CONFERENCE PRE-PRINT

REINFORCEMENT LEARNING-BASED PLASMA SHAPE CONTRO VIA LSOFLUX SCHEME ON SUPERCONDUCTOR TOKAMAK

Haoyu Wang
University of Science and Technology of China
Hefei, China
Email: wangyhplasma@ipp.ac.cn

Yuehang Wang*
Institute of plasma physics, Chinese Academy of Sciences
Hefei, China

Abstract

Plasma shape control is a critical challenge in magnetic confinement fusion devices, where precise regulation of the magnetic flux distribution is essential to achieve stable plasma configurations. Traditional control strategies often rely on linear approximation and decoupling solution based on the physical model, which multiple linear approximations are required to achieve this step by step. Now reinforcement learning methods show great potential in solving highly complex, multidimensional coupled problems. This work proposes a reinforcement learning-based framework to optimize plasma shape control on superconductor tokamak through dynamic magnetic flux regulation. By formulating the control problem as a Markov decision process, the RL agent learns to coordinate the poloidal field coils power supply to simultaneously stabilize the plasma boundary and minimize flux deviations between boundary and X point. The observation of the RL agent has incorporated historical temporal information to adapt to the complex dynamic response caused by the double-layer vacuum chamber in the fully superconducting tokamak. A new reward design method is proposed to meet the requirements of ISOFLUX algorithm and the voltage limited characteristics of superconducting tokamak. Numerical simulations and experimental validations demonstrate that the RL driven controller achieves improvement in shape tracking accuracy compared to conventional proportional-integral-derivative methods. Furthermore, the system exhibits robust performance against magnetic perturbations, maintaining the plasma boundary within 10^{-3} Wb and $2 * 10^{-4}$ T of the target equilibrium. This work highlights the potential of data-driven reinforcement learning in bridging the gap between magnetic flux physics and high precision shape control for nextgeneration fusion reactors.

1. INTRODUCTION

Tokamak is a toroidal magnetic confinement fusion device that utilize strong magnetic field to confine a high-temperature plasma within a toroidal vacuum chamber, This enables deuterium-tritium fuel to undergo nuclear fusion reactions at temperatures reaching several hundred million degrees Celsius, resulting in the release of substantial amounts of energv. Due to its environmental sustainability and high energy conversion efficiency, the tokamak is widely considered one of the most promising approaches toward achieving a long-term solution to the global energy crisis [1]. Experimenta Advanced Superconducting Tokamak (EAST) is a fully superconducting tokamak experimental device. Due to the fact that its coil is based on a superconducting design, is far from the plasma, and the coil functions overlap, it leads to a strong coupling among current-driven, position-driven and shape-driven. This poses a huge challenge to the control system [2].

Plasma shape control in EAST is effected by modulating the currents in external coils so as to regulate the plasma's position and geometry inside the vacuum vessel. Originally, only global parameters-elongation, triangularity, and similar macroscopic metrics—were targeted, simply to maximise use of the available vessel volume [3]. The subsequent introduction of auxiliary heating schemes and strike-point management, however, has shifted the control objective from these global parameters to finer, localised quantities such as the boundary magnetic flux or the gap between the last-closed flux surface and the wall [4]. At present the experiment employs the ISOFLUX [5] algorithm, which enforces equality between the flux at a set of pre-selected control points and that at the X-point while simultaneously fixing the X-point location, thereby achieving sub-centimetre accuracy in shape reconstruction. By suppressing—or at least retarding—a variety of disruption-precursor instabilities, this high-precision control appreciably broadens the operational window for stable, high-performance discharges [6].

Conventional control methods often rely on sequentially linearizing and manually decoupling the underlying physical dynamics. This typically involves chaining together multiple piecewise linear approximations in a multi-step process. In comparison, Reinforcement Learning (RL) has matured into a powerful and versatile framework for decision-making in complex settings. Not only has RL achieved superhuman performance in specific games [7], but modern RL—supported by deep neural networks and reward modeling—has also produced measurable gains in real-world, high-dimensional applications. Like sim-to-real robotic manipulation [8], portfolio management optimization [9], and coordinated traffic signal control within urban power networks [10]. Moreover, RL has become essential for aligning large language models after training, substantially reducing hallucinations in systems like GPT and improving sample efficiency by orders of magnitude [11]. As a result, RL is transitioning from an experimental set of algorithms into an interpretable and verifiable universal decision-making architecture—able to replace or enhance traditional control and optimization methods in highly complex, dynamic systems.

Magnetic confinement of plasmas exemplifies this trend. RL has been successfully integrated into various plasma control scenarios, including shape control on TCV [12], tearing-mode stabilization on KSTAR [13] [14], betap control on EAST [15], and vertical displacement stabilization for ITER [16]. By directly encoding control goals into a scalar reward function, RL shifts the focus from 'how to implement control' to 'what objectives should be pursued'. Effective control policies are learned through model-free interaction, replacing intricate cascades of hand-tuned classical controllers with a unified, end-to-end approach that greatly simplifies the design process.

In this study, we first train the agent through extensive interaction with a high-fidelity tokamak simulator [17].The learned control policy is then deployed directly into the EAST plasma control system (PCS) [18] and evaluated through real-world experiments. As a fully data-driven approach, the agent is able to manage entire discharge sequences without pre-programmed gain schedules. Even when the simulation does not fully match real conditions, a carefully designed reward function effectively narrows the sim-to-real gap during operation. With continued progress in artificial intelligence, RL stands out as a highly capable framework for dealing strongly coupled, multi-input multi-output (MIMO) control challenges. Its growing adoption in fusion control systems points toward a future where RL becomes a standard—rather than experimental—element of plasma control architecture.

2. BACKGROUND

2.1. Reinforcement Learning

Reinforcement learning is an important branch of machine learning. Its core idea lies in that the Agent continuously interacts with the Environment and learns the optimal decision-making strategy based on the obtained reward signals to maximize the long-term cumulative return. This framework provides a strong theoretical basis for solving sequential decision-making problems. In the standard reinforcement learning setting, the interaction Process between the agent and the environment can be formally described through the Markov Decision Process (MDP). An MDP can be represented by A five-tuple (S, A, P, R, γ), where:

— S represents the set of all possible states of the environment;
— A represents the set of actions that the agent can perform;
— $P(s'|s, a)$ represents the probability of state transition, describing the probability of transitioning to state s' after performing action a in state s;
— $R(s, a, s')$ is the reward function, which is used to evaluate the quality of state-action pairs;
— $\gamma \in [0, 1]$ is the discount factor, which is used to balance immediate rewards and long-term returns.

The objective of the agent is to learn a strategy $\pi(a|s)$, which defines the probability distribution of choosing action a in state s to maximize the expected discounted cumulative reward obtained from the initial state:

$$J(\pi) = E_{\pi}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})] \tag{1}$$

However, in actual control systems, agents often cannot directly obtain the complete state information of the environment but can only acquire partial observations through sensors. For this reason, the Partially Observable MDP (POMDP) provides a more applicable modeling framework. In POMDP, agents maintain an internal confidence state of the environment based on current observations and historical interaction information, and make decisions on this basis. Unlike traditional control methods, RL does not require prior knowledge of the precise mathematical model of the system, but rather discovers the optimal behavioral strategy through a trial-and-error mechanism. At each time step, the agent selects an action based on the current strategy and executes it, and the environment returns a reward signal and new observations. The design of the reward function $R(s, a, s')$ has a crucial impact on the training effect: it not only needs to accurately reflect the ultimate goal of the task but also provide sufficient learning signals during the exploration process. Overly sparse rewards can make it difficult for agents to learn, while poorly designed reward functions may cause agents to learn behaviors that do not match expectations. However, this design will allow researchers to shift their focus from model decoupling to more direct task requirements.

2.1.1. Proximal Policy Optimization

In the context of sequential decision-making problems based on MDP, the goal of an agent is to learn an optimal policy, i.e., a mapping from states to actions—through interaction with the environment. Proximal Policy Optimization (PPO) [19] is a widely adopted and efficient policy search algorithm. Its core design objective is to ensure stability during learning, namely, to avoid drastic performance oscillations and collapses during policy updates, which is crucial for real-world applications such as robotic control.

PPO achieves this via a clever clipping mechanism that directly constrains the magnitude of policy updates between consecutive iterations. The algorithm learns by optimizing a surrogate objective function, whose central component is given by:

$$L^{CLIP}(\theta) = \hat{E}_t[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)] \tag{2}$$

where, $r_t(\theta)$ denotes the probability ratio of selecting the same action under the new and old policies, and $\hat{A}_t$ is the advantage function, which estimates how much better or worse a specific action is compared to the average in a given state. The clip operation restricts the policy update step to the interval $[1-\varepsilon, 1+\varepsilon]$. This mechanism ensures that PPO updates the policy robustly toward performance improvement, while automatically avoiding excessively large updates that could degrade performance.

To accurately estimate $\hat{A}_t$, this study employs the Generalized Advantage Estimation (GAE) method [20]. The core idea of GAE is to balance the bias and variance in the estimation. By introducing a decay parameter λ, it combines the prediction errors across different time steps to produce an advantage estimate that is both relatively accurate and sufficiently stable:

$$\hat{A}_t^{GAE} = \sum_{t=0}^{\infty}(\gamma\lambda)^l \delta_{t+l}^V \tag{3}$$

where δ_t represents the temporal difference error. By adjusting λ, a smooth transition can be made between high bias/low variance (as $\lambda \to 0$) and low bias/high variance (as $\lambda \to 1$). GAE provides PPO with high-quality, low-noise policy evaluation signals, which is one of the key factors behind its high performance.

The PPO algorithm ensures training stability through its clipping mechanism, while the GAE technique contributes accuracy via its effective bias–variance trade-off. The combination of these two components results in a framework that achieves an exceptional balance among performance, stability, and implementation complexity, making it a highly effective tool for solving complex continuous control tasks.

2.2. Shape Control On EAST

Shape control is achieved using poloidal field (PF) coils located outside the vacuum vessel. The control strategy varies depending on the discharge phase of the plasma:

— During the current ramp-up phase, where the plasma shape changes rapidly, the shape control system regulates only the position of the plasma current centroid (estimated via the E-matrix). The plasma evolves toward the target configuration through feedforward current control.
— In the flat-top phase, to enhance discharge performance, precise shape control is applied. This involves obtaining the magnetic flux values at specified control points on the plasma boundary—along with either the magnetic field at the X-point or its actual coordinates—using real-time equilibrium reconstruction codes such as rtEFIT or PEFIT. The plasma control system (PCS) then computes the required PF coil currents based on these values. Subsequently, the appropriate voltage across each PF coil is determined according to the present coil current, and voltage commands are issued to the power supplies to execute control, i.e. ISOFLUX algorithm [2][21][22][23].

This study focuses on voltage-mode control during the flat-top phase. For shape control, the plasma is generally considered to be in a quasi-steady state, and the response time of the control loop is significantly longer than the characteristic time of passive structures. Under this assumption, the effect of eddy currents induced in passive structures on shape control can be neglected. The response model thus primarily accounts for variations in coil currents and the intrinsic plasma response. Under fixed equilibrium conditions, the plasma response model is constructed by evaluating how differences in magnetic flux at control points and at the X-point respond to changes in PF coil currents.

The control flowchart is shown in Fig. 1. In this scheme, the RL model replaces the PID module illustrated, with fast-control coils decoupled and controlled separately. The agent participates in feedback control without requiring current feedforward. It outputs PF voltage commands at 1kHz and determines the next command based on real-time error values, thereby closing the control loop.

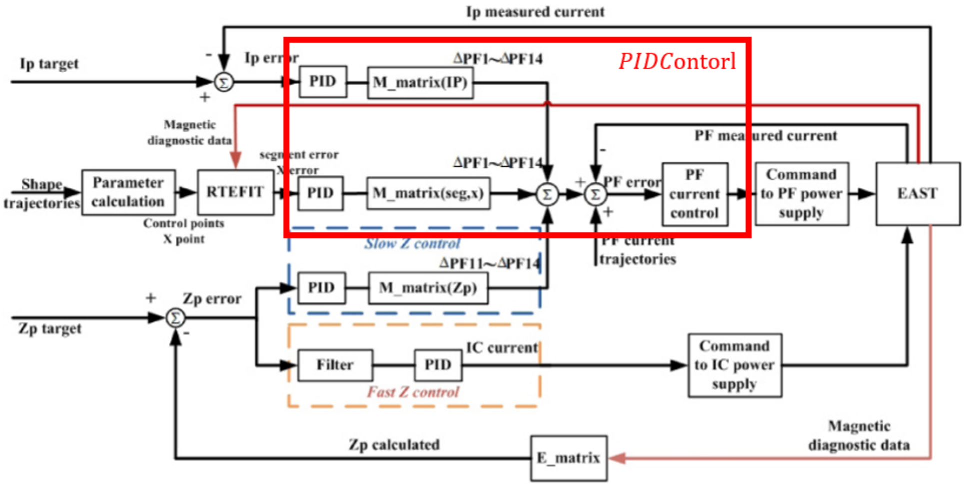


FIG. 1. EAST ISOFLUX shape control algorithm flow

3. TRAINING AND PERFORMANCE VALIDATION IN SIMULATION

3.1. Basic elements of RL

In this study, we apply RL to control the plasma current (Ip), the magnetic field strength at the X-point in both the R and Z directions, and the flux differences between the controlled points ($1 \sim 6$ and $8 \sim 9$, excluding strike point 7) and the X-point. On the EAST, the poloidal field (PF) system is unable to effectively respond to plasma dynamics with characteristic timescales shorter than $20ms$, due to limitations including the voltage constraints of superconducting PF coils, the finite response time of power supplies, and the shielding effect of the vacuum vessel against externally applied poloidal fields. As a result, the full penetration of external magnetic fields into the plasma—reaching saturation—requires more than $20ms$ [17]. Consequently, the current and magnetic field profiles within the plasma are influenced not only by the instantaneous state of external coils and heating systems, but also notably by their historical states.

The observed non-Markovian behavior of the system implies that a controller relying solely on instantaneous observations may fail to accurately infer the true state of the plasma, potentially leading to degraded control performance or even the excitation of plasma instabilities. To effectively incorporate historical information for the RL agent and ensure discharge integrity and stability on EAST, this study adopts a time-delayed embedded observation approach. Specifically, we construct an augmented observation vector by concatenating the current observation with those from the previous k time steps, which is then provided as input to the agent:

$$Observation = [O_{t-k}, O_{t-k+1}, ..., O_{t-1}, O_t], k = 20 \tag{4}$$

At each time step, the observation consists of the normalized values of all 11 controlled variables:

$$O_t = (\delta\psi_{1\sim6,8\sim9}, Br_X, Bz_X, Ip_{errror})_{scaled} \tag{5}$$

$$\delta\psi_i = \psi_i - \psi_X \tag{6}$$

$$Ip_{errror} = Ip - Ip_{target} \tag{7}$$

The scaling factors for the controlled variables are 0.01, 0.005 and 10000, respectively.This approach provides the agent with essential temporal context, enabling it to implicitly learn the dynamic evolution of the system state—particularly the time-delay effects associated with the penetration of poloidal magnetic fields—thereby facilitating more accurate and proactive control decisions.

However, providing historical observations alone is insufficient to guide the agent in mastering complex control tasks; the agent also requires a clear and precise objective to evaluate the quality of its actions. This objective is communicated through the reward function, which maps the agent's observations (along with their historical context) and actions to a scalar reward signal. The design of this function critically influences the behavior of the policy ultimately learned by the agent. Accordingly, the reward function designed in this work simultaneously incentivizes the reduction of tracking errors across multiple controlled variables. In addition, constraints on the action outputs are incorporated to account for practical power supply limitations, with penalties applied accordingly. The specific design is as follows:

— Tracking Error Reward: This component normalizes the tracking errors of the plasma current (Ip_{error}), the magnetic field at the X-point (Br_X, Bz_X), and the flux differences at each controlled point ($\delta\psi$). Different normalization strategies are applied segment-wise based on quantitatively defined reference thresholds.

$$norm(x) = \begin{cases} 1, & 0 \leq x \leq good \\ a \cdot e^{b \cdot x} + c, & good < x \leq stop \\ 0, & x > stop \end{cases} \tag{8}$$

The parameters a, b, c are determined by solving the system of equations using the $fsolve$ function from Python's scientific computing libraries [24]:

$$\begin{cases} a \cdot e^{b \cdot good} + c = y_{good} \\ a \cdot e^{b \cdot bad} + c = y_{bad} \\ a \cdot e^{b \cdot stop} + c = y_{stop} \end{cases} \tag{9}$$

$Good$ and bad represent the quality of the controlled variables, where (bad, y_{bad}) is used to adjust the gradient variation of the normalization curve. These parameters directly influence the convergence and difficulty of the training process. Although the parameter space contains multiple local optima, the parameter set employed in this study—while not proven to be globally optimal—was experimentally validated to effectively balance system sensitivity and stability.

The normalized parameter values are combined into a composite reward using a weighted $SmoothMax$ function. The core objective is to incentivize the agent to minimize the overall deviation of all control targets from their desired reference values. The mathematical formulation is as follows:

$$SmoothMax(x_{1 \ldots n}, w_{1 \ldots n}, \alpha) = \frac{\sum_{i=1}^{n} w_i x_i e^{\alpha x_i}}{\sum_{i=1}^{n} w_i e^{\alpha x_i}} \tag{10}$$

Here, the parameter α determines the direction and magnitude of the weighting. When $\alpha > 0$, the function output approaches the maximum value of its inputs; when $\alpha < 0$, it tends toward the minimum. To ensure overall stability in control applications, it is more appropriate to choose $\alpha < 0$.

— Action Smoothness Penalty: To enhance the stability of the control system and accommodate the physical response limits of the power supplies, a constraint on the rate of change of control actions is introduced. This term penalizes abrupt changes between consecutive actions output by the agent, thereby promoting smoother and more stable control policies while reducing stress on the actuators. The penalty is formulated as follows:

$$penalty = -\sqrt{\sum_{i=1}^{12} (a_{i_{t-1}} - a_{i_t})^2} \tag{11}$$

Therefore, at each time step t, the total reward obtained by the agent is given by:

$$f(x_1, ..., x_i) = SmoothMax(norm(x_1), ..., norm(x_i)) \tag{12}$$

$$r = \alpha * f(|\delta\psi_i|) + \beta * f(|Br_X|, |Bz_X|) + \gamma * f(|Ip_{error}|) + \delta * penalty, \alpha + \beta + \gamma = 1 \tag{13}$$

3.2. RL Training and Testing

We employs a parallelized training architecture to enhance learning efficiency. While PPO, as an on-policy algorithm, offers exceptional training stability, its requirement for real-time data collection under the current policy can lead to limited sample efficiency. To address this, we implemented a parallelized sampling framework by synchronously running multiple environment instances, significantly improving data throughput. Compared to off-policy methods that rely on experience replay, parallel PPO maintains policy evaluation consistency and avoids importance sampling bias, while effectively mitigating the sample efficiency limitations inherent in on-policy approaches. This architecture preserves the inherent stability of the original algorithm and achieves an order-of-magnitude improvement in data collection efficiency through parallelization. The approach trades off a degree of sample efficiency for exceptionally high training stability and reliability.

Based on an equilibrium configuration with Ip of $250kA$ and κ of 1.7, the training simulated 2 seconds of flat-top phase shape control. Random deviations $\Delta\beta_p$ were introduced as perturbations during training. The result of the training is showed in Fig. 2.

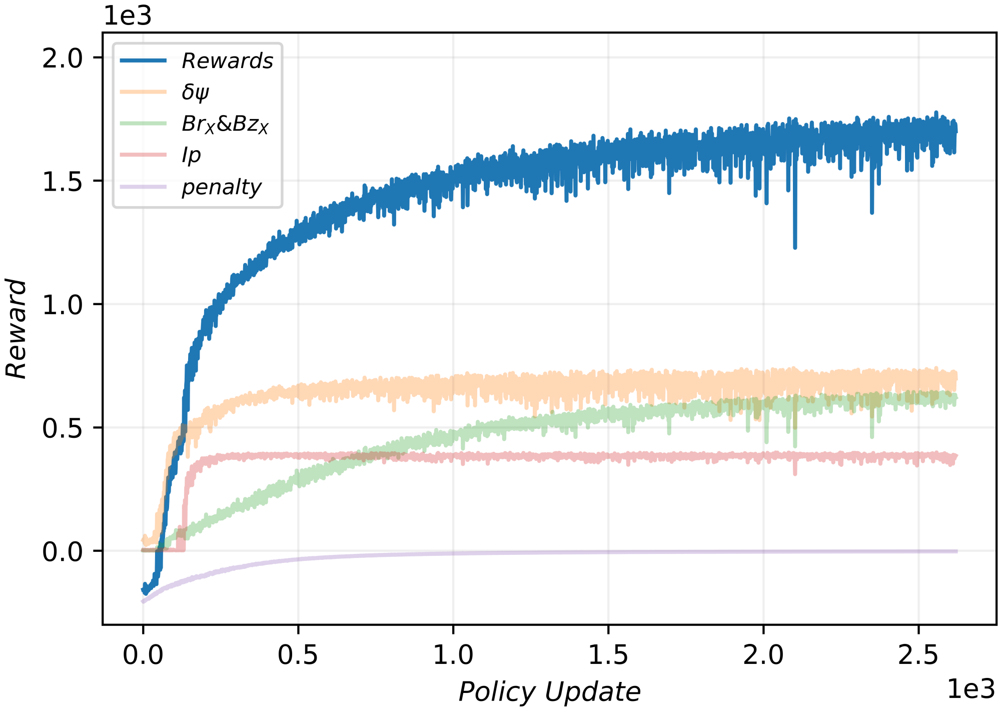


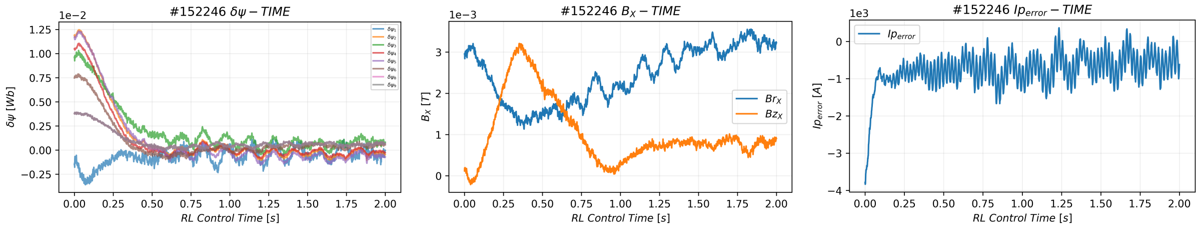
FIG. 2. Training progress. The maximum achievable reward is set to 2000, indicating that all controlled variables fall within the desired performance range (0, good). We employ an asymmetric Actor-Critic architecture, in which the Actor network comprises two hidden layers with 256 and 64 units, respectively, while the Critic consists of three hidden layers, each with 128 units. All layers use the tanh activation function.

4. EXPERIMENTAL RESULTS AND APPLICATION ON EAST

The policy model trained on PyTorch was converted into the ONNX format and integrated into the Plasma Control System (PCS) to enhance cross-platform deployment capability and operational efficiency [25]. During actual discharge experiments, the RL agent received real-time system observations, based on which it generated voltage control commands for the PF coils and achieved real-time control through the PCS.

A phased control strategy was adopted: the plasma was first sustained using a PID controller until $t = 3s$, after which control was switched to the RL agent, showd in Fig. 3.

Results indicate that although the agent received large error signals initially, it responded rapidly and effectively reduced the control error. After maintaining RL-based control for 2.06 seconds, the discharge was terminated due to the activation of the protection mechanism, triggered when the current in the PF4 coil exceeded the safety threshold of $12.5kA$. To mitigate this issue, additional preprocessing steps were implemented on the input data. Two specific measures were adopted: first, the scaling factor of the X-point magnetic flux was increased



(a) $\delta\psi$ (b) B_X (c) Ip_{error}

FIG. 3. Time Evolution of Controlled Variable Error During RL Control in Shot #152246. The horizontal axis represents the time elapsed since the start of RL control.

from 0.005 to 0.01; second, the error signal at the X-point was processed through a high-pass filter with a cut-off time constant $\tau_d = 0.7s$ to enhance its responsiveness to rapidly varying high-frequency components. The corresponding experimental results are presented in Fig. 4, Fig. 5.

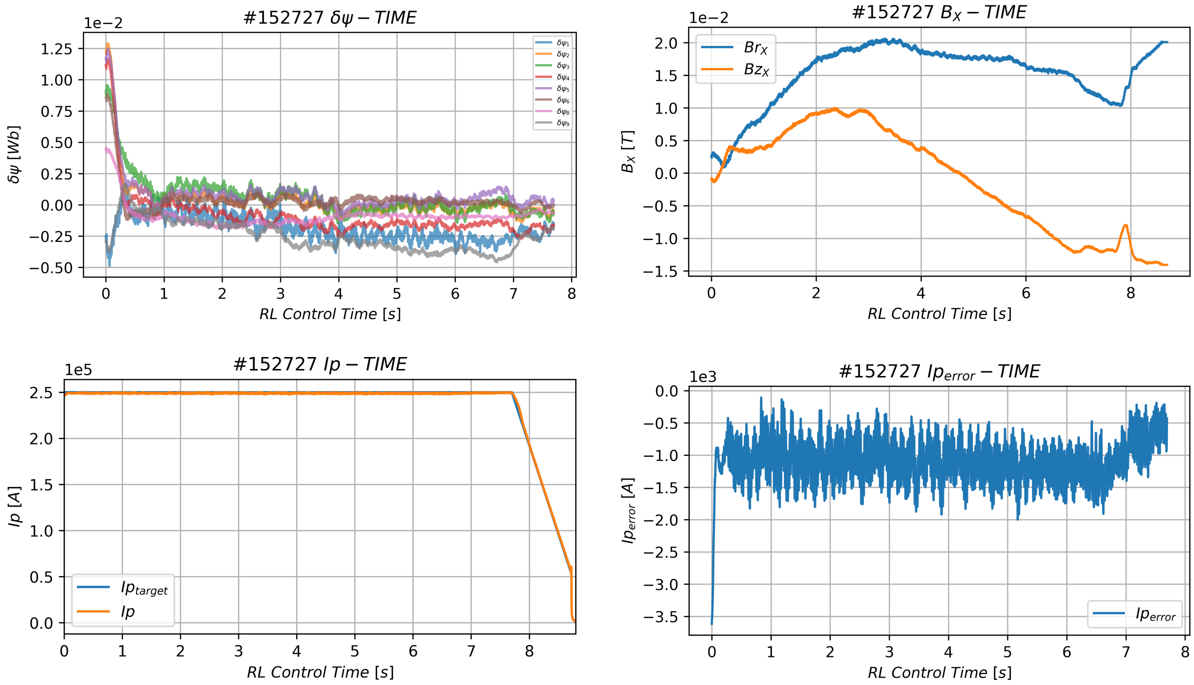
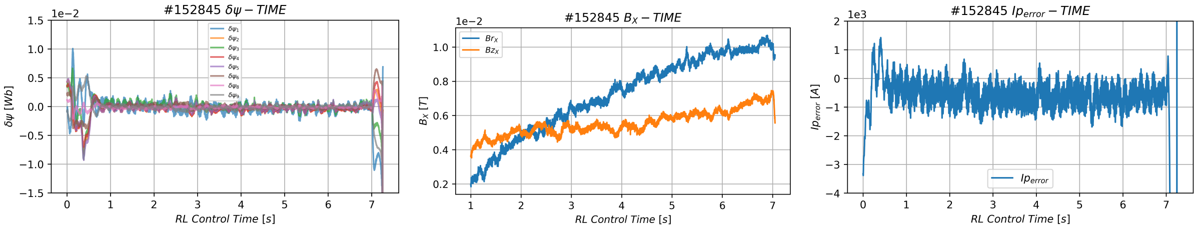


FIG. 4. Time Evolution of Controlled Variable Error During RL Control in Shot #152727 of Scheme I.

Following systematic data preprocessing, the discharge duration was significantly extended, and the RL controller demonstrated robust performance. Taking discharge experiment #152727 as an example, RL-based control was maintained until the current decay phase and was terminated only when the plasma current dropped to approximately 75 KA due to an overcurrent in the PF6 coil. It is worth noting that during training, the agent received a fixed reference signal indicating Ip of 250 KA. The experimental results indicate that the controller exhibits a notable degree of generalization capability. Its control mechanism relies on feedback-based error signals, emphasizing dynamic regulation of system deviations rather than direct tracking of absolute parameter values. When the operating current changes, the plasma system transitions to a new equilibrium state with altered dynamic response characteristics. Nevertheless, the proposed agent effectively adapts to such variations, achieving stable and robust control of the system state. In discharge experiment #152845, the agent successfully completed the full discharge process.

Building on this result, we further increased the Ip setpoint to evaluate the agent's ability to maintain plasma equilibrium and shape control under higher current conditions, showd in Fig. 6.The results demonstrate that although the agent was trained around an equilibrium at 250 KA, it maintained stable steady-state control even at the elevated current of 350 KA.

For the three distinct input signal strategies described above, we systematically evaluated their control perfor-



(a) $\delta\psi$ (b) B_X (c) Ip_{error}

FIG. 5. Time Evolution of Controlled Variable Error During RL Control in Shot #152845 of Scheme II.

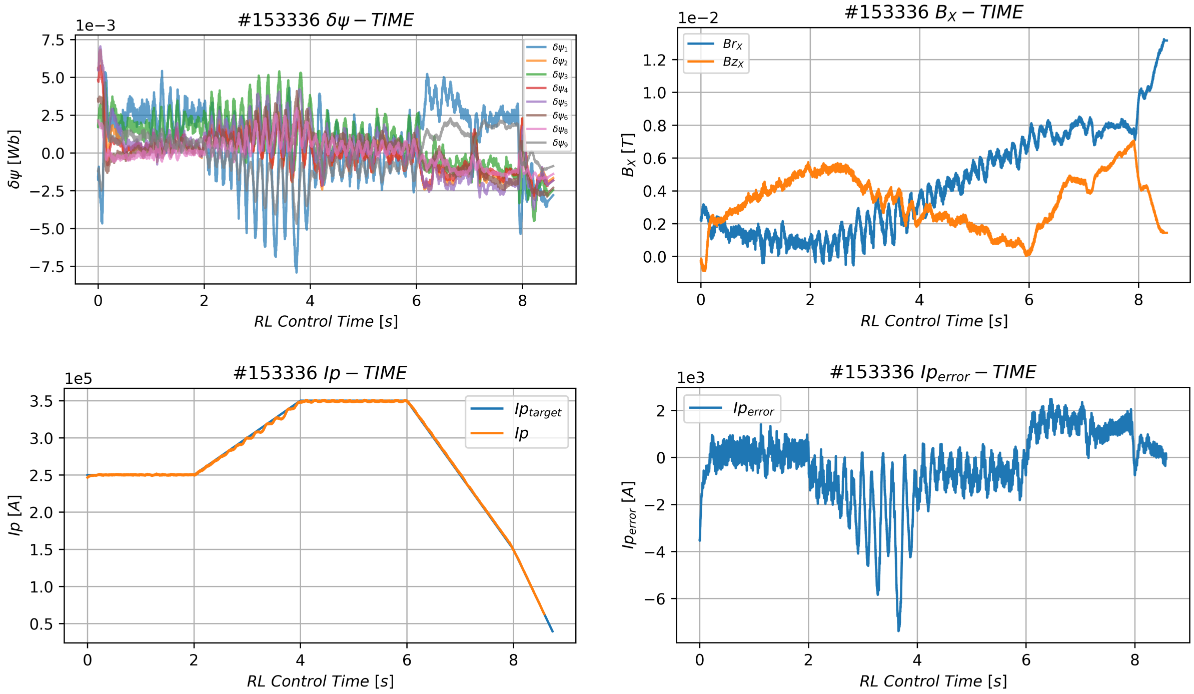


FIG. 6. Time Evolution of Controlled Variable Error During RL Control in Shot #153336.

mance using the root mean square error (RMSE) as a quantitative metric. The statistical results are summarized in Table 1.

TABLE 1. COMPARISON OF CONTROLLED VARIABLE ERRORS

	$\delta\psi[cm]$	$B_X[T]$	$Ip[KA]$
#152246	2.13	2.06×10^{-3}	0.91
#152727	0.96	1.34×10^{-2}	1.17
#152845	2.18	7.12×10^{-3}	0.97
#153336	5.15	4.32×10^{-3}	1.86

The analysis indicates that although the preprocessed input signals allowed a complete discharge, they led to some reduction in control accuracy. Therefore, achieving full discharge while preserving the original error signals remains a key objective for our future research.

5. CONCLUSION

The simplest RZIP model under a rigid assumption was utilized as the RL training environment. By applying the PPO algorithm, we trained a control model that fulfills experimental requirements. Experimentally, we achieved full discharge scenarios through control of plasma position and shape. Furthermore, successful enhancement of the plasma current demonstrated a degree of system robustness. However, an overcurrent condition in the poloidal field coils—unforeseen during training—posed a challenge. Although additional input processing was applied to ensure comprehensive controllability at the cost of minor accuracy loss, this workaround remains provisional, indicating significant room for improvement. Attaining precise configuration control with agents that reliably meet real-world demands remains the ultimate objective. The underlying issue is attributed to out-of-distribution (OOD) data, which the agent had not previously encountered. Nevertheless, these findings confirm that the linear equilibrium evolution model exhibits sufficient fidelity to support transferable controller development, justifying this approach for testing control strategies in future devices. The OOD challenge represents one of the prominent sim-to-real gaps in RL. Incorporating historical experimental data into the reinforcement learning experience replay buffer may alleviate this issue. The current training framework allows for minor adjustments to enable AI-assisted control during discharge. Future work will focus on developing a basic control policy integrated with real-time reinforcement learning optimization to enhance robustness and control capability.

Reinforcement learning offers a paradigm shift in multiple-input multiple-output (MIMO) control—by transitioning from iterative decoupling to requirement-driven design, it substantially reduces the workload for researchers. A more stable and realistic plasma parameter evolution model is essential to support broader control objectives, such as density control, enabling holistic plasma optimization and more stable discharges.In summary, RL-based control remains a highly attractive alternative. This approach has the potential to become a standard tool for routine discharge operation, with numerous promising pathways for further development.

ACKNOWLEDGEMENTS

This work is supported by the National MCF Energy R&D Program of China (Grant Nos. 2024YFE03020003 and 2022YFE03010002), the National Natural Science Foundation of China (Grant Nos.12575247, 12575246 and 11905256), the Institute of Energy, Hefei Comprehensive National Science Center under Grant No. 24KZS304, and the funding support from International Atomic Energy Agency with Research Contract Number of 26478.

REFERENCES

[1] John Wesson. *Tokamaks*. Oxford University Press, 2011. ISBN: 978-0-19-959223-4.

[2] Q.P. Yuan et al. “Plasma current, position and shape feedback control on EAST”. In: *Nuclear Fusion* 53.4 (2013), p. 043009. DOI: 10.1088/0029-5515/53/4/043009.

[3] Gianmaria De Tommasi. “Plasma Magnetic Control in Tokamak Devices”. In: *Journal of Fusion Energy* 38.3–4 (2018), pp. 406–436. DOI: 10.1007/s10894-018-0162-5.

[4] Y. Guo et al. “Preliminary results of a new MIMO plasma shape controller for EAST”. In: *Fusion Engineering and Design* 128 (2018), pp. 38–46. ISSN: 0920-3796. DOI: https://doi.org/10.1016/j.fusengdes.2018.01.025.

[5] G. Ambrosino and R. Albanese. “Magnetic control of plasma current, position, and shape in Tokamaks: a survey or modeling and control approaches”. In: *IEEE Control Systems Magazine* 25.5 (2005), pp. 76–92. DOI: 10.1109/MCS.2005.1512797.

[6] R. Ambrosino et al. “Model-based MIMO isoflux plasma shape control at the EAST tokamak: experimental results”. In: *2020 IEEE Conference on Control Technology and Applications (CCTA)*. 2020, pp. 770–775. DOI: 10.1109/CCTA41146.2020.9206391.

[7] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *Nature* 529.7587 (Jan. 2016), pp. 484–489. DOI: 10.1038/nature16961.

[8] Xue Bin Peng et al. “Sim-to-Real Transfer of Robotic Control with Dynamics Randomization”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)* (2017), pp. 1–8.

[9] Yifan Zhang et al. “Cost-Sensitive Portfolio Selection via Deep Reinforcement Learning”. In: *IEEE Transactions on Knowledge and Data Engineering* 34.1 (2022), pp. 236–248. DOI: 10.1109/TKDE.2020.2979700.

[10] Xiang (Ben) Song, Bin Zhou, and Dongfang Ma. “Cooperative traffic signal control through a counterfactual multi-agent deep actor critic approach”. In: *Transportation Research Part C: Emerging Technologies* 160 (2024), p. 104528. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2024.104528.

[11] Long Ouyang et al. “Training language models to follow instructions with human feedback”. In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS ’22. New Orleans, LA, USA: Curran Associates Inc., 2022. ISBN: 9781713871088.

[12] Jonas Degrave et al. “Magnetic control of tokamak plasmas through deep reinforcement learning”. In: *Nature* 602.7897 (2022), pp. 414–419. DOI: 10.1038/s41586-021-04301-9.

[13] Jaemin Seo et al. “Avoiding fusion plasma tearing instability with deep reinforcement learning”. In: *Nature* 626.8000 (2024), pp. 746–751. DOI: 10.1038/s41586-024-07024-9.

[14] Brendan D. Tracey et al. “Towards practical reinforcement learning for tokamak magnetic control”. In: *Fusion Engineering and Design* 200 (2024), p. 114161. ISSN: 0920-3796. DOI: https://doi.org/10.1016/j.fusengdes.2024.114161.

[15] Y C Zhang et al. “Real-time feedback control of p based on deep reinforcement learning on EAST”. In: *Plasma Physics and Controlled Fusion* 66.5 (2024), p. 055014. DOI: 10.1088/1361-6587/ad3749.

[16] S. Dubbioso et al. “A Deep Reinforcement Learning approach for Vertical Stabilization of tokamak plasmas”. In: *Fusion Engineering and Design* 194 (2023), p. 113725. ISSN: 0920-3796. DOI: https://doi.org/10.1016/j.fusengdes.2023.113725.

[17] W Yuehang. “EAST shape and position control optimization and system identification”. PhD thesis. Ph. D. dissertation, Dept. Plasma Phys., Univ. Sci. Technol. China, Hefei, China, 2018.

[18] B.J. Xiao et al. “EAST plasma control system”. In: *Fusion Engineering and Design* 83.2 (2008), pp. 181–187. ISSN: 0920-3796. DOI: https://doi.org/10.1016/j.fusengdes.2007.12.028.

[19] John Schulman et al. “Proximal Policy Optimization Algorithms”. In: *ArXiv* abs/1707.06347 (2017).

[20] John Schulman et al. “High-Dimensional Continuous Control Using Generalized Advantage Estimation”. In: (June 2015). DOI: 10.48550/arXiv.1506.02438.

[21] Y. Huang, B. J. Xiao, Z. P. Luo, et al. “Implementation of GPU parallel equilibrium reconstruction for plasma control in EAST”. In: *Fusion Engineering and Design* 112 (2016), pp. 1019–1024.

[22] B. Xiao, Q. Yuan, Z. Luo, et al. “Enhancement of EAST plasma control capabilities”. In: *Fusion Engineering and Design* 112 (2016), pp. 660–666.

[23] B. J. Xiao, Q. P. Yuan, D. A. Humphreys, et al. “Recent plasma control progress on EAST”. In: *Fusion Engineering and Design* 87.12 (2012), pp. 1887–1890.

[24] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[25] Tian Jin et al. *Compiling ONNX Neural Network Models Using MLIR*. 2020. arXiv: 2008.08272 [cs.PL].