Kalman-NET-based Estimation of System Dynamics

Aditya Khariwal, Arin Tiwari, Saumya Raj, Siddhant Agarwal, and Shah Krish Sanjay
Under the Supervision of
Prof. Abdul Saleem Mir

Abstract - **For the study of dynamic systems, their state estimation is a crucial task. For the systems that can be represented by known linear Gaussian distributions, the Kalman filter (KF) is the most optimal and computationally easy solution. However, nonlinearities are bound to occur in a practical system, and the knowledge of the underlying state-space (SS) model is not encountered. To study systems and understand their dynamics here, we use the Kalman-Net. This real-time state estimator learns relations to obtain second-order state and observation moments from the data to perform filtering for systems with non-linear dynamics with partial information. We have demonstrated how Kalman net solves the non-linearities in Wind Turbine-driven Doubly Fed Induction Generator (DFIG) systems.**

Index Terms - Kalman Filters, Deep Learning, Recurrent Neural Networks.

I. INTRODUCTION

Kalman filtering is a largely used technique for state estimation in dynamic systems, relying on accurate knowledge of system dynamics. However, in real-world applications, the dynamics of the system may only be partially known or subject to uncertainty. This limitation motivates in the exploration of methods that can adaptively model system dynamics from given available data to enhance the estimation accuracy. Integrating neural networks with Kalman filtering gives us a promising approach to address this problem. Some research has investigated several methods, including using neural networks to learn system dynamics directly from data or combining neural network predictions with traditional Kalman filtering algorithms.

In this paper, we introduce KalmanNet, a novel framework that integrates neural networks with Kalman filtering for state estimation in dynamic systems with partially known dynamics. KalmanNet leverages the predictive capabilities of neural networks to adaptively estimate system dynamics from available data while harnessing the robustness of Kalman filtering for state estimation. By combining these approaches, KalmanNet offers a versatile solution for estimation problems in dynamic systems where the underlying dynamics are only partially known. Experimental results demonstrate the effectiveness of KalmanNet in various real-world scenarios, showcasing its potential for addressing estimation challenges in partially known dynamic systems.[3]

II. SYSTEM MODEL AND PRELIMINARIES

A. Deep Learning (DL)

Deep learning, a branch of machine learning (ML), specializes in the training of neural networks to acquire knowledge from extensive datasets and make observations without explicit programming. At the base of DL are some artificial neural networks (ANNs) made up of layers of interlinked nodes. Each node(neuron) receives an input, hence performs a computation (usually a weighted sum), then applies an activation function to it, and then passes that result to the next layer. DL models usually have multiple layers, allowing them to acquire progressively intricate and abstract representations of the data [1]

B. Recurrent Neural Networks (RNNs)

These are a class of neural networks mainly designed to handle sequential data or data with temporal dependencies. RNNs have a hidden state that takes the information or data about previous inputs in the sequence. This hidden state is updated at each time step and serves as the memory of the network.[2]

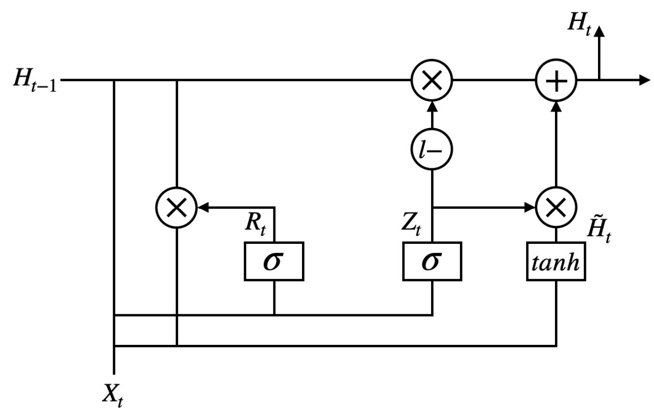

Fig 2.1 Gated Recurrent Unit (GRU)

C. System Model

We study discrete-time state-space models for dynamical systems [3], including nonlinear, Gaussian, and continuous models, represented for each time step as:

$$x_t = f(x_{t-1}) + e_t, \quad e_t \sim \mathcal{N}(0, Q), \quad x_t \in \mathbb{R}^m, \quad (1a)$$

$$y_t = h(x_t) + v_t, \quad v_t \sim \mathcal{N}(0, R), \quad y_t \in \mathbb{R}^n, \quad (1b)$$

In practical applications, the state-evolution model captures system dynamics, while the observation model reflects sensor measurements. Model parameters are often unknown, requiring real-time estimation mechanisms. State-space (SS) models are used for tasks like observation approximation (prediction, imputation, denoising) and hidden state recovery (filtering, smoothing). This paper specifically focuses on filtering—online recovery of the hidden state vector x_t from observed data, crucial for real-time state estimation in dynamic systems.

D. Kalman Filtering:

Kalman filtering is a method employed for estimating the state of a dynamic system in real time. It works in two main steps, prediction and update:

Prediction: The filter uses the system's equations to predict the current state's expected value and covariance matrix. These predictions form the prior estimates before taking into account new observations

Update: Once we get the new measurements, we take the conditional expectation of the expected value and covariance matrix thus updating them. From this, the Kalman gain is calculated. This Kalman gain helps us to identify whether the previous estimates were more reliable or the new measurements.

This recursive process of prediction and update allows. Kalman filtering to efficiently track the state of dynamic systems over time. [4]

E. Kalman-Net:

When partial knowledge about the domain is available, KalmanNet is an advanced architecture for non-linear dynamical systems that carries out real-time state estimation. This combines MB Kalman filtering with recurrent neural networks (RNNs) to efficiently handle model mismatch and non-linearities.

The idea behind the high-level structure of KalmanNet is to learn KG from data and then incorporate this learned KG into the flow of the overall KF. This is necessary because they lack other statistical moments like covariance matrices Q and R that are essential in traditional MB Kalman filtering.

KalmanNet carries out state estimation at every time step in two stages: prediction followed by an update. The current state and observation can be predicted using past data during the prediction stage while only relying on first-order statistics so as not to depend on unavailable higher-order statistics. On the other hand, the update step utilizes newly observed information to compute the posterior of current states including learned KG from RNN.

Characteristic inputs for the RNN:

- Changes between observations ($\Delta \tilde{y}_t$) – this denotes the difference between one observation and another as time progresses.
- Discrepancy in innovation (Δy_t) – this represents the deviation of observed values from their predicted equivalents.
- Forward evolution difference ($\Delta \tilde{x}_t$): Changes of state estimates based on past posterior estimates
- The forward update difference ($\Delta \hat{x}_t$): This can be defined as the change in the posterior state estimate concerning the prior state estimate.

These characteristics give RNN important statistics that are needed for KG computation. KalmanNet has two architectures for RNNs calculating KG:

1. This architecture employs GRU cells with large hidden state sizes to implicitly track the second-order statistical moments needed for computing the Kalman gain (KG) and to accommodate the joint dimensionality of the tracked moments.
2. In this setup, discrete GRU cells are dedicated to representing each specific second-order statistical moment, resulting in a more direct and constrained mapping compared to the non-model-based approach.

We will be using the second architecture to implement Kalman net. It uses separate GRU cells to track specific statistical moments, like mean, variance, and covariance. This approach allows the model to monitor each moment very closely improving the ability of this model for capturing complicated statistical relationships. It also enables the calculator to find the Kalman gain more efficiently by simply transposing each moment hence enhancing its estimate accuracy. The supervised learning with labeled data and state estimate-based loss function is used in training this model thereby enabling end-to-end learning. Efficient training methods like mini-batch stochastic gradient descent and backpropagation through time secure a steady, effective training process.

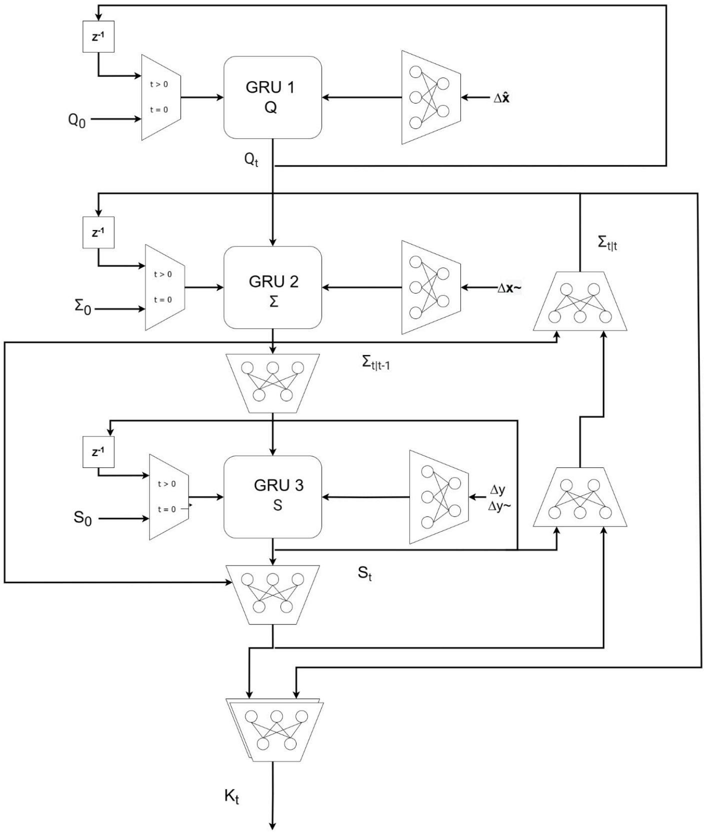

Fig 2.2 Internal architecture of KalmanNet showing GRU units with respective inputs and outputs

III Wind Turbine driven DFIG systems with Backlash Non-Linearity

A. Backlash in Gear Trains

Backlash nonlinearity is essentially the gap between the teeth of the driven gear and the driving gear [5]. This gap is critical in gear train systems as it provides space for lubrication and allows for the expansion of metal. It causes the gears to not be in contact for some time during operation. In wind turbine (WT) driven generator systems, which rely on a gearbox for power conversion, modeling backlash is crucial for analyzing system dynamics. The backlash may increase over time due to wear and tear of the gear train [6]. We intend to model this backlash using Kalman Net.

Fig. 3.1a depicts a WT-driven DFIG system. In this configuration, the stator is linked directly to the grid, and the rotor is linked to the grid via two converters. The rotor voltage of the DFIG has been adjusted to operate at its maximum power point characteristics at an average wind speed of 10m/s in this investigation.

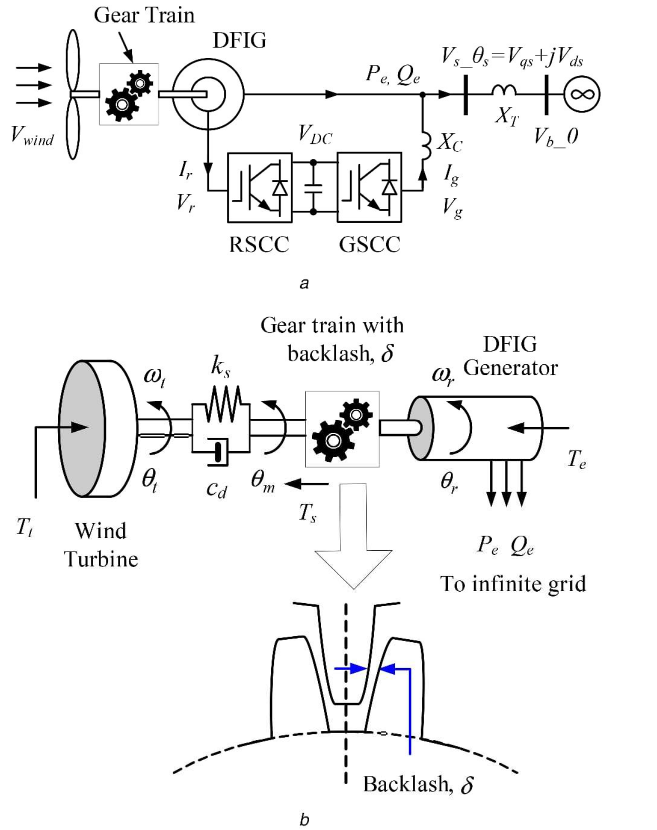

Fig 3.1 DFIG based wind turbine system
(a) Schematic of DFIG connected to the infinite grid (b) Backlash between the gears

The DFIG's rotor voltage has been set in such a way that it operates at the maximum power point. The average wind speed of 10 m/s in this research. The research concentrates on modeling and estimating backlash, so the controller is omitted for simplicity [7].

A typical backlash is shown in Fig. 3.1b. A gear train allows smooth transmission of power from a low-speed turbine shaft to a high-speed generator shaft. The main driving torque of the generator is T_t. It depends on the input ie, the wind energy. The power balance of the system is governed by the underlying equations 1-7

$$\dot{w}_t = \frac{1}{2H_t}(T_t - T_s) \tag{2}$$

$$\dot{w}_r = \frac{1}{2H_g}(T_s - T_e) \tag{3}$$

$$\dot{\theta}_t = w_B(w_t - w_s) \tag{4}$$

$$\dot{\theta}_r = w_B(w_r - w_s) \tag{5}$$

where w_t, w_r and w_B represents the turbine, generator, and base electrical angular speed, respectively; H_g and H_t are the generator and turbine inertia constant, respectively; θ_r and θ_t are the turbine and generator shaft angular positions, respectively;

T_t, T_s and T_e are the turbine, shaft, and electrical generator torque, respectively, as specified below

$$T_t = \frac{P_t}{w_t} \tag{6}$$

$$T_s = \gamma(k_s,\ c_d,\ \delta,\ \theta_{tw}) \tag{7}$$

$$T_e = \frac{1}{w_s}\left(E'_q\, I_{qs}\ +\ E'_d\, I_{ds}\right) \tag{8}$$

Here P_t is the input to the turbine. The input to the turbine depends on the wind speed and generator speed. Additionally, c_d and k_s denote the damping and mechanical stiffness of the low-speed turbine shaft while θ_{tw} signifies the twist angle of the turbine shaft. E'_q, E'_d, I_{qs} and I_{ds} are the stator voltages and currents. The shaft torque T_s, corresponds to power transmission from the low-speed shaft to the high-speed shaft, and it is characterized by a non-linear function of γ, encompassing backlash.

B. Modelling of gear train backlash

The input turbine torque is equal and opposite to the torque applied by the DFIG, at a steady state, keeping the teeth in contact. A decrease in torque due to a change in wind speed will cause the gears to lose contact. This can be modeled as

$$\theta_d = \theta_t - \theta_r \tag{9}$$

$$\theta_b = \theta_m - \theta_r \tag{10}$$

$$\theta_{tw} = \theta_t - \theta_m \tag{11}$$

Also, twist angle can be written as

$$\theta_{tw} = \theta_b - \theta_d \tag{12}$$

$$\dot{\theta_{tw}} = \dot{\theta_b} - \dot{\theta_d} \tag{13}$$

From Fig 3.1 b there can be 3 possible cases of θ_b and hence 3 possible functions for shaft torque T_s.

When $|\theta_b| \leq \delta$ shaft torque is zero hence

$$T_s = 0\ \Rightarrow k_s\, \theta_{tw} + c_d\, \dot{\theta_{tw}} = 0 \tag{14}$$

$$T_s = 0\ \Rightarrow k_s\,(\theta_d - \theta_b) + c_d\big(\dot{\theta_d} - \dot{\theta_b}\big) = 0$$

When $\theta_b = +\delta\ \Rightarrow \dot{\theta_b} = 0$ the shaft torque is positive

$$T_s > 0\ \Rightarrow\ T_s = k_s(\theta_d - \delta) + c_d\dot{\theta}_d > 0 \tag{15}$$

When $\theta_b = -\delta\ \Rightarrow \dot{\theta_b} = 0$ the shaft torque is negative

$$T_s < 0\ \Rightarrow\ T_s = k_s(\theta_d - \delta) + c_d\dot{\theta}_d < 0 \tag{16}$$

When $|\theta_b| \leq \delta$ shaft torque is zero and hence the dynamics of the generator are governed by electrical quantities thus θ_b need not be modeled separately.

C. Modeling DFIG and infinite grid connection

We are modeling the generator in the d-q frame [8]. Considering base speed as w_s as 1 pu.

$$k_1\dot{I_{qs}} = -R_1 I_{qs} + L'_s I_{ds} + \omega_r E'_q - \frac{E'_d}{T_r} - V_{qs} + k_m V_{qr} \tag{17}$$

$$k_1\dot{I_{ds}} = -R_1 I_{ds} + L'_s I_{qs} + \omega_r E'_d - \frac{E'_q}{T_r} - V_{ds} + k_m V_{dr} \tag{18}$$

$$\frac{1}{w_B}\dot{E'_q} = R_2 I_{ds} - \frac{E'_q}{T_r} + (1 - w_r)E'_d - k_m V_{dr} \tag{19}$$

$$\frac{1}{w_B}\dot{E'_d} = R_2 I_{qs} - \frac{E'_d}{T_r} + (1 - w_r)E'_q - k_m V_{qr} \tag{20}$$

where $k_1 = \frac{L'_s}{w_B}$, $T_r = \frac{L_r}{R_r}$, $L'_s = L_s - \frac{L_m^2}{L_r}$, $R_1 = R_2 + R_s$, $k_m = \frac{L_m}{L_r}$ and $R_2 = k_m^2 R_r$. R_r, R_s are stator and rotor resistances, respectively; Ls, L_r and L_m, are stator, rotor and mutual inductances, respectively, and V_{qs}, V_{ds}, V_{dr} and V_{qr} are q-axis and d-axis stator and rotor voltages, respectively.

DFIG, connected to the infinite grid by a transmission line and transformer has the bus voltage, V_b. The net impedance of the transmission line and transformer is X_t. The line current I_a, flows through the line corresponding to the power flow. The line current and stator voltage, in terms of DFIG states, are modeled as

$$I_a = I_{qs}\ +\ jI_{ds} \tag{21}$$

$$V_{qs} = real(V_b\ +\ jI_a X_t) \tag{22}$$

$$V_{ds} = imag(V_b\ +\ jI_a X_t) \tag{23}$$

The total active and reactive power delivered from DFIG to the grid can be given as

$$P_e = V_{ds}I_{qs} +\ V_{qs}I_{qs} + V_{dr}I_{dr} + V_{qr}I_{qr} \tag{24}$$

$$Q_e = V_{ds}I_{qs} +\ V_{qs}I_{ds} + V_{dr}I_{qr} + V_{qr}I_{dr} \tag{25}$$

Here I_{dr} and I_{qr} are the direct and quadrature axis rotor currents

$$I_{qr}\ =\ -\frac{E'_{ds}}{w_s L_m} - k_m I_{qs} \tag{26}$$

$$I_{dr}\ =\ -\frac{E'_{qs}}{w_s L_m} - k_m I_{ds} \tag{27}$$

A set of nonlinear equations used to represent the DFIG with backlash is given below

$$\dot{x(t)} = f\big(x(t), u(t)\big) + w(t) \tag{28}$$

$$y(t) = h\big(x(t)\big) + v(t) \tag{29}$$

where f and h are the state and observation function of the system. Process noise, w, and measurement noise, v have covariance, Q and R, respectively. Here x represents the state u signifies input, and y symbolizes the output. This is shown below

$$x = \left[I_{qs}\ I_{ds}\ E'_q\ E'_d\ w_r\ w_t\ \theta_r\ \theta_t\right]^{\mathrm{T}} \tag{30}$$

$$u = [V_w] \tag{31}$$

$$y = [w_r\ P_e\ Q_e]^{\mathrm{T}} \tag{32}$$

which may further be represented in the discrete-time form as

$$x_k = f(x_{k-1}, u_{k-1}) + w_k \tag{33}$$

$$y_k = h(x_k, u_k) + v_k \tag{34}$$

IV RESULTS AND CONCLUSION

A. Results

For highly nonlinear state functions first-order ODE solution will not be stable. For this reason, we use the Dormand-Prince method for the fifth order of ODE solution to obtain the prior expected value of the state.

The KalmanNet was trained on the simulation with an initial wind power of 1 pu and then suddenly dropped to 0 pu at 5 second time stamp and then revamped to 1 pu at 6 second time stamp

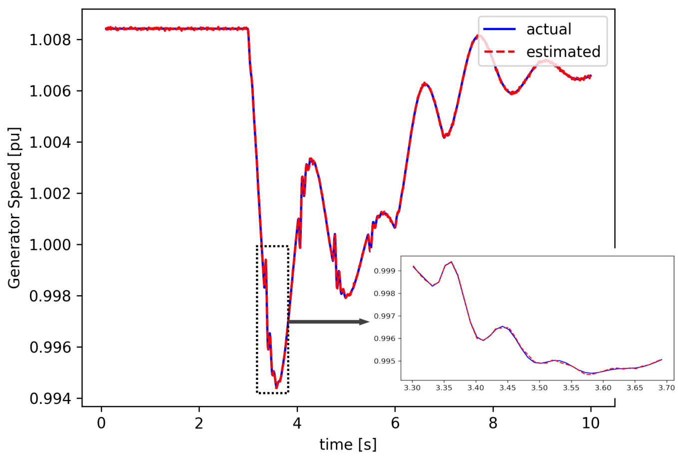

Fig 4.1 Generator Speed w_r

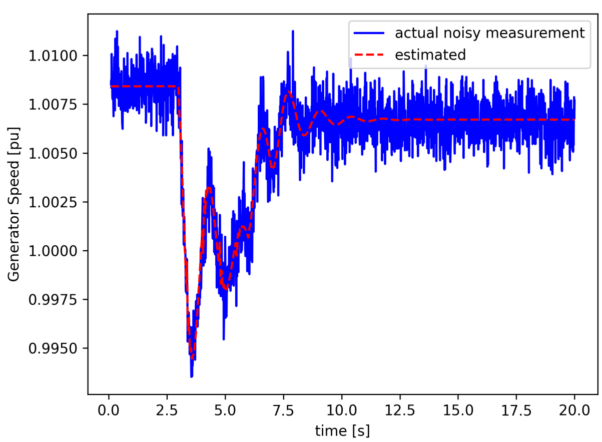

Fig 4.2 Generator Speed w_r noisy output vs estimation

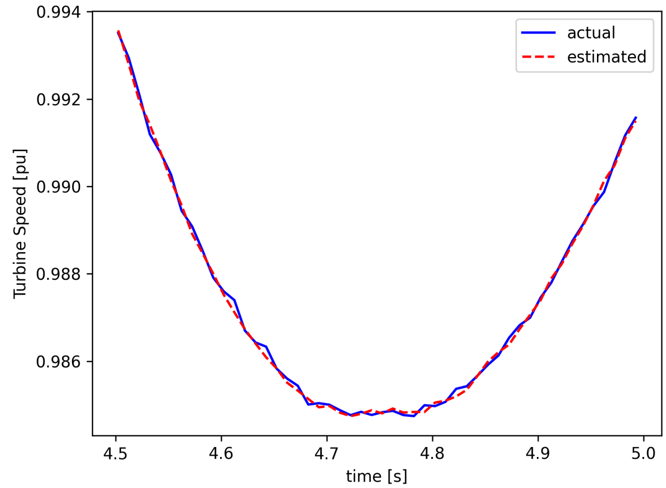

Fig 4.3 Turbine Speed w_t

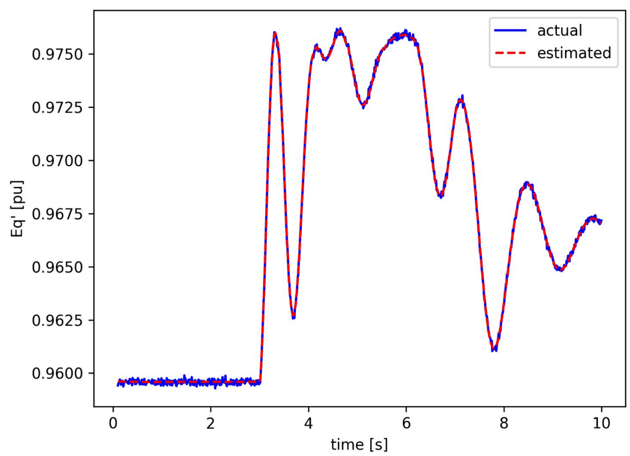

Fig 4.4 q axis voltage E_q'

the test simulation, the wind power was set at 1.0, 0.0, 0.5, 0.8 pu at timestamps 0, 3, 6, and 7 seconds respectively.

B. Conclusion

In this study, we employed a Kalman net for state estimation in a wind turbine system featuring a doubly-fed induction generator (DFIG) and modeled with gear backlash. The Kalman net, a neural network-based variant of the Kalman filter, proved to be a valuable tool for accurately estimating the system's state variables under various operating conditions, despite the challenges posed by gear backlash.

Our results demonstrate the effectiveness of the Kalman net in providing robust and accurate state estimation for the DFIG-wind turbine system. By leveraging the capabilities of the Kalman net, we were able to address the nonlinearities introduced by gear backlash and obtain reliable state estimates crucial for the system's control and monitoring.

V REFERENCES

[1] McCullum, N. (2021) Deep Learning Neural Networks explained in plain English, freeCodeCamp.org. Available at: https://www.freecodecamp.org/news/deep-learning-neural-networks-explained-in-plain-english/.

[2] Recurrent Neural Networks - Dive into Deep Learning 1.0.3 documentation. Available at: https://d2l.ai/chapter_recurrent-neural-networks/rnn.html

[3] Shlezinger, N., Sloun, R.J.G. van and Eldar, Y.C. (1970) Kalmannet: Data-driven Kalman filtering, Eindhoven University of Technology research portal. Available at: https://research.tue.nl/en/publications/kalmannet-data-driven-kalman-filtering.

[4] Becker, A. (2023) Kilman Filter: From the ground, up. KilmanFilter.NET.

[5] Describing function of common nonlinearities. Available at: https://erodesengunthar.ac.in/wp-content/uploads/2019/04/Describing-function-of-common-nonlinearities.doc

[6] Siyu, C., Jinyuan, T., Caiwang, L., et al.: 'Nonlinear dynamic characteristics of geared rotor bearing systems with dynamic backlash and friction', Mech. Mach. Theory, 2011, 4

[7]Anaya-Lara, O., Jenkins, N., Ekanayake, J., et al.: 'Wind energy generation: modelling and control' (John Wiley and Sons Ltd, England, 2009)

[8]'Eigenvalue sensitivity analysis based optimized control of wind energy conversion systems'. IEEE Innovative Smart Grid Technologies – Asia (ISGT-Asia), Melbourne, VIC, 2016, pp. 647–652