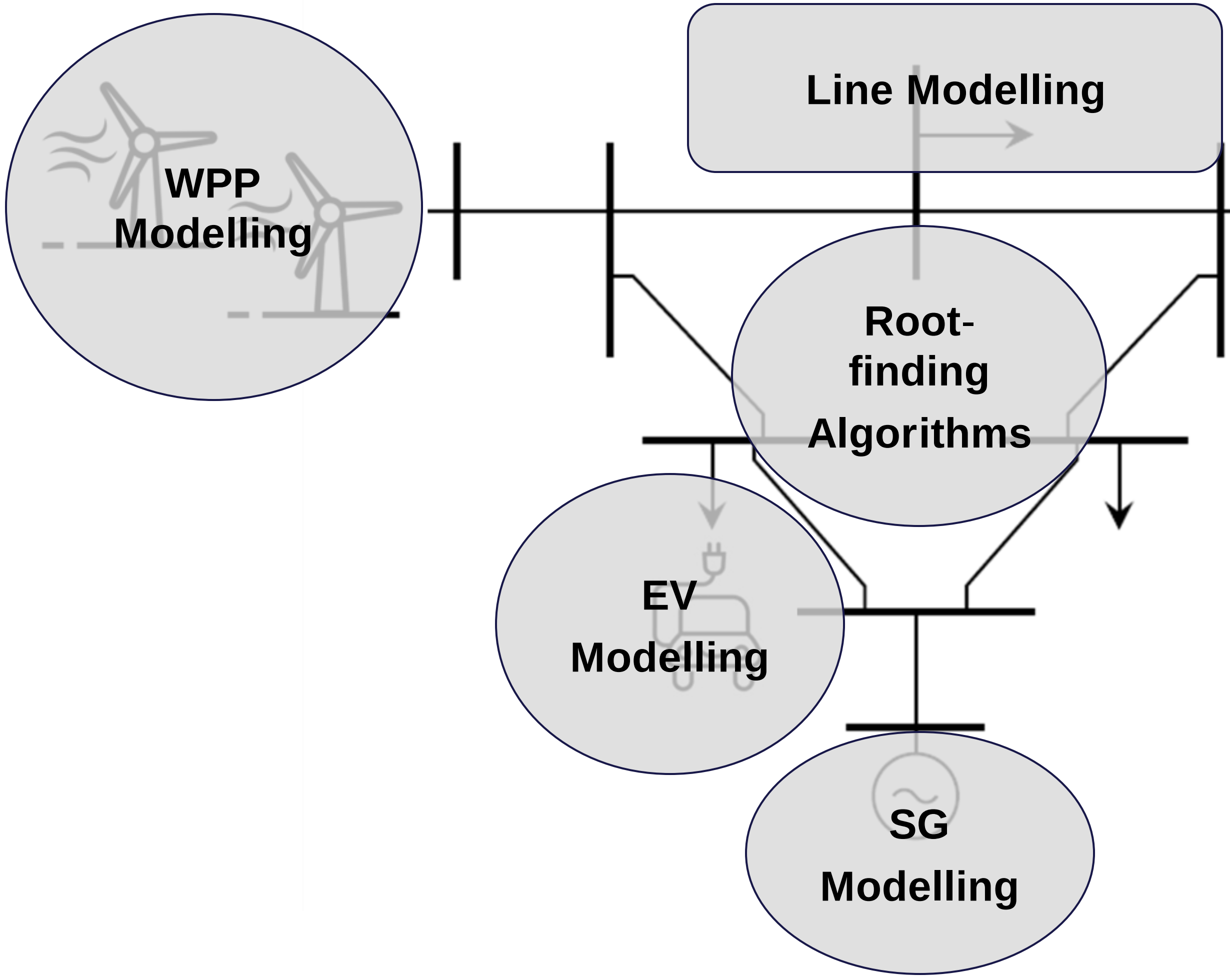
Rahul Nellikkath
Postdoc, Wind and Energy Systems,
Denmark Technical University

Physics-Informed Artificial Intelligence Simulator (PAISim) for Power System Applications

What is PAISim?

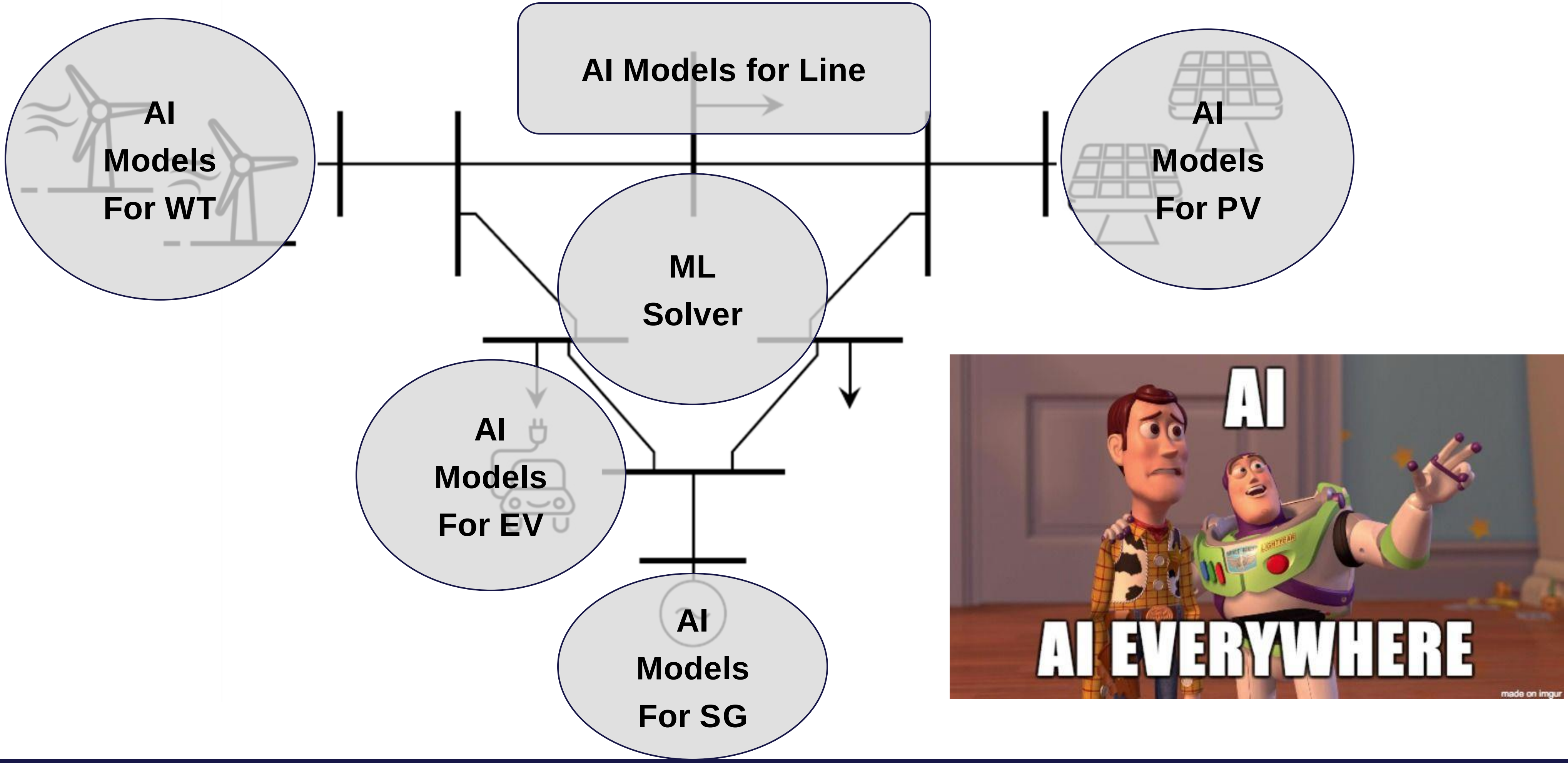


How Traditional Simulation Tools Operate

PSCAD

SIMULINK

What is PAISim?



Why Do We Need Another Simulator?

PSCAD

DIgSILENT MIDDLE EAST

DIgSILENT POWER SYSTEM SOLUTIONS

PSS®E

SIMULINK

MATPOWER

GridLAB-D

Why?

Why?

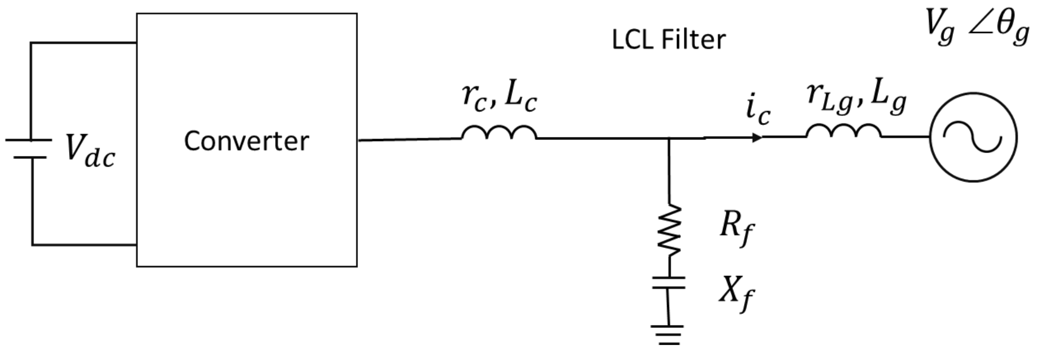
Motivation for a New Power System Simulation Tool

- Power generation is moving towards more greener renewables

>10 GW wind power plants

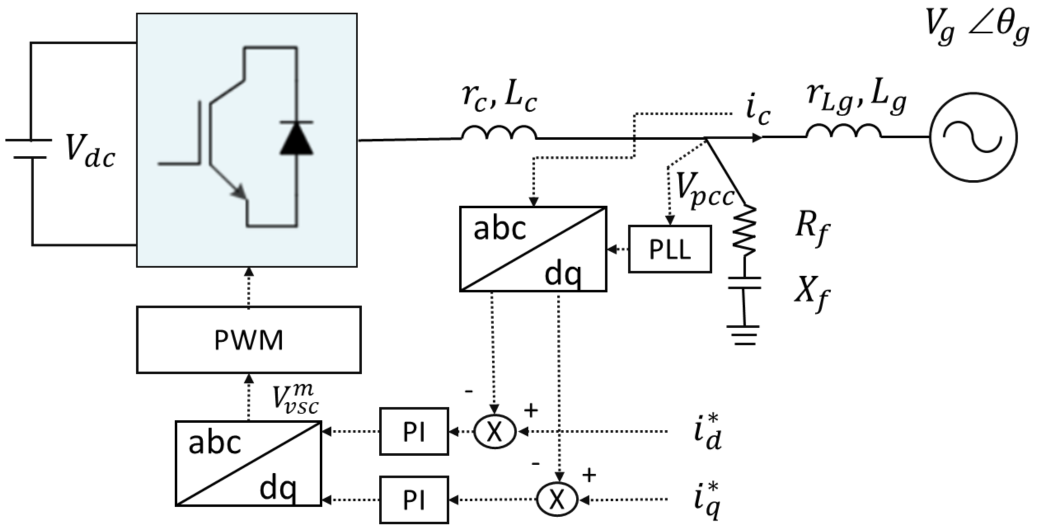
Motivation for a New Power System Simulation Tool

- Power generation is moving towards more greener renewables
- Grid-tied renewable ➡ Nonlinear Inverter based Controllers



Motivation for a New Power System Simulation Tool

- Power generation is moving towards more greener renewables
- Grid-tied renewable ➡ Nonlinear Controllers like Phase Locked Loop (PLL)



Motivation for a New Power System Simulation Tool

- PLL converter controler → Instantaneous grid voltage phase angle and frequency
- Grid faults → PLL unsynchronized → Grid instability
- Electromagnetic Transients (EMT) Simulations for analysing Grid Instabilities
  - **Time consuming** even for a power plant
  - Often only done for **a few** predefined scenarios

Objective: AI for Power System Stability

- Develop ML models to predict power system stability with both conventional and inverter-based resources

- Why Use Machine Learning?

  – ML models can be **100 to 1,000 times faster** than conventional ones with **good accuracy**

  – ML models can quickly screen many scenarios, focusing on critical ones for EMT simulations

Making AI Work for Power System Stability

Requirements:

1. Capture Power System Component Dynamics:
    - ➢ Conventional power system components
    - ➢ Inverter-based resources and grid components

2. Capture component interaction
    - ➢ Overall power grid behaviour

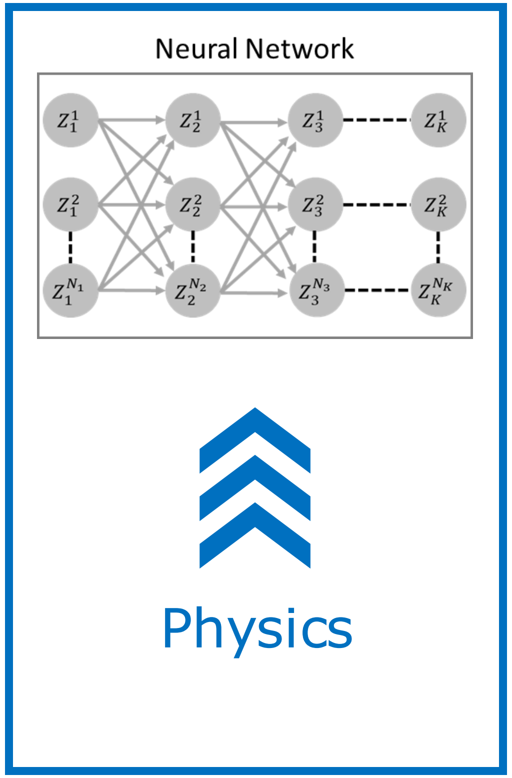
Limitations of Neural Network

Physics Informed Machine Learning

- **Traditional NN Training:** Relies heavily on large datasets with labeled data

- **Challenges:**
  - – Is there sufficient data available?
  - – Does the data encompass all relevant scenarios?
  - – How long will it take to create a robust dataset?

- Learn from the physics ⇨ Physics-Informed Machine Learning

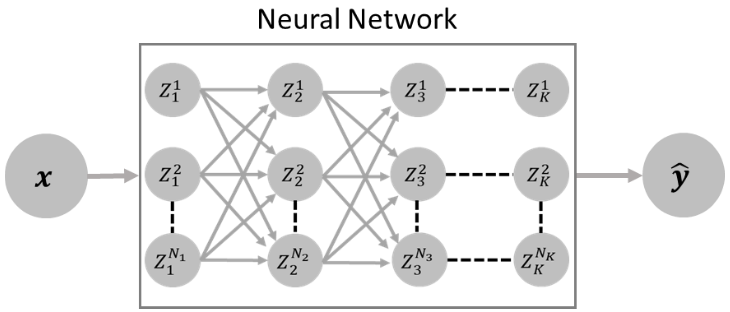


Physics Informed Machine Learning

Less Data, More Physics: PINN for Power System

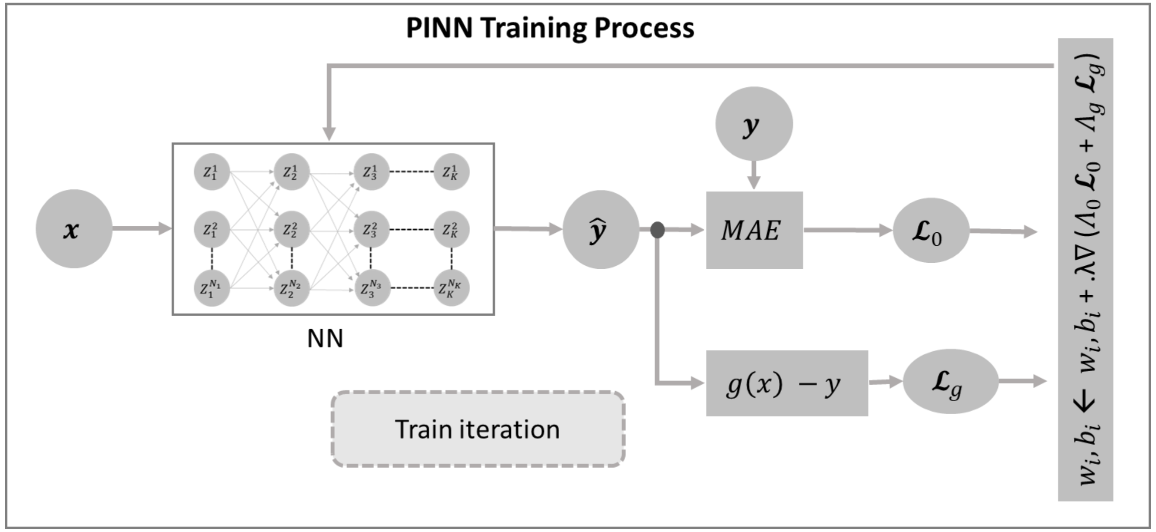
- Assume:

$$y = g(x),$$



- Training:

  generate data set $y_i = g(x_i)$,



PINN for Power System Applications

Modelling Power System Dynamics

$$\frac{d}{dt}\boldsymbol{x} = \boldsymbol{f}\left(t, \boldsymbol{x}; \boldsymbol{u}, \boldsymbol{\lambda}\right)$$

Trajectory: Temporal state evolution

$$\boldsymbol{x}(t) = \boldsymbol{x}_0 + \int_{t_0}^{t} \boldsymbol{f}\left(\tau, \boldsymbol{x}; \boldsymbol{u}, \boldsymbol{\lambda}\right) d\tau$$

Example: Single Machine Generator

$$\underline{\frac{d}{dt}}\underline{\begin{bmatrix} \delta \\ \Delta\omega \end{bmatrix}} = \begin{bmatrix} \omega_0 \Delta\omega \\ \frac{1}{2H}\left(\underline{P} - D\Delta\omega - \frac{E_q' V}{X_d}\sin\left(\delta - \theta\right)\right) \end{bmatrix}$$

- Time t
- State $\boldsymbol{x}$
- Control input $\boldsymbol{u}$
- Parameters $\boldsymbol{\lambda}$
- Nonlinear system with 2 states

Physics-Informed Neural Networks for Power Systems

Loss function from data

PINN Loss function

$$\min_{w,b} \boxed{\frac{1}{|N_\delta|}\sum_{i\,\epsilon\,N_\delta} |\hat{\delta} - \delta^i|^2} \boxed{+\frac{1}{|N_c|}\sum_{i\,\epsilon\,N_c} |f(\hat{\delta})|^2}$$

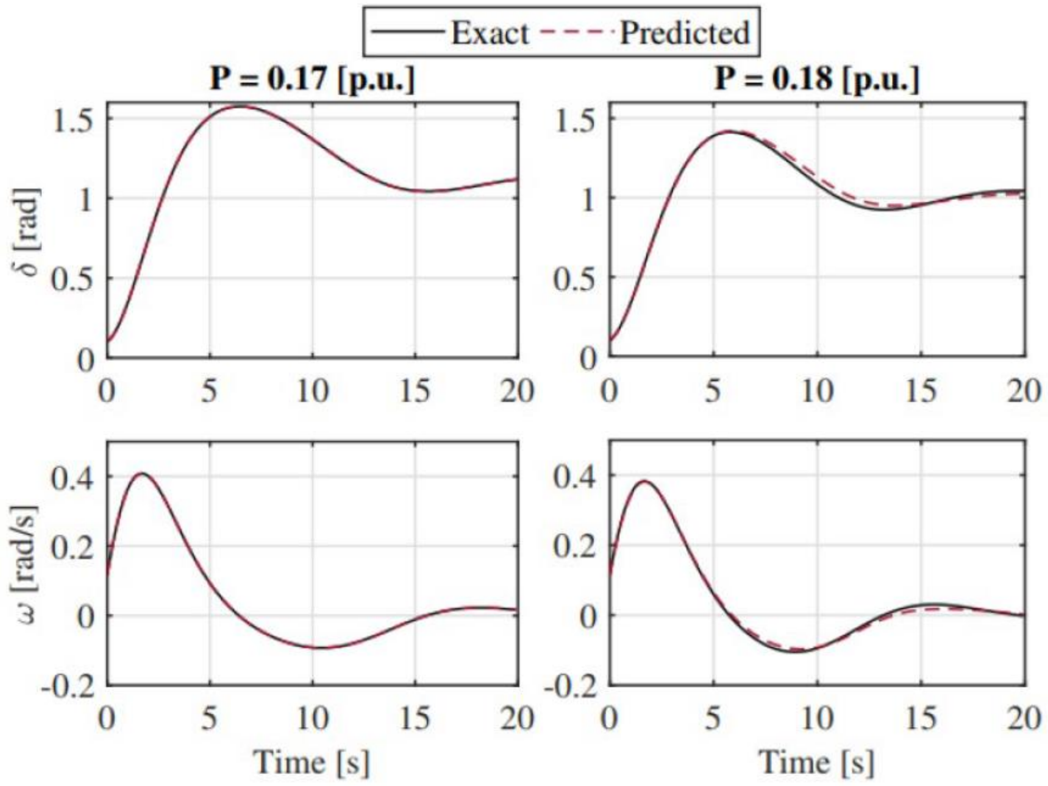
Neural Network Auto Differential

$$\hat{\delta} = NN(x_o, t, u)$$

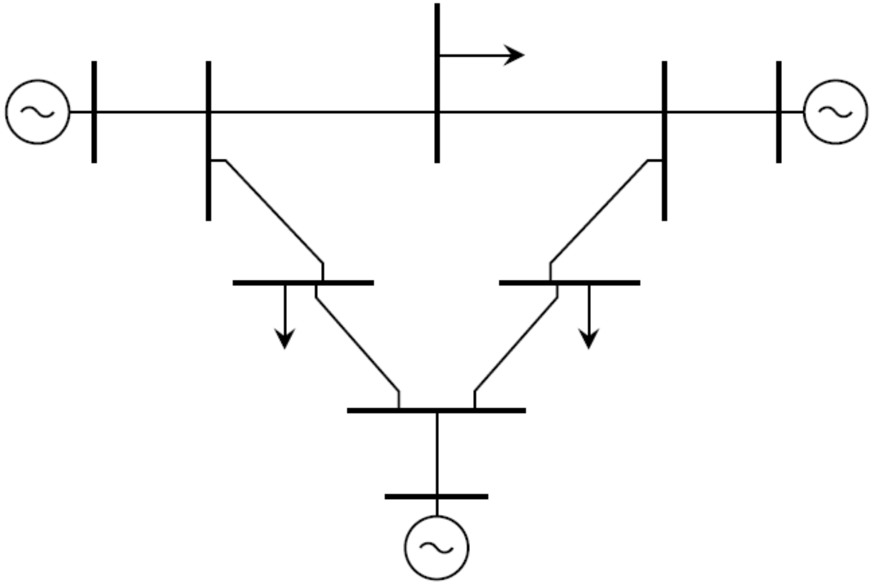
$$\dot{\hat{\delta}} = \frac{\partial \hat{\delta}}{\partial t},\ \ddot{\hat{\delta}} = \frac{\partial \dot{\hat{\delta}}}{\partial t}$$

Swing equation

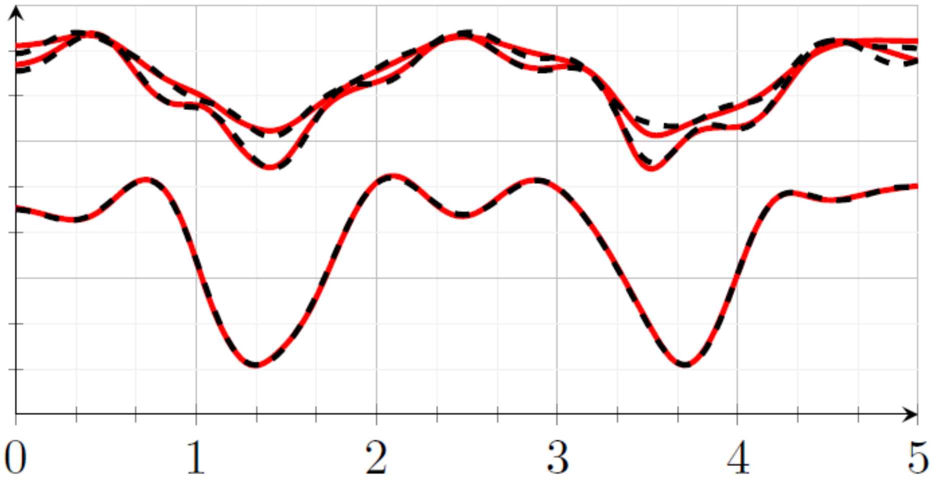
$$f(\hat{\delta}) = \mathrm{M}\ddot{\hat{\delta}} + D\dot{\hat{\delta}} + \mathrm{A}\sin\hat{\delta} - u$$



PINNs can Predict Trajectories of Multi-machine System



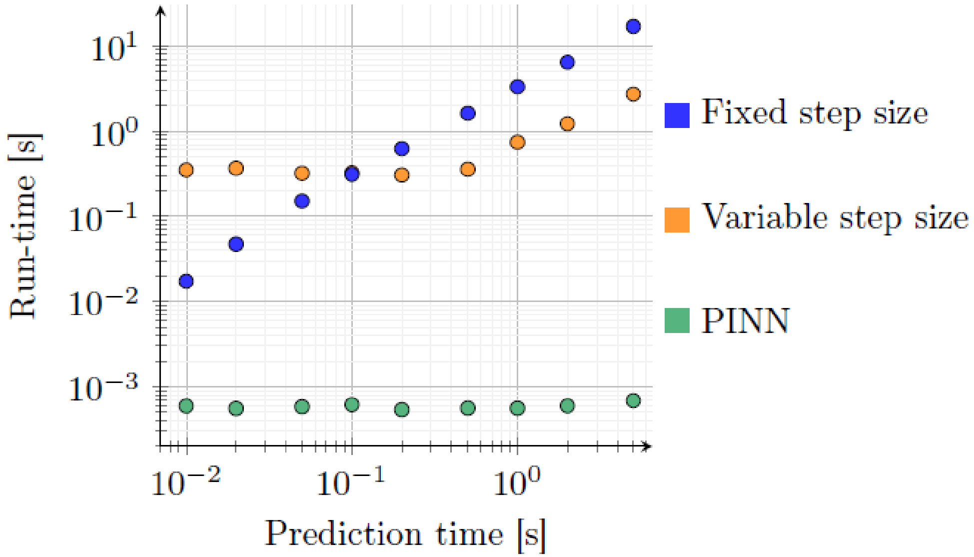
Trajectories of 24 variables (3 shown)



- Interacting dynamical systems
- Coupled by power exchange

Why PINNs !

- **PINNs Shift Learning Paradigm:** Transition from supervised to nearly unsupervised learning
- **Potential Impact:** PINNs could eventually replace differential-algebraic equation solvers
- **Power System Application:** Ultra-fast screening of critical contingencies
- **Capability:** Direct estimation of rotor angle at any time instant



Making AI Work for Power System Stability

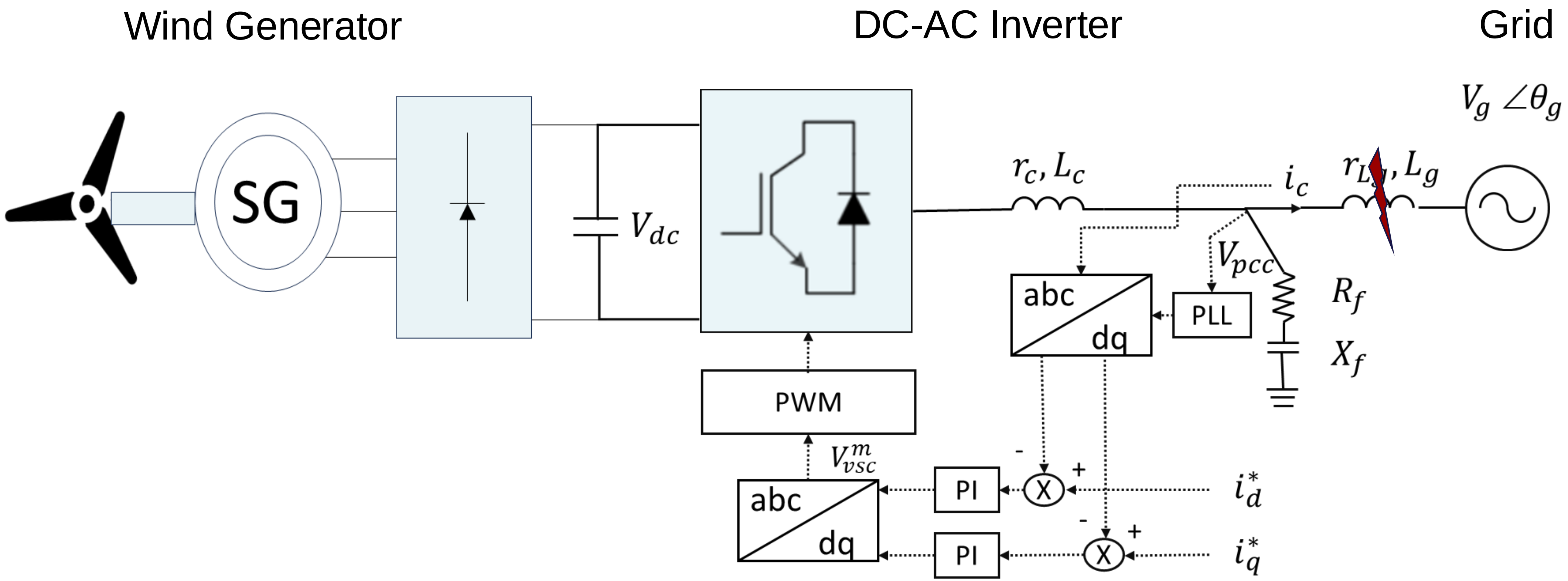
Requirements:

1. Capture Power System Component Dynamics:
   - ➢ Conventional power system components
   - ➢ Inverter-based resources and grid components

2. Capture component interaction
   - ➢ Overall power grid behavior

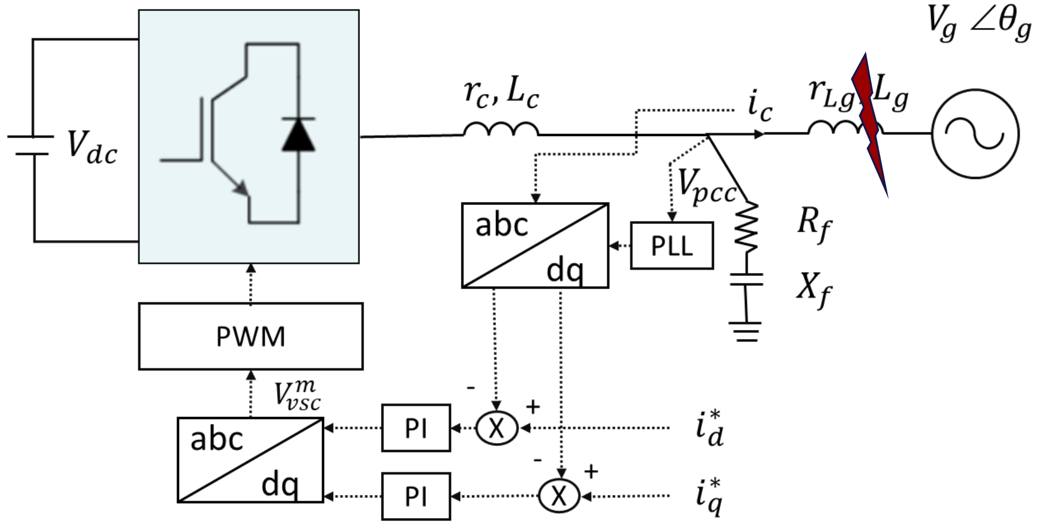
PINN for Inverter based Generators

Inverter based Generators

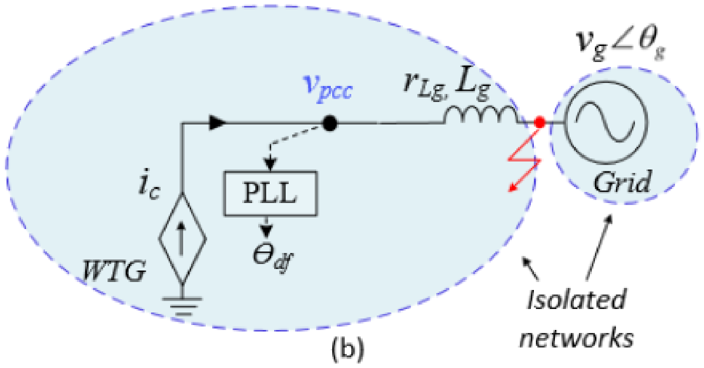


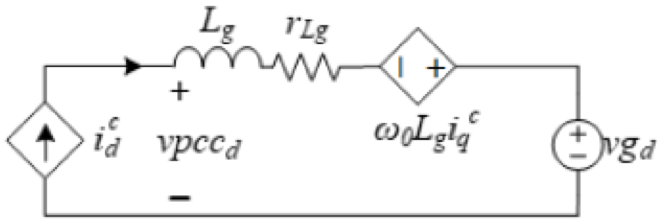
Reduced order model for Transient stability assessment of Inverter based Generators with PLL

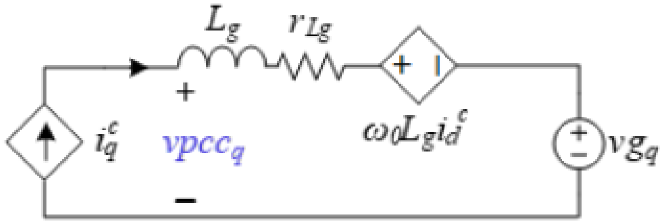
- Grid fault ➡ DC chopper is activated ➡ DC voltage is assume to be study



Reduced Order Model for Transient Stability Assessment of Inverter based Generators with PLL







$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{1}{M}(T'_m - T'_e - D' x_2) \end{bmatrix}$$

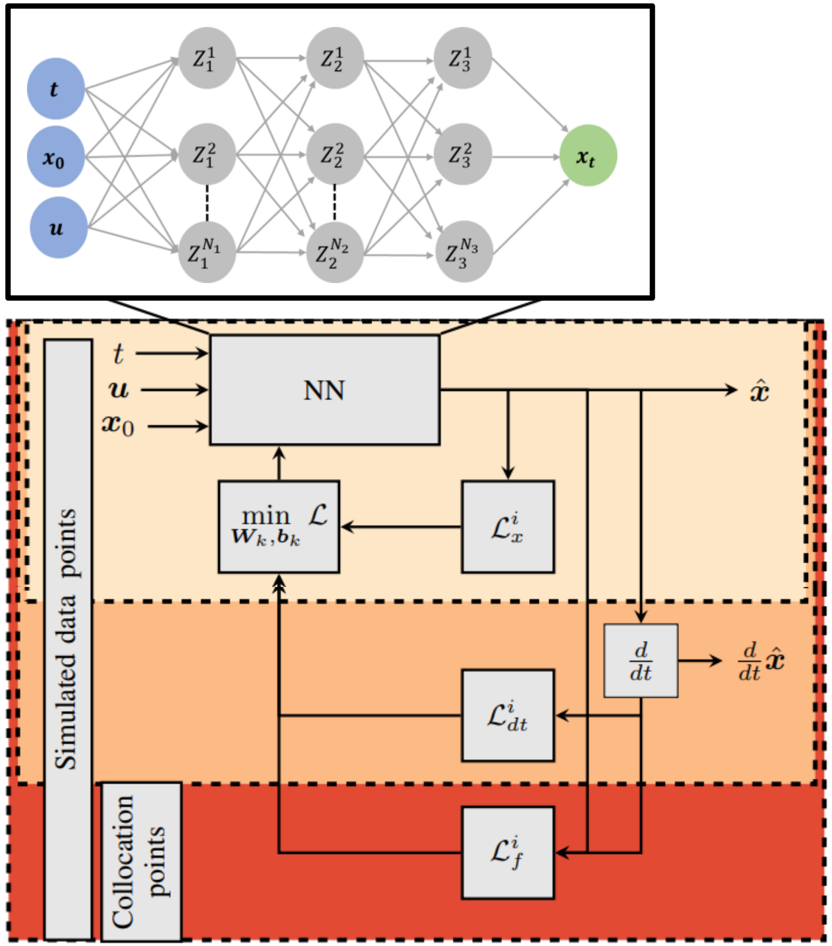
where, $x_1 = \delta$ and $x_2 = \dot{\delta}$, and

$$\frac{\partial x}{\partial t} = f(x, u)$$

$$\delta = (\theta_{df} - \theta_g)$$

Dowlatabadi, M. K. B., Ghosh, S., Kocewiak, L. ,& Yang, G. (2021). Transient stability assessment of Type-4 Wind Turbines based on an Improved reduced order model.

Proposed Physics-Informed Neural Network



$$\boxed{\min_{w,b} \frac{1}{|N_t|} \sum_{i\,\epsilon\,N_t} \mathcal{L}_x^i + \mathcal{L}_{dt}^i + \mathcal{L}_f^i}$$

$\hat{x}_t = x_0 + t{*}NN(t, x_0, u)$

ODE Solvers

$f(x^i)$

$\mathcal{L}_x^i = |\hat{x} - x^i|^2$

Some Physics

$\mathcal{L}_{dt}^i = |\dot{\hat{x}} - \dot{x}^i|^2,$ $\quad \dot{\hat{x}} = \frac{\partial \hat{x}}{\partial t}$

PINN

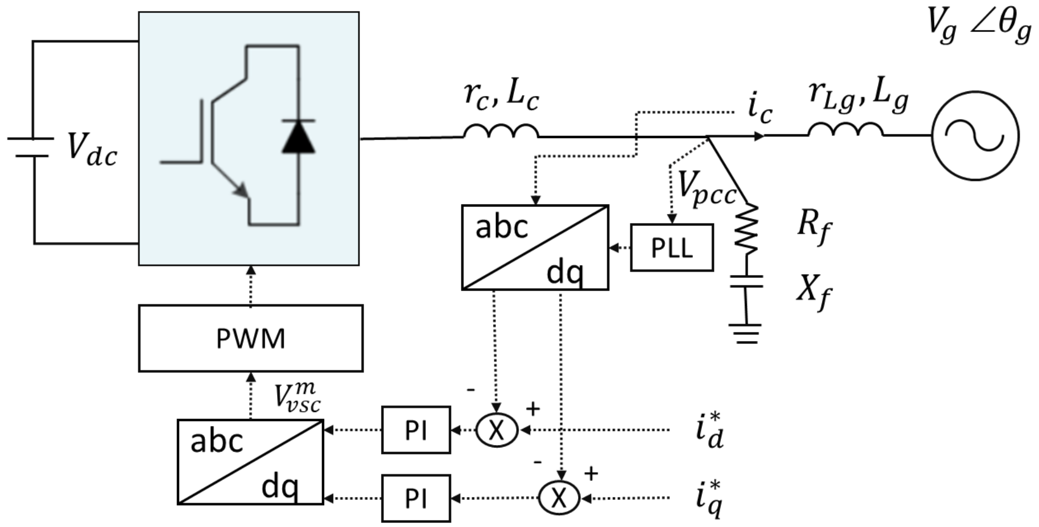
$\mathcal{L}_f^i = |\dot{\hat{x}} - f(\hat{x})|^2$

More Physics

Less Data

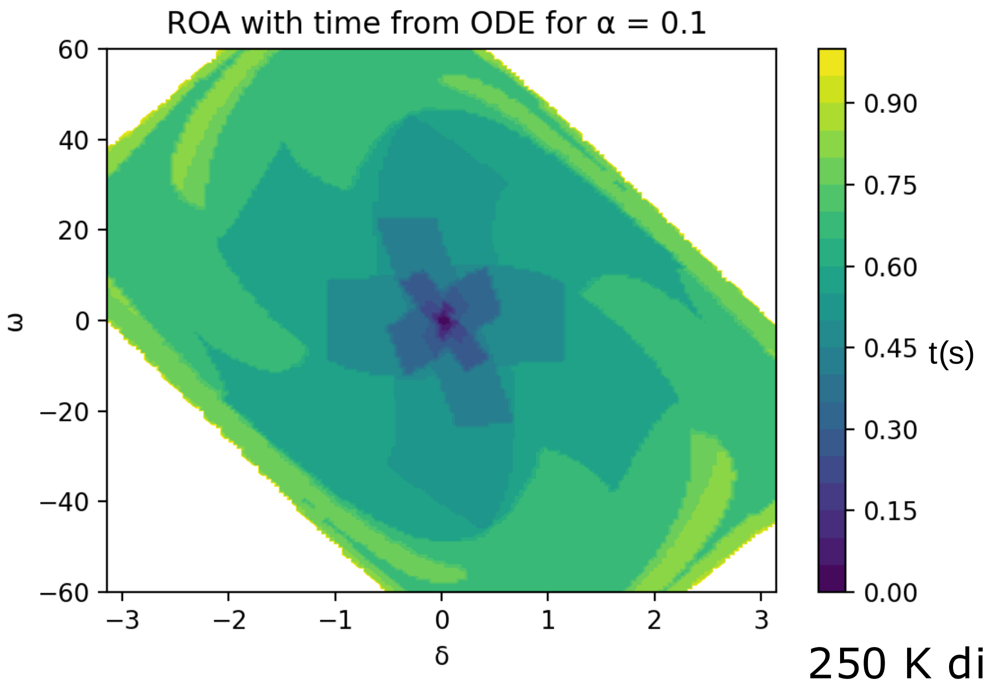
Results: Predicting Region of Attraction

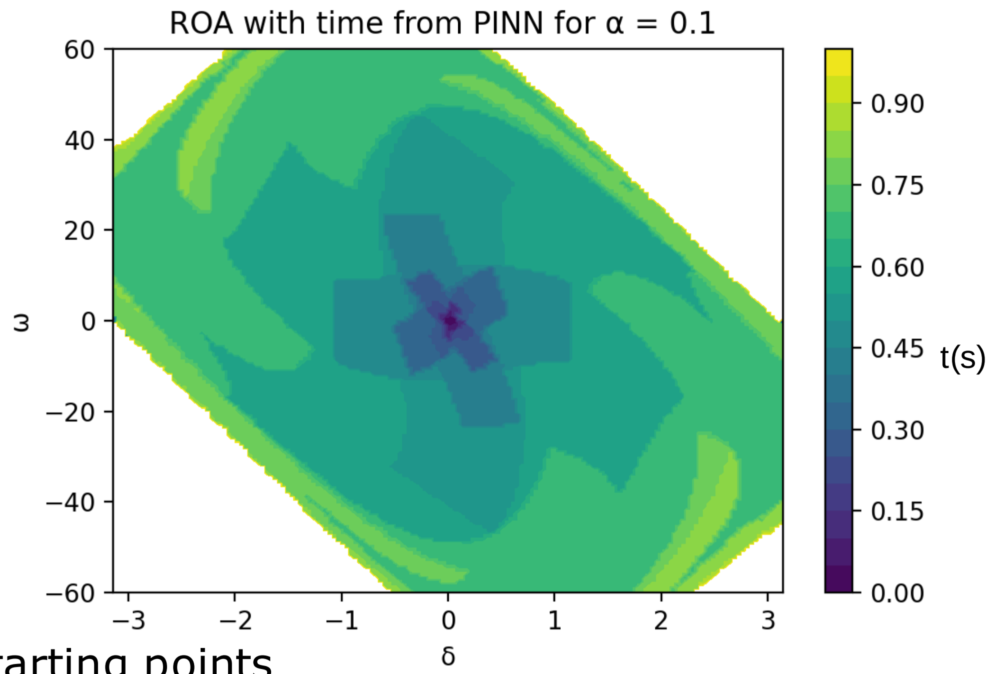
- ROA ➡ System states from which all the trajectories converge to a stable equilibrium point
- ROA for different grid L_g and r_{L_g}. Assuming L_g/r_{L_g} is fixed
- Both L_g and r_{L_g} are increase by a factor of α
- PINN is trained with α as a parameter



Results: Predicting Region of Attraction

- ROA ➡ System states from which all the trajectories converge to a stable equilibrium point
- ROA for different grid L_g and r_{L_g}. Assuming $^{L_g}/_{r_{L_g}}$ is fixed
- Both L_g and r_{L_g} are increase by a factor of α
- PINN is trained with α as a parameter

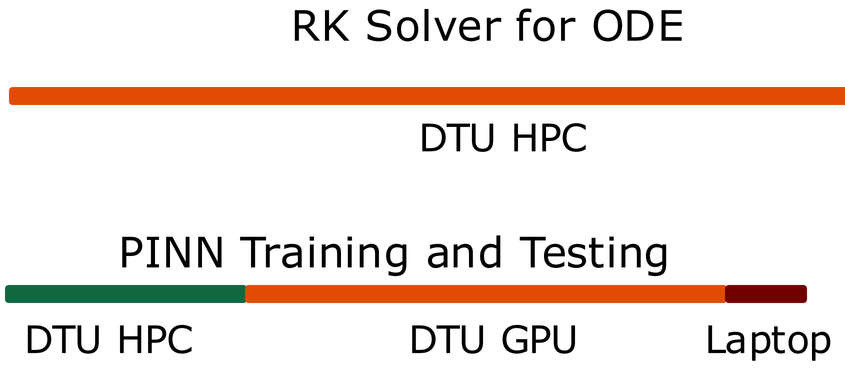




250 K different starting points

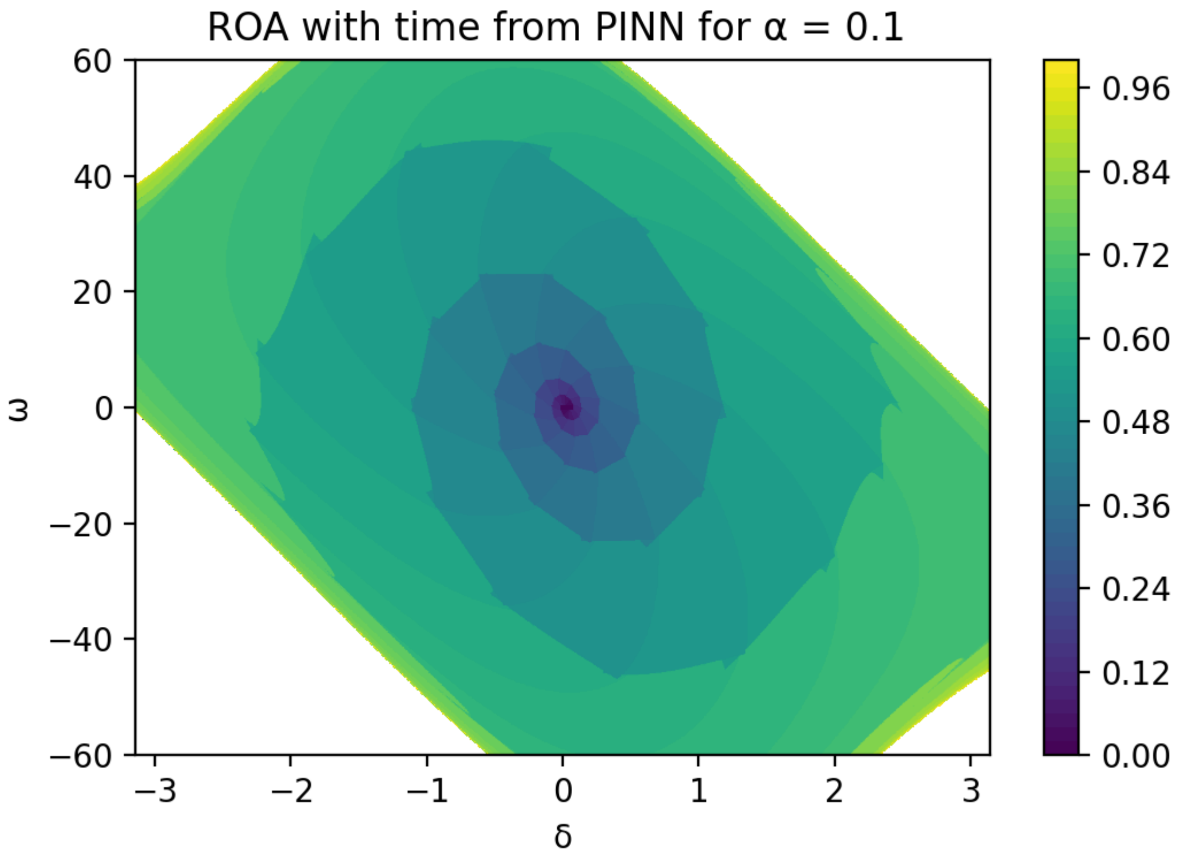
Time taken – 250 K Starting points

- **ODE** - 250 K starting points- 2 hrs 15 min in DTU HPC using RK solver python



- **PINN**
  - **Data collection** – only for the first 100 ms (~100 K starting points) < 30 min
  - **Training** – 10 min using DTU GPU
    - 30 to 60 min for testing different hyperparameters
  - **Predicting** for 250 K starting points ➡ predict δ and ω after fault is cleared for 1 sec ➡ <10 min
  - **Total** < 2 hrs

Region of Attraction with 5 M points



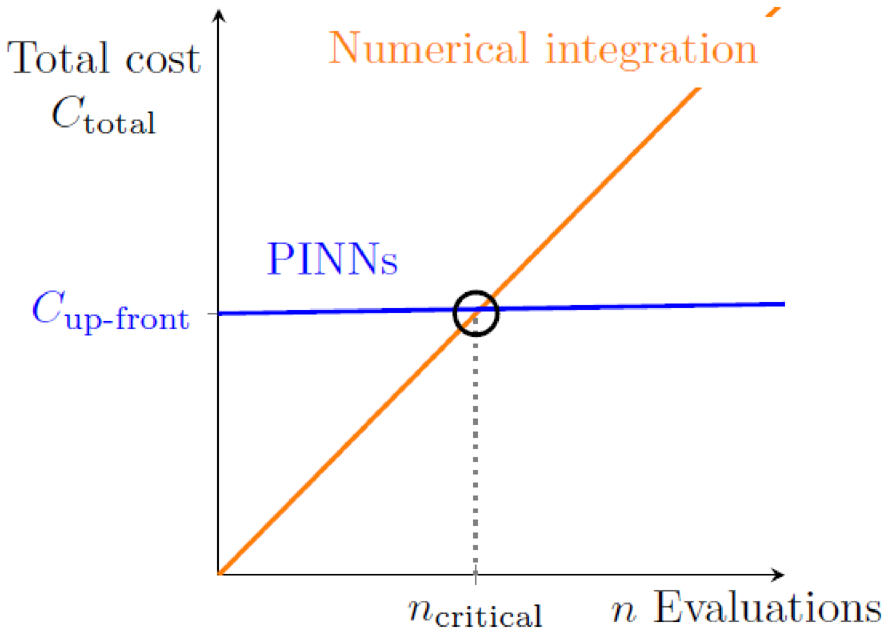
5 Million Starting points

PINN < 30 min

ODE Solution not available

Would take DTU HPC **> 2 days**

So..



PINN

→Access numerous scenarios

→identify critical case

→access using EMT simulation

Making AI Work for Power System Stability

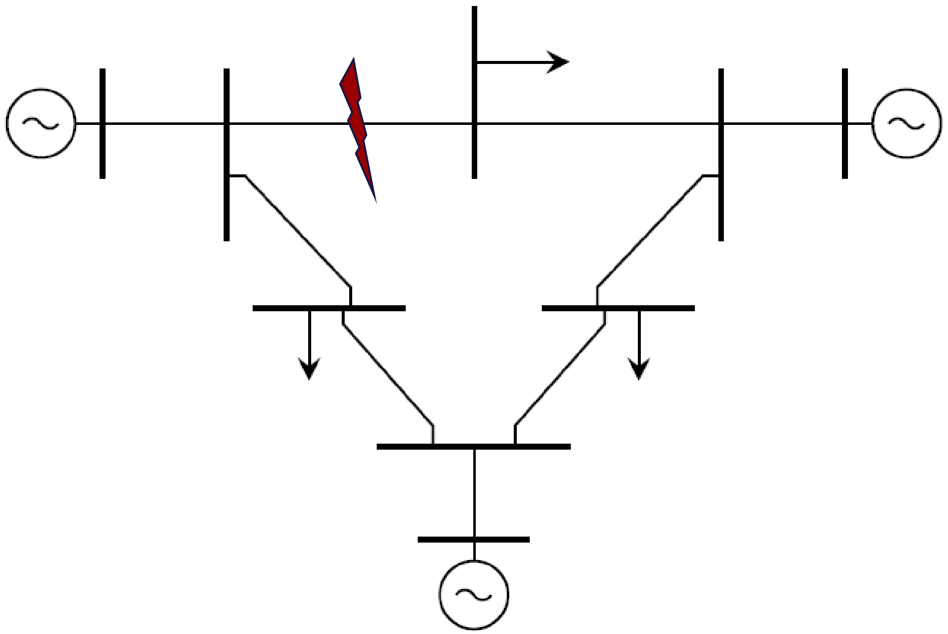
Requirements:

1. <u>Capture Power System Component Dynamics:</u>
   - ➢ Conventional power system components
   - ➢ Inverter-based resources and grid components

2. <u>Capture component interaction</u>
   - ➢ Overall power grid behaviour

Curse of dimensionality

Challenges of PINN in Power System



What happens when the setup changes?

Retrain NN every time

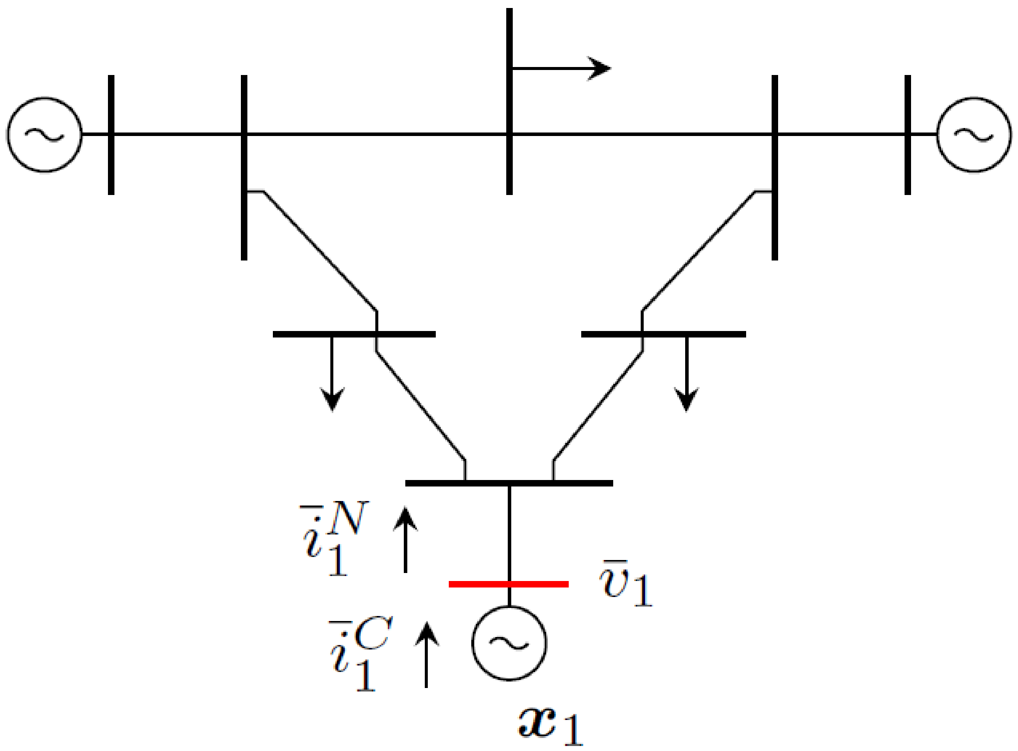
- Increases the training time
- Difficult to reach critical time

Incorporate all the variations

- Dimensionality of the learning problem increases
- More difficult and expansive training

PAISim – Connecting PINNs or AI models to Grid

Dynamical Systems Coupled by Power Transfer



Components inject currents $\bar{\iota}_i^C\ (\boldsymbol{x}_i, \bar{v}_i)$

- Component dynamics $f_i(t, \boldsymbol{x}_i, \bar{v}_i)$
- Depend on local voltage $\bar{v}_i$

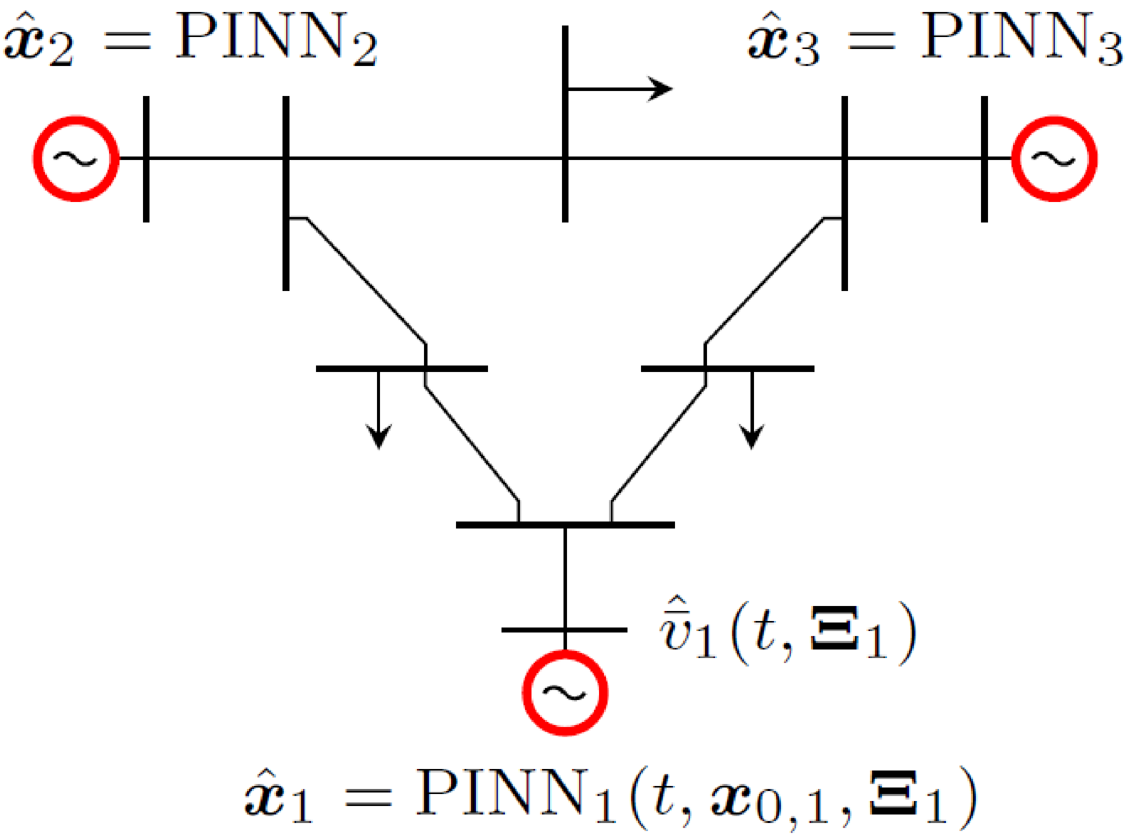
Network currents $\bar{\iota}_i^N(\bar{v}_i)$

- Depend on system structure

Need to satisfy current balance $\bar{\iota}_i^C = \bar{\iota}_i^N$

Functions of time: $\boldsymbol{x}_i(t), \bar{v}_i(t), \bar{\iota}_i^C(t), \bar{\iota}_i^N(t)$

PAISim Concept



Assume voltage evolution

- Parametrise with Ξ_i
- One evolution per bus $\hat{\bar{v}}_i(t, \Xi_i)$

Solve component dynamics with PINNs

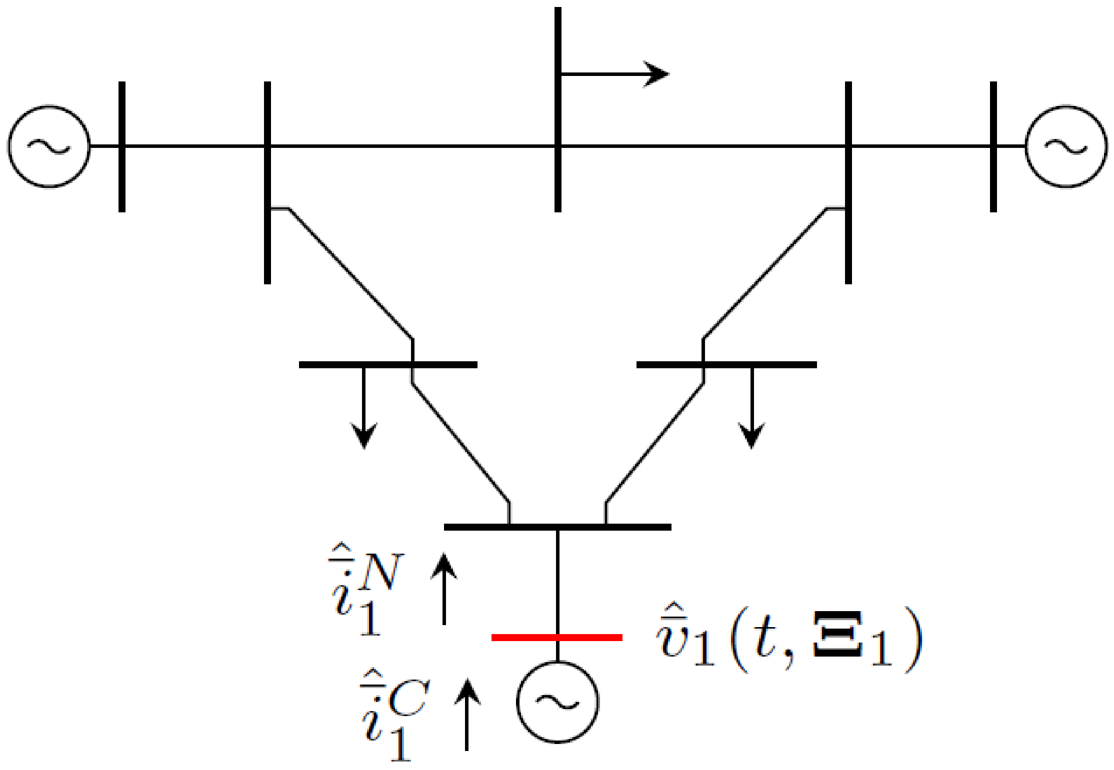
- $\boldsymbol{x}_i = PINN_i(t, x_0, \Xi_i)$
- One $PINN_i$ per component

Approximate current injections $\widehat{\bar{\imath}_i^C}$, $\widehat{\bar{\imath}_i^N}$

- Components $\widehat{\bar{\imath}_i^C}(t, x_0, \Xi_i)$
- Network $\widehat{\bar{\imath}_i^N}(t, \Xi_i)$

Task: Match current balance $\widehat{\bar{\imath}_i^C} = \widehat{\bar{\imath}_i^N}$

Find parameters that minimize error in current balance

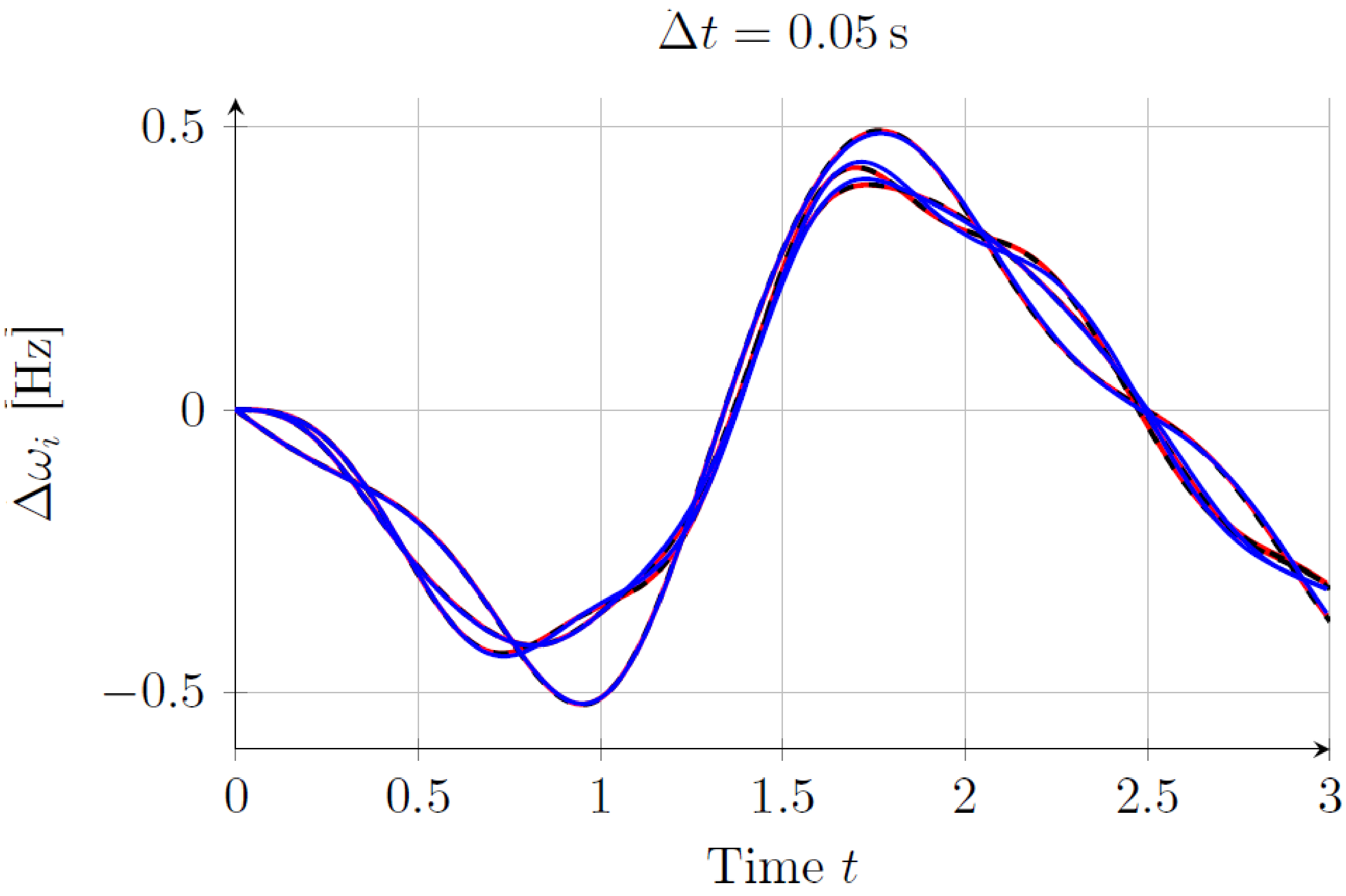


Provide initial guess for Ξ

Loop until convergence:

1. Evaluate $\hat{\imath}^C(\hat{x}, \Xi)$, $\frac{\partial}{\partial \Xi}\hat{\imath}^C$
2. Evaluate $\hat{\imath}^N(\Xi)$, $\frac{\partial}{\partial \Xi}\hat{\imath}^N$
3. Compute current balance error $\hat{\imath}^C - \hat{\imath}^N$
4. Reduce error by adjusting Ξ

PAISim allows for larger time steps



The error for larger time steps remains small

Less time steps needed -> potential acceleration

Making AI Work for Power System Stability

Requirements:

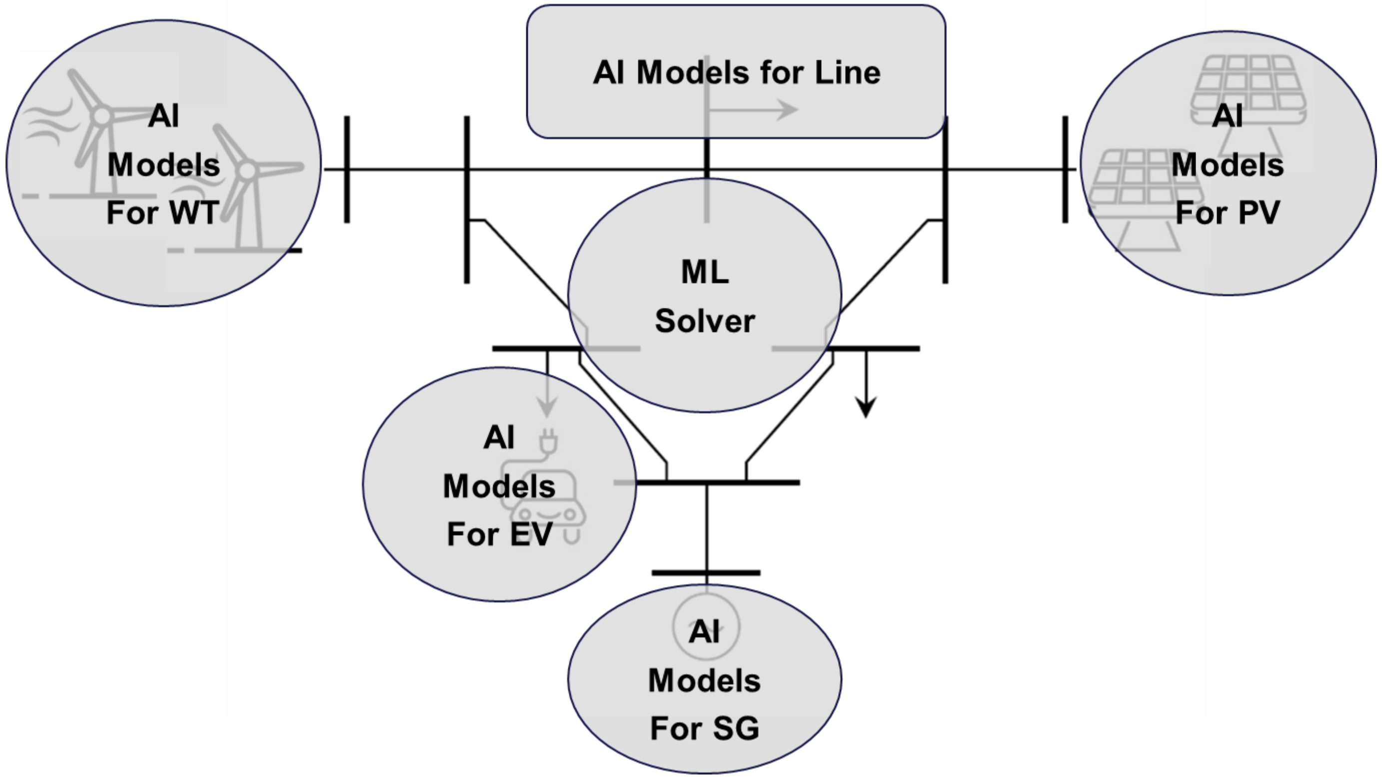
1. Capture Power System Component Dynamics:
   - Conventional power system components
   - Inverter-based resources and grid components

2. Capture component interaction
   - Overall power grid behaviour

End Goal

PAISim – Different AI Modules

- **AI/ODE Modules**: Pre-built libraries for various power system components
- **GPU-Accelerated Solver**: Simultaneously handles multiple N-1 scenarios with fast root-finding algorithms



This work would not have be possible without the hard work of several people! Many Thanks to..

Jochen
Stiasny

Rahul
Nellikkath

Andreas
Venzke

Spyros
Chatzivasileiadis

Guangya
Yang

Ilgiz
Murzakhanov

Sujay
Ghosh

Sam
Chevalier

Baosen

Zhang

Mohammad
Kazem

Thank You

PINNSim
Code

PINN for PLL
Paper

Rahul Nellikkath

Physics-Informed Artificial Intelligence Simulator (PAISim) for Power System Applications