# Enhanced physics-informed neural networks (PINNs) for high-order power grid dynamics

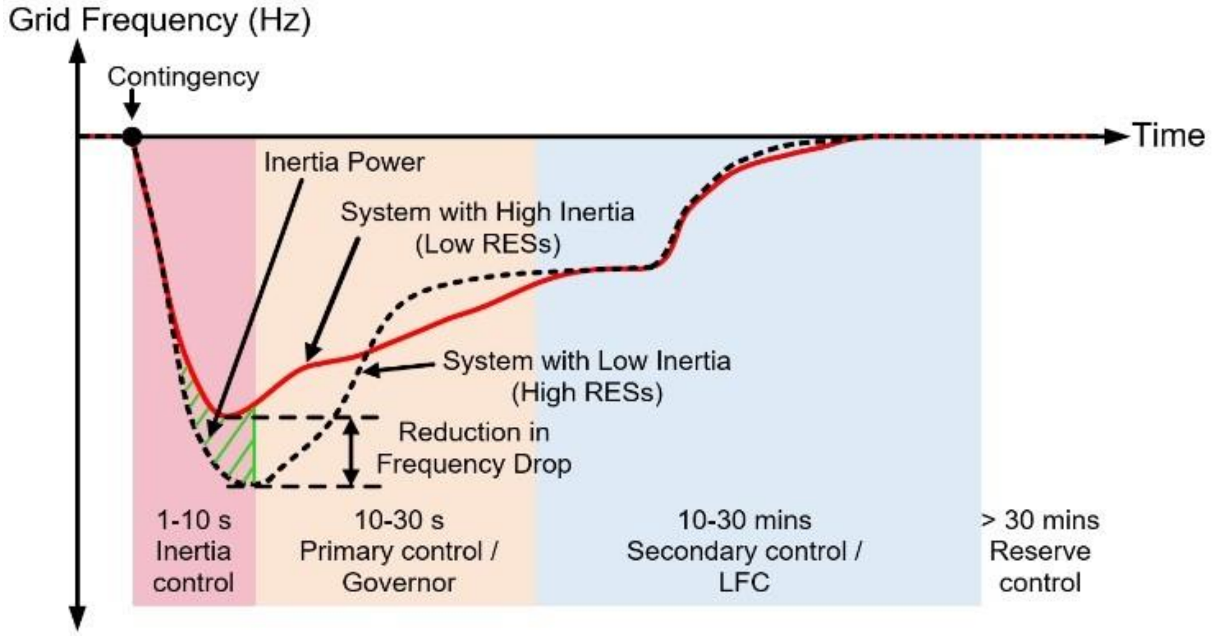
Vineet Jagadeesan Nair (jvineet9@mit.edu)*

Active-Adaptive Control Laboratory
Department of Mechanical Engineering
Massachusetts Institute of Technology

*Work done as part of a research internship at X, the moonshot factory (formerly Google X)

# Motivation

- Rapid power grid decarbonization
- Transition from fossil fuels to renewables
- Coal & natural gas plants are **synchronous generators** (SGs) → High inertia helps stabilize supply-demand imbalances
- Renewables (solar, wind) & batteries are **inverter**-based resources (IBRs) → Little to no inertia!
- Stability & reliability issues for the future grid



# Transient stability analysis

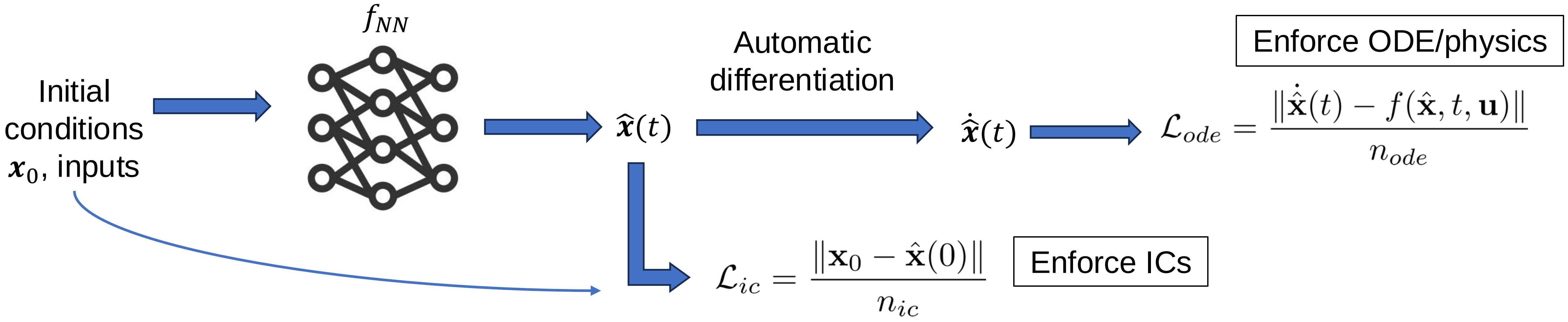
- Solve power system dynamics → Estimate grid frequency
- Crucial to assess transient stability esp. during disturbances (e.g. faults, line or generator outages, extreme weather)
- Involves systems of nonlinear ordinary differential equations (ODEs)
- Need to solve at very high resolution (milli- to micro-seconds)
- Cons of conventional numerical integration methods:
  - Expensive, may require small $\Delta t$ for stability
  - Not scalable for the future distributed grid with millions of IBRs

Use physics-informed machine learning to directly predict ODE solutions & accelerate dynamic simulations with high accuracy

# Physics-informed neural networks

- Approximate solution to initial value problem with neural network

$$\dot{\mathbf{x}} = f(\mathbf{x}, t, \mathbf{u}), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad \hat{\mathbf{x}}(t) = f_{NN}(\mathbf{x}_0, t, u\ \boldsymbol{\theta})$$




- Minimize combined loss function to train PINN (with MLP layers)

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \lambda_{ic}\mathcal{L}_{ic} + \lambda_{ode}\mathcal{L}_{ode}$$

# Prior work and contributions

- A few papers have applied PINNs for power systems
- Most studies focus on dynamics of synchronous generators (SGs)
- Only 1 paper has considered PINNs for inverters
- Prior works generally use simplified or reduced order models
  - E.g. $2^{nd}$ order swing equation for SG,
    Or only 1 component of inverter (i.e. phase locked loop converter)
- **<u>Our goal</u>**: Develop enhanced PINNs that can be also applied to more complex, higher-order, & higher-dimensional ODE models
  → $4^{th}$ order SG model
  → Full-order inverter model

# Challenges with training PINN

- Difficult to train for higher-order, stiff ODE systems
- Ill-conditioned loss function, especially near optimum
  → Due to differential operator in ODE residual term
  → Many local minima & saddle points
- Curse of dimensionality
  → Expensive training & inference for higher-order ODEs
- Poor convergence, stability, and generalization issues

# Proposed PINN enhancements

1. Initialize regularization weights $\lambda_{ic}, \lambda_{ode}$ using normalization strategies from multiobjective optimization
   → Based on *Utopia* and *Nadir* points in the Pareto set
2. Adaptively tune loss term weights during training
   → Based on intermediate values of their gradients

(1) + (2) → Both losses have similar magnitudes & gradients
→ Balance learning ODE while also satisfying ICs

3. Sequence-to-sequence learning: PINN only predicts the very next time step instead of the whole time domain
4. Hyperparameter tuning & optimization
   e.g. Combining optimizers: ADAM (1st order gradient-based method) followed by L-BFGS (2nd order, quasi-newton method)

# Synchronous generator (sgPINN)

Model both SG and inverter using nonlinear state space approach:
$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + R(\boldsymbol{x}, \mathrm{u})$ (matrix $R$ captures all nonlinearities)

<u>Predict all 4 states</u>

$(i_d, i_q)$: generator currents in the d-q coordinates

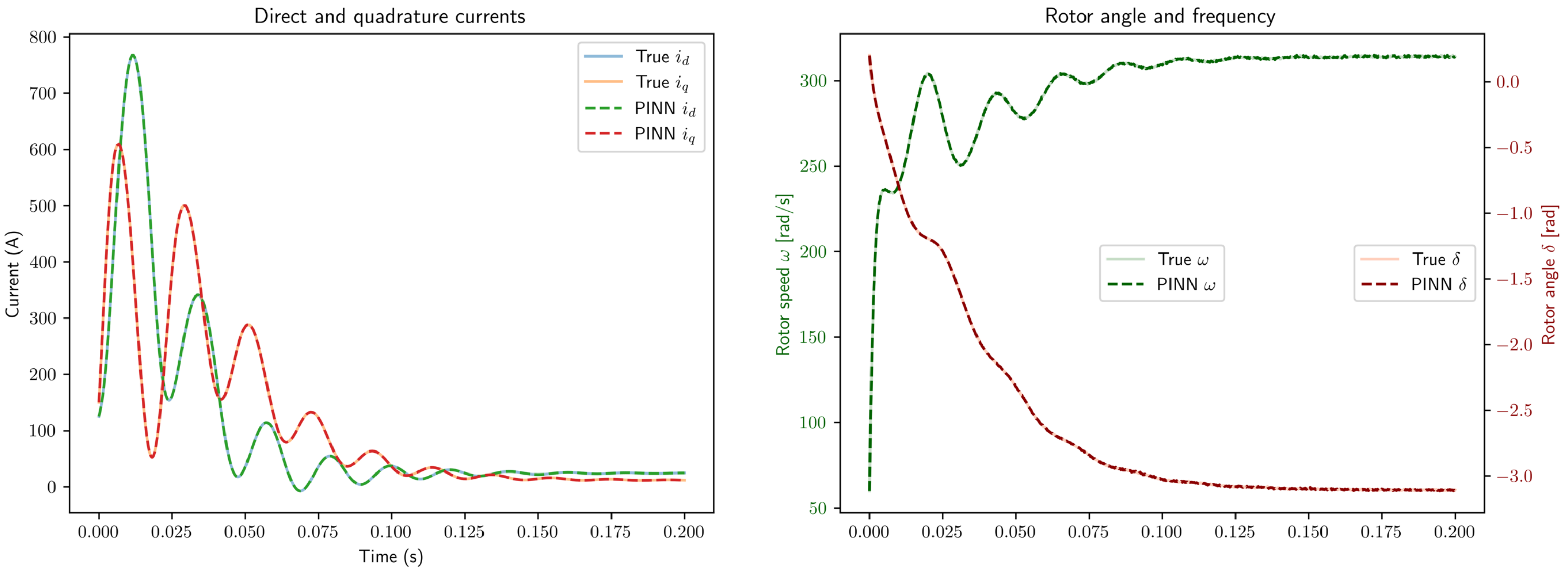
$\delta$: angle difference

**$\boldsymbol{\omega}$: generator frequency**

$$\begin{bmatrix} \dot{i}_d \\ \dot{i}_q \\ \dot{\omega} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_s} & \omega & 0 & 0 \\ -\omega & -\frac{R_s}{L_s} & -\frac{mi_f}{L_s} & 0 \\ 0 & \frac{mi_f}{J} & -\frac{D_p}{J} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_d \\ i_q \\ \omega \\ \delta \end{bmatrix} + \begin{bmatrix} \frac{V}{L_s}\sin\delta \\ \frac{V}{L_s}\cos\delta \\ \frac{T_m}{J} \\ -\omega_g \end{bmatrix}$$

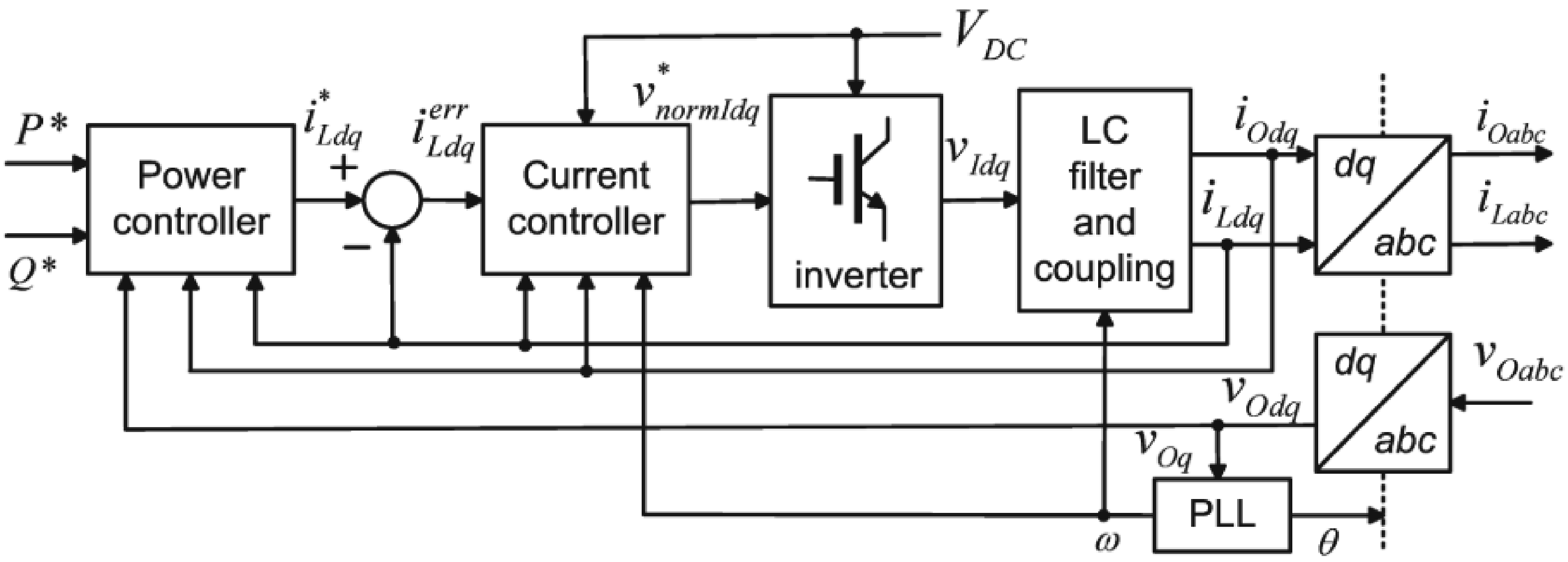
| **Parameter** | **Symbol** | **Value** |
|---|---|---|
| Nominal Grid Frequency | $\omega_g$ | $100\pi$ rad/sec |
| Stator Resistance | $R_s$ | 0.152 $\Omega$ |
| Stator Inductance | $L_s$ | 4.4 mH |
| Damping Coefficient | $D_p$ | 10.14 Nm/(rad/sec) |
| Inertia Constant | $J$ | 0.02 Kgm$^2$/rad |
| Mechanical Torque | $T_m$ | $[15.9 + D_p\omega_g]$ Nm |
| Nominal Voltage | $V$ | $230\sqrt{3}$ Volts |
| Field Excitation Constant | $m_{if}$ | $-1.38$ Voltsec |

# Preliminary sgPINN results



# Inverter PINN model (invPINN)

- Complex grid-following inverter model with 17 total states




$$\mathbf{x} = \begin{bmatrix} \theta & \Phi_{\text{PLL}} & i^*_{Ld} & i^*_{Lq} & q_{3Ld} & q_{3Lq} & q^{\text{err}}_{Ld} & q^{\text{err}}_{Lq} & i_{Ld} & i_{Lq} & i_{LO} & v_{Cd} & v_{Cq} & v_{CO} & i_{Od} & i_{Oq} & i_{OO} \end{bmatrix}^T$$

# Inverter PINN model (invPINN)

$$
A = \begin{bmatrix}
0 & K_I^{PLL} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\sqrt{2}\omega_c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega_c^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\sqrt{2}\omega_c & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega_c^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{R}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{R}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{C} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{C} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{C} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{C} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{R_{\text{coupl}}}{L_{\text{coupl}}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{R_{\text{coupl}}}{L_{\text{coupl}}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{C} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{C} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{C} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{C} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{R_{\text{coupl}}}{L_{\text{coupl}}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{R_{\text{coupl}}}{L_{\text{coupl}}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{3R_G - R_{\text{coupl}}}{L_{\text{coupl}}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{3R_G - R_{\text{coupl}}}{L_{\text{coupl}}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

Nonlinear state space model:

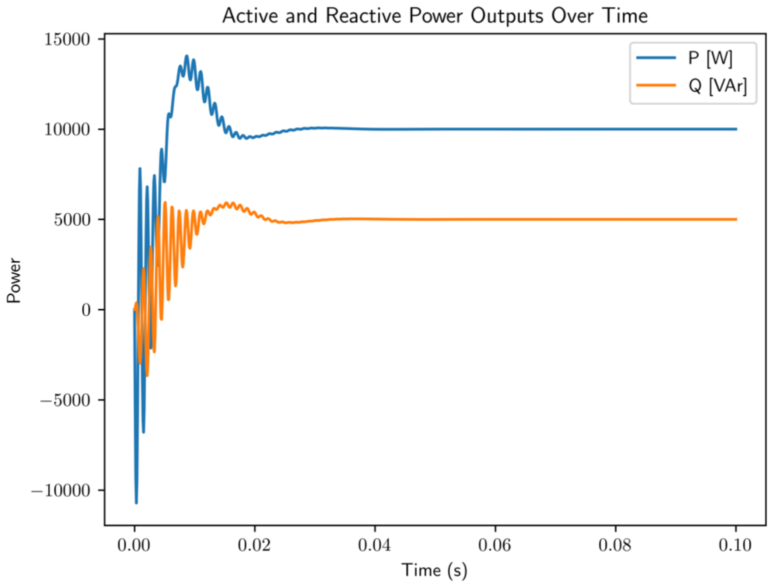
$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + R(\boldsymbol{x}, \mathrm{u})$$

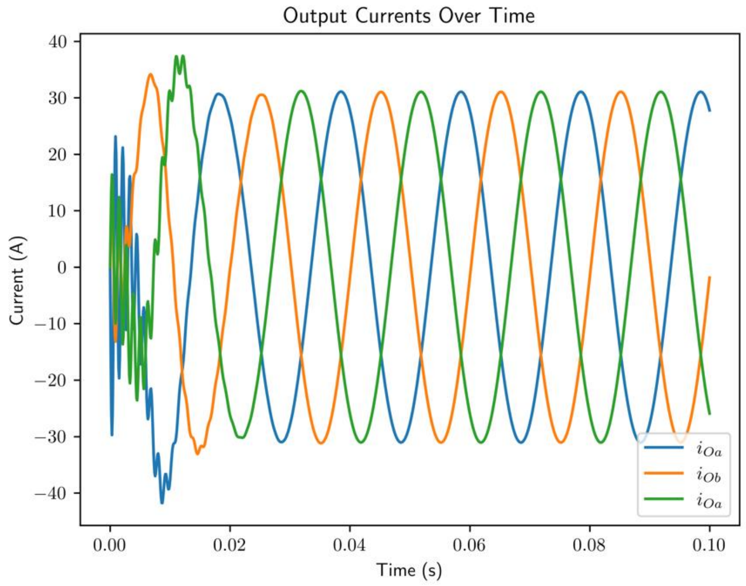
$$
R(\mathbf{x}, \mathbf{u}) = \begin{bmatrix}
K_P^{\text{PLL}} v_{Oq} \\
v_{Oq} \\
0 \\
0 \\
\frac{v_{Od}P^* - v_{Oq}Q^*}{v_{Od}^2 + v_{Oq}^2} \\
\frac{v_{Oq}P^* + v_{Od}Q^*}{v_{Od}^2 + v_{Oq}^2} \\
0 \\
0 \\
\frac{1}{L} v_{1d} + \omega i_{Lq} \\
\frac{1}{L} v_{1q} - \omega i_{Ld} \\
\omega v_{Cq} \\
-\omega v_{Cd} \\
-\frac{1}{L_{\text{coupl}}} v_{Od} + \omega i_{Oq} \\
-\frac{1}{L_{\text{coupl}}} v_{Oq} - \omega i_{Od}
\end{bmatrix}
$$

| **Parameter** | **Value** |
|---|---|
| Nominal phase voltage | 240 V |
| Grid frequency | 50 Hz |
| Coupling impedance $R_{coupl}$, $L_{coupl}$ | $(0.131 + j0.96)\ \Omega$ |
| DC bus voltage | 1000 V |
| **Coupling filter** | |
| Inductance $L$ | 1.35 mH |
| Capacitance $C$ | 50 $\mu$F |
| Resistance $R$ | 0.056 $\Omega$ |
| Ground resistance $R_G$ | 100 $\Omega$ |
| Switching frequency $\omega_c$ | 100 Hz |
| Reference active power $P^*$ | 10 kW |
| Reference reactive power $Q^*$ | 5 kVar |
| **Current controller** | |
| Proportional gain in branch $d$ $K_P^d$ | 1 |
| Integral gain in branch $d$ $K_I^d$ | 460 |
| Proportional gain in branch $q$ $K_P^q$ | 1 |
| Integral gain in branch $q$ $K_I^q$ | 460 |
| **PLL** | |
| Proportional gain $K_P^{PLL}$ | 2.1 |
| Integral gain $K_I^{PLL}$ | 5000 |

# Inverter simulations → invPINN challenges

- Poor performance & generalization for invPINN → Why?




- Higher order
- Predicting more states simultaneously
- Stiffer ODEs
- Need more careful parameter tuning to avoid numerical instability

**Fast timescales also make training challenging**

- Inverter transients settle down in less than 10-20 milliseconds
  → After which output powers track their reference setpoints
- But SG transients take around 100-200 milliseconds to settle down
- Inverter dynamics (smaller electrical time constants) much faster than SG (larger mechanical time constants)
- Under certain conditions and disturbances, inverters can have even faster transients (microseconds)

# Next steps

- Improve generalization of sgPINN to diverse inputs & ICs
- Improve overall performance of invPINN
- Implement other types of layers in our PINNs e.g. RNN, LSTM
- Possible extensions for future work
  - Apply PINNs for ODE parameter estimation
  - Larger-scale studies with multiple SGs & IBRs
  - Bayesian PINNs for uncertainty quantification