IES
Integrated Energy Systems

Training on Feasible Actuator Range Modifier (FARM)

IES Tools Virtual Workshop:
Capability Overview and Training

March 18, 2022

Haoyu Wang, Roberto Ponciroli, Richard Vilim

Argonne National Laboratory

haoyuwang@anl.gov

Table of Content

- FARM capability overview
- Software installation
- Input creation and running the code
- Output analysis
- Future Directions

1. FARM capability overview

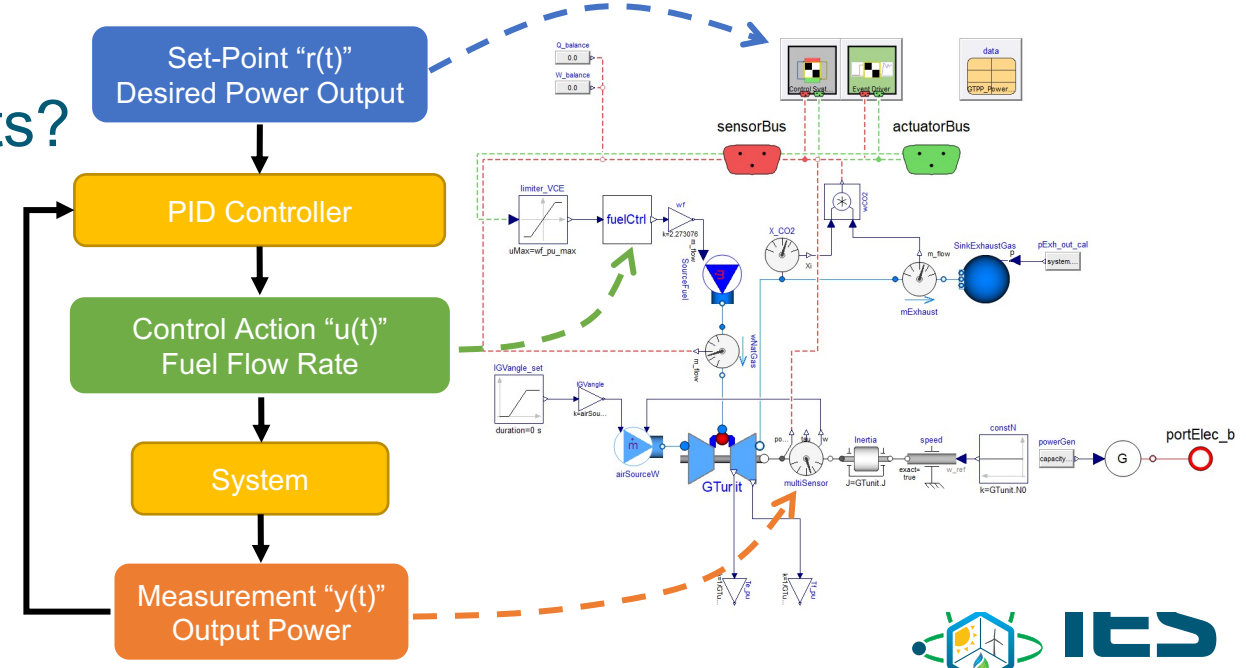
- FARM: Feasible Actuator Range Modifier
  - FARM is a RAVEN plugin to meet the supervisory control needs.
  - FARM helps validate the issued actuator value, to meet both
    - Explicit constraints, and
    - Implicit constraints.

- Q1: What are these constraints?

Let's use a Gas Turbine to explain:

Explicit constraints:
- Power output to grid;
- Power ramp rate, etc.

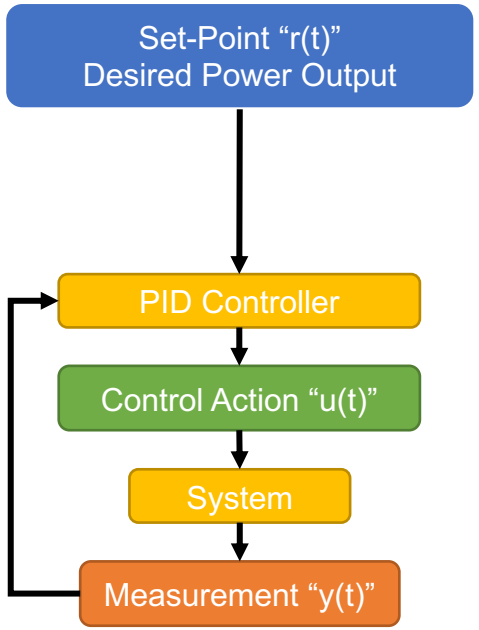
Implicit constraints:
- Firing Temperature, etc.



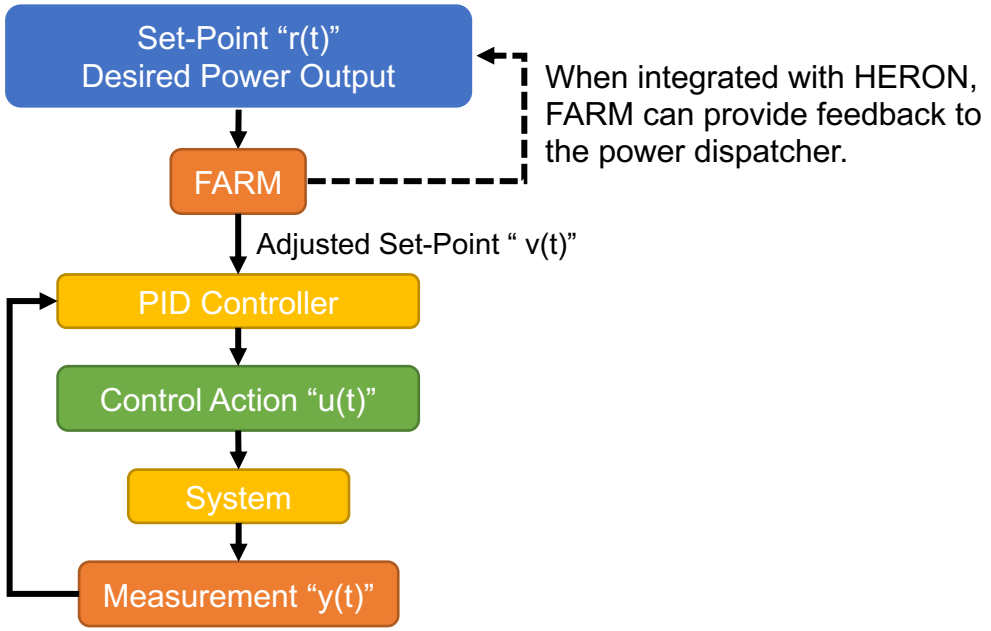
Integrated Energy Systems

1. FARM capability overview

- FARM: Feasible Actuator Range Modifier
  - Q2: Where is FARM in the feedback loop control?



Without FARM

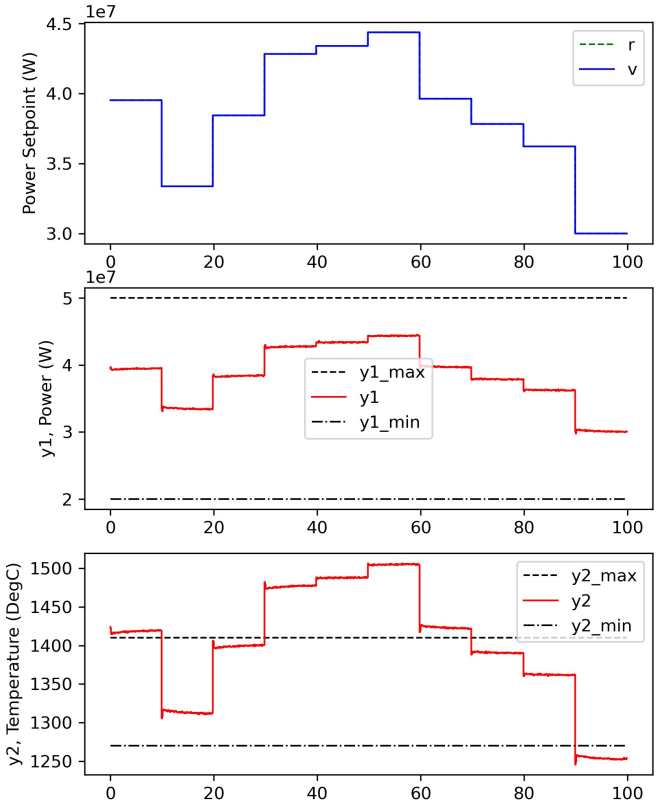


With FARM

IES
Integrated Energy Systems

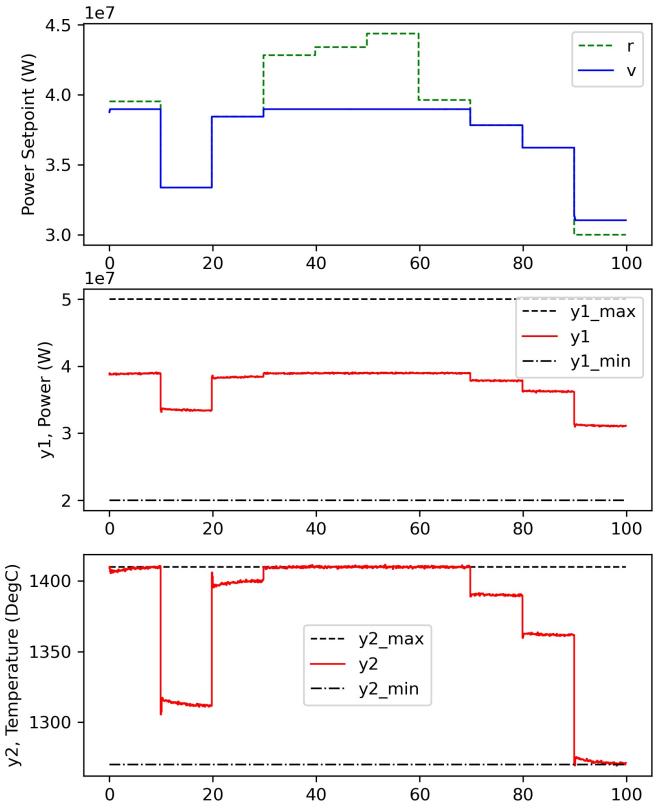
1. FARM capability overview

- FARM: Feasible Actuator Range Modifier
  - Q3: What's the effects of FARM?



Without FARM:
- System run on original power setpoint
- Implicit constraints were violated (Firing temperature)

With FARM:
- Power setpoint was adjusted
- Implicit constraints were met (Firing temperature)



Integrated Energy Systems

2. Software installation

- FARM is an open-source software
  - https://github.com/Argonne-National-Laboratory/FARM
  - In order to run FARM, RAVEN is a pre-requisite.
- FARM installation consists of 2 steps:
  - Step 1: Download FARM source code using git

```
haoyuwang@p075722 MINGW64 /d/GitProjects/training
$ git clone https://github.com/Argonne-National-Laboratory/FARM.git
```

  - Step 2: Register FARM plugin in RAVEN

```
haoyuwang@p075722 MINGW64 /d/GitProjects/training/raven (devel)
$ ./scripts/install_plugins.py -s /d/GitProjects/training/FARM/
```

3. Input creation and running the code

- FARM uses XML file as input
  - One example is in

    FARM / training031822 / FARM_para_SES.xml
  - We will focus on some key entries.

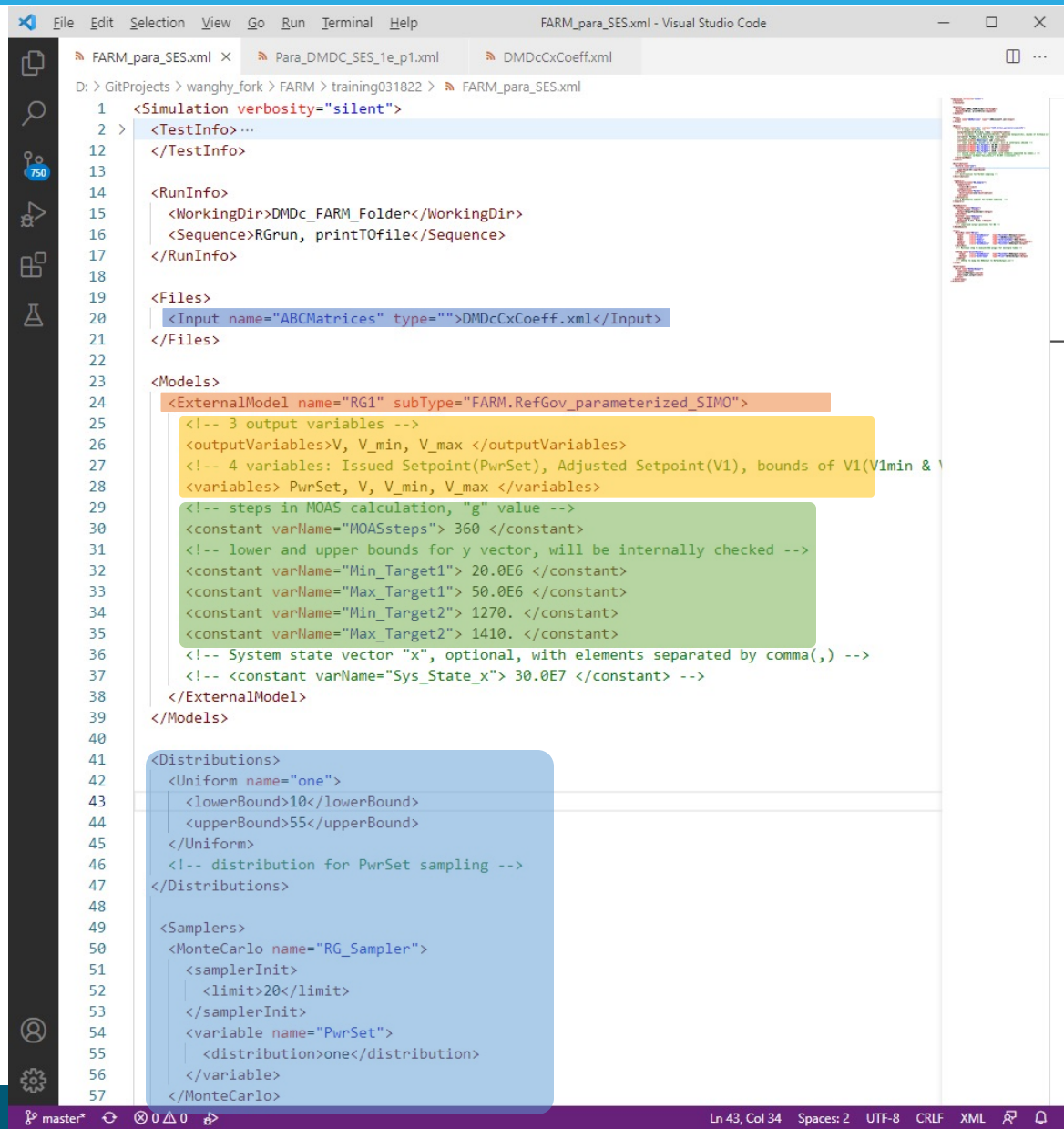
3.1. An XML file containing the state-space representation matrices;

3.2. FARM external model name;

3.3. Input and output variables for FARM;

3.4. Prediction time horizon, and operational constraints;

3.5. Random number generator for input variables creation



3. Input creation and running the code

- 3.1. An XML file containing the state-space representation matrices
  - A state-space matrix set [A,B,C,D] is required to describe the system.
  - Can be generated through RAVEN DMDc*.
  - One example is available at

    FARM / training031822 / Para_DMDC_SES_1e_p1.xml

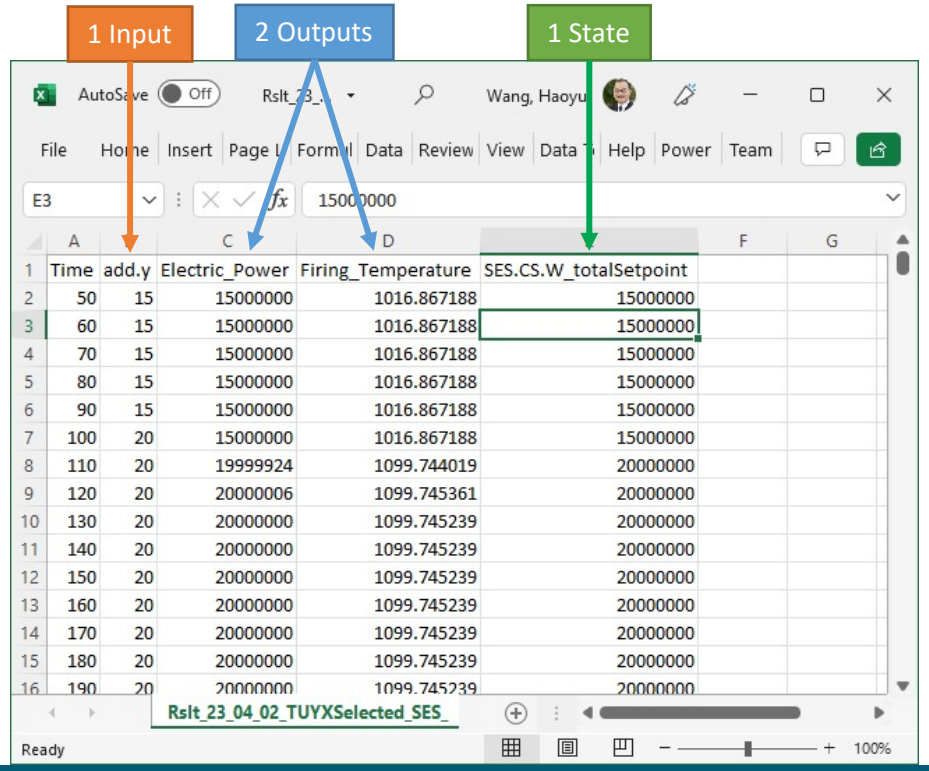
Input $v \in \mathbb{R}^m$

State-space model

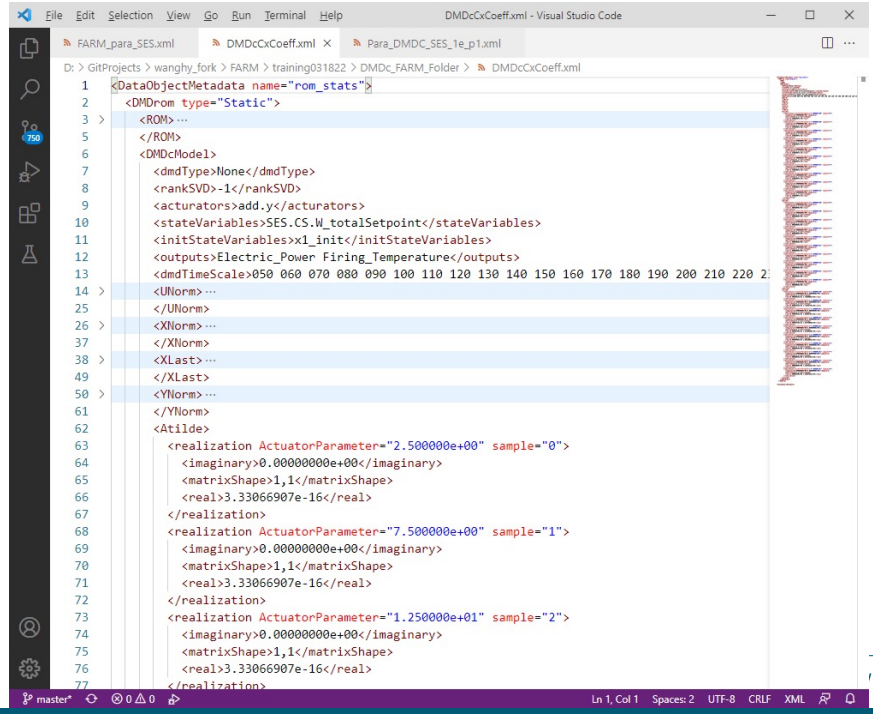
$$x(k+1) = Ax(k) + Bv(k)$$
$$y(k) = Cx(k) + Dv(k)$$

Output $y \in \mathbb{R}^p$

State $x \in \mathbb{R}^n$



RAVEN DMDc



*For more details, please refer to RAVEN user manual Section 15.3.11, DMDc

3. Input creation and running the code

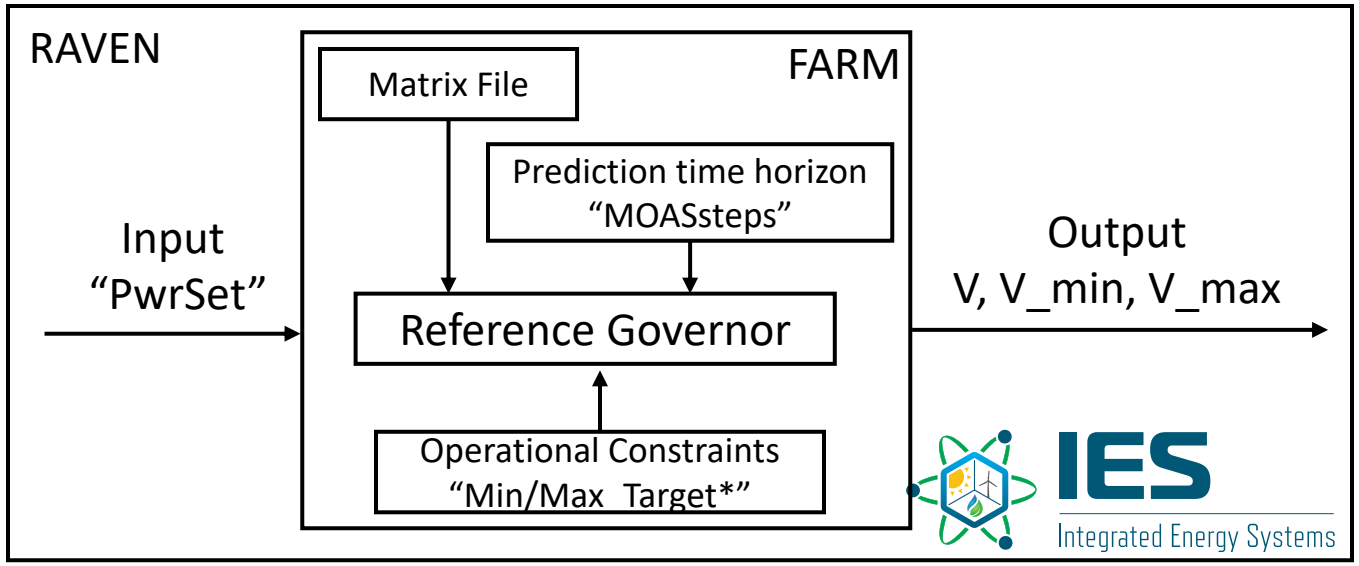
- 3.2. FARM external model name;
  - To use FARM, “FARM.RefGov_parameterized_SIMO” need to be specified as the external model.
  - Source code* is available at
    FARM / src / RefGov_parameterized_SIMO.py
- 3.3. Input and output variables for FARM;
  - Input: “PwrSet”, the power setpoint before any adjustment;
    - “PwrSet” should share the same unit as the actuator signal in DMDc training data;
  - Output: “V”, adjusted power setpoint; “V_min” and “V_max”, the min & max allowed V value.
- 3.4. “MOASsteps” for the prediction time horizon;
  - $\text{MOASsteps} = \frac{\text{Time Horizon}}{\text{Matrices interval}}$
  - Example:
    - To predict the response for 1 hour;
    - Matrices are in 10s interval
    - MOASsteps = 3600s / 10s = 360.

IES
Integrated Energy Systems

*For more details, please refer to FARM user manual Section 1.1

3. Input creation and running the code

- 3.4. “Min/Max_Target*” for the operational constraints;
  - “Min_Target_i” and “Max_Target_i” defines the bounds for the i^{th} system output y_i.
  - Example:
  - In training data, y_1 is Electric Power (W), y_2 is Firing Temperature (℃)
  - Then in the FARM input file,
    - Min_Target_1=20.0E6, Max_Target_1=50.0E6 → 20.0MW < Electric Power < 50.0MW
    - Min_Target_2=1270.0, Max_Target_2=1410.0 → 1270℃ < Firing Temperature < 1410℃
  - Mind the units.
- The structure of entire FARM Plugin:



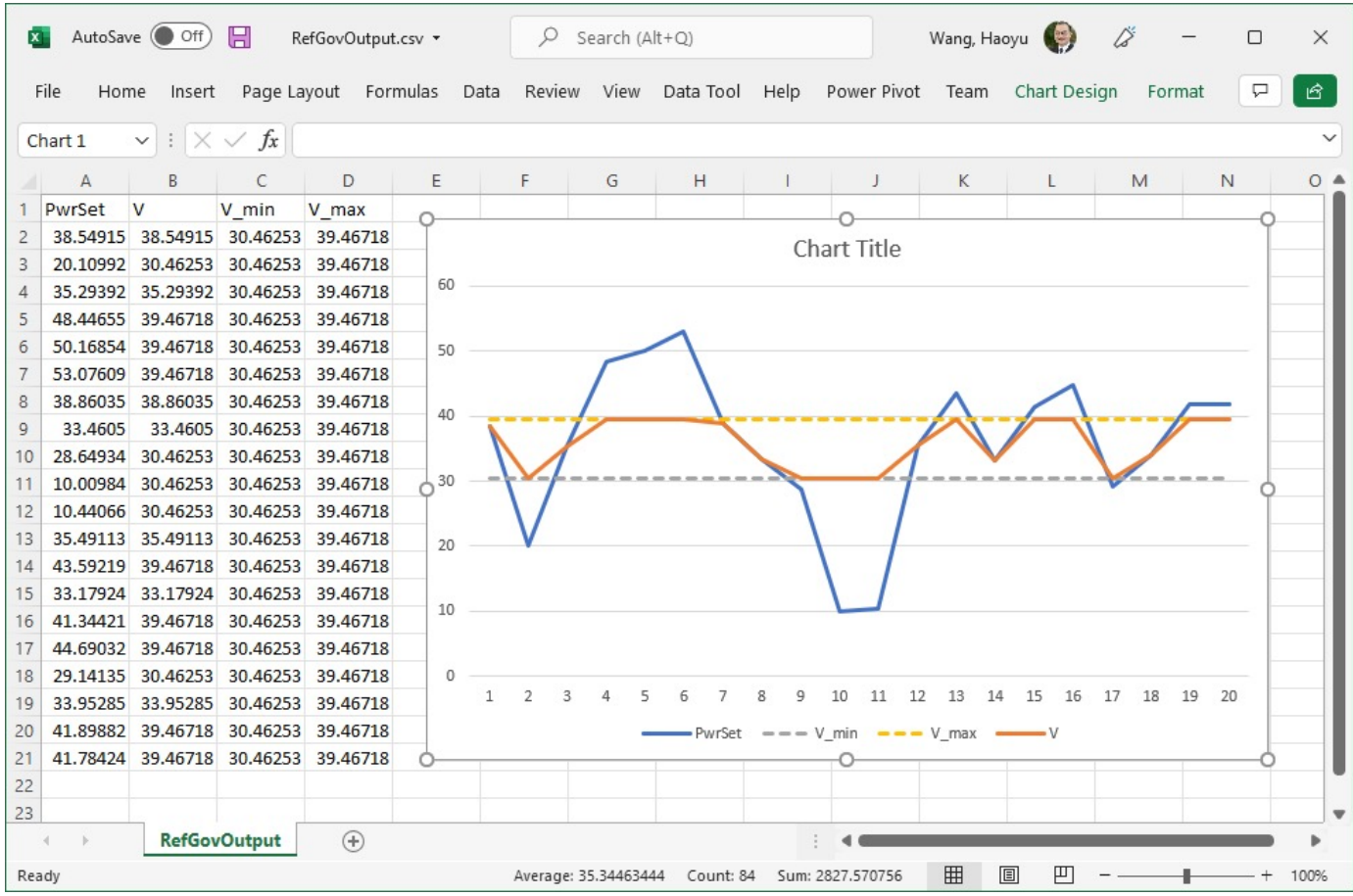
3. Input creation and running the code

- The FARM input file can be executed like other RAVEN input files:

```
haoyuwang@p075722 MINGW64 /d/GitProjects/training/FARM (master)
$ ../raven/raven_framework training031822/FARM_para_SES.xml
```

4. Output analysis

- The FARM output can be found in:
  - FARM / training031822 / DMDc_FARM_Folder / RefGovOutput.csv
    - 20 entries, with 4 column in each entry
      - Issued power setpoint “PwrSet”;
      - Adjusted power setpoint “V”;
      - Minimum allowed value “V_min”;
      - Maximum allowed value “V_max”;
    - The “PwrSet” are regulated to “V”, to meet both explicit and implicit constraints.



5. Future Directions

- FARM is being implemented into HERON to help with power dispatch problem. [1]
- Online system identification and matrices update (ETA: April 2022)
  - User do not need to generate matrices off-line;
  - Online data-driven derivation and update of A,B,C,D matrices;
  - Better supports the physics-based high-fidelity model.

References

[1] Wang, Haoyu, Roberto Ponciroli, and Richard B. Vilim. Automation of FARM from Alpha Phase to Beta Phase. No. ANL/NSE-22/6. Argonne National Lab.(ANL), Argonne, IL (United States), 2022.

IES
Integrated Energy Systems

IES

Integrated Energy Systems

Thank you!

Haoyu Wang

Argonne National Laboratory

haoyuwang@anl.gov