Received: 6 October 2022 | Revised: 5 December 2022 | Accepted: 13 December 2022

DOI: 10.1002/ctpp.202200153

ORIGINAL ARTICLE

Contributions to Plasma Physics

Real-time estimation of the electron temperature profile in DIII-D by leveraging neural-network surrogate models

Shira Morosohk | **Eugenio Schuster**

Plasma Control Group, Lehigh University, Bethlehem, Pennsylvania, USA

Correspondence
Shira Morosohk, Plasma Control Group, Lehigh University, Bethlehem, PA, USA.
Email: morosohk@lehigh.edu

Funding information
National Science Foundation, Grant/Award Number: 1842163; U.S. Department of Energy, Grant/Award Numbers: DE- FC02-04ER54698, DE-SC0010661

Abstract

Control of both the magnitude and the shape of tokamak profiles will be necessary to achieve stable, high-performance plasmas. In order to reject disturbances in real time, feedback-control algorithms rely on accurate real-time knowledge of the plasma state. When diagnostics alone are insufficient, because they are limited in number or their measurements are too noisy, observers can be used to combine diagnostic data with a response model to provide a better estimation of different plasma properties. An observer has been developed to estimate the electron temperature profile in real time using both diagnostic data from the Thomson scattering system and a model based on the electron heat transport equation describing the evolution of the electron temperature profile. Neural network surrogate models are leveraged to help improve the overall model prediction while staying within computation time constraints for real-time use. The observer algorithm is shown in offline tests to produce smooth profiles that are consistent with both the diagnostic data and the electron heat transport equation. When implemented into the real-time plasma control system, this observer will provide valuable information on the electron temperature profile to many potential feedback-control applications.

KEYWORDS

electron temperature profile, neural network, real-time estimation, state observer, surrogate models

1 | INTRODUCTION

The spatial variation of kinetic plasma properties such as temperature and density need to be carefully controlled to achieve high-performance, magnetohydrodynamically stable tokamak plasmas. This will be even more crucial for burning plasmas, where the reactor power and gain will need to be tightly regulated. In order to achieve this level of control, feedback techniques are useful for their ability to react to real-time disturbances and to return the plasma to its target state. However, many approaches to feedback control assume that the full state of the system can be measured. This allows a tracking error between the measured state and the target (or desired) state to be calculated in real time and used by the feedback control algorithm to determine the corrective actuation that is needed to reduce such error to as close to zero as possible. However, in practical applications, there are often limitations on how much of the system state can be measured in real-time. These limitations could arise from the diagnostics not having a fine enough spatial resolution, the relevant diagnostics not being available in real-time, or the diagnostics being too noisy to be fed directly to the feedback control algorithm.

Thomson scattering[1] is one of the most common diagnostics used to measure electron temperature and density. It uses laser light that is scattered by electrons, causing a change in both amplitude and frequency. By measuring the scattered light, electron density can be derived from the change in amplitude, and electron temperature can be derived from the change in frequency. While extremely useful, this diagnostic does contain a significant amount of noise. In real time, it does not produce a smooth enough profile to be useful for many feedback controllers. Offline, it is common to manually fit a smooth profile that visually fits the Thomson scattering data[2]; this approach requires significant physicist input. Work has also been done on automatically fitting the diagnostic data to an assigned shape such as a hyperbolic tangent function, with no physicist input.[3] Both of these approaches are dependent on the assumption of the shape of the fitting function. Gaussian process regression is a statistical tool that is also being explored for the purpose of profile fitting,[4] but there is currently no widely available tool that uses this method for DIII-D. All of these approaches are heavily reliant on just the diagnostic data and do not take into account much of the known physics of how the profile behaves.

Another way to determine the state of the system is to use an observer. In response to the issues presented by inadequate measurement capabilities, the Luenberger observer[5] was developed to estimate the unknown state of a linear system from a combination of a limited number of measurements and a model of the system. The use of both measurements and a model allows the observer, also known as an estimator, to handle both noises in the measured data and imperfections of the model. This approach grew into an entire subset of control theory known as state estimation, and now includes a huge variety of observer algorithms for both linear and nonlinear systems. State observers have been designed to estimate both kinetic[6–8] and magnetic[9,10] profiles in tokamaks. A variety of nonlinear estimation techniques have been applied, including the extended Kalman filter.[11] The Kalman filter is a well-known approach to determining the optimal estimator gain for linear systems; the extended Kalman filter is a nonlinear extension of the Kalman filter algorithm.

The observer designs cited above use a combination of physics-based and empirical models for the model component of the observer algorithm. There are three major criteria that models should be graded on in this context: accuracy, speed, and applicability across a wide variety of scenarios. First-principles models usually demonstrate good accuracy over a variety of scenarios, but often do not meet the speed criterion to be useful in real time. Empirical models tend to be fast enough but may only meet the accuracy criterion for the specific plasma scenario they are tuned to. Tokamak profiles are often modelled by transport equations that describe their evolution over time. For some components of transport equations, physics-based models are available that are simple enough to run in real time while also having high levels of accuracy. However, other necessary components of the model, especially auxiliary heating and current drive sources and transport coefficients, obey dynamics that are complicated enough that the models most commonly used for physics applications do not run fast enough to be useful for control applications. This is a significant issue when estimating kinetic profiles, which are highly sensitive to both external sources and transport. Machine learning offers the ability to develop models that can meet all three criteria. In tokamak applications, machine learning models have the potential to offer significant advantages for real-time operation. Because of this, a significant effort has been made to develop machine-learning surrogate models for various time-intensive physics codes.[12–14] However, there has been very little research using machine learning-based models integrated with traditional observer architecture for profile estimation in tokamaks, although it has been used for estimation in unrelated applications.[15,16]

In this work, an observer has been developed to estimate the electron temperature profile in a tokamak. The estimator uses a nonlinear model based on the electron heat transport equation that contains both analytical and machine learning-based components. Surrogate models for the Monte Carlo neutral beam code NUBEAM and for the turbulent transport code MMM have been developed for DIII-D, known as NubeamNet[17] and MMMnet,[18] respectively. NubeamNet is used here to calculate the neutral beam heating source, and MMMnet is used to calculate the electron thermal diffusivity. The diagnostic data used by the observer comes from the Thomson scattering system. The extended Kalman filter algorithm[11] is used to calculate an optimal gain for the nonlinear system. The goal of this work is to provide automatic, real-time estimations of the electron temperature profile that are consistent with both the known physics of the electron temperature profile evolution and the Thomson scattering data.

The rest of the paper is organized as follows. Section 2 introduces the nonlinear model used by the observer, including the two neural network components. Section 3 describes the extended Kalman filter algorithm and the way it is applied to this work. Section 4 shows the results of testing the observer offline. Section 5 presents conclusions and future work.

2 | MODEL

The model used by the proposed observer is based on the nonlinear 1D parabolic PDE that describes the evolution of the electron temperature profile; this is referred to as the electron heat transport equation.[19] The components of this equation are calculated using a combination of neural networks and analytical models. In order to facilitate calculation, the PDE is reduced to a system of ODEs through spatial discretization, and then to a system of equations through temporal discretization. This system of equations can then be used at each time step to predict the electron temperature profile at the next time step.

2.1 | Electron heat transport equation

The electron heat transport equation is given by

$$\begin{aligned}\frac{3}{2}\frac{\partial}{\partial t}\left[n_e T_e\right] &= \frac{1}{\rho_b^2 \widehat{H}}\frac{1}{\widehat{\rho}}\frac{\partial}{\partial \widehat{\rho}}\left[\widehat{\rho}\frac{\widehat{G}\widehat{H}^2}{\widehat{F}}\left(\chi_e n_e \frac{\partial T_e}{\partial \widehat{\rho}}\right)\right] + Q_e,\\ \frac{\partial T_e}{\partial \widehat{\rho}}(0,t) &= 0,\ \frac{\partial T_e}{\partial t}\left(\widehat{\rho}_{bdry}, t\right) = 0.\end{aligned} \tag{1}$$

The mean effective minor radius of the plasma is defined as $\rho = \sqrt{\pi B_{\phi,0}\Phi}$, where $B_{\phi,0}$ is the reference magnetic field at the geometric major radius R_0, and Φ is the toroidal magnetic flux. The scalar ρ_b is defined as the value of ρ at the boundary of the plasma. The normalized mean effective minor radius is defined as $\widehat{\rho} = \rho/\rho_b$. The model introduced here is only valid for the core region of the plasma, not the pedestal region. The boundary condition is thus set at the location $\widehat{\rho} = \widehat{\rho}_{bdry}$, which is chosen to be constant and less than the $\widehat{\rho}$ value of the pedestal location. The profiles $\widehat{F}$, $\widehat{G}$, and $\widehat{H}$ are geometrical factors related to the plasma equilibrium,[20] which is assumed prescribed in this control-oriented model. The equation for electron heating is written as

$$Q_e(\widehat{\rho},t) = Q_{ohm}(\widehat{\rho},t) + Q_{nbi}(\widehat{\rho},t) + Q_{ec}(\widehat{\rho},t) - Q_{ei}(\widehat{\rho},t) - Q_{rad}(\widehat{\rho},t), \tag{2}$$

and contains terms for Ohmic heating (Q_{ohm}), neutral beam heating (Q_{nbi}), electron cyclotron resonance heating (Q_{ec}), collision-driven electron-ion exchange heating (Q_{ei}), and radiative heat loss (Q_{rad}). Section 2.2 describes the two neural networks, NubeamNet and MMMnet that are used to calculate Q_{nbi} and χ_e, respectively. These are two terms that can be very difficult to calculate accurately using traditional real-time-capable models. Section 2.3 introduces analytical models for the electron density and for the other components of electron heating.

Equation (1) is rewritten to separate the different derivatives of T_e as

$$\frac{\partial T_e}{\partial t} = f_1(\widehat{\rho},t)\frac{\partial^2 T_e}{\partial \widehat{\rho}^2} + f_2(\widehat{\rho},t)\frac{\partial T_e}{\partial \widehat{\rho}} + f_3(\widehat{\rho},t)T_e(\widehat{\rho},t) + f_4(\widehat{\rho},t), \tag{3}$$

where

$$\begin{aligned} f_1(\widehat{\rho},t) &= \frac{2}{3}\frac{\widehat{G}\widehat{H}}{\rho_b^2\widehat{F}}\chi_e(\widehat{\rho},t),\\ f_2(\widehat{\rho},t) &= \frac{2}{3}\frac{\widehat{G}\widehat{H}}{\rho_b^2\widehat{F}}\left[\frac{\chi_e(\widehat{\rho},t)}{n_e(\widehat{\rho},t)}\frac{\partial n_e}{\partial \widehat{\rho}} + \frac{\partial \chi_e}{\partial \widehat{\rho}} + \frac{\chi_e}{\widehat{\rho}}\right] + \frac{\chi_e}{\rho_b^2\widehat{H}}\frac{\partial}{\partial \widehat{\rho}}\left(\frac{\widehat{G}\widehat{H}^2}{\widehat{F}}\right),\\ f_3(\widehat{\rho},t) &= \frac{-1}{n_e(\widehat{\rho},t)}\frac{\partial n_e}{\partial t},\\ f_4(\widehat{\rho},t) &= \frac{2Q_e(\widehat{\rho},t)}{3n_e(\widehat{\rho},t)}. \end{aligned} \tag{4}$$

This form of the equation sets up the discretization in Section 2.4.

2.2 | Neural networks

2.2.1 | NubeamNet

Neutral beam heating has a very significant effect on the evolution of the electron temperature profile and is also one of the most difficult components of Equation (1) to calculate analytically. NUBEAM[21] is a commonly used code that calculates the effects of neutral beam injection on the plasma using a Monte Carlo approach. Its outputs include profiles of the heat, current, and torque deposition from neutral beam injection, as well as scalar quantities such as neutron rate and shine through. NUBEAM has been extensively validated and is routinely used across many different tokamaks. However, it can take on the order of minutes to calculate these outputs for a single time step, making it impractical to use in real-time applications. Because of this, a neural network surrogate model for NUBEAM was developed, known as NubeamNet,[17] that can produce similar results on the order of a millisecond.

NubeamNet was originally developed for NSTX-U[12] to replicate the calculations of NUBEAM with a faster inference time, and a new version was later generated for DIII-D.[17] The DIII-D version of NubeamNet was trained on nearly 200,000 time slices of NUBEAM input and output data, including profiles. Each profile was reduced to a set of scalar values using principal component analysis (PCA).[22] In this technique, a profile is reduced to a linear combination of a set of basis functions; a specific value of the profile can then be defined by the coefficients of each basis function. This allowed the multi-layer perceptron (MLP) structure[23] (see Figure 1), which typically only accepts scalar data, to effectively handle the spatially-varying profile data. The PCA technique was applied to inputs such as electron temperature, electron density, and safety factory profiles, among others. The network was trained on 80% of the data, while 10% was used to determine the optimal hyperparameters, and the remaining 10% was used for testing. The MLP used here consists of two hidden layers, each containing 75 neurons; these hyperparameters were determined by a grid search. In addition, five separate neural networks were trained in parallel using the exact same hyperparameters and training data. The only difference between the networks was in the random initialization of the weights. The total output of the network is taken to be the average of the five predictions, and the standard deviation of the five predictions gives some quantification of the uncertainty of the network.

In this work, NubeamNet is used to calculate the beam heating source in Equation (1). NubeamNet predictions of this output compared to NUBEAM are shown in Figure 2. Figure 2a shows the evolution of the NUBEAM and NubeamNet predictions of one point in the beam heat deposition profile over the course of a shot. The shot shown in this plot was part of the testing dataset and was therefore never seen by the network during the training process. Figure 2b shows the NubeamNet prediction plotted against the actual NUBEAM output for every point in the testing dataset. On this plot, the line with a slope of 1 represents a perfect match between the NUBEAM and NubeamNet predictions. The coefficient of determination, or R^2 value of linear regression,[24] is calculated as 0.937 for the testing dataset. An in-depth discussion of these results can be found in Ref. [17]

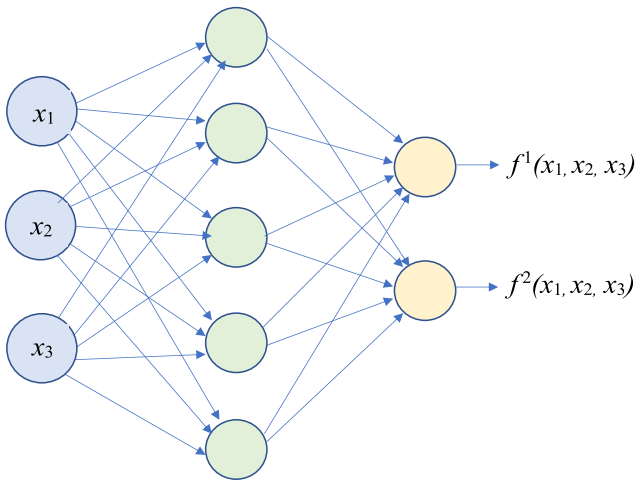
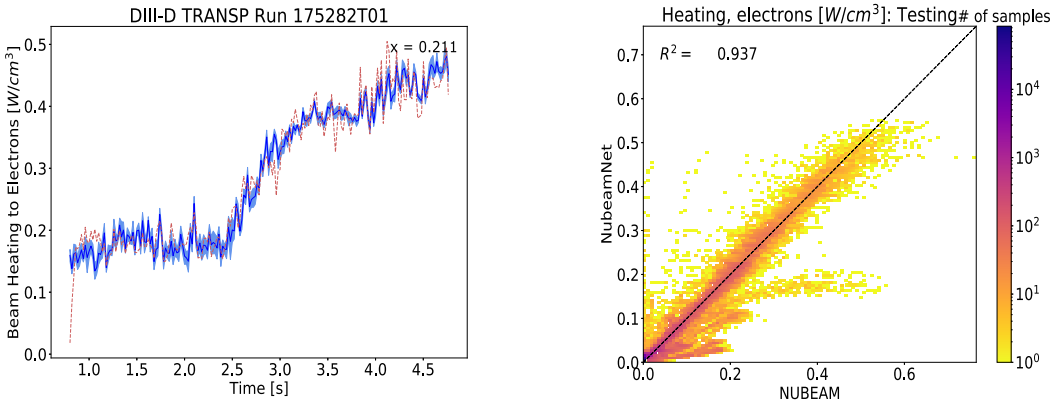


FIGURE 1 Structure of a multi-layer perceptron neural network[17]



(a) Heating to electrons as a function of time. The red line shows NUBEAM data, the dark blue line shows the average of the 5 NubeamNet predictions, and the light blue shaded area shows the standard deviation.

(b) Heating to electrons as calculated by NubeamNet plotted against heating to electrons as calculated by NUBEAM.

FIGURE 2 NubeamNet results compared to NUBEAM[17]

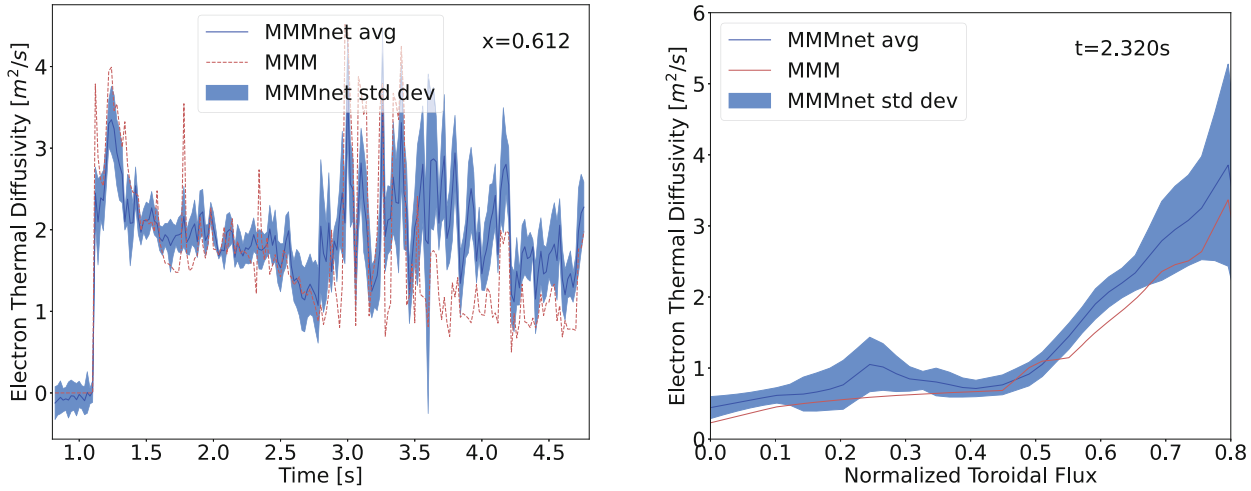
2.2.2 | MMMnet

The shape of the electron temperature profile strongly depends on the electron thermal diffusivity, which can be very difficult to calculate accurately. MMM[25] is a turbulent transport code that calculates a number of different thermal, particle, and momentum diffusivity coefficients. MMM runs significantly faster than NUBEAM, on the order of a couple of 100 ms, but this is still too slow for real-time control applications that run on the order of tens of milliseconds. Because of this, MMMnet[18] was developed. MMMnet is trained to predict three of the diffusivity coefficients produced by MMM: the electron thermal diffusivity (χ_e), ion thermal diffusivity (χ_i), and toroidal momentum diffusivity (χ_ϕ). In the future, it is planned to expand MMMnet to predict other outputs of MMM such as particle and poloidal momentum diffusivities. Like with NubeamNet, MMMnet uses the PCA technique to reduce the input and output profiles to a set of scalars. The PCA technique is applied to input profiles including electron and ion temperatures, electron, ion, and impurity densities, the safety factor, and ExB shear, among others. MMMnet is a multi-layer perceptron containing 3 hidden layers with 100 neurons in each layer. MMMnet also uses five parallel networks to help quantify the uncertainty introduced by using random initial weights.

In this work, MMMnet is used to calculate the electron thermal diffusivity coefficient (χ_e). Note that this diffusivity coefficient only represents transport driven by turbulence. Other factors such as neoclassical effects exist, but for electron temperature specifically, the vast majority of transport is in fact driven by turbulence, so those other factors are considered negligible. Figure 3 shows predictions by a slightly different version of MMMnet than the one presented in[18]; this version was updated to handle some difficulties the PCA had with certain normalized gradient inputs and to make the units of the inputs and outputs more consistent with the DIII-D Plasma Control System. Figure 3a shows the evolution of the MMM and MMMnet predictions of one point in the electron thermal diffusivity profile over the course of a shot from the testing dataset. Figure 3b shows the MMM and MMMnet predictions of the χ_e profile from $\widehat{\rho} = 0$ to $\widehat{\rho} = 0.8$ for a single time in the same shot. The version of MMM used to train this neural network is not valid in the pedestal region, so the training data predictions only extend to $\widehat{\rho} = 0.8$. This plot illustrates that, in the region where the predictions are valid, the PCA is able to reconstruct the shape of the diffusivity profile. However, Equation (1) requires values of χ_e in the region from $\widehat{\rho} = 0$ to $\widehat{\rho} = \widehat{\rho}_{bdry}$. In the case that $\widehat{\rho}_{bdry} > 0.8$, an extrapolation is made from the value of χ_e predicted by MMMnet at $\widehat{\rho} = 0.8$ out to $\widehat{\rho} = \widehat{\rho}_{bdry}$.

2.3 | Electron density and heating using analytical models

A simple model is used for the electron density, where the profile is assumed to have a constant shape defined by $n_e^{prof}(\widehat{\rho})$, and the magnitude is determined by the line-average electron density. This model is given by



(a) Electron thermal diffusivity as a function of time. The red line shows MMM data, the dark blue line shows the average of the 5 MMMnet predictions, and the light blue shaded area shows the standard deviation.

(b) Electron thermal diffusivity as a function of flux. The red line shows MMM data, the dark blue line shows the average of the 5 MMMnet predictions, and the light blue shaded area shows the standard deviation.

FIGURE 3 MMMnet results compared to MMM

$$n_e(\hat{\rho},t) = n_e^{prof}(\hat{\rho})\overline{n}_e(t). \tag{5}$$

Ohmic heating is calculated as

$$Q_{ohm}(\hat{\rho},t) = j_{tor}(\hat{\rho},t)^2\eta(\hat{\rho},t), \tag{6}$$

where the toroidal current density is defined as

$$j_{tor}(\hat{\rho},t) = -\frac{1}{\mu_0\rho_b^2 R_0\hat{H}}\frac{1}{\hat{\rho}}\frac{\partial}{\partial\hat{\rho}}\left(\hat{\rho}\hat{G}\hat{H}\frac{\partial\psi}{\partial\hat{\rho}}\right), \tag{7}$$

and the plasma resistivity is modelled as

$$\eta(\hat{\rho},t) = \frac{k_{sp}(\hat{\rho})Z_{eff}}{T_e(\hat{\rho},t)^{3/2}}. \tag{8}$$

The physical constant μ_0 is the permeability of free space, and R_0 is the major radius of the plasma. The poloidal stream function ψ is proportional to the poloidal magnetic flux of the plasma and is assumed to be known. The parameter $k_{sp}(\hat{\rho})$ is an empirically-derived normalizing factor, and the mean effective charge Z_{eff} is assumed to be constant. The value of Q_{nbi} is calculated by NubeamNet as described in Section 2.2.1. The Q_{ec} contribution is calculated as

$$Q_{ec}(\hat{\rho},t) = Q_{ec}^{ref}(\hat{\rho})P_{ec}(t), \tag{9}$$

where $Q_{ec}^{ref}(\hat{\rho})$ is a constant profile and $P_{ec}(t)$ is the power coming from the electron cyclotron heating system. The heat exchanged between electrons and ions are calculated as

$$Q_{ei}(\hat{\rho},t) = \nu_e(\hat{\rho},t)n_e(\hat{\rho},t)\left(T_e(\hat{\rho},t) - T_i(\hat{\rho},t)\right), \tag{10}$$

where the collisionality is defined as

$$\nu_e = \frac{0.041T_e(\hat{\rho},t)^{-3/2}\left(n_e(\hat{\rho},t) * 10^{-19}\right)}{A_{eff}}, \tag{11}$$

the ion temperature T_i is assumed to be a scalar multiple of T_e, and A_{eff} is the mean effective mass number of the ions. Although a number of different radiative effects exist, most of the heat lost by electrons through radiation comes from Bremsstrahlung radiation,[26] which is the only effect considered here (other radiation sources could be modelled if necessary or desirable). This is calculated as

$$Q_{rad}(\widehat{\rho},t) = k_{brem} Z_{eff} n_e(\widehat{\rho},t)^2 \sqrt{T_e(\widehat{\rho},t)}, \tag{12}$$

where k_{brem} is the Bremsstrahlung constant.

2.4 | Model discretization

2.4.1 | In space

The discrete spatial grid used in this work is defined by n number of points evenly spaced between $\widehat{\rho}=0$ and $\widehat{\rho}=\widehat{\rho}_{bdry}<1$. The spatial grid cuts off at $\widehat{\rho}=\widehat{\rho}_{bdry}$ because the model described in this section is not valid in the pedestal region of H-mode plasmas. The grid is then defined as

$$\Delta\widehat{\rho} = \frac{\widehat{\rho}_{bdry}}{n-1}, \quad \widehat{\rho}_i = (i-1)\Delta\widehat{\rho}, \quad i \in \{1, \dots, n\}. \tag{13}$$

The notation $T_{e,i}$ is used to represent $T_e\left(\widehat{\rho}_i, t\right)$. The form of the electron heat transport equation shown in Equation (3) is discretized in space using the second-order finite difference approximations

$$\left.\frac{\partial T_e}{\partial\widehat{\rho}}\right|_{\widehat{\rho}_i} = \frac{T_{e,i+1}-T_{e,i-1}}{2\Delta\widehat{\rho}}, \qquad \left.\frac{\partial^2 T_e}{\partial\widehat{\rho}^2}\right|_{\widehat{\rho}_i} = \frac{T_{e,i+1}+T_{e,i-1}-2T_{e,i}}{\Delta\widehat{\rho}^2}. \tag{14}$$

For $i \in [2, n-1]$, Equation (3) is reduced to a system of ODEs written as

$$\dot{T}_{e,i} = f_1\left(\widehat{\rho}_i,t\right)\left(\frac{T_{e,i+1}+T_{e,i-1}-2T_{e,i}}{\Delta\widehat{\rho}^2}\right) + f_2\left(\widehat{\rho}_i,t\right)\left(\frac{T_{e,i+1}-T_{e,i-1}}{2\Delta\widehat{\rho}}\right) + f_3\left(\widehat{\rho}_i,t\right)T_{e,i} + f_4\left(\widehat{\rho}_i,t\right). \tag{15}$$

The boundary condition at the edge of the profile is written as

$$\dot{T}_{e,n} = 0. \tag{16}$$

The boundary condition at the centre of the profile is discretized using the forward finite difference equation

$$\left.\frac{\partial T_e}{\partial\widehat{\rho}}\right|_{\widehat{\rho}_1} = \frac{4T_{e_2}-T_{e_3}-3T_{e_1}}{2\Delta\widehat{\rho}}. \tag{17}$$

Because this boundary condition equation is equal to zero, this can be simplified to

$$4T^j_{e,2} - T^j_{e,3} - 3T^j_{e,1} = 0. \tag{18}$$

2.4.2 | In time

The discrete temporal grid used in this work is characterized by the time step size Δt. The grid is then defined as

$$t^j = j\Delta t, \qquad j \in \{0, 1, \dots\}. \tag{19}$$

The notation $T^j_{e,i}$ is used to represent $T_e\left(\widehat{\rho}_i, t^j\right)$. The system of ODEs from Equation (15) is discretized in time using the first-order finite difference approximation

$$\dot{T}_{e,i}^{j+1} = \frac{T_{e,i}^{j+1} - T_{e,i}^{j}}{\Delta t}. \tag{20}$$

A partially implicit, partially explicit finite difference method is used where the T_e and spatial derivative of T_e terms are evaluated at t^{j+1}, and the coefficient functions f_1, f_2, f_3, and f_4 are evaluated at t^j. This hybrid approach retains the stability properties of a true implicit approximation while significantly simplifying the computation. Equation (15) is thus rewritten as

$$\frac{T_{e,i}^{j+1} - T_{e,i}^{j}}{\Delta t} = f_1\left(\widehat{\rho}_i, t^j\right)\left(\frac{T_{e,i+1}^{j+1} + T_{e,i-1}^{j+1} - 2T_{e,i}^{j+1}}{\Delta\widehat{\rho}^2}\right) + f_2\left(\widehat{\rho}_i, t^j\right)\left(\frac{T_{e,i+1}^{j+1} - T_{e,i-1}^{j+1}}{2\Delta\widehat{\rho}}\right) + f_3\left(\widehat{\rho}_i, t^j\right) T_{e,i}^{j+1} + f_4\left(\widehat{\rho}_i, t^j\right), \tag{21}$$

and Equation (16) is rewritten as

$$T_{e,n}^{j+1} = T_{e,n}^{j}. \tag{22}$$

The system state x is defined as

$$x \triangleq \begin{bmatrix} T_{e,1} & T_{e,2} & \dots & T_{e,n} \end{bmatrix}^T. \tag{23}$$

The state x_i^{j+1} is calculated using Equation (21) for $i \in [2, n-1]$ and Equation (22) for $i = n$. For $i = 1$, the value of x_1^{j+1} is computed implicitly from Equation (18). Manipulating Equation (21) to isolate the x^{j+1} terms and defining the input to the system u as

$$u \triangleq \begin{bmatrix} \overline{n}_e & P_{ec} & P_{nbi1} & \dots & P_{nbi8} & I_p \end{bmatrix}^T, \tag{24}$$

the full system can be written as

$$x^{j+1} = G\left(x^j, u^j, t^j\right). \tag{25}$$

3 | OBSERVER DESIGN

The extended Kalman filter[11] algorithm assumes that the nonlinear, noisy system takes the form

$$\begin{aligned} x^{j+1} &= G\left(x^j, u^j, t^j\right) + w^j, \\ y^j &= C\left(x^j, u^j, t^j\right) + v^j, \end{aligned} \tag{26}$$

where G denotes the noiseless nonlinear system, C denotes the measurable output, w represents internal noise, and v represents measurement noise. In this work, G is defined as the model presented in Section 2 (see Equation (25)). The output y is defined as a subset of length m of the state x, so the function $C(x^j, u^j, t^j)$ can be written as Mx where M is a matrix of ones and zeros. Some statistical properties of the noise are assumed to be known and will be defined in Section 4.2.

The extended Kalman filter algorithm is composed of two main steps: the prediction step and the correction step. The prediction step consists of the following two equations:

$$\begin{aligned} \widetilde{x}^j &= G\left(x^{j-1}, u^{j-1}, t^{j-1}\right), \\ \widetilde{y}^j &= C\left(x^j, u^j, t^j\right), \end{aligned} \tag{27}$$

where $\widetilde{x} \in \mathbb{R}^{n\times 1}$ is the predicted state and $\widetilde{y} \in \mathbb{R}^{m\times 1}$ is the predicted output. The connections between u, G, $\widetilde{x}$, C, and $\widetilde{y}$ are shown visually in Figure 4. The correction step consists of the following five equations:

$$
\begin{aligned}
e^j &= y^j - \tilde{y}^j, \\
\tilde{P}^j &= F^{j-1} P^{j-1} F^{j-1^T} + Q^{j-1}, \\
K^j &= \tilde{P}^j H^{j^T} \left(H^j \tilde{P}^j H^{j^T} + R^j \right)^{-1}, \\
\bar{x}^j &= \tilde{x}^j + K^j e^j, \\
P^j &= \left(I - K^j H^j \right) \tilde{P}^j,
\end{aligned} \tag{28}
$$

where $y \in \mathbb{R}^{m\times 1}$ is the noisy measured output, $e \in \mathbb{R}^{m\times 1}$ is the output error, $\tilde{P} \in \mathbb{R}^{n\times n}$ is the predicted covariance of the state, $F \in \mathbb{R}^{n\times n}$ is the Jacobian of G (see Section 3.1), $Q \in \mathbb{R}^{n\times n}$ is the covariance of the internal noise (see Section 4.2), $H \in \mathbb{R}^{m\times n}$ is the Jacobian of C (see Section 3.1), $R \in \mathbb{R}^{m\times m}$ is the covariance of the measurement noise (see Section 4.2), $K \in \mathbb{R}^{n\times m}$ is the observer gain, $\bar{x} \in \mathbb{R}^{n\times 1}$ is the corrected state, and $P \in \mathbb{R}^{n\times n}$ is the corrected covariance of the state. The goal of the observer algorithm is to generate an estimated state $\bar{x}$ in real time that is consistent with both the available measurements and the model in Equation (1). A visual representation of the observer algorithm is shown in the block diagram in Figure 4.

3.1 | Calculation of Jacobians

The Jacobian of the function G is defined as

$$
F = \begin{bmatrix} \frac{\partial G_1}{\partial x_1} & \cdots & \frac{\partial G_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial G_n}{\partial x_1} & \cdots & \frac{\partial G_n}{\partial x_n} \end{bmatrix}. \tag{29}
$$

Two methods of calculating this Jacobian have been tested in the work: a numerical method and an analytical method. For each of these methods, F is recalculated at each time step of the simulation to effectively linearize the model around the state at that time. The numerical method is implemented by solving the first line of Equation (27) $n+1$ times, once with the actual value of x^{j-1} and once using $\left[\hat{x}^{j-1}(i) = x_1^{j-1} \quad \dots \; x_i^{j-1} + \Delta x \quad \dots \; x_n^{j-1}\right]^T$ for $i \in [1, n]$ for a very small Δx. Each column of F is then calculated as

$$
F(i, :) = \frac{G\left(\hat{x}^{j-1}(i)\right) - G\left(x^{j-1}\right)}{\Delta x}. \tag{30}
$$

The analytical method uses analytical derivatives with respect to x of each component of the model, including the neural networks, to calculate F. Results of the observer algorithm using both approaches to calculating the Jacobian will be shown in Section 4.3.

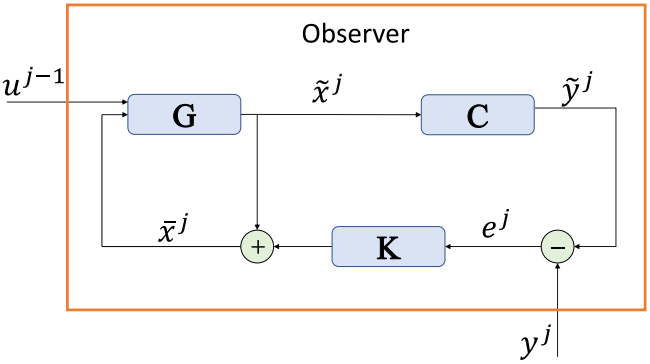


FIGURE 4 Block diagram showing the prediction ($\tilde{x}^j$) and correction ($\bar{x}^j$) steps of the observer with K denoting the gain calculated by the extended Kalman filter

Because the output y is defined as a subset of the state x, the output function C is a matrix M multiplied by the vector x. In this case, the Jacobian H is simply equal to M. Because M is a constant and is only a function of the definition of y (see Section 4.1), the Jacobian H does not need to be calculated at every time step of the simulation.

4 | RESULTS

4.1 | Definition of outputs

Observability is a property of the system that determines if the outputs contain enough information to allow for the estimation of the internal state. During the tests run in this work, observability was checked at every time point. In linear systems, observability can be determined by examining the rank of what is known as the observability matrix. In this case, the system is nonlinear, but it is effectively linearized around the state at each time step by the use of the Jacobians. Therefore, the observability matrix can be calculated as $\begin{bmatrix} H & HF & HF^2 & \dots & HF^{n-1} \end{bmatrix}^T$. If this matrix is full rank, the system is observable at that time point. If the observability matrix is less than full rank, the observer algorithm does not have enough information to work correctly and more outputs need to be added to the definition of a system.

The output vector y is defined as a subset of the state x at certain values of $\widehat{\rho}$. Initial tests were run using $\widehat{\rho}_{bdry} = 0.9$, $n = 10$, and $y = \begin{bmatrix} x|_{\widehat{\rho}=0} & x|_{\widehat{\rho}=0.5} & x|_{\widehat{\rho}=0.9} \end{bmatrix}^T$. It was determined that y must consist of values of the state in at least three different spatial locations in order for the system to be observable. When y only includes one or two spatial locations, the system is not observable at any time step. While only three states are necessary to meet the observability threshold, including additional states in y can still enhance the performance of the observer. The results were noticeably improved by adding a fourth state, but less difference was seen by adding a fifth state. Because of this, the results presented in Section 4.3 use a y vector that contains the values of the state at four locations.

4.2 | Characterization of noise

The extended Kalman filter algorithm assumes that both the internal and measurement noise follow Gaussian distributions with known covariance matrices Q and R, respectively. In order to use the observer algorithm effectively, these covariance matrices need to be determined, either by a statistical analysis of experimental data or using a heuristic approach. In this work, these covariance matrices were chosen to produce results that appeared physically reasonable while also visually matching the Thomson scattering data. It was assumed that each of these matrices will always have the same structure, but that the magnitude can change. This approach has the advantage of allowing the magnitude of the covariance matrices to be easily changed in real time during an experiment if machine conditions change.

The electron temperature in a tokamak will always be highest in the centre of the plasma and decrease toward the edge. This work assumes that the internal noise in the system is loosely proportional to the actual T_e profile. Because of this, the internal noise covariance matrix is defined as

$$Q = \begin{bmatrix} n & n-1 & n-2 & \dots & 1 \\ n-1 & n-1 & n-2 & \dots & 1 \\ n-2 & n-2 & n-2 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \times Q_{mag}, \tag{31}$$

where Q_{mag} is an adjustable parameter that defines the magnitude of the internal noise. Note that this matrix structure is not perfectly proportional to the T_e profile because the T_e profile is not linear; however, it does convey that the noise should be higher where the electron temperature is higher.

Like the internal noise covariance matrix, the measurement noise covariance matrix is assumed to be larger toward the centre of the profile than at the edge. For all the tests shown in this work, the Thomson scattering data were taken only from the core Thomson scattering system, which covers the range from $\widehat{\rho} \approx 0.2$ to $\widehat{\rho} \approx 0.9$. This data is then interpolated onto the observer spatial grid. The resulting Thomson data at $\widehat{\rho} = 0$ is less reliable than at other spatial locations, which is

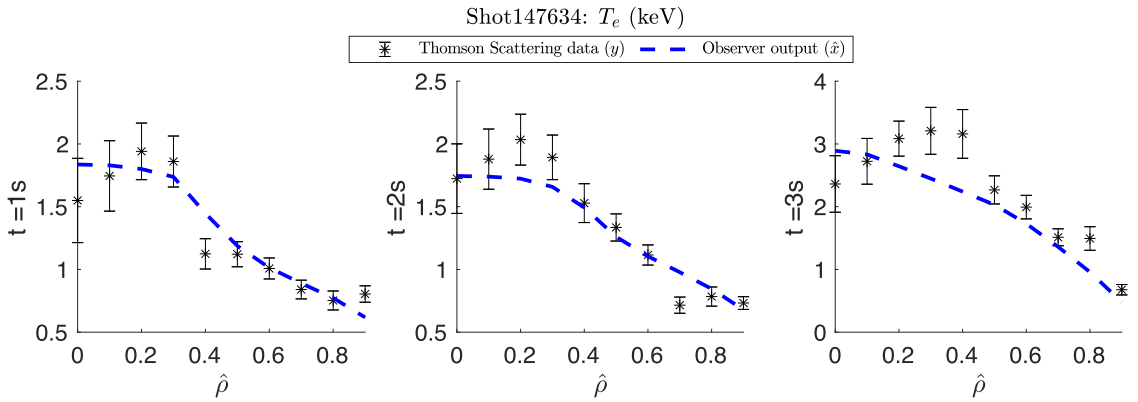
communicated to the observer by setting the noise level at the point higher. Unlike the internal noise covariance matrix, the measurement noise covariance matrix is chosen to be diagonal because individual channels of the Thomson scattering system are not highly dependent on other channels. The diagonal elements are defined as $n+1-i$ where i is the index that corresponds to each radial location of the state included in y, in decreasing order. So, if $n = 10$, $\hat{\rho}_{bdry} = 0.9$, and $y = \left[x\big|_{\hat{\rho}=0}\ x\big|_{\hat{\rho}=0.2}\ x\big|_{\hat{\rho}=0.6}\ x\big|_{\hat{\rho}=0.9} \right]^T$,

$$R = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times R_{mag}, \tag{32}$$

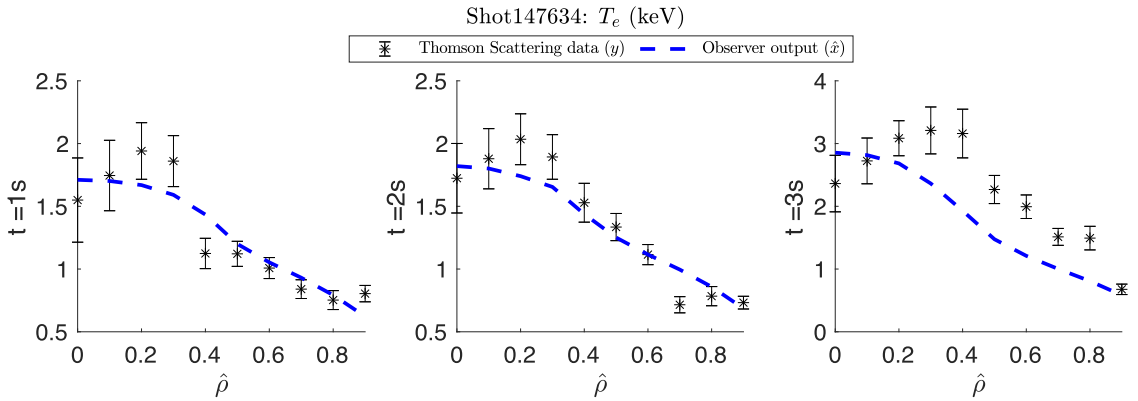
where R_{mag} is an adjustable parameter that defines the magnitude of the measurement noise.

4.3 | Analysis of results

Results shown in this section were obtained from running the observer offline using Thomson scattering data taken directly from DIII-D experiments as the measured output y. Tests were run for shot 147634, which is the shot that pieces of the model were based on (e.g., equilibrium values $\widehat{F}$, $\widehat{G}$, and $\widehat{H}$). For these tests, $n = 10$, $\hat{\rho}_{bdry} = 0.9$, $Q_{mag} = 0.3$, $R_{mag} = 2$, and y is defined as $\left[x\big|_{\hat{\rho}=0}\ x\big|_{\hat{\rho}=0.2}\ x\big|_{\hat{\rho}=0.6}\ x\big|_{\hat{\rho}=0.9}\right]^T$. Tests were performed using both the numerical and the analytical



(a) Using the numerical method to calculate the Jacobian.

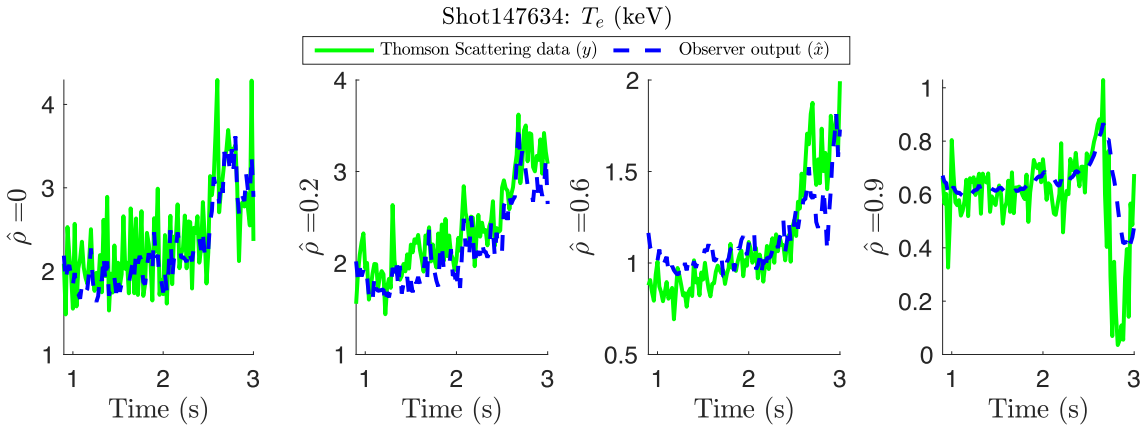


(b) Using the analytical method to calculate the Jacobian.

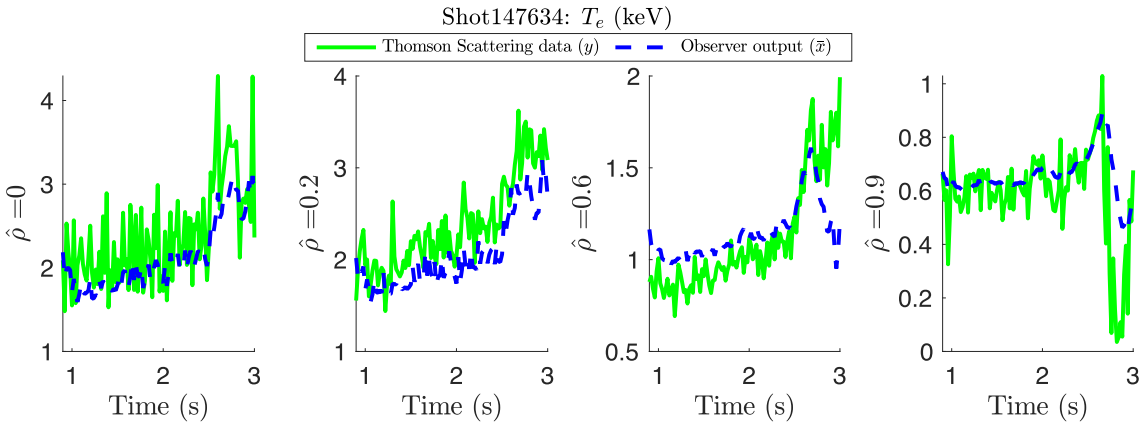
FIGURE 5 Observer-estimated state shown across the full profile three times during a test of shot 147634

approaches to calculating the Jacobian using all of the same parameters, and these results are compared in Figures 5 and 6. Figure 5 shows the results across the whole profile at three different times, and Figure 6 shows the results across time at four different spatial locations. Figures 5a and 6a show the results of using the numerical method of calculating the Jacobian F. These plots demonstrate that the observer is capable of producing a physically consistent T_e profile that is compatible with the Thomson scattering data while rejecting a significant amount of the noise. Figures 5b and 6b show results of using the analytical method of calculating the Jacobian F. These figures demonstrate that the analytical method produces comparable results to the numerical method.

Another test was run for shot 187076. This is a lower single null shot, as opposed to shot 147634 which is upper single null. Figure 7a shows the results of this test using the same parameters as the previous test at three different times. There are a significant number of Thomson scattering data points above the observer profiles, and very few Thomson scattering points below the profiles. At $t = 2\ s$ especially, the observer appears to be noticeably underestimating the Thomson scattering data. Because of the different equilibrium configurations, parts of the model are expected to be less accurate for this shot. These kinds of assumptions or simplifications in the model are treated by the observer as internal noise. To account for the different equilibrium, the internal noise magnitude is raised, telling the observer to trust the diagnostic data more. For another test, Q_{mag} is set to 0.7, and the results are shown in Figures 7b and 8. Figure 7b presents results of simulating shot 187076 across the whole profile at three times, and shows much less of an underprediction than is seen in Figure 7a. Figure 8 presents results across time at four spatial locations, and shows the value of T_e varying significantly over time due to large fluctuations in the neutral beam power; the observer is able to follow the fluctuations very closely. These plots demonstrate that the observer is capable of producing quality results even for shots that have very different equilibria from the one used to generate the model.

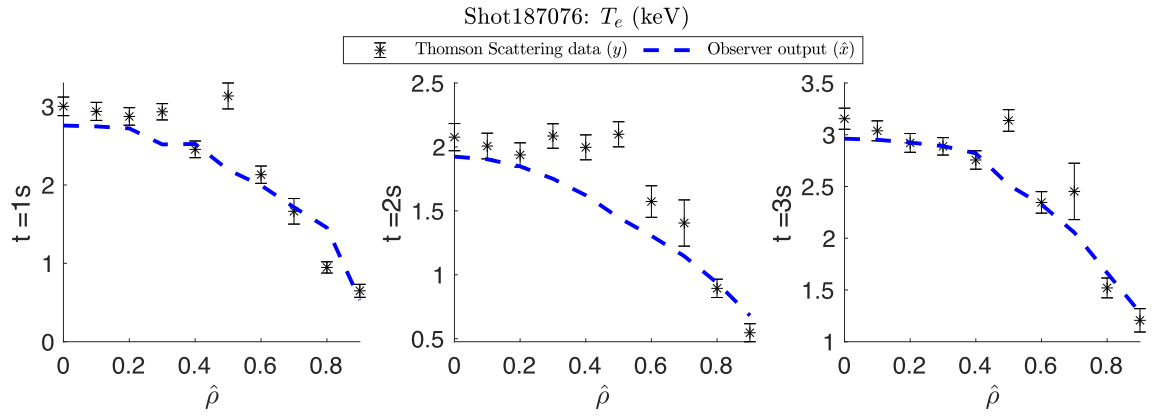


(a) Using the numerical method to calculate the Jacobian.

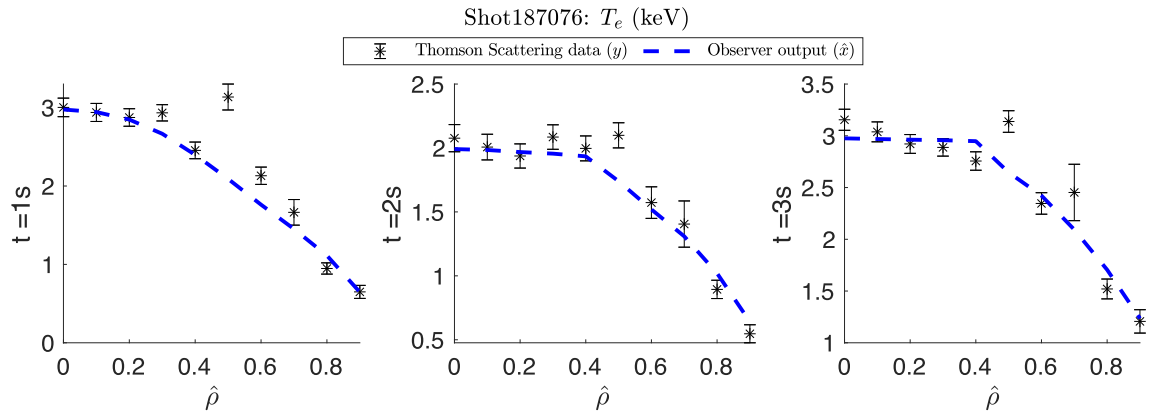


(b) Using the analytical method to calculate the Jacobian.

FIGURE 6 Observer-estimated state shown at the four spatial locations included in the observer output across time during a test of shot 147634



(a) $Q_{mag} = 0.3$



(b) $Q_{mag} = 0.7$

FIGURE 7 Observer-estimated state shown across the full profile three times during a test of shot 187076 using the numerical method to calculate the Jacobian

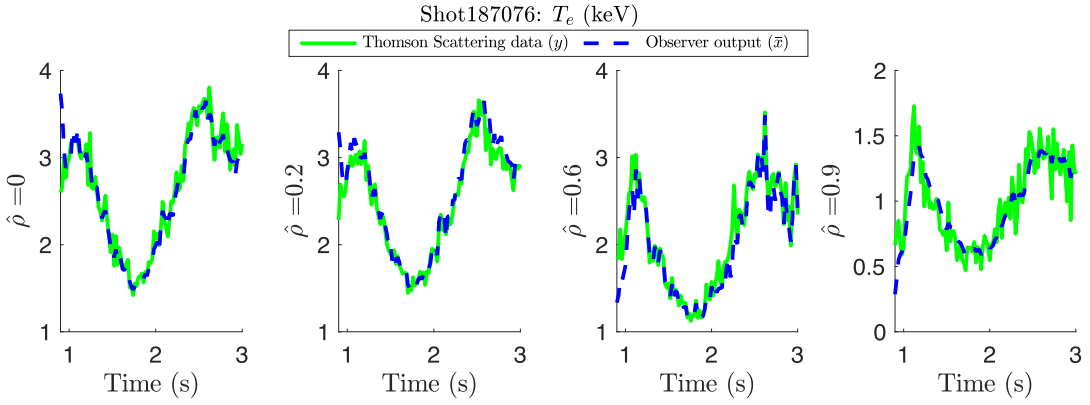


FIGURE 8 Observer-estimated state shown at the four spatial locations included in the observer output across time during a test of shot 187076 with Q_{mag} set to 0.7 and using the numerical method to calculate the Jacobian

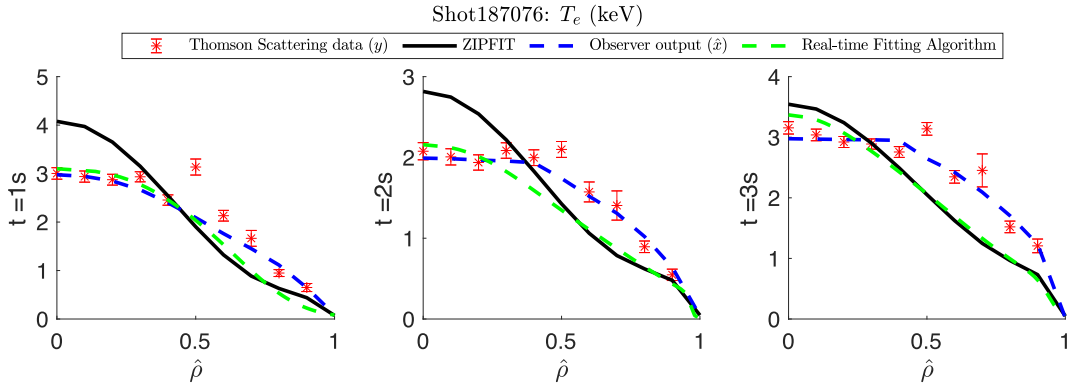


FIGURE 9 Observer-estimated state shown next to profile fits taken from the offline (ZIPFIT) algorithm and the real-time fitting algorithm

Figure 9 shows a comparison between the profile estimated by the observer algorithm and both offline and real-time profile fits for shot 187076. Note that, unlike the other plots shown, this one extends to the edge of the plasma. A linear interpolation is assumed between the final value predicted by the observer at $\hat{\rho} = 0.9$ and a value of 0 at $\hat{\rho} = 1$. The offline fit shown is taken from TRANSP[27] run in analysis mode, which actually takes profile fits as input. The profile fits used for this TRANSP run were originally generated by ZIPFIT,[28] a code that automatically generates profile fits in between each DIII-D shot. ZIPFIT by default tries to fit the diagnostic data to a hyperbolic tangent function but has other functions that it can try if the fit is not good enough. ZIPFIT does not run fast enough for real-time use. The real-time fit shown uses a hyperbolic tangent function and is available in real time. Both the real-time fitting algorithm and the offline ZIPFIT attempt to fit the entire spatial range of the profile, including the pedestal. When fitting to a set function, fitting the shape of the pedestal well can in some cases conflict with fitting the shape of the core well. The real-time fitting algorithm specifically prioritizes fitting the pedestal well. The observer algorithm is not attempting to predict the pedestal, and so does not have to make this kind of trade-off. In these plots, the observer calculations appear to match the shape of the Thomson scattering data in the core better than either of the other continuous profiles, which are both based on the same diagnostic data.

5 | CONCLUSIONS

A state observer has been developed to estimate the electron temperature profile in the DIII-D tokamak. The model used by the observer is based on the nonlinear transport equation describing the evolution of the electron temperature profile and uses neural network-based surrogate models to calculate the neutral beam heating and electron thermal diffusivity. The model is observable with as few as three measured outputs, which allows the algorithm to run with limited real-time processing of diagnostic data. The observer is based on the extended Kalman filter, a well-known nonlinear extension of the optimal Kalman filter algorithm. The observer has been shown to produce T_e profiles that are consistent with both the transport equation and the Thomson scattering data for the shot the model was based on. In addition, by adjusting the assumed magnitude of the internal noise, the observer was shown to produce similar quality results for a shot with a completely different equilibrium. It has also proven effective using either a numerical or analytical approach to calculating the Jacobian of the model. The observer appears to produce a better match to the shape of the core Thomson scattering data than other available profile-fitting approaches.

The observer is currently only valid in the core region of the plasma. In order to extend it into the pedestal region, a pedestal model could be added to the overall model. This would allow for a better prediction of the location and height of the pedestal and the estimation of the full profile. One candidate for this extension is the EPEDNN model,[13] a neural network version of the EPED pedestal model. In addition, the results shown here use noise covariance matrices chosen using a heuristic approach to produce outputs that look physically reasonable and visually match the diagnostic data. A more rigorous statistical approach to determining the covariance matrices of the noise could be implemented in the future.

This observer is in the process of being integrated into the real-time Plasma Control System (PCS) for DIII-D. The PCS implementation will take data from both the core and tangential Thomson scattering systems to better cover the whole spatial range. The observer is planned to be available for general use for the 2023 experimental campaign and is expected to add a valuable tool to the arsenal of real-time data available for feedback controllers to use.

ACKNOWLEDGEMENTS

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences, using the DIII-D National Fusion Facility, a DOE Office of Science user facility, under Awards DE-SC0010661, DE-FC02-04ER54698, and by the National Science Foundation Graduate Research Fellowship Program under Grant No. 1842163. S. Morosohk (morosohk@lehigh.edu) and E. Schuster are with the Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA 18015, USA. This report was prepared as an account of work sponsored by an agency of the US Government. Neither the US Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favouring by the US Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the US Government or any agency thereof.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] K.R. Middaugh, B.D. Bray, C.L. Hsieh, B.B. McHarg, and B.G. Penaflor. DIII-D Thomson scattering diagnostic data acquisition, processing, and analysis software. In 1999 IEEE Conference on Real-Time Computer Applications in Nuclear Particle and Plasma Physics. 11th IEEE NPSS Real Time Conference. Conference Record (Cat. No.99EX295), 342–345, **1999**.
[2] N. C. Logan, B. A. Grierson, S. R. Haskey, S. P. Smith, O. Meneghini, D. Eldon, *Fus. Sci. Technol.* **2018**, *74*, 125.
[3] Z. A. Xing, D. Eldon, A. O. Nelson, M. A. Roelofs, W. J. Eggert, O. Izacard, A. S. Glasser, N. C. Logan, O. Meneghini, S. P. Smith, R. Nazikian, E. Koleman, *Fus. Eng. Des.* **2021**, *163*, 112163.
[4] J. Leddy, S. Madireddy, E. Howell, S. Kruger, *Plasma Phys. Controll. Fus.* **2022**, *64*(10), 104005.
[5] D. G. Luenberger, *IEEE Transact. Milit. Electr.* **1964**, *8*(2), 74.
[6] T. C. Blanken, F. Felici, C. J. Rapson, M. R. de Baar, W. P. M. H. Heemels, *Fus. Eng. Des.* **2018**, *126*, 87.
[7] T. O. S. J. Bosman, M. van Berkel, M. R. de Baar, *J. Phys. Commun.* **2021**, *5*(11), 115015.
[8] B. Vincent, N. Hudon, L. Lefévre, D. Dochain, *IFAC-Pap. Online* **2016**, *49*(24), 93 2th IFAC Workshop on Thermodynamic Foundations for a Mathematical Systems Theory TFMST 2016.
[9] H. Wang, J. E. Barton, and E. Schuster, Poloidal flux profile reconstruction from pointwise measurements via extended Kalman filtering in the DIII-D tokamak. In 2015 IEEE Conference on Control Applications (CCA), 1309–1314, **2015**.
[10] F. Felici, J. Citrin, A. A. Teplukhina, J. Redondo, C. Bourdelle, F. Imbeaux, O. Sauter, JET Contributors, *Nucl. Fus.* **2018**, *58*, 96006.
[11] B. D. O. Anderson, J. B. Moore, Optimal Filtering, Prentice-Hall, Englewood Cliffs, NJ **1979**.
[12] M. D. Boyer, S. Kaye, K. Erickson, *Nucl. Fus.* **2019**, *59*, 56008.
[13] O. Meneghini, S. P. Smith, P. B. Snyder, G. M. Staebler, J. Candy, E. Belli, L. Lao, M. Kostuk, T. Luce, T. Luda, J. M. Park, F. Poli, *Nucl. Fus.* **2017**, *57*, 86034.
[14] J. Citrin, S. Breton, F. Felici, F. Imbeaux, T. Aniel, J. F. Artaud, B. Baiocchi, C. Bourdelle, Y. Camenen, J. Garcia, *Nucl. Fus.* **2015**, *55*, 92001.
[15] A. Lakhal, A. Tlili, N. Braiek, et al., *Int. J. Contr. Automat.* **2010**, *3*, 4.
[16] T. Fretheim, R. Shouresh, S. Vincent, D. Torgerson, and J. Work. A general approach to non-linear output observer design using neural network models. In Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334), *2*, 924–928, **2000**.
[17] S. M. Morosohk, M. D. Boyer, E. Schuster, *Fus. Eng. Des.* **2021**, *163*, 112125.
[18] S. M. Morosohk, A. Pajares, T. Rafiq, E. Schuster, *Nucl. Fus.* **2021**, *61*, 106040.
[19] V. Basiuk, J. F. Artaud, F. Imbeaux, et al., *Nucl. Fus.* **2003**, *43*(9), 822.
[20] Y. Ou, T. C. Luce, E. Schuster, J. R. Ferron, M. L. Walker, C. Xu, D. A. Humphreys, *Fus. Eng. Des.* **2007**, *82*, 1153.
[21] A. Pankin, D. McCune, R. Andre, G. Bateman, A. Kritz, *Comput. Phys. Commun.* **2004**, *159*, 157.
[22] H. Hotelling, *J. Educ. Psychol.* **1933**, *24*, 417.
[23] M. A. Nielsen, Neural Networks and Deep Learning, San Francisco, Determination Press, **2015**.

[24] S. A. Glantz, B. K. Slinker, T. B. Neilands, Primer of Applied Regression and Analysis of Variance, 3 e, McGraw-Hill Education, New York, NY **2017**.
[25] T. Rafiq, A. H. Kritz, J. Weiland, A. Y. Pankin, L. Luo, *Phys. Plasm.* **2013**, *20*, 32506.
[26] E. Haug, W. Nakel, The Elementary Process of Bremsstrahlung, Vol. *73*, World Scientific, River Edge, NJ **2004**.
[27] Joshua Breslau, Marina Gorelenkova, Francesca Poli, Jai Sachdev, and Xingqiu Yuan, TRANSP. **2018**.
[28] Q. Peng, R. J. Groebner, L. L. Lao, J. Schachter, D. P. Schissel, M. R. Wade, *Fus. Eng. Des.* **2002**, *60*, 319.

How to cite this article: S. Morosohk, E. Schuster, *Contributions to Plasma Physics* **2023**, e202200153. https://doi.org/10.1002/ctpp.202200153