SAND2021-10947C

ERI ELECTRONICS RESURGENCE INITIATIVE SUMMIT & MTO Symposium

Xyce Circuit Simulator and Surrogate Modeling

Eric Keiter, Jason Verley, Heidi Thornquist, Paul Kuberry, Edgar Galvan, Sandia National Laboratories

Posh Open Source Hardware (POSH)

Mitigating Design Costs

Topic

Parallel Circuit Simulator →

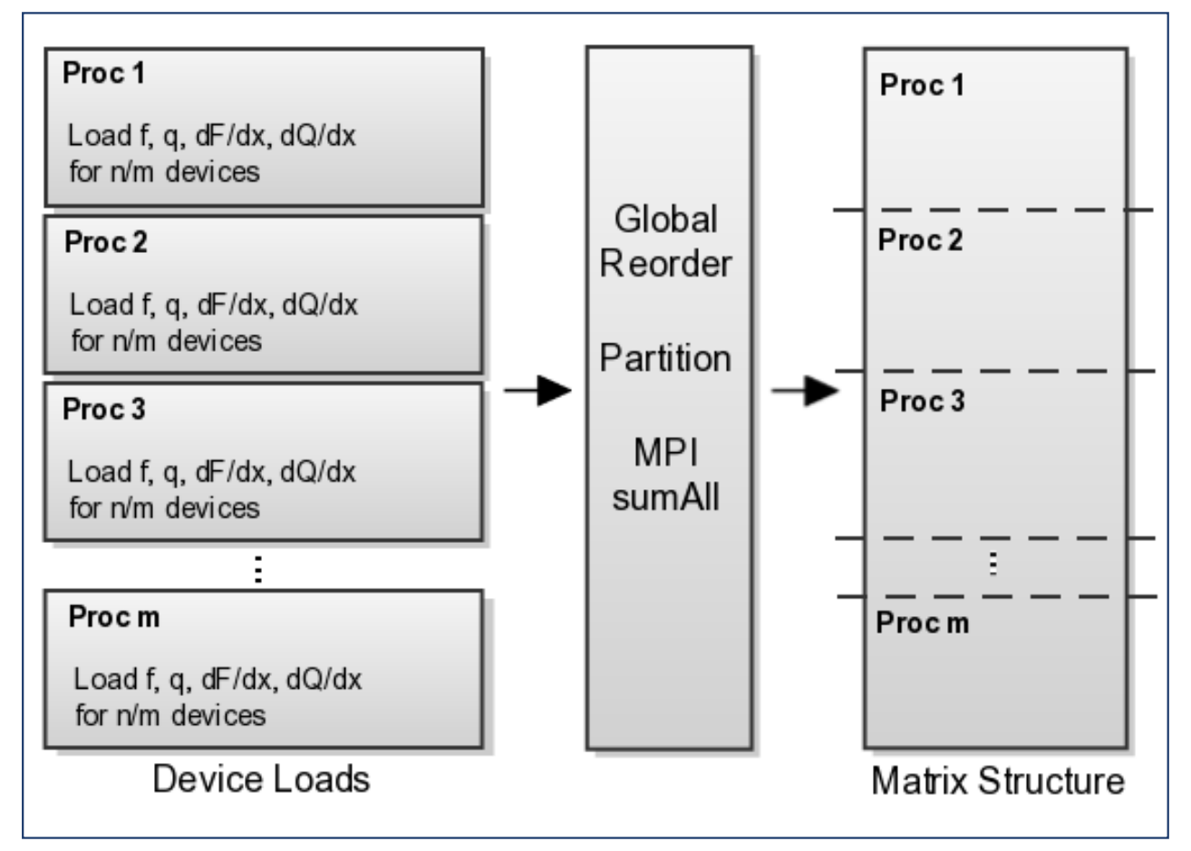
New Xyce Feature: PyMI: Python model interface →

… **which enables:** Surrogate models via Dynamic Mode Decomposition (DMD) →

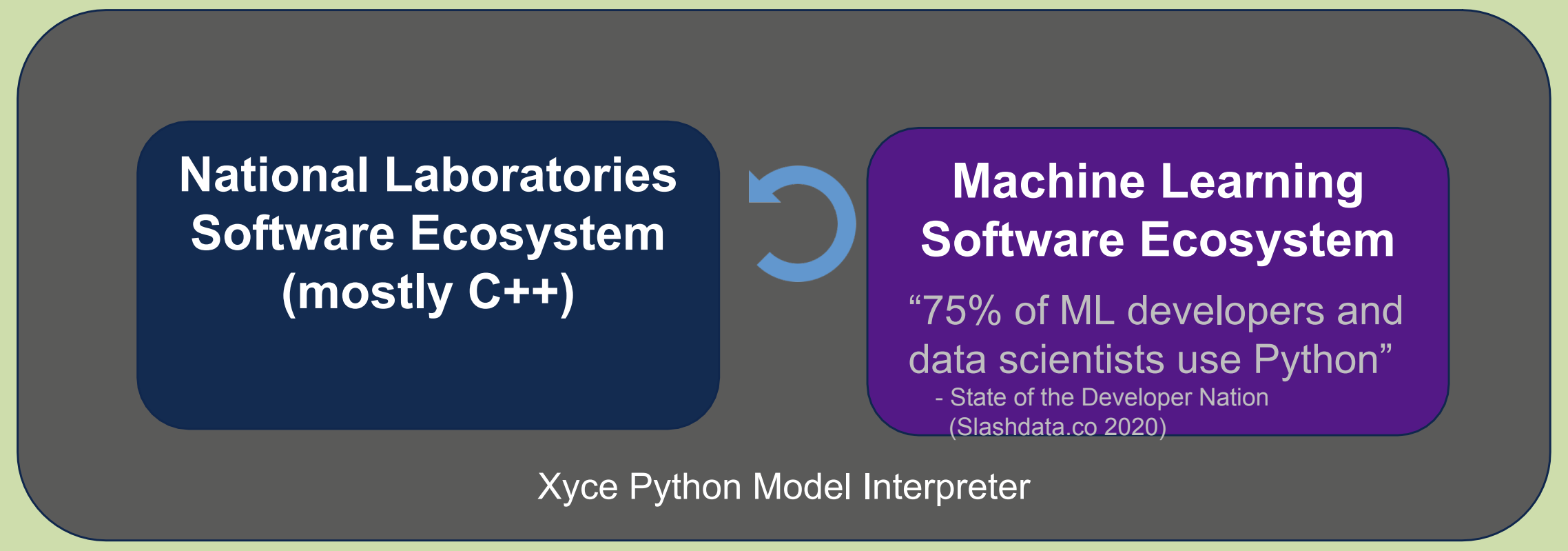
Background

The Xyce Parallel Circuit simulator is a SPICE-style analog circuit simulator, developed from-the-ground-up using a distributed memory paradigm, that provides a modular framework for developing state-of-the-art continuation algorithms, time-integration methods, preconditioned linear solvers, and parallel partitioning techniques

- **Xyce separates device distribution from matrix partitioning**
  - Cost of device evaluation unrelated to matrix partitioning
  - Enables each to separately determine the distribution that is efficient for communication and computation.
  - Hierarchical information leveraged in either distribution to enact efficiencies or integrate "fast-SPICE" techniques.


Proc 1
Load f, q, dF/dx, dQ/dx for n/m devices
Proc 2
Load f, q, dF/dx, dQ/dx for n/m devices
Proc 3
Load f, q, dF/dx, dQ/dx for n/m devices
Proc m
Load f, q, dF/dx, dQ/dx for n/m devices
Device Loads
Global Reorder
Partition
MPI sumAll
Proc 1
Proc 2
Proc 3
Proc m
Matrix Structure

Xyce-PyMi (Paul Kuberry)



National Laboratories Software Ecosystem (mostly C++)

Machine Learning Software Ecosystem

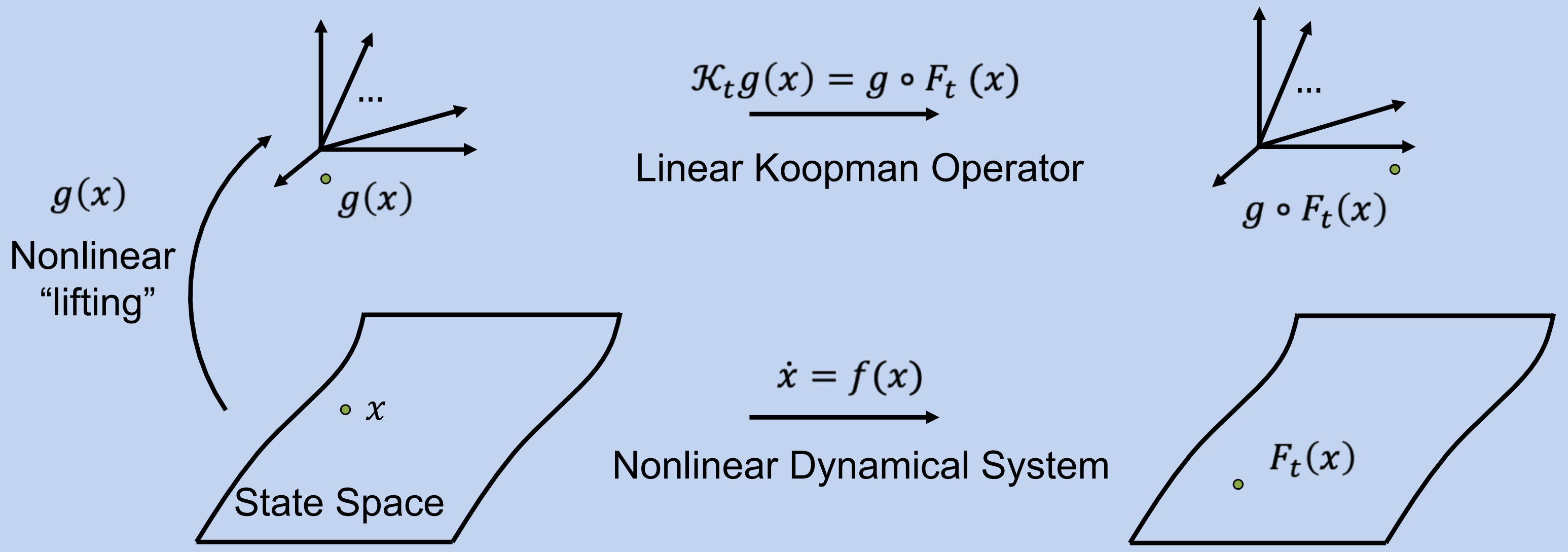
"75% of ML developers and data scientists use Python"
- State of the Developer Nation (Slashdata.co 2020)

Xyce Python Model Interpreter

- Goal is to enable ML advancements to impact circuit design via data-driven models
- Leverages General External interface (C++) from Xyce, by Tom Russo
- Easily installable and callable capability for executing Python ML models in a production circuit simulation software (Xyce <-> PyBind11)
- Actively engaged with compact model development groups and circuit designers at Sandia
- Provide data-driven compact device modeling approaches (GMLS, splines, deep neural networks, **Dynamic Mode Decomposition, see below**) from various ongoing research projects at Sandia

Model Abstraction via Koopman Operator

Koopman Operator: Infinite dimensional *linear* operator that fully describes the underlying evolution of scalar observables of a nonlinear (even chaotic) system. The Koopman operator can be approximated by **lifting** the state space to a higher dimensional space where its evolution is approximately linear.



$g(x)$ Nonlinear "lifting"

$\mathcal{K}_t g(x) = g \circ F_t\ (x)$

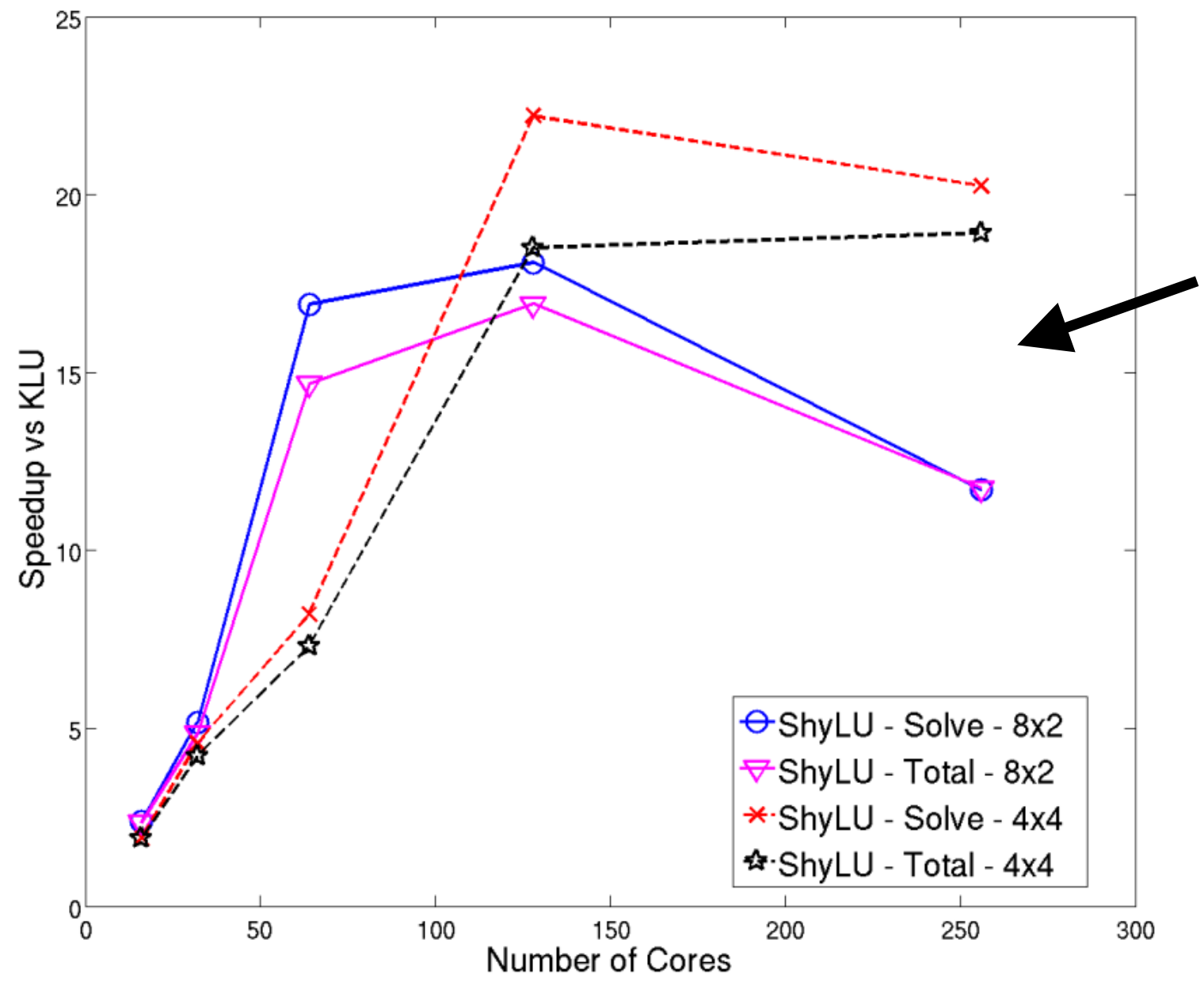
Linear Koopman Operator

$\dot{x} = f(x)$

Nonlinear Dynamical System

Approach

Xyce has succeeded in performing scalable simulation of postlayout ASICs with >1M devices [1] and a history of developing specialized parallel linear solvers [2] to enable efficient simulation on modern architectures.



Xyce ASIC simulation scaling example. The circuit has 1.6M devices and 1.9M unknowns. Study is of strong scaling of Xyce simulation time and linear solve time (ShyLU) for different configurations of MPI Tasks X Threads per node.

[1] "A Hybrid Approach for Parallel Transistor-Level Full Chip Circuit Simulation," Thornquist et al.
DOI: 10.1007/978-3-319-17353-5_9

[2] "A parallel preconditioning strategy for efficient transistor-level circuit simulation," Thornquist, et al.
DOI: 10.1145/1687399.1687477

Xyce Xyce-PyMI to invoke an ML Python device or subcircuit

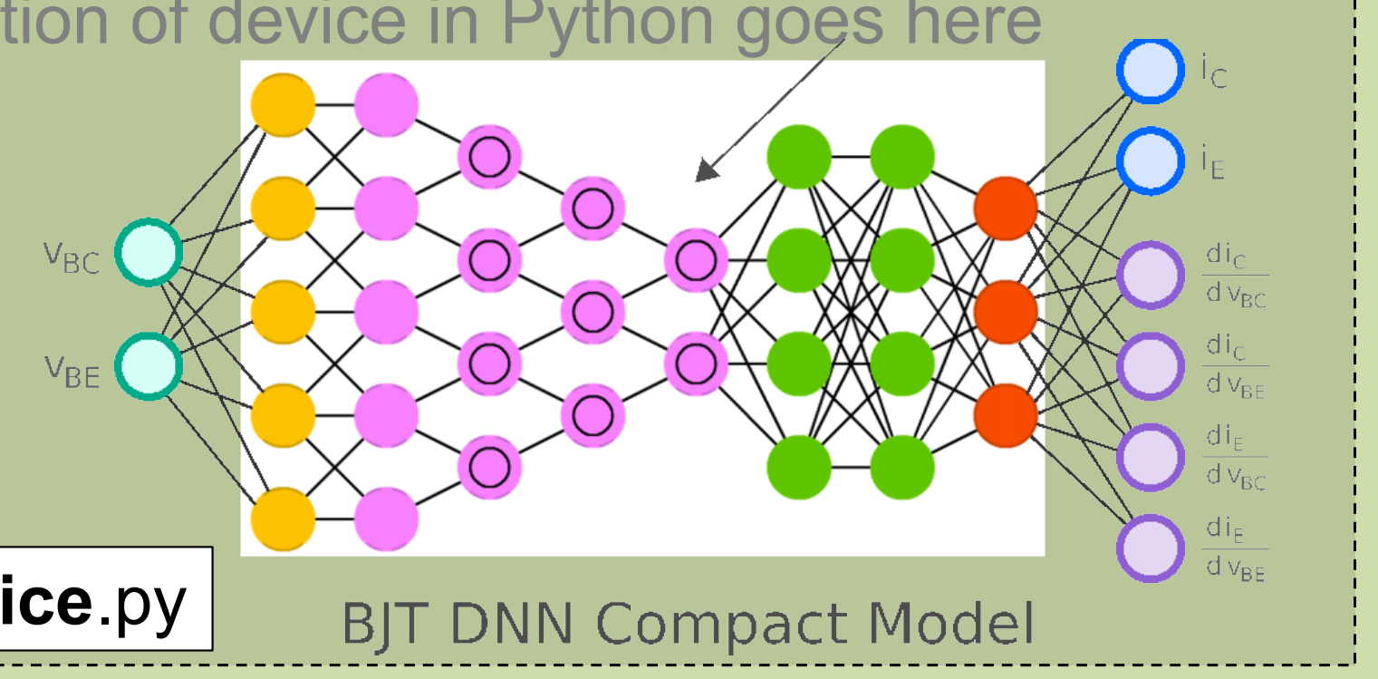
```
>> Xyce-PyMi example.cir
```

Xyce-PyMi (Xyce + **Py**thon **M**odel **I**nterpreter)

- Full Xyce functionality + devices / subcircuits / circuits defined in Python
- Python class defines how F, Q, B, dF/dX, and dQ/dX vectors/matrices are populated for Xyce DAE equation:
  - residual = f(x,t) + dq(x,t)/dt - b(t)
- Supports all popular machine learning frameworks such as TensorFlow, Keras, PyTorch, Jax, Numba, Numpy, etc…

```
* example.cir (Example of easy inclusion in a netlist)
YGENEXT devicename terminal1 terminal2
+ SPARAMS={NAME=MODULENAME
+ VALUE=PythonDevice.py}
```

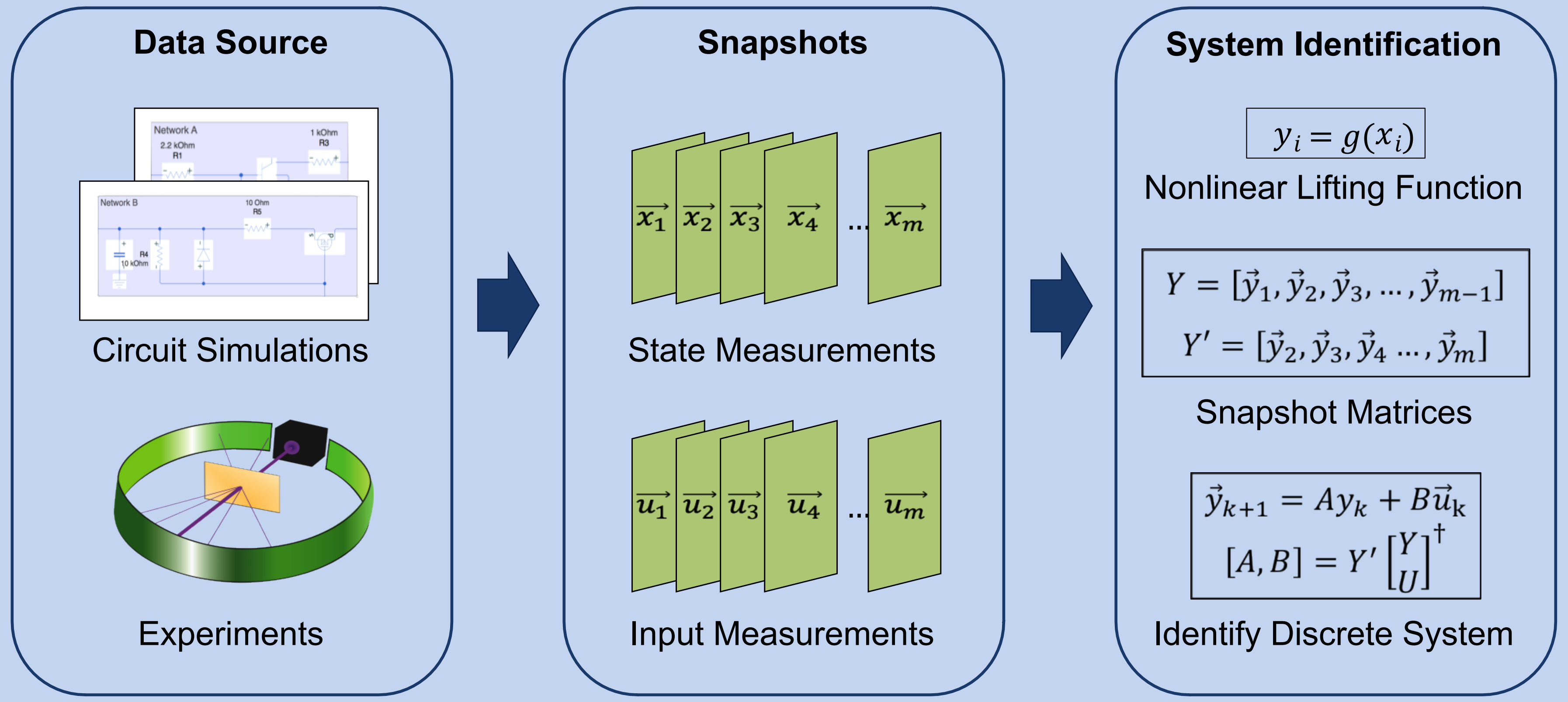
```
from BaseDevice import BaseDevice
class Device(BaseDevice):
   def computeXyceVectors(…):
     # definition of device in Python goes here
```



PythonDevice.py

BJT DNN Compact Model

Extended Dynamic Mode Decomposition (EDMD)



Data Source

Circuit Simulations

Experiments

Snapshots

$\vec{x_1}$ $\vec{x_2}$ $\vec{x_3}$ $\vec{x_4}$ … $\vec{x_m}$

State Measurements

$\vec{u_1}$ $\vec{u_2}$ $\vec{u_3}$ $\vec{u_4}$ … $\vec{u_m}$

Input Measurements

System Identification

$y_i = g(x_i)$

Nonlinear Lifting Function

$$Y = [\vec{y}_1, \vec{y}_2, \vec{y}_3, \ldots, \vec{y}_{m-1}]$$
$$Y' = [\vec{y}_2, \vec{y}_3, \vec{y}_4 \ldots, \vec{y}_m]$$

Snapshot Matrices

$$\vec{y}_{k+1} = Ay_k + B\vec{u}_k$$
$$[A, B] = Y' \begin{bmatrix} Y \\ U \end{bmatrix}^{\dagger}$$

Identify Discrete System

Results and Impact

Obtaining Xyce

https://xyce.sandia.gov/ for Xyce binaries and documentation

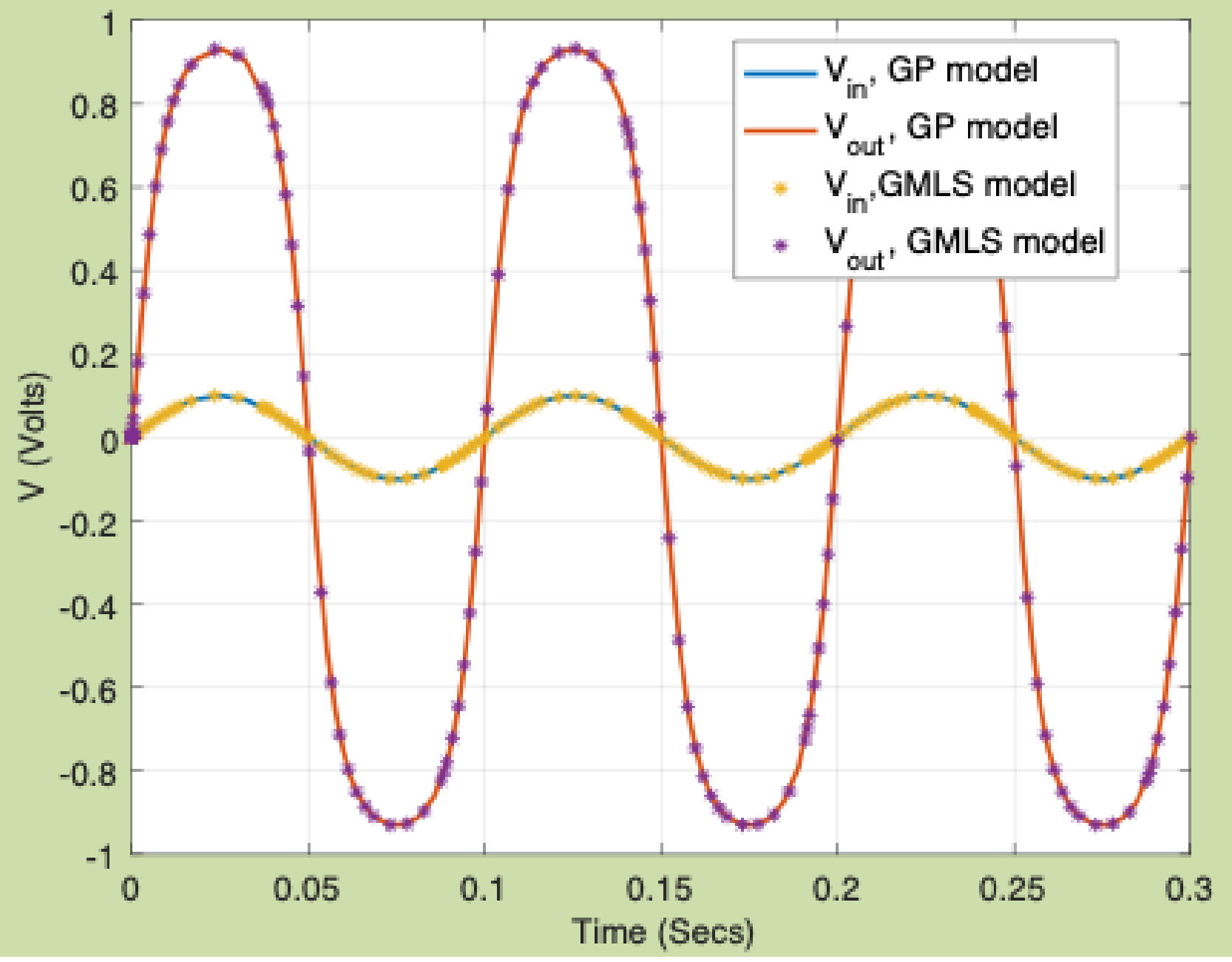
https://github.com/Xyce for Xyce source code and regression tests

Recent developments

- Xyce version 7.3 released in May 2021
- Xyce Data Model (XDM) netlist translator version 2.3
- Support for 65nm 55nm, 14nm 12nm GF PDKs
- Support for Skywater 130 PDK
- Support for polynomial chaos expansion methods for uncertainty quantification
- Support for Verilog Procedural Interface (VPI) to support mixed signal simulation
  - Link to Yale discrete event simulator
- New Python model interface (PyMI). See below!
- Link to Synopsys analog assertions tool, Amstaff

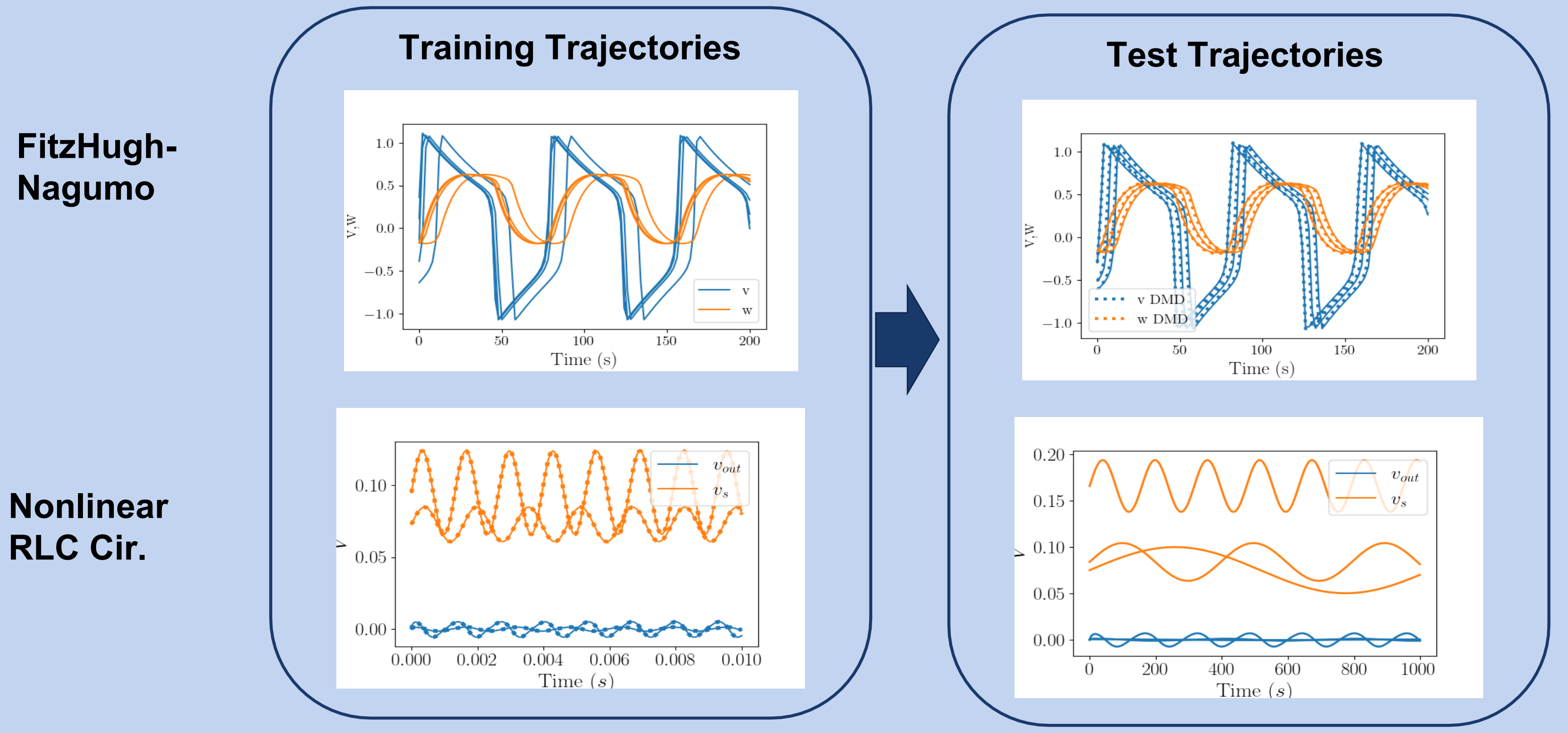
Xyce-PyMI example

Operational Amplifier with BJT models replaced by GMLS* models
GMLS = Generalized Moving Least Squares



* **from:** Development, Demonstration, and Validation of Data-driven Compact Diode Models for Circuit Simulation and Analysis K. Aadithya, P. Kuberry, B. Paskaleva, P. Bochev, K. Leeson, A. Mar, T. Mei, E. Keiter, arXiv:2001.01699

Simulations with EDMD Models



Training Trajectories

Test Trajectories

FitzHugh-Nagumo

Nonlinear RLC Cir.

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA).

The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Distribution Statement A – Approved for Public Release, Distribution Unlimited

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

All images are Author's own