Neural Networks in RSCAD: Enhancing MMC-based HVDC Simulation with Advanced Machine Learning

- Rahul Rane
- TU Delft

This work is done by B. Masalmeh, as part of his MSc thesis: ``Neural Network-Based Predictive Control for Modular Multilevel Converters in HVDC Transmission Grids'', and B. Masalmeh, R. Prasad, V. Nougain and A. Lekić , "Neural Networks in RSCAD: Enhancing MMC-based HVDC Simulation with Advanced Machine Learning," in *IEEE Transactions on Industry Applications*, 2024 (Under Review)

UGM RTDS USER'S GROUP MEETING

2024 EUROPE USER'S GROUP MEETING
DELFT, NETHERLANDS

RTDS Technologies AMETEK

Introduction

- **Advancements in Control Systems and Need for Real-Time Simulation Platform**
  - Rapid progress in control systems for inverter-based resources, with dynamics in the order of several kHz
  - Real-time simulation platform is the most suitable for testing
- **Neural Networks' Strengths**
  - Managing highly nonlinear systems
  - Handling large-scale data
  - Adapting to changing system conditions
- **Constraints in Testing Neural Network Models in Real-Time**
  - Computational hardware capabilities
  - Lack of complex programming features
  - Mathematical complexity of models

Research and Innovation

- **Unique Development and Application**
  - Development of a multilayer neural network library in RSCAD
  - A library designed to run on RTDS through the CBuilder interface to test neural network performance in a real-time simulation
- **Predictive Adaptive Control**
  - A predictive adaptive model-free approach for MMC control that serves as a replacement for the traditional PI controller
  - Application of such control methods on large-scale MMC controllers for high-power HVDC transmission
- **Real-Time Control Strategies**
  - Potential for real-time complex control strategies, offering a new direction for future research in control methods

What's Included in the Neural Network Toolbox?

- **Artificial Neural Network (ANN) Component**
  - Key Features:
    - Supports up to 4 hidden layers
    - Can manage up to 100 weights (connections) between layers
- **Recurrent Neural Network (RNN) Component**
  - Used for processing sequential data over time
  - Key Features:
    - Can handle up to 10 variables (input features)
    - Supports up to 20 time-steps for sequence-based data
- **Layer Component**
  - Build flexible, deep neural networks
  - Key features:
    - No limit on the number of hidden layers
    - Each layer can be customized with different activation functions, and learning rates

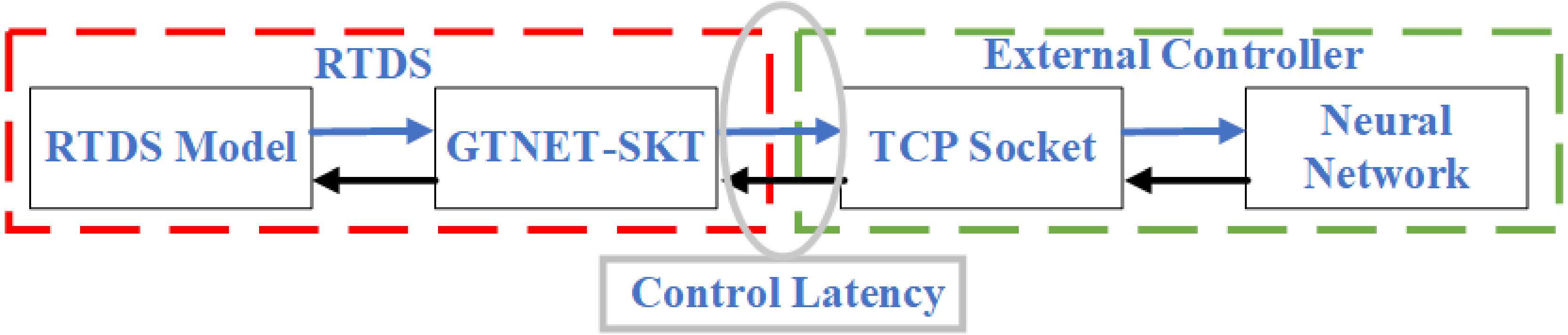
Training Neural Networks in RSCAD

- **Online Training**
  - Real-time training of neural networks
- **Offline Training**
  - Can be used to reduce computational time and complexity
  - Utilizes generated simulated data to train an offline neural network
  - Loading the weights and bias matrices in the online neural network component

Key Considerations for Developing the Neural Network Toolbox

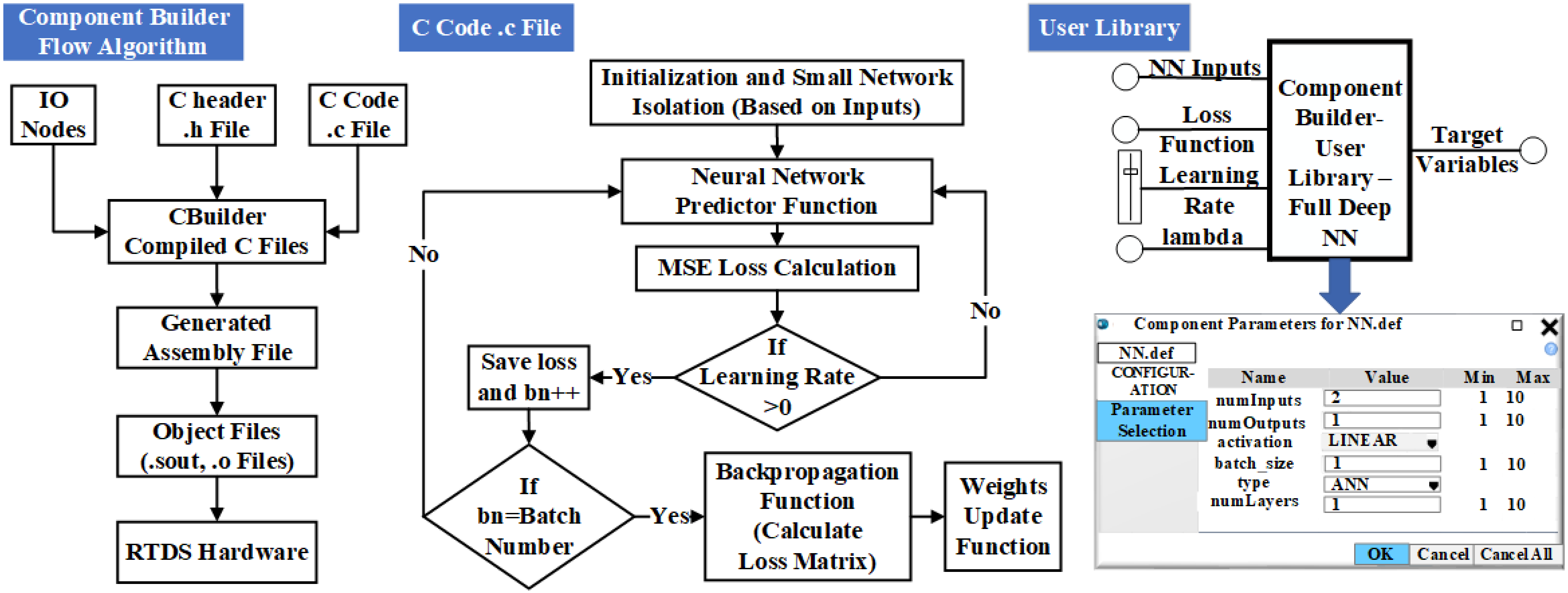
- **Execution Time Constraint**
  - Ensure execution time remains shorter than 50 µs
- **Overfitting Risk from Constant Data Stream**
  - Constant data streaming can lead to overfitting in models if the untrained model sees too many data points in the same output range
- **Utilizing CBuilder Functionalities (No Dynamic Memory Allocation)**
  - CBuilder functionalities must be used without relying on dynamic memory allocation (i.e., no malloc())
- **Communication Delays Between External Devices**
  - Delays in communication with external devices can affect the system's real-time performance

Communication Delays Between External Devices

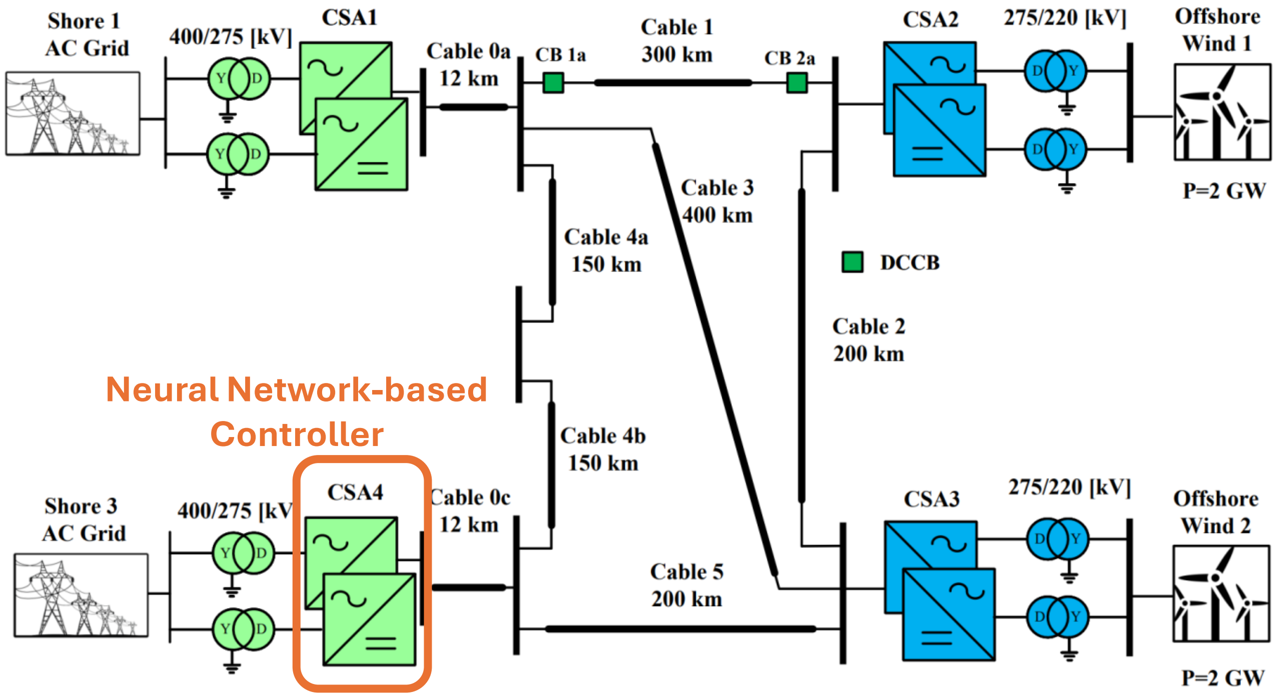


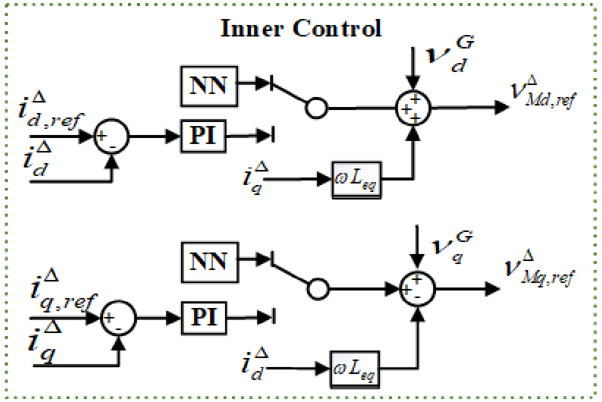
- Significant control latency can hinder the functionality of the controller

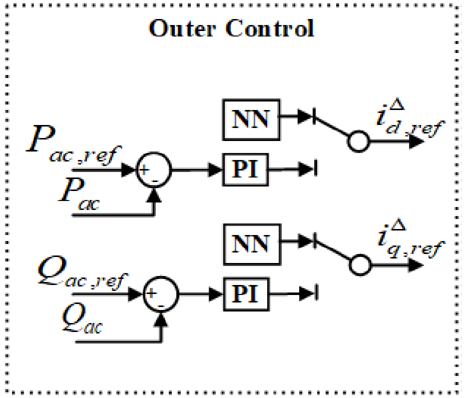
Development of the Toolbox using CBuilder



Test Case: Large-Scale MMC Controllers for High-Power HVDC Transmission





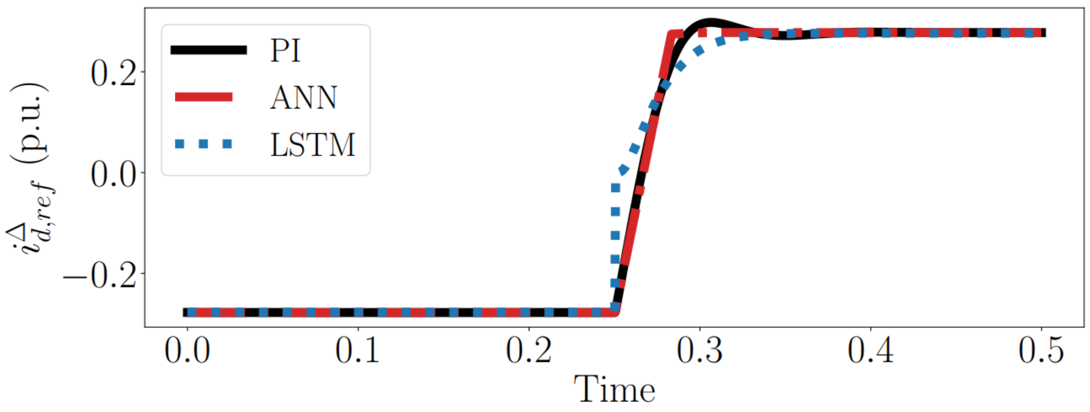


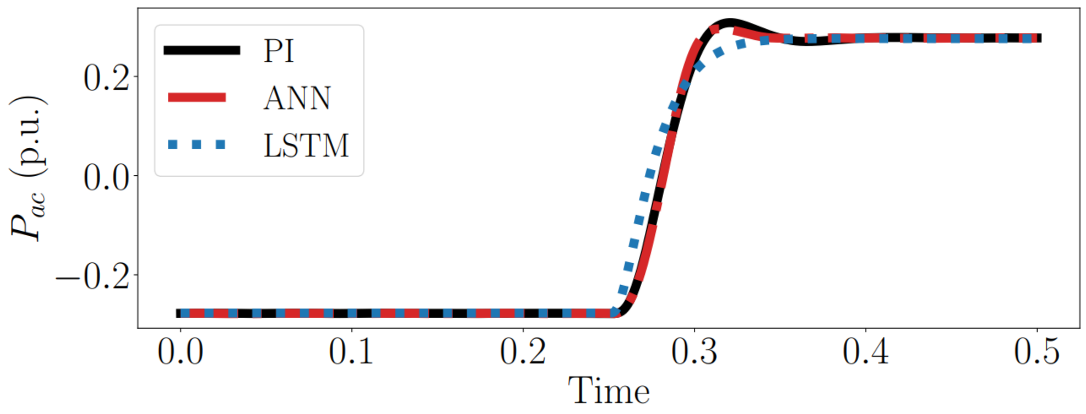
A. Shetgaonkar, M. Popov, A. Lekic, T. Karmokar, "Enhanced Real-Time Multi-Terminal HVDC Power System Benchmark Models with Performance Evaluation Strategies", *CIGRE Science and Engineering*, 32, 2024

Reinforced Learning through an Error-Tracking Mechanism

- **Training Process**
  - The neural network trains using the direct value from the PI controller
  - Since the target values represent the PI control action, the neural network essentially mimics the PI controller
- **Improving Neural Network Performance Over PI**
  - To surpass the performance of the PI, the loss function needs an additional term
  - This additional term accounts for the shortcomings of the PI by considering the remaining error after the PI control action
- **Modified Loss Function**
  - The loss function can be defined as: $Loss = MSE\ Loss + \lambda\big(err(t+1)\big)$
  - Here $err(t+1)$ represents the next time step error due to the control action, and λ is a scaling constant.

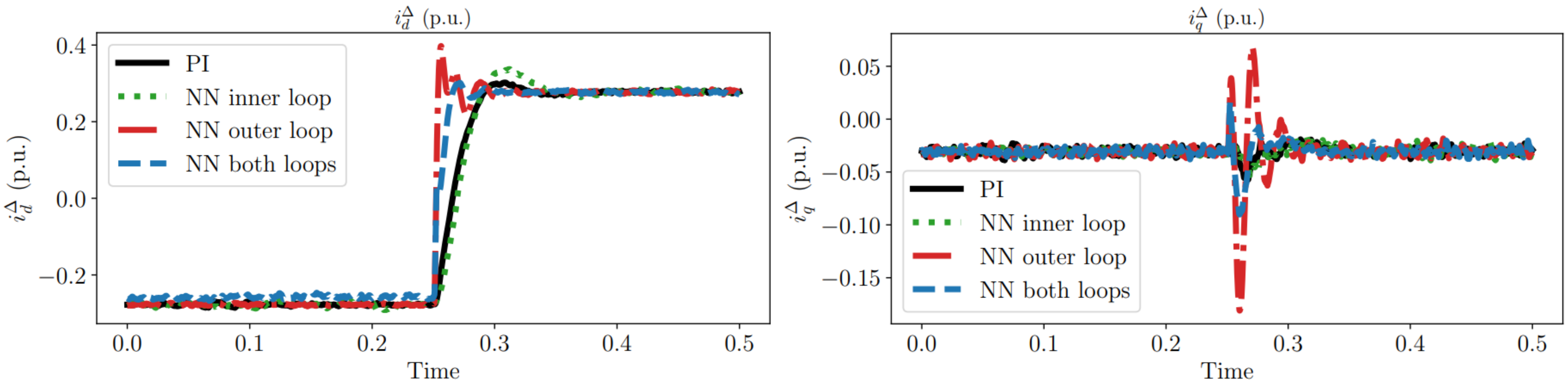
Comparison of Different Types of Neural Network Models





- Significantly lower peak overshoot is observed when using an LSTM-based controller

Comparison of Conventional PI and Neural Network-based Controllers



- Using neural network-based controllers provide a better transient response

Conclusion

- This work presents a distinctive application of advanced neural network control techniques implemented within RTDS
- Demonstrated the stability of MMC controllers in a large-scale test case
- Toolbox Features:
  - Options for offline, online, or combined training of neural network components
  - Supports both ANNs and RNNs with 1, 2, 3, or more layers
  - Provides full flexibility in defining the structure of the neural network and the learning method
  - Supports any loss function defined in RSCAD as the training loss
  - Facilitates training for control of both inner and outer control loops or just one of them
- The toolbox opens the door to numerous possibilities for real-time testing using neural networks

Thanks for your attention!

Rahul Rane

TU Delft