Design and Implementation of Real-Time Signal Processing Applications

Jegan Mani - Pilot Engineer

Viju Ravichandran, PhD - Education Technical Evangelist
...incorporate functions of **sensing, actuation, and control** in order to describe and analyze a situation, and **make decisions** based on the **available data** in a **predictive or adaptive** manner, thereby performing smart actions....

Smart Systems are everywhere
Challenges in Signal Processing 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.”
Motivation

- Demo
  - Android App based Edge Detection
  - Raspberry Pi based Edge Detection and Image inversion
Agenda

- Acquiring and Analyzing Signals
- Developing Signal Processing algorithms
- Deployment on Hardware
- Audio System Toolbox
Technical Computing Workflow

Access
- Files
- Software
- Hardware

Explore and Discover
- Data Analysis & Modeling
- Algorithm Development
- Application Development

Share
- Reporting and Documentation
- Outputs for Design
- Deployment

Code and Applications

Hardware

Software

Files

MathWorks
Technical Computing Workflow

**Access**
- Files
- Software
- Hardware

**Explore and Discover**
- Data Analysis & Modeling
- Algorithm Development
- Application Development

**Share**
- Reporting and Documentation
- Outputs for Design
- Deployment

MathWorks
Problem Statement

- Activity classification using motion sensors on a mobile device
- **Classification**: Walking, Running, Stairs, Rest (no motion)
Data Acquisition

- Data access via Sensors on MATLAB mobile app (Android/iOS)
  - Data to be used: Accelerometer data in three dimensions

- Acquisition process:

```matlab
>> connector on
DNS name: bgl-lravichal1.dhcp.mathworks.com
IP address: 172.18.75.174
Use this link to test the MATLAB Connector:
http://bg1-lravichal1.dhcp.mathworks.com:31415/
If the test is successful, but MATLAB Mobile cannot
your computer might have multiple IP addresses. To
the correct one, see Determining the DNS Name or IP
>> m = mobiledev
```
Algorithm Development

Data exploration to identify distinguishing features between the activities.

Fourier analysis yields dominant frequency components, but not time-frequency localization (essential for mixture of activities).

Explore and Discover

Data Analysis & Modeling

Algorithm Development

Application Development

MathWorks

for k=1:max
x = fft(data)
y = 20*log1
Algorithm Development

- Time and Frequency Localization
  - Frame-based Fourier spectrum
  - Spectrogram

Explore and Discover

Data Analysis & Modeling

Algorithm Development

Application Development

```
for k=1:max
    x = fft(data)
    y = 20*log1
```
Algorithm Development

- Frame size of 10

- Features from accelerometer data (3-D) for each frame
  - Mean (1-D)
  - Standard Deviation (3-D)
  - Maximum (3-D)
  - Maximum of FFT magnitude (1-D)

- Three-layer neural network classifier using above features
Algorithm Development

- Training
Algorithm Development

- Classification
Algorithm Deployment

- Simulink model for deployment
Same device ... different approaches

**Tethered Approach**

Write code and communicate with the device

**Embedded Approach**

Develop a model and program the device
Algorithm Deployment

- Hardware Connectivity – Single-click deployment solution

- Use algorithm as-is and deploy on hardware of your choice e.g. Android/iOS device, Raspberry Pi, Beaglebone Black
Prototyping for product development

MATLAB algorithm

Early validation (listening tests)

Advanced prototyping or production
Prototyping for product development

Why?:
- From algorithm developers to other stakeholders
- Human in the loop – “How does it sound?”
- Interactive tuning – “What if we changed that?”

Existing challenges
- Owned by “algorithm” or “DSP” engineer
- C/C++ hard and time-consuming; External libraries into MATLAB don’t work well
- Plugin API is challenging to learn
- Code not re-used for production – Wasteful
Audio System Toolbox
Design and test audio processing systems

- Libraries of audio processing algorithms and examples

- Low-latency audio streaming from and to standard audio interfaces (e.g. ASIO, CoreAudio, ALSA)

- Live-tuning of MATLAB and Simulink via UI and MIDI controls

- VST plugin generation to run on Digital Audio Workstations
Design and test audio processing systems using Audio System Toolbox

- **Desktop prototyping and listening tests**
  - **Pain:** Traditional prototyping is costly and time-consuming
  - **Solution:** Audio streaming in MATLAB and VST plugin generation

- **Custom real-time measurements**
  - **Pain:** Audio test & measurement equipment can be expensive and inflexible
  - **Solution:** Audio acquisition and unlimited custom analysis

- **Real-time audio for teaching DSP**
  - **Pain:** C/C++ and hardware kits take away time from putting signal processing into practice
  - **Solution:** Real-time audio without C/C++ or embedded kits
Plugin generation and source code generation

- PC Application
- Custom Plugin
- Custom prototyping infrastructure
- Speedgoat audio real-time target
- Embedded prototype
- …

MATLAB code

VST plugin

Design & Validation

Prototyping & code export
Challenges in Signal Processing Design

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++ and HDL code generation that enables design continuity and faster simulation
Questions?
Thank you!