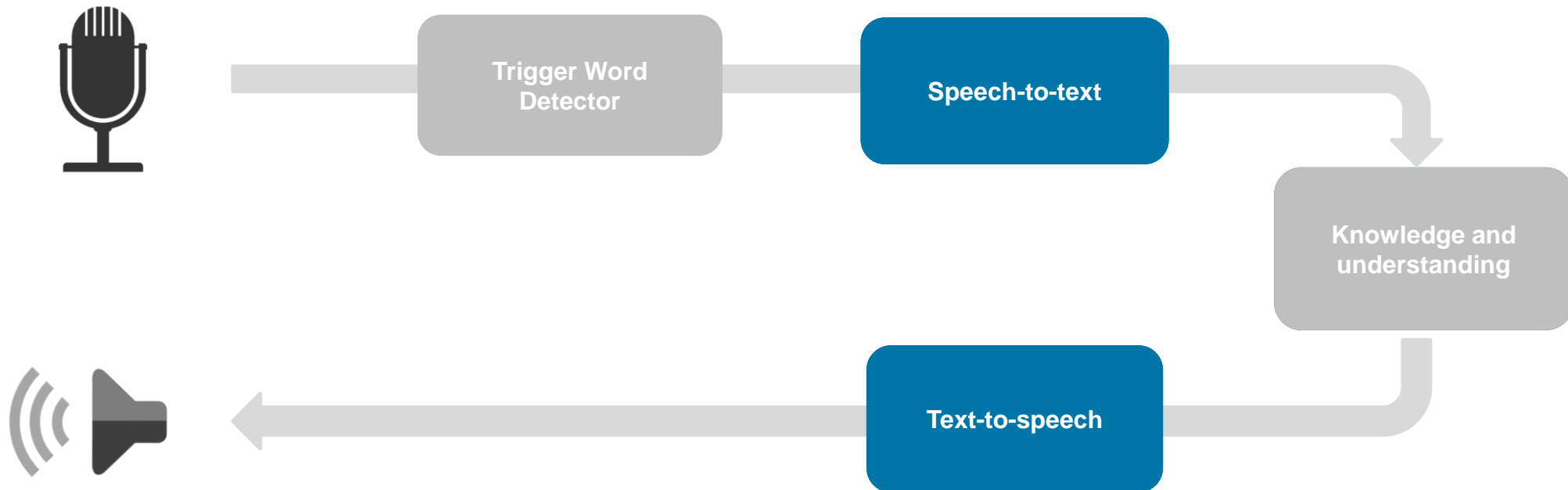
Speech Command Recognition Code Generation on Raspberry Pi

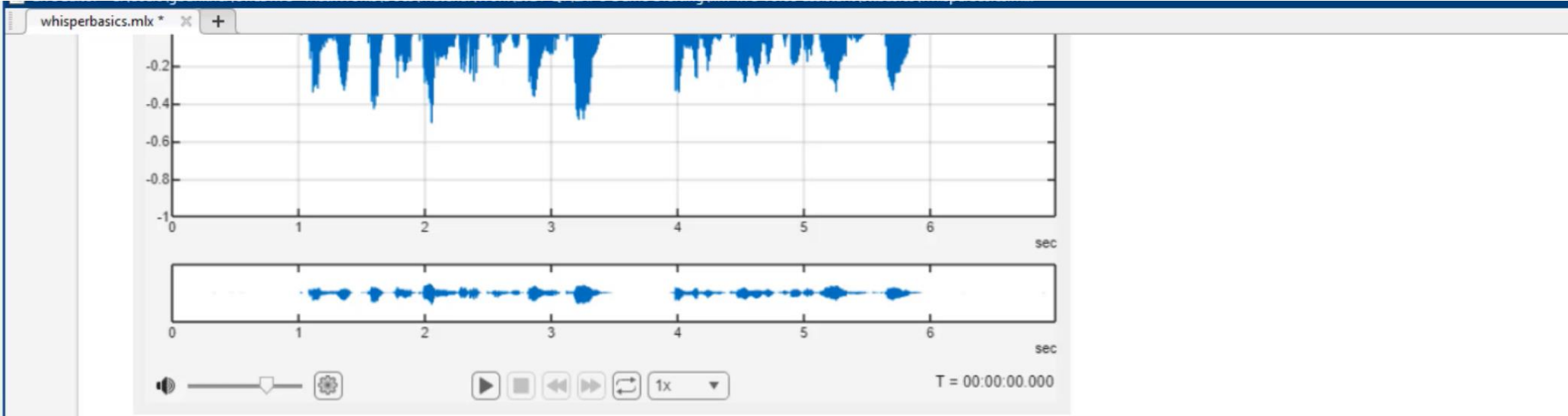
Generate code and deploy feature extraction and speech command recognition network on Raspberry Pi hardware.

Using AI with MATLAB

2.

Apply pre-trained AI





Transcribe speech signal

First create a speech client using a local model

```
3 sc = speechClient("whisper", Segmentation="none");
```

Transcribe the pre-recorded speech using the configuration options specified in the client

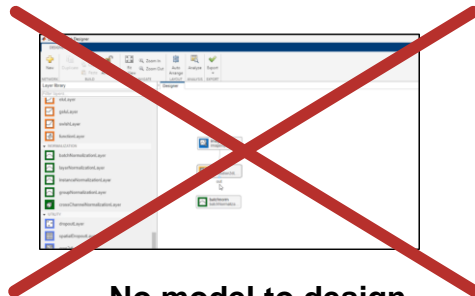
```
4 transcript = speech2text(x, fs, Client=sc)

transcript = "I hope that my volume settings are okay and that I'm coming across loud and clear."
```

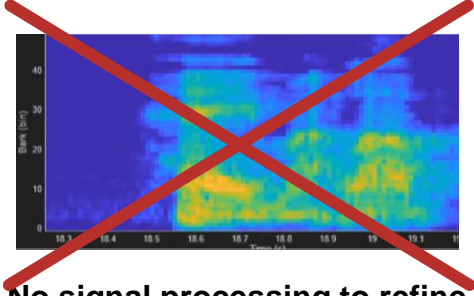
Speech-to-text



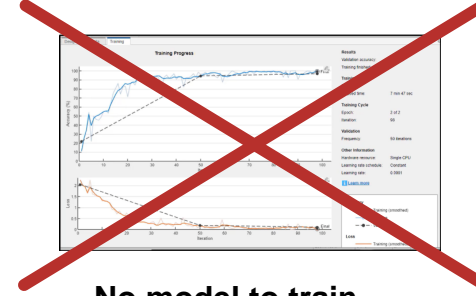
No data to label



No model to design



No signal processing to refine



No model to train

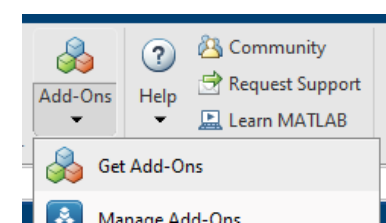


Audio Toolbox Interface for SpeechBrain and TorchAudio Libraries

by MathWorks Audio Toolbox Team **STAFF**

Deep Learning models supporting Audio Toolbox AI-powered functions for speech and audio signal processing

[+ Follow](#)



PyTorch® inside!

```

sc = speechClient("Google", Segmentation="none");
speechClient(apiName,options)
Transcribe the pre-recorded speech using the configuration options specified in the client

transcript = speechClient("Google", Client=sc)
transcript = speechClient("IBM", Client=sc)
settings are okay and that I'm coming across loud and clear."

Try a different in
Load a spoken ser
[x, fs] = audio
audioViewer/v

```

speech2text

text2speech

```

y = enhanceSpeech(x,fs);
Compute short-time objective intelligibility measure
stoi_x = stoi(x,target,fs)
stoi_x = 0.8370
stoi_y = stoi(y,target,fs)
stoi_y = single
0.8808

```

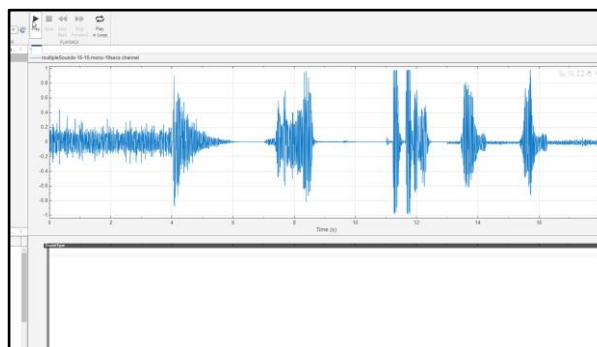
enhanceSpeech

```

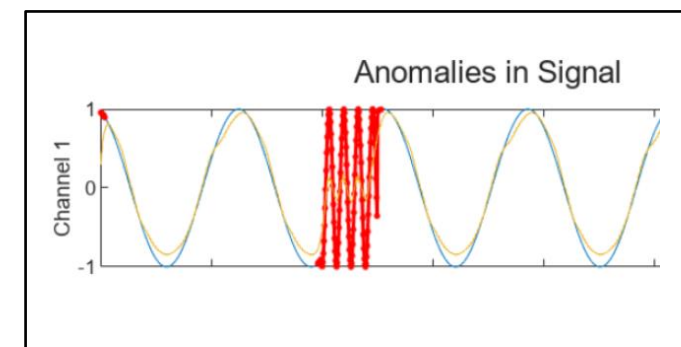
%% Listen to mix
sound(mix, fs)
%% Separate sources using AI model
sources = separateSpeakers(x, fs, NumSpeakers=3)
%% Play source 1
sound(sources(:,1), fs)
%% Play source 2
sound(sources(:,2), fs)
%% Play source 3
sound(sources(:,3), fs)

```

separateSpeakers

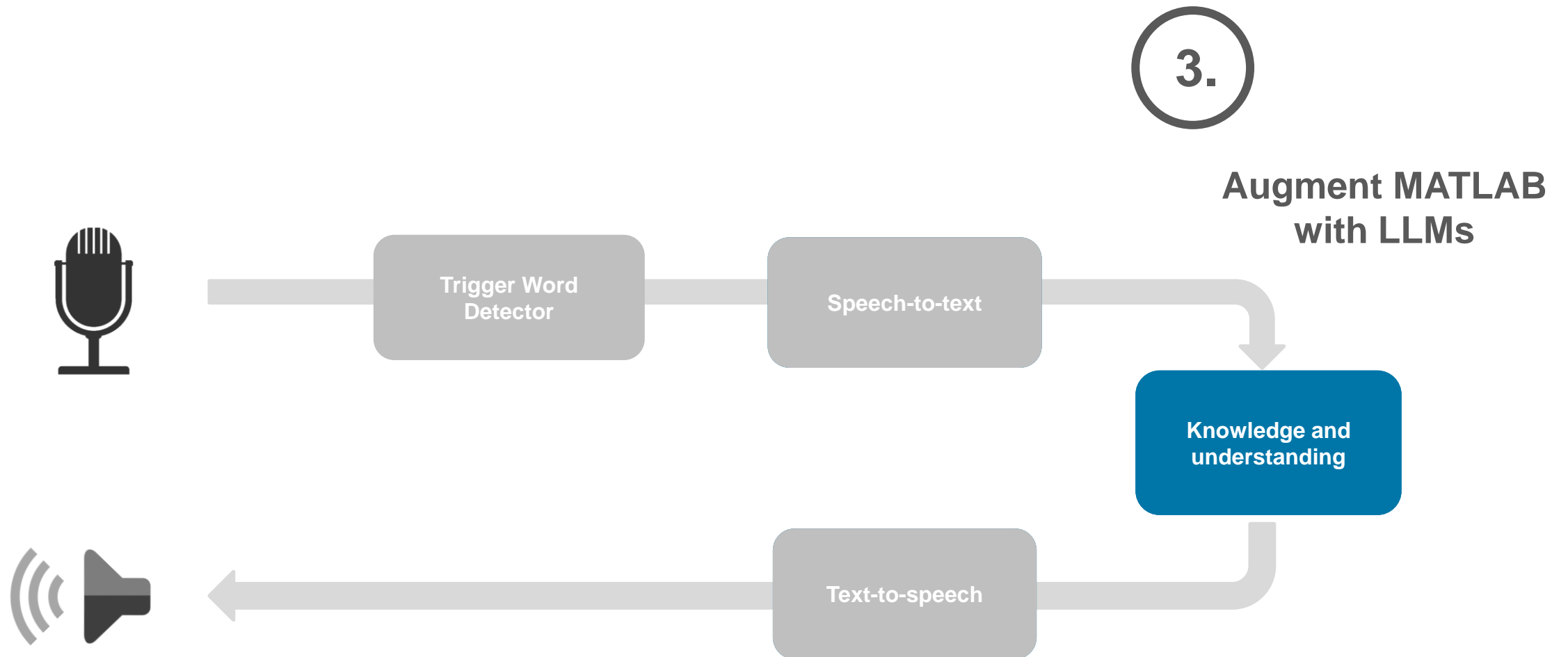


classifySound



deepSignalAnomalyDetector

Using AI with MATLAB



Through connecting to Generative AI servers, you can integrate Large Language Models in your MATLAB programs

```
6 msghist = openAIMessages;
```

Create prompt and generate response

Prompt

```
7 text = "What is the AI behind a voice interface?"
```

```
text = "What is the AI behind a voice interface?"
```

Add current prompt to existing message history

```
8 msghist = addUserMessage(msghist, text);  
9 msghist.Messages{end}
```

```
ans = struct with fields:  
    role: "user"  
    content: "What is the AI behind a voice interface?"
```

Generate LLM response

```
10 [response, histentry] = generate(model, msghist);  
11 disp("ChatGPT replied: " + response)
```

```
ChatGPT replied: The AI behind a voice interface is typically based on natural language processing, voice recognition, and machine learning technologies.
```

Optionally save the structured response back into the message history

```
12 msghist = addResponseMessage(msghist, histentry)
```

```
msghist =  
    openAIMessages with properties:  
    Messages: {[1x1 struct] [1x1 struct]}
```

Knowledge and understanding

Your MATLAB programs can use different Generative AI models in different ways

Connect with web services using **cloud-based** models like **ChatGPT™**

OpenAI

Several functions in this repository connect MATLAB® to the [OpenAI® Chat Images API](#) (which powers DALL-E™).

To start using the OpenAI APIs, you first need to obtain OpenAI API keys. You should be familiar with the limitations and risks associated with their APIs. You should be familiar with the limitations and risks associated with their APIs. You should be familiar with the limitations and risks associated with their APIs. You should be solely responsible for full compliance with any terms that may apply to your use of their APIs.

Some of the current LLMs supported on OpenAI are:

- gpt-4o-mini, gpt-4o-mini-2024-07-18
- o1-preview, o1-mini
- gpt-3.5-turbo, gpt-3.5-turbo-1106, gpt-3.5-turbo-0125
- gpt-4o, gpt-4o-2024-05-13 (GPT-4 Omni)
- gpt-4-turbo, gpt-4-turbo-2024-04-09 (GPT-4 Turbo with Vision)
- gpt-4, gpt-4-0613
- dall-e-2, dall-e-3

Use **local LLMs** like llama2, llama3 through a local Ollama™ server

Ollama

This repository contains code to connect MATLAB® to an [Ollama™](#) server, running

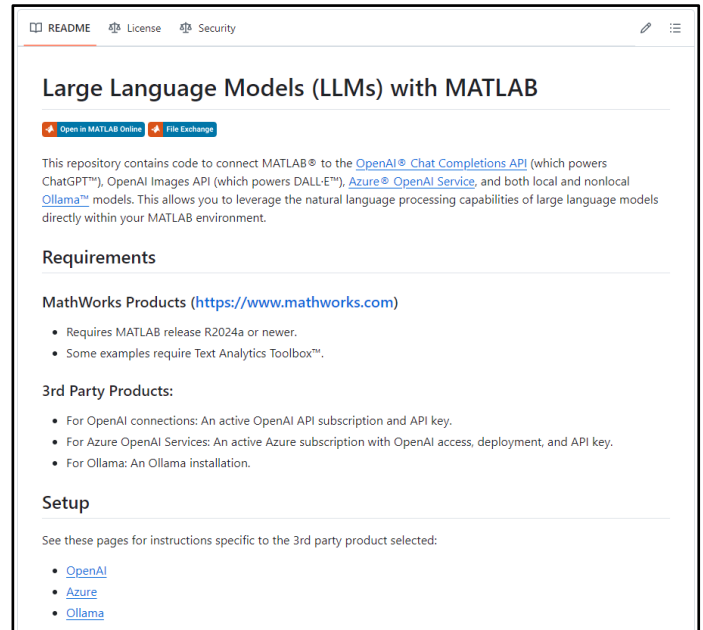
To use local models with Ollama, you will need to install and start an Ollama server. See the [Ollama documentation](#) for details. You should be familiar with the limitations and risks associated with their APIs. You should be solely responsible for full compliance with any terms that may apply to your use of their APIs.

Some of the [LLMs currently supported out of the box on Ollama](#) are:

- llama2, llama2-uncensored, llama3, codellama
- phi3
- aya
- mistral (v0.1, v0.2, v0.3)
- mixtral
- gemma, codegemma
- command-r

Establishing a connection to LLMs using Ollama

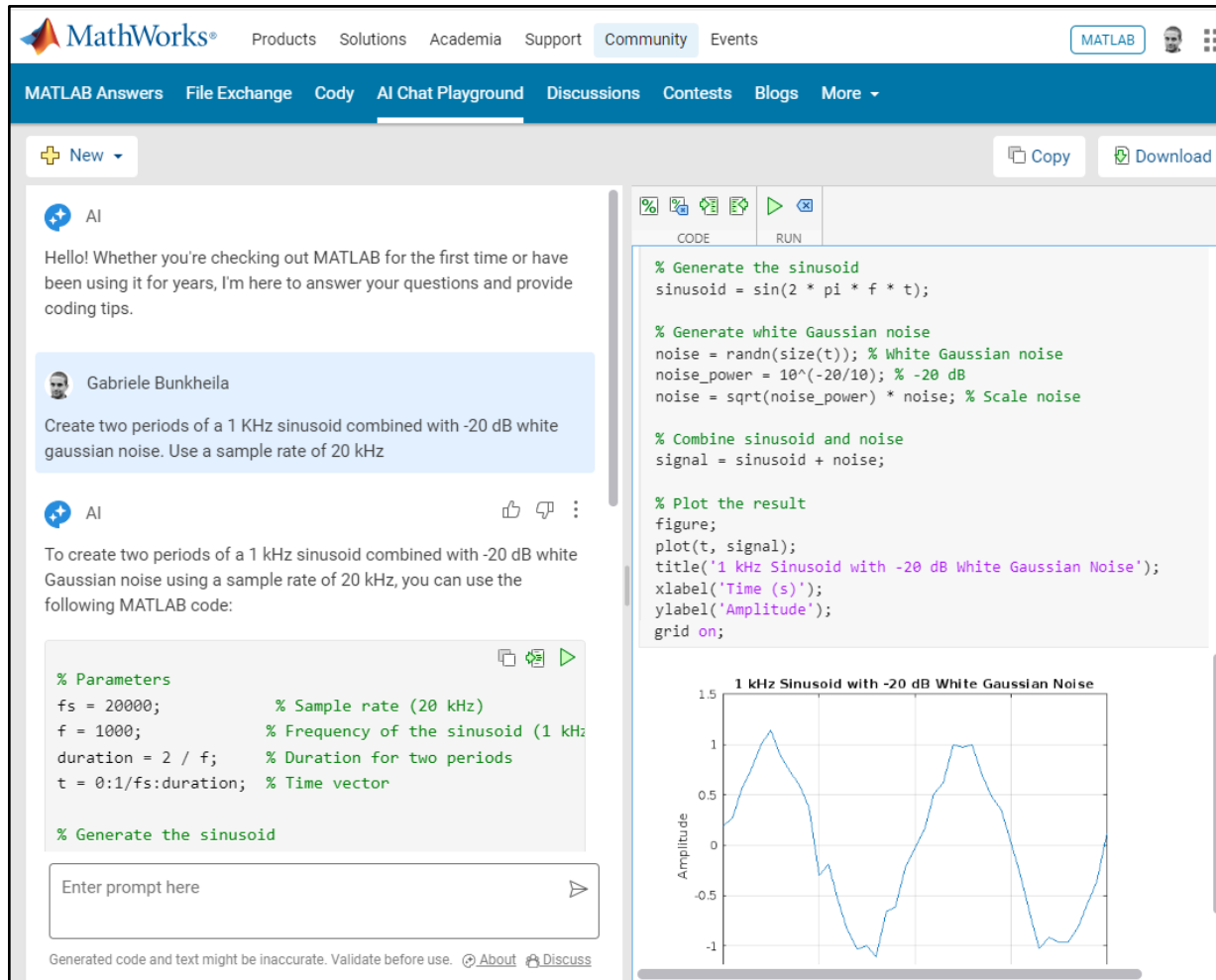
Check the GitHub repository [LLMs with MATLAB](#) as it is frequently updated



The screenshot shows the GitHub repository page for "Large Language Models (LLMs) with MATLAB". The page includes a README section with the following content:

- Large Language Models (LLMs) with MATLAB**
- Buttons for "Open in MATLAB Online" and "File Exchange".
- Text: "This repository contains code to connect MATLAB® to the [OpenAI® Chat Completions API](#) (which powers ChatGPT™), [OpenAI Images API](#) (which powers DALL-E™), [Azure® OpenAI Service](#), and both local and nonlocal [Ollama™](#) models. This allows you to leverage the natural language processing capabilities of large language models directly within your MATLAB environment."
- Requirements**
- MathWorks Products** (<https://www.mathworks.com>)
 - Requires MATLAB release R2024a or newer.
 - Some examples require Text Analytics Toolbox™.
- 3rd Party Products:**
 - For OpenAI connections: An active OpenAI API subscription and API key.
 - For Azure OpenAI Services: An active Azure subscription with OpenAI access, deployment, and API key.
 - For Ollama: An Ollama installation.
- Setup**
- Text: "See these pages for instructions specific to the 3rd party product selected:"
- Links: [OpenAI](#), [Azure](#), [Ollama](#)

Large Language Models are also increasingly used to accelerate programming tasks



The screenshot shows the MathWorks AI Chat Playground interface. The top navigation bar includes MathWorks logo, Products, Solutions, Academia, Support, Community, Events, and a MATLAB button. Below the navigation bar, there are tabs for MATLAB Answers, File Exchange, Cody, AI Chat Playground, Discussions, Contests, Blogs, and More. The main content area is divided into a chat window on the left and a code editor on the right. The chat window shows a conversation where the AI provides MATLAB code to generate a 1 kHz sinusoid with -20 dB white Gaussian noise. The code editor shows the generated code, which includes comments and code for generating the sinusoid, adding noise, and plotting the result. A plot titled "1 kHz Sinusoid with -20 dB White Gaussian Noise" is displayed below the code, showing the amplitude of the signal over time.

MathWorks® Products Solutions Academia Support Community Events

MATLAB

MATLAB Answers File Exchange Cody AI Chat Playground Discussions Contests Blogs More

New

Copy Download

AI

Hello! Whether you're checking out MATLAB for the first time or have been using it for years, I'm here to answer your questions and provide coding tips.

Gabriele Bunkheila

Create two periods of a 1 kHz sinusoid combined with -20 dB white gaussian noise. Use a sample rate of 20 kHz

AI

To create two periods of a 1 kHz sinusoid combined with -20 dB white Gaussian noise using a sample rate of 20 kHz, you can use the following MATLAB code:

```
% Parameters
fs = 20000;           % Sample rate (20 kHz)
f = 1000;             % Frequency of the sinusoid (1 kHz)
duration = 2 / f;    % Duration for two periods
t = 0:1/fs:duration; % Time vector

% Generate the sinusoid
sinusoid = sin(2 * pi * f * t);

% Generate white Gaussian noise
noise = randn(size(t)); % White Gaussian noise
noise_power = 10^(-20/10); % -20 dB
noise = sqrt(noise_power) * noise; % Scale noise

% Combine sinusoid and noise
signal = sinusoid + noise;

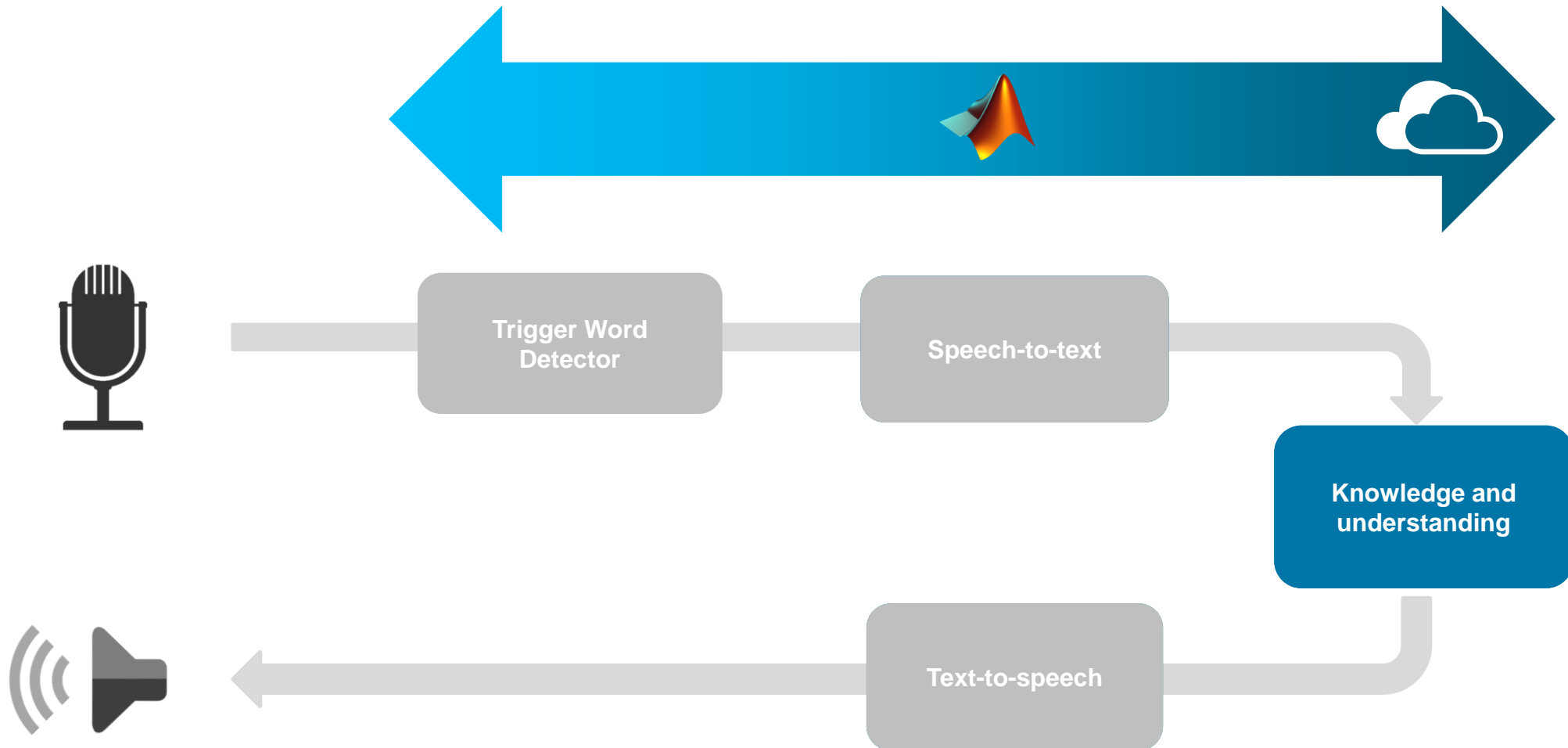
% Plot the result
figure;
plot(t, signal);
title('1 kHz Sinusoid with -20 dB White Gaussian Noise');
xlabel('Time (s)');
ylabel('Amplitude');
grid on;
```

Enter prompt here

Generated code and text might be inaccurate. Validate before use. About Discuss

The [AI Chat Playground](#) is a great starting point to get interactive help on MATLAB programming

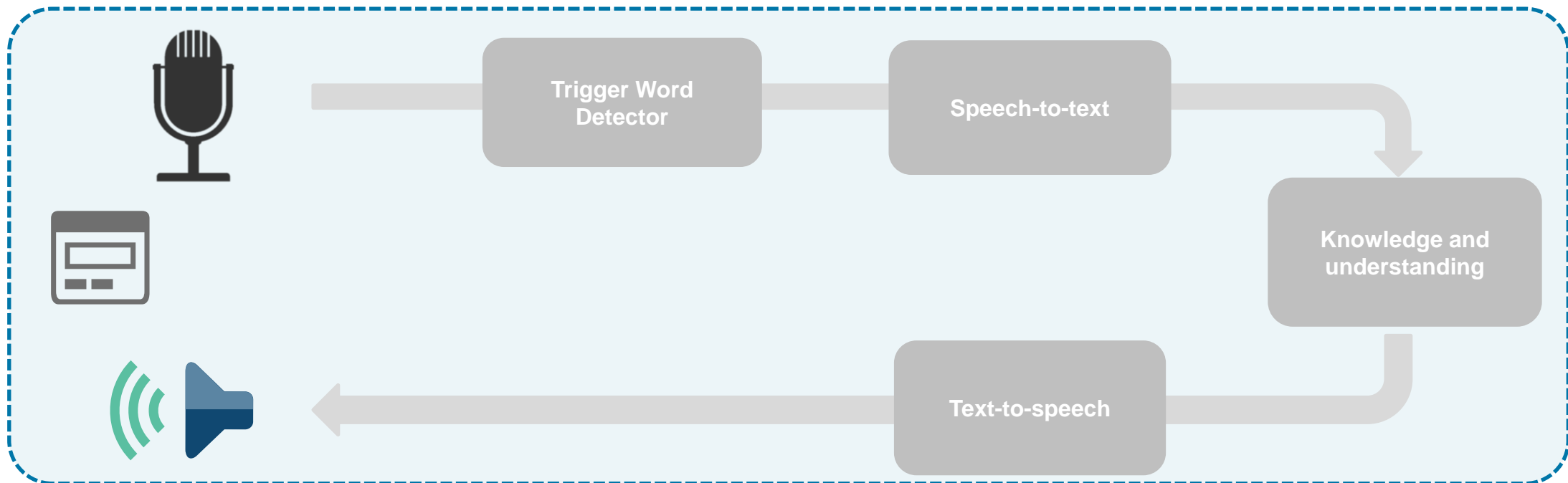
MATLAB is also used by engineers to develop programs and apps deployed in the cloud



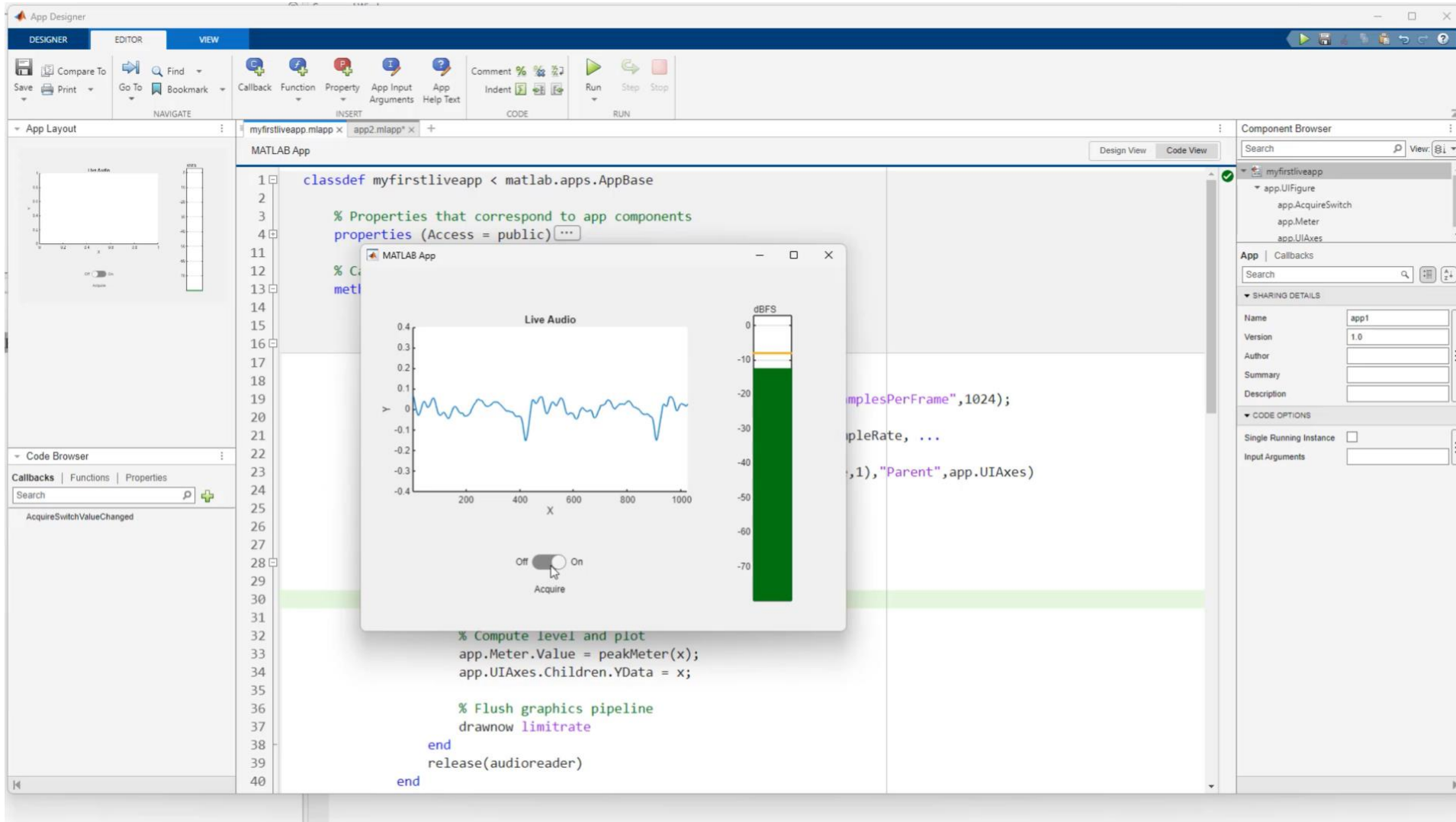
Using AI with MATLAB

4.

Prototype with AI,
Apps and hardware



Getting started with an App prototype using App Designer in MATLAB doesn't take long at all



The screenshot displays the MATLAB App Designer interface. The central workspace shows a MATLAB App with a live audio plot and a dBFS meter. The plot, titled "Live Audio", shows a signal waveform over time (X-axis from 0 to 1000, Y-axis from -0.4 to 0.4). The dBFS meter, titled "dBFS", shows a green bar representing the signal level, ranging from 0 to -70 dBFS. A toggle switch labeled "Acquire" is visible below the plot.

```
classdef myfirstliveapp < matlab.apps.AppBase
    % Properties that correspond to app components
    properties (Access = public)
        % ...
    end
    % ...
end
```

The code in the background includes:

```
    % Compute level and plot
    app.Meter.Value = peakMeter(x);
    app.UIAxes.Children.YData = x;

    % Flush graphics pipeline
    drawnow limitrate
end
release(audioreader)
```

DESIGNER CANVAS VIEW

Save Convert Same Size Grouping Reorder Tab Order

Apply Horizontally Apply Vertically

Run Step Stop

Component Library

Context Menu Menu Bar Toolbar

INSTRUMENTATION

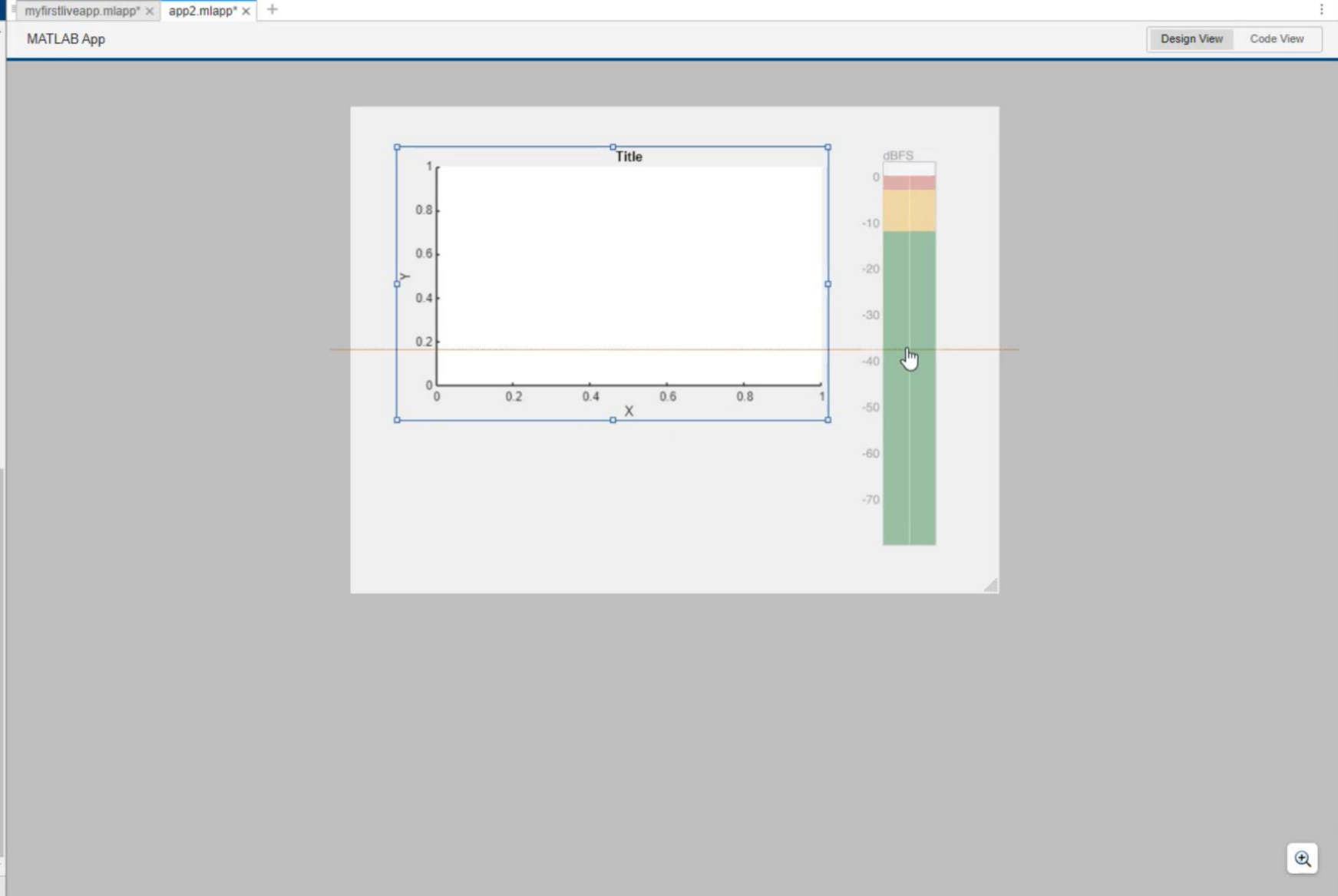
- Gauge Gauge (90 Degree) Gauge (Linear)
- Gauge (Semicircular) Knob Knob (Discrete)
- Lamp Switch Switch (Rocker)
- Switch (Toggle)

AUDIO

- Audio Meter

SIMULINK

- Load Input Button Save Output Button Simulation Controls
- Simulation Progress Time Scope Variable Tuner



Component Browser

Search View:

- app2
 - app.UIFigure
 - app.UIAxes

Axes | Callbacks

Search

LABELS

- Title.String
- XLabel.String
- YLabel.String
- ZLabel.String
- Subtitle.String
- TitleHorizontalAlignment

FONT

- FontName
- FontSize
- FontWeight

TICKS

- XTick
- XTickLabel
- YTick
- YTickLabel

RULERS

GRIDS

MULTIPLE PLOTS

COLOR AND TRANSPARENCY MAPS

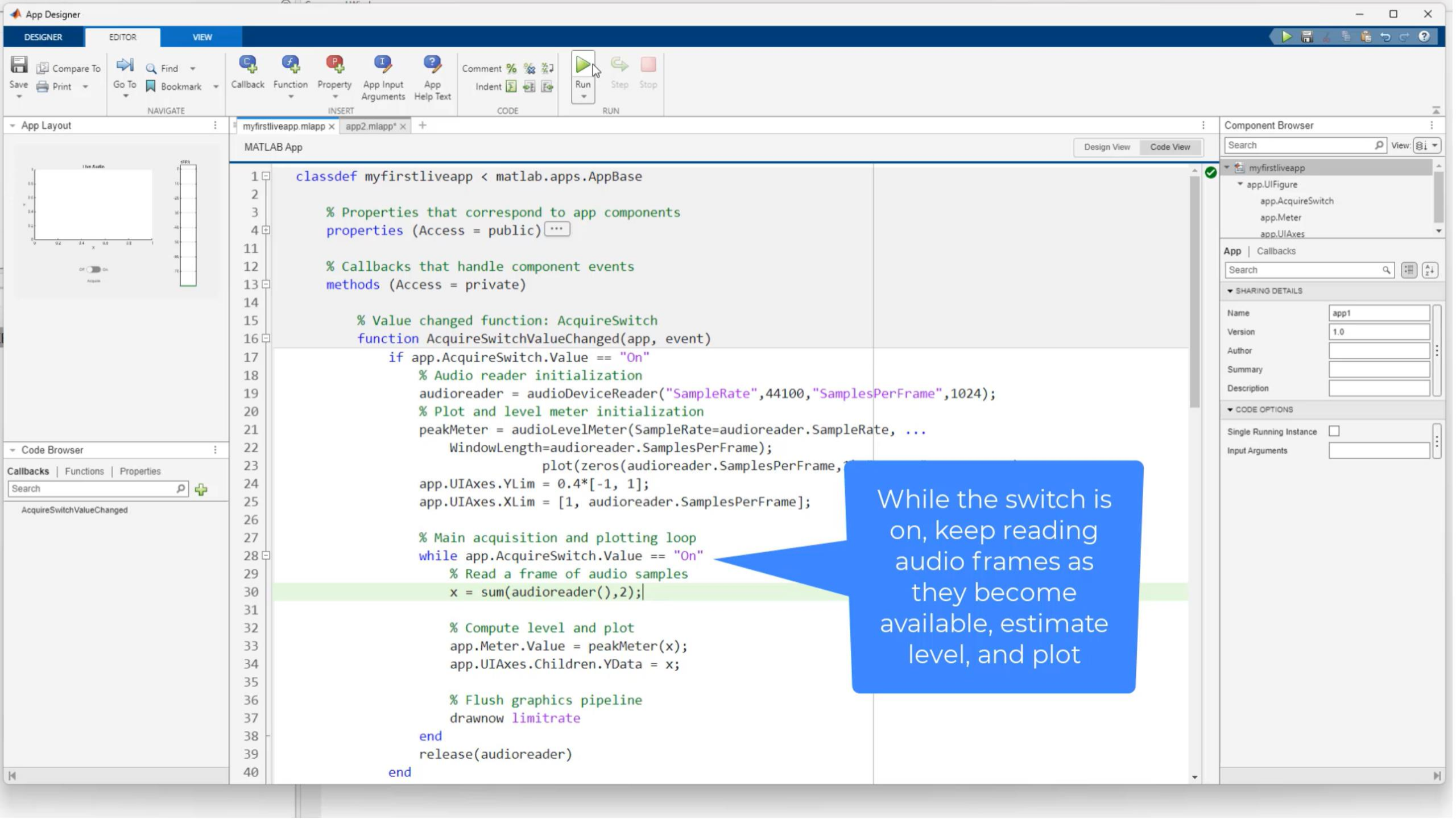
BOX STYLING

POSITION

VIEWING ANGLE

INTERACTIVITY

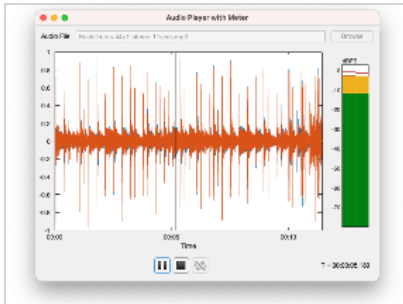
CALLBACK EXECUTION CONTROL



```
1 classdef myfirstliveapp < matlab.apps.AppBase
2
3     % Properties that correspond to app components
4     properties (Access = public) ...
11
12     % Callbacks that handle component events
13     methods (Access = private)
14
15         % Value changed function: AcquireSwitch
16         function AcquireSwitchValueChanged(app, event)
17             if app.AcquireSwitch.Value == "On"
18                 % Audio reader initialization
19                 audioreader = audioDeviceReader("SampleRate",44100,"SamplesPerFrame",1024);
20                 % Plot and level meter initialization
21                 peakMeter = audioLevelMeter(SampleRate=audioreader.SampleRate, ...
22                     WindowLength=audioreader.SamplesPerFrame);
23                 plot(zeros(audioreader.SamplesPerFrame, ...
24                     app.UIAxes.YLim = 0.4*[-1, 1];
25                     app.UIAxes.XLim = [1, audioreader.SamplesPerFrame];
26
27                 % Main acquisition and plotting loop
28                 while app.AcquireSwitch.Value == "On"
29                     % Read a frame of audio samples
30                     x = sum(audioreader(),2);
31
32                     % Compute level and plot
33                     app.Meter.Value = peakMeter(x);
34                     app.UIAxes.Children.YData = x;
35
36                     % Flush graphics pipeline
37                     drawnow limitrate
38                 end
39                 release(audioreader)
40             end
41         end
42     end
43 end
```

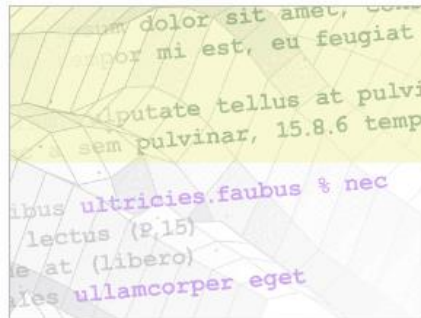
While the switch is on, keep reading audio frames as they become available, estimate level, and plot

Your handouts include additional resources to get started with App prototypes, including with hardware I/O and AI models



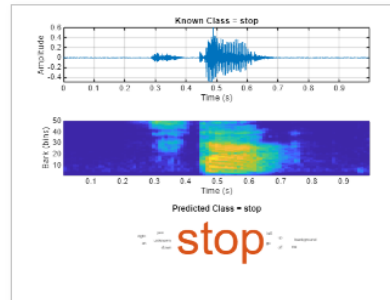
Create an App to Play and Visualize Audio Files

Create an app in App Designer to play and visualize audio files.



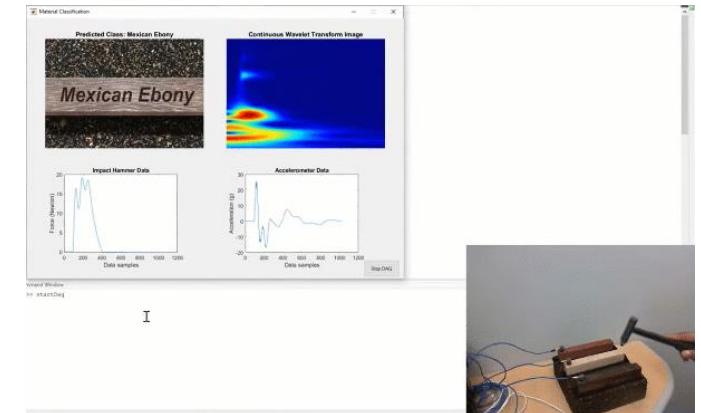
Audio Input and Audio Output

Read audio from a file and write audio to speakers.



Speech Command Recognition Using Deep Learning

Use a pretrained deep learning model to perform speech command recognition on streaming audio.



[Wood Species Classification Using Vibration Signature](#) on GitHub

LLM-Based Voice Assistant

Off On

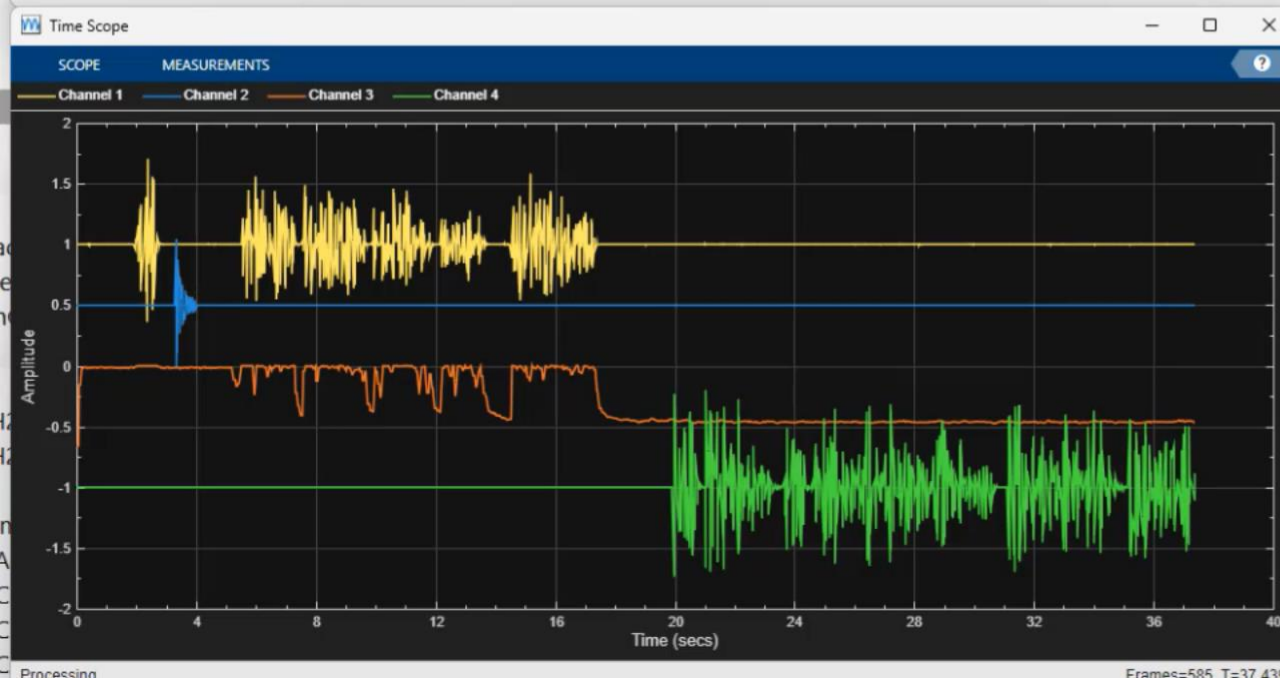
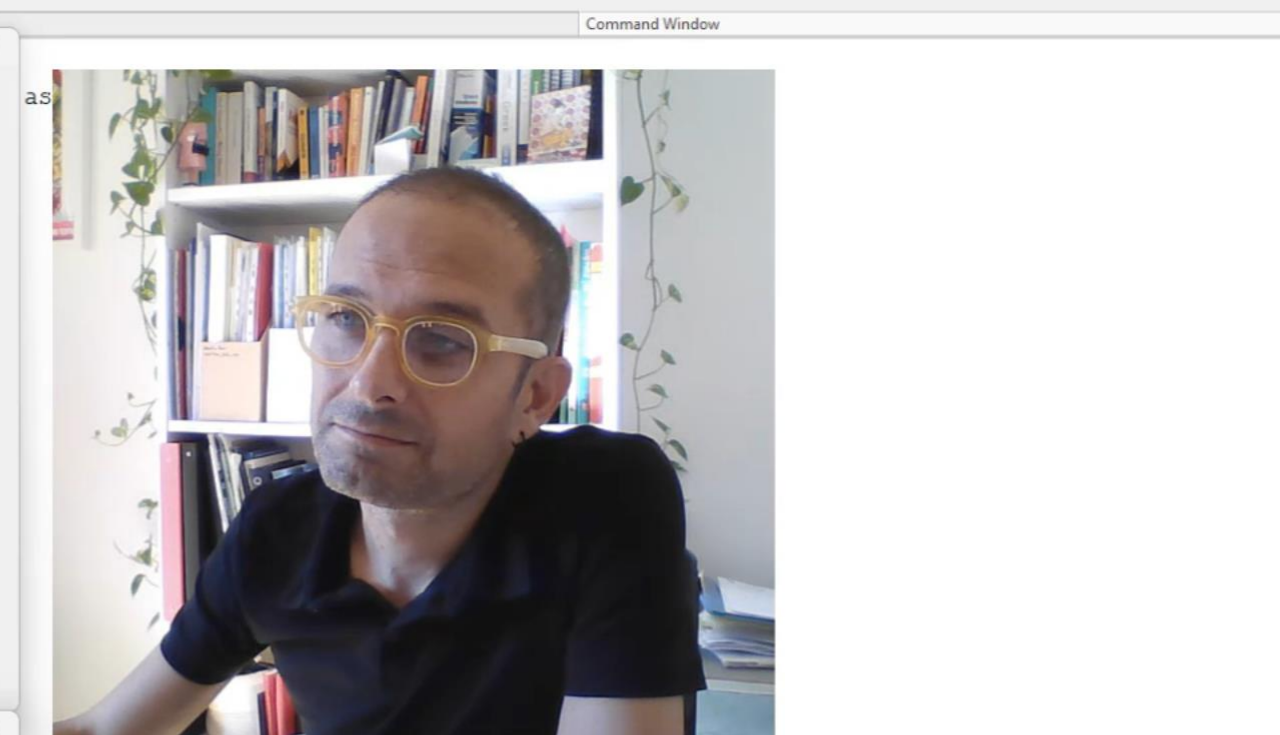
dBFS

0.5
0
-0.5

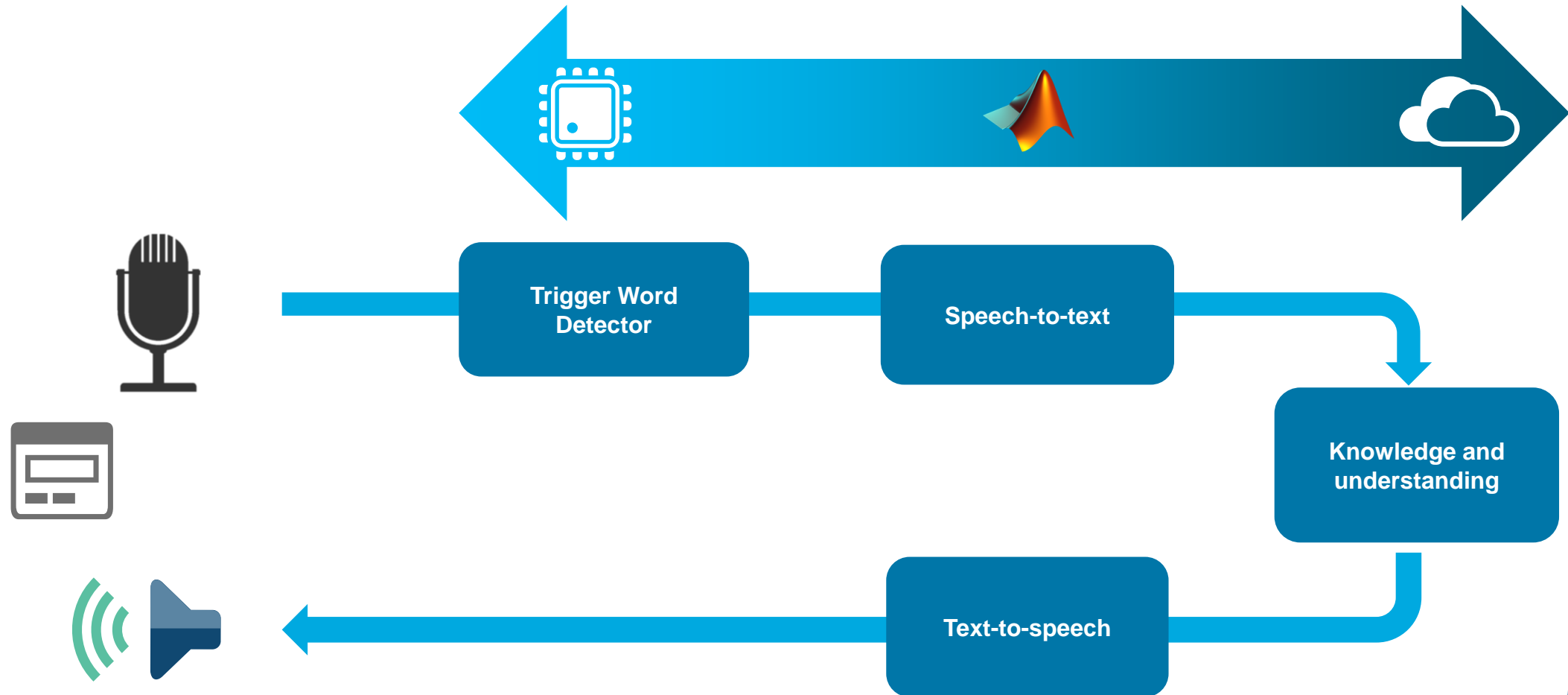
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

types of AI at work behind this Voice Assistant app. Do you think that may have been enough to give them a decent idea?

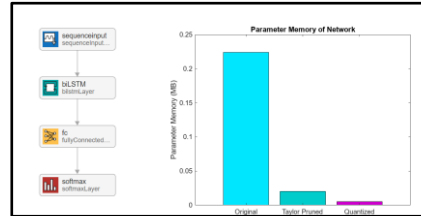
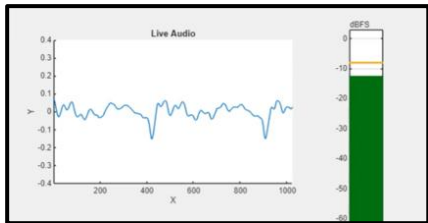
Machine: As an AI, I don't have direct insight into your presentation, but generally, a 20-minute explanation should be adequate to provide a basic understanding of AI and voice assistant technology. If your audience wants to learn more detailed or complex aspects, they may need additional resources or explanations.



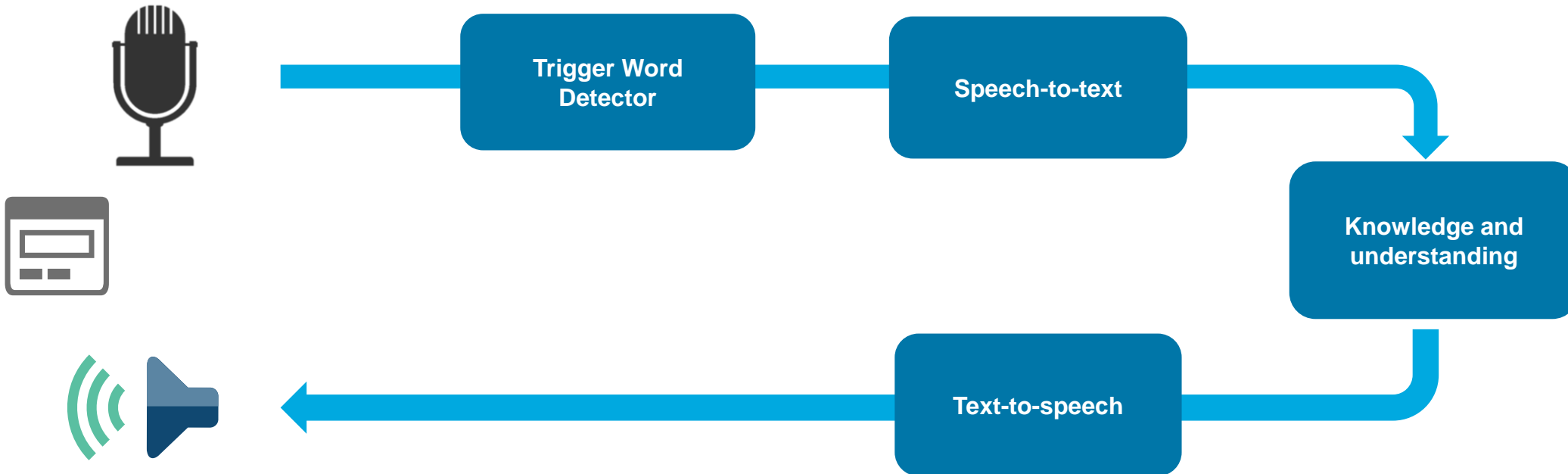
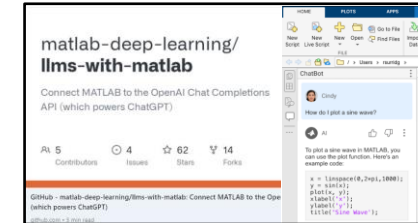
In conclusion: using AI can mean many different things, and all can be brought together in MATLAB to create application prototypes



In conclusion: using AI can mean many different things, and all can be brought together in MATLAB to create application prototypes



```
[x, fs] = text2speech("hello world")  
  
Listen to the synthesized speech.  
  
sound(x, fs)
```



In conclusion: using AI can mean many different things, and all can be brought together in MATLAB to create application prototypes

Questions?

