Fusion Engineering and Design 194 (2023) 113725

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

A Deep Reinforcement Learning approach for Vertical Stabilization of tokamak plasmas

S. Dubbioso [a,b,*], G. De Tommasi [a,b], A. Mele [c], G. Tartaglione [d,b], M. Ariola [d,b], A. Pironti [a,b]

[a] Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, Università degli Studi di Napoli Federico II, Napoli, Italy
[b] Consorzio CREATE, Napoli, Italy
[c] Dipartimento di Economia, Ingegneria, Società e Impresa, Università degli Studi della Tuscia, Viterbo, Italy
[d] Dipartimento di Ingegneria, Università degli Studi di Napoli Parthenope, Naples, Italy

ARTICLE INFO

Keywords:
Plasma vertical stabilization
Deep reinforcement learning
Hyper-parameters tuning

ABSTRACT

Reinforcement Learning has emerged as a promising approach to implement efficient data-driven controllers for a variety of applications. In this paper, a Deep Deterministic Policy Gradient (DDPG) algorithm is used to train a Vertical Stabilization agent, to be considered as a possible alternative to the model-based solutions usually adopted in existing machines. The agent is trained and validated considering the ITER tokamak magnetic control as case study environment. The tuning of the DDPG algorithm's hyper-parameters is motivated through a sensitivity analysis.

1. Introduction

The axisymmetrical magnetic control of plasma is a well-understood problem in tokamak control [1,2], as well as the Vertical Stabilization (VS) one, which occurs when vertically unstable elongated plasmas are pursued. Such a control problem is usually solved by means of model-based control techniques, which rely on control-oriented models describing the response of the plasma and of the surrounding conductive structures. In order to achieve the required robustness, it is very common to resort to adaptive control strategies that take into account the features of the considered plasma scenario and of the specific experimental device. As an example see [3] for the VS system at the JET, [4] for the DIII-D system, [5] for ITER, and [6] for DEMO.

Data-driven approaches represent an alternative to achieve the required level of robustness. Indeed, a possibility is to exploit the capability of Reinforcement Learning (RL) algorithms to learn from data in order to obtain a single VS agent able to robustly deal with the different plasma operating conditions.

RL is a framework that allows solving control problems by making an agent (the controller) interact with the environment (the plant), via a *trial and error* strategy, until an optimal control policy is reached [7]. In particular, this approach allows to specify control goals in terms of a scalar reward function, meaning that the agent is not told which actions to take *a priori*, but it decides what to do based on the observations coming from the environment and a reward signal expressing how well it has performed (see Fig. 1). The training procedure aims at maximizing the cumulative long-term reward, i.e. the sum of the episode rewards in the long run.

A preliminary attempt to solve the VS problem using a RL approach is reported in [8], where a tabular approach, based on the Q-learning algorithm, was adopted. An application of Deep Reinforcement Learning (DRL) to the whole plasma magnetic control system can be found in [9], where an unprecedented control architecture design is described and experimentally validated on TCV.

In this paper, we investigate the applicability of DRL to the VS problem. With respect to the standard Q-learning algorithm used in [8], the DDPG algorithm adopted here allows considering continuous action and state spaces, essential for a fair representation of the plasma behavior. Moreover, in addition to what has been proposed in [8], not only the VS system but the whole ITER plasma magnetic control is taken into account during the training process.

The main objective of this paper is not only to obtain a data-driven VS agent and validate it against scenarios not considered during the training but primarily to highlight the strategy adopted to tune the algorithm hyper-parameters. Indeed, even if their tuning plays an important role in eliciting the best results, the optimal choice or the range of values considered is often not reported in the literature (refer to [10] for a wider analysis).

The rest of the article is organized as follows. The next section describes in detail how the DRL has been exploited to solve the VS problem for the ITER tokamak. The validation of the proposed solution is presented in Section 3, where a sensitivity analysis on the RL

* Corresponding author at: Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, Università degli Studi di Napoli Federico II, Napoli, Italy.
E-mail address: sara.dubbioso@unina.it (S. Dubbioso).

https://doi.org/10.1016/j.fusengdes.2023.113725
Received 12 October 2022; Received in revised form 13 March 2023; Accepted 3 April 2023
Available online 12 April 2023

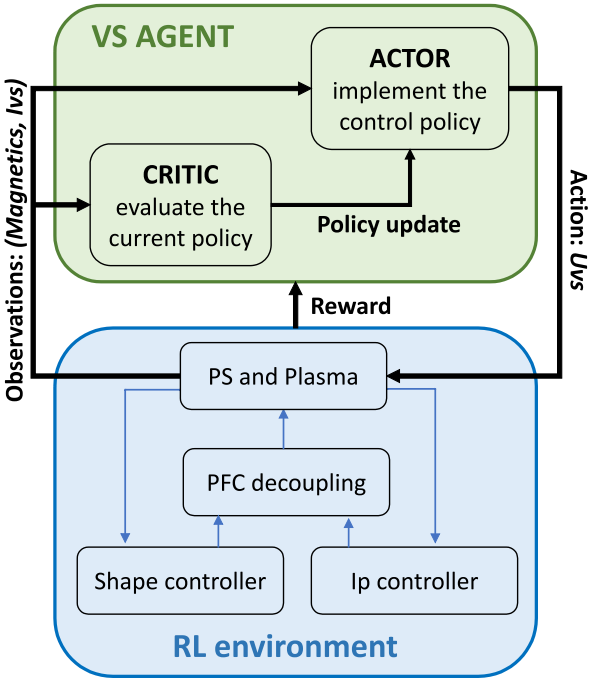

Fig. 1. DDPG scheme of interaction between the RL environment of plasma magnetic control and the actor–critic VS agent.

training hyper-parameters is also reported. Some conclusive remarks are eventually given in Section 4.

2. Simulation setup

A scheme showing how the DRL agent has been applied to the ITER VS is shown in Fig. 1. The agent interacts with a RL environment, consting of a linear model of the plasma dynamics generated by the CREATE-NL equilibrium code [11], the power supplies model, and which includes the interaction of the VS system with the other magnetic control loops, i.e. the plasma current and shape controllers.

More in detail, the considered plasma linearized model can be written as

$$\dot{x}(t) = Ax(t) + Bu(t) + E\dot{w}(t) \tag{1a}$$

$$y(t) = Cx(t) + Fw(t), \tag{1b}$$

where:

- the state vector $x = \left(\delta I_{PF}^T \quad \delta I_e \quad \delta I_p\right)^T$ holds the variations of the PF currents, of the currents in the passive conductive structures, and of the plasma current. The current in the PF circuits can be further partitioned as $\delta I_{PF} = \left(\delta I_{SC}^T \quad \delta I_{VS}\right)^T$, where δI_{SC} is the vector of the currents variations in the ex-vessel superconductive circuits used for plasma current and shape control, while δI_{VS} is the current variation in the in-vessel copper-made circuit dedicated to vertical stabilization. It is worth noticing that, since the linear model describes the plasma behavior around a given equilibrium, it is $\delta I_{VS} = I_{VS}$, being the equilibrium in-vessel current equal to zero;
- the input vector $\delta u_{PF} = \left(\delta u_{SC} \quad \delta u_{VS}\right)^T$ consists of the voltage variations applied to the control circuits. As for the current in the in-vessel coils, given that all the input voltages are equal to zero at the equilibrium, it turns out that $\delta u_{PF} = u_{PF} = \left(u_{SC} \quad u_{VS}\right)^T$;
- $w = \left(\delta\beta_p \quad \delta l_i\right)^T$ is the vector of the variations of poloidal beta β_p and internal inductance l_i, that can be regarded as exogenous disturbances, as far as magnetic control is concerned.

The simulation scheme used in this work as training and validation environment for the proposed VS agent is shown in Fig. 2.

The actuator considered for the VS agent is the in-vessel circuit, with the control action as $a = u_{VS}$. The feedback signals are gathered in the observed vector $s = (I_{VS}, y_{mag})$, and are, respectively, the current in the VS circuit I_{VS}, and the vector of magnetic diagnostic signals y_{mag}, that are part of the model outputs and normally used to reconstruct the plasma centroid vertical position Z_c and velocity $\dot{Z}_c$. Specifically, the measurements of in-vessel Mirnov coils and flux sensors are included in y_{mag}.

The RL algorithm implemented is a model-free and off-policy method called DDPG. The DDPG agent is an actor-critic method, which simultaneously learns the control policy function and evaluates it (see Fig. 1). The actor and critic neural networks are feed-forward networks with

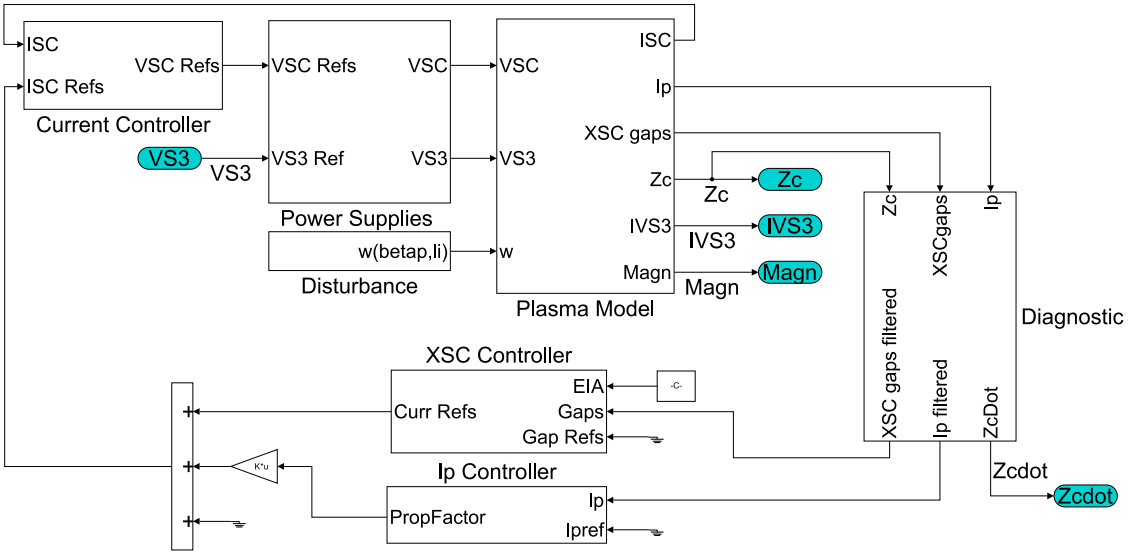

Fig. 2. Simulation scheme used as environment for both training and validation of RL-based VS agent. The scheme includes the plasma-coils model, the model of the power supplies as well as the controllers for both plasma current and shape.

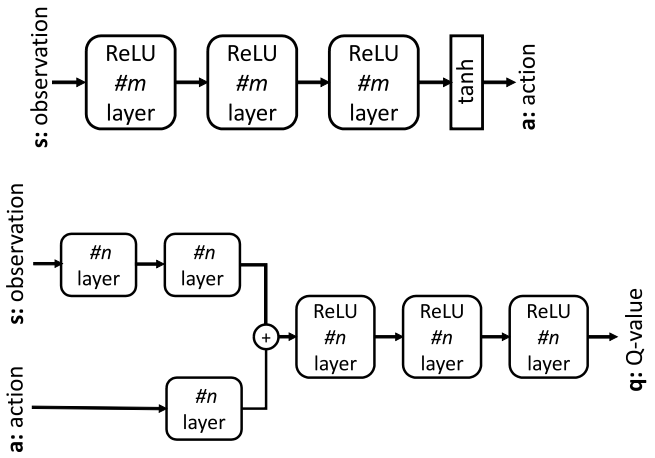

Fig. 3. Actor and Critic networks architecture. The networks are only feed-forward, with no recurrent element, and have been implemented using fully connected layers.

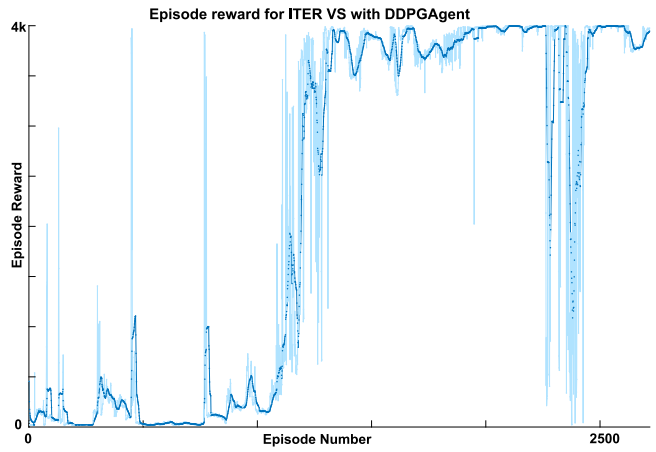

Fig. 4. Episodes cumulative rewards obtained with the best choice of the considered hyper-parameters and for the reward function coefficients set equal to $k_1 = 1, k_2 = 2, k_3 = 1$ (which also correspond to the setup considered for *Training A* in Section 3.2).

fully connected layers and they have been implemented as reported in Fig. 3. Rectified Linear Unit (ReLU) activation functions have been chosen, defined as $ReLU(x) = \max(x, 0)$.

The chosen reward function is chosen as a function of the agent state s and action a, and is given by:

$$R(s, a) = -k_1 \cdot \left(\frac{\dot{Z}_c(y_{mag})}{\dot{Z}_{c_{\max}}} \right)^2 - k_2 \cdot \left(\frac{I_{VS}}{I_{VS_{\max}}} \right)^2 - k_3 \cdot \left(\frac{u_{VS}}{u_{VS_{\max}}} \right)^2 \quad (2)$$

where $\dot{Z}_{c_{\max}}$, $I_{VS_{\max}}$ and $u_{VS_{\max}}$, respectively, refer to the maximum values specified for the plasma centroid vertical speed and the in-vessel coils current and voltage. A proper value for $\dot{Z}_{c_{\max}}$ was found by analyzing the results of past simulations carried out with the model-based VS system proposed in [12].

The reward function (2) reflects the main objective of a VS system i.e. to stop the unstable vertical motion of the plasma to avoid disruption while keeping the in-vessel current as low as possible, and limiting the control voltage. An additional penalty is then added to the reward (2) and the training episode is terminated if the centroid position variation with respect to the equilibrium value δZ_c exceeds a threshold over which disruption cannot be avoided. Furthermore, a +2 bonus is added to (2) at each simulation time step if the agent manages to keep δZ_c within the prescribed bound. These bonuses, summed over all the episode time samples, turn into a maximum value for the cumulative reward that becomes positive. In particular, given the sampling time $T_s = 2.5$ ms, the possible maximum cumulative reward could be 4000 for the episodes whose duration is equal to 5 s (see Figs. 4, 6 and 7), while if the episode duration is 20 s the maximum could reach 16000 (see Fig. 5).

Moreover, at each time step of the DDPG training process the expected reward y_i is computed as

$$y_i = R_i + \Gamma Q(s_{i+1}, a_{i+1}) \quad (3)$$

where R_i is the experienced reward at the ith step, Γ is the discount factor and $Q(s_{i+1}, a_{i+1})$ is the action-value function predicted by the critic network. The discount factor in RL algorithm serves to specify how much the agent cares about fuure rewards with respect to the immediate one. In (3), Γ is used to scale the value of Q that represents the estimated future cumulative rewards.

The described setup has been implemented in MATLAB® by using the Reinforcement Learning Toolbox® [13], in order to exploit Simulink® to integrate the VS agent with the other components of the ITER plasma magnetic control, already available in this environment.

3. Results

In this section, the training of the VS agent and its validation are discussed. In particular, the sensitivity analysis which allowed to choose the best-performing set of training hyper-parameters is reported in Section 3.1, while the validation of the obtained VS agents is illustrated in Section 3.2.

3.1. Sensitivity analysis

The effects of some hyper-parameters and their tuning are analyzed with respect to reward convergence. This study allowed finding the set of parameters that led to the successful training of the VS agent. Specifically, in this paper, we consider the effect of the following hyper-parameters: episode duration, number of hidden layers for both the actor and critic networks and action-noise variance decay rate. Table 1 reports the setting of all the DDPG hyper-parameters and the range of variation for those that have been changed during our analysis.

Fig. 4 reports the *training graph*, i.e. the trace of the cumulative reward as a function of the ith episode, when the coefficients in (2) are set equal to $k_1 = 1, k_2 = 2, k_3 = 1$, and the *optimal choice* for the three considered hyper-parameters has been made. In particular, the latter have been set equal to the values reported in bold in Table 1.

In the following, for each hyper-parameter variation considered the corresponding training graph is reported, and a brief discussion is made to motivate our choice. Notice that in the training graph not only the trace of the cumulative reward as a function of the ith episode is reported (lighter trace), but also the averaged cumulative reward (darker trace) over the 20 most recent episodes

Episode duration

Initially, the duration of an episode has been set equal to 20 s; the corresponding training is shown in Fig. 5. Comparing this training with the one shown in Fig. 4, it can be seen that a longer interaction between the DDPG agent and the plasma environment leads to higher rewards, but does not ensure convergence toward an optimum. Moreover, when the episode duration is set equal to 20 s, the obtained agents focus more on satisfying the performance at steady-state, rather than during the transient. Therefore, an episode duration of 5 s has been chosen for the agent training procedure.

Table 1
Set of the DDPG hyper-parameters. The range of variations exploited during the sensitivity analysis is specified for those parameters whose setting was changed. When multiple values are specified, those reported in **boldface** are the ones chosen to obtain the results reported in Fig. 4.

Hyper-parameter	Considered values	
Sampling time T_s	2.5 ms	
Episode duration T	**5 s**	20 s
Actor learning rate	5×10^{-4}	
Critic learning rate	10^{-3}	
Actor hidden layers #m	**64**	128
Critic hidden layers #n	**32**	128
Discount factor Γ	0.99	
Batch size	256	
OUP variance	1840	
OUP variance decay rate	$\mathbf{8.66 \times 10^{-6}}$	3.5×10^{-6}

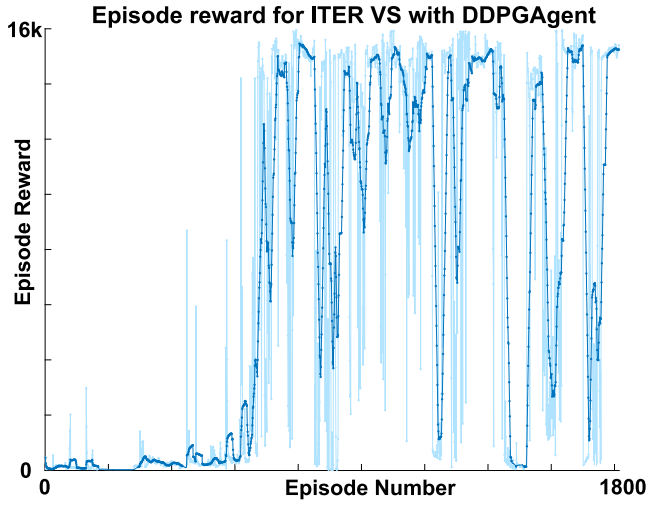

Fig. 5. Episodes cumulative rewards obtained with an episode duration of 20 s.

OUP variance decay rate

The agent uses the Ornstein–Uhlenbeck action noise model for exploration. The noise variance and its decay rate are computed as

$$\sigma^2 \cdot \sqrt{T_s} = (1\%\ to\ 10\%)\ of\ \Delta A$$

while its half-life, in time steps, is given by

$$HL = \frac{\ln(0.5)}{\ln(1-\sigma_{dr}^2)}$$

where σ^2 is the noise variance, ΔA is the range of the action variable, HL is the noise half-life and σ_{dr}^2 is the noise variance decay rate. Two values of the decay rate have been considered in our analysis. This parameter has been first set equal to 3.5×10^{-6}, which is equivalent to about 2×10^5 time samples. The resulting training, reported in Fig. 6, shows that even if the exploration seems to terminate after about 1000 episodes, there is a sudden drop in the reward value between episodes 1700 and 2200, after which the agent does not fully recover. On the other hand, when the decay rate is set equal to 8.66×10^{-6}, corresponding to a variance half-life of about 8×10^4 time samples, the results shown in Fig. 4 are obtained. In this case, once the plateau is reached, the behavior of the reward oscillates less up to episode 2500. Therefore, in our setup, we set the decay rate equal to 8.66×10^{-6}.

Critic and actor hidden layers size

The first choice of the size for the fully connected layers in both the critic and actor networks architecture was equal to 128, as shown in Fig. 3. From the training reported in Fig. 7, it appears that the network architectures can significantly impact the results and the convergence and that considering a simpler network can produce better

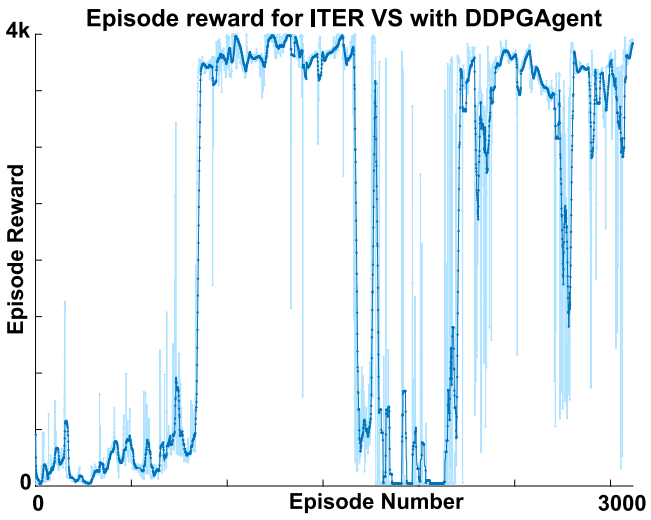

Fig. 6. Episodes cumulative rewards obtained with a agent noise decay rate of 3.5×10^{-6}.

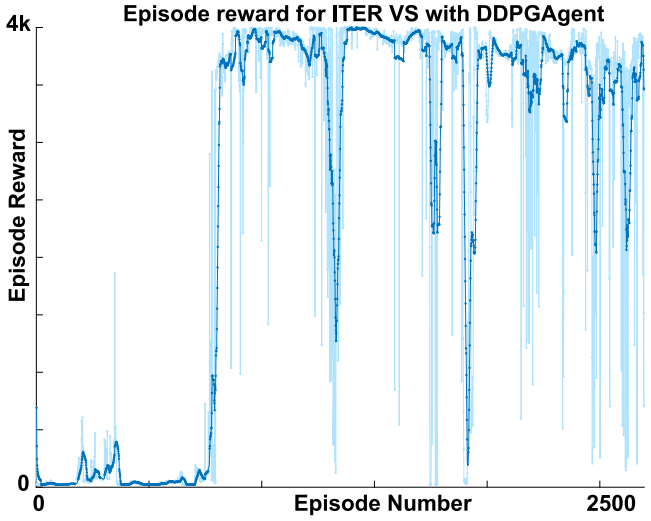

Fig. 7. Episodes cumulative rewards obtained with, both actor and critic networks implemented using fully connected layers with a size 128.

results. Hence, in the training reported in Fig. 4, 64 layers were chosen for the actor, and 32 for the critic.

3.2. Agents validation

For the agent validation, two equilibria different from the one used during the training have been taken into account. The nominal values of the plasma parameters for which the corresponding free-boundary equilibrium problem has been solved are reported in Table 2. All the considered equilibria correspond to different time instants of a 15 MA ITER discharge. *Eq. #1* (the one used for training) and *Eq. #2* are two different snapshots taken at the beginning of the 15 MA flat-top, while *Eq. #3* refers to the end of the flat-top. More in detail, *Eq. #2* is taken before the transition from low to high confinement, while *Eq. #1* is an equilibrium taken soon after such a transition. As for the linear state space of *Eq. #1*, also the models of the two validation equilibria have been generated by the CREATE-NL equilibrium code.

In addition to an agent obtained from the training shown in Fig. 4 (hereafter referred to as *Training A*), two more agents have been selected; these correspond to two alternative choices of the reward function aimed at improving the VS performance, and in particular at

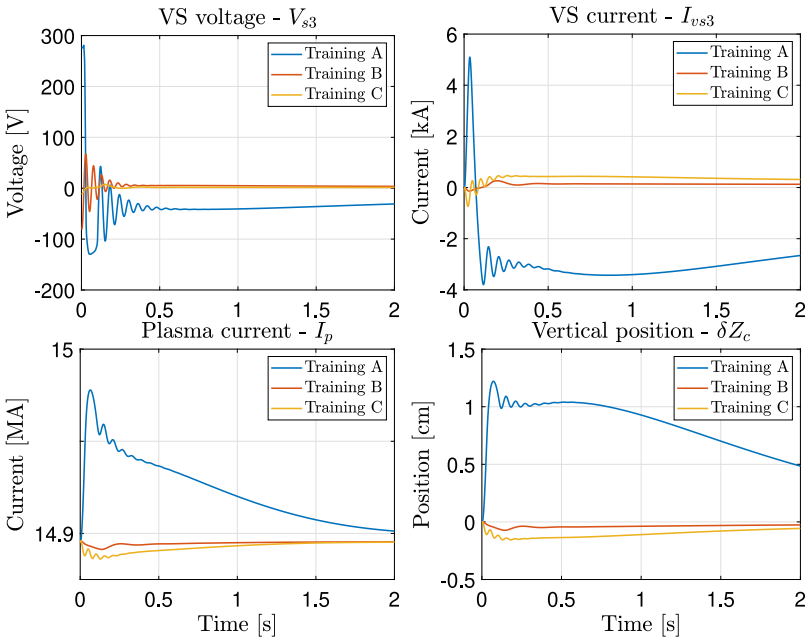

Fig. 8. Simulation results obtained with *Eq. #2* for the VS agents corresponding to *Training A* (blue trace), *Training B* (blue trace) and *Training C* (red trace). The u_{VS} voltage, the I_{VS} current, the plasma current I_p, and the vertical displacement of the plasma centroid δZ_c with respect to the equilibrium are shown.

Table 2
Plasma parameters for the equilibria considered for the VS agents training and validation. They correspond to an equilibrium I_p of 15 MA, while the table reports the values of the profile parameters $\beta_{p_{eq}}$ and $l_{i_{eq}}$, and of plasma growth rate γ.

ITER equilibrium	$\beta_{p_{eq}}$	$l_{i_{eq}}$	γ
Eq. #1 (Training)	0.66	0.88	4.9 s^{-1}
Eq. #2 (Validation)	0.08	0.92	9.1 s^{-1}
Eq. #3 (Validation)	0.82	0.71	2.9 s^{-1}

Table 3
Values of the penalty parameters in the reward function (2) for the different considered training.

	$\mathbf{k_1}$	$\mathbf{k_2}$	$\mathbf{k_3}$	$\mathbf{k_4}$
Training A	1	2	1	0
Training B	1	2000	1	0
Training C	1	10	1	20

reducing the steady-state current in the in-vessel coils. Namely, *Training B*, corresponds to a higher value of k_2, while *Training C* was obtained by adding the term $-k_4 \cdot \left(\frac{\overline{I}_{VS}}{\overline{I}_{VS_{\max}}}\right)^2$ to the reward function. This additional term allows penalizing also the integral value $\overline{I}_{VS} = \frac{1}{T}\int_0^T I_{VS}(\tau)d\tau$ of the current in the in-vessel coils. The reward function parameters for the considered training are summarized in Table 3. A preliminary performance assessment of the three considered agents is reported in what follows.

Validation without disturbances

The model corresponding to *Eq. #2* is used to assess the capability of the various agents to stabilize a plasma different from the one used for training when no external disturbances are applied. At the beginning of the considered simulation, the plasma starts from the considered equilibrium. The RL-based agent, however, can lead to small initial displacement in the plasma position that needs to be actively compensated. Fig. 8 shows the in-vessel circuit voltage u_{VS} and current I_{VS}, the plasma current I_p, and the variation of the plasma centroid vertical position with respect to the equilibrium value δZ_c for the considered training options. It can be noticed that, although all the considered agents achieve the stabilization objective, the ones corresponding to *Training B* and *Training C* are preferable, mainly because they require a lower steady-state in-vessel coil current since they bring back the centroid closer to its equilibrium value.

Validation in presence of a vertical displacement event

Further validation is performed by considering the rejection of a Vertical Displacement Event (VDE). A VDE is an uncontrolled growth of the plasma unstable vertical mode. It can be considered as an instantaneous change in plasma vertical position and surrounding currents. Hence, it can be modeled by properly initializing the value of the plasma model (1) state. Specifically, in the case of a VDE the plasma model's initial state is obtained by computing the unstable eigenvector of the associated initial model and rescaling it so that the corresponding output on the Z_c channel has the desired amplitude (see also [14]).

For the agents validation, a VDE of 5 cm has been applied to *Eq. #3* and Fig. 9 shows the comparison between the results obtained using the three considered VS agents. Also in this case the agents corresponding to *Training B* and *Training C* allow to minimize the steady-state current in the in-vessel coil, confirming the results obtained with *Eq. #2*. Moreover, the agent corresponding to *Training C* shows a smoother behavior in terms of plasma current and vertical displacement variations.

Comparison with a linear VS

A comparison between a model-based linear VS algorithm and the validated VS agents is reported in Fig. 10. The former controller computes the voltage u_{VS} to be applied to the in-vessel coils as a combination of the plasma vertical speed $\dot{Z}_c$ and of the current I_{VS} flowing in the VS circuit. The interested reader can refer to [12,15] for more details; in particular in [12] a detailed discussion on how to tune the linear controller gains in order to improve robustness is given. In order to compare the two considered approaches, here we report the results of the simulation of a 5 cm VDE applied to *Eq. #3*. For this considered case, Fig. 10 shows that all the RL agents have performance similar to the model-based VS, in terms of settling time. Moreover, RL approaches require a lower control effort in terms of applied voltage, during the initial phase of the simulation, when the plasma displacement with respect to the equilibrium value is the maximum.

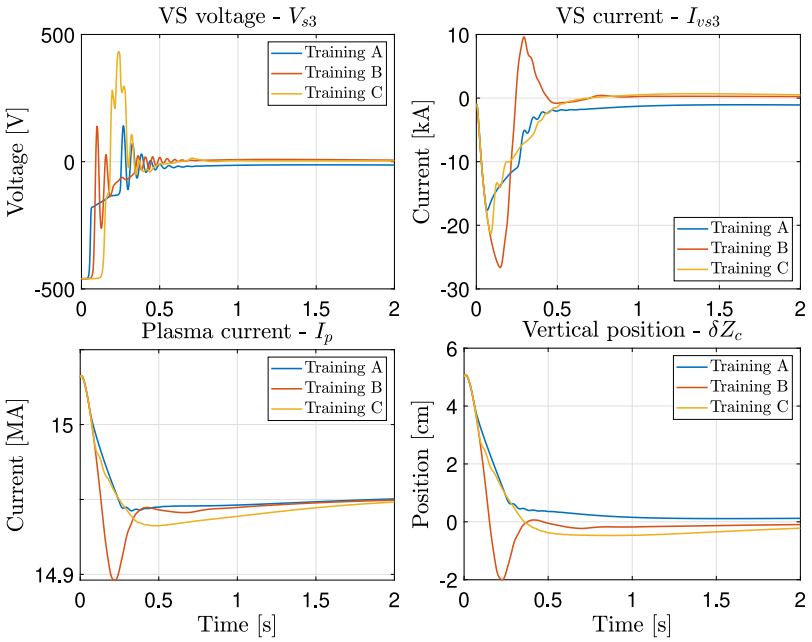

Fig. 9. Rejection of a 5 cm VDE applied to the linear model corresponding to *Eq. #3*. The time traces of the u_{VS} voltage, the I_{VS} current, the plasma current I_p, and the vertical displacement of the plasma centroid δZ_c with respect to the equilibrium are shown.

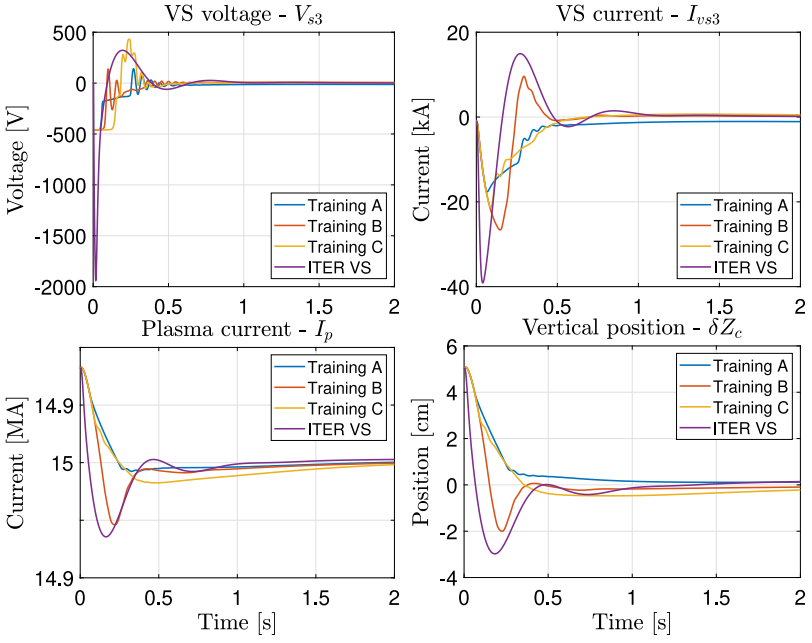

Fig. 10. Comparison between the model-based linear VS algorithm (purple trace) and the RL agents corresponding to *Training A* (blue trace), *Training B* (blue trace), *Training C* (red trace) in case of rejection of a VDE of 5 cm applied to *Eq. #3*.

4. Conclusions

In this work, we investigated the possibility of controlling the unstable vertical dynamic of a tokamak plasma by means of an RL-based controller. RL allowed the implementation of a single VS agent that can deal with different plasma operation conditions without the need of adapting the controller parameters. The tuning of the DDPG hyper-parameters was illustrated by means of a sensitivity analysis. Preliminary results obtained in simulation for the ITER VS system have been also shown.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 - EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them.

The work of Adriano Mele has been supported by the Italian Ministry for University and Research (MUR) under the European Social Fund REACT EU - PON Ricerca e Innovazione 2014–2020 (D.M. 1062/2021).

This work was partially supported by the Italian Research Ministry under the PRIN20177BZMAH.

References

[1] G. De Tommasi, Plasma magnetic control in Tokamak devices, Journal of Fusion Energy 38 (3–4) (2019) 406–436.
[2] M. Ariola, A. Pironti, Magnetic Control of Tokamak Plasmas, second ed., Springer, 2016.
[3] A. Neto, et al., Exploitation of modularity in the JET Tokamak vertical stabilization system, Control Engineering Practice 20 (9) (2012) 846–856.
[4] E. Schuster, et al., Plasma vertical stabilization with actuation constraints in the DIII-D Tokamak, Automatica 41 (7) (2005) 1173–1179.
[5] S. Gerkšič, G. De Tommasi, Vertical stabilization of ITER plasma using explicit model predictive control, Fusion Engineering and Design 88 (6–8) (2013) 1082–1086.
[6] W. Biel, et al., Development of a concept and basis for the DEMO diagnostic and control system, Fusion Eng. Des. 179 (2022) 113122.
[7] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.
[8] G. De Tommasi, et al., A RL-based vertical stabilization system for the EAST Tokamak, in: Proceeding of 2022 American Control Conference, 2022, pp. 5328–5333.
[9] J. Degrave, et al., Magnetic control of tokamak plasmas through deep reinforcement learning, Nature 602 (7897) (2022) 414–419.
[10] P. Henderson, et al., Deep reinforcement learning that matters, in: Proceeding of the 2018 AAAI Conference on Artificial Intelligence, 2018.
[11] R. Albanese, R. Ambrosino, M. Mattei, CREATE-NL+: A robust control-oriented free boundary dynamic plasma equilibrium solver, Fusion Engineering and Design 96–97 (2015) 664–667.
[12] R. Albanese, et al., ITER-like vertical stabilization system for the EAST Tokamak, Nuclear Fusion 57 (8) (2017) 086039.
[13] Matlab Reinforcement Learning Toolbox, The Mathworks, 2022, https://www.mathworks.com/help/reinforcement-learning.html.
[14] G. Ambrosino, et al., Plasma vertical stabilization in the ITER Tokamak via constrained static output feedback, IEEE Transactions on Control Systems Technology 19 (2) (2011) 376–381.
[15] G. De Tommasi, A. Mele, A. Pironti, Robust plasma vertical stabilization in Tokamak devices via multi-objective optimization, Optimization and Decision Science: Methodologies and Applications (2017) 305–314.