Submodule Capacitor Voltage Estimation in Modular Multilevel Converters (MMCs)

1st Andrew Rouze
dept. of Electrical Engineering
Colorado School of Mines
Golden, USA
andrewrouze@mines.edu

2nd Dylan Wald
dept. of Electrical Engineering
Colorado School of Mines
Golden, USA
dylanwald@mines.edu

3rd Ghazaleh Sarfi
dept. of Electrical Engineering
Colorado School of Mines
Golden, USA
ghazaleh_sarfi@mines.edu

***Abstract*—This study focuses on the estimation of submodule capacitor voltages in Modular Multilevel Converters (MMCs). It addresses the challenges in operating and controlling MMCs, emphasizing the need for accurate estimation of these voltages to function efficiently. Traditional sensor-based methods, model-driven estimation, and newer data-driven estimation techniques are discussed, highlighting their roles in reducing costs and simplifying system architecture. The paper introduces a novel approach combining model-driven and data-driven methods, utilizing Physics-Informed Neural Networks (PINNs) to enhance estimation accuracy and generalizability across various MMC operating conditions. This innovative method integrates known physical laws with machine learning, offering a promising solution for efficient state estimation in MMCs.**

***Index Terms*—PINN, MMC, state estimation**

I. Introduction

The modular multilevel converter (MMC) has become an attractive multilevel converter topology for medium/high-power applications. The MMC offers many features, including modularity and scalability to meet any voltage level requirements, high efficiency (important in high-power applications), and superior harmonic performance to reduce passive filter size to name a few. The MMC is a sophisticated technology used primarily in power electronics to convert electricity from one form to another, especially in high-voltage direct current (HVDC) transmission systems.

An MMC consists of multiple sub-modules (SMs), each containing a small energy storage component such as a capacitor, and power electronic switches. These SMs are connected in series to form a 'chain', offering scalability and flexibility in voltage levels, see Fig. 1. The key advantage of an MMC is its ability to produce high-quality voltage waveforms with minimal harmonic distortion, which is crucial for efficient and stable power transmission. Additionally, MMCs offer benefits such as reduced switching losses, modular structure for easy maintenance, and better control over power flow, making them an increasingly popular choice in renewable energy systems, power grids, and other applications [1].

Over the past few years, there has been a significant effort towards addressing the technical challenges associated with the operation and control of MMCs as well as broadening its applications [2], [3]. The MMC configuration necessitates numerous current and voltage measurements, as well as extensive communication within the converter arms and between the arm controllers and the central control circuit. To simplify the circuit design and minimize hardware needs, thereby cutting costs and enhancing the MMC's operational reliability, various strategies have been suggested for estimating its internal electrical variables or parameters. These strategies focus on estimating SM capacitor voltages, DC-link voltages, arm currents, or the overall arm energy [4]. The objective of these methods is to either eliminate or minimize the number of sensors in the circuit, decrease communication needs, or achieve a combination of both. In this work, we focus on the estimation of SM capacitor voltages V_c.

Accurate estimation of SM V_c is crucial for the stable and efficient operation of MMCs. Traditionally, sensor-based methods have been used for this purpose. However, the advent of estimation methods has opened new avenues for making MMCs sensorless, which is beneficial for reducing costs, increasing reliability, and simplifying the system architecture [5].

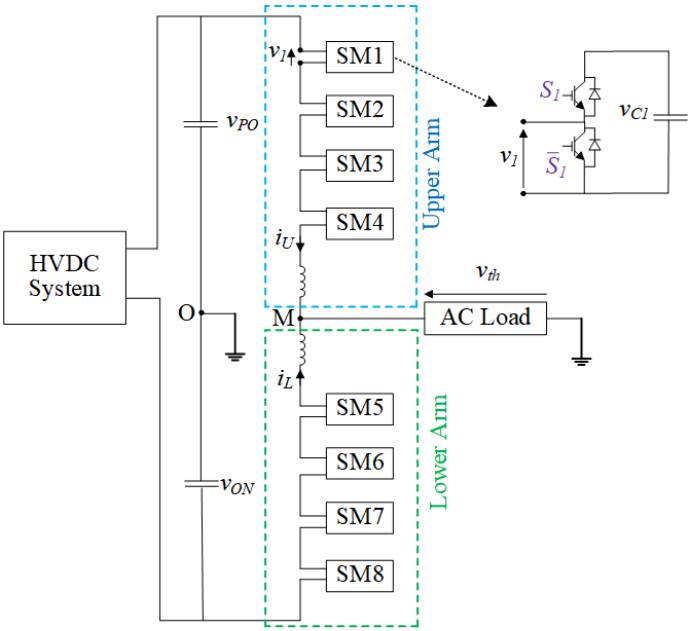

Fig. 1. The Modular Multilevel Converter (MMC) scheme used in this work.

A. Existing Literature

Many methods exist to estimate SM V_c within MMCs to circumvent the need for direct measurements. These methods generally rely on either the mathematical modeling of MMCs (*model-driven methods*) or data obtained from actual MMC operation (*data-driven methods*).

Model driven methods use Kirchoff's laws to obtain equations that describe the MMC's behavior, and once these are obtained, various V_c estimation techniques can be applied. Multiple articles utilized state observers to estimate V_c and minimize the number of required sensors. A closed-loop state observer is implemented in [6] after developing the state space equations for the given MMC, which resulted in a high degree of accuracy for V_c estimation. The robustness of the aforementioned closed-loop state observer was confirmed via different loading and fault conditions. Different variations on state observer models were analyzed as well. An adaptive backstepping observer was utilized to accurately estimate both SM V_c along with SM capacitance while measuring only MMC arm currents [7]. Similar results of accurate V_c and SM capacitance estimates were achieved in [8] with the use of an interconnected observer. A highly flexible observer model was developed in [9] that could be adapted to three-phase MMCs with any number of SMs as well as different SM topologies (i.e., half-bridge, full-bridge, etc.). However, the SM capacitance must be known. Both [10] and [11] utilize adaptive observers to estimate SM V_c and capacitance, with the latter model being additionally capable of SM leakage resistance.

Other models were built using sliding-mode observers and Kalman filters. The accuracy of sliding-mode observers is demonstrated in [12] and [13], while a discretized form of the sliding mode observer is shown in [14]. The method proposed in [15] utilizes a Kalman filter to accurately predict SM V_c for single phase MMCs with any amount of SMs requiring only one voltage sensor per arm. The need for SM voltage sensors is entirely eliminated by [16] and [5], which also utilize Kalman filters to estimate SM voltage. In general, model based V_c estimation methods can be accurate, but are sensitive to variations in the physical characteristics of the MMC, and can be very computationally intensive.

To alleviate computational burden and improve accuracy associated with *model-driven methods*, some *data-driven methods* have been applied to the SM V_c estimation problem in MMCs. The work in [17] introduces an encoder-decoder method to estimate V_c values. They employ a convolutional neural network encoder and recurrent neural network decoder to capture slow and fast dynamics of the MMC, respectively. The work in [18] uses a distributed neural network (DNN) to estimate SM V_c. Using only one V_c sensor per arm, the DNN can parallelize computation and accurately estimate SM V_c in transient and steady-state operation, in real time. While these methods tend to return higher estimation accuracy than *model-based methods*, they require a large amount of learning parameters, data, and computational effort. Additionally, as black-box methods, they do not generalize well to data they are not trained on.

B. Contribution

In this work, we propose a SM V_c estimation method that exploits the benefits of *model-driven* and *data-driven methods*: a Physics Informed Neural Network (PINN). PINNs incorporate physical laws into the neural network's learning process, enabling them to utilize both data and the intrinsic physical principles of the system. This fusion results in models that are accurate and generalizable across various operating conditions, which is essential for estimating SM V_c in MMCs. Thanks to their integration of underlying physical laws, PINNs can generalize better to unseen scenarios compared to black-box data driven methods, a feature particularly advantageous in MMC applications where operating conditions are highly variable [19].

The state estimation problem, which involves estimating an unmeasurable state using measurable states and outputs, aligns well with the PINN approach. State estimators typically rely on known physical equations of the system, and integrating this concept with PINNs allows for the effective estimation of unmeasured states. This idea was tested in [19], where the authors augment the loss function of a neural network using known physical equations to estimate an unmeasurable state. Hence, in this work we leverage the findings in [19] to estimate the SM V_c values in a more efficient and reliable manner than both *model-driven methods* and *data-driven methods*.

II. Methods

In order to train the PINN estimator, a dataset must be obtained, a learning algorithm employed, and the loss function must be augmented. This section describes the dataset generation using MATLAB, the mathematical model of the MMC used in this work, the learning algorithm, and finally the method used to create the PINN estimator.

A. MATLAB Model

Data for the PINN estimator was created synthetically using a MATLAB MMC model. The MATLAB MMC model simulated a single-phase MMC with four SM on each arm, i.e., a total of eight SMs [20]. Each DC source of the MMC was set to 2000 V, and each SM capacitor has an initial condition of 1000 V. A PWM controller was used to control each SM, yielding an approximately sinusoidal waveform with an amplitude of 2000 V. The following data were measured and used to train and test the performance of the PINN estimator: upper arm current i_1, lower arm current i_2, output voltage V_{th}, switching signals S, and capacitor voltage V_c. We define the dataset $\mathcal{D} = \{(\mathbf{X}_i, \mathbf{y}_i)\}_{i=1}^{N}$ where $\mathbf{X}_i = (i_{1,i}, i_{2,i}, S_{1,i}, .., S_{8,i})$ and $\mathbf{y}_i = (V_{th,i})$ where N is the number of datapoints.

To test the robustness of the model, three scenarios were tested to simulate probable events an MMC would encounter. In scenario 1), the MMC operated under normal conditions at steady state. In scenario 2), a short-circuit fault was applied to the output of the MMC. This short-circuit fault then returned

to normal conditions after two cycles. In scenario 3), a single SM was bypassed for two cycles. The regular switching pattern was re-applied after the two cycles.

B. Analytical MMC Model

With a dataset acquired, a mathematical model of the MMC is identified. To be consistent with the MATLAB model described in Section II-A, we model a single-phase MMC with eight SMs as shown in Fig. 1.

Using the theory introduced in [21], the continuous time V_c dynamics in each SM can be described as

$$\frac{dV_{ci}(t)}{dt} = \frac{1}{C_i}(-i_j(t)S_i(t)) \tag{1}$$

where t is time, i_j is the current in arm $j \in \{1, 2\}$ (1 denotes the upper arm and 2 denotes the lower arm), S_i is the switching signal in SM $i \in \{1, ..., 8\}$, and C_i is the capacitance of SM i. Note that for $j = 1$, $i \in \{1, ..., 4\}$ and for $j = 2$, $i \in \{5, ..., 8\}$. In addition to the V_c dynamics, the authors in [21] define the output (Thevenin) voltage V_{th} as

$$V_{th} = \frac{V_{ON} - V_{PO}}{2} - \sum_{i=1}^{4} \frac{V_{ci}S_i}{2} + \sum_{i=5}^{8} \frac{V_{ci}S_i}{2} \tag{2}$$

where V_{ON} and V_{PO} are constants.

In this work, we use discrete data and thus we must discretize the continuous time dynamics in (1). To do this, we use the forward Euler method [22]. The discretized dynamics are described as

$$V_{ci}[k+1] = V_{ci}[k] + \alpha \left[\frac{1}{C_i}(-i_j[k]S_i[k])\right] \tag{3}$$

where $\alpha = 1e-5$ and k is the discrete time step. Because (2) is not a differential equation, the forward Euler method need not be applied. For the remainder of this report, we refer to the state dynamics as $\mathbf{f_{MMC}}(\cdot)$ and the output dynamics as $\mathbf{g_{MMC}}(\cdot)$. For a full description of $\mathbf{f_{MMC}}(\cdot)$, see (6) in the Appendix.

C. Physics Informed Neural Network

As previously mentioned, machine learning and physics knowledge is fused to build a PINN. In this section, the learning algorithm and augmented loss function are described in detail. The PINN framework is summarized in Fig. 2.

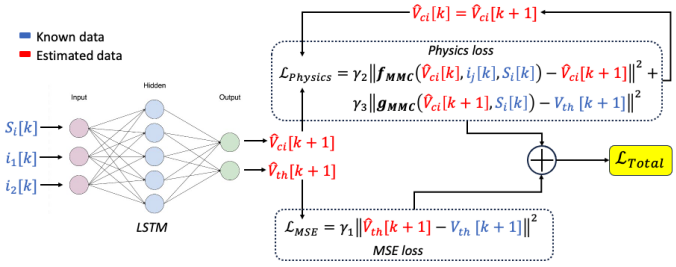

Fig. 2. The PINN framework used in this work.

1) Learning algorithm: Recurrent neural networks (RNNs) are a type of artificial neural network that can retain a sort of memory, allowing it to learn patters in sequences of data [23]. The performance of a typical RNN tends to degrade due to a vanishing or exploding gradient, especially over long sequences. A Long Short-term Memory (LSTM) network is a type RNN that aims to mitigate the vanishing or exploding gradient problem [24]. While more complex than an RNN, the LSTM has proven successful in many applications that involve learning patterns in sequential data, such as forecasting.

An estimation problem involves estimating a state in a dynamical system, in this case V_c in an MMC, at time $k+1$ based on measurements and state estimations at the previous timestep k. In other words, the next state depends on the current and past states. In machine learning problems, the goal is to map input data to output data. In the case of a dynamical system, an MMC in this problem, this input-output data is sequential. For this reason, the LSTM is a clear candidate for this application and thus is used as the learning algorithm in this work. Using data generated from MATLAB, the LSTM is applied to learn how V_{th} and $V_{c,i}$ change sequentially through time with respect to the two arm currents i, and switching pattern S for each SM. The left side of Fig. 2 shows how the LSTM fits into the PINN framework.

2) Loss function: A typical regression problem uses a mean square error (MSE) type loss function, which has the form

$$\mathcal{L}_{MSE} = \gamma_1 ||\hat{y} - y||^2 \tag{4}$$

where $\hat{y}$ is the output from a model and y is the true output. In this work, $\hat{y} = \hat{V}_{th}[k+1]$ and $y = V_{th}[k+1]$. Note that (4) cannot be applied to learn $V_{c,i}$ directly since $V_{c,i}$ data cannot be measured. Therefore, we utilize known dynamics $\mathbf{f_{MMC}}(\cdot)$ and $\mathbf{g_{MMC}}(\cdot)$ to add to the LSTM loss function as shown below

$$\begin{aligned}\mathcal{L}_{physics} = \\ &\gamma_2 ||\mathbf{f_{MMC}}(\hat{V}_{c,i}[k], i_j[k], S_i[k]) - \hat{V}_{c,i}[k+1]||^2 \\ &+ \gamma_3 ||\mathbf{g_{MMC}}(\hat{V}_{c,i}[k+1], S_i[k]) - \hat{V}_{th}[k+1]||^2\end{aligned} \tag{5}$$

where $\gamma_{1,2,3}$ are arbitrary weights on each loss. The first part of (5) compares the physics-estimated states $\bar{V}_{c,i}$ using known inputs, past estimated states, and dynamics $\mathbf{f_{MMC}}(\cdot)$ to the LSTM-estimated states $\hat{V}_{c,i}$. The second part of (5) compares the physics estimated output $\bar{V}_{th}$ using known inputs, past estimated states, and dynamics $\mathbf{g_{MMC}}(\cdot)$ to the LSTM-estimated output $\hat{V}_{th}$. The full PINN loss function is then simply $\mathcal{L}_{PINN} = \mathcal{L}_{MSE} + \mathcal{L}_{physics}$.

III. Results and Discussion

With data obtained and the PINN framework defined, the PINN estimator is trained. The dataset contains 0.2 seconds of data, with a time step of $1e-5$ sec ($N = 20000$). The PINN is trained using the Adam optimzer with 300 Epochs and a batch size of 200. A 80% - 20% train-test split is performed on the data. Additionally, a simple scalar normalization of 0.001 and 0.1 is applied to the input and output datasets,

respectively. Last, the hyperparameters of the PINN are tuned using the Optuna black-box hyperparameter optimization tool [25]. Table I below displays the optimal parameters used in this work.

TABLE I
OPTIMAL PINN HYPERPARAMETERS

Hyperparameter	Value
Hidden nodes	201
Learning rate	0.00813
γ_1	1.514
γ_2	0.8441
γ_3	0.000579

As mentioned in Section II-A, three different scenarios are analyzed: 1) normal conditions, 2) a fault condition, and 3) a bypass condition. First, the PINN estimator results on the training data for scenario 1 is analyzed. From Fig. 3, it can

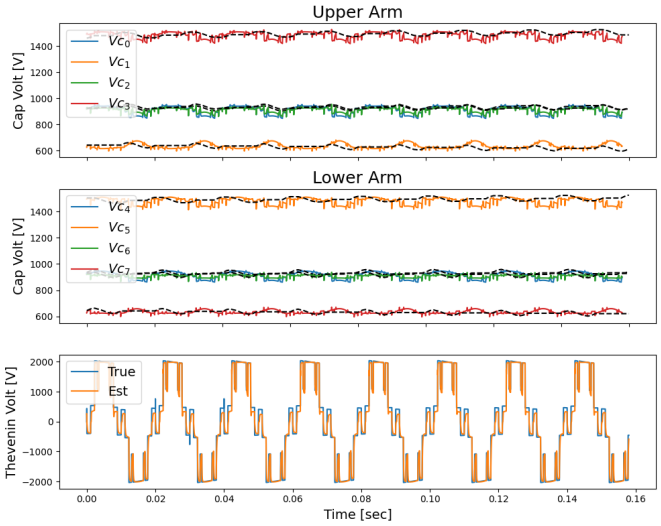

Fig. 3. PINN performance on training data under normal, steady-state conditions. **Top**: upper arm capacitor voltages, **Middle**: lower arm capacitor voltages, and **Bottom**: Thevenin voltage.

be seen that the PINN performs well on the training data for scenario 1. While not perfect, the estimated V_c values in both the upper and lower arm follow the corresponding true values well. The response is much noisier than the true values, but the magnitude and general response is captured by the PINN estimator. The bottom plot shows the PINN output prediction, which performs extremely well on the training data.

From Fig. 4, it can be seen that the performance of the PINN estimator for scenario 2 is much worse than in scenario 1. It is worse by a factor of more than 10 based on the results in Table II. While the PINN captures the knowledge of the fault occurrence and the general trend of the states and outputs during the fault, the accuracy is poor. Additionally, the PINN response prior to the fault is skewed, where in the Fig. 3 it is favorable. Last, the output prediction is much worse, especially for the bottom portion on the period. Note that extending the

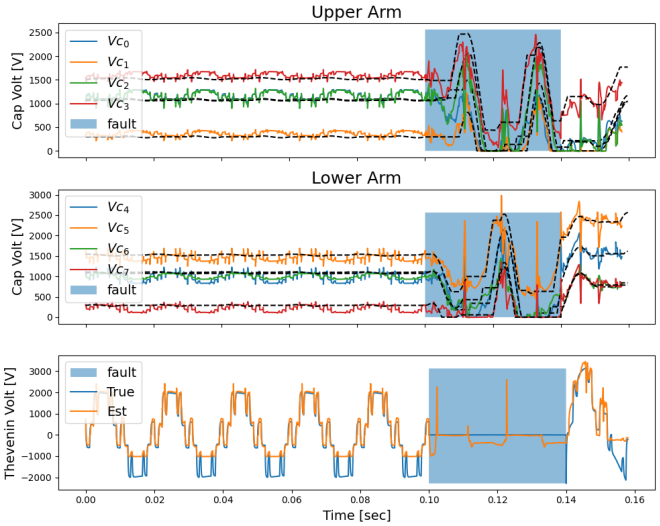

Fig. 4. PINN performance on training data under a fault condition. **Top**: upper arm capacitor voltages, **Middle**: lower arm capacitor voltages, and **Bottom**: Thevenin voltage.

number of epochs improves the result on the training data for scenario 1, but results in overfitting.

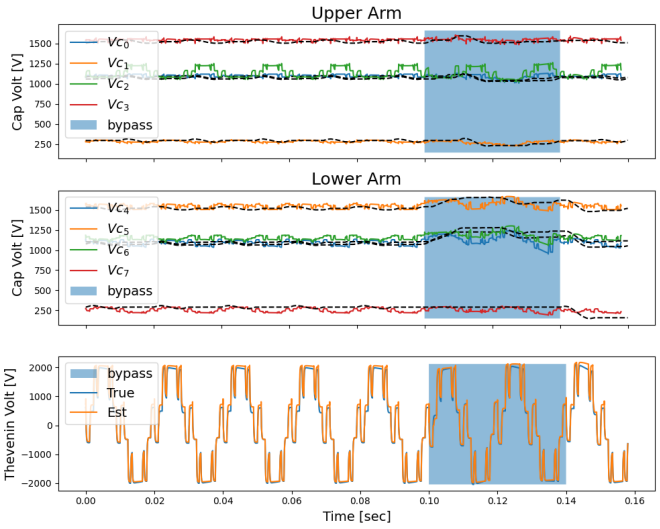

Fig. 5. PINN performance on training data under a bypass condition. **Top**: upper arm capacitor voltages, **Middle**: lower arm capacitor voltages, and **Bottom**: Thevenin voltage.

From Fig. 5, it can be seen that the performance of the PINN estimator for scenario 3) is relatively good. The V_c estimates in the upper arm as well as the output are predicted well. However, the lower leg V_c estimates are poorer. Table III in the Appendix, indicates that while the performance of the PINN in scenario 3 is comparable to that of scenario 1, the individual component MSE values are switched. This indicates that the PINN can more accurately estimate the SM V_c values under normal conditions, but struggles under bypass or fault conditions. The opposite can be concluded about the MMC

output voltage estimate.

Last, from the results in Tables II and III in the Appendix, it can be concluded that the PINN is not overfitting for the normal and bypass cases as their total MSE values are comparable for both datsets. However, this is not the case for the fault dataset. The test set total MSE is more than 10 times as high as the train set total MSE. Note that while the PINN parameters in Table I were tuned to the normal condition case, the same parameters were used for each of the three cases. Because the fault condition case includes data that is quite different than in the normal and bypass condition cases, the parameters may be inadequate for this condition. Further hyperparameter optimization is something we wish to investigate in future work.

IV. Conclusion

Reducing the number of sensors required for an MMC is critical to reduce its cost and improve its reliability. Various V_c estimation methods have been proposed to reduce or eliminate V_c sensors in an MMC. The proposed method utilizes a novel PINN estimator to do so. For normal operating conditions, the PINN accurately estimates the output of the MMC system V_{Th} and can estimate individual capacitor voltages $V_{c,i}$ with some degree of accuracy. However, the $V_{c,i}$ estimates are much more noisy than the true data. Hence, further work would be required to determine if these $V_{c,i}$ estimates are acceptable for MMC control schemes or how to improve the PINN estimations. The PINN did not perform as well under both fault conditions and SM bypass conditions. This is most likely due to the fact that the PINN's hyperparameters were tuned to the normal operating conditions. Hence, future work would be required to tune these hyperparameters for additional operating conditions.

References

[1] J. Pou, M. A. Perez, and R. P. Aguilera, "Modular multilevel converters," *IEEE Trans. on Industrial Electronics*, vol. 66, no. 3, pp. 2204–2206, 2019.

[2] S. Debnath, J. Qin, B. Bahrani, M. Saeedifard, and P. Barbosa, "Operation, control, and applications of the modular multilevel converter: A review," *IEEE Trans. on Power Electronics*, vol. 30, no. 1, pp. 37–53, 2015.

[3] M. A. Perez, S. Ceballos, G. Konstantinou, J. Pou, and R. P. Aguilera, "Modular multilevel converters: Recent achievements and challenges," *IEEE Open Journal of the Industrial Electronics Society*, vol. 2, pp. 224–239, 2021.

[4] M. Trabelsi, M. Ghanes, O. Ellabban, H. Abu-Rub, and L. Ben-Brahim, "An interconnected observer for modular multilevel converter," in *2016 IEEE ECCE*, pp. 1–7, 2016.

[5] M. D. Islam, R. Razzaghi, and B. Bahrani, "Arm-sensorless sub-module voltage estimation and balancing of modular multilevel converters," *IEEE Trans. on Power Delivery*, vol. 35, no. 2, pp. 957–967, 2020.

[6] H. Liu, K. Ma, P. C. Loh, and F. Blaabjerg, "Design of state observer for modular multilevel converter," in *2015 IEEE 6th International Symposium on PEDG*, pp. 1–6, 2015.

[7] V. Najmi, H. Nademi, and R. Burgos, "An adaptive backstepping observer for modular multilevel converter," in *2014 IEEE ECCE*, pp. 2115–2120, 2014.

[8] M. Trabelsi, M. Ghanes, O. Ellabban, H. Abu-Rub, and L. Ben-Brahim, "An interconnected observer for modular multilevel converter," in *2016 IEEE ECCE*, pp. 1–7, 2016.

[9] L.-A. Grégoire, W. Wang, S. I. Seleme, and M. Fadel, "High reliability observers for modular multilevel converter capacitor voltage evaluation," in *2016 IEEE 8th IPEMC-ECCE Asia*, pp. 2332–2336, 2016.

[10] H. Nademi, A. Das, and L. E. Norum, "Modular multilevel converter with an adaptive observer of capacitor voltages," *IEEE Trans. on Power Electronics*, vol. 30, no. 1, pp. 235–248, 2015.

[11] M. Liu and Z.-T. Li, "Parameter estimation of sub-module capacitors of modular multi-level converter based on adaptive observer," in *2021 China Automation Congress (CAC)*, pp. 4803–4807, 2021.

[12] G. S. da Silva, R. P. Vieira, and C. Rech, "Modified sliding-mode observer of capacitor voltages in modular multilevel converter," in *2015 IEEE 13th Brazilian Power Electronics Conference and 1st Southern Power Electronics Conference (COBEP/SPEC)*, pp. 1–6, 2015.

[13] A. Al-Wedami, K. Al-Hosani, and A. R. Beig, "Sliding mode observer of submodular capacitor voltages in modular multilevel converter," in *2015 International Workshop on RASM*, pp. 1–6, 2015.

[14] G. S. da Silva, R. P. Vieira, and C. Rech, "Discrete-time sliding-mode observer for capacitor voltage control in modular multilevel converters," *IEEE Trans. on Industrial Electronics*, vol. 65, no. 1, pp. 876–886, 2018.

[15] O. S. H. M. Abushafa, M. S. A. Dahidah, S. M. Gadoue, and D. J. Atkinson, "Submodule voltage estimation scheme in modular multilevel converters with reduced voltage sensors based on kalman filter approach," *IEEE Trans. on Industrial Electronics*, vol. 65, no. 9, pp. 7025–7035, 2018.

[16] O. Abushafa, S. Gadoue, M. Dhaidah, and D. Aktinson, "Capacitor voltage estimation in modular multilevel converters using a kalman filter algorithm," in *2015 IEEE ICIT*, pp. 3016–3021, 2015.

[17] S. Langarica, G. Pizarro, P. M. Poblete, F. Radrigán, J. Pereda, J. Rodriguez, and F. Núñez, "Denoising and Voltage Estimation in Modular Multilevel Converters Using Deep Neural-Networks," *IEEE Access*, vol. 8, pp. 207973–207981, 2020. Conference Name: IEEE Access.

[18] P. Poblete, G. Pizarro, G. Droguett, F. Núñez, P. D. Judge, and J. Pereda, "Distributed Neural Network Observer for Submodule Capacitor Voltage Estimation in Modular Multilevel Converters," *IEEE Trans. on Power Electronics*, vol. 37, pp. 10306–10318, Sept. 2022. Conference Name: IEEE Transactions on Power Electronics.

[19] R. Patel, S. Bhartiya, and R. D. Gudi, "State Estimation Using Physics Constrained Neural Networks," in *2022 IEEE International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, pp. 61–66, Aug. 2022.

[20] T. M. Inc., "Matlab version: 9.13.0 (r2023b)," 2023.

[21] E. Solas, G. Abad, J. A. Barrena, A. Cárear, and S. Aurtenetxea, "Modelling, simulation and control of modular multilevel converter," in *Proceedings of 14th International Power Electronics and Motion Control Conf. EPE-PEMC 2010*, pp. T2–90–T2–96, 2010.

[22] "1.2: Forward Euler method," July 2022.

[23] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, 2018.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[25] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

Appendix

TABLE II
Training Data MSE Loss

Simulation	MSE Losses		
	V_C	V_{Th}	***Total***
Normal conditions	6075	33480	39555
Bypass conditions (SM 8)	22630	9522	32152
Fault Conditions	227107	214601	441708

TABLE III
TEST DATA MSE LOSS

Simulation	**MSE Losses**		
	V_C	V_{Th}	***Total***
Normal conditions	7986	33254	41240
Bypass conditions (SM 8)	17183	9786	26969
Fault Conditions	3472528	1469844	4942372

The full discretized state dynamics:

$$\begin{bmatrix} V_{c,1}[k+1] \\ V_{c,2}[k+1] \\ V_{c,3}[k+1] \\ V_{c,4}[k+1] \\ V_{c,5}[k+1] \\ V_{c,6}[k+1] \\ V_{c,7}[k+1] \\ V_{c,8}[k+1] \end{bmatrix} = \begin{bmatrix} V_{c,1}[k] \\ V_{c,2}[k] \\ V_{c,3}[k] \\ V_{c,4}[k] \\ V_{c,5}[k] \\ V_{c,6}[k] \\ V_{c,7}[k] \\ V_{c,8}[k] \end{bmatrix} - \alpha \begin{bmatrix} \frac{1}{C_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{C_2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{C_3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{C_4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{C_5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{C_6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{C_7} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{C_8} \end{bmatrix} \begin{bmatrix} i_1[k]S_1[k] \\ i_1[k]S_2[k] \\ i_1[k]S_3[k] \\ i_1[k]S_4[k] \\ i_2[k]S_5[k] \\ i_2[k]S_6[k] \\ i_2[k]S_7[k] \\ i_2[k]S_8[k] \end{bmatrix} \tag{6}$$

Link to GitHub repository:
https://github.com/dylan-wald/AI4PS_Project