IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 January 2024, accepted 17 January 2024, date of publication 22 January 2024, date of current version 29 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3356611

RESEARCH ARTICLE

Data-Driven Approach for Wind Farm Control: Toward an Alternative to FLORIS

MINJEONG KIM AND SUNGSU PARK
Department of Aerospace Engineering, Sejong University, Seoul 05006, South Korea

Corresponding author: Sungsu Park (sungsu@sejong.ac.kr)

This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) funded by the Korean Government [Ministry of Trade, Industry and Energy (MOTIE)] through the Development of Localized Control System for Wind Power Systems under Grant 20213030020230.

ABSTRACT In this paper, we introduce a data-driven approach to wind farm control, offering an alternative to the FLORIS wind farm simulator. Our method estimates the power of a wind farm and determines the optimal yaw angle to maximize power generation. Initially, we develop a power estimation neural network using data from FLORIS for power estimation and validate its accuracy and reliability. Subsequently, this power estimation neural network is employed to determine the optimal yaw angle for maximum power production. The efficacy of this yaw decision neural network is verified through various performance metrics. We then present dynamic simulations by integrating the yaw decision neural network, constructed through our data-driven approach, with a dynamic wind farm simulator. We believe this addresses the limitations of FLORIS, a steady-state simulator. Our results demonstrate the effectiveness of the proposed yaw decision neural network in dynamic environments, underscoring the potential of a data-driven approach to overcome the challenges posed by the steady-state wind farm simulator. This study offers innovative solutions for the efficient control and optimization of wind farm.

INDEX TERMS Wind farm control, data-driven approach, deep neural network, optimal yaw.

I. INTRODUCTION

Wind energy has emerged as a critical renewable energy source due to its sustainability, abundance, and potential to decrease greenhouse gas emissions. With the ongoing momentum in the development and deployment of wind power technologies, wind farms have been established worldwide to harness energy efficiently. However, wake interactions among turbines substantially influence the power production efficiency of wind farms, resulting in decreased output, increased mechanical loads, elevated maintenance costs, and shortened lifetimes. Addressing these issues necessitates the evolution of innovative control and optimization strategies [1].

To provide a comprehensive understanding of our approach towards addressing these challenges, Figure 1 offers a visual representation of the entire research process and methodology we have proposed.

Within the evolving landscape of wind farm control, a notable challenge emerges: the long-standing reliance on model-based optimization methods. While these traditional approaches, grounded in predefined mathematical models, have been instrumental, they occasionally exhibit limitations, especially when faced with the multifarious intricacies inherent to real-world wind farm dynamics. Our paper advocates a paradigm shift, emphasizing a model-free, data-driven approach. Harnessing the richness of real-world data, this methodology promises solutions marked by enhanced accuracy, adaptability, and freedom from the constraints intrinsic to conventional models.

In light of recent strides in wind farm control, research has progressively underlined the necessity for adept control strategies. Various studies have ventured into domains from power curve optimization to wake steering optimization, emphasizing collaborative strategies and the integration of diverse renewable energy sources [2], [3], [4], [5], [6]. Our work seeks to further this narrative, pioneering a pragmatic, innovative, data-driven solution. Although the

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaowei Zhao.

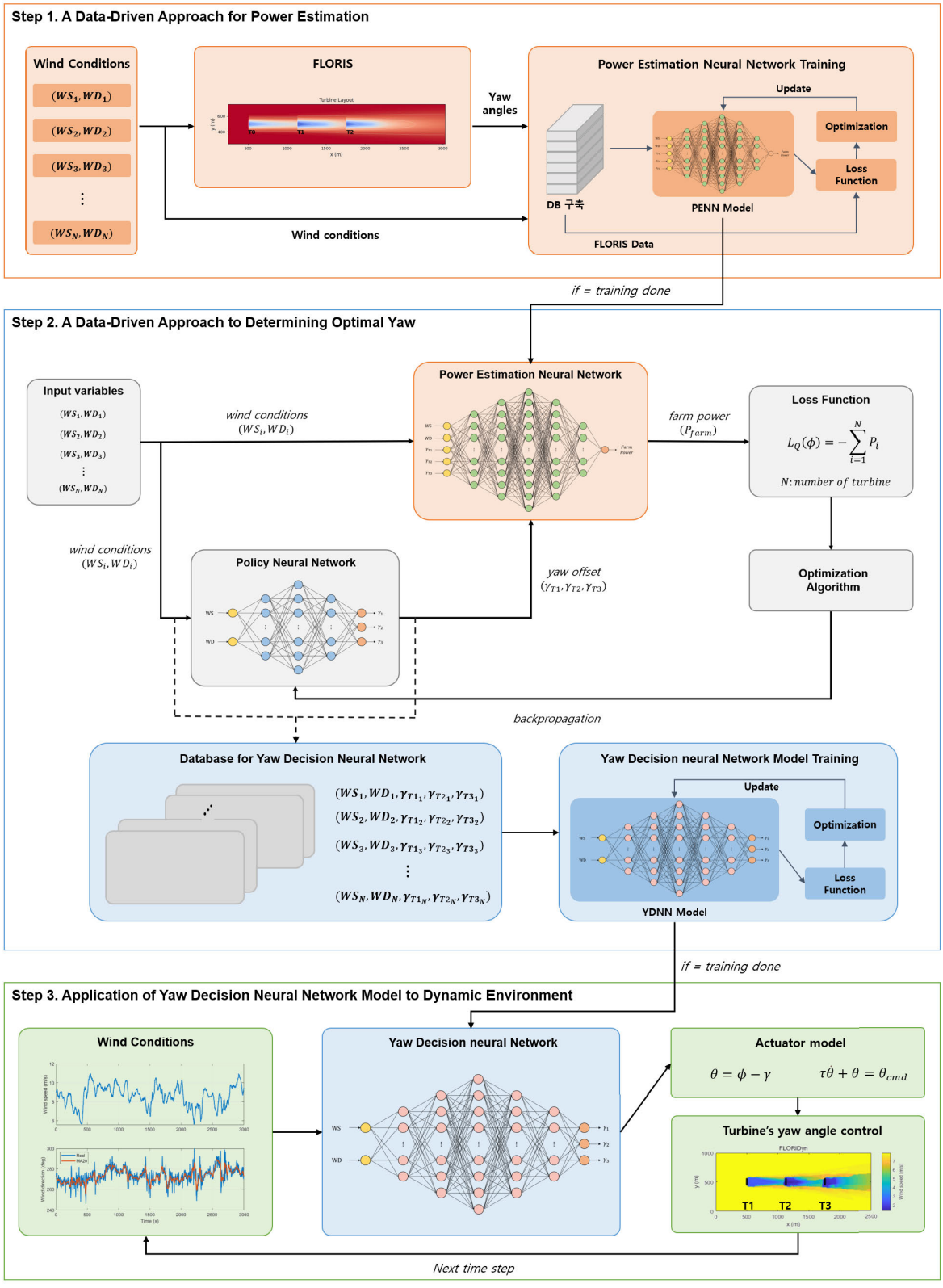

FIGURE 1. Overview of the research framework.

current emphasis is on simulations, the methodology we advocate sets the stage for subsequent research centered on actual wind farm data.

Wind farm simulators, such as the FLORIS model, play an indispensable role in shaping our understanding of wind farm aerodynamics, subsequently informing the genesis of tailored control strategies [7]. Amid myriad control tactics to boost wind farm power output, the yaw control strategy has garnered significant attention. It optimizes the yaw angle to minimize the wake effect of downstream turbines, bolstering overall power production while reducing energy costs [8], [9], [10], [11], [12], [13], [14]. Additionally, the application of reinforcement learning in yaw angle control has carved out a promising niche in recent years [15], [16], [17], [18], [19], [20].

Our contributions in this paper are fourfold:

1) We propose and evaluate a data-driven power estimation neural network as a replacement for the FLORIS model.
2) We introduce a methodology that uses the power estimation neural network to determine the yaw angle that maximizes the power output of a wind farm under various wind conditions.
3) We construct, evaluate, and validate yaw decision neural networks for optimal yaw angle identification using data-driven methods.
4) We integrate the yaw decision neural network into a dynamic wind farm simulator to demonstrate its capability and efficiency.

While our research accentuates a specific turbine layout to underscore our data-driven approach's advantage, its scope is not circumscribed to this layout. The methodology we propound forms a foundational scaffold, extendable to diverse wind farm structures, ensuring adaptability across a spectrum of wind energy setups.

The remainder of this paper is structured as follows: Section II provides an in-depth overview of the theoretical background required to understand the proposed work. Section III discusses the development and evaluation of power estimation neural networks using a data-driven approach as an alternative to FLORIS. Section IV describes how we use the power estimation neural network to determine the optimal yaw angle for maximizing wind farm power and details the configuration and evaluation of the yaw decision neural network for various wind conditions. In Section V, we demonstrate the capability of the proposed approach by applying the yaw decision neural network to a dynamic wind farm simulator. The final section summarizes the paper, discusses its limitations, and presents future research directions.

II. THEORETICAL BACKGROUND

A. WIND FARM SIMULATOR AND TURBINE MODEL

In this study, we utilize two key tools: the FLORIS wind farm simulator [7], a product of the National Renewable Energy Laboratory (NREL), and FLOW Redirection and Induction Dynamics (FLORIDyn) [21], developed by the Delft System and Control Center at Delft University of Technology. Our in-depth analysis of these wind farm simulators is grounded in the academic research conducted by Kim et al. [22]. The utility of FLORIS becomes evident as we harness it to construct the data-driven power estimation neural network proposed in this study. Conversely, FLORIDyn is employed to provide comparative insights into the data-driven method under realistically simulated conditions, furthering our investigation into the efficacy of the proposed approach.

For this study, we utilize the NREL 5MW wind turbine model. This particular model is held in high regard and is commonly employed within the domain of wind energy research, primarily due to its detailed and reliable specifications. The comprehensive specifications for the 5MW turbine model were sourced from the review of relevant literature [23].

B. TURBINE COORDINATE SYSTEM AND TERMINOLOGY

Figure 2 provides a visual representation of the wind turbine coordinate system. This system is vital for measuring the wind direction ϕ and the heading of the turbine θ. The coordinate system is defined with reference to the north direction. The wind direction and the heading direction of the turbine are measured clockwise relative to this northern reference.

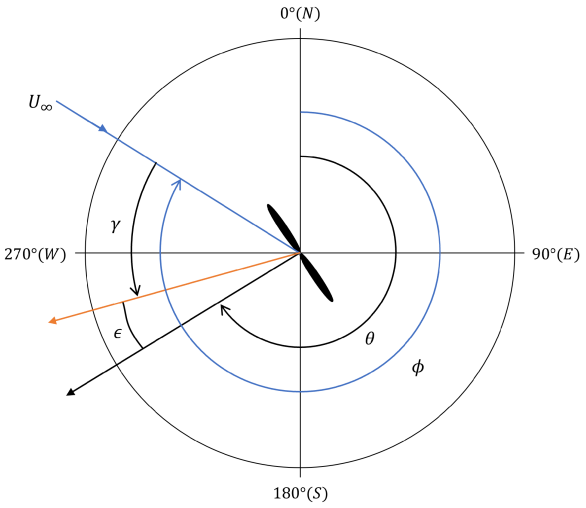

FIGURE 2. Coordinate system for wind turbines.

In this coordinate system, the desired yaw offset of the turbine, denoted as γ, is a crucial measure. It indicates the deviation of the turbine's heading from the direction of the incoming wind. The yaw offset is determined relative to the oncoming wind direction. It is set to positive (+) if the turbine is oriented counterclockwise from the wind direction, and negative (−) if it is oriented clockwise. The difference between the heading direction of the turbine and the desired yaw offset is epsilon ϵ and follows the following equation 1.

The yaw angle is controlled by reducing the ϵ values. In the context of our study, we have taken steps to set the ϵ value to zero. This decision emphasizes our methodology's specific approach to achieve the desired turbine orientation relative to the incoming wind, ensuring optimal power capture.

$$\epsilon = (\phi - \gamma) - \theta. \tag{1}$$

A thorough understanding of the turbine coordinate system and the concept of yaw offset (hereafter referred to as 'yaw') is essential for discussing and implementing the control strategies and optimization methods highlighted in this study.

C. YAW CONTROL IN WIND FARM

Yaw control is essential for optimizing wind farm performance due to its influence on wake interactions among wind turbines. By adjusting the yaw angle—the orientation of the turbine rotor relative to the wind direction—the downstream wake can be effectively managed. This wake, characterized by increased turbulence and reduced wind speeds, decreases the power production of downstream turbines, thereby emphasizing the importance of yaw control.

Adjusting the yaw angle allows the wake to be redirected away from downstream turbines, reducing interference and enabling increased power production. For instance, Figure 3 depicts how adjustments in the yaw angle can redirect wake flow, minimizing interference and enhancing power generation.

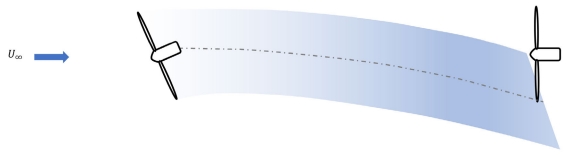

FIGURE 3. Effect of yaw angle control on wake.

Furthermore, yaw control can diminish the mechanical load on turbines, potentially extending their lifespan, reducing maintenance costs, and ensuring efficient wind resource utilization. Therefore, it plays a pivotal role in wind farm optimization. In this study, we focus on introducing a novel, data-driven methodology for yaw control.

D. DATA-DRIVEN APPROACHES: TECHNIQUES AND ADVANTAGES

Data-driven approaches, recognized for their efficacy in analyzing and learning from vast datasets, facilitate the creation of accurate models that predict and optimize performance across diverse conditions. Accordingly, several researchers have adopted a data-driven approach in wind farm research [24], [25], [26]. In the context of wind farm control, these techniques utilize the data-driven methods proposed in this study to predict wind performance and determine the optimal yaw angle for generation.

The key component of our data-driven approach is the use of neural networks, which are complex nonlinear relationships between variables. Compared to traditional methods like FLORIS, which rely on physical modeling and computational fluid dynamics simulations, data-driven models can handle large datasets and provide faster computation times. Furthermore, data-driven models can represent complex interactions more accurately that might not be entirely captured by physical models, resulting in more reliable predictions.

In this study, we aim to introduce a data-driven approach to wind farm control. We are confident that our proposed approach will provide a rapid, precise, and effective means of optimizing wind farm performance.

III. A DATA-DRIVEN APPROACH FOR POWER ESTIMATION

A. EXPERIMENTAL SETUP

In the present section, we detail the experimental setup, which includes both the turbine layout and the array of input/output variables used for the neural network's training process. The selected turbine layout for our experiment is a 1×3 configuration, maintaining a turbine-to-turbine distance of 5D. This specific configuration is commonly recognized as a foundational unit in wind farm research, as it effectively captures the wake effects produced by the turbines. In this context, 'D' denotes the rotor diameter of the turbine. For our experiment, the rotor diameter is set at 126m, adhering to the specifications of the NREL 5MW turbine. A graphical depiction of the experimental arrangement can be found in Figure 4.

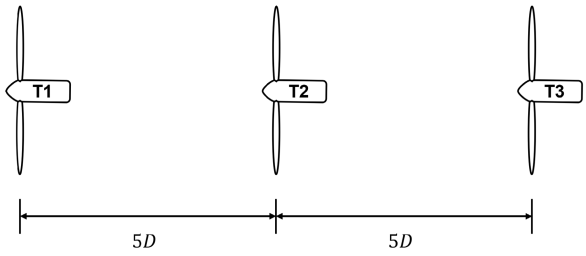

FIGURE 4. Turbine layout used in experiments.

The input/output variables used in neural network learning for the wind farm power estimation were collected within Region 2, where power generation is proportional to the cube of wind speed based on the power curve. The details of the data collection and experimentation are provided in Table 1. Wind speed measurements were conducted within the range of the NREL 5MW turbine's cut-in wind speed of 3 m/s and its rated wind speed of 11.4 m/s. The wind direction was set at 0 degrees north latitude, with angles defined in a clockwise fashion. In this study, the wind direction was limited to a range between 180 and 360 degrees, which was considered appropriate due to the symmetrical nature of the turbine. Furthermore, the yaw angle of the turbine was limited to a specific range to consider the mechanical load on the system.

TABLE 1. Range of input/output variables for power estimation neural network training.

Variables	Range
Wind speed (m/s)	$3 \leq WS \leq 11.4$
Wind direction (deg)	$180 \leq WD \leq 360$
Yaw angles (deg)	$-25 \leq \gamma_{T1}, \gamma_{T2}, \gamma_{T3} \leq 25$

While this study focused on these specific conditions, future investigations have the potential to expand the scope.

B. DATA ACQUISITION AND PREPROCESSING

This section presents the development of the power estimation neural network (PENN) designed to estimate the power of a wind farm based on wind speed, wind direction, and turbine's yaw angle. The training data for PENN is primarily obtained from the FLORIS simulator, consisting of 30,000 input/output pairs. The input variables encompass wind speed, wind direction, and yaw angle, while the output variable corresponds to the power of the wind farm. Each input variable is randomly selected within a predefined range specified in Table 1, and the corresponding power from FLORIS is recorded. This collected training data serves as the input and output variables for training the power estimation neural network.

The quality and uniformity of training data are pivotal for the performance of deep learning models. Rigorous data collection was undertaken to minimize biases and comprehensively capture factors affecting the power of the wind farm. The uniform data distribution ensures effective pattern recognition by the model, enhancing its accuracy on new data. Figure 5 illustrates the uniform distribution of key input variables such as the wind speed, wind direction, and yaw angle.

Using input variables with diverse units and ranges directly can impede neural network training. To remedy this, we utilized z-score normalization, which shifts the data's mean to zero and standardizes the standard deviation to one. This approach not only uniformly scales the data but also facilitates comparisons between data points and diminishes the effects of outliers. The transformation follows the given formula:

$$Z = \frac{X - \mu}{\sigma}. \tag{2}$$

where X represents a data point, μ denotes the mean of all data points, and σ signifies the standard deviation of all data points. The normalized values of the input variables can be verified through the accompanying Figure 6, which demonstrates successful normalization across all variables. This clearly demonstrates that every variable, post-normalization, is bounded within this range.

Finally, to assess the PENN's generalization performance, an additional test dataset of 10,000 samples, distinct from the training data, was collected. This dataset was not used during the PENN's training phase but was employed to evaluate the model's performance after the training process concluded.

TABLE 2. Hyperparameters used in PENN.

Hyperparameters	Range
The number of input layer	5
The number of output layer	1
The number of hidden layer	5
The number of nodes in the hidden layer	64, 256, 512, 256, 32
The number of parameters	288,193
The number of training data	30,000
Activation function	Tanh
Optimizer	Adam
Learning rate	0.00001
Epoch	100,000 (patience=100)
Batch size	30,000
Loss function	MSE

C. MODEL ARCHITECTURE AND TRAINING

The PENN is meticulously designed to capture the inherent complexity of estimating power output based on wind speed, wind direction, and the turbine's yaw. As depicted in Figure 7, the neural network model consists of an input layer, five hidden layers, and an output layer. The input layer takes normalized data, which includes wind speed, wind direction, and the yaw angle of each turbine. In contrast, the output layer predicts the power generated by the wind farm. The hidden layers are fully connected, with 64, 256, 512, 256, and 32 neurons, respectively. Furthermore, the hyperbolic tangent function (tanh) was chosen as the activation function due to its proficiency in learning complex patterns and nonlinear relationships, while also addressing the vanishing gradient problem.

Additionally, the hyperparameters such as learning rate, batch size, and the number of epochs were fine-tuned to optimize the model's performance. These values were established based on a series of preliminary experiments and cross-validation, all aimed at ensuring the robustness and accuracy of the neural network models. The hyperparameters utilized in the PENN are detailed in the Table 2.

The training process commences with random weight initialization. Next, the input data is input into the neural network to compute the output, and the error between the network's output and the output of FLORIS is calculated. The weights are then adjusted through backpropagation of this error through the neural network. This propagation and backpropagation process is repeated for several epochs until the model's prediction error is minimized. The mean squared error (MSE) was used as the loss function. The MSE loss function calculates the mean squared difference between the predicted output and the actual output, providing a comprehensive measure of prediction error. The formula is follows:

$$Loss = \frac{1}{2}\sum_{i=1}^{N}(P_{FLORIS} - P_{PENN})^2. \tag{3}$$

where P_{FLORIS} represents the power from FLORIS and P_{PENN} denotes the power from PENN. The model was trained using the Adam optimizer, an optimization algorithm that

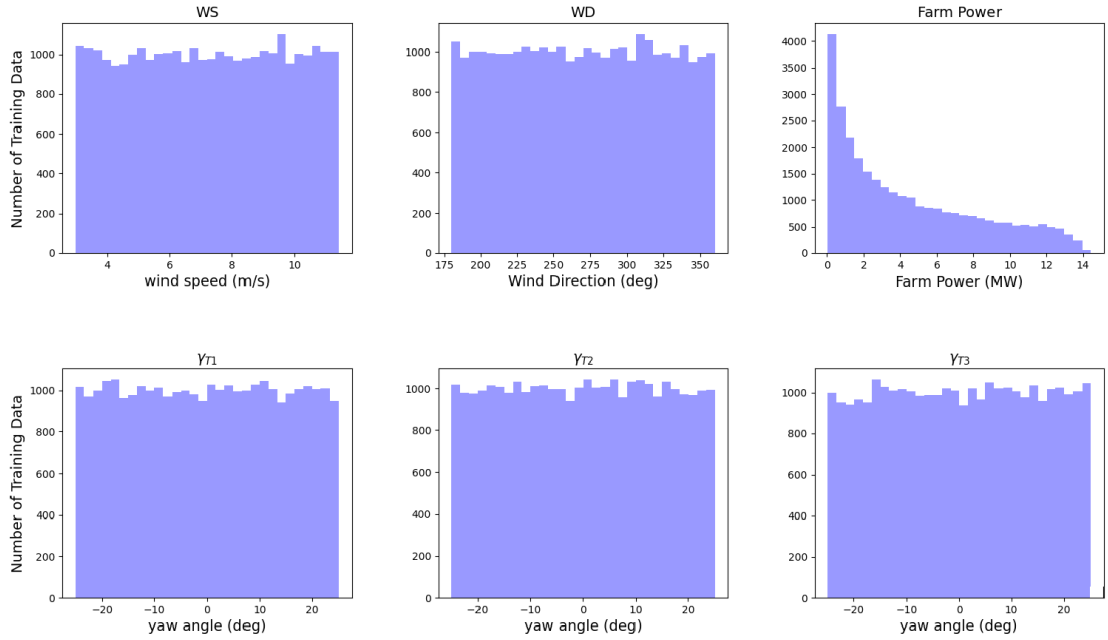

FIGURE 5. Distribution of training dataset.

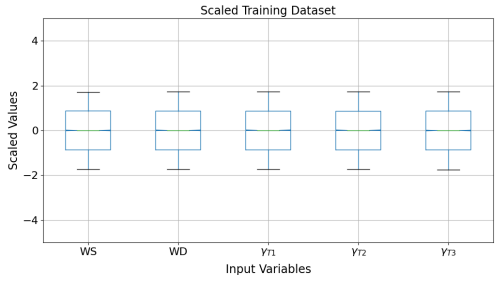

FIGURE 6. The normalized values of the input variables in the training data.

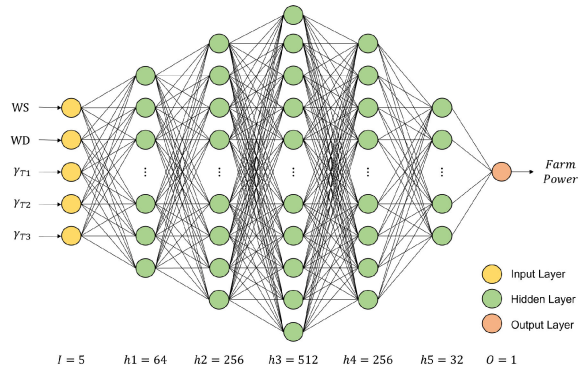

FIGURE 7. Power estimation neural network architecture.

adjusts the learning rate of the model's weights to enhance the speed and performance of the training process. The hyperparameters for the Adam optimizer, such as $\beta_1 \approx 0.9$, $\beta_2 \approx 0.999$, $\epsilon \approx 10^{-8}$, and $\eta = 0.0001$, were used during the training process [27]. Furthermore, the PENN model was trained using 10-fold cross-validation to ensure balanced training, maintain generalization capabilities, and achieve reliable neural network results.

D. MODEL EVALUATION AND VALIDATION

1) EVALUATION METRICS

The evaluation and test phase of the model is considered the most critical phase of this study as it aims to assess the trained PENN's accuracy in predicting the wind farm's power output. In this study, various performance metrics were employed to evaluate the model's performance. The metrics used in the model performance evaluation are as follows.

Mean Absolute Error (MAE):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|P_{FLORIS} - P_{PENN}|. \tag{4}$$

Mean Squared Error (MSE):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(P_{FLORIS} - P_{PENN})^2. \tag{5}$$

Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\sum_{i=1}^{N}\frac{(P_{FLORIS} - P_{PENN})^2}{N}}. \tag{6}$$

Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|P_{FLORIS} - P_{PENN}|}{P_{FLORIS}}. \tag{7}$$

TABLE 3. Performance for testset.

Metrics	MAE	MSE	RMSE	MAPE	R^2
Values	0.000051	0.000081	0.009003	0.339736	0.999994

Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}(P_{FLORIS} - P_{PENN})^2}{\sum_{i=1}(P_{FLORIS} - \bar{P}_{FLORIS})^2}. \quad (8)$$

where N denotes the number of samples in the test dataset, P_{PENN} represents the power predicted by the PENN, P_{FLORIS} denotes the power derived by FLORIS, and $\bar{P}_{FLORIS}$ represents the mean power derived by FLORIS.

In addition, the prediction results of FLORIS and PENN were visually validated using the coefficient of determination, which evaluates the model's performance under various conditions. These assessments affirm the model's generalization and robustness. We have demonstrated that our neural network models ensure consistent reliability across different scenarios.

2) SIMULATION RESULTS AND COMPARISON WITH FLORIS

The test dataset used for the simulation was sourced from data collected in advance during the time of gathering training data. The input wind conditions, namely wind speed and direction for the test dataset, were generated within the range specified in Table 1. This dataset represents a new set of data, distinct from the training and validation datasets, ensuring an unbiased evaluation of the PENN model's performance.

The simulation results indicated that the PENN model outputs power predictions with remarkable accuracy. A comparative analysis with FLORIS revealed a high degree of correlation in the predictions predicted by the PENN model. Key performance indicators for the PENN model, including MAE, MSE, RMSE, MAPE, and R^2, are detailed in Table 3. This table demonstrates that the PENN model performs nearly as well as FLORIS.

The PENN model accurately predicts wind farm power output, as evidenced by its R-squared values nearing 1, suggesting a strong correlation with the power data from FLORIS. Figure 8 visually confirms this. When compared to the established FLORIS model, the PENN model's performance is on par. Thus, these simulation results validate the data-driven approach to power estimation at wind farm.

The results of this study carry profound implications. The PENN model's ability to precisely forecast power output suggests its substantial promise for practical use in wind farms. This opens avenues for enhancing the reliability and efficiency of these power facilities, facilitating improved planning and upkeep, and ensuring optimal power generation. Fundamentally, this research underscores the value of integrating data-driven techniques in the realm of renewable energy, laying a foundation for future progress in the domain.

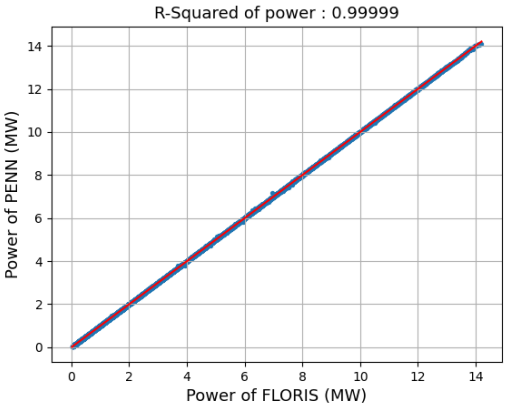

FIGURE 8. Comparison of FLORIS and PENN through coefficient of determination.

IV. A DATA-DRIVEN APPROACH TO DETERMINING OPTIMAL YAW

A. OPTIMAL YAW DETERMINING USING THE POWER ESTIMATION NEURAL NETWORK

1) METHODOLOGY FOR YAW OPTIMIZATION

In this section, we introduce a methodology for determining the optimal yaw angle using a PENN, as depicted in Figure 9. The term 'optimal yaw' denotes the yaw angle of a turbine that maximizes the wind farm's power output. This method incorporates a novel neural network, termed the 'policy neural network'. Its primary function is to compute the optimal yaw angle through backpropagation learning, using input pairs of wind speed and wind direction. This optimization is continuously updated to maximize the power of the wind farm, leveraging the PENN pre-trained in the previous section.

In this methodology, the input variables, namely wind speed and wind direction, are confined to the range specified in Table 1. The yaw angle, which serves as the output of the policy neural network and the input to the PENN, exhibits a discontinuity at a wind direction of 270 degrees. Such discontinuities can complicate data analysis and modeling, and they may also pose challenges in controlling the actual turbine. To address this, the yaw angle of the turbine is restricted to a positive range, specifically between 0 and 25 degrees. This approach effectively mitigates the yaw angle's discontinuities and is anticipated to be crucial for the efficient operation of actual wind turbines.

The policy neural network is designed with wind speed and wind direction as input variables and yaw angle as output variables. This network has a simple architecture, composed of three fully connected layers with 128, 256, and 64 nodes respectively. The output obtained from the policy neural network is then inputted into the PENN, which was developed in a previous section of this dissertation. Consequently, the power produced by the wind farm informs the learning process of the policy neural network, using the prediction from the PENN as a loss function. The optimization of this network is carried out using Adam optimization.

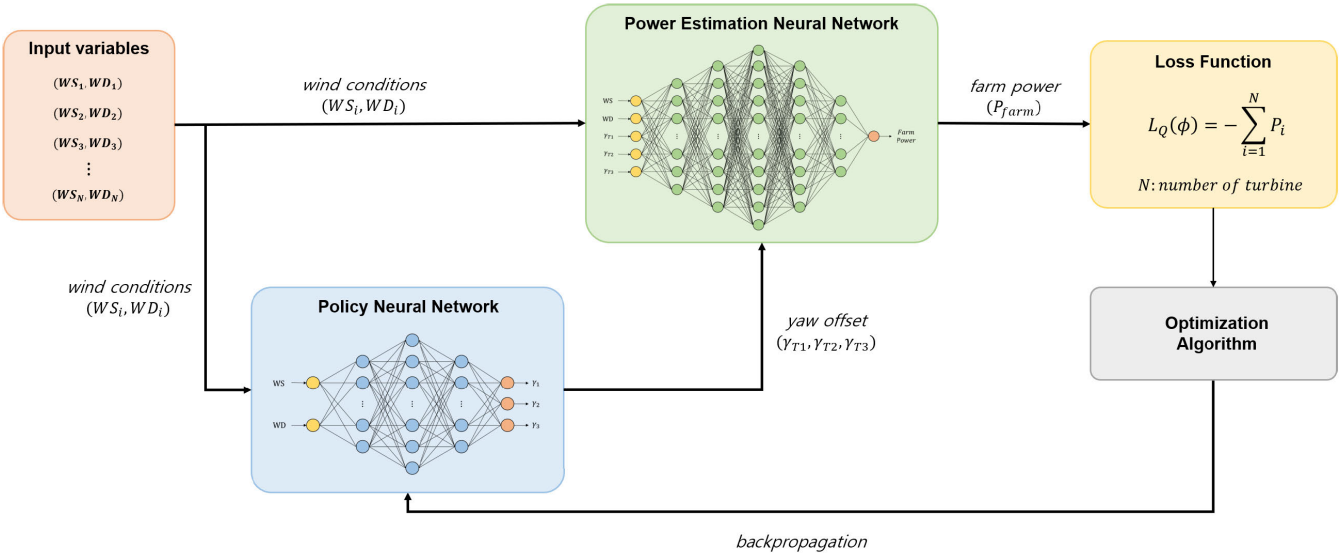

FIGURE 9. Conceptual diagram of determining optimal yaw using a PENN model.

TABLE 4. Test wind conditions for experiments.

Case	Wind speed (m/s)	Wind direction (deg)
1	6	270
2	6	275
3	6	280
4	8	270
5	8	275
6	8	280
7	10	270
8	10	275
9	10	280

2) ANALYSIS OF THE OPTIMIZATION RESULTS

This section details the analysis of the optimization results and compares the performance of the proposed data-driven approach to the serial-refine (SR) method, an optimization technique provided by FLORIS. The optimization process was performed under nine different wind conditions, including both wind speed and wind direction changes as indicated. The nine cases were selected as the wind direction conditions for the best observation of the wake effect in the 1×3 turbine layout. Additionally, these test conditions are summarized in Table 4.

The optimal yaw angles obtained using the SR method in FLORIS, as well as the yaw angles derived from our proposed policy neural network under these nine distinct wind input conditions, are depicted in Figure 10. The figure clearly illustrates that the results from the proposed policy neural network closely align with those produced by the SR method in FLORIS. This underscores the efficacy of the data-driven approach in determining optimal yaw angles, achieving performance comparable to the established FLORIS method.

Moreover, the policy neural network is compared to two other control methods: the greedy control method and the SR method of FLORIS, as illustrated in a power output graph. This graph depicts the power generated by the wind farm, as calculated using the PENN network discussed in a previous section. Figure 11 shows the power comparison among the three yaw control methods. It's clear that both the policy neural network and the SR method of FLORIS consistently outperform the greedy control method in power generation. Additionally, the policy neural network and the SR method display similar power output patterns, underscoring their effectiveness in maximizing power production.

Furthermore, we validated the performance of two other strategies, namely the power growth rate of the SR method and the policy neural network (Policy NN), compared with the greedy control approach. The power growth rate was computed using the following formula:

$$PGR_{SR} = \frac{P_{SR} - P_{greedy}}{P_{greedy}} * 100. \tag{9}$$

$$PGR_{PolicyNN} = \frac{P_{PolicyNN} - P_{greedy}}{P_{greedy}} * 100. \tag{10}$$

where PGR_{SR} means the power growth rate of the SR method as compared to greedy, and $PGR_{PolicyNN}$ represents the power growth rate of the policy neural network as compared to greedy. The simulation results derived from the formula are presented in Table 5.

As shown in Table 5, these findings underscore the near-identical performance of the SR method and the data-driven policy neural network method, emphasizing the potential of data-driven approaches. The results also indicate that yaw control can enhance wind farm power generation. The efficiency of the data-driven method, combined with results comparable to the FLORIS' SR method, suggests its viability as an alternative for specific turbine layouts.

In summary, the data-driven method emerges as a promising approach for determining optimal yaw angles in wind

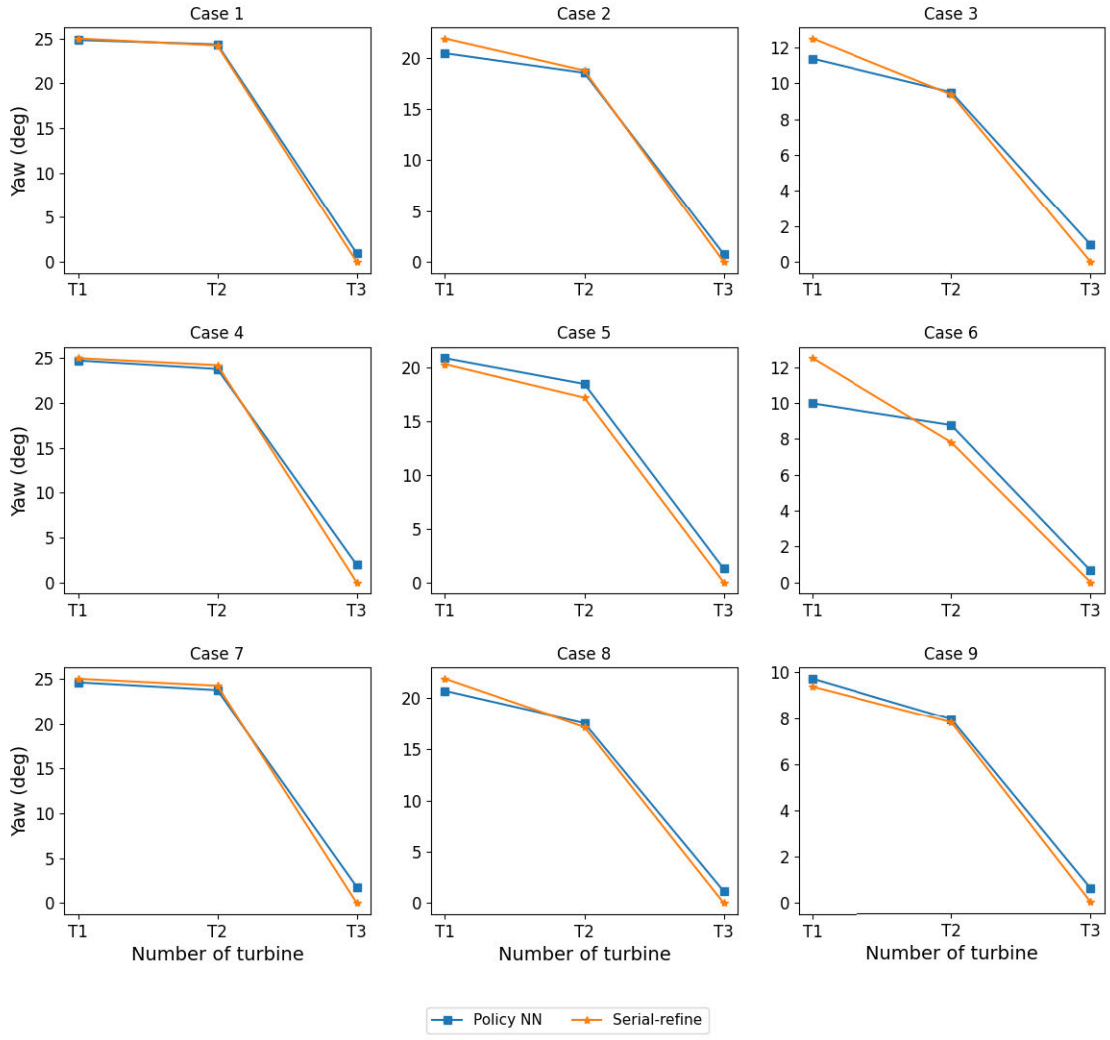

FIGURE 10. Comparison of yaw angle between SR method and policy neural network model.

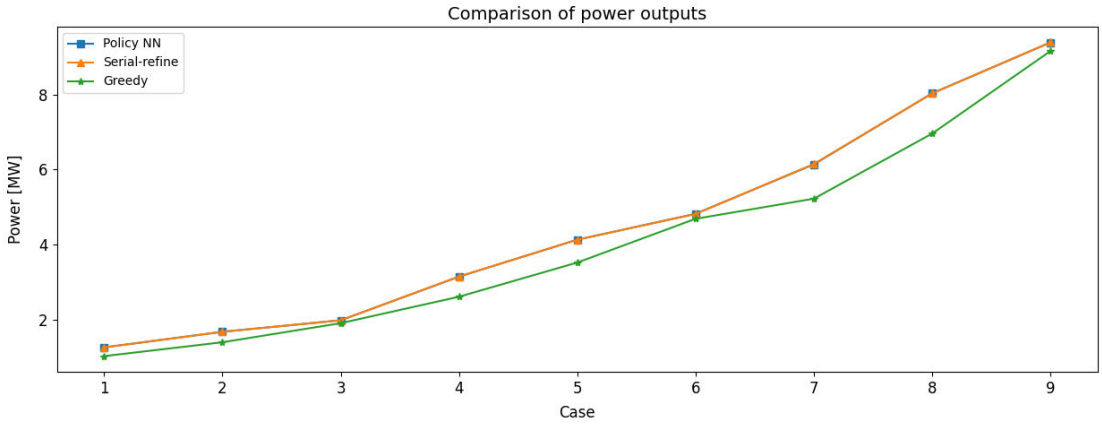

FIGURE 11. Power comparison for the three yaw control methods.

farms, delivering results consistent with the FLORIS' SR method and showing superior computational efficiency. The subsequent chapter will delve into and validate a yaw decision neural network founded on this approach.

TABLE 5. Power growth rate for serial-refine and policy neural network.

Case	PGR_{SR} (%)	$PGR_{PolicyNN}$ (%)
1	22.7981	22.8006
2	20.0745	20.0978
3	4.0454	4.0925
4	20.6998	20.5710
5	17.2714	17.3452
6	2.7451	2.8515
7	17.7205	17.5504
8	15.3877	15.4256
9	2.5302	2.5198

B. DEVELOPMENT OF THE NEURAL NETWORK MODEL TO DETERMINE OPTIMAL YAW

1) DATA ACQUISITION AND PREPROCESSING

The development of our yaw decision neural network (YDNN) model requires meticulous data acquisition and preprocessing. Our training set is composed of 30,000 input-output pairs, with wind speed and direction as inputs, and the turbine's yaw angle as the output. Data is randomly selected based on parameters from Table 1, and the optimal yaw angle is ascertained using the policy neural network algorithm. Preprocessing, especially normalization, is crucial to ensure all input features are standardized. We utilize z-score normalization for both wind speed and direction values. This methodical approach to data management lays a solid foundation for our YDNN, facilitating precise predictions of optimal yaw angles across varied wind conditions.

2) MODEL ARCHITECTURE AND TRAINING

To achieve optimal yaw control for wind turbines, we utilize neural network models, particularly those based on the multilayer perceptron (MLP) architecture. This architecture is renowned for its adaptability and its ability to capture intricate nonlinear relationships within datasets. The neural network's design commences with an input layer comprised of two nodes. These nodes have been strategically crafted to process normalized data points representing wind speed and wind direction. Both factors significantly influence the yaw angle, which, in turn, determines the power generation efficiency of the turbine. The architecture also incorporates five fully connected sequential hidden layers. The number of neurons in these layers, specifically 64, 256, 512, 256, and 32. This ensures an optimal balance between computational efficiency and the model's learning capability. These layers are instrumental in transforming the raw input data by extracting critical features and forwarding them to subsequent layers. Each neuron within these hidden layers employs the hyperbolic tangent (tanh) as its activation function. The tanh function adeptly addresses a prevalent issue in deep neural networks: the vanishing gradient problem, thus guaranteeing a more seamless and efficacious training process. The architecture concludes with an output layer with three nodes. Each of these nodes is designed to predict the optimal yaw angle for a turbine. The predictions from this layer are subsequently compared with actual yaw angles. Throughout the training phase, the model fine-tunes its weights utilizing the mean squared error (MSE) loss function to enhance the accuracy of its yaw angle predictions.

This model is termed the yaw decision neural network (YDNN) given this methodological approach and the architecture. Its design and functionalities underscore our commitment to optimizing yaw control in wind turbines, thereby promoting superior efficiency and sustainability in wind power generation.

$$Loss = \frac{1}{2}\sum_{i=1}^{N}(\gamma_{SR} - \gamma_{YDNN})^2. \tag{11}$$

where γ_{SR} denotes the power of the wind farm for the yaw obtained from the SR technique of FLORIS, and γ_{YDNN} denotes the power of the wind farm for the yaw obtained from the yaw decision neural network.

For weight updates, we use the Adam optimization algorithm in conjunction with backpropagation. The Adam optimizer is renowned for its efficiency because it dynamically adjusts the learning rate during training, taking into account the first and second moments of the gradients. This approach ensures optimal weight adjustments. Training persists either until the loss function attains its minimum value or after a specified number of epochs. The model's generalization capability is then tested on a validation dataset. The chosen architecture, which includes a fully connected MLP, specific activation functions, and the MSE loss function, allows the model to predict optimal yaw angles across different wind conditions.

3) MODEL EVALUATION AND VALIDATION

Following the completion of the YDNN model's training phase, rigorous evaluation and validation become crucial. This step ensures that our model, while theoretically robust, is also practically dependable when addressing real-world challenges. By rigorously evaluating and validating the trained model, we can discern its ability to generalize effectively, especially when presented with novel data.

For an exhaustive evaluation, we tested the model against an independent dataset comprising 10,000 distinct input-output pairs. This dataset, distinct from the training set, offers a thorough and unbiased measure of the model's performance. To provide a detailed quantification of the model's performance, we deployed a suite of evaluation metrics: MAE, MSE, RMSE, and R-squared. Each metric offers unique insights into the model's effectiveness. The results of these evaluations are systematically presented in Table 6, which breaks down the model's performance in predicting yaw angles for individual turbines—represented as γ_{T1}, γ_{T2}, γ_{T3}—and for the entire dataset, represented as γ_{all}.

A review of Table 6 reveals the model's stellar performance. The near-zero values of MAE, MSE, and RMSE signify the model's adeptness at making accurate yaw angle predictions that align closely with actual values. A standout metric is the R-squared value of 0.96, suggesting that the

TABLE 6. Performance for testset (units: deg).

Metrics	MAE	MSE	RMSE	R^2
γ_{T1}	0.007513	0.011238	0.106011	0.999350
γ_{T2}	0.006837	0.013177	0.114791	0.998954
γ_{T3}	0.001913	0.009508	0.097513	0.875433
γ_{all}	0.005421	0.011308	0.106339	0.957912

model can account for an impressive 96% of the variance in the dependent variable. Collectively, these metrics bolster the model's robustness, precision, and reliability, instilling confidence in its applicability across real-world scenarios.

V. APPLICATION OF YAW DECISION NEURAL NETWORK MODEL TO DYNAMIC ENVIRONMENT

A. ACTUATOR MODEL

To regulate the yaw angle of wind turbines, a first-order actuator model is employed. This actuator model is a linear time-invariant (LTI) input/output system, characterized by a first-order differential equation. The transient response characteristics of the actuator model depend solely on the time constant, which determines the rate at which the system responds to changes in the nacelle of the turbine instigated by the commanded yaw angle. The actuator model is mathematically represented by the following first-order differential equation:

$$\theta = \phi - \gamma. \tag{12}$$

$$\tau\dot{\theta} + \theta = \theta_{cmd}. \tag{13}$$

where τ represents the time constant, γ denotes the yaw offset angle derived from the neural network model, ϕ is wind direction, θ is actual turbine's heading angle and θ_{cmd} represents the commanded turbine's heading angle.

In this study, we integrate an actuator model into the yaw decision neural network to achieve a dynamic yaw control response. Given that the turbine's heading angle θ is restricted to 0.3 degrees per second in our study, τ is set to 90.

B. INTEGRATING CONTROLLER INTO DYNAMIC ENVIRONMENTS

We used the FLORIDyn wind farm simulator, known for its ability to simulate dynamic wind conditions, to evaluate our yaw decision neural network (YDNN) model. Given that the YDNN was trained under steady-state conditions, we integrated an actuator model to facilitate dynamic simulations, mirroring real-world wind fluctuations and turbine constraints. As depicted in Figure 12, the YDNN continuously gathers wind and turbine data from FLORIDyn, computes the optimal yaw angles γ, and relays them to the actuator model. This model subsequently modifies the turbine heading angles θ to emulate real-world limitations. These adjusted angles are reintegrated into FLORIDyn, influencing both turbine power output and conditions for downstream turbines. This comprehensive integration allows for a robust assessment of the YDNN in fluctuating wind environments, highlighting its real-world application potential.

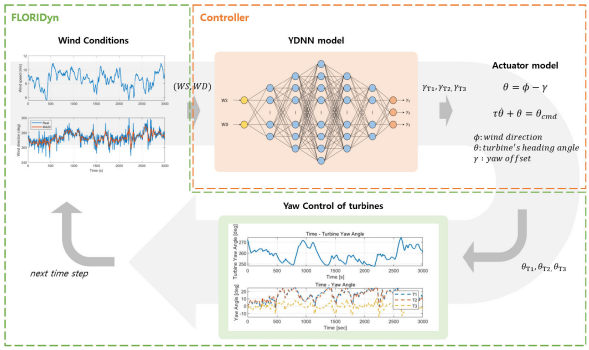

FIGURE 12. YDNN models working in dynamic environments.

C. SIMULATION RESULTS AND ANALYSIS

This study conducted a series of simulations to evaluate the effectiveness of the proposed data-driven yaw determination neural network model, specifically in a 1×3 turbine layout. The wind conditions utilized in the simulations were based on 3,000 seconds of wind field-measured data, collected at 1-second intervals. The wind speed included in the model ranged between the cut-in wind speed and the rated wind speed. The wind direction was simulated within the range where the wake effect could be most effectively observed. The wind conditions used to evaluate the model performance are shown in the Figure 13. By simulating these time-varying wind conditions, we evaluated the performance of our proposed yaw decision neural network model.

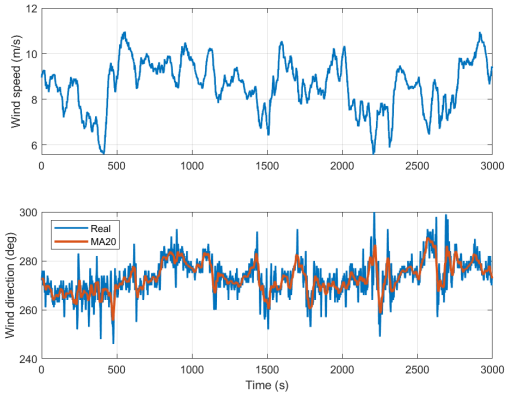

FIGURE 13. Wind conditions used in dynamic simulations.

In each simulation scenario, the output and yaw angle of each turbine can be visualized using Figures 14 and 15. Figure 14 depicts the accumulated power of the wind farm using three different control methods over a span of 3,000 seconds. Both the SR-based LUT and YDNN demonstrate nearly identical trends, and outperforming the greedy control. Figure 15 shows the yaw angle of each turbine under the three control techniques. As indicated by the figure, SR-LUT and YDNN display similar patterns.

The simulation results reveal that the YDNN model effectively increases the wind farm's output by adjusting the

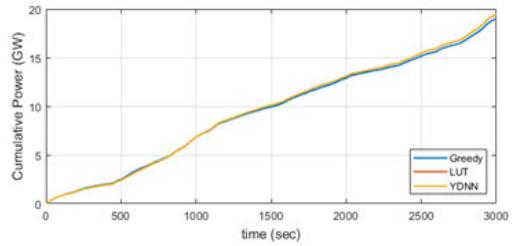

FIGURE 14. Comparison of wind farm power results for dynamic simulation.

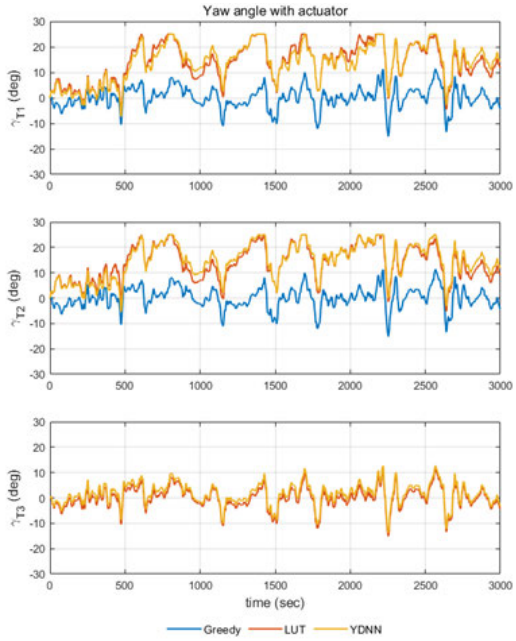

FIGURE 15. Comparison of yaw angle for dynamic simulation.

yaw angle to dynamic wind conditions. This aligns with our predictions, confirming the model's ability to optimize power production. In essence, our proposed YDNN model has proven effective, showing potential to significantly enhance wind farm power output. Future work should focus on refining the model and exploring its practical applications in various wind farm scenarios.

Additionally, the proposed YDNN model was compared to the greedy control and SR-based LUT control methods in terms of cumulative power and power growth rate. The power growth rate is depicted as the cumulative power growth rate for greedy control, as illustrated in the following equations:

$$PGR_{SR-LUT}(\%) = \frac{P_{SR-LUT} - P_{Greedy}}{P_{Greedy}} * 100. \quad (14)$$

$$PGR_{YDNN}(\%) = \frac{P_{YDNN} - P_{Greedy}}{P_{Greedy}} * 100. \quad (15)$$

Table 7 presents the cumulative power growth rates for SR-based LUT control and YDNN. It reveals that both methods produce about 2% more power than greedy control. Notably, the difference in cumulative power between the SR-based LUT control and YDNN is a mere 0.014GW, underscoring their nearly identical ability. This observation further substantiates the equivalent performance of the two methods in power generation optimization.

TABLE 7. Comparison of three control methods.

Control	Greedy	SR-LUT	YDNN
Power(GW)	19.010	19.446	19.432
PGR(%)	0.0	2.293	2.219

Furthermore, the cumulative power growth rates of SR-based LUT control and YDNN are strikingly similar, suggesting their equivalent performance in optimizing power generation.

These findings demonstrate the effectiveness of the proposed YDNN model in wind farm control, outperforming traditional greedy control methods. Furthermore, the similarity in performance between SR-based LUT control and YDNN suggests the potential of our data-driven approach as an alternative to existing control methods.

VI. CONCLUSION AND FUTURE WORK

This paper presents a new data-driven approach for wind farm control as an alternative to the traditional FLORIS simulator. The proposed approach involves the utilization of two neural networks: one is the power estimation neural network based on wind conditions and turbine yaw angle, and another neural network to determine the optimal yaw angle. The PENN serves as a loss function in the optimization process, with the policy neural network deriving the optimal yaw. Using this methodology, we developed a YDNN model that determines the optimal yaw angle, integrating seamlessly with the actuator model in dynamic wind conditions. The uniqueness of this data-driven approach lies in its reliance solely on data. The simulation results were found to be nearly identical to those obtained using the FLORIS simulator. This suggests that the data-driven approach proposed in this paper can replace FLORIS for specific turbine layout.

Future research will emphasize validating this method with real-world data, encompassing larger wind farms and extreme weather conditions, to ensure its practical applicability. Thus, this study underscores the potential and efficacy of a data-driven approach to wind farm control, offering innovative solutions for efficient control and optimization by accurately estimating power output and optimizing yaw.

REFERENCES

[1] H. Dong, J. Xie, and X. Zhao, "Wind farm control technologies: From classical control to reinforcement learning," *Prog. Energy*, vol. 4, no. 3, Jun. 2022, Art. no. 032006.

[2] S. Tamaro and C. L. Bottasso, "A new wind farm active power control strategy to boost tracking margins in high-demand scenarios," in *Proc. Amer. Control Conf. (ACC)*, May 2023, pp. 192–197.

[3] R. Jahantigh, S. M. Esmailifar, and S. A. Sina, "Wind farm control and power curve optimization using induction-based wake model," *Meas. Control*, vol. 56, nos. 9–10, pp. 1751–1763, Jun. 2023.

[4] B. Foloppe, L. Dewitte, and W. Munters, "Exploring cooperation between wind farms: A wake steering optimization study of the Belgian offshore wind farm cluster," *J. Phys., Conf. Ser.*, vol. 2505, Jun. 2023, Art. no. 012055.
[5] N. Kumar, M. G. Mishra, and I. Luthra, "Modelling and simulation of hybrid renewable energy system using real-time simulator," in *Artificial Intelligence and Machine Learning in Satellite Data Processing and Services*, vol. 970. Springer, Jan. 2023, pp. 79–87. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-19-7698-8_9
[6] B. Desalegn, D. Gebeyehu, B. Tamrat, and T. Tadiwose, "Wind energy-harvesting technologies and recent research progresses in wind farm control models," *Frontiers Energy Res.*, vol. 11, Feb. 2023, Art. no. 1124203.
[7] P. Fleming, J. King, C. J. Bay, E. Simley, R. Mudafort, N. Hamilton, A. Farrell, and L. Martinez-Tossas, "Overview of FLORIS updates," *J. Phys., Conf. Ser.*, vol. 1618, no. 2, Sep. 2020, Art. no. 022028.
[8] B. M. Doekemeijer, D. van der Hoek, and J.-W. van Wingerden, "Closed-loop model-based wind farm control using FLORIS under time-varying inflow conditions," *Renew. Energy*, vol. 156, pp. 719–730, Aug. 2020.
[9] F. González-Longatt, P. Wall, and V. Terzija, "Wake effect in wind farm performance: Steady-state and dynamic behavior," *Renew. Energy*, vol. 39, no. 1, pp. 329–338, Mar. 2012.
[10] S. K. Kanev, F. J. Savenije, and W. P. Engels, "Active wake control: An approach to optimize the lifetime operation of wind farms," *Wind Energy*, vol. 21, no. 7, pp. 488–501, Feb. 2018.
[11] C. L. Archer and A. Vasel-Be-Hagh, "Wake steering via yaw control in multi-turbine wind farms: Recommendations based on large-eddy simulation," *Sustain. Energy Technol. Assessments*, vol. 33, pp. 34–43, Jun. 2019.
[12] E. Simley, P. Fleming, J. King, and M. Sinner, "Wake steering wind farm control with preview wind direction information," in *Proc. Amer. Control Conf. (ACC)*, May 2021, pp. 1783–1789.
[13] G.-W. Qian and T. Ishihara, "Wind farm power maximization through wake steering with a new multiple wake model for prediction of turbulence intensity," *Energy*, vol. 220, Apr. 2021, Art. no. 119680.
[14] M. F. Howland, S. K. Lele, and J. O. Dabiri, "Wind farm power optimization through wake steering," *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 29, pp. 14495–14500, Jul. 2019.
[15] Z. Xu, H. Geng, B. Chu, M. Qian, and N. Tan, "Model-free optimization scheme for efficiency improvement of wind farm using decentralized reinforcement learning," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 12103–12108, 2020.
[16] H. Dong and X. Zhao, "Composite experience replay-based deep reinforcement learning with application in wind farm control," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 3, pp. 1281–1295, May 2022.
[17] H. Zhao, J. Zhao, J. Qiu, G. Liang, and Z. Y. Dong, "Cooperative wind farm control with deep reinforcement learning and knowledge-assisted learning," *IEEE Trans. Ind. Informat.*, vol. 16, no. 11, pp. 6912–6921, Nov. 2020.
[18] S. Vijayshankar, P. Stanfel, J. King, E. Spyrou, and K. Johnson, "Deep reinforcement learning for automatic generation control of wind farms," in *Proc. Amer. Control Conf. (ACC)*, May 2021, pp. 1796–1802.
[19] P. Stanfel, K. Johnson, C. J. Bay, and J. King, "A distributed reinforcement learning yaw control approach for wind farm energy capture maximization," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2020, pp. 4065–4070.
[20] J. Xie, H. Dong, X. Zhao, and A. Karcanias, "Wind farm power generation control via double-network-based deep reinforcement learning," *IEEE Trans. Ind. Informat.*, vol. 18, no. 4, pp. 2321–2330, Apr. 2022.
[21] M. Becker, D. Allaerts, and J. W. van Wingerden, "FLORIDyn—A dynamic and flexible framework for real-time wind farm control," *J. Phys., Conf. Ser.*, vol. 2265, May 2022, Art. no. 032103.
[22] M. Kim, H. Lim, and S. Park, "Comparative analysis of wind farm simulators for wind farm control," *Energies*, vol. 16, no. 9, p. 3676, Apr. 2023, doi: 10.3390/en16093676.
[23] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, "Definition of a 5-MW reference wind turbine for offshore system development," National Renew. Energy Lab., Golden, CO, USA, Tech. Rep. NREL/TP-500-38060, Feb. 2009.
[24] N. Zehtabiyan-Rezaie, A. Iosifidis, and M. Abkar, "Data-driven fluid mechanics of wind farms: A review," *J. Renew. Sustain. Energy*, vol. 14, no. 3, May 2022, Art. no. 032703.
[25] C. Adcock and R. N. King, "Data-driven wind farm optimization incorporating effects of turbulence intensity," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 695–700.
[26] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
[27] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, San Diego, CA, USA, 2015, p. 2. [Online]. Available: https://arxiv.org/abs/1412.6980

MINJEONG KIM received the B.S., M.S., and Ph.D. degrees in aerospace engineering from Sejong University, Seoul, South Korea, in 2016, 2018, and 2023, respectively. She is currently a Postdoctoral Researcher with the Flight Dynamics and Control Laboratory, Sejong University. Her primary research interests include the modeling and control of dynamic systems, data analytics, and machine learning.

SUNGSU PARK received the B.S. and M.S. degrees from Seoul National University, and the Ph.D. degree from UC Berkeley. He is currently a Professor in aerospace engineering with Sejong University, Seoul, South Korea. His current research interests include machine learning and optimization applications in guidance, navigation, and control system design.