Machine Learning Methods for Feedforward Power Flow Control of Multi-Active-Bridge Converters

Mian Liao, *Student Member, IEEE*, Haoran Li, *Student Member, IEEE*, Ping Wang, *Student Member, IEEE*, Tanuj Sen, *Student Member, IEEE*, Yenan Chen, *Member, IEEE*, and Minjie Chen, *Senior Member, IEEE*

***Abstract*—Controlling the multiway power flow in a multi-active-bridge (MAB) converter is important for achieving high performance and sophisticated functions. Traditional feedforward methods for MAB converter control rely on precise lumped circuit models. This paper presents a machine learning (ML) method for feedforward power flow control of a MAB converter without a precise circuit model. A feedforward neural network (FNN) was developed to capture the non-linear characteristics and predict the phases needed to achieve the targeted power flow. The neural network was trained with a large amount of data, collected with a set of known phase angles. This trained network was used to predict the phases to achieve the targeted power flow. A 6-port MAB converter was built and tested to validate the methodology and demonstrate the "machine-learning-in-the-loop" implementation. Transfer learning was proven to be effective in reducing the size of the training data needed to obtain an accurate ML model. ML-based feedforward power flow control can achieve comparable accuracy as traditional model-based methods, and can function without a precise lumped circuit element model of the MAB converter.**

***Index Terms*—multi-active-bridge converter, power flow control, machine learning, artificial intelligence, neural network, transfer learning, machine-learning-in-the-loop**

I. Introduction

MULTI-Active-Bridge (MAB) converters connect many sources and loads through a single power conversion stage and manage the multiway power flow. MAB converters are attractive in many applications including multiport energy routers, battery balancers, and differential power processing (DPP) systems in data centers and photovoltaic applications [1]–[6]. Fig. 1 illustrates the operation principles of a MAB converter with many dc-ac units coupled to a single magnetic core. The power flow is controlled by the phase-shift modulation. Each port is connected to a dc source or a load with a constant voltage. In a MAB converter, the ports with leading phases inject power into the multi-winding transformer, and vice versa for the ports with lagging phases [7], [8]. The advantages of MAB converters include: 1) low power conversion stress; 2) low component count; 3) fast dynamic response; 4) high system efficiency; and 5) high power density. Many control strategies have been developed for MAB converters [9]–[11].

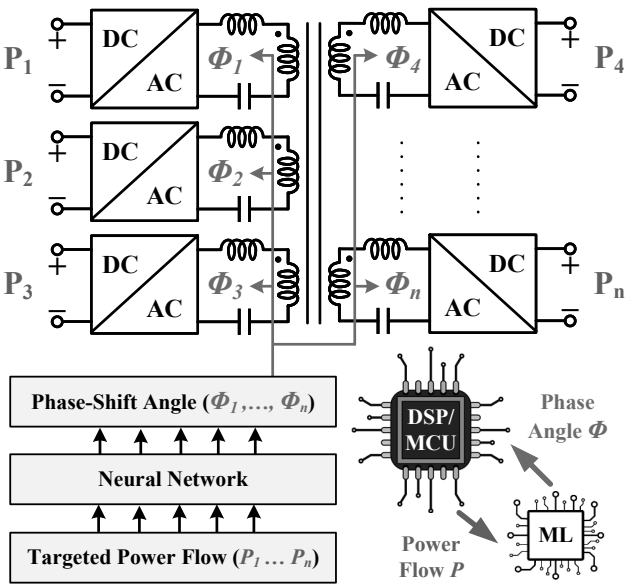


Fig. 1. An example MAB converter with power flow controlled by phase shifts. Leveraging microprocessors with specialized ML accelerators, the phase angles of a MAB converter can be predicted by a neural network. The training of the neural network is offline trained and online inferred.

One way to determine the required phase shifts for the desired power flow in a MAB converter is to derive an equivalent circuit model and find numerical solutions as presented in [12], [13]. However, the equations connecting phase shifts to power flow are usually non-linear and may require iterative algorithms to solve, such as the Newton-Raphson (NR) method, the Quasi-Newton (QN) method, and the Stochastic Gradient Descent (SGD) [14], [15]. Iterative algorithms need parameter initialization, and may not converge. Equation-based control cannot capture non-idealities and non-linearities such as parasitic capacitances/inductances and equivalent resistances, leading to inaccurate results. Moreover, there is no closed-form equation to derive the power flow in MAB converters operating with piecewise-sinusoidal currents.

The focus of this paper is feedforward power flow control

This paper is an extension of a previously published conference paper, "Machine Learning Methods for Power Flow Control of Multi-Active-Bridge Converters" in IEEE COMPEL 2021 [1]. This work was jointly supported by the ARPA-E CIRCUITS program, National Science Foundation, and American Tower Corporation through the Princeton Andlinger Center E-ffiliates Program. *(Corresponding Author: Minjie Chen & Yenan Chen)*

M. Liao, H. Li, P. Wang, T. Sen, and M. Chen are with the Department of Electrical and Computer Engineering and the Andlinger Center for Energy and the Environment at Princeton University, Princeton, NJ, 08540, USA (e-mail: mianl, haoranli, ping.wang, tsen, minjie@princeton.edu).

Y. Chen is with ZJU-Hangzhou Global Scientific and Technological Innovation Center and the College of Electrical Engineering at Zhejiang University, Hangzhou, China (e-mail: yenanc@zju.edu.cn). He was with the Department of Electrical and Computer Engineering and the Andlinger Center for Energy and the Environment at Princeton University, Princeton, NJ, 08540, USA.

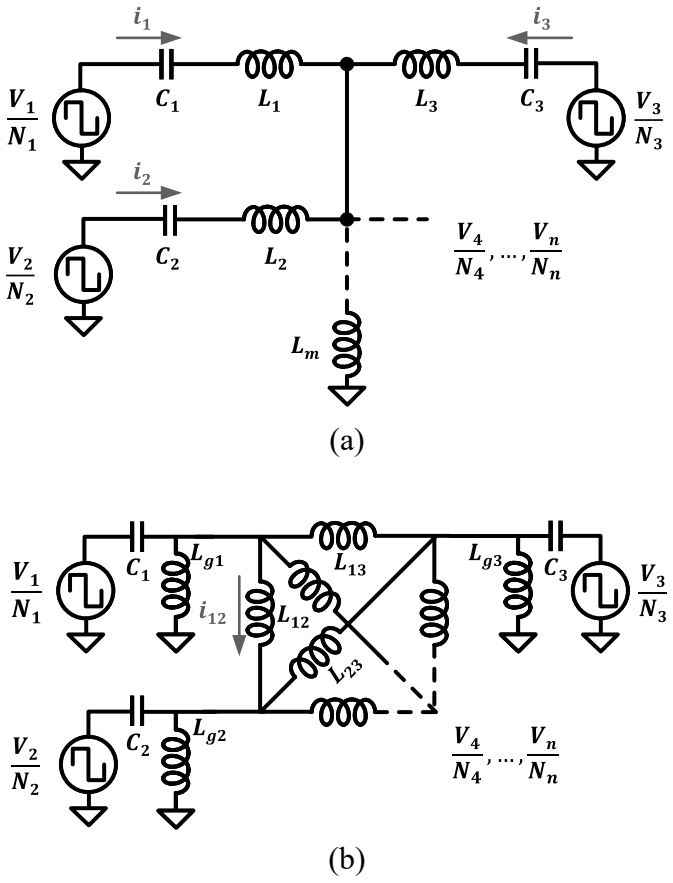


Fig. 2. (a) Star and (b) Delta equivalent circuit model of an N-port MAB converter. The element values in this model may be non-linear or non-ideal.

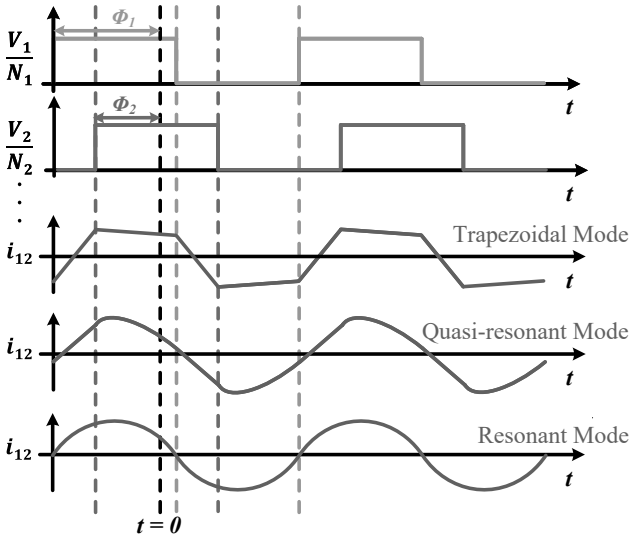


Fig. 3. Operating principles of a MAB converter in trapezoidal, quasi-resonant and resonant modes with phase shift control. If the blocking capacitor is large, the MAB converter operates with trapezoidal branch current. If the blocking capacitor is small, the MAB converter current is piecewise-sinusoidal.

of multi-active-bridge converters. Similar to the feedforward power flow control models in dual-active-bridge (DAB) converters [7], [8], we develop a neural network model to estimate the phase needed to achieve a target power flow in a multi-active-bridge converter (MAB). The recent advances in ML and artificial intelligence (AI) offer new opportunities in building models and developing control algorithms for sophisticated systems without precise physical models. Artificial neural networks (ANN) have been proven effective in capturing complex non-linear relationships among clearly defined inputs and outputs, and have been explored in power electronics applications [16], including the modeling of semiconductor devices [17], motors and drives [18], converter design [19], [20], magnetics [21]–[24], and grid impedances [25]. Many edge computing controller platforms, such as STM NanoEdgeAIStudio, TI Edge AI, and Arduino Nano, have embedded ML accelerators.

This paper applies ML methods to control the power flow in a MAB converter. A Neural Network (NN) model replaces the traditional numerical solver and serves as the feed-forward path of a complete feedback control loop. It takes the targeted power flow as the input and predicts the phase angles that can produce the targeted power flow. Transfer learning is used to pretrain a model with theoretical data and optimize the pretrained neural network with a small experimental dataset. A 6-port MAB converter was built and tested to verify the theory, prove the concept, and to demonstrate the hardware implementation of the ML-based power flow control. Experimental results show that the ML-based power flow control can achieve comparable accuracy as traditional NR methods when precise lumped circuit models are known, and can capture the non-linearites and non-idealities that cannot be captured by traditional model-based methods.

Feedforward control allows the system to rapidly transition from one steady-state to another steady-state, and is critical for MAB converters operating as energy routers [11]. The main contribution of this paper is a systematic approach to implementing feedforward power flow control of MAB converters with ML. We demonstrate the unique strengths of ML as compared to traditional model-based feedforward control, and present an end-to-end ML solution with hardware implementation, including (1) automatic data acquisition; (2) neural network control architecture design; (3) hardware implementation of machine learning; and (4) transfer learning for data size reduction. The concepts and methodologies explored in this paper can be widely applied to other sophisticated control problems in power electronics. The proposed ML framework can be extended to include more advanced ML methods, such as deep learning, reinforcement learning, and physics-informed ML [26].

II. Traditional Power Flow Models and Control Methods for MAB Converters

Fig. 2 shows the (a) Star and (b) Delta equivalent circuit models of a MAB converter. Each branch of the MAB converter is modeled as a voltage source (V_i) connected in series with a capacitor (C_i) and a linking inductor (L_i). There is a magnetizing inductance L_m in each branch. The power flow between ports is determined by the linking inductance

and the series capacitance. In specific cases, the Star model is interchangeable with the Delta model as presented in [12].

The voltage and current waveforms at a port, with phase-shift modulation, are illustrated in Fig. 3. If the impedance of the series capacitance (C_i) is negligible, the system operates with piecewise-linear current waveform. If the impedance of the series capacitance is comparable to the impedance of the linking inductance (L_i), the system operates with piecewise-sinusoidal current waveform. The power flow can be calculated based on the lumped circuit model and the corresponding phase shift, but there is no explicit equation to solve for the phase shift based on the required power flow.

Both feedforward and feedback loops are needed to achieve the desired power flow in a MAB converter. The feedforward control can quickly predict control variables based on a theoretical circuit model, while the feedback control can compensate for non-idealities due to disturbances or manufacturing variations of components. A traditional feedforward control uses iterative algorithms to calculate the phase shift from the required power flow. The feedback control presented in [4] used PI controllers. Feedforward and feedback control loops can be applied together to precisely regulate the power flow for large-signal modulation and small-signal stability. The focus of this paper is to develop a feedforward control strategy for MAB converters using ML.

A. Trapezoidal Operation of MAB Converters

If the capacitance values are large enough such that only the linking inductances impact the power flow, the MAB converter operates with trapezoidal current. In a Delta model, the inductance L_{ij} can be calculated from the linking inductance (L_1 to L_n) in a Star model following [27]:

$$L_{ij} = \left(L_i + L_{THi}\right)\left[L_j\left(\frac{1}{L_m} + \sum_{k\neq i,j}^{n}\frac{1}{L_k}\right) + 1\right], \quad (1)$$

where the Thevenin-equivalent inductance seen by port i is:

$$L_{THi} = \left(\frac{1}{L_m} + \sum_{k\neq i}^{n}\frac{1}{L_k}\right)^{-1}. \quad (2)$$

As a result, the total power fed into the passive network from the i^{th} port is a function of ϕ_{ij}, V_i, and V_j:

$$P_i = \sum_{j=1}^{n}\frac{V_i V_j}{2\pi f_s N_i N_j L_{ij}}\phi_{ij}\left(1 - \frac{|\phi_{ij}|}{\pi}\right). \quad (3)$$

The power fed into the i^{th} port is determined by the phase angle and voltage of all ports. The power between each port monotonically changes from negative to positive when ϕ_{ij} sweeps from $-\frac{1}{2}\pi$ to $\frac{1}{2}\pi$. The non-idealities in the circuit element values may cause mismatches between the theoretical analysis and experimental measurements, resulting in inaccurate phase estimations for power flow control.

B. Quasi-Resonant Operation of MAB Converters

If the series L-C tank resonates at the switching frequency with low impedance, the MAB converter operates with resonant current. With the fundamental frequency analysis, the input of each port can be considered as a sinusoidal voltage source. In this case, the impedance matrix of the passive network described as a Star network is:

$$Z = \begin{bmatrix} Z_1 + Z_m & Z_m & \dots & Z_m \\ Z_m & Z_2 + Z_m & \dots & Z_m \\ \vdots & \vdots & \ddots & \vdots \\ Z_m & Z_m & \dots & Z_n + Z_m \end{bmatrix}. \quad (4)$$

where Z_i represents the branch impedance of port i and Z_m represents the impedance of the magnetizing inductance L_m.

$$Z_i = j\left(\omega L_i - \frac{1}{\omega C_i}\right), \quad Z_m = j\omega L_m. \quad (5)$$

The admittance matrix is the inverse of the impedance matrix:

$$Y = Z^{-1} = \begin{bmatrix} Y_{11} & \dots & Y_{1n} \\ \vdots & \ddots & \vdots \\ Y_{n1} & \dots & Y_{nn} \end{bmatrix}, \quad (6)$$

$$Y_{ij} = G_{ij} + B_{ij} = \begin{cases} y_i + \sum_{k\neq i} y_{ik} & if\ i = j \\ -\,y_{ij} & if\ i \neq j \end{cases} \quad (7)$$

As described in [12], if a MAB converter operates with resonant current, the active power fed into each port can be modeled following the methods in power system analysis [28]:

$$P_i = \frac{8}{\pi^2}\sum_{k=1}^{n} V_i V_k \left(G_{ik}\cos(\phi_{ik}) + B_{ik}\sin(\phi_{ik})\right). \quad (8)$$

where V_i and V_k are the amplitudes of the square wave voltage at port i and port k.

For a MAB converter operating with purely sinusoidal current waveforms, the power between each port also monotonically changes from negative to positive when ϕ_{ij} sweeps from $-\frac{1}{2}\pi$ to $\frac{1}{2}\pi$, which is the same as that for an ideal MAB converter operating in the trapezoidal mode, yet the nonidealities and non-linearities in the circuit may cause discrepancies in the power flow analysis. For a MAB converter operating in the quasi-resonant mode, a piecewise-sinusoidal current results in sophisticated power flow that cannot be solved for by explicit equations [12]. Equation-based models cannot accurately predict the power flow in this case. The accuracy of equation-based feed-forward power flow control may rapidly decrease as the switching frequency or the number of ports increases.

C. Limitation of Traditional Feedforward Power Flow Control

Traditional feedforward power flow control utilizes iterative algorithms to calculate phases from required power flow based on Eq. (3) or Eq. (8). However, these equations cannot capture non-idealities and non-linearities in real circuits. Non-idealities come from circuit elements that are not incorporated by lumped circuit models, including parasitic capacitances,

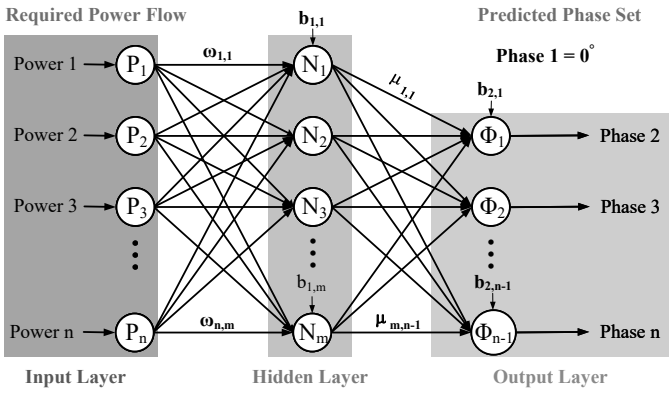


Fig. 4. Architecture of an example 3-layer feedforward neural network (FNN). The FNN has an input layer (P), an output layer (Φ), and a few hidden layers, each with weight (ω, μ) and bias (b) parameters.

leakage inductances, variable resistances, etc. Non-linearities come from the non-linear behavior of the circuit elements depending on the operating conditions. The capacitance value of a capacitor may change depending on the dc-bias voltage, and the inductance value of an inductor may change depending on the dc-bias current. The hysteresis loss of multi-winding transformer may change depending on frequency, dc bias and current waveform.

The circuit models for the multiwinding transformer of the MAB converter may be unknown to designers and may change significantly during the operation of the circuit. ML-based control can address these design challenges with unknown circuit parameters by capturing non-idealities and non-linearities from measured data. As long as the data used for training the neural network captures the change of the circuit parameters, the neural network model will capture the impact of the change in the circuit parameters. The neural network models developed in this work will replace the traditional feedforward models and function together with feedback loops to handle the change in circuit parameters and operating conditions.

III. Neural Network Models for Power Flow Control of MAB Converters

It is challenging to explicitly calculate the phase angles for a MAB converter required to achieve a target power flow based on equations. However, the input and output parameters of a MAB converter control problem are clear. They are similar to those of typical ML problems [26]. In this paper, the challenge of power flow control in a MAB converter is considered as a multi-input multi-output (MIMO) ML problem and solved using ML tools. The input is a vector of the power flow through each port, and the output is a vector of the phase shifts at each port. The phase at each port is modulated by an ML-based controller to achieve the targeted power flow.

We select Feedforward Neural Network (FNN) as an example method to control the MIMO power flow [26]. FNN is one of the most widely used neural networks with mature support for hardware deployment. In FNN, the information always moves in one direction — forward from the input nodes, through the hidden nodes, to the output nodes. No cycles or loops exist in the network. Fig. 4 shows an example architecture of a FNN applied to the MAB power flow control. The neurons in the input, output and hidden layers of this FNN are connected by:

$$N_j = S\left(\sum_{i=1}^{n}(w_{i,j}\cdot P_i)+b_{1,j}\right)\ ,\ j=1,\ldots,m, \tag{9}$$

$$\Phi_j = \sum_{i=1}^{m}(\mu_{i,j}\cdot N_i)+b_{2,j}, j=1,...,n-1. \tag{10}$$

$\omega_{i,j}$ is the weight of the neuron connected to the hidden layer. $b_{1,j}$ is the bias of the hidden layer neuron N_j. $\mu_{i,j}$ is the weight of the neuron connected to the output layer and $b_{2,j}$ is the bias of the output layer neuron Φ_j. *S* in Eq. (9) represents the *sigmoid* function, a commonly used activation function that creates non-linearity for neuron outputs. This FNN structure is used to capture the relation between the phase vector and the power vector for a MAB converter. Assuming the phase of Port #1 is selected as a reference port with $0°$, the input layer of the neural network has *n* neurons (representing the targeted power flowing into each port). The output layer of the neural network has *n-1* neurons (representing the desired phase at each port). One can place multiple hidden layers between the input and output layers. For a particular MAB converter, the numbers of input and output neurons are fixed while the numbers of hidden layers and the numbers of neurons in each layer are adjustable. Each neuron in the hidden layer or the output layer contains a bias parameter b. Every two neurons in different layers are connected by a weight parameter w. In the training stage, the $Adam$ optimizer [29] in TensorFlow is used to update the weights and biases based on the Mean Square Error (MSE) between predicted outputs and ground truth outputs. Weights and biases are updated after each training iteration to optimize these parameters and reduce the MSE. The performance of the FNN typically increases as the number of hidden layers or the number of neurons in each layer increases, but the size and complexity of the network will also increase. One should carefully size the neural network depending on the complexity of the problem and the number of training data to achieve high accuracy and avoid over-fittings.

IV. Machine Learning Infrastructure

Machine learning is data engineering. The size and quality of the datasets determine the applicability and accuracy of the modeling results. The computational needs in data acquisition and off-line model training are heavy, while the computational needs for model deployment and on-line algorithm inference are light. Here we introduce the hardware and software infrastructure needed to perform data acquisition and model training for our ML-based feedforward power flow control.

A. Hardware Infrastructure

The block diagram of the machine-learning data acquisition platform is presented in Fig. 5, including dataset collection, neural network training, ML-based control implementation and accuracy evaluation. In the training stage, substantial pre-selected phases are fed into the MAB converter to produce

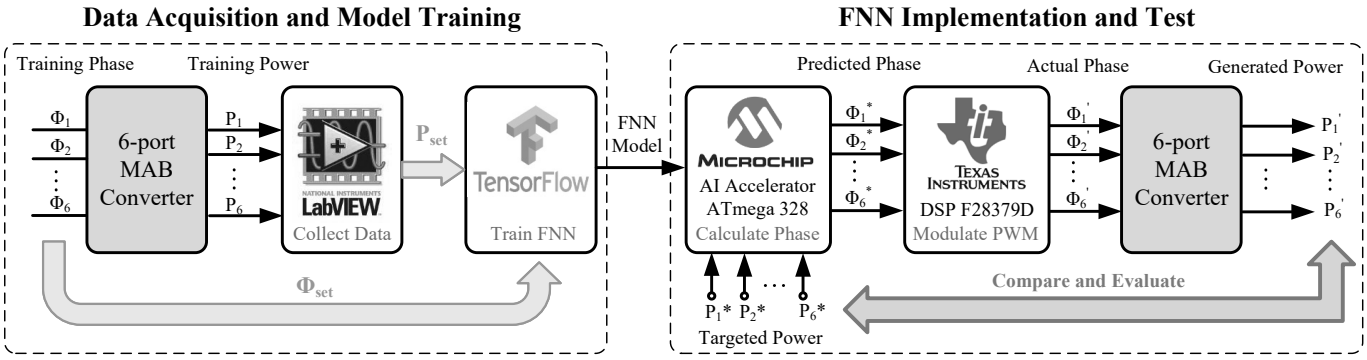


Fig. 5. Data acquisition system and control architecture of the MAB converter comprising 1) a 6-port MAB converter as the power stage, 2) a LabView system for collecting data, 3) a TensorFlow platform for model training, 4) an Al accelerator for model deployment, and 5) a DSP for power stage control.

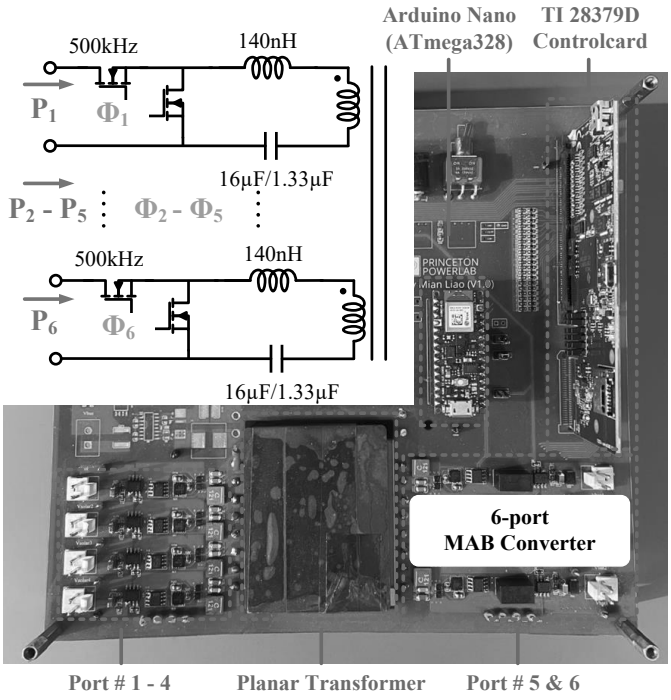


Fig. 6. The prototype 6-port MAB power stage and control circuitry. The 6-winding transformer was implemented as a planar transformer with windings embedded in the printed circuit board. The rated power for each port is 36 W in trapezoidal mode and 48 W in sinusoidal mode. The MAB converter switches at 500 kHz. The MOSFETs used in this prototype are CSD17318Q2T from Texas Instruments, and the magnetic core is ER64/13/51-3C92 from Ferroxcube. Each port switches at 12 V.

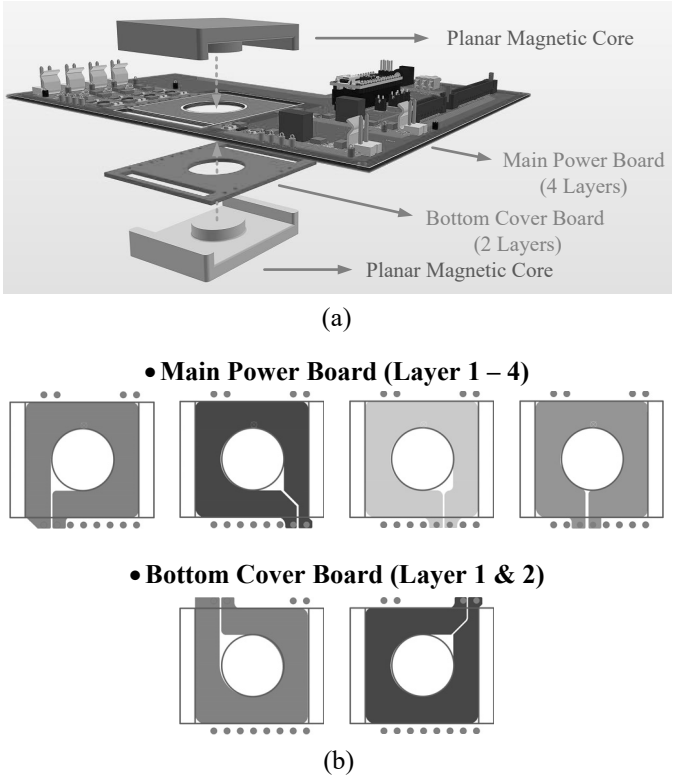


Fig. 7. (a) Assembly view of the MAB converter prototype with a planar magnetic core, two stacked PCBs, and related control and auxiliary circuits; (b) Layout pattern of the six layer PCB winding with a single turn.

corresponding power flow. A LabVIEW platform is used to measure and store the power flow to form phase-power datasets. Then, the generated datasets are utilized to train the FNN in the TensorFlow platform [30] and the trained FNN model is stored in the AI accelerator. In the implementation stage, when the system requests a specific power flow, the neural network accelerator infers the FNN model and calculates the corresponding phases. The calculated phase vector is sent from the neural network accelerator to a Digital Signal Processor (DSP), where the actual phase-shifted PWM signals are produced, in accordance with its phase resolution, and fed into the 6-port MAB converter. The LabVIEW system measures the voltage and current of each port and records the resulting power flow. Finally, the generated power flow is compared with the targeted power to verify the prediction accuracy of the ML method.

B. MAB Converter Power Stage Design

Fig. 6 shows the hardware prototype designed as the platform for testing the ML-based MAB power flow control. The system comprises a 6-port MAB power stage, a TI DSP 28379D microcontroller for generating the phase-shifted PWM signals, an ATmega 328 AI accelerator integrated in an Arduino Nano board for inferring the machine learning

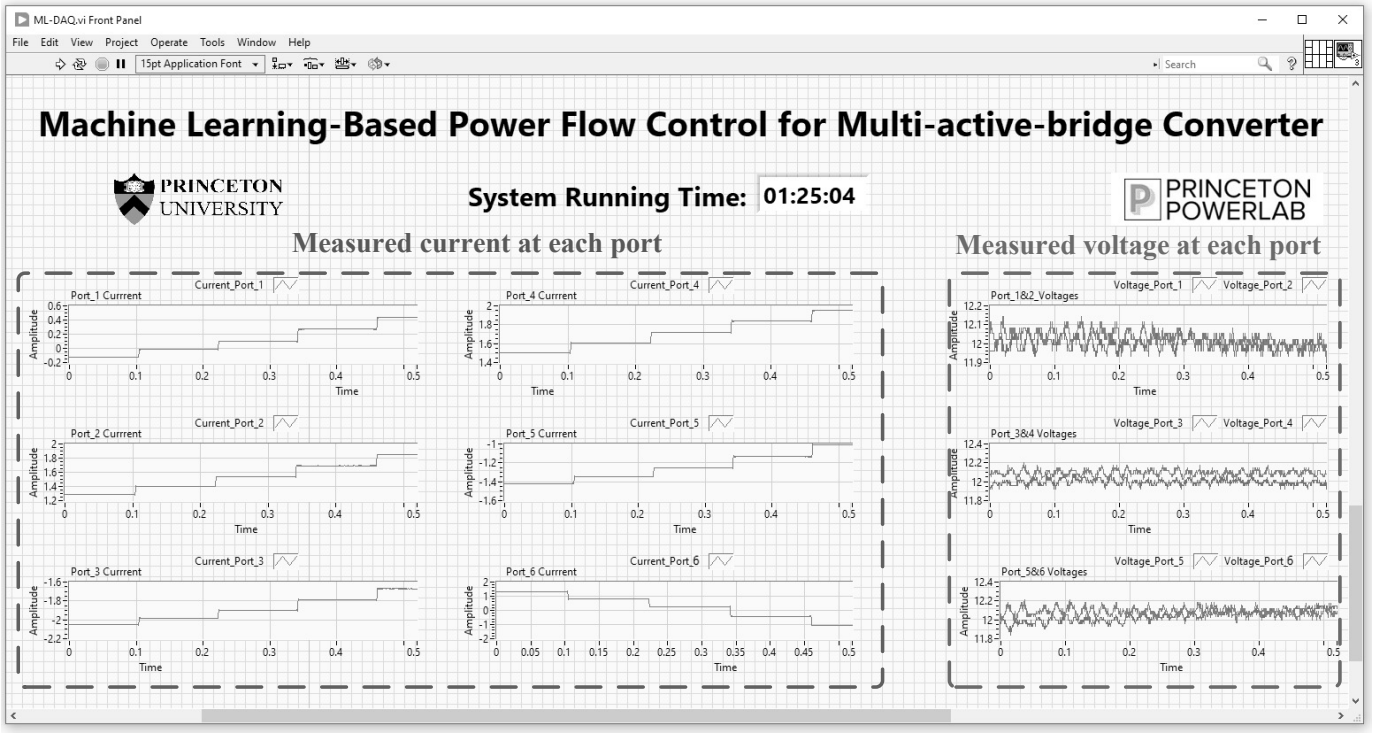


Fig. 8. Graphical user interface of the LabVIEW platform for power measurement. The voltages of all ports are maintained at 12 V, and the current of all ports are dynamically controlled depending on the needs. The system refreshes the power flow at 10 Hz and record the data for neural network training.

model, and a National Instrument LabVIEW platform for data acquisition. The AI accelerator contains a neural network module which is compatible with the TensorFlow platform. It can store and infer neural networks to predict the control output. In this paper, the neural network inference (ATmega 328) and the phase-shift modulation (TI DSP 28379D) are performed separately by two independent processors. One may implement a control strategy to combine the machine learning and the PWM modulation in a unified microcontroller platform, such as the TI C66x series. This paper also focuses on developing these algorithms and infrastructures for offline machine learning. Online learning and hybrid online/offline learning methods can also be applied for such control purposes, however, these methods will require hardware with greater computational capability.

The 3D assembly view of the 6-winding planar transformer is shown in Fig. 7. The transformer comprises two ER magnetic cores and two stacked printed circuit boards (PCBs). The 4-layer main power board hosts the PCB windings for four ports and the 2-layer bottom cover board hosts the PCB windings for the rest two ports. Each port is connected to a single turn and all ports are designed to have the same maximum volt-per-second sharing the same core. The series inductance (external & leakage) of each port is 140 nH and the switching frequency is 500 kHz. The phase signals at each port can be set between $-\frac{1}{2}\pi$ and $\frac{1}{2}\pi$, providing monotonic relationship between phase and power. The phase resolution is 1.8° with 500 kHz switching frequency, determined by the 100 MHz clock frequency of the PWM module in the DSP. The predicted phases in experiments are rounded to the nearest integer multiple of 1.8° in accordance with the hardware phase resolution. Two hardware configurations are explored to examine the effectiveness of the ML-based control in different operating conditions. For the MAB converter operating with trapezoidal current waveforms, 16 μF dc-blocking capacitors are used. The rated voltage is 12 V and the rated current is 3 A at each port. For the MAB converter operating with piecewise-sinusoidal current waveforms, 1.33 μF dc-blocking capacitors are used. The rated voltage is 12 V and the rated current is 4 A at each port. In practice, the load can be a (P, V) load in which the load is a voltage source, or a (P, Q) load in which the load has a known impedance [31], [32]. All examples explored in this paper are (P, V) loads with a constant voltage source behavior (such as batteries). This paper also demonstrates phase-shift modulation for power flow control with a 50% fixed duty cycle and a 500 kHz fixed switching frequency. Neural network models for pulse-width modulation or variable frequency modulation [11], as well as for interfacing with current source loads can also be developed in a similar way.

C. Data Acquisition with LabVIEW Platform

We use a LabVIEW system to automatically collect a large amount of power data. Fig. 8 shows the software interface of the LabVIEW system. The real-time dc voltage at each port and the dc current through each port are listed, together with the total running time of the acquisition process. The visualized data streams can help to monitor the operating status of the circuit and cease the data collection process after all tasks are completed. During the data acquisition process, each

Algorithm 1 neural network training

Input:
Phase data set from pre-definition Φ_n;
Power data set from LabVIEW platform P_n;
Output:
Tensorflow Lite model T_{nn};
1: Match the order of the phase data Φ_n with the order of the power data P_n;
2: Split the phase-power data set (P_n, Φ_n) into a training set (P_{train}, Φ_{train}) and a test set (P_{test}, Φ_{test});
3: Shuffle and Standardize the training set (P_{train}, Φ_{train}) and the test set (P_{test}, Φ_{test});
4: Define the neural network (NN) structure in TensorFlow;
5: Select optimizer ($Adam$), activation function ($sigmoid$) and loss function (MSE) for NN training;
6: Set initial learning rate (0.1), learning rate schedule (decay by 0.7 per 100 epochs) and batch size (64) for NN training;
7: Train the NN with the training set (P_{train}, Φ_{train});
8: Test the NN with the test set (P_{test}, Φ_{test});
9: Convert the NN into TensorFlow Lite model T_{nn};
10: **return** T_{nn};

Algorithm 2 neural network implementation

Input:
Power demand vector P_0;
Neural network in TensorFlow Lite format T_{nn};
Output:
Phase-shift vector for the MAB converter Φ_{actual};
1: Standardize the power demand P_0;
2: Call the neural network model T_{nn};
3: Infer T_{nn} to predict a phase-shift vector Φ_0 from P_0;
4: Generate the actual phase shifts Φ_{actual} from Φ_0 with the limited phase resolution (1.8°);
5: Feed Φ_{actual} into the MAB converter;
6: **return** Φ_{actual};

state generated by the corresponding phase shifts is applied for 100 ms, providing enough time for the MAB converter to reach the steady state. The voltage and current values of each state are extracted to form the power dataset and paired with the corresponding phase values provided by the microcontroller.

D. FNN Training and Inference

The rapid development of machine learning frameworks such as PyTorch and Tensorflow, together with numerous open-source examples available online have greatly accelerated the process of developing neural network models. For modelling power flow, the feedforward neural network is trained on the Tensorflow platform and converted to the Tensorflow Lite format which can link the software and hardware system. The model can then be encoded and downloaded into a neural network accelerator for phase prediction. When the AI accelerator receives the power demand, the neural network module stored in the accelerator is activated and infers the model to calculate the corresponding phases. Algorithm 1 shows the pseudo code for the network training implemented in TensorFlow. Offline training is adopted for this control application since the circuit parameters remain fixed for short term operation, disregarding component aging and other long term effects.

E. Implementation of Feedforward Power Flow Control

When a particular power flow is required, either by the system command or by the requirement from the feedback loop, the AI accelerator on the ATmega 328 board is activated. The accelerator sets the required power flow as input and infers the neural network model to obtain the corresponding phase shifts. The information of the phase shifts is sent from the accelerator to the DSP through I^2C communication protocol. The DSP modulates the phase-shifted PWM signals to its limited phase resolution (1.8° in this design), and generates the desired power flow. Algorithm 2 shows the pseudo code for FNN inference in the microcontroller implemented in C. Since the phase resolution of our controller implementation is 1.8°, a phase prediction error of less than 1.8°is needed for a majority of the operating points and is our control target.

V. Experimental Setup

Two experimental setups were selected to demonstrate the advantages of ML-based control: (1) when the series capacitance is large (16 μF), the MAB converter operates in the trapezoidal mode with trapezoidal current waveforms. This experiment is used to compare the performance of ML-based control against the Newton-Raphson method used in power systems analysis [12], [28]; (2) when the series capacitance is small (1.33 μF), the MAB converter operates in quasi-resonant mode with piecewise-sinusoidal current waveforms. This experiment is used to demonstrate the effectiveness of the ML-based control method in capturing the non-linear system characteristics when explicit power flow equations do not exist.

A. Trapezoidal Operation Mode

Fig. 11 shows a pair of example voltage and current waveforms for port #1 when the MAB converter operates in trapezoidal mode. The following three datasets were collected for this mode:

- **Dataset I with phase-input power-output:** Port #1 is assumed as the reference port with its phase as 0°. The phases of ports #2 to #6 are swept through the following 9 steps: [-21.6°, -16.2°, -10.8°, -5.4°, 0°, 5.4°, 10.8°, 16.2°, 21.6°]. The resulting power, ranging from -36 W to 36 W, is recorded for each port. A total number of $9^5 = 59,049$ data points (phase-power pairs) are collected by the LabView system.
- **Dataset II with phase-input power-output:** Port #1 is assumed as the reference port with its phase as 0°. The phases of port #2 to #6 are swept through the following 7 steps [-21.6°, -14.4°, -7.2°, 0°, 7.2°, 14.4°, 21.6°]. The resulting power, ranging from -36 W to 36 W, is recorded for each port. A total of $7^5 = 16,807$ data points are collected.
- **Dataset III with power-input phase-output:** $10,000$ targeted power vectors are randomly created for the six ports, ranging from -36 W to 36 W, as the input of this neural

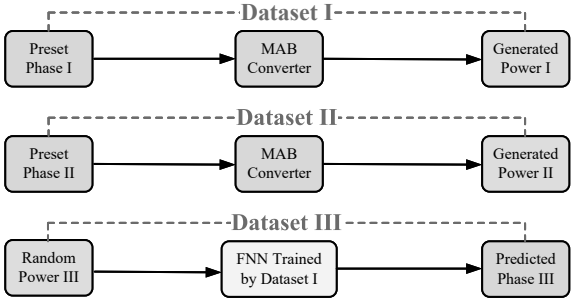


Fig. 9. Block diagrams of the data acquisition process. Three types of datasets are collected for different purposes.

network. Dataset I is used to train a neural network with power as the input and phase as the output. This neural network predicts the needed phase vectors to generate the targeted power flow. The predicted phases are fed into the hardware system. Finally, the output power are measured in the experimental setup.

The collected phase and power data sets are combined and utilized in TensorFlow to train the FNN. The FNN comprises one input layer, one output layer and one hidden layer. The input layer has six neurons, representing the power flowing into six ports. The output layer has five neurons, representing the phases of five ports (excluding the first port as it is the phase shift reference). The FNN was encoded in the AI accelerator to infer the phases based on the targeted power flow. The predicted phases are converted to actual phases by the pulse-width-modulation (PWM) module in the TI DSP. The measured and targeted power flow were compared to validate the effectiveness of the ML-based power flow control strategy. Different numbers of neurons were used in the hidden layer and their performance were compared. The batch size was 128. Each training run had 500 epochs. The initial learning rate was set as 0.01, decaying by 70% every 100 epochs. The tests were repeated multiple times with data shuffling.

The following four experiments were performed:

• **Experiment A: Training with 85% of Dataset I; Testing with 15% of Dataset I.** The purpose of this experiment is to evaluate the capability of the NN model in operating under scenarios similar to those from the training data.

• **Experiment B: Training with 100% of Dataset I; Testing with 100% of Dataset II.** The purpose of this experiment is to evaluate the capability of the NN model in operating under scenarios different from the training data. Note that Dataset I and Dataset II are collected with different sets of phase angles.

• **Experiment C: Training with 100% of Dataset I; Testing with 100% of Dataset III.** The purpose of this experiment is to evaluate the capability of a FNN in a real application - solving for the phases based on the targeted power. Dataset I is collected with pre-selected phase angles. Dataset III is collected with calculated phase angles for targeted power.

• **Experiment D: Measured Power Flow based on Phases Predicted by NR-methods.** The purpose of this experiment is to evaluate the effectiveness of the NR-method for MIMO power flow control with targeted phase and calculated phase for Experiment A and B, and targeted power and measured

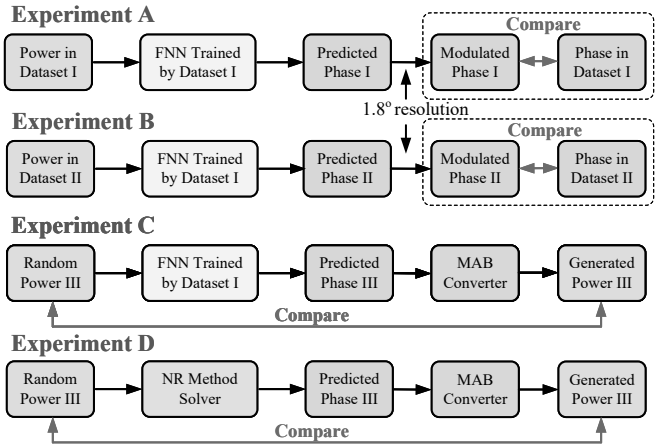


Fig. 10. Block diagrams of the experimental process. Different experiments and different datasets are used to verify different hypotheses.

power for Experiment C. With an error target of 10 mW, it usually takes three iterative steps for the algorithm to converge. Each iterative step includes a Jacobian matrix inversion [12].

Table I summarizes the experimental results with different number of neurons in the hidden layers, and compares the results of the ML method and the NR method. The ML model's performance in Experiment A and Experiment B are evaluated by means of the average absolute phase mismatch per port between the measured and predicted phases. Even a small phase mismatch may lead to a significant power mismatch. The performance of the model in Experiment C is evaluated based on the average absolute power mismatch per port between the targeted power and the measured power, normalized to the maximum power rating of each port as a percentage value. In the worst case scenario, the power error can reach $\pm 10\%$, but they appear for very few data points, while the error for most of the data points is very low. The average power mismatch for the Neural network model, as reported in Table I, is usually below 3%.

B. Quasi-Resonant Operation Mode

Fig. 14 shows the voltage and current waveforms of port #1 when the MAB converter operates in quasi-resonant mode. The following three datasets were collected for this mode:

• **Dataset I with phase-input power-output:** Port #1 is assumed as the reference port with its phase as 0°. The phases of ports #2-6 are swept through the following 9 steps: [-21.6°, -16.2°, -10.8°, -5.4°, 0°, 5.4°, 10.8°, 16.2°, 21.6°]. The resulting power, ranging from -48 W to 48 W, is recorded for each port. A total number of $9^5 = 59,049$ data points are collected.

• **Dataset II with phase-input power-output:** Port #1 is assumed as the reference port with its phase as 0°. The phases of port #2-6 are swept through the following 7 steps: [-21.6°, -14.4°, -7.2°, 0°, 7.2°, 14.4°, 21.6°]. The resulting power, ranging from -48 W to 48 W, is recorded for each port. A total number of $7^5 = 16,807$ data points are collected.

• **Dataset III with power-input phase-output:** $10,000$ targeted power vectors are randomly created for the six ports, ranging from -48 W to 48 W, as the input of this neural

TABLE I
MISMATCH BETWEEN PREDICTED AND TARGETED CONTROL VARIABLES WITH MACHINE LEARNING (ML) AND NEWTON-RALPHSON (NR) METHODS WHEN THE MAB CONVERTER OPERATES WITH TRAPEZOIDAL CURRENT WAVEFORM

Method	ML Methods (Experiment A, B, C)					NR Method (Experiment D)
Number of Neurons in the Hidden Layer	4	5	10	30	50	N/A
Total Number of Parameters in the Neural Network	53	65	125	365	605	N/A
Experiment A: Avg. Abs. Phase Mismatch [°]	3.295°	0.208°	0.033°	0.006°	0.003°	0.853°
Experiment B: Avg. Abs. Phase Mismatch [°]	4.271°	0.245°	0.032°	0.007°	0.005°	0.884°
Experiment C: Avg. Abs. Power Percentage Mismatch [%]	N/A	2.895%	2.254%	2.319%	2.332%	5.230%

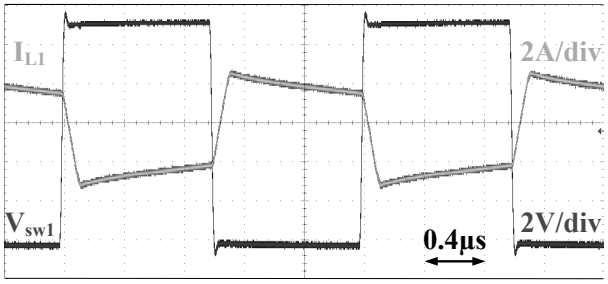


Fig. 11. Example voltage and current waveforms of port #1 when the MAB converter operates in the *trapezoidal mode* with trapezoidal current. The port-to-port operation is similar to a dual-active-bridge converter.

network. Dataset I was used to train a neural network with power-input and phase-output. This neural network predicts the needed phase vectors to generate the targeted power flow as a part of the control loop. The phases are fed into the power stage and the power-output vectors are measured. The following three experiments were performed with these datasets:

• **Experiment A: Training with 85% of Dataset I; Testing with 15% of Dataset II.** The purpose of this experiment is to evaluate the capability of the NN model in operating under scenarios similar to the training data.

• **Experiment B: Training with 100% of Dataset I; Testing with 100% of Dataset II.** The purpose of this experiment is to evaluate the capability of the NN model in operating under scenarios different from the training data. Note that Dataset I and Dataset II are collected with different sets of pre-selected phase angles.

• **Experiment C: Training with 100% of Dataset I; Testing with 100% of Dataset III.** The purpose of this experiment is to evaluate the capability of a FNN in performing under real application scenarios - solving for the phases based on the targeted power flow. Dataset I is collected with pre-selected phase angles. Dataset III is collected with back-calculated phase angles.

We find that the ML methods are capable of predicting the phase angles for the MAB converter, even in the quasi-resonant mode of operation. In comparison, the NR method cannot capture the power flow for MAB converters operating in resonant modes because there are no explicit equations for power calculation. Machine learning methods do not rely on precise lumped circuit models or power flow equations to predict the required phase-shift, and thus is applicable to MAB converters operating in quasi-resonant mode without explicit power flow models.

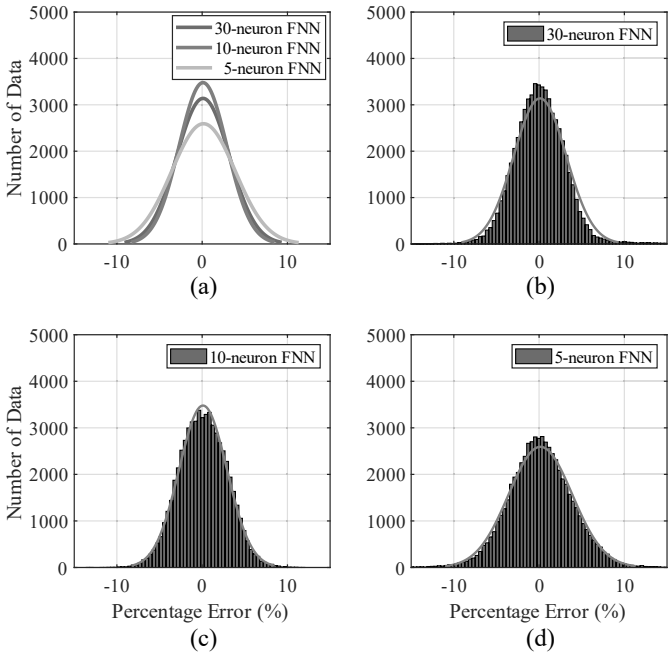


Fig. 12. Error distribution of Experiment C when the MAB converter operates with trapezoidal current: (a) Error distribution curves for 30, 10 and 5-neuron FNNs; (b)-(d): Error histograms for 30, 10 and 5-neuron FNNs. The average power percentage error per port is usually below 3%.

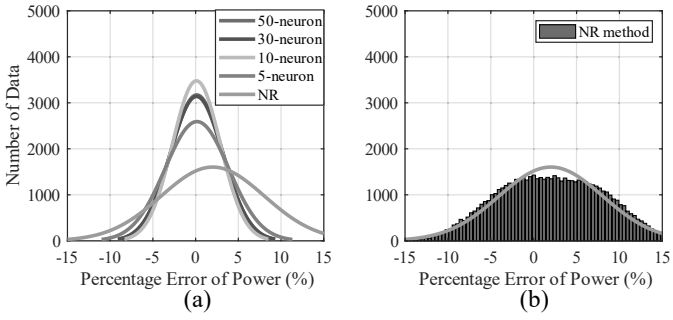


Fig. 13. Results of Experiment D in trapezoidal mode: (a) Error distribution curves for FNN with 50, 30, 10, 5 neurons & the NR method; (b) Error histogram of the NR method.

The performance of the ML-based power flow control methods are evaluated by sweeping the number of neurons in the hidden layers and statistically comparing the model

TABLE II
MISMATCH BETWEEN PREDICTED AND TARGETED RESULTS WITH DIFFERENT SIZES OF THE NEURAL NETWORK WHEN THE MAB CONVERTER OPERATES WITH QUASI-RESONANT CURRENT WAVEFORM.

Number of Neurons in the Hidden Layer	4	5	10	20	30	50
Total Number of Parameters in the Neural Network	53	65	125	245	365	605
Experiment A: Avg. Abs. Phase Mismatch [°]	4.330°	0.238°	0.099°	0.047°	0.013°	0.007°
Experiment B: Avg. Abs. Phase Mismatch [°]	4.576°	0.267°	0.100°	0.050°	0.049°	0.033°
Experiment C: Avg. Abs. Power Percentage Mismatch [%]	N/A	3.381%	2.981%	3.008%	3.302%	4.101%

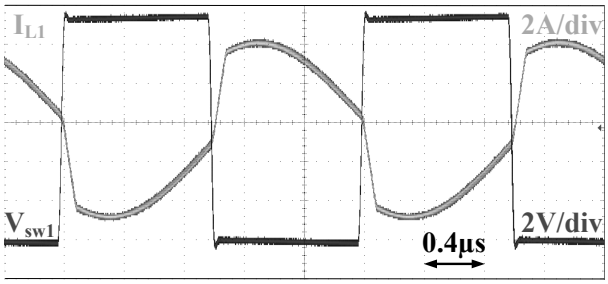


Fig. 14. Example voltage and current waveforms of port #1 when the MAB converter operates in the *quasi-resonant mode* with piecewise sinusoidal current. The port-to-port operation is similar to a series-resonant converter.

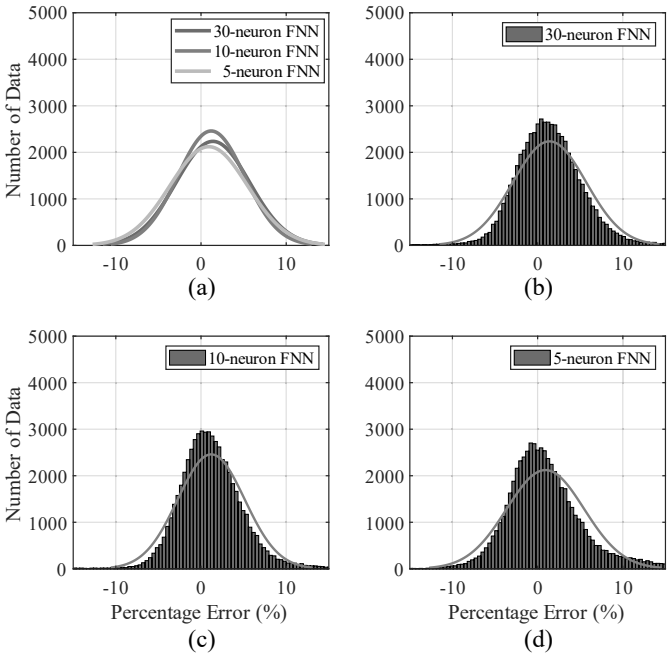


Fig. 15. Error distribution in Experiment C under quasi-resonant situation: (a) Error distribution curves for 30, 10 & 5-neuron FNNs; (b)-(d): Error histograms for 30, 10 & 5-neuron FNNs. The average power percentage error per port is usually below 5%.

accuracy. MSE is used as the loss function for neural network training, aiming to inflict more penalty on data with larger error and keeping the deviation in phase error values small. Average absolute phase mismatch (the difference between the desired phase and predicted phase), and absolute power percentage mismatch (the percentage difference between the desired power and achieved power) are used as the figures-of-merit for performance comparison. 95^{th} percentile error rate is also used as a figure-of-merit to evaluate the error in the section of transfer learning.

VI. NEURAL NETWORK PERFORMANCE EVALUATION

A. Trapezoidal Operation Mode

Table I summarizes the results of the three experiments. The number of parameters in the NN increases as the number of neurons in the hidden layer increases. In Experiment A & B, the prediction error decreases as the network complexity increases and finally reaches saturation. Insufficient number of neurons may lead to underfitting and hence, large prediction error. The phases' mean absolute average error in Experiment A & B are similar, indicating that FNN can be applied to predict operating conditions that are not fully covered by the training data. In Experiment C, the average power error generated by the FNN-based phases is below 3%, verifying the effectiveness of the ML-based control on the test dataset. Fig. 12 shows the power error distributions of FNN in Experiment C. The deviation of ML-based control is relatively small. Fig. 13 shows the comparison of power error distributions between FNNs (Experiment C) and NR method (Experiment D). In the selected examples, the deviation of FNN is smaller than that of NR method. The error for the NR method is distributed in the range between -10° to +10°. Dc biases are observed in the NR method due to non-idealities in the lumped circuit parameters. Fig. 16 (a) shows the power flow at 6 ports for 16 different operating conditions in trapezoidal mode, and each condition is active for 60 ms. It compares the desired and measured power flow as a time series when a 5 neurons FNN was used to predict the desired phase. Power flow controlled by ML usually matches better with the target power than the power flow controlled by a NR-based algorithm. Fig. 16 (b) shows one example power flow step change at 6 ports. It takes about 1.2 ms for the system to transit from one power flow operating condition to another.

B. Quasi-Resonant Operation Mode

Table II summarizes the results of the three experiments conducted under the quasi-resonant mode of operation. Again, we observe that the error decreases as the neural network size increases. The prediction error of the ML algorithm in the quasi-resonant mode is larger than that of trapezoidal mode, partly due to the more complicated MIMO power flow. However, with a large enough NN, the average phase

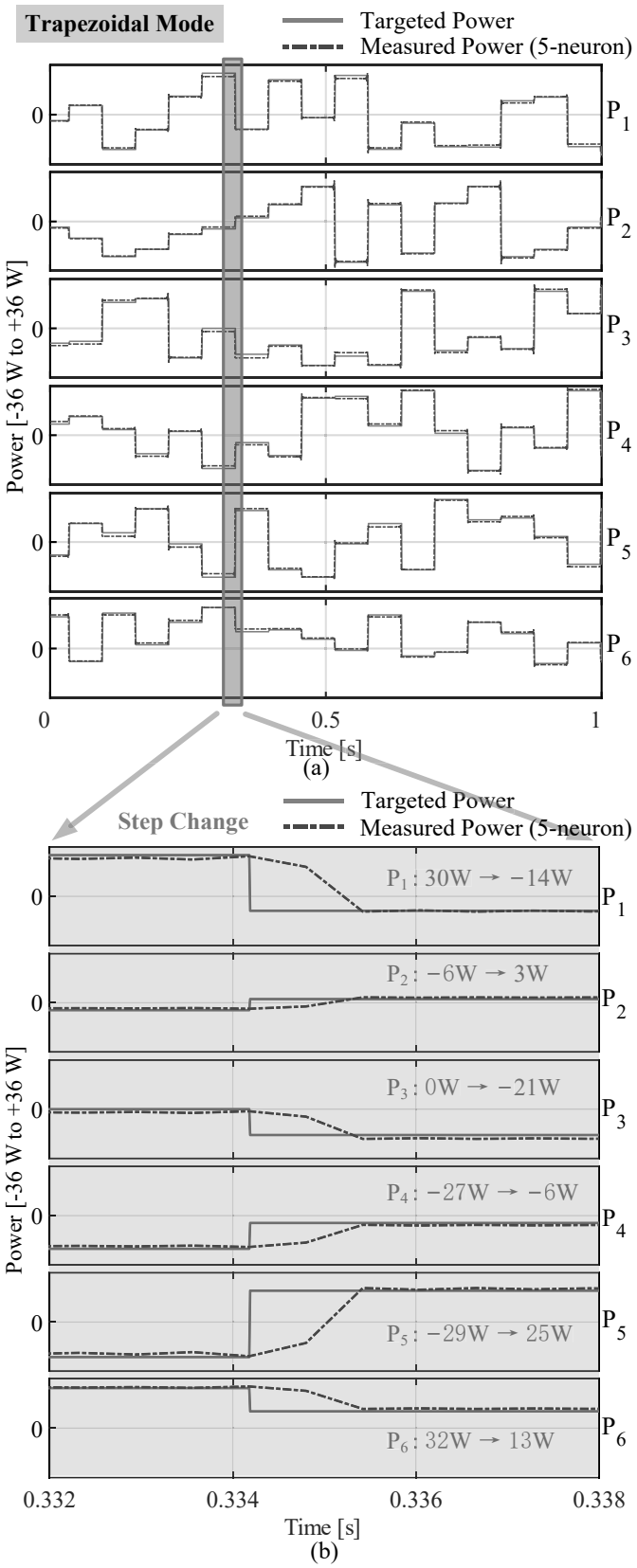


Fig. 16. Trapezoidal Mode Operation: (a) Targeted and measured power across the six ports (P_1-P_6) using a 5-neuron FNN. The power flow is updated every 60 ms; (b) Power step change across the six ports. The transient time of the power flow is about 1.2 ms.

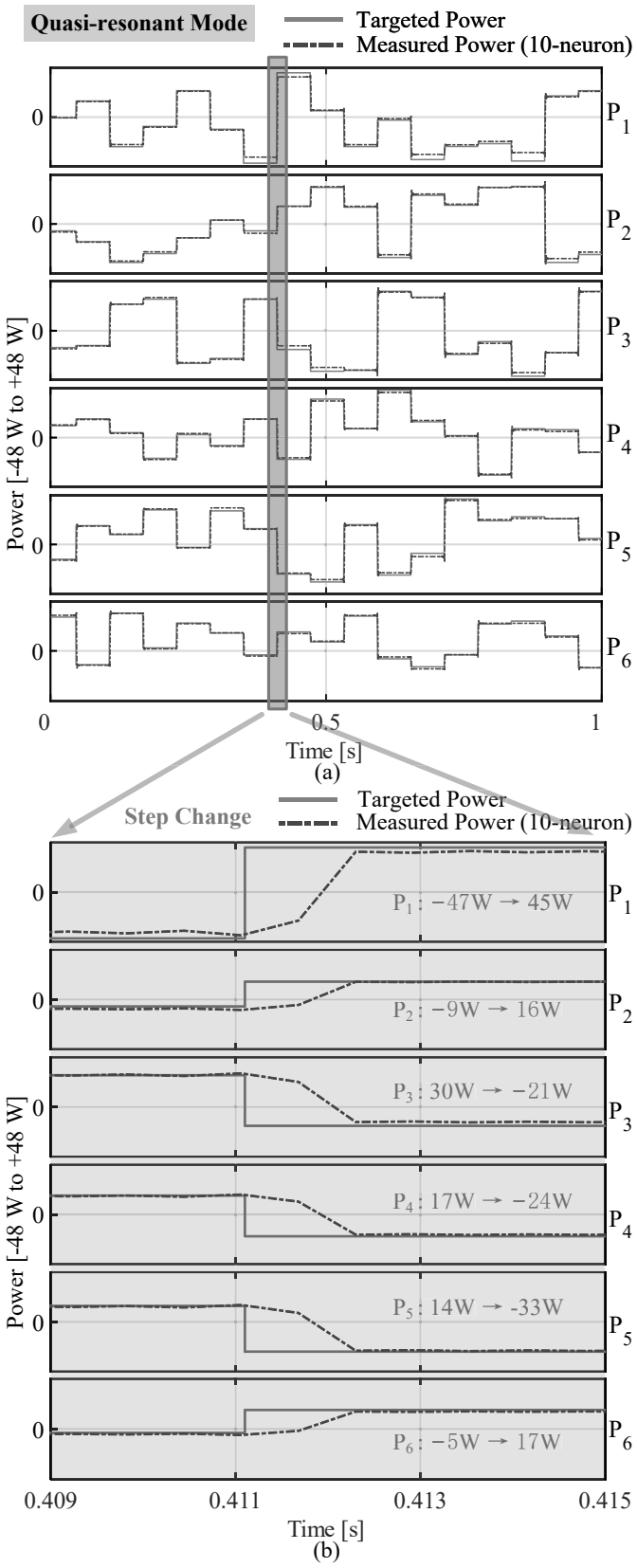


Fig. 17. Quasi-resonant Mode Operation: (a) Targeted and measured power across the six ports (P_1-P_6) using a 10-neuron FNN. The power flow is updated every 60 ms; (b) Power step change across the six ports. The transient time of the power flow is about 1.2 ms.

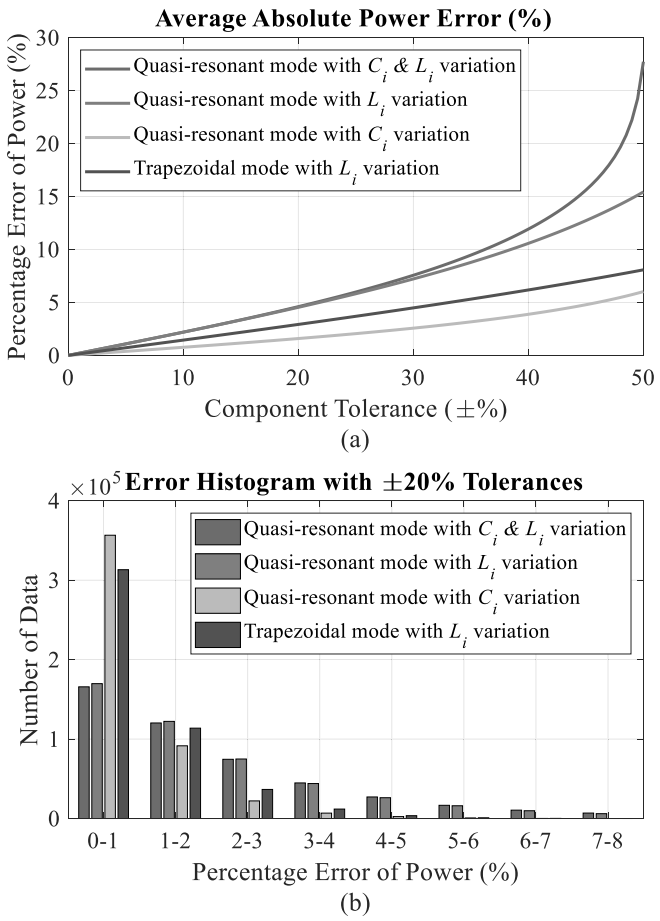


Fig. 18. Monte Carlo simulation to investigate the impact of component tolerances: (a) Average absolute % of power error in both trapezoidal and quasi-resonant modes with capacitance and inductance variations; (b) Power error distribution with $\pm 20\%$ inductance tolerances in four cases.

mismatch can be lowered, as it can be seen that the average absolute power mismatch per port of a 10-neuron FNN is below 3%. Fig. 17 (a) shows the power flow at 6 ports for 16 different operating conditions in quasi-resonant mode, and each condition is active for 60 ms. It compares the desired and measured power flow as a time series when a 10 neurons FNN was used to predict the desired phase. Fig. 17 (b) shows one example power flow step change at 6 ports. It takes about 1.2 ms for the system to transit from one power flow operating condition to another.

C. Impact of Component Tolerances

We performed Monte Carlo simulation to investigate the impact of component tolerances on feedforward model. The component values (inductance and capacitance) are assumed to follow a normal distribution - the average of the normal distribution is the desired component value and the tolerance defines the boundary within which 95% of component values lie. We randomly generate 400 component values following the normal distribution and use the theoretical models to evaluate the power flow. The 400 component values are combined with 400 random pairs of phase shifts (6 per pair), leading to a total of 960,000 data points for the power flow. The generated power is calculated and estimated from Eq. (3) for trapezoidal mode and Eq. (8) for quasi-resonant mode.

Fig. 18 (a) shows the average absolute power error with the tolerance swept from 0% to $\pm 50\%$. Fig. 18 (b) shows the power error histogram with $\pm 20\%$ component tolerances. In the trapezoidal mode, only the series inductance L_i has impact on the power flow. When the series inductance tolerance increases from $\pm 5\%$ to $\pm 50\%$, the absolute average of the power error increases from 0.71% to 8.1%, respectively. In the quasi-resonant mode, power error is analyzed for three cases: (i) only the tolerance of the series inductance L_i is increasing; (ii) only the tolerance of the series capacitance C_i is increasing; (iii) the tolerances of both the series inductance L_i and the series capacitance C_i are increasing. The worst case happens when both C_i and L_i have $\pm 50\%$ tolerance, leading to about 30% of average absolute power error. The results also reveal that the power flow of this particular design is more sensitive to inductance variation than capacitance variation.

VII. Reduce Data Size with Transfer Learning

Transfer learning is a ML technique where a model developed for one task is reused as the starting point to initiate the model for a new task as long as they share similar features [22]. This technique can greatly reduce the amount of data needed to train the model for the new task. Fig. 19 illustrates the principle of the transfer learning method used for the MAB converter. Transfer learning is applied to control the power flow in a MAB converter by pre-training the neural network with data obtained through theoretical calculations (in trapezoidal mode), then fine-tuning the model with data obtained through experimental measurements to include the effects of non-idealities and component variations. Since models formed by ideal and experimental data coherently share similar features, the knowledge of traditional modeling can be reused and transferred from one scenario to another. The accuracy of the neural network obtained through transfer learning is compared with the accuracy of a neural network obtained through the traditional method with a large amount of data.

A. Data Collection for Training

1) Trapezoidal Operation Mode: Eq. (3) is used to produce numerous ideal power-phase data with predetermined phase information. This training data is not precise due to the unavoidable mismatch between models and physical characteristics, leakage inductances, and parasitics. The physical behaviors of the multiwinding magnetics cannot be well captured by lumped circuit models. The following three datasets are collected for training and testing the NN in trapezoidal operation mode.

• **Dataset 1 from equation-based calculations for training:** Port #1 is assumed as the reference port with its phase as 0°. The phases of ports #2 to #6 are swept through the following 9 steps: [-21.6°, -16.2°, -10.8°, -5.4°, 0°, 5.4°, 10.8°, 16.2°, 21.6°]. The resulting power, ranging from -36 W to 36 W, is calculated for each port. A total number of $9^5 = 59,049$ data points are collected.

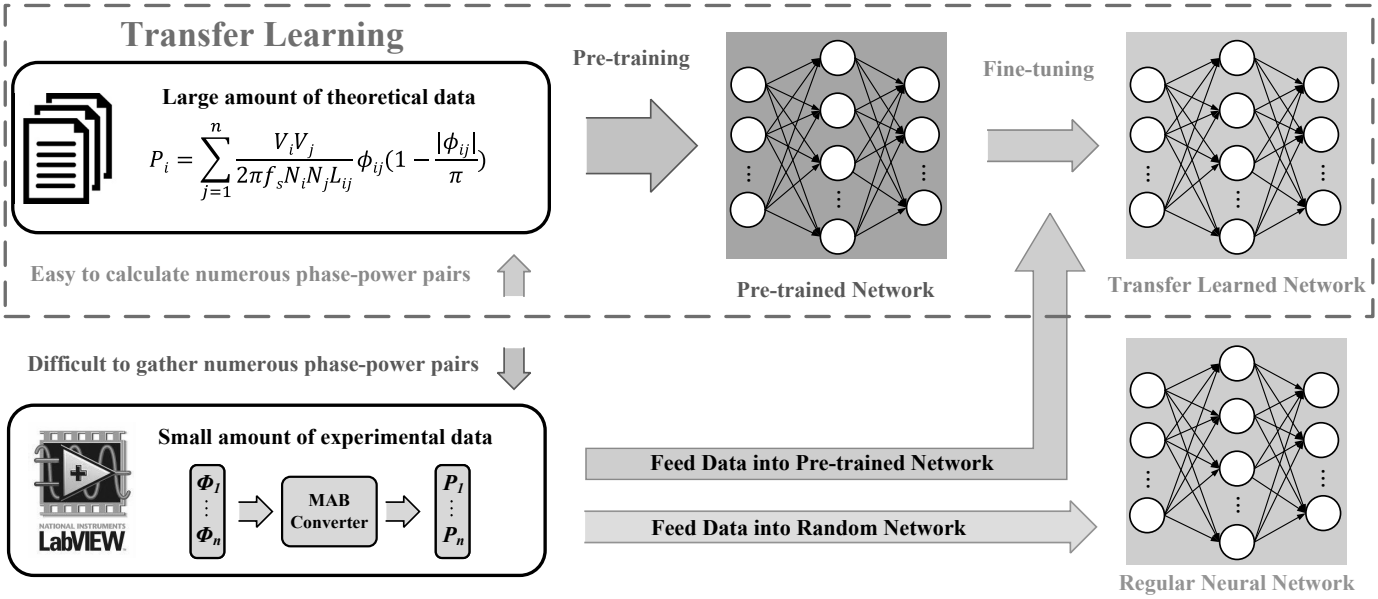


Fig. 19. Principles of the transfer learning method for a MAB converter modelling. The red and green paths show the process of the transfer learning method. The orange path shows the process of the traditional ML method. Data are collected from both equation models and experimental measurements.

• **Dataset 2 from experimental measurements for training:** Port #1 is assumed as the reference port with its phase as 0°. The phases of ports #2 to #6 are swept through the following 9 steps: [-21.6°, -16.2°, -10.8°, -5.4°, 0°, 5.4°, 10.8°, 16.2°, 21.6°]. The resulting power, ranging from -36 W to 36 W, is recorded for each port. A total number of $9^5 = 59,049$ data points are collected.

• **Dataset 3 from experimental measurements for testing:** Port #1 is assumed as the reference port with its phase as 0°. The phases of ports #2 to #6 are swept through the following 7 steps: [-21.6°, -14.4°, -7.2°, 0°, 7.2°, 14.4°, 21.6°]. The resulting power, ranging from -36 W to 36 W, is recorded. A total number of $7^5 = 16,807$ data points are collected.

2) Quasi-resonant Operation Mode: It is shown in [12] that Eq. (8) can be used to generate theoretical data of phase-power pairs for MAB converters operating with piecewise sinusoidal current, even in quasi-resonant mode. The mismatch between the predicted power and measured power is small enough and thus, can be use to pre-train the neural network model. The following three datasets are collected for training and testing the NN in quasi-resonant operation mode.

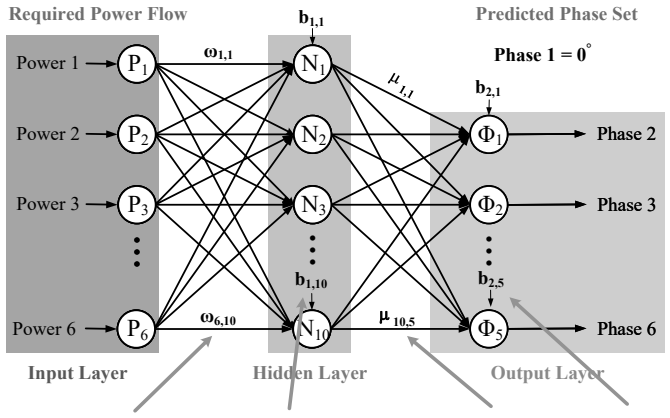
• **Dataset 4 from equation-based calculations for training:** Port #1 is assumed as the reference port with its phase as 0°. The phases of ports #2 to #6 are swept through the following 9 steps: [-21.6°, -16.2°, -10.8°, -5.4°, 0°, 5.4°, 10.8°, 16.2°, 21.6°]. The resulting power, ranging from -48 W to 48 W, is calculated for each port. A total number of $9^5 = 59,049$ data points are collected.

• **Dataset 5 from experimental measurements for training:** Port #1 is assumed as the reference port with its phase as 0°. The phases of ports #2 to #6 are swept through the following 9 steps: [-21.6°, -16.2°, -10.8°, -5.4°, 0°, 5.4°, 10.8°, 16.2°, 21.6°]. The resulting power, ranging from -48 W to 48 W, is recorded for each port. A total number of $9^5 = 59,049$ data points are collected.

• **Dataset 6 from experimental measurements for testing:** Port #1 is assumed as the reference port with its phase as 0°. The phases of ports #2 to #6 are swept through the following 7 steps: [-21.6°, -14.4°, -7.2°, 0°, 7.2°, 14.4°, 21.6°]. The resulting power, ranging from -48 W to 48 W, is recorded for each port. In total, $7^5 = 16,807$ data points are collected.

B. Neural Network Structure and Pre-training

The selected FNN comprises one hidden layer with 10 neurons as the testing platform for transfer learning. The FNN also has an input layer with 6 neurons representing the power of the 6 ports, and an output layer with 5 neurons representing the control phases of 5 ports with one port as the phase reference port (0°). MSE is used as the loss function and $Adam$ is used as the NN optimizer. During the pre-training process, Dataset 1 is used to train a "Pre-trained Network" as shown in Fig. 20. The initial learning rate is 0.5, decaying to 70% every 10 epochs. The batch size is 64 and the number of training epochs is 50. The number of training epochs for pre-training should be much smaller compared to re-training or regular training process because the pre-training process only needs to capture the main features of the training data while the details can be ignored. The pre-training process should be designed to keep the maximum generality for retraining purposes. With pre-training, the parameters (weights & biases) of the neural network are initialized with patterns learnt from the theoretical model. Without pre-training, these parameters are randomly initialized. Pretraining the model with theoretical data can greatly reduce the prediction error when limited experimental data is available.



$\omega_{1,m}$	$\omega_{2,m}$	$\omega_{3,m}$	$\omega_{4,m}$	$\omega_{5,m}$	$\omega_{6,m}$	$b_{1,m}$
1.51	-1.91	-0.18	-0.24	-0.12	-0.12	-0.12
-1.76	-0.16	1.67	-0.11	-0.10	-0.09	-0.09
-2.49	1.43	-0.71	-0.57	-0.69	-0.57	-0.57
-1.26	0.41	0.40	0.45	2.14	0.44	0.44
2.08	0.44	0.28	0.23	-1.71	0.34	0.34
2.01	0.25	0.15	-1.70	0.19	0.26	0.26
1.90	0.08	-1.89	0.09	0.04	-0.10	-0.10
2.73	0.43	0.45	0.44	0.49	-1.92	-1.92
-2.41	-0.50	-0.43	-0.38	-0.24	1.84	1.84
-2.31	-0.31	-0.42	1.86	-0.52	-0.47	-0.47

$\mu_{m,1}$	$\mu_{m,2}$	$\mu_{m,3}$	$\mu_{m,4}$	$\mu_{m,5}$	$b_{2,n}$
4.85	0.09	-0.01	0.22	0.03	-0.09
0.04	-4.75	-0.07	0.06	0.04	-0.57
-4.30	-0.08	0.20	-0.02	0.04	0.44
-0.13	-0.14	0.00	-5.04	-0.01	0.34
0.14	-0.12	0.20	4.27	0.19	0.26
-0.13	0.02	4.82	-0.01	0.07	
0.03	4.72	-0.10	0.07	-0.01	
-0.12	-0.14	-0.08	-0.03	4.27	
0.03	0.18	-0.09	0.10	-4.44	
0.33	-0.07	-4.12	-0.06	0.10	

$1 \leqslant m \leqslant 10, 1 \leqslant n \leqslant 5$

Fig. 20. Parameter visualization of a pre-trained FNN model in trapezoidal operation mode with weight (ω, μ) and bias (b). Weight ω between input layer and hidden layer has $6 \times 10 = 60$ parameters. Weight μ between input layer and hidden layer has $10 \times 5 = 50$ parameters. Bias b in the hidden layer has 10 parameters, while bias b in the output layer has 5 parameters. A large percentage of the weight parameters are negligible and ineffective for phase inference. Techniques such as network pruning [33] can be applied to further reduce the size of the model.

C. Re-training and Experimental Results

When the pre-trained model is directly applied to the testing set with experimental data, the prediction error comes from the mismatch between the theoretical models and the actual physical model, due to non-idealities and non-linearities. Re-training the pre-trained model with a small amount of experimental data can fix the mismatch. Fine-tuning the pre-trained model with a small dataset can yield a comparable accuracy to the case when a much larger dataset is used. In comparison, if the model is initialized without any pre-trained information and trained with the same amount of experimental data, the accuracy is usually much worse than the transfer-learned model.

The following experiments are designed to verify the effectiveness of transfer learning. Experiment A&B is for trapezoidal mode while Experiment C&D is for quasi-resonant mode. Mean absolute error is used to compare the differences between transfer learning and traditional learning. The 95^{th} percentile error is also used as a figure-of-merit. Over 95% of the testing results fall within the 95^{th} percentile error range.

• **Experiment A:** Pre-training with 100% of Dataset 1; Re-training with variable data points of Dataset 2; Testing with 100% of Dataset 3. The amount of data used for retraining is swept from 1 to 3000 in ten steps. The purpose of this experiment is to evaluate the performance and efficiency of transfer learning as the size of the retraining dataset increases.

• **Experiment B:** No pre-training; Training with a variable data points in Dataset 2; Testing with 100% of Dataset 3. The size of the training data is swept from 1 to 3000 in ten steps. The purpose of this experiment is to evaluate the performance and efficiency of a traditional machine learning process as the size of the training dataset increases.

Fig. 21 (a) and (b) show the mean absolute error and the 95^{th} percentile error of phase predictions in trapezoidal mode. The experiment for each number of retraining data size is repeated 10 times with data shuffling. When there is only one experimental data point, the 95^{th} percentile error of the transfer learning model is $3.0°$, and is much better than the result obtained from a model with randomly initiated parameters ($28.2°$). With 100 retraining data points, the 95^{th} percentile error of the transfer learning model is reduced to $1.0°$, while it takes over 1,000 data points for a randomly initiated model to reach the same level of accuracy.

• **Experiment C:** Pre-training with 100% of Dataset 4; Re-training with variable data points of Dataset 5; Testing with 100% of Dataset 6. The size of the retraining dataset is swept from 1 to 3000 in ten steps. The purpose of this experiment is to evaluate the performance and efficiency of the transfer learning method as the size of the retraining dataset increases.

• **Experiment D:** No pre-training; Training with a variable data size in Dataset 5; Testing with 100% of Dataset 6. The size of the training dataset is swept from 1 to 3000 in ten steps. The purpose of this experiment is to evaluate the performance and efficiency of a traditional machine learning process as the size of the training dataset increases.

Fig. 21 (c) and (d) show the mean absolute error and the 95^{th} percentile error of phase predictions when the MAB converter operates in quasi-resonant mode. The experiment for each subset of the experimental data is repeated 10 times with data shuffling. When there is only one experimental data point, the 95^{th} percentile error of the transfer learned model ($3.0°$) is more than 10 times lower than that of the randomly initiated model ($31.6°$). It only takes 200 retraining data points for the 95^{th} percentile error of the transfer learned model to reach $1.3°$. It takes over 1,000 data points for the traditional learned model to reach the same level of accuracy.

Machine learning methods are data-driven. The performance of machine learning methods highly depends on the size and quality of the database. In application scenarios when the sizes of the datasets are limited, transfer learning is a powerful and an efficient way to reduce the size of the experimental data needed to train a neural network model to achieve high enough accuracy for controlling the power flow of a MAB converter.

VIII. Conclusions

This paper presents the methodology and hardware implementation of a machine learning based power flow control of MAB converters. A 6-port MAB converter with 48 W of maximum power per port and 500 kHz switching frequency was used a platform to validate the concept. The ML algorithms are implemented in hardware as a part of the power flow controller. The MAB converter was tested in two operation

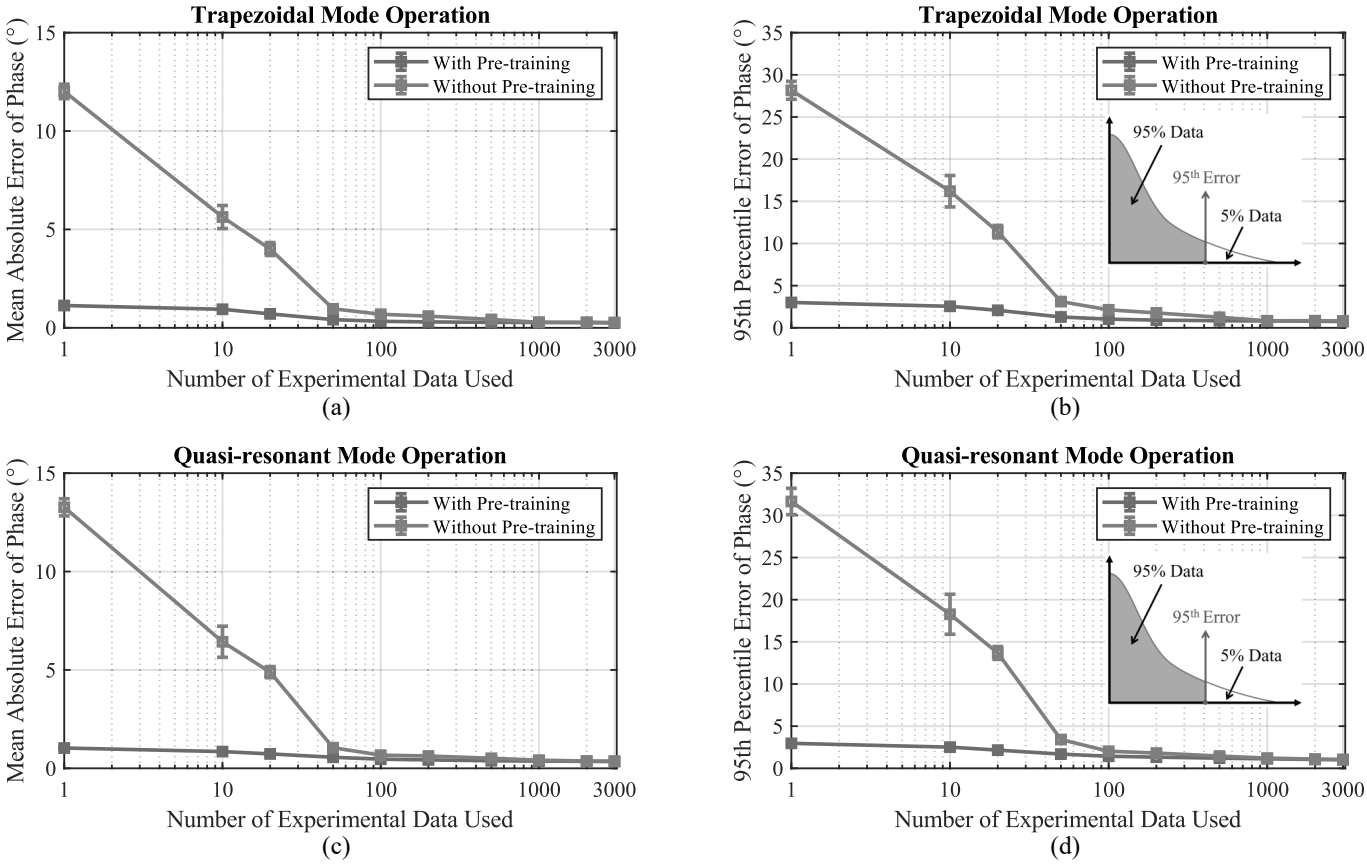


Fig. 21. Phase error under different conditions: (a) the mean absolute error of phase in trapezoidal mode; (b) the 95^{th} percentile error of phase in trapezoidal mode; (c) the mean absolute error of phase in quasi-resonant mode; (d) the 95^{th} percentile error of phase in quasi-resonant mode.

modes: trapezoidal mode and quasi-resonant mode. In trapezoidal mode with trapezoidal current, the performance of the ML-based approach is comparable to that of a traditional NR method and can achieve high accuracy. In quasi-resonant mode with piecewise-resonant current, the traditional NR method is not applicable, while the ML methods can still maintain high accuracy through a model-free learning approach. Transfer learning was used to reduce the amount of experimental data needed for a neural network to achieve high accuracy. It is shown that machine learning methods can overcome a few fundamental challenges of controlling the power flow in MAB converters and can be greatly enhanced by transfer learning.

IX. Acknowledgements

The authors would like to thank Mr. Edward Knapp, Mr. Andrew Pigney, and Mr. Anthony Noble of American Tower Corporation for their insights and discussions.

References

[1] M. Liao, H. Li, P. Wang, Y. Chen, and M. Chen, "Machine learning methods for power flow control of Multi-Active-Bridge Converters," *IEEE Workshop on Control and Model. of Power Electron. (COMPEL)*, Nov. 2021.

[2] A. K. Bhattacharjee, N. Kutkut and I. Batarseh, "Review of Multiport Converters for Solar and Energy Storage Integration," in *IEEE Trans. Power Electron.*, vol. 34, no. 2, pp. 1431-1445, Feb. 2019.

[3] M. Evzelman, M. Rehman, K. Hathaway, R. Zane, D. Costinett, and D. Maksimovic, "Active Balancing System for Electric Vehicles With Incorporated Low-Voltage Bus," *IEEE Trans. Power Electron.*, vol. 31, no. 11, pp. 7887-7895, Nov. 2016.

[4] P. Wang, Y. Chen, J. Yuan, R. C. Pilawa-Podgurski, and M. Chen, "Differential Power Processing for Ultra-Efficient Data Storage," *IEEE Trans. Power Electron.*, vol. 36, no. 4, pp. 4269-4286, Apr. 2021.

[5] M. Liu, Y. Chen, Y. Elasser and M. Chen, "Dual Frequency Hierarchical Multilayer Battery Balancer Architecture," *IEEE Trans. Power Electron.*, vol. 36, no. 3, pp. 3099-3110, Mar. 2021.

[6] Y. Chen, P. Wang, Y. Elasser, and M. Chen, "Multicell Reconfigurable Multi-Input Multi-Output Energy Router Architecture," *IEEE Trans. Power Electron.*, vol. 35, no. 12, pp. 13210–13224, Dec. 2020.

[7] M. N. Kheraluwala, R. W. Gascoigne, D. M. Divan, and E. D. Baumann, "Performance Characterization of a High-Power Dual Active Bridge DC-to-DC Converter," *IEEE Trans. Ind. Appl.*, vol. 28, no. 6, pp. 1294–1301, Nov./Dec. 1992.

[8] R. W. De Doncker, D. M. Divan, and M. H. Kheraluwala, "A Three-phase Soft-Switched High Power Density DC/DC Converter for High Power Applications," *IEEE Trans. Ind. Appl.*, vol. 27, no. 1, Jan/Feb. 1991, pp. 63-73.

[9] L. F. Costa, F. Hoffmann, G. Buticchi, and M. Liserre, "Comparative Analysis of Multiple Active Bridge Converters Configurations in Modular Smart Transformer," *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 191-202, Jan. 2019.

[10] C. Zhao, S. D. Round and J. W. Kolar, "An Isolated Three-Port Bidirectional DC-DC Converter With Decoupled Power Flow Management," *IEEE Trans. Power Electron.*, vol. 23, no. 5, pp. 2443-2453, Sep. 2008.

[11] Y. Chen, P. Wang, H. Li and M. Chen, "Power Flow Control in Multi-Active-Bridge Converters: Theories and Applications," *IEEE Applied Power Electron. Conf. and Expo. (APEC)*, 2019, pp. 1500-1507.

[12] P. Wang and M. Chen, "Towards Power FPGA: Architecture, Modeling and Control of Multiport Power Converters," *IEEE Workshop on Control and Modeling of Power Electronics (COMPEL)*, Padua, Italy, June 2018.

[13] M. Chen, M. Araghchini, K. K. Afridi, J. H. Lang, C. R. Sullivan and D. J. Perreault, "A Systematic Approach to Modeling Impedances and Current Distribution in Planar Magnetics," *IEEE Trans. Power Electron.*, vol. 31, no. 1, pp. 560-580, Jan. 2016.

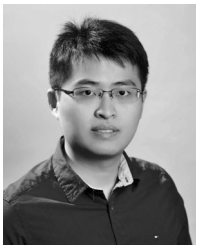
[14] J. E. Dennis, and J. J. More, "Quasi-Newton methods, motivation and theory," *SIAM review*, 19(1), pp. 46-89, Jan. 1977.
[15] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv:1609.04747*, Sep. 2016.
[16] S. Zhao, F. Blaabjerg, and H. Wang, "An Overview of Artificial Intelligence Applications for Power Electronics," *IEEE Trans. Power Electron.*, vol. 36, no. 4, pp. 4633-4658, Apr. 2021.
[17] P. Yang, W. Ming, J. Liang, I. Lüdtke, S. Berry and K. Floros,"Hybrid Data-Driven Modeling Methodology for Fast and Accurate Transient Simulation of SiC MOSFETs," *IEEE Trans. Power Electron.*, vol. 37, no. 1, pp. 440-451, Jan. 2022.
[18] B. K. Bose, "Neural Network Applications in Power Electronics and Motor Drives—An Introduction and Perspective," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 14-33, Feb. 2007.
[19] Y. Zeng et al., "Active Disturbance Rejection Control Using Artificial Neural Network for Dual-Active-Bridge-Based Energy Storage System," *IEEE J. Emerg. Sel. Topics Power Electron*, eary access.
[20] G. Rojas-Dueñas, J.-R. Riba and M. Moreno-Eguilaz, "A Deep Learning-Based Modeling of a 270 V-to-28 V DC-DC Converter Used in More Electric Aircrafts," *IEEE Trans. Power Electron.*, vol. 37, no. 1, pp. 509-518, Jan. 2022.
[21] H. Li, S. R. Lee, M. Luo, C. R. Sullivan, Y. Chen and M. Chen, "MagNet: A Machine Learning Framework for Magnetic Core Loss Modeling," *IEEE Workshop on Control and Modeling of Power Electronics (COMPEL)*, Aalborg, Denmark, 2020.
[22] E. Dogariu, H. Li, D. Serrano, S. Wang, M. Luo and M. Chen, "Transfer Learning Methods for Magnetic Core Loss Modeling," *IEEE Workshop on Control and Modeling of Power Electronics (COMPEL)*, Cartagena de Indias, Colombia, 2021.
[23] H. Li, D. Serrano, T. Guillod, E. Dogariu, A. Nadler, S. Wang, M. Luo, V. Bansal, Y. Chen, C. R. Sullivan, and M. Chen, "MagNet: an Open-Source Database for Data-Driven Magnetic Core Loss Modeling," *IEEE Applied Power Electron. Conf. (APEC)*, Houston, 2022.
[24] T. Guillod, P. Papamanolis and J. W. Kolar, "Artificial Neural Network (ANN) Based Fast and Accurate Inductor Modeling and Design," *IEEE Open Journal of Power Electronics*, vol. 1, pp. 284-299, 2020.
[25] M. Zhang, X. Wang, D. Yang and M. G. Christensen, "Artificial Neural Network Based Identification of Multi-Operating-Point Impedance Model," *IEEE Trans. Power Electron.*, vol. 36, no. 2, pp. 1231-1235, Feb. 2021.
[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
[27] S. Falcones, R. Ayyanar, and X. Mao, "A DC-DC Multiport-Converter Based Solid-State Transformer Integrating Distributed Generation and Storage," *IEEE Trans. Power Electron.*, vol. 28, no. 5, pp. 2192-2203, May 2013.
[28] J. D. Glover, M. S. Sarma, and T. J. Overbye, *Power System Analysis and Design*, 4th ed. Toronto, ON, Canada: Thomson, 2008.
[29] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.
[30] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," arXiv:1603.04467 [cs.DC], Mar. 2016.
[31] J. Machowski, J. W. Bialec, and J. R. Bumby, *Power System Dynamics and Stability*, John Wiley&Sons, 1998.
[32] M. Ilic, and J. Zaborszky, *Dynamics and Control of Large Electric Power System*, John Wiley&Sons, 2000.
[33] D. Blalock, J. J. O. Gonzalez, J. Frankle, and J. Guttag, "What is the State of Neural Network Pruning?," arXiv:2003.03033.

Mian Liao (Student Member, IEEE) received the B.S. degree in electrical engineering from Virginia Tech, VA, USA, in 2020, and the M.A. degree in electrical engineering in 2022 from Princeton University, NJ, USA, where he is currently working toward the Ph.D. degree.

His research interests include machine learning control in power electronics, multiport dc-dc converters, and point-of-load converters for CPU and data center applications.

Haoran Li (Student Member, IEEE) received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2019. He received the M.A. degree in electrical and computer engineering from Princeton University, Princeton, NJ, USA in 2021 and is currently working towards the Ph.D. degree.

His research interests include the measurement and characterization of power magnetics materials, and neural network methods for the modeling of power magnetics.



Ping Wang (Student Member, IEEE) received the B.S. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2017, and the M.A. degree in electrical engineering in 2019 from Princeton University, NJ, USA, where he is currently working toward the Ph.D. degree.

His research interests include high-efficiency/ high-density power converters, multiport dc-dc converters, and high-performance power electronics design for data center and telecom applications.

Mr. Wang received two Prize Paper Awards of the IEEE Transactions on Power Electronics in 2020 and 2021 respectively, two National Scholarship Awards in 2014 and 2016 at Shanghai Jiao Tong University, two First Place Awards of the IEEE ECCE Best Student Project Demonstration Competition in 2019 and 2021, the First Place Award from the Innovation Forum of Princeton University in 2019, and the outstanding presentation award in the 2022 IEEE APEC.

Tanuj Sen (Student Member, IEEE) received the bachelor's degree in Electrical and Electronics Engineering from Birla Institute of Technology and Science, Pilani, India in 2015 and his Master of Sciences degree in Electrical Engineering and Information Technology from ETH Zurich, Zurich, Switzerland in 2019, respectively. He received his M.A. degree in Electrical and Computer Engineering from Princeton University in 2022.

He is currently pursuing his Ph.D. degree in Electrical and Computer Engineering at Princeton University. His research interests include the design of high-power density, high-frequency resonant power electronic inverters for powering plasma heating coils in fusion reactors, as well as the design and analysis of coupled inductors and their application in high-frequency power electronic circuits.

Yenan Chen (Member, IEEE) received his honors degree of engineering from the Chu Kochen College, Zhejiang University in 2010, the bachelor's degree and Ph.D. degree in Electrical Engineering from the College of Electrical Engineering, Zhejiang University, Hangzhou, China, in 2010 and 2018, respectively.

He was a Postdoctoral Research Associate with the Department of Electrical Engineering, Princeton University, NJ, USA, from 2018 to 2021. Since December 2021, he has been with the Hangzhou Global Scientific and Technological Innovation Center and the College of Electrical Engineering at Zhejiang University, China, as a Research Scientist and Principal Investigator. He holds five issued Chinese patents. His research interests include power electronic topologies, architectures and control for data center, renewable energy and transportation.

Dr. Chen was the recipient of two Prize Paper Awards of the IEEE Transactions on Power Electronics (2020, 2021), the IEEE COMPEL Best Paper Award in 2020, the IEEE APEC Outstanding Presentation Award in 2019, and the First Place Award from the Innovation Forum of Princeton University in 2019.

Minjie Chen (Senior Member, IEEE) received the B.S. degree from Tsinghua University, Beijing, China, in 2009, and the S.M., E.E., and Ph.D. degrees from Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2012, 2014, and 2015, respectively. He is an Assistant Professor of Electrical and Computer Engineering and Andlinger Center for Energy and the Environment at Princeton University, where he leads the Princeton Power Electronics Research Lab. His research interests include complex power architecture, circuit-magnetics co-design, control, and machine learning methods for power magnetics modeling.

Dr. Chen is the Vice Chair of IEEE PELS Technical Committee on Design Methodologies (TC10), Associate Editor of IEEE Transactions on Power Electronics, Associate Editor of IEEE Journal of Emerging and Selected Topics in Power Electronics, Associate Technical Program Committee Chair of IEEE Energy Conversion Congress and Exposition (ECCE) in 2019, Student Activity Chair of IEEE Energy Conversion Congress and Exposition (ECCE) in 2020, and the Technical Program Committee Chair of IEEE International Conference on DC Microgrids (ICDCM) in 2021.

Dr. Chen is a recipient of the Princeton SEAS E. Lawrence Keyes, Jr./Emerson Electric Co. Junior Faculty Award (2022), five IEEE Transactions Prize Paper Awards (2016, 2017, 2020, 2021, 2021), a COMPEL Best Paper Award (2020), an OCP Best Paper Award (2021), the NSF CAREER Award (2019), a Dimitris N. Chorafas Award for outstanding MIT Ph.D. Thesis (2015), three ECCE Best Demonstration Awards (2014, 2019, 2021), multiple APEC Outstanding Presentation Awards, a Siebel Energy Institute Research Award (2017), a C3.ai DTI Research Award (2021), the First Place Award from the Innovation Forum of Princeton University (2019), and many other awards. He was included in the Princeton Engineering Commendation List for Outstanding Teaching from 2019 to 2022. He holds 7 U.S. patents.