Pacific Northwest
NATIONAL LABORATORY

GridSTAGE
Spatio Temporal Adversarial Scenario GEneration framework

April 1, 2020

The PowerDrone Team

Sai Pushpak Nandanoori, Seemita Pal,

Sutanay Choudhury, Soumya Kundu and Khushbu Agarwal

U.S. DEPARTMENT OF ENERGY BATTELLE

eML Brown Bag

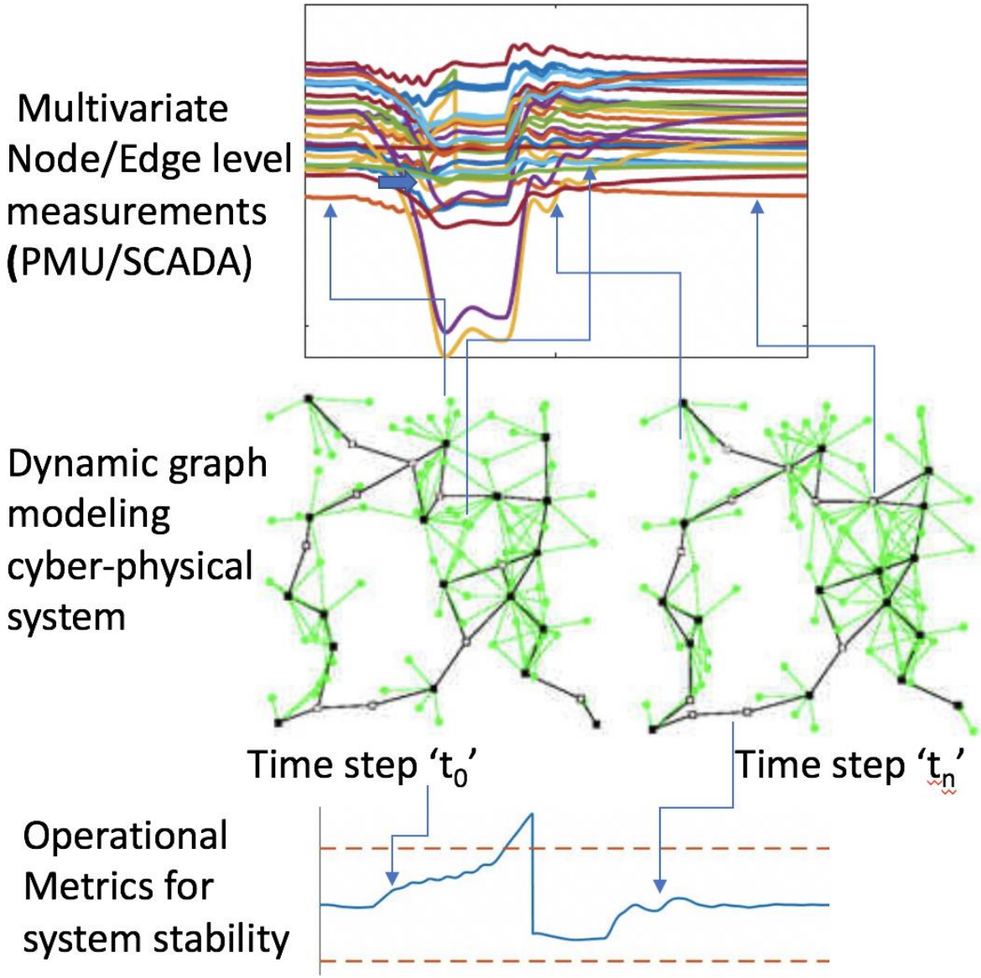
Outline

- Motivation
- Background on Automatic Generation Control
- Simulation Framework – Overview, Parameters and Data Generation Setup
- Demonstration
- Potential Applications

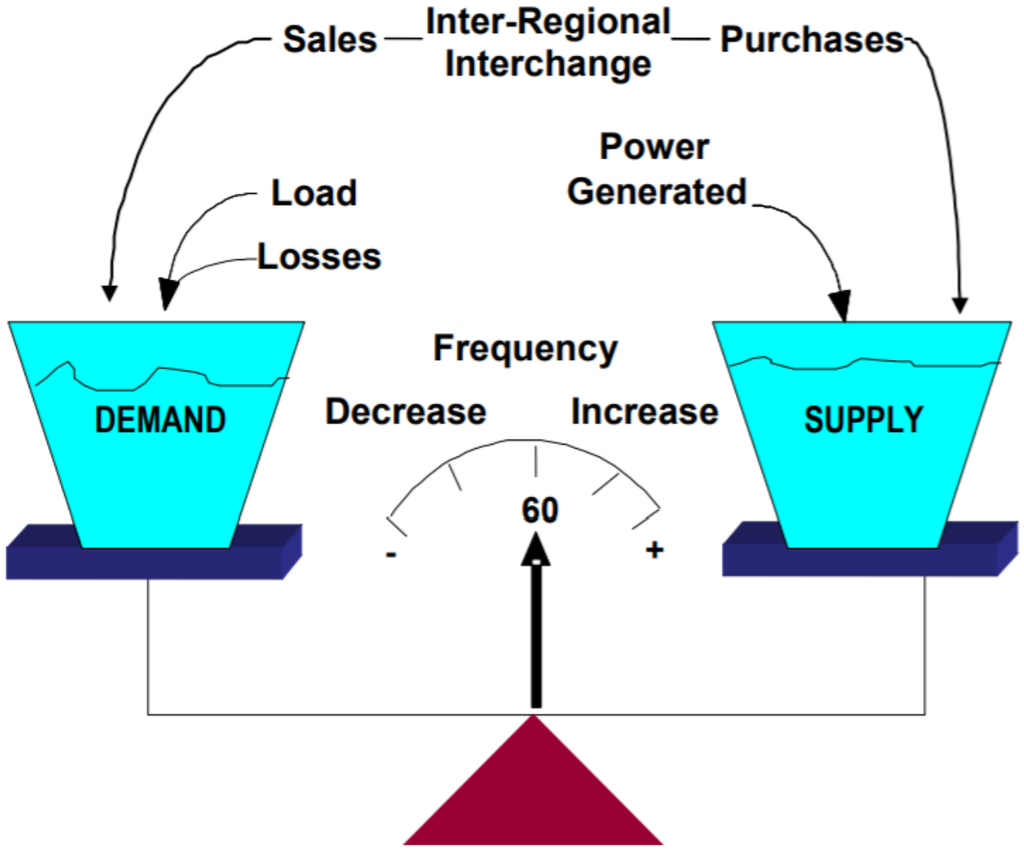
Motivation

- One of the most critical infrastructures and attractive target for nation-state actors, disgruntled insiders or casual hackers
- Increased deployment of sensors
- Automatic Generation Control (AGC) is an automatic closed-loop control regulating the system-wide frequency
- Control capabilities should be complemented by better detection and mitigation application/s

Thorough analysis or application development requires simulation framework for generating appropriate datasets



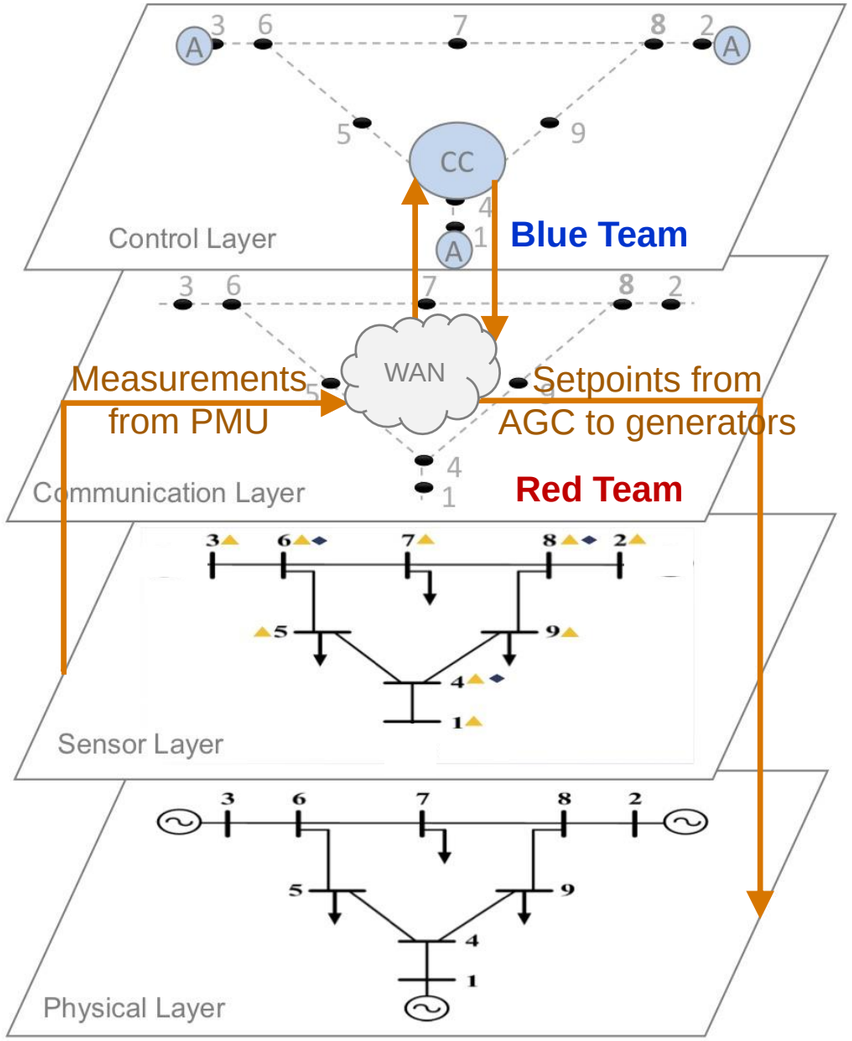
High-Level Understanding of Automatic Generation Control



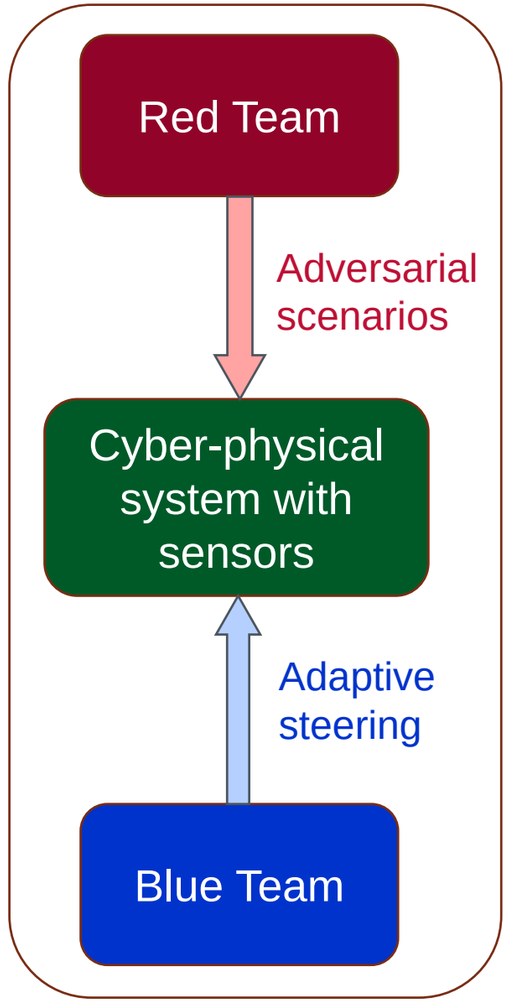
Area Control Error (ACE) = (Difference between actual and scheduled net interchange) + (Difference between actual and nominal frequency) * bias factor

Source: NERC Balancing and Frequency Control

Modular Simulation Framework

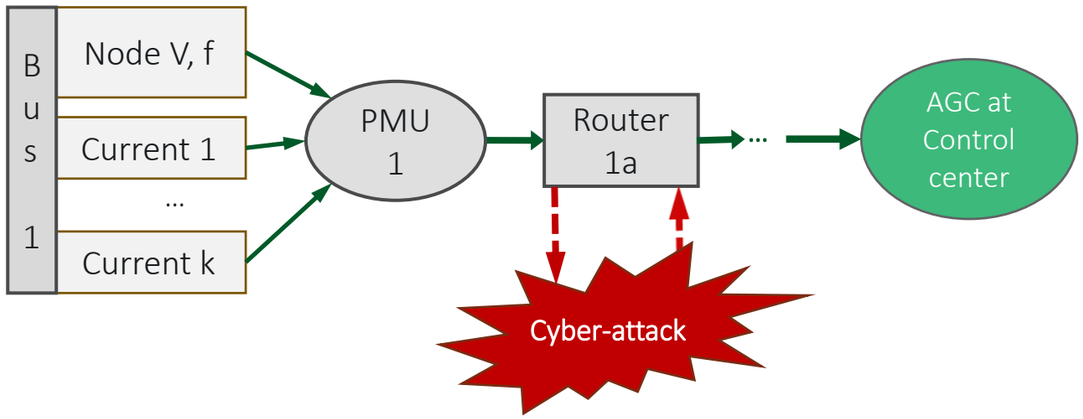


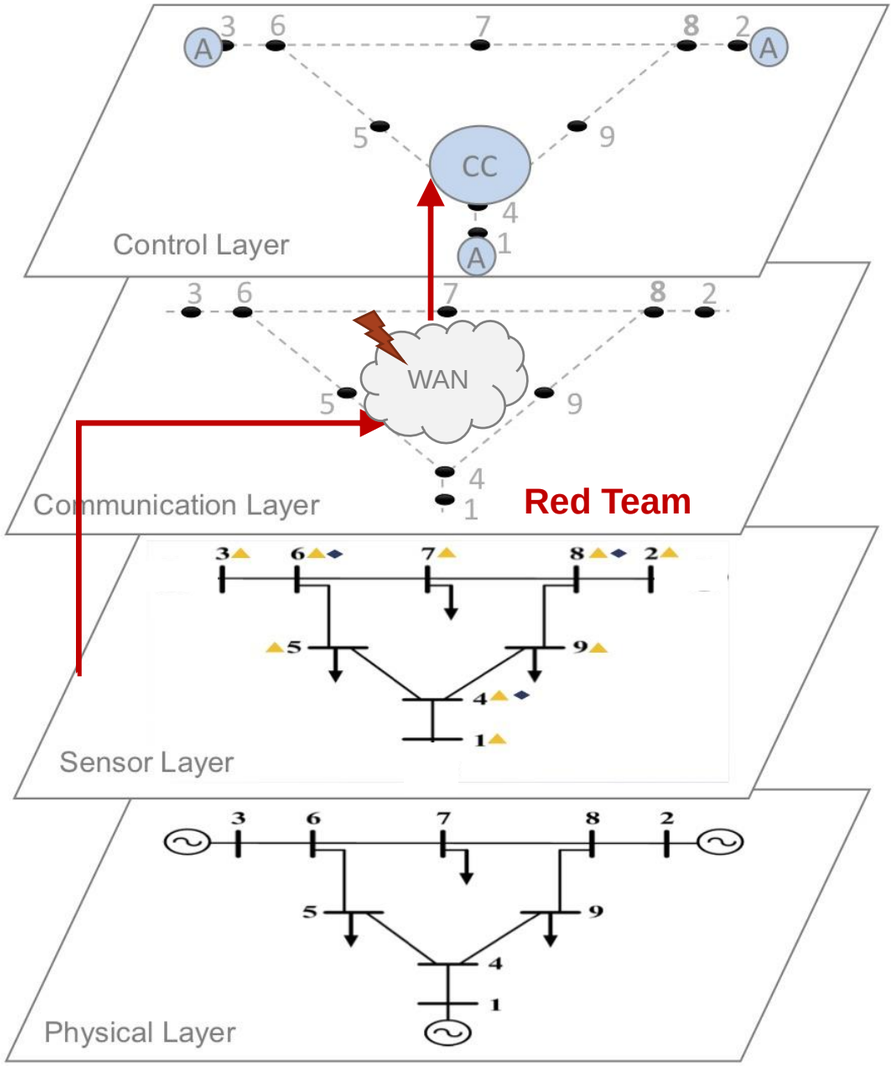
- Red Team: Simulates adversarial actions through cyber layer
- Blue Team: Detects adversarial conditions and steers system to stable operating conditions using appropriate controls
- System: Cyber-physical system model with sensors for monitoring and control



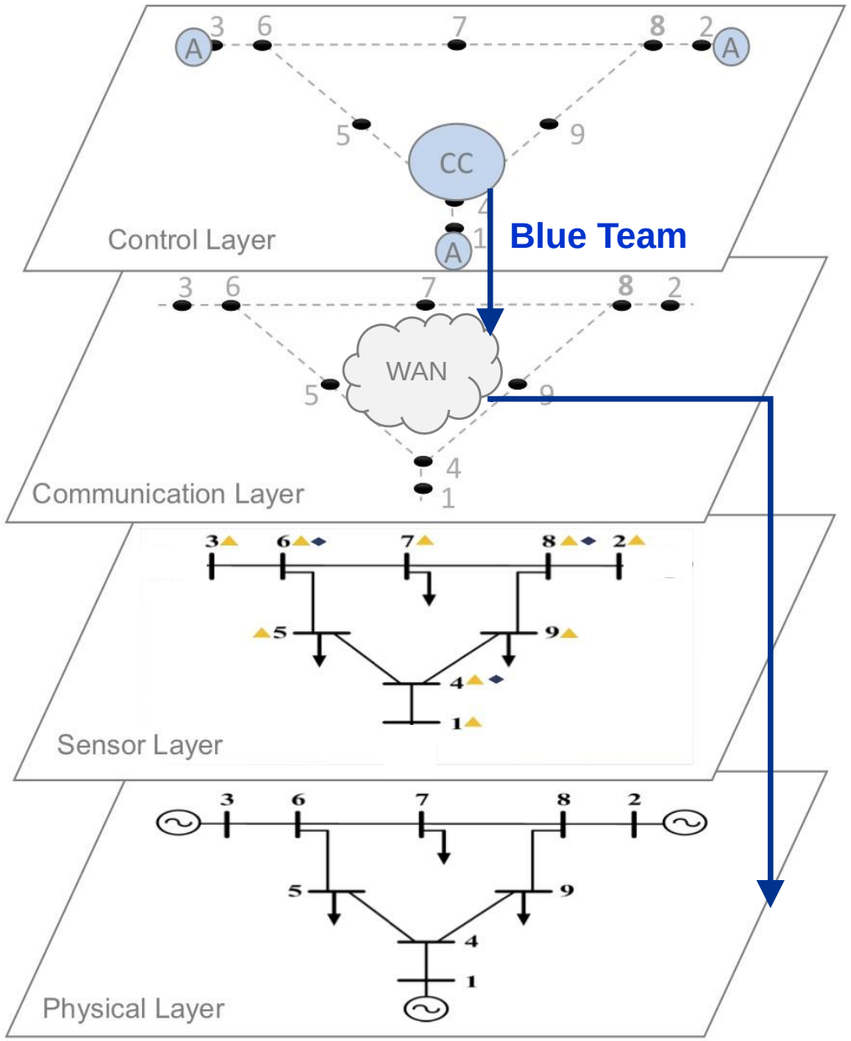
Red Team Overview

- Mimics real-world adversarial scenarios (modeled and integrated in the framework)
- Possible to implement cyber-attacks on integrity or availability
  - Step attacks (up or down)
  - Ramp attacks (up or down)
  - Random attacks
  - Packet drops attack (random, consecutive or sequential)





Blue Team Overview

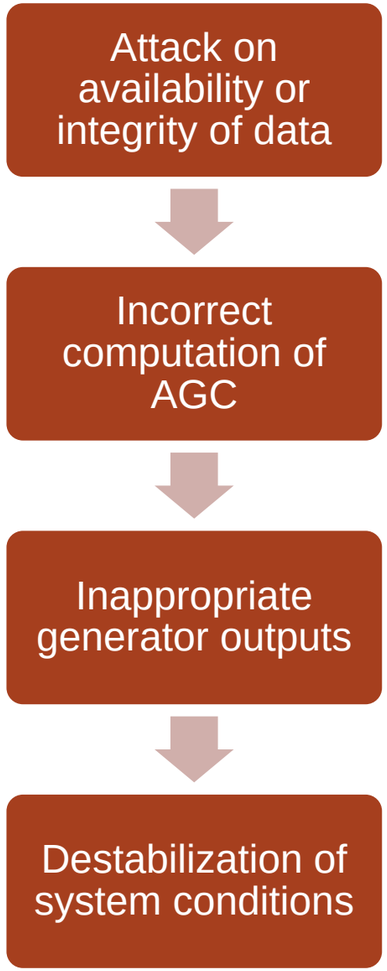


Current Implementation added by team

- Controller uses high-frequency PMU data to determine the Area Control Error (ACE)
- Computed ACE value determines the setpoints of all the generators in the system
- Possible to perform AGC using SCADA data

Impact Quantification

- Enable quantification of impact using state measurements like frequency or voltage which have known nominal values or SCADA measurements can be used for reference
- 1-norm impact metric = $||\text{state measurement} - \text{nominal value}||_1$
  - Average impact due to cyber-attack
  - Most sensitive to cyber-attack and divergence indicates attack duration
- 2-norm impact metric = $||\text{state measurement} - \text{nominal value}||_2$
  - R.M.S. impact due to cyber-attack, eliminates noise
- Inf-norm impact metric = $||\text{state measurement} - \text{nominal value}||_{inf}$
  - Quantification of worst-case impact due to cyber-attack
- Divergence Factor = |metric(t) – metric(t-1)|
  - In case of attack, divergence factor value will be more



Overview of Simulation Framework

- Requirements:
  - MATLAB
  - Power System Toolbox
    http://www.etk.ee.kth.se/personal/vanfretti/pst/Power_System_Toolbox_Webpage/PST.html
- Integrated relevant data and controller files from Power System Toolbox in order to simulate power system dynamics
- Programs modeling attack scenarios and load changes integrated

Simulation Framework: User Inputs

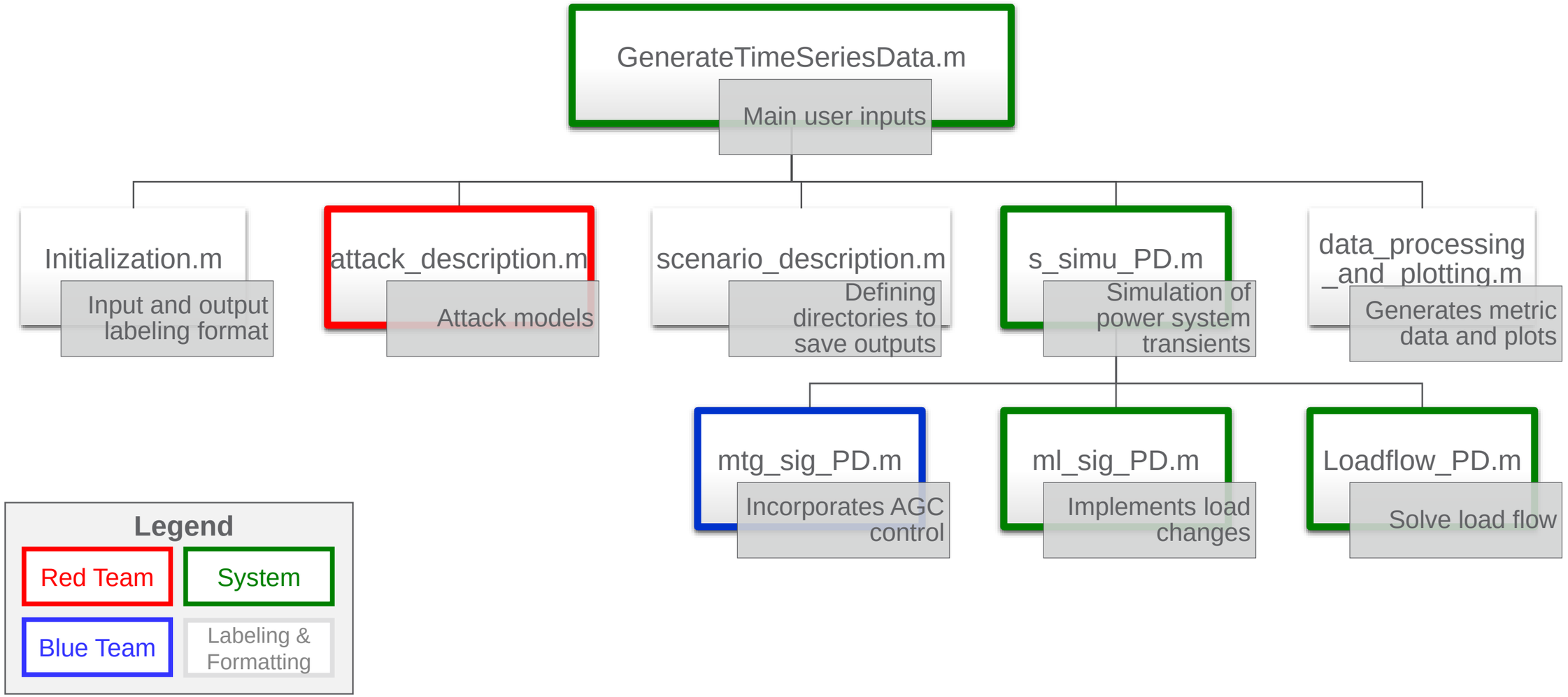
- Network: Choose the IEEE bus system: '9', '39', '68', '145’
- Time (total time of run and time-step)
- AGC control (enable/disable and time-step)
- PSS control (enable/disable)
- Number of areas (1 or 2)
- Load changes (enable/disable, location, time)
- Cyber-attack (enable/disable, type, time, location and magnitude)
- Faults (enable/disable, type and location, time of occurrence)

Simulation Framework: User Inputs

LAYER	DEVICE	LOCATION	SAMPLING INTERVAL
Sensor Layer	PMU	Limited number of buses*	0.02 s*
	SCADA	All buses*	2 s*
Control Layer	AGC	Control center	1 s*
Physical Layer	Generators	Generator buses	Depends on AGC's decision

* can be user-defined as well

Workflow of Framework



User-defined Parameters (Inside GenerateTimeSeriesData.m)

```
% System configurations
Network = '68'; % Choose the IEEE bus system: '9', '39', '68', '145'
agc_control = 1; % '1' enables AGC control; '0' disables AGC control
agc_time_step = 1; % time interval in seconds between agc control
pss_control = 0; % '1' enables PSS control; '0' disables PSS control
num_area = 2; % 1- one area and 2 - two area
load_changes = 1; % '1' enables load changes; '0' disables load changes
TimeStep_of_simulation = 0.01;
SimulationTime = 60;
PMU_SamplingFreq  = 50; % Measurements every second
```

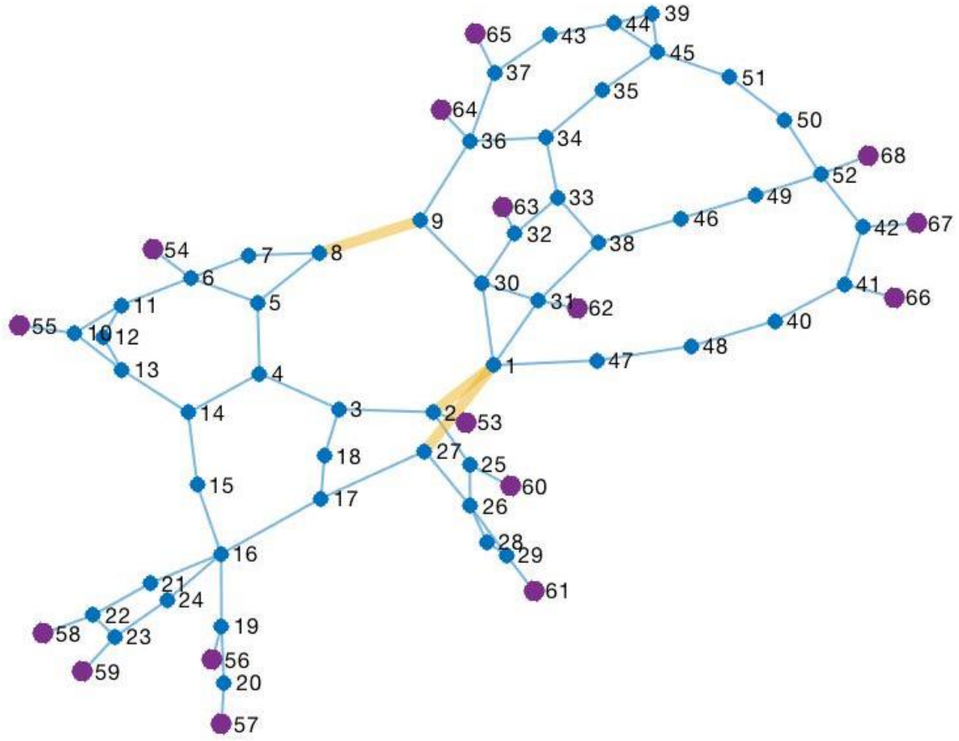
```
% Attack Parameters
PMU_attack = 0; % '1' enables cyber-attacks on PMUs; '0' disables cyber-attacks on PMUs
AttackTypes = {'Latency','PacketDrop','Ramp','Step','Poisoning'};
AT = AttackTypes{3};
% Cyber-attack is to be introduced in PMU sensors at attack location bus
% AttackTypes{1}: 'Latency' attack (additional delays introduced  in PMU packet latencies)
% AttackTypes{2}: 'PacketDrop' attack (unauthorized  dropping  of  PMU packets)
% AttackTypes{3}: 'Ramp' attack (PMU measurement gradually modified over attack period
% AttackTypes{4}: 'Step' attack (PMU measurement scaled based on scaling factor)
% AttackTypes{5}: 'Poisoning' attack (PMU measurement are randomly corrupted
% by noise [noise parameters are picked from a Gaussian distribution])
% Predefine the mean and variance for data poisioning
data_poison.mean = 0.0;
data_poison.var  = 0.02;
% Predefined set of attack magnitude percentages to be simulated
attack_magnitudes_percent = [0.05 0.1 0.2]; % Attack magnitudes
```

Fault Types and Simulation Set-up (Inside GenerateTimeSeriesData.m)

```
sw_con = [...
    0.0    0    0    0    0    0      0.005; %sets intitial time step
    0.05  line(i_fault_location,1)   line(i_fault_location,2)   0    0   6       0.005;
    0.1 0    0    0    0    0      0.005; %clear fault at bus
    0.2 0    0    0    0    0      0.005; %clear remote end
    0.5 0    0    0    0    0      0.005; % increase time step
    %             1.0  0    0    0    0    0    0.005; % increase time step
    SimulationTime  0    0    0    0    0    0]; % end simulation
%{
row 1 col1  simulation start time (s) (cols 2 to 6 zeros)
      col7  initial time step (s)
row 2 col1  fault application time (s)
      col2  bus number at which fault is applied
      col3  bus number defining far end of faulted line
      col4  zero sequence impedance in pu on system base
      col5  negative sequence impedance in pu on system base
      col6  type of fault  - 0 three phase
                           - 1 line to ground
                           - 2 line-to-line to ground
                           - 3 line-to-line
                           - 4 loss of line with no fault
                           - 5 loss of load at bus
                           - 6 no action
      col7  time step for fault period (s)
row 3 col1  near end fault clearing time (s) (cols 2 to 6 zeros)
      col7  time step for second part of fault (s)
row 4 col1  far end fault clearing time (s) (cols 2 to 6 zeros)
      col7  time step for fault cleared simulation (s)
row 5 col1  time to change step length (s)
      col7  time step (s)
%}
```

Other System Configuration Parameters (Inside the data file)

- *bus* – consists of each bus information (P_gen, Q_gen, P_load, Q_load, type of bus)
- *line* – consists of connectivity information (From bus, To bus resistance, reactance and other parameters)
- *mac_con* – consists of machine (synchronous generator) parameters
- *exc_con* – excitor controller parameters
- *tg_con* – turbine governor controller parameters
- *pss_con* – power system stabilizer parameters
- *load_con* – non-conforming load buses: indicate the load buses whose load values need to be modified
- *fmeas_con* – indicate the location of PMU buses here
- *Vmeas_con* – indicate the location of SCADA buses here



Scenario Generation

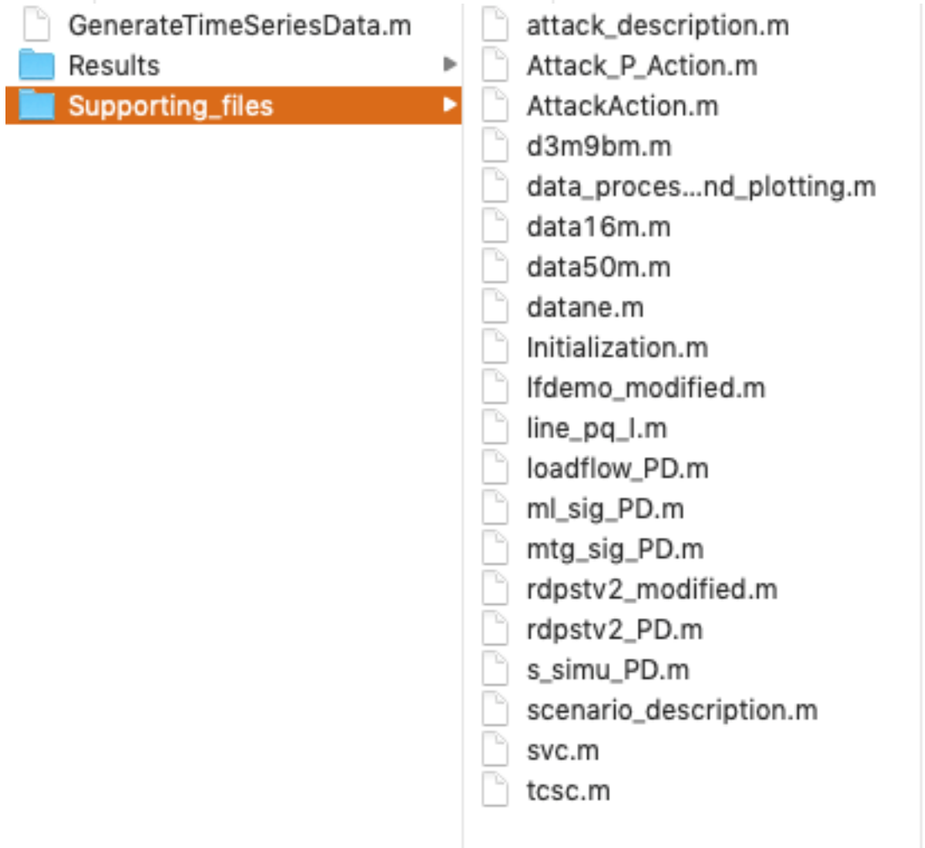
- Scenario #: nonlinear time-domain simulations corresponding to a choice of user-defined parameters
- Each scenario information with all the choice of parameters is saved into a ScenarioDescription.csv file and ScenarioDescription .txt file inside the scenario # directory itself

View of ScenarioDescription.csv for one scenario:

Fault	Attack	Load Changes	AGC	PSS	Fault location	Fault type	Attack location	Attack type	Attack start time (sec)	Attack end time (sec)	Attack duration (sec)
1	1	1	1	0	1,2	three phase fault	4 57	Ramp	25	35	10

Attack amplitude (%)	Location(s) of load changes	Magnitude(s) of load changes	Start time(s) for load changes	End time(s) for load changes
0.299999999999989	8 48 15 23 39 29 50 49	0.4952 0.90781 0.24796 -0.28023 -0.40956 0.43813 -0.056534 -0.54102	4	

Directory Organization

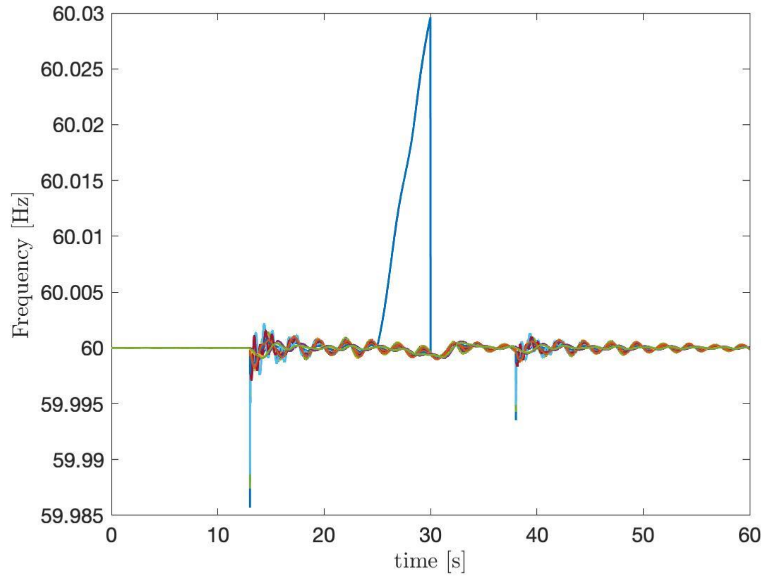


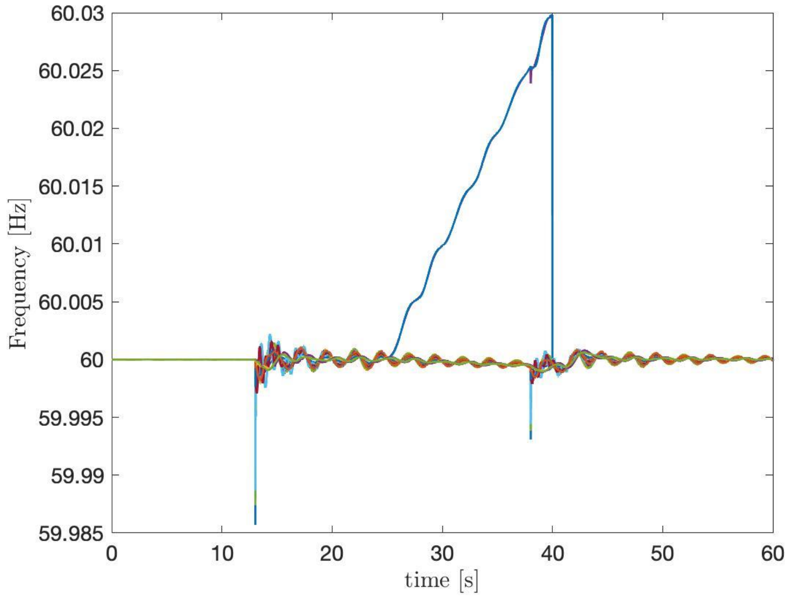
User Inputs for Multiple Scenario Generation (Included in GenerateTimeSeriesData.m)

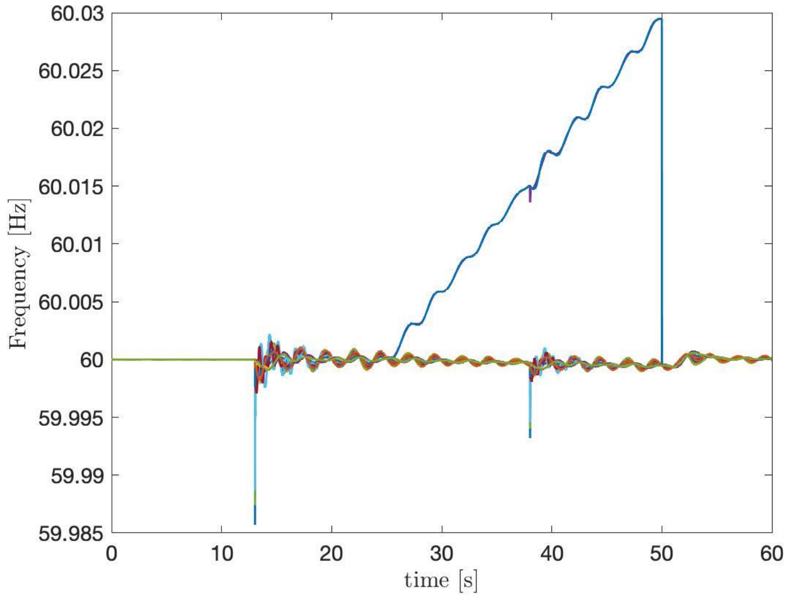
Multiple scenarios can be generated by sweeping over a range of attack characteristics, fault types

- Attack location
- Attack duration
- Attack magnitude
- Attack type
- Fault location
- Fault type

For example, ramp attack with 5 sec, 15 sec and 25 sec attack durations:







Scenario Generation: High-Dimensional Sampling

- Operating scenarios should *appropriately represent the uncertainties* in grid, driven by unpredictability in load and generation (e.g. solar/wind variations)
- Challenges:
  - *High-dimensional uncertainty space* (~1000s of loads and generators)
  - Exponential complexity of brute-force enumeration of *all possible* scenarios
  - Uncertain parameters with possible correlations (e.g. collocated wind)
- Sampling Approach: ***Latin Hypercube Sampling***
  - Scalable method to represent underlying uncertainties with performance guarantee

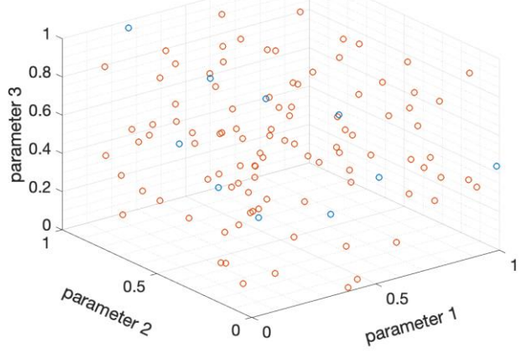
INPUT

# of uncertain parameters (N)
# of sample scenarios (M)

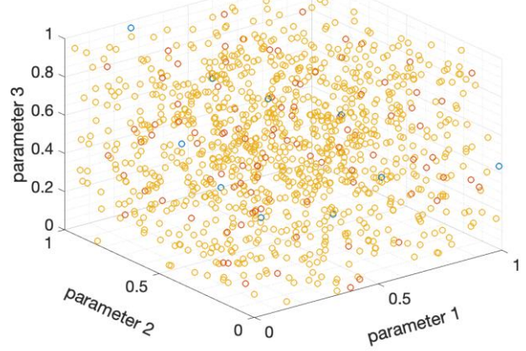
Probability distributions (e.g. uniform, Gaussian)

Any correlations

N=3, M=100



N=3, M=1000



OUTPUT

M points in N dimensions

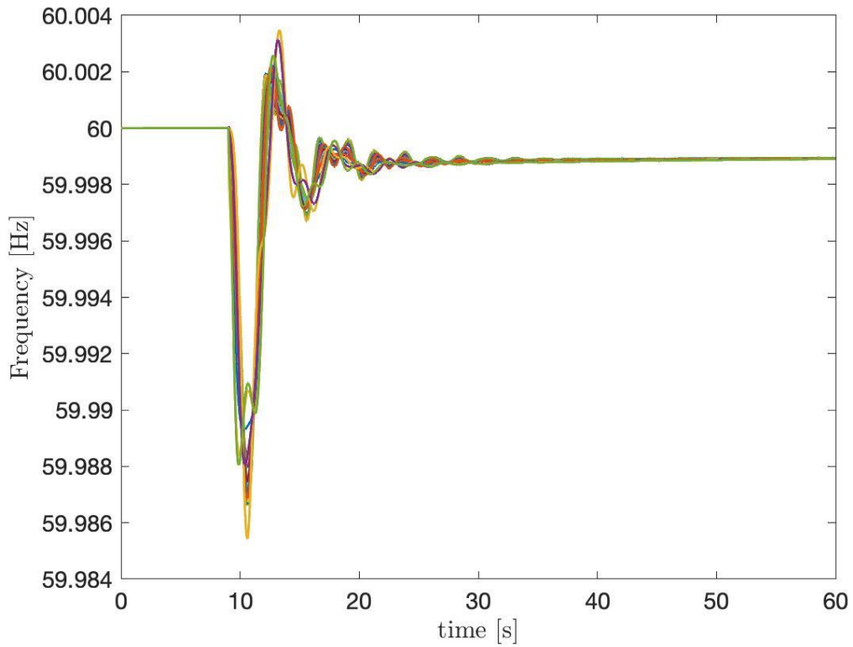
Follows the prescribed underlying statistics

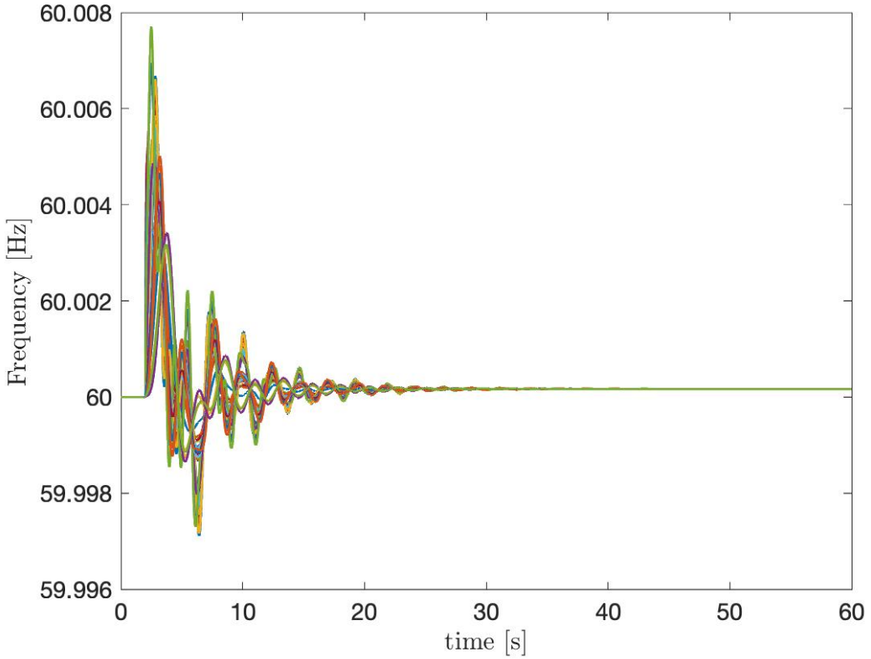
Handles arbitrary distributions

User Inputs for Multiple Scenario Generation (Included in GenerateTimeSeriesData.m)

- Nominal operating data can be generated by perturbing the system with minor load changes across the network
  - Location of load changes, time of load changes are chosen randomly (or user-defined)
  - Amount of load change is chosen based on Latin hyper cube sampling

For example, time-series data corresponding to two scenarios with load changes:





Main User Inputs for Scenario Generation (Included in GenerateTimeSeriesData.m)

```
% Give the number of PMU_locations where cyber-attacks will be simulated
n_attacks_on_location = 1;
% Number of fault locations (data with respect to each fault location will be saved as a single scenario)
n_fault_locations = 1;
% Number of fault-types (data with respect to each fault type will be saved as a single scenario)
n_fault_type = 1;
% Below select number of attack_magnitudes_percent values to be simulated from the set above
n_attacks_on_magnitude = 1; % Should be less than max(size(attack_magnitudes_percent))
% Below select number of attacks on attack duration
% (data corresponding to each attack duration value will be saved as a scenario)
n_attacks_on_duration_of_attack = 1;
attack.start_time_in_sec = 20; % randi(round(0.8*simParams.simTime),1,1);
attack_durations = linspace(1,15,n_attacks_on_duration_of_attack);
```

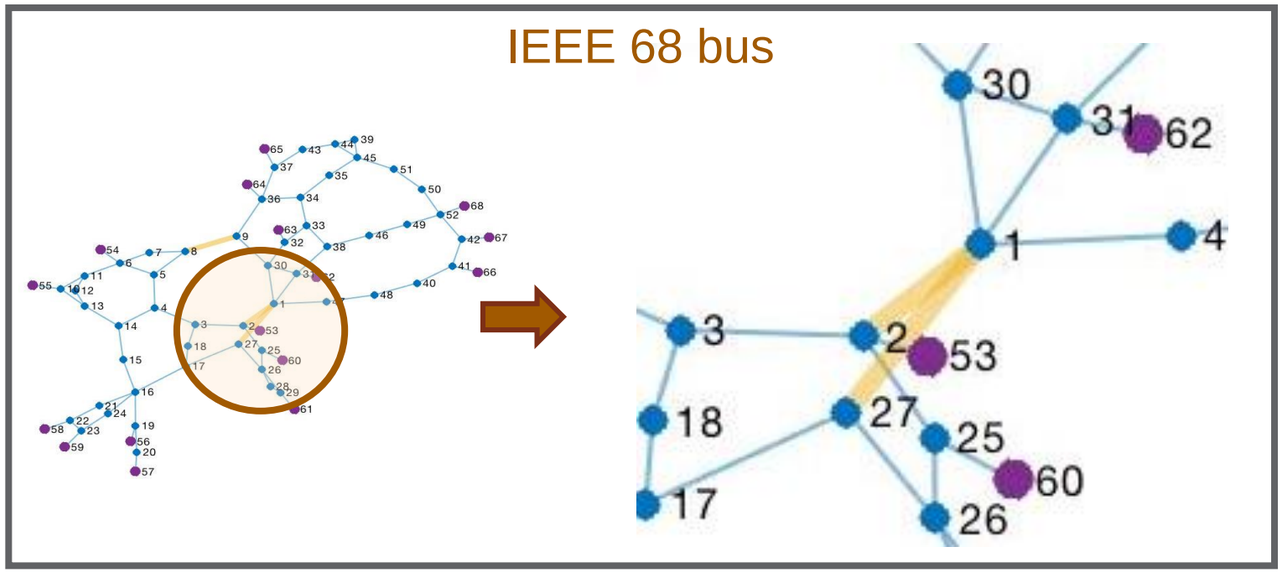
```
n_load_changes = 1; % Number of load changes (== # num of scenarios corresponding to the load changes)
n_lc_events = 1; % Number of load changes in single scenario
% The below load_change_parameters start_time and end_time properties can be either
% scalar or vector [depending on how many load changes are needed in one run].
% %% start_time is chosen randomly
% load_change_parameters.start_time = diag(sort(randi(SimulationTime-15,n_lc_events),'ascend'));
% %% start_time is chosen deterministically
load_change_parameters.start_time = randi(10,1);
% %% load changes are permanent: leave the end_time variable as empty
% %% load changes are temporary: end_time variable is nonempty
load_change_parameters.end_time   = []; % load_change_parameters.start_time + 0.01;
```

Output Data

```
>> PMU

PMU =

  struct with fields:

            Vm: [3000×68 double]
            Va: [3000×68 double]
             f: [3000×68 double]
          fdot: [2999×68 double]
            Im: {1×68 cell}
            Ia: {1×68 cell}
            Id: {1×68 cell}
             P: {1×68 cell}
             Q: {1×68 cell}
    TimeStamps: [3000×1 double]

>> PMU.Id{1}

ans =

  1×5 cell array

    {'I_1_2'}    {'I_1_30'}    {'I_1_31'}    {'I_1_47'}    {'I_1_27'}

>> PMU.Id{2}

ans =

  1×4 cell array

    {'I_2_1'}    {'I_2_3'}    {'I_2_25'}    {'I_2_53'}

>>
```

time-steps

states

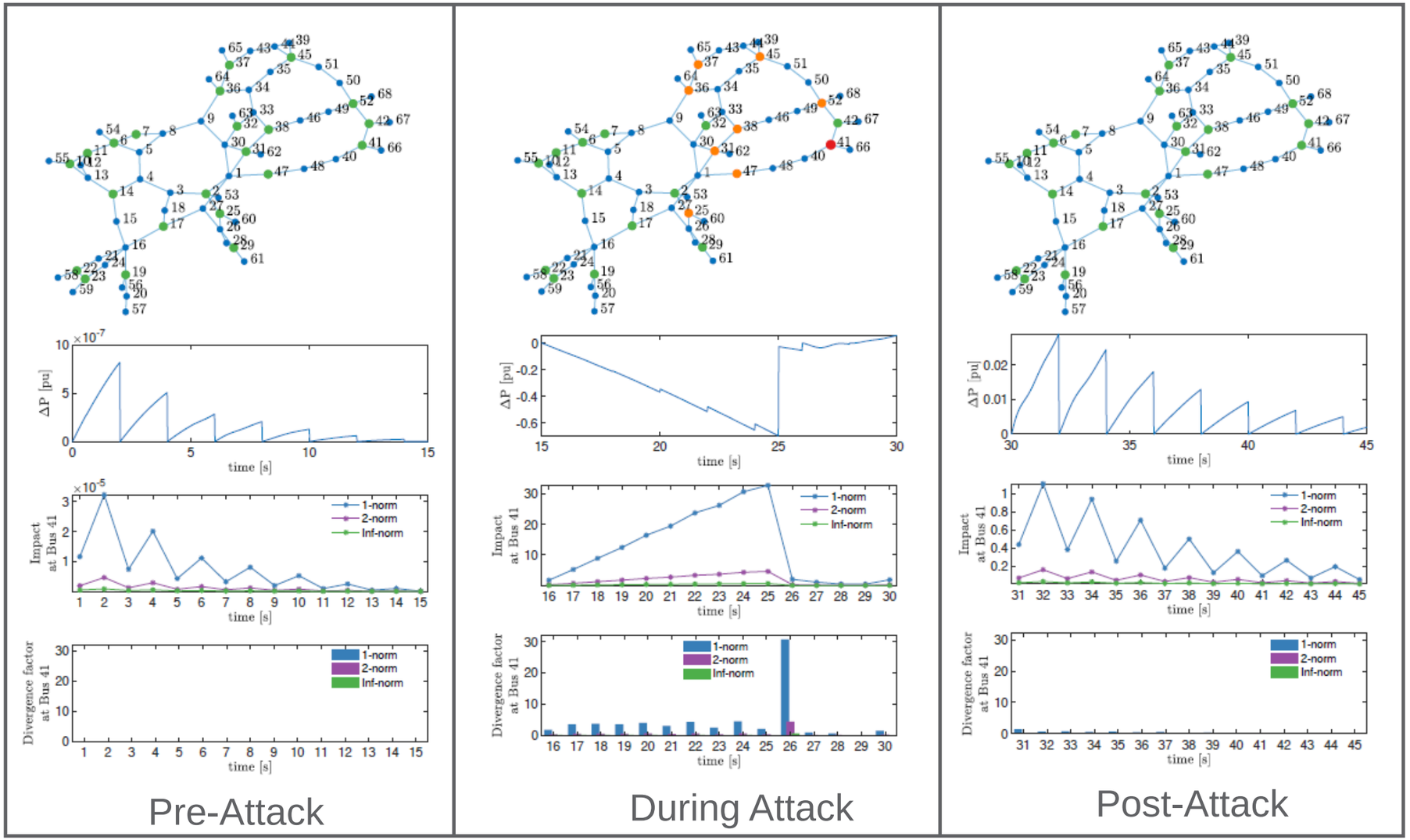



```
>> SCADA

SCADA =

  struct with fields:

    Vm: [3000×68 double]
     P: {1×68 cell}
     Q: {1×68 cell}
     d: {1×68 cell}

>> SCADA.d{1}

ans =

  1×5 cell array

    {'I_1_2'}    {'I_1_30'}    {'I_1_31'}    {'I_1_47'}    {'I_1_27'}
```

Demonstration

Sample Results for Ramp Attack Scenario

- Ramp Attack
- Duration: 15-25 s
- Magnitude: 2%
- Location: Bus 41

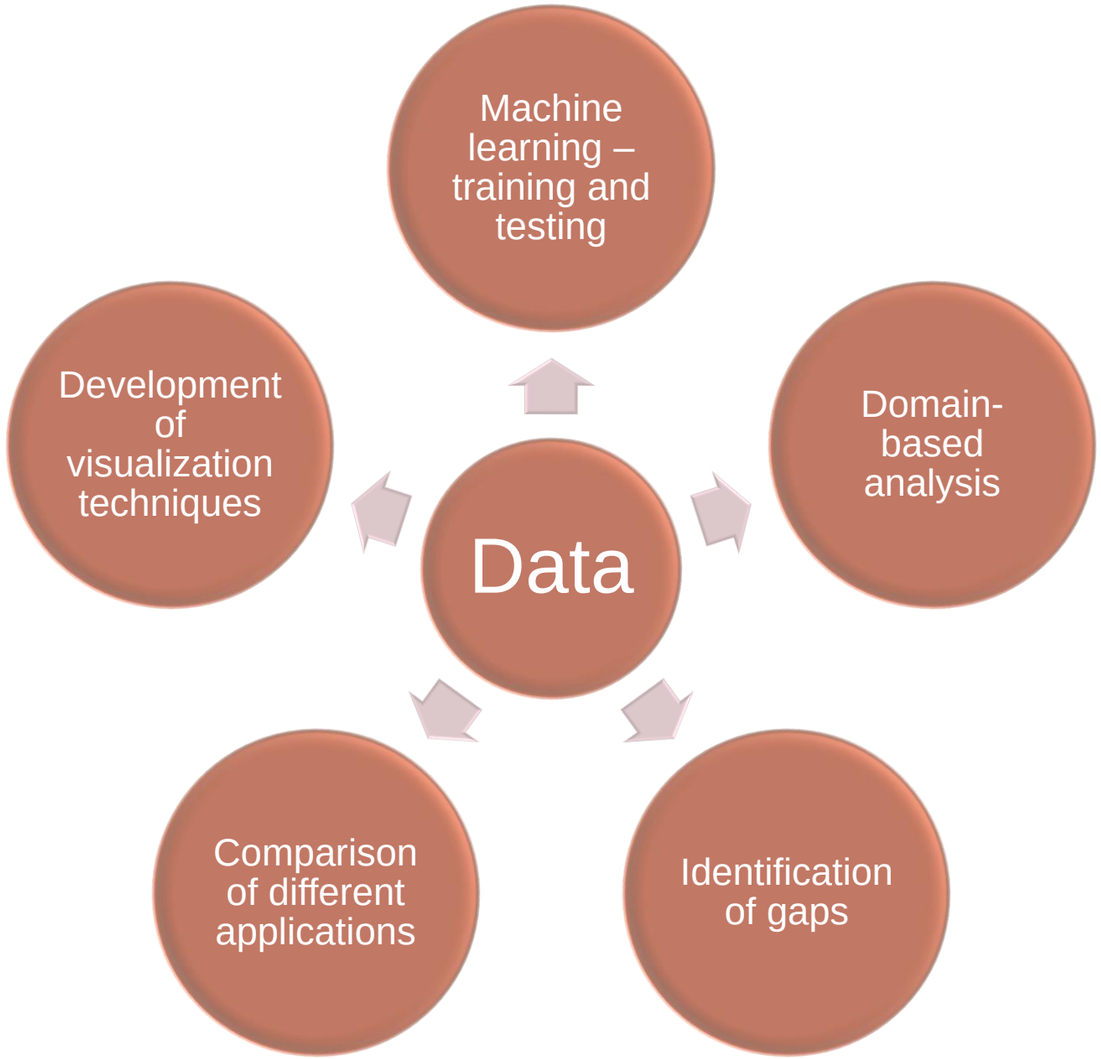


Application of Simulation Framework (example: PowerDrone Project)

Scenario Type	State of the art	Shortcomings	PowerDrone's Application
Normal load changes	AGC control based on slow SCADA measurements (with reporting interval 2-4 s)	SCADA time-step is very large (compared to the dynamics), so detection is slower	Smart RL-based control capable of timely determination of appropriate AGC control irrespective of scenario type using high-frequency PMU measurements
Data Integrity attacks (ramp, step or random)	AGC requires human in the loop to respond with appropriate control	Unable to perform timely detection of adversarial scenarios and needs human to determine AGC based response	
Communication problems (packet-drop or delay injection)	AGC requires human in the loop to respond appropriately	Absence of continuous monitoring or widely accepted mitigation approach	

Applications of the Data Generated

- Applied Koopman Operator Theory for attack detection to compute attack identifiers
- Compared effectiveness of various ML-based classifiers for detecting ramp type of cyber-attack with raw data as well as impact metric data
- Currently developing a RL-based algorithm for determining generator control actions



How You Can Adapt the Framework?

- Pre-generated datasets [for IEEE 68 bus system]
  - with ramp attacks (~1000 scenarios)
  - with step attacks (~1000 scenarios)
  - with (minor) load changes (~200 scenarios)
- Is modular to generate data with other bus systems (subject to PST compatibility)
- Data (system configuration, user-inputs and outputs) of each scenario is appropriately labeled making it usable for different types of applications
- Framework allows creation of wide range of scenarios through different combination of user-defined inputs and this enables wide coverage of data for training and testing
- This code can be extended to include other custom controllers. For example, a controller to modify the generator shaft torque can be implemented inside mpm_sig.m file

Acknowledgements

- We would like to thank the AGM program and Alireza Ghassemian for supporting this project.
- We would also like to thank Prof. Alejandro Dominguez Garcia (UIUC) and Laurentiu D Marinovici (PNNL).

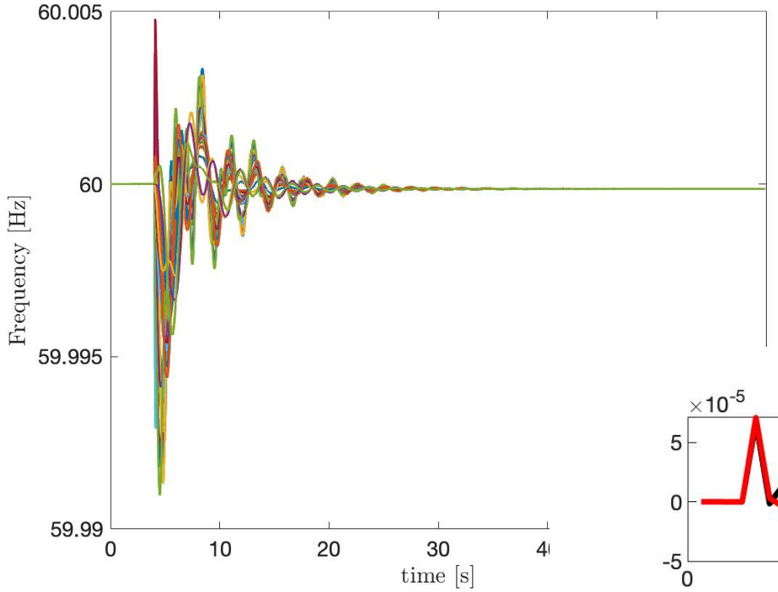
Pacific Northwest
NATIONAL LABORATORY

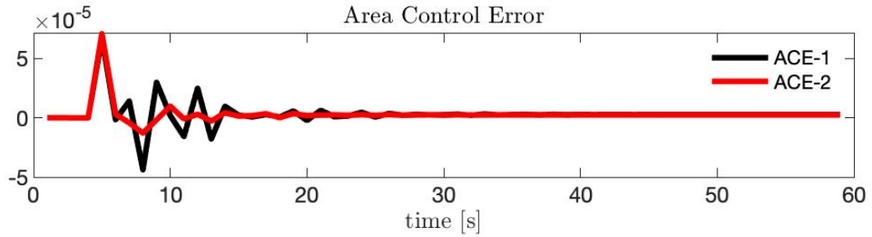
Thank You

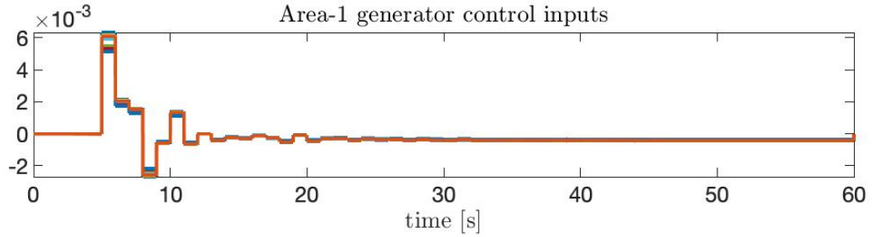
U.S. DEPARTMENT OF ENERGY BATTELLE

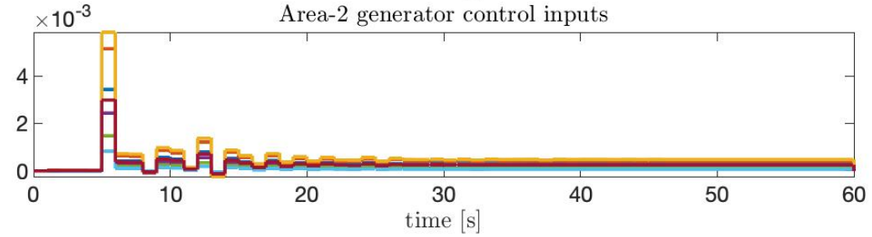
PNNL is operated by Battelle for the U.S. Department of Energy

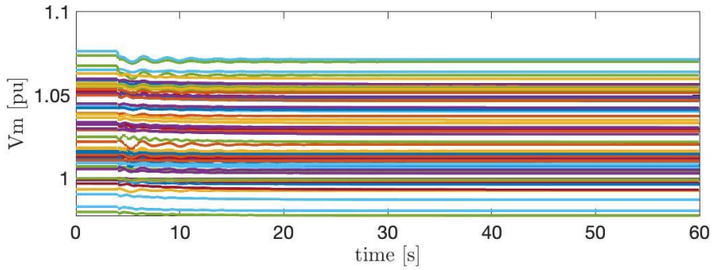
Results for Sample Load Change Scenarios

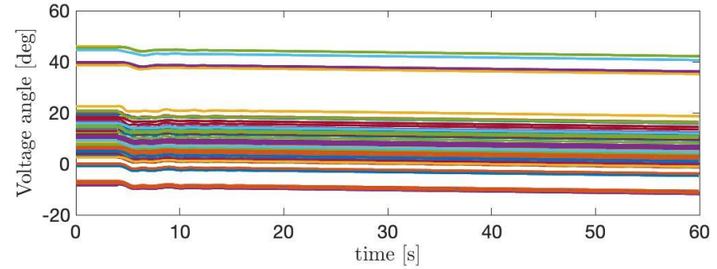




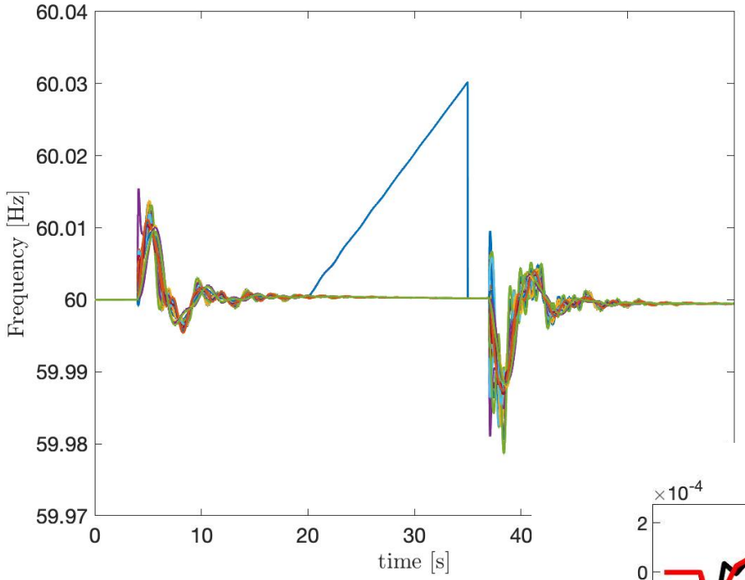


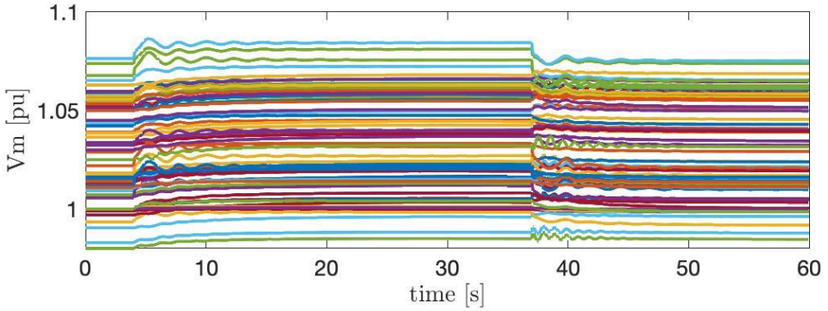


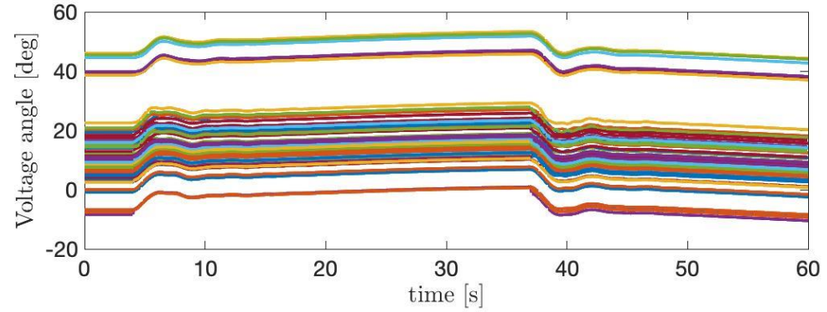


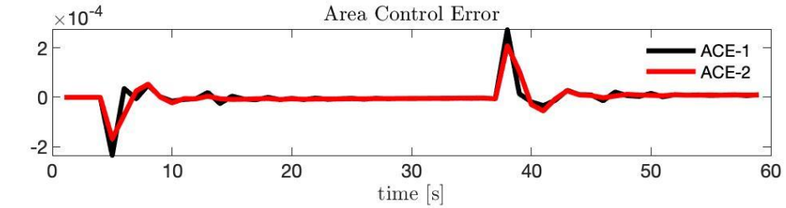


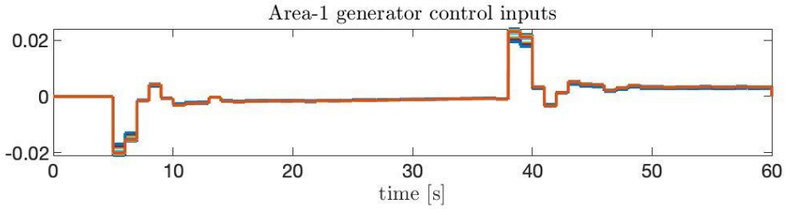
Results for Sample Load Change Scenarios with Ramp attack

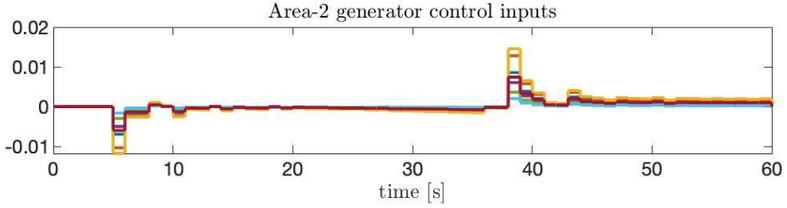












Output Data:

```
[INFO] Load changes: YES
[INFO] Power system stabilizer: NO
[INFO] Automatic generation control: YES
[INFO] PMU attack: YES
[INFO] Type of attack: RAMP
[INFO] Fault: NO FAULT
[INFO] Scenario description saved.
Scenario: 13
[INFO] Attack locations: Buses 4
[INFO] Attack locations: Buses 57
[INFO] Attack magnitudes in percentages: 0.05%
elapsed time = 67.9623s
------------------------*--------------------------

 All the data is saved and the simulation is complete now.
>> |
```

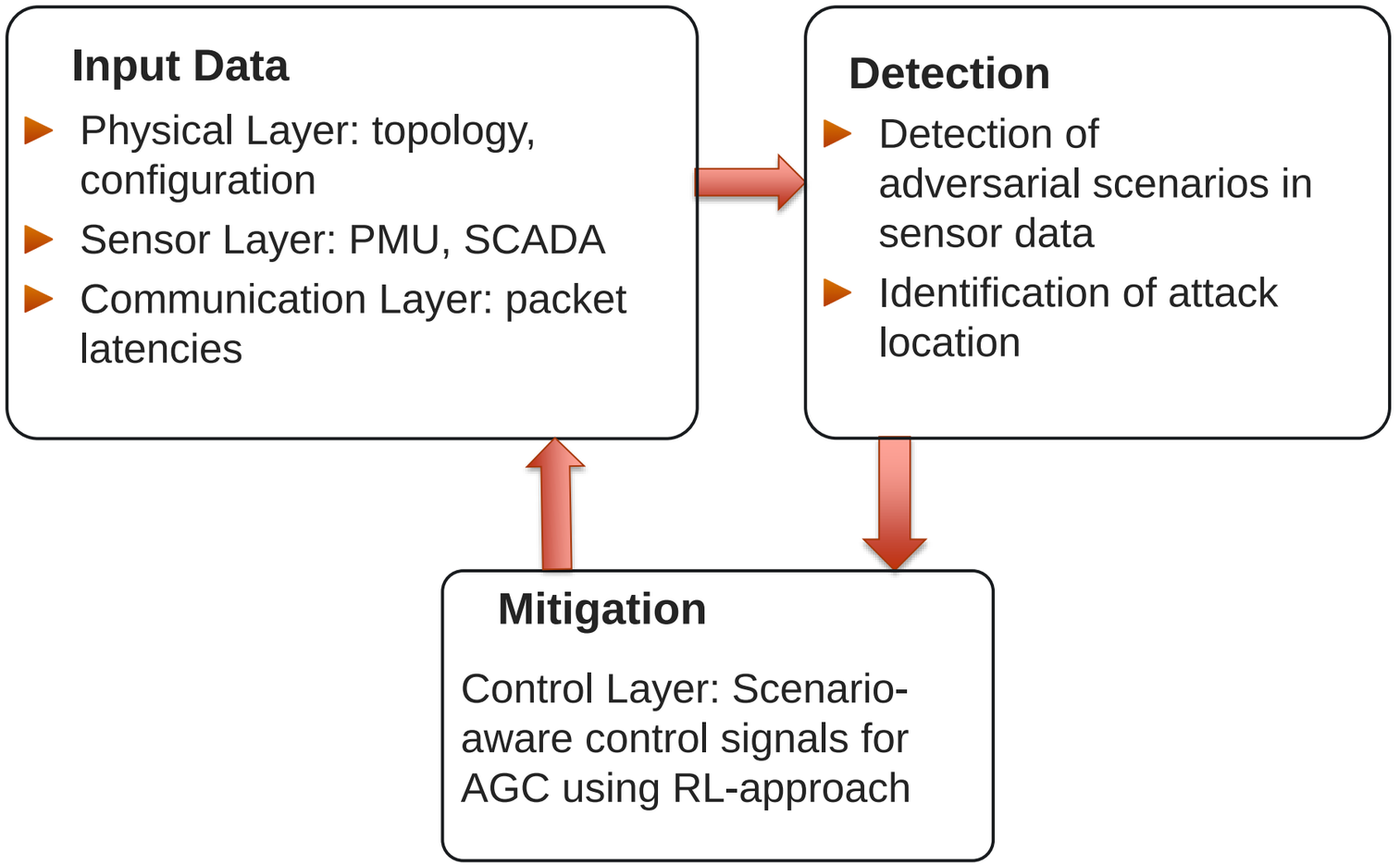
Controller Files Description:

- ml_sig_PD.m
  - modulates active power load at each bus;
  - Computes bus frequencies, voltages, rate of change of bus frequencies at each bus;
  - several attack modules are implemented in this code

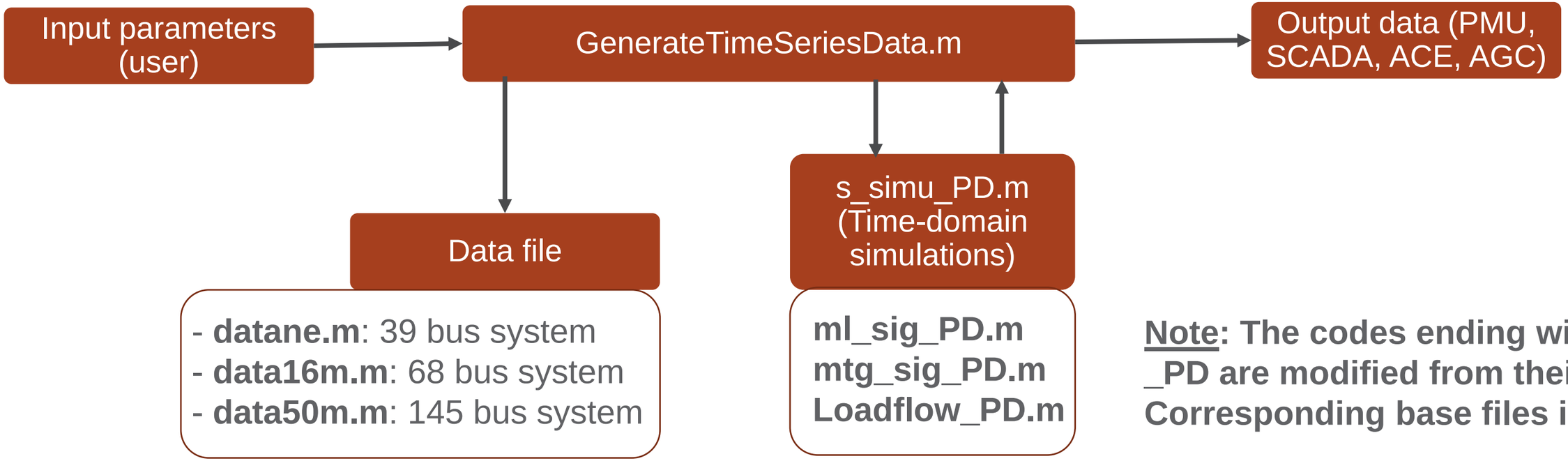
- mtg_sig_PD.m
  - modulates the turbine governor power reference
  - Area control error (ACE) is computed
  - Automatic generation control (AGC) is implemented

- mpm_sig.m – modulates the generator shaft torque
- rml_sig.m – modulates the reactive power load at each bus
- mexc_sig.m – modulates the excitor voltage reference

System Overview



Simulation Framework: Code Organization



Note: The codes ending with _PD are modified from their Corresponding base files in PST

- **GenerateTimeSeriesData.m:** Main code accepting user inputs and configuring batch runs
- **data16m.m:** Data file with bus and connectivity information, generator parameters and sensor location information
- **mtg_sig_PD.m:** modulates the turbine governor power reference; Area Control Error (ACE) computation and AGC implementation
- **ml_sig_PD.m:** modulates active power load at each bus; electrical quantities computation at each bus; implementation of several attack modules