공학박사 학위논문

Tokamak operation trajectory design with deep reinforcement learning in KSTAR

강화학습을 이용한
KSTAR 토카막 운전 궤적 설계

2022 년 8 월

서울대학교 대학원

에너지시스템공학부 원자핵공학 전공

서 재 민

Tokamak operation trajectory design with deep reinforcement learning in KSTAR

지도 교수 나 용 수

이 논문을 공학박사 학위논문으로 제출함
2022 년 8 월

서울대학교 대학원
에너지시스템공학부 원자핵공학 전공
서 재 민

서재민의 공학박사 학위논문을 인준함
2022 년 8 월

위 원 장 함 택 수 (인)

부위원장 나 용 수 (인)

위 원 황 용 석 (인)

위 원 김 성 우 (인)

위 원 이 정 원 (인)

Abstract

In order to conduct a sophisticated physics experiment in a tokamak, it is necessary to achieve and sustain a specific target plasma state first. Especially, the commercial fusion reactor requires controlling plasmas within a stable parametric range and maintaining a favorable plasma state for high fusion power generation. Conventionally, we had to conduct numerous simulations with various tokamak operating conditions and experiment with trials and errors for achieving a target plasma state. This takes lots of labor and time and requires the same level of trial and error for different targets each time.

This thesis addresses the development of a reinforcement learning (RL)-based algorithm that designs the tokamak operation trajectory to achieve a given target plasma state. This algorithm replaces the conventional manual tasks of numerous simulative experiments and provides a probable tokamak operation condition faster and more efficiently. First, the tokamak simulator, corresponding to the training environment of the RL agent that designs the operation trajectory, was developed. An LSTM-based neural network was trained that sequentially predicts the plasma state over time by learning the patterns of the KSTAR experimental data. Various numerical techniques were applied to prevent overfitting and error accumulation during the training process. The trained model showed reasonable prediction accuracy for various operation scenarios in KSTAR, and reliability analyses verified that the model was not significantly overfitted. Furthermore, based on the trained model, we developed a graphical user interface (GUI) to enable virtual tokamak experiments through real-time interaction. By adjusting the tokamak operation parameters on the GUI, the user can visually check the plasma evolution in real time, which can be useful not only for physics research but also for expert education.

Second, an artificial agent was trained using a reinforcement learning technique, that adjusts the operation parameters to achieve a target plasma state in the developed simulator. This agent can design a plausible tokamak operation trajectory to achieve a given target after training. First, the agent was trained to determine the plasma current, the plasma shape, and the heating power to achieve the target β_N. We conducted a KSTAR experiment with the operation trajectory designed by the trained agent, and it was verified that the target performance was achieved within the tolerance range. In particular, it was observed that the confinement enhancement factor was improved by adjusting the plasma shape to achieve high performance under limited heating conditions. Moreover, in order to achieve a more specific plasma state, another RL agent was trained to achieve multiple targets of β_p, q_{95}, and l_i simultaneously. The KSTAR experiment with the RL operation design showed that multiple plasma parameters were successfully controlled to the target values.

The RL-based algorithm addressed in this thesis can provide clues for the research of advanced operation scenarios and can be applied to achieve initial plasma states in experiments that require sophisticated physical conditions. By applying this algorithm to real-time feedback control in the future, it will become a basis for developing a self-operating fusion reactor that can be autonomously controlled to achieve high power generation.

Keyword: Tokamak, Plasma, KSTAR, Machine Learning, Reinforcement Learning, Tokamak Simulation, Plasma Control

Student Number: 2017-29826

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Nuclear fusion is one of the next-generation energy sources to fulfill carbon neutrality, which will be accomplished by cutting-edge science, engineering, and technology. Fusion does not emit greenhouse gases such as carbon dioxide and does not generate high-level radioactive waste during electric power generation. Fusion fuels are widely available from seawater and are nearly inexhaustible compared to other fossil energy sources. The fusion reaction of Deuterium (D) and Tritium (T) produces Helium (He) and the energy based on the principle of mass-energy equivalence. To commercialize the fusion reactor, we need to sustain the fusion reaction by confining the high-temperature hydrogenic plasma stably and stationaliry.

A tokamak, the most promising concept for the first nuclear

fusion reactor, was introduced by Andrei Sakharov and Igor Tamm in the 1950s. Its basic components include the toroidal field (TF) coils, the poloidal field (PF) coils, and the central solenoid, as shown in Figure 1.1. The tokamak confines the plasmas with the helical magnetic field produced by the TF coils and the plasma current. The plasma current can be driven by several ways; the central solenoid, the self-generating mechanisms, and the external sources such as neutral beam injection (NBI) and electron cyclotron (EC) resonance. The NBI and EC can also heat up the plasmas to reach the fusion ignition condition. The largest tokamak in the world, International Thermonuclear Experimental Reactor (ITER), is currently under construction in France, collaborating with China, the European Union, India, Japan, the Republic of Korea, Russia, and the United States [1–3]. Various experiments are being conducted in present tokamak devices such as KSTAR [4] to establish the scientific and engineering basis for the successful operation of ITER.

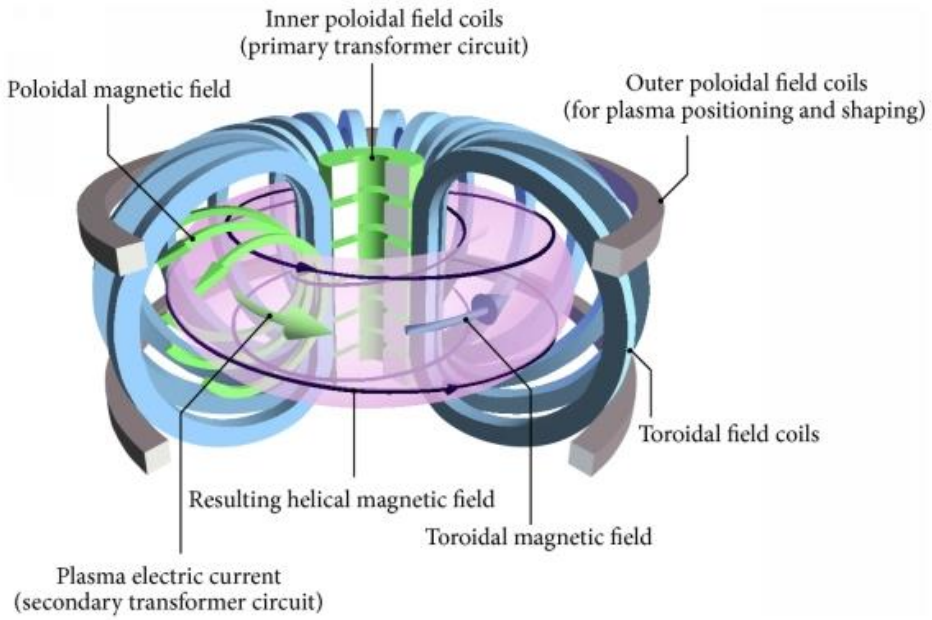


Figure 1.1 Schematic view of a tokamak device and magnetic field structure (This figure has been re-used from an open access article [5] which permits unrestricted use with citation.)

1.1. Advanced operation scenario in tokamak

Since a high-confinement plasma operation mode (H-mode) was discovered by diverted plasma configuration and NBI heating in the 1980s [6], fusion researchers have explored various tokamak operation modes to achieve a more advanced-confinement, stable, or steady-state (fully non-inductive) condition. To define and interpret those advanced operation scenarios in various tokamaks, it is required to properly quantify the plasma states, such as the global plasma performance or the magnetic structure. The 0D dimensionless parameters can be good indicators to compare the plasma states in

different machines with each other regardless of different configurations and magnetic field strength. For this reason, advanced operation scenarios are often characterized by those 0D dimensionless parameters such as the normalized (β_N) and poloidal (β_p) beta, the confinement enhancement factor (H_{89} and H_{98}), the safety factor (q_0 and q_{95}), and the internal inductance (l_i). For example, the high-β_p scenario [7, 8] and the high-l_i scenario [9] are coined with these dimensionless parameters, and the hybrid scenario is defined with $\beta_N \gtrsim 2.4$, $H_{89} \gtrsim 2$, and $q_{95} < 6.5$ in KSTAR and ITER [10, 11].

1.2. Machine learning in fusion research

1.2.1. Precedent research

As machine learning (ML) research fields were actively developed and popularized in the 2010s, various ML techniques have also been applied for various purposes in fusion research.

Plasma disruption has been considered one of the most critical issues for commercializing fusion reactors since it halts fusion power production and significantly damages the plasma-facing components. J. Kates-Harbeck et al. [12] presented a method based on deep learning for forecasting disruptive instabilities in tokamaks. By using recurrent and convolutional neural networks trained on past experimental data, the plasma disruption events could be alarmed in advance in two different machines. These initial results could

illustrate the potential for deep learning to accelerate progress in fusion science and the understanding and prediction of complex physical systems. The techniques for predicting plasma disruption using deep learning are being developed continuously [13].

The predictive simulation for modeling the whole fusion device requires self-consistently coupled physics models, including plasma heating, turbulent transport, and edge pedestal stability. However, physically accurate calculation of each element with theory-based models, such as QuaLiKiz [14], TGLF [15], and EPED1 [16], take minutes to hours for the calculation of a single time-slice target. In order to enable a faster simulation, J. Citrin et al. [17] and O. Meneghini et al. [18] proposed the deep neural network (DNN)-based approach to perform nonlinear multivariate regression of theory-based models. The DNN-based approach provides the reliable prediction of physical quantities with a computational speedup of several orders of magnitudes.

Edge localized mode (ELM) is one of the typical characteristics observed in H-mode plasmas and can induce significant damage to the plasma-facing components. It is required to detect the H-mode transition and the ELM occurrence for mitigating and suppressing the ELM crash during the discharge. G. Shin et al. [19, 20] applied the support vector machine and the recurrent neural network to detect and classify the plasma operation regimes, including the L-, H-mode, and ELMy phases.

1.2.2. AI operation trajectory design and control

To implement the advanced operation scenario in a tokamak, we have to control the plasma state and maintain it within the desired window. Active control of plasma parameters has been widely attempted with various techniques, such as the proportional–integral–differential (PID) feedback algorithm [21, 22] and model predictive control (MPC) [23]. Recently, a machine learning algorithm using deep reinforcement learning (RL) has been applied for real–time magnetic coil control to form various types of plasma configuration, which requires sophisticated decision–making in multi–dimensional engineering space based on the knowledge of magneto–hydrodynamics (MHD) [24]. Deep RL is a machine learning technique that empirically learns the optimal decision–making at given states by receiving a high reward for a good action [25]. The RL agent uses the decision–making network trained beforehand and only requires a single feedforward pass to make an optimal decision during the online environment, making the control much faster than MPC. In addition, the RL method can pursue a more future target achievement, whereas the PID controller determines the action based on the previous history of the errors without expectation of future behaviors of the system.

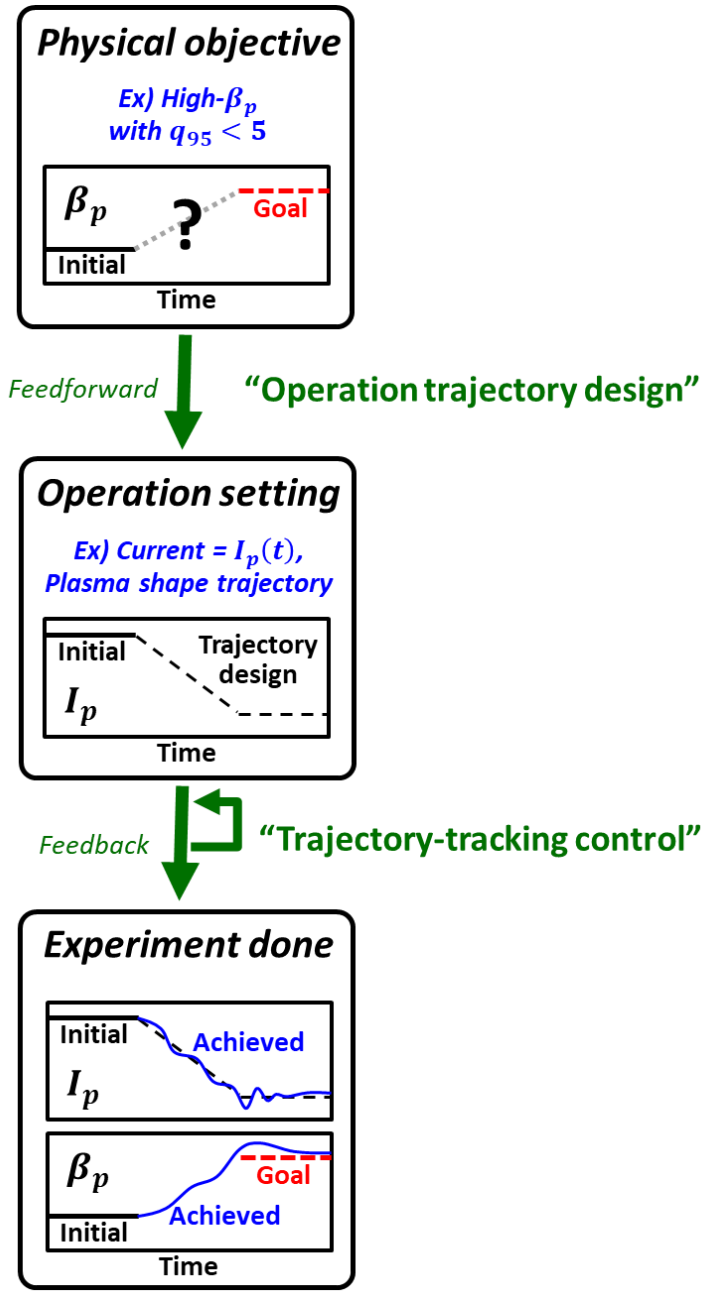


Figure 1.2 Typical tokamak experiment process composed of two stages; the operation trajectory design and the trajectory-tracking control.

Figure 1.2 shows the typical tokamak experiment process for developing advanced operation scenarios, which consists of two stages. First, we need to design the operation trajectories of the engineering parameters, such as the plasma current, the heating power, and the plasma boundary shape, to achieve the physical target of interest. This is a feedforward process, which is determined by researchers after numerous trials and errors in simulations and experiments. Second, in order to implement the tokamak experiment according to the designed operation trajectories, we need feedback controllers that adjust the magnetic coil system and the heating actuators to follow the given operation trajectories.

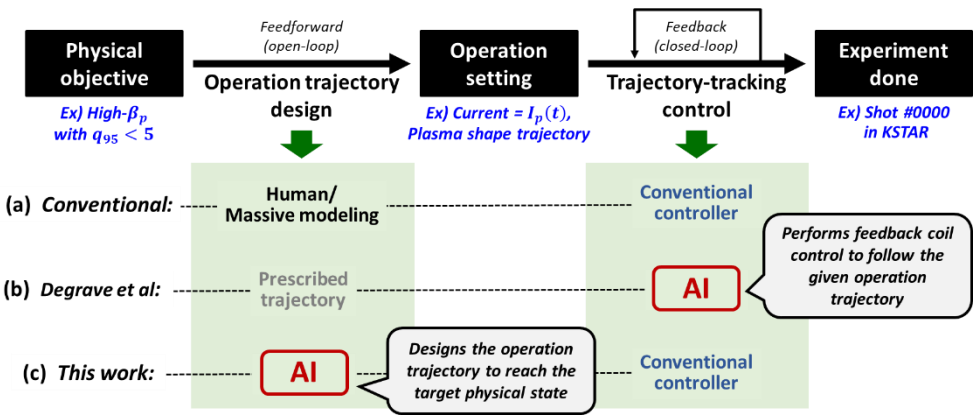


Figure 1.3 The methods for the two stages used in conventional ways (a), in the related work [24] (b), and in this work (c).

Conventionally (shown in Figure 1.3 (a)), the trajectories of the tokamak operation parameters have been designed by humans within the feasible engineering range. It requires trials and errors or massive simulations to find an appropriate trajectory to achieve each

physical target we desire. Then, the experiment is conducted with a trajectory-tracking algorithm, such as the PID feedback control, to follow the pre-programmed operation trajectory. The RL model developed in [24] replaces the latter part, the trajectory-tracking controller, which controls the coil current to follow the prescribed operation trajectories (Figure 1.3 (b)). However, designing those operation trajectories is another problem. In this work, we suggest an AI algorithm designing the tokamak operation trajectory, which replaces the human tasks of trials and errors or massive simulations (Figure 1.3 (c)).

1.2.3. Simulation of tokamak plasmas

Numerical simulation methods for tokamak plasmas have been widely developed for different purposes and different spatiotemporal scales of the analysis.

Table 1.1 Comparison of different modeling methods for tokamak plasmas

	Gyro-kinetics	Fluid model	Integrated modeling
Physical fidelity	Highest	High	Intermediate
Spatiotemporal scale	Drift wave scale	≤Minor rad. / Alfvén time	Minor rad. / Pulse duration
Computation speed	Slow	Intermediate	Fast

Gyro-kinetic formalism uses gyro-averaged physical quantities for modeling the kinetic distribution of charged particles. It describes the first-principle plasma dynamics and can deal with micro-scale to meso-scale plasma turbulence and its interaction with other particles. However, it calculates multiple species' evolution in at least five dimensions and requires huge computational costs and time. In order to reduce this computational cost, fluid models such as single/two-fluid MHD and Landau fluid model have been introduced.

They use fluid quantities expressed with moments of the velocity-space of the plasma particles, such as the density, the mean flow velocity, and the temperature. A well-established MHD model can describe small- to large-scale instabilities in a tokamak [26] or provide useful information by reconstructing the plasma equilibrium structure from kinetic diagnostics [27, 28]. However, the MHD models are still not fast enough for the entire tokamak simulation for the pulse duration.

The integrated modeling, a coupled system of multiple models describing different physics, can simulate the entire tokamak plasma combined with various phenomena such as plasma equilibrium, plasma transport, and heating and current drive [29]. It provides a self-consistent simulation considering various engineering and physical aspects during the tokamak operation and is faster than full gyro-kinetic or MHD simulations. Thus, the integrated modeling method has been widely used for predictive simulation for future tokamaks like ITER or virtual experiments of advanced operation scenarios.

Although the simulation with integrated modeling is faster than other numerical simulations shown in Table 1.1, there still have been difficulties with its use of real-time-level prediction. A single step of the simulation with reliable physics modules such as TGLF [30], NUBEAM [31], and EPED1 [16] requires tens to hundreds of seconds per CPU. The entire simulation for a tokamak discharge takes at least hours, depending on how frequently we call each module. Recently, to speed up the integrated simulation further, neural-network (NN)-based accelerated models have been introduced [18, 32].

A deep neural network (DNN), or deep learning, is a computing system inspired by biological neural networks, which consists of the multiple layers of functional neurons and their connections.

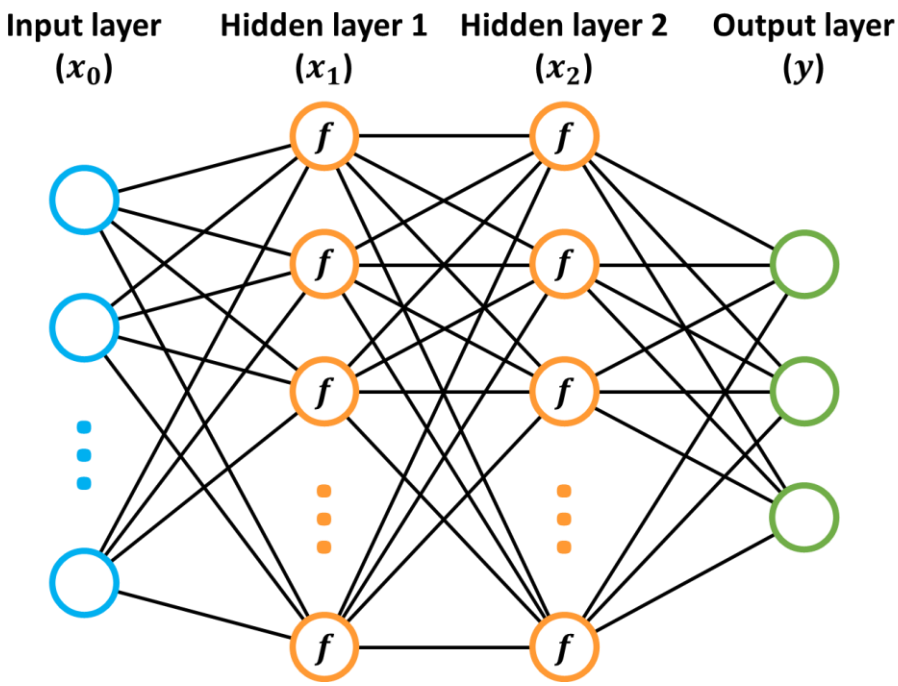


Figure 1.4 A basic, fully connected (dense) structure of a deep neural network

$$y=f(wx+b) \tag{2.1}$$

Each neuron is a function of the vector (x) from the previous layer that outputs the next vector (y), as expressed in Equation (2.1). Here, w is the weight matrix and b is the bias matrix. In order to add nonlinearity to this operation, f is defined with a nonlinear function

such as a hyperbolic tangent or sigmoid function. From the input vector (light blue circles) fed in, the DNN calculates the output vector (green circles) through more than one hidden layer (orange circles), as shown in Figure 1.4. The weight and bias parameters of the DNN are optimized with the gradient-descent method in the direction of best describing the given dataset. After the optimization of the internal parameters with the dataset, so-called training, the network can be used for a function that describes the response of the target system.

With the explosive development of deep learning technology in the 2010s, predictive analysis using DNN has been introduced in fusion research as well. An LSTM [33]-based DNN could predict disruptive instabilities of the tokamak plasmas 30 milliseconds before each event [12] and also predict the tearing mode onset [34]. In addition, DNN has been used for accelerating the computationally heavy modules to allow the integrated modeling to be capable of real-time simulation. A predictive modeling workflow has been established with NN-accelerated modules, TGLF-NN and EPED1-NN [18], and real-time-capable first-principle simulation has been demonstrated [32].

Most of the predictive modeling tools using DNN for fusion plasmas are based on the NN acceleration of several high-fidelity physics models. Those modeling systems still require solving partial differential equations (PDEs) in multi-dimensions, which takes non-negligible time for each step. The plasma equilibrium reconstruction and heating modules are not quite suitable for NN-acceleration, so there is a limit to improve the computational speed further. Thus, in this work, we introduce an end-to-end DNN simulator that does not

require any physical governing equations as a medium. This simulator will be trained with experimental data of a real tokamak. After training, it will predict the plasma behaviors that are most probable based on the dataset with a much faster speed than conventional integrated modeling.

Table 1.2 Comparison of physics-based and data-driven simulation

	Physics-based simulation	Data-driven simulation
Basis	Physics model	Experimental data
Advantage	Physically interpretable	Experimentally relevant, Fast
Disadvantage	Limited by the model's validity, Computationally costly	Hard to interpret (black-box)

Table 1.2 shows the comparison of the physics-based and the data-driven simulations. Since physics-based modeling uses a physical governing equation or formulas, we can interpret the process of deriving the output results. However, the prediction results and costs can vary depending on which models we use, and the prediction accuracy is limited by the validity of the used models. There has been reported that the physics-based simulation sometimes does not explain the experiments [10]. On the other hand,

the data-driven simulation, based on the DNN model trained with experimental data, provides a more experimentally relevant prediction. Since the prediction with DNN only requires arithmetic operations without solving any PDEs, the computational speed can be faster than real-time, which is suitable for RL training. The drawback of the data-driven simulation is that it is difficult to verify whether the derived outputs are physically reliable.

1.3. Objective and outline of this dissertation

In developing the advanced operation scenarios of a tokamak, it is often necessary to design an operation trajectory to reach a specific plasma state. In this work, an RL-based design algorithm that provides a possible operation trajectory for the desired plasma state is developed. Training the artificial agent with the deep RL technique requires thousands to millions of trials and errors to learn the action policy in a given environment. Real tokamak experiments for such learning are too costly, and the device would get substantial damage by the controller's disruptive actions during the exploration. Therefore, we first need a tokamak simulator for the training environment, which is fast enough and experimentally relevant. In this work, we first develop a data-driven tokamak simulator and then train the RL agent in the developed simulator.

Chapter 2 describes the data-driven simulator developed with experimental data in KSTAR as a virtual tokamak environment for RL training. The plasma boundary modeling and the predictive simulation of the internal plasma state are developed with deep neural networks.

Various numerical schemes to solve the obstacles in the modeling are also introduced.

Chapter 3 deals with the RL-based algorithm that designs the tokamak operation trajectories for target plasma states. The RL agent learns the best action policy for the tokamak control in the simulation environment. The numerical settings for RL training and the training process are described. The results of several KSTAR experiments with the RL-designed operation trajectories are discussed.

Finally, Chapter 4 summarizes the contents and suggests possible future works.

Chapter 2

Data-driven tokamak simulator

In order for the RL agent to learn a good enough action policy, it requires thousands to millions of experiences consisting of exploration and fine-tuning, depending on the problem. In various engineering fields to which the RL is applied, the training environment is constructed with a numerical simulation system rather than real-world experiences. The training in the simulation environment prevents possible real-world accidents or device damage that may occur during the agent's exploration. In this chapter, we develop a simulation system for predicting the behavior of the tokamak plasmas, which can be used as a training environment for RL in the next chapter.

2.1. Predictive modeling with DNN

The superconducting tokamak in Korea, KSTAR, has been operating for more than ten years since its first plasma discharge in 2008. All the experimental data during the operation have been managed with a tree-based distributed data acquisition system, MDSplus [35]. By using the numerous engineering and physical data stored in the MDSplus server and the methodology of inferencing the target output signals by learning the data patterns, such as deep learning and decision tree model, we can construct a data-driven model that predicts the target output variables of interest. The data-driven model can calculate the outputs much faster than the first-principle theory-based model and provides a more experiment-relevant prediction, including the phenomena that are hard to be explained with existing theories. In this work, the DNN structure has been used to learn the patterns from the continuous time-series experimental data, and then it infers the plasma responses from the actuators' control inputs.

2.1.1. Construction and training of the LSTM-based model

Predicting the internal plasma state is solving a time-dependent physics problem. Physical phenomena of the plasma are determined by multiple governing equations, such as MHD equation, energy transport equation, and magnetic flux diffusion equation. These governing equations have different characteristic time-scale

according to the geometric scale and the field strength. For example, in KSTAR, the MHD instabilities grow in $< O(10^{-3})$ s, the transient energy stored by external heating is confined in $O(10^{-1})$ s, and the perturbation of the edge electric field is penetrated into the core region in $O(10^{0})$ s. Due to these time-dependent properties of the plasmas, even under the same external control, they respond differently according to their previous and current state. This property causes the bifurcation or the hysteresis phenomena of the fusion plasmas. In order to predict these plasma responses properly, the model should be capable of capturing time-dependent information by reflecting the previous history of the plasma state. In this work, the long short-term memory (LSTM) [33]-based neural network has been adopted to reflect the previous history of the plasma state and the external control.

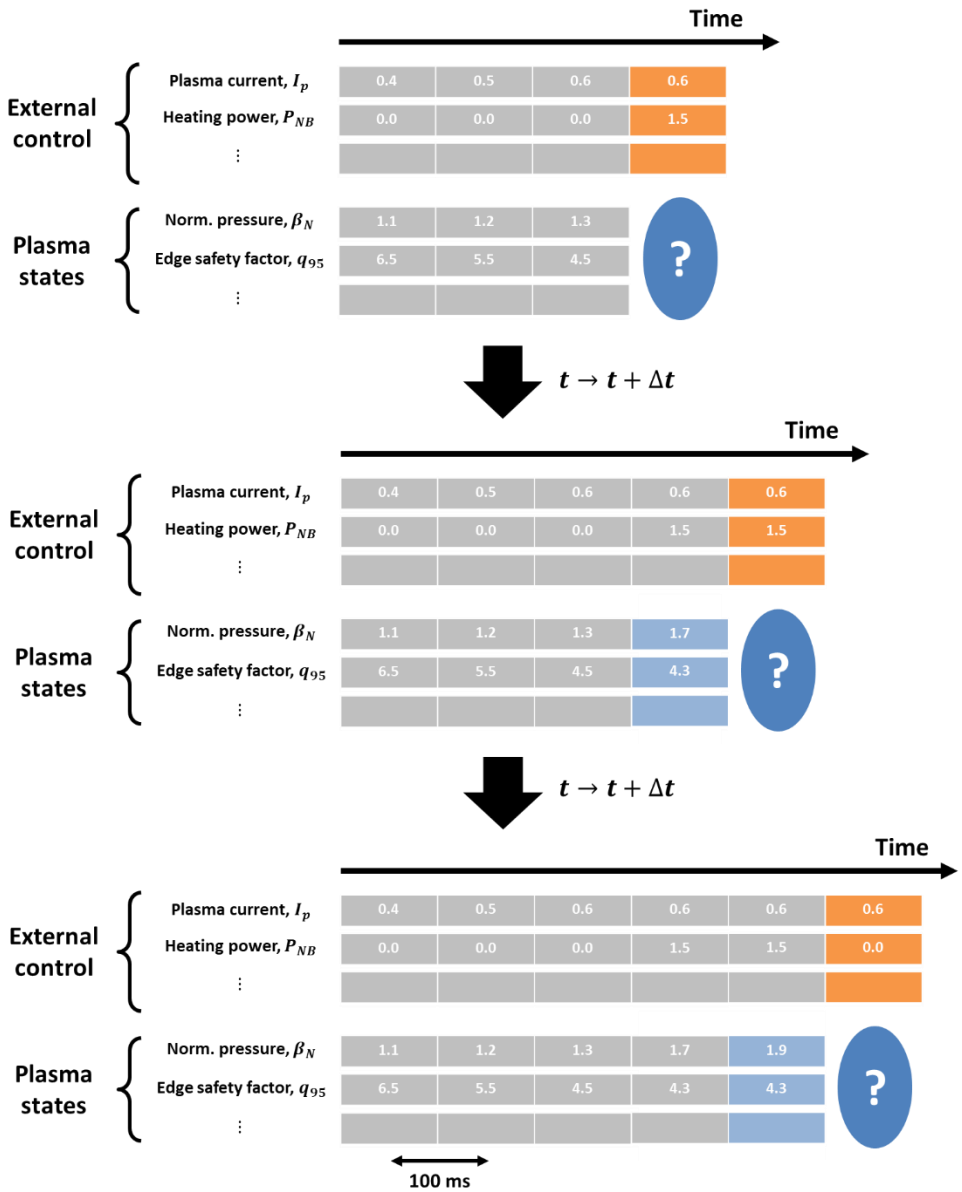


Figure 2.1 Sequentially predicting process of the plasma state with LSTM-based model. After the prediction of a single step, the predicted plasma states are used for the next step prediction.

Figure 2.1 shows the process of predicting the plasma states using the LSTM-based DNN model. As input features of the model,

the sequences of the external control variables and the previous plasma state variables are given. Then, the model predicts the new plasma state after $\Delta t = 100$ ms. The detailed descriptions of the input and output variables are listed in Table 2.1.

Table 2.1 The input and output variables used in the DNN model to predict the plasma state.

Input variable	Description	Mean (Standard deviation)
$\boldsymbol{I_p(:t)}$	Plasma current [MA]	0.51 (±0.11)
$\boldsymbol{B_t(:t)}$	Toroidal magnetic field [T]	1.98 (±0.30)
$\boldsymbol{\overline{n_e}/n_{GW}(:t)}$	Greenwald density fraction	0.31 (±0.09)
$\boldsymbol{P_{NB1A}(:t)}$	Power of NBI (NB1A) [MW]	0.81 (±0.71)
$\boldsymbol{P_{NB1B}(:t)}$	Power of NBI (NB1B) [MW]	0.66 (±0.73)
$\boldsymbol{P_{NB1C}(:t)}$	Power of NBI (NB1C) [MW]	0.45 (±0.69)
$\boldsymbol{P_{EC2}(:t)}$	Power of 2nd EC [MW]	0.16 (±0.29)
$\boldsymbol{z_{EC2}(:t)}$	z at 2nd EC resonance [cm]	0.85 (±7.71)
$\boldsymbol{P_{EC3}(:t)}$	Power of 3rd EC [MW]	0.06 (±0.18)
$\boldsymbol{z_{EC3}(:t)}$	z at 3rd EC resonance [cm]	−0.11 (±4.71)
$\boldsymbol{R_{maj}(:t)}$	Geometric major radius [m]	1.76 (±0.02)
$\boldsymbol{a_{min}(:t)}$	Minor radius [m]	0.47 (±0.02)
$\boldsymbol{\kappa(:t)}$	Elongation	1.60 (±0.19)
$\boldsymbol{\delta_u(:t)}$	Upper triangularity	0.33 (±0.15)
$\boldsymbol{\delta_l(:t)}$	Lower triangularity	0.57 (±0.27)
$\boldsymbol{div(:t)}$	Diverted (1) / limited (0)	0.66 (±0.47)
$\boldsymbol{yr(:t)}$	Campaign year	2018.21 (±1.77)
$\boldsymbol{\beta_N(:t-\Delta t)}$	Norm. plasma beta	1.26 (±0.72)
$\boldsymbol{q_{95}(:t-\Delta t)}$	Safety factor at $\psi_N = 0.95$	5.29 (±1.61)
$\boldsymbol{q_0(:t-\Delta t)}$	Safety factor at $\psi_N = 0$	1.67 (±0.97)
$\boldsymbol{l_i(:t-\Delta t)}$	Norm. internal inductance	1.11 (±0.24)
Output variable	Description	Mean (Standard deviation)
$\boldsymbol{\beta_N(t)}$	Norm. plasma beta	1.26 (±0.72)
$\boldsymbol{q_{95}(t)}$	Safety factor at $\psi_N = 0.95$	5.29 (±1.61)
$\boldsymbol{q_0(t)}$	Safety factor at $\psi_N = 0$	1.67 (±0.97)
$\boldsymbol{l_i(t)}$	Norm. internal inductance	1.11 (±0.24)

The DNN model's output variables are the plasma state parameters, $\{\beta_N(t), q_{95}(t), q_0(t), l_i(t)\}$, which have been determined to sufficiently cover the internal plasma states. The input variables are composed of the plasma current, the toroidal magnetic field, the Greenwald fraction of the line-averaged electron density, the information of NBI and EC actuators, and the plasma geometry. Even though there are additional actuators affecting the plasma, such as the 3D magnetic coils or impurity-seeding system, they were excluded to reduce the complexity of the model training, which will be discussed in detail later again. Especially, the plasma density is determined by the combination of multiple fueling systems and pumping methods. However, those quantities are not stored carefully in the data server, and the fueling and pumping performance is different according to the technical condition, even with the same operating voltage. Therefore, that information has been capsulized with a measurable variable, line-averaged density.

In practice, KSTAR plasmas often show different performances each year, even with the same discharge setting. The wall conditions, radio-frequency antenna settings, port usage, and measurement calibration are different each year, and those factors significantly affect the plasma behaviors. The input variable, yr, in Table 2.1 can provide information about those potential year-dependent factors during the training.

The input variables also include the previous history of the plasma states ($\beta_N(:t-\Delta t), q_{95}(:t-\Delta t), q_0(:t-\Delta t)$, and $l_i(:t-\Delta t)$). This way allows the model to predict the next plasma response considering the current plasma state. For example, even under the same external control, the next response can vary depending on

whether the current plasma is in an L−mode or an H−mode condition. By including the previous plasma state variables into the inputs, the model can infer that information. In practical usage for predictive simulation, the model predicts the plasma state at $t + \Delta t$, and then the predicted state variables are fed into new inputs for the next prediction at $t + 2\Delta t$. This sequential prediction allows the simulator to predict the plasma response in a complete discharge from a given initial sequence of the plasma state. The initial sequence can be obtained from the evolution data of the ramp−up phase of a reference discharge or guessed data from an additional model.

However, this sequential prediction with the input structure shown in Figure 2.1 and Table 2.1 has a drawback when a small error occurs during the prediction. The actual plasma state does not change much in 100 ms in most experimental data, especially in the saturated flattop phase. Therefore, the model trained with those data tends to just copy the previous state, rather than properly predicting the physical response using the external control information. Furthermore, once the model makes an erroneous prediction, it can continuously predict abnormally from the incorrectly predicted input values fed in. In order to prevent this error accumulation, we need to slightly weaken the weights connected with the previous state variables. Weakening the dependency on the state variables allows the model to consider more the external control variables' information for predicting the next plasma response. Therefore, we added a noise layer right above the previous state variables, as shown in Figure 2.2. The noise layer induces the model to predict with more consideration of the external control parameters, not just copy the previous state.

C. Wan et al. [36] have used a different technique to deal with the experimental data to train the DNN model. Since the plasma state changes less in the flattop phase than in the ramp-up or the ramp-down phases, they resampled the experimental data with a different rate between the flattop and the other phases. For example, the loop voltage data is resampled ten times less in the flattop phase than in the ramp-up phase. This method can reduce the data imbalance due to the different data patterns in different phases and alleviate the excessive weights on the unintended signals, which is overfitting. However, most of the important physical phenomena and high performance of the fusion plasma appear in the flattop phase, and the plasma in the flattop still continuously evolves even though it is slower than in the ramp-up phase. The reduced resampling rate in the flattop phase may induce the loss of physically meaningful experimental data. The noise layer used in our work can adjust the weights of the DNN model without reducing the information in the original dataset.

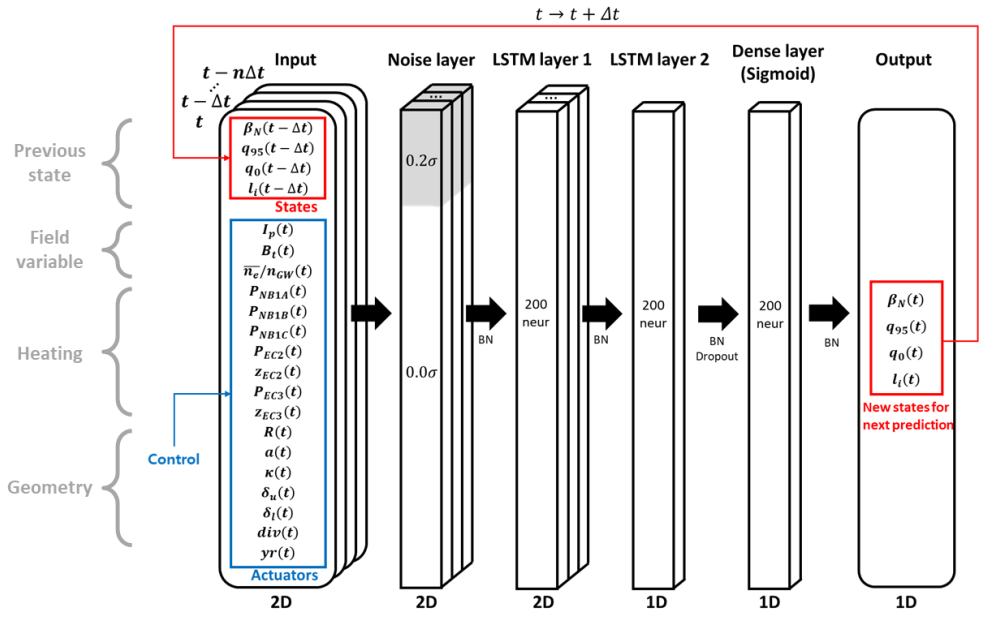


Figure 2.2 The model diagram for predicting the plasma state

Figure 2.2 shows the model diagram used in this work. The network comprises input/output layers, a noise layer, and three hidden layers (two LSTM and one dense layer) with 200 neurons each. The input variables, consisting of n steps of time-series data, pass through the noise layer before being fed into the first LSTM layer. The noise layer adds random Gaussian noise with an amplitude of 0.2 standard deviations to four state variables (red box in the input layer in Figure 2.2) without perturbing the other inputs. An appropriate noise level prevents excessive dependence on the previous step's state variables during the training. If the noise level is too large, the model does not reflect the previous states' information. Therefore, the noise level, 0.2, was determined carefully after several scans. After the noise layer, the data pass through three hidden layers with batch normalization between adjacent layers. In

order to reduce the overfitting of the model, the dropout regularization method [37] was used before the last hidden layer, and ensemble-averaged prediction with ten networks was applied during its practical usage. The L2 regularization [38] has also been applied to hidden layers. The network was trained by the Adam scheme with a loss function of mean-squared errors.

For the training of the model, we gathered five-year experimental data (2016 to 2020) in KSTAR. In order to reduce the model's complexity and overfitting, we filtered out several discharge cases. The impurity (Ar, Ne, or Kr)-seeded discharges were excluded and the discharges with resonant or non-resonant magnetic perturbation by 3D coils were also filtered from the dataset. The discharges that were disrupted too early ($t < 1.7\,\mathrm{s}$) and the long-pulse phase ($t > 20\,\mathrm{s}$) were also excluded to prevent data imbalance problems. Eventually, 330,250 time-slices of the data composed of 5,487 L- and H-mode discharges have been gathered with all input and output variables available. Here, the data at each time slice were averaged for 100 ms to avoid the outlier points due to the transient instability, actuator's blip, and the diagnostics' errors. The gathered data were split into training and test sets at a ratio of 8:2 to assess the overfitting of the model. After training the model for 100 epochs, the accuracy of the model becomes saturated near $R^2 \approx 0.98$ for the test dataset, as shown in Figure 2.3.

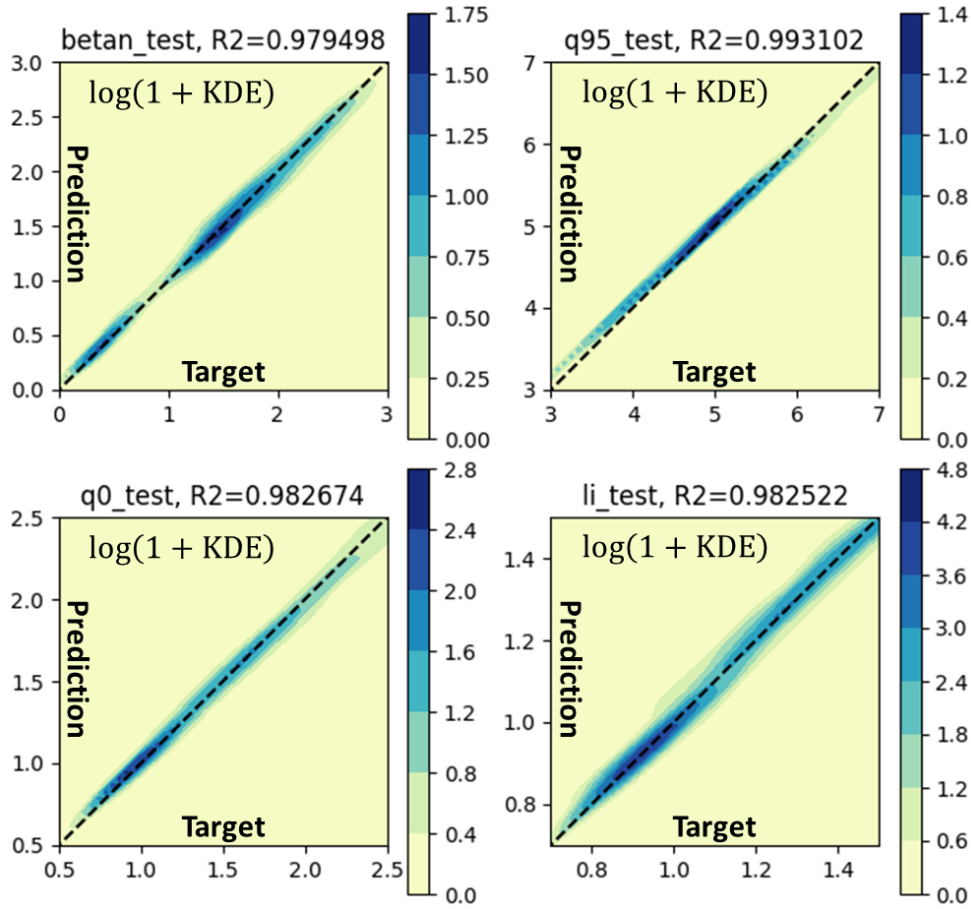
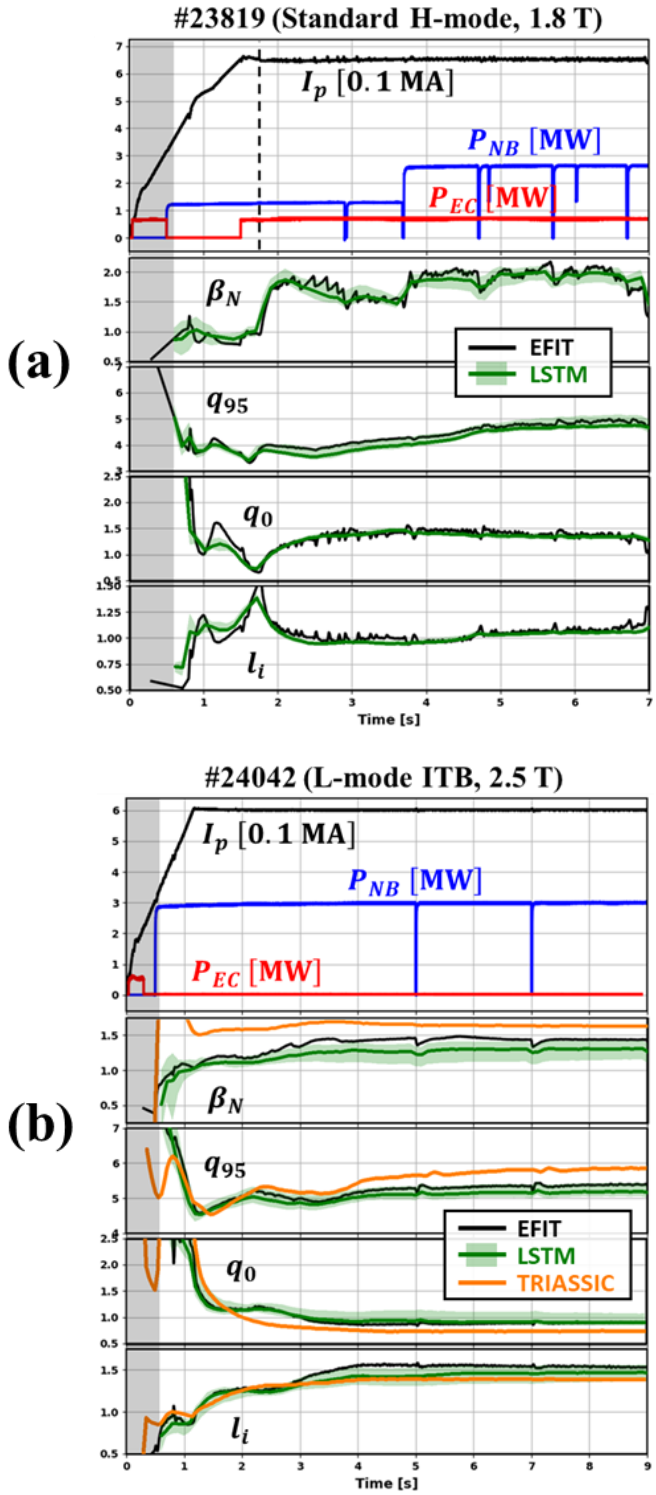


Figure 2.3 The ensemble regression of the trained model for four plasma parameters.

2.1.2. Demonstration and validation



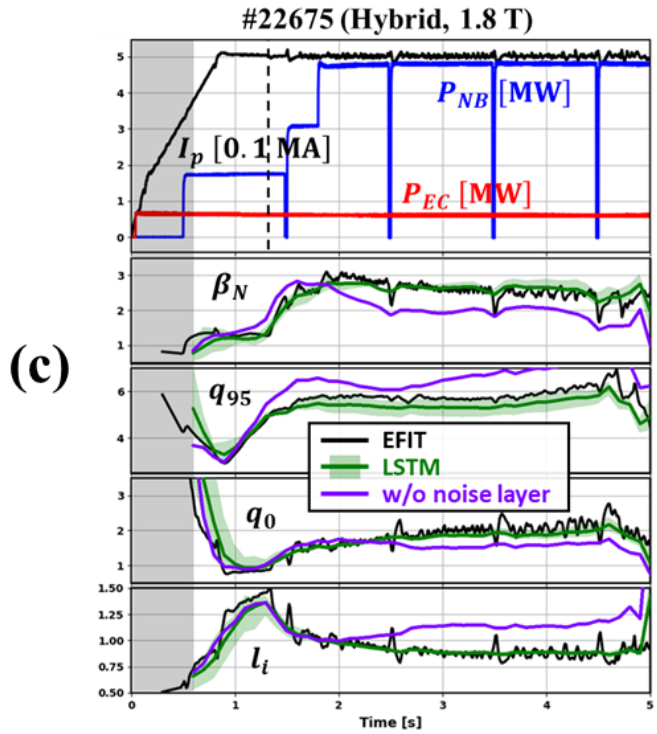


Figure 2.4 Test prediction for different operation scenario discharges.

Figure 2.4 shows the test prediction using the trained model for different operation scenario discharges; Standard H-mode (Figure 2.4 (a), #23819), L-mode internal transport barrier (ITB) (Figure 2.4 (b), #24042), and hybrid scenario (Figure 2.4 (c), #22675). The top graph of each case shows the external controls, and the rests show the plasma responses obtained with EFIT (black lines). The model successfully predicts the plasma evolution in different scenarios, as shown with green lines in Figure 2.4. In (b), a physics-based simulation result using an integrated modeling suite, TRIASSIC [29], is also shown in orange for comparison. The data-driven prediction with the DNN model provides a more experiment-relevant prediction than the physics-based simulation. The result from a DNN

model trained without the noise layer is indicated with purple in (c). Without the noise layer, the model has a strong weight on the previous plasma state, and the prediction errors accumulate over time due to the errors being fed into the inputs continuously.

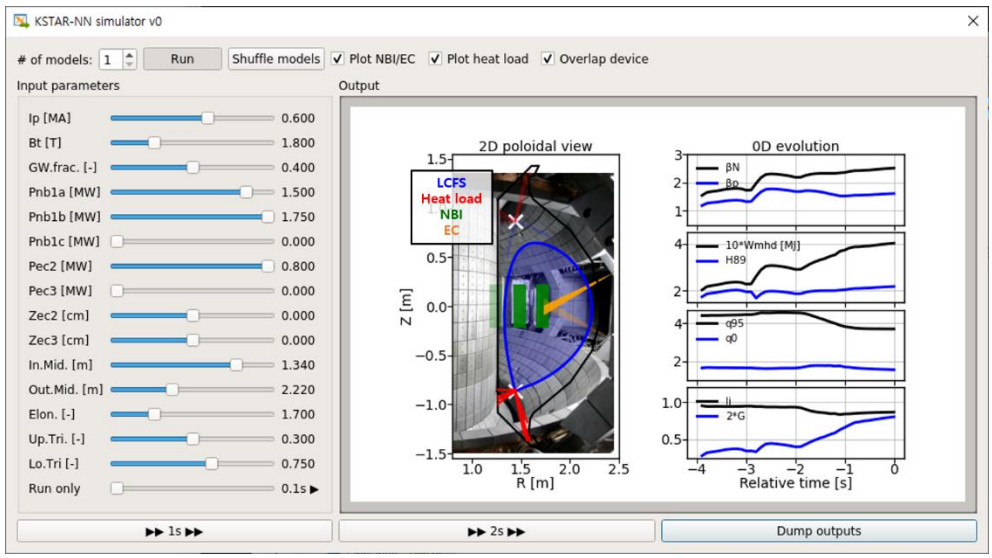


Figure 2.5 Interactive graphical user interface for predictive modeling with the data-driven model.

Using the plasma boundary modeling and the predictive modeling for the internal plasma state developed in this chapter, an interactive graphical user interface (GUI) has been developed, as shown in Figure 2.5. The controllable input parameters shown on the left in Figure 2.5 are described in Table 2.1. The PyQt5 framework [42] was used for materializing the interface. The executable code script with trained weights can be seen in the GitHub repository [43].

This achievement is the first development of an end-to-end

tokamak simulator where we can perform a virtual interactive experiment in real-time. Because we can visually observe the plasma behavior as the input parameters are adjusted in real-time with this tool, it would be useful for educational purposes as well as physics research. In Chapter 3, this simulator will be used as a training environment for RL.

2.2. Analysis for model reliability

2.2.1. Uncertainties in dataset

In actual tokamak experiments, some factors are not described by the input variables proposed in Table 2.1. For example, the tokamak wall condition can differ for each shot due to the impurity or Hydrogen retention on the Carbon wall, and that information is not stored in data. The particle fueling and pumping methods also have a high degree of freedom. Furthermore, the diagnostics or the plasma reconstruction method like EFIT inherently has about 10 % inaccuracies [27]. These factors cause data uncertainties that can disturb the training of the DNN model. We need to be careful not to overfit the model to the data with many uncertainties.

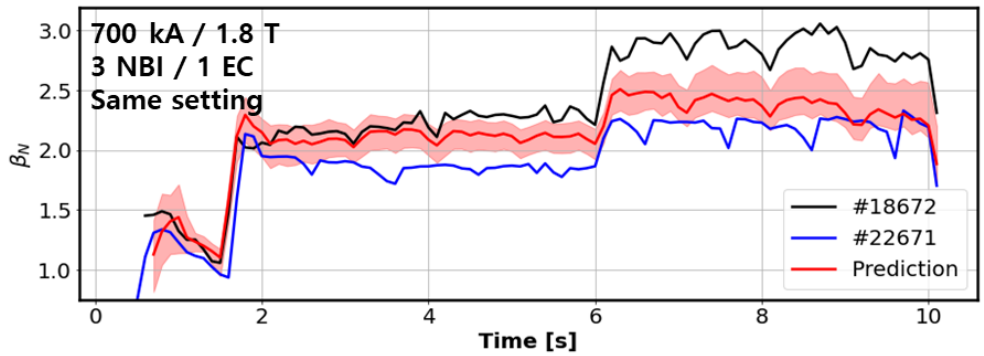


Figure 2.6 The different evolution of β_N in almost the same discharges, #18672 and #22671.

Figure 2.6 shows two discharges in KSTAR, which were conducted under almost the same control setting but showed quite different plasma evolution. In a situation with such uncertainty, the model can overfit a slight input difference in order to minimize the prediction errors in those two discharges.

In order to prevent overfitting of the model, we have applied (i) the dropout technique, (ii) the noise layer, (iii) the L2 regularization, and (iv) the ensemble average during the training. As a consequence, the model could predict a reasonable and averaged prediction for #18672 and #22671, as shown with the red line in Figure 2.6.

2.2.2. Consistency with prior knowledge

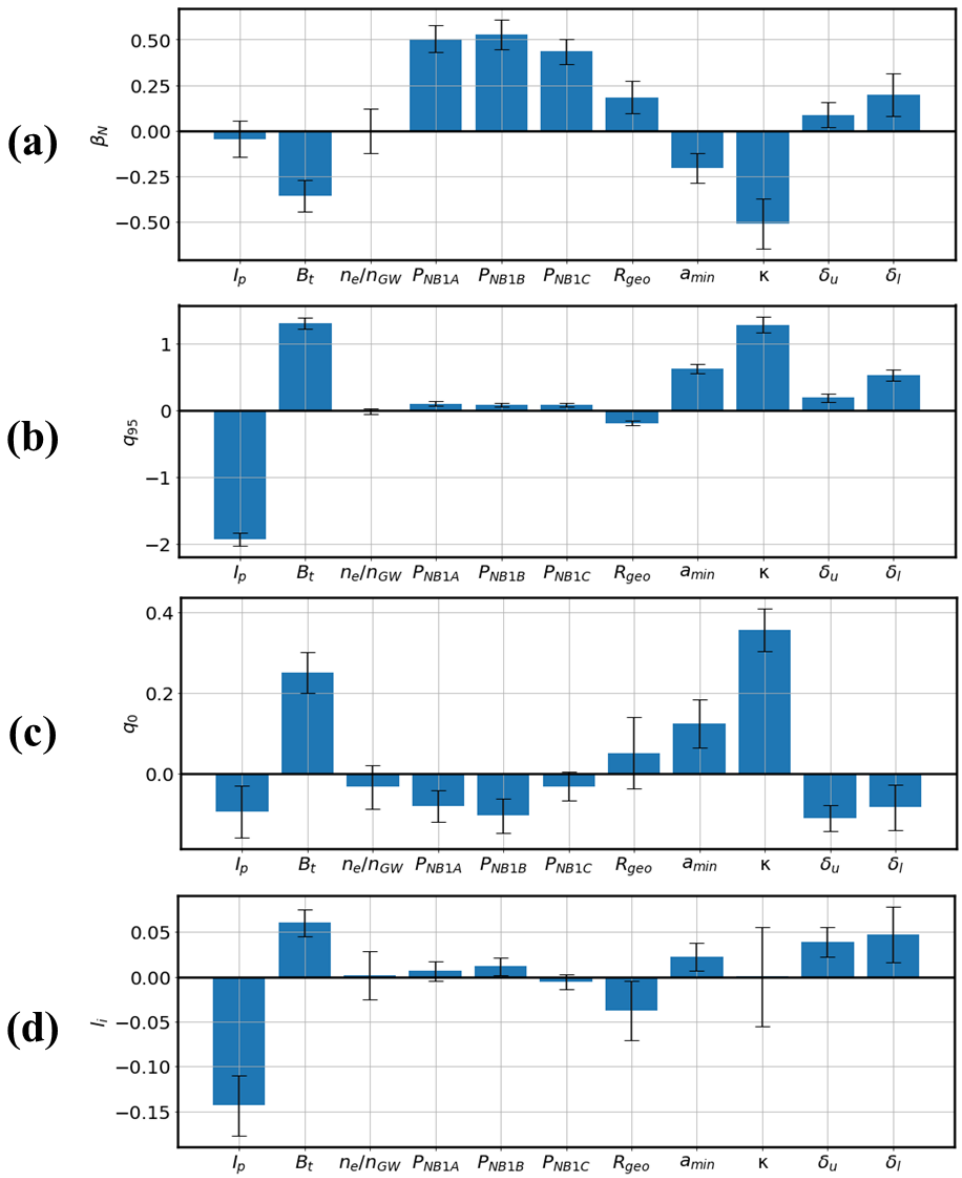


Figure 2.7 Sensitivity analysis with several input variables for four output variables, (a) β_N, (b) q_{95}, (c) q_0, and (d) l_i.

Figure 2.7 shows the sensitivity scan of several input variables for each output variable's evolution. For example, in (a), β_N

increases if the NBI powers ($P_{NB1A,B,C}$) increase. Empirically, we have prior knowledge that the NB1B is the most efficient and the NB1C is the least. The sensitivity scan indicates that the model could also learn this empirical knowledge from the pattern of the experimental data. Other input sensitivity results, such as the relation between the plasma performance and the triangularity and the boundary shape dependency on the edge safety factor, also agree with physical and empirical prior knowledge. These results indicate that the data-driven model is reliable to simulate the actual plasma response in KSTAR.

Chapter 3

Operation trajectory design algorithm

In Chapter 2, a data-driven simulator that predicts the plasma response in KSTAR with sufficient speed and reliability has been developed. In this chapter, we train an artificial agent that designs the tokamak operation trajectory on the developed simulator with the RL technique.

In order to achieve a physical objective or to conduct an experiment for observing the desired phenomenon, we have to design the tokamak operation trajectory to reach the target plasma state. Conventionally, the trajectory of the operation parameters, such as the plasma current and boundary shape, has been designed by humans within the feasible engineering range. It requires trials and errors or massive simulations to find an appropriate trajectory to achieve each physical target we want. We had to scan different operation trajectories of various engineering parameters in the simulations and verify whether a similar result was achieved in actual

experiments. To reduce these trials and errors, we need an algorithm that suggests a probable tokamak operation trajectory to reach the desired target plasma state. Therefore, we develop an RL-based algorithm that designs the operation trajectory, which replaces the human task of massive simulations.

3.1. Environment of the RL training

In Chapter 2, we developed a virtual KSTAR environment that simulates the plasmas' behavior. With the GUI shown in Figure 2.5, we can conduct a virtual interactive experiment in real-time. However, in order for the RL agent to interact with the environment, such as by getting observations or taking actions, it is necessary to wrap the environment in a specific format that the agent can understand.

OpenAI provides a standard format for the environment, Gym API, with which the RL agent can interact [39]. The Gym environment consists of several key functions. The *reset* function resets the episode of the agent's interaction with the environment, and the *step* function computes the new observation and the reward from the agent's action. Here, we need to define the action, observation, and reward according to our objective for each problem. Figure 3.1 shows the wrapping of the data-driven KSTAR simulator developed in Chapter 2 with Gym environment format.

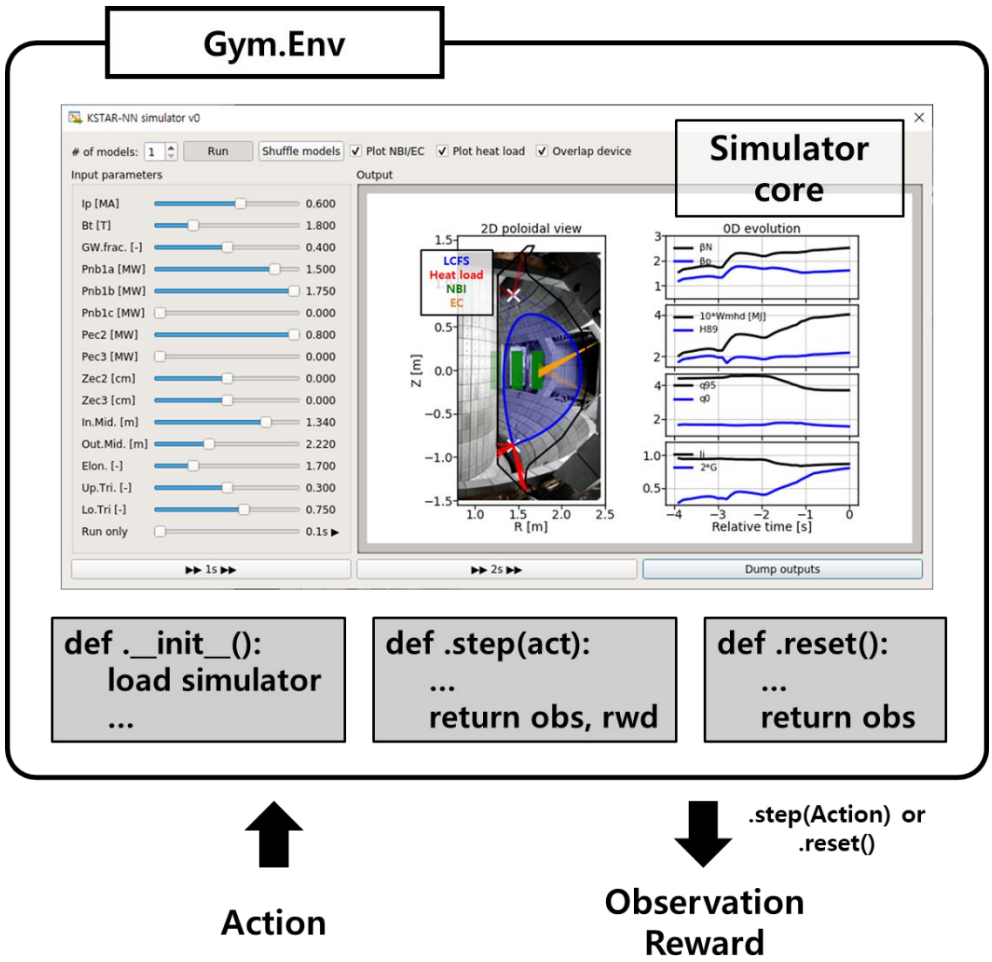


Figure 3.1 Wrapping the KSTAR simulator into a Gym environment format for RL agent to understand.

The RL agent that determines the action from the given observation is constructed with the twin delayed deep deterministic policy gradient (TD3) [40] implementation from Stable Baselines [41]. TD3 is a successor of a prominent RL algorithm, DDPG [42]. TD3 made up for several problems regarding Q-value overestimation and policy overfitting arising in DDPG, by taking the minimum Q-value between two critic networks' estimations, delaying the actor-

network update, and adding clipped noise to the determined action. All the actor and critic networks in this work consist of two hidden layers of 400 and 300 neurons each.

3.2. Control of normalized beta

The normalized beta, β_N, is the plasma pressure parameter normalized by the MHD stability limit [43]. For high fusion gain under the given condition, it is necessary to achieve a high value of β_N. In this section, we train an RL agent to reach a given target value of β_N. The agent adjusts the tokamak operation trajectory from the given reference shot in KSTAR to achieve the target.

3.2.1. The action, observation, and reward for RL

Each episode during the RL training, the agent determines the action, the control knobs for the operation trajectory, and observes the achieved β_N. After the episode, it receives a reward according to how close the achieved β_N is to the given target. To prevent diverging of the reward value when the achieved state is too close to the target, we clipped the reward not to exceed a limit, ε^{-1}, where $\varepsilon = 0.01$. Detailed descriptions of the action, observation, and reward for the RL training are shown in Table 3.1.

Table 3.1 The action, observation, and reward for the RL training. The control knobs and the target (β_{target}) vary for each case.

RL variables	Description
Action	Control knobs for each case ($I_p, P_{NB}, \kappa, \delta_{u,l}, \ldots$)
Observation	β_N
Reward	$\min\left(\left\|\beta_N - \beta_{target}\right\|^{-1}, \varepsilon^{-1}\right)$

In this section, we set three different targets and control knobs to determine the operation trajectory. According to the difficulty of achieving the given target, we set more control knobs that the agent can control. The target values of β_N and the control knobs available for each case are shown in Table 3.2.

Table 3.2 The available control knobs for three different targets.

	Reference β_N	Target β_N	Control knobs
Case 1 #26411	2.1 (#25672)	2.4, 1.8	Elongation (κ), Triangularities ($\delta_{u,l}$)
Case 2 #26719	2.1 (#25672)	2.7	3rd NBI power (P_{NB1C}), Elongation (κ), Triangularities ($\delta_{u,l}$)
Case 3 #26413	3.0 (#25894)	3.5	Plasma current (I_p), Plasma density ($\overline{n_e}/n_{GW}$), 3rd NBI power (P_{NB1C}), Elongation (κ), Triangularities ($\delta_{u,l}$)

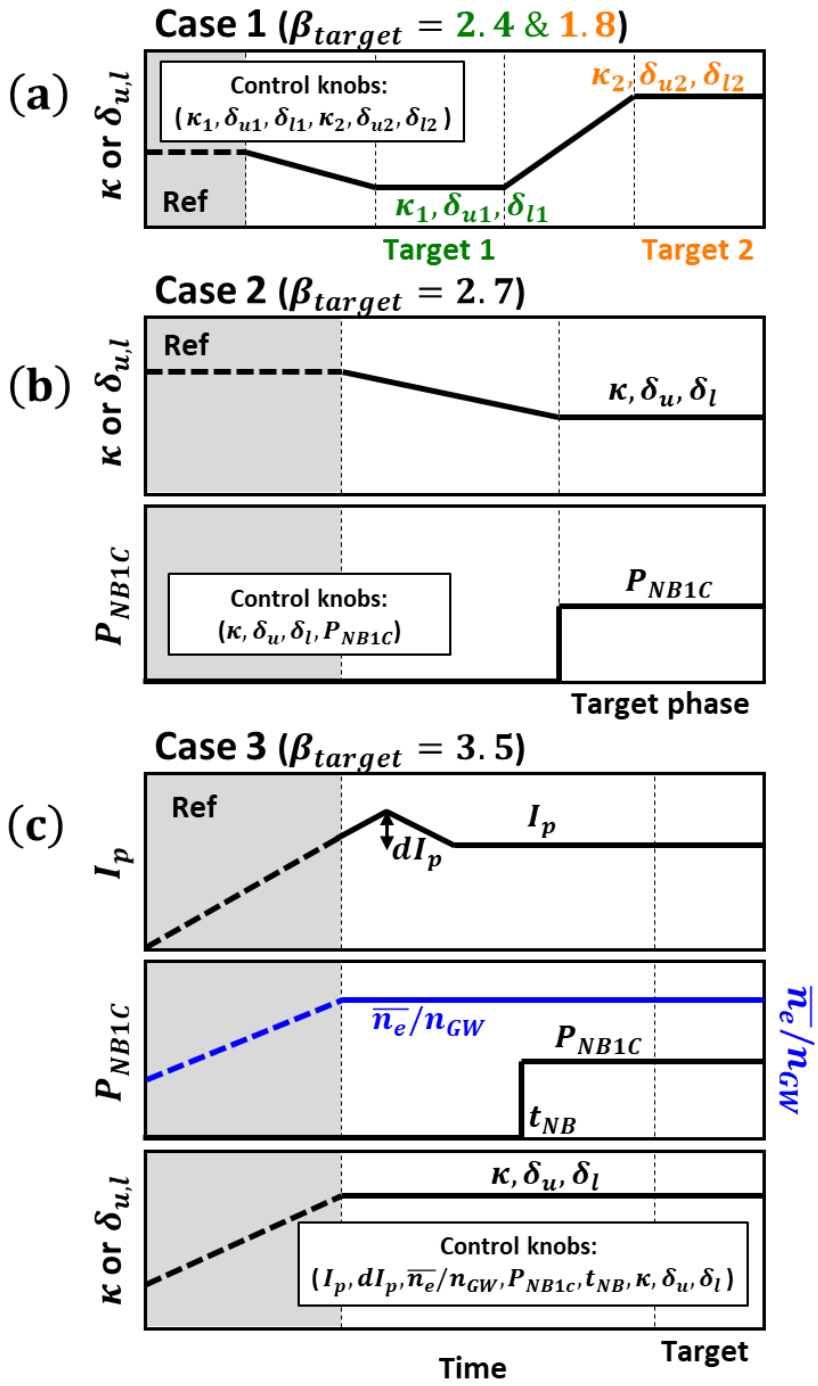


Figure 3.2 Parametrized operation trajectories with different control knobs for each target case.

Figure 3.2 shows the parametrized tokamak operation trajectories whose parameters are determined by the agent during the RL training. The RL agent determines the parametrized operation trajectory as an action in each training episode, then the RL environment shown in Figure 3.1 provides a consequent β_N at the target phase.

Case 1 is to control β_N within the normal range by using the plasma shape parameters only. The reference shot, #25672 (and), shows $\beta_N \simeq 2.1$ with two NBI heating sources. Here, the RL agent determines the trajectory of the shape parameters, κ, δ_u, and δ_l under the same heating condition to control β_N to 2.4 and 1.8 sequentially, as shown in Figure 3.2 (a). Case 2 is to achieve a higher performance target, $\beta_N = 2.7$, with an additional control knob of the 3rd NBI power, limited to a maximum of 1.15 MW. In Case 3, we try to achieve $\beta_N = 3.5$ with more complicated operation trajectory. $\beta_N = 3.5$ has been a challenging target in KSTAR, and we selected it to understand the limitation of this work.

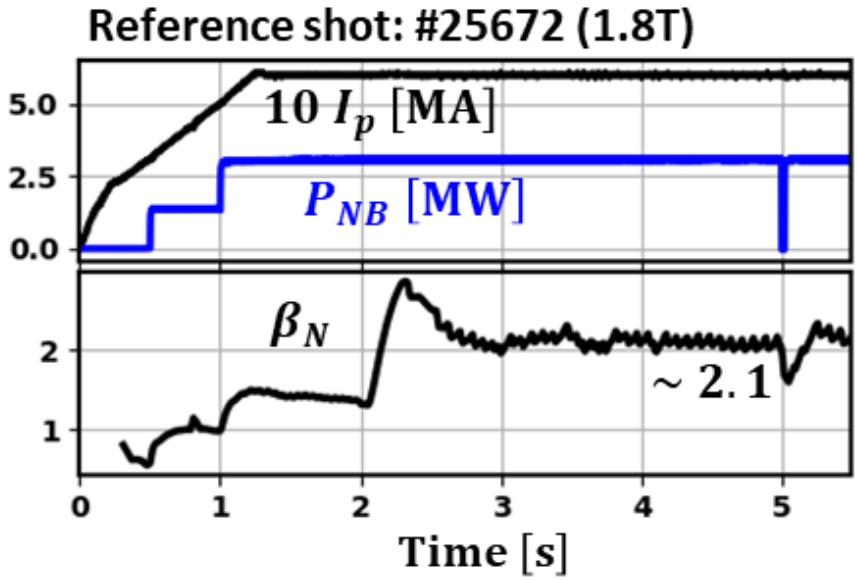


Figure 3.3 The overview of the reference shot, #25672.

3.2.2. RL training

Table 3.3 The converged RL solution after training.

	Ref β_N	Target β_N	Control knobs	Converged solutions
Case 1 (#26411)	2.1 (#25672)	2.4, 1.8	Elongation, κ Upper triangularity, δ_u Lower triangularity, δ_l	$\kappa = 1.65, 1.85$ $\delta_u = 0.21, 0.38$ $\delta_l = 0.65, 0.61$
Case 2 (#26719)	2.1 (#25672)	2.7	3rd NBI, P_{NB1C} Elongation, κ Upper triangularity, δ_u Lower triangularity, δ_l	$P_{NB1C} = 1.15$ MW $\kappa = 1.66$ $\delta_u = 0.23$ $\delta_l = 0.79$
Case 3 (#26413)	3.0 (#25894)	3.5	Plasma current, I_p Line−averaged density, $\overline{n_e}/n_{GW}$ 3rd NBI, P_{NB1C} Elongation, κ Upper triangularity, δ_u Lower triangularity, δ_l	$I_p = 0.58$ to 0.5 MA $\overline{n_e}/n_{GW} = 0.48$ $P_{NB1C} =$ 1.15 MW at $t =$ 2 s $\kappa = 1.65$ $\delta_u = 0.34$ $\delta_l = 0.84$

After a few 10^4 episodes of the RL training, the agent determines a converged solution of the action variables to achieve the target β_N in each case. The converged results are shown in Table 3. With these converged RL-determined operation trajectories, we will conduct the KSTAR experiments in Chapter 4.

3.3. Simultaneous control of multiple parameters

We have seen that the control of only β_N occasionally yields MHD instabilities in the experiment due to a lack of information regarding the MHD stabilities in the simulator. In this section, we develop an RL agent that designs the operation trajectory for multiple physical parameters' control. By controlling multiple parameters simultaneously, we can constrain the plasma state more specifically in terms of the stability or the magnetic structure. This algorithm calculates the trajectory of the plasma current and boundary shape to control β_p, q_{95}, and l_i into arbitrarily given targets in the feedforward process. Therefore, it can provide the tokamak operation trajectory to achieve a specific plasma state required for advanced operation scenarios or detailed physics experiments.

3.3.1. The action, observation, and reward for RL

In the environment described in Section 3.1.2, the RL agent determines the next tokamak controls (action) for each episode, and the simulator delivers the resulting plasma response (observation).

The agent gets a reward depending on how well the given target is achieved by the control at each episode. Detailed descriptions of the action, observation, and reward are shown below.

$$Action = \left(I_p, \kappa, \delta_u, \delta_l, R_{in}, R_{out}\right)_{new} \tag{3.1}$$

$$Observation = \left(I_p, \kappa, \delta_u, \delta_l, R_{in}, R_{out}\right)_{old}, \left(\beta_p, q_{95}, l_i\right)_{old}, \left(\beta_p, q_{95}, l_i\right)_{target}, \left(P_{NB1A}, P_{NB1B}, P_{NB1C}\right) \tag{3.2}$$

$$Reward = -\mathrm{RMS}\left(\frac{y_{new} - y_{target}}{\epsilon_y}\right)_{y=\beta_p, q_{95}, l_i} \tag{3.3}$$

$$\left(\epsilon_{\beta p}, \epsilon_{q95}, \epsilon_{li}\right) = (0.2, 0.5, 0.05) \tag{3.4}$$

Table 3.4 The description and the bounds of the action variables.

Variable	Description	Lower limit	Upper limit	Rate limit
$\boldsymbol{I_p}$	Plasma current	0.35 [MA]	0.75 [MA]	0.2 [MA/s]
$\boldsymbol{\kappa}$	Elongation	1.68	1.90	0.11 [/s]
$\boldsymbol{\delta_u}$	Upper triangularity	0.2	0.5	0.15 [/s]
$\boldsymbol{\delta_l}$	Lower triangularity	0.5	0.8	0.15 [/s]
$\boldsymbol{R_{in}}$	Major radius at inner midplane	1.265 [m]	1.340 [m]	0.0375 [m/s]
$\boldsymbol{R_{out}}$	Major radius at outer midplane	2.180 [m]	2.290 [m]	0.055 [m/s]

Table 3.5 The description and the bounds at target reset for the state variables.

Variable	Description	Lower limit	Upper limit
$\boldsymbol{\beta_p}$	Poloidal plasma beta	1.1	2.1
$\boldsymbol{q_{95}}$	Safety factor at $\psi_N = 0.95$	3.8	6.2
$\boldsymbol{l_i}$	Internal inductance	0.84	1.06
$\boldsymbol{P_{NB1A}}$	Power of NB1A [MW]	1.15	1.75
$\boldsymbol{P_{NB1B}}$	Power of NB1B [MW]	1.15	1.75
$\boldsymbol{P_{NB1C}}$	Power of NB1C [MW]	0.45	0.6

The RL agent determines the action, composed of plasma current and boundary shape parameters shown in Equation (3.1) and Table 3.4 at a given state. The state variables in Equation (3.2) include the previous action, plasma parameters (β_p, q_{95}, and l_i), and their target values to be achieved. By letting the agent see the current state of β_p, q_{95}, and l_i as well as their target values, it can make flexible decision-making by expecting future plasma response that varies depending on the current state. Additionally, three NBI powers (P_{NB1A}, P_{NB1B}, and P_{NB1C}) are also contained in the state variables. Although the NBI powers are critical actuators that determine the plasma state, they were set as state variables, not action variables, for several reasons; (i) it is relatively trivial to control the plasma performance by increasing or decreasing the heating power, (ii) the available NBI powers for each discharge are sometimes unexpectable relying on the machine condition of the experimental day, and (iii) it is limited to change the NBI power in a single shot in KSTAR. The target values and the NBI powers in Equation (3.2) are randomly reset after each episode in the range within the lower and the upper

limit shown in Table 3.5. The bound values for the target reset were determined to be empirically reliable and not to produce outliers in the original dataset. Equation (3.3) shows the reward function, estimated by the negative root-mean-square (RMS) of the difference between the plasma state and the targets. Therefore, the closer the plasma reaches the target, the higher reward the agent receives. In estimating the reward, each difference of parameter y is normalized by a tolerance ϵ_y, which can be interpreted as a criterion for determining the target achievement. The tolerance values for β_p, q_{95}, and l_i used in this work are shown in Equation (3.4). Since the agent tries to get a higher reward, after enough training, it determines the optimal condition of plasma current and boundary shape to achieve the given targets of β_p, q_{95}, and l_i under the given NBI power condition.

3.3.2. RL training

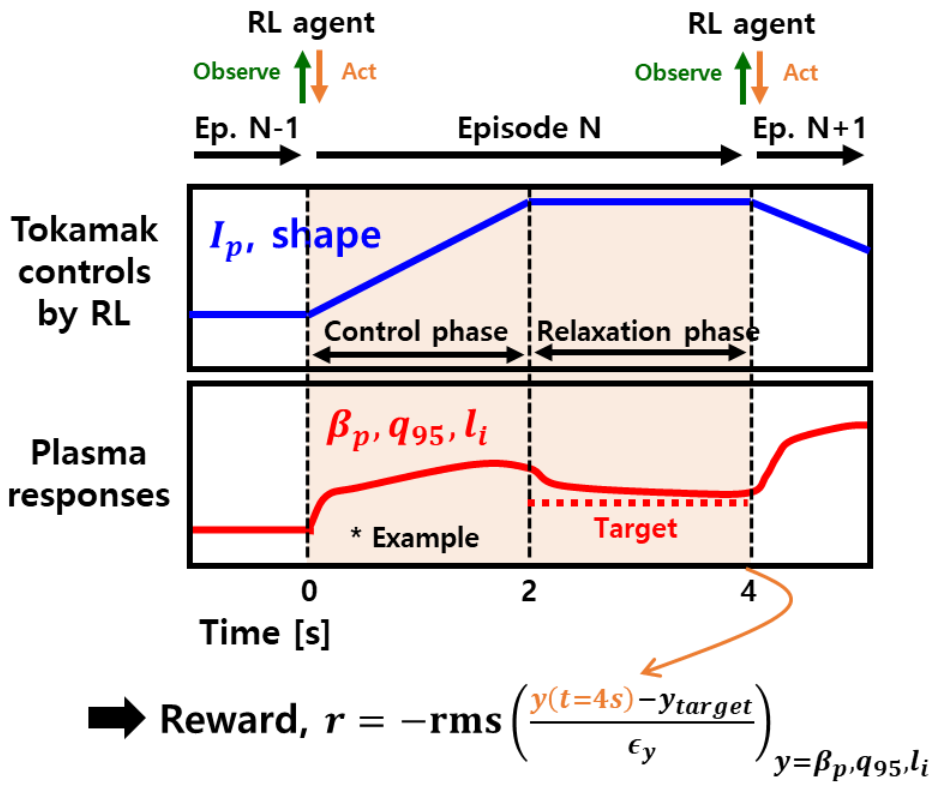


Figure 3.4 The schematic view of a single episode during the RL training.

Figure 3.4 shows the process of one episode in the RL training simulation. At the beginning of each episode, the RL agent observes the plasma state and the control target described in Equation (3.2) and determines a new action (Equation (3.1)) to reach the given target. A single episode consists of a control phase and a relaxation phase. In the control phase, the plasma current and boundary shape vary linearly from the previous values to the new ones determined by the agent at the beginning. In the relaxation phase, it waits for

enough saturation of the plasma state after the plasma transport and flux diffusion. The period for each phase was determined to be 2 s. The current diffusion time scale is typically $\mathbf{O(1)\,s}$ and the time scales for the plasma transport and the fast ion slowing−down are <1 s in KSTAR, which means the period for the relaxation phase is physically sufficient to reach a nearly stationary state. At the end of the relaxation phase, the reward is estimated by Equation (3.3), and the state−action−reward data is stored in the memory buffer for the RL agent's learning. After an episode, the target values for β_p, q_{95}, and l_i, and also the available NBI powers are randomly reset for the next episode. The randomly selected target state of β_p, q_{95}, and l_i might not always be physically feasible even though each possible range is bounded within the empirically reliable range shown in Table 3.5. However, by setting the reward as a continuous function of the distance from the target, the RL agent can still learn the relation between the target and reward even if the target is outside the achievable range.

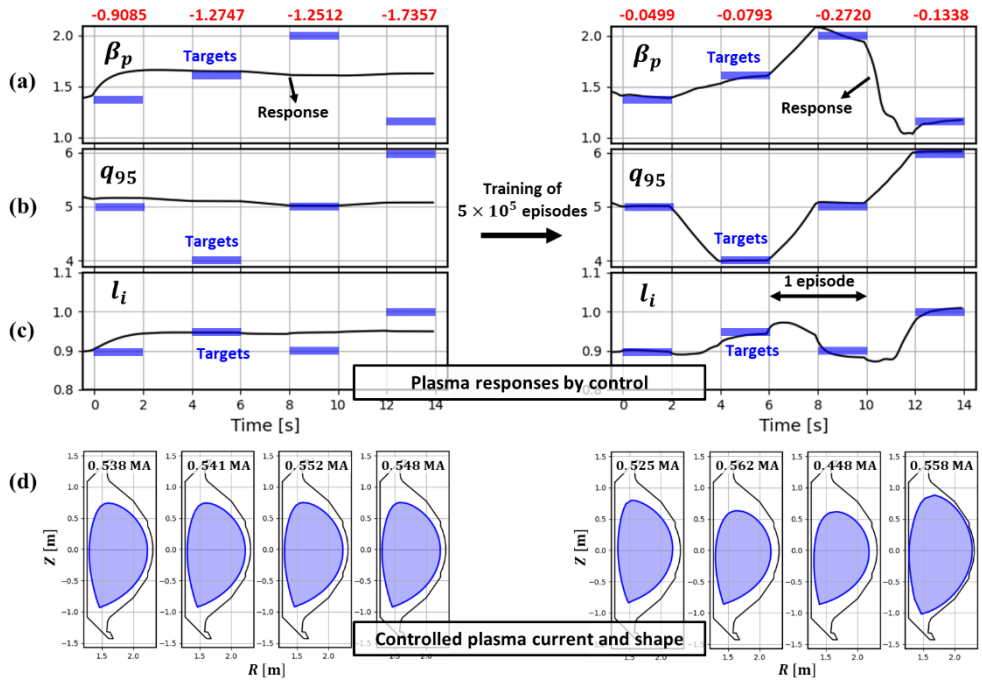


Figure 3.5 The RL-determined operation trajectory and the plasma responses for various targets. The left is the results before training, and the right is those after training.

Figure 3.5 shows sampled operation trajectories designed by the RL agent and the consequent plasma responses before (left) and after (right) the training. Figure 3.5 (a-c) shows the evolution of the plasma state parameters (black) and the given targets (blue). The reward that the RL agent receives for each target is written in red above (a). The plasma current and boundary shape determined by the agent for each target is shown in (d). Before the training, the neural networks composing the agent were not optimized, which means that the agent did not have any knowledge of the relation between the targets and the decision-making. Therefore, it determined meaningless actions that are almost the same for

different states and targets, which yielded relatively low reward values. As the training goes on, the RL agent does random exploration and encounters various targets, and then, it gradually learns which action led to a high reward through those experiences. After training for half a million episodes, the AI dynamically changes the plasma current and boundary shape to reach different target values, as shown on the right side of Figure 3.5. It requires a complicated combination of action variables' determination to match three parameters into the targets simultaneously, and the AI successfully finds out the appropriate condition of the action variables for different targets.

Chapter 4

Validation in KSTAR

In this chapter, we present the actual experiment results on KSTAR, which were conducted with the RL-designed operation trajectories shown in Chapter 3. With the experiment results, the control accuracy and the prospective of the RL methodology will be discussed.

4.1. Control of normalized beta

4.1.1. Case 1: β_N control to 2.4 and 1.8

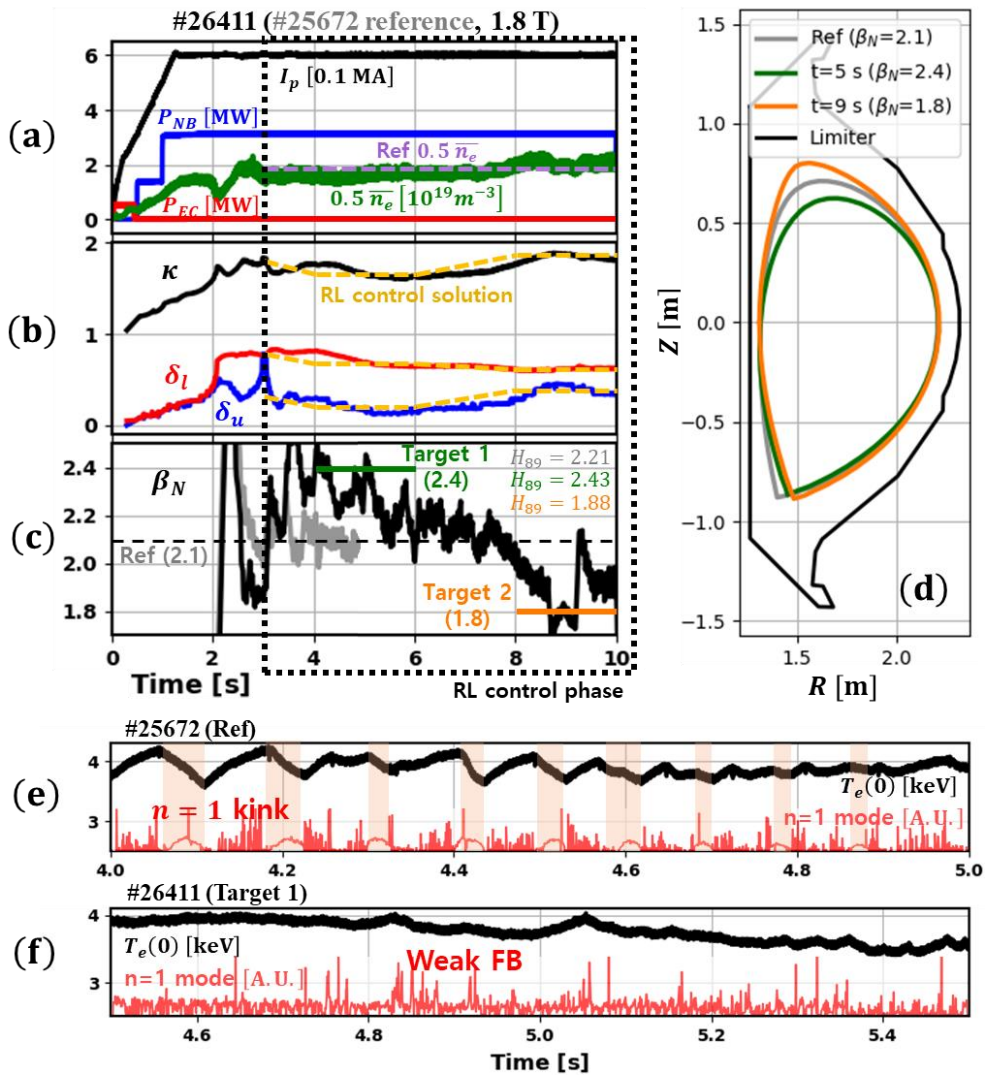


Figure 4.1 The KSTAR discharge (#26411) conducted with the RL-determined operation trajectory for Case 1.

Figure 4.1 shows the KSTAR discharge, #26411, conducted with the RL-determined operation trajectory for Case 1 shown in Table 3.2. The magnetic field, the plasma current, and the heating powers are identical to the reference shot, #25672, which has shown $\beta_N \simeq 2.1$. The density level is also kept as close as possible to the reference shot (purple dashed line in Figure 4.1 (a)) during the discharge. The target values of β_N are set with $\beta_N = 2.4$ for phase 1 and $\beta_N = 1.8$ for phase 2, and only the plasma shape is adjusted under the fixed heating condition. Figure 4.1 (b) shows the plasma shape parameters being controlled to the RL solution (yellow dashed line) to achieve the target beta. In Figure 4.1 (c), the time-evolutions of β_N are shown with the grey and the black line for #25672 (the reference shot) and #26411, respectively. In the reference shot, $\beta_N \simeq 2.1$ is sustained until $t \sim 5$ s. However, by adjusting the plasma shape with the RL solutions in #26411, β_N increases up to the first target, 2.4, at t=5 s, and decreases to the second target, 1.8, at t=9 s. The adjusted plasma boundary shapes at the target phases are shown in Figure 4.1 (d), with the grey line for the reference shot ($\beta_N = 2.1$), the green line for the first target ($\beta_N = 2.4$), and the orange line for the second target ($\beta_N = 1.8$). β_N oscillates near the target values probably because there are some delayed or overshooting responses in the shape control with the feedback control algorithm using real-time EFIT in KSTAR. In addition, too fast shape variation could also result in the plasma control being unstable. Nonetheless, we can see that the trend of β_N approximately follows the target we set.

The confinement enhancement factor, H_{89}, estimated in each phase is written in Figure 4.1 (c). H_{89} is increased up to 2.43 in the

first target phase and reduced to 1.88 in the second target phase. This indicates that the β_N change is not simply due to its inverse proportionality to the volume but due to the confinement enhancement by the boundary shape adjustment. It is noteworthy that the reference shot is an H-mode discharge with core electron temperature oscillations by n = 1 kink (Figure 4.1 (e)), but the first target phase in #26411 is in the regime without n = 1 kink (Figure 4.1 (f)). It is hard to evaluate the operation regime with q-profiles since dedicated diagnostics were unavailable in this experiment. However, this discharge is likely a hybrid regime since it shows the typical characteristics of KSTAR hybrid scenarios of $\beta_N \sim 2.4$ and $H_{89} \gtrsim 2.0$ with fishbone activity instead of sawtooth [10] (See also Figure 4.3 for comparison with other hybrid scenarios in KSTAR.). It shows that the RL operation design leads to access to the hybrid-like regime favorable for the higher β_N target under the same heating condition, even though we did not provide this prior knowledge while training the RL agent. These results imply that the RL operation design can also provide clues for advanced scenario development.

4.1.2. Case 2: β_N control to 2.7

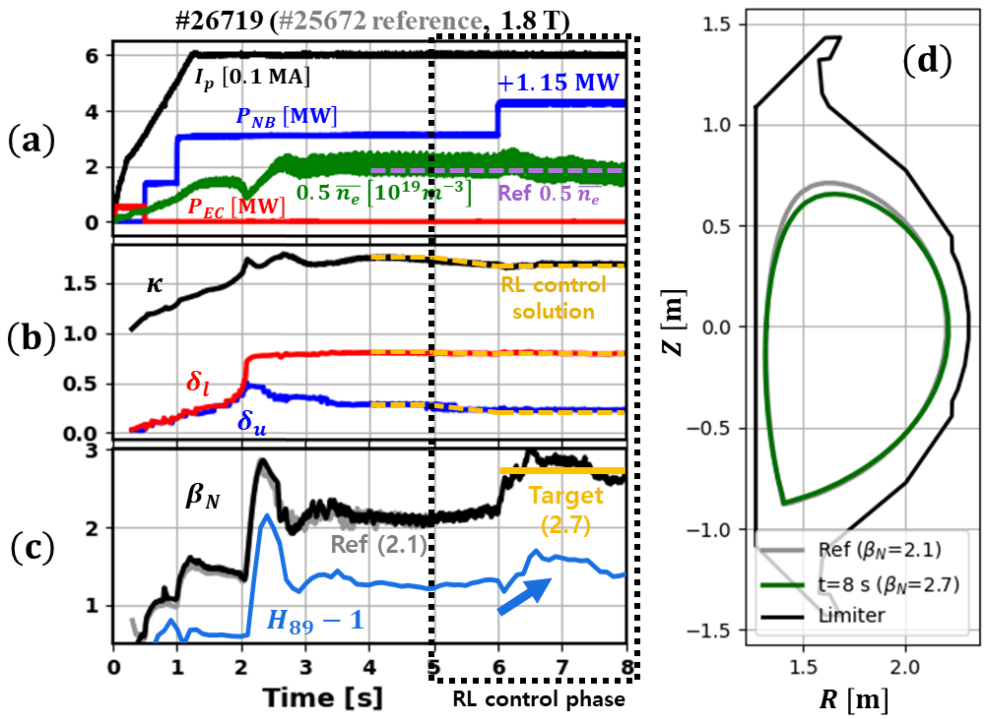


Figure 4.2 The KSTAR discharge (#26719) conducted with the RL-determined operation trajectory for Case 2.

Figure 4.2 is for the KSTAR discharge, #26719, with the RL-determined operation trajectory for Case 2 from the same reference shot as Case 1. We allowed an additional control knob of the 3rd NBI power to achieve a higher target ($\beta_N = 2.7$), while only two beams are used in the reference shot, #25672. The available power of the 3rd NBI was limited to ≤1.15 MW at that experiment. The RL agent determines 3rd NBI power of 1.15 MW and slightly lower plasma elongation and upper triangularity than the reference shot. Figure 4.2

(a–c) shows the actuator controls, the shape control, and the consequent evolution of β_N, respectively. (d) shows the plasma boundary shape for the reference (grey) and the RL–determined one (green). Before t~5 s, the discharge controls and the resulting plasma performance follow the reference shot identically. Then, the plasma shape adjustment is exerted until t=6 s, and the 3rd NBI is injected. As the elongation and the upper triangularity are adjusted from t=5 s to t=6 s, β_N slightly increases about 0.1. After the 3rd NBI is injected, β_N increases up to ~2.7 at t=8 s, which is the target we initially set. The target of $\beta_N = 2.7$ requires quite higher confinement enhancement ($H_{89} \sim 2.5$) than the reference under the given operating condition. It implies that the plasma regime has entered a more favorable state for performance improvement at the 3rd NBI phase through the delicate shape design by the RL agent. In Figure 4.3, the increment of β_N and H_{89} in Cases 1 and 2 are compared with other KSTAR H–mode and hybrid scenarios presented in [10]. We can see that the RL–determined tokamak operations yield higher confinement enhancement than other typical H–mode discharges in KSTAR.

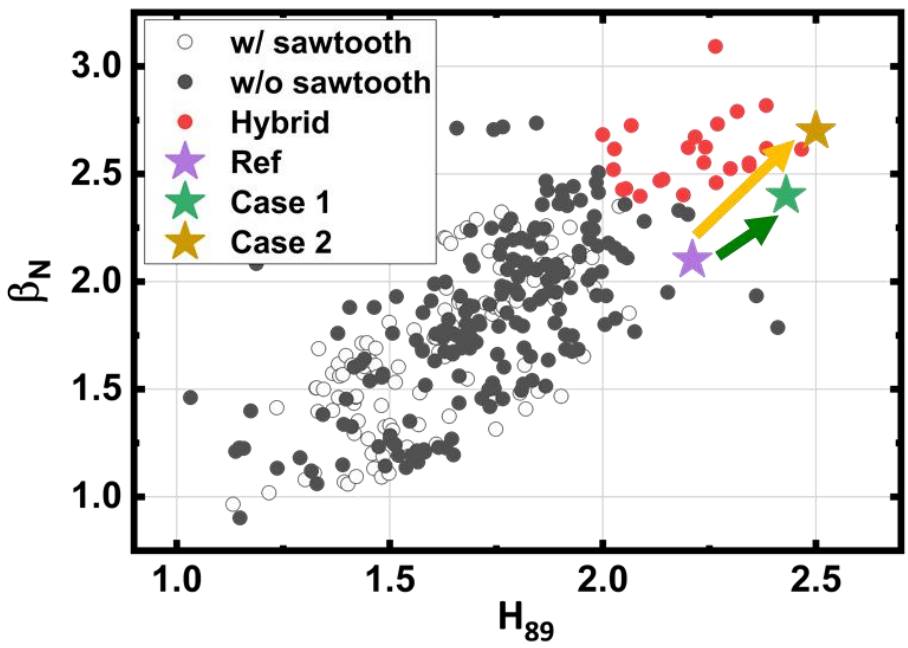


Figure 4.3 Comparison of the performance enhancement by the RL-designed operation in Cases 1 and 2 to other KSTAR discharges shown in [10].

4.1.3. Case 3: β_N control to 3.5

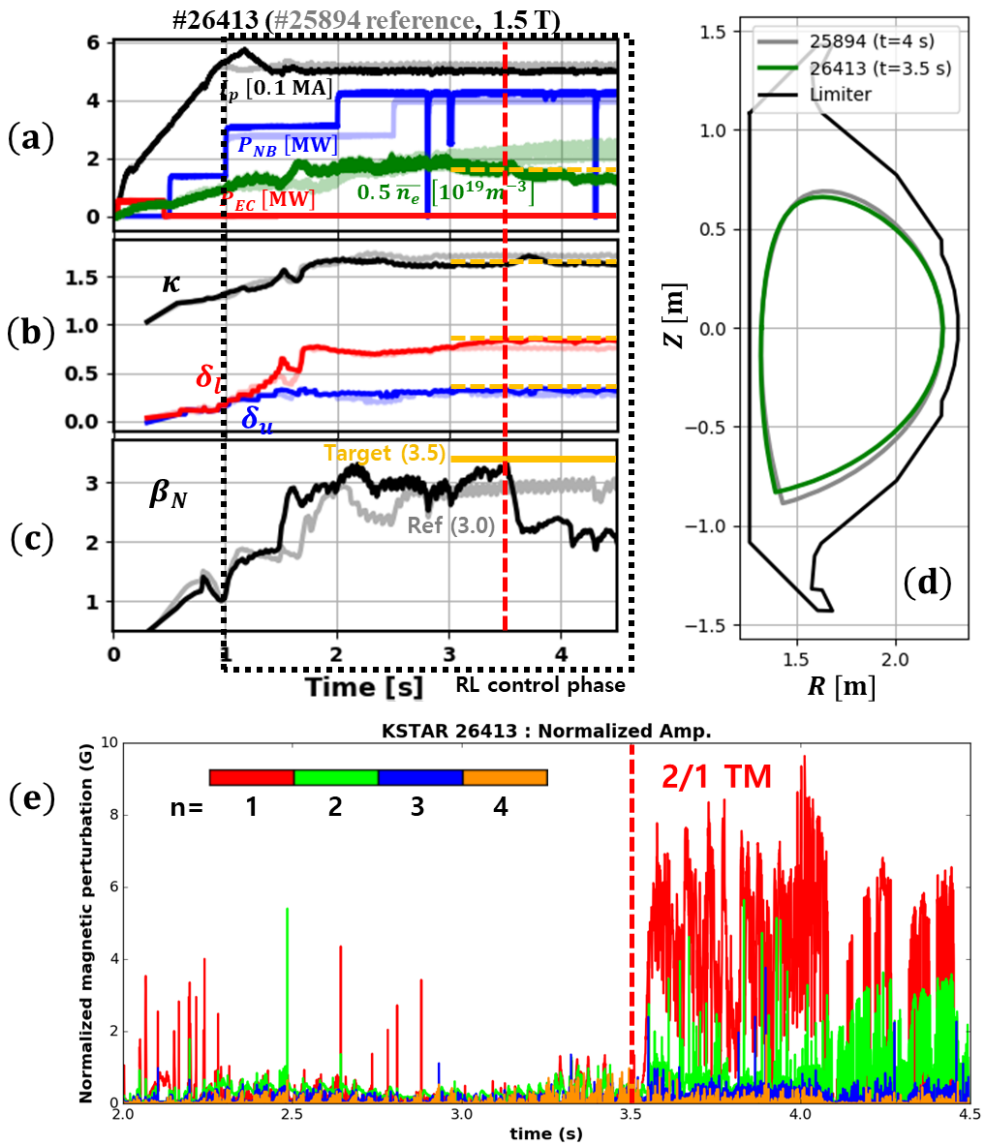


Figure 4.4 The KSTAR discharge (#26413) conducted with the RL-determined operation trajectory for Case 3.

Figure 4.4 is for the KSTAR discharge, #26413, with the RL-

determined operation trajectory for Case 3. In Cases 1 and 2, we constrained the discharge settings to the same as the reference shot except for the 3rd NBI and the plasma shape. In Case 3, however, we need to increase the degree of freedom of the control knobs since the target of $\beta_N = 3.5$ is much more challenging than Cases 1 and 2. We only constrained the ramp-up phase (t<1 s) of #26413 with the reference shot, #25894, then, the RL agent designed a new scenario with different $I_p, \overline{n_e}, P_{NB}$, and shape as shown in Figure 4.4 (a) and (b), based on the RL solutions for achieving $\beta_N = 3.5$ in (c). The RL agent determined the plasma current overshoot with 80 kA, which can be favorable for the core micro-instability stabilization [44], and the consequent β_N in the 2nd NBI phase has already been more improved than the reference shot, as shown in Figure 4.4 (c). Then, the 3rd NBI is injected at t=2 s, and the boundary shape control is completed at t=3.5 s. We achieve $\beta_N \gtrsim 3.0$ intermittently after the 3rd NBI, and the performance gradually increases as the shape control ends. However, before β_N reaches the target value, a significant 2/1 tearing mode, most likely to be a neoclassical tearing mode, occurred at t~3.5 s (vertical red dashed lines in Figure 4.4), resulting in the plasma performance suddenly dropping. The reduced plasma performance could not recover till the end of the discharge.

As β_N increases in t≲3.5 s, the plasma state approaches the marginal core MHD stability regime. However, the KSTAR simulator used for the RL training has an inherent limitation in predicting the MHD phenomena with a shorter time scale than 100 ms. In addition, the target, $\beta_N = 3.5$, is in the region outside three times the standard deviation from the mean in the dataset, which can also induce a high prediction uncertainty. Therefore, the RL-designed operation

trajectory might not properly predict and avoid the MHD instabilities.

4.2. Simultaneous control of multiple 0D parameters

4.2.1. Experiment with RL-designed trajectory

Even though the RL-based operation design showed a reliable performance in the simulation in Figure 3.5, we need to validate it in a real tokamak experiment. Real experiments have various uncertainties, such as the wall condition or the plasma shape control inaccuracy, which are not fully covered in the simulator. Therefore, the control performance in a real experiment might be lower than that in the simulation. In this work, the tolerance values in Equation (3.4) were chosen as the marginal error range to determine whether the control performance was reliable or not in the experiment.

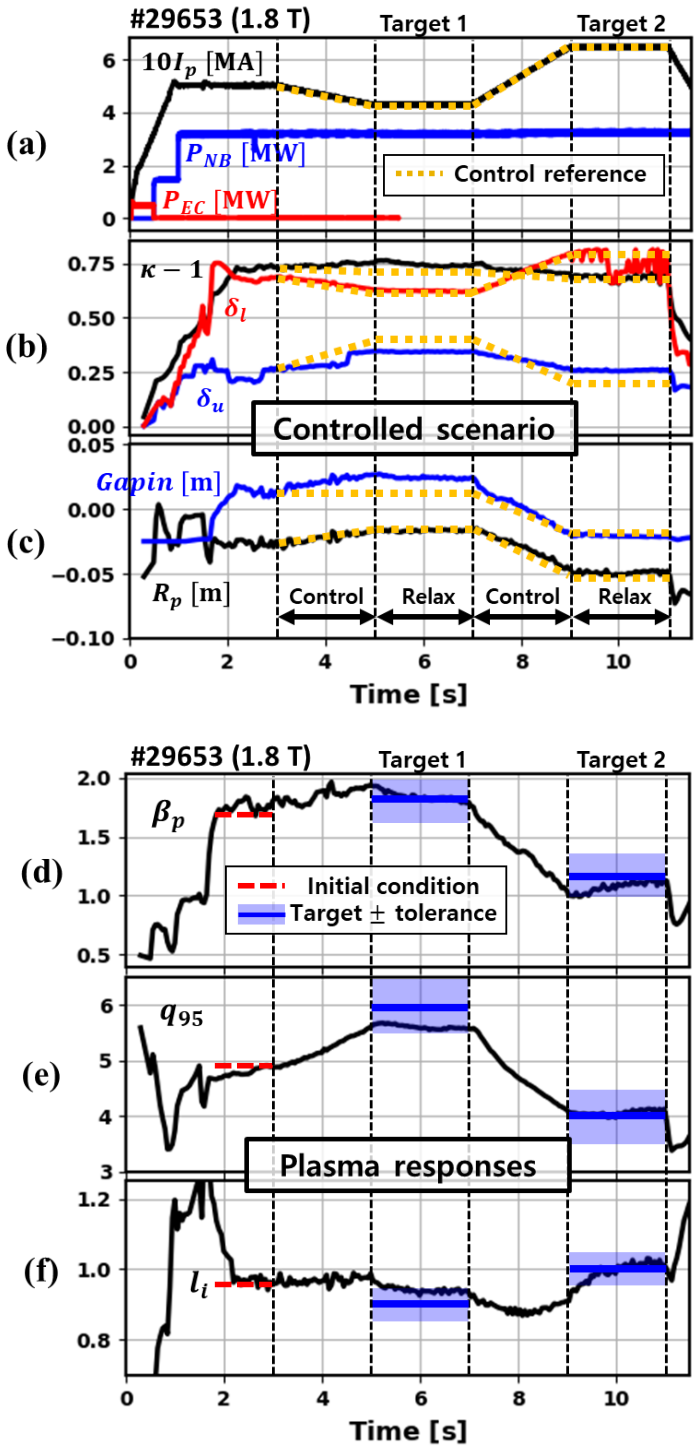


Figure 4.5 The KSTAR discharge (#29653) conducted with the RL-determined operation trajectory for multiple parameters' control.

Figure 4.5 shows the experimental demonstration of the RL-designed operation trajectory to achieve given targets of multiple 0D parameters, β_p, q_{95}, and l_i, in discharge #29653. Figure 4.5 (a-c) shows the external controls and the programmed references determined by the RL agent for two given targets, and (d-f) shows the resultant evolution of three 0D parameters and their target values. Here, the plasma boundary, β_p, q_{95}, and l_i, were obtained by the off-line magnetic EFIT [27]. In the experiment, the plasma state at t=3 s of the identical reference shot is fed into the initial condition for the RL operation design. The target values for three 0D parameters are set to be $(\beta_{p,target}, q_{95,target}, l_{i,target}) = (1.8, 6.0, 0.9)$ & $(1.2, 4.0, 1.0)$. The periods for control and relaxation phases are set to 2 s, the same as the modeling environment described in Figure 3.4. Then, the RL agent determines a control sequence of I_p and plasma shape for the two given targets, and the discharge setting is programmed according to this operation solution. We can see that the resultant tendency of the plasma evolution follows the given targets well as seen in Figure 4.5 (d-f). The agent determines to decrease (or increase) I_p to increase (or decrease) β_p under the fixed heating condition. At the same time, the plasma shape parameters are adjusted to match q_{95} and l_i with target values. Especially, even though the transient (at t$\gtrsim$7 s) and the long-term (at t$\gtrsim$10 s) response of l_i are different in the second target phase, the agent provides the control solution to match the delayed response with the target well.

However, there is a slight error between the target and actual response in the first target phase, larger than the control in the simulation in Figure 3.5. This error is because of the discrepancy

between the real experiment and the training environment modeling. First, the uncertainty due to wall conditions could induce different plasma behaviors from the simulation. Second, even though the RL agent designed an appropriate operation trajectory, the actual operation could not follow the programmed trajectory perfectly, as shown in Figure 4.5 (b–c), due to the inaccuracy of the coil control system. The second part can be improved if we adopt a more reliable feedback algorithm for coil control, such as a recently developed RL–based magnetic control [24]. A full framework that combines the operation design algorithm presented in this work and the RL–based magnetic control system will be able to perform the tokamak experiment by itself to reach the desired physical state we need for fusion reactor or physics exploration.

4.2.2. Comparison with other shots in dataset

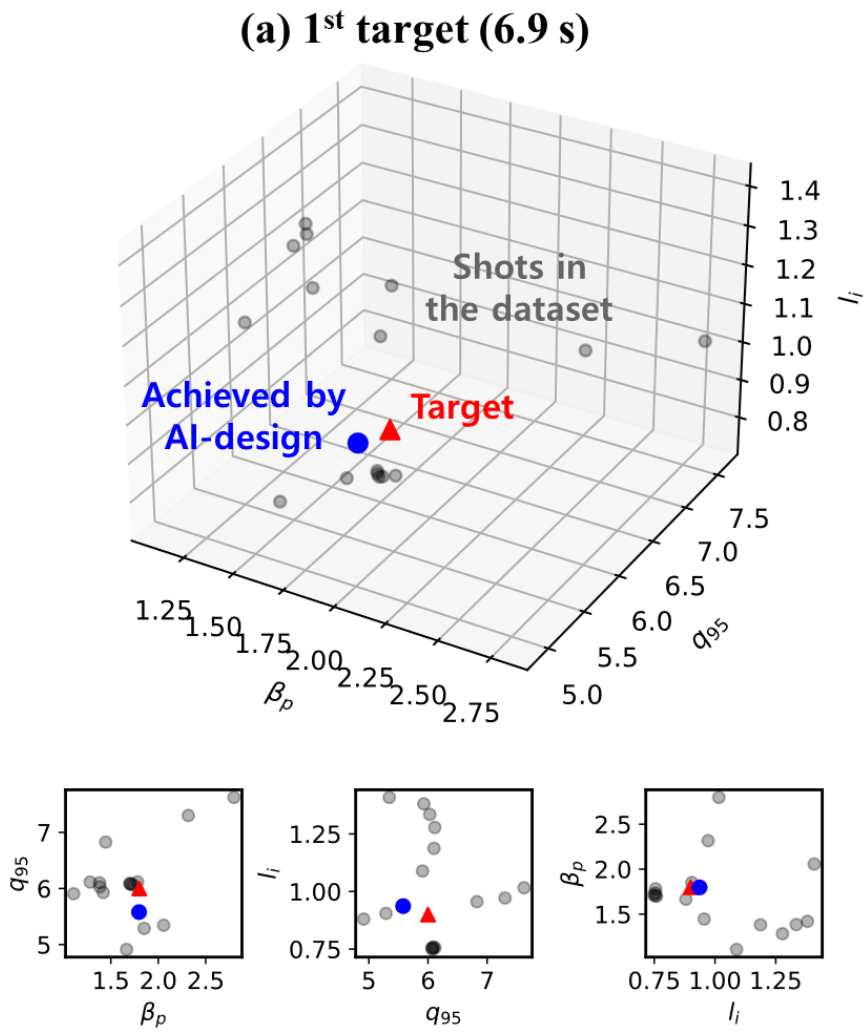


Figure 4.6 Comparison of the achieved plasma states at $t = 6.9\,s$ in #29653 with the shots under similar operating conditions in the dataset.

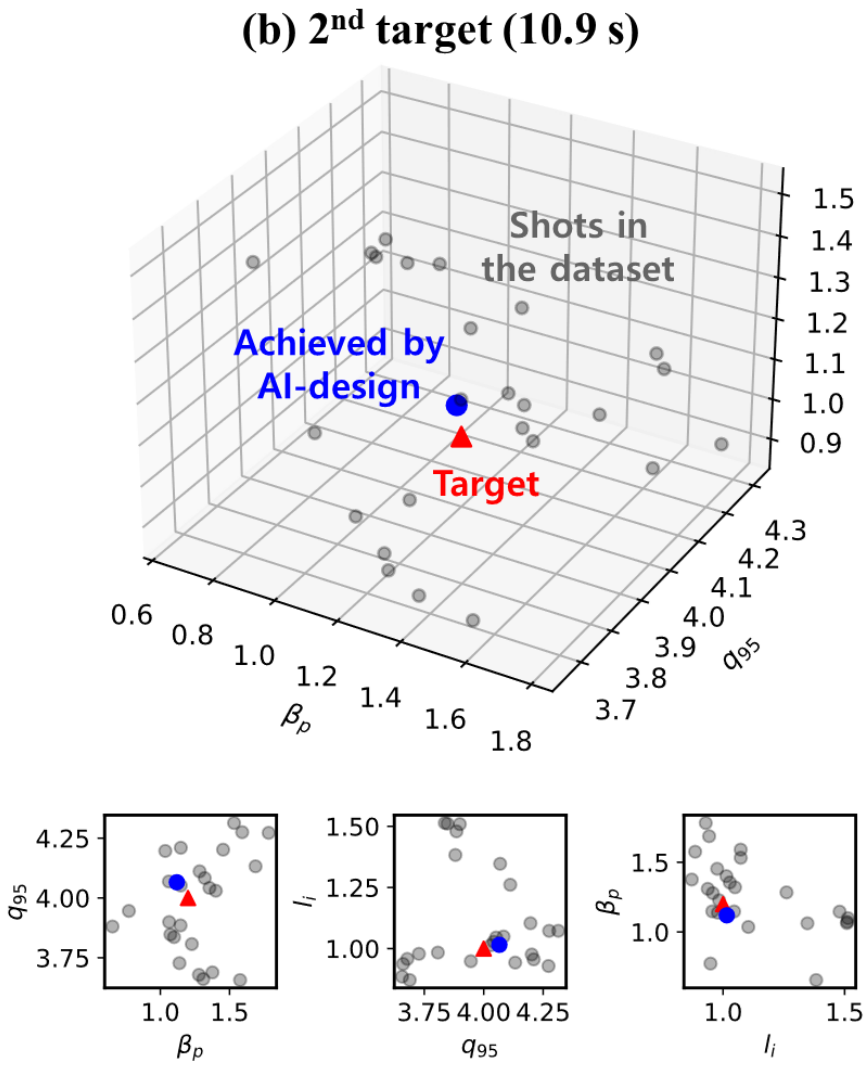


Figure 4.7 Comparison of the achieved plasma states at $t = 10.9\,s$ in #29653 with the shots under similar operating conditions in the dataset.

Figure 4.6 and Figure 4.7 show the distance between the target (red triangle) and the achieved plasma states (blue circle) in 3-dimensional parametric space for the first and second phases in #29653, respectively. The grey circles are the shots contained in

the original dataset used for the training of the data-driven simulator, whose operating conditions are similar to each phase in #29653. Those grey shots are selected to be under the condition within the range of $B_t \pm 0.05\,T, I_p \pm 0.05\,MA, P_{NB} \pm 0.1\,MW, \kappa \pm 0.05,$ and $\delta_{u,l} \pm 0.1$ from each phase. It shows that there are not the same states with #29653 in the original dataset, which means that the RL decision-making is not just a repetition of the previous discharge or overfitting of the dataset. It is noteworthy that the consequent plasma state can differ depending on the previous trajectory, even under similar operating conditions. Although there is a distance between the target and the achieved state by the RL agent, it provides a reasonable operation solution that makes the plasma state reach the target relatively closer than other shots.

Chapter 5

Conclusion

In this work, we developed an RL-based tokamak operation design algorithm that provides a reasonable operation trajectory to reach given target plasma states. The artificial agent was trained with the RL technique to control the physics parameters in the KSTAR tokamak. For the training environment of the RL agent, we first developed a data-driven simulator to provide an imitative plasma response in a tokamak. The data-driven model was trained with 5-year KSTAR experimental data and it showed a reasonable prediction of the plasma responses. Using this KSTAR simulator, we trained an RL agent to design the operation trajectory that yields the target plasma state. Not only the RL operation design could achieve the target of a single parameter, β_N, but also it could achieve the targets of multiple parameters, β_p, q_{95}, and l_i. We conducted tokamak experiments with the RL-designed operation trajectories in KSTAR, and the physics parameters were successfully controlled to the

targets. Although there were slight errors between the achieved state and the target, mainly due to the inaccuracy of the plasma shape controller, we could show that the RL-based design algorithm did not just repeat the original data but rather provided reasonable solutions for new conditions. We expect better control accuracy if combined with a more reliable algorithm for plasma shape control [24]. This RL operation design algorithm will be able to replace the human task of searching for an appropriate operation setting for given physical objectives and only requires a single fast calculation for each target. It will be able to guide a new advanced scenario development and also provide an operating condition to maintain a specific plasma state required for sophisticated physics studies.

In order to develop a more advanced algorithm capable of real-time plasma control in the future, the control error and the uncertainty of the diagnostics caused by the real-time EFIT [45] can be reflected in the training environment for RL. Adding uncertainties during training will make the AI more flexible and stable against the errors occurring in the real-time feedback loops. The ML-based plasma control system would be able to become a basis for the autonomously operating fusion reactor in the future.

Bibliography

[1] The way ahead for fusion, Nat. Phys. **16** 9 (2020) 889.

[2] BARBARINO, M., A brief history of nuclear fusion, Nat. Phys. **16** 9 (2020) 890.

[3] REICHERT, S., Inside ITER, Nat. Phys. **16** 9 (2020) 895.

[4] LEE, G.S. et al., Design and construction of the KSTAR tokamak, Nucl. Fusion **41** 10 (2001) 1515.

[5] LI, S., JIANG, H., REN, Z., XU, C., Optimal tracking for a divergent-type parabolic PDE system in current profile control, Abstr. Appl. Anal. **2014** (2014) 1.

[6] WAGNER, F. et al., Regime of Improved Confinement and High Beta in Neutral-Beam-Heated Divertor Discharges of the ASDEX Tokamak, Phys. Rev. Lett. **49** 19 (1982) 1408.

[7] KAMADA, Y., JT-60 TEAM, Extended JT-60U plasma regimes for high integrated performance, Nucl. Fusion **41** 10 (2001) 1311.

[8] GAROFALO, A.M. et al., Compatibility of internal transport

barrier with steady-state operation in the high bootstrap fraction regime on DIII-D, Nucl. Fusion **55** 12 (2015) 123025.

[9] FERRON, J.R. et al., High internal inductance for steady-state operation in ITER and a reactor, Nucl. Fusion **55** 7 (2015) 073030.

[10] NA, Y.-S. et al., On hybrid Scenarios in KSTAR, Nucl. Fusion **60** (2020) 086006.

[11] LUCE, T.C. et al., Development of advanced inductive scenarios for ITER, Nucl. Fusion **54** (2014) 013015.

[12] KATES-HARBECK, J., SVYATKOVSKIY, A., TANG, W., Predicting disruptive instabilities in controlled fusion plasmas through deep learning, Nature **568** (2019) 526.

[13] VEGA, J. et al., Disruption prediction with artificial intelligence techniques in tokamak plasmas, Nat. Phys. **18** 7 (2022) 741.

[14] BOURDELLE, C. et al., A new gyrokinetic quasilinear transport model applied to particle transport in tokamak plasmas, Phys. Plasmas **14** 11 (2007) 112501.

[15] STAEBLER, G.M., KINSEY, J.E., WALTZ, R.E., Gyro-Landau fluid equations for trapped and passing particles, Phys. Plasmas **12** (2005) 102508.

[16] SNYDER, P.B., GROEBNER, R.J., LEONARD, A.W., OSBORNE, T.H., WILSON, H.R., Development and validation of a

predictive model for the pedestal height, Phys. Plasmas **16** (2009) 056118.

[17] CITRIN, J. et al., Real-time capable first principle based modelling of tokamak turbulent transport, Nucl. Fusion **55** 9 (2015) 092001.

[18] MENEGHINI, O. et al., Self-consistent core-pedestal transport simulations with neural network accelerated models, Nucl. Fusion **57** (2017) 086034.

[19] SHIN, G.W., JUHN, J.W., KWON, G.I., SON, S.H., HAHN, S.H., Automatic detection of L-H transition in KSTAR by support vector machine, Fusion Eng. Des. **129** (2018) 341.

[20] SHIN, G., JUHN, J.W., KWON, G.I., HAHN, S.H., Real-time classification of L-H transition and ELM in KSTAR, Fusion Eng. Des. **157** (2020) 111634.

[21] HAN, H. et al., Demonstration of real-time control for poloidal beta in KSTAR, Fusion Eng. Des. **95** (2015) 44.

[22] KIM, K. et al., Time-dependent simulations of feedback stabilization of neoclassical tearing modes in KSTAR plasmas, Fusion Eng. Des. **89** 6 (2014) 859.

[23] KIM, H.S. et al., Feasibility experiment of physics-based global electron temperature profile control in KSTAR, Fusion Eng. Des. **135** (2018) 1.

[24] DEGRAVE, J. et al., Magnetic control of tokamak plasmas through deep reinforcement learning, Nature **602** 7897

(2022) 414.

[25] MNIH, V. et al., Human-level control through deep reinforcement learning, Nature **518** (2015) 529.

[26] HUYSMANS, G.T.A., PAMELA, S., VAN DER PLAS, E., RAMET, P., Nonlinear MHD simulations of edge localized modes (ELMs), Plasma Phys. Control. Fusion **51** 12 (2009) 124012.

[27] LAO, L.L., JOHN, H.S., STAMBAUGH, R.D., KELLMAN, A.G., PFEIFFER, W., Reconstruction of current profile parameters and plasma shapes in tokamaks, Nucl. Fusion **25** (1985) 1611.

[28] KIM, S.K., KEFIT PACKAGE, GitHub repository, https://github.com/ksk911211/KEFIT_PACKAGE.

[29] LEE, C.Y. et al., Development of integrated suite of codes and its validation on KSTAR, Nucl. Fusion **61** 9 (2021) 096020.

[30] STAEBLER, G.M., KINSEY, J.E., WALTZ, R.E., A theory-based transport model with comprehensive physics, Phys. Plasmas **14** 5 (2007) 055909.

[31] PANKIN, A., MCCUNE, D., ANDRE, R., BATEMAN, G., KRITZ, A., The tokamak Monte Carlo fast ion module NUBEAM in the national transport code collaboration library, Comput. Phys. Commun. **159** (2004) 157.

[32] FELICI, F. et al., Real-time-capable prediction of temperature and density profiles in a tokamak using RAPTOR and a first-principle-based transport model, Nucl. Fusion **58**

(2018) 096006.

[33] HOCHREITER, S., SCHMIDHUBER, J., Long Short-Term Memory, Neural Comput. **9** (1997) 1735.

[34] BUTTERY, R.J. et al., On the form of NTM onset scalings, Nucl. Fusion **44** 5 (2004) 678.

[35] STILLERMAN, J.A., FREDIAN, T.W., KLARE, K.A., MANDUCHI, G., MDSplus data acquisition system, Rev. Sci. Instrum. **68** 1 (1998) 939.

[36] WAN, C., YU, Z., WANG, F., LIU, X., LI, J., Experiment data-driven modeling of tokamak discharge in EAST, Nucl. Fusion **61** 6 (2021) 066015.

[37] SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I., SALAKHUTDINOV, R., Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. **15** (2014) 1929.

[38] CORTES, C., MOHRI, M., ROSTAMIZADEH, A., L2 Regularization for Learning Kernels, Proc. 25th Conf. Uncertain. Artif. Intell. (2012) 109.

[39] BROCKMAN, G. et al., OpenAI Gym, ArXiv Prepr. ArXiv1606.01540 (2016).

[40] FUJIMOTO, S., VAN HOOF, H., MEGER, D., Addressing Function Approximation Error in Actor-Critic Methods, 35th International Conference on Machine Learning, ICML 2018, (2018) 1–15.

[41] HILL, A. et al., Stable Baselines, GitHub repository, https://github.com/hill-a/stable-baselines.

[42] LILLICRAP, T.P. et al., Continuous Control with Deep Reinforcement Learning, 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings, (2016) 1–14.

[43] TROYON, F., GRUBER, R., SAURENMANN, H., SEMENZATO, S., SUCCI, S., MHD-Limits to Plasma Confinement, Plasma Phys. Control. Fusion **26** 1A (1984) 209.

[44] HOBIRK, J. et al., Improved confinement in JET hybrid discharges, Plasma Phys. Control. Fusion **54** (2012) 095001.

[45] FERRON, J.R. et al., Real time equilibrium reconstruction for tokamak discharge control, Nucl. Fusion **38** (1998) 1055.

Abstract in Korean

토카막에서 정밀한 물리 실험을 하기 위해서는 먼저 특정한 내부 플라즈마 상태를 달성하고 유지하는 것이 필요하다. 특히 상용 핵융합로 운전을 위해서는 자기유체역학적으로 안정적인 플라즈마 영역 내에서의 제어가 필수적이며, 고출력의 핵융합 반응을 일으킬 수 있는 플라즈마 상태를 유지할 수 있어야 한다. 기존에는 실험에서 목표로 하는 플라즈마 상태를 달성하기 위해, 다양한 토카막 운전 조건에서의 사전 시뮬레이션과 실험에서의 추가적인 시행착오가 필요하였다. 이 경우 많은 인적 노동력과 시간이 소요되었으며, 새로운 목표 상태들에 대해 매번 동일한 수준의 시행착오가 요구된다는 문제가 있다.

본 논문은 목표 플라즈마 상태를 달성하기 위한 토카막의 운전 경로를 설계하는 기계학습 기반의 알고리즘 개발을 다룬다. 해당 알고리즘은 기존의 상당한 시뮬레이션 및 시행착오를 수행하는 작업들을 대체할 수 있으며, 이를 통해 보다 빠르고 효율적으로 가능성 높은 운전 조건을 도출할 수 있다. 첫 번째로, 토카막 운전 설계 인공지능 모델의 훈련 환경에 해당하는 토카막 시뮬레이션 기술이 연구되었다. KSTAR 실험 데이터의 패턴을 학습하여 시간에 따른 플라즈마 상태를 순차적으로 예측하는 LSTM 기반의 인공신경망 모델을 개발하였다. 데이터 학습 과정에서 과적합 및 오차 누적 등의 문제를 해결하기 위해 다양한 수치적 기법들이 적용되었다. 학습된 모델은 KSTAR의 다양한 운전 시나리오 방전들에 대해 우수한 예측 정확도를 보여주었으며, 신뢰도 분석을 통해 모델이 과적합되지 않음을 확인하였다. 또한 해당 모델을 기반으로, 실시간 상호작용을 통한 가상 토카막 실험이 가능하도록 그래픽 사용자 인터페이스 (GUI)를 개발하였다. 해당 GUI 상에서 사용자가 토카막 운전 변수들을 조정함에 따라 플라즈마의 변화를 실시간으로 시각적으로 확인할 수 있기 때문에 물리 연구 뿐 아니라 전문가 교육용으로서의 의의가 있

다.

두 번째로, 개발된 시뮬레이터 상에서 스스로 운전 변수들을 조정하여 목표로 하는 플라즈마 상태를 달성하는 인공지능 모델을 강화학습 기법을 이용하여 훈련하였다. 이를 통해 목표 플라즈마 상태를 달성하기 위한 적절한 토카막 운전 경로를 설계하는 알고리즘을 개발할 수 있다. 먼저 목표 β_N 달성을 위해 플라즈마 전류, 플라즈마 형태 및 가열 파워를 결정하는 모델을 훈련하였다. 훈련된 모델이 설계한 운전 경로를 이용하여 실험을 수행해본 결과 오차범위 내에서 목표 β_N이 도출됨을 검증하였다. 특히 한정된 가열 조건에서 높은 성능을 달성하기 위해 플라즈마 형태를 적절히 조정하여 가둠 성능을 향상시키는 것을 확인하였다. 이후 보다 더 구체적인 플라즈마 상태를 달성하기 위해, 플라즈마 압력 (β_p) 뿐 아니라 자기장 구조 (q_{95}) 및 내부 인덕턴스 (l_i)의 다중 파라미터들의 목표값을 동시에 달성케 하는 인공지능 모델 또한 훈련하였다. 해당 모델이 설계한 운전 경로를 실제 실험에 적용해본 결과, 다중 플라즈마 파라미터들이 성공적으로 목표값으로 제어됨을 확인하였다.

본 논문에서 개발된 기계학습 기반 알고리즘은 추후 고성능 운전 시나리오 연구에 도움을 줄 수 있으며, 정밀한 물리 조건을 요구하는 실험에서 초기 조건 달성을 위한 기술로 적용될 수 있을 것으로 기대된다. 추후 실시간 피드백 제어에 적용됨으로써 다양한 상황에서 자율적으로 제어되는 핵융합로 기술 개발을 위한 초석이 될 수 있을 것으로 전망한다.

주요어: 토카막, 플라즈마, KSTAR, 기계학습, 강화학습, 토카막 시뮬레이션, 플라즈마 제어

학번: 2017-29826