MATLAB EXPO 2019

에너지 최적화를 위한 에너지 관리 시스템(EMS)

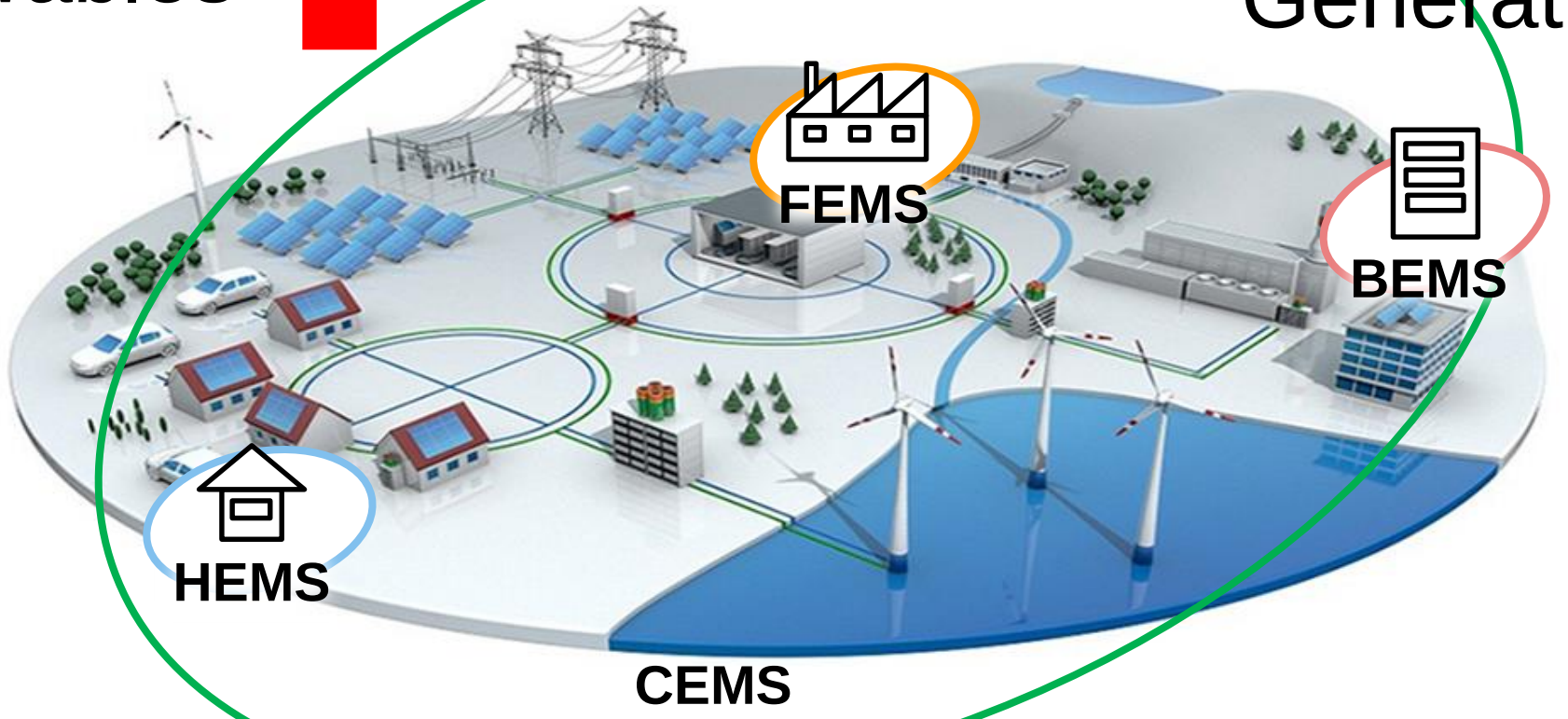
강효석

Motivation

Renewables

Uncertainty

Distributed Generation



Data

Regulatory Oversight

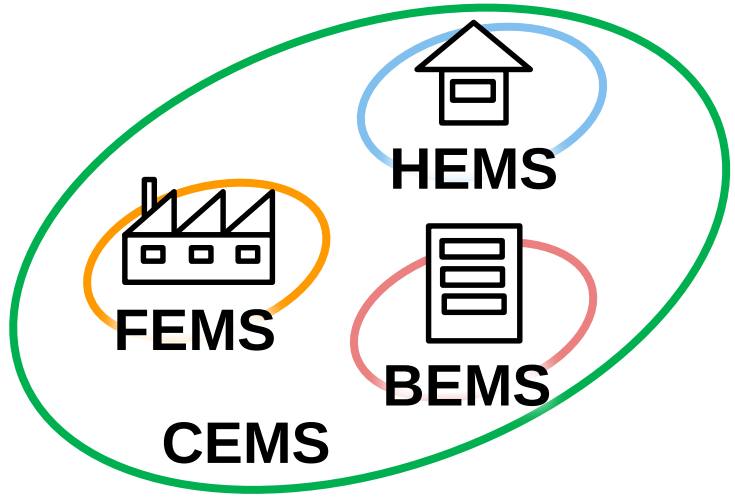
Motivation

As-is

Static policies
+
Off-line data
+
Manual operation

To-be

Dynamic policies
+
Real-time data
+
Automated operation



Smart Energy Management Systems (EMS) are a MUST in a smart energy society

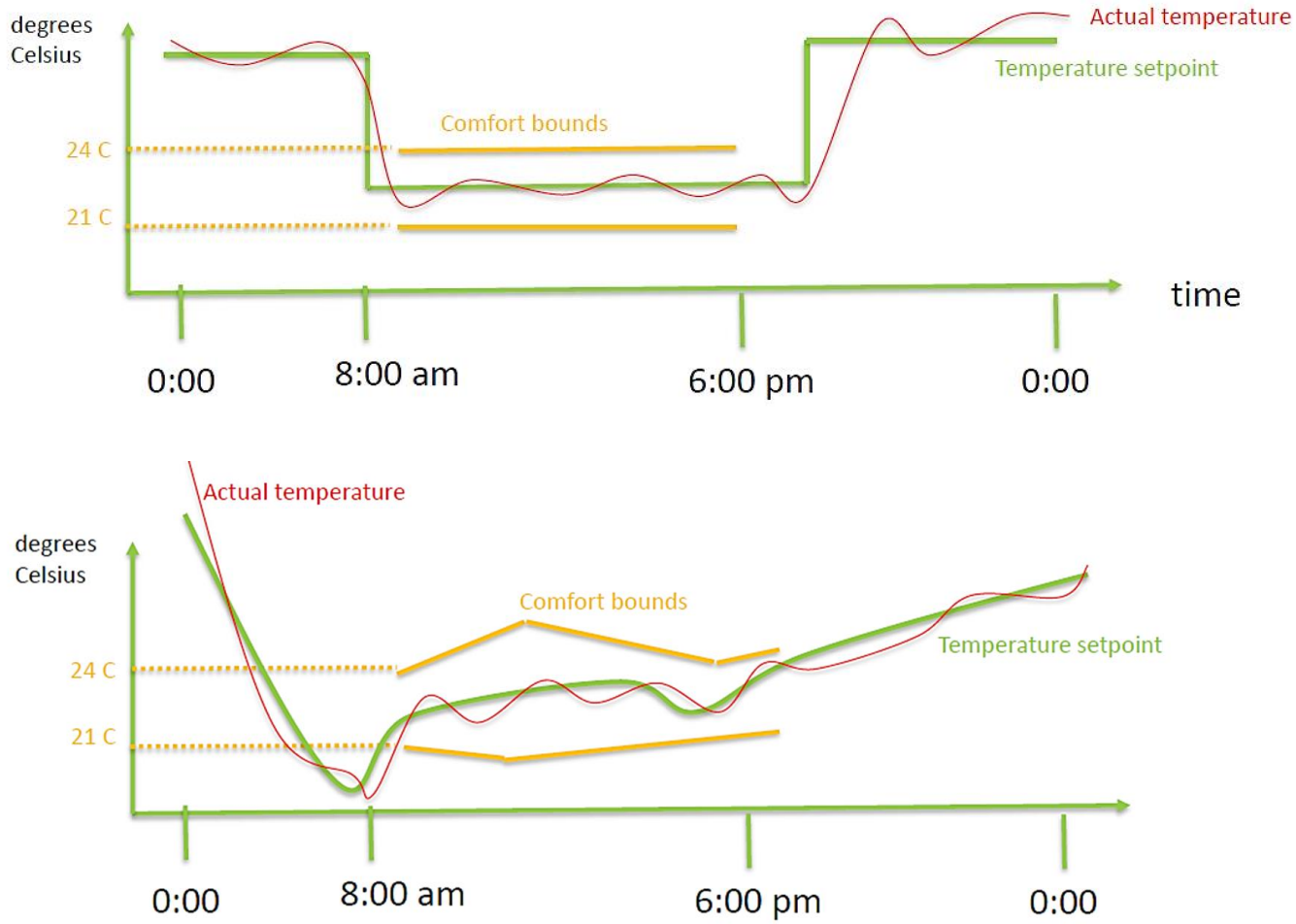
BuildingIQ

MathWorks

Traditional EMS

25% cost reduction

Smart EMS



Develop Smart EMS

Big Data

Data Analytics

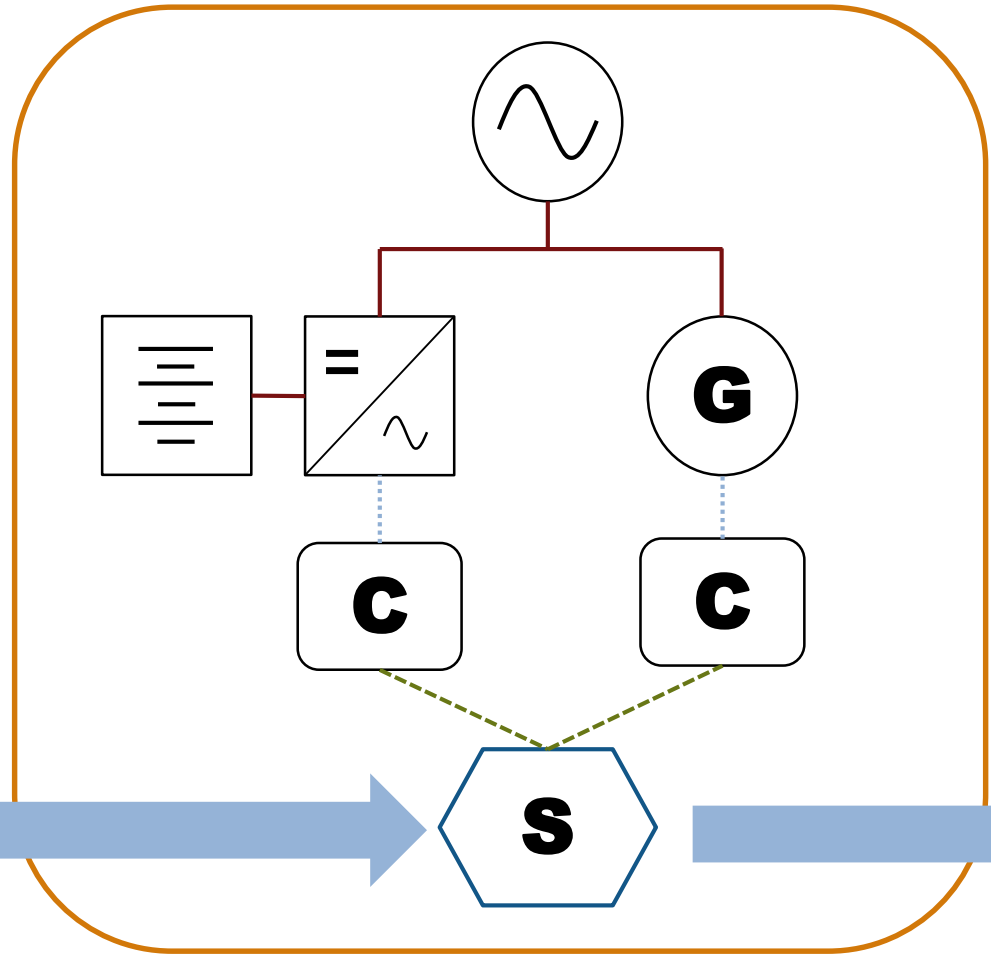
Prediction Model

Forecast data

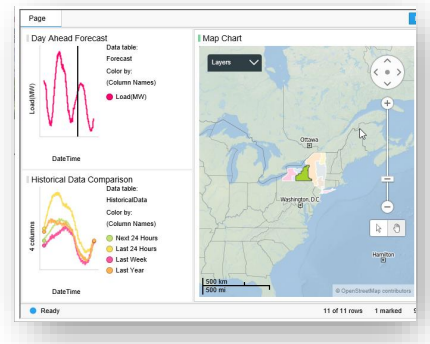
Optimization

Control Algorithm

System Simulation



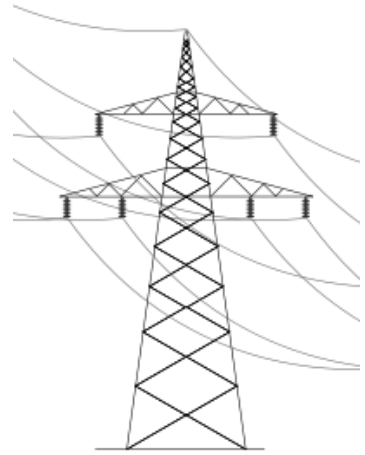
Deploy



Community EMS

CEMS – *community with variable loads*

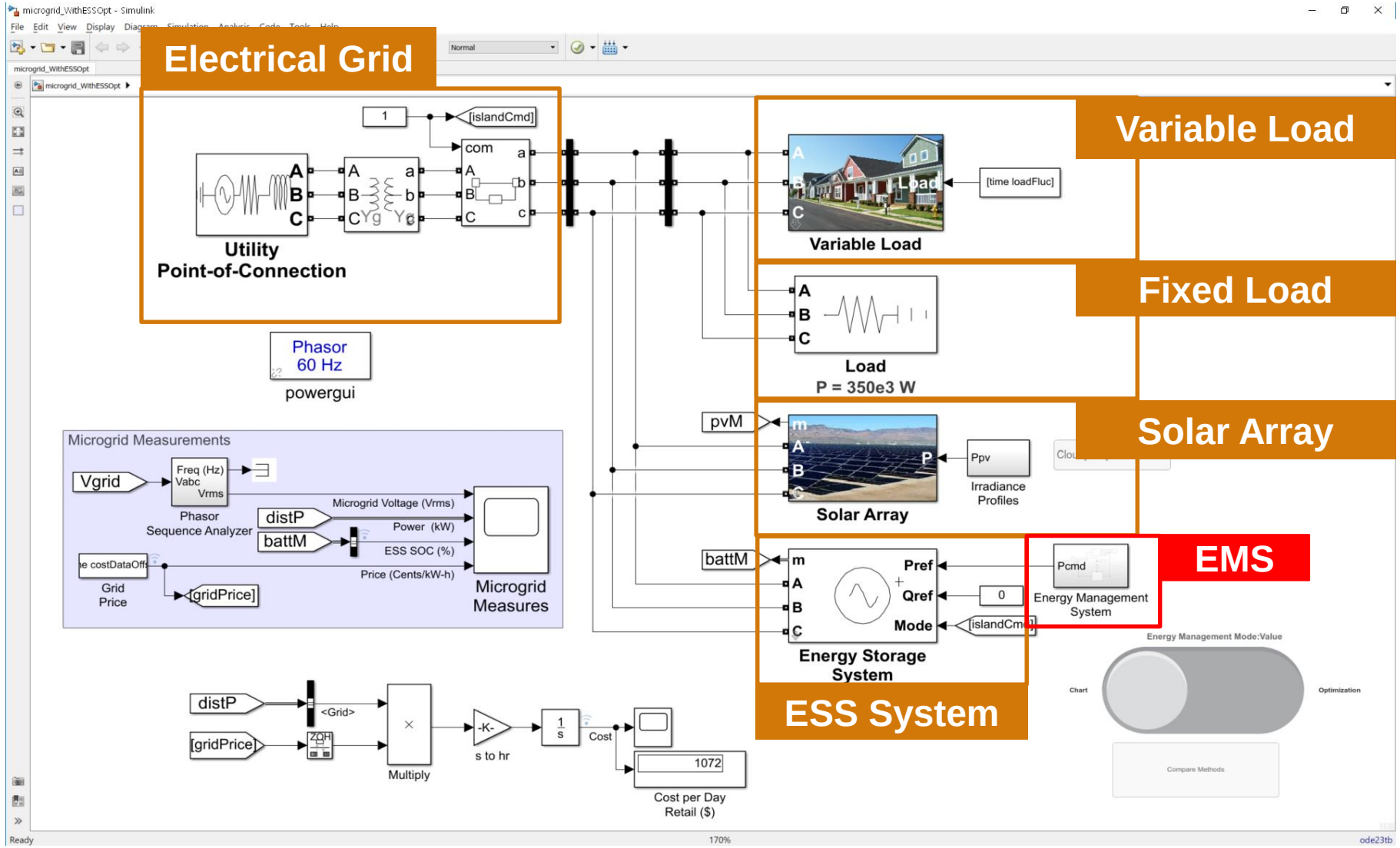
PV Panels:
- MPPT Control
- Power electronics



Decision Logic:
- Store/draw power
- Use external grid
- Optimize energy

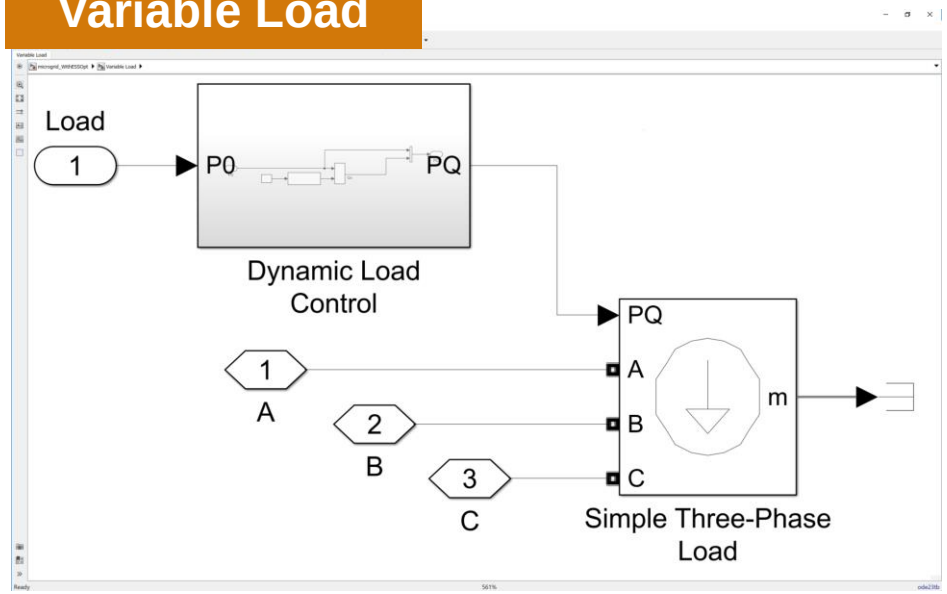
Battery System:
- Charge controls
- Discharge controls
- Power electronics

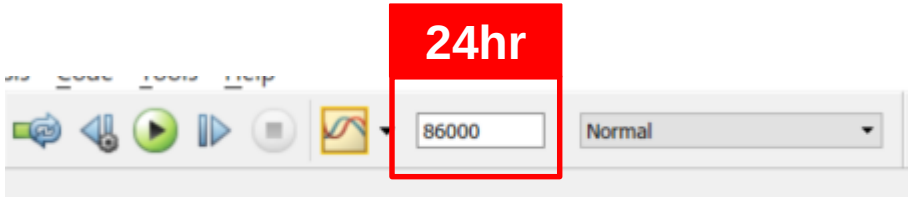
Community EMS



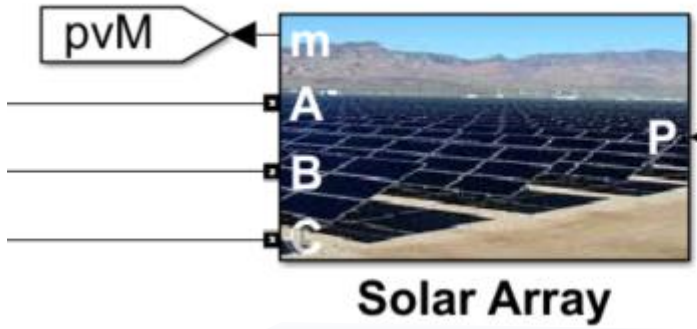
Community EMS

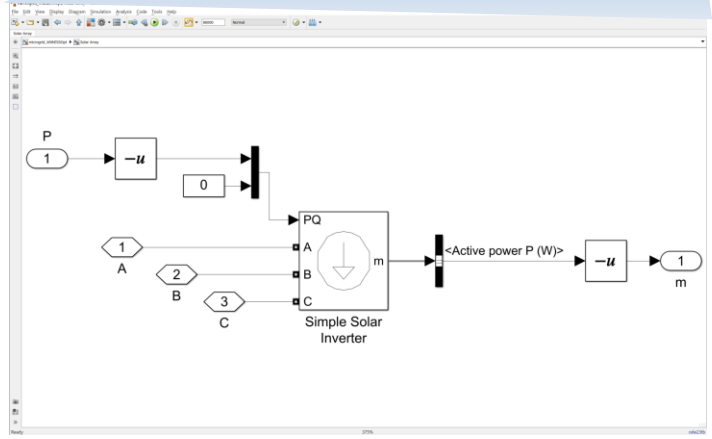
Variable Load

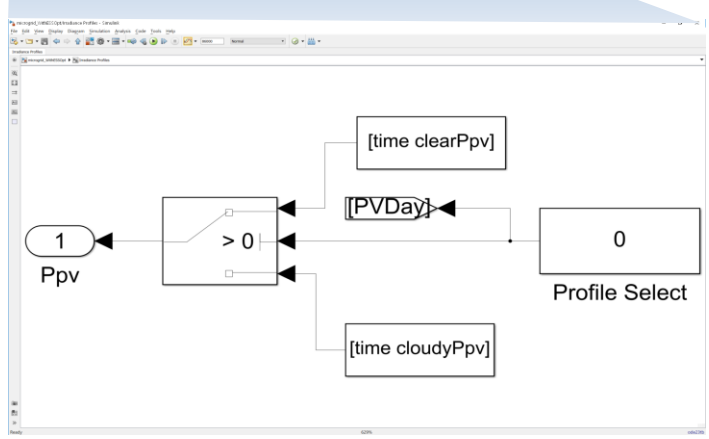




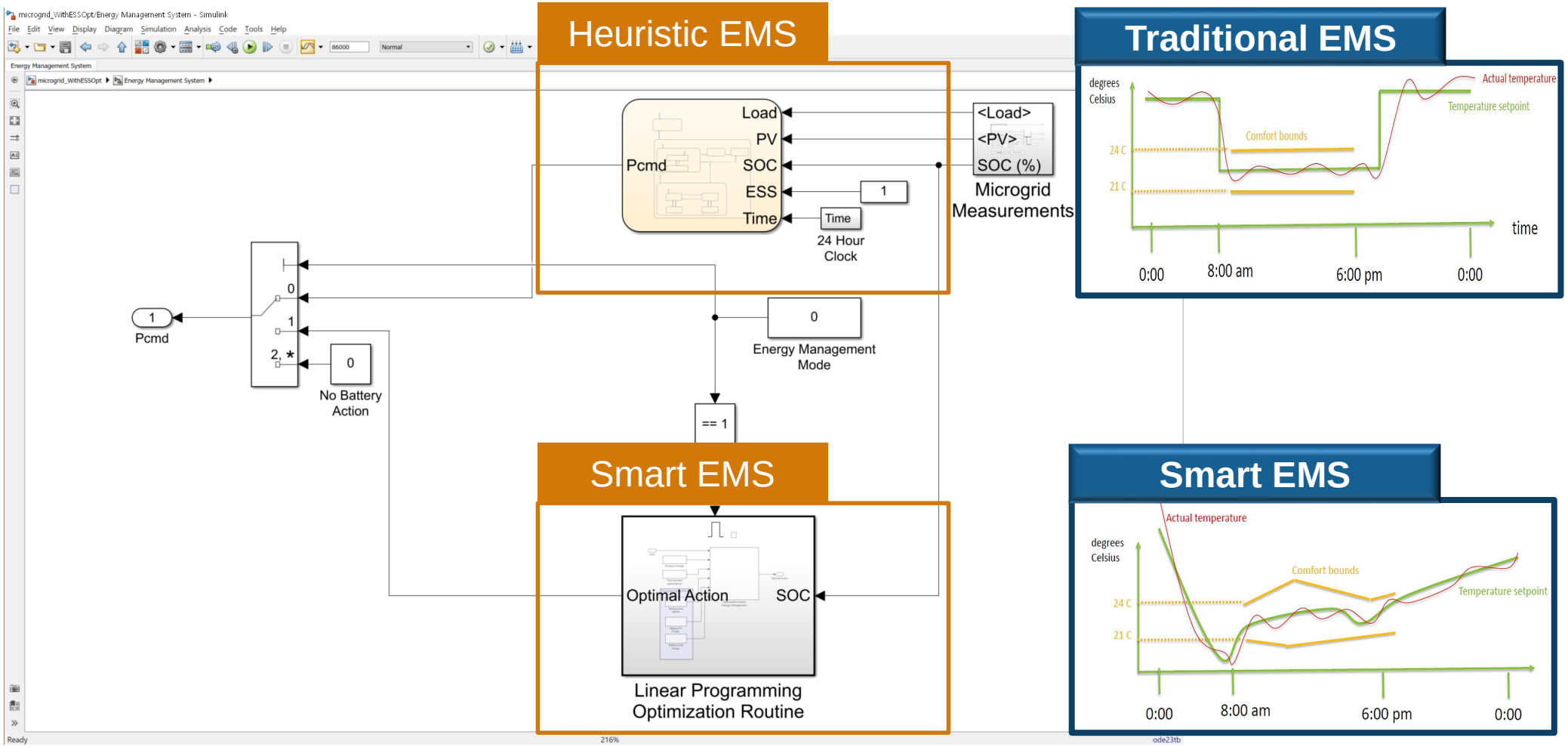
Solar Array



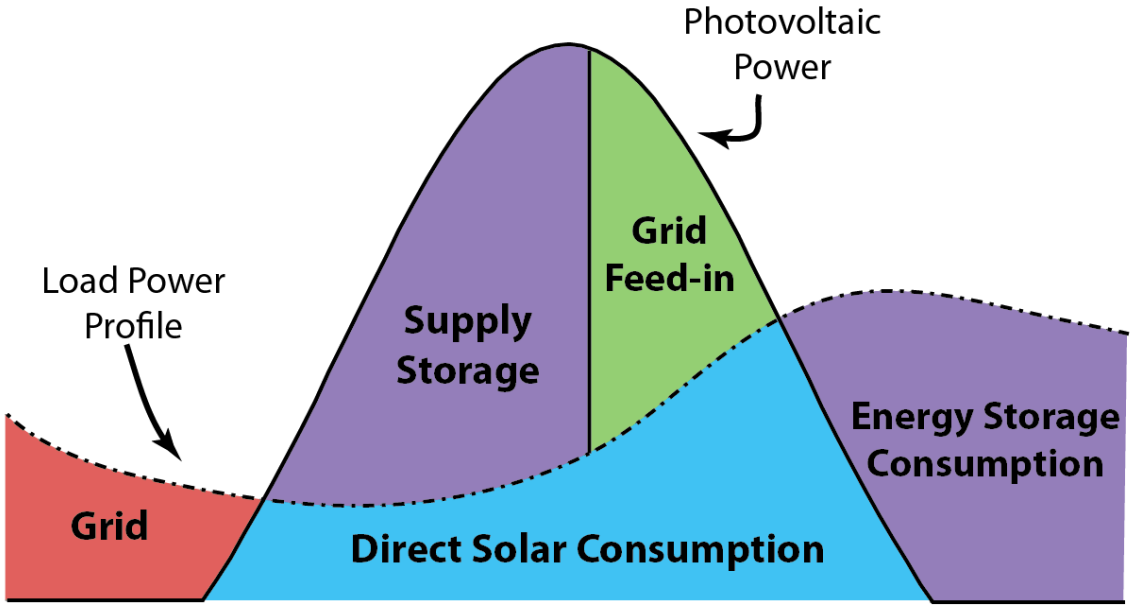




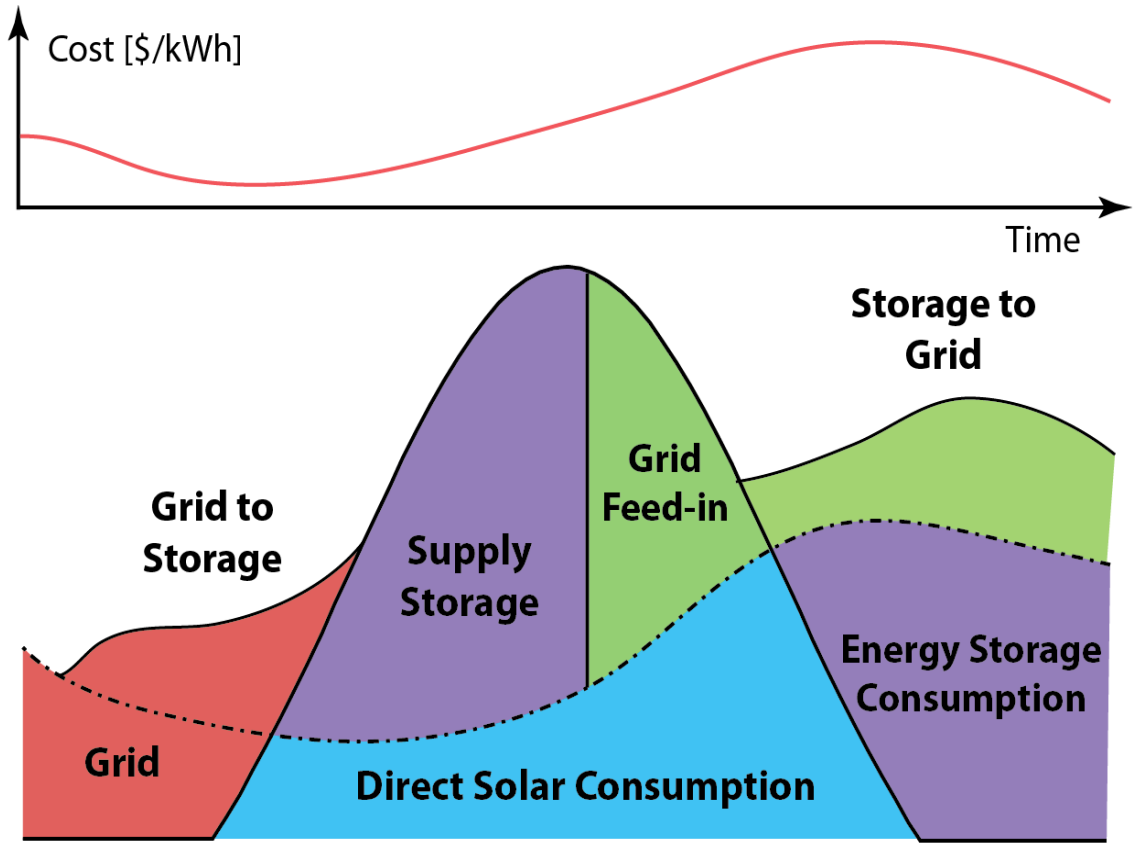
EMS Logic


Heuristic EMS
Load
PV
Pcmd
SOC
ESS
Time
1
Time
24 Hour Clock
<Load>
<PV>
SOC (%)
Microgrid Measurements
0
Energy Management Mode
1
Pcmd
0
1
2, *
0
No Battery Action
== 1
Smart EMS
Optimal Action
SOC
Linear Programming Optimization Routine
Traditional EMS
degrees Celsius
Actual temperature
Temperature setpoint
Comfort bounds
24 C
21 C
time
0:00
8:00 am
6:00 pm
0:00
Smart EMS
Actual temperature
degrees Celsius
Comfort bounds
Temperature setpoint
24 C
21 C
0:00
8:00 am
6:00 pm
0:00

EMS Logic



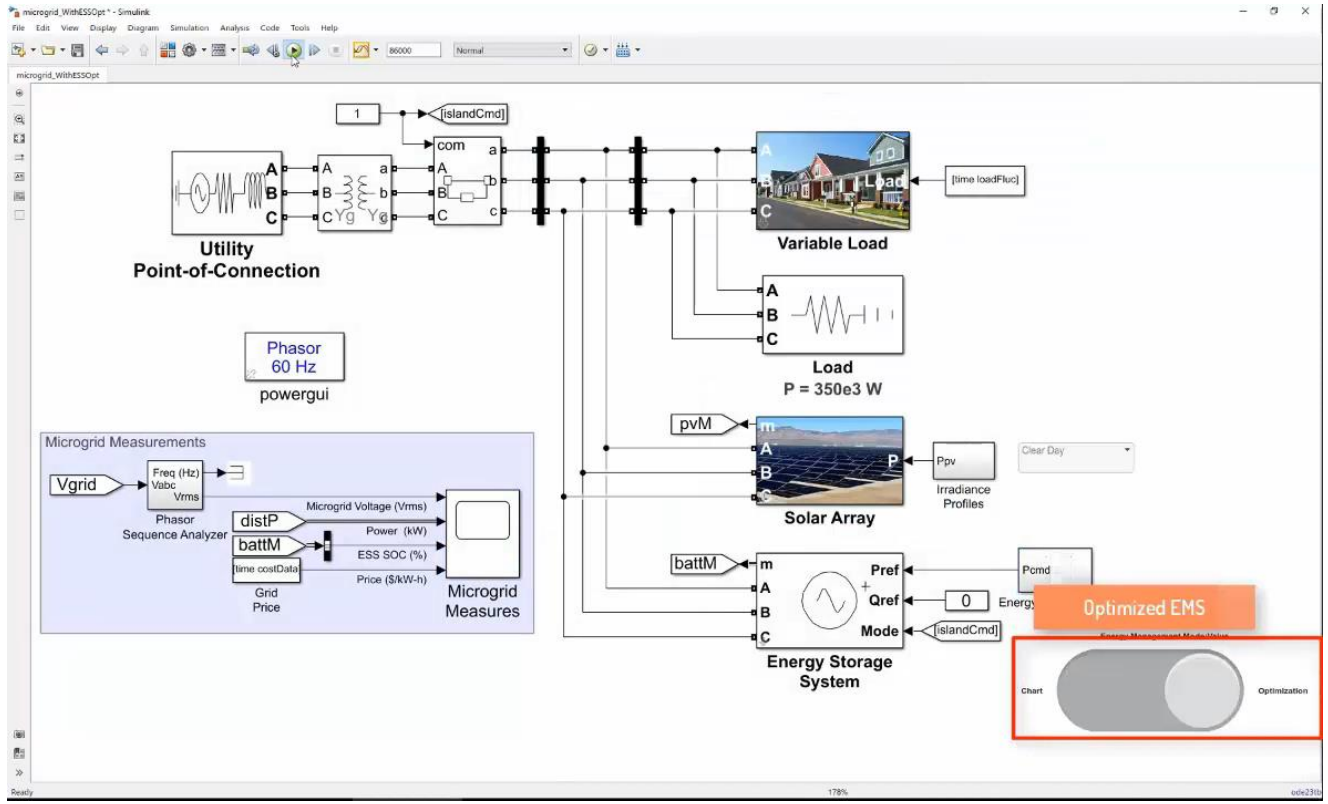
Heuristic EMS

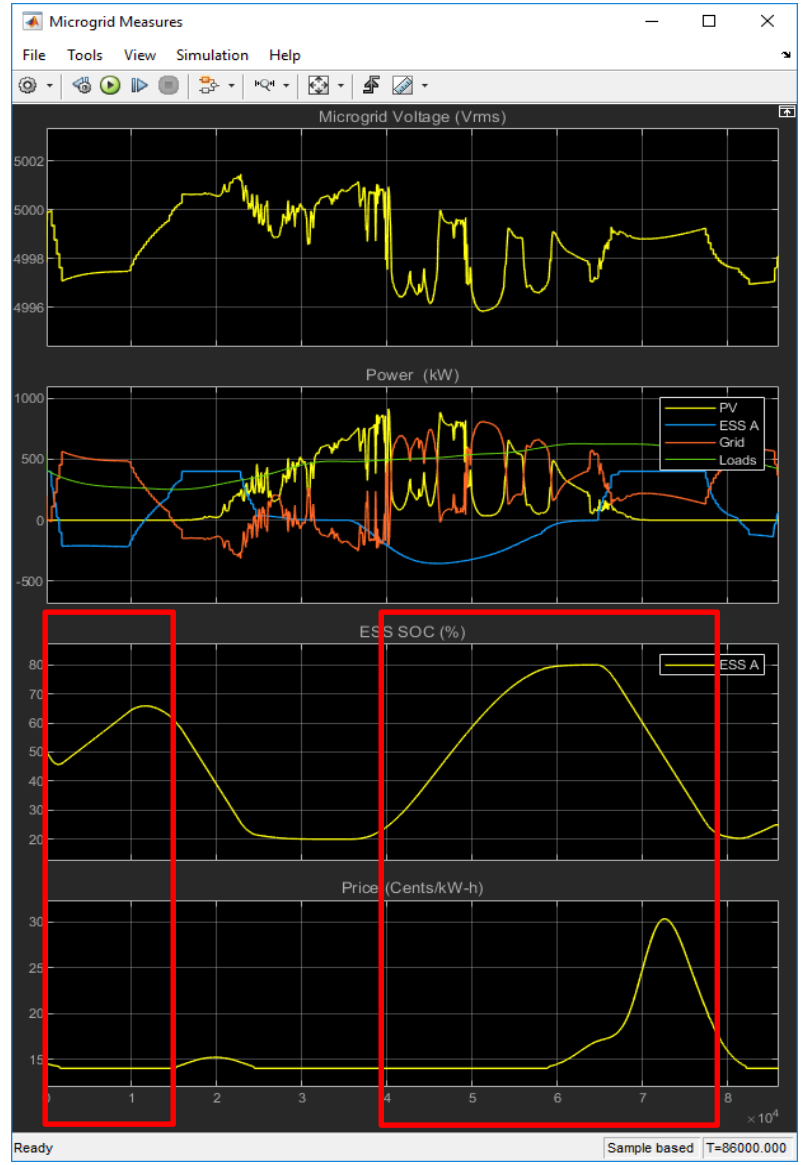


Smart EMS

Simulation Results

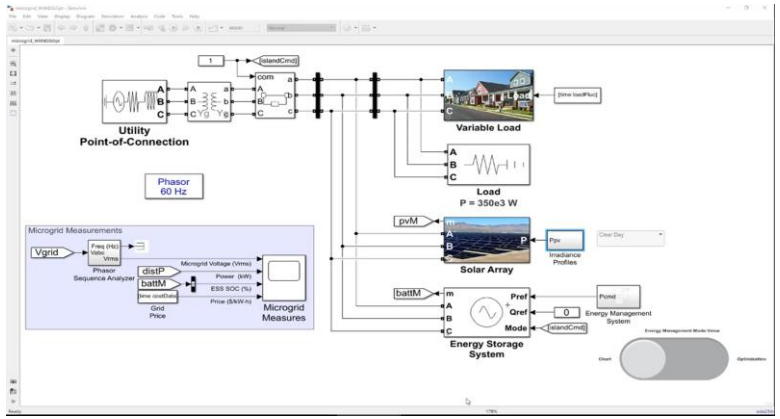
Smart EMS

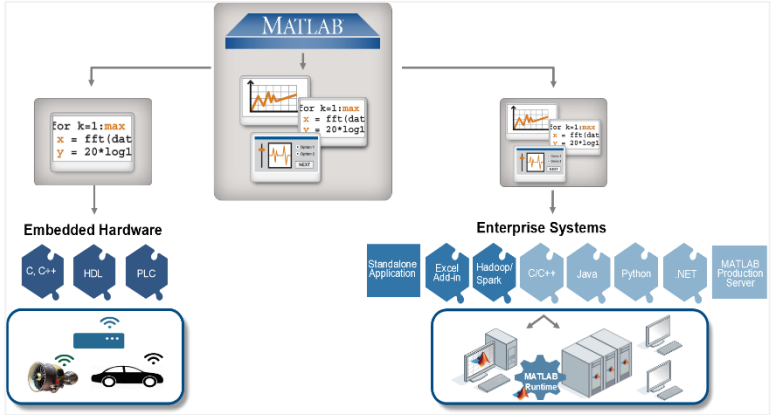




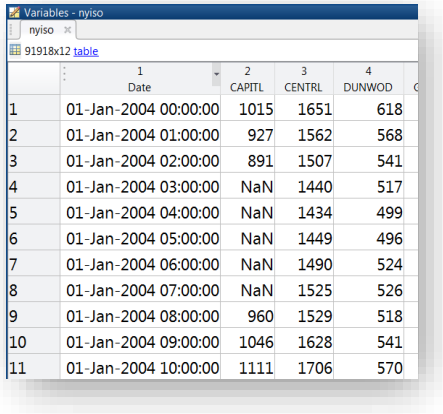
What do you need to build a smart EMS?

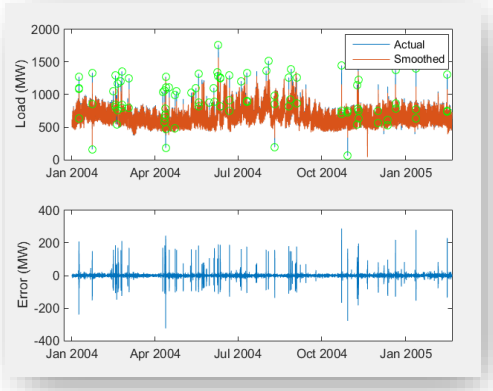
- Integrated development environment
  - Data analysis
  - Predictive modeling
  - Optimization
  - Control
  - System Design
- Virtual prototyping
- Deployment options
  - Deploy to embedded systems
  - Deploy to enterprise systems

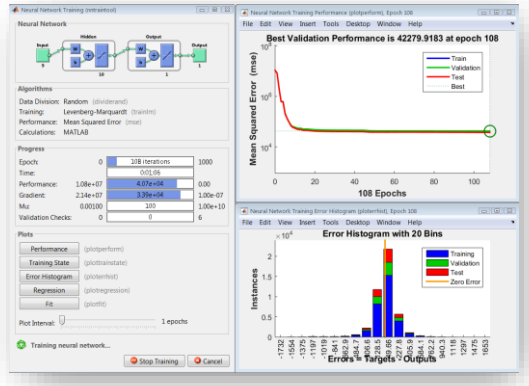


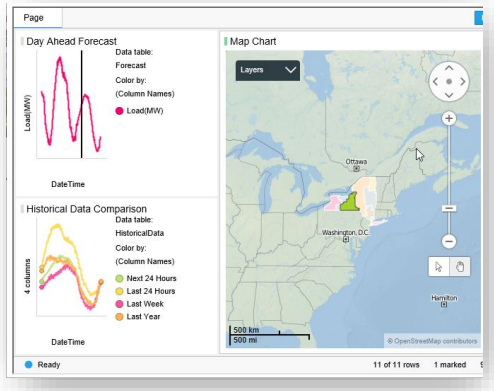


EMS Development Workflow



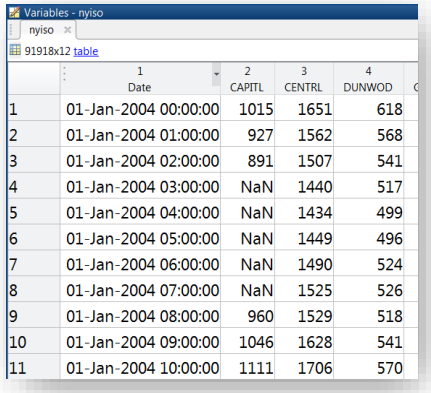


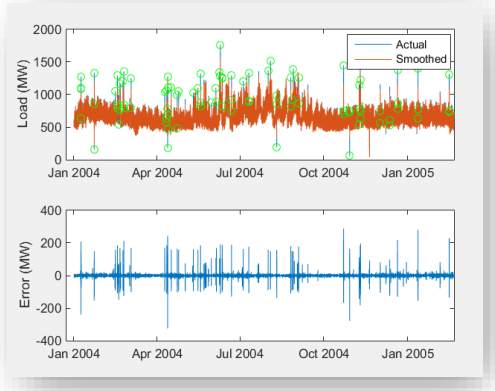


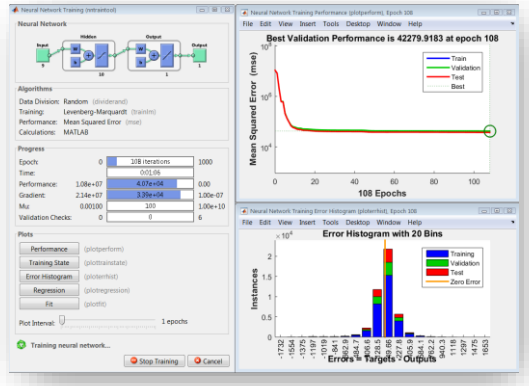


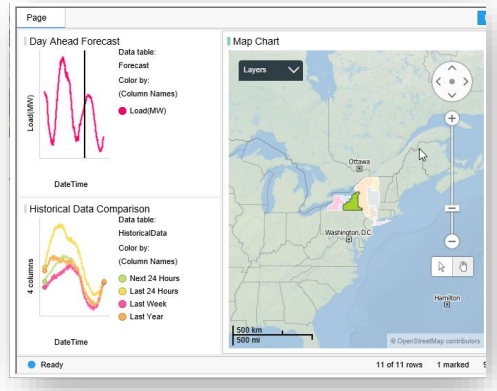
Access Data | **Analyze Data** | **Develop** | **Deploy**

EMS Development Workflow

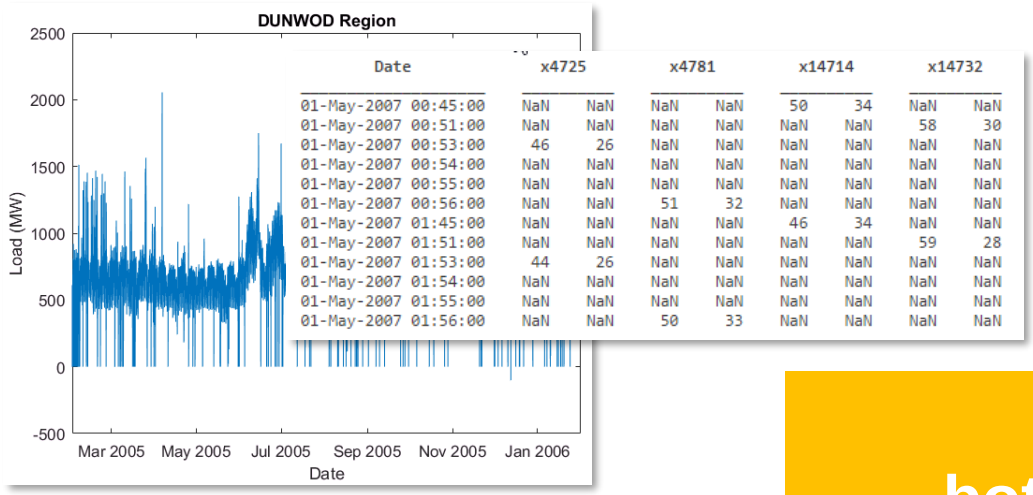








Access Data | **Analyze Data** | **Develop** | **Deploy**



Engineering Data

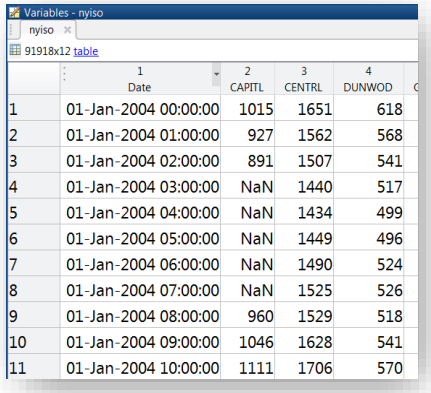
- Electric load: user's electrical load, cooling load, ...
- Sensor data: irradiance, wind speed, temperature, ...

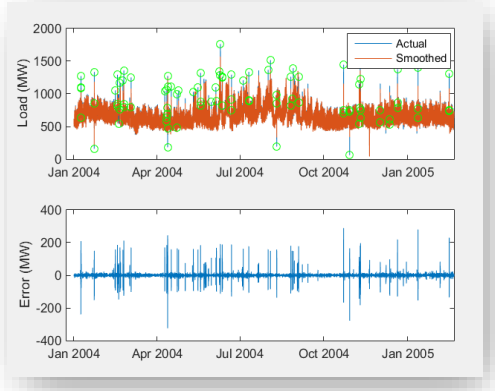
Business Data

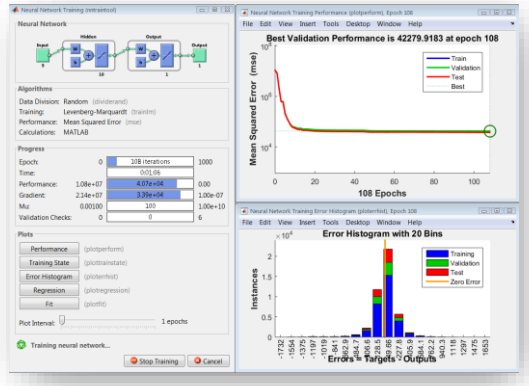
- Market information: electricity prices, gas prices, equipment costs, interest rates, ...

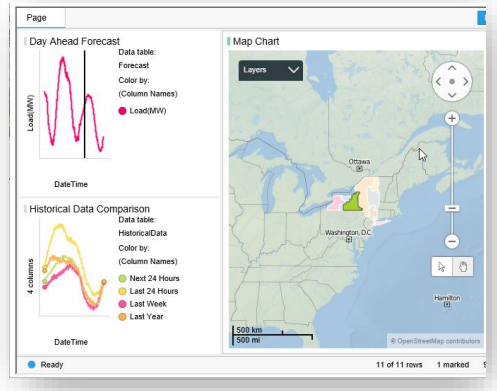
Need ways to access both business and engineering data

EMS Development Workflow

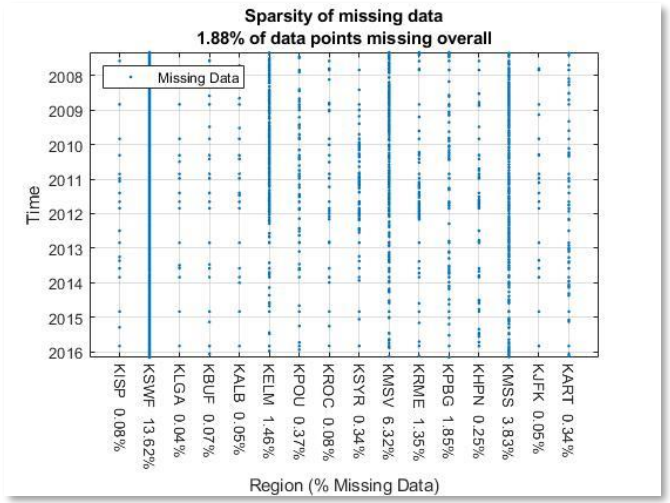








Access Data | **Analyze Data** | Develop | Deploy



Data Preprocessing Messy Data

- Missing data, outliers, sampling, ….
- Filtering and smoothing, resampling, …
- Join, stack, group, discretize

Need a powerful technical computing environment

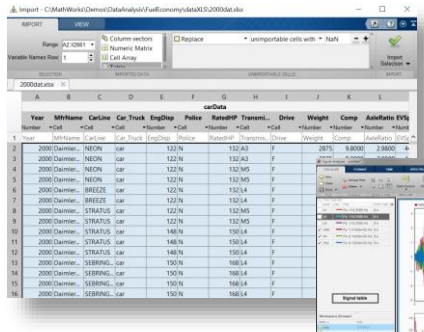
Access and Analyze Data

Access and Explore Data | **Analyze Data**

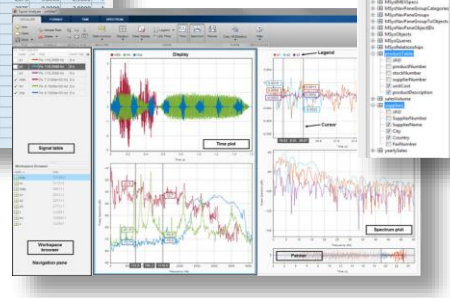
Files

Databases

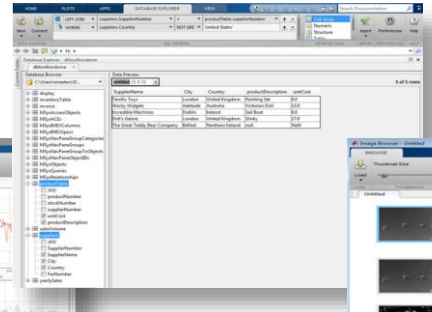
Sensors



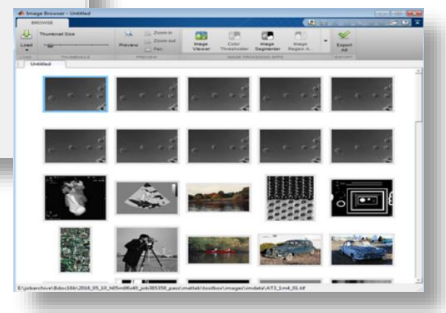
Files



Signals



Databases

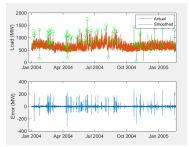


Images

- Point and click tools to access variety of data sources
- High-performance environment for big data

- Built-in algorithms for data preprocessing including sensor, image, audio, video and other real-time data

Working with Messy Data



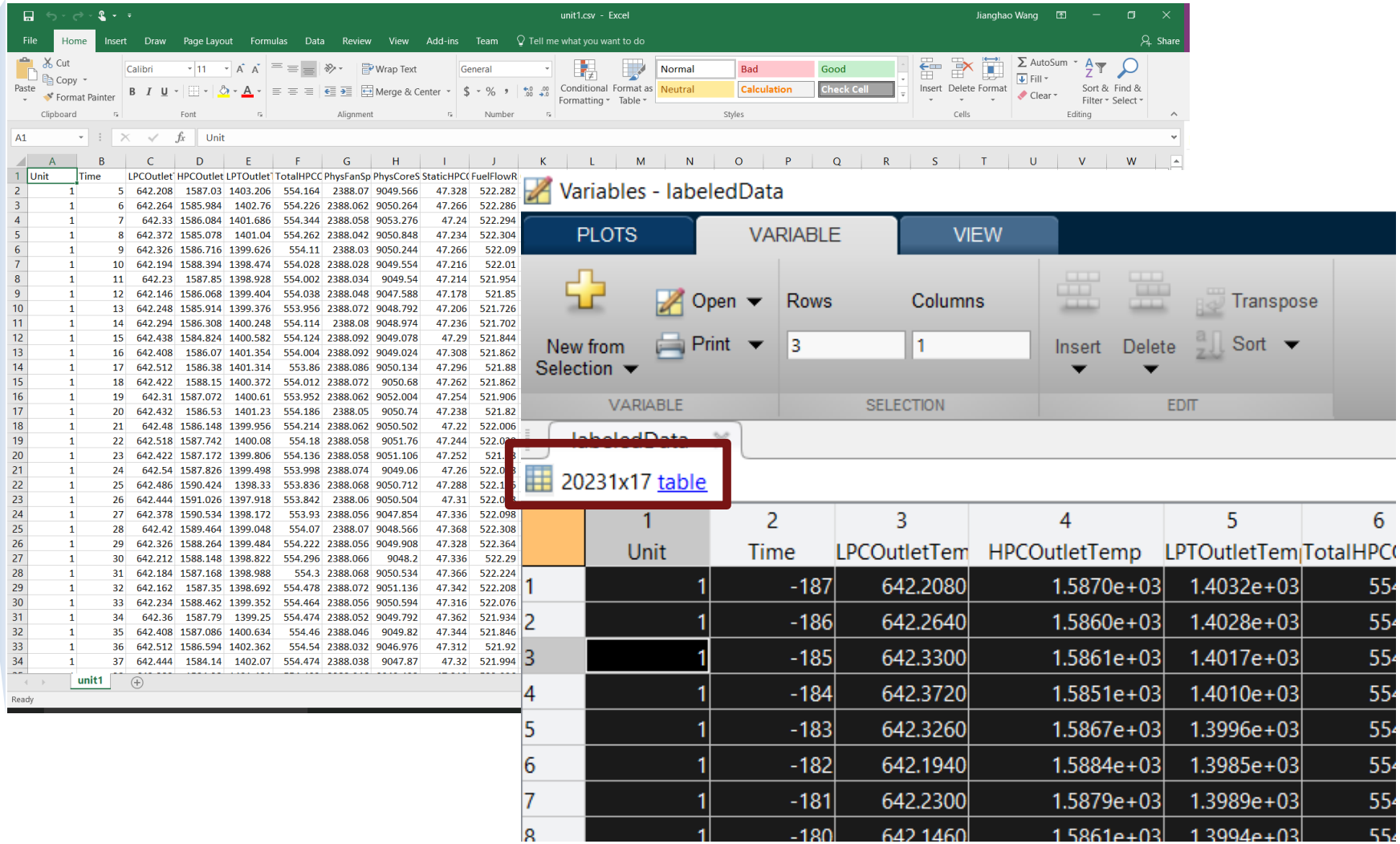
Data Reduction/ Transformation

Feature Extraction

Access Data

Name

unit81.csv



Make it easy to handle Data for Data Analytics

Table

- **For:**
  - Mixed-type tabular data
  - Include metadata
  - Time-stamped tabular data

- **Provides:**
  - Flexible indexing
  - Data organization
    - joins, stack/unstack, etc.
  - Indexing by time, time range, or within a tolerance around a time
  - Retiming to create a constant sample rate

Timetables

```
data(1:10,["begin_timestamp","state","event_type","event_narrative","damage_total"])
```

ans = 10×5 table

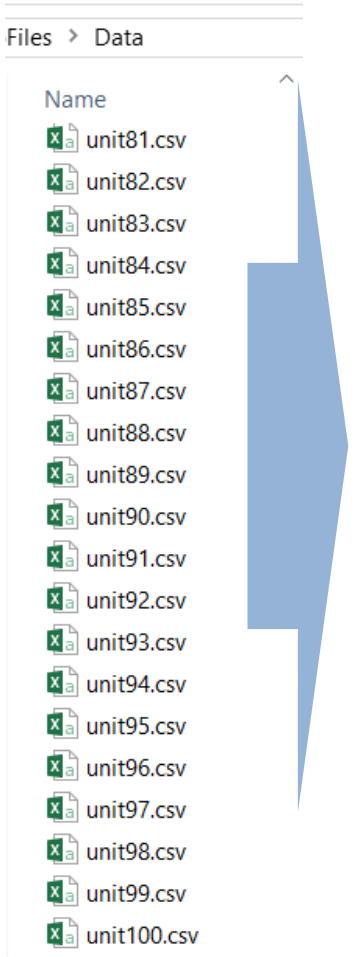
	begin_timestamp	state	event_type	event_narrative	damage_total
1	02-May-2003 13:55:00	ALABAMA	Hail	"tennis ball size hail…	0
2	20-Apr-1999 10:20:00	FLORIDA	Dust Devil	"a dust devil caused…	1
3	26-Mar-2014 04:17:00	MASSACH…	Blizzard	"the automated surf…	65
4	21-Oct-1996 14:00:00	CALIFORNIA	Wildfire	"santa ana winds av…	37400
5	08-May-2001 18:17:00	KANSAS	Tornado	"tornado entered mi…	0
6	15-Jun-2017 20:55:00	GEORGIA	Thunderstorm…	"the twiggs county 9…	6
7	27-Dec-2008 12:50:00	TEXAS	Strong Wind	"a large tree was do…	1
8	11-Oct-1997 21:10:00	COLORADO	Thunderstorm…	"the supercell thund…	0
9	29-May-2015 17:40:00	ARKANSAS	Thunderstorm…	"a tree fell on a hom…	25
10	27-Jan-2009 22:00:00	NEW YORK	Winter Storm	"a winter storm brou…	0

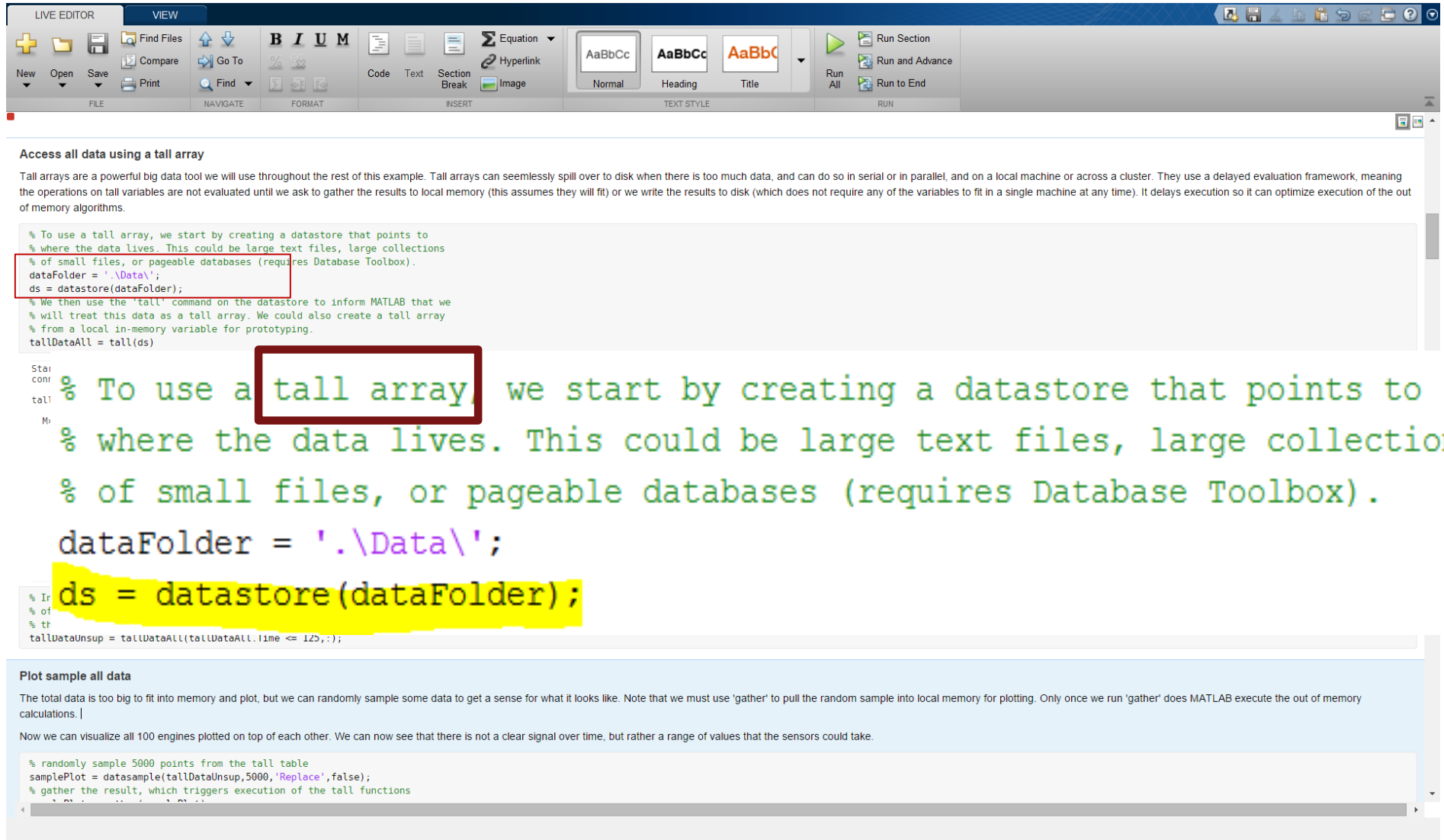
```
data(timerange("01-Jan-2017","17-Mar-2017"),:)
```

ans = 161×4 timetable

	begin_timestamp	state	event_type	event_narrative	damage_total
1	21-Jan-2017 13:02:00	GEORGIA	Thunderstorm…	"a tree was blown d…	0
2	21-Jan-2017 05:14:00	ALABAMA	Tornado	"the tornado first tou…	750
3	05-Jan-2017 04:00:00	OHIO	Winter Weather	"the county garage …	0
4	05-Mar-2017 18:00:00	OREGON	Snow	"there were reports …	0
5	04-Feb-2017 12:15:00	WYOMING	Wind	"the wydot sensor a…	0
6	08-Feb-2017 08:00:00	INDIANA	Winter Weather	"the observers locat…	0
7	18-Jan-2017 18:00:00	CALIFORNIA	Winter Weather	"a spotter in moonri…	0
8	07-Feb-2017 07:00:00	CALIFORNIA	Flood	"major flooding from…	0
9	13-Jan-2017 15:00:00	KANSAS	Ice Storm	"ice accretion was 3…	0
10	23-Jan-2017 00:00:00	NEW YORK	Wind	"a mesonet station…	50

Access Data


Files > Data
Name
unit81.csv
unit82.csv
unit83.csv
unit84.csv
unit85.csv
unit86.csv
unit87.csv
unit88.csv
unit89.csv
unit90.csv
unit91.csv
unit92.csv
unit93.csv
unit94.csv
unit95.csv
unit96.csv
unit97.csv
unit98.csv
unit99.csv
unit100.csv

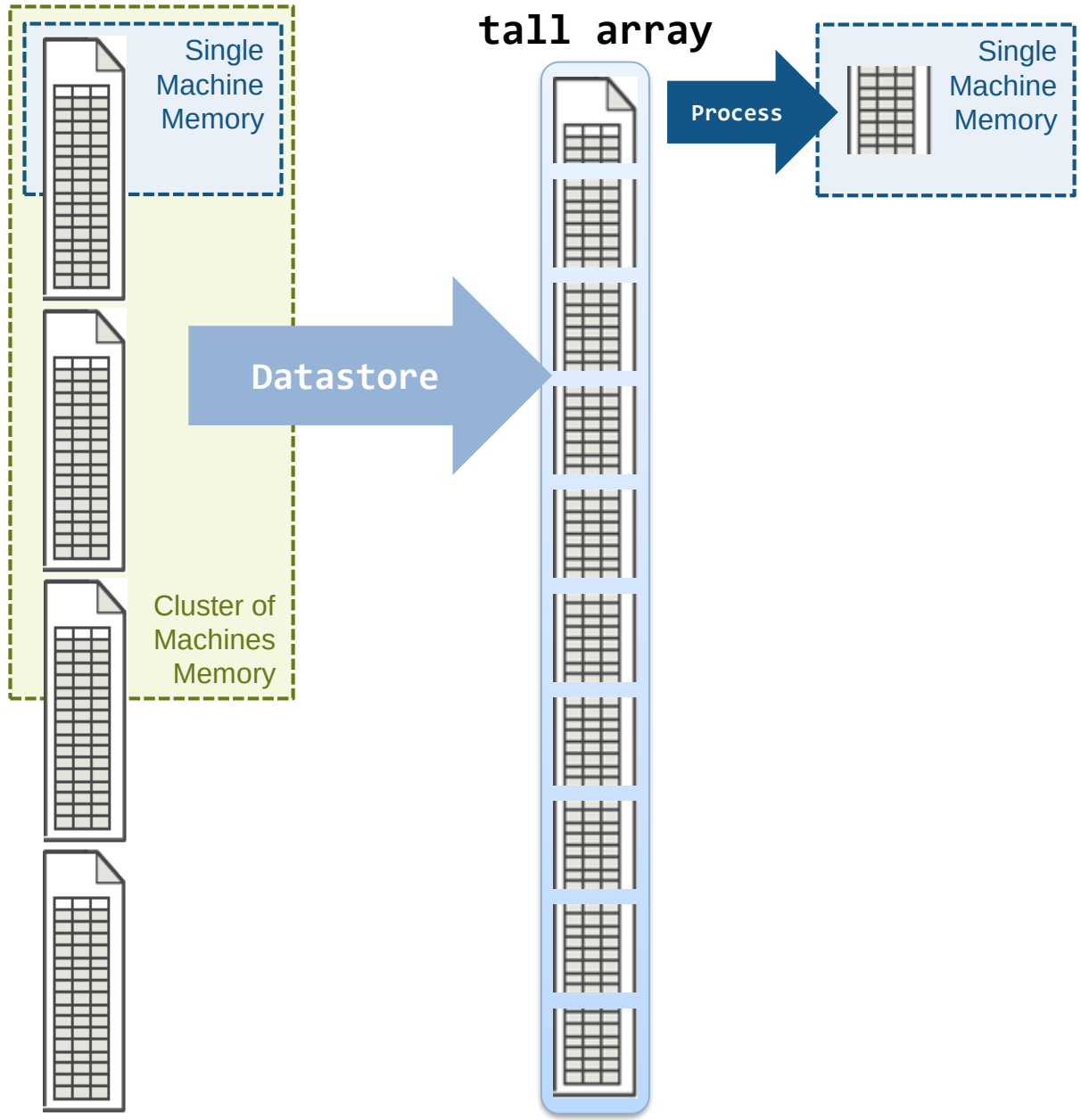

LIVE EDITOR
VIEW
New
Open
Save
Find Files
Compare
Print
Go To
Find
Code
Text
Section Break
Equation
Hyperlink
Image
Normal
Heading
Title
Run All
Run Section
Run and Advance
Run to End
FILE
NAVIGATE
FORMAT
INSERT
TEXT STYLE
RUN
Access all data using a tall array
Tall arrays are a powerful big data tool we will use throughout the rest of this example. Tall arrays can seemlessly spill over to disk when there is too much data, and can do so in serial or in parallel, and on a local machine or across a cluster. They use a delayed evaluation framework, meaning the operations on tall variables are not evaluated until we ask to gather the results to local memory (this assumes they will fit) or we write the results to disk (which does not require any of the variables to fit in a single machine at any time). It delays execution so it can optimize execution of the out of memory algorithms.
% To use a tall array, we start by creating a datastore that points to
% where the data lives. This could be large text files, large collections
% of small files, or pageable databases (requires Database Toolbox).
dataFolder = '.\Data\';
ds = datastore(dataFolder);
% We then use the 'tall' command on the datastore to inform MATLAB that we
% will treat this data as a tall array. We could also create a tall array
% from a local in-memory variable for prototyping.
tallDataAll = tall(ds)
% To use a tall array, we start by creating a datastore that points to
% where the data lives. This could be large text files, large collections
% of small files, or pageable databases (requires Database Toolbox).
dataFolder = '.\Data\';
ds = datastore(dataFolder);
Plot sample all data
The total data is too big to fit into memory and plot, but we can randomly sample some data to get a sense for what it looks like. Note that we must use 'gather' to pull the random sample into local memory for plotting. Only once we run 'gather' does MATLAB execute the out of memory calculations.
Now we can visualize all 100 engines plotted on top of each other. We can now see that there is not a clear signal over time, but rather a range of values that the sensors could take.
% randomly sample 5000 points from the tall table
samplePlot = datasample(tallDataUnsup,5000,'Replace',false);
% gather the result, which triggers execution of the tall functions

Tall Arrays

- Data is in one or more files
- Typically tabular data
- Files stacked vertically
- Data doesn't fit into memory
  (even cluster memory)

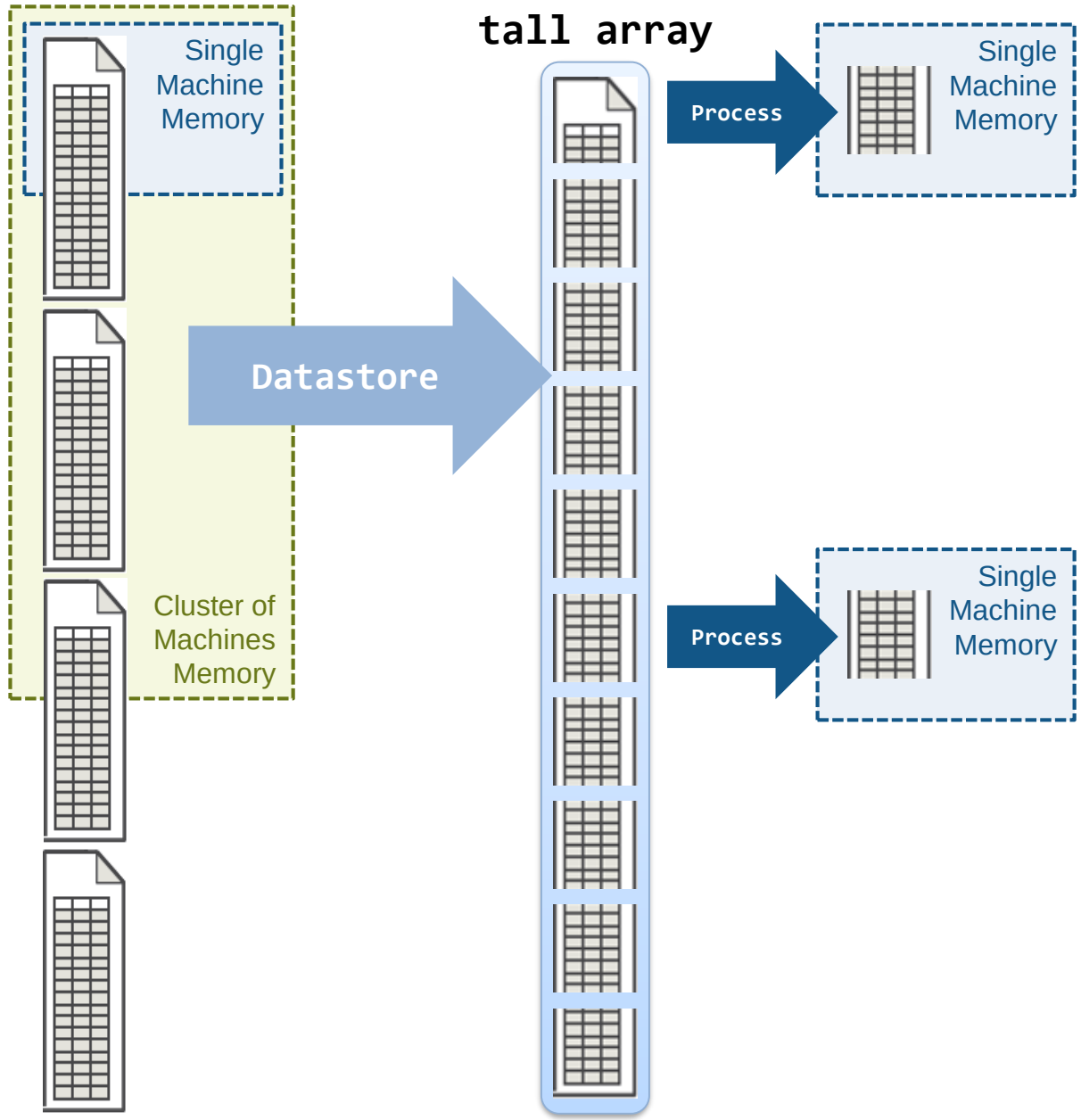
- Create tall table from datastore

```
ds = datastore('*.csv')
tt = tall(ds)
```



Tall Arrays

- With Parallel Computing Toolbox, process several pieces at once

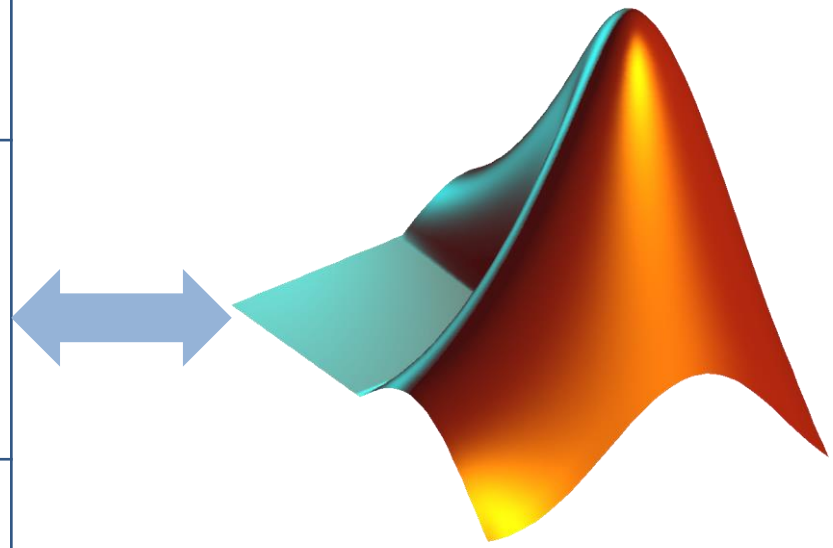


Enterprise Data Access

Data

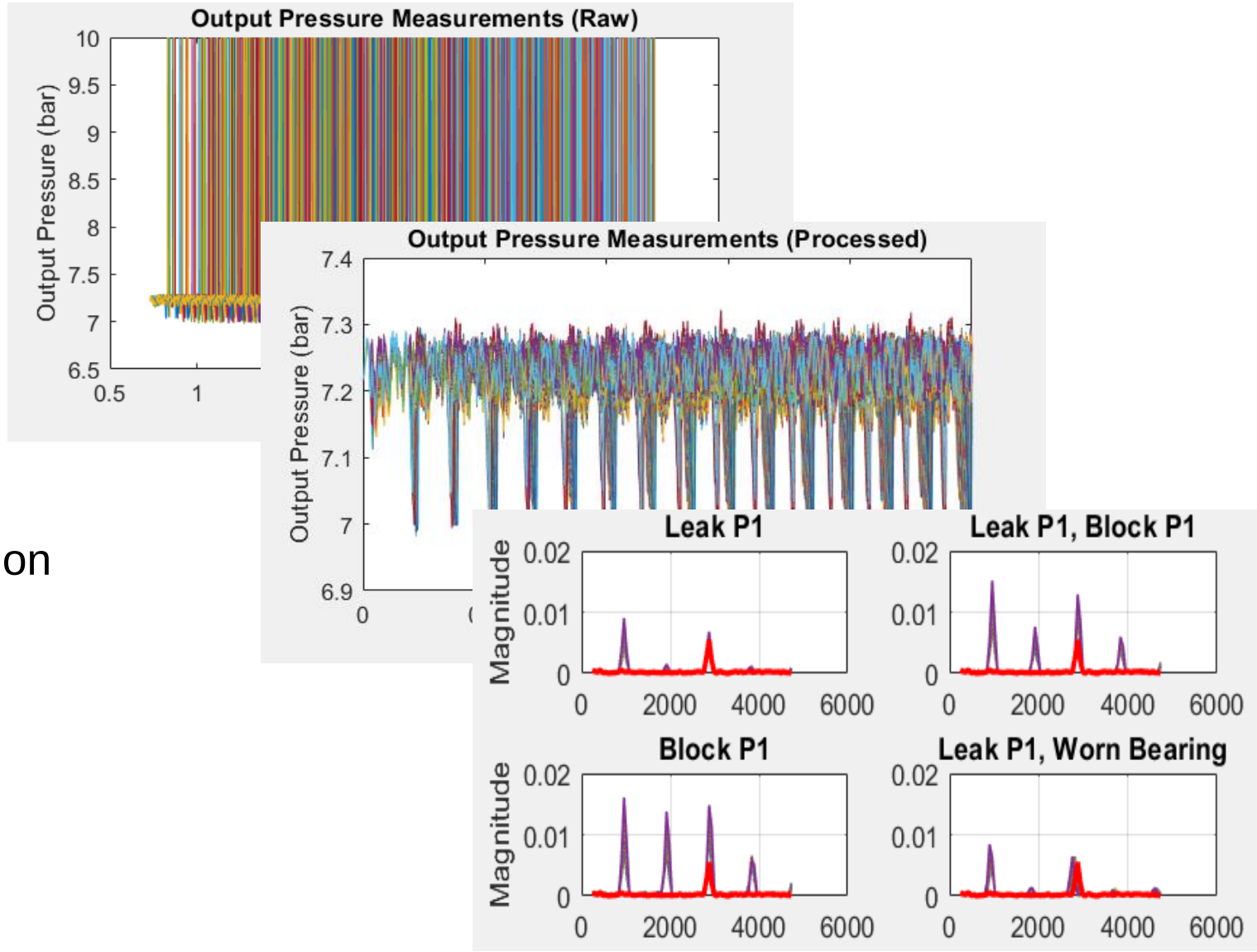
Databases	Cassandra, Microsoft SQL Server, mongoDB
Cloud Storage	Azure Blob, Azure SQL, amazon web services S3
IoT & Big Data	kafka, Azure IoT Hub, HORTONWORKS, cloudera

Analytics

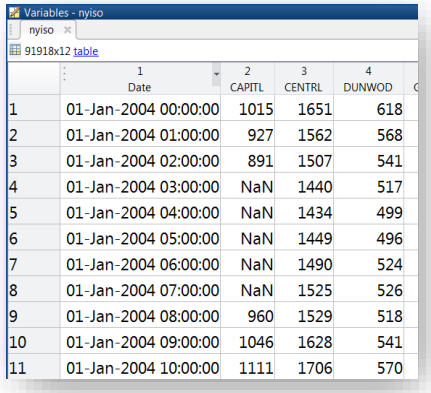


Analyze Data

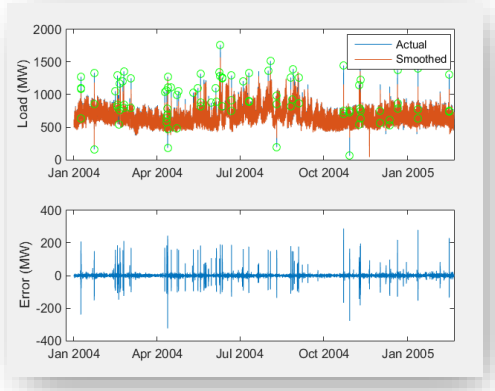
- Pre-processing
  - Data Cleaning
    - Missing Data
    - Merging Data
    - Outliners and Smoothing
    - Filtering
  - Normalization/Calibration
  - Aggregation/Resampling
  - Data Reduction/ Transformation
- Post-processing
  - Feature Extraction
  - Grouping
  - Regression
  - Classification

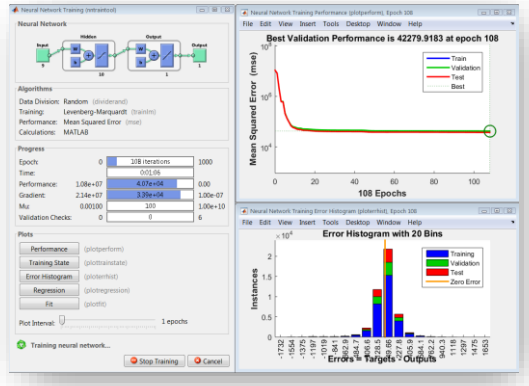


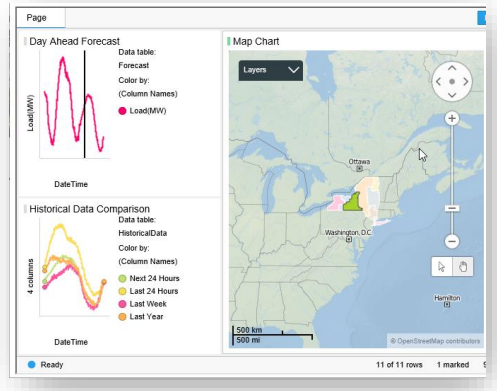
EMS Development Workflow



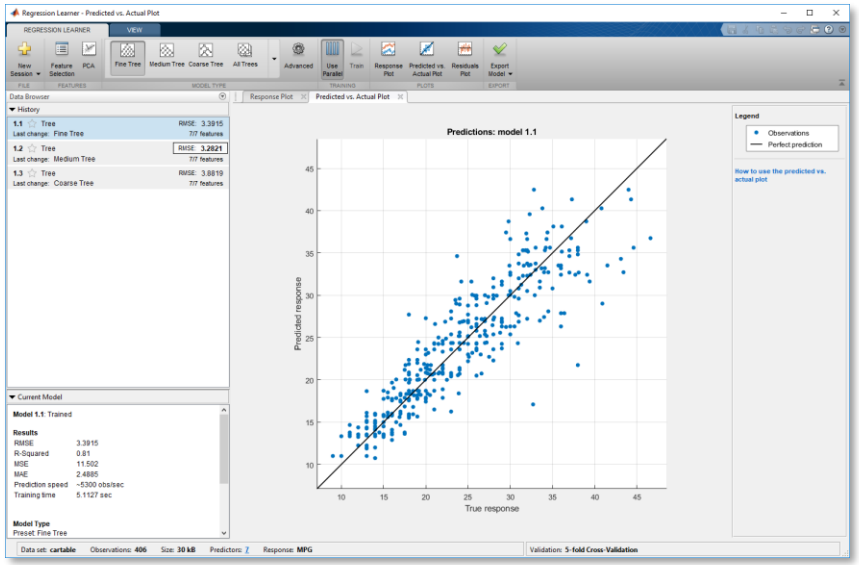
	1 Date	2 CAPITL	3 CENTRL	4 DUNWOD
1	01-Jan-2004 00:00:00	1015	1651	618
2	01-Jan-2004 01:00:00	927	1562	568
3	01-Jan-2004 02:00:00	891	1507	541
4	01-Jan-2004 03:00:00	NaN	1440	517
5	01-Jan-2004 04:00:00	NaN	1434	499
6	01-Jan-2004 05:00:00	NaN	1449	496
7	01-Jan-2004 06:00:00	NaN	1490	524
8	01-Jan-2004 07:00:00	NaN	1525	526
9	01-Jan-2004 08:00:00	960	1529	518
10	01-Jan-2004 09:00:00	1046	1628	541
11	01-Jan-2004 10:00:00	1111	1706	570







Access Data | **Analyze Data** | **Develop** | **Deploy**

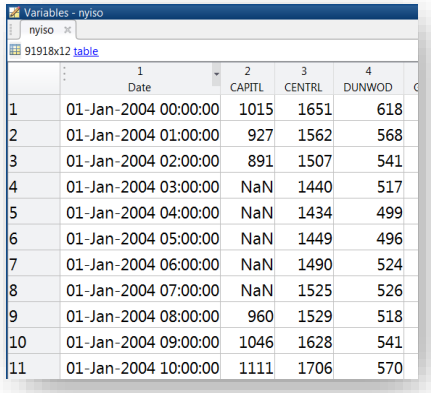


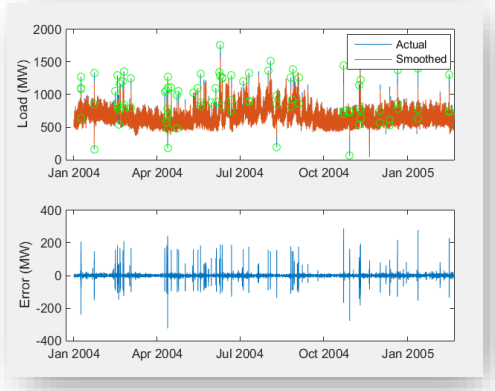
Predictive Models

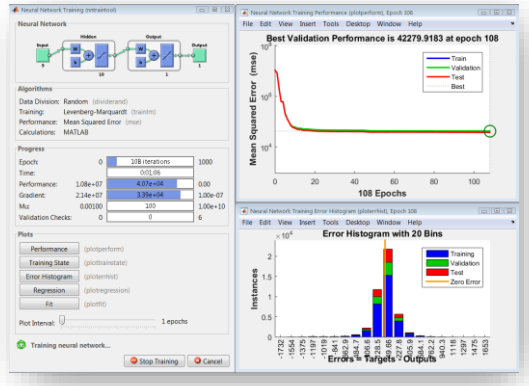
- Energy demand
- Electricity price
- Weather
- Consumer behavior

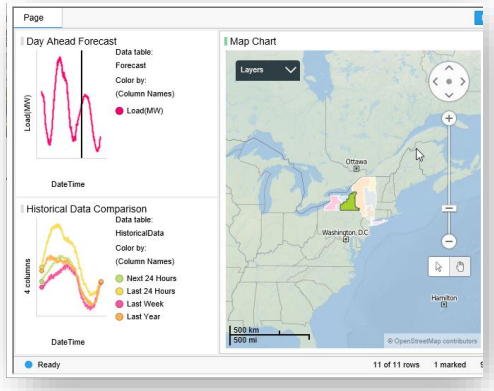
Need quick iteration of various predictive models

EMS Development Workflow

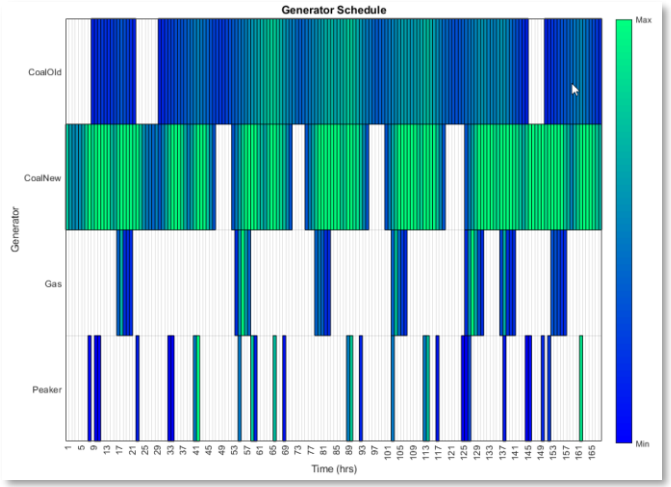








Access Data | **Analyze Data** | **Develop** | **Deploy**

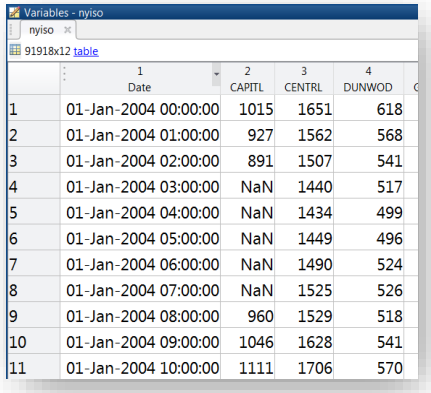


Optimization Models

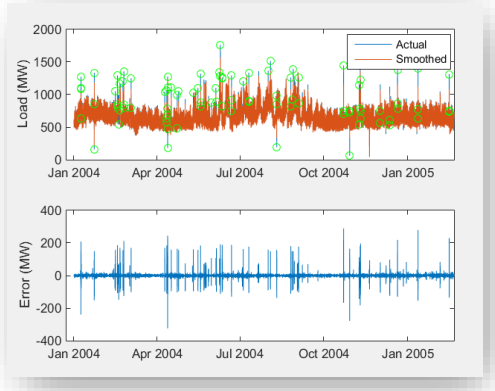
- Objectives on cost, comfort, reliability
- Constraints to meet demand, respect equipment and system limitations

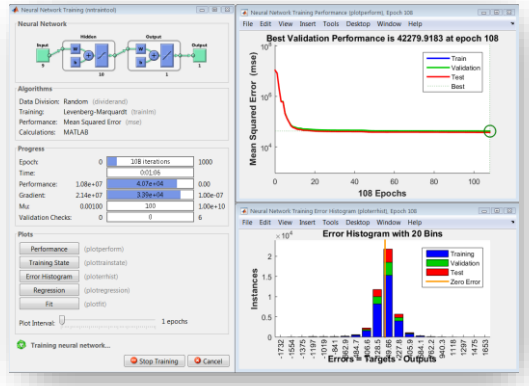
Need reliable optimization solvers

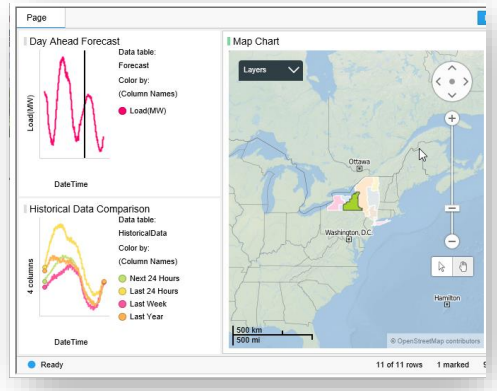
EMS Development Workflow



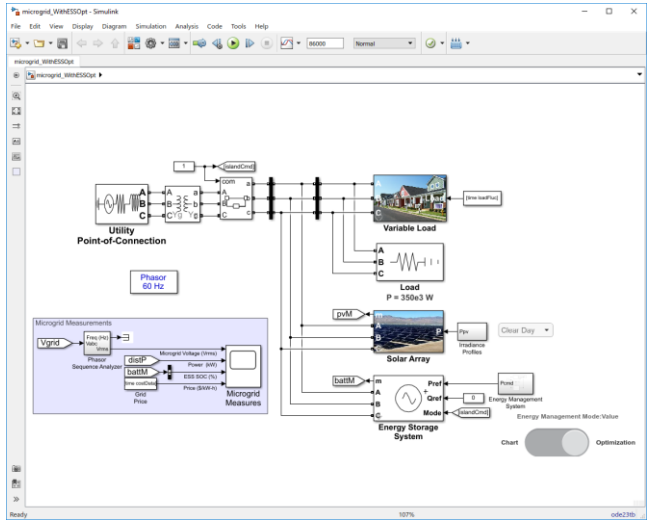
	1 Date	2 CAPITL	3 CENTRL	4 DUNWOD
1	01-Jan-2004 00:00:00	1015	1651	618
2	01-Jan-2004 01:00:00	927	1562	568
3	01-Jan-2004 02:00:00	891	1507	541
4	01-Jan-2004 03:00:00	NaN	1440	517
5	01-Jan-2004 04:00:00	NaN	1434	499
6	01-Jan-2004 05:00:00	NaN	1449	496
7	01-Jan-2004 06:00:00	NaN	1490	524
8	01-Jan-2004 07:00:00	NaN	1525	526
9	01-Jan-2004 08:00:00	960	1529	518
10	01-Jan-2004 09:00:00	1046	1628	541
11	01-Jan-2004 10:00:00	1111	1706	570







Access Data | Analyze Data | **Develop** | Deploy



System Simulation Models

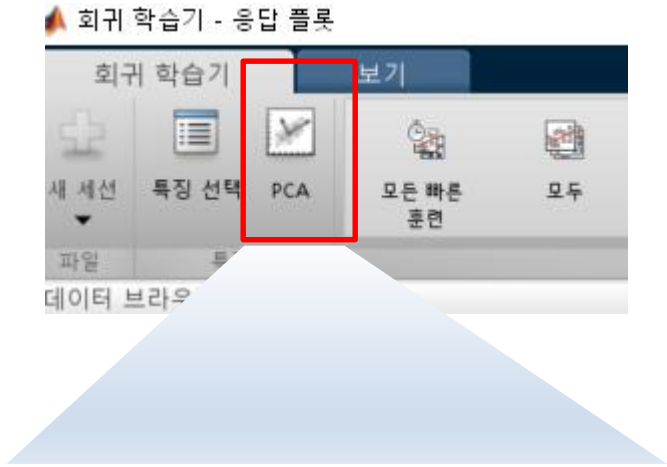
- Plan strategies of system function and performance on desktop
- Simulate physical system performance with high fidelity
  - Analyze edge conditions
- Simulate multiple scenarios quickly with low fidelity
  - Perform statistical analysis on results

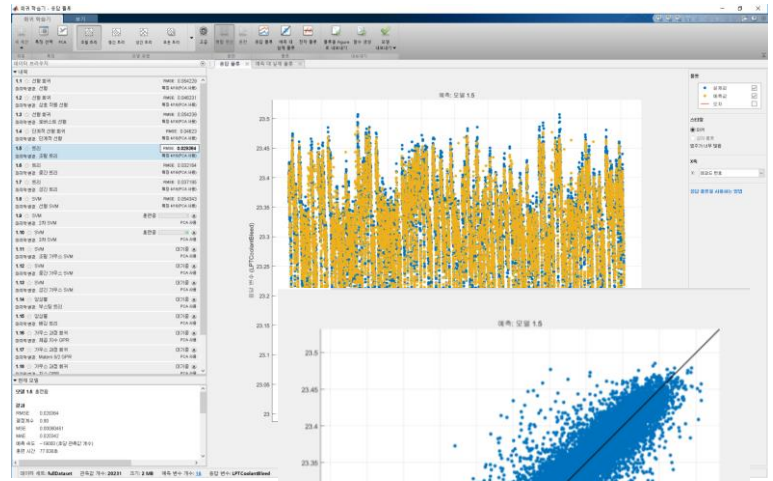
Need virtual prototyping environment

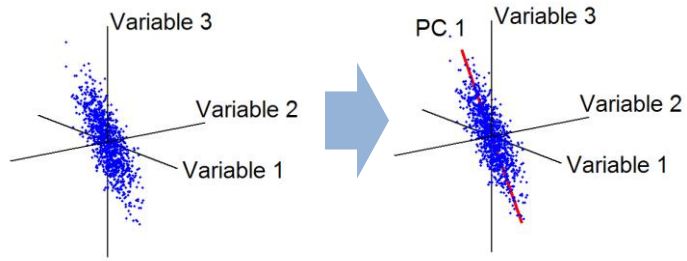
Prediction Example: Energy Demand

- Make prediction model using the pattern of energy demand with the data of the grid → Regression
- Find the important variables for the prediction of energy demand

- Important variables:
  - Customer behavior
  - Temperature
  - Price
  - Illumination
  - Hour, holiday, month
  - …



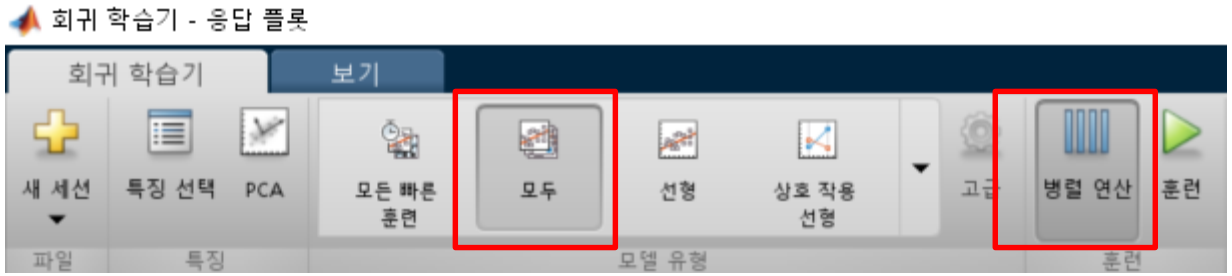




Evaluate all Regression Models

Finding the best predition model

- Train all models using training data and compare accuracy of each one
  - Trainings can run in parallel
- Multiple methods to assess accuracy



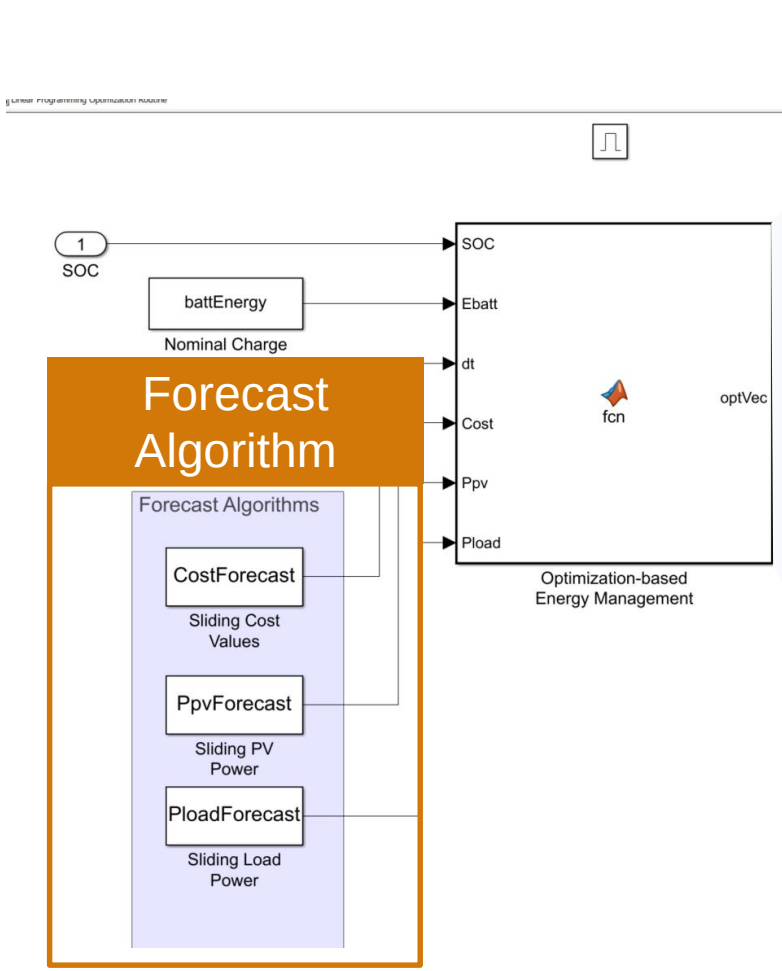
Regression Models

데이터 브라우저

내역

1.1 선형 회귀 마지막 변경: 선형	RMSE: 0.054229 특징 4/16(PCA 사용)
1.2 선형 회귀 마지막 변경: 상호 작용 선형	RMSE: 0.046231 특징 4/16(PCA 사용)
1.3 선형 회귀 마지막 변경: 로버스트 선형	RMSE: 0.054239 특징 4/16(PCA 사용)
1.4 단계적 선형 회귀 마지막 변경: 단계적 선형	RMSE: 0.04623 특징 4/16(PCA 사용)
1.5 트리 마지막 변경: 조밀 트리	RMSE: 0.028364 특징 4/16(PCA 사용)
1.6 트리 마지막 변경: 중간 트리	RMSE: 0.032164 특징 4/16(PCA 사용)
1.7 트리 마지막 변경: 성긴 트리	RMSE: 0.037195 특징 4/16(PCA 사용)
1.8 SVM 마지막 변경: 선형 SVM	RMSE: 0.054343 특징 4/16(PCA 사용)
1.9 SVM 마지막 변경: 2차 SVM	RMSE: 0.046012 특징 4/16(PCA 사용)
1.10 SVM 마지막 변경: 3차 SVM	RMSE: 0.047342 특징 4/16(PCA 사용)
1.11 SVM 마지막 변경: 조밀 가우스 SVM	RMSE: 0.042549 특징 4/16(PCA 사용)
1.12 SVM 마지막 변경: 중간 가우스 SVM	RMSE: 0.045173 특징 4/16(PCA 사용)
1.13 SVM 마지막 변경: 성긴 가우스 SVM	RMSE: 0.049373 특징 4/16(PCA 사용)
1.14 앙상블 마지막 변경: 부스팅 트리	RMSE: 0.98836 특징 4/16(PCA 사용)
1.15 앙상블 마지막 변경: 배깅 트리	RMSE: **0.027709** 특징 4/16(PCA 사용)

Optimization: Community EMS with PV and Battery



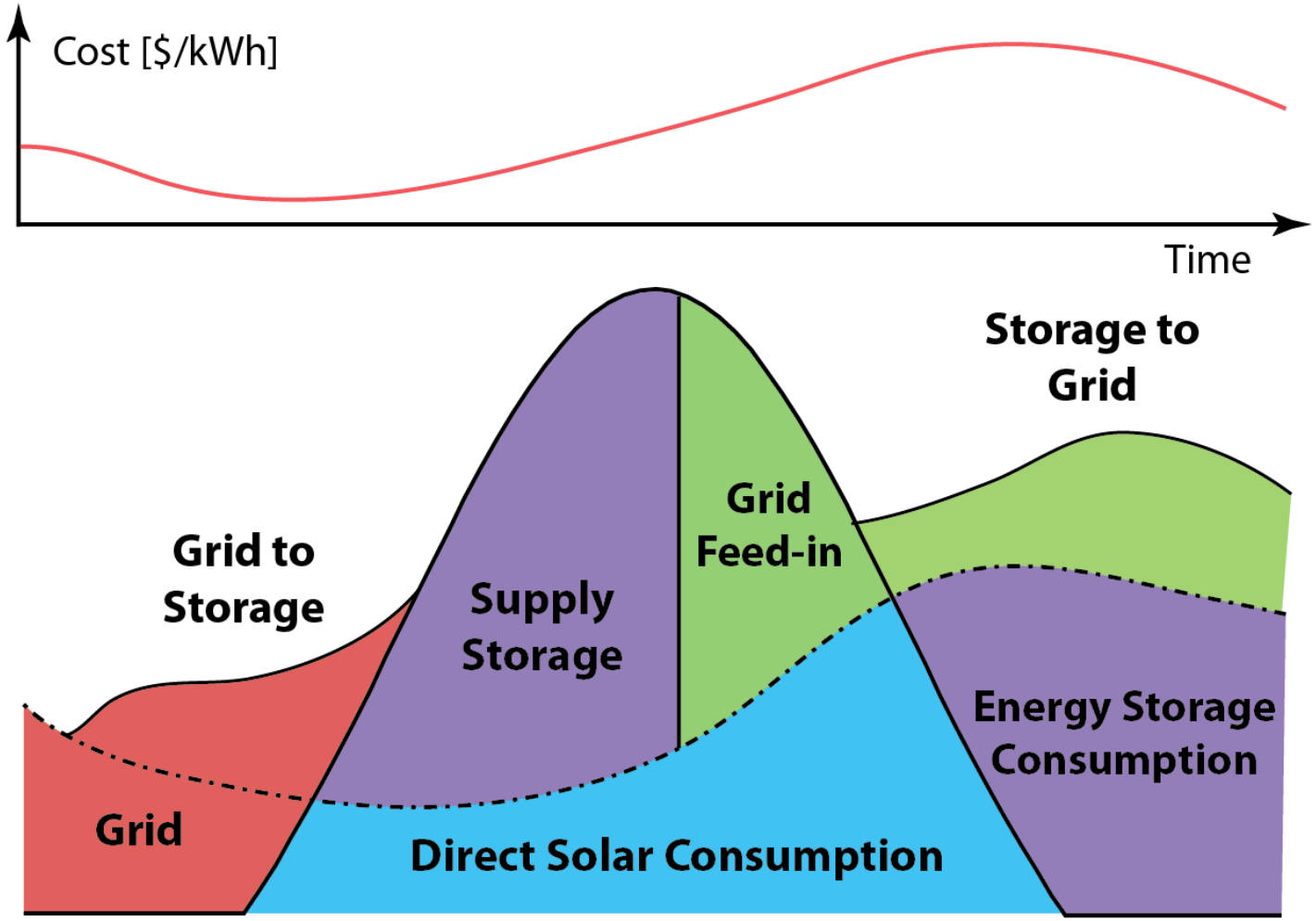

```
function optVec  = fcn(SOC,Ebatt,dt,Cost,Ppv,Pload)
%#codegen

M = 241;
optVec = 0;
Pbatt = zeros(M,1);
EnergyWeight = 1;

% Energy constraints
batteryMinMax.Emax = 0.8*Ebatt
batteryMinMax.Emin = 0.2*Ebatt

% Power constraints
batteryMinMax.Pmin = -400e3;
batteryMinMax.Pmax = 400e3;

% Declare function (contains
coder.extrinsic('battSolarOpti

% Battery parameters
Einit = SOC/100*Ebatt;

[~,Pbatt,~] = battSolarOptimiz
    EnergyWeight,batteryMinMax

optVec = mean(Pbatt(1:5));
```

Editor - C:\Demos\Microgrid_EMS\battSolarOptimize.m

battSolarOptimize.m

```
function [Pgrid,Pbatt,Ebatt] = battSolarOptimize(N,dt,Ppv,Pload,Einit,Cost,FinalWeight,batteryMinMax)

% Minimize the cost of power from the grid while meeting load with power
% from PV, battery and grid

prob = optimproblem;

% Decision variables
PgridV = optimvar('PgridV',N);
PbattV = optimvar('PbattV',N,'LowerBound',batteryMinMax.Pmin,'UpperBound',batteryMinMax.Pmax);
EbattV = optimvar('EbattV',N,'LowerBound',batteryMinMax.Emin,'UpperBound',batteryMinMax.Emax);
PgridDelta = optimvar('gridDelta',N-1);

% Minimize cost of electricity from the grid
% - Account for final battery storage
% - Smooth period-to-period changes with a penalty
prob.ObjectiveSense = 'minimize';
prob.Objective = dt*Cost'*PgridV - FinalWeight*EbattV(N) + sum(PgridDelta);

% Power input/output to battery
prob.Constraints.energyBalance = optimconstr(N);
prob.Constraints.energyBalance(1) = EbattV(1) == Einit;
prob.Constraints.energyBalance(2:N) = EbattV(2:N) == EbattV(1:N-1) - PbattV(1:N-1)*dt;

% Satisfy power load with power from PV, grid and battery
prob.Constraints.loadBalance = Ppv + PgridV + PbattV == Pload;

% Track change from period to period in electricity from the grid
prob.Constraints.deltaPlus = PgridV(2:N) - PgridV(1:N-1) <= PgridDelta;
prob.Constraints.deltaMinus = PgridV(1:N-1) - PgridV(2:N) <= PgridDelta;

% Solve the linear program
options = optimoptions(prob.optimoptions,'Display','none');
[values,~,exitflag] = solve(prob,'Options',options);

% Parse optmization results
if exitflag <= 0
    Pgrid = zeros(N,1);
    Pbatt = zeros(N,1);
    Ebatt = zeros(N,1);
else
    Pgrid = values.PgridV;
    Pbatt = values.PbattV;
    Ebatt = values.EbattV;
end
```

Optimization: Community EMS with PV and Battery



Optimization: Community EMS with PV and Battery

minimize $\sum_{t=1}^{N} \delta c_t G_t - wE_N + \sum_{t=1}^{N-1} g_t$

subject to

$$E_1 = E_{initial}$$

$$E_{t+1} = E_t - \delta B_t$$

$$s_t + G_t + B_t = d_t$$

$$G_{t+1} - G_t \leq g_t$$

$$G_t - G_{t+1} \leq g_t$$

G_t	Power from grid
$l_B \leq B_t \leq u_B$	Power from battery
$l_E \leq E_t \leq u_E$	Stored energy
g_t	Change in power from grid

Editor - C:\Demos\Microgrid_EMS\battSolarOptimize.m

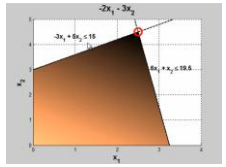
battSolarOptimize.m

```
function [Pgrid,Pbatt,Ebatt] = battSolarOptimize(N,dt,Ppv,Pload,Einit,Cost,FinalWeight,batteryMinMax)

% - Smooth period-to-period changes with a penalty
prob.ObjectiveSense = 'minimize';
prob.Objective = dt*Cost'*PgridV - FinalWeight*EbattV(N) + sum(PgridDelta);

% Satisfy power load with power from PV, grid and battery
prob.Constraints.loadBalance = Ppv + PgridV + PbattV == Pload;

% Solve the linear program
options = optimoptions(prob.optimoptions,'Display','none');
[values,~,exitflag] = solve(prob,'Options',options);
```

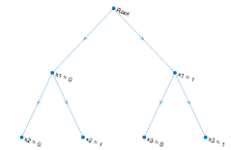
Solving: Problem Types and Algorithms

Optimization Toolbox

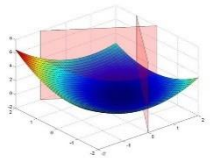
Global Optimization Toolbox



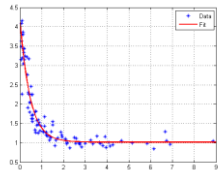
- Linear programming
  - Simplex and interior point



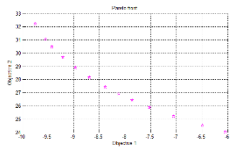
- Mixed-integer linear programming
  - Branch and cut



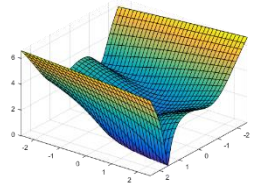
- Quadratic programming
  - Interior point and trust region



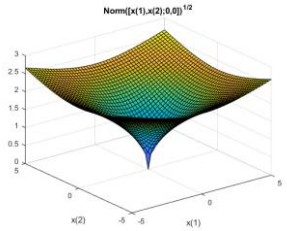
- Least-squares and nonlinear equations
  - Interior point, trust region, Levenberg-Marquardt

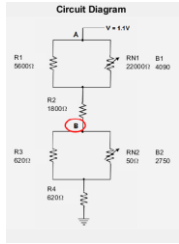


- Multiobjective optimization
  - Weighted and goal-attainment
  - *Genetic algorithm*
  - *Pattern (direct) search*



- Nonlinear optimization
  - Interior point
  - SQP
  - Trust region
  - Nelder-Mead simplex
  - *MultiStart & GlobalSearch*
  - *Pattern (direct) search*
  - *Genetic algorithm*
  - *Simulated annealing*
  - *Particle swarm*
  - *Surrogate optimization*





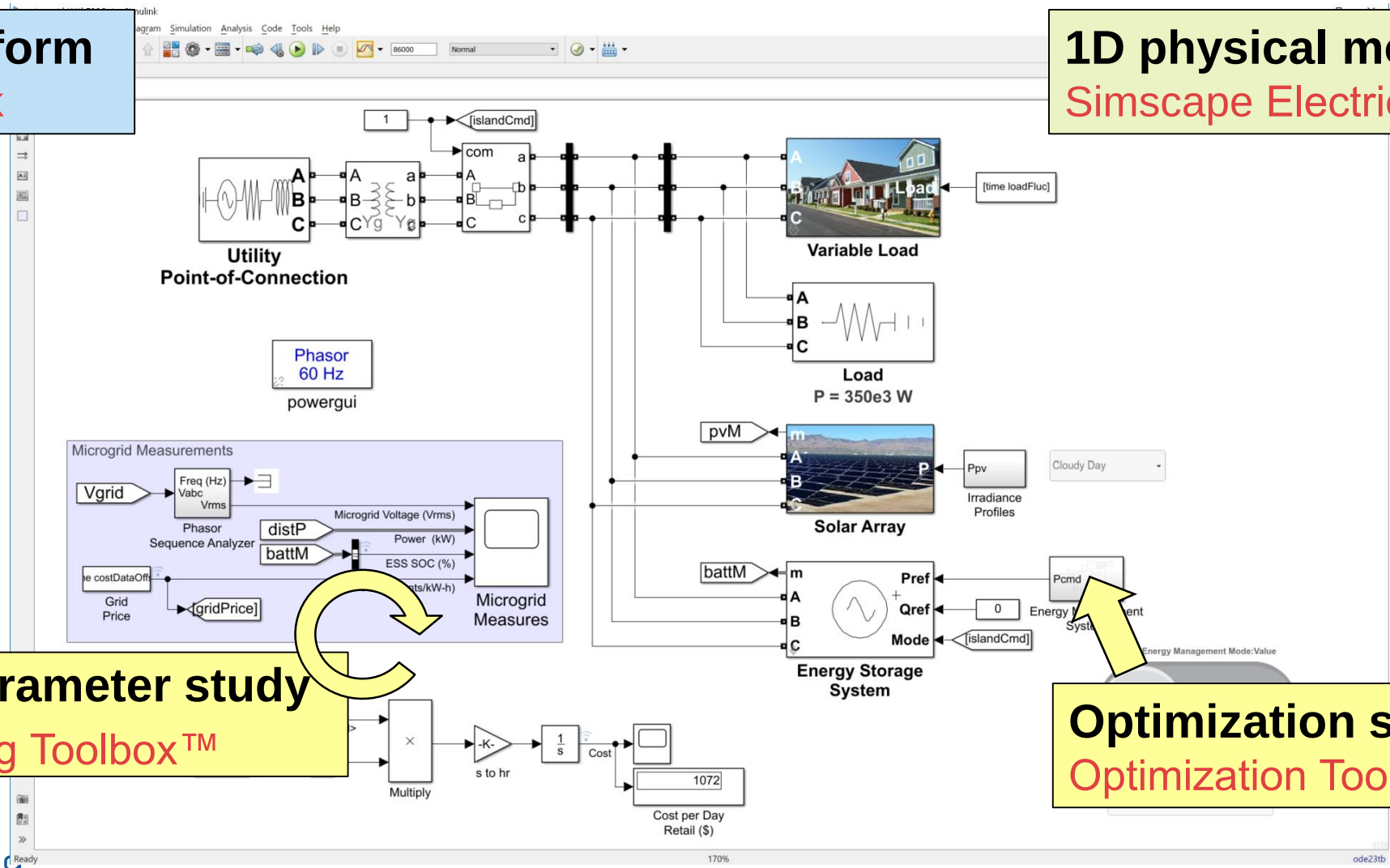
- Mixed-integer nonlinear optimization
  - *Genetic algorithm*

System Simulation – Model Configuration

- Community EMS

Simulation platform
MATLAB, Simulink

1D physical modeling
Simscape Electrical™

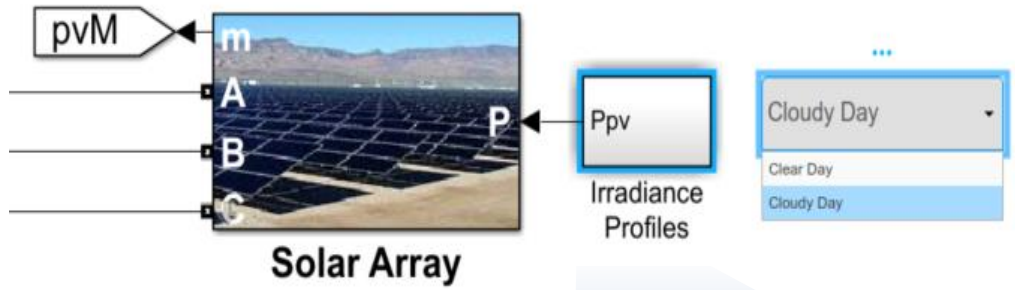


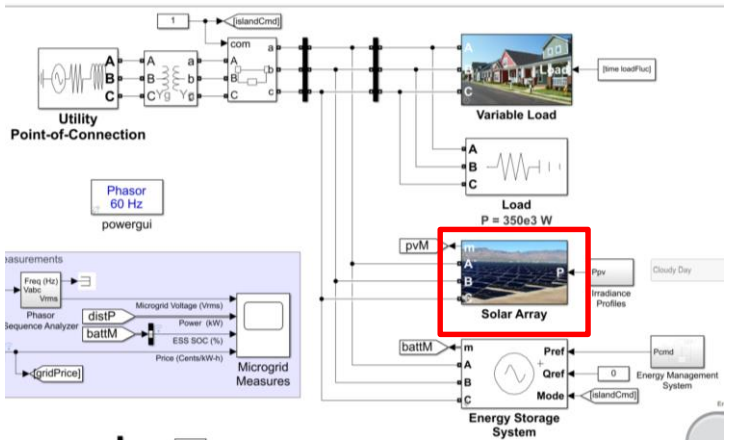
Speed-up of parameter study
Parallel Computing Toolbox™

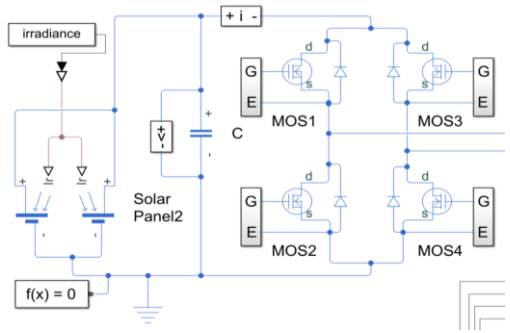
Optimization solver
Optimization Toolbox™

System Simulation – Plant Modeling

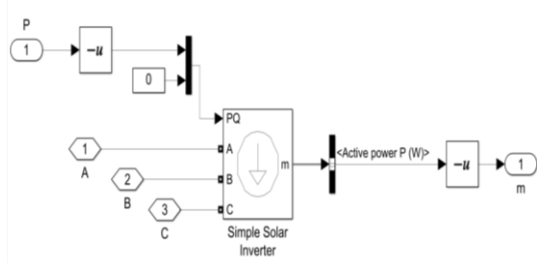
- Advantage of Simscape Electrical
  - Easy to build a circuit model
  - Fidelity change for each component
  - Integration with data
  - High scalability and reusability





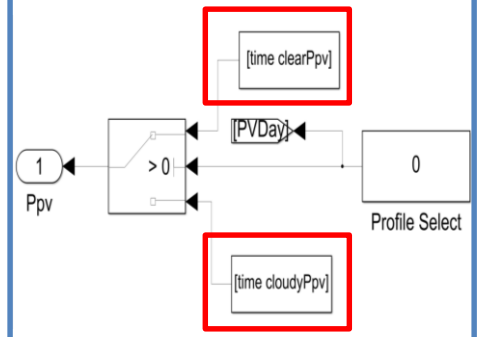


or



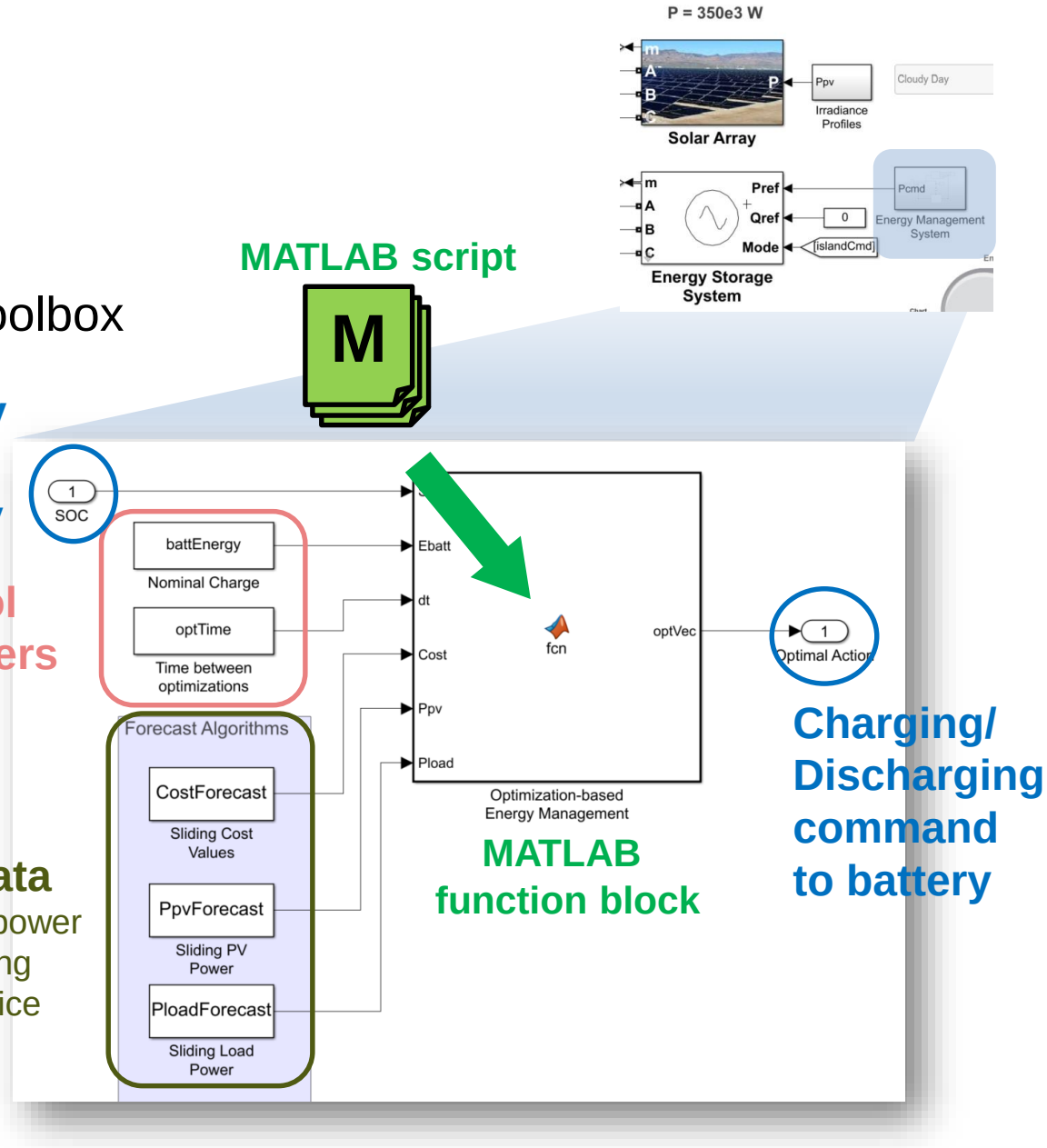
High Fidelity for Accuracy

Low Fidelity for Simulation Speed



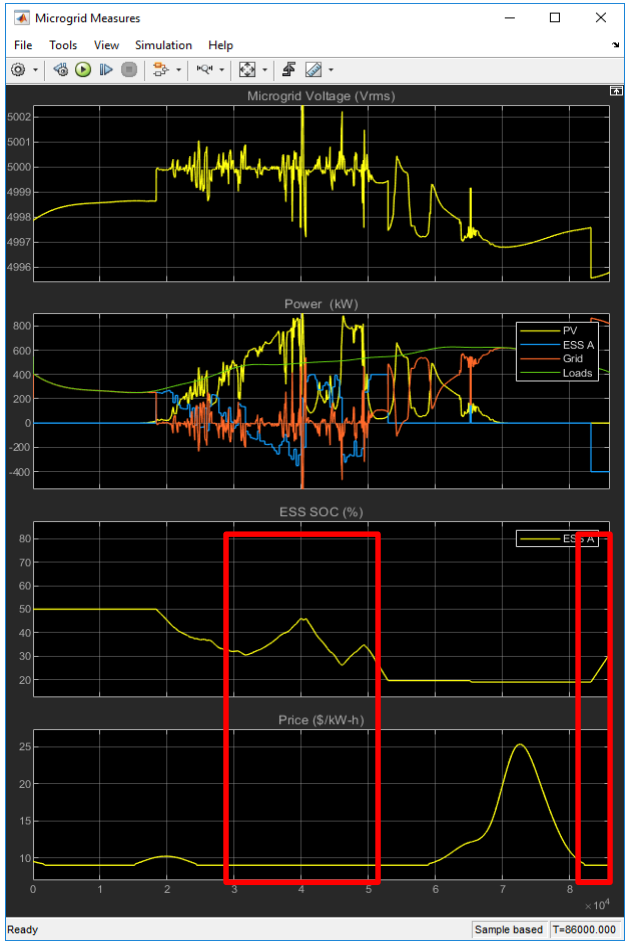
System Simulation – Smart EMS

- Optimize electricity cost every few hours
  - Linear programming solver in Optimization Toolbox
  - Input
    - Battery stored energy
    - Control Parameters
    - Predicted data
  - Output
    - Charging/Discharging command

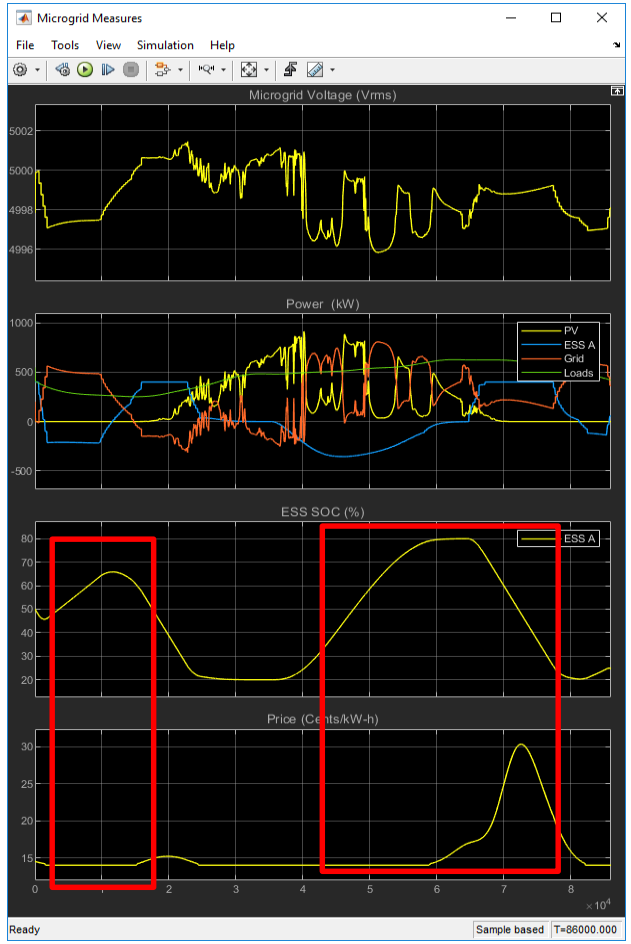


Policy Comparison - Cloudy Day

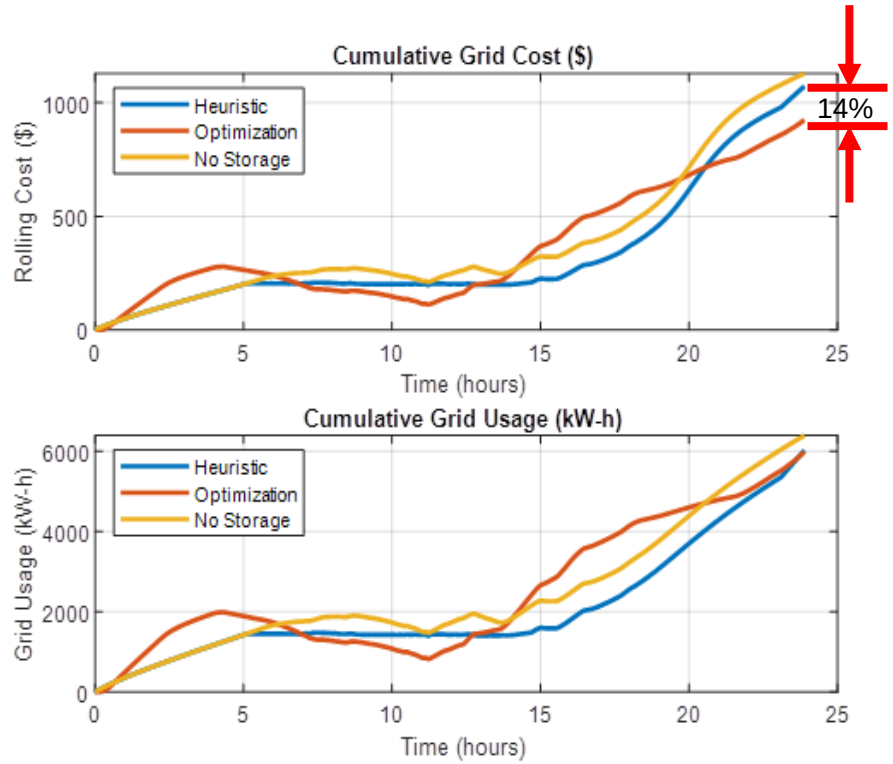
Heuristic



Optimized



Comparison

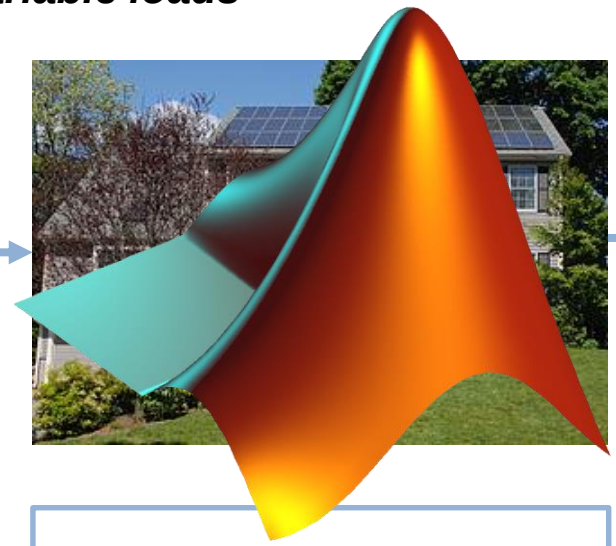


14% lower cost with optimization

Community EMS

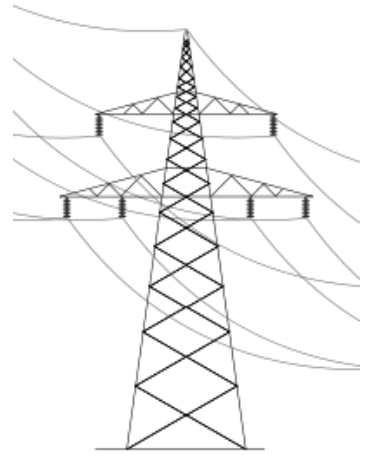
CEMS – *community with variable loads*

PV Panels:
- MPPT Control
- Power electronics

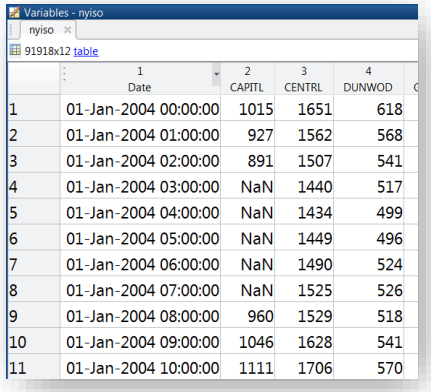


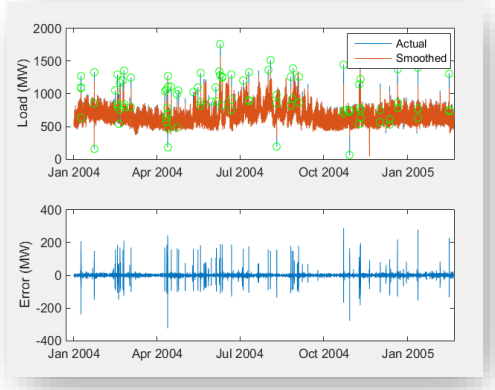
Decision Logic:
- Store/draw power
- Use external grid
- Optimize energy

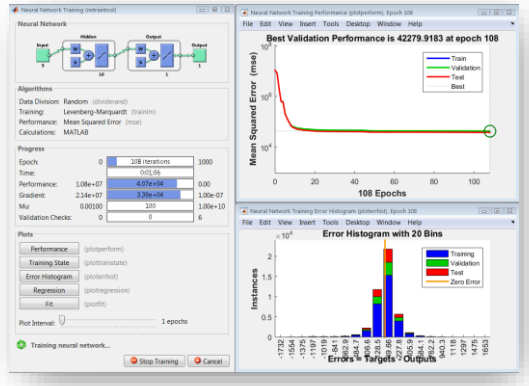
Battery System:
- Charge controls
- Discharge controls
- Power electronics

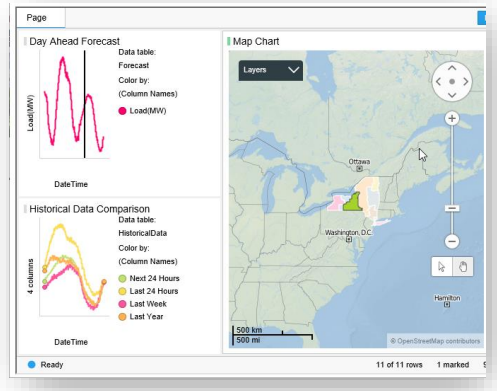


EMS Development Workflow









Access Data | **Analyze Data** | **Develop Models** | **Deploy**

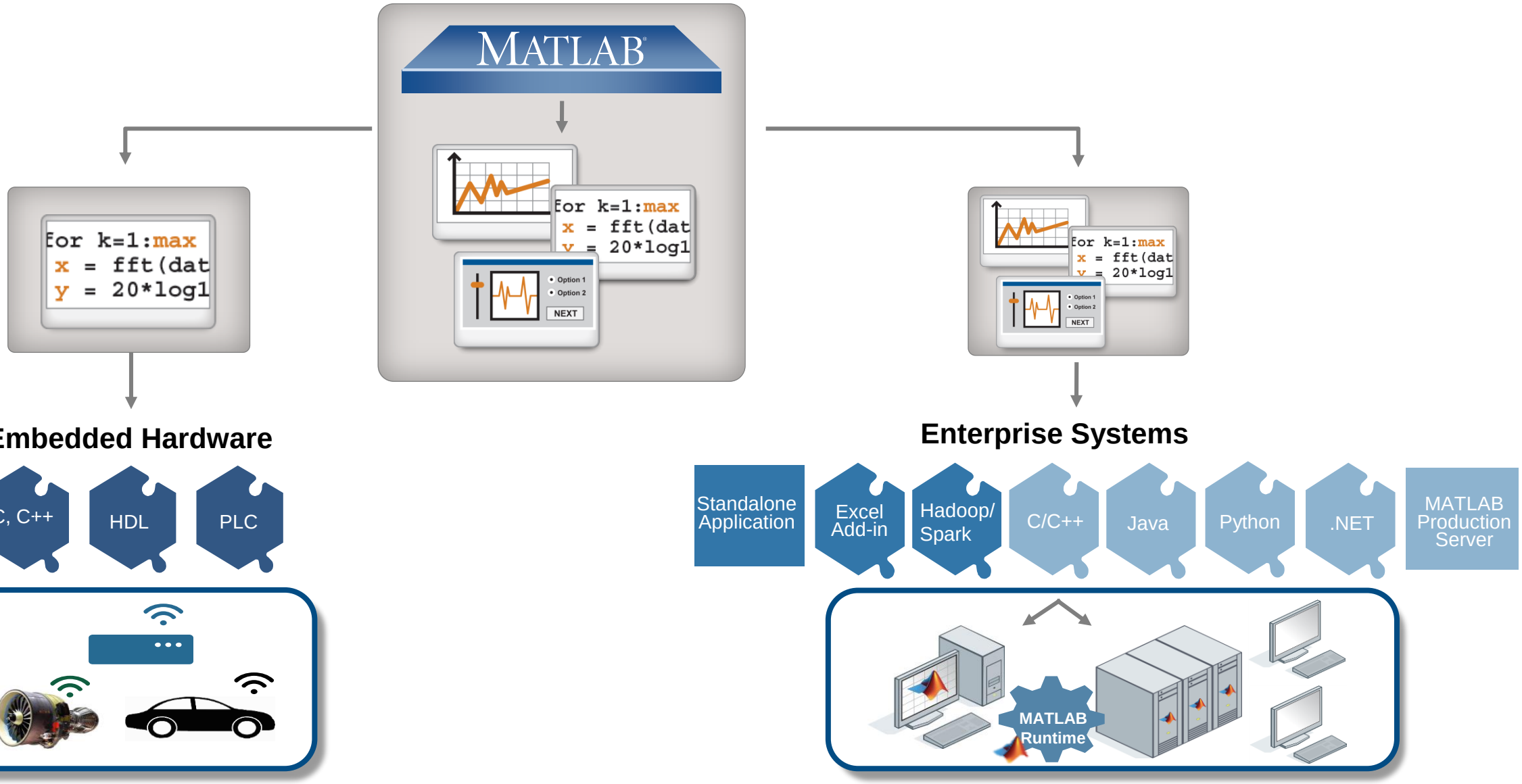
- Desktop apps
- Enterprise systems
- Embedded devices

Deployment

- System integration, system test
- Continually monitor performance
  - Monitor for predictive maintenance
  - Use models as digital twins
  - Analyze against system objectives

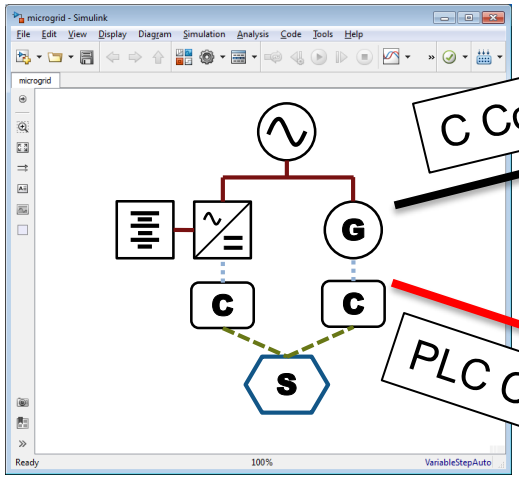
Need to consider integration to both embedded systems and enterprise IT Workflows

Integrate Analytics with Systems



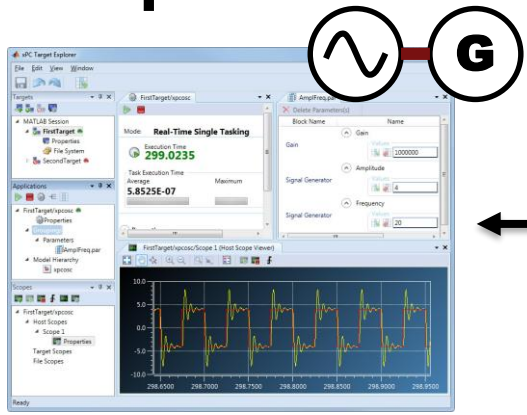
Deployment Workflow on Embedded Systems

Design

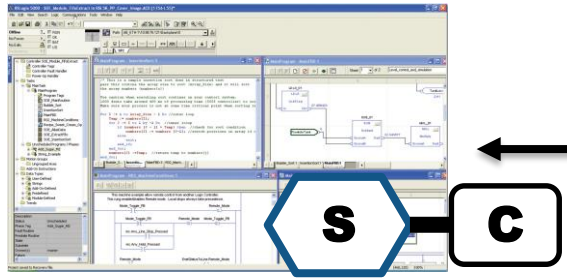


Desktop Simulation

Implement

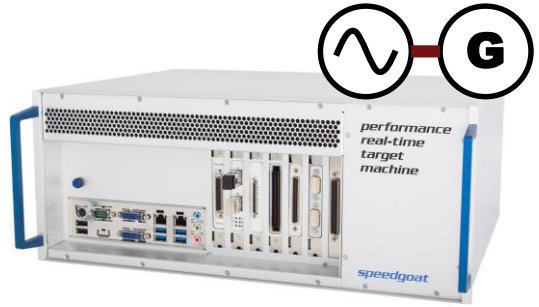


Real-time Interface



PLC Development Environment

Test



Equipment Simulation

Bus

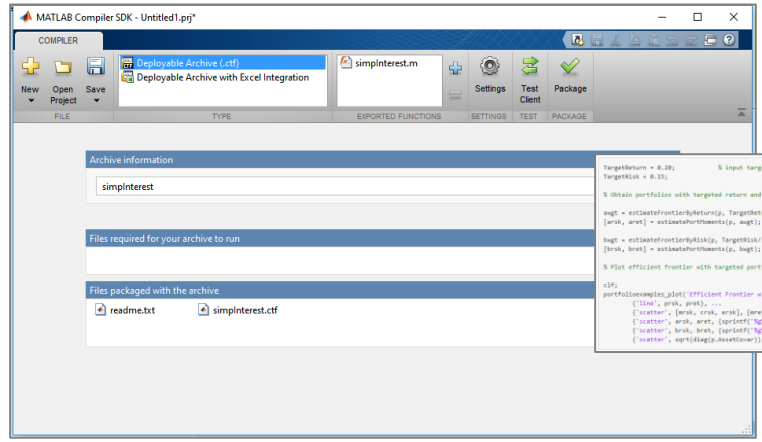
Industrial Controller

Deployment Workflow on Cloud and Business Systems

Click the ‘Package’ button and wait for the compiler to generate the deployable archive

.ctf file

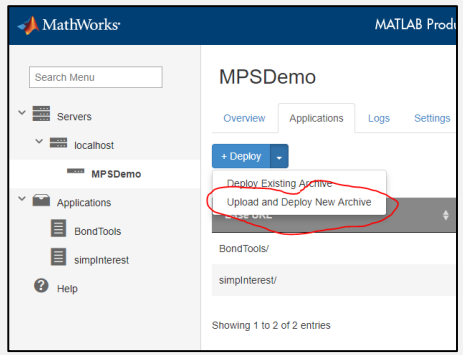
MATLAB Compiler SDK



MATLAB

Code / test

MATLAB Production Server



Copy .ctf file into the auto_deploy folder or use web dashboard

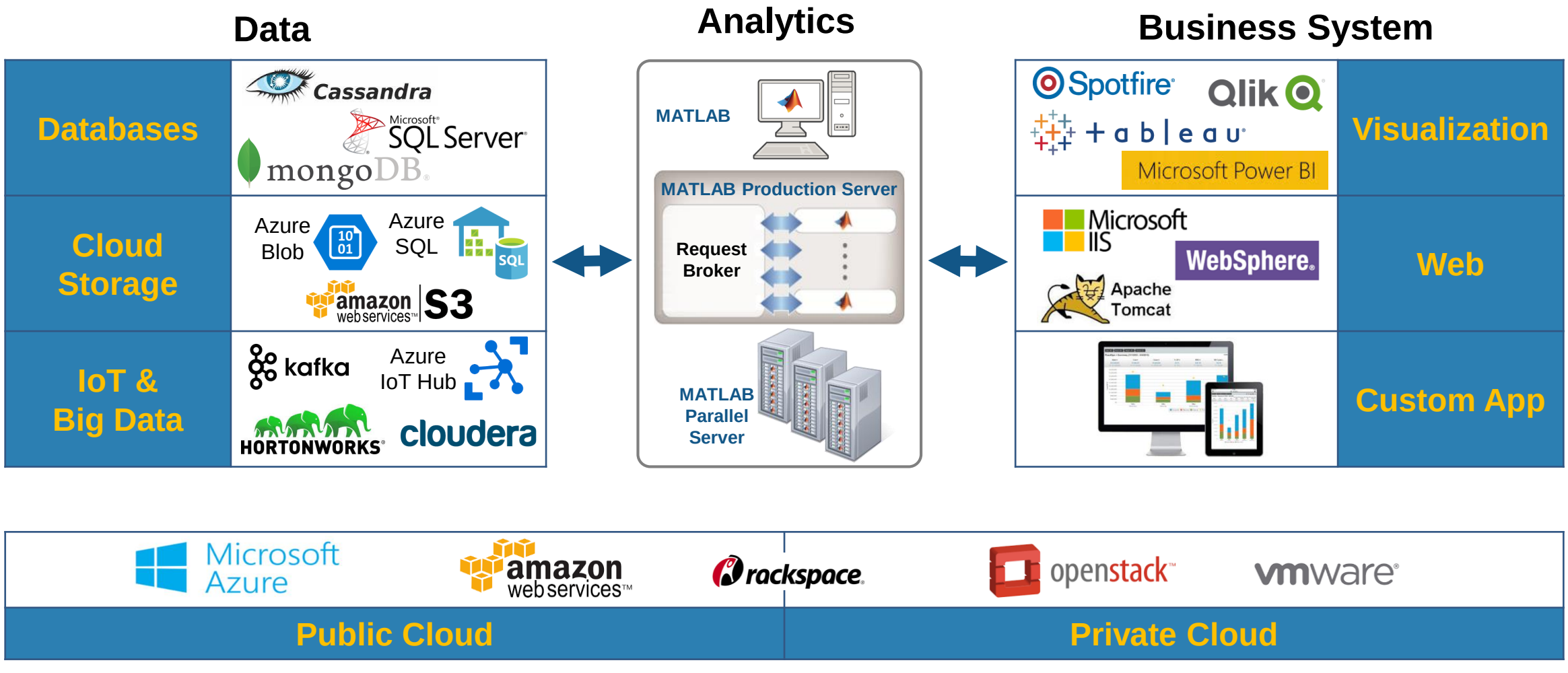
Server file system

Enterprise Application

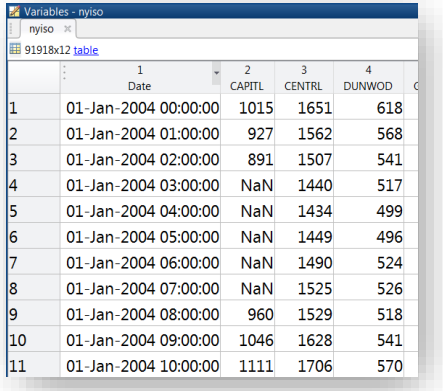
Mobile / Web Application

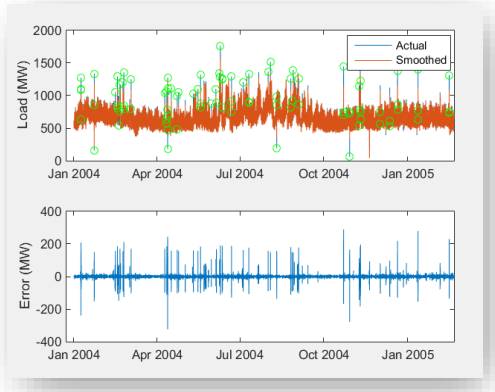
3rd party dashboard

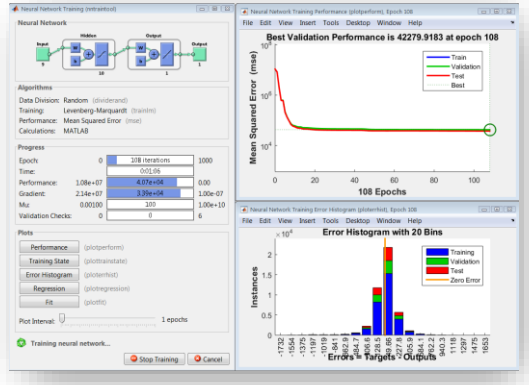
Integration with Enterprise IT

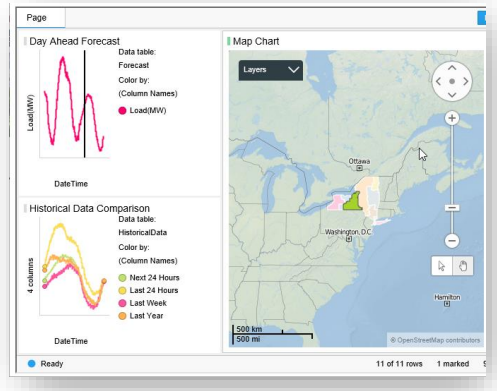


EMS Development Workflow

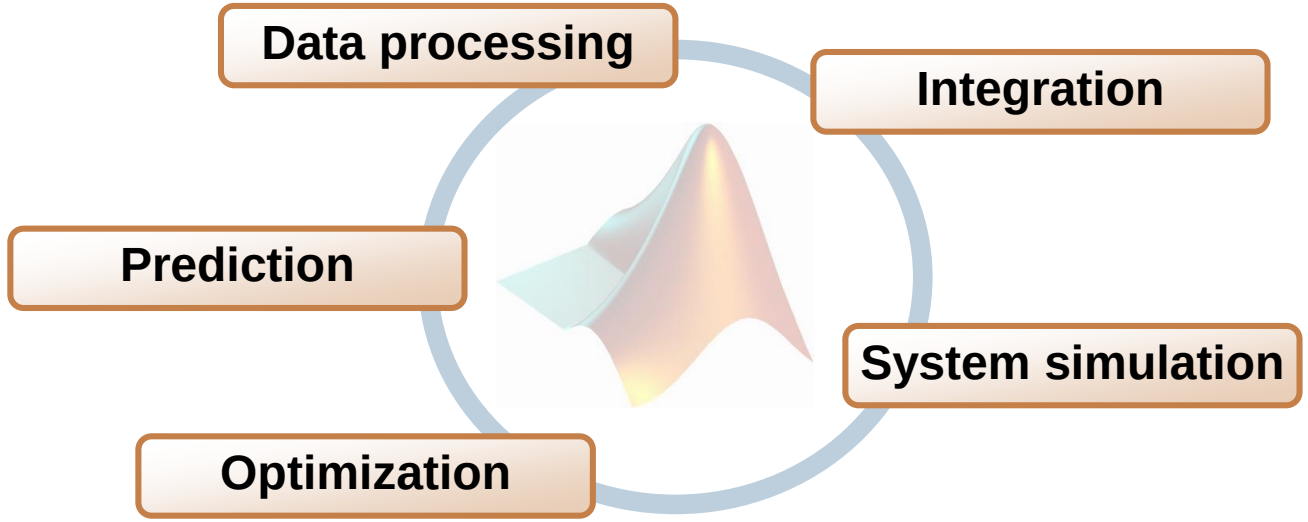








Access Data | **Analyze Data** | **Develop** | **Deploy**



BuildingIQ Develops Proactive Algorithms for HVAC Energy Optimization in Large-Scale Buildings

Challenge

Develop a real-time system to minimize HVAC energy costs in large-scale commercial buildings via proactive, predictive optimization

Solution

Use MATLAB to analyze and visualize big data sets, implement advanced optimization algorithms, and run the algorithms in a production cloud environment

Results

- Gigabytes of data analyzed and visualized
- Algorithm development speed increased tenfold
- Best algorithmic approaches quickly identified

Large-scale commercial buildings can reduce energy costs by 10–25% with BuildingIQ's energy optimization system.

"MATLAB has helped accelerate our R&D and deployment with its robust numerical algorithms, extensive visualization and analytics tools, reliable optimization routines, support for object-oriented programming, and ability to run in the cloud with our production Java applications."

- Borislav Savkovic, Building IQ

Shanghai Electric Builds and Deploys Cost-Saving Enterprise Software for Planning and Designing Distributed Energy Systems

Challenge

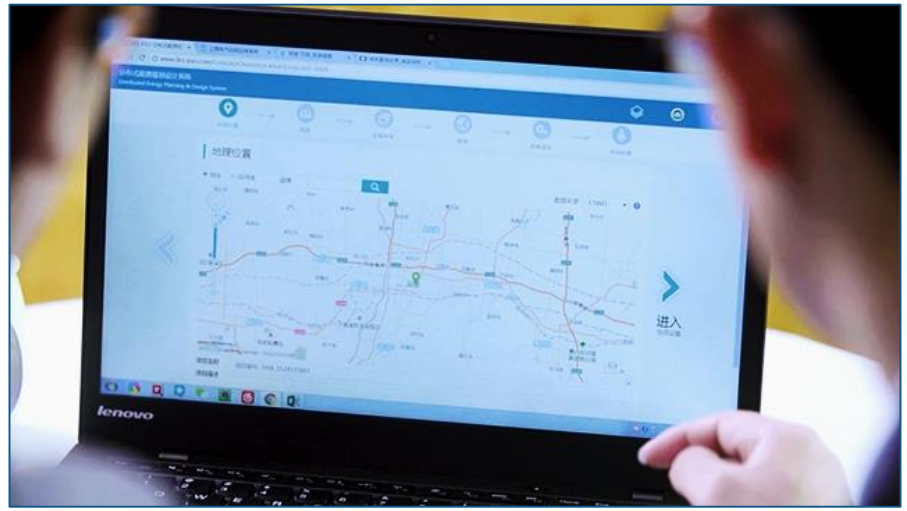
Develop web-accessible software for planning and designing distributed energy systems

Solution

Use MATLAB to develop algorithms that compute investment return based on models of energy production subsystems, loads, and grids, and then use MATLAB Production Server to deploy the algorithms in a production IT system

Results

- Delivery time reduced by six months
- 2 million Chinese yuan saved on a single project
- Updates deployed immediately and without IT assistance



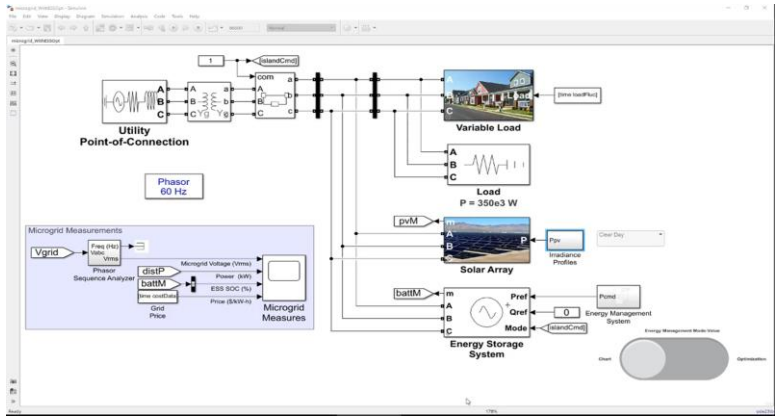
DES-PSO web user interface

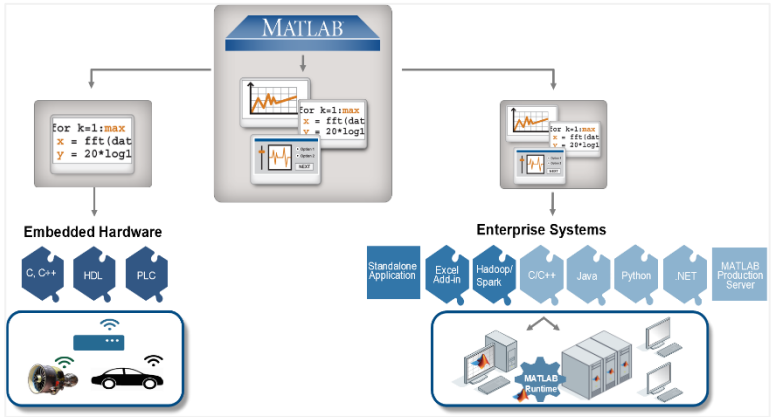
"My team's expertise is in energy modeling or algorithm development, not in deploying software into production. MATLAB saved us months of development time on the models and algorithms, and then made it easy to deploy them as part of a stable, reliable web application without recoding."

- Yunjiao Gu, Shanghai Electric

MATLAB & Simulink help you build a smart EMS

- Integrated development environment
  - Data analytics
  - Predictive modeling
  - Optimization
  - Control
  - System Design
- Virtual prototyping
- Deployment options
  - Deploy to embedded systems
  - Deploy to enterprise systems

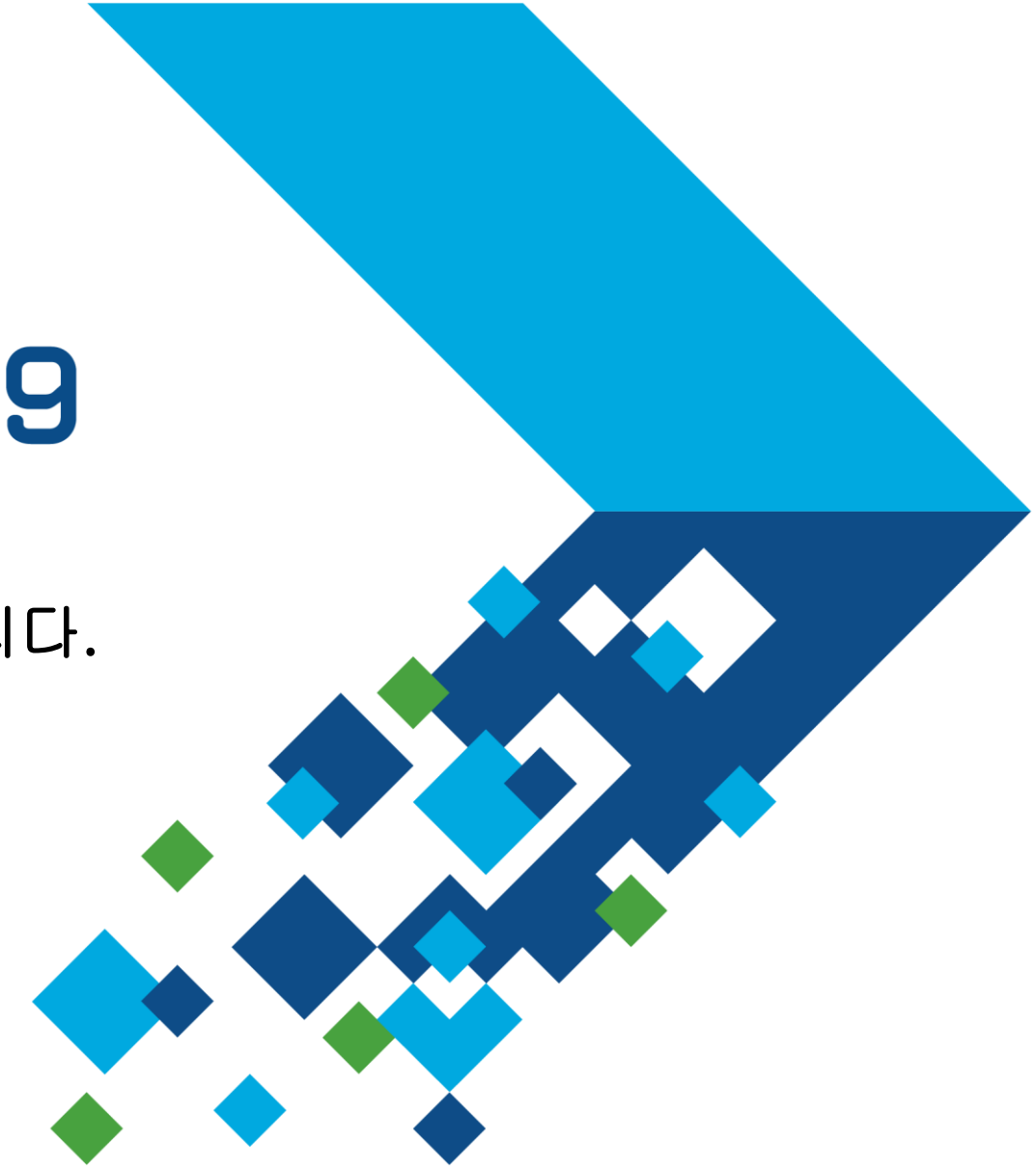




MATLAB EXPO 2019

데모 부스와 상담부스로 질문 하시기 바랍니다.

감사합니다



Learn More

On the web

- Microgrid System Development and Analysis - video series
- Data Analytics with MATLAB - webinar
- Linear and Mixed-Integer Linear Programming in MATLAB - webinar

Products

In the microgrid demo

- MATLAB
- Simulink
- Simscape
- Simscape Electrical
- Stateflow
- Optimization Toolbox

Additional products for EMS

- Statistics & Machine Learning Toolbox
- Model Predictive Control Toolbox
- Signal Processing Toolbox
- Control System Toolbox
- MATLAB Compiler
- Embedded Coder
- Simulink Test
- MATLAB Production Server
- MATLAB Parallel Server