SURROGATE MODELS FOR REAL TIME MULTI-ASSET CONTROL

HAOYU WANG, ROBERTO PONCIROLI, RICHARD VILIM
Argonne National Laboratory

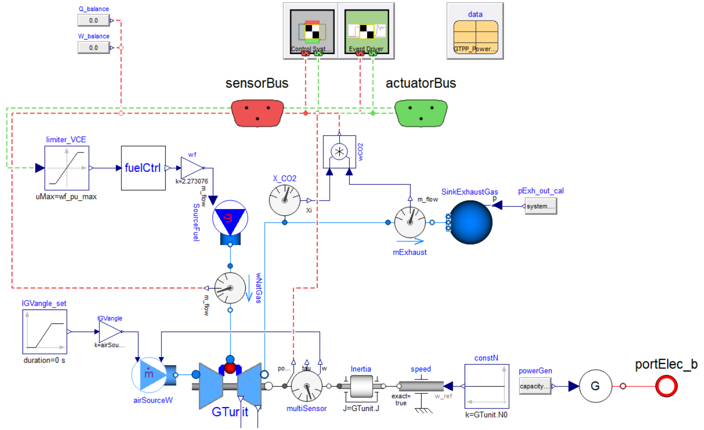
ANDREA ALFONSI
Ultra Safe Nuclear Corporation

Advanced Reactor Controls Workshop
July 12-14, 2023, Argonne National Laboratory, Lemont IL

OBJECTIVES

- Real-time simulation of Digital Twins is mandatory when introducing Advanced Reactors to electrical grid.
- Digital twins of various complexity levels:
  - High-fidelity model for off-line applications, which can be slow in execution
    - Exploration of the operational margin;
    - Optimization of design parameters, etc.
  - Surrogate model for on-line applications, which needs to be fast in execution
    - Real-time control and performance optimization;
    - Economic dispatch optimization, etc.
- We introduce a novel data-driven method to derive surrogate models from High-Fidelity Digital Twins, to be used by advanced control algorithms, for real-time multi-asset control.

WHAT IS SURROGATE MODEL?



Input variables $\vec{u}$

System state $\vec{x}$ evolution

$$\vec{x}(k+1) = A^d\vec{x}(k) + B^d\vec{u}(k)$$

$$\vec{y}(k) = C^d\vec{x}(k) + D^d\vec{u}(k)$$

Output variables $\vec{y}$

- High Fidelity Digital Twin (HFDT):
  - Solving for physics-based equations, high-resolution results;
  - Tracking of 1,000+ process variable response;
  - Long execution time (10s to simulate 1 hour);
  - Not ideal for repetitive execution.

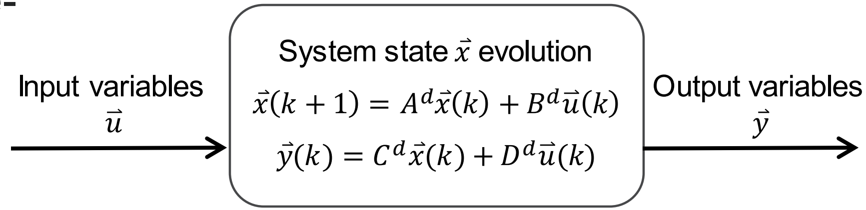
- Surrogate Model:
  - Derived from physical HFDT, use linear functions to capture system dynamics;
  - Predict a few state / output variables with physical meanings;
  - Fast in execution (1ms to simulate 1 hour);
  - Ideal for repetitive execution.

How to derive?

Argonne NATIONAL LABORATORY

CHALLENGES AND SOLUTION

- State variables (x) selection is the key in state-space model derivation.
  - Challenges in state variable selection:
    - Data only, no physics-based equations;
    - 1,000+ candidates, impossible to test all possible combinations;
    - Need a consistent set of variables to cover all known operational conditions.



- Developed solution: RFE-DMDc
  - Recursive Feature Elimination (RFE):
    - Identify the most impactful candidates, reduce the search domain;
  - Dynamic Mode Decomposition with Control (DMDc)
    - Derive the dynamics through data;
  - Cross correlation
    - Resolve cross-talks between multiple assets.

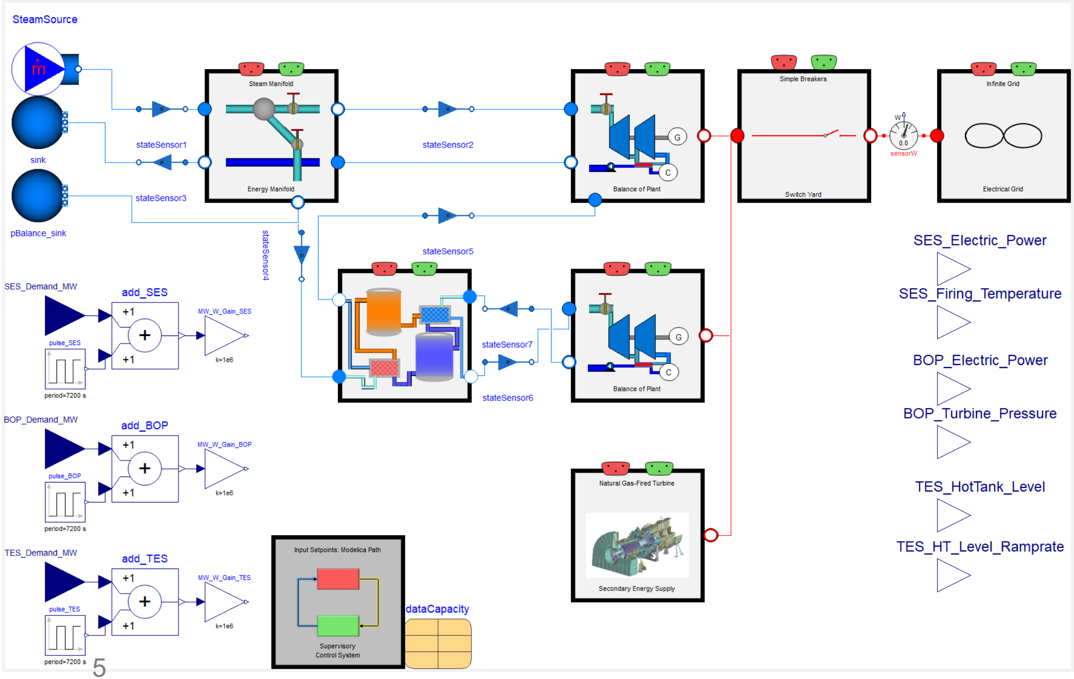
TEST-CASE AND DATA

A thermally coupled power system simulator is used as the HFDT.

Three sub-systems:
- Balance of Plant;
- Thermal Energy Storage;
- Gas Turbine.

Variables:
- 3 inputs;
- 6 outputs;
- 7,087 recorded process variables.

Thermal couplings between two sub-systems



Argonne NATIONAL LABORATORY

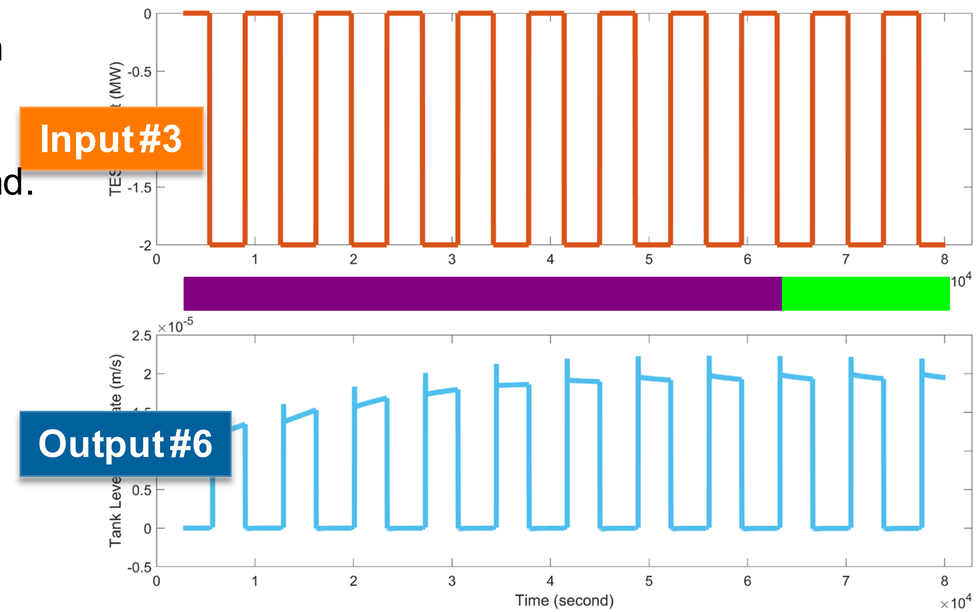
TEST-CASE AND DATA

Power transient data under 72 possible operational conditions are simulated through HFDT:

- Each case contains 22-hour transient;
- Execution time of each case ~110 second.

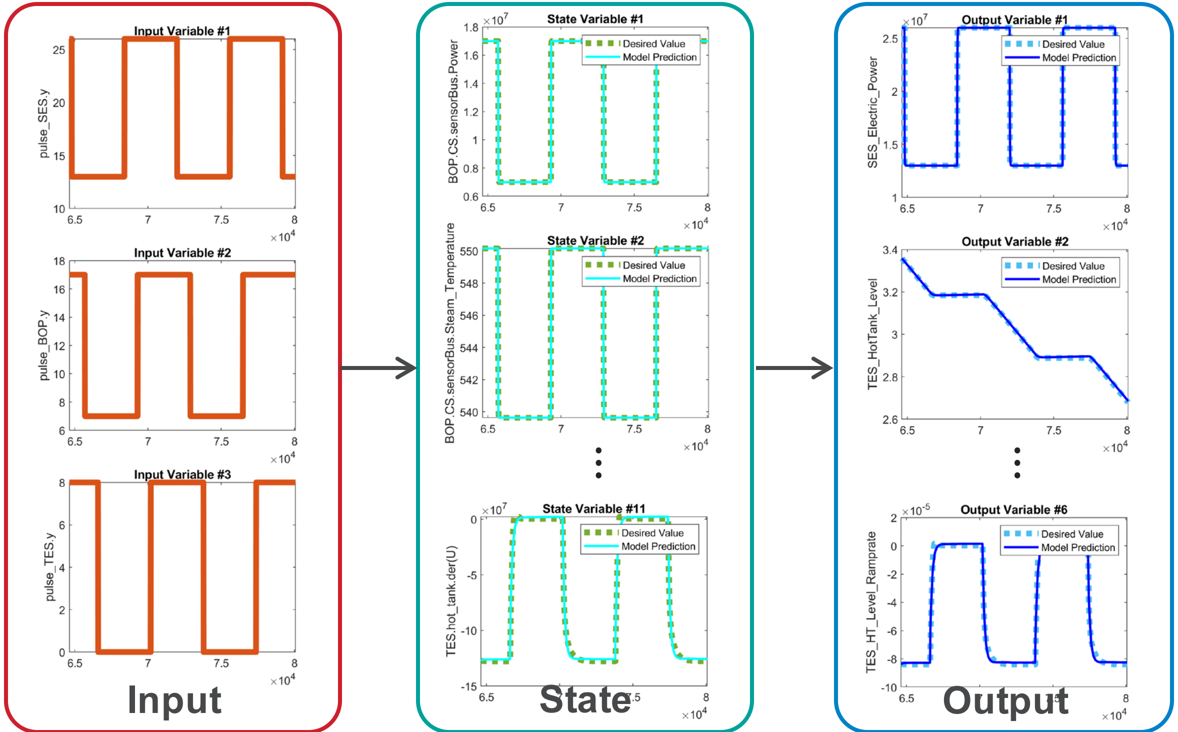
Training and testing Scheme:

- Train on the first 80% of the data
- Test on the final 20% of the data



PREDICTION ACCURACY OF SURROGATE MODEL

- A surrogate model with 11 selected state variables are built, using Training data only;
- This model provides accurate prediction of the state and output responses, but 6,000 times faster;



- Time to reconstruct the final 4.4-hour response: 3.2 ms
- In contrast, HFDT takes 22 seconds

CONCLUSIONS

- A data-driven approach was developed to derive surrogate models for real-time multi-asset control.
- The surrogate model learns from cold-start and steady data, and provide accurate prediction on key process variables.
- Computational burden can be reduced by 6,000 ~ 7,000 times, which made it deployable for massive repetitive executions.
- Such surrogate model can be deployed in advanced control loop, which aids the real-time performance optimization, and improves the efficiency and flexibility of operation.

THANK YOU

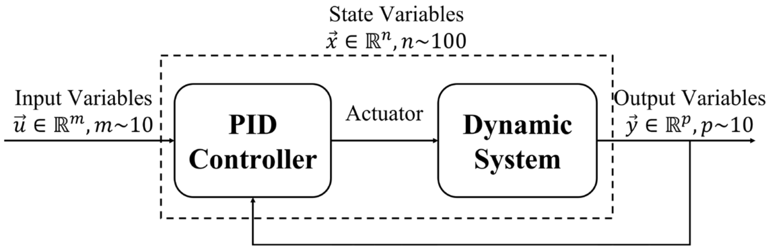
www.anl.gov

Argonne
NATIONAL LABORATORY

BACK-UP SLIDES

STATE VARIABLE SELECTION ALGORITHM: DMDC-RFE

Dynamic Mode Decomposition with Control (DMDc)



Discrete Time Domain:

$$\vec{x}(k+1) = A^d\vec{x}(k) + B^d\vec{u}(k)$$

$$\vec{y}(k) = C^d\vec{x}(k) + D^d\vec{u}(k)$$

1. Derived A^d and B^d matrices represent the best-fit, the first linear equation does not hold exactly:

$$\vec{x}(k+1) \approx A^d\vec{x}(k) + B^d\vec{u}(k)$$

2. A^d and B^d matrices can be constructed by collecting temporal snapshots of the system states $\vec{x}(k)$ and inputs $\vec{u}(k)$ over l time steps:

$$\mathrm{X}' = \begin{bmatrix} | & | & & | \\ x(2) & x(3) & \cdots & x(l) \\ | & | & & | \end{bmatrix}$$

$$\mathrm{X} = \begin{bmatrix} | & | & & | \\ x(1) & x(2) & \cdots & x(l-1) \\ | & | & & | \end{bmatrix}$$

$$\mathrm{U} = \begin{bmatrix} | & | & & | \\ u(1) & u(2) & \cdots & u(l-1) \\ | & | & & | \end{bmatrix}$$

3. Rewrite the equation in a matrix form to include the new data matrices:

$$\mathrm{X}' \approx A^d\mathrm{X} + B^d\mathrm{U}$$

Argonne NATIONAL LABORATORY

STATE VARIABLE SELECTION ALGORITHM: DMDC-RFE

Dynamic Mode Decomposition with Control (DMDc)

4. Concatenate using $G = [A^d, B^d]$ and $\Omega = [X, U]^T$:

$$\mathrm{X}' \approx G\Omega$$

5. Perform singular value decomposition (SVD) to Ω:

$$\Omega = \mathrm{U}\Sigma\mathrm{V}^* \approx \tilde{\mathrm{U}}\tilde{\Sigma}\tilde{\mathrm{V}}^*$$

where $\tilde{\mathrm{U}} \in \mathbb{R}^{(n+m)\times q}$, $\tilde{\Sigma} \in \mathbb{R}^{q\times q}$, and $\tilde{\mathrm{V}} \in \mathbb{R}^{(l-1)\times q}$ are the truncated SVD components.

6. Since $\tilde{\mathrm{U}}$ and $\tilde{\mathrm{V}}$ are unitary matrices, $\tilde{\Sigma}$ is diagonal:

$$G \approx \mathrm{X}'\tilde{\mathrm{V}}\tilde{\Sigma}^{-1}\tilde{\mathrm{U}}^*$$

7. Breaking the linear operator $\tilde{\mathrm{U}}$ according to the dimensions of $\vec{x}(k)$ and $\vec{u}(k)$

$$[\tilde{A}^d \quad \tilde{B}^d] \approx [\mathrm{X}'\tilde{\mathrm{V}}\tilde{\Sigma}^{-1}\tilde{\mathrm{U}}_1^* \quad \mathrm{X}'\tilde{\mathrm{V}}\tilde{\Sigma}^{-1}\tilde{\mathrm{U}}_2^*]$$

8. The C^d, D^d matrices in the second equation

$$\vec{y}(k) = C^d\vec{x}(k) + D^d\vec{u}(k)$$

can be treated in the similar way.

But if D^d matrix is assumed to be zero in the studied system:

$$\vec{y}(k) = C^d\vec{x}(k)$$

9. Construct the matrix of temporal snapshot of system output vector $\vec{y}(k)$:

$$\mathrm{Y} = \begin{bmatrix} | & | & & | \\ y(1) & y(2) & \cdots & y(l-1) \\ | & | & & | \end{bmatrix}$$

10. C^d matrix can be found through pseudo inverse:

$$\tilde{C}^d \approx \mathrm{Y}\mathrm{X}^{-1}$$

STATE VARIABLE SELECTION ALGORITHM : DMDC-RFE

DMDc with Recursive Feature Elimination (DMDc-RFE)

Stage 1, Pre-Filtering:

1,000+ process variables → 10s candidates

1.a. Perform pair-wise linear correlation analysis for all the recorded variables, and remove:

- Constant variables, and
- Mutually linear-correlated variables;

1.b. Perform an iterative ranking and feature elimination:

- Construct an evaluation model on a complete feature space $\tilde{C}^d \approx \mathrm{YX}^{-1}$
- Compute a ranking criterion showing the importance of each variable;
- Remove the variable with lowest importance iteratively.
- A set of candidates is derived $[x_1, x_2, \dots, x_n]$

1.c. For systems with multiple sub-systems:

- Perform 1.b to assess each sub-system independently:

$$\tilde{C}^{d,A} \approx Y^A (X^A)^{-1}$$
$$\tilde{C}^{d,B} \approx Y^B (X^B)^{-1}$$

and two sets of candidates are derived (x^A and x^B):

$$x^A = [x_1^A, x_{2,}^A \dots, x_n^A]$$
$$x^B = [x_1^B, x_{2,}^B \dots, x_n^B]$$

- The two sets of state variable are analyzed with a cross-correlation technique, to account for the cross-talk between sub-systems

(A)		(B)
$x^A = \begin{bmatrix} x_1^A \\ x_{2,}^A \\ \vdots \\ x_n^A \end{bmatrix}$	$\rightleftarrows$	$x^B = \begin{bmatrix} x_1^B \\ x_{2,}^B \\ \vdots \\ x_n^B \end{bmatrix}$

STATE VARIABLE SELECTION ALGORITHM : DMDC-RFE

DMDc with Recursive Feature Elimination (DMDc-RFE)

Stage 2, Parallel searching :

10s candidates → Optimal set of State variables

Parallel looping through all possible combinations of variables from the candidates of $[x^A, x^B, x^{A\rightarrow B}, x^{B\rightarrow A}]$. For each combination:

2.a. Derive A,B,C matrices through DMDc;

2.b. Evaluate the predicted state variables $\hat{x}(k \geq 0)$ and the output variables $\hat{y}(k \geq 0)$:

$$\hat{x}(0) = \vec{x}(0)$$

$$\hat{x}(k+1) = \tilde{A}^d \hat{x}(k) + \tilde{B}^d \vec{u}(k), k \geq 0$$

$$\hat{y}(k) = \tilde{C}^d \hat{x}(k)\,, k \geq 0$$

2.c. Calculate the cost function:

- Mean square error (MSE) of predicted state $\hat{x}(k \geq 1)$ and the known state $\vec{x}(k \geq 1)$, and
- MSE of the predicted output $\hat{y}(k \geq 1)$ and the known output $\vec{y}(k \geq 1)$.

$$\text{Cost function} = \frac{1}{n \cdot l} \sum_{j=1}^{n} \sum_{k=1}^{l} \left(\frac{\hat{x}_j(k) - \vec{x}_j(k)}{\sigma(\vec{x}_j)} \right)^2 + \frac{1}{p \cdot l} \sum_{j=1}^{p} \sum_{k=1}^{l} \left(\frac{\hat{y}_j(k) - \vec{y}_j(k)}{\sigma(\vec{y}_j)} \right)^2$$

Finally, selects the combination that minimizes the cost function, as the result.