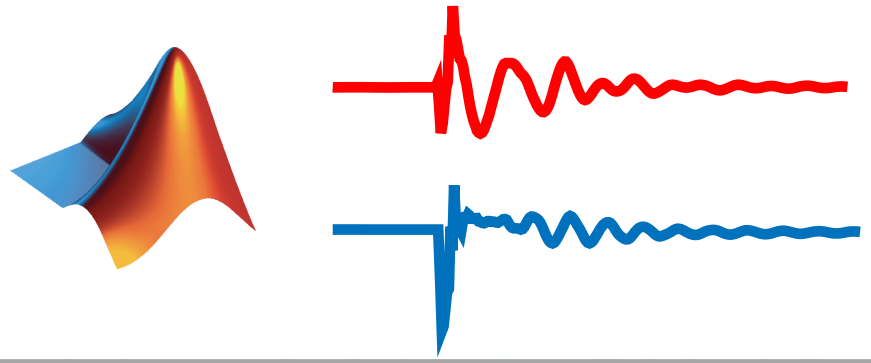
PMU Data-Based Big Data Analytics and Its Deployment Using MATLAB



Mil Shastri

Senior Application Engineer - Energy and Automation

Agenda

- Introduction
- Big Data Synchro-phasor Monitoring and Analytics for Resiliency Tracking (BDSMART)
  - Work by Quanta Technologies LLC using MATLAB under DOE FOA 1861
- Big-data in MATLAB
- Deployment options
- Other relevant work

Agenda

- Introduction
- Big Data Synchrophasor Monitoring and Analytics for Resiliency Tracking (BDSMART)
  - Work by Quanta Technologies LLC using MATLAB
- Big-data in MATLAB
- Deployment options
- Other relevant work

Headquarters

Natick, MA USA

North America

United States

Europe

Finland
France
Germany
Ireland
Italy
Netherlands
Spain
Sweden
Switzerland
UK

Asia-Pacific

Australia
China
India
Japan
Korea

6000+ staff

in 34 offices around the world

$1.65+ billion

in revenues

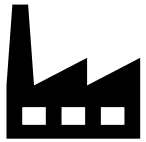
Privately held

and profitable every year

“fleet” data requires efficiently storing and processing huge timeseries datasets.

Automotive

- Vehicles
- Engines
- Controllers



Manufacturing

- Pick & Place machines
- Welding robots
- Material handling systems

Energy

- Wind Turbines
- Solar Panels
- Generators

Agriculture

- Harvesters
- Tractors
- Mining

Healthcare

- Surgical tools
- Wearables
- Digital health equipment

Infrastructure

- Charging stations
- Parking spaces
- Electronic toll collection

Agenda

- Introduction
- Big Data Synchrophasor Monitoring and Analytics for Resiliency Tracking (BDSMART)
  - Work by Quanta Technologies LLC using MATLAB
- Big-data in MATLAB
- Deployment options
- Other relevant work

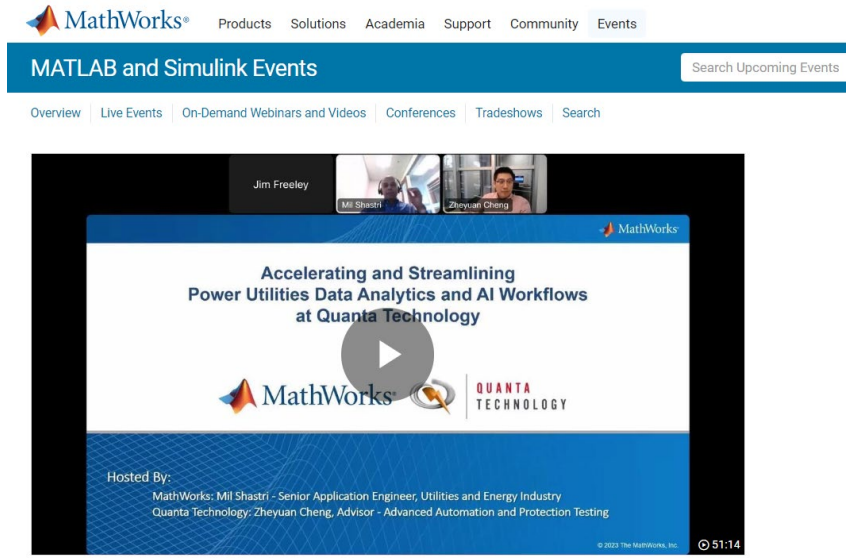
Big Data Synchrophasor Monitoring and Analytics for Resiliency Tracking (BDSMART)

FOA 1861 Awards

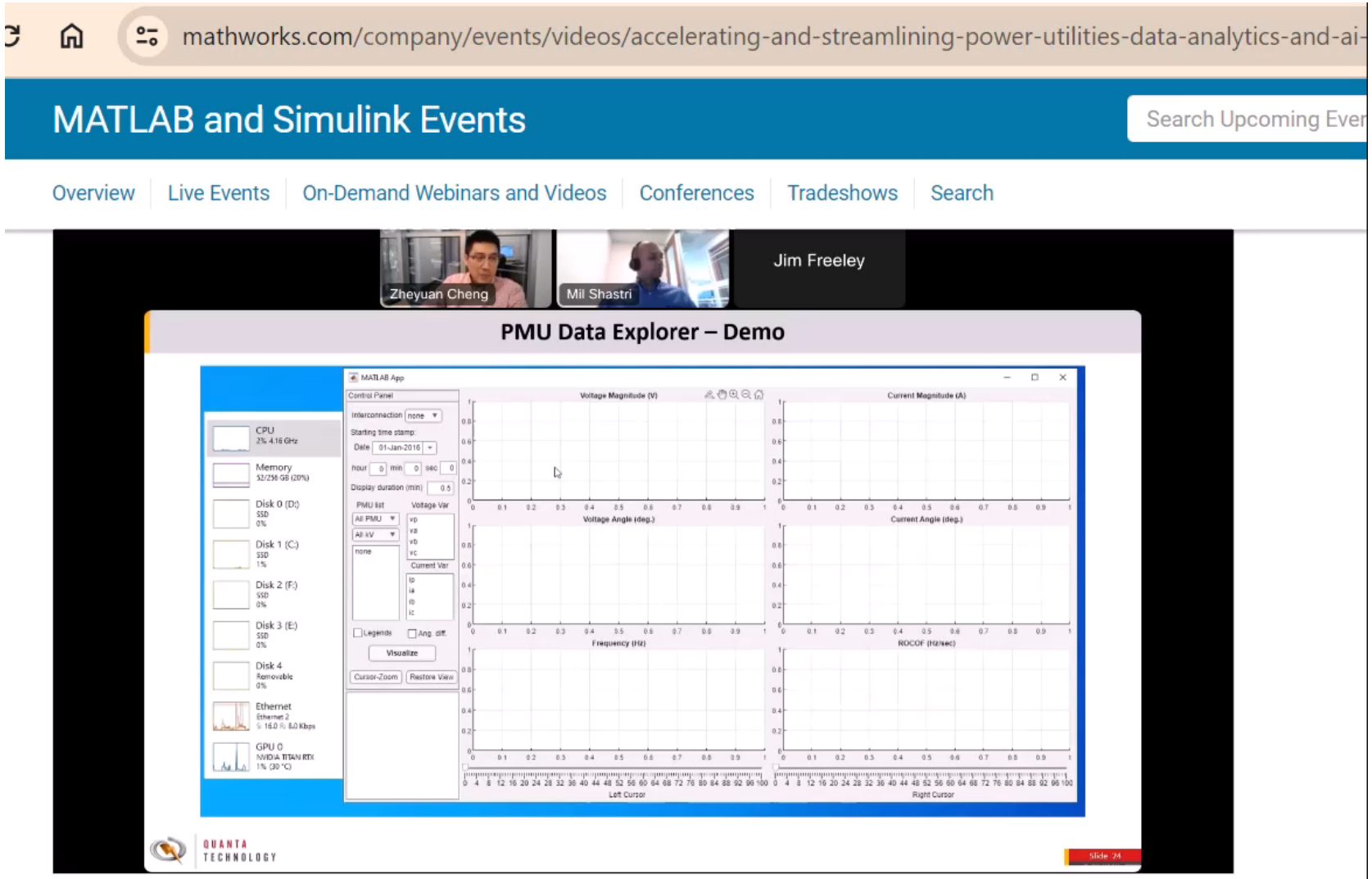
Performer	Team Members	
Iowa State	Electric Power Group (EPG), Google Brain, IBM	Robust Learning of Dynamic Interactions for Enhancing Power System Resilience
SEL	Oregon State University	Machine Learning Guided Operational Intelligence from Synchrophasors
University of California Riverside	EPG, Michigan Technological University	Discovery of Signatures, Anomalies, and Precursors in Synchrophasor Data with Matrix Profile and Deep Recurrent Neural Networks
University of Nevada, Reno	Arizona State University (ASU), IBM, Virginia Tech	Robust Event Diagnostics Platform: Integrating Tensor Analytics and Machine Learning Into Real-time Grid Monitoring
GE	GE Grid Solutions	PMU-Based Data Analytics using Digital Twin and PhasorAnalytics Software
Siemens	Southern Methodist University, Temple University	MindSynchro
Ping Things	NA	Combinatorial Evaluation of Physical Feature Engineering and Deep Temporal Modeling for Synchrophasor Data at Scale
Texas A&M	Temple University, Quanta Technology	Big Data Synchrophasor Monitoring and Analytics for Resiliency Tracking (BDSMART)

DOE FOA 1861

- Challenges:
  - Large data: 27 TB, 1 GB each file from 3 interconnections
  - Data quality issues: missing, oversampled, abnormal
- Approach Quanta Technology used:
  - Preprocess data:
    - Fragmenting data
    - File naming scheme
    - Indexing: monthly, weekly, daily
    - Distributed storage
  - Parallelize
    - Parfor on 64 cores



Accelerating and Streamlining Power Utilities Data Analytics and AI Workflows at Quanta Technology



Accelerating and Streamlining Power Utilities Data Analytics and AI Workflows at Quanta Technology

Other results:

- Process 27 TB of data within hours
- Event detection algorithm using a random forest classifier

Source: URL

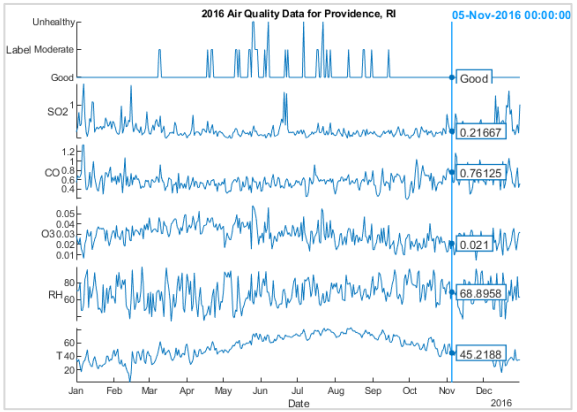
Agenda

- Introduction
- Big Data Synchrophasor Monitoring and Analytics for Resiliency Tracking (BDSMART)
  - Work by Quanta Technologies LLC using MATLAB
- Big-data in MATLAB
- Deployment options
- Other relevant work

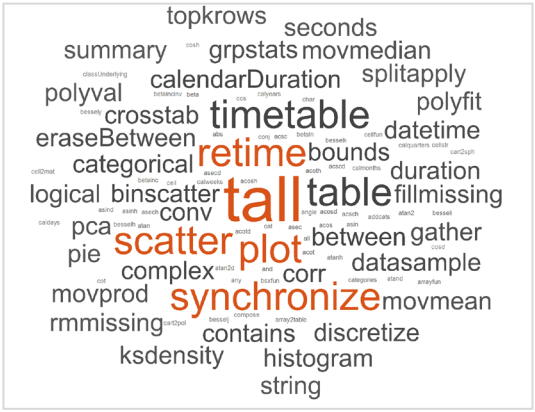
Performing big-data analytics in MATLAB

Performing big-data analytics in MATLAB

Access & Analysis

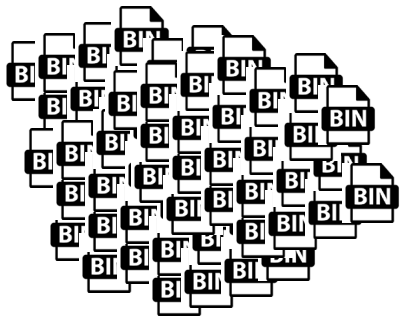


Visualization



Big Data

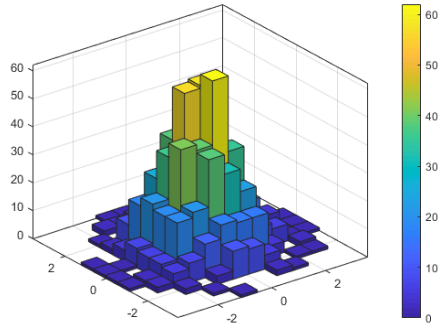
Big Data Engineering and Analysis



Data Engineering

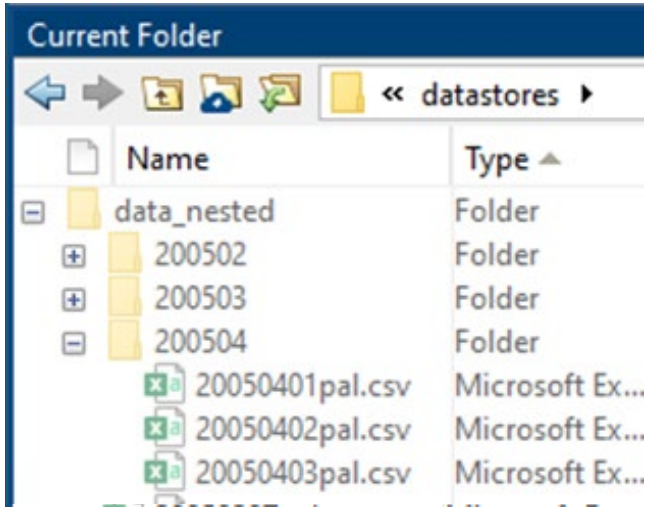
`datastores()`
`+`
`writeall()`

Data Analysis



`datastores()`
`+`
`tall()`

Simplify data engineering workflows with **`datastore`** and **`writeall`**




```
ds = datastore("data_nested\","OutputType","timetable",...
    "IncludeSubfolders",true);
tds = transform(ds,@myfcn);
writeall(tds,"data_writeall1\","OutputFormat","csv",...
    "FolderLayout","duplicate","UseParallel",true)
```

```
function t_out = myfcn(t_in)
    tu = unstack(t_in,"Load","Name");
    t_out = retime(tu,"hourly");
end
```

Efficient and performant big data storage with Parquet files

- `parquetread`, `parquetwrite`, and `parquetinfo` for single files
- `parquetDatastore` and `writeall` for large collections of files

- `rowfilter` object allows filtering of data at read-time

R2021b: 30s

```
data = parquetread(filename);
data = data(data.pickup_datetime >= "2012-03-10" & data.pickup_datetime < "2012-03-11", :);
head(data)
```

- Determine and define row groups

R2022a: 0.3s

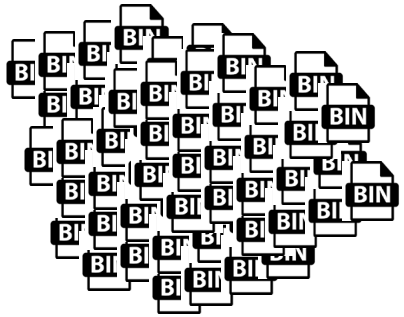
```
filter = rowfilter("pickup_datetime");
filter = filter.pickup_datetime >= "2012-03-10" & filter.pickup_datetime < "2012-03-11"
```

```
filter =
  RowFilter with constraints:

    pickup_datetime >= 2012-03-10 & pickup_datetime < 2012-03-11

  VariableNames: pickup_datetime
```

```
data = parquetread(filename, RowFilter=filter);
head(data)
```

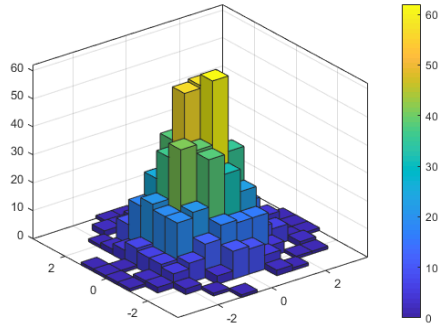
- Convert, import, and export nested data structures

Big Data Engineering and Analysis



Data Engineering

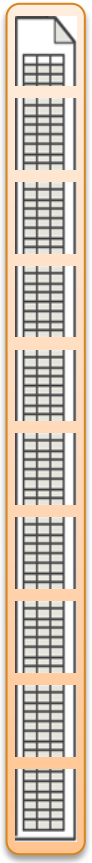
Data Analysis



```
datastores()
+
writeall()
```

```
datastores()
+
tall()
```

`tall` Arrays



- Data type designed for data that doesn't fit into memory
- Lots of observations (hence "tall")
- Lazy evaluation only performs the necessary computations when needed
- Looks like a normal MATLAB array
  - Supports numeric types, tables, datetimes, strings, etc…
  - Supports several hundred functions for basic math, stats, indexing, etc.
  - **Statistics and Machine Learning Toolbox** support
  - Logical indexing with `parquetDatastore` uses `rowfilter` automatically R2022b

Big Data Analysis Without Big Changes

One file

Access Data

```
measured = readtable('PumpData.csv');
measured = table2timetable(measured);
```

Preprocess Data

Select data of interest

```
measured = measured(timerange(seconds(1),seconds(2)),'Speed')
```

Work with missing data

```
measured = fillmissing(measured,'linear');
```

Calculate statistics

```
m = mean(measured.Speed);
s = std(measured.Speed);
```

One hundred files

Access Data

```
measured = datastore('PumpData*.csv');
measured = tall(measured);
measured = table2timetable(measured);
```

Preprocess Data

Select data of interest

```
measured = measured(timerange(seconds(1),seconds(2)),'Speed')
```

Work with missing data

```
measured = fillmissing(measured,'linear');
```

Calculate statistics

```
m = mean(measured.Speed);
s = std(measured.Speed);
```

```
[m,s] = gather(m,s);
```

Customization support for the entire big data workflow

- **Import**
  - Custom datastore
- **Process**
  - Custom datastore transforms
  - Custom tall array functions

```
tall_output1 = matlab.tall.transform(@fcn, tall_input);
tall_output2 = matlab.tall.reduce(@fcn, @reducefcn, tall_input);
tall_output3 = matlab.tall.movingWindow(@movefcn, windowSize, tall_input);
```

- **Export**
  - Custom write capabilities for both datastore `writeall` and tall `write`

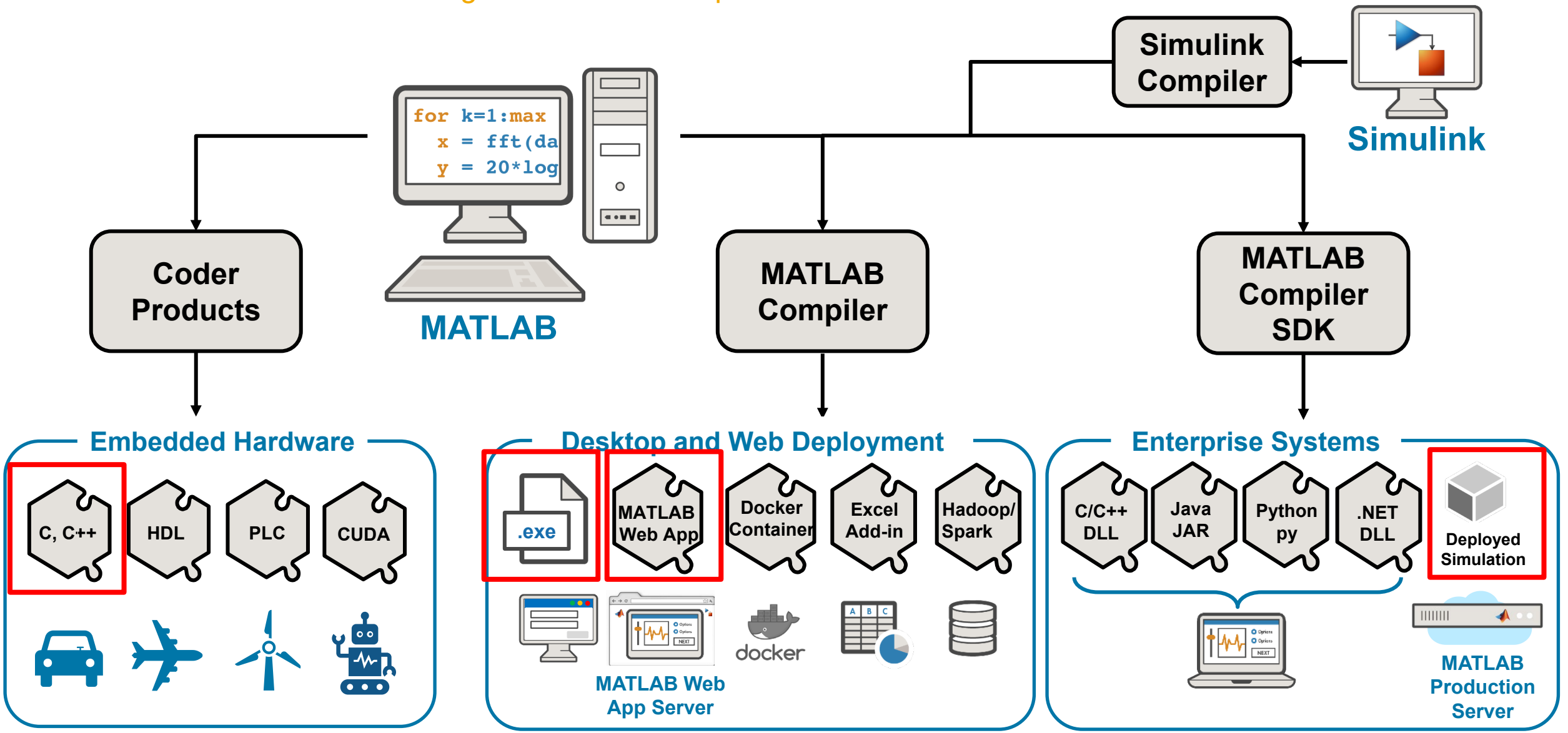
```
write("OutputFolder", tt, "FileType", "text");
write("OutputFolder", tt, "FileType", "parquet");
write("OutputFolder", tt, "WriteFcn", @myWriteFcn);
```

Agenda

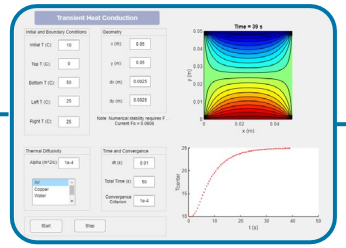
- Introduction
- Big Data Synchrophasor Monitoring and Analytics for Resiliency Tracking (BDSMART)
  - Work by Quanta Technologies LLC using MATLAB
- Big-data in MATLAB
- Deployment options
- Other relevant work

Power Utility Monitoring and System Deployment

MathWorks workflows can integrate with different platforms:

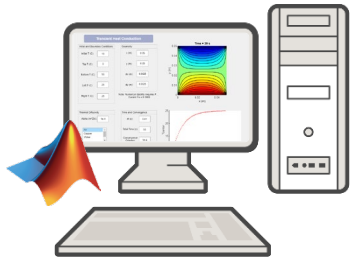


Sharing desktop and web apps

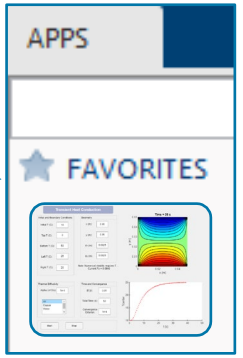


App Designer

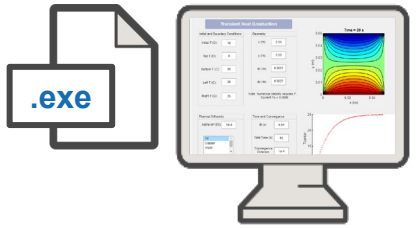
.mlapp



.mlappinstall



MATLAB
Compiler

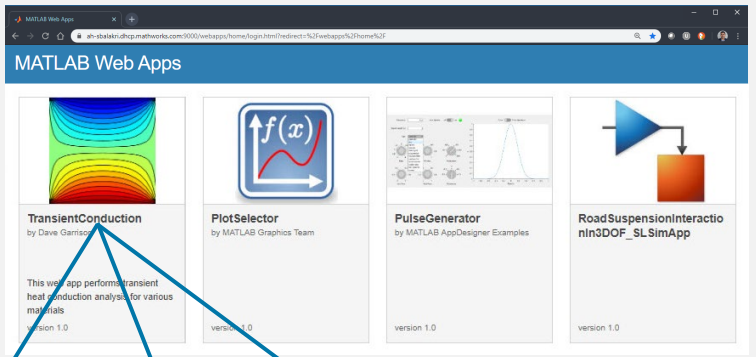


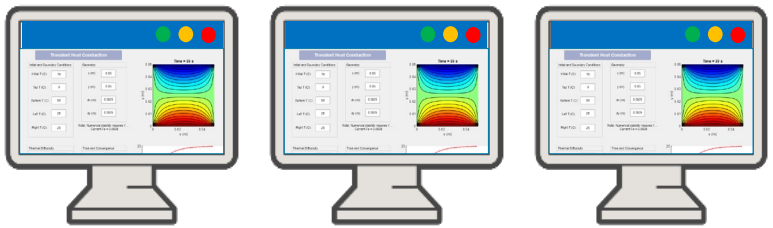
.exe

Desktop App

MATLAB
Web App

MATLAB Web
App Server

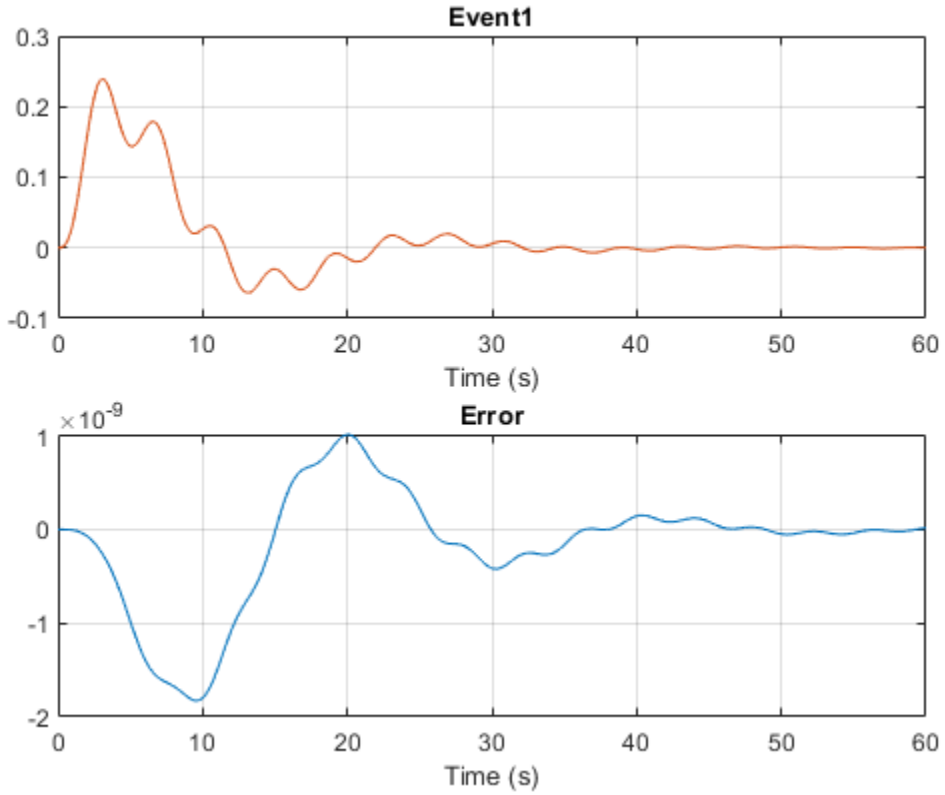


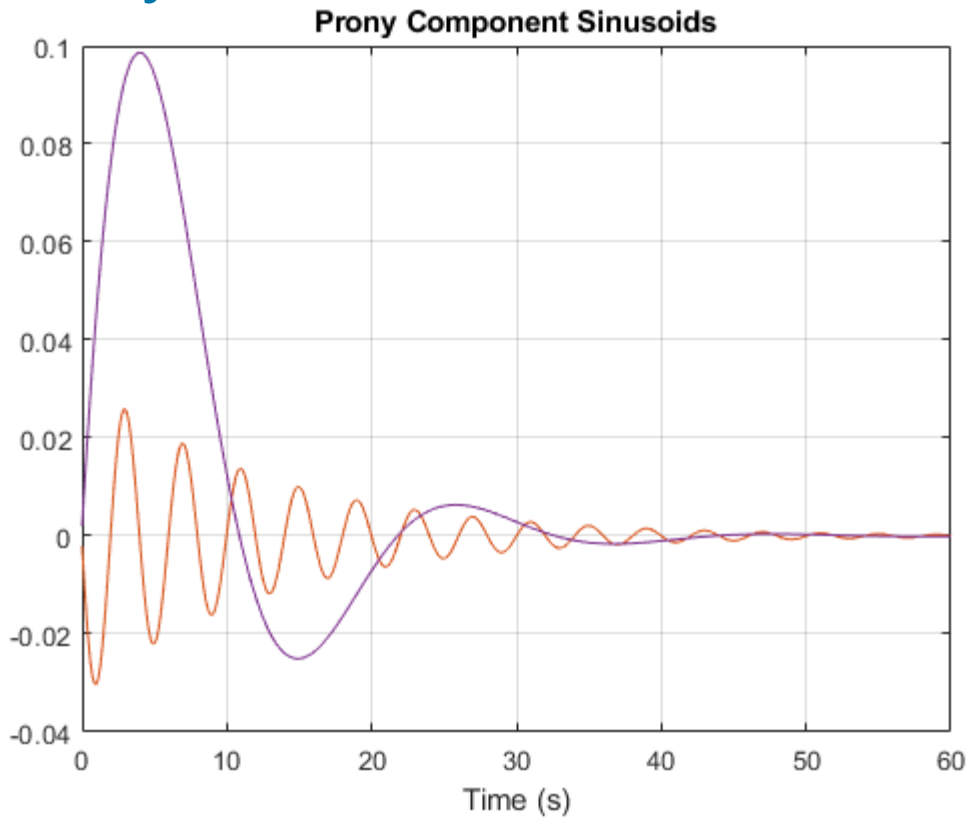


MATLAB
MATLAB Compiler
MATLAB Web App Server

Example

Deployment Example:
Event Characterization using Prony Analysis in MATLAB





	Location	FrequencyHz	DampingRatio
1	"Event1"	0.0500	0.4000
2	"Event1"	0.2500	0.0500

Oscillation Signal Reconstruction with Simple Sinusoids

Oscillation can be reconstructed with a series of exponential responses and sinusoids:

$$y(t) = \sum_{i=1}^{N} A_i e^{\sigma_i t} \cos[2\pi f_i t + \theta_i]$$

Exponential damping (or unstable mode)

Oscillation Frequency and Phase

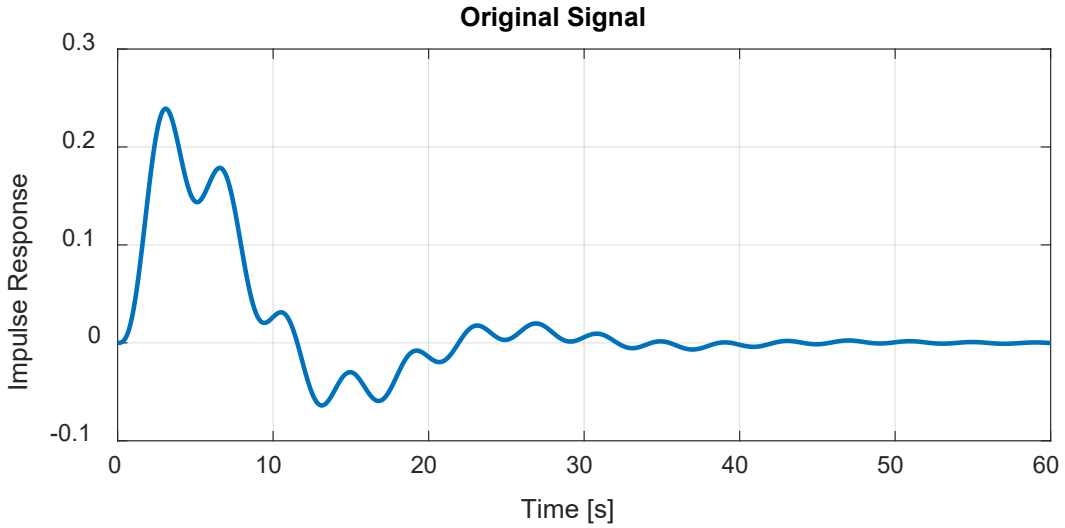
A_i - Amplitude

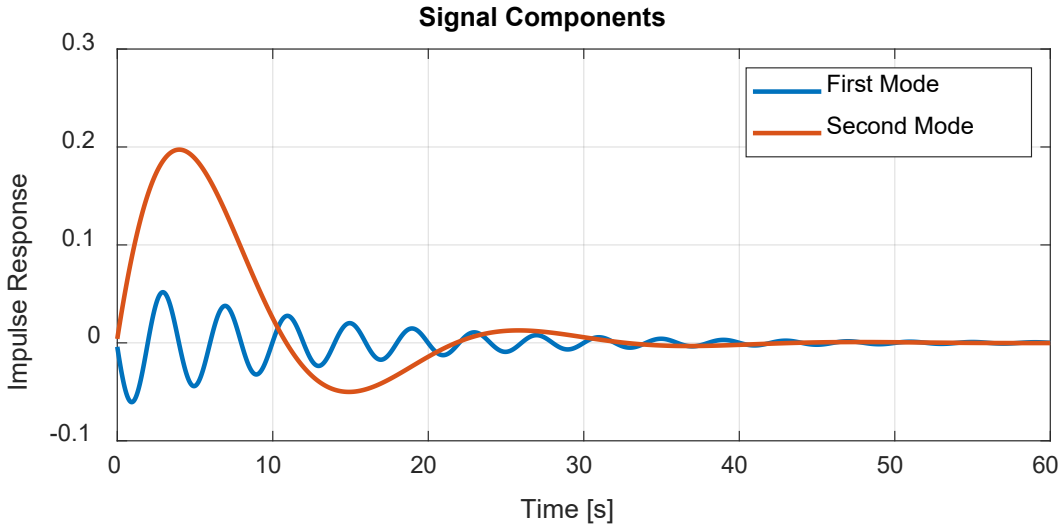
σ_i - Damping (stable/unstable)

f_i - Frequency

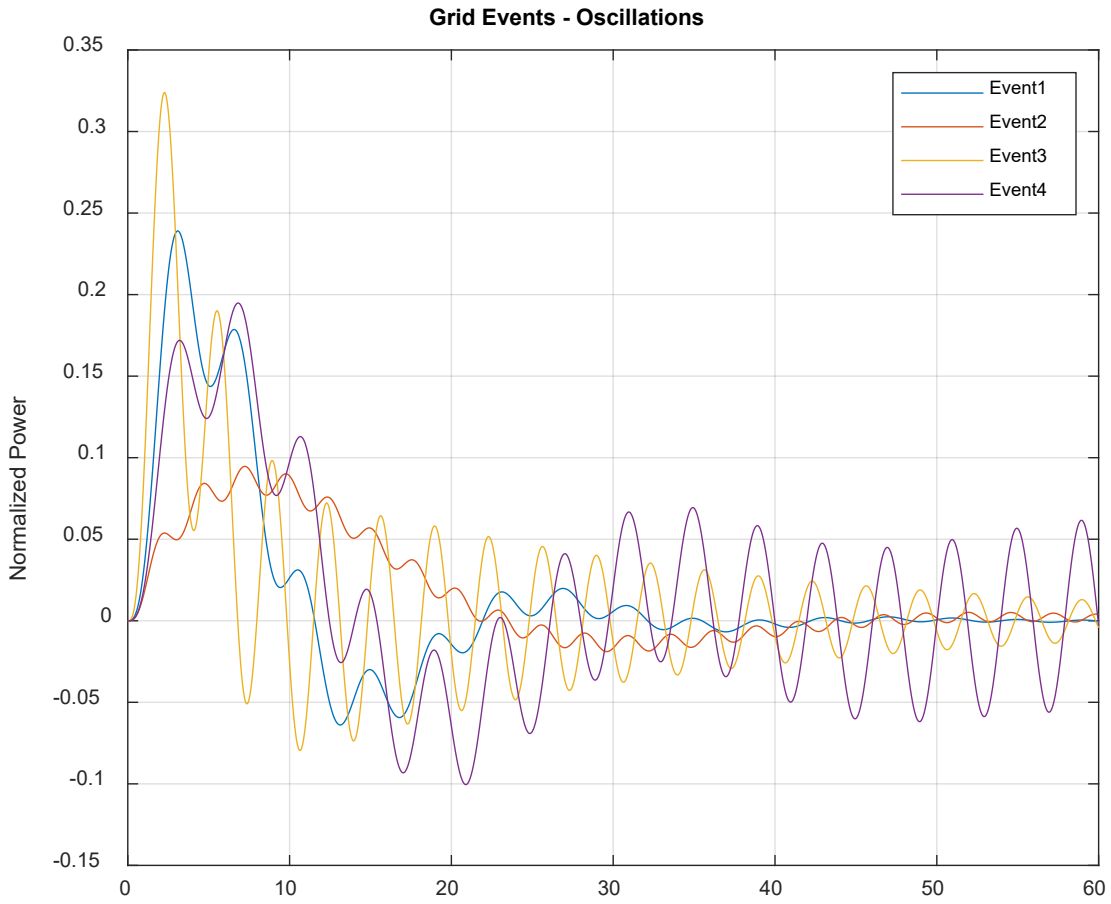
θ_i - Phase

N - Number of Modes



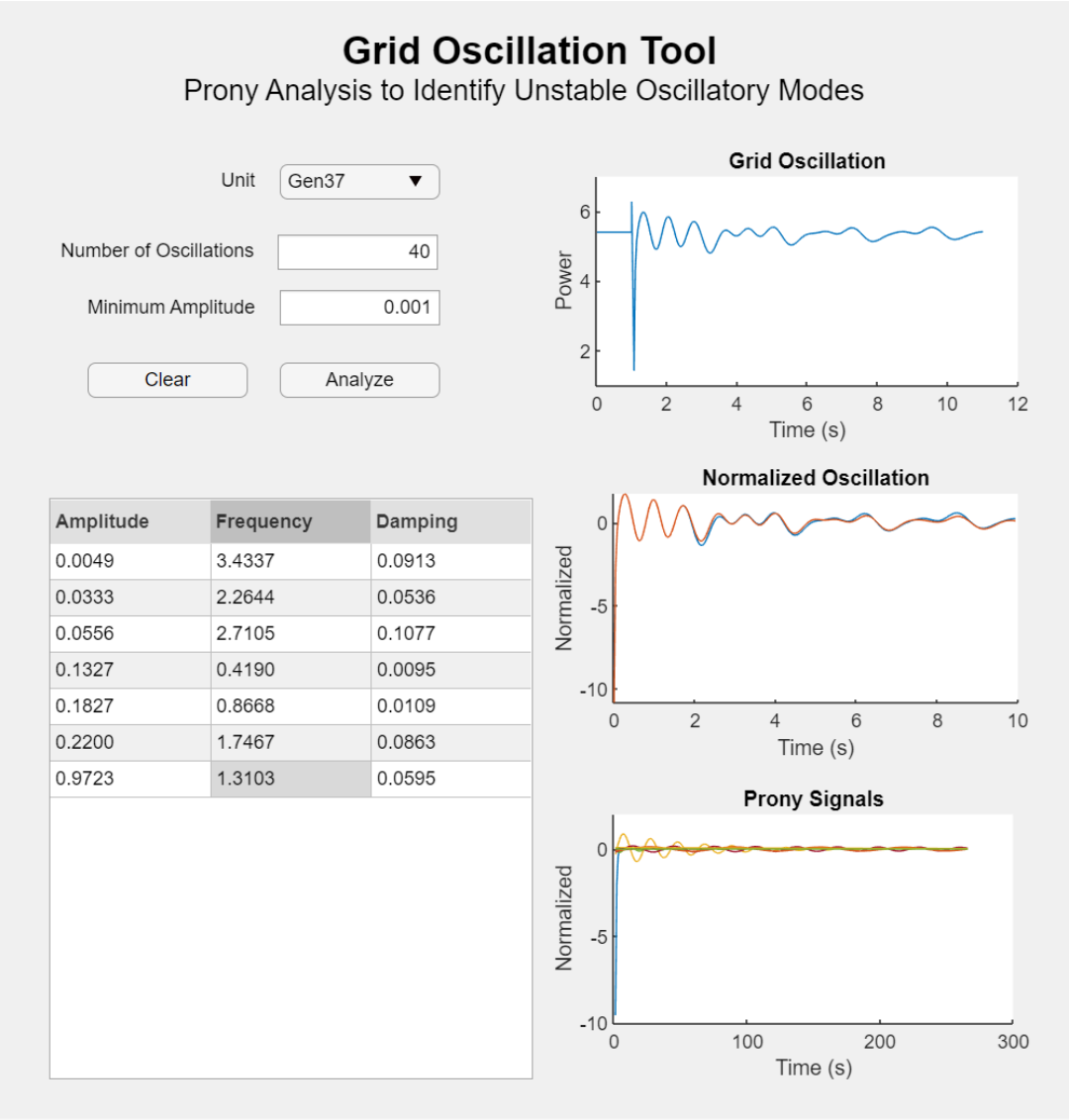


Multi-Event Identification of Damping



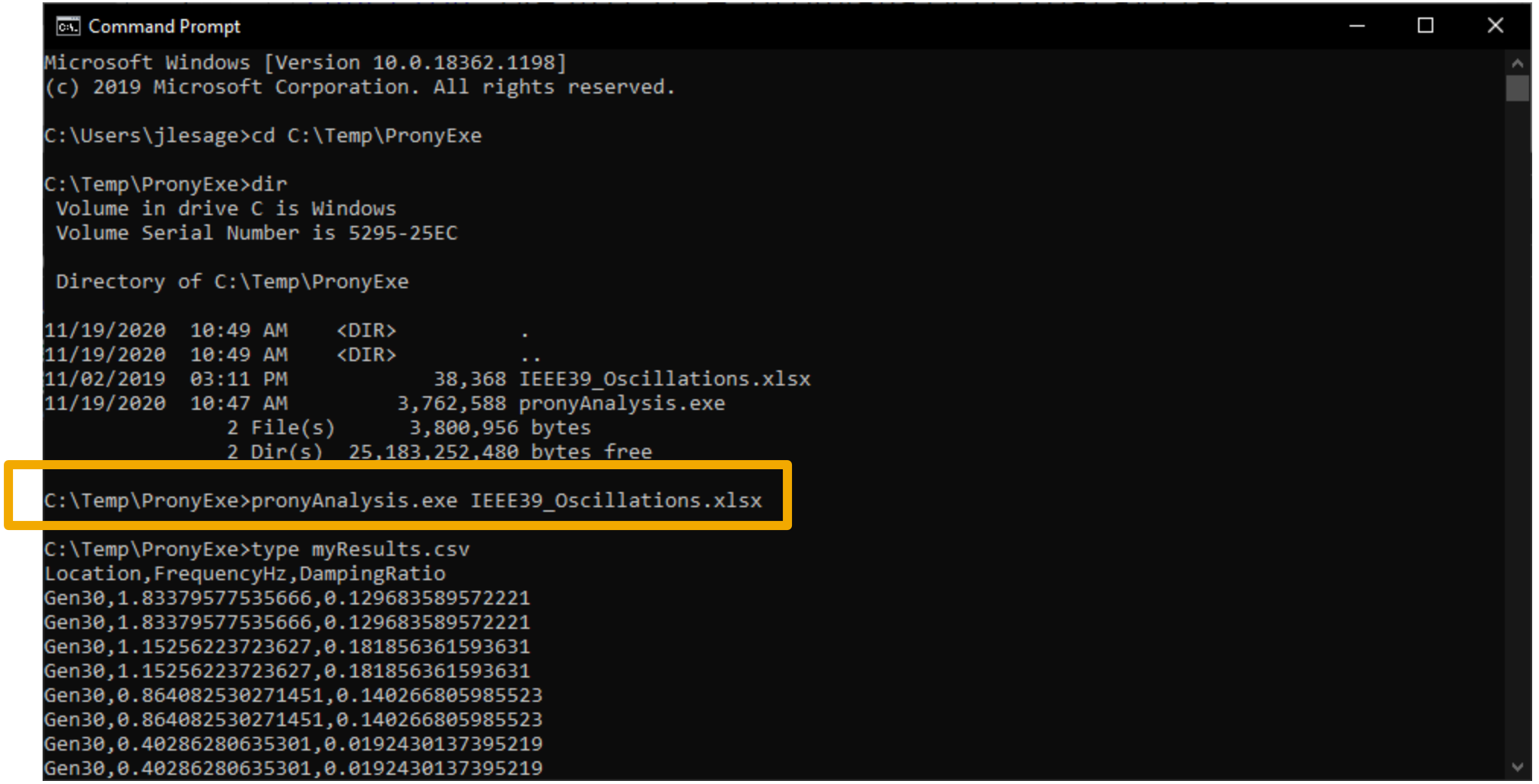
Signal Number	Frequency (Hz)	Damping Ratio
Event1	0.050	0.400
Event1	0.250	0.050
Event2	0.025	0.500
Event2	0.380	0.010
Event3	0.076	0.800
Event3	0.300	0.020
Event4	0.038	0.300
Event4	0.250	-0.005

MATLAB example: Web-app



Command-Line Automation without MATLAB Installed

Enable Connection to Other Tools/Workflows



Agenda

- Introduction
- Big Data Synchrophasor Monitoring and Analytics for Resiliency Tracking (BDSMART)
  - Work by Quanta Technologies LLC using MATLAB
- Big-data in MATLAB
- Deployment options
- Other relevant work

Other resources on Electrification

Motor Drives and Traction Motors

Develop embedded software for motor-inverter systems

Renewable Energy and Energy Storage

Perform grid-scale integration studies, develop wind and solar farm architecture and control systems

Electric Vehicles and Transportation

Perform vehicle-level electrical system and control design for electric transportation

Battery Systems

Design battery packs and develop battery management systems

Microgrid, Smart Grid, and Charging Infrastructure

Develop network architecture and perform system-level and control system design of power system infrastructure

Fuel Cells and Electrolyzers

Develop architectures and controls for PEM fuel cells and electrolyzers in hydrogen systems

Power Conversion

Develop embedded software for high, medium, and low power converter architectures

Generation, Transmission, and Distribution

Conduct bulk power grid analysis and planning for generation, transmission, and distribution systems

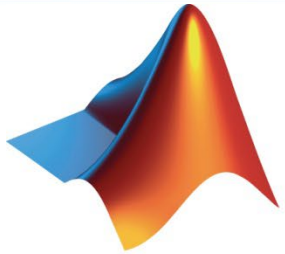
Building Energy Management

Perform power system analysis and energy management design for residential and commercial buildings

AI for Electrification

Apply artificial intelligence (AI) techniques to the design, control, and operation of power electronic devices and power systems.

Page URL



MathWorks®

Accelerating the pace of engineering and science

Thank you